Exploratory Data Analysis

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

Four Step Analysis

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

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

This four step process is an oversimplification of a much more detailed process which is outlined below. I have derived this process as an amalgamation of several other approaches.

Useful Guides for Exploratory Data Analysis

The hybrid process which I use to summarize the amino acid dataset is based on three sets of guidelines;

1. NIST Handbook of Statistics,[2]
2. Roger Peng's booklet on 'Exploratory Data Analysis with R,' [3]
3. 'Exploratory Data Analysis Using R', by Ronald K. Pearson.[4]

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 skew values greater than 2.0 are found, can a transformation be used for normalization?
   1.2 If so, what transformation should be used?
2. Can **Feature Selection** be used and which procedures are appropriate?
   2.1 Can the Random Forest technique known as Boruta[5], be used for feature importance or reduction?
   2.2 Will coefficients of correlation (R) find collinearity and reduce the number of features?
   2.3 Will principle 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?[6]

[1] https://en.wikipedia.org/wiki/Exploratory_data_analysis
[2] https://www.itl.nist.gov/div898/handbook/
[3] Peng, Roger, Exploratory Data Analysis with R, https://leanpub.com/exdata, 2016
[4] Ronald Pearson, 'Exploratory Data Analysis Using R', P.11, CRC Press, ISBN:9781138480605, 2018
[5] Miron Kursa, Witold Rudnicki, Feature Selection with the Boruta Package, DOI:10.18637/jss.v036.i11, 2010
[6] Ronald Pearson, 'Exploratory Data Analysis Using R', P.11, CRC Press, 2018

Begin Exploratory Data Analysis

Import libraries

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

for (i in Libraries) {
    library(i, character.only = TRUE)
}
```

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 command line interface followed by the command `less`.
2. Check for binary instead of ASCII and bad Unicode.

Inspect RAW dataframe structure, `str()`

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 2340 obs. of  23 variables:
##  $ Class  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ TotalAA: num  226 221 624 1014 699 ...
##  $ PID    : chr  "C1" "C2" "C3" "C4" ...
##  $ A      : num  0.2655 0.2081 0.0433 0.0661 0.0644 ...
##  $ C      : num  0 0 0.00962 0.01381 0.03577 ...
##  $ D      : num  0.00442 0.00452 0.04647 0.06114 0.02861 ...
##  $ E      : num  0.031 0.0271 0.0833 0.074 0.0472 ...
##  $ F      : num  0.00442 0.00452 0.02564 0.02959 0.06295 ...
##  $ G      : num  0.0708 0.0769 0.0817 0.07 0.0443 ...
##  $ H      : num  0 0 0.0176 0.0187 0.0157 ...
##  $ I      : num  0.00885 0.0181 0.03045 0.04734 0.0701 ...
##  $ K      : num  0.28761 0.27602 0.00962 0.12426 0.05579 ...
##  $ L      : num  0.0442 0.0452 0.0577 0.0888 0.1359 ...
##  $ M      : num  0.00442 0.00452 0.01442 0.02465 0.02289 ...
##  $ N      : num  0.0177 0.0136 0.0641 0.0355 0.0558 ...
##  $ P      : num  0.0841 0.0995 0.0449 0.0434 0.0472 ...
##  $ Q      : num  0.00442 0.00905 0.04327 0.03353 0.02861 ...
##  $ R      : num  0.0133 0.0181 0.1202 0.0325 0.0415 ...
##  $ S      : num  0.0575 0.0724 0.1875 0.0838 0.0787 ...
##  $ T      : num  0.0531 0.0633 0.0625 0.0414 0.0744 ...
##  $ V      : num  0.0442 0.0543 0.0385 0.0671 0.0458 ...
##  $ W      : num  0 0 0.00481 0.01282 0.00715 ...
##  $ Y      : num  0.00442 0.00452 0.01442 0.03156 0.0372 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Class = col_double(),
##   ..   TotalAA = col_double(),
```

```
##   ..   PID = col_character(),
##   ..   A = col_double(),
##   ..   C = col_double(),
##   ..   D = col_double(),
##   ..   E = col_double(),
##   ..   F = col_double(),
##   ..   G = col_double(),
##   ..   H = col_double(),
##   ..   I = col_double(),
##   ..   K = col_double(),
##   ..   L = col_double(),
##   ..   M = col_double(),
##   ..   N = col_double(),
##   ..   P = col_double(),
##   ..   Q = col_double(),
##   ..   R = col_double(),
##   ..   S = col_double(),
##   ..   T = col_double(),
##   ..   V = col_double(),
##   ..   W = col_double(),
##   ..   Y = col_double()
##   .. )
```

Check RAW data `head` & `tail`

```
head(c_m_RAW_AAC, n = 2)
```

```
## # A tibble: 2 x 23
##   Class TotalAA PID       A     C       D      E       F      G     H       I
##   <dbl>   <dbl> <chr> <dbl> <dbl>   <dbl>  <dbl>   <dbl>  <dbl> <dbl>   <dbl>
## 1     0     226 C1    0.265     0 0.00442 0.0310 0.00442 0.0708     0 0.00885
## 2     0     221 C2    0.208     0 0.00452 0.0271 0.00452 0.0769     0 0.0181
## # ... with 12 more variables: K <dbl>, L <dbl>, M <dbl>, N <dbl>, P <dbl>,
## #   Q <dbl>, R <dbl>, S <dbl>, T <dbl>, V <dbl>, W <dbl>, Y <dbl>
```

```
tail(c_m_RAW_AAC, n = 2)
```

```
## # A tibble: 2 x 23
##   Class TotalAA PID        A       C      D      E      F      G      H      I
##   <dbl>   <dbl> <chr>  <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1     1     335 M1123 0.0567 0.00299 0.0537 0.0716 0.0507 0.0507 0.0388 0.0776
## 2     1      43 M1124 0.0698 0       0.116  0.116  0.0930 0.0465 0      0.0233
## # ... with 12 more variables: K <dbl>, L <dbl>, M <dbl>, N <dbl>, P <dbl>,
## #   Q <dbl>, R <dbl>, S <dbl>, T <dbl>, V <dbl>, W <dbl>, Y <dbl>
```

Check RAW data types

```
is.data.frame(c_m_RAW_AAC)
```

```
## [1] TRUE
```

```
class(c_m_RAW_AAC$Class)        # Col 1
```

```
## [1] "numeric"
```

```
class(c_m_RAW_AAC$TotalAA)      # Col 2
```

```
## [1] "numeric"
```

```
class(c_m_RAW_AAC$PID)          # Col 3
```

```
## [1] "character"
```

```
class(c_m_RAW_AAC$A)            # Col 4
```

```
## [1] "numeric"
```

Check RAW dataframe dimensions

```
dim(c_m_RAW_AAC)
```

```
## [1] 2340   23
```

Check RAW for missing values

- **No missing values found.**

```
apply(is.na(c_m_RAW_AAC), 2, which)
```

```
## integer(0)
```

```
# 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

```
##
##    0    1
## 1216 1124
```

Numerical summary of RAW features

```
##      Class           TotalAA           PID                  A
##  Min.   :0.0000   Min.   :   2.0   Length:2340        Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.: 109.8   Class :character   1st Qu.:0.05108
##  Median :0.0000   Median : 154.0   Mode  :character   Median :0.07364
##  Mean   :0.4803   Mean   : 353.8                      Mean   :0.07835
##  3rd Qu.:1.0000   3rd Qu.: 407.0                      3rd Qu.:0.10261
##  Max.   :1.0000   Max.   :4660.0                      Max.   :0.28000
##        C                  D                 E                 F
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.03401   1st Qu.:0.05435   1st Qu.:0.03801
##  Median :0.007034   Median :0.05195   Median :0.07143   Median :0.04545
##  Mean   :0.011970   Mean   :0.04900   Mean   :0.07451   Mean   :0.05135
##  3rd Qu.:0.020408   3rd Qu.:0.06567   3rd Qu.:0.09091   3rd Qu.:0.05501
##  Max.   :0.159420   Max.   :0.17647   Max.   :0.50000   Max.   :0.37500
```

```
##        G                H                I                K
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.04544   1st Qu.:0.01324   1st Qu.:0.04348   1st Qu.:0.05797
##  Median :0.06394   Median :0.02297   Median :0.05992   Median :0.08182
##  Mean   :0.06193   Mean   :0.02890   Mean   :0.06839   Mean   :0.08386
##  3rd Qu.:0.08625   3rd Qu.:0.04095   3rd Qu.:0.08216   3rd Qu.:0.12081
##  Max.   :0.36364   Max.   :0.13333   Max.   :0.50000   Max.   :0.28761
##        L                M                N                P
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.07480   1st Qu.:0.01087   1st Qu.:0.01948   1st Qu.:0.02464
##  Median :0.09136   Median :0.01948   Median :0.04145   Median :0.03401
##  Mean   :0.09313   Mean   :0.01949   Mean   :0.04228   Mean   :0.03825
##  3rd Qu.:0.11688   3rd Qu.:0.02721   3rd Qu.:0.05788   3rd Qu.:0.04772
##  Max.   :0.25000   Max.   :0.11111   Max.   :0.12563   Max.   :0.20635
##        Q                R                S                T
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.02212   1st Qu.:0.01476   1st Qu.:0.04348   1st Qu.:0.03247
##  Median :0.03598   Median :0.03896   Median :0.05564   Median :0.05194
##  Mean   :0.03342   Mean   :0.03818   Mean   :0.06191   Mean   :0.04838
##  3rd Qu.:0.04545   3rd Qu.:0.05370   3rd Qu.:0.06964   3rd Qu.:0.06522
##  Max.   :0.18182   Max.   :0.24324   Max.   :0.22619   Max.   :0.18750
##        V                W                Y
##  Min.   :0.00000   Min.   :0.000000   Min.   :0.00000
##  1st Qu.:0.04575   1st Qu.:0.001899   1st Qu.:0.01463
##  Median :0.05844   Median :0.011492   Median :0.02865
##  Mean   :0.06512   Mean   :0.012327   Mean   :0.03644
##  3rd Qu.:0.07405   3rd Qu.:0.017889   3rd Qu.:0.04564
##  Max.   :0.20000   Max.   :0.133333   Max.   :0.14286
```

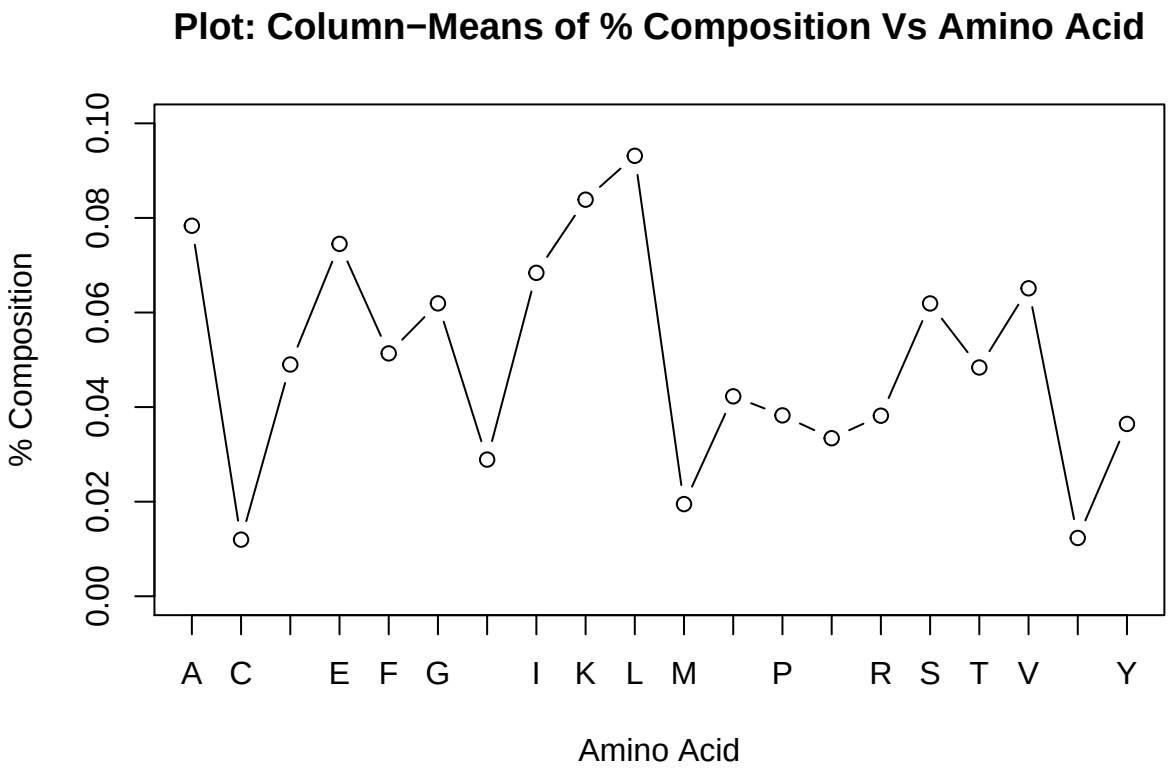
Visualize RAW Data With Descriptive Statistics

Formulas for mean:

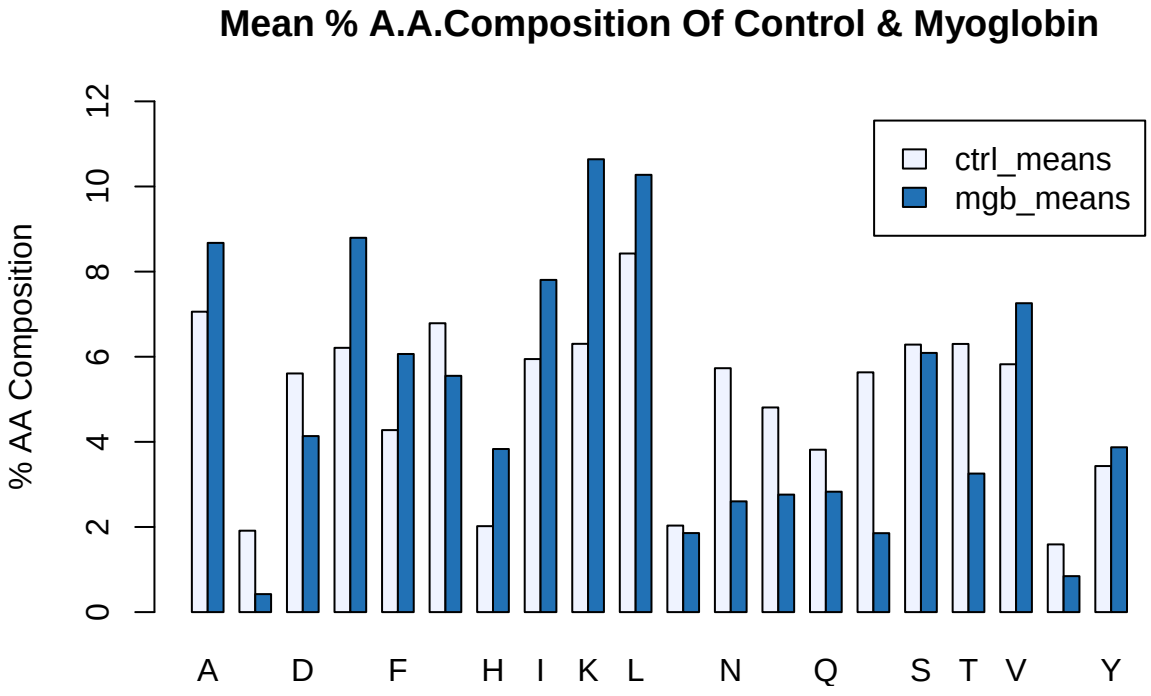
$$E[X] = \sum_{i=1}^{n} x_i p_i \quad ; \qquad \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Scatter plot of means of *Myoglobin-Control* amino acid composition of `c_m_RAW_AAC` dataframe

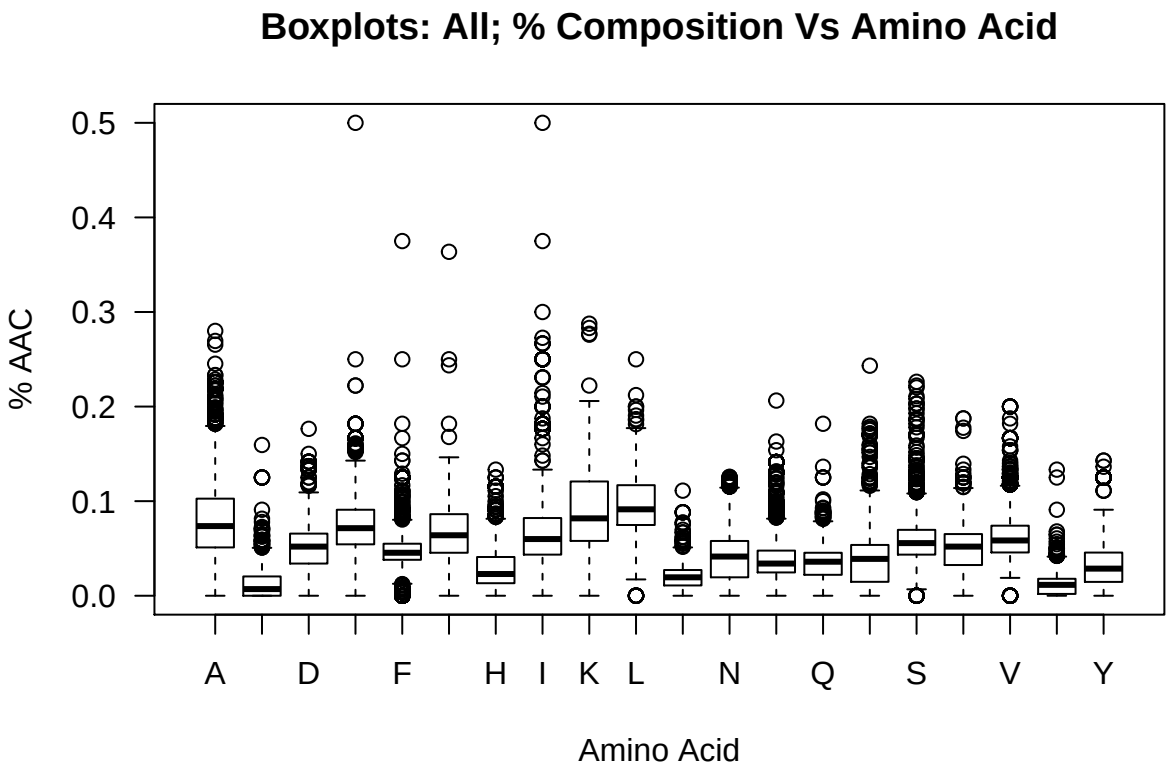
- This plot shows the means for each feature (column-means) in the dataset. The means represent the ungrouped or total of all proteins (where n = 2340) versus AA type.



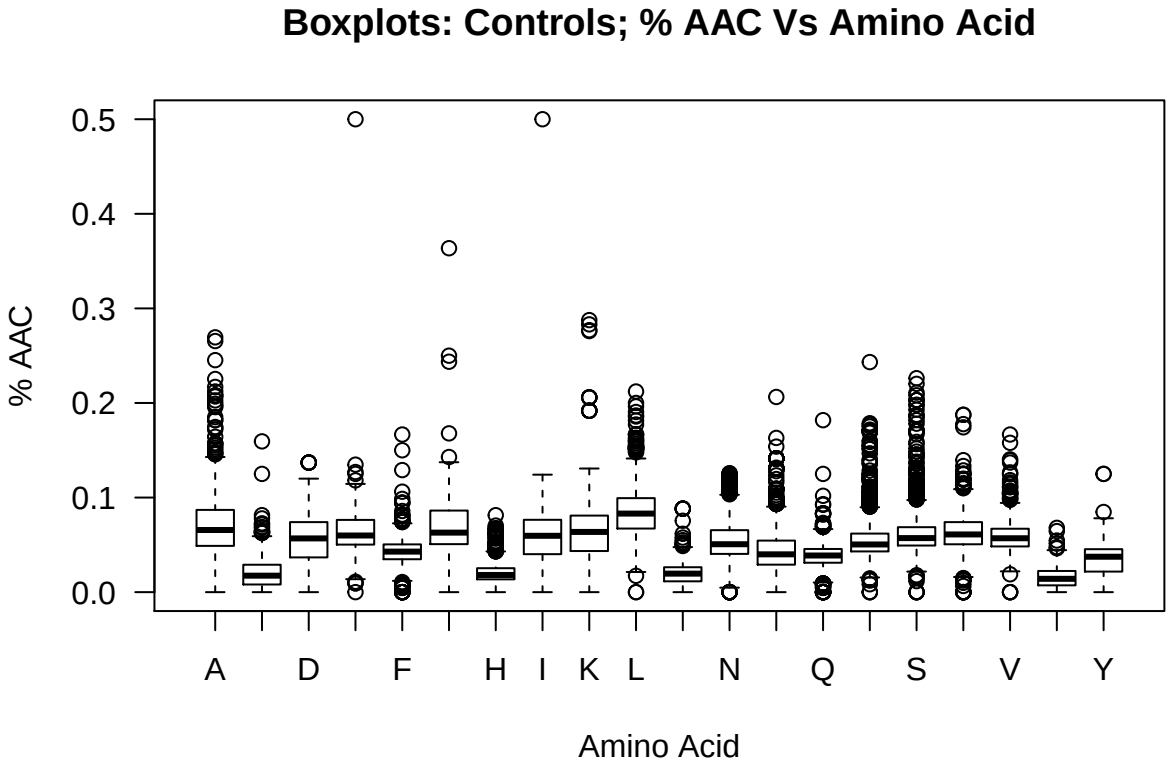
Means of percent amino acid composition of control & myoglobin categories, RAW data



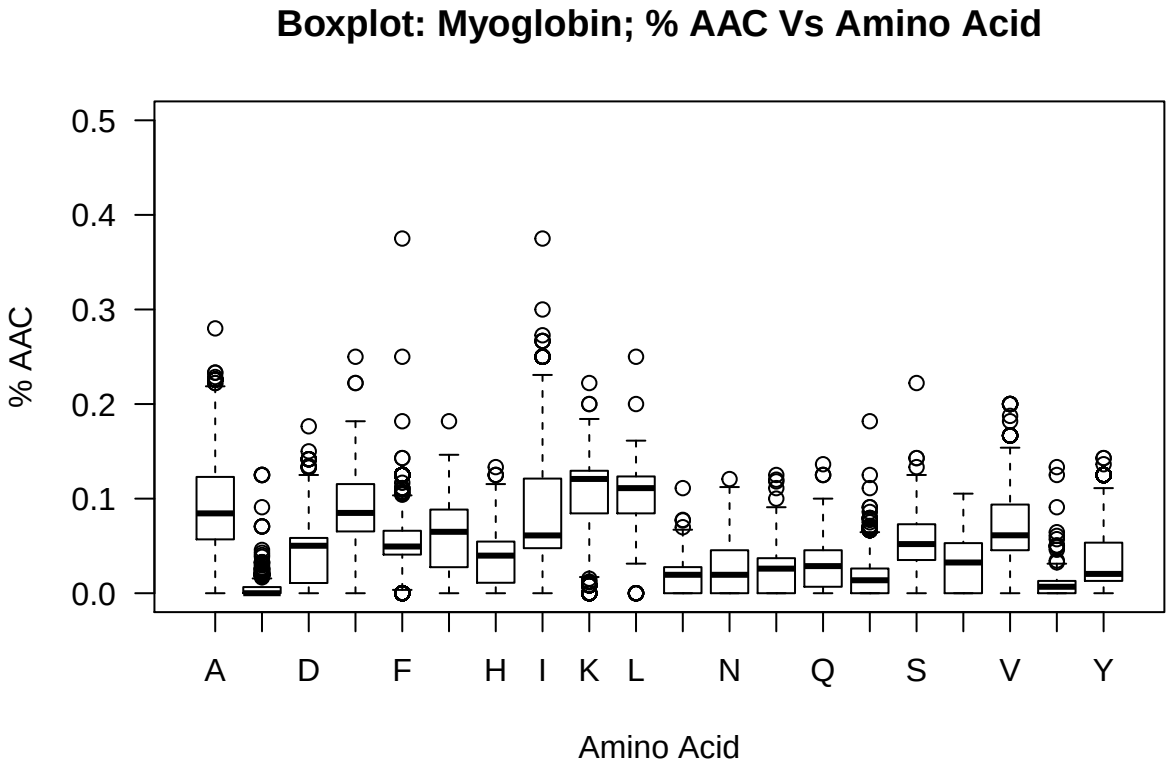
Boxplots of grand-means of overall amino acid composition, RAW data

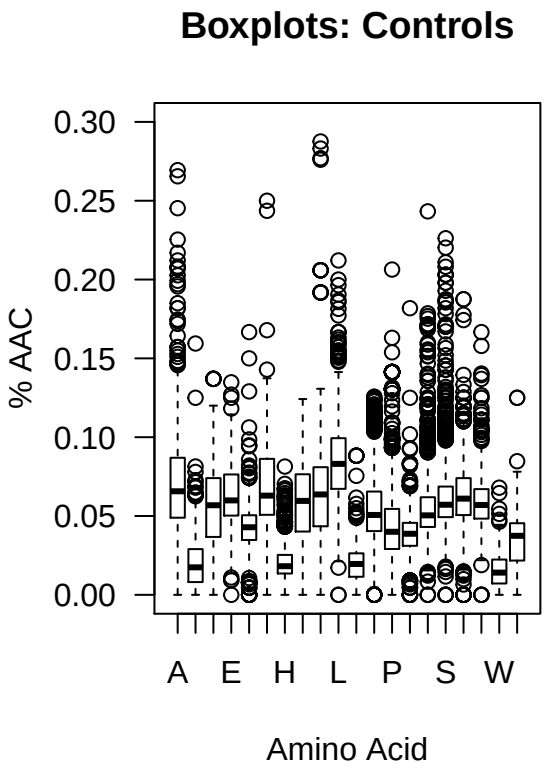


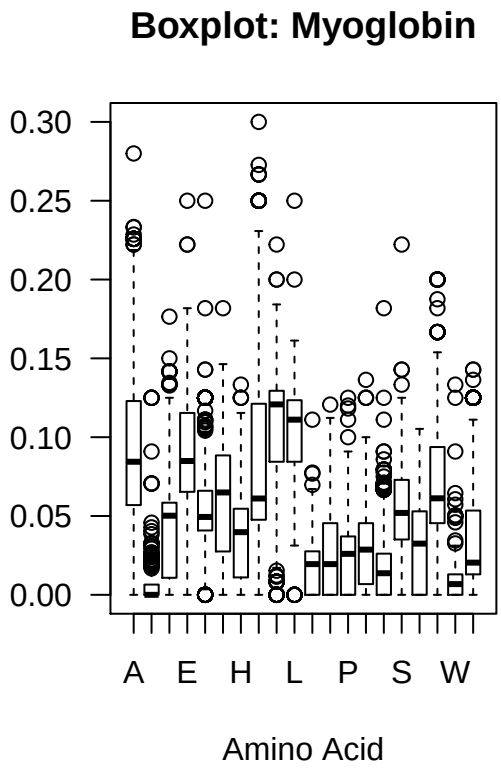
Boxplots of amino acid compositions for control (only), RAW data



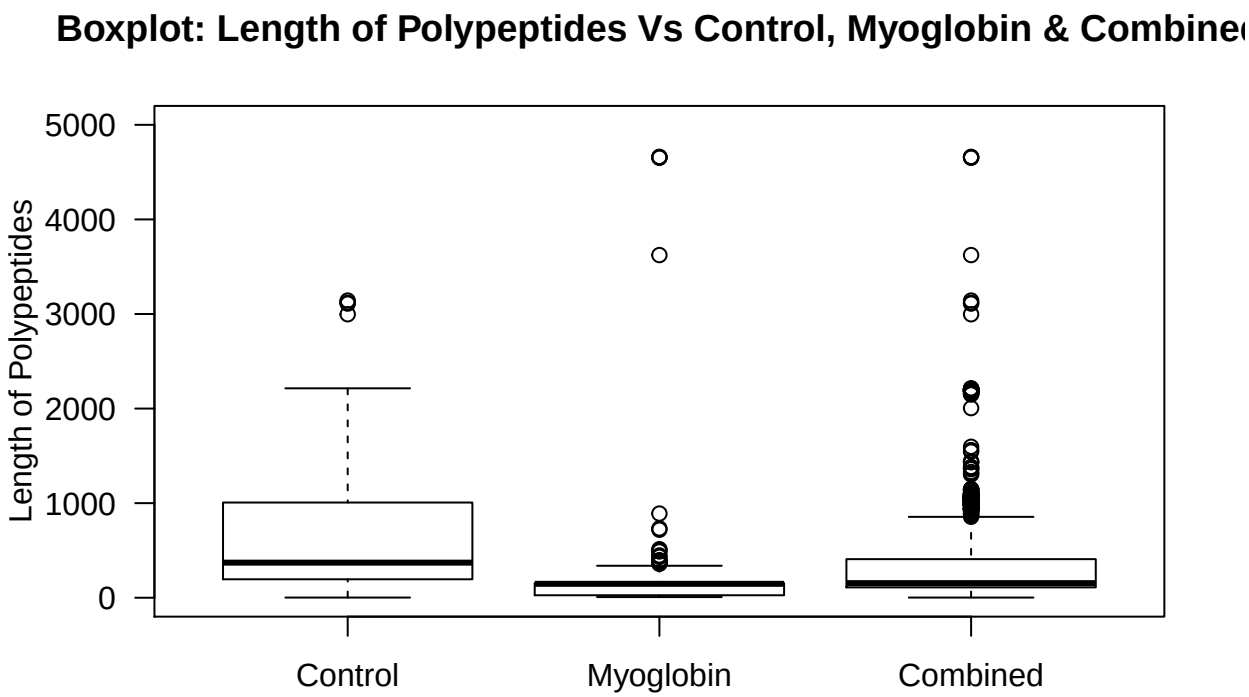
Boxplots of amino acid compositions for myoglobin (only), RAW data







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

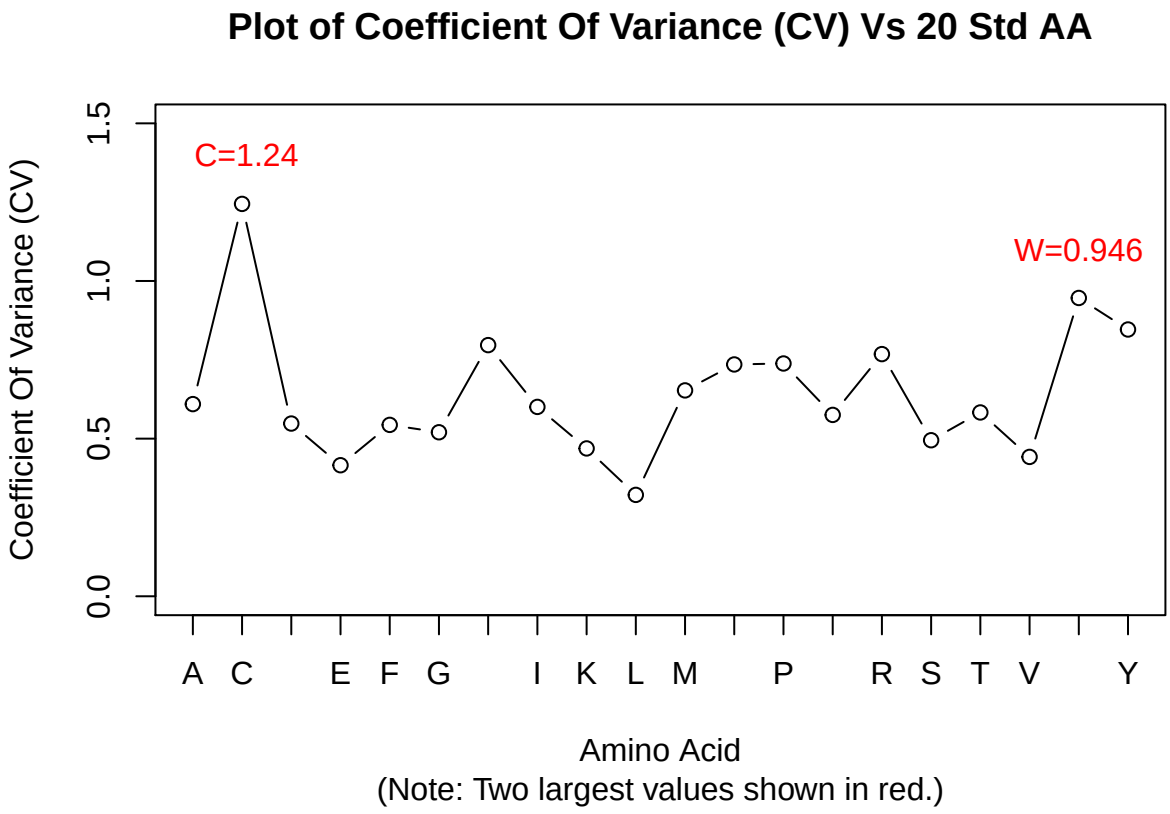


Plot of normalized standard deviations / coefficient of variance (CV), RAW data

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

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

$$CV = \frac{1}{\bar{x}} \cdot \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$




```
AA_var_norm
```

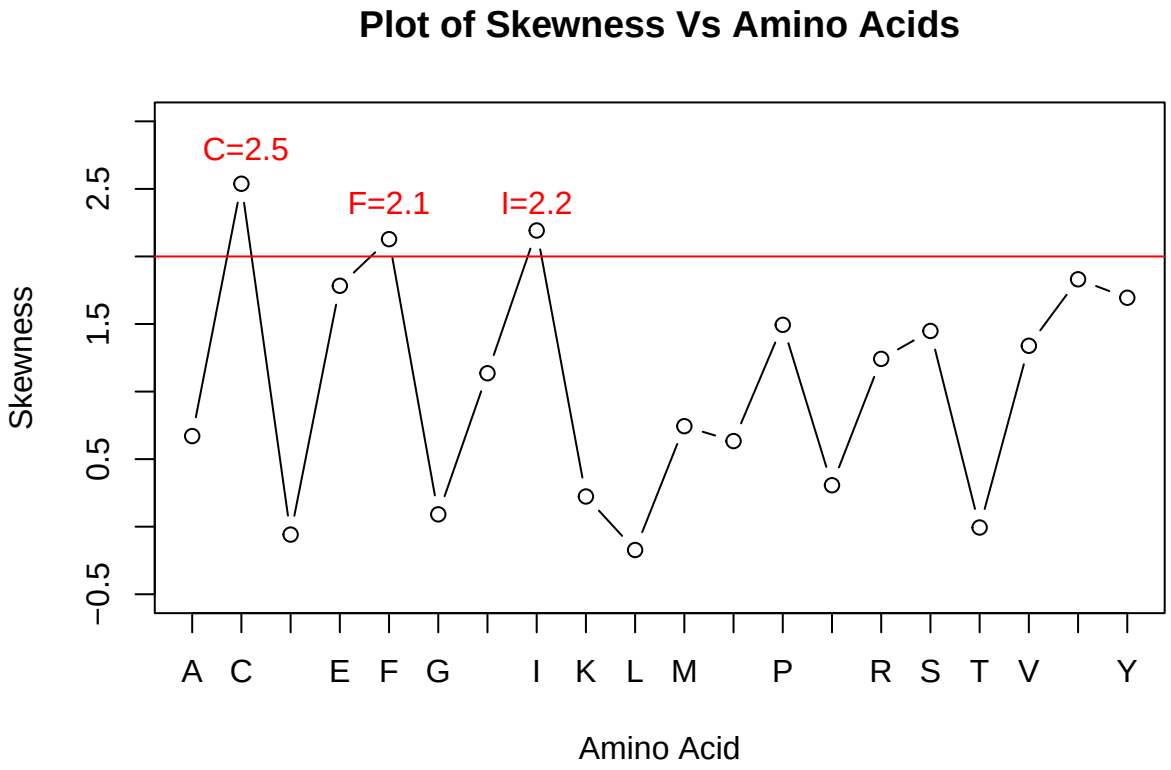
```
##         A         C         D         E         F         G         H         I
## 0.6095112 1.2444944 0.5478540 0.4156102 0.5436243 0.5201625 0.7966296 0.6005962
##         K         L         M         N         P         Q         R         S
## 0.4689544 0.3215591 0.6529752 0.7352478 0.7383244 0.5752622 0.7680977 0.4948690
##         T         V         W         Y
## 0.5830352 0.4420595 0.9461276 0.8461615
```

Skewness of distributions, RAW data

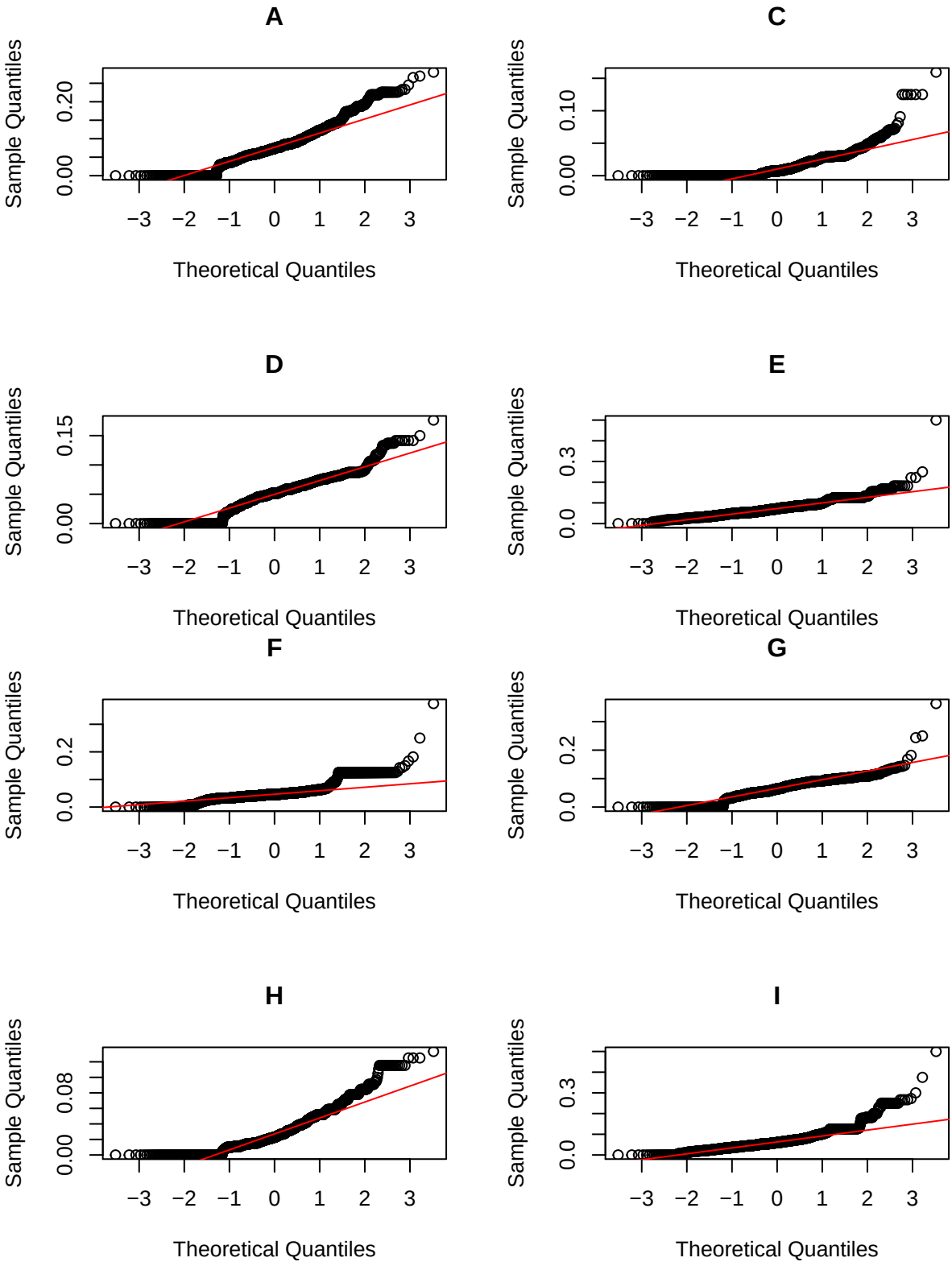
$$Skewness \ = E\left[\left(\frac{X-\mu}{\sigma(x)}\right)^3\right] \quad where \quad \sigma(x) \equiv \sqrt{E[x-\mu]^2}$$

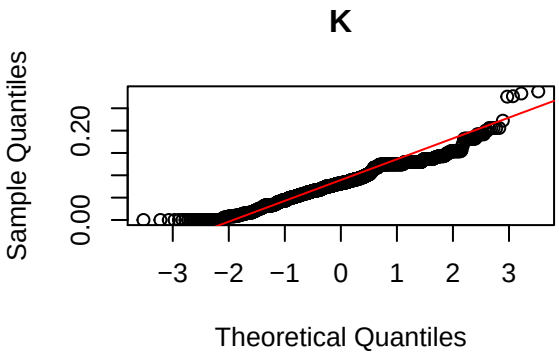
$$Skewness \ = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i-\bar{x})^3}{\left(\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2}\right)^3}$$

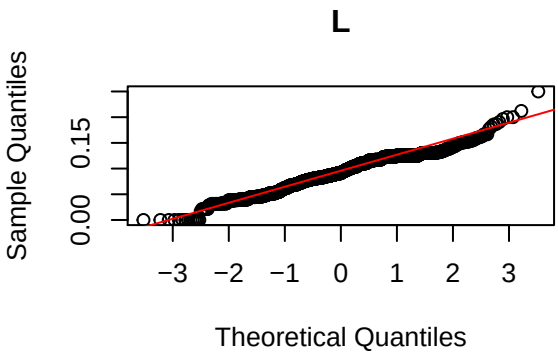
Skewness values for each A.A. will be determined in totality

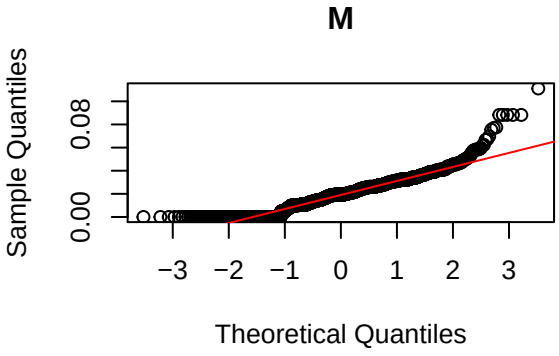

Plot of Skewness Vs Amino Acids
C=2.5
F=2.1
I=2.2
Skewness
2.5
1.5
0.5
−0.5
A C E F G I K L M P R S T V Y
Amino Acid

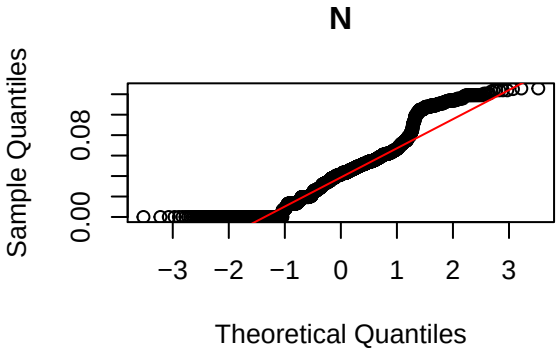
QQ-Plots of 20 amino acids, RAW data

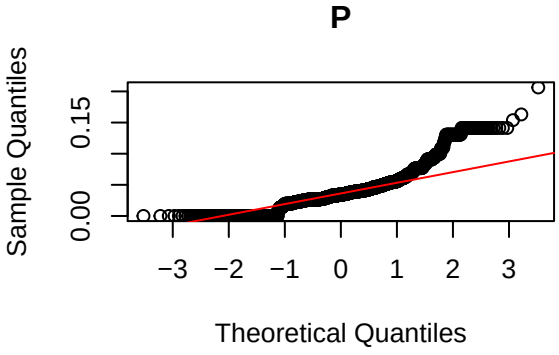


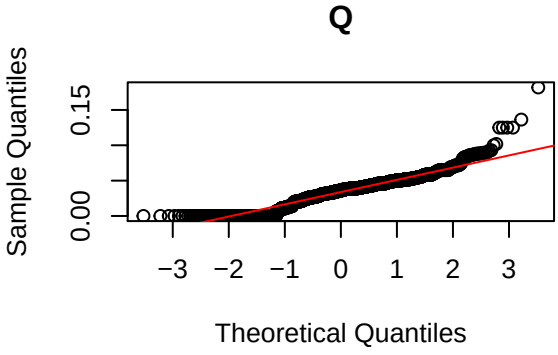

K
Sample Quantiles
0.00
0.20
−3
−2
−1
0
1
2
3
Theoretical Quantiles

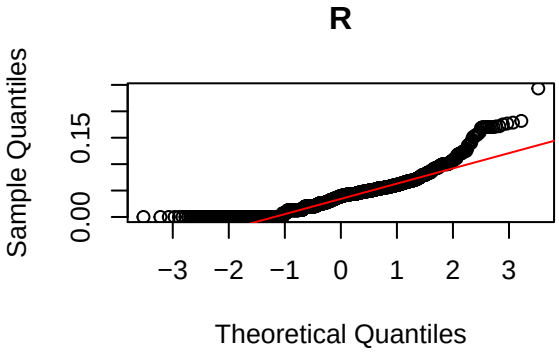

L
Sample Quantiles
0.00
0.15
−3
−2
−1
0
1
2
3
Theoretical Quantiles

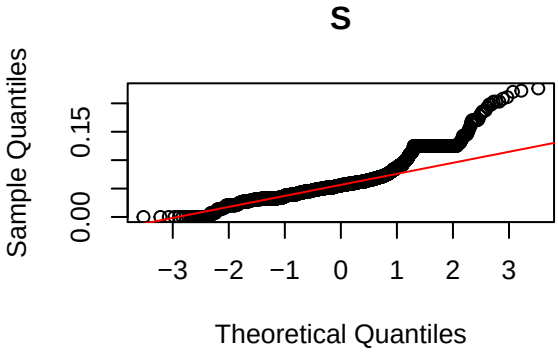

M
Sample Quantiles
0.00
0.08
−3
−2
−1
0
1
2
3
Theoretical Quantiles

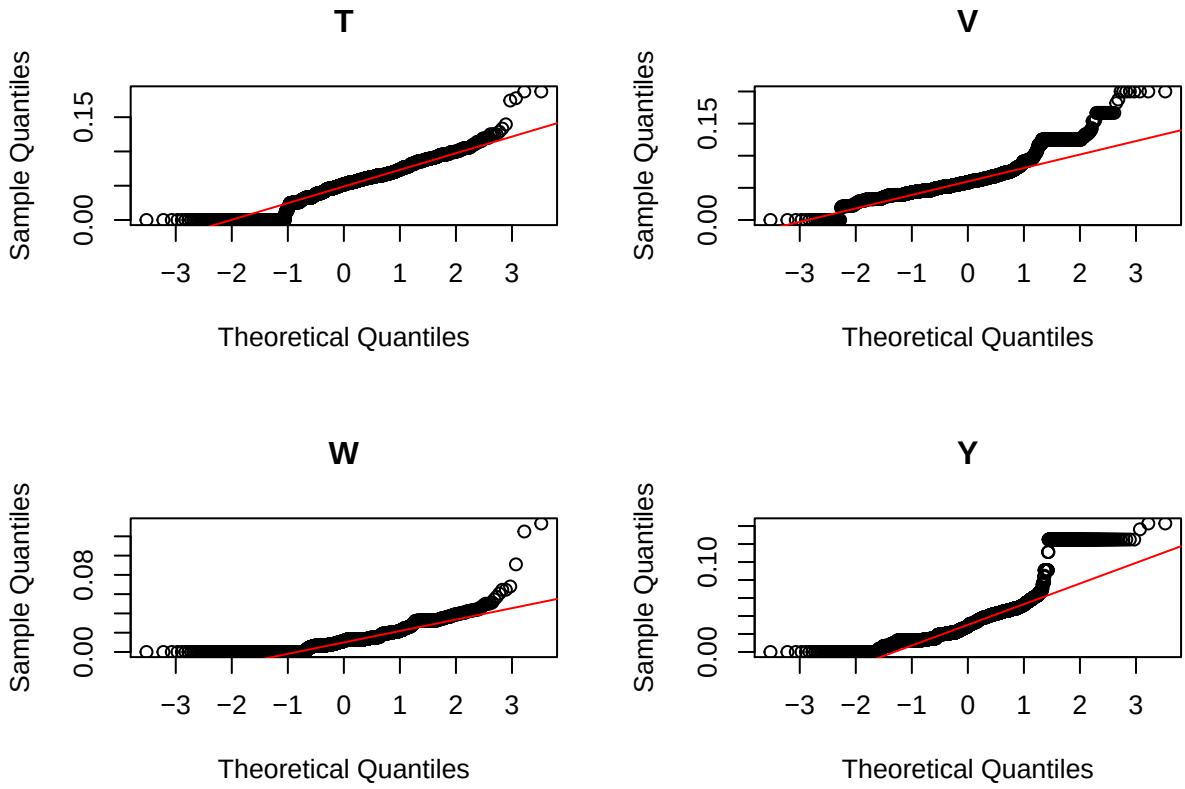

N
Sample Quantiles
0.00
0.08
−3
−2
−1
0
1
2
3
Theoretical Quantiles


P
Sample Quantiles
0.00
0.15
−3
−2
−1
0
1
2
3
Theoretical Quantiles


Q
Sample Quantiles
0.00
0.15
−3
−2
−1
0
1
2
3
Theoretical Quantiles


R
Sample Quantiles
0.00
0.15
−3
−2
−1
0
1
2
3
Theoretical Quantiles


S
Sample Quantiles
0.00
0.15
−3
−2
−1
0
1
2
3
Theoretical Quantiles



Determine coefficients of correlation, RAW data

An easily interpretable test, is correlation 2D-plot for investigating multicollinearity or feature reduction. It is clear that fewer attributes "means decreased computational time and complexity. Secondly, if two predictors are highly correlated, this implies that they are measuring 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".[7]

Pearson's correlation coefficient:

$$\rho_{x,y} = \frac{E\left[(X - \mu_x)(X - \mu_y)\right]}{\sigma_x \sigma_y}$$

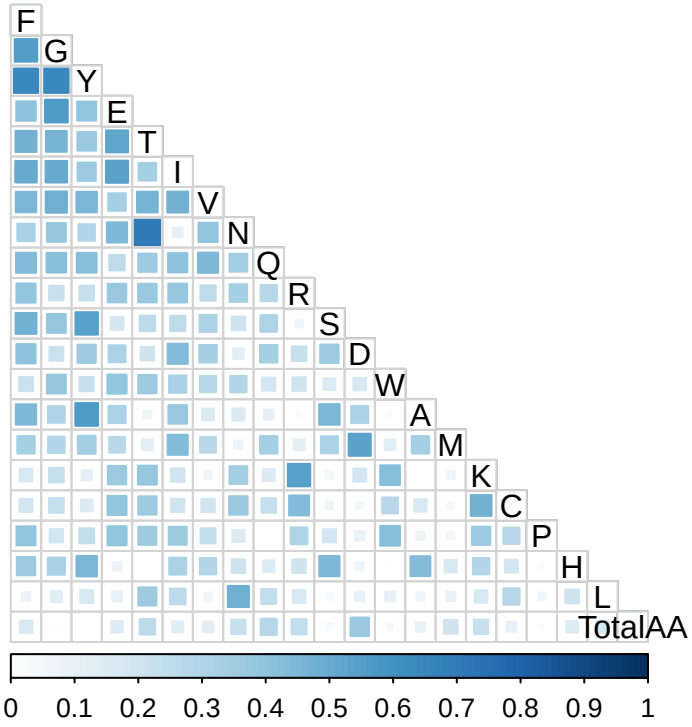
$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_1 - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

```
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.
```

[7] "Applied Predictive Modeling", Max Kuhn and Kjell Johnson, Springer Publishing, 2018, P.43

```
        mar = c(1, 2, 1, 2),
        tl.col = "black")
```

Correlation Plot Of AAC Features




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

1. Maximum value of Correlation between T & N.

```
## [1] 0.7098085
```

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 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.[8]
   3. Determine if the two predictors associated with the largest 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 larger average correlation, remove it; otherwise, remove predictor B.
   6. Repeat Steps 2–4 until no absolute correlations are above the threshold.

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

Boruta - dimensionality reduction, RAW data

```
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
```

Perform Boruta search

NOTE: *mcAdj = TRUE*, If True, multiple comparisons will be adjusted using the Bonferroni method to ca

```
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.
```

```
## After 11 iterations, +17 secs:
```

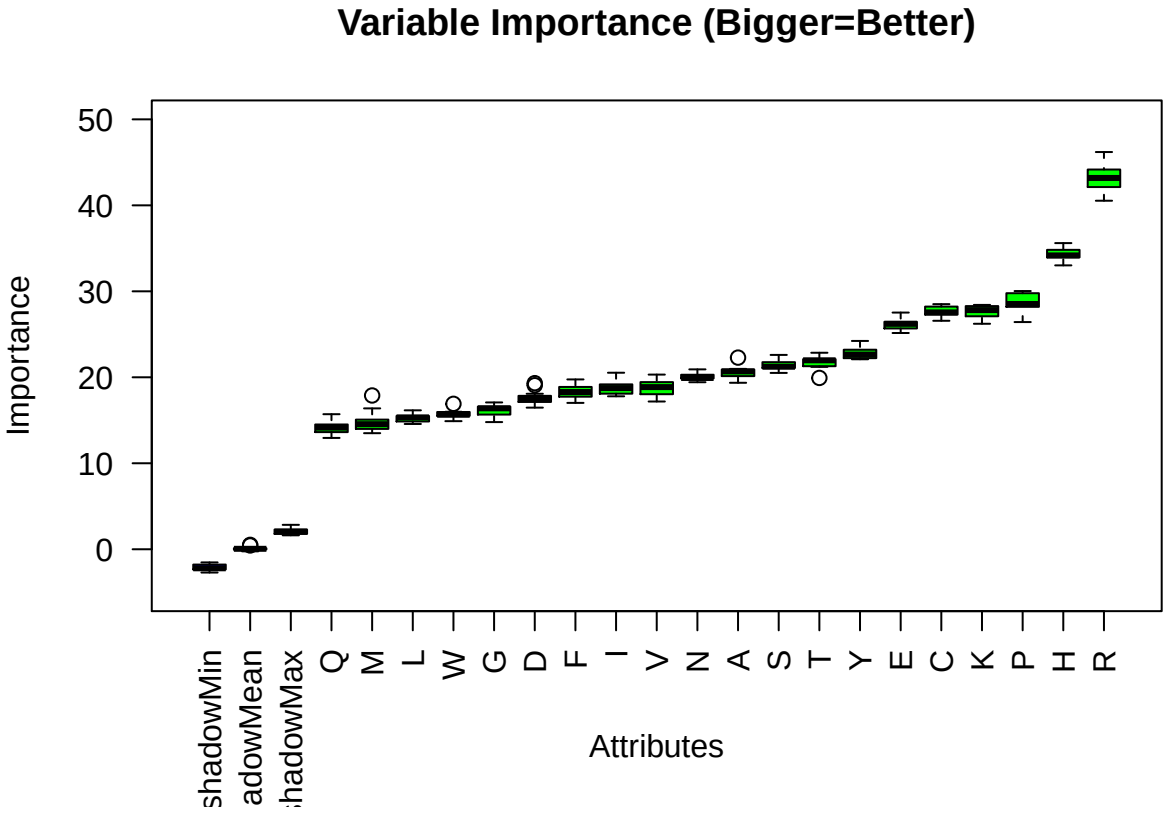
```
##  confirmed 20 attributes: A, C, D, E, F and 15 more;
```

```
##  no more attributes left.
```

```
registerDoSEQ()  # Stop multi-processor mode
end_time <- Sys.time()   # End timer
end_time - start_time    # Display elapsed time
```

```
## Time difference of 17.32416 secs
```

Plot variable importance



Variable importance scores

```
## Warning in TentativeRoughFix(boruta_output): There are no Tentative attributes!
## Returning original object.
```

Table 1: Mean Importance Scores & Decision

	meanImp	decision
R	43.18824	Confirmed
H	34.29757	Confirmed
P	28.70225	Confirmed
C	27.67710	Confirmed
K	27.60808	Confirmed
E	26.18884	Confirmed
Y	22.85337	Confirmed
T	21.67689	Confirmed
S	21.43716	Confirmed
A	20.53089	Confirmed
N	20.09681	Confirmed
V	18.77054	Confirmed
I	18.76492	Confirmed
F	18.31240	Confirmed
D	17.64592	Confirmed
G	16.15461	Confirmed
W	15.74107	Confirmed
L	15.27767	Confirmed
M	14.82861	Confirmed

	meanImp	decision
Q	14.13939	Confirmed

Conclusion for Boruta random forest test

```
All features are important. None should be dropped.
```

Conclusions For EDA, RAW data

It was determined that three amino acids (C, F, I) from the single amino acid percent composition should be 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

Exploratory Data Analysis Of `c_m_TRANSFORMED.csv`

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 was 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 in order 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

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

```
dim(c_m_TRANSFORMED)
```

```
## [1] 2340   23
```

Check Transformed for missing values

```
apply(is.na(c_m_TRANSFORMED), 2, which)
```

```
## integer(0)
```

```
No missing values found.
```

Count Transformed data for number of polypeptides per class

Number of polypeptides per Class:

- Class 0 = Control,
- Class 1 = Myoglobin

```
##
##    0    1
## 1216 1124
```

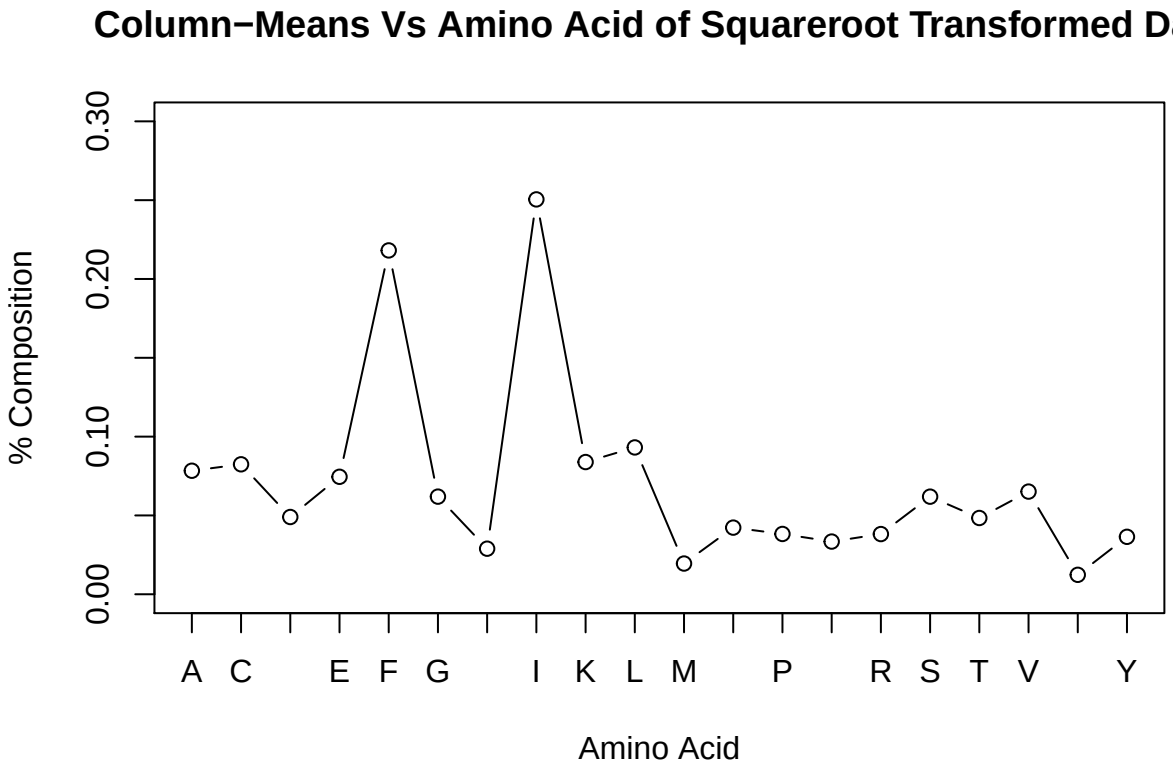
Visualization of Transformed Data Descriptive Statistics

Formulas for mean:

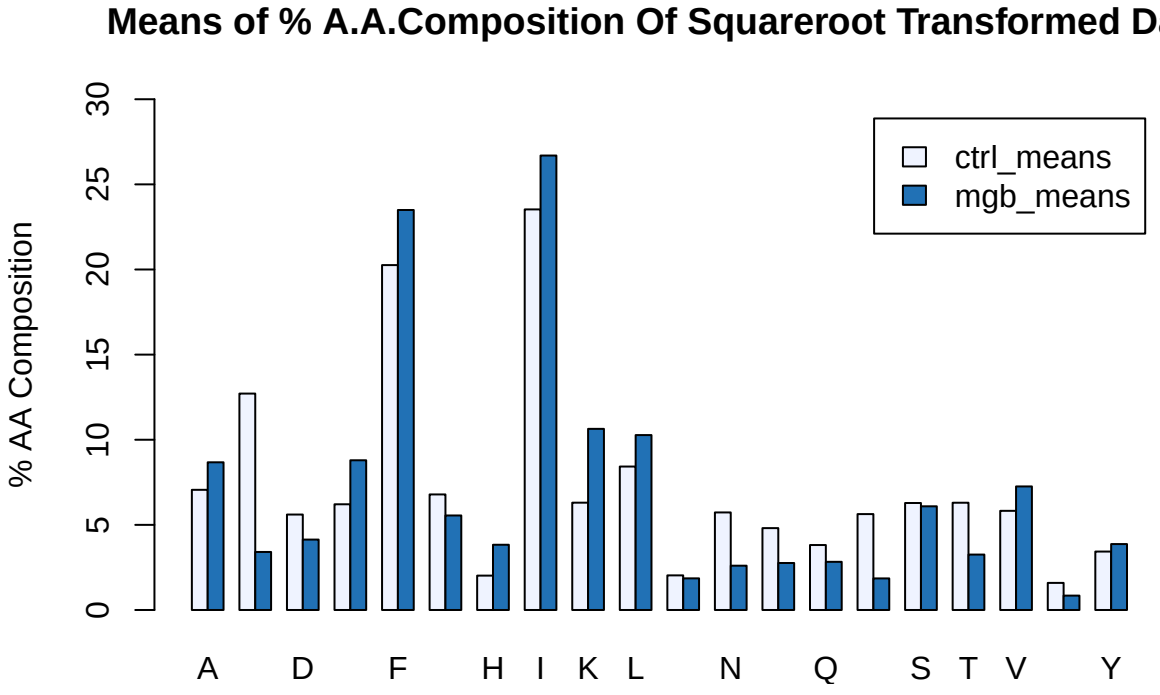
$$E[X] = \sum_{i=1}^{n} x_i p_i \quad ; \qquad \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Scatter plot of means of *Myoglobin-Control* amino acid composition $\sqrt{x}_i$ Transformed (c_m_TRANSFORMED) dataframe

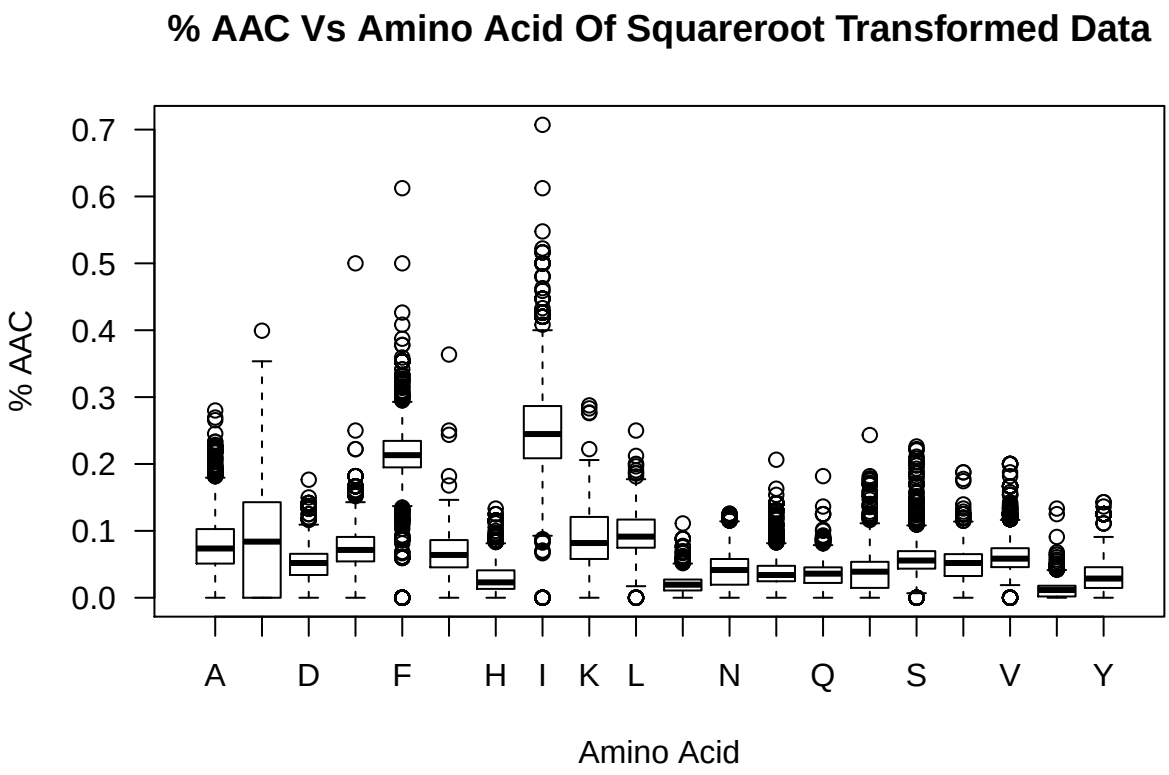
- This plot shows the means for each feature (column-means) in the dataset. The means represent the ungrouped or total of all proteins (where n=2340) versus AA type.



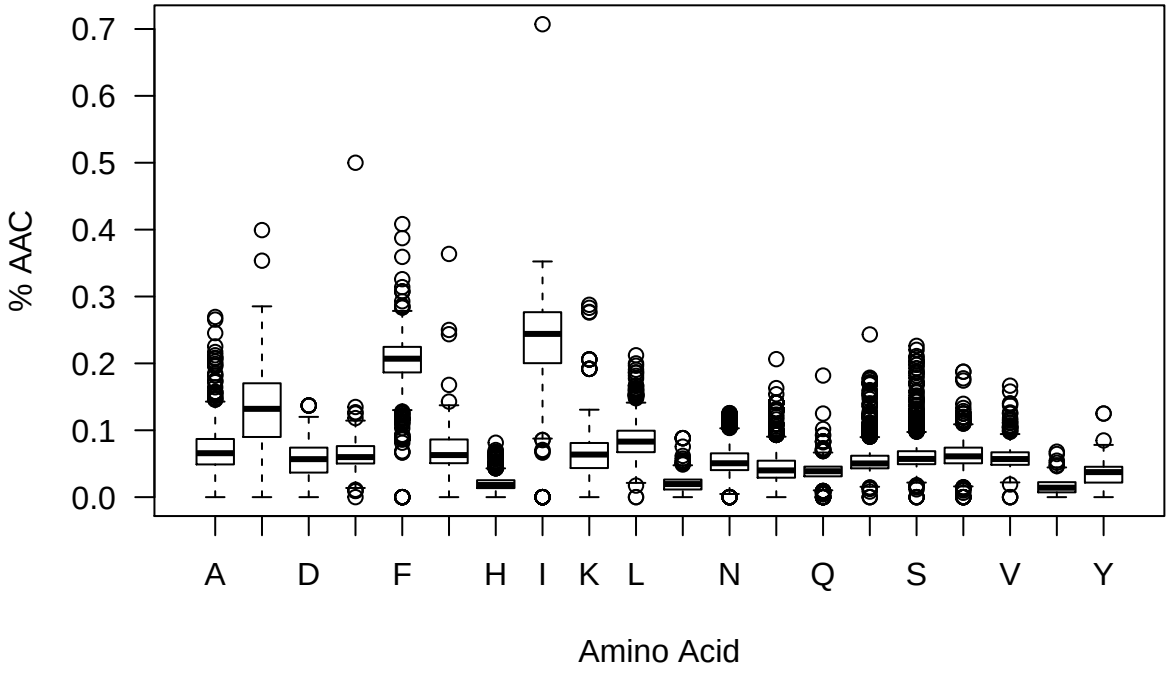
Grouped barchart of means for percent amino acid composition of Transformed Data; control & myoglobin categories



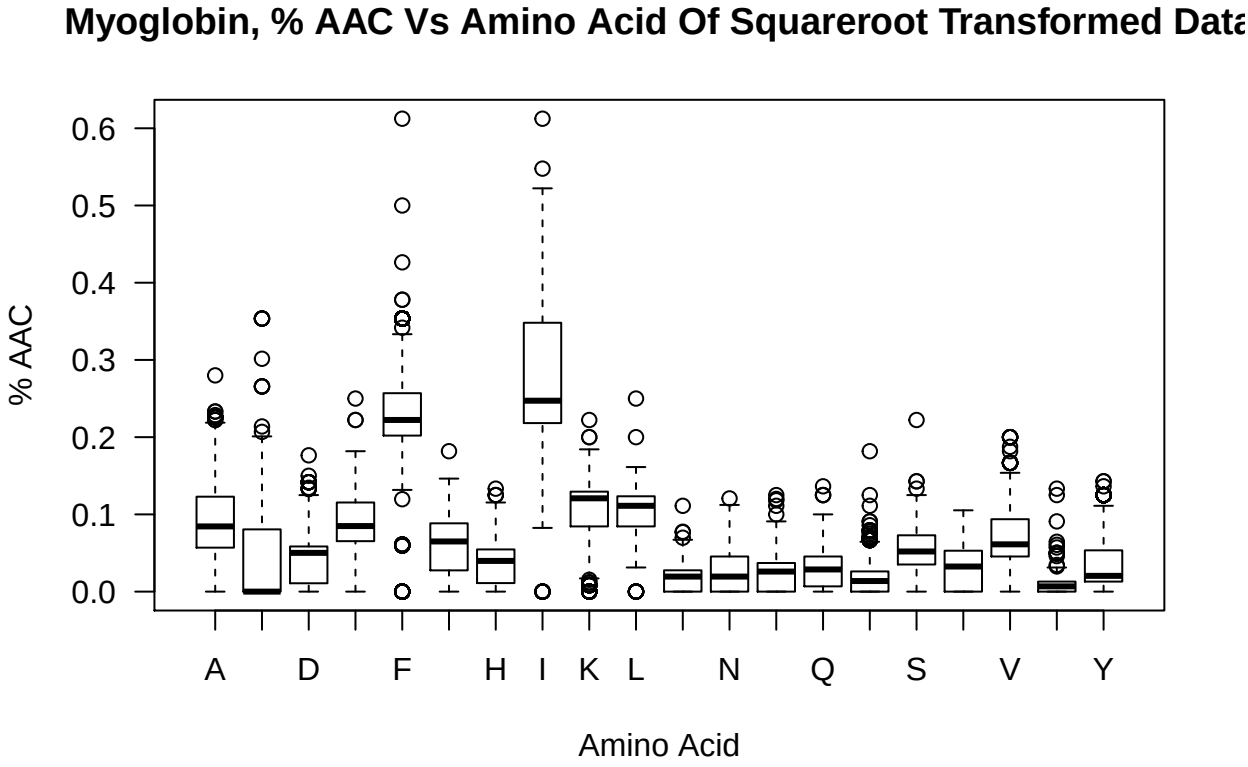
Boxplots of grand-means of overall amino acid composition of squareroot transformed data



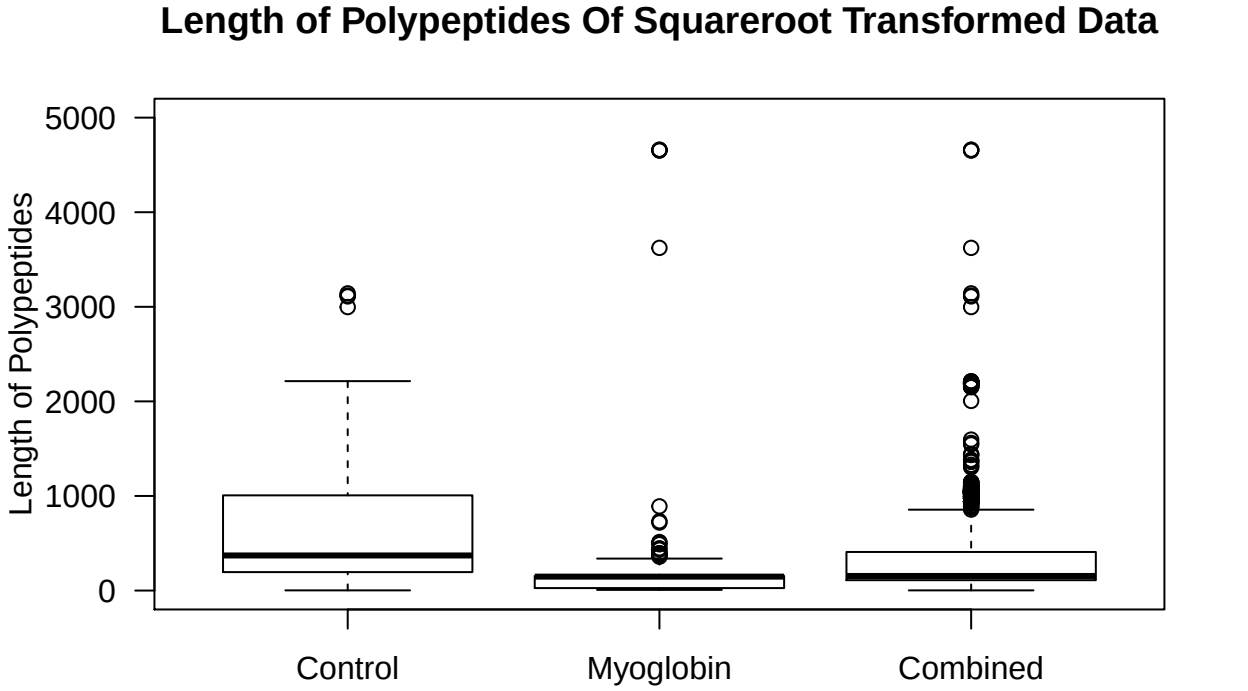
Boxplots of amino acid compositions for control (only) of squareroot transformed data



Boxplots of amino acid compositions for myoglobin of squareroot transformed Data(only) of squareroot transformed data



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



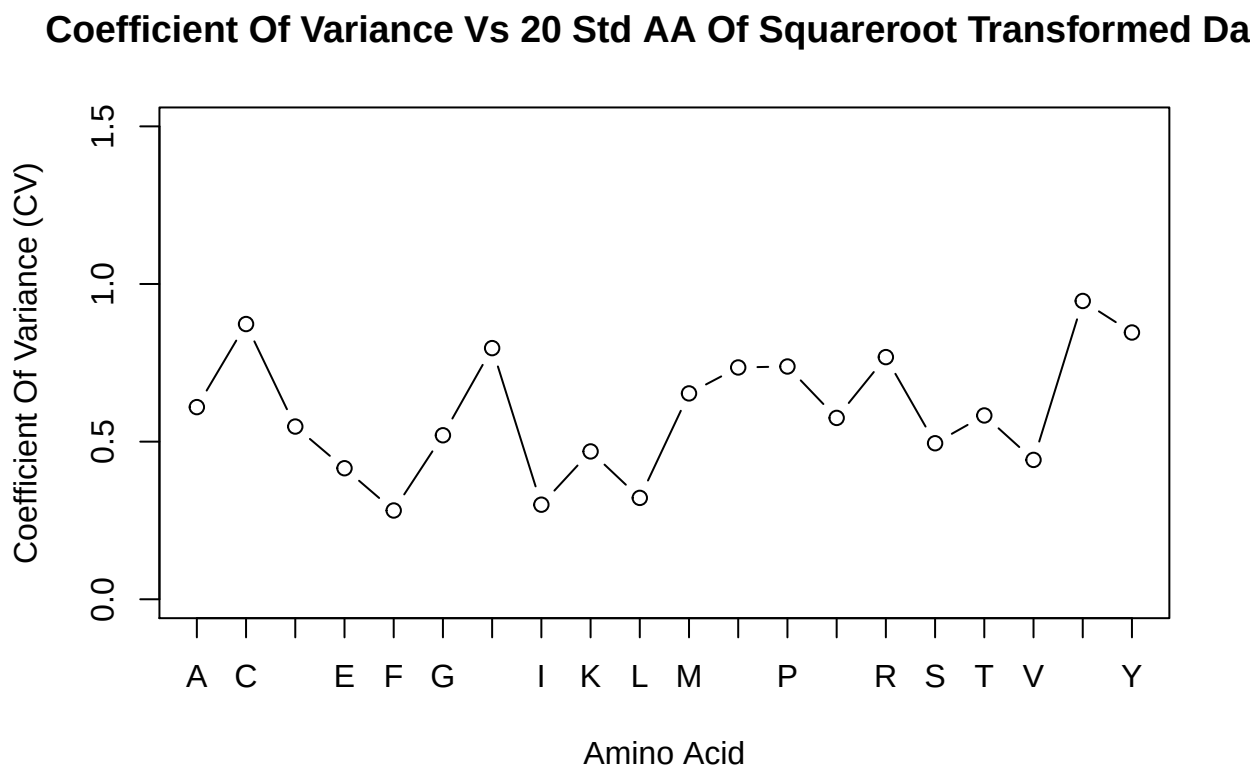
Coefficient of variance (CV) Of Transformed data

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

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

$$CV = \frac{1}{\bar{x}} \cdot \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

Plot of normalized standard deviations




```
AA_var_norm
```

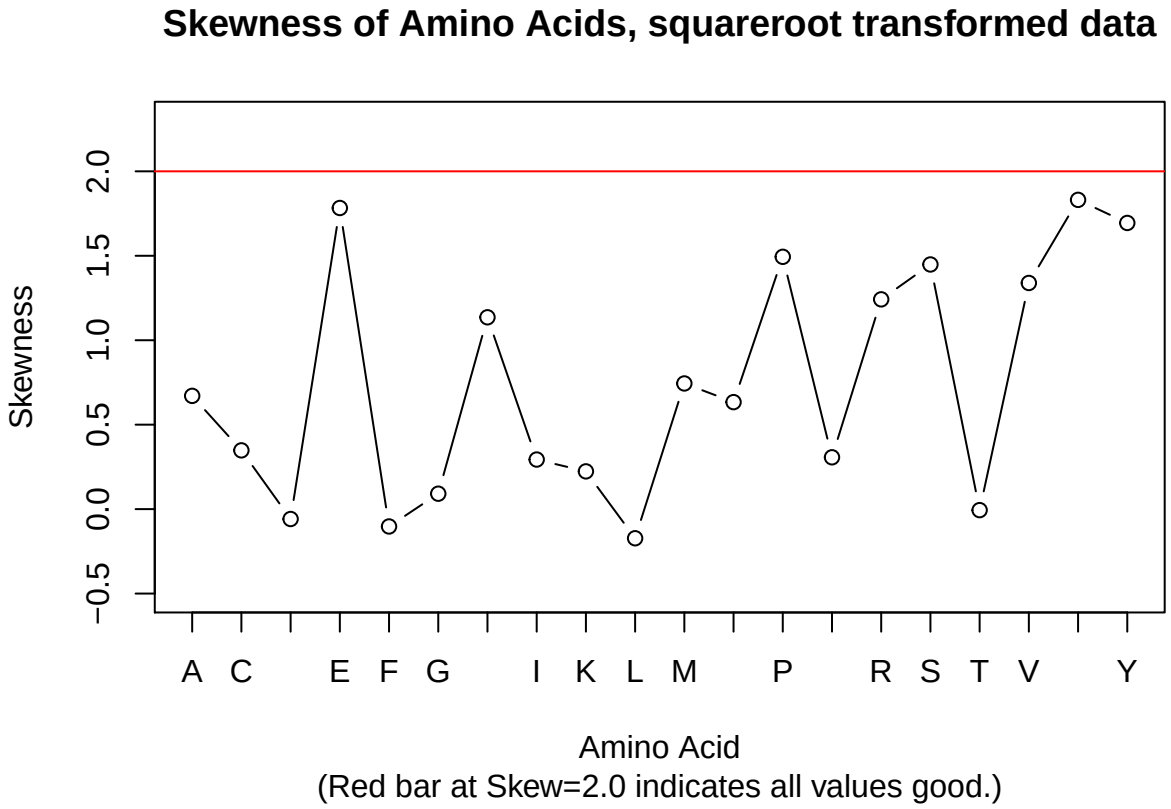
```
##         A         C         D         E         F         G         H         I
## 0.6095112 0.8729758 0.5478540 0.4156102 0.2815745 0.5201625 0.7966296 0.2999687
##         K         L         M         N         P         Q         R         S
## 0.4689544 0.3215591 0.6529752 0.7352478 0.7383244 0.5752622 0.7680977 0.4948690
##         T         V         W         Y
## 0.5830352 0.4420595 0.9461276 0.8461615
```

Skewness of distributions Of Transformed Data

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

$$Skewness = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\left(\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)^3}$$

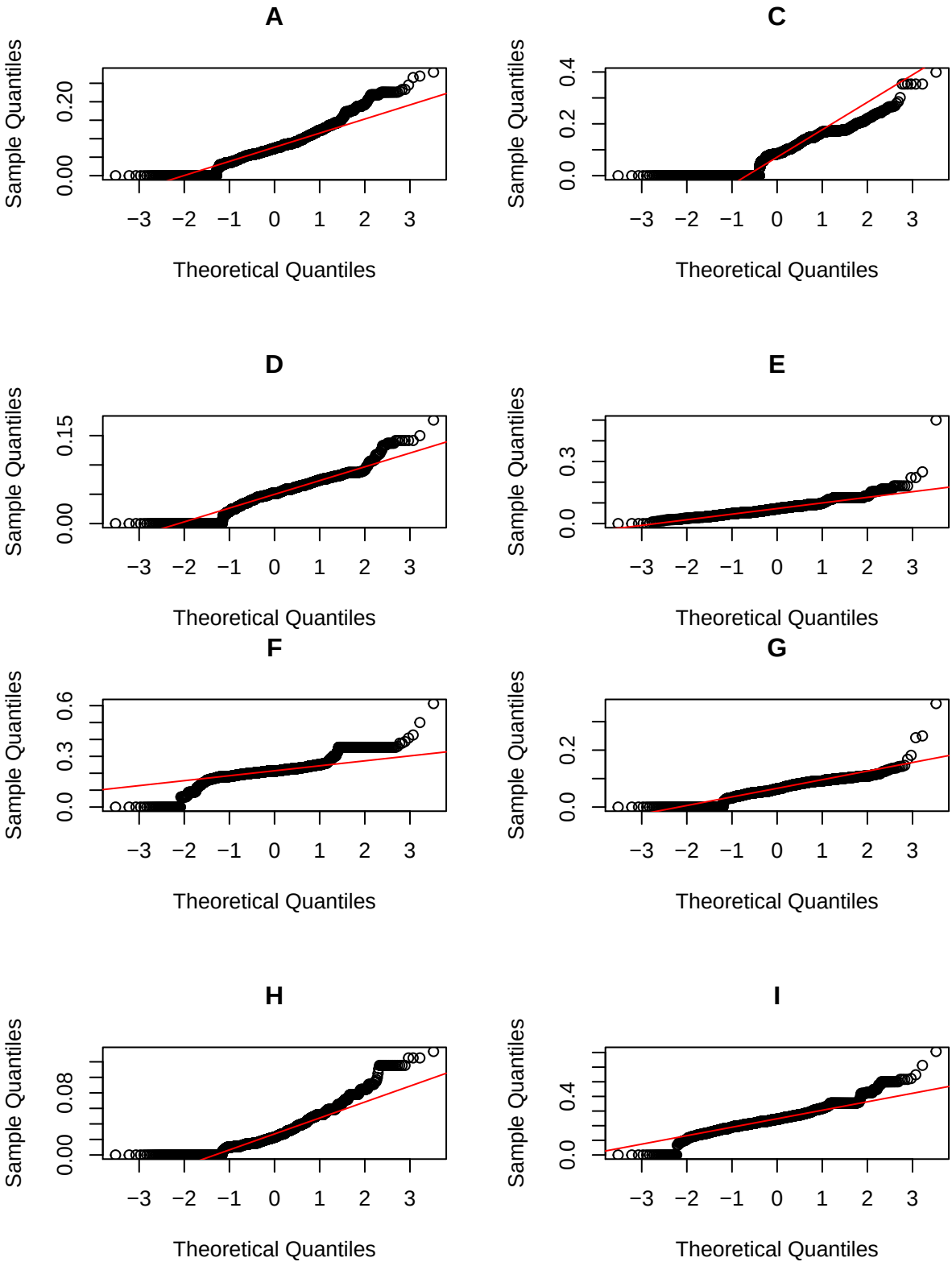
- Skewness values for each A.A. by Class of squareroot transformed data

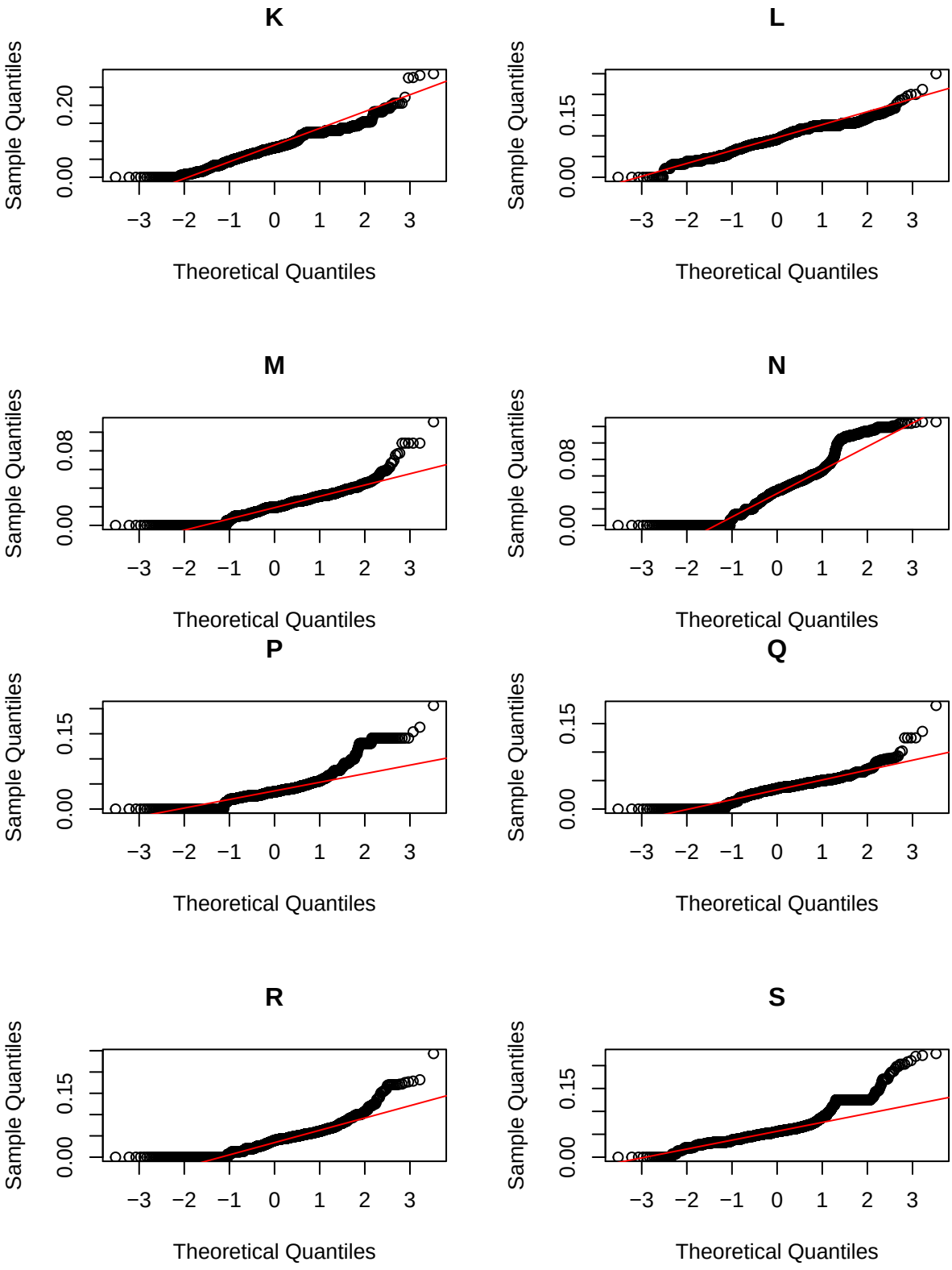


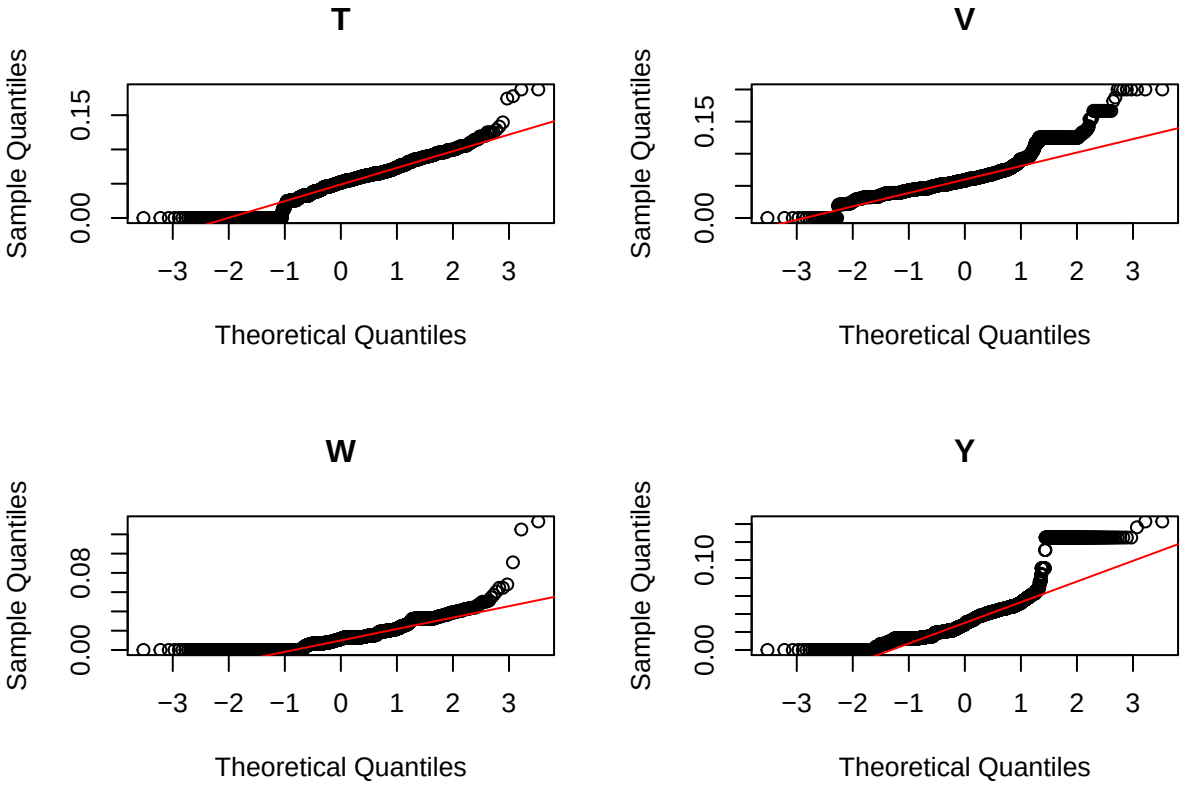

```
AA_skewness
```

```
##           A           C           D           E           F           G
##  0.670502595  0.347813248 -0.058540442  1.782876260 -0.102739748  0.091338300
##           H           I           K           L           M           N
##  1.135783661  0.293474879  0.223433207 -0.172566877  0.744002991  0.633532783
##           P           Q           R           S           T           V
##  1.493903282  0.306716333  1.241930812  1.448521897 -0.006075043  1.338971930
##           W           Y
##  1.831047440  1.694362388
```

QQ Plots of 20 amino acids of Transformed data




K
L
M
N
P
Q
R
S
Sample Quantiles
Theoretical Quantiles
−3 −2 −1 0 1 2 3
0.00 0.20
0.00 0.15
0.00 0.08
0.00 0.15



Determine coefficients of correlation of Transformed Data

An easily interpretable test, is correlation 2D-plot for investigating multicollinearity or feature reduction. It is clear that fewer attributes "means decreased computational time and complexity. Secondly, if two predictors are highly correlated, this implies that they are measuring 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".[9]

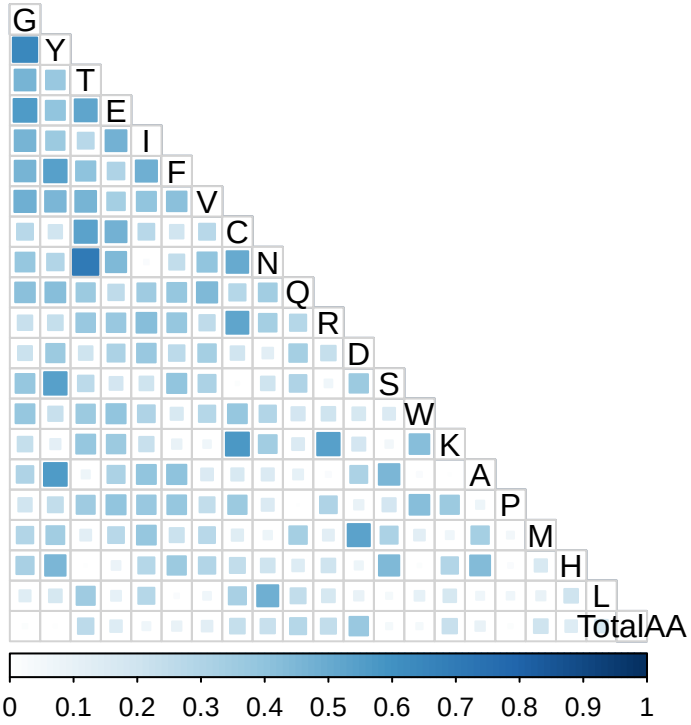
Pearson's correlation coefficient:

$$\rho_{x,y} = \frac{E\left[(X - \mu_x)(X - \mu_y)\right]}{\sigma_x \sigma_y}$$

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_1 - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

[9] "Applied Predictive Modeling", Max Kuhn and Kjell Johnson, Springer Publishing, 2018, P.43

Correlation Plot Of Transformed Features




```
c_m_corr_mat["T", "N"]
```

```
## [1] 0.7098085
```

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

Boruta - dimensionality reduction of Transformed data

Perform Boruta search

```
NOTE: *mcAdj = TRUE*: If True, multiple comparisons  will be adjusted using the Bonferroni method to ca
```

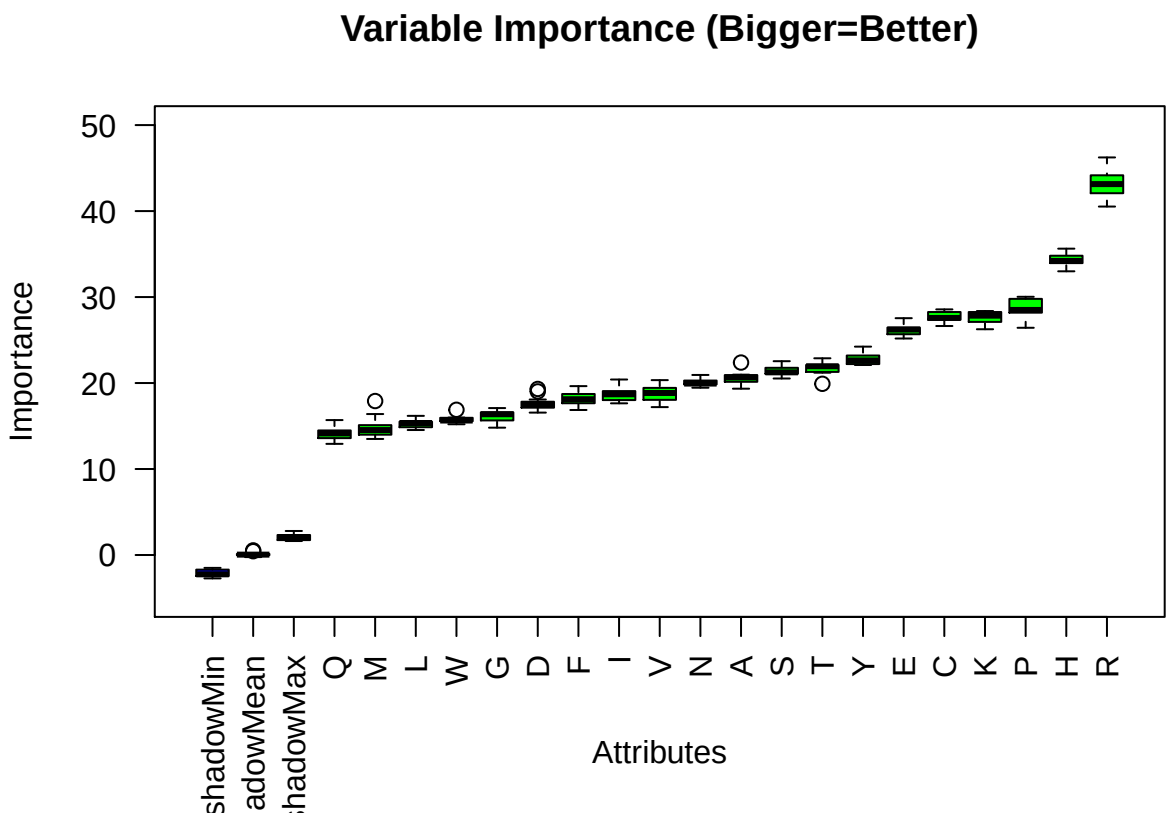
```
set.seed(1000)
registerDoMC(cores = 3)  # Start multi-processor mode
start_time <- Sys.time() # Start timer

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
```

```
## Time difference of 17.23543 secs
```

Plot variable importance

```
plot(boruta_output,
     cex.axis = 1,
```

```
     las = 2,
     ylim = c(-5, 50),
     main = "Variable Importance (Bigger=Better)")
```



Variable importance scores

```
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")
```

Table 4: Mean Importance Scores & Decision

	meanImp	decision
R	43.17613	Confirmed
H	34.30370	Confirmed
P	28.70674	Confirmed
C	27.72357	Confirmed
K	27.60838	Confirmed
E	26.18872	Confirmed
Y	22.84975	Confirmed
T	21.66359	Confirmed
S	21.44119	Confirmed

	meanImp	decision
A	20.54316	Confirmed
N	20.10100	Confirmed
V	18.77068	Confirmed
I	18.69155	Confirmed
F	18.18632	Confirmed
D	17.64435	Confirmed
G	16.15207	Confirmed
W	15.77085	Confirmed
L	15.27614	Confirmed
M	14.83421	Confirmed
Q	14.12976	Confirmed

Conclusion for Boruta random forest test

All features are important. None should be dropped.

EDA Conclusions For Transformed Data

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 appreciablly change any important measures such as the feature importance derived from the Boruta random forest feature selection work. Nor did the transformations of C,F,I apreciably change the correlation coeffiecient matrix, therefore the transformed data will be used throughout this experiment.

Regarding Boruta which is used for dimensionality reduction of Transformed data, it showed that all features (x variables) are important for the generation of a Decision Tree. My belief is that this would imply that given that 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 positive mean importance which are generated by a Gini calculation.

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