FIA FORMULA 1: FORECASTING AND PREDICTING RACE WINS AND DRIVER ACCIDENTS

Alyssa Brunen

New York Sumer 1 2023
Forecasting & Predicting the Future using Data
DAT- 3531

Table of Contents

Table of Figures

Table of Tables

Introduction

Experience the adrenaline-filled thrills of Formula 1, a sport that has captivated the imaginations of millions across the world. Formula 1's appeal knows no boundaries, from the roaring engines to the stunning overtakes. It's a thrilling mix of speed, talent, and drama that keeps spectators on the edge of their seats, waiting for the next race.

Winning in Formula One is the ultimate goal for drivers and their teams. They are propelled ahead by hours of commitment, rigorous preparation, and the quest of excellence. The urge to stand on the podium's top step and have their names written in history ignites their determination.

However, Formula One is not for the faint of heart. Every turn, every split-second decision is filled with risk. It is a slim line between success and tragedy. The extreme speeds, limited circuits, and harsh nature of the sport create an adrenaline-fueled environment in which daring and talent are combined. It is a genuine test of human endurance and the never-ending quest of perfection.

However, it is precisely this danger that adds an unmistakable fascination to Formula 1, where countless lives have already been lost. Formula 1 is an unequaled spectacle due to the combination of ability, strategy, cutting-edge technology, and the ever-present element of risk. It's a symphony of speed and perfection that has captivated spectators all around the world.

This report transports the reader into a world of speed, emotion, and drive, while forecasting and predicting what makes a race winner, and when the sport may suffer its next accident.

Correlation of Variables: Race Wins

Race victories are the pinnacle of racing success, marking the ultimate victory for drivers and teams, which is why, Race_Wins can be considered a business success. They demonstrate outstanding talent, strategy, collaboration, and the ability to overcome obstacles like car performance, track conditions, tire management, and tactical decision-making to win on the podium. When looking at individual drivers and their performance during their active years, whether they are still active on the grid or not, multiple factors stand out as what creates a Race Winner. In *figure 1* below, a correlation matrix can be seen for some of these race factors. Some variables that will be analyzed further: Pole_Rate, Win_Rate, FastLap_Rate, FastestLap_Rate, and Podium_Rate.

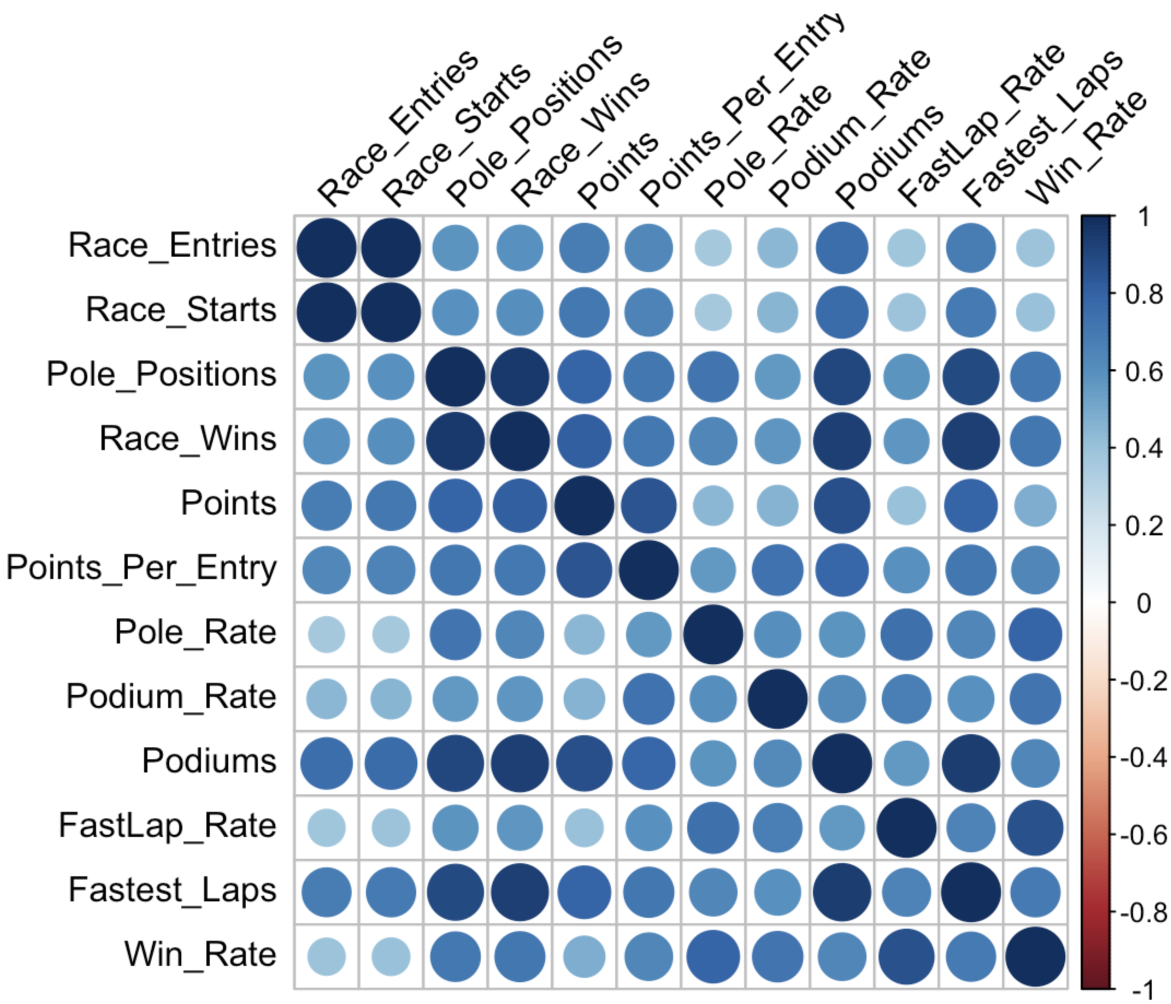

Figure 1 Correlation Matrix All Drivers 1950-Present

However, not the same can be said when looking at the current grid of 2023 in terms of variables influencing race wins, which can be seen in *figure 2* below. This is due to the not-as-intense championship battles one sees today; the starting grid is dominated by the teams such as Red Bull, Mercedes, Aston Martin, and Ferrari, as it has over the past years, due to constant regulatory changes. The likes of Max Verstappen (RB), Sergio Perez (RB), Lewis Hamilton (MERC), or Charles Leclerc (FER) have dominated and have left little to no room for many other race winners to appear. The correlation matrix represents the on-track action accurately and is also the reason why the following sections will refrain from using the current 20 drivers to analyze what makes a race winner. This is not only due to the changing regulations by the FIA making it harder for low-tier teams to create competitive chassis but also due to three rookies in these teams joining the 2023 season who will make the forecasting & prediction inaccurate.

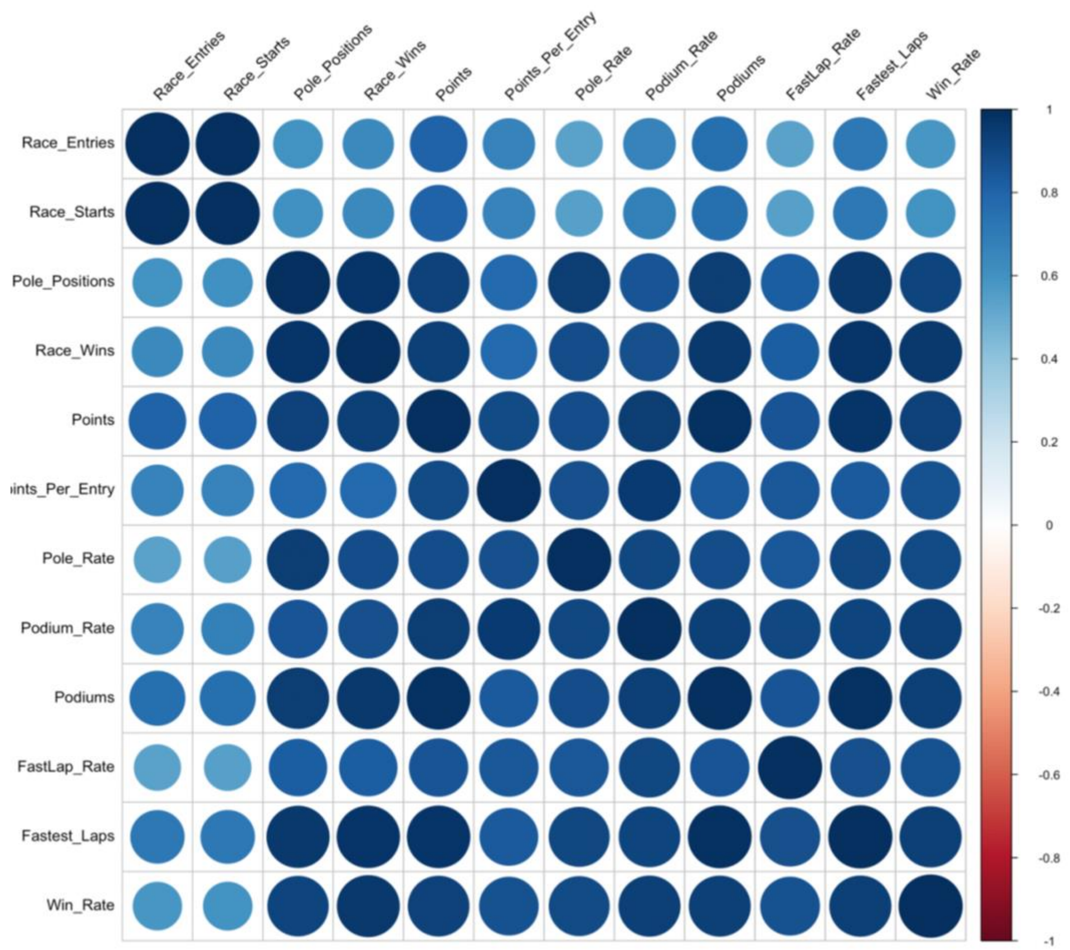

Figure 2 Correlation Matrix Active Drivers 2023

Forecasting: What makes a Race Winner?

As it was determined earlier, being a race winner while being in Formula One is considered a business success. However, other variables, as seen in the previous section, also relate to winning on track. In *figure 3* below, a GINI tree can be seen where zero represents business failure or, in this case, not a race winner, while one represents business success or being a race winner. It is important to note that Formula One has had 868 drivers since starting in the 1950s. Out of these 868 drivers, only 113 were race winners, which means they have won at least one race while there were actively on the grid, making the number of business successes relatively low.

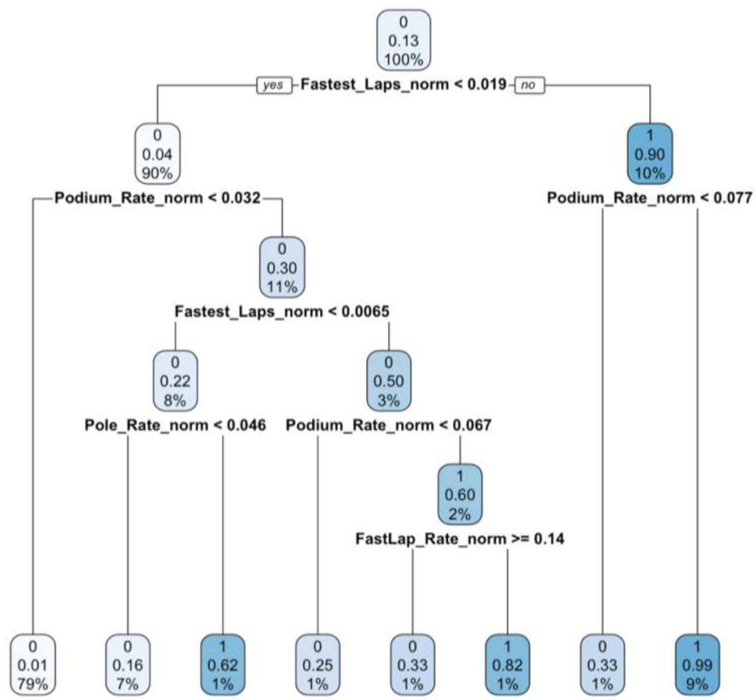

Figure 3 Decicison Tree Race Winners

As to the insights one can get from this decision tree, one can see that having the fastest laps and a high podium rate are the key factors that create a race winner. What that means on the other side, though, is that if a driver is given a poor-performing car that drives slow, he is less likely to have a fast lap which in turn means he is less likely to become a race winner. This is very accurate since Formula One is the pinnacle of motorsport, where the fastest cars are more likely to win not only a race but the championship of the year as well. This tells us that it becomes even more important for the team to focus on the R&D of their chassis to reach the highest performance possible to give their drivers the best car to arrive with on track to maximize potential race wins.

Correlation of Variables: Driver Accidents

In the world of Formula One, where speed and competitiveness collide, accidents are a vital factor that must be thoroughly understood. With high-speed racing and drivers pushing the limits, understanding the possible frequency and consequences of accidents is critical. This insight helps teams, drivers, and stakeholders to improve safety measures, optimize strategy, and safeguard the health of participants in this thrilling but fundamentally dangerous sport. In the database found, dates of fatal accidents, the age of the driver, team ('car'), and session type at the time of occurrence were given. Teams and Session types were converted into a factor within the data frame in order to continue. In the correlation matrix, seen in *figure 4* below, the team and session variables are being considered going forward.

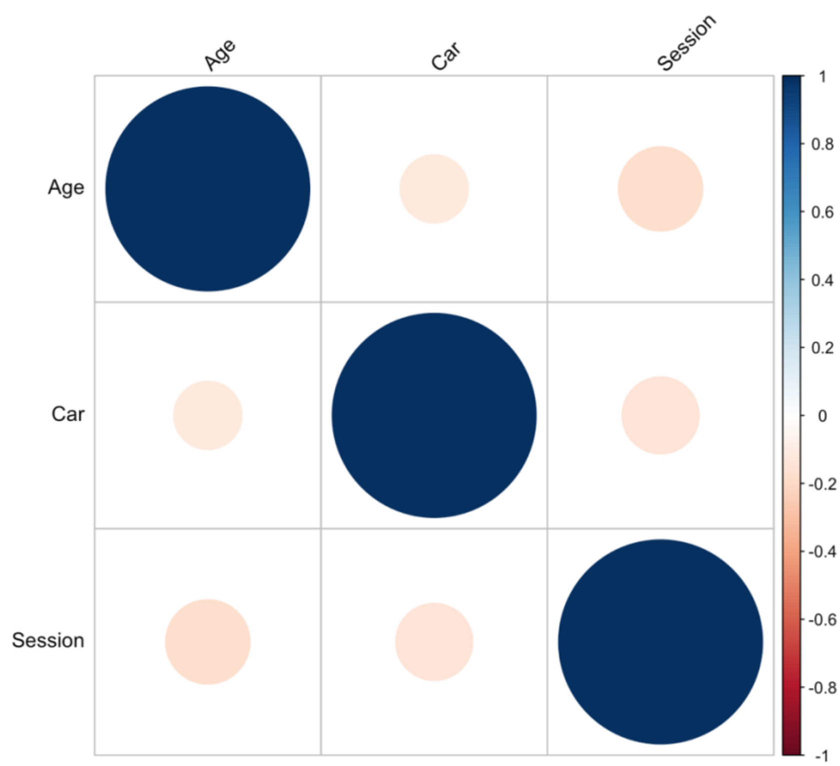

Figure 4 Correlation Matrix Fatal Accidents

.

Predicting: At what stage will next accident occur?

In Formula 1's history, 51 drivers have lost their lives. The last fatal accident occurred in 2014 at the Japanese Grand Prix, with Jules Bianchi, a young driver driving for the team Marussia. The one before occurred in 2012, and the one before that in 2002. Most fatal accidents happened in the first 35 years on an almost yearly basis. Most teams who have lost one of their drivers are not active today due to becoming bankrupt shortly after incidents. Due to this, the variable of Session type is the most important to consider going forward, instead of the teams ('Car') . As mentioned, the session vector within the data frame was converted to a factor with the levels shown below in *table 1*.

Table 1 Session Type Factor Levels

Level	1	2	3	4	5	6
Name	Demon-stration	Practice	Pre-Race Test	Qualifying	Race	Test

In order to visualize how many accidents have occurred within the first years of Formula One as a motor sport *figure 5* was created which can be seen below.

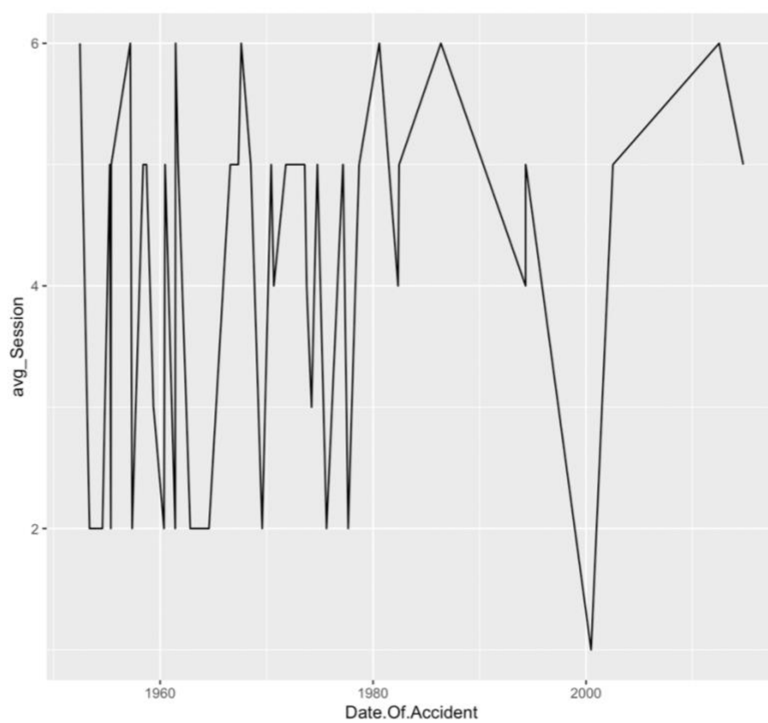

Figure 5 Date of Accident vs Session type

To continue the prediction in what Session type the next fatal accident would occur, an adf test was conducted, which yielded a p value below 0.05, meaning the data is stationary, which is then followed by creating an ARMA model by using ACF (*figure 6*) as well as PACF (*figure 7*).

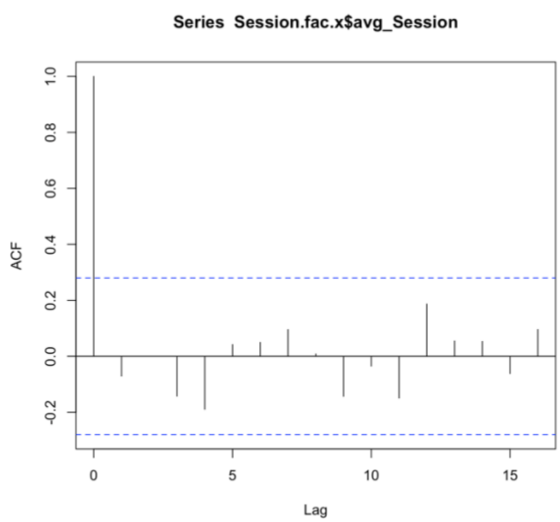

Figure 6 ACF Session Type

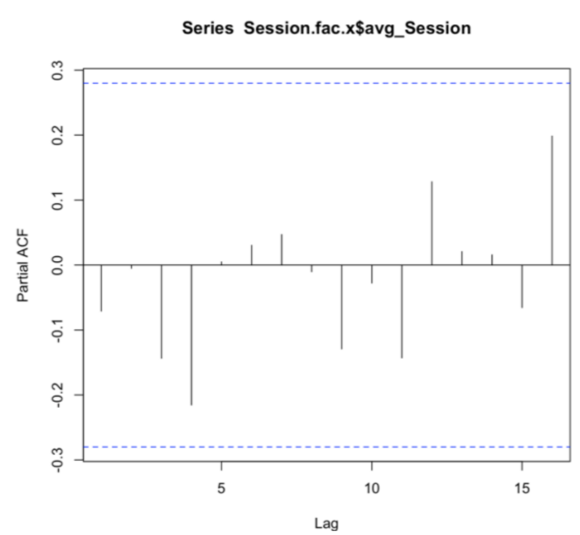

Figure 7 PACF Session Type

After creating an ARMA Model, it predicted that these will be the sessions in which a fatal accident could occur:

1. 5
2. 5
3. 5
4. 5
5. 5

The number 5 is related to the factor level of the Session type. Level 5 is related to the name 'Race', therefore the model is predicting that the next fatal accidents will all occur during an active Race. As this is the most dangerous part of Formula 1, this data output can be considered as an accurate prediction, allthough the volatility is relatively high:

1. 2.222049 – 222%
2. 3.142451 – 314%
3. 3.848701 – 385%
4. 4.444097 – 497%
5. 4.968652 – 497%

The explanation for this can be that there needs to be more data as there are only 51 occurrences and only three in the last 20 years, which on the other hand, can also be considered a business success because the sport has become safer over time.

To put into perspective that the model accurately predicted the occurrence of an incident during an active race, it is helpful to look at the frequency of accidents that have already occurred during a race, which can be seen below in *figure 8*. It is very clear that almost half of the fatal accidents took place during a race, therefore making the ARMA model highly accurate.

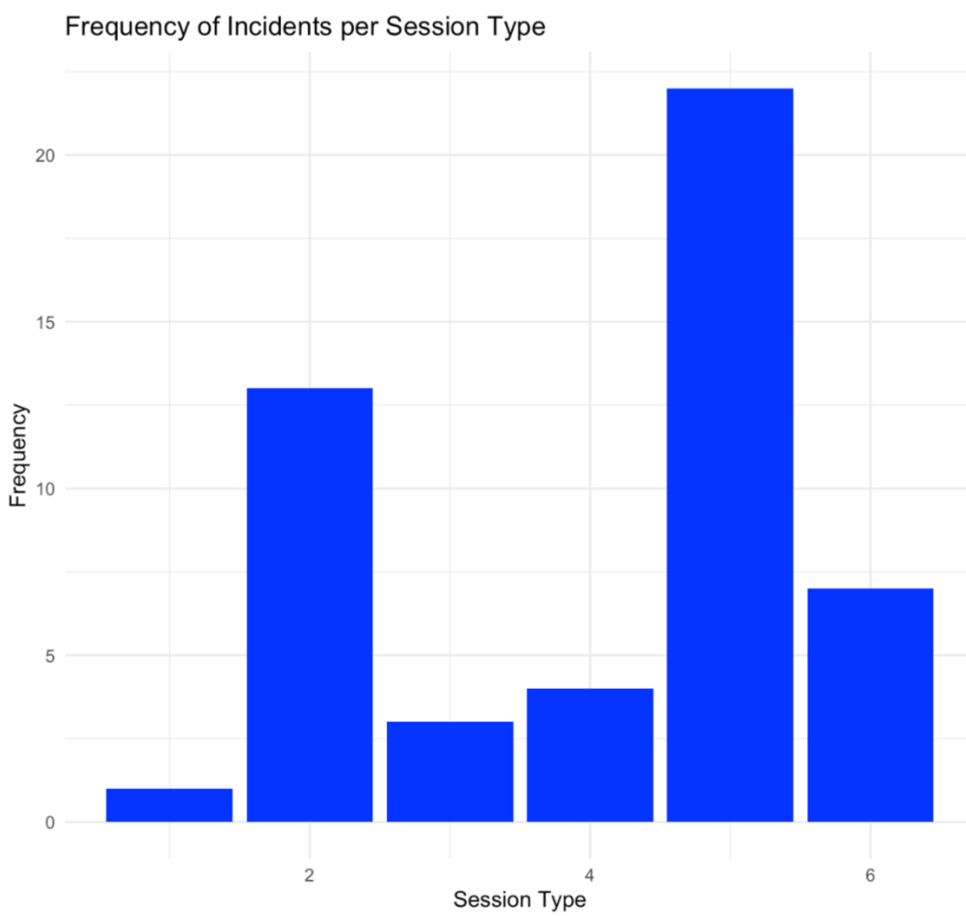

Figure 8 Frequency of Occurance of Session Type

Conclusion

To summarize, a Formula One driver's performance is highly tied to the speed and capabilities of their car. When a team creates a fast chassis during a race, the driver's chances of carrying a rapid pace throughout the race improve dramatically, enhancing their chances of winning a better pole position and establishing a new race winner. As the forecasting model showed, having a fast pace during a race is one of the most significant factors that make a race winner; this is only possible if this driver has a fast car. Without this speed, a high qualifying position is less likely, reducing the driver's pole rate and minimizing the possibilities of winning a race in their careers.

However, it is critical to recognize the inherent risks of Formula One racing, particularly during the race stage, when accidents are more frequent and are also correctly predicted to appear. This emphasizes the necessity for ongoing development in safety standards to reduce hazards and prioritize the safety of drivers, teams, and fans, not only during a race but at every stage within Formula 1. It is important to note that fewer accidents have occurred in the sport over time, which shows that the correct steps are being made in the right direction for higher safety regulations and safety for the drivers, teams, and also spectators.

By achieving a balance between speed, performance, and safety, Formula One can continue attracting spectators worldwide while emphasizing the well-being of everybody involved.

Appendix

Appendix 1: Race Winner R Code

```
# Race Winners Code
# Alyssa Brunen
# Hult International Business School
# New York Summer 1 2023
# Forecasting and Predicting the future using data class

# Purpose:
# Forecasting what creates a Race Winner in Formula 1
# Why: Team and Driver Selection for my Fanatsy League Team

###############################################################################
##########

####################   IMPORTANT MESSAGE IF RUNNING CODE
################################

#### EACH SECTION HAVE TO RUN SEPERATLY
##################################################

#### This is because a seperate DF was created to filter active drivers ONLY ############
#### When normalizing data that is added to this filtered df       #####################
#### it gives error message as there are 868 obsv. in original DF vs 20 in filtered #####
#### After that, continue as usual
######################################################

###############   END OF IMPORTANT MESSAGE
###############################################

###############################################################################
###########

#bringing in all libraries/ installing packages
#install.packages('corrplot')
library(rpart)
library(ggplot2)
library(lattice)
library(rpart.plot)
library(caret)
library(tidyverse)
library(tseries)
library(rugarch)
library(dplyr)
```

```
library(corrplot)

#Loading the dataset, attached to submission

F1_df <-read.csv("/Users/alyssabrunen/Desktop/Hult Docs/Summer 1
NY/Forecasting/Individual submission/F1DriversDataset.csv")

# Massaging data

# Dataframe only with active drivers
F1_df$binary <- gsub("True", 1, F1_df$Active)
F1_df$binary <- gsub("False", 0, F1_df$binary)
F1_df$binary <- as.numeric(F1_df$binary)



for(i in 1:ncol(F1_df)){
  my_min<- try(min(F1_df[,i],na.rm=TRUE))
  my_max<- try(max(F1_df[,i],na.rm=TRUE))
  my_mean<- try(mean(F1_df[,i],na.rm=TRUE))
  my_sd<- try(sd(F1_df[,i],na.rm=TRUE))
  print(c(my_min,my_max,my_mean,my_sd))
}

#normalizing data

min_max <- function(x){
  normalize <- (x-min(x))/(max(x)-min(x))
  return(normalize)
}

F1_df$Race_Entries_norm <- min_max(F1_df$Race_Entries)
F1_df$Race_Starts_norm <- min_max(F1_df$Race_Starts)
F1_df$Pole_Positions_norm <- min_max(F1_df$Pole_Positions)
F1_df$Race_Wins_norm <- min_max(F1_df$Race_Wins)
F1_df$Points_norm <- min_max(F1_df$Points)
F1_df$Points_per_Entry_norm <- min_max(F1_df$Points_Per_Entry)
F1_df$Pole_Rate_norm <- min_max(F1_df$Pole_Rate)
F1_df$Podium_Rate_norm <- min_max(F1_df$Podium_Rate)
F1_df$Podiums_norm <- min_max(F1_df$Podiums)
F1_df$FastLap_Rate_norm <- min_max(F1_df$FastLap_Rate)
F1_df$Fastest_Laps_norm <- min_max(F1_df$Fastest_Laps)
F1_df$Win_Rate_norm <- min_max(F1_df$Win_Rate)

#
cor_matrix<- cor(F1_df[,c( "Race_Entries","Race_Starts","Pole_Positions",
```

```
        "Race_Wins","Points","Points_Per_Entry","Pole_Rate", "Podium_Rate",
        "Podiums", "FastLap_Rate", "Fastest_Laps", "Win_Rate")])

corrplot(cor_matrix, method = "circle", tl.col = "black", tl.srt = 45)


################## From here run sections individually ###################


# filter out all non active drivers in a new df
filtered_df <- subset(F1_df, binary != 0)

# Output the filtered data frame
print(filtered_df)

# adding normalized data to the active driver only dataframe
filtered_df$Race_Entries_norm <- min_max(F1_df$Race_Entries)
filtered_df$Race_Starts_norm <- min_max(F1_df$Race_Starts)
filtered_df$Pole_Positions_norm <- min_max(F1_df$Pole_Positions)
filtered_df$Race_Wins_norm <- min_max(F1_df$Race_Wins)
filtered_df$Points_norm <- min_max(F1_df$Points)
filtered_df$Points_per_Entry_norm <- min_max(F1_df$Points_Per_Entry)
filtered_df$Pole_Rate_norm <- min_max(F1_df$Pole_Rate)
filtered_df$Podium_Rate_norm <- min_max(F1_df$Podium_Rate)
filtered_df$Podiums_norm <- min_max(F1_df$Podiums)
filtered_df$FastLap_Rate_norm <- min_max(F1_df$FastLap_Rate)
filtered_df$Fastest_Laps_norm <- min_max(F1_df$Fastest_Laps)
filtered_df$Win_Rate_norm <- min_max(F1_df$Win_Rate)


############## Continue still running each section individually , ignore error#######

#change Race Winners to 1 and 0 (1 True, 0 False)
filtered_df$Race_Winners <- ifelse(filtered_df$Race_Wins_norm > 0, 1, 0)
F1_df$Race_Winners <- ifelse(F1_df$Race_Wins_norm > 0, 1, 0)

############################## NOW ALL CAN BE RUN AGAIN
#################################



# building a decision tree

my_tree <- rpart(Race_Winners ~ Pole_Positions_norm+
            Pole_Rate_norm+Podium_Rate_norm+Fastest_Laps_norm+FastLap_Rate_norm,
           data=F1_df, method="class")
```

```
#Plot the decision tree
rpart.plot(my_tree, box.palette = "Blues")



#Testing Accuracy of tree
my_df_tree_predict <- predict(my_tree, F1_df, type="prob")

confusionMatrix(data = as.factor(as.numeric(my_df_tree_predict[,2]>0.5)) ,
              reference= as.factor(as.numeric(F1_df$Race_Winners)))

#Confusion Matrix and Statistics Output

#            Reference
#Prediction    0   1
#0           749  23
#1             6  90

#Accuracy : 0.9666
#95% CI : (0.9524, 0.9775)
#No Information Rate : 0.8698
#P-Value [Acc > NIR] : < 2.2e-16


# Count the occurrences of the number 1 in the 'Race_Winners' variable of the dataframe
count <- sum(F1_df$Race_Winners == 1, na.rm = TRUE)

# Print the count
print(count)


##################################################
### Following not used in Report ################
##################################################

## Trial of forecasting 3 variables with higer influence on Race Wins
## Podium_Rate, Fastest_Laps and Pole_Rate

#Group by variable by year
Podium_Rate <- F1_df %>%
  group_by(Decade) %>%
  summarize(avg_Podium_Rate = mean(Podium_Rate, na.rm = TRUE))

Fastest_Laps <- F1_df%>%
  group_by(Decade) %>%
```

```
  summarize(avg_Fastest_Laps = mean(Fastest_Laps, na.rm = TRUE))

Pole_Rate <- F1_df %>%
  group_by(Decade) %>%
  summarize(avg_Pole_Rate = mean(Pole_Rate, na.rm = TRUE))

# Plot each Data
ggplot(data=Podium_Rate)+
  geom_line(aes(x=Decade, y=avg_Podium_Rate))

ggplot(data=Fastest_Laps)+
  geom_line(aes(x=Decade, y=avg_Fastest_Laps))

ggplot(data=Pole_Rate)+
  geom_line(aes(x=Decade, y=avg_Pole_Rate))


#adf test for each variable
adf.test(Podium_Rate$avg_Podium_Rate)

adf.test(Fastest_Laps$avg_Fastest_Laps)

adf.test(Pole_Rate$avg_Pole_Rate)
# All variables hat p-values above .05, we accept the null hypothesis
#time series is non-stationary
#time-dependent structure & does not have constant variance over time

# Decomposition of the non-stationary data
ts_Podi <- ts(Podium_Rate[,c("Decade", "avg_Podium_Rate")],frequency = 5, start=c(1950))
dec_Podi <- decompose(ts_Podi)
plot(dec_Podi)


ts_FL <- ts(Fastest_Laps[,c("Decade", "avg_Fastest_Laps")], frequency = 5, start=c(1950))
dec_FL <- decompose(ts_FL)
plot(dec_FL)

ts_PL <- ts(Pole_Rate[,c("Decade", "avg_Pole_Rate")], frequency = 5, start=c(1950))
dec_PL <- decompose(ts_PL)
plot(dec_PL)

#####
# All three variables were stationary

#acf and pacf
acf(Podium_Rate$avg_Podium_Rate)
```

```
acf(Fastest_Laps$avg_Fastest_Laps)

acf(Pole_Rate$avg_Pole_Rate)

pacf(Podium_Rate$avg_Podium_Rate)

pacf(Fastest_Laps$avg_Fastest_Laps)

pacf(Pole_Rate$avg_Pole_Rate)
```

Appendix 2: Race Accidents R Code

```
# Accidents Code
# Alyssa Brunen
# Hult International Business School
# New York Summer 1 2023
# Forecasting and Predicting the future using data class



# Purpose:
# Predicting the next stage of a Formula 1 Session where a fatal accident will occur
# Why: Safety for drivers, safety regulation changes if necessary

###################################

#Submission Code:


#Load Libraries
library(rpart)
library(ggplot2)
library(lattice)
library(rpart.plot)
library(caret)
library(tidyverse)
library(tseries)
library(rugarch)
library(dplyr)

#Load DataFrame that is being used
Acci_df <-read.csv("/Users/alyssabrunen/Desktop/Hult Docs/Summer 1
NY/Forecasting/Individual submission/fatal_accidents_drivers.csv")

# Changing ages from integer into numeric
Acci_df$Age<- as.numeric(Acci_df$Age)
# Chaning dates from character into date format
Acci_df$Date.Of.Accident<- as.Date(Acci_df$Date.Of.Accident, format= "%m/%d/%Y")

#################################################################
##################   Not Used in Report    #####################
#################################################################
#Personal Trial: Seeing what ages that had fatal accidents were
Age <- Acci_df %>%
  group_by(Date.Of.Accident) %>%
  summarize(avg_Age = mean(Age, na.rm = TRUE))
```

```
#plotting the data
ggplot(data=Age)+
  geom_line(aes(x=Date.Of.Accident, y=avg_Age))
# Personal trial if p value below 0.05 (True)
adf.test(Age$avg_Age)

acf(Age$avg_Age)

pacf(Age$avg_Age)
# Predicting the Age of the next driver with a fatal accident
Age_arima <- arima(Age$avg_Age,
                   order=c(0,1,0))
predict(Age_arima, n.ahead =5)

# Result:
#$pred
#Time Series:
#Start = 50
#End = 54
#Frequency = 1
#[1] 25 25 25 25 25

#$se
#Time Series:
#Start = 50
#End = 54
#Frequency = 1
#[1]  9.506577 13.444330 16.465874 19.013153 21.257352

################################################################
##################  Used in Report  #######################
##########   Car (Team) not used in Report  ################
################################################################
#Setting Car (Team) and Session type as factors and numeric in different variables
Acci_df$Car.fac <- as.factor(Acci_df$Car)
Acci_df$Car.fac.x <- as.numeric(Acci_df$Car.fac)
Acci_df$Session.fac <- as.factor(Acci_df$Session)
Acci_df$Session.fac.x <- as.numeric(Acci_df$Session.fac)

#Printing the level names for both Car (Team) and also Session Type
print(Acci_df$Session.fac)
print(Acci_df$Car.fac)

#Creating Correlation Matrix
cor_matrix <- cor(Acci_df[,c("Age", "Car.fac.x", "Session.fac.x"
)])
```

```
corrplot(cor_matrix, method = "circle", tl.col = "black", tl.srt = 45)


##############Prediction for Session
#Grouping session type by date
Session.fac.x <- Acci_df %>%
  group_by(Date.Of.Accident) %>%
  summarize(avg_Session = mean(Session.fac.x, na.rm = TRUE))

#Plotting session type by date
ggplot(data=Session.fac.x)+
  geom_line(aes(x=Date.Of.Accident, y=avg_Session))

#ADF test to see if Data is Stationary for Session Type
adf.test(Session.fac.x$avg_Session)
#p value below 0.05 = Data stationary

# ACF and PACF for Session
acf(Session.fac.x$avg_Session)

pacf(Session.fac.x$avg_Session)

# Creating Time Series & Plot
ts_Session <- ts(Session.fac.x[,c("Date.Of.Accident", "avg_Session")],frequency = 5,
start=c(1950))
dec_Session <- decompose(ts_Session)
plot(dec_Session)

#Creating ARMA Model for Stationary Data

Session_arma <- arma(Session.fac.x$avg_Session, order=c(0,0))
summary(Session_arma)
#however, to use the predict() function we need to use the arima function
# we use a 0 for i as a forecast
Session_arima <- arima(Session.fac.x$avg_Session,
                       order=c(0,1,0))
predict(Session_arima, n.ahead =5)

#ARMA Model output
#$pred
#Time Series:
#Start = 50
#End = 54
#Frequency = 1
#[1] 5 5 5 5 5
```

```
#$se
#Time Series:
#Start = 50
#End = 54
#Frequency = 1
#[1] 2.222049 3.142451 3.848701 4.444097 4.968652


##################################################
###### Prediction for Car (Team) ###############
#############  Not used in Report ##############
# Doing the same for car (Team)
Car.fac.x <- Acci_df %>%
  group_by(Date.Of.Accident) %>%
  summarize(avg_Car = mean(Car.fac.x, na.rm = TRUE))

ggplot(data=Car.fac.x)+
  geom_line(aes(x=Date.Of.Accident, y=avg_Car))

adf.test(Car.fac.x$avg_Car)
# Adf test p value above 0.05 = Non Stationary data

# Creating Time series for Car(Team)
ts_Car <- ts(Car.fac.x[,c("Date.Of.Accident", "avg_Car")],frequency = 5, start=c(1950))
dec_Car <- decompose(ts_Car)
plot(dec_Car)

#ACF and PACF for Car (team)
acf(Car.fac.x$avg_Car)

pacf(Car.fac.x$avg_Car)

#ARIMA forecasting for Cars (teams)
Car_arima <- arima(Car.fac.x$avg_Car,
                   order=c(1,1,2))
predict(Car_arima, n.ahead =5)

#ARIMA Model Output for Cars (Teams)
#$pred
#Time Series:
#Start = 50
#End = 54
#Frequency = 1
#[1] 14.17018 13.68226 13.42870 13.29692 13.22844

#$se
```

```
#Time Series:
#Start = 50
#End = 54
#Frequency = 1
#[1] 6.516498 6.874050 6.996722 7.047257 7.072723

### Unuseful data for Cars(Teams) as the predicted teams do not exsist anymore
#########   Therefore not used in the report ###############
############################################################


########## Data for visualization in Report
# Counting Race Fatalities
count.Race.Fat <- sum(Acci_df$Session.fac.x == 5, na.rm = TRUE)
# Print the count
print(count.Race.Fat)


#### Visualization of Fatalities per Session Type Code
# Count the frequency of each factor level in session type
frequency <- table(Acci_df$Session.fac.x)

# create a new dataframe for that
df <- data.frame(Number = as.numeric(names(frequency)),
                 Frequency = as.numeric(frequency))

# Create the bar plot
ggplot(df, aes(x = Number, y = Frequency)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(x = "Session Type", y = "Frequency", title = "Frequency of Incidents per Session Type") +
  theme_minimal()
```