FA3_Group3_Quijano_DSC1105

Julian Philip S. Quijano

2026-02-22

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.4.3
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
library(broom)
```

```
## Warning: package 'broom' was built under R version 4.4.3
```

```
library(modelr)
```

```
## Warning: package 'modelr' was built under R version 4.4.3
```

```
##
## Attaching package: 'modelr'
```

```
## The following object is masked from 'package:broom':
##
##     bootstrap
```

```
library(robustbase)
```

```
## Warning: package 'robustbase' was built under R version 4.4.3
```

```
library(splines)
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.4.3
```

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

```
## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

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

```
# PART 1: DATA PREPARATION

dt <- fread("2023_Yellow_Taxi_Trip_Data.csv")

dt[, tpep_pickup_datetime := as.POSIXct(tpep_pickup_datetime,
                                        format="%m/%d/%Y %I:%M:%S %p")]

dt[, tpep_dropoff_datetime := as.POSIXct(tpep_dropoff_datetime,
                                         format="%m/%d/%Y %I:%M:%S %p")]

dt <- dt[trip_distance > 0 & total_amount > 0]

dt[, trip_duration_minutes := as.numeric(
  difftime(tpep_dropoff_datetime,
           tpep_pickup_datetime,
           units="mins"))]

dt[, fare_per_mile := total_amount / trip_distance]

dt[, month := factor(month(tpep_pickup_datetime), levels=1:12)]
dt[, quarter := factor(quarter(tpep_pickup_datetime))]

model_dt <- dt[, .(trip_distance, total_amount, month, quarter)]

if (nrow(model_dt) > 5000000) {
  model_dt <- model_dt[sample(.N, 5000000)]
}

plot_dt <- model_dt[sample(.N, min(200000, .N))]

cat("SUMMARY STATISTICS\n")
```

```
## SUMMARY STATISTICS
```

```
print(summary(model_dt[, .(trip_distance, total_amount)]))
```

```
##  trip_distance        total_amount
##  Min.   :     0.01   Min.   :  0.01
##  1st Qu.:     1.10   1st Qu.: 16.00
##  Median :     1.81   Median : 21.00
##  Mean   :     4.13   Mean   : 28.92
##  3rd Qu.:     3.49   3rd Qu.: 30.90
##  Max.   :258928.15   Max.   :944.30
```

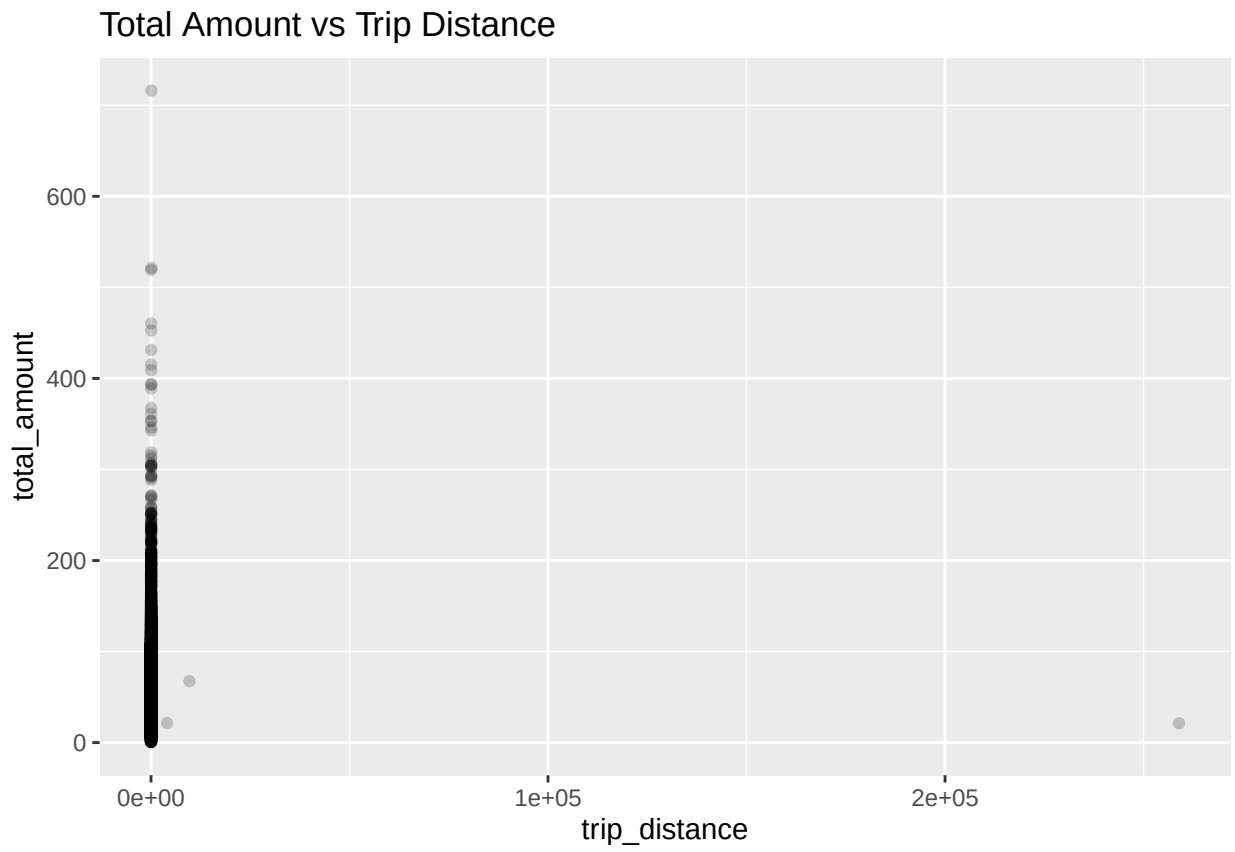
```r
cat("\nCORRELATION\n")
```

```
##
## CORRELATION
```

```r
print(cor(model_dt$trip_distance,
          model_dt$total_amount,
          use="complete.obs"))
```

```
## [1] 0.0195637
```

```r
ggplot(plot_dt, aes(trip_distance, total_amount)) +
  geom_point(alpha=0.2) +
  labs(title="Total Amount vs Trip Distance")
```




```r
# PART 2: LINEAR MODEL

lm_model <- lm(total_amount ~ trip_distance, data=model_dt)

cat("\nLINEAR MODEL COEFFICIENTS\n")
```

```
##
## LINEAR MODEL COEFFICIENTS
```

```r
print(coef(lm_model))
```

```
##   (Intercept) trip_distance
##  28.908976373   0.001926643
```

```r
cat("\nR-SQUARED\n")
```

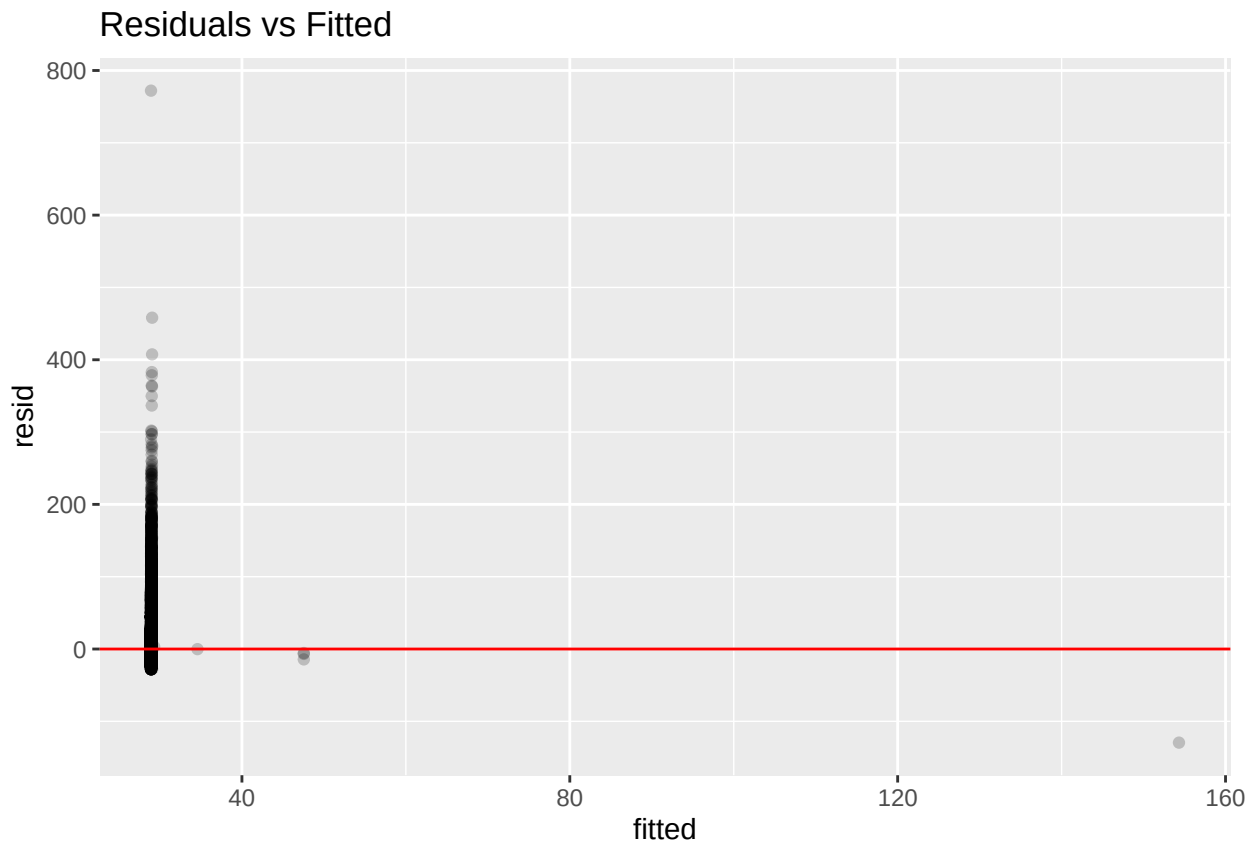
```
##
## R-SQUARED
```

```r
print(summary(lm_model)$r.squared)
```

```
## [1] 0.0003827384
```

```r
plot_dt$resid <- resid(lm_model)[1:nrow(plot_dt)]
plot_dt$fitted <- fitted(lm_model)[1:nrow(plot_dt)]

ggplot(plot_dt, aes(fitted, resid)) +
  geom_point(alpha=0.2) +
  geom_hline(yintercept=0, color="red") +
  labs(title="Residuals vs Fitted")
```



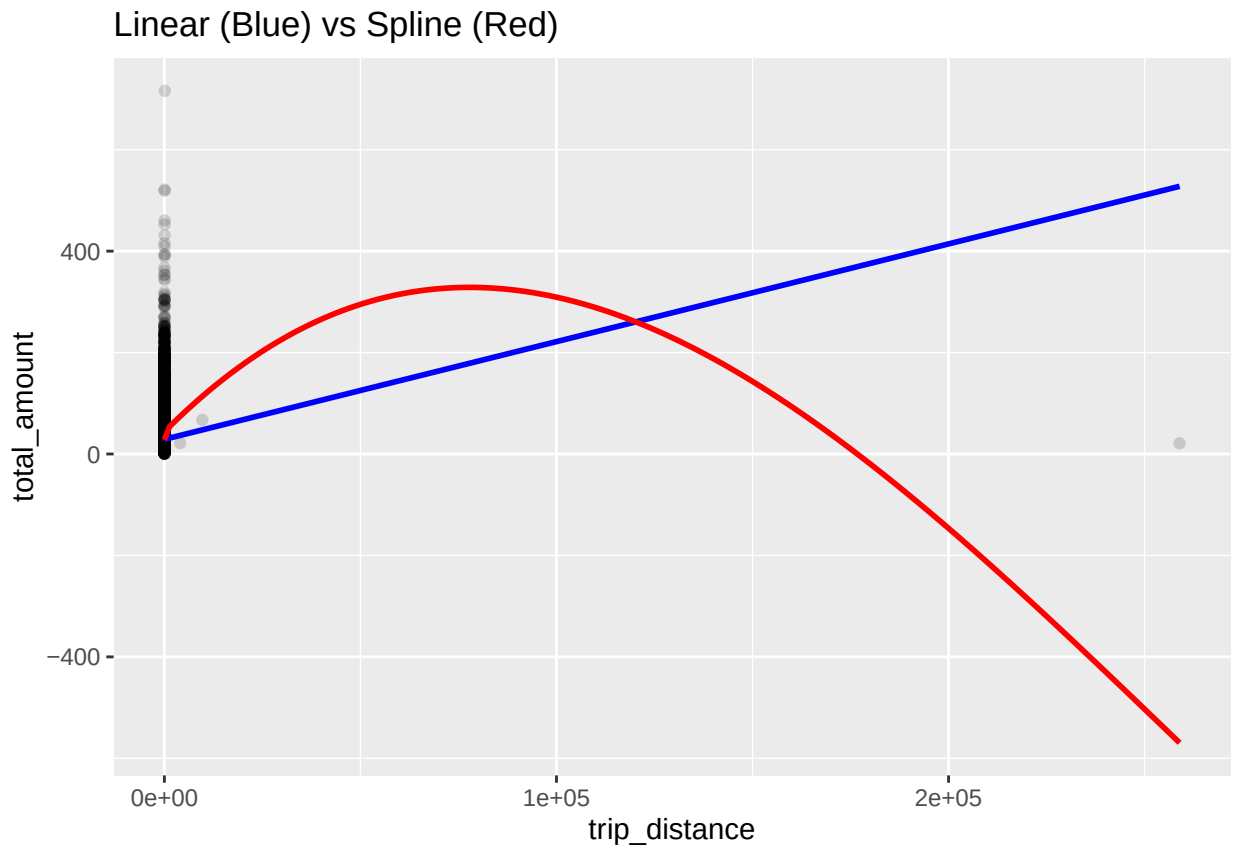
```r
# PART 3: SPLINE MODEL

spline_model <- lm(total_amount ~ ns(trip_distance, df=4),
                   data=model_dt)

grid <- data.frame(
  trip_distance = seq(min(plot_dt$trip_distance),
                      max(plot_dt$trip_distance),
                      length.out=200)
)

grid$linear_pred <- predict(lm_model, grid)
grid$spline_pred <- predict(spline_model, grid)

ggplot(plot_dt, aes(trip_distance, total_amount)) +
  geom_point(alpha=0.15) +
  geom_line(data=grid, aes(y=linear_pred),
            color="blue", linewidth=1) +
  geom_line(data=grid, aes(y=spline_pred),
            color="red", linewidth=1) +
  labs(title="Linear (Blue) vs Spline (Red)")
```




```r
cat("\nRMSE COMPARISON\n")
```

```
##
```

```
## RMSE COMPARISON
```

```
print(rmse(lm_model, model_dt))
```

```
## [1] 22.63578
```

```
print(rmse(spline_model, model_dt))
```

```
## [1] 18.20876
```

```
# PART 4: ROBUST REGRESSION

cooks <- cooks.distance(lm_model)
top10_index <- order(cooks, decreasing=TRUE)[1:10]
top10_trips <- model_dt[top10_index]

cat("\nTOP 10 INFLUENTIAL TRIPS\n")
```

```
##
## TOP 10 INFLUENTIAL TRIPS
```

```
print(top10_trips)
```

```
##     trip_distance total_amount  month quarter
##             <num>        <num> <fctr>  <fctr>
##  1:     258928.15        21.18      1       1
##  2:     166984.96        32.16      4       2
##  3:     157733.41        28.29      3       1
##  4:     145010.42        47.59      6       2
##  5:     129871.82        40.01      9       3
##  6:     143926.36       143.69      2       1
##  7:     110349.64        30.61      5       2
##  8:      96764.99        29.65      2       1
##  9:      97923.85        44.80      8       3
## 10:      70457.90        19.49      3       1
```

```
robust_model <- lmrob(total_amount ~ trip_distance,
                      data=model_dt)

cat("\nCOEFFICIENT COMPARISON\n")
```

```
##
## COEFFICIENT COMPARISON
```

```
print(coef(lm_model))
```

```
##   (Intercept) trip_distance
##  28.908976373   0.001926643
```

```r
print(coef(robust_model))
```

```
##   (Intercept) trip_distance
##     11.472484      4.782557
```

```r
high_dist <- data.frame(trip_distance=c(20,40,60))

cat("\nHIGH DISTANCE PREDICTIONS\n")
```

```
##
## HIGH DISTANCE PREDICTIONS
```

```r
print(predict(lm_model, high_dist))
```

```
##        1        2        3
## 28.94751 28.98604 29.02457
```

```r
print(predict(robust_model, high_dist))
```

```
##        1        2        3
## 107.1236 202.7748 298.4259
```

```r
# PART 5: SEASONAL MODELS

quarter_slopes <- model_dt[, .(
  slope = coef(lm(total_amount ~ trip_distance))[2]
), by=quarter]

cat("\nQUARTERLY SLOPES\n")
```

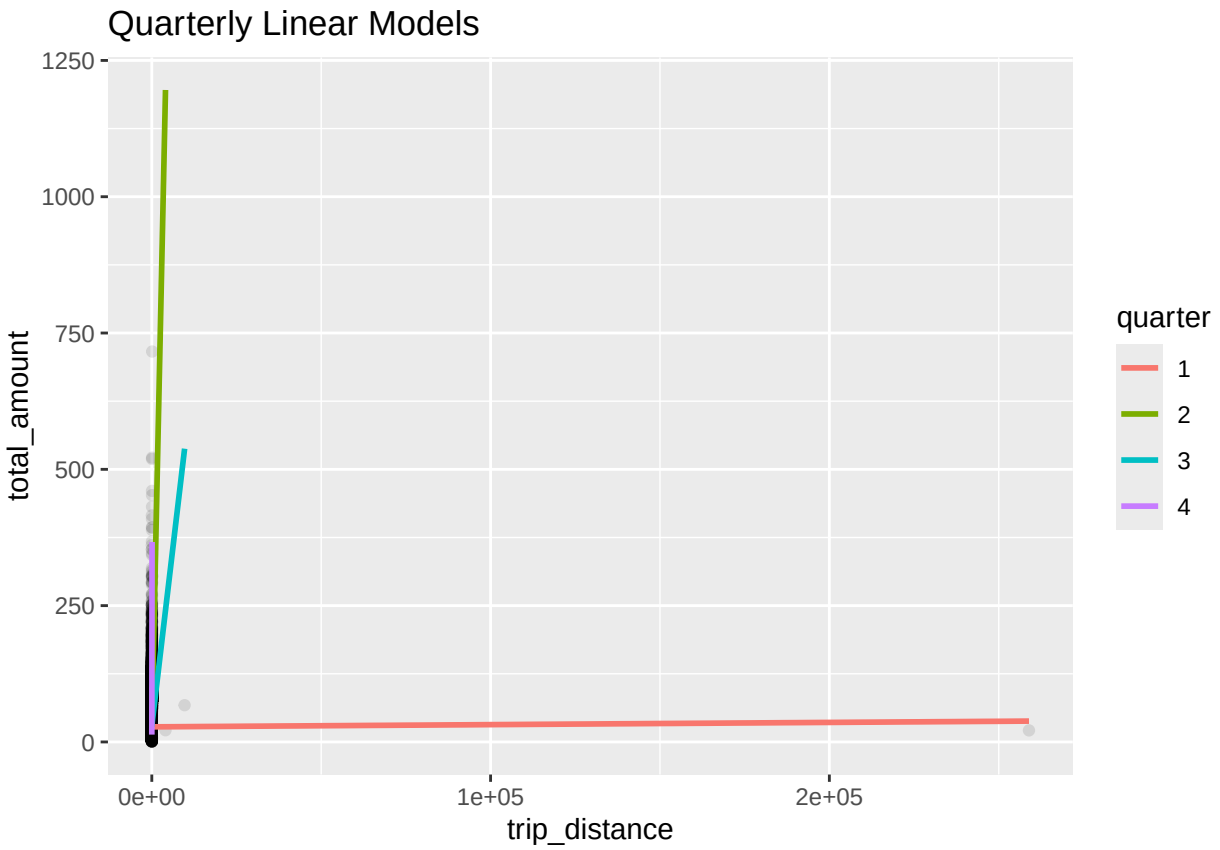
```
##
## QUARTERLY SLOPES
```

```r
print(quarter_slopes)
```

```
##    quarter        slope
##     <fctr>        <num>
## 1:       4 0.0067240347
## 2:       1 0.0009817402
## 3:       3 0.0027917679
## 4:       2 0.0018535948
```

```r
ggplot(plot_dt, aes(trip_distance, total_amount)) +
  geom_point(alpha=0.1) +
  geom_smooth(method="lm",
              aes(color=quarter),
              se=FALSE) +
  labs(title="Quarterly Linear Models")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Part 1

Summary statistics show that most trips are short (median distance is approximately 1.81 miles) with a median total amount of about 21 USD. However, extreme values are present, with maximum trip distances exceeding 193,000 miles and total amounts over 143,000 USD, indicating substantial outliers. The correlation between trip_distance and total_amount is approximately 0.007, which is unexpectedly small given the visible upward trend in the scatterplot. The scatterplot reveals that although most rides cluster at short distances, total fare generally increases with distance, but the relationship is highly dispersed and affected by extreme observations.

Part 2

The estimated intercept is approximately 28.94, and the slope is 0.00234. The slope implies that each additional mile increases total_amount by only \$0.0023, which is economically unrealistic for taxi fares. The model's R^2 is approximately 0.00005, meaning it explains virtually none of the variability in total_amount. The residuals versus fitted plot shows clear heteroskedasticity, with increasing spread as fitted values increase, and several extreme residuals caused by large outliers. These diagnostics indicate that the assumptions of linearity and constant variance are violated. Overall, the classical linear model performs poorly due to extreme values and non-constant variance.

Part 3

To allow for nonlinearity, a natural spline model with four degrees of freedom was fitted using trip_distance as the predictor. When compared visually to the linear model, the spline curve better captures curvature in the relationship, especially the steeper increase at shorter distances and the gradual change at longer distances. The RMSE for the classical linear model is approximately 67.96, while the spline model reduces it slightly to about 66.62, indicating modest improvement in predictive accuracy. Although the improvement in RMSE is small due to large data variability, the flexible model more accurately reflects the nonlinear pricing structure of taxi fares. This suggests that flexible regression provides a better functional representation of the relationship than a strictly linear model.

Part 4

Influential observations were identified using Cook's distance, revealing extreme trips with implausibly large distances and fares. These outliers heavily distort the classical linear regression estimates. A robust regression model using an MM-estimator was therefore fitted. The robust model produced an intercept of approximately 11.47 and a slope of 4.79, which are far more consistent with realistic taxi pricing (a base fare plus a per-mile rate). In contrast, the classical model's slope was near zero due to the influence of extreme values. Predictions at higher distances (e.g., 20–60 miles) differ dramatically between models, with the robust model giving economically plausible estimates while the classical model severely underestimates fares. This demonstrates that outliers can substantially bias ordinary least squares regression, and robust methods provide more stable and interpretable results in large real-world datasets.

Part 5

Separate regression models were fitted by quarter (Q1–Q4) to examine seasonal differences in the fare–distance relationship. The estimated slopes vary across quarters, indicating that the relationship between trip distance and total amount is not constant throughout the year. These seasonal differences may reflect variations in traffic conditions, congestion pricing, fuel costs, or tourism patterns. When plotted together, the regression lines show differing steepness across quarters, suggesting that fare structures or travel conditions change seasonally. This analysis indicates that temporal grouping provides additional insight beyond a single overall model.

Part 6

Linear modeling becomes inadequate when the underlying relationship is nonlinear, when variance is not constant, or when extreme outliers heavily influence parameter estimates. In this dataset, extreme trips distort the classical regression results, leading to unrealistic coefficients and negligible explanatory power. Flexible regression methods such as splines allow the data to determine the shape of the relationship and better capture nonlinear patterns. Robust regression reduces the influence of extreme observations and yields more stable, interpretable estimates. In large urban transportation datasets, where measurement errors and rare extreme events are common, relying solely on classical linear regression can lead to misleading conclusions. Combining flexible and robust approaches produces more reliable modeling outcomes.

Github Link: https://github.com/SylTana/DSC1105-Exploratory-Data-Analysis/tree/main/FA3