Project 1: Exploratory Data Analysis

SDS348

```
library(tidyverse)
library(dplyr)
library(tidyverse)
library(dplyr)
library(tidyr)
library(cluster)
library(ggplot2)
library(lmtest)
library(sandwich)
library(vegan)
library(tidyverse)
library(dplyr)
data_1 <- read.csv("data 1.csv")
data_2 <- read.csv("data 2.csv")
select <- dplyr::select
glimpse(data_1)
```

```
## Rows: 202
## Columns: 7
## $ ï..ID <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18...
## $ rcc   <dbl> 3.96, 4.41, 4.14, 4.11, 4.45, 4.10, 4.31, 4.42, 4.30, 4.51, 4...
## $ wcc   <dbl> 7.5, 8.3, 5.0, 5.3, 6.8, 4.4, 5.3, 5.7, 8.9, 4.4, 5.3, 7.3, 7...
## $ hc    <dbl> 37.5, 38.2, 36.4, 37.3, 41.5, 37.4, 39.6, 39.9, 41.1, 41.6, 4...
## $ hg    <dbl> 12.3, 12.7, 11.6, 12.6, 14.0, 12.5, 12.8, 13.2, 13.5, 12.7, 1...
## $ ferr  <int> 60, 68, 21, 69, 29, 42, 73, 44, 41, 44, 38, 26, 30, 48, 30, 2...
## $ bmi   <dbl> 20.56, 20.67, 21.86, 21.88, 18.96, 21.04, 21.69, 20.62, 22.64...
```

```
glimpse(data_2)
```

```
## Rows: 202
## Columns: 8
## $ ï..ID  <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1...
## $ ssf    <dbl> 109.1, 102.8, 104.6, 126.4, 80.3, 75.2, 87.2, 97.9, 75.1, 65...
## $ pcBfat <dbl> 19.75, 21.30, 19.88, 23.66, 17.64, 15.58, 19.99, 22.43, 17.9...
## $ lbm    <dbl> 63.32, 58.55, 55.36, 57.18, 53.20, 53.77, 60.17, 48.33, 54.5...
## $ ht     <dbl> 195.9, 189.7, 177.8, 185.0, 184.6, 174.0, 186.2, 173.8, 171....
## $ wt     <dbl> 78.9, 74.4, 69.1, 74.9, 64.6, 63.7, 75.2, 62.3, 66.5, 62.9, ...
## $ sex    <chr> "f", "f", "f", "f", "f", "f", "f", "f", "f", "f", "f", "f", ...
## $ sport  <chr> "B_Ball", "B_Ball", "B_Ball", "B_Ball", "B_Ball", "B_Ball", ...
```

```
data_2 %>% dplyr::inner_join(data_1, by = "ï..ID") %>% na.omit()
```

```
##   ï..ID   ssf pcBfat   lbm    ht   wt sex  sport  rcc wcc   hc   hg ferr   bmi
## 1     1 109.1  19.75 63.32 195.9 78.9   f B_Ball 3.96 7.5 37.5 12.3   60 20.56
## 2     2 102.8  21.30 58.55 189.7 74.4   f B_Ball 4.41 8.3 38.2 12.7   68 20.67
## 3     3 104.6  19.88 55.36 177.8 69.1   f B_Ball 4.14 5.0 36.4 11.6   21 21.86
## 4     4 126.4  23.66 57.18 185.0 74.9   f B_Ball 4.11 5.3 37.3 12.6   69 21.88
## 5     5  80.3  17.64 53.20 184.6 64.6   f B_Ball 4.45 6.8 41.5 14.0   29 18.96
## 6     6  75.2  15.58 53.77 174.0 63.7   f B_Ball 4.10 4.4 37.4 12.5   42 21.04
## 7     7  87.2  19.99 60.17 186.2 75.2   f B_Ball 4.31 5.3 39.6 12.8   73 21.69
##  [ reached 'max' / getOption("max.print") -- omitted 195 rows ]
```

```
data_3 <- data_2 %>% inner_join(data_1, by = "ï..ID") %>% na.omit()
glimpse(data_3)
```

```
## Rows: 202
## Columns: 14
## $ ï..ID  <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 1...
## $ ssf    <dbl> 109.1, 102.8, 104.6, 126.4, 80.3, 75.2, 87.2, 97.9, 75.1, 65...
## $ pcBfat <dbl> 19.75, 21.30, 19.88, 23.66, 17.64, 15.58, 19.99, 22.43, 17.9...
## $ lbm    <dbl> 63.32, 58.55, 55.36, 57.18, 53.20, 53.77, 60.17, 48.33, 54.5...
## $ ht     <dbl> 195.9, 189.7, 177.8, 185.0, 184.6, 174.0, 186.2, 173.8, 171....
## $ wt     <dbl> 78.9, 74.4, 69.1, 74.9, 64.6, 63.7, 75.2, 62.3, 66.5, 62.9, ...
## $ sex    <chr> "f", "f", "f", "f", "f", "f", "f", "f", "f", "f", "f", "f", ...
## $ sport  <chr> "B_Ball", "B_Ball", "B_Ball", "B_Ball", "B_Ball", "B_Ball", ...
## $ rcc    <dbl> 3.96, 4.41, 4.14, 4.11, 4.45, 4.10, 4.31, 4.42, 4.30, 4.51, ...
## $ wcc    <dbl> 7.5, 8.3, 5.0, 5.3, 6.8, 4.4, 5.3, 5.7, 8.9, 4.4, 5.3, 7.3, ...
## $ hc     <dbl> 37.5, 38.2, 36.4, 37.3, 41.5, 37.4, 39.6, 39.9, 41.1, 41.6, ...
## $ hg     <dbl> 12.3, 12.7, 11.6, 12.6, 14.0, 12.5, 12.8, 13.2, 13.5, 12.7, ...
## $ ferr   <int> 60, 68, 21, 69, 29, 42, 73, 44, 41, 44, 38, 26, 30, 48, 30, ...
## $ bmi    <dbl> 20.56, 20.67, 21.86, 21.88, 18.96, 21.04, 21.69, 20.62, 22.6...
```

```
data_3 %>% select(ï..ID, sport, rcc) %>% pivot_wider(names_from = "sport",
    values_from = "rcc")
```

```
## # A tibble: 202 x 11
##    ï..ID B_Ball   Row Netball  Swim Field T_400m T_Sprnt Tennis   Gym W_Polo
##    <int>  <dbl> <dbl>   <dbl> <dbl> <dbl>  <dbl>   <dbl>  <dbl> <dbl>  <dbl>
##  1     1   3.96    NA      NA    NA    NA     NA      NA     NA    NA     NA
##  2     2   4.41    NA      NA    NA    NA     NA      NA     NA    NA     NA
##  3     3   4.14    NA      NA    NA    NA     NA      NA     NA    NA     NA
##  4     4   4.11    NA      NA    NA    NA     NA      NA     NA    NA     NA
##  5     5   4.45    NA      NA    NA    NA     NA      NA     NA    NA     NA
##  6     6   4.1     NA      NA    NA    NA     NA      NA     NA    NA     NA
##  7     7   4.31    NA      NA    NA    NA     NA      NA     NA    NA     NA
##  8     8   4.42    NA      NA    NA    NA     NA      NA     NA    NA     NA
##  9     9   4.3     NA      NA    NA    NA     NA      NA     NA    NA     NA
## 10    10   4.51    NA      NA    NA    NA     NA      NA     NA    NA     NA
## # ... with 192 more rows
```

```
data_4 <- data_3 %>% pivot_wider(names_from = "sport", values_from = "rcc")
data_4 <- data_4 %>% pivot_longer(13:22, names_to = "sport",
    values_to = "rcc", values_drop_na = T)
n_distinct(data_4)
```

```
## [1] 202
```

```
data_4 %>% summarize_if(is.numeric, c(max = max, min = min, mean = mean,
    median = median)) %>% pivot_longer(everything()) %>% separate(name,
    into = c("var", "stat")) %>% pivot_wider(names_from = "stat",
    values_from = "value")
```

```
## # A tibble: 12 x 6
##    var    ID        max       min       mean      median
##    <chr>  <list>    <list>    <list>    <list>    <list>
##  1 ï      <dbl [4]> <NULL>    <NULL>    <NULL>    <NULL>
##  2 ssf    <NULL>    <dbl [1]> <dbl [1]> <dbl [1]> <dbl [1]>
##  3 pcBfat <NULL>    <dbl [1]> <dbl [1]> <dbl [1]> <dbl [1]>
##  4 lbm    <NULL>    <dbl [1]> <dbl [1]> <dbl [1]> <dbl [1]>
##  5 ht     <NULL>    <dbl [1]> <dbl [1]> <dbl [1]> <dbl [1]>
##  6 wt     <NULL>    <dbl [1]> <dbl [1]> <dbl [1]> <dbl [1]>
##  7 wcc    <NULL>    <dbl [1]> <dbl [1]> <dbl [1]> <dbl [1]>
##  8 hc     <NULL>    <dbl [1]> <dbl [1]> <dbl [1]> <dbl [1]>
##  9 hg     <NULL>    <dbl [1]> <dbl [1]> <dbl [1]> <dbl [1]>
## 10 ferr   <NULL>    <dbl [1]> <dbl [1]> <dbl [1]> <dbl [1]>
## 11 bmi    <NULL>    <dbl [1]> <dbl [1]> <dbl [1]> <dbl [1]>
## 12 rcc    <NULL>    <dbl [1]> <dbl [1]> <dbl [1]> <dbl [1]>
```

```
data_4 %>% summarize_all(mean)
```

```
## # A tibble: 1 x 14
##   ï..ID   ssf pcBfat   lbm    ht    wt   sex   wcc    hc    hg  ferr   bmi sport
##   <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  102.  69.0   13.5  64.9  180.  75.0    NA  7.11  43.1  14.6  76.9  23.0    NA
## # ... with 1 more variable: rcc <dbl>
```

Data Wrangling and Data Exploration

Instructions

A knitted R Markdown document (as a PDF) and the raw R Markdown file (as .Rmd) should both be submitted to Canvas by 11:59pm on 3/8/2020. These two documents will be graded jointly, so they must be consistent (i.e., don't change the R Markdown file without also updating the knitted document).

The text of the document should provide a narrative structure around your code/output. All results presented must have corresponding code. Any answers/results/plots etc. given without the corresponding R code that generated the result will not be considered. Furthermore, all code contained in your final project document must work correctly (knit early, knit often)! Please do not include any extraneous code or code which produces error messages. (Code that produces warnings is acceptable, as long as you understand what the warnings mean!)

Find data:

Find two (!) datasets with one variable in common (e.g., dates, times, states, counties, countries, sports players), both with at least 50 observations (i.e., rows) in each. Please think carefully it makes sense to combine your datasets! When combined, the resulting/final dataset must have **at least 4 different variables (at least 3 numeric) in addition to the common variable** (i.e., five variables total).

You can have as many variables as you would like! If you found two datasets that you like but they don't have enough variables, find a third dataset with the same common variable and join all three.

Guidelines

1. If the datasets are not tidy, you will need to reshape them so that every observation has its own row and every variable its own column. If the datasets are both already tidy, you will make them untidy with `pivot_wider()/spread()` and then tidy them again with `pivot_longer/gather()` to demonstrate your use of the functions. It's fine to wait until you have your descriptives to use these functions (e.g., you might want to pivot_wider() to rearrange the data to make your descriptive statistics easier to look at); it's fine long as you use them at least once!

    - Depending on your datasets, it might be a good idea to do this before joining. For example, if you have a dataset you like with multiple measurements per year, but you want to join by year, you could average over your numeric variables to get means/year, do counts for your categoricals to get a counts/year, etc.
    - If your data sets are already tidy, demonstrate the use of `pivot_longer()/gather()` and `pivot_wider()/spread()` on all or part of your data at some point in this document (e.g., after you have generated summary statistics in part 3, make a table of them wide instead of long).

2. Join your 2+ separate data sources into a single dataset

    - You will document the type of join that you do (left/right/inner/full), including a discussion of how many cases in each dataset were dropped (if any) and why you chose this particular join

3. Create summary statistics

    - Use *all six* core `dplyr` functions (`filter, select, arrange, group_by, mutate, summarize`) to manipulate and explore your dataset. For mutate, create a new variable that is a function of at least one other variable, preferably using a dplyr vector function (see dplyr cheatsheet). It's totally fine to use the `_if`, `_at`, `_all` versions of mutate/summarize instead (indeed, it is encouraged if you have lots of variables)
    - Create summary statistics (`mean, sd, var, n, quantile, min, max, n_distinct, cor`, etc) for each of your numeric variables both overall and after grouping by one of your categorical variables (either together or one-at-a-time; if you have two categorical variables, try to include at least one statistic based on a grouping of two categorical variables simultaneously). If you do not have any categorical variables, create one using mutate (e.g., with `case_when` or `ifelse`) to satisfy the `group_by` requirements above. Ideally, you will find a way to show these summary statistics in an easy-to-read table (e.g., by reshaping). (You might explore the kable package for making pretty tables!) If you have lots of numeric variables, or your categorical variables have too many categories, just pick a few (either numeric variables or categories of a categorical variable) and summarize based on those. It would be a good idea to show a correlation matrix for your numeric variables!

4. Make visualizations (three plots)

    - Make a correlation heatmap of your numeric variables
    - Create at least two additional plots of your choice with ggplot that highlight some of the more interesting findings that your descriptive statistics have turned up.
    - Each plot (besides the heatmap) should have at least three variables mapped to separate aesthetics
    - At least one plot should include `stat="summary"`
    - Each plot should include a supporting paragraph describing the relationships that are being visualized and any trends that are apparent

  - It is fine to include more, but limit yourself to 4. Plots should avoid being redundant! Four bad plots will get a lower grade than two good plots, all else being equal.
- Make them pretty! Use correct labels, etc.

5. Perform k-means/PAM clustering or PCA on (at least) your numeric variables.

   - Include all steps as we discuss in class, including a visualization.
   - If you don't have at least 3 numeric variables, or you want to cluster based on categorical variables too, convert them to factors in R, generate Gower's dissimilarity matrix on the data, and do PAM clustering on the dissimilarities.
   - Show how you chose the final number of clusters/principal components
   - Interpret the final clusters/principal components
   - For every step, document what your code does (in words) and what you see in the data!

Rubric

Prerequisite: Finding appropriate data from at least two sources per the instructions above: Failure to do this will result in a 0! You will submit a .Rmd file and a knitted document (pdf).

0. Introduction (4 pts)

- Write a narrative introductory paragraph or two describing the datasets you have chosen, the variables they contain, how they were acquired, and why they are interesting to you. Expand on potential associations you may expect, if any.

1. Tidying: Rearranging Wide/Long (8 pts)

- Tidy the datasets (using the `tidyr` functions `pivot_longer`/`gather` and/or `pivot_wider`/`spread`)
- If you data sets are already tidy, be sure to use those functions somewhere else in your project
- Document the process (describe in words what was done per the instructions)

2. Joining/Merging (8 pts)

- Join your datasets into one using a `dplyr` join function
- If you have multiple observations on the joining variable in either dataset, fix this by collapsing via summarize
- Discuss the process in words, including why you chose the join you did
- Discuss which cases were dropped, if any, and potential problems with this

3. Wrangling (40 pts)

- Use all six core `dplyr` functions in the service of generating summary statistics (18 pts)
  - Use mutate to generate a variable that is a function of at least one other variable
- Compute at least 10 different summary statistics using summarize and summarize with group_by (18 pts)
  - At least 2 of these should group by a categorical variable. Create one by dichotomizing a numeric if necessary
  - If applicable, at least 1 of these 5 should group by two categorical variables
  - Strongly encouraged to create a correlation matrix with `cor()` on your numeric variables
- Summarize/discuss all results in no more than two paragraphs (4 pts)

4. Visualizing (30 pts)

- Create a correlation heatmap of your numeric variables
- Create two effective, polished plots with ggplot
  - Each plot should map 3+ variables to aesthetics
  - Each plot should have a title and clean labeling for all mappings
  - Change at least one default theme element and color for at least one mapping per plot
  - For at least one plot, add more tick marks (x, y, or both) than are given by default
  - For at least one plot, use the stat="summary" function
  - Supporting paragraph or two (for each plot) describing the relationships/trends that are apparent

5. Dimensionality Reduction (20 pts)

- Either k-means/PAM clustering or PCA (inclusive "or") should be performed on at least three numeric variables in your dataset
  - All relevant steps discussed in class
  - A visualization of the clusters or the first few principal components (using ggplot2)
  - Supporting paragraph or two describing results found

6. Neatness!

- Your project should not knit to more than 30 or so pages. You will lose points if you print out your entire dataset, etc. If you start your project in a fresh .Rmd file, you are advised to paste the set-up code from this document (lines 14-17) at the top of it: this will automatically truncate if you accidentally print out a huge dataset, etc. Imagine this is a polished report you are giving to your PI or boss to summarize your work researching a topic.

Where do I find data?

OK, brace yourself!

You can choose ANY datasets you want that meet the above criteria for variables and observations. I'm just sitting here but off the top of my head, if you are into amusement parks, you could look at amusement-park variables, including ticket sales per day etc.; then you could join this by date in weather data. If you are interested in Game of Thrones, you could look at how the frequency of mentions of character names (plus other character variables) and the frequency of baby names in the USA... You could even take your old Biostats data and merge in new data (e.g., based on a Google forms timestamp).

You could engage in some "me-search": You can request your Spotify data or download Netflix viewing activity, Amazon purchase history, etc. You can use your Google Fit/Fitbit/Apple watch data, etc. These can be combined (e.g., with each other, with other data sources).

You can make it as serious as you want, or not, but keep in mind that you will be incorporating this project into a portfolio webpage for your final in this course, so choose something that really reflects who you are, or something that you feel will advance you in the direction you hope to move career-wise, or something that you think is really neat. On the flip side, regardless of what you pick, you will be performing all the same tasks, so it doesn't end up being that big of a deal.

If you are totally clueless and have no direction at all, log into the server and type

```
data(package = .packages(all.available = TRUE))
```

This will print out a list of **ALL datasets in ALL packages** installed on the server (a ton)! Scroll until your eyes bleed! Actually, do not scroll that much... To start with something more manageable, just run the command on your own computer, or just run `data()` to bring up the datasets in your current environment. To read more about a dataset, do `?packagename::datasetname`.

If it is easier for you, and in case you don't have many packages installed, a list of R datasets from a few common packages (also downloadable in CSV format) is given at the following website: https://vincentarelbundock.github.io/Rdatasets/datasets.html (including types/numbers of variables in each)

- A good package to download for fun/relevant data is `fivethiryeight`. Just run `install.packages("fivethirtyeight` `load the packages with`library(fivethirtyeight)`, run`data()', and then scroll down to view the datasets. Here is an online list of all 127 datasets (with links to the 538 articles). Lots of sports, politics, current events, etc: https://cran.r-project.org/web/packages/fivethirtyeight/vignettes/fivethirtyeight.html

- If you have already started to specialize (e.g., ecology, epidemiology) you might look at discipline-specific R packages (vegan, epi, respectively). We will be using some tools from these packages later in the course, but they come with lots of data too, which you can explore according to the directions above
- However, you *emphatically DO NOT* have to use datasets available via R packages! In fact, I would much prefer it if you found the data from completely separate sources and brought them together (a much more realistic experience in the real world)! You can even reuse data from your SDS328M project, provided it shares a variable in common with other data which allows you to merge the two together (e.g., if you still had the timestamp, you could look up the weather that day: https://www.wunderground.com/history/). If you work in a research lab or have access to old data, you could potentially merge it with new data from your lab!
- Here is a curated list of interesting datasets (read-only spreadsheet format): https://docs.google.com/spreadsheets/d/1wZhPLMCHKJvwOkP4juclhjFgqIY8fQFMemwKL2c64vk/edit
- Here is another great compilation of datasets: https://github.com/rfordatascience/tidytuesday
- Here is the UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/index.php
  - See also https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research#Biological_data
- Here is another good general place to look: https://www.kaggle.com/datasets
- To help narrow your search down or to see interesting variable ideas, check out https://www.tylervigen.com/spurious-correlations. This is the spurious correlations website, and it is fun, but if you look at the bottom of each plot you will see sources for the data. This is a good place to find very general data (or at least get a sense of where you can scrape data together from)!
- If you are interested in medical data, check out www.countyhealthrankings.org
- If you are interested in scraping UT data, the university makes *loads* of data public (e.g., beyond just professor CVs and syllabi). Check out all the data that is available in the statistical handbooks: https://reports.utexas.edu/statistical-handbook

Broader data sources: Data.gov 186,000+ datasets!

Social Explorer is a nice interface to Census and American Community Survey data (more user-friendly than the government sites). May need to sign up for a free trial.

U.S. Bureau of Labor Statistics

U.S. Census Bureau

Gapminder, data about the world.

...