Sports_Analysis

Sergio Abbate

15/6/2021

1. Read the file

```
library(readxl)
df = read_xlsx("EPL 2018-2019 Performance Stats (FinalData) Excel.xlsx")
head(df)
```

```
## # A tibble: 6 x 49
##   full_name   age birthday birthday_GMT        league season position `Current Club`
##   <chr>     <dbl>    <dbl> <dttm>              <chr>  <chr>  <chr>    <chr>
## 1 Aaron Cr~    31   6.30e8 1989-12-15 00:00:00 Premi~ 2018/~ Defender West Ham Unit~
## 2 Aaron Le~    33   5.46e8 1987-04-16 00:00:00 Premi~ 2018/~ Midfiel~ Burnley
## 3 Aaron Wa~    23   8.81e8 1997-11-26 00:00:00 Premi~ 2018/~ Midfiel~ Crystal Palace
## 4 Abdoulay~    28   7.26e8 1993-01-01 00:00:00 Premi~ 2018/~ Midfiel~ Watford
## 5 Adalbert~    23   8.65e8 1997-05-31 00:00:00 Premi~ 2018/~ Forward  Watford
## 6 Adam Dav~    32   5.79e8 1988-05-10 00:00:00 Premi~ 2018/~ Midfiel~ Liverpool
## # ... with 41 more variables: minutes_played_overall <dbl>,
## #   minutes_played_home <dbl>, minutes_played_away <dbl>, nationality <chr>,
## #   appearances_overall <dbl>, appearances_home <dbl>, appearances_away <dbl>,
## #   goals_overall <dbl>, goals_home <dbl>, goals_away <dbl>, assists_overall <dbl>,
## #   assists_home <dbl>, assists_away <dbl>, penalty_goals <dbl>,
## #   penalty_misses <dbl>, clean_sheets_overall <dbl>, clean_sheets_home <dbl>,
## #   clean_sheets_away <dbl>, conceded_overall <dbl>, conceded_home <dbl>,
## #   conceded_away <dbl>, yellow_cards_overall <dbl>, red_cards_overall <dbl>,
## #   goals_involved_per_90_overall <chr>, assists_per_90_overall <chr>,
## #   goals_per_90_overall <chr>, goals_per_90_home <chr>, goals_per_90_away <chr>,
## #   min_per_goal_overall <dbl>, conceded_per_90_overall <chr>,
## #   min_per_conceded_overall <dbl>, min_per_match <dbl>,
## #   min_per_card_overall <dbl>, min_per_assist_overall <dbl>,
## #   cards_per_90_overall <chr>, rank_in_league_top_attackers <dbl>,
## #   rank_in_league_top_midfielders <dbl>, rank_in_league_top_defenders <dbl>,
## #   rank_in_club_top_scorer <dbl>, annual_salary <dbl>, weekly_salary <dbl>
```

Define and Use 2 functions to summarize the dataset and check for Quality issues

```
#Define the functions

summarize_factor = function(dataset) {

  dataset = select_if(dataset, is.factor)
  summary.table = data.frame(Attribute = names(dataset))

  summary.table = summary.table %>%
    mutate('Missing Values' = apply(dataset, 2, function (x) sum(is.na(x))),
           'Unique Values' = apply(dataset, 2, function (x) length(unique(x))),
    )
  summary.table
}

summarize_numeric = function(dataset) {

  dataset = select_if(dataset, is.numeric)
  summary.table = data.frame(Attribute = names(dataset))

  summary.table = summary.table %>%
    mutate('Missing Values' = apply(dataset, 2, function (x) sum(is.na(x))),
           'Unique Values' = apply(dataset, 2, function (x) length(unique(x))),
           'Mean' = colMeans(dataset, na.rm = TRUE),
           'Min' = apply(dataset, 2, function (x) min(x, na.rm = TRUE)),
           'Max' = apply(dataset, 2, function (x) max(x, na.rm = TRUE)),
           'SD' = apply(dataset, 2, function (x) sd(x, na.rm = TRUE))
    )
  summary.table
}
```

3. Drop unused columns

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
df = df%>% select(-birthday, -league, -season, -nationality, -birthday_GMT)
```

4. Ajust Column Types

```
df$position = factor(df$position, order = TRUE, levels = c("Goalkeeper", "Defender", "Midfiel
der", "Forward"))
df$`Current Club` = factor(df$`Current Club`)
df$goals_involved_per_90_overall = as.numeric(df$goals_involved_per_90_overall)
df$assists_per_90_overall = as.numeric(df$assists_per_90_overall)
df$goals_per_90_overall = as.numeric(df$goals_per_90_overall)
df$goals_per_90_home = as.numeric(df$goals_per_90_home)
df$goals_per_90_away = as.numeric(df$goals_per_90_away)
df$conceded_per_90_overall = as.numeric(df$conceded_per_90_overall)
df$cards_per_90_overall = as.numeric(df$cards_per_90_overall)
df$assists_per_90_overall = as.numeric(df$assists_per_90_overall)
```

5. Check for Quality Issues

```
format(summarize_numeric(df), scientific = FALSE)
```

```
##                           Attribute Missing Values Unique Values             Mean
## 1                               age              0            22      28.54934211
## 2             minutes_played_overall              0           264    1732.60197368
## 3                minutes_played_home              0           239     870.46710526
## 4                minutes_played_away              0           233     862.13486842
## 5                appearances_overall              0            39      23.14802632
## 6                   appearances_home              0            20      11.52302632
## 7                   appearances_away              0            20      11.62500000
## 8                      goals_overall              0            18       2.54605263
## 9                         goals_home              0            15       1.35855263
## 10                        goals_away              0            11       1.18750000
## 11                   assists_overall              0            13       1.73684211
## 12                      assists_home              0             9       0.94078947
## 13                      assists_away              0             7       0.79605263
## 14                     penalty_goals              0             7       0.22039474
## 15                    penalty_misses              0             3       0.04605263
## 16              clean_sheets_overall              0            22       6.86184211
## 17                 clean_sheets_home              0            13       3.92434211
## 18                 clean_sheets_away              0            12       2.93750000
## 19                  conceded_overall              0            54      22.42434211
## 20                     conceded_home              0            28       9.93421053
## 21                     conceded_away              0            34      12.49013158
## 22              yellow_cards_overall              0            15       3.00000000
## 23                 red_cards_overall              0             3       0.11842105
## 24  goals_involved_per_90_overall              0            77       0.21506579
## 25          assists_per_90_overall              0            39       0.09006579
## 26            goals_per_90_overall              0            56       0.12516447
## 27               goals_per_90_home              0            56       0.12450658
## 28               goals_per_90_away              0            52       0.11588816
## 29             min_per_goal_overall              0           169     566.48355263
## 30          conceded_per_90_overall              0           122       1.10023026
## 31         min_per_conceded_overall              0            97      80.38486842
## 32                    min_per_match              0            65      66.97697368
## 33             min_per_card_overall              0           216     608.43421053
## 34           min_per_assist_overall              0           160     564.67763158
## 35             cards_per_90_overall              0            55       0.15914474
## 36    rank_in_league_top_attackers              0           270     183.34210526
## 37 rank_in_league_top_midfielders              0           270     178.39802632
## 38    rank_in_league_top_defenders              0           122      28.34210526
## 39          rank_in_club_top_scorer              0            29      11.23355263
## 40                    annual_salary              0            76 3683430.72039474
## 41                    weekly_salary              0            80  103608.10855263
##       Min         Max            SD
## 1      19       40.00     3.7973675
## 2       0     3420.00  1057.7107493
## 3       0     1710.00   537.9703627
## 4       0     1710.00   532.3947810
## 5       0       38.00    11.8160179
## 6       0       19.00     5.9839827
## 7       0       19.00     6.0140786
## 8       0       22.00     4.1471909
## 9       0       18.00     2.5562437
## 10      0       11.00     1.9568889
## 11      0       12.00     2.3328246
## 12      0        9.00     1.4498597
## 13      0        6.00     1.1931533
## 14      0       10.00     0.9478716
```

```
## 15     0        3.00       0.2892762
## 16     0       21.00       5.0023748
## 17     0       12.00       2.9401398
## 18     0       11.00       2.5118941
## 19     0       66.00      15.3401757
## 20     0       28.00       7.2144621
## 21     0       43.00       8.7762751
## 22     0       15.00       2.9236715
## 23     0        2.00       0.3434288
## 24     0        1.48       0.2676705
## 25     0        1.48       0.1380773
## 26     0        1.45       0.1970391
## 27     0        1.43       0.2160440
## 28     0        1.55       0.1988794
## 29     0     3403.00     791.9675751
## 30     0        4.29       0.5197355
## 31     0      353.00      49.0785714
## 32     0       90.00      24.6672134
## 33     0     3420.00     660.6430628
## 34     0     3420.00     758.0675893
## 35     0        1.43       0.1619821
## 36    -1      418.00     132.9908359
## 37    -1      419.00     130.2801607
## 38    -1      163.00      45.3490468
## 39    -1       28.00       7.6024179
## 40 36000 19500000.00 2886953.5130781
## 41   692  3120000.00  290006.8469620
```

```
format(summarize_factor(df), scientific = FALSE)
```

```
##      Attribute Missing Values Unique Values
## 1     position              0             4
## 2 Current Club              0            17
```

There are No Missing Values in any of the columns

Ranking columns

```
df %>% select(rank_in_club_top_scorer, rank_in_league_top_attackers, rank_in_league_top_midfi
elders, rank_in_league_top_defenders) %>% summarize_numeric()
```

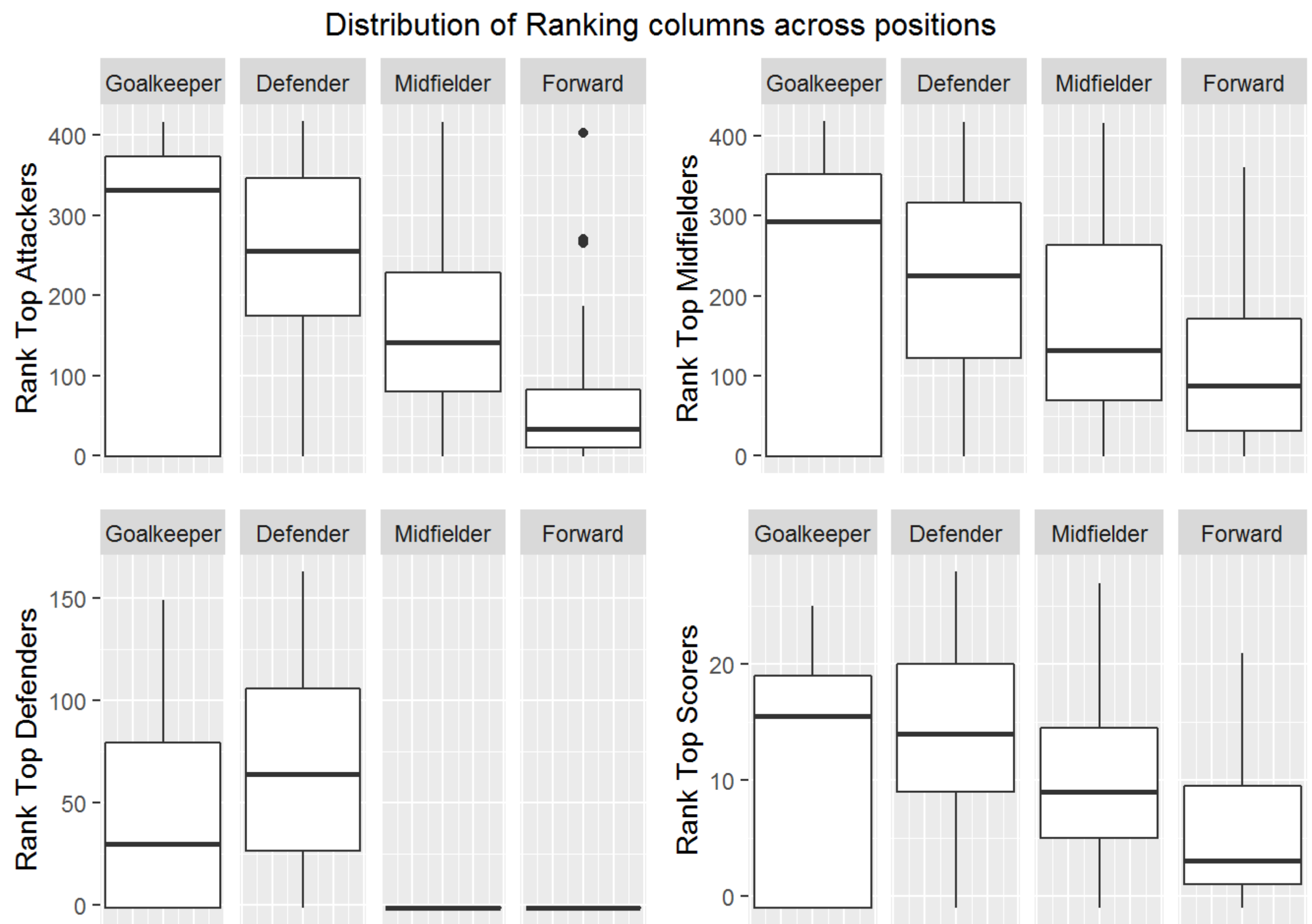
```
##                       Attribute Missing Values Unique Values      Mean Min Max
## 1        rank_in_club_top_scorer              0            29  11.23355  -1  28
## 2   rank_in_league_top_attackers              0           270 183.34211  -1 418
## 3 rank_in_league_top_midfielders              0           270 178.39803  -1 419
## 4   rank_in_league_top_defenders              0           122  28.34211  -1 163
##           SD
## 1   7.602418
## 2 132.990836
## 3 130.280161
## 4  45.349047
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
g1 = ggplot(df) + geom_boxplot(aes(y = rank_in_league_top_attackers)) + facet_grid(~position)
+ theme(axis.ticks.x = element_blank(),axis.text.x = element_blank()) + ylab("Rank Top Attack
ers")
g2 = ggplot(df) + geom_boxplot(aes(y = rank_in_league_top_midfielders)) + facet_grid(~positio
n) + theme(axis.ticks.x = element_blank(),axis.text.x = element_blank()) + ylab("Rank Top Mid
fielders")
g3 = ggplot(df) + geom_boxplot(aes(y = rank_in_league_top_defenders)) + facet_grid(~position)
+ theme(axis.ticks.x = element_blank(),axis.text.x = element_blank()) + ylab("Rank Top Defend
ers")
g4 = ggplot(df) + geom_boxplot(aes(y = rank_in_club_top_scorer)) + facet_grid(~position) + th
eme(axis.ticks.x = element_blank(),axis.text.x = element_blank()) + ylab("Rank Top Scorers")

grid.arrange(g1,g2,g3,g4 , nrow = 2, top =  "Distribution of Ranking columns across position
s")
```



Just drop the ranking columns

```
df  = df %>% select(-rank_in_club_top_scorer, -rank_in_league_top_attackers, -rank_in_league_
top_midfielders, -rank_in_league_top_defenders)
```

6. Divide dataset into 4 subsets: Goalkeeper GK, Defender DF, Midfielder MD, Forward F

```
goalkeepers = df %>% filter(position == "Goalkeeper")
defenders = df %>% filter(position == "Defender")
midfielders = df %>% filter(position == "Midfielder")
forwards = df %>% filter(position == "Forward")
```

7. Subselect the appropiate columns to analyze each position

For the Goalkeepers, we are not interested in the goals scored or assist, but in the clean sheets, goals conceded and related stats.

```
goalkeepers = goalkeepers %>% select(-goals_overall, -goals_home, -goals_away, -assists_overa
ll, -assists_home, -assists_away, -penalty_goals, -penalty_misses, -goals_involved_per_90_ove
rall, -assists_per_90_overall, -goals_per_90_away, -goals_per_90_overall, -goals_per_90_home,
-min_per_goal_overall, min_per_assist_overall)
```

For the Defenders, usually they do not score Goals or provide many assists, but some of them do, so we will keep these columns and all related columns, since they are a measure of performance and somehow can affect the Salary. However the most important columns to analyze a defender's performance are: - Clean Sheets - Goals Conceded - Yellow and Red Cards - Rank

```
### Defenders ###
defenders = defenders %>% select(-penalty_goals, -penalty_misses)
```

For the Midfielders, actually we are interested in almost all the features, because there are some offensive and defensive midfielders. We will leave most of the features.

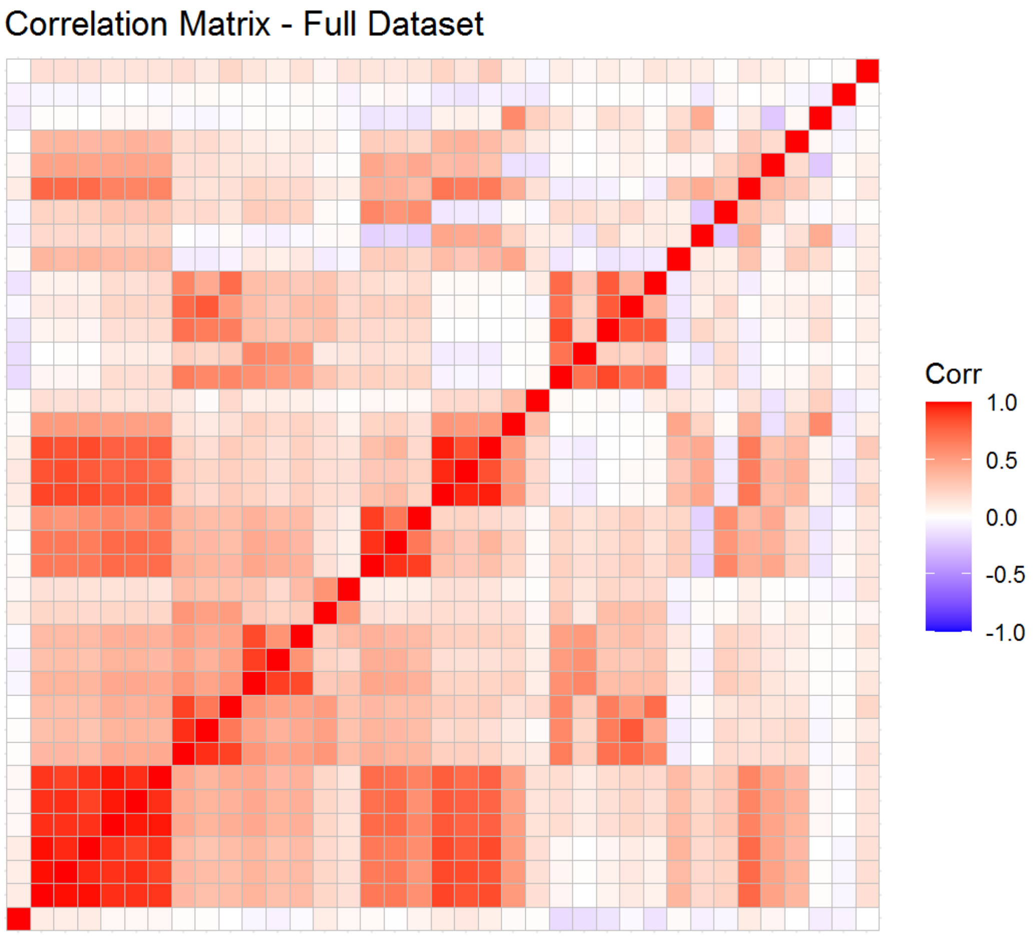
```
### Midfielders ###
```

For Forwards, we are mostly interested in the measures of goals, assists and offensive characteristics, not clean sheets or goals conceeded.

```
### Forwards ###
forwards = forwards %>% select(-clean_sheets_overall, -clean_sheets_home, -clean_sheets_away,
-conceded_overall, -conceded_home, -conceded_away, -conceded_per_90_overall, -min_per_concede
d_overall)
```

General Correlation

```
library(ggcorrplot)
General_CorrMat = round(cor(df %>% select_if(is.numeric)),2)
ggcorrplot(General_CorrMat) + theme(axis.text.x = element_blank())+ theme(axis.text.y = eleme
nt_blank()) + ggtitle("Correlation Matrix - Full Dataset")
```



Specifics about overall vs home and away

```
f1 = round(cor(df %>% select(appearances_overall,appearances_home, appearances_away)),2) %>%
 ggcorrplot(lab = TRUE)
f2 = round(cor(df %>% select(assists_overall,assists_home, assists_away)),2) %>% ggcorrplot(l
ab = TRUE)
f3 = round(cor(df %>% select(goals_overall,goals_home, goals_away)),2) %>% ggcorrplot(lab = T
RUE)
f4 = round(cor(df %>% select(clean_sheets_overall,clean_sheets_home, clean_sheets_away)),2) %
>% ggcorrplot(lab = TRUE)
```

8. Get rid of multicolinearity

In general, for the purpose of this analysis, there is no need or sense to discriminate goals, assists, cards in away or home, so the column with 'Overall' will do. In fact the 2 columns of away and home are directly related to the column overall. We will look into details into the correlation and decide which columns to drop.

There are too many correlated columns, we will take a deeper look

```
goalkeepers %>% select(appearances_overall,appearances_home, appearances_away) %>% cor()
```

```
##                     appearances_overall appearances_home appearances_away
## appearances_overall           1.0000000        0.9977919        0.9977063
## appearances_home              0.9977919        1.0000000        0.9910073
## appearances_away              0.9977063        0.9910073        1.0000000
```

```
midfielders %>% select(assists_overall,assists_home, assists_away) %>% cor()
```

```
##                 assists_overall assists_home assists_away
## assists_overall       1.0000000    0.8628558    0.8655604
## assists_home          0.8628558    1.0000000    0.4937221
## assists_away          0.8655604    0.4937221    1.0000000
```

```
forwards %>% select(goals_overall,goals_home,goals_away) %>% cor()
```

```
##               goals_overall goals_home goals_away
## goals_overall     1.0000000  0.9386144  0.8624746
## goals_home        0.9386144  1.0000000  0.6349424
## goals_away        0.8624746  0.6349424  1.0000000
```

But we actually know from subject knowledge that this relationship holds: Overall = Away + Home, so all the columns 'overall' are composed from away + home. It is reasonable to drop this discrimination and to keep only the overall columns

```
# They are the same: Relationship holds
v1 = goalkeepers$minutes_played_away + goalkeepers$minutes_played_home
v2 = goalkeepers$minutes_played_overall
tail(matrix(c(v1, v2), ncol = 2))
```

```
##       [,1] [,2]
## [27,] 3330 3330
## [28,]    0    0
## [29,] 1755 1755
## [30,] 1575 1575
## [31,]  180  180
## [32,] 3420 3420
```

```
# They are the same: Relationship holds
v1 = forwards$goals_overall
v2 = forwards$goals_home + forwards$goals_away
head(matrix(c(v1, v2), ncol = 2))
```

```
##      [,1] [,2]
## [1,]    0    0
## [2,]    3    3
## [3,]   13   13
## [4,]    0    0
## [5,]    7    7
## [6,]    2    2
```

```r
## Remove all the Away and Home columns and just leave the overall columns ##

## Goalkeepers

goalkeepers = goalkeepers %>% select(-minutes_played_home, -minutes_played_away, -appearances
_home, -appearances_away, -clean_sheets_home, -clean_sheets_away, -conceded_home, -conceded_a
way, -min_per_assist_overall)

## Defenders

defenders = defenders %>% select(-minutes_played_home, -minutes_played_away, -appearances_hom
e, -appearances_away, -goals_home, -goals_away, -assists_home, -assists_away, -clean_sheets_h
ome, -clean_sheets_away, -conceded_home, -conceded_away, -goals_per_90_home, -goals_per_90_aw
ay)

## Midfielders

midfielders = midfielders %>% select(-minutes_played_home, -minutes_played_away, -appearances
_home, -appearances_away, -goals_home, -goals_away, -assists_home, -assists_away, -clean_shee
ts_home, -clean_sheets_away, -conceded_home, -conceded_away, -goals_per_90_home, -goals_per_9
0_away)

## Forwards

forwards = forwards %>% select(-minutes_played_home, -minutes_played_away, -appearances_home,
-appearances_away, -goals_home, -goals_away, -assists_home, -assists_away, -goals_per_90_hom
e, -goals_per_90_away)
```

```r
#Vector of correlations btw minutes played overall and appearances overall

c(cor(df$minutes_played_overall, df$appearances_overall),cor(goalkeepers$minutes_played_overa
ll, goalkeepers$appearances_overall),cor(defenders$minutes_played_overall, defenders$appearan
ces_overall),cor(midfielders$minutes_played_overall, midfielders$appearances_overall),cor(for
wards$minutes_played_overall, forwards$appearances_overall))
```

```
## [1] 0.9367994 0.9999621 0.9873110 0.9263244 0.8961652
```

```r
# Too high correlations - We will drop appearances_overall, since 1 appearance can correspond
to 1 minute or 90 minutes. Minutes is a wider and more complete metric

goalkeepers = goalkeepers %>% select(-appearances_overall)
defenders = defenders %>% select(-appearances_overall)
midfielders = midfielders %>% select(-appearances_overall)
forwards = forwards %>% select(-appearances_overall)
```

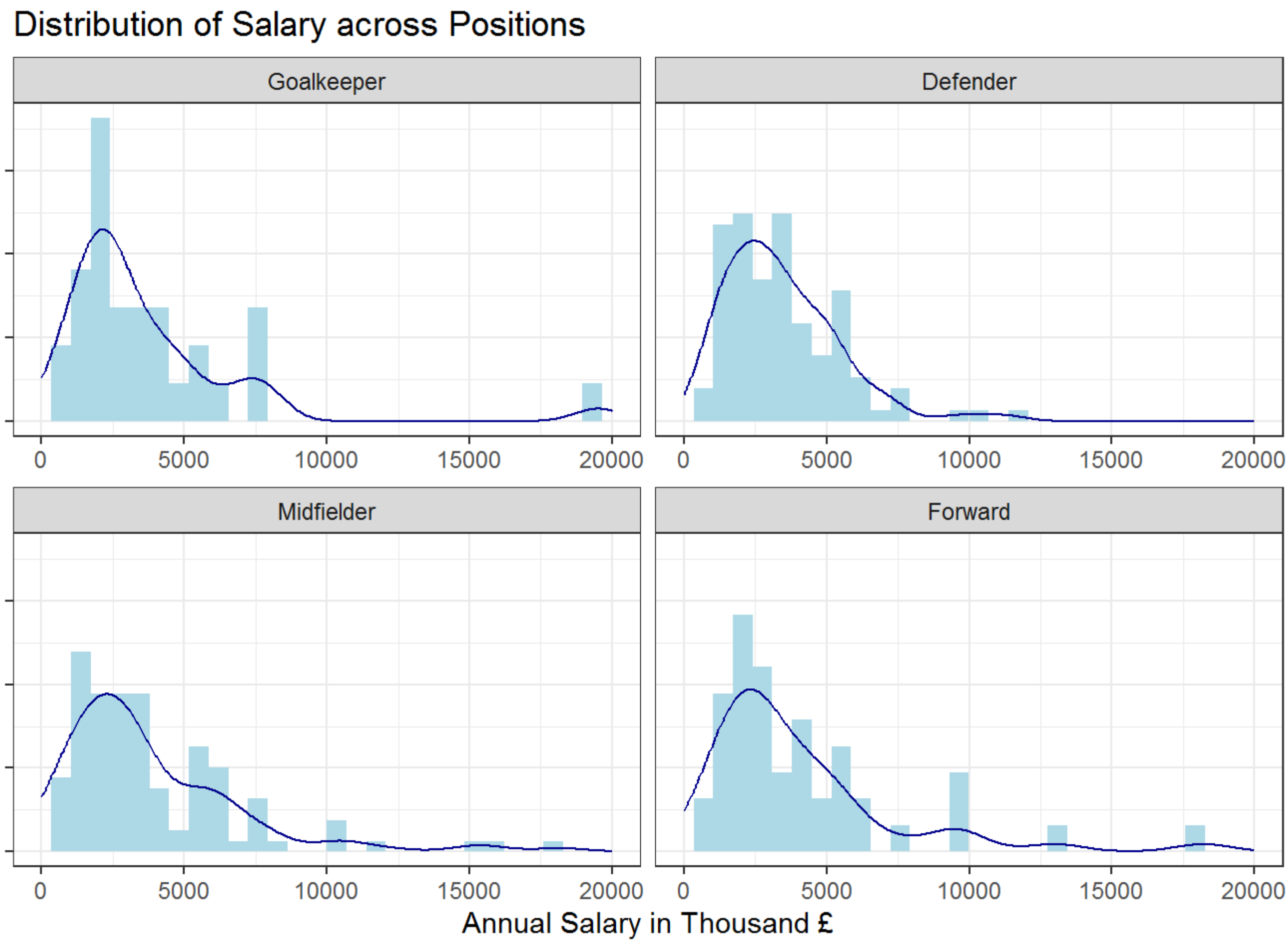
9. Initial Data Review

Histogram of Salaries across positions

```
options(scipen = 999)

## Annual Salary ##
ggplot(df, aes(x = annual_salary/1000, y = ..density..)) +
  geom_histogram(fill = "lightblue") + geom_line(stat = "density", color = "darkblue") + scal
e_y_continuous(labels = NULL) + theme(axis.ticks.y = element_blank()) + xlab("Annual Salary i
n Thousand £") + scale_x_continuous(limits = c(0, 20000)) + theme_bw() + ylab("") + facet_wra
p(~position, scales = "free_x") + ggtitle("Distribution of Salary across Positions")
```

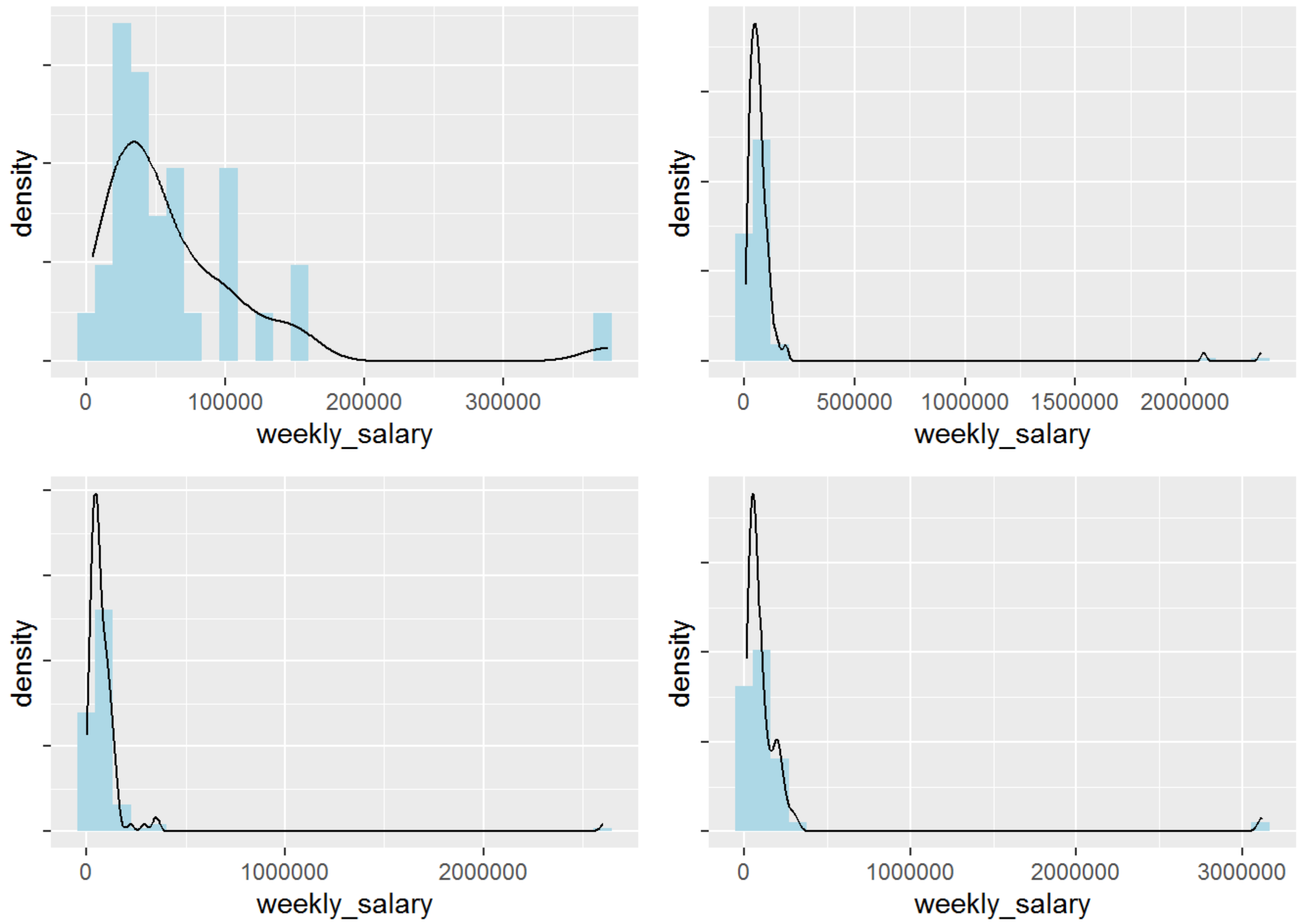
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 8 rows containing missing values (geom_bar).
```



```r
## Weekly Salary ##

g1 = ggplot(goalkeepers, aes(x = weekly_salary, y = ..density..)) +
  geom_histogram(fill = "lightblue") + geom_line(stat = "density") +
  scale_y_continuous(labels = NULL)
g2 = ggplot(defenders, aes(x = weekly_salary, y = ..density..)) +
  geom_histogram(fill = "lightblue") + geom_line(stat = "density") +
  scale_y_continuous(labels = NULL)
g3 = ggplot(midfielders, aes(x = weekly_salary, y = ..density..)) +
  geom_histogram(fill = "lightblue") + geom_line(stat = "density") +
  scale_y_continuous(labels = NULL)
g4 = ggplot(forwards, aes(x = weekly_salary, y = ..density..)) +
  geom_histogram(fill = "lightblue") + geom_line(stat = "density") +
  scale_y_continuous(labels = NULL)

library(gridExtra)
grid.arrange(g1, g2, g3, g4, nrow = 2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Correlations between performance metrics and Salary across positions

Subselect the most important performance metrics

In here, there is a component of subject knowledge, to identify which are the most important metrics in each position. I also just selected the 'pure' metrics and not the composite ones, for example, I chose to select goals_overall instead of goals_per_90 which is calculated from goals_overall, and so on.

```
## Goalkeepers ##
names(goalkeepers)
```

```
##  [1] "full_name"                 "age"
##  [3] "position"                  "Current Club"
##  [5] "minutes_played_overall"    "clean_sheets_overall"
##  [7] "conceded_overall"          "yellow_cards_overall"
##  [9] "red_cards_overall"         "conceded_per_90_overall"
## [11] "min_per_conceded_overall"  "min_per_match"
## [13] "min_per_card_overall"      "cards_per_90_overall"
## [15] "annual_salary"             "weekly_salary"
```

```
goalkeepers_perf = goalkeepers %>% select(age, minutes_played_overall, clean_sheets_overall,
conceded_overall, cards_per_90_overall,min_per_match, annual_salary)

## Defenders ##
names(defenders)
```

```
##  [1] "full_name"                      "age"
##  [3] "position"                       "Current Club"
##  [5] "minutes_played_overall"         "goals_overall"
##  [7] "assists_overall"                "clean_sheets_overall"
##  [9] "conceded_overall"               "yellow_cards_overall"
## [11] "red_cards_overall"              "goals_involved_per_90_overall"
## [13] "assists_per_90_overall"         "goals_per_90_overall"
## [15] "min_per_goal_overall"           "conceded_per_90_overall"
## [17] "min_per_conceded_overall"       "min_per_match"
## [19] "min_per_card_overall"           "min_per_assist_overall"
## [21] "cards_per_90_overall"           "annual_salary"
## [23] "weekly_salary"
```

```
defenders_perf = defenders %>% select(age, minutes_played_overall, goals_overall, assists_ove
rall, clean_sheets_overall, conceded_overall, cards_per_90_overall, goals_involved_per_90_ove
rall, min_per_match, annual_salary)

## Midfielders ##
names(midfielders)
```

```
##  [1] "full_name"                    "age"
##  [3] "position"                     "Current Club"
##  [5] "minutes_played_overall"       "goals_overall"
##  [7] "assists_overall"              "penalty_goals"
##  [9] "penalty_misses"               "clean_sheets_overall"
## [11] "conceded_overall"             "yellow_cards_overall"
## [13] "red_cards_overall"            "goals_involved_per_90_overall"
## [15] "assists_per_90_overall"       "goals_per_90_overall"
## [17] "min_per_goal_overall"         "conceded_per_90_overall"
## [19] "min_per_conceded_overall"     "min_per_match"
## [21] "min_per_card_overall"         "min_per_assist_overall"
## [23] "cards_per_90_overall"         "annual_salary"
## [25] "weekly_salary"
```

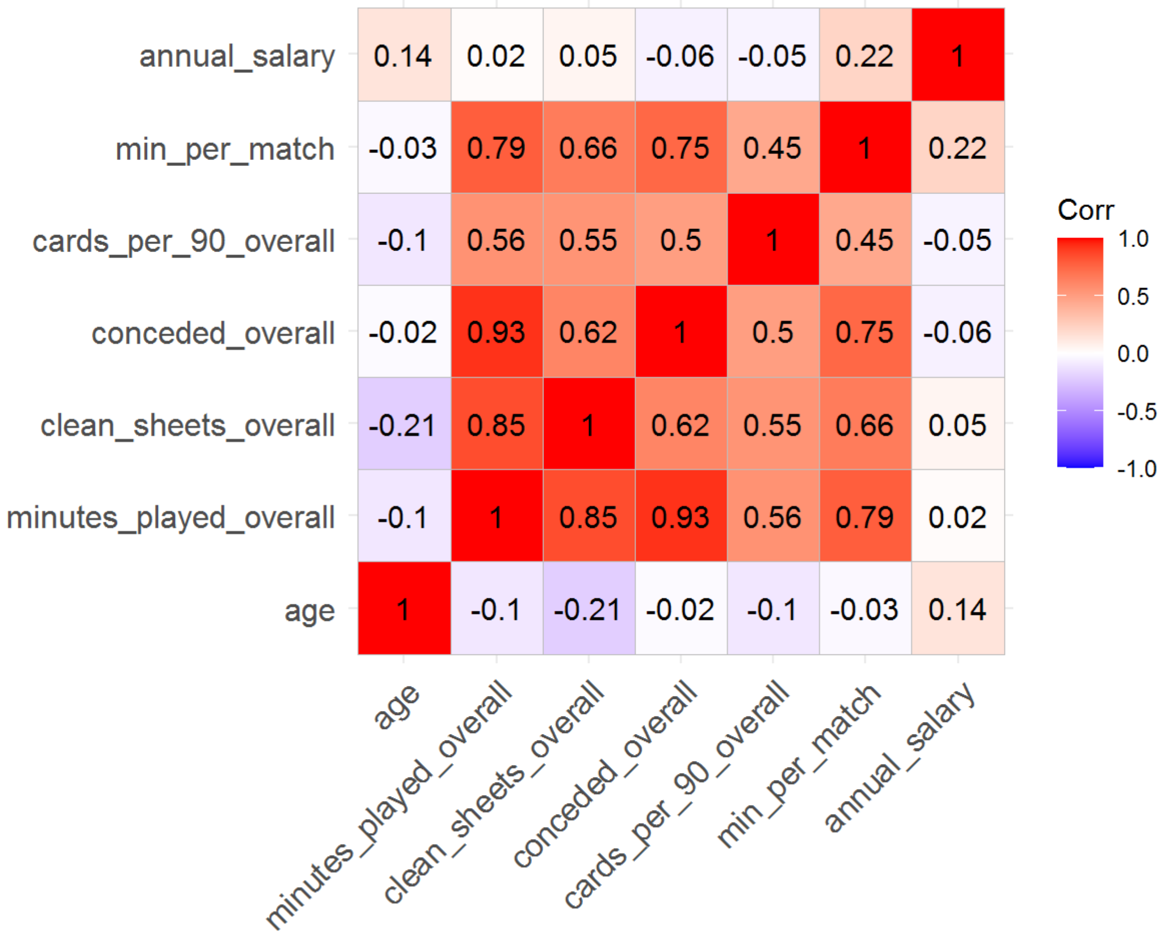
```
midfielders_perf = midfielders %>% select(age, minutes_played_overall, goals_overall, assists
_overall, penalty_goals, penalty_misses, cards_per_90_overall, goals_involved_per_90_overall,
min_per_match, annual_salary)

## Forwards ##
names(forwards)
```

```
##  [1] "full_name"                    "age"
##  [3] "position"                     "Current Club"
##  [5] "minutes_played_overall"       "goals_overall"
##  [7] "assists_overall"              "penalty_goals"
##  [9] "penalty_misses"               "yellow_cards_overall"
## [11] "red_cards_overall"            "goals_involved_per_90_overall"
## [13] "assists_per_90_overall"       "goals_per_90_overall"
## [15] "min_per_goal_overall"         "min_per_match"
## [17] "min_per_card_overall"         "min_per_assist_overall"
## [19] "cards_per_90_overall"         "annual_salary"
## [21] "weekly_salary"
```

```
forwards_perf = forwards %>% select(age, minutes_played_overall, goals_overall, assists_overa
ll, penalty_goals, penalty_misses, goals_involved_per_90_overall, min_per_match, cards_per_90
_overall, annual_salary)
```

Correlations btw Performance and Salary

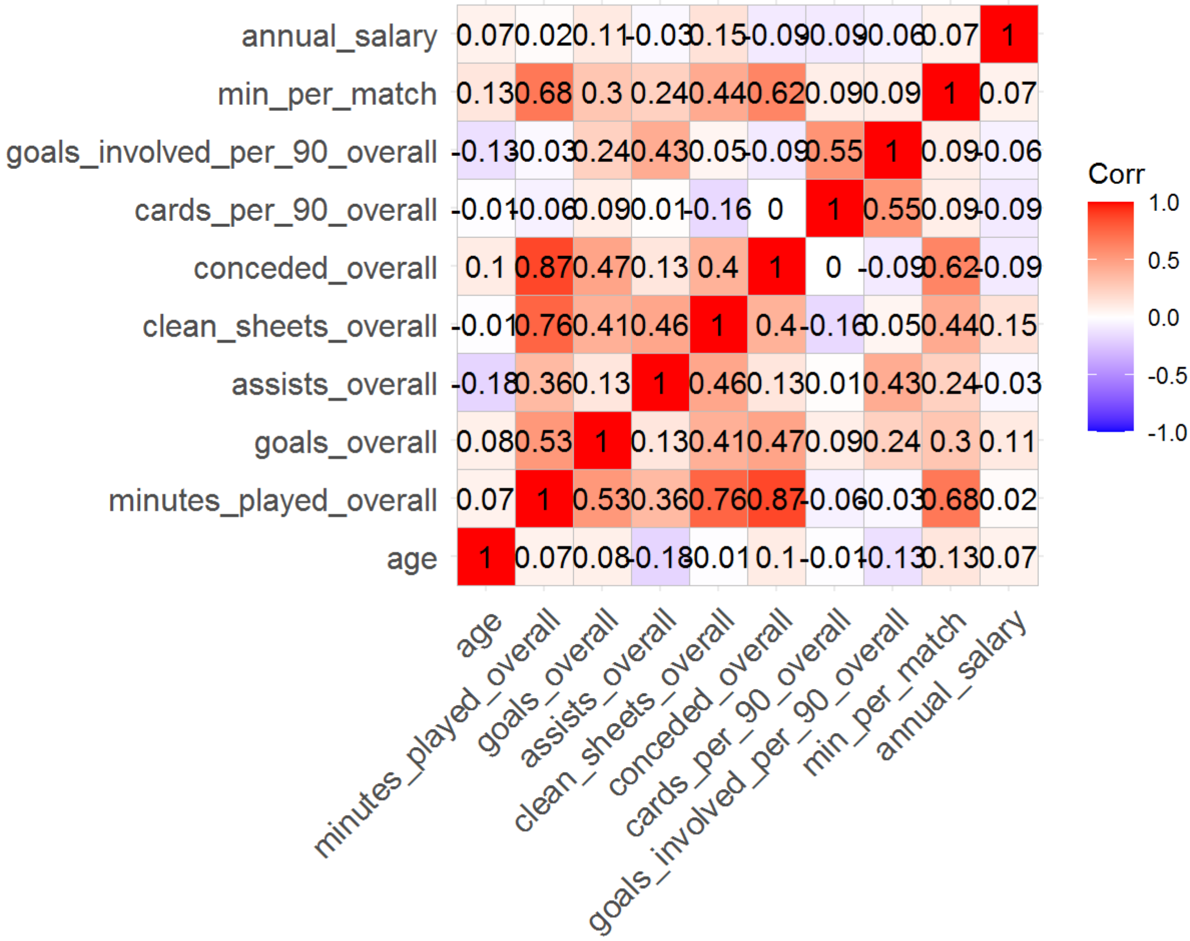
```
## Goalkeepers ##
Corr_goalkeepers = goalkeepers_perf %>% cor()

ggcorrplot(Corr_goalkeepers, lab = TRUE)
```




```
#Dataframe of correlations btw salary and performance metrics
data.frame(Corr_goalkeepers[,ncol(Corr_goalkeepers)])
```

```
##                         Corr_goalkeepers...ncol.Corr_goalkeepers..
## age                                                     0.14447521
## minutes_played_overall                                  0.01520224
## clean_sheets_overall                                    0.05171419
## conceded_overall                                       -0.05551119
## cards_per_90_overall                                   -0.04515734
## min_per_match                                           0.22185017
## annual_salary                                           1.00000000
```

```
## Defenders ##
Corr_defenders = defenders_perf %>% cor()

ggcorrplot(Corr_defenders, lab = TRUE)
```

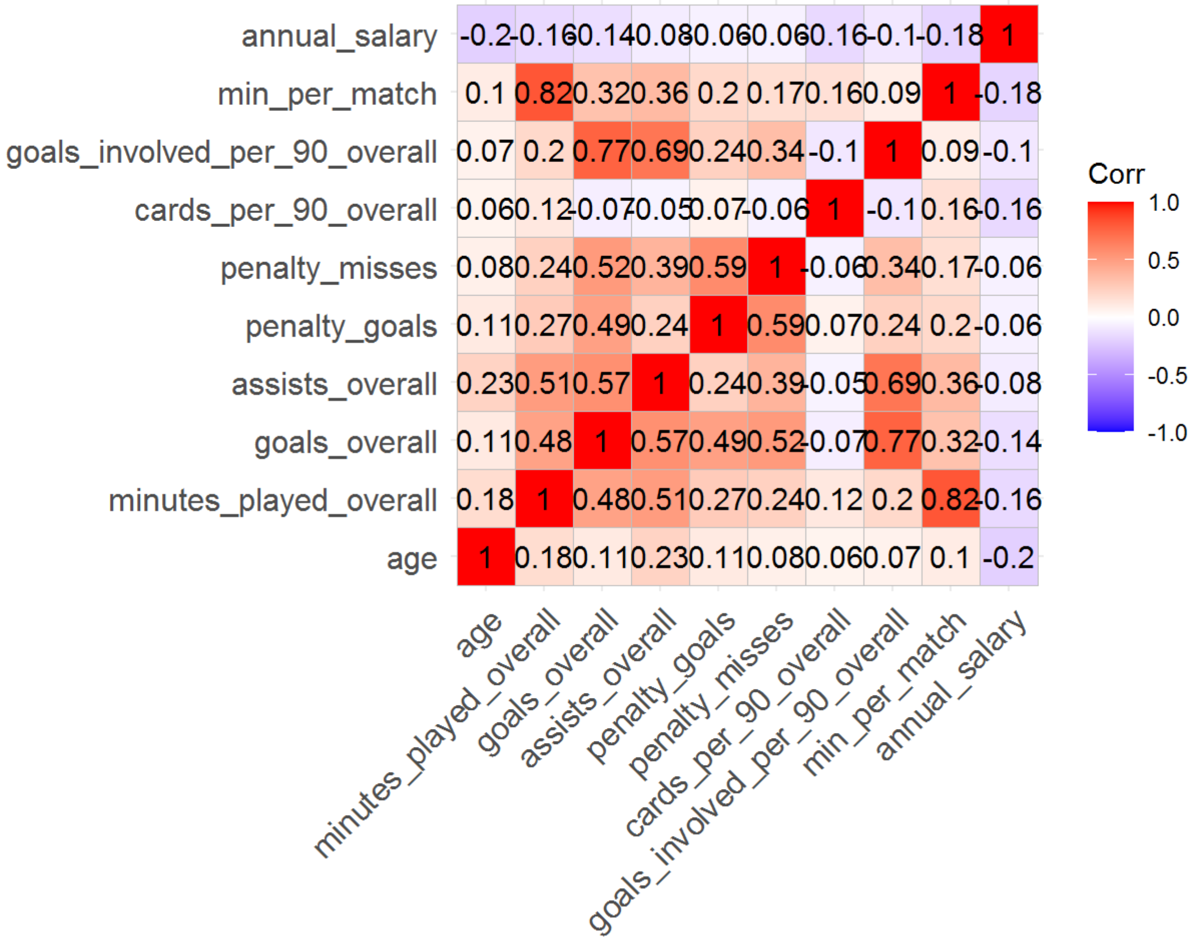



```
#Dataframe of correlations btw salary and performance metrics
data.frame(Corr_defenders[,ncol(Corr_defenders)])
```

```
##                                 Corr_defenders...ncol.Corr_defenders..
## age                                                         0.07089991
## minutes_played_overall                                      0.01712190
## goals_overall                                               0.10813921
## assists_overall                                            -0.03142507
## clean_sheets_overall                                        0.14940085
## conceded_overall                                           -0.08826101
## cards_per_90_overall                                       -0.08568118
## goals_involved_per_90_overall                              -0.05984899
## min_per_match                                               0.07033516
## annual_salary                                               1.00000000
```

```
## Relatively low correlations - nothing significant ##
```

```
## Midfielders ##
Corr_midfielders = midfielders_perf %>% cor()

ggcorrplot(Corr_midfielders, lab = TRUE)
```

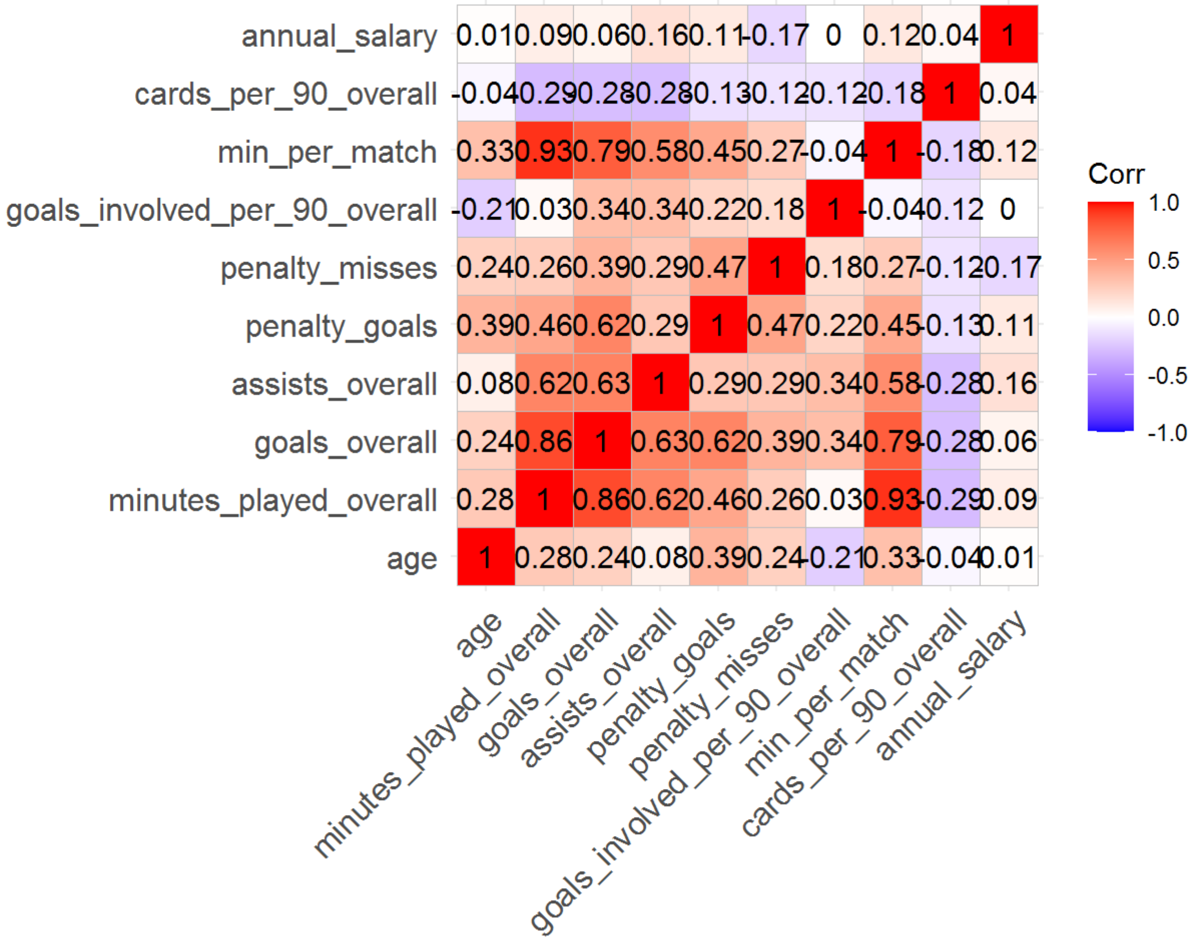



```
data.frame(Corr_midfielders[,ncol(Corr_midfielders)])
```

```
##                               Corr_midfielders...ncol.Corr_midfielders..
## age                                                          -0.20071235
## minutes_played_overall                                       -0.15833773
## goals_overall                                                -0.13790066
## assists_overall                                              -0.07961335
## penalty_goals                                                -0.05858686
## penalty_misses                                               -0.05902781
## cards_per_90_overall                                         -0.15902242
## goals_involved_per_90_overall                                -0.10380705
## min_per_match                                                -0.18373243
## annual_salary                                                 1.00000000
```

```
## No significantly high correlations, in fact there are many illogical negative relationship
s like minutes_played with salary, appearances and goals and assists with Salary
```

```
## Forwards ##
Corr_forwards = forwards_perf %>% cor()

ggcorrplot(Corr_forwards, lab = TRUE)
```



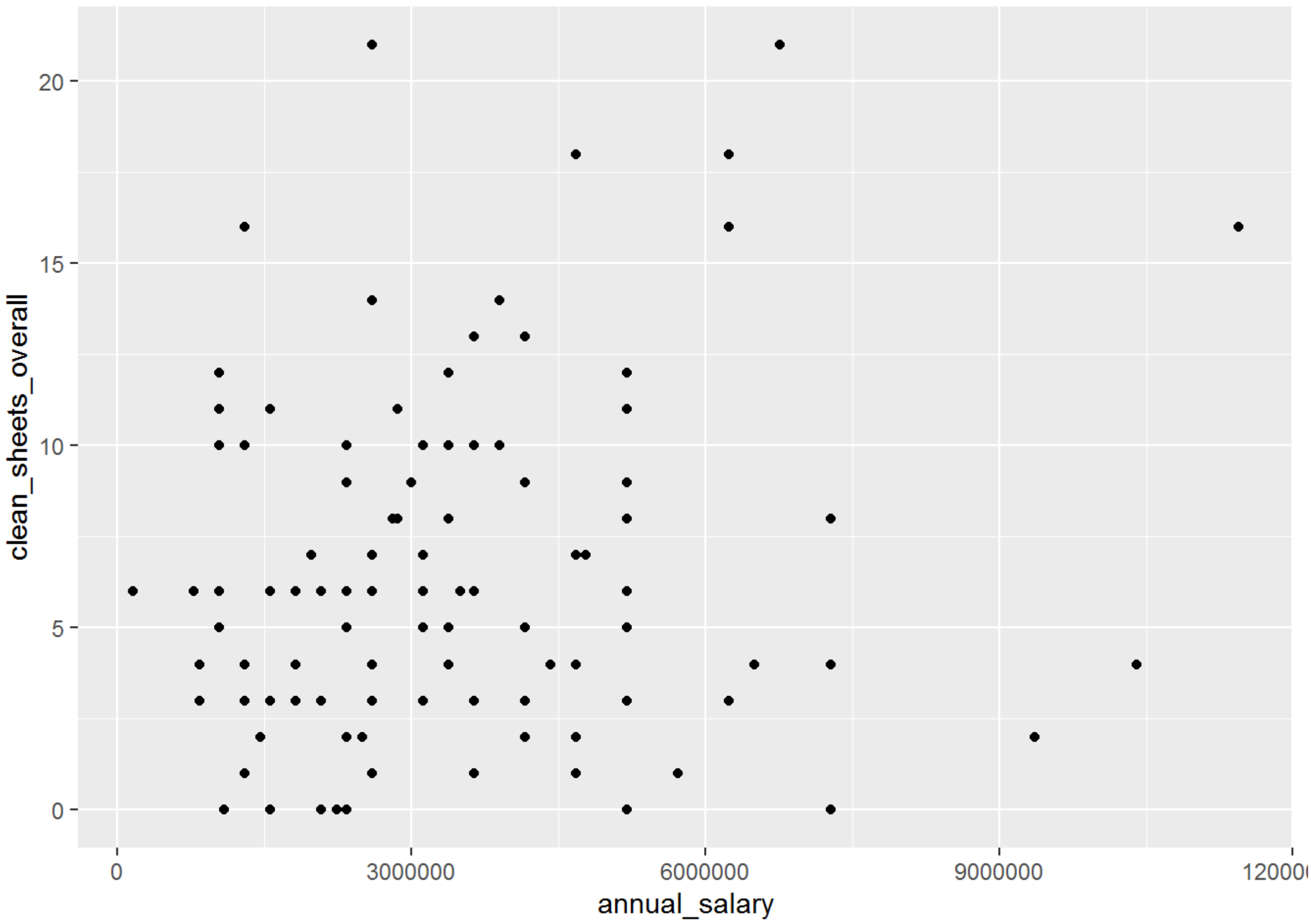
```
data.frame(Corr_forwards[,ncol(Corr_forwards)])
```

```
##                                 Corr_forwards...ncol.Corr_forwards..
## age                                                     0.0051215341
## minutes_played_overall                                  0.0873108230
## goals_overall                                           0.0587713228
## assists_overall                                         0.1574862711
## penalty_goals                                           0.1086632501
## penalty_misses                                         -0.1662336648
## goals_involved_per_90_overall                           0.0003531679
## min_per_match                                           0.1150383357
## cards_per_90_overall                                    0.0383737275
## annual_salary                                           1.0000000000
```

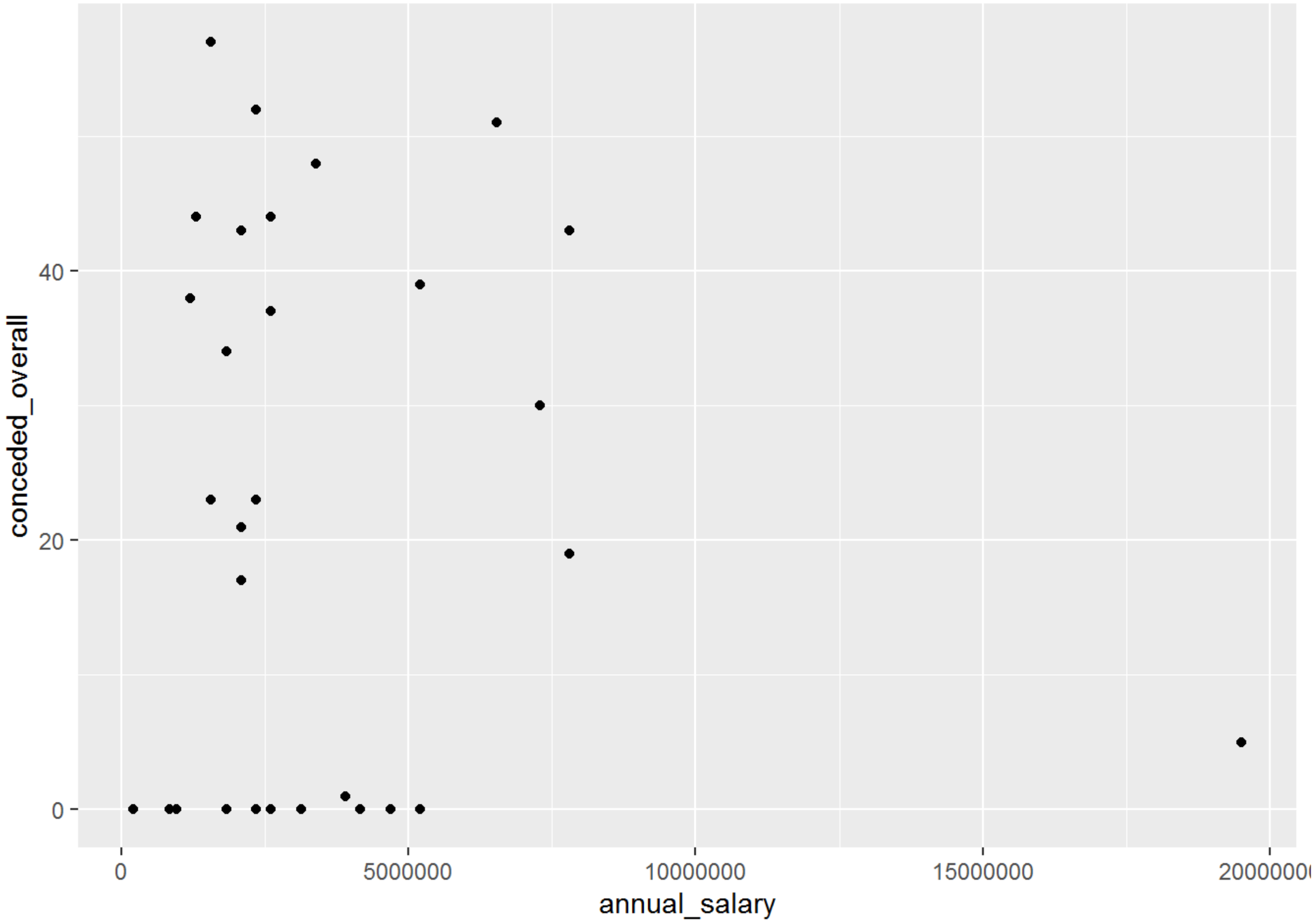
```
## More reasonable and logical correlations. Positive correlation with appearances, goals, as
sists, minutes played, etc.
```

Performance Metrics against Salary PLOTS

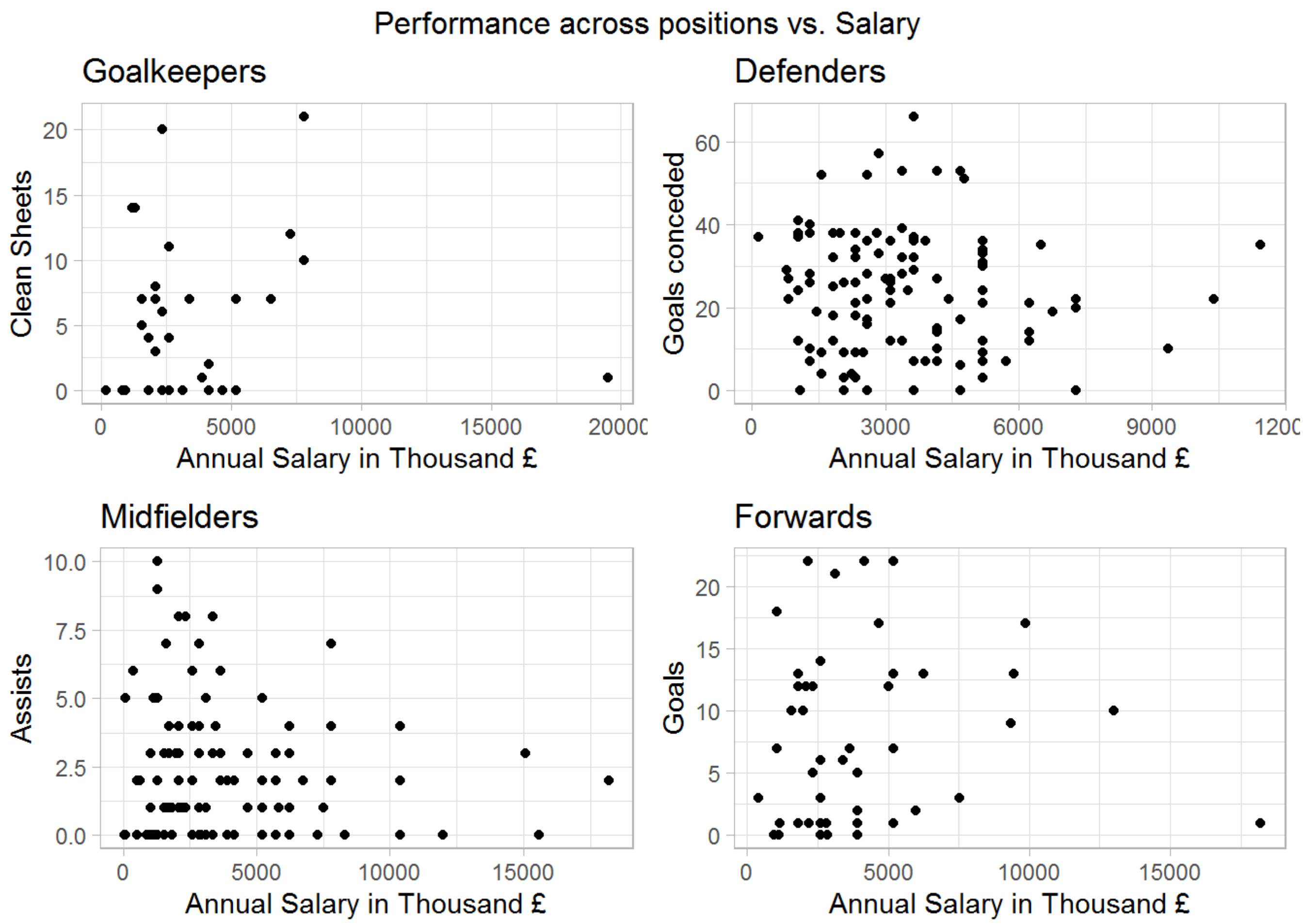
```
s1 = ggplot(forwards) + geom_point(aes(x = annual_salary/1000, y = goals_overall)) + xlab("An
nual Salary in Thousand £") + ylab("Goals") + ggtitle("Forwards") + theme_light()
# Some high scoring players that have low salary and some low scoring players that have high
 salaries

s2 = ggplot(midfielders) + geom_point(aes(x = annual_salary/1000, y = assists_overall)) + xla
b("Annual Salary in Thousand £") + ylab("Assists") + ggtitle("Midfielders") + theme_light()
# same pattern

ggplot(defenders) + geom_point(aes(x = annual_salary, y = clean_sheets_overall))
```




```
# somehow different pattern, shifted to the right, which means the higher the clean sheets, t
hen higher paid
s3 = ggplot(defenders) + geom_point(aes(x = annual_salary/1000, y = conceded_overall)) + xlab
("Annual Salary in Thousand £") + ylab("Goals conceded") + ggtitle("Defenders") + theme_light
()

s4 = ggplot(goalkeepers) + geom_point(aes(x = annual_salary/1000, y = clean_sheets_overall))
 + xlab("Annual Salary in Thousand £") + ylab("Clean Sheets") + ggtitle("Goalkeepers") + them
e_light()

ggplot(goalkeepers) + geom_point(aes(x = annual_salary, y = conceded_overall))
```




```
grid.arrange(s4,s3,s2,s1, nrow = 2, top = "Performance across positions vs. Salary")
```



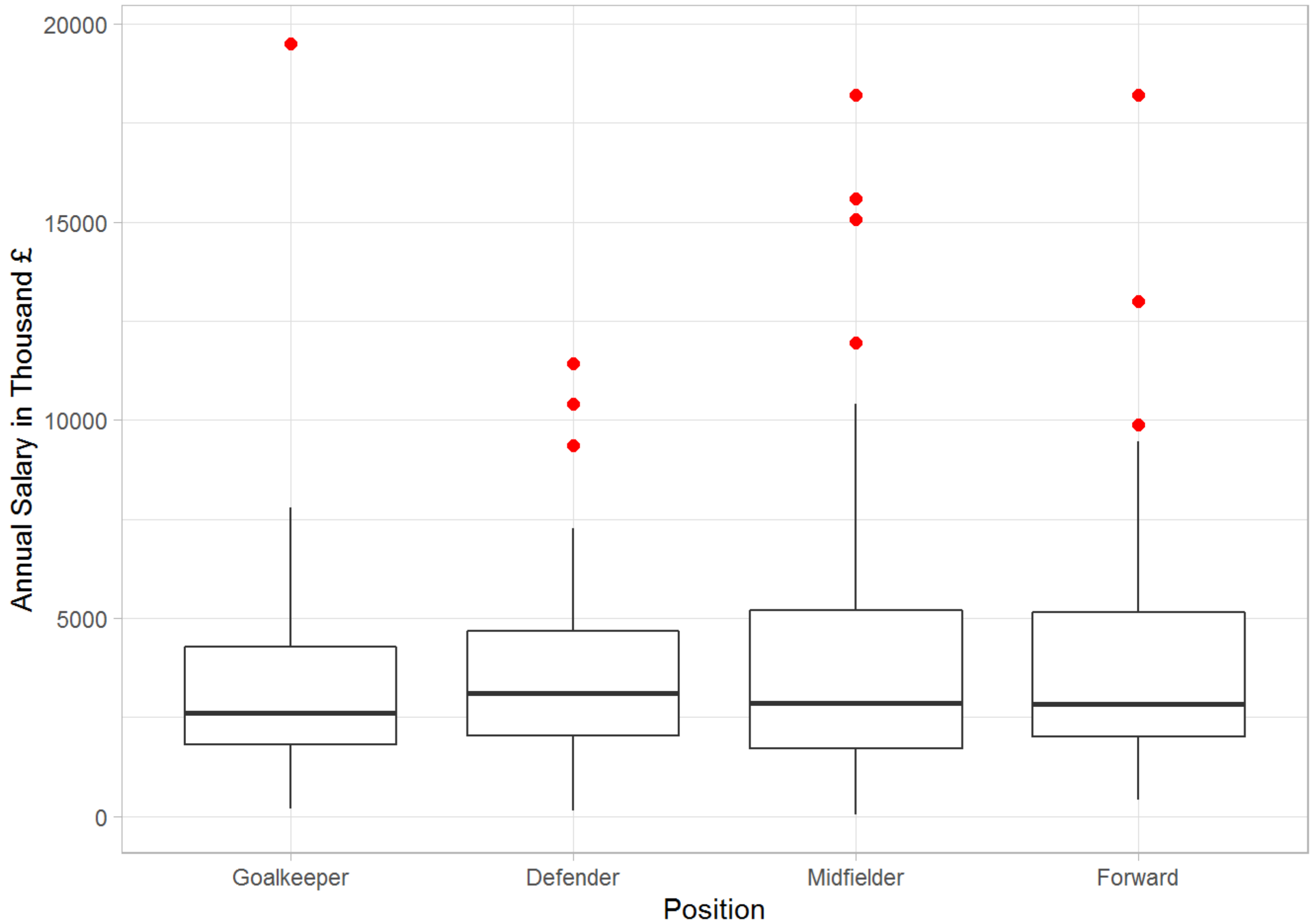
```
## Which position is the best paid ##
df %>% group_by(position) %>% summarise(avg_salary = mean(annual_salary))
```

```
## # A tibble: 4 x 2
##   position   avg_salary
##   <ord>           <dbl>
## 1 Goalkeeper   3691656.
## 2 Defender     3412348.
## 3 Midfielder   3798122.
## 4 Forward      4045115.
```

Analysis of variance: salary accross Positions

First we will look into the distribution of the 4 positions

```
library(ggplot2)

ggplot(df) + geom_boxplot(aes(x = position, y = annual_salary/1000), outlier.size=2,outlier.c
olour="red") + xlab("Position") + ylab("Annual Salary in Thousand £") + theme_light()
```




```
#They kind of have the same distribution, but different ranges, some wider
```

```
anova = aov(annual_salary ~ position, data = df)
summary(anova)
```

```
##               Df           Sum Sq       Mean Sq F value Pr(>F)
## position      3   15689287840951 5229762613650   0.625  0.599
## Residuals   300 2509664389921258 8365547966404
```

```
TukeyHSD(anova)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = annual_salary ~ position, data = df)
##
## $position
##                            diff        lwr     upr     p adj
## Defender-Goalkeeper   -279307.9 -1778596.4 1219981 0.9631979
## Midfielder-Goalkeeper  106465.5 -1386976.8 1599908 0.9977793
## Forward-Goalkeeper     353458.4 -1366615.4 2073532 0.9515188
## Midfielder-Defender    385773.4  -608480.9 1380028 0.7481140
## Forward-Defender       632766.3  -677510.7 1943043 0.5970342
## Forward-Midfielder     246992.9 -1056590.4 1550576 0.9613962
```

10. Linear Regression

Goalkeepers

```
## Multiple Linear Regression on Goalkeepers ##

m1_goalkeepers = lm(annual_salary ~ age +minutes_played_overall + clean_sheets_overall + conc
eded_overall + min_per_match + cards_per_90_overall,goalkeepers_perf)

summary(m1_goalkeepers)
```

```
##
## Call:
## lm(formula = annual_salary ~ age + minutes_played_overall + clean_sheets_overall +
##     conceded_overall + min_per_match + cards_per_90_overall,
##     data = goalkeepers_perf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3801439 -1906755  -142082  1242328 12599784
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)               -60153    4695228  -0.013   0.9899
## age                        85200     145305   0.586   0.5629
## minutes_played_overall      6813       4192   1.625   0.1167
## clean_sheets_overall     -805915     524318  -1.537   0.1368
## conceded_overall         -417500     210646  -1.982   0.0586 .
## min_per_match              51941      22936   2.265   0.0325 *
## cards_per_90_overall     -708193   27843072  -0.025   0.9799
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3358000 on 25 degrees of freedom
## Multiple R-squared:  0.2663, Adjusted R-squared:  0.09027
## F-statistic: 1.513 on 6 and 25 DF,  p-value: 0.2148
```

```
#R2 of 0.30 which is pretty low

# If we drop all not significant predictors #

m2_goalkeepers = lm(annual_salary ~ conceded_overall + min_per_match, goalkeepers_perf)

summary(m2_goalkeepers)
```

```
##
## Call:
## lm(formula = annual_salary ~ conceded_overall + min_per_match,
##     data = goalkeepers_perf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3485481 -2296431  -544213  1553558 12880930
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2627034    1004071   2.616   0.0140 *
## conceded_overall    -87799      44272  -1.983   0.0569 .
## min_per_match        49234      20934   2.352   0.0257 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3330000 on 29 degrees of freedom
## Multiple R-squared:  0.1628, Adjusted R-squared:  0.105
## F-statistic: 2.819 on 2 and 29 DF,  p-value: 0.07608
```

```
# By dropping attributes R2 decreases to 0.163 and just min_per_match remains significant
```

Defenders

```
## Multiple Linear Regression on Defenders ##

m1_defenders = lm(annual_salary ~ age + minutes_played_overall + clean_sheets_overall + conce
ded_overall + assists_overall + goals_overall + goals_involved_per_90_overall + min_per_match
+ cards_per_90_overall, defenders_perf)

summary(m1_defenders)
```

```
##
## Call:
## lm(formula = annual_salary ~ age + minutes_played_overall + clean_sheets_overall +
##     conceded_overall + assists_overall + goals_overall + goals_involved_per_90_overall +
##     min_per_match + cards_per_90_overall, data = defenders_perf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3011227 -1216848  -290247   980006  6930345
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1917235.6  1740685.4   1.101    0.273
## age                             24340.8    57081.9   0.426    0.671
## minutes_played_overall           -331.2      997.1  -0.332    0.740
## clean_sheets_overall           104044.0   108760.2   0.957    0.341
## conceded_overall               -36179.4    45368.8  -0.797    0.427
## assists_overall                -65146.2   130925.2  -0.498    0.620
## goals_overall                  306973.6   194563.9   1.578    0.118
## goals_involved_per_90_overall -1331459.0  1713741.0  -0.777    0.439
## min_per_match                   20932.7    13898.0   1.506    0.135
## cards_per_90_overall          -310017.6  1331622.2  -0.233    0.816
##
## Residual standard error: 1989000 on 101 degrees of freedom
## Multiple R-squared:  0.1066, Adjusted R-squared:  0.02701
## F-statistic: 1.339 on 9 and 101 DF,  p-value: 0.2263
```

Midfielders

```
## Multiple Linear Regression on Midfielders ##

m1_midfielders = lm(annual_salary ~ age + minutes_played_overall + assists_overall + goals_ov
erall + penalty_goals + penalty_misses + goals_involved_per_90_overall + min_per_match + card
s_per_90_overall, midfielders_perf)

summary(m1_midfielders)
```

```
##
## Call:
## lm(formula = annual_salary ~ age + minutes_played_overall + assists_overall +
##     goals_overall + penalty_goals + penalty_misses + goals_involved_per_90_overall +
##     min_per_match + cards_per_90_overall, data = midfielders_perf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -5039379 -1628797  -537694   977564 13362440
##
## Coefficients:
##                                 Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)                  11186576.67  2634830.19   4.246 0.0000472 ***
## age                           -170826.01    83348.44  -2.050    0.0429 *
## minutes_played_overall            -16.18      660.41  -0.025    0.9805
## assists_overall                223642.78   229851.79   0.973    0.3328
## goals_overall                  -33134.82   221982.05  -0.149    0.8816
## penalty_goals                  128018.94   314140.93   0.408    0.6845
## penalty_misses                -286290.36   921744.66  -0.311    0.7567
## goals_involved_per_90_overall -2849642.96  3515873.31  -0.811    0.4195
## min_per_match                  -28411.65    26731.59  -1.063    0.2903
## cards_per_90_overall          -2821203.39  2009980.74  -1.404    0.1634
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3146000 on 105 degrees of freedom
## Multiple R-squared:  0.1024, Adjusted R-squared:  0.02552
## F-statistic: 1.332 on 9 and 105 DF,  p-value: 0.2295
```

```
# R2 of 0.144

# Dropping all non significant predictors

m2_midfielders = lm(annual_salary ~ age, midfielders_perf)

summary(m2_midfielders)
```

```
##
## Call:
## lm(formula = annual_salary ~ age, data = midfielders_perf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4418369 -1918661  -640878  1179829 13919122
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8618142    2232362   3.861 0.000189 ***
## age          -173491      79659  -2.178 0.031490 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3136000 on 113 degrees of freedom
## Multiple R-squared:  0.04029,    Adjusted R-squared:  0.03179
## F-statistic: 4.743 on 1 and 113 DF,  p-value: 0.03149
```

```
# Just age remains significant and R2 is 0.04, really low
```

Forwards

```
## Multiple Linear Regression on Forwards ##

m1_forwards = lm(annual_salary ~ age + minutes_played_overall + assists_overall + goals_overa
ll + penalty_goals + penalty_misses + goals_involved_per_90_overall + min_per_match + cards_p
er_90_overall, forwards_perf)

summary(m1_forwards)
```

```
##
## Call:
## lm(formula = annual_salary ~ age + minutes_played_overall + assists_overall +
##     goals_overall + penalty_goals + penalty_misses + goals_involved_per_90_overall +
##     min_per_match + cards_per_90_overall, data = forwards_perf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3194771 -1959852  -782341   467703 14003428
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     3610147    5314958   0.679    0.501
## age                              -41093     172155  -0.239    0.813
## minutes_played_overall             -580       1907  -0.304    0.763
## assists_overall                  334307     277369   1.205    0.236
## goals_overall                    -83333     215029  -0.388    0.701
## penalty_goals                    745264     590665   1.262    0.215
## penalty_misses                 -4181208    2489993  -1.679    0.102
## goals_involved_per_90_overall  -287687    2159804  -0.133    0.895
## min_per_match                     33634      68242   0.493    0.625
## cards_per_90_overall            1366950    4474726   0.305    0.762
##
## Residual standard error: 3552000 on 36 degrees of freedom
## Multiple R-squared:  0.1275, Adjusted R-squared:  -0.09062
## F-statistic: 0.5845 on 9 and 36 DF,  p-value: 0.8008
```

```
# R2 of 0.15 and all predictors are not significant
```

Most Appropiate Variables

I think that these are the most important performance metrics for each position

```
## Goalkeepers ##
msimple_goalkeepers = lm(annual_salary ~ minutes_played_overall + clean_sheets_overall + conc
eded_overall, goalkeepers_perf)

summary(msimple_goalkeepers)
```

```
##
## Call:
## lm(formula = annual_salary ~ minutes_played_overall + clean_sheets_overall +
##     conceded_overall, data = goalkeepers_perf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3594693 -1371274  -674404   842701 15727395
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)             3794693     921982   4.116 0.000308 ***
## minutes_played_overall     6813       4266   1.597 0.121489
## clean_sheets_overall    -706426     511520  -1.381 0.178193
## conceded_overall        -353671     210004  -1.684 0.103277
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3519000 on 28 degrees of freedom
## Multiple R-squared:  0.0972, Adjusted R-squared:  0.0004668
## F-statistic: 1.005 on 3 and 28 DF,  p-value: 0.4052
```

```
#Worst than before

## Defenders ##
msimple_defenders = lm(annual_salary ~ minutes_played_overall +  conceded_overall + goals_inv
olved_per_90_overall, defenders_perf)

summary(msimple_defenders)
```

```
##
## Call:
## lm(formula = annual_salary ~ minutes_played_overall + conceded_overall +
##     goals_involved_per_90_overall, data = defenders_perf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3044895 -1388530  -205963  1156949  7603971
##
## Coefficients:
##                                Estimate Std. Error t value           Pr(>|t|)
## (Intercept)                   3547413.5   409136.4   8.670 0.0000000000000511 ***
## minutes_played_overall            779.9      376.3   2.073             0.0406 *
## conceded_overall               -59852.7    26087.9  -2.294             0.0237 *
## goals_involved_per_90_overall -1018295.3  1109215.3  -0.918             0.3607
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1992000 on 107 degrees of freedom
## Multiple R-squared:  0.05053,    Adjusted R-squared:  0.02391
## F-statistic: 1.898 on 3 and 107 DF,  p-value: 0.1343
```

```
#Worst than before

## Midfielders ##
msimple_midfielders = lm(annual_salary ~ minutes_played_overall + goals_overall + assists_ove
rall + goals_involved_per_90_overall, midfielders_perf)

summary(msimple_midfielders)
```

```
##
## Call:
## lm(formula = annual_salary ~ minutes_played_overall + goals_overall +
##     assists_overall + goals_involved_per_90_overall, data = midfielders_perf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4528261 -1965182  -790842  1303150 14136669
##
## Coefficients:
##                                Estimate Std. Error t value      Pr(>|t|)
## (Intercept)                   5085502.0   797903.2   6.374 0.00000000444 ***
## minutes_played_overall           -628.1      449.6  -1.397         0.165
## goals_overall                     626.3   187892.9   0.003         0.997
## assists_overall                166934.2   215902.1   0.773         0.441
## goals_involved_per_90_overall -2465952.8  3330635.7  -0.740         0.461
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3185000 on 110 degrees of freedom
## Multiple R-squared:  0.03615,    Adjusted R-squared:  0.001106
## F-statistic: 1.032 on 4 and 110 DF,  p-value: 0.3943
```

```
#Worst than before

## Forwards ##
msimple_forwards = lm(annual_salary ~ minutes_played_overall + goals_overall + assists_overal
l + goals_involved_per_90_overall, forwards_perf)

summary(msimple_forwards)
```

```
##
## Call:
## lm(formula = annual_salary ~ minutes_played_overall + goals_overall +
##     assists_overall + goals_involved_per_90_overall, data = forwards_perf)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2986963 -2192260  -944680  1001801 13891795
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   3601432.6  1719197.3   2.095   0.0424 *
## minutes_played_overall            155.6     1288.3   0.121   0.9045
## goals_overall                  -47318.1   186378.3  -0.254   0.8009
## assists_overall                251798.1   266121.5   0.946   0.3496
## goals_involved_per_90_overall -372703.4  2014111.1  -0.185   0.8541
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3509000 on 41 degrees of freedom
## Multiple R-squared:  0.03024,    Adjusted R-squared:  -0.06437
## F-statistic: 0.3196 on 4 and 41 DF,  p-value: 0.8632
```

```
#Worst than before
```