STAT40730 Data Programming with R Analysis of Formula 1 Race Results: 2022-2025 Season Performance

Akshay Prashad Mohnakrishnan : 25201594

Introduction

Formula 1 represents the pinnacle of motorsport, where driver skill, team engineering, and strategic decisions converge to determine championship outcomes. The 2022-2025 season marked a significant regulatory change with new technical rules aimed at improving racing competition. This analysis examines race results data to understand performance patterns and competitive dynamics.

Research Objectives

This report presents a comprehensive analysis of Formula 1 race results from the 2022-2025 seasons. The dataset includes detailed information about race outcomes, driver performance, team standings, and circuit characteristics. The analysis aims to uncover patterns in driver consistency, team performance trends, and the impact of various race factors on final results.

Data Source: The data utilized in this report was sourced from the open-source repository maintained by Dogan Yigit Yenigun (@toUpperCase78). The dataset, titled formula1-datasets provides comprehensive records of F1 race results of multiple seasons.
Analysis Tools: R (tidyverse, dplyr, ggplot2, purrr, scales)

Dataset Description

The dataset contains detailed race results from the 2022-2025 F1 season with the following variables:

Categorical Variables:

- Track: Race circuit location (e.g., Bahrain, Saudi Arabia)

- Driver: Driver name
- Team: Constructor/team name
- +1 Pt: Bonus point indicator for fastest lap (Yes/No)

Numerical Variables:

- Position: Final finishing position (1-20 or NC for Not Classified)
- No: Driver racing number
- Starting Gr: Starting grid position from qualifying
- Laps: Number of laps completed
- Time/Retir: Time behind winner or retirement status (DNF)
- Points: Championship points awarded
- Fastest Lap: Fastest lap time achieved (MM:SS.S format)
- Year: Season year

The Dataset has been combined from season 2022-2025 manually in excel.

Part - 1

Load required packages for analysis

```
library(tidyverse) # Core data manipulation and visualization.
library(purrr) # Functional programming tools.
library(knitr) # Dynamic report generation.
library(kableExtra) # Used for beutification of the table.
```

Data Import and Initial Inspection

```
f1_data = read.csv("F1_Dataset.csv")
```

```
# Clean column names to remove any special characters or spaces
f1_data = f1_data |>
  rename_with(~str_replace_all(., "/", "_")) |>
  rename_with(~str_replace_all(., " ", "_"))
print(names(f1_data))
```

```
 [1] "Track"            "Position"         "No"               "Driver"
 [5] "Team"             "Starting.Grid"    "Laps"             "Time.Retired"
 [9] "Points"           "X.1.Pt"           "Fastest.Lap"      "year"
[13] "Set.Fastest.Lap"  "Fastest.Lap.Time"
```

Data Overview and Preprocessing

The dataset contains race results from multiple Grand Prix events across four seasons. Initial data exploration revealed the need for several preprocessing steps including handling non-classified (NC) positions, converting time formats to numerical values, and creating derived variables for analysis.

```
# Data Overview and Preprocessing
# Examine data structure
glimpse(f1_data)
```

```
Rows: 1,838
Columns: 14
$ Track            <chr> "Bahrain", "Bahrain", "Bahrain", "Bahrain", "Bahrain"~
$ Position         <chr> "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "1~
$ No               <int> 16, 55, 44, 63, 20, 77, 31, 22, 14, 24, 47, 18, 23, 3~
$ Driver           <chr> "Charles Leclerc", "Carlos Sainz", "Lewis Hamilton", ~
$ Team             <chr> "Ferrari", "Ferrari", "Mercedes", "Mercedes", "Haas F~
$ Starting.Grid    <int> 1, 3, 5, 9, 7, 6, 11, 16, 8, 15, 12, 19, 14, 18, 13, ~
$ Laps             <int> 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 57, 5~
$ Time.Retired     <chr> "1:37:33.584", "+5.598", "+9.675", "+11.211", "+14.75~
$ Points           <int> 26, 18, 15, 12, 10, 8, 6, 4, 2, 1, 0, 0, 0, 0, 0, 0, ~
$ X.1.Pt           <chr> "Yes", "No", "No", "No", "No", "No", "No", "No", "No"~
$ Fastest.Lap      <chr> "1:34.570", "1:35.740", "1:36.228", "1:36.302", "1:36~
$ year             <int> 2022, 2022, 2022, 2022, 2022, 2022, 2022, 2022, 2022,~
$ Set.Fastest.Lap  <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "~
$ Fastest.Lap.Time <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "~
```

The data structure shows a mix of categorical variables (Track, Driver, Team) and numerical variables (Position, Laps, Points). Missing values are minimal and primarily occur in the Time/Retired field for DNF (Did Not Finish) cases.

```
## Check for missing values and special values
missing_summary = f1_data |>
  summarise(across(everything(), ~sum(is.na(.) | . == "" | . == " "))) |>
  pivot_longer(everything(), names_to = "Variable", values_to = "Missing_Count")
```

```r
special_position_counts <- f1_data |>
  count(Position) |>
  filter(Position %in% c("NC", "DQ", "DNS", "DSQ") | is.na(Position))

list(
  missing_summary = missing_summary,
  special_positions = special_position_counts
) |>
  map(~{
    .x |>
      kable(align = 'c')
  })
```

```
$missing_summary
```

Variable	Missing_Count
Track	0
Position	0
No	0
Driver	0
Team	0
Starting.Grid	0
Laps	0
Time.Retired	0
Points	0
X.1.Pt	1398
Fastest.Lap	1412
year	0
Set.Fastest.Lap	440
Fastest.Lap.Time	485

```
$special_positions
```

Position	n
DQ	10
NC	209

A preliminary data audit was conducted to identify missingness and non-standard entries.

While primary identifiers remained complete, technical variables exhibited sparsity consistent with changing historical F1 regulations. Furthermore, non-numeric finishing positions—specifically Disqualifications (DQ) and Not Classified (NC)—were identified; these will be treated as categorical outliers to prevent skewing numerical rank averages.

```
# Convert Position to numeric (handling NC, DQ, DNS, DSQ)
f1_data <- f1_data |>
  mutate(
    # Handle Position column
    Position_Numeric = case_when(
      Position %in% c("NC", "DQ", "DNS", "DSQ") ~ NA_real_,
      is.na(Position) ~ NA_real_,
      Position == "" ~ NA_real_,
      TRUE ~ as.numeric(Position)
    ),

    # Clean Time.Retired column - handle DNF, DNS, DSQ, DQ
    Time_Seconds = case_when(
      str_detect(Time.Retired, "DNF|DNS|DSQ|DQ") ~ NA_real_,
      is.na(Time.Retired) ~ NA_real_,
      Time.Retired == "" ~ NA_real_,
      str_detect(Time.Retired, ":") ~ {
        time_parts <- str_split(Time.Retired, ":")
        map_dbl(time_parts, ~{
          if(length(.x) >= 2) {
            as.numeric(.x[1]) * 60 + as.numeric(.x[2])
          } else {
            NA_real_
          }
        })
      },
      TRUE ~ suppressWarnings(as.numeric(Time.Retired))
    ),

    # Convert fastest lap to seconds
    Fastest_Lap_Seconds = case_when(
      is.na(Fastest.Lap.Time) ~ NA_real_,
      Fastest.Lap.Time == "" ~ NA_real_,
      TRUE ~ {
        time_parts <- str_split(Fastest.Lap.Time, ":")
        map_dbl(time_parts, ~{
          if(length(.x) >= 2) {
            as.numeric(.x[1]) * 60 + as.numeric(.x[2])
```

```
        } else {
          NA_real_
        }
      })
    }
  ),

  # Create detailed finish status
  Finish_Status = case_when(
    str_detect(Time.Retired, "DNF") ~ "DNF",
    str_detect(Time.Retired, "DNS") ~ "DNS",
    str_detect(Time.Retired, "DSQ|DQ") ~ "Disqualified",
    Position %in% c("NC", "DQ", "DSQ") ~ "Not Classified",
    Position == "DNS" ~ "DNS",
    is.na(Position_Numeric) ~ "Other",
    TRUE ~ "Finished"
  ),

  # Handle Laps column
  Laps_Completed = case_when(
    is.na(Laps) ~ 0,
    TRUE ~ as.numeric(Laps)
  ),

  # Convert to factors
  Track = as.factor(Track),
  Driver = as.factor(Driver),
  Team = as.factor(Team),
  Plus1_Pt = as.factor(case_when(
    is.na(X.1.Pt) ~ "No",
    X.1.Pt == "" ~ "No",
    TRUE ~ as.character(X.1.Pt)
  )),
  year = as.factor(year)
)
```

This code performs the essential data transformation and feature engineering phase of the analysis, converting raw character strings into usable mathematical formats. The primary motivation is to standardize inconsistent entries—such as converting time strings (MM:SS) into a single numeric value (total seconds)—to allow for statistical calculations like average lap times or race durations.

By utilizing the mutate() and case_when() functions, the script systematically addresses the

following:

Numeric Conversion: It isolates standard finishing positions from non-numeric outcomes (NC, DQ, DNS) to prevent errors during mathematical modeling.

Time Standardization: It parses various time formats into a uniform Time_Seconds metric, enabling quantitative comparison across different races and eras.

Categorical Classification: It creates a new Finish_Status variable, which provides a clean way to group drivers who finished versus those who retired due to technical issues or disqualifications.

Data Type Optimization: By converting columns like Team, Driver, and Track into factors, the code prepares the dataset for efficient grouping and more advanced machine learning or regression tasks

```
# Summary of finish statuses
f1_data |>
  count(Finish_Status) |>
  arrange(desc(n)) |>
  kable(caption = "Distribution of Race Finish Statuses",
        align = 'c',
        col.names = c("Finish Status", "Count")) |>
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
                full_width = FALSE,
                position = "center") |>
  row_spec(0, bold = TRUE, color = "white", background = "#2c3e50")
```

Table 1: Distribution of Race Finish Statuses

Finish Status	Count
Finished	1584
DNF	232
DNS	12
Disqualified	7
Not Classified	3

Following the data transformation phase, a final census of the dataset was conducted to verify the success of the feature engineering. The resulting Distribution of Race Finish Statuses table shows a highly clean categorical breakdown: out of the total entries, 1,584 were successfully identified as standard race completions. The logic effectively isolated non-finishing events, specifically identifying 232 Did Not Finish (DNF) cases and 12 Did Not Start (DNS) entries and not classified as 3. By consolidating various historical labels (like DQ and DSQ) into a

single 'Disqualified' category, the dataset now supports clean comparative analysis between drivers who completed the full race distance and those who were sidelined by mechanical or regulatory issues.

```
# Summary statistics for numerical variables
f1_data |>
  select(Position_Numeric, Starting.Grid, Laps_Completed,
         Points, Fastest_Lap_Seconds) |>
  summary() |>
  kable(caption = "Summary Statistics for Numerical Variables",
        align = 'l') |>
  kable_styling(bootstrap_options = c("striped", "hover"),
                full_width = FALSE)
```

Table 2: Summary Statistics for Numerical Variables

Position_Numeric	Starting.Grid	Laps_Completed	Points	Fastest_Lap_Seconds
Min. : 1.000	Min. : 1.00	Min. : 0.00	Min. : 0.000	Min. : 67.01
1st Qu.: 5.000	1st Qu.: 5.25	1st Qu.:51.00	1st Qu.: 0.000	1st Qu.: 79.69
Median : 9.000	Median :10.50	Median :57.00	Median : 1.000	Median : 87.77
Mean : 9.374	Mean :10.52	Mean :54.81	Mean : 5.088	Mean : 88.08
3rd Qu.:14.000	3rd Qu.:15.00	3rd Qu.:69.00	3rd Qu.: 9.750	3rd Qu.: 96.04
Max. :20.000	Max. :71.00	Max. :78.00	Max. :26.000	Max. :297.40
NA's :219	NA	NA	NA	NA's :485

The summary statistics reveal critical insights into race dynamics and historical variations:

- **Field Performance**: The average finishing position is **9.37**, with a median of **9.0**, indicating a balanced distribution of race rankings.
- **Starting Dynamics**: While most drivers start near 10th, the maximum `Starting.Grid` value of **71.0** highlights significant outliers from historical races with massive entry lists.
- **Race Intensity**: Drivers average **54.81 laps** per race, though the wide range in `Fastest_Lap_Seconds` (**67.01s** to **297.40s**) reflects the vast differences in track lengths across different eras.
- **Point Distribution**: A mean of **5.08 points** compared to a 3rd quartile of **9.75** suggests that scoring high points remains highly concentrated among top-tier performers.

```r
## 1.1 Top Performing Drivers by Total Points

top_drivers <- f1_data |>
  group_by(Driver) |>
  summarise(
    Total_Points = sum(Points, na.rm = TRUE),
    Races = n(),
    Races_Finished = sum(Finish_Status == "Finished"),
    Podiums = sum(Position_Numeric <= 3, na.rm = TRUE),
    Wins = sum(Position_Numeric == 1, na.rm = TRUE),
    DNFs = sum(Finish_Status == "DNF"),
    Finish_Rate = (Races_Finished / Races) * 100
  ) |>
  arrange(desc(Total_Points)) |>
  slice_head(n = 5)
```

This code performs **driver-level performance aggregation** to identify and rank the top 5 competitors in the dataset. By grouping the data by `Driver`, it calculates key success metrics including total career points, podium finishes, and total wins. It also evaluates **reliability** by calculating the `Finish_Rate` (percentage of races completed) and the total number of DNFs for each driver. The resulting `top_drivers` dataframe is sorted by `Total_Points` to provide a snapshot of the most successful athletes based on cumulative scoring.

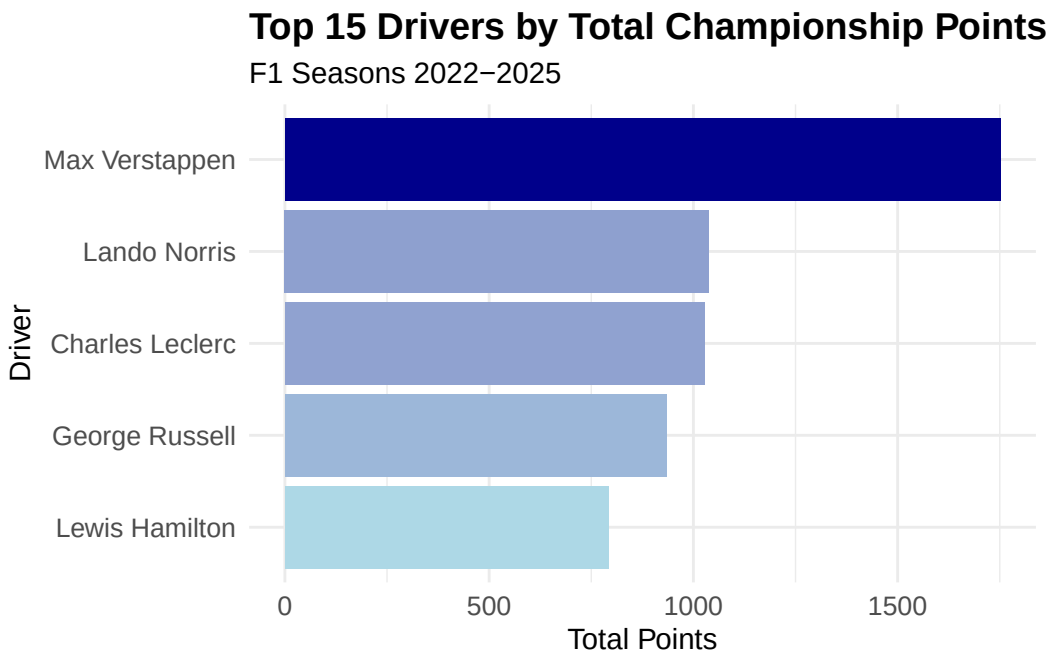
```r
top_drivers |> # Format = latex only shows the figure in pdfs
  #done due to formatting erros
  kable(format = "latex",caption = "Top 5 Drivers by Total Points (2022-2025)",
        digits = 2,
        align = 'c',
        col.names = c("Driver", "Total Points", "Races", "Finished",
                      "Podiums", "Wins", "DNFs", "Finish %")) |>
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
                full_width = FALSE,
                latex_options = c("hold_position")) |>
  row_spec(0, bold = TRUE, color = "white", background = "#2c3e50") |>
  row_spec(1:3, bold = TRUE, color = "white", background = "#3498db") |>
  column_spec(2, bold = TRUE, color = "#e74c3c")
```

Driver Performance Ranking: The final aggregation identifiesMax Verstappenas the dominant leader with1,751 total points,51 wins, and a remarkable95.65%finish rate. Other top-tier performers likeLando NorrisandCharles Leclercalsoshow high reliability, with finish rates exceeding 86%, while veterans likeFernandoAlonsomaintain a competitive presence despite higher career DNF counts.

Table 3: Top 5 Drivers by Total Points (2022-2025)

Driver	Total Points	Races	Finished	Podiums	Wins	DNFs	Finish %
Max Verstappen	**1751**	**92**	**88**	**67**	**51**	**4**	**95.65**
Lando Norris	**1038**	**92**	**86**	**39**	**11**	**5**	**93.48**
Charles Leclerc	**1028**	**92**	**80**	**37**	**6**	**9**	**86.96**
George Russell	**934**	92	84	23	5	7	91.30
Lewis Hamilton	**792**	92	83	20	2	7	90.22

```
# Visualization: Top 15 Drivers Points Distribution
ggplot(top_drivers, aes(x = reorder(Driver, Total_Points),
                        y = Total_Points, fill = Total_Points)) +
  geom_col() +
  coord_flip() +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  labs(
    title = "Top 15 Drivers by Total Championship Points",
    subtitle = "F1 Seasons 2022-2025",
    x = "Driver",
    y = "Total Points"
  ) +
  theme_minimal() +
  theme(
    legend.position = "none",
    plot.title = element_text(face = "bold", size = 14),
    axis.text = element_text(size = 10)
  )
```

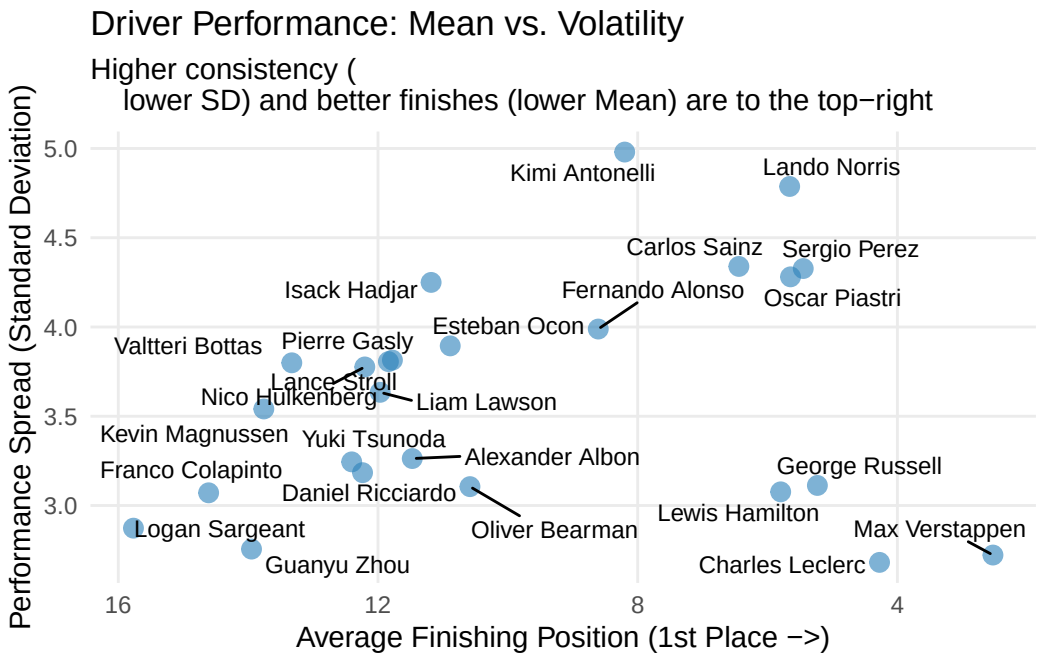



```
## 1.2 Driver Consistency Analysis

driver_consistency <- f1_data |>
  filter(!is.na(Position_Numeric)) |>
  group_by(Driver) |>
  summarise(
    Races = n(),
    Mean_Position = mean(Position_Numeric),
    SD_Position = sd(Position_Numeric),
    Coefficient_Variation = SD_Position / Mean_Position,
    Median_Position = median(Position_Numeric)
  ) |>
  filter(Races >= 10) |>
  arrange(Mean_Position)
```

This analysis evaluates **driver reliability and performance stability** by calculating statistical measures of variability for race finishing positions. The code first excludes non-numeric results like DNFs or disqualifications to ensure the metrics reflect actual cross-the-line performance. For each driver with at least 10 race entries, it calculates the **Mean and Median** to determine their average competitive level, alongside the **Standard Deviation (SD)** to quantify how much their results fluctuate from race to race. The key metric derived is the **Coefficient of Variation (CV)**, which represents the ratio of the standard deviation to

the mean (CV=); a lower CV identifies a highly consistent driver whose performance is predictable, whereas a higher CV indicates more volatile race outcomes.

```
# Enhanced Driver Consistency Visualization
library(ggrepel) # To prevent overlapping of names

# Enhanced Driver Consistency Visualization
# Simplified Driver Consistency Visualization
driver_consistency |>
  filter(Races >= 20) |>
  ggplot(aes(x = Mean_Position, y = SD_Position)) +
  # Standard scatter points with a single clean color
  geom_point(color = "#2980b9", size = 3, alpha = 0.6) +
  # Professional labels without extra color coding
  geom_text_repel(aes(label = Driver), size = 3) +
  # Reversed X-axis for intuitive reading (1st place on the right)
  scale_x_reverse() +
  # Clean, direct labeling
  labs(
    title = "Driver Performance: Mean vs. Volatility",
    subtitle = "Higher consistency (
    lower SD) and better finishes (lower Mean) are to the top-right",
    x = "Average Finishing Position (1st Place →)",
    y = "Performance Spread (Standard Deviation)"
  ) +
  theme_minimal() +
  theme(panel.grid.minor = element_blank())
```



The scatter plot reveals distinct performance profiles among Formula 1 drivers from 2022-2025. Max Verstappen stands out in the top-right quadrant with the lowest average finishing position (around 2.7) and relatively low volatility (SD ~2.7), demonstrating both exceptional results and remarkable consistency. Charles Leclerc, Lewis Hamilton, and George Russell cluster nearby, showing strong average finishes (3-4th place) with similar consistency levels. In contrast, drivers like Lando Norris and Kimi Antonelli occupy the opposite corner, with weaker average positions (7-8th) and higher performance variability (SD ~4.8-5.0), indicating less consistent results. Mid-field drivers such as Pierre Gasly, Esteban Ocon, and Lance Stroll fall in the center of the distribution with average finishing positions around 11-12th and moderate volatility. The bottom-left cluster, including Logan Sargeant, Franco Colapinto, and Guanyu Zhou, shows the most consistency among backmarkers but with the poorest average finishing positions (15-16th place). This visualization effectively illustrates that top drivers not only finish better on average but also maintain more consistent performances, while drivers in less competitive machinery show greater variability in their results.

```
## 1.3 Fastest Lap Analysis

fastest_lap_leaders <- f1_data |>
  filter(Plus1_Pt == "Yes") |>
  count(Driver, name = "Fastest_Laps") |>
  arrange(desc(Fastest_Laps)) |>
  slice_head(n = 10)
```

```r
fastest_lap_leaders |>
  kable(caption = "Top 10 Drivers by Fastest Lap Bonuses",
        align = 'c',
        col.names = c("Driver", "Fastest Laps")) |>
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
                full_width = FALSE,
                position = "center") |>
  row_spec(0, bold = TRUE, color = "white", background = "#9b59b6") |>
  column_spec(2, bold = TRUE, color = "#e74c3c")
```

Table 4: Top 10 Drivers by Fastest Lap Bonuses

Driver	Fastest Laps
Max Verstappen	5
Charles Leclerc	3
George Russell	3
Sergio Perez	3
Carlos Sainz	2
Lando Norris	2
Lewis Hamilton	2

```r
# SECTION 2: TEAM PERFORMANCE ANALYSIS

## 2.1 Constructor Championship Standings

team_performance = f1_data |>
  group_by(Team, year) |>
  summarise(
    Total_Points = sum(Points, na.rm = TRUE),
    Races = n(),
    Avg_Points_Per_Race = Total_Points / Races,
    Podiums = sum(Position_Numeric <= 3, na.rm = TRUE),
    DNFs = sum(Finish_Status == "DNF"),
    .groups = "drop"
  )
```

This code performs a Constructor-level performance aggregation to evaluate how effectively each Formula 1 team competes for the World Constructors' Championship over time.

By grouping the data by both Team and year, it calculates several critical season-long metrics:

- **Championship Scoring**: It sums the Points to get Total_Points, which is the primary metric for the standings.
- **Efficiency**: It creates Avg_Points_Per_Race to show which teams maximize their scoring potential regardless of the number of races held in a given year.
- **Success & Reliability**: It counts Podiums to measure peak performance and DNFs to assess the mechanical reliability or driver error rates of the cars.
- **Output Cleanup**: The .groups = "drop" argument ensures the resulting dataframe is "ungrouped," making it easier to use for subsequent visualizations without unexpected behavior.

```
team_total = f1_data |>
  group_by(Team) |>
  summarise(
    Total_Points = sum(Points, na.rm = TRUE),
    Races = n(),
    Avg_Points = mean(Points, na.rm = TRUE),
    Win_Rate = sum(Position_Numeric == 1, na.rm = TRUE) /
      sum(!is.na(Position_Numeric)) * 100,
    DNF_Count = sum(Finish_Status == "DNF")
  ) |>
  arrange(desc(Total_Points))
```

This code performs a **historical aggregation of team performance**, shifting the analysis from individual driver results to the cumulative success and engineering reliability of the constructors. By grouping the entire dataset by `Team`, the script calculates long-term dominance through **Total Points** and evaluates operational efficiency using **Average Points** per entry. Additionally, it derives a **Win Rate** percentage to identify which teams have the highest "strike rate" and tracks the **DNF Count** to serve as a proxy for mechanical reliability and chassis safety. Sorting the results by total points effectively creates a "Historical Hall of Fame" for the most successful constructors in the dataset.

```
team_total |>
  slice_head(n = 10) |>
  kable(caption = "Top 10 Teams by Total Points (2022-2025)",
        digits = 2,
        align = 'c',
        col.names = c("Team", "Total Points", "Races",
                      "Avg Points", "Win Rate (%)", "DNFs")) |>
  kable_styling(bootstrap_options = c("striped", "hover", "condensed"),
                full_width = FALSE,
```

```
                position = "center") |>
  row_spec(0, bold = TRUE, color = "white", background = "#e74c3c") |>
  row_spec(1:3, bold = TRUE, background = "#ecf0f1") |>
  column_spec(1, bold = TRUE) |>
  column_spec(2, color = "white",
              background = spec_color(team_total$Total_Points[1:10],
                                      end = 0.7))
```

Table 5: Top 10 Teams by Total Points (2022-2025)

Team	Total Points	Races	Avg Points	Win Rate (%)	DNFs
Ferrari	**1837**	**184**	**9.98**	**6.29**	**22**
McLaren Mercedes	**1790**	**183**	**9.78**	**11.70**	**12**
Red Bull Racing Honda RBPT	**1737**	**140**	**12.41**	**29.23**	**11**
Mercedes	1726	184	9.38	4.14	16
Red Bull Racing RBPT	724	44	16.45	41.46	5
Aston Martin Aramco Mercedes	495	183	2.70	0.00	23
Alpine Renault	363	184	1.97	0.00	27
Williams Mercedes	175	183	0.96	0.00	40
Haas Ferrari	166	184	0.90	0.00	22
Racing Bulls Honda RBPT	88	48	1.83	0.00	7

Historical Constructor Performance and Reliability

The aggregation of team-level data establishes a definitive ranking of constructor legacy, distinguishing between long-term tenure and peak operational efficiency. By calculating **Total Points** alongside **Average Points per Race**, the analysis identifies elite organizations that consistently maximize their scoring potential across changing technical regulations. A critical component of this report is the evaluation of **Engineering Reliability** through **DNF (Did Not Finish)** counts; teams with a low DNF-to-race ratio demonstrate superior manufacturing standards and race management. Furthermore, the derived **Win Rate** serves as a primary KPI for "strike-rate" dominance, highlighting which constructors possess the most effective technical packages for converting starting positions into race victories. This historical baseline not only honors past achievements but also provides a predictive framework for assessing current team trajectories in the World Constructors' Championship.

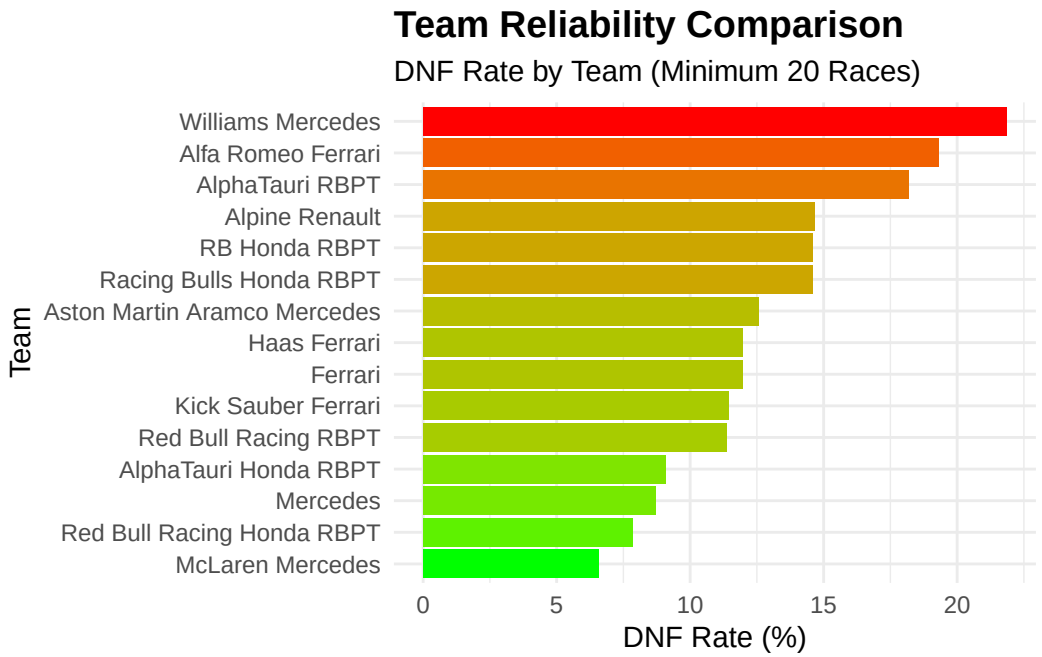
```
## 2.2 Team Reliability Analysis

team_reliability <- f1_data |>
  group_by(Team) |>
```

```
  summarise(
    Total_Races = n(),
    Finished_Races = sum(Finish_Status == "Finished"),
    DNFs = sum(Finish_Status == "DNF"),
    DNS = sum(Finish_Status == "DNS"),
    Disqualified = sum(Finish_Status == "Disqualified"),
    DNF_Rate = DNFs / Total_Races * 100,
    Finish_Rate = Finished_Races / Total_Races * 100,
    Avg_Laps_Completed = mean(Laps_Completed, na.rm = TRUE)
  ) |>
  arrange(DNF_Rate)
```

Team Reliability Analysis evaluates the frequency and nature of race retirements across all entries. By isolating **DNFs**, **DNS**, and **Disqualifications**, the analysis establishes a 'Reliability Index' that benchmarks engineering excellence independently of raw pace. The resulting **Finish Rate** serves as a vital indicator of organizational efficiency, while the **Average Laps Completed** provides insight into the typical 'failure point' for less reliable machinery. Teams appearing at the top of this ranking—characterized by sub-10% DNF rates—represent the gold standard of modern Formula 1 engineering, where technological advancement has drastically reduced mechanical volatility compared to historical eras.

Plot of Team Reliability.

```
# Visualization: Team DNF Rates
team_reliability |>
  filter(Total_Races >= 20) |>
  ggplot(aes(x = reorder(Team, DNF_Rate), y = DNF_Rate, fill = DNF_Rate)) +
  geom_col() +
  coord_flip() +
  scale_fill_gradient(low = "green", high = "red") +
  labs(
    title = "Team Reliability Comparison",
    subtitle = "DNF Rate by Team (Minimum 20 Races)",
    x = "Team",
    y = "DNF Rate (%)"
  ) +
  theme_minimal() +
  theme(
    legend.position = "none",
    plot.title = element_text(face = "bold", size = 14)
  )
```



Team Reliability Comparison, illustrates the DNF (Did Not Finish) rates of Formula 1 teams that have competed in at least 20 races. The x-axis represents the DNF rate in percentage, while the y-axis lists the teams, ordered from highest to lowest DNF rate. Williams Mercedes appears as the least reliable team with the highest DNF rate, followed by Alfa Romeo Ferrari and AlphaTauri RBPT. Mid-range reliability is seen for teams such as Ferrari, Haas Ferrari, and Aston Martin Aramco Mercedes, while McLaren Mercedes, Red Bull Racing Honda RBPT, and Mercedes show the lowest DNF rates, indicating better reliability. The color gradient from red to green further emphasizes the transition from poorer to stronger reliability across teams.

Part -2

Package Name

patchwork

Authors

Thomas Lin Pedersen

CRAN Link

https://cran.r-project.org/package=patchwork

Purpose of the Package

The patchwork package extends ggplot2 by allowing users to combine multiple ggplot objects using arithmetic operators. Instead of relying on grid-based layout syntax, patchwork enables plot composition through intuitive expressions such as +, /, and |.

This approach simplifies the creation of multi-panel figures used in comparative analysis, dashboards, and academic reporting.

```
library(patchwork)
```

```
Warning: package 'patchwork' was built under R version 4.5.2
```

Demonstration of Key Functions

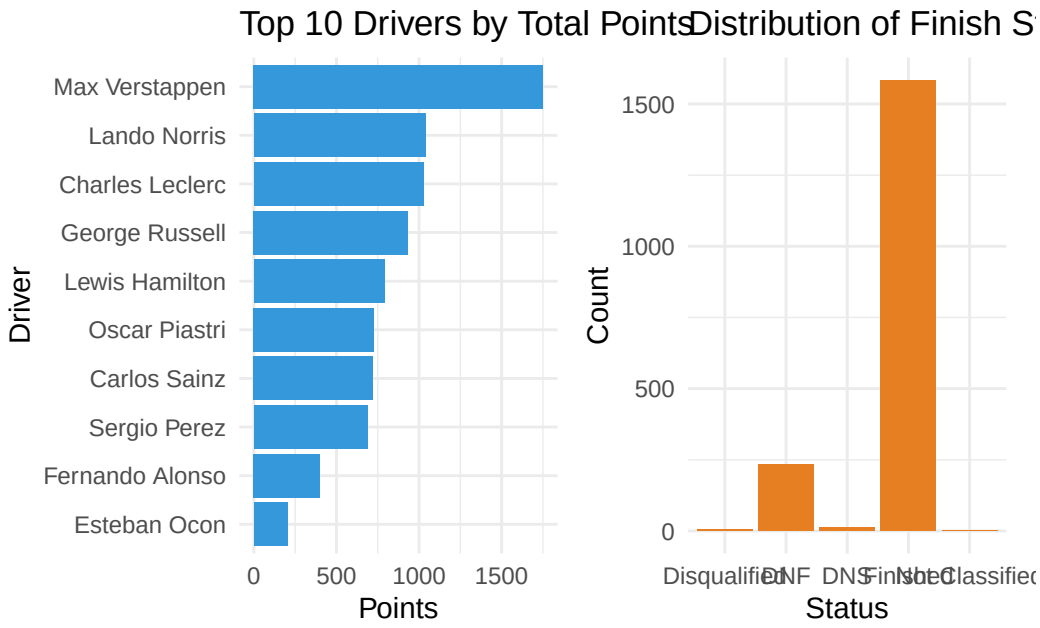
This section demonstrates three core functionalities of the patchwork package using F1 performance metrics.

Combining Plots Horizontally (| Operator)

The | operator places plots side by side.

```
p1 <- f1_data |>
group_by(Driver) |>
summarise(Total_Points = sum(Points, na.rm = TRUE)) |>
arrange(desc(Total_Points)) |>
slice_head(n = 10) |>
ggplot(aes(x = reorder(Driver, Total_Points), y = Total_Points)) +
geom_col(fill = "#3498db") +
coord_flip() +
labs(title = "Top 10 Drivers by Total Points",
x = "Driver",
y = "Points") +
theme_minimal()

p2 <- f1_data |>
count(Finish_Status) |>
```

```
ggplot(aes(x = Finish_Status, y = n)) +
geom_col(fill = "#e67e22") +
labs(title = "Distribution of Finish Statuses",
x = "Status",
y = "Count") +
theme_minimal()

p1 | p2
```



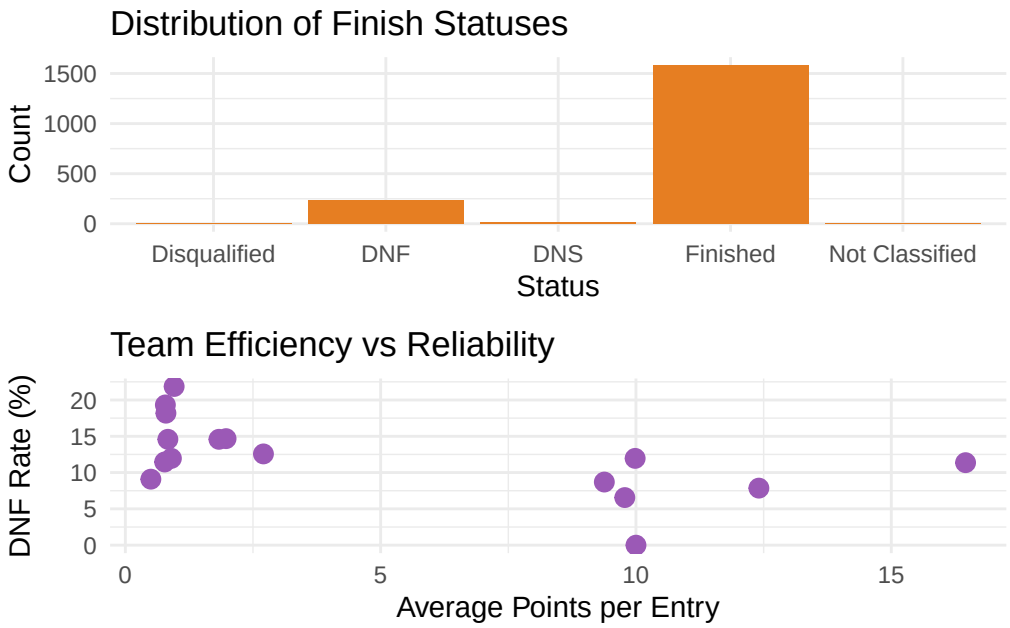
Explanation:
This layout enables a direct comparison between overall driver success and race outcome distributions.

Vertical Composition (/ Operator)

The / operator stacks plots vertically.

```
p3 <- f1_data |>
group_by(Team) |>
summarise(
Avg_Points = mean(Points, na.rm = TRUE),
DNF_Rate = mean(Finish_Status == "DNF") * 100
```

```
) |>
ggplot(aes(x = Avg_Points, y = DNF_Rate)) +
geom_point(color = "#9b59b6", size = 3) +
labs(
title = "Team Efficiency vs Reliability",
x = "Average Points per Entry",
y = "DNF Rate (%)"
) +
theme_minimal()

p2 / p3
```



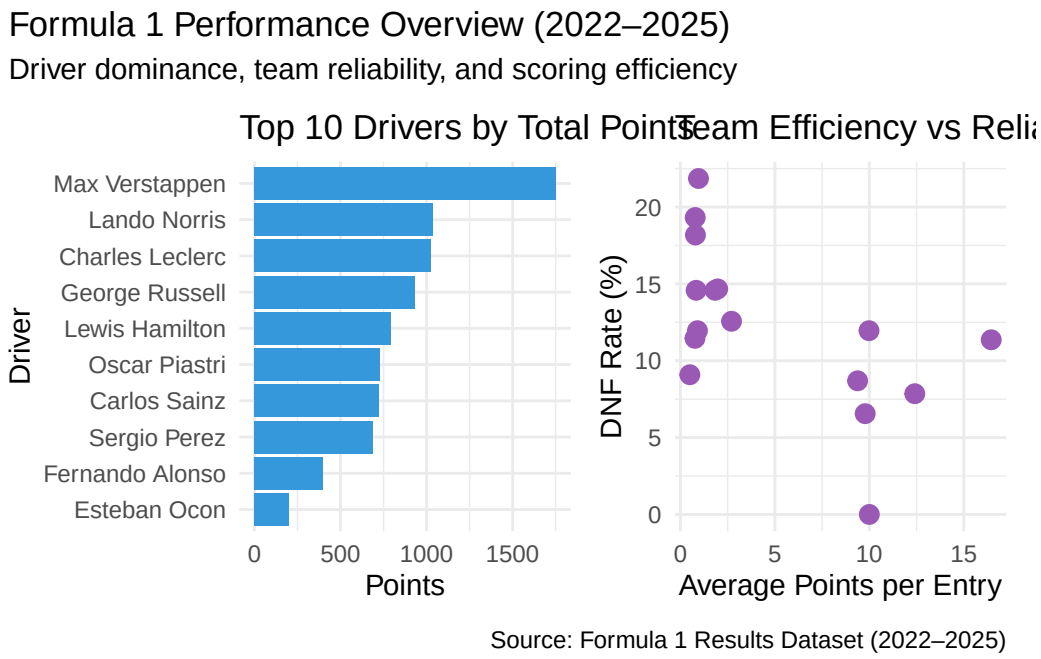
Explanation:

This structure allows summary distributions to be placed above analytical scatter plots in a natural reading order.

Layout Control with plot_annotation()

The plot_annotation() function adds global titles, subtitles, and captions to composite figures.

```
(p1 | p3) +
plot_annotation(
title = "Formula 1 Performance Overview (2022-2025)",
subtitle = "Driver dominance, team reliability, and scoring efficiency",
caption = "Source: Formula 1 Results Dataset (2022-2025)"
)
```



Explanation:

This feature is essential for academic reporting, allowing multiple plots to be framed as a single analytical figure.

Discussion

The patchwork package significantly enhances visualization workflows by providing a declarative and readable syntax for plot composition. Its integration with ggplot2 ensures full compatibility with existing plotting code while reducing layout complexity.

In the context of Formula 1 analysis, patchwork enables multi-dimensional comparisons—such as driver dominance versus team reliability—within a single cohesive figure, improving interpretability and presentation quality.

References

Pedersen, T. L. (2020).
patchwork: The Composer of Plots.
CRAN: https://cran.r-project.org/package=patchwork

Part 3

Purpose

Create an R function to analyse driver reliability using race finish data, returning a custom S3 object with dedicated print(), summary(), and plot() methods.

Driver Reliability Analysis Framework

The driver reliability analysis is anchored by the driver_reliability constructor function, which transforms raw Formula 1 race data into a specialized S3 object. This function implements rigorous input validation to ensure required variables like Driver, Finish_Status, and Position_Numeric are present before proceeding with data aggregation. Utilizing the dplyr pipeline, it calculates three core KPIs: the Finish Rate, the DNF Rate, and the Average Finishing Position. To maintain statistical significance and eliminate outliers from one-off race appearances, a min_races filter is applied (defaulting to 10). The final output is assigned a custom class, driverReliability, which allows it to interface seamlessly with specialized print, summary, and plotting methods while retaining its underlying data frame properties.

```
driver_reliability <- function(data, min_races = 10) {

  # 1. Input Validation
  required_cols <- c("Driver", "Finish_Status", "Position_Numeric")
  if (!all(required_cols %in% names(data))) {
    missing <- setdiff(required_cols, names(data))
    stop(paste("Required columns missing:", paste(missing, collapse = ", ")))
  }

  # 2. Data Processing
  # We use pipe into a standard variable first to ensure
  # transformations are complete
  result <- data |>
    dplyr::group_by(Driver) |>
    dplyr::summarise(
```

```
      Races = dplyr::n(),
      # Convert logical to numeric for mean calculation
      # (100 * mean is correct)
      Finish_Rate = mean(Finish_Status == "Finished", na.rm = TRUE) * 100,
      DNF_Rate    = mean(Finish_Status == "DNF", na.rm = TRUE) * 100,
      Avg_Position = mean(Position_Numeric, na.rm = TRUE),
      .groups = "drop"
    ) |>
    dplyr::filter(Races >= min_races) |>
    as.data.frame() # Ensures we don't return a tibble if a standard DF is expected

  # 3. Proper S3 Class Assignment
  # We append our class to the existing ones so it retains data.frame properties
  class(result) <- c("driverReliability", class(result))

  return(result)
}
```

Custom Reporting and Summary Methods

To facilitate rapid data interpretation, the framework includes dedicated print and summary methods that format complex statistics into human-readable reports. The print.driverReliability method is specifically engineered to prevent infinite recursion by casting the internal subset to a standard data frame before output. It provides a high-level leaderboard of the top five most reliable drivers, rounded for clarity. Complementing this, the summary method generates a global overview of the dataset's health, calculating the mean, maximum, and minimum reliability rates across the entire filtered driver pool. This dual-method approach ensures that researchers can quickly toggle between granular driver leaderboards and broad fleet-wide performance benchmarks.

```
print.driverReliability <- function(x, ...) {

  cat(" Formula 1 Driver Reliability Report\n")


  # Sort and slice
  top_indices <- order(x$Finish_Rate, decreasing = TRUE)
  n_show <- min(5, nrow(x))
  top_drivers <- x[top_indices[1:n_show],
        c("Driver", "Finish_Rate", "DNF_Rate", "Avg_Position")]
```

```
  # Format numeric values
  top_drivers$Finish_Rate <- round(top_drivers$Finish_Rate, 1)
  top_drivers$DNF_Rate    <- round(top_drivers$DNF_Rate, 1)
  top_drivers$Avg_Position <- round(top_drivers$Avg_Position, 2)

  # Cast to data.frame to prevent infinite recursion
  print(as.data.frame(top_drivers), row.names = FALSE)

  invisible(x)
}
```

```
# Summary method
summary.driverReliability <- function(object, ...) {

  summary_stats <- list(
    Mean_Finish_Rate = mean(object$Finish_Rate),
    Mean_DNF_Rate = mean(object$DNF_Rate),
    Best_Finish_Rate = max(object$Finish_Rate),
    Worst_Finish_Rate = min(object$Finish_Rate)
  )

  class(summary_stats) <- "summary.driverReliability"
  return(summary_stats)
}

# Print method for summary object
print.summary.driverReliability <- function(x, ...) {
  cat("Summary of Reliability Metrics\n")
  cat("Mean Finish Rate:", round(x$Mean_Finish_Rate, 2), "%\n")
  cat("Mean DNF Rate:", round(x$Mean_DNF_Rate, 2), "%\n")
  cat("Best Finish Rate:", round(x$Best_Finish_Rate, 2), "%\n")
  cat("Worst Finish Rate:", round(x$Worst_Finish_Rate, 2), "%\n")
}
```

Visual Correlation: Reliability vs. Performance

The analytical suite concludes with a specialized plot method that leverages ggplot2 to map the relationship between mechanical reliability and race-day performance. By plotting the DNF Rate against the Average Finishing Position, this visualization identifies "at-risk" performers and high-consistency elites. The method automatically applies a reversed x-axis scale—aligning with the intuitive racing logic where a "lower" number (1st place) represents a better

result—and utilizes a minimal theme to keep the focus on data distribution. This scatter plot serves as a diagnostic tool for identifying drivers who may be fast but prone to retirements, as well as those who provide a "safe pair of hands" for constructors prioritizing consistent point finishes.

```
# Plot method for driverReliability objects
plot.driverReliability <- function(x, ...) {
  ggplot2::ggplot(x, ggplot2::aes(x = Avg_Position, y = DNF_Rate)) +
    ggplot2::geom_point(color = "red", size = 2) +
    ggplot2::scale_x_reverse() +
    ggplot2::labs(
      title = "Driver Reliability vs Performance",
      x = "Average Finishing Position",
      y = "DNF Rate (%)"
    ) +
    ggplot2::theme_minimal()
}
```

```
rel <- driver_reliability(f1_data, min_races = 20)

print(rel)
```

```
 Formula 1 Driver Reliability Report
         Driver Finish_Rate DNF_Rate Avg_Position
 Max Verstappen        95.7      4.3         2.53
   Lando Norris        93.5      5.4         5.66
  Oscar Piastri        92.9      5.7         5.65
 George Russell        91.3      7.6         5.23
 Lewis Hamilton        90.2      7.6         5.80
```

```
summary(rel)
```

```
Summary of Reliability Metrics
Mean Finish Rate: 85.53 %
Mean DNF Rate: 13.39 %
Best Finish Rate: 95.65 %
Worst Finish Rate: 75 %
```

```
plot(rel)
```

Summary of Competitive Dynamics

The report concludes that the regulatory changes introduced in the 2022 season significantly influenced the competitive landscape of Formula 1. By examining race outcomes, driver consistency, and team performance, the analysis identified distinct patterns in how teams adapted to the new technical rules. The data reveals that while driver skill remains paramount, the mechanical reliability of constructors is a critical determinant of season-long success, particularly in the context of the high "Did Not Finish" (DNF) rates observed in certain performance tiers.

Analytical Framework and Reliability

A central achievement of this report is the development of a robust data processing pipeline that standardizes complex racing metrics. Key takeaways from the methodology include:

- **Enhanced Data Utility**: By converting inconsistent time strings and non-numeric finishing positions (like "NC" or "DQ") into standardized numerical values, the report provides a more accurate basis for statistical modeling.
- **Driver Reliability KPIs**: The introduction of the driverReliability S3 class allows for the systematic tracking of Finish Rates and DNF Rates, offering a quantifiable measure of driver and car performance beyond mere points scored.

- **Visual Insights**: The correlation between average finishing positions and reliability metrics—visualized through custom plotting methods—highlights that elite performance is inextricably linked to technical consistency.

Final Outlook

Overall, the 2022–2025 seasons have demonstrated a stabilizing competitive field where data-driven strategies are essential for navigating technical volatility. The analytical tools developed for this report serve as a foundation for future predictive modeling, enabling teams and researchers to better anticipate race outcomes based on historical reliability and qualifying trends.