Contents

1. Introduction

The aim of this project is to apply multivariate analysis methods using one real world example with presenting the effectiveness of various factors on championship of Fenerbahçe during the 2016 – 2017 season of Euro League.

Begin with the preparation of the data with basic summary statistics, the assumptions of normality and the assumptions of linear regression are checked with some model diagnostics. After performing a principal component analysis on the quantitative variables, classification and clustering methods are applied followed by factor analysis. In this project. R-Studio is used for the whole process of this analysis with Machine Learning algorithms.

1.1 Data Description

This project includes a detailed study of multivariate analysis with the dataset of 119 observations, each represents the basketball players' statistics in Euro League in 2016 – 2017 Season, along with 21 continuous variables and 5 discrete variables collected from the website www.basketball.realgm.com, and the official website of Euro League which is www.euroleague.net . The descriptions of the variables are follow:

Player	Player Name
Team	Team Name
GP	Games Played
MPG	Minutes Per Game
FGM	Field Goals Made
FGA	Field Goals Attempts
FG%	Field Goal Percentage
3PM	Three-Point Field Goals Made
3PA	Three-Point Field Goals Attempted
3P%	Three-Point Field Goal Percentage
FTM	Free Throws Made
FTA	Free Throws Attempted
FT%	Free Throw Percentage
TOV	Turnovers
PF	Personal Fouls
ORB	Offensive Rebounds
DRB	Defensive Rebounds
RPG	Rebounds per Game
APG	Assists per Game
SPG	Steals per Game
BPG	Blocks per Game
PPG	Points per Game
IR	Index Rating
FF	Final Four
Position	Positions of Players

```
       GP                 MPG                FGM                FGA                FG.          
 Min.   :-2.3043000   Min.   :-1.8526   Min.   :-1.987200   Min.   :-1.691200   Min.   :-1.6348  
 1st Qu.:-0.5378000   1st Qu.:-0.7032   1st Qu.:-0.681700   1st Qu.:-0.777100   1st Qu.:-0.7759  
 Median :-0.1302000   Median :-0.1285   Median : 0.014600   Median :-0.061800   Median :-0.2424  
 Mean   : 0.0000025   Mean   : 0.0000   Mean   : 0.000005   Mean   :-0.000002   Mean   : 0.0000  
 3rd Qu.: 0.6851000   3rd Qu.: 0.7643   3rd Qu.: 0.667400   3rd Qu.: 0.554200   3rd Qu.: 0.5188  
 Max.   : 1.7722000   Max.   : 2.5603   Max.   : 3.322000   Max.   : 4.031500   Max.   : 3.0107  
      X3PM               X3PA               X3P.               FTM          
 Min.   :-1.515900   Min.   :-1.6026000   Min.   :-2.341400   Min.   :-1.314900  
 1st Qu.:-0.698200   1st Qu.:-0.5647000   1st Qu.:-0.176600   1st Qu.:-0.695200  
 Median :-0.044000   Median :-0.0458000   Median : 0.273700   Median :-0.178800  
 Mean   : 0.000005   Mean   :-0.0000008   Mean   : 0.000005   Mean   : 0.000003  
 3rd Qu.: 0.610200   3rd Qu.: 0.6353500   3rd Qu.: 0.601500   3rd Qu.: 0.492550  
 Max.   : 2.245700   Max.   : 2.7435000   Max.   : 1.534400   Max.   : 4.675600  
      FTA                FT.                TOV                PF           
 Min.   :-1.476100   Min.   :-2.4100000   Min.   :-1.629400   Min.   :-2.3594000  
 1st Qu.:-0.705300   1st Qu.:-0.6503000   1st Qu.:-0.728200   1st Qu.:-0.7211000  
 Median :-0.191400   Median : 0.1218000   Median :-0.127500   Median : 0.0981000  
 Mean   : 0.000004   Mean   :-0.0000017   Mean   : 0.000003   Mean   : 0.0000025  
 3rd Qu.: 0.493700   3rd Qu.: 0.8266500   3rd Qu.: 0.473300   3rd Qu.: 0.5077000  
 Max.   : 4.519000   Max.   : 1.8457000   Max.   : 4.077800   Max.   : 2.3507000  
      ORB                DRB                RPG                APG           
 Min.   :-1.1629000   Min.   :-1.811200   Min.   :-1.5758000   Min.   :-1.0863000  
 1st Qu.:-0.7107000   1st Qu.:-0.779500   1st Qu.:-0.8207000   1st Qu.:-0.7533000  
 Median :-0.4092000   Median :-0.123000   Median :-0.2543000   Median :-0.3537000  
 Mean   : 0.0000042   Mean   : 0.000002   Mean   :-0.0000042   Mean   :-0.0000092  
 3rd Qu.: 0.6461000   3rd Qu.: 0.533600   3rd Qu.: 0.7525000   3rd Qu.: 0.3789000  
 Max.   : 3.0581000   Max.   : 3.535000   Max.   : 2.8919000   Max.   : 2.7764000  
      SPG                BPG                PPG                IR           
 Min.   :-1.857500   Min.   :-0.809200   Min.   :-1.848900   Min.   :-1.9510000  
 1st Qu.:-0.613900   1st Qu.:-0.524300   1st Qu.:-0.662600   1st Qu.:-0.7046000  
 Median : 0.007800   Median :-0.239400   Median :-0.115000   Median : 0.0119000  
 Mean   : 0.000004   Mean   :-0.000016   Mean   : 0.000001   Mean   : 0.0000034  
 3rd Qu.: 0.318700   3rd Qu.: 0.187950   3rd Qu.: 0.539000   3rd Qu.: 0.5822000  
 Max.   : 4.360200   Max.   : 5.458200   Max.   : 4.022000   Max.   : 2.9872000  
```

Figure 1: Summary of the data

Played Games, Minutes per Game, Field Goals Attempts, Points per Game and Index Rating have higher range comparing with other variables. It means there is more variety in these variables.

1.2 Correlation

The correlation coefficient ranges from -1 to 1, and it can be interpreted as if the correlation coefficient is closer to 1, it means it's highly correlated, and the same can be said for -1. If the correlation coefficient is closer to 0, it means that the variables are not correlated.

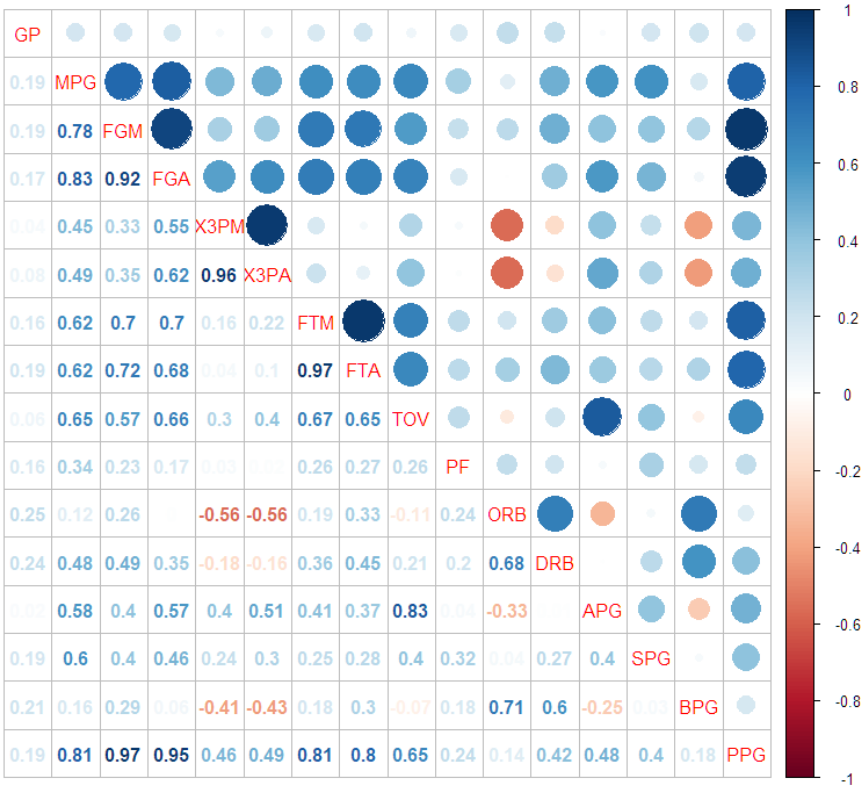

Figure 2: Correlation matrix of dataset

Figure 2 shows the correlation between each variable. According to the *Figure 2*, correlation between Offensive Rebounds and Blocks is quite high, thus this correlation simply interpretted as for whom has higher block rate has also tendency of grabbing more offensive rebounds. Additionally, there is another high correlation between Tornovers and Assists. Again, this is also a positive correlation which means players who are trying to serve the ball more (Assist), more likely to have higher Tornovers. However, the correlation between 3Points Made and Blocks appears to be negative. That means, players who can block, cannot made 3 Point Shots.

2. Assumption Checking

2.1 Normality Test

A Shapiro-Wilk procedure, interpreted based on the p-value, is applied to each variable to test for normality. The null hypothesis of this test is that the variables come from a normal distribution.

H0: The variable is distributed normally.

Ha: The variable is distributed non-normally.

P-value: 0.05

```
> shapiro_test_df(a)
$statistic
     GP.W     MPG.W     FGM.W     FGA.W     FG..W    X3PM.W    X3PA.W    X3P..W
0.9561331 0.9836181 0.9842829 0.9632806 0.9417758 0.9620468 0.9722916 0.8240883
    FTM.W     FTA.W     FT..W     TOV.W      PF.W     ORB.W     DRB.W     RPG.W
0.8794504 0.9097409 0.9633775 0.9336844 0.9901148 0.8713189 0.9522568 0.9293356
    APG.W     SPG.W     BPG.W     PPG.W      IR.W
0.8525923 0.9352960 0.7249932 0.9574169 0.9814851

$p.value
          GP           MPG           FGM           FGA           FG.          X3PM
6.655894e-04 1.578345e-01 1.812474e-01 2.470277e-03 6.071063e-05 1.957720e-03
        X3PA          X3P.           FTM           FTA           FT.           TOV
1.461640e-02 1.313717e-10 2.212571e-08 7.001737e-07 2.516085e-03 1.777553e-05
          PF           ORB           DRB           RPG           APG           SPG
5.508532e-01 9.594087e-09 3.384140e-04 9.476574e-06 1.576544e-09 2.256184e-05
         BPG           PPG            IR
1.240786e-13 8.371139e-04 1.007133e-01

$significance
  GP  MPG  FGM  FGA  FG. X3PM X3PA X3P.  FTM  FTA  FT.  TOV   PF  ORB  DRB  RPG  APG
"Ha" "H0" "H0" "H0" "Ha" "Ha" "H0" "Ha" "Ha" "Ha" "H0" "Ha" "H0" "Ha" "Ha" "Ha" "Ha"
 SPG  BPG  PPG   IR
"Ha" "Ha" "Ha" "H0"

$method
[1] "Shapiro-Wilks test with Bonferroni Correction"
```

Figure 3: Shapiro-Wilk Test

It can be seen that from the output, most of the variables have p-value greater than the alpha level 0.05, then the null hypothesis that the data come from a normal distribution cannot be rejected. The variables Minutes per Game (MPG), Field Goals Made (FGM), Field Goals Attempts (FGA), Three-Point Field Goals Attempted (3PM), Free Throw Percentage (FT), Personal Fouls (PF), and Index Rating (IR) are not seems to be normally distributed.

2.2 Multivariate Normality

```
$multivariateNormality
     Test        H      p value MVN
1 Royston 228.0127 1.947014e-41  NO

$univariateNormality
           Test  Variable Statistic   p value Normality
1  Shapiro-Wilk    GP        0.9561   7e-04       NO
2  Shapiro-Wilk    MPG       0.9836  0.1578       YES
3  Shapiro-Wilk    FGM       0.9843  0.1812       YES
4  Shapiro-Wilk    FGA       0.9633  0.0025       NO
5  Shapiro-Wilk    FG.       0.9418   1e-04       NO
6  Shapiro-Wilk   X3PM       0.9620   0.002       NO
7  Shapiro-Wilk   X3PA       0.9723  0.0146       NO
8  Shapiro-Wilk   X3P.       0.8241  <0.001       NO
9  Shapiro-Wilk    FTM       0.8795  <0.001       NO
10 Shapiro-Wilk    FTA       0.9097  <0.001       NO
11 Shapiro-Wilk    FT.       0.9634  0.0025       NO
12 Shapiro-Wilk    TOV       0.9337  <0.001       NO
13 Shapiro-Wilk    PF        0.9901  0.5509       YES
14 Shapiro-Wilk    ORB       0.8713  <0.001       NO
15 Shapiro-Wilk    DRB       0.9523   3e-04       NO
16 Shapiro-Wilk    RPG       0.9293  <0.001       NO
17 Shapiro-Wilk    APG       0.8526  <0.001       NO
18 Shapiro-Wilk    SPG       0.9353  <0.001       NO
19 Shapiro-Wilk    BPG       0.7250  <0.001       NO
20 Shapiro-Wilk    PPG       0.9574   8e-04       NO
21 Shapiro-Wilk    IR        0.9815  0.1007       YES
```

NO: *There is not normality*

YES: *There is normality*

Figure 4: Royston's Multivariate Normality Test

The applied test is Royston's Multivariate Normality Test. Royston's H test uses Shapiro-Wilk's W statistic for multivariate normality. Here, H is the value of Royston's H statistic at significance level 0.05 and p-value is an approximate p-value for the test. According to the Royston's Multivariate Normality Test, the data set does not appear to follow a multivariate normal distribution. ($p<0.001$)

3. Index Rating Estimation

3.1 Model Selection

To find the most important factors one multiple regression is conducted using IR as the response and all the other variables as predictors. Why IR is the response? It is actually calculated used the other continuous variables that represents any actions carried out during the game.

```
Call:
lm(formula = unlist(y) ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-0.82890 -0.21259  0.01116  0.21379  0.88480

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.48556    0.96081   0.505 0.614438
xGP         -0.01012    0.01111  -0.911 0.364613
xMPG         0.00284    0.01884   0.151 0.880523
xFGM         0.48741    1.18553   0.411 0.681875
xFGA        -1.08145    0.15414  -7.016 2.98e-10 ***
xFG.        -0.79526    1.78993  -0.444 0.657805
xX3PM        0.26871    0.61354   0.438 0.662377
xX3PA       -0.12018    0.14226  -0.845 0.400282
xX3P.       -0.31012    0.50075  -0.619 0.537150
xFTM         0.52370    0.64352   0.814 0.417733
xFTA         0.07391    0.27667   0.267 0.789914
xFT.        -0.04166    0.60875  -0.068 0.945578
xTOV        -1.00610    0.14982  -6.715 1.23e-09 ***
xPF         -0.89965    0.09411  -9.559 1.10e-15 ***
xORB         2.40796    0.77636   3.102 0.002514 **
xDRB         2.79639    0.78981   3.541 0.000613 ***
xRPG        -1.60394    0.77913  -2.059 0.042184 *
xAPG         1.06460    0.06010  17.715  < 2e-16 ***
xSPG         0.89678    0.15728   5.702 1.25e-07 ***
xBPG         0.74901    0.16169   4.632 1.11e-05 ***
xPPG         1.29537    0.55639   2.328 0.021958 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3885 on 98 degrees of freedom
Multiple R-squared:  0.9929,    Adjusted R-squared:  0.9915
F-statistic: 687.4 on 20 and 98 DF,  p-value: < 2.2e-16
```

Figure 5: Summary of the model

The model summary is given in the Figure 5. The significance level is explained in the table with "*". The more the stars beside the variable the more significant it is. It seems many of the variables do not actually represents the IR significantly. One good thing is the R-square which is 99%. This actually means that the significant variables mostly can define the variability in that IR variable.

3.2 Model Diagnostics

Let's look at the Residual Plot to see how they are behaving.

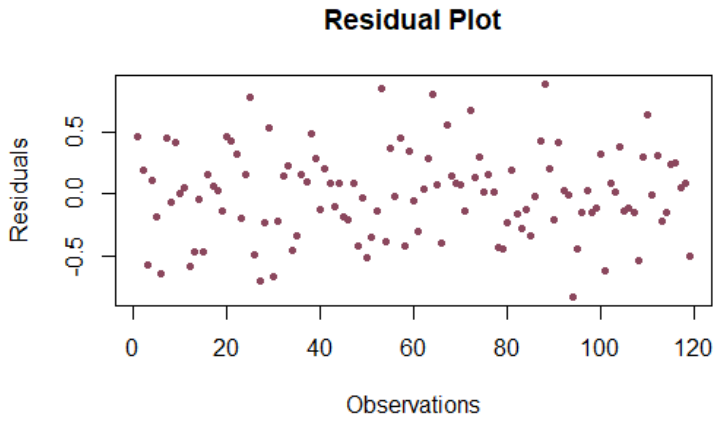

Figure 6: Residual Plot

The residuals are distributed around 0 (the fitted values are very close to the original values) which is desired in this kind of models.

A "well-behaved" Residuals vs. Fitted plot should be relatively shapeless without clear patterns in the data, no obvious outliers, and be generally symmetrically distributed around the 0 line without particularly large residuals.

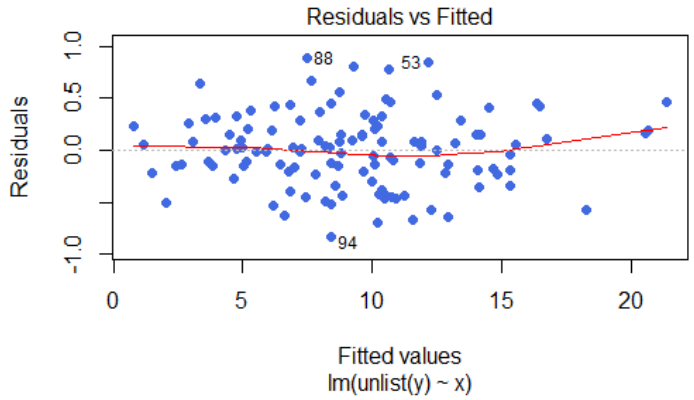

Figure 7: Residuals & Fitted Plot

Here is the residuals appear on the ***y*** axis and the fitted values appear on the ***x*** axis. This Residuals vs. Fitted plot shows no sign of unwanted patterns. However, just like the Cook Distance plot, it shows there are 3 outliers present but these are different then the ones from Cook Distance plot. On the other hand, there is no obvious sign of heteroscedasticity present, but if the residuals were much scattered away from the fitted line, then presence of heteroscedasticity could be taken into account.

3.2.1 Normality of Residuals

The residuals measure the deviations between the prediction of the dependent variable using the independent variables and the observed values. Since the residuals represent the information that the model hasn't accounted for, we create a normal probability plot of the residuals. If the data follow a normal distribution, then a plot of the theoretical quantiles of the normal distribution versus the standardized residuals should be approximately linear. If the plot is approximately linear, it can be assumed that the residuals (error terms) are normally distributed.

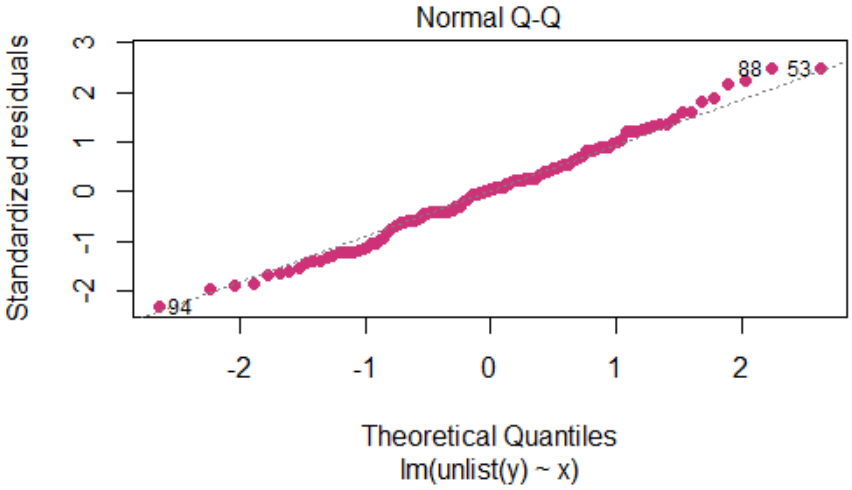

Figure 8: Normal Q-Q Plot

From the Q-Q plot, it is obvious that the residuals look like normally distributed but then to be surer of it, let's look at the Shapiro-Wilk Test results:

```
> shapiro.test(ei)

        Shapiro-Wilk normality test

data:  ei
W = 0.99158, p-value = 0.687
```

Figure 9: Shapiro Wilk Test

The p-value is greater than the alpha level 0.05. So, the residuals are normally distributed.

3.2.2 Multicollinearity Checking

One of the biggest assumptions of linear modeling is independence of predictors, apart from the normality of residuals and homogeneity of variance. If one or more of the predictors in a model are correlated, then the model may produce unstable parameter estimates with highly inflated standard errors, resulting in an overall significant model with no significant predictors. If the VIF value is high, it indicates that that predictor is highly correlated with other predictors, it contains little or no information, and there is redundancy in the set of predictors.

```
> vif(fit)
        GP         MPG         FGM         FGA         FG.        X3PM        X3PA        X3P.
  1.306878    6.590472 1450.692607  117.634887   14.794026  110.052271   37.611940    3.778250
       FTM         FTA         FT.         TOV          PF         ORB         DRB         RPG
303.652619   81.612756    3.594967    7.781370    1.651574  207.422496  554.579880 1199.022039
       APG         SPG         BPG         PPG
  6.367935    2.001140    2.519002 2615.939387
```

Figure 10: Values of Variance Inflation Factor

In the Figure, some of the VIF scores are bigger than 10, it indicates that there is a multicollinearity problem. Hence, most of the variables in the dataset are highly correlated.

3.2.3 Cook Distance (Outlier Test)

It can be used to spot outliers in the model. A general rule is that observations with a Cook's Distance of more than 3 times the mean, μ, is a possible outlier.

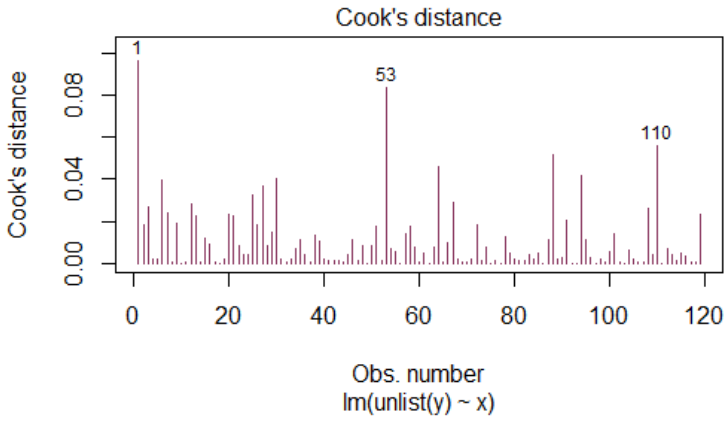

Figure 11: Cook's Distance Plot

From the Figure 11, it can be said that there are 3 influential points in the model. They are the 1st, 53rd and 110th observation but these have to be checked whether if they are outliers or not.

```
> outlierTest(fit)
No Studentized residuals with Bonferonni p < 0.05
Largest |rstudent|:
   rstudent unadjusted p-value Bonferonni p
53 2.540718           0.012649           NA
```

Figure 12: Outlier Test

3.2.4 Heteroscedasticity of Residuals (Constant Variance Test)

To satisfy the regression assumptions and be able to trust the results, the residuals should have a constant variance. Heteroscedasticity means inequality in the spread of the residuals over the range of measured values. It is a problem because the regression assumes that all residuals are drawn from the model that has a constant variance (homoscedasticity).

H0: The variance is constant.

Ha: There is heteroscedasticity.

To check heteroscedasticity, Breusch-Pagan Test is used.

```
> bptest(reg)

        studentized Breusch-Pagan test

data:  reg
BP = 21.585, df = 20, p-value = 0.3634
```

Figure 13: Heteroscedasticity of residuals with Breusch Pagan test

From the results of test, p-value is greater than 0.05 which means we cannot reject the null hypothesis. Therefore, it can be said that there is no heteroscedasticity present in the data.

4. Multinomial Regression for Position Prediction

It is preferable to use multinomial regression because the variable named Position is a categorical variable. Therefore, logistics and other the regressions were a mismatch for the situation. The position variable includes Center, Guard, and Forward. All the other variables are used to predict the position of the players.

```
> summary(mp2)

Call:
vglm(formula = Position ~ APG + BPG + X3PM + DRB + ORB + PF +
    IR + MPG, family = multinomial, data = subsetMulti, method = "vglm.fit")


Pearson residuals:
                       Min       1Q    Median         3Q    Max
log(mu[,1]/mu[,3]) -1.159 -0.05097 -0.007243 -0.0004072 12.327
log(mu[,2]/mu[,3]) -3.826 -0.21978 -0.011072  0.2300248  2.082

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept):1  -6.1516     4.1224  -1.492 0.135635
(Intercept):2  -1.8901     2.0646  -0.915 0.359932
APG:1          -1.8969     1.0441  -1.817 0.069255 .
APG:2          -1.6443     0.6491  -2.533 0.011306 *
BPG:1           8.5721     4.9126   1.745 0.081000 .
BPG:2           7.9625     4.6883   1.698 0.089438 .
X3PM:1          5.0084     2.4457   2.048 0.040576 *
X3PM:2          2.1526     1.3567   1.587 0.112582
DRB:1           1.9707     1.8051   1.092 0.274944
DRB:2           3.9922     1.3601   2.935 0.003333 **
ORB:1          11.4461     3.2121   3.563 0.000366 ***
ORB:2           6.0350     2.1402   2.820 0.004805 **
PF:1           -2.5070     1.6418  -1.527 0.126775
PF:2           -3.0036     1.3198  -2.276 0.022857 *
IR:1            0.1758     0.6188   0.284 0.776372
IR:2           -0.8413     0.3980  -2.114 0.034514 *
MPG:1          -0.4879     0.3528      NA       NA
MPG:2           0.1027     0.1994   0.515 0.606693
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Number of linear predictors:  2

Names of linear predictors: log(mu[,1]/mu[,3]), log(mu[,2]/mu[,3])

Residual deviance: 69.6584 on 220 degrees of freedom

Log-likelihood: -34.8292 on 220 degrees of freedom

Number of iterations: 8
```

Figure 14: Summary of the fit with vector generalized linear models

Looking at the summary of model, it seems that the most significant variable is Offensive Rebounds, followed by the variable Defensive Rebounds. Other significant variables are Assists per Game, Three-Point Field Goals Made, Personal Fouls, and Index Rating. Since the residual deviance over degrees of freedom is away from 1, this can be interpreted as over dispersion problem. Thus, it can be said that the multinomial regression is not suitable for this analysis.

5. Factor Analysis

Before start to analysis, "factorability" of the data should be check. The Kaiser-Meyer-Olkin (KMO) measure of is a better measure of factorability. If Overall MSA (Measure of sampling Adequacy) is higher than 0.50, you can apply factor analysis to your data.

```
> KMO(data.st)
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = data.st)
Overall MSA =  0.65
MSA for each item =
          GP           MPG           FGM           FGA          X3PM          X3PA           FTM
        0.68          0.93          0.67          0.69          0.52          0.88          0.63
         FTA           TOV            PF           ORB           DRB           APG           SPG
        0.95          0.70          0.27          0.59          0.47          0.46          0.54
         BPG           PPG index_rating
        0.68          0.69          0.61
```

Figure 15: The Kaiser-Meyer-Olkin factor adequacy

According to *Figure 15*, overall MSA is 0.65. It is higher than 0.50, but MSA for PF (Personal Foul) is too low. This item will be dropped and KMO will be applied again.

```
> KMO(data.st.d)
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = data.st.d)
Overall MSA =  0.7
MSA for each item =
          GP           MPG           FGM           FGA          X3PM          X3PA           FTM
        0.67          0.88          0.66          0.79          0.51          0.86          0.62
         FTA           TOV           ORB           DRB           APG           SPG           BPG
        0.94          0.69          0.76          0.55          0.53          0.69          0.83
         PPG index_rating
        0.69          0.69
```

Figure 16: The Kaiser-Meyer-Olkin factor adequacy

Now, overall MSA increased to 0.70 and there is no variable whose MSA is too low. As a result, I can apply factor analysis to my data.

For choosing number of factors, parallel analysis is used. It involves generating random correlation matrices and after factor analyzing them, comparing the resulting eigenvalues to the eigenvalues of the observed data

```
> parallel<- fa.parallel(data.st.d,fm="ML",fa = "fa")
Parallel analysis suggests that the number of factors =  3
```

Figure: 17: Parallel analysis

Figure 17 suggests that number of factors should be chosen 3.

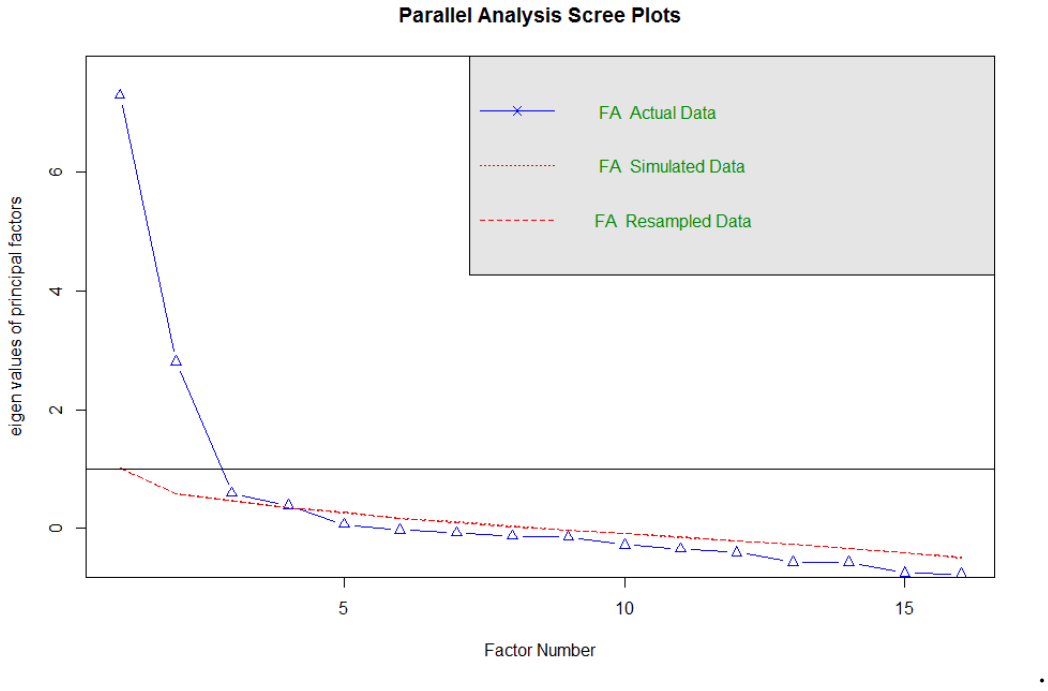

Figure 18: Parallel Analysis Scree Plot

In *Figure 18*, the two blue lines show you the observed eigenvalues. The red dotted lines show you the random eigenvalues or the simulated data line. Each point on the blue line which stays above the simulated data line (red line) is a factor to extract. According to Figure x4, you can see that 3 factors stay above the corresponding simulated data line.

```
> fit <- fa(data.st.d, nfactors = 3, max.iter = 100, rotate = "varimax", fm = "ML")
> fit$communalities
         GP          MPG          FGM          FGA         X3PM         X3PA          FTM
0.03928402   0.67843457   0.99500000   0.94098049   0.94253843   0.97765639   0.99500000
        FTA          TOV          ORB          DRB          APG          SPG          BPG
0.96590538   0.52193947   0.59847898   0.39747095   0.35316272   0.18639642   0.43286407
        PPG index_rating
0.99500000   0.86309244
```

Figure 19: The variance percentages for each variable

The percentage of variance that can be explained by the retained factors for each variable can be seen in *Figure 19*. All of the explained percentage of variance in variables are high enough except GP (Games Played).

```
> fit <- fa(data.st.d, nfactors = 3, max.iter = 100, rotate = "varimax", fm = "ML")
> fit
Factor Analysis using method =  ml
Call: fa(r = data.st.d, nfactors = 3, rotate = "varimax", max.iter = 100,
    fm = "ML")
Standardized loadings (pattern matrix) based upon correlation matrix
              ML1  ML2   ML3    h2     u2 com
GP           0.16 0.11  0.05 0.039 0.9609 2.0
MPG          0.72 0.39 -0.08 0.678 0.3218 1.6
FGM          0.91 0.37  0.18 0.996 0.0039 1.4
FGA          0.86 0.42 -0.14 0.941 0.0590 1.5
X3PM         0.51 0.05 -0.83 0.943 0.0575 1.7
X3PA         0.51 0.12 -0.84 0.978 0.0223 1.7
FTM          0.38 0.92  0.10 0.995 0.0048 1.4
FTA          0.39 0.87  0.24 0.966 0.0341 1.6
TOV          0.41 0.58 -0.14 0.522 0.4783 1.9
ORB          0.10 0.09  0.76 0.598 0.4015 1.1
DRB          0.38 0.19  0.47 0.398 0.6024 2.3
APG          0.37 0.33 -0.33 0.354 0.6460 3.0
SPG          0.41 0.12 -0.08 0.187 0.8125 1.2
BPG          0.16 0.07  0.63 0.432 0.5676 1.2
PPG          0.84 0.54  0.00 0.997 0.0030 1.7
index_rating 0.69 0.54  0.30 0.863 0.1368 2.3

                       ML1  ML2  ML3
SS loadings           4.76 3.19 2.94
Proportion Var        0.30 0.20 0.18
Cumulative Var        0.30 0.50 0.68
Proportion Explained  0.44 0.29 0.27
Cumulative Proportion 0.44 0.73 1.00
```

Figure 20: Summary of Factor Analysis

By looking at our factor loadings in *Figure 20*, FGM, FGA, PPG, MPG, Index Rating, SPG and APG have high factor loadings on Factor 1 (ML1).

FTM, FTA and TOV have its highest factor loadings on Factor 2 (ML2).

X3PA, X3PM, ORB, BPG and DRB have its highest factor loadings on Factor 3 (ML3).

Let visualize the analysis and see the relationship between variables and factors.

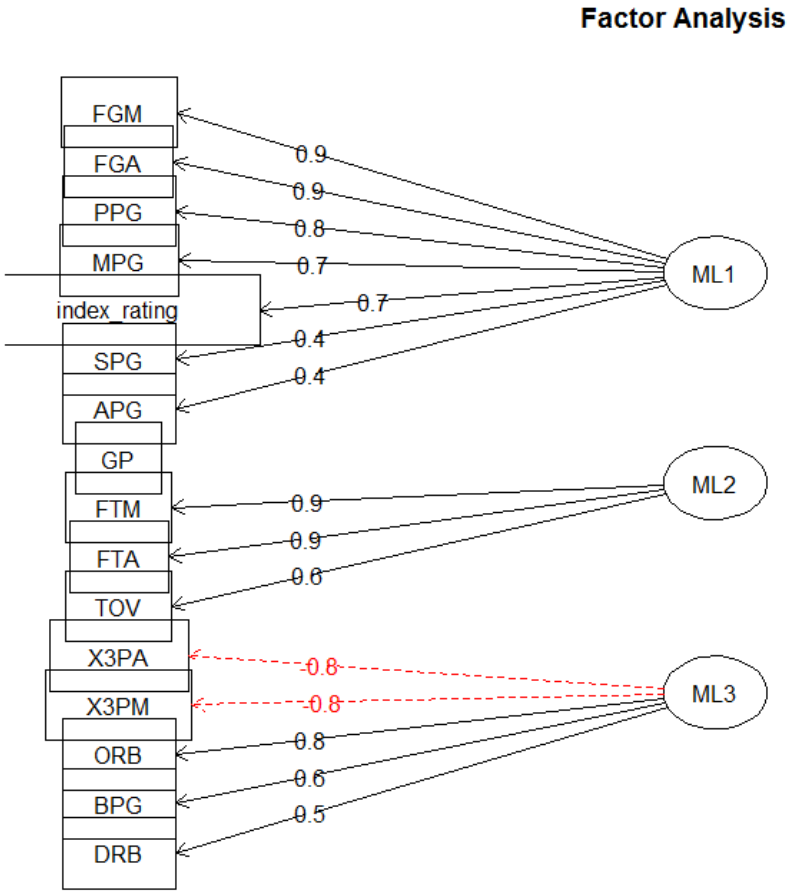

Figure 21: Explanation relations of variables by factors

As you can see in the *Figure 21*, GP cannot be explained by the factors. In real life, that is something can be expected since Games Played and other variables such as Field Goals Made, Free Throw Made might not be explained by the same common factors.

The figure also shows the eigenvalues for each variable and factor and which variable is explained by which factor. To illustrate, Factor 1 explains FGM, FGA, PPG, MPG, Index Rating, SPG and APG.

6. Classification

6.1 Classification on the basis of Team (Final Four Team or NOT)

In this section, individuals are tried to identify whether they are Final Four Team or not by using machine learning algorithms.

```
> sapply(mydata, class)
          GP          MPG          FGM          FGA          FG.         X3PM
   "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
        X3PA         X3P.          FTM          FTA          FT.          TOV
   "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
          PF          ORB          DRB          RPG          APG          SPG
   "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
         BPG          PPG           FF index_rating
   "numeric"    "numeric"     "factor"    "numeric"
```

Figure 22: Class of the dataset

Figure 22 shows that the data include 1 factor variable and 21 continuous variables.

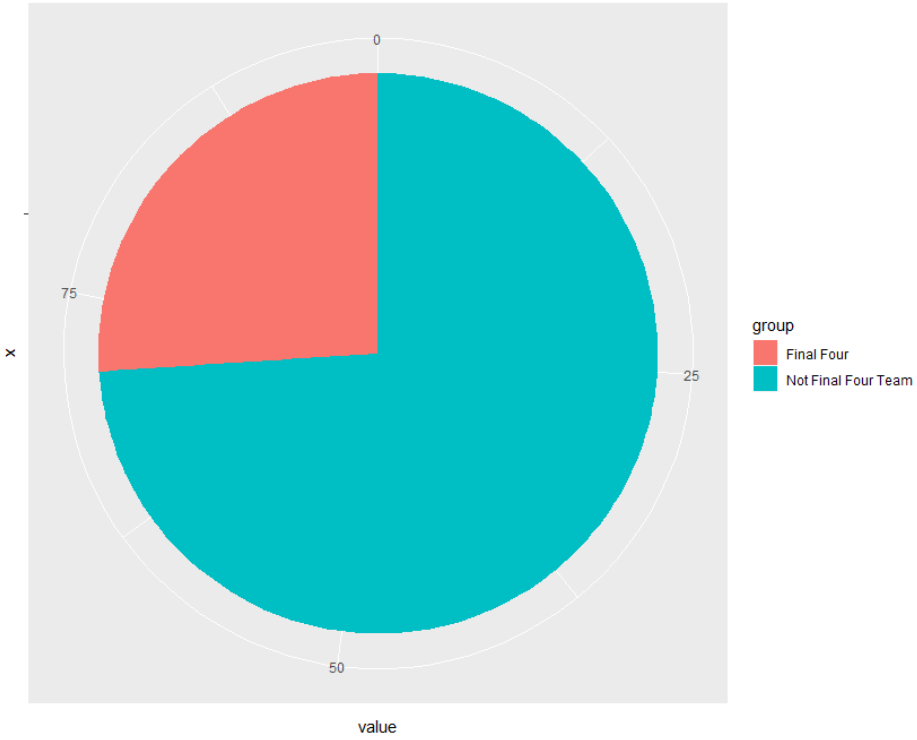

Figure 23: Pie Chart for the variable Final Four

As you can see in the in *Figure 23*, in the train set 26% of the observations belongs to Final Four Team and almost 74% of the observations do not belong to Final Four Team.

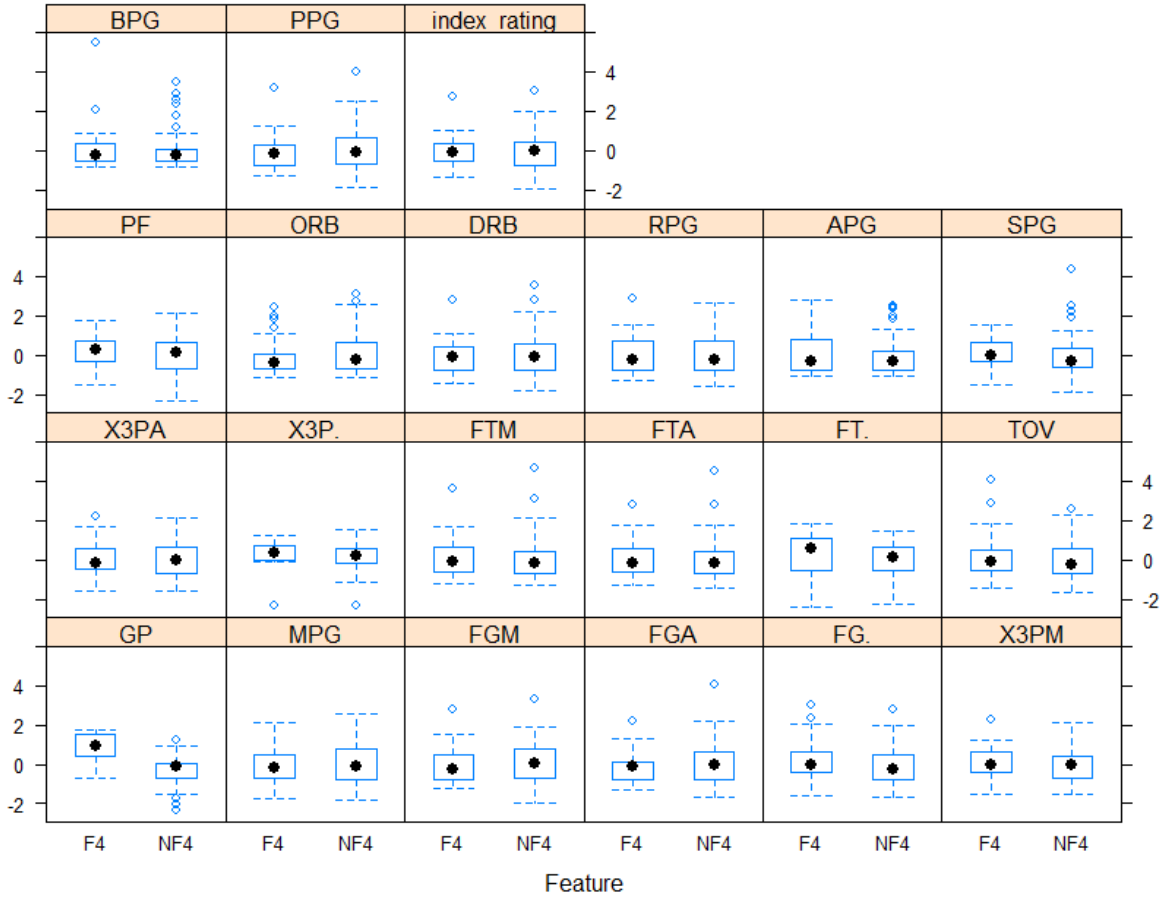

Figure 24: Feature Box Plot of the variables

As you can see in the Feature Box Plot, there is no obvious difference between mean of Final Four Team and mean of Not Final Four Team. Only Games Played is difference, and it is not something can be used to identify individuals as a Final Four Team or not. Now, let see density plot and try to find variables which can be used to classify individuals.

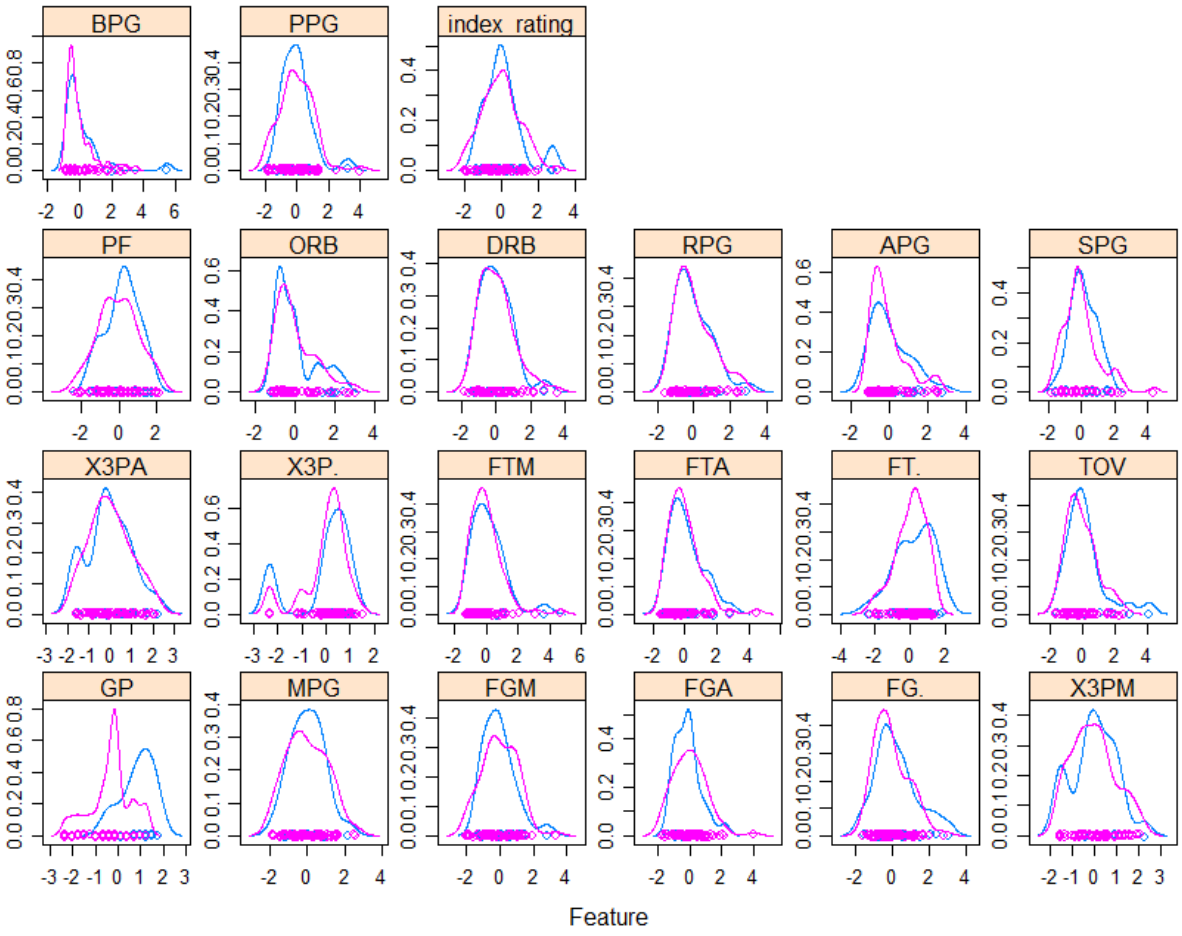

Figure 25: Feature Density Plot of the variables

As you can see in the *Feature Density Plot*, almost all variables show same pattern for Final Four Team or Not Final Four Team, but only Games Played shows different pattern for Final Four Team and Not Final Four Team.

According to both, Box Plot and Density Plot, there is no obvious variable which can help to classify individuals as a Final Four Team or Not Final Four Team. For this reason, while constructing the model all variables are used to classify individuals, and accuracy, specificity and sensitivity is checked.

5 models are used with 3 different types of algorithms which are linear, nonlinear and advanced. The first model is built by using Latent Dirichlet Allocation, the second model is built by Classification and Regression Trees, the third model is built by K-Nearest Neighbors, the fourth model is built by Support Vector Machines Radial, the fifth model is built by Random Forest. In all models, 10-fold cross validation is used to get the most accurate model. That means, for example in Latent Dirichlet Allocation, the train set is split into 10 different train (yes again) and test set. After that, the function choses the most accurate model and store it into fit.lda. With this way, 5 best of the best models can be found.

Now, let compare the accuracy of those models.

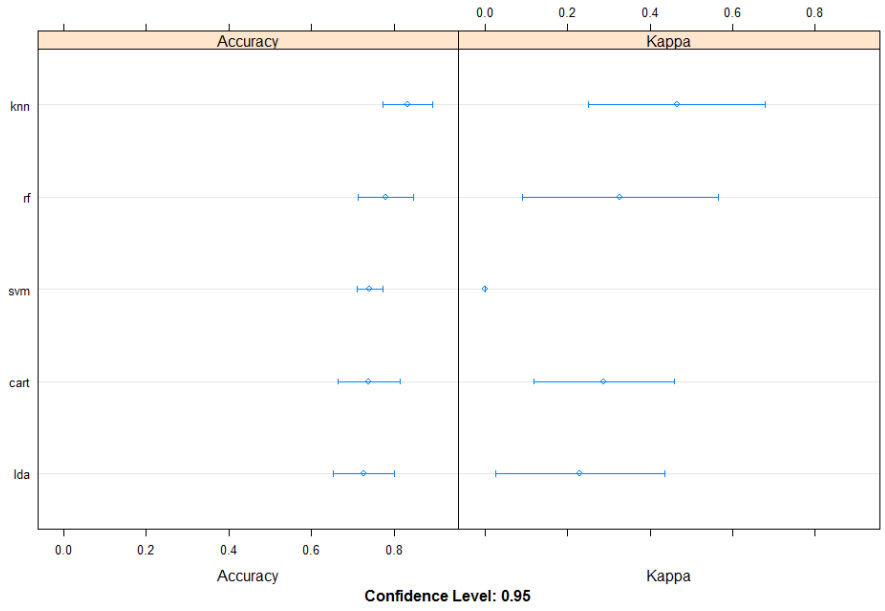

Figure 26: *Accuracy and Kappa Plot*

Figure 26 clearly shows that K-Nearest Neighbors model has the highest accuracy among the models with a nearly %80. It also has the highest kappa which gives more reliable result than accuracy since kappa does not count the true prediction which due to major class of the data set will be truly predicted with a high percentage. The kappa which is 0.42 means fair level of agreement.

```
> print(fit.knn)
k-Nearest Neighbors

96 samples
21 predictors
 2 classes: 'F4', 'NF4'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 86, 87, 86, 87, 87, 86, ...
Resampling results across tuning parameters:

  k  Accuracy   Kappa
  5  0.8133333  0.4274342
  7  0.8133333  0.3828321
  9  0.8333333  0.4649303

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 9.
```

Figure 27: K-Nearest Neighbors

According to *Figure 27*, the K-Nearest Neighbors model used all continuous variable as predictors and the most accurate result is given by the 9th model.

```
> # estimate skill of KNN on the test dataset
> predictions <- predict(fit.knn, mydata.test)
> confusionMatrix(predictions, mydata.test$FF)
Confusion Matrix and Statistics

          Reference
Prediction F4 NF4
       F4   2   1
       NF4  4  16

               Accuracy : 0.7826
                 95% CI : (0.563, 0.9254)
    No Information Rate : 0.7391
    P-Value [Acc > NIR] : 0.4209

                  Kappa : 0.3275
 Mcnemar's Test P-Value : 0.3711

            Sensitivity : 0.33333
            Specificity : 0.94118
         Pos Pred Value : 0.66667
         Neg Pred Value : 0.80000
             Prevalence : 0.26087
         Detection Rate : 0.08696
   Detection Prevalence : 0.13043
      Balanced Accuracy : 0.63725

       'Positive' Class : F4
```

Figure 28: Confusion Matrix & Statistics

The accuracy is 78%. That means the model can classify 78% of the observations truly.

Sensitivity is 33%, so the model is able to detect 33% of the individuals whore are in Final Four Team. The model misses 67% of the individuals whore are in Final Four Team.
The model has 94% specificity. In other words, 16 individuals out of 17 individuals with Not Final Four Team are truly Not Final Four Team and 1 individual classified as Final Four Team which he doesn't belong.

The Positive Predicted Value is 66%. That means among the individuals who are identified by the model as Final Four Team, 66% of them belong to Final Four Team.

The Negative Predicted value is 80%. That means among the individuals who are identified by the model as Not Final Four Team, 80% of them belong to Not Final Four Team.

All in all, model accuracy is very high, sensitivity is very low, and specificity is very high. That means the models is very good at predicting Not Final Four Team, but it is very bad at prediction Final Four Team. It is most probably because of the d This is because individuals are not distributed equally according to their Final Four variable. (%25 is Final Four Team, %75 Not Final Four Team).

6.2 Classification on the basis of Position

In this section, position of individuals is tried to identify to. Types of position are Center represented by C, Forward represented by F, Guard represented by G.

```
> mydata$GP<-as.numeric(mydata$GP)
> sapply(mydata, class)
          GP          MPG          FGM          FGA          FG.         X3PM
   "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
        X3PA         X3P.          FTM          FTA          FT.          TOV
   "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
          PF          ORB          DRB          RPG          APG          SPG
   "numeric"    "numeric"    "numeric"    "numeric"    "numeric"    "numeric"
         BPG          PPG     Position index_rating
   "numeric"    "numeric"     "factor"    "numeric"
```

Figure 29:

Figure 29 shows that the data include 1 factor variable which is Position and 21 continuous variables.

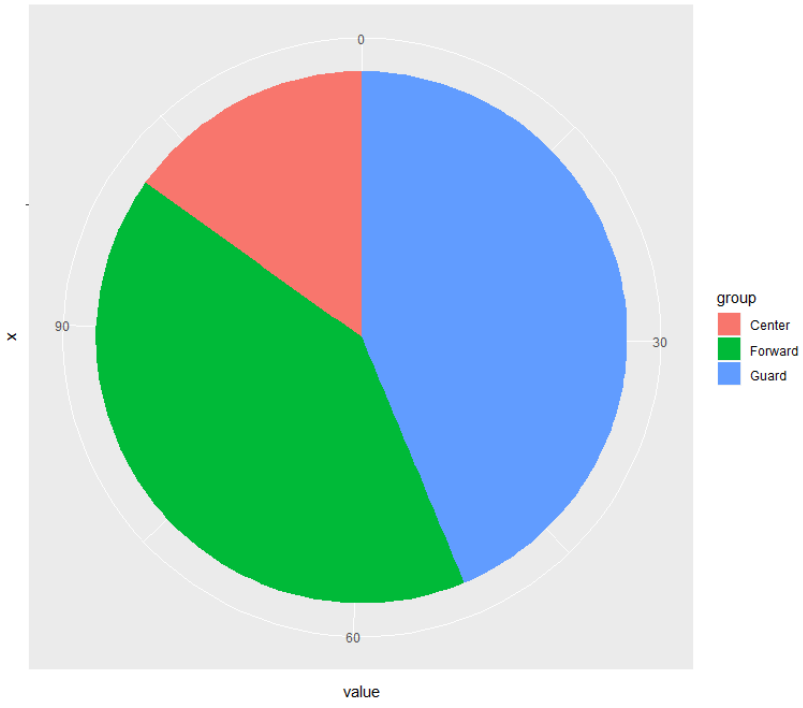

Figure 30: Pie Chart for Position variable

As you can see in the in *Figure 30*, in the train set 15% of the observations are Center, 41% of the observations are Forward and 43% of the observations are Guard.

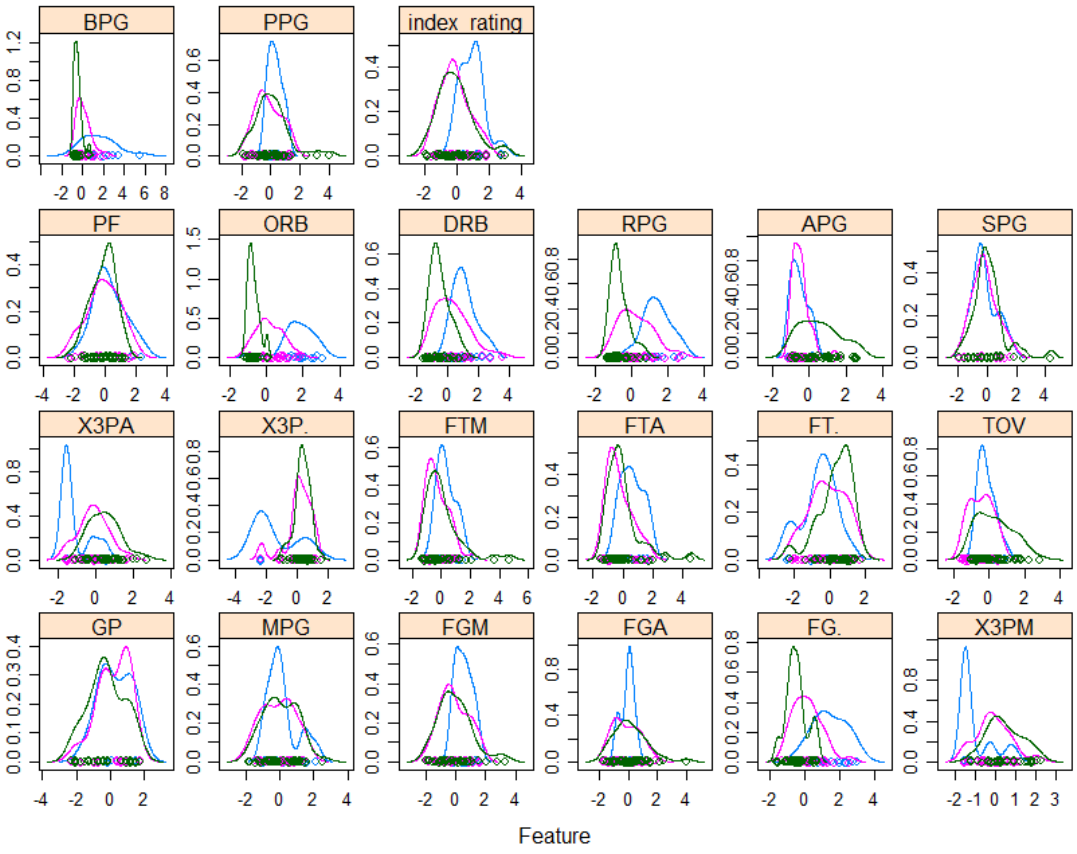

Figure 31: Feature Density Plot

As you can see in the Feature Density Plot, BPG, ORB, DRB, X3PA, X3PM shows different pattern for each Position. They will be more active than other variables to classify individuals as a Guard, Center or Forward. Let us build models with the train data set and all variables.

In the previous part, 5 models is used with 3 different types of algorithms which are linear, nonlinear and advanced. The first model is built by using Latent Dirichlet Allocation, the second model is built by Classification and Regression Trees, the third model is built by K-Nearest Neighbors, the fourth model is built by Support Vector Machines Radial, the fifth model is built by Random Forest. In all models, 10-fold cross validation is used to get the most accurate model. That means, for example in Latent Dirichlet Allocation, our train set is splitted into 10 different

train (yes again) and test set. After that, the function choses the most accurate model and store it into *fit.lda*. With this way, 5 best of the best models can be found.

Now, let compare the accuracy of those models.

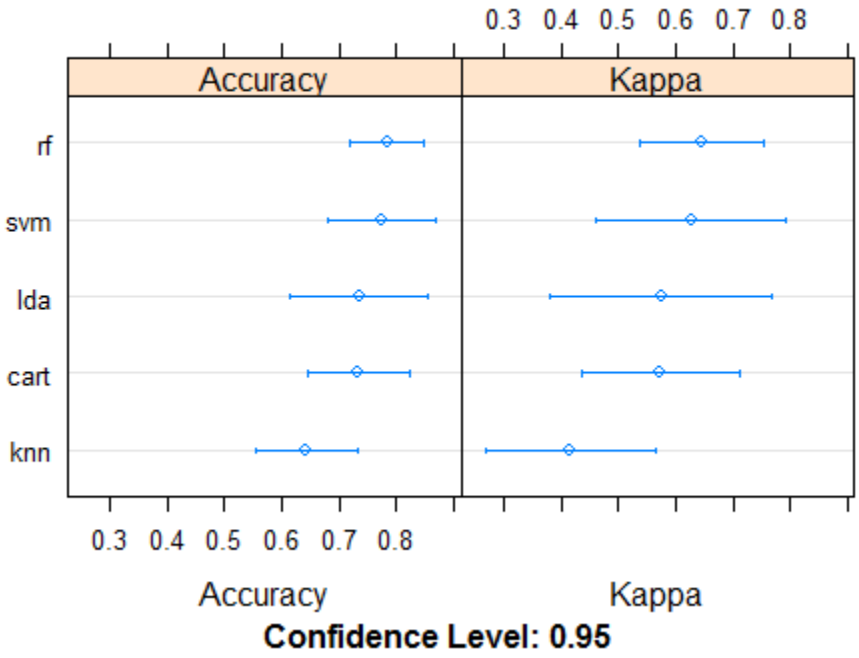

Figure 32: Accuracy and Kappa Plot

It is clear to see that in *Figure 32*, Random Forest model has the highest accuracy in all models with an almost %80 accuracy. It also has the highest kappa which gives more reliable result than accuracy since kappa does not count the true prediction which due to major class of the data set will be truly predicted with a high percentage. The kappa which is 0.64 means good level of agreement.

```
> print(fit.rf)
Random Forest

97 samples
21 predictors
 3 classes: 'C', 'F', 'G'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 87, 88, 87, 87, 88, 87, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
   2    0.7744444  0.6283715
  11    0.7722222  0.6238091
  21    0.7833333  0.6446582

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 21.
```

Figure 33: Random Forest Output for Positions

It is possible to say that the Random Forest model used all countinuios variable as predictors and the most accurate result is given by the 21th model.

```
> # estimate skill of RF on the test dataset
> predictions <- predict(fit.rf, mydata.test)
> confusionMatrix(predictions, mydata.test$Position)
Confusion Matrix and Statistics

          Reference
Prediction  C  F  G
         C  3  0  0
         F  0  7  0
         G  0  2 10

Overall Statistics

               Accuracy : 0.9091
                 95% CI : (0.7084, 0.9888)
    No Information Rate : 0.4545
    P-Value [Acc > NIR] : 1.055e-05

                  Kappa : 0.8493
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: C Class: F Class: G
Sensitivity            1.0000   0.7778   1.0000
Specificity            1.0000   1.0000   0.8333
Pos Pred Value         1.0000   1.0000   0.8333
Neg Pred Value         1.0000   0.8667   1.0000
Prevalence             0.1364   0.4091   0.4545
Detection Rate         0.1364   0.3182   0.4545
Detection Prevalence   0.1364   0.3182   0.5455
Balanced Accuracy      1.0000   0.8889   0.9167
```

Figure 34: Confusion Matrix & Statistics

The accuracy is 90%. That means the model can classify 90% of the observations truly. The Sensitivity for Class C is 100%, so the model is able to detect 100% of the individuals whore are in Class C. The model misses none of them.
The Sensitivity for Class F is 77%, so the model is able to detect 100% of the individuals whore are in Class F. The model misses only 23% of them. The Sensitivity for Class G is 100%, so the model is able to detect 100% of the individuals whore are in Class F. The model misses none of them. The Positive Predicted Value for Class G is 83%. That means among the individuals who are identified by the model as Class G, 83% of them are in actually in Class G.

7. K-Means Clustering

```
K-means clustering with 3 clusters of sizes 38, 52, 29

Cluster means:
        GP      MPG      FGM      FGA       FG.      X3PM     X3PA      X3P.
1 27.97368 16.77632 1.952632 4.410526 0.4424474 0.7236842 1.915789 0.3433158
2 32.09615 21.52885 3.130769 6.482692 0.4918462 0.8211538 2.161538 0.3015962
3 30.86207 27.74483 4.479310 9.906897 0.4572069 1.3827586 3.751724 0.3430000
        FTM       FTA       FT.       TOV       PF       ORB       DRB       RPG
1 0.7973684 1.073684 0.7256579 0.8447368 1.857895 0.5394737 1.492105 2.044737
2 1.4711538 1.953846 0.7571538 1.2788462 2.111538 1.1076923 2.628846 3.730769
3 2.3620690 2.982759 0.7831379 1.8724138 2.200000 0.8827586 2.896552 3.779310
       APG       SPG       BPG       PPG       IR
1 1.186842 0.5052632 0.1421053  5.431579  4.939211
2 1.755769 0.7076923 0.3615385  8.571154  9.826154
3 3.220690 0.9310345 0.3310345 12.713793 13.865517

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 2 3 3 2 3 3 3 2 2 2 2 2 2 2 2 3
 [41] 2 2 2 2 2 2 3 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 1 1 2 1 1
 [81] 2 1 1 1 2 2 1 2 1 1 1 1 1 2 2 2 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Within cluster sum of squares by cluster:
[1] 1250.572 1700.614 1606.586
 (between_SS / total_SS =  55.0 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

Figure 35: Summary of K-Means Clustering

In order to see if there is a significant difference between the players' statistics with respect to their positions, K-means clustering technique is applied. *Figure 35* shows the k-means clustering outputs when k is chosen as 3 since there are 3 positions in the data.

```
> table(data.cluster3$cluster,datadata$Position)

    C  F  G
1   1 17 20
2  14 20 18
3   3 12 14
```

Figure 36:Confusion Matrix for Position variable

In the clustering table columns represent the players' position and the rows represent each cluster. *Figure 36* shows that k-means clustering method is not handy to categorise the players according to their positions.

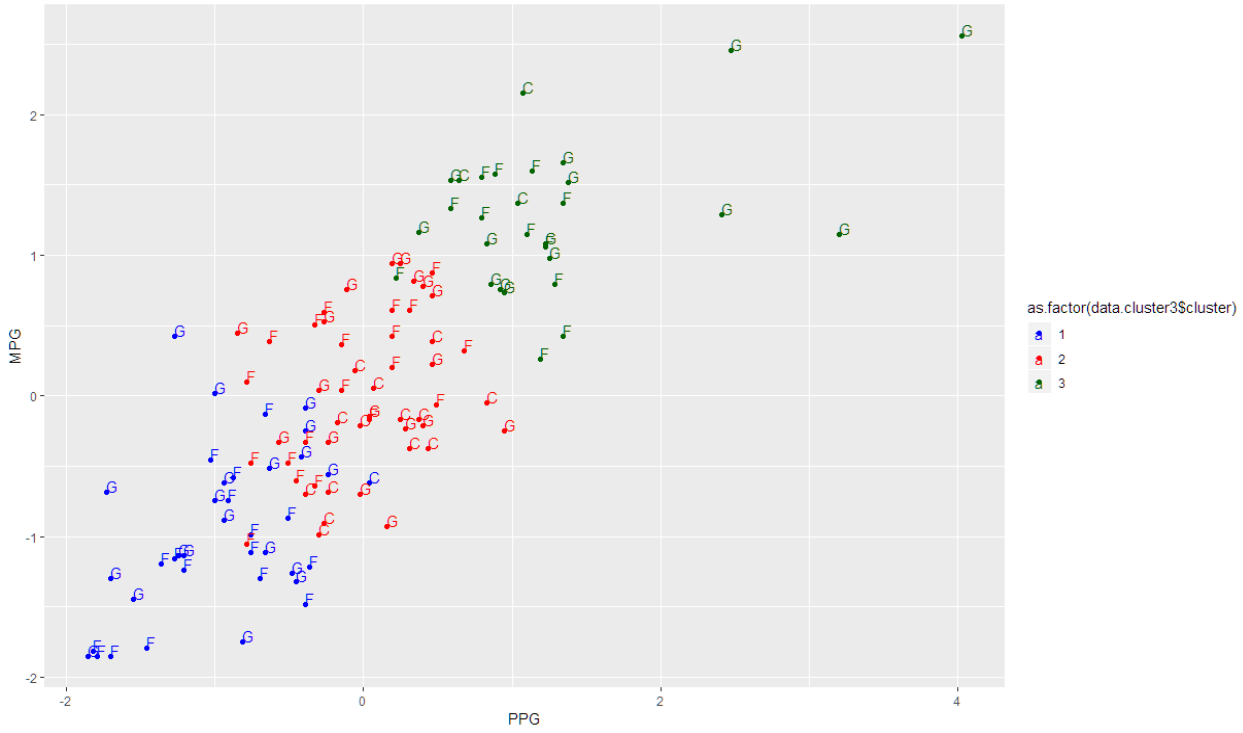

Figure 37: K-Means Clustering Plot

After the k-means clustering, in the *Figure 37*, it can be seen k-means clustering does not help to separate the players with respect to their positions. However, by looking it the *Figure 40*, this k-means is actually useful to cluster the players with their importance for their teams. The green ones are the players who are playing more in average and their contribution to score is higher than the others, whereas the players with blue dots are the ones who are playing above the average and their scoring ability are not as good as the red or green ones. Additionally, in the 2nd cluster, represented as red, average players take place. In other words, they are playing on average and scoring on average again.

8. Regression

Stepwise Model Selection

```
No more variables to be added.

Variables Entered:

✔ FGA
✔ SPG
✔ index_rating
✔ PF
✔ X3PM
✔ PPG
✔ X3PA
✔ APG
✔ BPG
✔ FG.
✔ ORB


Final Model Output
------------------

                         Model Summary
---------------------------------------------------------------
R                       0.926       RMSE                 0.405
R-Squared               0.858       Coef. Var          685.809
Adj. R-Squared          0.839       MSE                  0.164
Pred R-Squared          0.810       MAE                  0.298
---------------------------------------------------------------
 RMSE: Root Mean Square Error
 MSE: Mean Square Error
 MAE: Mean Absolute Error

                              ANOVA
-----------------------------------------------------------------------
              Sum of
             Squares        DF    Mean Square      F          Sig.
-----------------------------------------------------------------------
Regression     82.404        11          7.491    45.692     0.0000
Residual       13.608        83          0.164
Total          96.012        94
-----------------------------------------------------------------------
```

Figure 38: Summary of the model with the MPG as a response

In the *Figure 38,* Stepwise model selection technique suggests in order to predict Minutes Per Game for each player, those variables (FGA, SPG, PF, DRB, APG, 3PM, ORB, Position) should be taking accound and the others will be negligible. Hence, by using those 9 variables provides opportunuty roughly 80% to explain the variation in the response variable. Additionally, Anova result shows that the model is significant.

Parameter Estimates

model	Beta	Std. Error	Std. Beta	t	Sig	lower	upper
(Intercept)	0.052	0.042		1.224	0.224	-0.032	0.135
FGA	0.725	0.248	0.742	2.925	0.004	0.232	1.217
SPG	0.137	0.062	0.125	2.196	0.031	0.013	0.261
index_rating	0.421	0.206	0.442	2.044	0.044	0.011	0.830
PF	0.186	0.055	0.188	3.387	0.001	0.077	0.295
X3PM	0.746	0.208	0.727	3.583	0.001	0.332	1.160
PPG	-0.512	0.332	-0.532	-1.542	0.127	-1.172	0.148
X3PA	-0.592	0.245	-0.581	-2.413	0.018	-1.081	-0.104
APG	0.180	0.087	0.183	2.071	0.041	0.007	0.354
BPG	0.084	0.067	0.086	1.244	0.217	-0.050	0.217
FG.	-0.135	0.090	-0.128	-1.496	0.138	-0.314	0.044
ORB	0.114	0.106	0.115	1.077	0.285	-0.097	0.325

Selection Summary

Step	Variable Entered	R-Square	Adj. R-Square	C(p)	AIC	RMSE
1	FGA	0.7005	0.6973	70.4092	162.0647	0.5560
2	SPG	0.7517	0.7463	44.8460	146.2757	0.5091
3	index_rating	0.7822	0.7751	30.3608	135.7892	0.4793
4	PF	0.8029	0.7941	21.2417	128.3333	0.4586
5	X3PM	0.8179	0.8076	15.1665	122.8233	0.4433
6	PPG	0.8442	0.8336	2.9655	109.9787	0.4123
7	X3PA	0.8481	0.8359	2.8566	109.5622	0.4094
8	APG	0.8521	0.8383	2.7353	109.0679	0.4064
9	BPG	0.8540	0.8385	3.7044	109.8316	0.4061
10	FG.	0.8563	0.8392	4.4537	110.3098	0.4053
11	ORB	0.8583	0.8395	5.3869	110.9922	0.4049

Figure 39: Summary of selection and the parameter estimates

Moreover, *Figure 39* illustrates the estimations for the parameters in the model. In other words, this provides the coefficients of each variable in the model. Also, in the Selection Summary section, for every variable that comes in the model, table shows how much is the contribution that particular varaiable makes.

```
        Shapiro-Wilk normality test

data:  resid
W = 0.98846, p-value = 0.5796
```

Figure 40: Shapiro-Wilk Normality Test

Shapiro – Wilk Normality Test result is shown in *Figure 40*, since the p-value of the test is (0.5796) is bigger than the significance level (0.05), it can be concluded that the normality assumption for the residuals is hold.

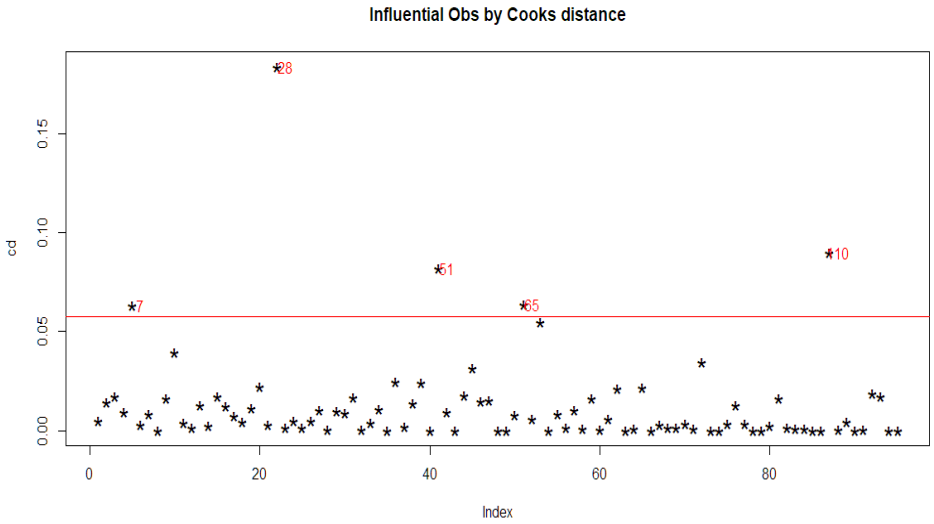

Figure 41: Cook' Distance Plot

According to the Cook's distance result, there are 5 influencial observations in this data. By simply looking at the *Figure 41,* it can be said that the 4 of 5 influencial points are not quite higher than the limit but the 28th observarion is questionable. Let's check whether it is an outlier or not.

```
> outlierTest(model3MPG)
No Studentized residuals with Bonferonni p < 0.05
Largest |rstudent|:
   rstudent unadjusted p-value Bonferonni p
28 3.084613          0.0027588      0.26209
```

Figure 42: Outlier Test

According to *Figure 42,* also the 28th observation is an outlier since p value is bigger than 0.05. For this reason the 28th observation is deleted and continued with principle component analysis.

9. Principle Component Analysis

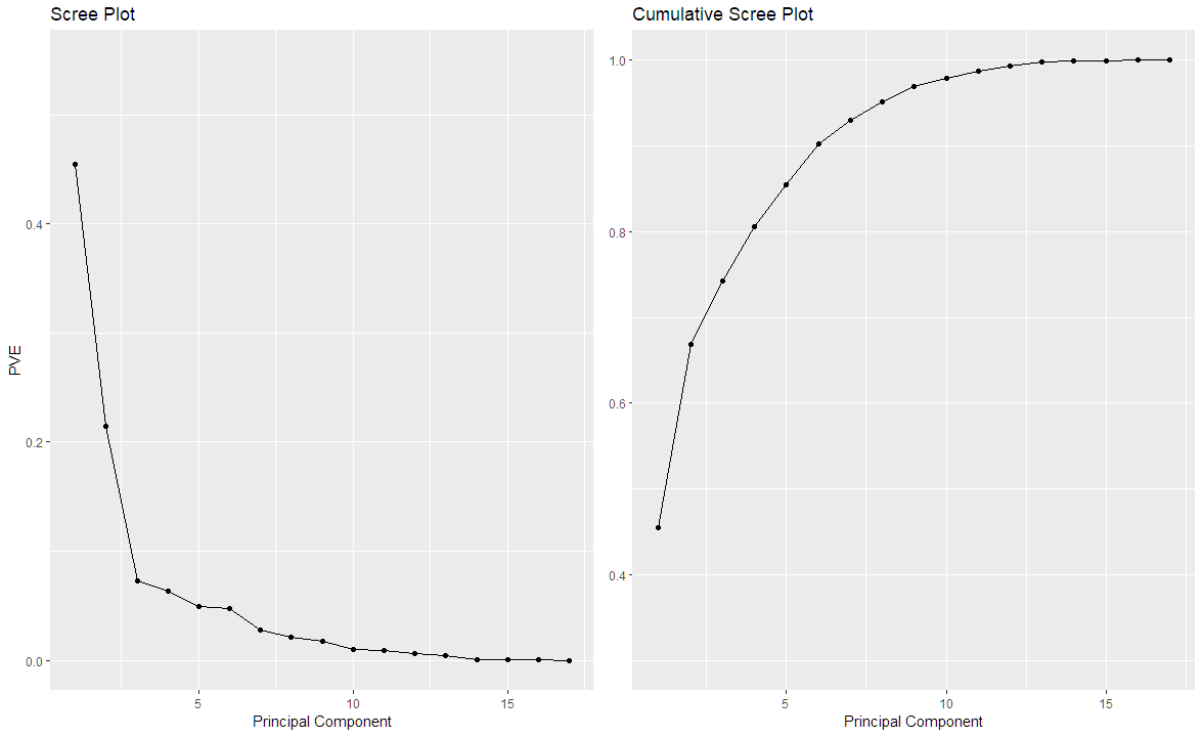

Figure 43: Scree Plots

According to the scree plot in *Figure 43,* it is suggested to take first 5 Principle Components since after the 5th one, in the there is not a significant increase in the percentage of explained variance. Moreover, on the right hand side of the *Figure x,* it can be seen that by using only 5 Principle Components, nearly 80% of the variation can be explained.

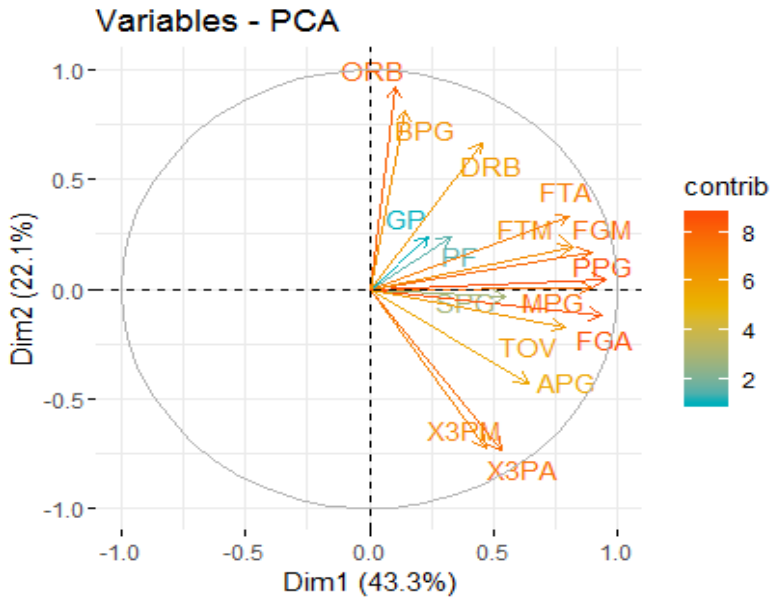

Figure 44: Biplot of Principal Component Analysis

As shown in *Figure 44*, the first principle componant explains 43.3% of the variation in the data. Also, the second principle component explains the 22.1% of the variation. Variables that coloured as blue are the least contributors and the red ones have the highest contribution while explaining the variation in the data.

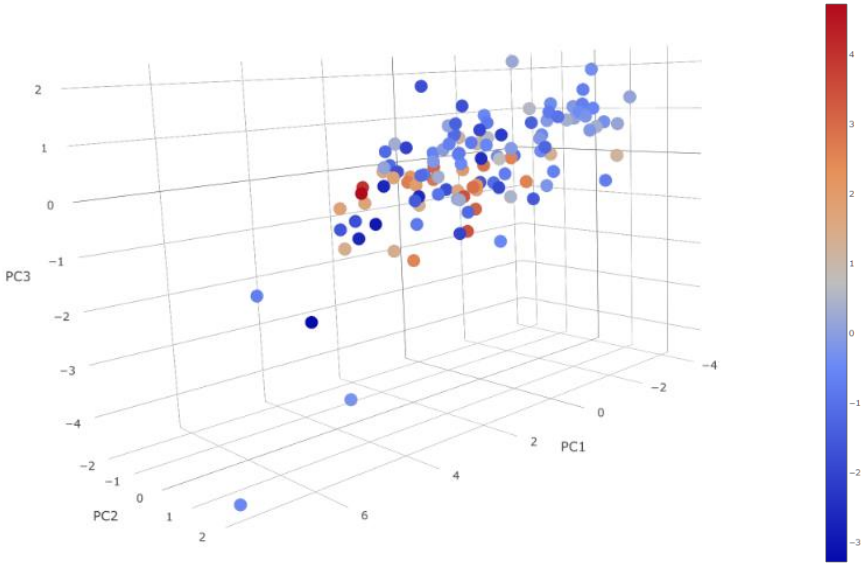

Figure 45: Scatter Plot of PC's

In *Figure 45*, there are 3 principle components are plotted in the same time and the color shows the 4th principle component.

9.1 Principle Component Regression:

```
Data:   X dimension: 95 20
        Y dimension: 95 1
Fit method: svdpc
Number of components considered: 5

VALIDATION: RMSEP
Cross-validated using 10 random segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps
CV           1.016   0.5364   0.5102   0.4781   0.4595   0.4616
adjCV        1.016   0.5334   0.5090   0.4750   0.4575   0.4603

TRAINING: % variance explained
     1 comps  2 comps  3 comps  4 comps  5 comps
X      36.89    63.27    70.40    76.72    82.15
MPG    72.34    76.35    80.03    80.89    80.93
```

Figure 46: Regression of the model with 5 Principal Components

After constracting regression model using 5 principle components, as suggested in *Figure xx*, the output is shown above in *Figure 46.* In that output, it can be seen that by using 5 Principle Components 82% of the variation in the variables can be explained and also rougly 81% of the variation in the response variable can be explained again by using 5 Principle Components.

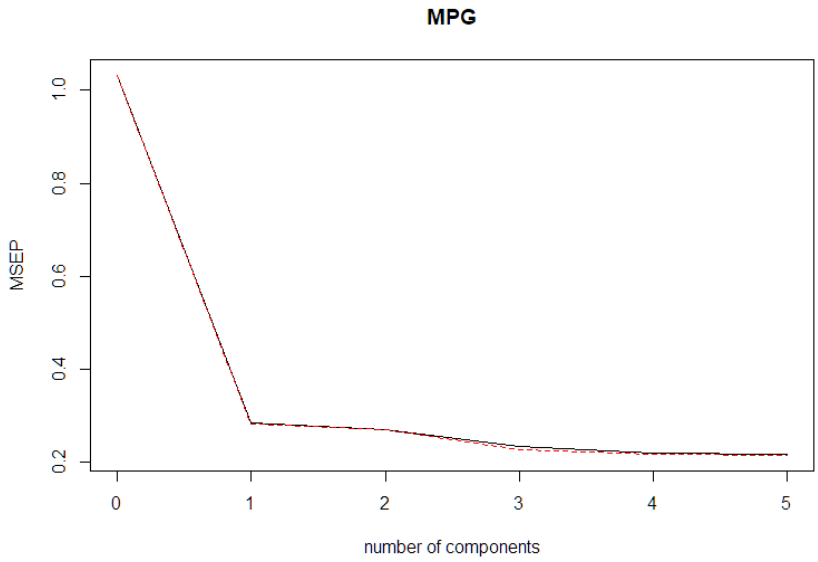

Figure 47: Mean Square Error Percentage of MPG for each PC

Figure 47 shows the MSE percentage for each Principle Component.

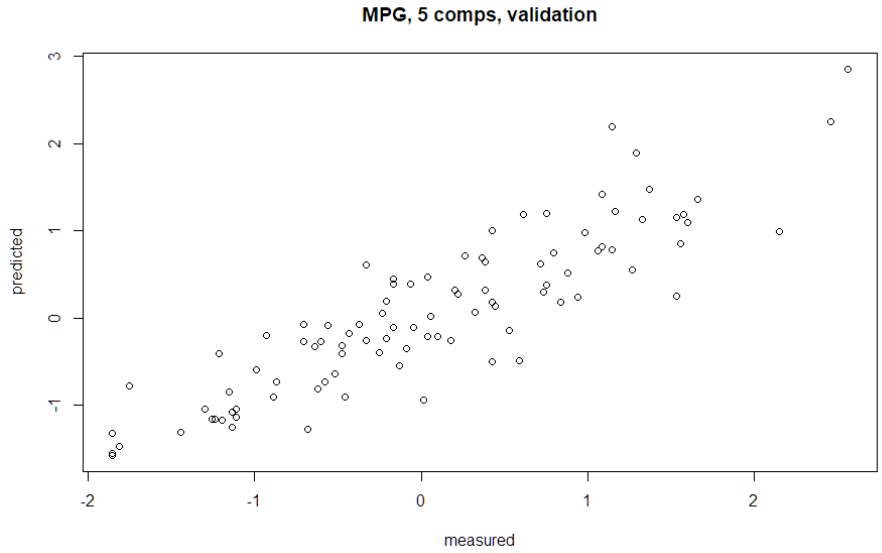

Figure 48: Plot of prediction and measured values of PC's

As it can be seen in the *Figure 48*, most of the measured values are predicted very close to their actual value since linear pattern can be seen.

10. Agglomerative Hierarchical Clustering

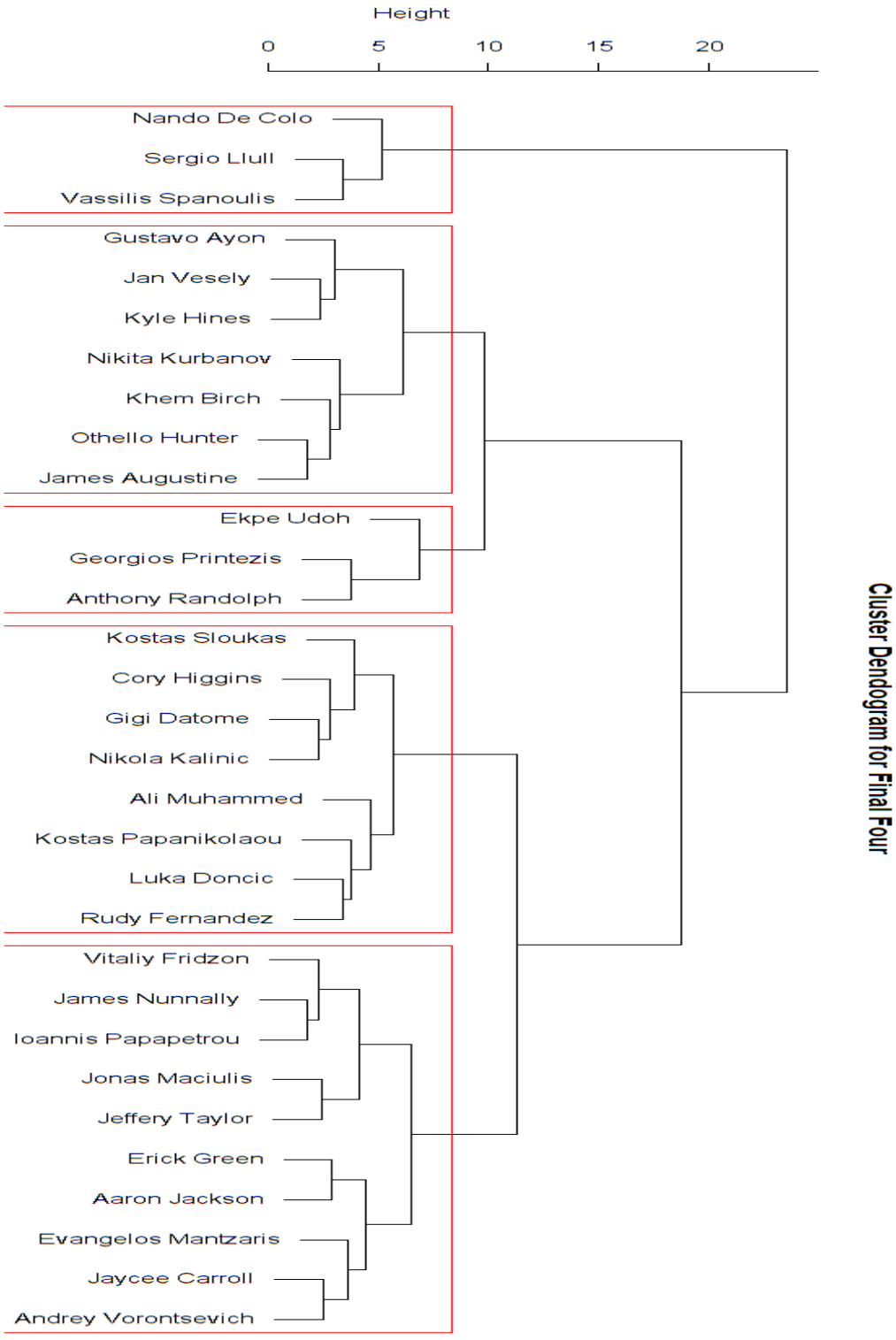

Figure 49: Cluster Dendogram for Final Four

In the *Figure 49,* there is a cluster dendogram for the players of the teams that are qualified for Final Four stage in Euroleage. In this agglomerative hierarchical clustering method, it can be easily seen that players who have similar features appear closer in the scheme. Also, in that clustering technique ward distance is used to create this dendogram. The reason why the Ward's Method is used is this method is most appropriate for quantitative variables, and not binary variables. To illustrate, let's consider the second cluster, in which Centers take place; also, Ayon – Vesely – Hines are considered as they have really close features. However, the Point Guards like Sloukas – Muhammed – Doncic etc. appear in another cluster as expected. The reason could be simple as it is, they are in different clusters probably because they have different abilities than Centers have. For instance, while Sloukas has great assists and 3 Point statistics, Hines has amazing Block and Rebound statistics. Additionally, the last cluster contains more like 6th players and bench scorers, in other words those are the players who are not in the starting five in most of the games.

There is one interesting cluster in this dendogram (*Figure 52*) which is the first cluster that contains DeColo – Llul – Spanoulis. Those are the leader players of the teams CSKA – Real Madrid – Olympiakos respectively which means in those teams critical decisions are made by those 3 players generally and they take much more responsibility than the others. Here is the interesting part of this analysis; although CSKA, Real Madrid and Olympiakos have one player in the "leaders cluster", there is not any player in this cluster from Fenerbahçe.

11. Conclusion

In this project, multivariate analysis techniques were applied to Euroleague Basketball Players' individual statistics during 2016-2017 season. Main point is to indicate the factors that made Fenerbahçe to reach championship. The dataset consists top 7 or 8 players from each of 16 teams that competed in 2016-2017 season in Euroleague.

Firstly, multivariate assumptions had to be checked in order to deep down in analysis. Unfortunately, multivariate normality assumption was not hold; so, transformation methods were applied but still this application was not enough to satisfy that assumption and further analysis conducted with assuming that all assumptions were hold. First analysis was estimating the index rating. To do this, multiple linear regression method was applied. Although normality and constant variance assumption were satisfied, there was a multicollinearity problem as expected since index rating is combination of other variables that means they are highly correlated with response variable. There was not a possible solution for this problem because if highly correlated variables were removed, there were no variables left to construct the model. So, index rating analysis was stopped.

Secondly, multinomial regression was applied to predict players' positions. After constructed multinomial model, over dispersion problem was appeared. After some effort on solving this problem, still over dispersion problem was exist. For this reason, factor analysis was applied to reduce the dimensionality so that positions can be well predicted. Factor analysis suggested positions can be explained by three factors; however, factors could not be named according to the variables that they explain. So, factor analysis was not suitable for predicting players' position.

Thirdly, to see if there is a statistically significant difference between players whose teams were qualified to final-four stage and those players whose weren't, some machine learning algorithms such as support vector machines radial and classification and regression trees were used. Although K-Nearest Neighbors model provided the highest accuracy and kappa, still provided kappa score was not enough to classify players whether they played in final-four or not since the data includes relatively smaller final four players compared to others, and the model tends to classify most of the players in to not final four players.

Fourthly, again machine learning algorithms were used to classify players according to their positions. In that analysis, random forest method was able to predict players' positions more accurate than other models, also, it has the highest kappa. This time kappa score was acceptable, and model was trustworthy. Logically, k-means clustering method should have clustered players based on their positions since each position shows different behaviors. For example, centers tend to grab more rebounds than guards, and guards are more likely to assist than centers. K-means clustering technique was applied to see whether there is a difference between players statistics based on their position or not. Although k-means clustering was not useful to separate players

based on their positions, it clustered them according to their importance for their teams. It revealed that minutes per game variable in the dataset is account for importance.

Fifthly, multiple linear regression analysis was conducted to estimate minutes per game variable. By using stepwise forward selection, 11 variables appeared to be significant for the model which explains great amount of variation in the MPG. After adequacy check, it seemed to there was a multicollinearity problem. So, principle component analysis was conducted to eliminate multicollinearity problem. By using five principle components, major share of the total variation can be explained. After selection, principle component regression analysis was conducted. By principle component regression model, it is possible to explain nearly as much as the variation explained by multiple linear regression model.

Sixthly, agglomerative hierarchical clustering method was applied for the players of the teams that are qualified for Final Four stage in Euroleage. In that clustering technique ward distance is used to create this dendogram so that players who have similar features could appear closer in the scheme. Appearently this clustering method was convenient to separate players according to their features.

Lastly, one of the clusters consists only players who are said to be leaders of their teams but there were not a player from Fenerbahçe. Consequently, relying on a single player do not make a team champion but team play does. This analysis ended up with that team play made Fenerbahçe won that championship.

12. References

Coghlan, A. (2019). *A Little Book of R For Multivariate Analysis*. [online] Media.readthedocs.org. Available at: https://media.readthedocs.org/pdf/little-book-of-r-for-multivariate-analysis/latest/little-book-of-r-for-multivariate-analysis.pdf [Accessed 18 Jan. 2019].

Epiville.ccnmtl.columbia.edu. (2019). *Epiville: How to Calculate Kappa*. [online] Available at: http://epiville.ccnmtl.columbia.edu/popup/how_to_calculate_kappa.html [Accessed 18 Jan. 2019].

Little-book-of-r-for-multivariate-analysis.readthedocs.io. (2019). *Using R for Multivariate Analysis — Multivariate Analysis 0.1 documentation*. [online] Available at: https://little-book-of-r-for-multivariate-analysis.readthedocs.io/en/latest/src/multivariateanalysis.html [Accessed 18 Jan. 2019].

Newonlinecourses.science.psu.edu. (2019). *14.7 - Ward's Method | STAT 505*. [online] Available at: https://newonlinecourses.science.psu.edu/stat505/node/146/ [Accessed 18 Jan. 2019].

Personality-project.org. (2019). [online] Available at: http://personality-project.org/r/psych/HowTo/factor.pdf [Accessed 18 Jan. 2019].

Prabhakaran, S. (2019). *How to detect heteroscedasticity and rectify it?*. [online] DataScience+. Available at: https://datascienceplus.com/how-to-detect-heteroscedasticity-and-rectify-it/ [Accessed 18 Jan. 2019].

Promptcloud.com. (2019). *Exploratory Factor Analysis in R | | PromptCloud*. [online] Available at: https://www.promptcloud.com/blog/exploratory-factor-analysis-in-r/ [Accessed 18 Jan. 2019].

R-statistics.co. (2019). *Outlier Treatment With R | Multivariate Outliers*. [online] Available at: http://r-statistics.co/Outlier-Treatment-With-R.html [Accessed 18 Jan. 2019].

Statpower.net. (2019). [online] Available at: http://www.statpower.net/Content/312/R%20Stuff/Exploratory%20Factor%20Analysis%20with%20R.pdf [Accessed 18 Jan. 2019].

13. Appendix

```
## Data preparation
data <- read.table("a.txt", header = TRUE, sep = "\t")
dim(data)
head(data,20)
data.names <- data[,c(2:3,24,25)]
data.num <- data[,c(4:7,9,10,12,13,15:18,20:23,26)]
data.num2 <- data[,-c(2:3,24:25)]
head(data.num)
head(data.names)
dim(data.names);dim(data.num)
data.s <- as.data.frame(round(scale(data.num),4))
data.st <- as.data.frame(cbind(data.names,data.st))
head(data.st)
data.st <- data.frame(data.names, data.s)
head(data.s)

## Correlation
M <- cor(data.s)
corrplot(M, method = "circle")
#corrplot(M, method = "color")
corrplot(M, method = "number")
corrplot(M, type = "upper", tl.pos = "d")
corrplot(M, add = TRUE, type = "lower", col = "Black", method = "number",
        diag = FALSE, tl.pos = "n", cl.pos = "n")

## Standardization Data
data2 <- data[,-c(1,2,3,25,26)]
data3 <- data.frame(round(scale(data2),4))
```

```
## Normality test
shapiro_test_df <- function(df, bonf= TRUE, alpha= 0.05) {
  l <- lapply(df, shapiro.test)
  s <- do.call("c", lapply(l, "[[", 1))
  p <- do.call("c", lapply(l, "[[", 2))
  if (bonf == TRUE) {
    sig <- ifelse(p > alpha / length(l), "H0", "Ha")
  } else {
    sig <- ifelse(p > alpha, "H0", "Ha")
  }
  return(list(statistic= s, p.value= p,significance= sig,
              method= ifelse(bonf == TRUE, "Shapiro-Wilks test with Bonferroni Correction",
                             "Shapiro-Wilks test without Bonferroni Correction")))
}
shapiro_test_df(a)


## MVN
mvn(a, subset=NULL, mvnTest = c("mardia"), covariance = TRUE, tol = 1e-25, alpha = 0.5, scale =
FALSE,
    desc = TRUE, transform = "sqrt", R = 1000, univariateTest = c("SW"), univariatePlot = "none",
    multivariatePlot = "none", multivariateOutlierMethod = "none",
    showOutliers = FALSE, showNewData = FALSE)
warnings()
result<-mvn(a,mvnTest = "royston") #bunu kullan
#r2
r2=sreg$r.squared
r2
r2adj=sreg$adj.r.squared
r2adj
r2pred=1-PRESS/SST
r2pred
```

```
# Simple regression
fit=lm(IR ~ . , data=a)
summary(fit)

## Multicollinearity
vif(fit)
library(mctest)
omcdiag(x,y)
imcdiag(x,y)

## Outlier /Multivariate(?)
mod <- lm(IR ~ ., data=a)
cooksd <- cooks.distance(mod)
plot(cooksd, pch="*",col="dodgerblue3", cex=2, main="Influential Obs by Cooks distance")  # plot
cook's distance
abline(h = 4*mean(cooksd, na.rm=T), col="violetred4")  # add cutoff line
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4*mean(cooksd,
na.rm=T),names(cooksd),""), col="red")  # add labels
outlierTest(fit)
reg = lm(unlist(y)~x)
reg
sreg=summary(reg)
areg=anova(reg)
sreg
areg
SST=sum((areg$'Sum Sq')[1:length(areg$'Sum Sq')])
SST
c.di=cooks.distance(reg) #cooksd
c.di
hii=lm.influence(reg)$hat #leverages
hii
ei=residuals(reg) #Residuals
ei
```

```
di=studres(reg) #studentized Residuals
press_res=ei/(1-hii) #press resids
press_res
PRESS=sum(press_res^2) #PRESS score
PRESS

## Constant Variance test
shapiro.test(ei)
bptest(reg)
sigma <- sigma.hat(reg)
residual.plot(reg, ei, sigma)
plot(ei, main="Residual Plot", xlab="Observations",ylab="Residuals", col="palevioletred4",pch=20,
cex=1.0)
c.di=as.data.frame(c.di)
mu=mean(ei)

## Multinominal for position
head(data)
subsetMulti <- data[,-c(1:3)]
subsetMulti
mp2 <- vglm(Position ~ APG + BPG + X3PM + DRB + ORB + PF + IR + MPG ,
        data = subsetMulti, family = multinomial, method="vglm.fit")
summary(mp2)

## Factor Analysis
head(data)
data.names <- data[,c(2:3,24,25)]
data.num <- data[,c(4:7,9,10,12,13,15:18,20:23,26)]
data.st <- as.data.frame(round(scale(data.num),4))
head(data.st)
```

```
## Evaluating the “factorability” of our data with KMO
KMO(data.st)
data.st.d<-data.st[,-10]
KMO(data.st.d)

## Choosing number of factors to explain matrix.
parallel<- fa.parallel(data.st.d,fm="ML",fa = "fa")
parallel
fit <- fa(data.st.d, nfactors = 3, max.iter = 100, rotate = "varimax", fm = "ML")
fit$communality
fa.diagram(fit)

## Classification On The Basis Of Team ( Final Four Team or Not)
head(data)
dim(mydata)
dim(data)
head(data)
mydata<-data[,-c(1,2,3,25)]
head(mydata)
mydata$GP<-as.numeric(mydata$GP)
sapply(mydata, class)
dim(mydata)
head(mydata)

## FF<-mydata[,21]
data.num<-mydata[,-21]
data.st <- as.data.frame(round(scale(data.num),4))
mydata<-cbind(data.st,FF)
head(mydata)
validation_index <- createDataPartition(mydata$FF, p=0.80, list=FALSE)
```

```
## Select 20% of the data for validation
mydata.test <- mydata[-validation_index,]

# Use the remaining 80% of data to training and testing the models
mydata.train <- mydata[validation_index,]
sapply(mydata.train, class)
percentage <- prop.table(table(mydata.train$FF)) * 100
cbind(freq=table(mydata.train$FF), percentage=percentage)

##
df <- data.frame(
  group = c("Final Four", "Not Final Four Team"),
  value = c(25, 71)
)
bp<- ggplot(df, aes(x="", y=value, fill=group))+
  geom_bar(width = 1, stat = "identity")
bp
pie <- bp + coord_polar("y", start=0)
pie

##
x <- mydata.train[,c(1:21)]
y <- mydata.train[,22]
plot(y)
featurePlot(x=x, y=y, plot="box")

## Density plots for each attribute by class value
scales <- list(x=list(relation="free"), y=list(relation="free"))
featurePlot(x=x, y=y, plot="density", scales=scales)

## Run algorithms using 10-fold cross validation
control <- tr
```

```
control <- trainControl(method="cv", number=10)
metric <- "Accuracy"

## a) linear algorithms
set.seed(7)
fit.lda <- train(FF~., data=mydata.train, method="lda", metric=metric, trControl=control)

## b) nonlinear algorithms
## CART
set.seed(7)
fit.cart <- train(FF~., data=mydata.train, method="rpart", metric=metric, trControl=control)

## kNN
set.seed(7)
fit# .knn <- train(FF~., data=mydata.train, method="knn", metric=metric, trControl=control)

## c) advanced algorithms
## SVM
set.seed(7)
fit.svm <- train(FF~., data=mydata.train, method="svmRadial", metric=metric, trControl=control)

## Random Forest
set.seed(7)
fit.rf <- train(FF~., data=mydata.train, method="rf", metric=metric, trControl=control)
summary(results)
results <- resamples(list(lda=fit.lda, cart=fit.cart, knn=fit.knn, svm=fit.svm, rf=fit.rf))

## Compare accuracy of models
dotplot(results)
```

```
## Summarize Best Model
print(fit.knn)
fit.knn$modelInfo

## Estimate skill of KNN on the test dataset
predictions <- predict(fit.cart, mydata.test)
confusionMatrix(predictions, mydata.test$FF)

## Classification On The Basis Of Position
level(mydata$Position)
head(data)
dim(data)
mydata<-data[,-c(1,2,3,24)]
head(data)
head(mydata)
mydata$GP<-as.numeric(mydata$GP)
sapply(mydata, class)
Position<-mydata[,21]
data.num<-mydata[,-21]
data.st <- as.data.frame(round(scale(data.num),4))
mydata<-cbind(data.st,Position)
head(mydata)
validation_index <- createDataPartition(mydata$Position, p=0.80, list=FALSE)
## select 20% of the data for validation
mydata.test <- mydata[-validation_index,]

## use the remaining 80% of data to training and testing the models
mydata.train <- mydata[validation_index,]
percentage <- prop.table(table(mydata$Position)) * 100
cbind(freq=table(mydata$Position), percentage=percentage)
```

```
## pie
df <- data.frame(
  group = c("Center", "Forward", "Guard"),
  value = c(18, 49, 52)
)
bp<- ggplot(df, aes(x="", y=value, fill=group))+
  geom_bar(width = 1, stat = "identity")
bp
pie <- bp + coord_polar("y", start=0)
pie
summary(mydata)
head(mydata)

## plot(y)
x <- mydata.train[,c(1:21)]
y <- mydata.train[,22]
head(mydata.train)

## Density plots for each attribute by class value
scales <- list(x=list(relation="free"), y=list(relation="free"))
featurePlot(x=x, y=y, plot="density", scales=scales)

## with variables
control <- trainControl(method="cv", number=10)
metric <- "Accuracy"
set.seed(7)
fit.lda <- train(Position~., data=mydata.train, method="lda", metric=metric, trControl=control)

# b) nonlinear algorithms
# CART
set.seed(7)
fit.cart <- train(Position~., data=mydata.train, method="rpart", metric=metric, trControl=control)
```

```
## kNN
set.seed(7)
fit.knn <- train(Position~., data=mydata.train, method="knn", metric=metric, trControl=control)

# c) advanced algorithms
## SVM
set.seed(7)
fit.svm <- train(Position~., data=mydata.train, method="svmRadial", metric=metric, trControl=control)

## Random Forest
set.seed(7)
fit.rf <- train(Position~., data=mydata.train, method="rf", metric=metric, trControl=control)
results <- resamples(list(lda=fit.lda, cart=fit.cart, knn=fit.knn, svm=fit.svm, rf=fit.rf))
summary(results)

## Compare accuracy of models
dotplot(results)

## Summarize Best Model
print(fit.rf)
fit.rf$modelInfo
## Estimate skill of RF on the test dataset
predictions <- predict(fit.rf, mydata.test)
confusionMatrix(predictions, mydata.test$Position)

## K-means
data.names <- data[,c(2:3,24,25)]
data.num <- data[,c(4:7,9,10,12,13,15:18,20:23,26)]
data.num2 <- data[,-c(2:3,24:25)]
head(data.num)
head(data.names)
```

```
dim(data.names);dim(data.num)
data.s <- as.data.frame(round(scale(data.num),4))
head(data.st)
data.st <- data.frame(data.names, data.s)
head(data.s)
data.num2 <- data.frame(round(scale(data.num2),4))
datadata <- data.frame(data.names, data.num2)
names(datadata)
data.cluster3 <- kmeans(datadata[,-c(1:5)],3)#hepsi
data.cluster3
table(data.cluster3$cluster,datadata$Position)
a3 <- ggplot(datadata, aes(PPG,MPG, colour = as.factor(data.cluster3$cluster), label = Position)) +
  geom_point() + geom_text(aes(label= Position),hjust=0, vjust=0)
print(a3 + scale_colour_manual(values = c("Blue", "Red", "dark green")))

## Multivariate Regression
names(data.st)
set.seed(7)
testNumbers <- sample(1:119,size = 24, replace = FALSE)
testSet <- data.st[testNumbers,-c(1:3)]
trainSet <- data.st[-testNumbers,-c(1:3)]
head(trainSet)
modelMPG <- lm(MPG ~ . ,data=trainSet)
summary(modelMPG)
ols_step_forward_p(modelMPG,details = TRUE)
model2MPG <- lm(MPG ~ FGA + SPG + PF  + DRB + APG + X3PM + ORB + Position + X3PA , data
= trainSet)
summary(model2MPG)
model3MPG <- lm(MPG ~ FGA + SPG + PF  + DRB + APG + X3PM + ORB + Position, data = trainSet)
summary(model3MPG)
```

```
## Adequecy check
resid <- model3MPG$residuals
qqnorm(resid);qqline(resid)
shapiro.test(resid)
cd <- cooks.distance(model3MPG)
plot(cd, pch="*", cex=2, main="Influential Obs by Cooks distance")  # plot cook's distance
abline(h = 4*mean(cd, na.rm=T), col="red")  # add cutoff line
text(x=1:length(cd)+1, y=cd, labels=ifelse(cd>4*mean(cd, na.rm=T),names(cd),""), col="red")

## Constant var
plot(model3MPG,1)
dataNum <- data[,-c(1:3,24,25)]
head(dataNumS)
dataNumS <- data.frame(round(scale(dataNum),4))
dataName <- data[,c(1:3,24,25)]
pcaNew <- princomp(dataNumS)
names(pcaNew)
print(pcaNew)
summary(pcaNew, loadings = TRUE)
biplot(pcaNew)
set.seed(7)
testNumbers <- sample(1:119,size = 24, replace = FALSE)
testSet <- dataNumS[testNumbers,]
trainSet <- dataNumS[-testNumbers,]
set.seed(7)
pcr_model <- pcr(MPG ~ ., data = trainSet, scale = TRUE,ncomp = 5, validation = "CV")
pcr_pred <- predict(pcr_model, testSet, ncomp = 3)
mean((pcr_pred - y_test)^2)
summary(pcr_model)
validationplot(pcr_model, val.type = "MSE")
predplot(pcr_model)
```

```
res_pcr <- pcr_model$residuals
shapiro.test(res_pcr)
qqnorm(res_pcr);qqline(res_pcr)
coefs <- pcr_model$coefficients ########ORÇUN BAKACAK BURAYA!!!!
pcr_model$validation
scatter3D(PC1, PC2, PC3, pch = 16,phi=0, bty = "g",ticktype = "detailed")
p <- plot_ly(PC, x = ~PC1, y = ~PC2, z = ~PC3,
          marker = list(color = ~PC4, colorscale = c('#FFE1A1', '#683531'), showscale = TRUE)) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'PC1'),
                      yaxis = list(title = 'PC2'),
                      zaxis = list(title = 'PC3')),
         annotations = list(
           x = 1.13,
           y = 1.05,
           text = 'Miles/(US) gallon',
           xref = 'paper',
           yref = 'paper',
           showarrow = FALSE
         ))
## Create a shareable link to your chart
chart_link = api_create(p, filename="osman")
chart_link


## Cluster Analysis
head(data)
FF <- data[which(data$FF=="F4"),]
data.names.ff <- FF[,c(2:3)]
data.num.ff <- FF[,c(4:7,9,10,12,13,15:18,20:23)]
data.st.ff <- as.data.frame(round(scale(data.num.ff),4))
d <- dist(data.st.ff, method = "euclidean") # distance matrix
fit <- hclust(d, method="ward.D")
```

```
plot(fit,labels = data.names.ff[,1], main = "Cluster Dendogram for Final Four" , xlab = "") # display dendogram

groups <- cutree(fit, k=5) # cut tree into 5 clusters


## Draw dendogram with red borders around the 5 clusters

rect.hclust(fit, k=5, border="red")
```