Module Code: CT051-3-M-DM

Module Name: Data Management

Assignment – Part 2

Structuring UK Train Journey Data for Analysis: Preprocessing, Feature Engineering, EDA, and Hypothesis Testing

Student Name: Muhammad Yousouf Ali Budullah

TP Number: TP086704

Intake Code: APDMF2501DSBA(BI)(PR)

Programme: MSc Data Science & Business Analytics

Module Lecturer: Dr. Murugananthan Velayutham

Date of Submission: May 29, 2025

Abstract

Effective analysis of railway performance is often reliant on structured, reliable data. The raw dataset of UK Train rides explored in Part 1 required significant cleaning to address typographical inconsistencies, missing values, and formatting issues. After preprocessing and cleaning the data, it facilitated better accuracy in grouping and analysis of journey trends. Conducting exploratory analysis revealed clear variation in delays and cancellations across departure stations, ticket types, and payment methods. Feature engineering presented new variables, including delay duration, route combinations, and temporal breakdowns such as hour and day of the week, which added depth and clarity to the dataset. Hypotheses were formulated to test relationships between delay severity and factors such as weather, time of day, and ticket price. While not all tests provided significant results, a relationship was found between severe delays and peak travel hours. Overall, the data preparation and transformation processes helped uncover meaningful patterns in service disruption and enhanced the dataset's suitability for visualisation, future predictive modelling and integration into data warehouse environments.

Table of Contents

List of Tables and Figures

1 Introduction

Railway networks play a considerable role in public transportation, but they are often reliant on how well disruptions such as delays and cancellations are handled. With growing commuter expectations and operational complexity, data driven insights have become necessary to improve performance. However, working with railway data brings its own challenges, spanning from dirty data like inconsistent inputs to missing values. These can easily affect how accurate and useful any analysis can be.

Part 2 of the assignment focuses on preprocessing the data, preparing and transforming the UK train rides data into a structured, cleaned format that can support in depth analysis. The dataset has a wide variety of fields such as ticket type, payment method, station names, delay reasons, and journey status. During Part 1 we identified several quality issues such as spelling inconsistencies in categorical fields, incomplete entries, and time format mismatches. Rather than applying automated fixes, data was cleaned manually or through contextual rules, ensuring that the logic behind each change was transparent.

Once the dataset was cleaned, it was explored using both graphical and statistical techniques to better understand the relationships between different variables. The aim wasn't to solely describe what the data illustrates, but to also identify useful insights, like whether prices varied with the class or time of purchase. Feature engineering was applied to transform and create new variables and was especially useful in transforming raw values into new variables that offered greater insight. For instance, calculating exact delay durations, categorising them into severity levels, extracting time-based features like hour of travel, and combining departure and arrival points into full routes.

After feature engineering was completed five hypotheses were ideated and tested to see whether certain assumptions were valid. They were evaluated using tests like chi-square and ANOVA. Though most displayed no significant relationship, others featured notable patterns such as severe delays being more likely during peak hours, or cancellation frequency being higher on specific routes. These provided an additional layer of context to the previous EDA and consolidated the case for more focused modelling in future work.

Overall, Part 2 of the assignment emphasised how crucial preprocessing and feature engineering are. Without them any patterns or trends seen in data would be hard to trust or explain. Utilising the available techniques to clean and engineer features not only supports better EDA but also prepares the data for predictive modelling or warehouse integration later on.

2 Related Works

While the dataset I'm working with isn't primarily about delays, most of the literature found that deals with train data tends to focus on delay prediction. However, these studies are still useful as they demonstrate how people have worked with similar types of data such as journey times, weather conditions, station information and more. This section looks at what the literature discusses about the variables used, how data was cleaned and pre-processed, and what sort of feature engineering was conducted. Rather than looking at the models themselves, we discuss what we can learn from how they prepared their data.

2.1 Understanding Influential Variables in Railway Data

Across the relevant literature, a consensus emerges regarding the types of variables used when working with train journey or delay-related datasets. Operational data, weather conditions, and temporal factors like time of day or calendar dates are often identified as key features. Tiong et al. (2023) for instance, group variables under categories such as operations, network, calendar, and maintenance, showing a structured way of approaching exploratory analysis. Wang and Yu (2021) delve further into the environmental side, demonstrating how even moderate changes in temperature, snow, or humidity can significantly impact a journey's punctuality. On the other hand, Lapamonpinyo et al. (2022) utilise both static and dynamic features, like ridership, geography, and historical delay profiles, suggesting that past performance and social patterns can be just as useful as operational statistics. These studies justify that even if the final goal isn't delay modelling, understanding how others approach journey-related variables can help shape a more centred and meaningful dataset.

3.2 Data Preprocessing and Cleaning Approaches

All the literature reviewed agree that poor data leads to poor results. The method in which data is cleaned, filtered, and handled during the early stages has a considerable impact. Marques et al. (2025) and Tiong et al. (2023) both adopt statistical thresholds like standard deviation and interquartile range to remove extreme outliers, which is especially useful in datasets that can be skewed by rare but severe disruptions. Marques et al. (2025) also states the importance of handling duplicates and missing entries early on, and document the process in detail. Tiong et al. (2023)

supports this with their approach of removing incomplete journeys that can't be used reliably and effectively.

Even in unconventional forms, preprocessing matters. Liu et al. (2023) focuses on unstructured text and apply methods like stop word removal and word segmentation before they even start building features. This facilitates consistency and makes their vectorisation process cleaner. While not a cleaning paper per se, Yong et al. (2025) raise the issue of inconsistency between preprocessing approaches across studies. Their AP-GRIP framework doesn’t provide a cleaning method, but it does point out how important it is to make preprocessing clear and repeatable. Together, these works support the idea that cleaning and transformation aren’t just technical but are part of how credibility and insight are built into the dataset.

3.3 Feature Engineering and Non-Traditional Data Sources

When discussing feature creation, the literature shows a lot of variation but also a lot of creativity. Lapamonpinyo et al. (2022) introduce features like Historical Delay Profiles at Destination, which aggregates past performance metrics for each route and time into something statistically useful. Marques et al. (2025) engineer network based features, like centrality and connectivity, that help explain how issues that occur in one part of the system might affect another. These structural metrics go beyond typical row-based features and are particularly relevant for rail systems with transfer points or shared lines.

Liu et al. (2023) take a different approach by using Word2Vec to convert delay reasons written in plain text into numeric vectors. This allows the integration of cause-based semantics into models that previously relied solely on operational metrics. More conventional methods like one-hot encoding are used by Tiong et al. (2023), who also carry out correlation checks to eliminate redundant features. Together, the literature supports a flexible but intentional approach to feature engineering, one that adapts to the dataset but follows clear logic and structure to enhance the dataset’s richness and uncover patterns not immediately visible in the raw data.

3 Methods

3.1 Data Pre-Processing

This section describes the data pre-processing methods applied to prepare the railway dataset for feature engineering and Exploratory Data Analysis (EDA). The decisions made here are based on the data quality assessment conducted in Part 1, where issues like missing values, inconsistent spellings, and format mismatches were discovered. As those inconsistencies were explored and confirmed earlier, we were able to apply direct rule-based corrections, rather than relying on general automated detection.

3.1.1 Correction of Categorical Inconsistencies

In Part 1, several categorical attributes were identified as having typographical inconsistencies. These fragmented frequency counts and raised a risk to accurate grouping and analysis. The following standardisations were applied.

For the Ticket_Type column, observations like "Adavnce" and "dAvance" were corrected to the correct spelling "Advance". In Journey_Status, the common misspellings "On Tmie" and "On Tiem" were replaced with "On Time" so that all on-time journeys would be grouped correctly.

The Reason_for_Delay field had a mix of issues. One common case was "Signal failure", which was updated to "Signal Failure" for consistency. There were also several values related to weather disruptions, including incomplete or inconsistent entries like "Weather", "Weather Conditi", and "Weather Condition". To simplify things and avoid fragmentation, any entry that included the word "weather" was replaced with a single label, "Weather".

```
/* Fixing typos in ticket type, journey status, and reason for delay */
data cleaned;
    set clean_railway;

    length Ticket_Type $8 Journey_Status $9 Reason_for_Delay $20;

    if lowcase(Ticket_Type) in ("adavnce", "davance") then Ticket_Type =
    "Advance";
    if Journey_Status in ("On Tmie", "On Tiem") then Journey_Status = "On
    Time";

    if index(lowcase(Reason_for_Delay), "weather") > 0 then
    Reason_for_Delay = "Weather";
    if Reason_for_Delay = "Signal failure" then Reason_for_Delay = "Signal
    Failure";
run;
```

Code Snippet 1: Correcting Typos in Categorical Fields

Reason_for_Delay	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Frequency Missing = 2623				
Signal Failure	88	23.34	88	23.34
Staff Shortage	29	7.69	117	31.03
Staffing	41	10.88	158	41.91
Technical Issue	71	18.83	229	60.74
Traffic	33	8.75	262	69.50
Weather	115	30.50	377	100.00

Table 1: Frequency of Reason for Delay after label correction

Ticket_Type	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Advance	1642	54.73	1642	54.73
Anytime	519	17.30	2161	72.03
Off-Peak	839	27.97	3000	100.00

Table 2: Frequency of Tickey Type after correcting inconsistent entries

Journey_Status	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Cancelled	162	5.40	162	5.40
Delayed	215	7.17	377	12.57
On Time	2623	87.43	3000	100.00

Table 3: Frequency of Journey Status after standardisation

3.1.2 Standardisation of Station Names

Part 1 also found variations in the spelling of departure station names, particularly for stations like Manchester Piccadilly and London St Pancras. In the Departure_Station column, several misspellings were observed that could have affected route analysis. These included typographical variations such as "amnchester piccadilly" and "macnhester piccadilly" for Manchester, and "London St Pnacras" for Pancras. These were corrected so that the same station would not appear under multiple labels.

```
/* Fixing known station name variants */
data cleaned;
    set cleaned;

    length Departure_Station $30;

    if lowcase(Departure_Station) in (
        "amnchester piccadilly", "mancehster piccadilly",
        "mnachester piccadilly", "macnhester piccadilly")
        then Departure_Station = "Manchester Piccadilly";

    if lowcase(Departure_Station) = "london st pnacras" then
        Departure_Station = "London St Pancras";
run;
```

Code Snippet 2: Standardising Station Names

Departure_Station	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Birmingham New Street	211	7.03	211	7.03
Bristol Temple Meads	3	0.10	214	7.13
Edinburgh Waverley	8	0.27	222	7.40

Departure_Station	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Liverpool Lime Street	421	14.03	643	21.43
London Euston	459	15.30	1102	36.73
London Kings Cross	382	12.73	1484	49.47
London Paddington	441	14.70	1925	64.17
London St Pancras	395	13.17	2320	77.33
Manchester Piccadilly	528	17.60	2848	94.93
Oxford	10	0.33	2858	95.27
Reading	50	1.67	2908	96.93
York	92	3.07	3000	100.00

Table 4: Frequency of Departure Station after fixing known typos

3.1.3 Contextual Imputation of Missing Values

In Part 1, several missing values were identified in key fields, but many of these were not due to data loss. Instead, they represented cases, such as passengers not using a railcard or journeys that arrived on time and therefore didn't have a delay reason. Rather than removing these records or leaving them null, values were imputed to improve interpretability and support later analysis.

For the Railcard column, most missing entries occurred when the passenger had no card. These were replaced with the label "No Railcard" to distinguish them from actual railcard usage. Similarly, in the Reason_for_Delay field, missing values were often linked to journeys marked "On Time". These were filled with "No Delay" to make that relationship explicit.

```
/* Handling missing values */
data cleaned;
    set cleaned;

    length Railcard $8 Reason_for_Delay $20;

    /* Assign 'None' to missing or blank Railcard values */
      if missing(Railcard) then Railcard = "None";

    /* Assign 'No Delay' if delay reason is missing but journey was on
time */
    if missing(Reason_for_Delay) and Journey_Status = "On Time" then
        Reason_for_Delay = "No Delay";
run;
```

Code Snippet 3: Imputing Missing Values

Railcard	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Adult	448	14.93	448	14.93
Disabled	279	9.30	727	24.23
None	1977	65.90	2704	90.13
Senior	296	9.87	3000	100.00

Table 5: Frequency of Railcard values after imputation

Reason_for_Delay	Frequency	Percent	Cumulative Frequency	Cumulative Percent
No Delay	2623	87.43	2623	87.43
Signal Failure	88	2.93	2711	90.37
Staff Shortage	29	0.97	2740	91.33
Staffing	41	1.37	2781	92.70
Technical Issue	71	2.37	2852	95.07
Traffic	33	1.10	2885	96.17
Weather	115	3.83	3000	100.00

Table 6: Final Freiquency of Reason for Delay after imputing missing values

3.1.4 Conversion of Time Formats

In Part 1, it was also observed that the original dataset contained mixed time and date formats (e.g., 2:00 PM vs 14:00:00, or March 1, 2024 vs 01/03/24). The imported time variables, such as Departure_Time, Arrival_Time, and Actual_Arrival_Time, were stored as numeric time values using the TIME20.3 format automatically. Since they were already appropriate for calculations, no further conversion was required. These values were kept as in their default formats and will be used directly for time-based feature engineering and analysis in the next sections.

3.2 Exploratory Data Analysis (EDA)

3.2.1 Journey Status Distribution

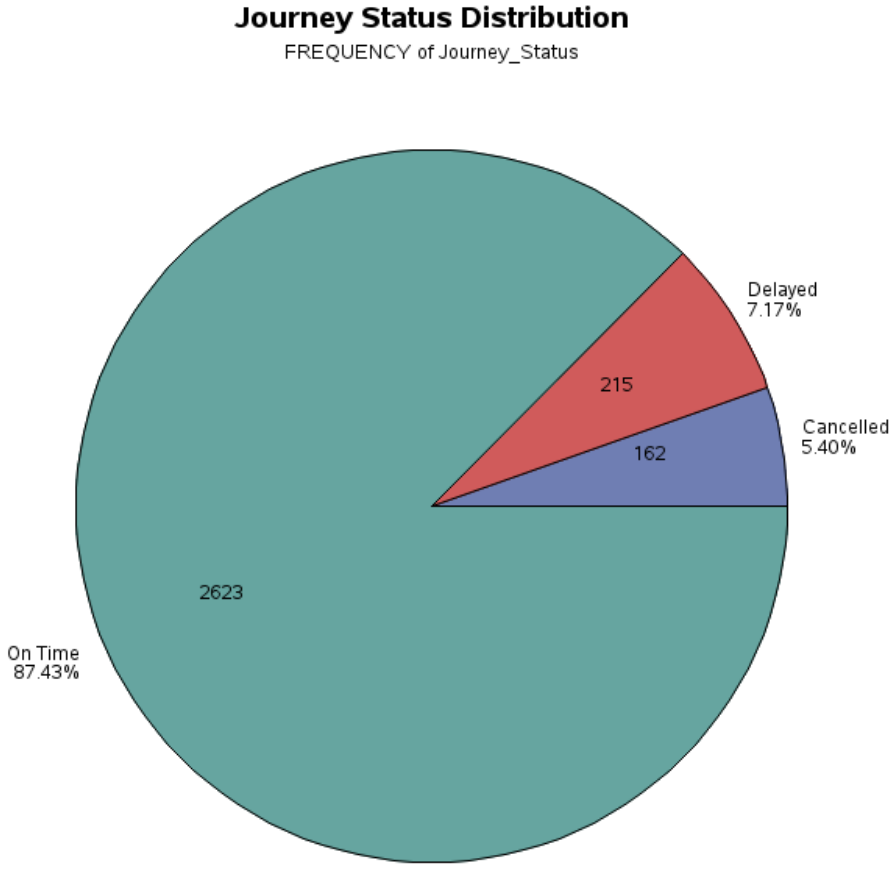

Figure 1: Journey Status Distribution Pie Chart

This pie chart shows the overall distribution of journey statuses in the dataset. A total of 87.43% of journeys were completed on time, while 7.17% experienced delays and 5.40% were cancelled. The proportions provide a clear numerical and visual summary of service performance across the sampled journeys. Majority of journeys were “On Time” and indicates a generally reliable service, but the remaining 12.6% being disrupted journeys highlights the need for further investigation into when and where delays and cancellations occur.

3.2.2 Journey Status by Payment Method

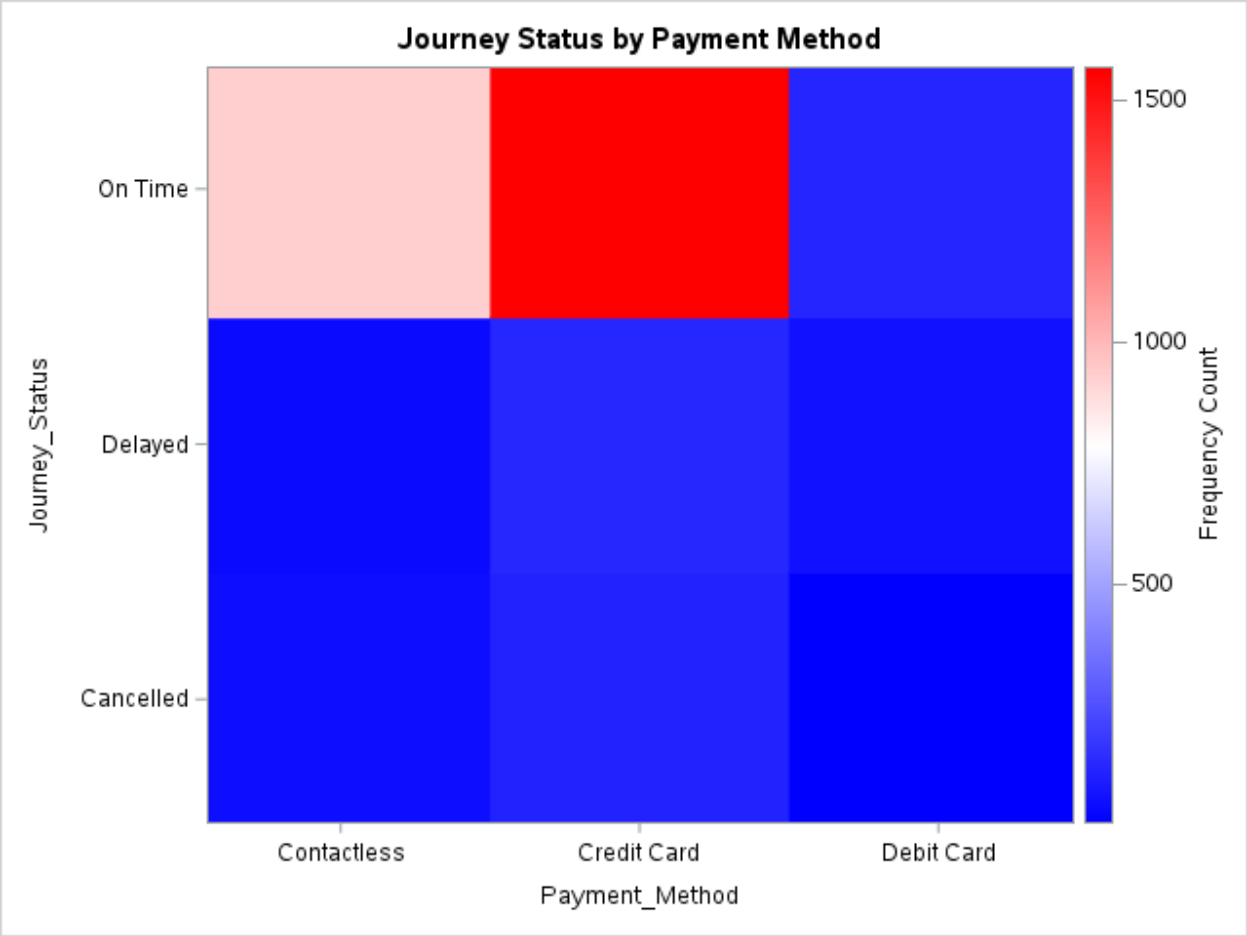

Figure 2: Heat Map Journey Status by Payment Method

The heatmap shows the relationship between journey status and the payment method used. Journeys paid using credit cards have the highest count of on-time completions, followed by contactless and debit card transactions. Delays and cancellations appear in all payment methods but with lower frequencies. There is no strong visual indication of one method being significantly more prone to disruptions than another, though minor differences may reflect user behaviour.

3.2.3 Journey Status by Departure Station

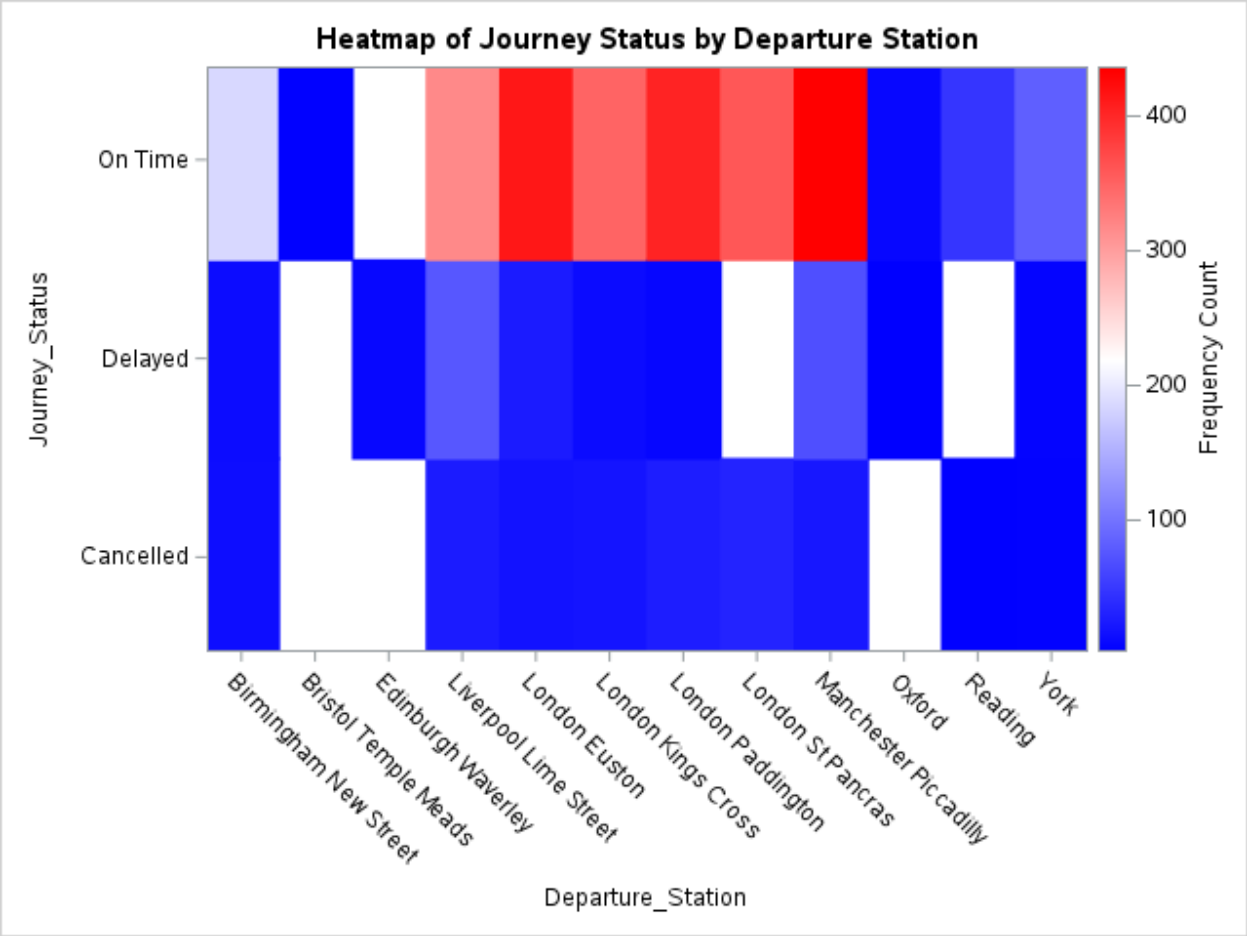

Figure 3: Heat Map Journey Status by Departure Station

The heatmap shows journey outcomes across different departure stations. Stations such as Manchester Piccadilly, London Paddington, and London Euston show the highest number of on-time journeys. Delays and cancellations are also present at these major stations but occur with lower frequency. Other stations display more even distributions, suggesting variation in service number or reliability depending on the station.

3.2.4 Journey Status Trend Over Time

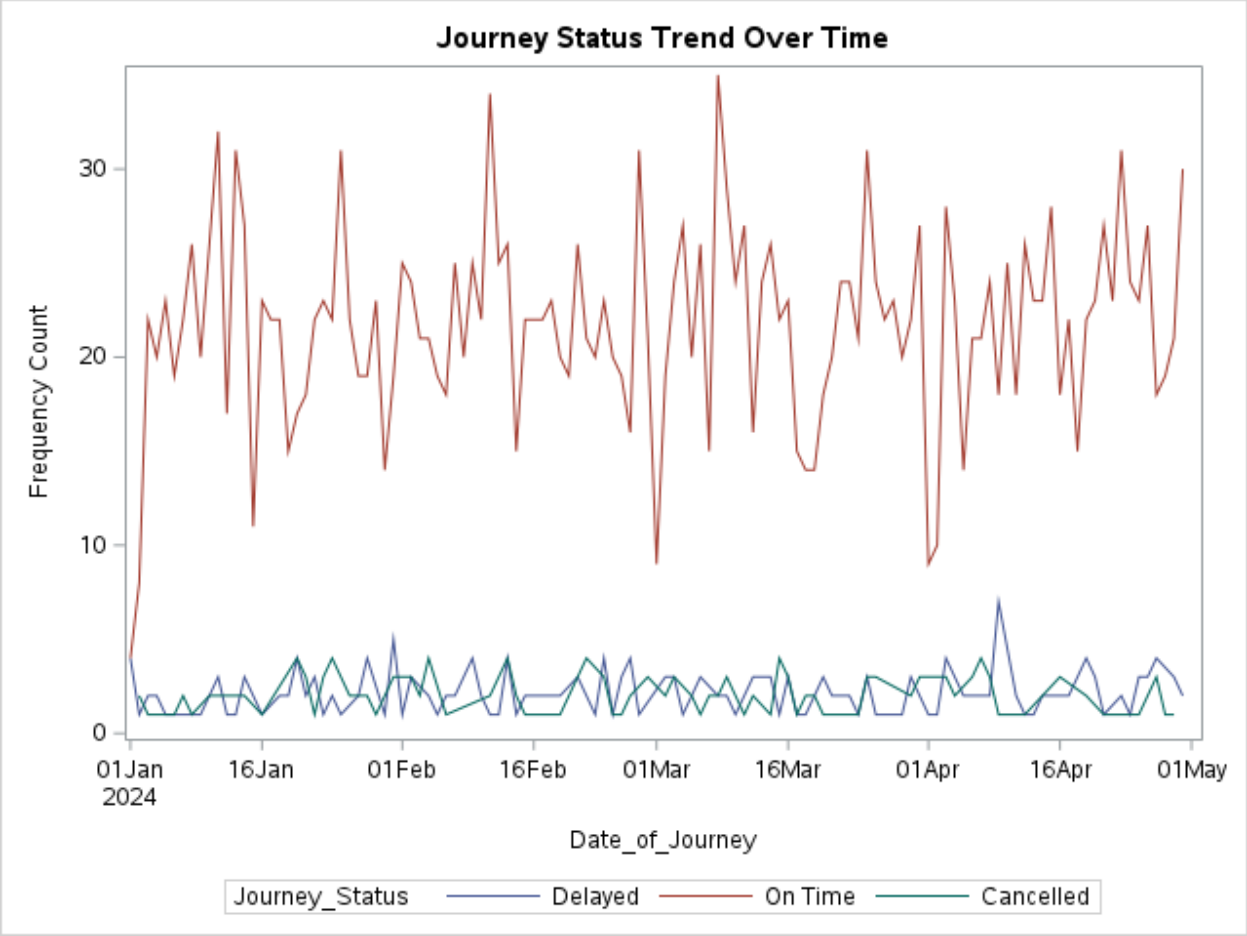

Figure 4: Line Chart Journey Status Trend Over Time

This line chart shows how journey status changes across time from January to May 2024. On-time journeys are consistently high each day, but there are visible drops and rises, suggesting operational fluctuations. Delays and cancellations occur at lower but stable level, with occasional peaks that may correspond to specific incidents or service disruptions.

3.2.5 Ticket Price by Class

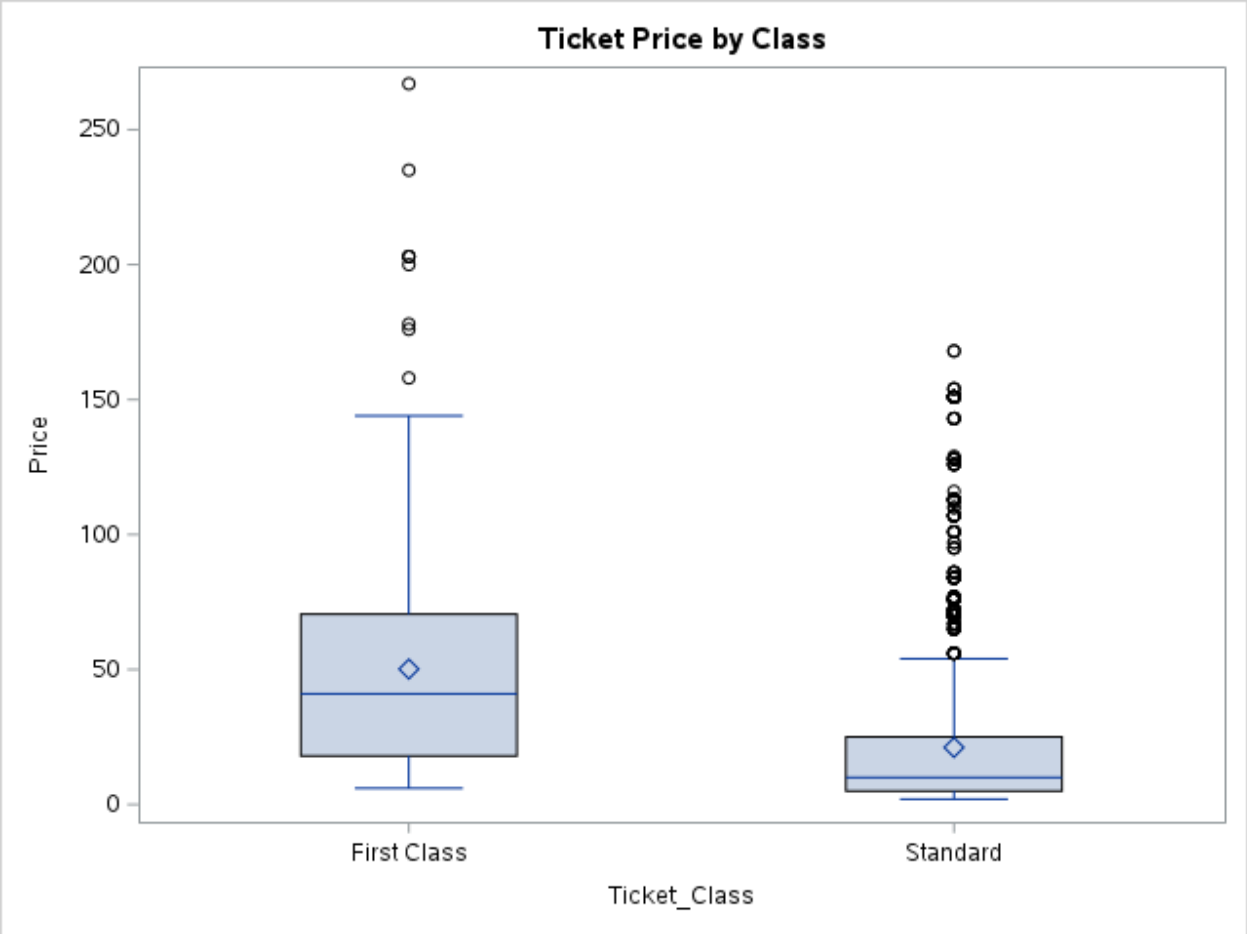

Figure 5: Box Plot Ticket Price by Class

This boxplot compares ticket prices between First Class and Standard. First Class clearly costs more on average and shows a wider price spread, with some fares exceeding £250. Standard tickets are more tightly clustered at the lower end, which matches general pricing structure. The difference in range also suggests that First Class tickets are subject to more variation in pricing, possibly due to flexible bookings or peak travel times.

3.2.6 Ticket Price by Ticket Type

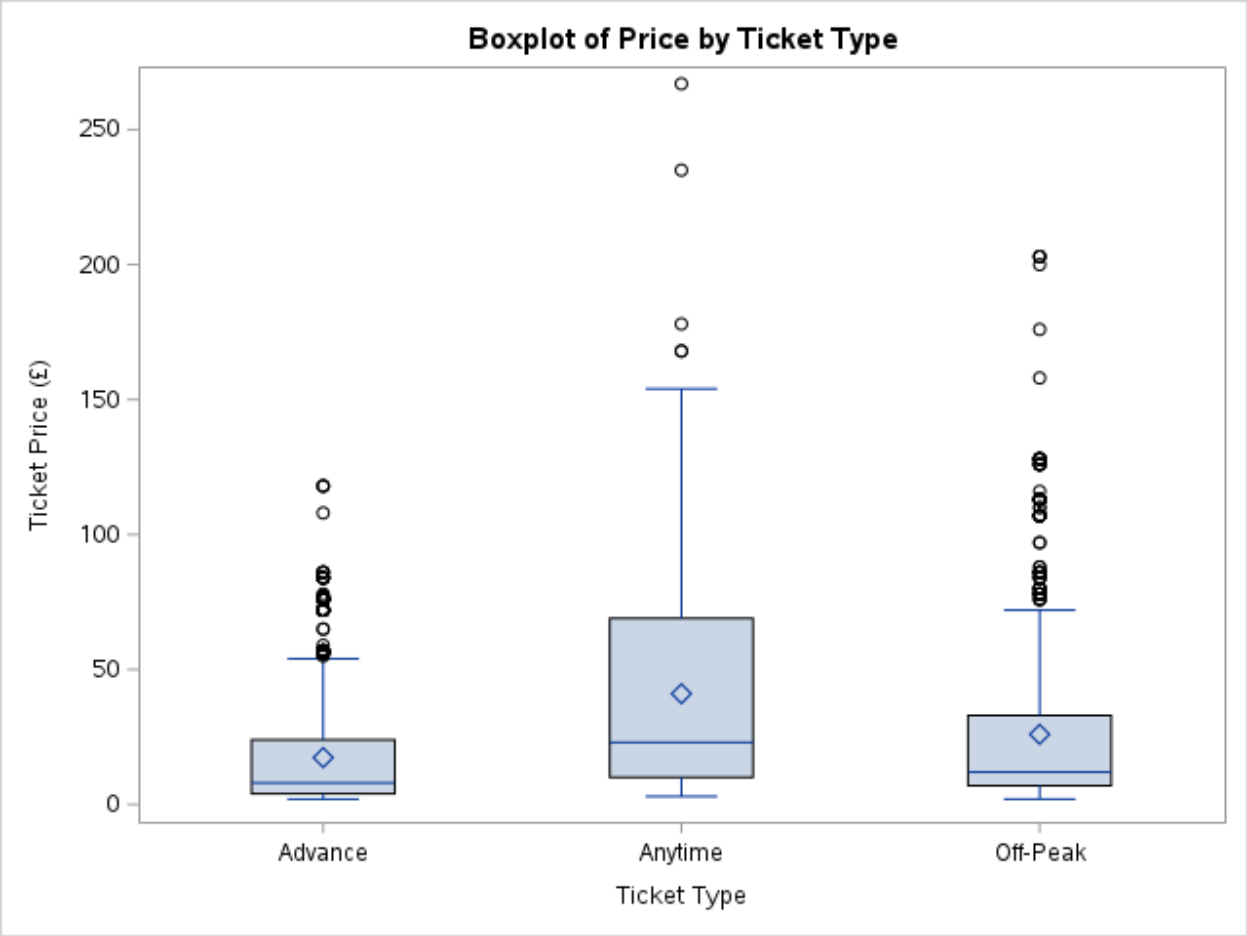

Figure 6: Box Plot Ticket PRice by Ticket Type

This boxplot shows how prices compare between Advance, Anytime, and Off-Peak tickets. Anytime fares clearly stand out with the highest median and widest spread, suggesting they're more flexible but also more expensive. Advance and Off-Peak tickets are lower in price and more tightly distributed, which fits their purpose as discounted or time restricted options. Outliers are present across all types, but they're especially high for Anytime tickets.

3.2.7 Ticket Price by Payment Method

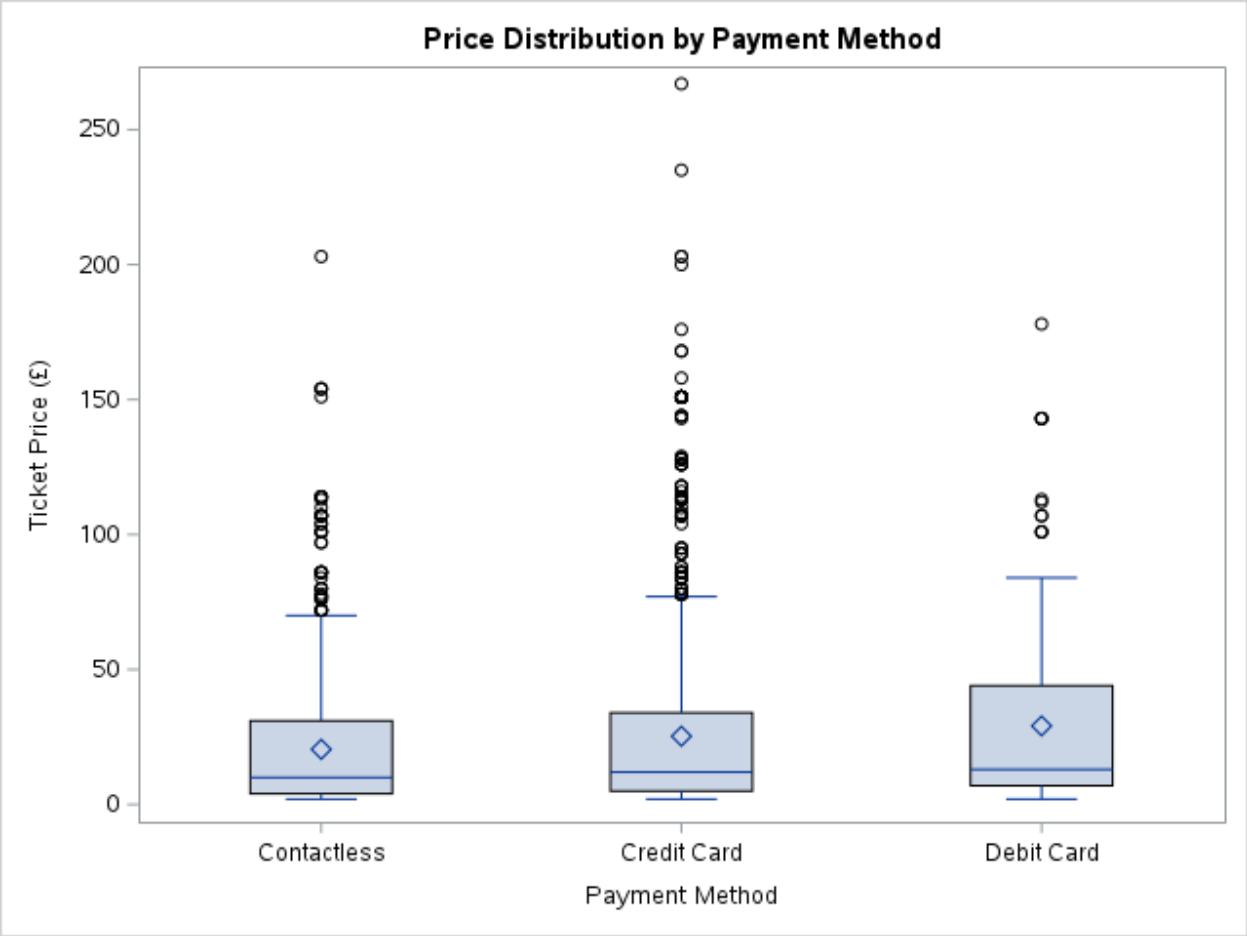

Figure 7: Box Plot Ticket Price by Payment Method

This boxplot compares ticket prices across contactless, credit card, and debit card payments. The spread is consistent across all three, but credit card payments show a slightly higher median and more extreme outliers. This could suggest that higher-value or more flexible fares are more commonly purchased with credit cards, possibly due to added payment protection.

3.2.8 Ticket Price by Time of Purchase

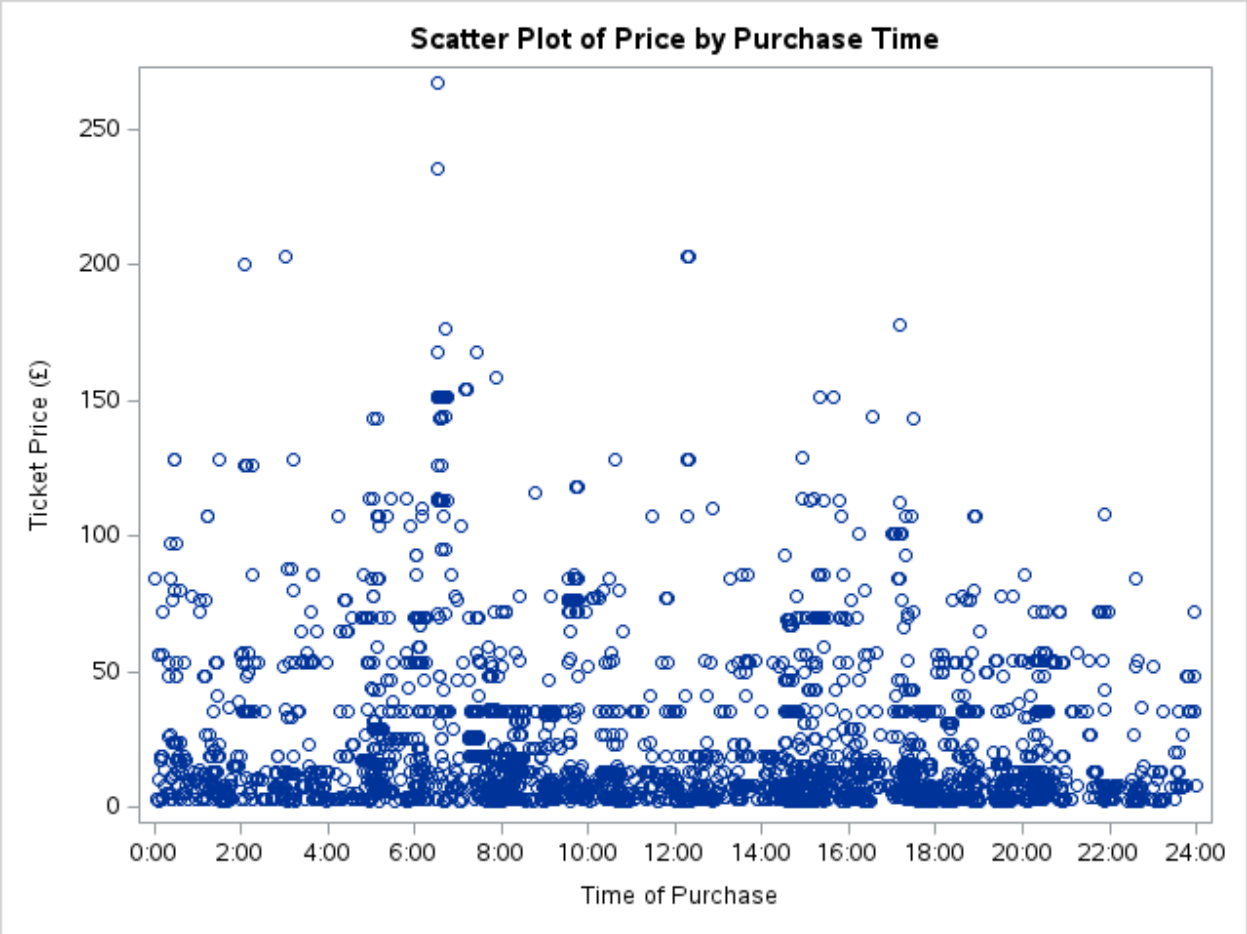

Figure 8: Scatter Plot Ticket Price by Time of Purchase

This scatter plot shows how ticket prices are distributed across different purchase times. Prices appear widely scattered throughout the day, with no visible trend, but there is a higher concentration of mid- to high-priced tickets in the early morning and late afternoon. This could reflect peak booking windows or time sensitive price fluctuations, possibly because of commuter demand or last-minute purchases.

3.3 Feature Engineering

Feature engineering helps us enhance the structure and analytical value of the railway dataset by transforming and creating new variables. This section describes the feature engineering techniques performed in SAS Studio, building on the preprocessed dataset from Section 3.1. The feature engineering is divided into two parts, variable transformation and feature creation.

3.3.1 Variable Transformation

3.3.1.1 Log Transformation of Price: Log Transformation

The Price variable was found to be right-skewed due to a small number of high-value tickets in Part 1. A new variable called Log_Price was created using the natural logarithm to mitigate this skewness and improve the distribution. This makes it easier to analyse price trends and compare them across journeys with different ticket types.

```
/* Log transform for right-skewed price variable */
data engineered;
    set cleaned;

    if Price > 0 then Log_Price = log(Price);
run;
```

Code Snippet 4: Log transformation to Price

Moments			
N	3000	**Sum Weights**	3000
Mean	2.52645753	**Sum Observations**	7579.3726
Std Deviation	1.13353211	**Variance**	1.28489503
Skewness	0.28542262	**Kurtosis**	-0.8283745
Uncorrected SS	23002.3632	**Corrected SS**	3853.4002
Coeff Variation	44.8664619	**Std Error Mean**	0.02069537

Table 7: Moments of log-transformed Price

Basic Statistical Measures			
Location		**Variability**	
Mean	2.526458	**Std Deviation**	1.13353
Median	2.484907	**Variance**	1.28490
Mode	1.098612	**Range**	4.89410
		Interquartile Range	1.91692

Table 8: Basis statistical Measures of log-transformed Price

The log transformation applied to the Price variable successfully reduced skewness and normalised the distribution. The resulting Log_Price has a skewness of 0.285, demonstrating near symmetry. The mean 2.53 and median 2.48 are closely aligned. This confirms the transformation improved the distribution and made the variable more suitable for analysis.

3.3.1.1 Grouping Delay Reasons: Grouping Operations

Reason_for_Delay included too many specific labels that fragmented frequency analysis. To simplify this, a new variable Reason_Group was created to group similar reasons into categories. This helps reduce the number of unique labels and makes it easier to analyse reasons for delays.

```
/* Group similar delay reasons into broader categories */
data engineered;
    set engineered;
    length Reason_Group $20;
    if Reason_for_Delay in ("Signal Failure", "Technical Issue") then
Reason_Group = "Infrastructure";
    else if Reason_for_Delay in ("Staff Shortage", "Staffing", "Traffic")
then Reason_Group = "Operational";
    else if Reason_for_Delay = "Weather" then Reason_Group = "Weather";
    else if Reason_for_Delay = "No Delay" then Reason_Group = "None";
    else Reason_Group = "Other";
run;
```

Code Snippet 5: Grouping Reason For Delays Values

Reason_Group	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Infrastructure	159	5.30	159	5.30
None	2623	87.43	2782	92.73
Operational	103	3.43	2885	96.17

Reason_Group	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Weather	115	3.83	3000	100.00

Table 9: Frequency of grouped Reason For Delays (Reason_Group)

3.3.2 Feature Creation

3.3.2.1 Delay Duration

Delay_Duration was created by calculating the difference between scheduled arrival time from actual arrival time. This gives a clear indicator showing how long each journey was delayed. This allows delays to be compared more precisely rather than just looking at if they were marked "Delayed". The delay duration was only calculated for journeys where the Journey_Status was "Delayed". For all other statuses, the values were left missing.

```
/* Calculate delay duration only for delayed journeys */
data engineered;
    set cleaned;
    if Journey_Status = "Delayed" and
       not missing(Actual_Arrival_Time) and
       not missing(Arrival_Time) then do;
        Delay_Duration = Actual_Arrival_Time - Arrival_Time;
        Delay_Duration_Minutes = Delay_Duration / 60;
    end;
run;
```

Code Snippet 6: Calculating Delay Duration in seconds and minutes

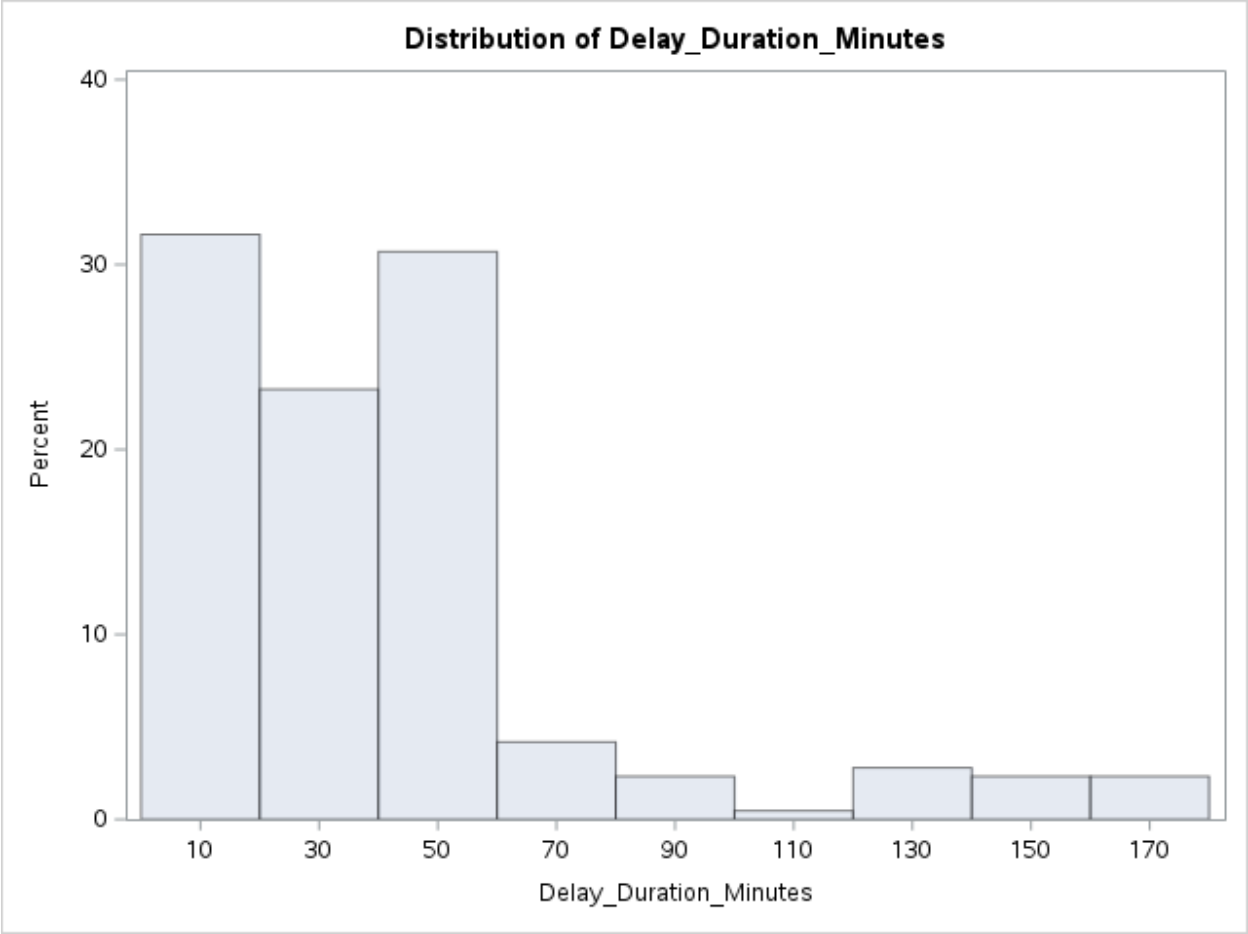

Figure 9: Histogram of Delay Duration in Minutes

3.3.2.2 Delay Category: Binning

To simplify the analysis of delays, a new variable Delay_Category was created by grouping delay durations into categories. Delay duration was measured in minutes, and the categories used were "No Delay", "Minor", "Moderate", and "Severe". Furthermore, journeys marked as "Cancelled" were given their own category to clearly separate them from journeys that occurred but were late.

The classification used both Journey_Status and Delay_Duration_Minutes to avoid incorrect labelling when delay data was missing or not applicable. This ensures that cancelled journeys were not mistakenly labelled as on-time, and that all other journeys were classified according to how late they were.

```
/* Categorise delay duration into bands */
data engineered;
    set engineered;

    if Journey_Status = "Cancelled" then Delay_Category = "Cancelled";
    else if Delay_Duration_Minutes = . then Delay_Category = "No Delay";
    else if Delay_Duration_Minutes <= 10 then Delay_Category = "Minor";
    else if Delay_Duration_Minutes <= 30 then Delay_Category = "Moderate";
    else Delay_Category = "Severe";
run;
```

Code Snippet 7: Categorising Delay Duration

Delay_Category	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Cancelled	162	5.40	162	5.40
Minor	23	0.77	185	6.17
Moderate	74	2.47	259	8.63
No Delay	2623	87.43	2882	96.07
Severe	118	3.93	3000	100.00

Table 10: Frequency of Delay Category

3.3.2.3 Hour of Departure: Extracting Date

To discover trends based on time of day, a new variable called Hour was created by extracting the hour from the Departure_Time field. Since SAS stores time values in seconds since midnight, the hour() function was used to isolate the hour component. This makes it possible to group and analyse journeys based on when they departed, such as comparing peak-hour delays versus off-peak performance.

```
/* Extract hour of departure from departure time */
data engineered;
    set engineered;

    Hour = hour(Departure_Time);
run;
```

Code Snippet 8: Extracting Hour from Departure Time

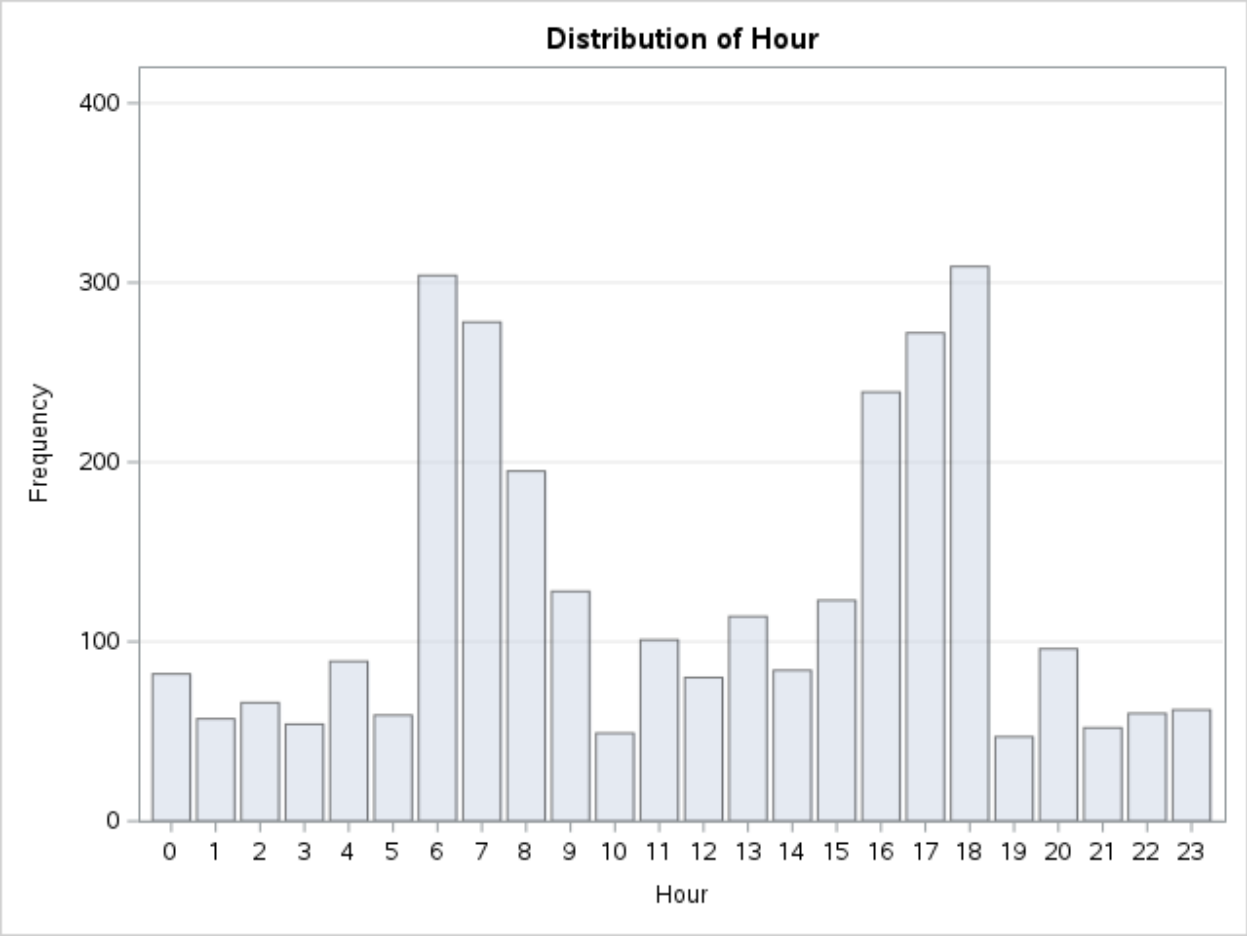

Figure 10: Histogram of Journey by the Hour

3.3.2.4 Day of the Week: Extracting Date

To analyse whether delays were more frequent on certain days, a new variable called DayOfWeek was created from the Date_of_Journey field. SAS stores dates as numeric values, so the weekday() function was used to extract the day of the week as a number (1 = Sunday, 2 = Monday, etc.). This allows journeys to be grouped by weekday to identify whether delays were more common on specific days, such as weekends.

```
/* Extract day of week from journey date */
data engineered;
    set engineered;

    DayOfWeek = weekday(Date_of_Journey);
run;
```

Code Snippet 9: Extracting Day from Date Of Journey

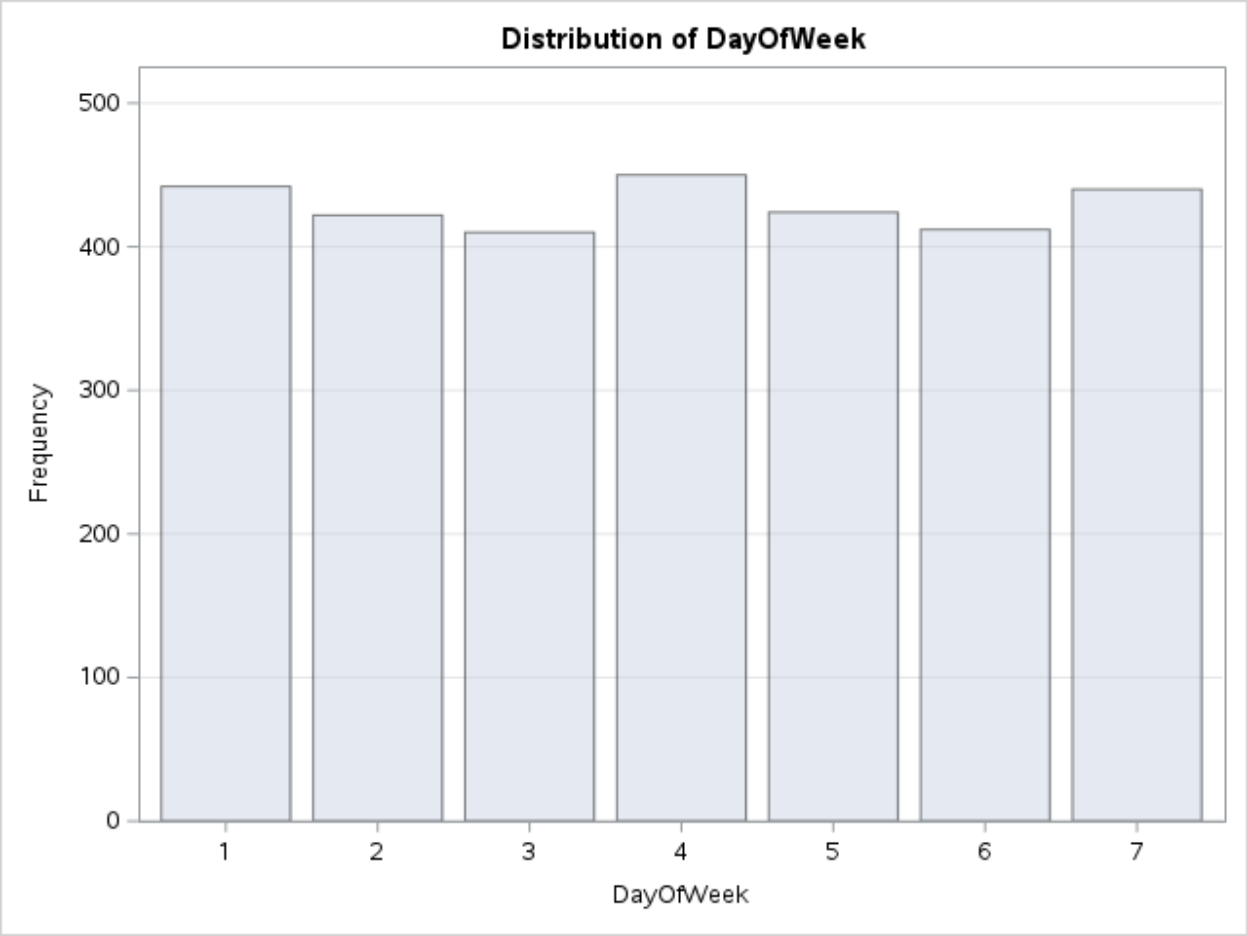

Figure 11: Histogram of Journeys by Day of the Week

3.3.2.5 Route: Feature Combination

To include route-based analysis, a new variable called Route was created by combining Departure_Station and Arrival_Destination. This allows journeys to be grouped by full route rather than by station itself. Creating this feature makes it easier to identify delay patterns or high-risk journeys between specific stations.

```
/* Extract day of week from journey date */
data engineered;
    set engineered;

    DayOfWeek = weekday(Date_of_Journey);
run;
```

Code Snippet 10: Creating Route from Departure Station and Arrival Station

Obs	Route	COUNT	PERCENT
1	Manchester Piccadilly to Liverpool Lime Street	429	14.3000
2	London Euston to Birmingham New Street	390	13.0000
3	London Paddington to Reading	373	12.4333
4	London St Pancras to Birmingham New Street	347	11.5667
5	London Kings Cross to York	346	11.5333
6	Liverpool Lime Street to Manchester Piccadilly	267	8.9000
7	Liverpool Lime Street to London Euston	107	3.5667
8	Birmingham New Street to London St Pancras	68	2.2667
9	London Euston to Manchester Piccadilly	66	2.2000
10	London Paddington to Oxford	59	1.9667

Table 11: Top 10 Most Common Routes

3.4 Hypothesis

3.4.1 Hypothesis 1: Delay severity varies by day of the week

H_0 (Null Hypothesis)

There is no association between the day of the week and the severity of the delay.

H_1 (Alternative Hypothesis)

The severity of the delay is associated with the day of the week.

```
/* Chi-square test to check if delay severity varies by day of the week */
proc freq data=engineered;
    tables DayOfWeek*Delay_Category / chisq plots=freqplot;
    title "Chi-Square Test: Delay Category by Day of Week";
run;
```

Code Snippet 11: Chi-Square Test for Association Between Day of the Week and Delay Severity

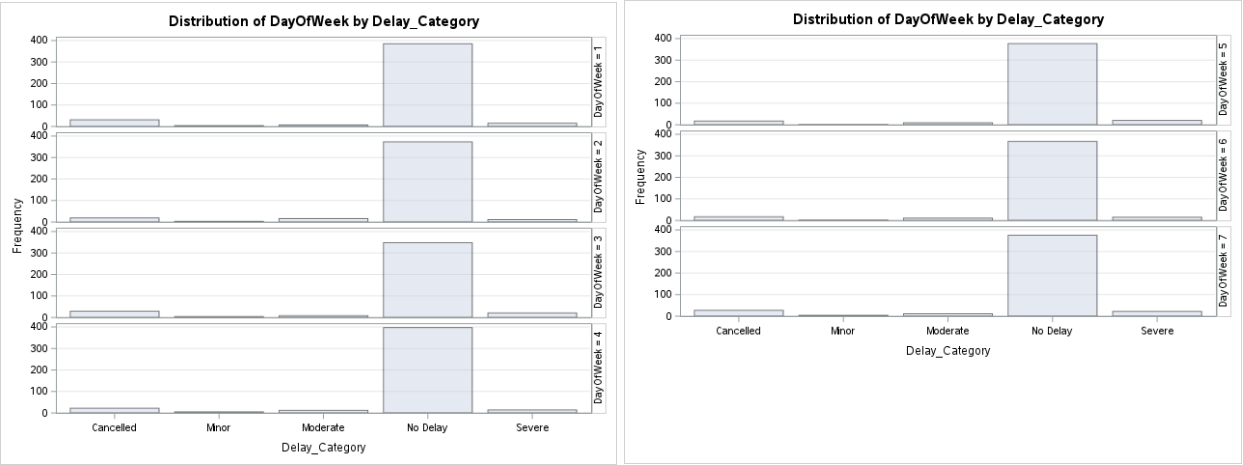

Figure 12: Bar Chart of Delay Categories by Day of the Week

The bar charts illustrates the distribution of delay categories across each day of the week. Visually, the proportions of "No Delay" journeys appear consistent, with only minor variation in delay and cancellation rates. The Chi-Square test yielded a p-value of 0.4643, which is above the standard 0.05 threshold for significance. As a result, we do not reject the null hypothesis and conclude that there is no statistically significant relationship between delay severity and the day of the week.

3.4.2 Hypothesis 2: Longer delays are associated with weather-related disruptions

H_0 (Null Hypothesis)

There is no difference in average delay duration across different delay reason groups.

H_1 (Alternative Hypothesis)

Average delay duration differs across reason groups with weather-related delays being longer.

```
/* One-way ANOVA to test if delay durations differ by reason group */
proc anova data=engineered;
    class Reason_Group;
    model Delay_Duration_Minutes = Reason_Group;
    title "ANOVA: Delay Duration by Reason Group";
run;c
```

Code Snippet 12: ANOVA Test Comparing Delay Durations Across Delay Reason Groups

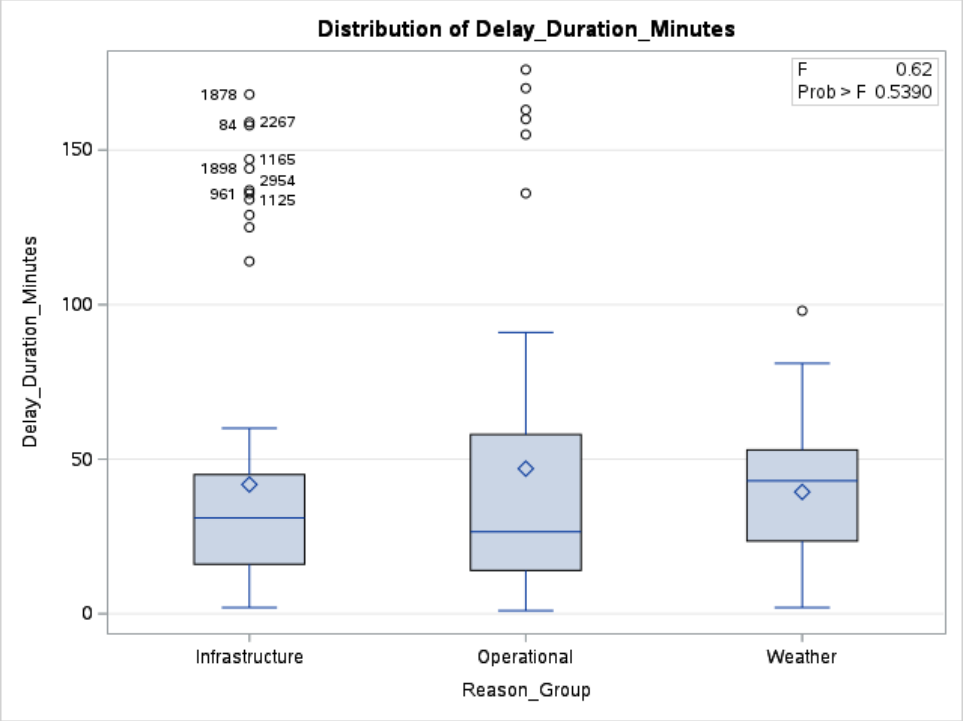

Figure 13: Boxplot of Delay Duration by Delay Reason Group

The boxplot shows the distribution of delay durations in minutes across grouped delay reasons. While Operational delays appear to have a slightly wider spread, the differences between groups are not large. The ANOVA test returned a p-value of 0.5390, which exceeds the 0.05 significance level. Therefore, we fail to reject the null hypothesis, concluding that there is no statistically significant difference in average delay durations across Infrastructure, Operational, and Weather-related disruptions.

3.4.3 Hypothesis 3: Severe delays are more common during peak hours

H_0 (Null Hypothesis)

There is no association between time of day and delay severity.

H_1 (Alternative Hypothesis)

Delay severity is associated with time of day, severe delays are more likely during peak hours.

```
/* Chi-square test to check if delay severity is associsated with time of
day */
proc freq data=engineered(where=(Journey_Status = "Delayed"));
    tables Hour*Delay_Category / chisq plots=freqplot;
    title "Chi-Square Test: Delay Category by Hour";
run;
```

Code Snippet 13: Chi-Square Test for Association Between Hour of Departure and Delay Severity

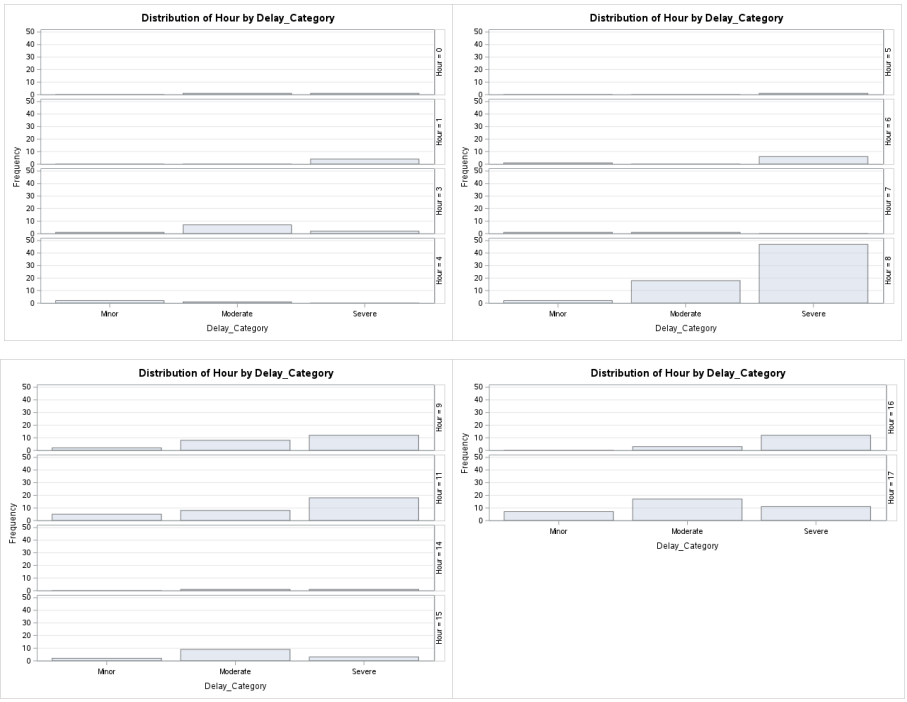

Figure 14: Frequency Plot of Delay Severity by Hour of Departure

The frequency plots illustrates variations in the distribution of delay categories across each hour of the day, with notable increases in severe delays around 8 AM and 5 PM. The chi-square test returned a p-value of 0.0003 which is less than the 0.05 significance level, indicating a statistically significant association between hour and delay severity. Based on this, we reject the null hypothesis and conclude that delay severity does significantly vary by hour of departure, suggesting a potential link with peak travel periods.

3.4.4 Hypothesis 4: Higher ticket prices result in fewer severe delays

H_0 (Null Hypothesis)

There is no difference in ticket price between delay severity categories.

H_1 (Alternative Hypothesis)

Ticket prices differ between delay severity categories, with severe delays being linked to lower-priced tickets.

```
/* ANOVA test to determine if price differs by delay category */
proc anova data=engineered;
    class Delay_Category;
    model Log_Price = Delay_Category;
    title "ANOVA: Log Price by Delay Severity";
run;
```

Code Snippet 14: ANOVA Test Comparing Log Ticket Prices Across Delay Categories

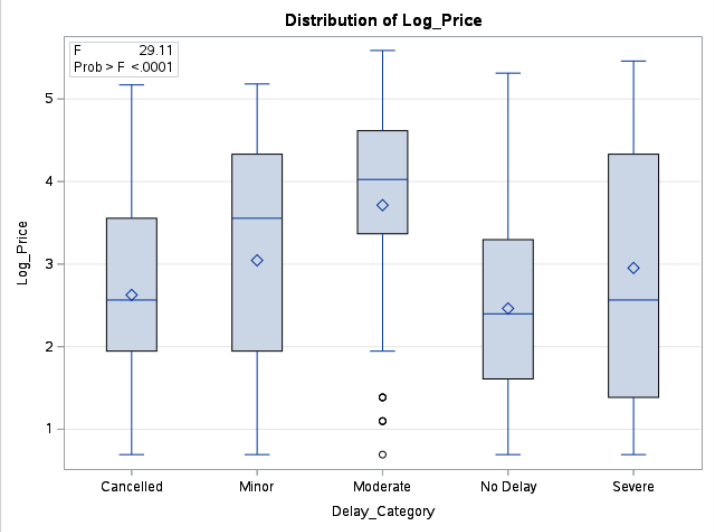

Figure 15: Boxplot of Log-Transformed Ticket Prices by Delay Category

A boxplot of log-transformed ticket prices was used to compare how pricing varies across delay categories. At a glance, “Moderate” delays had the highest median log price, while “Cancelled” and “Severe” delays showed lower price ranges. An ANOVA test was used to evaluate the statistical significance, returning a p-value of < 0.0001, well below the 0.05 threshold. Thus, we reject the null hypothesis and conclude that ticket prices significantly differ across delay severity categories. This implies that journey prices may be linked to service quality, potentially due to differences in ticket type, time of booking, or travel class.

3.4.5 Hypothesis 5: Some routes experience more cancellations than others

H_0 (Null Hypothesis)

Cancellation frequency is evenly distributed across routes.

H_1 (Alternative Hypothesis)

Some routes have significantly higher cancellation counts than others.

```
/* Frequency table for cancellations by route */
proc freq data=engineered(where=(Delay_Category = "Cancelled")) noprint;
    tables Route / out=cancel_routes;
run;

/* Sort by cancellation count */
proc sort data=cancel_routes;
    by descending count;
run;

/* Print top 10 routes with most cancellations */
proc print data=cancel_routes(obs=10);
    title "Top 10 Routes with Highest Cancellations";
run;
```

Code Snippet 15: Frequency Analysis of Cancellation by Route

Obs	Route	COUNT	PERCENT
1	London St Pancras to Birmingham New Street	29	17.9012
2	Manchester Piccadilly to Liverpool Lime Street	20	12.3457
3	London Kings Cross to York	18	11.1111
4	London Paddington to Reading	16	9.8765
5	London Euston to Birmingham New Street	15	9.2593
6	Liverpool Lime Street to London Euston	13	8.0247
7	Liverpool Lime Street to Manchester Piccadilly	11	6.7901
8	London Paddington to Oxford	11	6.7901
9	Birmingham New Street to London St Pancras	4	2.4691
10	Birmingham New Street to Manchester Piccadilly	4	2.4691

Table 12: Top 10 Routes by Number of Cancellations

A frequency analysis was conducted to identify which routes had the highest number of cancellations. The results showed a clear disparity. The route London St Pancras to Birmingham New Street had the highest number of cancellations (29), followed by Manchester Piccadilly to Liverpool Lime Street (20) and London Kings Cross to York (18). These three alone accounted for over 40% of all cancelled journeys. Based on this result, we reject the null hypothesis and conclude that cancellation frequency is not evenly spread across routes. This suggests that certain routes are more prone to disruptions and may require closer review.

4 Discussion & Conclusion

The analysis conducted in Part 2 demonstrated how crucial data preparation is when working with raw data such as the UK railway journeys. It highlights how important preprocessing and feature engineering are in transforming raw data into reliable data. The inconsistencies discovered in variables during Part 1 such as Journey_Status, Ticket_Type, and Reason_for_Delay fragmented frequency counts and made trends harder to detect. After cleaning typographical errors, imputing missing values and standardising values, the final distributions became clearer. For instance, “On Time” journeys accounted for 87.43% of entries, with “Delayed” and “Cancelled” making up 7.17% and 5.40%, respectively. Furthermore, grouping delay reasons into categories like Weather, Operational, and Infrastructure helped reduce noise and enabled more meaningful insights.

In addition, EDA helped in revealing pattern across station performance, pricing, and payment behaviour. Stations like Manchester Piccadilly and London Euston recorded the most journeys, as well as delays and cancellations. While no meaningful relationships were discovered between payment method and delay, pricing patterns were clearer. Boxplots showed that Anytime fares had the widest spread and highest median, whereas Advance and Off-Peak tickets were more tightly grouped.

Furthermore, feature engineering introduced depth to the analysis. Creating a Delay_Duration feature gave a continuous measure of lateness, while binning it into Delay_Category allowed for summary analysis. Extracting the Hour of departure, and day of the journey exposed patterns not apparent in raw timestamps, especially in relation to severe delays. Route analysis revealed that “London St Pancras to Birmingham New Street” had the most cancellations, accompanied by insights that would be lost without engineered variables like Route. It also prepares the data for any modelling that clients may wish to implement such as a predictive model for delays.

Finally, through hypothesis testing we discovered that delay severity varied significantly by hour of day, with peaks around 8 AM and 5 PM aligning with general commuter times. Ticket prices were also found to differ across delay categories with lower fares more likely to be related to cancelled or severely delayed journeys, this could be due to ticket flexibility or class. However, no

significant relationships were found between delay reason and delay duration or between day of the week and delay severity, implying that these factors may need wider context or more detailed data to explore fully.

This assignment gave me valuable insight into the flows and pipelines required in data management. It demonstrated how essential preprocessing is in contributing to effective visualisations and models. Cleaning inconsistencies and imputing missing values added clarity and made the data more meaningful to work with. I also learned the importance of exploring the data's domain and gaining a broader understanding to help me apply more creativity when engineering features and visualising relationships. It made the process feel more intentional.

I believe readers of this report will gain a better understanding of train ride data, especially within the railway domain where delay prediction is predominantly the focus. While the techniques and EDA I've conducted are insightful, there is still more to explore, particularly if future datasets include greater detail than the dataset I've used. In conclsion, this assignment strengthened my practical understanding of how data needs to be shaped before it can support deeper analysis or integration.

Bibliography

Lapamonpinyo, P. ;, Derrible, S. ;, & Corman, F. (2022). Real-time passenger train delay prediction using machine learning A case study with Amtrak passenger train routes Rights / license: Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International. https://doi.org/10.3929/ethz-b-000562375

Liu, Q., Wang, S., Li, Z., Li, L., Zhang, J., & Wen, C. (2023). Prediction of high-speed train delay propagation based on causal text information. Railway Engineering Science, 31(1), 89–106. https://doi.org/10.1007/s40534-022-00286-x

Marques, L., Moro, S., & Ramos, P. (2025). Data-driven insights to reduce uncertainty from disruptive events in passenger railways. Public Transport. https://doi.org/10.1007/s12469-024-00380-9

Tiong, K. Y., Ma, Z., & Palmqvist, C. W. (2023). Analyzing factors contributing to real-time train arrival delays using seemingly unrelated regression models. Transportation Research Part A: Policy and Practice, 174. https://doi.org/10.1016/j.tra.2023.103751

Wang, J., & Yu, J. (2021). Train performance analysis using heterogeneous statistical models. Atmosphere, 12(9). https://doi.org/10.3390/atmos12091115

Yong, T. K., Ma, Z., & Palmqvist, C. W. (2025). AP-GRIP evaluation framework for data-driven train delay prediction models: systematic literature review. In European Transport Research Review (Vol. 17, Issue 1). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1186/s12544-024-00704-7