Assignment2

Ben Lu

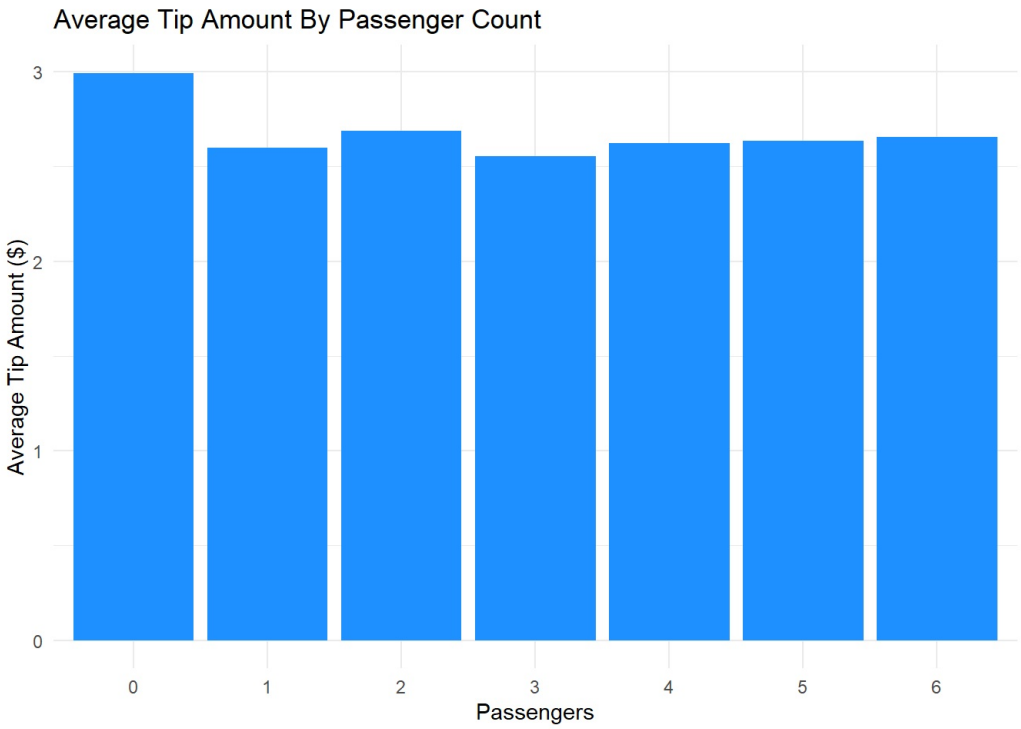
2024-08-13

Explore Original Data

```
# read csv, filter for credit card payment
week2_data <- read_csv("week2.csv", show_col_types = FALSE) %>% filter(payment_type == 1)
week4_data <- read_csv("week4.csv", show_col_types = FALSE) %>% filter(payment_type == 1)
zone_data <- read_csv("taxi+_zone_lookup.csv", show_col_types = FALSE)
latitude_map = read_csv("taxilatlong.csv", show_col_types = FALSE)

# sample 10% of original week 2 data for exploration
my_sample <- sample(nrow(week2_data), round(nrow(week2_data) * 0.1))
week2_sample <- week2_data[my_sample,]

# join rows for PU and DO borough and service zones, latitude and longitude
week2_sample <- week2_sample %>%
  left_join(zone_data, by = c("PULocationID" = "LocationID")) %>%
  rename(PU_Borough = Borough,
         PU_Zone = Zone,
         PU_service_zone = service_zone)

week2_sample <- week2_sample %>%
  left_join(zone_data, by = c("DOLocationID" = "LocationID")) %>%
  rename(DO_Borough = Borough,
         DO_Zone = Zone,
         DO_service_zone = service_zone)

week2_sample <- week2_sample %>%
  left_join(latitude_map, by = c("PULocationID" = "LocationID"))

# summarise each variable and inspect for outliers
week2_sample %>% summary()
```

```
##     VendorID     tpep_pickup_datetime
##  Min.   :1.000   Min.   :2017-02-06 00:00:01.00
##  1st Qu.:1.000   1st Qu.:2017-02-07 20:06:27.00
##  Median :2.000   Median :2017-02-09 19:38:28.00
##  Mean   :1.556   Mean   :2017-02-09 16:02:01.33
##  3rd Qu.:2.000   3rd Qu.:2017-02-11 11:55:08.00
##  Max.   :2.000   Max.   :2017-02-12 23:59:52.00
##
##  tpep_dropoff_datetime            passenger_count trip_distance
##  Min.   :2017-02-06 00:03:00.00   Min.   :0.00    Min.   : 0.000
##  1st Qu.:2017-02-07 20:19:50.00   1st Qu.:1.00    1st Qu.: 1.000
##  Median :2017-02-09 19:49:23.00   Median :1.00    Median : 1.690
##  Mean   :2017-02-09 16:17:25.94   Mean   :1.62    Mean   : 2.889
##  3rd Qu.:2017-02-11 12:09:59.00   3rd Qu.:2.00    3rd Qu.: 3.020
##  Max.   :2017-02-13 20:28:50.00   Max.   :6.00    Max.   :56.200
##
##    RatecodeID     store_and_fwd_flag  PULocationID    DOLocationID
##  Min.   : 1.000   Length:151493      Min.   :  1.0   Min.   :  1.0
##  1st Qu.: 1.000   Class :character   1st Qu.:114.0   1st Qu.:107.0
##  Median : 1.000   Mode  :character   Median :162.0   Median :162.0
##  Mean   : 1.046                      Mean   :163.8   Mean   :161.3
##  3rd Qu.: 1.000                      3rd Qu.:233.0   3rd Qu.:234.0
##  Max.   :99.000                      Max.   :265.0   Max.   :265.0
##
##   payment_type  fare_amount         extra           mta_tax
##  Min.   :1     Min.   :-52.00   Min.   :0.0000   Min.   :-0.5000
##  1st Qu.:1     1st Qu.:  6.50   1st Qu.:0.0000   1st Qu.: 0.5000
##  Median :1     Median :  9.50   Median :0.0000   Median : 0.5000
##  Mean   :1     Mean   : 12.77   Mean   :0.3402   Mean   : 0.4978
##  3rd Qu.:1     3rd Qu.: 14.50   3rd Qu.:0.5000   3rd Qu.: 0.5000
##  Max.   :1     Max.   :325.00   Max.   :4.5000   Max.   : 0.5000
##
##    tip_amount       tolls_amount     improvement_surcharge  total_amount
##  Min.   :-10.560   Min.   : 0.0000   Min.   :-0.3000       Min.   :-65.31
##  1st Qu.:  1.290   1st Qu.: 0.0000   1st Qu.: 0.3000       1st Qu.:  9.30
##  Median :  2.000   Median : 0.0000   Median : 0.3000       Median : 12.36
##  Mean   :  2.614   Mean   : 0.2986   Mean   : 0.2999       Mean   : 16.83
##  3rd Qu.:  2.960   3rd Qu.: 0.0000   3rd Qu.: 0.3000       3rd Qu.: 18.35
##  Max.   :202.000   Max.   :52.5000   Max.   : 0.3000       Max.   :387.30
##
##  congestion_surcharge airport_fee     PU_Borough          PU_Zone
##  Mode:logical         Mode:logical   Length:151493      Length:151493
##  NA's:151493          NA's:151493    Class :character   Class :character
##                                      Mode  :character   Mode  :character
##
##
##
##
##  PU_service_zone     DO_Borough          DO_Zone          DO_service_zone
##  Length:151493      Length:151493      Length:151493      Length:151493
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##       long             lat
##  Min.   :-74.19   Min.   :40.53
##  1st Qu.:-73.99   1st Qu.:40.73
##  Median :-73.98   Median :40.75
##  Mean   :-73.98   Mean   :40.75
##  3rd Qu.:-73.97   3rd Qu.:40.77
##  Max.   :-73.74   Max.   :40.90
##  NA's   :2392     NA's   :2392
```

```
# plot passenger count and average tip amount
week2_sample %>%
  group_by(passenger_count) %>%
  summarise(average_tip = mean(tip_amount)) %>%
  ggplot(aes(x=factor(passenger_count, levels = 0:6), y=average_tip)) +
  geom_bar(stat = "identity",  fill = "dodgerblue") +
  labs(title = "Average Tip Amount By Passenger Count", x = "Passengers", y = "Average Tip Amount ($)")   + theme
_minimal()
```

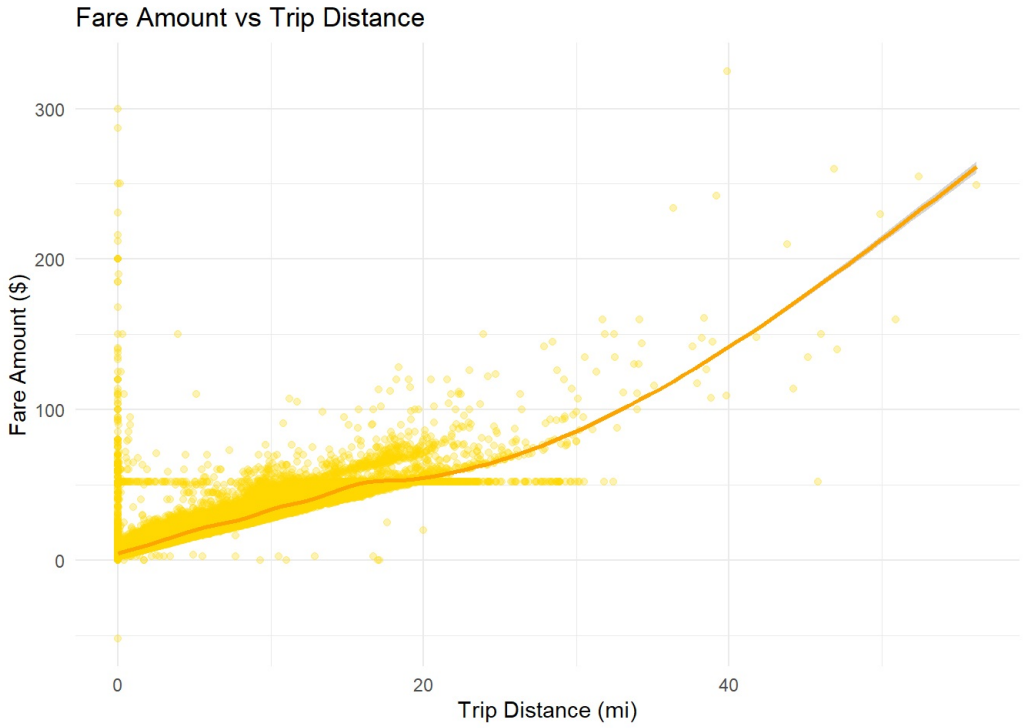



```
# plot tip amount by distance traveled
week2_sample %>%
  ggplot(aes(x = trip_distance, y = tip_amount)) +
  geom_point(alpha = 0.3, colour = "gold") +
  geom_smooth(colour = "orange") +
  labs(title = "Tip Amount vs Distance Travelled", x = "Trip Distance (mi)", y = "Tip Amount ($)") + theme_minima
l()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

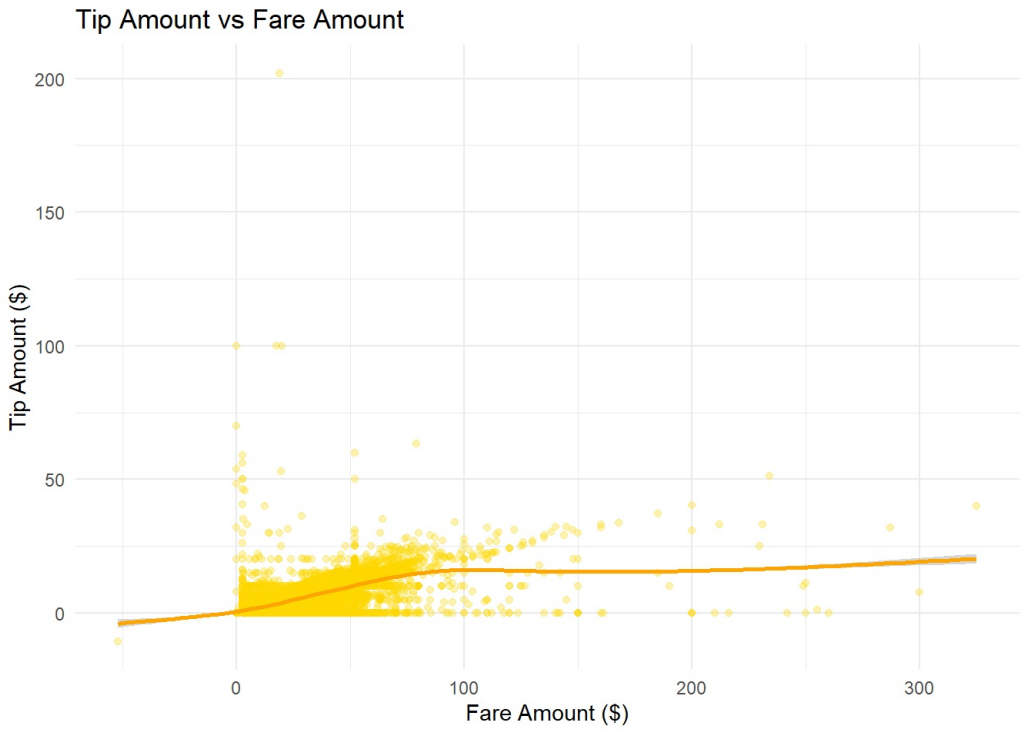

```
# plot fare amount by distance traveled
week2_sample %>%
  ggplot(aes(x = trip_distance, y = fare_amount)) +
  geom_point(alpha = 0.3, colour = "gold") +
  geom_smooth(colour = "orange") +
  labs(title = "Fare Amount vs Trip Distance", x = "Trip Distance (mi)", y = "Fare Amount ($)") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



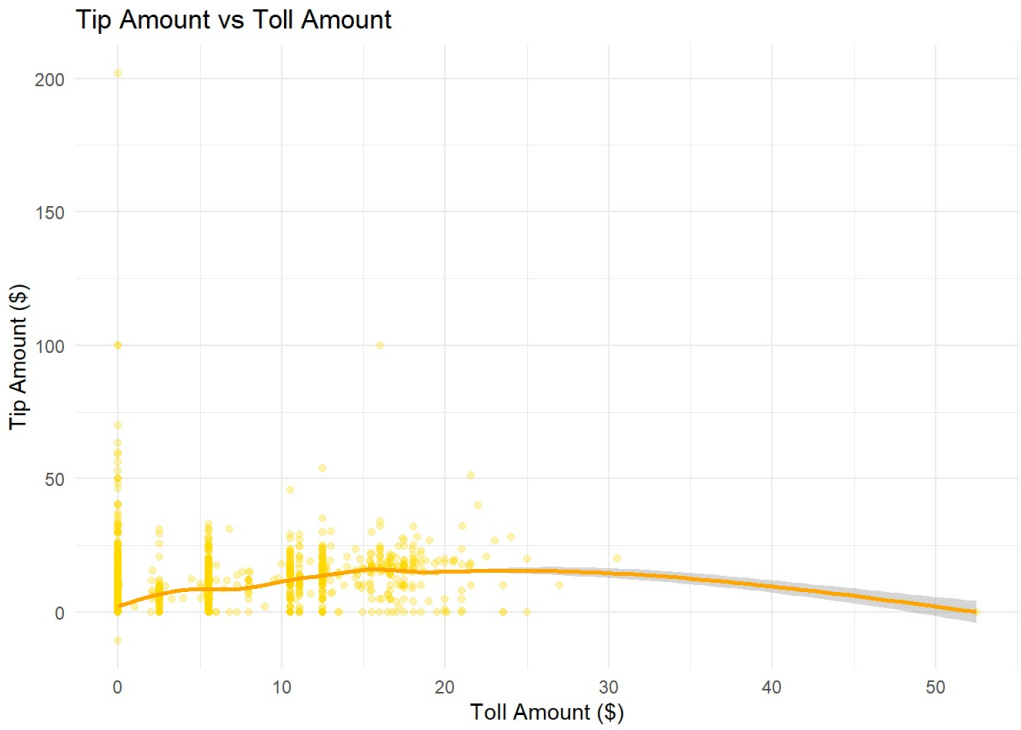

```
# plot tip amount by fare amount
week2_sample %>%
  ggplot(aes(x = fare_amount, y = tip_amount)) +
  geom_point(alpha = 0.3, colour = "gold") +
  geom_smooth(colour = "orange") +
  labs(title = "Tip Amount vs Fare Amount", x = "Fare Amount ($)", y = "Tip Amount ($)") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

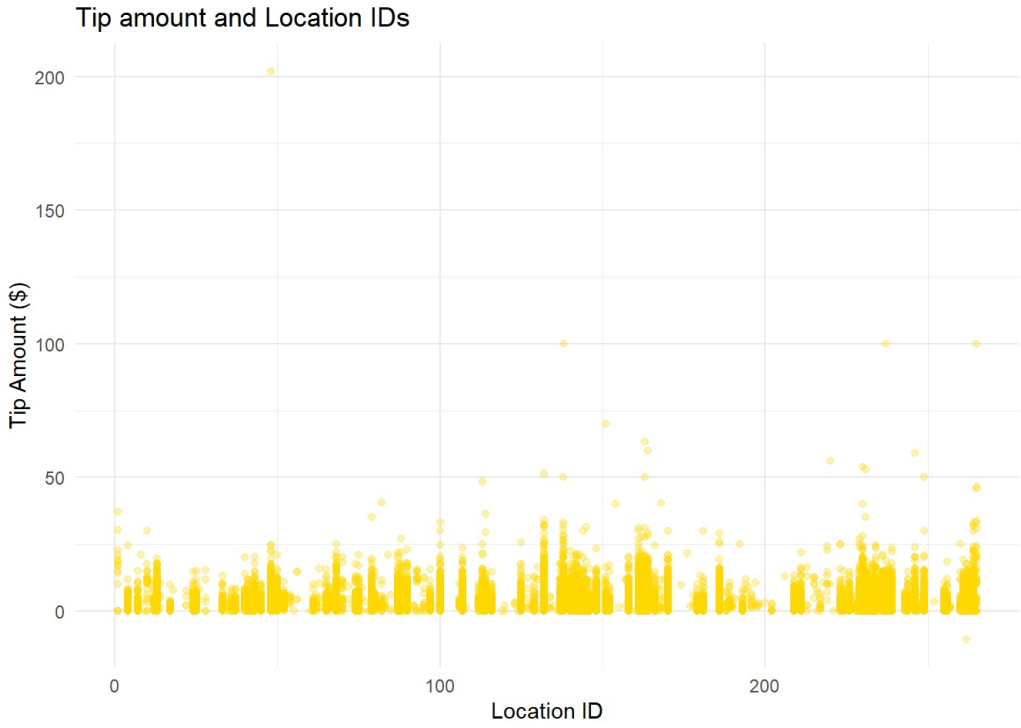



```
# plot toll tip amount and toll amount
week2_sample %>%
  ggplot(aes(x = tolls_amount, y = tip_amount)) +
  geom_point(alpha = 0.3, colour = "gold") +
  geom_smooth(colour = "orange") +
  labs(title = "Tip Amount vs Toll Amount", x = "Toll Amount ($)", y = "Tip Amount ($)") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
# tip amounts by pickup location ID
week2_sample %>%
  ggplot(aes(x = PULocationID, y = tip_amount)) +
  geom_point(alpha = 0.3, colour = "gold") +
  labs(title = "Tip amount and Location IDs", x = "Location ID", y = "Tip Amount ($)") +
  theme_minimal()
```




```
# generate summaries of average tip by different pickup locations
week2_sample %>%
  group_by(PU_Borough) %>%
  summarise(average_tip = mean(tip_amount))
```

```
## # A tibble: 7 × 2
##   PU_Borough    average_tip
##   <chr>               <dbl>
## 1 Bronx                4.46
## 2 Brooklyn             2.89
## 3 EWR                 16.0
## 4 Manhattan            2.34
## 5 Queens               7.69
## 6 Staten Island       13.9
## 7 Unknown              2.91
```

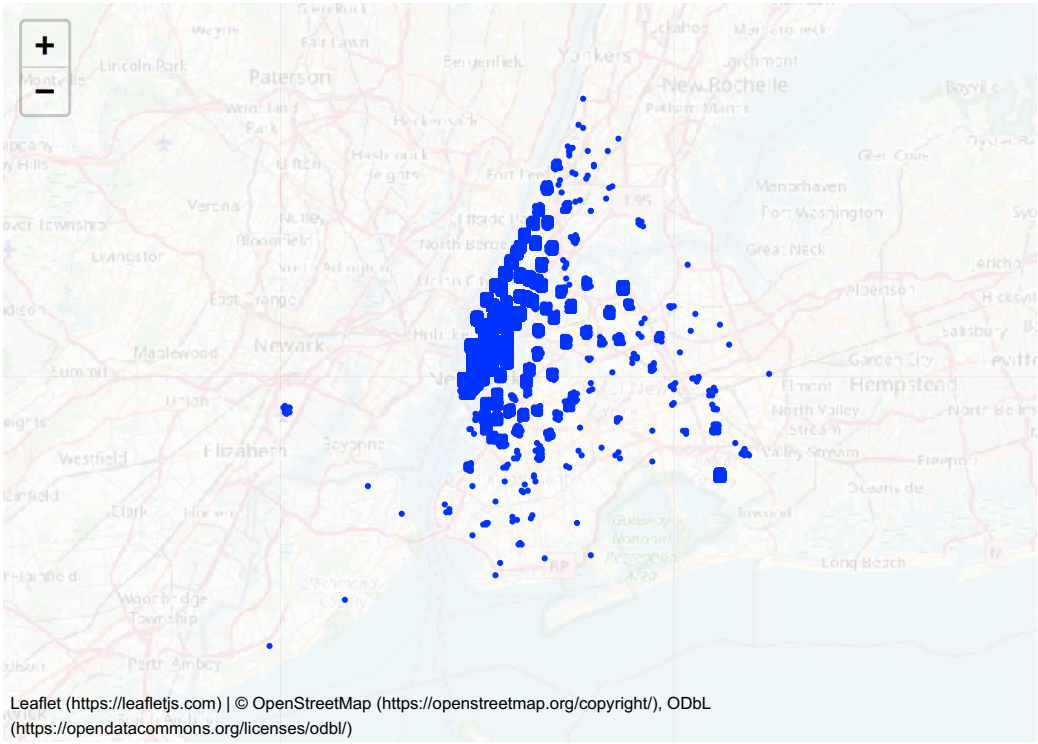
```
week2_sample %>%
  group_by(PU_service_zone) %>%
  summarise(average_tip = mean(tip_amount))
```

```
## # A tibble: 5 × 2
##   PU_service_zone average_tip
##   <chr>                 <dbl>
## 1 Airports               8.41
## 2 Boro Zone              2.75
## 3 EWR                   16.0
## 4 N/A                    2.91
## 5 Yellow Zone            2.34
```

```
week2_sample %>%
  group_by(PU_Zone) %>%
  summarise(average_tip = mean(tip_amount)) %>% slice(1:10)
```

```
## # A tibble: 10 × 2
##    PU_Zone           average_tip
##    <chr>                   <dbl>
##  1 Alphabet City            2.27
##  2 Astoria                  2.39
##  3 Astoria Park            12.0
##  4 Auburndale               0
##  5 Baisley Park            10.7
##  6 Bath Beach              10
##  7 Battery Park             3.78
##  8 Battery Park City        3.09
##  9 Bay Ridge                5.58
## 10 Bedford                  1.82
```

```
# create slight variations in locations and plot pickup locations on map
week2_sample <- week2_sample %>%
  rename(pickup_latitude = lat,  pickup_longitude = long) %>%
  mutate(pickup_latitude = jitter(pickup_latitude, amount = .003),
         pickup_longitude = jitter(pickup_longitude, amount = .003))

leaflet(week2_sample) %>%
  addTiles() %>%
  addCircleMarkers(~pickup_longitude,~pickup_latitude, radius=2,stroke = FALSE, opacity=1, fillOpacity =1)
```



Explore summary data

Part of our introductory data exploration involves joining the original week 2 data and location data tables together, and generating a summary to give us an overall view of all the original variables. Upon inspection of our summary, we notice some unusual outliers such as:

1. Negative fare amounts, tip amounts and total amounts.
2. Passenger counts of 0 which is an error, or 6 which exceed the 5 passengers limit in NYC.
3. Trip distances being 0 miles or exceeding 25 miles.
4. Location ID's being outside of the location data table, such as 265.
5. Extra charges of 4.5 when the amounts can only be either 0,0.5 or 1.
6. Improvement surcharge being negative.
7. Missing NA values in other variables such as congestion charges, airport fees, location latitudes and longitudes.

Explore how tip amounts are influenced by passengers, fare amounts and toll amounts

In observing the bar chart of the average amount of tips by passenger count, we noticed that the average tip of all passenger counts to be very similar, and there no signs of any major differences in the tip amounts by the number of passengers. Hence, we would consider excluding this variable when constructing our model.

Our dot plot of the tip amount and the distance traveled in miles shows a slightly increasing trend which indicates that as the distance traveled increases, passengers generally tend to tip slightly more. However, the trend line begins to flatten around 30 miles and curve slightly downwards.

Our dot plot of the fare amount and the distance traveled shows an interesting relationship of how the taxi meter calculates the fare amount based in distance. We can see a trend that the fare amount increases linearly as the distance traveled reaches 17 (km), then from that point onwards it remains flat until 20 (km) before the trend increases linearly again. This shows that the taxi meter calculates the fare amount mostly based on a linear relationship.

Out dot plot of the tip amount and fare amount shows that the tip amount increases linearly as the fare amount approaches $80, from which point the trend flattens. [We could also see that the trend line extended left and downwards which are due to outliers]

Our dot plot of the tip amount and the toll amount shows a parabola, as the toll amount increases, the tip amount gradually increases but eventually decreases back to 0.

As part of our introductory data exploration, we found that passenger counts has little influence on the amount they tip. The important variables we should consider are distance traveled, fare amounts and toll amounts, in which cases the tip amount has a noticeable relationship. These variables should be included when constructing a model.

Explore how tip amounts are influenced by pickup locations

In observing our dot plot of the tip amounts by different pickup location ID's, it appears that some groups of location IDs have slightly higher or lower tips than other groups. [We also created a map to display the pickup locations]

We generated summaries for the average tip amounts by different pickup boroughs, service zones and zones and it appears that the average tip amount vary substantially in different boroughs, service zones and zones. This suggests that tipping behavior is influenced by different areas.

When deciding on a location variable to build our model, we considered that location ID's and zones should be excluded as it they have too many categories which regsubsets cannot handle. Out of all these location variables, we prefer to use boroughs as it has 6 categories (excluding "Unknown") which is optimal.

Other variables we considered removing are vendor ID's and store and forward flag as these do not influence the passenger behavior or tip amounts. MTA tax and improvement surcharge are fixed variables, and airport fees are an empty column, hence they will not be used for building our model.

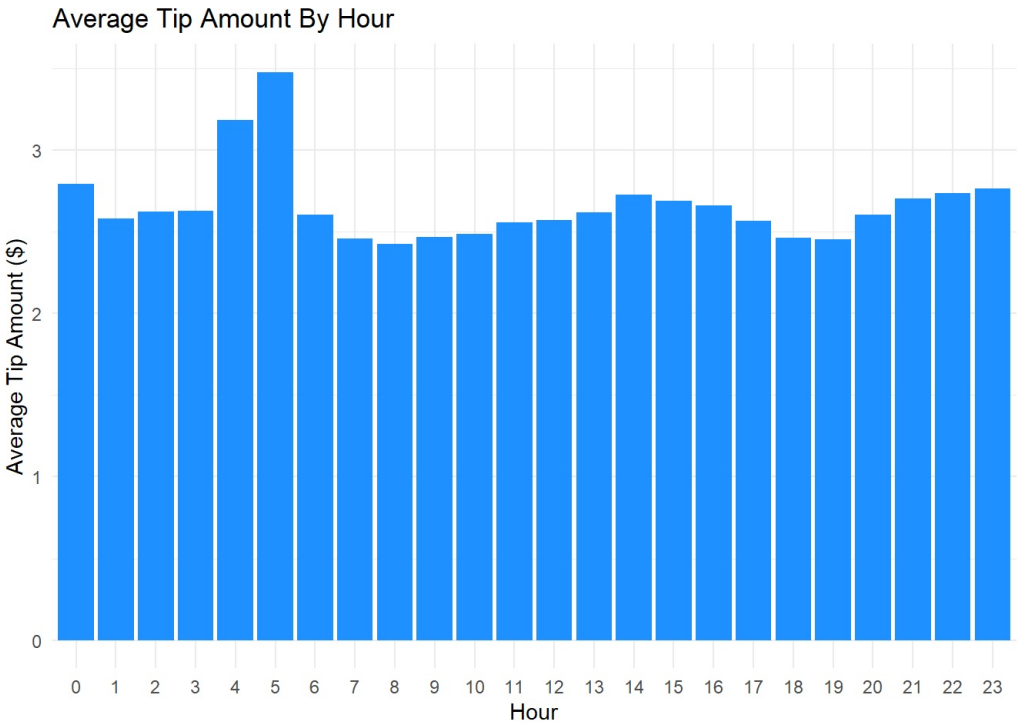
Data Cleaning and Creating New Variables

```
# Join rows for PU and DO borough and service zones for week 2 data
week2_data <- week2_data %>%
  left_join(zone_data, by = c("PULocationID" = "LocationID")) %>%
  rename(PU_Borough = Borough,
         PU_Zone = Zone,
         PU_service_zone = service_zone) %>%
  left_join(zone_data, by = c("DOLocationID" = "LocationID")) %>%
  rename(DO_Borough = Borough,
         DO_Zone = Zone,
         DO_service_zone = service_zone)

# Data cleaning - filter out any outliers such as:
# the location id's not belonging to the latitude table,
# tip amounts being negative or greater than 50
# passenger count being 0 or greater than 5
# extra charges being any other than 0, 0.5 or 1.
# fare amounts being negative or exceeding 100
# trip distances being greater than 25 miles
# tolls amount being greater than 10 dollars is unusual
# any erroneous location data such as "Unknown", "NA" or "N/A"
```

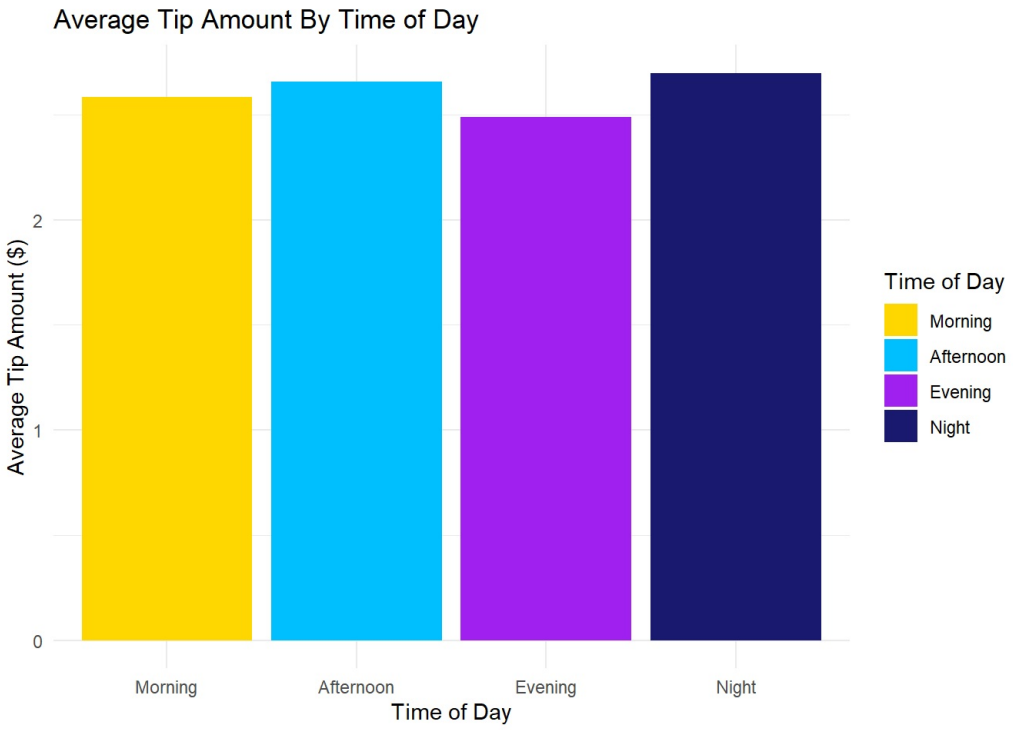
```
week2_clean <- week2_data %>%
  filter(
    PULocationID %in% latitude_map$LocationID,    # PULocationID belongs to latitude table
    DOLocationID %in% latitude_map$LocationID,    # DOLocationID belongs to latitude table
    tip_amount > 0 & tip_amount < 50,             # tip_amount is positive
    passenger_count > 0 & passenger_count <= 5,   # Passenger count (between 1 and 5)
    extra %in% c(0, 0.5, 1),                      # Valid extra charges (0, 0.5, 1)
    fare_amount > 0 & fare_amount <100,           # Positive fare amount
    trip_distance < 25,                           # Trip distances under 25 miles
    tolls_amount <10,                             # Tolls amount under 10 dollars
    PU_Borough != "Unknown",                      # Filter out data entry errors for location
    PU_Zone != "NA",
    PU_service_zone != "N/A",
    DO_Borough != "Unknown",
    PU_Zone != "NA",
    PU_service_zone != "N/A")

# Rename columns and create new variables for:
# day of the week
# trip duration in hours (in hours because we will be calculating miles per hour)
# taxi driving speed in miles per hour
# time of the day (morning, afternoon, evening and night)
# rush hour (yes or no)
# airport trip (yes if either of pickup or drop off locations are airports)

morning_hours <- c("0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11")
afternoon_hours <- c("12", "13", "14", "15", "16")
evening_hours <- c("17", "18", "19")
night_hours <- c("20", "21", "22", "23")
airports <- c("JFK Airport", "LaGuardia Airport")
fare_types <- c("Standard","JFK","Newark","Nassau/Westchester","Negotiated","Group Ride")
time_of_day <- c("Morning","Afternoon","Evening","Night")
boroughs <- c("Manhattan","Queens","Brooklyn","Bronx","EWR","Staten Island")

week2_clean <- week2_clean %>%
  rename("dropoff_datetime" = "tpep_dropoff_datetime",
         "pickup_datetime" = "tpep_pickup_datetime") %>%
  mutate(dow = wday(pickup_datetime,label=TRUE, week_start = 1),
         hour_trip_start = factor(hour(pickup_datetime)),
         trip_duration = as.numeric(difftime(dropoff_datetime,pickup_datetime,units="hours")),
         trip_speed = trip_distance/trip_duration,
         extra = factor(extra, levels = c(0,0.50,1)),
         fare_type = case_when(
          RatecodeID == 1 ~ "Standard",
          RatecodeID == 2 ~ "JFK",
          RatecodeID == 3 ~ "Newark",
          RatecodeID == 4 ~ "Nassau/Westchester",
          RatecodeID == 5 ~ "Negotiated",
          RatecodeID == 6 ~ "Group Ride"),
         tod_trip_start = case_when(
           hour_trip_start %in% morning_hours ~ "Morning",
           hour_trip_start %in% afternoon_hours ~ "Afternoon",
           hour_trip_start %in% evening_hours ~ "Evening",
           hour_trip_start %in% night_hours ~ "Night"),
         hour_rush = case_when(
          hour_trip_start %in% 8:9 ~ "Yes",
          hour_trip_start %in% 15:19 ~ "Yes", TRUE ~ "No"),
         trip_airport = case_when(
          PU_Zone %in% airports ~ "Yes",
          DO_Zone %in% airports ~ "Yes", TRUE ~ "No"),
         dow = factor(dow, levels = c("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"), order = FALSE),
         fare_type = factor(fare_type, levels = fare_types),
         tod_trip_start = factor(tod_trip_start, levels = time_of_day),
         hour_rush = factor(hour_rush, levels = c("No","Yes")),
         trip_airport = factor(trip_airport, levels = c("No", "Yes")),
         PU_Borough = factor(PU_Borough, levels = boroughs),
         DO_Borough = factor(DO_Borough, levels = boroughs))

# further clean our data for new outliers such as:
# trip duration having negative hours or being greater than 2 hours
# trip speed being negative miles per hour or greater than 50 miles per hour limit
week2_clean <- week2_clean %>%
  filter(trip_duration > 0 & trip_duration < 2,
         trip_speed > 0  & trip_speed <= 50,
  )

# select useful columns, omit missing values
chosen_columns <- c(5,11,12,14,15,20,23,26,27,28,29,30,31,32,33)
week2_model_data <- week2_clean[,chosen_columns] %>% na.omit()
```

Data Visualisation

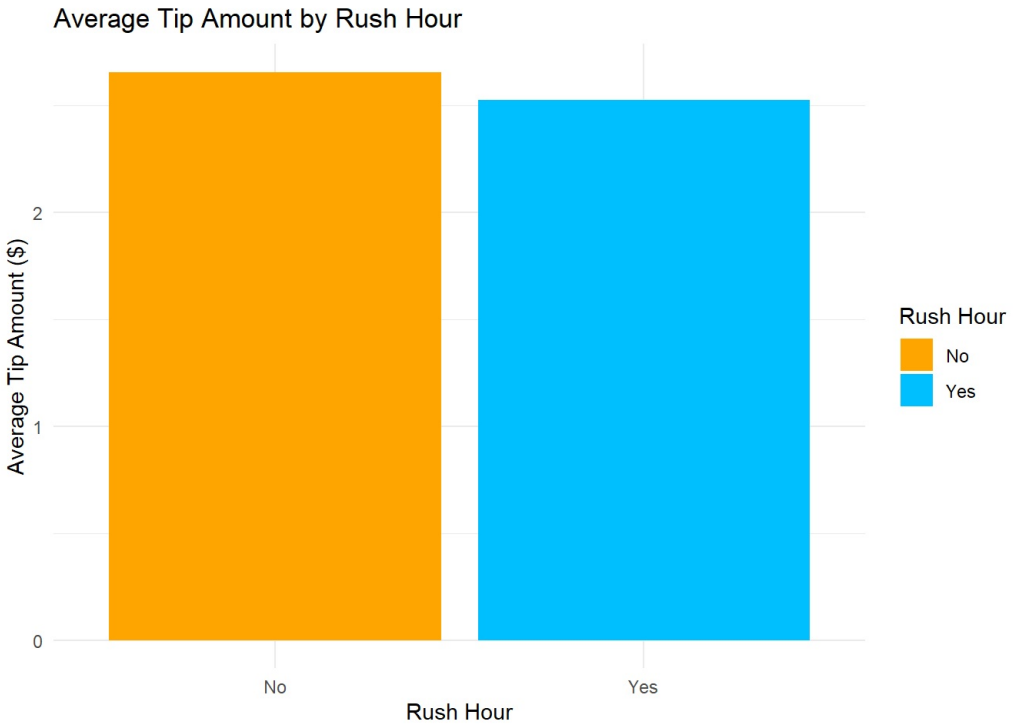
```
# sample 10% of data
my_sample <- sample(nrow(week2_model_data), round(nrow(week2_model_data) * 0.1))

# plot average tip amount by hour
week2_model_data[my_sample,] %>%
  group_by(hour_trip_start) %>%
  summarise(average_tip = mean(tip_amount)) %>%
  ggplot(aes(x = hour_trip_start, y = average_tip, fill = hour_trip_start)) +
  geom_col(fill = "dodgerblue") +
  labs(title = "Average Tip Amount By Hour", x = "Hour", y = "Average Tip Amount ($)") +
  theme_minimal()
```



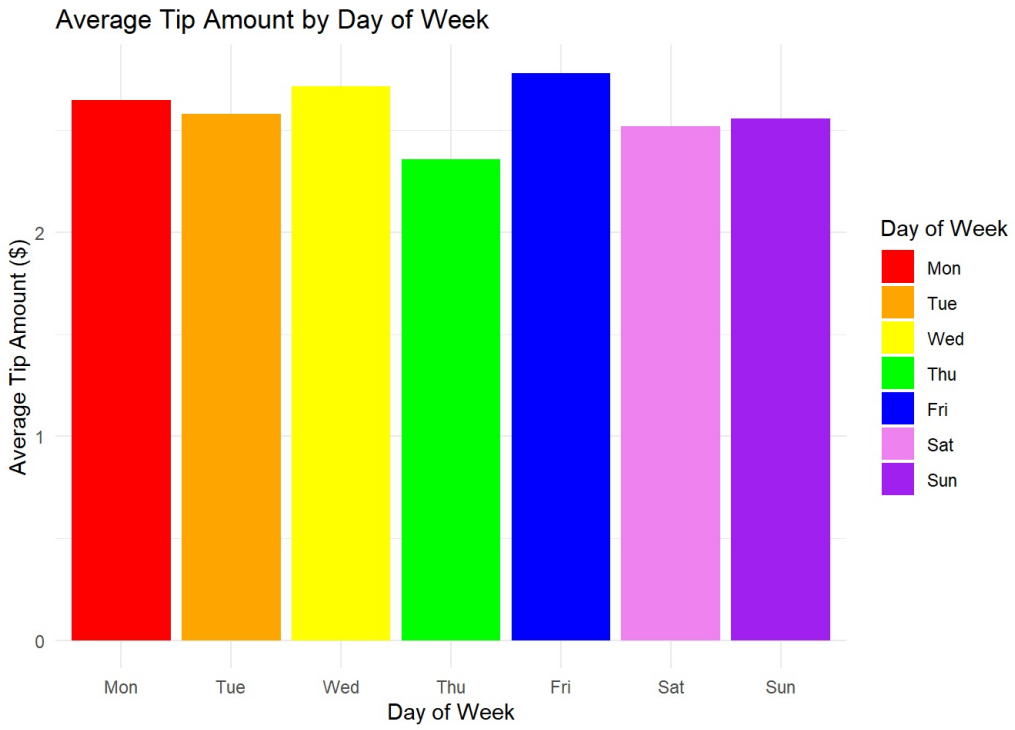

```
# plot average tip amount by time of the day
week2_model_data[my_sample,] %>%
  group_by(tod_trip_start) %>%
  summarise(average_tip = mean(tip_amount)) %>%
  ggplot(aes(x = tod_trip_start, y = average_tip, fill = tod_trip_start)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("Morning" = "gold", "Afternoon" = "deepskyblue", "Evening" = "purple", "Night" = "
midnightblue")) +
  labs(title = "Average Tip Amount By Time of Day", x = "Time of Day", y = "Average Tip Amount ($)", fill = "Time
of Day") +
  theme_minimal()
```



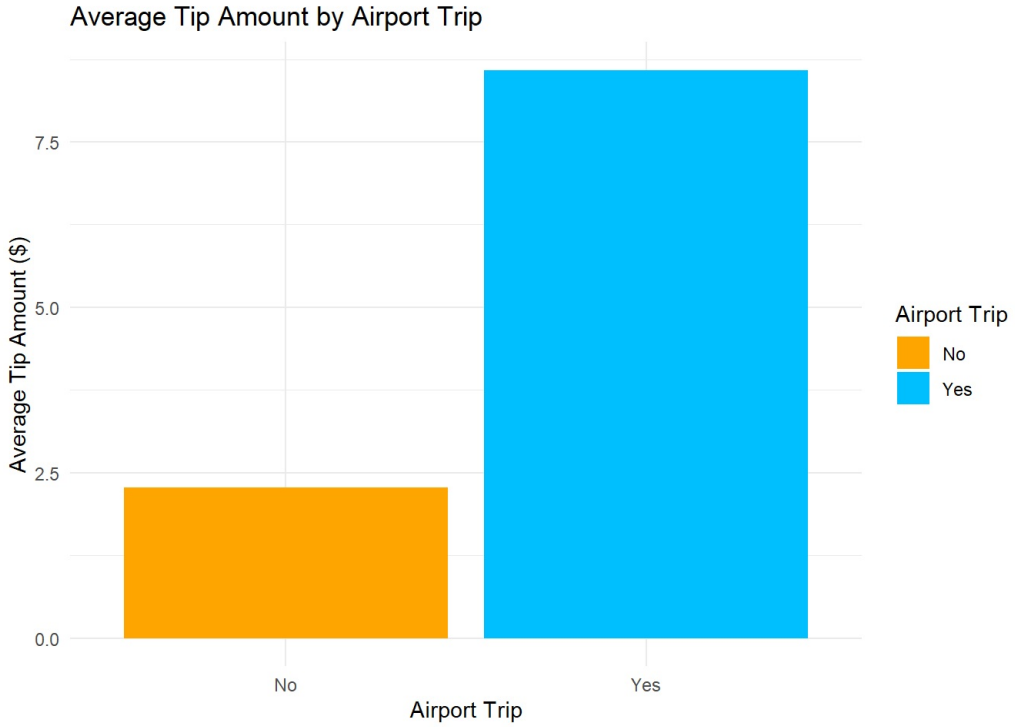

```
# plot average tip amount by rush hour
week2_model_data[my_sample,] %>%
  group_by(hour_rush) %>%
  summarise(average_tip = mean(tip_amount)) %>%
  ggplot(aes(x = hour_rush, y = average_tip, fill = hour_rush)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("No" = "orange", "Yes" = "deepskyblue")) +
  labs(title = "Average Tip Amount by Rush Hour", x = "Rush Hour", y = "Average Tip Amount ($)", fill = "Rush Hou
r") +
  theme_minimal()
```




```
# plot average tip amount by day of the week
week2_model_data[my_sample,] %>%
  group_by(dow) %>%
  summarise(average_tip = mean(tip_amount)) %>%
  ggplot(aes(x = dow, y = average_tip, fill = dow)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("Mon" = "red", "Tue" = "orange", "Wed" = "yellow", "Thu" = "green", "Fri" = "blue"
, "Sat" = "violet", "Sun" = "purple")) +
  labs(title = "Average Tip Amount by Day of Week", x = "Day of Week", y = "Average Tip Amount ($)", fill = "Day
of Week") +
  theme_minimal()
```

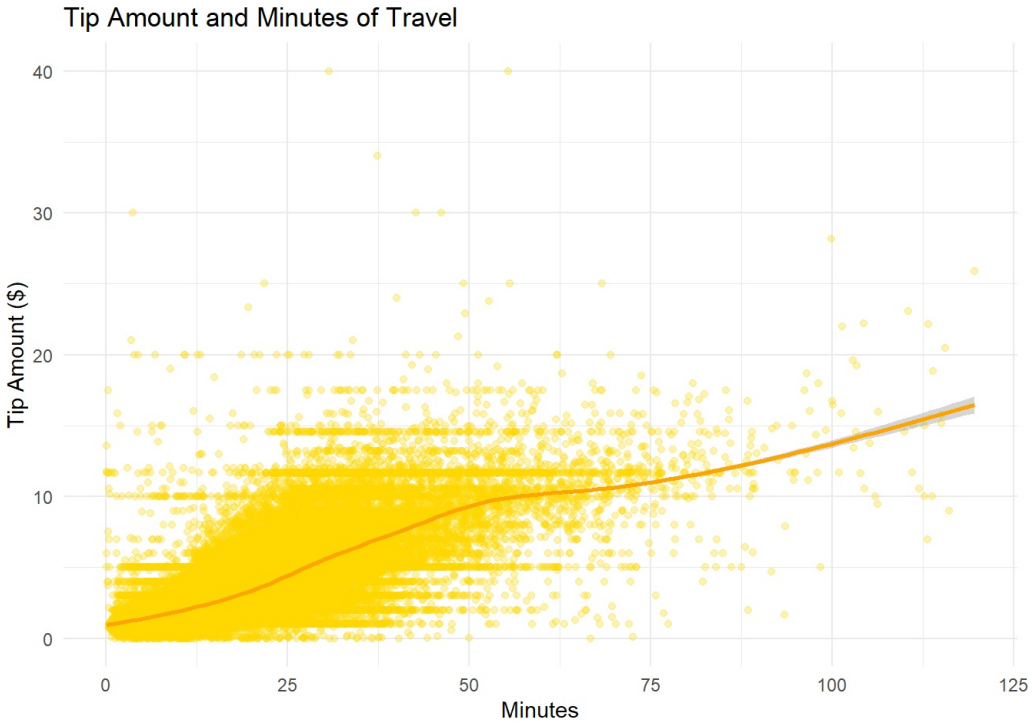



```
# plot average tip amount by airport trip
week2_model_data[my_sample,] %>%
  group_by(trip_airport) %>%
  summarise(average_tip = mean(tip_amount)) %>%
  ggplot(aes(x = trip_airport, y = average_tip, fill = trip_airport)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("No" = "orange", "Yes" = "deepskyblue")) +
  labs(title = "Average Tip Amount by Airport Trip", x = "Airport Trip", y = "Average Tip Amount ($)", fill = "Ai
rport Trip") +
  theme_minimal()
```



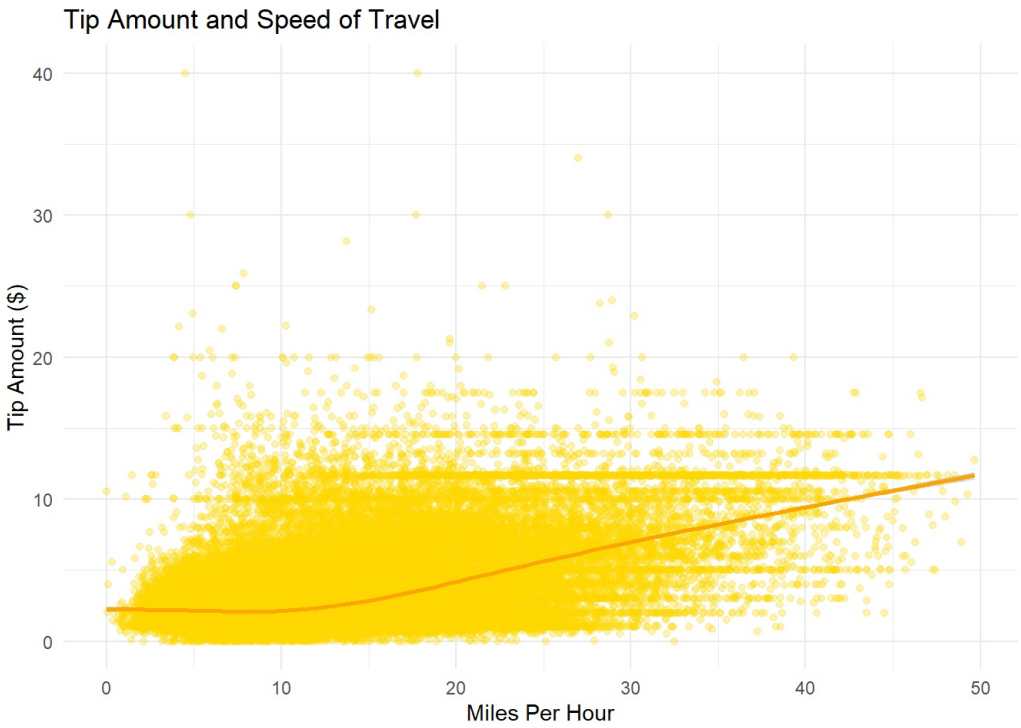

```
# plot average tip amount by minutes of travel
week2_model_data[my_sample,] %>%
  ggplot(aes(x = trip_duration*60, y = tip_amount)) +
  geom_point(alpha = 0.3, colour = "gold") +
  geom_smooth(colour = "orange") +
  labs(title = "Tip Amount and Minutes of Travel", x = "Minutes", y = "Tip Amount ($)") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

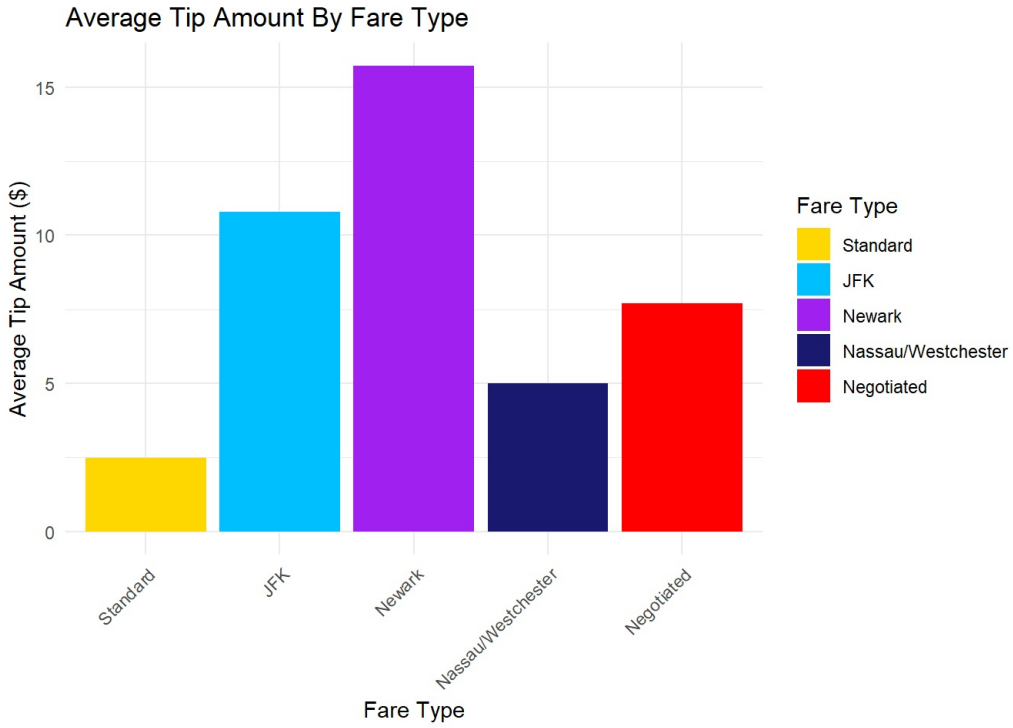



```
# plot average tip amount by speed of travel
week2_model_data[my_sample,] %>%
  ggplot(aes(x = trip_speed, y = tip_amount)) +
  geom_point(alpha = 0.3, colour = "gold") +
  geom_smooth(colour = "orange") +
  labs(title = "Tip Amount and Speed of Travel", x = "Miles Per Hour", y = "Tip Amount ($)") +
  theme_minimal()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```




```
# plot average tip amount by rate code/fare type
week2_model_data[my_sample,] %>%
  group_by(fare_type) %>%
  summarise(average_tip = mean(tip_amount)) %>%
  ggplot(aes(x = fare_type, y = average_tip, fill = fare_type)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("Standard" = "gold", "JFK" = "deepskyblue", "Newark" = "purple", "Nassau/Westchest
er" = "midnightblue", "Negotiated" = "red", "Group Ride" = "blue")) +
  labs(title = "Average Tip Amount By Fare Type", x = "Fare Type", y = "Average Tip Amount ($)", fill = "Fare Typ
e") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))
```




```
# remove hour of day column
week2_model_data <- week2_model_data[,-9]

# display data frame with finalized chosen variables
slice(week2_model_data, 500:510)
```

```
## # A tibble: 11 × 14
##    trip_distance fare_amount extra tip_amount tolls_amount PU_Borough DO_Borough
##            <dbl>       <dbl> <fct>      <dbl>        <dbl> <fct>      <fct>
##  1           1.5         7.5 0.5         1.76            0 Manhattan  Manhattan
##  2           4.7        14   0.5         3.05            0 Manhattan  Manhattan
##  3           1.1         5   0.5         1.25            0 Manhattan  Manhattan
##  4           7.1        21   0.5         4.45            0 Manhattan  Manhattan
##  5           1           5.5 0.5         1.35            0 Queens     Queens
##  6           6.8        22.5 0.5         4.75            0 Queens     Manhattan
##  7           3.8        13.5 0.5         2               0 Manhattan  Manhattan
##  8           3.26       11.5 0.5         2               0 Manhattan  Manhattan
##  9           2.83       13   0.5         2.86            0 Manhattan  Manhattan
## 10           1.42        6.5 0.5         1.2             0 Manhattan  Manhattan
## 11           2.4        10.5 0.5         2.35            0 Manhattan  Manhattan
## # i 7 more variables: dow <fct>, trip_duration <dbl>, trip_speed <dbl>,
## #   fare_type <fct>, tod_trip_start <fct>, hour_rush <fct>, trip_airport <fct>
```

How average tip amount is influenced by hour, time of the day, rush hour and weekdays

In observing our bar chart of the average tip amounts by hour, we can see that the average tip amounts are generally slightly higher at around 10pm to 12am, 2pm to 3pm and generally a bit lower at around 7am to 8am and 6pm to 7pm. We believe that the tip amounts are slightly lower during these hours because workers in NYC are busy getting to work, and they may be less inclined to give generous tips. We also notice the average tip amount spikes above $3.00 at 4am and 5am, which indicates some passengers during these hours are in a hurry to reach their destinations.

In our bar chart of the average tip amounts by time of the day, the tip amounts are generally a bit higher during the afternoon and night times, and lower during the morning and evening times. This suggests that during the morning and evening times when workers are trying to get to work or get home, they are generally tip slightly less.

Our bar chart of the average tip amounts categorized by rush hour shows that during rush hours, passengers tend to tip slightly less than if it were not during the rush hours. This reinforces our earlier assumption that during certain times of the day, when passengers are in a hurry to get to work or getting home from work, they are less in the mood to offer higher tips. It may also be attributed to the fact that some passengers regularly use their taxi services to get to work and hence they provide lower tips.

During times which are not rush hours such as in the afternoon and night, passengers often use taxi services for purposes often not related to work, but to travel or for a special occasion. Hence, they may be more likely to offer higher tips.

The average tip amount by weekday varies across Monday to Sunday, and it appears that the tips are slightly higher on Wednesdays and Fridays but lower on Thursdays.

Average tip amount and airport related trips

Our next bar chart of the average tip amount and whether the trip was an airport pickup or drop off shows a large difference in the tip amount. If the passengers were on a trip to get dropped off from the airport or are getting picked up from the airport, they generally tip more than 3 times the average tip amount by passengers that are not in an airport related trip.

Tip amount, time and speed of travel

Our dot plot shows that as travel time in minutes increases, the tip amount gradually increases.

Our dot plot of the tip amounts and speed of travel shows that the tip amount remains stationary at approximately $2.5 as the taxi travel speed approaches 15 miles per hour, from that point on wards the tip amounts gradually increases as travel speed increases.

Average Tip amount and fare types

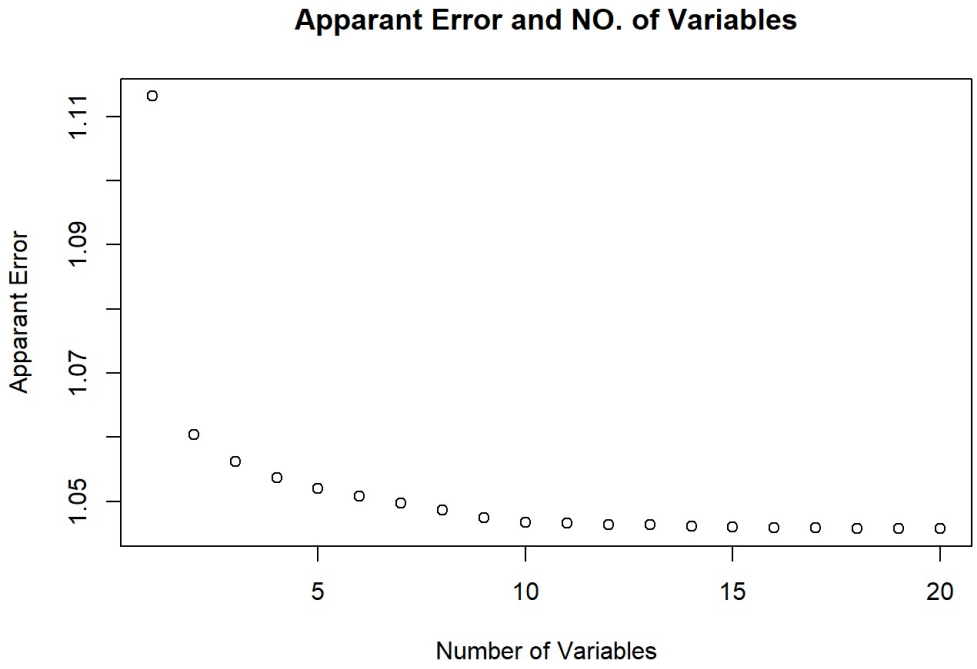
We can see that the tip amounts vary substantially between the different types of taxi trips. Standard trips have the lowest average tips of around $2.5, and Newark has the highest of around $25. Surprisingly, negotiated trips have a much higher average tip of around $7 than standard tips.

Finalising the variables for model building

We decided to keep all the variables in the cleaned data set, except for hour of the day. In our earlier observations of the bar charts of the hour of the day and time of the day, we noticed that the time of the day (morning, afternoon, evening, night) has reasonably grouped the hours where the tips were higher or lower. As hours has 24 categorical levels, it may take up unnecessary space when generating our model coefficients, hence we would remove that variable as their effects could also be captured by the time of the day. We believe keeping variables which explain a similar effect that has less categorical factors in favor of another variable that has many factors would allow us to create more accurate models.

Model Traning

```
# create week 2 model matrix
model_frame2 <-model.frame(tip_amount~., data=week2_model_data)
X2 <- model.matrix(tip_amount~., model_frame2)[,-1]

# use regsubsets to select best models
bsearch <- regsubsets(X2, week2_clean$tip_amount, nvmax=20, method="back")
bsum <- summary(bsearch)

# plot apparant error
bapparant_error <- bsum$rss / (nrow(week2_clean) - (1:20))
plot(bapparant_error, main = "Apparant Error and NO. of Variables", xlab = "Number of Variables", ylab = "Apparan
t Error")
```



```
# function to cross validate
allyhat<-function(xtrain, ytrain, xtest,lambdas,nvmax=20){
  n<-nrow(xtrain)
  yhat<-matrix(nrow=nrow(xtest),ncol=length(lambdas))
  search<-regsubsets(xtrain,ytrain, nvmax=nvmax, method="back")
  summ<-summary(search)
  for(i in 1:length(lambdas)){
    penMSE<- n*log(summ$rss)+lambdas[i]*(1:nvmax)
    best<-which.min(penMSE)  #lowest penMSE
    betahat<-coef(search, best) #coefficients
    xinmodel<-cbind(1,xtest)[,summ$which[best,]] #predictors in that model
    yhat[,i]<-xinmodel%*%betahat
  }
  yhat
}

# 10-fold cross-validation, calculate predicted tips for each lambda
y2 <- week2_model_data$tip_amount
n<-nrow(X2)
folds<-sample(rep(1:10,length.out=n))
lambdas<-c(2,4,6,8,10,12)
fitted<-matrix(nrow=n,ncol=length(lambdas))
for(k in 1:10){
  train<- (1:n)[folds!=k]
  test<-(1:n)[folds==k]
  fitted[test,]<-allyhat(X2[train,],y2[train],X2[test,],lambdas)
}
```

```
## Reordering variables and trying again:
## Reordering variables and trying again:
```

```
# calculate MSPE for each lambda, select best lambda
lambda <- lambdas[which.min(colMeans((y2-fitted)^2))] #best lambda is 2

# select best predictive model
penMSE<- n*log(bsum$rss)+ lambda*(1:20)
best<-which.min(penMSE)
betahat<-coef(bsearch, best) #coefficients of best model

# data frame to illustrate model selection
data.frame("variables" = 1:20,
           "penalised_MSE" = penMSE)
```

```
##    variables penalised_MSE
## 1          1      19578899
## 2          2      19512168
## 3          3      19506682
## 4          4      19503442
## 5          5      19501183
## 6          6      19499707
## 7          7      19498322
## 8          8      19496875
## 9          9      19495290
## 10        10      19494384
## 11        11      19494170
## 12        12      19493946
## 13        13      19493825
## 14        14      19493632
## 15        15      19493455
## 16        16      19493310
## 17        17      19493221
## 18        18      19493144
## 19        19      19493101
## 20        20      19493060
```

```
# MSPE for predicting week 2 tips
xinmodel2<-cbind(1,X2)[,bsum$which[best,]] #predictors for week 2 data
yhat_2 <-xinmodel2%*%betahat
MSPE_w2 <- mean((y2[1:100]-yhat_2[1:100])^2)
MSPE_w2 <- round(MSPE_w2,3)
```

Training our model

We firstly created a model matrix using our week 2 data with the tip amount regressed against all the other variables and used regsubsets to generate different models with up to 20 variables. We calculated the apparant errors of our models by penalizing between 1 to 20 variables, and it appears that the apparant error decreases very slightly after 10 variables.

When implementing our 10-fold cross validation, we created a function called “allyhat” which takes the split training and testing week 2 data as parameters and various lambda multiplier values. Inside every loop of this function, it calculates the predicted tip amounts for different values of lambda. For our 10-fold cross validation, we split our data into 10 random portions, use 9 parts for training and 1 part for testing. After cross-validation, we use the predicted tip amounts to calculate the MSPE for each lambda, we had the lowest MSPE when lambda was 2.

Next, we used the best lambda to find the optimal number of coefficients for our model. We calculated the penalized MSE, with the number of coefficients ranging from 1 to 20. Our results suggests that including all 20 variables was optimal, and we extracted our model coefficients.

Model Testing

```
# start of cleaning and processing week 4 data --------------------------------------
week4_data <- week4_data %>%
  left_join(zone_data, by = c("PULocationID" = "LocationID")) %>%
  rename(PU_Borough = Borough,
         PU_Zone = Zone,
         PU_service_zone = service_zone) %>%
  left_join(zone_data, by = c("DOLocationID" = "LocationID")) %>%
  rename(DO_Borough = Borough,
         DO_Zone = Zone,
         DO_service_zone = service_zone)

week4_clean <- week4_data %>%
  filter(
    PULocationID %in% latitude_map$LocationID,    # PULocationID belongs to latitude table
    DOLocationID %in% latitude_map$LocationID,    # DOLocationID belongs to latitude table
    tip_amount > 0 & tip_amount < 50,             # tip_amount is positive
    passenger_count > 0 & passenger_count <= 5,   # Passenger count (between 1 and 5)
    extra %in% c(0, 0.5, 1),                      # Valid extra charges (0, 0.5, 1)
    fare_amount > 0 & fare_amount <100,           # Positive fare amount
    trip_distance < 25,                           # Trip distances under 25 miles
    tolls_amount <10,                             # Tolls amount under 10 dollars
    PU_Borough != "Unknown",                      # Filter out data entry errors for location
    PU_Zone != "NA",
    PU_service_zone != "N/A",
    DO_Borough != "Unknown",
    PU_Zone != "NA",
    PU_service_zone != "N/A")


week4_clean <- week4_data %>%
  rename("dropoff_datetime" = "tpep_dropoff_datetime",
         "pickup_datetime" = "tpep_pickup_datetime") %>%
  mutate(dow = wday(pickup_datetime,label=TRUE, week_start = 1),
         hour_trip_start = factor(hour(pickup_datetime)),
         trip_duration = as.numeric(difftime(dropoff_datetime,pickup_datetime,units="hours")),
         trip_speed = trip_distance/trip_duration,
         extra = factor(extra, levels = c(0,0.50,1)),
         fare_type = case_when(
          RatecodeID == 1 ~ "Standard",
          RatecodeID == 2 ~ "JFK",
          RatecodeID == 3 ~ "Newark",
          RatecodeID == 4 ~ "Nassau/Westchester",
          RatecodeID == 5 ~ "Negotiated",
          RatecodeID == 6 ~ "Group Ride"),
         tod_trip_start = case_when(
           hour_trip_start %in% morning_hours ~ "Morning",
           hour_trip_start %in% afternoon_hours ~ "Afternoon",
           hour_trip_start %in% evening_hours ~ "Evening",
           hour_trip_start %in% night_hours ~ "Night"),
         hour_rush = case_when(
          hour_trip_start %in% 8:9 ~ "Yes",
          hour_trip_start %in% 15:19 ~ "Yes", TRUE ~ "No"),
         trip_airport = case_when(
          PU_Zone %in% airports ~ "Yes",
          DO_Zone %in% airports ~ "Yes", TRUE ~ "No"),
         dow = factor(dow, levels = c("Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"), order = FALSE),
         fare_type = factor(fare_type, levels = fare_types),
         tod_trip_start = factor(tod_trip_start, levels = time_of_day),
         hour_rush = factor(hour_rush, levels = c("No","Yes")),
         trip_airport = factor(trip_airport, levels = c("No", "Yes")),
         PU_Borough = factor(PU_Borough, levels = boroughs),
         DO_Borough = factor(DO_Borough, levels = boroughs))

week4_clean <- week4_clean %>%
```

```
  filter(trip_duration > 0 & trip_duration < 2,
         trip_speed > 0  & trip_speed <= 50,
  )

week4_model_data <- week4_clean[,c(5,11,12,14,15,20,23,26,28,29,30,31,32,33)] %>% na.omit()
# end of processing week 4 data ----------------------------------------------------------------

# create week 4 model matrix
model_frame4 <-model.frame(tip_amount~., data=week4_model_data)
X4 <- model.matrix(tip_amount~., model_frame4)[,-1]

# calculate MSPE for week 4
y4 <- week4_model_data$tip_amount
xinmodel4<-cbind(1,X4)[,bsum$which[best,]] #predictors for week 4 data
yhat_4 <-xinmodel4%*%betahat
MSPE_w4 <- mean((y4-yhat_4)^2)
MSPE_w4 <- round(MSPE_w4,3)
```

We cleaned and processed our week 4 data in the same method as for our week 2 data, we created a model matrix to extract all the predictors and used the model coefficients to calculate the predicted tip amounts. After that, we calculated the MSPE for week 4.

Model Results

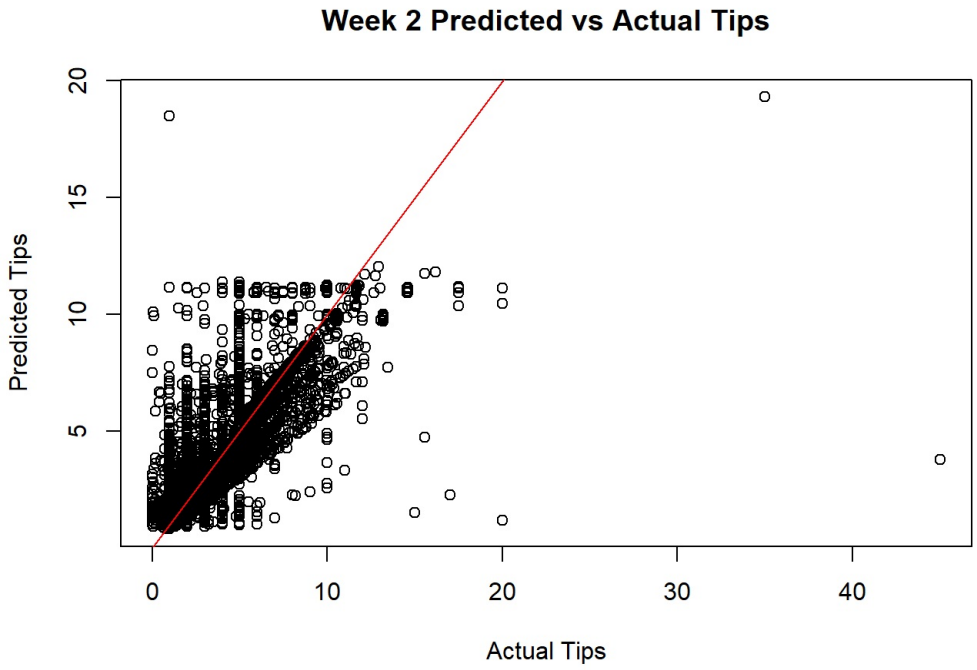
```
# Estimate coefficients
betahat
```

```
##             (Intercept)           trip_distance             fare_amount
##             0.296102890            -0.008943315             0.167086742
##                extra0.5                  extra1            tolls_amount
##             0.102096407             0.155625489             0.215609729
##           PU_BoroughEWR        DO_BoroughQueens      DO_BoroughBrooklyn
##            -6.588812851             0.206671701             0.199670550
##           DO_BoroughEWR DO_BoroughStaten Island                 dowTue
##             3.052764861             3.380980228             0.032297031
##                  dowWed                  dowThu                  dowSat
##             0.037900423             0.101147617            -0.024817396
##           trip_duration           fare_typeJFK         fare_typeNewark
##             0.510690062             0.135766458             4.136643761
##     fare_typeGroup Ride tod_trip_startAfternoon         trip_airportYes
##             9.116203248             0.027092705             0.545157163
```

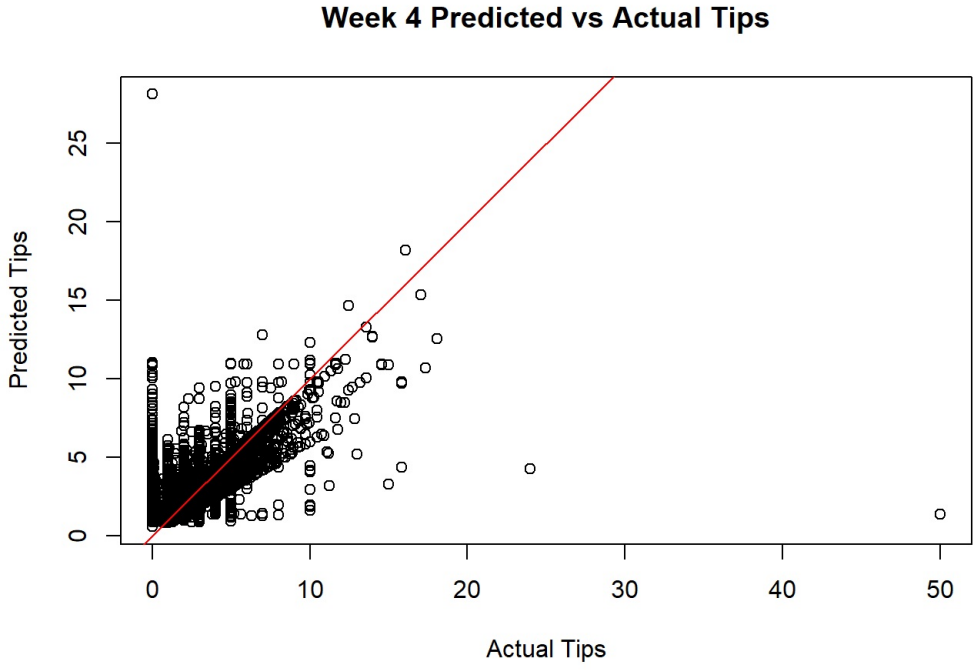
```
# MSPE table
data.frame(`Week_2_MSPE` = MSPE_w2,
           `Week_4_MSPE` = MSPE_w4)
```

```
##   Week_2_MSPE Week_4_MSPE
## 1       2.741       2.204
```

```
# Plot week 2 predicted and actual values
plot(y2[1:1e4], yhat_2[1:1e4], main = "Week 2 Predicted vs Actual Tips", xlab = "Actual Tips", ylab = "Predicted
Tips" )
abline(0, 1, col = "red", lwd = 1, lty = 1)
```




```
# Plot week 4 predicted and actual values
plot(y4[1:1e4], yhat_4[1:1e4], main = "Week 4 Predicted vs Actual Tips", xlab = "Actual Tips", ylab = "Predicted
Tips" )
abline(0, 1, col = "red", lwd = 1, lty = 1)
```



Conclusion

As we inspect a few of our model coefficients, the coefficient for fare amount is 0.167 which suggests that for every dollar increase in fare amount, the tip amount increases by $0.167. This is roughly 16.7% of the fare amount, which is consistent with the tourist's advice that a tip should be between 15% to 20%. Our coefficient for airport trip was 0.545, which implies that if the trip was an airport trip, the tip amount increases by $0.545 in comparison to if the trip was not airport related. This was unexpected because our earlier analysis showed that airport trips had an average tip of around 8.6 dollars, compared to 2.3 dollars for non-airport trips. However, there may be other factors which influencing this. Our coefficient for tolls is 0.216, for every dollar increase in toll amount, the tip amount increases by $0.216. This aligns with our earlier data analysis. However, our trip distance has a negative coefficient of 0.0089, which implies that as distance traveled increases, the tip amount decreases.

Our MSPE for week 2 and 4 taxi data was approximately 2.741 and 2.204, respectively. The mean squared prediction error is an indicator of the models accuracy. A lower MSPE indicates a more accurate model as the predicted values are closer to the actual values. From the perspective of our data where tip amounts which ranges from 0 to 50, our MSPE for week 2 and week 4 was reasonably accurate, as it indicates about a dollar and half difference from the actual values.

As we observe the week 4 predicted and fitted values (for 10,000 observations), we could still see a substantial amount of data being quite far from the trend line, the closer the data points to the trend line, the more accurate the predictions are. We could see some outliers, such as the predicted tip being over 25 dollars when the actual tip was close to zero, and the actual tip being 50 dollars when the predicted tip was about 2 dollars. These

outliers suggests that our model could be improved.

One flaw of our linear model is that it assumes linearity when there may be non-linear relationships between our variables. For example, the relationship between tip amount and distance was non-linear, however, our model assumes its linear which gives us unreliable estimates.

Another flaw of our linear model is that it's prone to multi-collinearity, where if some variables are highly correlated, it makes it difficult to create accurate estimates of our coefficients. We attempted to reduce multi-collinearity when we removed hour of the day and kept time of the day as they were highly correlated, however, there may be other variables that are also highly correlated.

Another flaw of our linear model is it assumes homoscedasticity, our model assumes that the variance of our residuals are constant across all predictor variables. For example, if the variance of the residuals increases as fare amount increases, our linear model does not account for this and we would obtain biased standard errors and our estimates would be less reliable.

Overall, in terms of linear models, we were very satisfied with our estimates.