Analysis on Car Prices through Correlation with `corrplot`

Dominic Tu

12/2/2017

Post 2

Introduction

Throughout my time in Stats 133, I have learned that many findings and graphs dealt with correlation. Correlation is a huge part in predicting and analysis. I want to dive deeper into the topic of correlation using `corrplot`. This package allows users insightful different correlograms. It graphicallly displays correlation matrix and confidence interval. The package also contains some algorithms to do matrix reordering. The details like text label, color labels, and layour adds to the success and impactfulness of the package.

However, I believe the one post that I have used a correlogram is not enough to learn about all the functionality available in `corrplot`. I want to use this post to dive deeper into `corrplot`, using the other graphing tools that we have not touched in class. We will use these function to further analyze a certain data set, specifically data in car prices.

Coming into my adult life, I have been interested wage trends for young male. I believe it is important to spend time researching the trends before acting on certain career strategies. I though it was a great opportunity to analyze data in the wages similar to the data set we work with in this course. So the analysis should seem familiar, but with a refreshing new set of numbers.

Through this post, you will be able to full immerse yourself into `corrplot` and finally mastering a package that is extremely helping in data reporting and data visualization. I want help other solidify or improve on their current skills in R. This post will be a learning to tools for other and a project for my our curiousity.

Data Preparation

First we have to load `ggplot2` and download the dataset.

```
#load package corrplot
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.4.2
```

```
## corrplot 0.84 loaded
```

```
#load data set
github <- "https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/Stat2Data/"
csv <- "FirstYearGPA.csv"
download.file(url = paste0(github, csv), destfile = 'FirstYearGPA.csv')
dat <- read.csv('FirstYearGPA.csv', stringsAsFactors = FALSE)
```

Correlogram

1. Different Visualization Methods

There are 7 different visualization methods in `corrplot`: square, circle, ellispe, numer, shade, color, pie. `Corrplot` only takes input that are a matrix, not a dataframe. So we have to do some data prepation, converting the dataframe into a matrix. In this case, we are going to make a matrix out of the variable correlations

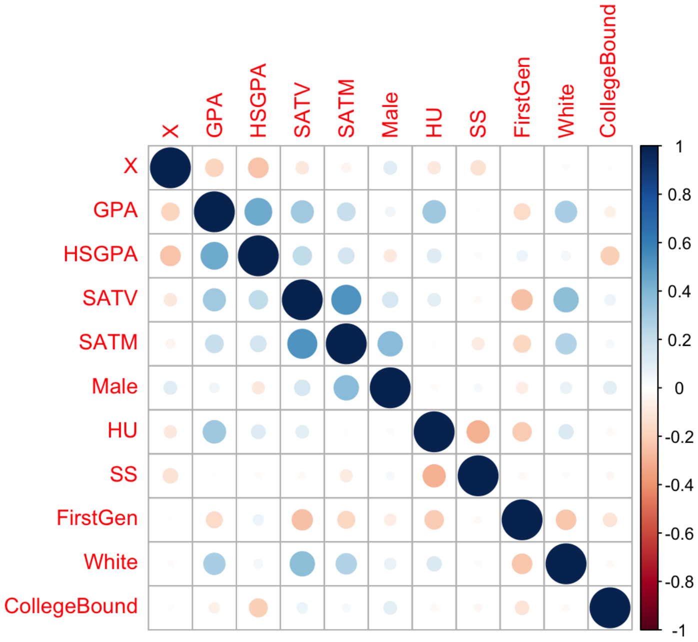
```
#data preparation
#converting the dataframe into a correlation matrix (correlogram) using the function cor()
GPAcorr = cor(dat)
GPAcorr
```

```
##                         X         GPA       HSGPA        SATV         SATM
## X             1.000000000 -0.18693938 -0.24169420 -0.09436226 -0.047312698
## GPA          -0.186939380  1.00000000  0.44688735  0.30431137  0.194343851
## HSGPA        -0.241694202  0.44688735  1.00000000  0.21032124  0.152839634
## SATV         -0.094362258  0.30431137  0.21032124  1.00000000  0.526943819
## SATM         -0.047312698  0.19434385  0.15283963  0.52694382  1.000000000
## Male          0.101212743  0.05284917 -0.09031714  0.14555703  0.370991668
## HU           -0.091577232  0.31465575  0.11603117  0.09874856 -0.009601549
## SS           -0.129656169 -0.00356909 -0.01725443 -0.02646987 -0.087839974
## FirstGen      0.002725596 -0.15657732  0.06418575 -0.25657713 -0.177387395
## White        -0.015249216  0.28177214  0.04604668  0.36823365  0.259465227
## CollegeBound -0.010257334 -0.06302497 -0.20003903  0.06484473  0.039322063
##                     Male           HU          SS     FirstGen       White
## X             0.10121274 -0.091577232 -0.12965617  0.002725596 -0.01524922
## GPA           0.05284917  0.314655754 -0.00356909 -0.156577322  0.28177214
## HSGPA        -0.09031714  0.116031169 -0.01725443  0.064185751  0.04604668
## SATV          0.14555703  0.098748556 -0.02646987 -0.256577125  0.36823365
## SATM          0.37099167 -0.009601549 -0.08783997 -0.177387395  0.25946523
## Male          1.00000000 -0.018843863  0.03507603 -0.076105261  0.07696022
## HU           -0.01884386  1.000000000 -0.30660787 -0.212565615  0.12593391
## SS            0.03507603 -0.306607866  1.00000000 -0.023663260  0.01673417
## FirstGen     -0.07610526 -0.212565615 -0.02366326  1.000000000 -0.23790392
## White         0.07696022  0.125933908  0.01673417 -0.237903920  1.00000000
## CollegeBound  0.09981773 -0.029972303 -0.03170649 -0.110511005 -0.02391156
##              CollegeBound
## X             -0.01025733
## GPA           -0.06302497
## HSGPA         -0.20003903
## SATV           0.06484473
## SATM           0.03932206
## Male           0.09981773
## HU            -0.02997230
## SS            -0.03170649
## FirstGen      -0.11051100
## White         -0.02391156
## CollegeBound   1.00000000
```

As you can see, although cor shows that cor() gets the correlation of each variable, it is very easy to over look numbers. Correlation is best visualized by a graph. I still introduce the different type of correllogram possible through `corrplot`

Circle

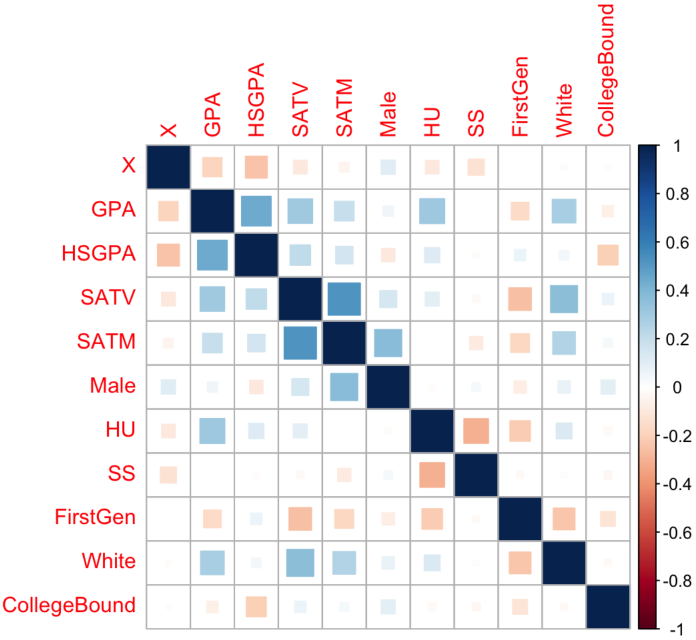
```
corrplot(GPAcorr)
```



Analysis: The default shape of `corrplot` is circle. The spectrum of colors displays the correlation between variables through a color visualization. The red means a negative correlation. White means no correlation. Blue means a positive correlation. The size of the circle shows the magnitude of the correlation.

Square

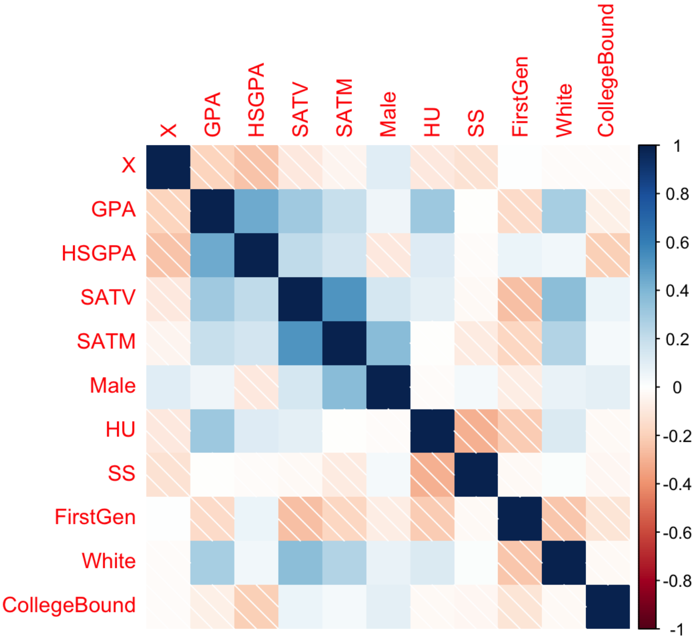
```
corrplot(GPAcorr, method = 'square')
```



Analysis: The color spectrum are the same as the circle method. This method is very similar to previous method. The size of the square shows the magnitude of the correlation. For example, we see that HS GPA has a high correlation to College GPA, indicated with a big darkish blue.

Shade

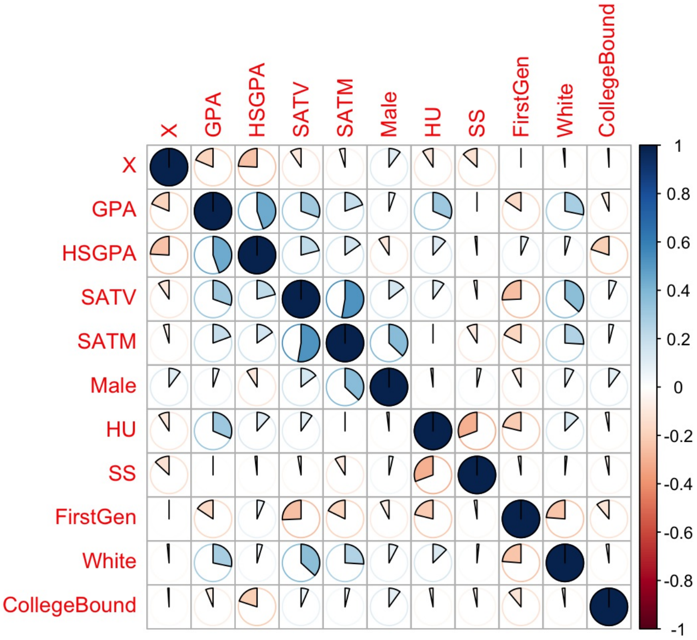
```
corrplot(GPAcorr, method = 'shade')
```



Analysis: With the shade methods, the visualization is pretty similar. The color representation is same with red symbolizing a negative correlation and blue symbolizing a positive correlation. However, the size of the square is all the same. A negative correlation is represented with a white-diagonal lines within the square. A positive correlation is represented with a solid square.

Pie

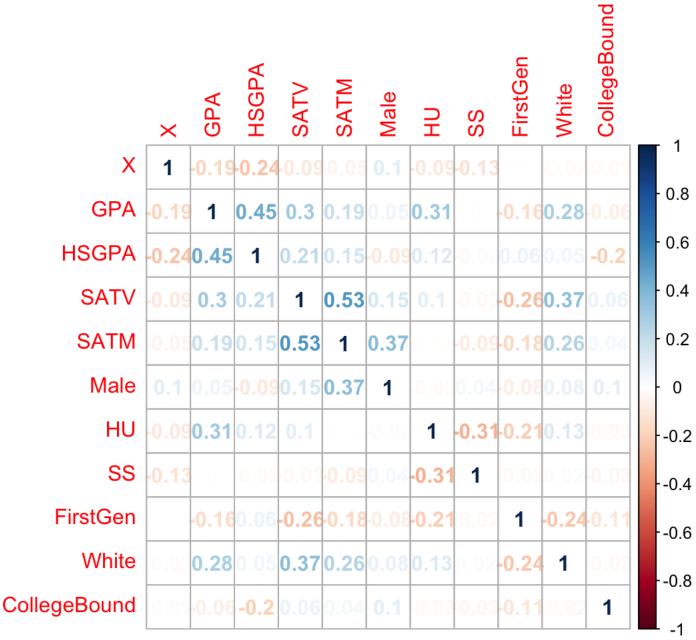
```
corrplot(GPAcorr, method = 'pie')
```



Analysis: As previously mentioned before, the color of the pie and color spectrum represent the correlation of the variables. The proportion of each pie shows the magnitude of the correlation.

Number

```
corrplot(GPAcorr, method = 'number')
```



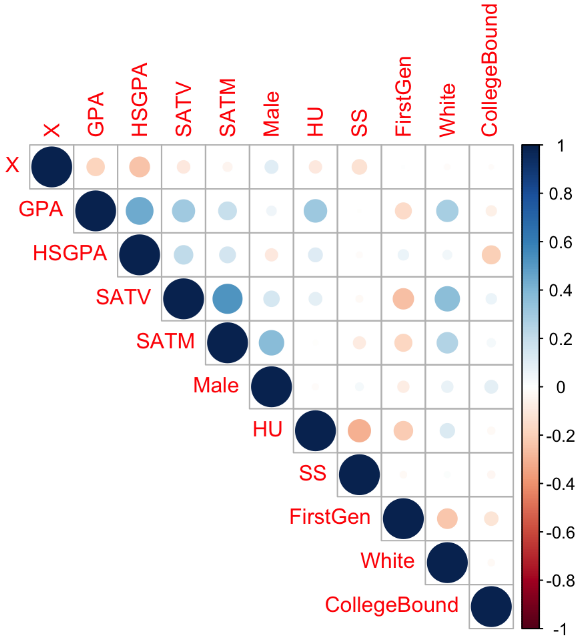
Analysis: Corrplot also allows us to use number to represent the correlation with color. Together, the numbers and shade of color represent the specific correlation of the variables.

2. Layout

In `corrplot`, there are three different layouts: full, upper, and lower. The default layout that have been used in the previous correlation have all been “full”. In this section, I’ll show the other types of layouts.

Upper

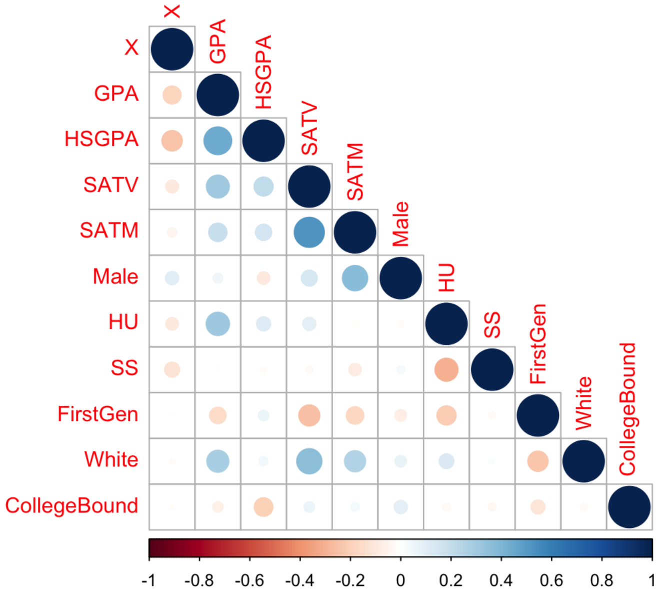
```
corrplot(GPAcorr, type = 'upper')
```



Analysis: The upper layout shows the upper triangle of the correlation matrix.

Lower

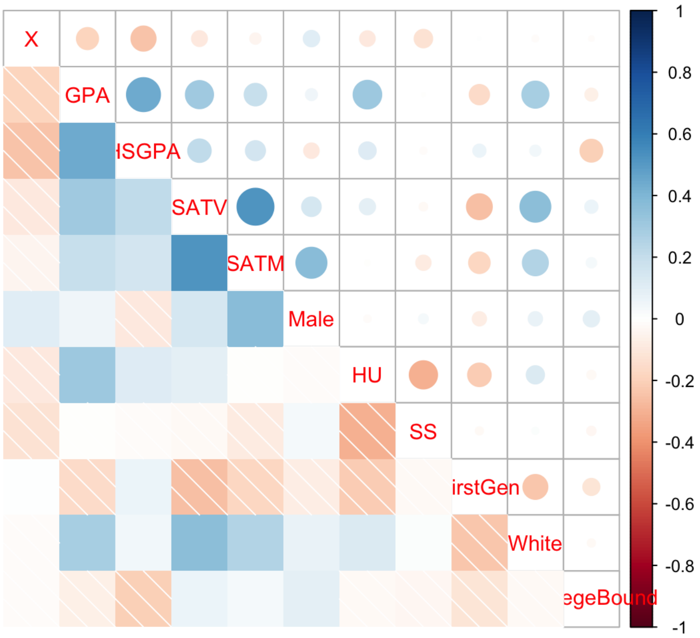
```
corrplot(GPAcorr, type = 'lower')
```



Analysis: The lower layout shows the lower triangle of the correlation matrix. If you have noticed, the full is the combination of both the upper and lower layout. Layout is just for the user’s preference.

Mixed

```
corrplot.mixed(GPAcorr, lower = 'shade', upper = 'circle')
```



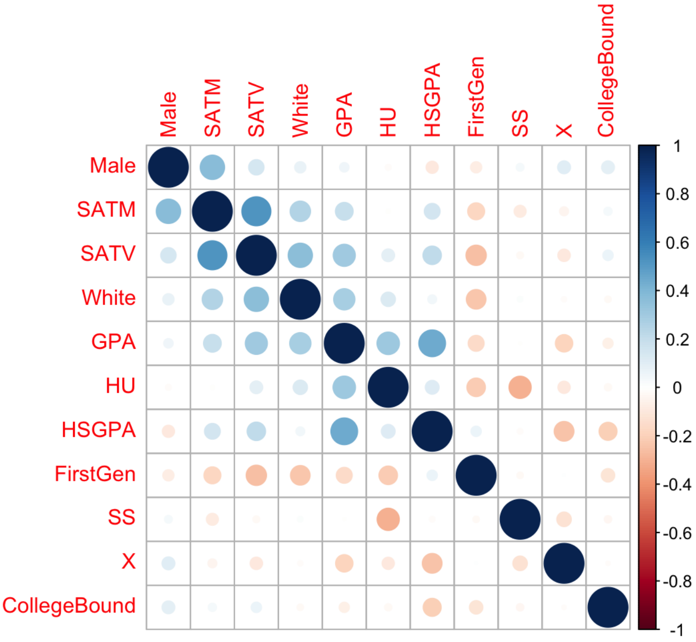
Analysis: With the function `corrplot.mixed`, we are able to combind two different methods separated by the upper and lower. The variables are labeled diagonally in the graph. It is nice to display data in different ways.

3. Reorder

Although the graph is a huge improvement in interpreting correlation data, the correlation matrix can be reordered by the correlation coefficient. There are four method in `corrplot`: AOE, FPC, hclust, and alphabet.

AOE

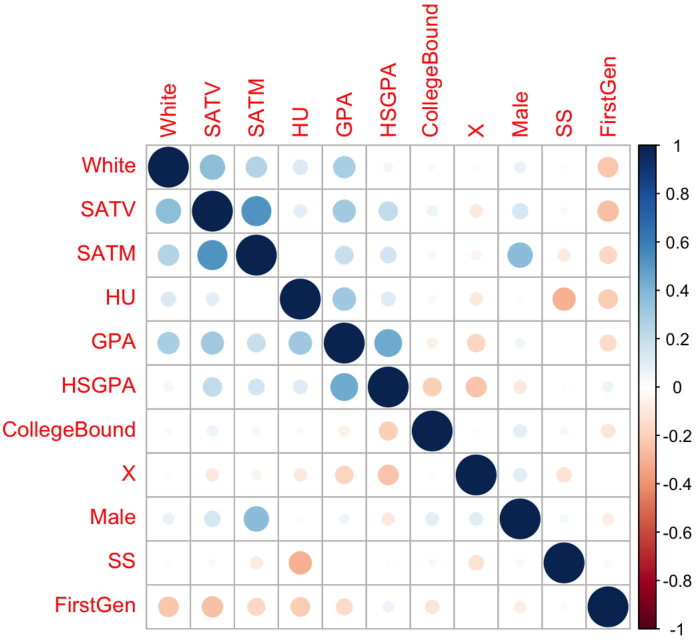
```
corrplot(GPAcorr, order = "AOE")
```



Analysis: AOE order the matrix through the argular order of eigenvector. All the positive correlation are group in the upper left corner. The higher the magnitude, the closer the points are to the middle diagonal.

hclust

```
corrplot(GPAcorr, order = 'hclust')
```

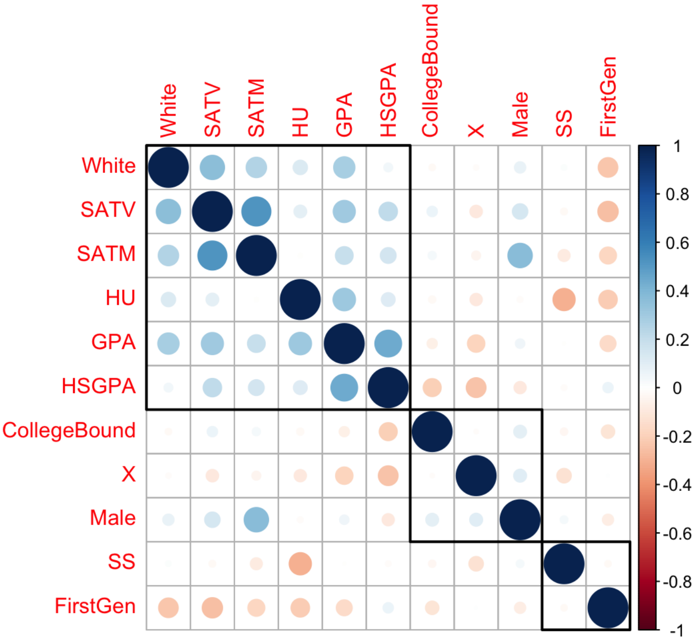


Analysis: `hclust` stands for heirarachial lustering order. The function performs a “hierarchicial luster analysis”, which uses dissimilarities sets of the variables. Again in this order, the positive correlated points are aggregate in the top. For example, we can see that ‘White’ and ‘SAT scores’ has a positive correlation with GPA.

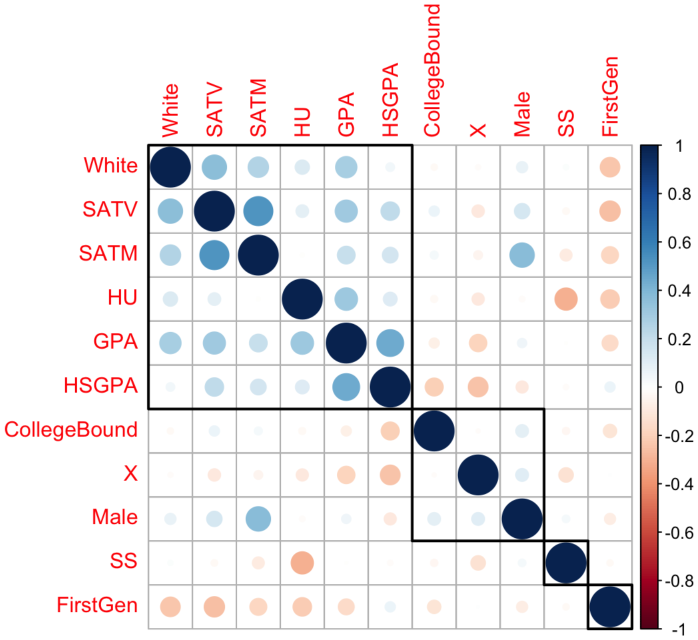
Retangles around the Clustering

When using `hclust`, we can draw rectangles in the correlogram around the clusters based on hierarchy.

```
corrplot(GPAcorr, order = 'hclust', addrect = 3)
```



```
corrplot(GPAcorr, order = 'hclust', addrect = 4)
```



Analysis: As you can tell, there are rectangles drawn around the clusters all near the middle diagonal. This clearly directs attention to the more important and high magnitude correlations. the addrect controls the numbers rectangles drawn on the correlogram. It can identify the different clustering.

As previously mentioned before, we see that being White, higher SAT score, or higher HU has a strong correlation with overall GPA.

Conclusion

This concludes my deeper `corrplot` tutorial. We dive deeper into correlograms through different shapes, reordering, and tricks. For this specific post. we found out the correlations to GPA. Higher SAT and HU scores, as well as being white has a high correlation with GPA. Whereas, we see first generation student have a difficult time in school, which is quite understable. `corrplot` is a very powerful tool for data visualization. This is will help you in any statistical/data analysis. Thank you for tuning into my post!

Reference

- https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html
- https://stat.ethz.ch/R-manual/R-devel/library/stats/html/hclust.html
- http://rpubs.com/melike/corrplot
- https://raw.githubusercontent.com/vincentarelbundock/Rdatasets/master/csv/datasets/crimtab.csv
- https://cran.r-project.org/web/packages/corrplot/corrplot.pdf
- https://www.rdocumentation.org/packages/corrplot/versions/0.2-0/topics/corrplot
- http://www.sthda.com/english/wiki/visualize-correlation-matrix-using-correlogram