Formula 1 Analysis in R

1950 to 2017

Nitesh Kamboj

California State University, Los Angeles

Nitesh Kamboj, College of Business & Economics

Corresponding concerning to this article should be sent to Nitesh Kamboj, College of Business & Economics, California State University, Los Angeles, California 90032

Contact: nkamboj@calstatela.edu

Abstract

This paper is the research work done on Formula 1 racing which is a very famous and expensive sport. 9 major insights have been presented in this paper using the data for 67 year which is broken into the collection of 13 data files each containing different information of Formula 1. The purpose of this paper is to mainly show the hidden insights and very crucial and meaningful information about drivers, manufacturers, and racing circuits using R.

Motivation

We all have seen racing on TV at least once in life (at least sometimes while changing the channels on TV and passing through some sports channel) and some like me were curious to know more about this sport. I was always excited to watch racing and wanted to learn more about it since I was just a kid. My interest and passion to learn more about this field of sports drove me the point in my undergrad in Computer Applications that I decided to pursue my career in it. I entered racing in year 2009 and became sponsored racer in year 2010. Won national championship title in year 2011 and was doing professional X-country rally racing till year 2017 with couple of titles under my name. Later in 2017 I moved to Los Angeles for earning my second master's degree in computer information systems. I am looking forward to aligning my future goals with some racing team or automobile company and learn improved techniques to handle, learn, and project advanced learnings and knowledge using race data.

I found a racing dataset on http://ergast.com/mrd/ but I could use only APIs with GET command to load the data to my file system and which is not required for this course project. Later I found a project done on the same data set from same source on Kaggle and I decided to use it for this paper. Further, I think this dataset offers an exciting insight into a billion-dollar industry, enjoyed by hundreds of millions of viewers all over the world.

The topic "Formula 1 Descriptive Analytics & Visualization" has already been considered by Bouchet, Jonathan. *F1 Data Analysis*. December 2017. In his work which could be found at *https://www.kaggle.com/jonathanbouchet/f1-data-analysis/data* he used R notebook to engineer and visualize the data using many R libraries.

Taking inspiration from Jonathan Bouchet, I have added more insights for more detailed information. I used solely Rstudio to engineer the data and then perform various types of joins to bring different csv file together and answer 9 different questions with the help of visualization. This paper is limited to descriptive analysis since we are using the data from past and not using any predictive models for predictions. Nevertheless, I will try to conclude my understanding with the analysis after finding the answers through visualization.

The datasets consist of 13 related csv files which are being joined differently for various findings. List of files is as below:

1. Circuits: Contains a list of every formula 1 circuit, including name, location and geographic data.
2. ConstructorResults: Details of the results for every race, including race, constructor, and awarded points to all the participating constructor in F1
3. Constructor: Contains a list of every constructor team including name and nationality.
4. ConstructorStandings: Information about points, position, and wins of constructors since the beginning of their career in F1
5. Drivers: Contains a list of every Formula 1 driver, including full name, dob, and nationality.
6. DriverStandings: It is similar to ConstructorStandings and have inofmation about points, position, and wins of drivers
7. LapTimes: Details the lap times for every race, including driver, lap number, position and time

8. Pitstops: Details of every pitstop in Formula 1, including the time of the pit stop, the duration, the race and driver
9. Qualifying: The results of every qualifying session, including the race, driver, constructor, position, and times for Q1, Q2 and Q3.
10. Races: Details of every race, including year, date, time, circuit and round
11. Results: Details of grid, driver number, position, position order, fastest lap, fastest lap time, fastest lap speed.
12. Seasons: A list of every season and corresponding Wikipedia link
13. Status: A table of status codes and their status

These 13 datasets contain very crucial information within them under different features. If these datasets are observed individually then there is not really any good information provided but when you consider joining them together then one can appreciate the valuable information this dataset contains and various types of analysis which could be done on this dataset. Although, for this project I am only doing descriptive analysis using R but I would not miss an opportunity to take a step further and try to do predictive and prescriptive analysis on it.

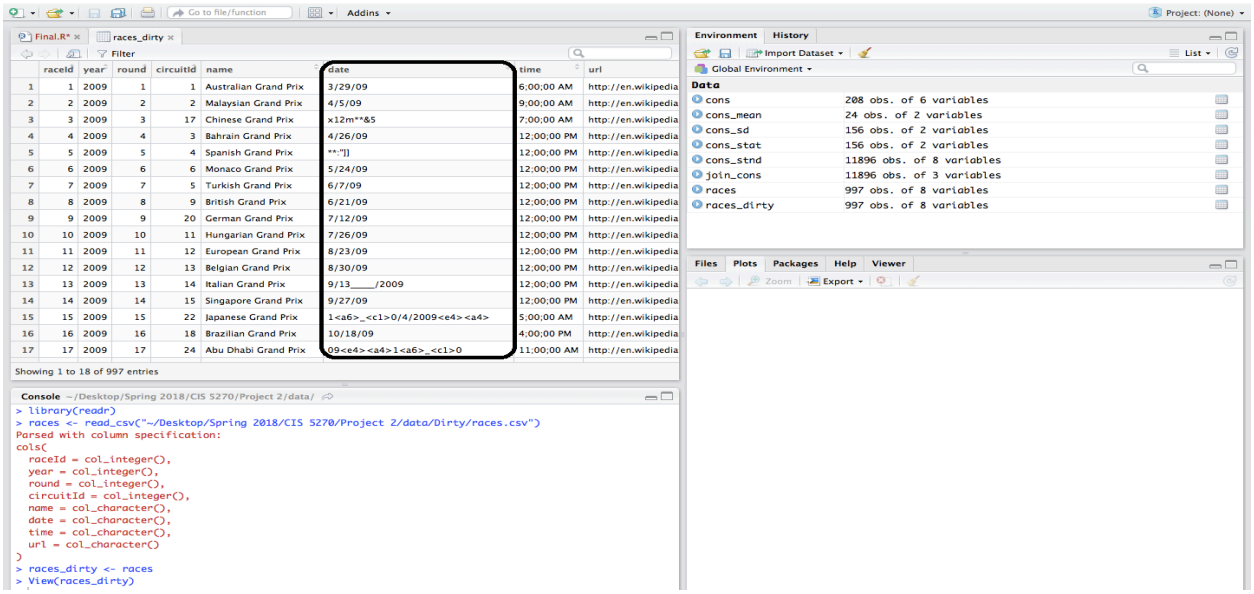
Dataset can be downloaded from http://ergast.com/mrd/db/#csv and *https://www.kaggle.com/jonathanbouchet/f1-data-analysis/data*

Cleaning of Data

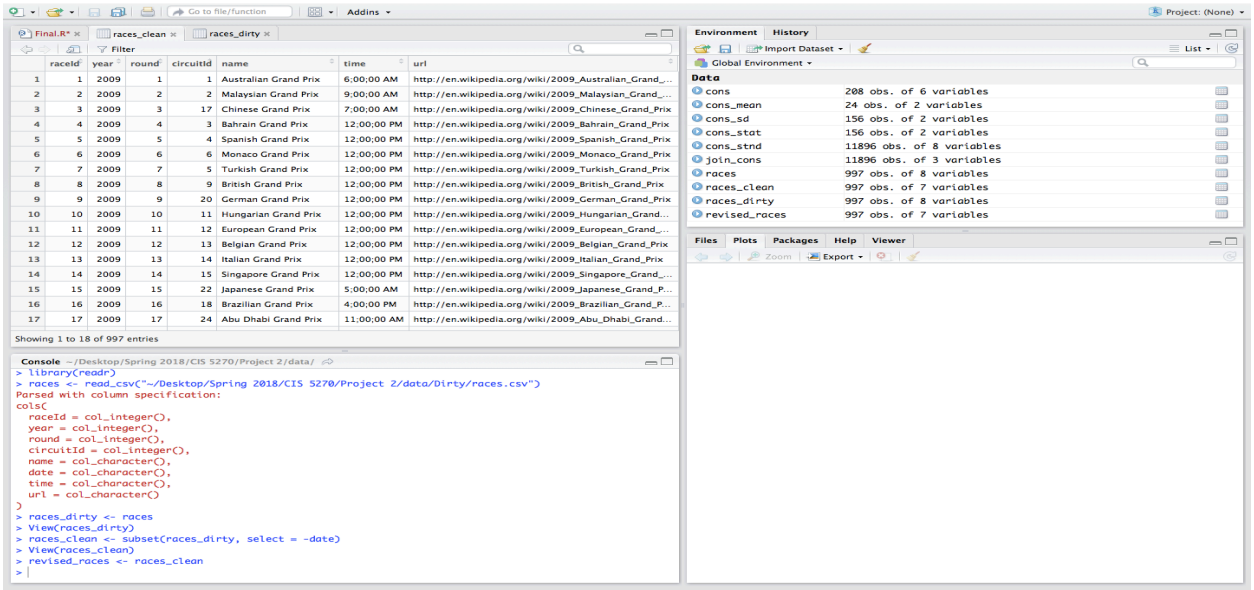
There are total 86 labels in 13 different files and some of it require cleaning in order to be used for the research. Some of the datasets have dirty values which will lead to inefficient analysis. So, using different techniques in R, I will try to clean the data before proceeding with analysis.

Removing invalid column:

Before



After



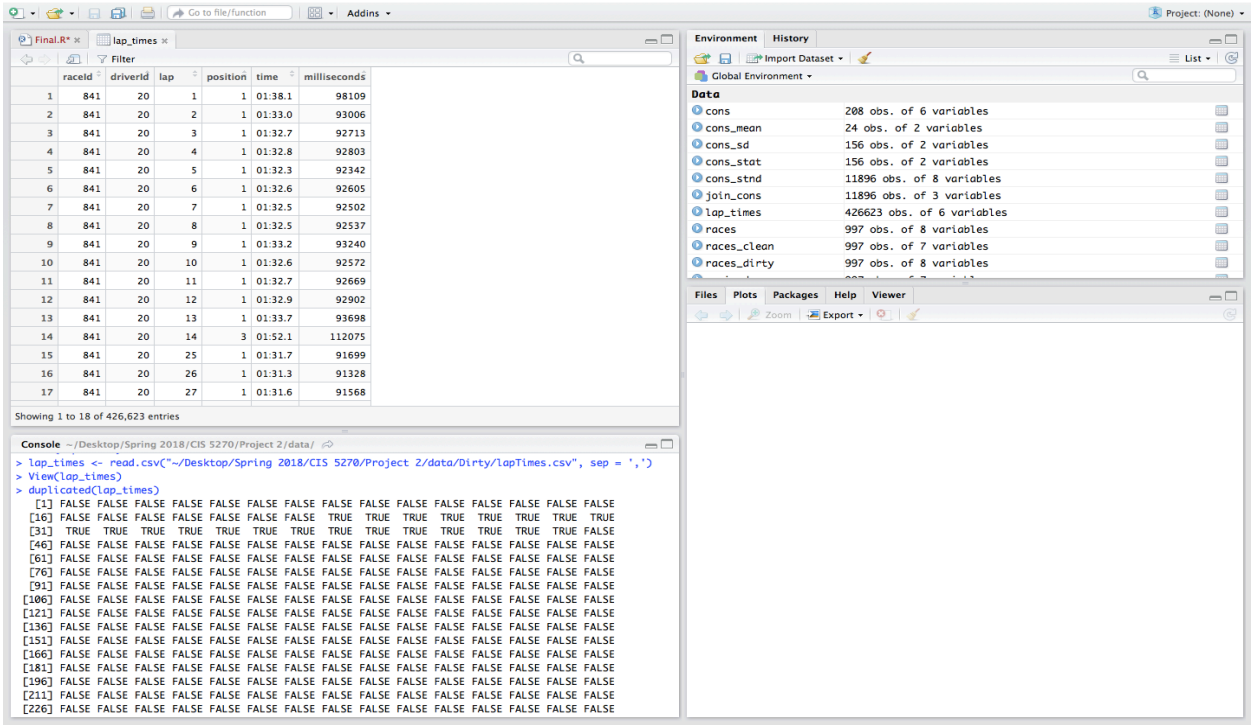
The above figures represent the removal of date column since it had invalid values which was supposed to represent on which date the race was being held.

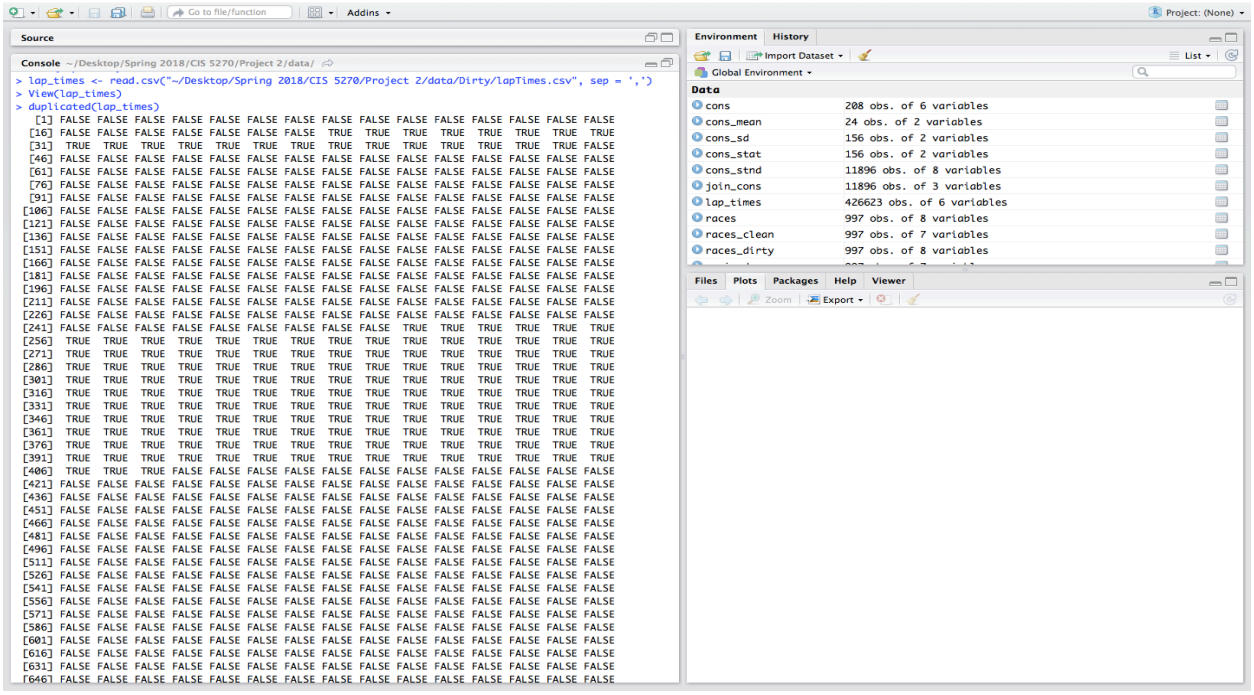
Code:

```
> library(readr)
> races <- read_csv("~/Desktop/Spring 2018/CIS 5270/Project 2/data/Dirty/races.csv")
> races_dirty <- races
> View(races_dirty)
> races_clean <- subset(races_dirty, select = -date)
> View(races_clean)
```

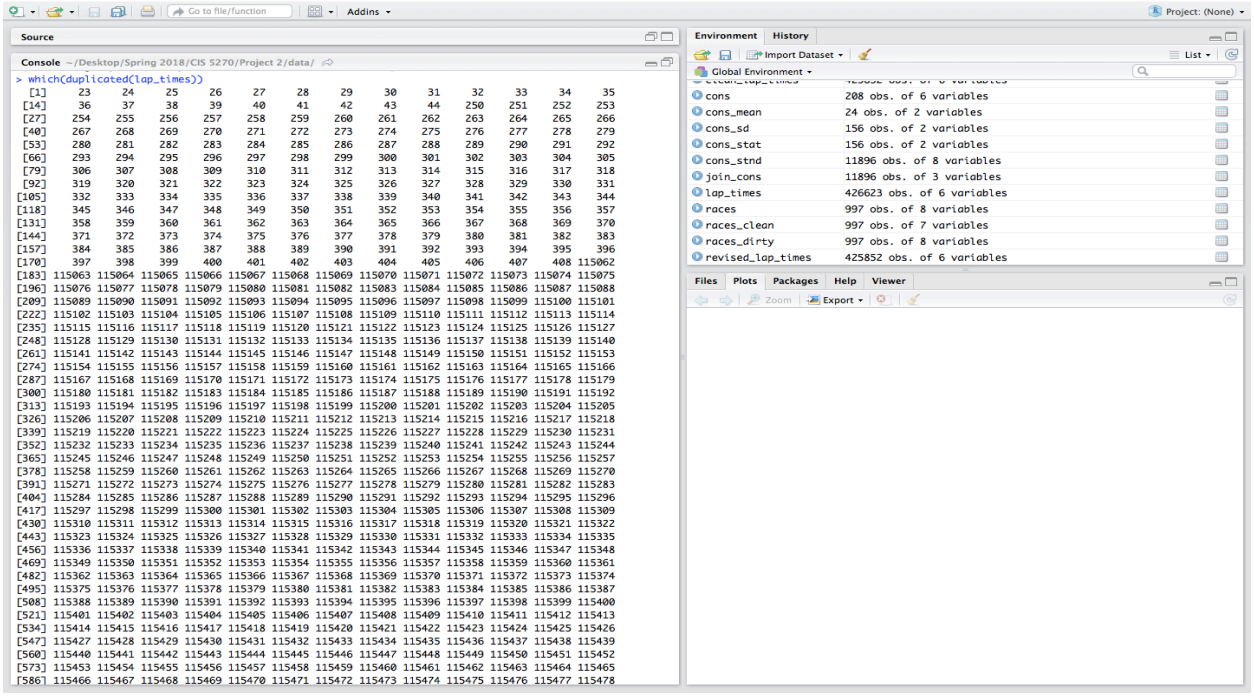
Duplicated records:

Before



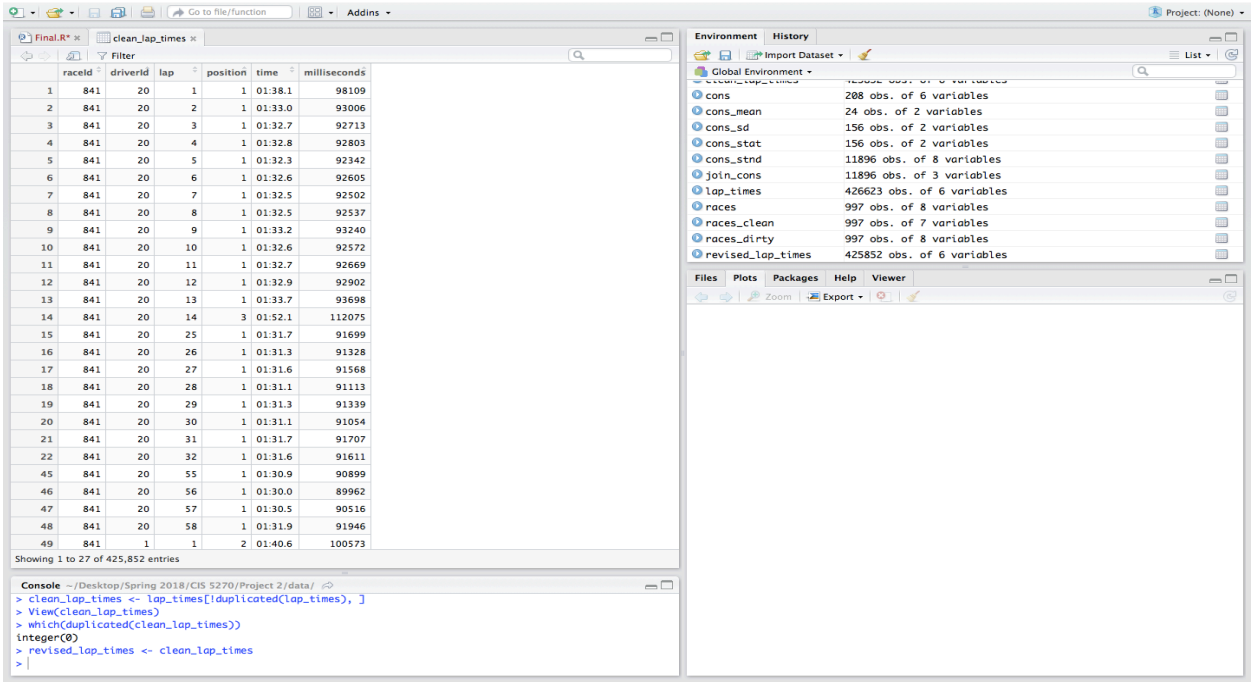


All TRUE depicts the repetitions which could be displayed using duplicated() in R.



Using which(duplicated()) will display the values which are duplicates.

After



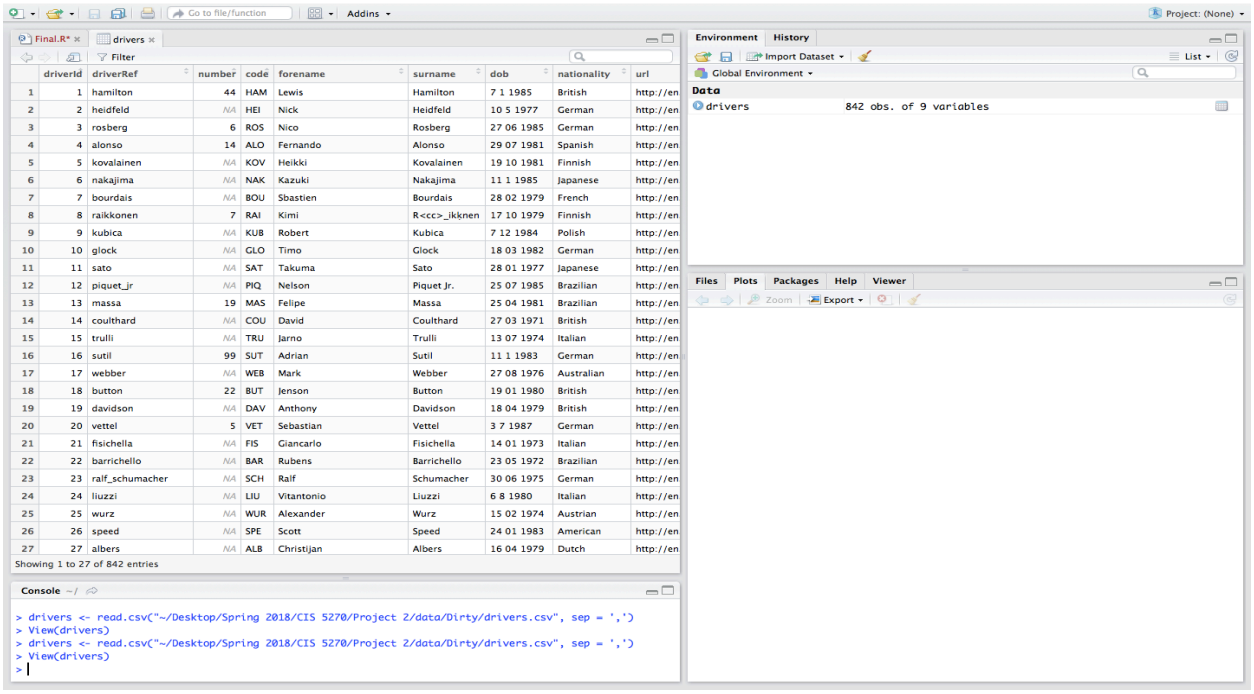
In LapTimes.csv there are total 6 labels and there was total of 771 duplicate rows, duplicated () was used to find the duplicates and remove them so to have unbiased dataset.

Code:

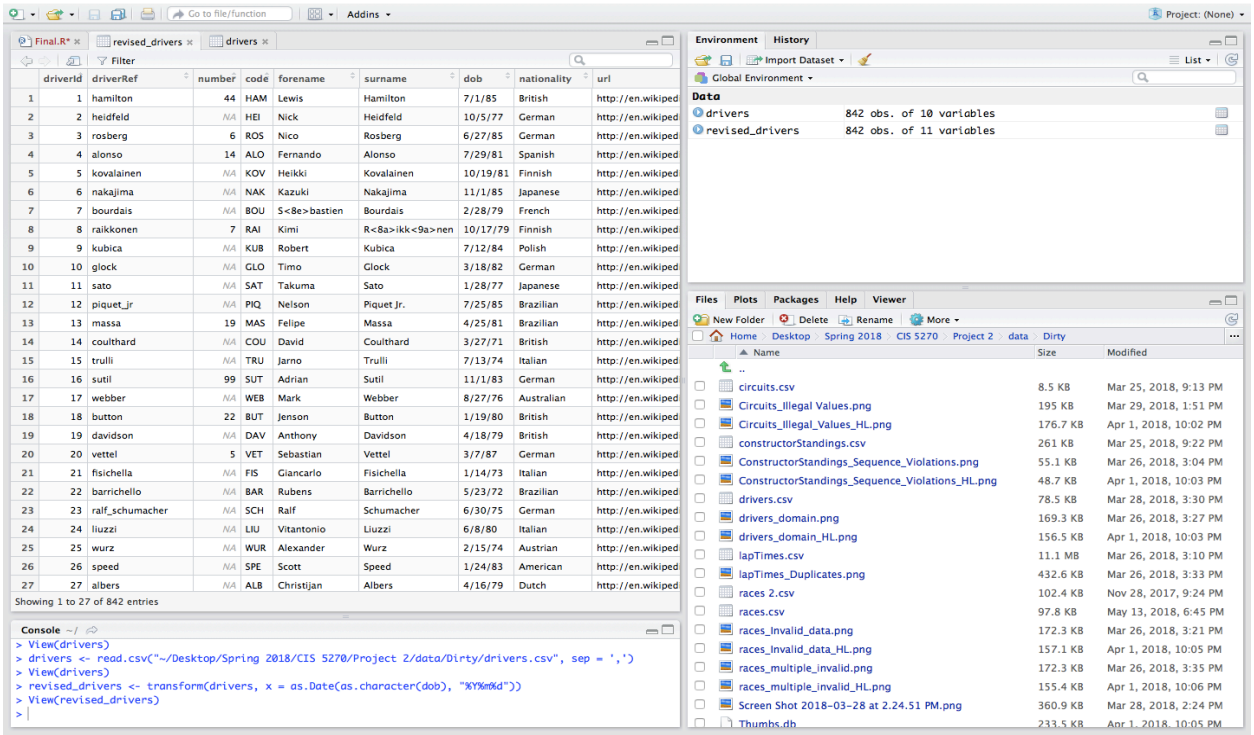
```
> lap_times <- read.csv("~/Desktop/Spring 2018/CIS 5270/Project 2/data/Dirty/lapTimes.csv", sep = ',')
> duplicated(lap_times)
> which(duplicated(lap_times))
> clean_lap_times <- lap_times[!duplicated(lap_times), ]
> which(duplicated(clean_lap_times))
> revised_lap_times <- clean_lap_times
```

Invalid values

Before



After



The dob column had invalid formatting of values, used transform() to fix the error and return the date of birth in correct format.

Code:

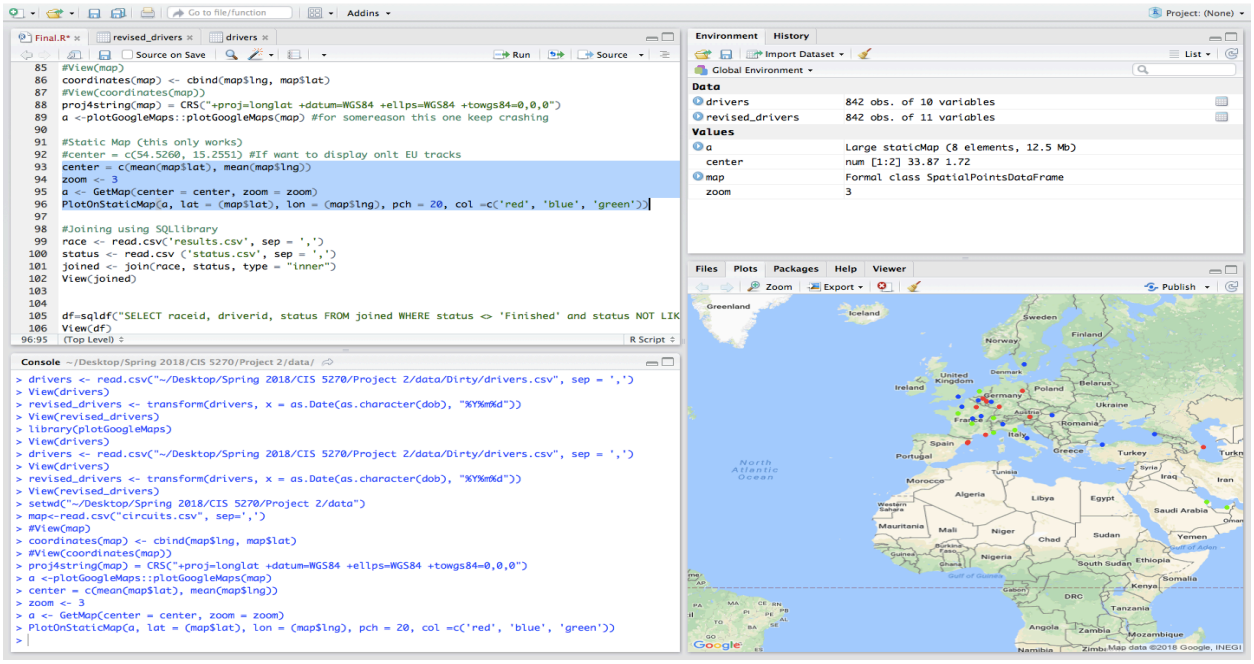
```
> drivers <- read.csv("~/Desktop/Spring 2018/CIS 5270/Project 2/data/Dirty/drivers.csv", sep =
',')
> revised_drivers <- transform(drivers, x = as.Date(as.character(dob), "%Y%m%d"))
```

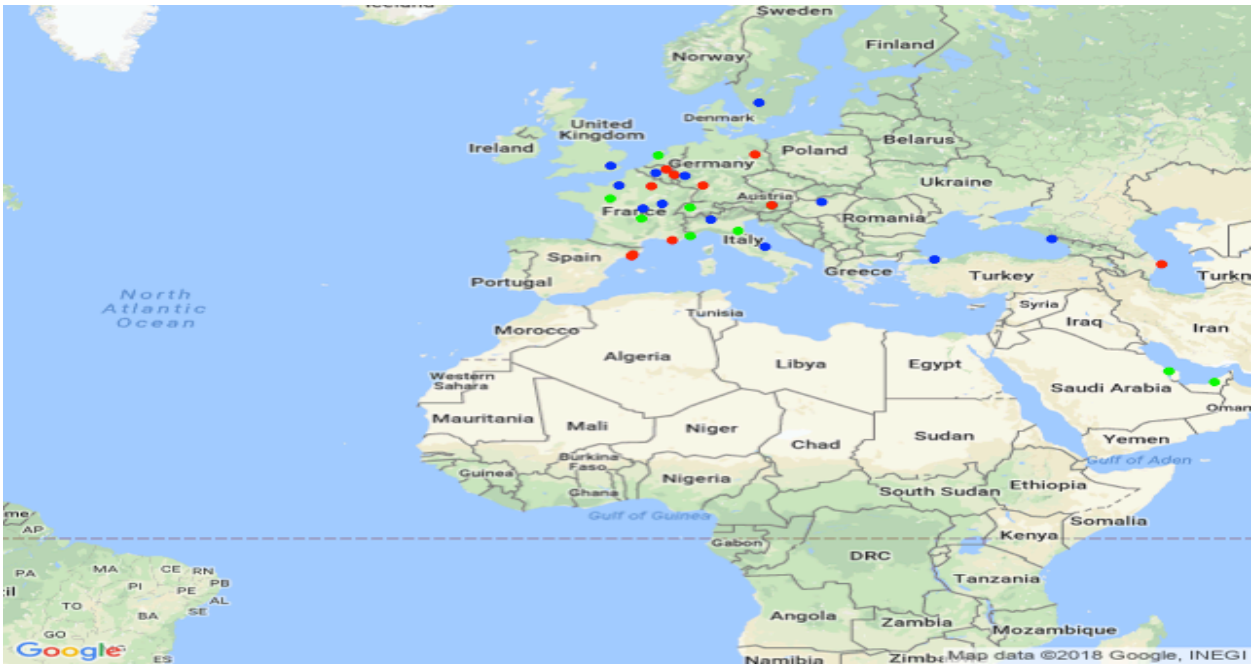
Note: Used might face syntax error if above codes are pasted to console as MS-Word changes the formatting of some special characters which are being used in code.

Data Visualization

In this section I will explain the findings of the research. I have answered 9 major questions which are very crucial for this research analysis.

1. Plotting racetracks on the world map.

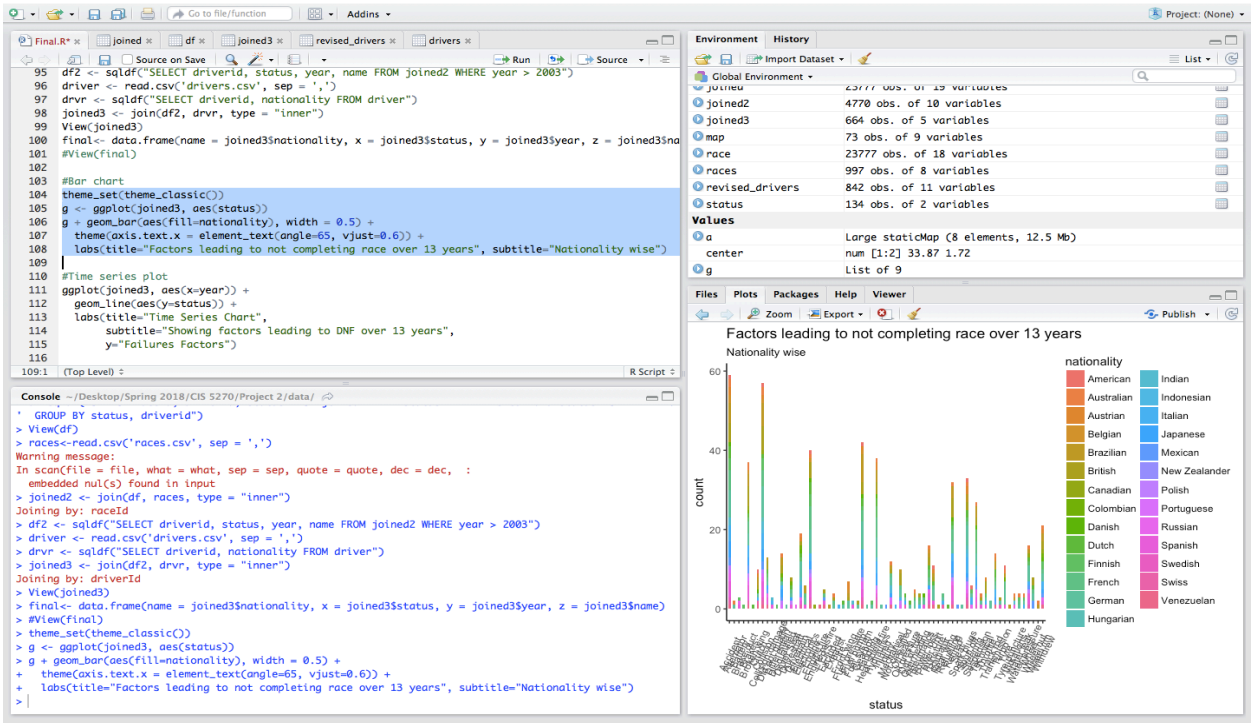


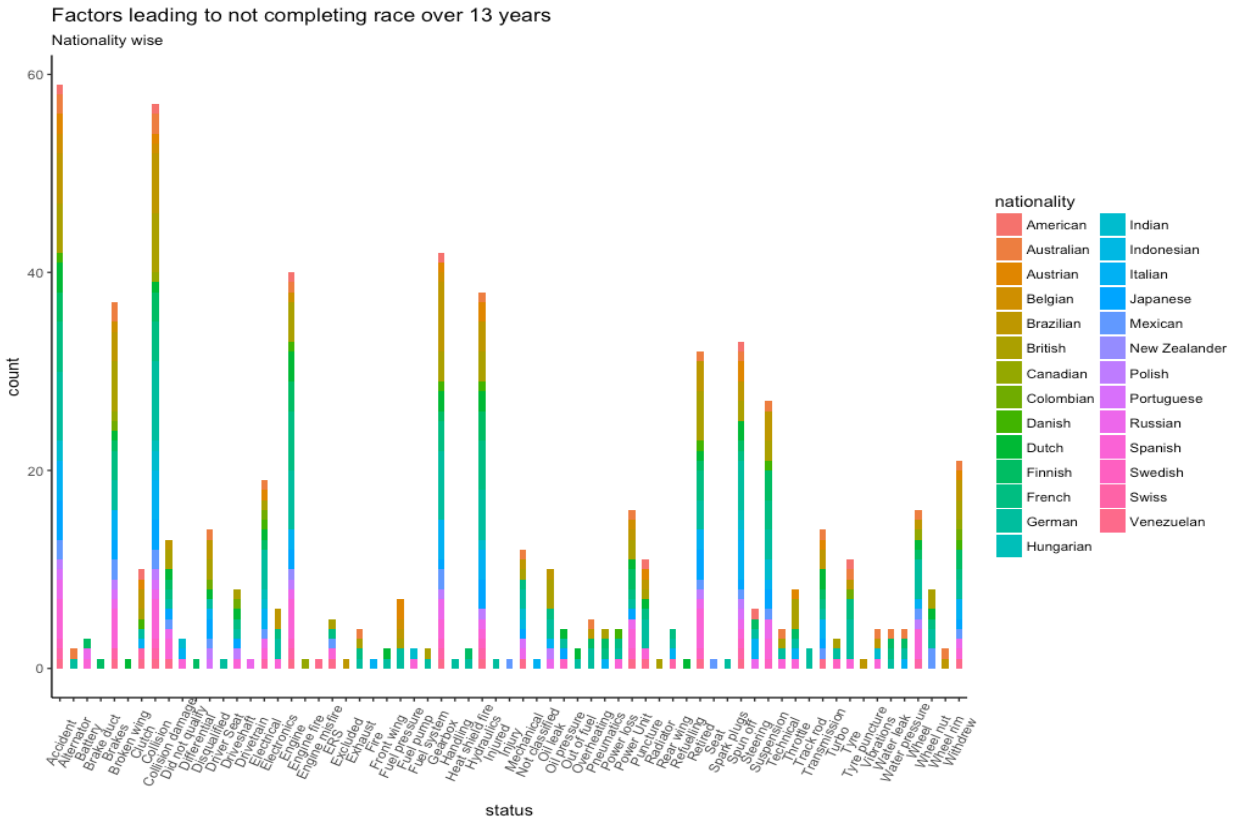


Insight: This visualization will give the reader graphical information about where the racetracks are located. Since the F1 was originated in Europe and you'll see most of the major racetracks there. This finding will help enthusiast spectators to plan their trips for races and exploring the region.

Although there are more racetracks existing but since the research is limited to using dataset from some source I couldn't add more information on this finding.

2. Factors leading to DNF (Did Not Finish) in race based on nationality



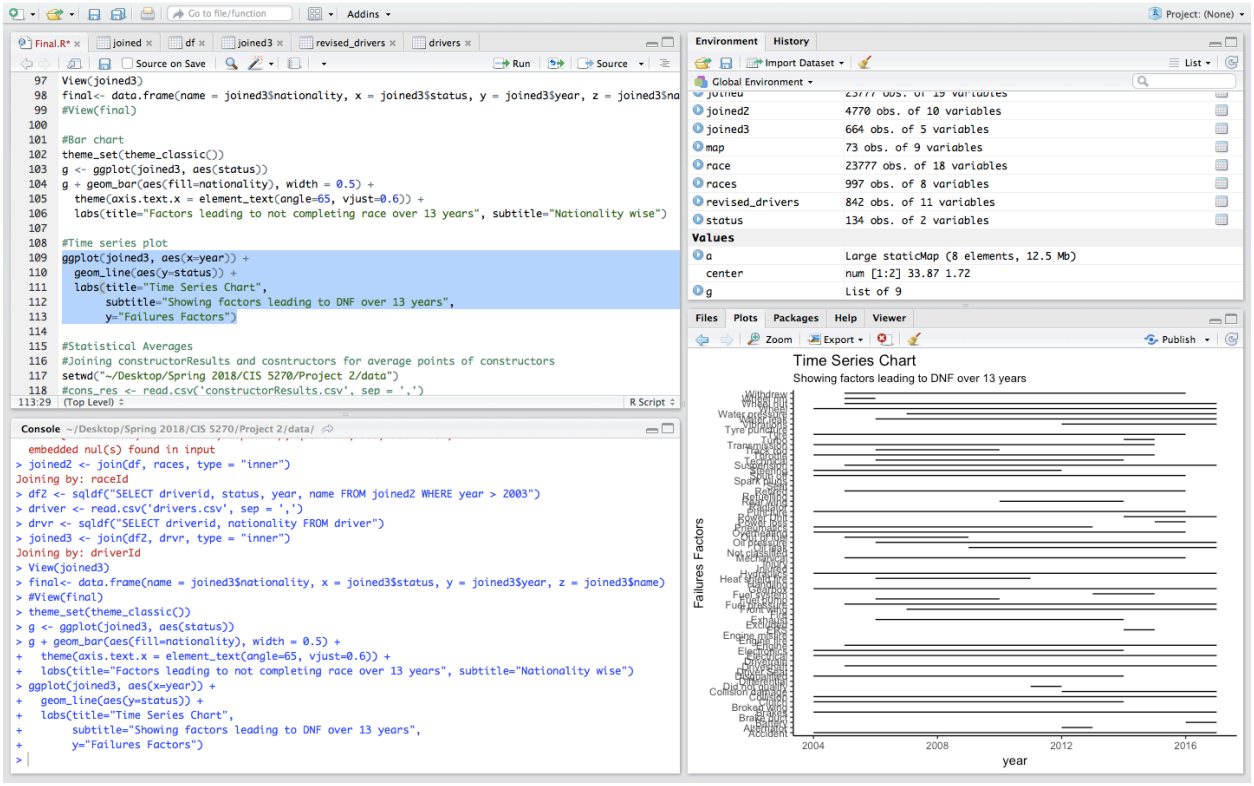


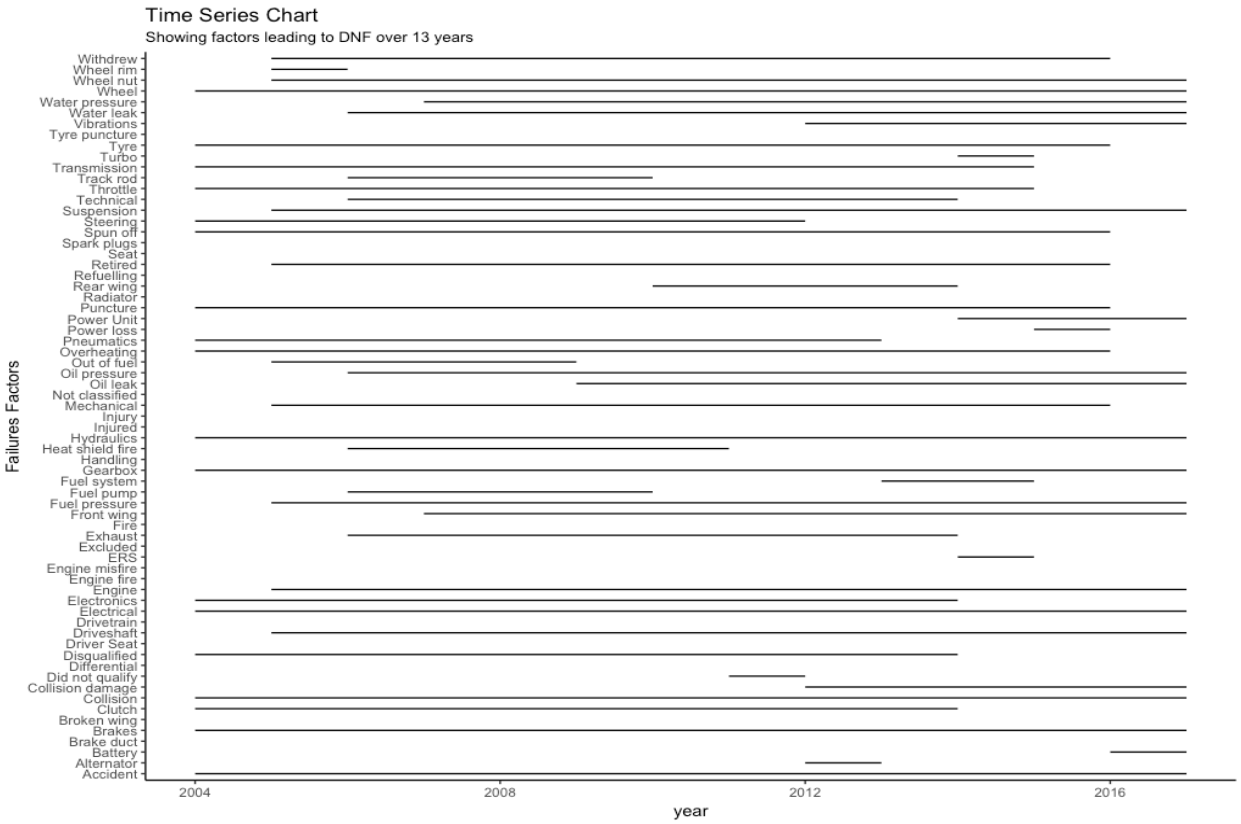
Insight: The above findings provides the audience with the information on factors leading to DNF in the race. Every F1 season racing teams around the world put in best of their efforts and resources to make the best of it. DNF is the most tragic thing which could happen to any racing team in F1 and there are various reasons which lead to this misfortune. The above finding can provide racers a better understanding of most common occurring reasons for DNF.

Accidents and collision are the 2 major reasons leading to DNF. Since, these factors are part of this sport so there's not lot which could be done to avoid these factors. But it is interesting to see that the 3rd most common reason of DNF is Fuel Systems, which can be improved by the teams.

On the X-axis you will see the status which are the reasons for the failures and on Y-axis you will see the count which is the number of time these factors have occurred over the period of 67 years. Also, audience can see drivers of which nationality have suffered from which failures most.

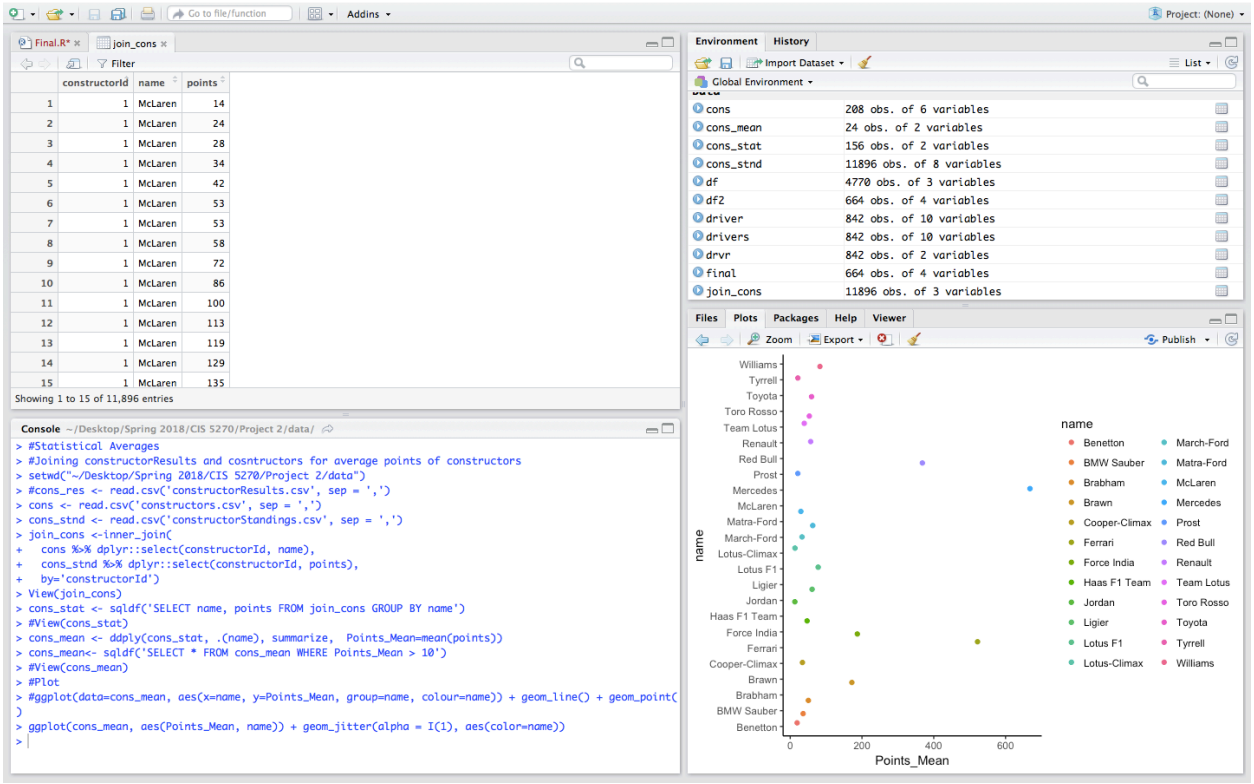
3. Time series chart of the DNF factors

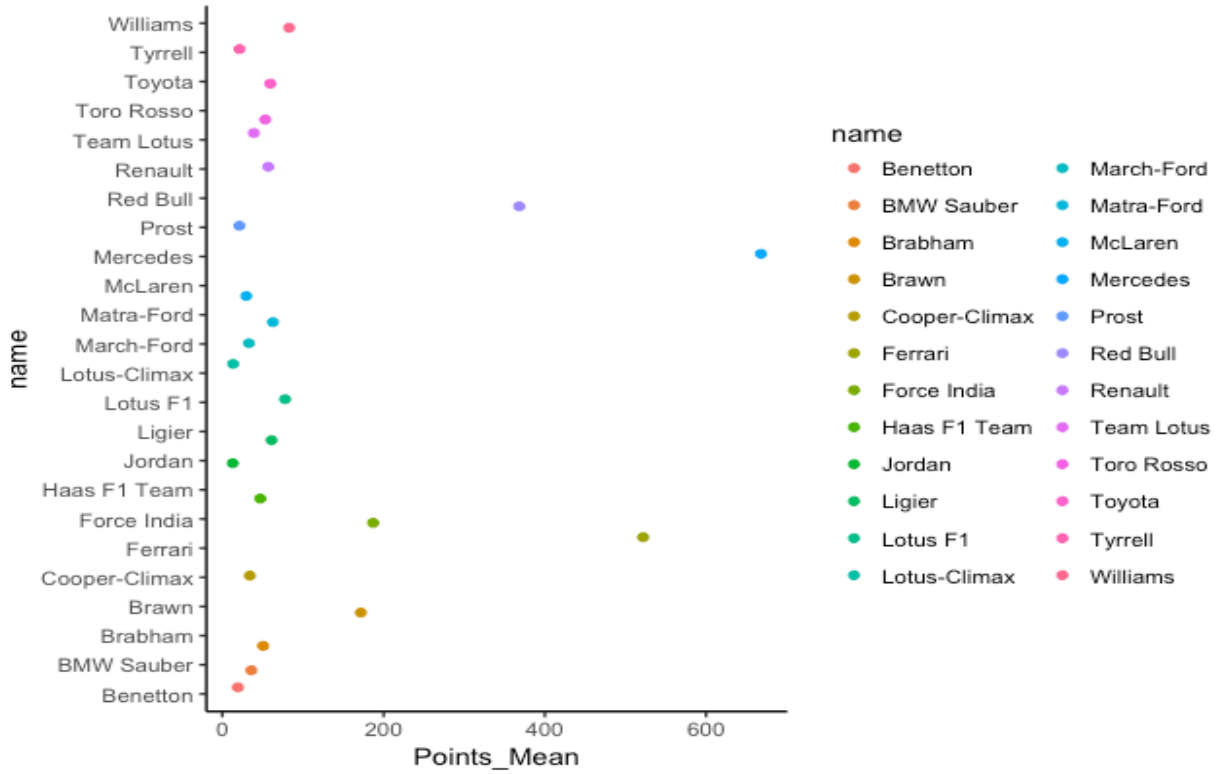




Insight: Addition to the last analysis, this chart represents the failure factors over the timeline of 13 years (2004 to 2017). By observing the above figure closely audience can interpret that the reason for fail over 13 years. Teams can look at the major reasons of DNF over this long span of time and can take corrective measures as there could be chances that teams couldn't reconcile the various factors of failures over 13 years and just try to fix the most frequent problem occurred. On the X-axis you will see timeline and on Y-axis you will see the factors of failures. After accidents and collision which are least controllable, brakes and electronics are the highest occurring factors of failures in the races.

4. Mean of total points earned by all teams over 67 years

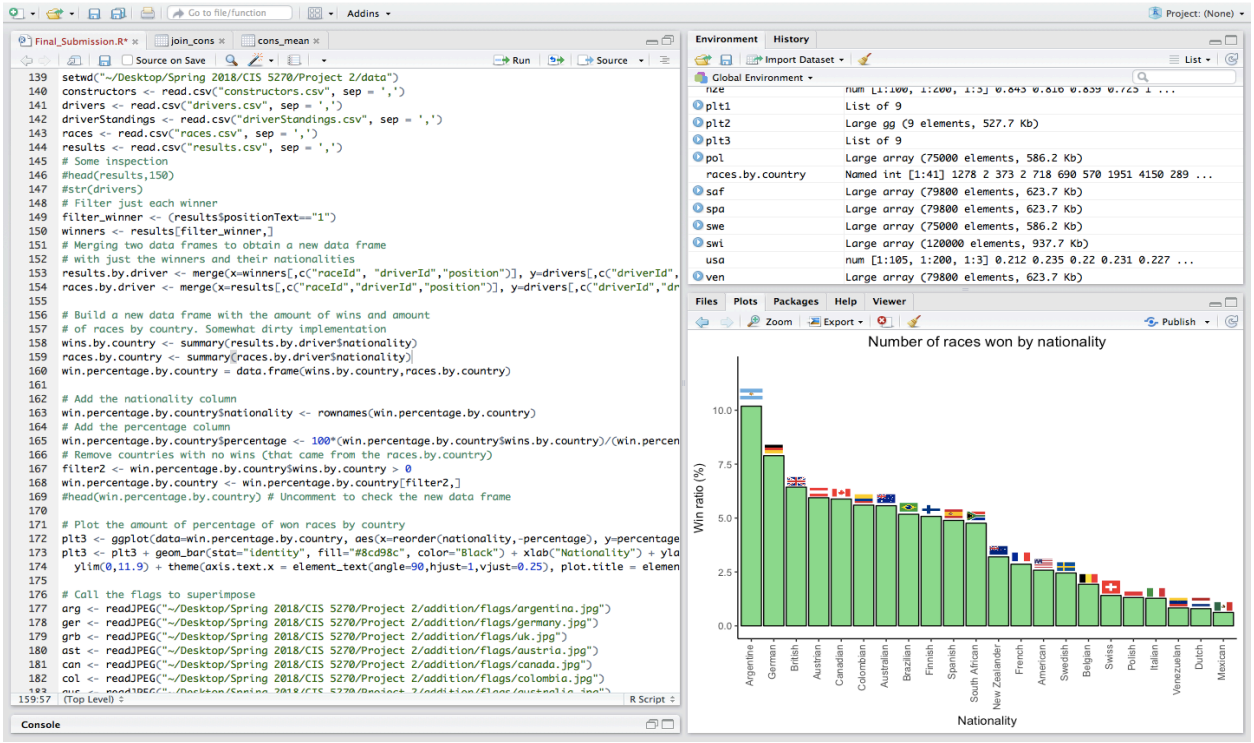


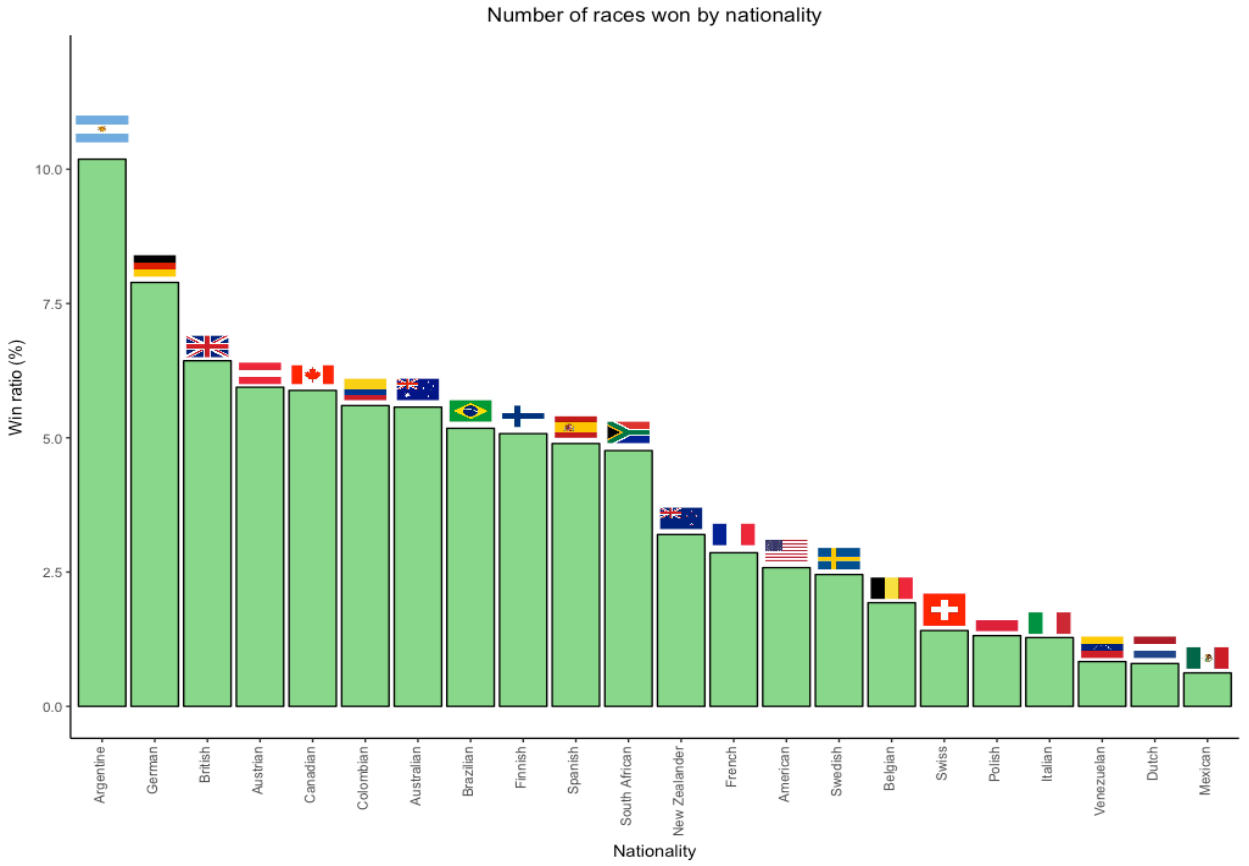


Insight: F1 racing works on season format i.e. couple of races will be held around the globe throughout one season and teams will be awarded points based on their performance. At the end of the season, one with the highest points will become the National champion. The above insight represents Top 24 teams based on mean of their points through these 67 glorious years of F1. This insight can be used by motorsports enthusiasts, investors, sponsors, and drivers to pick the team they would like to be working with.

On X-axis you will see name of the constructors/teams which have mean value of points more than 10 (which filtered out 132 other teams) and on Y-axis you will see top 24 teams with highest mean points.

5. Number of races won by nationality with respect to win ration (%)



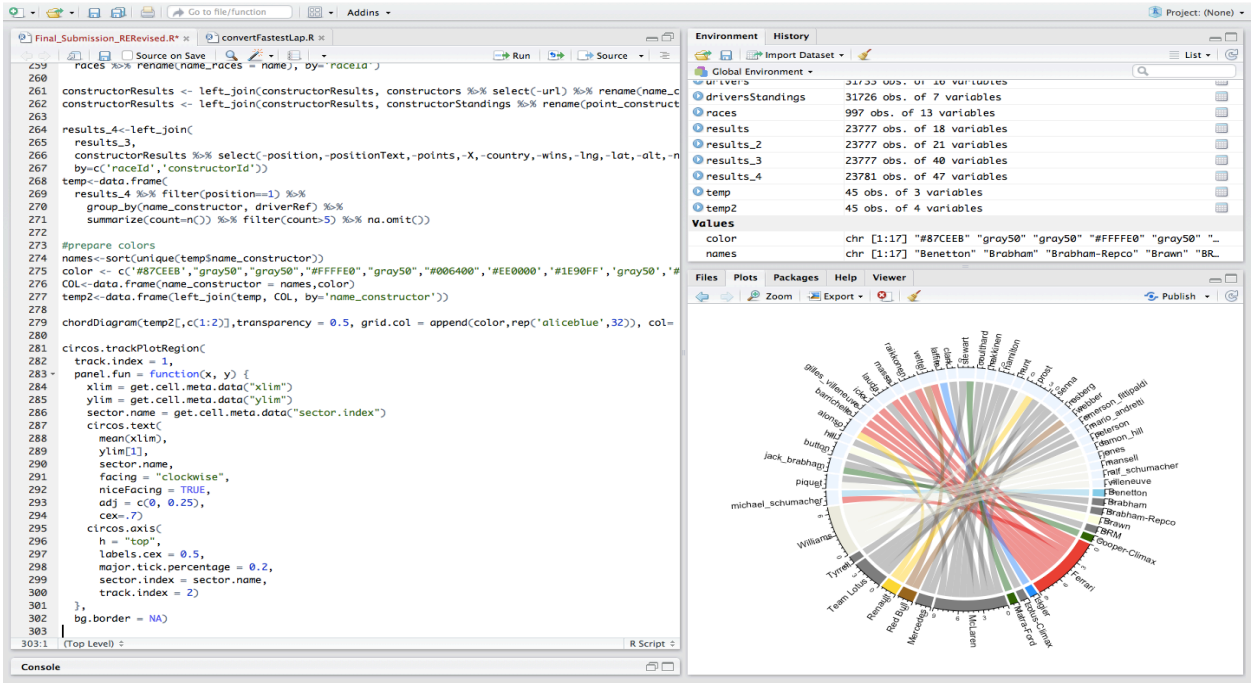


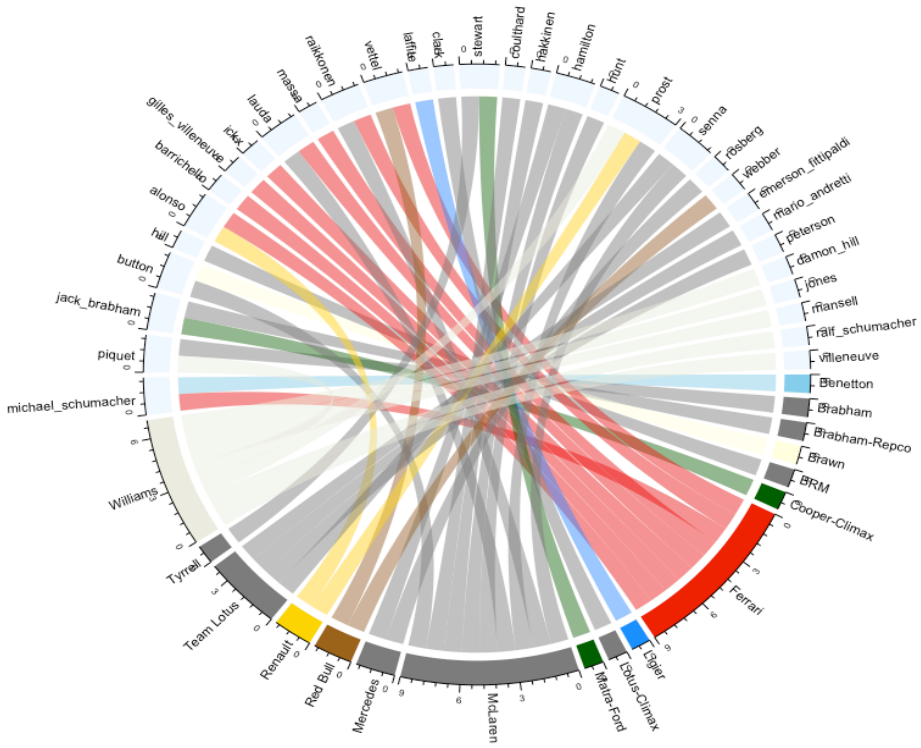
Insight: This analysis represents the nationality of racers and the ratio of their winning among other drivers. 67 years of data is being used for this insight. On the Y-axis the ratio is being calculated based on the winning on the scale of 10.0 and audience can clearly see that racers from Argentina have been performing best through these last 67 years.

This insight can be taken into consideration by teams looking for new drivers or planning to switch drivers from other teams. Top 3 nationalities are Argentina, German, & British.

On the X-axis you can see win ration of the drivers on the scale of 0-10 and on Y-axis you will see the nationality of the drivers. Argentinian drivers have been performing best throughout this course of 67.

6. Driver and constructor relation

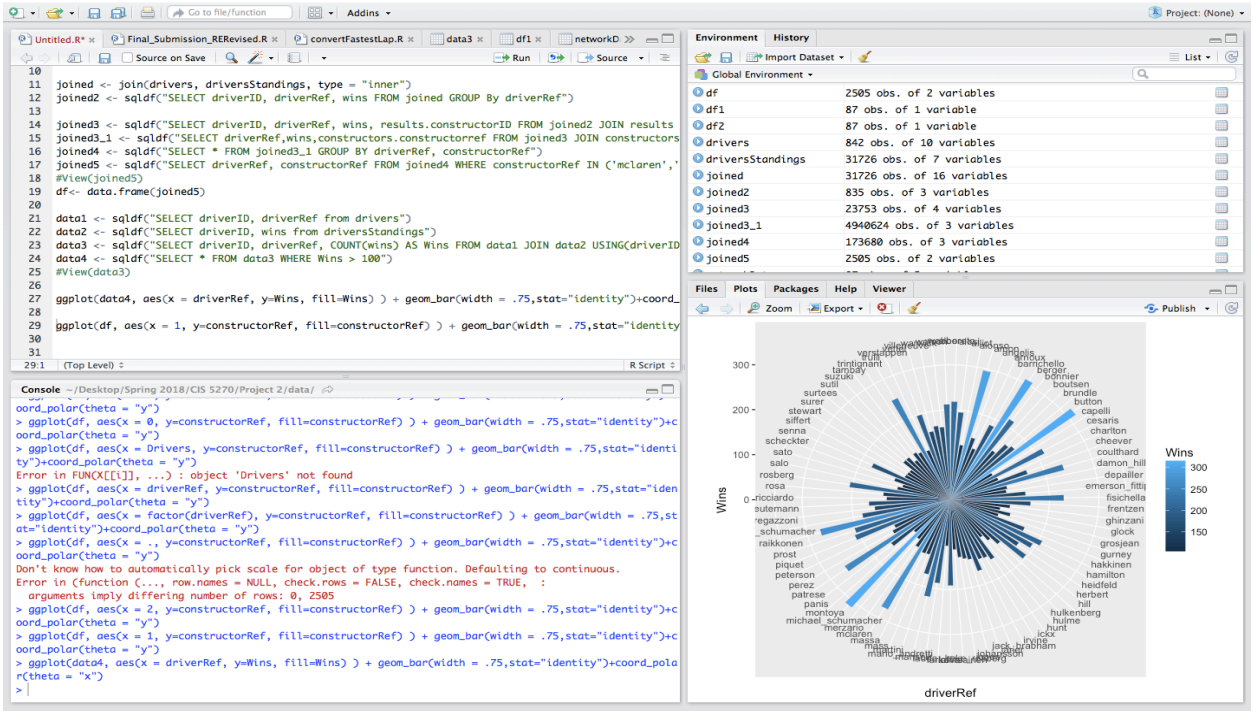


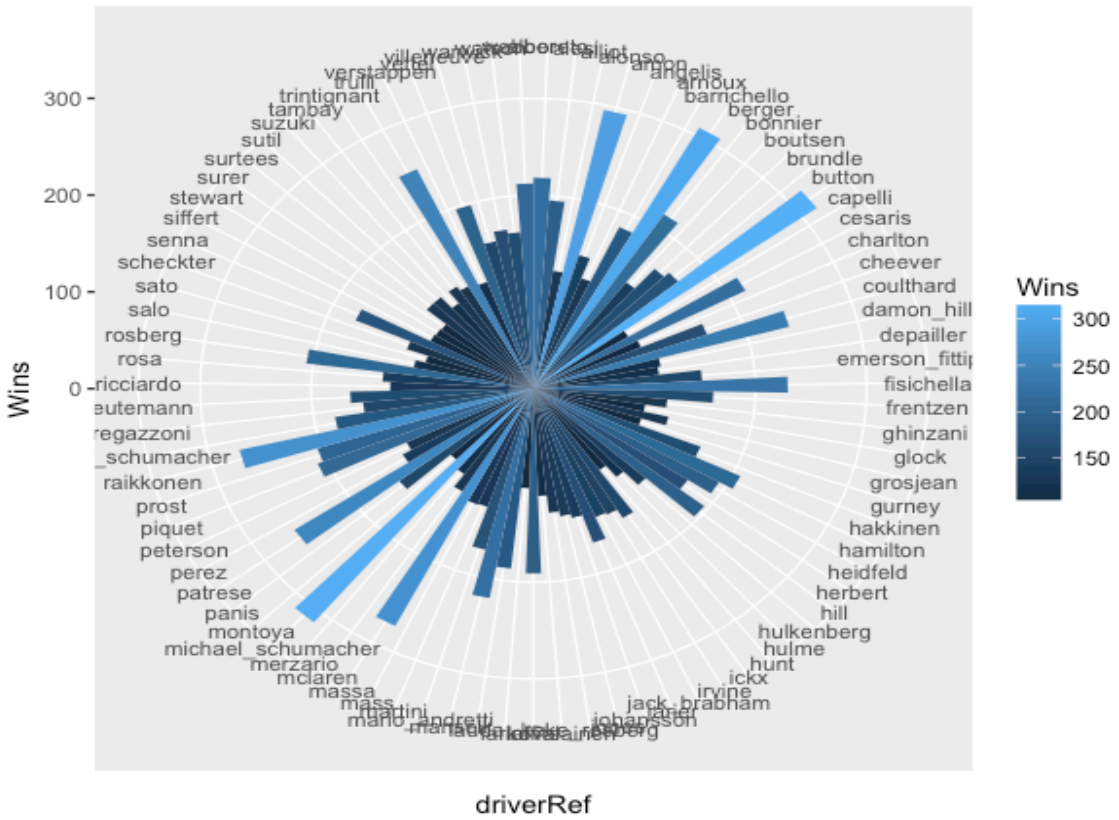


Insight: This analysis was done to find out which constructor have been dominating the F1 and the contributing drivers. This chord diagram will enable the audience to identify the backbone drivers of the constructors. By looking closely, it can be observed that McLaren, Ferrari, and Williams have been dominating the F1 overall (considering 8 datasets with various features). This insight can be used by anyone for identifying drivers associated with constructors/teams and their performance.

On this chord diagram you can see direct relationship between the drivers and constructors. Legendry drivers like Seena, Prost, Vettel, Shumaker, and Lauda have driven mostly for the top three constructors which are McLaren, Ferrari, & Williams. The reason behind the success of these constructors is the consistent and consecutive hard work of these drivers.

7. Driver who have won > 100 races in their career

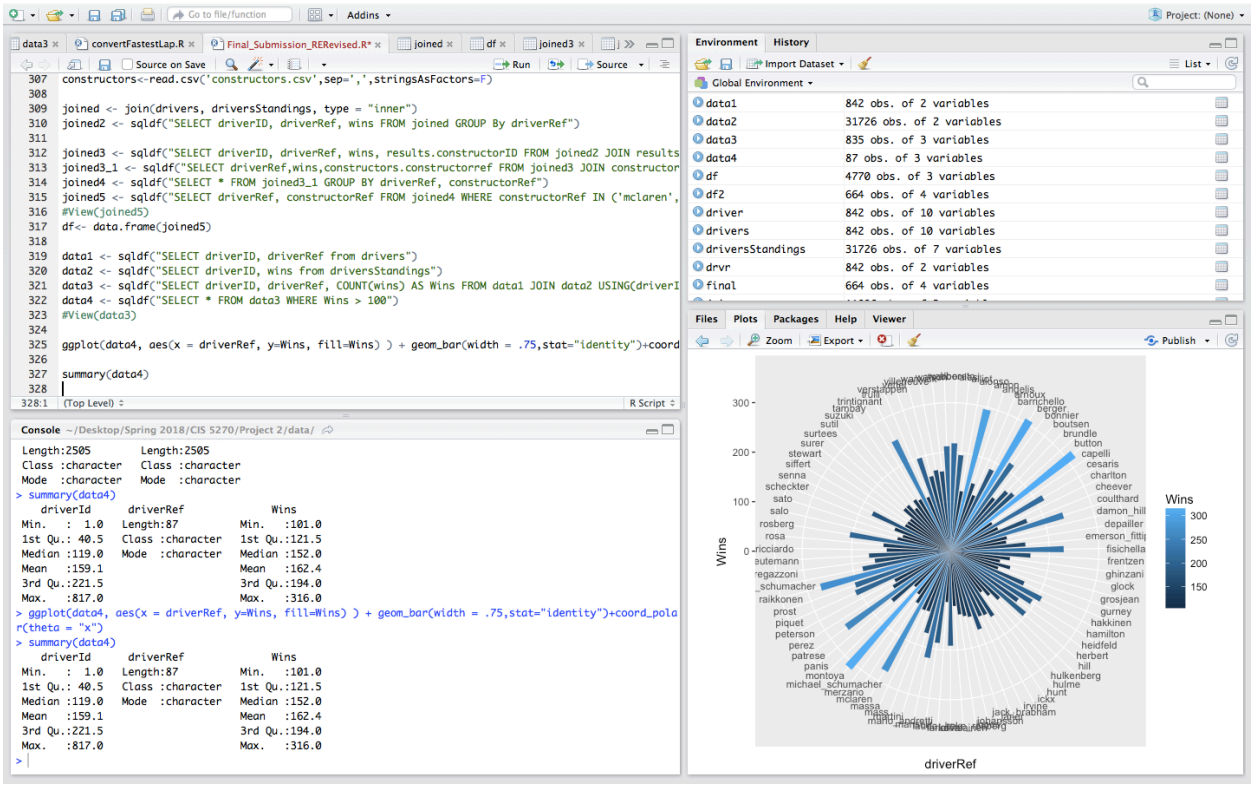




Insight: This is a very interesting finding where audience can see the drivers which have the highest wins in the career throughout. This analysis takes wins of drivers over 67 years (1950 – 2017). This analysis can answer the very important question which is being asked over and over again in F1 "Who has the highest victories in his career?". On the perimeter you can see the former name of the drivers and bars extending from the center towards the name represents the victories.

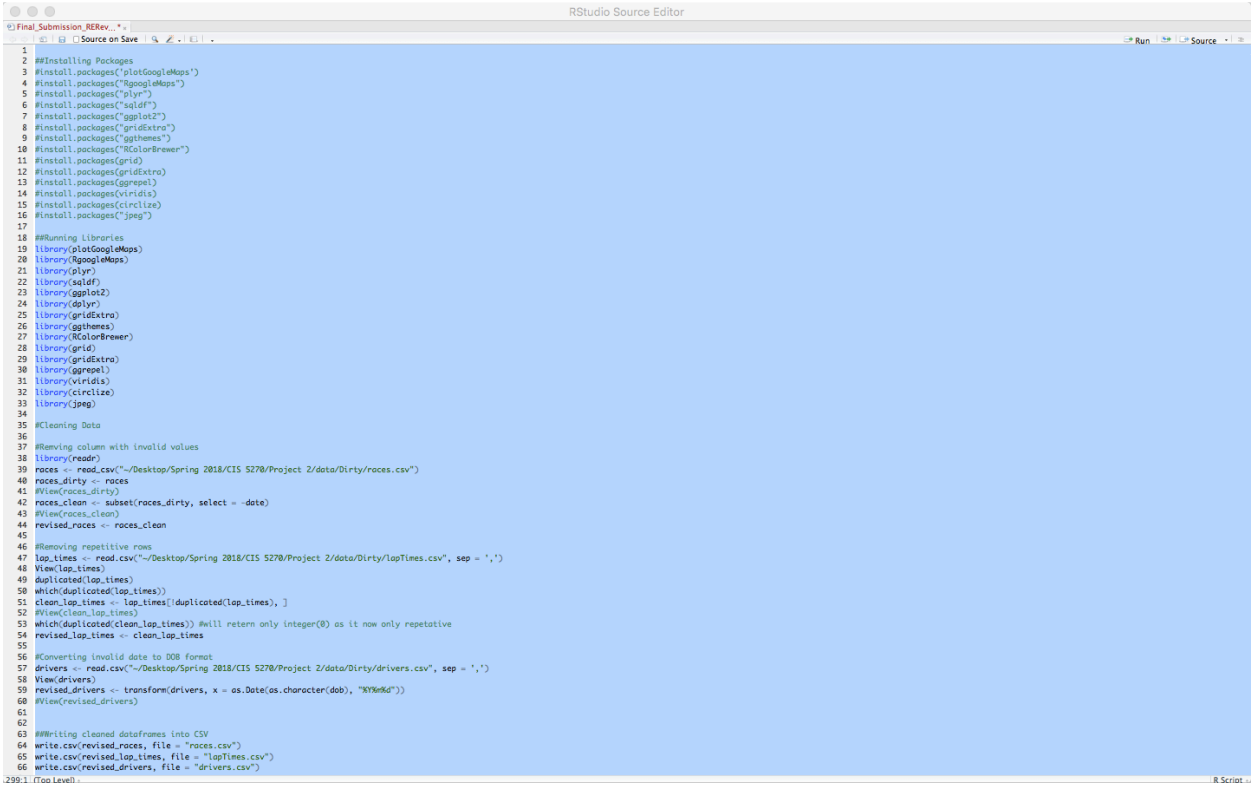
On this visualization audience can look at the top drivers which have more than 100 wins in their career.

Summary of dataset used for above analysis:



```
> summary(data4)
   driverId        driverRef            Wins
 Min.   :  1.0   Length:87          Min.   :101.0
 1st Qu.: 40.5   Class :character   1st Qu.:121.5
 Median :119.0   Mode  :character   Median :152.0
 Mean   :159.1                      Mean   :162.4
 3rd Qu.:221.5                      3rd Qu.:194.0
 Max.   :817.0                      Max.   :316.0
```

Script



Complete script is of ~400 lines which will execute the code for entire project. It starts with installing packages required for these analyses to plotting graphs and figures.

Since it is not possible to take screenshot of entire scripts and paste it this paper, I am uploading it on Moodle which you can download and use it for references.

Codes for data cleaning

Cleaning 1 - Removing invalid column

```
> library(readr)
> races <- read_csv("~/Desktop/Spring 2018/CIS 5270/Project 2/data/Dirty/races.csv")
> races_dirty <- races
> View(races_dirty)
> races_clean <- subset(races_dirty, select = -date)
> View(races_clean)
```

Cleaning 2 – Duplicated rows

```
> lap_times <- read.csv("~/Desktop/Spring 2018/CIS 5270/Project 2/data/Dirty/lapTimes.csv",
sep = ',')
> duplicated(lap_times)
> which(duplicated(lap_times))
> clean_lap_times <- lap_times[!duplicated(lap_times), ]
> which(duplicated(clean_lap_times))
> revised_lap_times <- clean_lap_times
```

Cleaning 3 – Invalid Values

```
> drivers <- read.csv("~/Desktop/Spring 2018/CIS 5270/Project 2/data/Dirty/drivers.csv", sep =
',')
> revised_drivers <- transform(drivers, x = as.Date(as.character(dob), "%Y%m%d"))
```

Below are the R libraries which are being used for successfully completing this research project:

Installing Packages

```
install.packages('plotGoogleMaps')

install.packages("RgoogleMaps")

install.packages("plyr")

install.packages("sqldf")

install.packages("ggplot2")

install.packages("gridExtra")

install.packages("ggthemes")

install.packages("RColorBrewer")

install.packages(grid)

install.packages(gridExtra)

install.packages(ggrepel)

install.packages(viridis)

install.packages(circlize)

install.packages("jpeg")
```

Running Libraries

```
library(plotGoogleMaps)

library(RgoogleMaps)

library(plyr)

library(sqldf)

library(ggplot2)

library(dplyr)

library(gridExtra)

library(ggthemes)

library(RColorBrewer)

library(grid)

library(gridExtra)

library(ggrepel)

library(viridis)

library(circlize)

library(jpeg)
```

Code for Q1 - Plotting racetracks on the world map.

```
> setwd("~/Desktop/Spring 2018/CIS 5270/Project 2/data")
> map<-read.csv("circuits.csv", sep=',')
> center = c(mean(map$lat), mean(map$lng))
> zoom <- 3
> a <- GetMap(center = center, zoom = zoom)
> PlotOnStaticMap(a, lat = (map$lat), lon = (map$lng), pch = 20, col =c('red', 'blue', 'green'))
```

Code for Q2 - Factors leading to DNF (Did Not Finish) in race based on nationality

```
> race <- read.csv('results.csv', sep = ',')
> status <- read.csv ('status.csv', sep = ',')
> joined <- join(race, status, type = "inner")
Joining by: statusId
> df=sqldf("SELECT raceid, driverid, status FROM joined WHERE status <> 'Finished' and
status NOT LIKE '+%'  GROUP BY status, driverid")
> races<-read.csv('races.csv', sep = ',')
> joined2 <- join(df, races, type = "inner")
Joining by: raceId
> df2 <- sqldf("SELECT driverid, status, year, name FROM joined2 WHERE year > 2003")
> driver <- read.csv('drivers.csv', sep = ',')
> drvr <- sqldf("SELECT driverid, nationality FROM driver")
> joined3 <- join(df2, drvr, type = "inner")
Joining by: driverId
```

```
> final<- data.frame(name = joined3$nationality, x = joined3$status, y = joined3$year, z =
joined3$name)
> theme_set(theme_classic())
> g <- ggplot(joined3, aes(status))
> g + geom_bar(aes(fill=nationality), width = 0.5) + theme(axis.text.x = element_text(angle=65,
vjust=0.6)) + labs(title="Factors leading to not completing race over 13 years",
subtitle="Nationality wise")
```

Code for Q3 - Time series chart of the DNF factors

```
> ggplot(joined3, aes(x=year)) + geom_line(aes(y=status)) + labs(title="Time Series Chart",
subtitle="Showing factors leading to DNF over 13 years", y="Failures Factors")
```

Code for Q4 - Mean of total points earned by all teams over 67 years

```
> setwd("~/Desktop/Spring 2018/CIS 5270/Project 2/data")
> #cons_res <- read.csv('constructorResults.csv', sep = ',')
> cons <- read.csv('constructors.csv', sep = ',')
> cons_stnd <- read.csv('constructorStandings.csv', sep = ',')
> join_cons <-inner_join(cons %>% dplyr::select(constructorId, name), cons_stnd %>%
dplyr::select(constructorId, points), by='constructorId')
> cons_stat <- sqldf('SELECT name, points FROM join_cons GROUP BY name')
> cons_mean <- ddply(cons_stat, .(name), summarize,  Points_Mean=mean(points))
> cons_mean<- sqldf('SELECT * FROM cons_mean WHERE Points_Mean > 10')
> ggplot(cons_mean, aes(Points_Mean, name)) + geom_jitter(alpha = I(1), aes(color=name))
```

Code for Q5 - Number of races won by nationality with respect to win ration (%)

```
> setwd("~/Desktop/Spring 2018/CIS 5270/Project 2/data")
> constructors <- read.csv("constructors.csv", sep = ',')
> drivers <- read.csv("drivers.csv", sep = ',')
> driverStandings <- read.csv("driverStandings.csv", sep = ',')
> races <- read.csv("races.csv", sep = ',')
> results <- read.csv("results.csv", sep = ',')
> filter_winner <- (results$positionText=="1")
> winners <- results[filter_winner,]
> results.by.driver <- merge(x=winners[,c("raceId", "driverId","position")],
y=drivers[,c("driverId","driverRef","nationality")], by="driverId")
> races.by.driver <- merge(x=results[,c("raceId","driverId","position")],
y=drivers[,c("driverId","driverRef","nationality")], by="driverId")
> wins.by.country <- summary(results.by.driver$nationality)
> races.by.country <- summary(races.by.driver$nationality)
> win.percentage.by.country = data.frame(wins.by.country,races.by.country)
> win.percentage.by.country$nationality <- rownames(win.percentage.by.country)
> win.percentage.by.country$percentage <-
100*(win.percentage.by.country$wins.by.country)/(win.percentage.by.country$races.by.country)
> filter2 <- win.percentage.by.country$wins.by.country > 0
> win.percentage.by.country <- win.percentage.by.country[filter2,]
```

```
> plt3 <- ggplot(data=win.percentage.by.country, aes(x=reorder(nationality,-percentage),
y=percentage))
> plt3 <- plt3 + geom_bar(stat="identity", fill="#8cd98c", color="Black") + xlab("Nationality")
+ ylab("Win ratio (%)") + ggtitle("Number of races won by nationality")
+   ylim(0,11.9) + theme(axis.text.x = element_text(angle=90,hjust=1,vjust=0.25), plot.title =
element_text(hjust=0.5))
> arg <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/argentina.jpg")
> ger <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/germany.jpg")
> grb <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/uk.jpg")
> ast <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/austria.jpg")
> can <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/canada.jpg")
> col <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/colombia.jpg")
> aus <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/australia.jpg")
> bra <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/brazil.jpg")
> fin <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/finland.jpg")
> spa <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/spain.jpg")
> saf <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/south-africa.jpg")
> nze <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/new-
zealand.jpg")
> fra <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/france.jpg")
> usa <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/usa.jpg")
> swe <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/sweden.jpg")
> bel <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/belgium.jpg")
```

```
> swi <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/switzerland.jpg")
> pol <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/poland.jpg")
> ita <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/italy.jpg")
> ven <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/venezuela.jpg")
> net <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/netherlands.jpg")
> mex <- readJPEG("~/Desktop/Spring 2018/CIS 5270/Project 2/addition/flags/mexico.jpg")
> plt3 + annotation_raster(arg, ymin = 10.5,ymax=11,xmin=0.5,xmax=1.5) +
annotation_raster(ger, ymin = 8.0,ymax=8.4,xmin=1.6,xmax=2.4) +
annotation_raster(grb, ymin = 6.5,ymax=6.9,xmin=2.6,xmax=3.4) +
annotation_raster(ast, ymin = 6.0,ymax=6.4,xmin=3.6,xmax=4.4) +
annotation_raster(can, ymin = 6.0,ymax=6.35,xmin=4.6,xmax=5.4) +
annotation_raster(col, ymin = 5.7,ymax=6.1,xmin=5.6,xmax=6.4) +
annotation_raster(aus, ymin = 5.7,ymax=6.1,xmin=6.6,xmax=7.4) +
annotation_raster(bra, ymin = 5.3,ymax=5.7,xmin=7.6,xmax=8.4) +
annotation_raster(fin, ymin = 5.2,ymax=5.6,xmin=8.6,xmax=9.4) +
annotation_raster(spa, ymin = 5.0,ymax=5.4,xmin=9.6,xmax=10.4) +
annotation_raster(saf, ymin = 4.9,ymax=5.3,xmin=10.6,xmax=11.4) +
annotation_raster(nze, ymin = 3.3,ymax=3.7,xmin=11.6,xmax=12.4) +
annotation_raster(fra, ymin = 3.0,ymax=3.4,xmin=12.6,xmax=13.4) +
annotation_raster(usa, ymin = 2.7,ymax=3.1,xmin=13.6,xmax=14.4) +
annotation_raster(swe, ymin = 2.55,ymax=2.95,xmin=14.6,xmax=15.4) +
annotation_raster(bel, ymin = 2.0,ymax=2.4,xmin=15.6,xmax=16.4) +
annotation_raster(swi, ymin = 1.5,ymax=2.1,xmin=16.6,xmax=17.4) +
```

annotation_raster(pol, ymin = 1.4,ymax=1.8,xmin=17.6,xmax=18.4) +

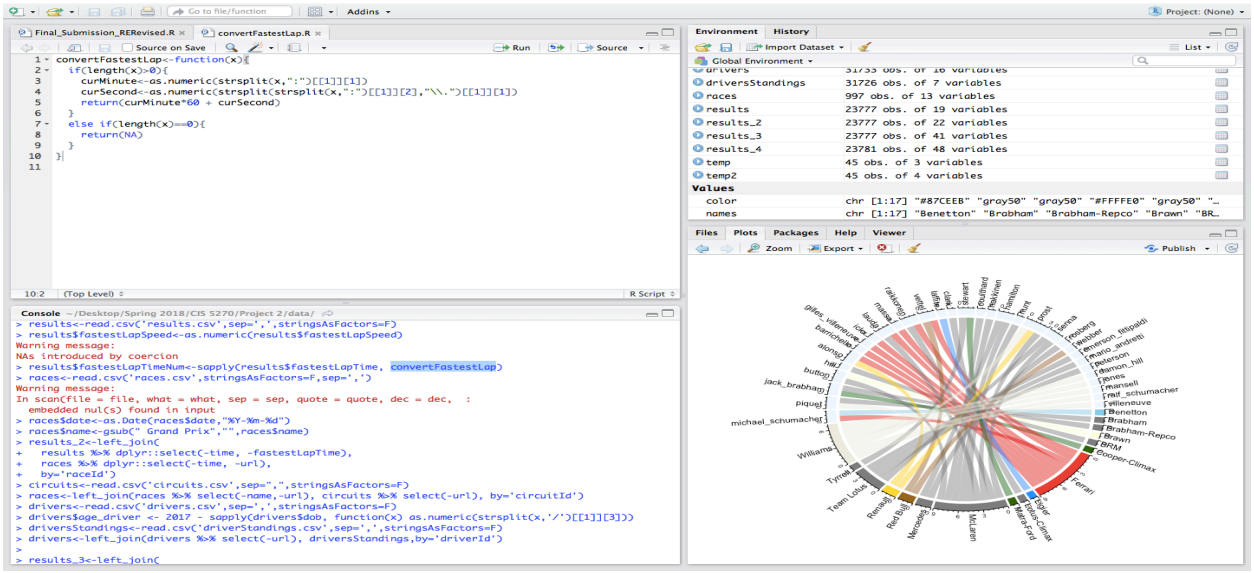
annotation_raster(ita, ymin = 1.35,ymax=1.75,xmin=18.6,xmax=19.4) +

annotation_raster(ven, ymin = 0.9,ymax=1.3,xmin=19.6,xmax=20.4) +

annotation_raster(net, ymin = 0.9,ymax=1.3,xmin=20.6,xmax=21.4) +

annotation_raster(mex, ymin = 0.7,ymax=1.1,xmin=21.6,xmax=22.4)

Code for Q6 - Driver and constructor relation



Function name converFastestLap is created to feature FastestLap(character) to numeric(secondes) which is being used below in the code.

Code for function:

```
convertFastestLap<-function(x){
  if(length(x)>0){
    curMinute<-as.numeric(strsplit(x,":")[[1]][1])
    curSecond<-as.numeric(strsplit(strsplit(x,":")[[1]][2],"\\.")[[1]][1])
    return(curMinute*60 + curSecond)
```

```
  }
  else if(length(x)==0){
    return(NA)
  }
}
> results<-read.csv('results.csv',sep=',',stringsAsFactors=F)
> results$fastestLapSpeed<-as.numeric(results$fastestLapSpeed)
> results$fastestLapTimeNum<-sapply(results$fastestLapTime, convertFastestLap)
Error in match.fun(FUN) : object 'convertFastestLap' not found
> races<-read.csv('races.csv',stringsAsFactors=F,sep=',')
> races$date<-as.Date(races$date,"%Y-%m-%d")
> races$name<-gsub(" Grand Prix","",races$name)
> results_2<-left_join(
+   results %>% dplyr::select(-time, -fastestLapTime),
+   races %>% dplyr::select(-time, -url),
+   by='raceId')
> circuits<-read.csv('circuits.csv',sep=",",stringsAsFactors=F)
> races<-left_join(races %>% select(-name,-url), circuits %>% select(-url), by='circuitId')
> drivers<-read.csv('drivers.csv',sep=',',stringsAsFactors=F)
> drivers$age_driver <- 2017 - sapply(drivers$dob, function(x) as.numeric(strsplit(x,'/')[[1]][3]))
> driversStandings<-read.csv('driverStandings.csv',sep=',',stringsAsFactors=F)
> drivers<-left_join(drivers %>% select(-url), driversStandings,by='driverId')
```

```
> results_3<-left_join(results, drivers %>% dplyr::rename(number_drivers = number) %>%
select(-points, -position, -positionText), by=c('driverId','raceId'))
> results_3<-left_join(results_3,races %>% select(-time), by='raceId')
> constructors<-read.csv('constructors.csv',sep=',',stringsAsFactors=F)
> constructorStandings<-read.csv('constructorStandings.csv',sep=',',stringsAsFactors=F)
> constructorResults<-read.csv("constructorResults.csv",sep=",",stringsAsFactors=F)
> constructorResults<-left_join(constructorResults, races %>% rename(name_races = name),
by='raceId')
> constructorResults <- left_join(constructorResults, constructors %>% select(-url) %>%
rename(name_constructor = name), by='constructorId')
> constructorResults <- left_join(constructorResults, constructorStandings %>%
rename(point_constructor = points) %>% select(-X), by=c('constructorId','raceId'))
> results_4<-left_join(results_3, constructorResults %>% select(-position,-positionText,-points,-
X,-country,-wins,-lng,-lat,-alt,-nationality,-circuitRef,-round, -circuitId,-year,-time,-date,-
location), by=c('raceId','constructorId'))
> temp<-data.frame(results_4 %>% filter(position==1) %>% group_by(name_constructor,
driverRef) %>% summarize(count=n()) %>% filter(count>5) %>% na.omit())
> names<-sort(unique(temp$name_constructor))
> color <-
c('#87CEEB',"gray50","gray50","#FFFFE0","gray50","#006400",'#EE0000','#1E90FF','gray50','
#006400','#7F7F7F','#7F7F7F','#9C661F','#FFD700','gray50','gray50','#EEEEE0')
> COL<-data.frame(name_constructor = names,color)
> temp2<-data.frame(left_join(temp, COL, by='name_constructor'))
```

```
> chordDiagram(temp2[,c(1:2)],transparency = 0.5, grid.col = append(color,rep('aliceblue',32)),
col= as.character(temp2$color),annotationTrack = "grid", preAllocateTracks = 1)
> circos.trackPlotRegion(track.index = 1, panel.fun = function(x, y) {xlim =
get.cell.meta.data("xlim")
ylim = get.cell.meta.data("ylim")
sector.name = get.cell.meta.data("sector.index")
circos.text(mean(xlim), ylim[1], sector.name, facing = "clockwise", niceFacing = TRUE,
adj = c(0, 0.25), cex=.7)
circos.axis(h = "top", labels.cex = 0.5, major.tick.percentage = 0.2, sector.index = sector.name,
track.index = 2)
}, bg.border = NA)
```

Code for Q7 - Driver who have won > 100 races in their career.

```
drivers<-read.csv('drivers.csv',sep=',',stringsAsFactors=F)
driversStandings<-read.csv('driverStandings.csv',sep=',',stringsAsFactors=F)
results <- read.csv('results.csv', sep = ',')
constructors<-read.csv('constructors.csv',sep=',',stringsAsFactors=F)
joined <- join(drivers, driversStandings, type = "inner")
joined2 <- sqldf("SELECT driverID, driverRef, wins FROM joined GROUP By driverRef")
joined3 <- sqldf("SELECT driverID, driverRef, wins, results.constructorID FROM joined2 JOIN
results USING(driverID)")
joined3_1 <- sqldf("SELECT driverRef,wins,constructors.constructorref FROM joined3 JOIN
constructors")
```

```
joined4 <- sqldf("SELECT * FROM joined3_1 GROUP BY driverRef, constructorRef")
joined5 <- sqldf("SELECT driverRef, constructorRef FROM joined4 WHERE constructorRef IN
('mclaren','ferrari','williams')")
df<- data.frame(joined5)
data1 <- sqldf("SELECT driverID, driverRef from drivers")
data2 <- sqldf("SELECT driverID, wins from driversStandings")
data3 <- sqldf("SELECT driverID, driverRef, COUNT(wins) AS Wins FROM data1 JOIN data2
USING(driverID) GROUP By driverRef")
data4 <- sqldf("SELECT * FROM data3 WHERE Wins > 100")
ggplot(data4, aes(x = driverRef, y=Wins, fill=Wins) ) + geom_bar(width =
.75,stat="identity")+coord_polar(theta = "x")
```

```
> summary(data4)
   driverId        driverRef            Wins
 Min.   :  1.0   Length:87          Min.   :101.0
 1st Qu.: 40.5   Class :character   1st Qu.:121.5
 Median :119.0   Mode  :character   Median :152.0
 Mean   :159.1                      Mean   :162.4
 3rd Qu.:221.5                      3rd Qu.:194.0
 Max.   :817.0                      Max.   :316.0
```

References

Bouchet, Jonathan. (2017). F1 Data Analysis.

https://www.kaggle.com/jonathanbouchet/f1-data-analysis/data