Formula 1 Algorithm Analyzation Project

Gabriel Downs

2024-06-24

A Deep Unsupervised Algorithm 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 that will be utilized within this project circumnavigates the key concepts of unsupervised learning. The methods and models that will be utilized are standardizing the data, K-means clustering, some Hierarchical clustering, and principal component analysis (PCA).

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)
library(pheatmap)
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, Great Britain, Hungary, Italy, and Spain we can enhance our code to begin exempting the lower performing teams based on time. This will give us a more enhanced look at potentially observing any underlying patterns that we can expand on.

```
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", "Hungary",
                           "Italy", "Spain"))

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)
```

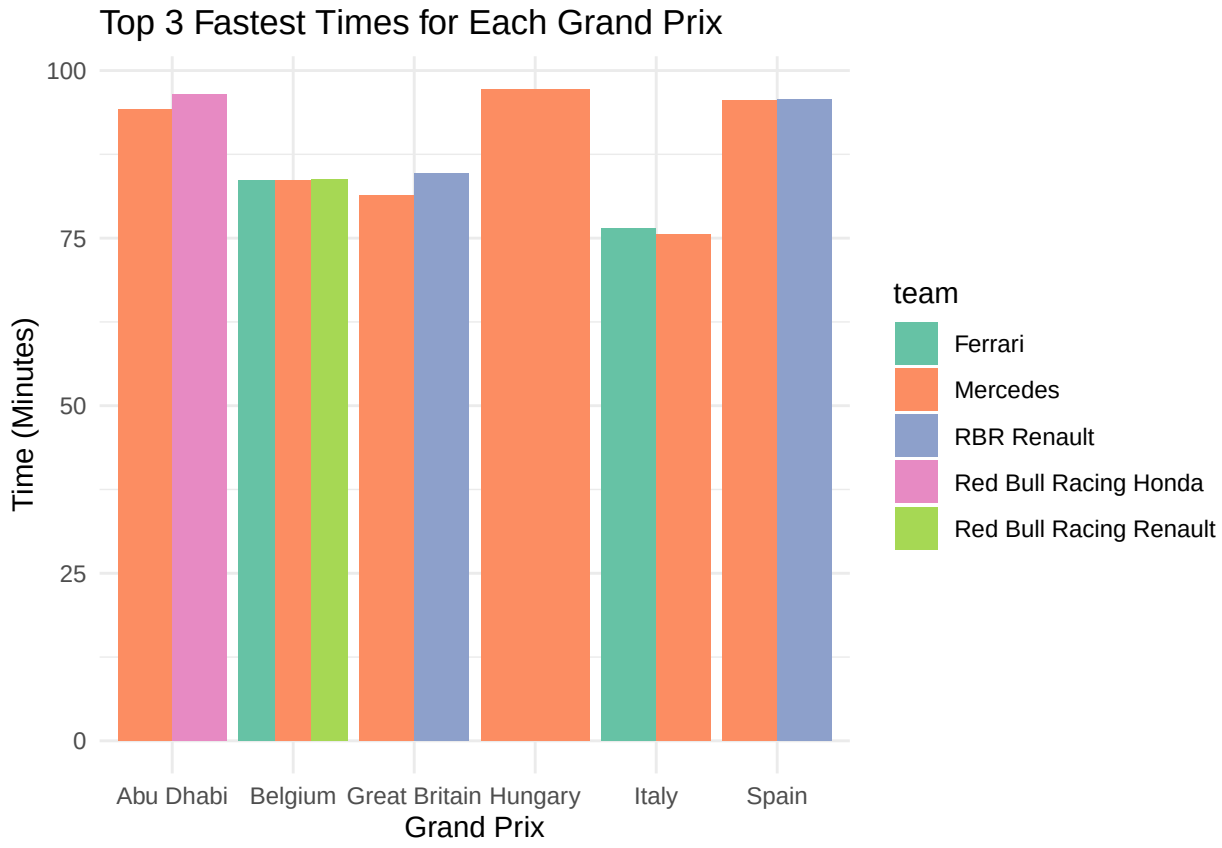
```
## # A tibble: 66 x 4
##    grand_prix date       team                    avg_time_minutes
##    <chr>      <date>     <chr>                              <dbl>
##  1 Abu Dhabi  2019-12-01 Mercedes                            94.1
##  2 Abu Dhabi  2017-11-26 Mercedes                            94.2
##  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 56 more rows
```

Another additional note to mention is that these circuits, although not the same, are actual race circuits instead of street circuits. This is an important feature to implement because the speed at which the cars carry throughout each location will be more synonymous with one another than if there were a street circuit.

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

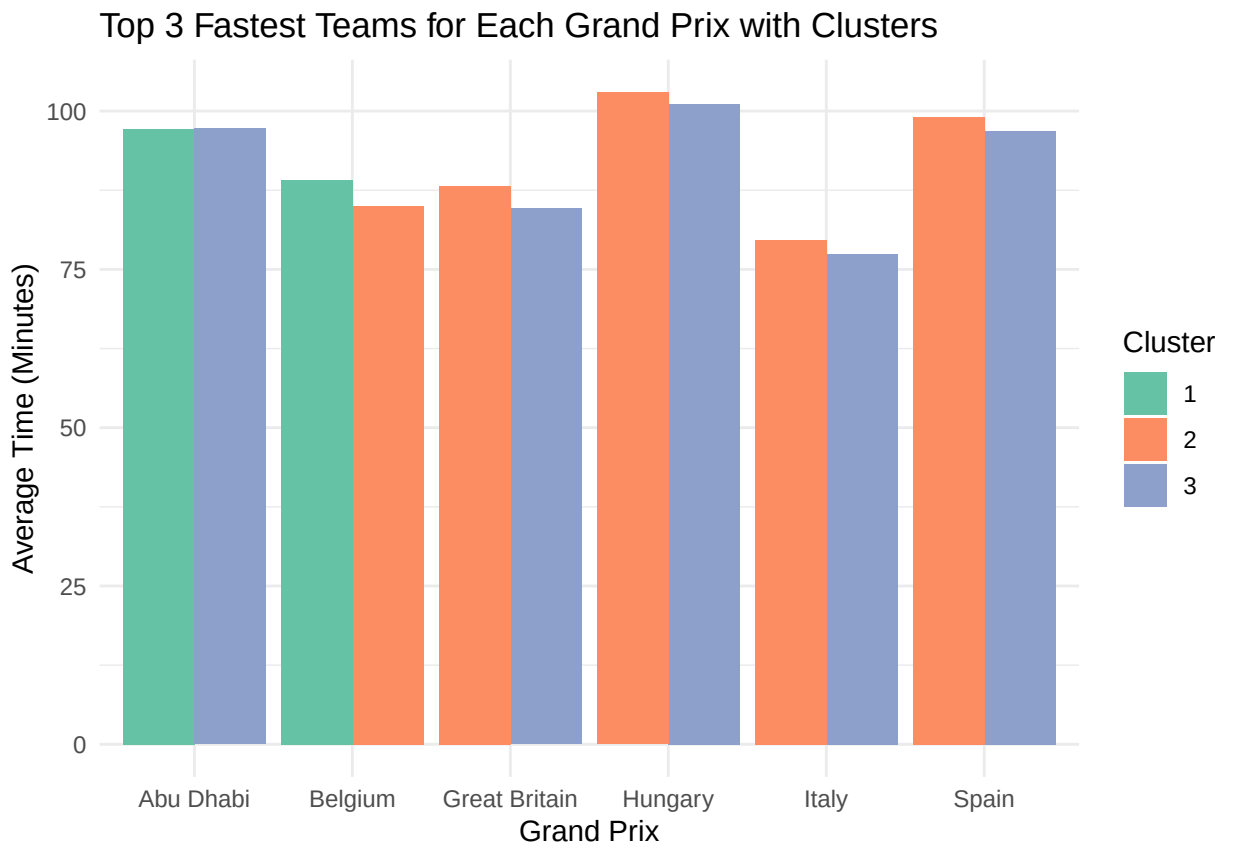
In this section, we will begin to dive a little deeper in the comparative aspects and begin to implement graphs depicting K-means clustering and hierarchical clustering. The K-means clustering will be important to implement because it'll allow us to observe average positions of the teams which is a crucial performance metric. For the hierarchical clustering, I hope it brings in data supporting the possibility of bringing in performance similarities/dissimilarities for the teams.

Step 1: Implement a K-Means Cluster with the original code

```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.
```

```r
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", "Hungary",
                           "Italy", "Spain"))

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

top3_times <- avg_performance %>%
  arrange(grand_prix, avg_time_minutes) %>%
  group_by(grand_prix) %>%
  slice(1:3)

performance_matrix <- top3_times %>%
  pivot_wider(names_from = grand_prix, values_from = avg_time_minutes) %>%
  replace(is.na(.), 0) %>%
  column_to_rownames("team")

performance_scaled <- scale(performance_matrix)

set.seed(42)
kmeans_result <- kmeans(performance_scaled, centers = 3)

top3_times <- top3_times %>%
  mutate(cluster = kmeans_result$cluster[match(team, rownames(performance_matrix))])

ggplot(top3_times, aes(x = grand_prix, y = avg_time_minutes, fill = as.factor(cluster))) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(title = "Top 3 Fastest Teams for Each Grand Prix with Clusters",
       x = "Grand Prix",
       y = "Average Time (Minutes)",
       fill = "Cluster") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")
```




```
print(top3_times)
```

```
## # A tibble: 18 x 4
## # Groups:   grand_prix [6]
##    grand_prix    team                    avg_time_minutes cluster
##    <chr>         <chr>                              <dbl>   <int>
##  1 Abu Dhabi     Red Bull Racing Honda               96.5       1
##  2 Abu Dhabi     McLaren Mercedes                    97.2       1
##  3 Abu Dhabi     Mercedes                            97.3       3
##  4 Belgium       Ferrari                             83.7       2
##  5 Belgium       Red Bull Racing Renault             85.0       2
##  6 Belgium       McLaren Mercedes                    89.1       1
##  7 Great Britain RBR Renault                         84.6       3
##  8 Great Britain Red Bull Racing Renault             85.2       2
##  9 Great Britain Ferrari                             88.1       2
## 10 Hungary       Mercedes                            98.3       3
## 11 Hungary       RBR Renault                        101.        3
## 12 Hungary       Ferrari                            103.        2
## 13 Italy         Ferrari                             75.9       2
## 14 Italy         Mercedes                            77.4       3
## 15 Italy         Red Bull Racing Renault             79.7       2
## 16 Spain         RBR Renault                         95.7       3
## 17 Spain         Mercedes                            96.9       3
## 18 Spain         Red Bull Racing Renault             99.0       2
```

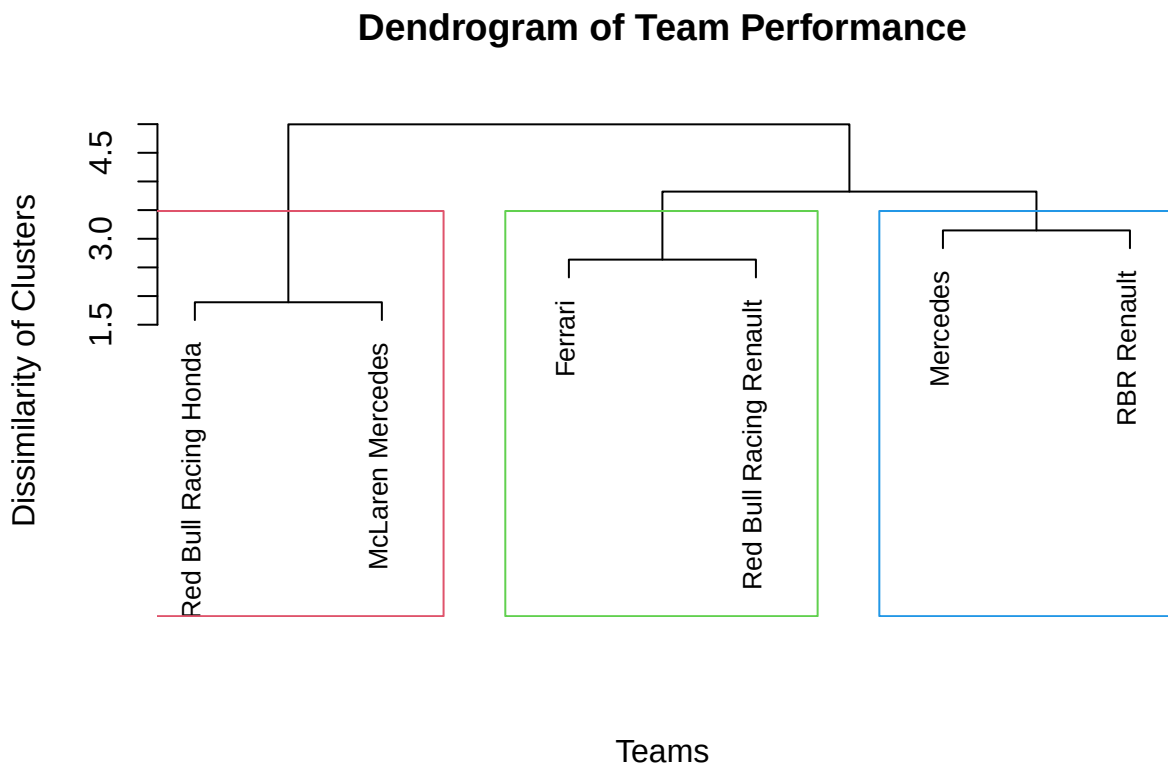
Step 2: Begin to implement a hierarchical cluster to determine any other patterns or correlations brought

by the teams within the decades worth of data.

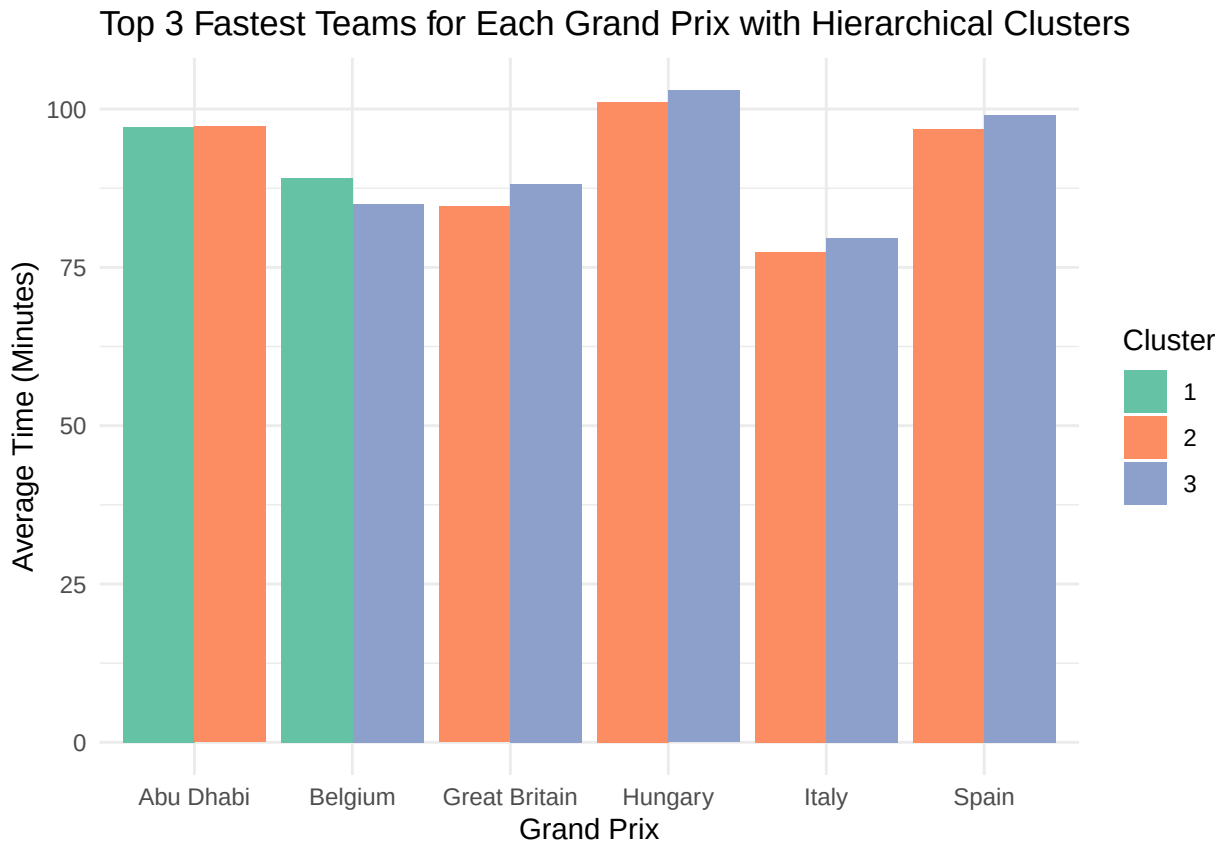
```
mydata <- read_csv("~/Downloads/f1_2010-2021.csv")
```

```
## 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", "Hungary",
                           "Italy", "Spain"))

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

top3_times <- avg_performance %>%
  arrange(grand_prix, avg_time_minutes) %>%
  group_by(grand_prix) %>%
  slice(1:3)

performance_matrix <- top3_times %>%
  pivot_wider(names_from = grand_prix, values_from = avg_time_minutes) %>%
  replace(is.na(.), 0) %>%  # Replace NA values with 0
  column_to_rownames("team")

performance_scaled <- scale(performance_matrix)

dist_matrix <- dist(performance_scaled, method = "euclidean")
hc <- hclust(dist_matrix, method = "ward.D2")

plot(hc, labels = rownames(performance_matrix), main = "Dendrogram of Team Performance",
     xlab = "Teams", ylab = "Dissimilarity of Clusters", sub = "", cex = 0.8)
rect.hclust(hc, k = 3, border = 2:4)
```




```r
clusters <- cutree(hc, k = 3)
top3_times <- top3_times %>%
  mutate(cluster = clusters[match(team, rownames(performance_matrix))])

ggplot(top3_times, aes(x = grand_prix, y = avg_time_minutes, fill = as.factor(cluster))) +
  geom_bar(stat = "identity", position = position_dodge()) +
  labs(title = "Top 3 Fastest Teams for Each Grand Prix with Hierarchical Clusters",
       x = "Grand Prix",
       y = "Average Time (Minutes)",
       fill = "Cluster") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")
```




```
print(top3_times)
```

```
## # A tibble: 18 x 4
## # Groups:   grand_prix [6]
##    grand_prix    team                    avg_time_minutes cluster
##    <chr>         <chr>                              <dbl>   <int>
##  1 Abu Dhabi     Red Bull Racing Honda               96.5       1
##  2 Abu Dhabi     McLaren Mercedes                    97.2       1
##  3 Abu Dhabi     Mercedes                            97.3       2
##  4 Belgium       Ferrari                             83.7       3
##  5 Belgium       Red Bull Racing Renault             85.0       3
##  6 Belgium       McLaren Mercedes                    89.1       1
##  7 Great Britain RBR Renault                         84.6       2
##  8 Great Britain Red Bull Racing Renault             85.2       3
##  9 Great Britain Ferrari                             88.1       3
## 10 Hungary       Mercedes                            98.3       2
## 11 Hungary       RBR Renault                        101.        2
## 12 Hungary       Ferrari                            103.        3
## 13 Italy         Ferrari                             75.9       3
## 14 Italy         Mercedes                            77.4       2
## 15 Italy         Red Bull Racing Renault             79.7       3
## 16 Spain         RBR Renault                         95.7       2
## 17 Spain         Mercedes                            96.9       2
## 18 Spain         Red Bull Racing Renault             99.0       3
```

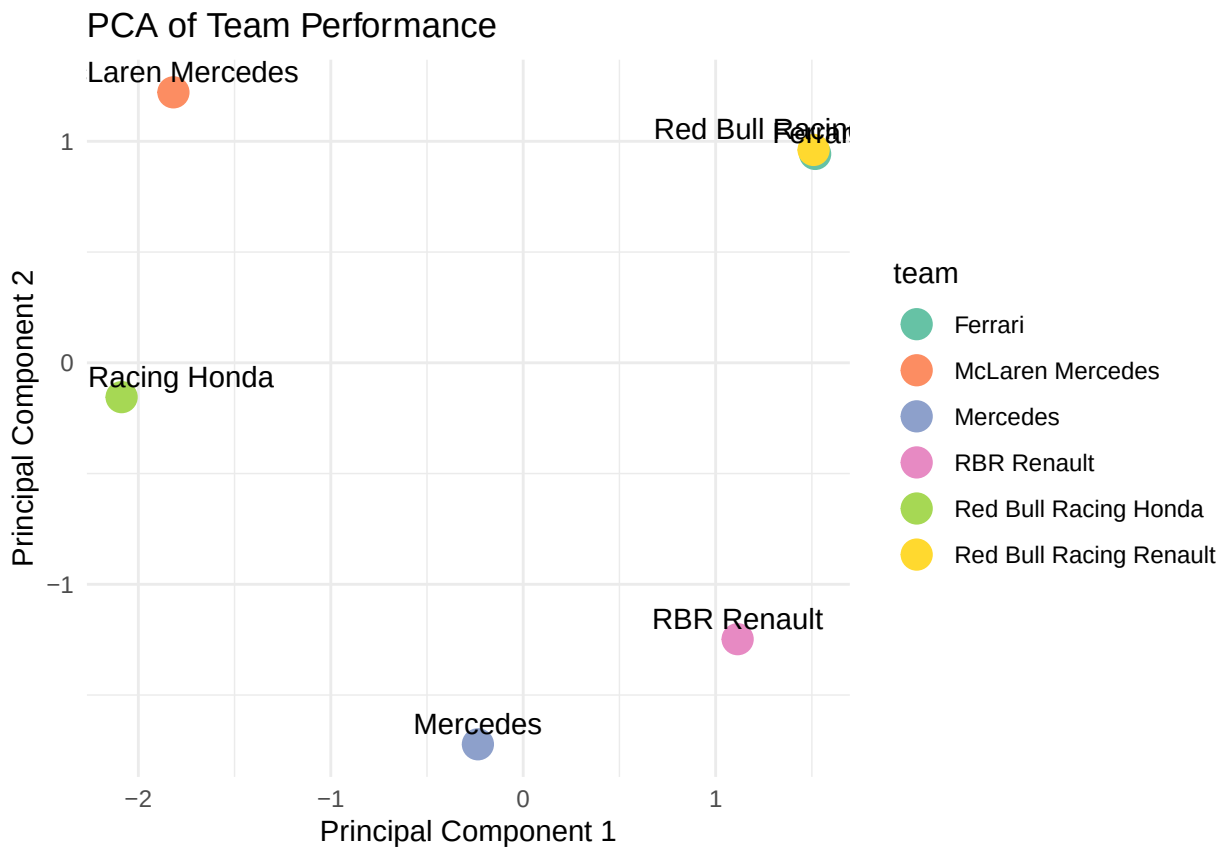
Step 3: Let's implement a principal component analysis (PCA) in the form of a scatter plot to compare the

teams performances with the different circuits. The clusters we will focus on will be the first and second. Justification for this selection is because statistically these teams are more similar to one another as shown on the hierarchy and K-means cluster graph.

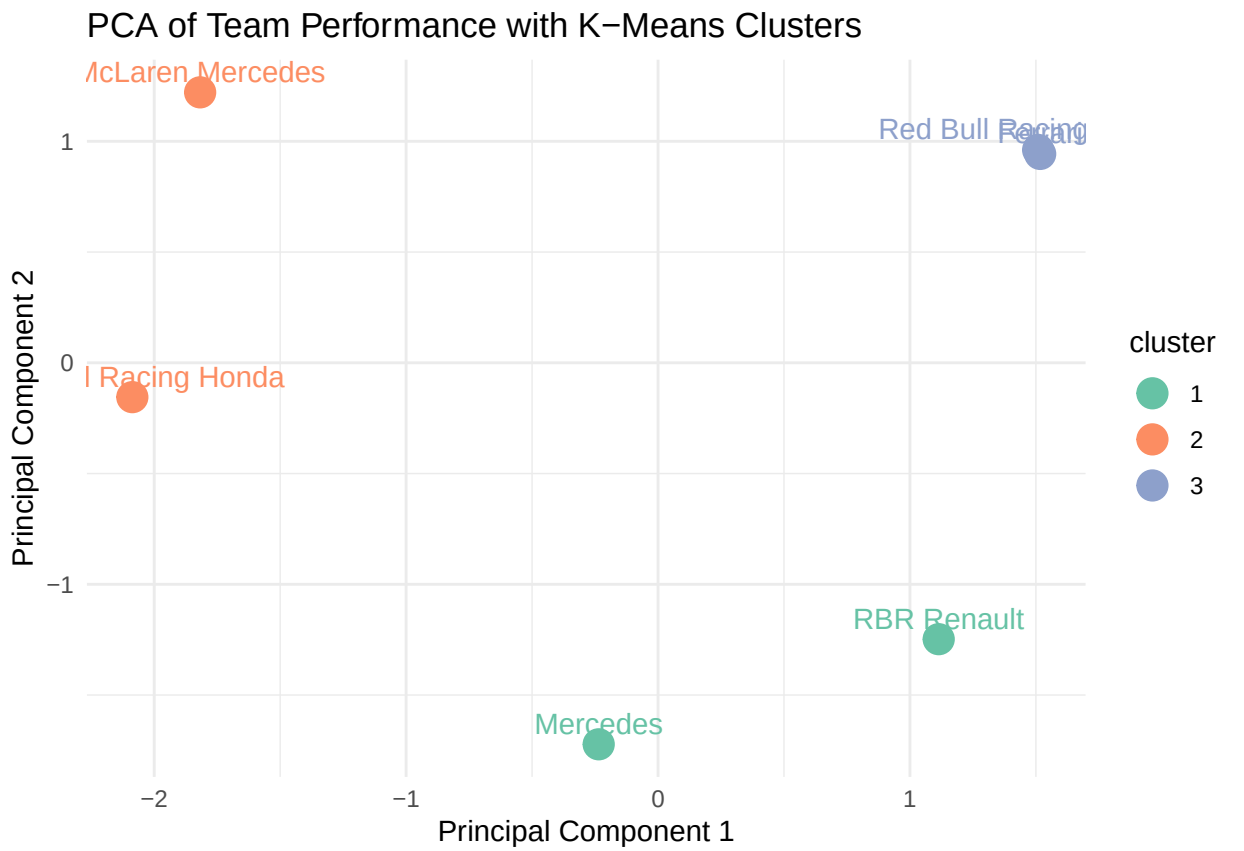
```
mydata <- read_csv("~/Downloads/f1_2010-2021.csv")
```

```
## 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", "Hungary",
                           "Italy", "Spain"))

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

top3_times <- avg_performance %>%
  arrange(grand_prix, avg_time_minutes) %>%
  group_by(grand_prix) %>%
  slice(1:3)

performance_matrix <- top3_times %>%
  pivot_wider(names_from = grand_prix, values_from = avg_time_minutes) %>%
  replace(is.na(.), 0) %>%
  column_to_rownames("team")

performance_scaled <- scale(performance_matrix)

pca_result <- prcomp(performance_scaled, center = TRUE, scale. = TRUE)

pca_scores <- data.frame(pca_result$x)
pca_scores$team <- rownames(pca_scores)

ggplot(pca_scores, aes(x = PC1, y = PC2, label = team)) +
  geom_point(aes(color = team), size = 5) +
  geom_text(vjust = -0.5, hjust = 0.5) +
  labs(title = "PCA of Team Performance", x = "Principal Component 1", y = "Principal Component 2") +
  theme_minimal() +
  scale_color_brewer(palette = "Set2")
```




```r
kmeans_result <- kmeans(performance_scaled, centers = 3)
pca_scores$cluster <- factor(kmeans_result$cluster)

ggplot(pca_scores, aes(x = PC1, y = PC2, color = cluster, label = team)) +
  geom_point(size = 5) +
  geom_text(vjust = -0.5, hjust = 0.5) +
  labs(title = "PCA of Team Performance with K-Means Clusters", x = "Principal Component 1", y = "Princ
  theme_minimal() +
  scale_color_brewer(palette = "Set2")
```



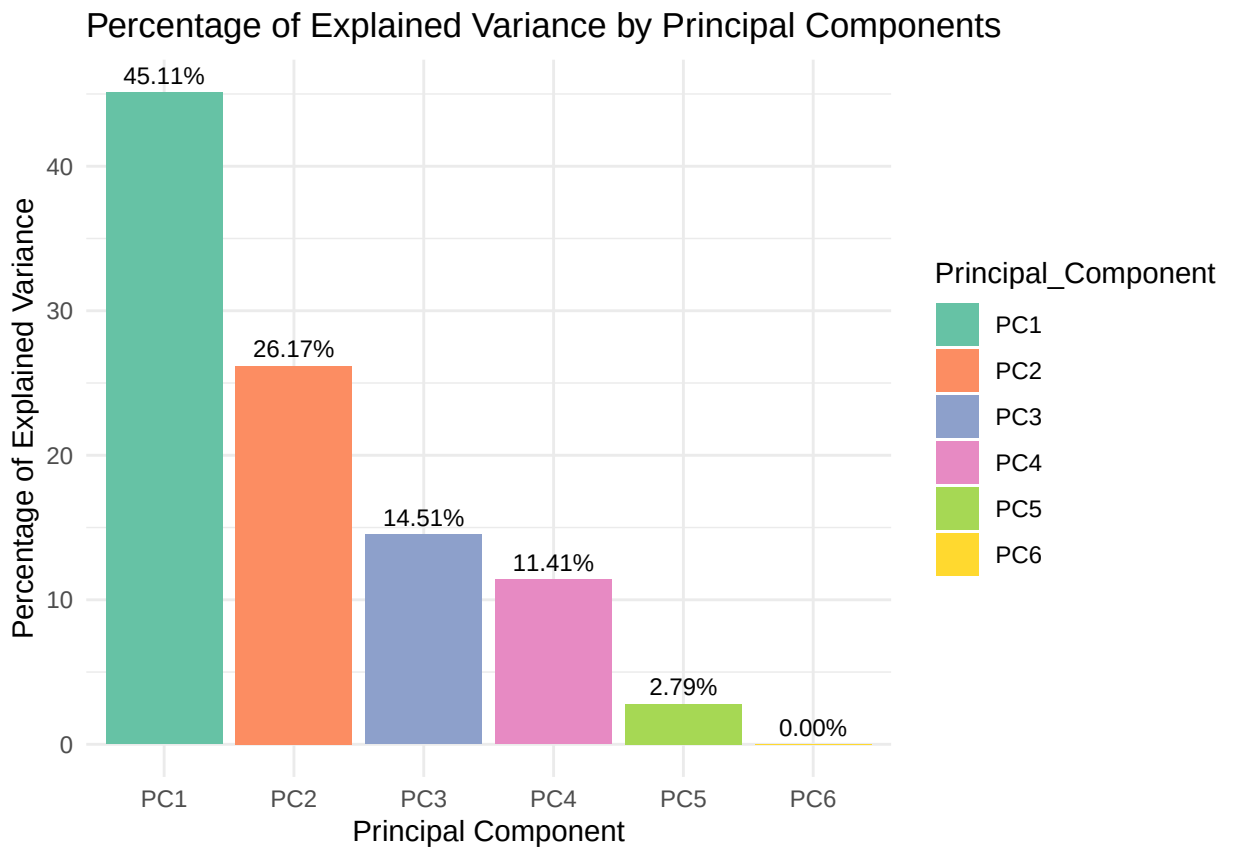
Step 4: Refine the given data to reflect a barplot via the barplot function to plot the percentages of the variations in the data.

```
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", "Hungary",
                           "Italy", "Spain"))

avg_performance <- filtered_data %>%
  group_by(grand_prix, team) %>%
```

```r
  summarize(avg_time_minutes = mean(time_minutes, na.rm = TRUE), .groups = "drop")

top3_times <- avg_performance %>%
  arrange(grand_prix, avg_time_minutes) %>%
  group_by(grand_prix) %>%
  slice(1:3)

performance_matrix <- top3_times %>%
  pivot_wider(names_from = grand_prix, values_from = avg_time_minutes) %>%
  replace(is.na(.), 0) %>%
  column_to_rownames("team")

performance_scaled <- scale(performance_matrix)

pca_result <- prcomp(performance_scaled, center = TRUE, scale. = TRUE)

explained_variance <- pca_result$sdev^2 / sum(pca_result$sdev^2) * 100

explained_variance_df <- data.frame(
  Principal_Component = paste0("PC", 1:length(explained_variance)),
  Explained_Variance = explained_variance
)

ggplot(explained_variance_df, aes(x = Principal_Component, y = Explained_Variance, fill = Principal_Comp
  geom_bar(stat = "identity") +
  geom_text(aes(label = sprintf("%.2f%%", Explained_Variance)),
            vjust = -0.5, size = 3) +
  labs(title = "Percentage of Explained Variance by Principal Components",
       x = "Principal Component",
       y = "Percentage of Explained Variance") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set2")
```



EDA Discussion:

During the process of this analysis our goals were to understand the correlation between the dominant teams within the 2010s time period (11 years precisely) and determine the dissimilarity between the model cars (teams). The data source (referenced above) takes into account a multitude of teams that were competing during this time period in which the variables utilized revolved around time metrics such as 'time' and 'date' as well as other metrics such as 'grand_prix' and 'team'. Data cleaning and preparation methods were conducted by omitting any "NA" values. Additionally, the 'time' valuables were given in hours initially but subsequently converted into minutes to better analyze the data. Uni-variate analysis was conducted by analyzing the distribution of the lap times presented by each circuit/team. Multivariate analysis was conducted within the methods of K-means clustering, hierarchy clustering, and principal component analysis (PCA).

K-Means clustering presented value by compiling teams that performed similarly. While, hierarchy clustering presented performance relationships that were prevalent in the dendrogram model of team performances. Finally, PCA allowed the viewing to show an in-depth driving performance graph. Within the *data visualization* process we were able to depict the exact variation specifications by utilizing the PCA findings and combining it with standard deviation to then extract percentages correlated to each of the principal components.

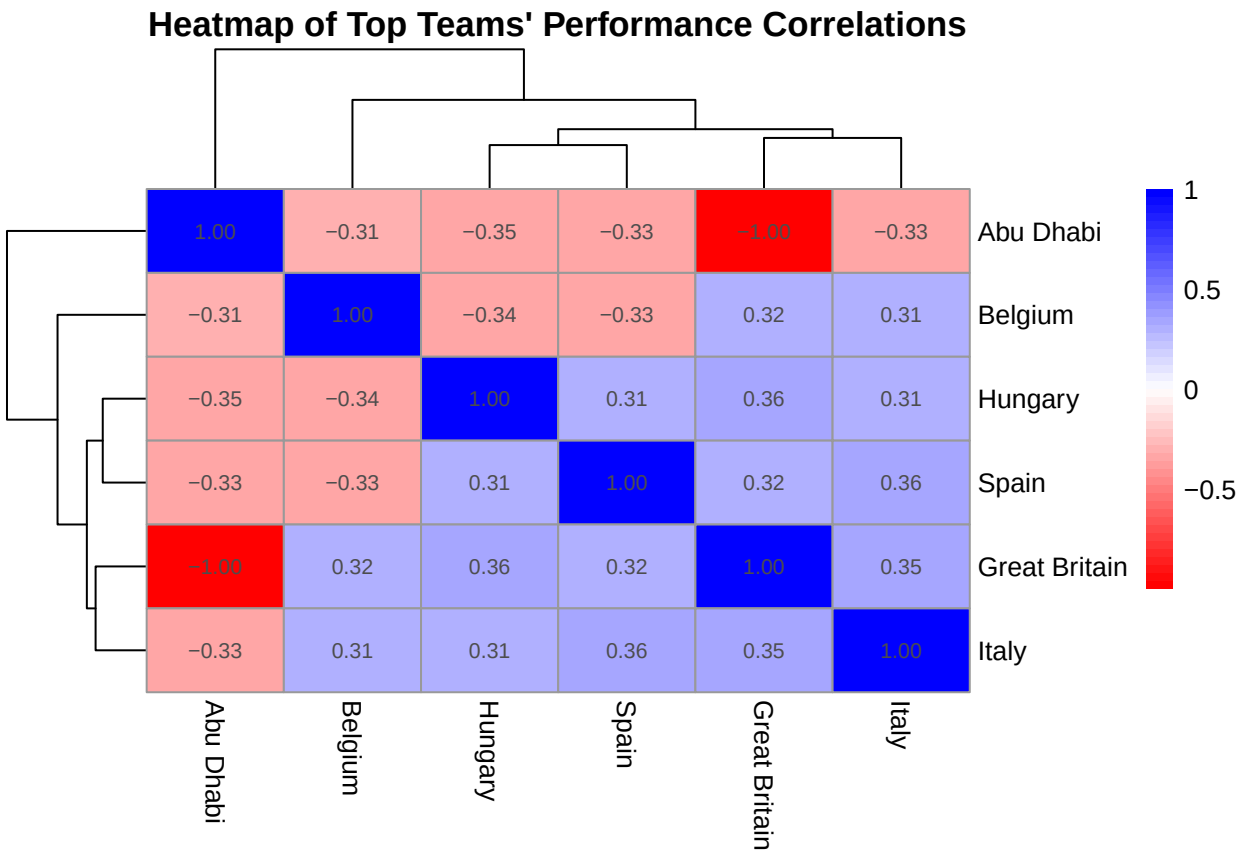
Results And Analysis:

```
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))

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

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

top3_times <- avg_performance %>%
  arrange(grand_prix, avg_time_minutes) %>%
  group_by(grand_prix) %>%
  slice(1:3)

performance_matrix <- top3_times %>%
  pivot_wider(names_from = grand_prix, values_from = avg_time_minutes) %>%
  replace(is.na(.), 0) %>%
  column_to_rownames("team")

performance_scaled <- scale(performance_matrix)

correlation_matrix <- cor(performance_scaled)

pheatmap(correlation_matrix,
         main = "Heatmap of Top Teams' Performance Correlations",
         display_numbers = TRUE,
         color = colorRampPalette(c("red", "white", "blue"))(50),
         cluster_rows = TRUE,
         cluster_cols = TRUE)
```



Overall, the main objective of this project was to serve as a comparative measure into the dissimilarities and similarities between the teams performances over the course of the most used circuits within the 11 year time period (in which the circuits were used 100% of that time). The full list of those circuits were: Abu Dhabi, Belgium, Hungary, Spain, Great Britain, as well as Italy. As mentioned previously, it's important to note that these circuits are all street circuits in which will give each one a fair comparison in this project. The deterministic metrics of this project were based primarily on the time intervals set by each of the top three teams in which it was converted from hours to minutes. The models that were utilized were K-means clustering, hierarchy means clustering, principal component analysis (PCA), and a heat-map.

The K-means clustering model assisted in clustering the performance data from the teams in which were subsequently utilized to form the PCA plot. This analysis allowed us to see hidden patterns of performance between teams from the competitive performances. Hierarchy clustering was utilized to create the dendrogram which allowed an easier way to visualize the hierarchical relationships in which the dissimilarity (height) was shown. The PCA plots *scaled* the performance data as the first two principals were used to show how teams were categorized and clustered together; in which, was then used to create a boxplot based off of the standard deviation of variances discovered in each of the principal components.

Conclusively, this analysis allowed for a deep comparison into the performance metrics in which gives a comprehensive understanding of the Formula 1 racing data from the 2010s. The utilization of the elements/methods were key *visualization* implementations that allowed significant insight on the performance of the top performing teams across the six circuits that were used in each of the years of the sample. This alone provides a sense of stability and balance for the data as the outliers observed didn't have a significant impact on this comparison project.

Discussion/Conclusion Of The Findings

Although, the overall model performed well there are key areas that could be further expanded upon in future works. For instance, in the final section under the results tab the heat-map could be switched out

with a t-distributed stochastic neighbor embedding (t-SNE) graph/chart which could visualize more high dimensional implementations from the vast variables within the dataset. On the other hand, the heat map could be used by the teams to discover how their performance metrics may stack up against another teams depending on a particular track. This is a very crucial area in which elements such as throttle lift, braking points, acceleration patterns can be identified within other teams to better predict how one team's car may compete against another team's car in a statistical, yet, theoretical matter. The EDA and clustering analysis within the project showcases great insights on the simplistic, yet, necessary data table in which all of this is based off of.

https://github.com/GabrielDowns12/Formula-1-Algorithm-Analysis-Project