Formula 1 Analyzation Project

Gabriel Downs

2024-04-28

A Deep Analysis Of The Top Formula 1 Teams From The 2010's

The main premise of this project will be to focus on the performances from the top Formula 1 teams during the 2010's which include two crucial eras for the automotive racing series. During this decade racing technicians, physicists, drivers, constructor personnel, and a bit of aerodynamic crews were hard at work to shift the gas-guzzling vehicles from their powerful V8 engines of the early 2000's to their new state-of-the-art V6 predecessors. Hence, a changing of crucial infrastructure dynamics, which exceeds more than just mathematics but instead causes a new era to be brought forth. This project will prove to hold significant analytical value between the alteration of the Schumacher Era (one of the most dominate drivers of all time) and the Hybrid Era (the introduction of the hybrid V6 engines which drew great skepticism which began in *2014*).

The type of learning/algorithm will be based primarily on examining the differentials of time intervals of the older generations cars to the (at the time) newer generation cars. We will examine the intervals taken on the same circuit which will provide as much valuable and comparable data while reducing as much noise as possible. Additionally, the *task* that will be implemented to focus on the performance metrics will be a *regression task*.

The database that will be utilized will be from: Buğa, A. B. (2021, December 13). Formula 1 (2010-2021). Kaggle. https://www.kaggle.com/datasets/ahmetburabua/formula-1-20102021/code

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
library(ggplot2)
mydata <- read_csv("~/Downloads/f1_2010-2021.csv")
```

```
## Rows: 237 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): index, team, code, name, surname, grand_prix, date
```

```
## dbl  (2): level_0, laps
## time (1): time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Data Description:

Within this CSV file there are exactly 237 rows and 10 columns of categorical information, as shown above. In retrospect, we will dilute the majority of these rows and columns and focus our attention on the top performing teams and performance metrics only. This will of course include the drivers of the highest caliber.

Cleaning The Data To Allow Observations Of High Performance Metrics

Step 1: We will begin to implement a code to figure out which race locations (grand_prix) were utilized the most to begin scraping down the given data and figure out which of the teams performed better than others.

```
GP_counts <- mydata %>%
  count(grand_prix, sort = TRUE)
print(GP_counts)
```

```
## # A tibble: 55 x 2
##    grand_prix        n
##    <chr>         <int>
##  1 Abu Dhabi        11
##  2 Belgium          11
##  3 Great Britain    11
##  4 Hungary          11
##  5 Italy            11
##  6 Spain            11
##  7 Australia        10
##  8 Bahrain          10
##  9 Brazil           10
## 10 Canada           10
## # i 45 more rows
```

It's worth noting that the variable "n" in the findings of the code above represents the amount of times Formula 1 has raced in that location. Of course, this would mean the 11 value represents a 100% rate during the studied time.

Step 2: Now, that we have discovered that the most common grand prix locations at the time were Abu Dhabi, Belgium, and Great Britain we can enhance our code to begin exempting the lower performing teams based on time. To start we will bring in crucial columns such as time, teams, and arguably most importantly the dates.

```
mydata <- read_csv("~/Downloads/f1_2010-2021.csv")
```

```
## Rows: 237 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): index, team, code, name, surname, grand_prix, date
## dbl  (2): level_0, laps
```

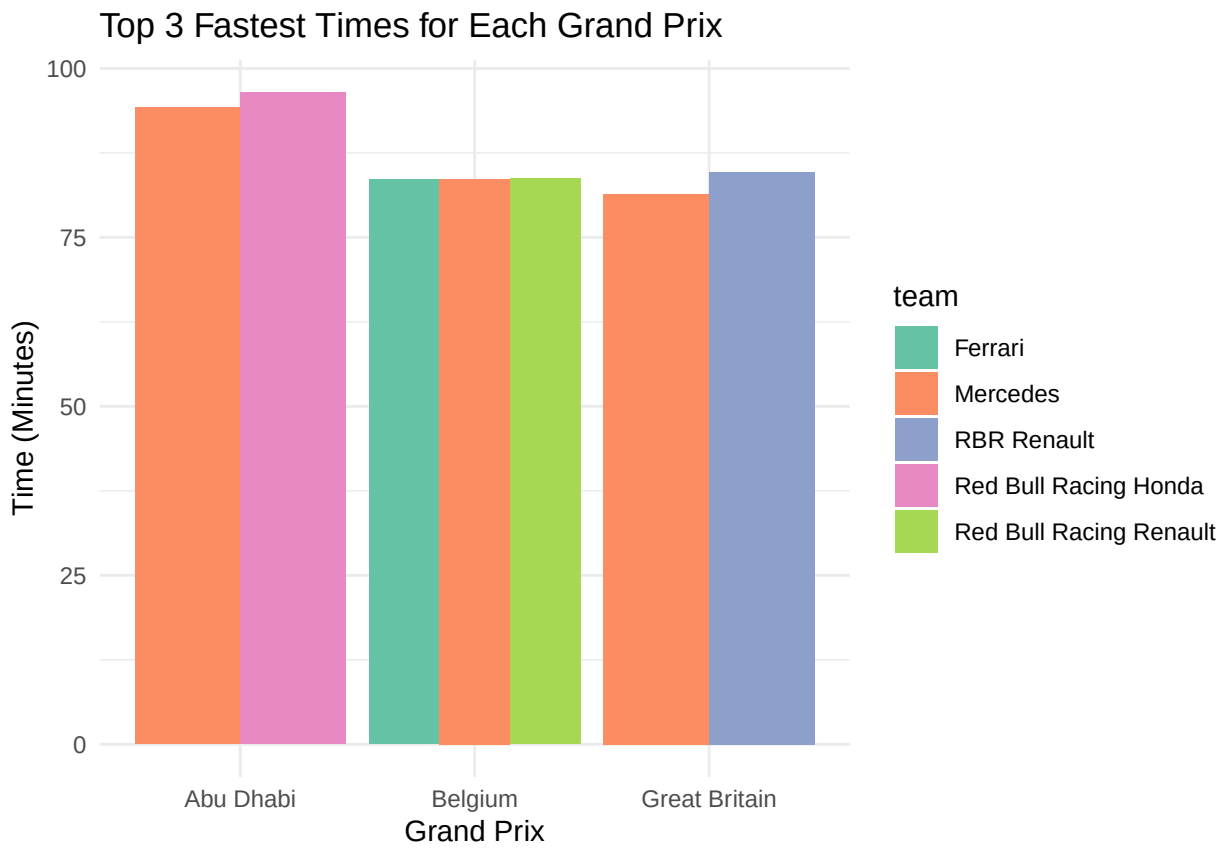
```
## time (1): time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
mydata <- mydata %>%
  mutate(time_minutes = as.numeric(hms(time)) / 60,
         date = dmy(date))

filtered_data <- mydata %>%
  filter(grand_prix %in% c("Abu Dhabi", "Great Britain", "Belgium"))

avg_performance <- filtered_data %>%
  group_by(grand_prix, date, team) %>%
  summarize(avg_time_minutes = mean(time_minutes, na.rm = TRUE), .groups = "drop")

avg_performance <- avg_performance %>%
  arrange(grand_prix, avg_time_minutes)

print(avg_performance)
```

```
## # A tibble: 33 x 4
##    grand_prix date       team                    avg_time_minutes
##    <chr>      <date>     <chr>                              <dbl>
##  1 Abu Dhabi  2019-12-01 Mercedes                            94.1
##  2 Abu Dhabi  2017-11-26 Mercedes                            94.2
##  3 Abu Dhabi  2020-12-13 Red Bull Racing Honda               96.5
##  4 Abu Dhabi  2011-11-13 McLaren Mercedes                    97.2
##  5 Abu Dhabi  2016-11-27 Mercedes                            98.1
##  6 Abu Dhabi  2013-11-03 Red Bull Racing Renault             98.1
##  7 Abu Dhabi  2015-11-29 Mercedes                            98.5
##  8 Abu Dhabi  2014-11-23 Mercedes                            99.0
##  9 Abu Dhabi  2010-11-14 RBR Renault                         99.6
## 10 Abu Dhabi  2018-11-25 Mercedes                            99.7
## # i 23 more rows
```

Step 3: For comparative measures, we will begin to implement the three fastest times for each circuit into a bar graph to visually depict the discrepancy between the two eras being studied.

```
top3_times <- filtered_data %>%
  group_by(grand_prix) %>%
  arrange(time_minutes) %>%
  slice(1:3)  # Selects the three fastest times

# Creating the bar chart
ggplot(top3_times, aes(x = grand_prix, y = time_minutes, fill = team)) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(title = "Top 3 Fastest Times for Each Grand Prix",
       x = "Grand Prix",
       y = "Time (Minutes)") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")
```



Discussion:

Based on the given steps above, these were the necessary steps in order to take our data from the original quantity down to the newly cleaned and easily accessible data quantity. To *briefly summarize*, we began diluting the data from our first loaded data and then began issuing conversion calls, such as for time, to give a better comparison between the two era of cars being operated. Along with this data filtering would be applied as well as the reordering and reconstruction of our selected data. Lastly, we were able to visualize these findings in a simple, yet effective, bar graph. Fortunately, no data was found to hold NA or null values; although, one major complication I didn't foresee was the multiple categorization of the Red Bull team under multiple names throughout the years. Simplifying this data by filtering it into just one team would prove to be more ideal and potentially still allow a fair comparison.

Exploratory Data Analysis

In this section, we will begin to dive a little deeper in the comparative aspects and begin to implement graphs such as a histogram and a correlation matrix. Along with this the discussion will be posted to analyze our findings and/or any complications.

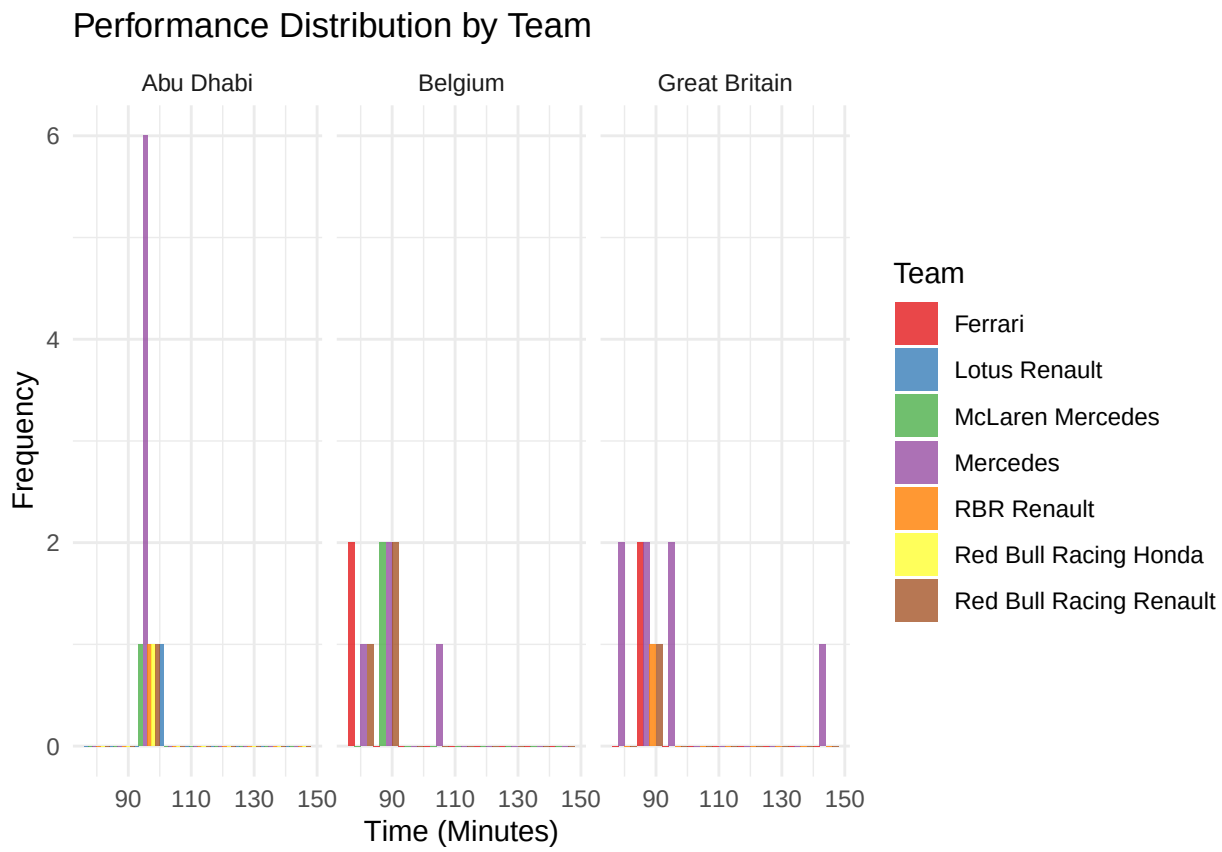
Step 1: Implement a histogram

```
mydata <- read_csv("~/Downloads/f1_2010-2021.csv")
```

```
## Rows: 237 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): index, team, code, name, surname, grand_prix, date
## dbl  (2): level_0, laps
## time (1): time
```

```
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```r
# Creating the histogram from the filtered data above
ggplot(filtered_data, aes(x = time_minutes, fill = team)) +
  geom_histogram(binwidth = 8, position = "dodge", alpha = 0.8) +
  labs(title = "Performance Distribution by Team",
       x = "Time (Minutes)",
       y = "Frequency",
       fill = "Team") +
  facet_wrap(~ grand_prix) +
  scale_fill_brewer(palette = "Set1") +
  theme_minimal()
```



Step 2: Begin to implement a correlation matrix to reintroduce years into the chart

```r
mydata <- read_csv("~/Downloads/f1_2010-2021.csv")
```

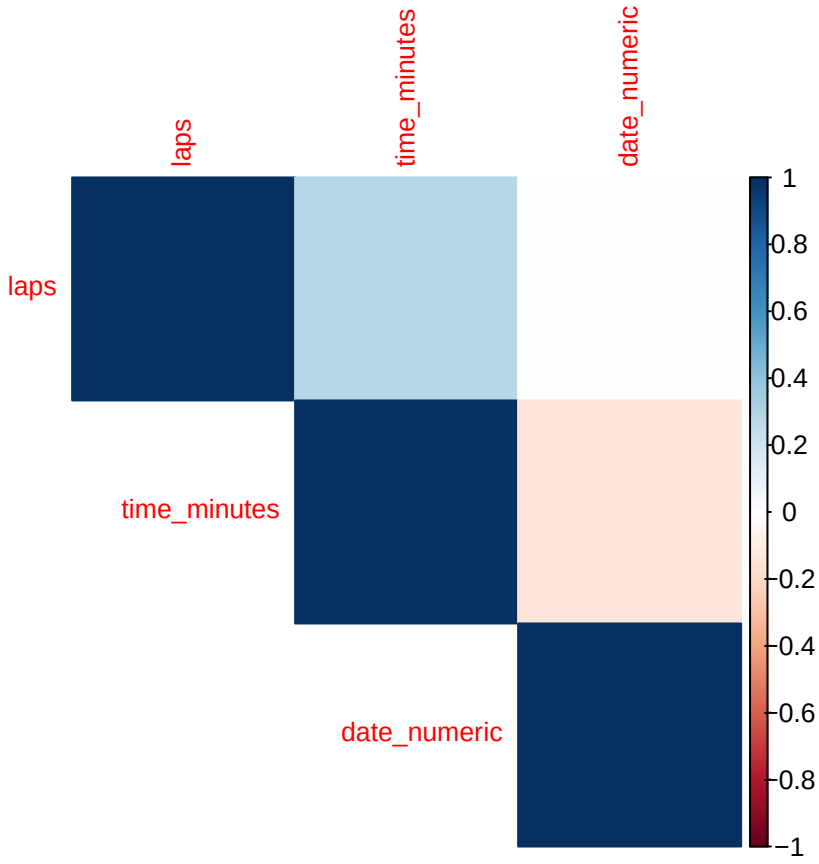
```
## Rows: 237 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): index, team, code, name, surname, grand_prix, date
## dbl  (2): level_0, laps
## time (1): time
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
mydata <- mydata %>%
  mutate(time_minutes = as.numeric(hms(time)) / 60,
         date = dmy(date))

mydata <- mydata %>%
  select(laps, time_minutes, date) %>%
  mutate(date_numeric = as.numeric(date))

cor_matrix <- cor(mydata %>% select(-date))
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(cor_matrix, method = "color", type = "upper", tl.cex = 0.8)
```



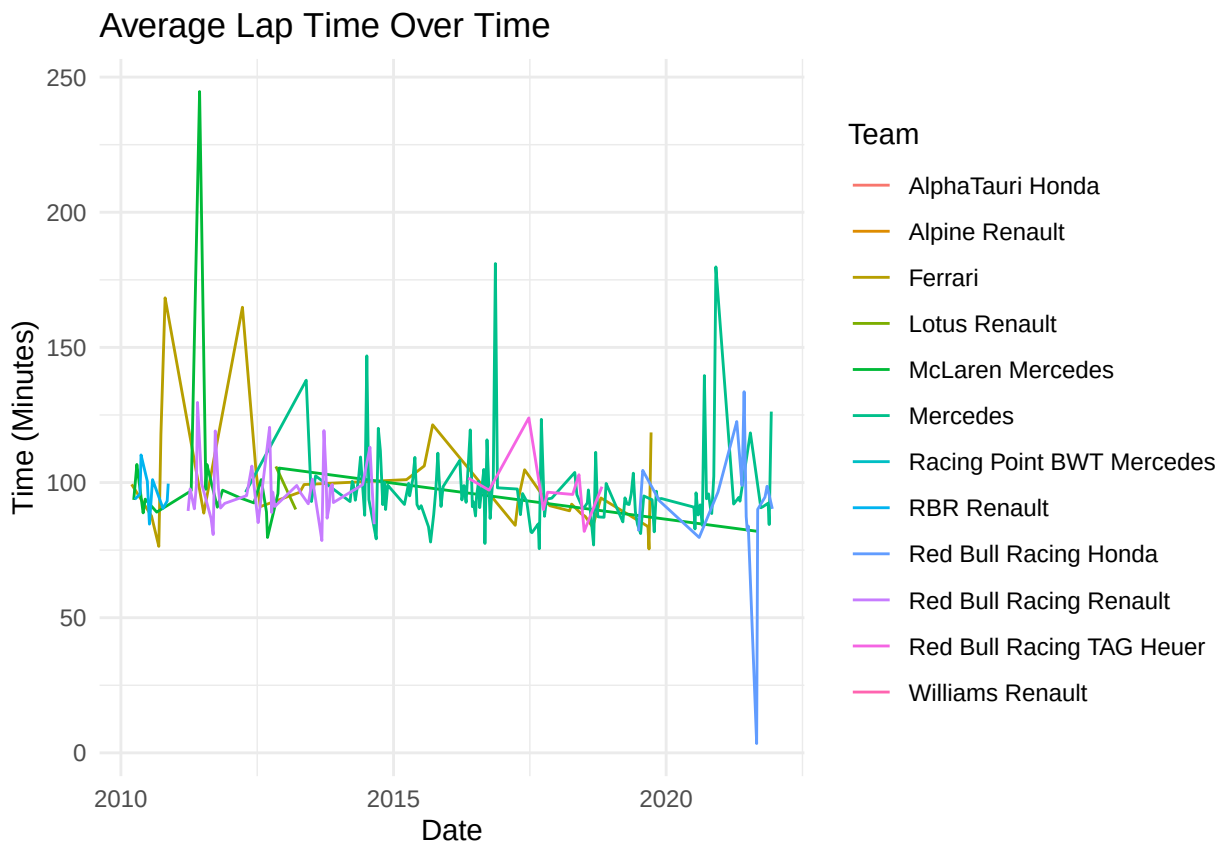
Step 3: Let's implement a fairly simple line chart to illustrate the overall progression between the difference seasons and the given performance metrics.

```
mydata <- read_csv("~/Downloads/f1_2010-2021.csv")
```

```
## Rows: 237 Columns: 10
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): index, team, code, name, surname, grand_prix, date
## dbl  (2): level_0, laps
## time (1): time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
mydata <- mydata %>%
  mutate(time_minutes = as.numeric(hms(time)) / 60,
         date = dmy(date))
avg_lap_time <- mydata %>%
  group_by(date, team) %>%
  summarize(avg_time_minutes = mean(time_minutes, na.rm = TRUE), .groups = "drop")

# Creating a line chart
ggplot(avg_lap_time, aes(x = date, y = avg_time_minutes, color = team)) +
  geom_line() +
  labs(title = "Average Lap Time Over Time",
       x = "Date",
       y = "Time (Minutes)",
       color = "Team") +
  theme_minimal()
```

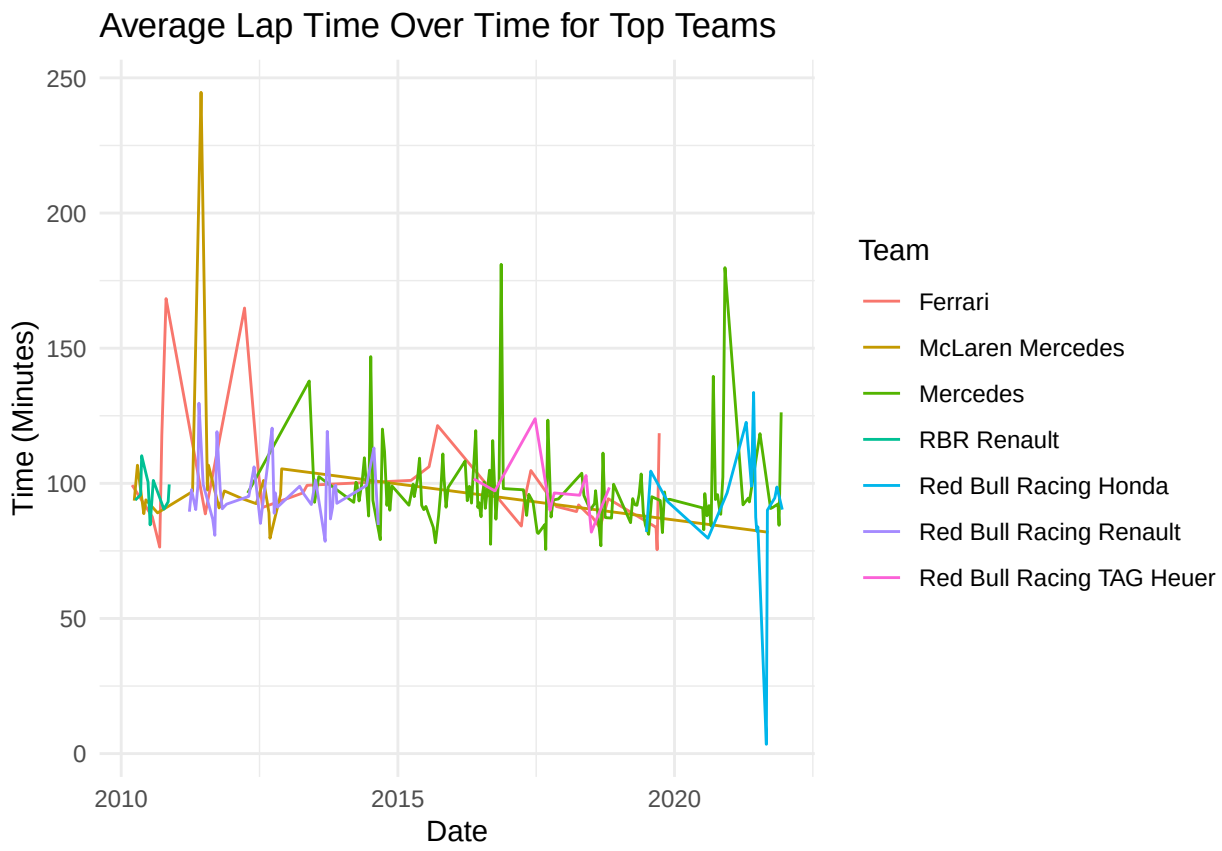


Step 4: Refine the given data to reflect only the dominant teams during this era

```
mydata <- read_csv("~/Downloads/f1_2010-2021.csv")
```

```
## Rows: 237 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): index, team, code, name, surname, grand_prix, date
## dbl  (2): level_0, laps
## time (1): time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
mydata <- mydata %>%
  mutate(time_minutes = as.numeric(hms(time)) / 60,
         date = dmy(date))
#Filtering out to gain perspective into the evolution of the dominant teams
filtered_data <- mydata %>%
  filter(team %in% c("RBR Renault", "Red Bull Racing TAG Heuer",
                     "Red Bull Racing Honda", "Red Bull Racing Renault",
                     "McLaren Mercedes", "Mercedes", "Ferrari"))

avg_lap_time <- filtered_data %>%
  group_by(date, team) %>%
  summarize(avg_time_minutes = mean(time_minutes, na.rm = TRUE), .groups = "drop")

ggplot(avg_lap_time, aes(x = date, y = avg_time_minutes, color = team)) +
  geom_line() +
  labs(title = "Average Lap Time Over Time for Top Teams",
       x = "Date",
       y = "Time (Minutes)",
       color = "Team") +
  theme_minimal()
```



EDA Discussion:

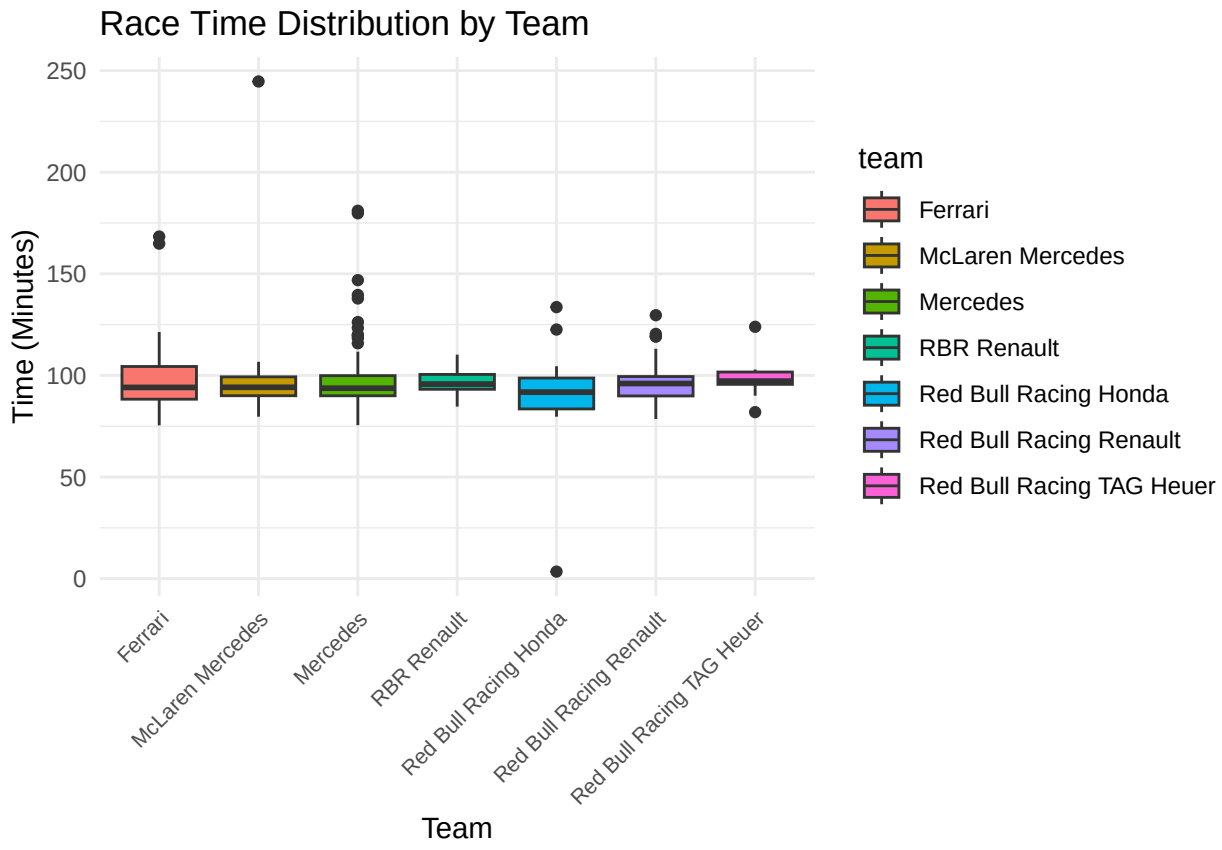
Upon speculation of the found data we can come up with an insight on multiple perspectives and conclusions to our main goal of answering whether or not the different generations were truly faster and more dominant. Based on the shown data we can begin to dissect that there are a definite sign of trends and patterns amongst the performance metrics, but not necessarily a vast difference. With the exclusion of the two biggest outliers, that of the McLaren Mercedes spike and the Red Bull Racing Honda spike we begin to understand that the cars are very similarly paced despite having two completely different engines. One key note to point out would be the Mercedes slight decline in time during the 2020 season as that spec (the W11) is factually the most dominant Formula 1 car ever produced as it handled straight-line speed, cornering, and other race performances extremely well.

Although, one huge discrepancy between this period was the impact Covid-19 had on the season as the race quantity dipped down to only 17 races instead of the 21+ races that normally occur. With this being noted, bias may be introduced as the cost in the low variations of races would show one way or another. Additionally, looking back on the correlation matrix created, it's *very crucial* to observe that the drawn conclusion of the V8 vs. the V6 engines can also be noted in this matrix, specifically in the "time_minutes" - "date_numeric" box as the red tint applied demonstrates there isn't a strong correlation present. This is great news for the circumstances this project is built off of as we now have multiple areas of sound evidence that show the progression of Formula 1 is not seeing reduced metrics in performances.

It's also important to note there is a strong sense of collinearity as discussed earlier with the correlation matrix. As opposed to the non-collinearity of the "time_minutes" - "date_numeric" functions we can depict that the categories of "laps" - "time_minutes" have a fairly positive corelation to each other, which is statistically and graphically just.

Results And Analysis:

```r
filtered_data <- mydata %>%
  filter(team %in% c("RBR Renault", "Red Bull Racing TAG Heuer",
                     "Red Bull Racing Honda", "Red Bull Racing Renault",
                     "McLaren Mercedes", "Mercedes", "Ferrari"))

ggplot(filtered_data, aes(x = team, y = time_minutes, fill = team)) +
  geom_boxplot() +
  labs(title = "Race Time Distribution by Team",
       x = "Team",
       y = "Time (Minutes)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, size = 8))
```




```r
set.seed(42)
train_indices <- sample(1:nrow(mydata), size = 0.8 * nrow(mydata))
train_data <- mydata[train_indices,]
test_data <- mydata[-train_indices,]

lm_model <- lm(time_minutes ~ laps + date, data = train_data)

predictions <- predict(lm_model, newdata = test_data)

actuals <- test_data$time_minutes
rmse <- sqrt(mean((actuals - predictions) ^ 2))
mae <- mean(abs(actuals - predictions))
```

```r
print(paste("Root Mean Square Error (RMSE):", rmse))
```

```
## [1] "Root Mean Square Error (RMSE): 11.6485835340967"
```

```r
print(paste("Mean Absolute Error (MAE):", mae))
```

```
## [1] "Mean Absolute Error (MAE): 9.02791213418723"
```

Overall, the main objective of this project was to serve as a comparative measure into two vastly different eras in Formula 1 from a mechanical perspective. By doing this, we needed to create a model to filter out excess amounts of data found from the original data base that spanned to 237 rows by itself. Throughout fine-tuning this model we are able to easily access needed data to implement factual and most of all reliable data. Although the data for the most part is sound from any unwanted noise, one unforeseen issue that could impact the data slightly would be the alterations of the track layouts themselves. This would be highly improbable to account for as the data supplied would not be able to differentiate if the track layout was altered from season to season. Additionally, this wouldn't skew the lap time data too much but instead vary from seconds. The outliers already presented would be fairly similar as well as the other data supplied proving this concern to essentially be negligible.

In the given boxplot above not only are we able to see the top performing teams of this time (from longevity stand point), but we are able to observe the standard deviations and how greatly they vary or how little. It's important to note that this model does not combine the Red Bull teams as one as from preliminary thoughts, it is presumed to be best to add the variation of the teams to not leave out any important speculation. Although, it is to be presumed the data could arguably be more stronger summed together as the variation would increase and the bias interpretation may decrease. This is why this boxplot metric was selected to be utilized.

The training and evaluation process of this project encompassed classical data preparation tactics such as cleaning the data, adding new components (i.e. duration of the races based on lap times initially supplied) which coincides with *feature engineering*, as well as splitting the data to ensure there is no over-fitting. This model could in turn be used as a predictive measure as the results show the cars behave in a similar pattern which can be presumed to be applied to the newer generation cars of the 2020's.

This *regression model* is tested by utilizing both *Root Mean Square Error (RMSE)* and *Mean Absolute Error (MSE)*. The resulting values yield the values for Root Mean Square Error (RMSE): 11.65 and the Mean Absolute Error (MAE): 9.03. These values are moderate and prove the overall the model is able to be used potentially as a predictive measure; but, these same scores also indicate the model could be improved more to become more efficient.

Discussion/Conclusion Of The Findings

The model itself performed well but, as mentioned, could be improved upon in working more efficiently for future comparison tests which would improve both the RMSE and MAE values. This model helps users learn about the statistical values of the two Formula 1 eras at hand and provides substantial evidence in performance metrics to cater to the philosophy of wondering how big the discrepancy was between two main generations of Formula 1 cars. Ways that this project could be improved upon would be to compile the data scripts into a more condensed fashion than dissecting each and every step into different mini-steps. In order to expand upon this project to incorporate the newer generation cars, additional linear modeling via regression modeling could suit well in creating a more *predictive model*. The feature engineering of implementing key factors also played a critical role in establishing the project's foundation. Conclusively, this model revolves around ML key factors of EDA, feature engineering, model evaluation as well as regression/comparative modeling.