

Time Series Report

FiFa Project



January 26, 2024

By: Mariam Baydoun.

The aim of this project is :

to predict the number of goals that Portugal will have in 2023.

EX:1

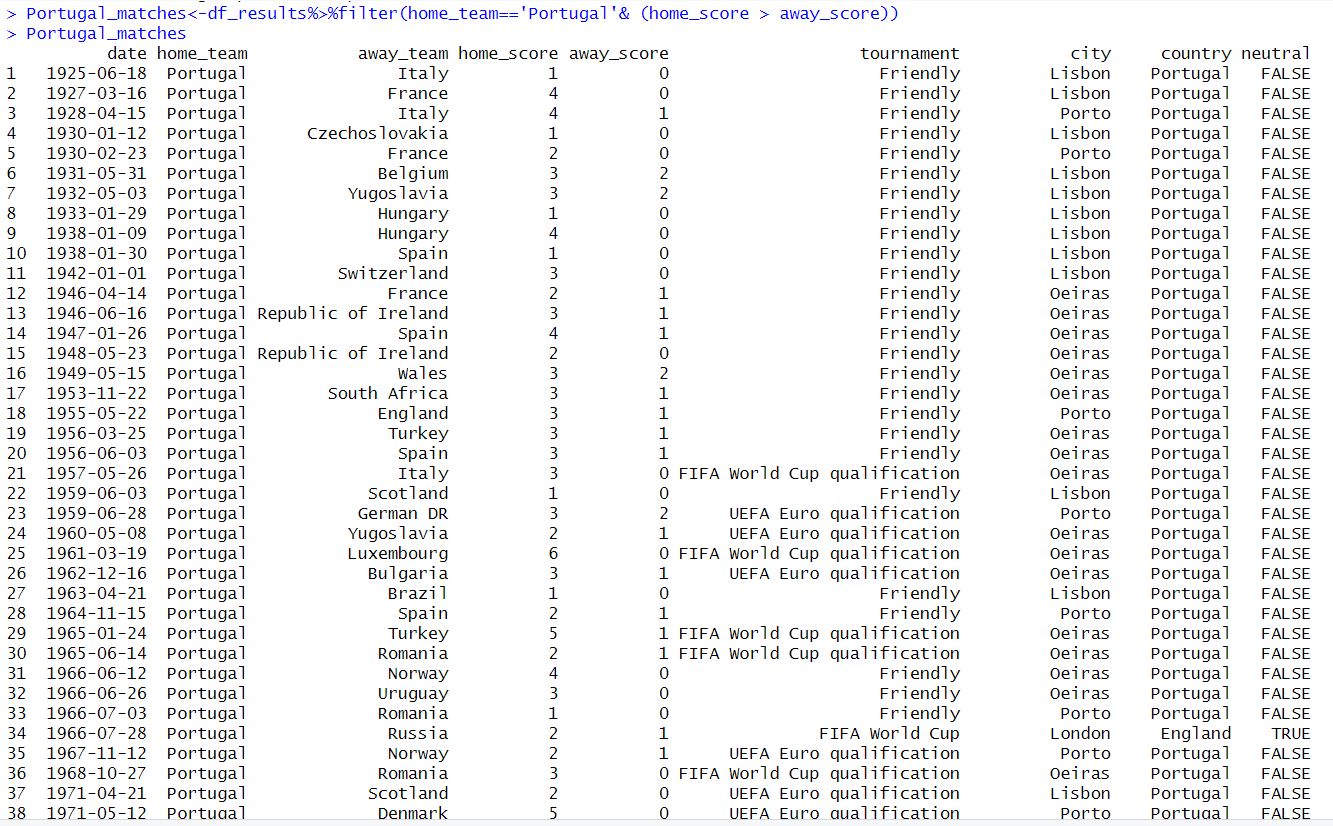
a) df\_results=read.csv("C:\\Users\\Mariam Baydoun\\Desktop\\Project\_TimeSeries\\results.csv")

df\_goal.scorers=read.csv("C:\\Users\\Mariam Baydoun\\Desktop\\Project\_TimeSeries\\goalscorers.csv")

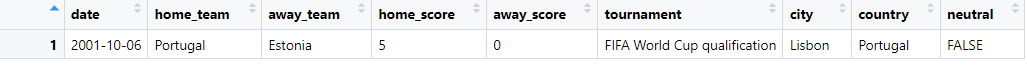
b) The data frame df\_results provides information about each match between two teams, including the match date, scores achieved by each team, with another some variables such as city and country.

The data frame df\_goal.scorers is data frame shows more who are the scorers of each game with the time and type of the score if it is own goal or penalty.

c) Portugal\_matches<-df\_results%>%filter(home\_team=='Portugal'& (home\_score > away\_score| away\_score > home\_score))



d) df\_match <- Portugal\_matches%>%filter(date == "2001-10-06")



e) df\_match\_details <- df\_goal.scorers %>% inner\_join(Portugal\_matches)%>%filter(date == "2001-10-06")



df\_scorer <- df\_match\_details %>% group\_by(scorer)%>% count()%>% arrange(desc(n))

df\_scorer[1,1]



With respect to this result: The player “Nuno” is scored the most goals.

f) df\_tournament\_c <- df\_results %>% group\_by(tournament) %>% count()

df\_tournament\_c <- df\_tournament\_c %>% arrange(desc(n))

df\_tournament\_c %>% head(5)



EX:2

a) library(lubridate)

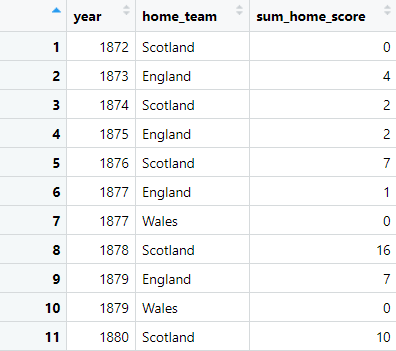
df\_fifa\_results <-df\_results %>% mutate(year=year(ymd(df\_results$date)))

b) df\_fifa\_results <- df\_fifa\_results %>% filter(year(date) != 2023)

c) part1

df\_home\_teams <- df\_fifa\_results %>% group\_by(year, home\_team) %>% summarize(sum\_home\_score = sum(home\_score))

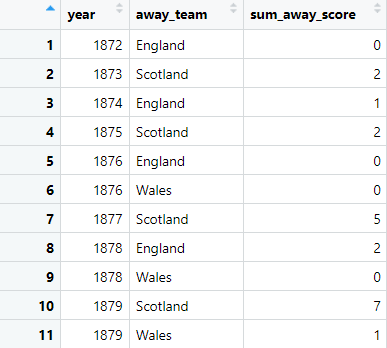
view(df\_home\_teams)



c) part2:

df\_away\_teams <- df\_fifa\_results %>% group\_by(year, away\_team) %>% summarize(sum\_away\_score = sum(away\_score))

view(df\_away\_teams)



C) part3

df\_home\_teams <- df\_home\_teams %>% rename(team = home\_team, scores = sum\_home\_score)

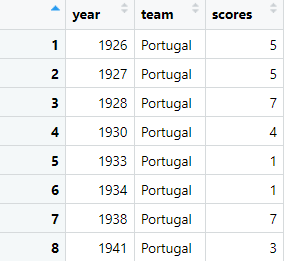
df\_away\_teams <- df\_away\_teams %>% rename(team = away\_team, scores = sum\_away\_score)

df\_fifa\_goals <- df\_home\_teams %>% inner\_join(df\_away\_teams, by = c("year","team"))

df\_fifa\_goals <- df\_fifa\_goals %>% mutate(scores = scores.x + scores.y)

df\_fifa\_goals <- df\_fifa\_goals %>% filter(team=="Portugal")%>%select(year, team, scores)

view(df\_fifa\_goals)



C:part4

library(ggplot2)

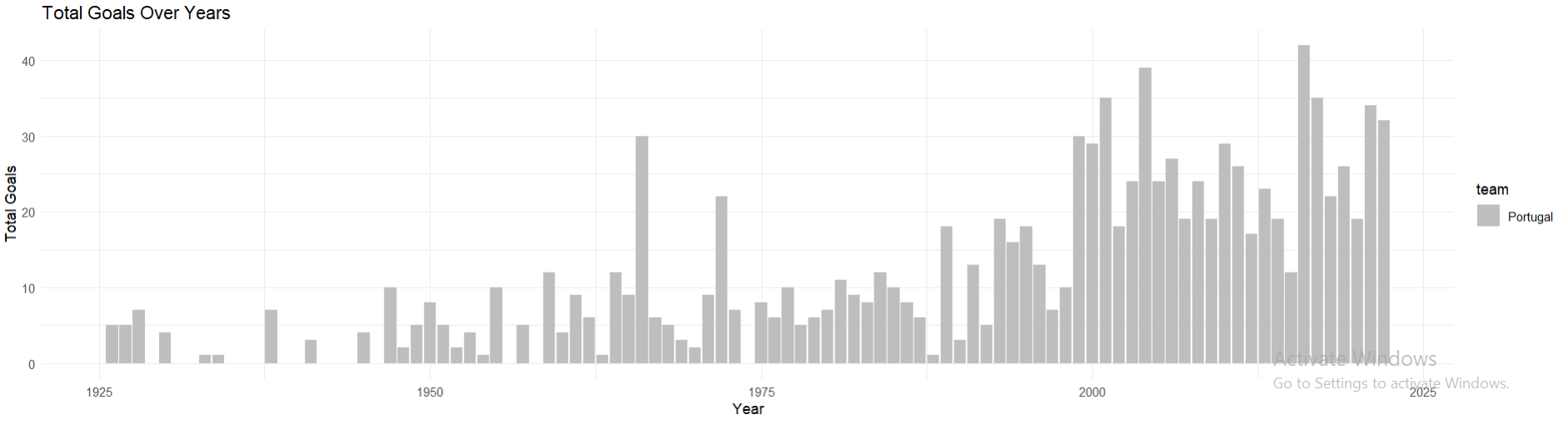
ggplot(df\_fifa\_goals, aes(x = year, y = scores, fill = team)) +

geom\_col() +

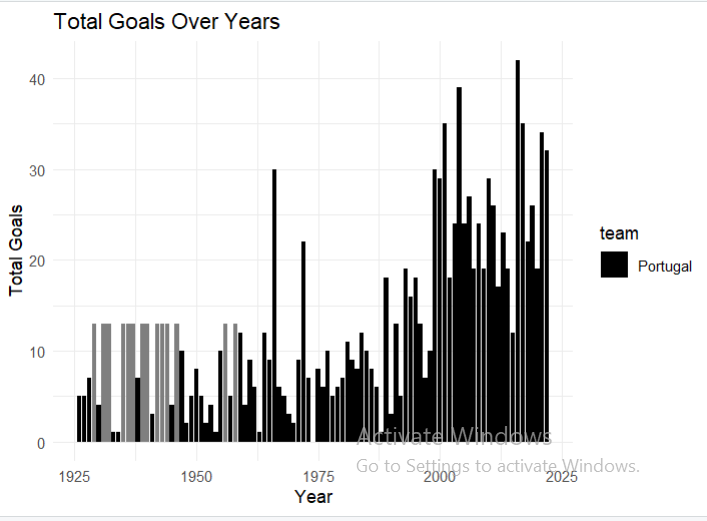
scale\_fill\_manual(values = c("Portugal" = "grey", "Other Teams" = "blue")) +

labs(title = "Total Goals Over Years", x = "Year", y = "Total Goals") +

theme\_minimal()



“This graph shows that as time increased from year 2000 to 2025 the number of goals increased significantly in these years, compared to the number of goals in the year before also, we can observe that we have missing years in our graph and the spread of this missing can be observed will ”



The white gray columns here represent the missing years in our graph.

3) show mean and variance

m <- mean(df\_fifa\_goals$scores,na.rm=TRUE) = 13

s <- sd(df\_fifa\_goals$scores) = 9.599262

Find the missing years :

year\_range <- min(df\_fifa\_goals$year):max(df\_fifa\_goals$year)

missing\_years <- setdiff(year\_range, df\_fifa\_goals$year)

df\_missing\_years <- data.frame(year = missing\_years, scores = rep(m, length(missing\_years)))

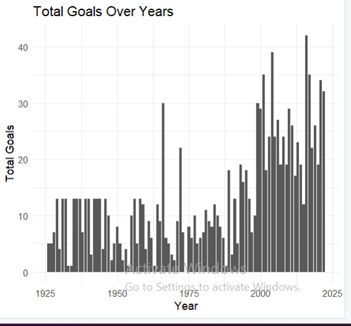
df\_fifa\_goals\_filled <- bind\_rows(df\_fifa\_goals, df\_missing\_years)

df\_fifa\_goals\_filled <- df\_fifa\_goals\_filled[order(df\_fifa\_goals\_filled$year), ]

ggplot(df\_fifa\_goals\_filled, aes(x = year, y = scores)) +

geom\_col() +

labs(title = "Total Goals Over Years", x = "Year", y = "Total Goals") +

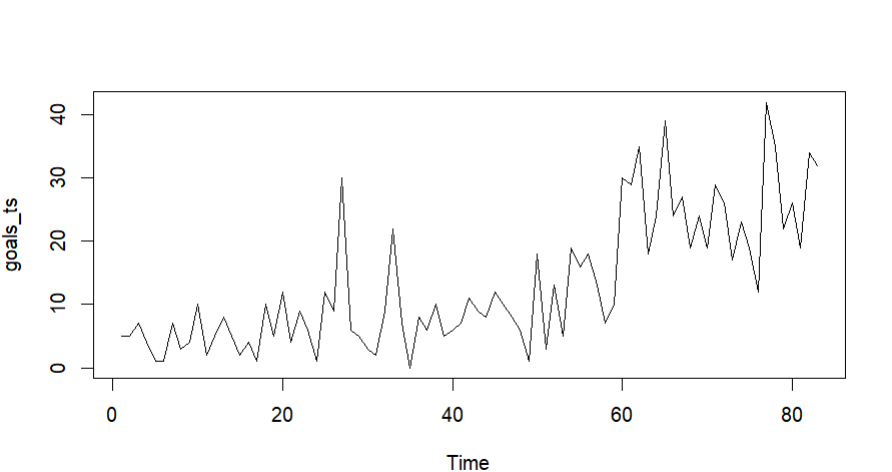
theme\_minimal()  “The graph after finding the missing years”

Now we will start modeling:

ARIMA modeling:

1. Convert your data to time series object:

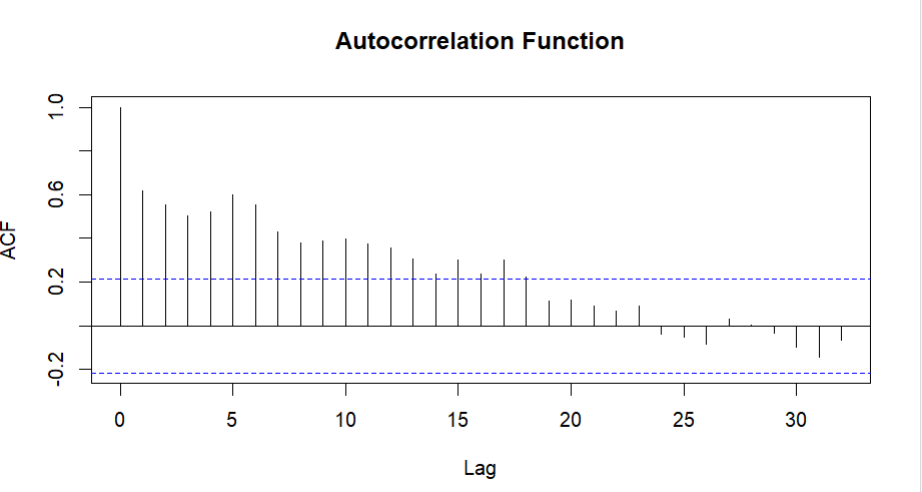
* goals\_ts <- ts(df\_fifa\_goals$scores, frequency = 1)
* plot(goals\_ts)



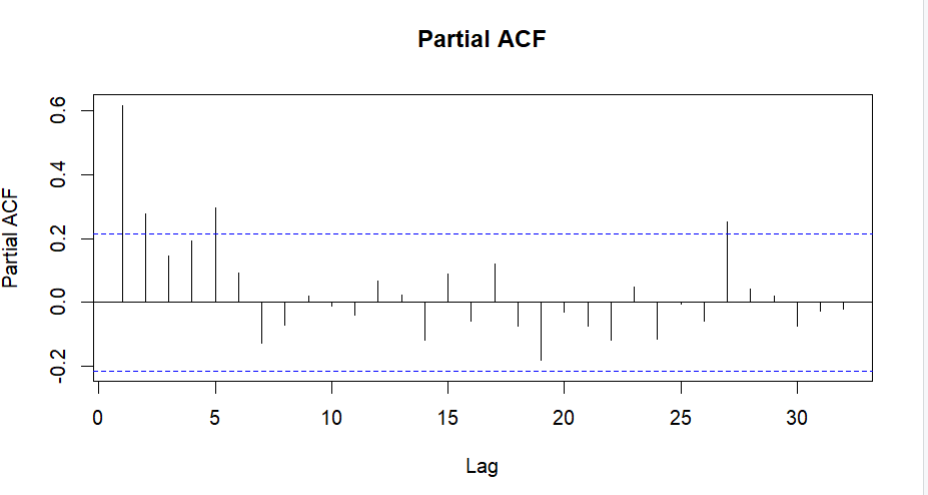
We can see here that we have trend that can be removed using differencing but there is no seasonality, also the data is not stationary.

2) Plot autocorrelation and partial autocorrelation function

acf(goals\_ts, main="Autocorrelation Function",lag.max = 32)



acf(goals\_ts, type="partial", main="Partial ACF",lag.max = 32)

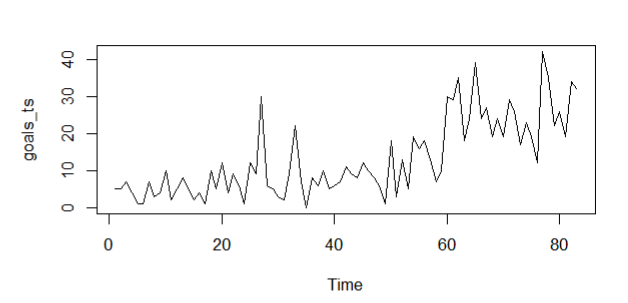


3) Does this series look stationary?

“ACF” figure shows that the spikes line above the significant line and decay in a slowly way this indicates that there is an autocorrelation in our data and it is not stationary ,using differencing the data will be e stationary .

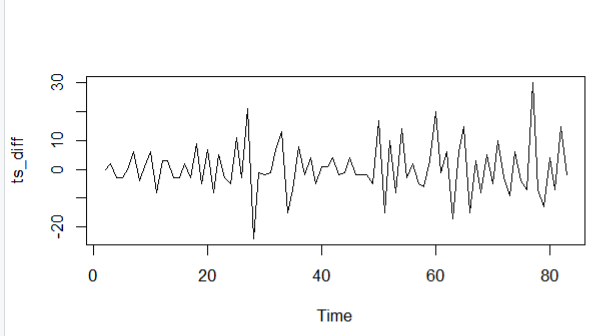
If not, perform the corresponding transformations and/or differencing to make it stationary:

Before making differencing:



We can show clearly that our data not stationary

After making differencing:



We can show that we do not have a trend and it is stationary

plot(goals\_ts)

ts\_diff<-diff(goals\_ts)

par(mfrow=c(2,1))

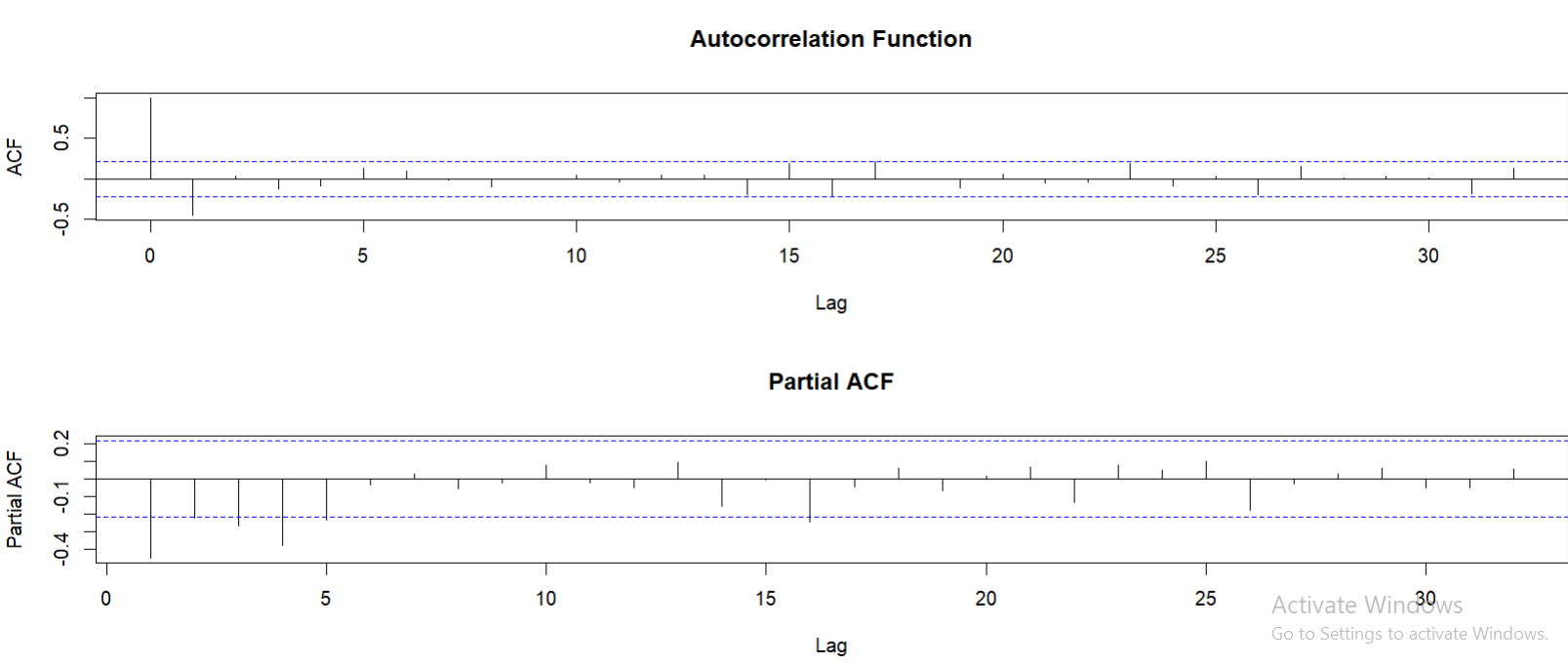
plot(ts\_diff)

4) Plot the ACF and PACF of the transformed series

par(mfrow=c(2,1))

acf(ts\_diff, main="Autocorrelation Function",lag.max = 32)

acf(ts\_diff, type="partial", main="Partial ACF",lag.max = 32)



“ACF” shows that the spikes inside the significant line except at lag 0 which is always =1 this that the data is stationary and no autocorrelation between our data.

The PACF shows that the spikes has exponential shape which indicates we have MA(1) since the ACf has last spike at lag=1.

5) Identify a couple of ARIMA models that might be useful in describing the time series. Which of your models is the best according to their AIC values?

arima111<-arima(goals\_ts, order = c(1,1,1))

arima111$ai

[1] 559.7913

arima211$aic

[1] 561.6683

arima212<-arima(goals\_ts, order = c(2,1,2))

arima212$aic

[1] 563.7779

According to AIC arima111(1,1,1) is better since its value which equal to =559.7913 is less than the

AIC values of . arima212= 563.7779 and arima211= 561.6683.

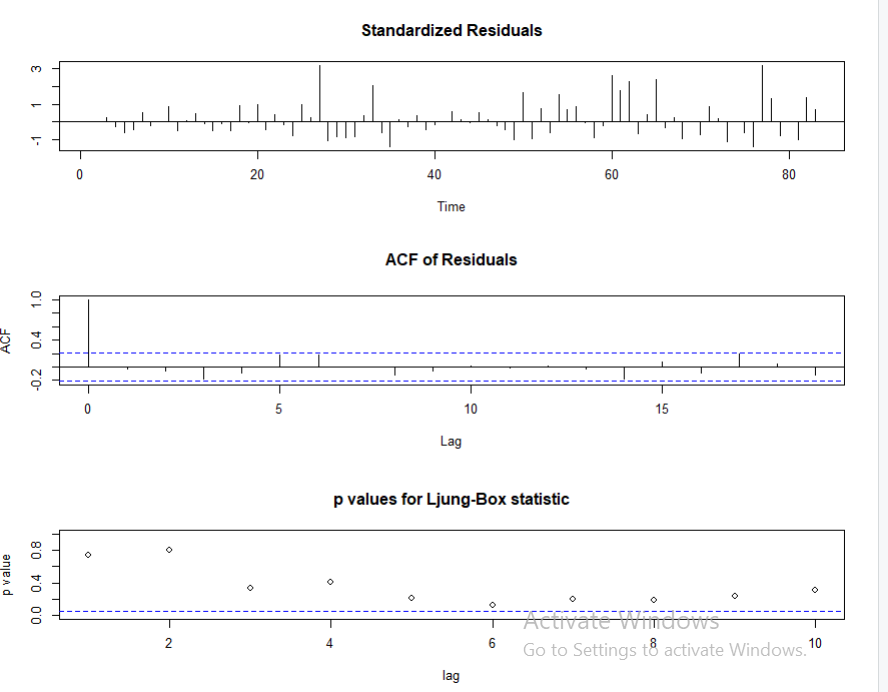
6) Use the ARIMA function to estimate and fit the identified models

library(forecast)

auto.arima(goals\_ts)

fit.arima <- arima(goals\_ts, order = c(1,1,1))

7) After estimating the parameters of your best model, perform the diagnostic testing on the residuals. Do the residuals resemble white noise? Normally distributed? Use shapiro test, Portmanteau test, plot the histogram of residuals



ACF shows white noise the spikes inside the significant lines and the P-value more than alpha=0.05. Null hypothesis is not rejected. It is serially uncorrelated and shows that the model is good.

tsdiag(fit.arima)

library(stats)

library(tseries)

residuals <- residuals(fit.arima)

# Test for normality

shapiro.test(residuals)

# Test for autocorrelation

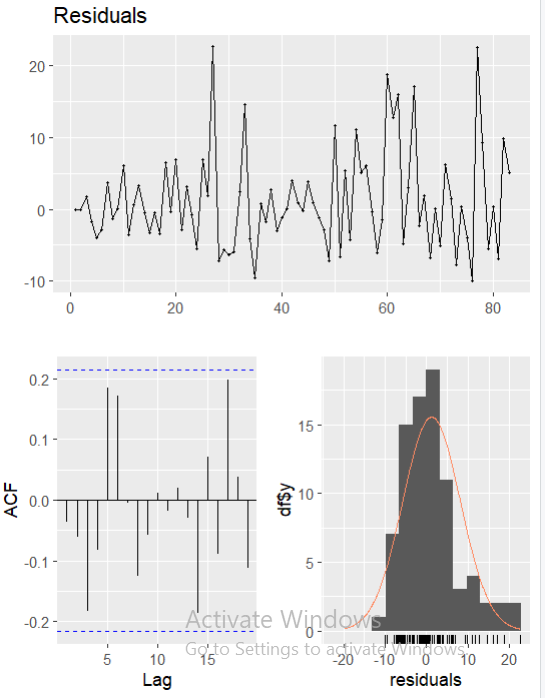
library(lmtest)

test<-Box.test(residuals,type=c("Ljung-Box"),lag = 10)

# Plot the residuals

hist(residuals)

checkresiduals(residuals)



The ACF shows that the residuals are white noise.

8) Forecast the number of goals in year 2023 using your best fitted model.

library(forecast)

forecast\_goals<-forecast(fit.arima,h=1)

forecast\_goals$mean

plot(forecast(fit.arima,h=1))

#so for one year which is 2023 the mean of scores in 28 goal

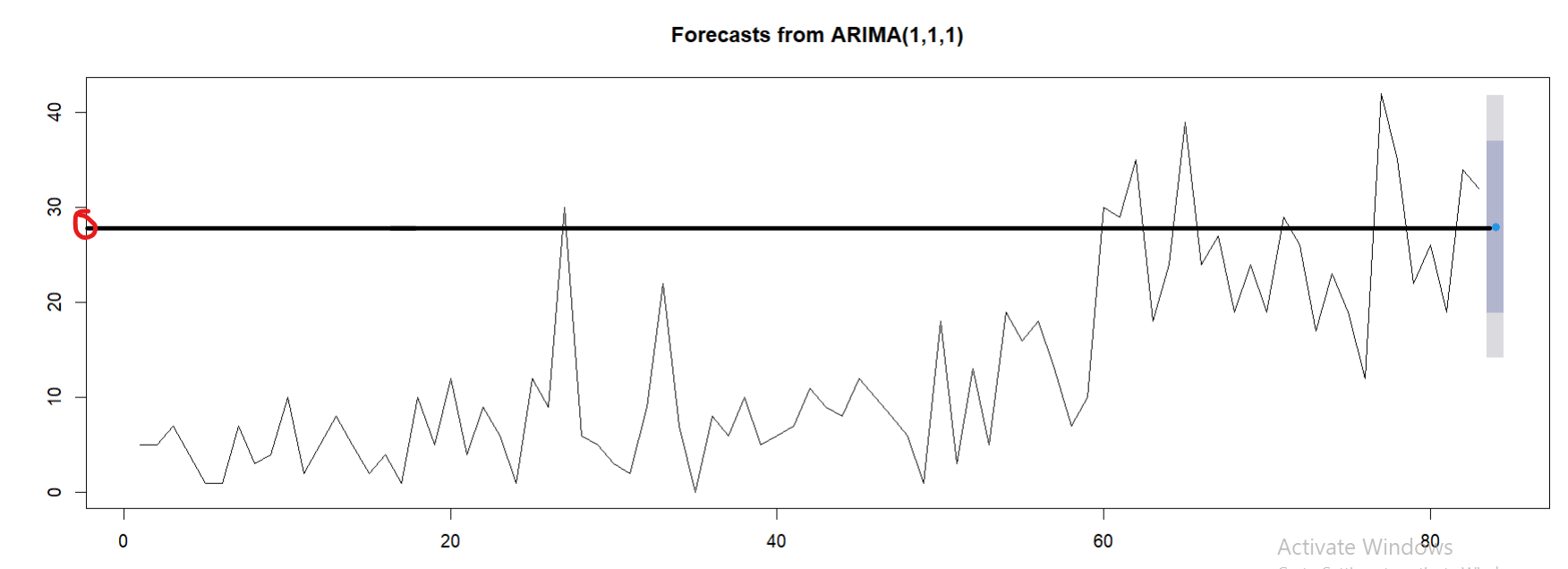
Time Series:

Start = 84

End = 84

Frequency = 1

27.9549



Nb of goals in 2023 Approximately=28

9)Based on our last lecture, try to add features to the dataset (time, observations, new variables, etc…) and solve the forecasting problem using neural network architecture

To solve the forecasting problem using a neural network architecture, we need to add additional features to the dataset. One common approach is to include time-related variables such as month, quarter, or year. We can also consider lagged values of the target variable as additional features

Conclusion:

Purtogal's data was filtered to obtain only relevant information for predicting their 2023 goals. The static data was removed to ensure it was stationary and fit the ARIMA model. The ACF and PACF were visualized to determine the order of the model. The best model was fitted from the min AIC, and residuals were found to be normal, indicating white noise. The model was then applied to the 2023 forecasting function, revealing Purtogal's goal forecast of 28 for 2023.