

# Who is really the most effective goalscorer

2022-11-14

This notebook is for analyzing goal scoring abilities of the best goalscorer in the last 8 year in the top 5 leagues in the world of football (season 2014/2015 to 2021/2022). We rank the players by the total amount of goals succeeding their total amount of expected Goals (xG), which means we take the difference between “total goals scored” minus “total goals expected”. Therefore this notebook is divided into 3 parts.

In the first part all necessary data will be imported via the “understatr” library.

In the second part data will be analyzed and prepared. Our model focuses on goals and expected Goals (xG). In order to get the meaningful xG ranges, in which we will compare different players, we have to modify the data.

In the third part the results will be outputted and explained.

Thanks go out to BiscuitChaserFC for inspiring this idea of our model and ewen\_ for the package for understatR! Link to the BiscuitChaserFC article: <https://biscuitchaserfc.blogspot.com/2020/09/shot-data-for-top-5-european-leagues.html>

Import libraries

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   0.3.5
## v tibble  3.1.8     v dplyr    1.0.10
## v tidyr   1.2.1     v stringr  1.4.1
## v readr   2.1.3     v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()   masks stats::lag()

library(understatr)
library("dplyr")
library(devtools)

## Loading required package: usethis
library(ggplot2)
library(ggsoccer)
library('bigasserrtr')
```

## Start of function declaration

These 2 functions are for the various fields- and shotmap plots used in this notebook

```
plot_areas_xG <- function(df, var, pltTitle) {

  ggplot() +
  annotate_pitch(colour = "black",
                 fill   = "white",
                 limits = FALSE) +
```

```

geom_rect(data = df, inherit.aes=FALSE, aes(xmin = xmin, xmax = xmax, ymin = ymin, ymax = ymax,
    fill = var), colour=NA, alpha=.45) +
    scale_fill_manual('xG Areas',
                      values=c("cyan", "darkred", "deeppink", "darkseagreen", "cornflowerblue", "yellow"),
                      guide = guide_legend(override.aes = list(alpha = 0.45))) +
    theme_pitch() +
    theme(panel.background = element_rect(fill = "white")) +
    coord_flip(xlim = c(50, 101),
               ylim = c(-12, 112)) +
    ggtitle(pltTitle)

}

plot_shot_player <- function(boolean, df, strName, lowTshld, upTshld) {

  df <- df[(df$player==strName & df$xG >= lowTshld & df$xG <= upTshld & df$situation != "Penalty"),]
  df <- df[,c("X","Y","xG","result", "situation", "xG_Ranges")]
  df$X <- df$X * 100
  df$Y <- df$Y * 100
  lowTshld <- lowTshld * 100
  upTshld <- upTshld * 100

  if (boolean ==TRUE) {
    a = geom_point(aes(x = X, y = 100 - Y, fill = result),
                   shape = 21,
                   size = 2.8)

    c = ggtitle(paste0(strName, ' shotmap where Expected Goals range between ',lowTshld, '% - ', upTshld))
  } else {
    df <- df[(df$result == "Goal"),]
    a = geom_point(aes(x = X, y = 100 - Y, fill= xG_Ranges ), #colour=xG_Ranges
                   shape = 21,
                   size = df$xG *4.5)
    c = ggtitle(paste0(strName, ' goalmap where Expected Goals range between ',lowTshld, '% - ', upTshld))
  }

  ggplot(df) +
  annotate_pitch(colour = "white",
                 fill    = "springgreen4",
                 limits = FALSE) +
  a +
  theme_pitch() +
  theme(panel.background = element_rect(fill = "springgreen4")) +
  coord_flip(xlim = c(50, 102),
             ylim = c(-12, 112)) +
  c
}

```

```

plot_areas <- function(df, strName, insideAreas, area) {

  df <- df[(df$player==strName & df$xGAreas == area & df$situation != "Penalty"),]
  df$X <- df$X * 100
  df$Y <- df$Y * 100
  ggplot(df) +
    annotate_pitch(colour = "white",
                  fill = "springgreen4",
                  limits = FALSE) +
    geom_point(aes(x = X, y = 100 - Y, fill = xG_Ranges),
               shape = 21,
               size = df$xG * 5) +
    geom_rect(data = insideAreas, inherit.aes=FALSE, aes(xmin = xmin, xmax = xmax, ymin = ymin, ymax = ymax,
                                                       fill = area), colour=NA, alpha=.045) +
    scale_fill_manual('xG Areas',
                      values=c("cyan", "darkred", "deeppink", "darkseagreen", "cornflowerblue", "yellow"),
                      guide = guide_legend(override.aes = list(alpha = 0.45))) +
    theme_pitch() +
    theme(panel.background = element_rect(fill = "springgreen4")) +
    coord_flip(xlim = c(50, 101),
               ylim = c(-12, 112)) +
    ggtitle(paste0(strName, ' goalmap ', area))

}

```

This function formats numbers in a from us desired format

```

so_formatter <- function(x) {
  dplyr::case_when(
    x < 1e3 ~ as.character(x),
    x < 1e6 ~ paste0(as.character(x/1e3), "K"),
    x < 1e9 ~ paste0(as.character(x/1e6), "M"),
    TRUE ~ "To be implemented..."
  )
}

```

This is a function to get the player with the most goals. We are also free to set a minimum threshold of goals must be scored in order to be considered. The boolean variable let's us decide whether we want to include penalties or not.

```

get_goals <- function(boolean, df, t) {
  if (boolean == TRUE) {
    df <- df[(df$situation != "Penalty"),]
  }
  df <- df %>% group_by(player_id, player, result) %>% summarize(count=n())
  df <- df[(df$result == "Goal" & df$count>=t),]
  df2<-df[order(-df$count),]
  return (df2)
}

```

This is just a helper function to create a dataframe dynamically

```

dynamically_df <- function(df, df2) {

  intAllShots = nrow(df)
  intGoals = sum(df$result == "Goal")
  dblConvRate= round(intGoals/intAllShots, digits =3);
  dblAvgXG = round(mean(df$xG, na.rm = TRUE), digits = 3)
  dblPerXG = round((dblConvRate - dblAvgXG), digits = 3)
  intGoalsxG = round((dblAvgXG * intAllShots), 2)
  dblPerXGTotal = intGoals - intGoalsxG

  df2[nrow(df2) + 1,] <- list(strName, dblConvRate, intGoalsxG, intGoals, intAllShots, dblAvgXG, dblPerXG)
  df2 <-df2[order(-df2$Performance_Goals),]

  return (df2)
}

```

This function calculates all interesting key figures for penalties or free kicks (definded by the boolean)

```

results_standards <- function(boolean, df, df2, strName) {

  if (boolean == TRUE) {
    df <- df[(df$player==strName & df$situation == "Penalty" ),]
  } else {
    df <- df[(df$player==strName & df$situation == "DirectFreekick" ),]
  }
  df2 <- dynamically_df(df, df2)
  return (df2)

}

```

This function calculates all interesting key figures important for a goalscorer and return the result ordered in a dataframe. The user has the freedom to choose which xG ranges he wants to consider, which are defined by “lowTshld” and “upTshld”.

```

results_ranges <- function(boolean, df, df2, strName, lowTshld, upTshld) {

  if (boolean == TRUE) {
    df <- df[(df$player==strName & df$xG >= lowTshld & df$xG <= upTshld & df$situation != "Penalty" ),]
  } else {
    df <- df[(df$player==strName & df$xG > lowTshld & df$xG <= upTshld ),]
  }
  df2 <- dynamically_df(df, df2)
  return (df2)

}

```

This function calculates all interesting key figures important for a goalscorer according to the areas we defined and return the result ordered in a dataframe

```

results_area <- function(boolean, df, df2, strName, strArea) {

  if (boolean == TRUE) {
    df <- df[(df$player==strName & df$xGAreas == strArea & df$situation != "Penalty" ),]
  } else {
    df <- df[(df$player==strName & df$xGAreas == strArea),]
  }
}

```

```

df2 <- dynamically_df(df, df2)
return (df2)

}

```

Variable Declaration

```
n = 3
```

## Part 1

All the code related to the process of pulling the “understatR” data is commented, because the performance of rendering the R markdown file to html was very slow as good as impossible. This pulling process can take up many hours. You can just comment out the code at this point if you want to test the code yourselves.

```
leagues<-get_leagues_meta() leagues<-leagues%>% filter(!league_name == "RFPL") #
```

```
These are the seasons we will take a closer look to # {r} unique(leagues$season) #
```

```
Pull team data with the purrr package to cycle through each unique league name and pull the team data for each match within the top 5 leagues # {r} team_data= data.frame() dfTemp = data.frame() #
```

```
Pull team data for the years desired with the purrr package to cycle through each unique league name and pull the team data for each match within the top 5 leagues # {r} for (year in c (2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021)) { dfTemp<-map_dfr(unique(leagues$league_name), get_league_teams_stats, year = year) team_data = rbind(team_data,dfTemp)} dfTemp= data.frame() #
```

```
Count the number of occurrences of each team by team_id and team_name # {r} print(team_data %>% group_by(team_id,team_name, league_name) %>% summarize(count=n())) #
```

```
Now we pull the player data for the desired years, here again we make use of the understatR library # {r}
player_data = data.frame() for (year in c (2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021)) { dfTemp<-map_dfr(unique(team_data$team_name), get_team_players_stats, year = year) player_data = rbind(player_data,dfTemp)} dfTemp= data.frame() #
```

```
First, create a vector of player_id (this is required in the UnderstatR get_player_shots() function # {r}
players<-c(player_data$player_id) #
```

```
Now we pull the event data and get all the shots in these time period # {r} shot_data <- players %>% map_dfr(.,possibly(get_player_shots,otherwise=NULL)) #
```

```
Let's remove the data we don't need anymore # {r} rm(leagues, player_data, team_data, players, year) #
```

```
Let's export the shot_data as the process of pulling it from the internet was taking many hours # {r}
write.csv(shot_data, "shot_data.csv") #
```

```
shot_data <- read.csv(file = 'shot_data.csv')
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec, :
## embedded nul(s) found in input
```

## Part 2

Print the column names and the structure. If we go through the data set we see that there a lot of interesting columns like the x-y-coordinates of every given shot, the assist giver and the xG of every shot and so on.

```

names(shot_data)

## [1] "id"           "minute"        "result"         "X"
## [5] "Y"            "xG"            "player"         "h_a"
## [9] "player_id"     "situation"      "year"          "shotType"
## [13] "match_id"      "h_team"         "a_team"        "h_goals"
## [17] "a_goals"       "date"          "player_assisted" "lastAction"

str(shot_data)

## 'data.frame': 370719 obs. of  20 variables:
##   $ id           : int  16299 16302 17594 17597 17601 ...
##   $ minute       : int  34 50 19 22 38 79 88 2 10 13 ...
##   $ result       : chr  "BlockedShot" "BlockedShot" "MissedShots" "BlockedShot" ...
##   $ X            : num  0.96 0.906 0.858 0.878 0.85 ...
##   $ Y            : num  0.545 0.49 0.45 0.705 0.66 ...
##   $ xG           : num  0.0584 0.0884 0.0577 0.0434 0.042 ...
##   $ player        : chr  "Christian Benteke" "Christian Benteke" "Christian Benteke" "Christian Benteke" ...
##   $ h_a           : chr  "a" "a" "a" "a" ...
##   $ player_id     : int  606 606 606 606 606 606 606 606 ...
##   $ situation     : chr  "FromCorner" "OpenPlay" "OpenPlay" "OpenPlay" ...
##   $ year          : int  2014 2014 2014 2014 2014 2014 2014 2014 ...
##   $ shotType      : chr  "Head" "RightFoot" "Head" "RightFoot" ...
##   $ match_id      : int  4700 4700 4718 4718 4718 4718 4657 4657 4657 ...
##   $ h_team         : chr  "Everton" "Everton" "Queens Park Rangers" "Queens Park Rangers" ...
##   $ a_team         : chr  "Aston Villa" "Aston Villa" "Aston Villa" "Aston Villa" ...
##   $ h_goals        : int  3 3 2 2 2 2 2 1 1 1 ...
##   $ a_goals        : int  0 0 0 0 0 0 0 2 2 2 ...
##   $ date           : chr  "2014-10-18T15:00:00Z" "2014-10-18T15:00:00Z" "2014-10-27T20:00:00Z" "2014-10-27T20:00:00Z" ...
##   $ player_assisted: chr  "Ashley Westwood" "Aly Cissokho" "Ashley Westwood" "Andreas Weimann" ...
##   $ lastAction     : chr  "Card" "Cross" "Throughball" "Pass" ...

nrow(shot_data)

## [1] 370719

unique(shot_data$result)

## [1] "BlockedShot" "MissedShots" "SavedShot"   "ShotOnPost"  "Goal"
## [6] "OwnGoal"

unique(shot_data$situation)

## [1] "FromCorner"    "OpenPlay"      "SetPiece"     "DirectFreekick"
## [5] "Penalty"

unique(shot_data$shotType)

## [1] "Head"          "RightFoot"     "LeftFoot"     "OtherBodyPart"

```

We see that there are 370719 total shots in the data set. The results and the situations of the shots give the expected outputs. Shot types are divided in to the both foots, header and other body parts.

Let's head the first 10 rows of the data set

```
#head(shot_data, 10)
```

Let's check for missing or negative values in the data set

```

print(sapply(shot_data, function(x) sum(is.na(x))))
##          id      minute     result      X      Y
##          0          0          0          0          0
##         xG      player     h_a player_id situation
##          0          0          0          0          0
##        year shotType   match_id    h_team   a_team
##          0          0          0          0          0
##       h_goals    a_goals      date player_assisted lastAction
##          0          0          0         98605          0

```

There are only missing values for the “player\_assisted” column which is quite normal, as not every shot is assisted by someone. There are cases when the shooter gets the ball by the opponents’s defender for example. It’s quite interesting though that around 26.6% of the shots don’t have an assist. Let’s take a closer look at free kicks and penalties.

```
nrow(shot_data[(shot_data$situation == "Penalty" | shot_data$situation == "DirectFreekick"),])/nrow(sho
```

```
## [1] 0.05712683
```

Free kick and penalties make up for around 5.7% of the data

We can see that in the data set penalties and direct freekicks don’t have an assist which then explaines the high percentage of approx. 26.6%

Let’s check for any negative values

```
print(shot_data %>% select_if(~any(. < 0)))
```

```
## data frame with 0 columns and 370719 rows
```

As expected there are none negative values. For this data set negative values wouldn’t make any sense

Let’s see what percentage of the total shots taken are penalty kicks

```
nrow(shot_data[(shot_data$situation != "Penalty"),])
```

```
## [1] 366015
```

```
1-(nrow(shot_data[(shot_data$situation != "Penalty"),])/nrow(shot_data))
```

```
## [1] 0.01268886
```

The total amount of given shots excluding penalties is 366015 and therefore penalties make up for 1.27% of total given shots

Here we make a new column and bin there different ranges for the xG data

```
shot_data<-shot_data%>%mutate(xG_Ranges = cut(xG, breaks = c(-Inf, 0.1, 0.2, 0.4, 0.65, 0.85, 1)))
shot_data$xG_Ranges <- as.character(shot_data$xG_Ranges)
```

But before that let’s quickly get the xG-probability for penalties within these 8 seasons

```
mean(shot_data$xG[shot_data$situation=="Penalty"])
```

```
## [1] 0.7564446
```

The probability of a penalty resulting in a goal is 75.64% for the top 5 leagues. This is an expected value but some might be surprised that it isn’t higher.

We are only interested for shots who were given after midfield. There might be goals scored from the own yard but outliers don’t interest us in this model. But we for sure will include all shots given, independent

of their position when we will analyze the conversion rate of all given shots per player. We also exclude penalties here, as we already calculated the xG for penalties.

```
areasDf <- shot_data[(shot_data$X >= 0.5 & shot_data$situation != "Penalty"),]
```

In this step we will divide the playfield into different areas in order to analyze the xG of these sections and compare the players qualities within these areas. Therefore the field must be divided accordingly with the right x-y-values within the coordinate system of the field.

```
areasDf <- areasDf %>%
  mutate(xGAreas = case_when(X < 0.8294 & Y < 0.211 ~ 'Area 10',
                             X < 0.8294 & Y > 0.789 ~ 'Area 10',
                             X < 0.7163 & Y >= 0.211 & Y <= 0.789 ~ 'Area 9',
                             X >= 0.8294 & Y < 0.211 ~ 'Area 8',
                             X >= 0.8294 & Y > 0.789 ~ 'Area 8',
                             X >= 0.7163 & X < 0.8294 & Y >= 0.211 & Y <= 0.3678 ~ 'Area 7',
                             X >= 0.7163 & X < 0.8294 & Y >= 0.6322 & Y <= 0.789 ~ 'Area 7',
                             X >= 0.7163 & X < 0.8294 & Y > 0.3678 & Y < 0.6322 ~ 'Area 6',
                             X >= 0.8294 & Y >= 0.211 & Y < 0.3678 ~ 'Area 5',
                             X >= 0.8294 & Y >= 0.6322 & Y <= 0.789 ~ 'Area 5',
                             X >= 0.8294 & X <= 0.8846 & Y > 0.3678 & Y < 0.6322 ~ 'Area 4',
                             X > 0.8846 & X < 0.9423 & Y > 0.3678 & Y < 0.6322 ~ 'Area 3',
                             X >= 0.9423 & Y > 0.3678 & Y < 0.44222 ~ 'Area 2',
                             X >= 0.9423 & Y > 0.55778 & Y < 0.6322 ~ 'Area 2',
                             X >= 0.9423 & Y >= 0.44222 & Y <= 0.55778 ~ 'Area 1',
                             TRUE ~ 'e'))
```

unique(sort(areasDf\$xGAreas))

```
## [1] "Area 1"  "Area 10" "Area 2"  "Area 3"  "Area 4"  "Area 5"  "Area 6"
## [8] "Area 7"  "Area 8"  "Area 9"
```

The output of the unique values of the xG areas shows us that there were no errors in the condition model. The error would be marked with the value “e”

We have to create a vector with our x-y-coordinates to plot in the field map. Therefore we assign the real values to variables to have a better readability of the vectors.

```
xA = 0.5
xB = 0.8294
xC = 0.9423
xD = 0.8846
xE = 0.7163
x1 = 1

yA = 0.211
yB = 0.789
yC = 0.3678
yD = 0.44222
yE = 0.55778
yF = 0.6322
y0 = 0
```

We have to create a vector with our x-y-coordinates to plot in the field map

```

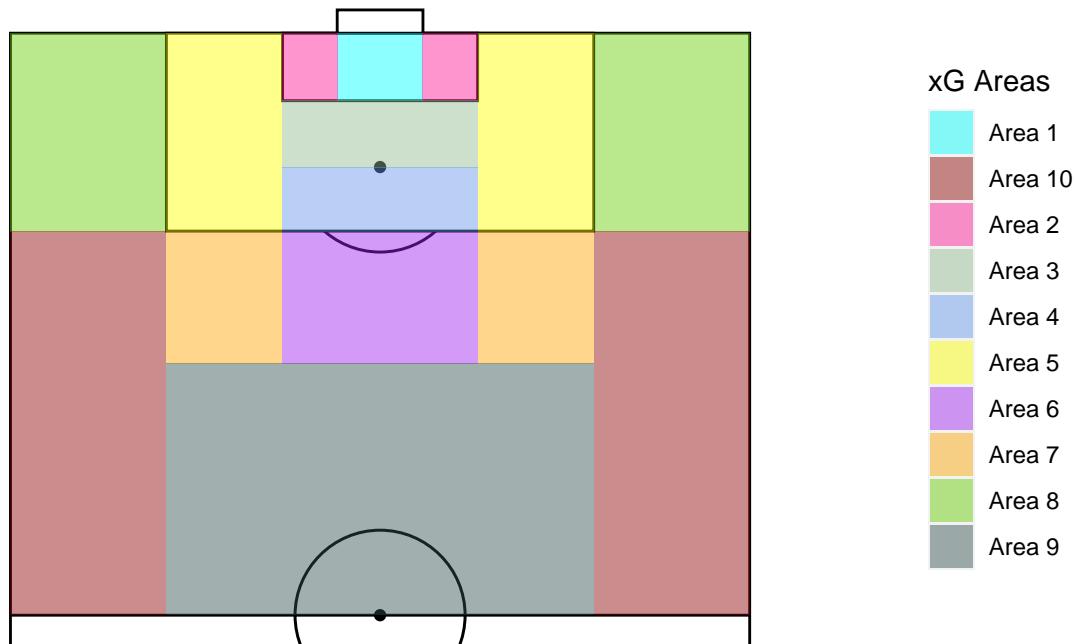
xmin <- c(xC, xC, xC, xD, xB, xB, xE, xE, xB, xB, xA, xA, xA) *100
xmax <- c(x1, x1, x1, xC, xD, x1, x1, xB, xB, x1, x1, xE, xB, xB) *100
ymin <- c(yD, yE, yC, yC, yF, yA, yC, yF, yA, yB, y0, yA, yB, y0) *100
ymax <- c(yE, yF, yD, yF, yB, yC, yF, yB, yC, x1, yA, yB, x1, yA) *100
xGAreas <- c("Area 1", "Area 2", "Area 2", "Area 3", "Area 4", "Area 5", "Area 5", "Area 6", "Area 7",

xG_plot_areas <- data.frame(xmin, xmax, ymin, ymax, xGAreas)
rm(xA, xB, xC, xD, xE, x1, yA, yB, yC, yD, yE, yF, y0, xmin, xmax, ymin, ymax)

pltTitle <- "Expected Goals (xG) divided into different areas"
plot_areas_xG(xG_plot_areas, xGAreas, pltTitle)

```

## Expected Goals (xG) divided into different areas



Let's have a look at the average xG of the different xG areas

```

xG_avrg <- areasDf %>%
  group_by(xGAreas) %>%
  summarise(across(xG, mean, na.rm = TRUE))

xG_avrg[order(-xG_avrg$xG),]

## # A tibble: 10 x 2
##   xGAreas      xG
##   <chr>     <dbl>
## 1 Area 1  0.499
## 2 Area 2  0.212
## 3 Area 3  0.169
## 4 Area 4  0.106
## 5 Area 5  0.0746
## 6 Area 8  0.0402
## 7 Area 6  0.0382
## 8 Area 7  0.0269
## 9 Area 10 0.0192

```

```
## 10 Area 9 0.0172
```

As expected the xG areas nearer to goal are significantly higher. Interesting to see is that the area 4 has a higher xG as the area 5, even though area 5 is nearer to the goal. The reason for this is that the degree of the shot position to the goal is really important for a xG model.

Now we will add the average xG to the “xG\_plot\_areas” dataframe, according to the xG areas

```
xG_plot_areas <- xG_plot_areas %>% full_join(xG_avrg)
```

```
## Joining, by = "xGAreas"  
rm(xG_avrg)
```

We prepare the data for the fieldmap and also format our xG probabilities for a nicer display

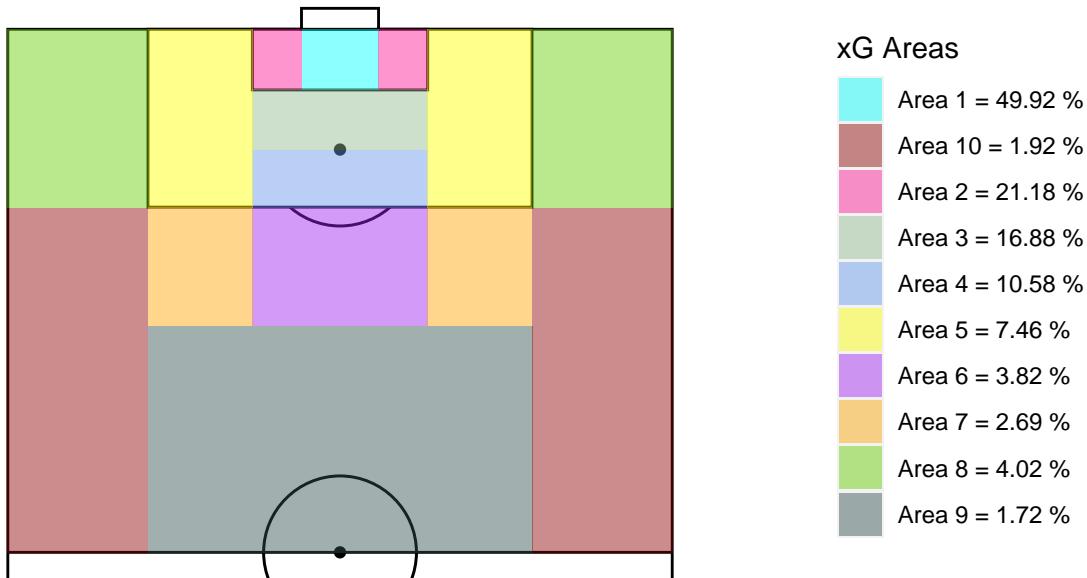
```
xG_plot_areas$xG <- xG_plot_areas$xG * 100  
xG_plot_areas$xG <- round(xG_plot_areas$xG, digits = 2)  
xG_plot_areas <- xG_plot_areas %>%  
  unite('xGChar', xGAreas:xG, sep= " = ",  
        remove = FALSE)  
xG_plot_areas$xG <- xG_plot_areas$xG / 100  
xG_plot_areas$xGChar <- paste(xG_plot_areas$xGChar, "%")  
head(xG_plot_areas)
```

```
##      xmin    xmax    ymin    ymax          xGChar xGAreas      xG  
## 1 94.23 100.00 44.222 55.778 Area 1 = 49.92 % Area 1 0.4992  
## 2 94.23 100.00 55.778 63.220 Area 2 = 21.18 % Area 2 0.2118  
## 3 94.23 100.00 36.780 44.222 Area 2 = 21.18 % Area 2 0.2118  
## 4 88.46  94.23 36.780 63.220 Area 3 = 16.88 % Area 3 0.1688  
## 5 82.94  88.46 36.780 63.220 Area 4 = 10.58 % Area 4 0.1058  
## 6 82.94 100.00 63.220 78.900 Area 5 = 7.46 % Area 5 0.0746  
xGChar <- xG_plot_areas$xGChar
```

Now we will plot the fieldmap with the xG probabilities for the different areas

```
pltTitle <- "Expected Goals (xG) divided into different areas with it's probabilities"  
plot_areas_xG(xG_plot_areas, xGChar, pltTitle)
```

## Expected Goals (xG) divided into different areas with it's probabilities



Some data manipulations

```
get_amount_group <- function(df, strColName, strNewCol) {
  df <- df %>%
    group_by(df[strColName]) %>%
    add_count(name = strNewCol)
  df$areaShots <- stri_c(df[strColName], ' = ', df[strNewCol])
  return(df)
}

dfTemp <- areasDf
dfTemp <- get_amount_group(dfTemp, "result", "shots_distribution")

## Warning in stri_c(..., sep = sep, collapse = collapse, ignore_null = TRUE):
## argument is not an atomic vector; coercing

## Warning in stri_c(..., sep = sep, collapse = collapse, ignore_null = TRUE):
## argument is not an atomic vector; coercing

head(dfTemp, 10)

## # A tibble: 10 x 24
## # Groups:   result [4]
##       id minute result      X      Y      xG player h_a playe~1 situa~2 year
##       <int>   <int> <chr>   <dbl> <dbl>  <dbl> <chr> <chr> <int> <chr> <int>
## 1 16299     34 BlockedS~  0.96  0.545  0.0584 Chris~ a      606 FromCo~ 2014
## 2 16302     50 BlockedS~  0.906 0.49   0.0884 Chris~ a      606 OpenPl~ 2014
## 3 17594     19 MissedSh~  0.858 0.45   0.0577 Chris~ a      606 OpenPl~ 2014
## 4 17597     22 BlockedS~  0.878 0.705  0.0434 Chris~ a      606 OpenPl~ 2014
## 5 17601     38 SavedShot  0.85  0.66   0.0420 Chris~ a      606 OpenPl~ 2014
## 6 17610     79 MissedSh~  0.912 0.456  0.0321 Chris~ a      606 FromCo~ 2014
## 7 17614     88 MissedSh~  0.705 0.583  0.00822 Chris~ a     606 OpenPl~ 2014
## 8 16881      2 BlockedS~  0.897 0.658  0.0613 Chris~ h      606 OpenPl~ 2014
## 9 16884     10 MissedSh~  0.903 0.472  0.242  Chris~ h      606 OpenPl~ 2014
## 10 16887    13 ShotOnPo~  0.853 0.54   0.0900 Chris~ h     606 OpenPl~ 2014
```

```

## # ... with 13 more variables: shotType <chr>, match_id <int>, h_team <chr>,
## #   a_team <chr>, h_goals <int>, a_goals <int>, date <chr>,
## #   player_assisted <chr>, lastAction <chr>, xG_Ranges <chr>, xGAreas <chr>,
## #   shots_distribution <int>, areaShots <chr>, and abbreviated variable names
## #   1: player_id, 2: situation

```

Format the data set

```
dfTemp["shots_distribution"] <- so_formatter(dfTemp["shots_distribution"])
```

We prepare the data for the fieldmap with total shots per area and also format the numbers for a nicer display

```
dfTemp <- areasDf
```

```
df_test <- dfTemp %>% group_by(xGAreas) %>% summarize(count=n())
assert_lengths(sum(df_test$count), nrow(areasDf))
df_test$count <- so_formatter(df_test$count)
dfTemp <- dfTemp %>% full_join(df_test)
```

```
## Joining, by = "xGAreas"
```

```
dfTemp$areaShots <- str_c(dfTemp$xGAreas, ' = ', dfTemp$count)
areaShots <- dfTemp[,c("xGAreas", "areaShots")]
areaShots = areaShots[!duplicated(areaShots), ]
xG_plot_areas <- xG_plot_areas %>% full_join(areaShots)
```

```
## Joining, by = "xGAreas"
```

```
rm(areaShots, df_test)
areaShots <- xG_plot_areas$areaShots
xG_plot_areas
```

	xmin	xmax	ymin	ymax	xGChar	xGAreas	xG	areaShots
## 1	94.23	100.00	44.222	55.778	Area 1 = 49.92 %	Area 1	0.4992	Area 1 = 13.413K
## 2	94.23	100.00	55.778	63.220	Area 2 = 21.18 %	Area 2	0.2118	Area 2 = 12.036K
## 3	94.23	100.00	36.780	44.222	Area 2 = 21.18 %	Area 2	0.2118	Area 2 = 12.036K
## 4	88.46	94.23	36.780	63.220	Area 3 = 16.88 %	Area 3	0.1688	Area 3 = 77.307K
## 5	82.94	88.46	36.780	63.220	Area 4 = 10.58 %	Area 4	0.1058	Area 4 = 47.83K
## 6	82.94	100.00	63.220	78.900	Area 5 = 7.46 %	Area 5	0.0746	Area 5 = 68.371K
## 7	82.94	100.00	21.100	36.780	Area 5 = 7.46 %	Area 5	0.0746	Area 5 = 68.371K
## 8	71.63	82.94	36.780	63.220	Area 6 = 3.82 %	Area 6	0.0382	Area 6 = 72.631K
## 9	71.63	82.94	63.220	78.900	Area 7 = 2.69 %	Area 7	0.0269	Area 7 = 52.024K
## 10	71.63	82.94	21.100	36.780	Area 7 = 2.69 %	Area 7	0.0269	Area 7 = 52.024K
## 11	82.94	100.00	78.900	100.000	Area 8 = 4.02 %	Area 8	0.0402	Area 8 = 1.886K
## 12	82.94	100.00	0.000	21.100	Area 8 = 4.02 %	Area 8	0.0402	Area 8 = 1.886K
## 13	50.00	71.63	21.100	78.900	Area 9 = 1.72 %	Area 9	0.0172	Area 9 = 16.53K
## 14	50.00	82.94	78.900	100.000	Area 10 = 1.92 %	Area 10	0.0192	Area 10 = 2.272K
## 15	50.00	82.94	0.000	21.100	Area 10 = 1.92 %	Area 10	0.0192	Area 10 = 2.272K

Now we will look at all the shots forward the midfield and draw the the associated areas

```
dfTemp <- dfTemp[,c("X", "Y", "xG", "result", "xGAreas", "areaShots")]
dfTemp$X <- dfTemp$X * 100
dfTemp$Y <- dfTemp$Y * 100
print(sum(is.na(dfTemp$xGAreas)))
```

```
## [1] 0
```

We also double check if the column “xGAreas” has some missing value, which is not the case

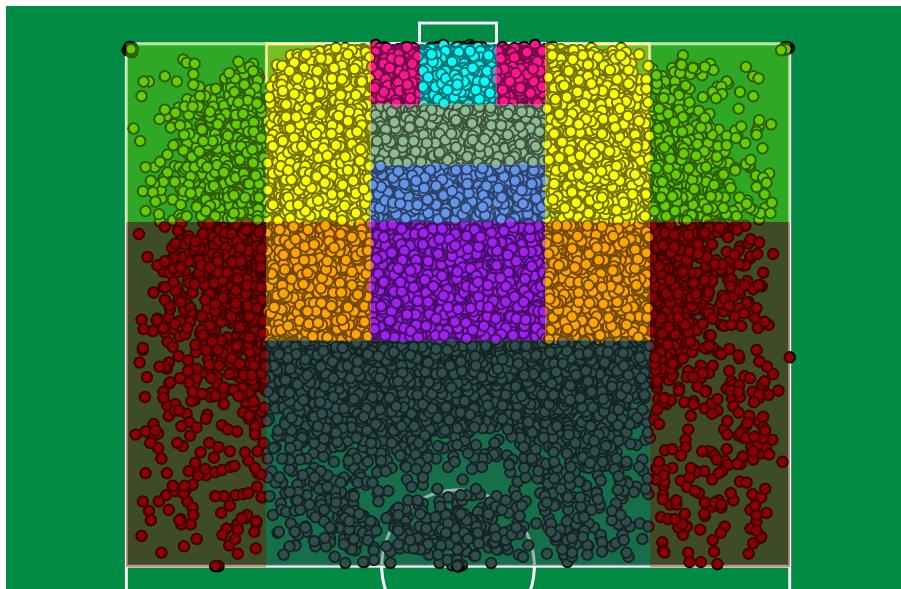
Let's plot all the shots given from the midfield and categorize them into our pre-defined areas

```

ggplot(dfTemp) +
  annotate_pitch(colour = "white",
                 fill    = "springgreen4",
                 limits = FALSE) +
  geom_point(aes(x = X, y = 100 - Y, fill = areaShots),
             shape = 21,
             size = 1.5) +
  geom_rect(data = xG_plot_areas, inherit.aes=FALSE, aes(xmin = xmin, xmax = xmax, ymin = ymin, ymax = ymax,
                                                         fill = areaShots), colour=NA, alpha=.45) +
  scale_fill_manual('xG Areas',
                    values=c("cyan", "darkred", "deeppink", "darkseagreen", "cornflowerblue", "yellow",
                            "lightblue", "lightgreen", "lightorange", "lightpurple"),
                    guide = guide_legend(override.aes = list(alpha = 0.45))) +
  theme_pitch() +
  theme(panel.background = element_rect(fill = "springgreen4")) +
  coord_flip(xlim = c(50, 101),
             ylim = c(-12, 112)) +
  ggtitle("All shots divided into different xG areas")

```

All shots divided into different xG areas



Most shots were taken in Area 6 which is not that surprising. It might be interesting that so many shots were taken in Area 9.

```
dfTemp = data.frame()
```

## Part 3 Results

In the next steps we will group the players and filter out players who scored less than 120 goals from the season 2014/2015 to 2021/2022, the reason behind this filter is to rank and find the best scoring players. The players had to score at least 15 goals in average per seasons, as we have 8 seasons.

```
t=120
df_best <- get_goals(FALSE, shot_data, t)
```

```

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

listPlayersFinal <- df_best$player
df_best

## # A tibble: 19 x 4
## # Groups:   player_id, player [19]
##   player_id player          result count
##       <int> <chr>        <chr>  <int>
## 1      227 Robert Lewandowski Goal    238
## 2      2097 Lionel Messi     Goal    237
## 3      2371 Cristiano Ronaldo Goal    233
## 4      647 Harry Kane       Goal    180
## 5      2098 Luis Suárez     Goal    179
## 6      318 Pierre-Emerick Aubameyang Goal    163
## 7      1209 Ciro Immobile   Goal    160
## 8      1250 Mohamed Salah   Goal    153
## 9      2370 Karim Benzema   Goal    147
## 10     594 Romelu Lukaku   Goal    136
## 11     3423 Kylian Mbappe-Lottin Goal    136
## 12     3294 Edinson Cavani  Goal    134
## 13     619 Sergio Agüero   Goal    133
## 14     755 Jamie Vardy     Goal    133
## 15     3210 Wissam Ben Yedder Goal    132
## 16     3277 Alexandre Lacazette Goal    130
## 17     2099 Neymar         Goal    127
## 18     1513 Mauro Icardi   Goal    125
## 19     2290 Iago Aspas     Goal    122

```

The output shows us that Robert Lewandowski was the player with the most goals counting 238, closely followed by Lionel Messi and Cristiano Ronaldo. Reminder: This output contains also penalties, we will look at the numbers without penalties in a moment.

As penalties distort the results we decided to filter them out. Thus we lowered the threshold of 120 goals to 103 goals which means that the players must have a scored at least approx. 13 goals on average per season without penalties.

```

t=100
df_best <- get_goals(TRUE, shot_data, t)

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

listPlayersFinal <- df_best$player
df_best

## # A tibble: 21 x 4
## # Groups:   player_id, player [21]
##   player_id player          result count
##       <int> <chr>        <chr>  <int>
## 1      2097 Lionel Messi     Goal    209
## 2      227 Robert Lewandowski Goal    203
## 3      2371 Cristiano Ronaldo Goal    182
## 4      2098 Luis Suárez     Goal    165
## 5      647 Harry Kane       Goal    152
## 6      318 Pierre-Emerick Aubameyang Goal    144

```

```

## 7      1250 Mohamed Salah          Goal     135
## 8      2370 Karim Benzema         Goal     129
## 9      3423 Kylian Mbappe-Lottin Goal     125
## 10     594 Romelu Lukaku        Goal     119
## # ... with 11 more rows

```

The output shows us that, without considering penalties, Lionel Messi was the player with the most goals counting 203, closely followed by Robert Lewandoski. As expected Cristiano Ronaldo takes the third place.

We can also output the players with the most Penalty and Free kick goals.

```

penalties_df <- shot_data[(shot_data$situation == "Penalty"),]
freeskicks_df <- shot_data[(shot_data$situation == "DirectFreekick"),]

t = 0
df_best<-get_goals(FALSE, penalties_df, t)

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

listPlayers <- df_best$player[1:10]
head(df_best, 10)

## # A tibble: 10 x 4
## # Groups:   player_id, player [10]
##   player_id player      result count
##       <int> <chr>      <chr> <int>
## 1     2371 Cristiano Ronaldo  Goal     51
## 2     1209 Ciro Immobile    Goal     44
## 3     227 Robert Lewandowski Goal     35
## 4     1230 Fabio Quagliarella Goal     31
## 5     1612 Domenico Berardi  Goal     29
## 6     647 Harry Kane        Goal     28
## 7     2097 Lionel Messi     Goal     28
## 8     2099 Neymar           Goal     28
## 9     3210 Wissam Ben Yedder Goal     28
## 10    1393 Lorenzo Insigne  Goal     27

```

After looking more into detail, we see that Ciro Immobile is also a player who benefits a lot from penalties.

Now we want to get all interesting key figures behind penalty data. Herefore we take the best 10 player who scored the most penalties. Via a pre-defined function we will now determine which player has the best Conversion Rate, Over/Underperformance of the xG and so on. The same procedure is followed when we analyzing free kicks.

```
col_header <- c("Player", "Conversion_Rate", "Goals_Expected", "Goals", "Shots", "Average_xG", "Performance")
```

Let's have a deeper look into all penalties given by the top 10 player and get the results.

```
df_best <- setNames(data.frame(matrix(ncol =length(col_header), nrow = 0)), col_header)
for (strName in listPlayers) {
```

```
  df_best <-results_standards(TRUE, shot_data, df_best, strName)
}
df_best
```

	Player	Conversion_Rate	Goals_Expected	Goals	Shots	Average_xG
## 3	Robert Lewandowski	0.921	28.80	35	38	0.758
## 6	Harry Kane	0.903	23.59	28	31	0.761

```

## 1 Cristiano Ronaldo      0.810      47.38     51     63     0.752
## 4 Fabio Quagliarella   0.861      27.40     31     36     0.761
## 8 Neymar                0.848      24.91     28     33     0.755
## 2 Ciro Immobile         0.815      41.09     44     54     0.761
## 9 Wissam Ben Yedder     0.824      25.81     28     34     0.759
## 10 Lorenzo Insigne      0.794      25.87     27     34     0.761
## 5 Domenico Berardi      0.784      28.16     29     37     0.761
## 7 Lionel Messi           0.718      28.98     28     39     0.743
##   Performance_xG Performance_Goals
## 3             0.163          6.20
## 6             0.142          4.41
## 1             0.058          3.62
## 4             0.100          3.60
## 8             0.093          3.09
## 2             0.054          2.91
## 9             0.065          2.19
## 10            0.033          1.13
## 5             0.023          0.84
## 7             -0.025         -0.98

```

From a purely statistical point of view we can see that Robert Lewandowski actually is the best penalty taker by the top 10 penalty goalscorers. He scored approx. 6 goals more by penalty regarding his xG value. The approx. number 6 is the difference between Expected Goals in total and the actual amount of scored Goals. He also has the best Conversion Rate. Lionel Messi under performed in this discipline. He scored approx. 1 penalty less as he should according to his xG value.

Interpreting the Conversion Rates means that if we take Lewandowski as example, we can conclude that statistically, he scores around 92 goals in 100 penalties. In contrast to that Messi only scores around 72 goals.

Let's do the same for freekicks

```
df_best<-get_goals(TRUE, freekicks_df, t)
```

```

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

```

```
listPlayers <- df_best$player[1:10]
head(df_best, 10)
```

```

## # A tibble: 10 x 4
## # Groups:   player_id, player [10]
##   player_id player      result count
##       <int> <chr>      <chr>  <int>
## 1     2097 Lionel Messi    Goal     28
## 2      843 James Ward-Prowse  Goal     14
## 3     1290 Miralem Pjanic   Goal     11
## 4     2378 Daniel Parejo    Goal     11
## 5      189 Hakan Calhanoglu Goal     10
## 6      621 Aleksandar Kolarov Goal     10
## 7     1294 Paulo Dybala    Goal     10
## 8      646 Christian Eriksen Goal      8
## 9     1445 Simone Verdi     Goal      8
## 10    3667 Nabil Fekir      Goal      8

```

The player with the most free kick goals is by far Lionel Messi followed by James Ward-Prowse. There are quite a few players with 10 or 11 free kick goals. We now will output the amount of free kicks taken by the top 10 of these players.

Let's have a deeper look into all free kicks given by the top 10 player and get the results.

```
df_best <- setNames(data.frame(matrix(ncol =length(col_header), nrow = 0)), col_header)
for (strName in listPlayers) {

  df_best <-results_standards(FALSE, shot_data, df_best, strName)
}

df_best

##           Player Conversion_Rate Goals_Expected Goals Shots Average_xG
## 2   James Ward-Prowse      0.141        6.14     14    99  0.062
## 1   Lionel Messi          0.093       21.00     28   300  0.070
## 3   Miralem Pjanic        0.125        5.54     11    88  0.063
## 7   Paulo Dybala          0.130        4.93     10    77  0.064
## 6 Aleksandar Kolarov      0.115        5.22     10    87  0.060
## 4   Daniel Parejo          0.096        6.90     11   115  0.060
## 9   Simone Verdi           0.100        4.48      8    80  0.056
## 10  Nabil Fekir            0.092        4.70      8    87  0.054
## 5   Hakan Calhanoglu       0.071        7.33     10   141  0.052
## 8   Christian Eriksen      0.067        6.78      8   119  0.057
##   Performance_xG Performance_Goals
## 2           0.079          7.86
## 1           0.023          7.00
## 3           0.062          5.46
## 7           0.066          5.07
## 6           0.055          4.78
## 4           0.036          4.10
## 9           0.044          3.52
## 10          0.038          3.30
## 5           0.019          2.67
## 8           0.010          1.22
```

No player has underperformed considering their xG value. James Ward-Prowse has the best Total Goal Performance and Conversion Rate (around 14 free kick goals in 100 attempts, from a statistical pov.). He succeeded his total amount of goals expected by approx. 8 goals followed by Lionel Messi who succeeded his expected amount of goals by 7 goals. If we consider the “Performance\_xG” column (“Conversion\_rate” - “Average\_xG”) we see that Lionel Messi is not the second best player. This column is a pure statistical ratio which is also very important as it doesn’t consider the amount of goals. But as we here talking still about football, the sheer amount of goals scored is one of the most important, if not the most skill for a goal scorer. Thus we decided to take this metric as our primary performance metric. This methodology will hold on also for the following analysis.

```
listPlayersFinal
```

```
## [1] "Lionel Messi"                  "Robert Lewandowski"
## [3] "Cristiano Ronaldo"             "Luis Suárez"
## [5] "Harry Kane"                   "Pierre-Emerick Aubameyang"
## [7] "Mohamed Salah"                 "Karim Benzema"
## [9] "Kylian Mbappe-Lottin"          "Romelu Lukaku"
## [11] "Ciro Immobile"                "Edinson Cavani"
## [13] "Antoine Griezmann"             "Sergio Agüero"
## [15] "Sadio Mané"                   "Jamie Vardy"
## [17] "Gonzalo Higuaín"              "Mauro Icardi"
## [19] "Wissam Ben Yedder"             "Son Heung-Min"
## [21] "Alexandre Lacazette"
```

These are the 21 players who made it into our list. We decided to take also players like Erling Haaland and Zlatan Ibrahimovic into the list because as we all now these players are prolific goal scorers whose time window of their prime unfortunately not lies in these 8 seasons (2014/2015 to 2021/2022) we are looking at. Also both of them didn't fully play in the top 5 leagues. Ibrahimovic played in the MLS for almost 2 years and Haaland was transferred to Borussia Dortmund at the beginning of 2020.

We also took Neymar (127 goals) and Iago Aspas (125 goals) into our list as they would have made it into the list in the first place as we considered players with 120 goals or more including penalties. Although we think that penalties distort our model we have to admit that being good in penalties is a very important skill set for a goal scorer and thus decided to include these 2 players too.

Antoine Griezmann had 119 goals which is the reason we also included him in the list, as he had only had 1 goal less than the threshold of 120 goals

```
listPlayers2 <- c("Erling Haaland", "Zlatan Ibrahimovic", "Neymar", "Iago Aspas")
listPlayersFinal <- append(listPlayersFinal, listPlayers2)
rm(listPlayers2)
```

Via a pre-defined function we will now determine which player has the best Conversion Rate and xG Performance for the different xG areas and ranges in the following statistics.

Let's have a look into all shots given by the player and get the results for the last season 2021/2022.

Let's have a look into all shots given by the player and get the results.

```
lowTshld = 0.0000
upTshld = 1.000
df_best <- setNames(data.frame(matrix(ncol = length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {

  df_best <- results_ranges(TRUE, shot_data, df_best, strName, lowTshld, upTshld)
}
listPlayers <- df_best$Player[1:n]
df_best
```

	Player	Conversion_Rate	Goals_Expected	Goals	Shots
## 1	Lionel Messi	0.162	173.40	209	1294
## 20	Son Heung-Min	0.182	73.45	103	565
## 13	Antoine Griezmann	0.185	90.14	112	605
## 9	Kylian Mbappe-Lottin	0.213	108.19	125	588
## 25	Iago Aspas	0.179	78.44	95	530
## 5	Harry Kane	0.154	135.63	152	990
## 11	Ciro Immobile	0.147	101.78	116	789
## 17	Gonzalo Higuaín	0.153	93.56	106	693
## 22	Erling Haaland	0.277	41.92	54	195
## 23	Zlatan Ibrahimovic	0.164	75.93	87	531
## 21	Alexandre Lacazette	0.193	92.02	103	535
## 4	Luis Suárez	0.194	167.26	178	919
## 7	Mohamed Salah	0.161	124.32	135	840
## 19	Wissam Ben Yedder	0.197	96.62	104	528
## 16	Jamie Vardy	0.191	99.68	107	560
## 15	Sadio Mané	0.177	104.25	111	628
## 10	Romelu Lukaku	0.171	114.68	119	695
## 14	Sergio Agüero	0.160	106.72	111	693
## 18	Mauro Icardi	0.195	100.79	105	539
## 3	Cristiano Ronaldo	0.131	179.31	182	1390
## 8	Karim Benzema	0.171	128.52	129	756

## 6	Pierre-Emerick Aubameyang	0.202	146.37	144	714
## 2	Robert Lewandowski	0.191	209.80	203	1065
## 12	Edinson Cavani	0.209	123.21	116	555
## 24	Neymar	0.157	107.90	99	631
## Average_xG Performance_xG Performance_Goals					
## 1	0.134	0.028	35.60		
## 20	0.130	0.052	29.55		
## 13	0.149	0.036	21.86		
## 9	0.184	0.029	16.81		
## 25	0.148	0.031	16.56		
## 5	0.137	0.017	16.37		
## 11	0.129	0.018	14.22		
## 17	0.135	0.018	12.44		
## 22	0.215	0.062	12.08		
## 23	0.143	0.021	11.07		
## 21	0.172	0.021	10.98		
## 4	0.182	0.012	10.74		
## 7	0.148	0.013	10.68		
## 19	0.183	0.014	7.38		
## 16	0.178	0.013	7.32		
## 15	0.166	0.011	6.75		
## 10	0.165	0.006	4.32		
## 14	0.154	0.006	4.28		
## 18	0.187	0.008	4.21		
## 3	0.129	0.002	2.69		
## 8	0.170	0.001	0.48		
## 6	0.205	-0.003	-2.37		
## 2	0.197	-0.006	-6.80		
## 12	0.222	-0.013	-7.21		
## 24	0.171	-0.014	-8.90		

The results show that Lionel Messi has the best numbers. He succeed his total amount of expected goals by approx. 36 goals, followed by Heung-Min Son with approx. 30 goals and Antoine Griezmann (approx. 22). It's important to note that Lionel Messi had an advantage as he also has the most goals scored (209) and thus his total number of "Performance\_Goals" will be higher/lower tendentiously. But as we said before our purpose of the model was to take the total amount of goals into consideration as it is the most important number for a striker. Please refer back to the free kicks/penalties results above where we explained the model metrics more in detail.

Neymar, Edinson Cavani and Robert Lewandowski didn't performed so well in our model. As for Messi, Lewandowski also has much more goals (203) than the average player on the list and thus his "Performance\_Goals" value will be tendentiously higher/lower. Next to these 3 players Pierre-Emerick Aubameyang is the only player who scored less goals than expected, approx. 2 goals.

Cristiano Ronaldo has a positive goal amount of value of approx. 3 goals which positions him on the 20th place of 25 players. As for Messi and Lewandowski, we also have to consider here, his sheer total amount of goals (182), which is over average. As expected these 3 players and also Harry Kane take with ease the most shots.

From a purely statistical point of view Erling Haaland has the best performance with a "Performance\_xG" value of 6.2%, followed by Son with 5.2%. He also has the best Conversion Rate of 27.7%, which means that approx. every 4th shot of his finds his way behind the net. Interesting to note is that Edinson Cavani has the third best Conversion of 2.09% which means that approx. every 5th shot results in a goal.

When focusing on the Average xG, it might give some insight about which players has a smart shot/position selection. Cavani has the highest Average xG with 22.2% which we can interpret as either he has tendentiously

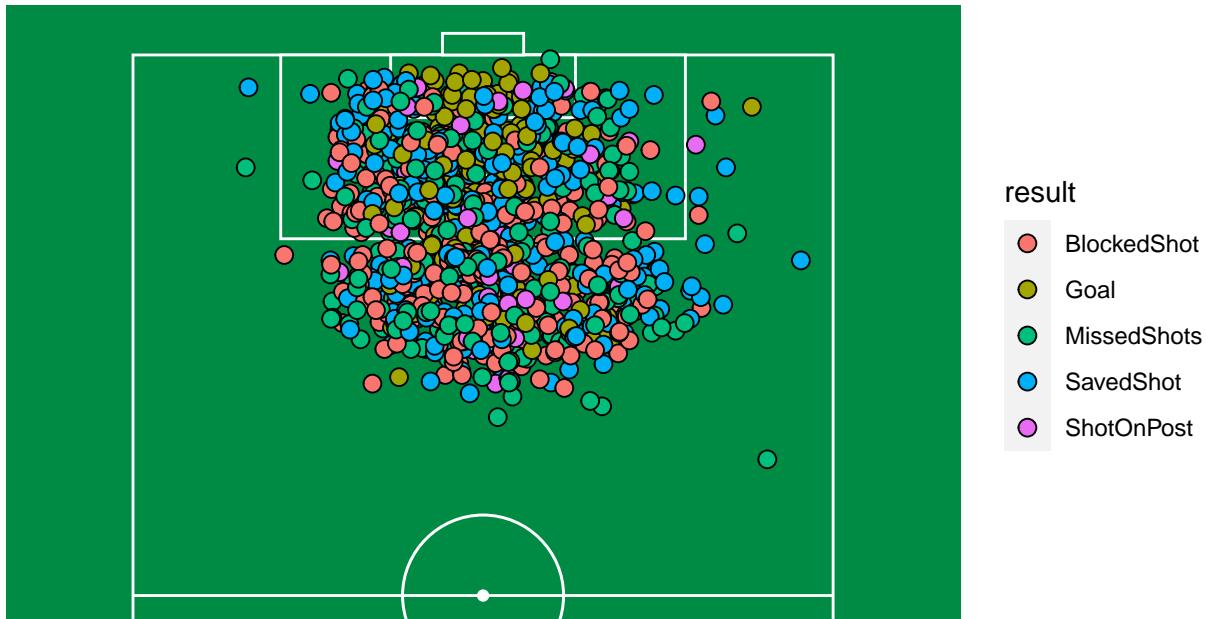
the smartest shot/position selection or that he tendentiously played in teams that provided him with good shot positions in average.

As Haaland is a young player it will be quite interesting to see, if he we will be able to outperform Lionel Messi in the Total Goals Performance (“Performance\_Goals”) in the future.

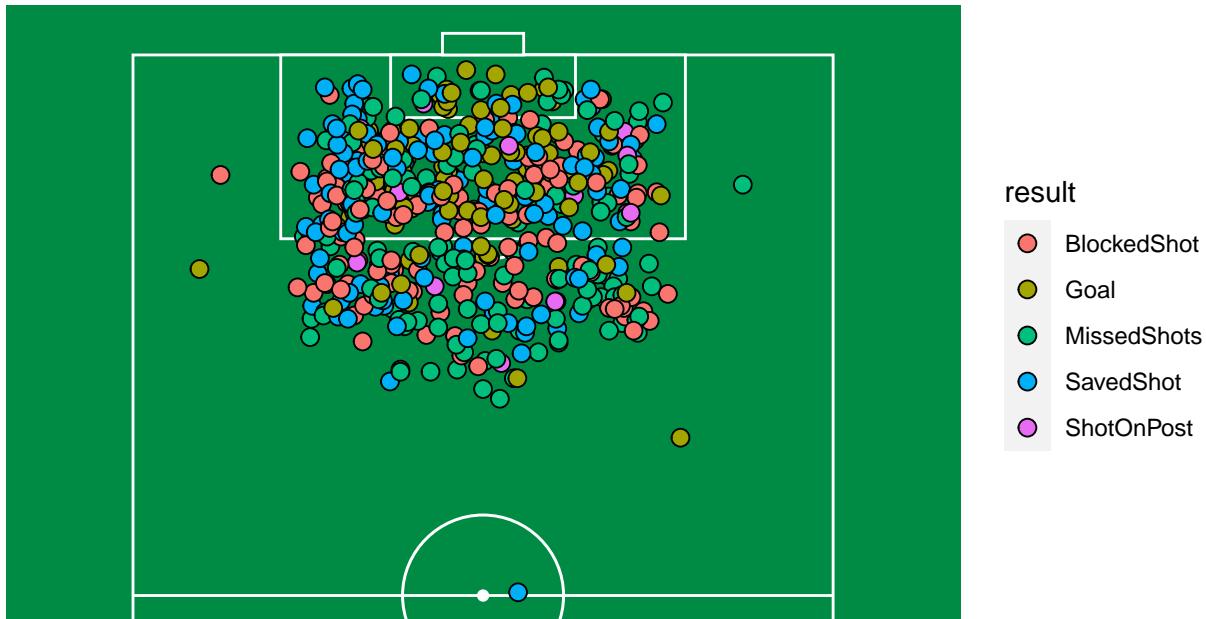
Let's look at the shotmap of the top three player in our list.

```
for (strName in listPlayers) {  
  print(plot_shot_player(TRUE , shot_data, strName, lowTshld, upTshld))  
}
```

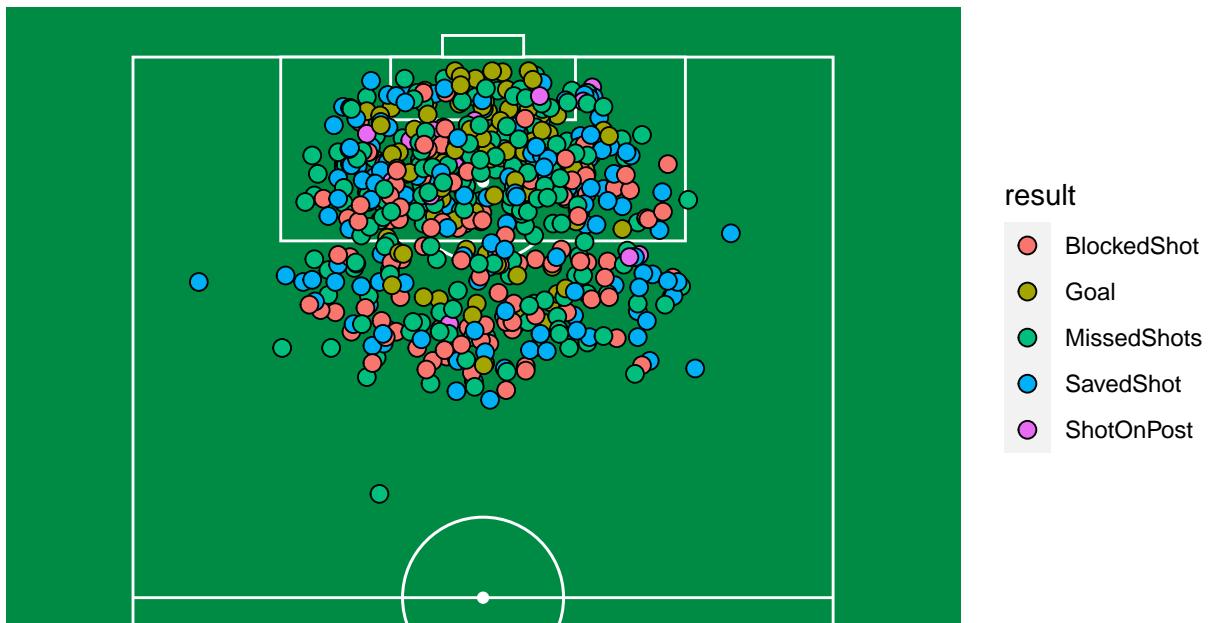
Lionel Messi shotmap where Expected Goals range between 0% – 100%



Son Heung–Min shotmap where Expected Goals range between 0% – 100%



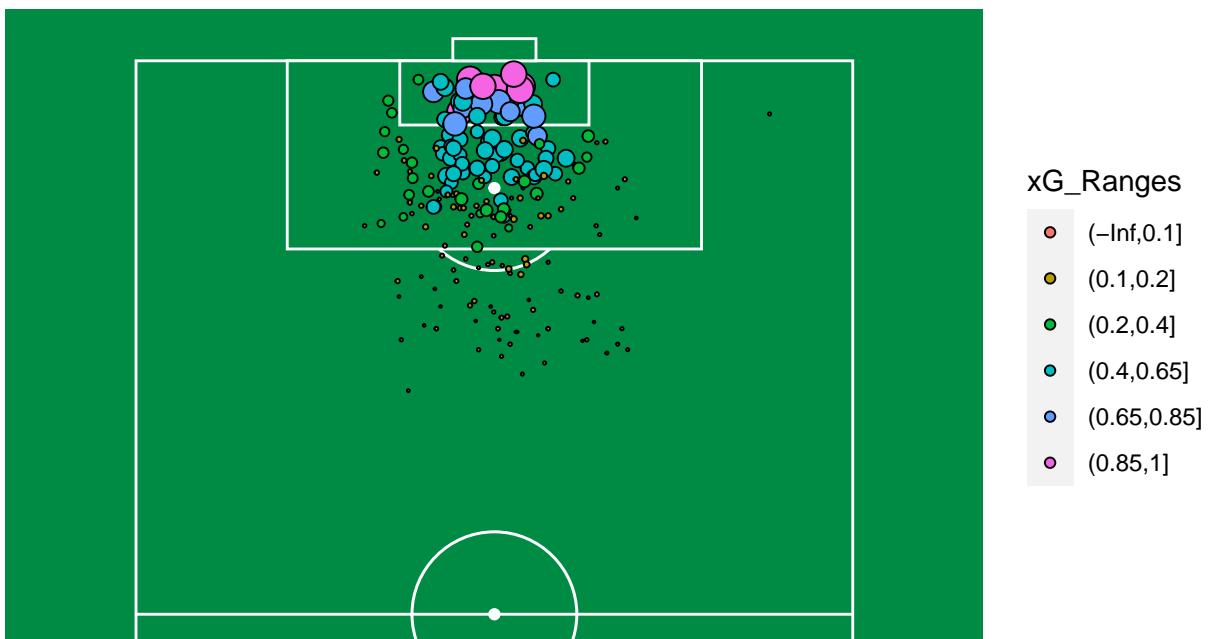
Antoine Griezmann shotmap where Expected Goals range between 0% – 100%



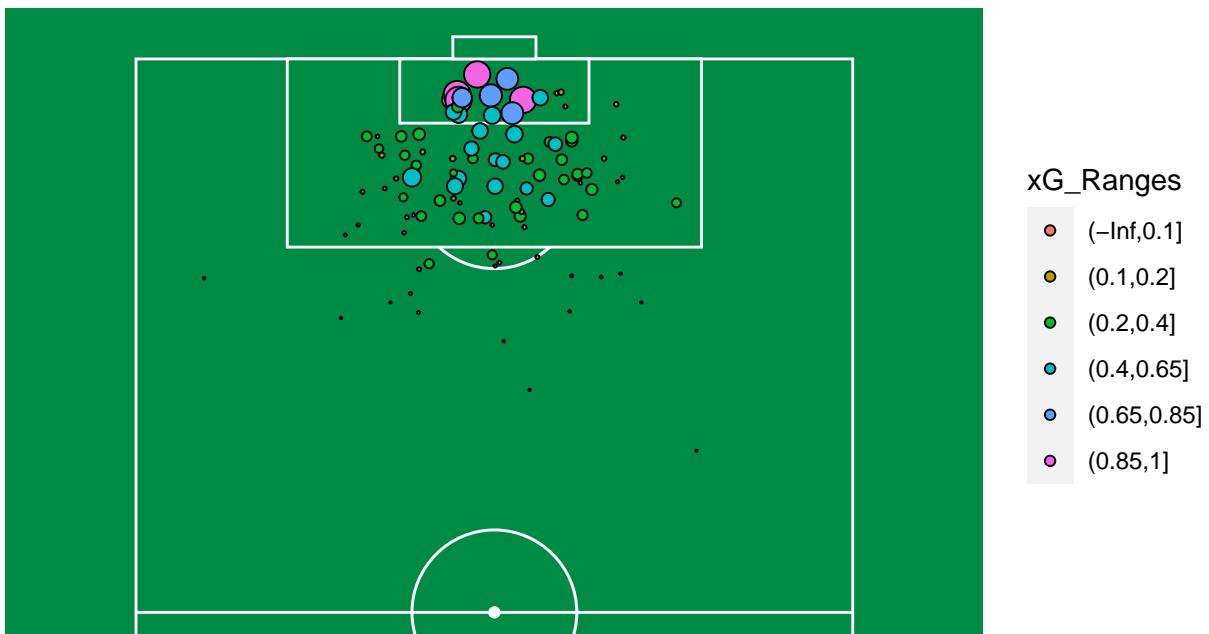
There is nothing unusual in the shotmap. The dots size is depending on the xG Value of each shot. We see that shots near to the goal and with a good degree to it, seem to have the biggest xG for each of these players. Let's plot only shots which resulted in a goal.

```
for (strName in listPlayers) {
  print(plot_shot_player(FALSE, shot_data, strName, lowTshld, upTshld))
}
```

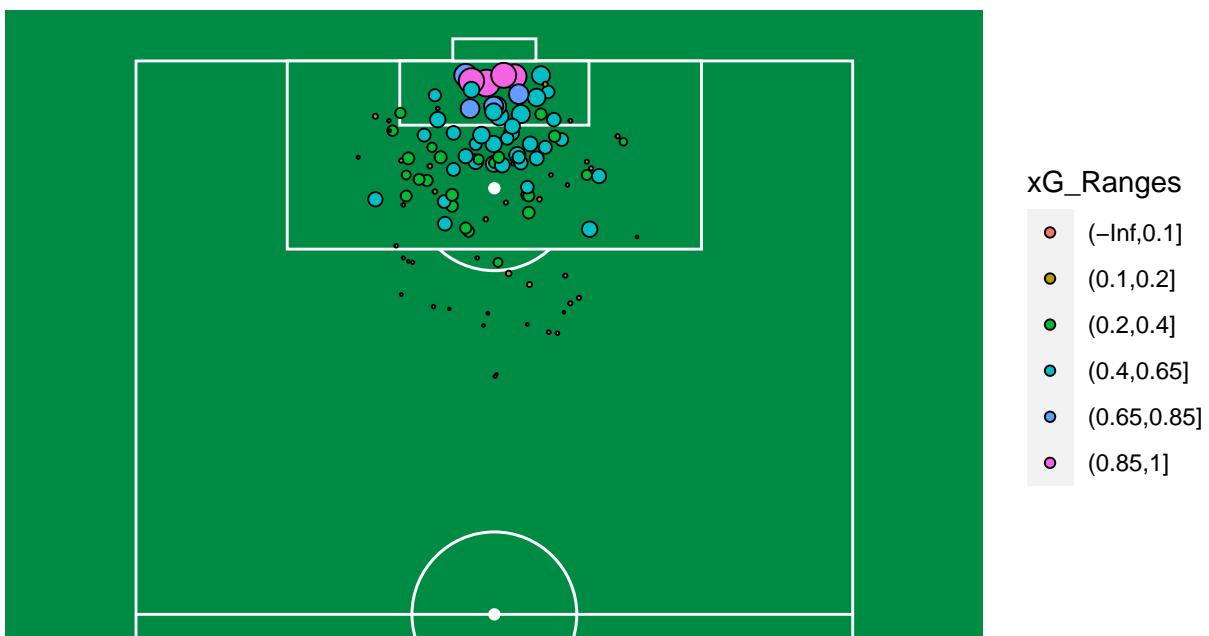
Lionel Messi goalmap where Expected Goals range between 0% – 100%



Son Heung–Min goalmap where Expected Goals range between 0% – 100%



Antoine Griezmann goalmap where Expected Goals range between 0% – 100%

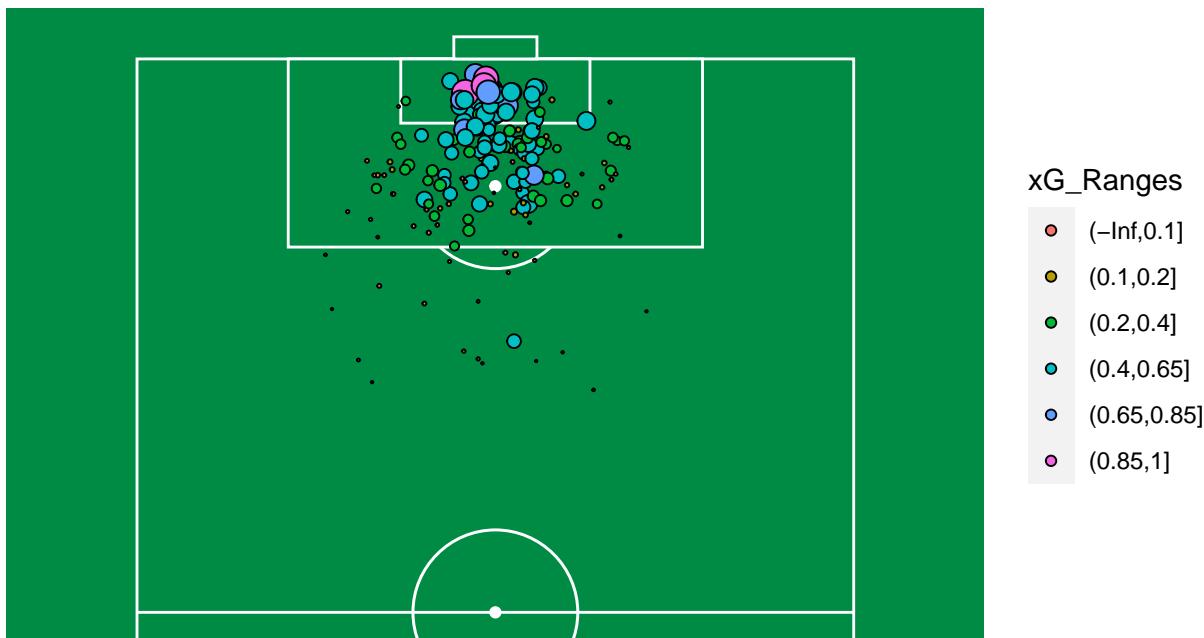


Pretty balanced for all of them. Interesting to see is, that all of them are pretty good regarding scoring goals with a xG of smaller 0.2

For your interest we also plotted Cristiano's goalmap down under

```
plot_shot_player(FALSE, shot_data, "Cristiano Ronaldo", lowTshld, upTshld)
```

## Cristiano Ronaldo goalmap where Expected Goals range between 0% – 100%



Let's have a look into all shots given by a player considering the xG Range, which means that we only consider situations where the shot had at least a 50% chance of becoming a goal. These shots can be seen as big chances.

```
lowTshld = 0.5
upTshld = 1.000
df_best <- setNames(data.frame(matrix(ncol =length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {

  df_best <- results_ranges(TRUE, shot_data, df_best, strName, lowTshld, upTshld)
}
listPlayers <- df_best$Player[1:n]
df_best
```

##	Player	Conversion_Rate	Goals_Expected	Goals	Shots
## 16	Jamie Vardy	0.755	31.21	37	49
## 25	Iago Aspas	0.784	23.75	29	37
## 15	Sadio Mané	0.679	34.61	38	56
## 11	Ciro Immobile	0.679	16.74	19	28
## 9	Kylian Mbappe-Lottin	0.667	32.74	34	51
## 22	Erling Haaland	0.727	15.09	16	22
## 10	Romelu Lukaku	0.627	36.23	37	59
## 21	Alexandre Lacazette	0.667	27.38	28	42
## 20	Son Heung-Min	0.655	19.08	19	29
## 13	Antoine Griezmann	0.625	25.20	25	40
## 23	Zlatan Ibrahimovic	0.581	19.00	18	31
## 1	Lionel Messi	0.620	50.16	49	79
## 17	Gonzalo Higuaín	0.548	24.44	23	42
## 5	Harry Kane	0.585	39.84	38	65
## 18	Mauro Icardi	0.600	34.98	33	55
## 14	Sergio Agüero	0.571	34.05	32	56
## 24	Neymar	0.589	35.28	33	56
## 4	Luis Suárez	0.591	54.47	52	88

```

## 8          Karim Benzema      0.577    44.02   41    71
## 7          Mohamed Salah     0.590    26.17   23    39
## 12         Edinson Cavani    0.571    44.31   40    70
## 6  Pierre-Emerick Aubameyang 0.584    63.63   59   101
## 19         Wissam Ben Yedder  0.520    31.85   26    50
## 2          Robert Lewandowski 0.583    80.01   74   127
## 3          Cristiano Ronaldo  0.522    56.40   48    92
## Average_xG Performance_xG Performance_Goals
## 16         0.637      0.118      5.79
## 25         0.642      0.142      5.25
## 15         0.618      0.061      3.39
## 11         0.598      0.081      2.26
## 9          0.642      0.025      1.26
## 22         0.686      0.041      0.91
## 10         0.614      0.013      0.77
## 21         0.652      0.015      0.62
## 20         0.658      -0.003     -0.08
## 13         0.630      -0.005     -0.20
## 23         0.613      -0.032     -1.00
## 1          0.635      -0.015     -1.16
## 17         0.582      -0.034     -1.44
## 5          0.613      -0.028     -1.84
## 18         0.636      -0.036     -1.98
## 14         0.608      -0.037     -2.05
## 24         0.630      -0.041     -2.28
## 4          0.619      -0.028     -2.47
## 8          0.620      -0.043     -3.02
## 7          0.671      -0.081     -3.17
## 12         0.633      -0.062     -4.31
## 6          0.630      -0.046     -4.63
## 19         0.637      -0.117     -5.85
## 2          0.630      -0.047     -6.01
## 3          0.613      -0.091     -8.40

```

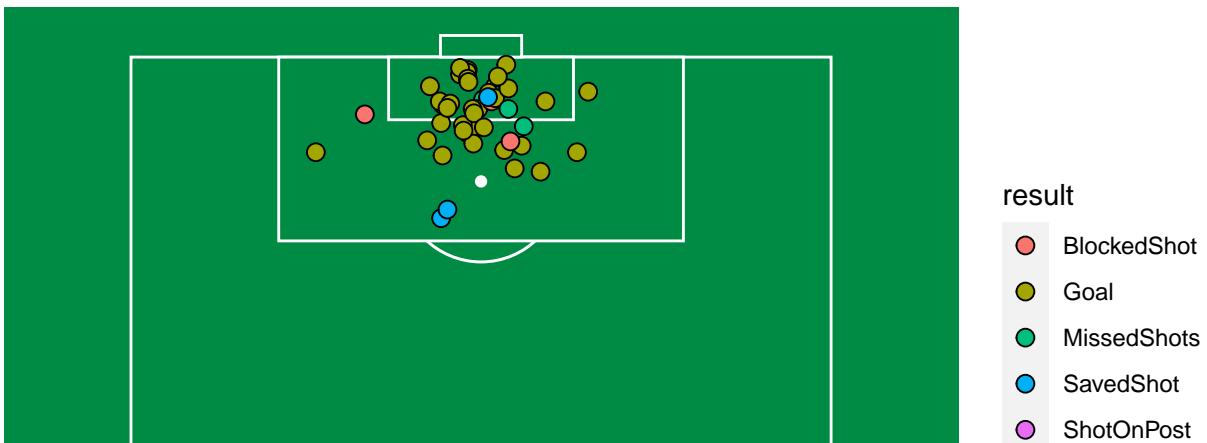
Jamie Vardy leads the list in this range with a positive value of approx. 6 goals. He is followed by Iago Aspas (approx. 5) and Sadio Mané (approx 3). It's quite interesting that most of the players (17 of 25) underperform in this range, which we classified as big chances, even though the the Conversion Rate is higher than 50% for every player, which means they all tend to score given a big chance (at least 50% xG).

```

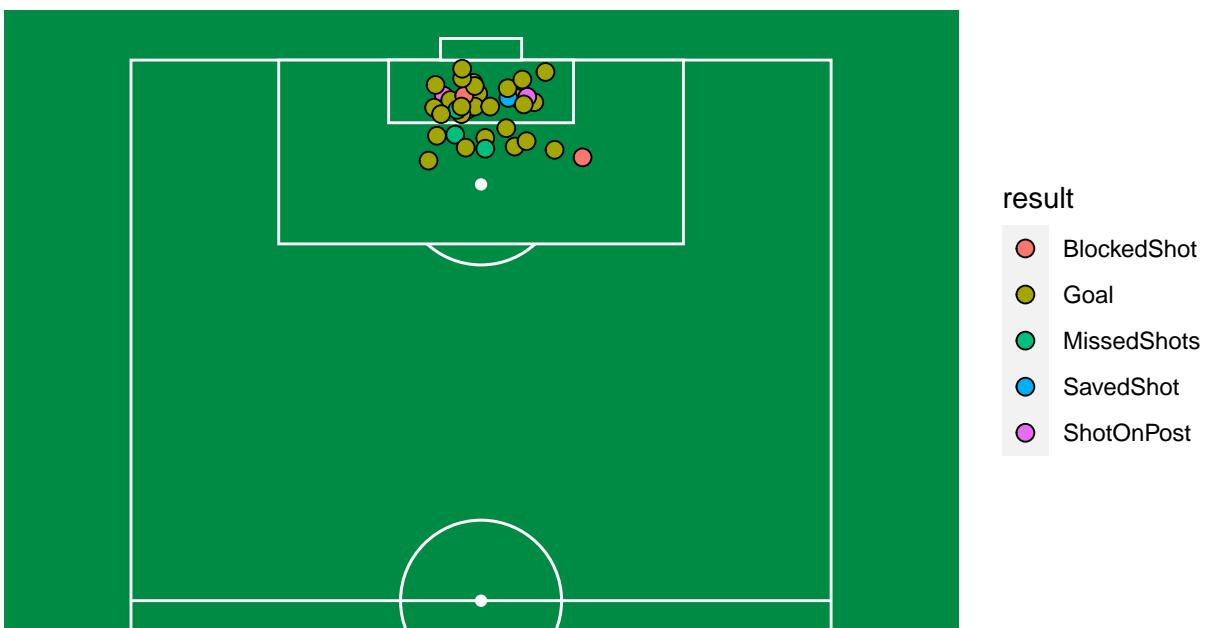
for (strName in listPlayers) {
  print(plot_shot_player(TRUE , shot_data, strName, lowTshld, upTshld))
}

```

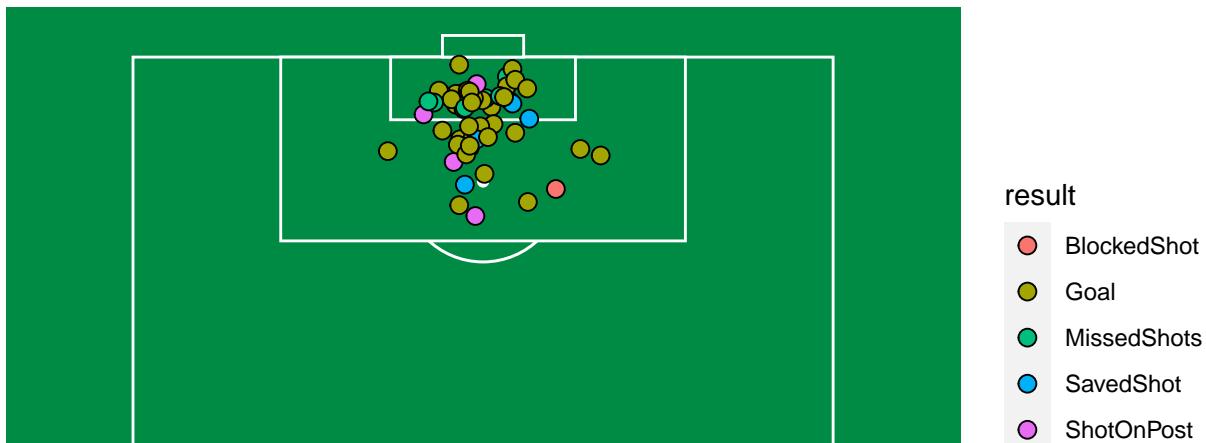
Jamie Vardy shotmap where Expected Goals range between 50% – 100%



Iago Aspas shotmap where Expected Goals range between 50% – 100%

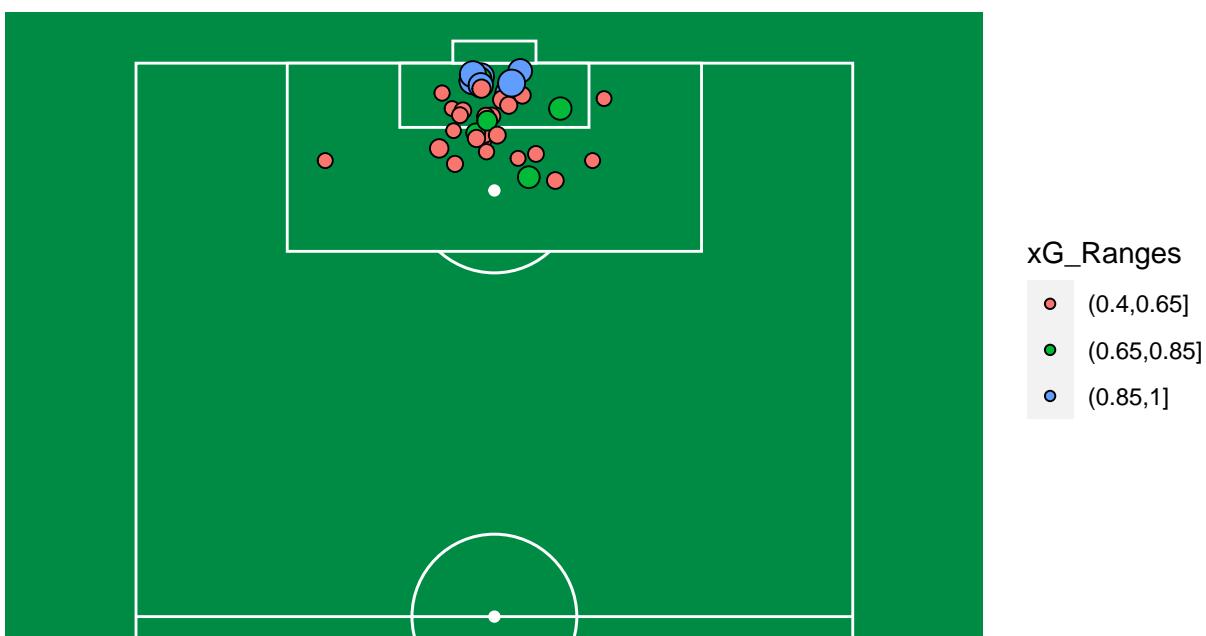


Sadio Mané shotmap where Expected Goals range between 50% – 100%

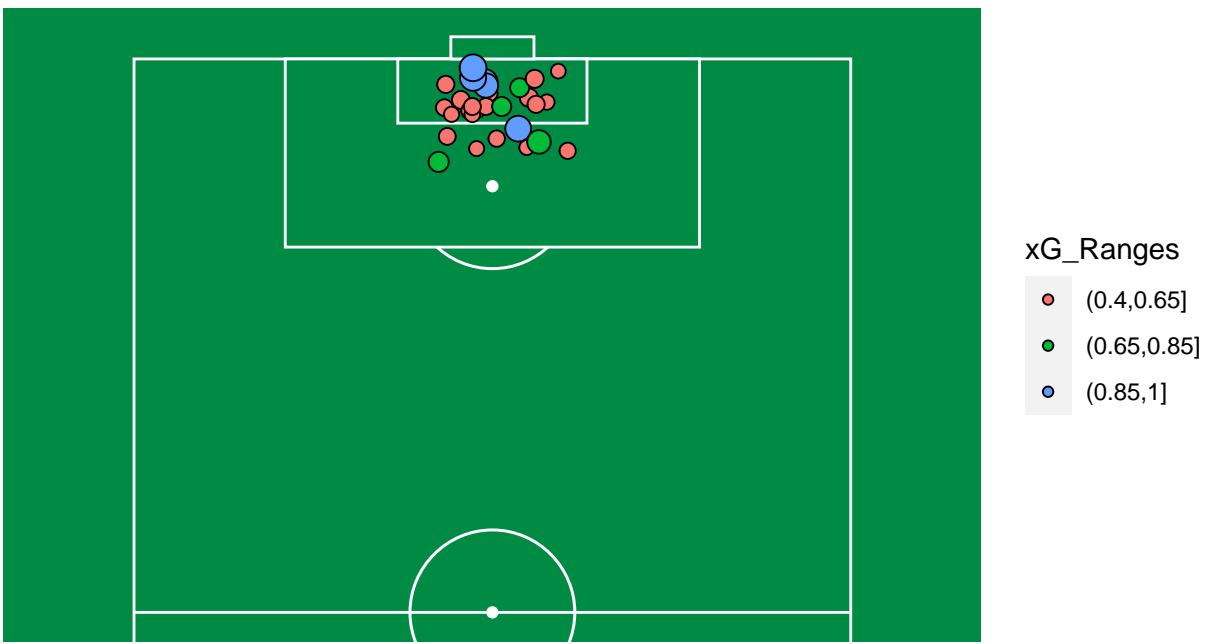


```
for (strName in listPlayers) {  
  print(plot_shot_player(FALSE , shot_data, strName, lowTshld, upTshld))  
}
```

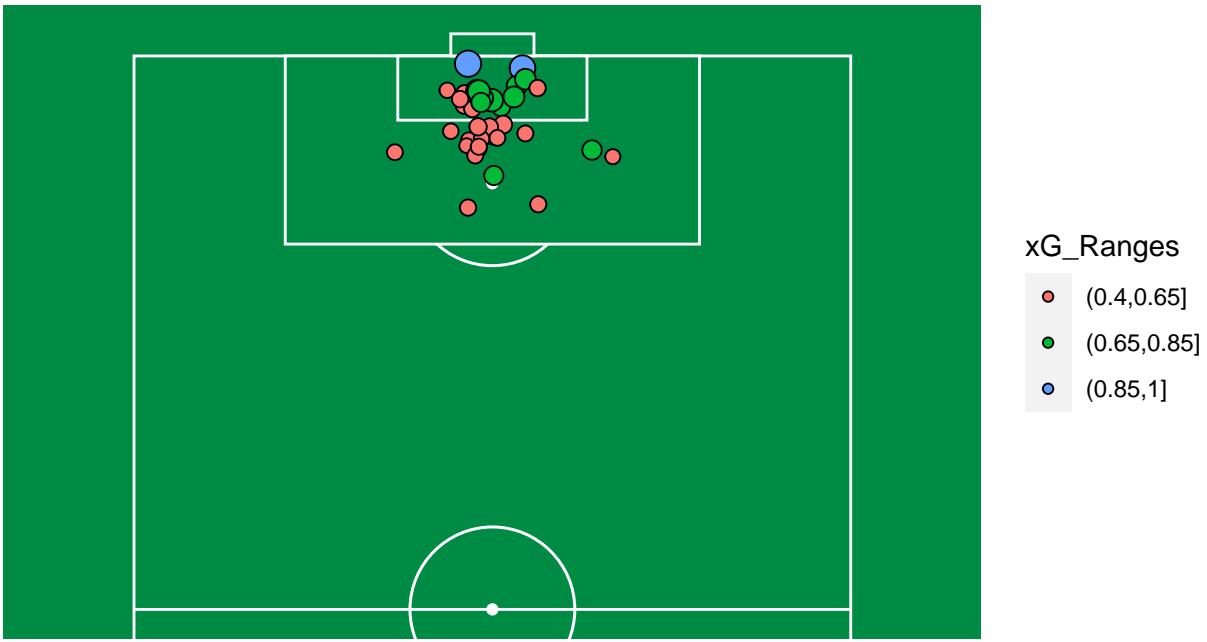
Jamie Vardy goalmap where Expected Goals range between 50% – 100%



Iago Aspas goalmap where Expected Goals range between 50% – 100%



Sadio Mané goalmap where Expected Goals range between 50% – 100%



see that Sadio Mane really made the difficult goals compared to the other 2

Let's have a look into all shots given within a xG range of 25% to 50%. These shots can be seen as middle chances.

```
lowTshld = 0.25
upTshld = 0.5
df_best <- setNames(data.frame(matrix(ncol =length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {

  df_best <- results_ranges(TRUE, shot_data, df_best, strName, lowTshld, upTshld)
}
```

We

```
listPlayers <- df_best$Player[1:n]
df_best
```

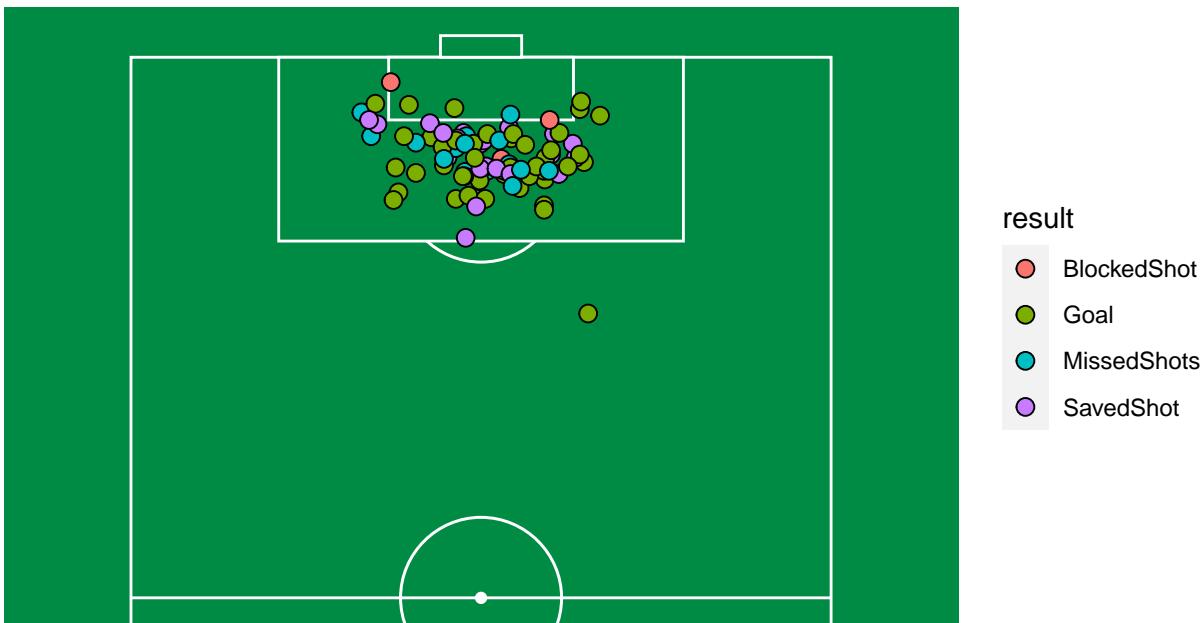
	Player	Conversion_Rate	Goals_Expected	Goals	Shots
## 17	Gonzalo Higuain	0.500	31.08	42	84
## 19	Wissam Ben Yedder	0.462	38.58	48	104
## 20	Son Heung-Min	0.468	27.26	36	77
## 22	Erling Haaland	0.585	15.33	24	41
## 5	Harry Kane	0.450	49.66	58	129
## 3	Cristiano Ronaldo	0.422	53.80	62	147
## 13	Antoine Griezmann	0.462	35.53	43	93
## 9	Kylian Mbappe-Lottin	0.418	44.77	51	122
## 15	Sadio Mané	0.413	33.67	38	92
## 21	Alexandre Lacazette	0.408	35.67	40	98
## 10	Romelu Lukaku	0.424	38.02	42	99
## 7	Mohamed Salah	0.394	52.26	56	142
## 25	Iago Aspas	0.417	26.28	30	72
## 14	Sergio Agüero	0.410	30.96	34	83
## 23	Zlatan Ibrahimovic	0.400	29.44	32	80
## 6	Pierre-Emerick Aubameyang	0.397	47.50	50	126
## 12	Edinson Cavani	0.369	53.94	55	149
## 1	Lionel Messi	0.388	44.08	45	116
## 8	Karim Benzema	0.391	42.35	43	110
## 16	Jamie Vardy	0.359	37.49	37	103
## 18	Mauro Icardi	0.360	38.10	36	100
## 11	Ciro Immobile	0.339	42.24	40	118
## 4	Luis Suárez	0.366	66.67	64	175
## 24	Neymar	0.353	39.17	36	102
## 2	Robert Lewandowski	0.354	73.90	69	195
##	Average_xG	Performance_xG	Performance_Goals		
## 17	0.370	0.130	10.92		
## 19	0.371	0.091	9.42		
## 20	0.354	0.114	8.74		
## 22	0.374	0.211	8.67		
## 5	0.385	0.065	8.34		
## 3	0.366	0.056	8.20		
## 13	0.382	0.080	7.47		
## 9	0.367	0.051	6.23		
## 15	0.366	0.047	4.33		
## 21	0.364	0.044	4.33		
## 10	0.384	0.040	3.98		
## 7	0.368	0.026	3.74		
## 25	0.365	0.052	3.72		
## 14	0.373	0.037	3.04		
## 23	0.368	0.032	2.56		
## 6	0.377	0.020	2.50		
## 12	0.362	0.007	1.06		
## 1	0.380	0.008	0.92		
## 8	0.385	0.006	0.65		
## 16	0.364	-0.005	-0.49		
## 18	0.381	-0.021	-2.10		
## 11	0.358	-0.019	-2.24		
## 4	0.381	-0.015	-2.67		
## 24	0.384	-0.031	-3.17		

```
## 2      0.379      -0.025     -4.90
```

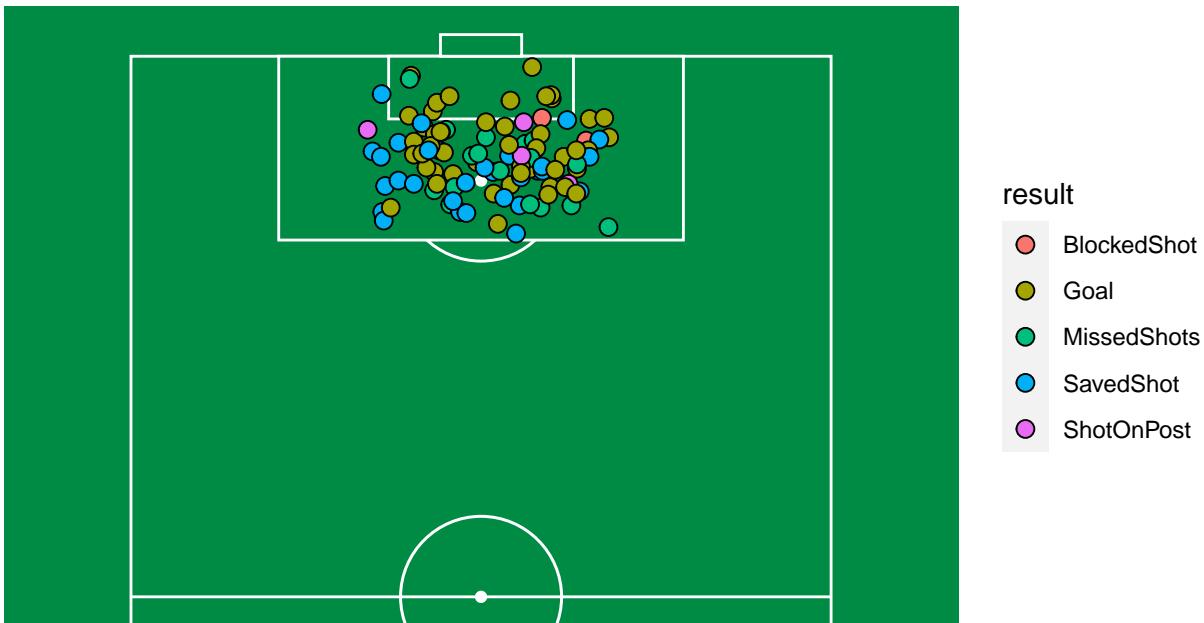
In this category Higuain leads the list with a Performance Goals value of almost 11. Haaland again convinces us with a Conversion Rate of almost 60%.

```
for (strName in listPlayers) {  
  print(plot_shot_player(TRUE , shot_data, strName, lowTshld, upTshld))  
}
```

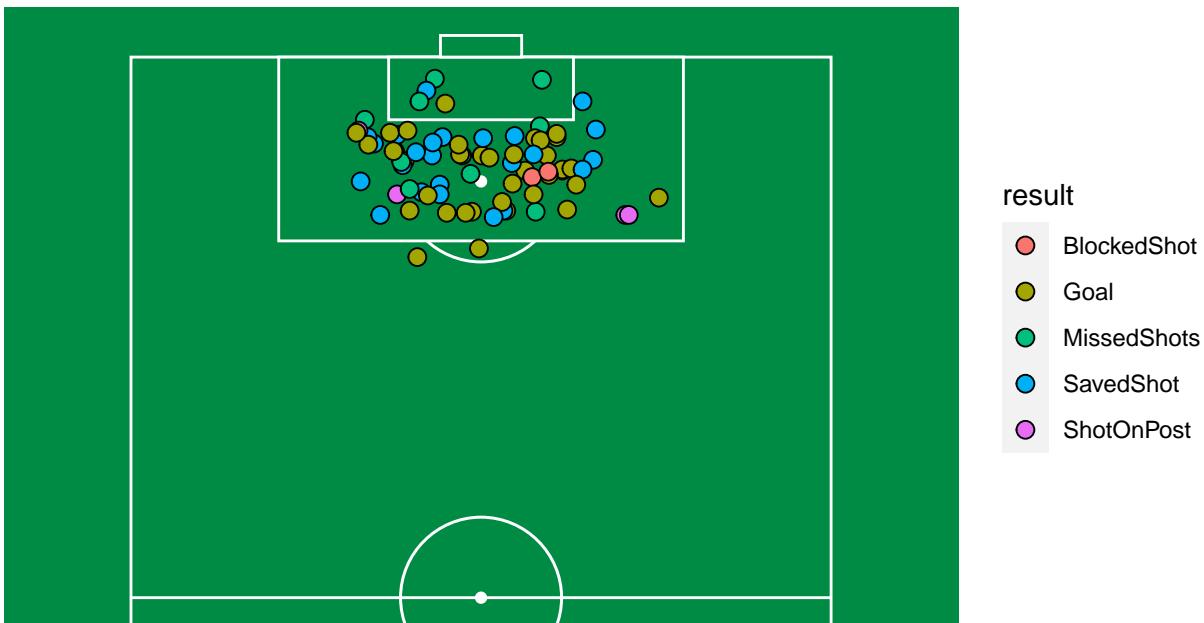
Gonzalo Higuaín shotmap where Expected Goals range between 25% – 50%



Wissam Ben Yedder shotmap where Expected Goals range between 25% – 50%



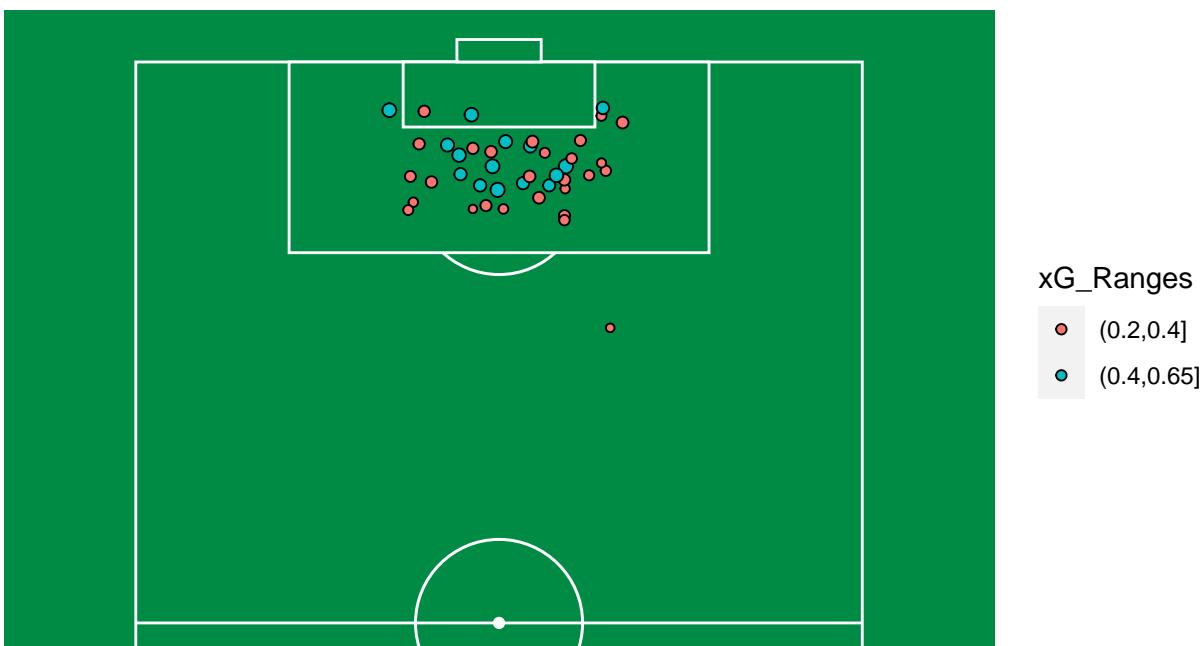
Son Heung–Min shotmap where Expected Goals range between 25% – 50%



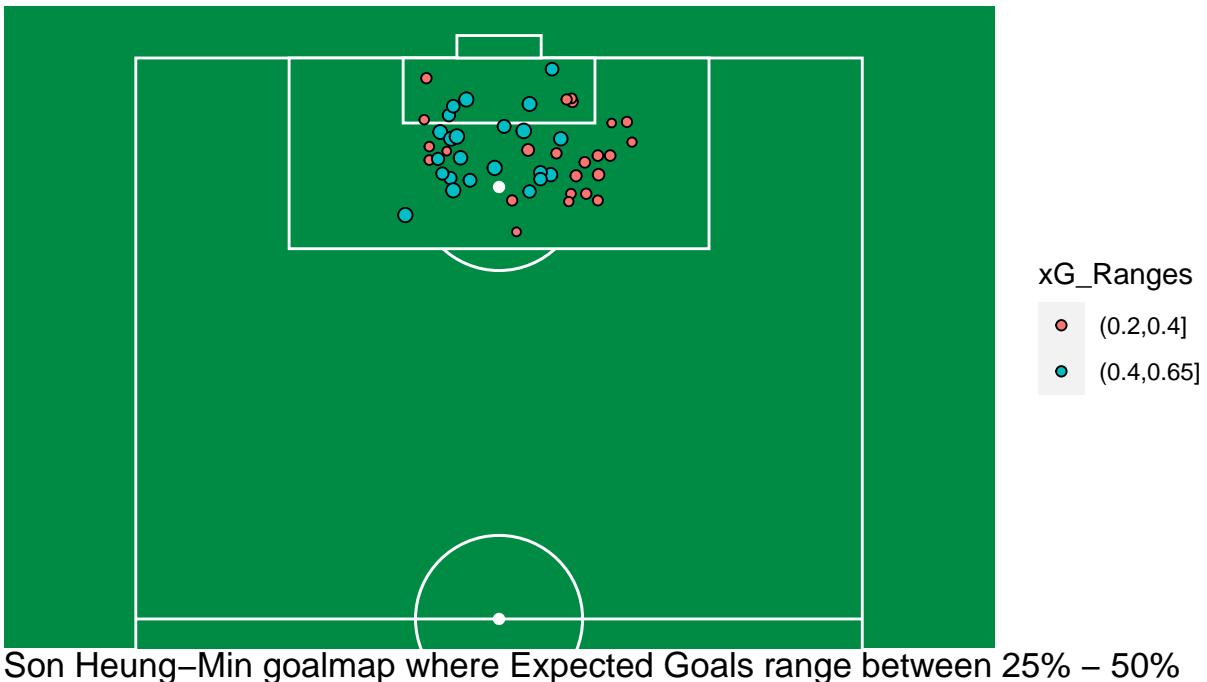
Interesting to see for the top 3 of our list is, that all of them are almost able to score every 2th shot for middle chances.

```
for (strName in listPlayers) {
  print(plot_shot_player(FALSE , shot_data, strName, lowTshld, upTshld))
}
```

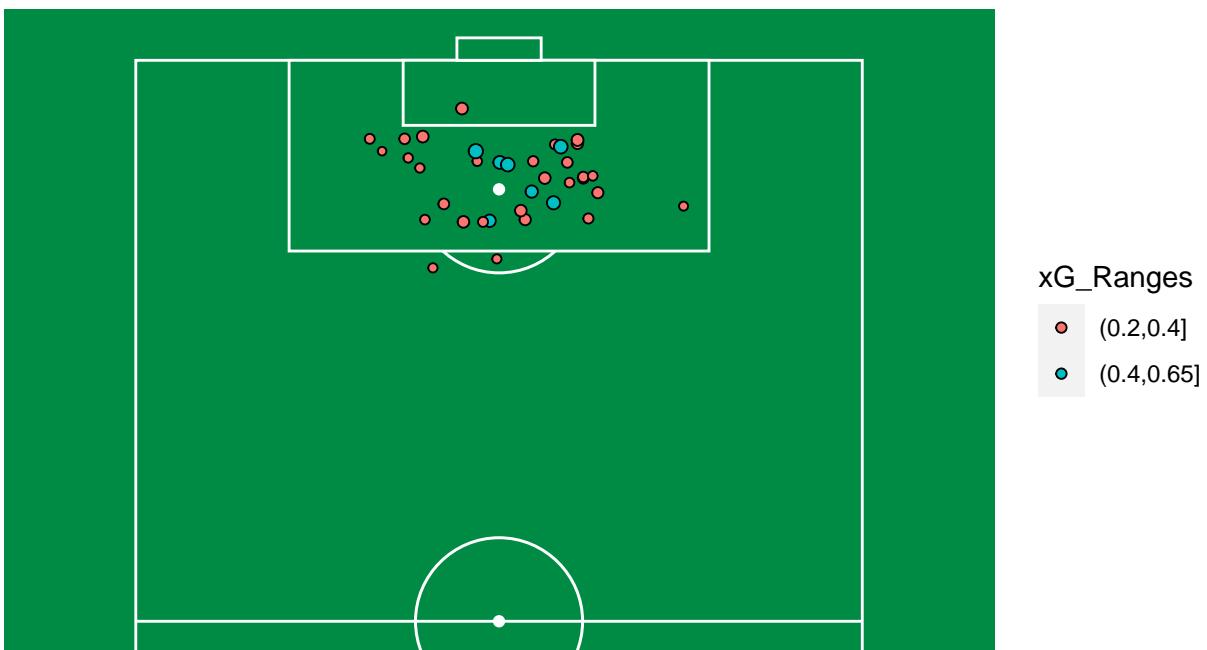
Gonzalo Higuaín goalmap where Expected Goals range between 25% – 50%



Wissam Ben Yedder goalmap where Expected Goals range between 25% – 50%



Son Heung-Min goalmap where Expected Goals range between 25% – 50%



Let's have a look into all shots given by a player considering the xG Range, which means that we only consider situations where the shot had at least a 10% and a maximum of 25% chance of becoming a goal. These shots can be seen as small chances.

```
lowTshld = 0.0
upTshld = 0.25
df_best <- setNames(data.frame(matrix(ncol = length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {

  df_best <- results_ranges(TRUE, shot_data, df_best, strName, lowTshld, upTshld)
```

```

}

listPlayers <- df_best$Player[1:n]
df_best

##          Player Conversion_Rate Goals_Expected Goals Shots
## 1      Lionel Messi        0.105     79.13    115 1099
## 20     Son Heung-Min       0.105     27.08     48 459
## 4      Luis Suárez        0.095     46.58     62 656
## 13     Antoine Griezmann   0.093     29.26     44 472
## 11     Ciro Immobile       0.089     42.44     57 643
## 5      Harry Kane          0.070     45.37     56 796
## 7      Mohamed Salah       0.085     45.47     56 659
## 9      Kylian Mbappe-Lottin 0.096     30.71     40 415
## 23     Zlatan Ibrahimovic  0.088     27.72     37 420
## 18     Mauro Icardi         0.094     27.65     36 384
## 25     Iago Aspas           0.086     28.21     36 421
## 21     Alexandre Lacazette 0.089     28.83     35 395
## 2      Robert Lewandowski  0.081     55.73     60 743
## 19     Wissam Ben Yedder    0.080     26.18     30 374
## 14     Sergio Agüero        0.081     41.55     45 554
## 8      Karim Benzema         0.078     41.97     45 575
## 22     Erling Haaland        0.106     11.48     14 132
## 17     Gonzalo Higuaín       0.072     38.56     41 567
## 16     Jamie Vardy           0.081     31.01     33 408
## 3      Cristiano Ronaldo    0.063     70.21     72 1151
## 6      Pierre-Emerick Aubameyang 0.072     35.06     35 487
## 10     Romelu Lukaku          0.074     40.27     40 537
## 15     Sadio Mané             0.073     36.00     35 480
## 24     Neymar                  0.063     33.58     30 473
## 12     Edinson Cavani          0.062     24.86     21 336
##          Average_xG Performance_xG Performance_Goals
## 1      0.072        0.033            35.87
## 20     0.059        0.046            20.92
## 4      0.071        0.024            15.42
## 13     0.062        0.031            14.74
## 11     0.066        0.023            14.56
## 5      0.057        0.013            10.63
## 7      0.069        0.016            10.53
## 9      0.074        0.022            9.29
## 23     0.066        0.022            9.28
## 18     0.072        0.022            8.35
## 25     0.067        0.019            7.79
## 21     0.073        0.016            6.17
## 2      0.075        0.006            4.27
## 19     0.070        0.010            3.82
## 14     0.075        0.006            3.45
## 8      0.073        0.005            3.03
## 22     0.087        0.019            2.52
## 17     0.068        0.004            2.44
## 16     0.076        0.005            1.99
## 3      0.061        0.002            1.79
## 6      0.072        0.000            -0.06
## 10     0.075       -0.001            -0.27
## 15     0.075       -0.002            -1.00

```

```

## 24      0.071      -0.008     -3.58
## 12      0.074      -0.012     -3.86

```

Messi really loves small chances. Interesting as he didn't perform so well for big chances. Son again is very visible far above in the result list. Suarez occupies the 3rd place. A name which shouldn't surprise you.

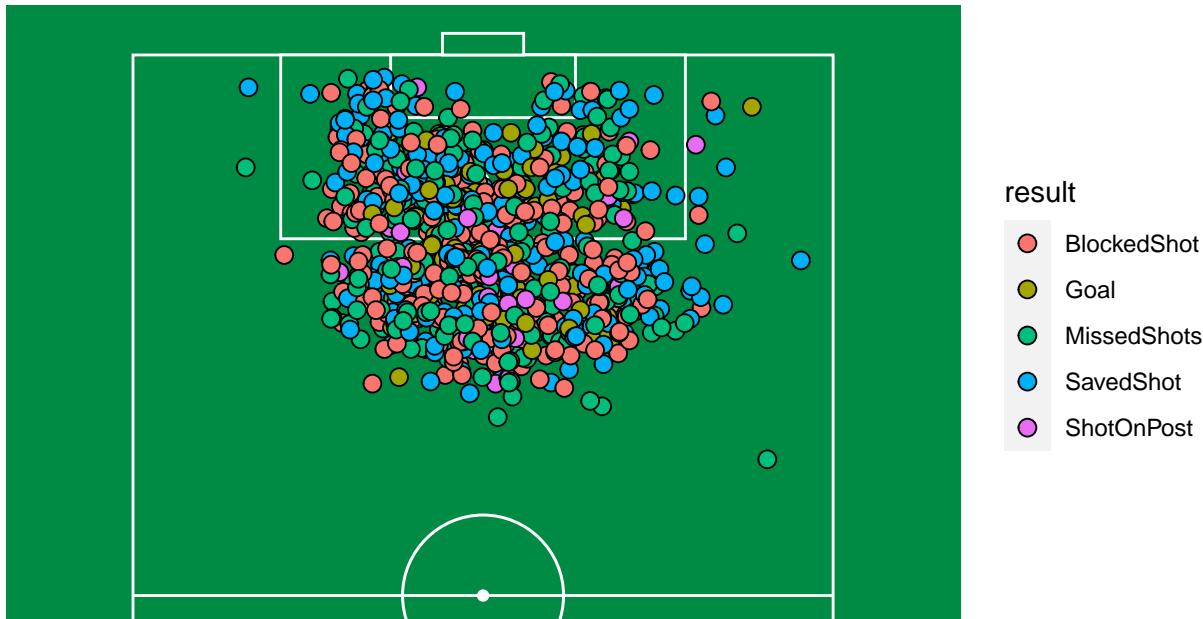
Here you can see the shot- and goalmap for the top 3

```

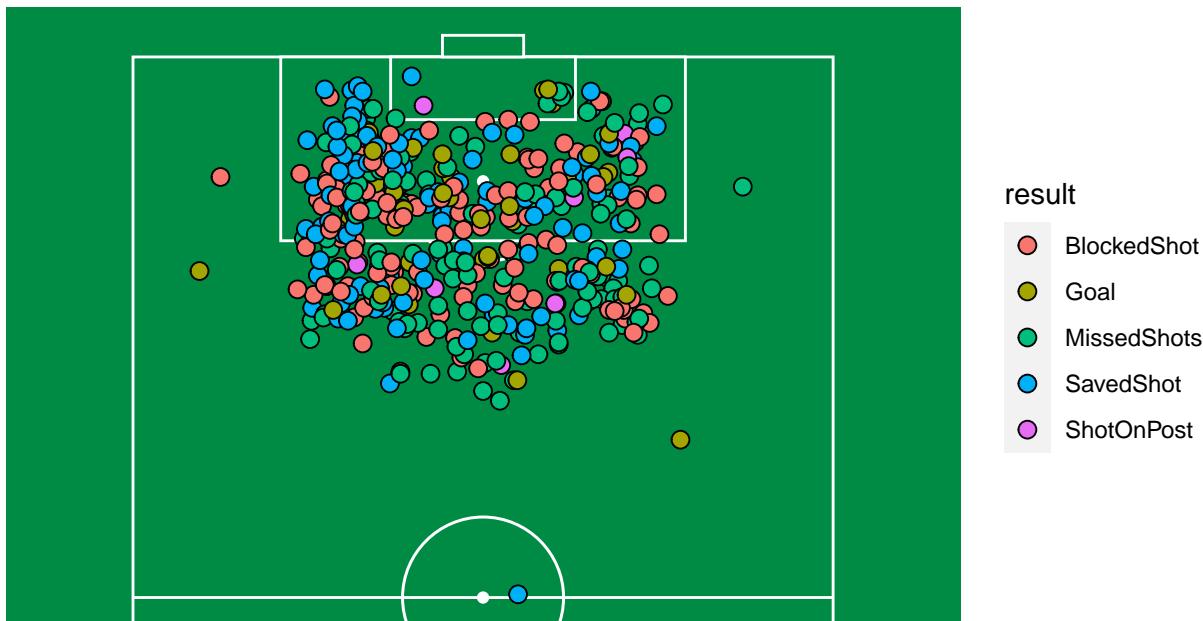
for (strName in listPlayers) {
  print(plot_shot_player(TRUE , shot_data, strName, lowTshld, upTshld))
}

```

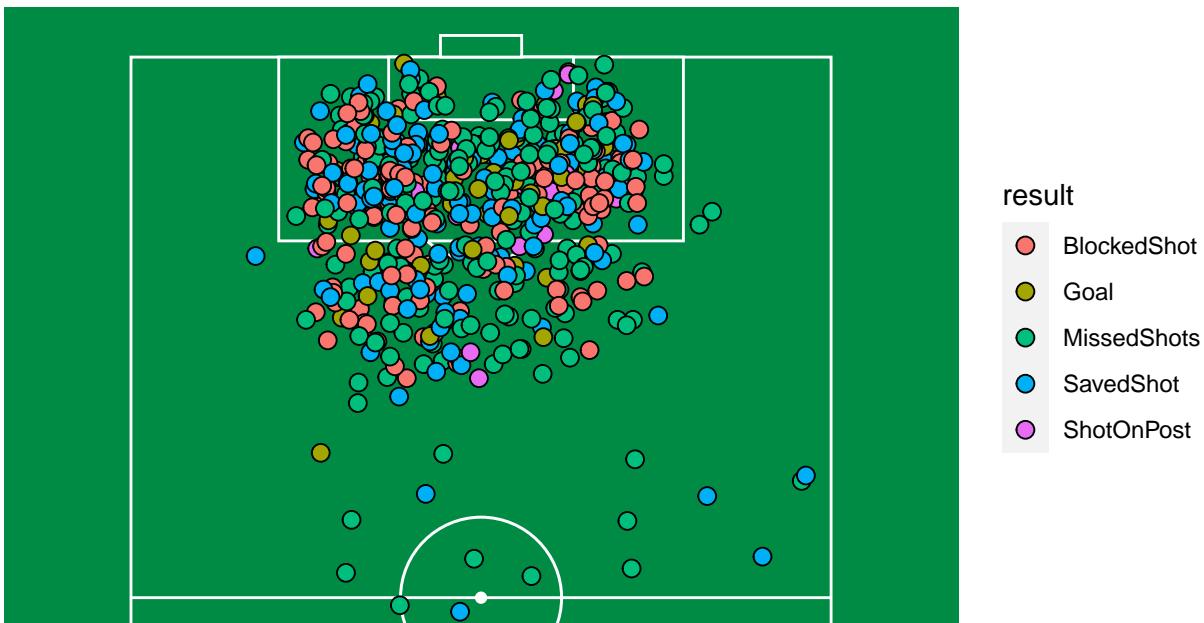
Lionel Messi shotmap where Expected Goals range between 0% – 25%



Son Heung–Min shotmap where Expected Goals range between 0% – 25%

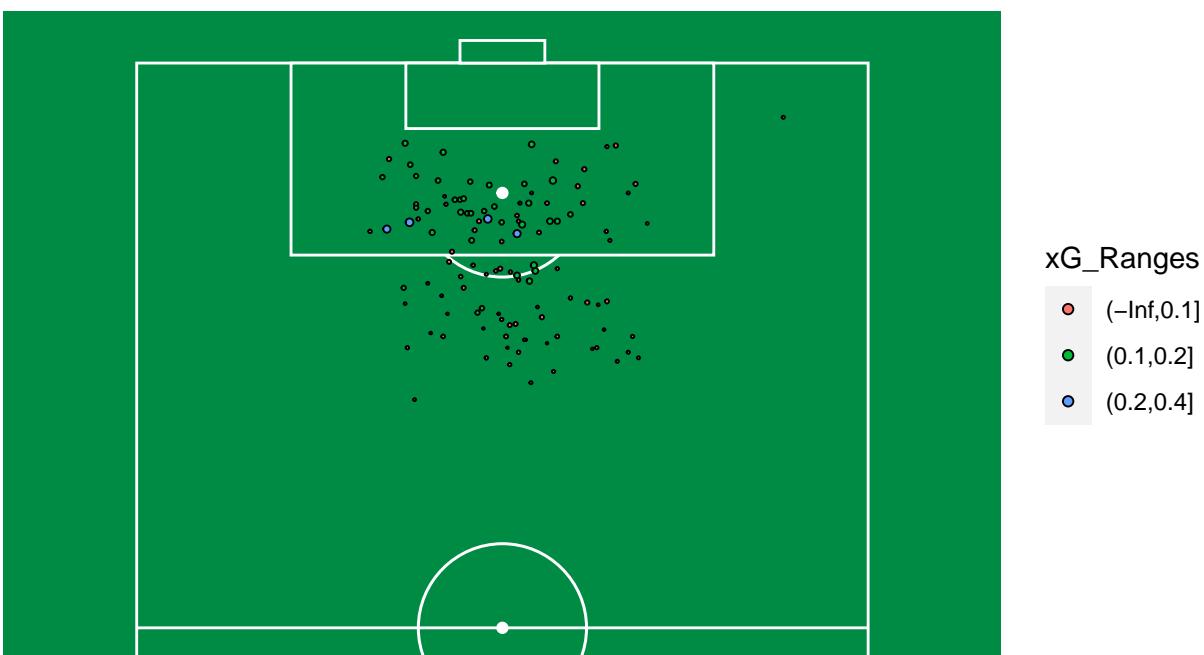


Luis Suárez shotmap where Expected Goals range between 0% – 25%

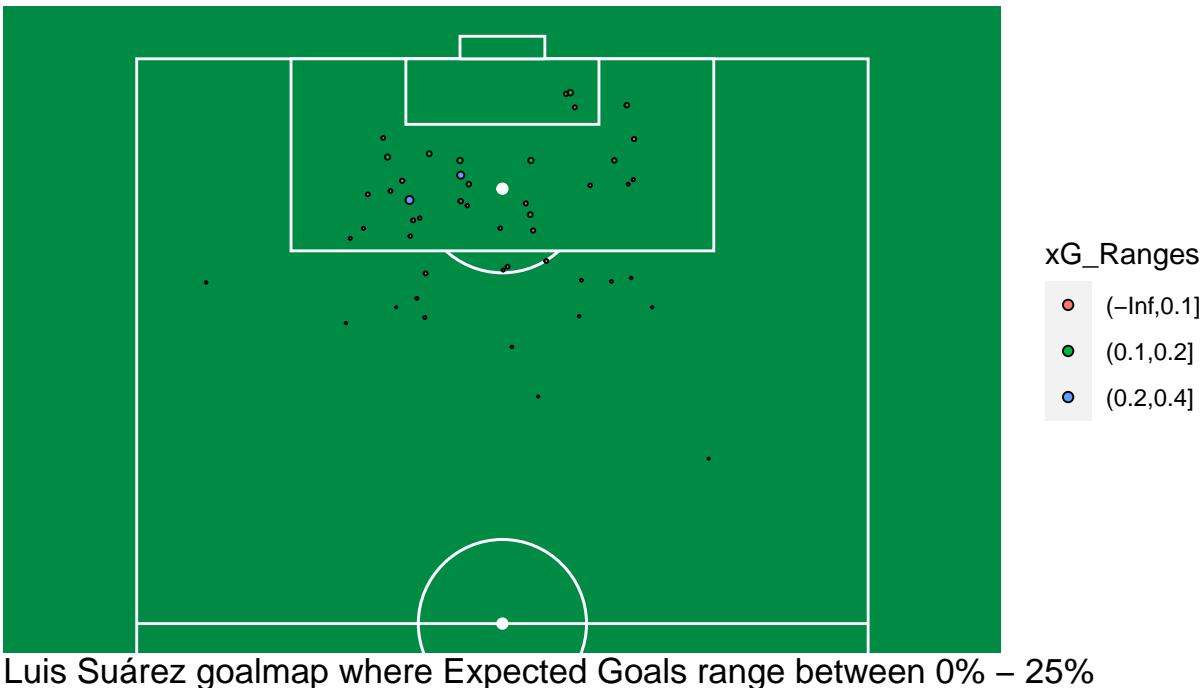


```
for (strName in listPlayers) {  
  print(plot_shot_player(FALSE , shot_data, strName, lowTshld, upTshld))  
}
```

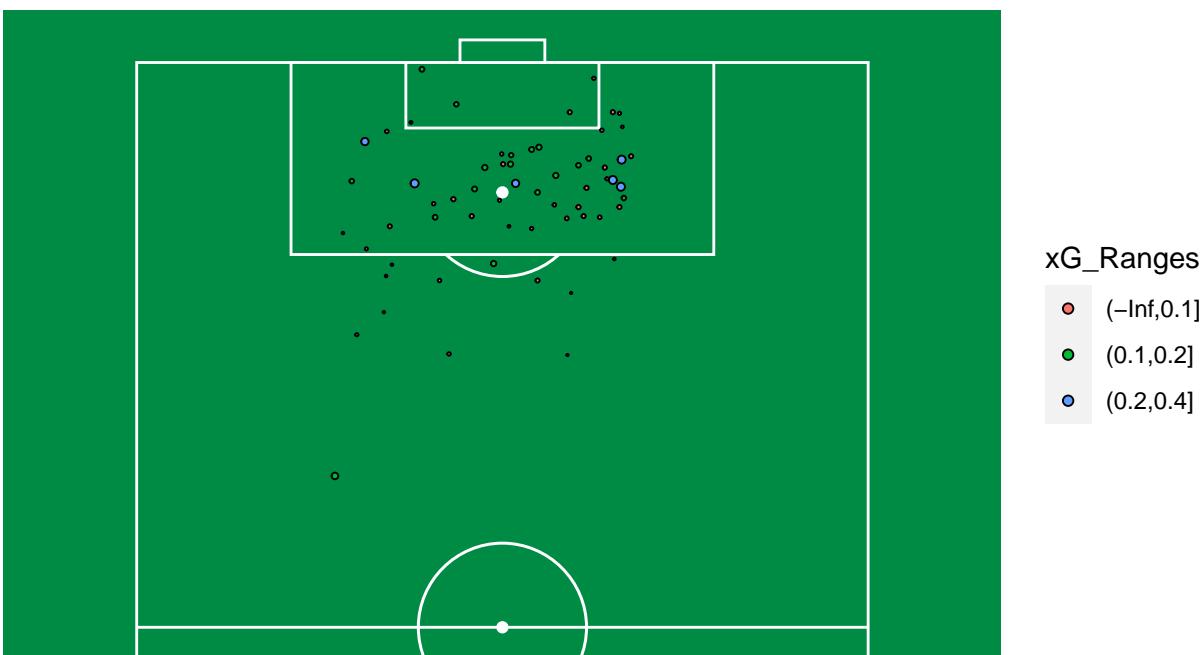
Lionel Messi goalmap where Expected Goals range between 0% – 25%



Son Heung–Min goalmap where Expected Goals range between 0% – 25%



Luis Suárez goalmap where Expected Goals range between 0% – 25%



Let's have a look who had the best values in the last season (2021/2022)

```
curr_shot_data <- shot_data[(shot_data$year == "2021" ),]
lowTshld = 0.0
upTshld = 1.00
df_best <- setNames(data.frame(matrix(ncol =length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {

  df_best <- results_ranges(TRUE, curr_shot_data, df_best, strName, lowTshld, upTshld)
}
```

```
listPlayers <- df_best$Player[1:n]
df_best
```

	Player	Conversion_Rate	Goals_Expected	Goals	Shots
## 20	Son Heung-Min	0.267	17.03	23	86
## 16	Jamie Vardy	0.273	10.01	15	55
## 11	Ciro Immobile	0.183	15.48	20	109
## 19	Wissam Ben Yedder	0.266	12.93	17	64
## 8	Karim Benzema	0.171	16.50	20	117
## 22	Erling Haaland	0.216	12.51	16	74
## 9	Kylian Mbappe-Lottin	0.175	21.88	25	143
## 25	Iago Aspas	0.165	10.88	14	85
## 23	Zlatan Ibrahimovic	0.157	6.58	8	51
## 24	Neymar	0.200	8.70	10	50
## 10	Romelu Lukaku	0.167	6.26	7	42
## 4	Luis Suárez	0.125	17.00	17	136
## 3	Cristiano Ronaldo	0.139	15.12	15	108
## 14	Sergio Agüero	0.167	1.16	1	6
## 6	Pierre-Emerick Aubameyang	0.206	14.21	14	68
## 15	Sadio Mané	0.163	16.86	16	98
## 12	Edinson Cavani	0.111	3.73	2	18
## 7	Mohamed Salah	0.135	19.82	18	133
## 13	Antoine Griezmann	0.064	6.06	3	47
## 18	Mauro Icardi	0.143	7.20	4	28
## 21	Alexandre Lacazette	0.047	5.33	2	43
## 1	Lionel Messi	0.067	9.88	6	89
## 2	Robert Lewandowski	0.194	34.10	30	155
## 5	Harry Kane	0.101	17.67	13	129
## 17	Gonzalo Higuaín	NaN	NaN	0	0
##	Average_xG	Performance_xG	Performance_Goals		
## 20	0.198	0.069	5.97		
## 16	0.182	0.091	4.99		
## 11	0.142	0.041	4.52		
## 19	0.202	0.064	4.07		
## 8	0.141	0.030	3.50		
## 22	0.169	0.047	3.49		
## 9	0.153	0.022	3.12		
## 25	0.128	0.037	3.12		
## 23	0.129	0.028	1.42		
## 24	0.174	0.026	1.30		
## 10	0.149	0.018	0.74		
## 4	0.125	0.000	0.00		
## 3	0.140	-0.001	-0.12		
## 14	0.193	-0.026	-0.16		
## 6	0.209	-0.003	-0.21		
## 15	0.172	-0.009	-0.86		
## 12	0.207	-0.096	-1.73		
## 7	0.149	-0.014	-1.82		
## 13	0.129	-0.065	-3.06		
## 18	0.257	-0.114	-3.20		
## 21	0.124	-0.077	-3.33		
## 1	0.111	-0.044	-3.88		
## 2	0.220	-0.026	-4.10		
## 5	0.137	-0.036	-4.67		

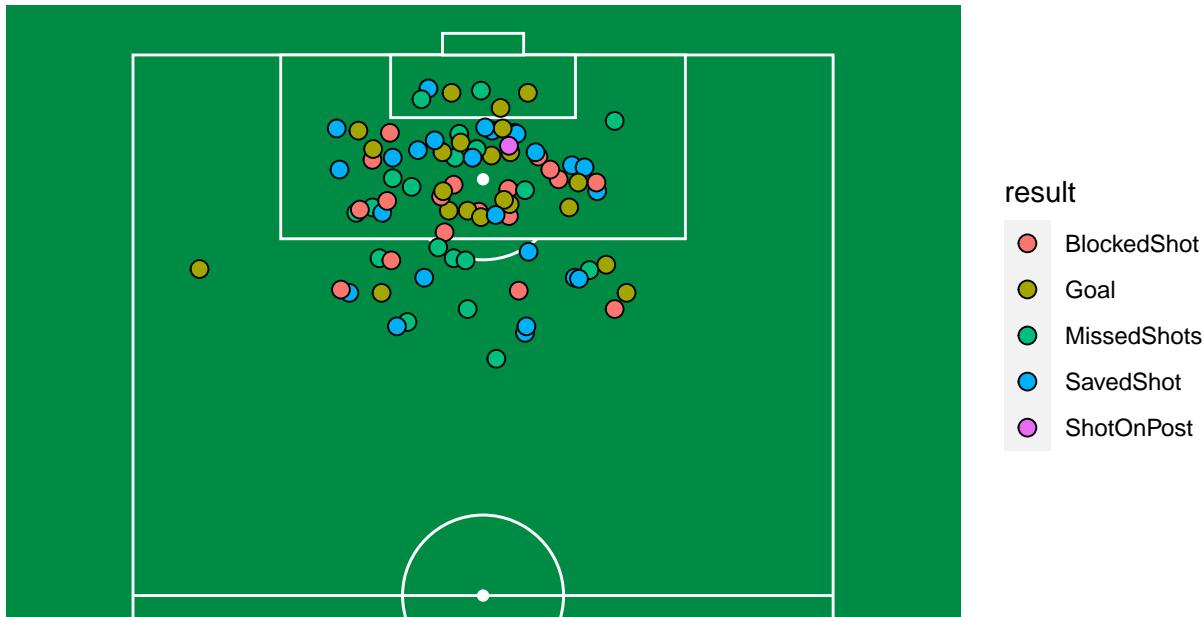
```
## 17      NaN      NaN      NaN
```

The best performer of the last season was Son followed by Vardy and Immobile. Jamie Vardy has the best “Performance\_xG” of 9.1%. Higauin has no values as he didn’t play in the top 5 league Europe last season. From a statistical point of view Son’s recent hype seems to be more than deserved.

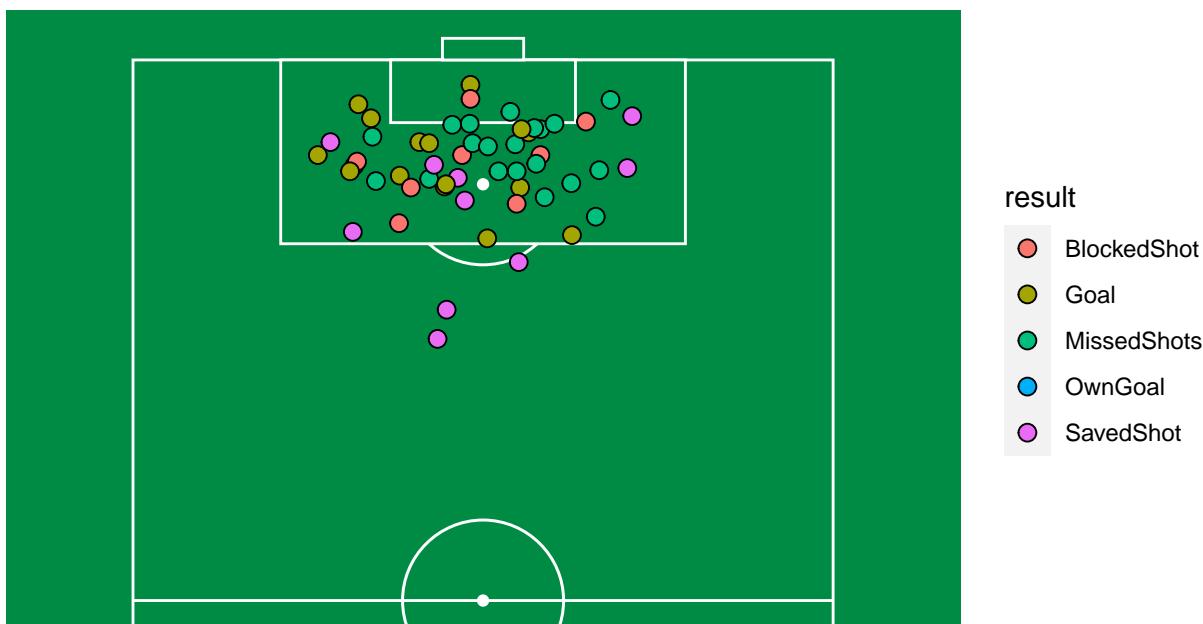
Here you can see the shot- and goalmaps.

```
for (strName in listPlayers) {  
  print(plot_shot_player(TRUE , curr_shot_data, strName, lowTshld, upTshld))  
}
```

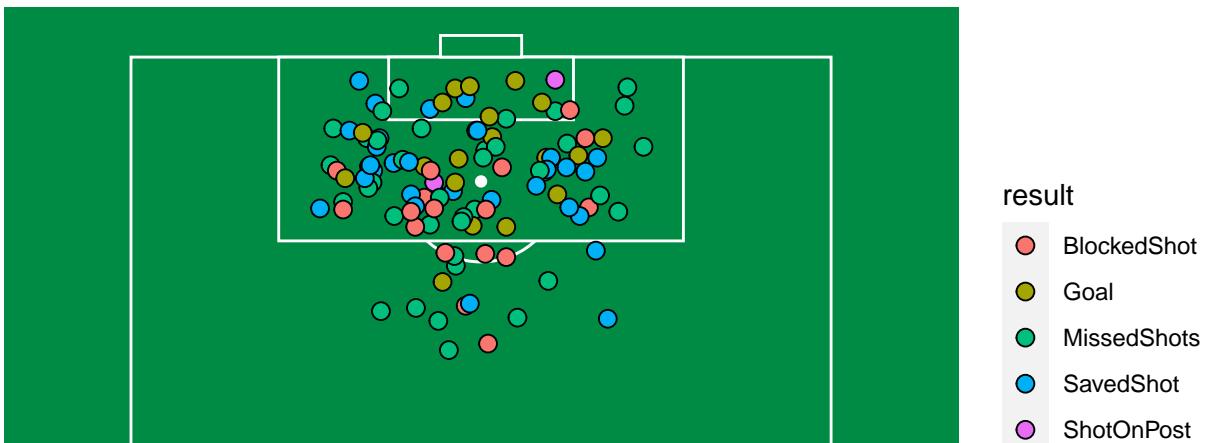
Son Heung–Min shotmap where Expected Goals range between 0% – 100%



Jamie Vardy shotmap where Expected Goals range between 0% – 100%

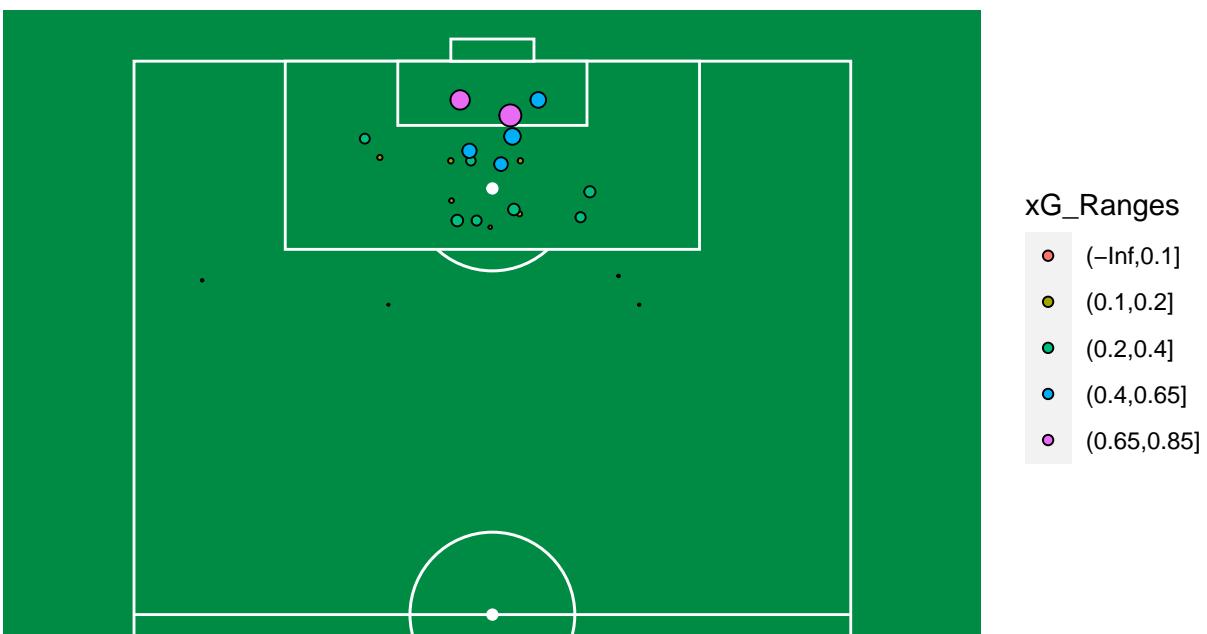


Ciro Immobile shotmap where Expected Goals range between 0% – 100%

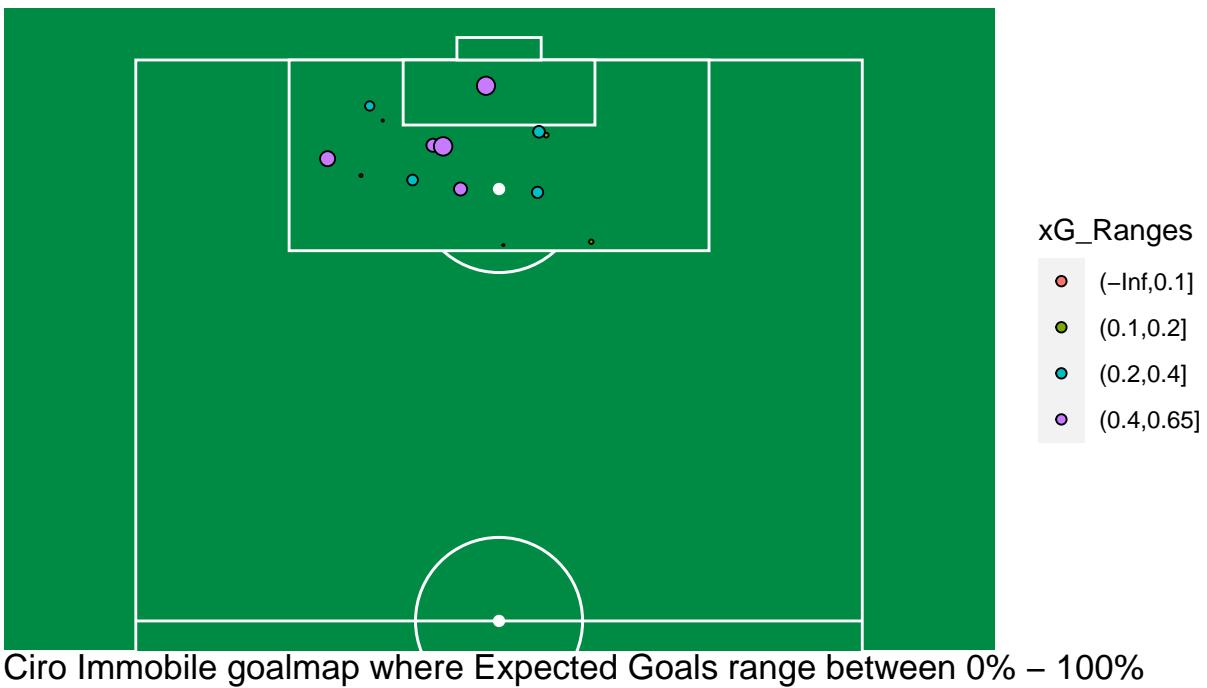


```
for (strName in listPlayers) {  
  print(plot_shot_player(FALSE , curr_shot_data, strName, lowTshld, upTshld))  
}
```

Son Heung–Min goalmap where Expected Goals range between 0% – 100%



Jamie Vardy goalmap where Expected Goals range between 0% – 100%



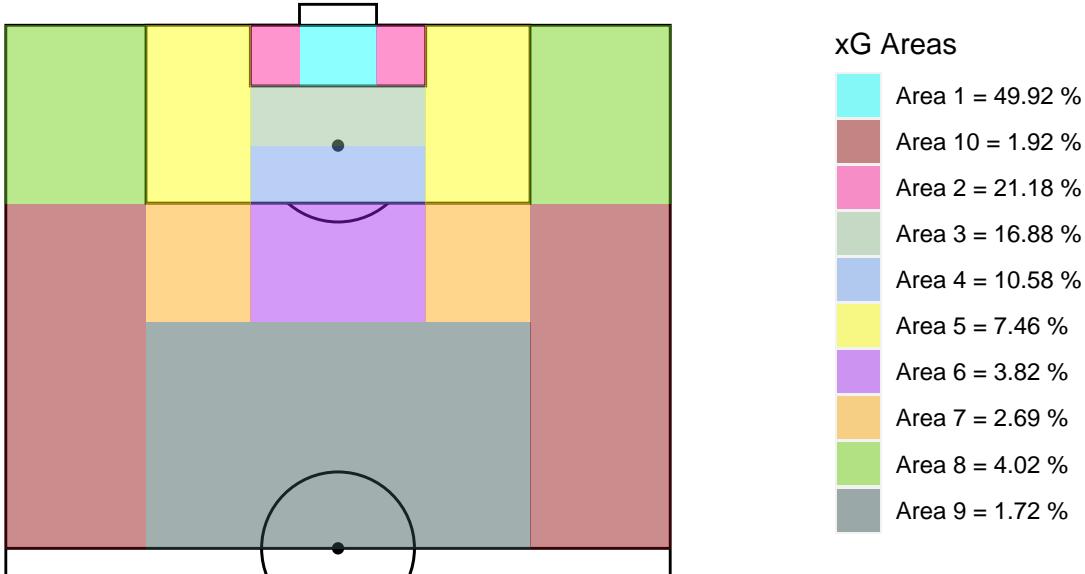
Ciro Immobile goalmap where Expected Goals range between 0% – 100%



#Now we will highlight the different areas which we defined before

```
pltTitle <- "Expected Goals (xG) divided into different areas with it's probabilities"  
plot_areas_xG(xG_plot_areas, xGChar, pltTitle)
```

## Expected Goals (xG) divided into different areas with it's probabilities



```

outbox = 'Outside Box'
insideBox = 'Inside Box'
inside5 = 'Inside 5'
areas_spec<- shot_data %>%
  mutate(xGAreas = case_when(X < 0.8294 & Y < 0.211 ~ 'Area 10',
                             X < 0.8294 & Y > 0.789 ~ 'Area 10',
                             X < 0.7163 & Y >= 0.211 & Y <= 0.789 ~ 'Area 9',
                             X >= 0.8294 & Y < 0.211 ~ 'Area 8',
                             X >= 0.8294 & Y > 0.789 ~ 'Area 8',
                             X >= 0.7163 & X < 0.8294 & Y >= 0.211 & Y <= 0.3678 ~ outbox,
                             X >= 0.7163 & X < 0.8294 & Y >= 0.6322 & Y <= 0.789 ~ outbox,
                             X >= 0.7163 & X < 0.8294 & Y > 0.3678 & Y < 0.6322 ~ outbox,
                             X >= 0.8294 & Y >= 0.211 & Y < 0.3678 ~ insideBox,
                             X >= 0.8294 & Y >= 0.6322 & Y <= 0.789 ~ insideBox,
                             X >= 0.8294 & X <= 0.8846 & Y > 0.3678 & Y < 0.6322 ~ insideBox,
                             X > 0.8846 & X < 0.9423 & Y > 0.3678 & Y < 0.6322 ~ insideBox,
                             X >= 0.9423 & Y > 0.3678 & Y < 0.44222 ~ insideBox,
                             X >= 0.9423 & Y > 0.55778 & Y < 0.6322 ~ insideBox,
                             X >= 0.9423 & Y >= 0.44222 & Y <= 0.55778 ~ insideBox,
                             TRUE ~ 'e')))

unique(sort(areas_spec$xGAreas))

## [1] "Area 10"      "Area 8"       "Area 9"       "Inside Box"   "Outside Box"
xmin <- c(0.8294) *100
xmax <- c(1) *100
ymin <- c(0.211) *100
ymax <- c(0.789) *100
insideAreas <- c("Inside Box")

```

```

insideBox_areas <- data.frame(xmin, xmax, ymin, ymax, xGAreas)

outsideAreas <- c("Outside Box")
outBox_areas <- data.frame(0.7163, 0.8294, 0.211, 0.789 , xGAreas)

df_best <- setNames(data.frame(matrix(ncol =length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {
  df_best <- results_area(TRUE, areas_spec, df_best, strName, insideBox)
  listPlayers <- df_best$Player[1:n]
}
df_best

```

	Player	Conversion_Rate	Goals_Expected	Goals	Shots
## 20	Son Heung-Min	0.227	66.23	84	370
## 1	Lionel Messi	0.226	137.08	154	682
## 11	Ciro Immobile	0.180	94.86	110	612
## 17	Gonzalo Higuaín	0.196	85.83	98	499
## 9	Kylian Mbappe-Lottin	0.239	104.00	116	486
## 25	Iago Aspas	0.223	68.57	80	359
## 22	Erling Haaland	0.283	41.03	52	184
## 13	Antoine Griezmann	0.226	81.61	91	402
## 15	Sadio Mané	0.213	99.06	108	508
## 5	Harry Kane	0.209	121.72	130	621
## 16	Jamie Vardy	0.213	95.84	104	489
## 7	Mohamed Salah	0.193	115.30	123	637
## 19	Wissam Ben Yedder	0.223	92.59	99	443
## 21	Alexandre Lacazette	0.220	85.69	92	418
## 23	Zlatan Ibrahimovic	0.199	67.89	73	367
## 4	Luis Suárez	0.226	159.21	164	727
## 18	Mauro Icardi	0.218	97.60	102	467
## 10	Romelu Lukaku	0.204	108.50	112	548
## 3	Cristiano Ronaldo	0.181	159.85	162	893
## 8	Karim Benzema	0.197	122.11	123	623
## 6	Pierre-Emerick Aubameyang	0.240	140.50	140	583
## 14	Sergio Agüero	0.180	99.01	98	544
## 24	Neymar	0.208	97.94	88	424
## 2	Robert Lewandowski	0.210	200.26	188	894
## 12	Edinson Cavani	0.223	116.75	104	467
##	Average_xG	Performance_xG	Performance_Goals		
## 20	0.179	0.048	17.77		
## 1	0.201	0.025	16.92		
## 11	0.155	0.025	15.14		
## 17	0.172	0.024	12.17		
## 9	0.214	0.025	12.00		
## 25	0.191	0.032	11.43		
## 22	0.223	0.060	10.97		
## 13	0.203	0.023	9.39		
## 15	0.195	0.018	8.94		
## 5	0.196	0.013	8.28		
## 16	0.196	0.017	8.16		
## 7	0.181	0.012	7.70		
## 19	0.209	0.014	6.41		
## 21	0.205	0.015	6.31		
## 23	0.185	0.014	5.11		

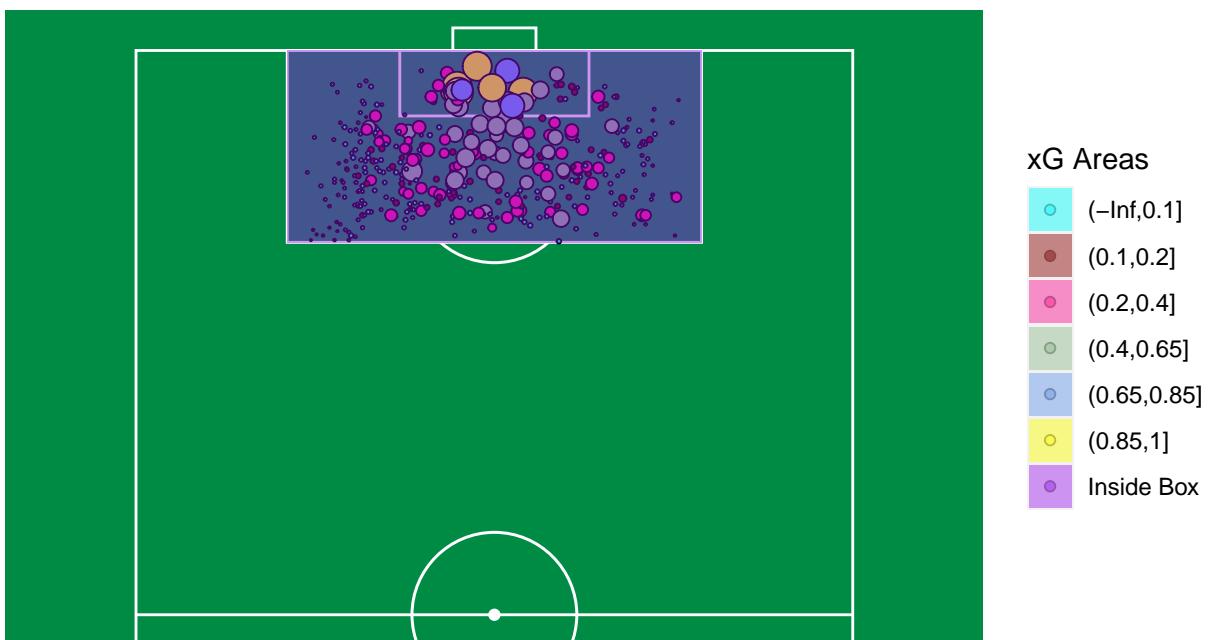
```

## 4      0.219      0.007      4.79
## 18     0.209      0.009      4.40
## 10     0.198      0.006      3.50
## 3      0.179      0.002      2.15
## 8      0.196      0.001      0.89
## 6      0.241      -0.001     -0.50
## 14     0.182      -0.002     -1.01
## 24     0.231      -0.023     -9.94
## 2      0.224      -0.014     -12.26
## 12     0.250      -0.027     -12.75

for (strName in listPlayers) {
  print(plot_areas(areas_spec, strName , insideBox_areas, insideAreas ))
}

```

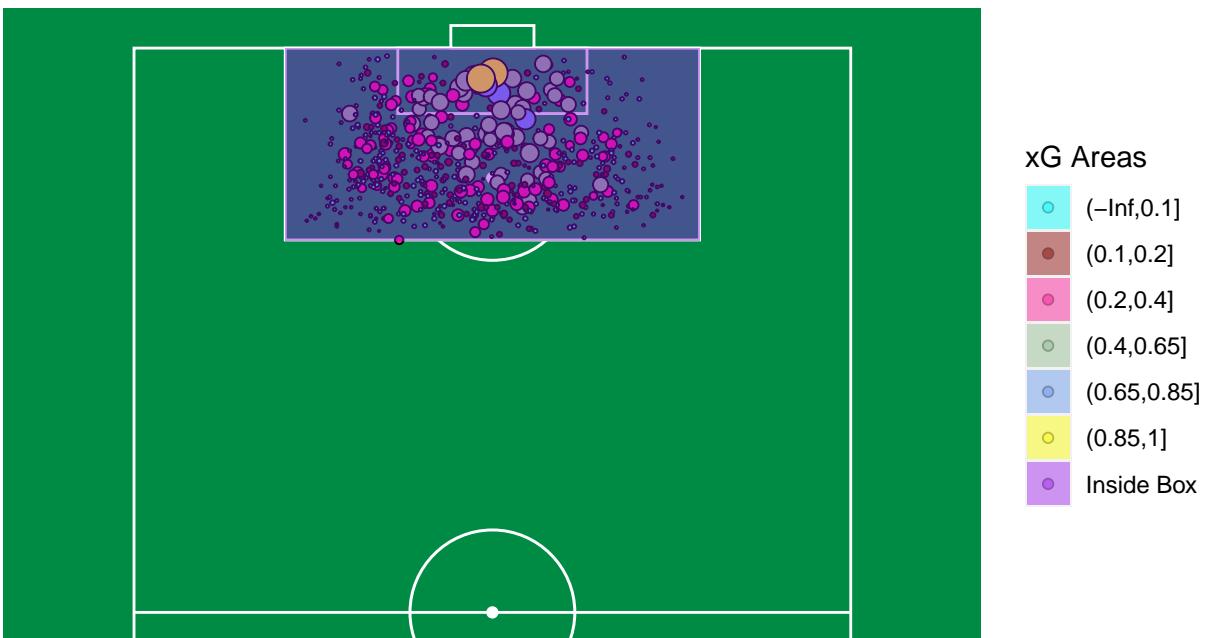
Son Heung–Min goalmap Inside Box



Lionel Messi goalmap Inside Box



Ciro Immobile goalmap Inside Box



Let's look who is the best outside the box.

```
df_best <- setNames(data.frame(matrix(ncol = length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {
  df_best <- results_area(TRUE, areas_spec, df_best, strName, outbox)
}
listPlayers <- df_best$Player[1:n]
df_best
```

	Player	Conversion_Rate	Goals_Expected	Goals	Shots
## 1	Lionel Messi	0.094	33.78	53	563

## 13	Antoine Griezmann	0.110	8.01	20	182
## 20	Son Heung-Min	0.090	6.59	16	178
## 23	Zlatan Ibrahimovic	0.102	7.11	13	127
## 2	Robert Lewandowski	0.090	9.30	15	166
## 14	Sergio Agüero	0.089	7.59	13	146
## 5	Harry Kane	0.057	12.72	18	318
## 9	Kylian Mbappe-Lottin	0.098	3.86	9	92
## 4	Luis Suárez	0.082	7.90	13	158
## 25	Iago Aspas	0.092	8.97	14	152
## 21	Alexandre Lacazette	0.099	5.99	11	111
## 12	Edinson Cavani	0.128	5.77	10	78
## 7	Mohamed Salah	0.056	8.62	11	196
## 24	Neymar	0.059	9.72	11	187
## 22	Erling Haaland	0.182	0.85	2	11
## 10	Romelu Lukaku	0.050	6.06	7	141
## 3	Cristiano Ronaldo	0.043	17.43	18	415
## 19	Wissam Ben Yedder	0.054	3.85	4	74
## 18	Mauro Icardi	0.046	2.86	3	65
## 17	Gonzalo Higuaín	0.042	7.94	8	189
## 16	Jamie Vardy	0.048	3.09	3	63
## 11	Ciro Immobile	0.032	5.93	5	156
## 6	Pierre-Emerick Aubameyang	0.035	5.02	4	114
## 8	Karim Benzema	0.031	5.59	4	127
## 15	Sadio Mané	0.018	4.93	2	112
##	Average_xG	Performance_xG	Performance_Goals		
## 1	0.060	0.034	19.22		
## 13	0.044	0.066	11.99		
## 20	0.037	0.053	9.41		
## 23	0.056	0.046	5.89		
## 2	0.056	0.034	5.70		
## 14	0.052	0.037	5.41		
## 5	0.040	0.017	5.28		
## 9	0.042	0.056	5.14		
## 4	0.050	0.032	5.10		
## 25	0.059	0.033	5.03		
## 21	0.054	0.045	5.01		
## 12	0.074	0.054	4.23		
## 7	0.044	0.012	2.38		
## 24	0.052	0.007	1.28		
## 22	0.077	0.105	1.15		
## 10	0.043	0.007	0.94		
## 3	0.042	0.001	0.57		
## 19	0.052	0.002	0.15		
## 18	0.044	0.002	0.14		
## 17	0.042	0.000	0.06		
## 16	0.049	-0.001	-0.09		
## 11	0.038	-0.006	-0.93		
## 6	0.044	-0.009	-1.02		
## 8	0.044	-0.013	-1.59		
## 15	0.044	-0.026	-2.93		

Over 25% of Messis goals is outside the box. He leads this list with around 19 goals more than expected. From a purely statistical point of view, Antoine Griezmann is the best performer in this list, followed by Mbappe and Cavani.

After analyzing the inside and outside area of the box, we will go in further detail for all the different 10 areas which we mentioned before.

Down under you can see the results.

Area 1

```
df_best <- setNames(data.frame(matrix(ncol =length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {
  df_best <- results_area(TRUE, areasDf, df_best, strName, "Area 1")
}
df_best
```

	Player	Conversion_Rate	Goals_Expected	Goals	Shots
## 25	Iago Aspas	0.654	14.48	17	26
## 11	Ciro Immobile	0.650	10.64	13	20
## 17	Gonzalo Higuaín	0.571	10.02	12	21
## 9	Kylian Mbappe-Lottin	0.778	12.53	14	18
## 22	Erling Haaland	0.733	9.66	11	15
## 4	Luis Suárez	0.617	27.82	29	47
## 18	Mauro Icardi	0.578	25.25	26	45
## 20	Son Heung-Min	0.750	11.66	12	16
## 19	Wissam Ben Yedder	0.600	17.70	18	30
## 14	Sergio Agüero	0.526	19.76	20	38
## 16	Jamie Vardy	0.545	18.28	18	33
## 5	Harry Kane	0.519	29.16	28	54
## 21	Alexandre Lacazette	0.500	17.18	16	32
## 15	Sadio Mané	0.486	18.48	17	35
## 23	Zlatan Ibrahimovic	0.429	10.56	9	21
## 6	Pierre-Emerick Aubameyang	0.559	40.19	38	68
## 7	Mohamed Salah	0.607	19.29	17	28
## 13	Antoine Griezmann	0.500	15.34	13	26
## 1	Lionel Messi	0.632	26.49	24	38
## 24	Neymar	0.536	17.72	15	28
## 10	Romelu Lukaku	0.434	27.56	23	53
## 8	Karim Benzema	0.471	28.61	24	51
## 3	Cristiano Ronaldo	0.444	37.01	32	72
## 12	Edinson Cavani	0.466	33.52	27	58
## 2	Robert Lewandowski	0.462	50.05	42	91
##	Average_xG	Performance_xG	Performance_Goals		
## 25	0.557	0.097	2.52		
## 11	0.532	0.118	2.36		
## 17	0.477	0.094	1.98		
## 9	0.696	0.082	1.47		
## 22	0.644	0.089	1.34		
## 4	0.592	0.025	1.18		
## 18	0.561	0.017	0.75		
## 20	0.729	0.021	0.34		
## 19	0.590	0.010	0.30		
## 14	0.520	0.006	0.24		
## 16	0.554	-0.009	-0.28		
## 5	0.540	-0.021	-1.16		
## 21	0.537	-0.037	-1.18		
## 15	0.528	-0.042	-1.48		
## 23	0.503	-0.074	-1.56		
## 6	0.591	-0.032	-2.19		

```

## 7      0.689      -0.082      -2.29
## 13     0.590      -0.090      -2.34
## 1      0.697      -0.065      -2.49
## 24     0.633      -0.097      -2.72
## 10     0.520      -0.086      -4.56
## 8      0.561      -0.090      -4.61
## 3      0.514      -0.070      -5.01
## 12     0.578      -0.112      -6.52
## 2      0.550      -0.088      -8.05

Area 2

df_best <- setNames(data.frame(matrix(ncol = length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {
  df_best <- results_area(TRUE, areasDf, df_best, strName, "Area 2")
}
df_best

##          Player Conversion_Rate Goals_Expected Goals Shots
## 13      Antoine Griezmann      0.333        7.36    11   33
## 11      Ciro Immobile       0.282        8.81    11   39
## 21 Alexandre Lacazette      0.257        7.42     9   35
## 19      Wissam Ben Yedder     0.312        8.48    10   32
## 20      Son Heung-Min       0.312        3.68     5   16
## 25      Iago Aspas          0.381        6.97     8   21
## 16      Jamie Vardy         0.312        4.11     5   16
## 22      Erling Haaland       0.800        3.14     4     5
## 10      Romelu Lukaku        0.259        6.99     7   27
## 18      Mauro Icardi         0.244       11.30    11   45
## 23      Zlatan Ibrahimovic    0.182        7.06     6   33
## 17      Gonzalo Higuaín       0.211        5.11     4   19
## 7       Mohamed Salah         0.182        7.62     6   33
## 9       Kylian Mbappe-Lottin    0.227        7.26     5   22
## 24      Neymar                 0.250        8.54     6   24
## 8       Karim Benzema         0.133        7.29     4   30
## 6 Pierre-Emerick Aubameyang   0.225       12.44    9   40
## 4       Luis Suárez           0.200       14.46   11   55
## 5       Harry Kane             0.159       10.78    7   44
## 3       Cristiano Ronaldo      0.146       10.74    6   41
## 15      Sadio Mané              0.071       7.06    2   28
## 12      Edinson Cavani         0.103        8.55    3   29
## 14      Sergio Agüero          0.088        8.94    3   34
## 2       Robert Lewandowski      0.135       13.00    7   52
## 1       Lionel Messi            0.154       12.52    6   39
##          Average_xG Performance_xG Performance_Goals
## 13      0.223        0.110        3.64
## 11      0.226        0.056        2.19
## 21      0.212        0.045        1.58
## 19      0.265        0.047        1.52
## 20      0.230        0.082        1.32
## 25      0.332        0.049        1.03
## 16      0.257        0.055        0.89
## 22      0.628        0.172        0.86
## 10      0.259        0.000        0.01
## 18      0.251       -0.007       -0.30

```

```

## 23      0.214      -0.032      -1.06
## 17      0.269      -0.058      -1.11
## 7       0.231      -0.049      -1.62
## 9       0.330      -0.103      -2.26
## 24      0.356      -0.106      -2.54
## 8       0.243      -0.110      -3.29
## 6       0.311      -0.086      -3.44
## 4       0.263      -0.063      -3.46
## 5       0.245      -0.086      -3.78
## 3       0.262      -0.116      -4.74
## 15      0.252      -0.181      -5.06
## 12      0.295      -0.192      -5.55
## 14      0.263      -0.175      -5.94
## 2       0.250      -0.115      -6.00
## 1       0.321      -0.167      -6.52

Area 3

df_best <- setNames(data.frame(matrix(ncol = length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {
  df_best <- results_area(TRUE, areasDf, df_best, strName, "Area 3")
}
df_best

##          Player Conversion_Rate Goals_Expected Goals Shots
## 10      Romelu Lukaku      0.252        43.05     52   206
## 13      Antoine Griezmann    0.315        30.26     39   124
## 20      Son Heung-Min      0.365        23.63     31    85
## 25      Iago Aspas         0.274        24.41     31   113
## 7       Mohamed Salah       0.292        43.43     50   171
## 23      Zlatan Ibrahimovic  0.265        32.49     39   147
## 8       Karim Benzema       0.264        54.03     60   227
## 15      Sadio Mané          0.250        42.43     48   192
## 3       Cristiano Ronaldo    0.220        64.55     70   318
## 1       Lionel Messi         0.303        53.82     59   195
## 4       Luis Suárez          0.297        68.39     73   246
## 5       Harry Kane           0.244        45.51     50   205
## 14      Sergio Agüero        0.259        39.95     44   170
## 22      Erling Haaland        0.275        14.97     19    69
## 9       Kylian Mbappe-Lottin   0.331        36.66     40   121
## 12      Edinson Cavani        0.260        44.53     47   181
## 21      Alexandre Lacazette   0.244        31.96     32   131
## 2       Robert Lewandowski    0.221        89.24     89   402
## 6       Pierre-Emerick Aubameyang 0.260        56.55     56   215
## 17      Gonzalo Higuaín        0.244        40.84     40   164
## 19      Wissam Ben Yedder       0.246        36.21     35   142
## 16      Jamie Vardy            0.197        40.59     39   198
## 11      Ciro Immobile          0.192        30.95     29   151
## 18      Mauro Icardi           0.179        44.46     42   234
## 24      Neymar                  0.258        45.30     39   151

##          Average_xG Performance_xG Performance_Goals
## 10      0.209        0.043        8.95
## 13      0.244        0.071        8.74
## 20      0.278        0.087        7.37
## 25      0.216        0.058        6.59

```

```

## 7      0.254      0.038      6.57
## 23     0.221      0.044      6.51
## 8      0.238      0.026      5.97
## 15     0.221      0.029      5.57
## 3      0.203      0.017      5.45
## 1      0.276      0.027      5.18
## 4      0.278      0.019      4.61
## 5      0.222      0.022      4.49
## 14     0.235      0.024      4.05
## 22     0.217      0.058      4.03
## 9      0.303      0.028      3.34
## 12     0.246      0.014      2.47
## 21     0.244      0.000      0.04
## 2      0.222      -0.001     -0.24
## 6      0.263      -0.003     -0.55
## 17     0.249      -0.005     -0.84
## 19     0.255      -0.009     -1.21
## 16     0.205      -0.008     -1.59
## 11     0.205      -0.013     -1.95
## 18     0.190      -0.011     -2.46
## 24     0.300      -0.042     -6.30

```

Area 4

```

df_best <- setNames(data.frame(matrix(ncol = length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {
  df_best <- results_area(TRUE, areasDf, df_best, strName, "Area 4")
}
df_best

```

	Player	Conversion_Rate	Goals_Expected	Goals	Shots
## 1	Lionel Messi	0.209	29.61	49	235
## 17	Gonzalo Higuain	0.175	14.38	24	137
## 3	Cristiano Ronaldo	0.196	19.45	28	143
## 9	Kylian Mbappe-Lottin	0.261	20.98	29	111
## 20	Son Heung-Min	0.253	12.09	20	79
## 5	Harry Kane	0.241	18.14	26	108
## 11	Ciro Immobile	0.181	20.15	28	155
## 24	Neymar	0.234	11.40	18	77
## 7	Mohamed Salah	0.172	22.17	28	163
## 15	Sadio Mané	0.172	20.99	26	151
## 8	Karim Benzema	0.164	17.15	22	134
## 6	Pierre-Emerick Aubameyang	0.176	14.58	19	108
## 16	Jamie Vardy	0.214	13.61	18	84
## 2	Robert Lewandowski	0.169	32.59	36	213
## 25	Iago Aspas	0.178	10.15	13	73
## 18	Mauro Icardi	0.167	10.84	13	78
## 13	Antoine Griezmann	0.176	13.52	15	85
## 10	Romelu Lukaku	0.161	18.60	20	124
## 19	Wissam Ben Yedder	0.163	15.39	16	98
## 4	Luis Suárez	0.162	22.44	23	142
## 22	Erling Haaland	0.125	5.04	5	40
## 23	Zlatan Ibrahimovic	0.116	8.21	8	69
## 21	Alexandre Lacazette	0.146	15.96	15	103
## 12	Edinson Cavani	0.169	17.09	15	89

```

## 14           Sergio Agüero      0.054      11.54     6    112
##   Average_xG Performance_xG Performance_Goals
## 1       0.126      0.083      19.39
## 17      0.105      0.070      9.62
## 3       0.136      0.060      8.55
## 9       0.189      0.072      8.02
## 20      0.153      0.100      7.91
## 5       0.168      0.073      7.86
## 11      0.130      0.051      7.85
## 24      0.148      0.086      6.60
## 7       0.136      0.036      5.83
## 15      0.139      0.033      5.01
## 8       0.128      0.036      4.85
## 6       0.135      0.041      4.42
## 16      0.162      0.052      4.39
## 2       0.153      0.016      3.41
## 25      0.139      0.039      2.85
## 18      0.139      0.028      2.16
## 13      0.159      0.017      1.48
## 10      0.150      0.011      1.40
## 19      0.157      0.006      0.61
## 4       0.158      0.004      0.56
## 22      0.126     -0.001     -0.04
## 23      0.119     -0.003     -0.21
## 21      0.155     -0.009     -0.96
## 12      0.192     -0.023     -2.09
## 14      0.103     -0.049     -5.54

```

Area 5

```

df_best <- setNames(data.frame(matrix(ncol = length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {
  df_best <- results_area(TRUE, areasDf, df_best, strName, "Area 5")
}
df_best

```

	Player	Conversion_Rate	Goals_Expected	Goals	Shots
## 21	Alexandre Lacazette	0.171	13.34	20	117
## 14	Sergio Agüero	0.132	18.62	25	190
## 19	Wissam Ben Yedder	0.142	14.80	20	141
## 15	Sadio Mané	0.147	10.20	15	102
## 11	Ciro Immobile	0.117	24.21	29	247
## 22	Erling Haaland	0.236	8.25	13	55
## 16	Jamie Vardy	0.152	19.43	24	158
## 18	Mauro Icardi	0.154	5.98	10	65
## 17	Gonzalo Higuaín	0.114	15.48	18	158
## 4	Luis Suárez	0.118	25.83	28	237
## 9	Kylian Mbappe-Lottin	0.131	26.54	28	214
## 23	Zlatan Ibrahimovic	0.113	9.60	11	97
## 1	Lionel Messi	0.091	14.70	16	175
## 6	Pierre-Emerick Aubameyang	0.118	16.87	18	152
## 5	Harry Kane	0.090	18.06	19	210
## 20	Son Heung-Min	0.092	15.14	16	174
## 7	Mohamed Salah	0.091	22.99	22	242
## 12	Edinson Cavani	0.109	13.31	12	110

```

## 25          Iago Aspas      0.087    12.35   11   126
## 2          Robert Lewandowski 0.103    15.50   14   136
## 10         Romelu Lukaku     0.072    12.01   10   138
## 13         Antoine Griezmann 0.097    15.01   13   134
## 8          Karim Benzema     0.072    15.02   13   181
## 3          Cristiano Ronaldo 0.082    28.07   26   319
## 24         Neymar           0.069    14.83   10   144
## Average_xG Performance_xG Performance_Goals
## 21         0.114      0.057      6.66
## 14         0.098      0.034      6.38
## 19         0.105      0.037      5.20
## 15         0.100      0.047      4.80
## 11         0.098      0.019      4.79
## 22         0.150      0.086      4.75
## 16         0.123      0.029      4.57
## 18         0.092      0.062      4.02
## 17         0.098      0.016      2.52
## 4          0.109      0.009      2.17
## 9          0.124      0.007      1.46
## 23         0.099      0.014      1.40
## 1          0.084      0.007      1.30
## 6          0.111      0.007      1.13
## 5          0.086      0.004      0.94
## 20         0.087      0.005      0.86
## 7          0.095      -0.004     -0.99
## 12         0.121      -0.012     -1.31
## 25         0.098      -0.011     -1.35
## 2          0.114      -0.011     -1.50
## 10         0.087      -0.015     -2.01
## 13         0.112      -0.015     -2.01
## 8          0.083      -0.011     -2.02
## 3          0.088      -0.006     -2.07
## 24         0.103      -0.034     -4.83

```

Area 6

```

df_best <- setNames(data.frame(matrix(ncol =length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {
  df_best <- results_area(TRUE, areasDf, df_best, strName, "Area 6")
}
df_best

```

	Player	Conversion_Rate	Goals_Expected	Goals	Shots
## 1	Lionel Messi	0.108	26.62	45	416
## 13	Antoine Griezmann	0.194	5.56	20	103
## 20	Son Heung-Min	0.122	4.41	11	90
## 5	Harry Kane	0.081	8.26	14	172
## 2	Robert Lewandowski	0.107	7.32	13	122
## 21	Alexandre Lacazette	0.123	4.74	9	73
## 23	Zlatan Ibrahimovic	0.106	5.36	9	85
## 25	Iago Aspas	0.101	6.63	10	99
## 14	Sergio Agüero	0.076	5.67	8	105
## 3	Cristiano Ronaldo	0.063	10.76	13	207
## 9	Kylian Mbappé-Lottin	0.121	1.88	4	33
## 12	Edinson Cavani	0.128	3.95	6	47

```

## 24           Neymar      0.078      6.39     8    103
## 4            Luis Suárez  0.068      5.66     7    103
## 22          Erling Haaland 0.222      0.74     2     9
## 18          Mauro Icardi  0.071      2.10     3    42
## 11          Ciro Immobile 0.048      4.37     5   104
## 6  Pierre-Emerick Aubameyang 0.056      3.67     4    72
## 19          Wissam Ben Yedder 0.057      2.06     2    35
## 10          Romelu Lukaku  0.045      4.49     4    88
## 17          Gonzalo Higuaín 0.040      6.70     6   149
## 16          Jamie Vardy    0.027      2.11     1    37
## 8            Karim Benzema  0.037      4.18     3    82
## 7            Mohamed Salah   0.039      6.53     5   128
## 15          Sadio Mané     0.024      4.16     2    85
##   Average_xG Performance_xG Performance_Goals
## 1        0.064        0.044       18.38
## 13       0.054        0.140       14.44
## 20       0.049        0.073       6.59
## 5         0.048        0.033       5.74
## 2         0.060        0.047       5.68
## 21       0.065        0.058       4.26
## 23       0.063        0.043       3.64
## 25       0.067        0.034       3.37
## 14       0.054        0.022       2.33
## 3         0.052        0.011       2.24
## 9         0.057        0.064       2.12
## 12       0.084        0.044       2.05
## 24       0.062        0.016       1.61
## 4         0.055        0.013       1.34
## 22       0.082        0.140       1.26
## 18       0.050        0.021       0.90
## 11       0.042        0.006       0.63
## 6         0.051        0.005       0.33
## 19       0.059       -0.002      -0.06
## 10       0.051       -0.006      -0.49
## 17       0.045       -0.005      -0.70
## 16       0.057       -0.030      -1.11
## 8         0.051       -0.014      -1.18
## 7         0.051       -0.012      -1.53
## 15       0.049       -0.025      -2.16

```

Area 7

```

df_best <- setNames(data.frame(matrix(ncol =length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {
  df_best <- results_area(TRUE, areasDf, df_best, strName, "Area 7")
}
df_best

```

	Player	Conversion_Rate	Goals_Expected	Goals	Shots
## 7	Mohamed Salah	0.088	2.11	6	68
## 4	Luis Suárez	0.109	2.20	6	55
## 14	Sergio Agüero	0.122	1.84	5	41
## 9	Kylian Mbappe-Lottin	0.085	2.01	5	59
## 20	Son Heung-Min	0.057	2.20	5	88
## 23	Zlatan Ibrahimovic	0.095	1.81	4	42

## 12	Edinson Cavani	0.129	1.83	4	31
## 25	Iago Aspas	0.075	2.23	4	53
## 10	Romelu Lukaku	0.057	1.64	3	53
## 16	Jamie Vardy	0.077	0.99	2	26
## 21	Alexandre Lacazette	0.053	1.25	2	38
## 17	Gonzalo Higuaín	0.050	1.28	2	40
## 1	Lionel Messi	0.054	7.35	8	147
## 19	Wissam Ben Yedder	0.051	1.79	2	39
## 2	Robert Lewandowski	0.045	2.02	2	44
## 22	Erling Haaland	0.000	0.11	0	2
## 24	Neymar	0.036	3.28	3	84
## 5	Harry Kane	0.027	4.38	4	146
## 8	Karim Benzema	0.022	1.44	1	45
## 15	Sadio Mané	0.000	0.76	0	27
## 18	Mauro Icardi	0.000	0.80	0	23
## 6	Pierre-Emerick Aubameyang	0.000	1.39	0	42
## 11	Ciro Immobile	0.000	1.61	0	52
## 3	Cristiano Ronaldo	0.024	6.86	5	208
## 13	Antoine Griezmann	0.000	2.53	0	79
<b>## Average_xG Performance_xG Performance_Goals</b>					
## 7	0.031	0.057	3.89		
## 4	0.040	0.069	3.80		
## 14	0.045	0.077	3.16		
## 9	0.034	0.051	2.99		
## 20	0.025	0.032	2.80		
## 23	0.043	0.052	2.19		
## 12	0.059	0.070	2.17		
## 25	0.042	0.033	1.77		
## 10	0.031	0.026	1.36		
## 16	0.038	0.039	1.01		
## 21	0.033	0.020	0.75		
## 17	0.032	0.018	0.72		
## 1	0.050	0.004	0.65		
## 19	0.046	0.005	0.21		
## 2	0.046	-0.001	-0.02		
## 22	0.054	-0.054	-0.11		
## 24	0.039	-0.003	-0.28		
## 5	0.030	-0.003	-0.38		
## 8	0.032	-0.010	-0.44		
## 15	0.028	-0.028	-0.76		
## 18	0.035	-0.035	-0.80		
## 6	0.033	-0.033	-1.39		
## 11	0.031	-0.031	-1.61		
## 3	0.033	-0.009	-1.86		
## 13	0.032	-0.032	-2.53		

Area 8

```
df_best <- setNames(data.frame(matrix(ncol =length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {
  df_best <- results_area(TRUE, areasDf, df_best, strName, "Area 8")
}
df_best
```

##	Player	Conversion_Rate	Goals_Expected	Goals_Shots
## 12	Edinson Cavani	0.129	1.83	4
## 25	Iago Aspas	0.075	2.23	4
## 10	Romelu Lukaku	0.057	1.64	3
## 16	Jamie Vardy	0.077	0.99	2
## 21	Alexandre Lacazette	0.053	1.25	2
## 17	Gonzalo Higuaín	0.050	1.28	2
## 1	Lionel Messi	0.054	7.35	8
## 19	Wissam Ben Yedder	0.051	1.79	2
## 2	Robert Lewandowski	0.045	2.02	2
## 22	Erling Haaland	0.000	0.11	0
## 24	Neymar	0.036	3.28	3
## 5	Harry Kane	0.027	4.38	4
## 8	Karim Benzema	0.022	1.44	1
## 15	Sadio Mané	0.000	0.76	0
## 18	Mauro Icardi	0.000	0.80	0
## 6	Pierre-Emerick Aubameyang	0.000	1.39	0
## 11	Ciro Immobile	0.000	1.61	0
## 3	Cristiano Ronaldo	0.024	6.86	5
## 13	Antoine Griezmann	0.000	2.53	0

## 15	Sadio Mané	0.5	0.04	1	2
## 5	Harry Kane	0.2	0.11	1	5
## 11	Ciro Immobile	0.2	0.30	1	5
## 1	Lionel Messi	0.1	0.61	1	10
## 25	Iago Aspas	0.2	0.64	1	5
## 4	Luis Suárez	0.0	0.03	0	2
## 13	Antoine Griezmann	0.0	0.06	0	2
## 20	Son Heung-Min	0.0	0.06	0	2
## 21	Alexandre Lacazette	0.0	0.06	0	3
## 12	Edinson Cavani	0.0	0.07	0	2
## 9	Kylian Mbappe-Lottin	0.0	0.08	0	2
## 8	Karim Benzema	0.0	0.10	0	2
## 16	Jamie Vardy	0.0	0.27	0	4
## 24	Neymar	0.0	0.29	0	5
## 3	Cristiano Ronaldo	0.0	0.75	0	15
## 2	Robert Lewandowski	NaN	NaN	0	0
## 6	Pierre-Emerick Aubameyang	NaN	NaN	0	0
## 7	Mohamed Salah	NaN	NaN	0	0
## 10	Romelu Lukaku	NaN	NaN	0	0
## 14	Sergio Agüero	NaN	NaN	0	0
## 17	Gonzalo Higuaín	NaN	NaN	0	0
## 18	Mauro Icardi	NaN	NaN	0	0
## 19	Wissam Ben Yedder	NaN	NaN	0	0
## 22	Erling Haaland	NaN	NaN	0	0
## 23	Zlatan Ibrahimovic	NaN	NaN	0	0
<b>## Average_xG Performance_xG Performance_Goals</b>					
## 15	0.022	0.478	0.96		
## 5	0.022	0.178	0.89		
## 11	0.061	0.139	0.70		
## 1	0.061	0.039	0.39		
## 25	0.129	0.071	0.36		
## 4	0.013	-0.013	-0.03		
## 13	0.031	-0.031	-0.06		
## 20	0.029	-0.029	-0.06		
## 21	0.021	-0.021	-0.06		
## 12	0.033	-0.033	-0.07		
## 9	0.040	-0.040	-0.08		
## 8	0.049	-0.049	-0.10		
## 16	0.067	-0.067	-0.27		
## 24	0.057	-0.057	-0.29		
## 3	0.050	-0.050	-0.75		
## 2	NaN	NaN	NaN		
## 6	NaN	NaN	NaN		
## 7	NaN	NaN	NaN		
## 10	NaN	NaN	NaN		
## 14	NaN	NaN	NaN		
## 17	NaN	NaN	NaN		
## 18	NaN	NaN	NaN		
## 19	NaN	NaN	NaN		
## 22	NaN	NaN	NaN		
## 23	NaN	NaN	NaN		

Area 9

```

df_best <- setNames(data.frame(matrix(ncol =length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {
  df_best <- results_area(TRUE, areasDf, df_best, strName, "Area 9")
}
df_best

```

	Player	Conversion_Rate	Goals_Expected	Goals	Shots
## 12	Edinson Cavani	0.400	0.24	2	5
## 20	Son Heung-Min	0.154	0.30	2	13
## 8	Karim Benzema	0.500	0.45	2	4
## 5	Harry Kane	0.057	0.80	2	35
## 7	Mohamed Salah	0.333	0.03	1	3
## 19	Wissam Ben Yedder	0.100	0.37	1	10
## 13	Antoine Griezmann	0.059	0.46	1	17
## 3	Cristiano Ronaldo	0.036	1.48	2	55
## 4	Luis Suárez	0.045	0.66	1	22
## 23	Zlatan Ibrahimovic	0.031	0.90	1	32
## 16	Jamie Vardy	0.000	0.02	0	2
## 14	Sergio Agüero	0.000	0.03	0	1
## 21	Alexandre Lacazette	0.000	0.04	0	3
## 17	Gonzalo Higuaín	0.000	0.05	0	4
## 18	Mauro Icardi	0.000	0.05	0	5
## 9	Kylian Mbappe-Lottin	0.000	0.11	0	5
## 15	Sadio Mané	0.000	0.12	0	6
## 2	Robert Lewandowski	0.000	0.14	0	5
## 10	Romelu Lukaku	0.000	0.14	0	4
## 1	Lionel Messi	0.033	1.17	1	30
## 11	Ciro Immobile	0.000	0.19	0	15
## 25	Iago Aspas	0.000	0.26	0	12
## 24	Neymar	0.000	0.35	0	12
## 6	Pierre-Emerick Aubameyang	0.000	0.50	0	16
## 22	Erling Haaland	NaN	NaN	0	0
	Average_xG	Performance_xG	Performance_Goals		
## 12	0.048	0.352	1.76		
## 20	0.023	0.131	1.70		
## 8	0.112	0.388	1.55		
## 5	0.023	0.034	1.20		
## 7	0.011	0.322	0.97		
## 19	0.037	0.063	0.63		
## 13	0.027	0.032	0.54		
## 3	0.027	0.009	0.52		
## 4	0.030	0.015	0.34		
## 23	0.028	0.003	0.10		
## 16	0.011	-0.011	-0.02		
## 14	0.034	-0.034	-0.03		
## 21	0.015	-0.015	-0.04		
## 17	0.012	-0.012	-0.05		
## 18	0.010	-0.010	-0.05		
## 9	0.021	-0.021	-0.11		
## 15	0.020	-0.020	-0.12		
## 2	0.028	-0.028	-0.14		
## 10	0.034	-0.034	-0.14		
## 1	0.039	-0.006	-0.17		
## 11	0.013	-0.013	-0.19		

```

## 25      0.022      -0.022      -0.26
## 24      0.029      -0.029      -0.35
## 6       0.031      -0.031      -0.50
## 22      NaN        NaN        NaN

Area 10

df_best <- setNames(data.frame(matrix(ncol =length(col_header), nrow = 0)), col_header)
for (strName in listPlayersFinal) {
  df_best <- results_area(TRUE, areasDf, df_best, strName, "Area 10")
}
df_best

##          Player Conversion_Rate Goals_Expected Goals Shots
## 20      Son Heung-Min        1.00        0.03     1     1
## 5       Harry Kane          0.25        0.08     1     4
## 10      Romelu Lukaku        0.00        0.01     0     1
## 11      Ciro Immobile        0.00        0.01     0     1
## 12      Edinson Cavani        0.00        0.01     0     2
## 19      Wissam Ben Yedder        0.00        0.01     0     1
## 25      Iago Aspas          0.00        0.01     0     1
## 14      Sergio Agüero        0.00        0.02     0     1
## 18      Mauro Icardi         0.00        0.02     0     1
## 4       Luis Suárez          0.00        0.04     0     5
## 9       Kylian Mbappe-Lottin        0.00        0.04     0     3
## 13      Antoine Griezmann        0.00        0.04     0     2
## 23      Zlatan Ibrahimovic        0.00        0.05     0     2
## 7       Mohamed Salah          0.00        0.06     0     3
## 24      Neymar              0.00        0.06     0     3
## 16      Jamie Vardy           0.00        0.20     0     1
## 1       Lionel Messi           0.00        0.32     0     9
## 3       Cristiano Ronaldo        0.00        0.44     0    11
## 2       Robert Lewandowski        NaN        NaN     0     0
## 6   Pierre-Emerick Aubameyang        NaN        NaN     0     0
## 8       Karim Benzema           0.00        0.00     0     0
## 15      Sadio Mané             0.00        0.00     0     0
## 17      Gonzalo Higuaín          0.00        0.00     0     0
## 21      Alexandre Lacazette        0.00        0.00     0     0
## 22      Erling Haaland            0.00        0.00     0     0
##          Average_xG Performance_xG Performance_Goals
## 20      0.028        0.972        0.97
## 5       0.020        0.230        0.92
## 10      0.015       -0.015       -0.01
## 11      0.012       -0.012       -0.01
## 12      0.007       -0.007       -0.01
## 19      0.011       -0.011       -0.01
## 25      0.012       -0.012       -0.01
## 14      0.021       -0.021       -0.02
## 18      0.019       -0.019       -0.02
## 4        0.008       -0.008       -0.04
## 9        0.012       -0.012       -0.04
## 13      0.020       -0.020       -0.04
## 23      0.024       -0.024       -0.05
## 7        0.020       -0.020       -0.06
## 24      0.021       -0.021       -0.06

```

```
## 16      0.201      -0.201      -0.20
## 1       0.036      -0.036      -0.32
## 3       0.040      -0.040      -0.44
## 2        NaN        NaN        NaN
## 6        NaN        NaN        NaN
## 8        NaN        NaN        NaN
## 15      NaN        NaN        NaN
## 17      NaN        NaN        NaN
## 21      NaN        NaN        NaN
## 22      NaN        NaN        NaN
```

#Herewith the notebook is completed. Big Thanks for sticking this long. We hope you enjoyed our analysis.  
We appreciate your feedback. See you next time!