# Data Acquisition

*Erica Wong, Cindy Kang, and Abby Vogel*

*4/25/2017*

Loading the Data:

```
data <- read.csv('../full_data_v4.csv', header = T)
data$X <- NULL
data$X.1 <- NULL
```

Number of Rows:

```
(nrow(data))
```

```
## [1] 713
```

First 5 Rows of Data:

```
head(data, 5)
```

```
##      id   int_wt  exam_wt psu stratum gender age        bmi opinion_wt
## 1 73579 63248.99 70708.03   1     110      F  12     normal     normal
## 2 73584 72700.38 72182.24   1     105      M  13 overweight overweight
## 3 73587 16220.74 15523.09   2     115      M  14      obese overweight
## 4 73599 25841.53 27288.18   2     107      F  13     normal     normal
## 5 73601 17430.86 18842.87   1     115      M  12     normal     normal
##   action_wt pe_yn freq_pe       enjoy_pe
## 1  maintain   yes       5          agree
## 2      lose   yes       3          agree
## 3      lose   yes       5 strongly agree
## 4  maintain   yes       3 strongly agree
## 5      lose   yes       5 strongly agree
```

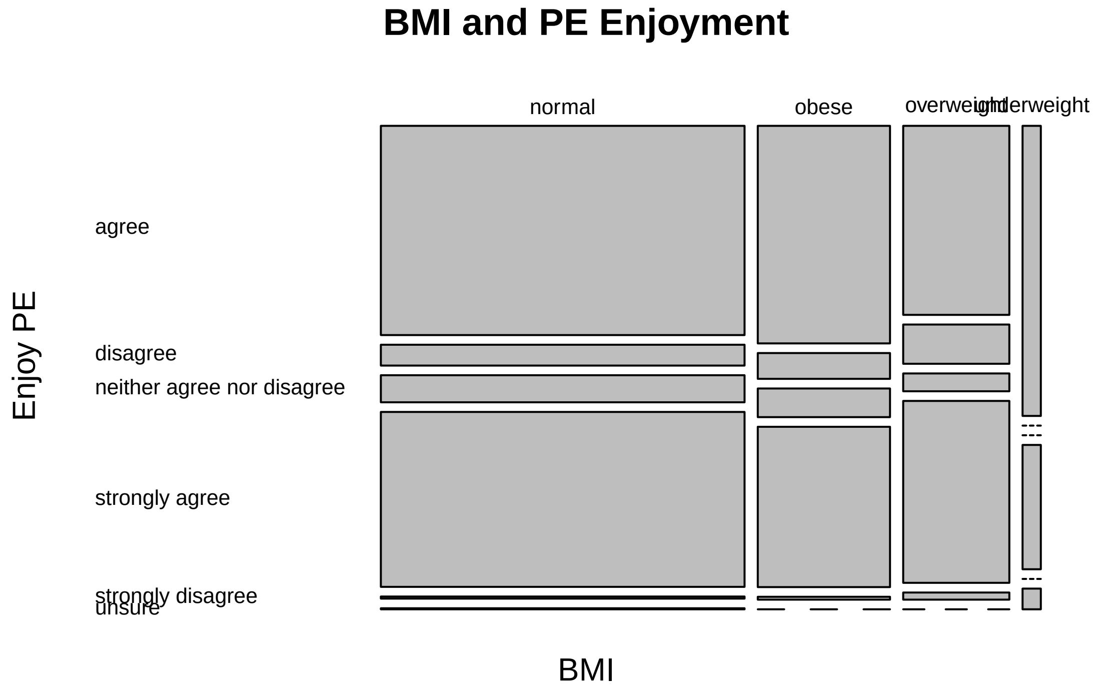
Summary:

```
summary(data)
```

```
##        id            int_wt          exam_wt            psu
##  Min.   :73579   Min.   :  5874   Min.   :  6367   Min.   :1.00
##  1st Qu.:75939   1st Qu.: 11738   1st Qu.: 12099   1st Qu.:1.00
##  Median :78582   Median : 15750   Median : 15870   Median :1.00
##  Mean   :78628   Mean   : 23421   Mean   : 24068   Mean   :1.44
##  3rd Qu.:81191   3rd Qu.: 21822   3rd Qu.: 22860   3rd Qu.:2.00
##  Max.   :83704   Max.   :102078   Max.   :104556   Max.   :2.00
##     stratum    gender       age                 bmi            opinion_wt
##  Min.   :104   F:348   Min.   :12.00   normal     :418   normal     :509
##  1st Qu.:107   M:365   1st Qu.:12.00   obese      :152   overweight :147
##  Median :111           Median :14.00   overweight :122   underweight: 57
##  Mean   :111           Mean   :13.49   underweight: 21
##  3rd Qu.:115           3rd Qu.:14.00
##  Max.   :118           Max.   :15.00
##     action_wt   pe_yn        freq_pe                            enjoy_pe
##  gain    : 87   no :128   Min.   :0.000   agree                      :344
##  lose    :286   yes:585   1st Qu.:2.000   disagree                   : 40
##  maintain:198             Median :3.000   neither agree nor disagree: 41
##  nothing :141              Mean   :3.192   strongly agree              :281
##  unsure  :  1              3rd Qu.:5.000   strongly disagree           :  5
##                             Max.   :5.000   unsure                      :  2
```

Plot:

```
library(ggplot2)
er <- table(data$bmi, data$enjoy_pe)
mosaicplot(er, las=1, xlab="BMI", ylab="Enjoy PE", main="BMI and PE Enjoyment")
```



What We Did:

In order to get the data to this point, we first looked at the code book of each of the data sets that we were interested in using. Then using the plyr and dplyr packages, we were able to join all of the different SAS files that we were interested in using based upon the given ID. Because the column names are coded by something that is impossible to understand without having the code book open, we started off by renaming our columns to something that can be understood by a person who is reading our code. Additionally, since all the variables are coded by numbers, we replaced the numbers with informative factors that allow us to know what the data is telling us about the subjects without needing to look up what each number for each column means.

Finally, we looked at our data and realized that there were some missing values, We noticed that in our data, some of the rows had weights of 0 and contained many NAs in the row. We removed these from our data because we believed this to be unit non-response, so there was nothing that we could do with those people and it would not make sense to try to fill in their response. Additionally, one of columns that was giving us a lot of NAs was freq_PE, which is the frequency of PE class. From looking at the rows that contained NA for that column, we realized that the NAs all occurred when a student didn't have PE at school. So, we recorded the missing value to 0. Finally, we had rows that had a weight, and contained information about one's age and gender. So, we decided to use imputation to fill in these missing values. What we did was that if a row had NA, we would pick another row of equal age and gender, and would use all of the values from that row. The reason we decided to do this is because we believe that many of our variables are correlated so it would not make sense to impute each column individually, thus we imputed the entire row. In the end, we went with using random hot deck imputation to take care of our NA values.