CSE 270 Sports Analytics

Homework 3

Name: Anna Martirosyan

11.11.2022

Competitive Balance

Problem 1 (50 points)

Noll-Scully number

1. Get the final tables for all countries in the dataset *f_data_sm* for all seasons(for each country). (10 points)

```
final_df = data.frame()
data(f_data_sm)
for (j in  unique(f_data_sm$COUNTRY)){
  for(i in unique(f_data_sm[f_data_sm$COUNTRY == j,]$SEASON)) {
    output = final_table(f_data_sm, j, i)
    output$Season = i
    output$Country = j
    final_df = rbind(final_df, output)
  }}

head(final_df, n=5)
```

```
##         TEAM  M  W  D  L GF GA DIFF POINTS POSITION Season Country
## 1 Man United 42 27 11  4 80 38   42     92        1   1994 England
## 2  Blackburn 42 25  9  8 63 36   27     84        2   1994 England
## 3  Newcastle 42 23  8 11 82 41   41     77        3   1994 England
## 4    Arsenal 42 18 17  7 53 28   25     71        4   1994 England
## 5      Leeds 42 18 16  8 65 39   26     70        5   1994 England
```

2. Calculate the wining percentage(ratio) for all the teams in all seasons. (Consider draws as half wins). (2 points)

```
final_df$WP = as.numeric(format(
  round(((final_df$W + final_df$D/2)/(final_df$W+final_df$D+final_df$L))*100,
                        1), nsmall = 1))
head(final_df,n=5)
```

```
##         TEAM  M  W  D  L GF GA DIFF POINTS POSITION Season Country   WP
## 1 Man United 42 27 11  4 80 38   42     92        1   1994 England 77.4
## 2  Blackburn 42 25  9  8 63 36   27     84        2   1994 England 70.2
## 3  Newcastle 42 23  8 11 82 41   41     77        3   1994 England 64.3
## 4    Arsenal 42 18 17  7 53 28   25     71        4   1994 England 63.1
## 5      Leeds 42 18 16  8 65 39   26     70        5   1994 England 61.9
```

3. Calculate the Noll-Scully number for each season and country using the winning ratio. (3 points)

```
id_s = 0.5/sqrt(16)
final_df$WR = as.numeric(format(
  round(((final_df$W + final_df$D/2)/(final_df$W+final_df$D+final_df$L)),
                         3), nsmall = 1))
final_dff<- final_df %>%
   group_by(Season, Country) %>%
   summarise(NS=sd(WR)/id_s)

head(final_dff, n=10)
```

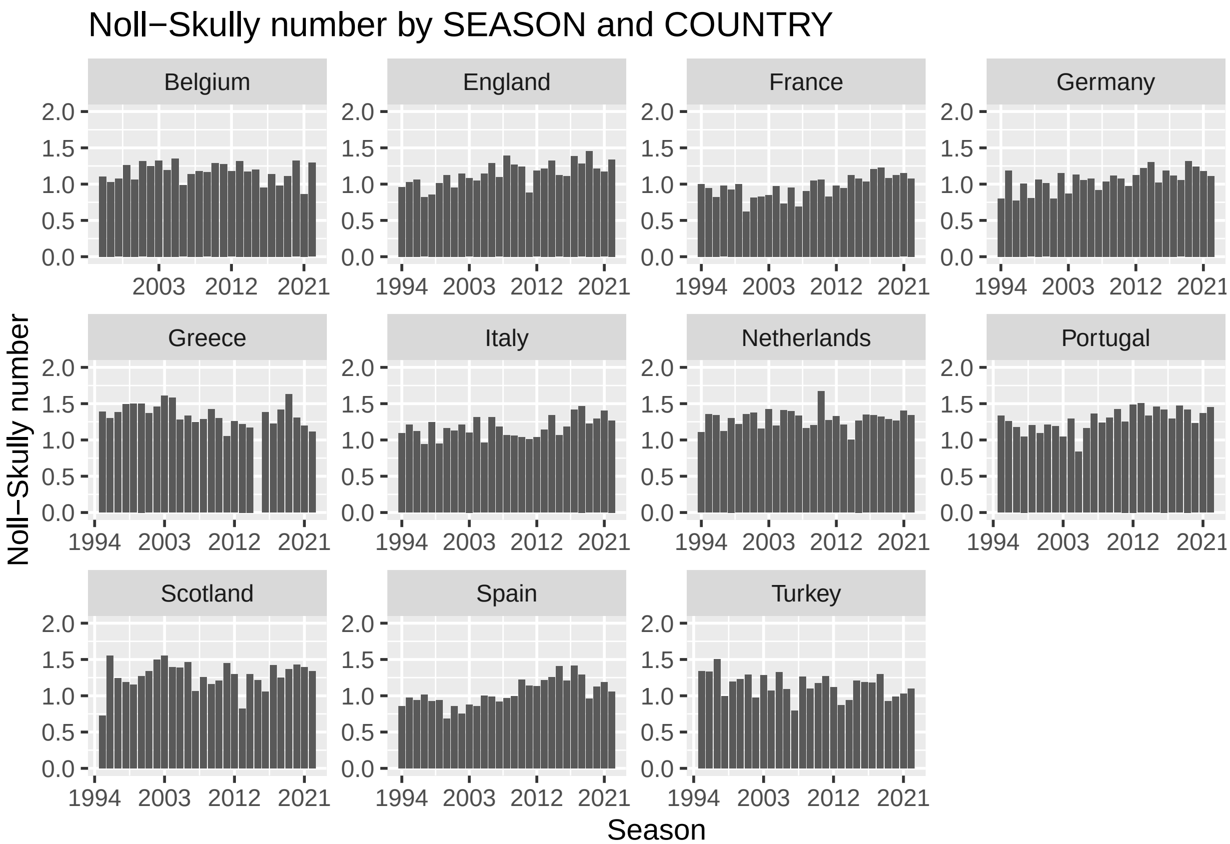
```
## # A tibble: 10 x 3
## # Groups:   Season [2]
##    Season Country        NS
##     <dbl> <chr>       <dbl>
##  1   1994 England     0.960
##  2   1994 France      1.00
##  3   1994 Germany     0.801
##  4   1994 Italy       1.09
##  5   1994 Netherlands 1.11
##  6   1994 Spain       0.861
##  7   1995 England     1.03
##  8   1995 France      0.949
##  9   1995 Germany     1.19
## 10   1995 Greece      1.39
```

Feedback Problem 1.3
2/3

It is not always the case that number of teams is equal to 16, it can be changed season or country based, so you should have used dynamic calculation for that

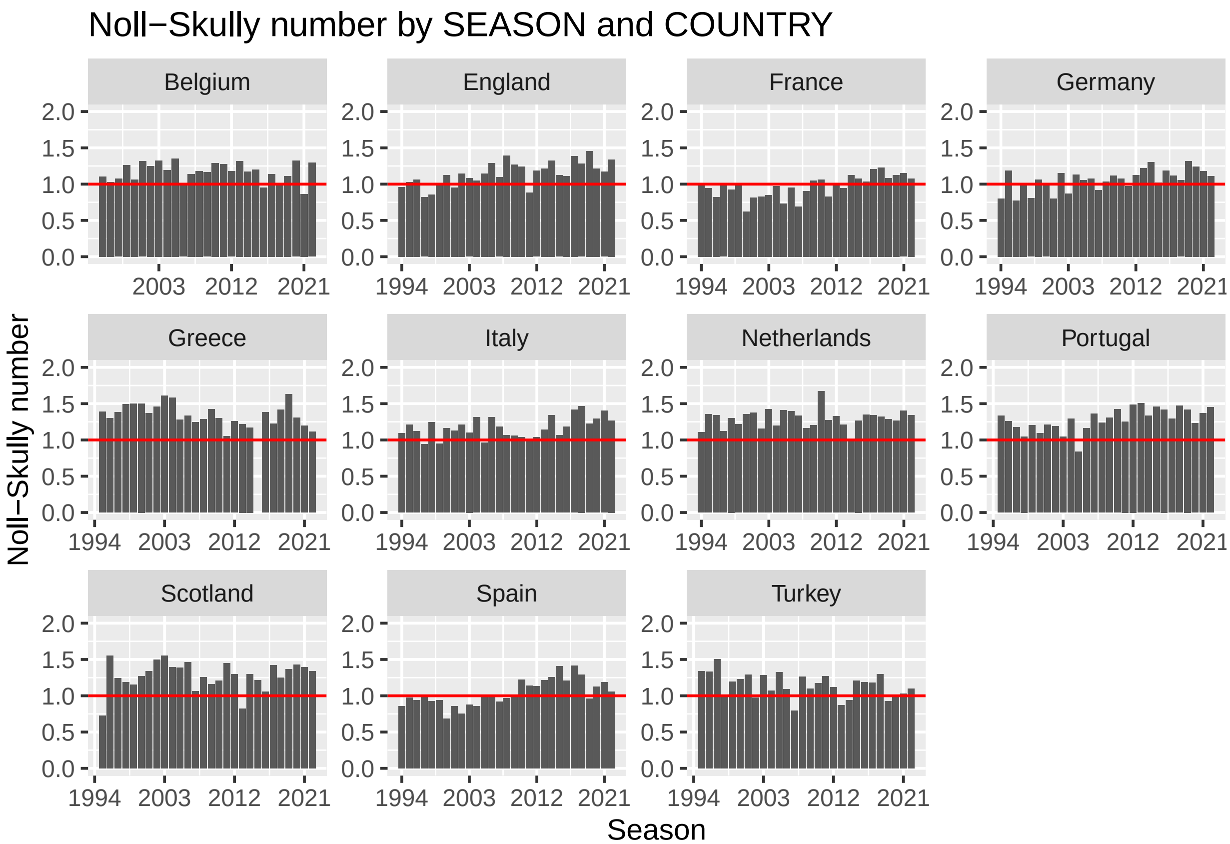
4. Visualize the results grouped by seasons and countries. Make sure to include meaningful title and axis names. (You can use facets) (10 points)

```
ggplot(final_dff, aes(Season, NS))+
  geom_bar(stat = "identity")+
  facet_wrap(~Country,scales = "free")+
  ylab("Noll-Skully number")+
  ylim(0, 2)+
  scale_x_continuous(breaks = seq(1994, 2022, 9))+
  ggtitle("Noll-Skully number by SEASON and COUNTRY")
```



5.Interpret the visualization (5 points)

```
ggplot(final_dff, aes(Season, NS))+
  geom_bar(stat = "identity")+
  facet_wrap(~Country,scales = "free")+
  ylab("Noll-Skully number")+
  ylim(0, 2)+
  scale_x_continuous(breaks = seq(1994, 2022, 9))+
  ggtitle("Noll-Skully number by SEASON and COUNTRY")+
  geom_hline(yintercept=1, col="red", size=0.5)
```




```
# To my interpretation I added also the below graph, with line passing through
# y=1 to clearly show that if the number is above 1.0 it means the teams are
# further away in wins than we would expect given the ideal. If the number is
# below 1.0, it means the  teams are closer in wins than we would expect.
# Higher is the number, lower is Competitive Balance. Closer is the number to
# 1, higher is Competitive Balance.
# As we can see from the visualizations in our case for most Countries
# Noll-Skully number is below 1, which means that the  teams are closer in wins
# than we would expect.
```

Good job :) !

C5 competitive balance

6.Calculate the C5 index for the number of goals scored by the top 5 teams in each country for each season. You should use the dataset from Problem 1.1. (5 points)

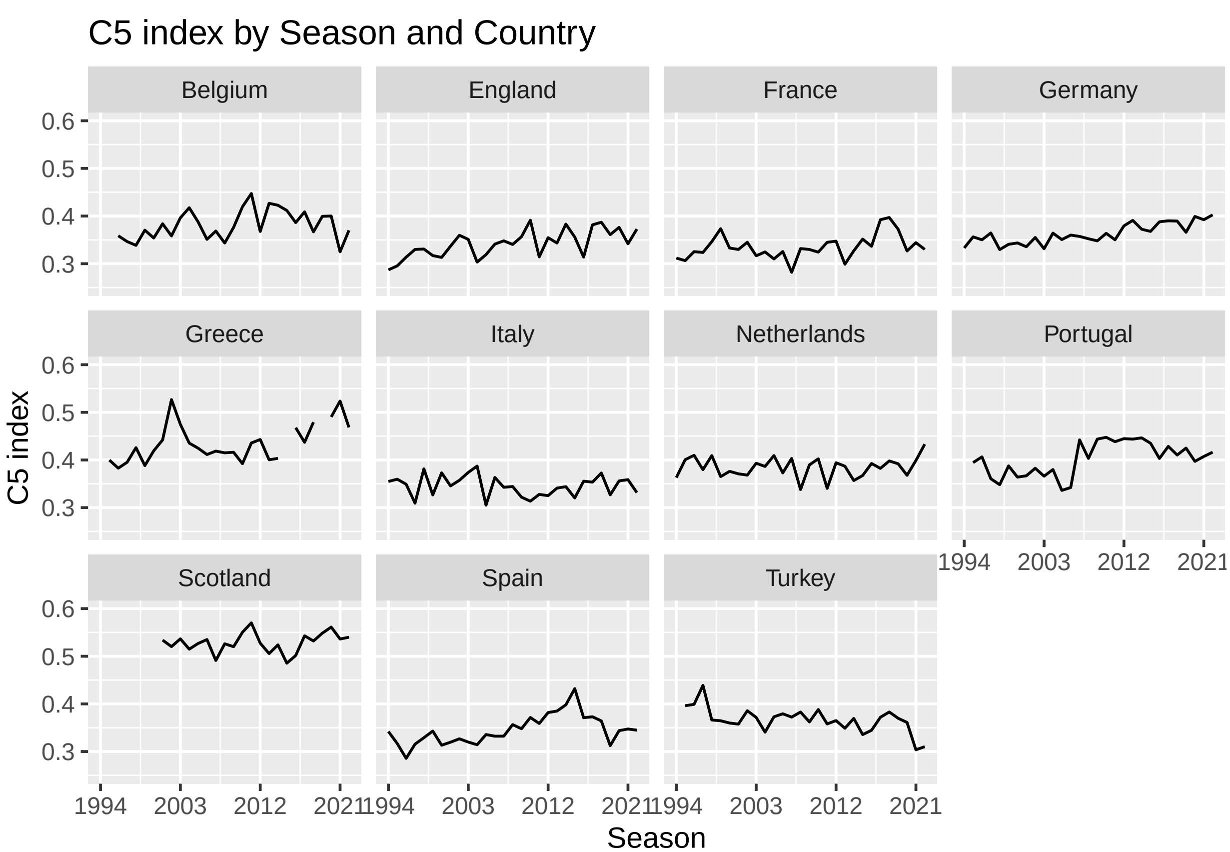
```
top5 <- final_df %>%
  group_by(Season, Country) %>%
  filter(POSITION <6) %>%
  summarize(TopP=sum(GF) )

all_teams <- final_df %>%
  group_by(Season, Country) %>%
  summarize(Pt=sum(GF))
```

```r
C5<-data.frame(Season=top5$Season,Country = top5$Country,
               C5=top5$TopP/all_teams$Pt)
head(C5,n=5)
```

```
##   Season     Country        C5
## 1   1994     England 0.2870293
## 2   1994      France 0.3117647
## 3   1994     Germany 0.3329609
## 4   1994       Italy 0.3549258
## 5   1994 Netherlands 0.3629301
```

7. Visualize the results for all countries and seasons. (Can be the same structure as in 1.4) (10 points)

```r
ggplot(C5, aes(Season, C5))+
  geom_line()+
  ylab("C5 index")+
  ylim(0.25, 0.6)+
  scale_x_continuous(breaks = seq(1994, 2022, 9))+
  ggtitle("C5 index by Season and Country")+
  facet_wrap(~Country)
```



8. Interpret the results (5 points)

The above graphs show the proportion of championships (or points) by top 5 teams, in other words the dominance of top 5 teams. For example from the graphs we can conclude that in 2003 for Portugal the points gained from top5 teams were approximately 40% of total number of points earned by all teams. The highest numbers can be seen in Scotland in 2011 (~ 57%) and in Greece in 2002 (~ 53%).

Feedback Problem 1.8
4/5

Not points, goals :)

Overall Problem 1
48/50

Individual Performance (50 points)

Problem 2

1. Load the dataset *nba_players* from the library *SportsAnalytics270* and filter it to only include seasons starting from 1992.
   Remove the columns blanl and blank2 and drop missing values or nas. (2 points)

```
library(tidyr)
data(nba_players)
nba_players = nba_players %>%
  filter(Year >= 1992)
nba_players <- nba_players[ , ! names(nba_players) %in% c("blanl", "blank2")]
nba_players <- nba_players %>% drop_na()
head(nba_players, n=6)
```

```
##       X Year             Player Pos Age  Tm  G GS   MP  PER   TS. X3PAr   FTr
## 1 10450 1992 Mahmoud Abdul-Rauf  PG  22 DEN 81 11 1538 12.6 0.469 0.111 0.128
## 2 10451 1992         Mark Acres   C  29 ORL 68  6  926 10.1 0.576 0.020 0.444
## 3 10452 1992      Michael Adams  PG  29 WSB 78 78 2795 17.1 0.506 0.313 0.292
## 4 10453 1992     Rafael Addison  SF  27 NJN 76  8 1175 10.9 0.477 0.113 0.176
## 5 10454 1992       Mark Aguirre  SF  32 DET 75 12 1582 13.7 0.479 0.090 0.292
## 6 10455 1992        Danny Ainge  SG  32 POR 81  6 1595 15.4 0.534 0.340 0.194
##   ORB. DRB. TRB. AST. STL. BLK. TOV. USG.  OWS DWS  WS WS.48 OBPM DBPM  BPM
## 1  1.5  6.8  4.0 21.0  1.4  0.2 11.6 26.7 -0.2 0.8 0.6 0.018 -1.7 -3.5 -5.2
## 2 11.4 19.2 15.2  3.2  1.3  1.0 15.5  9.6  1.1 0.8 1.9 0.097 -2.3  0.1 -2.2
## 3  2.2 10.1  6.1 31.5  2.5  0.2 13.2 24.2  2.6 2.6 5.3 0.090  2.5 -0.8  1.7
## 4  5.7  9.3  7.4  8.1  1.2  1.4  9.0 17.5  0.8 0.7 1.5 0.063 -2.3 -1.5 -3.8
## 5  4.8 12.0  8.4 13.4  1.7  0.4 10.6 27.6  0.1 2.0 2.2 0.066 -1.6 -1.0 -2.6
## 6  2.7  7.4  5.1 18.4  2.2  0.5  8.7 20.5  3.1 2.0 5.2 0.155  1.9 -0.9  1.0
##   VORP  FG  FGA   FG. X3P X3PA  X3P. X2P X2PA  X2P.  eFG.  FT FTA   FT. ORB DRB
## 1 -1.3 356  845 0.421  31   94 0.330 325  751 0.433 0.440  94 108 0.870  22  92
## 2 -0.1  78  151 0.517   1    3 0.333  77  148 0.520 0.520  51  67 0.761  97 155
## 3  2.6 485 1233 0.393 125  386 0.324 360  847 0.425 0.444 313 360 0.869  58 252
## 4 -0.5 187  432 0.433  14   49 0.286 173  383 0.452 0.449  56  76 0.737  65 100
## 5 -0.2 339  787 0.431  15   71 0.211 324  716 0.453 0.440 158 230 0.687  67 169
## 6  1.2 299  676 0.442  78  230 0.339 221  446 0.496 0.500 108 131 0.824  40 108
##   TRB AST STL BLK TOV  PF  PTS
## 1 114 192  44   4 117 130  837
## 2 252  22  25  15  33 140  208
## 3 310 594 145   9 212 162 1408
## 4 165  68  28  28  46 109  444
```

```
## 5 236 126  51  11 105 171  851
## 6 148 202  73  13  70 148  784
```

2. Clean the variable Pos from players with double positions, for example C-PF should be turned into C. (5 points)

```
nba_players$Pos <- trimws(nba_players$Pos, whitespace = '-.*')
head(nba_players,n=6)
```

```
##       X Year             Player Pos Age  Tm  G GS   MP  PER   TS. X3PAr   FTr
## 1 10450 1992 Mahmoud Abdul-Rauf  PG  22 DEN 81 11 1538 12.6 0.469 0.111 0.128
## 2 10451 1992         Mark Acres   C  29 ORL 68  6  926 10.1 0.576 0.020 0.444
## 3 10452 1992      Michael Adams  PG  29 WSB 78 78 2795 17.1 0.506 0.313 0.292
## 4 10453 1992     Rafael Addison  SF  27 NJN 76  8 1175 10.9 0.477 0.113 0.176
## 5 10454 1992       Mark Aguirre  SF  32 DET 75 12 1582 13.7 0.479 0.090 0.292
## 6 10455 1992        Danny Ainge  SG  32 POR 81  6 1595 15.4 0.534 0.340 0.194
##   ORB. DRB. TRB. AST. STL. BLK. TOV. USG.  OWS DWS  WS WS.48 OBPM DBPM  BPM
## 1  1.5  6.8  4.0 21.0  1.4  0.2 11.6 26.7 -0.2 0.8 0.6 0.018 -1.7 -3.5 -5.2
## 2 11.4 19.2 15.2  3.2  1.3  1.0 15.5  9.6  1.1 0.8 1.9 0.097 -2.3  0.1 -2.2
## 3  2.2 10.1  6.1 31.5  2.5  0.2 13.2 24.2  2.6 2.6 5.3 0.090  2.5 -0.8  1.7
## 4  5.7  9.3  7.4  8.1  1.2  1.4  9.0 17.5  0.8 0.7 1.5 0.063 -2.3 -1.5 -3.8
## 5  4.8 12.0  8.4 13.4  1.7  0.4 10.6 27.6  0.1 2.0 2.2 0.066 -1.6 -1.0 -2.6
## 6  2.7  7.4  5.1 18.4  2.2  0.5  8.7 20.5  3.1 2.0 5.2 0.155  1.9 -0.9  1.0
##   VORP  FG  FGA   FG. X3P X3PA  X3P. X2P X2PA  X2P.  eFG.  FT FTA   FT. ORB DRB
## 1 -1.3 356  845 0.421  31   94 0.330 325  751 0.433 0.440  94 108 0.870  22  92
## 2 -0.1  78  151 0.517   1    3 0.333  77  148 0.520 0.520  51  67 0.761  97 155
## 3  2.6 485 1233 0.393 125  386 0.324 360  847 0.425 0.444 313 360 0.869  58 252
## 4 -0.5 187  432 0.433  14   49 0.286 173  383 0.452 0.449  56  76 0.737  65 100
## 5 -0.2 339  787 0.431  15   71 0.211 324  716 0.453 0.440 158 230 0.687  67 169
## 6  1.2 299  676 0.442  78  230 0.339 221  446 0.496 0.500 108 131 0.824  40 108
##   TRB AST STL BLK TOV  PF  PTS
## 1 114 192  44   4 117 130  837
## 2 252  22  25  15  33 140  208
## 3 310 594 145   9 212 162 1408
## 4 165  68  28  28  46 109  444
## 5 236 126  51  11 105 171  851
## 6 148 202  73  13  70 148  784
```

3. Convert the vaiable Pos into a factor and remove the columns X, Year, Player and Tm. (3 points)

```
nba_players$Pos = as.factor(nba_players$Pos)
nba_players <- nba_players[ , ! names(nba_players) %in%
                              c("X", "Year", "Player", "Tm")]
head(nba_players, n=5)
```

```
##   Pos Age  G GS   MP  PER   TS. X3PAr   FTr ORB. DRB. TRB. AST. STL. BLK. TOV.
## 1  PG  22 81 11 1538 12.6 0.469 0.111 0.128  1.5  6.8  4.0 21.0  1.4  0.2 11.6
## 2   C  29 68  6  926 10.1 0.576 0.020 0.444 11.4 19.2 15.2  3.2  1.3  1.0 15.5
## 3  PG  29 78 78 2795 17.1 0.506 0.313 0.292  2.2 10.1  6.1 31.5  2.5  0.2 13.2
## 4  SF  27 76  8 1175 10.9 0.477 0.113 0.176  5.7  9.3  7.4  8.1  1.2  1.4  9.0
## 5  SF  32 75 12 1582 13.7 0.479 0.090 0.292  4.8 12.0  8.4 13.4  1.7  0.4 10.6
##   USG.  OWS DWS  WS WS.48 OBPM DBPM  BPM VORP  FG  FGA   FG. X3P X3PA  X3P. X2P
```

```
## 1 26.7 -0.2 0.8 0.6 0.018 -1.7 -3.5 -5.2 -1.3 356  845 0.421  31   94 0.330 325
## 2  9.6  1.1 0.8 1.9 0.097 -2.3  0.1 -2.2 -0.1  78  151 0.517   1    3 0.333  77
## 3 24.2  2.6 2.6 5.3 0.090  2.5 -0.8  1.7  2.6 485 1233 0.393 125  386 0.324 360
## 4 17.5  0.8 0.7 1.5 0.063 -2.3 -1.5 -3.8 -0.5 187  432 0.433  14   49 0.286 173
## 5 27.6  0.1 2.0 2.2 0.066 -1.6 -1.0 -2.6 -0.2 339  787 0.431  15   71 0.211 324
##   X2PA  X2P.  eFG.  FT FTA   FT. ORB DRB TRB AST STL BLK TOV  PF  PTS
## 1  751 0.433 0.440  94 108 0.870  22  92 114 192  44   4 117 130  837
## 2  148 0.520 0.520  51  67 0.761  97 155 252  22  25  15  33 140  208
## 3  847 0.425 0.444 313 360 0.869  58 252 310 594 145   9 212 162 1408
## 4  383 0.452 0.449  56  76 0.737  65 100 165  68  28  28  46 109  444
## 5  716 0.453 0.440 158 230 0.687  67 169 236 126  51  11 105 171  851
```

4. Divide the dataset into train and test datasets. (You can use dplyr or any other sampling method). (10 points)

```r
training_dataset  <- nba_players %>% dplyr::sample_frac(0.7)
testing_dataset   <- dplyr::anti_join(nba_players, training_dataset)

head(training_dataset, n=5)
```

```
##   Pos Age  G GS   MP  PER   TS. X3PAr   FTr ORB. DRB. TRB. AST. STL. BLK. TOV.
## 1  PF  28 52 52 1434 14.1 0.530 0.423 0.215  7.3 14.5 10.7  6.1  1.3  1.4  9.6
## 2  SF  28 31  7  731 11.1 0.487 0.100 0.264  3.3 11.4  7.0  7.8  2.2  0.5 11.3
## 3  PG  23 56  0  535 11.5 0.494 0.036 0.343  5.2  8.7  7.0 15.7  3.2  1.3 17.8
## 4  SG  27 28  0   99  9.7 0.469 0.478 0.043  1.1 10.2  5.5 21.5  0.5  0.0 13.0
## 5   C  30 66 66 2015 18.0 0.505 0.007 0.288  7.5 30.1 18.9 12.0  1.6  7.4 13.2
##   USG.  OWS DWS  WS  WS.48 OBPM DBPM  BPM VORP  FG FGA   FG. X3P X3PA  X3P. X2P
## 1 18.7  1.8 1.5 3.3  0.111  0.6  0.1  0.7  1.0 216 508 0.425  78  215 0.363 138
## 2 17.5  0.4 0.7 1.1  0.074 -2.3 -0.3 -2.6 -0.1  95 231 0.411   6   23 0.261  89
## 3 18.5 -0.2 0.5 0.3  0.028 -3.2  0.0 -3.2 -0.2  77 169 0.456   0    6 0.000  77
## 4 23.2 -0.1 0.0 0.0 -0.012 -2.1 -3.9 -6.0 -0.1  19  46 0.413   6   22 0.273  13
## 5 17.2  1.2 4.5 5.8  0.138 -2.4  5.8  3.4  2.8 279 600 0.465   0    4 0.000 279
##   X2PA  X2P.  eFG.  FT FTA   FT. ORB DRB TRB AST STL BLK TOV  PF PTS
## 1  293 0.471 0.502  79 109 0.725  99 183 282  55  38  27  59 120 589
## 2  208 0.428 0.424  55  61 0.902  23  69  92  36  30   5  33  68 251
## 3  163 0.472 0.456  38  58 0.655  26  43  69  59  35  12  42  68 192
## 4   24 0.542 0.478   0   2 0.000   1   9  10  13   1   0   7   5  44
## 5  596 0.468 0.465 125 173 0.723 131 530 661 152  61 199 103 177 683
```

```r
head(testing_dataset, n=5)
```

```
##   Pos Age  G GS   MP  PER   TS. X3PAr   FTr ORB. DRB. TRB. AST. STL. BLK. TOV.
## 1  PG  29 78 78 2795 17.1 0.506 0.313 0.292  2.2 10.1  6.1 31.5  2.5  0.2 13.2
## 2  PF  30 84 18 2104 14.5 0.498 0.002 0.323  6.3 19.5 12.8  5.5  0.8  3.3 10.2
## 3  PF  30 13  0  327 12.0 0.487 0.010 0.545  8.3 18.1 13.5  8.0  0.8  2.8 14.4
## 4  PF  30 71 18 1777 14.9 0.500 0.001 0.293  6.0 19.7 12.6  5.0  0.9  3.4  9.5
## 5  SF  28 75 75 2881 24.5 0.612 0.122 0.580 10.9 22.0 16.5 18.1  2.4  0.9 14.3
##   USG. OWS DWS   WS WS.48 OBPM DBPM  BPM VORP  FG  FGA   FG. X3P X3PA  X3P. X2P
## 1 24.2 2.6 2.6  5.3 0.090  2.5 -0.8  1.7  2.6 485 1233 0.393 125  386 0.324 360
## 2 21.5 0.8 1.8  2.6 0.059 -3.0 -0.8 -3.8 -1.0 368  836 0.440   0    2 0.000 368
## 3 19.3 0.1 0.4  0.5 0.079 -3.0  1.0 -2.0  0.0  39  101 0.386   0    1 0.000  39
## 4 21.9 0.7 1.4  2.0 0.055 -3.0 -1.1 -4.2 -1.0 329  735 0.448   0    1 0.000 329
```

```
## 5 25.1 8.6 3.7 12.3 0.205  5.9  1.5  7.4  6.9 622 1126 0.552  32  137 0.234 590
##   X2PA  X2P.  eFG.  FT FTA   FT. ORB DRB TRB AST STL BLK TOV  PF  PTS
## 1  847 0.425 0.444 313 360 0.869  58 252 310 594 145   9 212 162 1408
## 2  834 0.441 0.440 215 270 0.796 122 363 485  78  35 117 108 160  951
## 3  100 0.390 0.386  44  55 0.800  23  55  78  19   5  15  21  31  122
## 4  734 0.448 0.448 171 215 0.795  99 308 407  59  30 102  87 129  829
## 5  989 0.597 0.567 454 653 0.695 271 559 830 308 136  44 235 196 1730
```

5. Build a classifier model to predict the position of the player based on the playing attributes. Use the training set for fitting the model. (20 points) You can use multinomial regression for that.

```
training_dataset$Pos <- relevel(training_dataset$Pos, ref = "C")
multinom_model <- multinom(Pos ~ ., data = nba_players)
```

```
## # weights:  240 (188 variable)
## initial  value 18363.686581
## iter  10 value 12537.187069
## iter  20 value 11878.866944
## iter  30 value 11241.499937
## iter  40 value 11013.246448
## iter  50 value 10918.925560
## iter  60 value 10869.655966
## iter  70 value 10795.907981
## iter  80 value 10701.265320
## iter  90 value 10628.070543
## iter 100 value 10459.415163
## final  value 10459.415163
## stopped after 100 iterations
```

```
#summary(multinom_model)

(exp(coef(multinom_model)))
```

```
##    (Intercept)       Age         G        GS       MP      PER       TS.
## PF   0.5759274 0.9948658 0.9912748 0.9700737 1.003881 1.268119 0.9733364
## PG   2.2202277 0.9209468 1.0074331 0.9859960 1.002672 1.333966 1.2562552
## SF   4.8551798 0.9514355 0.9961174 0.9816635 1.004374 1.039463 1.5341185
## SG   5.4863899 0.9326219 1.0026193 0.9811238 1.003837 1.081970 2.1076839
##       X3PAr       FTr      ORB.      DRB.      TRB.      AST.     STL.
## PF 2.710283 0.5727255 0.8401767 0.8541605 1.3503717 0.9218644 1.904089
## PG 4.027133 0.6852427 0.4919984 0.6652818 1.4074061 1.0877497 1.396248
## SF 3.079851 1.1659500 0.9248737 1.0267948 0.8536728 0.9371411 1.546185
## SG 2.358125 2.4499499 0.9476481 0.9882311 0.7547182 0.9636265 1.549671
##         BLK.      TOV.      USG.       OWS       DWS        WS     WS.48
## PF 0.5118675 1.0034153 0.9405972 0.6352595 0.7440891 0.7006567 0.9911539
## PG 0.3687022 1.0139052 1.0223610 1.3587630 0.5674361 0.7704398 1.1767805
## SF 0.5883296 0.9644162 1.0821104 1.0326167 0.7045973 0.7411097 0.8707177
## SG 0.4153573 0.9785214 1.1098591 1.1335489 0.7122432 0.7588789 0.8862776
##         OBPM     DBPM       BPM     VORP        FG      FGA       FG.       X3P
## PF 0.7376288 1.104072 0.9421705 6.417351 0.9959231 1.010547 0.9008958 0.9754742
## PG 0.6138567 1.350724 0.9315670 3.354826 0.9920975 1.013974 0.9853303 0.9921600
## SF 0.7750657 1.176852 1.0796131 4.504802 0.9956492 1.012821 1.3270770 0.9749451
```

```
## SG 0.8222049 1.479286 0.9244192 4.018367 0.9942080 1.014186 1.5446157 0.9786914
##        X3PA     X3P.      X2P      X2PA      X2P.     eFG.       FT       FTA
## PF 1.016194 2.292026 1.020963 0.9944437 1.1672804 1.030282 1.021644 0.9933171
## PG 1.009743 1.241222 0.999937 1.0041901 0.7865315 1.268064 1.008239 1.0054126
## SF 1.015892 4.432490 1.021236 0.9969766 1.8615832 1.476995 1.015537 0.9999293
## SG 1.014439 2.491824 1.015854 0.9997502 1.8295748 1.775249 1.016772 0.9938342
##          FT.       ORB       DRB       TRB       AST       STL       BLK
## PF 0.6392606 0.9994415 0.9995825 0.9990242 0.9917097 0.9581426 0.9940092
## PG 4.2133580 1.0118959 0.9861928 0.9979244 0.9913000 0.9993484 0.9936403
## SF 1.4991471 0.9997595 0.9956327 0.9953933 0.9909197 0.9846440 0.9910272
## SG 2.4113162 0.9979741 0.9959402 0.9939225 0.9866408 0.9918113 0.9927531
##         TOV        PF       PTS
## PF 0.990751 0.9970911 0.9884776
## PG 1.006555 0.9924032 0.9845861
## SF 1.007528 0.9916600 0.9814962
## SG 1.007330 0.9908058 0.9836118
```

```r
head(round(fitted(multinom_model), 2))
```

```
##      C   PF   PG   SF   SG
## 1 0.00 0.00 0.31 0.05 0.63
## 2 0.37 0.49 0.00 0.12 0.03
## 3 0.00 0.00 0.71 0.10 0.18
## 4 0.06 0.09 0.04 0.31 0.50
## 5 0.02 0.06 0.02 0.25 0.65
## 6 0.01 0.01 0.23 0.15 0.59
```

6. Predict the position of the players for the test dataset. (5 points)

```r
testing_dataset$ClassPredicted <- predict(multinom_model,
                                          newdata = testing_dataset, "class")
# Building classification table
table <- table(testing_dataset$Pos, testing_dataset$ClassPredicted)
table
```

```
##
##        C  PF  PG  SF  SG
##   C  309  97   1  39  10
##   PF 174 269   0 155  61
##   PG   2   1 660   9 159
##   SF  19  61   7 290 325
##   SG   5  14 110  92 554
```

7. Interpret the predictive power of the model. (5 points)

```r
training_dataset$ClassPredicted <- predict(multinom_model,
                                           newdata = training_dataset, "class")
# Building classification table
tab <- table(training_dataset$Pos, training_dataset$ClassPredicted)

# Calculating accuracy - sum of diagonal elements divided by total observations
round((sum(diag(tab))/sum(tab))*100,2)
```

```
## [1] 60.64
```

```
# My model accuracy has turned out to be 60.7% in the training dataset, which is
# good because as Professor Madoyan says, "if it is higher that 50%, than it is
# a good approximation" :)
```

Feedback Problem 2
50/50

Ovearll Anna Martirosyan: 98/100