Stats503_HW1

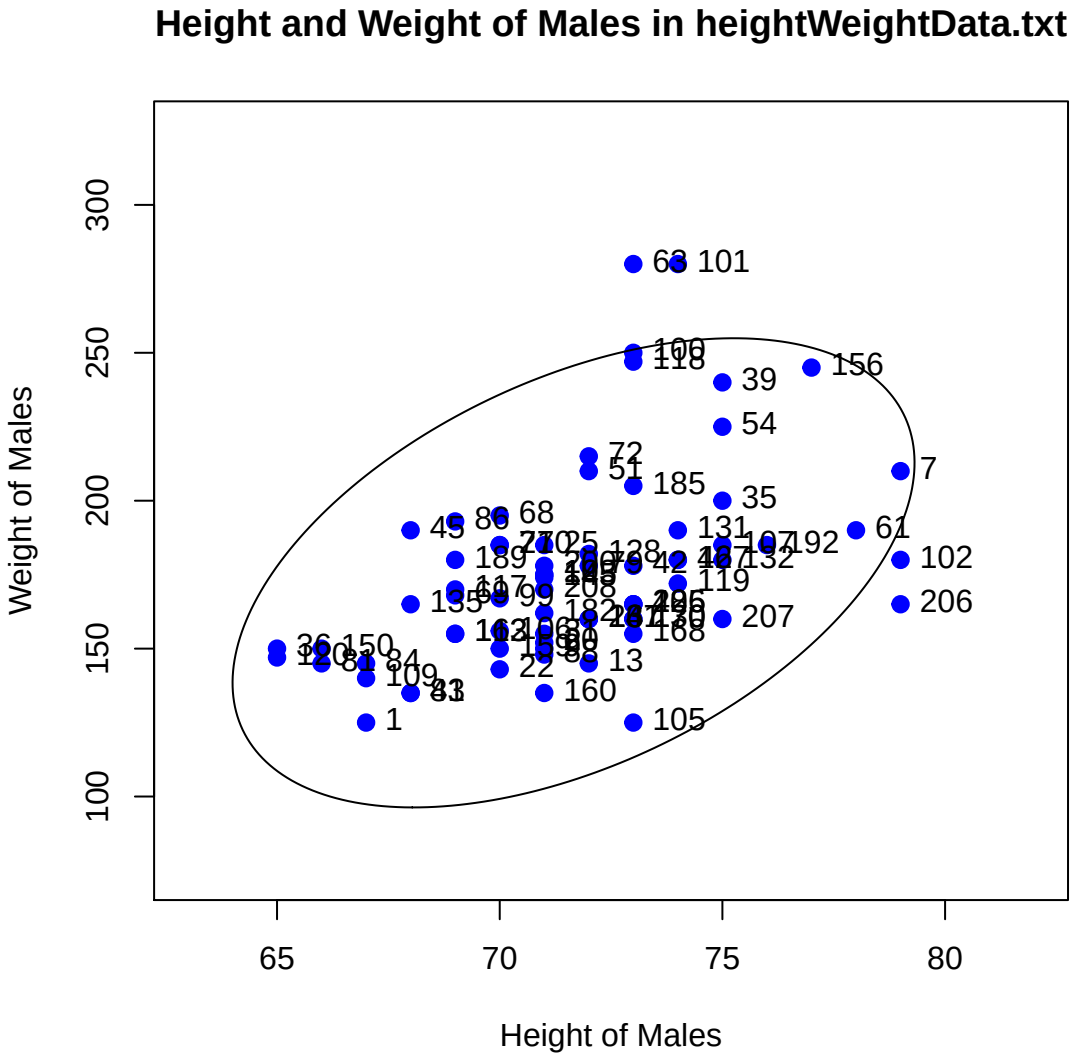
David Li

January 24, 2018

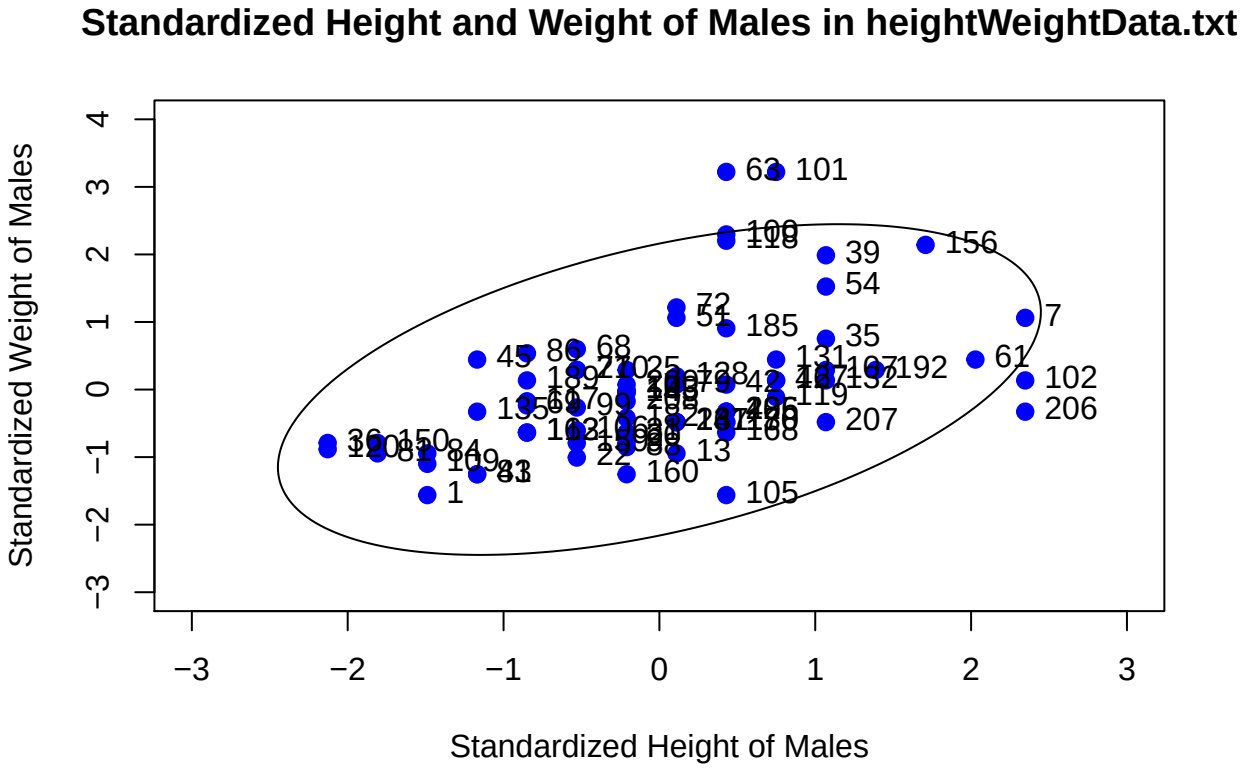
Purpose: This is a data analysis report for Stats 503 Homework 1.

Problem 3: Whitening and Standardizing

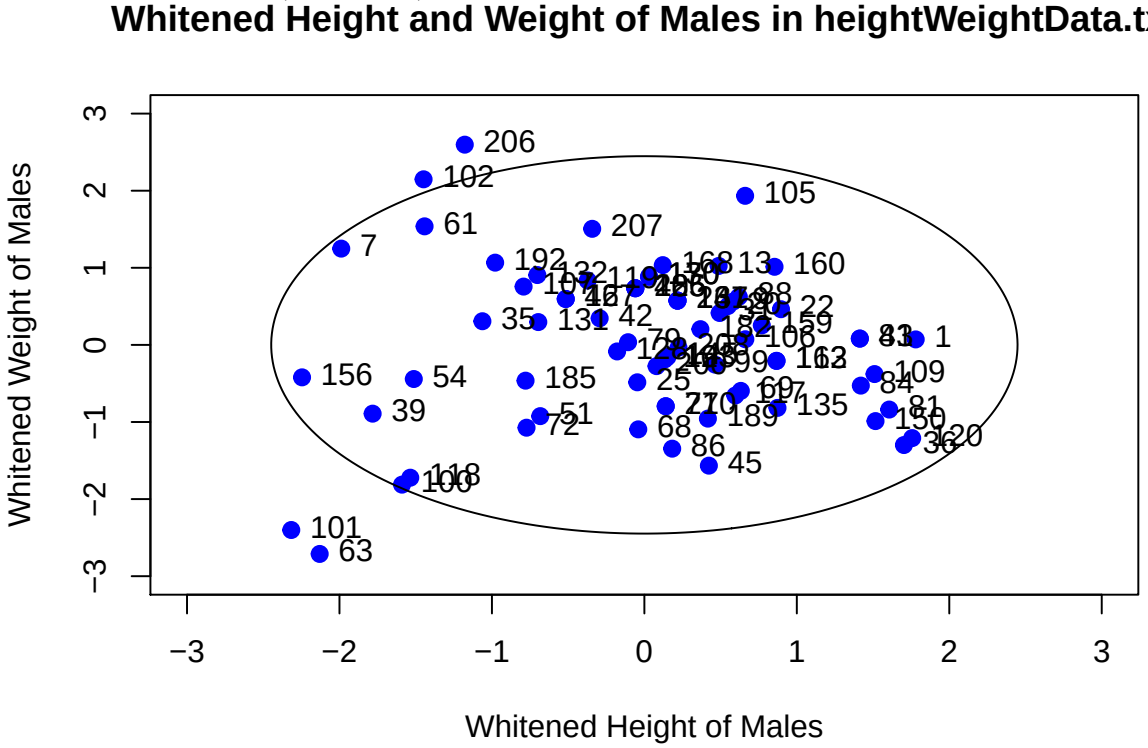
(a) Load the height/weight data from heightWeightData.txt in Canvas. The first column is the gender label (1 for male and 2 for female), the second coumn is height, the third weight. Extract the height/weight data corresponding to the males. Fit a 2-dim Gaussian to the male data, using the empirical mean and covariance. Plot your Gaussian distribution as an ellipse, superimposing on your scatter plot of data points, each which should be labeled by its index number (ranging from 1 to 210).



(b) Standardizing the data means ensuring the empirical variance along each dimension is 1. This can be done by computing (xij−x(bar)j) / j, where j is the empirical std of dimension j, x(bar)j the empirical mean. Standardize the data and replot.



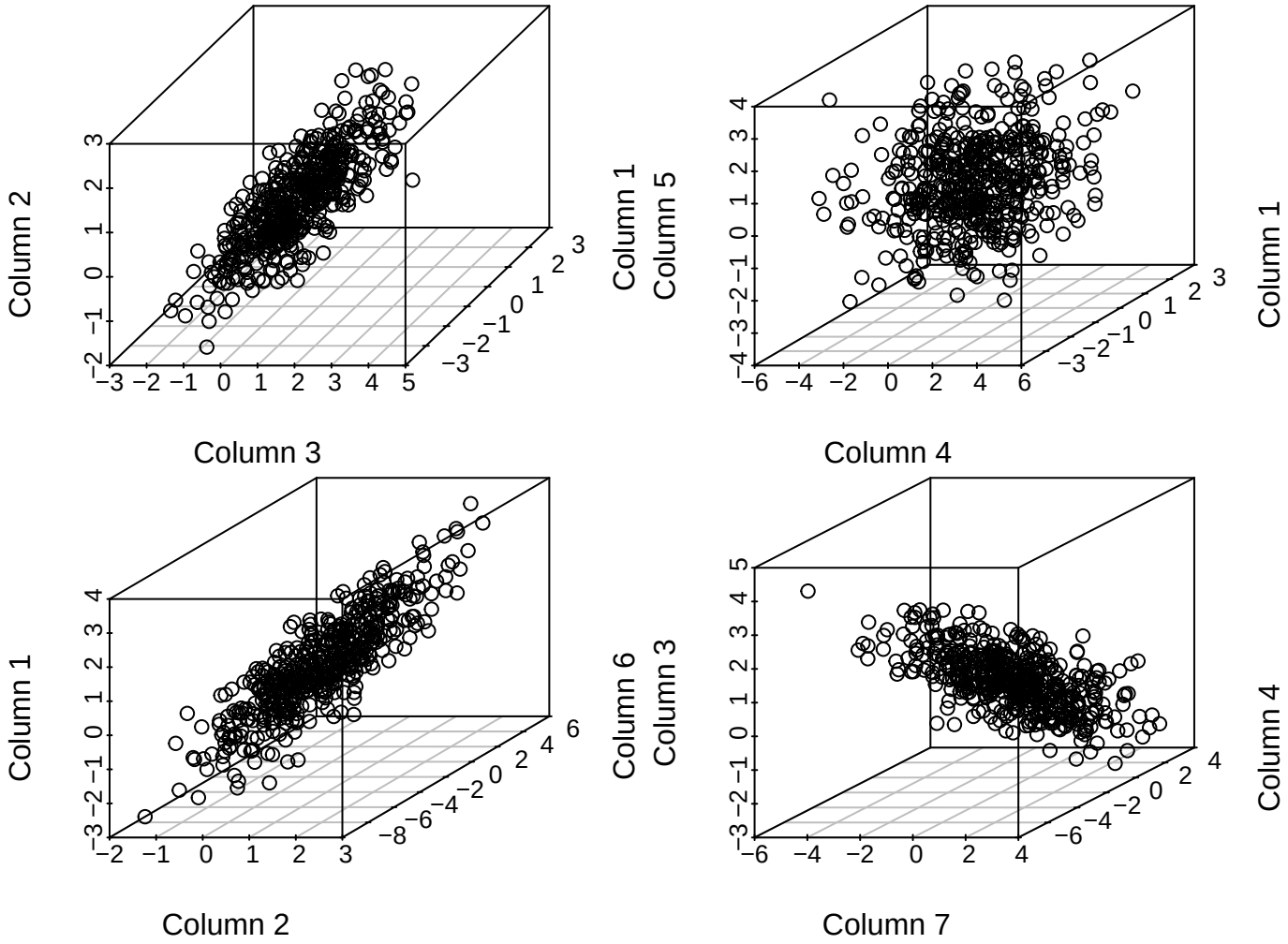
(c) Whitening or sphereing the data means ensuring its empirical covariance matrix is proportional to identity matrix, so the data is uncorrelated and of equal variance along each dimension. This can be done by computing $\Lambda^{(-1/2)}$ U^T x for each data vector x, where U are the eigenvectors and Λ the eigenvalues of the covariance matrix X^T X. Whiten the data and replot. Note that whitening rotates the data, so people (data points) move to counter-intuitive locations in the new coordinate systems.



Problem 4: PCA Warmup

On the class website there is a data set fa-data.txt consisting of 500 data points in 7 dimensions. The data are believed to lie mostly near a 2-dim linear submanifold. (Please see fa-gendata.m for the Matlab code I used to generate the data set).

(a) Produce a few visualizations of the data set by plotting only 3 dimensions selected randomly. Do you see that the data indeed lie near a 2-dim subspace?



Remarks: Some examples of 3D scatterplots drawn chosen from 3 random dimensions of the data. Indeed it does look like that the points compose a 2-dimensional plane or subspace from all angles of viewing.

(b & c) Write your own code of PCA to identify the principal components and the projections of the data set on to the 2-dim principal subspace. What is the proportion of total variance that is explained by PCA's two principal components?

```
# Writing a function for rough self-implementation
selfPCA = function(thedata){ # Input is a dataset
  scaled = apply(thedata, 2, scale) # Scaling
  covar = cov(scaled) # Covariance matrix
  eigens = eigen(covar) # Calculate Eigenvalues and eigenvectors
  loadings = eigens$vectors[,1:2]  # Get the loadings
  PC1 = as.matrix(scaled) %*% loadings[,1] # Component 1 Scores
  PC2 = as.matrix(scaled) %*% loadings[,2] # Component 2 Scores
```

```
  scores = cbind(PC1, PC2) # Put the columns together into a dataframe
  propvar = eigens$values / sum(eigens$values) # Proportion of variance explained by 2 PCs
  return(list(loadings, scores, propvar))
  # Returns a list containing the loadings, projections of the data, and proportion of variance
  # explained.
}
selfPCA(table)[3]
```

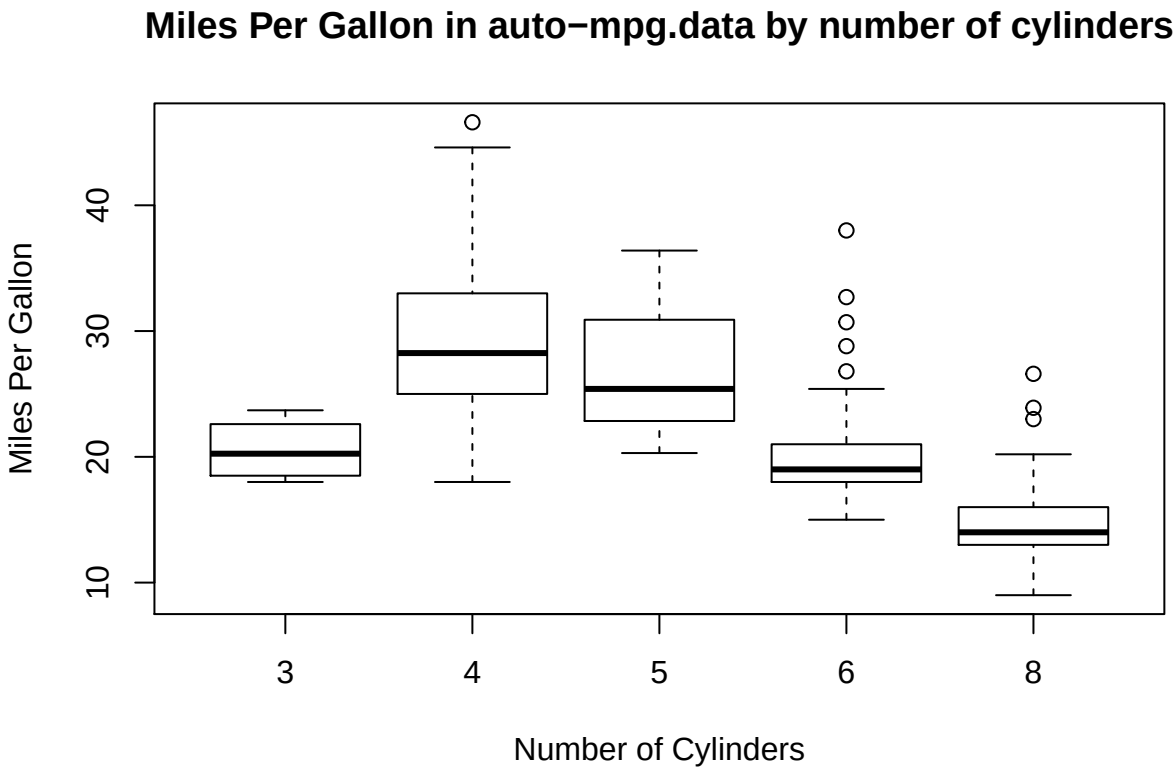
```
## [[1]]
## [1] 0.5141480274 0.4661556044 0.0078024624 0.0053865720 0.0035786742
## [6] 0.0022707467 0.0006579129
```

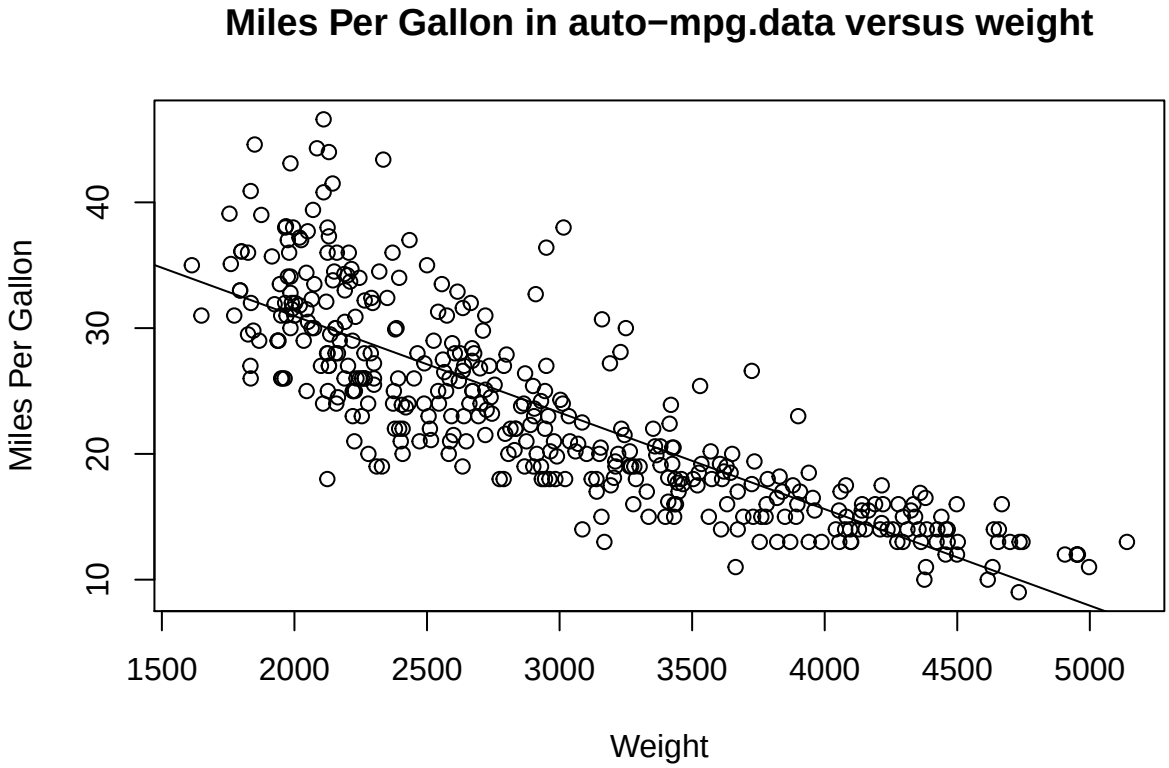
Remarks: According to the implementation (and numbers cross-checked by pre-existing methods of PCA analysis in R), the proportion of total variance that is explained by PCA's two principal components in total is about $0.5141 + 0.4662 = 0.9803$.

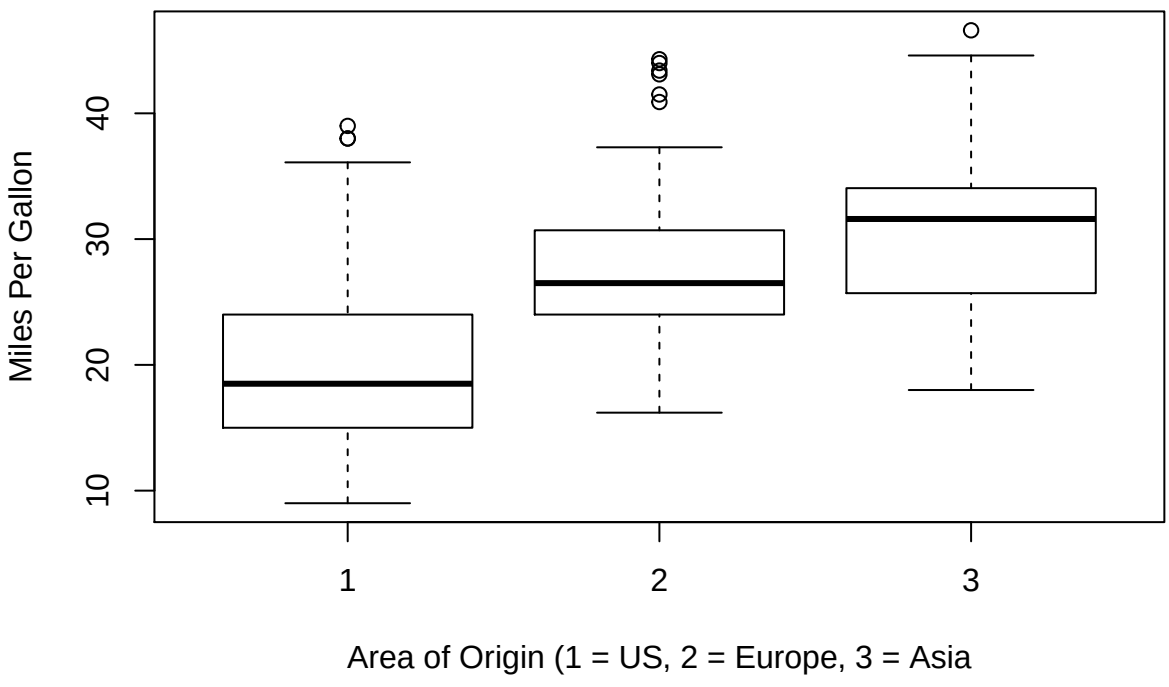
Problem 5: PCA

In this exercise you are welcome to use an existing PCA package. The data set (auto-mpg.data on Canvas) concerns city-cycle fuel consumption in miles per gallon (mpg) and other attributes collected for 398 vehicle instances. The variables are: mpg, cylinders, displacement, horsepower, weight, accleration, model year, origin and car name. Perform exploratory data analysis on this dataset including PCA and write a report summarizing your data analysis. In particular:

(a) Describe the data and present some initial pictorial and numerical summaries, such as scatterplots, histograms, etc.







Five Number Summary for Miles Per Gallon:

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    9.00   17.50   23.00   23.51   29.00   46.60
```

Remarks: The data of interest is a 398 x 9 dataset, with variables mpg, cylinders, displacement, horsepower, weight, acceleration, model year, origin, and car name. This dataset can be particularly good for looking at relationships between multiple variables, and we can analyze for patterns we may see. It is known that the categorical variables in this dataset are cylinders, model year, origin, and car name. There is an endless amount of exploratory data analysis that could be performed, but some personal analysis of interest is done above. For instance, it may be interesting to note the relationship between mpg and cylinders which looks like 4 cylinders is an optimal number for the most mpg with this dataset. Or we may be interested with which Area of Origin has the best mpg, perhaps the data suggesting Asia to produce cars with the generally best mpg values. Perhaps we may want to quantify the relationship between weight and mpg, which a quick regression and plot shows a expected negative relationship. Finally, a 5 number summary is displayed for minimum, Q1, mean, median, Q3, and maximum.

(b) Consider which variables should or should not be included in PCA on this dataset. Compare PCA on covariances and correlations (you may choose one of them to proceed with for the subsequent questions).

```
##                     Comp.1      Comp.2       Comp.3       Comp.4
## mpg           0.007595908  0.01744516  0.040534315  0.998983428
## displacement -0.114338397 -0.94599548  0.303145927  0.005037781
## horsepower   -0.038966145 -0.29827642 -0.948951620  0.043570470
## weight       -0.992647391  0.12085452  0.002748156  0.005341590
## acceleration  0.001352811  0.03483648  0.077089469 -0.008933812
##                     Comp.5
## mpg           0.005200813
## displacement  0.009821082
## horsepower    0.084292649
## weight       -0.003042445
## acceleration  0.996374423
```

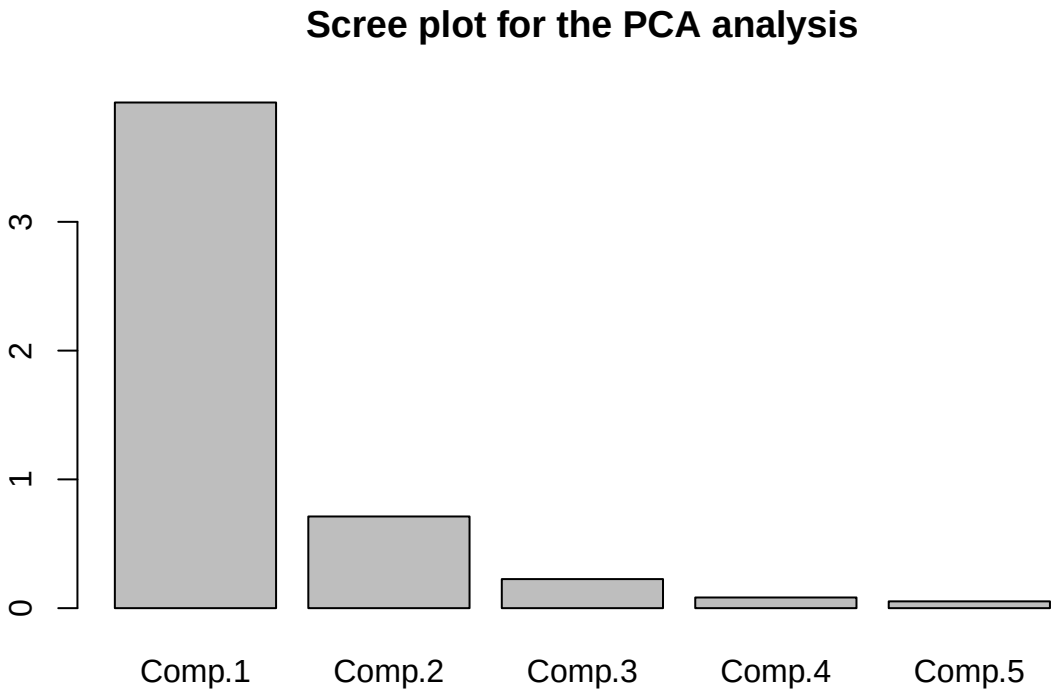
```
##                    Comp.1     Comp.2    Comp.3      Comp.4      Comp.5
## mpg           0.4442640 -0.3038692 0.8392164  0.01960783 -0.07499564
## displacement -0.4832332  0.1347900 0.3711101 -0.47619709  0.61954091
## horsepower   -0.4844417 -0.1242676 0.2064735  0.82560094  0.16007993
## weight       -0.4712207  0.3263218 0.3048759 -0.15942830 -0.74370590
## acceleration  0.3352350  0.8761089 0.1497078  0.25654937  0.17838348
```

Remarks: It is unadvised to use categorical variables in PCA analysis, since there is a significant challenge to find a appropriate manner to quantify distances between variable categories and the individuals in the PCA analysis space. So we should drop the categorical variables and end up using mpg, displacement, horsepower, weight, acceleration. We can use either covariances or correlations to construct the PCA analysis, but I will choose correlations to proceed with the PCA analysis. As of comparing the covariance loading matrix versus the correlation loading matrix, we can see that the correlation matrix has the data standardized. So we can get very different results if the scales of the variables are different. So covariance matrix is better to use if the scales of the variables are all similar, otherwise the correlation matrix is probably better. So we should use correlation matrix since for example the scale of weight and mpg are very very different.

(c) Comment on the percentage of variance explained and number of principal components to retain. Include a scree plot.

```
## Importance of components:
##                            Comp.1    Comp.2     Comp.3     Comp.4
## Standard deviation     1.9816040 0.8438043 0.47499662 0.28788096
## Proportion of Variance 0.7853509 0.1424011 0.04512436 0.01657509
## Cumulative Proportion  0.7853509 0.9277520 0.97287636 0.98945145
##                            Comp.5
## Standard deviation     0.22965790
## Proportion of Variance 0.01054855
```

```
## Cumulative Proportion  1.00000000
```



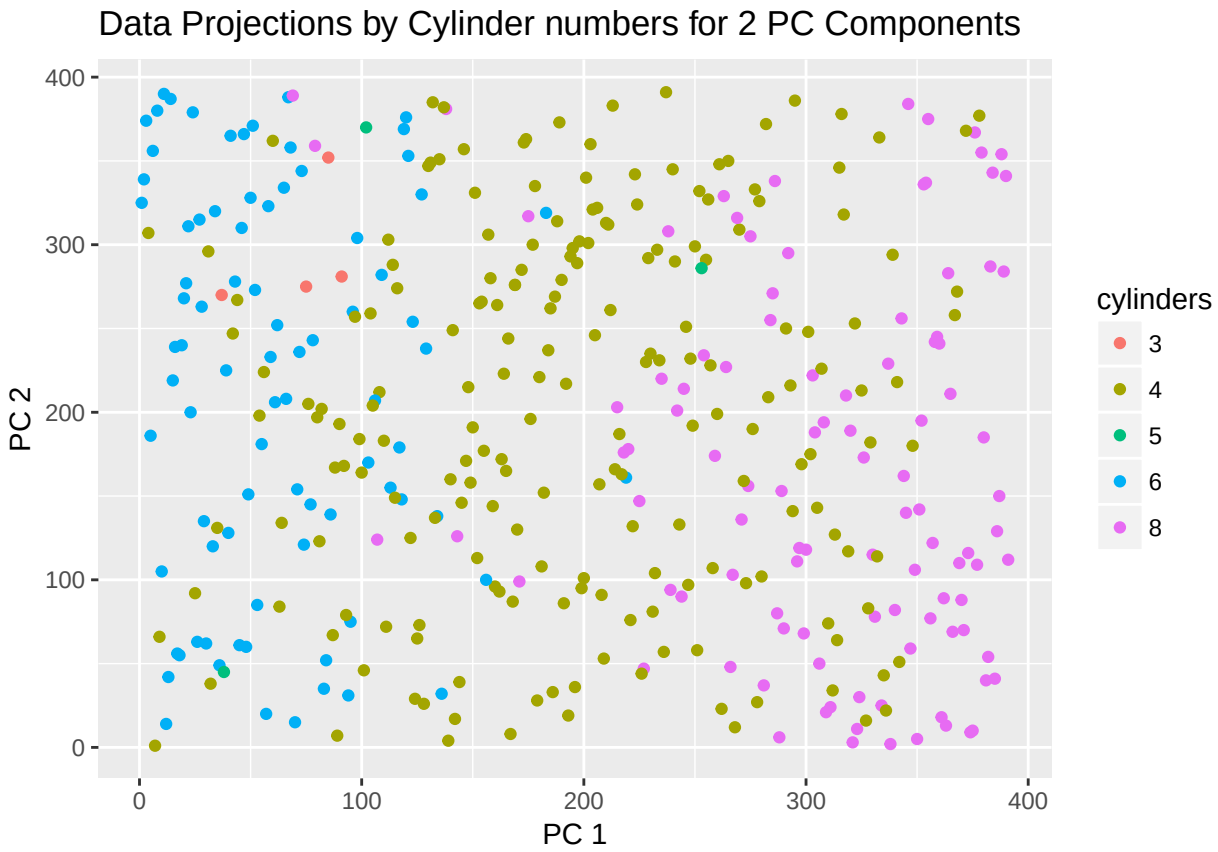
Remarks: The total amount of variance in the correlation matrix is calculated by adding the values on the diagonal. Thus the proportion of variance describes that with a particular amount of components how much of total amount of variance can be accounted for. The more amount of components included, the more variance that can be explained. The less amount of components included, the simpler amount of components that PCA can reduce to. Thus our optimal number of components can be determined by the smallest number of components that still captures the most amount of variance explained. In our case with this PCA analysis, it seems that an optimal number would be 2 or 3 components; an argument can be made if the increase of 5% variation explained by increasing components from 2 to 3 is significant enough. We can choose 3 components, and look at a scree plot generated to support the choice of 3 components.

(d) Comment on variable loadings and their potential interpretations.

```
##                 Comp.1     Comp.2    Comp.3      Comp.4      Comp.5
## mpg           0.4442640 -0.3038692 0.8392164  0.01960783 -0.07499564
## displacement -0.4832332  0.1347900 0.3711101 -0.47619709  0.61954091
## horsepower   -0.4844417 -0.1242676 0.2064735  0.82560094  0.16007993
## weight       -0.4712207  0.3263218 0.3048759 -0.15942830 -0.74370590
## acceleration  0.3352350  0.8761089 0.1497078  0.25654937  0.17838348
```

Remarks: In general with the variable loadings, we can analyze the variables through groups of number of components. Numbers can be compared to determine relative weight, and signs can be used to deduce variables that may move together with positive or negative correlation against each other. For instance in component 1, all the variables show to have similar weighting in the PCA but mpg and acceleration are correlated together and oppositely correlated to displacement horsepower and weight. In another example for component 2, acceleration has significant weight over the other variables. Similar interpretations can be determined for the other components.

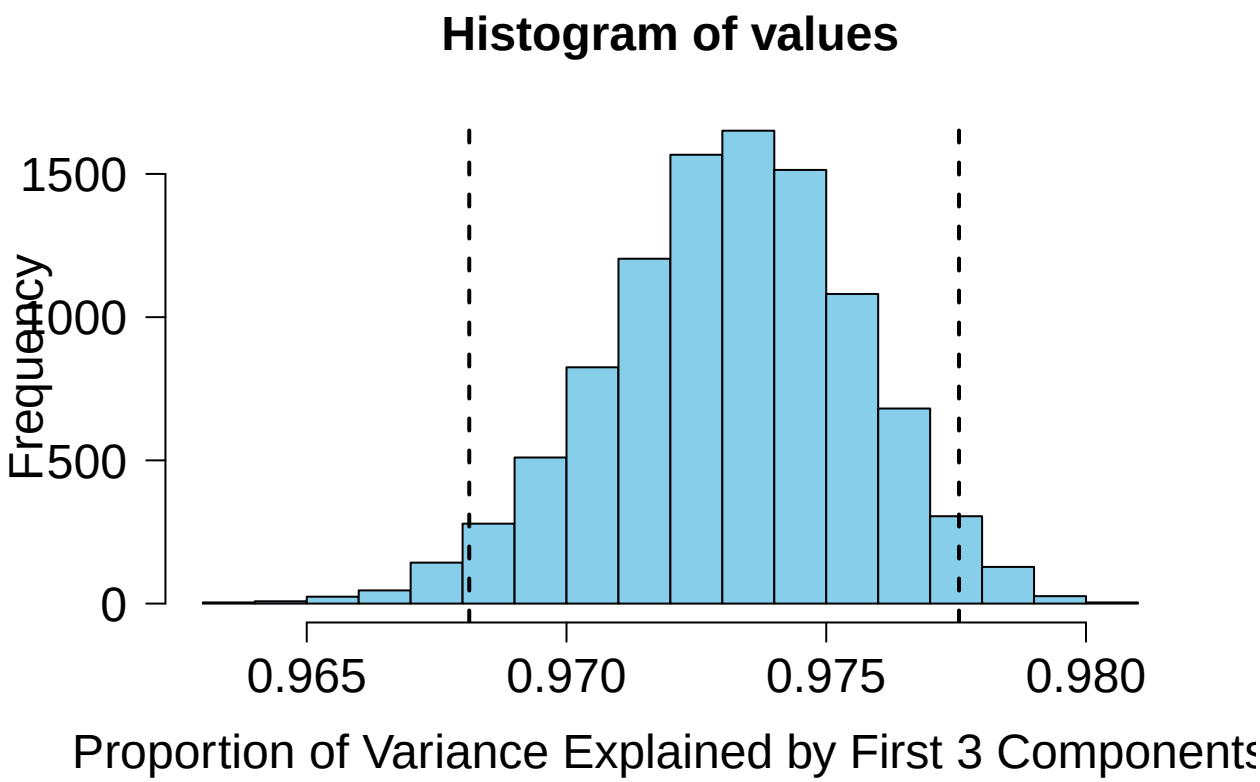
(e) Make a plot of the data projected on the first two PCs. Comment on any interesting features, including potential outliers, if any. By a visual display of PC scores, can you detect a categorical (discrete) attribute which is the most distinguishable (i.e., data are most separated according to the attribute values)?



Remarks: Though we can easily use data visualization techniques (ggplot2 in R) to make a graph of the data projected onto the first two PCs grouped by each and every categorical variable, it turns out that the most interesting categorical variable that shows distinguishment is cylinders. The plot is shown above, where we can see mostly a clear divisions of the data when we plot the data on the first two PCs grouped by cylinder numbers. There is not much data for 3 and 5 cylinders, and we can see that the division is not 100% perfect. For instance, we can see projections for 8 cylinders in the "region" that is typically for 6 cylinders. We can interpret these outlying points as outliers, which make sense since they fall far from the expected area of projection of the data points for 8 cylinders.

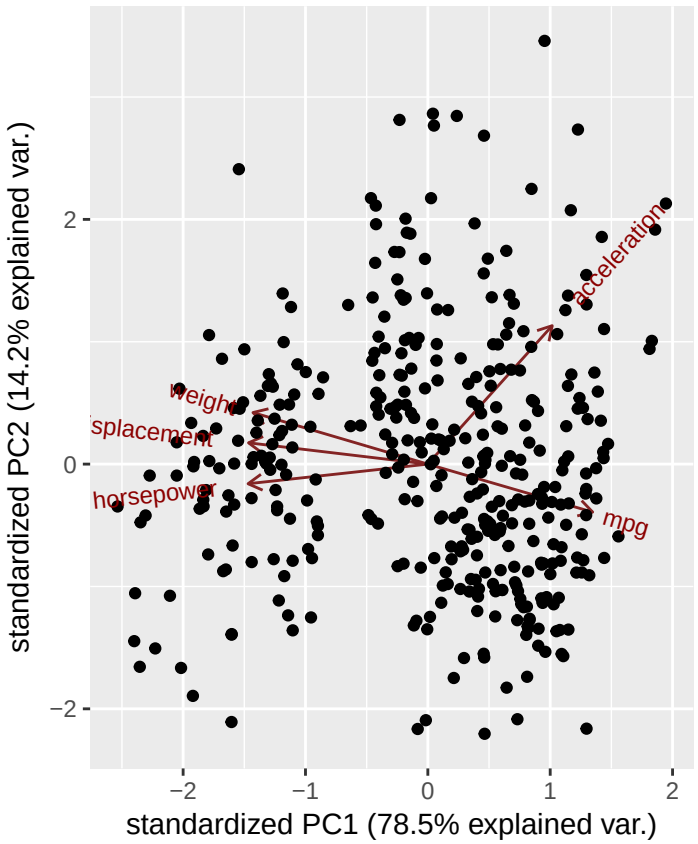
(f) Compute a bootstrap confidence interval for the percentage of variance explained by the first k PCs, where k is the number of PCs you recommend retaining for this dataset.

```
##      2.5%     97.5%
## 0.9681283 0.9775558
```



Remarks: We had recommended retaining 3 PCs for this dataset. Above is a histogram of the yielded proportion of variance explained after running a bootstrap on the data 10,000 times.

(g) Make a PCA biplot and comment on any interesting features



Remarks: From lecture notes, one excellent purpose of biplots is the ability to analyze the correlations between variables as the angles between variables is proportional to the correlation between the variables (given that the Principal Components do describe most of the variance). The variance is fairly decently explained, so we can point out that weight, displacement, and horsepower are all closely correlated with each other. Acceleration is not very correlated with any other variable. We can also analyze the pattern and direction of the arrows in respect to the PC axes. For instance, by the direction of the mpg arrow extending it seems that PC1 increases while PC2 decreases generally for mpg. Also weight and displacement and horsepower show that in PC1 they change at a greater rate over than in PC2 for these variables.

Appendix: R Code

```
# Using packages and other preliminaries
library(dplyr)
library(mixtools)
library(ggplot2)
library(devtools)
install_github("vqv/ggbiplot")
library(ggbiplot)
library(plotly)
library(scatterplot3d)
library(rgl)

# 3a
# Import Dataset and set column names
dataset = read.table("~/Desktop/School/Stats503/Datasets/heightWeightData.txt")
datasetmod = dataset %>%
  mutate(index = c(1:210))
colnames(datasetmod) = c("gender", "height", "weight", "index")

# Filter by male data
maledata = datasetmod %>%
  filter(gender == 1) %>%
  transmute(height, weight, index)

# Fitting 2-D Gaussian to data
mu <-c(mean(maledata$height), mean(maledata$weight))
sigma <- var(maledata[,1:2])
# Plotting Gaussian Ellipse on data points
plot(maledata[,1], maledata[,2], xlim = c(63, 82), ylim = c(75, 325), xlab = "Height of Males", ylab =
text(maledata[,1], maledata[,2], labels = maledata[,3], pos = 4)
ellipse(mu, sigma, npoints = 250, newplot = FALSE)

# 3b
# Standardizing male data first
dataset_stand = maledata %>%
  mutate(standheight = ((height - mean(height)) / sd(height)), standweight = ((weight - mean(weight)) /
  transmute(standheight, standweight, index)

# Fitting 2-D Gaussian to standardized data
mu2 <-c(mean(dataset_stand$standheight), mean(dataset_stand$standweight))
sigma2 <- var(dataset_stand[,1:2])
# Plotting Gaussian Ellipse on data points
plot(dataset_stand[,1], dataset_stand[,2], xlim = c(-3,3), ylim = c(-3,4), xlab = "Standardized Height
text(dataset_stand[,1], dataset_stand[,2], labels = dataset_stand[,3], pos = 4)
ellipse(mu2, sigma2, npoints = 250, newplot = FALSE)

# 3c
# Computing vectors, values, and matrixes needed to whiten the data
eigens = eigen(sigma2)
trans_eigenvalues = (eigens$values) ^ -0.5
lamb = matrix(c(trans_eigenvalues[1], 0, 0, trans_eigenvalues[2]), ncol = 2)
# ^ needs to be 2x2 for matrix multiplication to work out
```

```r
u = t(eigens$vectors)

target = matrix(0, nrow = 73, ncol = 2) # This matrix will hold the whitened data
for(i in 1:dim(dataset_stand)[1]){
  target[i,] = lamb %*% u %*% t(dataset_stand[i,1:2]) # Whiten the data from the standardized data
}

# Fitting 2-D Gaussian to whitened data
mu3 <-c(mean(target[,1]), mean(target[,2]))
sigma3 <- var(target[,1:2])
# ^ Verify here that the covariance matrix is proportional to identity matrix, which is true

# Plotting Gaussian Ellipse on data points
plot(target[,1], target[,2], xlim = c(-3,3), ylim = c(-3,3), xlab = "Whitened Height of Males", ylab = "
text(target[,1], target[,2], labels = dataset_stand[,3], pos = 4)
ellipse(mu3, sigma3, npoints = 250, newplot = FALSE)

# 4a
# Import Dataset
table = read.table("~/Desktop/School/Stats503/Datasets/fa_data.txt")

# A few Scatterplots in 3 dimensions
scatterplot3d(table[,3], table[,1], table[,2])
scatterplot3d(table[,4], table[,1], table[,5])
scatterplot3d(table[,2], table[,6], table[,1])
scatterplot3d(table[,7], table[,4], table[,3])

# 4b, c
# Writing a function for rough self-implementation
selfPCA = function(thedata){ # Input is a dataset
  scaled = apply(thedata, 2, scale) # Scaling
  covar = cov(scaled) # Covariance matrix
  eigens = eigen(covar) # Calculate Eigenvalues and eigenvectors
  loadings = eigens$vectors[,1:2]  # Get the loadings
  PC1 = as.matrix(scaled) %*% loadings[,1] # Component 1 Scores
  PC2 = as.matrix(scaled) %*% loadings[,2] # Component 2 Scores
  scores = cbind(PC1, PC2) # Put the columns together into a dataframe
  propvar = eigens$values / sum(eigens$values) # Proportion of variance explained by 2 PCs
  return(list(loadings, scores, propvar))
  # Returns a list containing the loadings, projections of the data, and proportion of variance
  # explained.
}
selfPCA(table)[3]

# 5a
# Import the dataset first
auto = read.table("~/Desktop/School/Stats503/Datasets/auto-mpg.data", stringsAsFactors = FALSE)
colnames(auto) = c("mpg", "cylinders", "displacement", "horsepower", "weight", "acceleration", "modelyea

# Some plots
boxplot(auto$mpg ~ auto$cylinders, main = "Miles Per Gallon in auto-mpg.data by number of cylinders", xl

plot(auto$weight, auto$mpg, main = "Miles Per Gallon in auto-mpg.data versus weight", xlab = "Weight", y
```

```r
reg1 <- lm(auto$mpg~auto$weight, data = auto)
abline(reg1)

boxplot(auto$mpg ~ auto$origin, main = "Miles Per Gallon in auto-mpg.data by Manufacturer Origin", xlab

# Useful 5 Number Summary
summary(auto$mpg)

# 5b
# Using non-categorical variables
PCAset = auto %>%
  transmute(mpg, displacement, horsepower, weight, acceleration)
# Coercions for component analysis to work properly
PCAset$horsepower <- as.numeric(as.character(PCAset$horsepower))
PCAset2 = na.omit(as.matrix(PCAset))
# PCA based on covariance matrix
PCAcov = princomp(PCAset2, cor = F)
loadings(PCAcov)[,1:5]
# PCA based on correlation matrix
PCAcorr = princomp(PCAset2, cor = T)
loadings(PCAcorr)[,1:5]

# 5c
summary(PCAcorr)
barplot(PCAcorr$sdev^2, main = "Scree plot for the PCA analysis")

# 5d
loadings(PCAcorr)[,1:5]

# 5e
# Establishing a new dataset with the PCA scores attached
newauto = auto
newauto$horsepower <- as.numeric(as.character(newauto$horsepower))
newauto = na.omit(as.matrix(newauto))
newdata = as.data.frame(cbind(newauto, PCAcorr$scores[,1], PCAcorr$scores[,2]))

# Displaying the PCA scores in first two PCs, colour-coded by number of cylinders
ggplot(newdata, aes(x = as.numeric(V10), y = as.numeric(V11), colour = cylinders)) +
  geom_point() + labs(title = "Data Projections by Cylinder numbers for 2 PC Components", x = "PC 1", y

# 5f
# Store the simulated values of variance explained by first 3 principal components
values = rep(0, 1000)
# Run the bootstrap for 10,000 bootstrap samples and store within the empty vector
for(i in 1:10000){
  resampled = PCAset2[sample(nrow(PCAset2),size=nrow(PCAset2),replace=TRUE),]
  PCAcorrelation = princomp(resampled, cor = T)
  variance_ex = PCAcorrelation$sdev^2 / sum(PCAcorrelation$sdev^2)
  values[i] = sum(variance_ex[1:3])
}
# Create a histogram displaying the variances and the 95% bootstrap confidence interval
hist(values, las=1, col='skyblue', xlab='Proportion of Variance Explained by First 3 Components',
     cex.axis=1.5, cex.lab=1.5, cex.main=1.5)
```

```r
boot_q = quantile(values, c(.025, .975))
boot_q # Display the CI
abline(v=boot_q, lty='dashed', lwd=2)

# 5g
ggbiplot(PCAcorr)
```