STAT 654 Statistical Computing with R and Python

Formula 1 Analysis & Prediction

Project Report

Prof.
Sharmistha Guha

Abdullatif AlNuaimi
(group 6)

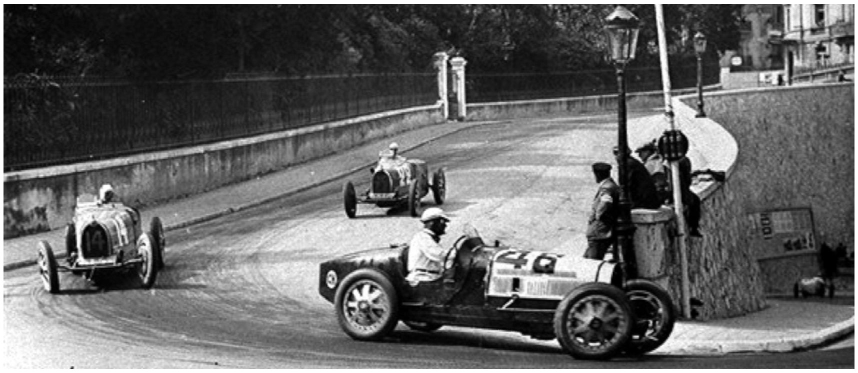
alnuaimi107@tamu.edu

731009420

Formula 1 History:

Formula 1 racing originated during the 1920-30s in Europe from other similar racing competitions. In 1946, the FIA standardized racing rules and this formed the basis of Formula One racing. The inaugural Formula One World Drivers' championship was then held in 1950, the first world championship series.

Apart from the world championship series, many other non-championship F1 races were also held, but as the costs of conducting these contests got higher, such races were discontinued after 1983.



What is Formula 1:

Formula 1, also known as F1, is the highest class of international racing for single-seater formula racing cars.

- The word formula in the name refers to the set of rules to which all participants' cars must conform.

- It is governed by the FIA — Fédération Internationale de l'Automobile (the International Automobile Federation).

Formula 1 Season:

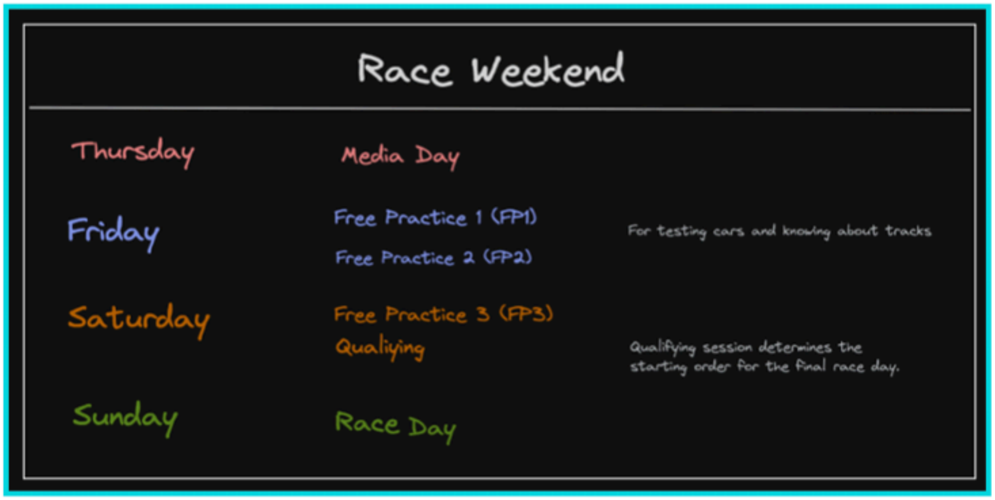
A Formula One season consists of a series of races, known as Grands Prix, which take place worldwide on both:

1. Purpose-built circuits
2. Closed public roads

The F1 season usually starts in March and ends in December. The latest Grands Prix races conducted in Saudi Arabia, Australia, and Italy.

Terminologies:

- **Teams & drivers**: There are 10 teams consisting of 20 drivers compete every season
- **Circuits**: Tracks that are built for conducting the races
- **Lab**: One complete circuit in a race, i.e., one trip around the entire track
- **Race Weekend**: Grand Prix takes place over a weekend, and the various sessions are distributed as follows:



Race Weekend

Thursday	Media Day	
Friday	Free Practice 1 (FP1) Free Practice 2 (FP2)	For testing cars and knowing about tracks
Saturday	Free Practice 3 (FP3) Qualifying	Qualifying session determines the starting order for the final race day.
Sunday	Race Day	

Formula 1 Championship Awards:

All the teams and drivers compete every season for two Championship awards based on the points scored in the season.

1. Drivers' Championship Award
2. Constructors' Championship Award (company that owns the intellectual right of the car's engine and chassis)

Winning Criteria:

F1 is a points-based competition so a driver or constructor with the most points at the end of a season is awarded the F1 Championship Award

General Analysis:

In our dataset, we have considered 2012-2017 data as our scope of analysis. We did general analysis that focus on drivers' pattern and the lab environments, so we have done an extensive data analysis on the underneath data and its interaction with all other variables

Driver Analysis:

To find out the performance of the drivers, we started our data exploration from the number of winnings to set up a baseline from where we would get into the analysis.

Name of the driver	Winnings
Lewis Hamilton	45
Sebastian Vettel	26
Nico Rosberg	23
Fernando Alonso	5
Daniel Ricciardo	5

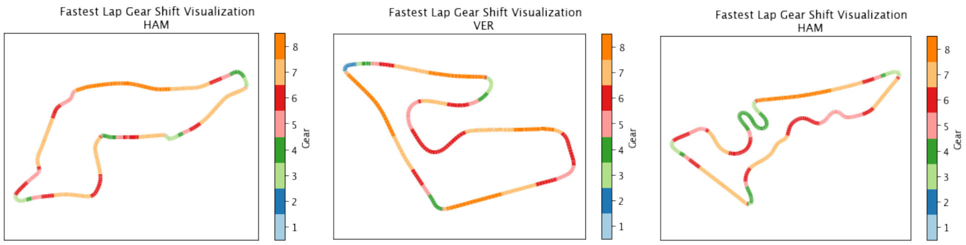
Winning counts

We clearly found that the first three drivers have the highest points count so we can focus more on them and see what their techniques in driving are. To do that we continue to analyze more data for three aspects including:

1. Pattern of driving
2. Fastest Speed
3. Lab time

Pattern of driving:

In the gear shift plot below, we observed there is certainly some difference in the way of driving of the drivers which is more correlated to the circuit



Gear shift in circuits

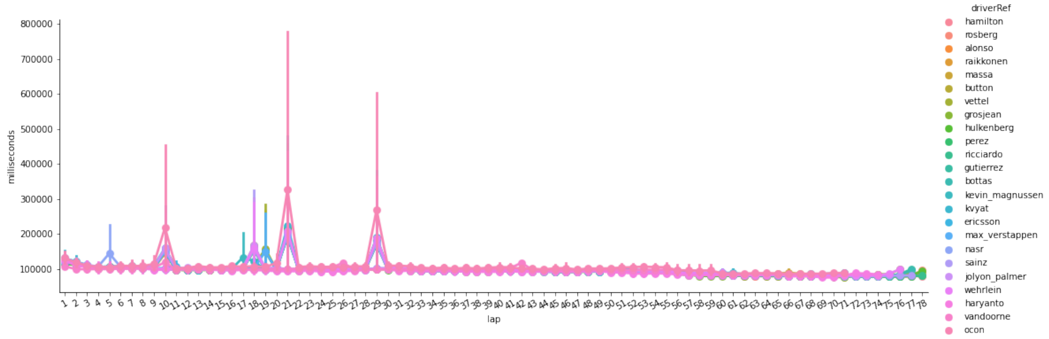
Fastest speed:

Also, the below table will tell us more, the speed of the lap, we found out the highest speed and lowest lap time for top 3 selected drivers. It is clearly shown that there are not much difference between the highest speed so we need to do more closer look into lap time and lap/lap plots.

Name	Most won constructor	Highest lap speed	Circuit
Lewis Hamilton	Mercedes	249.73 m/h	Italian GP 2017
Sebastian Vettel	Red Bull-Ferrari	239.18 m/h	Italian GP 2013
Nico Rosberg	Mercedes	240.82 m/h	Italian GP 2016

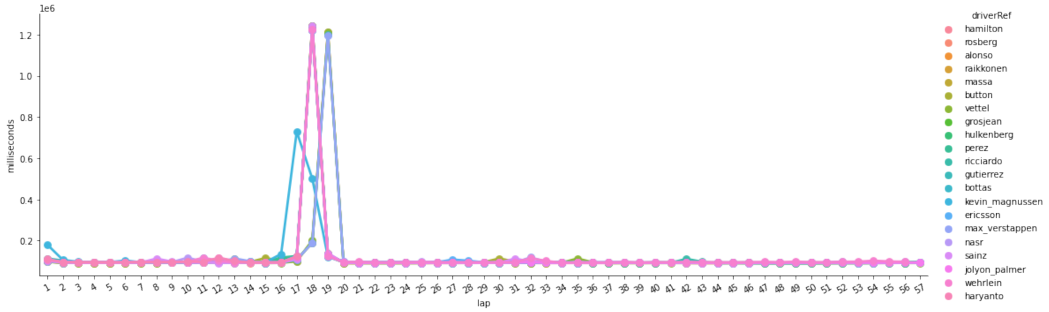
Lap time:

At first, we have a look at the overall yearly all season’s lap by lap analysis of the drivers and how much their patterns vary.



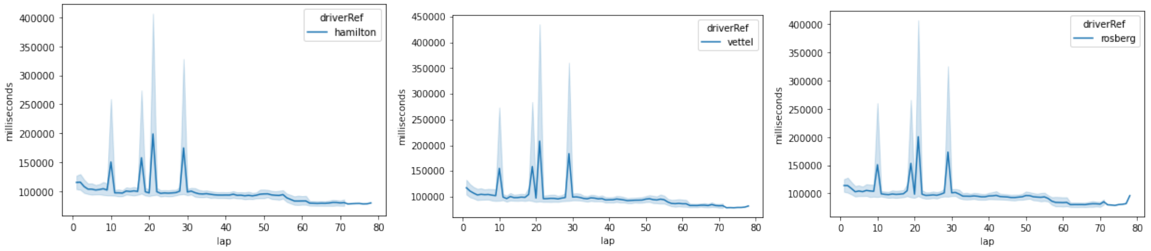
Lap vs lap times

In lap vs lap times, we saw some certain variation in few places over the period of time, which is not very clear as the number of data points are more. Another thought that we tried to look with individual instances of the seasonal record and their lap times.



Australia GP 2016

After fetching multiple instances, we observed multiple drivers has almost similar driving speed and lap completion time. So, pivoted the direction towards individual lap time in driving to see the variance within them.

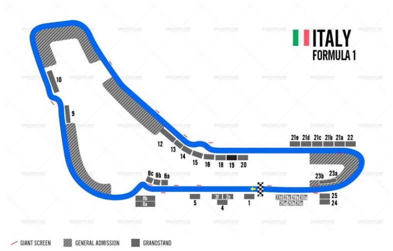


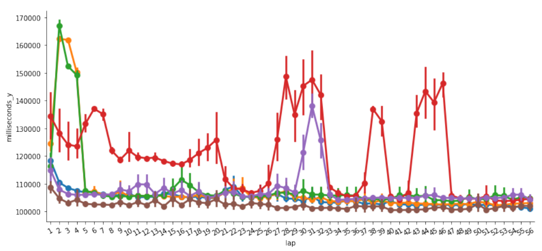
Individual driver lap completion

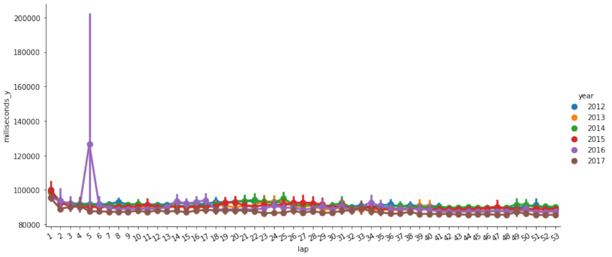
In individual lap time we observed that the driver whose rank high, his lap time goes to 400000ms. on the other hand, the non-winning or lower position drivers will cross 1000000ms, so we found an import correlation between lap timing and driver speed.

Lap Analysis:

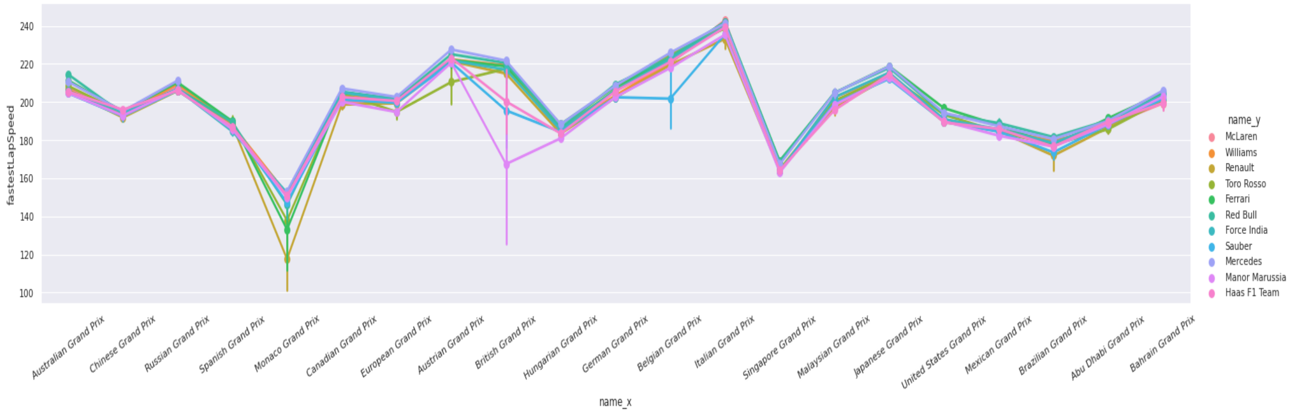
As we found previously, lap plays an essential role during the points as well as the position of the driver, so we tried to find out the complexity of the circuits and its implication on driver's speed. To proceed with this idea, we did some visual analysis on the type of circuits used during multiple seasons.







As we can see, there is huge dependency of speed change with the complexity of the circuit. With the complexity of the circuit, the speed and time taken to complete the lap is going to change which will be helpful to predict the position of the driver if he will be driving in certain circuits. As we saw earlier the highest speed was recorded in Italian Monza circuit because it has less curves and might be some good longitude and latitude. Similarly, we can see the same result from constructor perspective in terms of performance.



It can be clearly seen there is a stable pattern for many circuits where the speed of multiple constructers is approximately the same.

Regression Analysis:

Based on the general analysis and what we’ve learned so far about F1, we needed to decide what type of analysis would help predict who will win any given race. Discussing that with Ben, we came across the idea of fitting a model to certain features of the data that could potentially help us predict the position that a driver will have at the end of any given race.

Manipulating data set

The below code showing how we upload the data, setting up the structure of the data types, dividing data into multiples data frame based on years, splitting each data frame into training and test set.

```r
library(hms)
library(dplyr)
library(nnet)
data_new <- read.csv("/Users/benklein18/Desktop/STAT 654/df_result_race_driver_const.csv", header = TRUE)


attach(data_new)


data_new <- data_new[order(data_new$year, decreasing = FALSE),]
data_new <- data_new[-c(7,8)]
data_new <- data_new[-c(21:31)]


data_new$circuitId <- as.factor(data_new$circuitId)
data_new$driverId <- as.factor(data_new$driverId)
data_new$positionOrder <- as.factor(data_new$positionOrder)
data_new$fastestLapTime <- as_hms(strptime(data_new$fastestLapTime, "%H:%M.%S"))


data_2012 <- data_new[which(data_new$year == 2012),]
data_2013 <- data_new[which(data_new$year == 2013),]
data_2014 <- data_new[which(data_new$year == 2014),]
data_2015 <- data_new[which(data_new$year == 2015),]
data_2016 <- data_new[which(data_new$year == 2016),]
data_2017 <- data_new[which(data_new$year == 2017),]
```

```
train_2012 <- sample_frac(data_2012, .75)
sample_id <- as.numeric(rownames(train_2012))
test_2012 <- data_2012[-sample_id,]

train_2013 <- sample_frac(data_2013, .75)
sample_id <- as.numeric(rownames(train_2013))
test_2013 <- data_2013[-sample_id,]

train_2014 <- sample_frac(data_2014, .75)
sample_id <- as.numeric(rownames(train_2014))
test_2014 <- data_2014[-sample_id,]

train_2015 <- sample_frac(data_2015, .75)
sample_id <- as.numeric(rownames(train_2015))
test_2015 <- data_2015[-sample_id,]

train_2016 <- sample_frac(data_2016, .75)
sample_id <- as.numeric(rownames(train_2016))
test_2016 <- data_2016[-sample_id,]

train_2017 <- sample_frac(data_2017, .75)
sample_id <- as.numeric(rownames(train_2017))
test_2017 <- data_2017[-sample_id,]
```

Choosing the model

In order to find the best fit model, we need to test out models that actually fall under the criteria for our data.

1. Generalized linear model (GLM), a regular linear model
2. Multinomial logistic regression model

The reason for the lack of models to test is due to the type of response we have, multi-categorical variables that aren't 0/1 are a lot harder to fit on with common models used.

Model Diagnostic

The data that will be analyzed from year 2012 – 2017 on a yearly basis since each year some rules and regulations might changed in F1 also the type of technologies and circuits might be different. The below code only an example of 2012 data analysis and model fit.

Model Parameters

- **The data set**: 2012
- **Training and testing split**: 75/25
- **Features**: fastestLapTime, circuitId, driverId, laps (points was added then excluded due to collinearity)
- **Response**: positionOrder
- **Model selecting criteria**: AIC, train/test accuracy

Model Fit:

The below is the R code that we fit our two models which are glm and multinomial.

```
data_new_train_2012 <- sample_frac(data_2012, .75)

sample_id <- as.numeric(rownames(data_new_train_2012))

data_new_test_2012 <- data_2012[-sample_id,]

f1_12_glm_model <- glm(as.numeric(positionOrder) ~ fastestLapTime + circuitId
+ driverId + laps, data = data_new_train_2012)

f1_2012_model <- multinom(positionOrder ~ fastestLapTime + circuitId + driver
Id + laps, data = train_2012, maxit = 500, MaxNWts = 10000000)
```

Model Selection:

We use AIC and accuracy score to select the best model to use. The following code shows that the multinomial got highest AIC and also highest accuracy score between the predicted and both training & test data.

AIC score for glm model:

```
AIC(f1_12_glm_model)

## [1] 1960.414
```

AIC score for multinomial model:

```
extractAIC(f1_2012_model)

## [1] 2644.618
```

Accuracy score for glm model on training data:

```
data_new_train_2012$predicted <- predict(f1_12_glm_model, newdata = data_new_
train_2012, type = "response")

data_new_train_2012$predicted <- round(data_new_train_2012$predicted, digits
= 0)

ctable_2012 <- table(as.character(data_new_train_2012$positionOrder) == as.ch
aracter(data_new_train_2012$predicted))

round(ctable_2012[[2]]/sum(ctable_2012)*100, 2)
```

```
## [1] 11.92
```

Accuracy score for multinomial model on training data:

```
library(ggplot2)

train_2012$predicted <- predict(f1_2012_model, newdata = train_2012, type = "
class")

ctable_2012 <- table(as.character(train_2012$positionOrder) == as.character(t
rain_2012$predicted))

round(ctable_2012[[2]]/sum(ctable_2012)*100, 2)
```

```
## [1] 68.13
```

Accuracy score for glm model on test data:

```
data_new_test_2012$predicted <- predict(f1_12_glm_model, newdata = data_new_t
est_2012, type = "response")

data_new_train_2012$predicted <- round(data_new_train_2012$predicted, digits
= 0)

ctable_2012_test <- table(as.character(data_new_test_2012$positionOrder) == a
s.character(data_new_test_2012$predicted))

round(ctable_2012_test[[2]]/sum(ctable_2012_test)*100, 2)
```

```
## [1] 0.87
```

Accuracy score for multinomial model on test data:

```
test_2012$predicted <- predict(f1_2012_model, newdata = test_2012, type = "cl
ass")

ctable_2012_test <- table(as.character(test_2012$positionOrder) == as.charact
er(test_2012$predicted))

round(ctable_2012_test[[2]]/sum(ctable_2012_test)*100, 2)
```

```
## [1] 66.96
```

Model Assumption:

Since we choose to implement the multinomial model, the most important assumptions here to consider is:

1. **Independence**: each of the observations (data points) should be independent. This means that each value of the variables doesn't "depend" on any of the others. We know from fact that the data is independent.

2. **Multicollinearity**: refers to the scenario when two or more of the independent variables are substantially correlated amongst each other. The following code satisfy this assumption where there is no multicollinearity in the data.

```
t_12 <- train_2012[ ,c(3, 7, 9, 14, 17)]

t_12$driverId <- as.integer(t_12$driverId)

t_12$circuitId <- as.integer(t_12$circuitId)

t_12$fastestLapTime <- as.integer(t_12$fastestLapTime)

t_12$positionOrder <- as.integer(t_12$positionOrder)

cor(t_12)
```

```
##                   driverId positionOrder        laps fastestLapTime   circ
uitId

## driverId         1.00000000     0.2682764 -0.08881569             NA -0.029
13758

## positionOrder    0.26827639     1.0000000 -0.53646162             NA -0.034
32690

## laps            -0.08881569    -0.5364616  1.00000000             NA -0.050
71052

## fastestLapTime           NA            NA          NA              1
NA

## circuitId       -0.02913758    -0.0343269 -0.05071052             NA  1.000
00000
```

Model Accuracy Result:

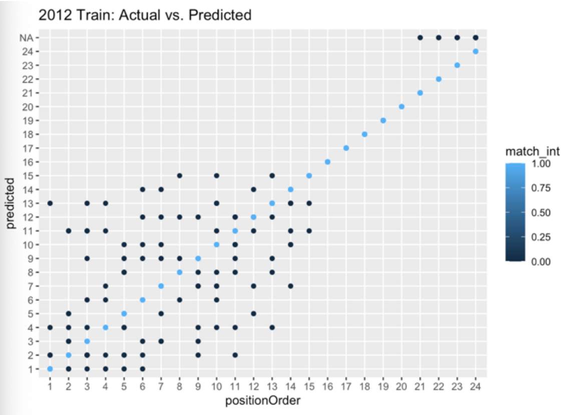
As we see earlier the accuracy, we got from using multinomial model are %68.13 on training and %66.96 on test set. Next, a visualization of the model accuracy is shown in the below code:

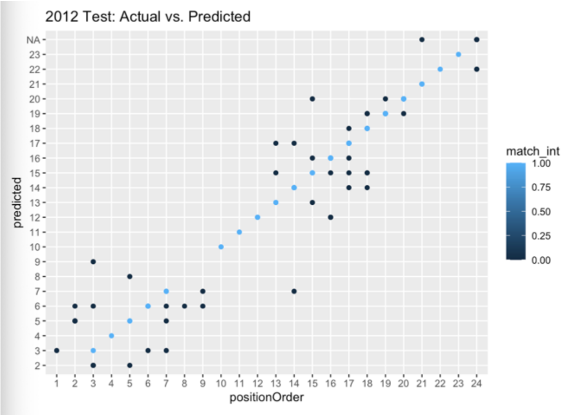
```
pred_pos_2012_train <- train_2012[c(7,22)]
pred_pos_2012_test <- test_2012[c(7,22)]

pred_pos_2012_train$match <- NA
for (i in 1:nrow(pred_pos_2012_train)){
  pred_pos_2012_train$match[i] <- pred_pos_2012_train$positionOrder[i] == pre
d_pos_2012_train$predicted[i]
}
pred_pos_2012_train$match_int <- as.integer(pred_pos_2012_train$match)
pred_pos_2012_train$match_int[is.na(pred_pos_2012_train$match_int)] = 0

pred_pos_2012_test$match <- NA
for (i in 1:nrow(pred_pos_2012_test)){
  pred_pos_2012_test$match[i] <- pred_pos_2012_test$positionOrder[i] == pred_
pos_2012_test$predicted[i]
}
pred_pos_2012_test$match_int <- as.integer(pred_pos_2012_test$match)
pred_pos_2012_test$match_int[is.na(pred_pos_2012_test$match_int)] = 0

par(mfrow = c(2,1))
ggplot(pred_pos_2012_train, aes(x = positionOrder, y = predicted, colour = ma
tch_int)) + geom_point() + ggtitle("2012 Train: Actual vs. Predicted")
ggplot(pred_pos_2012_test, aes(x = positionOrder, y = predicted, colour = mat
ch_int)) + geom_point() + ggtitle("2012 Test: Actual vs. Predicted")
```

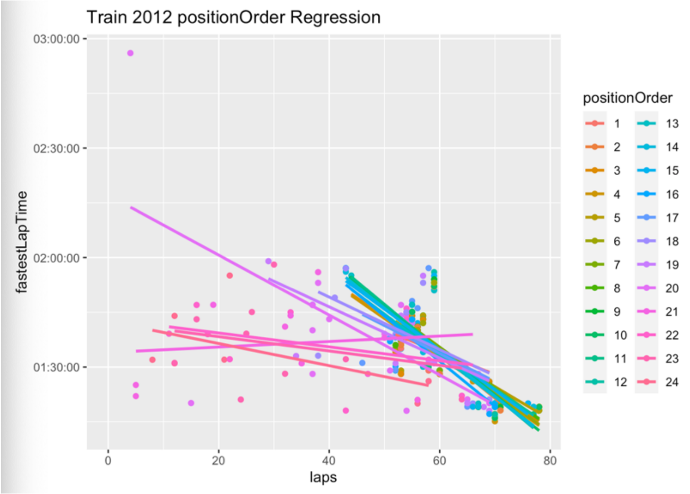




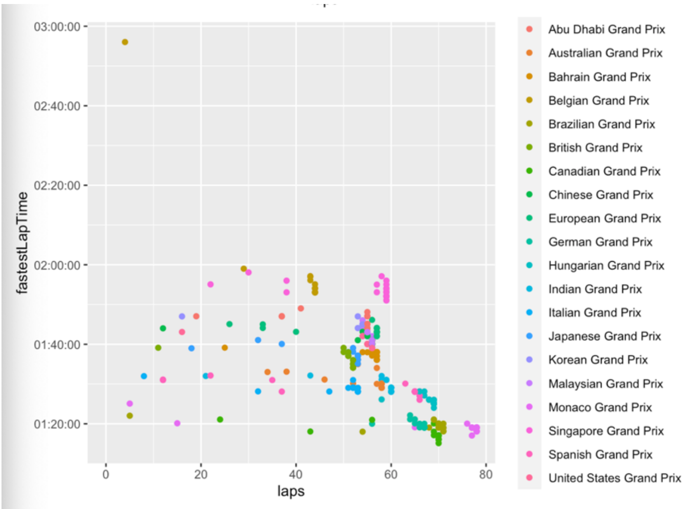
So this is for 2012, a series of other graphs and AICs scores generated in Ben’s report for the following years (2013 – 2017).

What is next / Conclusion:

We did the regression analysis on the data set from positionOrder point of view. We know that we can find many relationships in analyzing more multiple features, but we might face some issues regarding multicollinearity. Two interesting note here are that each driver has a unique relationship with where they will most likely be positioned in any given race given the laps and fastestLapTime.



Also another observation is that when we include more facts about the circuits, we will get a high effect of the circuit environment on the laps and fastestLapTime and consequently will affect the positionOrder as well.


fastestLapTime
03:00:00
02:40:00
02:20:00
02:00:00
01:40:00
01:20:00
laps
Abu Dhabi Grand Prix
Australian Grand Prix
Bahrain Grand Prix
Belgian Grand Prix
Brazilian Grand Prix
British Grand Prix
Canadian Grand Prix
Chinese Grand Prix
European Grand Prix
German Grand Prix
Hungarian Grand Prix
Indian Grand Prix
Italian Grand Prix
Japanese Grand Prix
Korean Grand Prix
Malaysian Grand Prix
Monaco Grand Prix
Singapore Grand Prix
Spanish Grand Prix
United States Grand Prix