---

output:

pdf\_document: default

html\_document: default

---

# Exploratory Data Analysis

>Exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods.[^1]

[^1]:https://en.wikipedia.org/wiki/Exploratory\_data\_analysis

## Introduction

The experiment described herein involves taking groups of proteins from the Uniprot.org database and comparing how well different machine learning techniques do at separating the positive from the negative control grouping. In this circumstance, proteins from the myoglobin family are analyzed against randomly chosen human proteins, which are not related to hemoglobin or myoglobin.

This work is to characterize the outliers derived from PCA and compare them to the false-positives and false-negatives generated from each of 6 machine learning approaches produces;

1. Logit,

2. SVM-Linear,

3. SVM-polynomial,

4. SVM-RBF,

5. Random Forest,

6. Neural Network with auto-encoding.

## Four-Step Analysis

At this stage, data is inspected in a careful and structured way. Hence, I have chosen a four-step process:

>\*\*Hypothesize -> Summarize -> Visualize -> Normalize\*\*

---

#### Useful Guides for Exploratory Data Analysis {-}

The summarization of the amino acid dataset is based on a hybrid set of guidelines;

1. NIST Handbook of Statistics,[^4]

2. Roger Peng's booklet on 'Exploratory Data Analysis with R,' [^5]

3. 'Exploratory Data Analysis Using R,' by Ronald K. Pearson.[^6]

---

[^4]:https://www.itl.nist.gov/div898/handbook/

[^5]:Peng, Roger, Exploratory Data Analysis with R, https://leanpub.com/exdata, 2016

[^6]:Ronald Pearson, 'Exploratory Data Analysis Using R,' P.11, CRC Press, ISBN:9781138480605, 2018

## Hypothesize - Questions During EDA

Although exploratory data analysis does not always have a formal hypothesis testing portion, I do, however, pose several questions concerning the structure, quality, and types of data.

1. Do the independent variables of this study have large skewed distributions?

1.1 If skews are greater than 2.0, then can a transformation be used for normalization?

1.2 Determine what transformation to use?

2. Can \*\*Feature Selection\*\* be used, and which procedures are appropriate?

2.1 Use the Random Forest technique known as Boruta[^2] for feature importance or reduction?

2.2 Will coefficients of correlation (R) find collinearity and reduce the number of features?

2.3 Will principal component analysis (PCA) be useful in finding hidden structures of patterns?

2.4 Can PCA be used successfully for Feature Selection?

3. What is the structure of the data?

3.1 Is the data representative of the entire experimental space?

3.2 Is missing data an issue?

3.3 Does the data have certain biases, either known or unknown?

3.4 What relationships do we expect from these variables?[^3]

[^2]:Miron Kursa, Witold Rudnicki, Feature Selection with the Boruta Package, DOI:10.18637/jss.v036.i11, 2010

[^3]:Ronald Pearson, 'Exploratory Data Analysis Using R,' P.11, CRC Press, 2018

## Analysis of RAW data

```{r message=FALSE}

# Import libraries

Libraries <- c("knitr", "readr", "RColorBrewer", "corrplot", "doMC", "Boruta")

for (i in Libraries) {

library(i, character.only = TRUE)

}

```

```{r include=FALSE}

opts\_chunk$set(cache = TRUE)

```

```{r message=FALSE, warning=FALSE}

# Import RAW data

c\_m\_RAW\_AAC <- read\_csv("../00-data/02-aac\_dpc\_values/c\_m\_RAW\_AAC.csv")

Class <- as.factor(c\_m\_RAW\_AAC$Class)

```

### Visually inspect RAW data files

1. Use the command-line interface followed by the command `less.`

2. Check for binary instead of ASCII and bad Unicode.

### Inspect RAW dataframe structure, `str()`

```{r echo=FALSE}

str(c\_m\_RAW\_AAC)

```

### Check RAW data `head` & `tail`

```{r}

head(c\_m\_RAW\_AAC, n = 2)

```

```{r}

tail(c\_m\_RAW\_AAC, n = 2)

```

### Check RAW data types

```{r echo=TRUE}

is.data.frame(c\_m\_RAW\_AAC)

class(c\_m\_RAW\_AAC$Class) # Col 1

class(c\_m\_RAW\_AAC$TotalAA) # Col 2

class(c\_m\_RAW\_AAC$PID) # Col 3

class(c\_m\_RAW\_AAC$A) # Col 4

```

### Check RAW dataframe dimensions

```{r}

dim(c\_m\_RAW\_AAC)

```

### Check RAW for missing values

- \*\*No missing values found.\*\*

```{r}

apply(is.na(c\_m\_RAW\_AAC), 2, which)

# sapply(c\_m\_RAW\_AAC, function(x) sum(is.na(x))) # Sum up NA by columns

# c\_m\_RAW\_AAC[rowSums(is.na(c\_m\_RAW\_AAC)) != 0,] # Show rows where NA's is not zero

```

### Number of polypeptides per Class:

- Class 0 = Control,

- Class 1 = Myoglobin

```{r echo=FALSE}

class\_table <- table(c\_m\_RAW\_AAC$Class)

class\_table

```

### Numerical summary of RAW features

```{r echo=FALSE}

summary(c\_m\_RAW\_AAC)

```

### Visualize RAW Data With Descriptive Statistics

Formulas for mean:

$$E[X] = \sum\_{i=1}^n x\_i p\_i ~~; ~~~~~~ \bar x = \frac {1}{n} \sum\_{i=1}^n x\_i$$

### Scatter plot of means of \*Myoglobin-Control\* amino acid composition of `c\_m\_RAW\_AAC` dataframe

- This plot shows the means for each feature (column-means) in the dataset. The means represent the ungrouped or total of all proteins (where n = 2340) versus AA type.

```{r echo=FALSE, fig.cap='Column-Means of % Composition Vs Amino Acid', out.width='100%', fig.asp=1, fig.align='center'}

AA\_ave <- colMeans(c\_m\_RAW\_AAC[, 4:23])

plot(AA\_ave,

main = "Plot: Column-Means of % Composition Vs Amino Acid",

ylab = "% Composition",

xlab = "Amino Acid",

ylim = c(0, 0.1),

type = "b",

xaxt = "n")

axis(1, at = 1:20, labels = names(c\_m\_RAW\_AAC[, 4:23]))

```

```{r cache=TRUE, include=FALSE}

# \* Pseudo-code for Data manipulation for Grouped Bar charts

# + A-1. Subset 7 protein groups, [Control:Ctrl, Myoglobin:Mgb] & Grand-Mean of both sets

# + A-2. Determine column means for each protein class

# + A-3. Calculate percentage values

# + A-4. Produce Grouped Bar Plot

# A-1

ctrl\_set <- c\_m\_RAW\_AAC[which(c\_m\_RAW\_AAC$Class == "0"), ]

mgb\_set <- c\_m\_RAW\_AAC[which(c\_m\_RAW\_AAC$Class == "1"), ]

# A-2

ctrl\_means <- apply(ctrl\_set[, 4:23], 2, mean)

mgb\_means <- apply(mgb\_set[, 4:23], 2, mean)

grand\_mean <- apply(c\_m\_RAW\_AAC[, 4:23], 2, mean)

# A-3

data <- data.frame(ctrl\_means, mgb\_means, grand\_mean)

percent\_aa <- as.matrix(t(100 \* data))

```

```{r echo=TRUE, eval=FALSE, message=FALSE, warning=FALSE, include=FALSE}

# A-4

### Grouped barchart of amino acid vs. protein category

barplot(percent\_aa,

main = "Mean % A.A.Composition Of 3 Protein Groupings",

ylab = "% AA Composition",

ylim = c(0, 12),

col = colorRampPalette(brewer.pal(4, "Blues"))(3),

legend = T,

beside = T)

```

### Means of percent amino acid composition of control & myoglobin categories, RAW data

```{r echo=FALSE, message=FALSE, warning=FALSE, cache=TRUE}

data2 <- data.frame(ctrl\_means, mgb\_means)

percent\_aa2 <- as.matrix(t(100 \* data2))

barplot(percent\_aa2,

ylim = c(0, 12),

main = "Mean % A.A.Composition Of Control & Myoglobin",

ylab = "% AA Composition",

col = colorRampPalette(brewer.pal(4, "Blues"))(2),

legend = T,

beside = T)

```

### Boxplots of grand-means of overall amino acid composition, RAW data

```{r echo=FALSE, message=FALSE, warning=FALSE}

boxplot(c\_m\_RAW\_AAC[, 4:23],

main = "Boxplots: All; % Composition Vs Amino Acid",

ylab = "% AAC",

xlab = "Amino Acid",

las = 1)

```

### Boxplots of amino acid compositions for control (only), RAW data

```{r echo=FALSE}

boxplot(ctrl\_set[, 4:23],

main = "Boxplots: Controls; % AAC Vs Amino Acid",

ylab = "% AAC",

xlab = "Amino Acid",

las = 1)

```

### Boxplots of amino acid compositions for myoglobin (only), RAW data

```{r echo=FALSE}

boxplot(mgb\_set[, 4:23],

ylim = c(0, 0.5),

main = "Boxplot: Myoglobin; % AAC Vs Amino Acid",

ylab = "% AAC",

xlab = "Amino Acid",

las = 1)

```

```{r echo=FALSE}

par(mfrow = c(1, 2))

boxplot(ctrl\_set[, 4:23],

ylim = c(0, 0.3),

main = "Boxplots: Controls",

ylab = "% AAC",

xlab = "Amino Acid",

las = 1)

boxplot(mgb\_set[, 4:23],

ylim = c(0, 0.3),

main = "Boxplot: Myoglobin",

xlab = "Amino Acid",

las = 1)

```

### Boxplots of Length of Polypeptides For Myoglobin, Control & Combined, RAW data

```{r echo=FALSE}

ctrl\_totalaa <- ctrl\_set[, 2]

mgb\_totalaa <- mgb\_set[, 2]

grand\_totalaa <- c\_m\_RAW\_AAC[, 2]

data <- c(ctrl\_totalaa, mgb\_totalaa, grand\_totalaa)

boxplot(data,

ylim = c(0, 5000),

main = "Boxplot: Length of Polypeptides Vs Control, Myoglobin & Combined",

ylab = "Length of Polypeptides",

xaxt = "n",

las = 2)

axis(1, at = 1:3, labels = c("Control", "Myoglobin", "Combined"))

```

### Plot the coefficient of variance (CV), RAW data

Standard deviations are sensitive to scale. Therefore I compare the normalized standard deviations. This normalized standard deviation is more commonly called the coefficient of variation (CV).

$$CV = \frac {\sigma (x)} {E [|x|]} ~~~ where ~~~ \sigma(x) \equiv \sqrt{ E[x - \mu]^2 }$$

$$CV ~~=~~ \frac{1}{\bar x} \cdot \sqrt{ \frac{1}{n-1} \sum^n\_{i=1} (x\_i - \bar x)^2}$$

```{r echo=FALSE}

AA\_var\_norm <- (apply(c\_m\_RAW\_AAC[, 4:23], 2, sd)) / AA\_ave

plot(AA\_var\_norm,

main = "Plot of Coefficient Of Variance (CV) Vs 20 Std AA",

sub = "(Note: Two largest values shown in red.)",

ylab = "Coefficient Of Variance (CV)",

xlab = "Amino Acid",

ylim = c(0, 1.5),

type = "b",

xaxt = "n")

axis(1, at = 1:20, labels = names(c\_m\_RAW\_AAC[, 4:23]))

text(x = 2, y = 1.4, label = " C=1.24", col = "red")

text(x = 19, y = 1.1, label = "W=0.946", col = "red")

```

```{r}

AA\_var\_norm

```

### Skewness of distributions, RAW data

$$Skewness ~= E\left[ \left( \frac{X - \mu}{\sigma(x)} \right)^3 \right] ~~~~ where ~~~~ \sigma(x) \equiv \sqrt{ E[x - \mu]^2 }$$

$$Skewness ~= \frac { \frac{1}{n} \sum^n\_{i=1} (x\_i - \bar x)^3 } { \left( \sqrt{ \frac{1}{n-1} \sum^n\_{i=1} (x\_i - \bar x)^2 } \right) ^ {3}}$$

Skewness values for each A.A. are determined in totality

```{r echo=FALSE}

AA\_skewness <- (apply(c\_m\_RAW\_AAC[, 4:23], 2, e1071::skewness))

plot(AA\_skewness,

main = "Plot of Skewness Vs Amino Acids",

ylab = "Skewness",

xlab = "Amino Acid",

type = "b",

ylim = c(-0.5, 3),

xaxt = "n")

axis(1, at = 1:20, labels = names(c\_m\_RAW\_AAC[, 4:23]))

abline(h = 2.0, col = "red")

text(x = 2, y = 2.8, label = " C=2.5", col = "red")

text(x = 5, y = 2.4, label = "F=2.1", col = "red")

text(x = 8, y = 2.4, label = "I=2.2", col = "red")

```

### QQ-Plots of 20 amino acids, RAW data

```{r echo=FALSE}

AA <- c("A", "C", "D", "E", "F", "G", "H", "I", "K", "L",

"M", "N", "P", "Q", "R", "S", "T", "V", "W", "Y")

par(mfrow=c(2,2))

for (i in 4:23) {

qqnorm(c\_m\_RAW\_AAC[[i]], main = AA[[i - 3]])

qqline(c\_m\_RAW\_AAC[[i]], col = "red")

}

```

### Determine coefficients of correlation, RAW data

An easily interpretable test is a correlation 2D-plot for investigating multicollinearity or feature reduction. Fewer attributes "means decreased computational time and complexity. Secondly, if two predictors are highly correlated, this implies that they measure the same underlying information. Removing one should not compromise the performance of the model and might lead to a more parsimonious and interpretable model. Third, some models can be crippled by predictors with degenerate distributions".[^11]

[^11]:Max Kuhn and Kjell Johnson, Applied Predictive Modeling, Springer Publishing, 2018, P.43

Pearson's correlation coefficient:

$$\rho\_{x,y} = \frac {E \left[(X - \mu\_x)(X - \mu\_y) \right]} {\sigma\_x \sigma\_y}$$

$$r\_{xy} = \frac {\sum^n\_{i=1} (x\_i - \bar x)(y\_1 - \bar y)} { {\sqrt {\sum^n\_{i=1} (x\_i - \bar x)^2 }} {\sqrt {\sum^n\_{i=1} (y\_i - \bar y)^2 }} }$$

```{r message=FALSE, warning=FALSE}

c\_m\_corr\_mat <- cor(c\_m\_RAW\_AAC[, c(2, 4:23)],

method = "p") # "p": Pearson test for continous variables

corrplot(abs(c\_m\_corr\_mat),

title = "Correlation Plot Of AAC Features",

method = "square",

type = "lower",

tl.pos = "d",

cl.lim = c(0, 1),

addgrid.col = "lightgrey",

cl.pos = "b", # Color legend position bottom.

order = "FPC", # "FPC" = first principal component order.

mar = c(1, 2, 1, 2),

tl.col = "black")

```

NOTE: Amino acids shown in First Principal Component order, top to bottom.

1. Maximum value of Correlation between T & N.

```{r echo=FALSE}

c\_m\_corr\_mat["T", "N"]

```

2. According to Max Kuhn[^9], correlation coefficients need only be addressed if the |R| >= 0.75.

3. Therefore is \*\*no reason to consider multicollinearity\*\*.

---

### How to: Dimension Reduction using High Correlation

How to reduce features given high correlation (|R| >= 0.75) {-}

1. Calculate the correlation matrix of the predictors.

2. If the correlation plot produced of any two variables is greater than or equal to (|R| >= 0.75), then we could consider feature elimination. This interesting heuristic approach would be used for determining which feature to eliminate.[^12]

3. Determine if the two predictors associated with the most significant absolute pairwise correlation (R > |0.75|), call them predictors A and B.

4. Determine the average correlation between A and the other variables. Do the same for predictor B.

5. If A has a more significant average correlation, remove it; otherwise, remove predictor B.

6. Repeat Steps 2–4 until no absolute correlations are above the threshold.

---

[^12]:Max Kuhn and Kjell Johnson, Applied Predictive Modeling, Springer Publishing, 2018, P.47 (http://appliedpredictivemodeling.com/)

### Boruta - dimensionality reduction, RAW data

```{r}

c\_m\_class\_20 <- c\_m\_RAW\_AAC[, -c(2, 3)] # Remove TotalAA & PID

Class <- as.factor(c\_m\_class\_20$Class) # Convert 'Class' To Factor

```

NOTE: \*mcAdj = TRUE\*, If True, multiple comparisons will be adjusted using the Bonferroni method to calculate p-values. Therefore, $p\_i \leq \large \frac {\alpha} {m}$ where $\alpha$ is the desired p-value and $m$ is the total number of null hypotheses.

```{r cache=TRUE}

set.seed(1000)

registerDoMC(cores = 3) # Start multi-processor mode

start\_time <- Sys.time() # Start timer

boruta\_output <- Boruta(Class ~ .,

data = c\_m\_class\_20[, -1],

mcAdj = TRUE, # See Note above.

doTrace = 1) # doTrace = 1, represents non-verbose mode.

registerDoSEQ() # Stop multi-processor mode

end\_time <- Sys.time() # End timer

end\_time - start\_time # Display elapsed time

```

```{r eval=FALSE, include=FALSE}

names(boruta\_output)

```

### Plot variable importance

```{r echo=FALSE}

plot(boruta\_output,

cex.axis = 1,

las = 2,

ylim = c(-5, 50),

main = "Variable Importance (Bigger=Better)")

```

### Variable importance scores

```{r echo=FALSE, message=FALSE}

roughFixMod <- TentativeRoughFix(boruta\_output)

imps <- attStats(roughFixMod)

imps2 <- imps[imps$decision != "Rejected", c("meanImp", "decision")]

meanImps <- imps2[order(-imps2$meanImp), ] # descending sort

knitr::kable(meanImps,

full\_width = F,

position = "left",

caption = "Mean Importance Scores & Decision")

```

```{r eval=FALSE, include=FALSE}

## Plot importance history

plotImpHistory(boruta\_output)

```

### Conclusion for Boruta random forest test

- All features are essential. None should be dropped.

### Conclusions For EDA, RAW data

Three amino acids (C, F, I) from the single amino acid percent composition are transformed by using the square root function. A quick investigation (data not shown) showed that a square root transformation would be sufficient. The square root transformation lowered the skewness from greater than 2 in all cases to {-0.102739 $\leq$ skew after transformation $\leq$ 0.3478132}.

| Protein | Initial skewness | Skew after square root transform |

| :--------------- | :--------------: | :------------------------------: |

| C, Cysteine | 2.538162 | 0.3478132 |

| F, Phenolalanine | 2.128118 | -0.102739 |

| I, Isoleucine | 2.192145 | 0.2934749 |

---

## Analysis of TRANSFORMED data

\*\*This EDA section is a reevaluation square root transformed, `c\_m\_RAW\_ACC.csv` data set, hence called `c\_m\_TRANSFORMED.csv.` \*\*

The $\sqrt x\_i$ \*Transformed\* data is derived from `c\_m\_RAW\_ACC.csv` where the amino acids C, F, I were transformed using a square root function. This transformation was done to reduce the skewness of these samples and avoid modeling problems arising from high skewness, as seen below.

| Amino Acid | Initial skewness | Skew after square root transformation |

| :--------------- | :--------------: | :-----------------------------------: |

| C, Cysteine | 2.538162 | 0.3478132 |

| F, Phenolalanine | 2.128118 | -0.102739 |

| I, Isoleucine | 2.192145 | 0.2934749 |

```{r message=FALSE, include=FALSE}

# Import libraries

Libraries <- c("knitr", "readr", "RColorBrewer", "corrplot", "doMC", "Boruta")

for (i in Libraries) {

library(i, character.only = TRUE)

}

opts\_chunk$set(cache = TRUE, warning = FALSE, message = FALSE)

```

```{r message=FALSE, warning=FALSE}

# Import Transformed data

c\_m\_TRANSFORMED <- read\_csv("../00-data/02-aac\_dpc\_values/c\_m\_TRANSFORMED.csv")

Class <- as.factor(c\_m\_TRANSFORMED$Class)

```

### Check Transformed dataframe dimensions

```{r}

dim(c\_m\_TRANSFORMED)

```

### Check Transformed for missing values

```{r}

apply(is.na(c\_m\_TRANSFORMED), 2, which)

```

- No missing values found.

### Count Transformed data for the number of polypeptides per class

Number of polypeptides per Class:

- Class 0 = Control,

- Class 1 = Myoglobin

```{r echo=FALSE}

class\_table <- table(c\_m\_TRANSFORMED$Class)

class\_table

```

### Visualization of Transformed Data Descriptive Statistics

Formulas for mean:

$$E[X] = \sum\_{i=1}^n x\_i p\_i ~~; ~~~~~~ \bar x = \frac {1}{n} \sum\_{i=1}^n x\_i$$

### Scatter plot of means of \*Myoglobin-Control\* amino acid composition $\sqrt x\_i$ Transformed (c\_m\_TRANSFORMED) dataframe

- This plot shows the means for each feature (column-means) in the dataset. The means represent the ungrouped or total of all proteins (where n=2340) versus AA type.

```{r echo=FALSE}

AA\_ave <- colMeans(c\_m\_TRANSFORMED[, 4:23])

plot(AA\_ave,

main = "Column-Means Vs Amino Acid of Squareroot Transformed Data",

ylab = "% Composition",

xlab = "Amino Acid",

#sub = "(Note: The red line at 0.1 is simply an arbitrary marker)",

ylim = c(0, 0.3),

type = "b",

xaxt = "n")

axis(1, at = 1:20, labels = names(c\_m\_TRANSFORMED[, 4:23]))

```

```{r echo=FALSE, cache=TRUE}

# \* Pseudo-code for Data manipulation for Grouped Bar charts

# + A-1. Subset 7 protein groups, [Control:Ctrl, Myoglobin:Mgb] & Grand-Mean of both sets

# + A-2. Determine column means for each protein class

# + A-3. Calculate percentage values

# + A-4. Produce Grouped Bar Plot

# A-1

ctrl\_set <- c\_m\_TRANSFORMED[which(c\_m\_TRANSFORMED$Class == "0"), ]

mgb\_set <- c\_m\_TRANSFORMED[which(c\_m\_TRANSFORMED$Class == "1"), ]

# A-2

ctrl\_means <- apply(ctrl\_set[, 4:23], 2, mean)

mgb\_means <- apply(mgb\_set[, 4:23], 2, mean)

grand\_mean <- apply(c\_m\_TRANSFORMED[, 4:23], 2, mean)

# A-3

data <- data.frame(ctrl\_means, mgb\_means, grand\_mean)

percent\_aa <- as.matrix(t(100 \* data))

```

```{r eval=FALSE, include=FALSE}

# A-4

## Grouped barchart of $\sqrt x\_i$ Transformed amino acid vs.

## protein category data

barplot(percent\_aa,

main = "Mean % A.A.Composition Of Squareroot Transformed Data",

ylab = "% AA Composition",

ylim = c(0, 30),

col = colorRampPalette(brewer.pal(4, "Blues"))(3),

legend = T,

beside = T)

```

### Grouped bar chart of means for percent amino acid composition of Transformed Data; control & myoglobin categories

```{r echo=FALSE, message=FALSE, warning=FALSE, cache=TRUE}

data2 <- data.frame(ctrl\_means, mgb\_means)

percent\_aa2 <- as.matrix(t(100 \* data2))

barplot(percent\_aa2,

ylim = c(0, 30),

main = "Means of % A.A.Composition Of Squareroot Transformed Data",

ylab = "% AA Composition",

col = colorRampPalette(brewer.pal(4, "Blues"))(2),

legend = T,

beside = T)

```

### Boxplots of grand-means of the overall amino acid composition of square-root transformed data

```{r echo=FALSE, message=FALSE, warning=FALSE}

boxplot(c\_m\_TRANSFORMED[, 4:23],

main = "% AAC Vs Amino Acid Of Squareroot Transformed Data",

ylab = "% AAC",

xlab = "Amino Acid",

las = 1)

```

### Boxplots of amino acid compositions for control (only) of square-root transformed data

```{r echo=FALSE}

boxplot(ctrl\_set[, 4:23],

main = "Control, % AAC Vs Amino Acid Of Squareroot Transformed Data",

ylab = "% AAC",

xlab = "Amino Acid",

las = 1)

```

### Boxplots of amino acid compositions for myoglobin of square-root transformed Data(only) of square-root transformed data

```{r echo=FALSE}

boxplot(mgb\_set[, 4:23],

main = "Myoglobin, % AAC Vs Amino Acid Of Squareroot Transformed Data",

ylab = "% AAC",

xlab = "Amino Acid",

las = 1)

```

### Boxplots Of Length Of Polypeptides Of Transformed Data; Myoglobin, Control & Combined

```{r echo=FALSE}

ctrl\_totalaa <- ctrl\_set[, 2]

mgb\_totalaa <- mgb\_set[, 2]

grand\_totalaa <- c\_m\_TRANSFORMED[, 2]

data <- c(ctrl\_totalaa, mgb\_totalaa, grand\_totalaa)

boxplot(data,

ylim = c(0, 5000),

main = "Length of Polypeptides Of Squareroot Transformed Data",

ylab = "Length of Polypeptides",

xaxt = "n",

las = 2)

axis(1, at = 1:3, labels = c("Control", "Myoglobin", "Combined"))

```

### Coefficient of variance (CV) Of Transformed data

Standard deviations are sensitive to scale. Therefore I compare the normalized standard deviations. This normalized standard deviation is more commonly called the coefficient of variation (CV).

$$CV = \frac {\sigma (x)} {E [|x|]} ~~~ where ~~~ \sigma(x) \equiv \sqrt{ E[x - \mu]^2 }$$

$$CV ~~=~~ \frac{1}{\bar x} \cdot \sqrt{ \frac{1}{n-1} \sum^n\_{i=1} (x\_i - \bar x)^2}$$

### Plot of Coefficient Of Variance (CV)

```{r echo=FALSE}

AA\_var\_norm <- (apply(c\_m\_TRANSFORMED[, 4:23], 2, sd)) / AA\_ave

plot(AA\_var\_norm,

main = "Coefficient Of Variance Vs 20 Std AA Of Squareroot Transformed Data",

ylab = "Coefficient Of Variance (CV)",

xlab = "Amino Acid",

ylim = c(0, 1.5),

type = "b",

xaxt = "n")

axis(1, at = 1:20, labels = names(c\_m\_TRANSFORMED[, 4:23]))

```

```{r}

AA\_var\_norm

```

### Skewness of distributions Of Transformed Data

$$Skewness ~= E\left[ \left( \frac{X - \mu}{\sigma(x)} \right)^3 \right] ~~~~ where ~~~~ \sigma(x) \equiv \sqrt{ E[x - \mu]^2 }$$

$$Skewness ~= \frac { \frac{1}{n} \sum^n\_{i=1} (x\_i - \bar x)^3 } { \left( \sqrt{ \frac{1}{n-1} \sum^n\_{i=1} (x\_i - \bar x)^2 } \right) ^ {3}}$$

- Skewness values for each A.A. by Class of square-root transformed data

```{r echo=FALSE}

AA\_skewness <- (apply(c\_m\_TRANSFORMED[, 4:23], 2, e1071::skewness))

plot(AA\_skewness,

main = "Skewness of Amino Acids, squareroot transformed data",

sub = "(Red bar at Skew=2.0 indicates all values good.)",

ylab = "Skewness",

xlab = "Amino Acid",

type = "b",

ylim = c(-0.5, 2.3),

xaxt = "n")

axis(1, at = 1:20, labels = names(c\_m\_TRANSFORMED[, 4:23]))

abline(h = 2.0, col = "red")

```

```{r}

AA\_skewness

```

### QQ Plots of 20 amino acids of Transformed data

```{r echo=FALSE}

AA <- c("A", "C", "D", "E", "F", "G", "H", "I", "K", "L",

"M", "N", "P", "Q", "R", "S", "T", "V", "W", "Y")

par(mfrow=c(2,2))

for (i in 4:23) {

qqnorm(c\_m\_TRANSFORMED[[i]], main = AA[[i - 3]])

qqline(c\_m\_TRANSFORMED[[i]], col = "red")

}

```

### Determine coefficients of correlation of Transformed Data

An easily interpretable test is a correlation 2D-plot for investigating multicollinearity or feature reduction. Fewer attributes "means decreased computational time and complexity. Secondly, if two predictors are highly correlated, this implies that they measure the same underlying information. Removing one should not compromise the performance of the model and might lead to a more parsimonious and interpretable model. Third, some models can be crippled by predictors with degenerate distributions".[^21]

[^21]:Max Kuhn and Kjell Johnson, Applied Predictive Modeling, Springer Publishing, 2018, P.43

Pearson's correlation coefficient:

$$\rho\_{x,y} = \frac {E \left[(X - \mu\_x)(X - \mu\_y) \right]} {\sigma\_x \sigma\_y}$$

$$r\_{xy} = \frac {\sum^n\_{i=1} (x\_i - \bar x)(y\_1 - \bar y)} { {\sqrt {\sum^n\_{i=1} (x\_i - \bar x)^2 }} {\sqrt {\sum^n\_{i=1} (y\_i - \bar y)^2 }} }$$

```{r echo=FALSE}

c\_m\_corr\_mat <- cor(c\_m\_TRANSFORMED[, c(2, 4:23)],

method = "p") # "p": Pearson test for continous variables

corrplot(abs(c\_m\_corr\_mat),

title = "Correlation Plot Of Transformed Features",

method = "square",

type = "lower",

tl.pos = "d",

cl.lim = c(0, 1),

addgrid.col = "lightgrey",

cl.pos = "b", # Color legend position bottom.

order = "FPC", # "FPC" = first principal component order.

mar = c(1, 2, 1, 2),

tl.col = "black")

```

```{r}

c\_m\_corr\_mat["T", "N"]

```

\*\*No values in the correlation matrix meet the 0.75 cut off criteria for problems.\*\*

### Boruta - dimensionality reduction of Transformed data

```{r echo=FALSE}

c\_m\_class\_20 <- c\_m\_TRANSFORMED[, -c(2, 3)] # Remove TotalAA & PID

Class <- as.factor(c\_m\_class\_20$Class) # Convert 'Class' To Factor

```

\*\*Perform Boruta search\*\*

NOTE: \*mcAdj = TRUE\*: If True, multiple comparisons will be adjusted using the Bonferroni method to calculate p-values. Therefore, $p\_i \leq \frac {\alpha} {m}$ where $\alpha$ is the desired p-value and $m$ is the total number of null hypotheses.

```{r cache=TRUE}

set.seed(1000)

registerDoMC(cores = 3) # Start multi-processor mode

start\_time <- Sys.time() # Start timer

boruta\_output <- Boruta(Class ~ .,

data = c\_m\_class\_20[, -1],

mcAdj = TRUE, # See Note above.

doTrace = 1) # doTrace = 1, represents non-verbose mode.

registerDoSEQ() # Stop multi-processor mode

end\_time <- Sys.time() # End timer

end\_time - start\_time # Display elapsed time

```

### Plot variable importance

```{r}

plot(boruta\_output,

cex.axis = 1,

las = 2,

ylim = c(-5, 50),

main = "Variable Importance (Bigger=Better)")

```

### Variable importance scores

```{r message=FALSE}

roughFixMod <- TentativeRoughFix(boruta\_output)

imps <- attStats(roughFixMod)

imps2 <- imps[imps$decision != "Rejected", c("meanImp", "decision")]

meanImps <- imps2[order(-imps2$meanImp), ] # descending sort

knitr::kable(meanImps,

full\_width = F,

position = "left",

caption = "Mean Importance Scores & Decision")

```

```{r eval=FALSE, include=FALSE}

## Plot importance history

plotImpHistory(boruta\_output)

```

### Conclusion for Boruta random forest test {-}

- All features are essential. None should be dropped.

## Overall EDA Conclusion

It was determined earlier that three amino acids (C, F, I) from the single amino acid percent composition should be transformed by using the square root function. The square root transformation lowered the skewness from greater than 2 in all cases to {-0.102739 $\leq$ skew after transformation $\leq$ 0.3478132}.

| Amino Acid | Initial skewness | Skew after square root transform |

| :--------------- | :--------------: | :------------------------------: |

| C, Cysteine | 2.538162 | 0.347813248 |

| F, Phenolalanine | 2.128118 | -0.102739748 |

| I, Isoleucine | 2.192145 | 0.293474879 |

The transformations of the three amino acids (C, F, I) did not appreciably change any critical measures, such as the feature importance derived from the Boruta random forest feature selection work. Nor did the transformations of C, F, I appreciably change the correlation coefficient matrix. Therefore the transformed data will be used throughout this experiment.

Boruta, which is used for dimensionality reduction of Transformed data, showed that all dependent features are essential for the generation of a Decision Tree. I believe that this would imply that given that the Random Forest approach will be used, it would wise to keep all features for that model test and throughout the generation of other models. All features have decisive mean importance, which is generated by a Gini calculation.

Regarding the coefficients of correlation of the Transformed dataset, there are no examples of coefficients that are greater than or equal to 0.75; therefore, this implies that no features are collinear.