STA 135 Final Project

Julia Webb

Instructor: Xiaodong Li

913768646

March 2020

1 Data Set 1

Method: Multiple Linear Regression:

Introduction:

The goal of this analysis is to fit the below linear model to our dataset, which measures the characteristics of 76 young bulls sold at auction, along with their sale price. Additionally, we will see what proportion of the variance in sale price can be experienced by the predictor variables in our model. Finally, confidence intervals for our B_i parameters, the mean response of Y, and a new response Y_0 corresponding to $\bar{z}_1, ..., \bar{z}_8$ will be calculated, where z_i is a column of our data matrix, $\mathbf{Z}$, which is a 76 x 9 matrix, with the first column being $\mathbf{1}_{76}$.

$$SalePrice = \beta_0 + \beta_1 Breed_i + \beta_2 YrHgt_i + \beta_3 FtFrBody_i + \beta_4 PrctFFB_i + \\ \beta_5 Frame_i + \beta_6 BkFat_i + \beta_7 SaleHt_i + \beta_8 SaleWt_i + \epsilon$$

Summary:

The dataset includes which includes 78 observations of 8 variables. Below are some summary statistics, including the mean vector of the predictors, along with the mean response $\bar{y}$. The elements of $\bar{\vec{x}}$ are labeled with their corresponding variates.

$$\bar{y} = 1742.434 \quad , \quad \bar{\vec{x}} = \begin{bmatrix} Breed: & 4.3815789 \\ YrHgt: & 50.5223684 \\ FtFrBody: & 995.9473684 \\ PrctFFB: & 70.8815789 \\ Frame: & 6.3157895 \\ BkFat: & 0.1967105 \\ SaleHt: & 54.1263158 \\ SaleWt: & 1555.2894737 \end{bmatrix}$$

The variance of our response values is: $\hat{\sigma}^2 = 179560.2$

Analysis:

F-test :

To test whether there is a relationship between our predictors and response, we will perform a hypothesis test such that:

$H_o : \beta_1 = ... = \beta_8 = 0$

We will compare $\frac{1}{\hat{\sigma}^2}(\| \hat{\vec{\in}}_{red}\|^2 - \| \hat{\vec{\in}}\|^2)$ and (r-q)$F_{r-q,n-r-1}(\alpha)$ to come to a conclusion, where $\vec{\in}_{red}$ is the residual vector of the reduced model, and $\hat{\vec{\in}}$ is the residual vector of the full model. Note, here our reduced model would simply be $Y = \beta_0 + \mathcal{E}$

$\frac{1}{\hat{\sigma}^2}(\| \hat{\vec{\in}}_{red}\|^2 - \| \hat{\vec{\in}}\|^2) = 47.55923$

(r-q)$F_{r-q,n-r-1}(\alpha) = 16.63938$

Since our test statistic 47.55923 is larger than our critical value 16.63938, we have sufficient evidence to reject the null hypothesis at significance level $\alpha = 0.05$. In other words, we have sufficient evidence that there is a relationship between our response and our predictors.

Next, $\hat{\vec{\beta}}$ vector was found using the equation: $\hat{\vec{\beta}} = (\mathbf{Z}^T\mathbf{Z})^{-1}(\mathbf{Z}^T\hat{\vec{Y}})$. The results are:

$$\hat{\vec{\beta}} = \begin{bmatrix} -6186.7767966 \\ -83.1372876 \\ 98.6848806 \\ -1.7107980 \\ -33.3062451 \\ 284.4993376 \\ 1084.5221245 \\ 85.9002423 \\ 0.4582524 \end{bmatrix}$$

With these parameters, our model would be:

$$SalePrice = -6186.7767966 - 83.1372876 Breed_i + 98.6848806 YrHgt_i - 1.7107980 FtFrBody_i - 33.3062451 PrctFFB_i +$$
$$284.499337 Frame_i + 1084.5221245 BkFat_i + 85.9002423 SaleHt_i + 0.4582524 SaleWt_i + \epsilon_i$$

Next, the R^2 statistic was calculated, using the equation $R^2 = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Yi - \bar{Y})}$, where $\hat{\vec{Y}} = \mathbf{Z}\hat{\vec{\beta}}$. The result was:

$$R^2 = 0.586722$$

This means that 58.6722% of proportion of the variance for Sale Price, the dependent variable in our model, can be explained by the predictor variables in our model.

Next, $\hat{Cov}(\hat{\vec{\beta}})$ was calculated:

$\hat{Cov}(\hat{\vec{\beta}}) =$

$$\left[\begin{array}{ccccccccc}
14928688.53 & 18715.53 & -315100.06 & 904.57 & -31315.46 & 512138.70 & -863868.09 & -14813.51 & -21.07 \\
18715.53 & 513.54 & -449.40 & -3.03 & 19.82 & 468.32 & 9185.35 & -9.29 & -0.59 \\
-315100.06 & -449.40 & 10882.07 & 2.35 & -52.08 & -13776.07 & 4279.54 & -2919.45 & 8.24 \\
904.57 & -3.03 & 2.35 & 0.99 & -14.83 & -12.15 & -188.41 & -10.38 & -0.17 \\
-31315.46 & 19.82 & -52.08 & -14.83 & 604.02 & -7.92 & 7926.03 & 31.19 & 1.69 \\
512138.70 & 468.32 & -13776.07 & -12.15 & -7.92 & 25693.27 & -10488.45 & 646.64 & -0.48 \\
& & & & & & & &
\end{array}\right.$$

$$\left.\begin{array}{ccccccccc}
-863868.09 & 9185.35 & 4279.54 - 188.41 & 7926.03 & -10488.45 & 692803.17 & 7212.43 & -146.06 & \\
-14813.51 & -9.29 & -2919.45 & -10.38 & 31.19 & 646.64 & 7212.43 & 3471.75 & -14.75 \\
-21.07 & -0.59 & 8.24 & -0.17 & 1.69 & -0.48 & -146.06 & -14.75 & 0.31
\end{array}\right]$$

Confidence Intervals:

Various 95% confidence intervals were performed for our coefficients:
One-at-a-time confidence interval for 95% Confidence Interval for $\vec{\beta_j}$, j = 1,...,8. For each separate interval, we are 95% confident that the true β_j parameter lies in its respective interval:

β	**95%ConfidenceInterval**
β_0	[-13898.89 , 1525.333]
β_1	[-128.3696 , -37.90501]
β_2	[-109.5331 , 306.9029]
β_3	[-3.693612 , 0.2720161]
β_4	[-82.36177 , 15.74928]
β_5	[-35.44325 , 604.4419]
β_6	[-576.8514 , 2745.896]
β_7	[-31.70772 , 203.5082]
β_8	[-0.6594623 , 1.575967]

95% simultaneous confidence intervals for the β_i parameters based on the confidence region. We are 95% confident that our β_i's simultaneously lie in these respective intervals.

β	**95%ConfidenceInterval**
β_0	[-22672.899013064 , 10299.3454199282]
β_1	[-179.83001490305 , 13.5554396863028]
β_2	[-346.42121578957 , 543.790977050877]
β_3	[-5.94944577763334 , 2.52784984823653]
β_4	[-138.171889971099 , 71.5593996840302]
β_5	[-399.439695175811 , 968.438370288347]
β_6	[-2466.98441247478 , 4636.02866139796]
β_7	[-165.509483825443 , 337.309968429232]
β_8	[-1.93107841492842 , 2.8475832924531]

95% simultaneous confidence intervals for beta hats based on Bonferroni correction. Again, we are 95% confident that our β_i's simultaneously lie in these respective intervals:

β	**95%ConfidenceInterval**
β_0	[-17258.7907234853 , 4885.23713034946]
β_1	[-148.075737607674 , -18.1988376090736]
β_2	[-200.246598452362 , 397.616359713669]
β_3	[-4.55745693033161 , 1.13586100093481]
β_4	[-103.733593960844 , 37.121103673775]
β_5	[-174.831388986255 , 743.830064098791]
β_6	[-1300.65550082323 , 3469.69974974642]
β_7	[-82.9455270360922 , 254.746011639881]
β_8	[-1.14641263290665 , 2.06291751043133]

95% confidence intervals were also performed in regards to Sale Price, the response variable, were also calculated:

A 95% confidence interval for the mean response. We are 95% confident that the mean response, or Sale Price, is captured in the following interval:

[1645.41436030239 , 1839.45406074941]

95% prediction interval for a new response Y_0 corresponding to $(\bar{z_1}, ..., \bar{z_8})$, where $\bar{z}_i$ is the mean of the i^{th} column of $\mathbf{Z}$. We are 95% confident that Y_0 would fall in this interval. Note that this is different from the confidence interval for the mean response, because using the mean of each predictor in our model does not necessarily produce the mean response:

$$[\ 891.08847994482\ ,\ 2593.77994110698\ ]$$

Conclusion:
Based on the results from our analysis, there is a relationship between Sale Price of a young bull and the eight predictor variables. Additionally, we know the general affect each predictor has on the sale price by the magnitude and sign of its corresponding coefficient. For example, the BkFat variable has the highest affect on Sale Price, since its coefficient of 1084.52 has the highest magintidue out of the beta coefficients. A one unit increase in BkFat corresponds to a 1084.52 unit increase in Sale Price. Additionally, since our R^2 value was only 0.587, some additional transformations may be necessary to our variates in order to achieve a better model.

2 Data Set 2

Method: Two Sample Test and LDA

Introduction:

The goal of this analysis is to build an Linear Discriminant Analysis classifier for the Milk Transportation-Cost Dataset. The variates in this dataset are Fuel, Repair, and Capital. Furthermore, the dataset is classified by truck type: Gasoline Truck, and Diesel Truck. In other words, our goal is to use a LDA classifier to classify an observation as belonging to either the Gasoline Truck class, or the Diesel Truck class. Additionally, we will determine the accuracy of our classifier based on the performance of our dataset.
Upon visualizing the data (see Analysis section), it appeared that the two classes of trucks have a close to linear division in the two-dimensional space of Fuel and Capital (see plot below). There seemed to be less of a linear division between Repair and the other two variables. With this in mind, we will only consider Capital and Fuel when building our classifier, and disregard the Repair variable. This narrows down our goal to classify an observation as either belonging to the Gasoline or Diesel class, based on its values of Fuel and Capital.

Summary:

The dataset contains 59 observations. Below are some summary statistics:

Overall mean: $\hat{\vec{x}} = \begin{bmatrix} Fuel : 11.39492 \\ Capital : 12.93407 \end{bmatrix}$

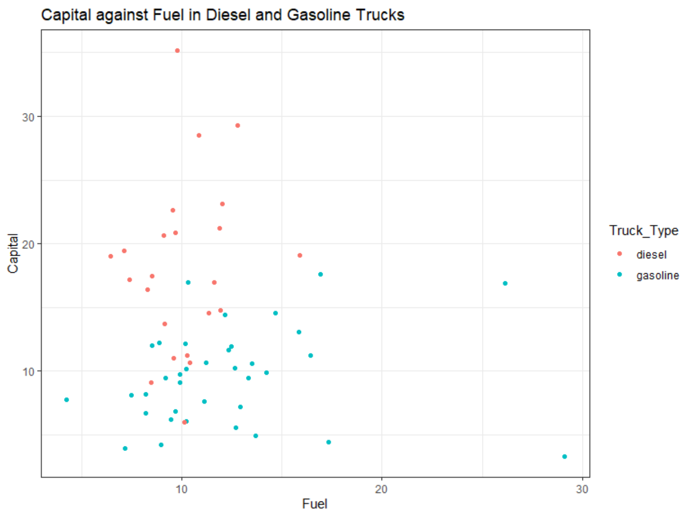
Mean per class: $\quad \hat{\vec{x}}_{gasoline} = \begin{bmatrix} Fuel : 12.218611 \\ Capital : 9.590278 \end{bmatrix} \quad \hat{\vec{x}}_{diesel} = \begin{bmatrix} Fuel : 10.10565 \\ Capital : 18.16783 \end{bmatrix}$

Class Sizes: $n_{gasoline} = 36 \quad n_{diesel} = 23$

Analysis:

Before proceeding with the analysis, we should note that an assumption of the LDA classifier is that the population covariances, $\Sigma_{gasoline}$ and Σ_{diesel}, are equal.

Plot before LDA split, only considering using Fuel and Capital predictors:



Next, a Hotelling's T^2 test was performed in R to test the null hypothesis $H_0 : \vec{\mu}_{gasoline} = \vec{\mu}_{diesel}$, at significance level $\alpha = 0.05$.

The results are: $T^2 = 46.88 \quad F_{2,56} = 0.051$

$T^2 > F_{2,56}$. This gives us sufficient evidence to reject $H_0 : \vec{\mu}_{gasoline} = \vec{\mu}_{diesel}$ at significance level $\alpha = 0.05$.

After conducting the test, Fisher's Rule for Classification can be used to classify a new observation $\vec{x}$:

$\vec{x}$ is assigned to Class 1: Gasoline trucks, if:

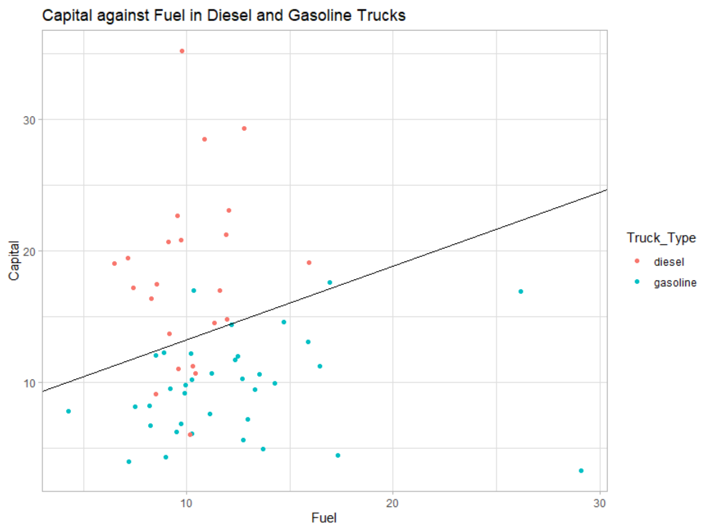
$$(\bar{\vec{x}}_{gasoline} - \bar{\vec{x}}_{diesel})^T \mathbf{S}^{-1}{}_{pooled}(\vec{x} - \tfrac{1}{2}(\bar{\vec{x}}_{gasoline} + \bar{\vec{x}}_{diesel})) = \begin{bmatrix} 0.1919475 & - & 0.3421672 \end{bmatrix} \left[\vec{x} - \begin{bmatrix} 22.32426 \\ 27.75810 \end{bmatrix} \right] \geq 0,$$

and Class 2: Diesel Trucks otherwise.

To visualize this division, we first project all data points onto a single line, so that our two samples are well-separated. This line is represented by $\hat{\vec{w}} = \mathbf{S}^{-1}(\bar{\vec{x}}_{gasoline} - \bar{\vec{x}}_{diesel})$:

$$\hat{\vec{w}} = \begin{bmatrix} 0.1919475 \\ -0.3421672 \end{bmatrix}$$

The line that divides the data in the plot below is the line at the midpoint of the vector $(\bar{\vec{x}}_{gasoline} - \bar{\vec{x}}_{diesel})$, and perpendicular to $\hat{\vec{w}}$. Notice that Fisher's Rule simplifies to the inner product of $\hat{\vec{w}}$ and the vector from the midpoint of class means to $\vec{x}$, the point we are classifying. Lets call the latter vector $\vec{v}$. Then, the inequality above is the inner product of $\hat{\vec{w}}$ and $\vec{v}$. This product equals the cosine of the angle between these two vectors, lets call it α. If $\cos(\alpha) \geq 0$, thus $\vec{v}$ is at most $90°$ from $\hat{\vec{w}}$, then $\bar{\vec{x}}$ is closer to Class 1: Gasoline Trucks. On the other hand, if $\cos(\alpha) < 0$ thus $\vec{v}$ is more than $90°$ from $\hat{\vec{w}}$, then $\bar{\vec{x}}$ is closer to Class 2: Diesel Trucks.



Our classifier seems to split the observations well. To further investigate the classifiers performance, the Apparent Error Rate and the Expected Actual Error Rate by Lachenbruch's Holdout were calculated using R, along with a Confusion Table:

Confusion Table:

	diesel	gasoline
diesel	18	5
gasoline	2	34

Apparent Error Rate: $\frac{\text{Missclassified Observations}}{\text{All Observations}} = \frac{7}{59} = 0.1186441$

Expected Actual Error Rate: $\frac{8}{59} = 0.1355932$

Conclusion:

In conclusion, the Milk Truck Transportation-Cost dataset can be classified with a Linear Discriminant Analysis Classifier, using Fuel and Capital as our variables. The Expected Actaul Error Rate is 0.1356, and the Apparent Error Rate is 0.1186441. As expected, the Apparent Error Rate underestimates the Expected Error Rate.

This classifier gives us the ability to predict if a new observation will correspond with a Gasoline Truck or Fuel Truck, based on its Fuel and Capital values.

3 Dataset 3

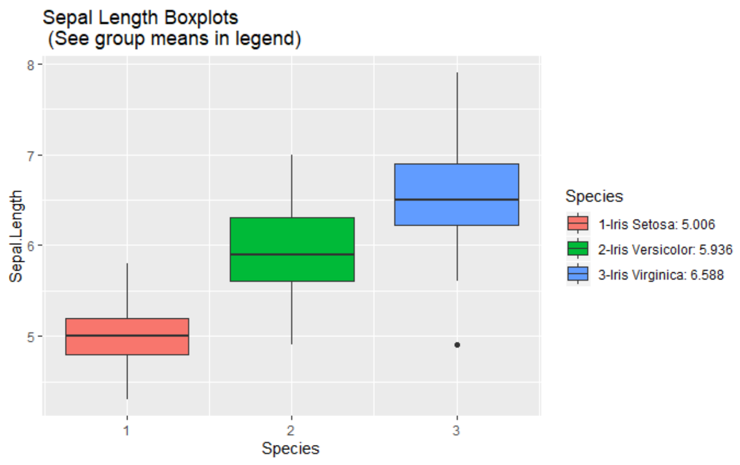
Method: PCA

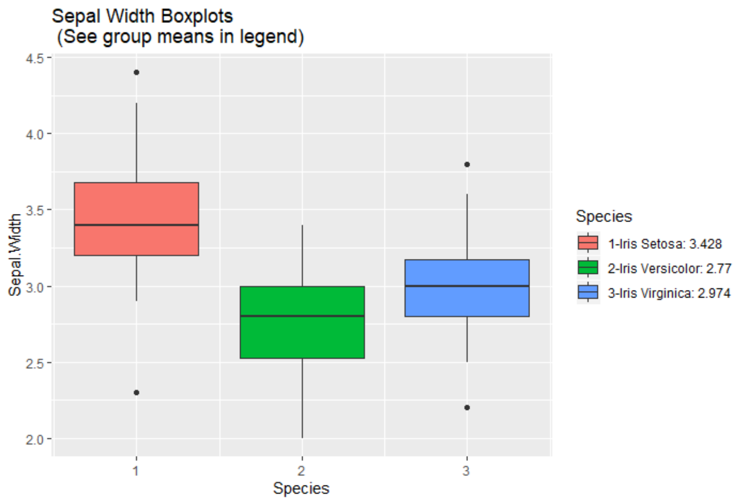
Introduction:

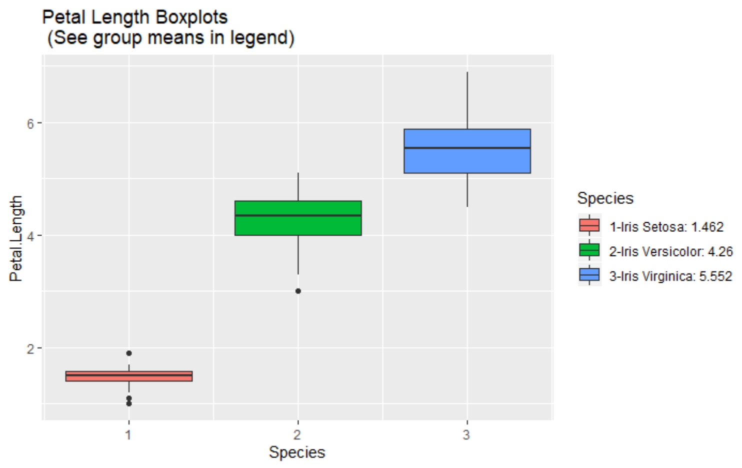
The goal of the analysis is to reduce the dimension of the Iris data set from table 11.5 of the textbook. The dataset contains 150 observations of four continuous variables: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width. The observations are classified by three Species: Iris setosa, Iris versicolor, Iris virginica. The species are included as a coded factor variable in the dataset. We will reduce the dimensions of the Iris dataset using Principal Component Analysis.

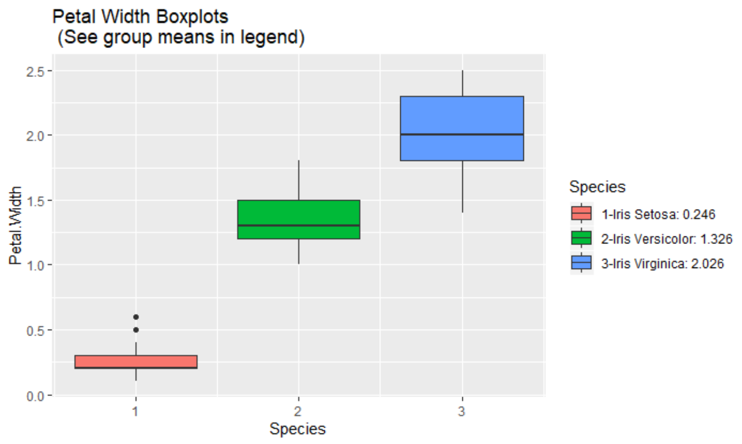
Summary:

Below are boxplots of our variables sorted by Species, with each group mean labeled in the legend.









Analysis:

A principal component object was created in R using the princomp() function. The below table is the summary of this object. Note that our principal components were calculated using R, the correlation matrix of our variables. It is good practice to use R instead of S, the covariance matrix. This is because if a variable has a high variance its loadings will be disproportionally large and its contribution for predicting the corresponding principal components will be misrepresented.

Importance of components:	Comp.1	Comp.2	Comp.3	Comp.4
Standard deviation	1.7083611	0.9560494	00.38308860	0.143926497
Proportion of Variance	0.7296245	0.2285076	0.03668922	0.005178709
Cumulative Proportion	0.7296245	0.9581321	0.99482129	1.000000000

Loadings:	Comp.1	Comp.2	Comp.3	Comp.4
Sepal.Length	0.521	0.377	0.720	0.261
Sepal.Width	-0.269	0.923	-0.244	-0.124
Petal.Length	0.580		-0.142	-0.801
Petal.Width	0.565		-0.634	0.524

Interpretation of the table:

Let us first consider the "Importance of components" section. The first row gives the standard deviation of each component, which is the square root of the corresponding eigenvalue of the correlation matrix R. The Proportion of Total Variance row provides the proportion of total variance due to each specific principal component. The Cumulative Proportion row gives the cumulative proportion of this, and is useful in determining how many principal components one could use. We see that the cumulative proportion of variance is approximately .96 at component 2, so we can replace our four variables with $\hat{Y}_1$ and $\hat{Y}_2$ without much loss of information. (Note, that $\hat{Y}_1$ and $\hat{Y}_2$ are the first and second principal components).

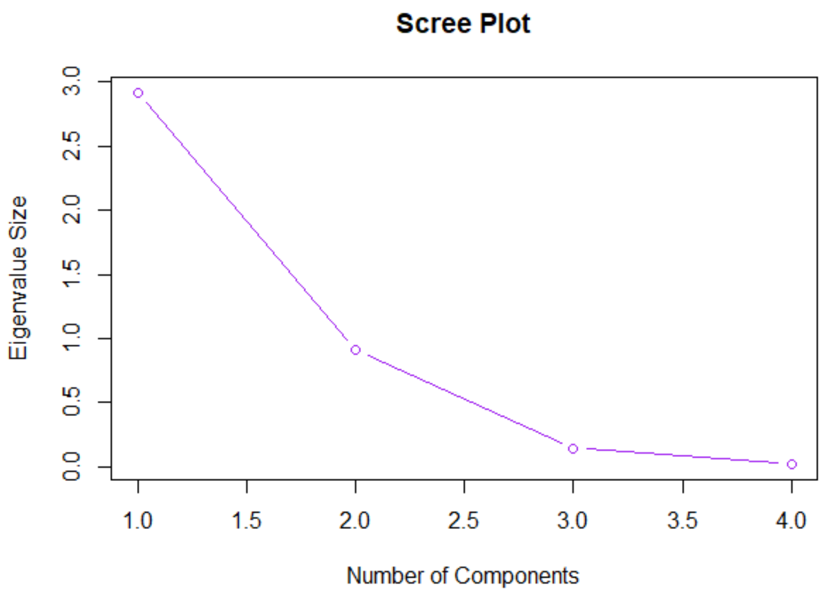
Next, lets consider the "Loadings" section. The loadings of the i^{th} principal component are elements of the i^{th} unit eigenvector of R. Additionally, they are the coefficients for our original variables in the expression for the i^{th} eigenvalue. In the expression for a given principal component, the magnitude of each variable's loading indicates its influence on the principal component. We see that the Petal.Length variable has the highest influence on the first principal component, and Sepal.Width was the highest influence on the second principal component. Note that R left out loadings with an absolute value less than 0.1, as this is an indicator a variable has little effect on the respective principal component.

Below is the variance of each principal Component, which is equal to the eigenvalues of the correlation matrix. These are the y-values in the upcoming Scree Plot.

$\boldsymbol{\lambda_1} = Var(\hat{Y}_1)$	$\boldsymbol{\lambda_2} = Var(\hat{Y}_2)$	$\boldsymbol{\lambda_3} = Var(\hat{Y}_3)$	$\boldsymbol{\lambda_4} = Var(\hat{Y}_4)$
2.91849782	0.91403047	0.14675688	0.02071484

Scree Plot:

The scree plot displays eigenvalues against their corresponding principal component index. This is a useful visualization of how well each component captures variation in your data. The ”elbow” of this plot is the point where the size of the eigenvalues begin to plateu, which is an indicator that the subsequent principal components may not be that significant in summarizing our data.
The elbow seems to occur at the second principal component. This makes sense, since the cumulative proportion of variance at the second principle component is 0.9502309 as discussed previously. With this in mind, it seems reasonable to use two principal components to describe our data set.



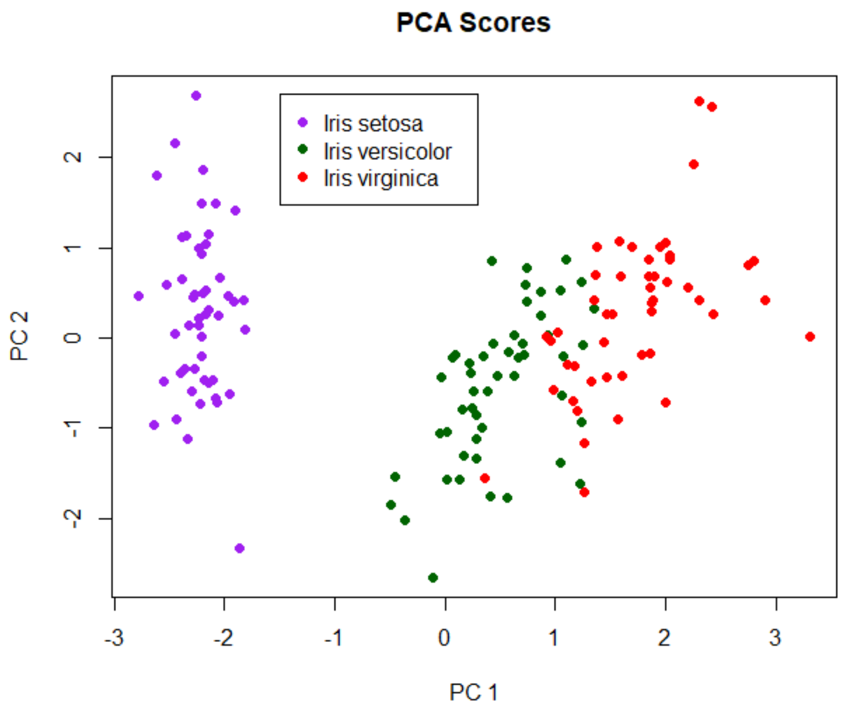
Expressions for the $\hat{Y}_1$ and $\hat{Y}_2$:

$$\hat{Y}_1 = 0.521 Sepal.Length - 0.269 Sepal.Width + 0.580 Petal.Length + 0.565 Petal.Width$$

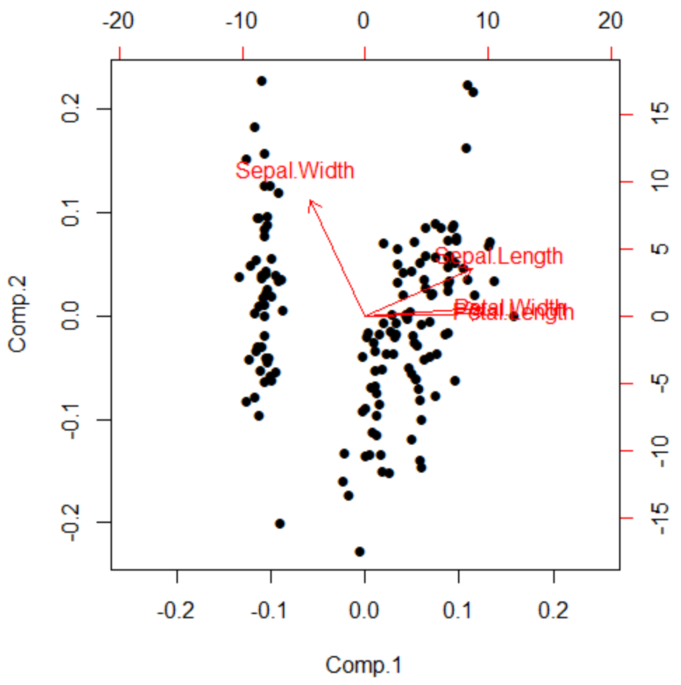
$$\hat{Y}_2 = 0.377 Sepal.Length + 0.923 Sepal.Width$$

Plotting PCA Scores:

Next is a scatter plot of the second principal component, $\hat{Y}_2$, against the first principal component, $\hat{Y}_1$. It seems that there are similarities in PCA scores with each class, which implies that their values for our original variables are similar.



Finally, lets consider the Biplot below.



First, consider the points on the biplot, each point represents an observation in the dataset. The x and y coordinates of each point correspond to its normalized PCA score for the first and second components, respectively.

Now, consider the vectors corresponding to each original variable. The right axis corresponds to a variables loading in the first principal component, and the top axis corresponds to that variables loading in the second principal component. Therefore, if a variable's corresponding vector has a large projection on a given axis, then its has a large loading in the corresponding principal component. In our plot, we see that Petal Width, Petal Length, and Sepal Length have a higher influence on the first principal component, while Sepal Width has a larger influence on the second.

Conclusion:
In conclusion, we have found that we can summarize the Iris data well with only two principal components. The first component is influenced the most by Petal Length, whose loading is 0.58, and the second principle component is influenced the most by Sepal Width, whose loading is 0.923. Plotting our data by their principal component scores allowed us to visualize similarities within species, which would not have been available to us without PCA analysis.

Code Appendix

Bulls: Multiple Linear Regression

Julia Webb

March 9, 2020

We aim at fitting the linear model Yi = ??0+??1zi1+??2zi2+□i, i= 1,2,…,7.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
setwd("C:/Users/jwebb/OneDrive/Desktop/STA 135/project")
bulls <- read.table('bulls.dat') #table 1.10 in book, pdf pg 33
colnames(bulls) <- c( "Breed",  "SalePr",   "YrHgt",  "FtFrBody",  "PrctFFB", "Frame",  "BkFat",
"SaleHt",  "SaleWt")
#View(bulls)
```

```
#matrix of predictors
bulls.predictors <- bulls[,-2] %>% as.matrix(76,8)

#Z...predictor matrix with column of ones in front
Z <- cbind(rep(1,76), bulls.predictors)

#Y matrix
Y <-   as.matrix(bulls$SalePr,76,1)  #sale price

#number of observations
n <- length(Y)

#number of predictors
r <- dim(Z)[2]-1

#head(Z)
#head(bulls.sale_price)
```

```
#summary stats
apply(bulls.predictors, 2, mean)
```

```
##        Breed        YrHgt     FtFrBody      PrctFFB        Frame        BkFat 
##    4.3815789   50.5223684  995.9473684   70.8815789    6.3157895    0.1967105 
##       SaleHt       SaleWt 
##   54.1263158 1555.2894737
```

(1) Find the least square estimate^~??;

```
# least square estimates
beta_hat <- solve(t(Z)%*%Z)%*%t(Z)%*%Y
colnames(beta_hat) <- NULL
beta_hat
```

```
##                   [,1]
##          -6186.7767966
## Breed      -83.1372876
## YrHgt       98.6848806
## FtFrBody    -1.7107980
## PrctFFB    -33.3062451
## Frame      284.4993376
## BkFat     1084.5221245
## SaleHt      85.9002423
## SaleWt       0.4582524
```

(2) Find theR2statistic;

```
# R^2 statistic
R_square <- 1 - sum((Y - Z%*%beta_hat)^2)/sum((Y-mean(Y))^2)
R_square
```

```
## [1] 0.586722
```

(3) Find ^??2and ^Cov??

```
# sigma_hat_square
sigma_hat_square <- sum((Y - Z%*%beta_hat)^2)/(n-r-1)
sigma_hat_square
```

```
## [1] 179560.2
```

```
# estimated covariance of hat{beta}
a <- (sigma_hat_square * solve(t(Z)%*%Z))
colnames(a) <- NULL
rownames(a) <- NULL
round(a,2)
```

```
##             [,1]     [,2]       [,3]    [,4]      [,5]      [,6]       [,7]
##  [1,] 14928688.53 18715.53 -315100.06  904.57 -31315.46 512138.70 -863868.09
##  [2,]    18715.53   513.54    -449.40   -3.03     19.82    468.32    9185.35
##  [3,]  -315100.06  -449.40   10882.07    2.35    -52.08 -13776.07    4279.54
##  [4,]      904.57    -3.03       2.35    0.99    -14.83    -12.15    -188.41
##  [5,]   -31315.46    19.82     -52.08  -14.83    604.02     -7.92    7926.03
##  [6,]   512138.70   468.32  -13776.07  -12.15     -7.92  25693.27  -10488.45
##  [7,]  -863868.09  9185.35    4279.54 -188.41   7926.03 -10488.45  692803.17
##  [8,]   -14813.51    -9.29   -2919.45  -10.38     31.19    646.64    7212.43
##  [9,]      -21.07    -0.59       8.24   -0.17      1.69     -0.48    -146.06
##            [,8]    [,9]
##  [1,] -14813.51  -21.07
##  [2,]     -9.29   -0.59
##  [3,]  -2919.45    8.24
##  [4,]    -10.38   -0.17
##  [5,]     31.19    1.69
##  [6,]    646.64   -0.48
##  [7,]   7212.43 -146.06
##  [8,]   3471.75  -14.75
##  [9,]    -14.75    0.31
```

(4) Find a 95% confidence interval for??1;

t-test for single coefficient H_0: beta_j = 0, H_a: beta_j != 0 ….seems unnecessary !

```
j <- 1
t_stat <- (beta_hat[j+1] - 0)/sqrt(sigma_hat_square * solve(t(Z)%*%Z)[j+1,j+1])
t_stat
```

```
## [1] -3.668679
```

```
alpha <- 0.05
cval_t <- qt(1-alpha/2, n-r-1)
cval_t
```

```
## [1] 1.996008
```

```
# One-at-a-time confidence interval for beta_j
for (i in c(0:8)){
j <- i
cat('[',
    beta_hat[j+1] - qt(1-alpha/2, n-r-1)*sqrt(sigma_hat_square * solve(t(Z)%*%Z)[j+1,j+1]),
    ',',
    beta_hat[j+1] + qt(1-alpha/2, n-r-1)*sqrt(sigma_hat_square * solve(t(Z)%*%Z)[j+1,j+1]),
    ']')
}
```

```
## [ -13898.89 , 1525.333 ][ -128.3696 , -37.90501 ][ -109.5331 , 306.9029 ][ -3.693612 , 0.2720
161 ][ -82.36177 , 15.74928 ][ -35.44325 , 604.4419 ][ -576.8514 , 2745.896 ][ -31.70772 , 203.5
082 ][ -0.6594623 , 1.575967 ]
```

(5) Find 95% simultaneous confidence intervals for each B based on the confidence region;

```
# confidence region based simultaneous confidence intervals
#this <- c()


all_ci_confidence_region <- c()
for (j in c(0:r)){
  ci <- paste("[", beta_hat[j+1] - sqrt((r+1)*qf(1-alpha, r+1, n-r-1))*sqrt(sigma_hat_square * s
olve(t(Z)%*%Z)[j+1,j+1]), "  ,  ",
    beta_hat[j+1] + sqrt((r+1)*qf(1-alpha, r+1, n-r-1))*sqrt(sigma_hat_square * solve(t(Z)%*%Z)
[j+1,j+1]), "]")

  all_ci_confidence_region <- c(all_ci_confidence_region, ci)

}
```

```
all_ci_confidence_region
```

```
## [1] "[ -22672.899013064   ,   10299.3454199282 ]"
## [2] "[ -179.83001490305   ,   13.5554396863028 ]"
## [3] "[ -346.42121578957   ,   543.790977050877 ]"
## [4] "[ -5.94944577763334   ,   2.52784984823653 ]"
## [5] "[ -138.171889971099   ,   71.5593996840302 ]"
## [6] "[ -399.439695175811   ,   968.438370288347 ]"
## [7] "[ -2466.98441247478   ,   4636.02866139796 ]"
## [8] "[ -165.509483825443   ,   337.309968429232 ]"
## [9] "[ -1.93107841492842   ,   2.8475832924531 ]"
```

```
library(stringr)
library(dplyr)

betas <- c("??0", "??1","??2", "??3", "??4", "??5", "??6", "??7", "??8" ) #??^0
betas <- str_replace(betas, "[?]", "B") %>% str_replace("[?]","")
tibble("Estimation for" = betas,  "95% Simultaneous Confidence Interval based on Confidence Regi
on" = all_ci_confidence_region)
```

```
## # A tibble: 9 x 2
##   `Estimation for` `95% Simultaneous Confidence Interval based on Confidence Re~
##   <chr>            <chr>
## 1 B0               [ -22672.899013064   ,   10299.3454199282 ]
## 2 B1               [ -179.83001490305   ,   13.5554396863028 ]
## 3 B2               [ -346.42121578957   ,   543.790977050877 ]
## 4 B3               [ -5.94944577763334   ,   2.52784984823653 ]
## 5 B4               [ -138.171889971099   ,   71.5593996840302 ]
## 6 B5               [ -399.439695175811   ,   968.438370288347 ]
## 7 B6               [ -2466.98441247478   ,   4636.02866139796 ]
## 8 B7               [ -165.509483825443   ,   337.309968429232 ]
## 9 B8               [ -1.93107841492842   ,   2.8475832924531 ]
```

(6) Find 95% simultaneous confidence intervals for beta hats based on Bonferroni correction;

```
# Bonferroni correction based simultaneous confidence intervals

all_ci_bonferroni <- c()
for (j in c(0:r)){
ci <- paste('[',
    beta_hat[j+1] - qt(1-alpha/(2*(r+1)), n-r-1)*sqrt(sigma_hat_square * solve(t(Z)%*%Z)[j+1,j+1
]),
    '   ,   ',
    beta_hat[j+1] + qt(1-alpha/(2*(r+1)), n-r-1)*sqrt(sigma_hat_square * solve(t(Z)%*%Z)[j+1,j+1
]),
    ']')

all_ci_bonferroni <- c(all_ci_bonferroni, ci)
}
```

`

```
tibble("Estimation for" = betas,  "95% Simultaneous Confidence Interval based on Bonferroni Corr
ection" = all_ci_bonferroni)
```

```
## # A tibble: 9 x 2
##   `Estimation for` `95% Simultaneous Confidence Interval based on Bonferroni Co~
##   <chr>            <chr>
## 1 B0               [ -17258.7907234853    ,    4885.23713034946 ]
## 2 B1               [ -148.075737607674    ,    -18.1988376090736 ]
## 3 B2               [ -200.246598452362    ,    397.616359713669 ]
## 4 B3               [ -4.55745693033161    ,    1.13586100093481 ]
## 5 B4               [ -103.733593960844    ,    37.121103673775 ]
## 6 B5               [ -174.831388986255    ,    743.830064098791 ]
## 7 B6               [ -1300.65550082323    ,    3469.69974974642 ]
## 8 B7               [ -82.9455270360922    ,    254.746011639881 ]
## 9 B8               [ -1.14641263290665    ,    2.06291751043133 ]
```

(7) TestH0:??1=??2=…..=??8 0 at the level of??= 0.05;

```
# F-test
# H_0: beta_1 = beta_2 = 0
attach(data.frame(bulls.predictors))
anova(aov(Y~Breed + YrHgt + FtFrBody + PrctFFB + Frame + BkFat + SaleHt + SaleWt))
```

```
## Analysis of Variance Table
##
## Response: Y
##           Df   Sum Sq  Mean Sq F value   Pr(>F)
## Breed      1  1464492  1464492  8.1560 0.005709 **
## YrHgt      1 11754421 11754421 65.4623 1.65e-11 ***
## FtFrBody   1   606563   606563  3.3780 0.070504 .
## PrctFFB    1  1345337  1345337  7.4924 0.007927 **
## Frame      1   567908   567908  3.1628 0.079872 .
## BkFat      1   474365   474365  2.6418 0.108780
## SaleHt     1   746156   746156  4.1555 0.045450 *
## SaleWt     1   120249   120249  0.6697 0.416062
## Residuals 67 12030533   179560
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(equivalent) F-test by comparing residuals

Since F stat reduced = (1/sigma hat sqrd)(|e.red|^2 - |e|^2) > critcal val = (r-q)(F(r-q)(n-r-1)(alpha)), we reject Ho that B1….B8 = 0 at signifance level alpha = 0.05

```
# fit the reduced model
beta_hat_reduced <- solve(t(Z[,1])%*%Z[,1])%*%t(Z[,1])%*%Y
colnames(beta_hat_reduced) <- NULL

f_stat_reduced <- ((sum((Y - Z[,1]%*%beta_hat_reduced)^2) - sum((Y - Z%*%beta_hat)^2))/2)/sigma_
hat_square # will compare this with
f_stat_reduced
```

```
## [1] 47.55923
```

```
#make critical value
q <- 0
critical_value <- (r-q)*qf(.95,r-q,n-r-1)
critical_value
```

```
## [1] 16.63938
```

(8) Find a 95% confidence interval for the mean responseE(Y0) =??0+??1 ??z1+??2 ??z where z is the sample mean vector

```
# confidence interval for z_0^T beta

z_0 <- apply(Z,2,mean)

ci_mean_response <-
paste('[',
    z_0%*%beta_hat - sqrt(sigma_hat_square)*sqrt(t(z_0)%*%solve(t(Z)%*%Z)%*%z_0)*qt(1-alpha/2, n
-r-1),
    ',',
    z_0%*%beta_hat + sqrt(sigma_hat_square)*sqrt(t(z_0)%*%solve(t(Z)%*%Z)%*%z_0)*qt(1-alpha/2, n
-r-1),
    ']')

ci_mean_response
```

```
## [1] "[ 1645.41436030239 , 1839.45406074941 ]"
```

(9) Find a 95% prediction interval for a new responseY0corresponding to (??z1, ??z2,… z8)

```
# prediction interval for Y_0 = z_0^T beta + epsilon_0

prediction_interval_y <-
paste('[',
    z_0%*%beta_hat - sqrt(sigma_hat_square)*sqrt(1+t(z_0)%*%solve(t(Z)%*%Z)%*%z_0)*qt(1-alpha/2,
n-r-1),
    ',',
    z_0%*%beta_hat + sqrt(sigma_hat_square)*sqrt(1+t(z_0)%*%solve(t(Z)%*%Z)%*%z_0)*qt(1-alpha/2,
n-r-1),
    ']')

prediction_interval_y
```

```
## [1] "[ 891.08847994482 , 2593.77994110698 ]"
```

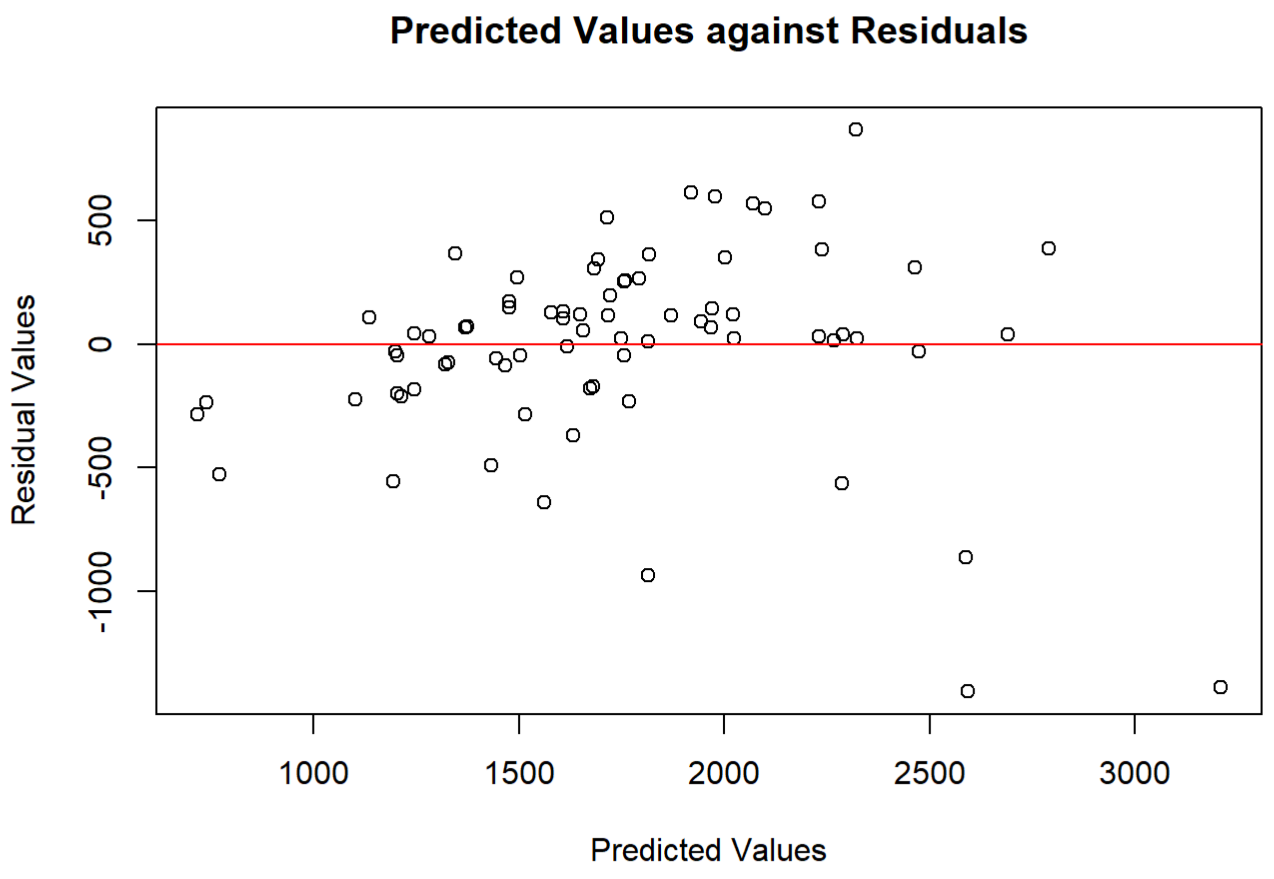
```
beta_hat
```

```
##                    [,1]
##          -6186.7767966
## Breed      -83.1372876
## YrHgt       98.6848806
## FtFrBody    -1.7107980
## PrctFFB    -33.3062451
## Frame      284.4993376
## BkFat     1084.5221245
## SaleHt      85.9002423
## SaleWt       0.4582524
```

lets look at our residuals.

```
beta_hat_no_labels <- beta_hat
rownames(beta_hat_no_labels) <- NULL

Z_no_labels <- Z
colnames(Z_no_labels) <- NULL
```

```
Y.pred <- Z_no_labels%*%beta_hat_no_labels
Res <- Y.pred - Y
plot(x= Y.pred, y= Res, xlab = "Predicted Values", ylab = "Residual Values",main = "Predicted Va
lues against Residuals")
abline(h=0,col="Red")
```



```
beta_hat
```

```
##                   [,1]
##          -6186.7767966
## Breed      -83.1372876
## YrHgt       98.6848806
## FtFrBody    -1.7107980
## PrctFFB    -33.3062451
## Frame      284.4993376
## BkFat     1084.5221245
## SaleHt      85.9002423
## SaleWt       0.4582524
```

So we know that there is some relationship between Y and our predictors, lets try increasing our reduced model.

results were not interesting; decided to not include in write-up

It looks like SaleWt and FtFrBody are not that impactful. let test a reduced reduced model where those are removed and q=7

```
b_reordered <- bulls[,-c(2,9,4)]
b_reordered <- cbind(b_reordered, "FtFrBody" = bulls$FtFrBody, "SaleWt"=bulls$SaleWt) #so 8,9 no
t included in reduced
Z.new <- as.matrix(cbind("One" = rep(1,length(b_reordered[,1])), b_reordered))
```

```
# fit the reduced model
beta_hat_reduced <- solve(t(Z.new[,1:7])%*%Z.new[,1:7])%*%t(Z.new[,1:7])%*%Y
colnames(beta_hat_reduced) <- NULL

f_stat_reduced <- ((sum((Y - Z.new[,1:7]%*%beta_hat_reduced)^2) - sum((Y - Z.new%*%beta_hat)^2
))/2)/sigma_hat_square # will compare this with
f_stat_reduced
```

```
## [1] -4249168
```

```
#make critical value
q <- 7
critical_value <- (r-q)*qf(.95,r-q,n-r-1)
critical_value
```

```
## [1] 3.984049
```

Again, we see that our f-statistic is greater than our critical value, so we reject the Null that B8= B9=0 (ie the betas that correspond with FtFrBody and SaleWt in our reduced model)

we can cross check our calculations of the beta hats with R’s lm() function, we see that the coefficients match.

```
beta_hat
```

```
##                    [,1]
##          -6186.7767966
## Breed      -83.1372876
## YrHgt       98.6848806
## FtFrBody    -1.7107980
## PrctFFB    -33.3062451
## Frame      284.4993376
## BkFat     1084.5221245
## SaleHt      85.9002423
## SaleWt       0.4582524
```

```
attach(data.frame(bulls.predictors))
```

```
## The following objects are masked from data.frame(bulls.predictors) (pos = 3):
##
##     BkFat, Breed, Frame, FtFrBody, PrctFFB, SaleHt, SaleWt, YrHgt
```

```
reg_obj <- lm(SalePr ~ Breed + YrHgt + FtFrBody + PrctFFB + Frame + BkFat + SaleHt + SaleWt, dat
a=bulls)

(reg_obj)
```

```
##
## Call:
## lm(formula = SalePr ~ Breed + YrHgt + FtFrBody + PrctFFB + Frame +
##     BkFat + SaleHt + SaleWt, data = bulls)
##
## Coefficients:
## (Intercept)        Breed        YrHgt     FtFrBody      PrctFFB        Frame
##  -6186.7768     -83.1373      98.6849      -1.7108     -33.3062     284.4993
##       BkFat       SaleHt       SaleWt
##   1084.5221      85.9002       0.4583
```

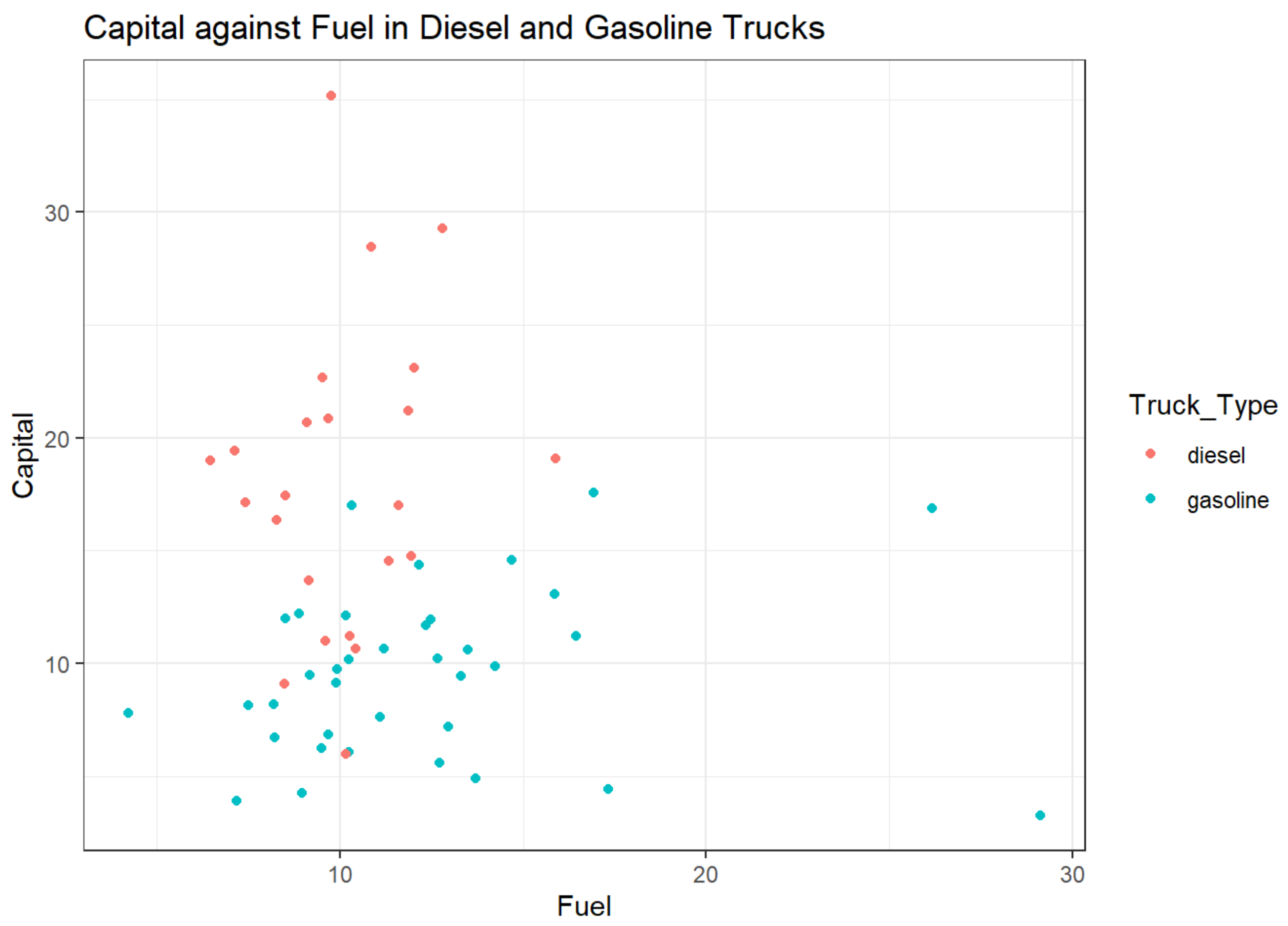
Milk Transportation Cost: LDA

Julia Webb

March 9, 2020

```
setwd("C:/Users/jwebb/OneDrive/Desktop/STA 135/project")
milk <- read.table("milk.dat")
colnames(milk) <- c("Fuel", "Repair","Capital", "Truck_Type")
#all measured in per mile basis
```

```
library(ggplot2)
ggplot(data=milk,aes(x=Fuel,y=Capital,col=Truck_Type)) + geom_point() + ggtitle("Capital against
Fuel in Diesel and Gasoline Trucks") +   labs(fill = "Truck Type") + theme_bw()
```



```
#ggplot(data=mi,aes(x=V2,y=V3,col=V4)) +geom_point()
```

Fuel and Capital seem to have close to a linear division, to we will remove the “Repair” column from our dataset and just investigate these variables

```
milk.2 <- milk[,-2]
```

separate classes

```
gasoline.class <- milk.2[which(milk.2$Truck_Type == "gasoline"),] #class 1
diesel.class <- milk.2[which(milk.2$Truck_Type == "diesel"),] #class 2
```

Hotellings T2 for diff between mean vectors:

```
#get class means
gasoline.class.2 <- gasoline.class
diesel.class.2 <- diesel.class

colnames(gasoline.class.2) <- NULL
colnames(diesel.class.2) <- NULL

mean.gasoline <- colMeans(gasoline.class.2[,1:2]) #leaving out truck_type col
mean.diesel <- colMeans(diesel.class.2[,1:2])#leaving out truck_type col



#get S pooled
n.gas <- length(gasoline.class[,1])
n.dies <- length(diesel.class[,1])
S.gas <- var(gasoline.class.2[,1:2])
S.dies <- var(diesel.class.2[,1:2])
S.gas <- var(gasoline.class.2[,1:2])
S.dies <- var(diesel.class.2[,1:2])
S.pooled <- ((n.gas-1)/(n.gas + n.dies -2))*S.gas + ((n.dies -1)/(n.gas + n.dies -2))*S.dies
```

```
#calculate T^2
T.2 <- t(mean.gasoline - mean.diesel)%*%solve(((1/(n.gas))+(1/n.dies))*S.pooled) %*%  (mean.gaso
line - mean.diesel)
```

```
#calcualte F
F.stat <- qf(.05,2, n.gas + n.dies -1 - 2)
```

T^2 > F, so we reject the null that the difference between the means of gasoline and diesel trucks is = 0.

```
#compare
c(T.2, F.stat)
```

```
## [1] 46.88069636  0.05134031
```

From this point forward, we will attempt to classify a truck as either diesel or gasoline based on its fuel and capital values

compute w and w0 pooled estimate for the covariance matrix:

```
w <- solve(S.pooled)%*%(mean.gasoline - mean.diesel)
w0 <- -(mean.gasoline + mean.diesel)%*%w/2
```

```
w
```

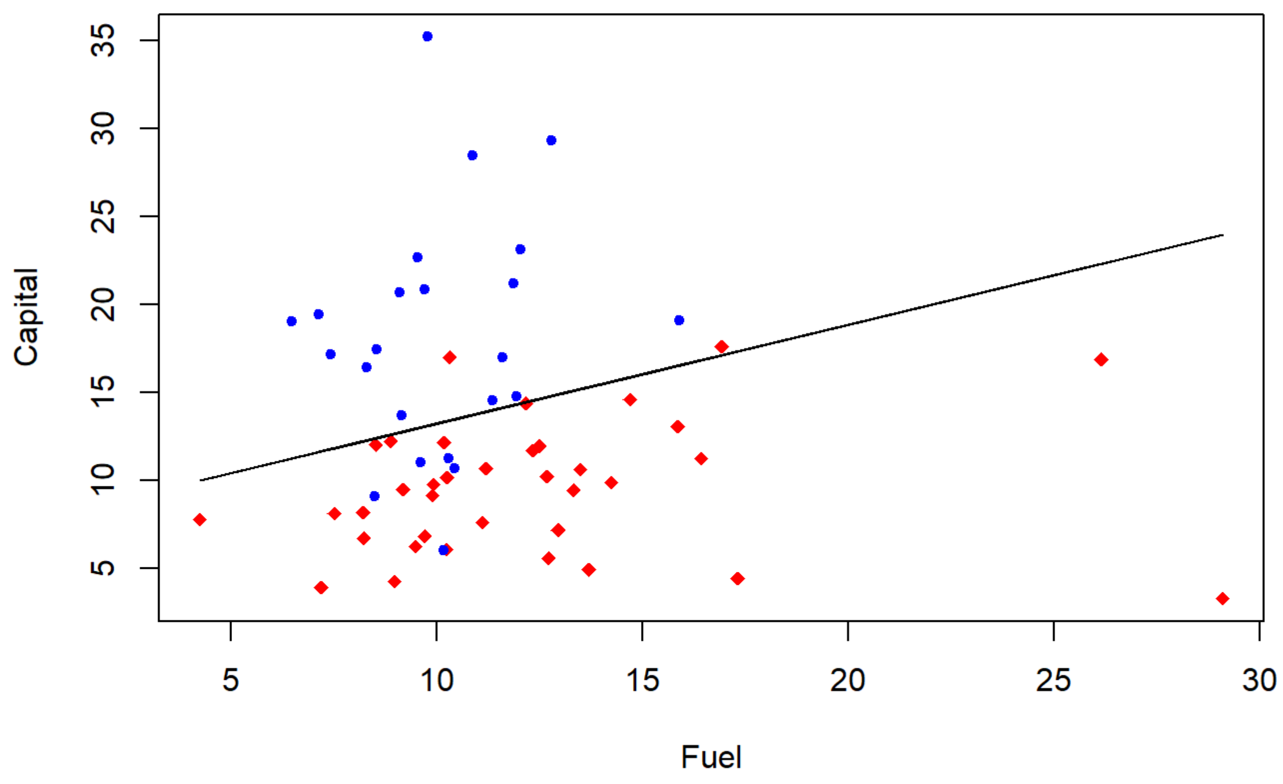
```
##              [,1]
## [1,]  0.1919475
## [2,] -0.3421672
```

```
w0
```

```
##            [,1]
## [1,] 2.606413
```

```
plot(milk[,c(1,3)],pch=rep(c(18,20),each=36),col=rep(c(2,4),each=36))
lines(milk$Fuel,-(w[1]*milk$Fuel+w0)/w[2])
```

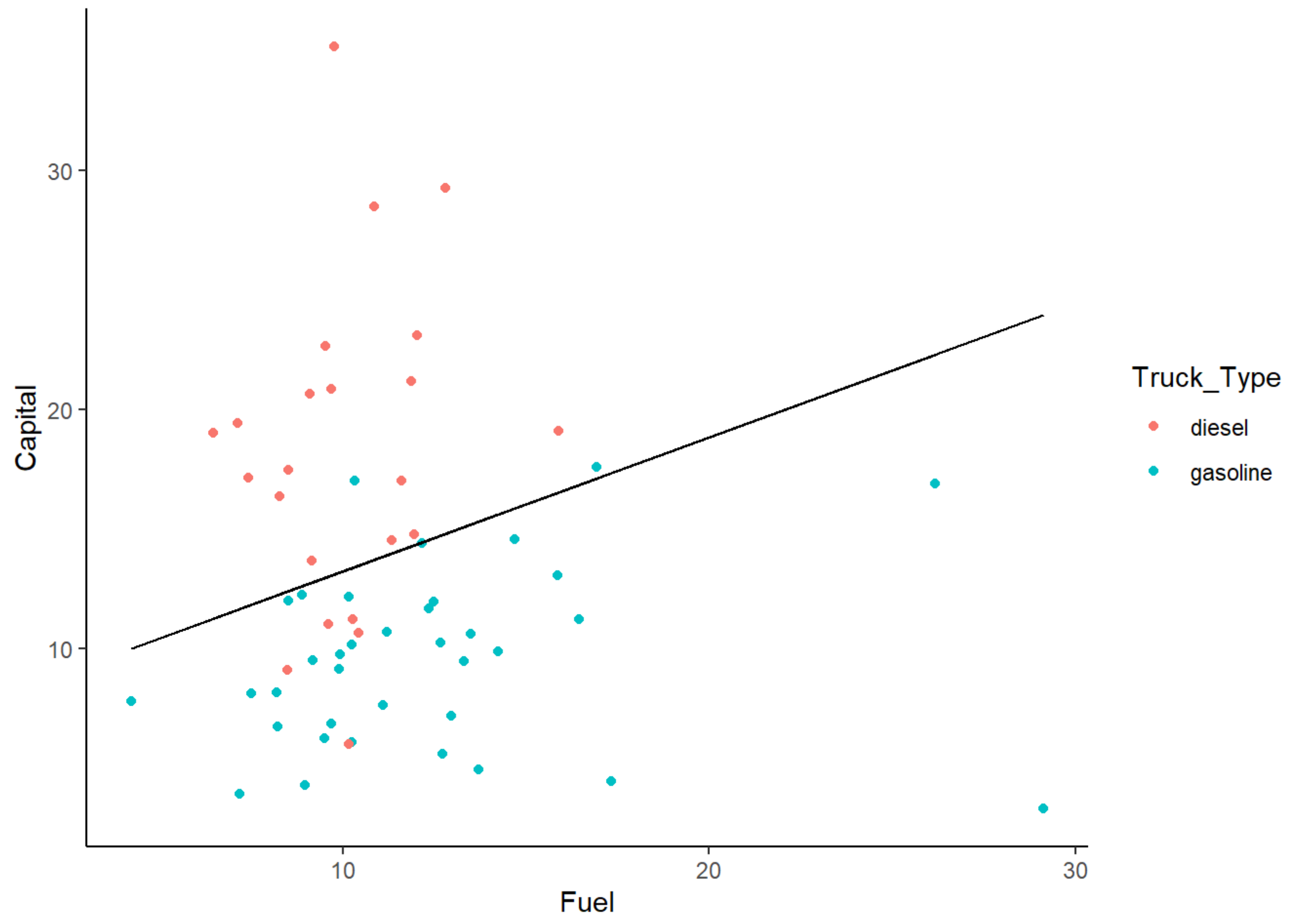
```
## Warning in w[1] * milk$Fuel + w0: Recycling array of length 1 in vector-array arithmetic is d
eprecated.
##   Use c() or as.vector() instead.
```



```
library(ggplot2)
ggplot(data=milk,aes(x=Fuel,y=Capital,col=Truck_Type)) +
  geom_point() +
  geom_segment(data= milk, aes(x=milk$Fuel, y=-(w[1]*milk$Fuel+w0)/w[2], xend = dplyr::lead(milk
$Fuel), yend = dplyr::lead(-(w[1]*milk$Fuel+w0)/w[2])), color="black")+ theme_classic()
```

```
## Warning in w[1] * milk$Fuel + w0: Recycling array of length 1 in vector-array arithmetic is d
eprecated.
##   Use c() or as.vector() instead.

## Warning in w[1] * milk$Fuel + w0: Recycling array of length 1 in vector-array arithmetic is d
eprecated.
##   Use c() or as.vector() instead.
```

```
## Warning: Removed 1 rows containing missing values (geom_segment).
```



Method 2: use function LDA in MASS package:

```
library(MASS)

lda.obj <- lda(Truck_Type ~ Fuel + Capital, data=milk, prior=c(1,1)/2)
plda <- predict(object=lda.obj,newdata=milk)
```

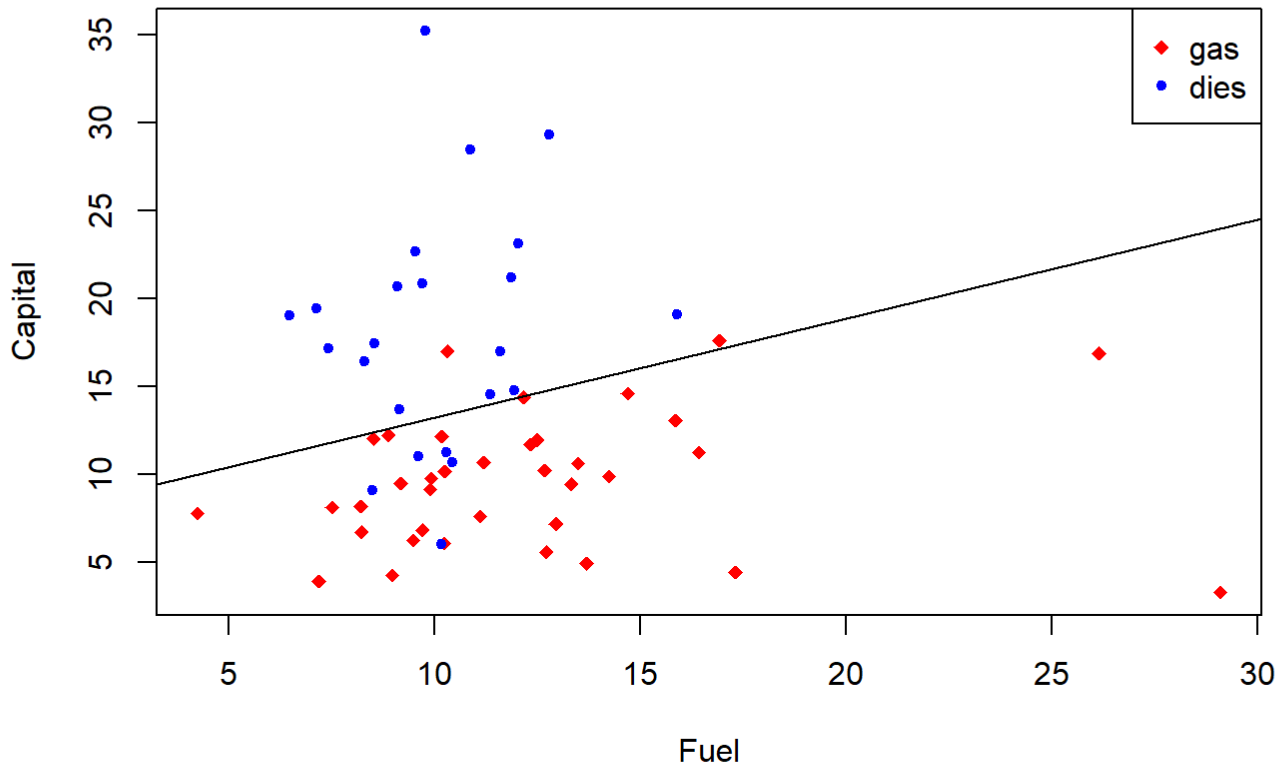
Plot decision boundary

```
library(MASS)
lda.obj <- lda(Truck_Type ~ Fuel + Capital, data=milk, prior=c(1,1)/2)
plda <- predict(object=lda.obj, newdata=milk)
```

plot decision line using plot()

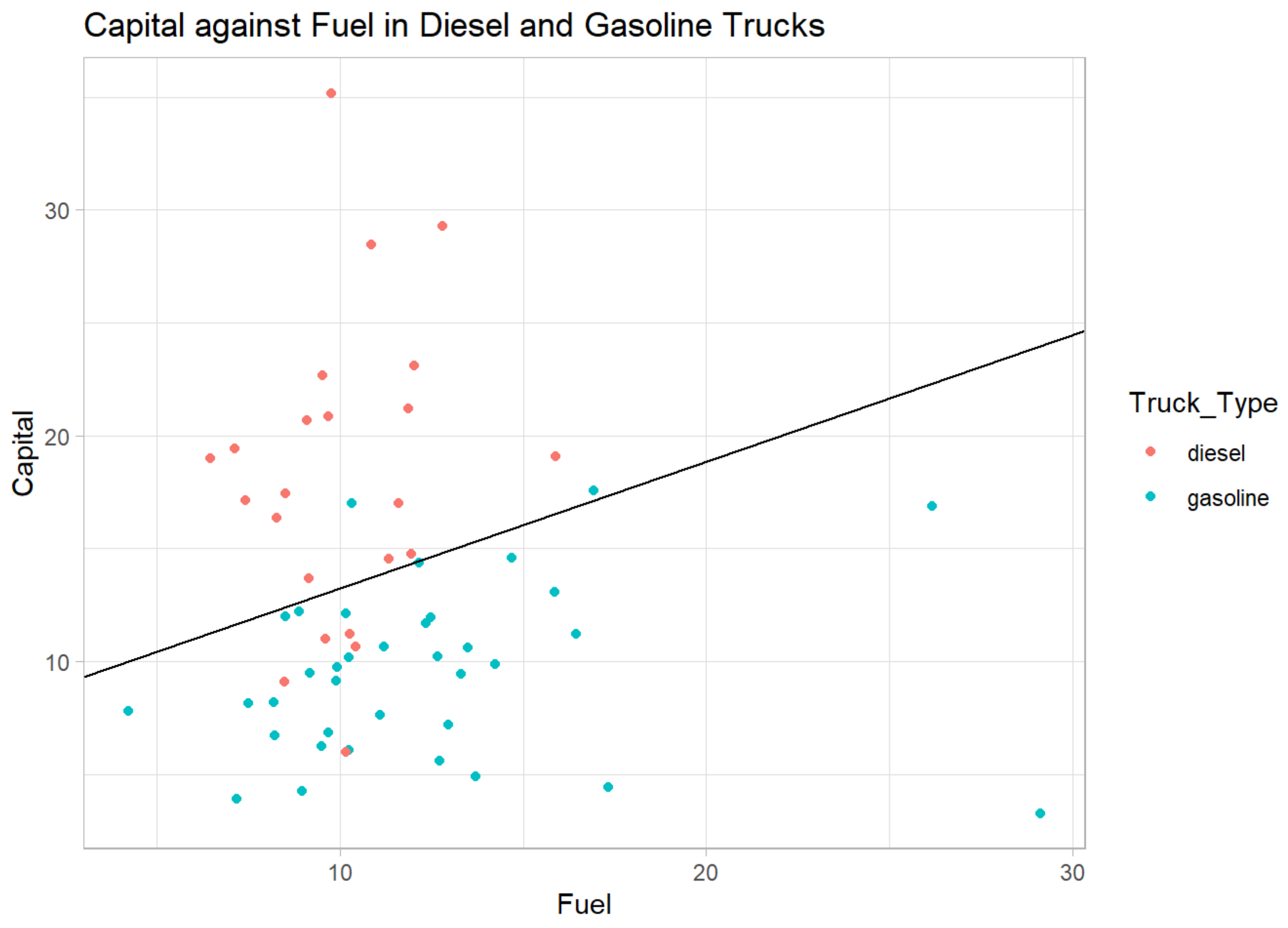
```
#plot the decision line
gmean <- lda.obj$prior %*% lda.obj$means
const <- as.numeric(gmean %*%lda.obj$scaling)
slo <- - lda.obj$scaling[1] / lda.obj$scaling[2]
int <- const / lda.obj$scaling[2]
```

```
plot(milk[,c(1,3)],pch=rep(c(18,20),each=36),col=rep(c(2,4),each=36))

abline(int, slo)
legend("topright",legend=c("gas","dies"),pch=c(18,20),col=c(2,4))
```



plot decison line using ggplot()

```
ggplot(data=milk,aes(x=Fuel,y=Capital,col=Truck_Type)) +
  geom_point() +
  geom_abline(intercept=int, slope=slo) +
  ggtitle("Capital against Fuel in Diesel and Gasoline Trucks") +
  labs(fill = "TruckType")+
  theme_light()
```



lets make a confusion table:

```
confusion <- table(milk$Truck_Type,plda$class)
confusion
```

```
##
##            diesel gasoline
##   diesel       18        5
##   gasoline      2       34
```

lets compute Apparent Error Rater (AER):

```
MASS::fractions(7/sum(confusion))
```

```
## [1] 7/59
```

functions to compute expected actual error rate by Lachenbruch’s holdout:

class 1: gas

```r
g <- as.matrix(gasoline.class.2[,1:2])
rownames(g) <-NULL


reclassify1 <- function(i){
#assembling classifier

  #start with group 1 and omit one obs
  gas.holdout <- g[-i,]
  holdout.obs <- g[i,]
  n.gas.holdout <- n.gas - 1

  #get new S.gas
  S.gas.holdout <- fractions(cov(gas.holdout))

  #mean of new class 1; note mean of class 2 is dies.mean
  gas.holdout.mean <- apply(gas.holdout,2,mean) #take the mean of each col and return a vector
  gas.holdout.mean <- as.matrix(gas.holdout.mean)
  rownames(gas.holdout.mean) <- NULL

  #S.pooled
  first_n_term <- (n.gas.holdout - 1)/(n.gas.holdout + n.dies -2)
  scnd_n_term <- (n.dies - 1)/(n.gas.holdout + n.dies - 2)

  S.pooled.holdout <- first_n_term*S.gas.holdout + scnd_n_term*S.dies
  S.pooled.inv.holdout <- solve(S.pooled.holdout)

  # classifier
  first_term <- t(gas.holdout.mean - mean.diesel)%*%S.pooled.inv.holdout
  second_term <- (holdout.obs - (1/2)*(gas.holdout.mean + mean.diesel))

  classifier <- first_term%*%second_term

  decision <- "ERR"
  ifelse(as.numeric(classifier) >= 0, decision <- 0, decision <- 1)

  return(decision)
}
```

```r
predictions.gas <- c()

for (i in c(1:n.gas)){

pred <- reclassify1(i)
predictions.gas <- c(predictions.gas, pred)

  }
```

class 2: diesel

```
d <- as.matrix(diesel.class.2[,1:2])
rownames(d) <-NULL

reclassify2 <- function(i){
#assembling classifier

  #start with group 2 and omit one obs
  dies.holdout <- d[-i,]
  holdout.obs <- d[i,]
  n.dies.holdout <- n.dies - 1

  #get new S.gas
  S.dies.holdout <- fractions(cov(dies.holdout))

  #mean of new class 1; note mean of class 2 is dies.mean
  dies.holdout.mean <- apply(dies.holdout,2,mean) #take the mean of each col and return a vector
  dies.holdout.mean <- as.matrix(dies.holdout.mean)
  rownames(dies.holdout.mean) <- NULL

  #S.pooled
  first_n_term <- (n.gas - 1)/(n.gas + n.dies.holdout -2)
  scnd_n_term <- (n.dies.holdout - 1)/(n.gas + n.dies.holdout - 2)

  S.pooled.holdout <- first_n_term*S.gas + scnd_n_term*S.dies.holdout
  S.pooled.inv.holdout <- solve(S.pooled.holdout)

  # classifier
  first_term <- t(mean.gasoline - dies.holdout.mean)%*%S.pooled.inv.holdout
  second_term <- (holdout.obs - (1/2)*(mean.gasoline + dies.holdout.mean))

  classifier <- first_term%*%second_term

  decision <- "ERR"
  if (classifier >= 0) decision = 0 else decision = 1

  return(decision)
}
```

```
predictions.dies <- c()
for (i in c(1:n.dies)){
pred <- reclassify2(i)
predictions.dies <- c(predictions.dies, pred)
  }
```

```
sum(predictions.dies != 1)
```

```
## [1] 5
```

expected actual error rate by Lachenbruch’s holdout:

```
MASS::fractions((sum(predictions.dies != 1) + sum(predictions.gas != 0))/length(milk.2[,1]))
```

```
## [1] 8/59
```

```
(sum(predictions.dies != 1) + sum(predictions.gas != 0))/length(milk.2[,1])
```

```
## [1] 0.1355932
```

Iris: PCA

Julia Webb

March 9, 2020

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

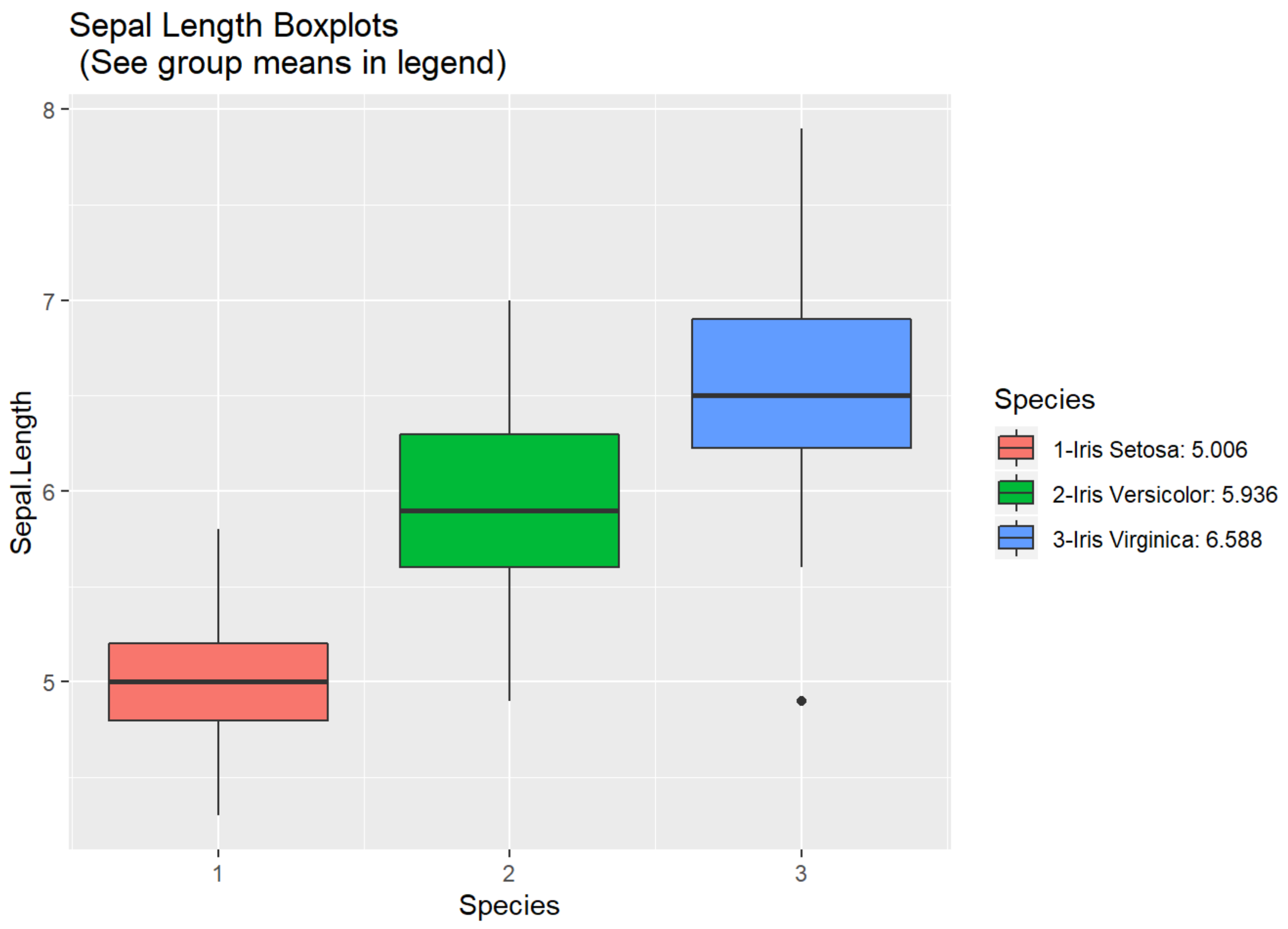
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

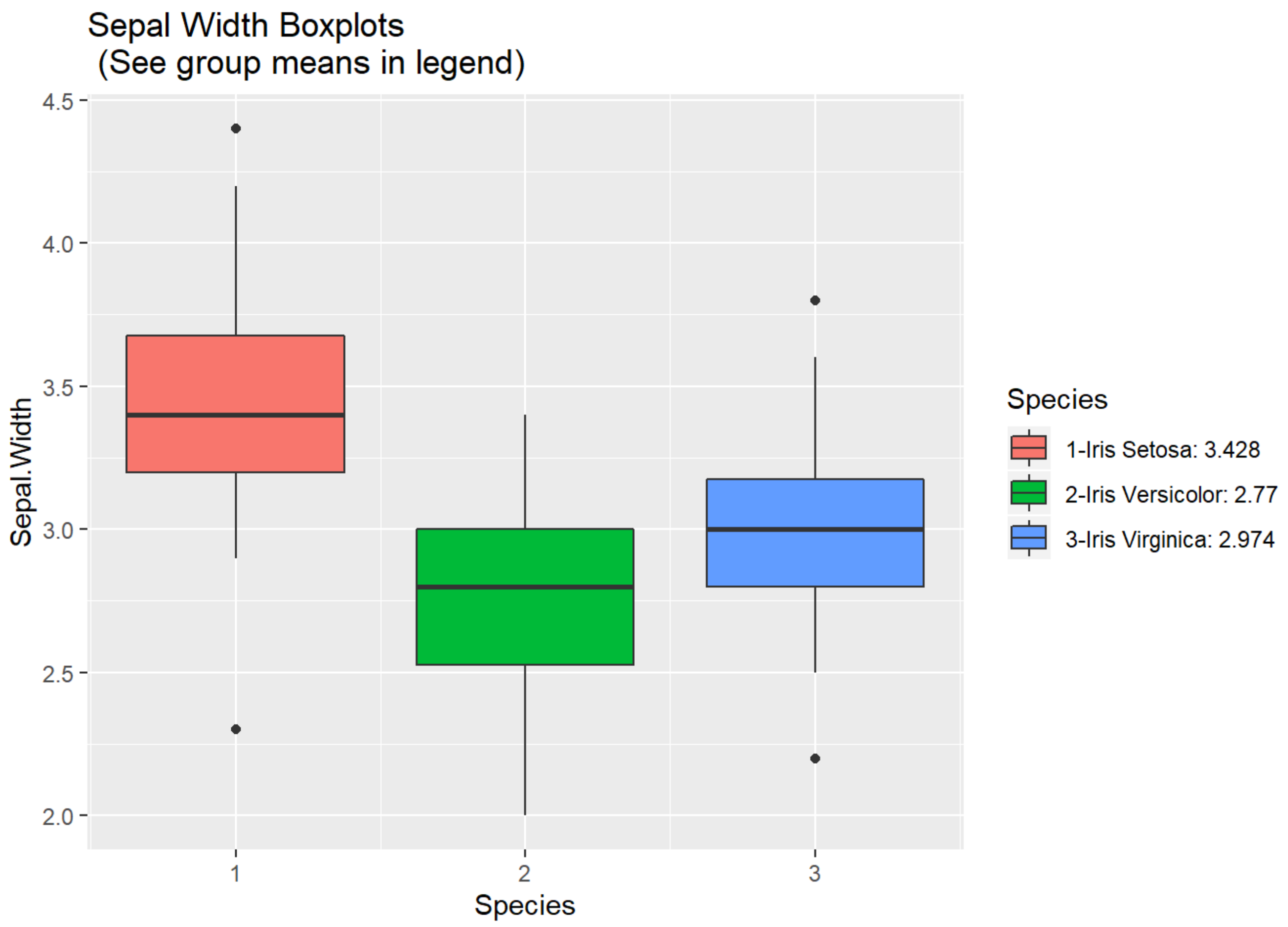
```r
iris <- read.table("t-11_5.dat")
colnames(iris) <- c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width", "Species")

iris.pca <- princomp(iris[,-5],cor=TRUE)
```

plot to summarize data

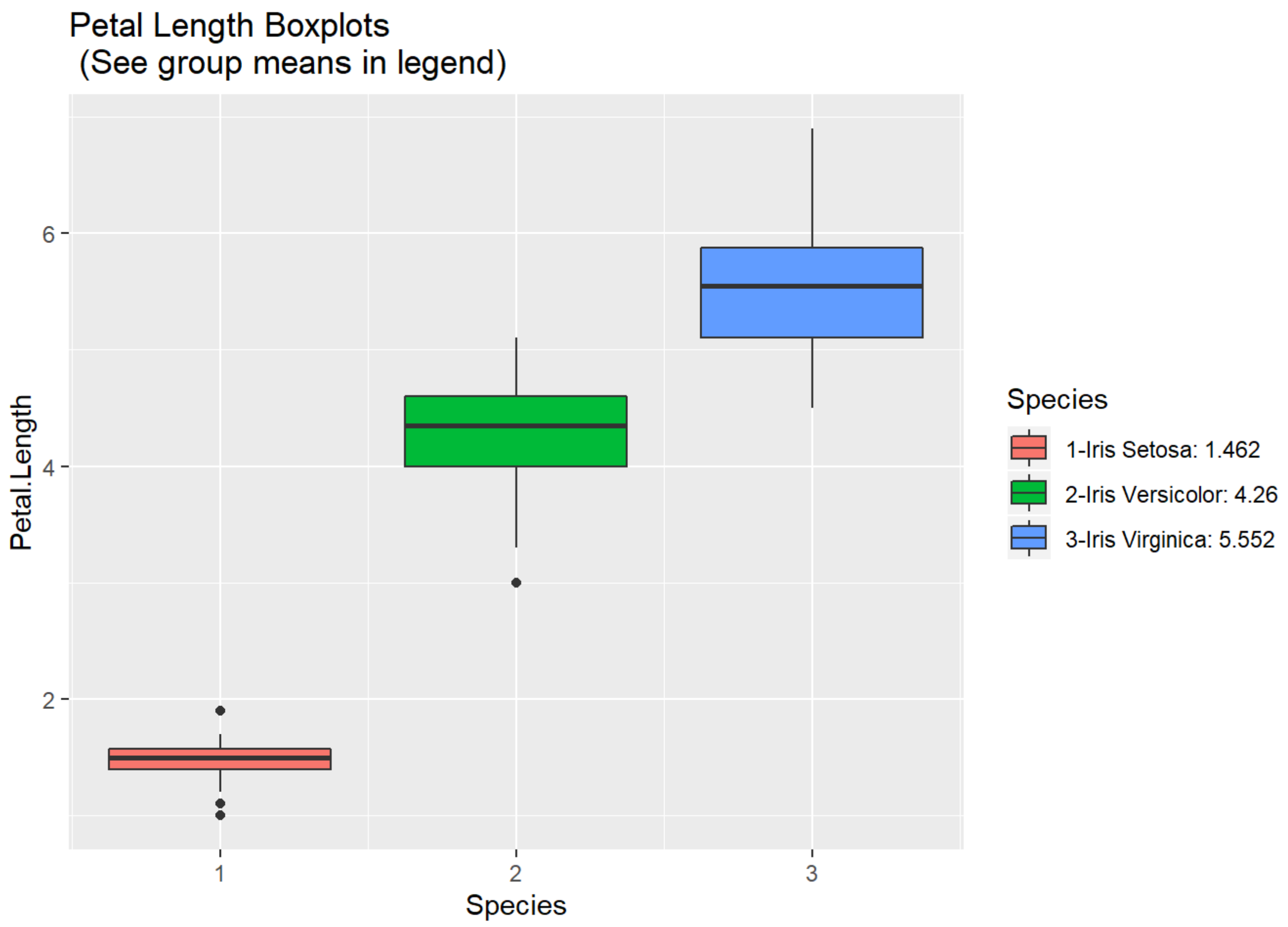
```r
library(ggplot2)

means.sepLength <- iris %>% group_by(Species) %>% summarise(m=mean(Sepal.Length)) #%>% pull(m)
 #gets tibble of means

means.sepWidth <- iris %>% group_by(Species) %>% summarise(m=mean(Sepal.Width))

means.petLength <- iris %>% group_by(Species) %>% summarise(m=mean(Petal.Length))

means.petWidth <- iris %>% group_by(Species) %>% summarise(m=mean(Petal.Width))


#Sepal

  #Length
ggplot(iris, aes(x=Species, y=Sepal.Length, group = Species, fill = factor(Species))) +
   geom_boxplot() +
   scale_fill_discrete(name = "Species", labels = c("1-Iris Setosa: 5.006", "2-Iris Versicolor:
 5.936", "3-Iris Virginica: 6.588")) +
   #geom_text(data = means.sepLength, aes(label = m, y = m + 0.08)) +
    ggtitle("Sepal Length Boxplots \n (See group means in legend)")
```



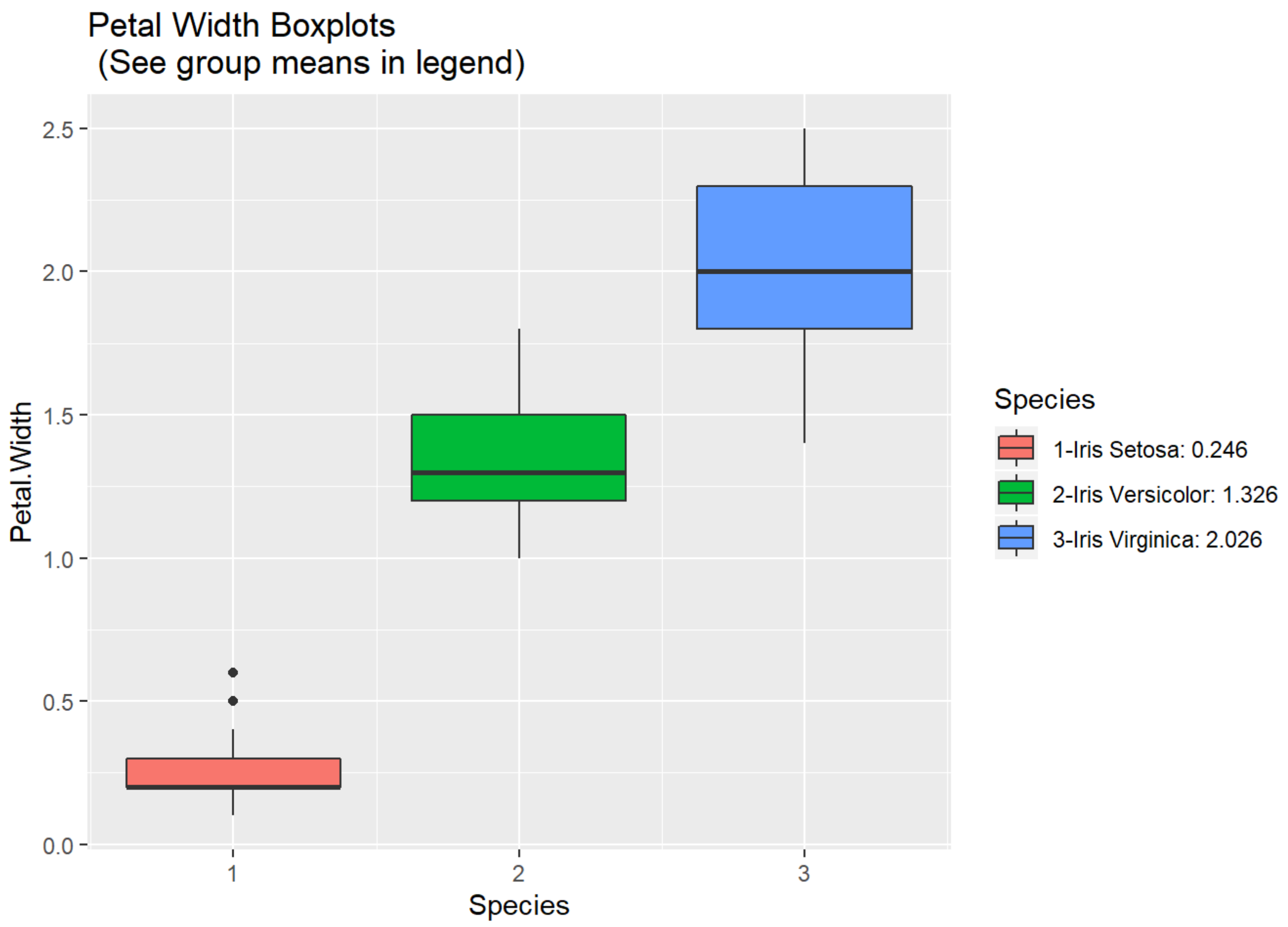
```
    #width
ggplot(iris, aes(x=Species, y=Sepal.Width, group = Species, fill = factor(Species))) +
   geom_boxplot() +
   scale_fill_discrete(name = "Species", labels = c("1-Iris Setosa: 3.428", "2-Iris Versicolor:
 2.77", "3-Iris Virginica: 2.974")) +
   #geom_text(data = means.sepWidth, aes(label = m, y = m + 0.1)) +
   ggtitle("Sepal Width Boxplots \n (See group means in legend)")
```



```
#Petal

   #Length
ggplot(iris, aes(x=Species, y=Petal.Length, group = Species, fill = factor(Species))) +
   geom_boxplot() +
   scale_fill_discrete(name = "Species", labels = c("1-Iris Setosa: 1.462", "2-Iris Versicolor:
 4.26", "3-Iris Virginica: 5.552")) +
  # geom_text(data = means.petLength, aes(label = m, y = m + 0.6)) +
   ggtitle("Petal Length Boxplots \n (See group means in legend)")
```



```
   #Width
ggplot(iris, aes(x=Species, y=Petal.Width, group = Species, fill = factor(Species))) +
   geom_boxplot() +
   scale_fill_discrete(name = "Species", labels = c("1-Iris Setosa: 0.246", "2-Iris Versicolor:
 1.326", "3-Iris Virginica: 2.026")) +
   #geom_text(data = means.petWidth, aes(label = m, y = m + .05)) +
   ggtitle("Petal Width Boxplots \n (See group means in legend)")
```



get principal components, along with importance of components

```
summary(iris.pca,loadings=TRUE)
```

```
## Importance of components:
##                           Comp.1    Comp.2     Comp.3      Comp.4
## Standard deviation     1.7083611 0.9560494 0.38308860 0.143926497
## Proportion of Variance 0.7296245 0.2285076 0.03668922 0.005178709
## Cumulative Proportion  0.7296245 0.9581321 0.99482129 1.000000000
## 
## Loadings:
##              Comp.1 Comp.2 Comp.3 Comp.4
## Sepal.Length  0.521  0.377  0.720  0.261
## Sepal.Width  -0.269  0.923 -0.244 -0.124
## Petal.Length  0.580        -0.142 -0.801
## Petal.Width   0.565        -0.634  0.524
```

Let's look at the eigenvalues of the correlation matrix, note this is the (standard deviation)^2 of each principal component, since each eigenvalue is the variance of the corresponding principal component

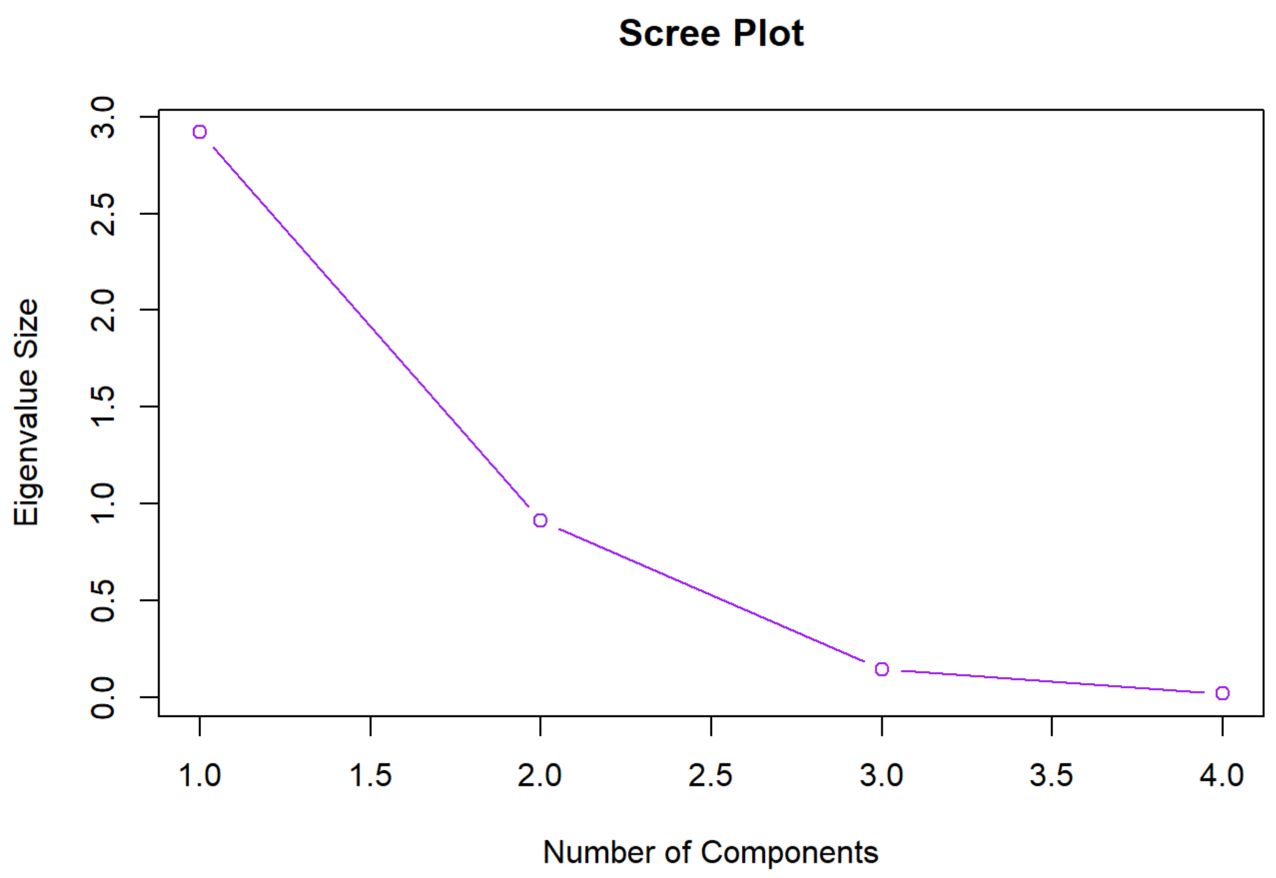
```
# Showing the eigenvalues of the correlation matrix:
(iris.pca$sdev)^2
```

```
##     Comp.1     Comp.2     Comp.3     Comp.4
## 2.91849782 0.91403047 0.14675688 0.02071484
```

A scree plot:

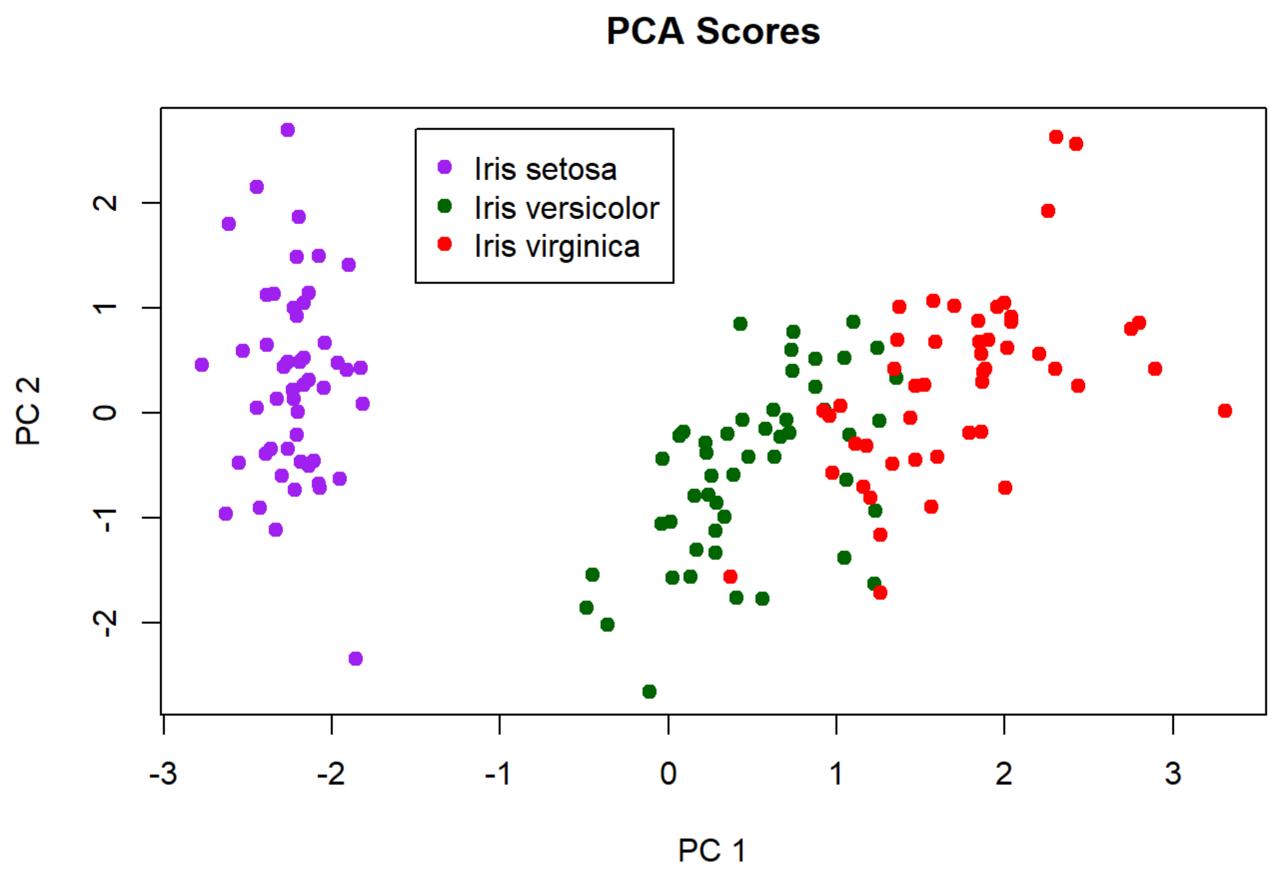
The elbow seems to occurr at the second principal component, which makes since the cumulative proportion of variance is 0.9502309 at principal component 2, as seen on the importance of components output above. With this in mind, it seems that 2 components is a reasonable number of PCs to use.

```
#plotting eigenvalues against principal component index ( ie 1...7)

plot(1:(length(iris.pca$sdev)),  (iris.pca$sdev)^2, type='b',
     main="Scree Plot", xlab="Number of Components", ylab="Eigenvalue Size", col = "purple")
```



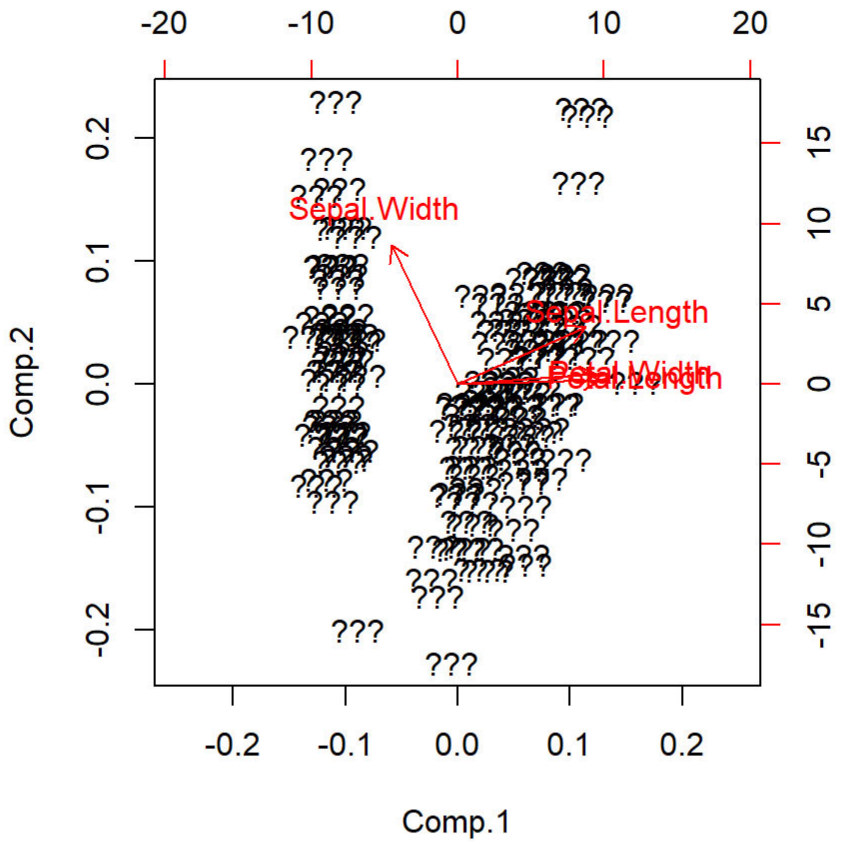
Plotting the PC scores for the sample data in the space of the first two principal components:

```
plot(iris.pca$scores[,1], iris.pca$scores[,2],
     xlab="PC 1", ylab="PC 2",pch=19, col=c(rep("purple",50),rep("dark green",50),rep("red",50
)), main="PCA Scores")

legend(-1.5,2.7,legend= c("Iris setosa", "Iris versicolor", "Iris virginica"),
       col=c("purple", "dark green", "red"), pch=19)
```



lets see a biplot

```
biplot(iris.pca, xlabs=rep("???", nrow(iris)),colby=iris$Species, xlim = c(-0.25,0.25))
```



After this analysis, we can conclude that it would be appropriate to include 2 principal components without risking much loss of information