Projet_Women_FIFA_WC23_Analysis

2024-05-17

Contents

1 Introduction

In today's world, sports are at the center of global culture. To continue to excel, players and teams must find solutions, both physical and tactical. Therefore, statistics will play a crucial role in optimizing performance. Previous studies have shed light on various aspects of football analytics. Collet studied the impact of possession in 2013. More recently Liu analyzed the environmental impact in 2021. However, the realm of soccer remains relatively untapped in terms of data exploration. Understanding the dynamics of offensive and defensive play is pivotal for teams aiming to excel in competitions.

The research gap lies in the need for a comprehensive analysis of football performance using advanced statistical methods, with a focus on data from platforms like StatsBomb. The impact of certain specific aspects of football analytics, such as shot analysis or passing patterns, remains unclear, and a comprehensive understanding of player and team performance is still lacking.

We aimed to address this gap by conducting a detailed analysis of football performance using StatsBomb data. We seeked to identify key performance indicators, assess their impact on match outcomes, and uncover underlying trends and patterns in player and team performance. This report outlines the methodology used for data collection and analysis, presents the findings from the study, and discusses their implications for the future of football analytics.

This report is divided into three parts. In the first section, we conduct an exploratory data analysis to identify certain trends, notably by analyzing shots and goals for each team. Then we seek an optimal statistical model to determine which parameters have the greatest impact on player performance. The last section contains the results of our analysis, including insights into player and team performance derived from StatsBomb data, with graphs examining successful shots and passes.

2 Descriptive data analysis

The package StatsbombR provides the data from 71 national and international competitions, for a total of over 3000 matches. For the sake of this project, we narrow our scope down to the most recent competition available : The FIFA Women's World Cup 2023 and its 64 matches. We begin by interpreting the different variables of this large data set.

The full data set contains 183 variables for analysis, with the majority having a significant proportion of missing values, as they were used to track very specific patterns of play. As an example, the parameter "goalkeeper.shot_saved_to_post" is attributed "True" only if the goalkeeper saved a shot from going inside the goal, by deflecting it onto a post.

2.1 Analysis of successful shots according to country

First, we will look at how the data is stored in the data set. We will do this by plotting the shots of a specific match; we chose Spain-England, which was the final match of the competition.

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
## Warning: Removed 1 rows containing missing values ('geom_rect()').
```

```
## Warning: Removed 1 rows containing missing values (‘geom_segment()‘).

## Warning: Removed 1 rows containing missing values (‘geom_rect()‘).

## Warning: Removed 1 rows containing missing values (‘geom_point()‘).
## Removed 1 rows containing missing values (‘geom_point()‘).

## Warning: Removed 30 rows containing missing values (‘geom_path()‘).
```

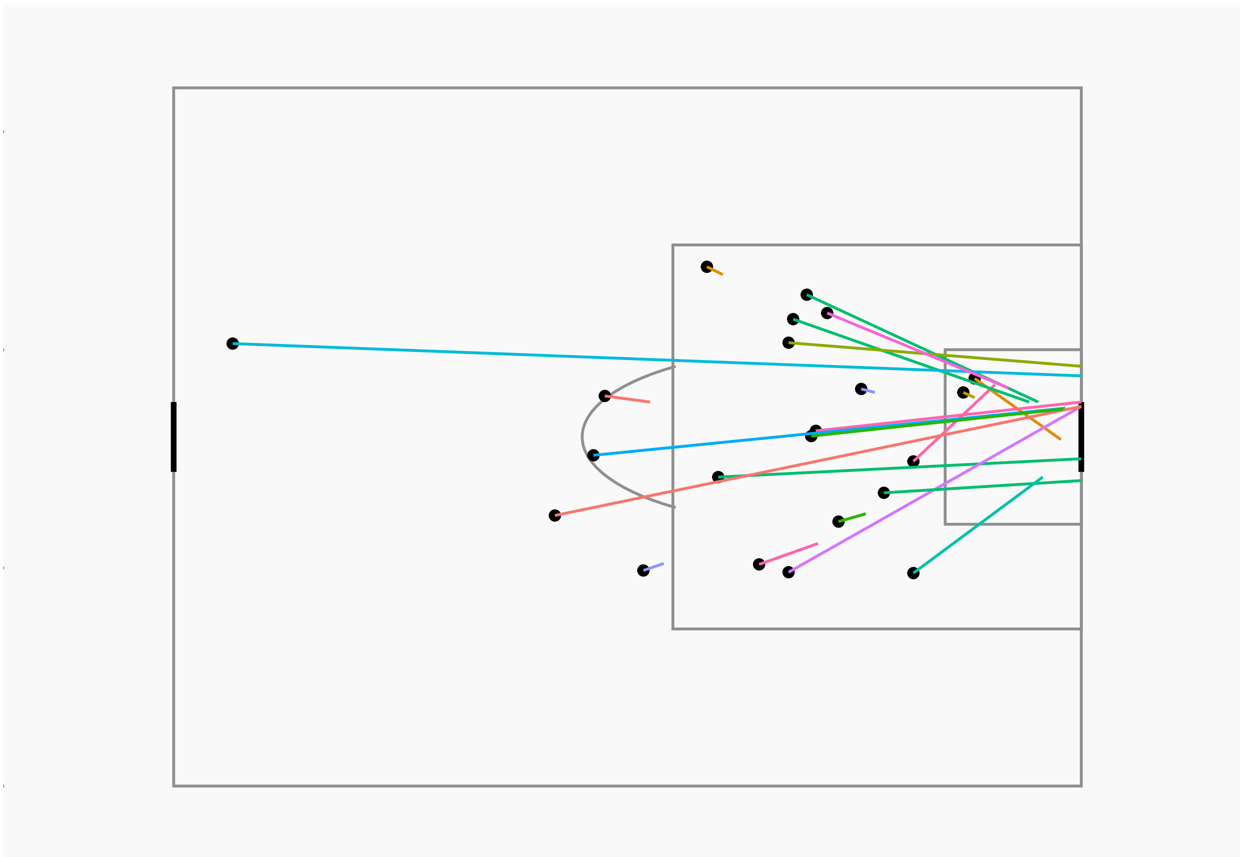


Figure 1: Visualization of the shots of Spain-England on the pitch

We can see that regardless of which team scores, the locations of the players are always tracked the same way : for a given team, the home goal is located at x = 0, which corresponds to the left of this graph, and the opposing goal is located at x = 120. This will allow us to directly use the provided variables, without further formatting the data.

Then, we took a look at the number of goals and shots in all matches for each team.
Figure 2 shows a visualization of these results.

From the above figure, we can already see a big disparity in team success. Some teams, like Vietnam and Haiti, didn't even manage to score a goal over the course of the competition, while Sweden and France were more successful in this aspect. However, simply displaying how much shots and goals a team made provides an incomplete understanding of team effectiveness, and is generally a flawed metric for comparison, as teams that made it further into the competition naturally scored more goals, and had more shots.

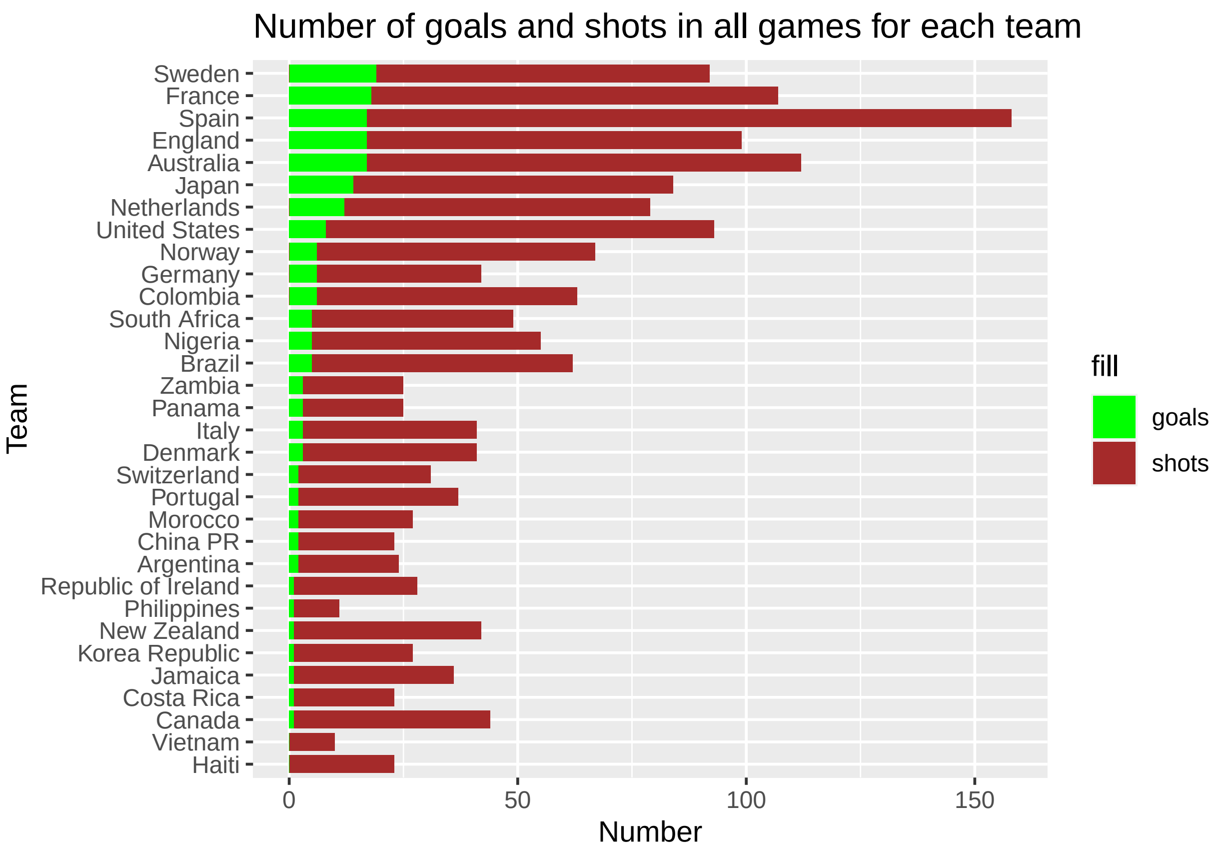


Figure 2: Diagram of the number of goals and shots in all matches for each team

Therefore, we calculated the percentage of shots leading to a goal for every team, and compared them in Figure 3.

From this graph, we are able to identify that Sweden was the most efficient team in scoring by over 2.5%. This is of course very biased, as Sweden finished third in the tournament, only losing one match throughout the entirety of the competition. Keeping that in mind, it is surprising that Spain scores this low on the graph, considering they won the World Cup. It could be that Spain, despite their evident success, was not a very efficient team, or that they had a different playing style than other teams.

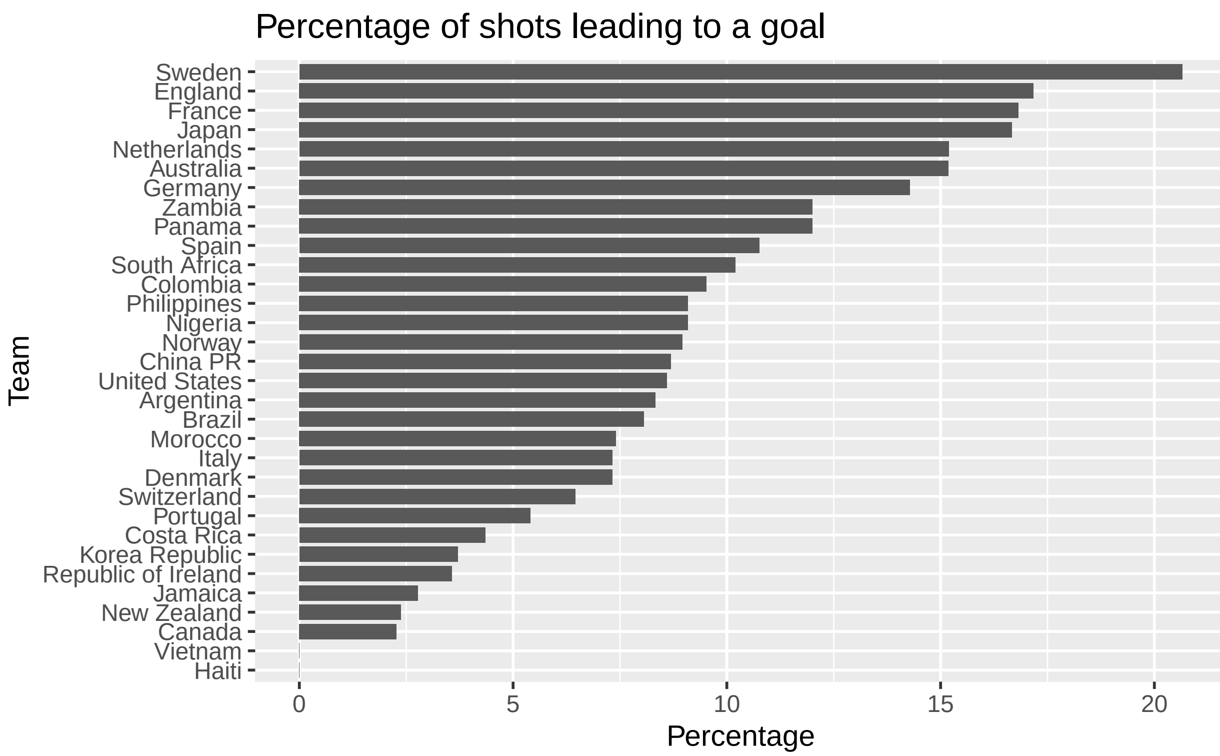


Figure 3: Diagram of the percentage of shots leading to a goal

Next, we wanted to realize the above analyses for singular matches. We chose the four matches played by team France, and created the same graphs for each of their matches. Figure 4 shows the results.

As we can see, there were a huge number of goals in two games : Panama France and Australia France. The first match ended on a exceptional 3-6 score for France, but the former ended on a nil-nil. However, teams went to penalties, and they scored a total of 13, making this observation flawed.

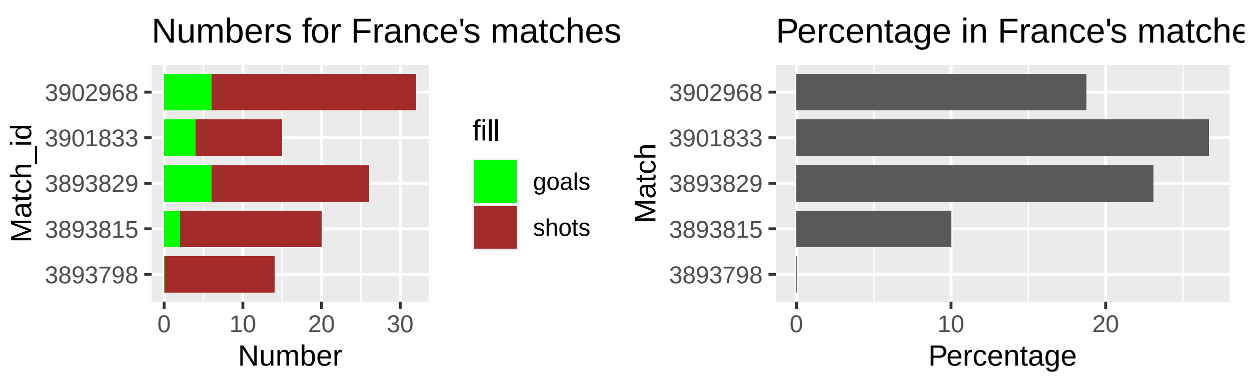


Figure 4: Diagram of the number of shots and goals for each French and percentage

2.2 Analysis of successful shots according to different variables

We first look at the different types of shots in Figure 4. Four different types of shots were differentiated in the data set : "Open Play", "Penalty", "Free Kick" as well as "Corner." A shot was deemed as being "Open Play" if it was taken during regular actions of the game. The "Corner" label only applies to a single shot made by Ireland's Katie McCabe, and went in. This is something to keep in mind, as it is sure to skew our future models. Other labels are self-explanatory.

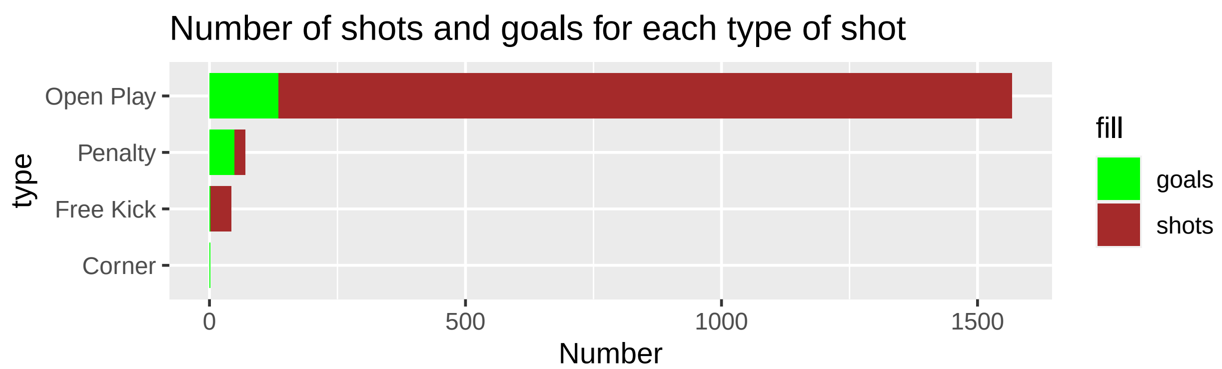


Figure 5: Diagram of the number of shots and goals for each type of shot

Next, we look at the different techniques used by players : how much are being kept a track in the data set, how much were each of them used, and which one produced the most goals.

Figure 5 shows that data set contains seven types of shots, although only three are consistently being used, that is : "Normal", "Half Volley" and "Volley". Naturally, the "Normal" shot was the most popular, and hence yielded the mos t goals. A "Lob" was very infrequently done, but could prove effective in the right situation : it seems a good portion of these shots went in.

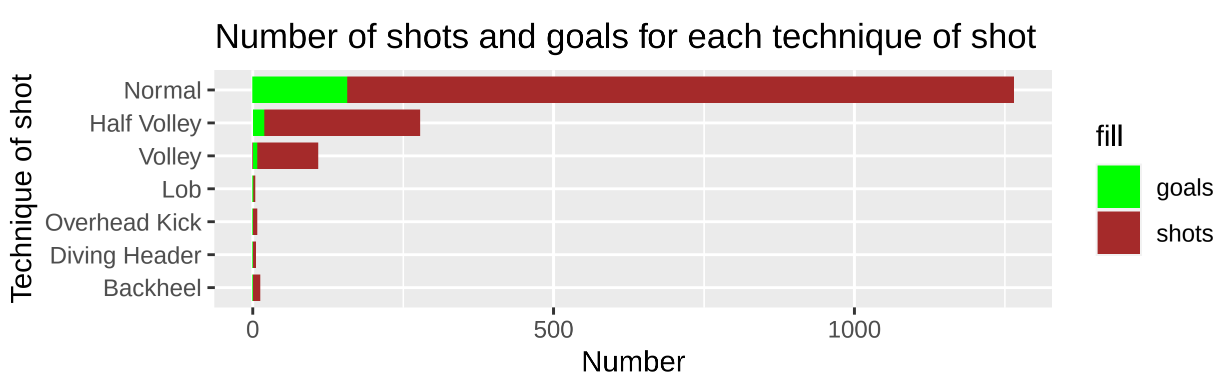
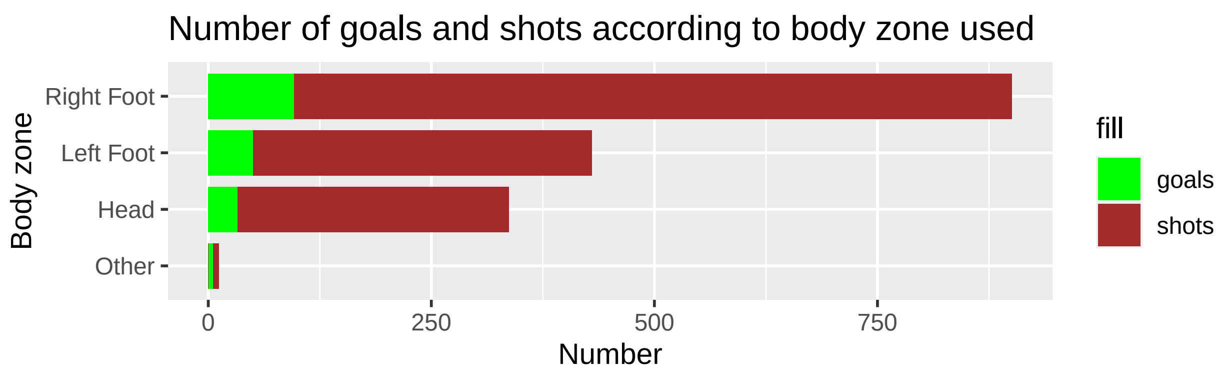


Figure 6: Diagram of the number of shots and goals for each technique of shot

Similarly, we visualize in Figure 6 the different body parts used in shooting.

Unsurprisingly, the right foot was most commonly used, and it seems every body part was equally as effective in scoring, apart from the "Other" body zone.



##Analysis of teams and players scoring ratio

In this part we want to describe the number of goals scored by players and teams by using the goal ratio, so the number of goals scored on the number of totals shots during the events.

###Players scoring ratio

Firstly, we looked at players, this may lead to find outliers so players which have performed a lot of shots but only few goals, and the opposite players which attempted only few shots but scored on a lot of them. This is a very basic way to start taking an interest in performance of players during this Word Cup.

Figure 7 shows a visualization of the goal ratio on the total shots performed by players which have performed at least 1 shots during the World Cup.

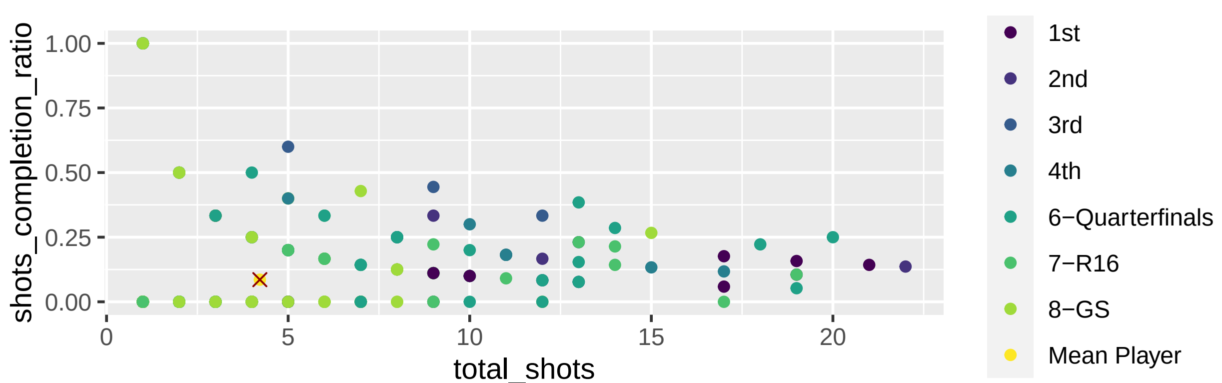


Figure 7: Goal ratio on the number of totals shots for players

The graph shows a clear trend: the more players have shot, the lower their ratio. Even so, the players on the best teams, those who finished in the best positions, often have a higher ratio than those on teams who lost earlier in the tournament. We also notice that the average (the red cross) is quite low compared to the points we see on the graph, intuitively we'd probably have placed it around 10 shots and 0.25 ratio. But then we realized that many points, especially those at the bottom of the graph with 0 goals scored, are superimposed, so there are many players who shot without scoring, which is fairly consistent with the match statistics of around ten shots per team for an average of between 1 and 4 goals per match. Now we miss an important information on this graphics. Indeed we can see that the most players attempted shots the less their ratio of successful shots is high, but we don't take into account the difficulty of the shots perfomed by the players. Indeed a penalty is intuitively easier to score than an open shot far for the goals and under the pressure of defenders. It's something we already saw in the Figure 5, that goals on penalty occurred more often even though that there are less frequent in games.

So, we ask ourselves base on which criteria we could try to implement the notion of shot's difficulty, and we find that the notion of Expected Goals is what we were looking for and that the Statsbomb dataset provide it with every shots attempt.

The "Expected Goal", often named "xG" is the probability to score given a lot of datas on the shot, for example it can be the shhoter and the goal position, the fact that the striker is under pressure or not, with which foot he is attempting this shot and various differents variables. This probability is computed by the xG model of Statsbomb which is probably created on a large amount of data that they were able to collect.

So this notion of xG is perfect to implement the notion of shots difficulty. We can compute the mean of the xG of every shots that one player attempted to see if the player had easy or hard shots to perform.

This his what the Figure **??** is showing true goal ratio on the Statsbomb xG ratio for every player which attempted at least 1 shots during the World Cup./

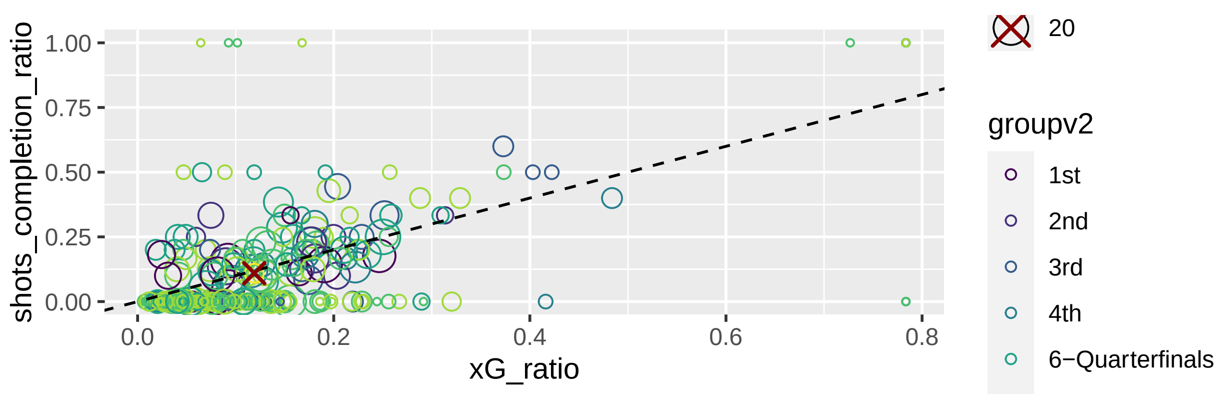


Figure 8: Goal ratio on the Expected Goal ratio (Statsbomb model) for players

So on this graph we can see that BLABLABLA.

###Teams scoring ratio

Secondly, we decided to look at teams, we are running the same analysis as for the players but this could lead to other analysis. Indeed it's reducing the number of outliers, because we are conduction an analysis on all team players that have shot at least once during the tournament. Indeed if a midfielder player has shot only once and scored and a striker has shot 10 times for only 2 goals this bias of the midfielder has less impact on her team. So, with this team analysis we could say overall which team underperform inter of goals, and try to find if it's correlated to the team who lost earliest during the competition.

Figure **??** shows a visualization of the goal ratio on the total shots performed by teams which have performed at least 1 shots during the World Cup.

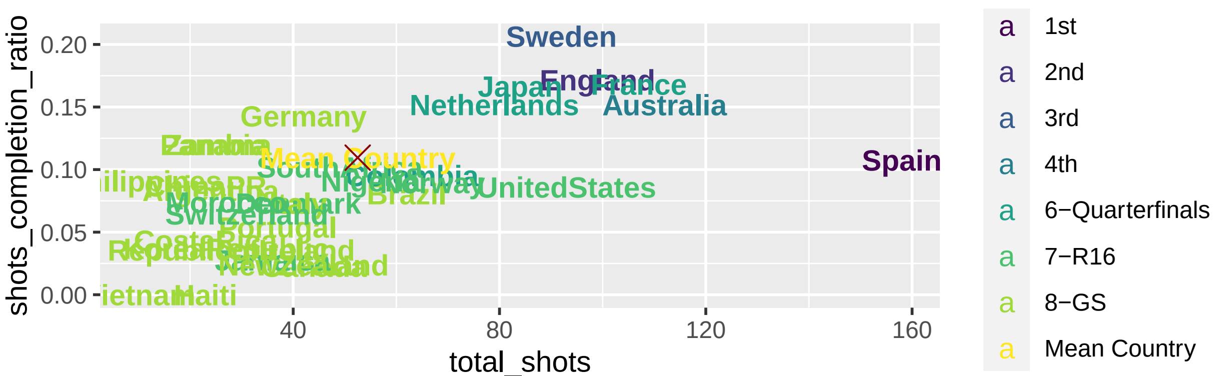


Figure 9: Goal ratio on the number of total shots for teams

So on this graph we can see that BLABLABLA.

As for the players wwe want to implement the Expected Goal data, so we can implement and analyze the dificulty of shots performed by teams.

This his what the Figure **??** is showing true goal ratio on the Statsbomb xG ratio for every teams during the World Cup./

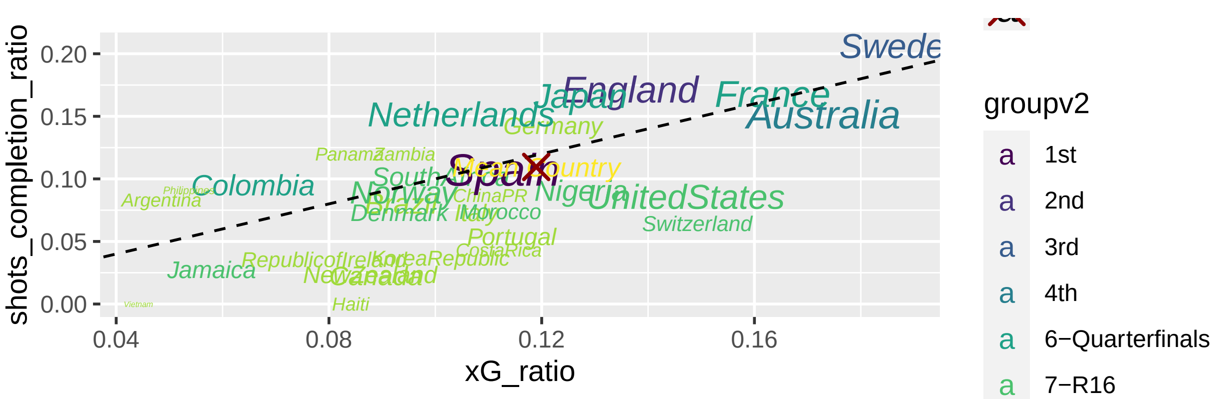


Figure 10: Goal ratio on the Expected Goal ratio (Statsbomb model) for teams

So on this graph we can see that BLABLABLA.

Now that we have compared the true goal ratio to the xG Statsbomb ratio, we were wondering how exactly is one xG model working. So, we decided to create our own xG model.

3 Models analysis

We wanted to create our own xG model. To do that we developed different models, finding the most relevant variables to predict goals.

We run a logistic regression model: we want the output to be 0 or 1 depending on whether the shot turns into a goal.

3.1 First model : body part, technique, type of shot

The first model keeps the variables studied previously : body part, technique, type of shot.

R^2 for the model without interaction is : 0.144105

R^2 for the model with interaction is : 0.1460338 .

3.2 Our target model : the expected goal variable

We now create a model composed of a single variable: the expected goal given in StatsBomb.

Our goal in creating the different models in this section is to find the most accurate model possible, which can have an R^2 close to this model (with only the expected goal as a variable), i.e. an R^2 close to : 0.262161.

3.3 Model 3 : Adding location.x and location.y

Any player on the field is assimilated as a moving point on a rectangle of size 80x120m. Its horizontal movement - that is, going from one goal post to another - is tracked by the variable location.x, while the vertical movement is associated to location.y. We will now add location.x and location.y to our previously adjusted model.

We test a regression without interaction, and obtain an R^2 of : 0.2054155 .

With interactions, we get an R^2 of : 0.2323348.

In this model, we targeted the main variables to obtain a good model and an R^2 as close to 1 as possible.

3.3.1 Significance of variables ?

We run several tests to see which variables are significant in the model.

```
## Analysis of Deviance Table
##
## Model 1: shot.outcome.name ~ (location.x + location.y + shot.body_part.name +
```

```
##     shot.type.name)^2
## Model 2: shot.outcome.name ~ (shot.body_part.name + shot.technique.name +
##     shot.type.name + location.x + location.y)^2
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      1658     917.76
## 2      1637     891.22 21   26.542   0.1865
```

We see that we can remove the technique because $p - value$>0.05 so we can accept the sub-model with a 95% level.

For this sub-model without the technique variable we obtain an R^2 of : 0.2094726

The R^2 is no greater than for model 3 with interactions: this is normal because the R^2 favors models with many variables.
We should look at other variables such as AIC score, which is minimal for model 3 without interactions.

3.3.2 Comparison of norms

We now want to compare model 3 with and without interaction : the closer the 2-norm is to 0, the better the model.

Norm L2 for the model_3 without interaction is equal to : 4.1435027.

The value for the model_3 with interactions is : 4.2588035.

We find the same results as with the AIC criterion. This is consistent with the fact that R^2 favors models with many variables, so it's better to evaluate with AIC. We can conclude that the model 3 without interaction is best.

We do the same to compare model 1 with and without interaction.

The L2 norms are respectively : 4.6785642 and 4.6786731.

Both models are less accurate than the 3rd one.

3.4 Model 4 : adding the under_pressure variable

We now create a new model like the model_3, but adding a variable : under_pressure.

3.4.1 Test for significance of single variables

First, we test the significance of this new variable.

```
##
## Call:
## glm(formula = shot.outcome.name ~ under_pressure, family = binomial(link = "logit"),
##     data = df_model_4)
##
## Coefficients:
```

```
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -1.94591    0.09486 -20.513   <2e-16 ***
## under_pressureTRUE -0.41957    0.16790  -2.499   0.0125 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1160.9  on 1679  degrees of freedom
## Residual deviance: 1154.5  on 1678  degrees of freedom
## AIC: 1158.5
##
## Number of Fisher Scoring iterations: 5
```

We see that $p_{\text{value}} < 0.05$, so we reject H_0 : playing under pressure is significant.

Estimated coefficients are negative, so playing under pressure reduces the probability of scoring.

Testing the model with only the shot.body_part.name variable gives us a p_{value} of : 0.042.
We reject H_0, the technique variable is significant.

We now want to test the model with only the shot.technique.name variable.

```
##
## Call:
## glm(formula = shot.outcome.name ~ shot.technique.name, family = binomial(link = "logit"),
##     data = df_model_4)
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -1.657e+01  6.927e+02  -0.024    0.981
## shot.technique.nameDiving Header  8.145e-08  1.277e+03   0.000    1.000
## shot.technique.nameHalf Volley    1.395e+01  6.927e+02   0.020    0.984
## shot.technique.nameLob            1.547e+01  6.927e+02   0.022    0.982
## shot.technique.nameNormal         1.461e+01  6.927e+02   0.021    0.983
## shot.technique.nameOverhead Kick  8.143e-08  1.141e+03   0.000    1.000
## shot.technique.nameVolley         1.389e+01  6.927e+02   0.020    0.984
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1160.9  on 1679  degrees of freedom
## Residual deviance: 1143.9  on 1673  degrees of freedom
## AIC: 1157.9
##
## Number of Fisher Scoring iterations: 15
```

The reference is backheel : all the other techniques are better, we have a lot of values close to 1, we could do a constant sub-model to see if this variable is significant.

We find a p_{value} of 0.009. We reject H_0, the technique variable is significant.

We are now testing the model with only the shot.type.name variable. We also run a sub-model test.

We find a p_{value} of 0.

The variable shot.type.name is significant, we reject H_0.

We do the same with the variable location.x :

We see a p_{value} of : 0. <0.05 so location.x is highly significant.

We check if the variable location.y is significant as well.

The p_{value} is : 0.915. $>$ 0.05 so location.y is not significant.

3.4.2 Testing the complete model

The model is now tested with all the following variables: shot.body_part.name,shot.technique.name,shot.type.name,location.x

We have an R^2 of 0.2054992 which is good, but it's normal because it's a model with many variables.

We also note a low AIC, which is equal to `954.3696823`.

3.5 Model 5 : adding the position of the goalkeeper

We create the same model as above, but adding the position of the goalkeeper.

3.5.1 Does the position of the goalkeeper in x and y improve our results ?

First we test the model with only the location.x.GK variable.

```
##
## Call:
## glm(formula = shot.outcome.name ~ location.x.GK, family = binomial(link = "logit"),
##     data = df_model_6)
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -36.22586    4.49351  -8.062 7.52e-16 ***
## location.x.GK   0.28758    0.03775   7.617 2.59e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1160.9  on 1679  degrees of freedom
## Residual deviance: 1104.7  on 1678  degrees of freedom
## AIC: 1108.7
##
## Number of Fisher Scoring iterations: 5
```

```
## [1] 0.04846669
```

Significant effect of goal position in x because both p_{values} are lower than 0.5.
The AIC value is low, equals to `1108.6754173`.

Then we do the same but with the location.y.GK variable.

We find that the variable for keeper position in y is significant as well. AIC is slightly higher than for position in x, it's equal to `1164.4331267`.

3.5.2 Complete model

For the model with all the preceding variables and without interaction, we find a very low AIC=`950.1303381`. We can conclude that this model is really good.

3.6 Keeping all significant variables and removing location.y

Since we found that location.y is not significant, we can remove it from the model.

Without this variable, the AIC is even lower, at `866.1274211`.

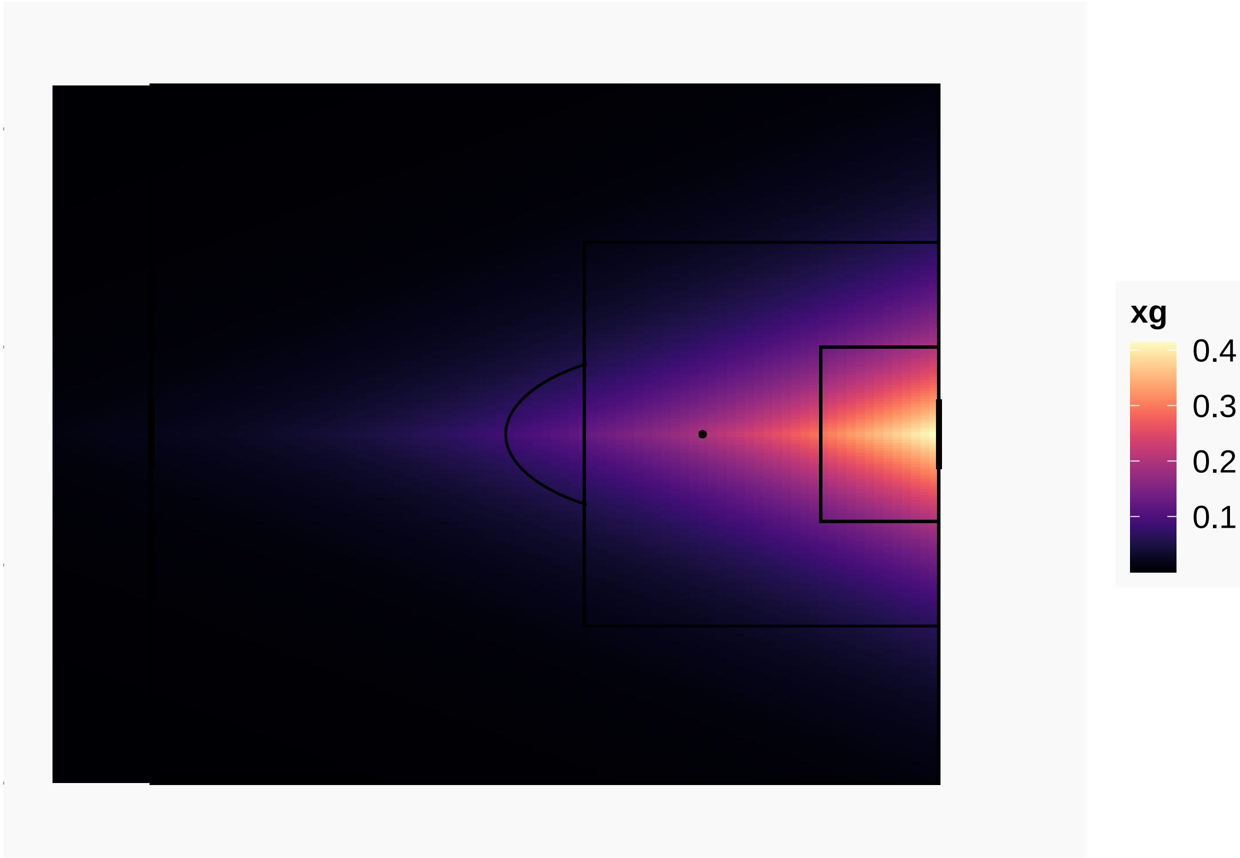
We can conclude that we have found our best model for now and it's composed of the variables :

body_part, shot.technique, shot.type, location.x, under_pressure, location.x.GK, location.y.GK.

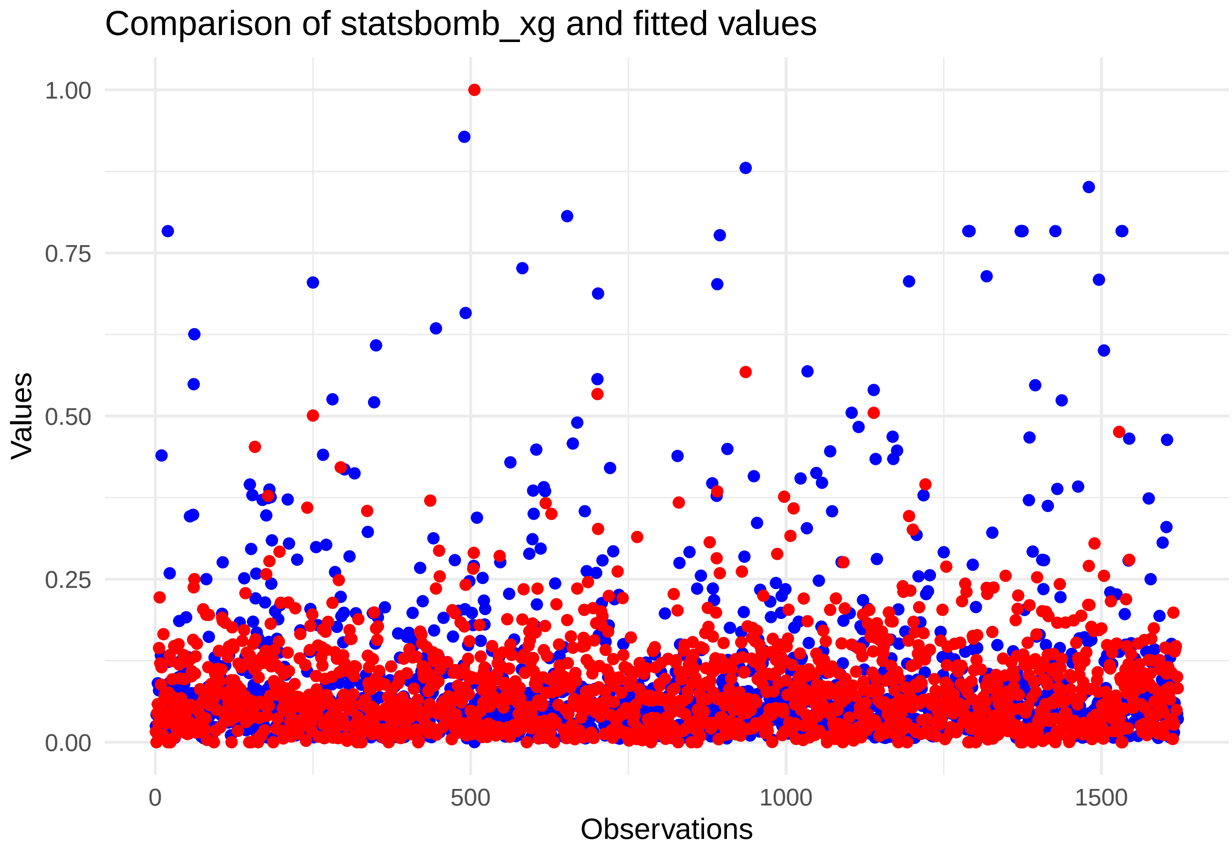
3.7 Replacing location.y

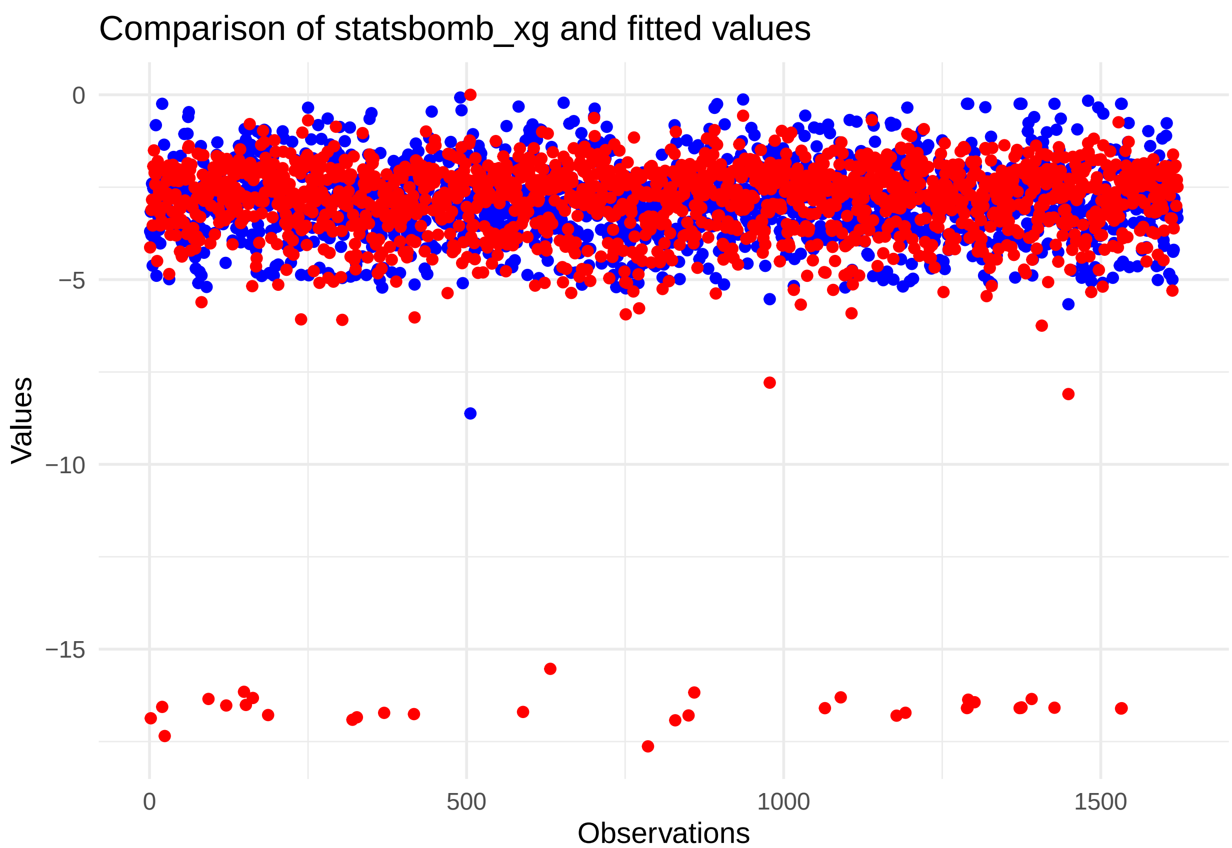
Seeing as location.y is a very insignificant variable, we now view offense as symmetrical with respect to the axis passing by the center point of the pitch and the penalty spots. We will now refer to location.y as the distance to center, and the new variable will vary from 0 to 40 meters. This hopefully will give more sense to the parameter, as a high distance to center means a player is very off-center.

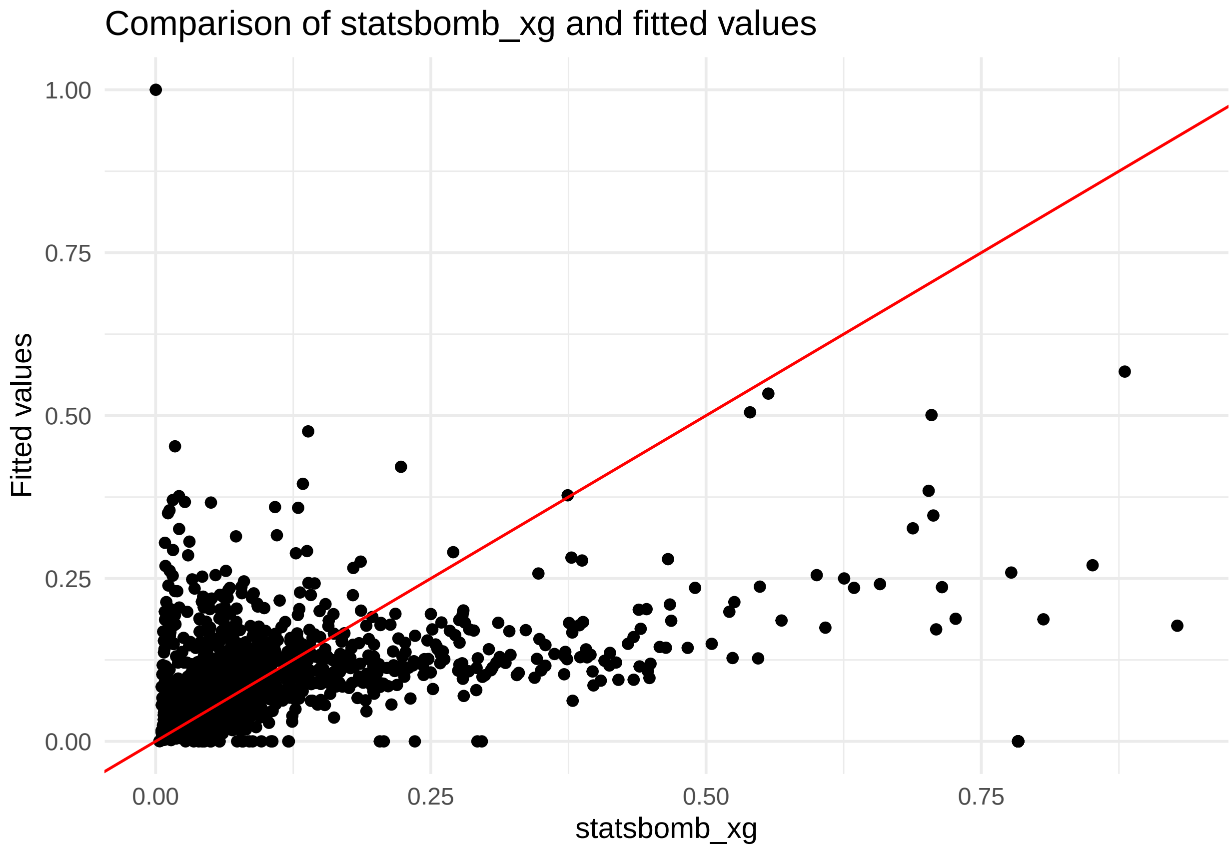
```
## Warning: Removed 40960 rows containing missing values (‘geom_tile()‘).

## Warning: Removed 65536 rows containing missing values (‘geom_rect()‘).

## Warning: Removed 65536 rows containing missing values (‘geom_segment()‘).

## Warning: Removed 65536 rows containing missing values (‘geom_rect()‘).

## Warning: Removed 65536 rows containing missing values (‘geom_point()‘).
## Removed 65536 rows containing missing values (‘geom_point()‘).

## Warning: Removed 30 rows containing missing values (‘geom_path()‘).
```

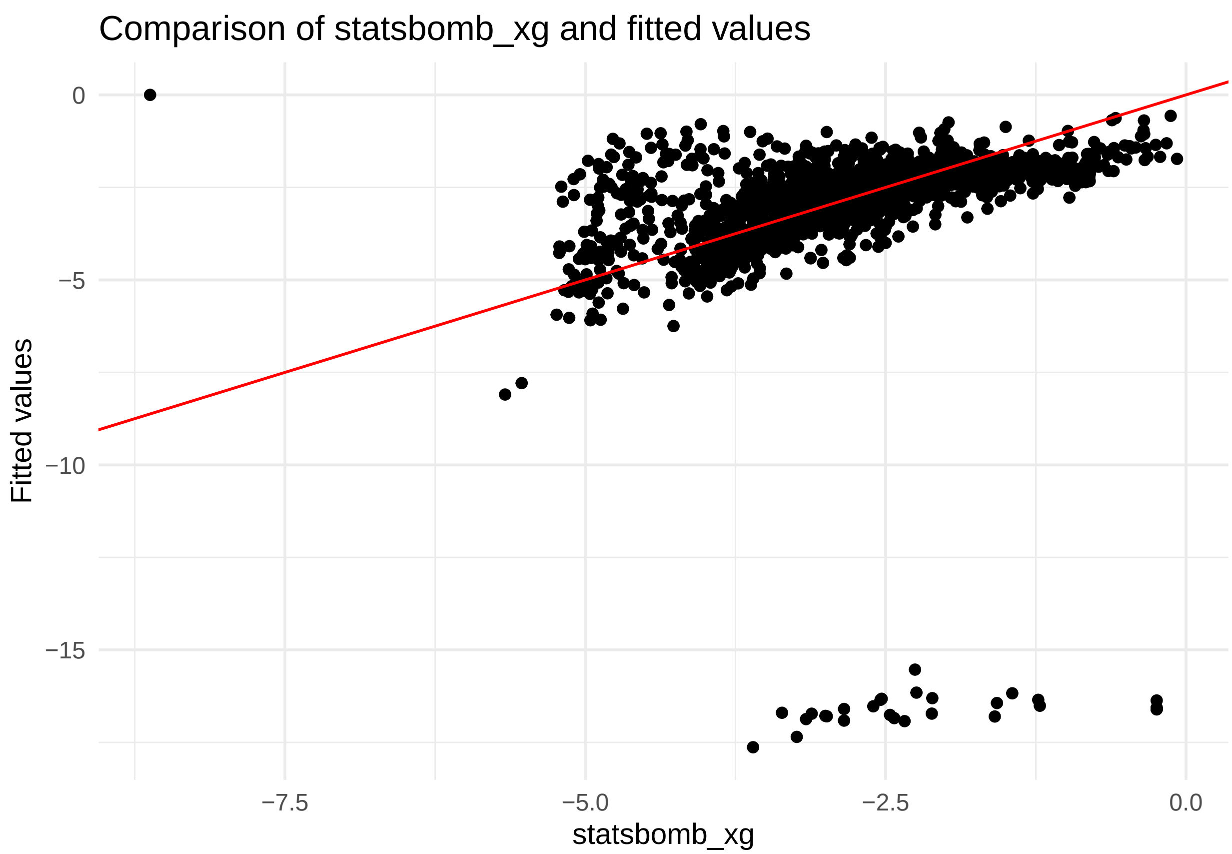


The above heatmap shows what is to be expected : from a very simple model, only using the location as well as the distance to center, we can see that the model expects a player to have better scoring chances with point-blank shots rather than shots outside the penalty area. This is explained by the fact that our data set does not contain many goals made from outside this area.


Comparison of statsbomb_xg and fitted values
1.00
0.75
0.50
0.25
0.00
Values
0
500
1000
1500
Observations


Comparison of statsbomb_xg and fitted values
Values
Observations
0
−5
−10
−15
0
500
1000
1500


Comparison of statsbomb_xg and fitted values
Fitted values
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
statsbomb_xg



We have a lot of values close to 0, so we do the log to make things clearer.

In log: there's a point (an observation) where we've overestimated the chance of scoring. There are a few points at the bottom right where, on the contrary, we've underestimated the probability, but overall we've got a good prediction based on the Xg of bomb stats.

3.8 Finding our best model with the AIC criterion

```
##
## Call:
## glm(formula = shot.outcome.name ~ shot.body_part.name + shot.technique.name +
##     shot.type.name + location.x + location.x.GK, family = binomial(link = "logit"),
##     data = df_model_6)
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        -0.26950 2496.42516   0.000  0.99991
## shot.body_part.nameLeft Foot        0.73885    0.28416   2.600  0.00932 **
## shot.body_part.nameOther            1.66474    0.62359   2.670  0.00759 **
## shot.body_part.nameRight Foot       0.61594    0.26028   2.366  0.01796 *
## shot.technique.nameDiving Header    1.15111 1270.44869   0.001  0.99928
## shot.technique.nameHalf Volley     14.61311  688.68298   0.021  0.98307
## shot.technique.nameLob             16.50566  688.68416   0.024  0.98088
## shot.technique.nameNormal          15.26610  688.68296   0.022  0.98231
## shot.technique.nameOverhead Kick    0.58723 1122.64416   0.001  0.99958
## shot.technique.nameVolley          14.48016  688.68305   0.021  0.98323
```

```
## shot.type.nameFree Kick           -17.10201 2399.54497  -0.007  0.99431
## shot.type.nameOpen Play           -16.88807 2399.54474  -0.007  0.99438
## shot.type.namePenalty             -13.31112 2399.54477  -0.006  0.99557
## location.x                          0.13002    0.01748   7.439 1.02e-13 ***
## location.x.GK                      -0.12563    0.04842  -2.594  0.00948 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1160.94  on 1679  degrees of freedom
## Residual deviance:  916.26  on 1665  degrees of freedom
## AIC: 946.26
##
## Number of Fisher Scoring iterations: 15
```

We can see finally that y-positions are useless even for the goalkeeper, only x-positions are significant.

Our best model is composed of 5 variables : The technique of shot, the type of shot, the body part used and the positions in x for the player and the goalkeeper.

4 Analysis of the Model performannce

Now that we have our model composed of these 5 variables, we asked ourselves if this type of model is better than the Statsbomb xG one, and overall we wanted to test the perfomance of our model ! So as we previsouly did in the "Descriptive Analysis" part of our report we are conducting a comparison on the ratio of goals. Before, we did it between data we were given, but now we will predict the xG of every shot with our best model and then compare these xG value to the Statsbomb one on graphics.

Figure **??** shows a visualization of these results.

```
## # A tibble: 1,680 x 10
##    shot.outcome.name shot.body_part.name shot.technique.name shot.type.name
##                <dbl> <chr>               <chr>               <chr>
##  1                 0 Right Foot          Normal              Open Play
##  2                 0 Right Foot          Overhead Kick       Open Play
##  3                 0 Head                Normal              Open Play
##  4                 0 Right Foot          Half Volley         Open Play
##  5                 0 Left Foot           Normal              Open Play
##  6                 0 Right Foot          Normal              Open Play
##  7                 0 Right Foot          Normal              Open Play
##  8                 0 Left Foot           Normal              Open Play
##  9                 0 Head                Normal              Open Play
## 10                 1 Right Foot          Half Volley         Open Play
## # i 1,670 more rows
## # i 6 more variables: location.x <dbl>, location.y <dbl>, under_pressure <lgl>,
## #   location.x.GK <dbl>, location.y.GK <dbl>, xG_predi <dbl>
```

```
## [1] 398
```

```
## [1] 398
```

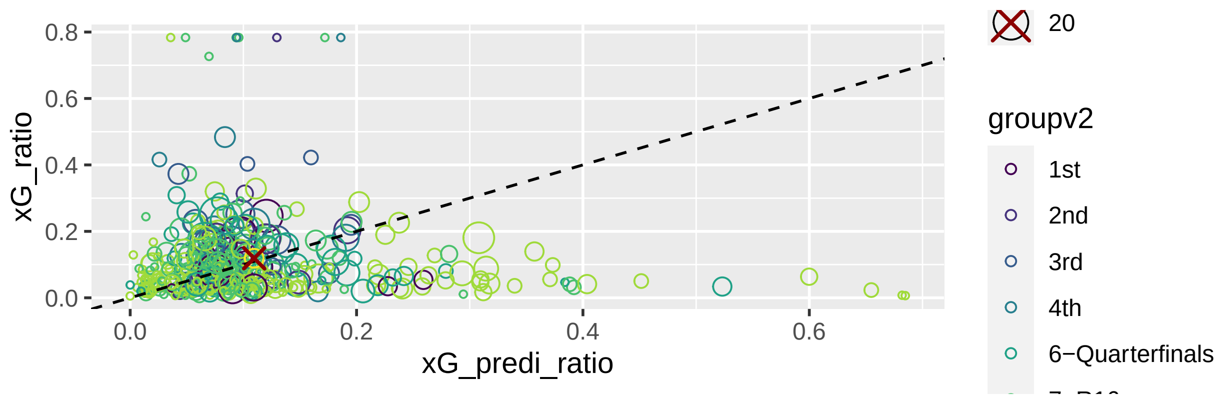


Figure 11: StatsBomb xG ratio on our xG ratio based on bestmod for players

Figure ?? shows a visualization of these results.

```
#pays = c("France", "England", "Spain", "Switzerland") #Countries we wish to observe
pays <- melt(df3$team) %>% distinct() %>% as.list() %>% unlist() %>% sort() %>% as.list() #select all t

#view(pays)

df3_pays <- df3[df3$team %in% pays,]
#view(df3_pays)

for (i in 0:length(pays))
  df3_pays[nrow(df3_pays) + 1, ] <- list(paste("",pays[i]), paste("", pays[i]),
                                        sum(df3_pays$total_shots[df3_pays$team == pays[i]]),
                                        sum(df3_pays$shots_completion_ratio[df3_pays$team == pays[i]]),
                                        sum(df3_pays$total_goals[df3_pays$team == pays[i]]),
                                        sum(df3_pays$xG_ratio[df3_pays$team == pays[i]]),
                                        sum(df3_pays$xG[df3_pays$team == pays[i]]),
                                        sum(df3_pays$xG_predi[df3_pays$team == pays[i]]),
                                        sum(df3_pays$xG_predi_ratio[df3_pays$team == pays[i]]))

df3_pays <- subset(df3_pays, select = -c(player))

#view(df3_pays)
n <- length(pays)

df3tail <- df3_pays[(nrow(df3_pays)-(n - 1)):nrow(df3_pays),]  %>% mutate(team = gsub(" ","",team),
                                                                   shots_completion_ratio = total_goals
                                                                   total_shots = total_shots,
                                                                   total_goals = total_goals,
                                                                   xG = xG,
                                                                   xG_ratio = xG/total_shots,
                                                                   xG_predi = xG_predi,
                                                                   xG_predi_ratio = xG_predi/total_sho
                                                                   .keep = "unused")

df3tail$groupv2 <- ifelse(df3tail$team %in% pays_gs, "8-GS",
                          ifelse(df3tail$team %in% pays_r16, "7-R16",
```

```r
                                                ifelse(df3tail$team %in% pays_quarterfinals, "6-Quarterfinals",
                                                       ifelse(df3tail$team == "Australia", "4th",
                                                              ifelse(df3tail$team == "Sweden", "3rd",
                                                                     ifelse(df3tail$team == "England", "2nd",
                                                                            ifelse(df3tail$team=="Spain","1st","NA

mean_shots_nb_pays = mean(df3tail$total_shots)
mean_goals_nb_pays = mean(df3tail$total_goals)
mean_shots_completion_ratio_pays = mean_goals_nb_pays/mean_shots_nb_pays
mean_shots_xG_pays = mean(df3tail$xG)
mean_shots_xG_ratio_pays = mean_shots_xG_pays/mean_shots_nb_pays
mean_shots_xG_predi_pays = mean(df3tail$xG_predi)
mean_shots_xG_predi_ratio_pays = mean_shots_xG_predi_pays/mean_shots_nb_pays

df3tail[nrow(df3tail) + 1,] <- list("Mean Country", mean_shots_nb_pays, mean_shots_completion_ratio_pay


#view(df3tail)
```

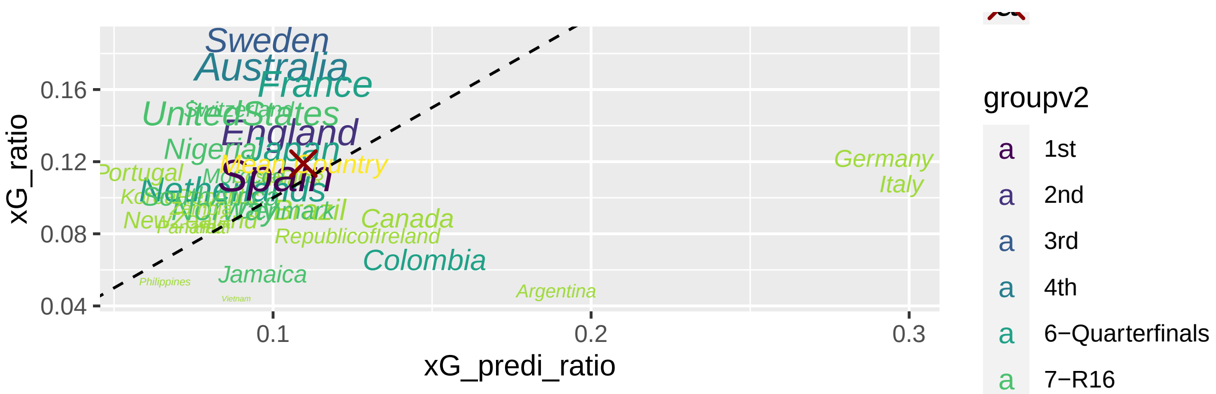


Figure 12: StatsBomb xG ratio on our xG ratio based on bestmod for teams