

The Asthma Mobile Health Study, a large-scale clinical observational study using ResearchKit

Yu-Feng Yvonne Chan^{1,2}, Pei Wang¹, Linda Rogers³, Nicole Tignor¹, Micol Zweig¹, Steven G Hershman⁴, Nicholas Genes^{1,2}, Erick R Scott¹, Eric Krock⁴, Marcus Badgeley¹, Ron Edgar⁴, Samantha Violante¹, Rosalind Wright^{3,5,6}, Charles A Powell³, Joel T Dudley^{1,7} & Eric E Schadt¹

The feasibility of using mobile health applications to conduct observational clinical studies requires rigorous validation. Here, we report initial findings from the Asthma Mobile Health Study, a research study, including recruitment, consent, and enrollment, conducted entirely remotely by smartphone. We achieved secure bidirectional data flow between investigators and 7,593 participants from across the United States, including many with severe asthma. Our platform enabled prospective collection of longitudinal, multidimensional data (e.g., surveys, devices, geolocation, and air quality) in a subset of users over the 6-month study period. Consistent trending and correlation of interrelated variables support the quality of data obtained via this method. We detected increased reporting of asthma symptoms in regions affected by heat, pollen, and wildfires. Potential challenges with this technology include selection bias, low retention rates, reporting bias, and data security. These issues require attention to realize the full potential of mobile platforms in research and patient care.

Three billion smartphones were in use worldwide in 2015, a figure expected to double by 2020 (ref. 1). Smartphones have replaced standard mail and landline phones for many people, creating a need to leverage mobile devices for research historically conducted by phone and mail. Mobile technology may also offer advantages over traditional data collection and management processes in research.

ResearchKit (Apple; Cupertino, CA, USA), an open source framework for mobile research can (i) obtain electronic informed consent, (ii) administer and collect questionnaires, (iii) actively and passively collect biometric data, (iv) provide reminders and notifications, and (v) reliably transmit and secure data in a central repository in compliance with regulatory requirements. Several research institutions and Sage Bionetworks (Seattle) collaborated with Apple to build the first mobile health applications using ResearchKit to demonstrate the feasibility of conducting research via this platform, and to provide an open source template to build third-party research apps^{2–5}. To this end, we developed the Asthma Health Application (AHA) and conducted the Asthma Mobile Health Study (AMHS).

As many as half of the 25 million Americans with asthma lack optimal asthma control, contributing to \$56 billion in annual disease costs⁶. A smartphone platform enabling large-scale, continuous collection of clinical, environmental, and passive biometric data may provide valuable insights for asthma research and clinical care. Our prospective observational mobile health study focused on assessing the following primary objectives: (i) feasibility of smartphone-based

recruitment; (ii) characteristics of a study cohort recruited through the ResearchKit platform; (iii) user engagement and retention patterns; and (iv) user data sharing preferences. We tested the quality and utility of self-reported data collected by this method by assessing correlation with trusted external sources and concordance with expected patterns. Lastly, we evaluated the reported clinical impact of AHA use in a subset of participants.

RESULTS

Study enrollment, user experience, and data sharing

After its Apple App Store release on March 9th 2015, the AHA was downloaded 49,963 times over the first 6 months, 40,683 of which were from United States. Only US residents were eligible for the study. **Figure 1** describes the AHA enrollment process, cohort and key sub-cohort definitions (see **Supplementary Table 1a** for a comprehensive description of study sub-cohorts), user experience, and the geographic distribution of the study participants. A total of 7,593 users, out of 8,524 completed the enrollment process. Participants were asked to complete a series of intake surveys on demographics, comorbidities, and asthma history over four consecutive days after enrollment. Participants were also asked to complete daily asthma surveys to log symptoms, presumed triggers, and medication adherence for the duration of the study. In addition, the AHA administered weekly surveys to capture participants' healthcare utilization (HCU) and quality of life over the previous 7 d. EQ-5D health questionnaire

¹Department of Genetics and Genomic Sciences, Icahn School of Medicine at Mount Sinai, New York, New York, USA. ²Department of Emergency Medicine, Icahn School of Medicine at Mount Sinai, New York, New York, USA. ³Department of Medicine, Pulmonary, Critical Care and Sleep Medicine, Icahn School of Medicine at Mount Sinai, New York, New York, USA. ⁴LifeMap Solutions, Inc., New York, New York, USA. ⁵Department of Pediatrics, Pulmonary and Critical Care, Icahn School of Medicine at Mount Sinai, New York, New York, USA. ⁶Department of Environmental Medicine & Public Health, Icahn School of Medicine at Mount Sinai, New York, New York, USA. ⁷Department of Population Health Science and Policy, Icahn School of Medicine at Mount Sinai, New York, New York, USA. Correspondence should be addressed to Y.-F.Y.C. (yu-fengyvonne.chan@mssm.edu) or E.E.S. (eric.schadt@mssm.edu).

Received 15 April 2016; accepted 13 February 2017; published online 13 March 2017; doi:10.1038/nbt.3826

and milestone surveys were also administered at less frequent intervals (see **Supplementary Table 1b** for survey descriptions and **Supplementary Note** for survey questions).

A total of 88% of participants chose to share de-identified data with researchers; 67% made their data available to all qualified researchers, and 21% shared data with study sponsors and partners. Study participants could send a copy of their own data to their e-mail by using the one-click 'Export Data' feature.

Different cohorts of AHA users and their baseline characteristics

Among the 7,593 enrolled participants, 6,470 responded to at least one survey in the study. We referred to these 6,470 users as the 'Baseline user' cohort. The 'Robust user' cohort ($n = 2,317$) included participants who: (i) were free from other lung disease and congestive heart failure, (ii) didn't smoke more than ten packs of cigarettes per year, and (iii) completed at least five daily or weekly surveys. Some of our analyses focused on the Robust users because of their more complete

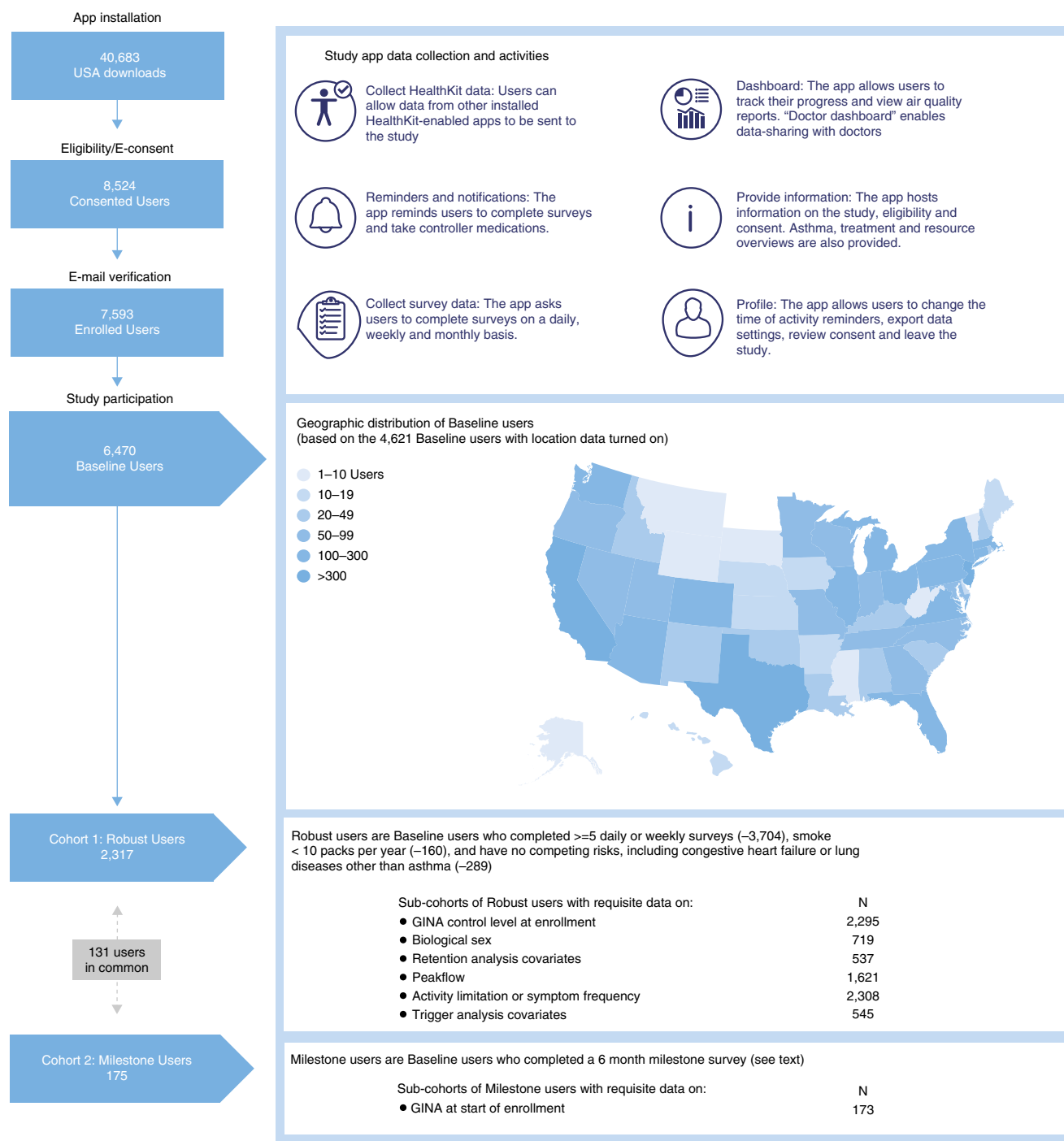


Figure 1 Recruitment process, user experience, and geographic distribution. The flow chart on the left indicates the recruitment process. The upper-right panel illustrates the activities experienced by users of AHA. The map illustrates the geographic distribution of Baseline users (i.e., Enrolled users who submitted at least one study survey). The box in the bottom outlines the selection of several of the key sub-cohorts used in our analyses, along with their sample sizes (see **Supplementary Fig. 1a** for more details).

baseline data, greater adherence to study survey completion, and fewer confounding variables.

The Milestone user cohort was a subset of ‘Enrolled users’ who completed the 6-month milestone survey. Of note, 131 out of the 175 Milestone users are included in the Robust user cohort. **Table 1** and **Supplementary Table 2** illustrate the demographic and baseline clinical characteristics of our Baseline, Robust, and Milestone users compared with population-based asthma statistics from the Centers for Disease Control and Prevention (CDC; Atlanta)⁶, (http://www.cdc.gov/asthma/most_recent_data.htm). AHA users tended to be younger, wealthier, more educated, and were more often male than asthma patients in the CDC asthma population. Based on the location data from 4,621 Baseline users, **Supplementary Figure 1a** shows the high correlation ($r = 0.85$) of asthma prevalence, by state, between AHA users and the CDC cohort.

Baseline users had a higher rate of hospitalization (6%) and emergency department visits (11%) in the 6 months before enrollment than CDC rates (2% and 8%, respectively; **Table 1**)³. Of note, 13% of Baseline users reported a history of intubation and 37% reported the use of oral steroids (in the 6 months before enrollment) to treat an asthma exacerbation (**Supplementary Table 2**). During the study, high

frequencies of symptoms were reported by a large number of users: (i) 47% of Baseline users reported symptoms in excess of twice weekly, (ii) 29% reported symptoms on most days or daily, and (iii) 37% reported the use of oral steroids (in the 6 months before enrollment) to treat an asthma exacerbation (**Supplementary Table 2**). Similar distributions were observed in Robust users and Milestone users. In addition, using Global Initiative for Asthma (GINA) criteria (<http://www.ginasthma.org/>) to assess asthma symptom control, uncontrolled asthma was reported in 43%, 44%, and 42% of Baseline users, Robust users, and Milestone users, respectively (Online Methods).

Download, enrollment, and retention patterns

Presumably, initial media publicity led to a high rate of AHA downloads, starting with 43,949 in the first month, yet decreasing to 300–400 per month by 5–6 months after the app launch. During months 3 to 6 of the study, the average monthly enrollment rate was 30% of downloads, with approximately 21 participants becoming Robust users per month in these later 3 months (**Supplementary Table 3a**).

Patients with worse asthma control enrolled at a relatively higher rate over time. For example, in the Robust user cohort, the proportion of GINA-uncontrolled participants in the first half of the study was

Table 1 Characteristics of patients at enrollment

Characteristic		AHA			CDC	
		Baseline	Robust	Milestone		% Dist.
Age ^a	18–34	756 (0.54)	369 (0.52)	56 (0.33)		36
	35–64	573 (0.41)	317 (0.44)	94 (0.56)		48
	65+	66 (0.05)	28 (0.04)	19 (0.11)		16
Gender ^a	F	539 (0.39)	285 (0.4)	80 (0.46)		59
	M	859 (0.61)	434 (0.6)	95 (0.54)		41
Race ^b	Black	203 (0.05)	108 (0.05)	8 (0.05)		14
	White	2,689 (0.68)	1,540 (0.7)	125 (0.77)		64
	Other	316 (0.08)	144 (0.07)	7 (0.04)		5
	Multirace	185 (0.05)	98 (0.04)	5 (0.03)		2
	Hispanic	587 (0.15)	315 (0.14)	18 (0.11)		13
Education ^b	HS nongrad	90 (0.02)	29 (0.01)	1 (0.01)		16
	HS grad	358 (0.09)	179 (0.08)	9 (0.05)		28
	Some college	1,521 (0.37)	844 (0.37)	60 (0.34)		33
	College grad	2,138 (0.52)	1,200 (0.53)	104 (0.6)		23
Income ^b	<\$14,999	232 (0.06)	114 (0.05)	7 (0.04)	<\$15,000	16
	\$15,000–21,999	207 (0.05)	105 (0.05)	7 (0.04)	\$15,000–24,999	17
	\$22,000–43,999	550 (0.14)	314 (0.15)	21 (0.13)	\$25,000–49,999	20
	\$44,000–60,000	516 (0.13)	282 (0.13)	24 (0.15)	\$50,000–79,999	11
	>\$60,000	2,121 (0.55)	1,217 (0.57)	97 (0.6)	\$75,000	21
	I don’t know	235 (0.06)	94 (0.04)	6 (0.04)		NA
Visited ER ^c	Yes	707 (0.11)	222 (0.07)	13 (0.07)		8
	No	5,484 (0.89)	2,087 (0.93)	162 (0.03)		92
Hospitalized ^c	Yes	398 (0.06)	104 (0.03)	6 (0.05)		2
	No	5,789 (0.94)	2,205 (0.97)	169 (0.95)		98
Age of diagnosis ^c	0–18	4,847 (0.8)	1,769 (0.78)	117 (0.68)		NA
	<18	1,208 (0.2)	492 (0.22)	56 (0.32)		NA
Asthma Control	Yes	3,772 (0.64)	1,613 (0.7)	142 (0.82)		NA
	No	1,896 (0.32)	630 (0.27)	30 (0.17)		NA
medication ^d	Not sure	192 (0.03)	49 (0.02)	1 (0.01)		NA
Daily inhaled medicine ^d	ICS/LABA ^f	2,233 (0.65)	948 (0.65)	87 (0.68)		NA
	ICS	1,202 (0.35)	506 (0.35)	41 (0.32)		NA
GINA ^e	Uncontrolled	2,534 (0.43)	1,004 (0.44)	73 (0.42)		50 ^g
	Partly controlled	2,246 (0.38)	947 (0.41)	82 (0.47)		NA
	Well controlled	1,067 (0.18)	344 (0.15)	18 (0.1)		NA

Comparison of demographic distributions for Baseline, Robust and 6-month Milestone users with CDC national asthma statistics (https://www.cdc.gov/asthma/most_recent_data.htm).
^aBased on data from 1,430 Baseline, 738 Robust, and 175 Milestone users. ^bBased on data from 4,274 Baseline, 2,317 Robust, and 175 Milestone users. ^cBased on data from 6,240 Baseline, 2,311 Robust, and 175 Milestone users. ^dBased on data from 5,898 Baseline, 2,300 Robust, and 175 Milestone users. ^eBased on data from 5,897 Baseline, 2,295 Robust and 175 Milestone users. All percentages rounded to the nearest 100th. ^fICS/LABA inhaled corticosteroid and long acting beta agonist combination therapy. ^gInstead of using the GINA criteria, the CDC used a slightly different criteria to define uncontrolled asthma patients as those who reported any of the following: (1) asthma symptoms more than two days a week in the past 30 days, (2) nighttime awakenings for more than one time a week in the past 30 days, or (3) short-acting β_2 -agonists use more than two days a week in the past three months (https://www.cdc.gov/asthma/asthma_stats/uncontrolled_asthma.htm). NA, not available.

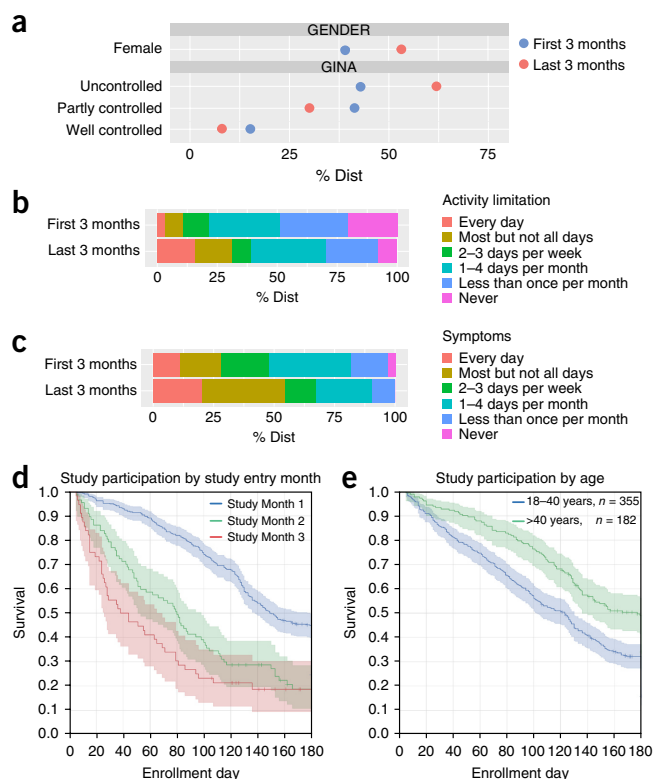


Figure 2 Enrollment and retention over time for Robust users. (a–c) The distributions of gender and GINA control (a), the frequency of activity limitation (b), and the frequency of daily asthma symptoms (c) are all significantly different between the first and the second halves of the study. Specifically, in the latter half of the study, the Robust users have a higher percentage of females (chi-squared = 4.74, d.f. = 1, $P = 0.03$, and $n = 719$), a higher percentage of users with uncontrolled asthma (chi-squared = 8.97, $P = 0.01$, and $n = 2,295$), and an increased frequency of symptoms (chi-squared = 22.3, d.f. = 5, $P = 0.001$, $n = 2,308$) and activity limitation (chi-squared = 36.9, d.f. = 5, $P = 0.0004$, $n = 2,308$). (d,e) Daily survey participation survival curves stratified by study entry month and reported age (Robust users, $n = 537$ participants, >90 d of post-enrollment follow-up). (d) Kaplan-Meier survival curve of daily survey participation stratified by study entry month and excluding participants entering after May ($n = 15$ participants). Study entry month of the participant was statistically significantly associated with daily survey participation longevity using a Cox proportional hazards model, $P = 2.99 \times 10^{-23}$, hazard ratio 1.847 (95% CI, 1.64–2.08) for each passing month. (e) Kaplan-Meier survival curve of daily survey participation stratified by age (18–40 years and >40 years of age). Age was statistically significantly associated with daily survey participation longevity using a Cox proportional hazards model, $P = 1.59 \times 10^{-7}$, hazard ratio 0.976 (95% CI, 0.806–0.96) for each additional year of age. Colored bands show 95% confidence intervals for each strata.

43%, which was significantly lower than that in the second half of the study (62%) (chi-squared test: $P = 0.01$) (Fig. 2a). The distributions of different activity limitation levels and symptom frequencies indicated the same trend (Fig. 2b,c). The percentage of female users was also significantly higher in the second (53%) versus the first half (38%) of the study (chi-squared test: $P = 0.03$; Fig. 2a). The gender distribution of the cohort recruited later more closely approximated the CDC asthma population statistics. We also confirmed the significant associations between enrollment time and gender, GINA scores, activity limitation, as well as symptom frequencies from baseline surveys through logistic and regular regression analyses (Supplementary Table 3b).

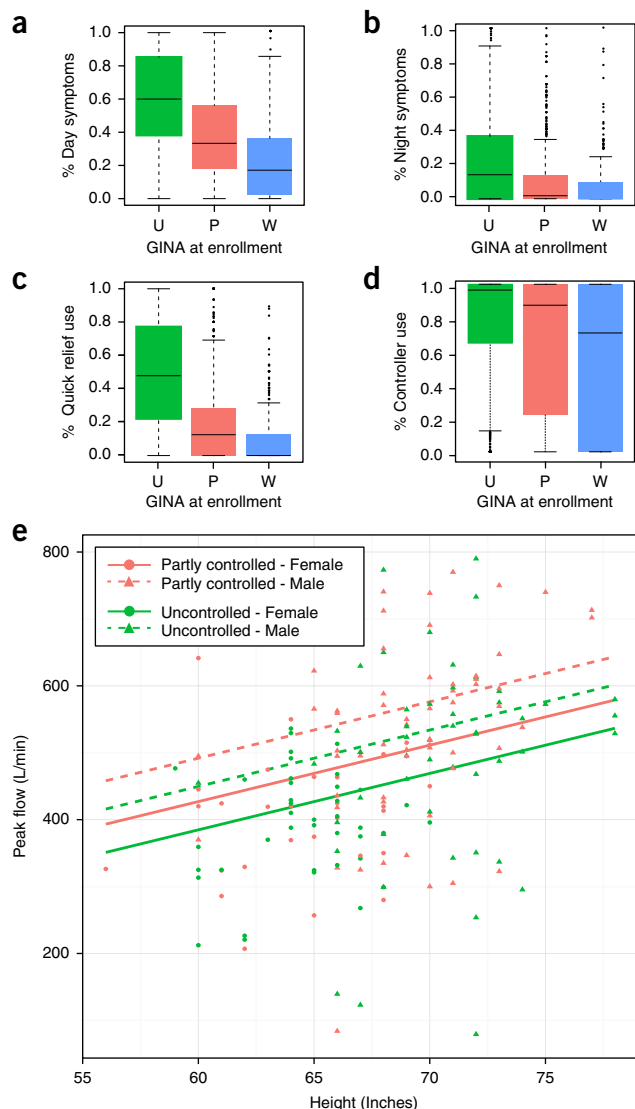


Figure 3 Concordance between GINA control at enrollment and prospectively collected daily symptoms reports during the study. (a–d) Distributions of frequencies of daytime symptoms (Kruskal–Wallis test; $H(2) = 471.94$, $P < 2.2 \times 10^{-16}$, $n = 2,295$) (a), nighttime symptoms (Kruskal–Wallis test; $H(2) = 232.23$, $P < 2.2 \times 10^{-16}$, $n = 2,295$) (b), inhaler puffs usage (Kruskal–Wallis test; $H(2) = 677.12$, $P < 2.2 \times 10^{-16}$, $n = 2,295$) (c), and controller medicine usage (Kruskal–Wallis test; $H(2) = 63.73$, $P = 1.4 \times 10^{-14}$, $n = 2,285$), and (d) among Robust users stratified according to their GINA control level at enrollment. (U, uncontrolled; P, partly controlled; W, well controlled). (e) Based on data from 183 Robust users, the lines illustrate a multiple linear regression model for peak flow trained on users' daily peak flow responses, GINA control assessed at enrollment, and HealthKit physique data, which demonstrates that male sex ($\beta = 64.847$, $t(179) = 2.836$, $P = 0.005$), controlled asthma ($\beta = 42.224$, $t(179) = 2.364$, $P = 0.02$), and height ($\beta = 8.435$, $t(179) = 2.695$, $P = 0.002$) are associated with greater peak flows.

Our study participants completed 79,297 daily and 10,969 weekly surveys over the 6-month study period. Of the 6,470 Baseline users, 6,023 and 2,520 participants responded to at least one daily and weekly survey, respectively. Total survey numbers collected in each month decreased over time (Supplementary Table 3a), consistent with an exponential decay function as observed in the other mobile health research studies⁷. To evaluate the impact of various factors on user

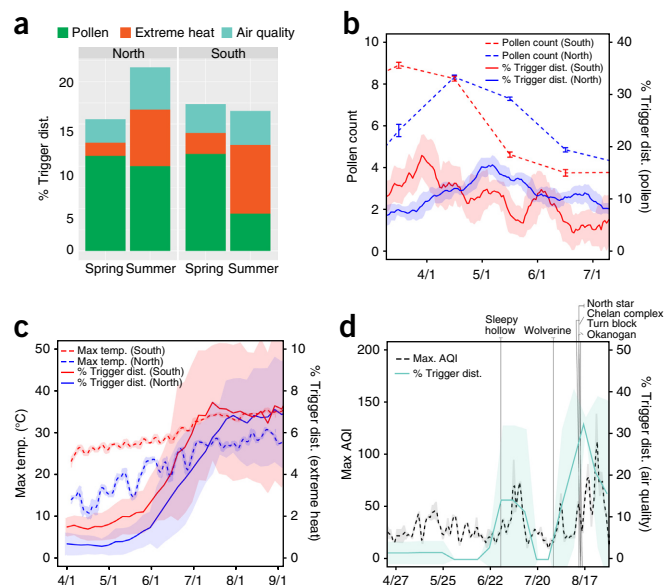


Figure 4 Geographic and seasonal trends in asthma triggers for Robust users. (a) The percentages of users reporting pollen, extreme heat or air quality as an asthma trigger (y axis) for southern (red) and northern (blue) regions of the contiguous US in the spring (March–May) and summer (Jun–Aug) respectively (based on $n = 545$ Robust users). (b) The percentage of users reporting pollen as an asthma trigger (solid) and the monthly pollen level (dashed) for southern (red) and northern (blue) regions of the US (based on $n = 64$ Robust users). (c) The percentage of users reporting extreme heat as their asthma triggers in southern and northern US regions for the spring and summer months (based on $n = 545$ Robust users). (d) The percentage of users reporting air quality as an asthma trigger for Washington state wildfires (solid, left y axis) and daily PM_{2.5} concentration (dashed, right-axis) in the same area (based on $n = 37$ Robust users). In (b–d), the shaded regions represent the ± 1 s.d. interval bands.

retention patterns, we focused on a subset of 537 users from the Robust user cohort, who were enrolled in the study for more than 90 d and who provided data for all the co-variables considered in the analysis. Both univariate and multivariate survival analyses of these 537 users, found earlier entrance into the study, (hazard ratio = 2.01 (95% CI, 1.73–2.33)) for each month following AHA launch, and increasing age, (hazard ratio = 0.978 (95% CI, 0.969–0.987)) for each additional year, significantly associated with greater likelihood of daily survey participation (Fig. 2d,e, Supplementary Fig. 2a and Supplementary Table 4a,b).

We also investigated the ‘individual response rate’, defined as the number of days with at least one daily survey question completed divided by the number of days enrolled through September 9, 2015, for each user. For the 537 users considered in the above retention analysis, the average individual response rate was 31%, with 104 of these users having an individual response rate >50% (Supplementary Fig. 2b). Increasing age and earlier study entry month were also associated with higher individual response rate (Supplementary Table 4c).

Relationship between baseline asthma control and prospectively collected data

Participants completed intake questionnaires assessing asthma control upon study enrollment and then prospectively reported daily and nightly asthma symptoms, quick-relief inhaler usage, controller medicine usage, and peak flow measures over the course of their participation in the study. Patients’ daily survey responses for the

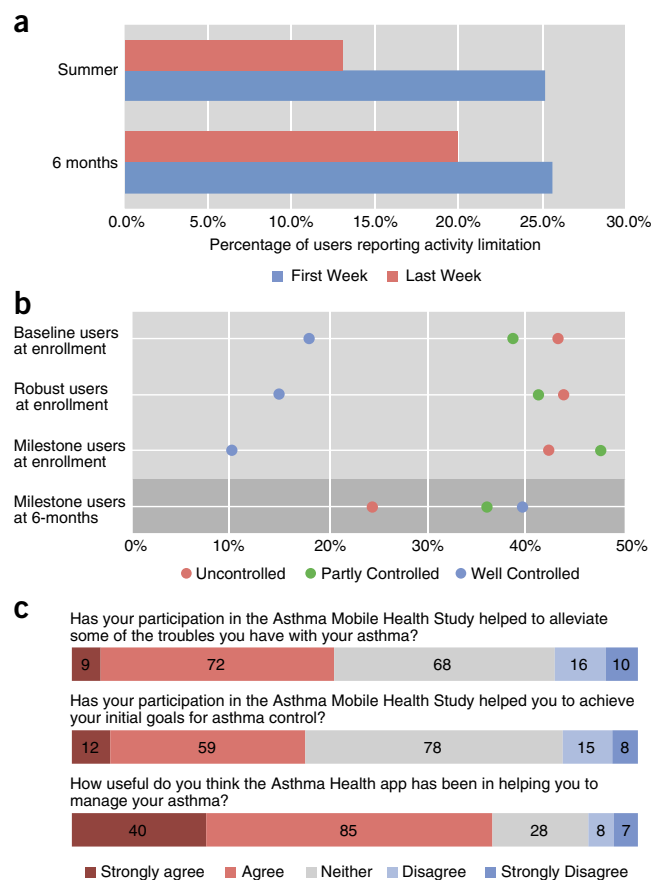


Figure 5 Positive impact of the app on user group. (a) The percentage of users reporting activity-limitation in their first week versus their last week in the summer (top, based on $n = 331$ Robust users) and in the entire 6-month study period (bottom, based on $n = 1,926$ Robust users). (b) The percent distribution of GINA control for all cohorts at enrollment (top three) and after 6-months of study participation (bottom). (c) Feedback and Milestone survey results based on data from Milestone users.

aforementioned four parameters were all found to be significantly associated with the GINA control levels calculated based on intake questionnaires from Robust users reporting daily (Kruskal–Wallis test; $H(2) = 471.94$, $P < 2.2 \times 10^{-16}$, $n = 2,295$) and nightly symptoms (Kruskal–Wallis test; $H(2) = 232.23$, $P < 2.2 \times 10^{-16}$, $n = 2,295$), quick-relief inhaler usage (Kruskal–Wallis test; $H(2) = 677.12$, $P < 2.2 \times 10^{-16}$, $n = 2,295$), and controller medicine usage (Kruskal–Wallis test; $H(2) = 63.73$, $P = 1.4 \times 10^{-14}$, $n = 2,285$) (Fig. 3a–d and Supplementary Fig. 3).

Of those in the Robust user cohort, 1,621 voluntarily submitted at least one peak flow measurement during the study period. As expected, patients with well-controlled asthma, and who were male and tall, had higher average peak flows throughout the study period (Fig. 3e). Those with uncontrolled asthma at baseline reported peak flows 42 liters/min lower than their well-controlled counterpart after adjusting for height and gender ($n = 183$).

We examined concordance of reported asthma symptoms, rescue inhaler use, and peak flow measurements using time series of daily survey responses from Robust users. Consistent with clinical expectation, we detected a positive correlation between daily or nightly symptoms and rescue inhaler use ($n = 979$ and $n = 761$, respectively), whereas these same variables were negatively correlated with peak flow values ($n = 235$ and $n = 173$, respectively) (Supplementary Fig. 4).

Geographic and temporal trends in asthma triggers

Animals, pollen, and upper respiratory tract infections were the top three asthma triggers reported by participants at time of enrollment (Supplementary Fig. 5). Figure 4a illustrates the distribution of three self-reported asthma triggers from the 545 users of the Robust user cohort for whom we had consent to obtain geolocation data for asthma triggers during the spring and summer months (Supplementary Fig. 1b,c). For example, before 1 April 2015, a greater percentage of users reported pollen as a trigger in Southern regions of the United States (where pollen counts were higher) than in Northern regions (Fig. 4b). Additionally, the percentage of users who reported extreme heat as a symptom trigger correlated with maximum daily temperature trends (i.e., climate data reports of the highest temperature recorded over a specified period of time)⁸. As expected, a higher percentage of Southern participants reported extreme heat as a trigger (Fig. 4c).

Our data also indicate that some asthma patients were sensitive to air quality changes caused by environmental events, such as wildfires. There was a marked increase of participants reporting air quality triggers in regions affected by the summer 2015 Washington state wildfires during the corresponding time periods (Fig. 4d).

Self-reported clinical impact of AHA use

We assessed reported activity levels in the subset of our study participants comprising the 1,926 users from the Robust user sub-cohort who had been enrolled for more than 90 d and for whom we had relevant data from the first and last weeks of participation. These participants reported a significant decrease of activity limitation from 25–20% (Wilcoxon signed-rank test, $P < 0.0001$; Fig. 5a and Supplementary Table 5a). The same trend persisted during the three summer months, from 25–13% (Wilcoxon signed-rank test, $P < 0.0001$, $n = 331$, Fig. 5a and Supplementary Table 5b), suggesting that milder weather did not drive changes in activity levels.

Likewise, 173 Milestone users for whom the 6-month milestone survey results and baseline GINA information were available, reported that their asthma control substantially improved over the study period. Specifically, the percentage of Uncontrolled users changed from 42–24% (paired Wilcoxon signed-rank test, $P < 0.0001$; Fig. 5b and Supplementary Table 5c). A modest majority of participants also reported that the AHA was useful in helping them manage their asthma (Fig. 5c and Supplementary Table 5d,e).

DISCUSSION

The AMHS is one of the few studies to examine the value and validity of the novel mobile health research platform, ResearchKit³. We conducted a prospective, observational study focused on the feasibility of conducting research remotely via this platform and observed certain strengths as well as limitations to this methodology. In terms of strengths, the AMHS demonstrated that a broad-scale asthma study can be conducted in its entirety via a smartphone application, including remote recruitment, consent, enrollment, and secure bi-directional data flow between investigators and participants. We prospectively collected detailed, multi-dimensional, longitudinal data on an asthma cohort more efficiently than traditional epidemiological studies by automating, standardizing, and accelerating various costly and time-consuming processes. Our study's rapid recruitment and participants' willingness to share de-identified data broadly highlight users' acceptance of this methodology for low-risk health studies.

As participant recruitment is a significant challenge in research today^{9,10}, digital health and social media could play a role in addressing that challenge. Within 1 d of the launch, the five

ResearchKit studies collectively enrolled over 15,000 participants across the country, beyond university catchment areas, demonstrating the power and potential of this technology¹¹. Characteristics of our study cohort compared with CDC Asthma Surveillance Data identified similarities and differences. Owners of iPhones have higher education levels and income than other smartphone users, who as a group have higher income and education levels than the general population¹² (Supplementary Fig. 6). Of note, only 5% of AHA users with asthma were Black, compared with 13% of the US population, an under-representation commonly encountered in clinical research in general. In the United States, 92% of Hispanics, 91% of Whites, and 94% of Blacks report using a mobile phone, with 64% of Hispanics, 66% of Whites and 64% of Blacks using a smartphone¹². Use of the Android platform is more common in some racial and/or ethnic groups, thus the availability of an Android version of AHA (facilitated by the open source ResearchKit framework) could capture a more representative sampling of the general population. Other nuances, such as a propensity to text rather than use apps in low-income Hispanic communities should be considered in attaining diverse participation in mobile health research^{13,14}.

An understanding of use patterns and the ability to reach and impact diverse populations regardless of platform used will become increasingly important as mobile health technology expands in health-care and research. Given the overwhelming trend toward use of digital communication, responses to landline telephone and mailed surveys may soon represent a more restricted and non-representative population¹². For example, the vast majority of New York City residents (96%) own a cell phone, 79% of which are smartphones, including smartphone ownership among 67% of low income New Yorkers¹². Therefore, the ability to recruit participants via digital technology without direct voice contact or print mailing will likely be needed to conduct population-based research. The possibility exists that the gold-standard population metrics obtained via traditional research methodology used for comparisons in our study may already contain biases. Offering and leveraging the strength of digital health and traditional research methods could optimize clinical study enrollment, participant retention, and data capture more than either method alone, getting us closer to a representative sampling of the general population. Moreover, offering a process that automates certain time-consuming and labor-intensive components, while incorporating a 'human touch' at selective key points may improve study retention and decrease costs^{15–17}.

Higher rates of HCU in AHA users than in CDC data may indicate a potential for selection bias in mobile health recruitment via uptake of this type of study by sicker patients. Conversely, use of a mobile platform may increase participant diversity in research by enabling the enrollment of traditionally under-represented and difficult-to-recruit populations (e.g., those with disabling, severe disease or limited healthcare access). The ability to complete study requirements at home on flexible time schedules may remove barriers (i.e., location, mobility, psychosocial factors, work hours) to research participation. The impact of selection bias and other potential threats to validity of mobile health research are currently poorly characterized and deserve ongoing investigation.

An important goal of our study was to evaluate the quality and validity of study data obtained via this mobile research platform. Patient-reported outcomes (PROs) are an important component of research in asthma and are often collected via paper diaries, where important issues in authenticity of data have been raised, including 'back filling', 'forward filling' and falsified data¹⁸. Because the accuracy of PRO data may be partially dependent on the relationship between

the reporter and the recipient¹⁹, the validity of PRO data obtained via mobile health platforms (without direct participant–investigator contact) warrants rigorous evaluation.

More widespread use of electronic versus paper diaries outside of industry research studies is currently prohibited by cost but may be fostered by an open-source platform such as ResearchKit. In clinical trials, the PROs are correlated with objective measures of lung function, but in epidemiologic asthma research, it is commonplace to use symptom-based surveys without corresponding lung function measurements²⁰. In this respect, our surveys and the types of epidemiologic data gathered do not differ from common practice, apart from the use of technology to scale and accelerate the process. Whereas concerns about falsifying information, such as identity authentication, have been raised in regard to mobile health studies. The consent and registration process of the AMHA is fairly vigorous and may mitigate such risk (e.g., e-mail verification of identity and entering a passcode to access the app). Validity of our data is supported by concordance between our cohort's self-reported asthma status at baseline and prospectively collected data. For example, participants' daily survey responses for day and night symptoms as well as inhaler and controller medication usage were all found to be significantly associated with the GINA control levels calculated based on intake questionnaires for these parameters. Similarly, the peak flow measurements submitted by participants were of expected range based on known trends for patients' sex, height, and asthma control status. Furthermore, we detected that patients' asthma symptoms correlated well with the frequency of rescue inhaler usage and peak flow values as would be expected based on the clinical behavior of asthma. Likewise, the self-reported asthma triggers (e.g., pollen, extreme temperature, air quality, pollutant exposures), mapped based on geography and time, correlated well with objective measures (e.g., external, validated environmental sources).

Of particular note, the summer 2015 Washington state wildfire analysis highlights that smartphone-based technology could provide innovative, scalable solutions for clinical research aspirations that were logistically not feasible or cost-prohibitive in the past. Specifically, we correlated and detected a marked increase in our study participants' daily asthma symptoms (air quality triggers) with real-time fine particulate matter (Environmental Protection Agency (EPA) air quality logs PM_{2.5}) levels in regions affected by wildfires during those corresponding time periods. Conventional assessments of the effect of natural phenomena on disease are usually very limited due to the aforementioned difficulties. Since our AHA data set already contain location-specific environmental data, such exploratory analyses require minimal additional effort or cost to accomplish. In summary, the consistent trending of variables that we expect to be interrelated based on our knowledge of the disease, the tracking of symptoms with known environmental triggers, and the expected correlation among symptoms, medication use, and lung function in a subset of patients for whom lung function was measured strongly suggest the validity of this new research-data collecting method.

Characterizing survey participation rates and engagement with the AHA over time was another major aim of this study. We observed a large initial number of downloads likely driven by media publicity, which decreased to a steady rate over time. We attribute the moderate enrollment rate (14–38% of downloads) to the relative ease of app download with some dropoff related to the rigor of the consent process. Additionally, the significant rate of attrition of participants observed from the initial cohort to some of the sub-analyses conducted raises issues of generalizability. Also of note, although we did not actively recruit participants (thus, no incremental budget was

required), we continued to enroll new participants daily a year after the launch of the study. (Please see **Supplementary Note** for study marketing details.)

The drop-off in user retention over time that we observed appears to be shared by multiple 'digital' use cases (e.g., mobile apps including entertainment 'gaming' apps, tutorial videos, open online courses) and speaks to the hardwired biopsychosocial tendencies of users. Because the ultimate goals of digital health generally rely on prolonged participation, the creators of these tools must understand users' psychosocial-behavioral needs and predilections to keep them continually engaged. Attention and resources must be devoted to incorporating social and behavioral principles in digital health design beyond technical ones. For the AMHS, our study participants were not offered financial incentives—a standard practice for clinical studies when participation for prolonged periods of time is required^{21,22}. In fact, we did not offer users any other 'tangible' rewards beyond features like environmental data, educational modules, and the ability to track entered data. Monetary incentivization, possibly in the form of micro-payments and/or 'advanced gamification'^{23,24} may improve retention, especially if a study requires long term follow-up.

Despite attrition of longitudinal participation over time, 85.2% of the 7,593 enrolled users responded to at least one survey in the study. Although the comparisons are not analogous, we referred to CDC's Behavioral Risk Factor Surveillance System (BRFSS-CDC) statistics for context and as a reference point in interpreting our findings. The BRFSS cooperation rates (defined as the number of complete and partial complete interviews divided by the number of contacted and eligible respondents) were 62.5% for landline-based surveys and 71.6% for cell-phone-based surveys²⁵. These findings suggest that studies relying on mobile apps could achieve similar or better cooperation rates than traditional (landline or mobile) phone-based population research methods.

We encountered some additional challenges in performing our study. We were unable to incorporate certain standard validated asthma surveys into our study due to licensing constraints. Besides the EuroQuol-5D, our AHA surveys were developed by asthma specialists who incorporated general content used by validated survey instruments. Because of an initial technical issue with the integration of HealthKit and ResearchKit data, we obtained only gender and/or age information for 1,398 participants in the study, limiting several analyses. Multiple versions of the AHA were released during the study period to address these software-related concerns and to implement new features (**Supplementary Table 6**). Similar to the other mobile health studies and large-scale research studies in general, we encountered substantial missing values in our AHA data, such that many parallel analyses were presented based on different sample sizes (**Fig. 1** and **Supplementary Table 1a**).

Furthermore, the analyses on the self-reported clinical impact of AHA use were based on subsets of our study participants and may not be representative of our cohort at large. We did note consistent positive feedback from our users (e.g., decreased activity limitation, helpful in disease management), and these sentiments were echoed by some AHA users on our participant panel (<http://apps.icaahn.mssm.edu/asthma/participant-stories/>). However, validating the clinical impact of AHA usage warrants rigorous future clinical trials.

Although mobile health apps and devices may promote health literacy and medication adherence, and mitigate exacerbations of chronic diseases, research on this technology thus far has been mostly small scale and yielded conflicting results. Studies to date have not found advantages in terms of HCU or cost, and adoption of the technology in healthcare settings remains low^{26–29}.

However, the field of devices and mobile health research has witnessed some promising recent advances. For example, D'Arcy *et al.*³⁰ developed a Bluetooth inhaler device that used acoustic recordings of inhaler usage to monitor temporal and technique adherence and assess the correlation between clinical outcomes and adherence. Furthermore, studies using the US Food and Drug Administration (FDA)-approved Propellor Health (Madison, WI, USA) digital platform and inhalers demonstrated the feasibility of tracking inhaler usage and triggers^{31–34}. Moreover, individuals randomized to the Propellor Health arm had more significant reductions in inhaler usage than those under routine care³⁵. Additionally, an observational mobile health study of Parkinson disease that leverages the ResearchKit platform, named mPower, interrogated aspects of the movement disorder and assessed high-resolution activity data collected through surveys and frequent sensor-based recordings from participants with and without Parkinson disease³⁶. The large-scale and repeated measurements of thousands of individuals may help establish baseline variability of real-world activity measurements collected via smartphones and lead to quantification of the ebbs and flows of Parkinson symptoms³⁶. The mPower research team developed a framework that accounts for research participant choices regarding clinical data sharing and also qualifies researchers requesting to access the de-identified data for secondary analyses³⁷. Lastly, the investigators from the five ResearchKit launch partners published a report on their collective experience and the platform's potential and limitations, including selection bias, identity uncertainty, design limitations, retention, and privacy³⁸.

Based on the studies above and the initial results from the AMHS, we believe research hypotheses with the following characteristics are a good match for the current ResearchKit methodology: 1) a requirement for rapid enrollment across diverse geographical locations; 2) a design that presents minimal risk to participants, allowing the use of electronic consent; 3) a hypothesis that can be answered in a short time period (1–3 weeks); 4) a requirement for frequent data collection events; 5) data collection that is passive (e.g., GPS, physical activity); 6) no assumption that results will be generalizable to participants recruited via traditional methods; and 7) a sample size and statistical analysis plan that account for the known attrition and/or missing data historically seen in internet and mobile app studies.

Mobile health research represents a promising new avenue for clinical research. It has the potential to open up new possibilities for data collection, provide novel insights into disease, and reach participants that traditional studies may fail to adequately represent. Mobile health research must be amenable to the demands of a fast-paced, variable research environment, as well as be methodologically rigorous. The challenges associated with this technology, including selection bias, potential reporting bias, data security, and low user-retention rate will need to be addressed in order to better understand the technology's true value and role in research and patient care. In the future, mobile platforms may serve as a primary driver for conducting large-scale studies, perhaps complemented by traditional means to leverage the strengths of both methods. Looking forward, the potential of ubiquitous smartphone technology to address the needs of clinical research to better understand health and disease appears to be more promising than ever.

METHODS

Methods, including statements of data availability and any associated accession codes and references, are available in the [online version of the paper](#).

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS

The study is funded by the Icahn School of Medicine at Mount Sinai, UL1TR001433-01, and with technology support from LifeMap Solutions.

AUTHOR CONTRIBUTIONS

Y.-F.Y.C. developed the initial study design/protocol, including electronic informed consent and the statistical plan, IRB submission and approval, app design and implementation, and budget management, ensured proper study execution, provided clinical support, refined surveys, and assisted in data interpretation, manuscript writing and revision and preparation for submission. P.W. contributed to study design and survey refinement, led statistical support and provided oversight for all data analysis and interpretation, generated figures and tables, and was a major contributor to manuscript writing and revision. L.R. assisted in the initial study design/protocol and IRB preparation and submission, led in the design of surveys, provided clinical support, participated in manuscript writing, and served as NJH liaison. N.T. provided statistical support, including data analysis and interpretation, generated figures and tables, participated in study design and survey refinement, provided major contributions to manuscript writing and revision, and served as a graphic artist liaison. M.Z. assisted in electronic informed consent design, led subsequent IRB submission and provided support, refined surveys, and was a major contributor to manuscript writing and preparation for submission. S.G.H. served as LifeMap Solutions scientific lead and provided support for AHA design, implementation, and functionality, served as a liaison to other technology partners, refined surveys, and assisted in data interpretation and manuscript writing. N.G. led the latter part of the study execution, provided subsequent IRB support, provided clinical support, refined surveys, and assisted in data interpretation and manuscript writing. E.R.S. provided statistical support and data analysis and interpretation, assisted in generating figures and tables, provided subsequent IRB support, and participated in manuscript writing. S.V. was involved in study execution and manuscript writing. M.B. assisted in data analysis and interpretation, and generated figures and tables. E.K. contributed to app design and implementation, as well as initial IRB document preparation. R.E. was the LifeMap Solutions technical lead, participated in app design and implementation, and helped ensure data integrity. R.W. assisted in study design and data interpretation. C.A.P. contributed to study design and data interpretation. J.T.D. contributed to study design input and manuscript revision. E.E.S. participated in study design, oversaw study execution, interpreted data, and participated in manuscript writing and revision.

COMPETING FINANCIAL INTERESTS

The authors declare competing financial interests: details are available in the [online version of the paper](#).

Reprints and permissions information is available online at <http://www.nature.com/reprints/index.html>.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Topol, E.J., Steinhubl, S.R. & Torkamani, A. Digital medical tools and sensors. *J. Am. Med. Assoc.* **313**, 353–354 (2015).
2. Ritter, S. Apple's research kit development framework for iPhone apps enables innovative approaches to medical research data collection. *J. Clin. Trials* **5**, 2–3 (2015).
3. Jardine, J., Fisher, J. & Carrick, B. Apple's ResearchKit: smart data collection for the smartphone era? *J. R. Soc. Med.* **108**, 294–296 (2015).
4. Morgan, A., Mooney, S., Aronow, B. & Brenner, S. Precision Medicine: Data and Discovery for Improved Health and Therapy". Pacific Symposium on Biocomputing. Accessed May 7, 2016. <http://psb.stanford.edu/psb-online/proceedings/psb16/intro-pm.pdf> (2016).
5. Savage, N. Mobile data: Made to measure. *Nature* **527**, S12–S13 (2015).
6. Akinbami, O.J., Moorman, J.E. & Liu, X. Asthma prevalence, health care use, and mortality: United States, 2005–2009. (US Department of Health and Human Services, Centers for Disease Control and Prevention, National Center for Health Statistics Washington, DC, 2011).
7. Dorsey, E.R. *et al.* The use of smartphones for health research. *Acad. Med.* **92**, 157–160 (2017).
8. Menne, M.J. *et al.* Global historical climatology network-daily (GHCN-Daily), Version 3. (NOAA National Climatic Data Center, 2012).
9. Gul, R.B. & Ali, P.A. Clinical trials: the challenge of recruitment and retention of participants. *J. Clin. Nurs.* **19**, 227–233 (2010).
10. Lovato, L.C., Hill, K., Hertert, S., Hunninghake, D.B. & Probstfield, J.L. Recruitment for controlled clinical trials: literature summary and annotated bibliography. *Control. Clin. Trials* **18**, 328–352 (1997).
11. Lovejoy, B. ResearchKit did in 24 hours what would normally take 50 medical centers a year—Stanford University. *9to5 Mac*. Accessed Aug 15, 2016. <https://9to5mac.com/2015/03/12/researchkit-usage/> (2015).

12. Smith, A. US Smartphone use in 2015. Accessed September 5 2016: <http://www.pewinternet.org/2015/04/01/us-smartphone-use-in-2015/> (Pew Research Center, 2015).
13. Arora, S. *et al.* Describing the evolution of mobile technology usage for Latino patients and comparing findings to national mHealth estimates. *J. Am. Med. Inform. Assoc.* **23**, 979–983 (2016).
14. Wei, J., Hollin, I. & Kachnowski, S. A review of the use of mobile phone text messaging in clinical and healthy behaviour interventions. *J. Telemed. Telecare* **17**, 41–48 (2011).
15. Dolan, B. How mobiles make clinical trials bigger, faster and more efficient. *Mobile Health News*. Accessed June 11, 2016. <http://www.mobihealthnews.com/28198/how-mobiles-make-clinical-trials-bigger-faster-and-more-efficient> (2013).
16. Miseta, E. Mobile devices in clinical trials: tested, effective, proven!. *Clinical Leader*. Accessed on May 5, 2016. <https://www.clinicalleader.com/doc/mobile-devices-in-clinical-trials-tested-effective-proven-0002> (2016).
17. Henderson, L. Cell phone key to REMOTE data collection. *Applied Clinical Trials*. Accessed on May 3, 2016. <http://www.appliedclinicaltrials.com/cell-phone-key-remote-data-collection> (2011).
18. Coons, S.J. *et al.* Capturing patient-reported outcome (PRO) data electronically: the past, present, and promise of ePRO measurement in clinical trials. *Patient* **8**, 301–309 (2015).
19. Podsakoff, P.M. & Organ, D.W. Self-reports in organizational research: problems and perspectives. *J. Manage.* **12**, 531–544 (1986).
20. Murphy, K.R. *et al.* Asthma management and control in the United States: results of the 2009 asthma insight and management survey. *Allergy Asthma Proc.* **33**, 54–64 (2012).
21. Halpern, S.D., Karlawish, J.H., Casarett, D., Berlin, J.A. & Asch, D.A. Empirical assessment of whether moderate payments are undue or unjust inducements for participation in clinical trials. *Arch. Intern. Med.* **164**, 801–803 (2004).
22. Russell, M.L., Moralejo, D.G. & Burgess, E.D. Paying research subjects: participants' perspectives. *J. Med. Ethics* **26**, 126–130 (2000).
23. King, D., Greaves, F., Exeter, C. & Darzi, A. 'Gamification': influencing health behaviours with games. *J. R. Soc. Med.* **106**, 76–78 (2013).
24. Miller, A.S., Cafazzo, J.A. & Seto, E. A game plan: Gamification design principles in mHealth applications for chronic disease management. *Health Informatics J.* **22**, 184–193 (2016).
25. Centers for Disease Control and Prevention. Behavioral risk factor surveillance system: summary data quality report. Accessed Aug. 14, 2016. https://www.cdc.gov/brfss/annual_data/2013/pdf/2013_dqr.pdf (CDC, 2013).
26. Alkhalidi, G. *et al.* The effectiveness of technology-based strategies to promote engagement with digital interventions: a systematic review protocol. *JMIR Res. Protoc.* **4**, e47 (2015).
27. Wantland, D.J., Portillo, C.J., Holzemer, W.L., Slaughter, R. & McGhee, E.M. The effectiveness of Web-based vs. non-Web-based interventions: a meta-analysis of behavioral change outcomes. *J. Med. Internet Res.* **6**, e40 (2004).
28. Ryan, D. *et al.* Clinical and cost effectiveness of mobile phone supported self monitoring of asthma: multicentre randomised controlled trial. *Br. Med. J.* **344**, e1756 (2012).
29. Bloss, C.S. *et al.* A prospective randomized trial examining health care utilization in individuals using multiple smartphone-enabled biosensors. *PeerJ* **4**, e1554 (2016).
30. D'Arcy, S. *et al.* A method to assess adherence in inhaler use through analysis of acoustic recordings of inhaler events. *PLoS ONE* **9**, e98701 (2014).
31. Su, J.G. *et al.* Feasibility of deploying inhaler sensors to identify the impacts of environmental triggers and built environment factors on asthma short-acting bronchodilator use. *Environ. Health Perspect.* **125**, 254–261 (2017).
32. Hai, A. *et al.* Adherence monitoring and E-health: how clinicians and researchers can use technology to promote inhaler adherence for asthma. *J. Allergy Clin. Immunol. Pract.* **1**, 446–454 (2013).
33. van Boven, J.F., Trappenburg, J.C., van der Molen, T. & Chavannes, N.H. Towards tailored and targeted adherence assessment to optimise asthma management. *NPJ Prim. Care Respir. Med.* **25**, 15046 (2015).
34. Howard, S., Lang, A., Patel, M., Sharples, S. & Shaw, D. Electronic monitoring of adherence to inhaled medication in asthma. *Curr. Respir. Med. Rev.* **10**, 50–63 (2014).
35. Merchant, R.K., Inamdar, R. & Quade, R.C. Effectiveness of population health management using the propeller health asthma platform: a randomized clinical trial. *J. Allergy Clin. Immunol. Pract.* **4**, 455–463 (2016).
36. Bot, B.M. *et al.* The mPower study, Parkinson disease mobile data collected using ResearchKit. *Sci. Data* **3**, 160011 (2016).
37. Wilbanks, J. & Friend, S.H. First, design for data sharing. *Nat. Biotechnol.* **34**, 377–379 (2016).
38. Trister, A.D., Dorsey, E.R. & Friend, S. Smartphones as New Tools in the Management and Understanding of Parkinson's Disease. 2016NJP-Parkinson's Disease. Accessed Aug 1 2016. <http://www.nature.com/articles/npjparkd20166>.

ONLINE METHODS

Setting for Apple ResearchKit and AHA. The Icahn School of Medicine at Mount Sinai collaborated with Sage Bionetworks, LifeMap Solutions (New York), and Apple to develop and launch the Asthma Health App and Asthma Mobile Health Study on March 9, 2015. This study was approved by the Institutional Review Board of the Icahn School of Medicine at Mount Sinai.

Code availability. The asthma health mobile app version 1.011 was built using Apple's ResearchKit framework (<http://researchkit.org/>), which is open source and available on GitHub (<https://github.com/researchkit/researchkit>). AppCore (<https://github.com/ResearchKit/AppCore>) is a layer built on top of ResearchKit that was shared among the five initial ResearchKit apps. The Bridge iOS SDK (<https://github.com/Sage-Bionetworks/Bridge-iOS-SDK>) provides integration with Sage Bionetworks' Bridge Server, a back-end data service designed for collection of participant donated study data (<https://sagebionetworks.jira.com/wiki/display/BRIDGE/Bridge+REST+API>)³⁶. Code used in this study is also available as **Supplementary Software**.

Participant recruitment and enrollment. The AHA is available in the iTunes App Store searchable using a combination of keywords including 'asthma', 'health', and/or 'Mount Sinai'. Links to the AHA store page were embedded within relevant web sites of Mount Sinai, LifeMap Solutions, and Apple Inc. Prospective participants can download the app using their own App Store credentials. After downloading and opening the AHA, prospective participants are first presented with an inclusion/exclusion criteria questionnaire. Participants who meet the eligibility criteria proceed to electronic informed consent screens. To ensure participants understand the risks, benefits, and options of study participation, they must then pass a comprehensive quiz before creating an account. Following the creation of an account, prospective participants are asked to verify their e-mail address by clicking on a link in their e-mail to confirm their enrollment. The AHA user experience and recruitment process is detailed in **Figure 1**. Please see **Supplementary Note** for information on marketing for the study.

Eligibility criteria for participants. Participants were eligible to enroll in the study if they were aged 18 or over, lived in the United States, had a diagnosis of asthma with physician-prescribed asthma medications, and had an iPhone with a data plan. Pregnant women, non-English speakers, and those who could not document understanding of the consent based on a series of key questions were ineligible for the study.

Study design, data flow, and security. **Supplementary Figure 7a** provides a simplified layout of the initialization process in the Asthma Mobile Health Application study. Data entered by participants were collected and deidentified using advanced data security technologies developed by our partners at Sage Bionetworks. Data from the study were not shared with Apple or with non-study personnel. Please see **Supplementary Figure 7b** for detailed description of the backend design on health data encryption and securely stored.

Data sources for active data collection (surveys). Study surveys appeared on the participant's 'Activities' screen. Baseline surveys collected data on: 1) asthma history, including the frequency and time of symptoms and activity limitation; 2) asthma experience, including triggers and personal management plans; 3) medical history; and 4) demographics. Additionally, baseline healthcare utilization, asthma medications, self-reported controller adherence, quick relief/rescue medication use, peak flow, and other clinically relevant data were collected. Participants were also asked to set a goal for asthma control and complete the EQ-5D-5L (EuroQol version 5D-5L). After the intake process, daily and weekly surveys were administered, with EuroQol, 6-month milestone, and app feedback surveys occurring at less frequent intervals.

Data sources for passive data collection. Participants were asked for permission for the app to read various fields of HealthKit data. HealthKit data that already exists in the user's iPhone from other apps can be collected by app and relayed to the study data. Also, with the participant's consent, the nearest EPA air-quality reading along with that station's city and state were passively collected by the app when the participants viewed the dashboard tab. As of version

1.0.6 released May 5, 2016, the app began to send reports hourly whenever the user's location changed.

Quantitative variables for CDC and AHA baseline and clinical demographics. Demographic data was obtained from the 2013 Behavioral Risk Factor Surveillance System (BRFSS) (http://www.cdc.gov/asthma/most_recent_data.htm). Rates of hospitalization and emergency department visits come from the CDC/NCHS National Ambulatory Medical Care Survey, National Hospital Ambulatory Medicare Care Survey, National Hospital Discharge Survey, National Vital Statistics System, and National Health Interview Survey 2001–2009. CDC defined 'Uncontrolled asthma patients' as those who reported any of the following: (1) asthma symptoms more than 2 d a week in the past 30 d, (2) nighttime awakenings more than once a week in the past 30 d, or (3) short-acting β_2 -agonists use more than 2 d a week in the past 3 months (http://www.cdc.gov/asthma/asthma_stats/uncontrolled_asthma.htm). We compared Baseline users who supplied location data with national asthma prevalence statistics percent distribution by state (<http://www.cdc.gov/asthma/brfss/2013/tableC1.htm>). Percent prevalence was $\log_{10}$ transformed and states or territories with fewer than five AHA users were omitted from analysis. The correlation coefficient was calculated based on $n = 49$ states representing 4,612 Baseline users.

Quantitative variables for GINA (Global Initiative for Asthma) symptom control. 1) daytime symptoms occur less than twice per week (<8 times per month); 2) no occurrence of nocturnal awakenings (0 per month); 3) quick relief of symptoms occurs fewer than twice per week (<8 puffs per month); and 4) no activity limitation due to asthma symptoms (0 per month). Asthma is considered 'Uncontrolled' if four of the above statements are true, 'Partly controlled' if 2–3, and 'Well controlled' if 0–1 (<http://ginasthma.org/>).

Statistical methods for survival analysis of daily survey participation. We calculated the number of days between enrollment and the completion of the last daily survey question for each Robust cohort participant using data collected between the launch of the study, March 9, 2015, and September 8, 2015. To avoid bias from using the September 9, 2015, cutoff, we only considered participants with at least 90 d of study enrollment (participants' enrollment before June 9, 2015). This leads to 537 Robust cohort participants. Participants were treated as censored if a daily or weekly survey question was answered within 2 weeks of September 9, 2015.

The Cox proportional hazards model implemented in the Lifelines python package (version 0.8.0.0) was used to identify features associated with longer daily survey participation. We created the following predictor categories to increase sample size: education (no college, some college, college graduate) and health insurance (no health insurance, or private/public health insurance). All reported statistically significant associations exhibited a monotonic relationship with the enrollment day of the last daily survey question completed in the absence of grouping. We excluded 'decline to answer' responses and participants with missing predictor or outcome data. Features were not standard scaled to preserve interpretability of the hazard ratios; however, standard scaling did not change the direction or statistical significance of the reported associations. We used the KaplanMeierFitter function in the Lifelines python package to plot the survival curves for three strata of self-reported education levels and two strata of self-reported age of asthma diagnosis. The Seaborn python package (version 0.6.0) was used to generate heat maps for the predictor correlation matrix, which was calculated using the `cor` function of the Pandas package (version 0.17.1).

We first performed univariate survival analyses for nine covariates based on 537 Robust users. We then carried out a multivariate survival analysis of daily survey participation to adjust for collinearity (**Supplementary Fig. 2a**). The direction, magnitude, and statistical significance of the relationship between age and study entry date with study participation were Robust to different variable collapsing strategies and stratification by study entry month.

Statistical methods for individual response rate regression. We focused on the same set of 537 participants considered in the survival analysis of the retention time. We calculated the 'individual response rate', defined as the number of days with at least one daily survey question completed divided by the number

of days enrolled through September 9, 2015. We then used ordinary least-squares regression (StatsModels version 0.6.1) to identify associations between the logit-transformed individual response rate.

Statistical methods for daily survey data. Daily survey responses collected on the same calendar day were combined in the following way. Any 'True' answer to questions about use of: control medication, night symptoms, day symptoms, or quick relief inhaler was used. The maximum reported value for questions about peak flow and quick relief puffs was used. The union of the reported asthma triggers reported on the same calendar day was used. A heatmap was generated using the matplotlib python library to show the self-reported data for day symptoms, night symptoms, and quick relief inhaler usage in daily surveys (Supplementary Fig. 3).

Statistical methods for concordance between GINA control based on baseline surveys and prospectively collected daily symptom reports during the study. The association between baseline GINA control group and prospectively collected daily symptom reports (% day symptoms, $n = 2,295$; % night symptoms, $n = 2,295$; quick relief usage frequencies, $n = 2,295$; and controller usage frequencies, $n = 2,285$) were evaluated with Kruskal–Wallis groupwise rank sum test. In addition, for each user, peak flow measures more than 900 and less than 60 were removed and were averaged across the study period. Then, a multiple linear regression model was used to evaluate the association between mean peak flow of each user during the study period and their baseline categorical GINA control groups, sex, height, and age ($n = 183$).

Statistical methods for concordance of survey responses within time-series. For each Robust user, we evaluated pair-wise Pearson correlations of user daily survey responses for peak flow, day and night symptoms (yes/no), and quick-relief usage (yes/no). We first filtered out users with fewer than ten pairs of observations. For each user, peak flow measures more than 900 and less than 60 were removed. Users whose survey responses had a s.d. of zero across the time-series for either pair of observations, or where the observations for each pair were exactly matching were also removed. The resulting distribution of Pearson correlations evaluated for each user for each pairwise comparison is shown in Supplementary Figure 4.

Statistical methods for delineation of United States into northern and southern regions based on temperature. Several app features were introduced after launch, including collection of encrypted user location data, which started on April 21, 2015. To compare data on asthma triggers and compare it to objective environmental measures, we first categorized 'user-date locations' as northern or southern based on the clustering of local temperature profiles, where user-date location refers to the latitude and longitude information of one user on one particular day. We obtained temperature data from the National Oceanic and Atmospheric Administration 4 (NOAA4), which collects daily maximum temperature from more than 53,000 US weather stations in the Global Historical Climatology Network (GHCND).

To facilitate comparison with other environmental data sets, clustering was performed at the zip code level. First, each user-date location was linked to the nearest zip code and GHCND weather station by minimum great circle distance using the 'fields' package in R. Latitude and longitude coordinates for US zip codes and weather stations were obtained from the NOAA4 and the US Census Bureau (<https://www.census.gov/geo/maps-data/>), respectively. User-date locations with nearest weather station or zip code more than 50 miles away were not included in subsequent analyses. In total, the 8,083 user locations collected during our study period map to 3,646 unique zip codes and 576 US weather stations.

Delineation of the US into northern and southern regions was based on hierarchical clustering of temperature profiles derived from the maximum daily temperature data corresponding to our 6-month study period. For each of 3,646 zip codes within 50 miles of any of the user locations collected during our study period, 37 features were constructed from the daily maximum temperature data collected at the corresponding US weather stations. Specifically, since not all of the user-date locations that link to the same zip code are linked to the same station, we first calculated the daily maximum temperature at each zip code by taking the mean daily maximum temperature over the set

of stations linked to each zip code via user-date location as described above. We then derived our 37 features by taking the median of the daily maximum temperature at each zip code across 5-d intervals of our 185-d study period.

Hierarchical clustering was performed using the 'hclust' package in R. Specifically, we first calculated the pairwise Euclidean distances between each feature–zip code and then performed hierarchical clustering on the resulting distance matrix using the complete linkage method. For further analysis, user zip codes were assigned to the north or south based on the first bifurcation of the resulting dendrogram (Supplementary Fig. 1b). Supplementary Figure 1c shows the geographical regions associated with each cluster.

Statistical methods for analysis of asthma triggers. Users were asked to provide triggers of their asthma symptoms at the start of enrollment, and then on a daily basis throughout the study period. To compare time-series trigger data in different regions, we assigned each user-date location to the north or south based on the clustering procedure described above. For users who remained within the same region throughout the period observed, we extrapolated their regional assignment (north/south) to their full enrollment period. This resulted in 24,720 user-date locations corresponding to 545 unique users from the Robust user cohort, for further analysis. Trigger distributions were calculated based on the total number of triggers reported for a given time period (season, 5 d or 1 d). Periods for which the total number of user-date data points were fewer than 10 were treated as missing. A comprehensive depiction of time-series data for the northern and southern regions at baseline and throughout the study period is shown in Supplementary Figure 5, where the ordering of triggers from top to bottom is based on the percentage rank at baseline. Each curve was generated using the R function `smooth.spline`. Note that data for the last 16 d are omitted in Supplementary Figure 5 because a very limited number of users in the southern region responded to the daily survey in those days.

Trigger distributions were compared with environmental data from several sources. Pollen count data were obtained from <https://www.pollen.com>, which provides monthly average pollen counts for zip codes throughout the United States. We obtained pollen count data for 49 zip codes corresponding to 53 users from the Robust user cohort with 5,601 user-date locations in the north and 16 zip codes corresponding to 11 users from the Robust user cohort with 123 user-date locations in the south. Zip codes were selected to ensure an adequate number of user-date locations ($n \geq 10$) for calculating the trigger distribution in the final half of our study period, where data for southern users are sparse. For the southern and northern regions, we calculated the average and the standard error of the pollen level across the corresponding zip codes for each region. The data are shown in Figure 4b as the dashed lines. In addition, in Figure 4b, the daily percentage of users reporting pollen as a trigger was plotted along time, where the running mean based ($n = 10$) and Bollinger bands (s.d. = 1) are shown.

In Figure 4c, the percentage of Robust users reporting extreme heat as their asthma trigger is compared with maximum daily temperature data obtained from the NOAA4. The percentage of users reporting extreme heat as an asthma trigger was calculated for 37 5-d intervals based on weather data from 492 weather stations corresponding to 432 users from the Robust user cohort with 19,031 user-date locations in the north and 108 weather stations corresponding to 113 users from the Robust user cohort with 8,347 in the south. Figure 4c shows running mean curves and Bollinger bands for maximum temperature and extreme heat trigger distributions based on a smoothing window of 11 and a s.d. of 1.

To investigate the effects of wildfire incidents occurring during our study period on AHA users, we collected data on wildfire locations and their start dates from the InciWeb incident information service (<https://inciweb.nwcg.gov>). We then searched for AHA users within a 200-mile radius of each fire. Using this procedure, we were able to find a total of 37 Robust users in proximity to Washington state wildfire locations, including the following: 35 users (499 user-date locations) near the Sleepy Hollow wildfire (6/28), 27 users (385 user-date locations) near the Wolverine Fires (7/29), 25 users (380 user-date locations) near the North Star wildfire (8/13), 29 users (398 user-date locations) near the Chelan Complex wildfire (8/14), 26 users (381) near the Turn Block wildfire (8/14), and 26 (381) near the Okanogan wildfire (8/15). In Figure 4d, we illustrate the percentage of triggers due to air quality complaints

for our study period along with EPA logs of air quality data (AQI) (http://www3.epa.gov/airdata/ad_data.html), where the maximum air quality index curve is based on a running average ($n = 3$). Bollinger bands based on a s.d. of 1 are shown.

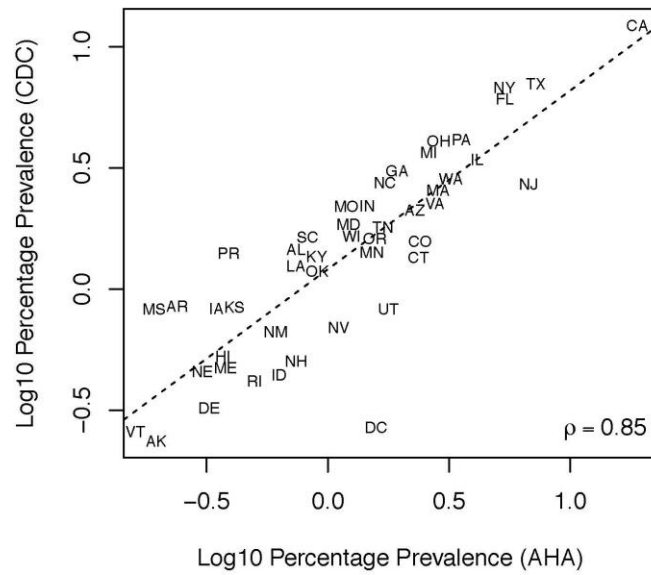
Statistical methods for activity limitation. In the weekly survey, users are asked to report whether they have experienced activity limitation in the past week. We compared the proportion of users reporting activity limitation in the first week and the last week of their enrollment for 1,926 users from the Robust user cohort who were enrolled for longer than 90 d using the Wilcoxon signed rank test. We performed the same analysis restricting to the subset of users who were enrolled exclusively in the summer months ($n = 331$).

Sample size. Because the goals of the various analyses in the study differed significantly, different subsets of users (with specific characteristics) were carefully selected for each analysis. In **Figure 1**, we introduced a few different sample cohorts and sub-cohorts. In **Supplementary Table 1a**, we summarized sample sizes for the analyses corresponding to each of our main figures. We provide further details on the sub-cohort sample sizes used in all analyses below. Note that there were no pre-specified sample sizes in these analyses. All sample size numbers are based on observed data from the study.

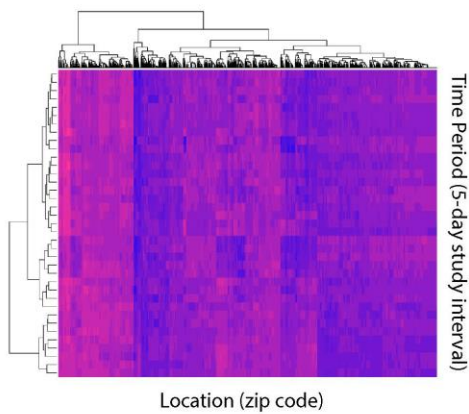
1. In **Table 1** and **Supplementary Table 2**, demographic and clinical characteristics were summarized for Baseline, Robust, and Milestone users based on available data for each variable. See accompanying legends for sample sizes particular to each calculation.
2. Geographic distribution assessment was performed for 4,621 users from the Baseline user cohort who supplied their location information (**Fig. 1**).
3. The geographic distribution of $n = 4,612$ Baseline users was compared with national asthma prevalence statistics from the CDC for 49 US states (**Supplementary Fig. 1a**).
4. Association analysis to identify factors impacting the time of enrollment was performed based on Robust users with adequate data to determine biological sex ($n = 719$), baseline GINA category ($n = 2,295$), frequency of activity limitation ($n = 2,308$), and symptoms ($n = 2,308$) (**Fig. 2a–c**).
5. Detection of factors affecting user retention patterns and response rates was carried out based on 537 users from the Robust user cohort, who were enrolled in the study for >90 d and provided data for all the covariates considered in the analysis (**Supplementary Fig. 2d,e** and **Supplementary Table 4a,c**).

6. Association between patients' daily survey responses and their baseline GINA categories were evaluated based on subsets of the Robust user cohort reporting daily ($n = 2,295$) and nightly symptoms ($n = 2,295$), quick-relief inhaler usage ($n = 2,295$), and controller medicine usage ($n = 2,285$) (**Fig. 3a,d**).
7. Regression analysis of peak flow information was based on data from 183 users from the Robust user cohort who voluntarily submitted at least one peak flow measurements during the study period, and all covariates (e.g., age of onset, gender, height) in the analysis (**Fig. 3e**).
8. Concordance analysis of reported asthma symptoms, rescue inhaler use, and peak flow measurements within time-series was based on subsets of Robust user cohort, who provided their corresponding information. Specifically, correlations between daily/nightly symptoms and rescue inhaler use were based on $n = 979$ and $n = 761$ users, respectively. Correlations between daily and nightly symptoms and between peak flow and puff usage were based on $n = 817$ and $n = 217$ users, respectively. And correlations between daily/nightly symptoms and peak flow values were based on $n = 235$ and $n = 173$ users, respectively. (**Supplementary Fig. 4**).
9. Geographic distribution of asthma triggers were evaluated based on 545 users of the Robust user cohort, who, throughout their respective enrollment periods, could be consistently mapped to one of two geographical regions (north/south) based on time series location data (**Fig. 4a** and **Supplementary Fig. 1b,c**).
10. Assessment of changes of activity limitation during the study period was based on 1,926 users who are from the Robust user cohort and had been enrolled for >90 d (**Fig. 5a**). In addition, 331 of the 1,926 users, who supplied enough data during the summer, were used to assess the impact of app usage on users' activity limitation during summer months (**Fig. 5a** and **Supplementary Table 5a,b**).
11. 173 of the 175 users in the Milestone user cohort supplied complete data to derive their GINA categories at the enrollment and after 6-month's App usage. Data of these 173 were used to evaluate the impact of 6-month's App usage on users' GINA categories (**Fig. 5b** and **Supplementary Table 5d**).
12. Feedback and milestone survey results were evaluated for Milestone users who replied to questions about whether the app helped them to alleviate their troubles with asthma ($n = 175$), achieve their initial goals for asthma control ($n = 172$), and manage their asthma ($n = 168$), (**Fig. 5c**).
13. The relationship between GINA category and milestone survey feedback indicating whether the app helped to prevent visits to the emergency department or doctor was evaluated for 125 and 127 Milestone users, respectively (**Supplementary Table 5d,e**).

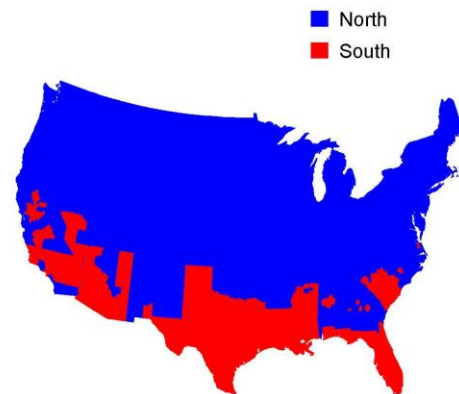
(a)



(b)



(c)

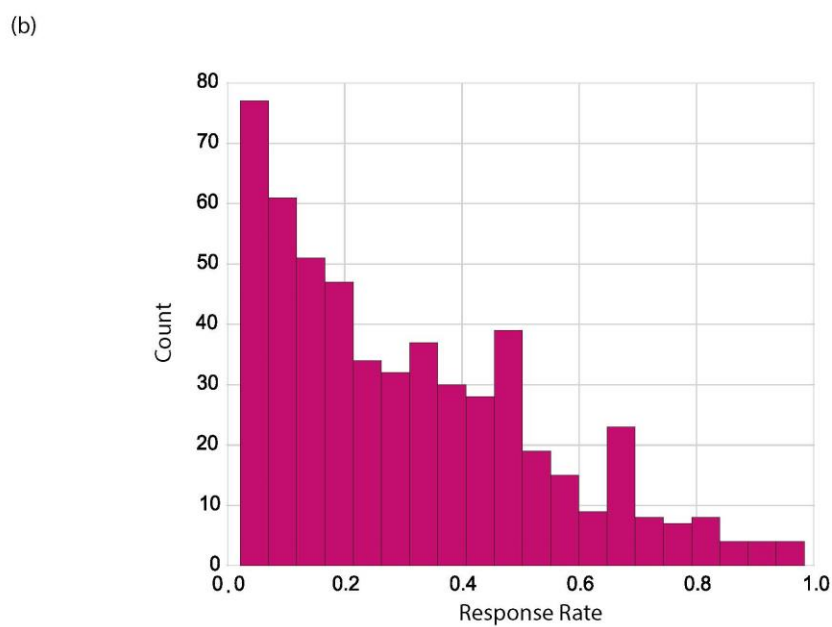
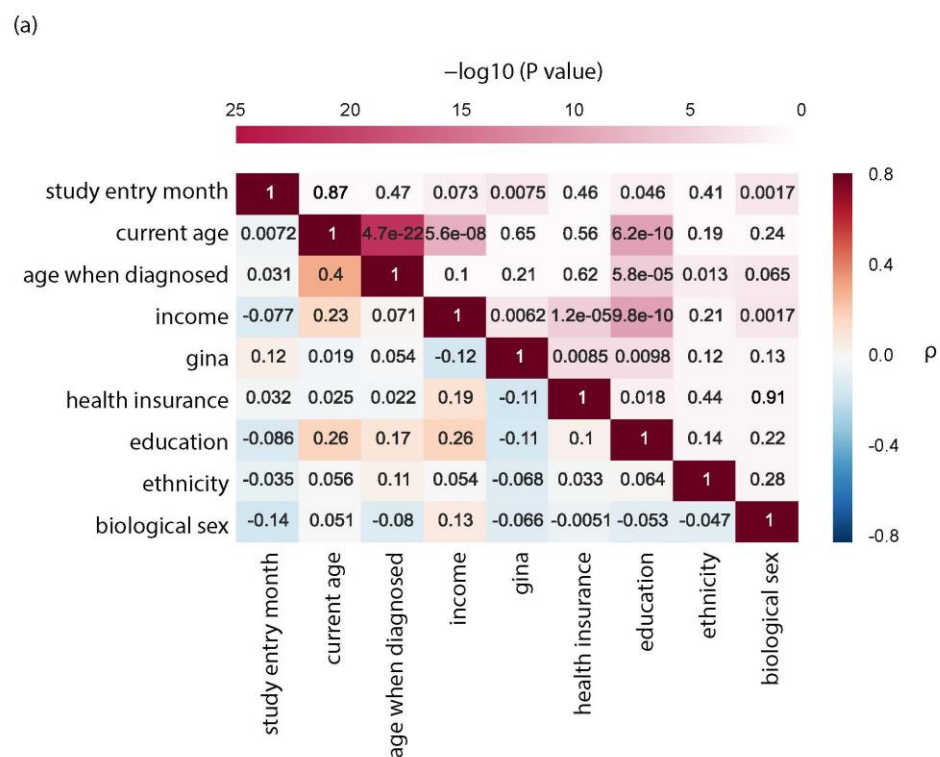


Supplementary Figure 1

Geographic distribution of asthma prevalence.

a) Based on location data from 4,612 *Baseline* users from 49 states, the log10 transformed percent distribution of asthma prevalence

for asthma health app users is compared to 2013 national asthma prevalence statistics. SOURCE: 2013 Behavioral Risk Factor Surveillance System (BRFSS) <http://www.cdc.gov/asthma/brfss/2013/tableC1.htm>, Downloaded on 1/19/16. (b) The heatmap illustrates 37 features derived from the median of the maximum daily temperature over 5-day intervals for 3,646 user-linked zip codes (see Online Methods for details of the clustering method). (c) Northern (blue) and southern (red) regions of Contiguous US as determined by clustering of maximum median temperature (see methods). Figure shows the overlay of the county map on top of the state map, where color coding (North, blue and South, red) is based, in each case, on majority rule of user location clustering assignment.

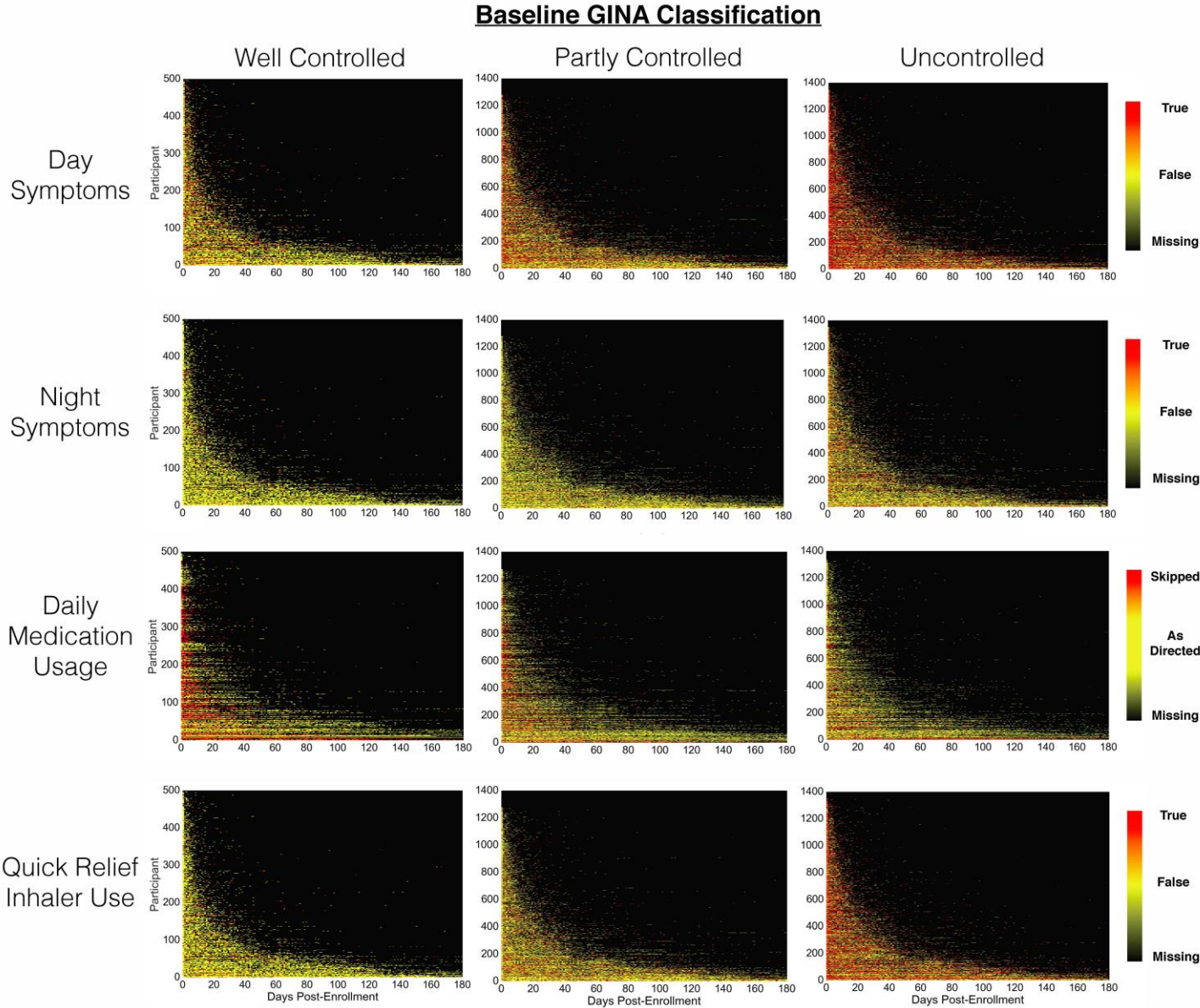


Supplementary Figure 2

Retention analysis: correlations among covariates and the distribution of response rates.

Both (a) and (b) are based on $n=537$ Robust users who enrolled between March 9 2015 and June 8 2015. (a) Heatmap of the

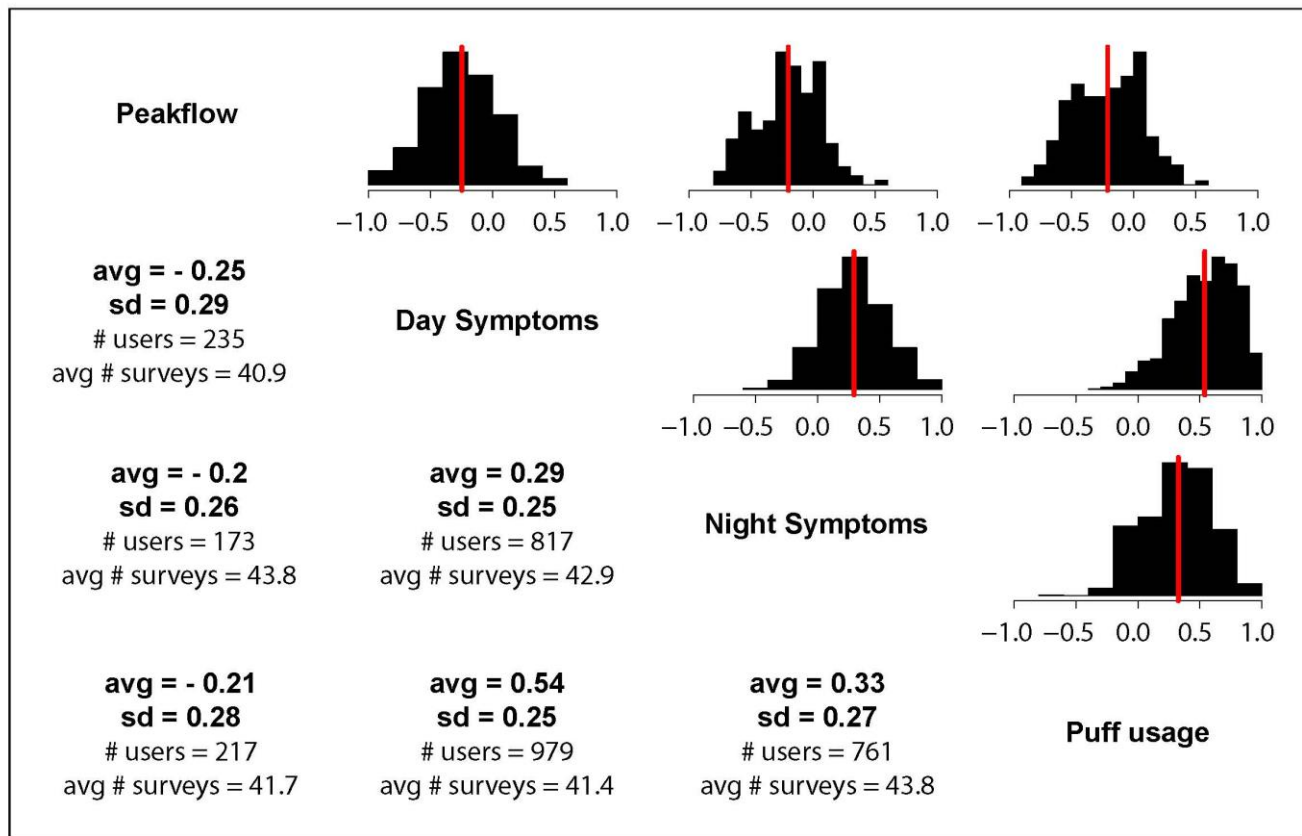
Pearson's correlation matrix of the predictors for retention time in the Cox Proportional Hazards model. Lower diagonal cells in (a) represent the Pearson's correlation coefficients; the upper diagonal cells indicate the p-values for testing zero correlation. (b) Histogram of individual response rates of daily surveys.



Supplementary Figure 3

Heat maps of daily survey response by baseline GINA control category.

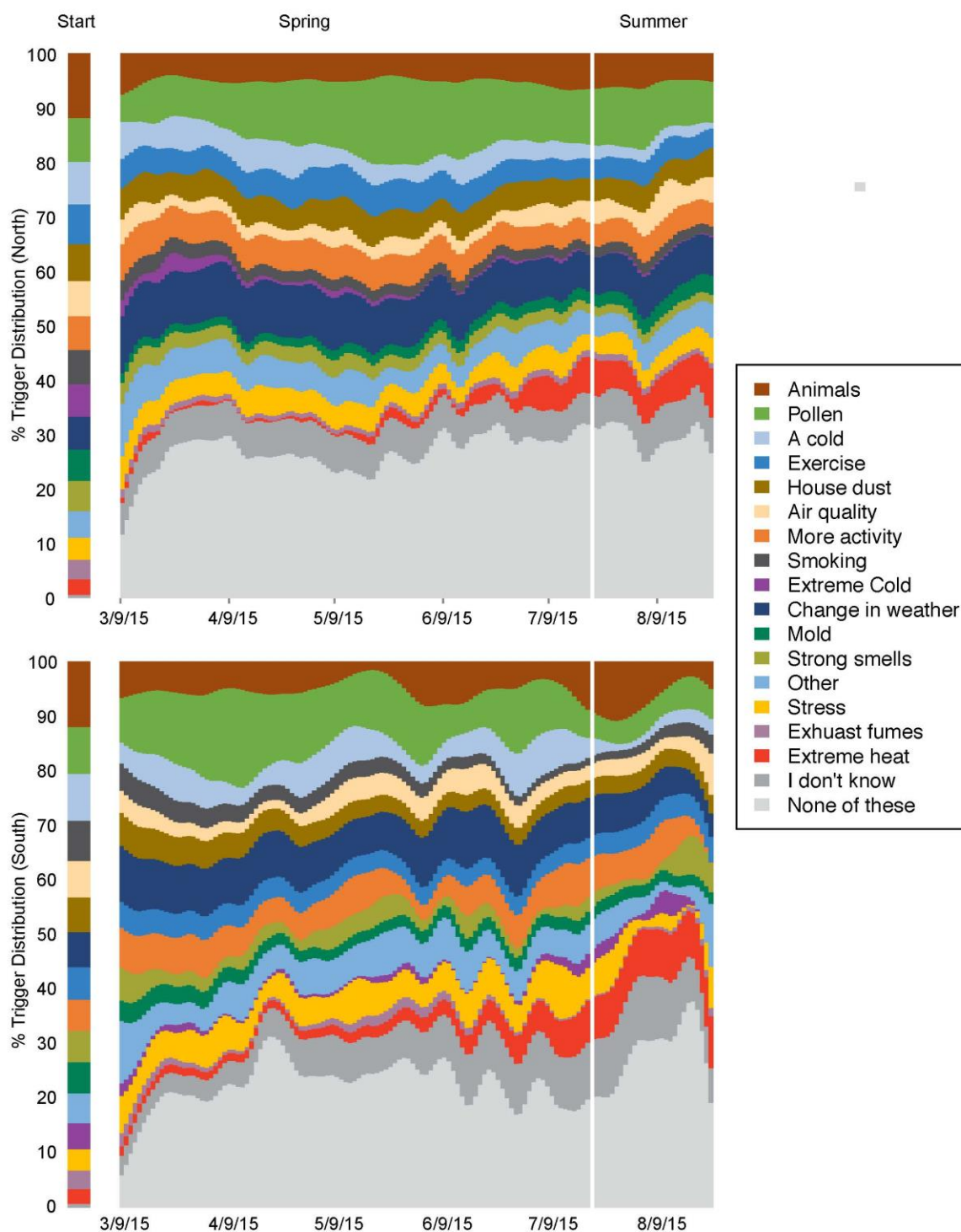
Heatmaps of self-reported day symptoms, night symptoms, and quick relief inhaler usage were plotted for participants (y-axis) stratified by baseline GINA classification across 180 days of study enrollment (x-axis). Only participants enrolled for at least 180 days by 9/9/2015 were plotted. Participants were sorted according to the number of non-missing responses for each plot. Responses to these questions were: True (red), False (yellow), or missing (black).



Supplementary Figure 4

Concordance within time-series.

Pair-wise comparisons of correlation based on self-reported peakflow, day symptoms, night symptoms, and number of quick relief puffs. Each plot shows a histogram of correlation coefficients evaluated for each user based on his or her daily survey response time series. Total number of Robust users, average number of surveys per Robust user, and summary statistics for each comparison are shown in the lower diagonal.



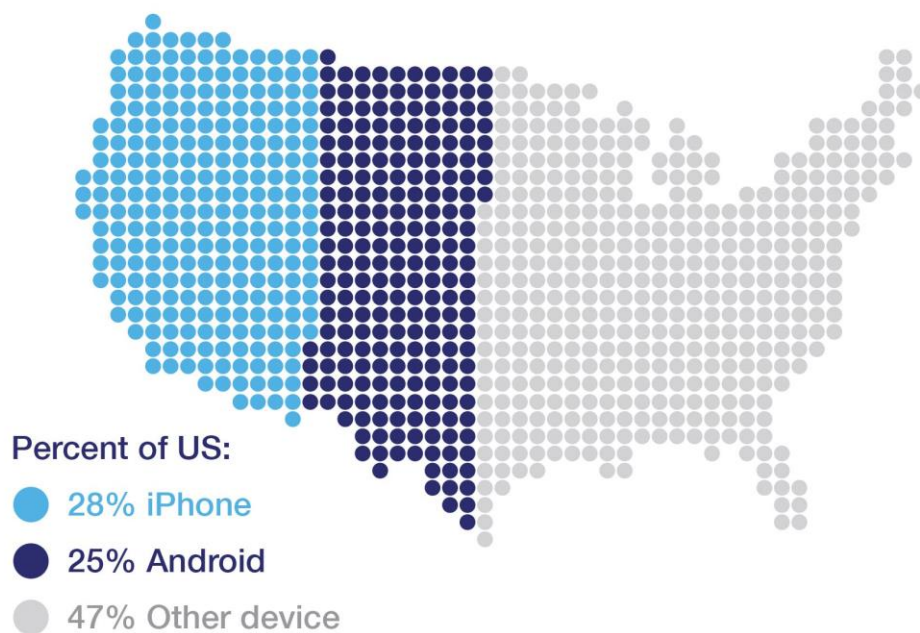
Supplementary Figure 5

Distributions of self-reported asthma triggers at the start of enrollment and throughout the study period.

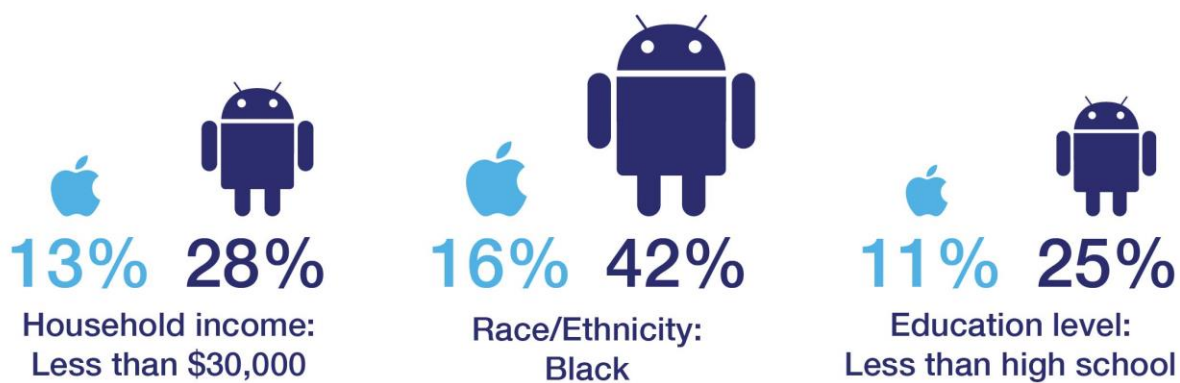
Based on data from 545 Robust users, barplots (left) show trigger distributions ordered by percentage rank of each asthma trigger for northern (top) and southern regions (bottom). Trigger distribution curves (right) are based on daily asthma trigger reports for robust

users ordered by percentage rank at the start of enrollment for northern (top) and southern (bottom) regions respectively (see online Methods).

(a)



(b)

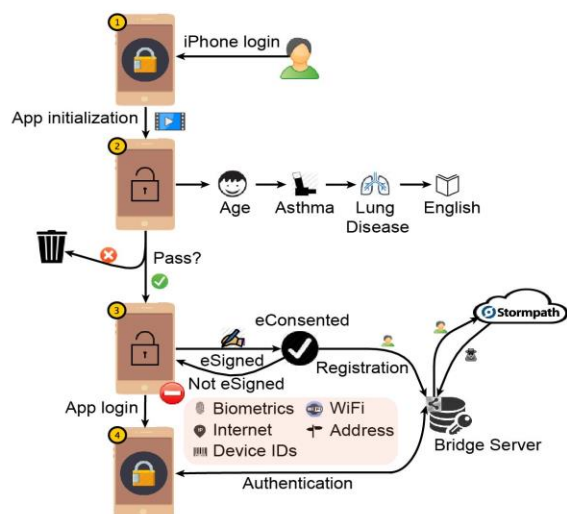


Supplementary Figure 6

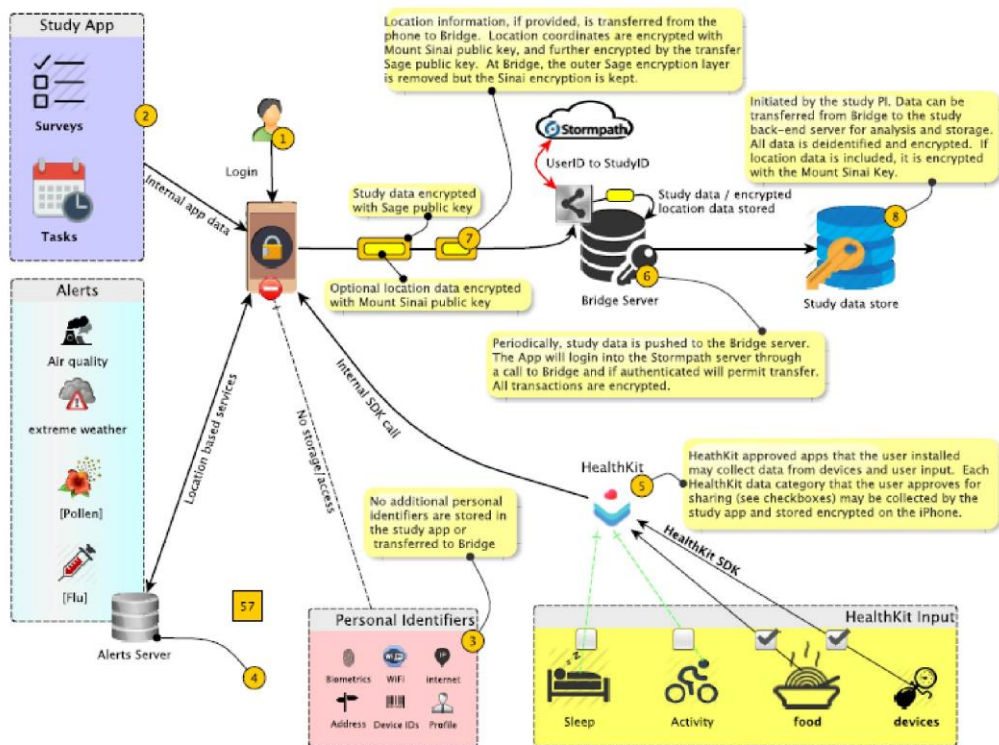
iPhone versus Android users.

(a) Potential Asthma Health app audience by iPhone, Android and other smart devices. (b) Targeted demographics for the Asthma Health app.

(a)



(b)



Supplementary Figure 7

A simplified layout of the initialization, the data flow and security measures in the Asthma Mobile Health Application study.

(a) The initialization process in the Asthma Mobile Health Application study. (b) The data flow and security measures in the Asthma Mobile Health Application study.

Supplementary Table 1. Sub-cohort definitions and summary of AHA surveys. (a) Sub-cohort definitions. Within each cohort, sub-cohorts were defined based on the number of users (N) with requisite data for each of our analyses (specified by the Figure column). The bulleting indicates the subset relationship within each sub-cohort. (b) Summary of AHA surveys. For surveys appearing for multiple times in the App (e.g. Asthma Medications, Asthma Daily/Weekly Survey, and EQ-5D), the “number of users” column reports the number of all users who responded to the corresponding survey for at least once during the study.

(a)

Cohort	Sub-cohorts with requisite data on	N	Figure
Baseline	Location	4,621	1
Robust	GINA at enrollment start	2,295	2a
	• Daily/nightly symptoms, or daily inhaler puffs	2,295	3a-c
	o Controller medicine usage	2,285	3d
	Biological sex	719	2a
	• Covariates used in the analysis of retention	537	2d-e
	o Peakflow and height	183	3e
	Activity limitation or symptom frequency	2,308	2b-c
	• Activity limitation in the first and last week of enrollment	1,926	5a
	o Activity limitation in the first and last week of summer	331	5a
	Trigger analysis covariates	545	4a
Milestone	GINA at enrollment start	173	5b
	Feedback on app use (manage asthma)	168	5c
	Feedback on app use (achieve goals)	172	5c
	• Pollen	64	4b
	• Temperature	545	4c
	• PM2.5 levels near the Washington state wildfires	37	4d

(b)

Survey Name	When & Frequency?	Number questions	Number of users	Content
Asthma History	Appears on first day	13	6240	Variety of questions about asthma history including those required to compute baseline GINA
About You	Appears on third day	8	4274	Demographics (ethnicity, race, income, education), smoking status, health insurance.
Asthma Medications	Appears on first day, Reappears on day 165 (updated survey)	20	5898	All asthma medications (Controller & Rescue) with their doses
Medical History	Appears on fourth day	17	3775	Questions about comorbidities (heart disease, lung disease, allergies, cancer, etc.)
Your Asthma	Appears on second day	9	4494	Questions about triggers, flu shot, asthma action plan, spirometer, asthma troubles and goals
Asthma Daily Survey	Rekurs daily	8	6023	Symptoms, rescue inhaler usage, controller inhaler compliance, triggers, peak flow
Asthma Weekly Survey	Rekurs every 7 days	16	2520	Questions about health care utilization, activity limitations, missed work and side effects
EQ-5D	Appears on first day, Rekurs every 90 days	6	3347	Questions about mobility, self-care, usual activities, pain / discomfort, anxiety /depression and overall score (0-100)
Asthma Milestone Survey	Appears on day 165	11	175	Demographics are recollected (height, weight, gender, age), repeat of questions from Asthma History including questions to compute GINA
Asthma Milestone -- App & Study Feedback	Appears on day 165	13	172	Ratings, feedback, (least) favorite feature, lessons learned, etc.

Supplementary Table 2. Characteristics of Patients at Enrollment. Demographic distributions for Asthma Health App users. Data on demographic variables comes from the 'Asthma History' survey administered at. All percentages rounded to the nearest 100th. Baseline users include all study participants. Robust users are Baseline users who completed ≥ 5 daily or weekly surveys, smoke < 10 packs per year, and have no competing risks, including congestive heart failure or lung diseases other than asthma. Milestone users are Baseline users who completed a 6-month milestone survey.

Characteristic		Enrolled	Robust	Milestone
Health Insurance ^a	Private	3420 (0.81)	1941 (0.84)	144 (0.83)
	Public	480 (0.11)	227 (0.1)	23 (0.13)
	I have no health insurance	209 (0.05)	96 (0.04)	5 (0.03)
	I choose not to answer	118 (0.03)	41 (0.02)	2 (0.01)
Intubated ^b	Yes	809 (0.13)	255 (0.11)	20 (0.11)
	No	4958 (0.8)	1921 (0.83)	147 (0.84)
	Not sure	420 (0.07)	127 (0.06)	8 (0.05)
Work missed in past 6 months ^b	I don't go work or go to school	319 (0.05)	110 (0.05)	26 (0.15)
	I did not miss work or school	4158 (0.67)	1621 (0.7)	102 (0.58)
	1 day	571 (0.09)	189 (0.08)	10 (0.06)
	2-5 days	789 (0.13)	269 (0.12)	24 (0.14)
	6-9 days	163 (0.03)	54 (0.02)	5 (0.03)
	10 days or more	186 (0.03)	62 (0.03)	8 (0.05)
Symptoms in past 6 months ^b	Everyday	749 (0.12)	261 (0.11)	26 (0.15)
	Most but not all days	1031 (0.17)	405 (0.18)	36 (0.21)
	2-3 days per week	1135 (0.18)	459 (0.2)	30 (0.17)
	1-4 days per month	1895 (0.31)	776 (0.34)	59 (0.34)
	Less than once a month	1069 (0.17)	342 (0.15)	21 (0.12)
	Never	315 (0.05)	65 (0.03)	3 (0.02)
Night symptoms (wake you up) in past 6 months ^b	Everyday	122 (0.02)	25 (0.01)	1 (0.01)
	Most but not all days	309 (0.05)	106 (0.05)	13 (0.07)
	2-3 days/week	486 (0.08)	183 (0.08)	13 (0.07)
	1-4 days/month	1361 (0.22)	532 (0.23)	31 (0.18)
	Less than once a month	1816 (0.29)	712 (0.31)	54 (0.31)
	Never	2098 (0.34)	751 (0.33)	63 (0.36)
Limited activity past 6 months ^b	Everyday	330 (0.05)	86 (0.04)	9 (0.05)
	Most but not all days	519 (0.08)	177 (0.08)	15 (0.09)
	2-3 days/week	646 (0.1)	248 (0.11)	14 (0.08)
	1-4 days/month	1719 (0.28)	685 (0.3)	54 (0.31)
	Less than once a month	1677 (0.27)	646 (0.28)	47 (0.27)
	Never	1302 (0.21)	466 (0.2)	36 (0.21)
# of times hospitalized in past 6 months ^b	0 times	12 (0.03)	5 (0.05)	0 (0)
	1-2 times	312 (0.79)	92 (0.88)	6 (1)
	3-5 times	52 (0.13)	6 (0.06)	0 (0)
	>5 times	18 (0.05)	1 (0.01)	0 (0)
# of times in the ER in past 6 months ^b	0 times	8 (0.01)	2 (0.01)	0 (0)
	1-2 times	561 (0.81)	189 (0.86)	11 (0.85)
	3-5 times	105 (0.15)	26 (0.12)	2 (0.15)
	6-10 times	10 (0.01)	3 (0.01)	0 (0)
	>10 times	12 (0.02)	0 (0)	0 (0)
Seen a doctor for asthma in past 6 months (other than routine) ^b	Yes	2198 (0.36)	799 (0.35)	68 (0.39)
	No	3990 (0.64)	1510 (0.65)	107 (0.61)
# of times seen a doctor for asthma past 6 months ^b	0 times	68 (0.03)	23 (0.03)	3 (0.04)
	1-3 times	1834 (0.84)	682 (0.85)	52 (0.78)
	4-6 times	216 (0.1)	74 (0.09)	11 (0.16)
	7-9 times	28 (0.1)	10 (0.01)	0 (0)
	>9 times	37 (0.02)	9 (0.01)	1 (0.01)
Oral steroids for 3 days or more to treat attack in past 6 months ^b	None	3899 (0.63)	1492 (0.65)	115 (0.66)
	One	1104 (0.18)	398 (0.17)	33 (0.19)
	Two	538 (0.09)	188 (0.08)	10 (0.06)
	Three or more	621 (0.1)	215 (0.09)	15 (0.09)

^a Based on data from 4272 Baseline, 2317 Robust, and 96 Milestone users.

^b Based on data from 6240 Baseline, 2311 Robust, and 96 Milestone users.

Supplementary Table 3. AHA Enrollment. (a) Monthly downloads, number enrolled and enrollment rates (in parentheses) for Enrolled, Baseline, Robust, and Milestone users. (b) Summary statistics for tests of association between enrollment date and gender, GINA, activity limitation, and symptom frequency for Robust users.

(a)

Enrollment Period	Downloads	Daily Survey ^a	Weekly Survey ^a	Enrolled ^b	Robust ^b	Milestone ^b
Mar 8- Apr 7	43949	30701 (0.41)	3458 (0.32)	6027 (0.14)	1959 (0.04)	133 (0.00)
Apr 8- May 8	3645	16486 (0.22)	2302 (0.21)	878 (0.24)	222 (0.06)	24 (0.01)
May 9 – Jun 8	1113	11979 (0.16)	1878 (0.17)	322 (0.29)	72 (0.06)	8 (0.01)
Jun 9 – Jul 9	501	8470 (0.11)	1200 (0.11)	123 (0.25)	29 (0.06)	3 (0.01)
Jul 10 – Aug 9	446	4845 (0.06)	1240 (0.11)	126 (0.28)	22 (0.05)	5 (0.01)
Aug 10 – Sep 9	309	3313 (0.04)	894 (0.08)	117 (0.38)	13 (0.04)	2 (0.01)
Total	49963	79297	10969	7593	2317	175

^a The values in parentheses correspond to percent distributions based on dividing by column totals.

^b The values in parentheses correspond to the proportion of each count to the number of app downloads in the same period.

(b)

Characteristic	OR/β	C.I.	p-value	N
GINA* (Uncontrolled = 1, Other = 0)	1.007	1.004–1.010	2E-05	2295
Gender* (Female = 1, Male = 0)	1.007	1.004–1.011	1E-04	719
Activity Limitation [†]	0.006	0.004–0.008	4E-09	2308
Symptoms [†]	0.005	0.003–0.007	6E-07	2308

* Based on a logistic model of regression.

[†] Based on a regular model of regression.

Supplementary Table 4. Prediction of user retention and response rate. Estimated Cox proportional hazards (CoxPH) models for predicting retention time based on univariate (a) and multivariate (b) analysis; (c) Estimated ordinary least squares model for predicting individual response rates of daily surveys. CoxPH models were used to identify features associated with longer daily survey participation. (a) shows the result from univariate analysis, i.e. each predictor was fit independently, but coefficient statistics were aggregated into one table; (b) instead shows the results of the multivariate CoxPH model. (c) illustrates the result from an ordinary least squares regression model for detecting associations between the logit- transformed response rates and 9 covariates. All these results are based on n = 537 Robust users who enrolled between March 9 2015 and June 8 2015.

(a)

	coef	Exp(coef)	se(coef)	z	p	p.adj	lower 0.95
Study entry month	0.637	1.891	0.075	8.523	0.000	0.000	1.633
Current age	-0.020	0.980	0.004	-4.907	0.000	0.000	0.972
Age when diagnosed	-0.009	0.991	0.004	-2.423	0.015	0.139	0.984
Ethnicity	-0.293	0.746	0.144	-2.029	0.042	0.382	0.562
Income	-0.089	0.915	0.045	-1.995	0.046	0.415	0.838
Health insurance	-0.462	0.630	0.272	-1.694	0.090	0.813	0.369
Education	-0.131	0.877	0.083	-1.586	0.113	1.000	0.746
Gina	0.083	1.086	0.072	1.153	0.249	1.000	0.944
Biological sex	-0.017	0.983	0.103	-0.164	0.869	1.000	0.803

(b)

	coef	exp(coef)	se(coef)	z	p	lower 0.95	upper 0.95
study entry month	0.698	2.009	0.076	9.156	0.000	1.730	2.333
current age	-0.023	0.978	0.005	-4.903	0.000	0.969	0.987
health insurance	-0.461	0.631	0.282	-1.636	0.102	0.363	1.096
ethnicity	-0.206	0.814	0.146	-1.408	0.159	0.611	1.084
biological sex	0.100	1.105	0.108	0.925	0.355	0.895	1.365
age when diagnosed	-0.004	0.996	0.004	-0.899	0.369	0.988	1.004
gina	0.061	1.063	0.073	0.831	0.406	0.921	1.227
education	0.047	1.048	0.089	0.530	0.596	0.881	1.248
income	-0.014	0.986	0.049	-0.285	0.776	0.896	1.086

(c)

	Coef.	Std.Err.	t	P> t	[0.025	0.975]
Intercept	-2.124	0.353	-6.026	0.000	-2.816	-1.432
C(health_insurance)[T.1.0]	0.610	0.295	2.068	0.039	0.030	1.190
C(ethnicity)[T.2.0]	-0.096	0.152	-0.631	0.528	-0.394	0.202
C(BiologicalSex)[T.Male]	-0.254	0.104	-2.449	0.015	-0.458	-0.050
l(study_entry_month)	-0.441	0.076	-5.802	0.000	-0.590	-0.292
l(CurrentAge)	0.023	0.004	5.702	0.000	0.015	0.031
l(education_stdscale)	-0.056	0.055	-1.020	0.308	-0.165	0.052

Supplementary Table 5. Self-reported clinical impact of AHA. (a) Detailed number of activity limitation events reported in the first and last week for the entire 6-month study period for Robust users. (b) The same information as in (a) for the summer three months. (c) Number of participants in each GINA control level category evaluated at enrollment and after 6 months of study participation for Milestone users. (d) The percentage of users reporting that the app has helped them to avoid an emergency department visit based on GINA evaluations at enrollment. (e) The percentage of Milestone users reporting that the app has helped them to avoid a doctor's visit based on GINA evaluations at enrollment.

(a)

Have Activity Limitation? (6 month)		First Week (25.4%)	
		Yes	No
Last Week (~20%)	Yes	297	85
	No	194	1350

(b)

Have Activity Limitation? (3 month)		First Week (25.1%)	
		Yes	No
Last Week (~13%)	Yes	25	18
	No	58	230

(c)

GINA control (<i>Milestone users</i>)		6-month milestone		
		(-1) Uncontrolled	(0) Partly controlled	(1) Well controlled
At Enrollment	(-1) Uncontrolled	31	25	17
	(0) Partly controlled	11	35	36
	(1) Well controlled	1	2	15

(d)

	GINA	Yes	No	Don't know
Prevented an ED visit	Well Controlled	0(0)	12(0.80)	3(0.20)
	Party Controlled	11(0.18)	35(0.58)	14(0.23)
	Uncontrolled	13(0.26)	29(0.58)	8(0.16)

(e)

	GINA	Yes	No	Don't know
Prevented a doctor's visit	Well Controlled	3(0.19)	7(0.43)	6(0.38)
	Party Controlled	13(0.21)	31(0.50)	17(0.28)
	Uncontrolled	14(0.28)	26(0.52)	10(0.20)

Supplementary Table 6. Asthma Health app releases and how they relate to data. The Asthma Health App was developed using agile methodologies with all features being tracked with JIRA (Atlassian). Initial release was timed for Apple's Spring Forward event on March 9, 2015. Between launch and September 9, 2015, six upgrades were released with new features and enhancements to data collection (ST1). Some detailed information can be found at <https://sagebionetworks.jira.com/wiki/display/BRIDGE/App+Data+Issues>,

Release	Date	Features
1.0	March 9, 2015	Initial release Air quality data sent to the server when the user viewed their "Dashboard"
1.0.2	March 24, 2015	First release with a EuroQoL survey (appeared for 1 day) Initial HealthKit integration (but data reception did not begin until April 14) Removed default value for height (5'0")
1.0.3	April 21, 2015	First release with weekly survey (appeared for 1 day) Added recontact survey Activity reminders were turned off
1.0.6	May 5, 2015	Activity reminders turned back on as default mode Implemented medication reminder that was activated 2 hours after activity reminders Collected non-Identifiable Demographics (height, weight, gender, age) data Location data sent along with air quality data Survey updates: bounded values to question about quick relief puffs and peak flow; removed question about medicine changing from daily survey so asked weekly instead Some users reported missing or duplicate activities in their activities feed
1.0.7	July 14, 2015	Weekly and EuroQoL surveys persisted for their intended duration Fixed daily surveys stoppage and duplicate activities
1.0.8	July 28, 2015	Collected usage analytics data with Amazon MobileAnalytics
1.0.9	August 24, 2015	Updated daily survey with improved skip logic
1.10	September 14, 2015	Updated medication survey Addition of doctor dashboard Daily survey wrote rescue inhaler counts and peak flows to HealthKit
1.11	September 29, 2015	First release with milestone and feedback surveys

1 **Supplementary Text 1. Survey Questions.**

About You		
Question	Name	Details
Ethnicity	ethnicity	one of {Hispanic/Latino, Non-Hispanic/Latino, I choose not to answer}
Race (check any that apply)	race	multiple choice from of {Black/African American, Asian, American Indian or Alaskan Native, Hawaiian or other Pacific Islander, White, Other, I choose not to answer}
Which of the following best describes the total annual income of all members of your household?	Income	one of {<\$14,999, \$15,000-21,999, \$22,000-43,999, \$44,000-60,000, >\$60,000, I don't know, I choose not to answer}
What is the highest level of education you have completed?	education	one of {8th grade or less, More than 8th grade but did not graduate high school, High school graduate or equivalent, Some college, Graduate of Two Year College or Technical School, Graduate of Four Year College, Post graduate studies, I choose not to answer}
What is your smoking status?	smoking_status	one of {Never (<100 Cigarettes in lifetime), Current, Former}
On average, how many cigarettes per day did you smoke daily?	avg_cigarettes	integer input
How many years in total did you smoke?	smoking_years	integer input
Do you have health insurance?	health_insurance	one of {Private (bought by you or your employer), Public (Medicare or Medicaid), I have no health insurance, I choose not to answer}
Feedback Survey		
Question	Name	Details
How useful do you think the Asthma Health app has been in helping you to manage your asthma?	manage_asthma	one of {Very useful, Somewhat useful, Useful, Neutral, Not very useful, Not at all useful}
Has your knowledge about asthma increased since participating in the Asthma Mobile Health Study?	asthma_knowledge	yes/no
Would you recommend the Asthma Health app to someone with Asthma?	recomend	yes/no
How would you rate the Asthma Health app, overall?	rate	{1 star – I hate it, 2 stars – I don't like it, 3 stars – It's OK, 4 stars – It's good, 5 stars – It's great}
What was your favorite feature(s) of the Asthma Health app?	favorite	multiple choice from of {Surveys, Reminders, Dashboard, Air Quality Near You, Asthma Symptom Control Dashboard and Calendar, Badges, Learn Tab}
What feature(s) did you use the least (or not at all) while participating in the Asthma Health app?	unused	multiple choice from of {Surveys, Reminders, Dashboard, Air Quality Near You, Asthma Symptom Control Dashboard and Calendar, Badges, Learn Tab}
What was your least favorite feature(s) of the Asthma Mobile Health Study?	least_favorite	multiple choice from of {Surveys, Reminders, Dashboard, Air Quality Near You, Asthma Symptom Control Dashboard and Calendar, Badges, Learn Tab}

Were there days that you did not complete the daily or weekly surveys?	incomplete_surveys	yes/no
What were your reasons for not completing the surveys?	why_incomplete	multiple choice from of {For days I did not have any symptoms, I often didn't feel like completing the surveys, For days I was too busy, I often didn't feel like completing the surveys, For days I was not feeling well, I often didn't feel like completing the surveys}
Were there other reasons for skipping surveys you'd like to mention?	other_reasons_incomplete	text input
Are you willing to participate in future mobile health clinical research studies?	again	yes/no
What feedback would you give to the Asthma Mobile Health Study team to improve this app?	other_feedback	text input
What is the most important information or lesson that you have learned through use of the Asthma Health app?	learned	text input
Daily Survey		
Question	Name	Details
In the last 24 hours, did you have any daytime asthma symptoms (cough, wheeze, shortness of breath or chest tightness)?	day_symptoms	yes/no
In the last 24 hours, did you have any nighttime waking from asthma symptoms (cough, wheeze, shortness of breath or chest tightness)?	night_symptoms	yes/no
Did you use your quick relief inhaler in the last 24 hours, except before exercise?	use_qr	yes/no
Except for use before exercise, how many total puffs of your quick relief medicine did you take over the past 24 hours ?	quick_relief_puffs	integer input
Did any of the following cause your asthma to get worse today? (check all that apply):	get_worse	multiple choice from of {A cold, Exercise, Being more active than usual (walking, running, climbing stairs), Strong smells (perfume, chemicals, sprays, paint), Exhaust fumes, House dust, Dogs, Cats, Other furry/feathered animals, Mold, Pollen from trees, grass or weeds, Extreme heat, Extreme cold, Changes in weather, Around the time of my period, Poor air quality, Someone smoking near me, Stress, Feeling sad, angry, excited, tense, Laughter, I don't know what triggers my asthma, None of these things trigger my asthma}
Enter your peak flow today? (L/min)	peakflow	integer input
Did you take your asthma control medicine in the last 24 hours?	medicine	one of {Yes, all of my prescribed doses, Yes, some but not all of my prescribed doses, No, I did not take

		them, I'm not sure}
Asthma History		
Question	Name	Details
How old were you when first diagnosed with asthma, or when you had your first symptoms?	age_when_diagnosed	integer input
Have you ever been intubated (in intensive care on a life support/breathing machine)?	intubated	one of {Yes, No, Not sure}
How often did you miss work or school due to asthma in the last 6 months?	miss_work	one of {I don't work or go to school, I did not miss work or school, 1 day, 2-5 days, 6-9 days, 10 days or more}
How often have you had asthma symptoms over the past month, on average?	symptoms	one of {Every day, Most but not all days, 2-3 days per week, 1-4 days per month, Less than once per month, Never}
How many nights did asthma symptoms wake you up from sleep over the past month, on average?	nights	one of {Every day, Most but not all days, 2-3 days per week, 1-4 days per month, Less than once per month, Never}
How many days has your asthma limited your activity over the past month, on average?	limited activity	one of {Every day, Most but not all days, 2-3 days per week, 1-4 days per month, Less than once per month, Never}
Have you been hospitalized for asthma in the past 6 months?	times_hospitalized	yes/no
How many times were you hospitalized for asthma in the past 6 months?	hospitalized_times	integer input
Have you been in emergency department for asthma in the past 6 months?	emergency	yes/no
How many times were you in the emergency room for asthma in the past 6 months?	emergency_times	integer input
Have you seen a doctor for asthma in the past 6 months, other than for a routine visit?	seen_doc	yes/no
How many times did you see a doctor for asthma in the past 6 months, other than your routine visits?	doc_times	integer input
Over the past 6 months, how many times have you taken a course of oral steroids (prednisone, medrol, etc.) for three days or more to treat an asthma attack?	oral steroids	one of {None, One, Two, Three or more}
Asthma Medication		
Question	Name	Details
Are you prescribed daily asthma control medication?	prescribed_asthma_control_medication	one of {Yes, No, Not Sure}
Which of the following daily inhaled asthma controller medicines are you prescribed?	daily_inhaled_medicine	one of {Advair Diskus, Alvesco, Advair HFA, Dulera, Qvar, Flovent Diskus, Symbicort, Flovent HFA, Pulmicort, Asmanex, Breo Ellipta, Aerospan

		(flunisolide)}
What is the dose of your Advair Diskus?	advair_diskus_dose	one of {100/50 mcg, 250/50 mcg, 500/50 mcg}
What is the dose of your Alvesco?	alvesco_dose	one of {80 mcg, 160 mcg}
What is the dose of your Advair HFA?	advair_hfa_dose	one of {45/21 mcg, 115/21 mcg, 230/21 mcg}
What is the dose of your Dulera?	dulera_dose	one of {100/5 mcg, 200/5 mcg}
What is the dose of your Qvar?	qvar_dose	one of {40 mcg, 80 mcg}
What is the dose of your Flovent Diskus?	flovent_diskus_dose	one of {100 mcg, 250 mcg}
What is the dose of your Symbicort?	symbicort_dose	one of {80/4.5 mcg, 160/4.5 mcg}
What is the dose of your Flovent HFA?	flovent_hfa_dose	one of {44 mcg, 110 mcg, 220 mcg}
What is the dose of your Pulmicort?	pulmicort_dose	one of {90 mcg, 180 mcg}
What is the dose of your Asmanex?	asmanex_dose	one of {110 mcg, 220 mcg}
What is the dose of your Breo Ellipta?	breo_dose	one of {100/25 mcg, 200/25 mcg}
How many total puffs per day do you take of your controller medicine?	control_puffs	integer input
On average, I take my prescribed asthma control medication...	controlmed	one of {Every day, 5-6 days per week, 2-4 days per week, Once per week or less, I am not prescribed asthma controller medication, I don't know which of my medicines is my daily asthma control medication}
Which of the following other daily inhaled, injected or oral controller medications are you prescribed? (check all that apply)	daily_controller_medication	multiple choice from of {Spiriva inhaler, Tudorza inhaler, Accolate or Zafirlukast, Singulair or Montelukast, Zflo or Zileuton, Theophylline, Xolair injections, None of the above, I choose not to answer}
Which of the following oral steroids are you prescribed?	steroid_which	one of {Medrol, Prednisone, Prednisolone, I am not prescribed an oral steroid}
What is the dose of your oral steroid (mg)?	steroid_dose	integer input
Do you use a nebulizer?	use_nebulizer	yes/no
Which medicines do you nebulize? (select all that apply)	nebulizer_meds	multiple choice from of {Albuterol, Levalbuterol/Xopenex, Budesonide/Pulmicort, Atrovent/Ipratropium, Arformoterol/Brovana, Duoneb (albuterol/ipratropium)}
Do you use your nebulizer daily?	nebulize_daily	yes/no
Which of the following medicines do you take for quick relief of asthma symptoms? (check all that apply)	quick_relief	multiple choice from of {Atrovent, Combivent, Proair, Proventil, Ventolin, Xopenex, Other}
Over the past month, how often did you use your quick relief inhaler, on average?	past_month_quick_relief	one of {Daily, Most but not all days, 2-3 days per week, 1-4 x per month, Not at all}
On days when you use your quick relief inhaler, how many total puffs do you take in a day, on average?	daily_yes	integer input

Do you use any other asthma medications? Please include the names, doses and how often you use them.	other_meds	text input
Weekly Survey		
Question	Name	Details
In the last week, did you see your asthma doctor/provider because your asthma got worse (other than for a regular visit)?	asthma_doc_visit	yes/no
Did your doctor/provider change your asthma medicine in the last week?	asthma_medicine	yes/no
Did your doctor/provider prescribe you oral steroids (prednisone, medrol, etc) or did you use oral steroids according to your Asthma Action Plan?	oral_steroids	yes/no
What day did you start your course of oral steroids (prednisone, medrol, etc)?	oral_steroids_when	date
Did you use oral steroids (prednisone, medrol, etc) according to your asthma action plan within the last week?	prednisone	yes/no
What day did you start your course of oral steroids?	prednisone_when	date
Did you visit the emergency room for an asthma attack within the last week?	emergency_room	yes/no
What day did you go to the Emergency Room?	er_when	date
Were you admitted to the hospital for an asthma attack within the last week?	admission	yes/no
What day were you admitted to the hospital?	admitted_when	date
What day were you released from the hospital?	admitted_end	multiple choice from of {Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, I'm still in the hospital}
Did you have any activity limitation due to asthma?	limitations	yes/no
How many days in the past week was your activity limited because of asthma symptoms?	limitations_days	integer input
Have you missed work or school due to your asthma?	missed_work	yes/no
Which days did you miss school or work in the past week because of asthma symptoms? (Select all that apply)	missed_work_days	multiple choice from of {Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday}
Did you have any side effects from your asthma medicines within the last week?	side_effects	one of {None, Minor, Made me change my medicine}

EQ_5D: The EuroQol Group (1990). EuroQol-a new facility for the measurement of health-related quality of life. Health Policy 16(3):199-208.		
Question	Name	Details
Your mobility TODAY	mobility	one of {I have no problems walking, I have slight problems walking, I have moderate problems walking, I have severe problems walking, I am unable to walk}
Your self-care TODAY	selfcare	one of {I have no problems washing or dressing myself, I have slight problems washing or dressing myself, I have moderate problems washing or dressing myself, I have severe problems washing or dressing myself, I am unable to wash or dress myself}
Your usual activities TODAY	usual_activities	one of {I have no problems doing my usual activities, I have slight problems doing my usual activities, I have moderate problems doing my usual activities, I have severe problems doing my usual activities, I am unable to do my usual activities}
Your pain / discomfort TODAY	pain	one of {I have no pain or discomfort, I have slight pain or discomfort, I have moderate pain or discomfort, I have severe pain or discomfort, I have extreme pain or discomfort}
Your anxiety / depression TODAY	depression	one of {I am not anxious or depressed, I am slightly anxious or depressed, I am moderately anxious or depressed, I am severely anxious or depressed, I am extremely anxious or depressed}
Please tap the scale to indicate how your health is TODAY (100 means the best health you can imagine)		
0 means the worst health you can imagine)		
	health_today	
Medical History		
Question	Name	Details
Have you ever had a heart attack?	heart_attack	yes/no
Have you ever had congestive heart failure?	Congestive	yes/no
Do you have or have you had peripheral vascular disease (disease of your circulation, including an aortic aneurysm)?	peripheral	yes/no
Do you have dementia?	dementia	yes/no
Do you have or have you had chronic pulmonary pulmonary disease (emphysema or chronic bronchitis)?	chronic_pulmonary_disease	yes/no
Do you have or have you had another lung disease other than asthma?	other_lung_disease	yes/no

Do you have or have you had connective tissue disease (lupus, rheumatoid arthritis, or similar diseases)?	tissue	yes/no
Do you have or have you had peptic ulcer disease?	ulcer	yes/no
Do you have or have you had moderate or severe renal (kidney) disease?	kidney	yes/no
Do you have or have you had leukemia (acute or chronic)?	leukemia	yes/no
Do you have or have you had malignant lymphoma?	malignant_lymphoma	yes/no
Have you had any malignant solid tumors within the past 5 years (including melanoma but excluding other skin cancer)?	tumor	one of {Metastatic (spread to other organs besides the starting organ), Non-metastatic (not spread to other organs), No}
Have you ever had any cerebrovascular disease (including a stroke or embolism)?	stroke	one of {With paralysis of one side of the body, Without paralysis of one side of the body, No}
Do you have or have you had liver disease?	liver	one of {Moderate or severe, Mild, No}
Do you have or have you had arthritis?	arthritis	yes/no
Have you been tested for allergies with skin tests or blood tests?	tested	yes/no
Which of the following cause you to have an allergic reaction? (check all that apply)	allergic_to	multiple choice from of {House dust mites, Mold, Weed pollen, Grass pollen, Tree pollen, Rodents (rats, mice), Cockroaches, Dogs, Cats, Other}
Milestone Survey		
Question	Name	Details
How tall are you? (inches)	height	integer input
How much do you weigh? (pounds)	weight	integer input
What is your gender?	gender	one of {Male, Female}
How old are you?	age	integer input
How often have you had asthma symptoms over the past month, on average?	symptoms2	one of {Every day, Most but not all days, 2-3 days per week, 1-4 days per month, Less than once per month, Never}
How many nights did asthma symptoms wake you up from sleep over the past month, on average?	nights2	one of {Every day, Most but not all days, 2-3 days per week, 1-4 days per month, Less than once per month, Never}
How many days has your asthma limited your activity over the past month, on average?	limited activity2	one of {Every day, Most but not all days, 2-3 days per week, 1-4 days per month, Less than once per month, Never}
Over the past month, how often did you use your quick relief inhaler, on average?	past_month_quick_relief2	one of {Daily, Most but not all days, 2-3 days per week, 1-4 x per month, Not at all}
On days when you use your quick relief inhaler, how many total puffs do you take in a day, on average?	daily_yes2	integer input

Have you used your asthma action plan (a written plan for how to change your medicine if your asthma gets worse from your doctor) since you have started the Asthma Mobile Health Study?	use_aap	yes/no
Has the Asthma Mobile Health Study prompted you to create an asthma action plan (a written plan for how to change your medicine if your asthma gets worse from your doctor)?	make_aap	yes/no
Have you had your lung function measured by your asthma doctor (a test called spirometry) since joining the study?	got_spirometry	yes/no
If you use your peak flow meter to monitor your asthma, enter your best peak flow. (L/min)	best_peakflow	integer input
What currently troubles you most about your asthma?	current_troubles	one of {Limits how active I am, Makes me miss work/school, I have to use my quick relief medicine all the time, I can't remember to take my controller medicine, I can't sleep because my asthma wakes me up at night, I never know when something will trigger my asthma, I hate it when I have an attack and I have to go to the emergency room or hospital, None of the above}
Has your participation in the Asthma Mobile Health Study helped to alleviate some of the troubles you have with your asthma?	alleviate_troubles	one of {Strongly agree, Agree, Neither agree nor disagree, Disagree, Strongly disagree}
Select your current goal(s) for asthma control:	current_goal	multiple choice from of {Be more active, Miss/interfere work/school less often, Use less quick relief medicine, Remember to take my control medicine, Sleep through the night without symptoms, Avoid things that trigger my asthma, Avoid asthma attacks, I do not have any current goals for my asthma control}
Has your participation in the Asthma Mobile Health Study helped you to achieve your initial goals for asthma control?	met_goal	one of {Strongly agree, Agree, Neither agree nor disagree, Disagree, Strongly disagree}
Do you think using the Asthma Health app helped you prevent a visit with a clinician?	prevent_visit	one of {Yes, No, I don't know/I am not sure}
Do you think using the Asthma Health app prompted a visit with a clinician about your asthma control?	cause_visit	one of {Yes, No, I don't know/I am not sure}
Do you think using the Asthma Health app helped you to prevent an ED visit or hospitalization?	prevent_ed	one of {Yes, No, I don't know/I am not sure}
Your Asthma		
Question	Name	Details

My asthma gets worse with (check all that apply):	asthma_gets_worse_with	multiple choice from of {A cold, Exercise, Being more active than usual (walking, running, climbing stairs), Strong smells (perfume, chemicals, sprays, paint), Exhaust fumes, House dust, Dogs, Cats, Other furry/feathered animals, Mold, Pollen from trees, grass or weeds, Extreme heat, Extreme cold, Changes in weather, Around the time of my period, Poor air quality, Someone smoking near me, Stress, Feeling sad, angry, excited, tense, Laughter, I don't know what triggers my asthma, None of these things trigger my asthma}
Have you had a flu shot this year?	flushot	yes/no
Did you know: Flu vaccine is recommended for most people with asthma?	flu_prompt	yes/no
Do you have an asthma action plan (a written plan for how to change your medicine if your asthma gets worse from your doctor)?	plan	yes/no
Did you know: People with asthma are recommended to develop an asthma action plan with their healthcare provider that gives specific instructions for early treatment of asthma symptoms?	aap_prompt	yes/no
Have you ever had your lung function measured by your asthma doctor (a test called spirometry)?	lung_function	one of {Yes, No, Not sure}
If you use your peak flow meter to monitor your asthma, enter your best peak flow (L/min)	peak_flow	integer input
What troubles you the most about your asthma? (select all that apply)	troubles_about_asthma	multiple choice from of {Limits how active I am, Makes me miss work/school, I have to use my quick relief medicine all the time, I can't remember to take my daily controller medicine, I can't sleep because my asthma wakes me up at night, I never know when something will trigger my asthma, I hate it when I have an attack and I have to go to the emergency room or hospital, None of the above}
Select your goal(s) for asthma control (select all that apply)	asthma_control	multiple choice from of {Be more active, Miss/interfere work/school less often, Use less quick relief medicine, Remember to take my control medicine, Sleep through the night without asthma symptoms, Avoid things that trigger my asthma, Avoid asthma attacks}

2
3
4
5

Supplementary Text 2. Marketing

As one of the first five ResearchKit launch partners, Asthma Mobile Health Study was part of Apple's marketing efforts. Additionally, the Icahn Institute for Genomics and Multiscale Biology at Icahn School of Medicine at Mount Sinai also publicized the study via:

- Created a Asthma Mobile Health Study microsite: <http://apps.icahn.mssm.edu/asthma>
- Partnered with Apple for a ResearchKit launch video and media outreach:
<https://www.youtube.com/watch?v=92V5QcOTbFw>
- Published two press releases and pitched the app to over 40 journalists - our top media hits, including CBS, Business Insider and Nature can be found at
<http://apps.icahn.mssm.edu/asthma/news>
- Co-promoted the app with advocacy groups including Asthma UK
- Published over 400 posts on 12 social media channels, including Mount Sinai Hospital's Facebook (215,000 subscribers - <https://www.facebook.com/mountsinainyc>)
- Centralized Twitter conversation around the app using #AsthmaApp hashtag:
<https://twitter.com/hashtag/asthmahealth?vertical=default>
<https://twitter.com/hashtag/asthmahealth?vertical=default>

Presented app findings at several conferences, including GET Labs (<http://www.getconference.org/get2016/labs.html>) and SciViz (https://www.youtube.com/watch?v=zfe1p_o_SE&feature=youtu.be)

Supplementary Text 3. IRB related challenges

Institutional support, the IRB and mobile health research studies pose new concerns not raised by traditional research methods. Our institutional leadership helped bring together key stakeholders necessary for the successful implementation and execution of the AMHS. This process required working iteratively with the IRB, weighing the risks and benefits of the use of this mobile health technology. We had extensive discussions around user authentication, privacy protection and data security—not only to ensure that the data were secure, but that it would be inaccessible by Apple. The IRB was particularly concerned with whether the informed consent delivered electronically would lead to better or worse informing compared with traditional informed consent procedures. The IRBs consulted with internal and external legal experts and ethicists to help address these important issues. This earnest effort by all stakeholders led to the development of new guidelines and tools to address the unmet needs of the evolving digital health landscape.

Through this protracted discussion with the IRB, we came to appreciate that IRBs will need to develop new expertise to evaluate these issues more efficiently and effectively. Specifically, mobile health study design requires a balance between appropriate human subject protection according to the regulatory guidelines and app design/user experience that both commercial app stores and prospective study participants find acceptable (i.e., not unduly burdensome). The need to obtain IRB approval for an app's design, increases the complexity and cost of the app development process because of additional usability and patient safety considerations. Poor mobile health app usability may result in commercial app stores not approving for distribution and/or too few subjects willing to participate in the study.

For example, the study team initially received guidance to require potential study participants to watch a lengthy video of the principal investigators highlighting (repeating) the contents of the study consent document as part of the e-consent process. This prototype was rejected for publication by the App Store because it failed to meet the App Store's guidelines for the onboarding experience. The consent video was made optional and the revised e-consent process and app were approved by both the IRB and the App Store. Additionally, mHealth studies may be developed in partnership with commercial entities such as app developers or mobile platform providers. If so, researchers may need to balance fixed commercial launch date deadlines with the more open-ended, iterative IRB review cycles.

Thus, we proposed the following for efficient IRB review of mHealth study protocols:

1. The inclusion of one or more IRB members with knowledge of current federal regulatory guidelines in the DH space; additionally, basic knowledge of app usability and user interface guidelines of the app store(s) in which researchers intend to distribute the app.
 2. Define clear standards for mHealth app consent and enrollment process. Ideally, the creation of a free, open source mHealth study app template for study developers.
- Specifically:

- 73 • Start screen showing controls for reading and emailing a copy of the study protocol
- 74 • Placeholders in the onboarding flow for optional study-specific content like educational videos
- 75 • Prospective screening flow to determine subject eligibility
- 76 • Consenting highlights educational flow including links so subjects who wish to can learn more about any
- 77 topic(s) of their choosing
- 78 • Quiz for verifying that the subject understands the most critical points of the consent
- 79 • Fields for patient to enter their name
- 80 • IRB-approved digital signature implementation
- 81 • Screens for subject to grant access to passive data (e.g. iOS HealthKit, Android Google Fit, etc.)
- 82 • Screens for collecting subject contact information such as email or phone number (including an e-mail
- 83 confirmation flow)
- 84 • Screens for gathering protected health information typically gathered for studies, such as birth date and
- 85 gender
- 86
- 87
- 88

Supplementary Text 4. Software development and technical challenges.

Apple and their software development partner created the first five ResearchKit apps using the Objective-C programming language, including AMHS through version (1.1). The features and issues for the AMHS were tracked using JIRA (Atlassian Software). During this phase, prioritization went to critical bugs that impacted app usage and data collection, such as bounding values on surveys, adding HealthKit data collection and of non-identifiable demographic information (e.g. height, weight, age and gender). Once the application code was transferred to the LifeMap Solutions, prioritization went to incorporation of Amazon Mobile Analytics and Crashlytics to better assess app performance and usage, fixing critical bugs (loss of activity reminders, etc.), creation of new features (Doctor Dashboard, etc), and complying with the IRB (modification submissions for enhancements). Other enhancements such as automation to the build and deployment process and the addition of unit testing could not be addressed during the first six months post app launch.

Subsequent app developments were completed using two-week agile sprints. Bug fixes and new features came in the form of pull request to a private BitBucket repository. These pull request were code reviewed before they were merged into the development branch. An engineer then created a manual app build for our tester (this process had since been automated). Once the testing was complete, the JIRA ticket was then closed.

As a pioneering app of ResearchKit, some technical challenges were expected and encountered in the process from coding to testing to deploying AHA. Testing an app that must perform over a 6-month period, with rolling enrollment introduced several challenges. Features, for example, once live, must perform properly for both the new users as well as the previously enrolled users who may be upgrading from any version of the app.

118

119 **Supplementary Text 5. Excerpt of research protocol and data analysis plan in Institutional**
120 **Review Board application.**

121 (Our current manuscript addressed Specific Aims 1, 2, and 4 of the research protocol approved
122 by the Mount Sinai IRB. Specific Aim 3 and 5 will be addressed independently of this paper. Also,
123 due to the exploratory nature of the study and the expansion of our research team to include
124 additional statisticians and data scientists, our analysis plan was subsequently fleshed out and
125 modified accordingly.)

126

127 *Brief Summary of Research:*

128 Mobile health applications (MHA) are increasingly being explored as tools to assist in management of
129 chronic diseases. Although quite a few chronic disease management MHA's have been developed, little
130 is known regarding which characteristics of MHAs are effective and there is limited data suggesting a
131 real-world impact on health outcomes. Asthma is one of the most common and costly of the chronic
132 diseases, impacting a broad range of the population including both children and adults. It is a variable
133 disease necessitating regular medication use, monitoring of symptoms, and avoidance of specific
134 triggers. These characteristics of asthma make it a chronic disease that is particularly amenable to having
135 an MHA facilitate active monitoring outside of periodic traditional medical visits. To date, most MHAs for
136 asthma have been studied in small, select patient populations in traditional research settings, limiting their
137 generalizability to the population of patients with asthma at large. Our group previously designed a MHA
138 focused on asthma subjects to test the feasibility of conducting an asthma mobile health application
139 (AHMA) study. This study is a collaboration between MHA developers and Mount Sinai faculty with
140 expertise in the fields of asthma, research design, data storage, and data analysis. The pilot AMHA
141 incorporated elements of usual clinical care (that may take place during typical office visits), such as
142 medication reminders, a daily asthma diary to track asthma control (AC) and medication use, patient
143 education and assessments of quality of life (QoL), and health care utilization (HCU).

144 *Research Questions:* Is recruiting, consenting, and enrolling patients in an asthma research study via a
145 MHA downloaded through the Apple App Store without direct participant contact feasible and how will use
146 of the AMHA impact asthma-related outcomes?

147 The primary objectives of the proposed study are to:

- 148 1. Determine the feasibility of recruiting, consenting and enrolling a larger cohort of subjects
149 remotely via the App Store
- 150 2. Collect data using an asthma diary, provide automated reminders for daily medication, and
151 administer health surveys to a larger cohort over an extended period of time (6 months to a year)
- 152 3. Assess whether participants will access and share additional information via optional features
153 over 6 months to a year of follow-up (e.g. educational videos, wearable health/ fitness monitoring
154 devices, etc.)

4. Assess asthma control (AC), quality of life (QoL), and health care utilization (HCU) of our cohort over the 6 months to a year study period

5. To develop algorithms for personalized trigger avoidance and a personalized asthma action plan that will be validated by a panel of asthma experts for medical accuracy and used, if validated, in future versions of the AMHA

Hypothesis 1 (H1): Participants can be recruited, enrolled, and consented to participate in an asthma study remotely via the Apple App Store using an AMHA.

Specific Aim 1 (SA1): To determine the feasibility of e-recruitment, e-consenting and remotely enrolling subjects into an asthma study via the Apple App Store without in-person contact with study investigators using an AMHA.

Hypothesis 2 (H2): Participants will input data regarding their asthma symptoms, medication use, and health care utilization into the AMHA using an electronic asthma diary over 6 months to a year.

Specific Aim 2 (SA2): To determine adherence rates to use of various AMHA features including a daily asthma diary, weekly assessments of asthma control and health care utilization, and monthly assessments of quality of life (EQ-5D-5L) over a 6 month to a year study period.

Hypothesis 3 (H3): Patients will access the AMHA's educational materials, utilize medication refill reminder options, environmental triggers alerts, and if applicable, share external monitoring device data.

Specific Aim 3 (SA3): To assess the frequency and persistence of usage of the AMHA's optional features (i.e. educational modules, environmental/weather alerts, sharing data from wearable health and fitness monitoring devices with the study, etc.) over 6 months to a year

Hypothesis 4 (H4): Asthma patients who utilize the AMHA can improve their Asthma Control, Quality of Life and/or their Health Care Utilization.

Specific Aim 4 (SA4): To perform an exploratory study to determine the effect of AMHA on patient's AC, QoL, HCU over the 6 month study period while controlling for confounding covariates.

Hypothesis 5 (H5): Data collected from the AMHA can be used to design algorithms for personalized trigger avoidance and to generate a personalized asthma action plan to assist with asthma management.

Specific Aim 5 (SA5): To develop and validate the medical accuracy of algorithms for personalized trigger avoidance (see specific aim 5A below) and a personalized asthma action plan based (see specific aim 5B below) on participants' data collected via the AMHA.

SA5A: Algorithms for personalized trigger avoidance will be developed based on forming

correlations between asthma symptoms, location data (only if the feature is available at the time of use, longitude and latitude collected at least one time daily when the app is opened and location services of the iPhone are turned on) and location specific environmental information (general local air quality, weather, pathogens, etc.).

SA5B: To develop an algorithm for a personalized asthma action plan using baseline medication input by the participants in according to recommendations outlined in the Global Initiative for Asthma 2014⁹

Feedback from SA5 algorithms **will not be** provided to participants in this study. For this specific aim, algorithms will be validated by asthma experts for medical accuracy and appropriateness prior to use in future versions of the AMHA. Please see Outcomes Measures and Statistical Plan sections below for details.

Outcome Measures (OM):

OM 1: Descriptive data to quantify and determine feasibility and implied participant acceptability of AMHA:

-All subjects (de-identified data): Number of people who downloaded AMHA, % aborted study participation before completing inclusion/exclusion criteria.

This data will be gathered for all subjects (including those who never finish the e-consent process) by using technical measures such as the server download log files and the use tracking features of the AMHA. The server download log files will be configured to automatically remove personally identifying information such as IP addresses that is automatically gathered in the course of normal server functioning, leaving only deidentified totals. The use tracking features of the AMHA will be configured to avoid gathering personally identifying information wherever possible in the first place (such as for the values that users type into fields), and where this may not be possible (such as for IP addresses automatically gathered by server logs), they will be configured to automatically remove the personally identifying information. Therefore, these first two data points will only be available in de-identified form.

-All subjects who complete e-consent process (coded data): demographics of enrolled subjects, subject retention and loss to follow-up, time after consenting before first use of app, frequency of use of app, use of specific features of the app, etc. In addition to the coding steps for the server logs and the application usage logging noted above, this information will be submitted through and coded by the Sage Bionetworks Bridge Server.

OM 2: Descriptive data to quantify and determine feasibility and implied participant acceptability of specific mandatory AMHA features:

222 Frequency of use and % of completion of each of the following for all participants and randomized strata:

223 Daily asthma diary (of asthma symptoms and use of quick relief medicine) and medication

224 reminders

225 Baseline and weekly questionnaires on AC and HCU, as well as monthly EQ-5D-5L

226 OM3: Descriptive data to quantify and determine feasibility and implied participant acceptability of specific

227 optional AMHA features:

228 Frequency of use of each of the following if applicable: Educational modules, utilize reminder

229 options, environmental triggers alerts, and sharing data from wearable health and fitness monitoring

230 devices

231 OM4: The change in AC, QoL, and HCU over the 6 month study period while controlling for confounding

232 covariates:

233 Monthly assessments of asthma control and assessment of QoL and HCU at 3 and

234 6 months adjusting for covariates known to be associated with AC, QoL, and HCU

235 including age, gender, race/ethnicity, socio-economic status, and BMI.

236 .

237 OM5: We will develop and validate algorithms *for personalized trigger avoidance (see OM5A) and*

238 *a personalized asthma action plan (see OM5B)* that will be developed based on participants' data

239 input into the AMHA.

240 OM 5A: We will develop and validate algorithms for personalized trigger avoidance.

241 Algorithms on personalized trigger avoidance will be developed based on our forming correlations

242 between asthma symptoms, location data (if available at the time of app use longitude and latitude

243 collected at least one time daily when the app is opened and location services of the iPhone are turned

244 on), and location specific environmental information (if available at the time of app use general local air

245 quality, weather, pathogens, etc.). Data for specific environmental information may be sourced from the

246 free Airnow.gov feed for pollution. Location data of this level of specificity will allow us to identify and look

247 at impact on asthma of specific weather events, point sources of allergens, and point sources of poor air

248 quality. As part of our research aims, the potential feedback produced by these algorithms will be

249 undergo a qualitative analysis by a panel of asthma experts who are coinvestigators in this study. This is

250 an exploratory outcome, and feedback based on the algorithms will not be provided to participants at this

251 stage.

252 OM5B: Validity of personalized asthma action plan generated by our algorithms. An

253 algorithm using baseline medication input by the participants will be used to generate an asthma

254 action plan according to best-studied regimens outlined by the Global Initiative for Asthma 2014⁹.

255 In a clinical setting, a patient and their provider develop an asthma action plan that provides

written instructions to increase the dose of inhaled corticosteroids or initiate a course of rescue prednisone based on symptoms or peak flow measurements specified by their providers in the event of worsening asthma. A number of standard changes to baseline inhaler regimens or initiation of oral prednisone are suggested based on clinical research studies and according to the level of baseline medication regimen (low, medium, or high dose inhaled corticosteroid) or use of combination inhaled corticosteroid/long acting beta agonist inhaler. These standard regimens would be amenable for programming into an app based on a participant's recorded baseline medication. Provision of an app based formula- driven action plan could potentially simplify this process for primary providers who are often not adequately trained in the skill of providing an action plan and thus often don't provide them to patients. In this exploratory aim, we will be generating formulas based on GINA 2014 recommended regimens to produce a suggested action plan. As part of our research aims, the resulting action plans will undergo a qualitative analysis by a panel of asthma experts who are coinvestigators in this study for accuracy and validity. They will not be provided as feedback to participants at this stage of our research.

Statistical Methods:

The statistical analyses will be conducted using SAS software (version 9.3, Cary, NC) and R software (version 3.0.1).

Statistical Plan for Specific Aims: These aims are descriptive. Univariate analyses of variables will be performed. For continuous variables, means and standard deviations or medians and interquartile range will be reported based on data distribution. Normality will be tested for continuous variables.

For categorical variables, we will calculate frequencies and proportions. Two-sided p values with a level of statistical significance of $p < 0.05$ will be used for all of the analyses unless otherwise specified

SA1: We will report the number of subjects that downloaded the app, aborted before completing the inclusion/exclusion criteria, demographics of enrolled subjects, subject retention and loss to follow-up, time after recruitment before first use of app, etc.

SA2: Frequency of use and % of completion of 1. daily asthma diary and reminders 2. Baseline then weekly AC and HCU, and three and 6 month EQ-5D-5L questionnaires. We will use a log-rank test to assess two-way superiority of current daily surveys versus modified daily surveys for each strata randomized to A/B daily surveys. We will also use McNemar's test to compare A/B matched strata for daily survey response rates on days 0-30, in order to control type-I error.

SA3: Frequency of use of educational modules, pollen/weather alerts, sharing data from wearable or supporting electronic devices, etc.

SA4: To perform an exploratory study to determine the effect of AMHA on patient's AC, QoL, and HCU over the 6 month study period while controlling for confounding covariates.

Outcome measure: Δ baseline to weekly AC and HCU as well as three month and 6 month QoL scores while controlling for covariates. Next, we will examine whether the effect of AMHA on endpoints are modified by covariates, as well as whether AMHA effect are different among certain patient groups (stratified by covariates). Repeated measure ANOVA or mixed effect model will be applied if >2 scores for each questionnaire are available. We will use time series to correct for seasonal variation in AC, QoL, and HCU scores. Furthermore, we will combine the 3 endpoint scores together and conduct a multivariate ANOVA analysis to assess the effect of the AMHA.

SA5: Validation of algorithms for trigger feedback and asthma action plans

SA5A: Validation of algorithms for trigger feedback.

Outcome measure: Correlation between asthma symptoms, location data, and air quality sources (general local air quality, weather, pollen levels, and location near point sources of pollution) will be performed using generalized linear models. Repeated measures will be accommodated with mixed effects models. R statistical package (version 3.0.2) will be used for the analyses. If a statistically significant pattern is noted between a specific trigger and patient's symptoms, a message will be generated to potentially provide as a trigger warning for future studies. For our current study, no feedback or messages will be sent to trial participants. We plan to use our proposed study to assess the validity of these trigger warnings via qualitative analyses (e.g. sensitivity/specificity) to determine the appropriateness of these algorithm generated recommendations.

SA5B: Validation of algorithms for asthma action plans

Outcome assessment: Qualitative analysis (e.g. sensitivity/specificity) by asthma experts of possible action plans generated by our algorithms to ascertain that they are in accordance to existing guidelines and standards of care.

```
#####  
#####AnalysisCode/  
#####
```

NBT_maincode.R ##This is the main analysis code which contains code for generating all figures/tables except those mentioned below.

amha_data_preprocessing_manuscript_Jan2017.html ##code for pre-processing of daily and weekly survey data

triggeranalysis.R ##code for generating main figure 4 and Supplementary Figures 1b-c and 5. Note that we do not provide the file 'longlat.txt', which contains location data, to protect user privacy

userRetentionFigures_manuscript-Jan2017.html ##code for generating Figure 2d-e, Supplementary Figure 3, Supplementary Table 4a-c, and Supplementary Figure 2a-b

main code.txt

```

library(ggplot2)
library(reshape2)
library(gplots)
library(maps)
library(mapproj)
library(xtable)
library(wordcloud)
library(colorspace)
##### FUNCTION DEFINITIONS
mon0 = c(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31)
get_y <- function(x, var) {
  y <- temp2[temp2$healthCode %in% x, var]
  if (length(y) == 0) {
    z = temp1[temp1$healthCode %in% x, var]
  } else {
    if (is.na(y)) {
      z = ifelse(length(temp1[temp1$healthCode %in% x, var]) > 0,
temp1[temp1$healthCode %in%
      x, var], y)
    } else {
      z = y
    }
  }
}

my.convert.day <- function(date.v, mon.index = mon0, year.c = year0) {
  ### day 0: 2015-3-9
  year = as.numeric(substring(date.v, 1, 4))
  mon = as.numeric(substring(date.v, 6, 7))
  day = as.numeric(substring(date.v, 9, 10))
  count = 0
  if (year - 2015 > 0) {
    count = (year - 2015) * year.c - 68
    if (mon > 1)
      count = count + sum(mon.index[1:(mon - 1)])
    count = count + day
  } else {
    if (mon > 3) {
      count = count + sum(mon.index[3:(mon - 1)]) - 9
      count = count + day
    } else {
      count = count + day - 9
    }
  }
  return(count)
}

GINA.cal = function(user.set) {
  GINA.matrix = matrix(NA, nrow = length(user.set), ncol = 4)
  colnames(GINA.matrix) = c("DaySym", "NighSym", "Reliever", "Activity")

  for (i in 1:length(user.set)) {
    cur.user = user.set[i]
    #### handle Quick reliver
    cur.pick = which(Med.data[, "healthCode"] == cur.user)
    if (length(cur.pick) != 0) {
      a = max(cur.pick)
      Quick.rel = Med.data[a, "past_month_quick_relief"]
      GINA.matrix[i, 3] = 0
      ### more than twice a week
      if (!is.element(Quick.rel, c("", "[5]", "[4]")))
        GINA.matrix[i, 3] = 1
    }
    #### handle the other three
    cur.pick = which(His.data[, "healthCode"] == cur.user)
    if (length(cur.pick) != 0) {
      a = max(cur.pick)
      Day.sym = His.data[a, "symptoms"]
    }
  }
}
lagina p

```

```

                                main code.txt
GINA.matrix[i, 1] = 0
if (!is.element(Day.sym, c("", "[5]", "[4]", "[6]")))
  ### day sym more than twice a week
GINA.matrix[i, 1] = 1

Nig.sym = His.data[a, "nights"]
GINA.matrix[i, 2] = 0
if (!is.element(Nig.sym, c("", "[5]", "[6]")))
  ### any night sym
GINA.matrix[i, 2] = 1

Act.sym = His.data[a, "limited activity"]
GINA.matrix[i, 4] = 0
if (!is.element(Day.sym, c("", "[5]", "[6]")))
  ### any limited activity
GINA.matrix[i, 4] = 1
}
}
GINA.v = apply(GINA.matrix, 1, sum)
GINA.indi = GINA.v
GINA.indi[GINA.v > 2] = "Uncontrolled"
GINA.indi[GINA.v > 0 & GINA.v < 3] = "Partly Controlled"
GINA.indi[GINA.v == 0] = "Well Controlled"
return(GINA.indi)
}

##### LOAD DATA
Dai.data <- read.table("Dai_data.txt", header = TRUE, sep = "\t", check.names =
FALSE) ##see: amha_data_preprocessing_manuscript_Jan2017.html
week.data <- read.table("week_data.txt", header = TRUE, sep = "\t", check.names
= FALSE) ##see: amha_data_preprocessing_manuscript_Jan2017.html
mile.data <- read.table("mile_data.txt", header = TRUE, sep = "\t", check.names
= FALSE)
You.data <- read.table("You_data.txt", header = TRUE, sep = "\t", check.names =
FALSE)
His.data <- read.table("His_data.txt", header = TRUE, sep = "\t", check.names =
FALSE)
MedH.data <- read.table("MedH_data.txt", header = TRUE, sep = "\t", check.names
= FALSE)
Med.data <- read.table("Med_data.txt", header = TRUE, sep = "\t", check.names =
FALSE)
YouA.data <- read.table("YouA_data.txt", header = TRUE, sep = "\t", check.names
= FALSE)
agesex1 = read.table("agesex1.txt", header = TRUE, sep = "\t", check.names =
FALSE)
Feed.data = read.table("Feed_data.txt", header = TRUE, sep = "\t", check.names =
FALSE)
cdc.prev = read.table("cdc_prev.txt", header = TRUE, sep = "\t", check.names =
FALSE)
enrollment = read.table("enrollment.txt", header = TRUE, sep = "\t", check.names
= FALSE)
##### define cohorts
baseline = as.character(You.data[, "healthCode"])
baseline = c(baseline, as.character(His.data[, "healthCode"]))
baseline = c(baseline, as.character(MedH.data[, "healthCode"]))
baseline = c(baseline, as.character(Med.data[, "healthCode"]))
baseline = c(baseline, as.character(YouA.data[, "healthCode"]))
baseline = c(baseline, as.character(Dai.data[, "healthCode"]))
baseline = unique(baseline)
length(baseline) #6,470
milestone.314 <- baseline[baseline %in% mile.data$healthCode]
length(milestone.314)
Mile.data = mile.data[mile.data$healthCode %in% milestone.314, ]
##### robust user definition
temp = apply(MedH.data[, c("Congestive", "chronic_pulmonary_disease",
"other_lung_disease")],
1, sum, na.rm = T)
user.nocompete = as.character(unique(MedH.data[temp == 0, "healthCode"]))
length(unique(MedH.data[temp != 0, "healthCode"]))

2agina p

```

```

                                main code.txt
pack.year = You.data[, "avg_cigarettes"]/20 * You.data[, "smoking_years"]
temp = You.data[, "smoking_status"] == "[1]"
temp1 = pack.year < 10
user.nonsmoke = as.character(You.data[temp | temp1, "healthCode"])
user.set0 <- as.character(intersect(user.nonsmoke, user.nocompete))
temp <- list(Dai.users = Dai.data[, "healthCode"], week.users = week.data[,
"healthCode"])
user.five <- as.character(names(table(unlist(temp)))[table(unlist(temp)) >= 5])
for_venn <- list(user.nonsmoke = user.nonsmoke, user.nocompetingrisk =
user.nocompete, user.five = user.five)
venn(for_venn)
user.set = names(table(unlist(for_venn))[table(unlist(for_venn)) == 3])
cohorts <- list(baseline = baseline, robust = user.set, milestone =
milestone.314)
lapply(cohorts, length)
##### FIGURE 1
map("state", interior = FALSE)
map("state", boundary = FALSE, col = "gray", add = TRUE)
# align data with map definitions by (partial) matching state, county names,
which include
# multiple polygons for some counties
my.state.name = state.name
for (i in 1:length(state.name)) {
  name = state.name[i]
  my.state.name[i] = paste(tolower(substr(name, 1, nchar(name))), sep = "")
}
temp = state.abb[match(map("state", plot = FALSE)$names, my.state.name)]
temp[34:37] = "NY"
temp[38:40] = "NC"
temp[53:55] = "VA"
temp[56:60] = "WA"
temp[20:22] = "MA"
temp[23:24] = "MI"
temp[8] = "DC"
mapstate <- temp
### we provide pre-processed file to protect exact user locations
bl.loc <- read.table("bl_loc.txt", header = TRUE, sep = "\t")
all.state = bl.loc$state
user.temp <- cohorts$baseline[!is.na(all.state)]
mytable <- table(bl.loc$state)
sum(mytable)
app.state.prev <- mytable/sum(mytable)
my.plot = list(NULL)
my.plot$percent = mytable[match(mapstate, names(mytable))]
colors = c("#F1EEF6", "#D4B9DA", "#C994C7", "#DF65B0", "#DD1C77", "#980043")
my.plot$colorBuckets <- as.numeric(cut(my.plot$percent, c(0, 10, 20, 50, 100,
300, 1000)))
leg.txt <- c("1-10", "10-19", "20-49", "50-99", "100-300", ">300")
colorsmatched <- my.plot$colorBuckets
# draw map
map("state", col = colors[colorsmatched], fill = TRUE, resolution = 0, lty = 0,
projection = "polyconic")
title("All Users")
legend("topright", leg.txt, horiz = TRUE, fill = colors, cex = 0.6)
##### FIGURE 1A
get_scatterplot2 <- function(x, y, my.xlab = "", my.ylab = "") {
  int <- intersect(names(x[!is.na(x)]), names(y[!is.na(y)]))
  xplot <- x[match(int, names(x))]
  yplot <- y[match(int, names(y))]
  forplot = data.frame(xplot, yplot, int)
  myrange = range(c(xplot, yplot))
  myrange <- range(myrange + 0.01)
  myrange <- range(myrange - 0.01)
  fit <- lm(log10(yplot) ~ log10(xplot))
  corr <- cor(log10(yplot), log10(xplot))
  out <- cor.test(log10(yplot), log10(xplot))
  mylabel = bquote(rho == .(format(corr, digits = 2)))
  textplot(log10(forplot$xplot), log10(forplot$yplot), row.names(forplot),
3

```

main code.txt

```

xlab = my.xlab,
  ylab = my.ylab, show.lines = FALSE, cex = 0.7)
abline(fit, lty = 2)
legend("bottomright", legend = as.expression(mylabel), bty = "n")
out
}
mytable <- table(all.state)
sum(mytable)
mytable <- mytable[mytable > 5]
app.state.prev <- mytable/sum(mytable)
cdc.perc.prev = as.numeric(cdc.prev$cdc.prev)
names(cdc.perc.prev) = cdc.prev$state
get_scatterplot2(app.state.prev * 100, cdc.perc.prev * 100, my.xlab = "Log10
Percentage Prevalence (AHA)",
  my.ylab = "Log10 Percentage Prevalence (CDC)")
##### TABLE 1
get_column <- function(user) {
  temp1 <- mile.data[mile.data$healthCode %in% user, c("gender", "age",
"healthCode")] # [1] is male [2] is female
  temp1$gender = as.character(ifelse(temp1$gender == "[1]", "Male", "Female"))
  temp2 <- agesex1[agesex1$healthCode %in% user,
c("NonIdentifiableDemographics.json.patientBiologicalSex",
  "NonIdentifiableDemographics.json.patientCurrentAge", "healthCode")]
  names(temp2) <- c("gender", "age", "healthCode")
  temp2$gender = as.character(temp2$gender)
  healthCode = unique(c(as.character(temp1$healthCode),
as.character(temp2$healthCode)))
  get_y <- function(x, var) {
    y <- temp2[temp2$healthCode %in% x, var]
    if (length(y) == 0) {
      z = temp1[temp1$healthCode %in% x, var]
    } else {
      if (is.na(y)) {
        z = ifelse(length(temp1[temp1$healthCode %in% x, var]) > 0,
temp1[temp1$healthCode %in%
      x, var], y)
      } else {
        z = y
      }
    }
  }
  age <- unlist(lapply(healthCode, get_y, "age"))
  gender <- unlist(lapply(healthCode, get_y, "gender"))
  age.class <- cut(age, c(min(age, na.rm = TRUE), 34, 64, max(age, na.rm =
TRUE)), labels = c("18-34",
  "35-64", "65+"))
  age <- table(age.class)
  gender <- table(gender)
  multirace <- levels(factor(You.data[!You.data$race %in% c("[1]", "[2]",
"[5]", "[1]",
  "[3]", "[6]", "[4]", "[7]", "[7,1]", "[7,2]", "[7,3]", "[7,4]", "[7,5]",
"[7,6]"),
]$race))
  hispanic <- You.data$ethnicity == "[1]" #587
  nonhispanic <- You.data$ethnicity == "[2]" #3396
  multi <- nonhispanic & You.data$race %in% multirace #196
  white <- You.data$race == "[5]" & nonhispanic #2776
  black <- You.data$race == "[1]" & nonhispanic #217
  noanswer <- You.data$race == "[7]" & nonhispanic #42
  other <- !multi & nonhispanic & !white & !noanswer #367, including
no answer
  new.race <- rep(NA, nrow(You.data))
  new.race[other] <- "other"
  new.race[multi] <- "multi"
  new.race[black] <- "black"
  new.race[hispanic] <- "hispanic"
  new.race[noanswer] <- "noanswer"
  new.race[white] <- "white"

```

main code.txt

```

new.race <- factor(new.race)
demo.data <- data.frame(You.data[You.data$healthCode %in% user, c("Income",
"health_insurance",
"education", "healthCode", "race")], new.race =
new.race[You.data$healthCode %in%
user])
insur <- table(factor(demo.data$health_insurance))
names(insur) <- c("Private", "Public", "none", "no answer")
race = table(factor(demo.data$new.race, levels = c("black", "white",
"other", "multi",
"hispanic"))))
education = table(factor(demo.data$education, levels =
levels(factor(demo.data$education))))
education <- c(hs.nongrad = sum(education[c("1]", "[2]"), na.rm = T),
hs.grad = as.numeric(education["3]"]),
some.college = sum(education[c("4]", "[5]"), na.rm = T), college.grad
= sum(education[c("6]",
"[7]"), na.rm = T))
income = table(factor(demo.data$Income, levels =
levels(factor(You.data$Income))))
income = income[!names(income) %in% "[7]"]
names(income) <- c("<$14,999", "$15,000-21,999", "$22,000-43,999",
"$44,000-60,000",
">$60,000", "I don't know")
clinical.data <- His.data[His.data$healthCode %in% user, c("healthCode",
"times_hospitalized",
"nights", "emergency", "emergency_times", "intubated", "symptoms",
"nights", "limited activity",
"seen_doc", "doc_times", "oral steroids", "age_when_diagnosed",
"miss_work", "hospitalized_times")]
emerg = table(factor(clinical.data$emergency, levels = c("TRUE", "FALSE")))
hosp = table(factor(clinical.data$times_hospitalized, levels = c("TRUE",
"FALSE")))
table(factor(demo.data$health_insurance, levels =
levels(factor(You.data$health_insurance))))
diag.class <- table(cut(clinical.data$age_when_diagnosed, c(0, 18, 24, 34,
44, 54, 64,
max(clinical.data$age_when_diagnosed, na.rm = TRUE))))
diag.class <- c("(0-18]" = as.numeric(diag.class["(0,18]"]), `19+` =
sum(diag.class[!names(diag.class) %in%
">18]"))
medi.data <- data.frame(Med.data[Med.data$healthCode %in% user,
c("prescribed_asthma_control_medication",
"daily_inhaled_medicine")])
acpresc <- table(factor(medi.data$prescribed_asthma_control_medication))
names(acpresc) <- c("Yes", "No", "Not sure")
dailymed <- table(factor(medi.data$daily_inhaled_medicine))
dailymed <- c(`ICS/LABA` = sum(dailymed[c("1]", "[3]", "[4]", "[7]")]), ICS
= sum(dailymed[c("2]",
"[5]", "[6]", "[8]", "[9]", "[10]"))))
#####
gina = table(factor(GINA.cal(user), levels = c("Uncontrolled", "Partly
Controlled", "Well Controlled")))
#####
count <- list(age = age, gender = gender, race = race, education =
education, income = income,
emerg = emerg, hosp = hosp, diag.class = diag.class, acpresc = acpresc,
dailymed = dailymed,
gina = gina)
perc <- lapply(count, function(x) round(x/sum(x), 2))
out <- paste(unlist(count), paste("(", unlist(perc), ")", sep = ""), sep = "
")
names(out) <- names(unlist(count))
out
}
get_column2 <- function(user) {
demo.data <- data.frame(You.data[You.data$healthCode %in% user,
c("health_insurance",

```

main code.txt

```

    "healthCode"]])
    insur = table(factor(demo.data$health_insurance))
    names(insur) <- c("Private", "Public", "none", "no answer")
    clinical.data <- His.data[His.data$healthCode %in% user, c("healthCode",
"times_hospitalized",
    "nights", "emergency", "emergency_times", "intubated", "symptoms",
"nights", "limited activity",
    "seen_doc", "doc_times", "oral steroids", "age_when_diagnosed",
"miss_work", "hospitalized_times")]
    intub <- table(factor(clinical.data$intubated))
    names(intub) <- c("Yes", "No", "Not sure")
    miss_work <- table(factor(clinical.data$miss_work))
    names(miss_work) <- c("not applicable", "did not miss", "1day", "2-5day",
"6-9day", "10+day")
    symp <- table(factor(clinical.data$symptoms))
    names(symp) <- c("everyday", "most", "2-3", "1-4", "less than once per
month", "never")
    night <- table(factor(clinical.data$nights))
    names(night) <- c("everyday", "most", "2-3", "1-4", "less than once per
month", "never")
    activity <- table(clinical.data["limited activity"])
    names(activity) <- c("everyday", "most", "2-3", "1-4", "less than once per
month", "never")
    hosptimes <- table(cut(clinical.data$hospitalized_times, c(0, 1, 3, 6, 55),
include.lowest = T,
    right = FALSE))
    emergtimes <- table(cut(clinical.data$emergency_times, c(0, 1, 3, 6, 11,
50), include.lowest = T,
    right = FALSE))
    seen_doc <- table(factor(clinical.data$seen_doc, levels = c("TRUE",
"FALSE")))
    doctimes <- table(cut(clinical.data$doc_times, c(0, 1, 4, 7, 10,
max(clinical.data$doc_times,
    na.rm = t)), include.lowest = T, right = FALSE))
    steroid <- table(factor(clinical.data[, "oral steroids"]))
    names(steroid) <- c("None", "One", "Two", "Three or more")
    count <- list(insur = insur, intub = intub, miss_work = miss_work, symp =
symp, night = night,
    activity = activity, hosptimes = hosptimes, emergtimes = emergtimes,
seen_doc = seen_doc,
    doctimes = doctimes, steroid = steroid)
    perc <- lapply(count, function(x) round(x/sum(x), 2))
    out <- paste(unlist(count), paste("(", unlist(perc), ")"), sep = ""), sep = "
")
    names(out) <- names(unlist(count))
    out
}
out <- list(Baseline = get_column(cohorts$baseline), Robust =
get_column(cohorts$robust),
    Milestone = get_column(cohorts$milestone))
out <- do.call(cbind, out)
xtable(out)
##### SUPPLEMENTARY TABLE 2
out2 <- list(Baseline = get_column2(cohorts$baseline), Robust =
get_column2(cohorts$robust),
    Milestone = get_column2(cohorts$milestone))
out2 <- do.call(cbind, out2)
xtable(out2)
#####
user.temp <- cohorts$baseline
enrol.date <- character()
get_date = function(x) {
    substring(as.character(x), 1, 10)
}
for (i in 1:length(user.temp)) {
    edate <- unique(c(get_date(His.data[His.data$healthCode == user.temp[i],
]$createdOn),
        get_date(MedH.data[MedH.data$healthCode == user.temp[i], ]$createdOn),
        6agina p

```

```

                                main code.txt
get_date(Med.data[Med.data$healthCode ==
                                user.temp[i], ]$createdOn), get_date(Med.data[Med.data$healthCode ==
user.temp[i], ]$createdOn), get_date(YouA.data[YouA.data$healthCode ==
user.temp[i], ]$createdOn),
                                get_date(Dai.data[Dai.data$healthCode == user.temp[i], ]$date)))
    if (length(edate) > 0) {
        edate = min(as.Date(edate))
    } else {
        edate = NA
    }
    enrol.date[i] = as.character(edate)
    print(i)
}
edate.data <- data.frame(healthCode = cohorts$baseline, edate =
as.Date(enrol.date, format = "%Y-%m-%d"))
mymonth.cut <- as.Date(c("2015-03-09", "2015-04-08", "2015-05-09", "2015-06-09",
"2015-07-10",
"2015-08-10", "2015-09-10"), format = "%Y-%m-%d")
user.set <- cohorts$robust
temp <- edate.data[edate.data$healthCode %in% user.set, ]
robust.enrol <- list()
for (i in 1:6) {
    robust.enrol[[i]] <- as.character(na.omit(temp[temp$edate >= mymonth.cut[i]
& temp$edate <
                                mymonth.cut[i + 1], ]$healthCode))
}
robust.GINA = list(NULL)
for (i in 1:6) {
    robust.GINA[[i]] = factor(GINA.cal(robust.enrol[[i]]), levels =
c("Uncontrolled", "Partly Controlled",
"well Controlled"))
}
robust.GINA.sum = matrix(NA, nrow = 6, ncol = 8)
for (i in 1:6) {
    robust.GINA.sum[i, 1] = length(robust.GINA[[i]])
    robust.GINA.sum[i, 2] = sum(is.na(robust.GINA[[i]]))
    temp = table(robust.GINA[[i]])
    robust.GINA.sum[i, c(3, 5, 7)] = temp
    robust.GINA.sum[i, c(4, 6, 8)] = round(temp/sum(temp), 2)
}
myfunc <- function(v1) {
    deparse(substitute(v1))
}
get_demo.plot <- function(x, mydata) {
    xname <- myfunc(x)
    temp <- data.frame(mydata, x)
    robust.temp = list(NULL)
    for (i in 1:6) {
        robust.temp[[i]] = factor(temp[temp$healthCode %in% robust.enrol[[i]],
xname], levels = levels(x))
    }
    n <- length(levels(x)) * 2 + 2
    robust.temp.sum = matrix(NA, nrow = 6, ncol = n)
    for (i in 1:6) {
        robust.temp.sum[i, 1] = length(robust.temp[[i]])
        robust.temp.sum[i, 2] = sum(is.na(robust.temp[[i]]))
        temp = table(robust.temp[[i]])
        robust.temp.sum[i, (3:n)[3:n%%2 == 1]] = temp
        robust.temp.sum[i, (3:n)[3:n%%2 == 0]] = round(temp/sum(temp), 2)
    }
    c1 <- apply(robust.temp.sum[1:3, (3:n)[3:n%%2 == 1]], 2, sum)
    c2 <- apply(robust.temp.sum[4:6, (3:n)[3:n%%2 == 1]], 2, sum)
    first <- c1/sum(c1)
    last <- c2/sum(c2)
    library("reshape2")
    temp <- cbind(first, last)
    temp.levels <- levels(x)

```

```

                                main code.txt
temp <- data.frame(temp.levels, temp)
melted <- melt(temp, "temp.levels")
names(melted)[3] <- "temp.Dist"
names(melted)[1] <- "name"
names(melted)[2] <- "Period"
list(melted, rbind(c1, c2))
}
user = cohorts$robust
temp1 <- mile.data[mile.data$healthCode %in% user, c("gender", "age",
"healthCode")] # [1] is male [2] is female
temp1$gender = as.character(ifelse(temp1$gender == "[1]", "Male", "Female"))
temp2 <- agesex1[agesex1$healthCode %in% user,
c("NonIdentifiableDemographics.json.patientBiologicalSex",
"NonIdentifiableDemographics.json.patientCurrentAge", "healthCode")]
names(temp2) <- c("gender", "age", "healthCode")
temp2$gender = as.character(temp2$gender)
healthCode = unique(c(as.character(temp1$healthCode),
as.character(temp2$healthCode)))
gender <- factor(unlist(lapply(healthCode, get_y, "gender"))))
mydata <- data.frame(healthCode = gender)
mydata <- mydata[mydata$healthCode %in% cohorts$baseline, ]
gender <- factor(mydata[mydata$healthCode %in% cohorts$baseline, ]$gender)
##### FIGURE 2A
out <- get_demo.plot(gender, mydata)
ggplot(out[[1]], aes(x = Period, y = temp.Dist, fill = name)) + geom_bar(stat =
"identity",
position = "stack")
chisq.test(out[[2]])
gina = factor(GINA.cal(unlist(robust.enrol)))
mydata = data.frame(healthCode = unlist(robust.enrol), gina)
out <- get_demo.plot(gina, mydata)
ggplot(out[[1]], aes(x = Period, y = temp.Dist, fill = name)) + geom_bar(stat =
"identity",
position = "stack")
chisq.test(out[[2]])
##### FIGURE 2B
clinical.data <- His.data[, c("healthCode", "times_hospitalized", "emergency",
"intubated",
"symptoms", "nights", "limited activity")]
activity <- factor(clinical.data[, "limited activity"])
out <- get_demo.plot(activity, clinical.data)
ggplot(out[[1]], aes(x = Period, y = temp.Dist, fill = name)) + geom_bar(stat =
"identity",
position = "stack")
chisq.test(out[[2]], simulate.p.value = T)
##### FIGURE 2C
symptoms <- factor(clinical.data$symptoms)
out <- get_demo.plot(symptoms, clinical.data)
ggplot(out[[1]], aes(x = Period, y = temp.Dist, fill = name)) + geom_bar(stat =
"identity",
position = "stack")
chisq.test(out[[2]], simulate.p.value = T)
##### FIGURE 2D,E See:
userretentionfigures_manuscript-jan2017.html SUPPLEMENTARY TABLE 3B
rob.date = edate.data[match(cohorts$robust, edate.data$healthCode), ]$edate
seq.day = seq(from = as.Date(range(rob.date)[1]), to =
as.Date(range(rob.date)[2]), by = "day")
rob.ind = match(rob.date, seq.day)
rob.gina = GINA.cal(cohorts$robust)
cfl = rob.gina
y = ifelse(cfl == "Uncontrolled", 1, 0)
x = rob.ind
fit = glm(factor(y) ~ x, family = binomial())
summary(fit)
confint(fit, parm = "x")
exp(coef(fit)["x"])
exp(confint(fit, parm = "x"))
cdplot(factor(y) ~ x, ylab = "well Controlled", xlab = "Enrollment Date")
8agina p

```

```

                                main code.txt
rob.date = edate.data[match(cohorts$robust, edate.data$healthCode), ]$edate
seq.day = seq(from = as.Date(range(rob.date)[1]), to =
as.Date(range(rob.date)[2]), by = "day")
rob.ind = match(rob.date, seq.day)
rob.gina = GINA.cal(cohorts$robust)
cf2 = gender[match(cohorts$robust, healthCode)]
x = rob.ind[!is.na(cf2)]
y = ifelse(cf2[!is.na(cf2)] == "Female", 1, 0)
fit = glm(factor(y) ~ x, family = binomial())
summary(fit)
confint(fit, parm = "x")
exp(coef(fit)["x"])
exp(confint(fit, parm = "x"))
cdplot(factor(y) ~ x, ylab = "Gender", xlab = "Enrollment Date")
### use negative version opposite coding
cf3 = clinical.data[match(cohorts$robust, clinical.data$healthCode), "limited
activity"]
cf3 = as.numeric(factor(cf3))
y = cf3[!is.na(cf3)]
x = rob.ind[!is.na(cf3)]
fit = lm(y ~ x)
summary(fit)
length(y)
coef(fit)["x"]
confint(fit, parm = "x")
#####
cf4 = clinical.data[match(cohorts$robust, clinical.data$healthCode), "symptoms"]
cf4 = as.numeric(factor(cf4))
y = cf4[!is.na(cf4)]
x = rob.ind[!is.na(cf4)]
fit = lm(y ~ x)
length(y)
summary(fit)
coef(fit)["x"]
confint(fit, parm = "x")
#####
temp = Dai.data[Dai.data$healthCode %in% cohorts$robust, ]
day <- unlist(tapply(factor(temp$day_symptoms, levels = c("True", "False")),
factor(temp$healthCode),
function(x) table(x)[[1]]/sum(table(x))))
sum(tapply(day, factor(GINA.cal(names(day))), length))
kruskal.test(day, factor(GINA.cal(names(day))))
boxplot(as.numeric(day) ~ factor(GINA.cal(names(day)), levels =
c("Uncontrolled", "Partly Controlled",
"well Controlled")))
#####
night <- unlist(tapply(factor(temp$night_symptoms, levels = c("True", "False")),
factor(temp$healthCode),
function(x) table(x)[[1]]/sum(table(x))))
sum(tapply(night, factor(GINA.cal(names(night))), length))
kruskal.test(night, factor(GINA.cal(names(night))))
boxplot(as.numeric(night) ~ factor(GINA.cal(names(night)), levels =
c("Uncontrolled", "Partly Controlled",
"well Controlled")))
#####
qr.use <- unlist(tapply(factor(temp$use_qr, levels = c("True", "False")),
factor(temp$healthCode),
function(x) table(x)[[1]]/sum(table(x))))
table(is.na(qr.use))
sum(tapply(qr.use, factor(GINA.cal(names(qr.use))), length))
kruskal.test(qr.use, factor(GINA.cal(names(qr.use))))
boxplot(as.numeric(qr.use) ~ factor(GINA.cal(names(qr.use)), levels =
c("Uncontrolled", "Partly Controlled",
"well Controlled")))
#####
medi <- read.table("Daily_medicine_ans_by_date_last_response.tsv", sep = "\t",
header = T)
medi <- medi[medi$healthCode %in% cohorts$robust, ]

```

```

main code.txt
medi$medicine <- factor(medi$medicine, levels = c(1, 2, 3))
Dai.data$medicine = medi[match(Dai.data$healthCode, medi$healthCode), ]$medicine
comp2 <- unlist(tapply(factor(medi$medicine), factor(medi$healthCode),
function(x) (table(x)[[1]] +
table(x)[[2]])/sum(table(x))))
sum(tapply(comp2, factor(GINA.cal(names(comp2))), length))
kruskal.test(comp2, factor(GINA.cal(names(comp2))))
boxplot(as.numeric(comp2) ~ factor(GINA.cal(names(comp2))), levels =
c("Uncontrolled", "Partly Controlled",
"well Controlled"))
##### FIGURE 3E
dai.peakflow = unlist(tapply(Dai.data$peakflow, Dai.data$healthCode, mean, na.rm
= T))
# cond=unlist(tapply(Dai.data$peakflow,Dai.data$healthCode,length))>2
nicole = data.frame(healthCode = cohorts$robust, GINA =
GINA.cal(cohorts$robust), BiologicalSex = agesex1[match(cohorts$robust,
agesex1$healthCode),
]$NonIdentifiableDemographics.json.patientBiologicalSex, HeightInches =
agesex1[match(cohorts$robust,
agesex1$healthCode), ]$NonIdentifiableDemographics.json.patientHeightInches,
WeightPounds = agesex1[match(cohorts$robust,
agesex1$healthCode), ]$NonIdentifiableDemographics.json.patientWeightPounds,
CurrentAge = agesex1[match(cohorts$robust,
agesex1$healthCode), ]$NonIdentifiableDemographics.json.patientCurrentAge,
peakflow = unlist(dai.peakflow[match(cohorts$robust,
names(dai.peakflow))]))
table(is.na(nicole$peakflow))
# # load('from_marcus_091316/robust_users_data.Rdata') pfs_gina <-
# robust_users_daily_summary; rm(robust_users_daily_summary) pfs_gina <-
pfs_gina[,
# !names(pfs_gina) %in% 'GINA'] colnames(pfs_gina)[2] <- 'GINA'
# marcus=pfs_gina[,match(names(nicole),names(pfs_gina))]
pfs_gina = nicole
# Clean bad data out
pfs_gina$HeightInches[ifelse(is.na(pfs_gina$HeightInches < 50), FALSE,
pfs_gina$HeightInches <
50)] <- NA
pfs_gina$HeightInches[ifelse(is.na(pfs_gina$HeightInches > 150), FALSE,
pfs_gina$HeightInches >
150)] <- NA
pfs_gina$WeightPounds[ifelse(is.na(pfs_gina$WeightPounds < 50), FALSE,
pfs_gina$WeightPounds <
50)] <- NA
pfs_gina$WeightPounds[ifelse(is.na(pfs_gina$WeightPounds > 350), FALSE,
pfs_gina$WeightPounds >
350)] <- NA
pfs_gina$peakflow[ifelse(is.na(pfs_gina$peakflow < 60), FALSE, pfs_gina$peakflow
< 60)] <- NA
pfs_gina$peakflow[ifelse(is.na(pfs_gina$peakflow > 900), FALSE,
pfs_gina$peakflow > 900)] <- NA
pfs_gina[complete.cases(pfs_gina[, c("GINA", "peakflow", "WeightPounds",
"BiologicalSex",
"HeightInches", "CurrentAge"))], "healthCode"]
write.table(pfs_gina[complete.cases(pfs_gina[, c("GINA", "peakflow",
"WeightPounds", "BiologicalSex",
"HeightInches", "CurrentAge"))], "healthCode"], file =
"healthCodes4PEFregression.tsv",
quote = FALSE, sep = "\t", row.names = FALSE, col.names = FALSE)
## ! constrain to only uncontrolled and partially controlled
pfs_gina <- pfs_gina[pfs_gina$GINA != "well Controlled", ]
fit <- lm(peakflow ~ GINA + WeightPounds + BiologicalSex + HeightInches +
CurrentAge, data = pfs_gina)
summary(fit)
fit <- lm(peakflow ~ GINA + BiologicalSex + HeightInches, data = pfs_gina)
summary(fit)
table(apply(pfs_gina[, c(2, 3, 4, 7)], 1, function(x) sum(is.na(x)) == 0))
## Generate smoothed curve for Peak Flow versus Height w/in stratified groups
range.height = range(pfs_gina$HeightInches, na.rm = T)
10agina p

```

```

                                main code.txt
sim = data.frame(matrix(ncol = 4, nrow = (range.height[2] - range.height[1] + 1)
* 2 * 2 *
50))
colnames(sim) <- c("Height", "Sex", "GINA", "predicted_PF")
sim$Height = rep(seq(from = range.height[1], to = range.height[2], by = 1), 4 *
50)
sim$Sex = rep(c(0, 1), (range.height[2] - range.height[1] + 1) * 50)
sim$GINA = c(rep(0, (range.height[2] - range.height[1] + 1) * 2 * 50), rep(1,
(range.height[2] -
range.height[1] + 1) * 2 * 50))
# sim <- transform(sim, predicted_PF = 86.48 - fit$coeff[[2]] * GINA +
fit$coeff[[3]]*Sex
# + fit$coeff[[4]]* Height)
sim <- transform(sim, predicted_PF = fit$coeff[[1]] - fit$coeff[[2]] * GINA +
fit$coeff[[3]] *
Sex + fit$coeff[[4]] * Height)
sim$GINA = ifelse(sim$GINA == 0, "Uncontrolled", "Partly Controlled")
sim$Sex = ifelse(sim$Sex == 0, "Female", "Male")
# pf_data=pfs_gina[!is.na(pfs_gina$peakflow), c(1:2, 6, 8:9)]
pf_data = pfs_gina[!is.na(pfs_gina$peakflow), ]
pf_data <- pf_data[complete.cases(pf_data), ]
# colnames(pf_data)[4] <- 'Sex'
colnames(pf_data)[3] <- "Sex"
ggplot(sim, aes(x = Height, y = predicted_PF, colour = GINA, linetype = Sex)) +
geom_line(size = 1.25) +
geom_point(data = pf_data, aes(x = HeightInches, y = peakflow, colour =
GINA, linetype = Sex,
shape = Sex)) + labs(x = "Height (inches)", y = "Predicted Peak Flow
(L/min)") +
theme_minimal()
##### SUPPLEMENTARY FIGURE 4
user.temp <- cohorts$robust
my.puff = Dai.data[, "quick_relief_puffs"]
my.puff[is.na(my.puff)] = 0
my.puff[my.puff > 0 & my.puff < 4] = 1
my.puff[my.puff > 3] = 2
my.qr = as.character(Dai.data$use_qr)
my.qr[my.qr %in% "True"] = 1
my.qr[my.qr %in% "False"] = 0
my.qr <- as.numeric(my.qr)
try0 <- tapply(my.qr, factor(Dai.data$healthCode, levels = user.temp),
as.numeric)
try1 <- tapply(my.puff, factor(Dai.data$healthCode, levels = user.temp),
as.numeric) ###puff
day <- as.character(Dai.data$day_symptoms)
day[day %in% "True"] = 1
day[day %in% "False"] = 0
day <- as.numeric(day)
try2 <- tapply(day, factor(Dai.data$healthCode, levels = user.temp), as.numeric)
###day
night <- as.character(Dai.data$night_symptoms)
night[night %in% "True"] = 1
night[night %in% "False"] = 0
night <- as.numeric(night)
try3 <- tapply(night, factor(Dai.data$healthCode, levels = user.temp),
as.numeric) ###night
## try1 puff try2 dat try3 night try4 peakflow remove outlier
hist(Dai.data[Dai.data$peakflow < 900 & Dai.data$peakflow > 60, ]$peakflow)
peakflow = Dai.data$peakflow
peakflow[peakflow > 900 | peakflow < 60] = NA
try4 <- tapply(peakflow, factor(Dai.data$healthCode, levels = user.temp),
as.numeric)
pdf("peakflow_hist.pdf")
hist(unlist(lapply(try4, sd)), main = "histogram of variance of user peakflow")
dev.off()
# try4[which(unlist(lapply(try4,sd))>100)]<-NA require more than 10 observations
get_cor <- function(x, y) {
if (length(x) > 1) {

```

main code.txt

```

temp <- cbind(x, y)
cond <- apply(temp, 1, function(x) sum(is.na(x))) == 0
n <- sum(cond)
if (n > 10 & sum(x[cond] == y[cond]) != length(x[cond]) & sd(x[cond]) >
0 & sd(y[cond]) >
0) {
  mycor <- cor(x[cond], y[cond])
  list(mycor = mycor, n = n)
}
}
pdf("histogram_correlation.pdf", height = 10, width = 20)
par(mfrow = c(4, 4))
layout(matrix(1:16, 4, 4, byrow = TRUE))
plot.new()
out <- mapply(get_cor, try4, try2) #peakflow versus day
## try1 puff try2 day try3 night try4 peakflow
mycor <- unlist(lapply(out, function(x) x$mycor))
mycor <- mycor[!is.na(mycor)]
nusers <- length(mycor)
n <- unlist(lapply(out, function(x) as.integer(x$n)))[!is.na(mycor)]
nobs <- mean(n)
mean = mean(mycor)
sd <- sd(mycor)
mysub <- paste("n: ", as.integer(nusers), "; mean # observations: ",
round(nobs, 1), "; mean: ",
round(mean, 2), "; sd:", round(sd, 2), sep = "")
hist(unlist(mycor), xlab = "cor", main = "peakflow vs. day", col = "black",
breaks = 10,
sub = mysub, xlim = c(-1, 1))
abline(v = mean(unlist(mycor), na.rm = T), col = "red")
out <- mapply(get_cor, try4, try3) #peakflow versus night
mycor <- unlist(lapply(out, function(x) x$mycor))
mycor <- mycor[!is.na(mycor)]
nusers <- length(mycor)
n <- unlist(lapply(out, function(x) as.integer(x$n)))[!is.na(mycor)]
nobs <- mean(n)
mean = mean(mycor)
sd <- sd(mycor)
mysub <- paste("n: ", as.integer(nusers), "; mean # observations: ",
round(nobs, 1), "; mean: ",
round(mean, 2), "; sd:", round(sd, 2), sep = "")
hist(unlist(mycor), col = "black", breaks = 10, xlab = "cor", main = "peakflow
vs. night",
sub = mysub, xlim = c(-1, 1))
abline(v = mean(unlist(mycor), na.rm = T), col = "red")
out <- mapply(get_cor, try4, try0) #peakflow versus puff
mycor <- unlist(lapply(out, function(x) x$mycor))
mycor <- unlist(lapply(out, function(x) x$mycor))
mycor <- mycor[!is.na(mycor)]
nusers <- length(mycor)
n <- unlist(lapply(out, function(x) as.integer(x$n)))[!is.na(mycor)]
nobs <- mean(n)
mean = mean(mycor)
sd <- sd(mycor)
mysub <- paste("n: ", as.integer(nusers), "; mean # observations: ",
round(nobs, 1), "; mean: ",
round(mean, 2), "; sd:", round(sd, 2), sep = "")
hist(unlist(mycor), col = "black", breaks = 10, xlab = "cor", main = "peakflow
vs. puff",
sub = mysub, xlim = c(-1, 1))
abline(v = mean(unlist(mycor), na.rm = T), col = "red")
plot.new()
plot.new()
out <- mapply(get_cor, try2, try3) #day vs night
mycor <- unlist(lapply(out, function(x) x$mycor))
mycor <- mycor[!is.na(mycor)]
nusers <- length(mycor)

```

```

                                main code.txt
n <- unlist(lapply(out, function(x) as.integer(x$n)))[!is.na(mycor)]
nobs <- mean(n)
mean = mean(mycor)
sd <- sd(mycor)
mysub <- paste("n: ", as.integer(nusers), "; mean # observations: ",
round(nobs, 1), "; mean: ",
round(mean, 2), "; sd:", round(sd, 2), sep = "")
hist(unlist(mycor), xlab = "cor", col = "black", breaks = 10, main = " day vs.
night", sub = mysub,
xlim = c(-1, 1))
abline(v = mean(unlist(mycor), na.rm = T), col = "red")
out <- mapply(get_cor, try2, try0) #day versus puff
mycor <- unlist(lapply(out, function(x) x$mycor))
mycor <- unlist(lapply(out, function(x) x$mycor))
mycor <- mycor[!is.na(mycor)]
nusers <- length(mycor)
n <- unlist(lapply(out, function(x) as.integer(x$n)))[!is.na(mycor)]
nobs <- mean(n)
mean = mean(mycor)
sd <- sd(mycor)
mysub <- paste("n: ", as.integer(nusers), "; mean # observations: ",
round(nobs, 1), "; mean: ",
round(mean, 2), "; sd:", round(sd, 2), sep = "")
hist(unlist(mycor), xlab = "cor", col = "black", breaks = 10, main = "day vs.
puff", sub = mysub,
xlim = c(-1, 1))
abline(v = mean(unlist(mycor), na.rm = T), col = "red")
plot.new()
plot.new()
plot.new()
out <- mapply(get_cor, try0, try3) #puff versu night
mycor <- unlist(lapply(out, function(x) x$mycor))
mycor <- mycor[!is.na(mycor)]
nusers <- length(mycor)
n <- unlist(lapply(out, function(x) as.integer(x$n)))[!is.na(mycor)]
nobs <- mean(n)
mean = mean(mycor)
sd <- sd(mycor)
mysub <- paste("n: ", as.integer(nusers), "; mean # observations: ",
round(nobs, 1), "; mean: ",
round(mean, 2), "; sd:", round(sd, 2), sep = "")
hist(unlist(mycor), xlab = "cor", col = "black", breaks = 10, main = "puff vs.
night", sub = mysub,
xlim = c(-1, 1))
abline(v = mean(unlist(mycor), na.rm = T), col = "red")
dev.off()
##### FIGURE 4 see:
triggeranalysis.R FIGURE 5A
WE.day.index = unlist(lapply(week.data[, "date"], my.convert.day))
user.set <- cohorts$robust
Last.week = matrix(NA, nrow = length(user.set), ncol = ncol(week.data) + 1)
First.week = matrix(NA, nrow = length(user.set), ncol = ncol(week.data) + 1)
for (i in 1:length(user.set)) {
  cur.user = user.set[i]
  pick = which(week.data[, "healthCode"] == cur.user)
  if (length(pick) > 0) {
    if (length(pick) == 1) {
      Last.week[i, ] = c(WE.day.index[pick], as.matrix(week.data)[pick, ])
      First.week[i, ] = Last.week[i, ]
    } else {
      temp = week.data[pick, ]
      temp1 = WE.day.index[pick]
      Last.week[i, ] = c(max(temp1), as.matrix(temp)[which.max(temp1)[1],
])
      First.week[i, ] = c(min(temp1), as.matrix(temp)[which.min(temp1)[1],
])
    }
  }
}

```

main code.txt

```

}
colnames>Last.week) = c("Day", colnames(week.data))
colnames(First.week) = c("Day", colnames(week.data))
Last.w.day = as.numeric>Last.week[, 1])
First.w.day = as.numeric(First.week[, 1])
Len.w = Last.w.day - First.w.day
Len.w[is.na(Len.w)] = 0
hist(Len.w[Len.w > 0], xlab = "# of days between the first and last weekly
survey", main = "")
a = 90
Last.week.3m = Last.week[Len.w > a, ]
First.week.3m = First.week[Len.w > a, ]
user.3m = user.set[Len.w > a]
col.n = c("asthma_doc_visit", "emergency_room", "prednisone", "limitations",
"missed_work",
"admission")
p.list = rep(NA, length(col.n))
for (k in 1:length(col.n)) {
  cur.n = col.n[k]
  for (i in 1:nrow>Last.week)) {
    cur = Last.week[i, cur.n]
    if (!is.na(cur)) {
      Last.week[i, cur.n] = gsub("\\ ", "", cur)
    }
    cur = First.week[i, cur.n]
    if (!is.na(cur)) {
      First.week[i, cur.n] = gsub("\\ ", "", cur)
    }
  }
  x.v = Last.week[, cur.n] == "True"
  x.v = x.v + 0
  y.v = First.week[, cur.n] == "True"
  y.v = y.v + 0
  temp = wilcox.test(x.v, y.v, paired = TRUE)
  p.list[k] = temp$p.value
}
round(p.list, 3)
length(user.3m)
cur.n = "limitations"
table>Last.week = factor>Last.week.3m[, cur.n], levels = c("True", "False")),
First.week = factor(First.week.3m[,
cur.n], levels = c("True", "False")))
wilcox.test(as.numeric(factor>Last.week.3m[, cur.n], levels = c("True",
"False"))), as.numeric(factor(First.week.3m[,
cur.n], levels = c("True", "False"))), paired = T)
table>Last.week = factor>Last.week[, cur.n], levels = c("True", "False")),
First.week = factor(First.week[,
cur.n], levels = c("True", "False")))
wilcox.test(as.numeric(factor>Last.week[, cur.n], levels = c("True", "False"))),
as.numeric(factor(First.week[,
cur.n], levels = c("True", "False"))), paired = T)

##### FIGURE 5B
GINA.m = matrix(NA, nrow = nrow(mile.data), ncol = 4)
colnames(GINA.m) = c("DaySym", "NighSym", "Reliever", "Activity")
for (i in 1:nrow(mile.data)) {
  GINA.m[i, 1] = is.element(mile.data[i, "symptoms2"], c("[1]", "[2]", "[3]"))
+ 0 ### day sym more than twice a week
  GINA.m[i, 2] = !is.element(mile.data[i, "nights2"], c("", "[5]", "[6]")) + 0
### any night sym
  GINA.m[i, 3] = is.element(mile.data[i, "past_month_quick_relief2"], c("[1]",
"[2]", "[3]")) +
0 ### more than twice a week
  GINA.m[i, 4] = !is.element(mile.data[i, "limited.activity2"], c("", "[5]",
"[6]")) +
0 ### any limited activity
}
GINA.v = apply(GINA.m, 1, sum)

```

main code.txt

```
GINA.indi = GINA.v
GINA.indi[GINA.v > 2] = "Uncontrolled"
GINA.indi[GINA.v > 0 & GINA.v < 3] = "Partly Controlled"
GINA.indi[GINA.v == 0] = "Well Controlled"
table(GINA.indi)
Mile.GINA = data.frame(healthCode = mile.data[, "healthCode"], GINA.indi)
get_cont.table = function(user.temp) {
  table(GINA.cal(user.temp), Mile.GINA[match(user.temp, Mile.GINA$healthCode),
]$GINA.indi)
}
get_cont.table(cohorts$milestone)
get_cont.table(cohorts$robust)
get_cont.table(cohorts$baseline)
##### FIGURE 5C
table(Mile.data[, 'alleviate_troubles'])
table(Mile.data[, "met_goal"])
table(Feed.data[Feed.data$healthCode %in% cohorts$milestone, "manage_asthma"])
###168 users
##### SUPPLEMENTARY TABLE 5d,e
[1], [2], [3] Yes, no, don't know
a = table(factor(GINA.cal(cohorts$milestone), levels = c("Well Controlled",
"Partly Controlled",
"Uncontrolled")), Mile.data[match(cohorts$milestone, Mile.data$healthCode),
"prevent_ed"])
a
a/apply(a, 1, sum)
a = table(factor(GINA.cal(cohorts$milestone), levels = c("Well Controlled",
"Partly Controlled",
"Uncontrolled")), Mile.data[match(cohorts$milestone, Mile.data$healthCode),
"prevent_visit"])
a
a/apply(a, 1, sum)
##### SUPPLEMENTARY TABLE 3A
edate.data <- data.frame(healthCode = cohorts$baseline, edate =
as.Date(enrol.date, format = "%Y-%m-%d"))
mymonth.cut <- as.Date(c("2015-03-09", "2015-04-08", "2015-05-09", "2015-06-09",
"2015-07-10",
"2015-08-10", "2015-09-10"), format = "%Y-%m-%d")
ecount.daily = table(cut(as.Date(Dai.data$date), mymonth.cut))
ecount.weekly = table(cut(as.Date(week.data$date), mymonth.cut))
get_ecount = function(x) {
  table((cut(as.Date(edate.data[edate.data$healthCode %in% x, ]$edate),
mymonth.cut)))
}
ecount.cohort = do.call(cbind, lapply(cohorts, get_ecount))
#####
enrol.m = enrollment[, 2:5]
enrol.m = as.matrix(enrol.m)
enrol.m = matrix(as.numeric(as.vector(enrol.m)), nrow = nrow(enrol.m))
enrol.s = matrix(NA, nrow = 6, ncol = 2)
colnames(enrol.s) = c("Downloads", "Enrolled")
month.cut = 1:6
month.cut[1] = 0
month.cut[2] = my.convert.day("2015-04-08")
month.cut[3] = my.convert.day("2015-05-09")
month.cut[4] = my.convert.day("2015-06-09")
month.cut[5] = my.convert.day("2015-07-10")
month.cut[6] = my.convert.day("2015-08-10")
month.cut[7] = my.convert.day("2015-09-10")
for (i in 1:6) {
  pick = enrol.m[, 1] == i
  enrol.s[i, 1] = sum(enrol.m[pick, 4], na.rm = T)
  temp = max(enrol.m[pick, 3], na.rm = T)
  temp1 = 0
  if (i > 1) {
    temp1 = max(enrol.m[enrol.m[, 1] == (i - 1), 3], na.rm = T)
  }
  enrol.s[i, 2] = temp - temp1
}
```

main code.txt

```
}
enrol.s1 = cbind(downloads = enrol.s[, 1], daily = ecount.daily, daily.dist =
round(ecount.daily/sum(ecount.daily),
  2), weekly = ecount.weekly, weekly.dist =
round(ecount.weekly/sum(ecount.weekly), 2),
  enrolled = enrol.s[, 2], enrol.perc = round(enrol.s[, 2]/enrol.s[, 1], 2),
baseline = ecount.cohort[,
  1], baseline.perc = round(ecount.cohort[, 1]/enrol.s[, 1], 2), robust =
ecount.cohort[,
  2], robust.perc = round(ecount.cohort[, 2]/enrol.s[, 1], 2), milestone =
ecount.cohort[,
  3], milestone.perc = round(ecount.cohort[, 3]/enrol.s[, 1], 2))
```

trigger analysis.txt

```
##### LOAD LIBRARIES
library(cluster)
library(colorspace)
library(reshape2)
library(gplots)
library(ggplot2)
library(maps)
library(mapproj)
library(caTools)
library(TTR)
library(fields)
library(parallel)
library(maptools)
library(RColorBrewer)
library(classInt)
library(ggmap)
library(zipcode)
data(zipcode)
##### LOAD DATA
longlat.data <- read.csv("longlat.txt", header = TRUE) ###this file is NOT
provided to protect user location information
pollen.data <- read.csv("pollen_month.csv") #source: http://www.pollen.com
wild.data <- read.csv("wildfire_longlat.csv", header = T) #source:
http://www3.epa.gov/airdata/ad_data.html
WA <- read.csv("waairquality12_24.csv") #source:
http://www3.epa.gov/airdata/ad_data.html
gaz <- read.table("2014_Gaz_zcta_national.txt", header = TRUE, colClasses =
c("character",
  "NULL", "NULL", "NULL", "NULL", "numeric", "numeric"))
##source:https://www.census.gov/geo/maps-data/data/gazetteer2014.html
gaz.place <- read.delim("2014_Gaz_place_national.txt", header = TRUE, sep =
"\t", colClasses = c("character",
  "character", "NULL", "character", "NULL", "NULL", "NULL", "NULL", "NULL",
  "NULL", "numeric",
  "numeric"))
##source:https://www.census.gov/geo/maps-data/data/gazetteer2014.html
temp.loc <- read.table("temp_loc.txt", header = TRUE) ###source:
https://www.ncdc.noaa.gov/data-access
zip <- readShapePoly("cb_2014_us_zcta510_500k.shp") ##### source:
https://www.census.gov/geo/maps-data/data/cbf/cbf_zcta.html
noaa_tmax.data <- read.table("noaa_tmax.txt", header = TRUE, sep = "\t",
check.names = FALSE) ###source: https://www.ncdc.noaa.gov/data-access
aqi <- read.table("aqi_pm2pt5.txt", header = FALSE) #source:
http://www3.epa.gov/airdata/ad_data.html
##### DEFINE FUNCTIONS
get_temp <- function(y, user.temp) {
  n.clust <- length(levels(factor(air.loc[, y])))
  cond <- unlist(lapply(lapply(tapply(air.loc[, y], air.loc$healthCode,
na.omit), unique),
length) == 1)
  temp <- unlist(lapply(tapply(air.loc[, y], air.loc$healthCode, na.omit),
unique)[cond])
  temp <- data.frame(healthCode = names(temp), clust.ext = as.character(temp))
  temp.data <- merge(air.loc, temp, by = "healthCode", all.x = T)
  temp.data$clust.ext <- factor(temp.data$clust.ext)
  temp.data[temp.data$healthCode %in% user.temp, ]
}
get_triggers <- function(x) {
  if (length(x) > 0) {
    x <- x[!is.na(x)]
    out <- gsub("\\\\[", "", unlist(strsplit(gsub(",", "]["), paste(x, collapse
= ", "))),
      "\\]"))
    out[out == "other"] <- "NA"
    out[out == ""] <- "NA"

    out <- factor(out[out != ""], levels = trig.lev0)
    out
  }
}
```

```

                                trigger analysis.txt
    } else {
      out <- factor(NA, levels = trig.lev1)
      out
    }
  }
get_top.trig <- function(x) {
  x <- factor(x, levels = trig.lev1)
  n <- sum(table(x))
  if (n > 10) {
    top.trig <- names(table(x))[which.max(table(x))]
  } else {
    NA
  }
}
get_out2 <- function(x, n, filter) {
  if (n >= filter) {
    out2 <- table(factor(x, levels = trig.lev1))/sum(table(x))
  } else {
    out2 <- rep(NA, length(trig.lev1))
  }
  out2
}
get_dist <- function(x, y, mycut.week, filter) {
  out <- tapply(x, cut(as.Date(y), as.Date(mycut.week), include.lowest = T,
right = T),
    as.character)
  N <- lapply(out, length)
  out1 <- lapply(out, get_triggers)
  out2 <- t(mapply(get_out2, out1, N, filter))
  out3 <- out2
  row.names(out3) <- names(out)
  list(out3 = out3, N = N)
}
get_out.table <- function(x, n) {
  if (n >= 10) {
    out2 <- table(factor(x, levels = c("True", "False")))/sum(table(x))
  } else {
    out2 <- rep(NA, 2)
  }
  out2
}
get_dist2 <- function(x, y, mycut.season) {
  out <- tapply(x, cut(as.Date(y), as.Date(mycut.season), include.lowest = T,
right = T),
    as.character)
  N <- lapply(out, length) #####number of individuals
  out2 <- t(mapply(get_out.table, out, N))
  out3 <- out2
  row.names(out3) <- names(out)
  list(out3 = out3, N = N)
}
my.bbands = function(x, n, sd) {
  mavg = runmean(x, n)
  sdev = runsd(x, n)
  up <- mavg + sd * sdev
  dn <- mavg - sd * sdev
  res <- cbind(dn, mavg, up)
  colnames(res) <- c("dn", "mavg", "up")
  res
}
get_station <- function(x) {
  ### long then lat
  distance <- rdist.earth(station.temp[, c(3, 2)], t(matrix(c(x$longitude,
x$latitude))))
  station <- as.character(station.temp[which.min(distance), ]$station)
  distance <- min(distance)
  list(station = station, distance = distance)
}

```

```

                                trigger analysis.txt
get_rad <- function(x, y) {
  as.numeric(rdist.earth(t(matrix(c(x[1], y[2]))), t(matrix(c(x[1], y[1])))))
}
clean_x <- function(x) {
  ifelse(is.null(x), NA, x)
}
get_zip.data <- function(x) {
  dist1 <- rdist.earth(gaz[, c(3, 2)], t(matrix(c(x$longitude, x$latitude))))
  dist2 <- rdist.earth(gaz.place[, c(5, 4)], t(matrix(c(x$longitude,
x$latitude))))
  pick1 <- which.min(dist1)
  pick2 <- which.min(dist2)
  geoid <- as.character(gaz[pick1, ]$GEOID)
  zip.distance <- dist1[pick1]
  state <- as.character(gaz.place[pick2, ]$USPS)
  name <- as.character(gaz.place[pick2, ]$NAME)
  place.distance <- dist2[pick2]
  list(zip.dist = zip.distance, geoid = geoid, state = state, place = name,
place.dist = place.distance)
}
get_weather.data <- function(x, y) {
  station.date <- tapply(x$date, x$station, as.Date, format = "%Y%m%d")
  station.temp <- tapply(x[, y], x$station, as.numeric)
  mapply(function(x, y) {
    temp = y[order(x)]
    names(temp) = x[order(x)]
    return(temp)
  }, station.date, station.temp)
}
latlong2state <- function(pointsDF) {
  # Prepare SpatialPolygons object with one SpatialPolygon per state (plus DC,
minus HI &
# AK)
  states <- map("state", fill = TRUE, col = "transparent", plot = FALSE)
  IDs <- sapply(strsplit(states$names, ":"), function(x) x[1])
  states_sp <- map2SpatialPolygons(states, IDs = IDs, proj4string =
CRS("+proj=longlat +datum=wgs84"))
  # Convert pointsDF to a SpatialPoints object
  pointsSP <- SpatialPoints(pointsDF, proj4string = CRS("+proj=longlat
+datum=wgs84"))
  # Use 'over' to get _indices_ of the Polygons object containing each point
  indices <- over(pointsSP, states_sp)
  # Return the state names of the Polygons object containing each point
  stateNames <- sapply(states_sp@polygons, function(x) x@ID)
  stateNames[indices]
}
get_zip.data <- function(x) {
  dist1 <- rdist.earth(gaz[, c(3, 2)], t(matrix(c(x$longitude, x$latitude))))
  dist2 <- rdist.earth(gaz.place[, c(5, 4)], t(matrix(c(x$longitude,
x$latitude))))
  pick1 <- which.min(dist1)
  pick2 <- which.min(dist2)
  geoid <- as.character(gaz[pick1, ]$GEOID)
  zip.distance <- dist1[pick1]
  state <- as.character(gaz.place[pick2, ]$USPS)
  name <- as.character(gaz.place[pick2, ]$NAME)
  place.distance <- dist2[pick2]
  list(zip.dist = zip.distance, geoid = geoid, state = state, place = name,
place.dist = place.distance)
}
get_station <- function(x) {
  distance <- rdist.earth(temp[, c(3, 2)], t(matrix(c(x$longitude,
x$latitude))))
  station <- as.character(temp[which.min(distance), ]$station)
  distance <- min(distance)
  list(station = station, distance = distance)
}
get_troubles <- function(x) {

```

```

                                trigger analysis.txt
if (length(x) > 0) {
  x <- x[!is.na(x)]
  out <- gsub("\\[", "", unlist(strsplit(gsub(",", "[", paste(x, collapse
= ", "))),
    "\\]"))
  out[out == "other"] <- "NA"
  out[out == ""] <- "NA"
  out <- factor(out[out != ""], levels = as.character(1:22))
  out
} else {
  out <- factor(NA, levels = as.character(1:22))
  out
}
}
##### PRE_PROCESSING CODE
zipcode <- subset(zipcode)
names(noaa_tmax) <- c("station", "date", "TMAX")
noaa <- noaa_tmax
temp <- temp.loc[temp.loc$station %in% noaa$station, ]
templ <- apply(longlat.data[, c(2, 3)], 1, as.list)
out <- mclapply(templ, get_station, mc.cores = 4)
station <- unlist(lapply(out, function(x) x[[1]]))
station.distance <- unlist(lapply(out, function(x) x[[2]]))
station.state <- temp.loc[match(station, temp.loc$station), ]$state
json <- gsub("\\.json", "", longlat.data$filename)
out <- data.frame(station, longlat.data[, c("longitude", "latitude")],
  station.state, station.distance,
  json)
### nearest station must be within 50 miles
pdf(file = "histogram_gcd.pdf", width = 10.5, height = 5.5)
par(mfrow = c(1, 2))
hist(out$station.distance, xlab = "Great-circle distance (miles)", main =
"Distance to nearest station")
hist(out[out$station.distance < 50, ]$station.distance, xlab = "Great-circle
distance (miles)",
  main = "Distance to nearest station < 50 miles")
dev.off()
apploc <- out[out$station.distance < 50, ]
##### map longitude and latitude to zip code
temp <- data.frame(longitude = apploc$longitude, latitude = apploc$latitude)
temp <- apply(temp, 1, as.list)
zip.data <- mclapply(temp, get_zip.data, mc.cores = 4)
zip.zcta <- unlist(lapply(zip.data, function(x) x$geoid))
zip.dist <- unlist(lapply(zip.data, function(x) x$zip.dist))
zip.state <- unlist(lapply(zip.data, function(x) x$state))
zip.place <- unlist(lapply(zip.data, function(x) x$place))
out <- data.frame(apploc, zip.dist, zip.state, zip.place, zip.zcta)
par(mfrow = c(1, 2))
hist(out$zip.dist, xlab = "Great-circle distance (miles)", main = "Distance to
nearest zip")
hist(out[out$zip.dist < 50, ]$zip.dist, xlab = "Great-circle distance (miles)",
main = "Distance to nearest zip < 50 miles")
apploc <- out[out$zip.dist < 50, ]
cond <- noaa$date >= "20150309" & noaa$date <= "20150909"
noaa_tmax <- noaa[cond, ]
my.tmax <- get_weather.data(noaa_tmax, "TMAX")
#####
names(aqi) <- c("aqi_json", "aqi_value", "aqi_date")
aqi <- data.frame(aqi)
temp = table.list[[25]]@values
json.data <- temp[, c("healthCode", "latlong.json", "createdOn",
"aqiResponse.json.reports")]
longlat <- paste(apploc$longitude, apploc$latitude, sep = ",")
data1 <- apploc[match(json.data$latlong.json, apploc$json), c("latitude",
"longitude")]
data2 <- aqi[match(json.data$aqiResponse.json.reports, aqi$aqi_json),
c("aqi_value", "aqi_date")]
air.loc <- data.frame(json.data, data1, data2)
4agina p

```

```

                                trigger analysis.txt
date <- as.Date(substring(air.loc$createdOn, 1, 10))
air.loc <- air.loc[date <= "2015-09-09" & date >= "2015-03-09", ]
##### take mean lat long, require variance be smaller than threshold
date <- as.Date(substring(air.loc$createdOn, 1, 10))
merge.a <- factor(paste(air.loc$healthCode, date, sep = " "))
m1 <- data.frame(air.loc, merge.a = factor(merge.a))
#####
long <- tapply(m1$longitude, factor(m1$merge.a), mean, na.rm = T)
long <- lapply(long, clean_x)
range.long <- tapply(m1$longitude, factor(m1$merge.a), function(x) range(x))
lat <- tapply(m1$latitude, factor(m1$merge.a), mean, na.rm = T)
lat <- lapply(lat, clean_x)
range.lat <- tapply(m1$latitude, factor(m1$merge.a), function(x) range(x))
##### max range in lat-long over a single day may not to exceed 5 miles
rad <- mapply(get_rad, range.long, range.lat)
rad <- unlist(lapply(rad, clean_x))
hist(rad[rad < 20])
#####
temp <- data.frame(longitude = unlist(long), latitude = unlist(lat))
temp <- apply(temp, 1, as.list)
zip.data <- mclapply(temp, get_zip.data, mc.cores = 4)
zip.zcta <- lapply(zip.data, function(x) x$geoid)
zip.zcta <- unlist(lapply(zip.zcta, clean_x))
zip.dist <- lapply(zip.data, function(x) x$zip.dist)
zip.dist <- unlist(lapply(zip.dist, clean_x))
zip.state <- lapply(zip.data, function(x) x$state)
zip.state <- unlist(lapply(zip.state, clean_x))
zip.place <- lapply(zip.data, function(x) x$place)
zip.place <- unlist(lapply(zip.place, clean_x))
zip.place.dist <- lapply(zip.data, function(x) x$place.distance)
zip.place.dist <- unlist(lapply(zip.place.dist, clean_x))
#####
names(noaa_tmax) <- c("station", "date", "TMAX")
names(noaa_tmax) <- c("station", "date", "TMAX")
noaa <- noaa_tmax
station.temp <- temp.loc[temp.loc$station %in% noaa$station, ]
station.data <- mclapply(temp, get_station, mc.cores = 4)
station <- lapply(station.data, function(x) x$station)
station <- unlist(lapply(station, clean_x))
station.dist <- lapply(station.data, function(x) x$distance)
station.dist <- unlist(lapply(station.dist, clean_x))
loc_daily <- data.frame(long = unlist(long), lat = unlist(lat), zip.zcta,
                        zip.dist, zip.state,
                        zip.place, zip.place.dist, station.dist, station)
#####
loc_daily <- loc_daily[rad < 20, ]
#####
merge.b <- paste(m1$healthCode, m1$aqi_date, sep = " ")
m1 <- data.frame(m1, merge.b)
b <- tapply(m1$aqi_date, factor(m1$merge.b), as.character)
z <- tapply(m1$aqi_value, factor(m1$merge.b), c)
air.date <- lapply(b, function(x) names(which.max(table(factor(x)))))
air.date[grepl("NULL", air.date)] <- NA
air.date <- as.character(unlist(air.date))
#### take mean across valid dates
air.value <- mapply(function(x, y, q) {
  mean(q[x %in% y], na.rm = T)
}, b, air.date, z)
air.value <- as.numeric(unlist(air.value))
#####
temp <- lapply(lapply(b, unique), length)
disc <- names(temp)[unlist(temp) >= 2]
####
out <- strsplit(names(b), " ")
healthCode <- unlist(lapply(out, function(x) x[1]))
air.date <- data.frame(healthCode = healthCode, pm2pt5.date = air.date,
pm2pt5.value = air.value)
temp <- paste(air.date$healthCode, air.date$pm2pt5.date, sep = " ")

```

trigger analysis.txt

```

table(duplicated(temp))
air_daily <- air.date
##### now merge all
m0 <- Dai.data[, c("healthCode", "date", "get_worse", "day_symptoms")]
merge.0 <- paste(Dai.data$healthCode, Dai.data$date, sep = " ")
m0 <- data.frame(m0, merge.0)
merge.1 <- row.names(loc_daily)
m1 <- data.frame(loc_daily, merge.1)
merge.2 <- paste(air_daily$healthCode, air_daily$pm2pt5.date, sep = " ")
m2 <- data.frame(air_daily, merge.2)
temp <- merge(m0, m1, by.x = "merge.0", by.y = "merge.1", all.x = T)
temp <- merge(temp, m2, by.x = "merge.0", by.y = "merge.2", all.x = T)
temp <- temp[, -1]
names(temp)[1] <- "healthCode"
temp <- data.frame(temp)
map.state <- c()
for (i in 1:nrow(temp)) {
  if (sum(is.na(temp[i, c(5, 6)])) == 0) {
    map.state[i] <- latlong2state(temp[i, c(5, 6)])
  } else {
    map.state[i] <- NA
  }
  print(i)
}
air.loc <- data.frame(temp, map.state)
#####
date.levels <- levels(factor(substring(Dai.data$date, 1, 10)))
date5 <- factor(cut(as.Date(date.levels), 37, labels = FALSE))
mytemp <- my.tmax #8,347 stations
temp <- lapply(mytemp, function(x) as.numeric(x[match(date.levels, names(x))]))
temp <- do.call(rbind, temp)
colnames(temp) <- date.levels
temp <- t(apply(temp, 1, function(x) tapply(x, date5, median, na.rm = T)))
mystations <- intersect(as.character(apploc$station), row.names(temp)) #1318
out.tmax <- apply(temp, 2, function(x) tapply(x, factor(row.names(temp), levels
= mystations),
  mean, na.rm = T))
cond <- apply(out.tmax, 1, function(x) sum(is.na(x)) == 0)
out.tmax <- out.tmax[cond, ] ####we have 4742 if 5day used and #3039 if all day
out <- list()
inames <- row.names(out.tmax)
for (i in 1:length(inames)) {
  # same station maps to different zips
  a <- unique(apploc[grep(inames[i], apploc$station), ]$zip.zcta)
  n <- length(a)
  iout <- matrix(rep(out.tmax[i, ], n), n, 37, byrow = TRUE)
  rownames(iout) <- as.character(a)
  out[[i]] <- iout
}
out <- do.call(rbind, out)
out.tmax <- out
dmat <- daisy(out.tmax)
myclust <- hclust(dmat)
plot(myclust, cex = 0.001, axes = T, sub = "", xlab = "", ylab = "", main =
"Median Maximum Temperature (5-day)",
  las = 2)
clust.1000 <- cutree(myclust, h = 1000)
clust.1400 <- cutree(myclust, h = 1400)
clust.5 <- cutree(myclust, k = 5)
temp <- data.frame(zips = row.names(out.tmax), clust.1400, clust.1000, clust.5)
out <- apply(out.tmax, 2, function(x) tapply(x, temp$clust.5, mean, na.rm = T))
apply(out, 1, mean)
zip.clusters <- temp
##### functions
get_error <- function(n, sd) {
  qnorm(0.975) * sd/sqrt(n)
}
trig.lab <- c("A cold", "Exercise", "More activity", "Strong smells", "Exhaust
6agina p

```

trigger analysis.txt

```
fumes", "House dust",
"Dogs", "Cats", "Other furry animals", "Mold", "Pollen", "Extreme heat",
"Extreme Cold",
"Change in weather", "Period", "Air quality", "Smoking", "Stress",
"Feelings", "Laughter",
"I don't know", "None of these")
date.levels <- levels(factor(substring(Dai.data$date, 1, 10)))
mycols <- rep(NA, 17)
mycols[7] <- "factor"
mycols <- as.list(mycols)
air.loc <- data.frame(air.loc, zip.clusters[match(air.loc$zip.zcta,
zip.clusters$zips), c("clust.1400",
"clust.1000", "clust.5")])
user.temp <- cohorts$robust
temp.zip <- get_temp("zip.zcta", cohorts$robust)
temp.state <- get_temp("zip.state", cohorts$robust)
temp.place <- get_temp("zip.place", cohorts$robust)
temp.clust2 <- get_temp("clust.1400", cohorts$robust)
#####
map("state", interior = FALSE)
map("state", boundary = FALSE, col = "gray", add = TRUE)
# align data with map definitions by (partial) matching state, county names,
which include
# multiple polygons for some counties
my.state.name = state.name
for (i in 1:length(state.name)) {
  name = state.name[i]
  my.state.name[i] = paste(tolower(substr(name, 1, nchar(name))), sep = "")
}
temp = state.abb[match(map("state", plot = FALSE)$names, my.state.name)]
temp[34:37] = "NY"
temp[38:40] = "NC"
temp[53:55] = "VA"
temp[56:60] = "WA"
temp[20:22] = "MA"
temp[23:24] = "MI"
temp[8] = "DC"
mapstate <- temp
temp <- temp.clust2
test <- tapply(temp$zip.state, temp$clust.ext, function(x)
sort(table(factor(x))))
tempo <- lapply(mapstate, function(x) as.numeric(c(test[[1]][x], test[[2]][x])))
colors.match <- rep(NA, length(tempo))
mylabel <- lapply(tempo, which.max)
colors.match[mylabel %in% 1] <- "blue"
colors.match[mylabel %in% 2] <- "red"
colors.match[mylabel %in% c(1, 2)] <- "gray"
##### SUPPLEMENTARY FIGURE
1B
map("state", col = colors.match, fill = TRUE, resolution = 0, lty = 0.5,
projection = "polyconic")
legend("topright", c("North", "South"), fill = c("blue", "red"))
trig.lev0 <- as.character(1:22)
# c('Pollen', 'Change.in.weather', 'Stress', 'Extreme heat', 'More.activity')
alt = c(11, 12, 16)
trig.lev1 = trig.lev0[alt]
trig.lab1 <- trig.lab[alt]
trig.colors1 <- trig.colors[alt]
summer.start <- "2015-06-21"
summer <- temp[as.Date(temp$date) >= summer.start, ]
summer.trigs <- tapply(summer$get_worse, summer$clust.ext, get_triggers)
summer.top.trig <- lapply(summer.trigs, get_top.trig)
spring <- temp[as.Date(temp$date) <= summer.start, ]
spring.trigs <- tapply(spring$get_worse, spring$clust.ext, get_triggers)
spring.top.trig <- lapply(spring.trigs, get_top.trig)
out <- cbind(spring.top.trig, summer.top.trig)
trig <- tapply(temp$get_worse, factor(temp$clust.ext), as.character)
date <- tapply(temp$date, factor(temp$clust.ext), as.character)

7agina p
```

```

                                trigger analysis.txt
mycut.5day <- levels(cut(as.Date(date.levels), 37))
mycut.month <- c("2015-03-09", "2015-04-09", "2015-05-09", "2015-06-09",
"2015-07-09", "2015-08-09",
"2015-09-09")
mycut.season <- c("2015-03-09", summer.start, "2015-09-9")
mycut.week <- levels(cut(as.Date(date.levels), "week"))
out <- list()
nout <- list()
for (i in 1:length(trig)) {
  out[[i]] <- get_dist(trig[[i]], date[[i]], mycut.season, 10)[[1]]
  nout[[i]] <- get_dist(trig[[i]], date[[i]], mycut.season, 10)[[2]]
#####this is the number of individuals
}
spring.out <- lapply(out, function(x) x[1, ])
spring.out <- do.call(rbind, spring.out)
summer.out <- lapply(out, function(x) x[2, ])
summer.out <- do.call(rbind, summer.out)
Season <- rep(c("spring", "summer"), nrow(spring.out))
forplot <- lapply(1:nrow(spring.out), function(x) rbind(spring.out[x, ],
summer.out[x, ]))
forplot <- do.call(rbind, forplot)
trig.colors = c("green", "red", "blue")
row.names(forplot) = rep(c("Spring", "Summer"), 2)
#####FIGURE 4A
barplot(t(forplot), col = trig.colors)
legend("topleft", trig.lab1, fill = trig.colors)
#####
date <- tapply(temp$date, factor(temp$clust.ext), as.Date)
temp <- temp.clust2
mycut <- mycut.5day
trig <- tapply(temp$get_worse, factor(temp$clust.ext), as.character)
date <- tapply(temp$date, factor(temp$clust.ext), as.Date)
trig.out <- list()
trig.nout <- list()
for (i in 1:length(date)) {
  trig.out[[i]] <- get_dist(trig[[i]], date[[i]], mycut, 10)[[1]]
  trig.nout[[i]] <- get_dist(trig[[i]], date[[i]], mycut, 10)[[2]]
}
heat.out <- t(do.call(rbind, lapply(trig.out, function(x) x[, trig.lev1 %in%
"12"])))
colnames(heat.out) <- names(trig)
get_tmax.plot <- function(astation, mycol, data.levels) {
  data <- lapply(mytemp, function(x) as.numeric(x[match(date.levels,
names(x))]))
  data <- do.call(rbind, data)
  data <- data[row.names(data) %in% astation, ]/10
  m <- apply(data, 2, mean, na.rm = T)
  s <- apply(data, 2, sd, na.rm = T)
  e <- get_error(apply(data, 2, sum, na.rm = T), s)
  # plot.mat=rbind(m+e,m,m-e)
  # main.lab<-as.character(apploc[match(azip,apploc$zip.zcta),]$zip.place)
  # plot(smooth.spline(m),ylim=c(0,50),main='',xlab='',ylab='mean maximum temp
  # (celsius)',col=mycol,type='l',lty=2)
  plot.mat = my.bbands(m, n = 4, sd = 1)
  matplot(plot.mat[, 1:3], ylim = c(0, 50), main = "", xlab = "", ylab = "mean
maximum temp (celsius)",
  col = mycol, type = "l", lwd = c(1, 2, 1), lty = c(1, 2, 1), xaxt = "n")
  #
  arrows(1:length(m),as.numeric(m-e),1:length(m),as.numeric(m+e),code=3,length=0.0
5,angle=90)
  out <- 1:nrow(plot.mat)
}
#####FIGURE 4B
par(mar = c(6.1, 6.1, 8.1, 8.1))
temp <- temp.clust2
extrap.zcta <- temp.zip$clust.ext
region.zip <- tapply(temp.clust2$zip.zcta, temp.clust2$clust.ext, function(x)
levels(factor(x)))

```

```

trigger analysis.txt
mymonth <- cut(as.Date(temp.clust2$date), "month", labels = c("Mar", "Apr",
"May", "Jun",
"Jul", "Aug", "Sep"))
mypollen <- mapply(function(x, y) pollen.data[match(x, pollen.data$ZIP),
colnames(pollen.data) %in%
y], extrap.zcta, mymonth)
a <- mymonth[extrap.zcta %in% pollen.data$ZIP & extrap.zcta %in%
region.zip[[1]]]
b <- mypollen[extrap.zcta %in% pollen.data$ZIP & extrap.zcta %in%
region.zip[[1]]]
pollen.north <- tapply(b, a, mean, na.rm = T)
a <- mymonth[extrap.zcta %in% pollen.data$ZIP & extrap.zcta %in%
region.zip[[2]]]
b <- mypollen[extrap.zcta %in% pollen.data$ZIP & extrap.zcta %in%
region.zip[[2]]]
pollen.south <- tapply(b, a, mean, na.rm = T)
clust.col <- c("blue", "red", "orange", "green", "purple")
check <- pollen.out[, apply(pollen.out, 2, function(x) sum(!is.na(x))) >= 5][,
1:2]
n.check <- nout[apply(pollen.out, 2, function(x) sum(!is.na(x))) >= 5][1:2]
mycut = date.levels
trig.out <- list()
trig.nout <- list()
for (i in 1:length(date)) {
  trig.out[[i]] <- get_dist(trig[[i]], date[[i]], mycut, 10)[[1]]
  trig.nout[[i]] <- get_dist(trig[[i]], date[[i]], mycut, 10)[[2]]
}
pollen.out <- t(do.call(rbind, lapply(trig.out, function(x) x[, trig.lev1 %in%
"11"])))
colnames(pollen.out) <- names(trig)
main.lab <- levels(temp$clust.ext)
for (i in 1:ncol(check)) {
  azip <- na.omit(as.character(unlist(tapply(temp$zip.zcta, temp$clust.ext,
unique)[colnames(check)[i]])))
  nzip <- as.numeric(unlist(n.check[[i]]))
  myplot <- get_meanplot(azip, "", clust.col[i])
  par(new = TRUE)
  out
}
for (i in 1:ncol(check)) {
  azip <- na.omit(as.character(unlist(tapply(temp$zip.zcta,
factor(temp$clust.ext, colnames(check)),
unique)[colnames(check)[i]])))
  nzip <- as.numeric(unlist(n.check[[i]]))
  par(new = T)
  y = check[, i]
  y <- y[!is.na(y)]
  forplot = BBands(y, n = 10, sd = 1)
  matplot(match(names(y), date.levels), forplot[, 1:3], type = "l", ylim =
c(0, 0.4), xlim = c(1,
185), axes = F, ylab = "", xlab = "", col = reg.colors[i], xaxt = "n",
lty = c(2,
1, 2), lwd = c(1, 2, 1))
}
legend("topright", col = c("red", "blue", "red", "blue"), c("Pollen count
(south)", "Pollen count (north)",
"% Trigger Dist (south)", "% Trigger Dist (north)"), lty = c(2, 2, 1, 1),
lwd = c(1,
1, 2, 2), cex = 0.75)
axis(4, at = 0:10/10, lab = 0:10/(2 * 5), las = 2)
axis(1, at = na.omit(match(mycut.month, date.levels)),
na.omit(date.levels[match(mycut.month,
date.levels)]), las = 2)
mtext("% Trigger Dist.", side = 4, line = 3)
##### FIGURE 4C
par(mar = c(6.1, 6.1, 8.1, 8.1))
check <- heat.out[, apply(heat.out, 2, function(x) sum(!is.na(x))) >= 5]
n.check <- nout[apply(heat.out, 2, function(x) sum(!is.na(x))) >= 5]

```

trigger analysis.txt

```

myplot <- mycut.5day
mytemp <- my.tmax
for (i in 1:ncol(check)) {
  astation <- na.omit(as.character(unlist(tapply(temp$station,
factor(temp$clust.ext, colnames(check)),
unique)[colnames(check)[i]])))
  azip <- na.omit(as.character(unlist(tapply(temp$zip.zcta,
factor(temp$clust.ext, colnames(check)),
unique)[colnames(check)[i]])))
  myplot <- get_tmax.plot(astation, clust.col[i], date.levels)
  nzip <- as.numeric(unlist(n.check[[i]]))
  par(new = TRUE)
}
reg.colors = c("blue", "red")
for (i in 1:ncol(check)) {
  azip <- na.omit(as.character(unlist(tapply(temp$zip.zcta,
factor(temp$clust.ext, colnames(check)),
unique)[colnames(check)[i]])))
  nzip <- as.numeric(unlist(n.check[[i]]))
  par(new = T)
  y = check[, i]
  forplot <- y[!is.na(y)]
  matplot(match(names(forplot), date.levels), my.bbands(forplot, 10, 1), type
= "l", ylim = c(0,
0.1), xlim = c(1, 185), axes = F, ylab = "", xlab = "", col =
reg.colors[i], lty = c(2,
1, 2), lwd = c(1, 2, 1))
}
legend("topleft", col = c("red", "blue", "red", "blue"), c("Max temp. (south)",
"Max temp. (north)",
"% Trigger Dist (south)", "% Trigger Dist (north)"), lty = c(2, 2, 1, 1),
lwd = c(1,
1, 2, 2), cex = 0.75)
axis(4, at = 0:10/100, lab = 0:10/(100), las = 2)
axis(1, at = na.omit(match(mycut.month, date.levels)),
na.omit(date.levels[match(mycut.month,
date.levels)]), las = 2)
mtext("% Trigger Dist.", side = 4, line = 3)
#####
pollen.data$ZIP <- as.character(pollen.data$ZIP)
pollen.data[1, 1] <- "07677"
get_meanplot <- function(azip, main.lab, mycol) {
  data <- as.matrix(pollen.data[pollen.data$ZIP %in% azip, 2:13])
  pollen.zip <- pollen.data[pollen.data$ZIP %in% azip, ]$ZIP
  m <- apply(data, 2, mean)
  s <- apply(data, 2, sd)
  e <- get_error(apply(data, 2, sum), s)
  # main.lab<-as.character(apploc[match(azip,apploc$zip.zcta),]$zip.place)
  # main.lab=main.lab
  out <- plot(1:12 + 0.5, m, ylim = c(0, 10), xlim = c(3, 9), main = main.lab,
ylab = "mean pollen count",
type = "l", col = mycol, lty = 2, xaxt = "n", las = 2, xlab = "")
arrows(1:12 + 0.5, as.numeric(m - e), 1:12 + 0.5, as.numeric(m + e), code =
3, xlim = c(3,
9), length = 0.05, angle = 90, col = mycol)
  pollen.zip
}
#####
temp <- temp.zip
wild <- data.frame(wild.data)
wa.fires = as.character(wild[wild$State %in% "Washington", ]$Name)
wa.data <- WA[, 6:13]
mymat <- apply(wa.data[, 2:ncol(wa.data)], 2, function(x) tapply(x,
wa.data$pm2pt5.date,
mean, na.rm = T))
mymat <- mymat[-1, ]
dat.mat = BBands(mymat[, 6][!is.na(mymat[, 6])], n = 2, "SMA", sd = 1)[, 1:3]
matplot(match(names(mymat[, 6][!is.na(mymat[, 6])]), date.levels), dat.mat, type
10agina p

```

```

                                trigger analysis.txt
= "l", ylim = c(0,
  200), xlim = c(1, 185), xaxt = "n", lty = c(1, 2, 1), xlab = "", ylab = "Max
AQI", col = "black")
axis(1, at = na.omit(match(mycut.month, date.levels)),
na.omit(date.levels[match(mycut.month,
  date.levels)]), las = 2)
get_zip.nearby <- function(x, dist) {
  cond <- rdist.earth(loc.data[, c("long", "lat")],
t(matrix(c(as.numeric(x$Longitude),
  as.numeric(x$Latitude)))) <= dist
  unique(as.character(loc.data[cond, ]$zip.zcta))
}
wa.fires = as.character(wild[wild$State %in% "Washington", ]$Name)
loc.data <- temp[unlist(apply(temp[, c("long", "lat")], 1, function(x)
sum(!is.na(x)) ==
  2)), c("zip.zcta", "long", "lat")]
wild <- apply(wild, 1, as.list)
par(new = T)
plot(NA, ylim = c(0, 200), xlim = c(1, 185), xlab = "", ylab = "", xaxt = "n")
forplot = unlist(lapply(wild, function(x) x[x$Name %in% wa.fires]$Start.Date))
lab.forplot = unlist(lapply(wild, function(x) x[x$Name %in% wa.fires]$Name))
abline(v = match(forplot, date.levels))
text(match(forplot, date.levels), 100, lab.forplot, srt = 90, cex = 0.5)
zip.nearby <- lapply(wild, get_zip.nearby, 200)
all.wa = unique(unlist(zip.nearby[unlist(lapply(wild, function(x) x$Name %in%
wa.fires))]))
wf.info = lapply(zip.nearby, function(x)
length(unique(temp.zip[temp.zip$zip.zcta %in% x,
  ]$healthCode)))
names(wf.info) = unlist(lapply(wild, function(x) x$Name))
aqi.out <- list()
zip.clust <- list()
mycut <- mycut.week
for (j in 1:length(zip.nearby)) {
  zip.clust[[j]] <- factor(temp.zip$clust.ext %in% zip.nearby[[j]])
  trig <- tapply(temp$get_worse, factor(zip.clust[[j]], levels = TRUE),
as.character)
  date <- tapply(temp$date, factor(zip.clust[[j]], levels = TRUE), as.Date)
  out <- list()
  nout <- list()
  for (i in 1:length(date)) {
    out[[i]] <- get_dist(trig[[i]], date[[i]], mycut, 5)[[1]]
    nout[[i]] <- get_dist(trig[[i]], date[[i]], mycut, 5)[[2]]
  }
  aqi.out[[j]] <- t(do.call(rbind, lapply(out, function(x) x[, trig.lev1 %in%
"16"])))
  colnames(aqi.out[[j]]) <- names(trig)
}
##### FIGURE 4D
pdf("check_aqi_wildfire.pdf")
clust.col <- rep("black", length(zip.clust))
for (j in 1:length(zip.clust)) {
  par(mar = c(6.1, 6.1, 8.1, 8.1))
  ####
  check <- aqi.out[[j]][, apply(aqi.out[[j]], 2, function(x) sum(!is.na(x)))
>= 5]
  n.check <- nout[apply(aqi.out[[j]], 2, function(x) sum(!is.na(x))) >= 5]
  if (length(check) > 0) {
    y = check
    forplot <- y[!is.na(y)]
    dat.mat = my.bbands(forplot, 3, 1)
    row.names(dat.mat) = names(forplot)
    matplot(match(row.names(dat.mat), date.levels), dat.mat, type = "l",
ylim = c(0,
  0.4), xlim = c(1, 185), axes = T, ylab = "", xlab = "", main = "")
    start = match(wild[[j]]$Start.Date, date.levels)
    end = ifelse(as.Date(wild[[j]]$End.Date) > "2015-09-09", 185,
  11
lagina p

```

```

                                trigger analysis.txt
match(wild[[j]]$End.Date,
      date.levels))
  rect(start, -1, end, 200, col = rgb(0.5, 0.5, 0.5, 1/4), border =
rgb(0.5, 0.5, 0.5,
    1/4))
  legend("topleft", col = c(clust.col[i]), c("PM2.5", "% Trigger Dist (Air
Quality)"),
        lty = c(2, 1), lwd = c(1, 3), cex = 0.75, bg = "white")
  axis(4, at = (0:10) * 20, lab = (0:10) * 5/(100), las = 2)
  mtext("% Trigger Dist.", side = 4, line = 2, cex = 1, las = 2)
}
}
dev.off()
#####
temp <- temp.clust2
reg.users = tapply(temp$healthCode, temp$clust.ext, unique)
bl.order = c("Animals", "Pollen", "A cold", "Smoking", "Exercise", "House dust",
"Air quality",
  "Change in weather", "More activity", "Mold", "Strong smells", "Extreme
Cold", "Other",
  "Stress", "Extreme heat", "I don't know", "None of these")
get_out.table = function(user.set) {
  out <- lapply(YouA.data[YouA.data$healthCode %in% user.set,
]$asthma_gets_worse_with,
    get_troubles)
  out.table <- table(unlist(out))/sum(table(unlist(out)))
  names(out.table) <- trig.lab
  other = sum(out.table[c(19, 20, 15)])
  animals = sum(out.table[7:9])
  mynew = out.table[c(1:5, 6, 10:14, 16:18, 21:22)]
  names(mynew) = trig.lab[c(1:5, 6, 10:14, 16:18, 21:22)]
  mynew = c(Animals = animals, Other = other, mynew)
  # mynew[match(bl.order, names(mynew))]
  sort(mynew, decreasing = T)
}
bl.north = get_out.table(reg.users[[1]])
bl.south = get_out.table(reg.users[[2]])
trig.lab <- c("A cold", "Exercise", "More activity", "Strong smells", "Exhaust
fumes", "House dust",
  "Dogs", "Cats", "Other furry animals", "Mold", "Pollen", "Extreme heat",
"Extreme Cold",
  "Change in weather", "Period", "Air quality", "Smoking", "Stress",
"Feelings", "Laughter",
  "I don't know", "None of these")
n <- length(trig.lab)
library("colorspace")
trig.colors <- sample(rainbow_hcl(n), n, replace = F)
trig.lev0 <- as.character(1:22)
mycut <- date.levels
trig.lev1 = trig.lev0
trig <- tapply(temp$get_worse, factor(temp$clust.ext), as.character)
date <- tapply(temp$date, factor(temp$clust.ext), as.Date)
out <- list()
nout <- list()
for (i in 1:length(date)) {
  out[[i]] <- get_dist(trig[[i]], date[[i]], mycut, 10)[[1]]
  nout[[i]] <- get_dist(trig[[i]], date[[i]], mycut, 10)[[2]]
}
get_stacks = function(x, bl.reg, mymain) {
  colnames(x) = trig.lab
  other = apply(x[, c(19, 20, 15, 5)], 1, sum)
  animals = apply(x[, 7:9], 1, sum)
  reg.ts <- cbind(Animals = animals, Other = other, x[, c(1:5, 6, 10:14,
16:18, 21:22)])
  reg.ts <- apply(reg.ts, 2, function(x) smooth.spline(na.omit(x))$y)
  reg.ts = t(reg.ts[, match(names(bl.reg), colnames(reg.ts))])
  reg.ts = cbind(bl.reg, reg.ts)
  barplot(prop.table(reg.ts, 2)[nrow(reg.ts):1, 1:133], main = mymain, col =
12agina p

```

```

                                trigger analysis.txt
trig.colors[match(row.names(reg.ts),
                    names(trig.colors))], border = NA, space = c(0, 10, rep(0, 104), 0.5,
rep(0, 26)),
                    xaxt = "n")
    legend("topright", names(trig.colors[match(row.names(reg.ts),
names(trig.colors))])),
        fill = trig.colors[match(row.names(reg.ts), names(trig.colors))][18:1],
cex = 0.8)

    axis(1, at = na.omit(match(mycut.month, date.levels)),
na.omit(date.levels[match(mycut.month,
        date.levels)]), las = 2)
}
##### SUPPLEMENTARY FIGURE 5
par(mfrow = c(3, 1))
trig.colors = sample(rainbow(length(bl.north)), length(bl.north))
names(trig.colors) = names(bl.north)
get_stacks(out[[1]], bl.north, "north")
get_stacks(out[[2]], bl.south, "south")

```

Cox proportional hazards - Study Attrition

Identify participant features that predict study attrition

Data pre-processed by amha_data_preprocessing.ipynb

Retention Analysis - One model with Age, Sex, Race, Education, Ethnicity, Gina, AgeOnset

Response Rate - logit(transform) 1) whole study period 2) survey / days in study

Table of Contents

['Cleaned Data'](#)

['EQ Survey Counts'](#)

['Weekly Survey Counts'](#)

['Daily Survey Counts'](#)

['Cohorts'](#)

['Cox Proportional Hazards Analysis'](#)

['Figure 2d: Study Retention By Enrollment Month'](#)

['Figure 2e: Study Retention By Age'](#)

['Suppl. Figure 2a: Predictor Correlation Heatmap '](#)

['Suppl. Figure 2b: Response Rate Histogram'](#)

['Supplementary Table 4a: univariate'](#)

['Supplementary Table 4b: multivariate'](#)

['Supplementary Table 4c: OLS'](#)

['Supplementary Figure 3: Heatmaps'](#)

Libraries

```
In [2]: import pandas as pd
import seaborn as sns
import os, sys
%config InlineBackend.figure_format = 'retina'
%matplotlib inline
%pylab inline
import seaborn
seaborn.set(rc={'axes.facecolor':'white', 'figure.facecolor':'white', 'grid.color':'lightgrey'})\
, font_scale=1)
```

```
rcParams['text.color'] = 'black'
rcParams['ytick.color'] = 'black'
rcParams['xtick.color'] = 'black'
rcParams['axes.facecolor'] = 'white'
rcParams['axes.edgecolor'] = 'white'
```

Populating the interactive namespace from numpy and matplotlib

Statistical Functions

```
In [3]: def construct_contingency_table(df, x, y):
        '''
        This function creates a contingency table
        using values from columns x and y in df
        '''
        return df.groupby([x,y])['conting_count'].sum().unstack()

def cramersV_from_df(df, x, y):
    '''
    This function returns cramersV using
    values from columns x and y in df
    '''
    from scipy import stats
    df['conting_count'] = 1
    contingency_df = df.groupby([x,y])['conting_count'].sum().unstack().values
    chi2, p, ddof, expected = stats.chi2_contingency(contingency_df)
    CV = np.sqrt(chi2 / len(df[x]) * (min(len(df[x].unique()), len(df[y].unique())) - 1))
    return CV

def cramersV(contingency_df, x_sum, x_unique, y_unique):
    '''
    This function calculates cramersV
    from the values of a N x 2 contingency table
    '''
    from scipy import stats
    chi2, p, ddof, expected = stats.chi2_contingency(contingency_df)

    CV = np.sqrt(chi2 / (x_sum * (min(x_unique, y_unique) - 1)))
    return CV
```

Functions

```
In [4]: def strip_brackets(series):

        def bracket_repalce(x):
            if x == '[':
                return np.NaN
            else:
                return x.rstrip(']').lstrip('[')

        series = [bracket_repalce(i) if type(i)==str else i for i in series]
```

```

try:
    return map(int, series)
except:
    return series

```

```

In [5]: def get_count(vector):
        '''
        This function counts booleans
        '''
        if True in vector or False in vector:
            return [1 if i in [True, False] else 0 for i in vector]

        vector_counts = []
        for i in vector:
            if i == '[]':
                vector_counts.append(0)
                continue
            if i > -1:
                vector_counts.append(1)
                continue
            vector_counts.append(0)

        return vector_counts

```

```

In [6]: def get_duplicates(df, col, date, id_name='healthCode'):
        '''
        This function returns a dataframe with duplicate
        entries ordered by time
        '''

        df.sort(date, inplace=True)

        dups = df.reset_index()[[id_name, col]].dropna().drop_duplicates()[id_name].value_counts()
        dups = dups[dups>1]
        df = df.ix[dups.index].set_index(date, append=True)[col]
        return df

```

```

In [7]: metadata_cols = ['recordId', 'createdOn', 'appVersion', 'ROW_VERSION', 'uploadDate', 'ROW_ID']
        set(metadata_cols)

        def data_columns(cols, metadata_cols=['recordId', 'createdOn', 'appVersion', 'ROW_VERSION', 'uploadDate', 'ROW_ID', 'phoneInfo']):
            return list( set(cols) - set(metadata_cols))

        def get_data(df, cols):

```

```
return df[data_columns(cols)]
```

```
In [8]: def select_date(df, date_col, year_month_day):
        '''
        This function returns a subset of the data
        with the date_col < year_month_day
        '''

        year, month, day = year_month_day
        return df[df[date_col] < pd.datetime(year, month, day).date()]
```

Data

```
In [9]: def get_table_columns(dir_path='./test_data/'):
        '''
        This function creates a dictionary of table columns
        indexed on table name
        '''

        csvs = [i for i in os.listdir(dir_path) if i.endswith('.csv')]

        table_columns = dict()
        for i in csvs:
            table_columns[i.rstrip('.csv')] = list(set(pd.read_table(dir_path
+ i, sep=',', nrows=1).columns.tolist()) - set(['healthCode', 'externalId']
))
        return table_columns
```

```
In [10]: table_columns = get_table_columns()

def get_table_responses(df, table, table_cols=table_columns):
    '''
    This function returns table rows with at least
    one non-Na value
    '''
    return df.ix[table][table_columns[table]].dropna(how='all')

table_columns
```

```
Out[10]: {'AboutYou': ['education',
                        'recordId',
                        'createdOn',
                        'appVersion',
                        'smoking_status',
                        'ROW_VERSION',
                        'race',
                        'smoking_years',
                        'uploadDate',
                        'ethnicity',
                        'Income',
                        'phoneInfo',
                        'ROW_ID',
```

```

    'health_insurance',
    'avg_cigarettes'],
'AsthmaDailyPrompt': ['get_worse',
    'quick_relief_puffs',
    'day_symptoms',
    'createdOn',
    'appVersion',
    'medicine_change',
    'recordId',
    'ROW_VERSION',
    'uploadDate',
    'night_symptoms',
    'medicine',
    'phoneInfo',
    'ROW_ID',
    'use_qr',
    'peakflow'],
'AsthmaHistory': ['doc_times',
    'nights',
    'intubated',
    'emergency',
    'times_hospitalized',
    'recordId',
    'limited activity',
    'createdOn',
    'appVersion',
    'age_when_diagnosed',
    'symptoms',
    'hospitalized_times',
    'ROW_VERSION',
    'emergency_times',
    'uploadDate',
    'miss_work',
    'phoneInfo',
    'ROW_ID',
    'oral steroids',
    'seen_doc'],
'AsthmaMedication': ['other_meds',
    'alvesco_dose',
    'ROW_ID',
    'flovent_diskus_dose',
    'prescribed_asthma_control_medication',
    'dulera_dose',
    'past_month_quick_relief',
    'nebulizer_meds',
    'steroid_dose',
    'appVersion',
    'symbicort_dose',
    'daily_yes',
    'phoneInfo',
    'pulmicort_dose',
    'steroid_which',
    'controlmed',
    'advair_hfa_dose',
    'daily_controller_medication',
    'uploadDate',
    'control_puffs',

```

```

'breo_dose',
'nebulize_daily',
'advair_diskus_dose',
'qvar_dose',
'daily_inhaled_medicine',
'asmanex_dose',
'recordId',
'ROW_VERSION',
'quick_relief',
'flovent_hfa_dose',
'createdOn',
'use_nebulizer'],
'AsthmaWeeklyPrompt': ['asthma_doc_visit',
'appVersion',
'oral_steroids_when',
'emergency_room',
'asthma_medicine',
'ROW_ID',
'missed_work_days',
'missed_work',
'recordId',
'admission',
'prednisone_when',
'admitted_end',
'er_when',
'phoneInfo',
'limitations',
'admitted_when',
'prednisone',
'oral_steroids',
'uploadDate',
'createdOn',
'ROW_VERSION',
'limitations_days',
'side_effects'],
'EQ_5D': ['health_today',
'slider_instructions',
'pain',
'mobility',
'recordId',
'createdOn',
'EQ5Instructions',
'appVersion',
'ROW_VERSION',
'intro',
'uploadDate',
'phoneInfo',
'selfcare',
'ROW_ID',
'usual_activities',
'depression'],
'HealthKitDataCollector': ['recordId',
'createdOn',
'appVersion',
'ROW_VERSION',
'uploadDate',
'phoneInfo',

```

```

'ROW_ID',
'data.csv'],
'MedicalHistory': ['peripheral',
'appVersion',
'stroke',
'tissue',
'other_lung_disease',
'ulcer',
'Congestive',
'ROW_ID',
'dementia',
'kidney',
'recordId',
'ROW_VERSION',
'malignant_lymphoma',
'phoneInfo',
'arthritis',
'leukemia',
'tested',
'uploadDate',
'heart_attack',
'createdOn',
'tumor',
'chronic_pulmonary_disease',
'allergic_to',
'liver'],
'Milestone': ['weight',
'alleviate_troubles',
'use_aap',
'symptoms2',
'met_goal',
'current_goal',
'nights2',
'ROW_ID',
'prevent_ed',
'appVersion',
'phoneInfo',
'current_troubles',
'recordId',
'make_aap',
'uploadDate',
'height',
'cause_visit',
'prevent_visit',
'gender',
'age',
'daily_yes2',
'best_peakflow',
'ROW_VERSION',
'past_month_quick_relief2',
'got_spirometry',
'createdOn',
'limited_activity2'],
'YourAsthma': ['asthma_gets_worse_with',
'lung_function',
'plan',
'recordId',

```

```
'createdOn',
'appVersion',
'flu_prompt',
'peak_flow',
'troubles_about_asthma',
'ROW_VERSION',
'aap_prompt',
'flushot',
'phoneInfo',
'asthma_control',
'ROW_ID',
'uploadDate'],
'aqiResponse': ['aqiResponse.json.reporting_area',
'aqiResponse.json.reports',
'recordId',
'createdOn',
'appVersion',
'ROW_VERSION',
'uploadDate',
'aqiResponse.json.state_code',
'phoneInfo',
'ROW_ID']}]}
```

```
In [11]: #Common columns across tables
from collections import Counter
c = Counter([n for i in table_columns.keys() for n in table_columns[i] ])
c.most_common(10)
```

```
Out[11]: [('recordId', 11),
('appVersion', 11),
('phoneInfo', 11),
('createdOn', 11),
('ROW_VERSION', 11),
('ROW_ID', 11),
('uploadDate', 11),
('quick_relief_puffs', 1),
('alleviate_troubles', 1),
('emergency_times', 1)]
```

Cohorts

['Table of Contents'](#)

```
In [12]: # Gina classifications

cohort = pd.read_table('../analysis/cohorts_first.tsv', sep='\t')
cohort.columns = ['cohort', 'healthCode', 'gina']
cohort['gina'] = cohort['gina'].str.replace(' ', '')

baseline = cohort[cohort.cohort=='baseline']
baseline.name = 'baseline'
```

```
robust = cohort[cohort.cohort=='robust']
robust.name = 'robust'

milestone = cohort[cohort.cohort=='milestone']
milestone.name = 'milestone'
```

Clean Synapse Data

['Table of Contents'](#)

```
In [13]: #SAGE DATA
df_filtered = pd.read_table('../analysis/sage_data_df_filtered_2015_12_04.
txt.gz', sep='\t', compression='gzip', \
                           index_col=['table', 'healthCode'])

df_filtered['study_entry'] = pd.to_datetime(df_filtered.study_entry)
df_filtered['date'] = pd.to_datetime(df_filtered.date)

#study entry for each healthcode
study_entry = select_date(df_filtered.ix['AsthmaDailyPrompt'], 'date', (20
15,9,10)).study_entry.dt.date
study_entry = study_entry.reset_index().drop_duplicates().set_index('healthCode')

#df_filtered = select_date(df_filtered, 'date', (2015,9,10))
#df_filtered = df_filtered[df_filtered.index.get_level_values(1).isin(base
line.healthCode)]
```

```
/Users/ers/anaconda2/lib/python2.7/site-packages/IPython/core/interactives
hell.py:2717: DtypeWarning: Columns (2,6,7,8,14,17,18,20,21,23,24,25,26,31
,35,37,42,43,48,49,55,56,58,59,62,66,67,68,74,75,77,80,81,83,84,85,89,91,9
8,99,100,101,107,110,112,124,125,126,127,129,131,132,133) have mixed types
. Specify dtype option on import or set low_memory=False.
interactivity=interactivity, compiler=compiler, result=result)
```

```
In [14]: #Setting entry data to 3-23
entry_date = df_filtered.groupby(level=1)['date'].min()
entry_date= entry_date.reset_index().set_index(pd.DatetimeIndex(entry_date
))
entry_date_daterange = entry_date['2015-3-23':'2015-3-29'].healthCode.uniq
ue()
entry_date_daterange.shape
```

Out[14]: (310,)

```
In [15]: #Baseline Data

baseline_data = pd.read_table('../analysis/baseline_first_survey_nicole_20
15_12.tsv.gz', sep='\t', \
                             compression='gzip', index_col='healthCode')
baseline_data.info()

<class 'pandas.core.frame.DataFrame'>
```

Index: 6363 entries, 000152e4-c14d-4e0e-8bc4-d96e53acc868 to ffee35de-8a9e-4105-a3fa-b9cd870cabe5		
Data columns (total 71 columns):		
doc_times	2183	non-null float64
nights	6192	non-null float64
emergency	6191	non-null object
times_hospitalized	6187	non-null object
miss_work	6186	non-null float64
emergency_times	701	non-null float64
age_when_diagnosed	6055	non-null float64
symptoms	6194	non-null float64
hospitalized_times	394	non-null float64
intubated	6187	non-null float64
createdOn_His_data_2015_12_08	6240	non-null object
limited activity	6193	non-null float64
seen_doc	6188	non-null object
oral steroids	6162	non-null float64
past_month_quick_relief	5861	non-null float64
flovent_hfa_dose	361	non-null float64
advair_diskus_dose	950	non-null float64
ymbicort_dose	662	non-null float64
controlmed	3420	non-null float64
createdOn_Med_data_2015_12_08	5898	non-null object
quick_relief	5651	non-null object
advair_hfa_dose	211	non-null float64
qvar_dose	345	non-null float64
asmanex_dose	148	non-null float64
steroid_dose	515	non-null float64
daily_controller_medication	3887	non-null object
daily_yes	5699	non-null float64
alvesco_dose	48	non-null float64
daily_inhaled_medicine	3435	non-null float64
flovent_diskus_dose	72	non-null float64
prescribed_asthma_control_medication	5860	non-null float64
pulmicort_dose	171	non-null float64
dulera_dose	362	non-null float64
steroid_which	3879	non-null float64
chronic_pulmonary_disease	3749	non-null object
heart_attack	3767	non-null object
createdOn_MedH_data_2015_12_08	3775	non-null object
peripheral	3753	non-null object
arthritis	3751	non-null object
leukemia	3748	non-null object
tumor	3759	non-null float64
tested	3753	non-null object
stroke	3757	non-null float64
tissue	3758	non-null object
other_lung_disease	3753	non-null object
malignant_lymphoma	3758	non-null object
ulcer	3757	non-null object
Congestive	3770	non-null object
dementia	3761	non-null object
allergic_to	3584	non-null object
kidney	3750	non-null object
liver	3756	non-null float64
smoking_status	4244	non-null float64
health_insurance	4227	non-null float64

```

race                4249 non-null object
smoking_years       930 non-null float64
ethnicity           4185 non-null float64
Income              4231 non-null float64
createdOn_You_data_2015_12_08  4274 non-null object
education           4234 non-null float64
avg_cigarettes      920 non-null float64
asthma_gets_worse_with  4472 non-null object
lung_function       4467 non-null float64
troubles_about_asthma  4460 non-null object
peak_flow           765 non-null float64
flu_prompt          2223 non-null object
aap_prompt          3538 non-null object
flushot             4468 non-null object
createdOn_YouA_data_2015_12_08  4494 non-null object
asthma_control       4416 non-null object
plan                4471 non-null object
dtypes: float64(39), object(32)
memory usage: 3.5+ MB

```

```

In [16]: #Age & Sex df
age_sex = pd.read_table('../data/raw_agesex_2015_12_4.tsv', sep='\t')
age_sex.set_index('healthCode', drop=False, inplace=True)
col_dict = dict((i, i.replace('NonIdentifiableDemographics.json.patient',
'')) for i in age_sex.columns)
age_sex.rename(columns=col_dict, inplace=True)
age_sex = age_sex.drop_duplicates()
age_sex.sort(columns='createdOn', inplace=True)
age_sex.info()

```

```

<class 'pandas.core.frame.DataFrame'>
Index: 3408 entries, 5288ddee-c1a6-479c-8dc2-307a96ca26df to 094e479f-0d0e
-419a-9ddf-308c225105b8
Data columns (total 14 columns):
recordId                3408 non-null object
healthCode              3408 non-null object
externalId              0 non-null float64
uploadDate              3408 non-null object
createdOn               3408 non-null object
appVersion              3408 non-null object
phoneInfo               3408 non-null object
WeightPounds            3405 non-null float64
BiologicalSex            3252 non-null object
HeightInches            3405 non-null float64
WakeUpTime              0 non-null float64
CurrentAge              3253 non-null float64
GoSleepTime             0 non-null float64
NonIdentifiableDemographics.json.item  3405 non-null object
dtypes: float64(6), object(8)
memory usage: 399.4+ KB

```

```

/Users/ers/anaconda2/lib/python2.7/site-packages/ipykernel/__main__.py:7:
FutureWarning: sort(columns=....) is deprecated, use sort_values(by=....)

```

```

In [17]: #Biological Sex df
biosex = age_sex[['healthCode', 'BiologicalSex', 'createdOn']].dropna(how=
'all').drop_duplicates(subset=['healthCode', 'BiologicalSex'])

```

```
biosex.sort('createdOn', inplace=True)
biosex = biosex.groupby('healthCode')['BiologicalSex'].value_counts().reset_index().drop_duplicates('healthCode').set_index('healthCode')['BiologicalSex']
```

```
/Users/ers/anaconda2/lib/python2.7/site-packages/ipykernel/__main__.py:3:
FutureWarning: sort(columns=....) is deprecated, use sort_values(by=.....)
app.launch_new_instance()
```

```
In [18]: #Age df
age = age_sex[['healthCode', 'CurrentAge', 'createdOn']].dropna(how='all')
        .drop_duplicates(subset=['healthCode', 'CurrentAge'])
age = age[~age.CurrentAge.isnull()]
age.sort('createdOn', inplace=True)
age = age.groupby('healthCode')['CurrentAge'].value_counts().reset_index()
        .drop_duplicates('healthCode').set_index('healthCode')['CurrentAge']
print age.shape
```

```
(2109,)
```

```
/Users/ers/anaconda2/lib/python2.7/site-packages/ipykernel/__main__.py:4:
FutureWarning: sort(columns=....) is deprecated, use sort_values(by=.....)
```

Daily & Weekly Surveys Collapsed

```
In [19]: #SAGE DATA
daily_weekly_df = pd.read_table('../analysis/daily_weekly_prompt_data_collapsed_2015_12_04.tsv.gz',\
                                sep='\t', compression='gzip', index_col=['healthCode', 'study_day'])

daily_weekly_df = daily_weekly_df.join(study_entry, how='left')

daily_weekly_df['dateTime'] = daily_weekly_df.dateTime.map(pd.to_datetime)
#entered study
daily_weekly_df['date'] = pd.to_datetime(daily_weekly_df.dateTime).dt.date
daily_weekly_df['study_entry'] = pd.to_datetime(daily_weekly_df.study_entry, unit='ms')
```

```
/Users/ers/anaconda2/lib/python2.7/site-packages/IPython/core/interactiveshell.py:2717: DtypeWarning: Columns (40,42,47,48,52,53,54) have mixed type s. Specify dtype option on import or set low_memory=False.
interactivity=interactivity, compiler=compiler, result=result)
```

```
In [20]: len(daily_weekly_df[daily_weekly_df.date>pd.datetime(2015,3,3).date()].index.get_level_values(0).unique())
```

```
Out[20]: 6515
```

Cox Proportional Hazards

[Table of Contents](#)

```
In [21]: days_till_2015_09_09 = pd.datetime(2015,9,8).date() - daily_weekly_df.reset_index().drop_duplicates(subset=['healthCode', 'study_entry']).set_index('healthCode')['study_entry'].dropna()
days_till_2015_09_09 = days_till_2015_09_09.dt.days.groupby(level=0).max().dropna()
days_till_2015_09_09.name = 'days_till_2015_09_09'
days_till_2015_09_09.shape

#restricting analysis to those in study for at least 180 days
days_till_2015_09_09_180days = days_till_2015_09_09[days_till_2015_09_09>179]
days_till_2015_09_09_180days.name = 'days_till_2015_09_09'
```

```
In [22]: #More efficient way to find censored healthCodes
daily_weekly_df_ts = daily_weekly_df.reset_index().set_index(pd.DatetimeIndex(daily_weekly_df.dateTime))
censored_healthcodes = daily_weekly_df_ts['2015-8-27':'2015-9-8'].healthCode.unique()
print censored_healthcodes.shape, 'censored healthCodes'

entered_before_june = daily_weekly_df_ts[:'2015-6-8'].healthCode.unique()
print entered_before_june.shape, 'entered before June'
```

```
(260,) censored healthCodes
(5768,) entered before June
```

```
/Users/ers/anaconda2/lib/python2.7/site-packages/pandas/indexes/base.py:3027: FutureWarning: In the future, 'NAT <= x' and 'x <= NAT' will always be False.
    result = func(np.asarray(other))
```

Creation of dataframes for Multivariate Cox Analysis

```
In [23]: import lifelines
from lifelines import CoxPHFitter
from lifelines.datasets import load_regression_dataset
from lifelines.utils import k_fold_cross_validation
pd.set_option('display.precision',4)

def study_entry_month_calendar(date):
    if date.day > 8: #march 10
        return date.month - 3
    return date.month - 4

regression_dataset = load_regression_dataset()
cf = CoxPHFitter(normalize=False)

aboutyou = baseline_data.copy()
study_entry_month = df_filtered.study_entry.map(study_entry_month_calendar).reset_index(level=0, drop=True)
study_entry_month = study_entry_month.reset_index().drop_duplicates().set_index('healthCode')
study_entry_month.columns = ['study_entry_month']
```

```

study_entry_week = df_filtered.date - pd.datetime(2015,03,9)
study_entry_week = study_entry_week.dt.days
study_entry_week = study_entry_week.map(lambda x: 180 if x > 180 else x)
study_entry_week = pd.cut(study_entry_week,bins=range(0,365,14), retbins=False, labels=False)
study_entry_week.name = 'study_entry_week'
study_entry_week.reset_index(0, drop=True, inplace=True)
study_entry_week = study_entry_week.groupby(level=0).min().astype(float)


daily_survey_q = ['quick_relief_puffs', 'medicine', 'day_symptoms', \
                  'night_symptoms', 'peakflow', 'get_worse', \
                  'use_qr', 'medicine_change']


daily_prompt_data_cols = ['quick_relief_puffs', 'medicine', 'get_worse', \
                          'day_symptoms', 'night_symptoms', 'peakflow', \
                          'value', 'study_day', 'createdOn', 'dateTime', \
                          'use_qr', 'medicine_change']


daily_survey_standardqs = [ 'day_symptoms', 'night_symptoms', 'peakflow', \
                             'get_worse', 'use_qr']


cox_cols = [ 'gina', 'last_survey_censored', \
             'event_observed', 'education', \
             'age_when_diagnosed', 'Income', \
             'health_insurance', 'ethnicity', \
             'CurrentAge', 'BiologicalSex', \
             'study_entry_month']


print 'Columns in multivariate CoxPH Model\n', cox_cols


#int values
last_study_day = daily_weekly_df[daily_survey_standardqs].dropna(how='all')
last_study_day = last_study_day.groupby('healthCode')['study_day'].max()
last_study_day = last_study_day.map(lambda x: 180 if x > 179 else x)
last_study_day.name = 'last_survey_censored'


#restricting to entrants before June
last_study_day = last_study_day[last_study_day.index.isin(entered_before_june)]


#print last_study_day.shape
last_study_day = aboutyou.join(last_study_day) #joining about you data
last_study_day = last_study_day.join(study_entry_month) #joining study entry month data
last_study_day = last_study_day.join(biosex) #joining biological sex data
last_study_day = last_study_day.join(age, how='inner') #joining age data

```

```

#print last_study_day.shape, 'after age sex joining'

last_study_day = last_study_day[last_study_day.index.isin(days_till_2015_0
9_09.index)] #restricting data up to 9/9/2015
#print last_study_day.shape
last_study_day = last_study_day.join(days_till_2015_09_09, how='inner')

#restricting to robust users
last_study_day = last_study_day.join(robust.drop_duplicates().set_index('h
ealthCode')['gina'], how='inner')

#adding censoring data
last_study_day['event_observed'] = [False if i in censored_healthcodes els
e True for i in last_study_day.index]

#print last_study_day.shape
cox_df = last_study_day[cox_cols].dropna(how='any')

#print cox_df.shape

replacement_dict = {'gina':{'Uncontrolled':2, 'PartlyControlled':1, 'WellC
ontrolled':0},\
                    'education':{'1':1, '2':1, '3':1, '4':2, '5':2, '6':3,
'7':3, '8':np.NaN},\
                    'BiologicalSex':{'Female':0, 'Male':1},\
                    'health_insurance':{'3':0, '1':1, '2':1, '4':np.NaN},\
                    #Nicoles: 'education':{'1':1, '2':1, '3':2, '4':3, '5':3, '6':4, '7':4
, '8':np.NaN}

# replacement_dict = {'gina':{'Uncontrolled':2, 'PartlyControlled':1, 'Wel
lControlled':0},\
#                    'BiologicalSex':{'Female':0, 'Male':1}}

cox_df['education'] = cox_df['education'].astype(int).astype(str)
cox_df['health_insurance'] = cox_df['health_insurance'].astype(int).astype
(str)
cox_df.replace(to_replace=replacement_dict, inplace=True)
cox_df['education'] = cox_df['education'].astype(float)

print
print 'Sample size before filtering on missing education, income, health i
nsurance, ethnicity, and current age\n', cox_df.shape
print
cox_df = cox_df[(cox_df.education < 8) & \
                (cox_df.Income < 6) & \
                (cox_df.health_insurance < 4) & \
                (cox_df.ethnicity < 3) & \
                (cox_df.CurrentAge > 0)] #

print 'Sample size after filtering on missing education, income, health in
surance, ethnicity, and current age\n', cox_df.shape
print

```

```

cox_df['Current_Age_binned'] = pd.cut(cox_df['CurrentAge'], range(0,100,10), labels=False, retbins=False)
#cox_df = cox_df[cox_df.age_when_diagnosed<20]

cox_df[list( set(cox_df.columns) - set(cox_cols) )]

#cox_df = cox_df.join(study_entry_week, how='inner')
#cox_cols.append('study_entry_week')

scores = k_fold_cross_validation(cf, cox_df, 'last_survey_censored', event_col='event_observed', k=5)

#Five fold cross validation iteration scores
# print
# print scores
# print mean(scores)
# print std(scores)

```

Columns in multivariate CoxPH Model

```

['gina', 'last_survey_censored', 'event_observed', 'education', 'age_when_diagnosed', 'Income', 'health_insurance', 'ethnicity', 'CurrentAge', 'BiologicalSex', 'study_entry_month']

```

Sample size before filtering on missing education, income, health insurance, ethnicity, and current age
(620, 11)

Sample size after filtering on missing education, income, health insurance, ethnicity, and current age
(537, 11)

In [24]: #Fitting Multivariate Cox PH model

```

m = cf.fit( cox_df[cox_cols], \
            'last_survey_censored', event_col='event_observed', \
            ) #
m_summary = m.summary
m_summary['abs_z'] = m_summary['z'].map(np.abs)
m_summary.sort(columns=['abs_z'], ascending=False, inplace=True)
del m_summary['abs_z']
#m_summary = m_summary.apply(np.round, args=[3])

#exp(coef) is the hazard ratio
#coef is the increase in having the event for every 1 unit increase in X
# therefore if coef is negative then the predictor decreases the hazard of an outcome

```

/Users/ers/anaconda2/lib/python2.7/site-packages/ipykernel/__main__.py:6: FutureWarning: sort(columns=....) is deprecated, use sort_values(by=.....)

In [25]: #CramersV Correlations between categorical variables

```

cox_cols_categ = ['health_insurance', \
                  'ethnicity', 'BiologicalSex']

cramersV_df = []

```

```
for categ1 in cox_cols_categ:
    cramersV_df.append([])
    for categ2 in cox_cols_categ:
        cramersV_df[-1].append(cramersV_from_df(cox_df, categ1, categ2))
cramersV_df = pd.DataFrame(cramersV_df, columns=cox_cols_categ, index=cox_cols_categ)
cramersV_df
```

Out[25]:

	health_insurance	ethnicity	BiologicalSex
health_insurance	NaN	0.0167	0.0061
ethnicity	0.0167	NaN	0.0413
BiologicalSex	0.0061	0.0413	NaN

Supplementary Figure 2a

Predictor Correlation Heatmaps

[Table of Contents](#)

```
In [26]: #Covariance Matrix for Numeric Predictors (Standardized)
#Pearson's correlation
cov_df_numeric = cox_df[['study_entry_month', 'CurrentAge', 'age_when_diagnosed', 'Income', 'gina', 'health_insurance', 'education', 'ethnicity', 'BiologicalSex']]
cov_df_numeric = cov_df_numeric.corr()
cov_df_numeric.columns = [i.replace('_', ' ').replace('CurrentAge', 'current age').replace('BiologicalSex', 'biological sex').lower() for i in cov_df_numeric.columns]
cov_df_numeric.index = [i.replace('_', ' ').replace('CurrentAge', 'current age').replace('BiologicalSex', 'biological sex').lower() for i in cov_df_numeric.columns]

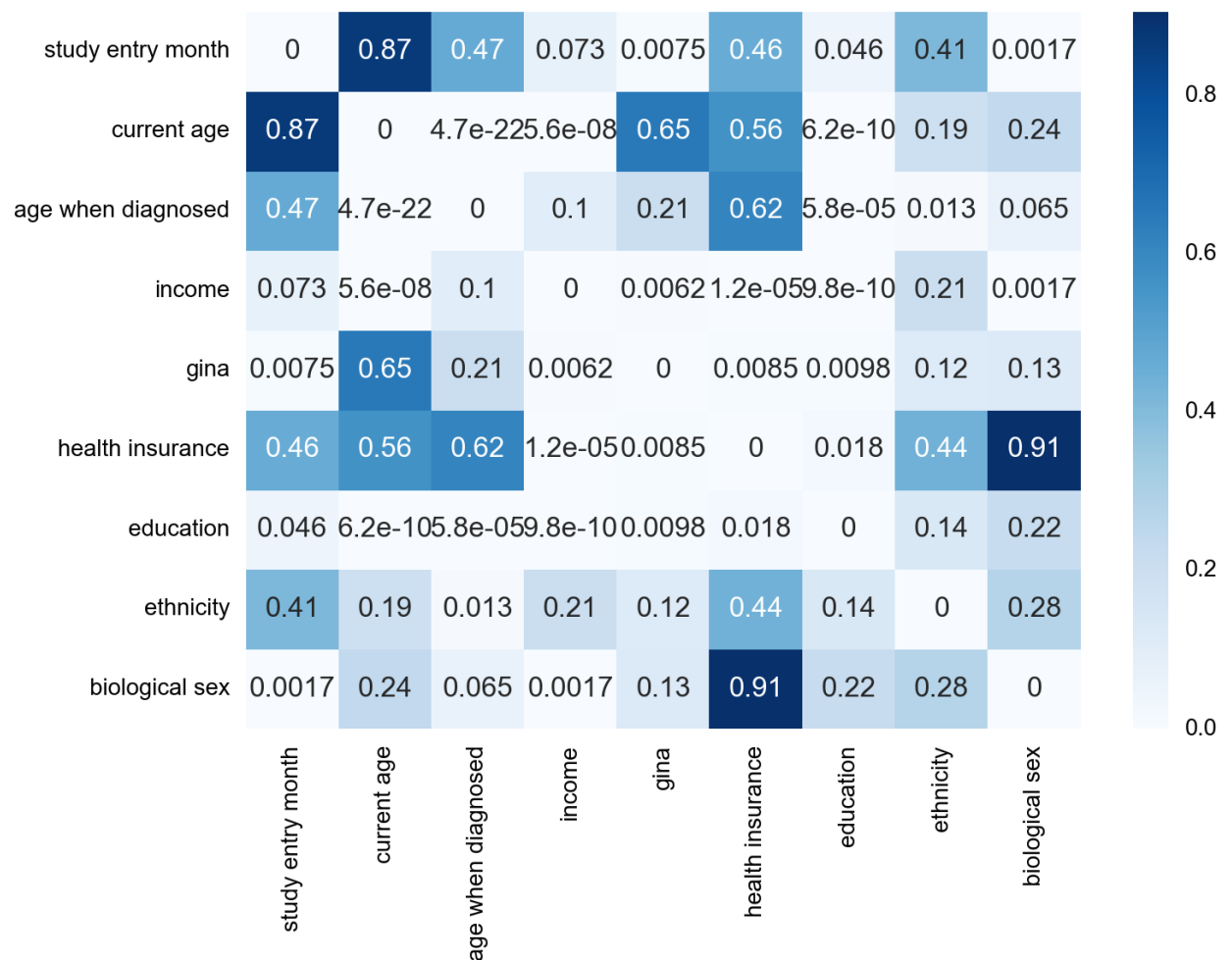
cov_df_numeric_heatmap = seaborn.heatmap(cov_df_numeric, annot=True)
```



```
In [27]: #Pearson's p-value
from scipy import stats
cov_df = cox_df[['study_entry_month', 'CurrentAge', 'age_when_diagnosed',
'Income', 'gina', 'health_insurance', 'education', 'ethnicity', 'Biological
Sex']]

cor_pval = []
for c1 in cov_df.columns:
    cor_pval.append([])
    for c2 in cov_df.columns:
        cor_pval[-1].append(stats.pearsonr(cov_df[c1], cov_df[c2])[-1])
cor_pval = pd.DataFrame(cor_pval, columns=cov_df.columns, index=cov_df.col
umns)
cor_pval.columns = [i.replace('_', ' ').replace('CurrentAge', 'current age
').replace('BiologicalSex', 'biological sex').lower() for i in cor_pval.co
lumns]
cor_pval.index = [i.replace('_', ' ').replace('CurrentAge', 'current age')
.replace('BiologicalSex', 'biological sex').lower() for i in cor_pval.colu
mns]
seaborn.heatmap(cor_pval, annot=True, cmap='Blues')
```

```
Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x120a56d90>
```



In [28]: #Merged Figure

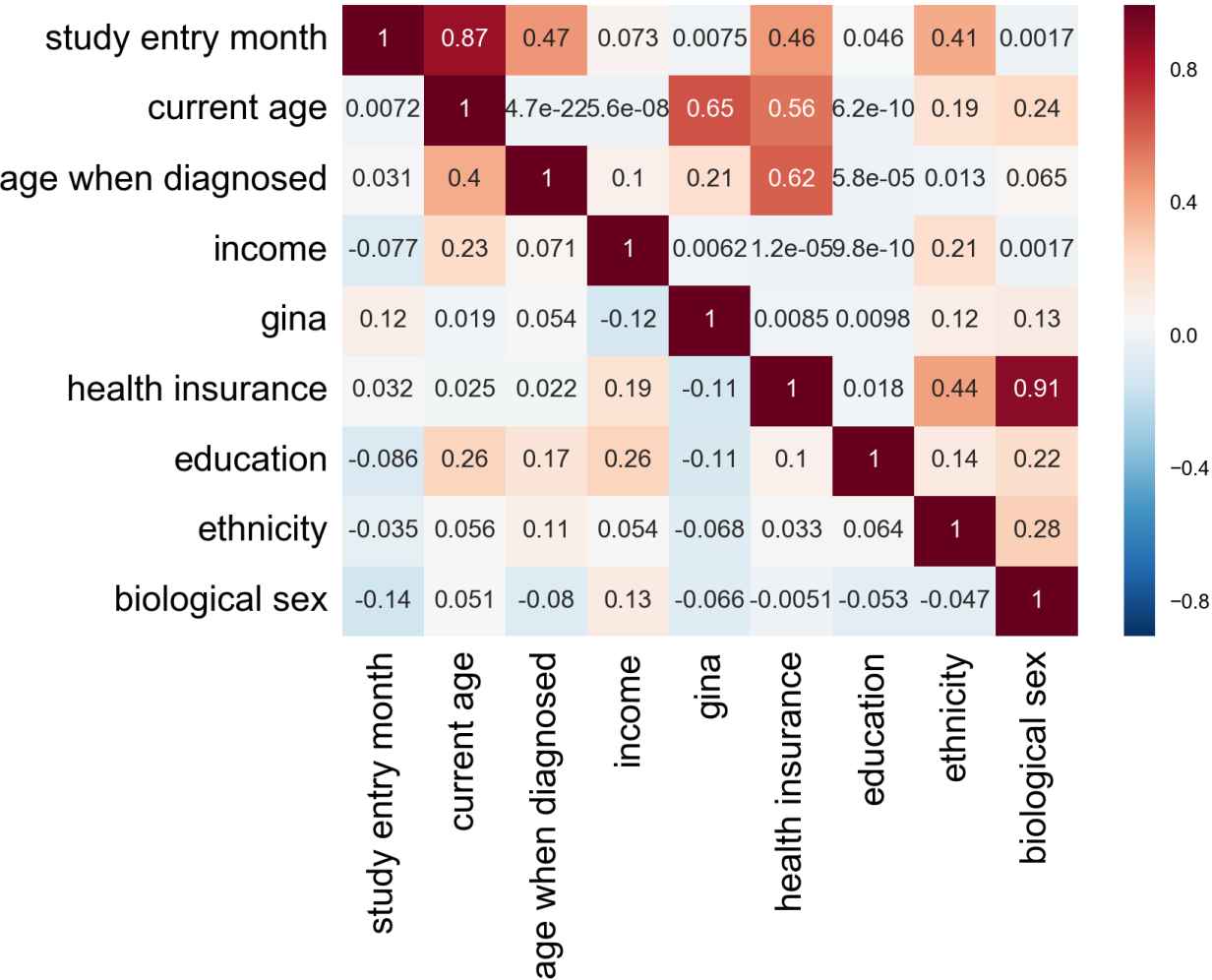
```
def triangle_merge(df_triu, df_tril):
    """
    Combines values from upper triangle
    and lower triangle from dfs
    """
    triu = pd.DataFrame(np.triu(df_triu.values))
    triu.replace(0.0, np.NaN, inplace=True)
    tril = pd.DataFrame(np.tril(df_tril))
    tril.replace(0.0, np.NaN, inplace=True)

    combined_df = triu.combine_first(tril)
    combined_df.columns = df_triu.columns
    combined_df.index = df_triu.index
    return combined_df

combined_df = triangle_merge(cor_pval, cov_df_numeric)
seaborn.heatmap(combined_df.replace(0.0, 1.0), annot=True, center=0.05)
plt.xticks(fontsize='x-large')
plt.yticks(fontsize='x-large')
#plt.tight_layout()
#plt.savefig('/Users/ers/Dropbox/feasibility_figures_ERS/CorrelationMatrix.svg', dpi=600, format='svg')
```

Out[28]: (array([0.5, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5]),

<a list of 9 Text yticklabel objects>)



Response Rate Histogram

Supplementary Figure 2b

Construction of dataframe for Ordinary Least Squares Model

Data also used for Supplemental Table 4c

[Table of Contents](#)

```
In [30]: #DATAFRAME CONSTRUCTION FOR ORDINARY LEAST SQUARES MODEL

import numpy as np
import statsmodels.api as sm
import statsmodels.formula.api as smf
```

```

from scipy.special import logit

daily_survey_standardqs = [ 'day_symptoms', 'night_symptoms', 'peakflow',
\
                             'get_worse', 'use_qr']

#number of days in study for each healthcode
days_in_study_2015_9_9 = select_date(daily_weekly_df, 'date', (2015,9,8))

days_in_study_2015_9_9 = days_in_study_2015_9_9.study_entry.dt.date
days_in_study_2015_9_9 = pd.to_datetime(days_in_study_2015_9_9, unit='ms')
days_in_study_2015_9_9 = pd.datetime(2015,9,8) - days_in_study_2015_9_9
days_in_study_2015_9_9 = days_in_study_2015_9_9.dt.days
days_in_study_2015_9_9.name = 'days_in_study_2015_9_9'
days_in_study_2015_9_9.sort(inplace=True, ascending=False)
days_in_study_2015_9_9 = days_in_study_2015_9_9.reset_index().drop_duplicates().set_index('healthCode')

days_till_2015_09_09 = pd.datetime(2015,9,8).date() - daily_weekly_df['study_entry'].dropna()
days_till_2015_09_09 = days_till_2015_09_09.reset_index('healthCode').drop_duplicates().set_index('healthCode')
days_till_2015_09_09_180days = days_till_2015_09_09[days_till_2015_09_09.study_entry.dt.days>90]
days_till_2015_09_09_180days.columns = ['days_till_2015_09_09']
days_till_2015_09_09_180days['days_till_2015_09_09'] = 184 - days_till_2015_09_09_180days['days_till_2015_09_09'].dt.days
days_till_2015_09_09_180days.shape

#DAILY SURVEYS, AT LEAST 1 QUESTION ANSWERED
daily_weekly_df_response = select_date(daily_weekly_df, 'date', (2015,9,8))
) #restrict to data before 9/19/2015

#calculate responses per healthcode
daily_survey_response_days = daily_weekly_df_response.query('study_day < 181').dropna(subset=[daily_survey_standardqs], how='all').groupby(level=0)['value'].sum()

#join days in study
daily_survey_response_days = pd.DataFrame(daily_survey_response_days).join(days_in_study_2015_9_9, how='inner')

#calculate rate #days responding / #days enrolled
daily_survey_response_days['ratio'] = daily_survey_response_days['value'].astype(float).divide(daily_survey_response_days['days_in_study_2015_9_9'].values, axis=0)
daily_survey_response_days = daily_survey_response_days.reset_index().drop_duplicates(['healthCode', 'ratio']).set_index('healthCode')
daily_survey_response_days.head()

cohort_response_rate = daily_survey_response_days.join(aboutyou)
cohort_response_rate = cohort_response_rate.join(robust.set_index('healthC

```

```

ode')['gina'], how='inner')

cohort_response_rate['intercept'] = 1

cohort_response_rate = cohort_response_rate.join(days_till_2015_09_09_180d
ays, how='inner')

replacement_dict = {'gina':{'Uncontrolled':2, 'PartlyControlled':1, 'WellC
ontrolled':0},\
                    'education':{'1':1, '2':1, '3':1, '4':2, '5':2, '6':3,
'7':3, '8':np.NaN},\
                    'BiologicalSex':{'Female':0, 'Male':1},\
                    'health_insurance':{'3.0':0, '1.0':1, '2.0':1, '4.0':np
.NaN},}

cohort_response_rate = cohort_response_rate.dropna(subset=['health_insuran
ce'])
cohort_response_rate['education'] = cohort_response_rate['education'].asty
pe(str)
cohort_response_rate['health_insurance'] = cohort_response_rate['health_in
surance'].astype(str)
cohort_response_rate.replace(to_replace=replacement_dict, inplace=True)
cohort_response_rate['education'] = cohort_response_rate['education'].asty
pe(float)#.astype(str)

cohort_response_rate = cohort_response_rate[(cohort_response_rate.educatio
n<8) ]
cohort_response_rate = cohort_response_rate.join(study_entry_month)

#DATA NORMALIZATION
cohort_response_rate['logit_response_rate'] = cohort_response_rate['ratio'
].map(logit)
cohort_response_rate['logit_response_rate'] = cohort_response_rate['logit_
response_rate'].map(lambda x: 1.0 if x > 1.0 else x)

def std_scale(x):

    mean_x = np.mean(x)
    std_x = np.std(x)
    return [(i - mean_x) / std_x for i in x]

cohort_response_rate = cohort_response_rate[cohort_response_rate.index.isi
n(cox_df.index)]
cohort_response_rate = cohort_response_rate.join(age)
cohort_response_rate['CurrentAge_binned'] = pd.cut(cohort_response_rate['
CurrentAge'], range(0,100,10), labels=False, retbins=False)
cohort_response_rate = cohort_response_rate.join(biosex)

for c in ['education', 'age_when_diagnosed', 'gina', 'CurrentAge', 'ethnic
ity', 'Income', 'health_insurance', 'study_entry_month']:
    cohort_response_rate[c + '_stdscale'] = cohort_response_rate[[c]].appl
y(std_scale)

```

```
results = smf.ols('logit_response_rate ~ I(study_entry_month) + I(CurrentAge) + C(health_insurance) + C(ethnicity) + C(BiologicalSex) + I(education_stdscale) ', \
                  data=cohort_response_rate[cohort_response_rate.index.isin(cox_df.index)]).fit()

results.summary()
```

/Users/ers/anaconda2/lib/python2.7/site-packages/ipykernel/__main__.py:21: FutureWarning: sort is deprecated, use sort_values(inplace=True) for INPLACE sorting
/Users/ers/anaconda2/lib/python2.7/site-packages/ipykernel/__main__.py:29: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

Out[30]: OLS Regression Results

Dep. Variable:	logit_response_rate	R-squared:	0.117
Model:	OLS	Adj. R-squared:	0.107
Method:	Least Squares	F-statistic:	11.70
Date:	Sun, 29 Jan 2017	Prob (F-statistic):	2.48e-12
Time:	23:27:08	Log-Likelihood:	-835.29
No. Observations:	537	AIC:	1685.
Df Residuals:	530	BIC:	1715.
Df Model:	6		
Covariance Type:	nonrobust		

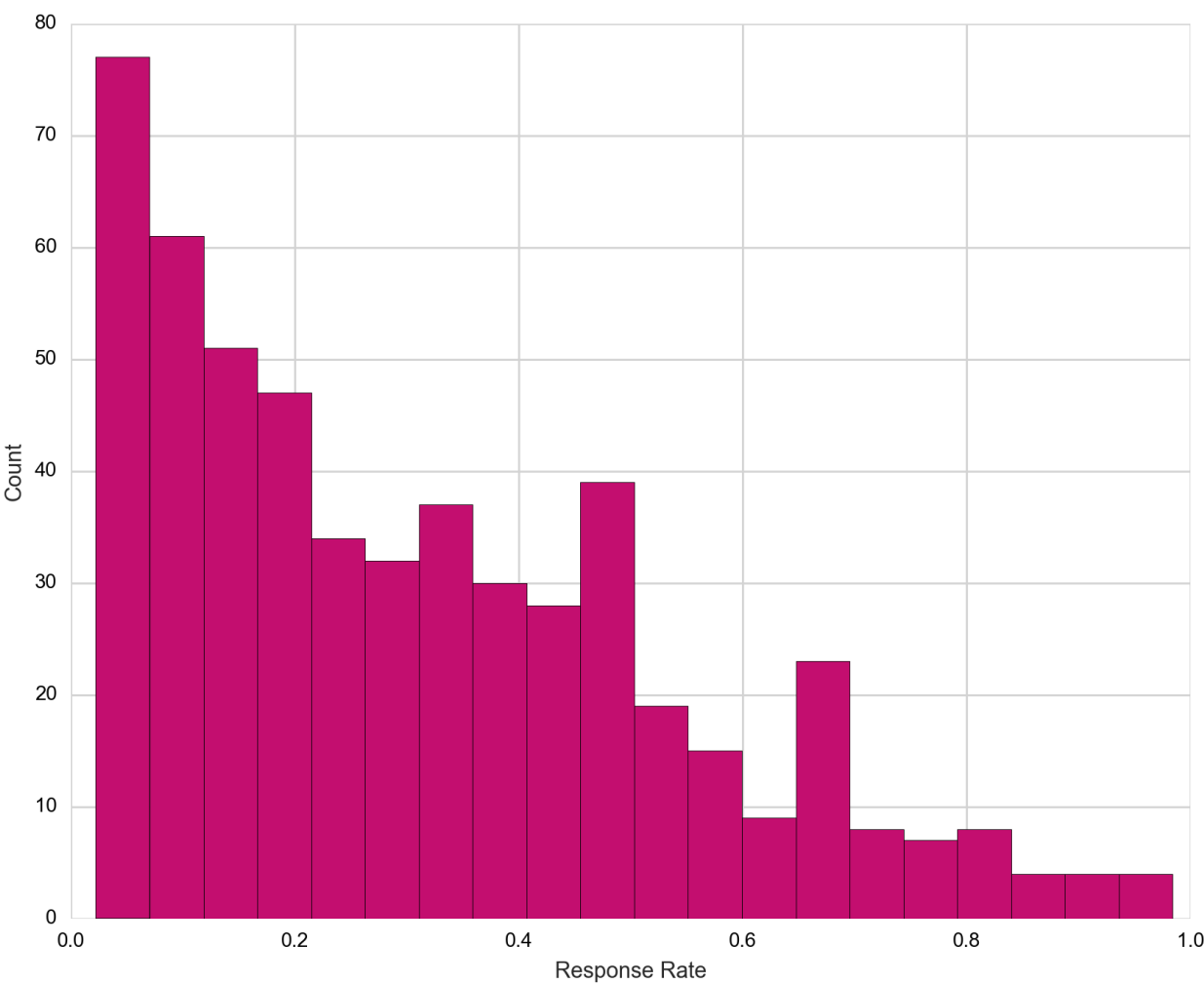
	coef	std err	t	P> t	[95.0% Conf. Int.]
Intercept	-2.1320	0.354	-6.021	0.000	-2.828 -1.436
C(health_insurance)[T.1.0]	0.6100	0.295	2.068	0.039	0.030 1.190
C(ethnicity)[T.2.0]	-0.0958	0.152	-0.631	0.528	-0.394 0.202
C(BiologicalSex)[T.Male]	-0.2541	0.104	-2.449	0.015	-0.458 -0.050
I(study_entry_month)	-0.4409	0.076	-5.802	0.000	-0.590 -0.292
I(CurrentAge)	0.0230	0.004	5.702	0.000	0.015 0.031
I(education_stdscale)	-0.0539	0.053	-1.020	0.308	-0.158 0.050

Omnibus:	18.452	Durbin-Watson:	1.993
		Jarque-Bera	

Prob(Omnibus):	0.000	(JB):	9.191
Skew:	-0.078	Prob(JB):	0.0101
Kurtosis:	2.378	Cond. No.	345.

```
In [31]: cohort_response_rate[cohort_response_rate.index.isin(cox_df.index)]['ratio']
        .hist(bins=20, color='#C30E6F', figsize=(10,8))
        plt.ylabel('Count')
        plt.xlabel('Response Rate')
```

Out[31]: <matplotlib.text.Text at 0x1209d1b10>



Univariate Model

Supplementary Table 4a

Estimated Cox proportional hazards models for predicting retention time based on univariate analysis.

[Table of Contents](#)

```
In [64]: model_summary_univariate = []
for c in cox_cols:
    print c
    if c in ['last_survey_censored', 'event_observed']: continue
    cf.fit( cox_df[['last_survey_censored', 'event_observed', c]], \
            'last_survey_censored', event_col='event_observed', \
            )

    model_summary = cf.summary
    model_summary.sort(columns=['p'], inplace=True)
    model_summary.index = [i.replace('_', ' ').upper().replace('BIOLOGICAL SEX', 'BIOLOGICAL SEX').lower() for i in model_summary.index]
    model_summary['lower 0.95'] = model_summary['lower 0.95'].map(np.exp)
    model_summary['upper 0.95'] = model_summary['upper 0.95'].map(np.exp)

    model_summary_univariate.append( model_summary)

pd.set_option('display.precision',3)
univariate_models = pd.concat(model_summary_univariate)
univariate_models['abs_z'] = univariate_models['z'].map(np.abs)
univariate_models.sort(columns=['abs_z'], ascending=False, inplace=True)
del univariate_models['abs_z']
univariate_models = univariate_models.rename(index={'currentage':'current age'})
univariate_models
```

gina
last_survey_censored
event_observed
education
age_when_diagnosed
Income
health_insurance

/Users/ers/anaconda2/lib/python2.7/site-packages/ipykernel/__main__.py:8:
FutureWarning: sort(columns=....) is deprecated, use sort_values(by=.....)

ethnicity
CurrentAge
BiologicalSex
study_entry_month

/Users/ers/anaconda2/lib/python2.7/site-packages/ipykernel/__main__.py:18:
FutureWarning: sort(columns=....) is deprecated, use sort_values(by=.....
)

Out[64]:

	coef	exp(coef)	se(coef)	z	p	lower 0.95	upper 0.95
study entry month	0.637	1.891	0.075	8.523	1.559e-17	1.633	2.189
current age	-0.020	0.980	0.004	-4.907	9.230e-07	0.972	0.988
age when diagnosed	-0.009	0.991	0.004	-2.423	1.541e-02	0.984	0.998
ethnicity	-0.293	0.746	0.144	-2.029	4.243e-02	0.562	0.990

income	-0.089	0.915	0.045	-1.995	4.606e-02	0.838	0.998
health insurance	-0.462	0.630	0.272	-1.694	9.025e-02	0.369	1.075
education	-0.131	0.877	0.083	-1.586	1.128e-01	0.746	1.031
gina	0.083	1.086	0.072	1.153	2.488e-01	0.944	1.250
biological sex	-0.017	0.983	0.103	-0.164	8.694e-01	0.803	1.204

Multivariate Cox PH Model

Supplemental Table 4b

[Table of Contents](#)

```
In [51]: #Multivariate Cox model
# outcome measurement: number of days until last daily survey completion

pd.set_option('display.precision',4)
m_summary.index = [i.replace('_', ' ').replace('CurrentAge', 'current age')
).replace('BiologicalSex', 'biological sex').lower() for i in m_summary.in
dex]
m_summary['lower 0.95'] = m_summary['lower 0.95'].apply(np.exp)
m_summary['upper 0.95'] = m_summary['upper 0.95'].apply(np.exp)

m_summary
```

Out[51]:

	coef	exp(coef)	se(coef)	z	p	lower 0.95	upper 0.95
study entry month	0.6976	2.0089	0.0762	9.1562	5.3751e-20	281.9345	29840.2139
current age	-0.0226	0.9776	0.0046	-4.9031	9.4336e-07	13.9411	14.6123
health insurance	-0.4605	0.6310	0.2816	-1.6355	1.0194e-01	4.2126	19.9167
ethnicity	-0.2060	0.8138	0.1464	-1.4075	1.5928e-01	6.3092	19.2461
biological sex	0.0996	1.1048	0.1077	0.9250	3.5497e-01	11.5421	50.1001
age when diagnosed	-0.0037	0.9963	0.0041	-0.8985	3.6890e-01	14.6815	15.3366
gina	0.0609	1.0628	0.0733	0.8305	4.0623e-01	12.3108	30.3105

education	0.0471	1.0483	0.0889	0.5301	5.9603e-01	11.1591	32.5612
income	-0.0139	0.9862	0.0489	-0.2849	7.7571e-01	11.5836	19.3139

Ordinary Least Squares Model

Supplemental Table 4c

[Table of Contents](#)

In [54]: results.summary()

Out[54]: OLS Regression Results

Dep. Variable:	logit_response_rate	R-squared:	0.117
Model:	OLS	Adj. R-squared:	0.107
Method:	Least Squares	F-statistic:	11.70
Date:	Fri, 27 Jan 2017	Prob (F-statistic):	2.48e-12
Time:	14:37:23	Log-Likelihood:	-835.29
No. Observations:	537	AIC:	1685.
Df Residuals:	530	BIC:	1715.
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[95.0% Conf. Int.]
Intercept	-2.1239	0.352	-6.026	0.000	-2.816 -1.431
C(health_insurance)[T.1.0]	0.6100	0.295	2.068	0.039	0.030 1.190
C(ethnicity)[T.2.0]	-0.0958	0.152	-0.631	0.528	-0.394 0.202
C(BiologicalSex)[T.Male]	-0.2541	0.104	-2.449	0.015	-0.458 -0.050
I(study_entry_month)	-0.4409	0.076	-5.802	0.000	-0.590 -0.292
I(CurrentAge)	0.0230	0.004	5.702	0.000	0.015 0.031
I(education_stdscale)	-0.0564	0.055	-1.020	0.308	-0.165 0.052

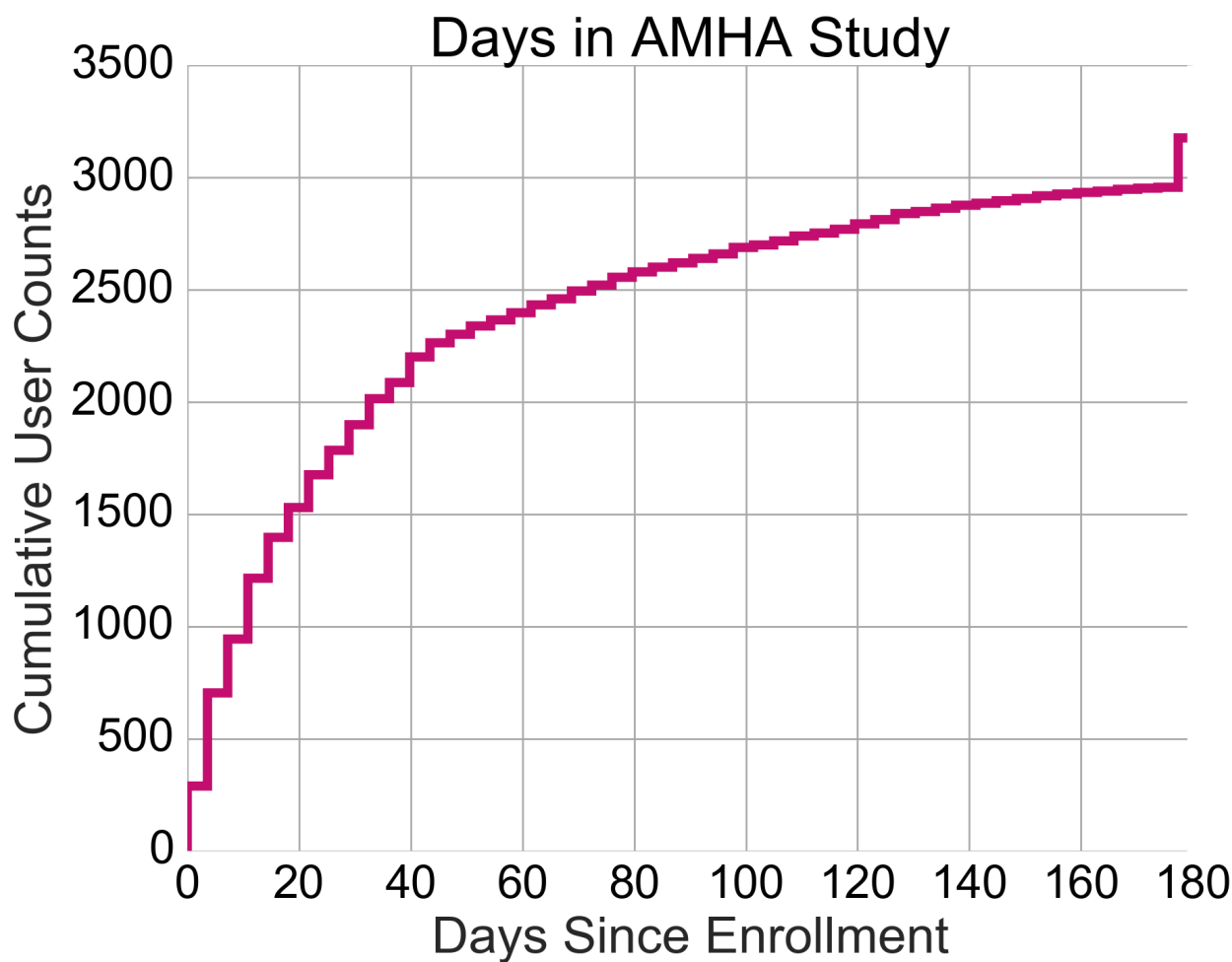
Omnibus:	18.452	Durbin-Watson:	1.993
----------	--------	----------------	-------

Prob(Omnibus):	0.000	Jarque-Bera (JB):	9.191
Skew:	-0.078	Prob(JB):	0.0101
Kurtosis:	2.378	Cond. No.	344.

CDF of Days in AMHA Study

```
In [55]: last_study_day = daily_weekly_df[daily_survey_standardqs].dropna(how='all')
        last_study_day = last_study_day.groupby('healthCode')['study_day'].max()
        last_study_day = last_study_day.map(lambda x: 181 if x > 179 else x)
        last_study_day.name = 'last_survey_censored'

        last_study_day.ix[baseline.healthCode].hist(bins=50, histtype='step', cumulative=True, linewidth=5, color='#C30E6F', figsize=(10,8))
        #plt.xlabel('Enrollment Day')
        plt.ylabel('Cumulative User Counts', rotation=90)
        plt.xlabel('Days Since Enrollment')
        plt.grid(color='darkgray')
        plt.title('Days in AMHA Study')
        plt.xlim(0,180)
        plt.vlines(x=180,ymin=0,ymax=3500, color='white', linewidth=6)
        plt.tight_layout()
        #plt.savefig('/Users/ers/Dropbox/feasibility_figures_ERS/CumulativeHist_Retention.svg',dpi=600, format='svg')
        plt.show()
```



```
In [56]: cox_df.last_survey_censored.hist(bins=50, histtype='step', linewidth=5, color='#C30E6F', figsize=(16,5))
plt.ylabel('Participant Count', rotation=90)
plt.grid(color='darkgray')
plt.title('Days in AMHA Study')
plt.tight_layout()
```

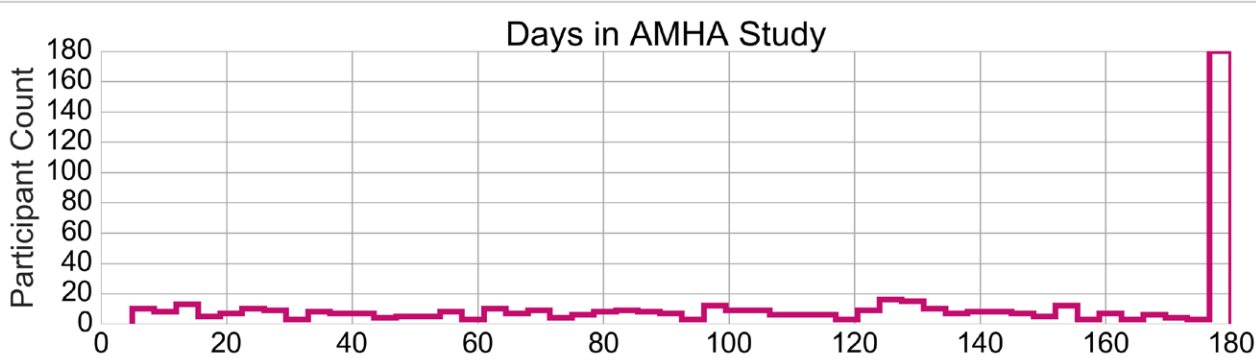


Figure 2d - Enrollment and retention over time for Robust users By Study Entry Month

'Table of Contents'

['cox_df used below created here'](#)

```
In [34]: #Baseline
#Kaplan Meier Curve
from lifelines import KaplanMeierFitter
# kmf_control = KaplanMeierFitter()
# kmf_control.fit(last_daily.last_daily.values, label='Asthma App').plot()

#Baseline
from lifelines import KaplanMeierFitter
from lifelines.plotting import add_at_risk_counts

rcParams['text.color'] = 'black'
rcParams['ytick.color'] = 'black'
rcParams['xtick.color'] = 'black'
rcParams['axes.facecolor'] = 'white'
rcParams['axes.edgecolor'] = 'white'

fig, ax = plt.subplots(1,1, figsize=(12,12))
fig.patch.set_facecolor('white')

kmfs = []
row_counter = 0
col_counter = 0
for i,n in cox_df[cox_df.index.isin(robust.healthCode)].groupby(['study_entry_month']):

    kmf_control = KaplanMeierFitter()
    kmf_df = n
    label_dict = {0:'Study Month 1', 1:'Study Month 2', 2:'Study Month 3',
3:'Study Month 4'}
    label_i = label_dict[i]

    kmf_control.fit(kmf_df.last_survey_censored.values, event_observed=kmf_df.event_observed, label=label_i)

    kmf_control.plot(label=i, ax=ax, cmap='jet', show_censors=True)
    kmfs.append(kmf_df)
    ax.set_axis_bgcolor('white')
    ax.set_title('Study Participation by Study Entry Month' )

    ax.set_xlabel('Enrollment Day', color='black')

    ax.set_ylabel('Survival', color='black')

plt.legend(fontsize='xx-large')
plt.tight_layout()
plt.yticks(arange(0,1.1,.1))
plt.savefig('../analysis/feasibility_figures_ERS/KM_by_study.svg', format=
```

```
'svg', dpi=600)
plt.show()
```

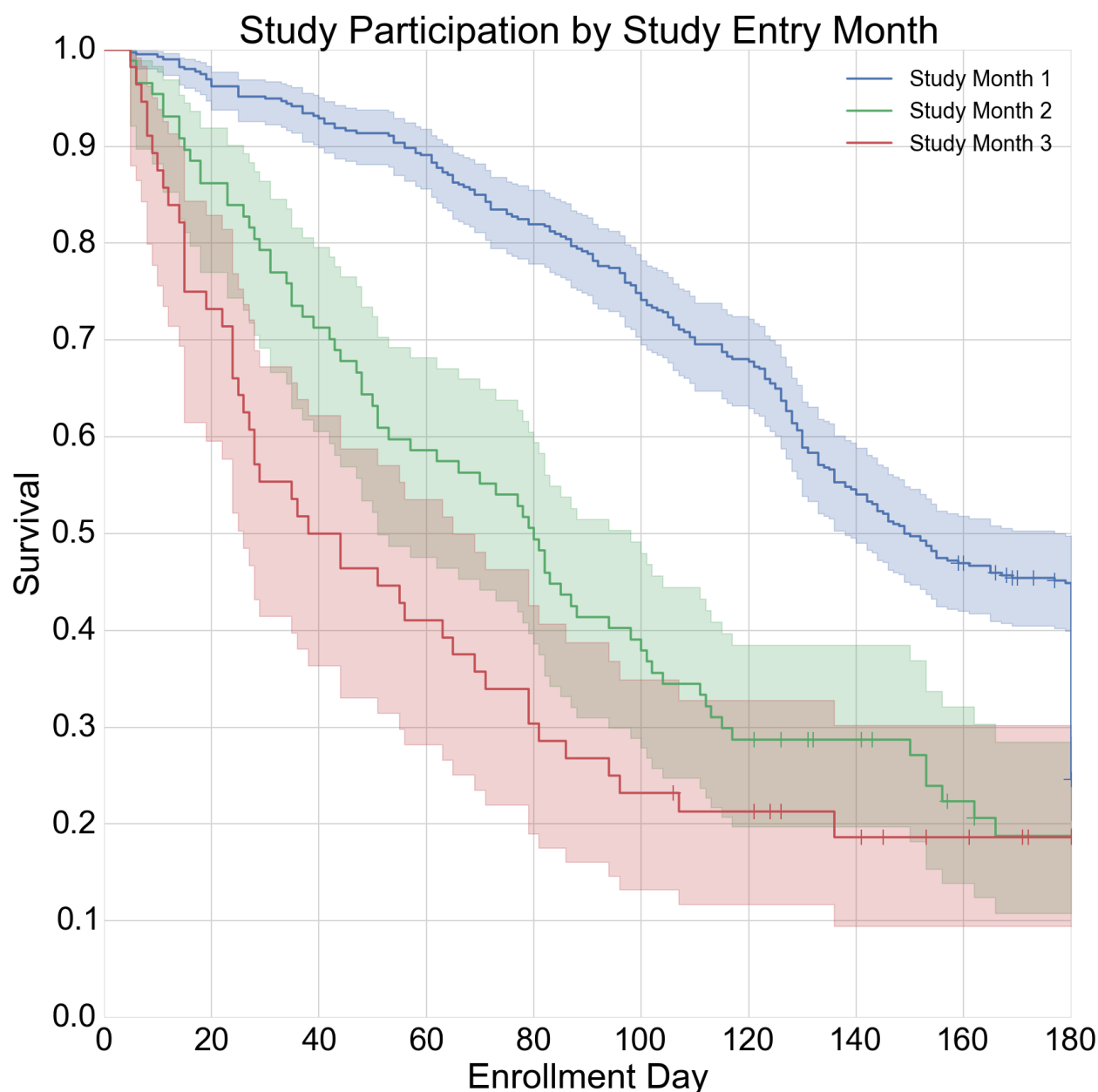


Figure 2e Enrollment and retention over time for Robust users By Age

[Table of Contents](#)

```
In [64]: #Baseline
#Kaplan Meier Curve
from lifelines import KaplanMeierFitter

#Baseline
from lifelines import KaplanMeierFitter
```

```

from lifelines.plotting import add_at_risk_counts

rcParams['text.color'] = 'black'
rcParams['ytick.color'] = 'black'
rcParams['xtick.color'] = 'black'
rcParams['axes.facecolor'] = 'white'
rcParams['axes.edgecolor'] = 'white'

fig, ax = plt.subplots(1,1, figsize=(12,12))
fig.patch.set_facecolor('white')

kmfs = []
row_counter = 0
col_counter = 0

cox_df_age = cox_df.copy()

cox_df_age['Current Age binned'] = pd.cut(cox_df['CurrentAge'], [0,40,100]
, labels=False, retbins=False)
for i,n in cox_df_age[cox_df_age.index.isin(baseline.healthCode)].groupby(
['Current Age binned']):

    print i
    if n.shape[0] < 10: continue
    kmf_control = KaplanMeierFitter()
    kmf_df = n

    if i ==0:
        label_i = '18-40 years, n=' + str(n.shape[0])
    else:
        label_i = '>40 years, n=' + str(n.shape[0])
    kmf_control.fit(kmf_df.last_survey_censored.values, event_observed=kmf
_df.event_observed, label=label_i)

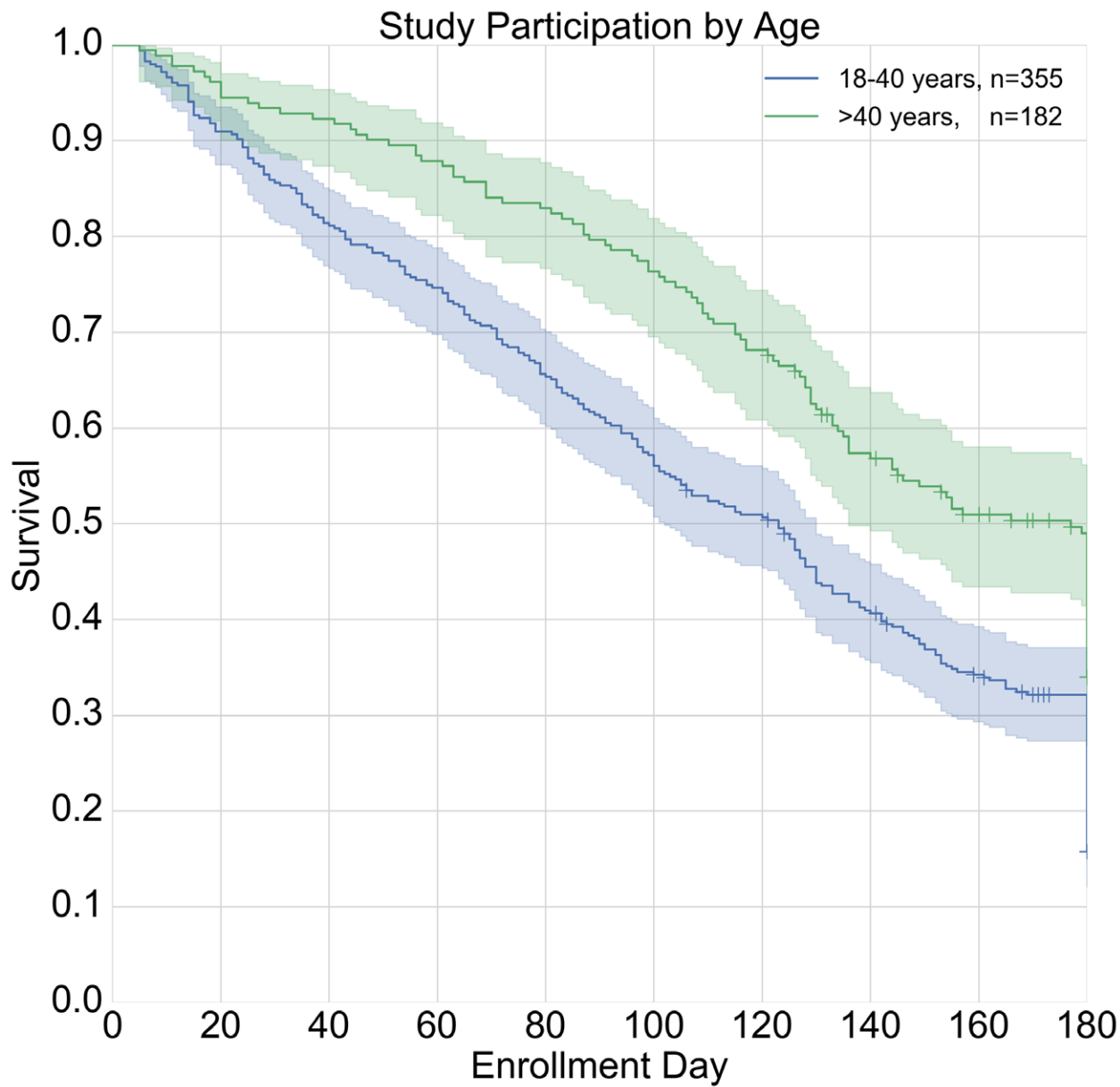
    kmf_control.plot(label=i, ax=ax, cmap='jet', show_censors=True, ci_sho
w=True)
    kmfs.append(kmf_df)
    ax.set_axis_bgcolor('white')
    ax.set_title('Study Participation by Age', fontsize=30)
    ax.set_ylabel('Survival', fontsize=30)
    ax.set_xlabel('Enrollment Day', fontsize=30)

    ax.set_xlabel('Enrollment Day', color='black')

    ax.set_ylabel('Survival', color='black')
plt.legend(fontsize='xx-large')
plt.tight_layout()
plt.yticks(arange(0,1.1,.1), fontsize=30)
plt.xticks(range(0,200, 20), fontsize=30)
plt.show()

```

1



Heatmaps

Supplemental Figure 3

[Table of Contents](#)

Symptom Heatmaps

day_symptoms, night_symptoms, use_qr

```

In [65]: rcParams['text.color'] = 'black'
         rcParams['ytick.color'] = 'black'
         rcParams['xtick.color'] = 'black'

from matplotlib.colors import LinearSegmentedColormap
vmax = 2.0
cmap = LinearSegmentedColormap.from_list('mycmap', [(0 / vmax, 'black'),
                                                    (1 / vmax, 'yellow'),
                                                    (2 / vmax, 'red')])

def heatmap_plot(heatmap_df, title, ylabel=True):
    fig, ax = plt.subplots(1,1, figsize=(15,8))

    heatmap_df = heatmap_df[range(0,180)]

    #Order participants by daily survey completion counts
    heatmap_df_order = heatmap_df.apply(lambda x: sum([1 if i in set([0,1]
) else 0 for i in x]), axis=1)
    heatmap_df_order.sort(inplace=True, ascending=False)
    heatmap_df_order

    p = ax.pcolormesh(heatmap_df.ix[heatmap_df_order.index].values, cmap=c
map)

    cbar = fig.colorbar(p, aspect=10, shrink=0.8, ticks=[-1,0,1])
    cbar.set_ticklabels([''])

    ax.set_xlabel('Days Post-Enrollment ', color='black')
    ax.set_axis_bgcolor('black')

    if ylabel:
        ax.set_ylabel('Participant')

    plt.show()

ids_180days_of_study = days_till_2015_09_09[days_till_2015_09_09>179].inde
x

for n,symptom in enumerate(['day_symptoms', 'night_symptoms', 'use_qr']):

    symptom_title = symptom.replace('_', ' ').upper()
    print symptom_title

    if symptom != 'medicine':
        heatmap_df = daily_weekly_df[symptom].map(sum).reset_index().pivot
_table(index='healthCode', columns='study_day', values=symptom)

    else:

```

```

heatmap_df = daily_weekly_df[[symptom]]
heatmap_df[symptom] = heatmap_df[symptom].map(med_multi_parse)
heatmap_df = heatmap_df.reset_index().pivot_table(index='healthCode', columns='study_day', values=symptom)

#LIMITS PLOTS TO PEOPLE IN STUDY FOR AT LEAST 180 DAYS BY 9/9/2015
heatmap_df = heatmap_df[heatmap_df.index.isin(ids_180days_of_study)]

wellcontrolled = heatmap_df[heatmap_df.index.isin(baseline[baseline.gina=='WellControlled']['healthCode'].values)]
heatmap_plot(wellcontrolled, 'Well Controlled, n='+ str(wellcontrolled.shape[0]), True)

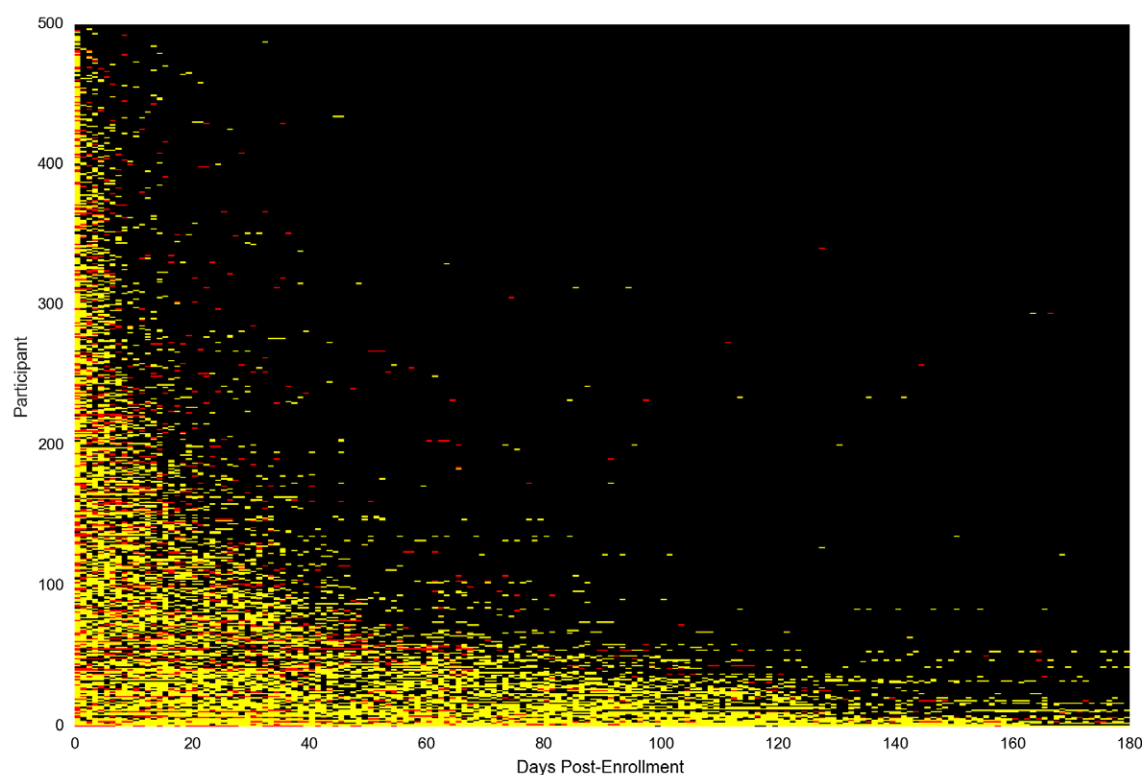
partlycontrolled = heatmap_df[heatmap_df.index.isin(baseline[baseline.gina=='PartlyControlled']['healthCode'].values)]
heatmap_plot(partlycontrolled, 'Partly Controlled, n=' + str(partlycontrolled.shape[0]), False)

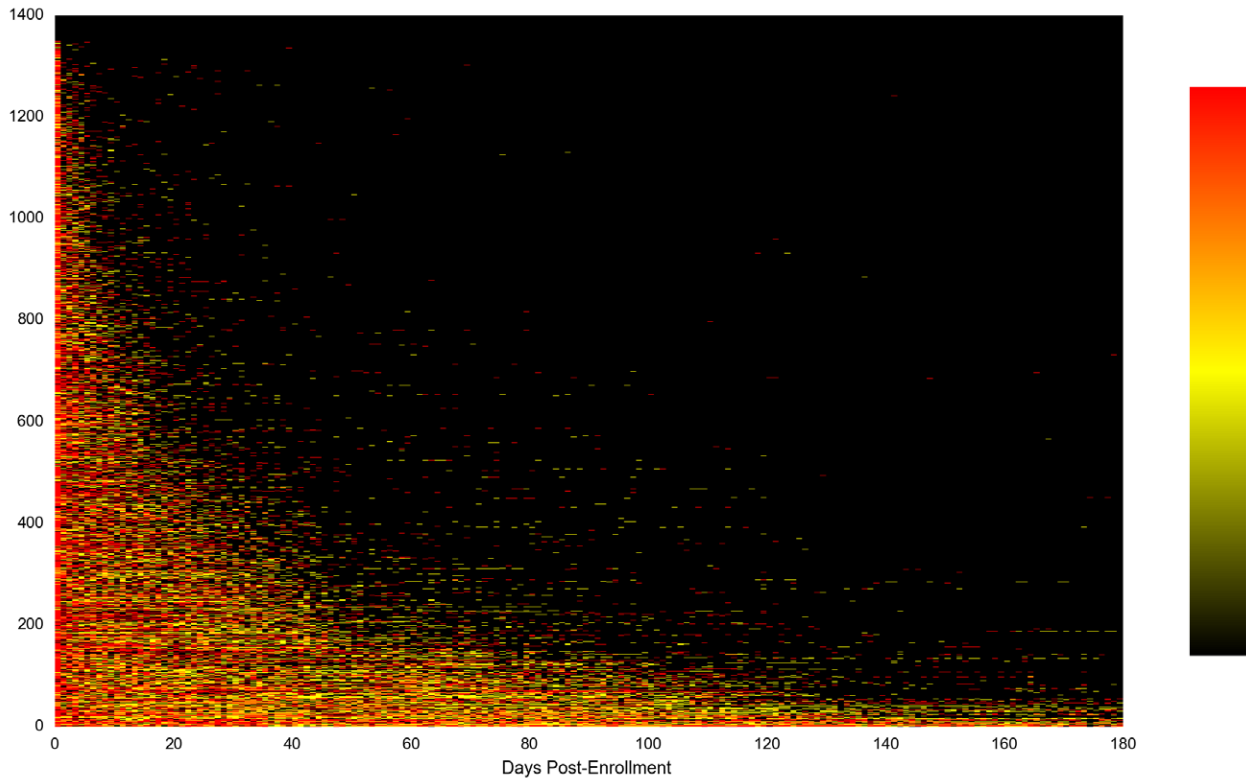
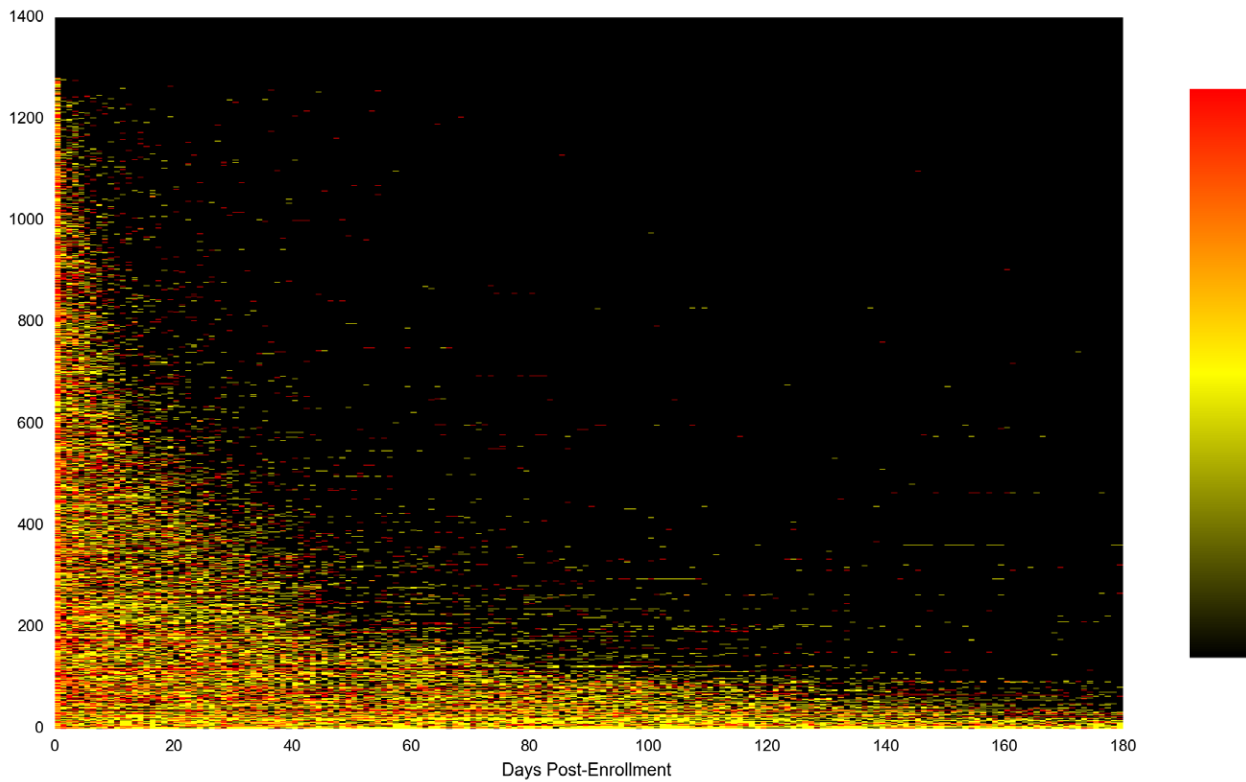
uncontrolled = heatmap_df[heatmap_df.index.isin(baseline[baseline.gina=='Uncontrolled']['healthCode'].values)]
heatmap_plot(uncontrolled, 'Uncontrolled, n=' + str(uncontrolled.shape[0]), False)

```

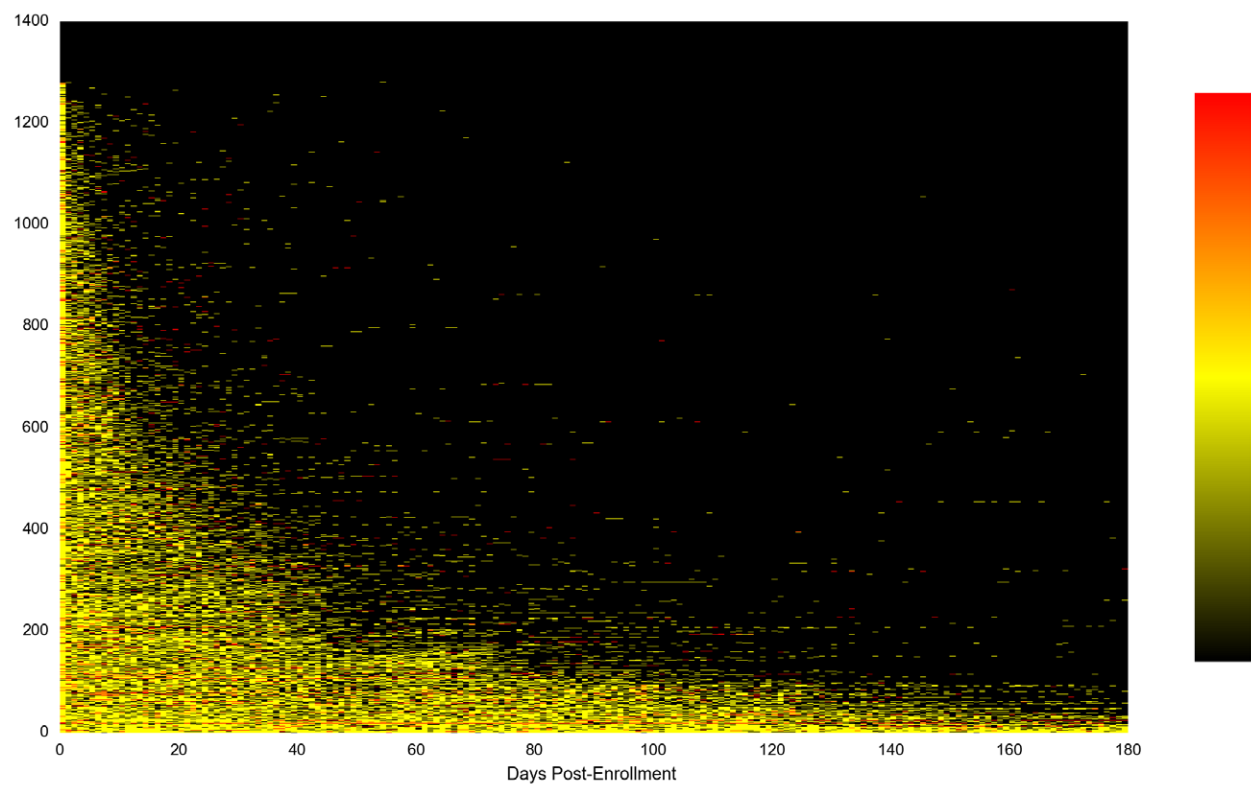
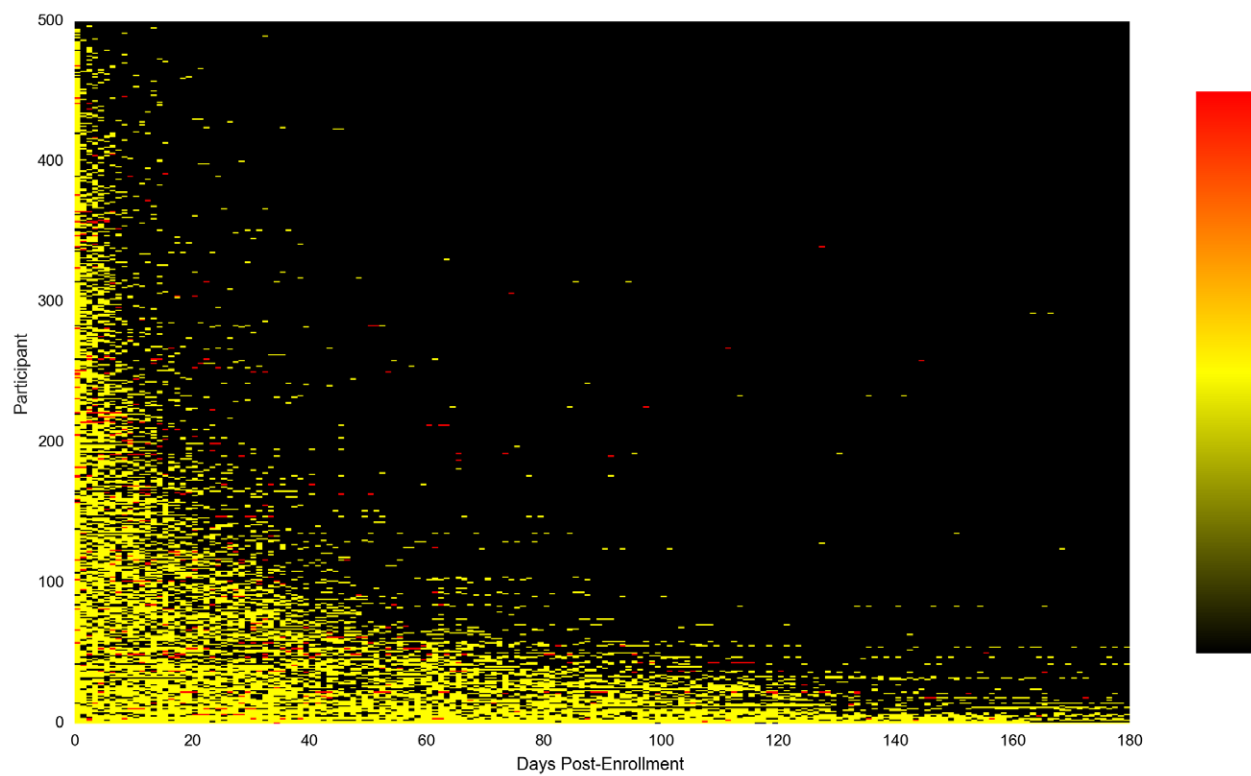
DAY SYMPTOMS

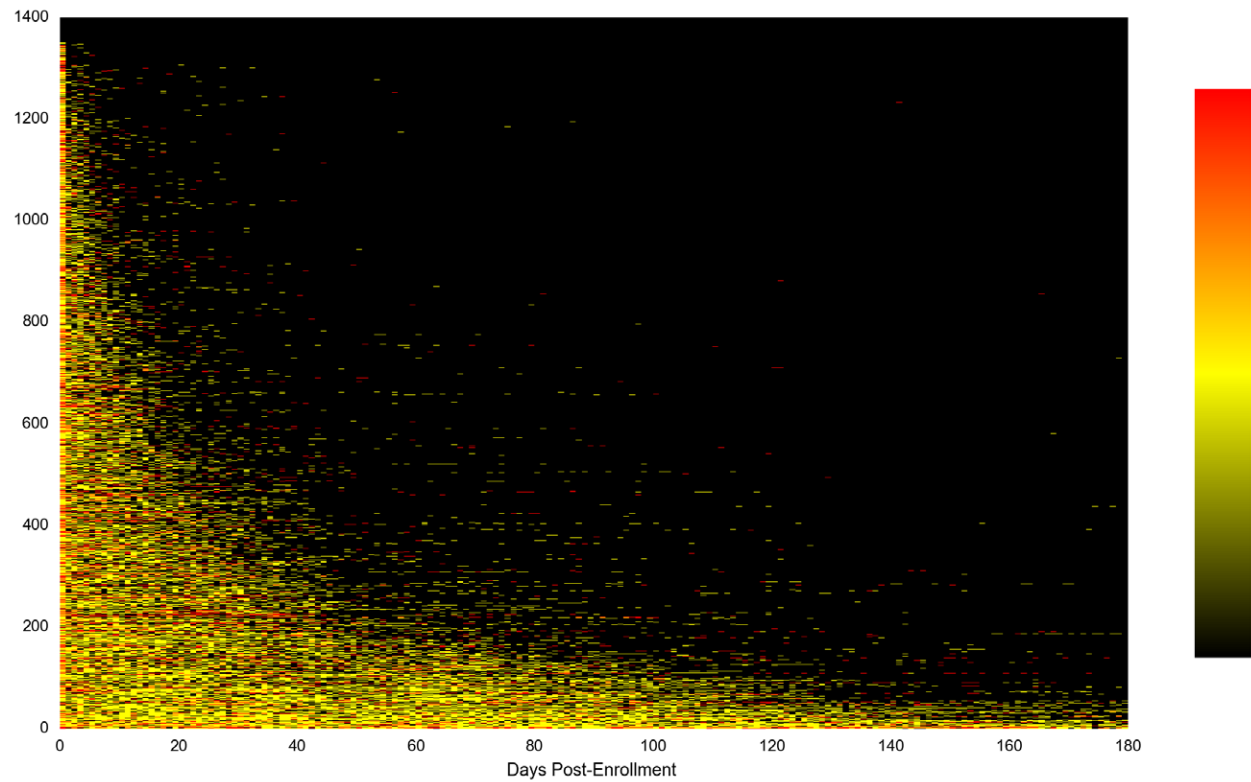
/Users/ers/anaconda2/lib/python2.7/site-packages/ipykernel/__main__.py:22:
FutureWarning: sort is deprecated, use sort_values(inplace=True) for INPLACE sorting



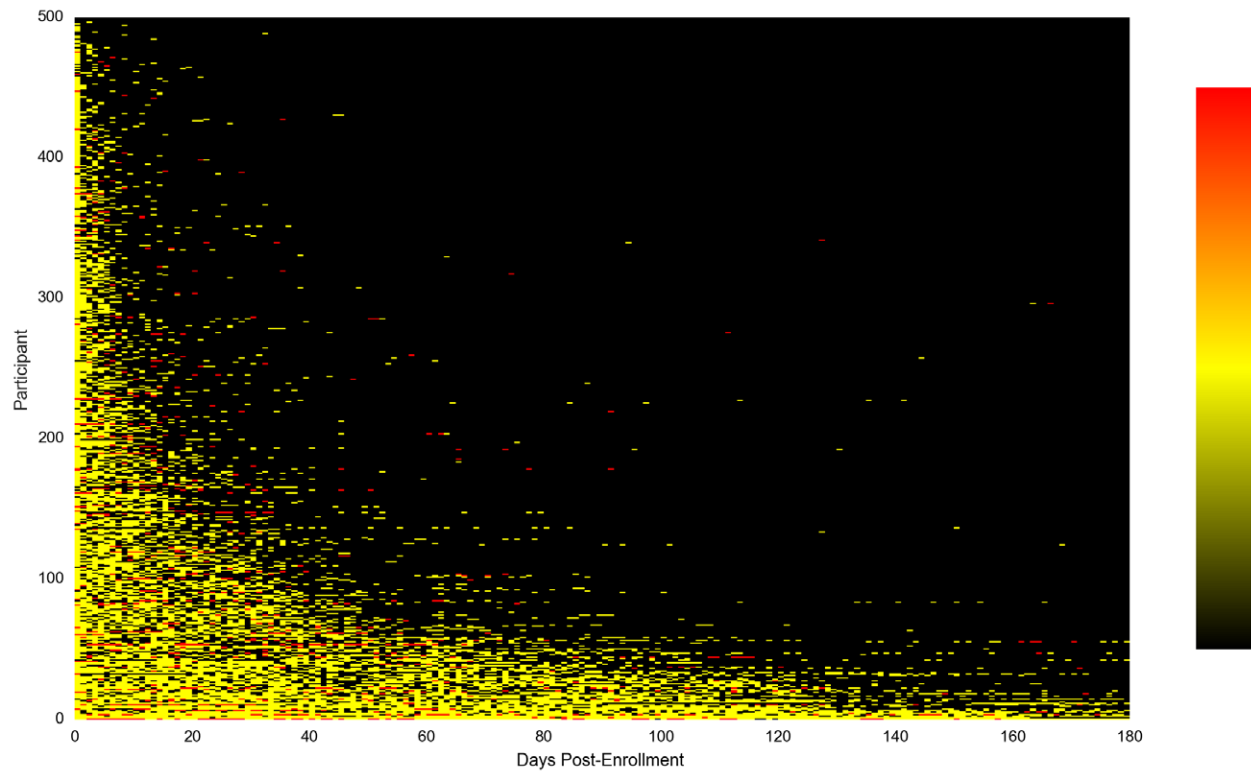


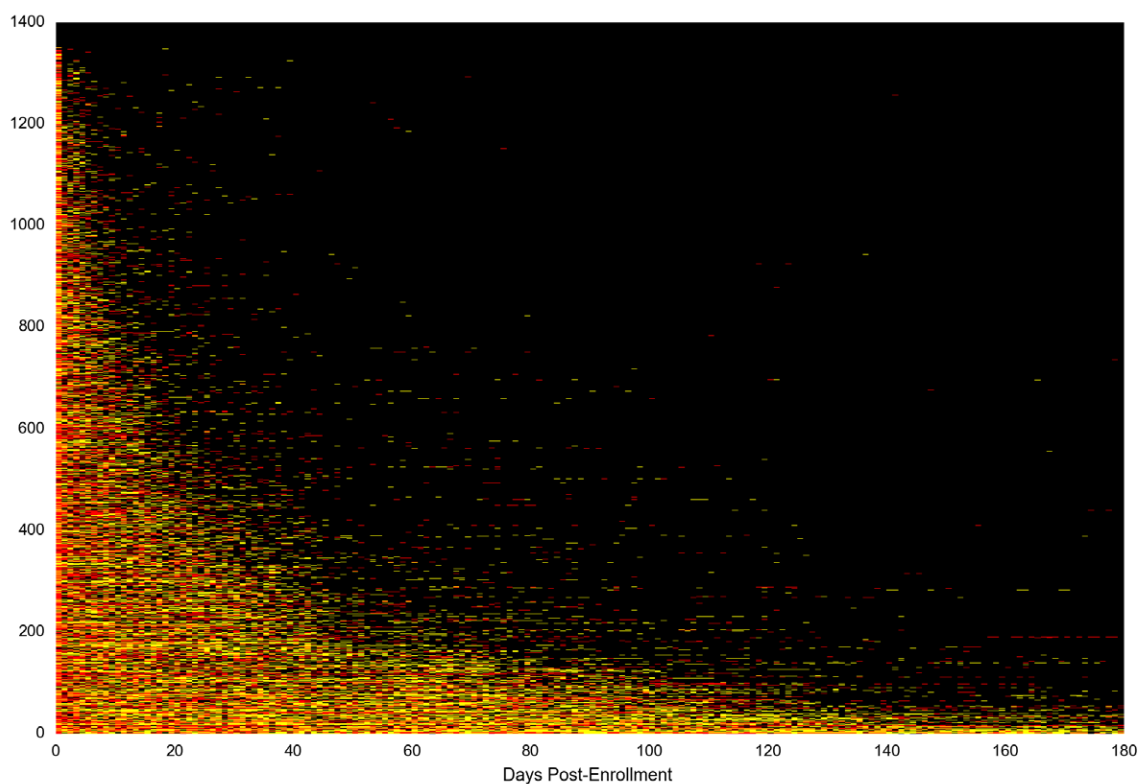
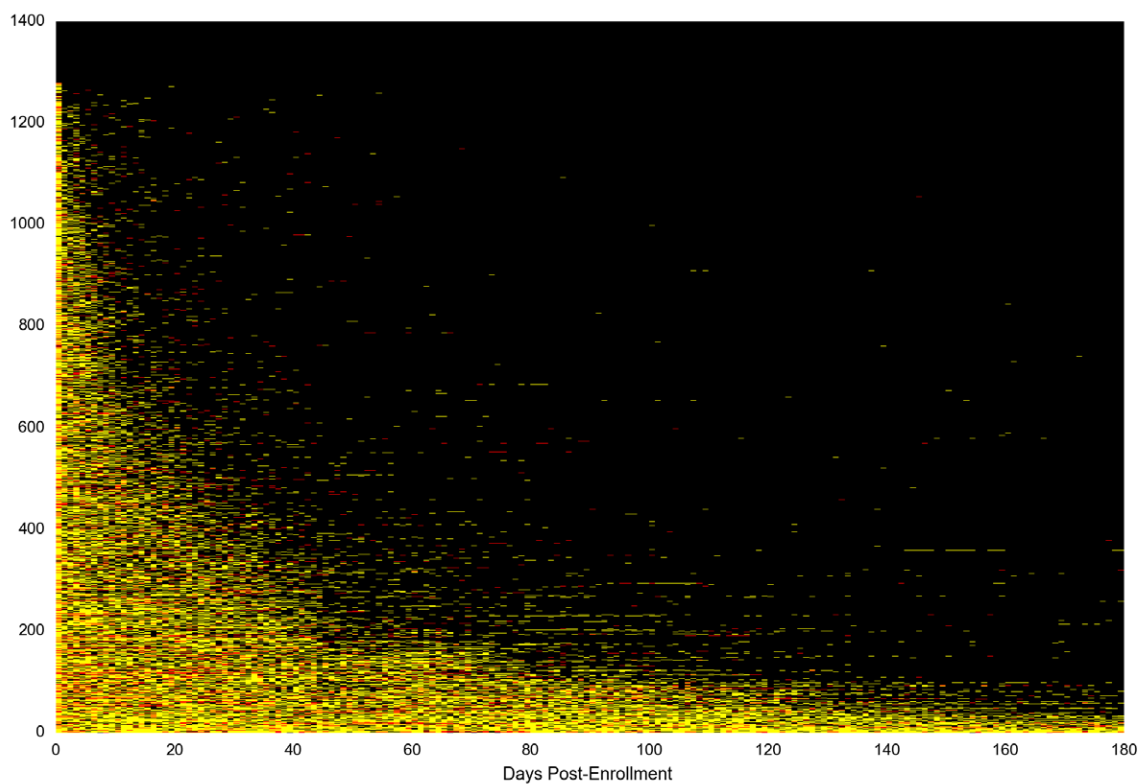
NIGHT SYMPTOMS





USE QR





Daily Controller Medication Usage Heatmap

In [37]: *#DAILY CONTROLLER MEDICATION USAGE HEATMAP*

```
rcParams['text.color'] = 'black'
rcParams['ytick.color'] = 'black'
```

```
rcParams['xtick.color'] = 'black'

from matplotlib.colors import LinearSegmentedColormap
vmax = 3.0
cmap = LinearSegmentedColormap.from_list('mycmap', [(0 / vmax, 'black'),
                                                    (1 / vmax, 'yellow'),
                                                    (2 / vmax, 'yellow'),
                                                    (3 / vmax, 'red')])

)

def heatmap_plot(heatmap_df, title, ylabel=True):
    fig, ax = plt.subplots(1,1, figsize=(15,8))

    heatmap_df = heatmap_df[range(0,180)]

    #Order participants by daily survey completion counts

    heatmap_df_order = heatmap_df.apply(np.sum, axis=1)
    heatmap_df_order.sort(inplace=True, ascending=False)
    heatmap_df_order

    p = ax.pcolormesh(heatmap_df.ix[heatmap_df_order.index].values, cmap=c
map)

    cbar = fig.colorbar(p, aspect=10, shrink=0.8, ticks=[0, 1, 2, 3])
    cbar.set_ticklabels([''])

    ax.set_axis_bgcolor('black')

    if ylabel:
        ax.set_ylabel('Participant')

    plt.show()

def med_multi_parse(x):
    if ',' in str(x):
        x = map(float, x.split(','))
        return min(x)
    return float(x)

ids_180days_of_study = days_till_2015_09_09[days_till_2015_09_09>179].inde
x

for n,symptom in enumerate(['medicine']):

    symptom_title = symptom.replace('_', ' ').upper()
    print symptom_title
```

```

    if symptom != 'medicine':
        heatmap_df = daily_weekly_df[symptom].map(sum).reset_index().pivot_
_table(index='healthCode', columns='study_day', values=symptom)

    else:
        heatmap_df = daily_weekly_df[[symptom]]
        heatmap_df[symptom] = heatmap_df[symptom].map(med_multi_parse)
        heatmap_df = heatmap_df[heatmap_df[symptom] != 4]
        heatmap_df = heatmap_df.reset_index().pivot_table(index='healthCod
e', columns='study_day', values=symptom)

    #LIMITS PLOTS TO PEOPLE IN STUDY FOR AT LEAST 180 DAYS BY 9/9/2015

    heatmap_df = heatmap_df.fillna(0)

    wellcontrolled = heatmap_df[heatmap_df.index.isin(baseline[baseline.gi
na=='WellControlled']['healthCode'].values)]
    heatmap_plot(wellcontrolled, 'Well Controlled, n='+ str(wellcontrolle
d.shape[0]), True)

    partlycontrolled = heatmap_df[heatmap_df.index.isin(baseline[baseline.
gina=='PartlyControlled']['healthCode'].values)]
    heatmap_plot(partlycontrolled, 'Partly Controlled, n=' + str(partlyco
ntrolled.shape[0]), False)

    uncontrolled = heatmap_df[heatmap_df.index.isin(baseline[baseline.gina
=='Uncontrolled']['healthCode'].values)]
    heatmap_plot(uncontrolled, 'Uncontrolled, n=' + str(uncontrolled.shap
e[0]), False)

```

/Users/ers/anaconda/lib/python2.7/site-packages/ipykernel/__main__.py:76:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: <http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy>

/Users/ers/anaconda/lib/python2.7/site-packages/ipykernel/__main__.py:26:

FutureWarning: sort is deprecated, use sort_values(inplace=True) for for I
NPLACE sorting

MEDICINE

