Appendix A

Exploratory Data Analysis

Raw distributions before preprocessing.

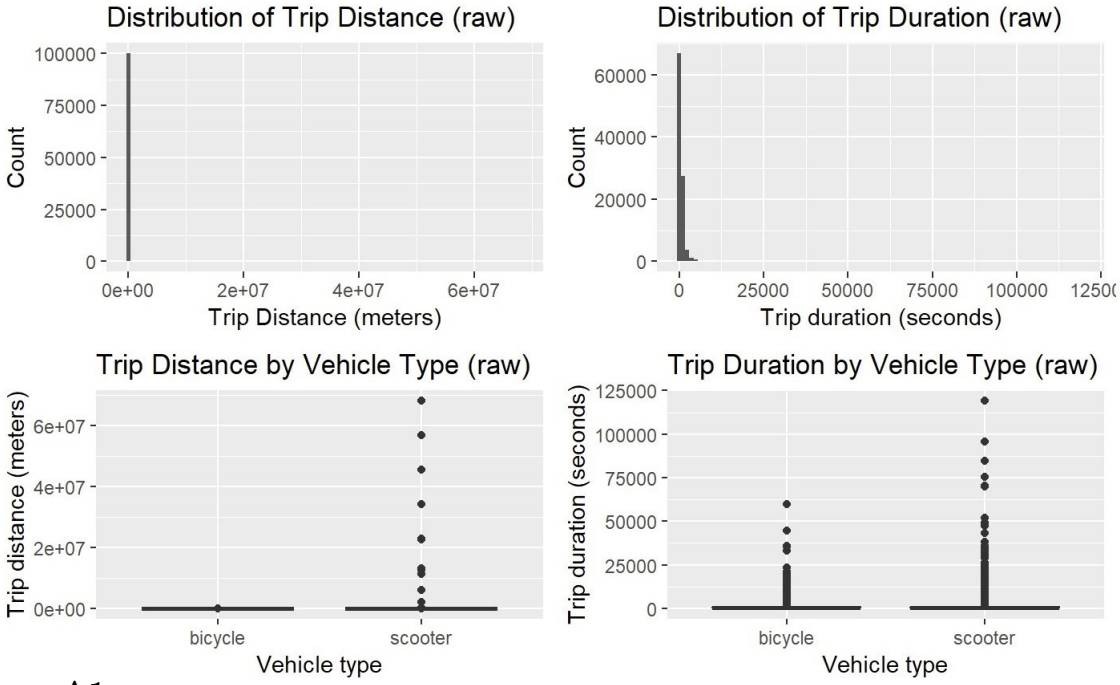

Figure A1

Raw-scale exploratory plots for trip distance and duration (with outliers).

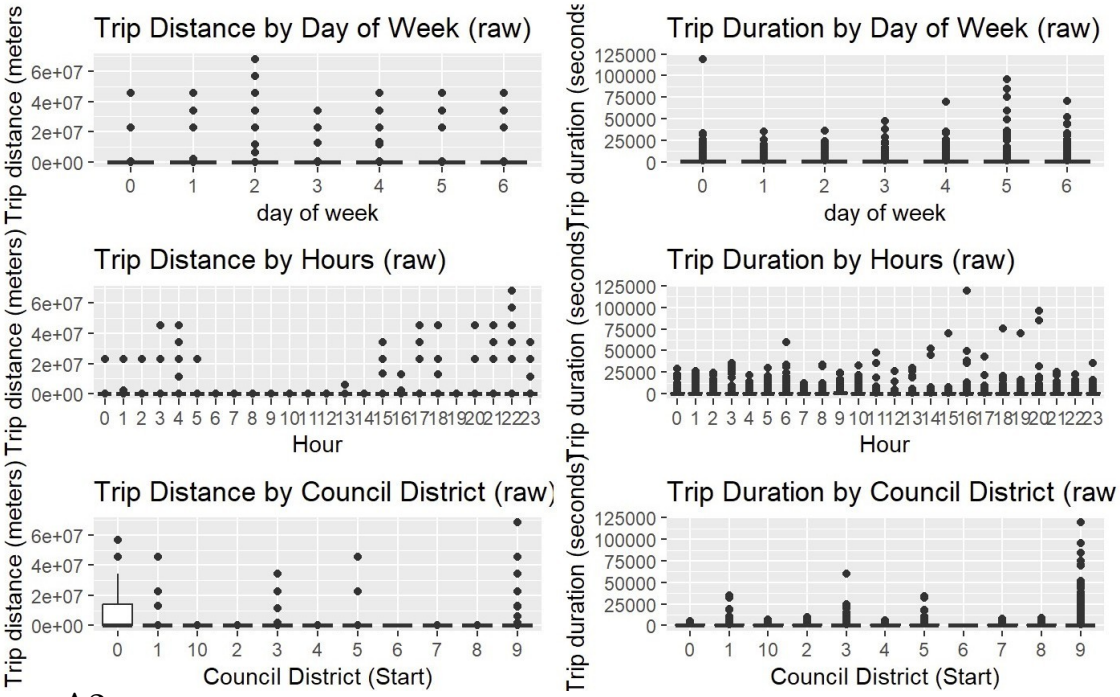

Figure A2

Additional raw-scale exploratory plots (with outliers).

Raw distributions after preprocessing (raw scale).

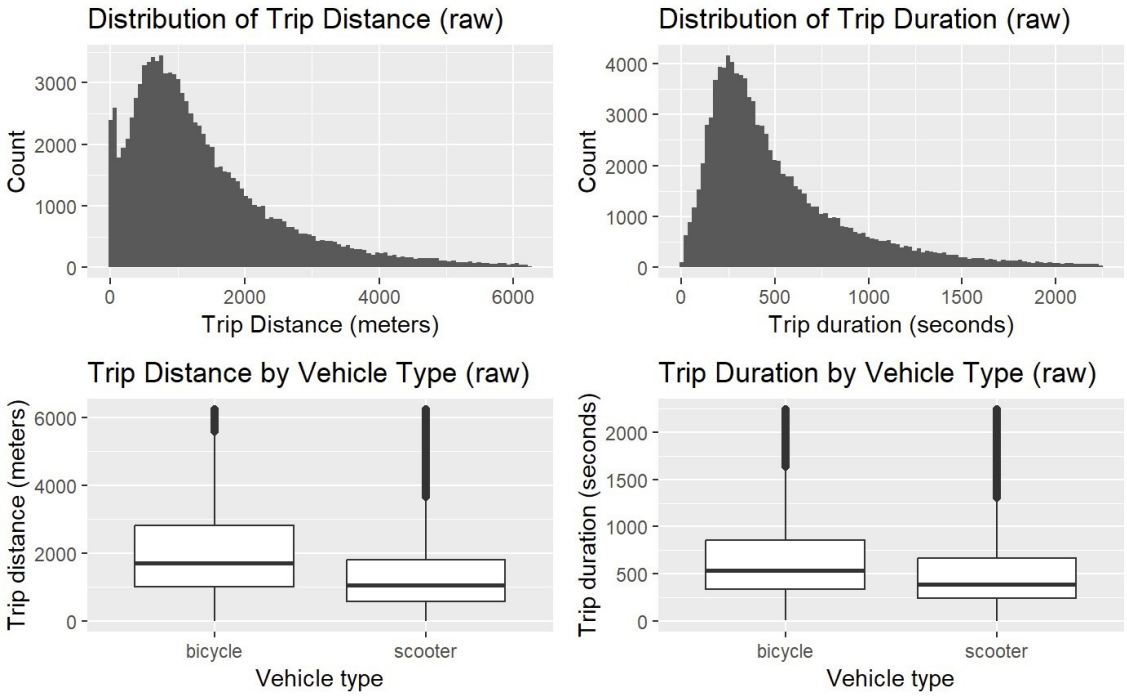

Figure A3

Raw-scale distribution plots after IQR-based trimming.

n <int>	mean_dist <dbl>	median_dist <dbl>	max_dist <dbl>	mean_dur <dbl>	median_dur <dbl>	max_dur <dbl>
5486004	1383.348	1078	6243	525.632	400	2243

Figure A4

Summary of data for raw-scale variables after IQR-based trimming.

Measure <chr>	Q1 <dbl>	Q3 <dbl>	IQR <dbl>	Lower_1_5 <dbl>	Upper_1_5 <dbl>	Lower_3 <dbl>	Upper_3 <dbl>
Trip Distance	611	2019	1408	0	4131	0	6243
Trip Duration	251	749	498	0	1496	0	2243

Figure A5

IQR-based trimming thresholds (range visualization).

Distributions after transformation (log scale).

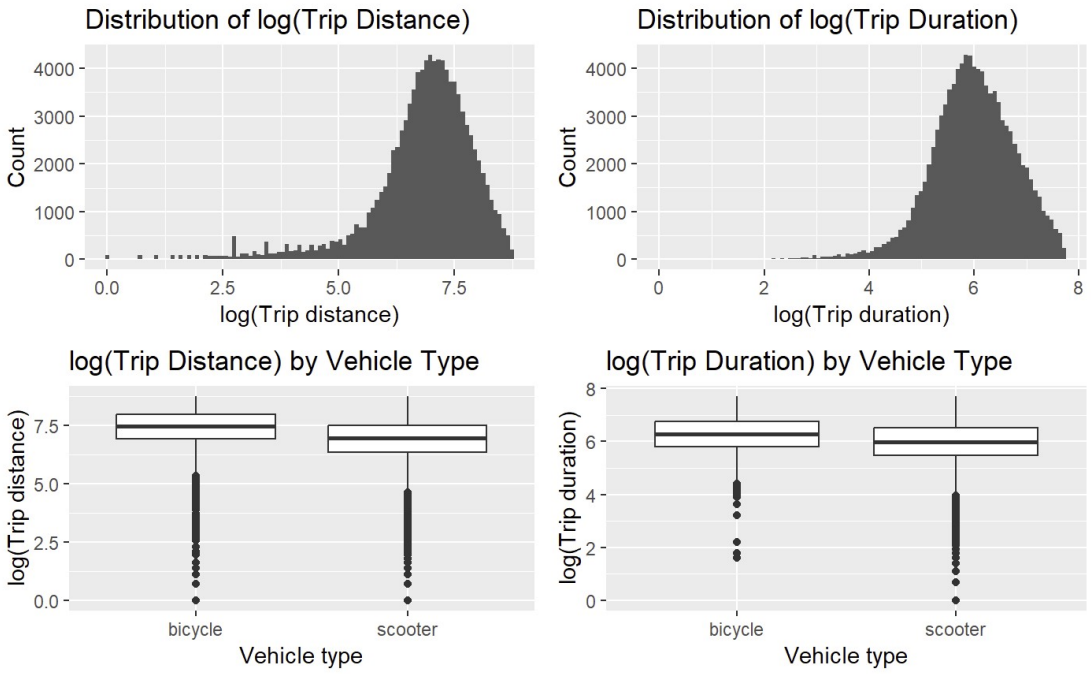

Figure A6

Log-scale exploratory plots for transformed trip distance and duration.

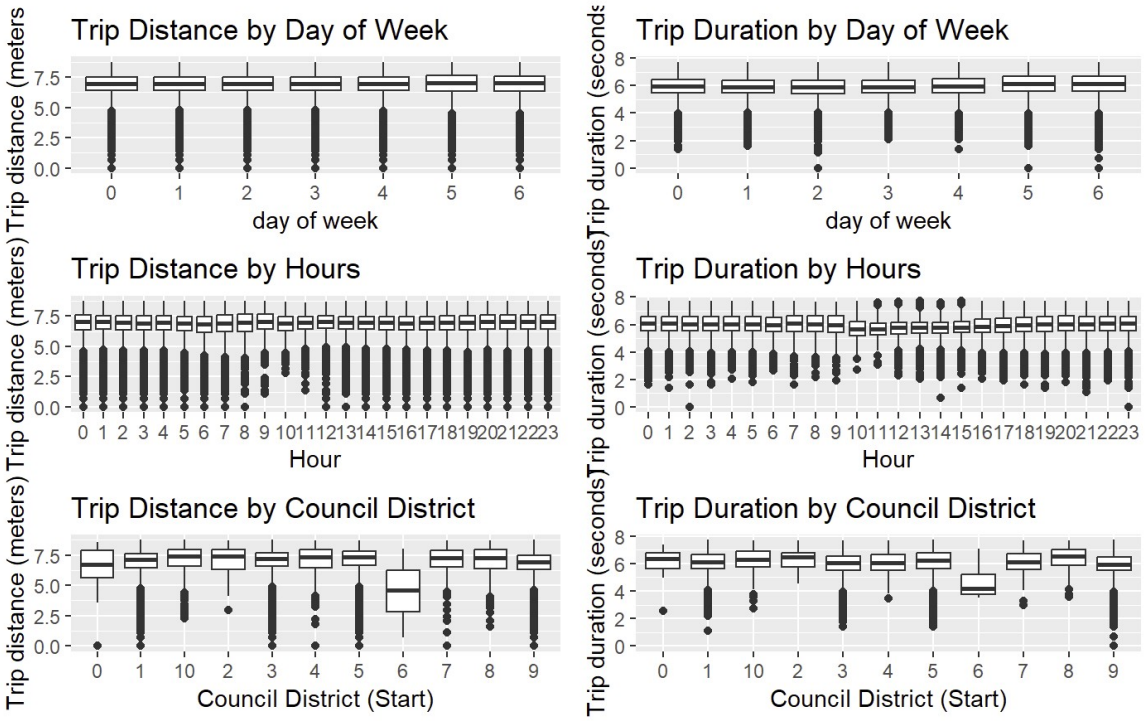

Figure A7

Additional log-scale exploratory plots (log scale).

Appendix B

Model Diagnostics: Question 1 (Trip Distance)

Raw distance model (no transformation).

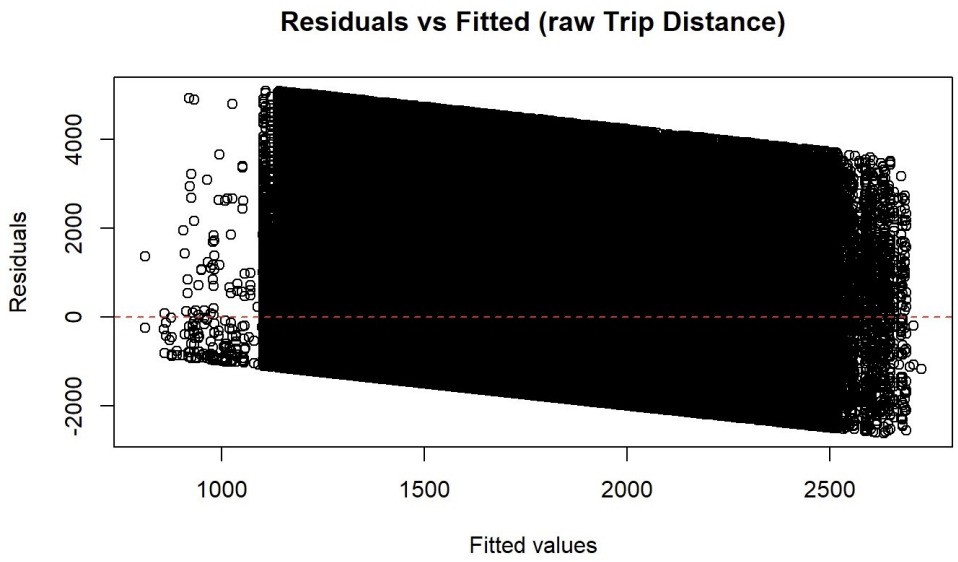

Figure B1

Residuals vs. fitted values for the raw-distance OLS model.

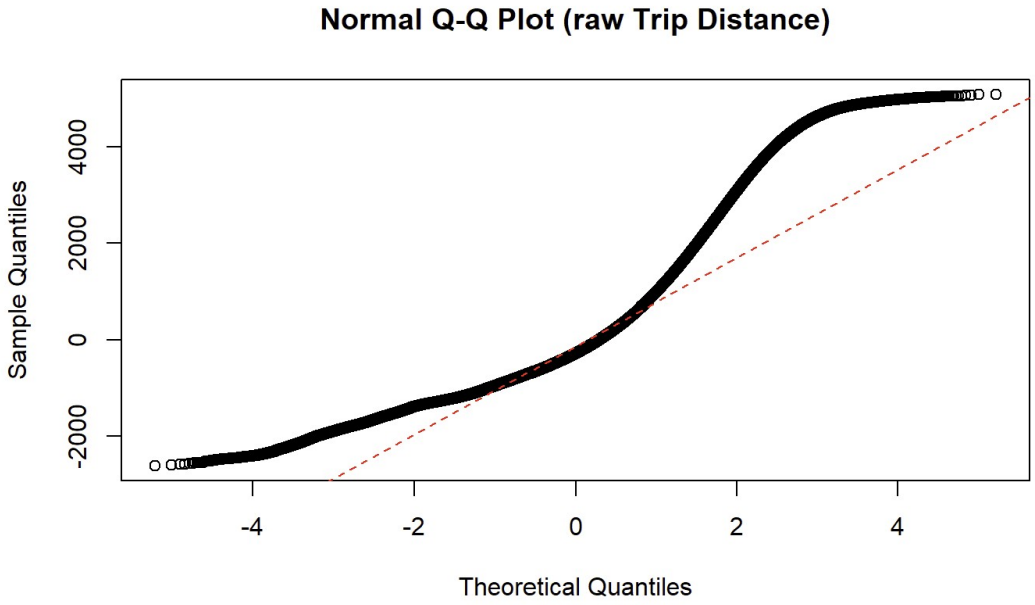

Figure B2

Normal Q–Q plot for the raw-distance OLS model residuals.

Log-transformed distance model.

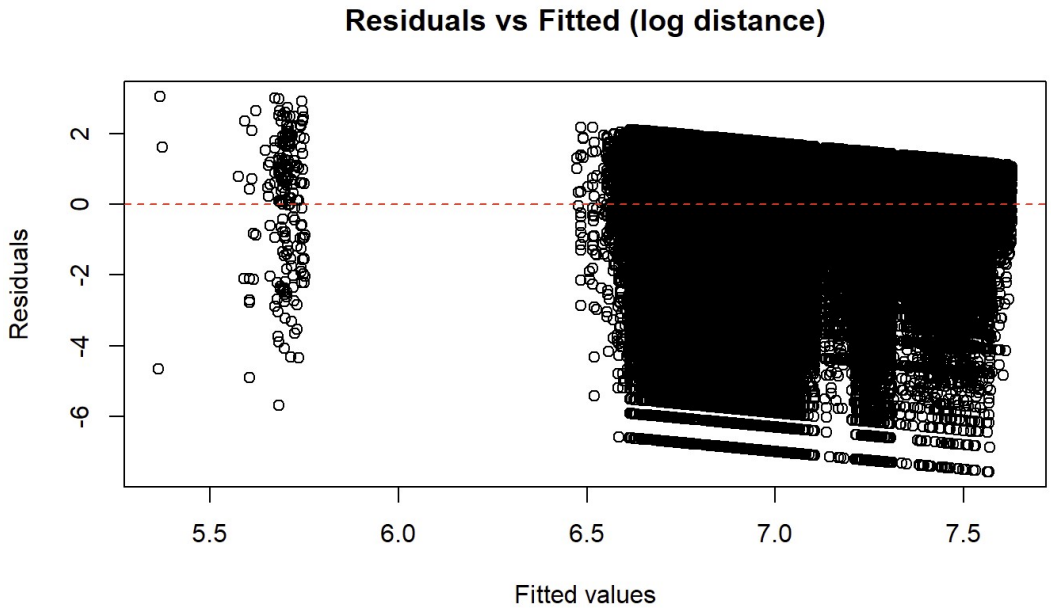

Figure B3

Residuals vs. fitted values for the log-distance OLS model.

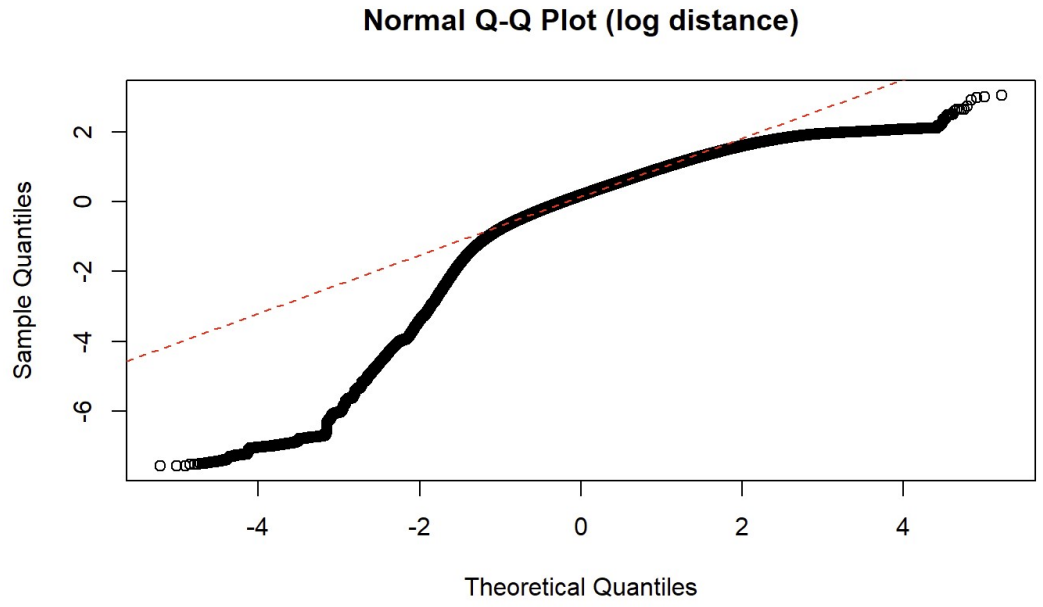

Figure B4

Normal Q–Q plot for the log-distance OLS model residuals.

Robustness checks.

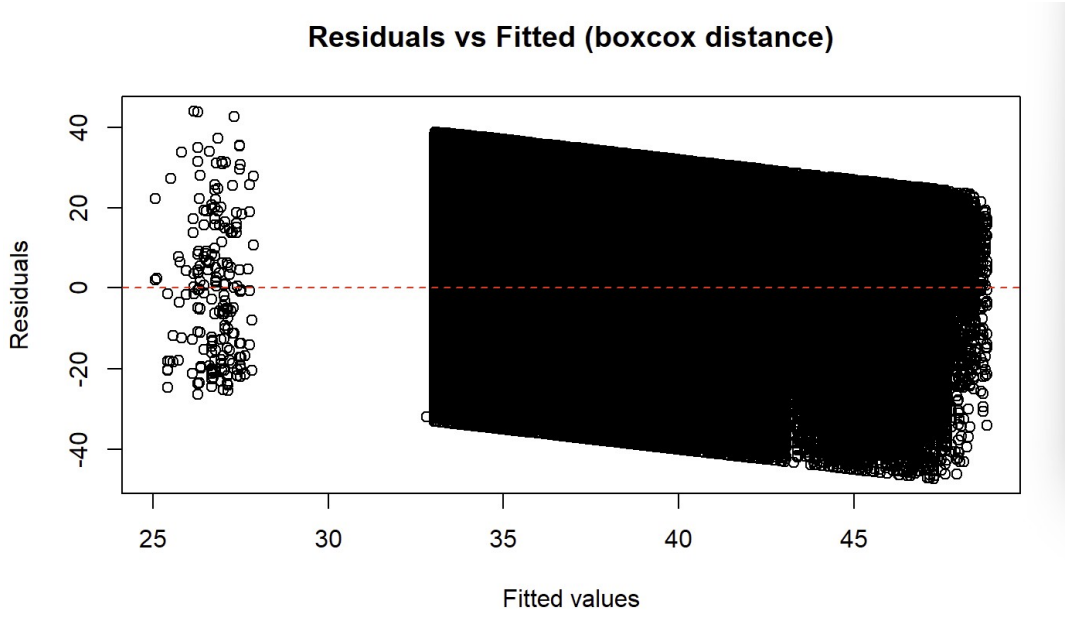

Figure B5

Residuals vs. fitted values for the Box–Cox distance model.

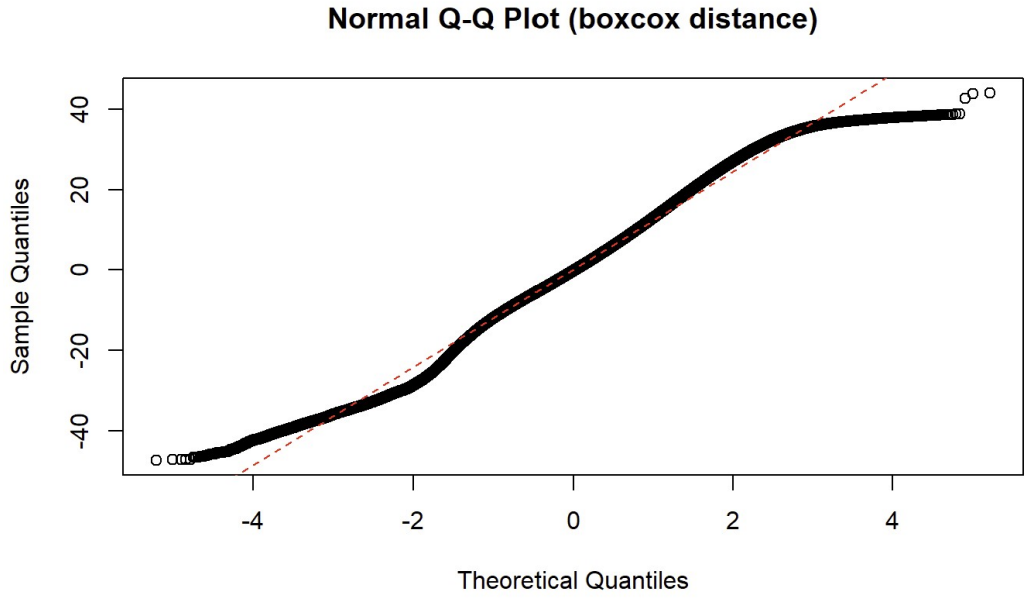

Figure B6

Normal Q–Q plot for the Box–Cox distance model residuals.

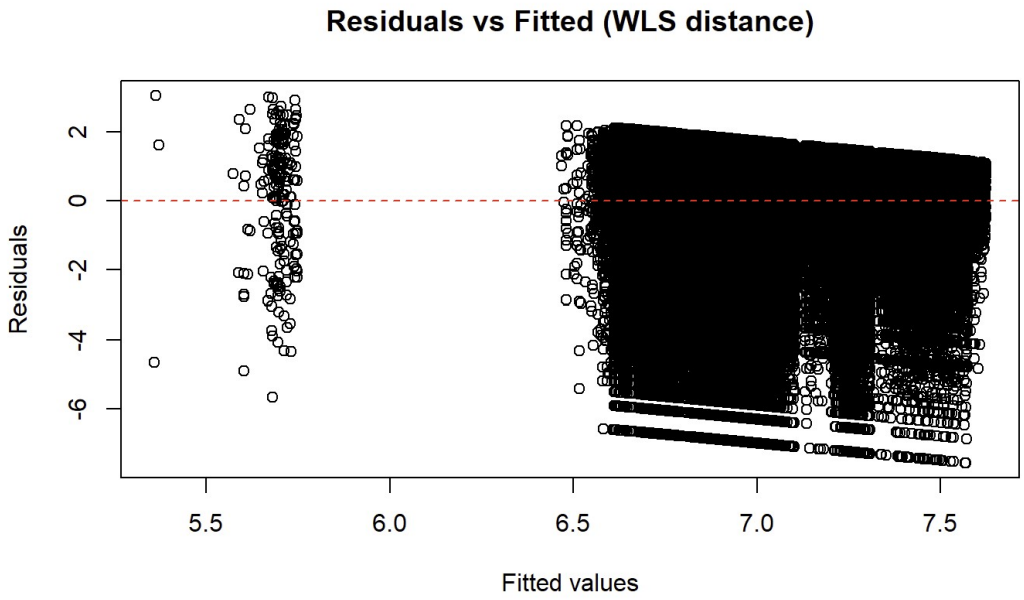

Figure B7

Residuals vs. fitted values for the WLS distance model.

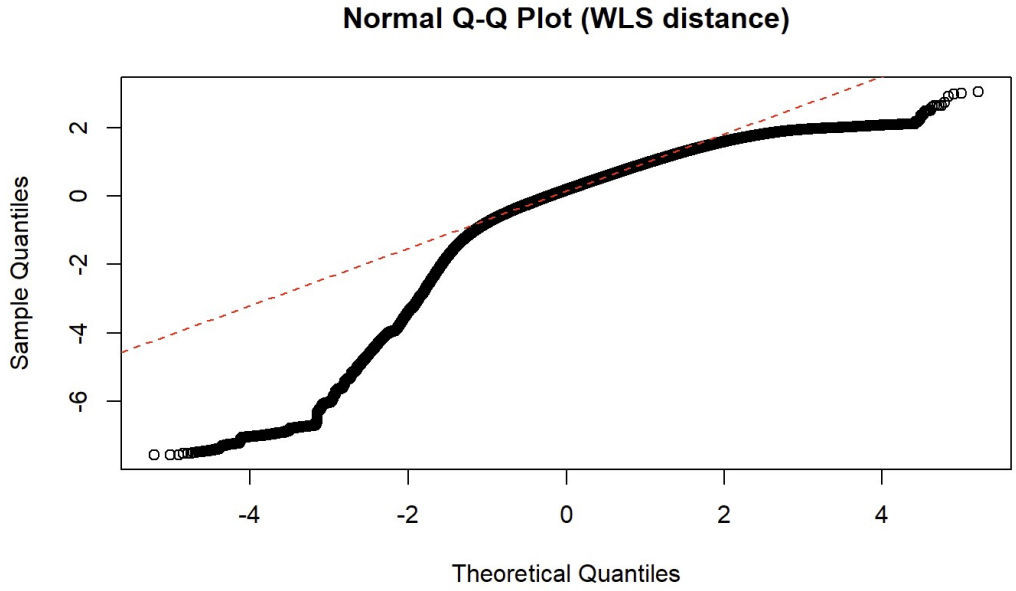

Figure B8

Normal Q–Q plot for the WLS distance model residuals.

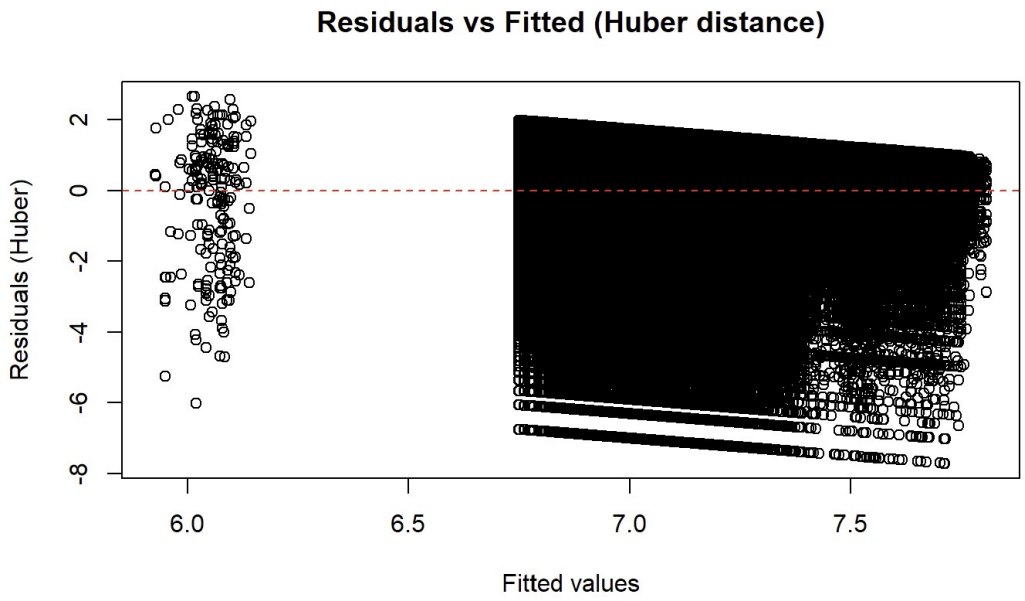

Figure B9

Residuals vs. fitted values for the Huber distance model.

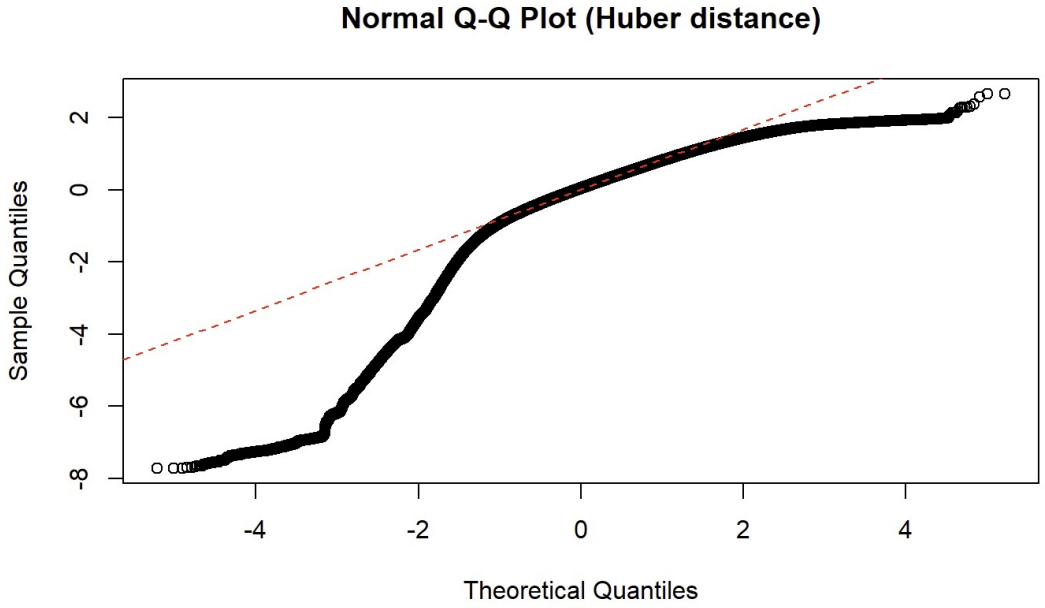

Figure B10

Normal Q–Q plot for the Huber distance model residuals.

	Vehicle <chr>	OLS <dbl>	Boxcox <dbl>	WLS <dbl>	Huber <dbl>
	bicycle (baseline)	0.0000000	0.000000	0.0000000	0.0000000
vehicle_typescooter	scooter	-0.5183198	-7.217821	-0.5190153	-0.4542413

Figure B11

Key coefficient comparison across distance models.

	GVIF	Df	GVIF^(1/(2*Df))
vehicle_type	1.009107	1	1.004543
hour_f	1.058705	23	1.001241
dow_f	1.054343	6	1.004420
district_start_f	1.018104	11	1.000816

Figure B12

GVIF diagnostics for the distance model predictors.

Model <chr>	BIC <dbl>
OLS_raw	92487424
OLS_log	17005040
BoxCox_log	43809476
WLS_log	17019218
Huber_log	17099228

Figure B13

BIC comparison across candidate distance models.

term <chr>	estimate <dbl>	std_error <dbl>	ratio <dbl>	pct_change <dbl>
Scooter (vs bicycle)	-0.5183198	0.002148842	0.5955203	-40.44797

Figure B14

Results of Question 1.

Appendix C

Model Diagnostics: Question 2 (Trip Duration)

Raw duration model and log-transformed model.

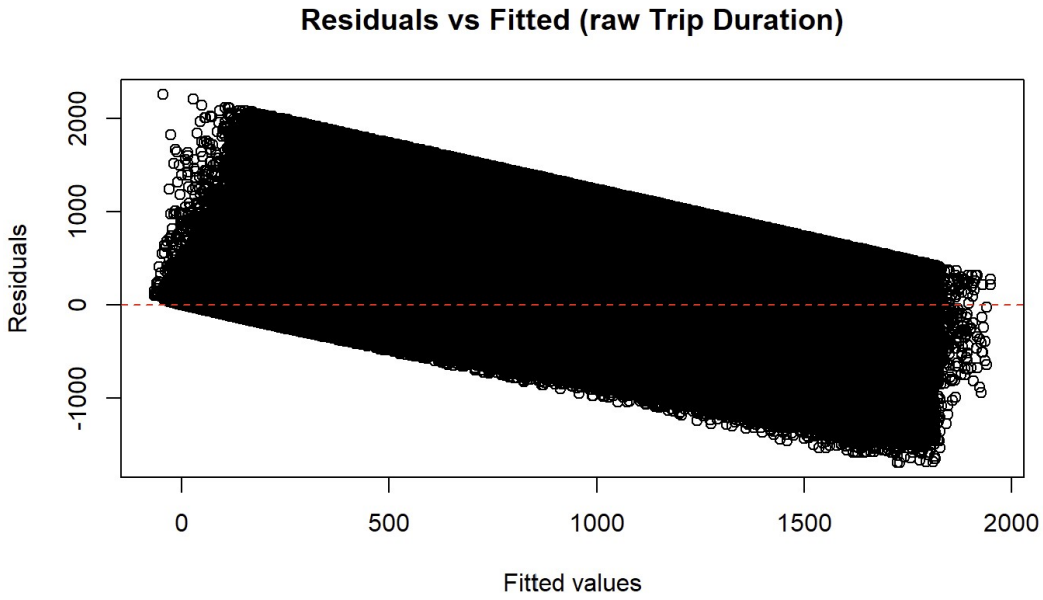

Figure C1

Residuals vs. fitted values for the raw-duration OLS model.

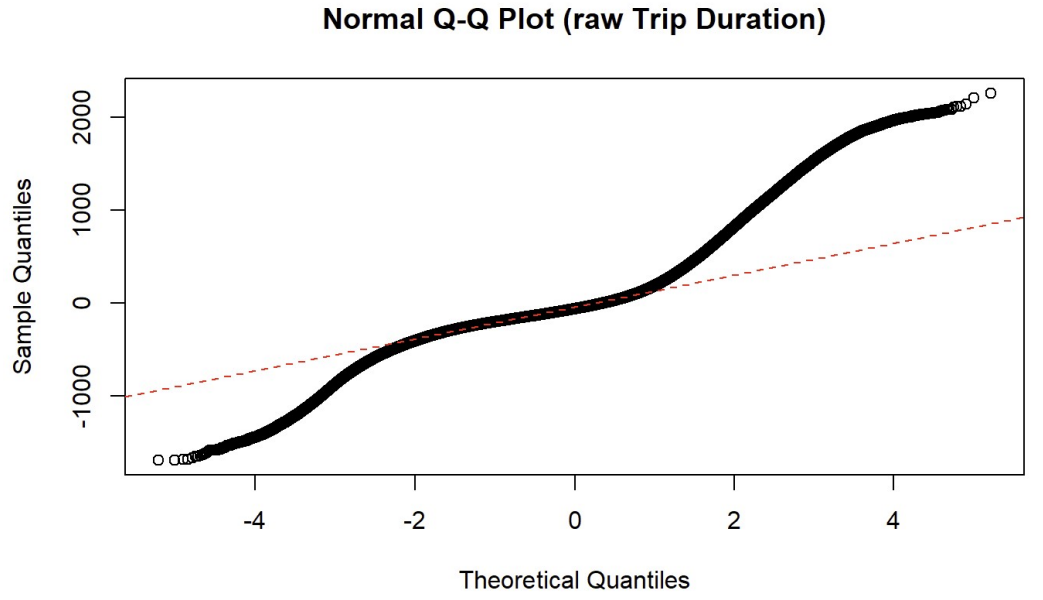

Figure C2

Normal Q–Q plot for the raw-duration OLS model residuals.

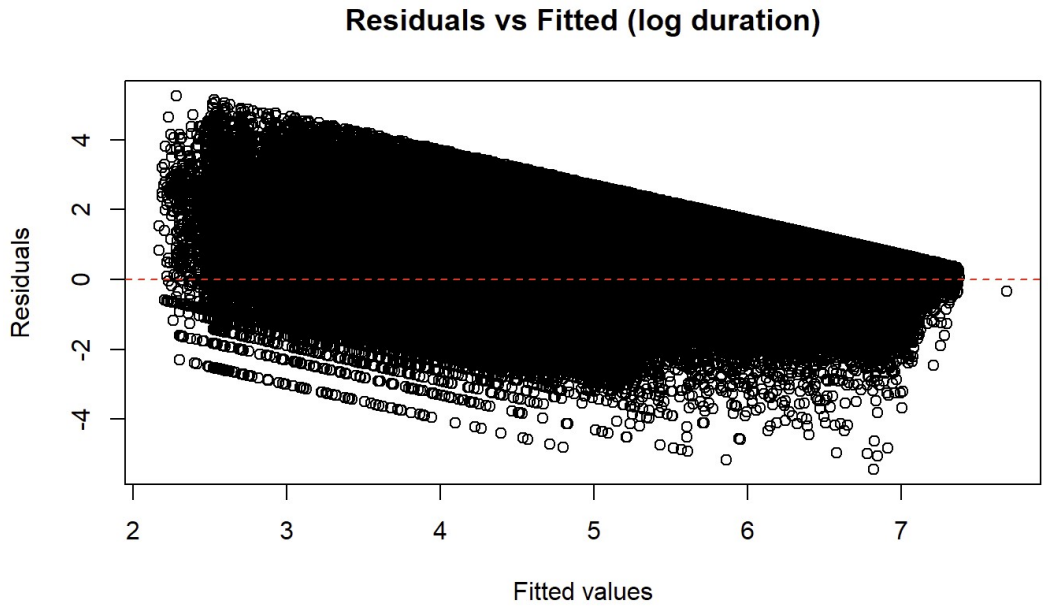

Figure C3

Residuals vs. fitted values for the log-duration OLS model.

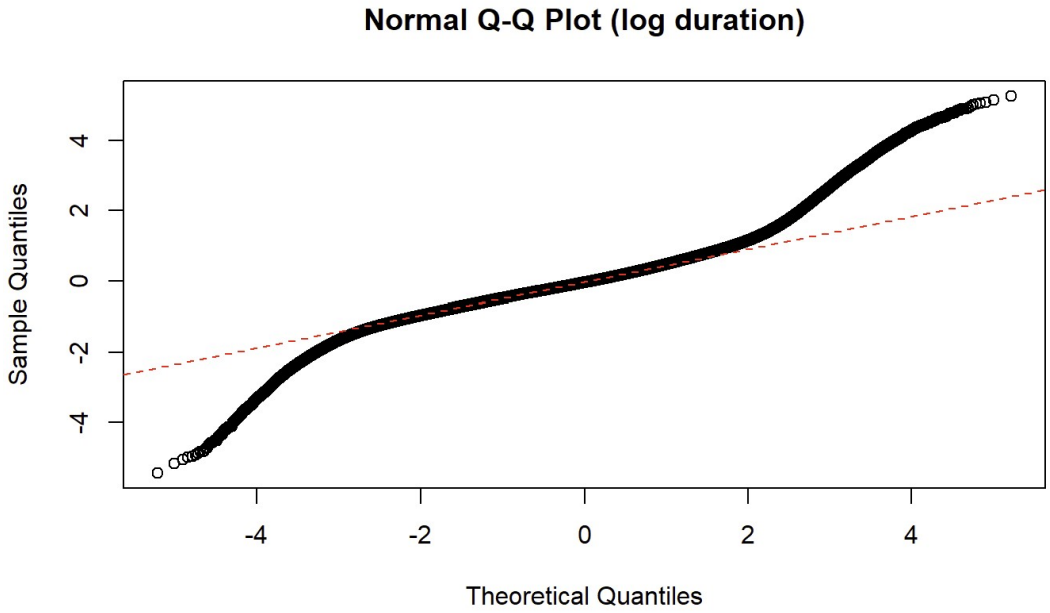

Figure C4

Normal Q–Q plot for the log-duration OLS model residuals.

Robustness checks and interaction model.

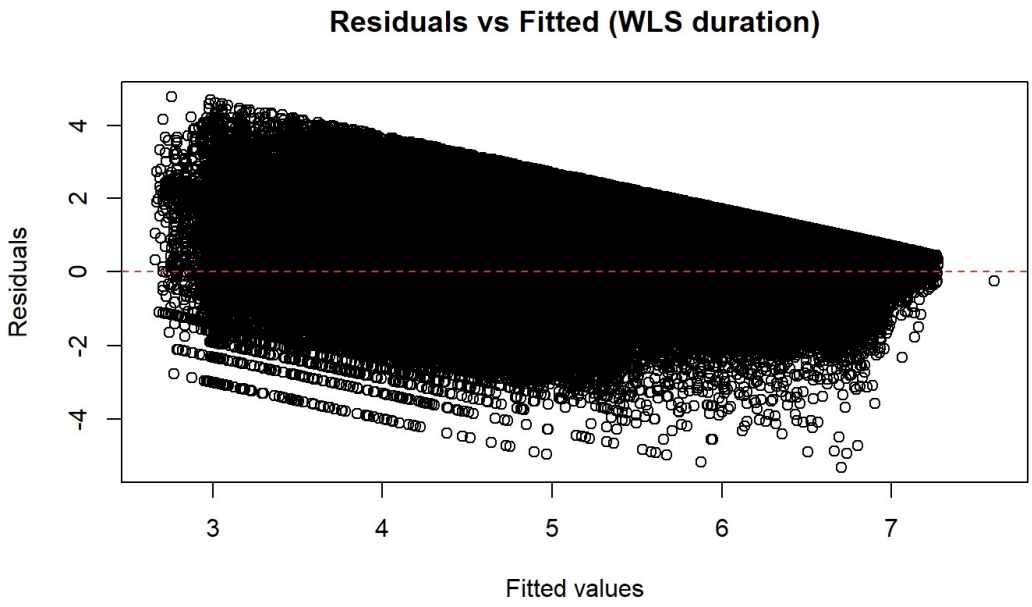

Figure C5

Residuals vs. fitted values for the WLS duration model.

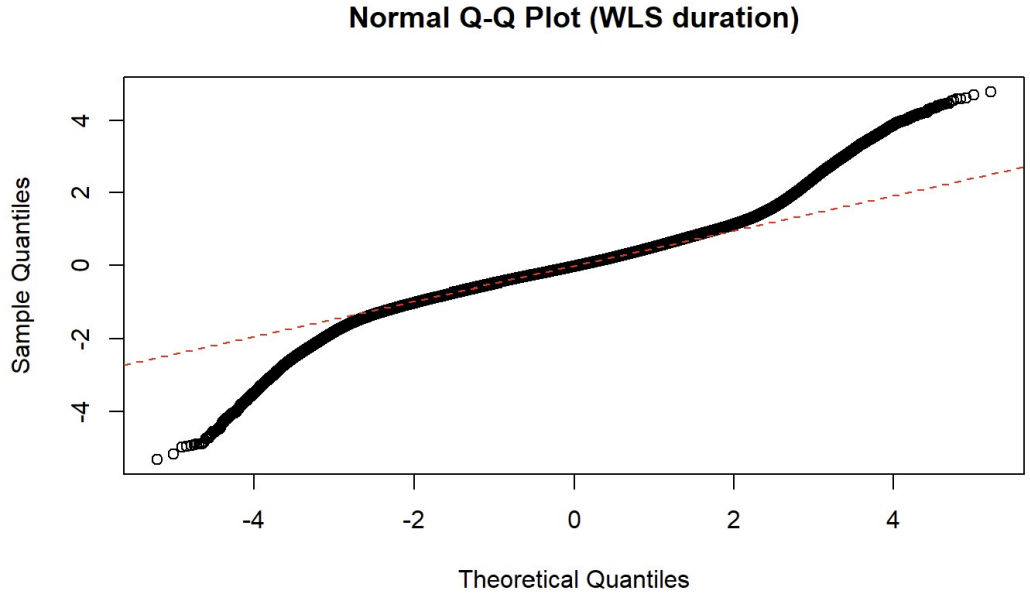

Figure C6

Normal Q–Q plot for the WLS duration model residuals.

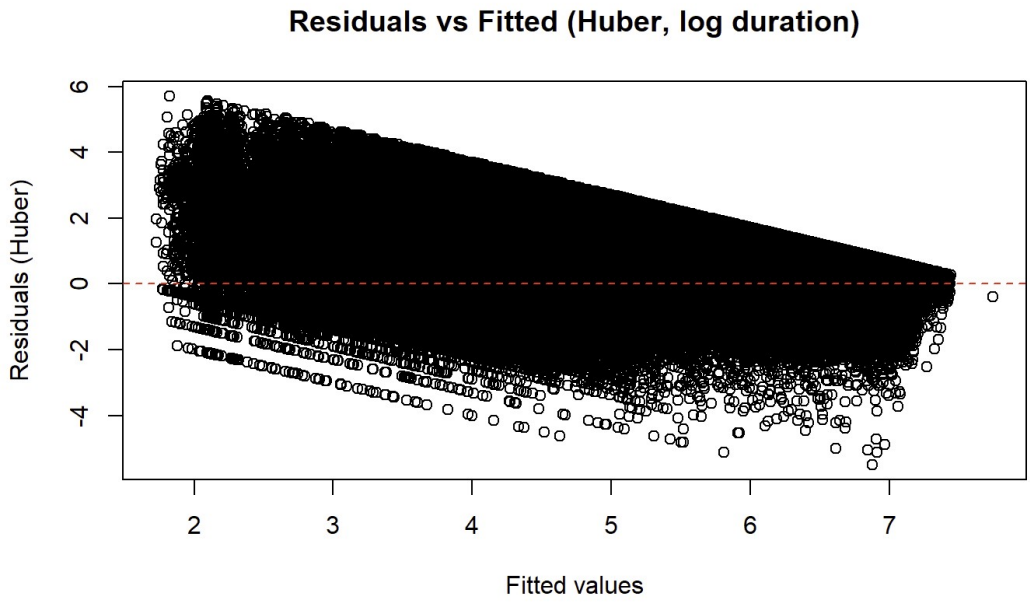

Figure C7

Residuals vs. fitted values for the Huber (log-duration) model.

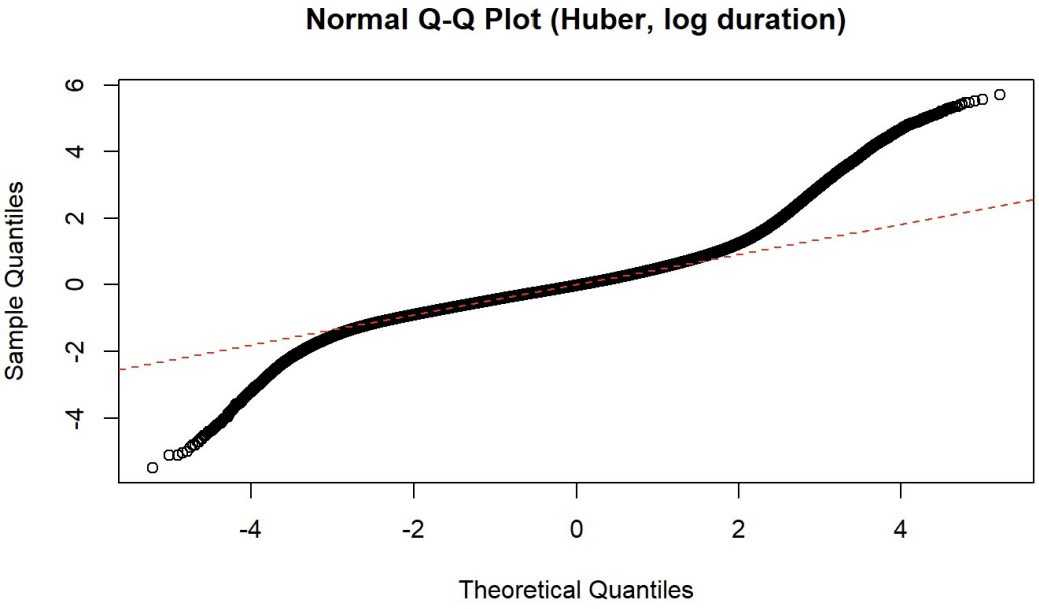

Figure C8

Normal Q–Q plot for the Huber (log-duration) model residuals.

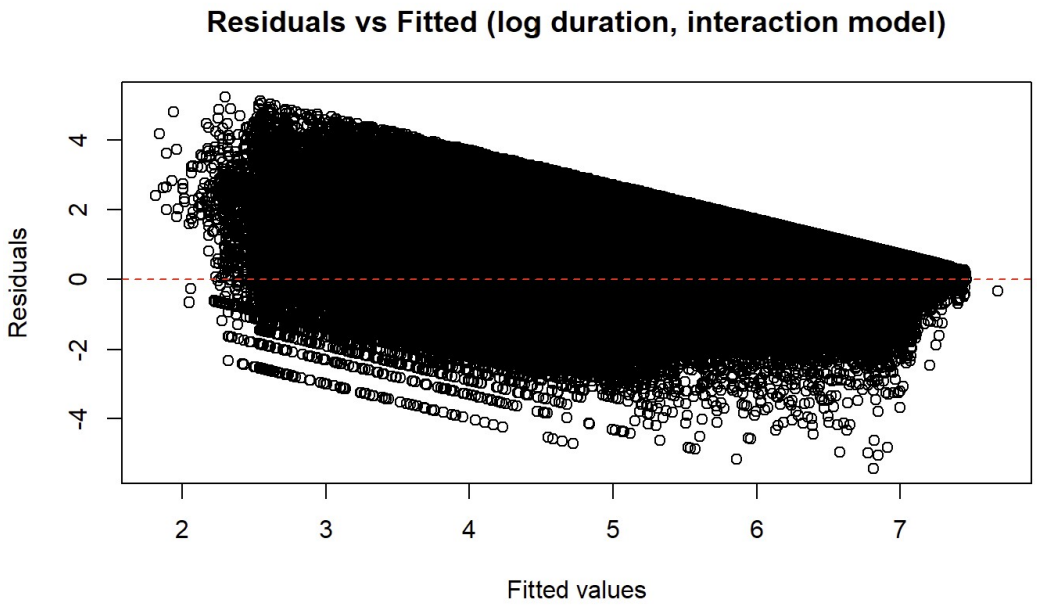

Figure C9

Residuals vs. fitted values for the log-duration interaction model.

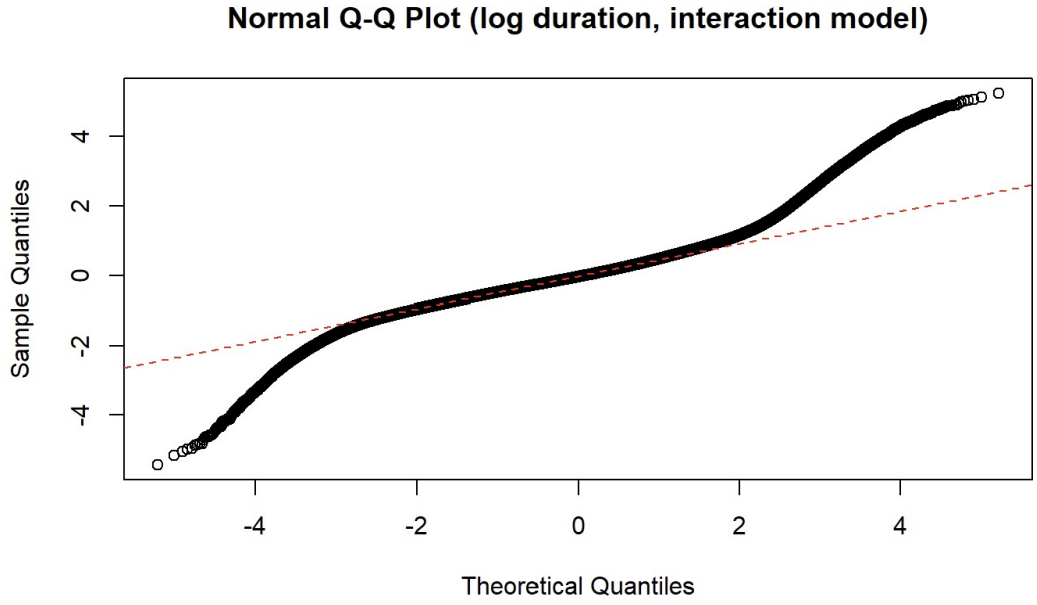

Figure C10

Normal Q–Q plot for the interaction model residuals.

Model <chr>	BIC <dbl>
OLS_raw	77480492
OLS_log	8621600
BoxCox_log	25917405
WLS_log	9992741
Huber_log	8728249

Figure C11

BIC comparison across candidate duration models.

Model <chr>	BIC <dbl>
main_log	8621600
Veh_dis_log	8618148
Veh_hour_log	8619729
Veh_district_log	8621300

Figure C12

BIC comparison across interaction candidates (log-duration).

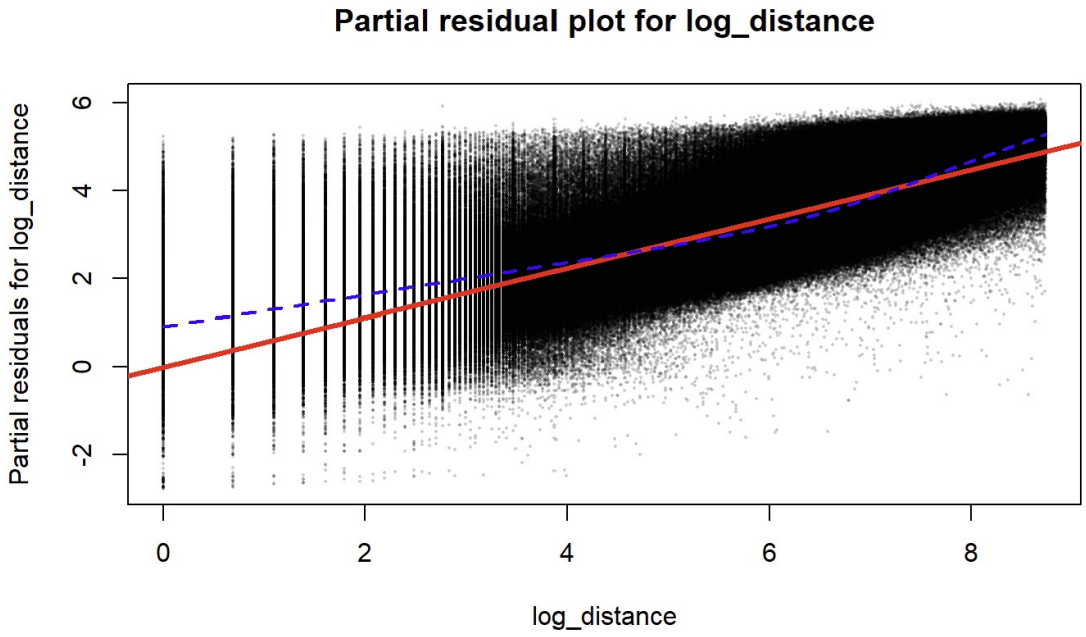

Figure C13

Partial residual plot for `log_distance` *in the duration model.*

	Vehicle <chr>	OLS <dbl>	BoxCox <dbl>	WLS <dbl>	Huber <dbl>
	bicycle (baseline)	0.00000000	0.0000000	0.00000000	0.000000000
vehicle_typescooter	scooter	-0.02805409	0.1982758	-0.05877432	-0.006723275

Figure C14

Key coefficient comparison across duration models.

	GVIF	Df	GVIF^(1/(2*Df))
vehicle_type	61.715967	1	7.855951
log_distance	28.067034	1	5.297833
hour_f	1.059670	23	1.001261
dow_f	1.054380	6	1.004422
district_start_f	1.023551	11	1.001059
vehicle_type:log_distance	79.985775	1	8.943477

Figure C15

GVIF diagnostics for the duration model predictors.

Model <chr>	a1_scooter <dbl>	a3_interaction <dbl>	log_distance_used <dbl>	delta_scooter_log <dbl>
Main-effects (no interaction)	-0.028054	*NA*	*NA*	-0.028054
Interaction (vehicle_type × log_distance)	0.428973	-0.062602	6.804167	0.003020

Figure C16

Results of Q2.

Appendix D

Full Summary of Selected Models

Model from Question 1: Full summary of OLS Model (log scale).

```
Call:
lm(formula = log_distance ~ vehicle_type + hour_f + dow_f + district_start_f,
    data = trips_2019)

Residuals:
    Min      1Q  Median      3Q     Max
-7.5680 -0.4148  0.1839  0.7148  3.0537

Coefficients:
                    Estimate Std. Error  t value Pr(>|t|)
(Intercept)         7.135650   0.038454  185.564  < 2e-16 ***
vehicle_typescooter -0.518320   0.002149 -241.209  < 2e-16 ***
hour_f1             -0.017469   0.002631   -6.640 3.15e-11 ***
hour_f2             -0.022280   0.002740   -8.130 4.28e-16 ***
hour_f3             -0.023225   0.002835   -8.191 2.58e-16 ***
hour_f4             -0.023485   0.002974   -7.898 2.84e-15 ***
hour_f5             -0.099775   0.003381  -29.508  < 2e-16 ***
hour_f6             -0.129893   0.003929  -33.064  < 2e-16 ***
hour_f7             -0.063489   0.004337  -14.639  < 2e-16 ***
hour_f8             -0.038733   0.005957   -6.502 7.95e-11 ***
hour_f9             -0.006326   0.008843   -0.715 0.474384
hour_f10             0.011898   0.009962    1.194 0.232368
hour_f11            -0.094305   0.007863  -11.993  < 2e-16 ***
hour_f12             0.005546   0.005068    1.094 0.273805
hour_f13             0.031317   0.003508    8.926  < 2e-16 ***
hour_f14            -0.017633   0.003193   -5.522 3.34e-08 ***
hour_f15            -0.057582   0.003138  -18.350  < 2e-16 ***
hour_f16            -0.048138   0.002936  -16.396  < 2e-16 ***
hour_f17            -0.049691   0.002717  -18.290  < 2e-16 ***
hour_f18            -0.030394   0.002644  -11.497  < 2e-16 ***
hour_f19            -0.017133   0.002627   -6.523 6.90e-11 ***
hour_f20            -0.008002   0.002593   -3.086 0.002026 **
hour_f21             0.002844   0.002556    1.113 0.265876
hour_f22             0.028587   0.002503   11.423  < 2e-16 ***
hour_f23             0.025952   0.002500   10.379  < 2e-16 ***
dow_f1              -0.016064   0.001986   -8.086 6.14e-16 ***
dow_f2              -0.014945   0.001988   -7.516 5.64e-14 ***
dow_f3              -0.004832   0.001943   -2.487 0.012887 *
dow_f4               0.001306   0.001864    0.701 0.483476
dow_f5              -0.003959   0.001788   -2.214 0.026853 *
dow_f6              -0.010944   0.001818   -6.021 1.73e-09 ***
district_start_f1    0.276188   0.038390    7.194 6.28e-13 ***
district_start_f10   0.460076   0.039137   11.755  < 2e-16 ***
district_start_f2    0.232989   0.042787    5.445 5.17e-08 ***
district_start_f3    0.336295   0.038366    8.765  < 2e-16 ***
district_start_f4    0.420622   0.040031   10.508  < 2e-16 ***
district_start_f5    0.405061   0.038426   10.541  < 2e-16 ***
district_start_f6   -0.896498   0.089802   -9.983  < 2e-16 ***
district_start_f7    0.416205   0.039506   10.535  < 2e-16 ***
district_start_f8    0.307449   0.038909    7.902 2.75e-15 ***
district_start_f9    0.141769   0.038340    3.698 0.000218 ***
district_start_fNone -1.224563   0.659186   -1.858 0.063213 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.14 on 5485962 degrees of freedom
Multiple R-squared:  0.01775,   Adjusted R-squared:  0.01774
F-statistic:  2418 on 41 and 5485962 DF,  p-value: < 2.2e-16
```

Figure D1

Model from Question 2: Full summary of Main-Effect Model (log scale).

```
Call:
lm(formula = log_duration ~ vehicle_type + log_distance + hour_f +
    dow_f + district_start_f, data = trips_2019)

Residuals:
    Min      1Q  Median      3Q     Max
-5.4317 -0.3336 -0.0367  0.2937  5.2592

Coefficients:
                    Estimate Std. Error  t value Pr(>|t|)
(Intercept)        2.8845026  0.0179665  160.549  < 2e-16 ***
vehicle_typescooter -0.0280541  0.0010061  -27.883  < 2e-16 ***
log_distance        0.4999024  0.0001989 2513.887  < 2e-16 ***
hour_f1             0.0072145  0.0012254    5.887 3.93e-09 ***
hour_f2             0.0012516  0.0012764    0.981 0.326814
hour_f3            -0.0040595  0.0013206   -3.074 0.002112 **
hour_f4            -0.0200463  0.0013850  -14.474  < 2e-16 ***
hour_f5             0.0158552  0.0015750   10.067  < 2e-16 ***
hour_f6             0.0122678  0.0018300    6.704 2.03e-11 ***
hour_f7             0.0267895  0.0020201   13.262  < 2e-16 ***
hour_f8            -0.0116477  0.0027748   -4.198 2.70e-05 ***
hour_f9            -0.1128507  0.0041186  -27.400  < 2e-16 ***
hour_f10           -0.2525382  0.0046400  -54.426  < 2e-16 ***
hour_f11           -0.3411018  0.0036626  -93.132  < 2e-16 ***
hour_f12           -0.3014430  0.0023604 -127.709  < 2e-16 ***
hour_f13           -0.2645194  0.0016341 -161.870  < 2e-16 ***
hour_f14           -0.2293047  0.0014872 -154.188  < 2e-16 ***
hour_f15           -0.1783729  0.0014616 -122.042  < 2e-16 ***
hour_f16           -0.1182519  0.0013675  -86.475  < 2e-16 ***
hour_f17           -0.0840287  0.0012654  -66.403  < 2e-16 ***
hour_f18           -0.0468970  0.0012313  -38.086  < 2e-16 ***
hour_f19           -0.0143503  0.0012234  -11.730  < 2e-16 ***
hour_f20           -0.0081430  0.0012076   -6.743 1.55e-11 ***
hour_f21           -0.0031135  0.0011906   -2.615 0.008919 **
hour_f22           -0.0087702  0.0011656   -7.524 5.31e-14 ***
hour_f23           -0.0042086  0.0011646   -3.614 0.000302 ***
dow_f1             -0.0475681  0.0009252  -51.412  < 2e-16 ***
dow_f2             -0.0613278  0.0009261  -66.221  < 2e-16 ***
dow_f3             -0.0407633  0.0009050  -45.044  < 2e-16 ***
dow_f4              0.0112928  0.0008684   13.004  < 2e-16 ***
dow_f5              0.1147309  0.0008330  137.730  < 2e-16 ***
dow_f6              0.1219175  0.0008466  144.012  < 2e-16 ***
district_start_f1  -0.2085172  0.0178808  -11.662  < 2e-16 ***
district_start_f10 -0.2159013  0.0182290  -11.844  < 2e-16 ***
district_start_f2  -0.0959542  0.0199289   -4.815 1.47e-06 ***
district_start_f3  -0.3238657  0.0178696  -18.124  < 2e-16 ***
district_start_f4  -0.2260694  0.0186450  -12.125  < 2e-16 ***
district_start_f5  -0.2250908  0.0178978  -12.576  < 2e-16 ***
district_start_f6   0.0220634  0.0418271    0.527 0.597853
district_start_f7  -0.2870359  0.0184006  -15.599  < 2e-16 ***
district_start_f8   0.0094983  0.0181223    0.524 0.600193
district_start_f9  -0.2617185  0.0178576  -14.656  < 2e-16 ***
district_start_fNone 0.5041681  0.3070251    1.642 0.100568
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5309 on 5485961 degrees of freedom
Multiple R-squared:  0.5505,    Adjusted R-squared:  0.5505
F-statistic: 1.6e+05 on 42 and 5485961 DF,  p-value: < 2.2e-16
```

Figure D2

Model from Question 2: Full summary of Interaction Model (log scale).

```
Call:
lm(formula = log_duration ~ vehicle_type * log_distance + hour_f +
    dow_f + district_start_f, data = trips_2019)

Residuals:
    Min      1Q  Median      3Q     Max
-5.4275 -0.3332 -0.0371  0.2932  5.2435

Coefficients:
                                  Estimate Std. Error  t value Pr(>|t|)
(Intercept)                      2.4415454  0.0194719  125.388  < 2e-16 ***
vehicle_typescooter              0.4289727  0.0078246   54.823  < 2e-16 ***
log_distance                     0.5602536  0.0010438  536.751  < 2e-16 ***
hour_f1                          0.0073281  0.0012251    5.982 2.21e-09 ***
hour_f2                          0.0014422  0.0012760    1.130 0.258382
hour_f3                         -0.0038601  0.0013202   -2.924 0.003456 **
hour_f4                         -0.0198047  0.0013846  -14.304  < 2e-16 ***
hour_f5                          0.0160625  0.0015745   10.201  < 2e-16 ***
hour_f6                          0.0124417  0.0018294    6.801 1.04e-11 ***
hour_f7                          0.0268699  0.0020195   13.306  < 2e-16 ***
hour_f8                         -0.0116280  0.0027739   -4.192 2.77e-05 ***
hour_f9                         -0.1125968  0.0041173  -27.347  < 2e-16 ***
hour_f10                        -0.2520008  0.0046386  -54.327  < 2e-16 ***
hour_f11                        -0.3409569  0.0036614  -93.122  < 2e-16 ***
hour_f12                        -0.3013432  0.0023596 -127.707  < 2e-16 ***
hour_f13                        -0.2645329  0.0016336 -161.930  < 2e-16 ***
hour_f14                        -0.2293100  0.0014867 -154.241  < 2e-16 ***
hour_f15                        -0.1782284  0.0014611 -121.981  < 2e-16 ***
hour_f16                        -0.1180901  0.0013670  -86.384  < 2e-16 ***
hour_f17                        -0.0838511  0.0012650  -66.283  < 2e-16 ***
hour_f18                        -0.0466565  0.0012310  -37.903  < 2e-16 ***
hour_f19                        -0.0142044  0.0012230  -11.614  < 2e-16 ***
hour_f20                        -0.0079734  0.0012072   -6.605 3.98e-11 ***
hour_f21                        -0.0030407  0.0011902   -2.555 0.010623 *
hour_f22                        -0.0088022  0.0011653   -7.554 4.23e-14 ***
hour_f23                        -0.0042743  0.0011642   -3.671 0.000241 ***
dow_f1                          -0.0476582  0.0009249  -51.526  < 2e-16 ***
dow_f2                          -0.0613926  0.0009258  -66.312  < 2e-16 ***
dow_f3                          -0.0408346  0.0009047  -45.137  < 2e-16 ***
dow_f4                           0.0112013  0.0008681   12.903  < 2e-16 ***
dow_f5                           0.1147095  0.0008328  137.748  < 2e-16 ***
dow_f6                           0.1218560  0.0008463  143.985  < 2e-16 ***
district_start_f1               -0.2073884  0.0178752  -11.602  < 2e-16 ***
district_start_f10              -0.2172245  0.0182232  -11.920  < 2e-16 ***
district_start_f2               -0.0934031  0.0199226   -4.688 2.75e-06 ***
district_start_f3               -0.3227818  0.0178640  -18.069  < 2e-16 ***
district_start_f4               -0.2252413  0.0186391  -12.084  < 2e-16 ***
district_start_f5               -0.2243001  0.0178921  -12.536  < 2e-16 ***
district_start_f6                0.0207705  0.0418139    0.497 0.619375
district_start_f7               -0.2860714  0.0183947  -15.552  < 2e-16 ***
district_start_f8                0.0090387  0.0181166    0.499 0.617838
district_start_f9               -0.2605560  0.0178519  -14.595  < 2e-16 ***
district_start_fNone             0.5020058  0.3069281    1.636 0.101927
vehicle_typescooter:log_distance -0.0626017  0.0010629  -58.897  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.5307 on 5485960 degrees of freedom
Multiple R-squared:  0.5508,     Adjusted R-squared:  0.5508
F-statistic: 1.564e+05 on 43 and 5485960 DF,  p-value: < 2.2e-16
```

Figure D3

Appendix E

Full R code

```
# Load packages

```{r, load packages}
# load packages
library(readr)
library(dplyr)
library(ggplot2)
library(gridExtra)
library(car)
library(MASS)
library(broom)
```
```{r, read data}
trips_raw <-
    read_csv("https://data.austintexas.gov/api/views/7d8e-dm7r/rows.csv?accessType=DOWNLOAD")
```
Preprocessing and EDA

** Preprocessing data**
```{r, preprocessing1}
################################################################################
## Cleaning data
# Filter for data of 2019 (year), get rid of NA's and negative values,
# select essential predictors that will be utilized
################################################################################

trips_2019 <- trips_raw %>%
  filter(
    `Year (US/Central)` == 2019,
    `Vehicle Type` %in% c("bicycle", "scooter"),
    !is.na(`Trip Distance`),
    !is.na(`Trip Duration`),
```
```

```
    `Trip Distance` > 0,
    `Trip Duration` > 0
  ) %>%
  select(
    `Vehicle Type`, `Trip Distance`, `Trip Duration`,
    `Hour`, `Day of Week`, `Council District (Start)`
  )


saveRDS(trips_2019, "trips_2019.rds")
```

```{r, preprocessing2}
################################################################################
## Change predictors from characters to factors
################################################################################
trips_2019_nolog <- trips_2019 %>%
  mutate(
    vehicle_type = factor(`Vehicle Type`),
    hour_f = factor(Hour),
    dow_f = factor(`Day of Week`),
    district_start_f = factor(`Council District (Start)`)
  )

saveRDS(trips_2019_nolog, "trips_2019_nolog.rds")
```

```{r, echo=FALSE}
trips_2019 <- readRDS("trips_2019.rds")
trips_2019_nolog <- readRDS("trips_2019_nolog.rds")
```

** Raw EDA**

```{r, 1st raw EDA}
```

```
################################################################################
## EDA of preprocessed raw data (no transformation applied)
# The preprocessed dataset is large, so take a random subset and explore the data
# Check predictor (vehicle type) with response (trip distance, trip duration)
################################################################################

# Create sample of 100,000
set.seed(500)
trips_sample_nolog <- trips_2019_nolog %>%
  sample_n(100000)

# Trip distance histogram
dis_hist <-
  ggplot(trips_sample_nolog, aes(x = `Trip Distance`)) +
  geom_histogram(bins =100) +
  labs(x = "Trip␣Distance␣(meters)", y = "Count",
       title = "Distribution␣of␣Trip␣Distance␣(raw)")

# Trip duration histogram
dur_hist <-
  ggplot(trips_sample_nolog, aes(x = `Trip Duration`)) +
  geom_histogram(bins = 100) +
  labs(x = "Trip␣duration␣(seconds)", y = "Count",
       title = "Distribution␣of␣Trip␣Duration␣(raw)")

# boxplot (x:vehicle_type / y:log_distance)
dis_box <-
  ggplot(trips_sample_nolog, aes(x = vehicle_type, y = `Trip Distance`)) +
  geom_boxplot() +
  labs(x = "Vehicle␣type", y = "Trip␣distance␣(meters)",
       title = "Trip␣Distance␣by␣Vehicle␣Type␣(raw)")

# boxplot (x:vehicle_type / y:log_duration)
dur_box <-
```

```
  ggplot(trips_sample_nolog, aes(x = vehicle_type, y = `Trip Duration`)) +
  geom_boxplot() +
  labs(x = "Vehicle type", y = "Trip duration (seconds)",
       title = "Trip Duration by Vehicle Type (raw)")

grid.arrange(dis_hist, dur_hist, dis_box, dur_box, nrow=2, ncol=2)
```

```{r, additional 1st EDA}
################################################################################
## EDA of preprocessed raw data (no transformation applied)
# Check other predictors with response predictor (trip distance, trip duration)
################################################################################

# boxplot (x:day of week / y:log_distance)
dis_box_dof <-
  ggplot(trips_sample_nolog, aes(x = dow_f, y = `Trip Distance`)) +
  geom_boxplot() +
  labs(x = "day of week", y = "Trip distance (meters)",
       title = "Trip Distance by Day of Week (raw)")

# boxplot (x:day of week / y:log_duration)
dur_box_dof <-
  ggplot(trips_sample_nolog, aes(x = dow_f, y = `Trip Duration`)) +
  geom_boxplot() +
  labs(x = "day of week", y = "Trip duration (seconds)",
       title = "Trip Duration by Day of Week (raw)")

# boxplot (x:hour / y:log_distance)
dis_box_hour <-
  ggplot(trips_sample_nolog, aes(x = hour_f, y = `Trip Distance`)) +
  geom_boxplot() +
  labs(x = "Hour", y = "Trip distance (meters)",
       title = "Trip Distance by Hours (raw)")
```

```
134
135 # boxplot (x:hour / y:log_duration)
136 dur_box_hour <- ggplot(trips_sample_nolog, aes(x = hour_f, y = `Trip Duration`)) +
137   geom_boxplot() +
138   labs(x = "Hour", y = "Trip duration (seconds)",
139        title = "Trip Duration by Hours (raw)")
140
141 # boxplot (x: Council District (Start) / y:log_distance)
142 dis_box_district <-
143   ggplot(trips_sample_nolog, aes(x = district_start_f, y = `Trip Distance`)) +
144   geom_boxplot() +
145   labs(x = "Council District (Start)", y = "Trip distance (meters)",
146        title = "Trip Distance by Council District (raw)")
147
148 # boxplot (x: Council District (Start) / y:log_duration)
149 dur_box_district <-
150   ggplot(trips_sample_nolog, aes(x = district_start_f, y = `Trip Duration`)) +
151   geom_boxplot() +
152   labs(x = "Council District (Start)", y = "Trip duration (seconds)",
153        title = "Trip Duration by Council District (raw)")
154
155 grid.arrange(dis_box_dof, dur_box_dof, dis_box_hour, dur_box_hour, dis_box_district,
    dur_box_district, nrow=3, ncol=2)
156 ```
157
158 ```{r, Calculate IQR}
159 ################################################################################
160 ## Calculate IQR
161 # 1.5IQR for checking potential outliers, 3IQR for extreme outliers
162 ################################################################################
163
164 # Trip Distance(d), Trip Duration(t)
165 qd <- quantile(trips_2019$`Trip Distance`, probs = c(0.25, 0.75), na.rm = TRUE)
166 IQR_d <- qd[2] - qd[1]
```

```
167
168 qt <- quantile(trips_2019$`Trip Duration`, probs = c(0.25, 0.75), na.rm = TRUE)
169 IQR_t <- qt[2] - qt[1]
170
171 # 1.5*IQR standard (for potential outliers)
172 lower_d_1_5 <- qd[1] - 1.5 * IQR_d
173 upper_d_1_5 <- qd[2] + 1.5 * IQR_d
174 lower_t_1_5 <- qt[1] - 1.5 * IQR_t
175 upper_t_1_5 <- qt[2] + 1.5 * IQR_t
176
177 # 3*IQR standard (for extreme outliers)
178 lower_d_3 <- qd[1] - 3 * IQR_d
179 upper_d_3 <- qd[2] + 3 * IQR_d
180 lower_t_3 <- qt[1] - 3 * IQR_t
181 upper_t_3 <- qt[2] + 3 * IQR_t
182
183 # Distance must be positive(cutoff > 0)
184 lower_d_1_5 <- max(lower_d_1_5, 0)
185 lower_d_3 <- max(lower_d_3, 0)
186 lower_t_1_5 <- max(lower_t_1_5, 0)
187 lower_t_3 <- max(lower_t_3, 0)
188
189 qd; qt
190 IQR_d; IQR_t
191 lower_d_1_5; upper_d_1_5
192 lower_d_3; upper_d_3
193 lower_t_1_5; upper_t_1_5
194 lower_t_3; upper_t_3
195
196 outlier_table <- data.frame(
197   Measure = c("Trip␣Distance", "Trip␣Duration"),
198   Q1 = c(unname(qd[1]), unname(qt[1])),
199   Q3 = c(unname(qd[2]), unname(qt[2])),
200   IQR = c(unname(IQR_d), unname(IQR_t)),
```

```
  Lower_1_5 = c(unname(lower_d_1_5), unname(lower_t_1_5)),
  Upper_1_5 = c(unname(upper_d_1_5), unname(upper_t_1_5)),
  Lower_3 = c(unname(lower_d_3), unname(lower_t_3)),
  Upper_3 = c(unname(upper_d_3), unname(upper_t_3)),
  check.names = FALSE
)

outlier_table
```

```{r, remove outlier by 3 X IQR}

## Remove extreme outliers by 3*IQR
trips_2019_nolog <- trips_2019_nolog %>%
  filter(
    `Trip Distance` >= lower_d_3,
    `Trip Distance` <= upper_d_3,
    `Trip Duration` >= lower_t_3,
    `Trip Duration` <= upper_t_3
  )

trips_2019_nolog %>%
  summarise(
    n = n(),
    mean_dist = mean(`Trip Distance`),
    median_dist= median(`Trip Distance`),
    max_dist = max(`Trip Distance`),
    mean_dur = mean(`Trip Duration`),
    median_dur = median(`Trip Duration`),
    max_dur = max(`Trip Duration`)
  )

saveRDS(trips_2019_nolog, "trips_2019_nolog.rds")
```

```
235
236 ** Raw data EDA without outliers (IQR)**
237 ```{r, Raw data EDA without outliers (IQR)}
238 ###############################################################################
239 ## Remove extreme outliers by 3*IQR and check dataset
240 ###############################################################################
241
242 # Create sample of 100,000
243 set.seed(500)
244 trips_sample_nolog <- trips_2019_nolog %>%
245   sample_n(100000)
246
247 # Trip distance histogram
248 dis_hist <-
249   ggplot(trips_sample_nolog, aes(x = `Trip Distance`)) +
250   geom_histogram(bins =100) +
251   labs(x = "Trip␣Distance␣(meters)", y = "Count",
252        title = "Distribution␣of␣Trip␣Distance␣(raw)")
253
254 # Trip duration histogram
255 dur_hist <-
256   ggplot(trips_sample_nolog, aes(x = `Trip Duration`)) +
257   geom_histogram(bins = 100) +
258   labs(x = "Trip␣duration␣(seconds)", y = "Count",
259        title = "Distribution␣of␣Trip␣Duration␣(raw)")
260
261 # boxplot (x:vehicle_type / y:log_distance)
262 dis_box <-
263   ggplot(trips_sample_nolog, aes(x = vehicle_type, y = `Trip Distance`)) +
264   geom_boxplot() +
265   labs(x = "Vehicle␣type", y = "Trip␣distance␣(meters)",
266        title = "Trip␣Distance␣by␣Vehicle␣Type␣(raw)")
267
268 # boxplot (x:vehicle_type / y:log_duration)
```

```
dur_box <-
  ggplot(trips_sample_nolog, aes(x = vehicle_type, y = `Trip Duration`)) +
  geom_boxplot() +
  labs(x = "Vehicle type", y = "Trip duration (seconds)",
       title = "Trip Duration by Vehicle Type (raw)")

grid.arrange(dis_hist, dur_hist, dis_box, dur_box, nrow=2, ncol=2)
```

```

# Question 1.

** Raw model**
```{r, raw model with no transformation}

## Check raw model results with no transformation
model_raw_dis <- lm(`Trip Distance` ~ vehicle_type + hour_f + dow_f +
    district_start_f, data=trips_2019_nolog)
summary(model_raw_dis)
```

** Raw model diagnostics**
```{r, Diagnostics for raw model}

## Check nonlinearity, heteroscedasticity, normality of raw model

fitted_raw_dis <- fitted(model_raw_dis)
resid_raw_dis <- resid(model_raw_dis)

# par(mfrow = c(2, 1))

# Residual vs Fitted
plot(fitted_raw_dis, resid_raw_dis,
     xlab = "Fitted values", ylab = "Residuals",
     main = "Residuals vs Fitted (raw Trip Distance)")
```
```

```
abline(h = 0, lty = 2, col = "red")

# QQ plot
qqnorm(resid_raw_dis,
       main = "Normal Q-Q Plot (raw Trip Distance)")
qqline(resid_raw_dis, col = "red", lty = 2)

```

-1-Log transformation of **response variables**

```{r, log-transform response variables}
################################################################################
# In order to overcome nonlinearity, heteroscedasticity, normality
# try log transformation
################################################################################

## Mutate log-transform response (distance & duration)
trips_2019 <- trips_2019_nolog %>%
  mutate(
    log_distance = log(`Trip Distance`),
    log_duration = log(`Trip Duration`)
  )

saveRDS(trips_2019, "trips_2019.rds")
```

-1-EDA: Exploration of Data (Log transformed)

```{r, 2nd EDA}
# Create sample of 100,000
set.seed(500)
trips_sample_log <- trips_2019 %>%
  sample_n(100000)

# log distance histogram
```

```
dis_hist_log <-
  ggplot(trips_sample_log, aes(x = log_distance)) +
  geom_histogram(bins = 100) +
  labs(x = "log(Trip␣distance)", y = "Count",
       title = "Distribution␣of␣log(Trip␣Distance)")

# log duration histogram
dur_hist_log <-
  ggplot(trips_sample_log, aes(x = log_duration)) +
  geom_histogram(bins = 100) +
  labs(x = "log(Trip␣duration)", y = "Count",
       title = "Distribution␣of␣log(Trip␣Duration)")

# boxplot (x:vehicle_type / y:log_distance)
dis_box_log <-
  ggplot(trips_sample_log, aes(x = vehicle_type, y = log_distance)) +
  geom_boxplot() +
  labs(x = "Vehicle␣type", y = "log(Trip␣distance)",
       title = "log(Trip␣Distance)␣by␣Vehicle␣Type")

# boxplot (x:vehicle_type / y:log_duration)
dur_box_log <-
  ggplot(trips_sample_log, aes(x = vehicle_type, y = log_duration)) +
  geom_boxplot() +
  labs(x = "Vehicle␣type", y = "log(Trip␣duration)",
       title = "log(Trip␣Duration)␣by␣Vehicle␣Type")


grid.arrange(dis_hist_log, dur_hist_log, dis_box_log, dur_box_log, nrow=2, ncol=2)
```

```{r, additional 2nd EDA}
# boxplot (x:day of week / y:log_distance)
dis_box_dof_log <-
```

```
  ggplot(trips_sample_log, aes(x = dow_f, y = log_distance)) +
  geom_boxplot() +
  labs(x = "day of week", y = "Trip distance (meters)",
       title = "Trip Distance by Day of Week")

# boxplot (x:day of week / y:log_duration)
dur_box_dof_log <-
  ggplot(trips_sample_log, aes(x = dow_f, y = log_duration)) +
  geom_boxplot() +
  labs(x = "day of week", y = "Trip duration (seconds)",
       title = "Trip Duration by Day of Week")

# boxplot (x:hour / y:log_distance)
dis_box_hour_log <-
  ggplot(trips_sample_log, aes(x = hour_f, y = log_distance)) +
  geom_boxplot() +
  labs(x = "Hour", y = "Trip distance (meters)",
       title = "Trip Distance by Hours")

# boxplot (x:hour / y:log_duration)
dur_box_hour_log <-
  ggplot(trips_sample_log, aes(x = hour_f, y = log_duration)) +
  geom_boxplot() +
  labs(x = "Hour", y = "Trip duration (seconds)",
       title = "Trip Duration by Hours")

# boxplot (x: Council District (Start) / y:log_distance)
dis_box_district_log <-
  ggplot(trips_sample_log, aes(x = district_start_f, y = log_distance)) +
  geom_boxplot() +
  labs(x = "Council District (Start)", y = "Trip distance (meters)",
       title = "Trip Distance by Council District")

# boxplot (x: Council District (Start) / y:log_duration)
```

```
dur_box_district_log <-
  ggplot(trips_sample_log, aes(x = district_start_f, y = log_duration)) +
  geom_boxplot() +
  labs(x = "Council District (Start)", y = "Trip duration (seconds)",
       title = "Trip Duration by Council District")

grid.arrange(dis_box_dof_log, dur_box_dof_log, dis_box_hour_log, dur_box_hour_log,
    dis_box_district_log, dur_box_district_log, nrow=3, ncol=2)
```

-1-Modeling & Diagnostics: Log transformed model

```{r, log transformed model diagnostics}
################################################################################
## Q1. Log transformed Model
# Modeling Diagnose
################################################################################

# Modeling Log transformed model
model_log_dis <- lm(log_distance ~ vehicle_type + hour_f + dow_f + district_start_f,
               data = trips_2019)

# Residual vs Fitted
plot(fitted(model_log_dis), resid(model_log_dis),
     xlab = "Fitted values",
     ylab = "Residuals",
     main = "Residuals vs Fitted (log distance)")
abline(h = 0, lty = 2, col = "red")

# QQ plot
qqnorm(resid(model_log_dis),
       main = "Normal Q-Q Plot (log distance)")
qqline(resid(model_log_dis), col = "red", lty = 2)
```

```
**-2-Modeling & Diagnostics: Box-Cox model**
```{r, Boxcox Diagnostics}
################################################################################
## Q1. Boxcox Model
# Modeling Diagnose
################################################################################

# Modeling Boxcox
boxcox_lamda_dis <- boxcox(model_raw_dis, plotit = TRUE)

lambda_hat_dis <- boxcox_lamda_dis$x[which.max(boxcox_lamda_dis$y)]
lambda_hat_dis

trips_2019_nolog$y_boxcox_dis <- if (abs(lambda_hat_dis) < 1e-3) {
  log(trips_2019_nolog$`Trip Distance`)
} else {
  (trips_2019_nolog$`Trip Distance`^lambda_hat_dis - 1) / lambda_hat_dis
}

model_boxcox_dis <- lm(y_boxcox_dis ~ vehicle_type + hour_f + dow_f +
    district_start_f,
              data = trips_2019_nolog)


# Residual vs Fitted
# par(mfrow = c(2,1))
plot(fitted(model_boxcox_dis), resid(model_boxcox_dis),
     xlab = "Fitted␣values",
     ylab = "Residuals",
     main = "Residuals␣vs␣Fitted␣(boxcox␣distance)")
abline(h = 0, lty = 2, col = "red")

# QQ plot
qqnorm(resid(model_boxcox_dis),
```
```

```
      main = "Normal Q-Q Plot (boxcox distance)")
qqline(resid(model_boxcox_dis), col = "red", lty = 2)
```

```

**-3-Modeling & Diagnostics: Weighted Least Square (WLS) model**
```{r, WLS Diagnostics}
################################################################################
## Q1. WLS Model
# Modeling Diagnose
################################################################################

# Modeling WLS
f_hat <- fitted(model_log_dis)
e_hat <- resid(model_log_dis)

w_simple <- 1 / (f_hat^2 + 1e-8)

model_wls_dis <- lm(log_distance ~ vehicle_type + hour_f + dow_f + district_start_f,
                data = trips_2019,
                weights = w_simple)


# Residual vs Fitted
# par(mfrow = c(2,1))
plot(fitted(model_wls_dis), resid(model_wls_dis),
     xlab = "Fitted values",
     ylab = "Residuals",
     main = "Residuals vs Fitted (WLS distance)")
abline(h = 0, lty = 2, col = "red")

# QQ plot
qqnorm(resid(model_wls_dis),
       main = "Normal Q-Q Plot (WLS distance)")
qqline(na.omit(resid(model_wls_dis)), col = "red", lty = 2)
```
```

```
```

```{r}
#############################################################################################
## Q1. Huber Model
# Modeling Diagnose
#############################################################################################

# Modeling Huber
model_huber_dis <- rlm(log_distance ~ vehicle_type + hour_f + dow_f +
    district_start_f,
                data = trips_2019)

# Residual vs Fitted
plot(fitted(model_huber_dis), resid(model_huber_dis),
     xlab = "Fitted values", ylab = "Residuals (Huber)",
     main = "Residuals vs Fitted (Huber distance)")
abline(h = 0, col = "red", lty = 2)

# QQ plot
qqnorm(resid(model_huber_dis),
       main = "Normal Q-Q Plot (Huber distance)")
qqline(resid(model_huber_dis), col = "red", lty = 2)
```

-5 Check BIC for candidate models

```{r}
#############################################################################################
## Q1. Compare BIC of candidate models
# - model_raw_dis : distance (raw)
# - model_log_dis : log_distance
# - model_boxcox_dis : boxCox
# - model_wls_dis : WLS (log_distance)
# - model_huber_dis : Huber (log_distance, rlm)
```

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

## BIC Comparison table
info_dis <- data.frame(
  Model = c("OLS_raw", "OLS_log", "BoxCox_log", "WLS_log", "Huber_log"),
  BIC = c(BIC(model_raw_dis),
          BIC(model_log_dis),
          BIC(model_boxcox_dis),
          BIC(model_wls_dis),
          BIC(model_huber_dis))
)
info_dis
```

-6 Compare Coefficient table

```{r}
# Compare vehicle_type coefficients
coef_table_dis <- data.frame(
  Vehicle = c("bicycle␣(baseline)", "scooter"),
  OLS = c(0, coef(model_log_dis)["vehicle_typescooter"]),
  Boxcox = c(0, coef(model_boxcox_dis)["vehicle_typescooter"]),
  WLS = c(0, coef(model_wls_dis)["vehicle_typescooter"]),
  Huber = c(0, coef(model_huber_dis)["vehicle_typescooter"])
)

coef_table_dis
```

** Model-based outlier, influencial point check**

```{r, Check Model-based outliers & influencial points}
################################################################################
## Q1. Check Model-based outliers & influencial points
# Apply studentized residuals, leverage, Cook's Distance
```

```
571 ###############################################################################
572
573 ## Define numbers of observations and coefficients
574 n_dis <- nrow(trips_2019)
575 p_dis <- length(coef(model_log_dis))
576 alpha <- 0.05
577
578
579 ## Calculate Diagnostic measures
580 student_dis <- rstudent(model_log_dis) # externally studentized residuals
581 lev_dis <- hatvalues(model_log_dis) # leverage (hat values)
582 cook_dis <- cooks.distance(model_log_dis) # Cook's distance
583
584
585 ## Set Cutoff
586 # studentized residual Bonferroni cutoff
587 cutoff_student_dis <- qt(1 - alpha/(2*n_dis), df = model_log_dis$df.residual)
588
589 # leverage rule of thumb: 2*(p+1)/n
590 cutoff_lev_dis <- 2 * (p_dis + 1) / n_dis
591
592 # Cook's distance rule of thumb: 4/n
593 cutoff_cook_dis <- 4 / n_dis
594
595
596 ## Visualize results as a Table
597 diag_tbl_dis <- data.frame(
598   id = 1:n_dis,
599   student = student_dis,
600   leverage = lev_dis,
601   cook = cook_dis,
602   flag_student = abs(student_dis) > cutoff_student_dis,
603   flag_leverage = lev_dis > cutoff_lev_dis,
604   flag_cook = cook_dis > cutoff_cook_dis
```

````
)

# filtering only the points that might need to be considered
diag_tbl_dis$flag_any <- with(diag_tbl_dis,
                        flag_student | flag_leverage | flag_cook)
suspects_dis <- subset(diag_tbl_dis, flag_any)
suspects_dis[order(-suspects_dis$cook), ][1:20, ]
```

** Check Collinearity**
```{r, check collinearity (VIF)}
vif(model_log_dis)
```

** Log transformed model interpretation**
```{r}
################################################################################
## Q1: scooter effect summary table (log-distance model)
################################################################################
summary(model_log_dis)

q1_scooter_tab <- tidy(model_log_dis) %>%
  filter(term == "vehicle_typescooter") %>%
  transmute(
    term = "Scooter␣(vs␣bicycle)",
    estimate = estimate,
    std_error = std.error,
    ratio = exp(estimate),
    pct_change = 100 * (exp(estimate) - 1)
  )
q1_scooter_tab
```

# Question 2.
````
````

```
639
640 ** Raw model**
641
642 ```{r, raw model with no transformation}
643
644 ## Check raw model results with no transformation
645 model_raw_dur <- lm(
646   `Trip Duration` ~ vehicle_type + `Trip Distance` + hour_f + dow_f +
        district_start_f,
647   data = trips_2019_nolog
648 )
649 ```
650
651 ** Raw model diagnostics**
652
653 ```{r, Diagnostics for raw model}
654
655 ## Check nonlinearity, heteroscedasticity, normality of raw model
656
657 fitted_raw_dur <- fitted(model_raw_dur)
658 resid_raw_dur <- resid(model_raw_dur)
659
660 # par(mfrow = c(2, 1))
661
662 # Residual vs Fitted
663 plot(fitted_raw_dur, resid_raw_dur,
664      xlab = "Fitted␣values", ylab = "Residuals",
665      main = "Residuals␣vs␣Fitted␣(raw␣Trip␣Duration)")
666 abline(h = 0, lty = 2, col = "red")
667
668 # QQ plot
669 qqnorm(resid_raw_dur,
670        main = "Normal␣Q-Q␣Plot␣(raw␣Trip␣Duration)")
671 qqline(resid_raw_dur, col = "red", lty = 2)
```

```
672
673 ```
674
675 **-1-Modeling & Diagnostics: Log transformed model**
676 ```{r, log transformed model diagnostics}
677 ################################################################################
678 ## Q2: Log transformed model for duration
679 ## Response: log_duration
680 ## Predictors: vehicle_type + log_distance + hour_f + dow_f + district_start_f
681 ################################################################################
682
683 # Modeling Log transformed model (duration)
684 model_log_dur <- lm(
685   log_duration ~ vehicle_type + log_distance + hour_f + dow_f + district_start_f,
686   data = trips_2019
687 )
688
689 # Residual vs Fitted
690 plot(fitted(model_log_dur), resid(model_log_dur),
691      xlab = "Fitted␣values",
692      ylab = "Residuals",
693      main = "Residuals␣vs␣Fitted␣(log␣duration)")
694 abline(h = 0, lty = 2, col = "red")
695
696 # QQ plot
697 qqnorm(resid(model_log_dur),
698        main = "Normal␣Q-Q␣Plot␣(log␣duration)")
699 qqline(resid(model_log_dur), col = "red", lty = 2)
700 ```
701
702 **-2-Modeling & Diagnostics: Box-Cox model**
703 ```{r, Boxcox Diagnostics}
704 ################################################################################
705 ## Q2: Box-Cox model for duration
```

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

# Box-Cox for duration
boxcox_lambda_dur <- boxcox(model_raw_dur, plotit = TRUE)

lambda_hat_dur <- boxcox_lambda_dur$x[which.max(boxcox_lambda_dur$y)]
lambda_hat_dur

# Box-Cox transformed response for duration
trips_2019_nolog$y_boxcox_dur <- if (abs(lambda_hat_dur) < 1e-3) {
  log(trips_2019_nolog$`Trip Duration`)
} else {
  (trips_2019_nolog$`Trip Duration`^lambda_hat_dur - 1) / lambda_hat_dur
}

model_boxcox_dur <- lm(
  y_boxcox_dur ~ vehicle_type + `Trip Distance` + hour_f + dow_f + district_start_f,
  data = trips_2019_nolog
)

# Residual vs Fitted (Box-Cox duration)
plot(fitted(model_boxcox_dur), resid(model_boxcox_dur),
     xlab = "Fitted values",
     ylab = "Residuals",
     main = "Residuals vs Fitted (Box-Cox duration)")
abline(h = 0, lty = 2, col = "red")

# QQ plot
qqnorm(resid(model_boxcox_dur),
       main = "Normal Q-Q Plot (Box-Cox duration)")
qqline(resid(model_boxcox_dur), col = "red", lty = 2)
```

-3-Modeling & Diagnostics: Weighted Least Square (WLS) model

```r
```{r}
################################################################################
## Q2: WLS model for log_duration
################################################################################

# Fitted values & residuals from log-duration model
f_hat_dur <- fitted(model_log_dur)
e_hat_dur <- resid(model_log_dur)

# Simple variance function: var(e)  fitted^2 (same idea as distance)
w_simple_dur <- 1 / (f_hat_dur^2 + 1e-8)

model_wls_dur <- lm(
  log_duration ~ vehicle_type + log_distance + hour_f + dow_f + district_start_f,
  data = trips_2019,
  weights = w_simple_dur
)

# Residual vs Fitted (WLS duration)
plot(fitted(model_wls_dur), resid(model_wls_dur),
     xlab = "Fitted values",
     ylab = "Residuals",
     main = "Residuals vs Fitted (WLS duration)")
abline(h = 0, lty = 2, col = "red")

# QQ plot
qqnorm(resid(model_wls_dur),
       main = "Normal Q-Q Plot (WLS duration)")
qqline(resid(model_wls_dur), col = "red", lty = 2)
```

-4-Modeling & Diagnostics: Huber model
```{r}
################################################################################
```
```

```
## Q2: Huber robust regression for log_duration
################################################################################

library(MASS)

model_huber_dur <- rlm(
  log_duration ~ vehicle_type + log_distance + hour_f + dow_f + district_start_f,
  data = trips_2019
)

# Residual vs Fitted (Huber duration)
plot(fitted(model_huber_dur), resid(model_huber_dur),
     xlab = "Fitted values",
     ylab = "Residuals (Huber)",
     main = "Residuals vs Fitted (Huber, log duration)")
abline(h = 0, col = "red", lty = 2)

# QQ plot
qqnorm(resid(model_huber_dur),
       main = "Normal Q-Q Plot (Huber, log duration)")
qqline(resid(model_huber_dur), col = "red", lty = 2)
```

```

**-6 Check BIC for candidate models**

```{r}
################################################################################
## Compare BIC of candidate models
# - model_raw_dur : duration (raw)
# - model_log_dur : log_duration
# - model_boxcox_dur : boxCox
# - model_wls_dur : WLS (log_duration)
# - model_huber_dur : Huber (log_duration, rlm)
################################################################################
## BIC Comparison table
```
```

```
info_dur <- data.frame(
  Model = c("OLS_raw", "OLS_log", "BoxCox_log", "WLS_log", "Huber_log"),
  BIC = c(BIC(model_raw_dur),
          BIC(model_log_dur),
          BIC(model_boxcox_dur),
          BIC(model_wls_dur),
          BIC(model_huber_dur))
)
info_dur
```

-7 Compare Coefficient table (duration, scooter effect)

```{r}
################################################################################
## Q2: Compare vehicle_type coefficients across OLS / WLS / Huber (duration)
################################################################################

coef_table_dur <- data.frame(
  Vehicle = c("bicycle␣(baseline)", "scooter"),
  OLS = c(0, coef(model_log_dur)["vehicle_typescooter"]),
  BoxCox = c(0, coef(model_boxcox_dur)["vehicle_typescooter"]),
  WLS = c(0, coef(model_wls_dur)["vehicle_typescooter"]),
  Huber = c(0, coef(model_huber_dur)["vehicle_typescooter"])
)

coef_table_dur
```

** Check Plausible Interaction models (optional candidates)**

```{r, Check plausible interaction models}
################################################################################
# model_int1: log_duration ~ log_distance * vehicle_type + hour_f + dow_f +
    district_start_f
```

```
# model_int2: log_duration ~ log_distance + vehicle_type * hour_f + dow_f +
    district_start_f
# model_int3: log_duration ~ log_distance + hour_f + dow_f + vehicle_type *
    district_start_f
################################################################################

# Vehicle_type x log_distance
model_int1 <- update(model_log_dur, . ~ . + vehicle_type:log_distance)

# Vehicle_type x hour_f
model_int2 <- update(model_log_dur, . ~ . + vehicle_type:hour_f)

# Vehicle_type x district_start_f
model_int3 <- update(model_log_dur, . ~ . + vehicle_type:district_start_f)

info_dur_interaction <- data.frame(
  Model = c("main_log", "Veh_dis_log", "Veh_hour_log", "Veh_district_log"),
  BIC = c(BIC(model_log_dur),
          BIC(model_int1),
          BIC(model_int2),
          BIC(model_int3))
)
info_dur_interaction
```

```

** Fit log interaction model**
```{r, fit log interaction model}
################################################################################
## Log duration model with interaction
## Response : log_duration
## Predictors : vehicle_type * log_distance + hour_f + dow_f + district_start_f
################################################################################
model_log_interaction <- lm(log_duration ~ vehicle_type * log_distance +
    hour_f + dow_f + district_start_f, data = trips_2019)
```
```

```
```

** Fit log interaction model**
```{r, log interaction model diagnostics}

# Residual vs Fitted
plot(fitted(model_log_interaction), resid(model_log_interaction),
     xlab = "Fitted values",
     ylab = "Residuals",
     main = "Residuals vs Fitted (log duration, interaction model)")
abline(h = 0, lty = 2, col = "red")

# Q-Q plot
qqnorm(resid(model_log_interaction),
       main = "Normal Q-Q Plot (log duration, interaction model)")
qqline(resid(model_log_interaction), col = "red", lty = 2)
```

** Check structure of log transformed model**
```{r, Check structure of log transformed model}

## partial residual plot
partial_resid <- residuals(model_log_interaction) +
                coef(model_log_interaction)["log_distance"] * trips_2019$log_distance

plot(trips_2019$log_distance, partial_resid,
     xlab = "log_distance",
     ylab = "Partial residuals for log_distance",
     main = "Partial residual plot for log_distance",
     pch = 16,
     cex = 0.3,
     col = adjustcolor("black", alpha.f = 0.2))
abline(a = 0,
       b = coef(model_log_interaction)["log_distance"],
```

```
      lwd = 3, col ="red")
lines(lowess(trips_2019$log_distance, partial_resid),lwd = 2, lty = 2, col = "blue")
```

```

** Model-based outlier, influencial point check**
```{r}
################################################################################
## Q2: Check Model-based outliers & influential points (duration model)
################################################################################

## Number of observations and coefficients
n_dur <- nrow(trips_2019)
p_dur <- length(coef(model_log_interaction))
alpha <- 0.05

## Diagnostic measures
student_dur <- rstudent(model_log_interaction) # externally studentized residuals
lev_dur <- hatvalues(model_log_interaction) # leverage (hat values)
cook_dur <- cooks.distance(model_log_interaction) # Cook's distance

## Cutoffs
cutoff_student_dur <- qt(1 - alpha/(2*n_dur), df = model_log_interaction$df.residual)
cutoff_lev_dur <- 2 * (p_dur + 1) / n_dur
cutoff_cook_dur <- 4 / n_dur

## Table of diagnostics
diag_tbl_dur <- data.frame(
  id = 1:n_dur,
  student = student_dur,
  leverage = lev_dur,
  cook = cook_dur,
  flag_student = abs(student_dur) > cutoff_student_dur,
  flag_leverage = lev_dur > cutoff_lev_dur,
  flag_cook = cook_dur > cutoff_cook_dur
```
```

```
)

diag_tbl_dur$flag_any <- with(diag_tbl_dur,
                              flag_student | flag_leverage | flag_cook)

suspects_dur <- subset(diag_tbl_dur, flag_any)
suspects_dur[order(-suspects_dur$cook), ][1:20, ]
```

** Check Collinearity**

```{r}
################################################################################
## Q2: Check collinearity (VIF) for duration model
################################################################################

library(car)
vif(model_log_interaction)
```

** Log transformed model interpretation**

```{r}
################################################################################
## Q2: Scooter vs Bicycle duration effect
## (1) Main-effects model: constant effect across distance
## (2) Interaction model: effect depends on log_distance via 1 + 3 x
################################################################################

# Summary of main-effects & interaction models
summary(model_log_dur) # main-effects
summary(model_log_interaction) # interaction

# mean log_distance
mean_log_dist <- mean(trips_2019$log_distance, na.rm = TRUE)
```

```
# Scooter effects from main-effects model:  = 1
coef_main <- coef(model_log_dur)
a1_main <- unname(coef_main["vehicle_typescooter"])

delta_main <- a1_main
ratio_main <- exp(delta_main)

# Scooter effects from Interaction model: (x) = 1 + 3 x
coef_int <- coef(model_log_interaction)
a1_int <- unname(coef_int["vehicle_typescooter"]) # 1
a3_int <- unname(coef_int["vehicle_typescooter:log_distance"]) # 3

delta_int_at_mean <- a1_int + a3_int * mean_log_dist
ratio_int_at_mean <- exp(delta_int_at_mean)

# Table of results
results_tbl <- data.frame(
  Model = c("Main-effects␣(no␣interaction)",
            "Interaction␣(vehicle_type␣␣log_distance)"),
  a1_scooter = c(a1_main, a1_int),
  a3_interaction = c(NA, a3_int),
  log_distance_used = c(NA, mean_log_dist),
  delta_scooter_log = c(delta_main, delta_int_at_mean),
  ratio_exp_delta = c(ratio_main, ratio_int_at_mean),
  pct_diff = 100 * (c(ratio_main, ratio_int_at_mean) - 1)
)


results_tbl_round <- within(results_tbl, {
  a1_scooter <- round(a1_scooter, 6)
  a3_interaction <- round(a3_interaction, 6)
  log_distance_used <- round(log_distance_used, 6)
  delta_scooter_log <- round(delta_scooter_log, 6)
  ratio_exp_delta <- round(ratio_exp_delta, 6)
```

```
  pct_diff <- round(pct_diff, 3)
})

results_tbl_round
```