ETE3-1.R

Pranab Rai-2447137

2025-01-04

```
library(dplyr)
```

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

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

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

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.4.2
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(ggplot2)
```

```
 # Load your data
  df <-
read.csv("C:\\Users\\prana\\OneDrive\\Desktop\\2trimester\\R\\ETE3\\test2.csv
")
```

```
  # Convert pickup datetime
  df$tpep_pickup_datetime <- ymd_hms(df$tpep_pickup_datetime)
```

```
  # Basic Summary
  print("Basic Summary:")
```

```
## [1] "Basic Summary:"
```

```
  print(summary(df))
```

```
##  pickup_date         pickup_hour       VendorID
##  Length:720         Min.   : 0.00   Min.   :1.00
##  Class :character   1st Qu.: 5.75   1st Qu.:1.75
##  Mode  :character   Median :11.50   Median :2.00
##                     Mean   :11.50   Mean   :1.75
##                     3rd Qu.:17.25   3rd Qu.:2.00
##                     Max.   :23.00   Max.   :2.00
##  tpep_pickup_datetime             tpep_dropoff_datetime passenger_count
##  Min.   :2001-09-21 00:00:15.00   Length:720            Min.   :0.000
##  1st Qu.:2008-09-21 00:12:05.50   Class :character      1st Qu.:1.000
##  Median :2016-03-22 00:11:48.00   Mode  :character      Median :1.000
##  Mean   :2016-03-22 00:11:59.49                         Mean   :1.274
##  3rd Qu.:2023-09-21 00:12:09.50                         3rd Qu.:1.000
##  Max.   :2030-09-21 00:23:06.00                         Max.   :6.000
##  trip_distance      RatecodeID     store_and_fwd_flag  PULocationID
##  Min.   : 0.000   Min.   : 1.000   Length:720         Min.   :  4.0
##  1st Qu.: 1.070   1st Qu.: 1.000   Class :character   1st Qu.:132.0
##  Median : 1.890   Median : 1.000   Mode  :character   Median :161.0
##  Mean   : 3.616   Mean   : 2.224                      Mean   :164.7
##  3rd Qu.: 3.595   3rd Qu.: 1.000                      3rd Qu.:233.0
##  Max.   :22.550   Max.   :99.000                      Max.   :265.0
##   DOLocationID    payment_type    fare_amount          extra
##  Min.   :  1.0   Min.   :1.000   Min.   :-107.30   Min.   :-5.000
##  1st Qu.:106.8   1st Qu.:1.000   1st Qu.:   9.30   1st Qu.: 0.000
##  Median :161.0   Median :1.000   Median :  13.50   Median : 1.000
##  Mean   :156.7   Mean   :1.228   Mean   :  19.93   Mean   : 1.417
##  3rd Qu.:231.0   3rd Qu.:1.000   3rd Qu.:  23.30   3rd Qu.: 2.500
##  Max.   :265.0   Max.   :4.000   Max.   : 130.40   Max.   : 9.250
##     mta_tax          tip_amount      tolls_amount
improvement_surcharge
##  Min.   :-0.5000   Min.   : 0.000   Min.   :-13.3800   Min.   :-1.0000
##  1st Qu.: 0.5000   1st Qu.: 1.000   1st Qu.:  0.0000   1st Qu.: 1.0000
##  Median : 0.5000   Median : 2.860   Median :  0.0000   Median : 1.0000
##  Mean   : 0.4736   Mean   : 3.738   Mean   :  0.5872   Mean   : 0.9542
##  3rd Qu.: 0.5000   3rd Qu.: 4.400   3rd Qu.:  0.0000   3rd Qu.: 1.0000
##  Max.   : 0.5000   Max.   :40.050   Max.   : 21.3800   Max.   : 1.0000
##   total_amount     congestion_surcharge  Airport_fee      day_of_week
##  Min.   :-121.68   Min.   :-2.500       Min.   :-1.7500   Length:720
##  1st Qu.:  15.86   1st Qu.: 2.500       1st Qu.: 0.0000   Class :character
##  Median :  21.68   Median : 2.500       Median : 0.0000   Mode  :character
##  Mean   :  28.85   Mean   : 2.205       Mean   : 0.1434
##  3rd Qu.:  31.32   3rd Qu.: 2.500       3rd Qu.: 0.0000
##  Max.   : 175.30   Max.   : 2.500       Max.   : 1.7500
```

```r
  # More Detailed Descriptive Statistics


  # Numerical Columns
  numerical_cols <- c("trip_distance", "fare_amount", "total_amount",
"passenger_count") # Add more as needed
```

```r
  for (col in numerical_cols) {
    if (col %in% names(df)) {
      print(paste("n", col, "Statistics:"))
      print(paste("Mean", col, ":", mean(df[[col]], na.rm = TRUE)))
      print(paste("Median", col, ":", median(df[[col]], na.rm = TRUE)))
      print(paste("Standard Deviation of", col, ":", sd(df[[col]], na.rm =
TRUE)))
      print(paste("Range of", col, ":", paste(range(df[[col]], na.rm=TRUE),
collapse = " - ")))
      print(paste("Interquartile Range (IQR) of", col, ":", IQR(df[[col]],
na.rm = TRUE)))
      print("Quantiles")
      print(quantile(df[[col]], probs = c(0.05,0.25, 0.5, 0.75,0.95),
na.rm=TRUE))
    }
  }
```

```
## [1] "n trip_distance Statistics:"
## [1] "Mean trip_distance : 3.61588888888889"
## [1] "Median trip_distance : 1.89"
## [1] "Standard Deviation of trip_distance : 4.5533884784586"
## [1] "Range of trip_distance : 0 - 22.55"
## [1] "Interquartile Range (IQR) of trip_distance : 2.525"
## [1] "Quantiles"
##      5%     25%     50%     75%     95%
##  0.4895  1.0700  1.8900  3.5950 16.3160
## [1] "n fare_amount Statistics:"
## [1] "Mean fare_amount : 19.9270277777778"
## [1] "Median fare_amount : 13.5"
## [1] "Standard Deviation of fare_amount : 20.0215538600284"
## [1] "Range of fare_amount : -107.3 - 130.4"
## [1] "Interquartile Range (IQR) of fare_amount : 14"
## [1] "Quantiles"
##   5%  25%  50%  75%  95%
##  5.1  9.3 13.5 23.3 70.0
## [1] "n total_amount Statistics:"
## [1] "Mean total_amount : 28.8541388888889"
## [1] "Median total_amount : 21.675"
## [1] "Standard Deviation of total_amount : 25.4034276449892"
## [1] "Range of total_amount : -121.68 - 175.3"
## [1] "Interquartile Range (IQR) of total_amount : 15.465"
## [1] "Quantiles"
##      5%     25%     50%     75%     95%
## 11.2760 15.8600 21.6750 31.3250 88.8505
## [1] "n passenger_count Statistics:"
## [1] "Mean passenger_count : 1.27361111111111"
## [1] "Median passenger_count : 1"
## [1] "Standard Deviation of passenger_count : 0.718422986186287"
## [1] "Range of passenger_count : 0 - 6"
## [1] "Interquartile Range (IQR) of passenger_count : 0"
```

```
## [1] "Quantiles"
##  5% 25% 50% 75% 95%
##   1   1   1   1   3
```

```
  # Categorical Columns
  categorical_cols <- c("VendorID", "payment_type") # Add more as needed
  for (col in categorical_cols) {
    if (col %in% names(df)) {
      print(paste( col, "Frequencies:"))
      print(table(df[[col]]))
      print("Proportions")
      print(prop.table(table(df[[col]])))
    }
  }
```

```
## [1] "VendorID Frequencies:"
##
##   1   2
## 180 540
## [1] "Proportions"
##
##    1    2
## 0.25 0.75
## [1] "payment_type Frequencies:"
##
##   1   2   3   4
## 596 102   4  18
## [1] "Proportions"
##
##           1           2           3           4
## 0.827777778 0.141666667 0.005555556 0.025000000
```

```
  # Combined Statistics (Example: Average Fare Amount by Hour of Day)
  if ("fare_amount" %in% names(df) & "pickup_hour" %in% names(df)){
    print("Average Fare Amount by Hour of Day:")
    print(aggregate(fare_amount ~ pickup_hour, data = df, FUN = mean,
na.rm=TRUE))
  }
```

```
## [1] "Average Fare Amount by Hour of Day:"
##    pickup_hour fare_amount
## 1            0    21.95967
## 2            1    19.77333
## 3            2    17.68333
## 4            3    16.32333
## 5            4    18.11667
## 6            5    25.17000
## 7            6    21.54667
## 8            7    16.93333
## 9            8    14.92000
## 10           9    13.48667
```

```
## 11          10    18.75067
## 12          11    22.41000
## 13          12    19.37000
## 14          13    31.18667
## 15          14    14.43000
## 16          15    26.08000
## 17          16    19.86667
## 18          17    19.83333
## 19          18    22.47000
## 20          19    22.86000
## 21          20    21.67333
## 22          21    17.89667
## 23          22    15.25833
## 24          23    20.25000
```

```
  #Checking for missing values
  print("Missing Values per column")
```

```
## [1] "Missing Values per column"
```

```
  print(colSums(is.na(df)))
```

```
##           pickup_date           pickup_hour              VendorID
##                     0                     0                     0
##  tpep_pickup_datetime tpep_dropoff_datetime       passenger_count
##                     0                     0                     0
##         trip_distance            RatecodeID    store_and_fwd_flag
##                     0                     0                     0
##          PULocationID          DOLocationID          payment_type
##                     0                     0                     0
##           fare_amount                 extra               mta_tax
##                     0                     0                     0
##            tip_amount          tolls_amount improvement_surcharge
##                     0                     0                     0
##          total_amount  congestion_surcharge           Airport_fee
##                     0                     0                     0
##           day_of_week
##                     0
```

```
  #Data Type of each column
  print("Data Type of each column")
```

```
## [1] "Data Type of each column"
```

```
  print(sapply(df, class))
```

```
## $pickup_date
## [1] "character"
##
## $pickup_hour
## [1] "integer"
##
```

```
## $VendorID
## [1] "integer"
##
## $tpep_pickup_datetime
## [1] "POSIXct" "POSIXt"
##
## $tpep_dropoff_datetime
## [1] "character"
##
## $passenger_count
## [1] "integer"
##
## $trip_distance
## [1] "numeric"
##
## $RatecodeID
## [1] "integer"
##
## $store_and_fwd_flag
## [1] "character"
##
## $PULocationID
## [1] "integer"
##
## $DOLocationID
## [1] "integer"
##
## $payment_type
## [1] "integer"
##
## $fare_amount
## [1] "numeric"
##
## $extra
## [1] "numeric"
##
## $mta_tax
## [1] "numeric"
##
## $tip_amount
## [1] "numeric"
##
## $tolls_amount
## [1] "numeric"
##
## $improvement_surcharge
## [1] "integer"
##
## $total_amount
## [1] "numeric"
```

```
##
## $congestion_surcharge
## [1] "numeric"
##
## $Airport_fee
## [1] "numeric"
##
## $day_of_week
## [1] "character"
```

```
  # Number of rows and columns
  print(paste("Number of rows:", nrow(df)))
```

```
## [1] "Number of rows: 720"
```

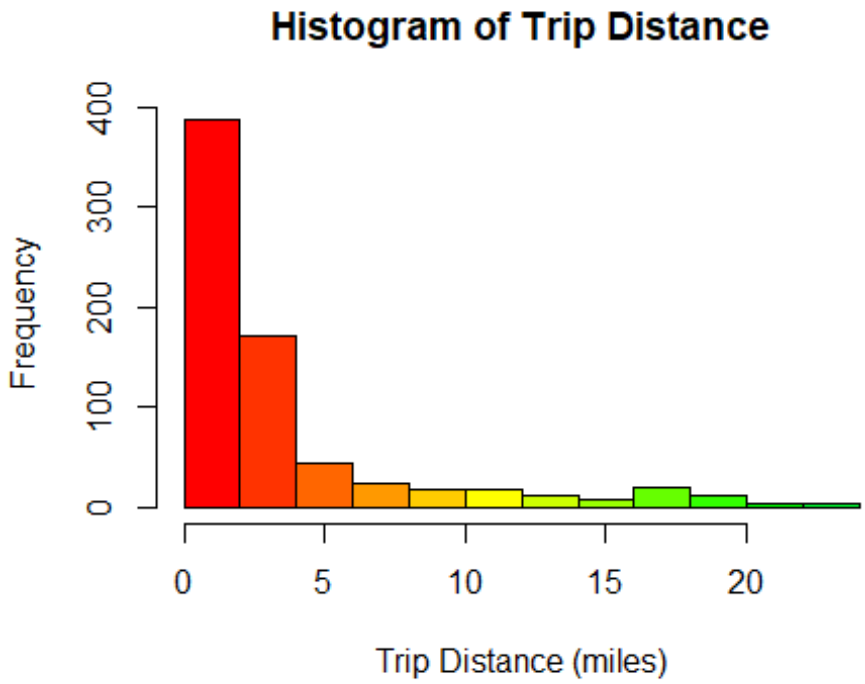
```
  print(paste("Number of columns:", ncol(df)))
```

```
## [1] "Number of columns: 22"
```

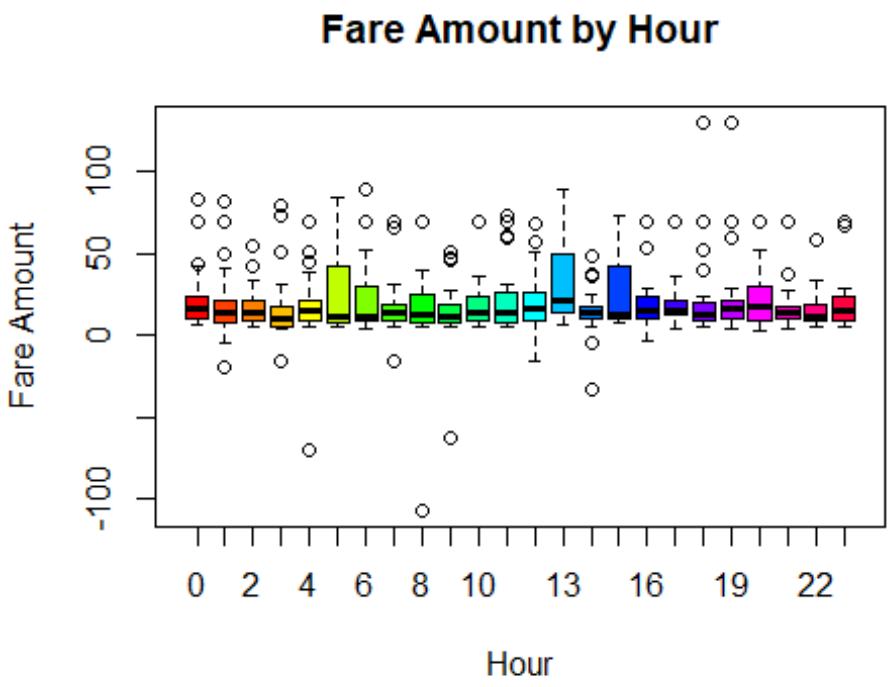
```
#########

  df <- df %>%
    mutate(
      pickup_date = as.Date(tpep_pickup_datetime, format = "%d-%m-%Y")
    )

  View(df)

  # --- Data Visualization ---
  print("Data Visualizations:")
```

```
## [1] "Data Visualizations:"
```

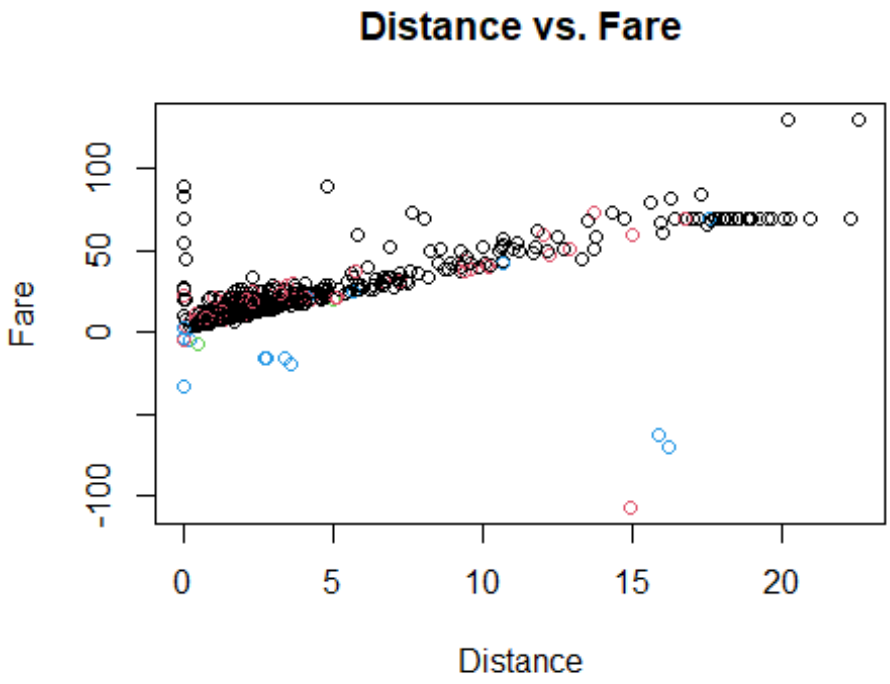
```
  # 1. Histogram of Trip Distance
  hist(df$trip_distance, main = "Histogram of Trip Distance", xlab = "Trip
Distance (miles)", na.rm = TRUE, col = rainbow(30))
```




```
  #The histogram of trip distances is heavily right-skewed, indicating that
most trips are short (less than 5 miles), with a long tail of less frequent,
longer trips extending beyond 15 miles.

  # 2. Boxplot of Fare Amount by Hour of Day
  boxplot(fare_amount ~ factor(pickup_hour), data = df, main = "Fare Amount
by Hour", xlab = "Hour", ylab = "Fare Amount", na.rm = TRUE, col =
rainbow(24))
```

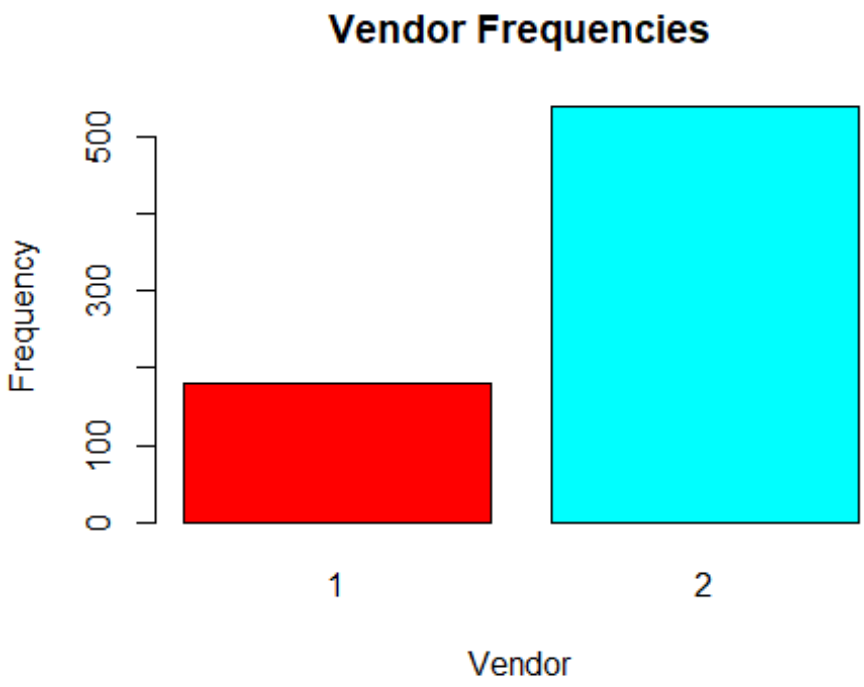



```
  #Fare amounts exhibit relatively consistent medians and interquartile
ranges across most hours, suggesting similar typical fare values throughout
the day, but there's increased variability and more frequent outliers during
certain periods, particularly around midday (hours 12-16) and some early
morning hours.

  # 3. Scatterplot of Trip Distance vs. Fare Amount with color by payment
type
  plot(df$trip_distance, df$fare_amount, main = "Distance vs. Fare", xlab =
"Distance", ylab = "Fare", col = factor(df$payment_type), na.rm = TRUE)
```



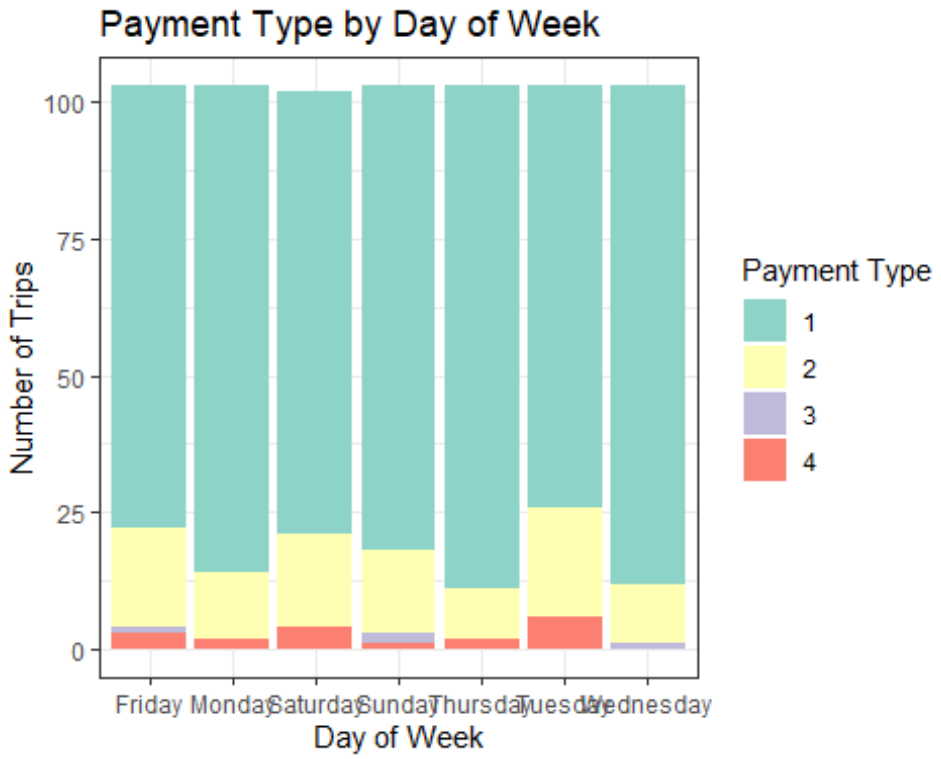

```
# fare generally increases with distance. Additionally, the presence of
outliers, especially at shorter distances, suggests other factors beyond
distance influence fare pricing.
```

```
  # 4. Bar Chart of Vendor ID
  barplot(table(df$VendorID), main = "Vendor Frequencies", xlab = "Vendor",
ylab = "Frequency", col = rainbow(nlevels(factor(df$VendorID))))
```

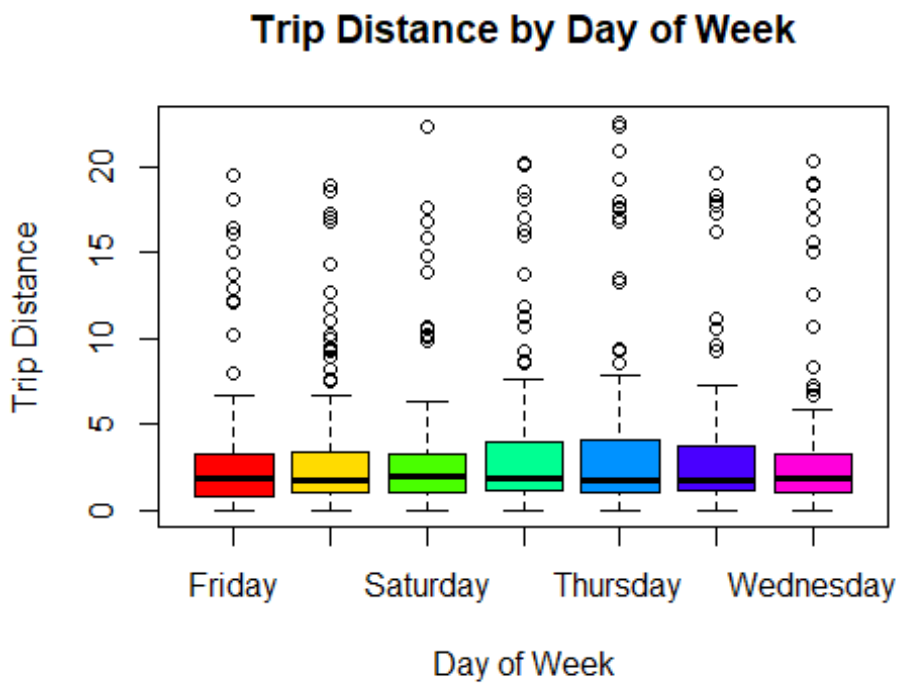



```
# Vendor 2 (VeriFone Inc.) has a substantially higher frequency of recorded trips than Vendor 1 (Creative Mobile Technologies, LLC), indicating that VeriFone is the more commonly used TPEP in this dataset.
```

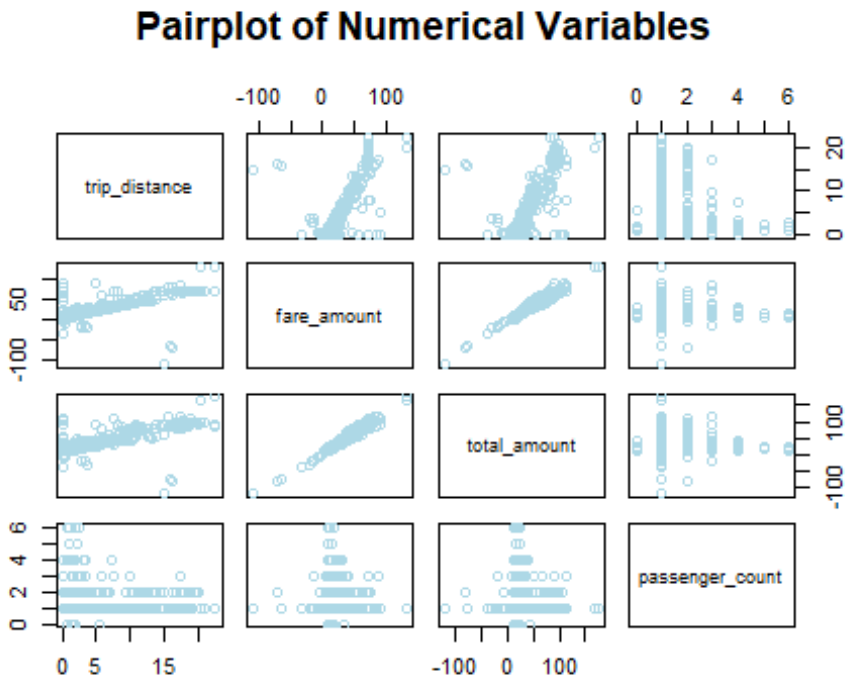
```
  # 5. Payment Type by Day of Week
  ggplot(df, aes(x = factor(day_of_week), fill = factor(payment_type))) +
    geom_bar(position = "stack") +
    labs(title = "Payment Type by Day of Week", x = "Day of Week", y = "Number of Trips", fill = "Payment Type") +
    theme_bw() +
    scale_fill_brewer(palette = "Set3")
```



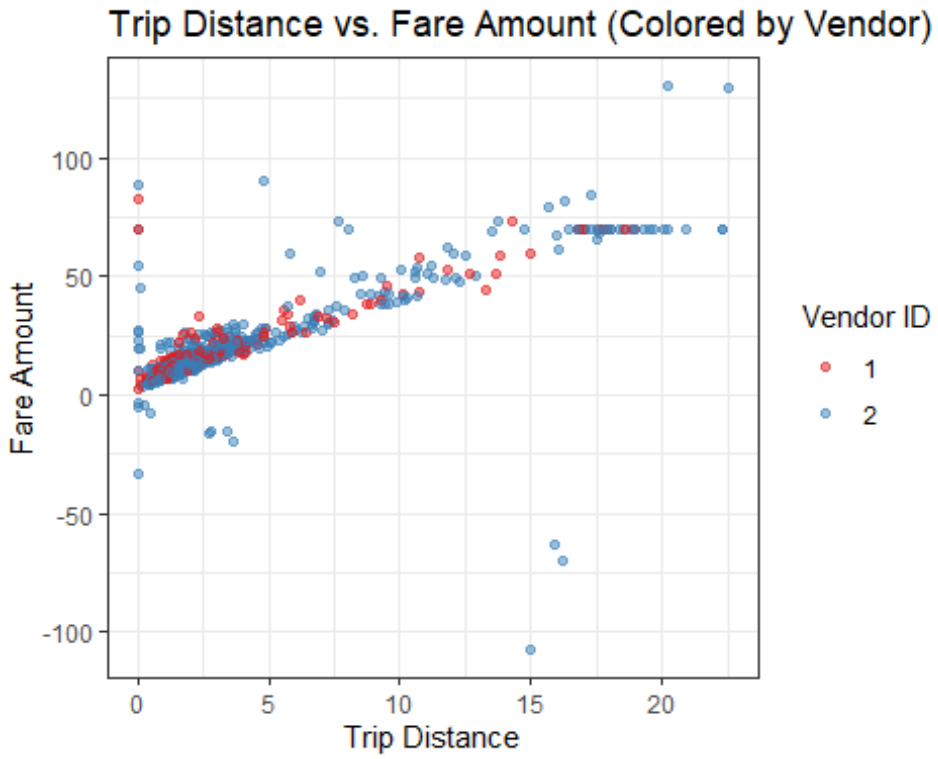

```
  #Credit card payments are the dominant payment method across all days of
the week, though cash payments show a slight increase on weekends (Friday and
Saturday), while "no charge" and "dispute" transactions remain consistently
low throughout the week.

  # 6. Boxplot of Trip Distance by Day of the Week
  boxplot(trip_distance ~ day_of_week, data = df, main = "Trip Distance by
Day of Week", xlab = "Day of Week", ylab = "Trip Distance", na.rm = TRUE, col
= rainbow(7))
```




```
  #Trip distances exhibit similar medians across all days of the week, but
weekends (Friday, Saturday, and Sunday) show increased variability and a
higher frequency of longer trips (outliers) compared to weekdays, suggesting
a wider range of trip lengths on weekends.

  # 7. Pair plot for numerical variables
  numerical_cols <- c("trip_distance", "fare_amount", "total_amount",
"passenger_count")
  numerical_data <- df[, numerical_cols[numerical_cols %in% names(df)]]
  if (ncol(numerical_data) >= 2) {
    pairs(numerical_data, main = "Pairplot of Numerical Variables", col =
"lightblue")
  }
```

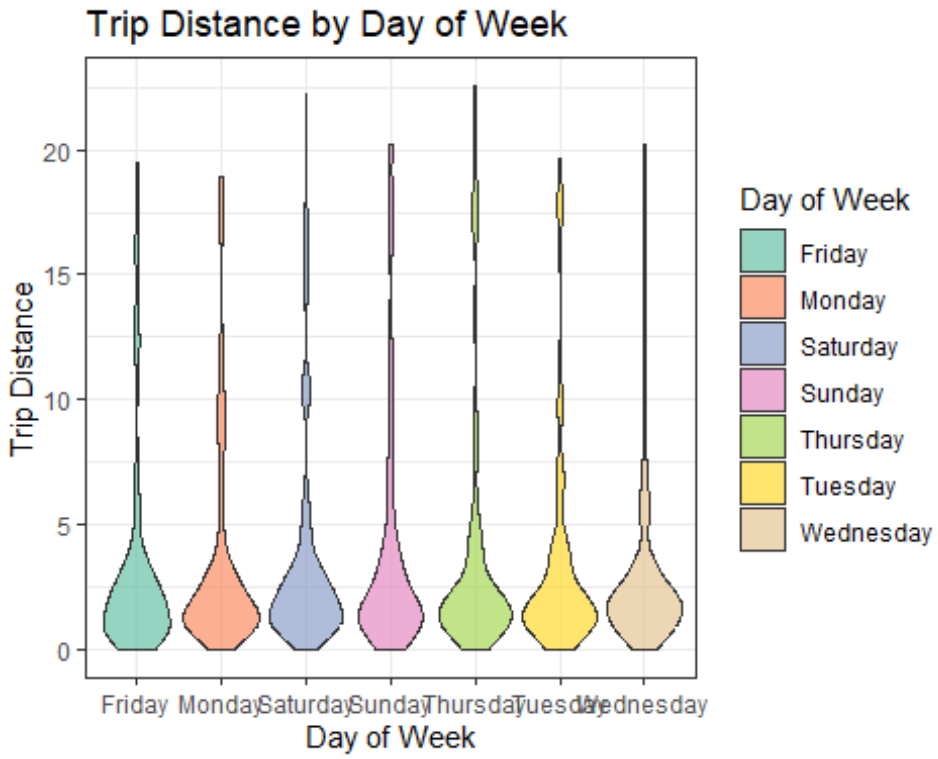



```
  #1. Trip Distance and Fare Amount: A strong positive correlation is
evident. As trip distance increases, fare amount generally increases as well.

  #2. Total Amount and Fare Amount: A very strong positive correlation
exists. This suggests that total_amount is highly influenced by fare_amount.

  #3. Passenger Count and Other Variables: Passenger count shows weaker
relationships with other variables. It is indicative that passenger count is
independent of other variables.

  # 8. Scatterplot of Trip Distance vs. Fare Amount, colored by Vendor
  ggplot(df, aes(x = trip_distance, y = fare_amount, color =
as.factor(VendorID))) +
    geom_point(alpha = 0.5) +
    labs(title = "Trip Distance vs. Fare Amount (Colored by Vendor)", x =
"Trip Distance", y = "Fare Amount", color = "Vendor ID") +
    theme_bw() +
    scale_color_brewer(palette = "Set1")
```

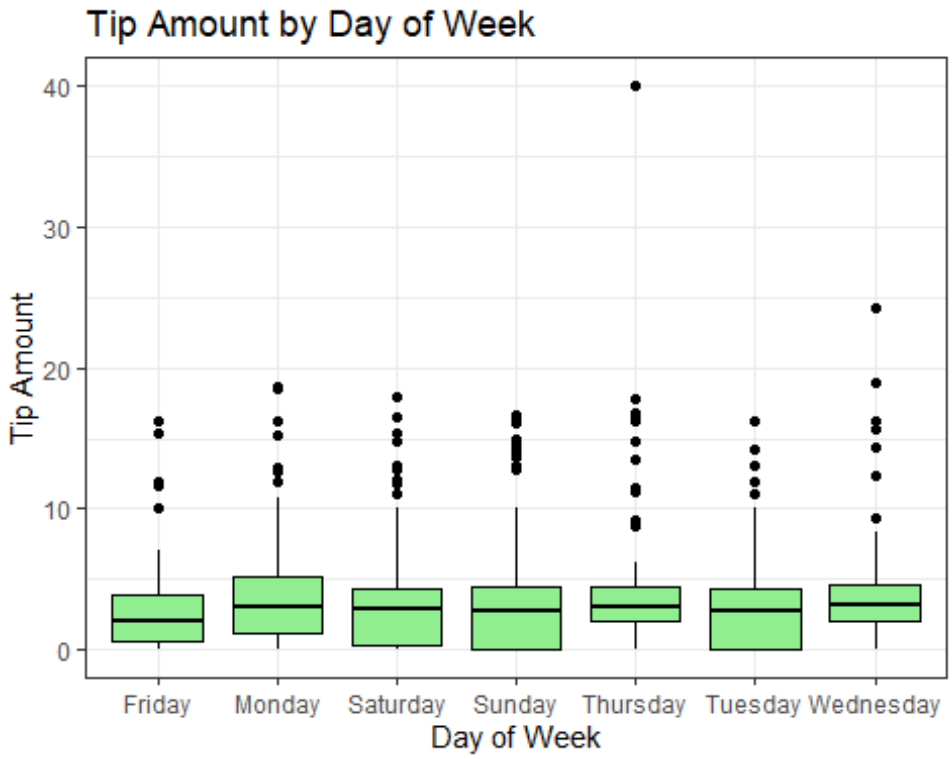



```
# The scatterplot shows positive correlation between distance and fare, meaning longer trips tend to have higher fares, but there's considerable variability, especially at shorter distances, and some notable outliers with unusually high or low fares for their respective distances.
```

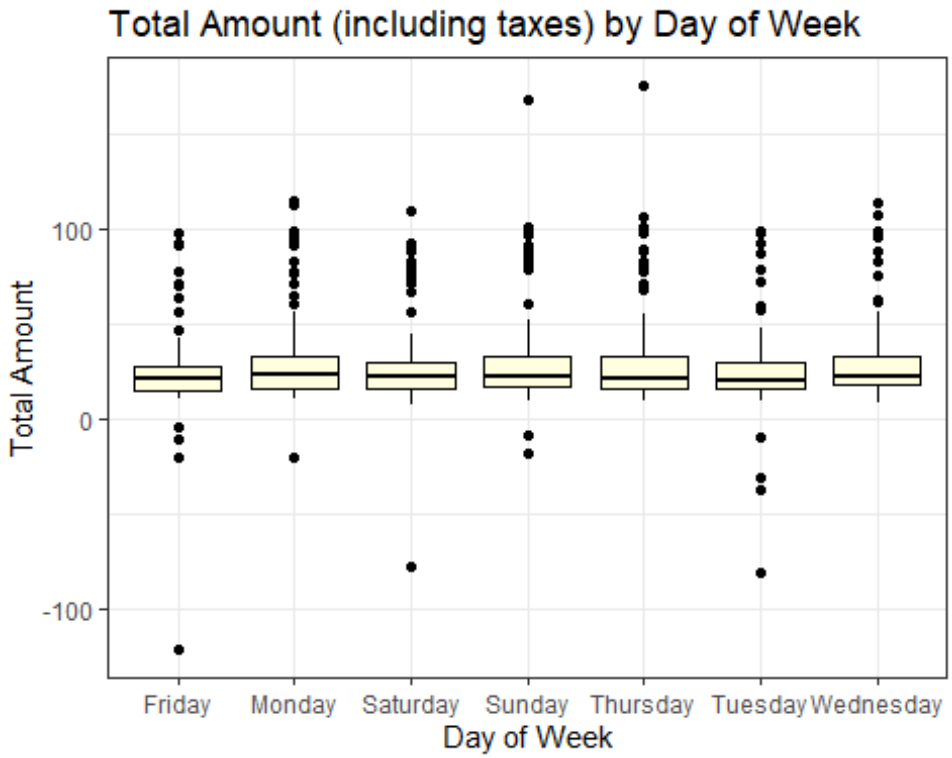
```
  # 9. Violin plot for Trip Distance by Day of the Week

 ggplot(df, aes(x = day_of_week, y = trip_distance, fill = day_of_week)) +
    geom_violin(alpha = 0.7) +
    labs(title = "Trip Distance by Day of Week", x = "Day of Week", y = "Trip
Distance", fill = "Day of Week") +
    theme_bw() +
    scale_fill_brewer(palette = "Set2")
```



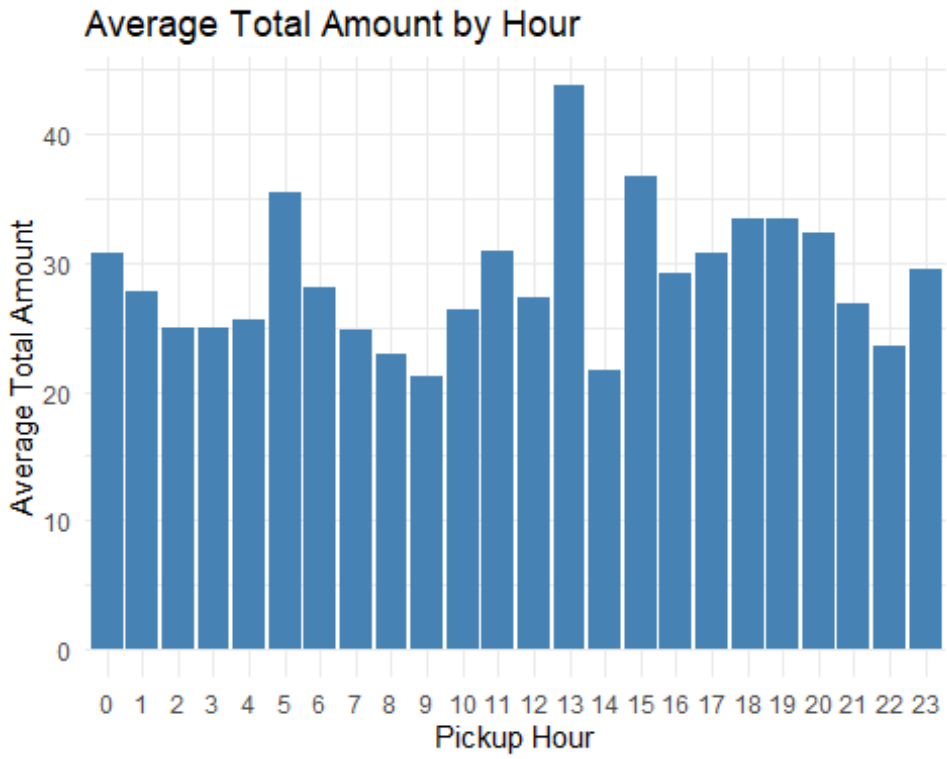

```
  #The shape and spread of the distributions vary slightly across the week.
  ##Weekends (Fri, Sat): The distributions are wider, suggesting more
variability in trip distances on weekends. There might be a larger proportion
of longer trips on these days.
  ##Weekdays (Mon-Thu): The distributions are more concentrated around the
median, indicating less variability in trip distances during weekdays. Trips
are likely to be shorter and more consistent in length.

  # 10. Boxplot of Tip Amount by Day of the Week
  ggplot(df, aes(x = day_of_week, y = tip_amount)) +
    geom_boxplot(fill = "lightgreen", color = "black") +
    labs(title = "Tip Amount by Day of Week", x = "Day of Week", y = "Tip
Amount") +
    theme_bw()
```



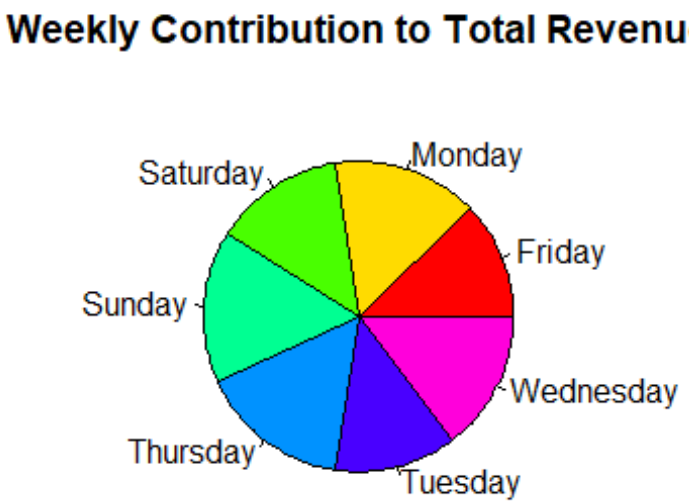

```
  #Tip amounts show consistent medians and interquartile ranges across all
days of the week, however, the presence of numerous outliers, especially on
weekends, suggests that significantly higher tips occur randomly throughout
the week, with a slightly higher probability on weekends.

  # 11. Boxplot of Total Amount (including taxes) by Day of the Week
  ggplot(df, aes(x = day_of_week, y = total_amount)) +
    geom_boxplot(fill = "lightyellow", color = "black") +
    labs(title = "Total Amount (including taxes) by Day of Week", x = "Day of
Week", y = "Total Amount") +
    theme_bw()
```



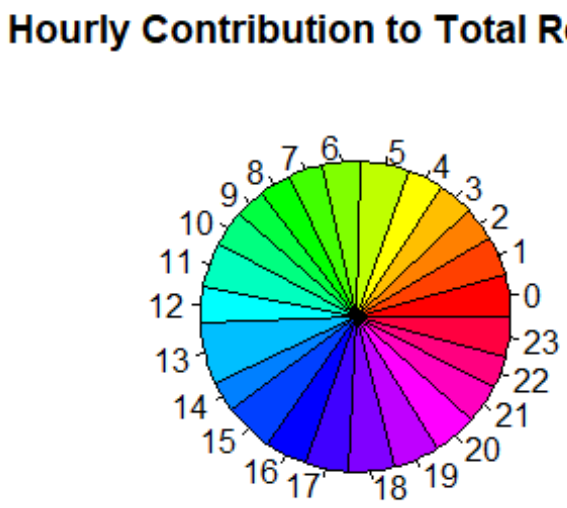

```
  #Total transaction amounts maintain a consistent median and interquartile
range across all days of the week, however, outliers, representing unusually
high or low transaction amounts, are present on all days, suggesting
consistent sporadic occurrences of atypical transactions.

  # 12. Bar Plot: Average Total Amount by Hour
  hourly_avg <- aggregate(total_amount ~ pickup_hour, data = df, mean)
  ggplot(hourly_avg, aes(x = factor(pickup_hour), y = total_amount)) +
    geom_bar(stat = "identity", fill = "steelblue") +
    labs(title = "Average Total Amount by Hour",
         x = "Pickup Hour",
         y = "Average Total Amount") +
    theme_minimal()
```



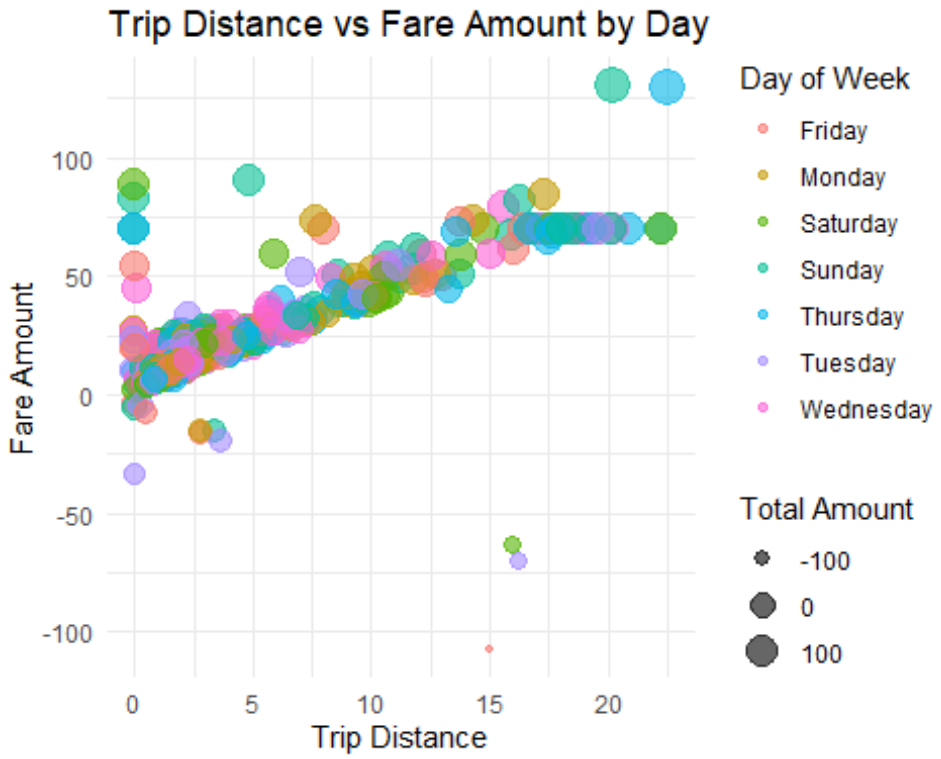

```
  #Average total transaction amounts fluctuate throughout the day, with
notable peaks around midday (hour 13) and late afternoon/early evening (hours
17-19), and troughs in the early morning hours (roughly 3-5) and mid-morning
(around 9-10), indicating variations in demand and/or trip characteristics
across different times of day.

  # 13. Pie Chart: Contribution of Each Day of the Week to Total Revenue
  daily_revenue <- aggregate(total_amount ~ day_of_week, data = df, sum)
  pie(daily_revenue$total_amount,
      labels = daily_revenue$day_of_week,
      col = rainbow(length(daily_revenue$day_of_week)),
      main = "Weekly Contribution to Total Revenue")
```



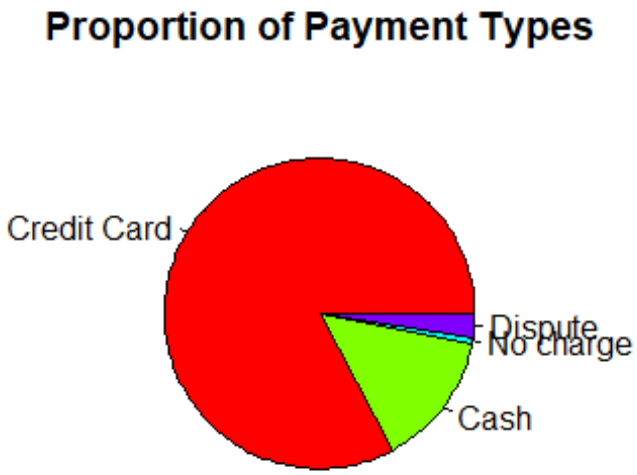

```
  #The pie chart shows the weekly contribution to total revenue, revealing
that revenue is distributed relatively evenly across the days of the week,
with perhaps slightly higher contributions from Friday and Saturday, and
slightly lower contributions from Sunday and Wednesday.

  # 14. Pie Chart: Contribution of Each Hour to Total Revenue
  hourly_revenue <- aggregate(total_amount ~ pickup_hour, data = df, sum)
  pie(hourly_revenue$total_amount,
      labels = paste0(hourly_revenue$pickup_hour),
      col = rainbow(length(hourly_revenue$pickup_hour)),
      main = "Hourly Contribution to Total Revenue")
```



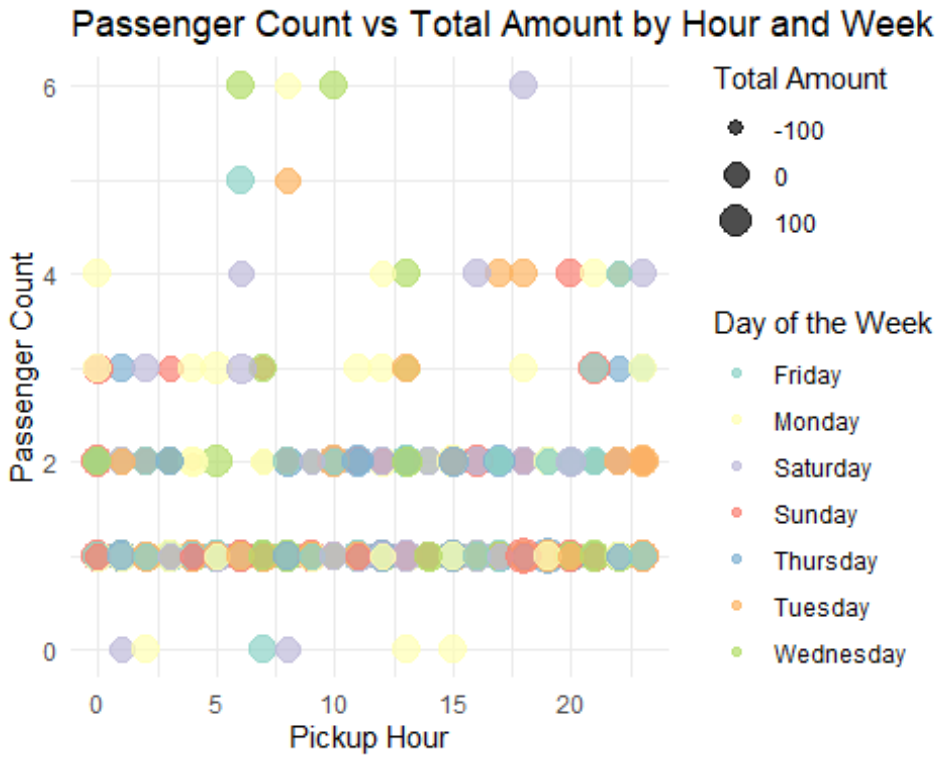

```
  #Revenue generation varies throughout the day, with peak contributions
observed in the afternoon/early evening hours (roughly 13:00 to 19:00, or 1
PM to 7 PM), and lower contributions during the night and early morning
hours.

  # 15. Bubble Chart: Trip Distance vs Fare Amount by Day
  ggplot(df, aes(x = trip_distance, y = fare_amount, size = total_amount,
color = day_of_week)) +
    geom_point(alpha = 0.6) +
    labs(title = "Trip Distance vs Fare Amount by Day",
         x = "Trip Distance",
         y = "Fare Amount",
         size = "Total Amount",
         color = "Day of Week") +
    theme_minimal()
```



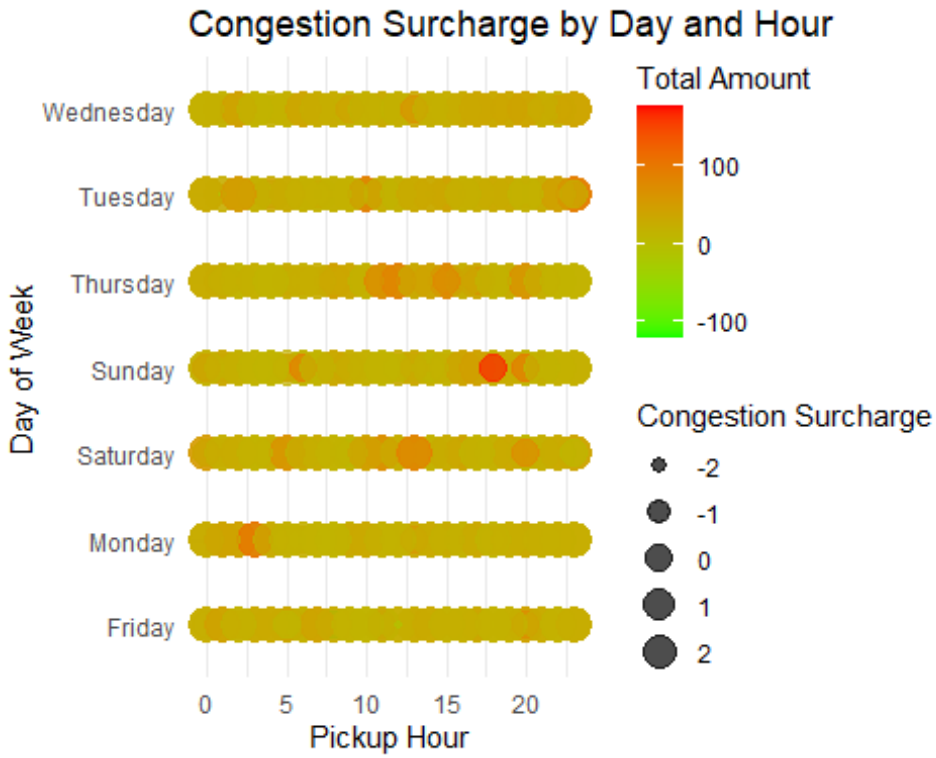

```
  #Positive Correlation: The overall positive relationship between trip
distance and fare amount. Longer trips generally cost more.
  #Day-of-Week Variation: There's no strong visual evidence suggesting that
specific days of the week consistently have higher or lower fares for a given
distance.
  #Total Amount Influence: Higher total amounts are associated with longer
trips (larger bubbles tend to appear towards the right of the plot). This
aligns with the expectation that longer trips result in higher fares and thus
higher total amounts.

#Outliers: There are some outliers present, particularly with unusually low
fares for given distances or unusual total amounts.


  # 16. Pie Chart: Proportion of Payment Types
  pie(table(df$payment_type),
      labels = c("Credit Card", "Cash", "No charge","Dispute"),
      col = rainbow(length(table(df$payment_type))),
      main = "Proportion of Payment Types")
```




```
  #Credit card is the overwhelmingly dominant method, representing a large
majority of transactions. Cash is the next most common, but its proportion is
significantly smaller. Payment types no charge and dispute represent only
very small fractions of the total transactions.

  # 17. Bubble Chart: Passenger Count vs Total Amount by Hour and Week
  ggplot(df, aes(x = pickup_hour, y = passenger_count, size = total_amount,
color = day_of_week)) +
    geom_point(alpha = 0.7) +
    labs(
      title = "Passenger Count vs Total Amount by Hour and Week",
      x = "Pickup Hour",
      y = "Passenger Count",
      size = "Total Amount",
      color = "Day of the Week"
    ) +
    scale_color_brewer(palette = "Set3") +
    theme_minimal()
```




```
  #Passenger Count and Total Amount: There's no clear linear relationship
between passenger count and total amount. While larger bubbles (higher total
amounts) appear across different passenger counts, the bubble size doesn't
consistently increase with passenger count. This suggests other factors (like
distance) have a greater influence on the total amount.
  #Pickup Hour and Total Amount: Higher total amounts (larger bubbles) appear
to be somewhat more common during certain hours, particularly in the
afternoon/evening (roughly 12-20).
  #Day of the Week: The different colors representing days of the week holds
no strong day-of-week effect on the relationship between passenger count and
total amount within each hour.

  # 18. Bubble Chart: Congestion Surcharge by Day and Hour
  ggplot(df, aes(x = pickup_hour, y = day_of_week, size =
congestion_surcharge, color = total_amount)) +
    geom_point(alpha = 0.7) +
    labs(
      title = "Congestion Surcharge by Day and Hour",
      x = "Pickup Hour",
      y = "Day of Week",
      size = "Congestion Surcharge",
      color = "Total Amount"
    ) +
    scale_color_gradient(low = "green", high = "red") +
    theme_minimal()
```



#Congestion Surcharge Pattern: Congestion surcharges are generally small or non-existent (most bubbles are small). Larger surcharges (larger bubbles) appear sporadically, with some concentration during typical commuting hours (7-9 AM and 4-6 PM) on weekdays. This aligns with the expected pattern of congestion pricing being applied during peak traffic times.

#Total Amount and Surcharge: There's no clear, direct correlation between total amount and congestion surcharge. The surcharge is applied independently of the total fare amount, based primarily on time and day.

#Weekend Surcharges: Surcharges are less frequent and less pronounced on weekends (Saturday and Sunday), consistent with reduced traffic congestion during those times.

#####

#Interpretation after Data Visualisation:

#Most trips are relatively short, as evidenced by the right-skewed trip distance distribution. Trip distances show greater variability on weekends, with more long trips occurring.

#Fare amounts are strongly positively correlated with trip distance, but other factors also influence pricing, as evidenced by outliers. While fare amounts are generally consistent throughout the day, variability increases around midday and early morning.

#Credit cards are the dominant payment method, with a slight increase in cash usage on weekends. "No charge" and "dispute" transactions are

```
infrequent.
  #Peak demand and revenue occur during the afternoon/early evening hours (1
PM to 7 PM). While total revenue is distributed relatively evenly across the
week, average total amounts are slightly higher mid-week. Congestion
surcharges are primarily applied during weekday commuting hours.
  #Passenger count does not appear to be a major driver of total amount.
  #There are minor differences in fare amounts between vendors, especially
for longer distances.
```