Data Appendix

Clara Li and Rose Porta

5/2/2020

```
# NOTE: To load data, you must download both the extract's data and the DDI
# and also set the working directory to the folder with these files (or change the path below).

if (!require("ipumsr")) stop("Reading IPUMS data into R requires the ipumsr package. It can be installe
```

```
## Loading required package: ipumsr
```

```
ddi <- read_ipums_ddi("nhis_00004.xml")
data <- read_ipums_micro(ddi)
```

```
## Use of data from IPUMS NHIS is subject to conditions including that users
## should cite the data appropriately. Use command `ipums_conditions()` for more
## details.
```

```
# Load necessary packages
library(base)
library(mosaic)
```

```
## Loading required package: dplyr

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: lattice

## Loading required package: ggformula

## Loading required package: ggplot2

## Loading required package: ggstance
```

```
## 
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':
## 
##     geom_errorbarh, GeomErrorbarh

## 
## New to ggformula?  Try the tutorials: 
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

## Loading required package: Matrix

## Registered S3 method overwritten by 'mosaic':
##   method                           from   
##   fortify.SpatialPolygonsDataFrame ggplot2

## 
## The 'mosaic' package masks several functions from core packages in order to add 
## additional features.  The original behavior of these functions should not be affected by this.
## 
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
## 
## Have you tried the ggformula package for your plots?

## 
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
## 
##     mean

## The following object is masked from 'package:ggplot2':
## 
##     stat

## The following objects are masked from 'package:dplyr':
## 
##     count, do, tally

## The following objects are masked from 'package:stats':
## 
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var

## The following objects are masked from 'package:base':
## 
##     max, mean, min, prod, range, sample, sum
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------- tidyverse 1.3.0 --

## v tibble  3.0.1     v purrr   0.3.4
## v tidyr   1.0.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x mosaic::count()           masks dplyr::count()
## x purrr::cross()            masks mosaic::cross()
## x mosaic::do()              masks dplyr::do()
## x tidyr::expand()           masks Matrix::expand()
## x dplyr::filter()           masks stats::filter()
## x ggstance::geom_errorbarh() masks ggplot2::geom_errorbarh()
## x dplyr::lag()              masks stats::lag()
## x tidyr::pack()             masks Matrix::pack()
## x mosaic::stat()            masks ggplot2::stat()
## x mosaic::tally()           masks dplyr::tally()
## x tidyr::unpack()           masks Matrix::unpack()
```

Initial Data Wrangling

```r
# Create activity level variable and filter out non-applicable values
data <- data %>%
  mutate( activity_score = (9 * VIG10FWK + 5 * MOD10FWK + 3 * STRONGFWK)) %>%
  filter(AGE >= 18,
         BMI != "0", BMI != "0.00", BMI != "99.80", BMI != "99.99",
         INCFAM07ON != "99", INCFAM07ON != "96",
         RACEA != "900", RACEA != "970", RACEA != "980", RACEA != "990", RACEA != "600",
         HEALTH != "0", HEALTH != "7", HEALTH != "8", HEALTH != "9",
         STRONGFWK != "00", STRONGFWK != "96", STRONGFWK != "97", STRONGFWK != "98", STRONGFWK != "99",
         MOD10FWK != "00", MOD10FWK != "96", MOD10FWK != "97", MOD10FWK != "98", MOD10FWK != "99",
         VIG10FWK != "00", VIG10FWK != "96", VIG10FWK != "97", VIG10FWK != "98", VIG10FWK != "99",
         HRSLEEP != "00", HRSLEEP != "97", HRSLEEP != "98", HRSLEEP != "99",
DEPFREQ != "0", DEPFREQ != "7", DEPFREQ != "8", DEPFREQ != "9",
         FLDIABETNO != "00", FLDIABETNO != "97", FLDIABETNO != "98", FLDIABETNO != "99",
         FLDIABETTP != "7", FLDIABETTP != "8", FLDIABETTP != "9") %>%
  mutate(SEX = ifelse(SEX == 1, 0, 1))
```

```r
# convert time to years and recode chronic variable to 0 and 1
data <- data %>%
  mutate(years = ifelse(FLDIABETTP == 1, FLDIABETNO / 365,
                      ifelse(FLDIABETTP == 2, FLDIABETNO/52,
                             ifelse(FLDIABETTP == 3, FLDIABETNO/12,
                                    FLDIABETNO))
         )) %>%
  mutate(chronic = ifelse(FLDIABETC == 2, 1, 0))
```

Variable Analysis

```
# Look at distributions of variables
favstats(~BMI, data = data)
```

```
##    min    Q1 median    Q3   max     mean       sd     n missing
##  14.08 23.74  27.11 31.31 85.78 28.18155 6.388462 21297       0
```

The minimum BMI is 16.16 and the maximum is 69.88. There are no missing values.

```
favstats(~HRSLEEP, data = data)
```

```
##  min Q1 median Q3 max     mean       sd     n missing
##    1  6      7  8  22 7.067944 1.421114 21297       0
```

The minimum value for hours of sleep is 2, and the maximum is 18. These both seem like very extreme values, but Q1 is 6 and Q3 is 8, which indicates that most individuals get between 6 and 8 hours of sleep, which makes sense. There are no missing values, which is ideal.

```
favstats(~AGE, data = data)
```

```
##  min Q1 median Q3 max     mean       sd     n missing
##   18 35     51 65  85 50.83894 18.18671 21297       0
```

The minimum age is 32 years and the maximum is 85 years. There are no missing values, which is ideal.

```
favstats(~STRONGFWK, data = data)
```

```
##  min Q1 median Q3 max    mean       sd     n missing
##    1  5     95 95  95 66.3763 42.30279 21297       0
```

```
favstats(~MOD10FWK, data = data)
```

```
##  min Q1 median Q3 max     mean       sd     n missing
##    1  3      7 95  95 37.34188 43.23382 21297       0
```

```
favstats(~VIG10FWK, data = data)
```

```
##  min Q1 median Q3 max     mean       sd     n missing
##    1  4     95 95  95 52.91694 45.26272 21297       0
```

For each of the 3 above variables that we used to create our activity_score variable, the minimum is 1 and the maximum is 95. The minimum makes sense, but it seems very extreme and almost impossible for someone to exercise 95 times per week. There are no missing values.

```
favstats(~activity_score, data = data)
```

```
##  min  Q1 median   Q3  max     mean      sd     n missing
##   17 318    896 1610 1615 862.0907 593.814 21297       0
```

We created the activity_score variable using the Oncology Nursing Society formula Weekly leisure activity score = (9 × Strenuous) + (5 × Moderate) + (3 × Light) from the Godin Leisure-Time Exercise Questionnaire. We substituted strength exercise for light exercise due to the availability of the IPUMS data, and we think strength training is analagous to light exercise for the purposes of our analysis. Each of the 3 variables used to create this are measured in number of times per week. The minimum value is 17 and the maximum is 1615. We are not sure exactly how to make sense of these numbers, but we are assuming that a higher score means a higher overall activity level. There are no missing values.

```
tally(~SEX, data = data)
```

```
## SEX
##     0     1
##  9898 11399
```

The number of females and males is roughly equal (with slighly more females), which makes sense.

```
tally(~INCFAM07ON, data = data)
```

```
## INCFAM07ON
##   10   11   12   21   22   23   24
##  277 6726 2477   57 3558 2878 5324
```

Each category corresponds to a family income-level category, with 10 being the lowest category and 24 being the highest. We thought it was necessary to collapse the categories into low, middle, and high income in order to avoid over-complicating our analysis. We noticed that there are very few people in the lowest category, but the greatest number of people in the second-lowest. Based on this distribution, we decided to combine the two lowest categories into the low income category, the 3 middle categories into the middle income category, and the two highest categories into the high income category.

```
tally(~RACEA, data = data)
```

```
## RACEA
##   100   200   310   411   412   416   434   580
## 17215  2542   303   244   270   286   369    68
```

Each category code corresponds to a race. We noticed that the vast majority of individuals are white (category 100).

```
tally(~HEALTH, data = data)
```

```
## HEALTH
##    1    2    3    4    5
## 5412 7256 5803 2206  620
```

1 corresponds to very good health and 5 corresponds to very poor health. We noticed that most people are in good or excellent health, and fewer people are in poor health. In order to simplify our analysis, we decided to collapse health into a binary variable: good health versus poor health. We combined the 1, 2, and 3 categories into good health (because the middle (3) category was coded in IPUMS as “good”), and the other two into poor health.

```
tally(~DEPFREQ, data = data)
```

```
## DEPFREQ
##     1     2     3     4     5
##   924  1249  1382  5532 12210
```

1 corresponds to the highest worry frequency (daily), and 5 corresponds to the lowest. We noticed that there are fewer people who worry a lot, and more who do not report worrying a lot. We think we will need to collapse some of the categories, but since it is not obvious from the tally which ones to combine, we will look at a faceted scatter plot using worry frequency as the explanatory and weight as the response in order to see if particular categories show particularly strong associations with weight.

```
favstats(~years, data = data)
```

```
##  min Q1 median Q3 max      mean       sd     n missing
##    0  0      0  0  65 0.3236364 2.844736 21297       0
```

```
tally(~chronic, data = data)
```

```
## chronic
##     0     1
## 20886   411
```

Additional Data Wrangling

```
# Collapse categorical variables
data <- data %>%
  # income_low is reference group
  mutate(income_high = ifelse(INCFAM07ON == "23" |  INCFAM07ON == "24", 1, 0),
         income_middle = ifelse(INCFAM07ON == "12" | INCFAM07ON == "21" | INCFAM07ON == "22", 1, 0)) %>
  # health_binary = 1 indicates good health
  mutate(health_binary = ifelse(HEALTH == "1" | HEALTH == "2" | HEALTH == "3", 1, 0)) %>%
  select(years, chronic, DEPFREQ, HRSLEEP, AGE, BMI, activity_score, SEX, RACEA,
         income_high, income_middle, health_binary)
```

Structure and Names

```
str(data, give.attr = FALSE)
```

```
## tibble [21,297 x 12] (S3: tbl_df/tbl/data.frame)
##  $ years         : num [1:21297] 0 0 0 0 0 0 0 0 0 26 ...
##  $ chronic       : num [1:21297] 0 0 0 0 0 0 0 0 0 1 ...
##  $ DEPFREQ       : 'haven_labelled' int [1:21297] 5 4 5 1 5 4 4 4 5 5 ...
##  $ HRSLEEP       : 'haven_labelled' int [1:21297] 8 5 7 5 8 6 8 7 6 10 ...
##  $ AGE           : 'haven_labelled' num [1:21297] 79 37 29 75 39 54 85 28 65 68 ...
##  $ BMI           : 'haven_labelled' num [1:21297] 23.6 32.8 43.6 22.3 23.7 ...
##  $ activity_score: 'haven_labelled' num [1:21297] 1615 301 356 1615 53 ...
##  $ SEX           : num [1:21297] 1 0 0 0 0 1 0 1 0 0 ...
##  $ RACEA         : 'haven_labelled' int [1:21297] 100 100 100 412 100 200 200 100 100 100 ...
##  $ income_high   : num [1:21297] 0 1 1 0 1 1 1 1 1 1 ...
##  $ income_middle : num [1:21297] 0 0 0 0 0 0 0 0 0 0 ...
##  $ health_binary : num [1:21297] 1 1 1 0 1 1 0 1 1 0 ...
```

After some initial data wrangling, our data set has 11 variables and 19925 observational units. The variables are:

1. WEIGHT is a number indicating the person's weight in pounds
2. HRSLEEP is an integer indicating self-reported number of hours of sleep per night an individual gets.
3. AGE is a number indicating the person's age in years
4. BMI is a number indicating the person's BMI
5. activity_score is a variable we created, and is a number that represents a person's activity level.
6. SEX is a number that indicates whether an individual is male(0) or female(1)
7. RACEA is an integer that gives a code that corresponds to an individual's race category
8. income_high is a number that indicates if an individual is high income based on our grouping (1) or not high-income (0).
9. income_middle is a number that indicates if an individual is middle income based on our grouping (1) or not middle-income (0).
10. health_binary is a character that indicates whether and individual's health is "good" or "poor"
11. WORFREQ is an integer that corresponds to a category (1-5) for how often an individual worries, 1 being daily and 5 being never.

Final Data Wrangling

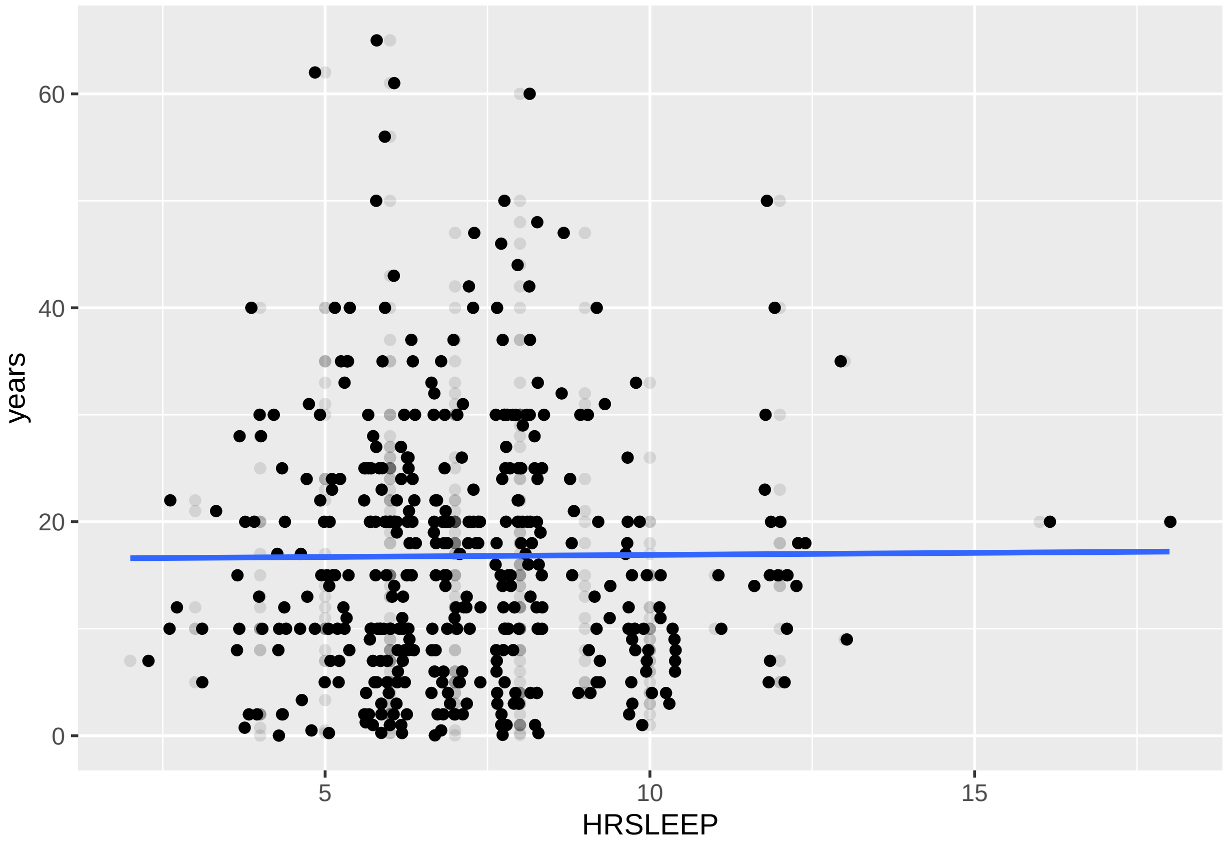
```r
data <- data %>%
  # collapse depfreq into 3 categories, "a few times a year" or "never" is reference
  mutate(depfreq_often = ifelse(DEPFREQ == 1 | DEPFREQ == 2, 1, 0),
         depfreq_monthly = ifelse(DEPFREQ == 3 , 1, 0)) %>%
  mutate(
    race_asian = ifelse(RACEA == "411" | RACEA == "412" | RACEA == "416" | RACEA == "434", 1, 0),
    race_black = ifelse(RACEA == "200", 1, 0),
    race_na = ifelse(RACEA == "310", 1, 0)
    ) %>%
  filter(RACEA != "580") %>%
  select(years, chronic, depfreq_often, depfreq_monthly, HRSLEEP, AGE, BMI, activity_score, SEX, race_as
         income_high, income_middle, health_binary, RACEA, DEPFREQ)
```

Visualization

```r
data2 <- data %>%
  filter(years > 0)
# Create scatter plot of weight versus hours of sleep
ggplot(data2, aes(x = HRSLEEP, y = years))+
  geom_point(alpha = 0.1) +
  geom_jitter() +
  geom_smooth(method = lm, se = FALSE)
```

```
## Don't know how to automatically pick scale for object of type haven_labelled. Defaulting to continuou
```

```
## `geom_smooth()` using formula 'y ~ x'
```



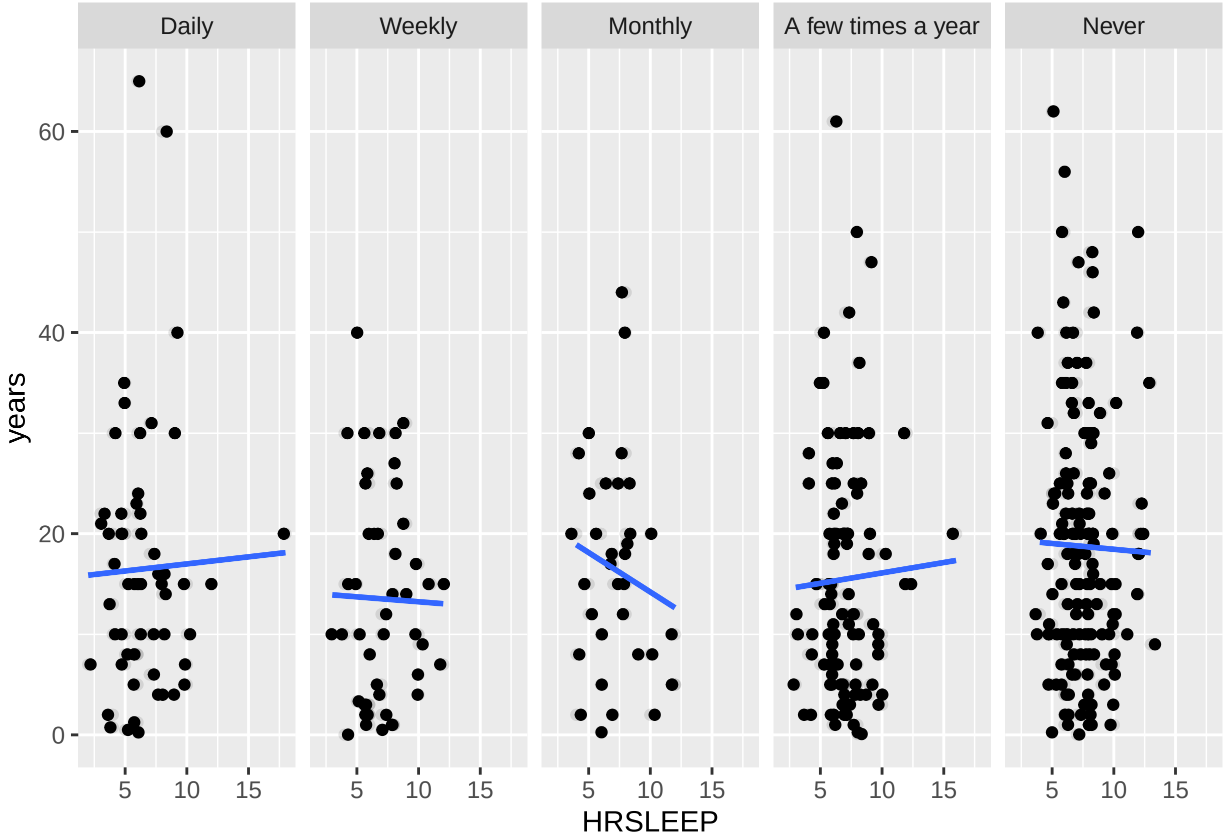
Although the strength and magnitude of the fitted line are weak, the slope is negative, indicating an inverse relationship between hours of sleep and weight in pounds as we hypothesized.

```
depfreq_labels <- c(
  "1" = 'Daily',
  "2" = 'Weekly',
  "3" = 'Monthly',
  "4" = 'A few times a year',
  "5" = 'Never'
)

# Facet scatterplot by worry frequency
ggplot(data2, aes(x = HRSLEEP, y = years))+
  geom_point(alpha = 0.1) +
  geom_jitter() +
  geom_smooth(method = lm, se = FALSE) +
  facet_grid(cols=vars(DEPFREQ), labeller=labeller(DEPFREQ = depfreq_labels))
```

```
## Don't know how to automatically pick scale for object of type haven_labelled. Defaulting to continuo
```

```
## `geom_smooth()` using formula 'y ~ x'
```



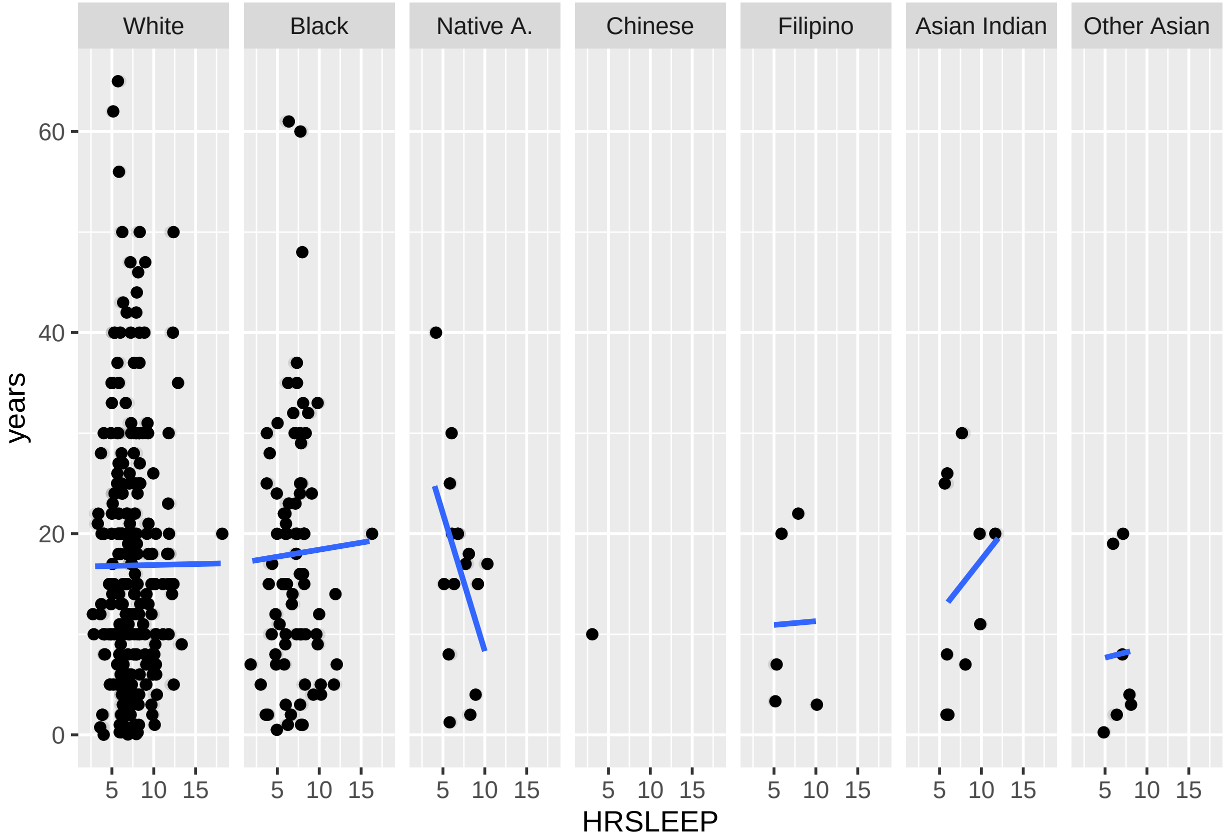
For all worry frequencies except WORFREQ = 1 (daily), the relationship between hours of sleep and weight remains negative. For observational units who worried daily, however, the relationship between HRSLEEP and WEIGHT appears slightly positive. This hints at a possibly significant interaction between HRSLEEP/WEIGHT and a binary WORFREQ variable with 1 coded as worrying daily and 0 coded as worrying less than daily.

```
race_labels <- c("100" = 'White',
              "200" = 'Black',
              "310" = 'Native A.',
              "411" = 'Chinese',
              "412" = 'Filipino',
              "416" = 'Asian Indian',
              "434" = 'Other Asian',
              "580" = 'NR')

# Facet scatterplot by race
ggplot(data2, aes(x = HRSLEEP, y = years))+
  geom_point(alpha = 0.1) +
  geom_jitter() +
  geom_smooth(method = lm, se = FALSE) +
  facet_grid(col=vars(RACEA), labeller=labeller(RACEA = race_labels))
```

```
## Don't know how to automatically pick scale for object of type haven_labelled. Defaulting to continuou
```

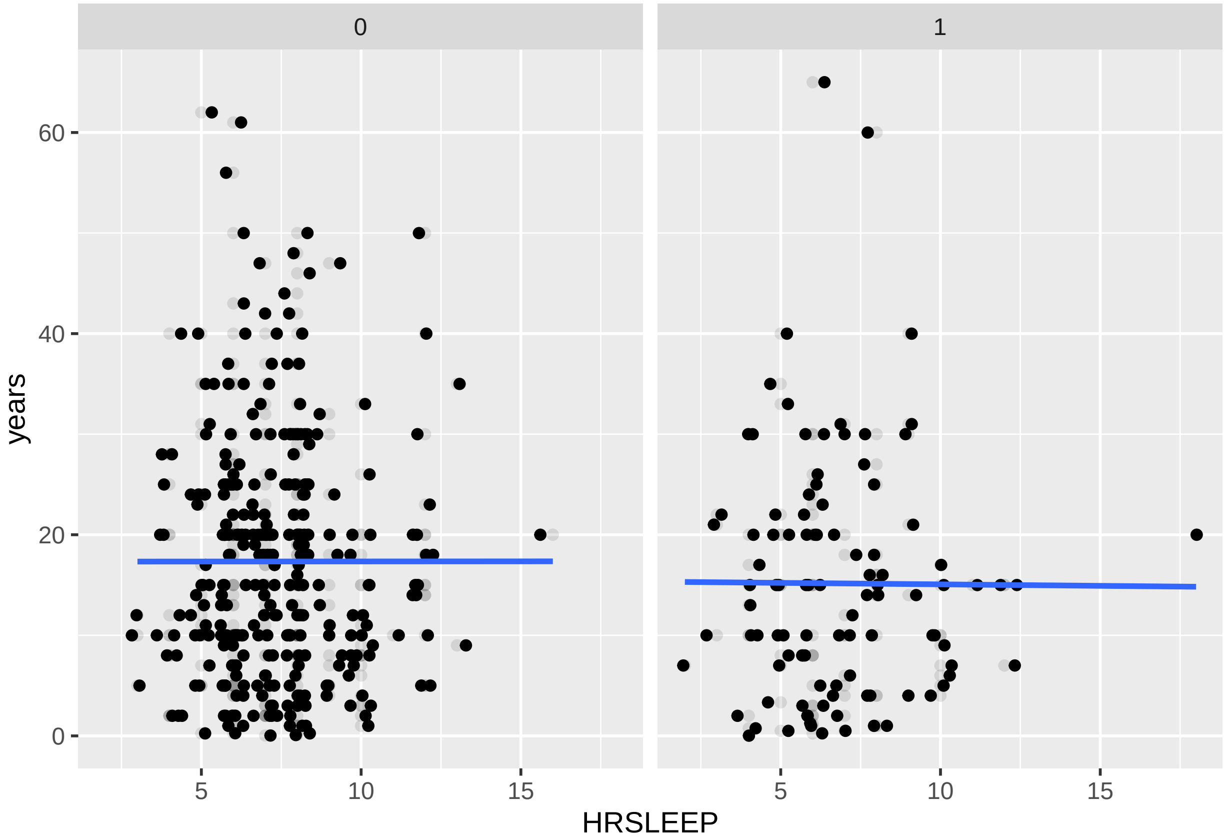
```
## `geom_smooth()` using formula 'y ~ x'
```



```
# Facet scatterplot by depfreq_often
ggplot(data2, aes(x = HRSLEEP, y = years))+
  geom_point(alpha = 0.1) +
  geom_jitter() +
  facet_wrap(~depfreq_often) +
  geom_smooth(method = lm, se = FALSE)
```

```
## Don't know how to automatically pick scale for object of type haven_labelled. Defaulting to continuou
```

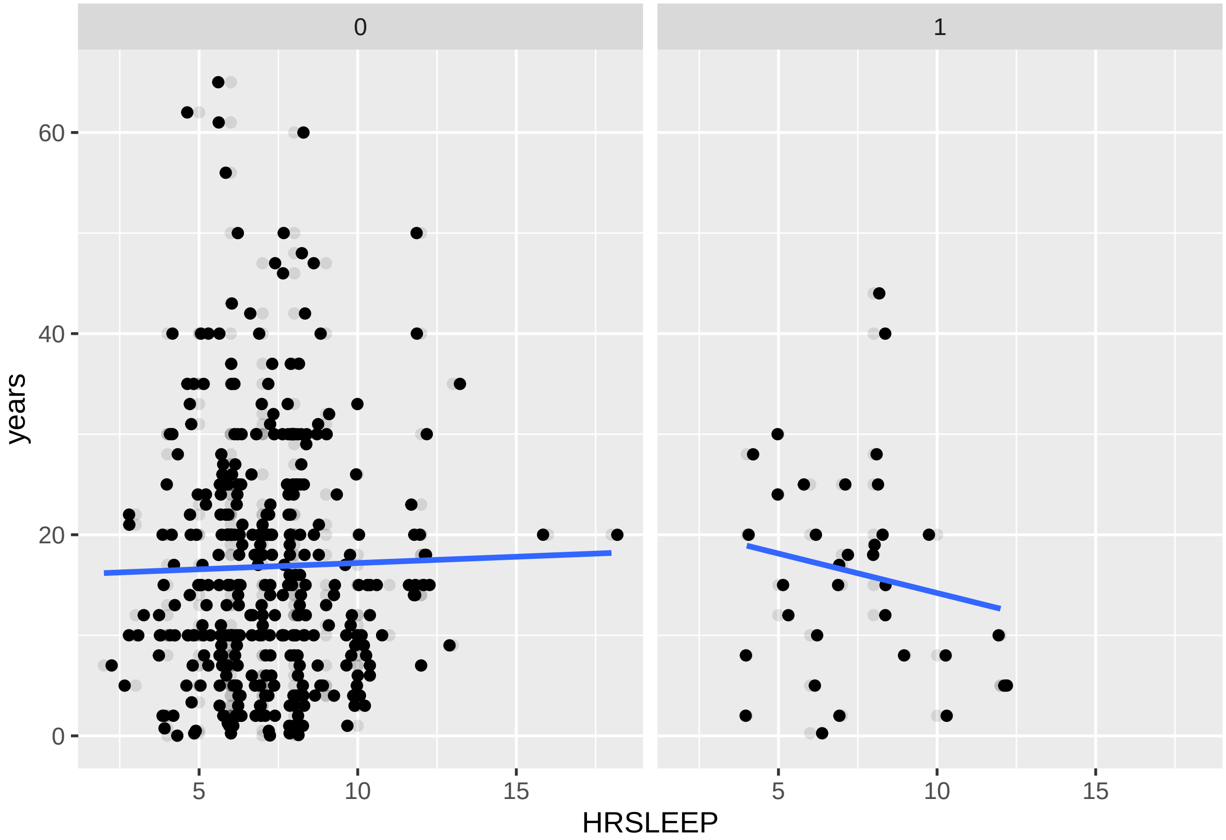
```
## `geom_smooth()` using formula 'y ~ x'
```



```r
# Facet scatterplot by depfreq_monthly
ggplot(data2, aes(x = HRSLEEP, y = years))+
  geom_point(alpha = 0.1) +
  geom_jitter() +
  facet_wrap(~depfreq_monthly) +
  geom_smooth(method = lm, se = FALSE)
```

```
## Don't know how to automatically pick scale for object of type haven_labelled. Defaulting to continuou
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Based on the previous scatterplots faceted by race, we decided to combine Asian ethnicities and keep Alaskan Native or American Indians separate. NR stands for primary race not releasable. We also recoded each categorical variable as a binary variable.

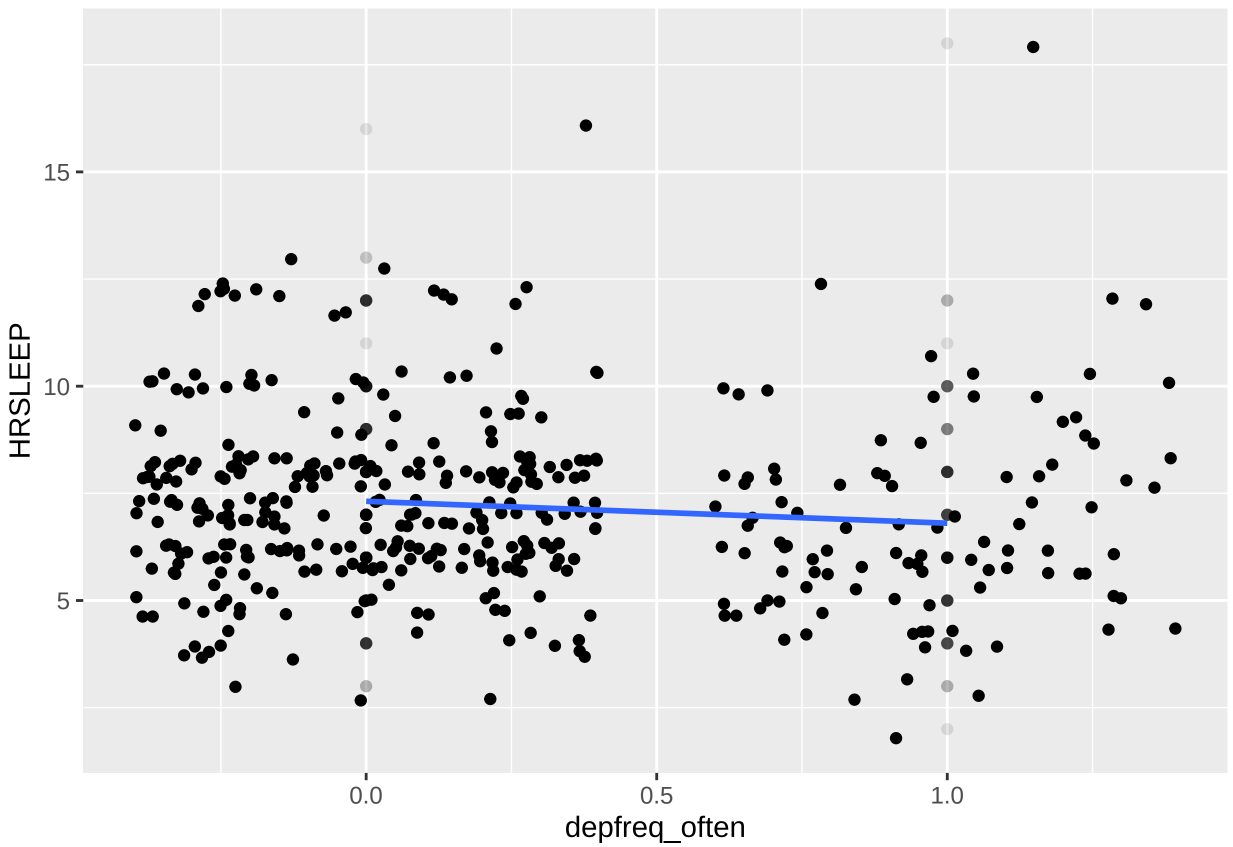
After recording worry frequency as a binary variable, we re-evaluated the relationship between sleep and weight faceted by daily vs non-daily worry frequency. We see a fairly clear difference in the two slopes of the two lines, suggesting the need for an interaction term between worry frequency and sleep in our multiple regression model.

Checking Collinearity between Sleep and Worry Frequency

```
ggplot(data2, aes(x = depfreq_often, y = HRSLEEP)) +
  geom_point(alpha = 0.1) +
  geom_jitter() +
  geom_smooth(method = lm, se = FALSE)
```

```
## Don't know how to automatically pick scale for object of type haven_labelled. Defaulting to continuou
```

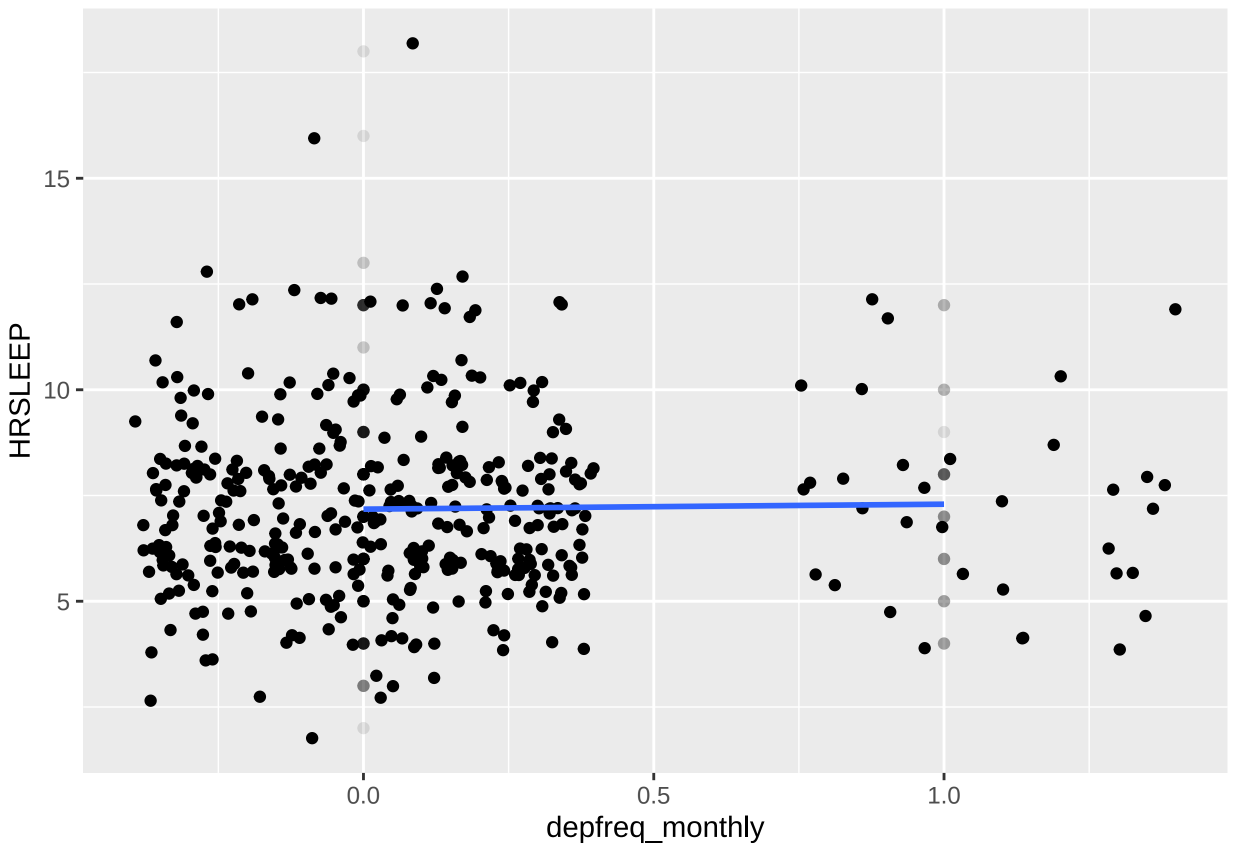
```
## `geom_smooth()` using formula 'y ~ x'
```



```
ggplot(data2, aes(x = depfreq_monthly, y = HRSLEEP)) +
  geom_point(alpha = 0.1) +
  geom_jitter() +
  geom_smooth(method = lm, se = FALSE)
```

```
## Don't know how to automatically pick scale for object of type haven_labelled. Defaulting to continuou
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Based on the scatterplots of binary worry frequency vs hours of sleep, there does not appear to be a significant relationship between the two, and so we do not suspect collinearity between our two explanatory variables.

```
# Exporting wrangled dataset for data analysis use
write.csv(data, "data.csv")
```

Tallies after data wrangling

```
# Look at distributions of variables
favstats(~BMI, data = data)
```

```
##    min    Q1 median    Q3   max     mean       sd     n missing
##  14.08 23.74  27.11 31.31 85.78 28.17239 6.379732 21229       0
```

```
favstats(~HRSLEEP, data = data)
```

```
##  min Q1 median Q3 max     mean       sd     n missing
##    1  6      7  8  22 7.069339 1.421356 21229       0
```

```
favstats(~AGE, data = data)
```

```
##  min Q1 median Q3 max     mean       sd     n missing
##   18 35     51 65  85 50.85831 18.18446 21229       0
```

```
favstats(~activity_score, data = data)
```

```
##  min  Q1 median   Q3  max     mean       sd     n missing
##   17 318    896 1610 1615 862.1115 593.6359 21229       0
```

```
tally(~SEX, data = data)
```

```
## SEX
##     0     1
##  9860 11369
```

```
tally(~income_high, data = data)
```

```
## income_high
##     0     1
## 13048  8181
```

```
tally(~income_middle, data = data)
```

```
## income_middle
##     0     1
## 15156  6073
```

```
tally(~race_asian, data = data)
```

```
## race_asian
##     0     1
## 20060  1169
```

```
tally(~race_black, data = data)
```

```
## race_black
##     0     1
## 18687  2542
```

```
tally(~race_na, data = data)
```

```
## race_na
##     0     1
## 20926   303
```

```
tally(~health_binary, data = data)
```

```
## health_binary
##     0     1
##  2817 18412
```

```
tally(~depfreq_often, data = data)
```

```
## depfreq_often
##     0     1
## 19062  2167
```

```
tally(~depfreq_monthly, data = data)
```

```
## depfreq_monthly
##     0     1
## 19853  1376
```

```
favstats(~years, data = data)
```

```
##  min Q1 median Q3 max      mean      sd     n missing
##    0  0      0  0  65 0.3219881 2.84201 21229       0
```

```
tally(~chronic, data = data)
```

```
## chronic
##     0     1
## 20822   407
```

Tables for results section

```
# code from https://cran.r-project.org/web/packages/qwraps2/vignettes/summary-statistics.html#count-and

our_summary1 <-
  list("Hours of Sleep" =
       list("min" = ~ min(.data$HRSLEEP),
            "max" = ~ max(.data$HRSLEEP),
            "mean (sd)" = ~ qwraps2::mean_sd(.data$HRSLEEP))

       )
```

```
library(qwraps2)
```

```
##
## Attaching package: 'qwraps2'

## The following object is masked from 'package:mosaic':
##
##     logit
```

```
summary_table(data, our_summary1)
```

```
##
## \begin{tabular}{l|l}
## \hline
##  & data (N = 21,229)\\
## \hline
## \bf{Hours of Sleep} & ~\\
## \hline
## ~~ min & 1\\
## \hline
## ~~ max & 22\\
## \hline
## ~~ mean (sd) & 7.07 $\pm$ 1.42\\
## \hline
## \end{tabular}
```