Data-Driven Visualization of Reliability and Race Performance in Formula 1

Mr. Murtuza Shabbir Lemonwala, Embry-Riddle Aeronautical University - Daytona Beach

Murtuza Lemonwala is a doctoral student in Mechanical Engineering, focusing on the development and application of Quantum Neural Networks. He earned his Master of Science in Mechanical Engineering with a concentration in Robotics and Autonomous Systems, gaining research experience in control systems, neural networks, and machine learning. He holds a dual Bachelor's degree in Aerospace Engineering and Mechanical Engineering, specializing in Astrodynamics and Propulsion. His research interests span intelligent systems, data-driven modeling, and computational reliability analysis. This study was motivated by his long-standing passion for Formula 1 and the desire to create a data visualization framework that preserves interpretability and human evaluation in an era increasingly shaped by artificial intelligence.

Ms. Kayla D. Taylor, Embry-Riddle Aeronautical University - Daytona Beach

Kayla Taylor is working toward her PhD in Electrical Engineering and Computer Science at Embry-Riddle Aeronautical University. She also holds a BS in Astronomy and Astrophysics and an MS in Aviation with a specialization in Space Studies from Embry-Riddle. Her research interests focus on balancing the opportunity and impact of human innovation, scientific inquiry, artificial intelligence, transportation, and space exploration. She is an avid technical writer and STEM communicator, and she enjoys working with her colleagues to conduct interdisciplinary research.

Caroline Howard, Embry–Riddle Aeronautical University, Daytona Beach
Brock Edward Decker, Embry–Riddle Aeronautical University, Daytona Beach

Data-Driven Visualization of Reliability and Race Performance in Formula 1

Abstract

Formula One (F1) is a data-rich environment that provides substantial analytical potential for data science educators and students. However, the complexity of F1 makes it difficult to extract meaningful insights without data visualization, particularly for non-experts or F1 novices. This paper describes a Tableau dashboard created by the authors that extends the principles of data science and visualization to the competitive and stochastic environment of F1 racing. We employed a data-driven framework to examine performance and reliability across constructors and drivers, offering an integrated perspective of a real-world domain with an abundance of historical, multi-attribute data. The datasets span the 2010–2024 F1 seasons, corresponding to the post-regulatory era that standardized vehicle architectures, power-unit designs, and point accumulation. The resulting data were visualized in Tableau using an interactive dashboard that integrates treemaps of cumulative points for constructors and drivers, circuits of five tracks displaying speed profiles for podium drivers, a grid visualization illustrating average points earned by starting position, and a reliability bar plot comparing the percentile reliability of top constructors and drivers normalized to the highest performer. Design choices are justified with key visualization concepts, such as Shneiderman's Mantra, Tufte's principles, and Gestalt laws, culminating in an interpretable visual and analytic framework for F1 racing. In contextualizing the data collection, analysis, and visualization techniques used for the F1 environment, we hope this dashboard will be used as a teaching tool and offer a framework for data science educators to the teach data visualization techniques to their own students. Ultimately, the dashboard and design choices demonstrate how data science educators and students can embody effective storytelling through data visualization, highlighting the importance of visualization as a useful tool in data-driven engineering analysis and education.

Keywords: Formula 1, performance, reliability, data visualization, Tableau

Introduction

Since its first formal championship in 1950, Formula One (F1) has been globally revered as the premier motorsport racing competition [1], [2]. F1 consists of an annual World Constructors' Championship for the companies (i.e., "constructors," such as Mercedes or Ferrari) that design and build cars and the World Drivers' Championship for individual drivers (such as Lewis Hamilton or Michael Schumacher) [1], [2]. In the ongoing 2025 season, F1 races are scheduled to take place across 21 countries on five continents for a total of 24 races [3]. Although each race varies in circuit (track) morphology, length, and time elapsed, the majority of races are around 300 km and last between 90–120 minutes [1], [3].

Winning constructors and drivers are determined by the sum of points accumulated over a single season. Each constructor team enters two cars at each race with up to four drivers across a single season, and the points earned by both cars are added together to determine winning constructors for the World Constructors' Championship. The individual driver with the most points at the end of a season is crowned the winner of the World Drivers' Championship. The modern point system, which was instituted in 2010 to prevent insurmountable leads by legacy constructors, grants points to the top 10 finishers, with 1st place earning 25 points, 2nd place earning 18 points, 3rd place earning 15 points, 4th earning 12 points, 5th earning 10 points, and so on until 10th place earns just a single point [1]. The top three finishers constitute the "podium drivers" at the end of each race.

While the point system shapes competition, the undeniable popularity of F1 shapes its cultural identity. F1 boasts a global fanbase of 827 million, and a record 3.9 million fans attended the first 14 races of 2025, which is the highest mid-season attendance in F1 history [4]. Revenue estimates for the 2024 F1 season range from $3.4–$3.6 billion [5], reinforcing the sport's status as highly lucrative and broadly appealing. As a result of its cultural prominence and engineering-centric focus, F1 is an inherently data-rich environment that provides substantial analytical potential for data science educators and students.

This paper describes a Tableau dashboard created by the authors that extends the principles of data science and visualization to the competitive and stochastic environment of F1 racing. We employed a data-driven framework to examine performance and reliability across constructors and drivers using a Tableau dashboard, offering an integrated perspective of a real-world domain with an abundance of historical, multi-attribute data. In discussing our dashboard components and connecting our design choices to core data science visualization principles, this paper contributes to ongoing discussions in engineering education about how data science and data visualization can build students' analytical proficiency and graphical design literacy for future career readiness in engineering. We hope this dashboard will be used as a teaching tool and offer a framework for data science educators to the teach data visualization techniques to their own students.

Problem Statement

The complexity and variety of F1 data make it difficult to extract meaningful insights without data visualization, especially from raw data or tables alone. This raises interpretive issues for non-expert analysts who may be able to provide summative insights from statistical methods but lack the nuanced knowledge of F1 to explain or discuss results. Data visualization can significantly improve the ease and efficacy of longitudinal comparison of F1 data in a way that is useful and straightforward for various audiences, regardless of level of familiarity with F1 metadata. These audiences include engineering professionals, motorsports researchers, financially-motivated stakeholders, such as sponsors, and F1 enthusiasts. Emerging data science students who are learning to apply multi-variable visualization techniques in real-world domains can also benefit from illustrative frameworks of F1 data that support deeper, more personalized interpretation opportunities and decision-making for specialized and general audiences alike. Students and the educators who prepare them for applied analytical work comprise the target audience of this paper.

Despite the complexity of F1 data, it is easily accessible and publicly available online. These comprehensive historical datasets are well-suited for creating data visualizations and honing visualization-related skills. Kaggle hosts F1 data spanning information on drivers, constructors, lap times, pit stops, and more from 1950–2024, originally compiled from the Ergast F1 application programming interface [6]. There is also a Python library for F1 data called "FastF1" that provides timing and telemetry data [7]. Together, these datasets offer a foundation for analyzing F1 dynamics through visualizations. Data from both the Kaggle and the FastF1 library were used for this project to provide an integrated view of F1 data.

Background of the Problem

F1 analytics are further complicated by the interdependent variables that constitute concepts of "performance" and "reliability," which are challenging to quantify. Although performance is largely driven by the engineering of the car and the ability and talent of the driver [2], these qualities lack standardized operational definitions, making it difficult to rank drivers or predict success [8]. Reliability generally refers to the ability of a constructor or driver to complete a race, reflected by race finishing rate, but it can also encompass operational or mechanical reliability [9]. Data visualization can consolidate performance and reliability metrics into interactive plots and charts, enabling users to engage with a high-level overview, explore the data with details-on-demand, compare metrics side-by-side, and detect patterns or anomalies. In this way, viewers can synthesize multiple dimensions of F1 datasets that simultaneously contribute to performance and reliability.

Data visualizations and their host dashboards are increasingly more effective when design choices are based on widely adopted visualization principles. Without deliberate consideration regarding chart types, layout, or color, even aesthetically pleasing visualizations can obscure insights rather than reveal them. A Google search of "F1 Dashboards" will return numerous results of dashboards that incorporate F1 data into visualizations, but the design choices vary widely in interpretive value and clarity. For example, many of the dashboards available through the Formula 1 Dashboard [10] provide high-level overviews of selected attributes, but they do not always provide interactive features that would help users explore the data in finer or more specific detail. The theoretical foundations behind our visualization design choices will be discussed in the Theoretical Framework section.

Layout

This paper explores how to meaningfully address the challenges of analyzing F1 data by designing an interactive Tableau dashboard founded on core data visualization principles. Following this section is a summary of the key theories and principles behind the visualization design choices leveraged in our F1 dashboard, highlighting Shneiderman's Mantra, Tufte's Principles, and Gestalt laws, followed by a discussion of the data preparation. The resulting Tableau dashboard components are then discussed to explain how each visualization contributes to an overarching understanding of performance and reliability in F1. Finally, the paper concludes with implications for data science education, highlighting how real-world contexts, such as F1, can serve as powerful teaching tools for developing analytical reasoning and visualization literacy. We aimed to organize this paper to support readers who are educating future data science practitioners or developing their own expertise in data science and visualization in engineering contexts.

Theoretical Framework

Data-centric thinking is increasingly viewed as a necessity for engineering students [11], particularly given the synergy of the engineering design process and the data visualization creation process [12]. Before students dive headfirst into creating visualizations, they must first understand best practices and underlying key principles of data visualization to maximize visualization effectiveness. These concepts, including Shneiderman's Mantra, Tufte's principles, and Gestalt laws, constituted the foundation with which our F1 dashboard was built, helping us explore ways we could meaningfully represent F1 data and support user experience across expertise levels. This section will provide an overview of each of these data visualization concepts and heuristics; how each concept influenced visualization design will be discussed in the Dashboard Components and Problem Exploration sections.

Shneiderman's Mantra

Ben Shneiderman, one of the most influential researchers in human-computer interaction and data visualization, conceptualized "a useful starting point for designing advanced graphical user interfaces" [13]. He named this starting point the "Visual Information Seeking Mantra," which is colloquially referred to as "Shneiderman's Mantra:"

"Overview first, zoom and filter, then details-on-demand" [13].

Shneiderman emphasizes that visualizations should first present an overview of the entire dataset, then allow users to zoom and/or filter information to aid in specificity, and finally provide selection or hover capabilities so users can gain or focus on additional details on an as-needed basis. Although Shneiderman expands on this task taxonomy in his original paper [13], his abbreviated mantra has been widely cited as a straightforward consideration tactic for both novice and seasoned visualization designers.

Tufte's Principles

Edward Tufte is another prominent figure in data visualization and design, who is well known for his efforts to improve pedagogical practices in data accessibility and comprehension through his visualization principles. Tufte has a number of visualization principles that have been applied by practitioners and analysts since the 1980s, but three of his core principles include the following:

- Tell the truth about data
- Communicate ideas with clarity, precision, and efficiency
- Minimize the data-ink ratio; that is, "give the viewer the greatest number of ideas in the shortest time with the least ink in the smallest space" [14].

By prioritizing analytical content, such as context that aids in data interpretation, and minimizing "chartjunk" elements that detract from the overall message, Tufte's principles aid in the design of honest and cognitively efficient visualizations.

Gestalt Laws

The Gestalt laws refer to phenomena in the human perceptual experience that have been documented over time by various psychologists, beginning with Max Wertheimer in 1912 [15], [16]. Although the number and name of the laws can vary by source, each law pertains to how viewers infer structure through "perceptual grouping and figure-ground organization" [16]. Examples of relevant laws for our F1 dashboard include the following:

- Law of Proximity: Objects that are close together are perceived as related.
- Law of Similarity: Objects that are similar in appearance are perceived as related.
- Law of Continuity: Points on a line or curve are perceived as being related to each other.
- Law of Common Fate: Objects moving in the same direction are perceived as related.

Applying the Gestalt laws in design choices aligns data visualizations with natural perceptual processes, thereby reducing cognitive load and improving analytical clarity.

Rhetorical Interrogation

Effective visualization design also requires that creators interrogate their dashboards and consider what questions the visualization should help answer, even when dashboard users may not have preexisting questions (perhaps they are simply exploring the data) [17]. As we built the F1 dashboard, we identified several rhetorical questions that guided the design of our visualizations. These rhetorical questions and the resulting dashboard components are described in the Dashboard Components and Problem Exploration sections.

Data Preparation

This section describes the steps taken to preprocess data to subsequently develop the F1 dashboard in Tableau. Data preparation consisted of sourcing and collecting relevant F1 data and transforming the raw datasets into structured, tidy formats suitable for Tableau. All source code for data preprocessing was written in Python, which is available in Appendix A (Kaggle preprocessing) and Appendix B (FastF1 Library preprocessing).

We first explored what F1 data was available for free from online sources and quickly came across the Kaggle F1 datasets [6] and the FastF1 Python library [7]. Kaggle offered 14 specific F1 datasets as comma-separated value (CSV) files from 1950–2024. While most of the data from FastF1 dated back to 1950, timing data, session information, and car telemetry and position data were only available from 2018 onward [7]. Noting that there could be potential discrepancies or missing data points due to mismatched years, and recalling that 2010 marked the start of a "post-regulatory era" that standardized vehicle architectures, power-unit designs, and point accumulation in F1 [18], [19], we filtered the Kaggle datasets to include data from 2010–2024. Incoming data from the ongoing 2025 season was not included. We then manually reviewed the data attributes from Kaggle and FastF1 to consider what visualizations could highlight performance and reliability or provide complementary insights regarding F1 dynamics.

Although there was significant overlap in the data attributes from Kaggle and FastF1, we observed that the FastF1 library had high-resolution telemetry data that was not available in the Kaggle corpus. This observation led to a distinct separation of how we utilized the Kaggle and FastF1 datasets. The Kaggle data would support our main performance and reliability visualization components, whereas the FastF1 telemetry would enable visualizations of circuit dynamics and track geometry for the most recent F1 season (2024). We recognized that circuit visualizations would not directly provide insight regarding performance or reliability, but we felt they would be an essential component to an F1 dashboard to contextualize where races occur and how speed changes around a circuit. The features extracted from Kaggle and FastF1 are summarized in Table 1.

Table 1

Kaggle and FastF1 Features Extracted During Preprocessing

Source	Datasets	Number of Features Extracted	Features
Kaggle	circuits.csv, constructors.csv, drivers.csv, lap_times.csv, races.csv, results.csv	7	Race identifier, points, constructor, driver, grid position, circuit, season
FastF1	N/A	13	track coordinates (x, y, z), distance around the lap, speed, throttle, brake, drag reduction system (DRS), NGear, revolutions per minute (RPM), SessionTime, and LapTime and split values

Processing Kaggle Datasets

The preprocessing workflow for the Kaggle datasets began by merging the core attributes from the relevant CSV files and removing observations prior to 2010. To resolve inconsistencies in driver labeling, we cleaned the naming fields by filling missing three-letter driver codes and concatenating forename and surname values with whitespace removed. We also renamed duplicated "name" columns to explicitly distinguish constructor and circuit names. Finally, we computed total points for each driver-constructor-circuit combination and produced a constructor-aggregated and driver-aggregated Excel file with dense ranking based on accumulated points. The two Excel files were uploaded to Tableau.

Processing FastF1 Data

When we first considered how to display circuit visualizations using the FastF1 data, we knew we would need to balance representational diversity of circuit geography and driver speed gradients with the spatial constraints of the dashboard. As a result, we limited the dashboard to five random interchangeable circuits that only illustrate the speeds of the fastest lap of the podium drivers from the 2024 season.

Processing the FastF1 telemetry data involved retrieving session-level data for the five circuits and extracting each podium driver's fastest lap. The five circuits were Monaco, Great Britain, Italy, Belgium, and Austria; these circuits were randomly selected using a random number generator. For every selected driver-circuit pair, we used FastF1's built-in functions to load validated timing and positional telemetry streams. We then merged these sources on the *time* variable to produce a unified lap-level dataset containing speed, throttle, brake, gear, drag reduction system (DRS), revolutions per minute (RPM), and coordinate information. Because FastF1 occasionally omits certain attributes depending on data availability, the pipeline preserved missing values rather than interpolating them, providing a clean but truthful representation of telemetry resolution for each race. The individual per-driver CSV files were concatenated into a master dataset, yielding a file suitable for constructing circuit visualizations in Tableau.

Dashboard Components and Problem Exploration

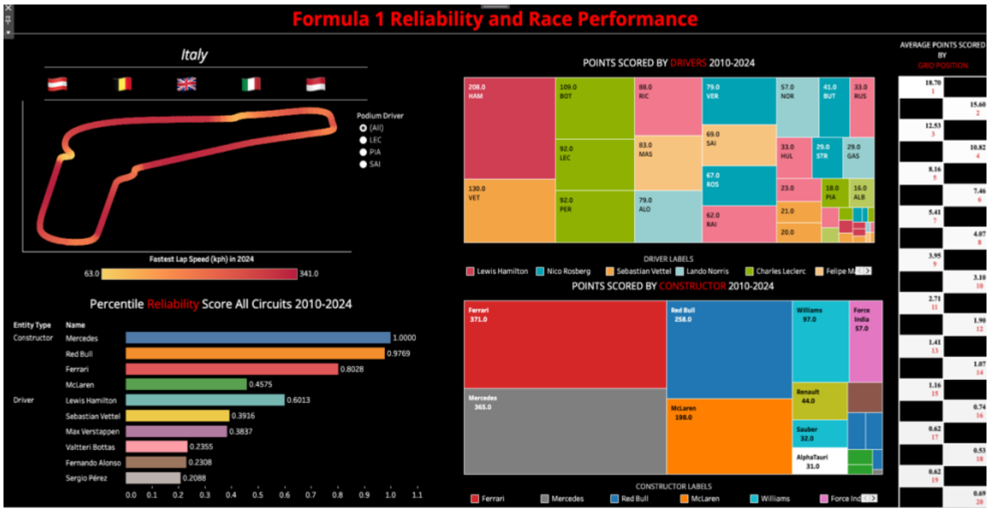
The processed datasets were then visualized in Tableau using an interactive dashboard. Ultimately, five visualizations were created to provide insight into performance, reliability, and F1 dynamics:

1. A treemap of cumulative points for drivers
2. A treemap of cumulative points for constructors
3. Circuits of five tracks displaying speed profiles for podium drivers
4. A grid visualization illustrating average points earned by starting position
5. A reliability bar plot comparing the percentile reliability of top constructors and drivers normalized to the highest performer

The final dashboard is shown in Figure 1, and it is available on Tableau Public [20].

Figure 1

F1 Reliability and Race Performance Dashboard on Tableau



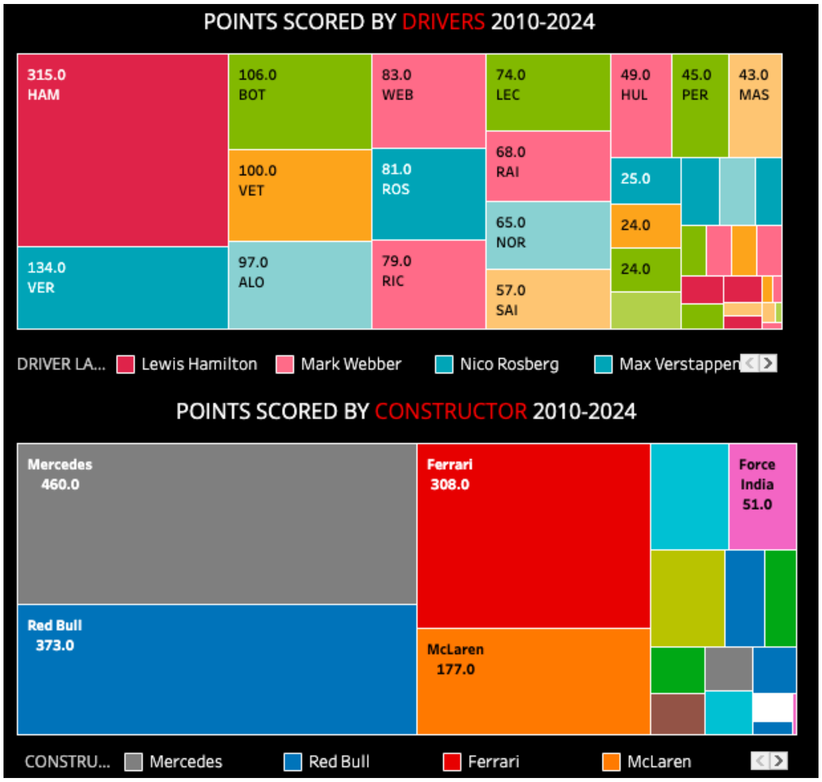
Treemaps

We first considered what questions users may pose regarding driver and constructor performance across seasons and circuits. Naturally, we concluded that questions like this could be answered with a visualization comparing point accumulation. Rhetorical questions that we considered included, *Which driver has accumulated the most points since 2010?*, *Which constructor has been most dominant since the new point system was instituted?*, and *Which drivers gain or lose relative standing when switching between races?* These anticipated questions guided our decision to display accumulated points for drivers and constructors as separate treemaps. We enabled dynamic filtering through a circuit flag selector, which updates the treemaps to align with the circuit visualization to reflect points earned by drivers and constructors from 2010–2024 at the selected circuit.

The size of each rectangle in the treemaps represents how many points the driver or constructor earned, and the colors distinguish individual drivers and constructors. We felt that treemaps were a particularly effective visualization to compare performance because they visually encode magnitude through area, allowing users to quickly identify the highest performers. We also enabled filtering capabilities to reveal how performance varies by context (driver, constructor, or circuit). The treemaps are presented in Figure 2.

Figure 2

Treemaps of Accumulated Points for Drivers and Constructors



The treemaps were designed with specific data visualization principles in mind to support clarity, interpretability, and meaningful comparison. Shneiderman's Mantra is echoed in how the treemaps provide an overview of point accumulation across drivers and constructors at one of the five circuits from 2010–2024. The interactive filters allow users to zoom in on a single circuit, driver, or constructor to provide focus with additional details-on-demand appearing through dynamic labels. Tufte's principles informed the visual encoding choices of providing the timeframe in the title and mapping area directly to point totals to maximize truthfulness and minimize the data-ink ratio. Finally, the Gestalt laws, particularly similarity and proximity, were applied to facilitate perceived structure within the treemaps. Drivers and constructors had consistent color assignments across circuits, and the proximity of rectangles encouraged users to compare areas. Collectively, we applied these principles to ensure that the treemaps communicate point distributions and foster intuitive exploration of performance.

Grid

We approached the next dashboard component by considering what users might want to know about the relationship between placement on the starting grid and race performance. In F1, starting placements are determined by the outcomes of three "qualifying" races on the Saturday of a Grand Prix weekend, referred to as Q1, Q2, and Q3 [21]. The ten slowest drivers are eliminated after Q1 and Q2, with each race eliminating five drivers. This decides the grid positions from 20^{th} to 11^{th}. The ten remaining drivers compete in the Q3 race to determine the top 10 grid positions. The fastest driver in Q3 is awarded the first grid position (called the "pole position") for Sunday's Grand Prix [21].

In designing the grid visualization, we identified several potential questions a user might ask about grid position and point accumulation, including *On average, which grid position accumulates the most points?* and *Does the average amount of points earned by grid position decrease with descending grid position?* Anticipating these questions led us to display the average points earned at each grid position, where each horizontal white bar represents a starting position from 1 to 20. The grid visualization adopts the characteristic color palette of the black and white checkered flag that marks the end of an F1 race to create a familiar visual context.

After considering rhetorical questions that users may ask related to grid position and performance, we transitioned our thinking to the core visualization principles that could maximize visualization effectiveness. Aligning with Shneiderman's Mantra, the grid visualization provides an immediate overview of average points earned from each starting position across the 2010–2024 seasons, enabling users to quickly see how starting order relates to performance. Tufte's principles influenced the color selection for the grid, leveraging the black and white monochrome palette for F1 familiarity while simultaneously allowing the red numerical labels for grid position to clearly stand out. The Gestalt laws reinforced our initial idea of structuring the grid in the same way that the grid is structured in F1 races (i.e., in a longitudinally staggered line). This structure corresponded to the laws of proximity, similarity, and common fate because the horizontal white boxes and red position labels appear perceptually related given their closeness, similar color scheme, and apparent movement in the same direction (i.e., away from the starting line). Together, these principles help the grid visualization effectively communicate how performance varies with starting position. The grid visualization is shown in Figure 3.

Figure 3

Grid Visualization of Starting Placement and Average Point Accumulation

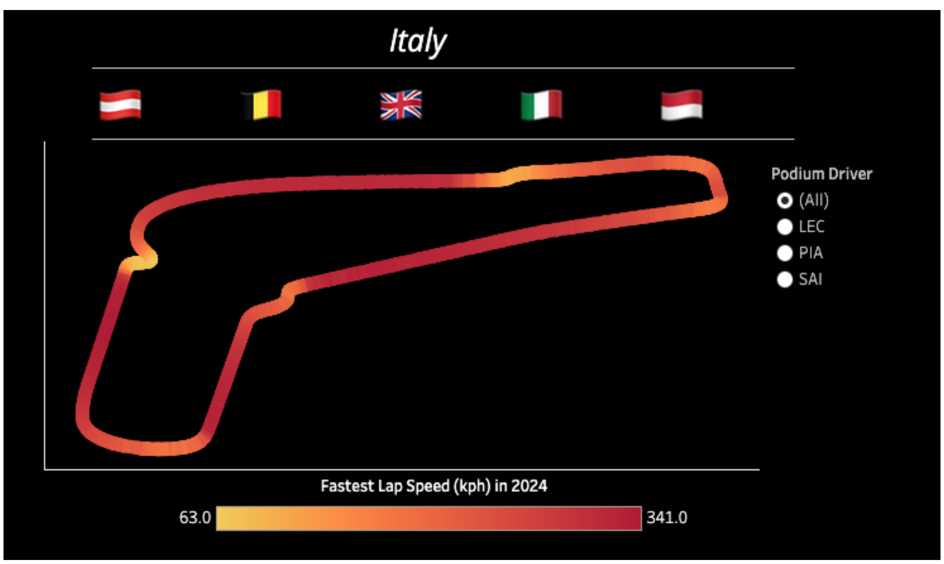
Circuit

Because each F1 circuit differs in geometry and speed profile, thus affecting the physical demands on the cars and driving strategies that contribute to performance, we anticipated user questions such as, *How does the shape of a circuit influence driver speed patterns?*, *Where on the track do podium drivers achieve the highest and lowest speeds?*, and *How do the fastest laps of podium finishers compare within a single race?* This narrative informed our decision to create a circuit visualization with a color gradient that reflects speed around the circuit. The circuit map is dynamic and can be changed by selecting a flag on the flag selector at the top of the circuit visualization. The different flags correspond to Monaco, Belgium, Great Britain, Italy, and Austria, which were the five random circuits chosen for inclusion on the dashboard, given the inherent space constraints of the dashboard layout. The color gradient represents each podium driver's fastest lap at that circuit in the 2024 season, with warmer (redder) colors representing higher speeds. Although the circuit visualization does not directly quantify reliability or long-term performance, it provides essential spatial context that improves understanding of how circuit geometry and speed relate to driver performance in 2024, supporting exploratory learning by connecting physical track attributes to telemetry patterns.

Once again, core data visualization principles were considered through the design of the circuit visualization to ensure that users could effectively and efficiently understand circuit dynamics and how it may contribute to performance. The circuit visualization draws upon Shneiderman's Mantra to provide an overview of each track's geometry, while the color-coded speed gradient supports zooming into places where speed varies along straight sections and curves. The flag selector enables filtering between five circuits, the podium driver selector enables filtering between the top three drivers at that circuit in 2024, and the accompanying fastest-lap details supply details-on-demand when the user hovers over an area of the circuit, completing Shneiderman's Mantra. Tufte's principles further informed our design choices: we had originally considered adding a map or country view with geographical borders so users could visualize where these circuits are located in the world. However, we felt this was too busy and could be considered "chartjunk," and we ultimately decided against the idea. The Gestalt law of continuity is most applicable to the circuit visualization, given that users will perceive the track as an uninterrupted lap path. In these ways, we designed the circuit visualization to support intuitive spatial reasoning, providing insight regarding how track geometry and speed shape performance. The circuit visualization is shown in Figure 4.

Figure 4

Circuit Visualization with Speed Gradient for Podium Drivers



Reliability Plot

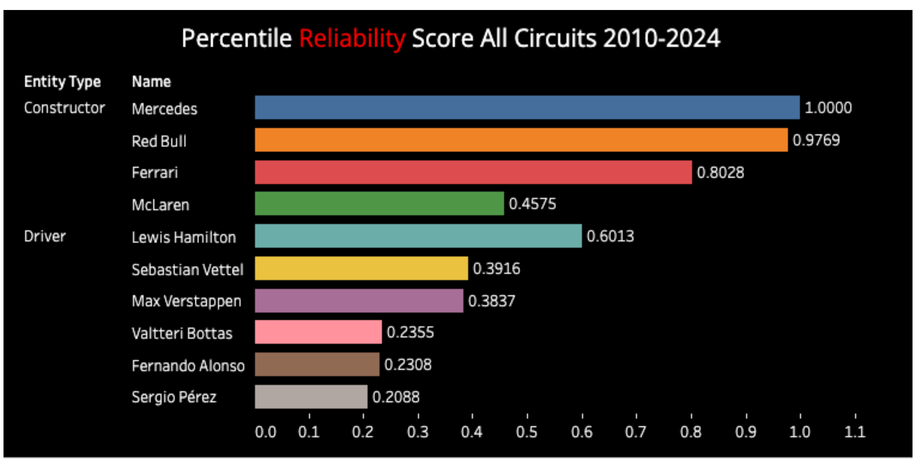
The reliability bar plot explicitly quantifies long-term reliability for drivers and constructors. To calculate reliability, we divided the cumulative points earned by a single driver or constructor across the 2010–2024 seasons by the highest point total achieved within the drivers or constructors as a whole. This normalization produced a percentile-like measure between 0 and 1, allowing users to compare how consistently each driver or constructor performed relative to the top performer for the 2010–2024 timeframe.

We again considered what questions users might generate regarding how reliable drivers or constructors are over time. We anticipated questions such as, *Which drivers and constructors demonstrate stable, high-performing race results over time?* and *How does long-term reliability compare between constructors and between individual drivers?* Incorporating a percentile-based bar plot that expresses reliability as a normalized measure relative to the highest performer in each category provides a clear representation of comparative reliability and allows users to immediately gauge which teams and drivers demonstrate prolonged competitive reliability.

We designed the reliability plot with the same visualization principles in mind that guided the other components of the F1 dashboard. Considering Shneiderman's Mantra, the reliability plot provides an overview of long-term reliability through normalized scores, and details-on-demand are available by hovering over the individual bars. Tufte's principles informed the choice of using color to distinguish entities rather than to decorate the visualization. The Gestalt laws further enhance interpretability of the reliability plot because the similarity in bar shape supports intuitive comparisons between drivers and constructors, and proximity organizes the bars into a coherent descending list that illustrates ranking. The reliability plot visualization is shown in Figure 5.

Figure 5

Circuit Visualization with Speed Gradient for Podium Drivers



Conclusion and Discussion

This project demonstrates how the data-rich environment of F1 can serve as a platform for developing data science and visualization competencies in engineering education. By working with various F1 datasets from varying online sources, we engaged directly with challenges common in professional engineering contexts, such as multi-attribute relationships, unclear variable definitions, and the need to translate raw data into interpretable insights. Constructing the Tableau dashboard required us to apply core visualization principles, including

Shneiderman's Mantra, Tufte's principles, and Gestalt laws, as practical design constraints that shaped the effectiveness of each visualization. In creating this dashboard, we experienced how user-centered design and principled data preparation support the design of meaningful data visualizations, especially in the context of complex systems that do not have operational definitions for certain metrics, which can also be common in engineering domains. In contextualizing the data collection, analysis, and visualization techniques used for the F1 environment, we hope this dashboard will be used as a teaching tool and offer a framework for data science educators to the teach data visualization techniques to their own students.

Future work on this dashboard could expand both its analytical depth and its pedagogical value by incorporating additional data sources, such as pit stop strategies or weather conditions, to enable deeper multi-variate exploration. Integrating machine learning models to predict reliability or race outcomes could further enrich the dashboard, giving us (as authors) or other engineering students who are building data visualizations the opportunity to build models that automate analysis, perhaps using live data from F1 races, and generate actionable insights, all of which mirror current industry practices. From an engineering education perspective, extending the dashboard in these ways would provide students with authentic, open-ended opportunities to practice systems thinking, experiment with alternative visualization techniques, and evaluate design choices through user testing. These future considerations could elevate the analytical capabilities of the dashboard and serve as a platform for teaching how visualization, computation, and domain knowledge intersect in modern engineering practice.

Acknowledgements

During the preparation of this work, the authors used ChatGPT for ideation, code assistance, organization scoping, and flow improvement. After using ChatGPT, the authors reviewed and edited the content as needed and take full responsibility for the content of the paper.

We also acknowledge that travel funding related to this project was supported by the Embry-Riddle Aeronautical University Office of Undergraduate Research (OUR).

References

[1] A. Serapiglia, "Formula One – a database project from start to finish," *Inf. Syst. Educ. J.,* vol. 16, no. 2, pp. 34–39, 2018.

[2] E.-J. van Kesteren and T. Bergkamp, "Bayesian analysis of Formula One race results: Disentangling driver skill and constructor advantage," *J. Quantitative Anal. in Sports*, vol. 19, no. 4, pp. 273–293, 2023.

[3] Formula One World Championship Limited. "Everything you need to know about F1 – Drivers, teams, cars, circuits and more." Accessed: Dec. 1, 2025. [Online.] Available: https://www.formula1.com/en/latest/article/drivers-teams-cars-circuits-and-more-everything-you-need-to-know-about.7iQfL3Rivf1comzdqV5jwc

[4] Formula One World Championship Limited. "Records, growth and unforgettable moments – the 2025 Formula 1 season so far." Accessed: Dec. 1, 2025. [Online.] Available: https://www.formula1.com/en/latest/article/records-growth-and-unforgettable-moments-the-2025-formula-1-season-so-far.1usXKBpUsYIndrYz22GQTq

[5] C. Kisby. "F1 prize money: How much do GP teams and drivers really make?" *MotorSport*, May 29, 2025. [Online.] Available: https://www.motorsportmagazine.com/articles/single-seaters/f1/f1-prize-money-how-much-do-gp-teams-and-drivers-really-make/

[6] Vopani, “Formula 1 World Championship (1950 - 2024),” Kaggle. [Online.] Available: https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020

[7] P. Schaefer. “Introduction to FastF1.” Accessed: Dec. 1, 2025. [Online.] Available: https://docs.fastf1.dev/

[8] E. El Haber, E. Sawaya, M. Attieh, A. Tannous, W. Ghazaly and M. Owayjan, “Formula 1 race winner prediction using random forest and SHAP analysis,” in *Int. Conf. Control, Automat., Instrum.*, Beirut, Lebanon, Feb. 2025, pp. 1270–1274.

[9] Formula One World Championship Limited. “The beginner’s guide to F1 engine and gearbox penalties.” Accessed: Dec. 1, 2025. [Online.] Available: https://www.formula1.com/en/latest/article/the-beginners-guide-to-formula-1-engine-and-gearbox-penalties.2TSy7BFgEvdNLojGLWS3F1

[10] Formula 1 Dashboard. “Formula 1 Dashboard | Race data and stats all in one place.” Accessed: Dec. 1, 2025. [Online.] Available: https://formula1dashboard.com/

[11] C. Ley, M. Tibolt and D. Fromme, “Data-centric engineering in modern science from the perspective of a statistician, an engineer, and a software developer,” *Data-Centric Eng.*, vol. 1, no. e2, 2020.

[12] V. L. Byrd, “Parallels between engineering graphics and data visualization: A first step toward visualization capacity building in engineering graphics design,” *Eng. Des. Graph. J.*, vol. 82, no. 2, 2019.

[13] B. Shneiderman, “The eyes have it: A task by data type taxonomy for information visualizations,” in *IEEE Symp. Vis. Lang.*, Boulder, Colorado, USA, September 1996.

[14] E. Tufte, *The Visual Display of Quantitative Information*, Cheshire, CT: Graphics Press, 1983.

[15] M. Wertheimer, “Experimental studies on the seeing of motion,” in *Classics in Psychology*, NY, NY: Philosophical Library, 1912, pp. 1032–1089.

[16] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M. Singh and R. von der Heydt, “A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organization,” *Psychol. Bull.*, vol. 138, no. 6, pp. 1172–1217, 2012.

[17] T. Munzner, *Visualization Analysis and Design*, Boca Raton, FL: CRC Press, 2015.

[18] Motorsport, “An analysis of the 2010 F1 regulations,” *Motorsport*, June 19, 2009. [Online.] Available: https://au.motorsport.com/f1/news/an-analysis-of-the-2010-f1-regulations/2932652/

[19] Reuters, “New F1 points system and other changes agreed,” *Reuters*, February 23, 2010. [Online.] Available: https://www.reuters.com/article/sports/motor-sports/new-f1-points-system-and-other-changes-agreed-idUSLDE61B12H/

[20] M. Lemonwala, K. D. Taylor, C. Howard, and B. Decker, “Formula 1 Tableau Dashboard,” Tableau Public. [Online.] Available: https://public.tableau.com/views/Formula1TableauDashboard/Dashboard1?:language=en-US&:sid=&:redirect=auth&:display_count=n&:origin=viz_share_link. [Accessed 9 February 2025].

[21] Formula One World Championship Limited. “The beginner’s guide to the F1 weekend.” Accessed: Dec. 2, 2025. [Online.] Available: https://www.formula1.com/en/latest/article/the-beginners-guide-to-the-formula-1-weekend.5RFZzGXNhEi9AEuMXwo987

Appendix A. Python Code Used for Kaggle Data Preparation

```
# -*- coding: utf-8 -*-
"""DataPreprocessing.ipynb

Automatically generated by Colab.

Original file is located at

https://colab.research.google.com/drive/1XwzeYu6j_19gNb8JqDW1hcz
dVM2RpATA
"""
"""## Group Project-DS 544
## Team Lead: Kayla Taylor
## Author: Murtuza Lemonwala
## **Dataset: Formula 1 World Championship (1950 - 2024)**

###Extracted Files: circuits, constructors, drivers and
lap_times
###Libraries used:

###Step 1: Extract data: Range 2010 to 2024, Select Needed
Columns, Merge
###Step 2:
###Step 3:
###Step 4: Build Analysis Table
###Step 5: Sort and Aggregate
"""

# Mount the Drive
from google.colab import drive
drive.mount('/content/drive', force_remount=True)

# Import Libraries
import os
import pandas as pd
import numpy as np

# Load Dataset
os.chdir('/content/drive/My Drive/Colab
Notebooks/DS544/Datasets')
dci=pd.read_csv('circuits.csv')
```

```
dco=pd.read_csv('constructors.csv')
ddr=pd.read_csv('drivers.csv')
dlt=pd.read_csv('lap_times.csv')
dres = pd.read_csv('results.csv',
usecols=['raceId','driverId','constructorId','points'])
drac = pd.read_csv('races.csv',
usecols=['raceId','year','circuitId'])

# Pre-Processing

# sum of points by driver-constructor-circuit since 2010)
df = dres.merge(drac, on='raceId', how='inner')

# join drivers / constructors / circuits with Kaggle column
names
df = df.merge(ddr[['driverId','code','forename','surname']],
on='driverId', how='left')

# constructors.csv
df = df.merge(dco[['constructorId','name']], on='constructorId',
how='left')
df = df.rename(columns={'name':'constructor_name'})

# circuits.csv
df = df.merge(dci[['circuitId','name']], on='circuitId',
how='left')
df = df.rename(columns={'name':'circuit_name'})

# clean labels
df['driver_code'] = df['code'].fillna(
    (df['forename'].fillna('').str[0] +
df['surname'].fillna('')).str.upper()
)
df['driver_name'] = (df['forename'].fillna('') + ' ' +
df['surname'].fillna('')).str.strip()

# filter (2010-2024)
df = df[df['year'] >= 2010].copy()

# group & sum points
pts_by_dcc = (
```

```
df.groupby(['driver_code','driver_name','constructor_name','circ
uit_name'], dropna=False)['points']
      .sum()
      .reset_index()
      .sort_values('points', ascending=False)
)

# Total points per driver (constructor-aggregated)
pts_by_driver = (
    df.groupby(['driver_code','driver_name','constructor_name'],
dropna=False)['points']
      .sum()
      .reset_index()
      .sort_values('points', ascending=False)
)
pts_by_driver['rank'] =
pts_by_driver['points'].rank(method='dense',
ascending=False).astype(int)

# Show first 15
pts_by_dcc.head(15), pts_by_driver.head(15)

# Save files, change this based on your personal colab
pts_by_dcc.to_excel('/content/drive/My Drive/Colab
Notebooks/DS544/Datasets/pts_by_dcc.xlsx', index=False)
pts_by_driver.to_excel('/content/drive/My Drive/Colab
Notebooks/DS544/Datasets/pts_by_driver.xlsx', index=False)

print("Both Excel files saved in your DS544/Datasets folder.")
```

Appendix B. Python Code Used for FastF1 Library Data Preparation

```
#  MASTER 2024 FASTEST-LAP PIPELINE FOR TABLEAU
#  - Pulls podium drivers for 5 races
#  - Extracts fastest lap telemetry
#  - Saves CSVs + merged dataset + track visuals

!pip install fastf1 requests-cache pandas matplotlib

import fastf1
import pandas as pd
import matplotlib.pyplot as plt
import os

os.makedirs("f1_cache_2024", exist_ok=True)
fastf1.Cache.enable_cache("f1_cache_2024")



race_drivers = {
    "Monaco": ["LEC", "PIA", "SAI"],
    "Great Britain": ["HAM", "VER", "NOR"],
    "Italy": ["LEC", "PIA", "SAI"],
    "Belgium": ["HAM", "RUS", "PIA"],
    "Austria": ["RUS", "PIA", "SAI"]
}


os.makedirs("telemetry_csv", exist_ok=True)
os.makedirs("track_visuals", exist_ok=True)

master_df = []


#process all races and drivers
for race_name, drivers in race_drivers.items():

    print(f"\n=== Loading {race_name} 2024 ===")
    session = fastf1.get_session(2024, race_name, "R")
    session.load()

    for drv in drivers:
```

```
        print(f"→ Processing: {race_name} — {drv}")

        # Get fastest lap
        laps = session.laps.pick_driver(drv)
        fastest = laps.pick_fastest()

        # Telemetry
        tel = fastest.get_car_data().add_distance()
        pos = fastest.get_pos_data()

        # Merge telemetry + position
        df = tel.merge(pos, on="Time", how="left")

        # Append metadata for Tableau
        df["Race"] = race_name
        df["Driver"] = drv
        df["LapTime"] = fastest["LapTime"]
        df["Compound"] = fastest.get("Compound", None)
        df["Team"] = fastest["Team"]
        df["LapNumber"] = fastest["LapNumber"]

        # Save individual CSV
        csv_path =
f"telemetry_csv/{race_name}_{drv}_fastestlap.csv"
        df.to_csv(csv_path, index=False)
        print(f"   ✓ Saved CSV: {csv_path}")

        # Add to master dataset
        master_df.append(df)

        # track visualization
        fig, ax = plt.subplots(figsize=(8, 8))

        sc = ax.scatter(
            df["X"],
            df["Y"],
            c=df["Speed"],
            s=3,
            cmap="turbo"
        )
```

```
        ax.set_title(f"{race_name} 2024 — {drv} Fastest Lap")
        ax.set_axis_off()

        cbar = plt.colorbar(sc, ax=ax)
        cbar.set_label("Speed (km/h)")

        png_path = f"track_visuals/{race_name}_{drv}_track.png"
        fig.savefig(png_path, dpi=300, bbox_inches="tight")
        plt.close(fig)

        print(f"   ✓ Saved Track Visualization: {png_path}")


# build master tableau dataset
master_df = pd.concat(master_df, ignore_index=True)
master_df.to_csv("2024_master_telemetry_tableau.csv",
index=False)
```