Appendix A - Player Selection and Competitiveness

The Extras - SOA 2022 Student Challenge

22/02/2022

## Packages and Data Import

rm(list = ls())  
library(readxl)  
library(tidyverse)  
library(gridExtra)  
library(Hmisc)  
library(visdat)  
library(gridExtra)  
library(RColorBrewer)  
library(tidyverse)  
library(Hmisc)  
library(visdat)  
library(mice)  
library(VIM)  
library(Rcpp)  
library(actuar)  
library(caret)  
library(ROSE)  
library(stepPlr)  
library(randomForest)  
library(gbm)  
library(boot)  
library(glmnet)  
library(MASS)  
library(corrplot)  
library(janitor)  
library(UpSetR)  
library(naniar)  
library(lpSolve)  
  
  
Data\_Fwd <- read\_excel("2022-student-research-case-study-player-data.xlsx", sheet="League Shooting", skip=11) %>%  
 rename("Games" = `90s`)  
Data\_Mid <- read\_excel("2022-student-research-case-study-player-data.xlsx", sheet="League Passing", skip=11)%>%  
 dplyr::select(-`90s`)  
Data\_Back <- read\_excel("2022-student-research-case-study-player-data.xlsx", sheet="League Defense", skip=11)%>%  
 dplyr::select(-`90s`)  
Data\_Goal <- read\_excel("2022-student-research-case-study-player-data.xlsx", sheet="League Goalkeeping", skip=11)   
GK\_Data <- read\_excel("2022-student-research-case-study-player-data.xlsx", sheet="League Goalkeeping", skip=11) %>% dplyr::select (-Pos)  
  
  
Tourn\_FW <- read\_xlsx("2022-student-research-case-study-player-data.xlsx", sheet="Tournament Shooting", skip=1539, col\_names = FALSE)   
names(Tourn\_FW) <- colnames(read\_xlsx("2022-student-research-case-study-player-data.xlsx", sheet="Tournament Shooting", skip=11, n\_max = 1))  
Tourn\_FW <- Tourn\_FW %>%  
 clean\_names() %>%  
 rename("games" = `x90s`)  
Tourn\_MF <- read\_excel("2022-student-research-case-study-player-data.xlsx", sheet="Tournament Passing", skip=11)%>%  
 clean\_names()%>%  
 dplyr::select(-`x90s`)  
Tourn\_DF <- read\_excel("2022-student-research-case-study-player-data.xlsx", sheet="Tournament Defense", skip=11)%>%  
 clean\_names()%>%  
 dplyr::select(-`x90s`)  
Tourn\_GK <- read\_excel("2022-student-research-case-study-player-data.xlsx", sheet="Tournament Goalkeeping", skip=11)%>%  
 clean\_names()%>%  
 rename(performance\_p\_katt\_y = "performance\_p\_katt") %>%  
 filter(year == 2021)  
  
Tourn\_Result <- read\_excel("2022-student-research-case-study-player-data.xlsx", sheet="Tournament Results", skip=10)[,4:5]%>%  
 clean\_names() %>%  
 rename(place = "x2021\_tournament\_place", nation = "country\_5")  
  
Salary\_2020 <- read\_excel("2022-student-research-case-study-player-data.xlsx", sheet="2020 Salaries", skip=11) %>%  
 mutate("Year" = 2020) %>%  
 dplyr::select(Year, `Player Name`, Squad, `Annualized Salary`) %>%  
 rename(Player = "Player Name") %>%  
 rename(Salary = "Annualized Salary")  
Salary\_2021 <- read\_excel("2022-student-research-case-study-player-data.xlsx", sheet="2021 Salaries", skip=11) %>%  
 mutate("Year" = 2021) %>%  
 dplyr::select(Year, `Player Name`, Squad, `Annualized Salary`) %>%  
 rename(Player = "Player Name") %>%  
 rename(Salary = "Annualized Salary")

## Combine data sets (Except Goalkeepers)

# Bind Salary Data  
Salary <- rbind(Salary\_2020, Salary\_2021)  
  
  
#Merge salary data with player data  
Data <- Data\_Fwd %>%  
 left\_join(Data\_Back, by = c("Player", "Nation", "Pos", "Age", "Born", "Year", "Squad", "League")) %>%  
 left\_join(Data\_Mid, by = c("Player", "Nation", "Pos", "Age", "Born", "Year", "Squad", "League")) %>%  
 left\_join(Data\_Goal, by = c("Player", "Nation", "Pos", "Age", "Born", "Year", "Squad", "League")) %>%  
 left\_join(Salary, by = c("Player", "Year", "Squad")) #Many Players played multiple squads  
  
Data %>%  
 dplyr::filter(Nation == "Rarita") %>%  
 count() #330 from Rarita

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 330

## EDA League

* Missing data for one variable correlates with missing data from another variable.
* Forwards are paid most, followed by mids, defenders and then goalkeepers. Also see that those that can do multiple positions are paid more than their counterparts.
* Can see that the salary is dependent on league (local league is also significantly less)
* Average Rarita salary is one of the outliers at the lower end this is likely because the local league is included in the analysis and this is confirmed when RPL is removed.
* Squads are generally made up of 25-30 players and players were more expensive in 2021 than 2020 (Need to consider superimposed inflation). No significant variation in average salary between teams within a league.
* Can see peak age is around mid to late 20s. Seems to be uptake in late ages but only applies to unique few (low numbers either side). Quadratic relationship.
* Evidently there is lots of data missing for players which have not had a shot on target or have not had a shot at this can be looked at further but seems to correlate with those that have had little game time (indicating a lack of exposure but varies by position).
* Many outliers and inconsistencies in data.
* Some varibales (such as expected goals) that would be effected to one position (forwards for expected goals) also seem to have an effect other positions (defenders). One model for all positions may be beneficial.
* Missing data is generated simply because these individuals have never taken a shot on goal, had a shot on target, completed a pass a certain distance, tackled etc. These variables show this happens general as effect of not having enough exposure (game time) or based on an individuals position. Fortunately, these variables are made by combining other variables contained in the set therefore these ratio based variables can be removed as tree based models (GBM and RF will be used for this analysis and the interaction will naturally be considered.)
* Distance from goal is the only one that doesn’t fit into this description.No obvious correlation with player value.
* Squad is also going to be removed from the variables as it is more likely to reflect a willingness to pay (budget) of a team but does not reflect ability of the player. This is a way in which pricing is distorted and makes room to take advantage of under and over pricing.
* Looking at wins, draws and losses for goalkeepers shows a roughly normal dist with approx mean 1 and sd .2.
  + This supports using a tree based method as it will remove some of the randomness which seems to be injected into the playing data.
  + This data may show something but the exact value cant be trusted hence splitting will be more effective. This a common theme amongst all data.
  + In the same way a tree based method is not effected by outliers it will be less effected by these errors in the data.

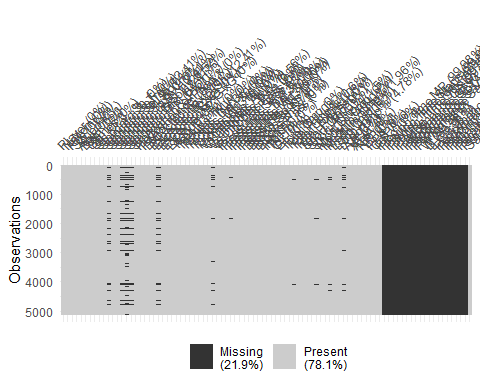
color <- brewer.pal.info[brewer.pal.info$category == 'qual',]  
colorvec <- unlist(mapply(brewer.pal, color$maxcolors, rownames(color)))  
  
  
summary(Data)

## Player Nation Pos Squad   
## Length:5555 Length:5555 Length:5555 Length:5555   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## Age Born Games Gls   
## Min. :14.00 Min. :1977 Min. :-0.10 Min. :-0.1000   
## 1st Qu.:22.00 1st Qu.:1991 1st Qu.: 3.85 1st Qu.:-0.0100   
## Median :25.00 Median :1994 Median :12.88 Median : 0.0600   
## Mean :25.25 Mean :1994 Mean :14.03 Mean : 0.1112   
## 3rd Qu.:28.00 3rd Qu.:1997 3rd Qu.:22.80 3rd Qu.: 0.1500   
## Max. :42.00 Max. :2005 Max. :38.10 Max. : 9.9600   
##   
## Standard Sh Standard SoT Standard SoT% Standard Sh/90   
## Min. :-0.100 Min. :-0.1000 Min. : -0.10 Min. :-0.100   
## 1st Qu.: 0.260 1st Qu.: 0.0400 1st Qu.: 17.40 1st Qu.: 0.260   
## Median : 0.780 Median : 0.1800 Median : 30.20 Median : 0.780   
## Mean : 1.155 Mean : 0.3671 Mean : 29.88 Mean : 1.186   
## 3rd Qu.: 1.740 3rd Qu.: 0.5600 3rd Qu.: 40.42 3rd Qu.: 1.750   
## Max. :30.040 Max. :10.1000 Max. :100.10 Max. :44.960   
## NA's :1031   
## Standard SoT/90 Standard G/Sh Standard G/SoT Standard Dist   
## Min. :-0.1000 Min. :-0.1000 Min. :-0.1000 Min. : 1.36   
## 1st Qu.: 0.0300 1st Qu.: 0.0000 1st Qu.: 0.0500 1st Qu.:13.19   
## Median : 0.1800 Median : 0.0700 Median : 0.2500 Median :17.16   
## Mean : 0.3693 Mean : 0.0833 Mean : 0.2736 Mean :17.34   
## 3rd Qu.: 0.5600 3rd Qu.: 0.1400 3rd Qu.: 0.4200 3rd Qu.:21.07   
## Max. :30.0200 Max. : 1.0900 Max. : 1.1000 Max. :74.08   
## NA's :1031 NA's :1769 NA's :1031   
## Standard FK Performance PK Performance PKatt.x Expected xG   
## Min. :-0.10000 Min. :-0.100000 Min. :-0.10000 Min. :-0.1000   
## 1st Qu.:-0.04000 1st Qu.:-0.050000 1st Qu.:-0.04000 1st Qu.: 0.0100   
## Median : 0.02000 Median : 0.010000 Median : 0.01000 Median : 0.0800   
## Mean : 0.04476 Mean : 0.009962 Mean : 0.01365 Mean : 0.1227   
## 3rd Qu.: 0.07000 3rd Qu.: 0.060000 3rd Qu.: 0.06000 3rd Qu.: 0.1700   
## Max. :10.01000 Max. : 3.360000 Max. : 3.30000 Max. : 4.0100   
##   
## Expected npxG Expected npxG/Sh Expected G-xG Expected np:G-xG   
## Min. :-0.1000 Min. :-0.0800 Min. :-4.10000 Min. :-4.08000   
## 1st Qu.: 0.0000 1st Qu.: 0.0300 1st Qu.:-0.07000 1st Qu.:-0.07000   
## Median : 0.0700 Median : 0.0900 Median :-0.01000 Median :-0.01000   
## Mean : 0.1112 Mean : 0.0906 Mean :-0.01232 Mean :-0.01136   
## 3rd Qu.: 0.1700 3rd Qu.: 0.1400 3rd Qu.: 0.05000 3rd Qu.: 0.05000   
## Max. : 4.0900 Max. : 0.7700 Max. :10.06000 Max. :10.07000   
## NA's :1031   
## League Year Tackles Tkl Tackles TklW   
## Length:5555 Min. :2020 Min. :-0.100 Min. :-0.100   
## Class :character 1st Qu.:2020 1st Qu.: 0.770 1st Qu.: 0.420   
## Mode :character Median :2021 Median : 1.510 Median : 0.890   
## Mean :2021 Mean : 1.605 Mean : 0.982   
## 3rd Qu.:2021 3rd Qu.: 2.210 3rd Qu.: 1.360   
## Max. :2021 Max. :19.980 Max. :10.090   
##   
## Tackles Def 3rd Tackles Mid 3rd Tackles Att 3rd Vs Dribbles Tkl   
## Min. :-0.1000 Min. :-0.1000 Min. :-0.1000 Min. :-0.1000   
## 1st Qu.: 0.1300 1st Qu.: 0.1900 1st Qu.: 0.0200 1st Qu.: 0.0900   
## Median : 0.6500 Median : 0.5200 Median : 0.1300 Median : 0.4400   
## Mean : 0.7685 Mean : 0.6275 Mean : 0.2088 Mean : 0.5326   
## 3rd Qu.: 1.1300 3rd Qu.: 0.8700 3rd Qu.: 0.3000 3rd Qu.: 0.7800   
## Max. :20.0900 Max. :10.1000 Max. :10.0600 Max. :10.1000   
##   
## Vs Dribbles Att Vs Dribbles Tkl% Vs Dribbles Past Pressures Press   
## Min. :-0.10 Min. : -0.10 Min. :-0.100 Min. : -0.10   
## 1st Qu.: 0.76 1st Qu.: 20.04 1st Qu.: 0.440 1st Qu.: 9.99   
## Median : 1.45 Median : 32.83 Median : 0.910 Median : 14.71   
## Mean : 1.59 Mean : 32.51 Mean : 1.059 Mean : 15.05   
## 3rd Qu.: 2.18 3rd Qu.: 44.79 3rd Qu.: 1.440 3rd Qu.: 19.54   
## Max. :30.05 Max. :100.10 Max. :20.040 Max. :119.93   
## NA's :653   
## Pressures Succ Pressures % Pressures Def 3rd Pressures Mid 3rd  
## Min. :-0.100 Min. : -0.10 Min. :-0.100 Min. :-0.100   
## 1st Qu.: 2.840 1st Qu.: 24.45 1st Qu.: 2.350 1st Qu.: 3.630   
## Median : 4.100 Median : 28.61 Median : 4.640 Median : 6.200   
## Mean : 4.239 Mean : 28.69 Mean : 4.692 Mean : 6.763   
## 3rd Qu.: 5.420 3rd Qu.: 33.21 3rd Qu.: 6.430 3rd Qu.: 9.340   
## Max. :39.970 Max. :100.10 Max. :59.940 Max. :60.050   
## NA's :195   
## Pressures Att 3rd Blocks Blocks Blocks Sh Blocks ShSv   
## Min. :-0.100 Min. :-0.100 Min. :-0.100 Min. :-0.100000   
## 1st Qu.: 0.670 1st Qu.: 0.770 1st Qu.: 0.030 1st Qu.:-0.040000   
## Median : 2.610 Median : 1.400 Median : 0.160 Median : 0.000000   
## Mean : 3.593 Mean : 1.435 Mean : 0.282 Mean : 0.005233   
## 3rd Qu.: 5.325 3rd Qu.: 1.940 3rd Qu.: 0.410 3rd Qu.: 0.050000   
## Max. :50.090 Max. :19.970 Max. :10.010 Max. : 1.210000   
##   
## Blocks Pass Int Tkl+Int Clr   
## Min. :-0.100 Min. :-0.1000 Min. :-0.100 Min. :-0.100   
## 1st Qu.: 0.590 1st Qu.: 0.2150 1st Qu.: 1.230 1st Qu.: 0.340   
## Median : 1.070 Median : 0.7000 Median : 2.370 Median : 1.170   
## Mean : 1.154 Mean : 0.7984 Mean : 2.403 Mean : 1.966   
## 3rd Qu.: 1.540 3rd Qu.: 1.1700 3rd Qu.: 3.320 3rd Qu.: 2.945   
## Max. :19.950 Max. :19.9100 Max. :29.980 Max. :20.050   
##   
## Err Total Cmp Total Att Total Cmp%   
## Min. :-0.10000 Min. : -0.10 Min. : -0.10 Min. : -0.06   
## 1st Qu.:-0.04000 1st Qu.: 22.44 1st Qu.: 31.07 1st Qu.: 72.07   
## Median : 0.02000 Median : 34.01 Median : 43.61 Median : 78.41   
## Mean : 0.02552 Mean : 86.71 Mean : 107.36 Mean : 77.59   
## 3rd Qu.: 0.07000 3rd Qu.: 48.46 3rd Qu.: 58.30 3rd Qu.: 84.74   
## Max. : 5.09000 Max. :2514.90 Max. :2784.98 Max. :100.10   
## NA's :31   
## Total TotDist Total PrgDist Short Cmp Short Att   
## Min. : -0.09 Min. : -0.1 Min. : -0.10 Min. : -0.10   
## 1st Qu.: 410.85 1st Qu.: 105.0 1st Qu.: 9.34 1st Qu.: 11.32   
## Median : 674.59 Median : 213.9 Median : 13.90 Median : 16.23   
## Mean : 1690.31 Mean : 544.0 Mean : 34.80 Mean : 39.50   
## 3rd Qu.: 951.65 3rd Qu.: 333.9 3rd Qu.: 20.14 3rd Qu.: 22.80   
## Max. :48037.97 Max. :24424.1 Max. :1077.03 Max. :1156.03   
##   
## Short Cmp% Medium Cmp Medium Att Medium Cmp%   
## Min. : -0.08 Min. : -0.100 Min. : -0.10 Min. : -0.09   
## 1st Qu.: 83.06 1st Qu.: 8.005 1st Qu.: 10.02 1st Qu.: 77.21   
## Median : 88.20 Median : 14.210 Median : 16.98 Median : 85.06   
## Mean : 86.91 Mean : 37.493 Mean : 43.12 Mean : 83.46   
## 3rd Qu.: 92.12 3rd Qu.: 21.810 3rd Qu.: 24.64 3rd Qu.: 91.91   
## Max. :100.10 Max. :1338.030 Max. :1391.08 Max. :100.10   
## NA's :89 NA's :105   
## Long Cmp Long Att Long Cmp% Ast   
## Min. : -0.10 Min. : -0.100 Min. : -0.10 Min. :-0.1000   
## 1st Qu.: 2.17 1st Qu.: 4.045 1st Qu.: 49.66 1st Qu.:-0.0200   
## Median : 4.63 Median : 7.980 Median : 60.09 Median : 0.0500   
## Mean : 12.80 Mean : 20.636 Mean : 59.89 Mean : 0.2093   
## 3rd Qu.: 8.17 3rd Qu.: 12.670 3rd Qu.: 71.38 3rd Qu.: 0.1400   
## Max. :523.93 Max. :881.010 Max. :100.10 Max. :12.0700   
## NA's :246   
## xA A-xA KP 1/3   
## Min. :-0.1000 Min. :-6.17000 Min. :-0.100 Min. : -0.100   
## 1st Qu.: 0.0000 1st Qu.:-0.07000 1st Qu.: 0.110 1st Qu.: 1.080   
## Median : 0.0700 Median :-0.01000 Median : 0.690 Median : 2.230   
## Mean : 0.1971 Mean : 0.01152 Mean : 1.925 Mean : 6.173   
## 3rd Qu.: 0.1500 3rd Qu.: 0.06000 3rd Qu.: 1.320 3rd Qu.: 3.860   
## Max. : 9.9200 Max. : 4.69000 Max. :99.960 Max. :288.970   
##   
## PPA CrsPA Prog Playing Time MP  
## Min. :-0.100 Min. :-0.1000 Min. : -0.100 Min. : 0.90   
## 1st Qu.: 0.090 1st Qu.: 0.0000 1st Qu.: 1.400 1st Qu.: 2.99   
## Median : 0.580 Median : 0.0800 Median : 2.790 Median :14.93   
## Mean : 1.777 Mean : 0.4614 Mean : 7.079 Mean :17.25   
## 3rd Qu.: 1.170 3rd Qu.: 0.3100 3rd Qu.: 4.310 3rd Qu.:31.93   
## Max. :97.080 Max. :32.0000 Max. :290.100 Max. :38.05   
## NA's :5147   
## Playing Time Starts Playing Time Min Playing Time 90s Performance GA   
## Min. :-0.090 Min. : 3.08 Min. :-0.01 Min. :-0.080   
## 1st Qu.: 2.962 1st Qu.: 269.97 1st Qu.: 2.98 1st Qu.: 1.070   
## Median :14.430 Median :1336.95 Median :14.78 Median : 1.430   
## Mean :17.065 Mean :1534.27 Mean :17.05 Mean : 1.530   
## 3rd Qu.:31.960 3rd Qu.:2879.95 3rd Qu.:31.96 3rd Qu.: 1.835   
## Max. :38.100 Max. :3420.10 Max. :38.10 Max. :10.060   
## NA's :5147 NA's :5147 NA's :5147 NA's :5147   
## Performance GA90 Performance SoTA Performance Saves Performance Save%  
## Min. :-0.090 Min. :-0.080 Min. :-0.090 Min. : -0.03   
## 1st Qu.: 1.070 1st Qu.: 3.292 1st Qu.: 2.190 1st Qu.: 63.05   
## Median : 1.415 Median : 4.070 Median : 2.725 Median : 68.61   
## Mean : 1.527 Mean : 4.136 Mean : 2.788 Mean : 67.51   
## 3rd Qu.: 1.842 3rd Qu.: 4.912 3rd Qu.: 3.345 3rd Qu.: 74.31   
## Max. : 8.230 Max. :10.080 Max. :10.070 Max. :100.09   
## NA's :5147 NA's :5147 NA's :5147 NA's :5152   
## W D L Performance CS   
## Min. :-0.100 Min. :-0.100 Min. :-0.090 Min. :-0.100   
## 1st Qu.: 0.100 1st Qu.: 0.080 1st Qu.: 0.200 1st Qu.: 0.070   
## Median : 0.310 Median : 0.220 Median : 0.375 Median : 0.220   
## Mean : 0.326 Mean : 0.239 Mean : 0.419 Mean : 0.247   
## 3rd Qu.: 0.480 3rd Qu.: 0.340 3rd Qu.: 0.530 3rd Qu.: 0.330   
## Max. : 1.100 Max. : 1.080 Max. :10.040 Max. : 9.940   
## NA's :5147 NA's :5147 NA's :5147 NA's :5147   
## Performance CS% Performance PKatt.y Penalty Kicks PKA Penalty Kicks PKsv  
## Min. : -0.10 Min. :-0.100 Min. :-0.100 Min. :-0.10   
## 1st Qu.: 10.08 1st Qu.: 0.048 1st Qu.: 0.030 1st Qu.:-0.04   
## Median : 23.16 Median : 0.150 Median : 0.110 Median : 0.02   
## Mean : 23.13 Mean : 0.222 Mean : 0.178 Mean : 0.03   
## 3rd Qu.: 33.26 3rd Qu.: 0.270 3rd Qu.: 0.220 3rd Qu.: 0.07   
## Max. :100.08 Max. : 5.060 Max. : 4.930 Max. : 1.09   
## NA's :5161 NA's :5147 NA's :5147 NA's :5147   
## Penalty Kicks PKm Penalty Kicks Save% Salary   
## Min. :-0.100 Min. : -0.10 Min. : 90000   
## 1st Qu.:-0.040 1st Qu.: 0.00 1st Qu.:12612500   
## Median : 0.010 Median : 0.10 Median :19505000   
## Mean : 0.014 Mean : 15.65 Mean :19308138   
## 3rd Qu.: 0.060 3rd Qu.: 25.00 3rd Qu.:26020000   
## Max. : 1.040 Max. :100.08 Max. :44330000   
## NA's :5147 NA's :5264 NA's :1

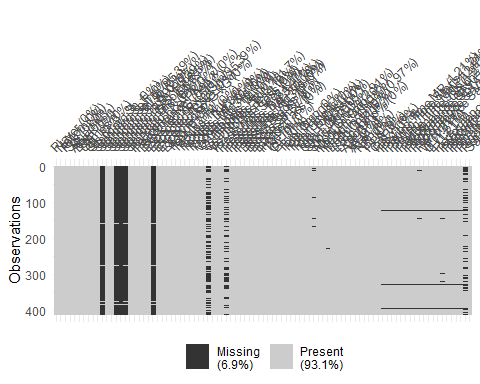
colSums(is.na(Data %>%  
 dplyr::filter(Pos!="GK")))

## Player Nation Pos Squad   
## 0 0 0 0   
## Age Born Games Gls   
## 0 0 0 0   
## Standard Sh Standard SoT Standard SoT% Standard Sh/90   
## 0 0 638 0   
## Standard SoT/90 Standard G/Sh Standard G/SoT Standard Dist   
## 0 638 1364 638   
## Standard FK Performance PK Performance PKatt.x Expected xG   
## 0 0 0 0   
## Expected npxG Expected npxG/Sh Expected G-xG Expected np:G-xG   
## 0 638 0 0   
## League Year Tackles Tkl Tackles TklW   
## 0 0 0 0   
## Tackles Def 3rd Tackles Mid 3rd Tackles Att 3rd Vs Dribbles Tkl   
## 0 0 0 0   
## Vs Dribbles Att Vs Dribbles Tkl% Vs Dribbles Past Pressures Press   
## 0 440 0 0   
## Pressures Succ Pressures % Pressures Def 3rd Pressures Mid 3rd   
## 0 53 0 0   
## Pressures Att 3rd Blocks Blocks Blocks Sh Blocks ShSv   
## 0 0 0 0   
## Blocks Pass Int Tkl+Int Clr   
## 0 0 0 0   
## Err Total Cmp Total Att Total Cmp%   
## 0 0 0 31   
## Total TotDist Total PrgDist Short Cmp Short Att   
## 0 0 0 0   
## Short Cmp% Medium Cmp Medium Att Medium Cmp%   
## 77 0 0 101   
## Long Cmp Long Att Long Cmp% Ast   
## 0 0 246 0   
## xA A-xA KP 1/3   
## 0 0 0 0   
## PPA CrsPA Prog Playing Time MP   
## 0 0 0 5142   
## Playing Time Starts Playing Time Min Playing Time 90s Performance GA   
## 5142 5142 5142 5142   
## Performance GA90 Performance SoTA Performance Saves Performance Save%   
## 5142 5142 5142 5142   
## W D L Performance CS   
## 5142 5142 5142 5142   
## Performance CS% Performance PKatt.y Penalty Kicks PKA Penalty Kicks PKsv   
## 5142 5142 5142 5142   
## Penalty Kicks PKm Penalty Kicks Save% Salary   
## 5142 5143 1

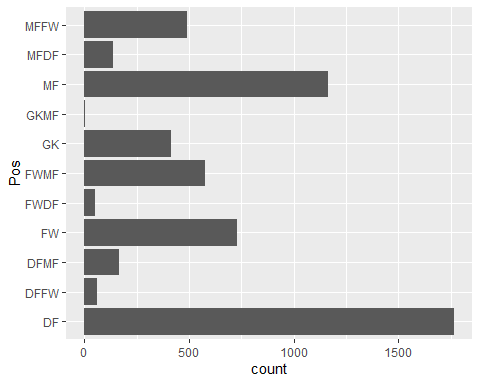
vis\_miss(Data %>%  
 dplyr::filter(Pos!="GK"))



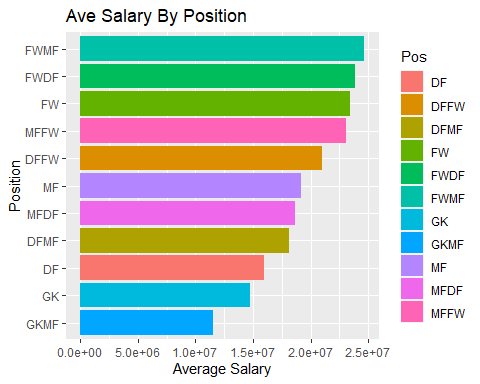
vis\_miss(Data %>%  
 dplyr::filter(Pos=="GK"))



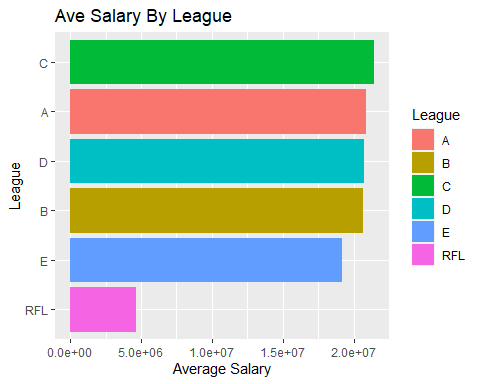
Data <- Data %>%  
 drop\_na(Salary)  
  
# Look at the different positions that exist  
Data %>%  
 ggplot(mapping = aes(x=Pos))+  
 geom\_bar(stat="count")+  
 labs(x="Pos")+  
 coord\_flip()



# Check Salary by Position  
  
Data %>%  
 group\_by(Pos) %>%  
 summarise(Ave\_Salary=mean(Salary)) %>%  
 ggplot(mapping = aes(x=reorder(Pos,Ave\_Salary),y=Ave\_Salary, fill=Pos))+  
 geom\_bar(stat = "identity")+  
 labs(x="Position",y="Average Salary",title = "Ave Salary By Position")+  
 coord\_flip()

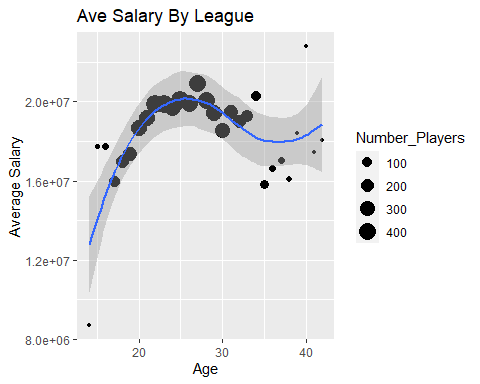


# Check Salary By League  
  
Data %>%  
 group\_by(League) %>%  
 summarise(Ave\_Salary=mean(Salary)) %>%  
 ggplot(mapping = aes(x=reorder(League,Ave\_Salary),y=Ave\_Salary, fill=League))+  
 geom\_bar(stat = "identity")+  
 labs(x="League",y="Average Salary",title = "Ave Salary By League")+  
 coord\_flip()

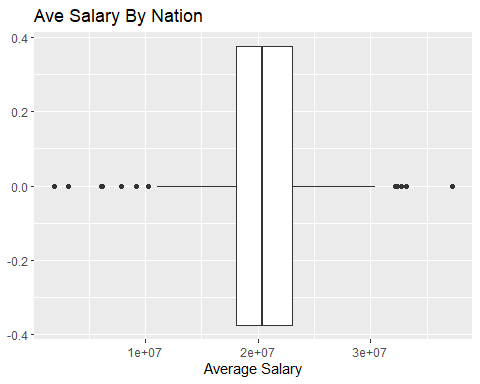


# Salary By Age  
Data %>%  
 group\_by(Age) %>%  
 summarise(Ave\_Salary=mean(Salary),  
 Number\_Players = n()) %>%  
 ggplot(mapping = aes(x=Age,y=Ave\_Salary))+  
 geom\_point(aes(size=Number\_Players))+  
 geom\_smooth()+  
 labs(x="Age",y="Average Salary",title = "Ave Salary By League")

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



## Salary By Nations  
Data %>%  
 group\_by(Nation) %>%  
 summarise(Ave\_Salary=mean(Salary)) %>%  
 ggplot(mapping = aes(x=Ave\_Salary))+  
 geom\_boxplot()+  
 labs(x="Average Salary",title = "Ave Salary By Nation")



#Ave Rarita Salary  
Data %>%  
 dplyr::filter(League!="RFL")%>%  
 group\_by(Nation) %>%  
 summarise(Ave\_Salary=mean(Salary)) %>%  
 dplyr::filter(Nation=="Rarita")

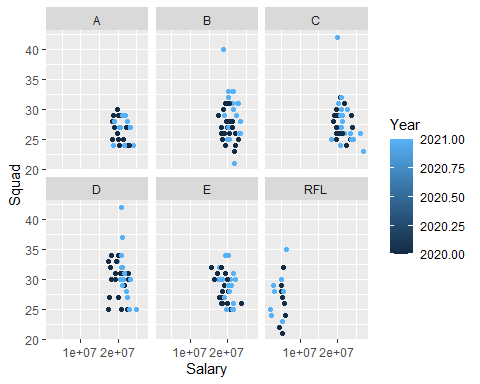
## # A tibble: 1 x 2  
## Nation Ave\_Salary  
## <chr> <dbl>  
## 1 Rarita 20455909.

Data %>%  
 dplyr::filter(League!="RFL")%>%  
 group\_by(Nation) %>%  
 summarise(Ave\_Salary=mean(Salary)) %>%  
 dplyr::filter(Nation=="Rarita")

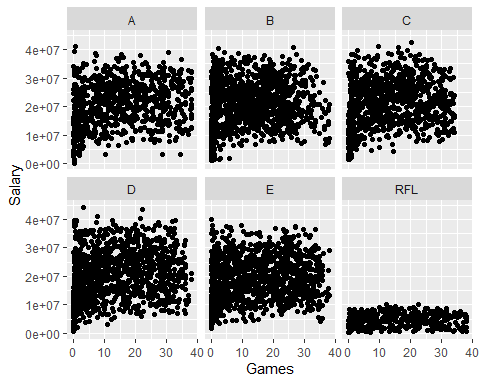
## # A tibble: 1 x 2  
## Nation Ave\_Salary  
## <chr> <dbl>  
## 1 Rarita 20455909.

## Salary By Team in Competition  
Data %>%   
 group\_by(Squad, League, Year) %>%  
 summarise(average\_salary = mean(Salary),  
 Squad\_Size = n()) %>%  
 ggplot(mapping = aes(x=Squad\_Size, y=average\_salary, color=Year)) +  
 geom\_point() +  
 labs(x = "Squad", y = "Salary") +  
 facet\_wrap(.~League) +  
 coord\_flip()

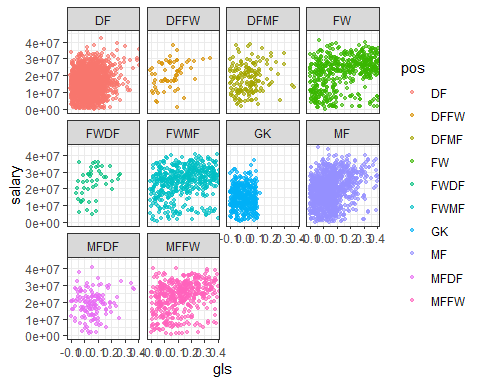
## `summarise()` has grouped output by 'Squad', 'League'. You can override using the `.groups` argument.



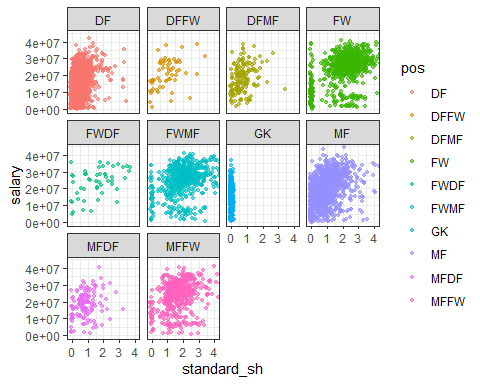
## Game Time and Salary  
  
Data %>%   
 ggplot(mapping = aes(x=Games, y=Salary)) +  
 geom\_point() +  
 labs(x = "Games", y = "Salary") +  
 facet\_wrap(.~League)



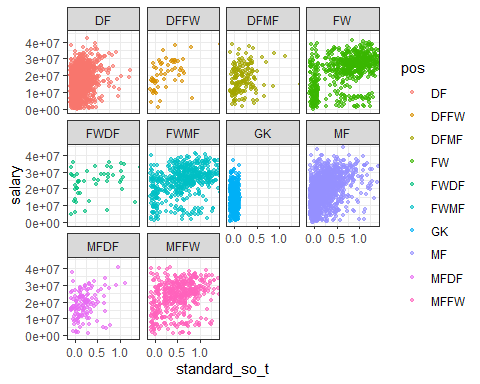
#No obvious link but Games played could be interpreted as exposure for other variables.  
  
Data <- Data[,c(seq(7),25,26,seq(8,24),seq(27,91))]  
  
theme\_set(  
 theme\_bw() +  
 theme(legend.position = "top")  
 )  
  
Data <- Data %>%  
 clean\_names()  
  
Quartile <- function(z){  
sapply(Data[,z], function(x) quantile(x, probs = seq(1/4, 3/4, 1/4), na.rm = TRUE))  
}  
  
  
# Comapare each variable to Salary by position  
for(i in 10:91) {  
 assign(paste0("Player\_Stat\_", i), Data %>%  
 dplyr::filter(pos != "GKMF") %>%  
 ggplot(aes\_string(x=colnames(Data[i]), y="salary", color="pos"))+  
 geom\_point(size = 1, alpha = 0.6, position = "jitter") +  
 facet\_wrap(~pos)+  
 theme\_bw()+  
 coord\_cartesian(xlim = c(NA,(Quartile(i)[3]\*2.5-Quartile(i)[1]))))  
}   
  
  
Player\_Stat\_10



Player\_Stat\_11

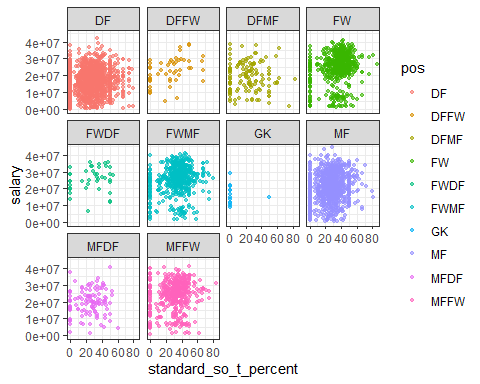


Player\_Stat\_12

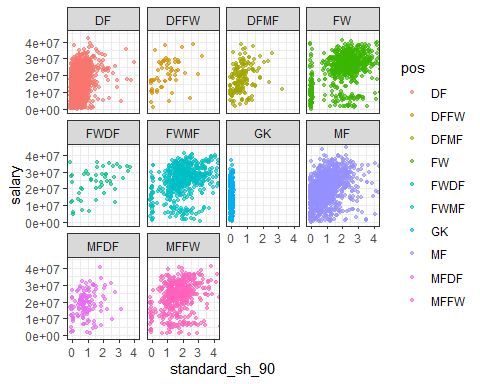


Player\_Stat\_13

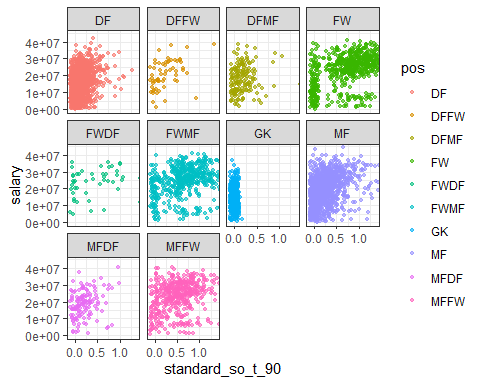
## Warning: Removed 1030 rows containing missing values (geom\_point).



Player\_Stat\_14

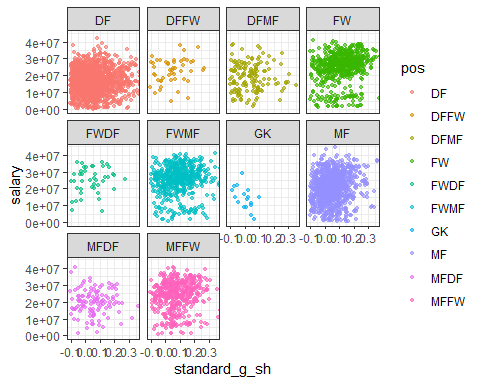


Player\_Stat\_15



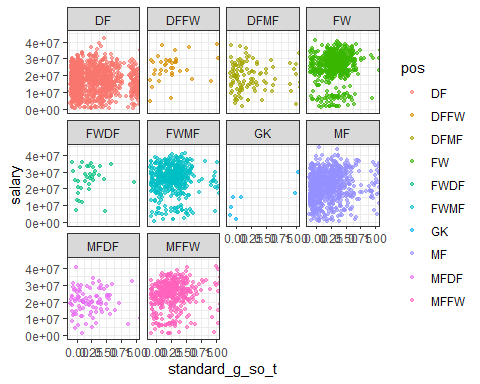
Player\_Stat\_16

## Warning: Removed 1030 rows containing missing values (geom\_point).



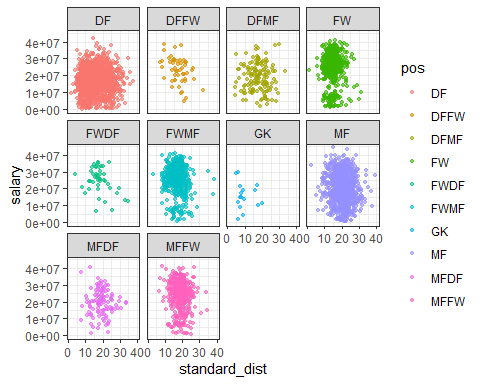
Player\_Stat\_17

## Warning: Removed 1768 rows containing missing values (geom\_point).

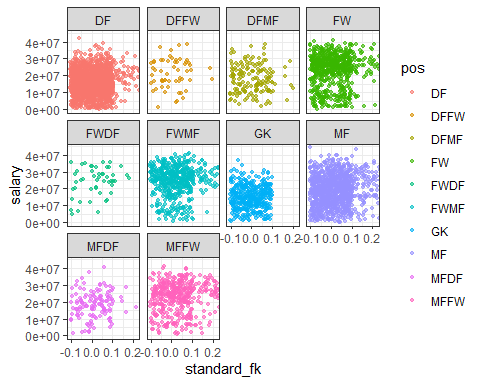


Player\_Stat\_18

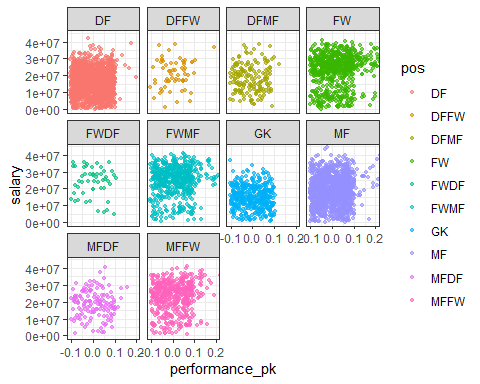
## Warning: Removed 1030 rows containing missing values (geom\_point).



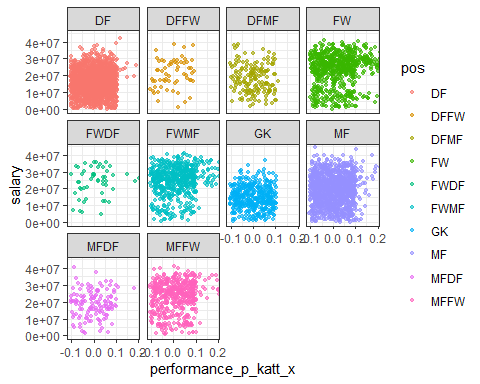
Player\_Stat\_19



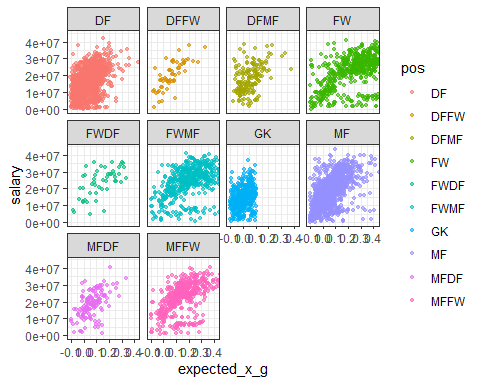
Player\_Stat\_20



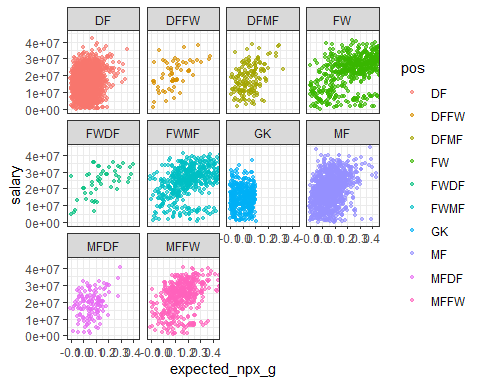
Player\_Stat\_21



Player\_Stat\_22

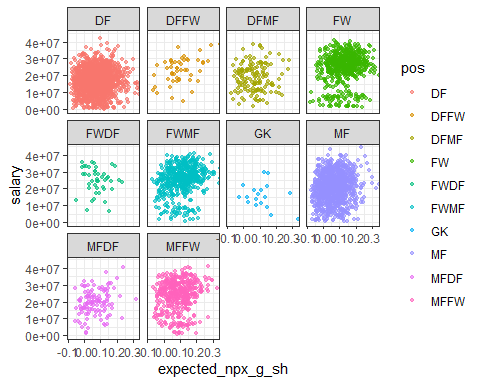


Player\_Stat\_23

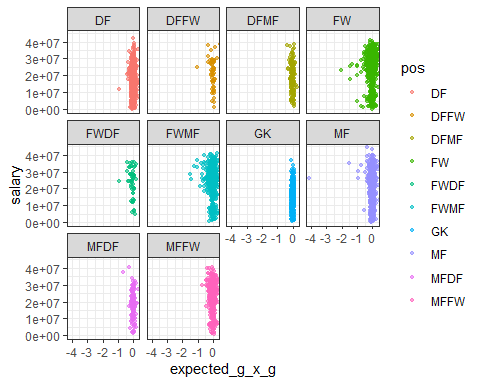


Player\_Stat\_24

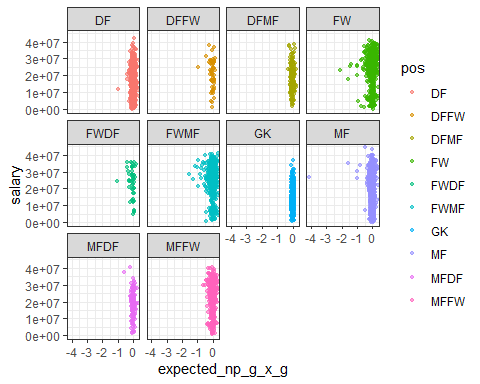
## Warning: Removed 1030 rows containing missing values (geom\_point).



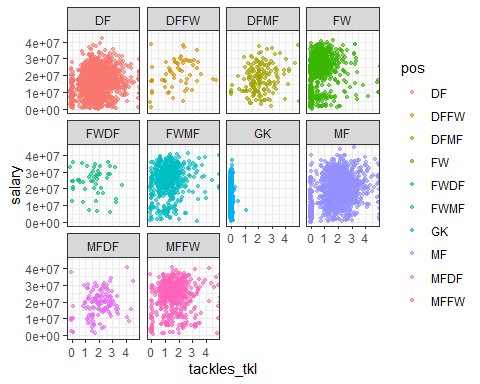
Player\_Stat\_25



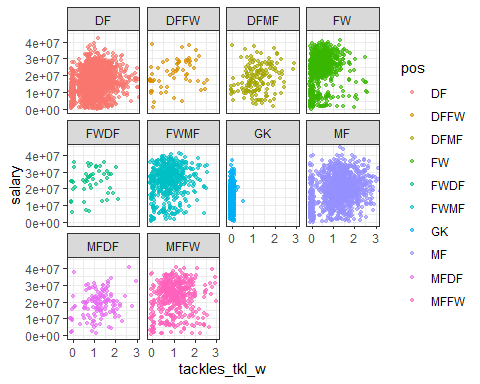
Player\_Stat\_26



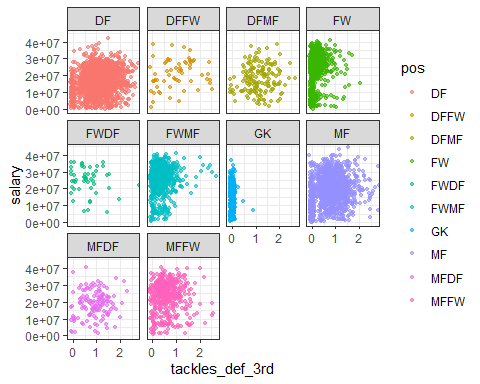
Player\_Stat\_27



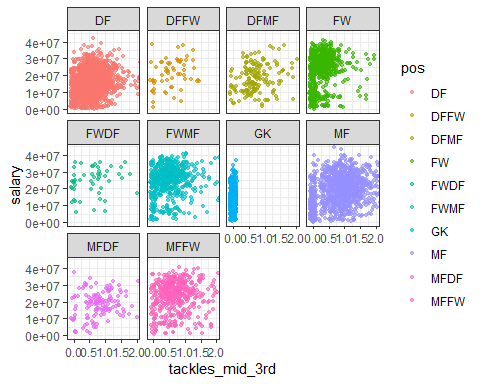
Player\_Stat\_28



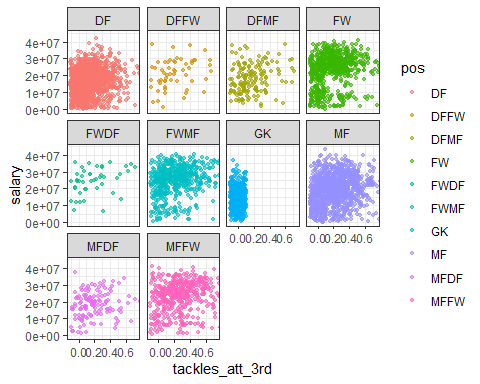
Player\_Stat\_29



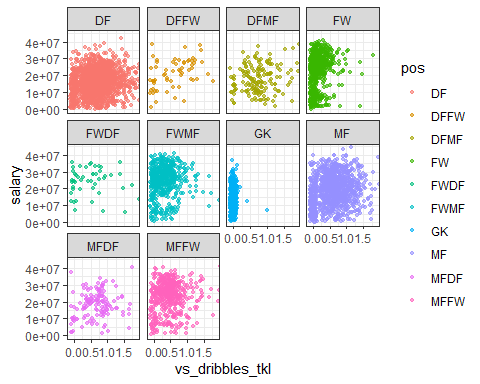
Player\_Stat\_30



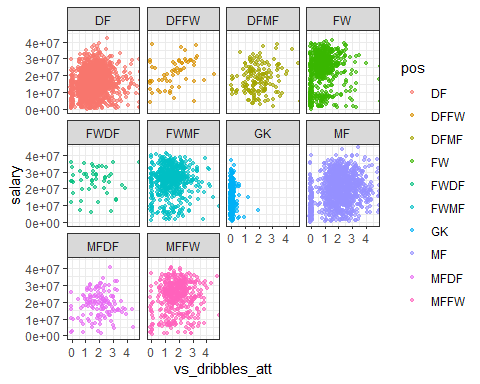
Player\_Stat\_31



Player\_Stat\_32

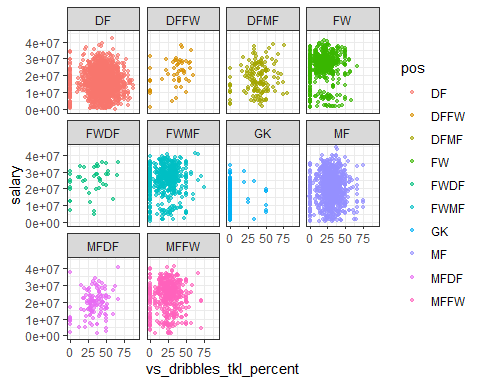


Player\_Stat\_33

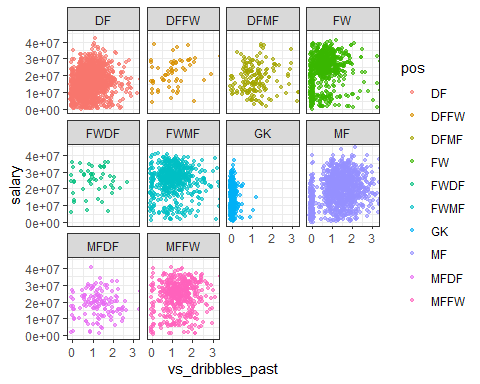


Player\_Stat\_34

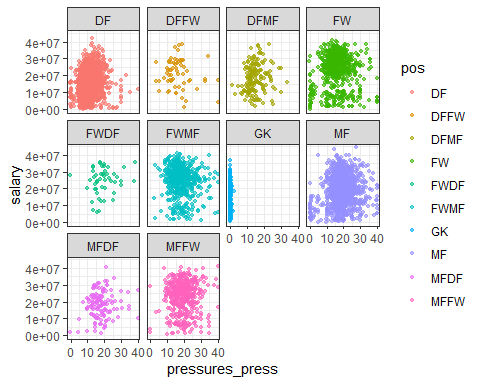
## Warning: Removed 652 rows containing missing values (geom\_point).



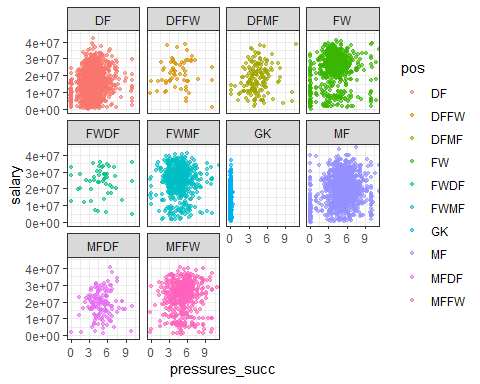
Player\_Stat\_35



Player\_Stat\_36

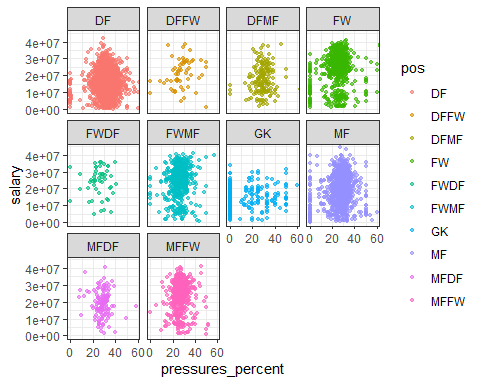


Player\_Stat\_37

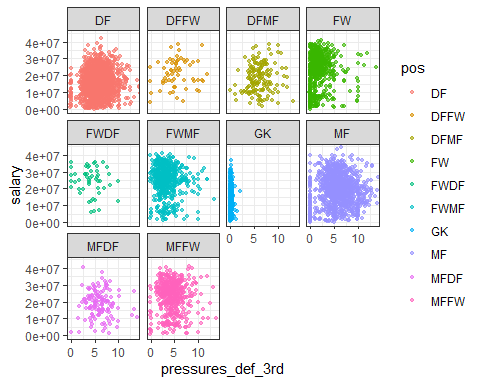


Player\_Stat\_38

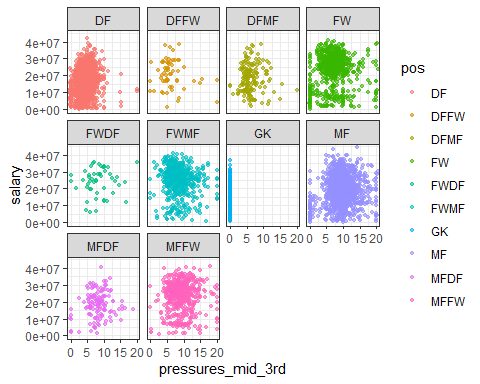
## Warning: Removed 195 rows containing missing values (geom\_point).



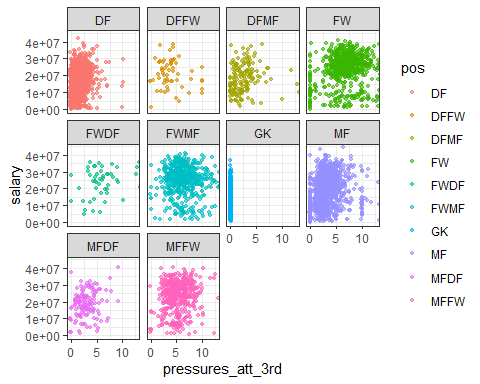
Player\_Stat\_39



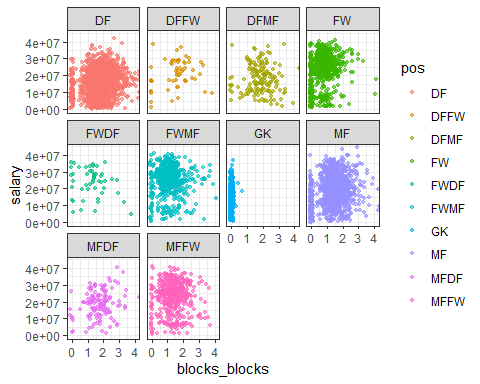
Player\_Stat\_40



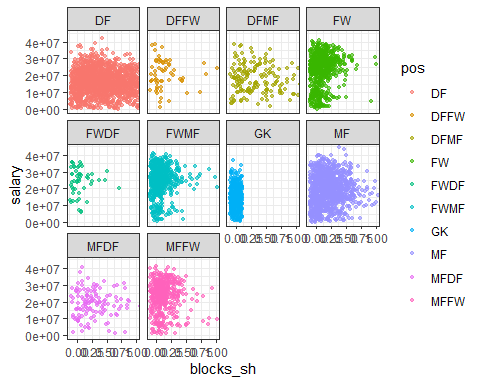
Player\_Stat\_41



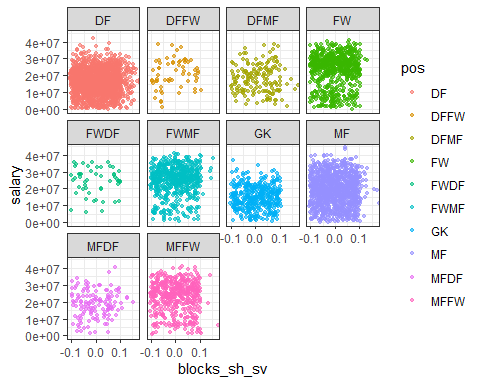
Player\_Stat\_42



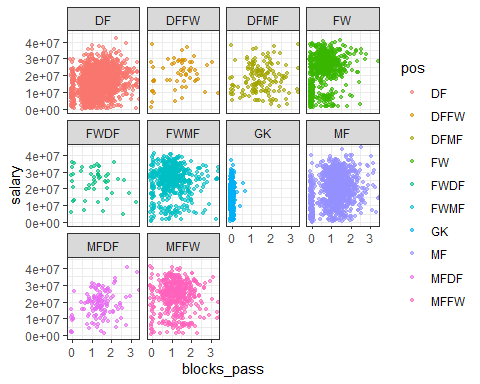
Player\_Stat\_43



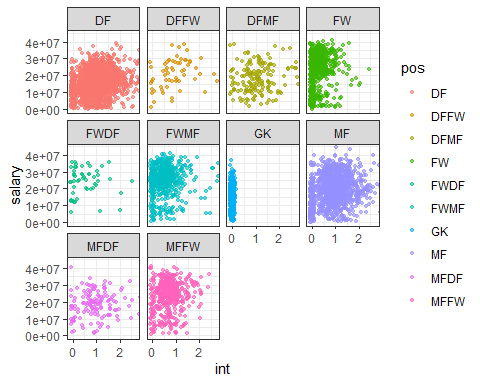
Player\_Stat\_44



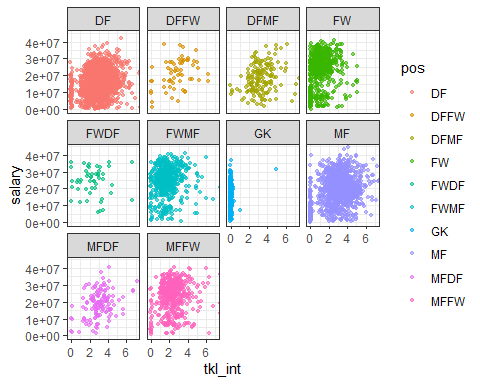
Player\_Stat\_45



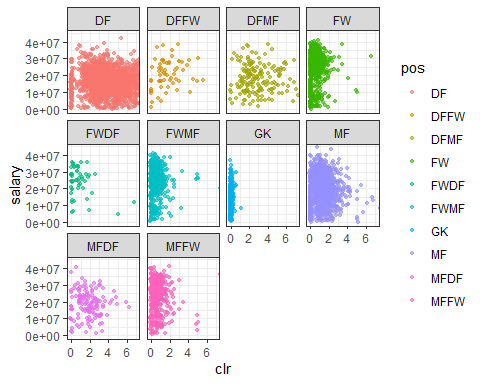
Player\_Stat\_46



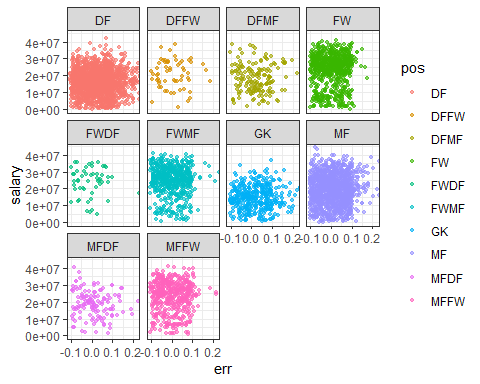
Player\_Stat\_47



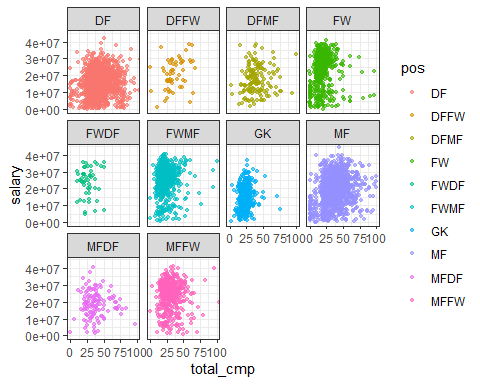
Player\_Stat\_48



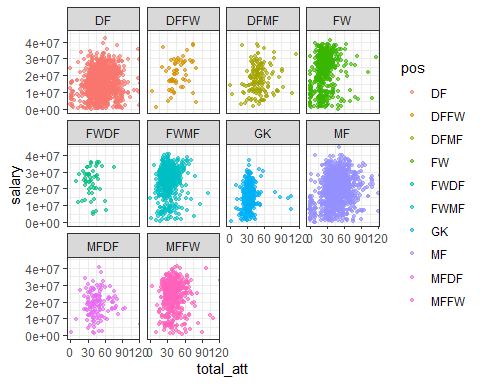
Player\_Stat\_49



Player\_Stat\_50

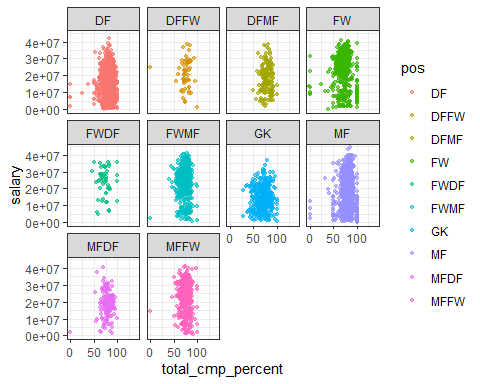


Player\_Stat\_51

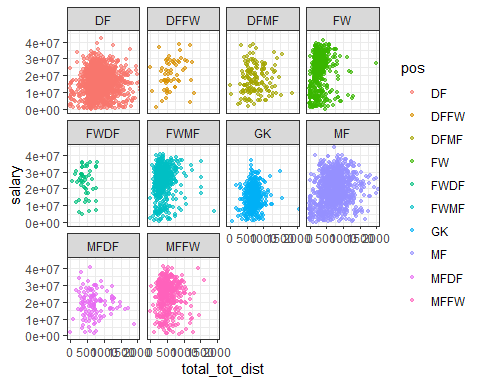


Player\_Stat\_52

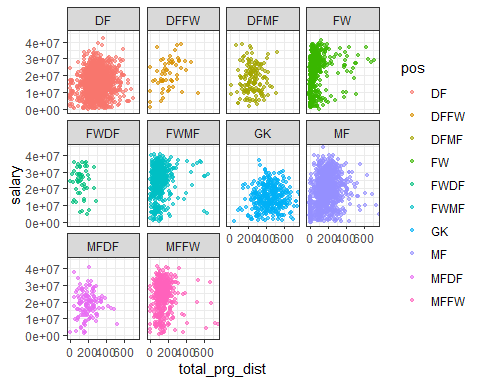
## Warning: Removed 31 rows containing missing values (geom\_point).



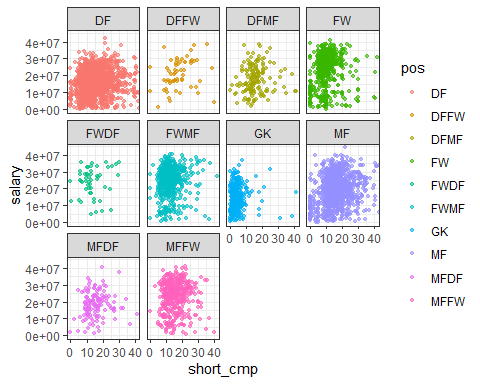
Player\_Stat\_53



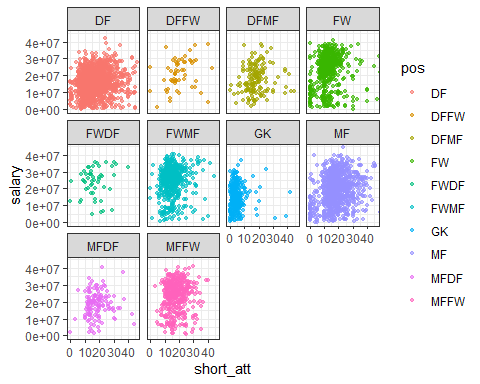
Player\_Stat\_54



Player\_Stat\_55

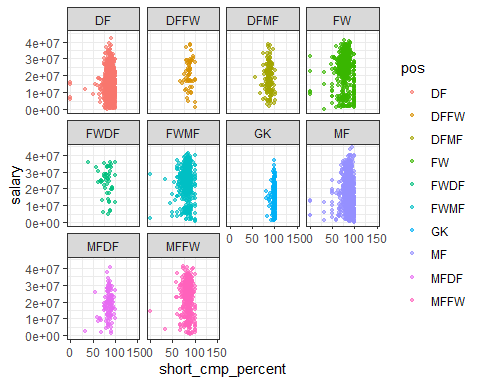


Player\_Stat\_56

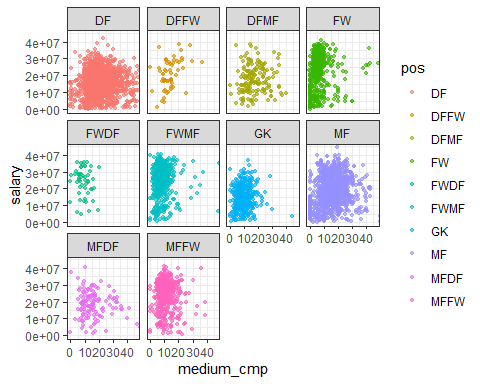


Player\_Stat\_57

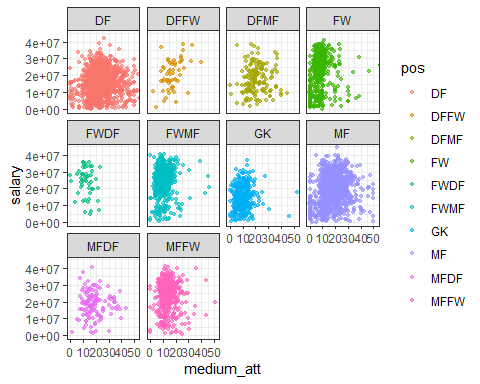
## Warning: Removed 89 rows containing missing values (geom\_point).



Player\_Stat\_58

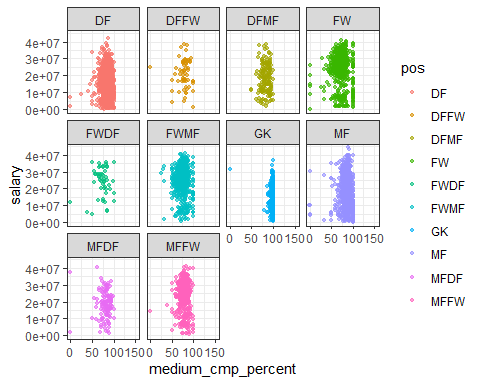


Player\_Stat\_59

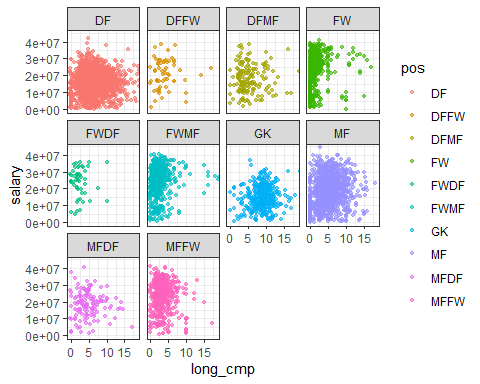


Player\_Stat\_60

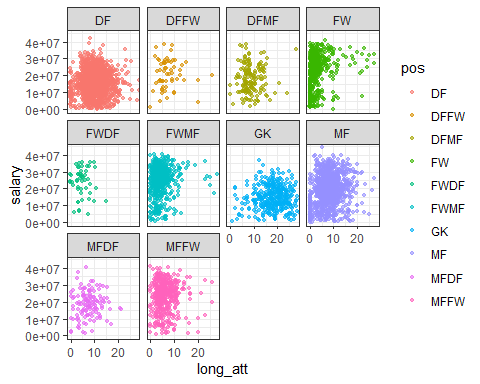
## Warning: Removed 105 rows containing missing values (geom\_point).



Player\_Stat\_61

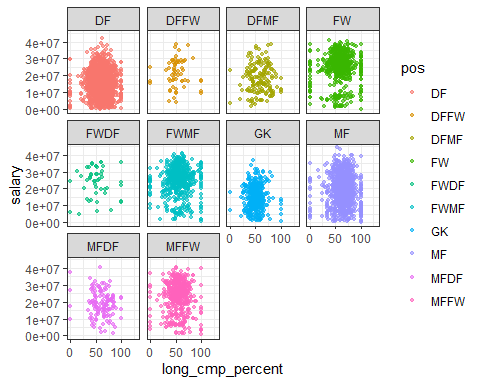


Player\_Stat\_62

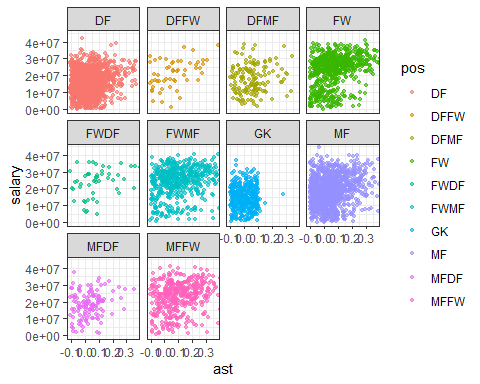


Player\_Stat\_63

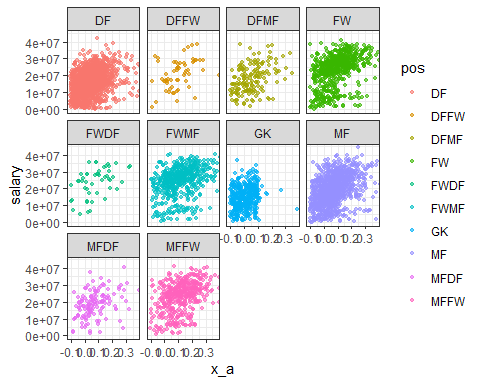
## Warning: Removed 246 rows containing missing values (geom\_point).



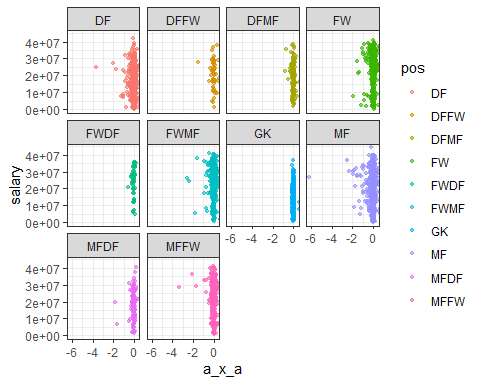
Player\_Stat\_64



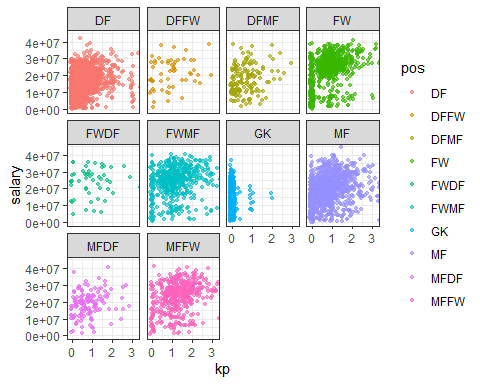
Player\_Stat\_65



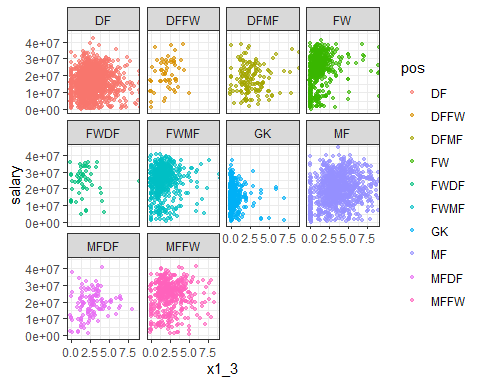
Player\_Stat\_66



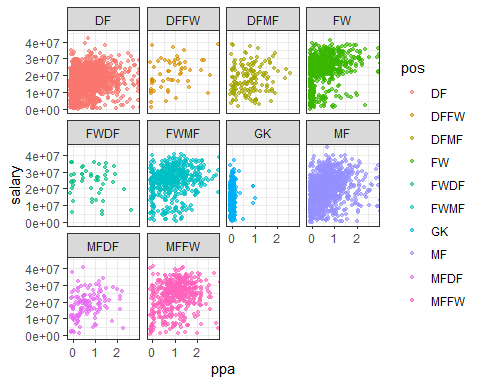
Player\_Stat\_67



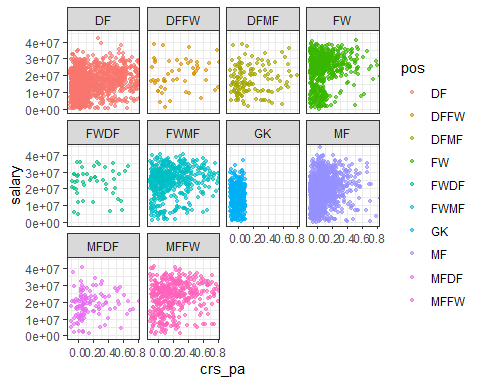
Player\_Stat\_68



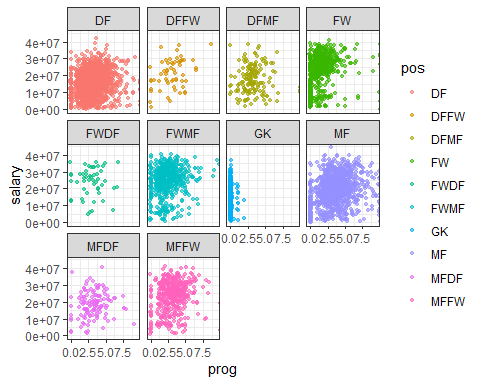
Player\_Stat\_69



Player\_Stat\_70

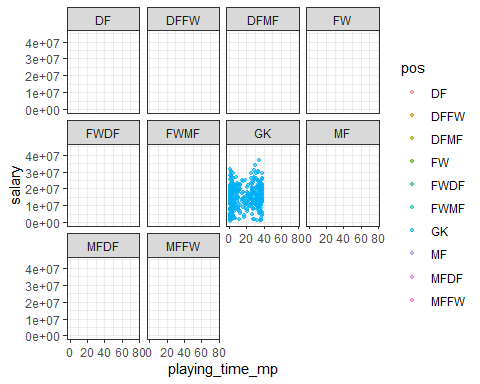


Player\_Stat\_71



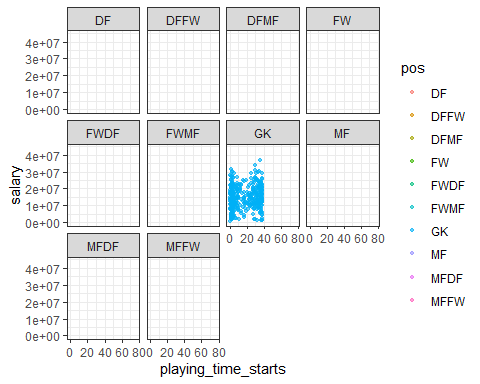
Player\_Stat\_72

## Warning: Removed 5146 rows containing missing values (geom\_point).



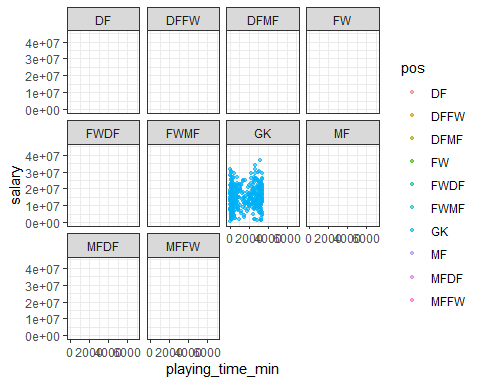
Player\_Stat\_73

## Warning: Removed 5146 rows containing missing values (geom\_point).



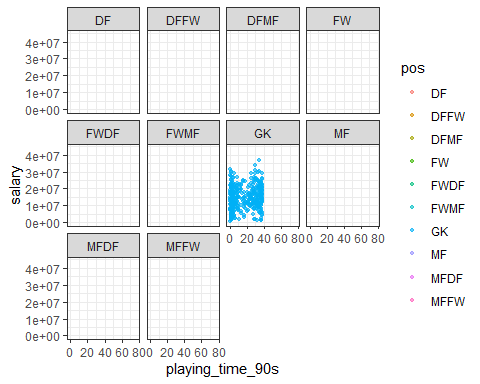
Player\_Stat\_74

## Warning: Removed 5146 rows containing missing values (geom\_point).



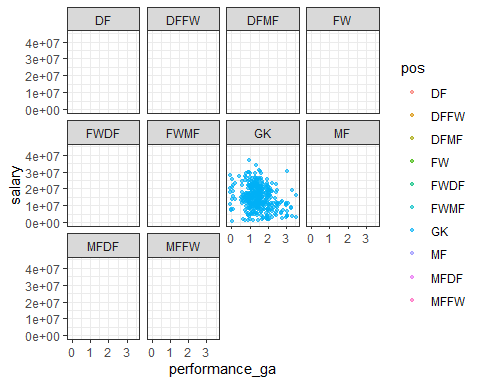
Player\_Stat\_75

## Warning: Removed 5146 rows containing missing values (geom\_point).



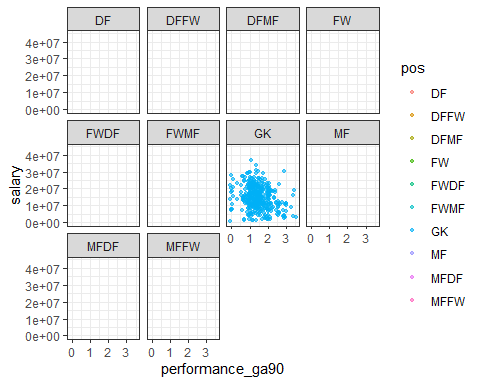
Player\_Stat\_76

## Warning: Removed 5146 rows containing missing values (geom\_point).



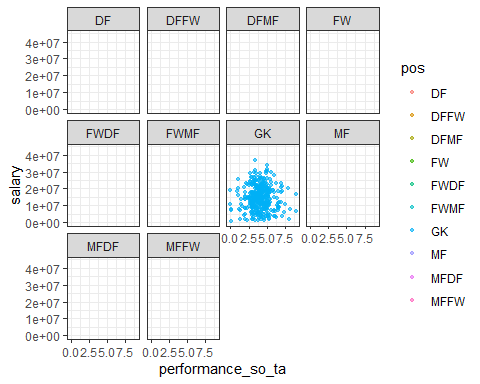
Player\_Stat\_77

## Warning: Removed 5146 rows containing missing values (geom\_point).



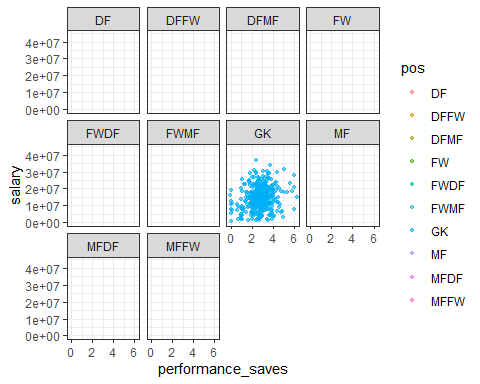
Player\_Stat\_78

## Warning: Removed 5146 rows containing missing values (geom\_point).



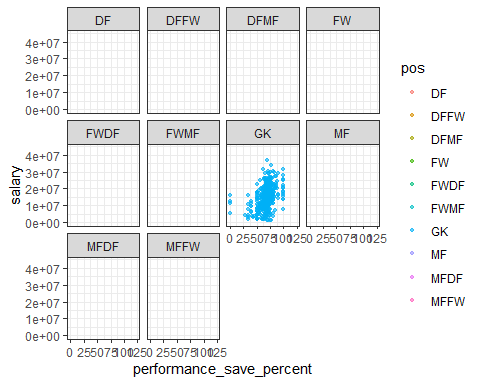
Player\_Stat\_79

## Warning: Removed 5146 rows containing missing values (geom\_point).



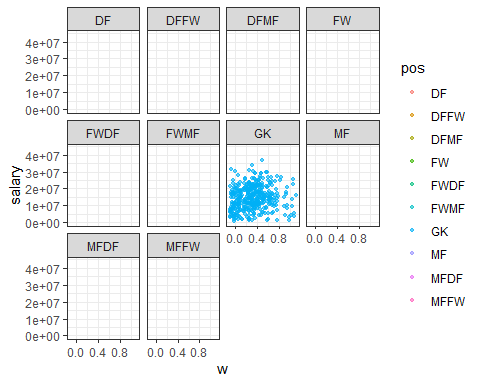
Player\_Stat\_80

## Warning: Removed 5151 rows containing missing values (geom\_point).



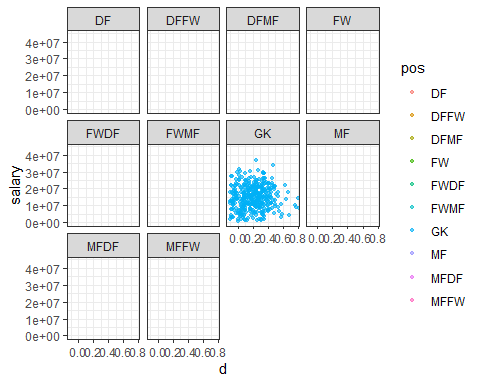
Player\_Stat\_81

## Warning: Removed 5146 rows containing missing values (geom\_point).



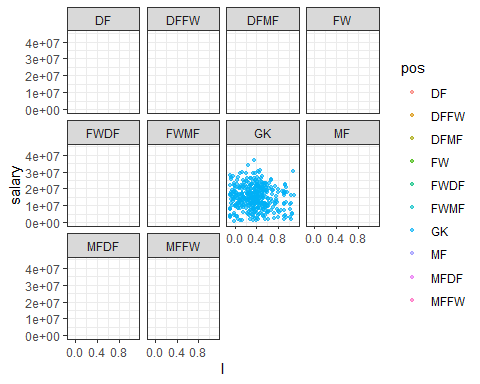
Player\_Stat\_82

## Warning: Removed 5146 rows containing missing values (geom\_point).



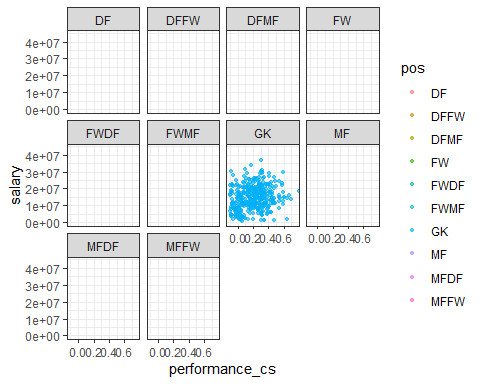
Player\_Stat\_83

## Warning: Removed 5146 rows containing missing values (geom\_point).



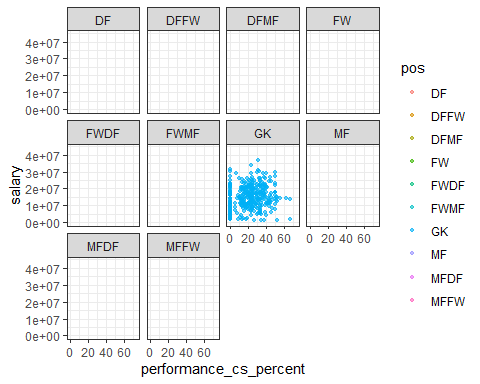
Player\_Stat\_84

## Warning: Removed 5146 rows containing missing values (geom\_point).



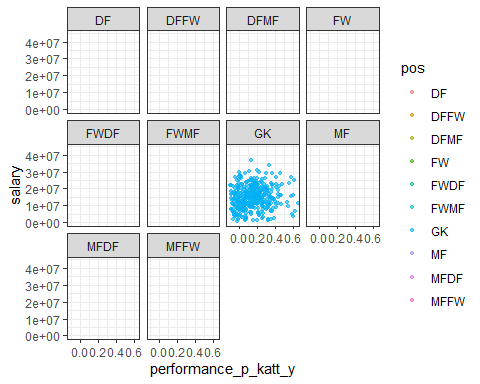
Player\_Stat\_85

## Warning: Removed 5160 rows containing missing values (geom\_point).



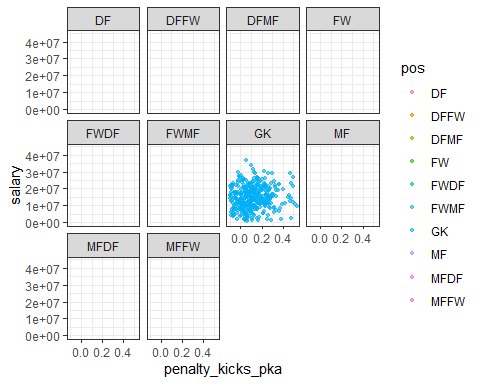
Player\_Stat\_86

## Warning: Removed 5146 rows containing missing values (geom\_point).



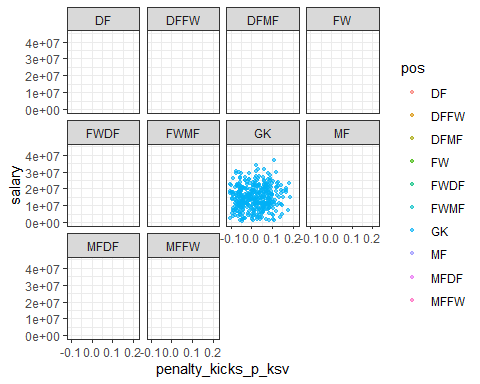
Player\_Stat\_87

## Warning: Removed 5146 rows containing missing values (geom\_point).



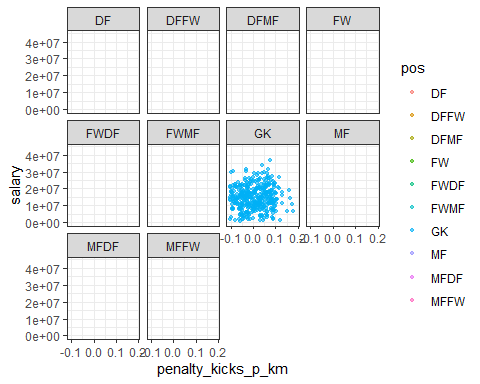
Player\_Stat\_88

## Warning: Removed 5146 rows containing missing values (geom\_point).



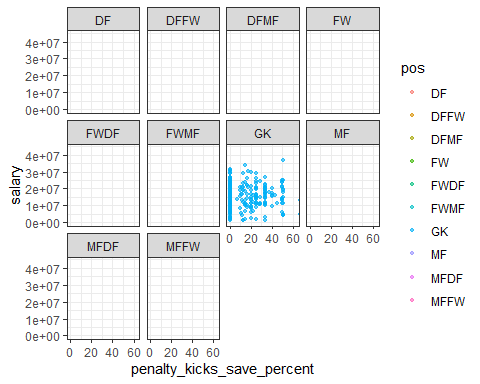
Player\_Stat\_89

## Warning: Removed 5146 rows containing missing values (geom\_point).

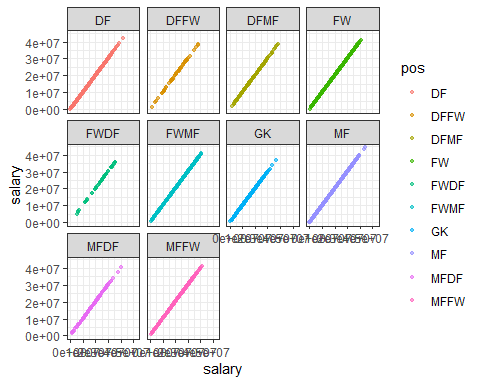


Player\_Stat\_90

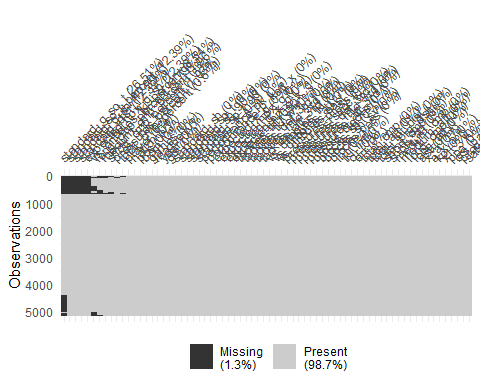
## Warning: Removed 5262 rows containing missing values (geom\_point).



Player\_Stat\_91

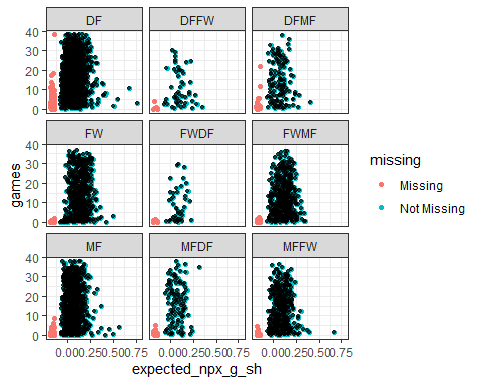


# Missing Data   
  
#Create data set that does not use goalkeeper and NAs are corrected for  
Data1 <- Data[,c(seq(3),6:71,91)] %>% #Remove variables associated with GK  
 dplyr::filter(!pos %in% c("GK","GKMF"))  
  
vis\_miss(Data1, cluster=TRUE, sort\_miss = TRUE) #Approximately 13% of observations have missing data



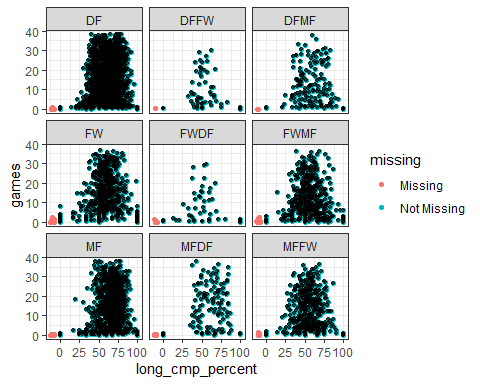
miss\_columns <- names(which(colSums(is.na(Data1))>0))  
  
Data\_miss <- Data1[miss\_columns]  
  
  
  
for(i in 1:11) {  
 assign(paste0("Missing\_", colnames(Data\_miss[i])), Data1 %>%  
 ggplot(aes\_string(x=colnames(Data\_miss[i]), y="games"))+  
 geom\_miss\_point()+  
 geom\_point(size = 1, alpha = 0.6, position = "jitter") +  
 facet\_wrap(~pos)+  
 theme\_bw())  
 }   
  
Missing\_expected\_npx\_g\_sh

## Warning: Removed 637 rows containing missing values (geom\_point).



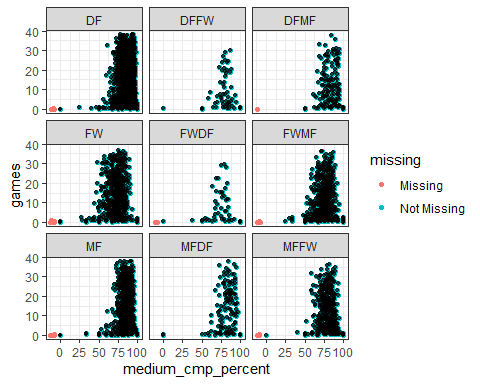
Missing\_long\_cmp\_percent

## Warning: Removed 246 rows containing missing values (geom\_point).



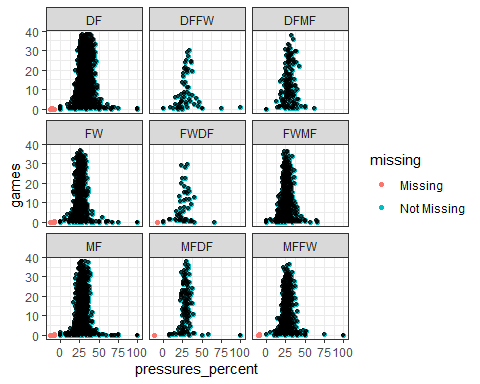
Missing\_medium\_cmp\_percent

## Warning: Removed 101 rows containing missing values (geom\_point).



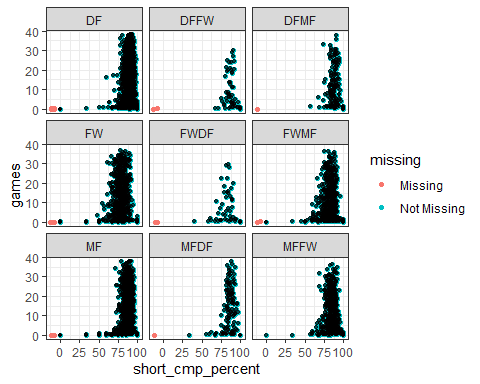
Missing\_pressures\_percent

## Warning: Removed 53 rows containing missing values (geom\_point).



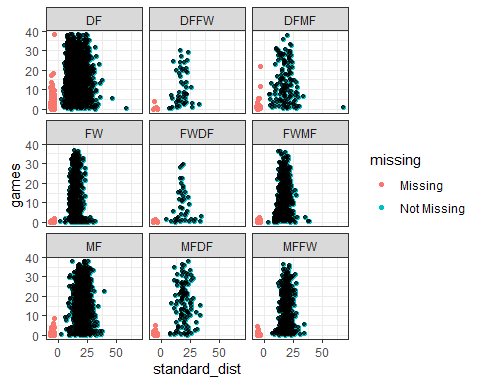
Missing\_short\_cmp\_percent

## Warning: Removed 77 rows containing missing values (geom\_point).



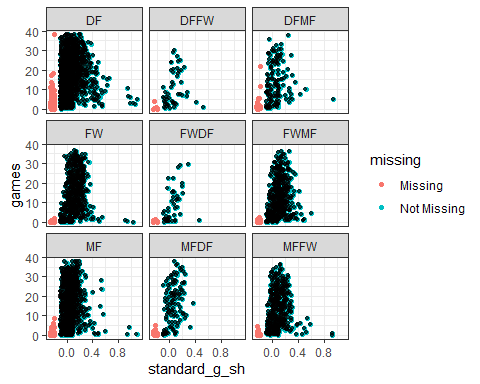
Missing\_standard\_dist

## Warning: Removed 637 rows containing missing values (geom\_point).



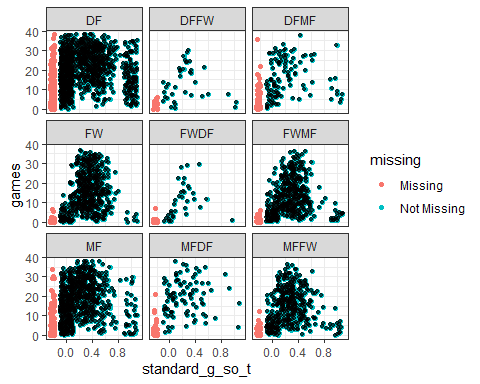
Missing\_standard\_g\_sh

## Warning: Removed 637 rows containing missing values (geom\_point).



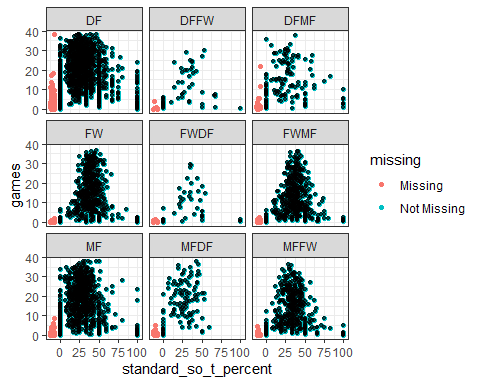
Missing\_standard\_g\_so\_t

## Warning: Removed 1363 rows containing missing values (geom\_point).



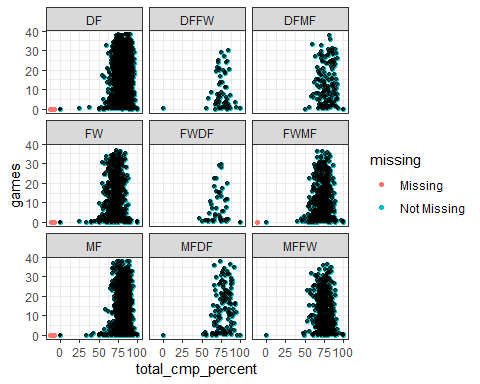
Missing\_standard\_so\_t\_percent

## Warning: Removed 637 rows containing missing values (geom\_point).



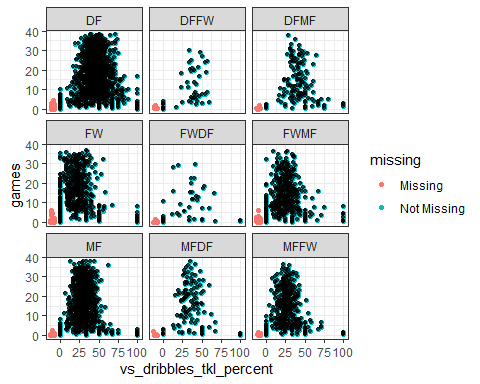
Missing\_total\_cmp\_percent

## Warning: Removed 31 rows containing missing values (geom\_point).



Missing\_vs\_dribbles\_tkl\_percent

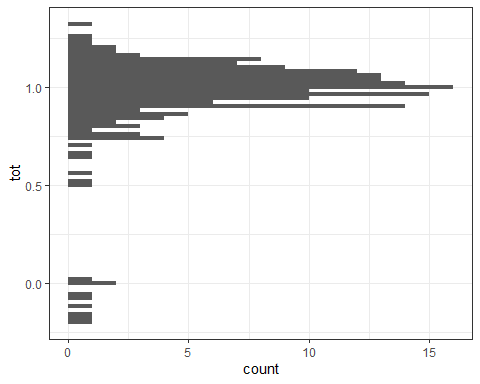
## Warning: Removed 439 rows containing missing values (geom\_point).



## Variable Adjustment

* Removes some variables and for reasons highlighted in the EDA allows may variables to be removed noting the use of a tree based method.
* Because we are using tree based method we can eliminate interaction terms (these terms also have large number of NAs) as the interaction will be captured by the tree. Therefore, in this case we would expect to have interaction depth greater than one if these interactions are a factor.

Variables\_remove <- Data1 %>%  
 dplyr::select(standard\_so\_t\_percent, standard\_sh\_90, standard\_so\_t\_90, standard\_g\_sh, standard\_g\_so\_t, expected\_npx\_g\_sh, expected\_np\_g\_x\_g, expected\_g\_x\_g, total\_cmp\_percent, short\_cmp\_percent, medium\_cmp\_percent, long\_cmp\_percent, a\_x\_a, vs\_dribbles\_tkl\_percent, vs\_dribbles\_att, pressures\_percent, tkl\_int)  
   
Data\_Simple <- Data1 %>%   
 dplyr::select(-c(colnames(Variables\_remove)))  
  
Data\_GK <- (Data %>%  
 dplyr::filter(pos==c("GK","GKMF")))[,c(1:6,8,9,72:91)] %>%  
 dplyr::select(-age) %>%  
 clean\_names()  
  
Data\_GK %>%  
 transmute(tot = w+l+d) %>%  
 na.omit() %>%  
 dplyr::filter(tot <= 2)%>%  
 ggplot(aes(y=tot))+  
 geom\_histogram(binwidth = 0.02)

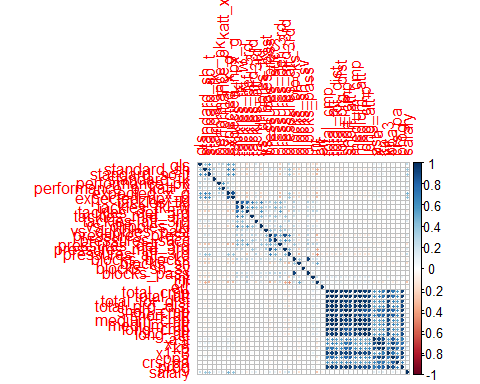


Data\_GK\_Simple <- Data\_GK %>%  
 dplyr::select(player, nation, born, league, year, playing\_time\_90s, performance\_ga, performance\_so\_ta, performance\_saves, w, d, l, performance\_cs, performance\_p\_katt\_y, penalty\_kicks\_pka, penalty\_kicks\_p\_ksv, penalty\_kicks\_p\_km, salary) %>%  
 na.omit() #Only removes 3 observations that had very little data.

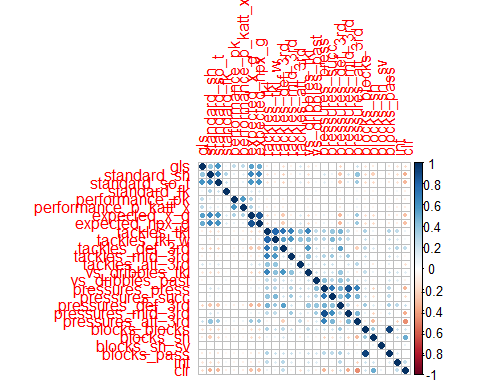
## Correlation

* A set of very highly correlated variables exist within passing therefore variables with a higher than 99% correlation have been removed.

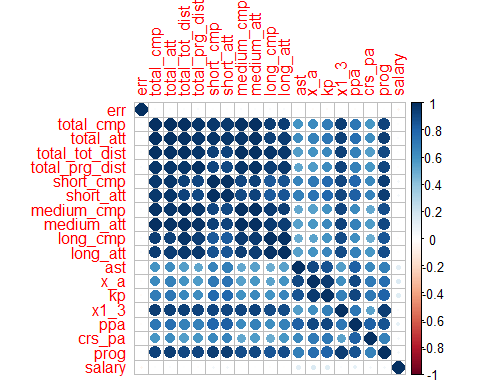
cor\_mat <- cor(Data\_Simple[,8:53] %>% dplyr::select(-standard\_dist), use = "pairwise.complete.obs")  
corrplot(cor\_mat) #Evidence of Highly correlated variables just get a list to know which to be aware of. However, The correlation is to be expected with > no. of shots likeley resulting in greater no. shots on target for an example



##Zoomed  
cor\_mat1 <- cor(Data\_Simple[,8:34] %>% dplyr::select(-standard\_dist), use = "pairwise.complete.obs")  
corrplot(cor\_mat1)



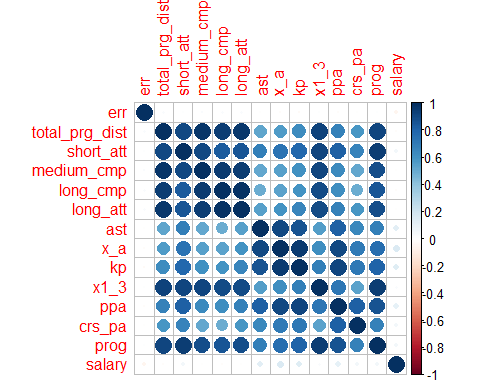
cor\_mat2 <- cor(Data\_Simple[35:53], use = "pairwise.complete.obs")  
corrplot(cor\_mat2)



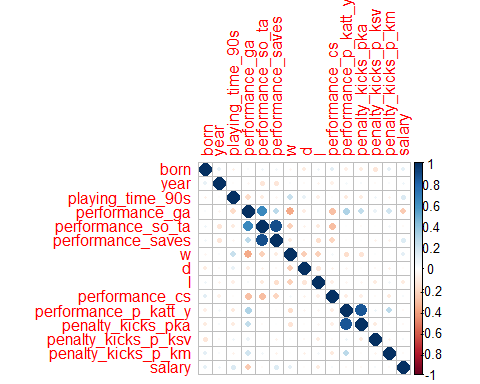
#Will remove correlations greater than 0.99  
  
high\_cor = findCorrelation(cor(Data\_Simple[,8:53] %>% dplyr::select(-standard\_dist)), cutoff=0.99)+8 # put any value as a "cutoff"   
colnames(Data\_Simple[,c(high\_cor)])

## [1] "total\_att" "total\_cmp" "total\_tot\_dist" "medium\_att"   
## [5] "short\_cmp"

reduced\_Data = Data\_Simple[,-c(high\_cor)]  
  
# Check new Correlation Matrix  
cor\_mat3 <- cor(reduced\_Data[35:48], use = "pairwise.complete.obs")  
corrplot(cor\_mat3)



# For Goalkeepers  
corrplot(cor(Data\_GK\_Simple %>%  
 dplyr::select(-player, -league, -nation), use = "pairwise.complete.obs")) #Nothing too bad



## Standardise and Imputation

* Standardised all the numerical variables.
* Imputing data for the standard dist from goal for shots as only variable with NAs that doesn’t have something to do with other variables.

#Use the shots from distance  
  
set.seed(55)  
reduced\_Data[,c(5,8:47)] <- scale(reduced\_Data[,c(5,8:47)])  
Rarita\_Data <- reduced\_Data %>% dplyr::filter(nation == "Rarita")  
Data2 <- reduced\_Data %>% dplyr::filter(nation != "Rarita") %>% dplyr::select(-nation)  
  
train\_index<-sample(nrow(Data2),4/5\*nrow(Data2))  
train<-Data2[train\_index,]  
test<-Data2[-train\_index,]  
#impute the missing values with KNN  
train2<-kNN(train,weightDist = FALSE,imp\_var = FALSE) %>% dplyr::select(-player)  
test2<-kNN(test,weightDist = FALSE,imp\_var = FALSE) %>% dplyr::select(-player)  
Rarita\_Data2 <- kNN(Rarita\_Data,weightDist = FALSE,imp\_var = FALSE)

## Data Splitting

#Split data  
  
# Non-Rarita set to fit and test model, Rarita set to predict Rarita player salary  
Data\_GK\_Simple[,6:17] <- scale(Data\_GK\_Simple[,6:17])  
GK\_Data\_Rarita <- Data\_GK\_Simple %>% dplyr::filter(nation == "Rarita")   
GK\_Data\_Other <- Data\_GK\_Simple %>% dplyr::filter(nation != "Rarita")  
  
#Only split non-Rarita data into test and training set  
set.seed(10)  
train\_set <- sample(1:nrow(GK\_Data\_Other),nrow(Data\_GK\_Simple)\*0.8)  
train3 <- dplyr::select(GK\_Data\_Other,-c(player,nation))[train\_set,]  
test3 <- dplyr::select(GK\_Data\_Other,-c(player,nation))[-train\_set,]

## Model Selection

* Using model to assign monetary value to players based on the values given to other players in the league.
* Using tree based models as they will automatically capture interaction between variables and can therefore remove variables which are just using interaction between two other variables.
* GBM was the model of choice with the following parameters selected for each model:
  + For players (not GK’s) interaction.depth = 7, n.trees = 950,shrinkage = 0.01,n.minobsinnode = 30
  + For GK’s interaction.depth = 1, n.trees = 7200, shrinkage = 0.001, n.minobsinnode =3

\*\* Note: skip to the final two model runs one for players and one for GKs to save time on the document.

gbmGrid <- expand.grid(interaction.depth = c(1,3,5,7),   
 n.trees = (1:25)\*200,   
 shrinkage = 0.001,  
 n.minobsinnode = 30)  
  
fitControl <- trainControl(## 10-fold CV  
 method = "repeatedcv",  
 number = 10,  
 repeats = 5)  
  
  
set.seed(846)  
gbmFit <- caret::train(salary ~ .,   
 data = train2,   
 method = "gbm",   
 trControl = fitControl,  
 verbose = FALSE,   
 tuneGrid = gbmGrid)  
  
plot(gbmFit, metric = "RMSE")  
  
predGBM <- predict(gbmFit, newdata=test2)  
  
gbmFit$bestTune  
  
RMSE\_GBM <- RMSE(predGBM, test2$salary)   
MAE\_GBM <- MAE(predGBM, test2$salary)   
  
gbmImp <- varImp(gbmFit, scale = TRUE)  
  
plot(gbmImp, top = 10)

gbmGrid2 <- expand.grid(interaction.depth = c(1,3,5,7),   
 n.trees = (1:30)\*50,   
 shrinkage = 0.01,  
 n.minobsinnode = 30)  
  
fitControl <- trainControl(## 10-fold CV  
 method = "repeatedcv",  
 number = 10,  
 repeats = 5)  
  
  
set.seed(26)  
gbmFit2 <- caret::train(salary ~ .,   
 data = train2,   
 method = "gbm",   
 trControl = fitControl,  
 verbose = FALSE,   
 tuneGrid = gbmGrid2)  
  
plot(gbmFit2, metric = "RMSE")  
  
predGBM2 <- predict(gbmFit2, newdata=test2)  
  
gbmFit2$bestTune  
gbmFit2$results  
  
RMSE\_GBM2 <- RMSE(predGBM2, test2$salary)   
MAE\_GBM2 <- MAE(predGBM2, test2$salary)   
  
gbmImp2 <- varImp(gbmFit2, scale = TRUE)  
  
plot(gbmImp2, top = 10)

ctrl <- trainControl(## 10-fold CV  
 method = "repeatedcv",  
 number = 10,  
 repeats = 5)  
  
  
rfGrid <- expand.grid(mtry = (2:15)\*2)   
  
set.seed(24)  
  
RF\_Fit <- caret::train(salary ~.,  
 data = train2,  
 method = "rf",  
 trControl = ctrl,  
 tuneGrid = rfGrid,  
 n.trees = 5000)  
  
plot(RF\_Fit, metric = "RMSE")  
  
predRF <- predict(RF\_Fit, newdata=test2)  
  
RF\_Fit$bestTune  
  
RMSE\_RF <- RMSE(predRF, test2$salary)   
MAE\_RF <- MAE(predRF, test2$salary)   
  
RF\_Imp <- varImp(RF\_Fit, scale = TRUE)  
  
plot(RF\_Imp, top = 15)

\*Final Model Players

Data\_f <- rbind(train2, test2)  
  
gbmGrid3 <- expand.grid(interaction.depth = 7,   
 n.trees = 950,   
 shrinkage = 0.01,  
 n.minobsinnode = 30)  
  
  
set.seed(29)  
gbmFitF <- caret::train(salary ~ .,   
 data = Data\_f,   
 method = "gbm",  
 verbose = FALSE,   
 tuneGrid = gbmGrid3)

## GK Models

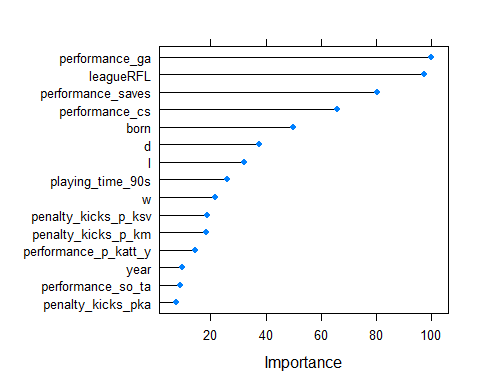
gbmGridGK <- expand.grid(interaction.depth = c(1,3,5,7),   
 n.trees = (1:30)\*35,   
 shrinkage = 0.01,  
 n.minobsinnode = c(2,4,6,8))  
  
fitControl <- trainControl(## 10-fold CV  
 method = "repeatedcv",  
 number = 10,  
 repeats = 5)  
  
  
set.seed(26)  
gbmFitGK <- caret::train(salary ~ .,   
 data = train3,   
 method = "gbm",   
 trControl = fitControl,  
 verbose = FALSE,   
 tuneGrid = gbmGridGK)  
  
plot(gbmFitGK, metric = "RMSE")  
  
GK\_Imp <- varImp(gbmFitGK, scale = TRUE)  
  
plot(GK\_Imp, top = 15)  
  
predGBM\_GK <- predict(gbmFitGK, newdata=test3)  
RMSE(predGBM\_GK, test3$salary)  
  
gbmFitGK$bestTune

gbmGridGK2 <- expand.grid(interaction.depth = c(1,3),   
 n.trees = (1:50)\*150,   
 shrinkage = 0.001,  
 n.minobsinnode = c(2,3,4))  
  
fitControl <- trainControl(## 10-fold CV  
 method = "repeatedcv",  
 number = 10,  
 repeats = 5)  
  
  
set.seed(26)  
gbmFitGK2 <- caret::train(salary ~ .,   
 data = train3,   
 method = "gbm",   
 trControl = fitControl,  
 verbose = FALSE,   
 tuneGrid = gbmGridGK2)  
  
plot(gbmFitGK2, metric = "RMSE")  
  
GK\_Imp2 <- varImp(gbmFitGK2, scale = TRUE)  
  
plot(GK\_Imp2, top = 15)  
  
predGBM\_GK2 <- predict(gbmFitGK2, newdata=test3)  
RMSE(predGBM\_GK2, test3$salary)  
  
gbmFitGK2$bestTune

ctrl <- trainControl(## 10-fold CV  
 method = "repeatedcv",  
 number = 10,  
 repeats = 5)  
  
  
rfGrid\_GK <- expand.grid(mtry = (2:10))   
  
set.seed(24)  
  
RF\_FitGK <- caret::train(salary ~.,  
 data = train3,  
 method = "rf",  
 trControl = ctrl,  
 tuneGrid = rfGrid\_GK,  
 n.trees = 4000)  
  
plot(RF\_FitGK, metric = "RMSE")  
  
predRF\_GK <- predict(RF\_FitGK, newdata=test3)  
RMSE(predRF\_GK, test3$salary)  
  
RF\_FitGK$bestTune  
  
  
RF\_Imp\_GK <- varImp(RF\_FitGK, scale = TRUE)  
  
plot(RF\_Imp\_GK, top = 15)

\*Final Model GK

#GBM optimal model  
gbmGridF\_GK <- expand.grid(interaction.depth = 1,   
 n.trees = 7200,   
 shrinkage = 0.001,  
 n.minobsinnode =3)  
  
  
set.seed(10)  
gbmFitF\_GK <- caret::train(salary ~ .,   
 data = rbind(test3, train3),   
 method = "gbm",   
 verbose = FALSE,   
 tuneGrid = gbmGridF\_GK,  
 metric = "RMSE")  
  
GK\_Imp <- varImp(gbmFitF\_GK, scale = TRUE)  
  
plot(GK\_Imp, top = 15)



## Tournament Edit Data

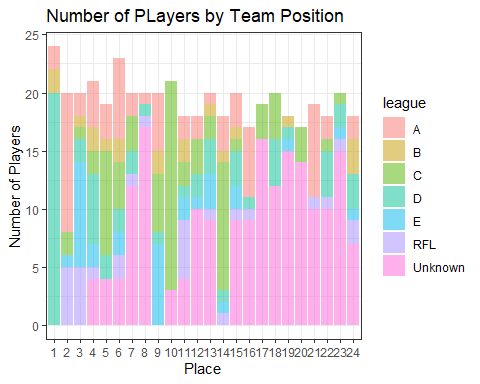
* Only using 2021 data most is missing from 2020 from what is there the data seems very unrelaible. (Age of GK’s is imposible given year and year of birth)

Tourn\_Data <- Tourn\_FW %>%  
 left\_join(Tourn\_MF, by = c("player", "nation", "pos", "age", "born", "year", "league")) %>%  
 left\_join(Tourn\_DF, by = c("player", "nation", "pos", "age", "born", "year", "league")) %>%  
 rename(performance\_p\_katt\_x = "performance\_p\_katt") %>%  
 dplyr::select(-league)%>%  
 left\_join(Data %>% dplyr::select(player, year, league), by = c("player", "year")) %>%  
 dplyr::filter(!pos %in% c("GK","GKMF"))  
  
Tourn\_Data2 <- Tourn\_Data %>%  
 dplyr::select(colnames(test2 %>% dplyr::select(-salary))) %>%  
 mutate(league = replace\_na(league, "RFL")) #This may seem like an odd assumption but it seems that those with data missing for league they regularly play, likely play in another local league like the RFL (not covered in the player data) unfortunately a standard imputation algorithm won't detect this hence the importance of using the league to determine salary in the first place.  
  
Tourn\_Data2<-kNN(Tourn\_Data2, weightDist = FALSE,imp\_var = FALSE) #This adds adds the standard\_dist data.  
  
Tourn\_Data2[,c(3,6:45)] <- scale(Tourn\_Data2[,c(3,6:45)])  
  
## For GK  
  
Tourn\_GK2 <- Tourn\_GK %>%  
 dplyr::select(colnames(Data\_GK\_Simple %>% dplyr::select(-salary,-league))) %>%  
 left\_join((Data\_GK\_Simple %>% dplyr::select(player, year, league)), by = c("player", "year")) %>%   
 mutate(league = replace\_na(league, "RFL"))  
   
  
Tourn\_GK2[,5:16] <- scale(Tourn\_GK2[,5:16])

## Tournament Prediction

* Find the salaries using the model determined salary for each of the players.
* Many players are have unknown league but assumed to be “RFL” (assumed equivalent to another local league)
  + This is largely assumed because we don’t have visibility of other local leagues in the league data.
  + The average Rarita player in the league data had a much lower average salary but if we remove all players that compete in the RFL from data it is above average (meaning the initial descrepency is likely due to the fact we don’t have visibility of other local leagues).

predTourn <- predict(gbmFitF, newdata=Tourn\_Data2)  
  
Tourn\_Data3 <- cbind(Tourn\_Data,predTourn) %>%  
 rename(model\_salary = "predTourn") %>%  
 separate(pos, c("pos1", "pos2"), sep=2)%>%  
 left\_join(Tourn\_Result, by="nation") %>%  
 dplyr::select(player, nation, pos1, pos2, league, model\_salary, place)  
  
  
Tourn\_Data3 %>%  
 mutate(league=replace\_na(league, "Unknown")) %>%  
 group\_by(place, league)%>%  
 count()%>%  
 ggplot(aes(x=as.factor(place), y=n, fill=league)) +  
 geom\_bar(position = "stack", alpha = 0.5, stat = "identity")+  
 theme\_bw() +  
 labs(x="Place",y="Number of Players",title = "Number of PLayers by Team Position")



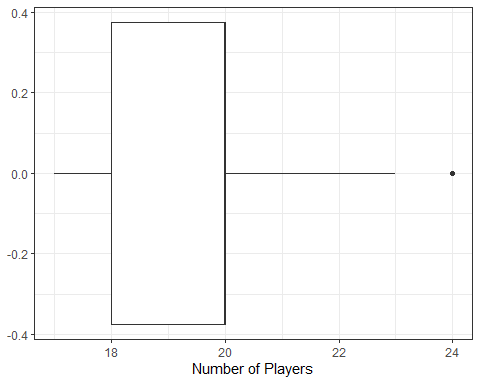
export\_tourn <- Tourn\_Data3 %>%  
 dplyr::select(player, model\_salary)  
  
writexl::write\_xlsx(export\_tourn, "Touranment\_Salary.xlsx")  
  
### Salary  
  
model\_salary <- predict(gbmFitF\_GK, newdata=(Tourn\_GK2 %>% dplyr::select(colnames(test3 %>% dplyr::select(-salary)))))  
  
Tourn\_GK3 <- cbind(Tourn\_GK2, model\_salary)%>%  
 left\_join(Tourn\_Result, by="nation")

## Tournament EDA

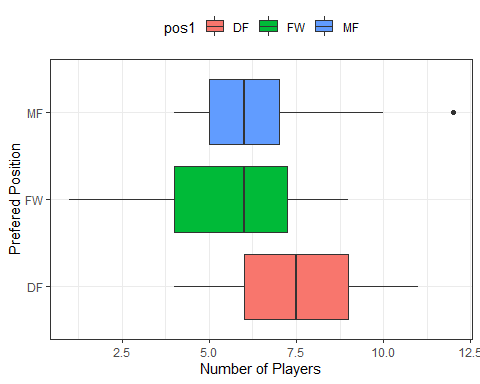
### Number of Players

* Examining if there is a notable effect on the number of players to the result.
* Overall it does show the team created should probably have around 20 people. Up to 24 is required if there are not people that play other positions. If there are secondary positions ideally it is midfield. This does show that it is the total number of people in the combined first and secondary positions that matter.
* No obvious relationship with total number of people able to play a position and performance.
* Shows not many multiposition players but in the best teams generally midfielders if they exist. The effect of the second position doesn’t seem significant. (If anything seems to be slight negative relationship between number of players with secondary position and performance)
* Some teams don’t have a goalkeeper… (serious data reliability issues). Best not to look at specifics but generalise expected team values.

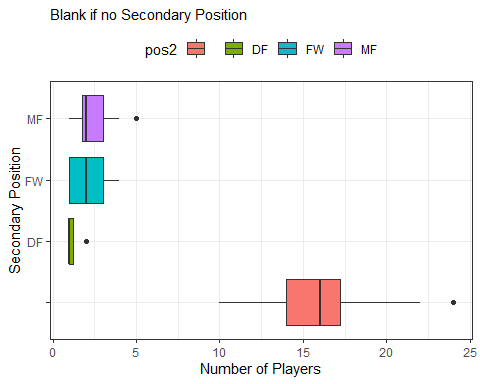
# Boxplots for no. of people in each position  
  
Tourn\_Data3 %>%  
 group\_by(nation) %>%  
 count() %>%  
 ggplot(aes(x=n)) +  
 geom\_boxplot() +  
 labs(x = "Number of Players")



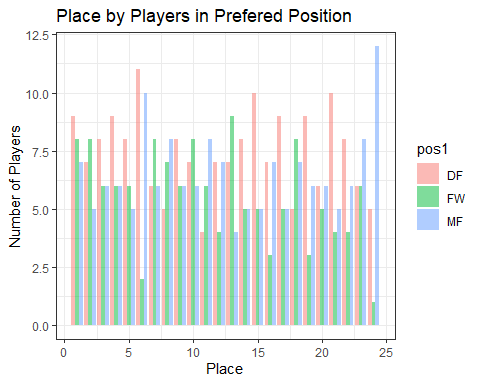
Tourn\_Data3 %>%  
 group\_by(nation, pos1) %>%  
 count() %>%  
 ggplot(aes(x=n, y=pos1, fill=pos1)) +  
 geom\_boxplot()+  
 labs(x = "Number of Players", y="Prefered Position")



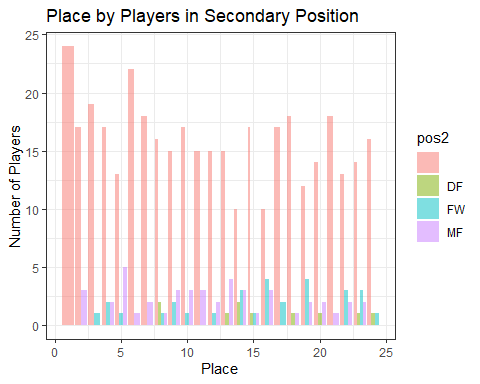
Tourn\_Data3 %>%  
 group\_by(nation, pos2) %>%  
 count() %>%  
 ggplot(aes(x=n, y=pos2, fill=pos2)) +  
 geom\_boxplot()+  
 labs(x = "Number of Players", y="Secondary Position", subtitle = "Blank if no Secondary Position")



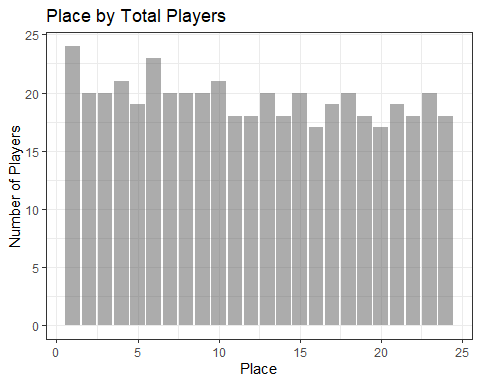
Tourn\_Data3 %>%  
 group\_by(place, pos1) %>%  
 count() %>%  
 ggplot(aes(x=place, y=n, fill=pos1)) +  
 geom\_bar(position = "dodge", alpha = 0.5, stat = "identity")+  
 theme\_bw() +  
 labs(y = "Number of Players", x="Place", title = "Place by Players in Prefered Position")



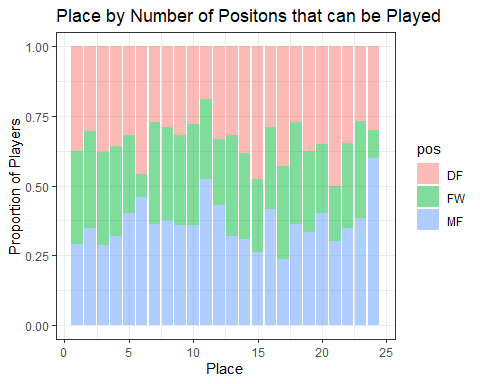
# Shows too many midfielders aren't ideal but fairly even break up of players  
  
Tourn\_Data3 %>%  
 group\_by(place, pos2) %>%  
 count() %>%  
 ggplot(aes(x=place, y=n, fill=pos2)) +  
 geom\_bar(position = "dodge", alpha = 0.5, stat = "identity")+  
 theme\_bw() +  
 labs(y = "Number of Players", x="Place", title = "Place by Players in Secondary Position")



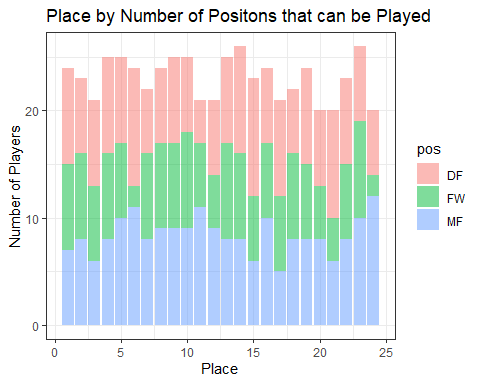
Tourn\_Data3 %>%  
 group\_by(place) %>%  
 count() %>%  
 ggplot(aes(x=place, y=n)) +  
 geom\_bar(alpha = 0.5, stat = "identity")+  
 theme\_bw() +  
 labs(y = "Number of Players", x="Place", title = "Place by Total Players")



#Combine positions  
  
a <- Tourn\_Data3 %>%  
 group\_by(place, pos1) %>%  
 count()   
b <- Tourn\_Data3 %>%  
 group\_by(place, pos2) %>%  
 count()  
  
comb\_pos <- a %>%  
 left\_join(b, by=c("place", "pos1"="pos2")) %>%  
 mutate(n.z=ifelse(is.na(n.y), 0, n.y)) %>%  
 transmute(no. = n.z+n.x) %>%  
 rename(pos = "pos1")   
   
  
comb\_pos %>%  
 ggplot(aes(x=place, y=no., fill=pos)) +  
 geom\_bar(position = "fill", alpha = 0.5, stat = "identity")+  
 theme\_bw() +  
 labs(y = "Proportion of Players", x="Place", title = "Place by Number of Positons that can be Played")



comb\_pos %>%  
 group\_by(place)%>%  
 ggplot(aes(x=place, y=no., fill=pos)) +  
 geom\_bar(position = "stack", alpha = 0.5, stat = "identity")+  
 theme\_bw() +  
 labs(y = "Number of Players", x="Place", title = "Place by Number of Positons that can be Played")

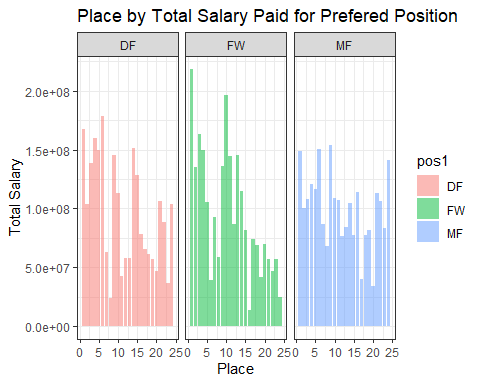


### Model Salary

* Can see a moderate positive relationship between total expected value of team and achievement.
* Teams that placed 7th and 8th in the tournament seem like outliers doing better than their suggested value, however, they had a large number of unknowns for league so perhaps not all the unknown players were from local leagues but instead had a few very high value players that played in leagues above A,B,C,D or E.
* Hard to consider the value of a second position (consider the the third and fourth letters in position to indicate the player’s least preferred position) and factoring it into player selection doesn’t seem to meaningfully change the outcome.
* Make a qualitative decision to have team proportions based on the proportion of salary spent on top 3 teams only looking at the preferred position.

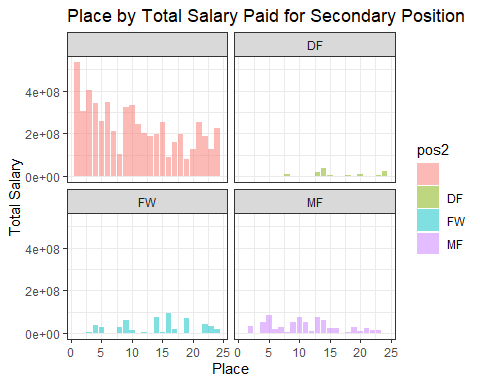
Tourn\_Data3 %>%  
 group\_by(place, pos1) %>%  
 summarise(cost=sum(model\_salary)) %>%  
 ggplot(aes(x=place, y=cost, fill=pos1)) +  
 geom\_bar(position = "dodge", alpha = 0.5, stat = "identity")+  
 facet\_wrap(~pos1)+  
 theme\_bw() +  
 labs(y = "Total Salary", x="Place", title = "Place by Total Salary Paid for Prefered Position")

## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.

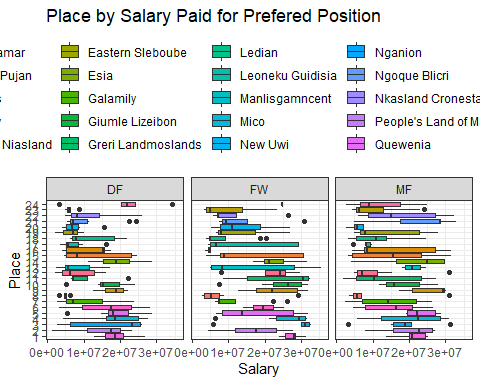


Tourn\_Data3 %>%  
 group\_by(place, pos2) %>%  
 summarise(cost=sum(model\_salary)) %>%  
 ggplot(aes(x=place, y=cost, fill=pos2)) +  
 geom\_bar(position = "dodge", alpha = 0.5, stat = "identity")+  
 facet\_wrap(~pos2)+  
 theme\_bw() +  
 labs(y = "Total Salary", x="Place", title = "Place by Total Salary Paid for Secondary Position")

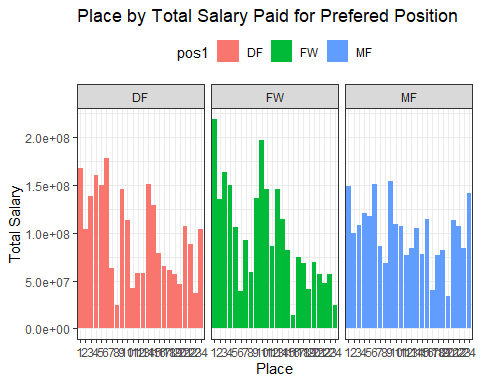
## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.



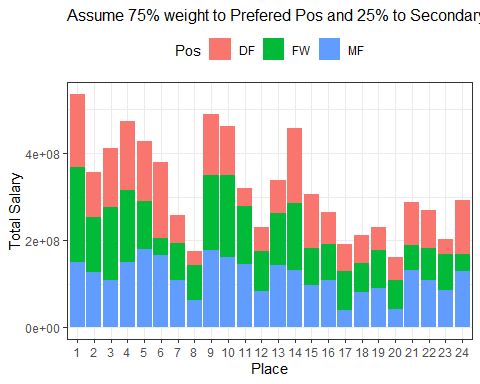
# Maybe a positive correlation with performance based on total spending on forwards, slightly negative with midfielders.  
  
Tourn\_Data3 %>%  
 group\_by(place, pos1) %>%  
 arrange(place)%>%  
 ggplot(aes(x=as.factor(place), y=model\_salary)) +  
 geom\_boxplot(aes(fill=nation))+  
 facet\_wrap(~pos1)+  
 coord\_flip()+  
 scale\_colour\_manual(values=colorvec)+  
 theme\_bw()+   
 theme(legend.position="top") +  
 labs(x = "Place", y="Salary", title = "Place by Salary Paid for Prefered Position")



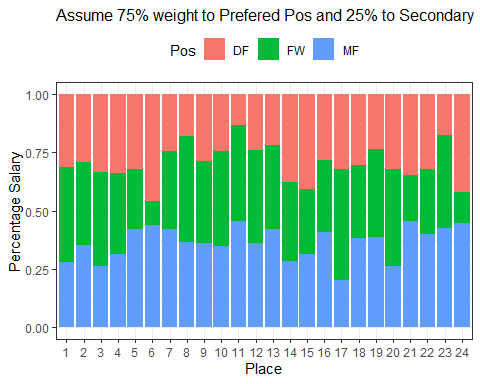
#This looks appealing but seems performance is more linked to the total expected value.  
  
Tourn\_Data3 %>%  
 group\_by(place, pos1) %>%  
 arrange(place)%>%  
 ggplot(aes(x=as.factor(place), y=model\_salary, fill=pos1)) +  
 geom\_bar(position = "stack", stat = "identity")+  
 facet\_wrap(~pos1)+  
 scale\_colour\_manual(values=colorvec)+  
 theme\_bw()+   
 theme(legend.position="top") +  
 labs(y = "Total Salary", x="Place", title = "Place by Total Salary Paid for Prefered Position")



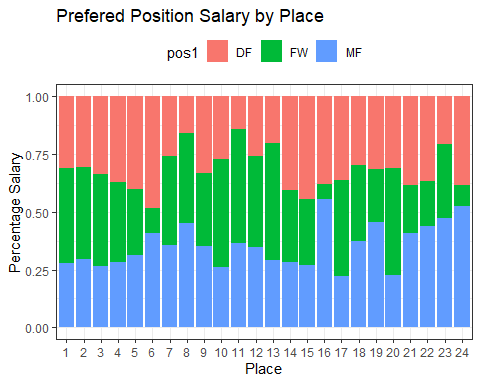
#Doesn't show much again see general reduction in total spending among all categories with place. WIll try and combine first and second position but give second position a quarter weight for the salary and first three quarters weight if they have two positions.  
  
## Stacked  
Tourn\_Data3 %>%  
 mutate\_all(na\_if,"") %>%  
 mutate(Two\_Pos=ifelse(is.na(pos2), 0, 1))%>%  
 pivot\_longer(c("pos1", "pos2"), names\_to = "Type", values\_to = "Pos", values\_drop\_na=TRUE)%>%  
 mutate(expected\_salary2 = ifelse(Two\_Pos==0, model\_salary, ifelse(Type=="Pos2", 0.25\*model\_salary, 0.75\*model\_salary)))%>%  
 dplyr::select(Pos, model\_salary, Type, expected\_salary2, place) %>%  
 ggplot(aes(x=as.factor(place), y=expected\_salary2, fill=Pos)) +  
 geom\_bar(position = "stack", stat = "identity")+  
 scale\_colour\_manual(values=colorvec)+  
 theme\_bw()+   
 theme(legend.position="top")+  
 theme(plot.title = element\_text(size=12)) +  
 labs(y = "Total Salary", x="Place", title = "Assume 75% weight to Prefered Pos and 25% to Secondary")



##Percentage  
Tourn\_Data3 %>%  
 mutate\_all(na\_if,"") %>%  
 mutate(Two\_Pos=ifelse(is.na(pos2), 0, 1))%>%  
 pivot\_longer(c("pos1", "pos2"), names\_to = "Type", values\_to = "Pos", values\_drop\_na=TRUE)%>%  
 mutate(expected\_salary2 = ifelse(Two\_Pos==0, model\_salary, ifelse(Type=="Pos2", 0.25\*model\_salary, 0.75\*model\_salary)))%>%  
 dplyr::select(Pos, model\_salary, Type, expected\_salary2, place) %>%  
 ggplot(aes(x=as.factor(place), y=expected\_salary2, fill=Pos)) +  
 geom\_bar(position = "fill", stat = "identity")+  
 scale\_colour\_manual(values=colorvec)+  
 theme\_bw()+   
 theme(legend.position="top")+  
 theme(plot.title = element\_text(size=12))+  
 labs(y = "Percentage Salary", x="Place", title = "Assume 75% weight to Prefered Pos and 25% to Secondary") #Is it much different to just considering first position?



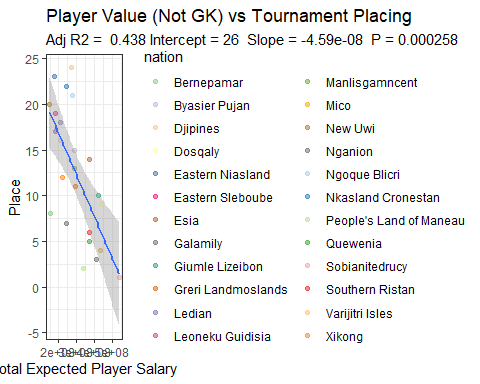
Tourn\_Data3 %>%  
 dplyr::select(pos1, model\_salary, place) %>%  
 ggplot(aes(x=as.factor(place), y=model\_salary, fill=pos1)) +  
 geom\_bar(position = "fill", stat = "identity")+  
 scale\_colour\_manual(values=colorvec)+  
 theme\_bw()+   
 theme(legend.position="top") +  
 labs(y = "Percentage Salary", x="Place", title = "Prefered Position Salary by Place")



#Considers proportion spent on each position for the team based on the top three in the tournament.  
  
#Model Value - Looking at total cost of players  
Tourn\_place <- function(fit){  
  
Tourn\_Data3 %>%  
 group\_by(place, nation) %>%  
 summarise(player\_value=sum(model\_salary))%>%  
 ggplot(aes\_string(x = names(fit$model)[2], y = names(fit$model)[1])) +  
 geom\_point(aes(colour=nation), alpha = 0.5, stat = "identity")+  
 stat\_smooth(method = "lm")+   
 theme(legend.position="top") +  
 scale\_colour\_manual(values=colorvec)+  
 theme\_bw() +  
 labs(subtitle = paste("Adj R2 = ",signif(summary(fit)$adj.r.squared, 3),  
 "Intercept =",signif(fit$coef[[1]],3 ),  
 " Slope =",signif(fit$coef[[2]], 3),  
 " P =",signif(summary(fit)$coef[2,4], 3)),  
 title = "Player Value (Not GK) vs Tournament Placing",  
 y = "Place", x="Total Expected Player Salary")  
}  
  
Tourn\_place(lm(place ~ player\_value, data = Tourn\_Data3%>%  
 group\_by(place, nation) %>%  
 summarise(player\_value=sum(model\_salary))))

## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.  
## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.

## `geom\_smooth()` using formula 'y ~ x'

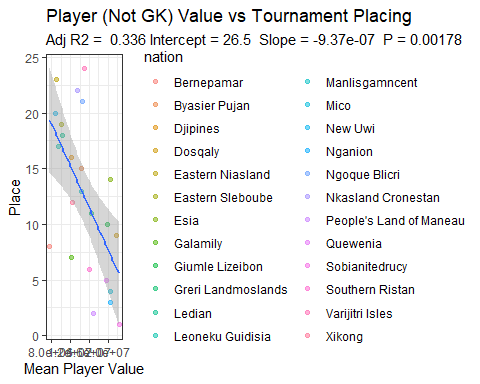


#Can see there is a relationship but it is moderate. Need to also adjust for where they were the previous year perhaps.  
  
#Model Value - Looking at mean cost of players  
Tourn\_place2 <- function(fit){  
  
Tourn\_Data3 %>%  
 group\_by(place, nation) %>%  
 summarise(player\_value=mean(model\_salary))%>%  
 ggplot(aes\_string(x = names(fit$model)[2], y = names(fit$model)[1])) +  
 geom\_point(aes(colour=nation), alpha = 0.5, stat = "identity")+  
 stat\_smooth(method = "lm")+   
 theme(legend.position="top")+  
 theme\_bw() +  
 labs(subtitle = paste("Adj R2 = ",signif(summary(fit)$adj.r.squared, 3),  
 "Intercept =",signif(fit$coef[[1]],3 ),  
 " Slope =",signif(fit$coef[[2]], 3),  
 " P =",signif(summary(fit)$coef[2,4], 3)),  
 title = "Player (Not GK) Value vs Tournament Placing",  
 y = "Place", x="Mean Player Value")  
}  
  
Tourn\_place2(lm(place ~ player\_value, data = Tourn\_Data3%>%  
 group\_by(place, nation) %>%  
 summarise(player\_value=mean(model\_salary))))

## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.

## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.

## `geom\_smooth()` using formula 'y ~ x'

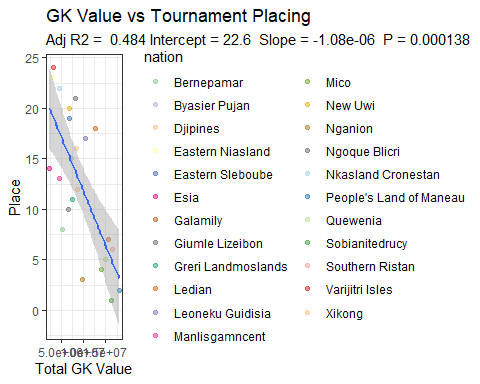


#Total expected salary seems a better indicator of place  
  
## For GK  
  
  
## MOdel value looking at total cost of GK  
Tourn\_place3 <- function(fit){  
  
Tourn\_GK3 %>%  
 group\_by(place, nation) %>%  
 summarise(player\_value=sum(model\_salary))%>%  
 ggplot(aes\_string(x = names(fit$model)[2], y = names(fit$model)[1])) +  
 geom\_point(aes(colour=nation), alpha = 0.5, stat = "identity")+  
 stat\_smooth(method = "lm")+  
 scale\_colour\_manual(values=colorvec)+   
 theme(legend.position="top")+  
 theme\_bw() +  
 labs(subtitle = paste("Adj R2 = ",signif(summary(fit)$adj.r.squared, 3),  
 "Intercept =",signif(fit$coef[[1]],3 ),  
 " Slope =",signif(fit$coef[[2]], 3),  
 " P =",signif(summary(fit)$coef[2,4], 3)),  
 title = "GK Value vs Tournament Placing",  
 y = "Place", x="Total GK Value")  
}  
  
Tourn\_place3(lm(place ~ player\_value, data = Tourn\_GK3%>%  
 group\_by(place, nation) %>%  
 summarise(player\_value=sum(model\_salary))))

## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.

## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.

## `geom\_smooth()` using formula 'y ~ x'

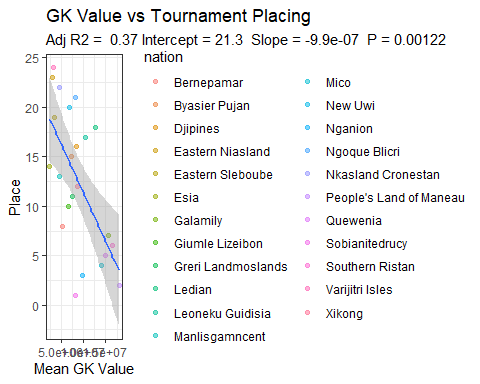


#Model Value - Looking at mean cost of GK  
Tourn\_place4 <- function(fit){  
  
Tourn\_GK3 %>%  
 group\_by(place, nation) %>%  
 summarise(player\_value=mean(model\_salary))%>%  
 ggplot(aes\_string(x = names(fit$model)[2], y = names(fit$model)[1])) +  
 geom\_point(aes(colour=nation), alpha = 0.5, stat = "identity")+  
 stat\_smooth(method = "lm")+   
 theme(legend.position="top")+  
 theme\_bw() +  
 labs(subtitle = paste("Adj R2 = ",signif(summary(fit)$adj.r.squared, 3),  
 "Intercept =",signif(fit$coef[[1]],3 ),  
 " Slope =",signif(fit$coef[[2]], 3),  
 " P =",signif(summary(fit)$coef[2,4], 3)),  
 title = "GK Value vs Tournament Placing",  
 y = "Place", x="Mean GK Value")  
}  
  
Tourn\_place4(lm(place ~ player\_value, data = Tourn\_GK3%>%  
 group\_by(place, nation) %>%  
 summarise(player\_value=mean(model\_salary))))

## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.

## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.

## `geom\_smooth()` using formula 'y ~ x'

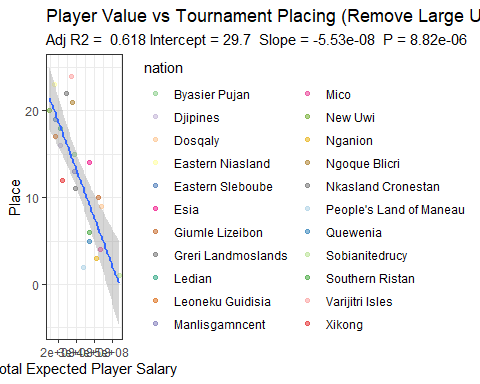


# This looks at model value removing the two teams which had alot of unknowns for league (where the local footbal league assumption has a large impact)  
Tourn\_place\_adj <- function(fit){  
  
Tourn\_Data3 %>%  
 dplyr::filter(!place %in% c(7,8)) %>%  
 group\_by(place, nation) %>%  
 summarise(player\_value=sum(model\_salary))%>%  
 ggplot(aes\_string(x = names(fit$model)[2], y = names(fit$model)[1])) +  
 geom\_point(aes(colour=nation), alpha = 0.5, stat = "identity")+  
 stat\_smooth(method = "lm")+   
 theme(legend.position="top")+  
 scale\_colour\_manual(values=colorvec)+  
 theme\_bw() +  
 labs(subtitle = paste("Adj R2 = ",signif(summary(fit)$adj.r.squared, 3),  
 "Intercept =",signif(fit$coef[[1]],3 ),  
 " Slope =",signif(fit$coef[[2]], 3),  
 " P =",signif(summary(fit)$coef[2,4], 3)),  
 title = "Player Value vs Tournament Placing (Remove Large Unknown League #",  
 y = "Place", x="Total Expected Player Salary")  
}  
  
Tourn\_place\_adj(lm(place ~ player\_value, data = Tourn\_Data3%>%  
 dplyr::filter(!place %in% c(7,8)) %>%  
 group\_by(place, nation) %>%  
 summarise(player\_value=sum(model\_salary))))

## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.

## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.

## `geom\_smooth()` using formula 'y ~ x'



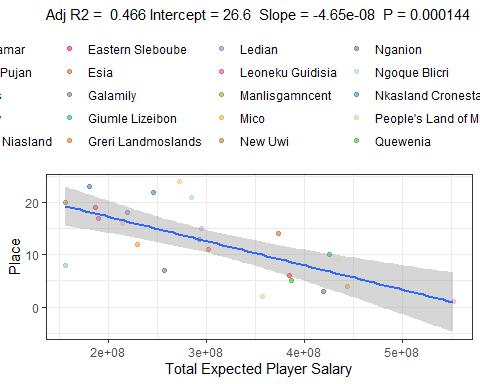
##Total salary of players combining unknowns  
  
#Model Value - Looking at total cost of players  
#Figure 2 - Player (Field and Goalkeeper) Value for 2021 Tournament Teams vs Tournament Placing  
Tourn\_place <- function(fit){  
  
rbind((Tourn\_Data3 %>% dplyr::select(-pos2)), (Tourn\_GK3 %>% dplyr::select(player, nation, league, model\_salary, place)%>% mutate(pos1 = "GK"))) %>%  
 group\_by(place, nation) %>%  
 summarise(player\_value=sum(model\_salary))%>%  
 ggplot(aes\_string(x = names(fit$model)[2], y = names(fit$model)[1])) +  
 geom\_point(aes(colour=nation), alpha = 0.5, stat = "identity")+  
 stat\_smooth(method = "lm")+  
 scale\_colour\_manual(values=colorvec)+   
 theme(legend.position="top")+  
 theme\_bw() +  
 labs(subtitle = paste("Adj R2 = ",signif(summary(fit)$adj.r.squared, 3),  
 "Intercept =",signif(fit$coef[[1]],3 ),  
 " Slope =",signif(fit$coef[[2]], 3),  
 " P =",signif(summary(fit)$coef[2,4], 3)),  
 y = "Place", x="Total Expected Player Salary")+  
 theme(legend.position = "top")  
}  
  
Team\_Value\_Vs\_Place\_2021\_Tournament <- Tourn\_place(lm(place ~ player\_value, data = (rbind((Tourn\_Data3 %>% dplyr::select(-pos2)), (Tourn\_GK3 %>% dplyr::select(player, nation, league, model\_salary, place)%>% mutate(pos1 = "GK")))%>%  
 group\_by(place, nation) %>%  
 summarise(player\_value=sum(model\_salary)))))

## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.

## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.

Team\_Value\_Vs\_Place\_2021\_Tournament

## `geom\_smooth()` using formula 'y ~ x'



## Salary Adjustments

* Salary is required to be adjusted to account for superimposed inflation up to year 2022.
* Likewise, in makes sense to account for the amount of games a player is expected to play ie. there is annualised salary data for a football season (38 games) but the amount of games expected to be played is significantly less for a national tournament and it makes sense for salaries to reflect this. (Didn’t end up dong this).

\*\* Note: In reality football players are normally paid a lot less when playing for their national team and many donate these funds to charity. This hasn’t been taken into account in this analysis.

Super\_Inflation <- (Data %>% group\_by(year) %>% summarise(mean(salary)))[2,2]/(Data %>% group\_by(year) %>% summarise(mean(salary)))[1,2]  
  
Game\_Ratio <- (Data%>%summarise(mean(games)))[1,1]/(Tourn\_FW%>%summarise(mean(games)))[1,1] #Not going to be used even though it would be anticipated that players would be paid more per game.

## Applying Model to Rarita

* Value of players found by using GBM.

#Predict Rarita Player Salary  
  
predGBM\_Rarita <-predict(gbmFitF\_GK, newdata=(GK\_Data\_Rarita %>% dplyr::select(colnames(test3))))  
  
Rarita\_GK <- cbind(GK\_Data\_Rarita,predGBM\_Rarita) %>% dplyr::select (player, nation, salary, predGBM\_Rarita, year) %>%   
 rename(model\_salary = "predGBM\_Rarita")%>%  
 mutate(differ = model\_salary - salary) #There aren't any undervalued Rarita GK's.  
  
list2020\_GK <- Rarita\_GK %>%  
 dplyr::filter(year==2020)  
  
list2021\_GK <- Rarita\_GK %>%  
 dplyr::filter(year==2021)  
  
play2020\_not2021\_GK <- anti\_join(list2020\_GK, list2021\_GK, by="player") #0 GK from 2020 didn't play 2021  
GK\_list <- rbind(play2020\_not2021\_GK, list2021\_GK)  
  
#All league GK's  
predGBM <-predict(gbmFitF\_GK, newdata=Data\_GK\_Simple %>% dplyr::select(colnames(test3)))  
  
Salary <- cbind(Data\_GK\_Simple,predGBM) %>% dplyr::select (player, year, nation, salary, predGBM) %>% mutate(Difference = salary - predGBM)   
  
  
write.csv(Salary, "League GK Salary.csv")

predRarita <- predict(gbmFitF, newdata=(Rarita\_Data2%>%dplyr::select(-player)))  
  
Rarita\_Data3 <- Rarita\_Data2 %>%  
 mutate(model\_salary = predRarita) %>%  
 mutate(differ = model\_salary-salary)  
  
list2020 <- Rarita\_Data3 %>%  
 separate(pos, c("pos1", "pos2"), sep=2) %>%  
 dplyr::filter(year==2020)  
  
list2021 <- Rarita\_Data3 %>%  
 separate(pos, c("pos1", "pos2"), sep=2) %>%  
 dplyr::filter(year==2021)  
  
play2020\_not2021 <- anti\_join(list2020, list2021, by="player") #47 Players from 2020 didn't play 2021  
player\_list <- rbind(play2020\_not2021, list2021) #To capture players that played 2020 but didn't play 2021 for whatever reason  
  
  
## Expected salary results for the league based on the model  
  
league\_salary <- reduced\_Data  
  
league\_salary$games <- as.numeric(league\_salary$games)  
league\_salary <- kNN(league\_salary,weightDist = FALSE,imp\_var = FALSE)  
  
pred2 <- predict(gbmFitF, newdata=league\_salary%>%dplyr::select(-nation))  
  
league\_salary2 <- league\_salary %>%  
 mutate(expected\_salary = pred2) %>%  
 dplyr::select(player, salary, year, expected\_salary)  
  
writexl::write\_xlsx(league\_salary2, "league\_salary.xlsx")

## Team to Choose From

* Must consider 2020 players that did not play in 2021

Rarita\_Data4 <- player\_list %>%  
 group\_by(pos1) %>%  
 mutate(model\_salary = ifelse(year==2020, model\_salary\*as.numeric(Super\_Inflation), model\_salary),  
 salary = ifelse(year==2020, salary\*as.numeric(Super\_Inflation), salary)) %>%  
 dplyr::select(player, pos1, salary, model\_salary, differ)  
  
Rarita\_GK <- GK\_list %>%  
 mutate(model\_salary = ifelse(year==2020, model\_salary\*as.numeric(Super\_Inflation), model\_salary),  
 salary = ifelse(year==2020, salary\*as.numeric(Super\_Inflation), salary),  
 pos1 = "GK") %>%  
 dplyr::select(player, pos1, salary, model\_salary, differ)  
  
Rarita\_Squad <- rbind(Rarita\_Data4, Rarita\_GK)

## Expected Value of Future Teams

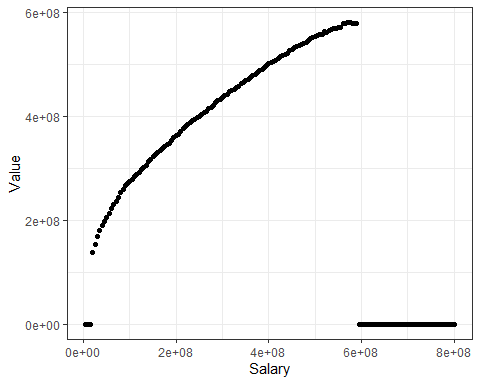
* Objectives of the Optimisation:
  + Maximize the value of the team.
  + Meet a budgetary constraint which is preset; have the proportion of budget inline with proportions from the top three teams in the tournament.
  + Minimum number in each position (once again only considering position 1)
* Creates a function to determine the value of a team based on its salary.
* Assumes that Rarita will have similar talent in the future. (This is a conservative assumption)
* Takes into account the superimposed inflation rate for the year, it is calculated bringing prices back to 2021.
* Eventually the increase in the predicted value plateaus likewise there is a minimum for the cheapest team that can be created. These have been determined so the polynomial equation can be fitted.
* A fitted polynomial equation can then be used to calculate most optimum allocation of resources to meet competitiveness requirements

\*\* Note: Function is called “Predicted\_Value” the first functions in this chunck are used to find the polynomial equations

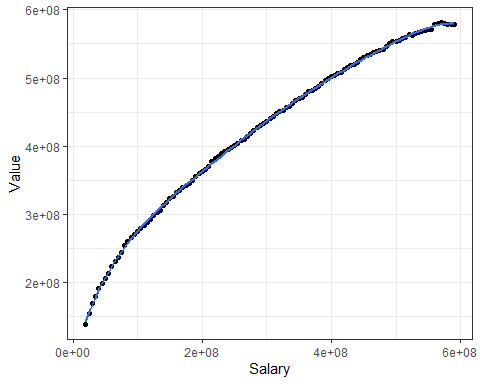
# Proportions  
Pos\_Prop <- rbind(Tourn\_Data3 %>% dplyr::select(place, pos1, model\_salary), (Tourn\_GK3%>%mutate(pos1 = "GK")%>% dplyr::select(place, pos1, model\_salary))) %>%  
 dplyr::filter(place==c(1,2,3))%>%  
 group\_by(pos1) %>%  
 summarise(salary\_total=sum(model\_salary)) %>%  
 mutate(Prop = salary\_total/sum(salary\_total)) %>%  
 rename(Pos = "pos1") %>%  
 dplyr::select(-salary\_total)

## Warning in place == c(1, 2, 3): longer object length is not a multiple of  
## shorter object length

DF\_Prop <- as.numeric(Pos\_Prop%>%  
 dplyr::filter(Pos=="DF")%>%  
 dplyr::select(Prop))  
  
FW\_Prop <- as.numeric(Pos\_Prop%>%  
 dplyr::filter(Pos=="FW")%>%  
 dplyr::select(Prop))  
  
MF\_Prop <- as.numeric(Pos\_Prop%>%  
 dplyr::filter(Pos=="MF")%>%  
 dplyr::select(Prop))  
  
GK\_Prop <- as.numeric(Pos\_Prop%>%  
 dplyr::filter(Pos=="GK")%>%  
 dplyr::select(Prop))  
  
# Probability Table  
  
Expected\_Value <- function(x){  
Opt <- Rarita\_Squad %>%  
 mutate(values = 1) %>%  
 pivot\_wider(names\_from = pos1, values\_from = values, values\_fill = 0) %>%  
 mutate(FW\_Sal = salary\*FW,  
 MF\_Sal = salary\*MF,  
 DF\_Sal = salary\*DF)  
  
  
Max\_Salary <- x/as.numeric(Super\_Inflation)  
Max\_Prop\_DF <- Max\_Salary\*DF\_Prop+0.05\*Max\_Salary  
Max\_Prop\_FW <- Max\_Salary\*FW\_Prop+0.05\*Max\_Salary  
Max\_Prop\_MF <- Max\_Salary\*MF\_Prop+0.05\*Max\_Salary  
Min\_Prop\_DF <- Max\_Salary\*DF\_Prop-0.05\*Max\_Salary  
Min\_Prop\_FW <- Max\_Salary\*FW\_Prop-0.05\*Max\_Salary  
Min\_Prop\_MF <- Max\_Salary\*MF\_Prop-0.05\*Max\_Salary  
Max\_DF <- 8  
Max\_FW <- 8  
Max\_MF <- 8  
Min\_DF <- 5  
Min\_FW <- 5  
Min\_MF <- 5  
Min\_GK <- 1  
  
## Set the coefficients of the decision variables -> C  
C <- Opt$model\_salary  
  
# Create constraint martix B  
A <- matrix(c(Opt$salary,  
 Opt$DF\_Sal,  
 Opt$FW\_Sal,  
 Opt$MF\_Sal,  
 Opt$DF\_Sal,  
 Opt$FW\_Sal,  
 Opt$MF\_Sal,  
 Opt$DF,  
 Opt$FW,  
 Opt$MF,  
 Opt$DF,  
 Opt$FW,  
 Opt$MF,  
 Opt$GK), nrow=14, byrow=TRUE)  
  
# Right hand side for the constraints  
B <- c(Max\_Salary, Max\_Prop\_DF, Max\_Prop\_FW, Max\_Prop\_MF, Min\_Prop\_DF, Min\_Prop\_FW, Min\_Prop\_MF, Max\_DF, Max\_FW, Max\_MF, Min\_DF, Min\_FW, Min\_MF, Min\_GK)  
  
# Direction of the constraints  
constranints\_direction <- c( "<=","<=", "<=", "<=", ">=", ">=", ">=","<=", "<=", "<=", ">=", ">=", ">=", "=")  
  
  
# Find the optimal solution  
optimum <- lp(direction="max",  
 objective.in = C,  
 const.mat = A,  
 const.dir = constranints\_direction,  
 const.rhs = B,  
 all.bin = TRUE)  
optimum$status  
optimum$solution  
  
  
Result <- cbind(Opt,optimum$solution) %>%  
 rename(picked = "optimum$solution") %>%  
 dplyr::filter(picked == 1)  
  
  
return(as.numeric(sum(Result$model\_salary)))  
  
}  
  
# Apply optimization model to see relationship between salary and expected salary  
  
Minimum\_model\_salary <- as.numeric(Rarita\_Squad %>%  
 group\_by(pos1) %>%  
 arrange(salary) %>%  
 mutate(Rank = rank(model\_salary)) %>%  
 dplyr::filter(Rank <= 5) %>%  
 dplyr::filter(!Rank >= 2|!pos1 == "GK") %>%  
 ungroup() %>%  
 summarise(min\_sal = sum(model\_salary))) #Roughly $50,000,000 is the minimum total salary with 6 players each position.  
  
Maximum\_model\_salary <- as.numeric(Rarita\_Squad %>%  
 group\_by(pos1) %>%  
 arrange(desc(model\_salary)) %>%  
 mutate(Rank = rank(desc(model\_salary))) %>%  
 dplyr::filter(Rank <= 8) %>%  
 dplyr::filter(!Rank >= 2|!pos1 == "GK") %>%  
 ungroup() %>%  
 summarise(max\_sal = sum(model\_salary))) #Roughly $600,000,000 is the maximum expected value with 8 players in each position (like the 2021 tournament winners)  
  
Value <- c()  
  
for(i in 1:160){  
Expected\_Salary = as.numeric(Expected\_Value(i\*5000000))  
Value[i] <- Expected\_Salary  
}  
  
salary\_to\_value <- as.data.frame(cbind(Salary = seq(1,160)\*5000000, Value))  
  
salary\_to\_value %>%  
 ggplot(aes(x=Salary, y=Value)) +  
 geom\_point()



salary\_to\_value2 <- salary\_to\_value %>%  
 dplyr::filter(Value >= Minimum\_model\_salary) %>%  
 dplyr::filter(Value <= Maximum\_model\_salary)  
  
sal\_to\_val <- lm(Value ~ poly(Salary, 6), data = salary\_to\_value2)  
  
# Figure 1 - Total Paid Salary for Rarita National Team Players Mapped to the Toal Underlying Team Value that can be Achieved (With a Fitted 6th Order Polynomial)  
  
salary\_to\_value2 %>%  
ggplot(aes(Salary,Value)) +   
 geom\_point() +   
 geom\_smooth(method="lm", formula=y~I(x^6)+I(x^5)+I(x^4)+I(x^3)+I(x^2)+x)



#The following function is used to predict the expected value in 2021 terms (taking into account inflation).  
  
  
  
Predicted\_Value <- function(x,y){  
dd <- as.data.frame(cbind(Future\_Salary = x, Year = y)) %>%  
 transmute(Salary = Future\_Salary\*as.numeric(Super\_Inflation)^(2021-Year))  
pred <- predict(sal\_to\_val, newdata = dd)  
return(as.numeric(pred))  
}

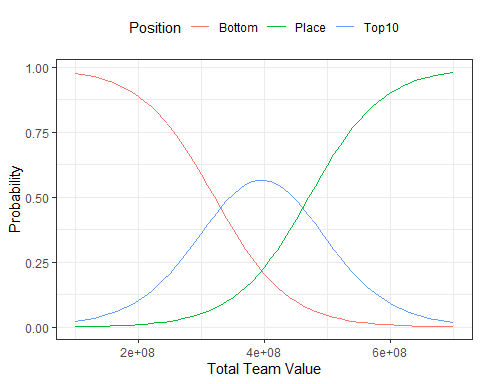
## Expected Value Result

* This is an ordinal logistics model which uses the expected value of a team from 2021 to determine its probability of finishing top 3, top 10, or outside of the top 10.
* It is too hard to determine probability of finishing first (as only one year of reliable data). This could be refined over the first couple years in the tournament. (common sense all models should be refined with new information).
* Plot to show place based on expected total player value for Rarita applied in 2021. (Maybe other factors but this seems most critical and once again using more than one variable on ne years data risks overfitting)
* Need to enter predicted total team value into function to obtain probabilities. (Function Called Probs)

Tourn\_Sum <- Tourn\_Data3 %>%  
 group\_by(place, nation) %>%  
 summarise(player\_value=sum(model\_salary))%>%  
 mutate(Finish = as.factor(if\_else(place<=3, "Place", if\_else(place<=10, "Top10", "Bottom"))))

## `summarise()` has grouped output by 'place'. You can override using the `.groups` argument.

m <- polr(Finish~player\_value, data = Tourn\_Sum, Hess =TRUE)  
  
player\_value <- seq(100000000, 700000000, 10000000)  
rarita\_cost <- data.frame(player\_value)  
  
m2 <- cbind(rarita\_cost,as.data.frame(predict(m, type="probs", newdata = rarita\_cost)))  
  
  
#Figure 3 - Total Team Value Discounted to 2021 and Corresponding Probabilities of Finishing Top 3 (Place), Top 10 But Not Top 3 (Top 10) or Outside Top 10 (Bottom)  
m2 %>%  
 pivot\_longer(!player\_value, names\_to = "Position", values\_to = "Probability")%>%  
 ggplot(aes(x=player\_value, y=Probability, color=Position))+   
 geom\_line() +  
 labs(x="Total Team Value")



Probs <- function(y){  
 vv <- as.data.frame(x=y)   
 colnames(vv)[1] <- "player\_value"  
 pred <- predict(m, type="probs", newdata = vv)   
   
 return((pred))  
}  
  
Probs(as.numeric(Predicted\_Value(x= 300000000, y=2025)))

## Place Top10 Bottom   
## 0.2189652 0.5654492 0.2155856

# For a 50% probability of finishing in a place average player expected player value will need to be 470500000 over the next 10 years in 2021 dollar (ie not adjusted for inflation).

## Determining Optimal Spending Over 10 Years

* Combines the fitted polynomial (matches total paid salary to total team value) and the probability curves (remember all total team values are determined using 2021 prices therefore must be adjusted for inflation)
* As the return on investment is yet to be determined must be able to adjust for this.
* Assumes constant superimposed inflation rate (5.353%)
* Generate response for different ROI and select situation that matches practical expectations for team performance whilst not exposing team to unnecessary investment risk. (ie. big spendings occurs at ideal times but don’t want unnecessary high target ROI as there will be a higher variability in portfolio return)
* ROI of 12% is ideal as key spending years move to years 4 and 5 which allow for operation to become established and models to be verified.
* Objectives:
  + Overall, 95% chance of finishing top 3 in next ten years.
  + 99% chance of finishing top 10 over next 5 years.
  + Minimise NPV of future salaries paid.

Each\_Year\_Spend <- function(v){  
RR <- as.numeric(v/Super\_Inflation)  
  
Opt2 <- expand\_grid(Year = seq(2022,2031),  
 Salary = seq(3,60)\*10000000)  
  
Prob\_Bottom <- function(x){  
 return(as.numeric(Probs(x)[3]))  
}  
  
Prob\_Top\_10\_Not\_Place <- function(x){  
 return(as.numeric(Probs(x)[2]))  
}  
  
Prob\_Place <- function(x){  
 return(as.numeric(Probs(x)[1]))  
}  
  
  
Opt3 <- Opt2 %>%  
 mutate(Salary\_NPV\_2022 = Salary\*(RR^(2022-Year))) %>% #Takes into account inflation and expected rate of return of money invested. Need to take into account superimposed inflation from 2021 but we get the money in 2022 so we were unable to invest it.  
 mutate(Expected\_Value\_2021 = mapply(function(x,y)Predicted\_Value(x, y),  
 Salary,  
 Year)) %>% #Determines expected value of team by first putting paid salaries in 2021 terms (taking into account inflation) then using fitted polynomial.  
 mutate(Prob\_Bottom = mapply(function(x)Prob\_Bottom(x),  
 Expected\_Value\_2021)) %>% #Uses expected value 2021 to find probability of finishing outside top 10.  
 mutate(Prob\_Bottom\_Adj = ifelse(Year <= 2026, log(Prob\_Bottom), 0)) %>% #Adjusts for the fact we only care about finishing in the top 10 within first 5 years.  
 mutate(Prob\_Top\_10\_Not\_Place = mapply(function(x)Prob\_Top\_10\_Not\_Place(x),  
 Expected\_Value\_2021)) %>% #Uses expected value in 2021 dollar value to find probability of placing.  
 mutate(Prob\_Not\_Place\_Adj = log(Prob\_Top\_10\_Not\_Place+Prob\_Bottom)) %>%  
 mutate(Y2022 = ifelse(Year==2022, 1, 0),  
 Y2023 = ifelse(Year==2023, 1, 0),  
 Y2024 = ifelse(Year==2024, 1, 0),  
 Y2025 = ifelse(Year==2025, 1, 0),  
 Y2026 = ifelse(Year==2026, 1, 0),  
 Y2027 = ifelse(Year==2027, 1, 0),  
 Y2028 = ifelse(Year==2028, 1, 0),  
 Y2029 = ifelse(Year==2029, 1, 0),  
 Y2030 = ifelse(Year==2030, 1, 0),  
 Y2031 = ifelse(Year==2031, 1, 0)  
 ) %>%  
 dplyr::filter(Expected\_Value\_2021 <= 550000000)%>% #This is where the benefit for spending extra money really start to plateau  
 dplyr::filter(Expected\_Value\_2021 >= 200000000) #Close to the minimum value of a team with at least 5 in a position.  
   
# Optimization  
  
Max\_Bottom <- log(0.01) #At least 99% chance (Almost certain) finishing in top 10 in first 5 years, we need to think of this as a 50% chance of making top 10 at least once over the next 5 years.  
Min\_Place <- log(0.05) #At least 95% chance finishing in top 3 over the next 10 years  
S2022 <- 1  
S2023 <- 1  
S2024 <- 1  
S2025 <- 1  
S2026 <- 1  
S2027 <- 1  
S2028 <- 1  
S2029 <- 1  
S2030 <- 1  
S2031 <- 1  
  
  
  
## Set the coefficients of the decision variables -> C  
C <- Opt3$Salary\_NPV\_2022  
  
# Create constraint matrix B  
A <- matrix(c(Opt3$Prob\_Bottom\_Adj,  
 Opt3$Prob\_Not\_Place\_Adj,  
 Opt3$Y2022,  
 Opt3$Y2023,  
 Opt3$Y2024,  
 Opt3$Y2025,  
 Opt3$Y2026,  
 Opt3$Y2027,  
 Opt3$Y2028,  
 Opt3$Y2029,  
 Opt3$Y2030,  
 Opt3$Y2031  
 ), nrow=12, byrow=TRUE)  
  
# Right hand side for the constraints  
B <- c(Max\_Bottom, Min\_Place, S2022, S2023, S2024, S2025, S2026, S2027, S2028, S2029, S2030, S2031)  
  
# Direction of the constraints  
constranints\_direction <- c("<=","<=", "=", "=", "=", "=", "=", "=", "=", "=", "=", "=")  
  
# Find the optimal solution  
optimum3 <- lp(direction="min",  
 objective.in = C,  
 const.mat = A,  
 const.dir = constranints\_direction,  
 const.rhs = B,  
 all.bin = TRUE)  
  
  
  
Result3 <- cbind(Opt3,optimum3$solution) %>%  
 rename(picked = "optimum3$solution") %>%  
 dplyr::filter(picked == 1)  
  
  
  
return(Result3)  
}  
  
# 4% PA Return on investments  
ROI\_4pc <- Each\_Year\_Spend(1.04) %>%  
 mutate(ROI = "4%")  
writexl::write\_xlsx(ROI\_4pc, "ROI\_4PC\_res.xlsx")  
  
# 6% PA Return on investments  
ROI\_6pc <- Each\_Year\_Spend(1.06)%>%  
 mutate(ROI = "6%")  
writexl::write\_xlsx(ROI\_6pc, "ROI\_6PC\_res.xlsx")  
  
# 8% PA Return on investments  
ROI\_8pc <- Each\_Year\_Spend(1.08)%>%  
 mutate(ROI = "8%")  
writexl::write\_xlsx(ROI\_8pc, "ROI\_8PC\_res.xlsx")  
  
# 10% PA Return on investments  
ROI\_10pc <- Each\_Year\_Spend(1.10)%>%  
 mutate(ROI = "10%")  
writexl::write\_xlsx(ROI\_10pc, "ROI\_10PC\_res.xlsx")  
  
# 12% PA Return on investments  
ROI\_12pc <- Each\_Year\_Spend(1.12)%>%  
 mutate(ROI = "12%")  
writexl::write\_xlsx(ROI\_12pc, "ROI\_12PC\_res.xlsx")  
  
# 14% PA Return on investments  
ROI\_14pc <- Each\_Year\_Spend(1.14)%>%  
 mutate(ROI = "14%")  
writexl::write\_xlsx(ROI\_14pc, "ROI\_14PC\_res.xlsx")  
  
# 16% PA Return on investments  
ROI\_16pc <- Each\_Year\_Spend(1.16)%>%  
 mutate(ROI = "16%")  
writexl::write\_xlsx(ROI\_16pc, "ROI\_16PC\_res.xlsx")  
  
ROI\_4pc

## Year Salary Salary\_NPV\_2022 Expected\_Value\_2021 Prob\_Bottom Prob\_Bottom\_Adj  
## 1 2022 5.0e+08 500000000 541074245 0.02251459 -3.7935916  
## 2 2023 6.0e+07 60776320 213405350 0.86195936 -0.1485472  
## 3 2024 5.5e+08 564324607 538943316 0.02333054 -3.7579921  
## 4 2025 6.0e+07 62359223 204190667 0.87965980 -0.1282200  
## 5 2026 7.0e+07 73693747 213217443 0.86234119 -0.1481043  
## 6 2027 7.0e+07 74647245 208562345 0.87152001 0.0000000  
## 7 2028 7.0e+07 75613081 204011039 0.87998454 0.0000000  
## 8 2029 8.0e+07 87533044 211175925 0.86643266 0.0000000  
## 9 2030 8.0e+07 88665605 206565701 0.87529437 0.0000000  
## 10 2031 8.0e+07 89812819 202060527 0.88346228 0.0000000  
## Prob\_Top\_10\_Not\_Place Prob\_Not\_Place\_Adj Y2022 Y2023 Y2024 Y2025 Y2026 Y2027  
## 1 0.2076232 -1.46907701 1 0 0 0 0 0  
## 2 0.1258515 -0.01226403 0 1 0 0 0 0  
## 3 0.2133259 -1.44114576 0 0 1 0 0 0  
## 4 0.1099093 -0.01048566 0 0 0 1 0 0  
## 5 0.1255083 -0.01222493 0 0 0 0 1 0  
## 6 0.1172487 -0.01129489 0 0 0 0 0 1  
## 7 0.1096162 -0.01045368 0 0 0 0 0 0  
## 8 0.1218287 -0.01180804 0 0 0 0 0 0  
## 9 0.1138471 -0.01091787 0 0 0 0 0 0  
## 10 0.1064762 -0.01011252 0 0 0 0 0 0  
## Y2028 Y2029 Y2030 Y2031 picked ROI  
## 1 0 0 0 0 1 4%  
## 2 0 0 0 0 1 4%  
## 3 0 0 0 0 1 4%  
## 4 0 0 0 0 1 4%  
## 5 0 0 0 0 1 4%  
## 6 0 0 0 0 1 4%  
## 7 1 0 0 0 1 4%  
## 8 0 1 0 0 1 4%  
## 9 0 0 1 0 1 4%  
## 10 0 0 0 1 1 4%

ROI\_6pc

## Year Salary Salary\_NPV\_2022 Expected\_Value\_2021 Prob\_Bottom Prob\_Bottom\_Adj  
## 1 2022 5.0e+08 500000000 541074245 0.02251459 -3.7935916  
## 2 2023 6.0e+07 59629597 213405350 0.86195936 -0.1485472  
## 3 2024 5.5e+08 543230238 538943316 0.02333054 -3.7579921  
## 4 2025 6.0e+07 58895637 204190667 0.87965980 -0.1282200  
## 5 2026 7.0e+07 68287393 213217443 0.86234119 -0.1481043  
## 6 2027 7.0e+07 67865829 208562345 0.87152001 0.0000000  
## 7 2028 7.0e+07 67446867 204011039 0.87998454 0.0000000  
## 8 2029 8.0e+07 76606276 211175925 0.86643266 0.0000000  
## 9 2030 8.0e+07 76133356 206565701 0.87529437 0.0000000  
## 10 2031 8.0e+07 75663355 202060527 0.88346228 0.0000000  
## Prob\_Top\_10\_Not\_Place Prob\_Not\_Place\_Adj Y2022 Y2023 Y2024 Y2025 Y2026 Y2027  
## 1 0.2076232 -1.46907701 1 0 0 0 0 0  
## 2 0.1258515 -0.01226403 0 1 0 0 0 0  
## 3 0.2133259 -1.44114576 0 0 1 0 0 0  
## 4 0.1099093 -0.01048566 0 0 0 1 0 0  
## 5 0.1255083 -0.01222493 0 0 0 0 1 0  
## 6 0.1172487 -0.01129489 0 0 0 0 0 1  
## 7 0.1096162 -0.01045368 0 0 0 0 0 0  
## 8 0.1218287 -0.01180804 0 0 0 0 0 0  
## 9 0.1138471 -0.01091787 0 0 0 0 0 0  
## 10 0.1064762 -0.01011252 0 0 0 0 0 0  
## Y2028 Y2029 Y2030 Y2031 picked ROI  
## 1 0 0 0 0 1 6%  
## 2 0 0 0 0 1 6%  
## 3 0 0 0 0 1 6%  
## 4 0 0 0 0 1 6%  
## 5 0 0 0 0 1 6%  
## 6 0 0 0 0 1 6%  
## 7 1 0 0 0 1 6%  
## 8 0 1 0 0 1 6%  
## 9 0 0 1 0 1 6%  
## 10 0 0 0 1 1 6%

ROI\_8pc

## Year Salary Salary\_NPV\_2022 Expected\_Value\_2021 Prob\_Bottom Prob\_Bottom\_Adj  
## 1 2022 5.0e+08 500000000 541074245 0.02251459 -3.7935916  
## 2 2023 6.0e+07 58525345 213405350 0.86195936 -0.1485472  
## 3 2024 5.5e+08 523296892 538943316 0.02333054 -3.7579921  
## 4 2025 6.0e+07 55683875 204190667 0.87965980 -0.1282200  
## 5 2026 7.0e+07 63367850 213217443 0.86234119 -0.1481043  
## 6 2027 7.0e+07 61810422 208562345 0.87152001 0.0000000  
## 7 2028 7.0e+07 60291271 204011039 0.87998454 0.0000000  
## 8 2029 8.0e+07 67210808 211175925 0.86643266 0.0000000  
## 9 2030 8.0e+07 65558929 206565701 0.87529437 0.0000000  
## 10 2031 8.0e+07 63947649 202060527 0.88346228 0.0000000  
## Prob\_Top\_10\_Not\_Place Prob\_Not\_Place\_Adj Y2022 Y2023 Y2024 Y2025 Y2026 Y2027  
## 1 0.2076232 -1.46907701 1 0 0 0 0 0  
## 2 0.1258515 -0.01226403 0 1 0 0 0 0  
## 3 0.2133259 -1.44114576 0 0 1 0 0 0  
## 4 0.1099093 -0.01048566 0 0 0 1 0 0  
## 5 0.1255083 -0.01222493 0 0 0 0 1 0  
## 6 0.1172487 -0.01129489 0 0 0 0 0 1  
## 7 0.1096162 -0.01045368 0 0 0 0 0 0  
## 8 0.1218287 -0.01180804 0 0 0 0 0 0  
## 9 0.1138471 -0.01091787 0 0 0 0 0 0  
## 10 0.1064762 -0.01011252 0 0 0 0 0 0  
## Y2028 Y2029 Y2030 Y2031 picked ROI  
## 1 0 0 0 0 1 8%  
## 2 0 0 0 0 1 8%  
## 3 0 0 0 0 1 8%  
## 4 0 0 0 0 1 8%  
## 5 0 0 0 0 1 8%  
## 6 0 0 0 0 1 8%  
## 7 1 0 0 0 1 8%  
## 8 0 1 0 0 1 8%  
## 9 0 0 1 0 1 8%  
## 10 0 0 0 1 1 8%

ROI\_10pc

## Year Salary Salary\_NPV\_2022 Expected\_Value\_2021 Prob\_Bottom Prob\_Bottom\_Adj  
## 1 2022 5.0e+07 50000000 201948053 0.8836601 -0.1236828  
## 2 2023 5.4e+08 517151231 547058100 0.0203696 -3.8937115  
## 3 2024 6.0e+07 55029917 208746172 0.8711677 -0.1379208  
## 4 2025 6.0e+07 52701462 204190667 0.8796598 -0.1282200  
## 5 2026 6.0e+08 504715293 534814553 0.0249944 -3.6891035  
## 6 2027 7.0e+07 56391943 208562345 0.8715200 0.0000000  
## 7 2028 7.0e+07 54005857 204011039 0.8799845 0.0000000  
## 8 2029 8.0e+07 59109408 211175925 0.8664327 0.0000000  
## 9 2030 8.0e+07 56608339 206565701 0.8752944 0.0000000  
## 10 2031 8.0e+07 54213097 202060527 0.8834623 0.0000000  
## Prob\_Top\_10\_Not\_Place Prob\_Not\_Place\_Adj Y2022 Y2023 Y2024 Y2025 Y2026 Y2027  
## 1 0.1062974 -0.01009319 1 0 0 0 0 0  
## 2 0.1921431 -1.54875328 0 1 0 0 0 0  
## 3 0.1175660 -0.01133025 0 0 1 0 0 0  
## 4 0.1099093 -0.01048566 0 0 0 1 0 0  
## 5 0.2246509 -1.38771431 0 0 0 0 1 0  
## 6 0.1172487 -0.01129489 0 0 0 0 0 1  
## 7 0.1096162 -0.01045368 0 0 0 0 0 0  
## 8 0.1218287 -0.01180804 0 0 0 0 0 0  
## 9 0.1138471 -0.01091787 0 0 0 0 0 0  
## 10 0.1064762 -0.01011252 0 0 0 0 0 0  
## Y2028 Y2029 Y2030 Y2031 picked ROI  
## 1 0 0 0 0 1 10%  
## 2 0 0 0 0 1 10%  
## 3 0 0 0 0 1 10%  
## 4 0 0 0 0 1 10%  
## 5 0 0 0 0 1 10%  
## 6 0 0 0 0 1 10%  
## 7 1 0 0 0 1 10%  
## 8 0 1 0 0 1 10%  
## 9 0 0 1 0 1 10%  
## 10 0 0 0 1 1 10%

ROI\_12pc

## Year Salary Salary\_NPV\_2022 Expected\_Value\_2021 Prob\_Bottom Prob\_Bottom\_Adj  
## 1 2022 5e+07 50000000 201948053 0.88366013 -0.1236828  
## 2 2023 6e+07 56435154 213405350 0.86195936 -0.1485472  
## 3 2024 6e+07 53082111 208746172 0.87116766 -0.1379208  
## 4 2025 6e+08 499282849 547347472 0.02027111 -3.8985586  
## 5 2026 6e+08 469618410 534814553 0.02499440 -3.6891035  
## 6 2027 7e+07 51533587 208562345 0.87152001 0.0000000  
## 7 2028 7e+07 48471765 204011039 0.87998454 0.0000000  
## 8 2029 8e+07 52104982 211175925 0.86643266 0.0000000  
## 9 2030 8e+07 49009211 206565701 0.87529437 0.0000000  
## 10 2031 8e+07 46097373 202060527 0.88346228 0.0000000  
## Prob\_Top\_10\_Not\_Place Prob\_Not\_Place\_Adj Y2022 Y2023 Y2024 Y2025 Y2026 Y2027  
## 1 0.1062974 -0.01009319 1 0 0 0 0 0  
## 2 0.1258515 -0.01226403 0 1 0 0 0 0  
## 3 0.1175660 -0.01133025 0 0 1 0 0 0  
## 4 0.1914148 -1.55265155 0 0 0 1 0 0  
## 5 0.2246509 -1.38771431 0 0 0 0 1 0  
## 6 0.1172487 -0.01129489 0 0 0 0 0 1  
## 7 0.1096162 -0.01045368 0 0 0 0 0 0  
## 8 0.1218287 -0.01180804 0 0 0 0 0 0  
## 9 0.1138471 -0.01091787 0 0 0 0 0 0  
## 10 0.1064762 -0.01011252 0 0 0 0 0 0  
## Y2028 Y2029 Y2030 Y2031 picked ROI  
## 1 0 0 0 0 1 12%  
## 2 0 0 0 0 1 12%  
## 3 0 0 0 0 1 12%  
## 4 0 0 0 0 1 12%  
## 5 0 0 0 0 1 12%  
## 6 0 0 0 0 1 12%  
## 7 1 0 0 0 1 12%  
## 8 0 1 0 0 1 12%  
## 9 0 0 1 0 1 12%  
## 10 0 0 0 1 1 12%

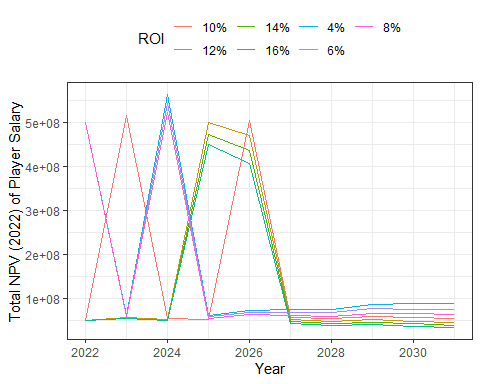
ROI\_14pc

## Year Salary Salary\_NPV\_2022 Expected\_Value\_2021 Prob\_Bottom Prob\_Bottom\_Adj  
## 1 2022 5e+07 50000000 201948053 0.88366013 -0.1236828  
## 2 2023 6e+07 55445064 213405350 0.86195936 -0.1485472  
## 3 2024 6e+07 51235918 208746172 0.87116766 -0.1379208  
## 4 2025 6e+08 473463127 547347472 0.02027111 -3.8985586  
## 5 2026 6e+08 437519888 534814553 0.02499440 -3.6891035  
## 6 2027 7e+07 47168952 208562345 0.87152001 0.0000000  
## 7 2028 7e+07 43588092 204011039 0.87998454 0.0000000  
## 8 2029 8e+07 46033230 211175925 0.86643266 0.0000000  
## 9 2030 8e+07 42538589 206565701 0.87529437 0.0000000  
## 10 2031 8e+07 39309247 202060527 0.88346228 0.0000000  
## Prob\_Top\_10\_Not\_Place Prob\_Not\_Place\_Adj Y2022 Y2023 Y2024 Y2025 Y2026 Y2027  
## 1 0.1062974 -0.01009319 1 0 0 0 0 0  
## 2 0.1258515 -0.01226403 0 1 0 0 0 0  
## 3 0.1175660 -0.01133025 0 0 1 0 0 0  
## 4 0.1914148 -1.55265155 0 0 0 1 0 0  
## 5 0.2246509 -1.38771431 0 0 0 0 1 0  
## 6 0.1172487 -0.01129489 0 0 0 0 0 1  
## 7 0.1096162 -0.01045368 0 0 0 0 0 0  
## 8 0.1218287 -0.01180804 0 0 0 0 0 0  
## 9 0.1138471 -0.01091787 0 0 0 0 0 0  
## 10 0.1064762 -0.01011252 0 0 0 0 0 0  
## Y2028 Y2029 Y2030 Y2031 picked ROI  
## 1 0 0 0 0 1 14%  
## 2 0 0 0 0 1 14%  
## 3 0 0 0 0 1 14%  
## 4 0 0 0 0 1 14%  
## 5 0 0 0 0 1 14%  
## 6 0 0 0 0 1 14%  
## 7 1 0 0 0 1 14%  
## 8 0 1 0 0 1 14%  
## 9 0 0 1 0 1 14%  
## 10 0 0 0 1 1 14%

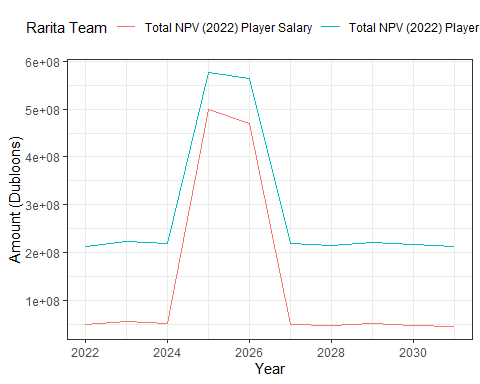
ROI\_16pc

## Year Salary Salary\_NPV\_2022 Expected\_Value\_2021 Prob\_Bottom Prob\_Bottom\_Adj  
## 1 2022 5e+07 50000000 201948053 0.88366013 -0.1236828  
## 2 2023 6e+07 54489114 213405350 0.86195936 -0.1485472  
## 3 2024 6e+07 49484393 208746172 0.87116766 -0.1379208  
## 4 2025 6e+08 449393460 547347472 0.02027111 -3.8985586  
## 5 2026 6e+08 408117528 534814553 0.02499440 -3.6891035  
## 6 2027 7e+07 43240483 208562345 0.87152001 0.0000000  
## 7 2028 7e+07 39268927 204011039 0.87998454 0.0000000  
## 8 2029 8e+07 40756744 211175925 0.86643266 0.0000000  
## 9 2030 8e+07 37013315 206565701 0.87529437 0.0000000  
## 10 2031 8e+07 33613712 202060527 0.88346228 0.0000000  
## Prob\_Top\_10\_Not\_Place Prob\_Not\_Place\_Adj Y2022 Y2023 Y2024 Y2025 Y2026 Y2027  
## 1 0.1062974 -0.01009319 1 0 0 0 0 0  
## 2 0.1258515 -0.01226403 0 1 0 0 0 0  
## 3 0.1175660 -0.01133025 0 0 1 0 0 0  
## 4 0.1914148 -1.55265155 0 0 0 1 0 0  
## 5 0.2246509 -1.38771431 0 0 0 0 1 0  
## 6 0.1172487 -0.01129489 0 0 0 0 0 1  
## 7 0.1096162 -0.01045368 0 0 0 0 0 0  
## 8 0.1218287 -0.01180804 0 0 0 0 0 0  
## 9 0.1138471 -0.01091787 0 0 0 0 0 0  
## 10 0.1064762 -0.01011252 0 0 0 0 0 0  
## Y2028 Y2029 Y2030 Y2031 picked ROI  
## 1 0 0 0 0 1 16%  
## 2 0 0 0 0 1 16%  
## 3 0 0 0 0 1 16%  
## 4 0 0 0 0 1 16%  
## 5 0 0 0 0 1 16%  
## 6 0 0 0 0 1 16%  
## 7 1 0 0 0 1 16%  
## 8 0 1 0 0 1 16%  
## 9 0 0 1 0 1 16%  
## 10 0 0 0 1 1 16%

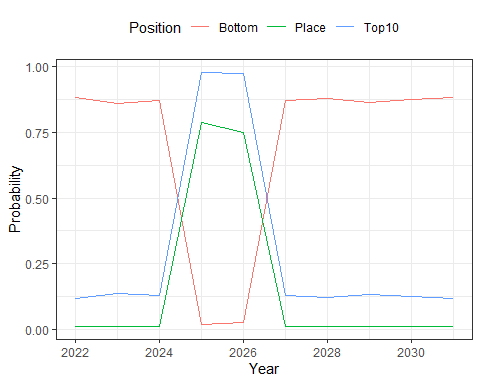
combine <- rbind(rbind(rbind(rbind(rbind(rbind(ROI\_4pc,ROI\_6pc),ROI\_8pc),ROI\_10pc),ROI\_12pc),ROI\_14pc),ROI\_16pc)  
  
#Figure ## -   
combine %>%  
 group\_by(ROI) %>%  
 rename(`Total NPV (2022) of Player Salary` = "Salary\_NPV\_2022") %>%  
 ggplot(aes(x=Year, y=`Total NPV (2022) of Player Salary`, colour=ROI))+  
 geom\_line()



# Figure ## - Total NPV of Rarita Player Salary and Value for the Next 10 Years and the Corresponding Probabilities for Tournament Position   
ROI\_12pc %>%  
 dplyr::select(Year, Salary\_NPV\_2022, Expected\_Value\_2021) %>%  
 mutate(Expected\_Value\_2021 = Expected\_Value\_2021\*as.numeric(Super\_Inflation))%>%  
 rename(`Total NPV (2022) Player Salary` = "Salary\_NPV\_2022") %>%  
 rename(`Total NPV (2022) Player Value` = "Expected\_Value\_2021")%>%  
 pivot\_longer(!Year, names\_to = "Rarita Team", values\_to = "Salary")%>%  
 ggplot(aes(x=Year, y=Salary, color=`Rarita Team`))+   
 geom\_line() +  
 labs(y="Amount (Dubloons)")



ROI\_12pc %>%  
 dplyr::select(Year, Prob\_Bottom, Prob\_Top\_10\_Not\_Place) %>%  
 mutate(Place = 1-Prob\_Bottom-Prob\_Top\_10\_Not\_Place)%>%  
 mutate(Prob\_Top\_10\_Not\_Place = 1-Prob\_Bottom)%>%  
 rename(Bottom = "Prob\_Bottom") %>%  
 rename(Top10 = "Prob\_Top\_10\_Not\_Place")%>%  
 pivot\_longer(!Year, names\_to = "Position", values\_to = "Probability")%>%  
 ggplot(aes(x=Year, y=Probability, color=Position))+   
 geom\_line()



## Create Team

* End function just requires the pre-determined budget for year 2022 and the squad will be selected.
* Function allows salary paid in 2022 to be input directly to find team.
* As per the strategy the near cheapest team (with best possible value) will be selected (spend 50,000,000 Dubloons)

## Model  
  
Final\_Rarita\_Selection <- function(x){  
Opt <- Rarita\_Squad %>%  
 mutate(values = 1) %>%  
 pivot\_wider(names\_from = pos1, values\_from = values, values\_fill = 0) %>%  
 mutate(FW\_Sal = salary\*FW,  
 MF\_Sal = salary\*MF,  
 DF\_Sal = salary\*DF)  
  
  
Max\_Salary <- x/as.numeric(Super\_Inflation)  
Max\_Prop\_DF <- Max\_Salary\*DF\_Prop+0.05\*Max\_Salary  
Max\_Prop\_FW <- Max\_Salary\*FW\_Prop+0.05\*Max\_Salary  
Max\_Prop\_MF <- Max\_Salary\*MF\_Prop+0.05\*Max\_Salary  
Min\_Prop\_DF <- Max\_Salary\*DF\_Prop-0.05\*Max\_Salary  
Min\_Prop\_FW <- Max\_Salary\*FW\_Prop-0.05\*Max\_Salary  
Min\_Prop\_MF <- Max\_Salary\*MF\_Prop-0.05\*Max\_Salary  
Max\_DF <- 8  
Max\_FW <- 8  
Max\_MF <- 8  
Min\_DF <- 5  
Min\_FW <- 5  
Min\_MF <- 5  
Min\_GK <- 1  
  
## Set the coefficients of the decision variables -> C  
C <- Opt$model\_salary  
  
# Create constraint martix B  
A <- matrix(c(Opt$salary,  
 Opt$DF\_Sal,  
 Opt$FW\_Sal,  
 Opt$MF\_Sal,  
 Opt$DF\_Sal,  
 Opt$FW\_Sal,  
 Opt$MF\_Sal,  
 Opt$DF,  
 Opt$FW,  
 Opt$MF,  
 Opt$DF,  
 Opt$FW,  
 Opt$MF,  
 Opt$GK), nrow=14, byrow=TRUE)  
  
# Right hand side for the constraints  
B <- c(Max\_Salary, Max\_Prop\_DF, Max\_Prop\_FW, Max\_Prop\_MF, Min\_Prop\_DF, Min\_Prop\_FW, Min\_Prop\_MF, Max\_DF, Max\_FW, Max\_MF, Min\_DF, Min\_FW, Min\_MF, Min\_GK)  
  
# Direction of the constraints  
constranints\_direction <- c( "<=","<=", "<=", "<=", ">=", ">=", ">=","<=", "<=", "<=", ">=", ">=", ">=", "=")  
  
# Find the optimal solution  
optimum <- lp(direction="max",  
 objective.in = C,  
 const.mat = A,  
 const.dir = constranints\_direction,  
 const.rhs = B,  
 all.bin = TRUE)  
  
  
Result <- cbind(Opt,optimum$solution) %>%  
 rename(picked = "optimum$solution") %>%  
 dplyr::filter(picked == 1)  
  
  
return(Result)  
  
}  
  
Picked\_Team <- Final\_Rarita\_Selection(50000000) %>%  
 dplyr::select(player, salary, model\_salary, DF, FW, MF, GK) %>%  
 pivot\_longer(c("DF", "FW", "MF", "GK"), names\_to = "Primary Position", values\_to = "Values") %>%  
 dplyr::filter(Values == 1) %>%  
 dplyr:: select(-Values) %>%  
 mutate(salary = salary\*as.numeric(Super\_Inflation))%>%  
 dplyr::select(player, `Primary Position`, salary, model\_salary)%>%  
 rename(`Salary (Dubloons)` = "salary") %>%  
 rename(Player = "player") %>%  
 rename(`Value (Dubloons)` = "model\_salary")  
  
sum(Picked\_Team$`Salary (Dubloons)`) #Proof total salary in 2022 is under 50,000,000 Dubloons

## [1] 49957117