STAT 412/512 HW 1

Due Sept 4 at 9 AM

HW 1 Instructions:

You may discuss the work with other students but should complete this assignment in an independent fashion (you should try the code and writing on your own but can benefit from discussions with other students). At the end of the assignment you will document any discussions that have a substantive impact on your results. Also document any resources you used outside of those that I provide, including things like ChatGPT.

Answer each question inline after the prompts with pertinent code, results, and graphs preceding your written answer.

Make sure to run spell-check using the "ABC check" button near the filename or use the text underlining suggestions in the document. Knit this .Rmd to word and submit just the knitted word document for grading in the D2L assignment folder.

Uncomment the lines of code above to install `ggResidpanel` and `catstats2` from my github repository. Then re-comment them for all following use of the file. Contact me ASAP if you have difficulties installing and loading the packages or finding the data set.

Back pain and sitting or sitting and back pain?

To get a chance to review and practice/learn to use R, we will explore the data set posted to accompany Gupta *et al.*'s "Is Objectively Measured Sitting Time Associated with Low Back Pain? A Cross-Sectional Investigation in the NOMAD study".

- Gupta, N., Christiansen, C., Hallman, D., Korshoj, M., Carneiro, I., and Holtermann, A. (2015) Is Objectively Measured Sitting Time Associated with Low Back Pain? A Cross-Sectional Investigation in the NOMAD study. *PLoS ONE* 10(3): e0121159. doi:10.1371/journal.pone.0121159

- Their primary analysis was using logistic regression with low back pain/not as a binary response variable. We'll explore this in more detail later, but:

1) What did they conclude about the relationship between sitting time and low back pain? Just a sentence or two that summarizes their findings relating these two variables.

The authors found a strong positive association between sitting time and lower-back pain. These results were corroborated by an additional strong association between leisure sitting and lower back pain, and although the p-value was >0.05 when analyzing occupational sitting and lower-back pain, these results suggest a long duration of sitting time will likely result in lower-back pain.

- The data set is provided two ways. The authors' posted it as an Excel spreadsheet and I extracted the first sheet as a .csv.
- The data set on D2L ("gupta_2015_sheet1.csv") is a .csv file. Download the file and save it into a *new* folder, where you will store the data **and** today's data analysis file.

```
## 1. Make sure this RMarkdown file is saved in the SAME folder as the
gupta_2015.csv file that contains the data
## 2. Now load the data into R

gupta_2015 <- read_csv("gupta_2015_sheet1.csv")

View(gupta_2015) #Great way to do a quick check of results of reading in

head(gupta_2015)
```

```
## # A tibble: 6 × 13
##   `serial number of the worker` `age in years` sex   `job seniority in
months`
##                           <dbl>          <dbl> <chr>
<dbl>
## 1                         11001             48 male
120
## 2                         11002             32 male
41
## 3                         11003             51 male
102
## 4                         11004             65 male
480
## 5                         11005             58 male
524
## 6                         11007             45 male
240
## # ℹ 9 more variables: `body mass index in kg/m2` <dbl>,
## #   `total sitting time in hours` <dbl>, `Influence at work in 0-100%`
<dbl>,
## #   `low back pain categories` <chr>, `smoking status categories` <chr>,
## #   `occupational lifting/carrying time categories` <chr>,
## #   `total sitting time as percentage of the mean measured time per day`
<dbl>,
## #   `categories of the percent total sitting time` <chr>,
## #   `categories of the total sitting time in hours` <chr>
```

```
tail(gupta_2015)
```

```
## # A tibble: 6 × 13
##   `serial number of the worker` `age in years` sex     `job seniority in
months`
##                           <dbl>          <dbl> <chr>
<dbl>
```

```
## 1                          17017             44 females
132
## 2                          17018             42 females
64
## 3                          17019             46 females
52
## 4                          17020             35 females
24
## 5                          17024             37 females
144
## 6                          17025             53 females
51
## # ℹ 9 more variables: `body mass index in kg/m2` <dbl>,
## #   `total sitting time in hours` <dbl>, `Influence at work in 0-100%`
<dbl>,
## #   `low back pain categories` <chr>, `smoking status categories` <chr>,
## #   `occupational lifting/carrying time categories` <chr>,
## #   `total sitting time as percentage of the mean measured time per day`
<dbl>,
## #   `categories of the percent total sitting time` <chr>,
## #   `categories of the total sitting time in hours` <chr>
```

```
glimpse(gupta_2015)
```

```
## Rows: 201
## Columns: 13
## $ `serial number of the worker`
<dbl> 1…
## $ `age in years`
<dbl> 4…
## $ sex
<chr> "…
## $ `job seniority in months`
<dbl> 1…
## $ `body mass index in kg/m2`
<dbl> 2…
## $ `total sitting time in hours`
<dbl> 8…
## $ `Influence at work in 0-100%`
<dbl> 3…
## $ `low back pain categories`
<chr> "…
## $ `smoking status categories`
<chr> "…
## $ `occupational lifting/carrying time categories`
<chr> "…
## $ `total sitting time as percentage of the mean measured time per day`
<dbl> 4…
## $ `categories of the percent total sitting time`
<chr> "…
```

```
## $ `categories of the total sitting time in hours`
<chr> "…
```

- If you are used to using the "Import Dataset" button in R-studio, you can select whether to use the `read_csv` function from `readr` or `read.csv`; `read_csv` reads the data set in as a `tibble`. If you use `read.csv`, it will be read the data in as a `data.frame`. You can use either. Tibbles make fewer assumptions about the variable types and you will need to transform any categorical variables as factors (using `factor`). In `data.frames`, assumptions are made about what is factor and what is numeric or character strings and can then require processing after the data are read in to deal with some ways variables are coded, especially when missing data are present and coded in some way other than empty cells.

- Another option in R-studio is to use the `readxl` package and its `read_excel` function to read data sets directly from Excel spreadsheets. This allows you to maintain multi-tab excel data sets, read in from the desired page, and even select the rows to read in. I posted the `gupta_2015.xlsx` file and it contains the same information as the .csv file in the first sheet. It also contains a second sheet we might explore at another time. The potential for keeping multiple sheets together (especially if the other sheets contain metadata or different related data sets) make the choice of Excel files attractive for data storage. The simpler comma-separated files (`.csv`) are still considered more stable ways of archiving data for future use.

- It is a good idea to explore the data set briefly. Explore the results of `View(bpdata)` (you will need to uncomment), `head(bpdata)`, and `tail(bpdata)`.

- The variable names in the original data set are not conducive to easy use in R, so we want to rename them. The following code will help you start to do that:

```
names(gupta_2015)
```

```
##  [1] "serial number of the worker"
##  [2] "age in years"
##  [3] "sex"
##  [4] "job seniority in months"
##  [5] "body mass index in kg/m2"
##  [6] "total sitting time in hours"
##  [7] "Influence at work in 0-100%"
##  [8] "low back pain categories"
##  [9] "smoking status categories"
## [10] "occupational lifting/carrying time categories"
## [11] "total sitting time as percentage of the mean measured time per day"
## [12] "categories of the percent total sitting time"
## [13] "categories of the total sitting time in hours"
```

```
library(tidyverse)
g2 <- gupta_2015 %>% dplyr::rename_all(list(~make.names(., unique=T)))
names(g2)
```

```
##  [1] "serial.number.of.the.worker"
##  [2] "age.in.years"
##  [3] "sex"
##  [4] "job.seniority.in.months"
##  [5] "body.mass.index.in.kg.m2"
##  [6] "total.sitting.time.in.hours"
##  [7] "Influence.at.work.in.0.100."
##  [8] "low.back.pain.categories"
##  [9] "smoking.status.categories"
## [10] "occupational.lifting.carrying.time.categories"
## [11] "total.sitting.time.as.percentage.of.the.mean.measured.time.per.day"
## [12] "categories.of.the.percent.total.sitting.time"
## [13] "categories.of.the.total.sitting.time.in.hours"
```

2) What impact did the code have on the names of the variables? Why is this needed?

The code above removed all spaces from the variable names and replaced them with a '.'. Additionally, *unique = T* ensures that all variable names will be different. This is important because (1), R does not handle spaces in names well and (2), to call upon these variables in the future, parentheses would be needed, which are cumbersome but also may not work depending on the operation.

In the past, I have used *gsub()* to remove spaces and other unwanted characters and symbols.

- We can change the name of a variable by over-writing that name slot with something better. Be careful with this and always double-check this sort of "forced" change as I demonstrate here with verifying the prior name and the changed name… For example, this code will rename the second variable (column) called `age.in.years` to `age`:

```
names(g2)[2]

## [1] "age.in.years"

names(g2)[2] <- "age"
names(g2)[2]

## [1] "age"
```

I also am going to rename a couple more variables we will use eventually using the `rename` function from the `tidyverse`:

```
g2 <- g2 %>% dplyr::rename(seniority = 'job.seniority.in.months',
                           BMI = 'body.mass.index.in.kg.m2',
                           influence = 'Influence.at.work.in.0.100.',
                           LBP = 'low.back.pain.categories',
                           smoking = 'smoking.status.categories')
```

3) Rename the `total.sitting.time.in.hours` to a more compact but still meaningful name and show code to verify the change.

```r
names(g2)
```

```
##  [1] "serial.number.of.the.worker"
##  [2] "age"
##  [3] "sex"
##  [4] "seniority"
##  [5] "BMI"
##  [6] "total.sitting.time.in.hours"
##  [7] "influence"
##  [8] "LBP"
##  [9] "smoking"
## [10] "occupational.lifting.carrying.time.categories"
## [11] "total.sitting.time.as.percentage.of.the.mean.measured.time.per.day"
## [12] "categories.of.the.percent.total.sitting.time"
## [13] "categories.of.the.total.sitting.time.in.hours"
```

```r
g3 <- g2 %>%
  rename(sitting_time = 'total.sitting.time.in.hours',
         olctc = 'occupational.lifting.carrying.time.categories') %>%
  print(n = 10)
```

```
## # A tibble: 201 × 13
##    serial.number.of.the.wor…¹   age sex   seniority   BMI sitting_time
influence
##                         <dbl> <dbl> <chr>     <dbl> <dbl>        <dbl>
<dbl>
##  1                      11001    48 male        120  23.0         8.37
37.5
##  2                      11002    32 male         41  29.6         8.28
81.2
##  3                      11003    51 male        102  24.8         8.51
75
##  4                      11004    65 male        480  26.8         7.16
62.5
##  5                      11005    58 male        524  28.2        10.9
93.8
##  6                      11007    45 male        240  23.5         8.94
81.2
##  7                      11008    33 male        182  25.2         7.72
81.2
##  8                      11009    61 male        508  25.7         9.23
43.8
##  9                      11011    37 male        207  40.8        10.4
18.8
## 10                      11012    27 male         23  19.0         8.71
6.25
## # ℹ 191 more rows
## # ℹ abbreviated name: ¹serial.number.of.the.worker
## # ℹ 6 more variables: LBP <chr>, smoking <chr>, olctc <chr>,
## #   total.sitting.time.as.percentage.of.the.mean.measured.time.per.day
```

```
<dbl>,
## #   categories.of.the.percent.total.sitting.time <chr>,
## #   categories.of.the.total.sitting.time.in.hours <chr>
```

I think it is good practice to provide a new name to the df when a change has been made so you can return to the previous if needed, rather than restarting your wrangling.

- We will focus on that total sitting time variable, measured in hours. The authors' used it as a predictor variable in their logistic regression model. The low back pain was the rating of the worst pain in the last month, converted from a 0 to 9 point scale to *low* or *high*. The total sitting time was based on an accelerometer and was actually the average of the sitting time across the work days available for each subject (four days).

- The back pain rating was based on information for the month prior to the sitting time measurements. It is more typical to use the "status" of a subject in an earlier time to predict/explain a response variable that is measured later in time, so I think turning the problem around is also reasonable here (they did this a bit but it was not their main focus). So we will consider the total sitting time as the response variable and low back pain (and other variables) as explanatory variables.

4) Generate and report the summary statistics for the total sitting time with all responses together and then broken down based on the low back pain groups. Report your R code and summary statistics. Compare these to the results in Table 1 in the paper. Note whether they match or don't match the results in the paper.

```
g3
```

```
## # A tibble: 201 × 13
##    serial.number.of.the.wor…¹   age sex   seniority   BMI sitting_time
influence
##                         <dbl> <dbl> <chr>     <dbl> <dbl>        <dbl>
<dbl>
##  1                      11001    48 male        120  23.0         8.37
37.5
##  2                      11002    32 male         41  29.6         8.28
81.2
##  3                      11003    51 male        102  24.8         8.51
75
##  4                      11004    65 male        480  26.8         7.16
62.5
##  5                      11005    58 male        524  28.2        10.9
93.8
##  6                      11007    45 male        240  23.5         8.94
81.2
##  7                      11008    33 male        182  25.2         7.72
81.2
##  8                      11009    61 male        508  25.7         9.23
43.8
```

```
##  9                      11011    37 male        207  40.8        10.4
18.8
## 10                      11012    27 male         23  19.0         8.71
6.25
## # ℹ 191 more rows
## # ℹ abbreviated name: ¹serial.number.of.the.worker
## # ℹ 6 more variables: LBP <chr>, smoking <chr>, olctc <chr>,
## #   total.sitting.time.as.percentage.of.the.mean.measured.time.per.day
<dbl>,
## #   categories.of.the.percent.total.sitting.time <chr>,
## #   categories.of.the.total.sitting.time.in.hours <chr>
```

```
# Summary stats for total sitting time
favstats(sitting_time ~ 1, data = g3)
```

```
##   1  min   Q1 median   Q3   max     mean       sd   n missing
## 1 1 2.48 5.86   7.22 8.71 13.52 7.282736 2.134157 201       0
```

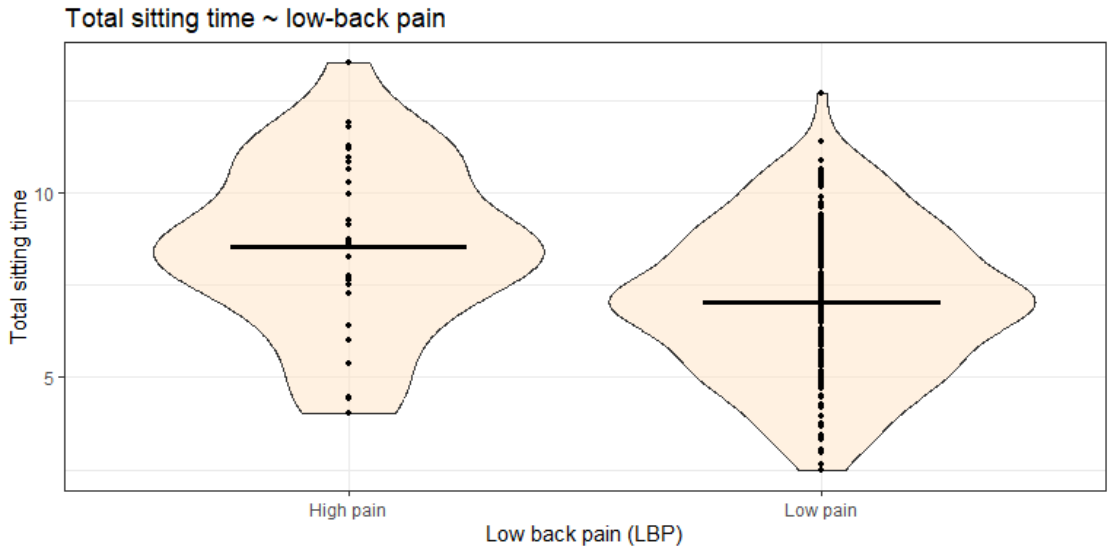
```
# Summary stats for total sitting time ~ LBP
favstats(sitting_time ~ LBP, data = g3)
```

```
##         LBP  min     Q1 median      Q3   max     mean       sd   n missing
## 1 high pain 4.02 7.5000   8.63 10.2700 13.52 8.550303 2.287796  33       0
## 2  low pain 2.48 5.5975   6.99  8.5125 12.70 7.033750 2.017884 168       0
```

I found the phrasing of this question to be confusing. At first, I attempted to recreate all of the summary stats in the table using this favstats function. Assuming my new answer is correct, then these results match the results in table 1.

5) Plot the total sitting time by low back pain groups. Report your R code and plots. No discussion.

```
ggplot(g3, aes(LBP, sitting_time))+
  geom_violin(fill = 'bisque', alpha = 0.5)+
  geom_point(size = 1)+
  stat_summary(fun = 'mean',
               geom = 'crossbar',
               width = .5,
               color = 'black')+
  labs(title = 'Total sitting time ~ low-back pain',
       x = 'Low back pain (LBP)',
       y = 'Total sitting time')+
  scale_x_discrete(labels = c("High pain", "Low pain"))
```



6) In their models reported in Table 3, they report a "crude model" and then one that is adjusted for age, gender, smoking, and bmi ("Step 1" in the second row of results). We will take inspiration from this to include these variables in a `lm` for total sitting time (response) along with the low back pain as a predictor. Fit this model and report a model `summary()` of it. No discussion - we will explore this more next week!

```
# step 1 model
m1 <- lm(sitting_time ~ LBP + age + sex + BMI + smoking, data = g3)
summary(m1)

##
## Call:
## lm(formula = sitting_time ~ LBP + age + sex + BMI + smoking,
##     data = g3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.3864 -1.4416 -0.1314  1.6021  5.2548
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       5.66056    1.14029   4.964 1.57e-06
## LBPlow pain                      -1.42988    0.40666  -3.516 0.000551
## age                               0.02415    0.01611   1.499 0.135531
## sexmale                           0.43643    0.31464   1.387 0.167091
## BMI                               0.04875    0.03094   1.576 0.116822
## smokingsmoker_daily or sometimes  0.42856    0.30635   1.399 0.163510
##
## Residual standard error: 2.065 on 184 degrees of freedom
##   (11 observations deleted due to missingness)
## Multiple R-squared:  0.09737,    Adjusted R-squared:  0.07284
## F-statistic:  3.97 on 5 and 184 DF,  p-value: 0.001922

confint(m1)
```

```
##                                          2.5 %      97.5 %
## (Intercept)                         3.410837643  7.91027744
## LBPlow pain                        -2.232187132 -0.62756652
## age                                -0.007632059  0.05593913
## sexmale                            -0.184328215  1.05718783
## BMI                                -0.012292056  0.10979553
## smokingsmoker_daily or sometimes -0.175838567  1.03296463
```

7) Replace the XXX's or pick in the brackets to complete the following sentence based on the previous output:

- There is **strong** evidence against the null hypothesis of no difference in **true mean** between high and low back pain groups ($t_{184} = -3.516$, p-value = 0.0005), after controlling for age, gender, smoking status, and BMI, so we would conclude that there [is] a difference and [keep] the LBP term [in] the model.

8) In Table 3 they note that their version of this model had n=190. The number of rows in `g2` is 201. What in the previous output suggests that we were also using 190 observations in this model?

```
dim(g3)
```

```
## [1] 201  13
```

The final df of a simple linear regression is n-2, where 2 represents the estimates of slope and the intercept. In this instance of a multiple linear regression, the final df is calculated using n-(k+1), where k represents the predictor variables. There are 5 predictors in this model and thus 190-5+1 = 184.

9) To help us with learning your names, go to OneNote from our MS Teams channel and post (copy and paste/insert) a selfie/picture that you are identifiable in (we want to be able to ID you to learn your name) in the "Homework or Project Feedback" tab in the "Homework 1 Selfie" page. No work here, just in OneNote for this question.

10) Document any resources you used outside of those provided in this class. This includes, but is not limited to, other students and generative AI. If the resource is not static, discuss how you used it and which questions you used it for.

I used this to obtain the syntax for putting the mean line into geom_violin. I used this to double check my understanding of multiple linear regression df calculations.

11) It's useful to record some information about the version of R you are using. When you `Knit` this documentation, it will report on the version of R that you are using. It should say 4.4.1 in your compiled word document:

- R version (short form): 4.4.1