Predicting multi-day dips in mood with smartphone data

Overview and Motivation

Technology has often enabled and quickened scientific innovation, and the rapid and widespread adoption of smartphones has that same potential. 2014 marked the first year in which there were more smartphone subscriptions globally than people (1). We use smartphones for most every facet of our lives and, by consequence, our phones are rich repositories of quantified social and behavioral information ripe for analysis and, potentially, clinical translation. Unsurprisingly, leveraging behavioral/psychological insights from smartphone data is a field growing as rapidly as the popularity of smartphones.

The wealth of research and commercial phone applications coming out of this burgeoning field are premised upon an exciting hypothetical: Patient X would have had adverse psychological event Y, but, by the saving grace of a smartphone, Patient X avoided Y. As just one example, several studies offer evidence that smartphones could be used to prevent schizophrenic relapse. These studies showed that longitudinal data streams — many collectedly passively (without user input) — can be used to infer measures that are predictive of relapse, like how much a person sleeps, moves and socializes (2,3). Such a prediction system has a compelling clinical upshot: With a means to forsee an impending relapse, treatments — perhaps built into the smartphone itself — can be administered to steer patient X away from that dangerous cliff.

The excitement and commercial interest this idea is generating, however, glosses over the many-layered conceptual scaffolding and copious technical detail necessary to make the jump from smartphone sensors to just-in-time intervention. This jump requires many tricky steps, of which are few major ones are: conceptualize and fruitfully define an adverse psychological event, identify variables that are predictive but also easy to collect over long timescales, and develop proper inference methods to deal with missing data, person-to-person variability and many other sources of technical snags. Not even the bottom-most layer of this rickety structure, defining an adverse psychological event, is firmly established. Indeed, the National Institute of Mental Health (NIMH) recently instituted a great deal of malleability and uncertainty in how mental disorders and symptoms are conceptualized. Within NIMH's new framework, mental disorders and their corresponding behaviors and symptoms are now not categorical but fluid, spectral-like constructs highly subject to revision if research warrants it (5).

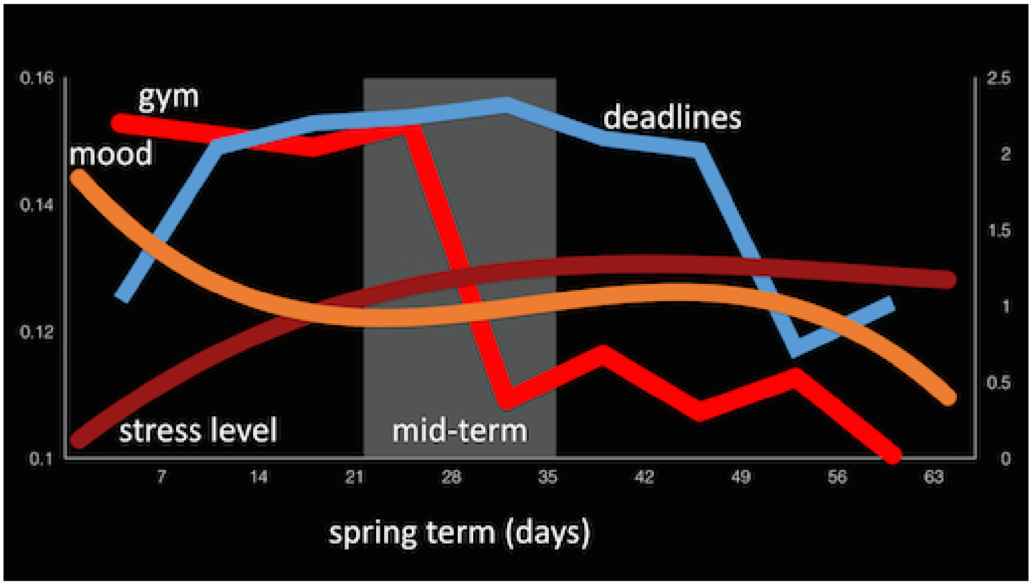
To understand the rickety scaffolding underlying this exciting field, we build this scaffolding from start to finish — from conceptualizing and defining an adverse psychological event to predicting it. We do this by completing an analysis parallel to one done by a top research group in this field. Barnett et al. (3), mentioned previously, used anomaly detection to predict schizophrenic relapse. In a parallel fashion, we use anomaly detection to predict a different adverse psychological event. We predict what we call "mood dips" — multi-day dips below an individual's typical self-reported happiness level.

Related Work

Our work was informed by several lines of research.

Principally, our work was strongly shaped by the aforementioned 2018 paper by Barnett et al. (3). We replicate or use similar versions of their methods and analyses. Like Barnett et al., we used a statistics-based (as opposed to machine learning-based) anomaly detection algorithm. We use the anomaly detection algorithm contained in an R package created by Filzmoser et al., the same group Barnett et al. cite as the inspiration for their algorithm (6). The difference between the two algorithms is small, with Filzmoser's slightly altered to better identify clusters of anomalies instead of only isolated anomalies. Also like Barnett et al., we use an imputation method that uses a weighted average of data points in close temporal proximity to the missing time points. How we organized our features for our anomaly detection algorithm also mirrors Barnett et al. To allow their results to be clinically meaningful, they grouped features into conglomerate categories. In this schema, for example, all sociability-related features are grouped, such that the anomaly detection algorithm produces a daily measure of how unusual a person's social activities were relative to what is typical for that individual. We grouped our features in a parallel manner. The analyses with which we assess the accuracy of our prediction algorithm also matches that of Barnett et al. Like them, we compare the rates of detected anomalies between time periods preceding vs. not preceding the psychological event of interest. Lastly, our shiny app is much like Barnett et al.'s Figure 1, but extended to our full dataset instead of a subset and demonstrating each step in our data pipeline, rather than just the anomaly detection step.

Our work was also informed by the research of the Dartmouth-based group that produced our dataset. The data is largely pre-processed (inferred daily summary measures) smartphone data tracking the social and behavioral trends of Dartmouth students over one semester. The results of the Dartmouth-based group demonstrated that student mood — both individually and across students — varied over the course of the semester. Their work also demonstrated that many behavioral changes coincided with those variations in mood (7,8,9,10,11).



Caption: Behavioral trends averaged across students over the course of the semester, from the Dartmouth-based research group (11).

These results inspired our decision to conceptualize and define mood dips as the adverse psychological event we sought to predict. This work also informed our choices for what features to use for anomaly detection— i.e. their work suggested some behavioral features that might be predictive of changes in mood.

Initial Questions

- Performing an analysis parallel to a paper that constitutes the current gold standard is a prime opportunity to explore the following: Within the many conceptual/technical layers underpinning this sort of work, what are the weak points? What conceptual underpinnings are questionable? Where is technical competence lacking or, per current standards, shoddily executed with no retribution?
- Within our available data streams, what adverse psychological event could we identify and/or justifiably conceptualize?
- Can we predict our chosen adverse psychological event?

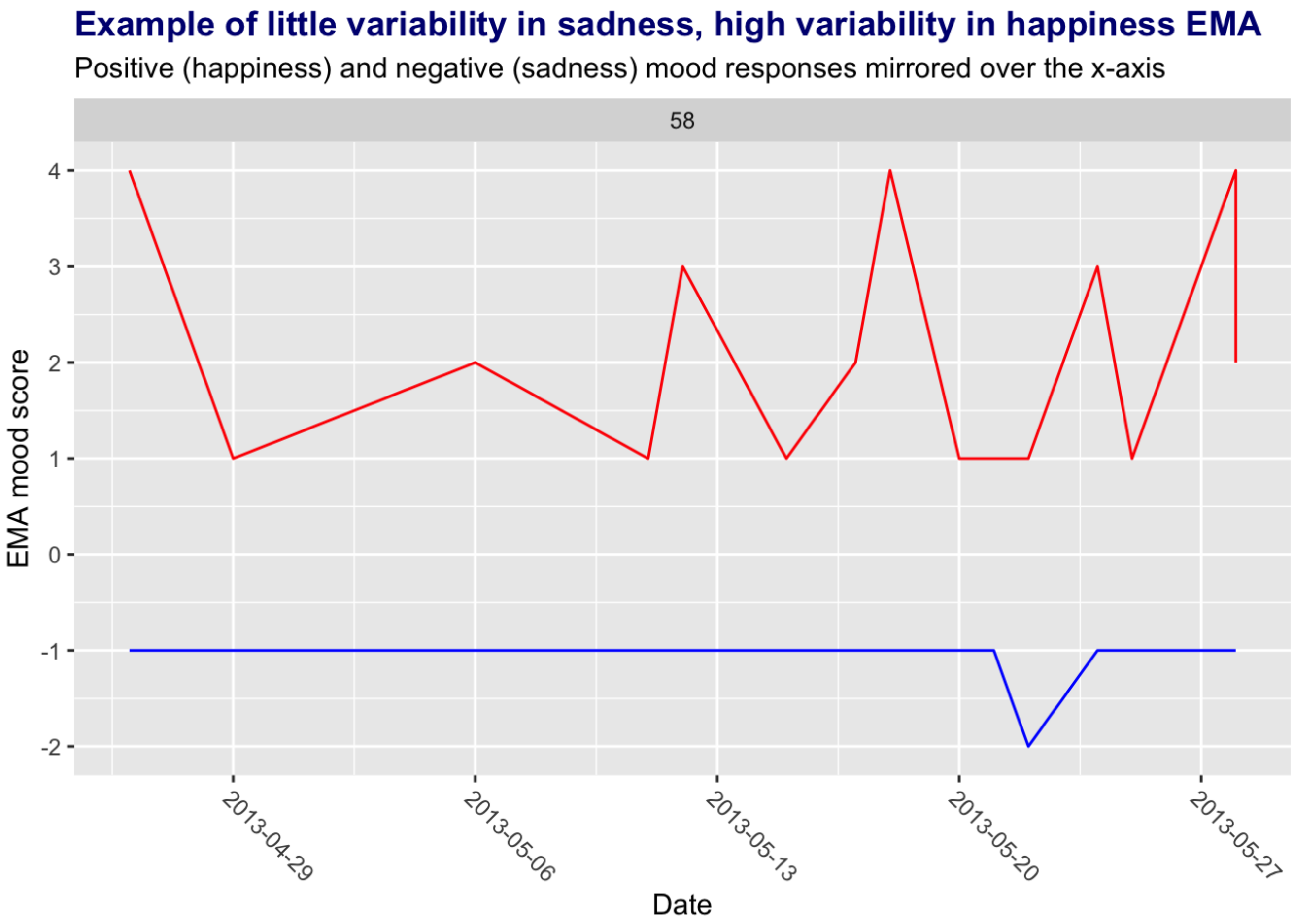
Exploratory Analysis

The major exploratory analysis component of our work was identifying and conceptualizing an adverse psychological event. We asked two guiding questions as our starting point. First, which data streams could be linked to well established psychological phenomena? Second, what trends found by the Dartmouth-based group captured a well established psychological phenomenon? The common answer to both questions suggested a psychological event of interest: An individual's dips in mood — henceforth "mood dips" — as measured by a daily happiness ecological momentary assessment (EMA) survey.

Our exploratory analyses fine-tuned this "mood dip" concept. We considered two mood EMAs in our data from which to conceptualize and define our mood dip concept: A daily sadness EMA rating and daily happiness EMA rating (11). Each day, subjects could rate on a 1 to 4 scale how happy and how sad they felt that day.

In many cases, we found students did not vary considerably overtime in their sadness ratings. By contrast, student happiness ratings showed considerable variability over time. We aimed to capture changes in mood. Therein, we worked exclusively with the happiness EMA data to construct our mood dip definition and concept.

```
## Please install the necessary packages, if not already installed on your system

#install.packages("studentlife")
#install.packages("plyr")
#install.packages("formattable")

## Please call the relevant libraries.
library("studentlife")
library(lubridate)
library(dplyr)
library(plyr)
library(ggplot2)
library(scales)
library(forcats)
library(formattable)

##################################### Download data from R package using a temporary driv
e and calling the Student Life tables,
##################################### based on instructions from the StudentLife site

#d <- tempdir()
#download_studentlife(location = d, url = "rdata")
#SL_tables

d <- getwd()

schema <- "EMA"
table <- "Mood"
mood <- readRDS(paste0(d, "/", schema, "/", table, ".Rds"))

##################################### End of download section

## The available responses in the mood module are

#Questions to "Mood" module:
#1) Do you feel AT ALL happy right now? ID: "happyornot"
#   Binary answers: 1=yes 2=no
#2) If you answered "Yes" on the first question, how happy do you feel? ID: "happy"
#   1=a little bit, 2=somewhat, 3=very much, 4=extremely
#3) Do you feel AT ALL sad right now? ID: "sadornot"
#   Binary answers: 1=yes, 2=no
#4) If you answered "Yes" on the third question, how sad do you feel? ID: "sad"
#   1=a little bit, 2=somewhat, 3=very much, 4=extremely

#Questions to "Mood 1" module:
#How do you think you will be this time tomorrow?
#1 = happy, 2=stressed, 3=tired

#Questions to "Mood 2" module:
#How are you right now? ID: "how"
#1 = happy 2=stressed 3=tired


# Within the EMA mood data, we reformatted the Timestamp data and created a new column,
 "date"
#Creating new column in dataframe: "date"
mood$date <- mood$timestamp
mood$date <- as.Date(as.POSIXct(mood$date, origin="1970-01-01"))

### Plotting EMA of example students
moodB <- mood %>%
  filter(uid %in% c(58))
moodB$sad = - moodB$sad
y1B <- moodB$happy
y2B <- moodB$sad
ggplot(moodB, aes(date)) +
  geom_line(aes(y=y1B), color = "red") +
  geom_line(aes(y=y2B), color = "blue") +
  scale_x_date(breaks = date_breaks("weeks")) +
  scale_y_continuous(breaks = seq(-4, 4, by = 1)) +
  ylab("EMA mood score") +
  xlab("Date") +
  labs(title = "Example of little variability in sadness, high variability in happiness EMA",
       subtitle = "Positive (happiness) and negative (sadness) mood responses mirrored over the x-axis") +
  theme(plot.title = element_text(face = "bold", colour = "Navy")) +
  theme(axis.text.x = element_text(angle=-45, hjust=0, vjust=1)) +
  facet_wrap(~uid)
```



The EMA happiness data had a high degree of missingness. Future work should ensure higher response rates. For lack of a better alternative, we ignored missing points, treating the existing data points as a continuous trend.

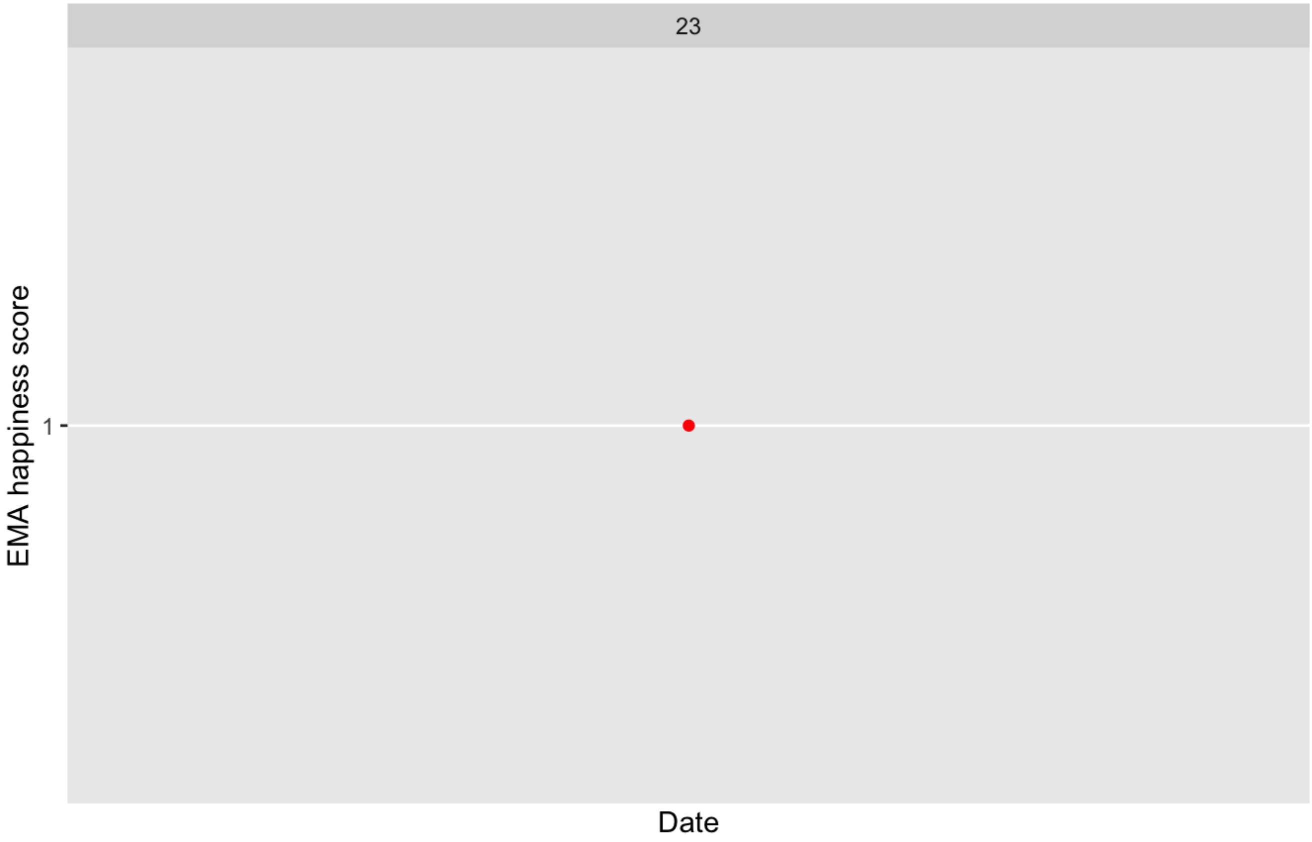
We first considered how we could define our mood dip concept in terms of a well-established diagnostic survey available in our data. Students took pre- and post-semester surveys, including PHQ-9, a survey often used to diagnose and assess the severity (minimal, mild, moderate, moderately severe, or severe) of depression (12). Several students showed a change in depression severity between the start and end of term, and, similarly, the Dartmouth-based research group found the cross-student average in EMA happiness showed a considerable steady decrease between the start and end of term. We thus considered the longest possible timescale on which a mood dip could play out for an individual student: an on-average change in happiness EMA between the start and end of the semester. We identified the students who had changes in PHQ-9 severity between the start and end of the semester.

```
library("readr")

################DOWNLOADED DATA FROM R PACKAGE##################
d <- getwd()

## import specific tables for analysis
schema <- "EMA"
table <- "Mood"
mood <- readRDS(paste0(d, "/", schema, "/", table, ".Rds")) # class: data frame
mood$date <- as.Date(as.POSIXct(mood$timestamp, origin="1970-01-01")) # change UNIX times
tamp to date

schema <- "EMA"
table <- "PAM"
pam <- readRDS(paste0(d, "/", schema, "/", table, ".Rds"))
pam$date <- as.Date(as.POSIXct(pam$timestamp, origin="1970-01-01"))

schema <- "survey"
table <- "PHQ-9"
phq9 <- readRDS(paste0(d, "/", schema, "/", table, ".Rds"))
phq9_pre_post <- phq9 %>% filter(type=="post")

## PHQ-9 EDA
# 9 questions (Q1 to Q9)
# Levels: Not at all=0, Several days=1, More than half the days=2, Nearly every day=3
# Depression severity level: 0-4 none, 5-9 mild, 10-14 moderate, 15-19 moderately severe,
20-27 severe
# change tibble to dataframe and only select entries for uids with mood
phq9 <- as.data.frame(phq9) %>% filter(uid %in% unique(mood$uid))

## create score columns according to Q1-Q9 word answers
for (i in 3:11){
  col_name <- paste0("score_Q", i-2)
  phq9[[col_name]] <- ordered(phq9[, i],
                        levels =c("Not at all", "Several days",
                                  "More than half the days", "Nearly every day"))
  phq9[[col_name]] <- as.integer(phq9[[col_name]]) - 1 # change 1,2,3,4 to 0,1,2,3
}

## sum up Q1-9 scores
phq9[["score_Total"]] <- rowSums(phq9[, 13:21], na.rm = TRUE)

## change score to depression severity level
for (i in 1:nrow(phq9)){
  if(phq9$score_Total[i] < 5){
    phq9$severity[i] <- "none"
  }else if(phq9$score_Total[i] < 10){
    phq9$severity[i] <- "mild"
  }else if(phq9$score_Total[i] < 15){
    phq9$severity[i] <- "moderate"
  }else if(phq9$score_Total[i] < 20){
    phq9$severity[i] <- "moderately severe"
  }else{
    phq9$severity[i] <- "severe"
  }
}

phq9$severity <- ordered(as.factor(phq9$severity),
                         levels =c("none", "mild", "moderate",
                                   "moderately severe", "severe"))

# construct data frames for pre and post separately
phq9_pre <- phq9 %>%
  filter(phq9$type == "pre") %>%
    select("uid", "score_Total", "severity")
colnames(phq9_pre)[2] <- "pre_score_Total"
colnames(phq9_pre)[3] <- "pre_severity"

phq9_post <- phq9 %>%
  filter(phq9$type == "post") %>%
  select("uid", "score_Total", "severity")
colnames(phq9_post)[2] <- "post_score_Total"
colnames(phq9_post)[3] <- "post_severity"

# combine 2 data frames
phq9_reshaped <- left_join(phq9_pre, phq9_post, by="uid")

## calculate score change and severity level change
# score_dif>0 denotes a depression score "increase" over time
# severity_change>0 denotes a depression severity level "increase" over time
phq9_reshaped$score_dif <- phq9_reshaped$post_score_Total - phq9_reshaped$pre_score_Total
phq9_reshaped$severity_change <- as.integer(phq9_reshaped$post_severity) - as.integer(phq
9_reshaped$pre_severity)

################## SAVED PHQ-9 TABLE################################
write_csv(phq9_reshaped, "Yin_PHQ9 summary for mood uids.csv")
################## SAVED PHQ-9 TABLE################################

# uids for those having a depression level "increase"
phq9_reshaped$uid[which(phq9_reshaped$severity_change > 0)]

# uids with severity changes
phq9_reshaped %>%
  filter(!(severity_change == 0 |is.na(severity_change)))
```

Those students were, however, relatively few and, within those few, they did not reliably show an on-average change in happiness EMA over the semester that was commensurate with their change in PHQ-9 depression severity.

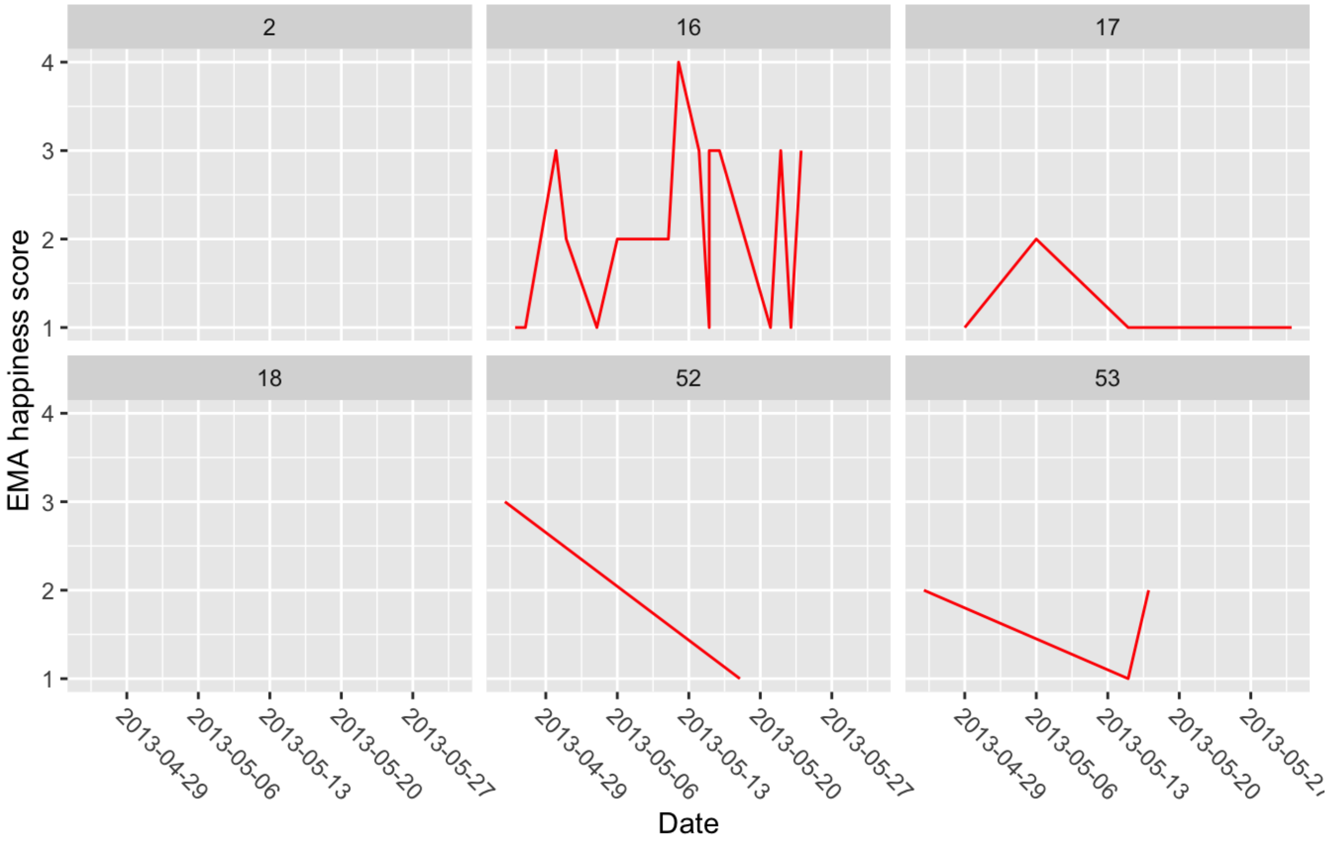
Students who moved from depressed to not depressed

Positive and negative mood responses mirrored over the x-axis



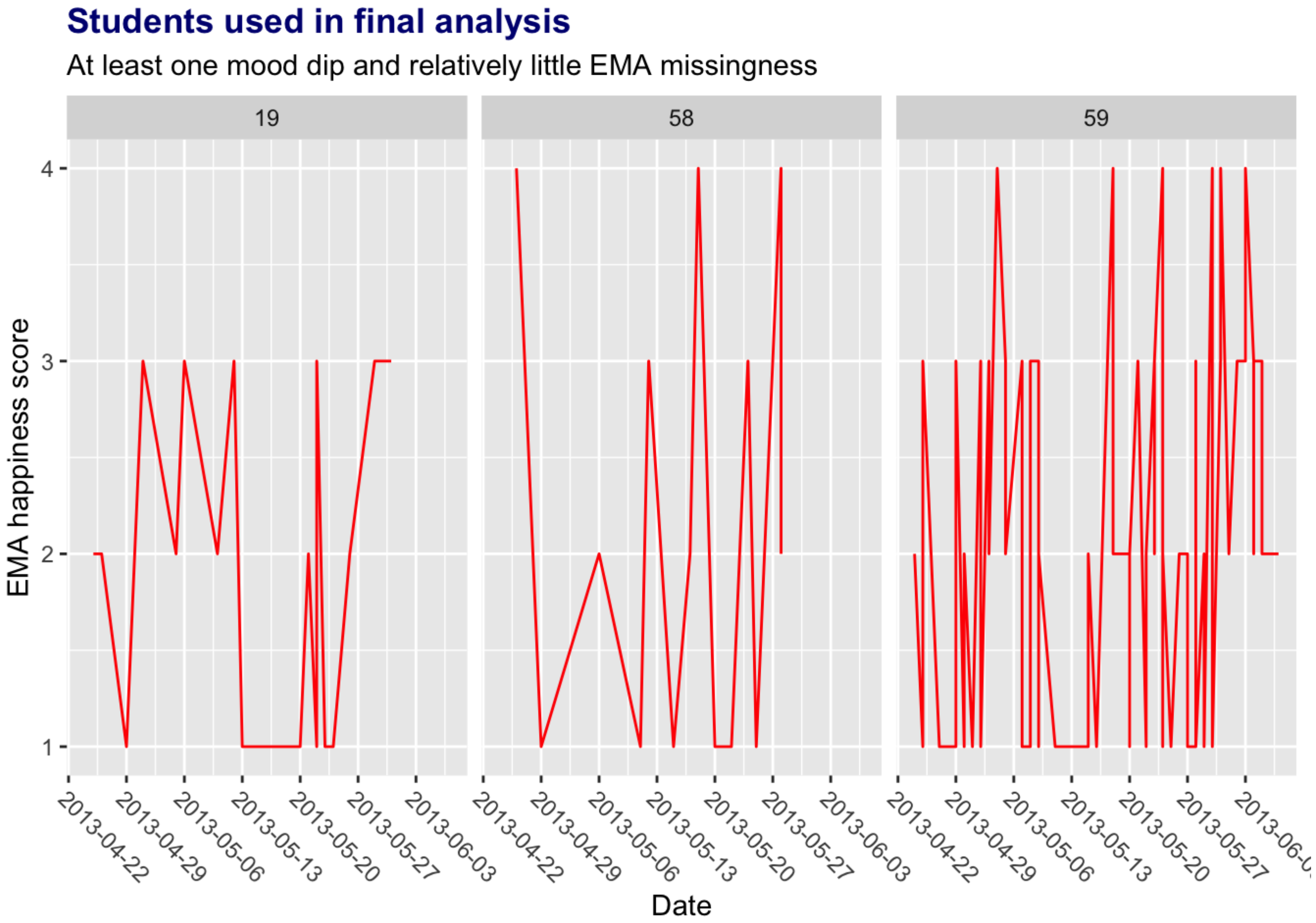
Students who moved from not depressed to depressed

Positive and negative mood responses mirrored over the x-axis



Next, we considered smaller timescales on which changes in mood could play out. Many students exhibited small timescale (multi-day) dips in mood away from a baseline EMA happiness level typical for an individual — a phenomenon that many of us can recognize in our own experience and which research suggests to be a veritable psychological phenomenon (13). We thus defined a mood dip as follows: a >= 3 day dip in an individual's happiness EMA score away from their baseline happiness level (personal median happiness EMA score).

Three students who had at least one mood dip are plotted below.



Data

The dataset tracks the social and behavioral habits of n = 48 Dartmouth students, mostly undergraduates, who took a specific programming class during the 10-week Spring term of 2013. Students took surveys and were interviewed at the start and end of the term. During the term, their phones collected multiple passive data streams and students were prompted to take daily surveys. In many cases, the Dartmouth-based group that produced this data pre-processed the data, providing daily summary measures instead of the raw data.

Features

The data streams and summary measures included in the dataset spans dozens of features. Full details are available in (14). We limit our discussion here to those features we considered for our anomaly detection task. Our final feature set is not exhaustive of all the features we considered. Omitted features features did not contain a sufficient spread of data for our chosen anomaly detection algorithm to work (The algorithm requires that <50% of timepoints share the same value). Below, we list those features ultimately included in our final analysis, organized by the conglomerate category into which each feature was grouped for anomaly detection — e.g. TotalConvo and ConvoDuration compose the "sociability" conglomerate category of features provided to our anomaly detection algorithm. We also specify the label by which each of these features are named in our data tables.

Sociability

- Daily conversation duration (ConvoDuration): The dataset provided timestamps denoting the start-time and end-time of conversations, inferred from audio recordings. The total daily time spent in conversation was computed as hours per day.
- Daily conversation count (TotalConvo): The total number of conversations per day. Using the same dataset for the “daily conversation duration” measure, this was computed as the sum of conversations each student had per day, without regard for conversation length.

Stress

- Daily number of deadlines (Deadlines): Collected via a daily survey, this is the number of deadlines a student had on a particular day.
- Rolling mean of deadlines in next week: (SevenDayAvg): The seven-day rolling average of deadlines a student had for the current day and the following six days. This was computed from the same daily survey data used for “Number of deadlines.”

Mobility

- Daily time spent stationary (Activity): The proportion of time per day, on a 0 - 1 scale, for which the individual was stationary. This was measured as the proportion of time per day for which the individual’s the phone was stationary (location did not change) as inferred from phone sensor data.
- Daily time spent exercising (Exercise): Collected via a daily survey, students estimates of how long they spent exercising per day. Students selected a time range, which were coded as 1 = None, 2 = < 30 minutes, 3 = 30 - 60 minutes, 4 = 60 - 90 minutes, 5 > 90 minutes). In some cases, students answered this survey question twice per day. In these cases, the average between the two surveys was used.

Sleep

- Student estimated nightly sleep (EMAsleep): Collected via a daily survey, student estimates of how long they slept that night. Student selected from a range of estimates, which were coded as 1 = < 3 hours, 2 = 3.5 hours, 3 = 4 hours, … 18 = 11.5 hours, 19 = 12+ hours.
- Inferred nightly sleep (InferredSleep): Estimated time spent asleep each night. This was inferred from non-negative least squares regression applied to phone sensor data, following a previous method used for this task (15). For full details, see “Inferring sleep via non-negative least squares regression” subsection.

Inferring sleep via non-negative least squares regression

The following code uses phone sensing data (audio, activity, charge, lock, dark) to infer hours slept per night.

First, load the required libraries and datasets (Note: the audio and activity datasets have 99 million and 22 millions rows, respectively. They might take a minute to load or exceed your computer’s memory.).

```
library(tidyverse)
library(lubridate)
library(dplyr)
library(ggplot2)


PhoneCharge <- read.csv("phonecharge.csv")
PhoneDark <- readRDS("dark.Rds")
PhoneLock <- readRDS("phonelock.Rds")
PhoneActivity <- read.csv("Sensing_Activity.csv")
PhoneAudio <- readRDS("audio.Rds")
SleepDat <- readRDS(file = "Sleep.Rds")    #This is an EMA
```

Next, convert Unix timestamps to dates for all the datasets.

```
PhoneCharge$startDate <- as.Date(as.POSIXct(PhoneCharge$`start_timestamp`, origin="1970-01-01"))
PhoneCharge$timeLapseCh <- (PhoneCharge$end_timestamp - PhoneCharge$start_timestamp)/3600

PhoneDark$startDate <- as.Date(as.POSIXct(PhoneDark$`start_timestamp`, origin="1970-01-01"))
PhoneDark$timeLapseDk <- (PhoneDark$end_timestamp - PhoneDark$start_timestamp)/3600

PhoneLock$startDate <- as.Date(as.POSIXct(PhoneLock$`start_timestamp`, origin="1970-01-01"))
PhoneLock$timeLapseLk <- (PhoneLock$end_timestamp - PhoneLock$start_timestamp)/3600

PhoneActivity$DateTime <- as_datetime(as.POSIXct(PhoneActivity$timestamp, origin="1970-01-01"))
PhoneActivity$Date <- as.Date(as.POSIXct(PhoneActivity$timestamp, origin="1970-01-01"))

PhoneAudio$Date <- as.Date(as.POSIXct(PhoneAudio$timestamp, origin="1970-01-01"))
PhoneAudio$DateTime <- as_datetime(as.POSIXct(PhoneAudio$timestamp, origin="1970-01-01"))

SleepDat <- SleepDat[order(SleepDat$timestamp, SleepDat$uid),]
SleepDat$Date <- as.Date(as.POSIXct(SleepDat$timestamp, origin="1970-01-01"))
```

The ‘hour’ column within the EMA sleep dataset (SleepDat) does not correspond to hours slept per night. It corresponds to answers to the following questionnaire:

“How many hours did you sleep last night?”

[1]<3, [2]3.5, [3]4, [4]4.5, [5]5, [6]5.5, [7]6, [8]6.5, [9]7, [10]7.5, [11]8, [12]8.5, [13]9, [14]9.5, [15]10, [16]10.5, [17]11, [18]11.5, [19]12

The following code creates a new column ‘HoursSleep’ which gives the number of hours each student slept per night.

```
SleepDat = SleepDat %>% mutate(HoursSleep = recode(hour,
                                                   `1`=3, #`1` actually means <3 hours of 
sleep
                                                   `2`=3.5,
                                                   `3`=4,
                                                   `4`=4.5,
                                                   `5`=5,
                                                   `6`=5.5,
                                                   `7`=6,
                                                   `8`=6.5,
                                                   `9`=7,
                                                   `10`=7.5,
                                                   `11`=8,
                                                   `12`=8.5,
                                                   `13`=9,
                                                   `14`=9.5,
                                                   `15`=10,
                                                   `16`=10.5,
                                                   `17`=11,
                                                   `18`=11.5,
                                                   `19`=12))
SleepDat <- SleepDat[-c(1,2,3,4,5)]
```

We will be using non-negative least squares regression to infer the hours slept per night using phone sensing data. The sensing datasets contain multiple rows per student, per day. We want to summarize the rows so that we only have one row per student, per day, within each dataset. The datasets were summarized in the following way:

PhoneCharge: This dataset contains time lapses during which the phone was charging. We summarized this dataset so that each student has one observation per day: the total phone charge time (in hours) between 10pm and 10am.

PhoneDark: This dataset contains time lapses during which the phone was in darkness. We summarized this dataset so that each student has one observation per day: the total time the phone was in the dark (in hours) between 10pm and 10am.

PhoneLock: This dataset contains time lapses during which the phone was locked. We summarized this dataset so that each student has one observation per day: the total time the phone was locked (in hours) between 10pm and 10am.

PhoneAudio: This dataset contains audio inferences collected every 2-3 seconds for 1 minute intervals. The sensing software within the smartphone app made audio inferences for 1 minute, then paused for 3 minutes before restarting. If the conversation classifier detected that there was a conversation going on, it kept running until the conversation finished. The meaning of audio inference is described below.

0: Silence, 1: Voice, 2: Noise, 3: Unknown

We summarized this dataset so that each student has one observation per day: the proportion of time the phone recorded silence, during typical sleeping hours (10pm to 10am)

PhoneActivity: This dataset contains activity inferences collected every 2-3 seconds for 1 minute intervals. The sensing software within the smartphone app made activity inferences for 1 minute, then paused for 3 minutes before restarting. The meaning of activity inference is described below.

0: Stationary, 1: Walking, 2: Running, 3: Unknown

We summarized this dataset so that each student has one observation per day: the proportion of time the phone detected no movement (activity inference = 0), during typical sleeping hours (10pm to 10am)

The motivation for summarizing the datasets in the manner described above came from this paper (%22https://www.researchgate.net/publication/261054378_Unobtrusive_Sleep_Monitoring_using_Smartphones%22), written by the authors of the StudentLife paper.

We found that restricting summaries of the five predictor variables to typical sleeping hours (10pm to 10am) produced the best sleep predictions.

```
PhoneChargeSum <- PhoneCharge %>%
  filter(hour(startDate)<=10 |hour(startDate)>=22 ) %>%
  group_by(startDate,uid) %>%
  summarise(totalTimeChg <- sum(timeLapseCh))

PhoneDarkSum <- PhoneDark %>%
  filter(hour(startDate)<=10 |hour(startDate)>=22 ) %>%
  group_by(startDate, uid) %>%
  summarise(totalTimeDrk <- sum(timeLapseDk))


PhoneLockSum <- PhoneLock %>%
  filter(hour(startDate)<=10 |hour(startDate)>=22 ) %>%
  group_by(startDate, uid) %>%
  summarise(totalTimeLck <- sum(timeLapseLk))


PhoneAudioMean <- PhoneAudio %>%
  filter(hour(DateTime)<=10 |hour(DateTime)>=22 ) %>%      #I'm finding the proportion of
audio_inference = 0 (this denotes silence) between 10pm and 10am
  group_by(Date, uid) %>%
  summarise(meanAudio <- sum(audio_inference==0)/length(Date))


PhoneActivityMean <- PhoneActivity %>%
  filter(hour(DateTime)<=10 |hour(DateTime)>=22 ) %>%      #I'm finding the proportion of
activity_inference = 0 (this denotes stillness) between 10pm and 10am
  group_by(Date, uid) %>%
  summarise(meanActivity <- sum(activity_inference==0)/length(Date))
```

Next, we merged the above datasets with a dataset containing the EMA sleep responses of students in the study.

```
PhoneAudio$uid = as.factor(PhoneAudio$uid)
PhoneActivity$uid = as.factor(PhoneActivity$uid)
PhoneCharge$uid = as.factor(PhoneCharge$uid)

dat_full <- inner_join(PhoneCharge, PhoneDark, by = c("startDate","uid"))
dat_full <- inner_join(dat_full, PhoneLock, by = c("startDate","uid"))
dat_full <- inner_join(dat_full, PhoneAudio, by = c("startDate"="Date", "uid"))
dat_full <- inner_join(dat_full, PhoneActivity, by = c("startDate"="Date", "uid"))
dat_full <- unique(inner_join(dat_full,SleepDat, by = c("startDate"="Date","uid")))

#Change column names
colnames(dat_full)[3] <- "totalTimeChg"
colnames(dat_full)[4] <- "totalTimeDrk"
colnames(dat_full)[5] <- "totalTimeLck"
colnames(dat_full)[6] <- "meanAudio"
colnames(dat_full)[7] <- "meanActivity"
dat_full <- dat_full[!is.na(dat_full$HoursSleep), ]  #there was one NA value. It is delet
ed here.
```

Next, we split the merged dataset into testing and training sets. Since this dataset is reasonably large (934 rows), we performed a 50-50 split.

```
set.seed(260)
test_indices <- sample(1:nrow(dat_full), round(length(dat_full$HoursSleep)*.5), replace=F
ALSE)

test_set <- dat_full[test_indices,]
train_set <- dat_full[-test_indices,]
```

We performed non-negative least squares regression to predict the hours slept per night, using the nnls function within the nnls package. This function takes as arguments a matrix of the predictor variables and a vector of the observed outcomes. The code below loads the nnls package and creates the predictor matrix and outcome vector, and runs the nnls function.

```
library(nnls)

#create the matrix and vector
matrix_dat <- as.matrix(data.frame(train_set$totalTimeChg, train_set$totalTimeDrk, train_
set$totalTimeLck, train_set$meanAudio, train_set$meanActivity))
EMASleep <- as.vector(train_set$HoursSleep)

##non-negative least squares
nnls(matrix_dat, EMASleep)
```

The nnls function returned coefficients for the five predictor variables. We applied the non-negative least squares regression model to the test dataset to determine its prediction accuracy.

```
test_set = test_set %>%
  mutate(InferSleep <- 0*totalTimeChg + 0*totalTimeDrk + 0.04682086*totalTimeLck + 0.2799
582*meanAudio + 5.559868*meanActivity)
colnames(test_set)[9] <- "InferredSleep"
```

Finally, we checked the accuracy of our method by examining the distribution of the absolute difference between inferred and EMA sleep

```
summary(abs(test_set$HoursSleep - test_set$InferredSleep))
```

The average absolute difference between the inferred sleep and EMA sleep is 0.85 hours, or 51 minutes. This is not far from the average absolute difference in the paper (%22https://www.researchgate.net/publication/261054378_Unobtrusive_Sleep_Monitoring_using_Smartphones%22) above, of approximately 41 minutes. According to this paper, “most health recommendations related to sleep duration (e.g., [4], [5]) assume hour-level accuracy or often consider just aggregate trends (e.g., Is a person sleeping less than normal?”. Therefore, we believe our prediction model accuracy is sufficient.

Now that we have our prediction model, we can apply it to the full dataset containing the five predictors. We start by merging the five predictor datasets to create a dataset called dat_final.

```
dat_final <- inner_join(PhoneLockSum, PhoneAudioMean, by = c("startDate"="Date", "uid"))
dat_final <- inner_join(dat_final, PhoneActivityMean, by = c("startDate"="Date", "uid"))

Change column names
colnames(dat_final)[3] <- "totalTimeLck"
colnames(dat_final)[4] <- "meanAudio"
colnames(dat_final)[5] <- "meanActivity"

#Note: Since we did not merge with the EMA sleep dataset, PhoneLockSum, or PhoneChargeSu
m, dat_final has 2536 rows. This is significantly greater than dat_full which only had 93
4 rows.
```

Next, we apply non-negative least squares regression, using the coefficients found above.

```
dat_final = dat_final %>%
 mutate(InferSleep <- 0.04682086*totalTimeLck + 0.2799582*meanAudio + 5.559868*meanActivi
ty)
colnames(dat_final)[6] <- "InferredSleep"
```

Lastly, export dat_final to a csv file.

```
library("readr")
write_csv(dat_final,"Inferred_Sleep_Data.csv")
```

Subject and timespan selection

Our final set of students and the data we used from each was chosen by two criteria. First, to amend for considerable missingness in the happiness EMA data, students had to have relatively little missingness in their EMA happiness data compared to other students. Second, students had to have at least one mood dip. Ultimately, to assess the accuracy of our algorithm, we compared anomaly rates in periods preceding vs. not preceding a mood dip. In the interest of increasing our sample size of periods not preceding a mood dip, why did we not include more students who satisfied criterion one but not two? As Barnett et al. found, and which our results parallel, anomaly rates vary considerably from student to student (4). For the sake of a fair assessment of anomaly rates in pre-dip periods versus other periods, we exclusively used students who satisfied both criteria. A total of three students fit these criteria.

From those three students, our final dataset included only the span of time for which the student had happiness EMA data. This culminated in a total n = 4 mood dips spread amongst the three students, over a total of about 15 weeks of data. The data pipeline by which we formed those final data tables is shown below. For brevity, we only show the data pipeline for one student included in our final analysis and for one feature of the many that went into our anomaly detection algorithm. The code we used to create data tables of other features is included in our project submission.

```
library(readr)
library(lubridate)
library(tidyverse)
library(zoo)
library(readr)
library(formattable)
#library(plyr)
library(dplyr)

#####################################################Creating and organizing tables for deadline feature, a part of the stress feature set

DeadlinesDat <- read.csv("Deadlines.csv")
ClassDat <- readRDS("Class.Rds")
Class2Dat <- readRDS("Class 2.Rds")

DeadlinesDat$date <- ymd(DeadlinesDat$date)
DeadlinesDat <- DeadlinesDat %>% filter(date <= as.Date("2013-06-05"))

DeadlinesDat$DaysIndex <- as.numeric(DeadlinesDat$date - as.Date("2013-03-24"),units="days")
DeadlinesDat = DeadlinesDat[c(2,4,1,3)]

DeadlinesRollmean = DeadlinesDat %>%
  group_by(uid) %>%
  mutate(SevenDayAvg = rollmean(deadlines,k=7,fill=NA, align = "left"))

#Save files separately for each student (deadline/day)
for (id in seq(0,60)){
  d = as.data.frame(DeadlinesDat[DeadlinesDat$uid == id,][-c(3)])
  write_csv(d, paste0("Deadlines_UID",id,".csv"))
  #print(id)
  #print(length(d$DaysIndex))
}

#Save files separately for each student (rollmean deadlines for the next 7 days)
for (id in seq(0,60)){
  d = as.data.frame(DeadlinesRollmean[DeadlinesRollmean$uid == id,][-c(3,4)])
  write_csv(d, paste0("DeadlinesRollmean_UID",id,".csv"))
  #print(id)
  #print(length(d$DaysIndex))
}

############################################Creating and organizing happiness EMA table for one student

#In the code below, we use uid 59 as the example.

d <- getwd()

schema <- "EMA"
table <- "Mood"
mood <- readRDS(paste0(d, "/", schema, "/", table, ".Rds"))
```

```
mood$date <- mood$timestamp
mood$date <- as.Date(as.POSIXct(mood$date, origin="1970-01-01"))

###Data frame with full dates to be merged with incomplete dateline data frame, adding NA
s for missing dates
### UID 59
Happiness_uid59 <- mood %>%
  filter(uid == 59) %>%
  group_by(date)
fulldates_59 <- seq(min(Happiness_uid59$date), max(Happiness_uid59$date),
                    by = "1 day")
fulldates_59 <- data.frame(date = fulldates_59)

## Merge the complete data frame with the incomplete to fill in the dates and add
## NAs for missing values
Happiness_uid59 <- merge(fulldates_59, Happiness_uid59, by = "date",
                         all.x = TRUE)

Happiness_uid59$dayindex <- 1:nrow(Happiness_uid59)

Happiness_uid59$uid <- fct_explicit_na(Happiness_uid59$uid, "59")

Happiness_uid59 <- select(Happiness_uid59, uid, dayindex, date, happy)


### Printing the tables
#formattable(Happiness_uid59)

###########################################Creating and organizing final feature table fo
r one student

id <- 59

################IMPORT DATA#################
Dining <- read.csv(paste0("Dining_UID",id,".csv"))
Dining$Date <- as.Date(Dining$Date)

ConvoDuration <- read.csv(paste0("ConvoDuration_UID",id,".csv"))
ConvoDuration$Date <- as.Date(ConvoDuration$Date)

TotalConvos <- read.csv(paste0("TotalConvos_UID",id,".csv"))
TotalConvos$Date <- as.Date(TotalConvos$Date)

Activity <- read.csv(paste0("ActivityProportion_UID",id,".csv"))
Activity$Date <- as.Date(Activity$Date)

Exercise <- read.csv(paste0("Exercise_UID",id,".csv"))
Exercise$Date <- as.Date(Exercise$Date)

Stress <- read.csv(paste0("Stress_UID",id,".csv"))
Stress$Date <- as.Date(Stress$Date)

Deadlines <- read.csv(paste0("Deadlines_UID",id,".csv"))
Deadlines$Date <- as.Date(Deadlines$date)
```

```
DeadlinesRollmean <- read.csv(paste0("DeadlinesRollmean_UID",id,".csv"))
DeadlinesRollmean$Date <- as.Date(DeadlinesRollmean$date)

EMASleep <- read.csv(paste0("EMASleep_UID",id,".csv"))
EMASleep$Date <- as.Date(EMASleep$Date)
names(EMASleep)[3] <- "HoursSlept"

InferredSleep <- read.csv(paste0("InferredSleep_UID",id,".csv"))
InferredSleep$Date <- as.Date(InferredSleep$Date)

#################IMPORT DATA##################

# set date timespan
ema_date = seq(as.Date("2013-04-24"), as.Date("2013-06-07"), by="day")

# create summary dataframe with pre-defined timespan
UID_summary <- data.frame(Date = ema_date, DaysIndex = seq(1, length(ema_date)))

# merge all dataframes
UID_summary <- UID_summary %>%
  left_join(Dining[, c(1,3)], by = "Date") %>%
  left_join(ConvoDuration[, c(1,3)], by = "Date") %>%
  left_join(TotalConvos[, c(1,3)], by = "Date") %>%
  left_join(Activity[, c(1,3)], by = "Date") %>%
  left_join(Exercise[, c(1,3)], by = "Date") %>%
  left_join(Stress[, c(1,3)], by = "Date") %>%
  left_join(Deadlines[, c(3,4)], by = "Date") %>%
  left_join(DeadlinesRollmean[, c(3,4)], by = "Date") %>%
  left_join(EMASleep[, c(1,3)], by = "Date") %>%
  left_join(InferredSleep[, c(1,3)], by = "Date")


################## SAVED EMA TABLE################################
write_csv(UID_summary,paste0("Summary_UID",id,".csv"))
################## SAVED EMA TABLE################################
```

Final Analysis

Potential to predict mood dips

To assess the accuracy of our mood dip prediction algorithm, we investigated the rate at which anomalies occurred in periods preceding vs not preceding a mood dip. We did this via the method described below.

We defined a period preceding a mood dip to be the three days prior to the start of a mood dip. To ensure our work investigated mood prediction and not mood dip identification, the comparison class (periods not preceding mood dips) was defined as periods that were neither mood dips nor periods preceding mood dips. Our results were robust to other timescales — i.e. we got comparable results regardless of whether we defined a period preceding a mood dip to be the 2, 3 or 4 days prior to the start of a dip. The results we present here are for the 3-day definition.

We identified anomalies as follows. Filzmoser’s anomaly detection algorithm produces anomaly scores for each feature at each timepoint — in this case, for daily sociability, stress, mobility and sleep of each student. Anomaly scores range between 0 and 1, with more anomalous data points assigned lower scores. We assessed anomaly rates using several cut-off scores for identifying anomalies, 0.05, 0.1 and 0.15. The results we present here are for the 0.1 cut-off. Our results were comparable for the other cut-off values.

We prepped our data for our anomaly detection algorithm as follows. Our anomaly detection algorithm cannot handle missing values, thus necessitating imputation. Following Barnett et al.’s approach, we used a weighted average of data points in close temporal proximity to the missing data points. Specifically, we used the linear subtype of the na_ma function within the imputeTS package in R. This function imputes the weighted average of two timepoints on either side of the missing value, expanding the window as necessary to get two existing timepoints, with the closest value weighted by 1/2 and the second closest weighted by 1/3. To identify anomalies, our anomaly detection algorithm requires that >=50% of a given feature not have the same value. If, after imputation, a feature failed to meet that criterion, that feature was excluded for the given student.

Below is the code with which we produced anomaly scores for one student.

```r
library(imputeTS)
library(mvoutlier)
library(ggplot2)
library(reshape2)
library(lubridate)
library(tidyverse)
library(readr)

################################Reading in data
id <- 59
dat <- read.csv(paste0("Summary_UID",id,".csv"))

##################Breaking data up into conglomerate features
sociability <- as.data.frame(dat[4:5])
#Exclude 7 for 59 because >= 50% of that feature is the same.
mobility <- as.data.frame(dat[6])
#Exclude 8 and 9 for 59  because >= 50% of that feature is the same.
stress <- as.data.frame(dat[10])
sleep <- as.data.frame(dat[11:12])

###################Imputing missing values with linear weighted average
#dining <- na_ma(dining, k = 2, weighting = "linear", maxgap = Inf)
sociability <- na_ma(sociability, k = 2, weighting = "linear", maxgap = Inf)
mobility <- na_ma(mobility, k = 2, weighting = "linear", maxgap = Inf)
stress <- na_ma(stress, k = 2, weighting = "linear", maxgap = Inf)
sleep <- na_ma(sleep, k = 2, weighting = "linear", maxgap = Inf)

#Getting AN scores
sociabilityAN=as.data.frame(pcout(sociability,makeplot=FALSE))
mobilityAN=as.data.frame(pcout(mobility,makeplot=FALSE))
stressAN=as.data.frame(pcout(stress,makeplot=FALSE))
sleepAN=as.data.frame(pcout(sleep,makeplot=FALSE))

# create a dataframe from combined data
naa <- data.frame(matrix(NA, nrow = length(stress), ncol = 1))
df = data.frame(dat$Date, dat$DaysIndex, sociabilityAN$wfinal, mobilityAN$wfinal, stressA
N$wfinal, sleepAN$wfinal)
names(df) <- c("date","day","sociability","mobility","stress","sleep")

#Save data
write_csv(df, paste0("AN_UID",id,".csv"))
```

We found evidence that our anomaly detection system can predict mood dips. For individuals who had mood dips, the rate of anomalies in periods preceding mood dips was 69.13% higher than the rate of anomalies detected on other days. Given the rarity of mood dips and large amount of missingness in the happiness EMA data, formal tests could not be done and sensitivity and specificity could not be computed.

Overall, our data suggests a humble conclusion: There is potential to predict multi-day dips in happiness from largely passive smartphone data. The difference in anomaly rates was small: Pre-dip periods had an average 0.204 anomalies per day, while periods not preceding a dip had an average of anomalies per day 0.126. Additionally, the difference in anomaly rates — as well as the difference in the raw features themselves — is not visually apparent. Our shiny app, which shows the anomaly rates and raw feature values for each student, illustrates this. With the benefit of more data, and less missingness, perhaps we would have found a more marked difference.

Our shiny app is coded below. This app, however, will only work if ran out of its own R script. That separate script is included in our project submission.

```
library(shiny)
library(ggplot2)
library(lubridate)
library(tidyverse)
library(dplyr)


#Download datasets
for (id in c(19,58,59)){
    assign(paste0("Summary_UID",id), read.csv(paste0("Summary_UID",id,".csv")))
    assign(paste0("Happiness_UID",id), read.csv(paste0("Happiness_uid",id,".csv")))
    assign(paste0("AN_UID",id), read.csv(paste0("AN_UID",id,".csv")))
}


Happiness_UID59  = Happiness_UID59 %>%
    group_by(date) %>%
    summarize(happy = round(mean(happy))) %>%
    ungroup() %>%
    mutate(uid = 59)

Happiness_UID19 = Happiness_UID19[-c(2)]
Happiness_UID58 = Happiness_UID58[-c(2)]
Happiness_UID59 = Happiness_UID59[c(3,1,2)]

Summary_UID19$UID = 19
Summary_UID58$UID = 58
Summary_UID59$UID = 59


AN_UID19$UID = 19
AN_UID58$UID = 58
AN_UID59$UID = 59


longDat = bind_rows(Summary_UID19, Summary_UID58, Summary_UID59)
longDatHappy = bind_rows(Happiness_UID19, Happiness_UID58, Happiness_UID59)
longDatAN = bind_rows(AN_UID19, AN_UID58, AN_UID59)

##Change Column Names
colnames(longDat)[4] = "Convo Duration"
colnames(longDat)[5] = "Total Convos"
colnames(longDat)[10] = "Avg Deadlines"
colnames(longDat)[11] = "Sleep"
colnames(longDat)[12] = "Inferred Sleep"

colnames(longDatAN)[3] = "Sociability"
colnames(longDatAN)[4] = "Mobility"
colnames(longDatAN)[5] = "Stress"
colnames(longDatAN)[6] = "Sleep"


ui <- fluidPage(
    titlePanel("Detecting Anomalous Behavioral Trends to Predict Mood Dips"),
    sidebarLayout(
        sidebarPanel(
            selectizeInput(inputId = "feature",
                           label = "Choose (up to 3) Features",
                           multiple = TRUE,
                           options = list(maxItems=3),
                           selected = "Sleep",
                           c("Stress","Deadlines","Sleep","Exercise","Activity","Total Co
nvos","Convo Duration","Inferred Sleep")

            ),
            selectInput(inputId = "UID",
                        label = "Choose a Student",
                        c("19","58","59")),

            dateRangeInput(
                inputId = "Date",
                label = "Choose Date Range",
                start = ymd("2013-04-25"),
                end = ymd("2013-06-01"),
                min = ymd("2013-04-25"),
                max = ymd("2013-06-01")
            ),

            tableOutput("summary")
        )

        ,
        mainPanel(
            verticalLayout(
                plotOutput("FeaturesPlot"),
                plotOutput("AnomalyPlot"),
                plotOutput("HappinessPlot")
            )
        )
    )

)
server <- function(input, output) {
    selected = reactive(longDat %>%
                            filter(UID == input$UID) %>%
                            filter(Date >= as.Date(input$Date[1], origin = "1970-01-01")
 & Date <= as.Date(input$Date[2], origin = "1970-01-01")) %>%
                            select(c("Date", as.vector(input$feature))) %>%
                            gather("feature","value",2:(1+length(as.vector(input$featur
e)))))

    selectedHappy = reactive(longDatHappy %>%
                                 filter(uid == input$UID) %>%
                                 filter(ymd(date) >= as.Date(input$Date[1], origin = "197
0-01-01") & ymd(date) <= as.Date(input$Date[2], origin = "1970-01-01")) %>%
                                 select(c("date","happy")))

    selectedAN = reactive(longDatAN %>%
                              filter(UID == input$UID) %>%
                              filter(ymd(date) >= as.Date(input$Date[1], origin = "1970-0
1-01") & ymd(date) <= as.Date(input$Date[2], origin = "1970-01-01")) %>%
                              select(c("date","Sociability","Mobility","Stress","Sleep"))
%>%
                              gather("feature","value",2:5))

    table = reactive(longDat %>%
                         filter(UID == input$UID) %>%
                         filter(Date >= as.Date(input$Date[1], origin = "1970-01-01") & D
ate <= as.Date(input$Date[2], origin = "1970-01-01")) %>%
                         select(c("Date", as.vector(input$feature))))

    ##Create named vectors defining dates of depressive dips
    uids1 = c(19,58,59)
    uids2 = c(19,58,59)
    uids3 = c(19,58,59)
    uids4 = c(19,58,59)
    uids5 = c(19,58,59)
    uids6 = c(19,58,59)
    names(uids1) = c("2013-05-15","2013-05-20","2013-04-27")
    names(uids2) = c("2013-05-18","2013-05-22","2013-04-29")
    names(uids3) = c("","","2013-05-11")
    names(uids4) = c("","","2013-05-13")

    FeatureColors = c("Stress" = "purple","Deadlines" = "purple","Sleep" = "green", "Infe
rred Sleep" = "darkgreen","Exercise" = "red","Activity"="red","Total Convos"="blue","Conv
o Duration"="blue")
    ANColors = c("Sociability" = "blue","Mobility"="red","Stress"="purple","Sleep"="gree
n")

    output$FeaturesPlot = renderPlot({
        selected() %>%
            ggplot(aes(Date,value, color = feature)) +
            geom_point(size=2.5) +
            scale_colour_manual(values =FeatureColors)+
            ggtitle(paste("Features by Day for Student", input$UID)) +
            xlab("")+ ylab("")+
            theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
            facet_grid(feature ~., scales = "free_y") +

            #The following lines plot the shaded region
            annotate("rect", xmin = names(uids1)[uids1 == input$UID], xmax = names(uids2)
[uids2 == input$UID], ymin = -Inf, ymax = Inf,
                 alpha = .2) +
            annotate("rect", xmin = names(uids3)[uids3 == input$UID], xmax = names(uids4)
[uids4 == input$UID], ymin = -Inf, ymax = Inf,
                     alpha = .2)

    })
    output$AnomalyPlot = renderPlot({
        selectedAN() %>%
            ggplot(aes(date, value,color = feature)) +
            ggtitle(paste("Anomaly Detection by Feature for Student", input$UID)) +
            theme(axis.text.x = element_text(angle = 90, hjust = 1))+
            xlab("")+ ylab("")+
            geom_point(aes(color = feature), size=2.5) +
            scale_colour_manual(values =ANColors) +

            #The following lines plot the shaded region
            annotate("rect", xmin = names(uids1)[uids1 == input$UID], xmax = names(uids2)
[uids2 == input$UID], ymin = -Inf, ymax = Inf,
                     alpha = .2) +
            annotate("rect", xmin = names(uids3)[uids3 == input$UID], xmax = names(uids4)
[uids4 == input$UID], ymin = -Inf, ymax = Inf,
                     alpha = .2)

    })

    output$HappinessPlot = renderPlot({
         selectedHappy() %>% na.omit() %>%
             ggplot(aes(as.Date(date),happy)) +
             ggtitle(paste("Happiness EMA Score for Student", input$UID)) +
             xlab("")+ ylab("")+
             theme(axis.text.x = element_text(angle = 90, hjust = 1))+
             geom_point(size=2.5)+
             geom_line() +

            #The following lines plot the shaded region
            annotate("rect", xmin = as.Date(names(uids1)[uids1 == input$UID]), xmax = as.
Date(names(uids2)[uids2 == input$UID]), ymin = -Inf, ymax = Inf,
                     alpha = .2) +
            annotate("rect", xmin = as.Date(names(uids3)[uids3 == input$UID]), xmax = as.
Date(names(uids4)[uids4 == input$UID]), ymin = -Inf, ymax = Inf,
                     alpha = .2)

    })

    output$summary = renderTable(

        table()
    )
}

# Run the application
shinyApp(ui = ui, server = server)
```

Comparable results to Barnett et al. and larger implications

Our results were comparable to that of Barnett et al. (3). They also dealt with a low sample size, not due to a low subject number, but due to missingness and other factors that necessitated excluding subjects and data. Likewise, perhaps due to that low sample size, they also found only a small difference in anomaly rates. Both sets of results provide an encouraging but limited preliminary view into smartphone-based psychological/behavioral insights (3).

The comparability of our results and Barnett et al.'s highlights the need for a concerted effort to scale and standardize this sort of work. There are a plethora of labs and companies producing similar behavioral research and smartphone applications from largely passively-collected smartphone data. Many of these efforts, like our results here, often have promising but negligible positive results. Such results might generate sufficient excitement to economically support those labs and companies in the short term. But truly riveting results — results that make effective smartphone applications a reality and results that forward the NIMH mission to deeply understand mental health — will require long-term scalability. Sample sizes must increase. Methods to competently handle missingness must be developed. To allow samples to be pooled and studies to be fairly compared, heterogeneity in what and how data is collected across smartphone models, phone applications and studies must be switched out for a universal standard of raw sensor data collection. As others have noted, high rates of smartphone ownership afford an unprecedented opportunity for scalability and longitudinal analysis (16,4). But this burgeoning field must work hard — and, importantly, work together — to seize those opportunities.

Citations

1. “Active Mobile Phones Outnumber Humans for the First Time,” International Business Times: https://www.ibtimes.co.uk/there-are-more-gadgets-there-are-people-world-1468947 (https://www.ibtimes.co.uk/there-are-more-gadgets-there-are-people-world-1468947).
2. “CrossCheck: Toward passive sensing and detection of mental health changes in people with schizophrenia,” Wang et al., 2016.
3. “Relapse prediction in schizophrenia through digital phenotyping: a pilot study,” Barnett et al., 2018.
4. “Harnessing Smartphone-Based Digital Phenotyping to Enhance Behavioral and Mental Health,” Onnela and Rauch, 2016.
5. “About RDoC,” National Institute of Mental Health: https://www.nimh.nih.gov/research/research-funded-by-nimh/rdoc/about-rdoc.shtml (https://www.nimh.nih.gov/research/research-funded-by-nimh/rdoc/about-rdoc.shtml)
6. “A multivariate outlier detection method,” Filzmoser, 2004.
7. “Patterns of Behavior Change in Students Over an Academic Term: A Preliminary Study of Activity and Sociability Behaviors Using Smartphone Sensing Methods,” Harari et al., 2017.
8. “The relationship between mobile phone location sensor data and depressive symptom severity,” Saeb et al., 2016.
9. “SmartGPA: How Smartphones Can Assess and Predict Academic Performance of College Students,” Wang et al., 2015.
10. “StudentLife: Assessing Mental Health, Academic Performance and Behavioral Trends of College Students using Smartphones,” Wang et al., 2014.
11. “Ecological Momentary Assessment,” Shiffman et al., 2008.
12. “The PHQ-9,” Kroenke et al., 2011.
13. “A computational and neural model of momentary subjective well-being,” Rutledge et al., 2014.
14. “Student Life Study,” Dartmouth: https://studentlife.cs.dartmouth.edu/ (https://studentlife.cs.dartmouth.edu/)
15. “Unobtrusive Sleep Monitoring using Smartphones,” Chen et al., 2013.
16. “New Tools for New Research in Psychiatry: A Scalable and Customizable Platform to Empower Data Driven Smartphone Research,” Torous et al., 2016