Homework #1

Justin Robinette

August 28, 2018

No collaborators for any problem

Problem #1: Calculate the median profit for the companies in the US and median profit for the companies in the UK, France, and Germany. This question will require you to make some assumptions. List your assumptions and how you interpreted the question.

Results: I found that the median profit for the US companies was higher, by approximately .03 billion, or $30 million USD, than the median profit for the UK, French, and German companies.

I assumed the question was asking for the collective mediian of the UK, France, and Germany companies due to it's wording.

I used the mice() function to impute the missing 'profit' values. Since the question asked specifically for the medians from the only column with missing data values, I assumed imputation was a part of the problem. I used a 'seed' value to ensure reproducible results.

Another assumption made in this problem was that the missing data values were missing at random from the data set. This is an assumption of the mice() function, and, through my use of the function, an assumption I made as well.

My final output answers the question by providing the 2 median profit values requested in the question.

```
## [1] "The median for US companies = 0.24 billion in USD."
```

```
## [1] "The median for UK, French, and German companies = 0.21 billion in USD."
```

Problem #2: Find all German companies with negative profit.

Results: The 13 German companies with negative profits are listed below. I printed just the names of the companies since the textbook asks to list the companies, making no mention of the other variables.

As part of my analysis, I used the subset() function to determine the number of German companies on the Forbes2000. The total number of German companies, in the data set, is 65. This tells me that 13 of the 65 German companies, or 20%, showed negative profits.

To see if German companies were more likely to have negative profits, I ran a subset of all companies with negative profits. I found that 283 of the 2000 companies on the list, or 14.15%, had negative profits. This code is included, but commented out, as it was not a request of the exercise.

```
##  [1] "Allianz Worldwide"       "Deutsche Telekom"
##  [3] "E.ON"                    "HVB-HypoVereinsbank"
##  [5] "Commerzbank"             "Infineon Technologies"
##  [7] "BHW Holding"             "Bankgesellschaft Berlin"
##  [9] "W&W-Wustenrot"           "mg technologies"
## [11] "Nurnberger Beteiligungs" "SPAR Handels"
## [13] "Mobilcom"
```

Problem #3: To which business category do most of the Bermuda island companies belong?

Results: The majority of the Bermuda companies belong to the "Insurance" 'category'. I found this by creating a subset of the Forbes2000 companies that had a 'country' value of "Bermuda". Next, I used the names() and which.max() functions together to show the most frequently used 'category'.

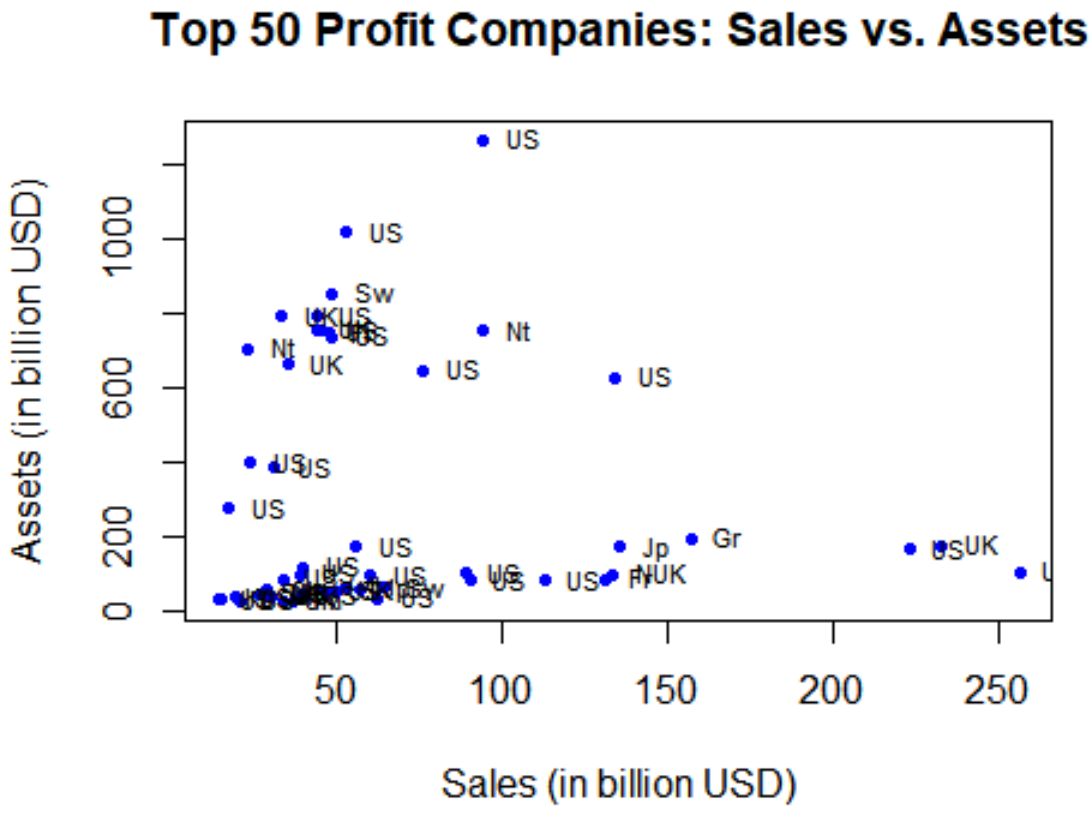
This seemed like an odd category to be prevalent in Bermuda, so I did a little independent research. I found a great New York Times article that discusses a federal tax loophole that American insurance companies can use simply by moving their headquarters to Bermuda or being acquired by a Bermuda insurer. This loophole allows them to no longer pay income taxes. Here is a link to the article if you'd like to examine this loophole in greater detail: https://www.nytimes.com/2000/03/06/business/bermuda-move-allows-insurers-to-avoid-taxes.html

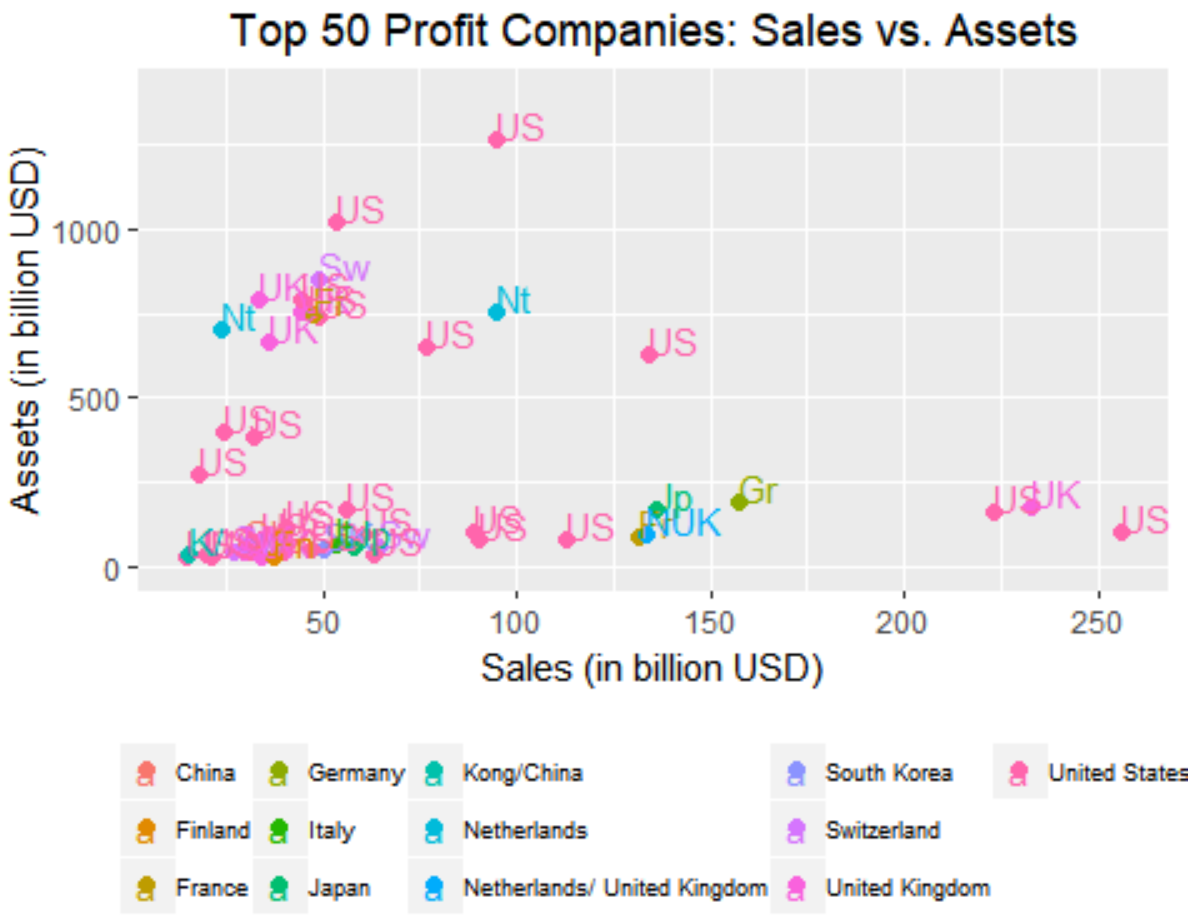
```
## [1] "The majority of Bermuda island companies belong to the Insurance category."
```

Problem #4: For the 50 companies in the Forbes data set with the highest profits, plot sales against assets labeling each point with the appropriate country name.

Results: The plots show that there is a negative correlation between the sales and asset variables. This is verified by the commented out code immediately proceeding the ggplot.

I started by ordering the data frame and selecting the top 50 'profit' values. Per the instructions, I then created 2 plots (one in base R and one using ggplot). The business life cycle tends to show declining sales as businesses achieve maturity. By the same token, I assume that assets are higher in businesses that have reached maturity. Therefore, I expected a negative correlation between assets and sales, which is shown by the plots.





Problem #5, Part 1: Find the average sales for the companies in each country in the Forbes data set.

Results: For *part one* of the problem, I used the 'dplyr' package to group countries together and summarize their sales means. To provide a clearer picture of the results, I presented them as a data frame in descending order. This data frame shows that the companies in the Netherlands/UK have much higher average sales than the rest of the countries. I found that to be unexpected, especially considering the large difference between them and the remaining countries.

To ensure that I hadn't made an error somewhere, I used subset() to take a look at the companies from Forbes2000 that had a 'country' value == "Netherlands/ United Kingdom". This explained the big gap as there are only 2 companies with that country value. The limited data set, and the relatively high sales from 'Royal Dutch/Shell Group', resulted in this outlier. I've commented out the code I used to derive this subset as it was not part of the exercise.

```
##                          country      mean
## 1   Netherlands/ United Kingdom 92.100000
## 2                        Germany 20.781385
## 3                         France 20.102063
## 4                    Netherlands 17.020714
## 5                          Korea 15.005000
## 6                     Luxembourg 14.185000
## 7                    Switzerland 12.456765
## 8     Australia/ United Kingdom 11.595000
## 9                         Norway 10.780000
## 10                United Kingdom 10.445109
```

```
## 11                      Finland 10.291818
## 12                        Italy 10.213902
## 13                        Japan 10.190633
## 14                      Belgium 10.114444
## 15                United States 10.058256
## 16   United Kingdom/ Australia 10.010000
## 17                  South Korea  7.969333
## 18                        Spain  7.843448
## 19                       Russia  7.672500
## 20                       Sweden  7.665769
## 21  United Kingdom/ Netherlands  7.540000
## 22                      Bermuda  6.840500
## 23                       Africa  6.820000
## 24                      Islands  6.670000
## 25                       Canada  6.429643
## 26                      Denmark  6.349000
## 27                       Brazil  6.338667
## 28       Panama/ United Kingdom  5.930000
## 29                   Kong/China  5.717500
## 30                    Australia  5.244595
## 31                        China  5.099600
## 32                      Ireland  4.765000
## 33                       Turkey  4.713333
## 34                       Poland  4.410000
## 35                      Austria  4.142500
## 36                 South Africa  4.124000
## 37                       Mexico  3.937647
## 38                     Portugal  3.884286
## 39                        India  3.868148
## 40                      Liberia  3.780000
## 41                    Singapore  3.685000
## 42                      Hungary  3.370000
## 43                       Taiwan  2.751429
## 44                  New Zealand  2.640000
## 45                       Greece  2.528333
## 46                     Thailand  2.513333
## 47                    Indonesia  2.450000
## 48                       Israel  2.060000
## 49 United Kingdom/ South Africa  2.060000
## 50               Hong Kong/China  2.044000
## 51                Czech Republic  1.805000
## 52                      Malaysia  1.716250
## 53                Cayman Islands  1.660000
## 54                         Chile  1.602500
## 55                    Philippines  1.565000
## 56                       Bahamas  1.350000
## 57                        Jordan  1.330000
## 58                      Pakistan  1.230000
## 59       France/ United Kingdom  1.010000
## 60                     Venezuela  0.980000
## 61                          Peru  0.170000
```

Problem #5, Part 2: Using the Forbes data set, find the number of companies in each country with profits above 5 billion US dollars.

Results: For the *second part* of the problem, I again used 'dplyr' package. First I created the subset of all companies with profit greater than 5.0 ($5 billion USD). Then I grouped the companies by 'country' and provided a count showing the number of companies, per country, that have profit in excess of 5 billion USD (n).

I expected that the US would dominate this list because nearly 40% of the companies on the list are from the US. The results confirmed my expectation.

```
## # A tibble: 9 x 2
## # Groups:   country [9]
##   country                        n
##   <fct>                      <int>
## 1 United States                 20
## 2 Switzerland                    3
## 3 United Kingdom                 3
## 4 China                          1
## 5 France                         1
## 6 Germany                        1
## 7 Japan                          1
## 8 Netherlands/ United Kingdom    1
## 9 South Korea                    1
```

Problem #6: Table 2.3 (household in the HSAUR3 package) shows the household expenditure of 20 single men and 20 single women on four commodity groups. The units of expenditure are Hong Kong dollars, and the four commodity groups are: housing, food, goods, and service. The aim of the survey was to investigate how the division of household expenditure between the four commodity groups depends on the total expenditure and to find whether the relationship differs for men and women.

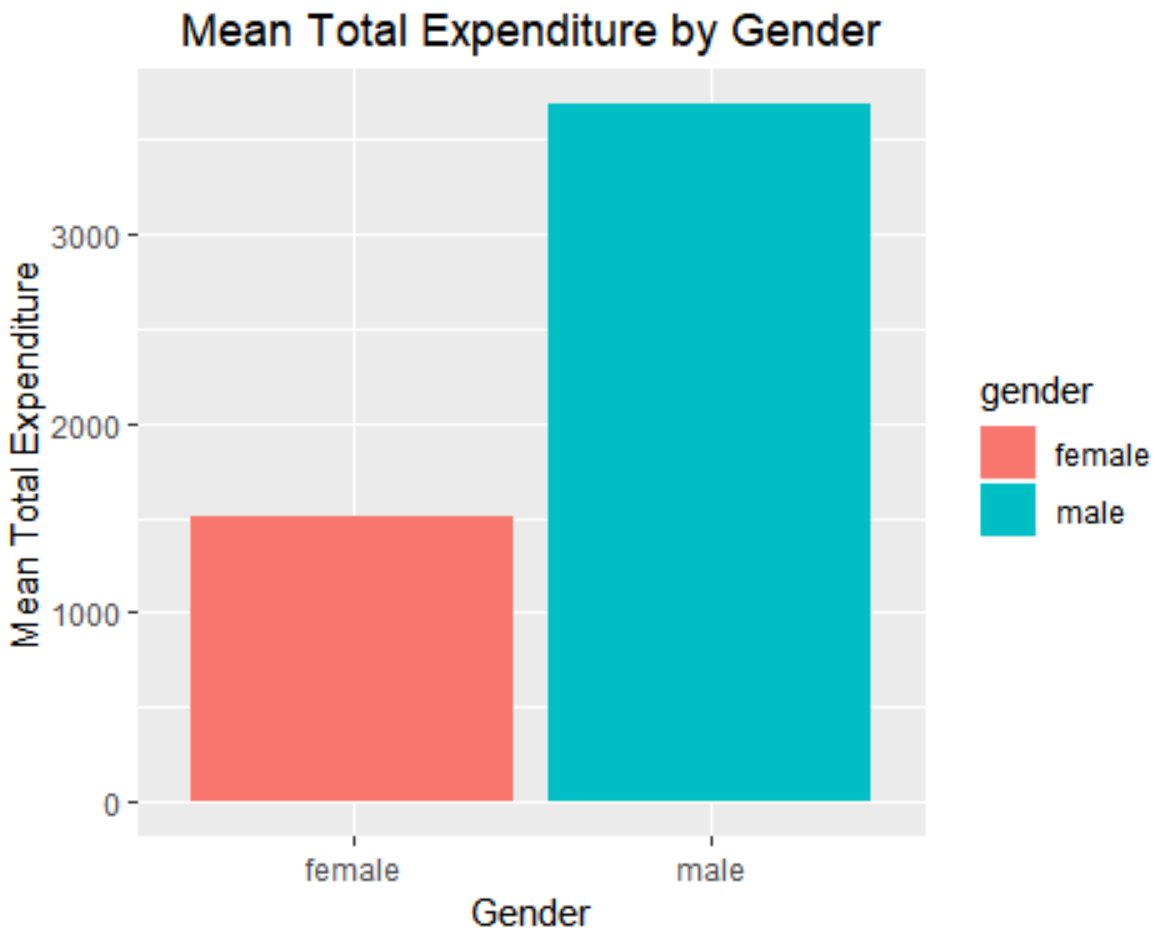
Results: My understanding of the goal of this exercise is to see how men and women's spending habits differ. The method for doing so, based on my assumption, is to look at their differences in total spending and at what percentage of their total expenditure comes from each commodity group. Graphs depicting these differences are on the next page.

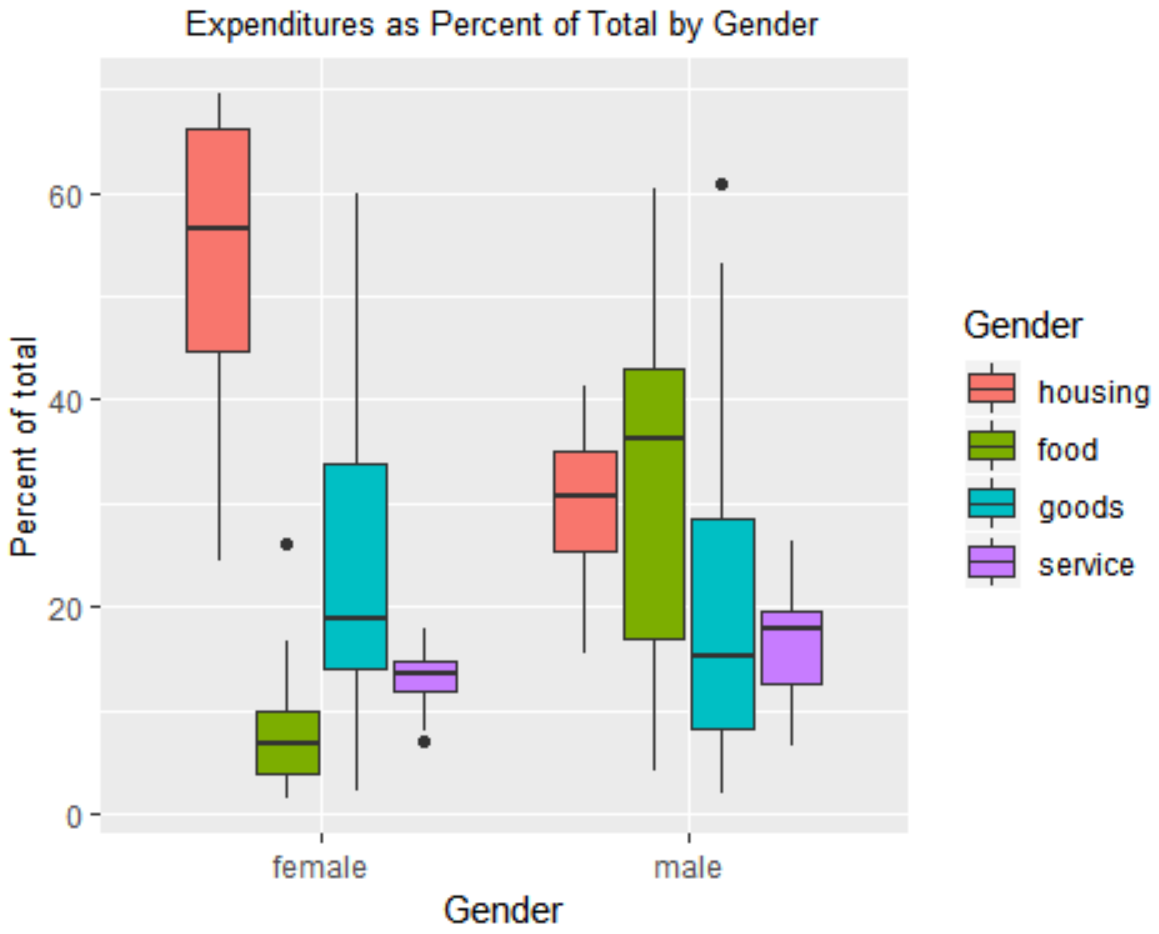
First, I examined the relationship between total spending and gender. As we can see from the histogram, the mean total spending for males far exceeded the mean total spending for females.

Next, I further examined the relationship between spending habits and gender. The results show that there is a significant difference, among some commodities, in the spending habits of the men and women surveyed.

To visually examine the relationship between spending habits and gender, I created boxplots to show the side-by-side comparisons. Immediately, 'housing' and 'food' stood out from the plots as different for men and women. The other two commodities ('goods' and 'service') were more similar. Based on this, I decided to use summaries of the aov() function to examine the differences closer.

From the set of aov summaries, we can see that there is a statistically significant difference between the 'housing' and 'food' spending habits, by gender, at 0.0. There is a statistically significant different in the 'service' spending habits between men and women at a level of 0.001. The spending habits relating to 'goods' did not differ in a statistically significant way between men and women.


Mean Total Expenditure by Gender
Mean Total Expenditure
3000
2000
1000
0
female
male
Gender
gender
female
male


Expenditures as Percent of Total by Gender
Percent of total
60
40
20
0
female
male
Gender
Gender
housing
food
goods
service

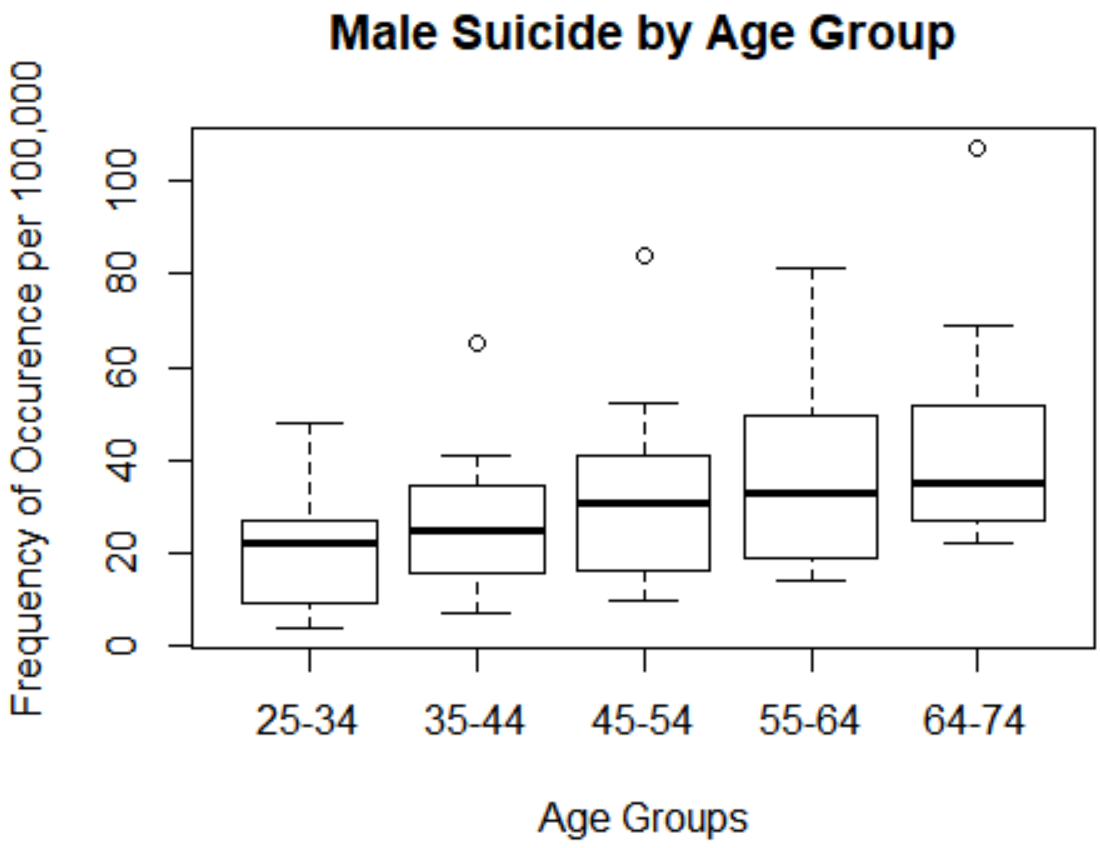
```
##                   Df Sum Sq Mean Sq F value   Pr(>F)
## household$gender  1   5260    5260   36.98 4.41e-07 ***
## Residuals        38   5405     142
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##                   Df Sum Sq Mean Sq F value   Pr(>F)
## household$gender  1   6194    6194   39.27 2.46e-07 ***
## Residuals        38   5994     158
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##                   Df Sum Sq Mean Sq F value Pr(>F)
## household$gender  1    341   341.4   1.105    0.3
## Residuals        38  11746   309.1

##                   Df Sum Sq Mean Sq F value  Pr(>F)
## household$gender  1  151.3  151.27   7.468 0.00948 **
## Residuals        38  769.7   20.25
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

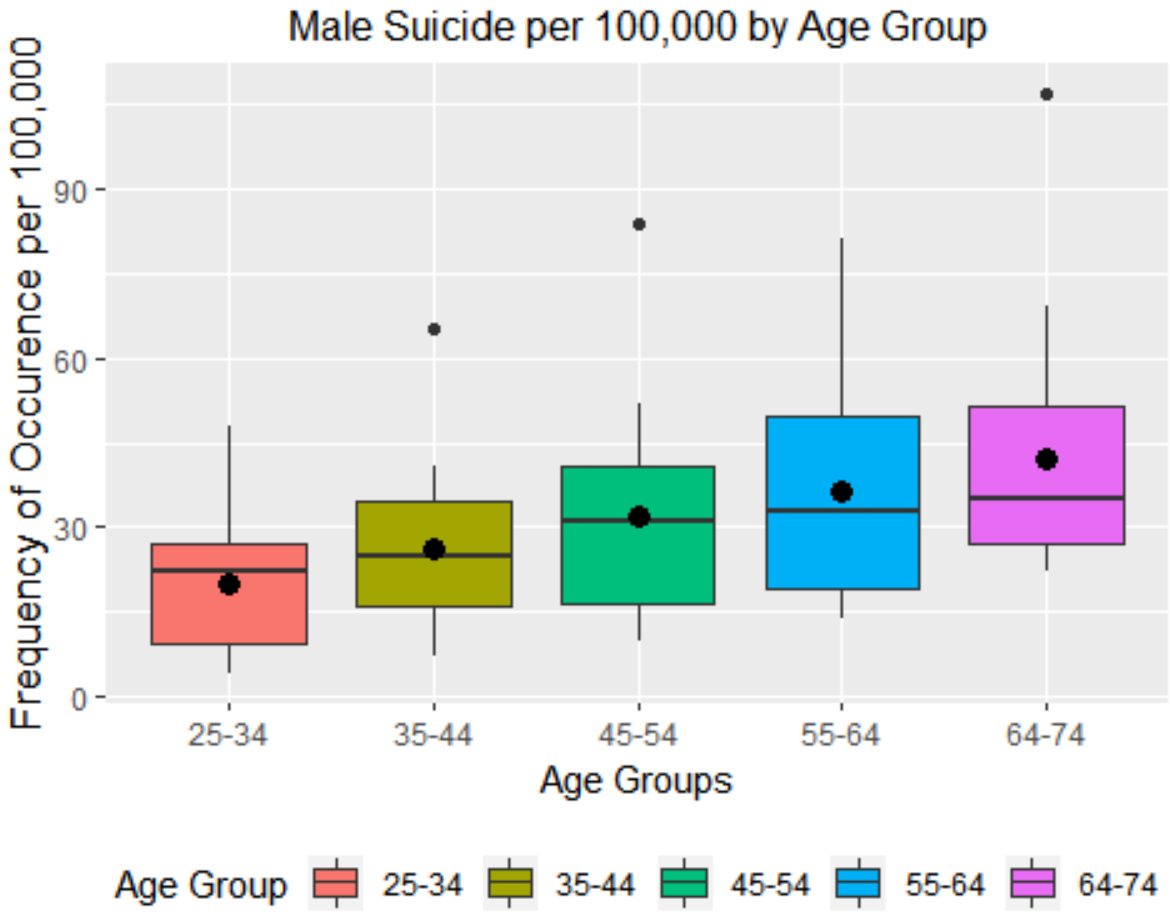
Problem #7: Mortality rates per 100,000 from male suicides for a number of age groups and a number of countries are given in Table 2.5 (suicides2 from the HSAUR package). Construct side-by-side box plots for the data from different age groups.

Results: The first step in the process was to make the age group columns more descriptive. I then used the melt() function from 'reshape2' to allow for plotting by age group.

The results show that the median and quartile values increase for each age group rather consistently. The increase from the 55-64 age group to the 65-74 age group was smaller than the preceeding consecutive groups. I also added the mean values to the ggplot boxplot, denoted by the black circle inside of the boxplot, and they seem to behave in a very consistent matter as well.

The results were the opposite of the results I expected. I had predicted that the suicide rate would decrease as men get older, presumably because maturity and rationality tend to increase as people age. These characteristics, in my personal opinion, are contrary to the act of suicide. Obviously, I was wrong in my prediction as shown by the plots.


Male Suicide by Age Group
Frequency of Occurence per 100,000
0
20
40
60
80
100
25-34
35-44
45-54
55-64
64-74
Age Groups


Male Suicide per 100,000 by Age Group
Frequency of Occurence per 100,000
0
30
60
90
25-34
35-44
45-54
55-64
64-74
Age Groups
Age Group
25-34
35-44
45-54
55-64
64-74

Problem #8: Using a single R expression, calculate the median absolute deviation, 1.4826 * median|x-mu|, where mu is the sample median. Use the dataset chickwts. Use the R function mad() to verify your answer.

Results: This problem is rather straight forward. I subtracted the median weight from the weight values, taking the absolute value of this differnce. Outside of this operation, I used median() to get the median value of the differences. This value was multiplied by the constant. This calculation took place within a single R expression.

To verify my answer, per the assignment instructions, I used the mad() function from the 'stats' package. My answer was the same as the one provided by the function. Both are shown below.

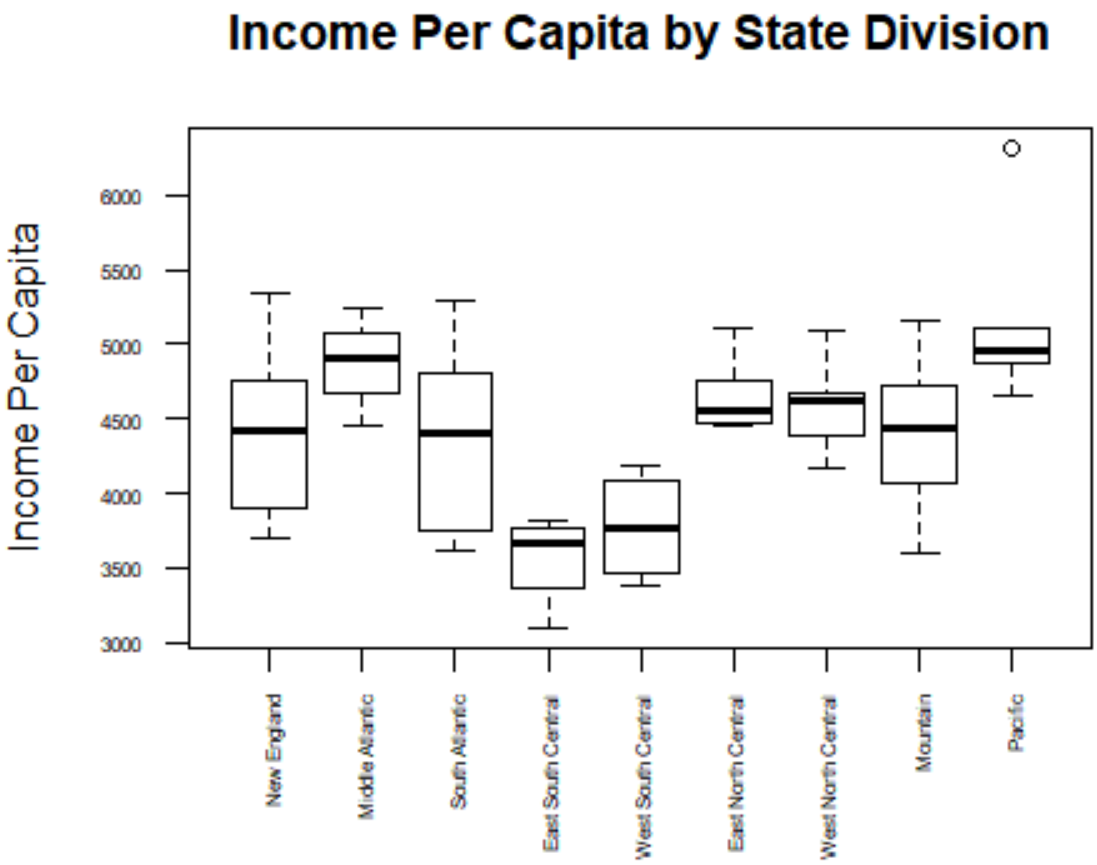
```
## [1] "Single expression to calculate the median absolute deviation:  91.9212"

## [1] "The mad() function to calculate the median absolute deviation: 91.9212"
```

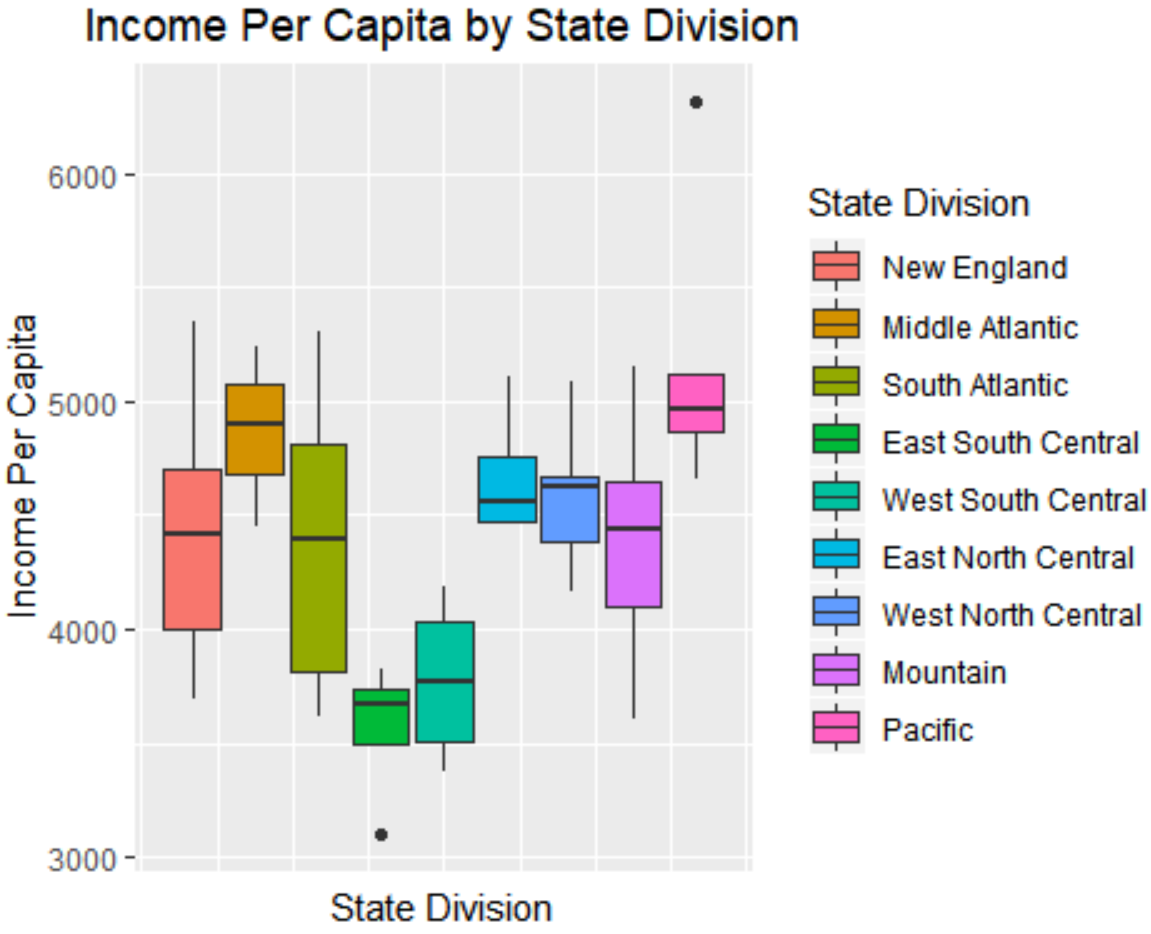
Problem #9: Using the data matrix 'state.x77', obtain side-by-side boxplots of the per capita income variable for the nine different divisions defined by the variable 'state.division'. Comment on the plot.

Results: For this exercise, I combined 'state.x77' and 'state.division' and then created 2 sets of boxplots, per the homework instructions. The first plot was done using base R and the second plot using ggplot2.

The plots followed the pattern I expected. Southern states, according to the plot, have a lower income per capita. Other divisions such as 'New England', 'Pacific', and 'Middle Atlantic', which I often think of as having higher incomes (and cost of living), had higher Income per Capita according to the plots.

The only complication I ran in to was in using the boxplot() function. The 'state.division' names were overlapping and some were being omitted from the plot all together. I attempted to use 'cex.axis' to get the divisions to fit but, in order to get them to fit, the 'cex.axis' value had to be so small that the words weren't legible. Therefore, I used 'las' to rotate the 'state.division' names to perpendicular to the axis which solved my issue.


Income Per Capita by State Division
Income Per Capita
6000
5500
5000
4500
4000
3500
3000
New England
Middle Atlantic
South Atlantic
East South Central
West South Central
East North Central
West North Central
Mountain
Pacific


Income Per Capita by State Division
Income Per Capita
6000
5000
4000
3000
State Division
State Division
New England
Middle Atlantic
South Atlantic
East South Central
West South Central
East North Central
West North Central
Mountain
Pacific

Problem #10: Using the data matrix state.x77, find the state with the minimum per capita income in the New England region as defined by the factor state.division. Use the vector state.name to get the state name.

Results: For this one I combined the three matrices and used the 'dplyr' library's mutate_if() function to change factor values to character values. Then I filtered 'New England' states and printed the 'state.name' corresponding to the min() 'Income' value.

I am not familiar enough with the living standards in the New England region to have formulated an expectation for this value. Therefore, I was not surprised when the result of the exercise was 'Maine'. Otherwise the problem was very straightforward and I had no complications.

```
## [1] "Maine has the minimum per capita income in the New England division."
```

Problem #11: Use subscripting operations on the dataset 'Cars93' to find the vehicles with highway mileage of less than 25 miles per gallon (variable 'MPG.highway') and weight (variable 'Weight') over 3500lbs. Print the model name, the price range (low, high), highway mileage, and the weight of the cars that satisfy these conditions.

Results: This problem was pretty straight forward. After loading in the dataset, I simply used subscripting and a single R expression to return the vehicles meeting the exercise's parameters and displaying the requested variables.

14 vehicles meet the parameters set out in the question and they are listed in the data set below.

```
##         Model Min.Price Max.Price MPG.highway Weight
## 16 Lumina_APV      14.7      18.0          23   3715
## 17      Astro      14.7      18.6          20   4025
## 26    Caravan      13.6      24.4          21   3705
## 28    Stealth      18.5      33.1          24   3805
## 36   Aerostar      14.5      25.3          20   3735
## 48        Q45      45.4      50.4          22   4000
## 49      ES300      27.5      28.4          24   3510
## 50      SC300      34.7      35.6          23   3515
## 56        MPV      16.6      21.7          24   3735
## 63   Diamante      22.4      29.9          24   3730
## 66      Quest      16.7      21.5          23   4100
## 70 Silhouette      19.5      19.5          23   3715
## 87     Previa      18.9      26.6          22   3785
## 89    Eurovan      16.6      22.7          21   3960
```

Problem #12: Form a matrix object named mycars from the variables Min.Price, Max.Price, MPG.city, MPG.highway, EngineSize, Length, Weight from the Cars93 dataframe from the MASS package. Use it to create a list object named cars.stats containing named components as follows: a) A vector of means, named Cars.Means b) A vector of standard errors of the means, named Cars.Std.Errors c) A matrix with two rows containing lower and upper limits of 99% Confidence Intervals for the means, named Cars.CI.99

Results: I formed the matrix with the requisite variables and named it 'mycars', per the instructions. Next, I created 'Cars.Means', 'Cars.Std.Errors' and 'Cars.CI.99'. I included these components in the list object named 'cars.stats' per the instructions.

The most difficult task of this exercise was part C. This was my first time creating a confidence interval in R. I used the qt() function and verified my results using the t.test() function. The t.test() code is commented out.

```
## $Cars.Means
## [1]   17.125806   21.898925   22.365591   29.086022    2.667742  183.204301
## [7] 3072.903226
##
## $Cars.Std.Errors
## [1]  0.9069210  1.1438051  0.5827473  0.5528742  0.1075695  1.5141964
## [7] 61.1694186
##
## $Cars.CI.99
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
## [1,] 14.74031 18.89034 20.83277 27.63178 2.384799 179.2215 2912.007
## [2,] 19.51131 24.90751 23.89841 30.54026 2.950685 187.1871 3233.799
```

Problem #13: Use the apply() function on the three-dimensional array iris3 to compute: a) Sample means of the variables Sepal Length, Sepal Width, Petal Length, Petal Width, for each of the three species Setosa, Versicolor, Virginica b) Sample means of the variables Sepal Length, Sepal Width, Petal Width for the entire data set.

Results: Here I followed the instructions and used the apply() function to calculate means of each variable, by species, and then the collective means of each variable for all species in the dataset. Knowing that each of the 3 species contained the same number of observations, I verified my results by averaging each of the four variables from my first output to confirm they matched my second output.

From the outputs, we can see that species 'Virginica' has the largest means for 'Sepal Length', 'Petal Length', and 'Petal Width'. The largest mean 'Sepal Width' belongs to the 'Setosa' species. 'Setosa' also has the smallest means for 'Sepal Length', 'Petal Length' and 'Petal Width'.

```
##      Species SepalLength.mu SepalWidth.mu PetalLength.mu PetalWidth.mu
## 1     Setosa          5.006         3.428          1.462         0.246
## 2 Versicolor          5.936         2.770          4.260         1.326
## 3  Virginica          6.588         2.974          5.552         2.026

##   Variables     Mean
## 1  Sepal L. 5.843333
## 2  Sepal W. 3.057333
## 3  Petal L. 3.758000
## 4  Petal W. 1.199333
```

Problem #14, Part A: Use the data matrix state.x77 and the tapply() function to obtain the mean per capita income of the states in each of the four regions defined by the factor state.region.

Results: For *#14, Part A*, I used the state.x77 'Income' variable and the 'state.region' factor, to calculate the Mean Income per Capita by region. This was done using the tapply() and mean functions. Here we can see that the south region has a much lower mean income per capita than the other 3 regions. The west region has the highest mean income per capita.

```
## # A tibble: 4 x 2
##   Region        IncPerCap.mu
##   <chr>                <dbl>
## 1 Northeast            4570.222
## 2 South                4011.938
## 3 North Central        4611.083
## 4 West                 4702.615
```

Problem #14, Part B: Use the data matrix state.x77 and the tapply() function to obtain the maximum illiteracy rates for states in each of the nine divisions defined by the factor state.division.

Results: For *#14, Part B*, I used the state.x77 'Illiteracy' variable and the 'state.division' factor, to calculate the the state with the highest illiteracy rate in each of the 9 divisions from 'state.division'. This was done using the tapply() and max functions. Again, I presented the data in a data frame for presentation purposes.

```
## # A tibble: 9 x 2
##   Division           Max.Illiteracy
##   <chr>                       <dbl>
## 1 New England                   1.3
## 2 Middle Atlantic               1.4
## 3 South Atlantic                2.3
## 4 East South Central            2.4
## 5 West South Central            2.8
## 6 East North Central            0.9
## 7 West North Central            0.8
## 8 Mountain                      2.2
## 9 Pacific                       1.9
```

Problem #14, Part C: Use the data matrix state.x77 and the tapply() function to obtain the number of states in each region.

Results: For *#14, Part C*, I used 'state.x77' and 'state.region' to calculate the number of states per region. Within my tapply() function, I specified length to get the count per region. This was the most challenging of the four exercises because of the extra step of finding a variable that contained all unique values and verifying that I did end up with 50 states once they were divided among the 4 regions. The south region has the most states with 16, while the northeast region has the fewest states at 9.

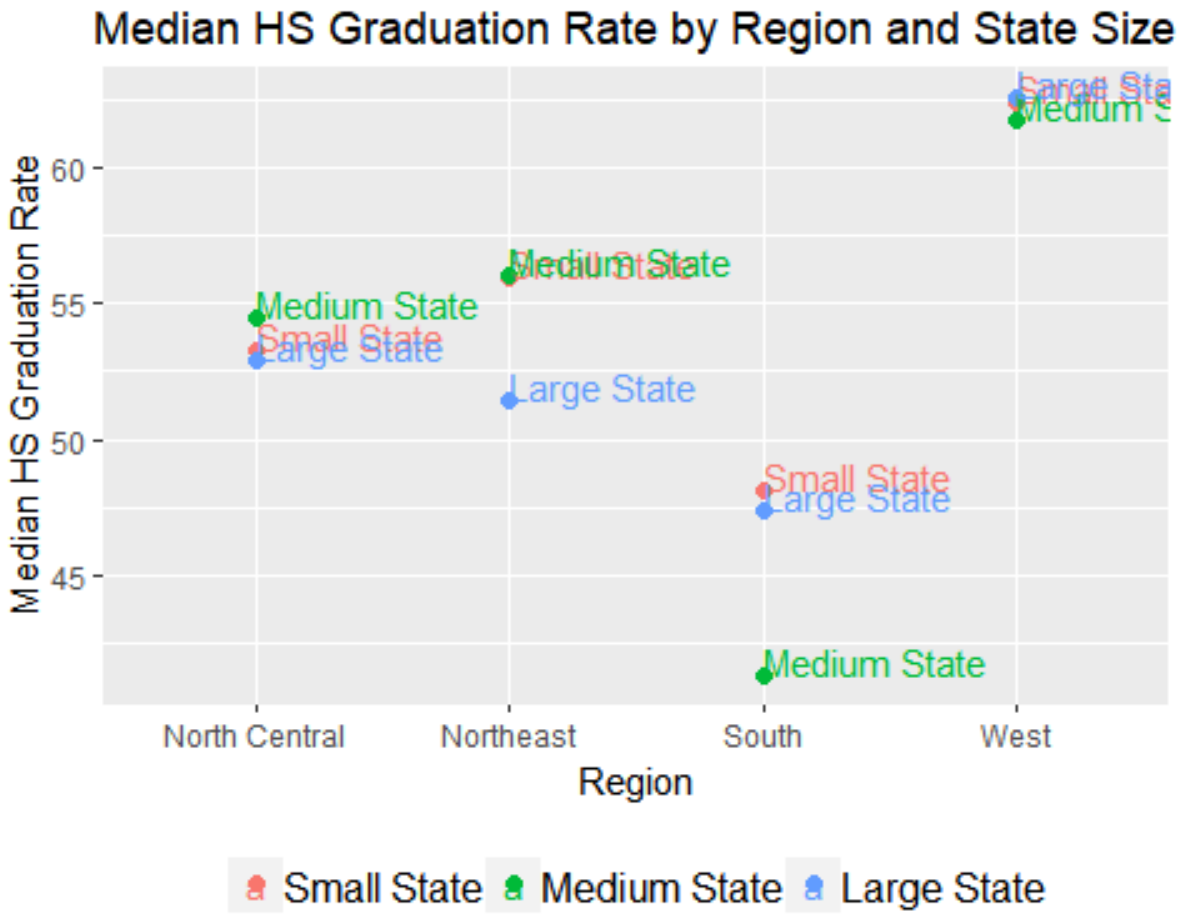
```
## # A tibble: 4 x 2
##   Region        NumberOfStates
##   <chr>                  <int>
## 1 Northeast                  9
## 2 South                     16
## 3 North Central             12
## 4 West                      13
```

Problem #14, Part D: Use the data matrix state.x77 and the tapply() function to obtain the median high school graduation rates for groups of states defined by combinations of the factors state.region and state.size.

Results: For *#14, Part D*, I did the same process as the prior 3 parts of this exercise. This time I produced the median grad rates by 'state.size' and region. I used the code provided to discern 'state.size'. Lastly, I used the melt() function to put the dataset into long form and plotted the Median Graduation Rate against Region, using the 'state.size' factor as my color.

From the plot, we see that 'Large States' in the West have the highest median graduation rate and 'Medium States' in the South have the lowest median graduation rate.

```
## # A tibble: 4 x 4
##   Region        GradMedian.Sm GradMedian.Med GradMedian.Lrg
##   <chr>                 <dbl>          <dbl>          <dbl>
## 1 Northeast              55.9           56             51.4
## 2 South                  48.1           41.3           47.4
## 3 North Central          53.3           54.5           52.9
## 4 West                   62.4           61.8           62.6
```



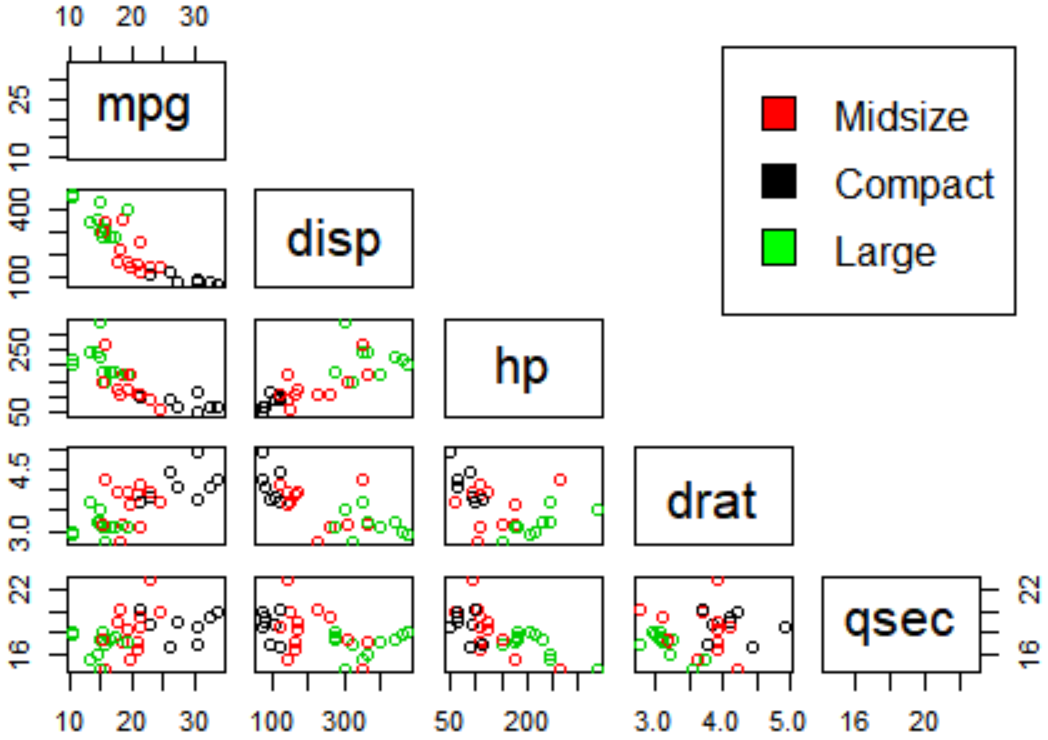
Problem #15: Using the dataframe mtcars, produce a scatter plot matrix of the variables mpg, disp, hp, drat, qsec. Use different colors to identify cars belonging to each of the categories defined by the carsize variable in different colors.

Results: I first used the code provided to create 'carsize' as instructed. Then to create the base R plot, I used pairs() to produce a scatter plot matrix. Positioning the legend of this type of plot was new to me and the most difficult portion of this part of the exercise.

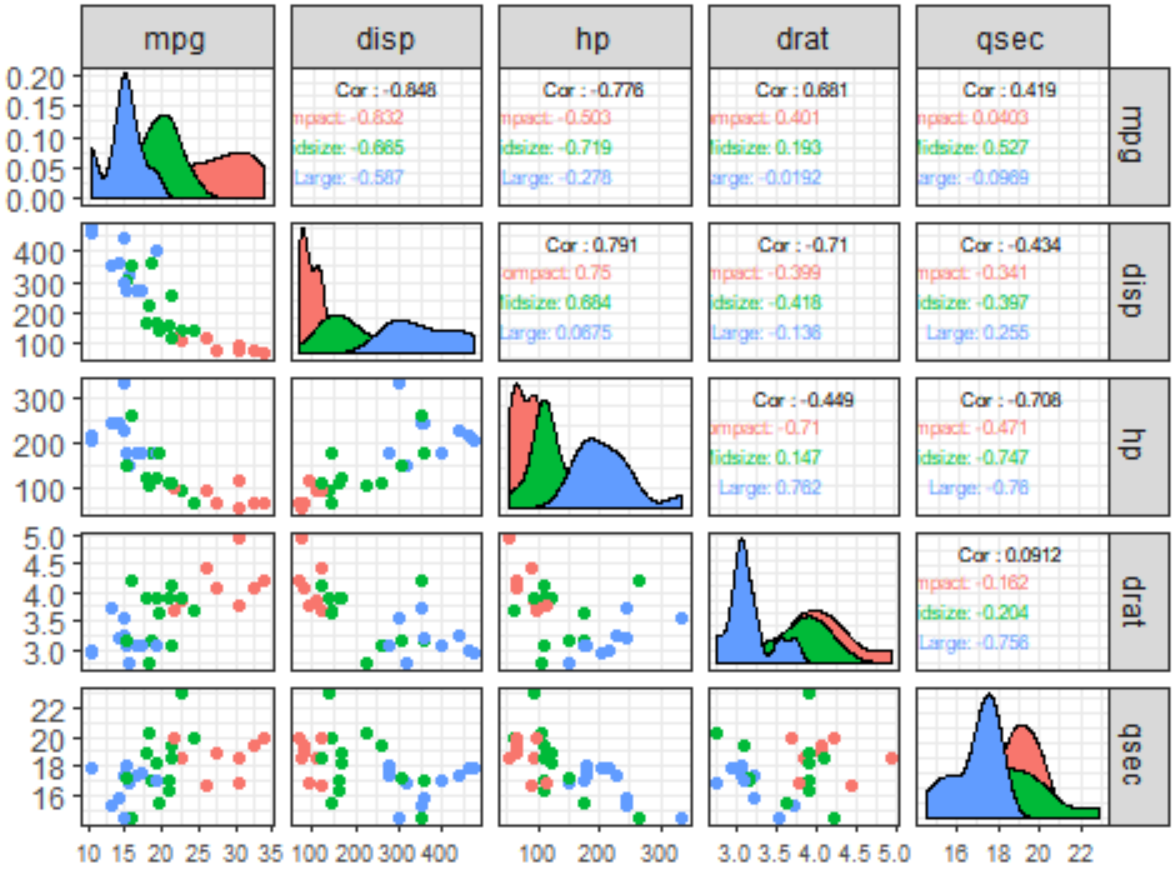
For the ggplot plot, I used ggpairs() from the 'GGally' library. This plot is easier to read, for me. I did not include a separate legend here because I wanted to show the upper.panel without covering the plot. Since the upper panel lists the 'carsize' in it's corresponding color, I believed this to be clear for someone else to read.

The two predictions that I had were that Horsepower and Miles per Gallon would be negatively correlated and that Horsepower and 'qsec' (1/4 mile time) would also be negatively correlated. These graphs confirm this prediction. Horsepower and MPG have a correlation of -0.776. Horsepower and 'qsec' have a correlation of -0.708. Both of these correlations are strong.

Motor Trend Car Road Test Scatter Plot Matrix



Motor Trend Car Road Test Scatter Plot Matrix



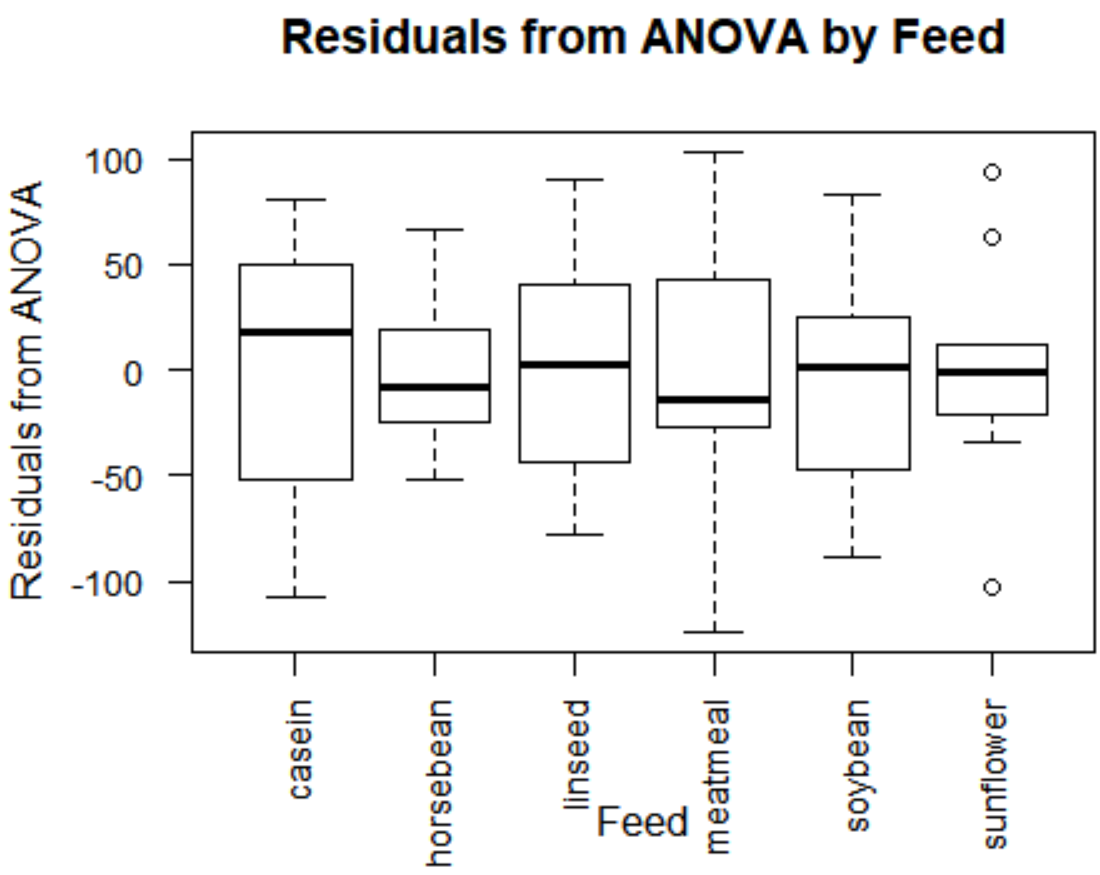
Problem #16: Use the function aov() to perform a one-way analysis of variance on the chickwts data with feed as the treatment factor. Assign the result to an object named chick.aov and use it to print an ANOVA table. Use this object to obtain side-by-side box plots of the residuals for each feed.

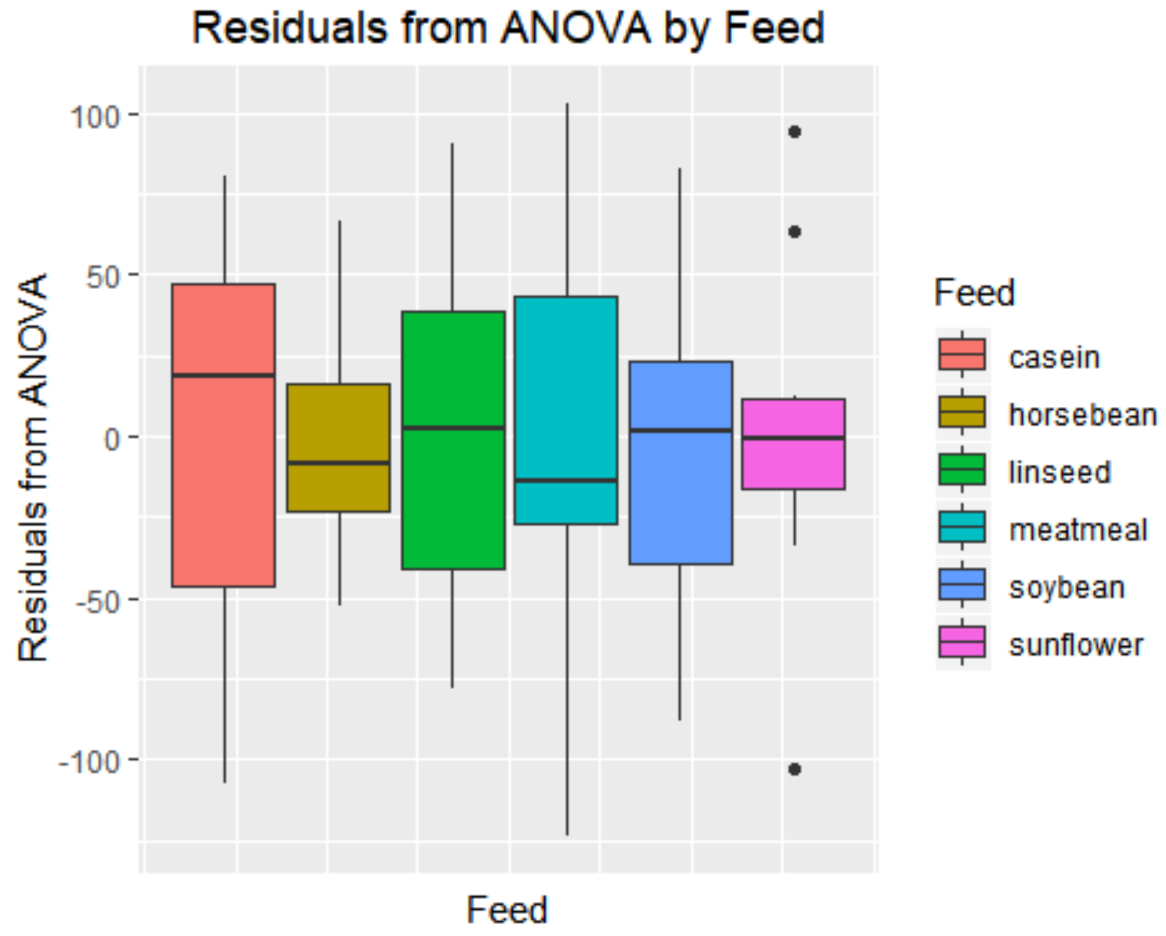
Results: I used the aov() function with feed as the treatment and weight as the response. Per the instructions, I set this equal to 'chick.aov' and printed results.

The side-by-side boxplots show residuals by feed. One plot, per the homework instructions, was created using the base R boxplot() function. The next plot was created using geom_boxplot from the 'ggplot2' library.

The sunflower 'feed' had the smallest distribution, but did contain the only outliers. 2 of these outliers were above the upper quartile and 1 was below the lower quartile. The largest variation is found in casein 'feed'.

```
## Call:
##    aov(formula = chickwts$weight ~ chickwts$feed)
##
## Terms:
##                 chickwts$feed Residuals
## Sum of Squares       231129.2  195556.0
## Deg. of Freedom             5        65
##
## Residual standard error: 54.85029
## Estimated effects may be unbalanced
```


Residuals from ANOVA by Feed
Residuals from ANOVA
100
50
0
-50
-100
casein
horsebean
linseed
meatmeal
soybean
sunflower
Feed


Residuals from ANOVA by Feed
Residuals from ANOVA
100
50
0
-50
-100
Feed
Feed
casein
horsebean
linseed
meatmeal
soybean
sunflower

Problem #17: Write an R function named ttest() for conducting a one-sample t-test. Return a list object containing the two components: - the t-statistic named T; - the two-sided p-value named P.

Use this function to test the hypothesis that the mean of the weight variable (in the chickwts dataset) is equal to 240 against the two-sided alternative. For this problem, please show the code of function you created as well as show the output.

Results: Here I created a function - ttest - per the instructions. The function accepts an 'x' and a 'mu' value. The first step in this function is to calculate 'T', which is the t-statistic. To do so, we subtract the sample mean from the population mean, given in the instructions as *240*. Then we divide by the standard deviation divided by the square root of the sample size (n or length(x)).

The next step is to calculate the 'P', or the two-sided p-value. To do so, we use the pt() function which accepts the 'T' and the degrees of freedom of the sample. The result of the pt() function is multiplied by 2 due to it being a two-sided p-value.

Then, per the instructions, the 'ttest' function returns a list object containing both the 'T' and the 'P'.

My last step was to verify the results by calling R's t.test() function. In doing so, I see that my results match the R function's results.

The one difficulty that I had with this exercise was that I forgot to multiply the result from the pt() function by 2. Once I remembered to do this step, the result matched the R function.

```
# created function 'ttest' that accepts 'x' and 'mu' and returns list of t-statistic(T) a
nd p-value(P)
ttest <- function(x, mu) {
  T <- (mean(x)-mu) / ((sd(x)/(sqrt(length(x)))))
  P <- 2*(pt(T, df=(length(x)-1), lower.tail = FALSE))
  print(list(c(T,P)))
}
# I called 'ttest' to test the hypothesis at mean = 240
ttest(chickwts$weight, mu=240)

## [[1]]
## [1] 2.29987903 0.02444107

# verified results
# t.test(x=chickwts$weight, mu=240, conf.level = 0.95)
```