ProgettoLampedusa

2023-12-07

Progetto numero 5

NBA moderna (1976-2011): VARIABILE DIPENDENTE: numero di vittorie in stagione COVARIATE: tutte le altre (o uno specifico insieme di queste, in base all'obiettivo di analisi) > Considerare solo le squadre che hanno giocato 82 partite (dataset$games==82)

INIZIALIZZAZIONE DATI E GRAFICI DATI

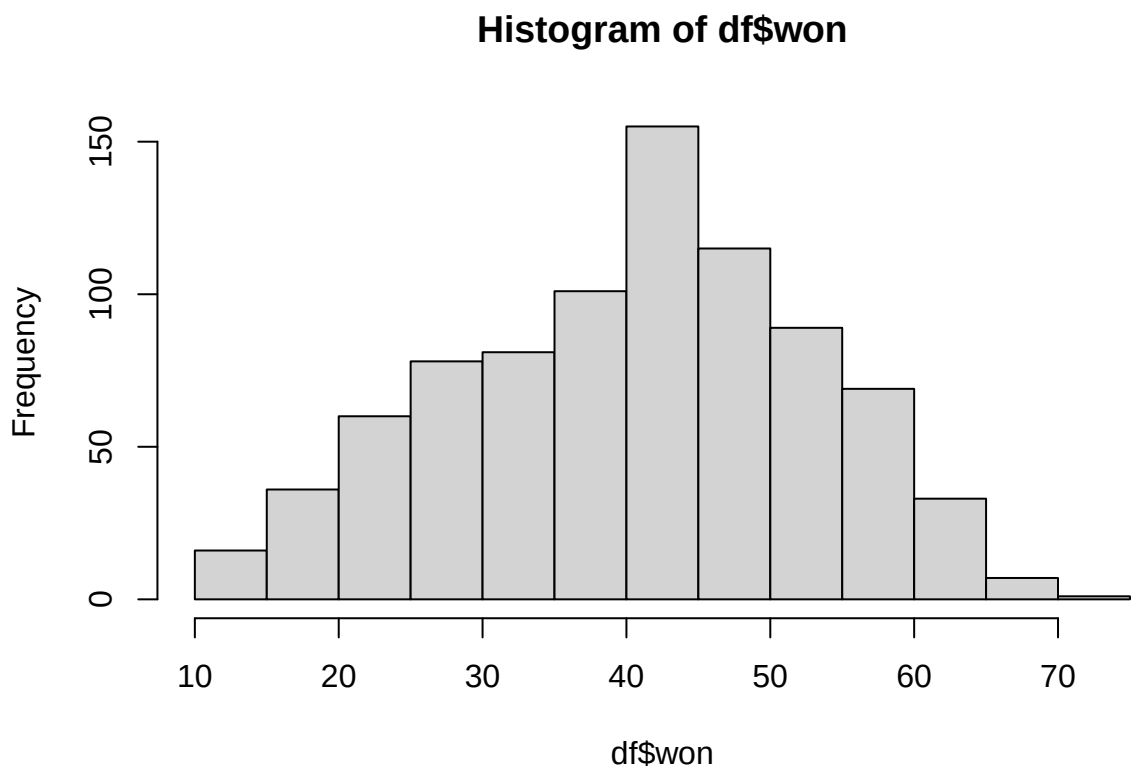
```
dataset <- read.delim("basketball_teams.txt") # andiamo a leggere il database fornito
FIRST <- 1976 # primo anno del range da considerare per lo studio
LAST <- 2011 # ultimo anno del range da considerare per lo studio

df <- dataset [dataset$lgID=="NBA" & dataset$year >= FIRST & dataset$year <= LAST & dataset$games==82,]

dataset$lgID <- as.factor(dataset$lgID) # perchè mi permettono di poter generare variabili dummy
summary(df)
```

```
##       year          lgID               tmID             franchID
##  Min.   :1976   Length:841         Length:841         Length:841
##  1st Qu.:1985   Class :character   Class :character   Class :character
##  Median :1993   Mode  :character   Mode  :character   Mode  :character
##  Mean   :1993
##  3rd Qu.:2001
##  Max.   :2008
##     confID             divID                rank          confRank
##  Length:841         Length:841         Min.   :0.000   Min.   : 1.000
##  Class :character   Class :character   1st Qu.:2.000   1st Qu.: 4.000
##  Mode  :character   Mode  :character   Median :3.000   Median : 7.000
##                                        Mean   :3.565   Mean   : 7.164
##                                        3rd Qu.:5.000   3rd Qu.:10.000
##                                        Max.   :8.000   Max.   :15.000
##    playoff              name               o_fgm          o_fga
##  Length:841         Length:841         Min.   :2565   Min.   :5972
##  Class :character   Class :character   1st Qu.:2981   1st Qu.:6592
##  Mode  :character   Mode  :character   Median :3220   Median :6903
##                                        Mean   :3239   Mean   :6941
##                                        3rd Qu.:3489   3rd Qu.:7253
##                                        Max.   :3980   Max.   :8868
##      o_ftm          o_fta          o_3pm           o_3pa            o_oreb
##  Min.   :1189   Min.   :1475   Min.   :  0.0   Min.   :   0.0   Min.   : 720
##  1st Qu.:1523   1st Qu.:2039   1st Qu.: 78.0   1st Qu.: 293.0   1st Qu.: 974
```

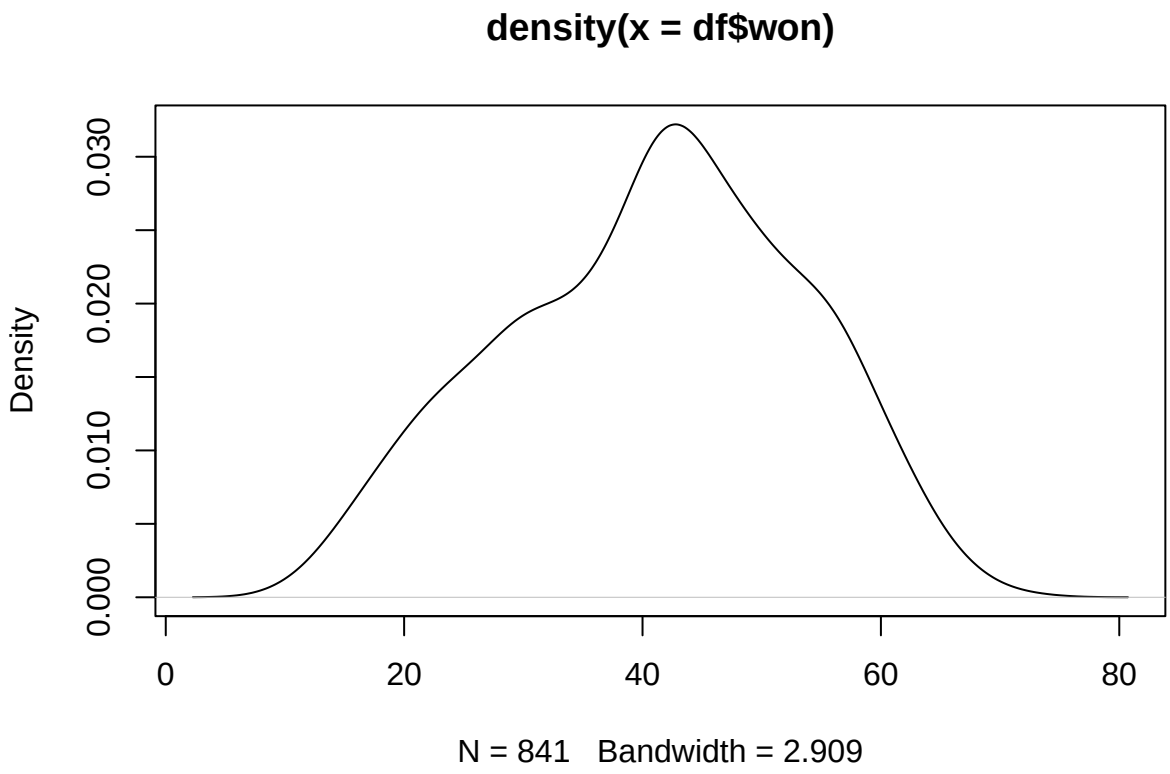
```
##  Median :1649   Median :2201   Median :283.0   Median : 814.0   Median :1083
##  Mean   :1666   Mean   :2212   Mean   :279.1   Mean   : 804.6   Mean   :1086
##  3rd Qu.:1797   3rd Qu.:2371   3rd Qu.:445.0   3rd Qu.:1267.0   3rd Qu.:1191
##  Max.   :2388   Max.   :3051   Max.   :837.0   Max.   :2283.0   Max.   :1520
##      o_dreb         o_reb          o_asts          o_pf          o_stl
##  Min.   :2044   Min.   :2922   Min.   :1422   Min.   :1476   Min.   : 455.0
##  1st Qu.:2348   1st Qu.:3381   1st Qu.:1760   1st Qu.:1777   1st Qu.: 613.0
##  Median :2433   Median :3506   Median :1934   Median :1900   Median : 671.0
##  Mean   :2434   Mean   :3520   Mean   :1934   Mean   :1909   Mean   : 681.3
##  3rd Qu.:2527   3rd Qu.:3647   3rd Qu.:2094   3rd Qu.:2033   3rd Qu.: 746.0
##  Max.   :2966   Max.   :4216   Max.   :2575   Max.   :2470   Max.   :1059.0
##       o_to           o_blk           o_pts           d_fgm          d_fga
##  Min.   : 910   Min.   :204.0   Min.   : 6901   Min.   :2488   Min.   :5638
##  1st Qu.:1207   1st Qu.:360.0   1st Qu.: 7958   1st Qu.:2978   1st Qu.:6593
##  Median :1311   Median :411.0   Median : 8404   Median :3243   Median :6911
##  Mean   :1338   Mean   :421.8   Mean   : 8423   Mean   :3239   Mean   :6941
##  3rd Qu.:1443   3rd Qu.:473.0   3rd Qu.: 8879   3rd Qu.:3493   3rd Qu.:7268
##  Max.   :2011   Max.   :716.0   Max.   :10371   Max.   :4265   Max.   :8142
##      d_ftm          d_fta           d_3pm           d_3pa          d_oreb
##  Min.   :1217   Min.   :  0.0   Min.   :   0.0   Min.   :1579   Min.   : 745
##  1st Qu.:1514   1st Qu.: 82.0   1st Qu.: 282.0   1st Qu.:2033   1st Qu.: 986
##  Median :1648   Median :280.0   Median : 836.0   Median :2203   Median :1095
##  Mean   :1666   Mean   :279.1   Mean   : 804.6   Mean   :2212   Mean   :1086
##  3rd Qu.:1808   3rd Qu.:446.0   3rd Qu.:1251.0   3rd Qu.:2395   3rd Qu.:1177
##  Max.   :2377   Max.   :683.0   Max.   :1768.0   Max.   :3071   Max.   :1495
##      d_dreb         d_reb          d_asts          d_pf          d_stl
##  Min.   :2012   Min.   :2976   Min.   :1336   Min.   :1434   Min.   :461.0
##  1st Qu.:2326   1st Qu.:3378   1st Qu.:1778   1st Qu.:1788   1st Qu.:623.0
##  Median :2431   Median :3497   Median :1939   Median :1900   Median :677.0
##  Mean   :2434   Mean   :3520   Mean   :1934   Mean   :1909   Mean   :681.3
##  3rd Qu.:2529   3rd Qu.:3653   3rd Qu.:2092   3rd Qu.:2020   3rd Qu.:734.0
##  Max.   :3067   Max.   :4309   Max.   :2537   Max.   :2453   Max.   :955.0
##       d_to           d_blk           d_pts        o_tmRebound  d_tmRebound
##  Min.   : 949   Min.   :264.0   Min.   : 6909   Min.   :0    Min.   :0
##  1st Qu.:1208   1st Qu.:380.0   1st Qu.: 7968   1st Qu.:0    1st Qu.:0
##  Median :1304   Median :419.0   Median : 8453   Median :0    Median :0
##  Mean   :1338   Mean   :421.8   Mean   : 8423   Mean   :0    Mean   :0
##  3rd Qu.:1444   3rd Qu.:460.0   3rd Qu.: 8841   3rd Qu.:0    3rd Qu.:0
##  Max.   :1980   Max.   :654.0   Max.   :10723   Max.   :0    Max.   :0
##     homeWon         homeLost        awayWon         awayLost        neutWon
##  Min.   : 6.00   Min.   : 1.00   Min.   : 1.00   Min.   : 8.00   Min.   :0
##  1st Qu.:21.00   1st Qu.:10.00   1st Qu.:10.00   1st Qu.:21.00   1st Qu.:0
##  Median :26.00   Median :15.00   Median :15.00   Median :26.00   Median :0
##  Mean   :25.63   Mean   :15.37   Mean   :15.37   Mean   :25.63   Mean   :0
##  3rd Qu.:31.00   3rd Qu.:20.00   3rd Qu.:20.00   3rd Qu.:31.00   3rd Qu.:0
##  Max.   :40.00   Max.   :35.00   Max.   :33.00   Max.   :40.00   Max.   :0
##     neutLoss    confWon         confLoss         divWon         divLoss
##  Min.   :0   Min.   : 5.00   Min.   : 7.00   Min.   : 1.00   Min.   : 1.00
##  1st Qu.:0   1st Qu.:20.00   1st Qu.:20.00   1st Qu.: 8.00   1st Qu.: 9.00
##  Median :0   Median :27.00   Median :26.00   Median :12.00   Median :12.00
##  Mean   :0   Mean   :26.91   Mean   :26.91   Mean   :12.27   Mean   :12.27
##  3rd Qu.:0   3rd Qu.:34.00   3rd Qu.:33.00   3rd Qu.:16.00   3rd Qu.:16.00
##  Max.   :0   Max.   :48.00   Max.   :52.00   Max.   :25.00   Max.   :27.00
##       pace             won          lost        games         min
```

```
##  Min.   :  0.00   Min.   :11   Min.   :10   Min.   :82   Min.   :19680
##  1st Qu.:  0.00   1st Qu.:31   1st Qu.:32   1st Qu.:82   1st Qu.:19780
##  Median :  0.00   Median :42   Median :40   Median :82   Median :19805
##  Mean   :  6.71   Mean   :41   Mean   :41   Mean   :82   Mean   :19817
##  3rd Qu.:  0.00   3rd Qu.:50   3rd Qu.:51   3rd Qu.:82   3rd Qu.:19855
##  Max.   :102.00   Max.   :72   Max.   :71   Max.   :82   Max.   :20080
##     arena            attendance       bbtmID
##  Length:841         Min.   :    0   Length:841
##  Class :character   1st Qu.:32767   Class :character
##  Mode  :character   Median :32767   Mode  :character
##                     Mean   :32728
##                     3rd Qu.:32767
##                     Max.   :32767
```

```
hist(df$won)
```

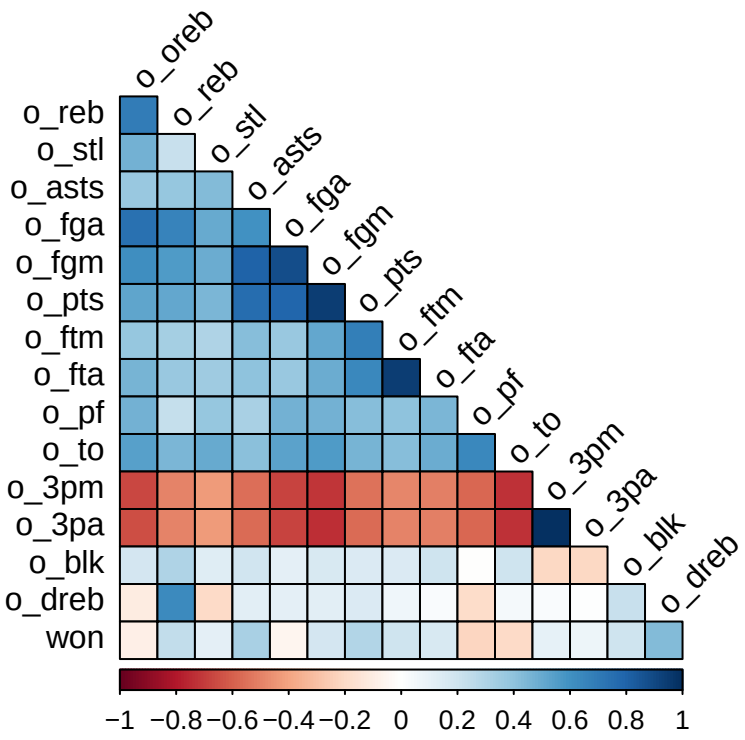



```
plot(density(df$won))
```

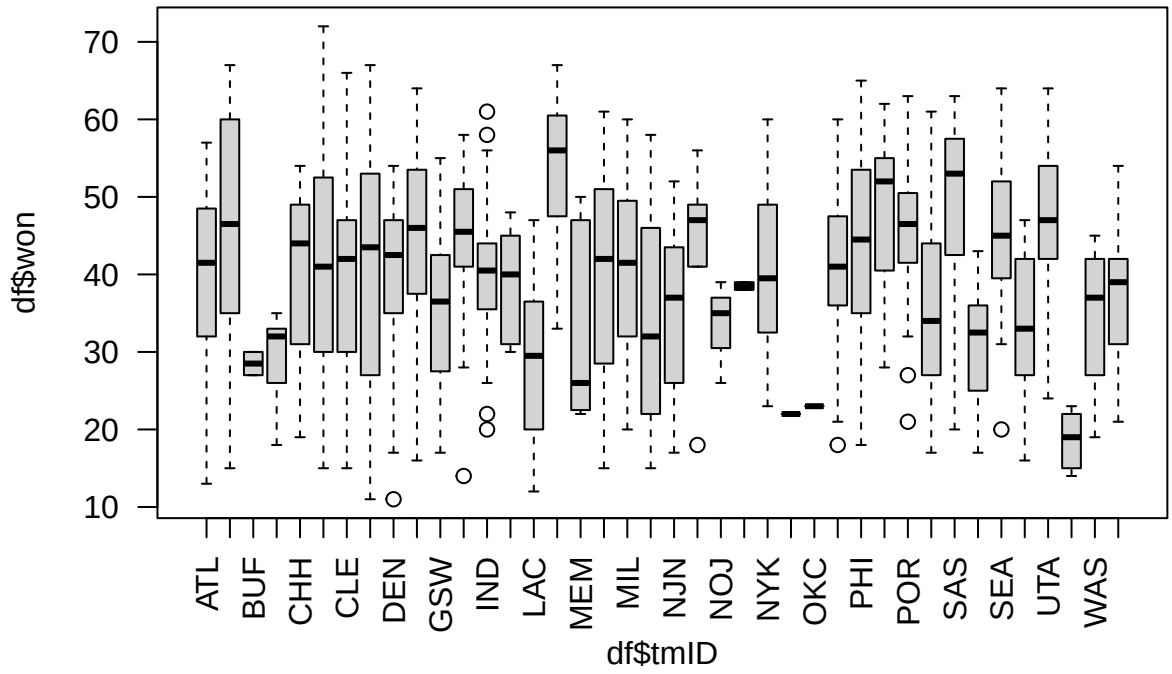



```
M <- cor(as.matrix(df[, c(11:25, 54)])) # correlation matrix
corrplot(M, method="color", outline = TRUE,type="lower",order = "hclust",
        tl.col="black", tl.srt=45, diag=FALSE,tl.cex = 1,mar=c(0,0,3,0),
        title="Correlation Matrix between Predictor and Outcome variables")
```

Correlation Matrix between Predictor and Outcome variables




```
boxplot(df$won ~ df$tmID, las=2)
```




```
df$reb <- df$o_reb + df$d_reb
```

TESTS DI VERIFICA

TEST ANDERSON-DARLING

```
ad.test(df$reb)
```

```
##
##  Anderson-Darling normality test
##
## data:  df$reb
## A = 3.6997, p-value = 3.1e-09
```

Con un livello di significatività (α) di 0.01 e un p-value molto piccolo (3.1e-09) ottenuto dal test di normalità di Anderson-Darling per i dati della variabile df$reb, puoi concludere che hai sufficiente evidenza statistica per respingere lipotesi nulla che i dati seguono una distribuzione normale.Con il tuo livello di significatività del 0.01 e il p-value molto piccolo (3.1e-09), il p-value è inferiore al livello di significatività, quindi respingeresti lipotesi nulla. Questo suggerisce che i dati nella variabile df$reb non seguono una distribuzione normale al livello di significatività del 0.01. In termini più pratici, hai abbastanza evidenza statistica per concludere che la variabile df$reb non segue una distribuzione normale basandoti sui risultati del test di Anderson-Darling.

TEST KOLMOGOROV SMIRNOV

```
ks.test(df$reb, "pnorm")
```

```
## Warning in ks.test.default(df$reb, "pnorm"): ties should not be present for the
## Kolmogorov-Smirnov test
```

```
##
##  Asymptotic one-sample Kolmogorov-Smirnov test
##
## data:  df$reb
## D = 1, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

Il risultato che hai ottenuto riguarda il test di Kolmogorov-Smirnov a campione singolo sui dati contenuti nella variabile df$reb. Il test KS confronta la distribuzione empirica dei tuoi dati con una distribuzione teorica (spesso una distribuzione uniforme). In breve, il risultato suggerisce che i tuoi dati non seguono la distribuzione teorica presunta, e cè un elevata probabilità che la differenza osservata sia statisticamente significativa.

TEST SHAPIRO WILK

```
sf.test(df$reb)
```

```
##
##  Shapiro-Francia normality test
##
## data:  df$reb
## W = 0.98016, p-value = 1.758e-08
```

In sintesi, il risultato del test di Shapiro-Francia indica che i tuoi dati nella variabile df$reb non seguono una distribuzione normale. Questo è supportato dal valore basso del p-value, il quale suggerisce che la differenza tra la distribuzione dei tuoi dati e una distribuzione normale è statisticamente significativa.

INIZIALIZZAZIONE MODELLO DI REGRESSIONE LINEARE

L'IMPORTANZA DEI RIMBALZI

Formula1 = $\frac{\text{Rimbalzi offensivi in attacco}}{\text{Tiri sbagliati su azione}}$ Rappresenta la capacità della squadra di ripossesso della palla dopo un tiro che non va a canestro e colpisce il tabellone.

Formula2 = $\frac{\text{Rimbalzi difensivi in difesa presi}}{\text{Tiri sbagliati su azione degli avversari}}$ Rappresenta la capacità della squadra di impossessarsi della palla dopo un tiro sbagliato della squadra avversaria che colpisce il tabellone, che troviamo un buon stimatore della capacità di contropiede della squadra.

Formula3 = $\frac{\text{Palle riprese in attacco}+1.5\times\text{Palle riprese in difesa}}{\text{Palle perse in attacco}+2\times\text{Rimbalzi subiti in difesa}}$ Rappresenta il rapporto tra le palle riprese nei rimbalzi (sia offensivi che difensivi) rispetto alle palle perse nei rimbalzi (sia offensivi che difensivi). I coefficienti sono stati scelti in base a ciò che riteniamo più importante in una partita, ossia la difesa del proprio canestro.

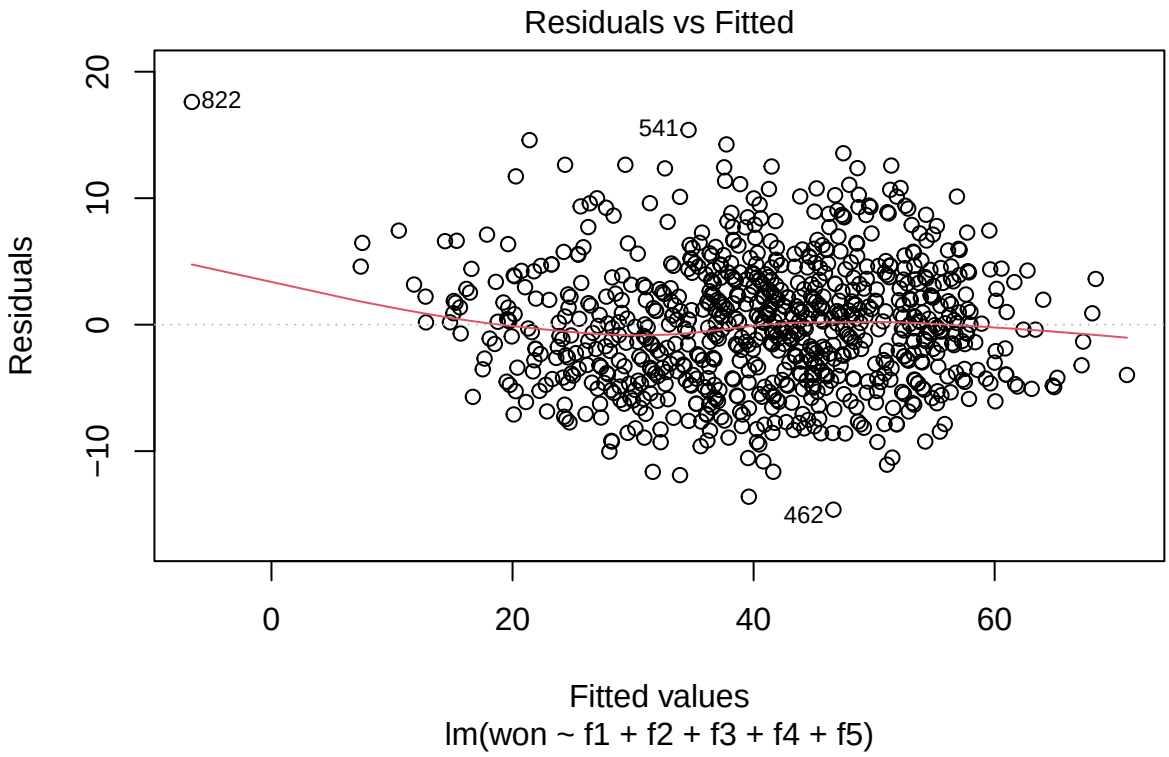
Formula4 = (Palle riprese in attacco - Palle perse in attacco)+1.5∗(Palle riprese in difesa - Palle perse in difesa) Cresce all'aumentare dei rimbalzi ottenuti e diminuisce all'aumentare dei rimbalzi subiti, considerando anche un coefficiente che da particolare importanza alla difesa.

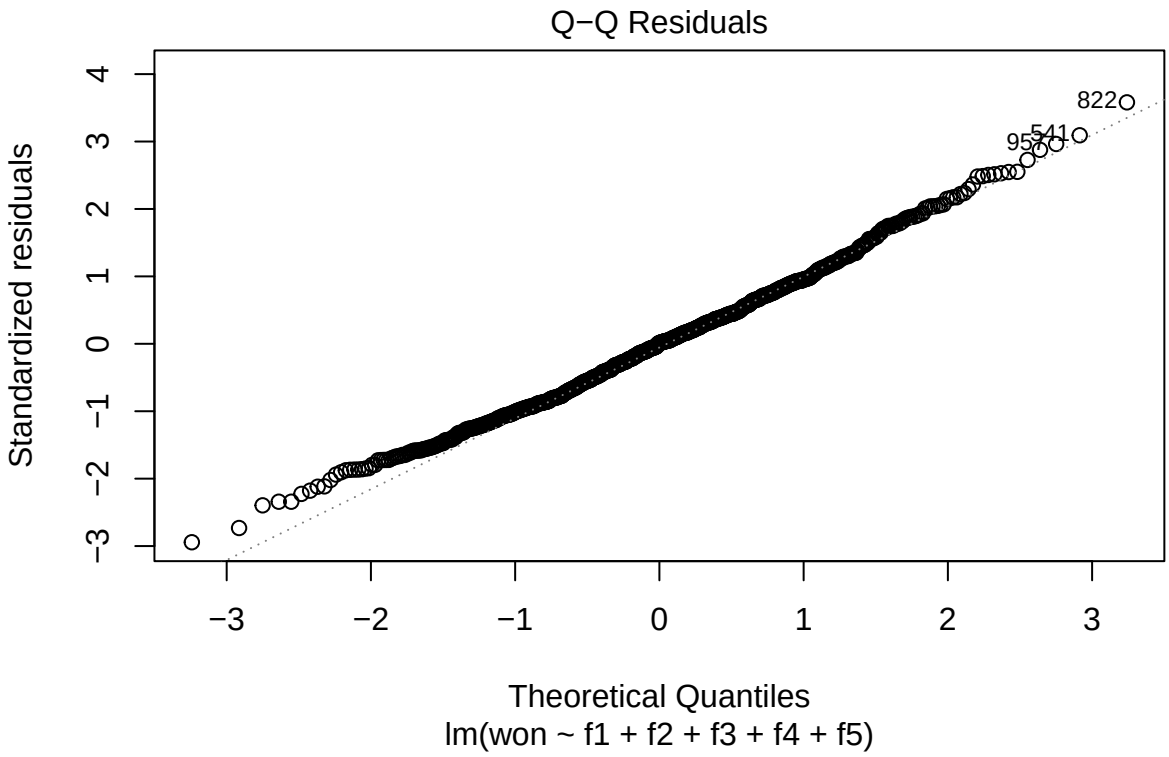
Formula5 = $\frac{(\frac{\text{Rimbalzi subiti in difesa}}{\text{Palle perse in difesa}})}{(\frac{\text{Rimbalzi subiti in attacco}}{\text{Palle perse in attacco}})}$ Mostra quanto siano influenti i rimbalzi nel rapporto tra le palle perse dalla squadra e le palle perse dagli avversari.

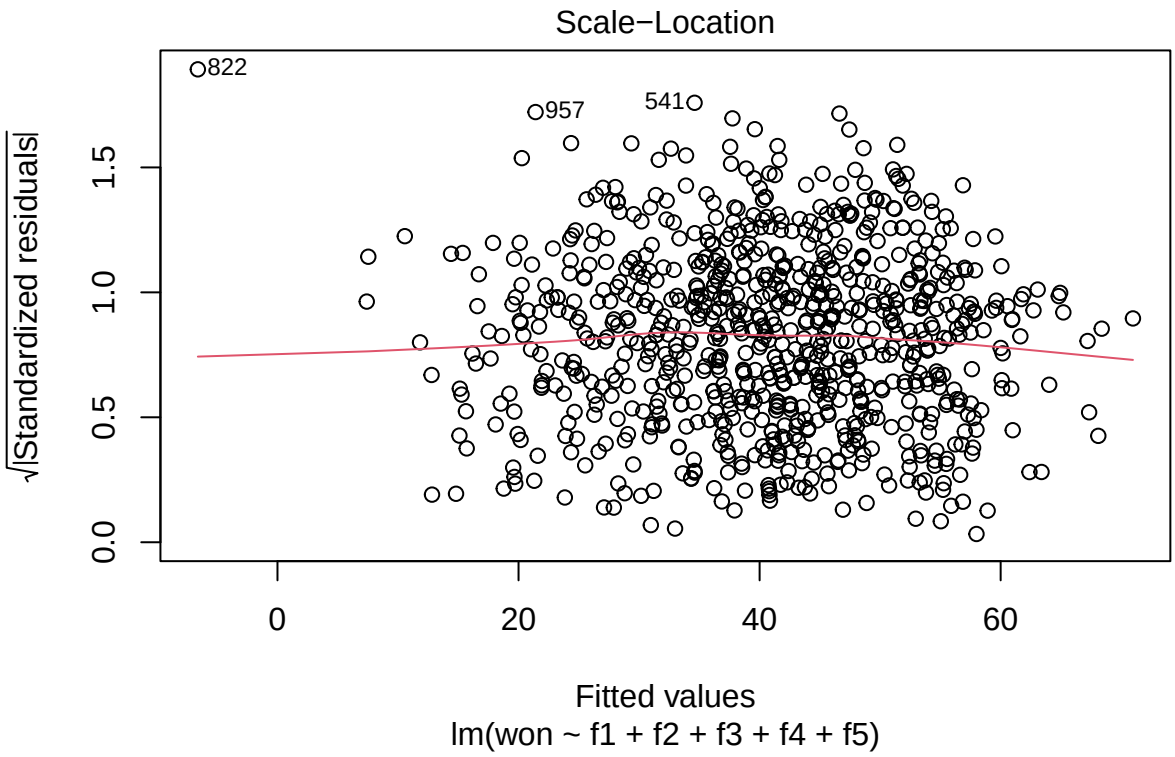
```
# o_oreb = Rimbalzi ottenuti in attacco
# o_dreb = Rimbalzi subiti in attacco
# o_reb  = totale rimbalzi in attacco
# d_oreb = Rimbalzi subiti in difesa
# d_dreb = Rimbalzi ottenuti in difesa
# d_reb  = totale rimbalzi in difesa

df$f1 <- (df$o_oreb)/(df$o_fga-df$o_fgm)
df$f2 <- (df$d_dreb)/(df$d_fga-df$d_fgm)
df$f3 <- (df$o_oreb + 1.5 * df$d_dreb)/(df$o_dreb + 2 * df$d_oreb)
df$f4 <- (df$o_oreb - df$o_dreb) + 1.5 * (df$d_dreb - df$d_oreb)
df$f5 <- (df$d_oreb / df$d_to) / (df$o_dreb / df$o_to)

linMod <- lm(won ~ f1 + f2 + f3 + f4 + f5, data = df)
summary (linMod)
```

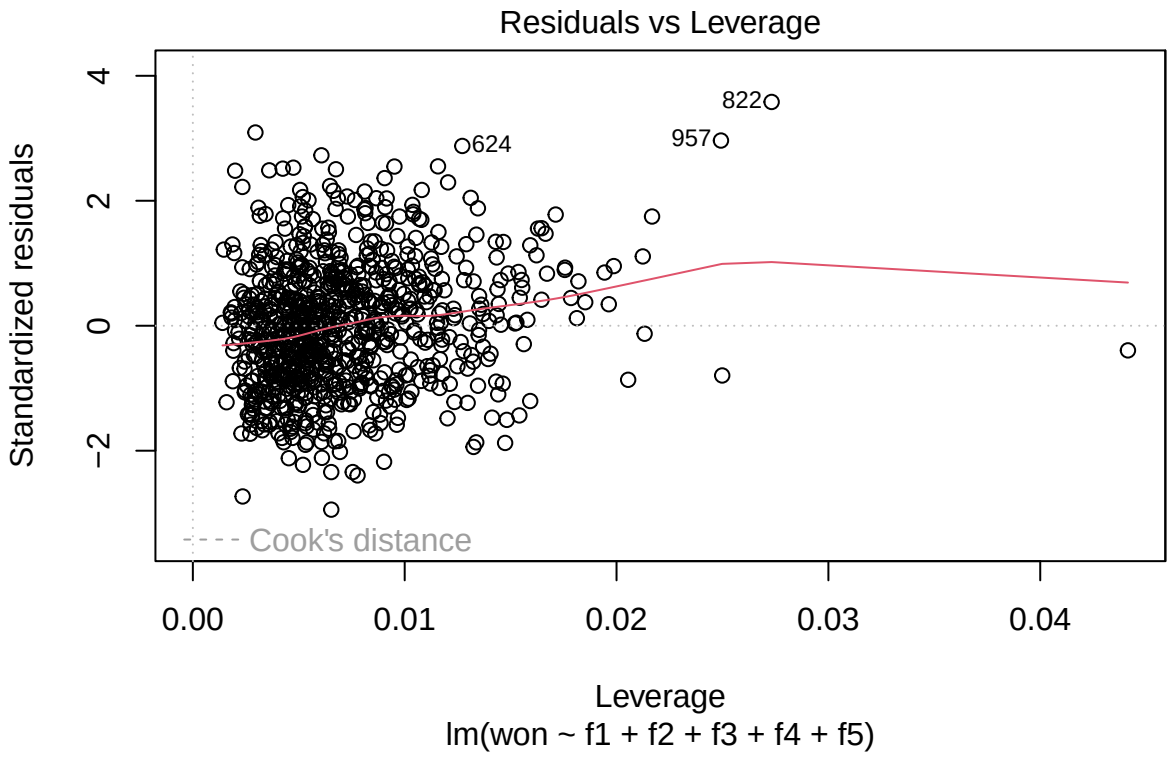
```
##
## Call:
## lm(formula = won ~ f1 + f2 + f3 + f4 + f5, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.624  -3.796   0.044   3.243  17.608
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  395.45317   18.36490  21.533  < 2e-16 ***
## f1            83.35460   10.45130   7.976 4.99e-15 ***
## f2           -59.76252   13.37858  -4.467 9.02e-06 ***
## f3          -273.63124   15.89419 -17.216  < 2e-16 ***
## f4             0.03391    0.00415   8.171 1.13e-15 ***
## f5          -178.48989    3.87823 -46.024  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.985 on 835 degrees of freedom
## Multiple R-squared:  0.8401, Adjusted R-squared:  0.8392
## F-statistic: 877.6 on 5 and 835 DF,  p-value: < 2.2e-16
```

```
plot(linMod)
```


Residuals vs Fitted
822
541
462
Residuals
20
10
0
−10
0
20
40
60
Fitted values
lm(won ~ f1 + f2 + f3 + f4 + f5)


Q−Q Residuals
Standardized residuals
−3
−2
−1
0
1
2
3
4
822
541
95
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(won ~ f1 + f2 + f3 + f4 + f5)


Scale−Location
822
957
541
√|Standardized residuals|
0.0
0.5
1.0
1.5
0
20
40
60
Fitted values
lm(won ~ f1 + f2 + f3 + f4 + f5)



INIZIALIZZAZIONE MODELLO DI REGRESSIONE LINEARE NORMALIZZATO

```r
# In un chunk diverso per minimizzare cpu-time

# Normalizziamo le covariate
df$f1_z <- scale(df$f1)
df$f2_z <- scale(df$f2)
df$f3_z <- scale(df$f3)
df$f4_z <- scale(df$f4)
df$f5_z <- scale(df$f5)

linModNormalized <- lm(won ~ f1_z + f2_z + f3_z + f4_z + f5_z, data = df)
```

TEST SUL MODELLO DI REGRESSIONE LINEARE

TEST BREUSCH-PAGAN (Test di omoschedasticità)

```r
# TEST SUL MODELLO DI REGRESSIONE LINEARE

#1 Summary
summary (linModNormalized)
```

```
##
## Call:
## lm(formula = won ~ f1_z + f2_z + f3_z + f4_z + f5_z, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.624  -3.796   0.044   3.243  17.608
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  41.0000     0.1719 238.505  < 2e-16 ***
## f1_z          2.6573     0.3332   7.976 4.99e-15 ***
## f2_z         -2.8564     0.6394  -4.467 9.02e-06 ***
## f3_z        -17.7693     1.0322 -17.216  < 2e-16 ***
## f4_z          9.1442     1.1191   8.171 1.13e-15 ***
## f5_z        -12.6019     0.2738 -46.024  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.985 on 835 degrees of freedom
## Multiple R-squared:  0.8401, Adjusted R-squared:  0.8392
## F-statistic: 877.6 on 5 and 835 DF,  p-value: < 2.2e-16
```

```
#2 R-quadrato e R-quadrato Adattato
summary_linModNormalized <- summary(linModNormalized)
r_squared <- summary_linModNormalized$r.squared
cat("R-squared:", r_squared, "\n")
```

```
## R-squared: 0.8401246
```

```
n <- length(df$o_oreb)
k <- length(linModNormalized$coefficients) - 1
adjusted_r_squared <- 1 - ((1 - r_squared) * (n - 1) / (n - k - 1))
cat("Adjusted R-squared:", adjusted_r_squared, "\n")
```

```
## Adjusted R-squared: 0.8391672
```

```
#2 test Shapiro per valutare la normalita' dei residui
shapiro.test(residuals(linModNormalized))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(linModNormalized)
## W = 0.99508, p-value = 0.008216
```

```
#3 test di omoschedasticita'
bptest(linModNormalized)
```

```
##
##  studentized Breusch-Pagan test
```

```
##
## data:  linModNormalized
## BP = 9.6069, df = 5, p-value = 0.08717
```

```
#4 test di multicollinearita'
car::vif(linModNormalized)
```

```
##      f1_z      f2_z      f3_z      f4_z      f5_z
##  3.751989 13.820282 36.007839 42.333607  2.534096
```

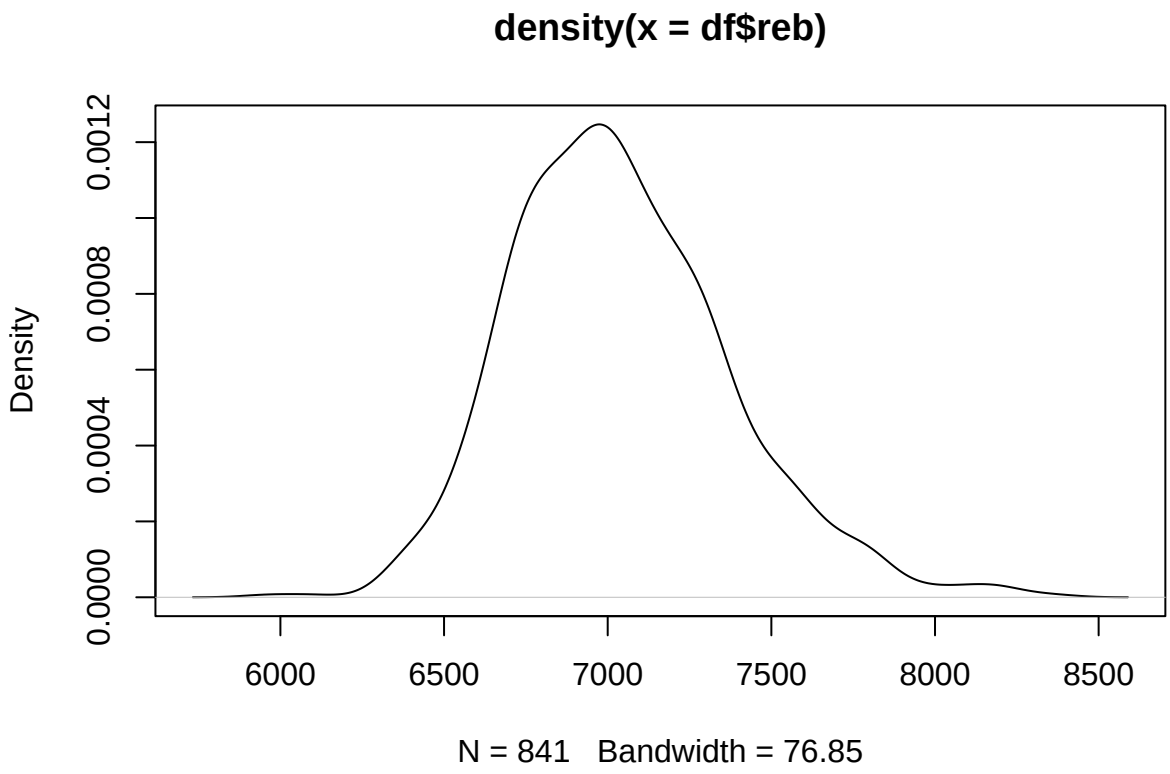
```
# Test di homoschedasticita' (Breusch-Pagan test) --> risultato suggerisce omoschedasiticita'

lmtest::bptest(linModNormalized)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  linModNormalized
## BP = 9.6069, df = 5, p-value = 0.08717
```

```
# Divisione in Test e Train per evitare che il modello fitti troppo bene sui nostri dati
sample <- sample(c(TRUE, FALSE), nrow(df), replace=TRUE, prob=c(0.7, 0.3))
train  <- df[sample, ]
test   <- df[!sample, ]

# m <- lm(won ~ o_canestriSuTotali_z + d_canestriSuTotali_z + d_stoppateSuTiri_z + o_rimbTiriSbagliati_
# summary(m)
```

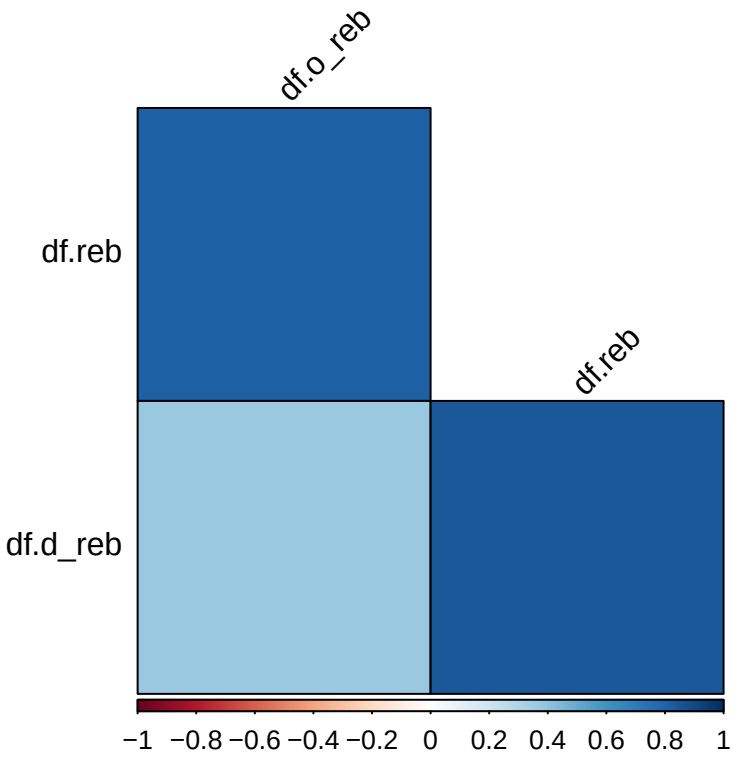
Il risultato suggerisce omoschedasiticita'

```
values <- aggregate(cbind(o_oreb, o_dreb, d_oreb, d_dreb, o_reb, d_reb) ~ tmID, data = df, FUN = sum)

# temp <- hist (temp, col = 'steelblue', main = 'caccaculo', xlab = 'balls')

plot(density(df$reb))
```

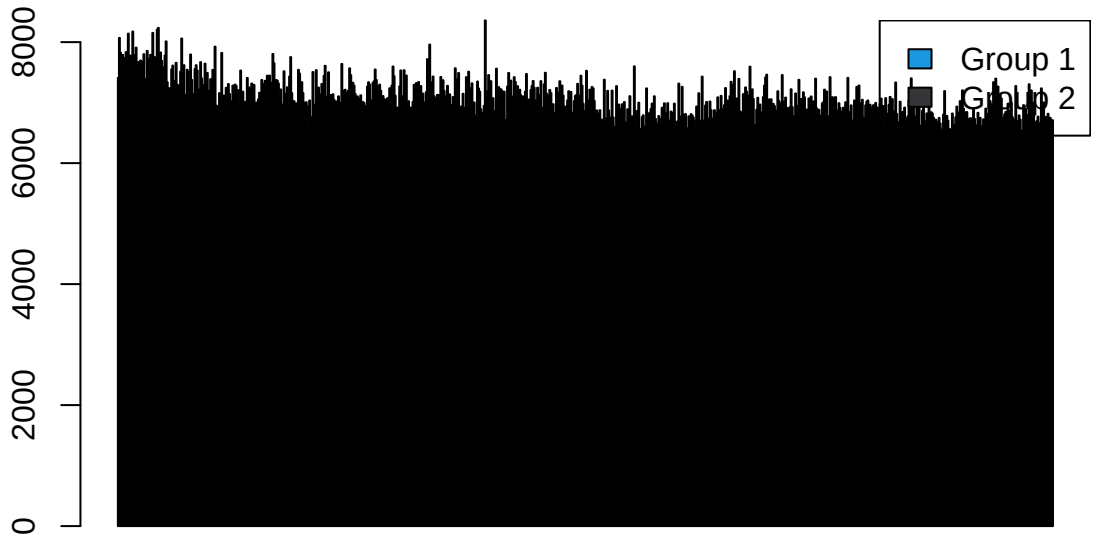



```
df_reb <- data.frame (df$reb, df$o_reb, df$d_reb)
M <- cor(df_reb) # correlation matrix
corrplot(M, method="color", outline = TRUE,type="lower",order = "hclust",
         tl.col="black", tl.srt=45, diag=FALSE,tl.cex = 1,mar=c(0,0,3,0),
         title="Correlation Matrix between Predictor and Outcome variables")
```

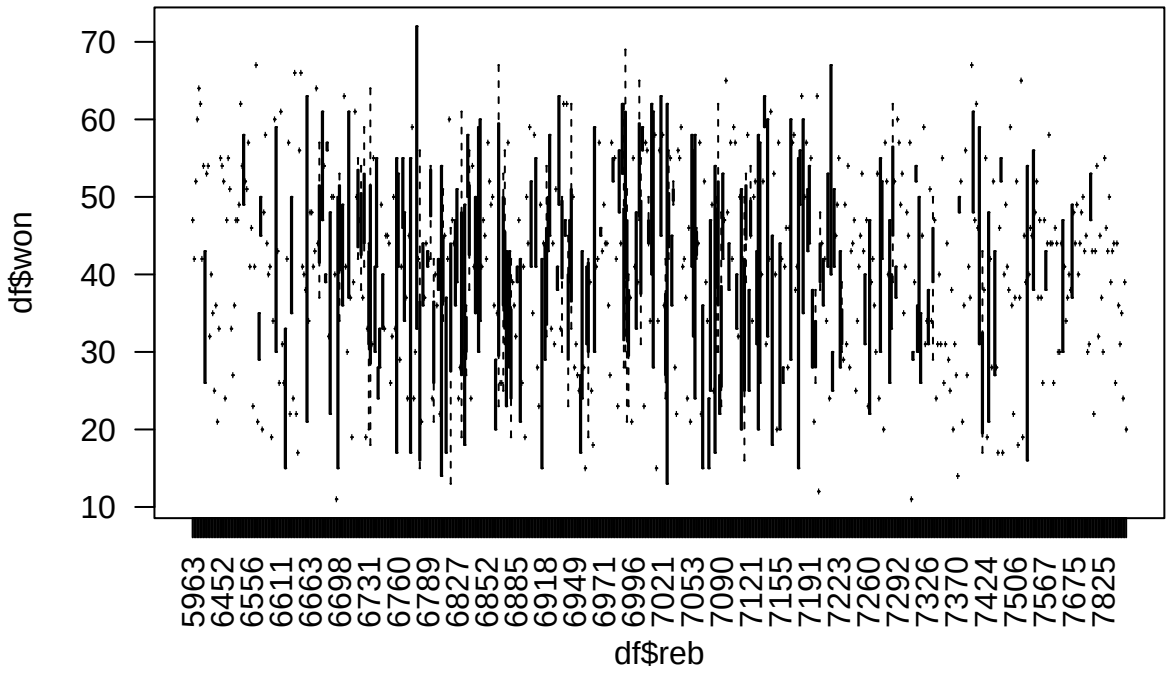
Correlation Matrix between Predictor and Outcome variables



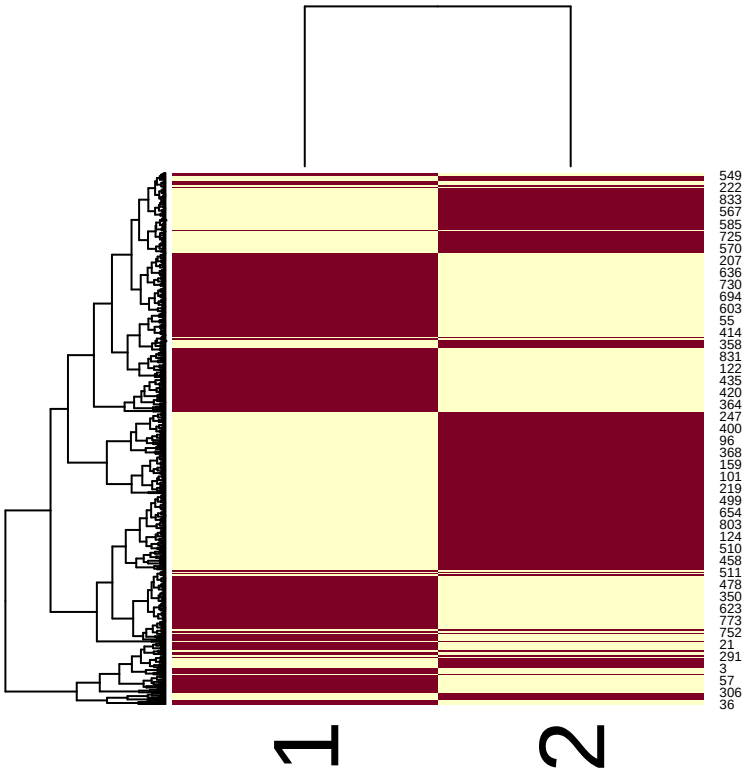

```r
barplot(df$reb, col = c("#1b98e0", "#353436"))

legend("topright", legend = c("Group 1", "Group 2"), fill = c("#1b98e0", "#353436"))
```




```
boxplot(df$won ~ df$reb, las=2)
```




```
heatmap(cbind(df$d_reb, df$o_reb))
```


549
222
833
567
585
725
570
207
636
730
694
603
55
414
358
831
122
435
420
364
247
400
96
368
159
101
219
499
654
803
124
510
458
511
478
350
623
773
752
21
291
3
57
306
36
1
2