Preprocessing_GLOBEM

Joe Gyorda

2023-03-27

This codebook contains the code corresponding to the manuscript "Detecting Longitudinal Trends Between Passively-Collected Phone Use and Anxiety among College Students". Data was leveraged from the GLOBEM study with their permission.

There is a growing body of research linking phone use to anxiety, with many existing theories citing both behavioral addiction and psychological pathway conceptualizations of phone use and anxiety. However, these studies are generally cross-sectional and do not reveal the longitudinal/causal pathways of phone use and how (or how not) it contributes to anxiety. We seek to use the GLOBEM dataset to understand how phone use (assessed via duration of unlock episodes each day) correlates with and predicts PHQ4 anxiety levels. We also seek to investigate the role of location with phone use and anxiety, examining an individual's phone use at home vs away from home.

The code book is organized as follows:

1. Preprocessing – read in all dataframes
2. Feature engineering – create all features to be used in modeling
3. Visualization and analysis of features
4. Model and results – mixed-effects logistic regression model
5. Appendix – contains supplementary code and visuals

1. Preprocessing

Let's read in the dataset with the PHQ-4 survey results.

```r
setwd('/users/joegyorda/Desktop/Jacobson lab/GLOBEM/')

# read in data from each study wave
phq_d2 = read_csv('globem-dataset/INS-W_2/SurveyData/ema.csv', show_col_types=F)
```

```
## New names:
## * `` -> `...1`
```

```r
phq_d3 = read_csv('globem-dataset/INS-W_3/SurveyData/ema.csv', show_col_types=F)
```

```
## New names:
## * `` -> `...1`
```

```r
phq_d4 = read_csv('globem-dataset/INS-W_4/SurveyData/ema.csv', show_col_types=F)
```

```
## New names:
## * `` -> `...1`
```

```r
phq_d2$wave = 2; phq_d3$wave = 3; phq_d4$wave = 4

# combine into one dataframe
phq_all = rbind(phq_d2,phq_d3,phq_d4)

# remove missing values -- we only will consider non-missing PHQ-4 anxiety records
phq_all = phq_all[complete.cases(phq_all$phq4_anxiety_EMA),]

# filter to only the phq4 anxiety data
phq_all = phq_all %>%
  dplyr::select(pid, date, wave, phq4_anxiety_EMA)
head(phq_all)
```

```
## # A tibble: 6 x 4
##   pid       date        wave phq4_anxiety_EMA
##   <chr>     <date>     <dbl>            <dbl>
## 1 INS-W_300 2019-03-31     2                1
## 2 INS-W_300 2019-04-07     2                1
## 3 INS-W_300 2019-04-11     2                2
## 4 INS-W_300 2019-04-21     2                2
## 5 INS-W_300 2019-04-28     2                2
## 6 INS-W_300 2019-05-05     2                0
```

```r
# check frequencies for individual IDs
paste("Person-years:", length(unique(phq_all$pid))) # 607 person-years
```

```
## [1] "Person-years: 607"
```

```r
summary(as.numeric(table(phq_all$pid)))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   9.000  10.000   9.728  11.000  11.000
```

```r
# remove original dataframes to save space
rm(phq_d2,phq_d3,phq_d4)
```

Looking at individual participant IDs, >75% have 10 occurrences (i.e., ~10 weeks) of PHQ-4 records, shown above.

Now, read in the phone use and location data. We subset to features capturing the summed phone unlock duration at the daily level at different locations (home, green space, etc).

First, read in and preprocess the phone usage data. We'll extract daily records pertaining to an individuals summed phone use duration in total and at different locations.

```r
phone_d2 = read_csv('globem-dataset/INS-W_2/FeatureData/screen.csv', show_col_types=F)
```

```
## New names:
## * `` -> `...1`
```

```r
phone_d3 = read_csv('globem-dataset/INS-W_3/FeatureData/screen.csv', show_col_types=F)
```

```
## New names:
## * `` -> `...1`
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```r
phone_d4 = read_csv('globem-dataset/INS-W_4/FeatureData/screen.csv', show_col_types=F)
```

```
## New names:
## * `` -> `...1`
```

```r
phone_d2$wave = 2; phone_d3$wave = 3; phone_d4$wave = 4
phone_all = rbind(phone_d2,phone_d3,phone_d4)

# subset phone use data
phone_all = phone_all %>%
  dplyr::select(pid,date,wave,matches('sumdurationunlock')) %>%
  dplyr::select(pid,date,wave,matches('allday'))
phone_all = phone_all[,1:9] # remove discretized/normalized features

colnames(phone_all) = c("pid","date","wave","sumdurationunlock","sumdurationunlock_exercise",
                        "sumdurationunlock_greens","sumdurationunlock_living",
                        "sumdurationunlock_study","sumdurationunlock_home")
head(phone_all, 5)
```

```
## # A tibble: 5 x 9
##   pid       date        wave sumdurationunlock sumdurationunlock_exercise
##   <chr>     <date>     <dbl>             <dbl>                      <dbl>
## 1 INS-W_300 2019-03-21     2              NA                         NA
## 2 INS-W_300 2019-03-22     2              NA                         NA
## 3 INS-W_300 2019-03-23     2              NA                         NA
## 4 INS-W_300 2019-03-24     2              97.3                       NA
## 5 INS-W_300 2019-03-25     2             329.                        32.4
## # i 4 more variables: sumdurationunlock_greens <dbl>,
## #   sumdurationunlock_living <dbl>, sumdurationunlock_study <dbl>,
## #   sumdurationunlock_home <dbl>
```

```r
length(unique(phone_all$pid)) # 550 person-years
```

```
## [1] 550
```

```r
unique(table(phone_d2$date)) # shows all dates are present! each date occurs 218 times
```

```
## [1] 218
```

```r
# remove original dataframes to save space
rm(phone_d2,phone_d3,phone_d4)
```

Note that we have phone use data at many locations, including at green space, while exercising, at home, etc. Let's check the missingness rates for each feature:

```r
phone_ids = unique(phone_all$pid)

missing_ph = matrix(ncol=6)
for (id in phone_ids) {
  sub_data = phone_all[phone_all$pid==id,]
  missing_ph = rbind(missing_ph, colMeans(is.na(sub_data[,-c(1:3)])))
}
missing_ph = missing_ph[-1,]
summary(missing_ph)
```

```
##  sumdurationunlock sumdurationunlock_exercise sumdurationunlock_greens
##  Min.   :0.1340    Min.   :0.2621             Min.   :0.2062
##  1st Qu.:0.1942    1st Qu.:0.8247             1st Qu.:0.4845
##  Median :0.2136    Median :0.9417             Median :0.9029
##  Mean   :0.2451    Mean   :0.8868             Mean   :0.7537
##  3rd Qu.:0.2718    3rd Qu.:1.0000             3rd Qu.:0.9903
##  Max.   :0.9223    Max.   :1.0000             Max.   :1.0000
##  sumdurationunlock_living sumdurationunlock_study sumdurationunlock_home
##  Min.   :0.1753           Min.   :0.2680          Min.   :0.1359
##  1st Qu.:0.3093           1st Qu.:0.4948          1st Qu.:0.2330
##  Median :0.9417           Median :0.9612          Median :0.2784
##  Mean   :0.7142           Mean   :0.7766          Mean   :0.3219
##  3rd Qu.:1.0000           3rd Qu.:1.0000          3rd Qu.:0.3669
##  Max.   :1.0000           Max.   :1.0000          Max.   :1.0000
```

The `missing_ph` dataframe examines the per-person missingness rates for each feature. Above, we see that an individual are missing a median of 21.36% of total phone unlock duration records and 27.84% at-home phone unlock duration daily records. While not ideal, note that this doesn't reflect our final sample since we haven't paired the phone use data with PHQ4 records yet. However, because the exercise/green/living/study phone unlock duration records all exhibit high missingness rates (median missingness>90% for each), we will not include these variables in our analyses. We'll only use total daily phone use, daily phone use at home, and daily phone use not at home (calculated below).

Great! Now we'll read in and preprocess the location data, which will tell us how much time an individual spent at different locations. We'll later use this to calculate the proportion of time an individual spent on their phone at each location.

```
location_d2 = read_csv('globem-dataset/INS-W_2/FeatureData/location.csv', show_col_types=F)
```

```
## New names:
## * `` -> `...1`
```

```
location_d3 = read_csv('globem-dataset/INS-W_3/FeatureData/location.csv', show_col_types=F)
```

```
## New names:
## * `` -> `...1`
```

```
location_d4 = read_csv('globem-dataset/INS-W_4/FeatureData/location.csv', show_col_types=F)
```

```
## New names:
## * `` -> `...1`
```

```
location_d2$wave = 2; location_d3$wave = 3; location_d4$wave = 4
location_all = rbind(location_d2,location_d3,location_d4)

# subset location data
location_all = location_all %>%
  dplyr::select('pid','date','wave',matches('allday')) %>%
  dplyr::select('pid','date','wave',matches(c('timeathome:','hometime:')))

colnames(location_all) = c("pid","date","wave","timeathome","hometime")
head(location_all,5)
```

```
## # A tibble: 5 x 5
##   pid       date        wave timeathome hometime
##   <chr>     <date>     <dbl>      <dbl>    <dbl>
## 1 INS-W_300 2019-03-21     2         NA       NA
## 2 INS-W_300 2019-03-22     2         NA       NA
## 3 INS-W_300 2019-03-23     2         NA       NA
## 4 INS-W_300 2019-03-24     2       199.     236.
## 5 INS-W_300 2019-03-25     2       798.     962.
```

```
length(unique(location_all$pid)) # 550 person-years
```

```
## [1] 550
```

```r
# unique(table(location_d4$date)) # shows all dates are present!

rm(location_d2,location_d3,location_d4) # don't need these anymore
```

Next, we'll combine the phone use and location dataframes:

```r
# subset columns from phone_all
phone_all2 = phone_all %>%
  dplyr::select(pid, date, wave, sumdurationunlock, sumdurationunlock_home)

# combine phone use data with location data
phone_loc_all = phone_all2 %>%
  inner_join(location_all, by=c("pid"="pid","date"="date","wave"="wave"))

# ensure we only keep phone/loc data for individuals for whom we have PHQ data
phone_loc_ids = unique(phone_loc_all$pid)
final_ids = intersect(phone_loc_ids,unique(phq_all$pid))
phone_loc_all = phone_loc_all[phone_loc_all$pid %in% final_ids,]

head(phone_loc_all,5)
```

```
## # A tibble: 5 x 7
##   pid       date        wave sumdurationunlock sumdurationunlock_home timeathome
##   <chr>     <date>     <dbl>             <dbl>                  <dbl>      <dbl>
## 1 INS-W_300 2019-03-21     2              NA                     NA         NA
## 2 INS-W_300 2019-03-22     2              NA                     NA         NA
## 3 INS-W_300 2019-03-23     2              NA                     NA         NA
## 4 INS-W_300 2019-03-24     2              97.3                   66.8      199.
## 5 INS-W_300 2019-03-25     2             329.                   220.       798.
## # i 1 more variable: hometime <dbl>
```

Let's check out the ranges of dates for which we have phone use & location data:

```r
# get date ranges - same for location data
date_range_p2 = range(phone_all$date[phone_all$wave==2])
date_range_p3 = range(phone_all$date[phone_all$wave==3])
date_range_p4 = range(phone_all$date[phone_all$wave==4])
cat(paste("Wave 2 date range: ", date_range_p2[1], '-', date_range_p2[2],
          "\nWave 3 date range: ", date_range_p3[1], '-', date_range_p3[2],
          "\nWave 4 date range: ", date_range_p4[1], '-', date_range_p4[2]))
```

```
## Wave 2 date range:  2019-03-21 - 2019-06-25
## Wave 3 date range:  2020-03-16 - 2020-06-26
## Wave 4 date range:  2021-03-29 - 2021-07-09
```

Below, we'll summarize phone usage with the median daily value from the 14 days prior to a PHQ4 record. Here, we'll make sure for each PHQ4 record that there's at least 14 days of phone/location data prior to its collection.

```r
# remove people without phone_loc data
phq_all = phq_all[phq_all$pid %in% final_ids,]
```

```r
# create new dataframe for filtered PHQ records
phq_all2 = data.frame(matrix(0,ncol=ncol(phq_all)))
colnames(phq_all2) = colnames(phq_all)
phq_all2$date = as.Date(phq_all2$date, origin='1970-01-01')

# filter phq data to make sure each time point has at least 2 weeks of phone_loc data
for (i in 1:nrow(phq_all)) {
  sub_data = phq_all[i,]
  if (sub_data$wave==2) {
    if ((sub_data$date - 14) >= date_range_p2[1]) phq_all2 = rbind(phq_all2,phq_all[i,])
  }
  else if (sub_data$wave==3) {
    if ((sub_data$date - 14) >= date_range_p3[1]) phq_all2 = rbind(phq_all2,phq_all[i,])
  }
  else if (sub_data$wave==4) {
    if ((sub_data$date - 14) >= date_range_p4[1]) phq_all2 = rbind(phq_all2,phq_all[i,])
  }
}
phq_all2 = phq_all2[-1,]

# update final_ids in case we dropped any participants
final_ids = unique(phq_all2$pid)

paste("Original number of PHQ4 records:", nrow(phq_all))
```

```
## [1] "Original number of PHQ4 records: 5404"
```

```r
paste("Updated number of PHQ4 records:", nrow(phq_all2))
```

```
## [1] "Updated number of PHQ4 records: 4505"
```

```r
paste("Original number of unique IDs:", length(unique(phq_all$pid)))
```

```
## [1] "Original number of unique IDs: 550"
```

```r
paste("New number of unique IDs:", length(unique(phq_all2$pid)))
```

```
## [1] "New number of unique IDs: 547"
```

Great! We'll examine again the number of person-years and distribution of records per ID:

```r
length(unique(phq_all2$pid)) # 547 person-years
```

```
## [1] 547
```

```r
final_ids = unique(phq_all2$pid)
summary(as.numeric(table(phq_all2$pid))) # more than half of people have >=9 points
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   8.000   9.000   8.236   9.000  10.000
```

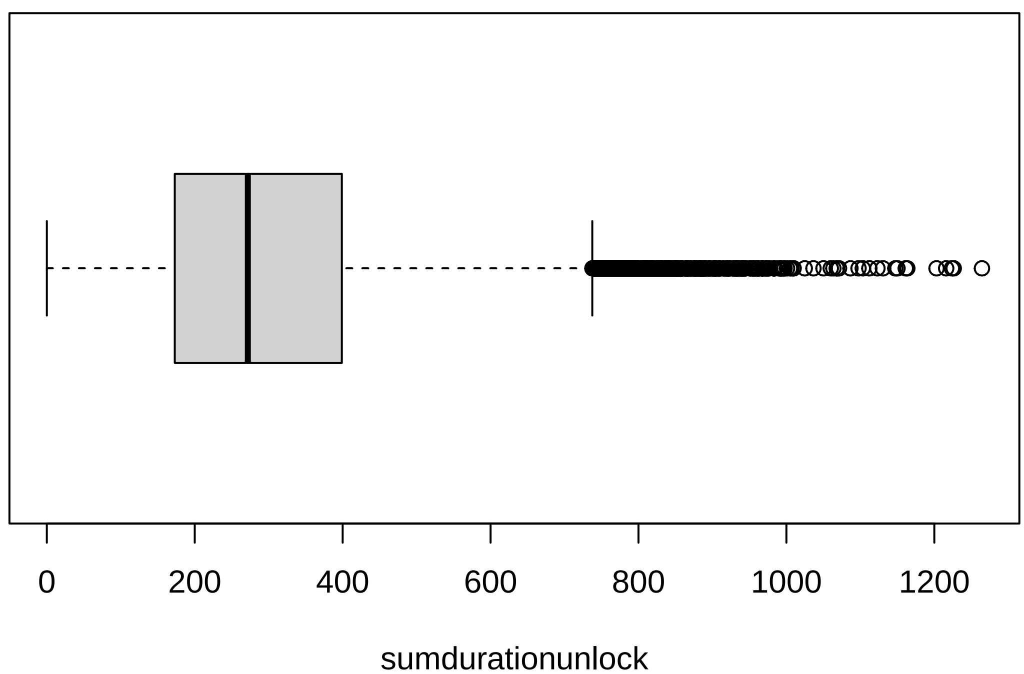
Checking that total unlock duration always greater than home unlock duration - it is!

```
summary(phone_loc_all$sumdurationunlock-phone_loc_all$sumdurationunlock_home)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##    0.000    1.848   40.717   75.593  109.129 1030.436    17850
```

Also quickly checking the distribution of phone unlock duration values in total/at home. We see that the distributions are right-skewed with a few hundred outliers, suggesting that some individuals are spending the majority of the day with their phones unlocked.

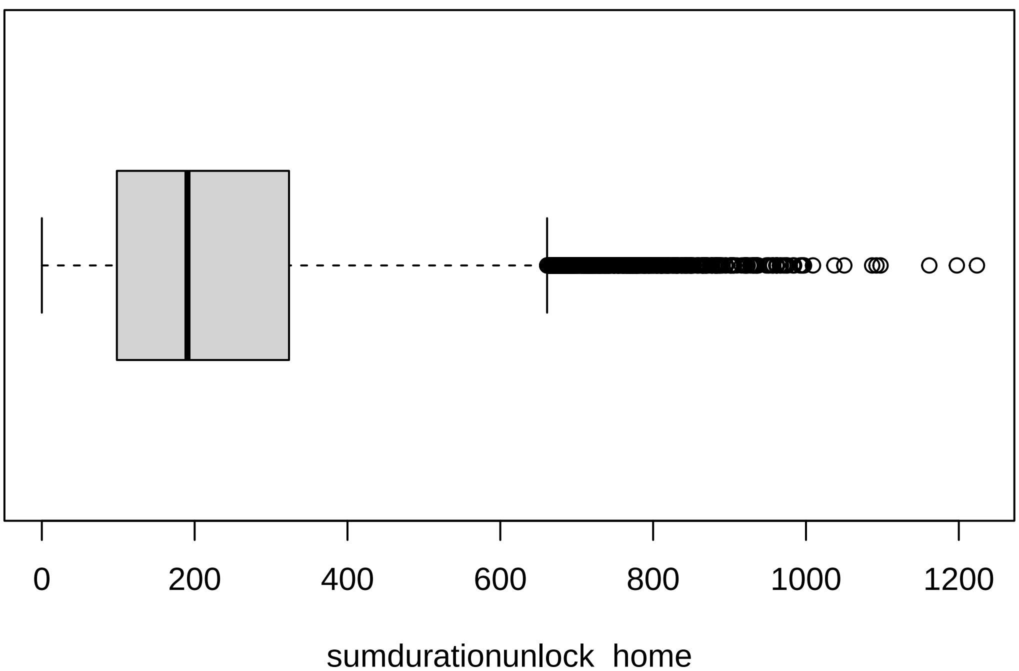
```
# 699 outliers, all on high end
subset(phone_loc_all, phone_loc_all$sumdurationunlock %in%
         boxplot(phone_loc_all$sumdurationunlock, horizontal=T,
                 xlab='sumdurationunlock')$out)
```




```
## # A tibble: 699 x 7
##    pid      date        wave sumdurationunlock sumdurationunlock_home timeathome
##    <chr>    <date>     <dbl>             <dbl>                  <dbl>      <dbl>
##  1 INS-W_3~ 2019-06-07     2              738.                   464.       769.
##  2 INS-W_3~ 2019-05-21     2              819.                   810.      1050.
##  3 INS-W_3~ 2019-03-31     2              997.                   877.      1035.
##  4 INS-W_3~ 2019-06-02     2              816.                   787.      1262.
##  5 INS-W_3~ 2019-05-25     2              879.                   413.       389.
```

```
## 6 INS-W_3~ 2019-04-14     2              763.                   433.      666.
## 7 INS-W_3~ 2019-04-24     2              740.                   217.      621.
## 8 INS-W_3~ 2019-05-01     2              884.                   300.      577.
## 9 INS-W_3~ 2019-03-26     2              749.                   310.      814.
## 10 INS-W_3~ 2019-05-05    2              783.                   731.      903.
## # i 689 more rows
## # i 1 more variable: hometime <dbl>
```

```
# 778 outliers, all on high end
subset(phone_loc_all, phone_loc_all$sumdurationunlock_home %in%
         boxplot(phone_loc_all$sumdurationunlock_home, horizontal=T,
                xlab='sumdurationunlock_home')$out)
```




```
## # A tibble: 778 x 7
##    pid      date        wave sumdurationunlock sumdurationunlock_home timeathome
##    <chr>    <date>     <dbl>             <dbl>                  <dbl>      <dbl>
##  1 INS-W_3~ 2019-05-21     2              819.                   810.      1050.
##  2 INS-W_3~ 2019-03-31     2              997.                   877.      1035.
##  3 INS-W_3~ 2019-06-02     2              816.                   787.      1262.
##  4 INS-W_3~ 2019-05-18     2              716.                   692.      1158.
##  5 INS-W_3~ 2019-06-05     2              702.                   697.      1209.
##  6 INS-W_3~ 2019-05-05     2              783.                   731.       903.
##  7 INS-W_3~ 2019-05-31     2              702.                   670.      1096.
##  8 INS-W_3~ 2019-06-02     2              866.                   765.       965.
##  9 INS-W_3~ 2019-06-05     2              735.                   721.       905.
```

```
## 10 INS-W_3~ 2019-04-11     2              875.                       857.      1252.
## # i 768 more rows
## # i 1 more variable: hometime <dbl>
```

2. Feature engineering

Let's update our phone use variables. We want to create new variables representing the **proportion (0-1)** of time spent on the phone while at different locations. This will make it easier to compare the relationships between phone use/anxiety across different locations and control for day-to-day variation in the amount of time individuals are spending at home/away from home.

First, we found some cases (693) where sumduration_home > timeathome, implying that individuals spent more time on their phones at home than they were actually at home. This is a relatively small subset of the data (2.09%), so to fix this we'll just set the timeathome and sumduration_home equal. This will cause the proportion of time spent on the phone while at home to be 1 for these instances.

```
# code to verify there are 693 cases where sumdurationunlock_home > timeathome
# phone_loc_all[phone_loc_all$sumdurationunlock_home>phone_loc_all$timeathome,][complete.cases(phone_lo
# nrow(phone_loc_all[complete.cases(phone_loc_all),]) = 33150
# so 693/33150*100 = 2.09% of cases. Will just set equal (so proportion=1).

phone_loc_all$sumdurationunlock_home =
  ifelse(phone_loc_all$sumdurationunlock_home>phone_loc_all$timeathome,
         phone_loc_all$timeathome, phone_loc_all$sumdurationunlock_home)
```

To estimate phone use away from home, we'll define the `nottimehome` variable to be the total time in a day (1440 minutes) minus time spent at home. We'll use the same logic to calculate phone use away from home by subtracting phone us at home from total phone use.

```
# new variables for time not at home and phone use not at home
phone_loc_all$nottimeathome = 1440 - phone_loc_all$timeathome # 1440 minutes in a day
phone_loc_all$sumdurationunlock_nothome = phone_loc_all$sumdurationunlock -
  phone_loc_all$sumdurationunlock_home

summary(phone_loc_all$nottimeathome) # always positive which is good
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##    0.033  239.707  577.991  644.214 1003.795 1440.000    13757
```

```
summary(phone_loc_all$sumdurationunlock_nothome) # always positive which is good
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##    0.000    1.954   41.122   76.626  109.892 1063.183    17859
```

We'll now create variables reflecting the daily proportion of time spent on the phone 1) in total, 2) at home, and 3) away from home.

```
# new variable for proportion of total time during day spent on phone
phone_loc_all$phoneuse_total = phone_loc_all$sumdurationunlock / 1440

# new variables for ratios of phone use at home vs not at home vs total
phone_loc_all$phoneuse_home = phone_loc_all$sumdurationunlock_home / phone_loc_all$timeathome
phone_loc_all$phoneuse_nothome = phone_loc_all$sumdurationunlock_nothome /
                                   phone_loc_all$nottimeathome

# some people spend no time at home (0) or all day at home (1440), so remove inf
```

```r
# phone_loc_all[is.infinite(phone_loc_all$phoneuse_nothome),]
phone_loc_all$phoneuse_home[phone_loc_all$timeathome==0]=0
phone_loc_all$phoneuse_nothome[phone_loc_all$timeathome==1440]=0
```

Let's examine the distributions of our new features:

```r
# summaries of features
summary(phone_loc_all$phoneuse_total) # max is 0.878
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   0.120   0.189   0.208   0.277   0.878   13591
```

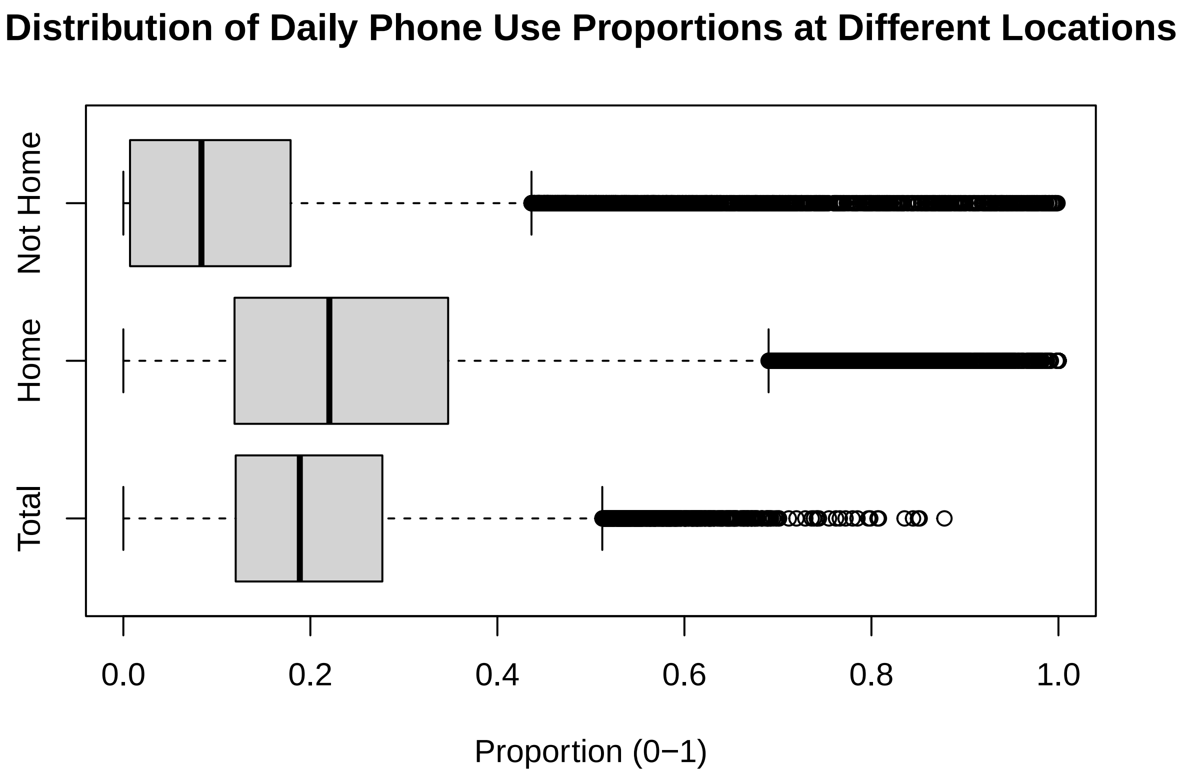
```r
summary(phone_loc_all$phoneuse_home) # max is now 1!
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   0.119   0.220   0.256   0.347   1.000   14234
```

```r
summary(phone_loc_all$phoneuse_nothome) # max 0.999
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   0.000   0.007   0.084   0.122   0.179   0.999   17859
```

```r
# look at distributions
boxplot(phone_loc_all$phoneuse_total, phone_loc_all$phoneuse_home,
        phone_loc_all$phoneuse_nothome, horizontal=TRUE,
        names=c("Total","Home","Not Home"),
        main="Distribution of Daily Phone Use Proportions at Different Locations",
        xlab="Proportion (0-1)")
```



Great! Now we have our raw features for subsequent analysis. In order to pair these with PHQ-4 records, we must obtain the median 14-day values for each phone use feature prior to PHQ-4 assessment.

```
# want 14-day median for phone use ratios at home, not at home, and total
# also count the number of NAs from the 14-day window
phone_loc_binned = NULL

# correspond to columns 6,8,10,11,12
names_summary = c('timeathome','nottimeathome','phoneuse_total','phoneuse_home','phoneuse_nothome')
for (id in final_ids) {
  sub_phone_loc = phone_loc_all[phone_loc_all$pid==id,]
  sub_phq = phq_all2[phq_all2$pid==id,]
  for (i in 1:nrow(sub_phq)) {
    dt = sub_phq$date[i]
    sub_phone_2 = sub_phone_loc %>%
      filter(date>=(dt-14) & date<=(dt-1)) %>%
      summarise(across(names_summary, median, na.rm=T, .names = "median_{.col}"),
                across(names_summary, ~sum(is.na(.)), .names = "NAcount_{.col}"))
    final_dat = cbind(sub_phone_loc$pid[1], dt, sub_phone_loc$wave[1], sub_phone_2,sub_phq$phq4_anxiety_
    phone_loc_binned = rbind(phone_loc_binned,final_dat)
  }
}
```

```
## Warning: There were 2 warnings in `summarise()`.
## The first warning was:
## i In argument: `across(names_summary, median, na.rm = T, .names =
```

```
##   "median_{.col}")`.
## Caused by warning:
## ! Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(names_summary)
##
##   # Now:
##   data %>% select(all_of(names_summary))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```r
colnames(phone_loc_binned)[c(1,2,3,14)] = c("pid","date","wave","phq4_anxiety_EMA")

# remove any remaining missing values -- people with no data in 14-day windows
all_data_comp = phone_loc_binned[complete.cases(phone_loc_binned),]

head(all_data_comp, 5)
```

```
##         pid       date wave median_timeathome median_nottimeathome
## 1 INS-W_300 2019-04-07    2          532.1730             907.8270
## 2 INS-W_300 2019-04-11    2          532.1730             907.8270
## 3 INS-W_300 2019-04-21    2          392.4209            1047.5791
## 4 INS-W_300 2019-04-28    2          518.7259             921.2741
## 5 INS-W_300 2019-05-05    2          518.7259             921.2741
##   median_phoneuse_total median_phoneuse_home median_phoneuse_nothome
## 1             0.1653593            0.2761865               0.1479561
## 2             0.1653593            0.2381411               0.1310305
## 3             0.1419311            0.1567718               0.1267991
## 4             0.1410983            0.1946071               0.1160321
## 5             0.1846067            0.2278710               0.1540018
##   NAcount_timeathome NAcount_nottimeathome NAcount_phoneuse_total
## 1                  0                     0                      0
## 2                  0                     0                      0
## 3                  0                     0                      0
## 4                  0                     0                      0
## 5                  0                     0                      0
##   NAcount_phoneuse_home NAcount_phoneuse_nothome phq4_anxiety_EMA
## 1                     1                        1                1
## 2                     1                        1                2
## 3                     0                        1                2
## 4                     0                        1                2
## 5                     0                        0                0
```

```r
paste("New number of unique IDs:", length(unique(all_data_comp$pid)))
```

```
## [1] "New number of unique IDs: 544"
```

```r
summary(as.numeric(table(all_data_comp$pid))) # more than half of people have >=9 points
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   8.000   9.000   8.138   9.000  10.000
```

Excellent, we now have our raw dataset, including the participant ID, date, wave, PHQ-4 anxiety measurement, and our phone use features summarized across 14-day windows prior to each PHQ-4 record.

Before moving onto modeling, note that some participants were included in multiple waves; however, in the raw GLOBEM dataset, they are assigned a unique ID in each wave. Hence, the '544' sample size we currently have is technically person-years. Upon request, the GLOBEM study coordinators provided supplementary data mapping each participant to a common ID across all waves. Here, I update the participant IDs so that any participant included in multiple waves has the same ID, allowing us to disentangle the relationships between our phone use features and anxiety both within and across study waves. Print statements are included (commented out) for manual debugging/verification.

```
# we now have data mapping participant IDs across waves, so we can update the
# preprocessed data frame and figure out how many unique individuals we have
id_mappings = read_csv('PID_mappings.csv', show_col_types=F)
colnames(id_mappings) = c("pid_2021","pid_2020","pid_2019","pid_2018")
id_mappings = id_mappings[,c(4,3,2,1)] # reorder columns
id_mappings = as.data.frame(id_mappings)

# helps if we make sure id_mappings and PID are the same format
for (i in 1:nrow(all_data_comp)) {
  all_data_comp$pid[i] = str_split(all_data_comp$pid[i],"_",simplify = TRUE)[2]
}
all_data_comp$pid = as.numeric(all_data_comp$pid)

# filter ID mappings to only IDs that are actually in all_data_comp
id_copies = id_mappings[,2:4]
unique_ids_maps = unlist(array(id_copies)); unique_ids_maps = unique_ids_maps[!is.na(unique_ids_maps)]
unique_ids = unique(all_data_comp$pid)
ids_to_remove = setdiff(unique_ids_maps,unique_ids) # ids in id_mappings not in our data

# set all IDs to remove in id_copies to NA so we know they don't occur in our data
id_copies[,1][id_copies[,1]%in%ids_to_remove] = NA
id_copies[,2][id_copies[,2]%in%ids_to_remove] = NA
id_copies[,3][id_copies[,3]%in%ids_to_remove] = NA

# create copy of ID variable to make it easier to track changes!
all_data_comp$new_id = all_data_comp$pid
all_data_comp = all_data_comp[,c(1,15,2:14)]
id_copies2 = id_copies # for checking our work later on

# main part - loop thru all IDs in all_data_comp, check which wave the ID
# belongs to, then if it's in id_mappings, it must have a mapping in another wave
# so check which wave the mapping is in and update the ID!
for (id in unique(all_data_comp$pid)) {
  # wave 2
  if (id >= 300 & id <= 599) {
    # print(id)
    if (id %in% id_copies[,1]) {   # check if ID in wave 2 id_copies
      i = which(id_copies[,1]==id)
      if (!is.na(id_copies[i,2])) { # we need to switch the ID in wave 3!
        replace = id_copies[i,2]
        # cat(id, replace,"\n")
        all_data_comp$new_id[all_data_comp$new_id==replace] = id
        id_copies2[i,2] = id # update it in copied id_copies
      }
```

```r
      if (!is.na(id_copies[i,3])) {  # we need to switch the ID in wave 4!
        replace = id_copies[i,3]
        # cat(id, replace,"\n")
        all_data_comp$new_id[all_data_comp$new_id==replace] = id
        id_copies2[i,3] = id # update it in copied id_copies
      }
    }
  }

  # wave 3
  if (id >= 600 & id <= 899) {
    # print(id)
    if (id %in% id_copies[,2]) {    # check if ID in wave 3 id_copies
      i = which(id_copies[,2]==id)
      if (!is.na(id_copies[i,3]) & is.na(id_copies[i,1])) {  # we need to switch the ID in wave 4!
        replace = id_copies[i,3]
        # cat(id, replace,"\n")
        all_data_comp$new_id[all_data_comp$new_id==replace] = id
        id_copies2[i,3] = id # update it in copied id_mappings
      }
    }
  }
}

# manually through id_copies2 to make sure each row identical - looks good!
paste("Number of person-years:", length(unique(all_data_comp$pid)))
```

```
## [1] "Number of person-years: 544"
```

```r
paste("Sample size:", length(unique(all_data_comp$new_id)))
```

```
## [1] "Sample size: 346"
```

```r
# more than half of people have >=9 points, higher mean now though
summary(as.numeric(table(all_data_comp$new_id)))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    8.00    9.00   12.79   17.00   27.00
```

So we went from 544 person-years to 346 unique participants in our sample. We also see that, across waves, the median number of PHQ-4 records has increased considerably (more than 50% of individuals have >12 records!).

We include Age as an additional demographic feature since, as aforementioned, only some participants are included in multiple waves, so Age and Time are not perfectly correlated and thus cannot be interpreted the same.

```r
age_mappings = read_csv('age_globem.csv', show_col_types=F)
age_2 = age_mappings[age_mappings$PID %in% all_data_comp$pid,]

# currently missing age for 1225
```

```r
# unique(all_data_comp$pid)[!unique(all_data_comp$pid) %in% age_2$PID]

# add age to dataframe
all_data_comp2 = all_data_comp %>%
  inner_join(age_2,by=c("pid"="PID")) # removes 1225 automatically

# drop 1021 for now - no age data, coded as NA
all_data_comp2 = all_data_comp2 %>% filter(new_id!=1021)

paste("New sample size:", length(unique(all_data_comp2$new_id)))
```

```
## [1] "New sample size: 344"
```

```r
paste("Median ages by wave:")
```

```
## [1] "Median ages by wave:"
```

```r
median(as.numeric(all_data_comp2$age)[all_data_comp2$wave==2])
```

```
## [1] 19
```

```r
median(as.numeric(all_data_comp2$age)[all_data_comp2$wave==3])
```

```
## [1] 20
```

```r
median(as.numeric(all_data_comp2$age)[all_data_comp2$wave==4])
```

```
## [1] 20
```

We lost two participants for not having age data. We also see that the median age of participants in each wave is 19-20, suggesting that new (younger) participants were recruited in each wave.

We'll take a quick look at the relationship between time and age in our dataset. We'll define time by setting the earliest point of data collection in Wave 2 as time=0, then each subsequent time point as the number of years since the initial day.

```r
time0 = min(all_data_comp2$date[all_data_comp2$wave==2])
paste("First date of data collection is", time0)
```

```
## [1] "First date of data collection is 2019-04-07"
```

```r
all_data_comp2$time = lubridate::time_length(all_data_comp2$date - as.Date(time0), "years")
# all_data_comp2$phq4_anxiety_EMA = ordered(all_data_comp2$phq4_anxiety_EMA)

cor(all_data_comp2$time, all_data_comp2$age, method='spearman')
```

```
## [1] 0.5318569
```

The rank-order correlation between Age and Time (0.53) indicates a moderate relationship between the two but not strong, as expected.

We now have our dataset ready for analysis!

3. Visualization and analysis of features

First, let's examine the missingness in our features:

```
# most 14-day windows had no missingness for phone use data!
paste("Distribution of NA counts per 14 day window:")
```

```
## [1] "Distribution of NA counts per 14 day window:"
```

```
summary(all_data_comp2$NAcount_phoneuse_total)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.5625  0.0000 13.0000
```

```
summary(all_data_comp2$NAcount_phoneuse_home)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.6685  0.0000 12.0000
```

```
summary(all_data_comp2$NAcount_phoneuse_nothome)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   1.374   2.000  13.000
```

```
# 79% of 14-day windows had no missingness for total phone use, but only 53% for phone use away from ho
cat("\nPercent of 14-day windows COMPLETE for each feature\n")
```

```
##
## Percent of 14-day windows COMPLETE for each feature
```

```
paste0("Total phone use: ",round(nrow(all_data_comp[all_data_comp2$NAcount_phoneuse_total==0,])
      /nrow(all_data_comp2)*100,3), "%")
```

```
## [1] "Total phone use: 79.235%"
```

```
paste0("Home phone use: ",round(nrow(all_data_comp[all_data_comp2$NAcount_phoneuse_home==0,])
      /nrow(all_data_comp2)*100,3), "%")
```

```
## [1] "Home phone use: 75.453%"
```

```
paste0("Not home phone use: ",round(nrow(all_data_comp[all_data_comp2$NAcount_phoneuse_nothome==0,])
      /nrow(all_data_comp2)*100,3), "%")
```

```
## [1] "Not home phone use: 53.759%"
```

Above, we see that completion rates are overall quite great, with >75% of PHQ-4 records being paired with complete data for total/home phone use, and >53% for not home phone use.

We'll conduct hypotheses tests to determine whether the values of our features are changing from wave to wave. Summary stats and test results to be included in table in paper.

```r
# total phone use
pairwise.wilcox.test(x=all_data_comp2$median_phoneuse_total, g=all_data_comp2$wave)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  all_data_comp2$median_phoneuse_total and all_data_comp2$wave
##
##   2      3
## 3 <2e-16 -
## 4 <2e-16 0.3
##
## P value adjustment method: holm
```

```r
summary(all_data_comp2$median_phoneuse_total[all_data_comp2$wave==2])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0215  0.1248  0.1642  0.1790  0.2187  0.4872
```

```r
summary(all_data_comp2$median_phoneuse_total[all_data_comp2$wave==3])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.02983 0.13567 0.21569 0.22096 0.28859 0.57613
```

```r
summary(all_data_comp2$median_phoneuse_total[all_data_comp2$wave==4])
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.009486 0.155468 0.211786 0.224155 0.287524 0.566346
```

Total phone use increased from wave 2 to 3-4!

```r
# phone use home
pairwise.wilcox.test(x=all_data_comp2$median_phoneuse_home, g=all_data_comp2$wave)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  all_data_comp2$median_phoneuse_home and all_data_comp2$wave
##
##   2       3
## 3 6.9e-16 -
## 4 2.3e-11 0.0088
##
## P value adjustment method: holm
```

```r
summary(all_data_comp2$median_phoneuse_home[all_data_comp2$wave==2])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.1428  0.2039  0.2235  0.2889  0.8110
```

```r
summary(all_data_comp2$median_phoneuse_home[all_data_comp2$wave==3])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.1594  0.2551  0.2647  0.3528  1.0000
```

```r
summary(all_data_comp2$median_phoneuse_home[all_data_comp2$wave==4])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.1606  0.2400  0.2482  0.3302  1.0000
```

Phone use at home increased from wave 2 to 3-4!

```r
# phone use not at home
pairwise.wilcox.test(x=all_data_comp2$median_phoneuse_nothome, g=all_data_comp2$wave)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  all_data_comp2$median_phoneuse_nothome and all_data_comp2$wave
##
##   2      3
## 3 <2e-16 -
## 4 <2e-16 <2e-16
##
## P value adjustment method: holm
```

```r
summary(all_data_comp2$median_phoneuse_nothome[all_data_comp2$wave==2])
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001982 0.081104 0.116881 0.128393 0.164299 0.527209
```

```r
summary(all_data_comp2$median_phoneuse_nothome[all_data_comp2$wave==3])
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.000000 0.000000 0.008696 0.055428 0.086183 0.588445
```

```r
summary(all_data_comp2$median_phoneuse_nothome[all_data_comp2$wave==4])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.01059 0.07421 0.09285 0.14331 0.45584
```

Phone use away from home dropped from wave 2-3 and then increased from 3-4!

```r
# age
pairwise.wilcox.test(x=as.numeric(all_data_comp2$age), g=all_data_comp2$wave)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  as.numeric(all_data_comp2$age) and all_data_comp2$wave
##
##   2      3
## 3 <2e-16 -
## 4 <2e-16 <2e-16
##
## P value adjustment method: holm
```

```
summary(as.numeric(all_data_comp2$age)[all_data_comp2$wave==2])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   18.00   19.00   18.75   19.00   21.00
```

```
summary(as.numeric(all_data_comp2$age)[all_data_comp2$wave==3])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   19.00   20.00   19.66   20.00   23.00
```

```
summary(as.numeric(all_data_comp2$age)[all_data_comp2$wave==4])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    18.0    19.0    20.0    20.2    21.0    23.0
```

Age was significantly different across waves, and the stats suggest that participants were slightly older on average from wave to wave.

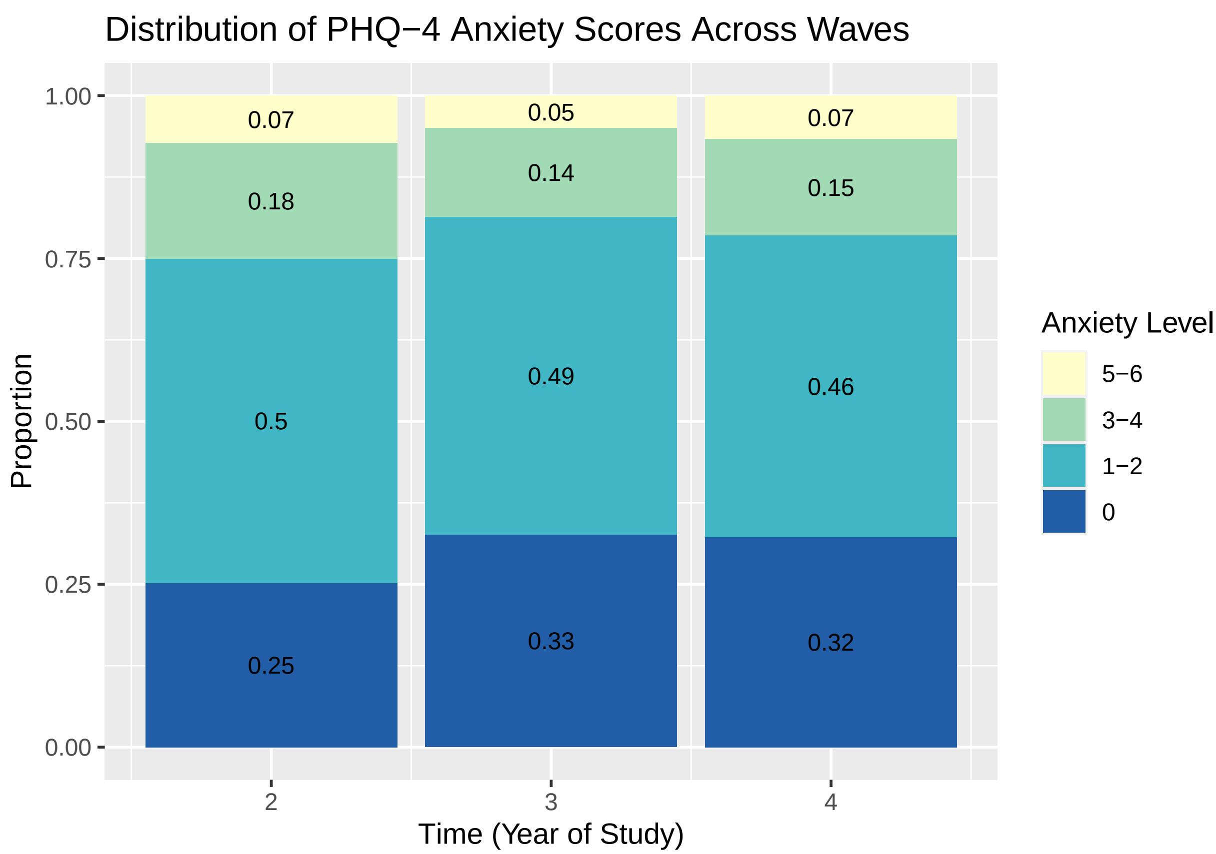
Let's visualize how anxiety levels change from wave to wave. Note that we bin anxiety levels (originally on 0-6 scale) into four bins (0 - no symptoms; 1-2 - light symptoms; 3-4 - moderate symptoms; 5-6 - severe symptoms) to reduce downstream model complexity and in line with PHQ-4 anxiety subscale interpretation; more detail provided in paper.

```
# rebin anxiety features into four levels
all_data_comp2$phq4_anxiety_EMA_binned = as.factor(ifelse(all_data_comp2$phq4_anxiety_EMA>=5, 3,
                                                          ifelse(all_data_comp2$phq4_anxiety_EMA==0,0,
                                                                 ifelse(all_data_comp2$phq4_anxiety_EMA>=
# data for plotting
df_proportions <- all_data_comp2 %>%
  group_by(wave, phq4_anxiety_EMA_binned) %>%
  summarise(n = n()) %>%
  mutate(proportion = n / sum(n))
```

```
## `summarise()` has grouped output by 'wave'. You can override using the
## `.groups` argument.
```

```
df_proportions$phq4_anxiety_EMA_binned <- factor(df_proportions$phq4_anxiety_EMA_binned,
                                                  levels = rev(levels(df_proportions$phq4_anxiety_EMA_bi

ggplot(df_proportions, aes(x = wave, y = proportion, fill = as.factor(phq4_anxiety_EMA_binned))) +
```

```r
  geom_bar(stat = "identity", position='stack') +
  scale_fill_manual(labels = c("5-6", "3-4", "1-2", "0"),
                    values = brewer.pal(4,"YlGnBu")) +
  labs(x = "Time (Year of Study)", y = "Proportion", fill = "Anxiety Level") +
  geom_text(aes(label = paste0(round(proportion,2))),
            position = position_stack(vjust = 0.5),
            color = "black",
            size = 3) +
  ggtitle("Distribution of PHQ-4 Anxiety Scores Across Waves")
```



Overall, the proportion of PHQ-4 records with no anxiety symptoms reported (anxiety level=0) jumped from wave 2 to 3-4. Fewer individuals reported clinically significant (anxiety level >3) symptoms from waves 2-3, then bumped from 3-4.

4. Models and results

Our outcome is ordinal (four levels, ranked by anxiety severity), and we have observations nested within individuals. We can create an ordinal logistic mixed-effects model:

```r
# ordinal logistic mixed effects model
mod1 = clmm(phq4_anxiety_EMA_binned ~ time*median_phoneuse_total + time*median_phoneuse_home
            + time*median_phoneuse_nothome + age + (time|new_id), data=all_data_comp2,
            method="nlminb", link='logit') # , control = list(method = "Nelder-Mead")

r2(mod1)
```

```
## # R2 for Mixed Models
##
##   Conditional R2: 0.683
##      Marginal R2: 0.007
```

```r
summary(mod1)
```

```
## Cumulative Link Mixed Model fitted with the Laplace approximation
##
## formula: phq4_anxiety_EMA_binned ~ time * median_phoneuse_total + time *
##     median_phoneuse_home + time * median_phoneuse_nothome + age +
##     (time | new_id)
## data:    all_data_comp2
##
##  link  threshold nobs logLik   AIC     niter       max.grad cond.H
##  logit flexible  4416 -3858.79 7745.58 1240(13039) 2.97e-03 3.5e+05
##
## Random effects:
##  Groups Name        Variance Std.Dev. Corr
##  new_id (Intercept) 6.655    2.580
##         time        1.339    1.157    -0.347
## Number of groups:  new_id 344
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## time                          0.05187    0.23554   0.220  0.82571
## median_phoneuse_total        -1.62331    1.73580  -0.935  0.34969
## median_phoneuse_home          1.77592    1.00773   1.762  0.07802 .
## median_phoneuse_nothome       3.95340    1.21095   3.265  0.00110 **
## age                           0.06422    0.13474   0.477  0.63363
## time:median_phoneuse_total    1.34746    1.19037   1.132  0.25765
## time:median_phoneuse_home    -1.13894    0.68702  -1.658  0.09736 .
## time:median_phoneuse_nothome -2.54899    0.84165  -3.029  0.00246 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##     Estimate Std. Error z value
## 0|1  -0.1926     2.5246  -0.076
## 1|2   3.9214     2.5257   1.553
## 2|3   6.3056     2.5273   2.495
```

At the p=0.05 level, we see that median_phoneuse_total and time*median_phoneuse_total are both significant! Let's look at the odds ratios to help w/ interpretation, divided by 100 to interpret as percents:

```
exp(coef(mod1)/100) # OR
```

```
##                         0|1                         1|2
##                   0.9980754                   1.0399926
##                         2|3                        time
##                   1.0650861                   1.0005188
##        median_phoneuse_total        median_phoneuse_home
##                   0.9838979                   1.0179178
##      median_phoneuse_nothome                         age
##                   1.0403259                   1.0006424
##   time:median_phoneuse_total   time:median_phoneuse_home
##                   1.0135658                   0.9886752
## time:median_phoneuse_nothome
##                   0.9748322
```

```
exp(confint(mod1, level=0.95)/100) # OR CI
```

```
##                                  2.5 %    97.5 %
## 0|1                          0.9498921 1.0487028
## 1|2                          0.9897645 1.0927696
## 2|3                          1.0136137 1.1191723
## time                         0.9959105 1.0051485
## median_phoneuse_total        0.9509877 1.0179470
## median_phoneuse_home         0.9980100 1.0382228
## median_phoneuse_nothome      1.0159253 1.0653125
## age                          0.9980033 1.0032886
## time:median_phoneuse_total   0.9901921 1.0374912
## time:median_phoneuse_home    0.9754516 1.0020781
## time:median_phoneuse_nothome 0.9588832 0.9910465
```

A ~4% increase in median proportion of time spent on phone away from home corresponded with higher odds of endorsing higher anxiety levels, while for a fixed median proportion of phone use away from home, an increase of 1-year decreased odds of endorsing higher anxiety levels by 2.5%.

Let's visualize the significant associations in the model by splitting into quartiles for phone use not at home. We see that <0.02462 marks the 0-25th quantiles and >0.14874 marks the 75-100th quantiles.

```
# quantiles for phone use not home - low is <0.02462 (25th), high is >0.14874 (75th)
summary(all_data_comp2$median_phoneuse_nothome)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.02462 0.08890 0.09911 0.14874 0.58845
```

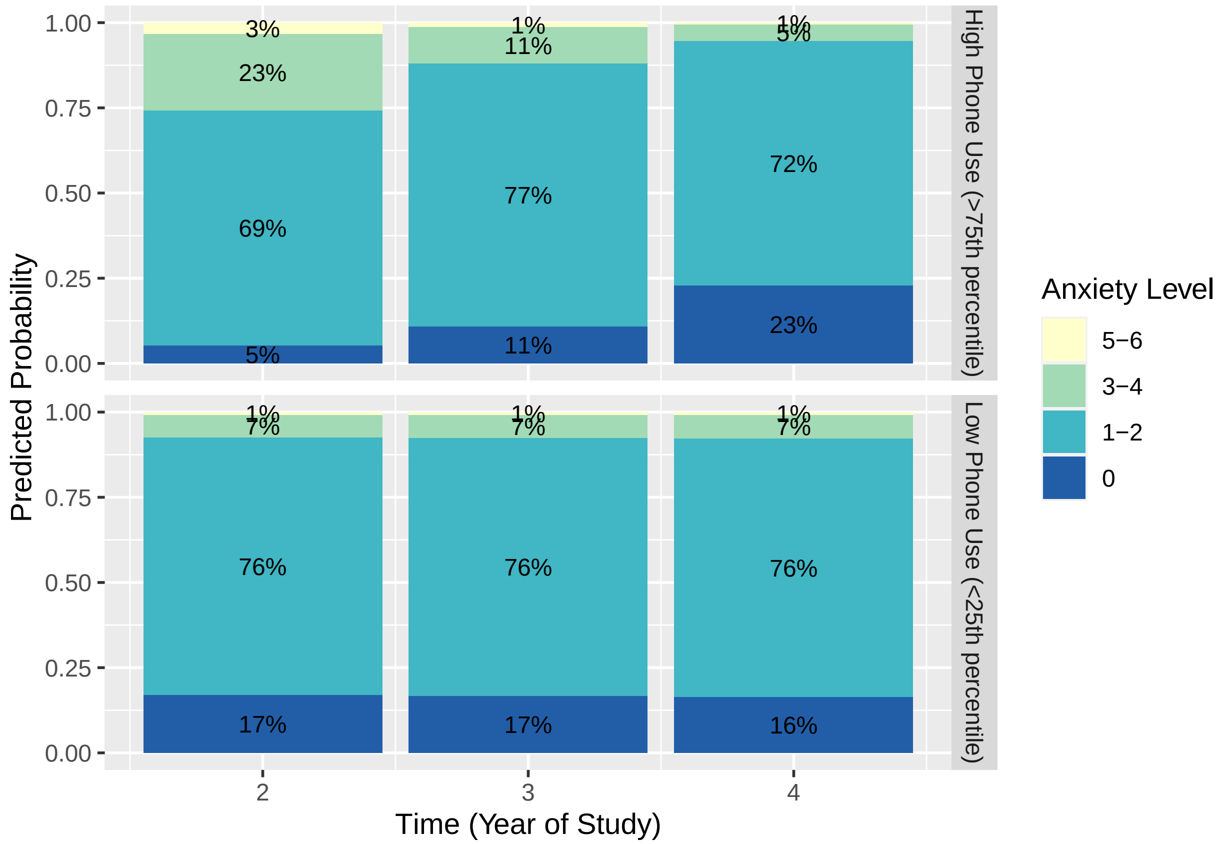
```
# get probabilities of each anxiety score at values of <25th %ile phone use
eff_low = effect(c('time','median_phoneuse_nothome'),mod=mod1, xlevels=list(time=c(0,1,2),
             median_phoneuse_nothome=seq(from=0,to=0.02462,length=1000)))
```

```
## NOTE: timemedian_phoneuse_nothome is not a high-order term in the model
```

```r
eff_df_low = cbind(eff_low$x, eff_low$prob)
eff_df_low$wave = ifelse(eff_df_low$time>=2, 4, ifelse(eff_df_low$time<1, 2, 3))

# get probabilities of each anxiety score at values of <75th %ile phone use
eff_high = effect(c('time','median_phoneuse_nothome'),mod=mod1, xlevels=list(time=c(0,1,2),
             median_phoneuse_nothome=seq(from=0.14874,to=0.58845,length=1000)))
```

```
## NOTE: timemedian_phoneuse_nothome is not a high-order term in the model
```

```r
eff_df_high = cbind(eff_high$x, eff_high$prob)
eff_df_high$wave = ifelse(eff_df_high$time>=2, 4, ifelse(eff_df_high$time<1, 2, 3))

# generate data for plotting - calculate per-wave means for probabilites
xl = eff_df_low %>%
  group_by(wave) %>%
  summarise(across(prob.X0:prob.X3, mean, .names = "mean_{.col}")) %>%
  gather(key=variable, value=value, -wave, convert=TRUE, factor_key=TRUE) %>%
  mutate(variable = factor(variable, levels = rev(levels(as.factor(variable)))))

xh = eff_df_high %>%
  group_by(wave) %>%
  summarise(across(prob.X0:prob.X3, mean, .names = "mean_{.col}")) %>%
  gather(key=variable, value=value, -wave, convert=TRUE, factor_key=TRUE) %>%
  mutate(variable = factor(variable, levels = rev(levels(as.factor(variable)))))

# merge data for plotting
xl$type = 'Low Phone Use (<25th percentile)'; xh$type = 'High Phone Use (>75th percentile)'
all_x = rbind(xl,xh)

# create stacked bar plot
ggplot(all_x,aes(x = wave, y = value, fill = variable)) +
  geom_bar(stat = "identity", position = 'stack') + # position_stack(reverse = TRUE)
  labs(fill = "Anxiety Level") + ylab("Predicted Probability") + xlab("Time (Year of Study)") +
  scale_fill_manual(labels = c("5-6", "3-4", "1-2", "0"),
                    values = brewer.pal(4,"YlGnBu"))+
                  # values = c("#FFEFCC","#A1DAB4","#41B6C4","#225EA8")) + # YlGnBu
  geom_text(aes(label = paste0(round(value * 100), "%")),
            position = position_stack(vjust = 0.5),
            color = "black",
            size = 3) +
  facet_grid(rows=vars(type))
```



We can clearly see that for individuals on the higher end of phone use away from home (>75th quantile), more reported >3 anxiety score in waves 2-3, but over the course of each wave, more reported anxiety score=0!

5. Appendix:

Supplementary analyses not included in/relevant to main paper are included here:

Note that the location dataset has two variables for time spent at home, timeathome and hometime. We proceed with timeathome in our analysis and describe our justification in the paper. Here, we conduct some exploration of the two variables:

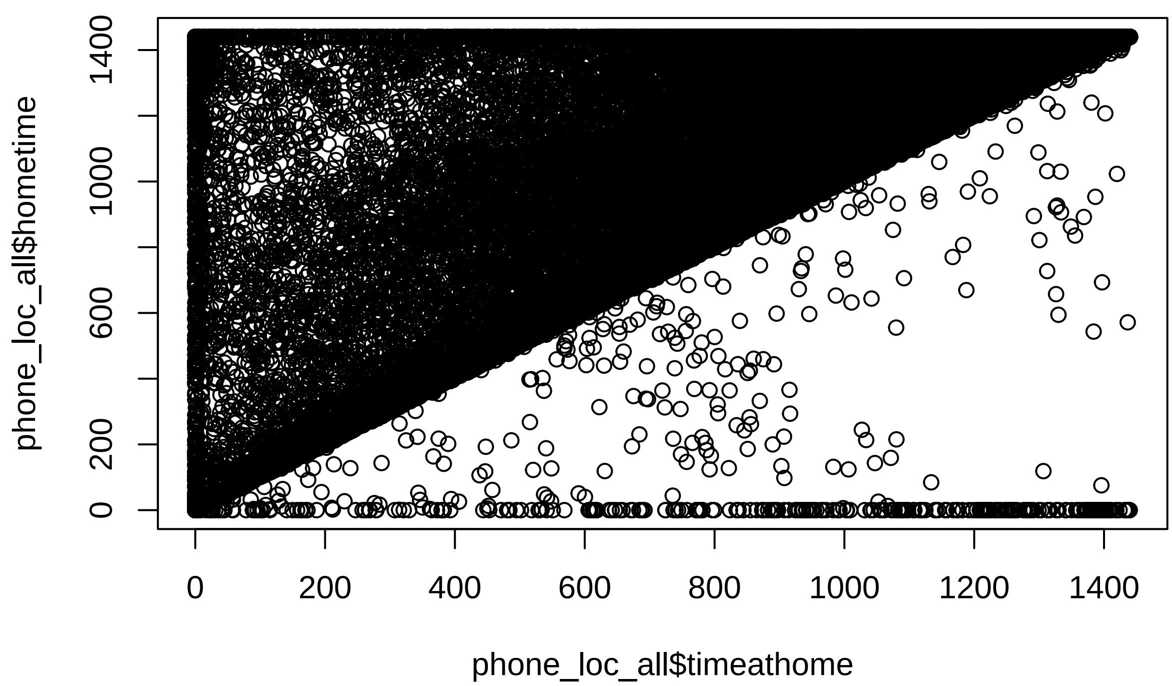
```r
summary(phone_loc_all$timeathome) # shouldn't exceed 1440
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0   436.2   862.0   795.8  1200.3  1440.0   13757
```

```r
summary(phone_loc_all$hometime) # shouldn't exceed 1440
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0   737.6  1111.0   979.4  1389.0  1440.0   18639
```

```r
plot(phone_loc_all$timeathome, phone_loc_all$hometime)
```




```r
rcorr(phone_loc_all$hometime,phone_loc_all$timeathome)
```

```
##      x    y
## x 1.00 0.77
```

```
## y 0.77 1.00
##
## n
##       x     y
## x 36703 36698
## y 36698 41585
##
## P
##   x  y
## x    0
## y  0
```

```r
summary(phone_loc_all$hometime-phone_loc_all$timeathome)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
## -1439.66    18.70    88.12   202.31   300.90  1440.00    18644
```

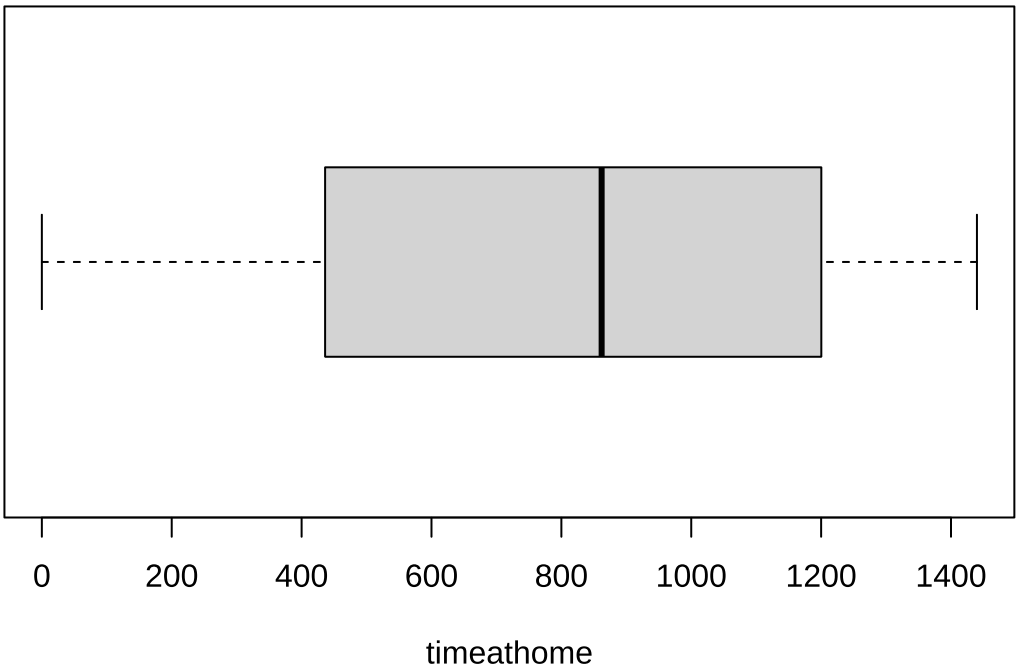
```r
sum(is.na(phone_loc_all$hometime))   # more NAs
```

```
## [1] 18639
```

```r
sum(is.na(phone_loc_all$timeathome)) # fewer NAs
```

```
## [1] 13757
```

```r
boxplot(phone_loc_all$timeathome, horizontal=T,
        xlab="timeathome") # no outliers!
```



Below are calculations for various correlation metrics between the predictor variables and anxiety levels not included in the final paper due to interpretability concerns. They are left here for the interested reader.

```
## spearman correlations ##
cor.test(all_data_comp2$median_phoneuse_home,as.numeric(all_data_comp2$phq4_anxiety_EMA), method = 'spea
```

```
## Warning in cor.test.default(all_data_comp2$median_phoneuse_home,
## as.numeric(all_data_comp2$phq4_anxiety_EMA), : Cannot compute exact p-value
## with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  all_data_comp2$median_phoneuse_home and as.numeric(all_data_comp2$phq4_anxiety_EMA)
## S = 1.3942e+10, p-value = 0.0573
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.02860884
```

```
cor.test(all_data_comp2$median_phoneuse_nothome,as.numeric(all_data_comp2$phq4_anxiety_EMA), method = '
```

```
## Warning in cor.test.default(all_data_comp2$median_phoneuse_nothome,
## as.numeric(all_data_comp2$phq4_anxiety_EMA), : Cannot compute exact p-value
## with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  all_data_comp2$median_phoneuse_nothome and as.numeric(all_data_comp2$phq4_anxiety_EMA)
## S = 1.3357e+10, p-value = 3.924e-06
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.0693887
```

```
cor.test(all_data_comp2$median_phoneuse_total,as.numeric(all_data_comp2$phq4_anxiety_EMA),method='spear
```

```
## Warning in cor.test.default(all_data_comp2$median_phoneuse_total,
## as.numeric(all_data_comp2$phq4_anxiety_EMA), : Cannot compute exact p-value
## with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  all_data_comp2$median_phoneuse_total and as.numeric(all_data_comp2$phq4_anxiety_EMA)
## S = 1.3593e+10, p-value = 0.0004344
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.05292101
```

```
## repeated measures correlations ##
rmcorr::rmcorr(new_id,median_phoneuse_home,as.numeric(phq4_anxiety_EMA),all_data_comp2)
```

```
## Warning in rmcorr::rmcorr(new_id, median_phoneuse_home,
## as.numeric(phq4_anxiety_EMA), : 'new_id' coerced into a factor
```

```
##
## Repeated measures correlation
##
## r
## 0.003426044
##
## degrees of freedom
## 4071
##
## p-value
## 0.8269742
##
## 95% confidence interval
## -0.02728926 0.03413489
```

```
rmcorr::rmcorr(new_id,median_phoneuse_nothome,as.numeric(phq4_anxiety_EMA),all_data_comp2)
```

```
## Warning in rmcorr::rmcorr(new_id, median_phoneuse_nothome,
## as.numeric(phq4_anxiety_EMA), : 'new_id' coerced into a factor
```

```
##
## Repeated measures correlation
##
## r
## 0.02731948
##
## degrees of freedom
## 4071
##
## p-value
## 0.08127757
##
## 95% confidence interval
## -0.003395808 0.05798326
```

```
rmcorr::rmcorr(new_id,median_phoneuse_total,as.numeric(phq4_anxiety_EMA),all_data_comp2)
```

```
## Warning in rmcorr::rmcorr(new_id, median_phoneuse_total,
## as.numeric(phq4_anxiety_EMA), : 'new_id' coerced into a factor
```

```
##
## Repeated measures correlation
##
## r
## -0.02727304
##
## degrees of freedom
## 4071
##
## p-value
## 0.08179641
##
## 95% confidence interval
## -0.05793695 0.003442278
```

```
# anxiety and time outside home
rmcorr::rmcorr(new_id,median_nottimeathome,as.numeric(phq4_anxiety_EMA),all_data_comp2)
```

```
## Warning in rmcorr::rmcorr(new_id, median_nottimeathome,
## as.numeric(phq4_anxiety_EMA), : 'new_id' coerced into a factor
```

```
##
## Repeated measures correlation
##
## r
## 0.01515132
##
## degrees of freedom
## 4071
##
## p-value
## 0.3336855
```

```
##
## 95% confidence interval
## -0.01556836 0.04584243
```

```r
rmcorr::rmcorr(new_id,median_timeathome,as.numeric(phq4_anxiety_EMA),all_data_comp2)
```

```
## Warning in rmcorr::rmcorr(new_id, median_timeathome,
## as.numeric(phq4_anxiety_EMA), : 'new_id' coerced into a factor
```

```
##
## Repeated measures correlation
##
## r
## -0.01515132
##
## degrees of freedom
## 4071
##
## p-value
## 0.3336855
##
## 95% confidence interval
## -0.04584243 0.01556836
```

```r
## polyserial correlation ##
#  polyserial correlation used for one continuous variable and one ordinal variable
polyserial(x=all_data_comp2$median_phoneuse_home, y=all_data_comp2$phq4_anxiety_EMA)
```

```
## [1] 0.0209338
```

```r
polyserial(x=all_data_comp2$median_phoneuse_nothome, y=all_data_comp2$phq4_anxiety_EMA)
```

```
## [1] 0.06937172
```

```r
polyserial(x=all_data_comp2$median_phoneuse_total, y=all_data_comp2$phq4_anxiety_EMA)
```

```
## [1] 0.0400897
```

```r
# polyserial(x=all_data_comp$median_timeathome, y=all_data_comp$phq4_anxiety_EMA)
# polyserial(x=all_data_comp$time, y=all_data_comp$phq4_anxiety_EMA)


## estimating repeated-measures spearman ##
# i=0
# for (id in unique(all_data_comp$new_id)) {
#   subdat = all_data_comp[all_data_comp$pid==id,]
#   sprmn = cor(subdat$median_phoneuse_home,as.numeric(subdat$phq4_anxiety_EMA))
# }

## some individuals only have one level of anxiety for all time, so can't calculate
## correlation for them
```

We can make another version of the effects plot but visualize the four quartiles of phone use away from home (i.e., 0-25, 25-50, 50-75, 75-100) instead of just 0-25 and 75-100. This is left here for the interested reader.

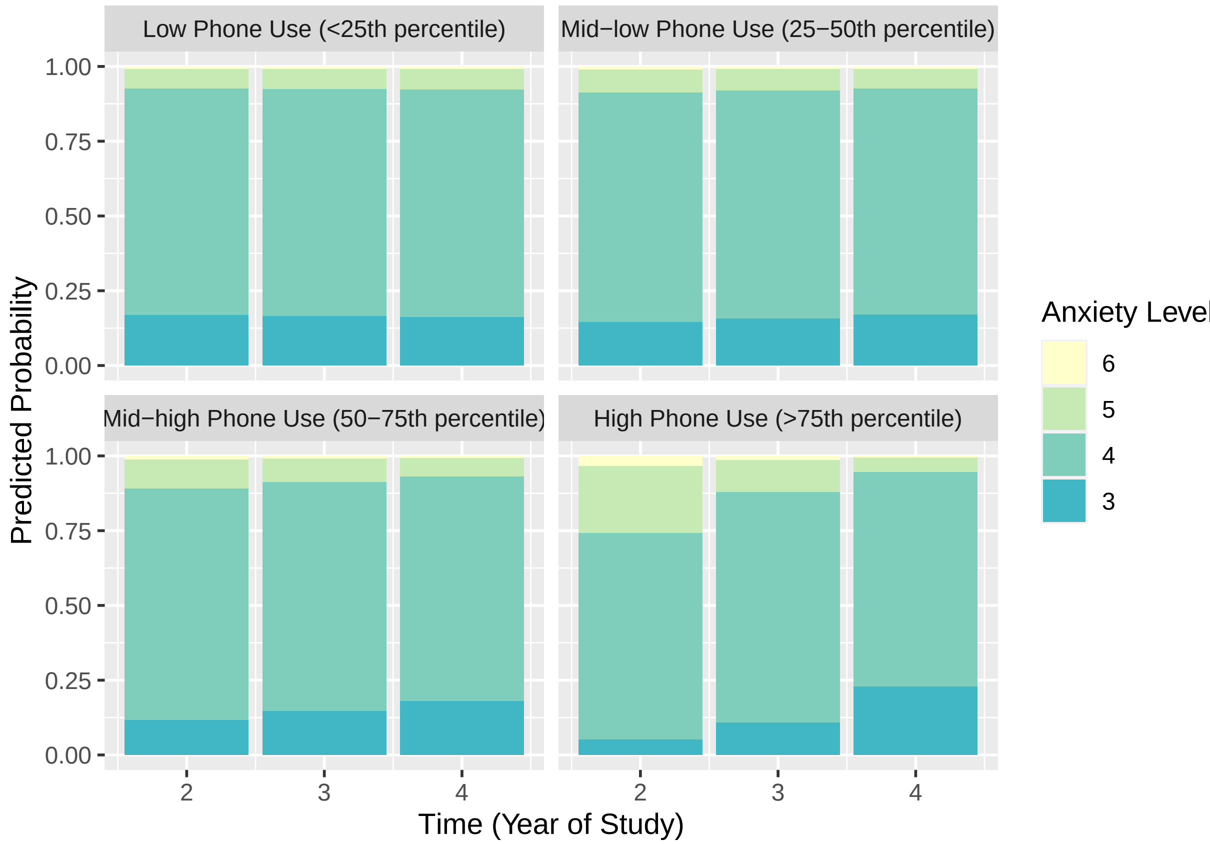
```
eff_l1 = effect(c('time','median_phoneuse_nothome'),mod=mod1, xlevels=list(time=c(0,1,2),
             median_phoneuse_nothome=seq(from=0.02462,to=0.08880,length=1000)))
```

```
## NOTE: timemedian_phoneuse_nothome is not a high-order term in the model
```

```
eff_df_l1 = cbind(eff_l1$x, eff_l1$prob)
eff_df_l1$wave = ifelse(eff_df_l1$time>=2, 4, ifelse(eff_df_l1$time<1, 2, 3))

eff_h1 = effect(c('time','median_phoneuse_nothome'),mod=mod1, xlevels=list(time=c(0,1,2),
             median_phoneuse_nothome=seq(from=0.08880,to=0.14866,length=1000)))
```

```
## NOTE: timemedian_phoneuse_nothome is not a high-order term in the model
```

```
eff_df_h1 = cbind(eff_h1$x, eff_h1$prob)
eff_df_h1$wave = ifelse(eff_df_h1$time>=2, 4, ifelse(eff_df_h1$time<1, 2, 3))

xl1 = eff_df_l1 %>%
  group_by(wave) %>%
  summarise(across(prob.X0:prob.X3, mean, .names = "mean_{.col}")) %>%
  gather(key=variable, value=value, -wave, convert=TRUE, factor_key=TRUE) %>%
  mutate(variable = factor(variable, levels = rev(levels(as.factor(variable)))))

xh1 = eff_df_h1 %>%
  group_by(wave) %>%
  summarise(across(prob.X0:prob.X3, mean, .names = "mean_{.col}")) %>%
  gather(key=variable, value=value, -wave, convert=TRUE, factor_key=TRUE) %>%
  mutate(variable = factor(variable, levels = rev(levels(as.factor(variable)))))

xl1$type = 'Mid-low Phone Use (25-50th percentile)'; xh1$type = 'Mid-high Phone Use (50-75th percentile

all_x = rbind(xl1,xh,xh1,xl)
all_x$type = factor(all_x$type, levels=c('Low Phone Use (<25th percentile)', 'Mid-low Phone Use (25-50t

ggplot(all_x,aes(x = wave, y = value, fill = variable)) +
  geom_bar(stat = "identity", position = 'stack') + # position_stack(reverse = TRUE)
  labs(fill = "Anxiety Level") + ylab("Predicted Probability") + xlab("Time (Year of Study)") +
  scale_fill_manual(labels = c("6", "5", "4", "3", "2", "1", "0"),
                    values = brewer.pal(7,"YlGnBu")) + # Blues
  facet_wrap(~type, nrow = 2)
```


Low Phone Use (<25th percentile)
Mid−low Phone Use (25−50th percentile)
Mid−high Phone Use (50−75th percentile)
High Phone Use (>75th percentile)
Predicted Probability
1.00
0.75
0.50
0.25
0.00
2
3
4
Time (Year of Study)
Anxiety Level
6
5
4
3