Lab 2: Subsetting and Basic Data Summaries
STAT 630, Fall 2019

Remark: This lab borrows from the "Introduction to Data" lab available here: `https://www.openintro.org/stat/labs`.

1 BRFSS Data Set

The Behavioral Risk Factor Surveillance System (BRFSS) is an annual telephone survey of over 400,000 people in the United States. The survey is conducted by the Centers for Disease Control and Prevention (CDC), a government agency focused on public health issues. As its name implies, the BRFSS is designed to identify risk factors in the adult population and report emerging health trends. For example, respondents are asked about their diet and weekly physical activity, their HIV/AIDS status, possible tobacco use, and level of healthcare coverage. The BRFSS web site `http://www.cdc.gov/brfss` contains a complete description of the survey, including the research questions that motivate the study and many interesting results derived from the data.

We will focus on a random sample of 20,000 people from the BRFSS survey conducted in the year 2000. While there are over 200 variables in this data set, we will work with a small subset.

We begin by loading the data set of 20,000 observations into the R workspace. After launching RStudio, enter the following command.

```
data_url <- "https://github.com/ericwfox/stat630data/raw/master/cdc.csv"
cdc <- read.csv(data_url)
```

The data are stored in a repository for this course. The function `read.csv()` reads the data set into R from the specified web address. The data set is in a CSV format (comma delimited values). You can type `?read.csv` into the console to learn more about this function. Note that `read.csv()` can also be used to read in a data set from a file located locally on your desktop.

To view the variable names and dimension of the `cdc` data frame type the following commands.

```
names(cdc)

## [1] "genhlth"  "exerany"  "hlthplan" "smoke100" "height"   "weight"
## [7] "wtdesire" "age"      "gender"

dim(cdc)

## [1] 20000     9
```

We can see clearly now the the data frame contains 20,000 entries (rows) on 9 variables. Each of the variables corresponds to a question that was asked in the survey. Descriptions of the variables are provided below:

- `genhlth`: a categorical vector indicating general health, with categories excellent, very good, good, fair, and poor
- `exerany`: a categorical vector, 1 if the respondent exercised in the past month and 0 otherwise
- `hlthplan`: a categorical vector, 1 if the respondent has some form of health coverage and 0 otherwise
- `smoke100`: a categorical vector, 1 if the respondent has smoked at least 100 cigarettes in their entire life and 0 otherwise
- `height`: a numerical vector, respondent's height in inches
- `weight`: a numerical vector, respondent's weight in pounds
- `wtdesire`: a numerical vector, respondent's desired weight in pounds
- `age`: a numerical vector, respondent's age in years
- `gender`: a categorical vector, respondent's gender

We can have a look at the first several rows of the data with the command

```
head(cdc)
```

```
##     genhlth exerany hlthplan smoke100 height weight wtdesire age gender
## 1      good       0        1        0     70    175      175  77      m
## 2      good       0        1        1     64    125      115  33      f
## 3      good       1        1        1     60    105      105  49      f
## 4      good       1        1        0     66    132      124  42      f
## 5 very good       0        1        0     61    150      130  55      f
## 6 very good       1        1        0     64    114      114  55      f
```

You could also look at all of the data frame at once by typing its name into the console, but that might be unwise here. We know `cdc` has 20,000 rows, so viewing the entire data set would mean flooding your screen. It's better to take small peeks at the data with `head()`, `tail()` or the indexing techniques covered during the last lab.

2 Summaries and Tables

The BRFSS questionnaire is a massive trove of information. A good first step in any analysis is to distill all of that information into a few summary statistics and graphics. As a simple example, the function `summary()` returns a numerical summary: minimum, first quartile, median, mean, third quartile, and maximum. For `weight` this is

```
summary(cdc$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    68.0   140.0   165.0   169.7   190.0   500.0
```

As discussed in the previous lab, R also has built-in functions to compute summary statistics one at a time. For instance, to calculate the mean, median, and standard deviation of `weight` type

```
mean(cdc$weight)

## [1] 169.683

median(cdc$weight)

## [1] 165

sd(cdc$weight)

## [1] 40.08097
```

While it makes sense to describe a numerical variable like `weight` in terms of these statistics, what about categorical data? We could instead consider the frequency or relative frequency distribution. The function `table()` does this for you by counting the number of times each kind of response was given. For example, to see the number of people who have smoked 100 cigarettes in their lifetime, type

```
table(cdc$smoke100)

##
##     0     1
## 10559  9441
```

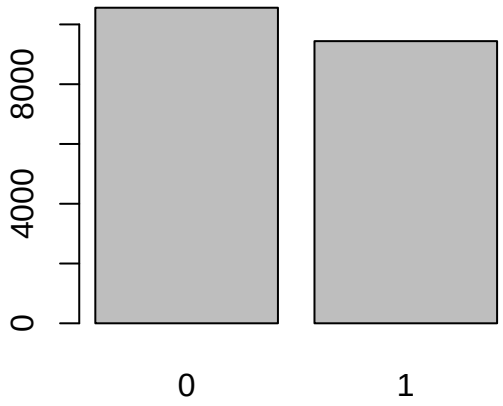
or instead look at the relative frequency distribution by typing

```
table(cdc$smoke100)/20000

##
##       0       1
## 0.52795 0.47205
```

Next, we make a bar plot of the entries in the table by putting the table inside the `barplot()` command.

```
barplot(table(cdc$smoke100))
```



Notice what we've done here! We've computed the table of `cdc$smoke100` and then immediately applied the graphical function, `barplot()`. This is an important idea: R commands can be nested. You could also break this into two steps by typing the following:

```
smoke <- table(cdc$smoke100)
barplot(smoke)
```

The `table()` command can be used to tabulate any number of variables that you provide. For example, to examine which participants have smoked across each gender, we could use the following.

```
table(cdc$gender,cdc$smoke100)

##
##        0    1
##   f 6012 4419
##   m 4547 5022
```

Here, we see column labels of 0 and 1. Recall that 1 indicates a respondent has smoked at least 100 cigarettes. The rows refer to gender. To add the margins (column and row totals) use `addmargins()`.

```
addmargins(table(cdc$gender,cdc$smoke100))
```

```
##
##           0     1   Sum
##   f    6012  4419 10431
##   m    4547  5022  9569
##   Sum 10559  9441 20000
```

3 Subsetting Data Frames

The previous lab went over how to extract rows, columns, and specific elements of a data frame using indexing (i.e., brackets `[]`) or by using `$` to extract columns (variables) by their names. However, it is also useful to extract rows of a data frame that have specific characteristics. For instance, suppose we want the extract the rows of the `cdc` data frame that correspond to a certain gender (male or female), or extract the rows corresponding to individuals that weigh over 140lbs. To do this we can use logical expressions and subsetting techniques.

To illustrate logical operations in R, lets work with a smaller portion of the `cdc` data frame that consists of the first 10 rows.

```
cdc10 <- cdc[1:10,]
cdc10
```

```
##      genhlth exerany hlthplan smoke100 height weight wtdesire age gender
## 1       good       0        1        0     70    175      175  77      m
## 2       good       0        1        1     64    125      115  33      f
## 3       good       1        1        1     60    105      105  49      f
## 4       good       1        1        0     66    132      124  42      f
## 5  very good       0        1        0     61    150      130  55      f
## 6  very good       1        1        0     64    114      114  55      f
## 7  very good       1        1        0     71    194      185  31      m
## 8  very good       0        1        0     67    170      160  45      m
## 9       good       0        1        1     65    150      130  27      f
## 10      good       1        1        0     70    180      170  44      m
```

The following command gives logical values (`TRUE`, `FALSE`) for whether each individual is male.

```
cdc10$gender
```

```
##  [1] m f f f f f m m f m
## Levels: f m
```

```
cdc10$gender == 'm'
```

```
##  [1]  TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE  TRUE
```

To extract the rows of the data frame `cdc10` corresponding to the males, use the `subset()` function.

```
subset(cdc10, gender == 'm')
```

```
##      genhlth exerany hlthplan smoke100 height weight wtdesire age gender
## 1       good       0        1        0     70    175      175  77      m
## 7  very good       1        1        0     71    194      185  31      m
## 8  very good       0        1        0     67    170      160  45      m
## 10      good       1        1        0     70    180      170  44      m
```

As an alternative, this command also does the same thing:

```
cdc10[cdc10$gender == 'm', ]
```

Similarly, we can extract the rows of the data frame `cdc10` corresponding to the individuals that weigh over 140lbs.

```
cdc10$weight
```

```
##  [1] 175 125 105 132 150 114 194 170 150 180
```

```
cdc10$weight > 140
```

```
##  [1]  TRUE FALSE FALSE FALSE  TRUE FALSE  TRUE  TRUE  TRUE  TRUE
```

```
subset(cdc10, weight > 140)
```

```
##      genhlth exerany hlthplan smoke100 height weight wtdesire age gender
## 1       good       0        1        0     70    175      175  77      m
## 5  very good       0        1        0     61    150      130  55      f
## 7  very good       1        1        0     71    194      185  31      m
## 8  very good       0        1        0     67    170      160  45      m
## 9       good       0        1        1     65    150      130  27      f
## 10      good       1        1        0     70    180      170  44      m
```

To extract the rows of `cdc10` corresponding to individuals that weigh over 140 pounds *and* are males use the following command.

```
subset(cdc10, gender == 'm' & weight > 140)
```

```
##      genhlth exerany hlthplan smoke100 height weight wtdesire age gender
## 1       good       0        1        0     70    175      175  77      m
## 7  very good       1        1        0     71    194      185  31      m
## 8  very good       0        1        0     67    170      160  45      m
## 10      good       1        1        0     70    180      170  44      m
```

As an exercise, we can also extract the individuals that weigh over 140 pounds *or* are males using the following command.

```
subset(cdc10, gender == 'm' | weight > 140)
```

```
##      genhlth exerany hlthplan smoke100 height weight wtdesire age gender
## 1       good       0        1        0     70    175      175  77      m
## 5  very good       0        1        0     61    150      130  55      f
## 7  very good       1        1        0     71    194      185  31      m
## 8  very good       0        1        0     67    170      160  45      m
## 9       good       0        1        1     65    150      130  27      f
## 10      good       1        1        0     70    180      170  44      m
```

The following table summarizes the different logical operators in R:

Operator	Description
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
==	exactly equal to
!=	not equal to
x | y	x OR y
x & y	x AND y

Note that = is used for assignment and is not the same as the logical operator ==.

Using these new subsetting tools we can explore some interesting aspects of the entire `cdc` data frame. For example, what is the average weight and desired weight for males and females? To answer this question create separate data frames for the males and females. Then use the `summary()` function on each subsetted data frame.

```
cdc_m <- subset(cdc, gender=='m')
cdc_f <- subset(cdc, gender=='f')

summary(cdc_m$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    78.0   165.0   185.0   189.3   210.0   500.0
```

```
summary(cdc_m$wtdesire)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    77.0   160.0   175.0   178.6   190.0   680.0
```

```
summary(cdc_f$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    68.0   128.0   145.0   151.7   170.0   495.0
```

```
summary(cdc_f$wtdesire)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    68.0   120.0   130.0   133.5   145.0   350.0
```

The mean and median desired weight is lower than the actual weight for both genders. The maximum desired weight for males is unusual since an individual has a desired weight of 680lbs! This is probably an outlier that we might want to remove since it can affect other statistics such as the mean (more on this next class!).

Similarly, we can now also answer a question such as what is the average weight and desired weight for males over 50 years old in the data set?

```
cdc_m_over50 <- subset(cdc, gender == "m" & age > 50)
summary(cdc_m_over50$weight)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    78.0   165.0   185.0   188.6   208.0   400.0

summary(cdc_m_over50$wtdesire)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    77.0   160.0   175.0   175.6   190.0   601.0
```

Lab 2 Assignment

Due: Thursday, September 5

Directions: Please submit the completed assignment to Blackboard. I suggest using R Markdown and knitting to PDF or HTML. If you are using Word, please convert your report into a PDF before submitting. Include all R code in your answers to each question.

Exercise 1. Use `plot()` to make a scatter plot with `weight` on the x-axis, and `wtdesire` on the y-axis. Label the x-axis "Weight" and the y-axis "Desired Weight". Superimpose a 1-1 line on your scatter plot by entering the command `abline(0,1)` after creating the plot.

Exercise 2. Based on the scatter plot created in the previous exercise, you should notice two outliers. The outliers correspond to individuals that have desired weights that far exceed their actual weights. Use the `subset()` function to identify the outliers by extracting the two rows of the `cdc` data frame corresponding to the individuals that have desired weights above 500lbs. What are the actual weights for these two individuals?
Next, create a new data frame called `cdc2` that has the outliers removed (Hint: use `subset()` again to do this). Then make another scatter plot with `weight` on the x-axis, and `wtdesire` on the y-axis, but this time with the outliers removed.

Exercise 3. Create a new data frame that contains the subset of respondents that are male *and* have exercised in the last month. Use the `summary()` function to compute summary statistics for the weight and desired weight of this subset of respondents.

Exercise 4. Use the `table()` function to make a contingency table between the general health, `genhealth`, and exercise, `exerany`, variables. Use `addmargins()` to include the row and column totals for this table. What proportion of respondents that reported to be in excellent health exercised in the past month? What proportion of respondents that reported to be in poor health exercised in the past month?