Formula 1 Data Analysis

Jon Montgomery

Formula 1 Data Analysis

1.

In this project, we will explore the Formula 1 World Championship data from 1950 to 2024. The data set is available on kaggle. It consists of various CSV files containing tabular data regarding drivers, circuits, teams, and results ranging from the granularity of championship standings to pit stop characteristics. The data are very high quality.

I will be working with Abhishek on this project.

2.

We want to use the historical data to create a Markov Chain model to predict the performance of a driver in the next race. We want to test the effectiveness of the Markov Chain predictions against naive models to see if we can gain an edge. Here are some examples of naive models:
1. A driver's performance is random.
2. A driver's performance in the next race will be the same as their performance in the prior race.
3. A driver's performance in the next race will be equal to the average of their performance in the prior 10 races.

We will split the data into random samplings of training and test data (60% training and 40% test). Once the model is trained, we will evaluate the performance of each model against the test data using the mean absolute error metric, where the error is the difference between the predicted driver rank and the actual driver rank.

Once we have calculated the mean absolute error for each model, we will test the differences in the errors for statistical significance. The method we will use to test for statistical significance will depend on whether or not the errors are normally distributed. If the errors are normally distributed, we will compare the results with the paired t-test. If the errors are not normally distributed, we will use the Wilcoxon signed-rank test.

3.

We will use git for version control in this project. Abhishek and I will both do preliminary data cleaning and exploration and then merge our findings. I will build the markov models and most of the code infrastructure for the project. Then Abhishek will review, we will test the models, and Abhishek will compile slides for the presentation. Once we have the data, we will write the paper, dividing it between us depending on who is most familiar with each part.

For cleaning and exploration, we will evaluate each table and field to decide which variables could hold predictive power and should be included in the Markov model. We will identify missing values and handle them appropriately on a case-by-case basis. We will also identify the breakpoints between seasons with discontinuous car specifications (every 8 years or so the "formula" of the cars changes drastically). We'll divide the data randomly into testing and training data and evaluate the models. We will complete this portion of the project this weekend March 2.

I'll build the Markov Chains between March 3-7 and we will train them and do analysis over the weekend March 8-9. We will spend the rest of our time preparing the paper and the presentation.

```
knitr::opts_chunk$set(echo = TRUE)
library(stringr)
library(ggplot2)
library(knitr)
library(unifed)
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(slider)
```

Markov Chain Parameters

- circuit id
- driver id
- driver's average rank over the past 10 races
- driver's most recent rank
- qualifying position
- constructor id
- constructor rank
- constructor's median pit stop time in the past 10 races
- weather ?

One row per driver per race.

Read the data into dataframes

```
circuits<-
  read_csv(file.path("data", "circuits.csv"),
           col_types  = cols(
           circuitId  = col_integer(),
           circuitRef = col_character(),
           name       = col_character(),
           location   = col_character(),
           country    = col_character(),
           lat        = col_double(),
           lng        = col_double(),
           alt        = col_integer(),
           url        = col_character())) %>%
  select(-c(
    "circuitRef",
    "location",
    "country",
    "lat",
    "lng",
    "alt",
    "url")) %>%
  rename(circuitName=name)

constructor.standings<-
  read_csv(file.path("data", "constructor_standings.csv"),
           col_types = cols(
             constructorStandingsId = col_integer(),
             raceId                 = col_integer(),
             constructorId          = col_integer(),
             points                 = col_double(),
             position               = col_integer(),
             positionText           = col_character(),
             wins                   = col_integer())) %>%
  select(-c(
    "points",
    "positionText",
    "wins")) %>%
  rename(constructorStandingsPosition = position)

constructors<-
  read_csv(file.path("data", "constructors.csv"),
           col_types = cols(
             constructorId  = col_integer(),
             constructorRef = col_character(),
             name           = col_character(),
             nationality    = col_character(),
             url            = col_character())) %>%
  select(-c(
    "nationality",
    "url",
    "constructorRef")) %>%
  rename(constructorName = name)

driver.standings<-
  read_csv(file.path("data", "driver_standings.csv"),
           col_types = cols(
             driverStandingsId = col_integer(),
             raceId            = col_integer(),
             driverId          = col_integer(),
             points            = col_double(),
             position          = col_integer(),
             positionText      = col_character(),
             wins              = col_integer())) %>%
  select(-c(
    "points",
    "positionText",
    "wins")) %>%
  rename(driverStandingsPosition = position)

drivers<-
  read_csv(file.path("data", "drivers.csv"),
           col_types = cols(
             driverId =    col_integer(),
             driverRef =   col_character(),
             number =      col_character(),
             code =        col_character(),
             forename =    col_character(),
             surname =     col_character(),
             dob =         col_date(),
             nationality = col_character(),
             url =         col_character())) %>%
  select(-c(
    "number",
    "code",
    "dob",
    "nationality",
    "url",
    "driverRef"))

races<-
  read_csv(file.path("data", "races.csv"),
           col_types = cols(
             raceId =    col_integer(),
             year =      col_integer(),
             round =     col_integer(),
             circuitId = col_integer(),
             name =      col_character(),
             date =      col_date(),
             time =      col_character(),
             url =       col_character(),
             fp1_date =  col_character(),
             fp1_time =  col_character())) %>%
  select(-c(
    "year",
    "round",
    "name",
    "time",
    "url",
    "fp1_date",
    "fp1_time",
    "fp2_date",
    "fp2_time",
    "fp3_date",
    "fp3_time",
    "quali_date",
    "quali_time",
    "sprint_date",
    "sprint_time"))

results<-
  read_csv(file.path("data", "results.csv"),
           col_types = cols(
             resultId =      col_integer(),
             raceId =        col_integer(),
             driverId =      col_integer(),
             constructorId = col_integer(),
             number =        col_character(),
             grid =          col_integer(),
             position =      col_character(),
             positionText =  col_character(),
             positionOrder = col_integer(),
             points =        col_double())) %>%
  select(-c(
    "number",
    "position",
    "positionText",
    "points",
    "laps",
    "time",
    "milliseconds",
    "fastestLap",
    "rank",
    "fastestLapTime",
    "fastestLapSpeed",
    "statusId")) %>%
  rename(raceResultPosition = positionOrder, startingPosition = grid)
#print(races)
#print(circuits)
#print(results)
#print(drivers)
#print(constructors)
#print(driver.standings)
#print(constructor.standings)
```

Creating the usable data

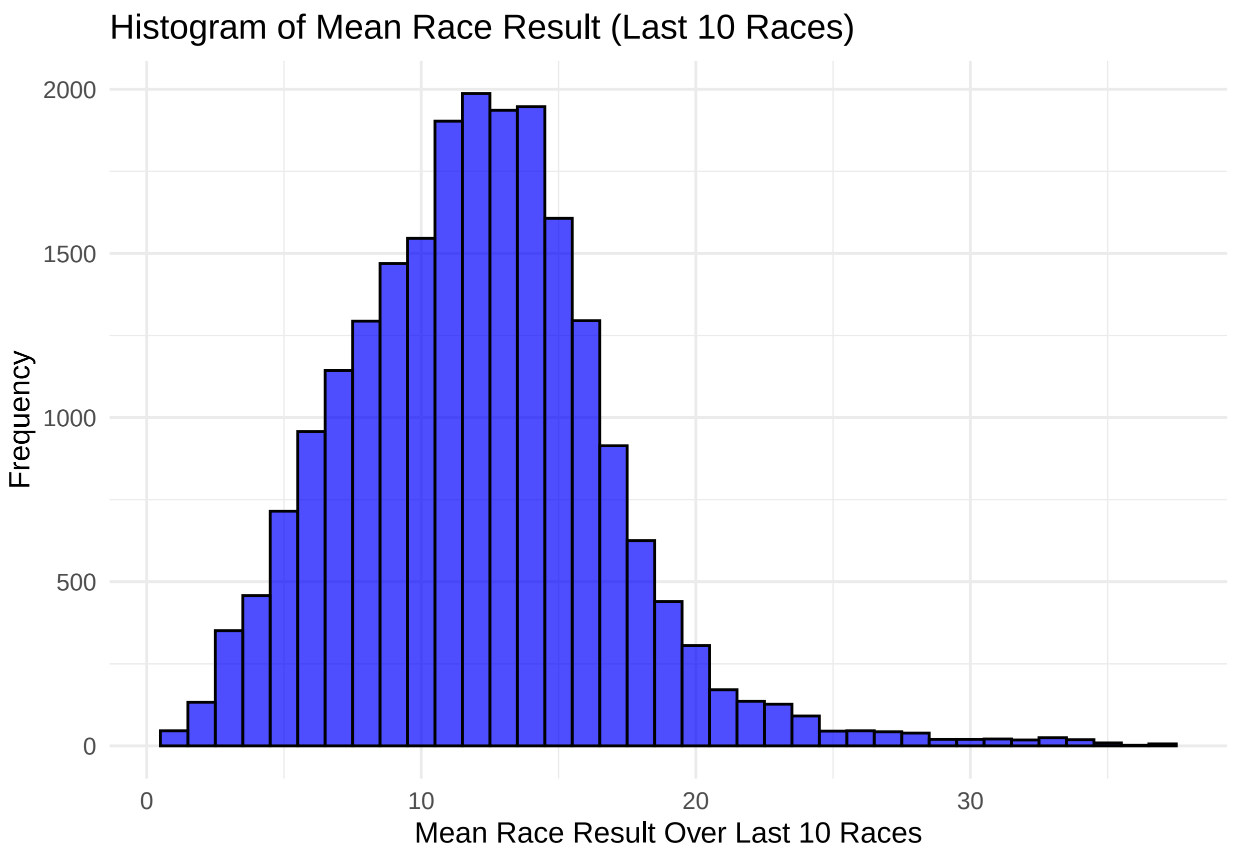
We only include results for drivers that have at least 10 races.

```
# Joining the datasets
data <- results %>%
  left_join(races, by = "raceId") %>%
  left_join(circuits, by = "circuitId") %>%
  left_join(drivers, by = "driverId") %>%
  left_join(constructors, by = "constructorId") %>%
  left_join(driver.standings, by = c("raceId", "driverId")) %>%
  left_join(constructor.standings, by = c("raceId", "constructorId")) %>%
  arrange(driverId, date) %>% # ensure the data is ordered correctly
  group_by(driverId) %>%
  mutate(
    meanRaceResult10 = slider::slide_dbl(
      raceResultPosition,  # Column to calculate rolling mean
      mean,  # Function to apply
      .before = 9,  # Use the prior 10 races (excluding current race)
      .complete = TRUE  # Only compute if there are at least 10 races
    )
  ) %>%
  ungroup() %>%
  filter(!is.na(meanRaceResult10))
```

```r
# Final view of the final data
glimpse(data)
```

```
## Rows: 21,910
## Columns: 17
## $ resultId                      <int> 575, 589, 615, 634, 658, 677, 717, 727, 1~
## $ raceId                        <int> 45, 46, 47, 48, 49, 50, 51, 52, 18, 19, 2~
## $ driverId                      <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ constructorId                 <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ startingPosition              <int> 10, 1, 2, 2, 4, 1, 1, 2, 1, 9, 3, 5, 3, 3~
## $ raceResultPosition            <int> 9, 1, 5, 2, 4, 1, 19, 7, 1, 5, 13, 3, 2, ~
## $ circuitId                     <int> 20, 11, 5, 14, 13, 16, 17, 18, 1, 2, 3, 4~
## $ date                          <date> 2007-07-22, 2007-08-05, 2007-08-26, 2007~
## $ circuitName                   <chr> "Nürburgring", "Hungaroring", "Istanbul P~
## $ forename                      <chr> "Lewis", "Lewis", "Lewis", "Lewis", "Lewi~
## $ surname                       <chr> "Hamilton", "Hamilton", "Hamilton", "Hami~
## $ constructorName               <chr> "McLaren", "McLaren", "McLaren", "McLaren~
## $ driverStandingsId             <int> 13808, 13831, 13854, 13878, 13902, 13926,~
## $ driverStandingsPosition       <int> 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 3, 2, 3, 1,~
## $ constructorStandingsId        <int> 25833, 25844, 25855, 25866, 25877, 25888,~
## $ constructorStandingsPosition <int> 11, 11, 11, 11, 11, 11, 11, 11, 1, 1, 3, ~
## $ meanRaceResult10              <dbl> 2.8, 2.6, 2.9, 2.9, 3.1, 3.0, 4.8, 5.4, 5~
```

```r
ggplot(data, aes(x = meanRaceResult10)) +
  geom_histogram(binwidth = 1, fill = "blue", color = "black", alpha = 0.7) +
  labs(title = "Histogram of Mean Race Result (Last 10 Races)",
       x = "Mean Race Result Over Last 10 Races",
       y = "Frequency") +
  theme_minimal()
```



```
colSums(is.na(data))
```

```
##                    resultId                       raceId
##                           0                            0
##                    driverId                constructorId
##                           0                            0
##            startingPosition           raceResultPosition
##                           0                            0
##                   circuitId                         date
##                           0                            0
##                 circuitName                     forename
##                           0                            0
##                     surname              constructorName
##                           0                            0
##           driverStandingsId      driverStandingsPosition
##                         328                          328
##      constructorStandingsId constructorStandingsPosition
##                         676                          676
##            meanRaceResult10
##                           0
```

What should we do with the null values? What do they mean?