

# Updated Tuesday F1 Racing

Viwen Suresh Kumar

1/20/2022

## I. Introduction and Overview

This report is about understanding the f1 Tuesday racing results by using the dataset from the package shown below. It contains data on racing points involving with lap speed, its nationality of the constructor, driver standings and other factors.

```
# The following packages will be installed.
if(!require(tidyverse)) install.packages("tidyverse")
if(!require(tidyuesdayR)) install.packages("tidyuesdayR")
if(!require(RColorBrewer)) install.packages("RColorBrewer")

# Load the following packages using the library() function.
library(tidyverse)
library(tidyuesdayR)
library(RColorBrewer)
theme_set(theme_light())
```

## Method & Analysis

For this TT, there are a couple of data sets related to F1 racing. The relationship between constructors and points won per-race. Data visualization is used to understand about the racing results on which F1 car has the fastest lap speed and make comparative results. Using the data that is shown below and being loaded and determine which data sets to use.

```
tt <- tt_load("2021-09-07")
```

```
## --- Compiling #TidyTuesday Information for 2021-09-07 ----
```

```
## --- There are 13 files available ---
```

```
## --- Starting Download ---
```

```
##
```

```
## Downloading file 1 of 13: 'circuits.csv'
```

```
## Downloading file 2 of 13: 'constructor_results.csv'
```

```
## Downloading file 3 of 13: 'constructor_standings.csv'
```

```
## Warning: One or more parsing issues, see 'problems()' for details

## Downloading file 4 of 13: 'constructors.csv'
## Downloading file 5 of 13: 'driver_standings.csv'
## Downloading file 6 of 13: 'drivers.csv'
## Downloading file 7 of 13: 'lap_times.csv'
## Downloading file 8 of 13: 'pit_stops.csv'
## Downloading file 9 of 13: 'qualifying.csv'
## Downloading file 10 of 13: 'races.csv'

## Only 10 Github queries remaining until 2022-01-27 12:18:14 -03.

## Downloading file 11 of 13: 'results.csv'

## Only 9 Github queries remaining until 2022-01-27 12:18:14 -03.

## Warning: One or more parsing issues, see 'problems()' for details

## Downloading file 12 of 13: 'seasons.csv'

## Only 8 Github queries remaining until 2022-01-27 12:18:14 -03.

## Downloading file 13 of 13: 'status.csv'

## Only 7 Github queries remaining until 2022-01-27 12:18:14 -03.

## --- Download complete ---
```

```
# For this project I will use these three data sets
tt$constructor_results
```

```
## # A tibble: 11,860 x 5
##   constructorResultsId raceId constructorId points status
##           <dbl>   <dbl>         <dbl>   <dbl> <chr>
## 1             1     18             1     14 "\\N"
## 2             2     18             2      8 "\\N"
## 3             3     18             3      9 "\\N"
## 4             4     18             4      5 "\\N"
## 5             5     18             5      2 "\\N"
## 6             6     18             6      1 "\\N"
## 7             7     18             7      0 "\\N"
## 8             8     18             8      0 "\\N"
## 9             9     18             9      0 "\\N"
## 10            10     18            10      0 "\\N"
## # ... with 11,850 more rows
```

```
tt$constructors
```

```
## # A tibble: 211 x 5
##   constructorId constructorRef name      nationality url
##   <dbl> <chr> <chr> <chr> <chr>
## 1         1 mclaren      McLaren British http://en.wikipedia.org~
## 2         2 bmw_sauber    BMW Sauber German http://en.wikipedia.org~
## 3         3 williams      Williams British http://en.wikipedia.org~
## 4         4 renault      Renault French http://en.wikipedia.org~
## 5         5 toro_rosso    Toro Rosso Italian http://en.wikipedia.org~
## 6         6 ferrari      Ferrari Italian http://en.wikipedia.org~
## 7         7 toyota      Toyota Japanese http://en.wikipedia.org~
## 8         8 super_aguri    Super Aguri Japanese http://en.wikipedia.org~
## 9         9 red_bull    Red Bull Austrian http://en.wikipedia.org~
## 10        10 force_india Force India Indian http://en.wikipedia.org~
## # ... with 201 more rows
```

```
tt$traces
```

```
## # A tibble: 1,058 x 8
##   raceId year round circuitId name      date      time      url
##   <dbl> <dbl> <dbl> <dbl> <chr> <date> <chr> <chr>
## 1     1   2009     1         1 Australian Grand Prix 2009-03-29 06:00:00 http:~
## 2     2   2009     2         2 Malaysian Grand Prix 2009-04-05 09:00:00 http:~
## 3     3   2009     3        17 Chinese Grand Prix 2009-04-19 07:00:00 http:~
## 4     4   2009     4         3 Bahrain Grand Prix 2009-04-26 12:00:00 http:~
## 5     5   2009     5         4 Spanish Grand Prix 2009-05-10 12:00:00 http:~
## 6     6   2009     6         6 Monaco Grand Prix 2009-05-24 12:00:00 http:~
## 7     7   2009     7         5 Turkish Grand Prix 2009-06-07 12:00:00 http:~
## 8     8   2009     8         9 British Grand Prix 2009-06-21 12:00:00 http:~
## 9     9   2009     9        20 German Grand Prix 2009-07-12 12:00:00 http:~
## 10    10   2009    10        11 Hungarian Grand Prix 2009-07-26 12:00:00 http:~
## # ... with 1,048 more rows
```

```
tt$results
```

```
## # A tibble: 25,220 x 18
##   resultId raceId driverId constructorId number grid position positionText
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <chr>
## 1     1     18     1         1     22     1 "1" 1
## 2     2     18     2         2     3     5 "2" 2
## 3     3     18     3         3     7     7 "3" 3
## 4     4     18     4         4     5    11 "4" 4
## 5     5     18     5         1    23     3 "5" 5
## 6     6     18     6         3     8    13 "6" 6
## 7     7     18     7         5    14    17 "7" 7
## 8     8     18     8         6     1    15 "8" 8
## 9     9     18     9         2     4     2 "\\N" R
## 10    10     18    10         7    12    18 "\\N" R
## # ... with 25,210 more rows, and 10 more variables: positionOrder <dbl>,
## #   points <dbl>, laps <dbl>, time <chr>, milliseconds <chr>, fastestLap <chr>,
## #   rank <chr>, fastestLapTime <chr>, fastestLapSpeed <chr>, statusId <dbl>
```

```

# Filtering the data to show races that happened exclusively between 2016-2019
races_filtered <- tt$races %>%
  filter(year %in% (2016:2019)) %>%
  select(-date, -time, -url)

# Merge the cleaned race data set with the constructor points, both data sets share the raceID column
race_constructor <- merge(x = races_filtered, y = tt$constructor_results)

# Filtering constructors to get only the data I need
clean_constructors <- tt$constructors %>%
  select(constructorId, name, nationality)

# Change name column because they are overlapping when merging
clean_constructors <- clean_constructors %>%
  rename(constructorName = name)

# Selecting fastest lap speed and constructorId
clean_results <- tt$results %>%
  select(constructorId, raceId, fastestLapSpeed) %>%
  mutate_if(is.character, as.numeric)

```

```
## Warning in mask$eval_all_mutate(quo): NAs introduced by coercion
```

## Results

In order to analyze whether there is correlation between constructors and points won per race where the two datasets are being merged and being cleaned up.

```

# Get max speed per race and constructor
constructors_maxspeed <- merge(x = clean_constructors, clean_results) %>%
  filter(!is.na(fastestLapSpeed))

# I get rid of the ID columns since I am no longer intending to merge any other data sets.
clean_data <- merge(x = constructors_maxspeed, y = race_constructor) %>%
  select(-status, -constructorResultsId, -circuitId, -constructorId) %>%
  rename(race = name)

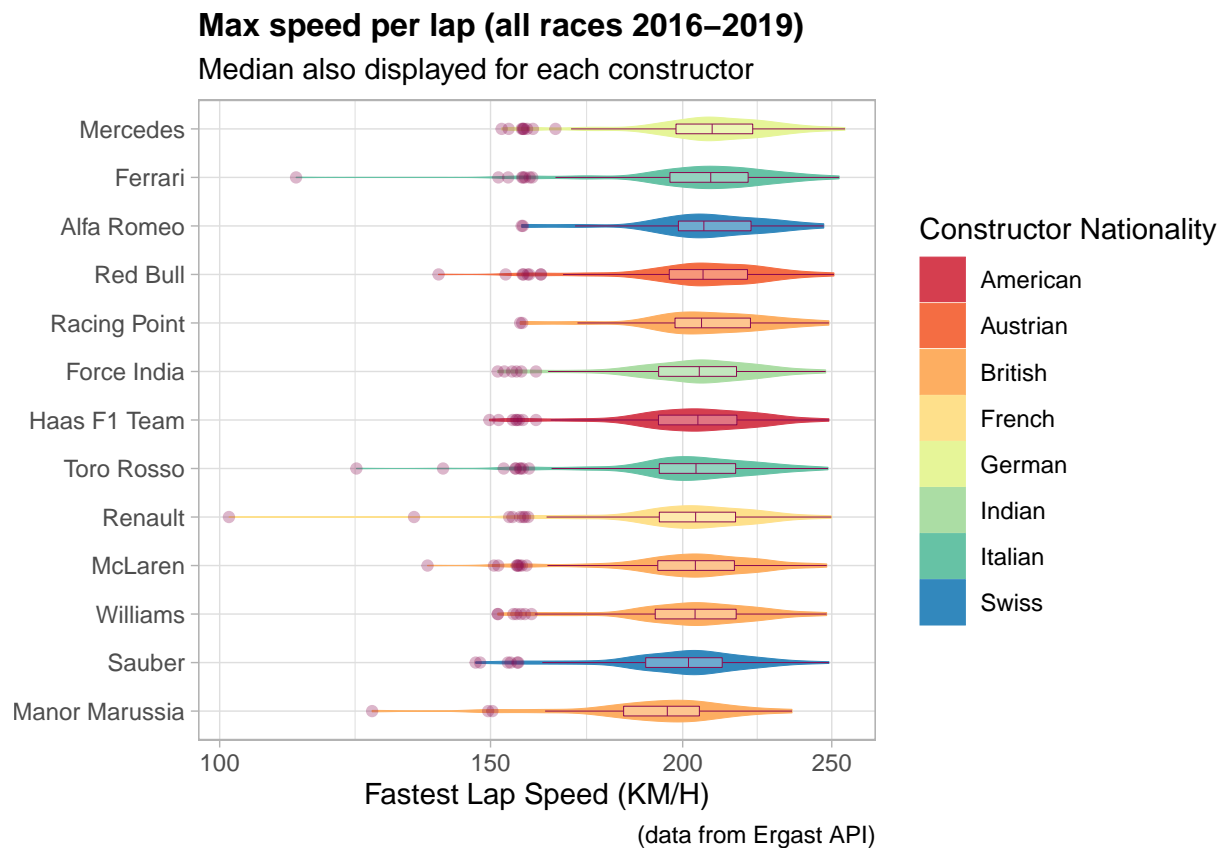
clean_data %>%
  mutate(constructorName = fct_reorder(constructorName, fastestLapSpeed)) %>%
  ggplot(aes(x= fastestLapSpeed, y = constructorName)) +
  geom_violin(aes(fill = nationality, color = nationality), width = 0.5, size = 0.1) +
  scale_fill_brewer(palette = "Spectral") +
  scale_color_brewer(palette = "Spectral") +
  geom_boxplot(width = 0.2, size = 0.05, alpha = 0.3, color = "deepPink4") +
  scale_x_log10() +
  ggtitle("Max speed per lap (all races 2016-2019)",
    subtitle = "Median also displayed for each constructor") +
  theme(
    plot.title = element_text(size = 12, face = "bold")
  ) +
  labs(

```

```

fill = "Constructor Nationality",
color = "Constructor Nationality",
caption = "(data from Ergast API)",
y = NULL,
x = "Fastest Lap Speed (KM/H)"

```



## Conclusions

Based on the diagram shown above, it is shown that Alfa Romeo, Racing Point and Red Bull have been maintaining their high speeds overall consistently which close to the speed of the Ferrari. Additionally, the reason is that they are shown in a lower position than Alfa Romeo & Ferrari is because the data is ranked by top fastest lap speed as opposed to an average speed.

Likewise, Alfa Romeo, Racing Point and Mercedes stand in the top three for lowest minimum max lap speed, which makes these specific constructors even more solid based on my opinion. Thus, this could mean that they require less time to ramp up compared to the the rest.

Lastly, as a novice to F1 racing it is interesting to me that British constructors were a majority between 2016 and 2019.

## Code I found useful

These are some lines of code that could be useful for future ones.

```
# Get the highest number per column
#tt$pit_stops %>%
  summarise_if(is.numeric, max)

# Show the amount of unique values of a specified column
#tt$drivers %>%
  summarise(count = n_distinct(code))
```