Home Advantage and Attendance Impact on Match Outcomes: An Analysis of Corinthians in the Brazilian League

Lucas Avila

2025-01-22

Table of contents

Objectives

This Quarto book has four objectives:

1. To practice and showcase my English skills.
2. To demonstrate some of my data analysis skills.
3. To learn how to use a Quarto book.
4. And last, but not least, to have a question cleared up: Does the crowd at a football game influence the outcome?

And it's based on those points that I will write from now on.

1 Introduction

It is well known that a team hosting a game has an advantage. This belief is primarily attributed to two factors:

A larger crowd "intimidates" the opposing team and the referee. It motivates the home team. Although this is widely accepted by most people, I aim to approach this study purely from a data-driven perspective to statistically determine whether it is true.

Initially, my plan is to use Categorical Data Analysis to examine whether there is an association between the rows (win/draw/defeat) and the columns (host/visitor). However, the assumption of independent data might be false, so I will have to explore more advanced models. To achieve this, I will collect data from the internet, as described in the next chapter.

2 Collecting data

For this I am going to use some of my webscraping abilities. But what data are we interessed in? And the answear is: everything that I can get about the games played by Corinthians, the team that I am fan of.

To do so I found some websites to help me, the links are the followings:

https://www.espn.com.br/futebol/time/estatisticas/_/id/874/liga/BRA.1/vista/rendimento

https://fbref.com/en/squads/bf4acd28/2023/all_comps/Corinthians-Stats-All-Competitions

And the most important one:

https://www.ogol.com.br/team_competition.php?id_comp=51&id_epoca=152&op=matches&id_equipa=2234&id_jogo=0

2.1 How to do the webscrap

The packages that are going to be used in this first step are the followings

```
library(rvest)
library(httr)
```

This other one is just for data manipulation

```
library(tidyverse)
```

Unfortunetly the site https://www.ogol.com.br/team_competition.php?id_comp=51&id_epoca=152&op=matches&id_equipa=2234&id_jogo=0 does not allow for web-scraping (as you can see by accescing the site https://www.ogol.com.br/robots.txt)

2.2 Second site attempt

The website https://www.meutimao.com.br/resultados-dos-jogos-do-corinthians/ is one specific for Corinthians, and allows web scrap.

NOTE: THE SITE ACTUALLY DOESN'T ALLOW WEB SCRAP. AFTER COLLECTING ALL THE DATA I GOT BLOCKED FROM THE WEBSITE AND GOT A MESAGE SAYING THAT IT IS NOT ALLOWED. PLEASE DO NOT REPRODUCE ANY OF THE FOLLOWING CODES.

As so, we are going to go for the website, get the content as text, read the parts that interest us the most (that is the argument of the function html_nodes, and I get the names by using the Google Chrome Extension "SelectorGadget").

```
url <- paste("https://fbref.com/pt/equipes/bf4acd28/2023/partidas/c24/schedule/Corinthians-Resulta
response <- httr::GET(url)
content <- httr::content(response, as = "text")
```

```
page <- read_html(content)
results <- page %>%
  html_nodes(".right , .left , .center") %>%
  html_text()
# head(results, 30)
df <- as.data.frame(matrix(results[19:length(results)], ncol = 18, byrow = TRUE))
colnames(df) <- results[1:18]
head(df)
```

This data frame is supposed to be (and luckly is) the following table that we find at the website

Data	Horário	Rodada	Dia	Local	Resultado	GP	GC	Oponente	xG	xGA	Posse	Público	Capitão	Formação	Árbitro	Relatório da Partida	Notas
2023-04-16	16:00	Rodada da semana 1	dom	Em casa	V	2	1	Cruzeiro	1.5	0.5	40	41.304	Cássio	4-2-3-1	Anderson Daronco	Relatório da Partida	
2023-04-23	19:00	Rodada da semana 2	dom	Visitante	D	1	3	Goiás	0.5	1.2	48	8.508	Cássio	4-2-3-1	Bruno Arleu de Araujo	Relatório da Partida	
2023-04-29	18:30	Rodada da semana 3	sáb	Visitante	D	1	2	Palmeiras	0.5	1.4	46	41.457	Cássio	4-2-3-1	Wilton Sampaio	Relatório da Partida	
2023-05-08	20:00	Rodada da semana 4	seg	Em casa	E	1	1	Fortaleza	1.5	0.8	49	36.512	Cássio	4-2-3-1	Ramon Abatti Abel	Relatório da Partida	
2023-05-11	19:30	Rodada da semana 5	qui	Visitante	D	0	3	Botafogo (RJ)	0.6	2.6	47	22.388	Cássio	4-1-4-1	Anderson Daronco	Relatório da Partida	
2023-05-14	16:00	Rodada da semana 6	dom	Em casa	E	1	1	São Paulo	0.9	0.9	34	41.118	Cássio	4-1-4-1	Bruno Arleu de Araujo	Relatório da Partida	

Figure 2.1: Resume of Corinthians in Brazilian Championship 2023

As we can see the webscrapping was, for it's mission, perfect.

But now we ought to solve one more problem. This data was collected for the year 2023, so it is a small sample. To solve this, we are going to take the the stats also for the previous years.

One way to do that is notice that we were at https://fbref.com/pt/equipes/bf4acd28/2023/partidas/c24/schedule/Corinthians-Resultados-e-Calendarios-Serie-A, to take the stats from 2023, and it is reasonable to think (or at least worth the shot) that to get the data from 2022 we need only to change the /2023/ to /2022/

Luckly it's true! If we change the link to https://fbref.com/pt/equipes/bf4acd28/2022/partidas/c24/schedule/Corinthians-Resultados-e-Calendarios-Serie-A we get what we

want, and for even older stats that does the trick.

Unfortunetly it only works until 2014, so that is the maximum amount of data that we can get (which I am going to consider, without any statistical support, enough).

So let's get the work done!

```
resultsFinal <- list(0)
listdf <- list(0)
for(x in 2014:2023){
url <- paste("https://fbref.com/pt/equipes/bf4acd28/",x,"/partidas/c24/schedule/Corinthians-Resulta
response <- httr::GET(url)
content <- httr::content(response, as = "text")
page <- read_html(content)
results <- page %>%
  html_nodes(".right , .left , .center") %>%
  html_text()
# head(results, 30)
resultsFinal[[x]] <- results[str_length(results) < 150]
listdf[[x]] <- as.data.frame(matrix(results[(which(results == "Notas")+1):length(results)], ncol = whicl
colnames(listdf[[x]]) <- results[1:which(results == "Notas")]
}
```

```
resultsFinal[[2014]] %>% head()
```

```
[1] "Data"       "Horário"    "Rodada"     "Dia"        "Local"      "Resultado"
```

```
listdf[[2014]] %>% head()
```

```
listdf2 <- list(0)
i <- 1
for(x in 2014:2023){
listdf2[[i]] <- as.data.frame(matrix(resultsFinal[[x]][(which(resultsFinal[[x]] == "Notas")+1):length(resu
colnames(listdf2[[i]]) <- resultsFinal[[x]][1:which(resultsFinal[[x]] == "Notas")]
i <- i + 1
}
```

Now let's verify if all the data has the same content

```
names <- sapply(listdf2, colnames)
isTRUE(unique(unlist(lapply(names, function(x) identical(names[[1]], x)))))
```

[1] FALSE

Unfortunatyle it doesn't. A quick view of the variable "names" show us why. Only in 2019 the columns "xG','xGA" were added. To make all the layers have the same content, lets remove this columns.

```
for(i in 1:10){
  if("xG" %in% colnames(listdf2[[i]])){
    listdf2[[i]] <- listdf2[[i]][c(1:9, 12:18)]
  }
}
```

```
listdf2[[9]] %>% head()
```

	Data	Horário	Rodada	Dia	Local	Resultado	GP	GC
1	2022-04-10	16:00	Rodada da semana 1	dom	Visitante	V	3	1
2	2022-04-16	19:00	Rodada da semana 2	sáb	Em casa	V	3	0
3	2022-04-23	19:00	Rodada da semana 3	sáb	Visitante	D	0	3
4	2022-05-01	16:00	Rodada da semana 4	dom	Em casa	V	1	0
5	2022-05-08	18:00	Rodada da semana 5	dom	Visitante	V	1	0
6	2022-05-14	19:00	Rodada da semana 6	sáb	Visitante	E	2	2

	Oponente	Posse	Público	Capitão	Formação	Árbitro
1	Botafogo (RJ)	44	36.898	Cássio	4-3-3	Wilton Sampaio
2	Avaí	54	30.497	Cássio	4-2-3-1	Bruno Arleu de Araujo
3	Palmeiras	56	23.973	Paulinho	4-3-3	Anderson Daronco
4	Fortaleza	55	37.018	Cássio	4-3-3	Savio Pereira
5	Bragantino	42	9.993	Cássio	4-2-3-1	Raphael Claus
6	Internacional	52	18.482	Cássio	4-3-1-2	Braulio da Silva Machado

	Relatório da Partida	Notas
1	Relatório da Partida	
2	Relatório da Partida	
3	Relatório da Partida	
4	Relatório da Partida	
5	Relatório da Partida	
6	Relatório da Partida	

```
names <- sapply(listdf2, colnames)
unique(apply(names, 2, function(x) all.equal(names[,1], x)))
```

```
[1] TRUE
```

Now all the layers have the exact same variables. To make the data manipulation easier, lets make one change. Access layers of a list is a difficult task, instead, lets create one variable for each of those dataframes.

```
varNames <- paste("br", 14:23, sep = "")
j <- 1
for(i in varNames){
assign(i, listdf2[[j]])
j <- j + 1
}
head(br14)
```

	Data	Horário	Rodada	Dia	Local	Resultado	GP	GC
1	2014-04-20	16:00	Rodada da semana 1	dom	Visitante	E	0	0
2	2014-04-27	16:00	Rodada da semana 2	dom	Em casa	V	2	0
3	2014-05-04	18:30	Rodada da semana 3	dom	Visitante	V	1	0
4	2014-05-11	16:00	Rodada da semana 4	dom	Visitante	E	1	1
5	2014-05-18	16:00	Rodada da semana 5	dom	Em casa	D	0	1
6	2014-05-21	22:00	Rodada da semana 6	qua	Em casa	E	1	1

	Oponente	Posse	Público	Capitão	Formação	Árbitro
1	Atlético Mineiro		8.724	4-3-2-1	Heber Roberto Lopes	
2	Flamengo		36.402	4-2-2-2	Leandro Pedro Vuaden	
3	Chapecoense		15.009	4-2-3-1	Wagner Reway	
4	São Paulo		14.000	4-2-2-2	Raphael Claus	
5	Figueirense		36.123	4-4-2	Jailson Macêdo Freitas	
6	Atl Paranaense		13.137	4-2-3-1	Marcelo de Lima Henrique	

	Relatório da Partida	Notas
1	Relatório da Partida	

2 Relatório da Partida

3 Relatório da Partida

4 Relatório da Partida

5 Relatório da Partida

6 Relatório da Partida

One important thing to do is change all the acute accent and special characters to regular letters, so R doesn't complain about that.

```
for(i in varNames){
assign(i, `colnames<-`(get(i), iconv(colnames(get(i)), to = "ASCII//TRANSLIT")))
}
```

There is still one more thing to do. The variable "Publico" (Public) seems to be NA in the years of the pandemic, which makes sense. To solve this, we can define as 0, which is a fair number to be imputed.

```
br20$Publico <- 0
br21[br21$Publico == "", ]$Publico <- "0"
```

Another thing that is bothering me about this column is the fact that the mile separator is ".". For this reason, if I apply the function as.numeric to the column, the data will be incorrectly changed. So first lets remove all the dots from the data frames.

```
for (i in varNames) {
  temp <- get(i)
  temp$Publico <- gsub("\\.", "", temp$Publico)
  assign(i, temp)
}
```

Great! Now there is only one more thing that I would like to change. But first, I am gonna gather all this information in one variable:

```
dfdfdf2 <- br14
```

```
for(i in varNames[2:10]){
  dfdfdf2 <- rbind(dfdfdf2, get(i))
}


dfdfdf2[8:10,]
```

	Data	Horario	Rodada	Dia	Local	Resultado	GP	GC
8	2014-05-28	22:00	Rodada da semana 8	qua	Em casa	V	1	0
9	2014-06-01	16:00	Rodada da semana 9	dom	Em casa	E	1	1
10	2014-07-17	19:30	Rodada da semana 10	qui	Em casa	V	2	1

	Oponente	Posse	Publico	Capitao	Formacao
8	Cruzeiro		17696		4-2-3-1
9	Botafogo (RJ)		37119		4-2-3-1
10	Internacional		32644		4-1-2-1-2

	Arbitro	Relatorio da Partida	Notas
8	Dewson Fernando Freitas da Silva	Relatório da Partida	
9	Leandro Pedro Vuaden	Relatório da Partida	
10	Wagner Reway	Relatório da Partida	

weird chacrter

And, as we can see in the 10th observation, for example, there is this weird symbol showing up on column “formacao”. I will remove this

```
weird_character <- str_sub(dfdfdf2[10,13],-1,-1)
weird_character
```

[1] ” ”

```
dfdfdf2$Formacao <- gsub(weird_character, ””, dfdfdf2$Formacao)
```

Once we have all our data in one variable, lets change the columns names to English.

```
data <- dfdfdf2
colnames(data) <- c(”date”, ”time”, ”round”, ”day”, ”venue”, ”result”, ”GS”, ”GC”, ”Opponent”, ”pos
```

Lets also translate some of the information

```
data$venue <- ifelse(data$venue == ”Em casa”, ”Home”, ”Away”)
```

Where “GS” stands for goals scored and “GC” goals conceded.

Lets see if there are more mistakes on our data. First, we know by a fact that every team has to play half the games away, and half the games at home, to verify we shall use

```
data$attendance <- as.numeric(data$attendance)
data$date <- as.Date(data$date)

data %>%
  mutate(Year = year(date)) %>%
  group_by(Year) %>%
```

```
  summarize(
    Away = sum(venue == "Away"),
    Home = sum(venue == "Home")
  )
```

```
# A tibble: 10 x 3
    Year  Away  Home
   <dbl> <int> <int>
 1  2014    19    19
 2  2015    19    19
 3  2016    19    19
 4  2017    19    19
 5  2018    19    19
 6  2019    19    19
 7  2020    14    13
 8  2021    24    25
 9  2022    19    19
10  2023    19    19
```

Woah! Some observations are not equal on both columns. Why is that?. As we can see, we have differences only on 2020 and 2021. The reason for that is that the year of the game isn't equal to the year of the championship, since it was postponed in consequence of the pandemic. Hence, we have to create a column for year of tournament 'manually'.

```
data$year <- rep(2014:2023, each = 38)
```

Now we can get the right result

```r
data %>%
  group_by(year) %>%
  summarize(
    Away = sum(venue == "Away"),
    Home = sum(venue == "Home")
  )
```

```
# A tibble: 10 x 3
    year  Away  Home
   <int> <int> <int>
 1  2014    19    19
 2  2015    19    19
 3  2016    19    19
 4  2017    19    19
 5  2018    19    19
 6  2019    19    19
 7  2020    19    19
 8  2021    19    19
 9  2022    19    19
10  2023    19    19
```

Perfect! Now, lets see if the formation sum up to 10 (as the goalkeeper isn't there).

```r
df_test <- data.frame(formation = data$formation)
df_test <- df_test %>%
  separate(formation, into = paste0("Coluna", 1:10), sep = "-", fill = "right")
df_test <- apply(df_test, 2, as.numeric)
```

```
apply(df_test, 1, sum, na.rm = TRUE) %>% unique()
```

```
[1] 10  0
```

Great!

3 EDA

```
library(tidyverse)
library(ggplot2)
```

```
data <- data[!(is.na(data$attendance) & data$year == 2023),]
data[is.na(data$attendance), ]
```

	date	time	round	day	venue	result	GS	GC
336	2022-10-22	19:00	Rodada da semana 33	sáb	Away	V	1	0
339	2022-11-02	21:30	Rodada da semana 35	qua	Away	V	2	1
341	2022-11-09	19:00	Rodada da semana 37	qua	Away	E	2	2
342	2022-11-13	16:00	Rodada da semana 38	dom	Home	D	0	1

	Opponent	possession	attendance	captain	formation
336	Santos	47	NA	Cássio	4-3-3
339	Flamengo	48	NA	Cássio	4-2-3-1
341	Coritiba	48	NA	Cássio	4-1-4-1
342	Atlético Mineiro	58	NA	Cássio	4-1-4-1

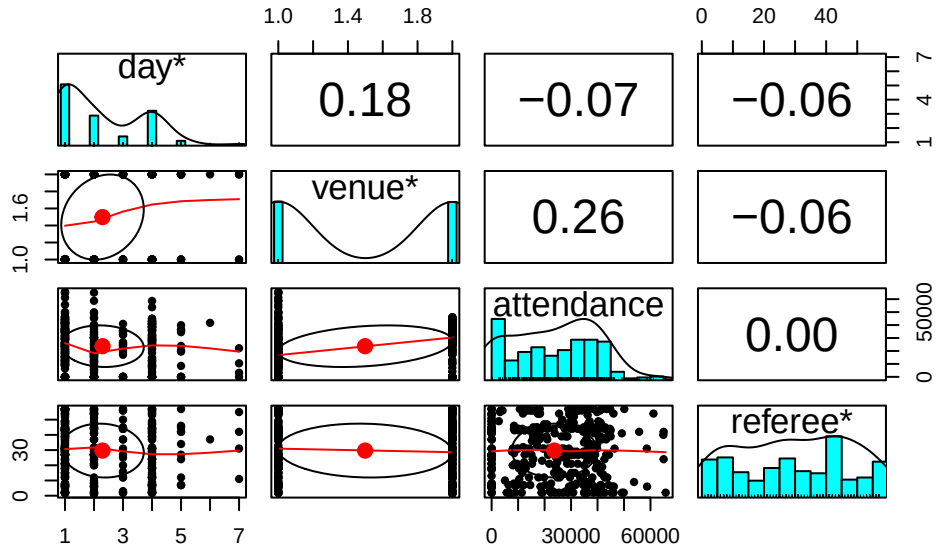
	referee	game report	notes	year
336	Flavio Rodrigues De Souza	Relatório da Partida		2022
339	Ramon Abatti Abel	Relatório da Partida		2022

341	Marcelo de Lima Henrique	Relatório da Partida	2022
342	Braulio da Silva Machado	Relatório da Partida	2022

As we can see, there are still some missing data (NA) remaining. Apparently, the scraped website did not compute these values. Let's fill them in manually, except for the last one, which I couldn't find anywhere. For this case, I will input the mean attendance of the home team for that year.

```
data[is.na(data$attendance), ]$attendance <- c(12872, 65000, 39852, mean(data[data$year == 2022 &
```

```
psych::pairs.panels(data[c(4,5,11,14)])
```



We can see that multicollinearity doesn't seen to be a problem for future analysis, lets confirm.

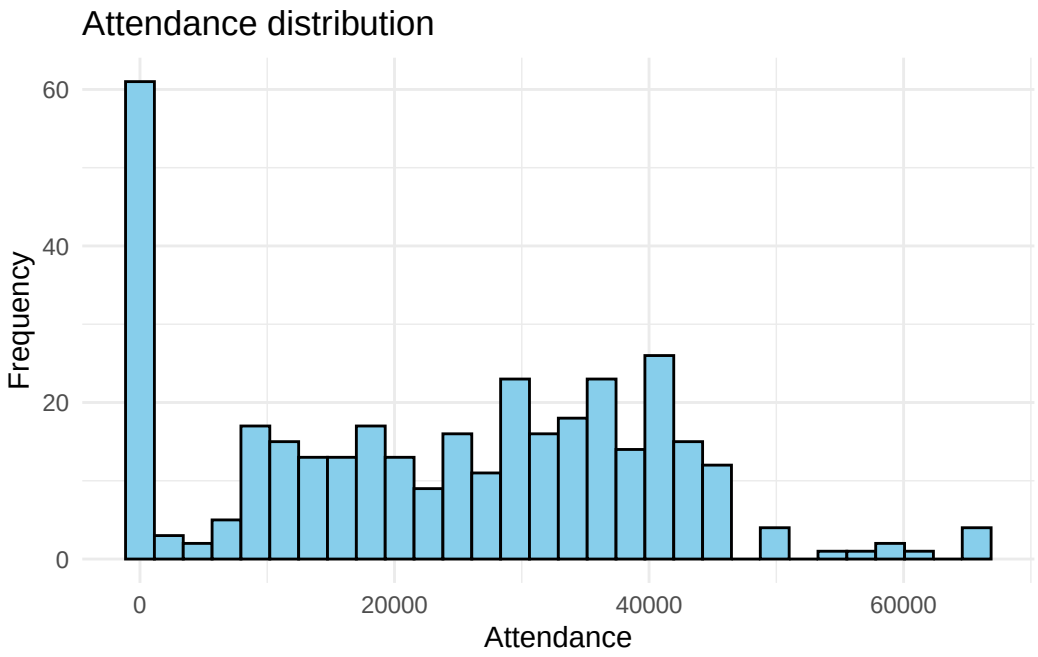
```
data$result <- as.factor(data$result)
linear_mod <- lm(as.numeric(result) ~ venue + attendance + referee + day, data = data)
car::vif(linear_mod)
```

	GVIF	Df	GVIF^(1/(2*Df))
venue	1.286197	1	1.134106
attendance	1.389097	1	1.178600
referee	5.502292	56	1.015341
day	4.366589	6	1.130694

VIF values also relatively low, great.

```
data %>%
  filter(result != "") %>%
  ggplot(aes(x = attendance)) +
  geom_histogram(fill = "skyblue", color = "black") +
  labs(title = "Attendance distribution",
       x = "Attendance",
       y = "Frequency") +
  theme_minimal()
```

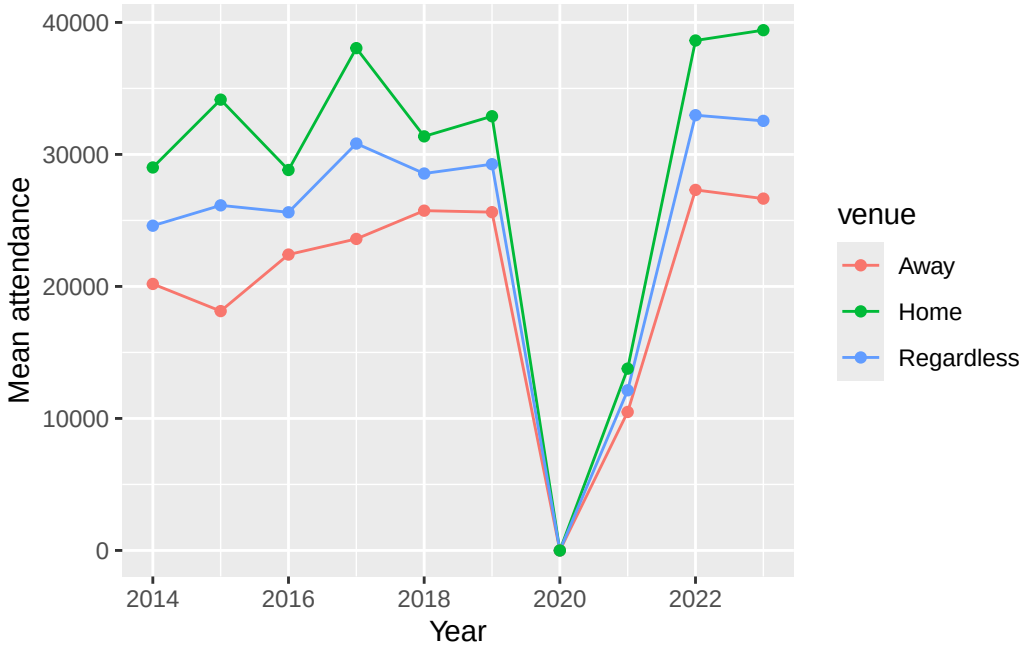
‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.



Doesn't seen to follow any common distribution, maybe it's a mixture of distributions? It's hard to say.

```
data_plot <- data %>%
  filter(result != "") %>%
  group_by(year) %>%
  reframe(mean_attendance = mean(attendance),
          venue = "Regardless") %>%
  rbind(data %>%
  filter(result != "") %>%
  group_by(year, venue) %>%
  reframe(mean_attendance = mean(attendance)))

data_plot %>%
  ggplot(aes(x = year, y = mean_attendance, col = venue)) +
```

```
geom_line() +
geom_point() +
xlab("Year") +
ylab("Mean attendance")
```



We can see that the mean public has been growing, but doesn't look like a big differences through time

```
contingency_table <- table(data$result, data$Opponent)
contingency_table
```

	América (MG)	Atl Goianiense	Atl Paranaense	Atlético Mineiro	Avaí	Bahia
D	2	2	4	8	0	4

E	3	3	7	4	4	2
V	4	3	8	6	4	6

	Botafogo (RJ)	Bragantino	Ceará	Chapecoense	Coritiba	Criciúma	Cruzeiro	CSA
D	6	3	4	1	0	0	3	1
E	2	2	2	4	6	1	3	0
V	7	2	4	9	6	1	7	1

	Cuiabá	Figueirense	Flamengo	Fluminense	Fortaleza	Goiás	Grêmio	Internacional
D	1	2	10	9	2	1	5	2
E	1	2	3	3	3	3	8	8
V	3	2	6	7	4	7	3	6

	Joinville	Juventude	Palmeiras	Paraná	Ponte Preta	Santa Cruz	Santos
D	0	1	9	0	2	0	6
E	0	2	5	0	1	0	5
V	2	1	5	2	3	2	8

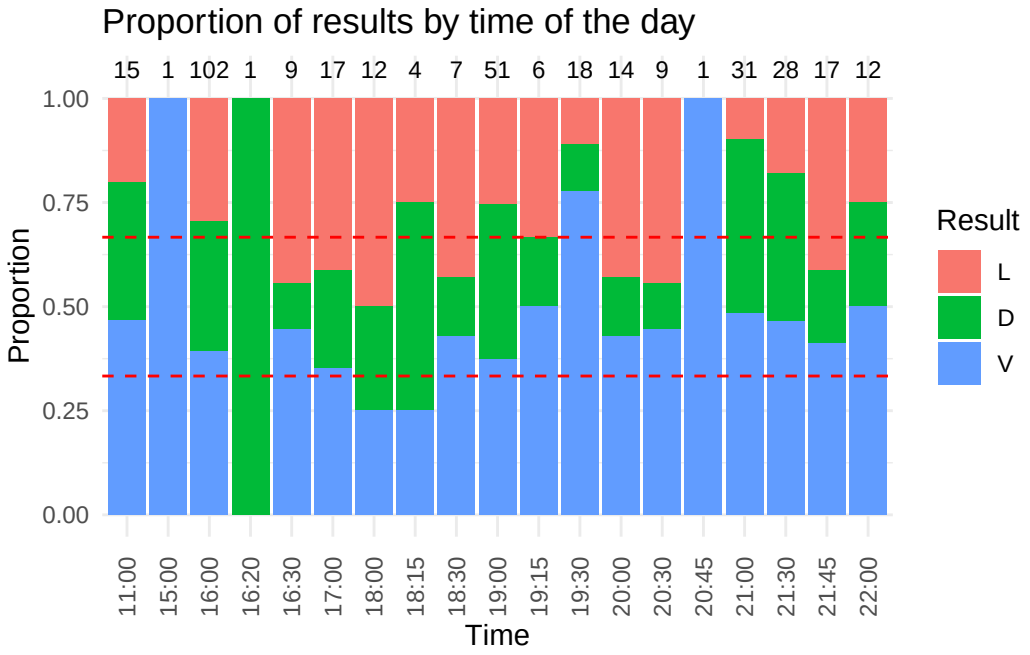
	São Paulo	Sport Recife	Vasco da Gama	Vitória
D	5	4	0	2
E	9	1	3	3
V	5	9	7	3

There seem to be different performances against different teams. For example, we seem to perform well against Vasco da Gama, have an equal record against São Paulo, and perform quite poorly against Flamengo. But as I just noticed, the data collected is entirely in Portuguese. For that reason, The next chunk is dedicated to translate all the

relevant information to Portuguese. The D E V schema (that was representing defeat, draw and victory respectively) will be represented as L D V (lose, draw, victory).

```
data[data$referee == "Wagner do Nascimento Magalhaes", ]$referee <- "Wagner do Nascimento Maga
data[data$referee == "Pericles Bassols Pegado Cortez", ]$referee <- "Péricles Bassols Pegado Cortez"
data[data$referee == "Luiz Flavio De Oliveira", ]$referee <- "Luiz Flávio de Oliveira"
data$referee <- factor(data$referee)
data$time <- as.factor(data$time)

data <- data %>%
  mutate(day = recode(day,
  "seg" = "Mon",
  "ter" = "Tue",
  "qua" = "Wed",
  "qui" = "Thu",
  "sex" = "Fri",
  "sáb" = "Sat",
  "dom" = "Sun"))

data$day <- factor(data$day, levels = c("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"))

data <- data %>%
  mutate(result = recode(result,
  "V" = "V",
  "D" = "L",
  "E" = "D"))

data$result <- factor(data$result, levels = c("L", "D", "V"))
```

```r
data_totals <- data %>%
  group_by(time) %>%
  summarize(total = n())

ggplot(data, aes(x = time, fill = result)) +
  geom_bar(position = "fill") +
  geom_text(
    data = data_totals,
    aes(x = time, y = 1.05, label = paste(total)),
    vjust = 0,
    size = 3,
    inherit.aes = FALSE
  ) +
  labs(x = "Time", y = "Proportion", fill = "Result") +
  ggtitle("Proportion of results by time of the day") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5)
  ) +
  geom_hline(yintercept = c(1/3, 2/3), linetype = "dashed", color = "red")
```



The time when the match begins doesn't seem to make much difference, but at night (7:15 PM and later), the results appear to be better, though the impact is not significant. We can perform a statistical test to verify if this impression is accurate. But first, let's categorize the data into more easily understandable groups.

```
data_time <- data
data_time$time <- gsub(":", "", data$time)
data_time$time <- gsub(" ", "", data_time$time)
data_time$time <- as.numeric(data_time$time)
unique(as.numeric(data_time$time)) %>% sort()
```

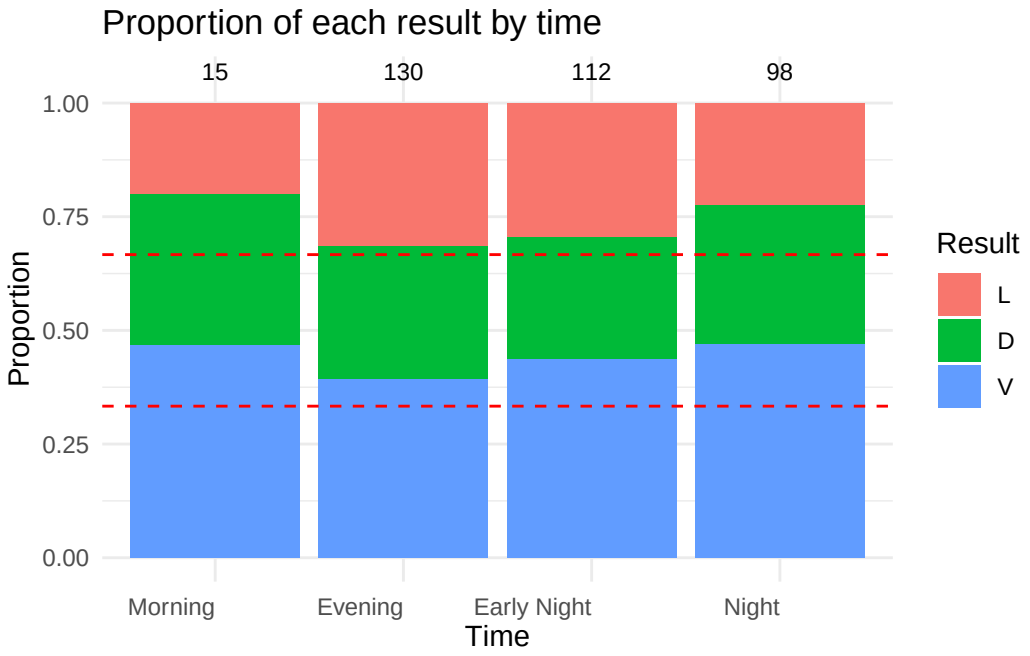
```
 [1] 1100 1500 1600 1620 1630 1700 1800 1815 1830 1900 1915 1930 2000 2030 2045
[16] 2100 2130 2145 2200
```

```
data_time$time <- cut(
  data_time$time,
  breaks = c(1059, 1459, 1759, 2001, 2201),
  labels = c("Morning", "Evening", "Early Night", "Night"))
```

I chose those timings because: 11:00 is the only game time before lunch (usually between 11:00 and 14:00). Then evening is considered between 15:00 and 18:00 (there is still some sunlight). Early night is between 18:01 and 20:00 because it's still a time that one can go watch the game and be home not too late (and by that I mean that one will not miss the game because they have to sleep, mainly a issue if they have to work next day). Late night is after 20:01 because usually the game would end around 22:00 or latter, which can be a issue to someone that works early in the morning next day, or have to take the metro that closes at midnight. That consideration is important because the public may vary alongside the game time.

```
total_data <- data_time %>%
  group_by(time) %>%
  summarize(total = n())

ggplot(data_time, aes(x = time, fill = result)) +
  geom_bar(position = "fill")  +
  geom_text(
    data = total_data,
    aes(x = time, y = 1.05, label = paste(total)),
    vjust = 0,
    size = 3,
    inherit.aes = FALSE
```

```
) +
labs(x = "Time", y = "Proportion", fill = "Result") +
ggtitle("Proportion of each result by time") +
theme_minimal() +
theme(
  axis.text.x = element_text(angle = 0, hjust = 1, vjust = 0.5)
) +
geom_hline(yintercept = c(1/3, 2/3), linetype = "dashed", color = "red")
```




```
time_table <- table(data$time, data$result)
chisq.test(as.matrix(time_table))
```

Warning in chisq.test(as.matrix(time_table)): Aproximação do qui-quadrado pode estar incorreta

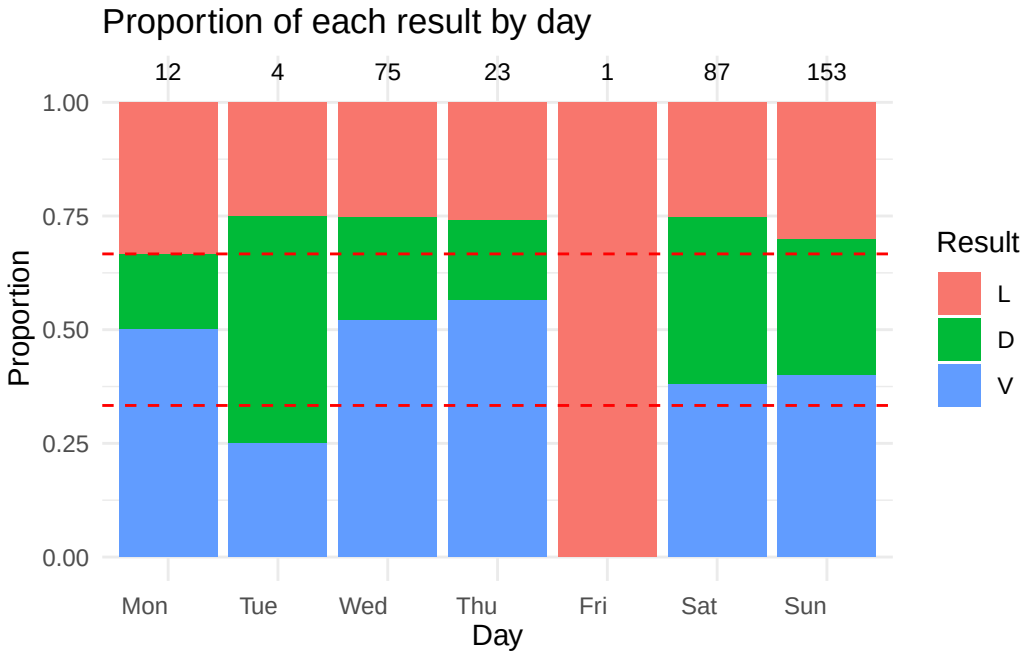
Pearson's Chi-squared test

data: as.matrix(time_table)
X-squared = 38.968, df = 36, p-value = 0.3377

It sens like the time when the match happens is independent from the result. Pretty interesting!

```
total_data <- data %>%
  group_by(day) %>%
  summarize(total = n())

ggplot(data, aes(x = day, fill = result)) +
  geom_bar(position = "fill") +
  geom_text(
    data = total_data,
    aes(x = day, y = 1.05, label = paste(total)),
    vjust = 0,
    size = 3,
    inherit.aes = FALSE
  ) +
  labs(x = "Day", y = "Proportion", fill = "Result") +
  ggtitle("Proportion of each result by day") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 0, hjust = 1, vjust = 0.5)
```

```
) +
geom_hline(yintercept = c(1/3, 2/3), linetype = "dashed", color = "red")
```




```
day_table <- table(data$day, data$result)
chisq.test(as.matrix(day_table))
```

Warning in chisq.test(as.matrix(day_table)): Aproximação do qui-quadrado pode estar incorreta

Pearson's Chi-squared test

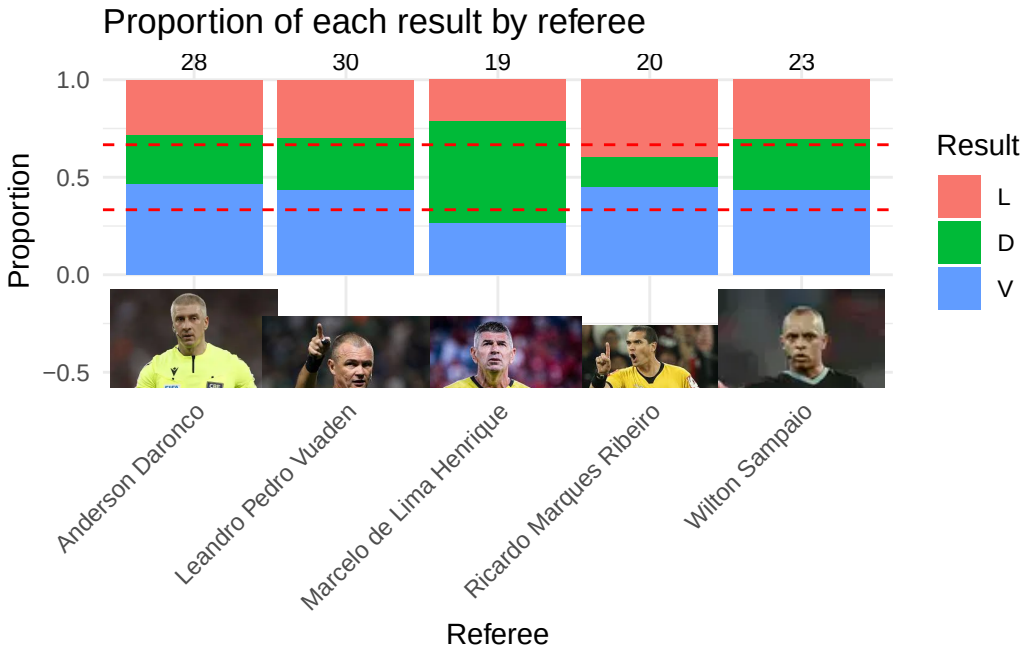
data: as.matrix(day_table)
X-squared = 12.303, df = 12, p-value = 0.4217

The day of the week doesn't seen to impact the result of the match! Pretty surprising given the graph.

```
library(ggimage)
```

```
total_data <- data %>%
  group_by(referee) %>%
  summarize(total = n()) %>%
  filter(total > quantile(total, 1-5/54)) %>%
  mutate(imagem = paste("'", referee, "'", sep = ""))

referee_graph <- data %>%
  filter(referee %in% total_data$referee) %>%
  ggplot(aes(x = referee, fill = result)) +
  geom_bar(position = "fill") +
  geom_text(
    data = total_data,
    aes(x = referee, y = 1.05, label = paste(total)),
    vjust = 0,
    size = 3,
    inherit.aes = FALSE
  ) +
  labs(x = "Referee", y = "Proportion", fill = "Result") +
  ggtitle("Proportion of each result by referee") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45,
                               hjust = 1#,
```

```
                        #vjust = 0.5
                        )
  ) +
  geom_hline(yintercept = c(1/3, 2/3), linetype = "dashed", color = "red")

data2 <- data %>%
  filter(referee %in% total_data$referee) %>%
  mutate(imagem = paste("Referee_image/", referee, ".webp", sep = ""))

final_graph <- referee_graph +
  geom_image(data = data2, aes(x = referee, y = -.5, image = imagem), size = .5) +
  theme(axis.text.x = element_text())

final_graph
```



Lets make the figures proportional

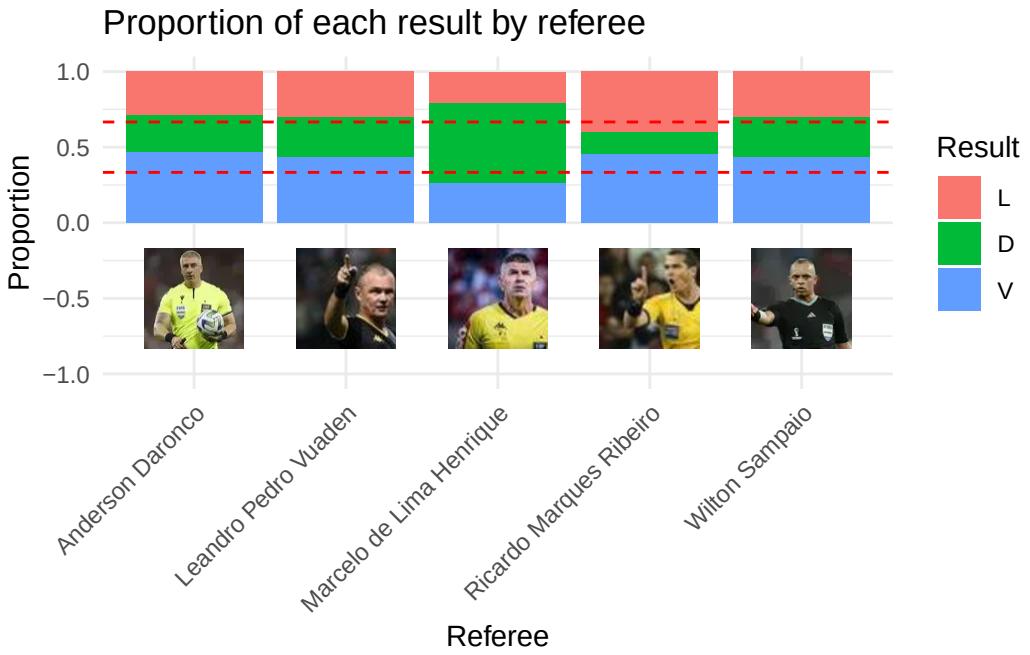
```
library(magick)

dir.create("Referee_img_stand")

images <- list.files("C:/Users/lucas/Documents/Estudo Importancia Torcida/Referee_image", full.nam
for (img in images) {
  image <- image_read(img)

  standardized_image <- image_resize(image, "100x100") %>%
                        image_crop("100x100+0+0")

  image_write(standardized_image,
              paste("Referee_img_stand/", sub(".*/", "", img), sep = "") )
}

images <- list.files("C:/Users/lucas/Documents/Estudo Importancia Torcida/Referee_img_stand", ful

for (img in images) {
  image <- image_read(img)



  adjusted_image <- image_trim(image) %>%
                    image_resize("100x100")

  image_write(adjusted_image,
              paste("Referee_img_stand/", sub(".*/", "", img), sep = "") )
```

```
}



images <- list.files("C:/Users/lucas/Documents/Estudo Importancia Torcida/Referee_img_stand", ful

for (img in images) {
  image <- image_read(img)

  width <- image_info(image)$width
  height <- image_info(image)$height

  if (width != height) {
    if (width > height) {
      crop_area <- paste0(height, "x", height)
    } else {
      crop_area <- paste0(width, "x", width)
    }
    image <- image_crop(image, crop_area)
  }

  standardized_image <- image_resize(image, "100x100")

  image_write(standardized_image, img)
}
```

```r
total_data <- data %>%
  group_by(referee) %>%
  summarize(total = n()) %>%
  filter(total > quantile(total, 1-5/54)) %>%
  mutate(imagem = paste("'", referee, "'", sep = ""))

referee_graph <- data %>%
  filter(referee %in% total_data$referee) %>%
  ggplot(aes(x = referee, fill = result)) +
  geom_bar(position = "fill") +
  geom_text(
    data = total_data,
    aes(x = referee, y = 1.05, label = paste(total)),
    vjust = 0,
    size = 3,
    inherit.aes = FALSE
  ) +
  labs(x = "Referee", y = "Proportion", fill = "Result") +
  ggtitle("Proportion of each result by referee") +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45,
                               hjust = 1#,
                               #vjust = 0.5
                               )
  ) +
  geom_hline(yintercept = c(1/3, 2/3), linetype = "dashed", color = "red")
```

```r
dados2 <- data %>%
  filter(referee %in% total_data$referee) %>%
  mutate(imagem = paste("Referee_img_stand/", referee, ".webp", sep = ""))

final_graph <- referee_graph +
  geom_image(data = dados2, aes(x = referee, y = -.5, image = imagem), size = 0.3) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ylim(-1, 1) # Ajustar o limite do eixo Y se necessário


final_graph
```

Warning: Removed 5 rows containing missing values or values outside the scale range (‘geom_text()‘).



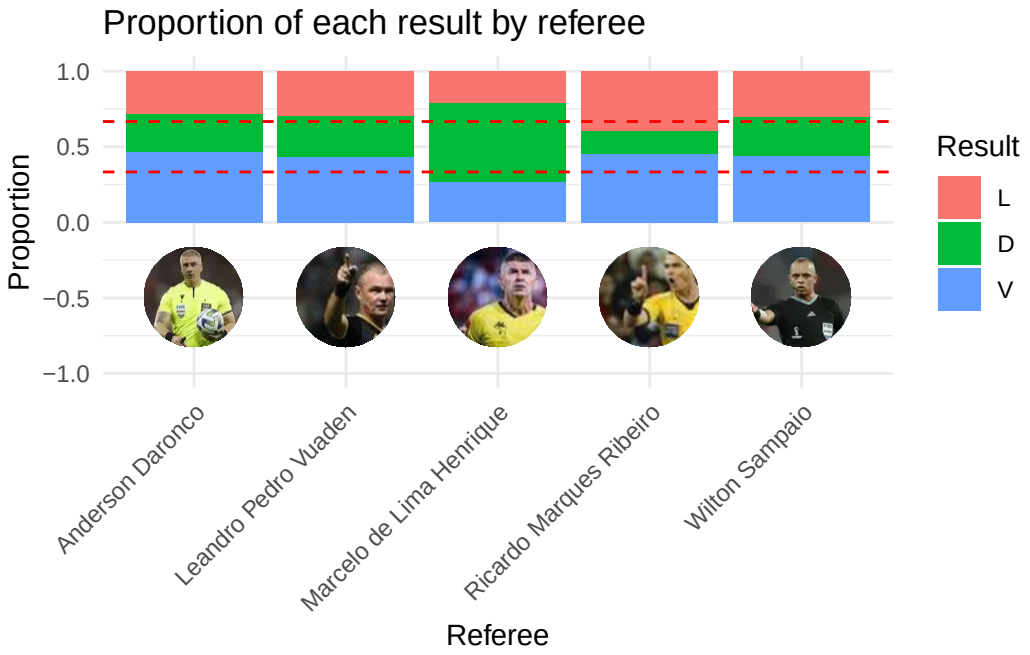
Making this graph but with circles (purely aesthetical step)

```
#devtools::install_github("doehm/cropcircles")
library(cropcircles)

 data2 <- data %>%
   filter(referee %in% total_data$referee) %>%
   mutate(imagem = paste("Referee_img_std/", referee, ".webp", sep = ""))
   #mutate(circular = circle_crop(images))

data3 <- data.frame(referee = total_data$referee,
                    circular = circle_crop(images))

data3 <- data2 %>%
  left_join(data3, by = join_by(referee == referee))

data3 %>% head(1)
```

```
        date  time              round day venue result GS GC Opponent
1 2014-04-27 16:00  Rodada da semana 2 Sun  Home      V  2  0 Flamengo
  possession attendance captain formation              referee
1         NA      36402           4-2-2-2 Leandro Pedro Vuaden
           game report notes year                                imagem
1 Relatório da Partida       2014 Referee_img_std/Leandro Pedro Vuaden.webp
                                                                 circular
1 C:\\Users\\lucas\\AppData\\Local\\Temp\\Rtmp2RvHzU\\cropped23bc51b6198f.png
```

```
final_graph <-
  referee_graph +
  geom_image(data = data3, aes(x = referee, y = -.5, image = circular), size = 0.3) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ylim(-1, 1)

final_graph
```

Warning: Removed 5 rows containing missing values or values outside the scale range (‘geom_text()‘).




```
referee_table <- table(data$referee, data$result)
chisq.test(as.matrix(referee_table))
```

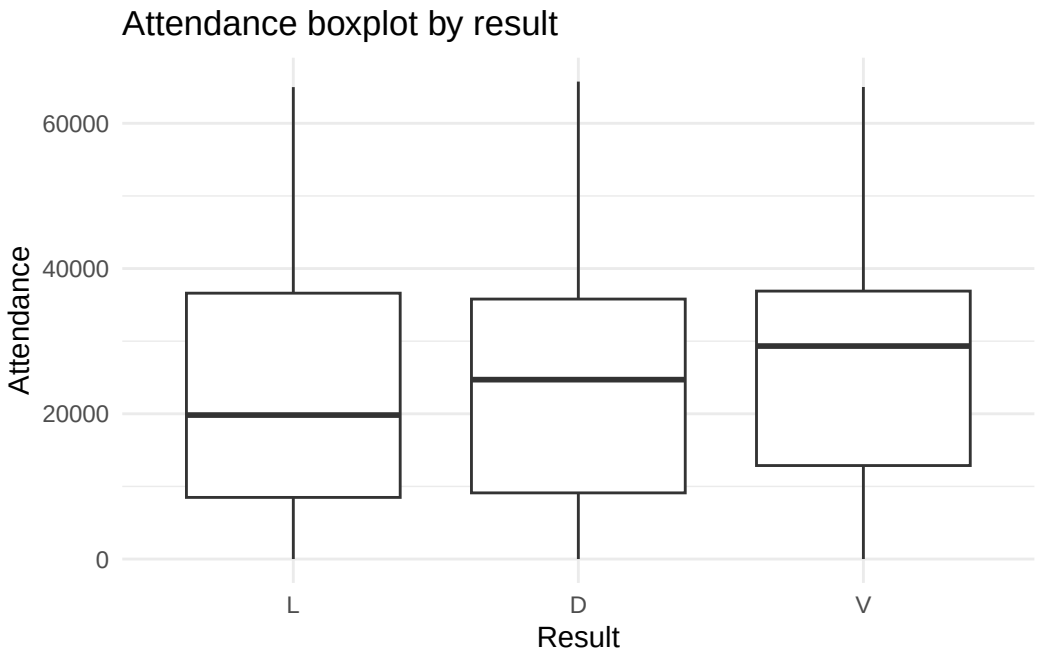
Warning in chisq.test(as.matrix(referee_table)): Aproximação do qui-quadrado pode estar incorreta

Pearson's Chi-squared test

data: as.matrix(referee_table)
X-squared = 103.52, df = 106, p-value = 0.55

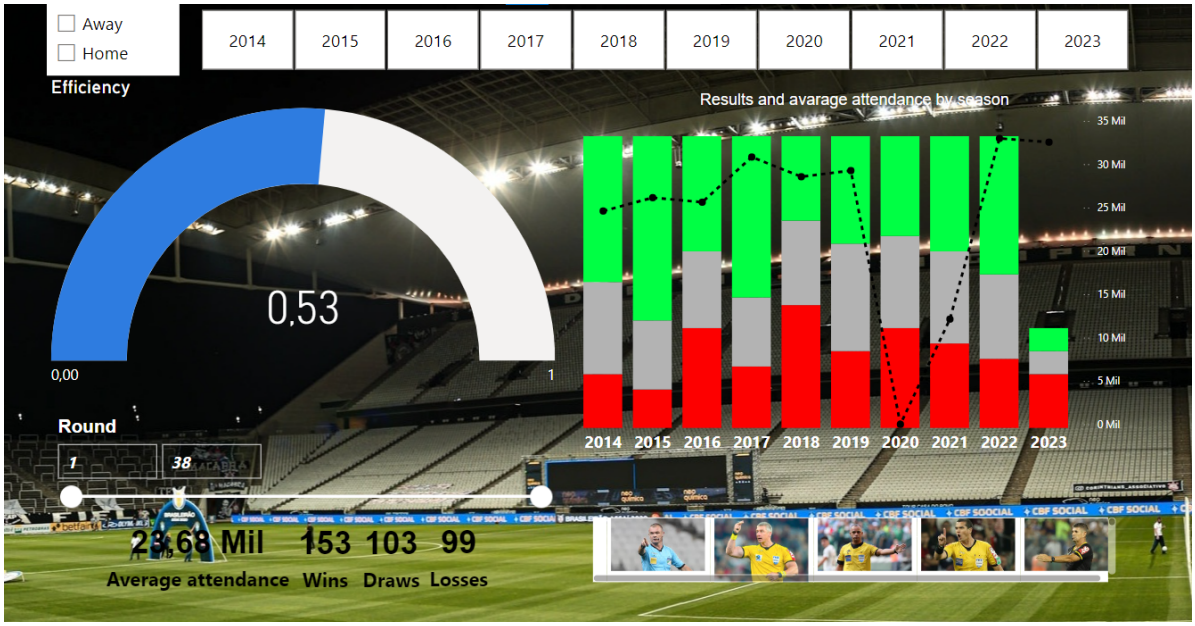
The referee doesn't seem to impact the outcome!

```
#box plots
ggplot(data, aes(x = result, y = attendance)) +
  geom_boxplot() +
  labs(x = "Result", y = "Attendance") +
  ggtitle("Attendance boxplot by result") +
  theme_minimal()
```



We can see that the median attendance increases as the results improve. However, it is important to note that this does not imply that people only support the team when the results are good, nor does it suggest that better results lead to higher attendance. Correlation does not imply causality.

Many more exploratory data analysis was made using Power BI. Unfortunately my account doesn't allow to share my work in a way the reader can explore the data, so I will just show a screenshot of the possibilities that it allow. Notice that you can easily filter by round, season, if it's a home or away game, by referee and etc.


Away
Home
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
Efficiency
0,53
0,00
1
Results and avarage attendance by season
35 Mil
30 Mil
25 Mil
20 Mil
15 Mil
10 Mil
5 Mil
0 Mil
2014 2015 2016 2017 2018 2019 2020 2021 2022 2023
Round
1
38
Mil
153 103 99
Average attendance Wins Draws Losses

4 Modeling

Basically, I want to test whether the outcome of the match—Victory, Defeat, or Draw—is independent of the location where the match was hosted—Home or Away. If they are independent, it would suggest that playing at home does not play any role in determining the winner. On the other hand, if they are not independent, it would indicate that playing at home provides some advantage to one of the teams.

To draw such conclusions, some statistical tests will be performed

4.1 Assuming sample independence

4.1.1 Chi-squared test

Pandemic period

```
pandemic <- data %>% filter(attendance == 0) %>%
  with(table(venue, result))


chisq.test(as.matrix(pandemic))
```

Pearson's Chi-squared test

data: as.matrix(pandemic)
X-squared = 0.25352, df = 2, p-value = 0.8809

```
non_pandemic <- data %>%
  filter(attendance != 0) %>%
  with(table(venue, result))


chisq.test(as.matrix(non_pandemic))
```

Pearson's Chi-squared test

data: as.matrix(non_pandemic)
X-squared = 52.442, df = 2, p-value = 4.096e-12

Such an interesting result! Note that when the attendance was zero, playing at home does not provide any advantage (p-value = 0.8809). However, when there is an audience, the evidence is overwhelming (p-value close to 10^{-12}), strongly indicating that the match location influences the result.

```
table_venue <- table(data$venue, data$result)
chisq.test(as.matrix(table_venue))
```

Pearson's Chi-squared test

data: as.matrix(table_venue)
X-squared = 46.489, df = 2, p-value = 8.037e-11

Really low p-values indicates that, in general, venue and result are not independent, therefore, where the match is placed does influence the result.

Another way to verify that is

```
library(gmodels)
t <- CrossTable(as.matrix(table_venue), resid = TRUE, sresid = TRUE,
                asresid = TRUE, format = "SPSS", prop.r = FALSE,
                prop.c = FALSE, prop.t = FALSE)
```

```
   Cell Contents
|-------------------------|
|                   Count |
| Chi-square contribution |
|                Residual |
|            Std Residual |
|            Adj Std Resid |
|-------------------------|

Total Observations in Table:  355

             |
             |         L  |         D  |         V  | Row Total |
```

Away	77	49	52	178
	15.081	0.135	7.963	
	27.361	-2.645	-24.715	
	3.883	-0.368	-2.822	
	6.476	-0.619	-5.298	
Home	22	54	101	177
	15.166	0.136	8.008	
	-27.361	2.645	24.715	
	-3.894	0.369	2.830	
	-6.476	0.619	5.298	
Column Total	99	103	153	355

Many absolute values of residuals above 1.96, meaning that the difference between the observed and the expected data is statistically significant, therefore, the outcome of the match and the local where it was hosted is not independent

4.1.2 Log-linear model

```
library(MASS)
df <- as.data.frame(table_venue)
```

```
reg <- loglm(Freq ~ Var1 + Var2, data = df)
reg
```

Call:
loglm(formula = Freq ~ Var1 + Var2, data = df)

Statistics:

	X^2	df	P(> X^2)
Likelihood Ratio	48.57404	2	2.833234e-11
Pearson	46.48864	2	8.037493e-11

P-value was really small, which means that Var1 and Var2 are not independent (it doesn't fit into the multiple independence model), therefore, once again, playing home gives someone the advantage.

```
tab <- as.data.frame(table(data[,(colnames(data) %in% c("venue", "result", "time"))]))
reg <- loglm(Freq ~ time + venue + result, data = tab)
summary(reg)
```

Formula:
Freq ~ time + venue + result
attr(,"variables")
list(Freq, time, venue, result)
attr(,"factors")

	time	venue	result
Freq	0	0	0
time	1	0	0

venue 0 1 0

result 0 0 1

attr(,”term.labels”)

[1] ”time” ”venue” ”result”

attr(,”order”)

[1] 1 1 1

attr(,”intercept”)

[1] 1

attr(,”response”)

[1] 1

attr(,”.Environment”)

<environment: R_GlobalEnv>

attr(,”predvars”)

list(Freq, time, venue, result)

attr(,”dataClasses”)

Freq time venue result

”numeric” ”factor” ”factor” ”factor”

Statistics:

X^2 df P(> X^2)

Likelihood Ratio 151.2864 92 9.771935e-05

Pearson 150.1888 92 1.230596e-04

```
extractAIC(reg)[2]
```

[1] 195.2864

Testing first order interactions

```
reg2 <- loglm(Freq ~ time + venue + result + time:venue, data = tab)
summary(reg2)
```

Formula:

Freq ~ time + venue + result + time:venue

attr(,"variables")

list(Freq, time, venue, result)

attr(,"factors")

	time	venue	result	time:venue
Freq	0	0	0	0
time	1	0	0	1
venue	0	1	0	1
result	0	0	1	0

attr(,"term.labels")

[1] "time" "venue" "result" "time:venue"

attr(,"order")

[1] 1 1 1 2

attr(,"intercept")

[1] 1

attr(,"response")

[1] 1

attr(,".Environment")

<environment: R_GlobalEnv>

attr(,"predvars")

list(Freq, time, venue, result)

attr(,"dataClasses")

Freq	time	venue	result

”numeric” ”factor” ”factor” ”factor”

Statistics:

	X^2	df	P(> X^2)
Likelihood Ratio	129.0302	74	7.884251e-05
Pearson	NaN	74	NaN

```
extractAIC(reg2)[2]
```

[1] 209.0302

```
reg3 <- loglm(Freq ~ time + venue + result + time:result, data = tab)
summary(reg3)
```

Formula:

Freq ~ time + venue + result + time:result

attr(,”variables”)

list(Freq, time, venue, result)

attr(,”factors”)

	time	venue	result	time:result
Freq	0	0	0	0
time	1	0	0	1
venue	0	1	0	0
result	0	0	1	1

attr(,”term.labels”)

[1] ”time” ”venue” ”result” ”time:result”

attr(,”order”)

[1] 1 1 1 2

attr(,"intercept")

[1] 1

attr(,"response")

[1] 1

attr(,".Environment")

<environment: R_GlobalEnv>

attr(,"predvars")

list(Freq, time, venue, result)

attr(,"dataClasses")

Freq	time	venue	result
"numeric"	"factor"	"factor"	"factor"

Statistics:

	X^2	df	P(> X^2)
Likelihood Ratio	110.1765	56	2.121326e-05
Pearson	NaN	56	NaN

```
extractAIC(reg3)[2]
```

[1] 226.1765

```
reg4 <- loglm(Freq ~ time + venue + result + venue:result, data = tab)
summary(reg4)
```

Formula:

Freq ~ time + venue + result + venue:result

attr(,"variables")

list(Freq, time, venue, result)

attr(,"factors")

	time	venue	result	venue:result
Freq	0	0	0	0
time	1	0	0	0
venue	0	1	0	1
result	0	0	1	1

attr(,"term.labels")

[1] "time" "venue" "result" "venue:result"

attr(,"order")

[1] 1 1 1 2

attr(,"intercept")

[1] 1

attr(,"response")

[1] 1

attr(,".Environment")

<environment: R_GlobalEnv>

attr(,"predvars")

list(Freq, time, venue, result)

attr(,"dataClasses")

Freq	time	venue	result
"numeric"	"factor"	"factor"	"factor"

Statistics:

	X^2	df	P(> X^2)
Likelihood Ratio	102.7124	90	0.1696832

Pearson 100.4088 90 0.2127138

```
extractAIC(reg4)[2]
```

[1] 150.7124

Once again we get to the same conclusion! With, and only with, the addition of the interaction between venue and result we have high p-values, meaning that those two factors are NOT independent.

4.1.3 Multinomial regression

```
library(nnet)
```

```
multi_model <- multinom(result ~ venue + attendance + attendance:venue, data = data)
```

weights: 15 (8 variable)
initial value 390.007362
iter 10 value 355.532753
final value 355.147170
converged

```
summary(multi_model)
```

Call:
multinom(formula = result ~ venue + attendance + attendance:venue,
    data = data)

Coefficients:

	(Intercept)	venueHome	attendance	venueHome:attendance
D	-0.03374428	0.4971558	-2.166837e-05	3.872944e-05
V	-0.10419221	0.9962857	-1.425605e-05	3.830826e-05

Std. Errors:

	(Intercept)	venueHome	attendance	venueHome:attendance
D	2.777067e-10	3.303411e-10	7.67296e-06	1.158192e-05
V	2.273974e-10	3.091649e-10	7.08233e-06	1.077347e-05

Residual Deviance: 710.2943

AIC: 726.2943

```
exp(summary(multi_model)$coeff)
```

	(Intercept)	venueHome	attendance	venueHome:attendance
D	0.9668187	1.644039	0.9999783	1.000039
V	0.9010521	2.708204	0.9999857	1.000038

Note that the coefficients for the interaction between venueHome and attendance is positive. That is a crucial piece of information to understand the meaning of this model.

```
summary_multi_model <- summary(multi_model)


coefs <- coef(summary_multi_model)
se <- summary_multi_model$standard.errors
```

```
z_values <- coefs / se
p_values <- 2 * (1 - pnorm(abs(z_values)))
p_values
```

	(Intercept)	venueHome	attendance	venueHome:attendance
D	0	0	0.004742969	0.0008259232
V	0	0	0.044124768	0.0003768353

That means that the public, venue, and their interaction are statistically significant to explain the outcome of the match, so I will keep all of them in the model.

We can create a fake data just to show how it works in prediction

```
example_test <- data[1:4,c(5,11)]
example_test[1,] <- c("Home", 20)
example_test[2,] <- c("Away", 20)
example_test[3,] <- c("Home", 40000)
example_test[4,] <- c("Away", 40000)
example_test$venue <- factor(example_test$venue)
example_test$attendance <- as.numeric(example_test$attendance)

predictions <- predict(multi_model,
                       newdata = example_test,
                       type = "probs")
head(predictions)
```

	L	D	V

1 0.19875038 0.3160190 0.4852306

2 0.34877293 0.3370541 0.3141730

3 0.09495234 0.2986372 0.6064104

4 0.52196983 0.2121154 0.2659147

Notice that we created four different kinds of sample:

1) playing home with low audience
2) playing away with low audience
3) playing home with high audience
4) playing away with high audience

Playing home we have the highest probabilities of winning (48.52% and 60.64%), while playing away the lowest (31.42% and 26.59%). Notice that higher audience doesn't imply in higher probability of victory for Corinthians, since the interaction between venue and attendance is in the model. Playing with a higher audience at home rises up the probability of victory by diminishing the lose probability (and just a little bit the chances of drawing).

When playing away the effect is the complete opposite. Playing away with higher audiences increases the chances of defeat while diminishing significantly the winning and drawing probabilities.

4.2 Assuming sample is not independent

Of course the assumption of sample independence is highly questionable as we are talking about the same team on different rounds/seasons. Also, as we saw on EDA, the match outcome is high dependable on the opponent team, so it must also be taken into account.

For that reason, we are gonna use non parametric test and mixed models to analyse the data.

4.2.1 Non parametric test

So, basically I wanna test considering that, by each year and opponent, the sample is correlated. By that I mean that I want to compare the same teams (paired sample) on different moments (playing home and away), so I will use the McNemar test.

4.2.1.1 Preparing the data

Lets separate only the clashes that happened both home and away on the same season

```
home <- data %>%
  filter(venue == 'Home') %>%
  filter(year < 2023)

away <- data %>%
  filter(venue == 'Visitante') %>%
  filter(year < 2023)
```

```
all.equal(home$Oponente, away$Oponente)
```

```
[1] TRUE
```

We wanna know if the proportion of victories changes when playing home in compared to when playing away, so lets turn the response variable into binary, 1 in case of victory and 0 otherwise.

```
data2 <- data
data2$binary_response <- ifelse(data$result == "V", 1, 0)
```

And now lets transform the data in a way that we can see, by every year and opponent, how Corinthians did.

```
transformed_data <-  data2 %>%
  group_by(Opponent, year) %>%
  reframe(
    Home = binary_response[venue == "Home"],
    Away = binary_response[venue == "Away"]
  ) %>%
  na.omit()

print(transformed_data %>% head())
```

```
# A tibble: 6 x 4
  Opponent        year  Home  Away
  <chr>          <int> <dbl> <dbl>
1 América (MG)    2016     1     1
2 América (MG)    2018     1     0
3 América (MG)    2021     0     1
4 América (MG)    2022     0     0
5 Atl Goianiense  2017     0     1
6 Atl Goianiense  2020     0     0
```

```
table_mc <- table(Home = transformed_data$Home, Away = transformed_data$Away)
table_mc
```

```
    Away
Home  0  1
   0 53 19
   1 67 32
```

The table show us that 53 times we did not win against a team when playing home and either when playing away. 19 times we did not win when playing home, but got the victory playing away. 67 times we won playing home, but draw or lost away. And 32 times we won both home and away.

```
mcnemar.test(table_mc)
```

```
    McNemar's Chi-squared test with continuity correction

data:  table_mc
McNemar's chi-squared = 25.686, df = 1, p-value = 4.017e-07
```

The test is telling us that the proportion of victory is really different when playing home than playing away. Just as expected!

4.2.2 Mixed Modells

As said before, I want to control based on opponent and season.

```
library(lme4)
```

Carregando pacotes exigidos: Matrix

Anexando pacote: 'Matrix'

Os seguintes objetos são mascarados por 'package:tidyr':

    expand, pack, unpack

```
data2$venue <- as.factor(data2$venue)
data2$victory <- ifelse(data2$result == "V", 1, 0)
data2$draw_loss <- 1 - data2$victory

mixed_mod <- glmer(
  cbind(victory, draw_loss) ~ venue + attendance + venue:attendance + (1 | Opponent) + (1 | year),
  family = binomial,
  data = data2
)

summary(mixed_mod)
```

Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial (logit)
Formula: cbind(victory, draw_loss) ~ venue + attendance + venue:attendance +

```
  (1 | Opponent) + (1 | year)
   Data: data2

     AIC      BIC   logLik deviance df.resid
   465.5    488.8   -226.8    453.5      349

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.4995 -0.7225 -0.5645  0.8833  1.8441

Random effects:
 Groups   Name        Variance Std.Dev.
 Opponent (Intercept) 0.02323  0.1524
 year     (Intercept) 0.08929  0.2988
Number of obs: 355, groups:  Opponent, 33; year, 10

Fixed effects:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)        -7.902e-01  2.429e-01  -3.253  0.00114 **
venueHome           8.263e-01  3.006e-01   2.749  0.00597 **
attendance         -6.414e-06  8.533e-06  -0.752  0.45222
venueHome:attendance  1.548e-05  9.984e-06   1.551  0.12098
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) venuHm attndn
```

venueHome -0.518

attendance -0.583 0.171

vnHm:ttndnc 0.288 -0.631 -0.422

fit warnings:

Some predictor variables are on very different scales: consider rescaling

optimizer (Nelder_Mead) convergence code: 0 (OK)

Model failed to converge with max|grad| = 2.28573 (tol = 0.002, component 1)

Model is nearly unidentifiable: very large eigenvalue

- Rescale variables?

Model is nearly unidentifiable: large eigenvalue ratio

- Rescale variables?

Note the fit warnings. Lets re scale the attendance variable to avoid errors.

```
data2$scaled_attendance <- (data2$attendance - mean(data2$attendance))/sd(data2$attendance)

mixed_mod <- glmer(
  cbind(victory, draw_loss) ~ venue + scaled_attendance + venue:scaled_attendance + (1 | Opponent
  family = binomial(),
  data = data2
)

summary(mixed_mod)
```

Generalized linear mixed model fit by maximum likelihood (Laplace

Approximation) [glmerMod]

Family: binomial (logit)

Formula:

```
cbind(victory, draw_loss) ~ venue + scaled_attendance + venue:scaled_attendance +
    (1 | Opponent) + (1 | year)
   Data: data2

     AIC      BIC   logLik deviance df.resid
   465.5    488.8   -226.8    453.5      349

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.4994 -0.7225 -0.5645  0.8833  1.8441

Random effects:
 Groups   Name        Variance Std.Dev.
 Opponent (Intercept) 0.02322  0.1524
 year     (Intercept) 0.08929  0.2988
Number of obs: 355, groups:  Opponent, 33; year, 10

Fixed effects:
                           Estimate Std. Error z value Pr(>|z|)
(Intercept)                -0.9421     0.2061  -4.571 4.85e-06 ***
venueHome                   1.1930     0.2419   4.931 8.19e-07 ***
scaled_attendance          -0.1031     0.1885  -0.547    0.584
venueHome:scaled_attendance  0.2489     0.2460   1.012    0.312
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
```

```
            (Intr) venuHm scld_t
venueHome   -0.642
scld_ttndnc  0.293 -0.227
vnHm:scld_t -0.168  0.001 -0.649
```

The model tell us the same as before. When playing home the coefficient for attendance is 0.2489 (that means that when playing home the bigger the audience, higher are the chances of winning), but when playing away it goes negative to -0.1031 (when playing away higher audiences means higher chances of not winning).

When playing home with no audience the coefficient is 1.1930, but when playing away with no audience is -0.9441, and we can understand that by the same way done above.

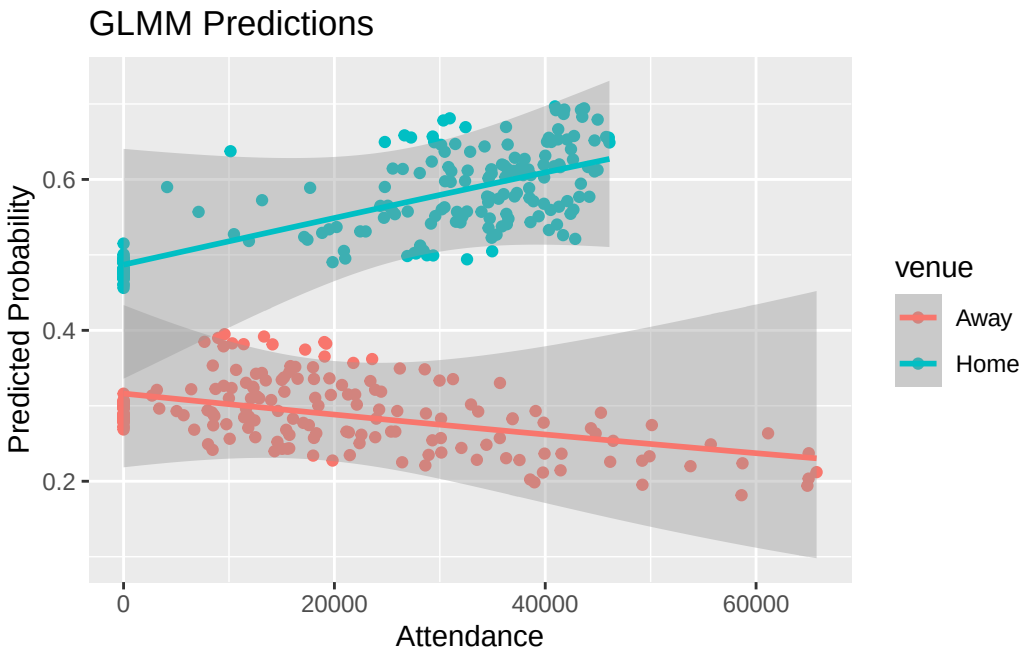
We can test this model the same way we did with the multinomial regression.

```
example_test2 <- data2[1:4,c(5,9, 17, 21)]
example_test2[1,] <- c("Home", "Grêmio", 2014, 20)
example_test2[2,] <- c("Away", "Grêmio", 2014, 20)
example_test2[3,] <- c("Home", "Grêmio", 2014, 40000)
example_test2[4,] <- c("Away", "Grêmio", 2014, 40000)
example_test2$scaled_attendance <- as.numeric(example_test2$scaled_attendance)
example_test2$scaled_attendance <- scale(example_test2$scaled_attendance)

predictions <- predict(mixed_mod,
                       newdata = example_test2,
                       type = "response"
                       )
predictions
```

```
        1         2         3         4
0.5444305 0.3101945 0.6060338 0.2733154
```

And we can also visualize the model by using

```
predictions <- predict(mixed_mod, type = "response", newdata = data2)
# Visualization
ggplot(data2, aes(x = attendance, y = predictions, color = as.factor(venue))) +
  geom_point() +
  geom_smooth(method = "glm", method.args = list(family = "binomial")) +
  labs(title = "GLMM Predictions",
       x = "Attendance",
       y = "Predicted Probability",
       color = "venue")
```



Notice that the intercept for playing home is much higher than for playing away. Additionally, when playing home the slope is positive. Meanwhile, when playing away it is negative, endorsing the results previously obtained.

This model has only one problem, but it's a major one: we notice that the attendance and the interaction between attendance and venue is not significant.

Now lets select variables based on their significance. Removing the interaction first:

```
mixed_mod_no_interaction <- glmer(
  cbind(victory, draw_loss) ~ venue + scaled_attendance + (1 | Opponent) + (1 | year), #result_bina
  family = binomial(),
  data = data2
)

summary(mixed_mod_no_interaction)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: cbind(victory, draw_loss) ~ venue + scaled_attendance + (1 |
    Opponent) + (1 | year)
   Data: data2

     AIC      BIC   logLik deviance df.resid
   464.6    483.9   -227.3    454.6      350

Scaled residuals:
    Min      1Q  Median      3Q     Max
```

-1.4521 -0.7312 -0.5664 0.8839 1.7931

Random effects:

Groups	Name	Variance	Std.Dev.
Opponent	(Intercept)	0.03958	0.1989
year	(Intercept)	0.10489	0.3239

Number of obs: 355, groups: Opponent, 33; year, 10

Fixed effects:

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-0.91078	0.20810	-4.377	1.20e-05	***
venueHome	1.19830	0.24093	4.974	6.57e-07	***
scaled_attendance	0.01952	0.14501	0.135	0.893	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:

	(Intr)	venuHm
venueHome	-0.631	
scld_ttndnc	0.221	-0.283

Removing the attendance (p-value of 0.893)

```
mixed_mod_no_attendance <- glmer(
  cbind(victory, draw_loss) ~ venue + (1 | Opponent) + (1 | year), #result_binary
  family = binomial(),
  data = data2
)
```

```
summary(mixed_mod_no_attendance)
```

```
Generalized linear mixed model fit by maximum likelihood (Laplace
  Approximation) [glmerMod]
 Family: binomial  ( logit )
Formula: cbind(victory, draw_loss) ~ venue + (1 | Opponent) + (1 | year)
   Data: data2

     AIC      BIC   logLik deviance df.resid
   462.6    478.1   -227.3    454.6      351

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.4419 -0.7344 -0.5659  0.8895  1.7971

Random effects:
 Groups   Name        Variance Std.Dev.
 Opponent (Intercept) 0.03457  0.1859
 year     (Intercept) 0.10805  0.3287
Number of obs: 355, groups:  Opponent, 33; year, 10

Fixed effects:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.9170     0.2032  -4.512 6.43e-06 ***
venueHome     1.2076     0.2310   5.227 1.73e-07 ***
---
```

Signif. codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Correlation of Fixed Effects:

(Intr)

venueHome -0.606

Note that in this model all variables are significant, and both AIC and BIC are lower than the original model (with interaction), hence that is chosen to be the best model.

And now we shall test it

```
example_test3 <- data2[1:4,c(5,9, 17)]
example_test3[1,] <- c("Home", "Flamengo", 2014) # actually won at home
example_test3[2,] <- c("Away", "Flamengo", 2014) # and lost away
example_test3[3,] <- c("Home", "Cruzeiro", 2014) # won both home and away
example_test3[4,] <- c("Away", "Cruzeiro", 2014)
example_test3[5,] <- c("Home", "Coritiba", 2014) # didn't win both home and away
example_test3[6,] <- c("Away", "Coritiba", 2014)

predictions <- predict(mixed_mod_no_attendance,
                       newdata = example_test3,
                       type = "response"
                       )
predictions
```

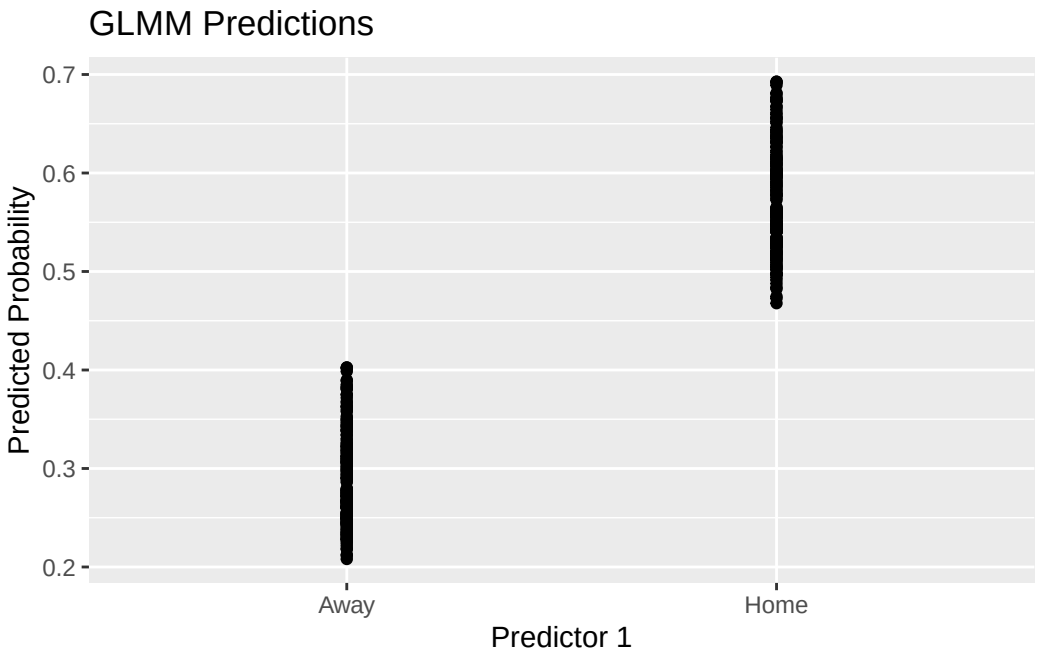
```
        1         2         3         4         5         6
0.5948399 0.3050061 0.6184727 0.3264004 0.6138216 0.3220915
```

```
data2 %>% filter(year == 2014) %>% dplyr::select(date, Opponent, result) %>% arrange(Opponent)
```

	date	Opponent	result
1	2014-08-03	Coritiba	D
2	2014-11-01	Coritiba	D
3	2014-05-28	Cruzeiro	V
4	2014-10-08	Cruzeiro	V
5	2014-04-27	Flamengo	V
6	2014-09-14	Flamengo	L

```
predictions <- predict(mixed_mod_no_attendance,
                       newdata = data2,
                       type = "response"
                       )
# Visualization
ggplot(data2, aes(x = venue, y = predictions)) +
  geom_point() +
  geom_smooth(method = "glm", method.args = list(family = "binomial")) +
  labs(title = "GLMM Predictions",
       x = "Predictor 1",
       y = "Predicted Probability")
```

‘geom_smooth()‘ using formula = ’y ~ x’



A much more boring graph but shows what the model says: When playing home the winning probability is much higher than playing away.

5 Conclusions

From what was saw on the previously chapter, we can conclude that for Corinthians over the past 10 years:

1. Impact of Audience Attendance: The size of the attendance does not appear to significantly influence match outcomes. Having an audience, even a small one, seems sufficient to provide an advantage. However, an attendance of zero might negate this advantage, as suggested by the McNemar test during the pandemic years. If it does have any impact, than it is important to note that higher audience gives the host team higher chances of winning, so having high audience doesn't necessarily means higher winning chances, you have to also be playing home, otherwise the effect is the complete opposite.

It is worth noting that the importance of venue and result can vary between seasons. For instance, in seasons where the team performs exceptionally well (2015 and 2017 for example), playing at home or away has little impact because the team's chances of winning are consistently high. Conversely, in weaker seasons, the venue might don't matter because the chances of winning are consistently low. Maybe the pandemic years were just one of those particular case. These nuances highlights the need for further investigation, as discussed later.

2. Correlation between attendance and venue: Audience attendance is strongly correlated with the location of the match, as expected. Playing home means, in general, higher audience.

3. Limitations of the Analysis: The findings are subject to several limitations, including:

    1. Limited Observations: There are only two observations per year for each opposing team, which may lead to imprecise estimates.
    2. Relegation Effects: Some teams are relegated after a single season, resulting in very few observations for those teams. This sparsity may negatively impact model fitting.
    3. Data Accuracy: While the data was carefully extracted from the website, we cannot verify its original accuracy or the methodology used to collect it.

Future Work:

To address these limitations, data should be collected for all teams that played in the Brazilian league over the past 10 years. This would provide a larger sample size and more robust observations for analysis. Additionally, employing cross-validation techniques would help improve the reliability of the results and reduce potential overfitting.