Big Data - Practice 01

yehorbolt

Table of contents

Setup

Setup packages for working + render in Quatro.

```
# Set the CRAN mirror for package installation
options(repos = c(CRAN = "https://cloud.r-project.org"))

required_packages <- c(
  "arrow", "dplyr", "MASS", "car", "readxl", "rgl", "rmarkdown",
  "nortest", "latex2exp", "pca3d", "ISLR", "pls", "corrplot",
  "glmnet", "broom", "mvtnorm", "biglm", "leaps", "lme4",
  "viridis", "ffbase", "ks", "KernSmooth", "nor1mix", "np",
  "locfit", "manipulate", "mice", "VIM", "nnet"
)

install_if_missing <- function(p) {
  if (!require(p, character.only = TRUE)) {
    install.packages(p, dependencies = TRUE)
  }
}

lapply(required_packages, install_if_missing)
```

---

1. Load and Prepare Data

First, we load the required libraries, read the data, and perform initial cleaning.

Step 2: Inspect Raw Categorical Columns

Before making any changes, it is essential to inspect the raw data. This helps us understand the unique values in our categorical columns and confirms that the data matches our expectations from the data dictionary.

```r
all_cols <- names(taxi_data)

# Loop through each column
for (col_name in all_cols) {
  col_data <- taxi_data[[col_name]]
  cat(paste("\n**Audit of `", col_name, "`**\n"))

  # Condition for Date-Time columns
  if (inherits(col_data, "POSIXct")) {
    cat("- **Type:** Date-Time\n")
    cat(paste("- **Earliest Record:**", min(col_data, na.rm = TRUE), "\n"))
    cat(paste("- **Latest Record:**", max(col_data, na.rm = TRUE), "\n"))

  # Condition for numeric columns that are not IDs
  } else if (is.numeric(col_data) && !grepl("ID$", col_name) && n_distinct(col_data) > 1
    cat("- **Type:** Continuous Numeric\n")
    cat("- **Summary Stats:**\n")
    print(summary(col_data))

  # Condition for categorical columns (or numeric IDs)
  } else {
    cat("- **Type:** Categorical\n")
    cat("- **Frequency of Top 10 Values:**\n")
    freq_table <- taxi_data %>%
      count(.data[[col_name]], sort = TRUE) %>%
      mutate(percentage = round(n / sum(n) * 100, 2))

    # Formatting
    print(kable(head(freq_table, 10)))
  }

  cat("\n---\n")
}
```

Audit of `VendorID` - **Type:** Categorical - **Frequency of Top 10 Values:**

VendorID	n	percentage
2	2719860	78.26
1	753671	21.69
7	1206	0.03
6	489	0.01

Audit of `tpep_pickup_datetime` - **Type:** Date-Time - **Earliest Record:** 2024-12-31 21:47:55 - **Latest Record:** 2025-02-01 01:00:44

Audit of `tpep_dropoff_datetime` - **Type:** Date-Time - **Earliest Record:** 2024-12-18 08:52:40 - **Latest Record:** 2025-02-02 00:44:11

Audit of `passenger_count` - **Type:** Categorical - **Frequency of Top 10 Values:**

passenger_count	n	percentage
1	2322434	66.83
NA	540149	15.54
2	407761	11.73
3	91409	2.63
4	59009	1.70
0	24656	0.71
5	17786	0.51
6	12004	0.35
8	11	0.00
7	4	0.00

Audit of `trip_distance` - **Type:** Continuous Numeric - **Summary Stats:** Min. 1st Qu. Median Mean 3rd Qu. Max. 0.000e+00 9.800e-01 1.670e+00 5.855e+00 3.100e+00 2.764e+05

Audit of `RatecodeID` - **Type:** Categorical - **Frequency of Top 10 Values:**

RatecodeID	n	percentage
1	2756472	79.32
NA	540149	15.54
2	94420	2.72
99	41963	1.21
5	26501	0.76
3	8622	0.25
4	7092	0.20
6	7	0.00

Audit of `store_and_fwd_flag` - **Type:** Categorical - **Frequency of Top 10 Values:**

store_and_fwd_flag	n	percentage
N	2927431	84.24
NA	540149	15.54
Y	7646	0.22

Audit of `PULocationID` - **Type:** Categorical - **Frequency of Top 10 Values:**

PULocationID	n	percentage
161	169977	4.89
237	163703	4.71
236	155647	4.48
132	146137	4.21
230	125829	3.62
186	119131	3.43
162	117930	3.39
142	110585	3.18
239	96614	2.78
163	95906	2.76

Audit of `DOLocationID` - **Type:** Categorical - **Frequency of Top 10 Values:**

DOLocationID	n	percentage
236	161376	4.64
237	149970	4.32
161	131258	3.78
230	108177	3.11
170	100060	2.88
142	98982	2.85
239	97559	2.81
162	93798	2.70
141	92675	2.67
68	89232	2.57

Audit of `payment_type` - **Type:** Categorical - **Frequency of Top 10 Values:**

payment_type	n	percentage
1	2444393	70.34
0	540149	15.54
2	390429	11.23
4	76481	2.20
3	23773	0.68
5	1	0.00

Audit of `fare_amount` - **Type:** Continuous Numeric - **Summary Stats:** Min. 1st Qu. Median Mean 3rd Qu. Max. -900.00 8.60 12.11 17.08 19.50 863372.12

Audit of `extra` - **Type:** Continuous Numeric - **Summary Stats:** Min. 1st Qu. Median Mean 3rd Qu. Max. -7.500 0.000 0.000 1.318 2.500 15.000

Audit of `mta_tax` - **Type:** Categorical - **Frequency of Top 10 Values:**

mta_tax	n	percentage
0.50	3379839	97.26
-0.50	57140	1.64
0.00	38170	1.10
1.00	64	0.00
10.50	5	0.00
4.75	3	0.00
3.75	2	0.00
4.00	2	0.00
6.50	1	0.00

Audit of `tip_amount` - **Type:** Continuous Numeric - **Summary Stats:** Min. 1st Qu. Median Mean 3rd Qu. Max. -86.00 0.00 2.45 2.96 3.93 400.00

Audit of `tolls_amount` - **Type:** Continuous Numeric - **Summary Stats:** Min. 1st Qu. Median Mean 3rd Qu. Max. -126.9400 0.0000 0.0000 0.4493 0.0000 170.9400

Audit of `improvement_surcharge` - **Type:** Categorical - **Frequency of Top 10 Values:**

improvement_surcharge	n	percentage
1.0	3377509	97.19
-1.0	59530	1.71
0.0	37694	1.08
0.3	493	0.01

Audit of `total_amount` - **Type:** Continuous Numeric - **Summary Stats:** Min. 1st Qu. Median Mean 3rd Qu. Max. -901.00 15.20 19.95 25.61 27.78 863380.37

Audit of `congestion_surcharge` - **Type:** Categorical - **Frequency of Top 10 Values:**

congestion_surcharge	n	percentage
2.5	2660818	76.57
NA	540149	15.54

congestion_surcharge	n	percentage
0.0	225938	6.50
-2.5	48321	1.39

Audit of `Airport_fee` - **Type:** Categorical - **Frequency of Top 10 Values:**

Airport_fee	n	percentage
0.00	2706446	77.88
NA	540149	15.54
1.75	218203	6.28
-1.75	10411	0.30
1.25	8	0.00
5.00	7	0.00
0.75	1	0.00
6.75	1	0.00

Audit of `cbd_congestion_fee` - **Type:** Categorical - **Frequency of Top 10 Values:**

cbd_congestion_fee	n	percentage
0.75	2246495	64.64
0.00	1222178	35.17
-0.75	6553	0.19

Findings from Raw Data Inspection:

- **Invalid Records**: A large block of **540,149 trips have NA values** in `payment_type` and several other columns, making them invalid for analysis.
- **Special Cases & Errors**: Other columns contain logically impossible values (e.g., `passenger_count` of 0), special fare codes (`RatecodeID` = 99 for group rides), and negative values in fare columns that should be excluded from a standard fare model.

Step 3: Clean, Filter, and Transform Data

Based on our inspection, we now apply a comprehensive set of filters to create a high-quality dataset for modeling.

Based on our inspection, we now apply a comprehensive set of filters to create a high-quality dataset for modeling. The key cleaning steps include:

- Removing the 540,149 rows with `NA` values.
- Filtering to include only standard-rate trips (`RatecodeID` = 1).
- Ensuring `passenger_count` is greater than 0.
- Removing trips with a `trip_distance` of 0 or less.
- Filtering out any records with a `total_amount` less than a logical minimum (e.g., $2.50).

Step 4: Verify the Cleaned Data

```
Rows: 2,617,562
Columns: 21
$ VendorID              <int> 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 1, ~
$ tpep_pickup_datetime  <dttm> 2025-01-01 01:18:38, 2025-01-01 01:32:40, 2025-
~
$ tpep_dropoff_datetime <dttm> 2025-01-01 01:26:59, 2025-01-01 01:35:13, 2025-
~
$ passenger_count       <int> 1, 1, 1, 3, 3, 2, 1, 1, 1, 3, 1, 1, 3, 1, 1, 2, ~
$ trip_distance         <dbl> 1.60, 0.50, 0.60, 0.52, 0.66, 2.63, 1.71, 2.29, ~
$ store_and_fwd_flag    <chr> "N", "N", "N", "N", "N", "N", "N", "N", "N", "N"~
$ PULocationID          <int> 229, 236, 141, 244, 244, 239, 237, 237, 263, 236~
```

```
$ DOLocationID          <int> 237, 237, 141, 244, 116, 68, 262, 75, 236, 151, ~
$ payment_type          <fct> 1, 1, 1, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 1, 2, 1, ~
$ fare_amount           <dbl> 10.0, 5.1, 5.1, 7.2, 5.8, 19.1, 11.4, 11.4, 5.8,~
$ extra                 <dbl> 3.5, 3.5, 3.5, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0~
$ mta_tax               <dbl> 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5~
$ tip_amount            <dbl> 3.00, 2.02, 2.00, 0.00, 0.00, 0.00, 0.00, 0.00, ~
$ tolls_amount          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ improvement_surcharge <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ total_amount          <dbl> 18.00, 12.12, 12.10, 9.70, 8.30, 24.10, 16.40, 1~
$ congestion_surcharge  <dbl> 2.5, 2.5, 2.5, 0.0, 0.0, 2.5, 2.5, 2.5, 2.5, 2.5~
$ Airport_fee           <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ cbd_congestion_fee    <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ pickup_hour           <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ day_of_week           <fct> Wednesday, Wednesday, Wednesday, Wednesday, Wedn~
```

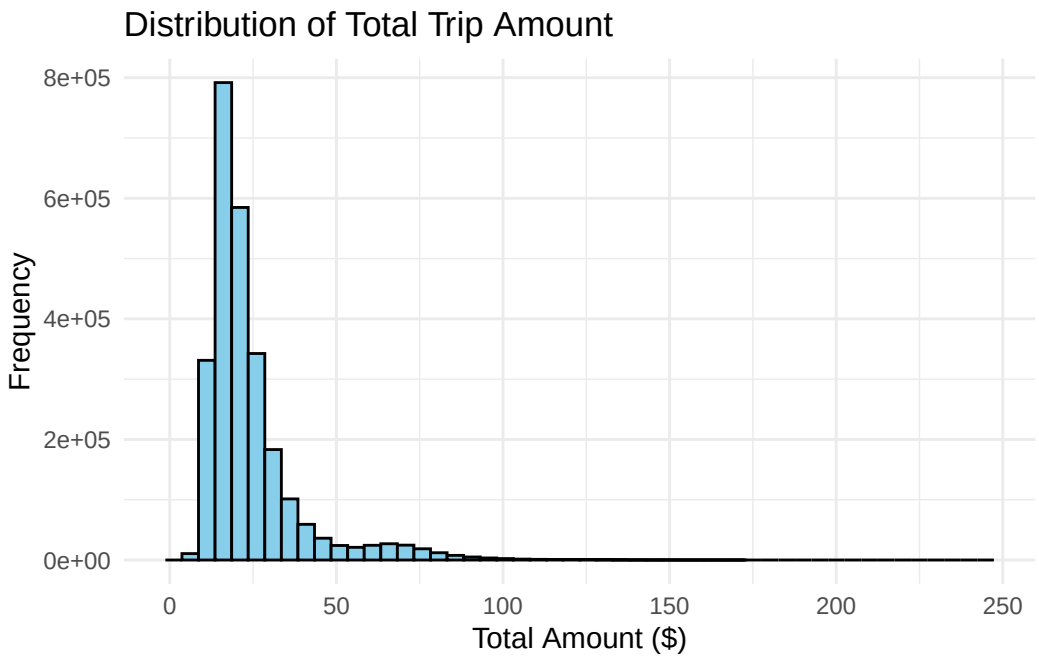
Verification: Our data is now significantly cleaner and focused on standard, meter-based trips. It is ready for reliable analysis.

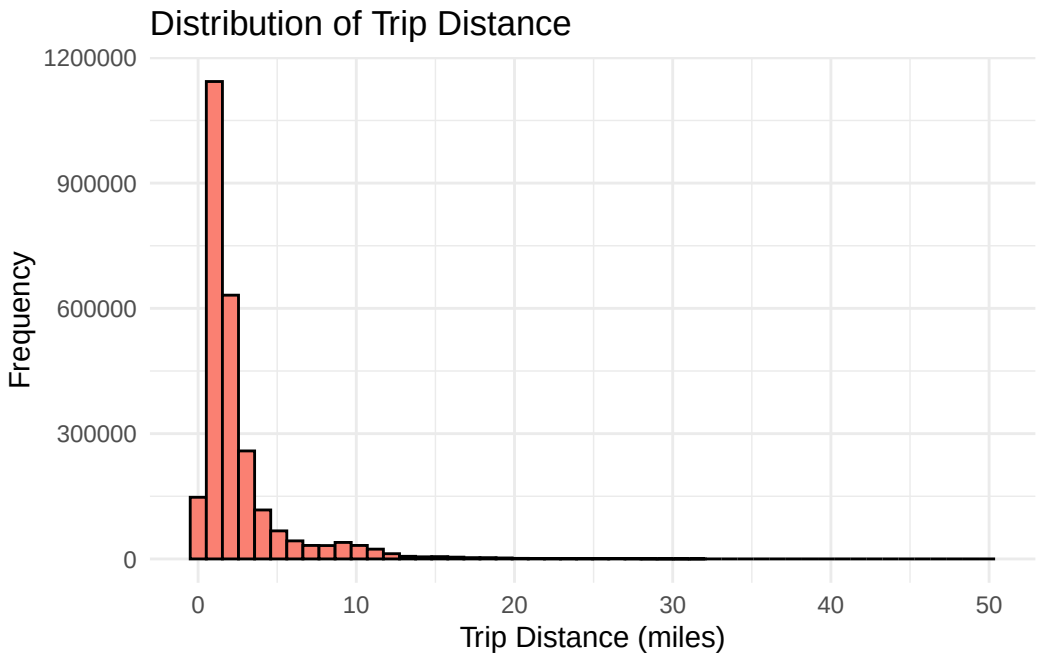
Exploratory Data Analysis (EDA)

EDA helps us understand the underlying patterns in the data before modeling.

Variable Distributions

First, we examine the distributions of our target variable, `total_amount`, and the primary predictor, `trip_distance`.

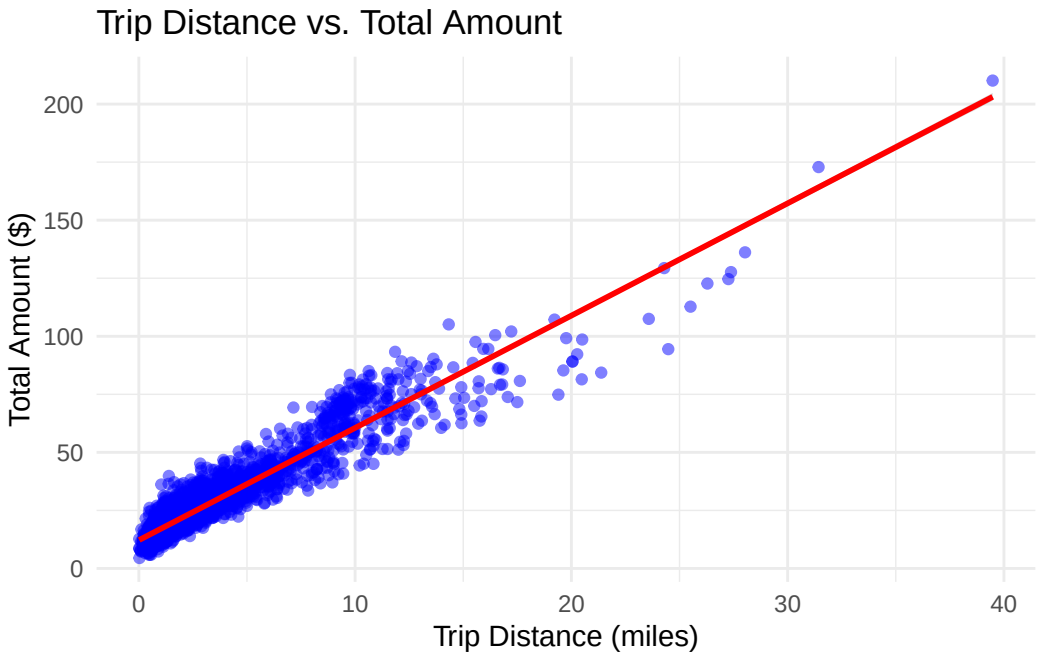


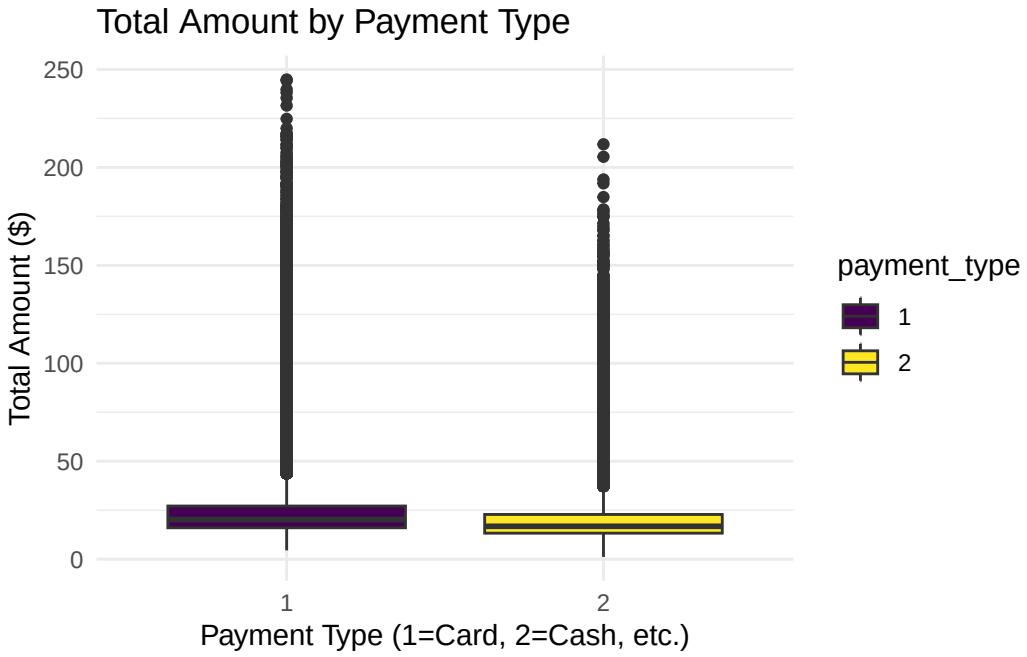


Finding: Both distributions are heavily **right-skewed**. Most trips are short and low-cost, with a long tail of more expensive, longer-distance trips.

Relationships Between Variables

Next, we visualize the relationships between key variables.

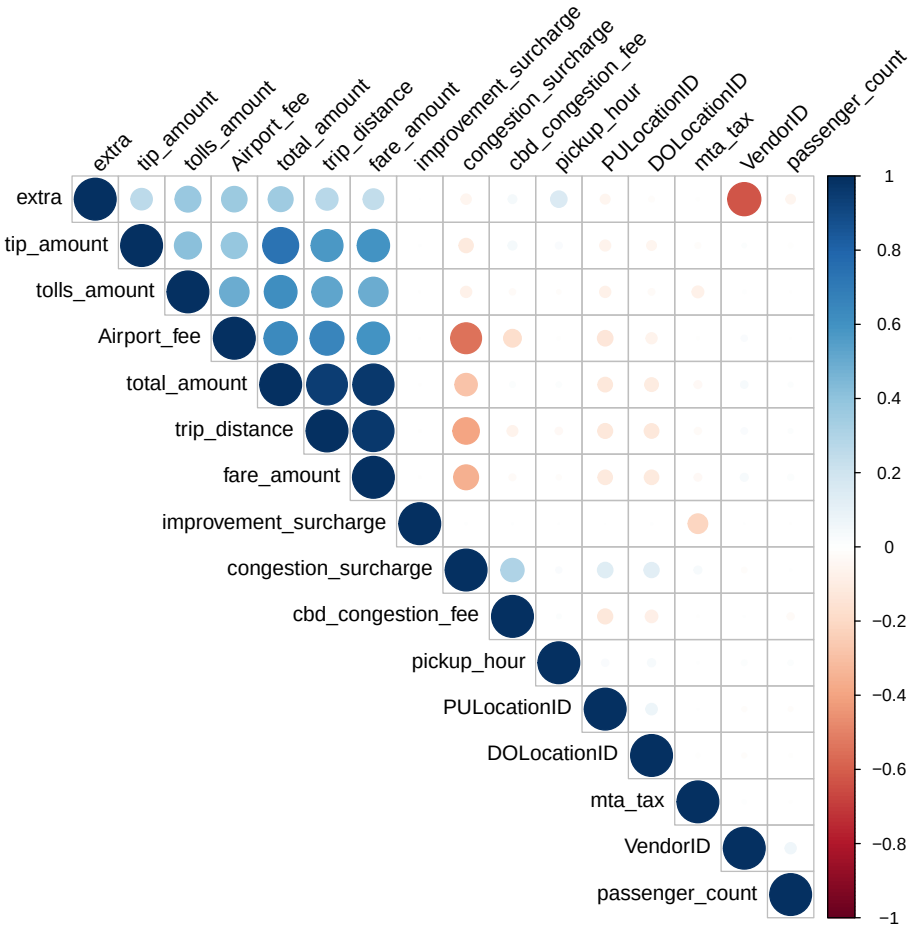




Finding: There is a strong, positive, linear relationship between `trip_distance` and `total_amount`. Additionally, trips paid by card (type 1) tend to have a higher median cost than those paid by cash (type 2).

Correlation Analysis

A correlation matrix provides a quantitative overview of the linear relationships between all numeric variables.



Finding: The matrix confirms a very strong positive correlation between `trip_distance`, `fare_amount`, and `total_amount`.

Linear Regression Modeling

Based on our EDA, we will now build regression models to predict `total_amount`.

Model 1: Simple Linear Regression

We start with a simple model using only `trip_distance` as the predictor.

```
Call:
lm(formula = total_amount ~ trip_distance, data = taxi_clean)

Residuals:
     Min       1Q   Median       3Q      Max
-195.560   -2.513   -0.503    1.998  220.896

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   12.132104   0.003898    3112   <2e-16 ***
trip_distance  4.837141   0.001022    4731   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.761 on 2617560 degrees of freedom
Multiple R-squared:  0.8953,    Adjusted R-squared:  0.8953
F-statistic: 2.239e+07 on 1 and 2617560 DF,  p-value: < 2.2e-16
```

Interpretation: The model estimates a base fare of approximately **$12.13**. For every additional mile of trip distance, the total amount is predicted to increase by **$4.84**. The **R-squared value of 0.895** indicates that trip distance alone explains about **89.5%** of the variability in the total fare.

Model 2: Multiple Linear Regression

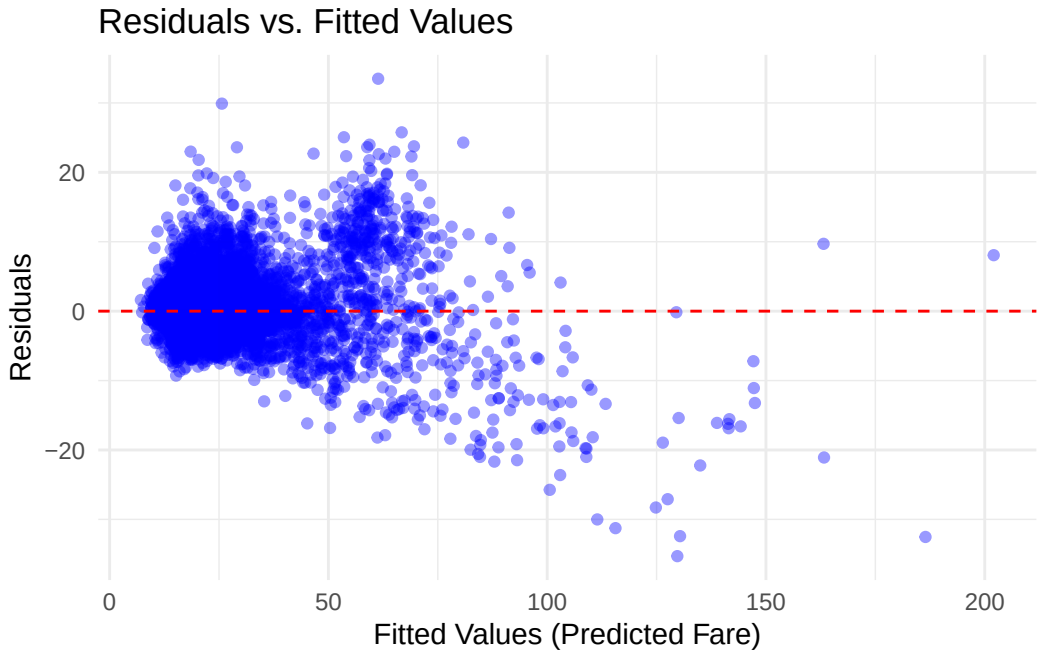
Next, we build a multiple regression model, adding `passenger_count`, `payment_type`, and `pickup_hour` to see if we can improve the prediction.

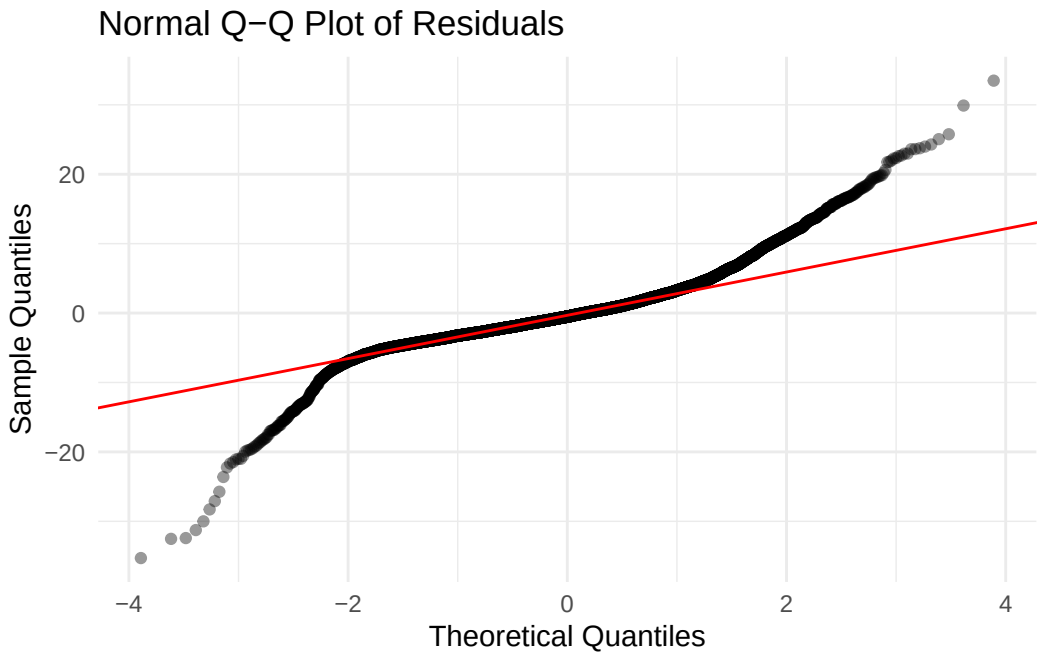
```
Call:
lm(formula = total_amount ~ trip_distance + passenger_count +
    payment_type + pickup_hour, data = taxi_clean)

Residuals:
     Min       1Q   Median       3Q      Max
-196.279   -2.398   -0.492    1.800  220.320

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     11.0741253  0.0091953  1204.3   <2e-16 ***
trip_distance    4.8491989  0.0009629  5035.8   <2e-16 ***
passenger_count  0.0884552  0.0037481    23.6   <2e-16 ***
payment_type2   -4.3405897  0.0081601  -531.9   <2e-16 ***
pickup_hour      0.1019232  0.0004662   218.6   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.482 on 2617557 degrees of freedom
Multiple R-squared:  0.9072,    Adjusted R-squared:  0.9072
F-statistic: 6.4e+06 on 4 and 2617557 DF,  p-value: < 2.2e-16
```

Interpretation: * The effect of `trip_distance` remains strong and consistent. * Holding other factors constant, choosing cash (`payment_type2`) is associated with a decrease in the total amount by approximately $4.34 compared to paying with a credit card (the baseline). This likely reflects that tips are often not recorded for cash payments. * Each additional `passenger_count` and later `pickup_hour` are also statistically significant predictors. * The **Adjusted R-squared increased to 0.907**, indicating that our new variables provide a small but significant improvement in the model's explanatory power.

Checking Model 2 Assumptions

Finally, we check the diagnostic plots for our final model to ensure the assumptions of linear regression are reasonably met.





Finding: The "Residuals vs Fitted" plot shows some deviation from a random scatter (a slight pattern), and the "Normal Q-Q" plot shows that the tails deviate from the line. This

suggests that while the model is powerful, it's not perfect, and non-linear relationships or transformations (like a log transform) could potentially improve it further.

Conclusion

In this analysis, we successfully identified key drivers of NYC taxi fares. **Trip distance** is overwhelmingly the most important predictor, explaining the vast majority of fare variation. We improved upon a simple model by incorporating additional features like payment method and time of day, creating a multiple regression model that explains **90.7%** of the variance in the total fare (based on the Adjusted R-squared). The exploratory analysis and modeling confirm strong, predictable patterns in the data, providing a solid foundation for fare prediction. Future work could explore interaction terms and variable transformations to capture more complex relationships in the data.

Project Summary and Reflection

Dataset Description

The dataset consists of trip records from NYC Yellow Taxis for January 2025. The goal of this analysis is to model the `total_amount` of a trip.

Original Dataset Columns

- `VendorID`: An identifier for the taxi service provider.
- `tpep_pickup_datetime`: The date and time when the passenger was picked up.
- `tpep_dropoff_datetime`: The date and time when the passenger was dropped off.
- `passenger_count`: The number of passengers in the vehicle.
- `trip_distance`: The total distance of the trip, measured in miles.
- `RatecodeID`: A code indicating the fare type applied to the trip (e.g., 1 for standard rate).

- `store_and_fwd_flag`: Indicates if the trip data was stored locally before server upload.
- `PULocationID`: The identifier for the NYC Taxi Zone where the trip began.
- `DOLocationID`: The identifier for the NYC Taxi Zone where the trip ended.
- `payment_type`: A numeric code for the payment method used.
- `fare_amount`: The base cost of the trip, calculated by time and distance.
- `extra`: Additional charges for factors like rush hour or overnight trips.
- `mta_tax`: A mandatory tax of $0.50 from the Metropolitan Transportation Authority (MTA).
- `tip_amount`: The amount of tip provided, typically recorded only for credit card payments.
- `tolls_amount`: The total cost of all tolls paid during the trip.
- `improvement_surcharge`: A mandatory surcharge for infrastructure improvements.
- `total_amount`: The **target variable**; the total amount paid by the passenger.
- `congestion_surcharge`: An additional fee for trips in high-traffic zones.
- `Airport_fee`: A fee for pickups or drop-offs at airports.

Engineered Features

- `payment_type (factor)`: The original numeric column was converted into a categorical factor to ensure the regression model treats it as distinct categories (e.g., Card, Cash) rather than a continuous number.
- `pickup_hour`: A numeric feature (0-23) extracted from `tpep_pickup_datetime`, representing the hour of the day the trip started.
- `day_of_week`: A categorical factor (e.g., "Monday") extracted from `tpep_pickup_datetime`, representing the day of the week the trip started.

Summary of Modeling Steps

The analysis began with a thorough data audit, followed by a cleaning process that filtered out over 500,000 invalid or outlier records. New features, such as `pickup_hour` and `day_of_week`, were engineered to capture temporal patterns. Exploratory Data Analysis (EDA) revealed a strong positive linear relationship between trip distance and total amount. Based on this, two linear regression models were built: a simple model with `trip_distance`

as the sole predictor, and a multiple regression model that also included `passenger_count`, `payment_type`, and `pickup_hour`.

Interpretation of Results

The final multiple regression model performed well, explaining approximately 90.7% of the variance in the total trip amount (`Adjusted R-squared = 0.9072`). `Trip_distance` was confirmed as the most significant predictor. The model also showed that cash payments are associated with a lower total amount, likely due to unrecorded tips, and that fares slightly increase with more passengers and later pickup hours. Diagnostic plots indicated that while the model is powerful, its residuals are not perfectly normally distributed, suggesting room for further refinement.

Challenges Encountered

A significant challenge was the sheer size of the dataset (over 3 million initial records), which made direct visualization and model diagnostics computationally intensive. To overcome this, I used sampling for creating scatter plots and diagnostic plots, which provided clear insights without crashing the R session. Another challenge was interpreting the model's limitations, particularly the violation of the normality assumption for residuals, which required careful consideration of potential next steps like data transformations.

Question for Peer Feedback

Given the clear right-skew in the target variable `total_amount` and the patterns observed in the "Residuals vs. Fitted" plot, would applying a log transformation to `total_amount` be the most justified next step to improve model fit and better satisfy the assumptions of linear regression?

Exam-Style Question and Answer

Question: Explain why the coefficient for `payment_type = 2` (Cash) is negative and statistically significant in the multiple regression model.

Answer: The negative coefficient for cash payments indicates that, holding trip distance and other factors constant, trips paid with cash are associated with a significantly lower `total_amount`. This is not because the fare itself is cheaper, but is most likely a data collection artifact. Credit card tips are electronically recorded and included in the `total_amount`, whereas cash tips are given directly to the driver and are often not entered into the meter system, thus remaining unrecorded in the dataset.