New York City Taxi Analysis

Tyler Nelson

June 26, 2017

NOTE: This document is intended to summarize the ideas created in the R code. To be able to reproduce the results one must go through the steps in the corresponding R code provided.

Question 1

Part 1

The current analysis was conducted using the statistical program, R. First, I programmatically downloaded the "Green" Taxi cab data for September 2015 using the `read.csv()` function as follows:

```
tripdat <- read.csv('https://s3.amazonaws.com/nyc-tlc/trip+data
/green_tripdata_2015-09.csv')
```

Per the code above, the Taxi cab dataset is named `tripdat`, and will be referred to as such throughout the challenge.

Part 2

To report the number of rows and columns in `tripdat` we use the function `dim()` as follows:

```
dim(tripdat}
```

When executed in R we have that the dataset consists of 1,494,926 observations (rows) and 21 variables (columns).

Question 2

We will now consider one of the 21 variables from `tripdat` called "Trip_distance". To get a visual representation of this variable a histogram plot is created using the function `ggplot()` with code:

```
ggplot(data = tripdat , aes(x = Trip_distance)) +
  geom_histogram()+ggtitle("Uncorrected Histogram of Trip Distance")
```

The resulting histogram is shown in Figure 1.

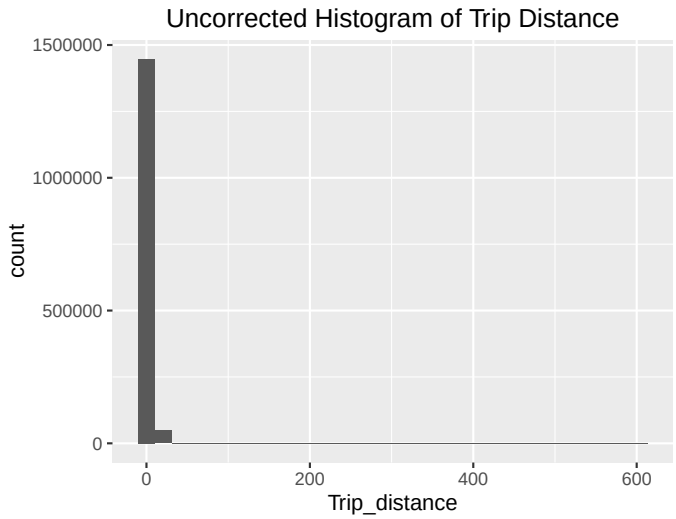

Figure 1: Histogram of Trip Distance without accounting for extreme values.

The range of the histogram in Figure 1 seems to suggest the presence of some extreme values which might warrant further investigation. Further exploration shows that there are just a few very extreme values, the largest being 603.1 with the second largest being 246.3. Thus, to provide a more helpful visualization of the data I have constructed a new dataset `NewDat` that only considers the middle 99% of the data. (Further explanation of how this was executed in R is provided in the R file containing the code used for the current Data Challenge). Figure 2 shows the histogram for the trimmed trip distance dataset, `NewDat`.

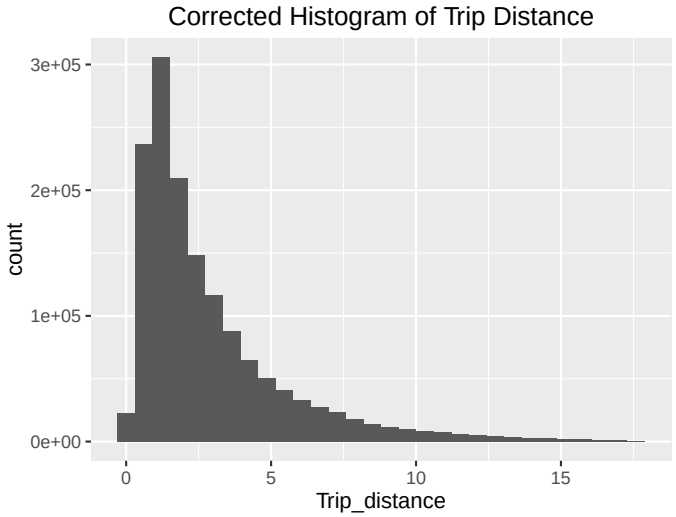

Figure 2: Histogram of Trip Distance with the middle 99% of the data.

The histogram in Figure 2 provides a more clear representation of the trip distance, namely that the trip distance is heavily skewed to the right. This may be unexpected as it is not often that people travel long distances in a taxi.

Question 3

Part 1

Next, we want to investigate the mean and median trip distance by the hour. To do this we first create a new variable that represents the time of day in which an individual was picked up. The `tripdat` dataset consists of a variable called "lpep_pickup_datetime", which records the day and time an individual was picked up to the second. The overall structure of this variable is a factor (what R defaults to specify it as). However, for computational convenience we convert this variable to a character string (a more user-friendly variable classification). Now, we use the package `stringr` accompanied with the function `str_extract()` to create a new variable called "Hour". The R code is as follows:

```
Pattern <- "[ ]([0-9][0-9])"
```

```
tripdat$Hour <- as.factor(str_extract(as.character(
tripdat$lpep_pickup_datetime), Pattern))
```

Note that this task would have been more difficult if the data provided was not "clean". Next, we can compute the mean and median of trip distance at each hour by using the following code:

```
MeanDistance <- sapply(levels(tripdat$Hour), function(x){
                mean(tripdat$Trip_distance[which(tripdat$Hour==x)])
                }
                )

MedianDistance <- sapply(levels(tripdat$Hour), function(x){
                  median(tripdat$Trip_distance[which(tripdat$Hour==x)])
                  }
                  )
```

This information is provided in Table 1 and Figure 3 below.

Table 1: Table of mean and median trip distance across hours of the day.

Hour	Mean Distance	Median Distance
00	3.120	2.200
01	3.020	2.120
02	3.050	2.140
03	3.210	2.200
04	3.530	2.360
05	4.130	2.900
06	4.050	2.840
07	3.280	2.170
08	3.050	1.980
09	3.000	1.960
10	2.940	1.920
11	2.910	1.880
12	2.900	1.890
13	2.880	1.840
14	2.860	1.830
15	2.860	1.810
16	2.780	1.800
17	2.680	1.780
18	2.650	1.800
19	2.720	1.850
20	2.780	1.900
21	3.000	2.030
22	3.190	2.200
23	3.190	2.220

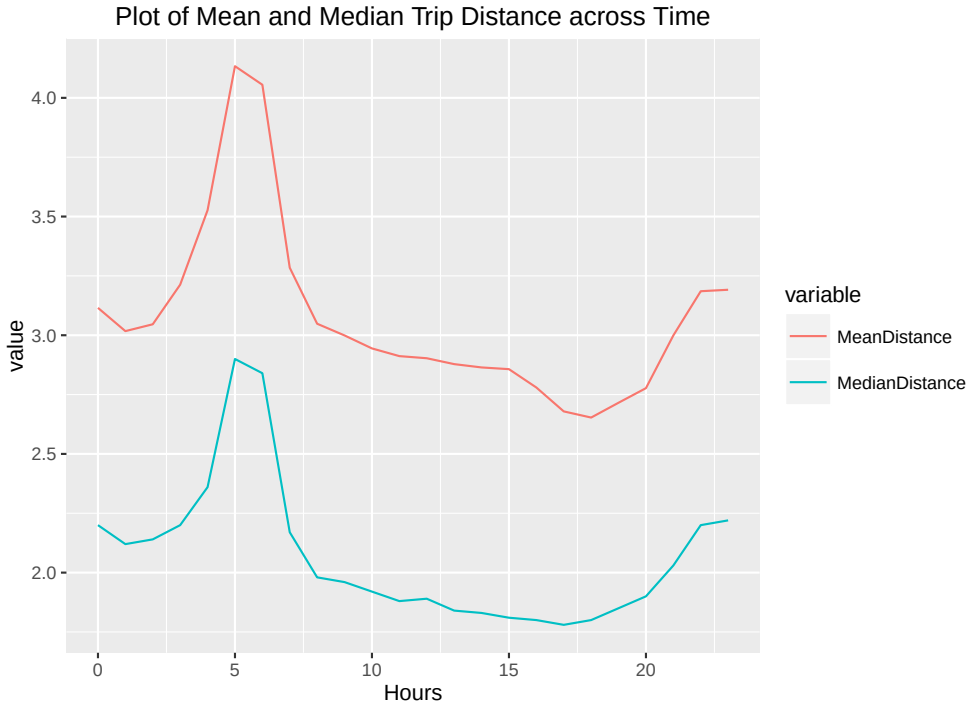

Figure 3: Plot of the mean and median travel distance across hours of the day.

Table 1 and Figure 3 suggest that the mean is larger than the median for all hours of the day. Hence, at each time point the data is right skewed. Furthermore, from Figure 3 we see that the peak travel distance is at 5:00am and then decreases throughout the rest of the day until about 5:00pm, at which point it increases. This is not surprising as the travel distance is elevated when people are heading to and from work. Therefore, a taxi driver of a green taxi would maximize their profit by working early in the morning and then later in the afternoon.

Part 2

Now we want to get a rough sense of identifying how many trips originate or terminate at one of the NYC area airports and an average estimate of fare amount. Further exploration of the data and reading an explanation of each

of the variables[1], I identified the variable called "RateCodeID" which has six different levels:

$$
\begin{aligned}
1 &= \text{Standard rate} \\
2 &= \text{JFK} \\
3 &= \text{Newark} \\
4 &= \text{Nassau or Westchester} \\
5 &= \text{Negotiated fare} \\
6 &= \text{Group ride.}
\end{aligned}
$$

From the list above there are two specified NYC area airports, namely, JFK and Newark. Per the explanation for the JFK and Newark airport codes, it seems that the JFK airport will provide a better sense of how many trips originate or terminate at the airport (Per the explanation To/From JFK airport). In contrast, the explanation for the Newark airport code [2] is when a taxi driver goes "To" Newark airport.Thus, the current analysis focuses only on the JFK airport. The following code was used to construct a dataset that consisting of only observations that were recorded with a RateCodeID of 2.

```
JFKdat <- tripdat[which(tripdat$RateCodeID == 2),]

# 1. The number of of transactions are
dim(JFKdat)[1]
# 2. The average fare is
mean(JFKdat$Fare_amount)
```

The above code yields that 4,435 (0.00297% of all transactions) originated or terminated at JFK airport. Moreover, the mean fare was $49.02 which is $36.48 more than the average green taxi rate. It is noteworthy that this is a "rough" estimate because there seem to be some recording errors in the dataset. For example, there are some negative values entered for the fare. Consequently, to obtain a more precise estimate requires further exploration.

[1] The PDF for all the variables is at http://www.nyc.gov/html/tlc/downloads/pdf/data_dictionary_trip_records_green.pdf

[2] http://www.nyc.gov/html/tlc/html/passenger/taxicab_rate.shtml#AirportTrips

For example, if I had the opportunity to investigate further, I would obtain the longitude and latitude of all the NYC area airports and then use these to compute a good estimate for the number of green taxi trips that originate or terminate at each particular airport. This would provide better insight in to which airports the taxi drivers should consider to maximize profit.

Question 4

A variable of interest is the tip percentage of the fare amount. This variable is not present in the original dataset. Thus, we construct this variable by taking the tip amount divided by the fare amount and adding this new variable to the the overall dataset, `tripdat`. This can be done using the following code:

```
tripdat$tip_percent <- tripdat$Tip_amount/tripdat$Fare_amount/
```

Next, we want to obtain a predictive model for tip as a percentage of the total fare. Prior to this, we need to consider that with such a large dataset, the presence of either misclassified values or extreme outliers that may skew the overall model. Thus, I have investigated each of the variables to be considered in the model and removed what I considered an "unstable" value. (See the R code for a more detailed explanation of this process). Therefore, I have created a new dataset called `trip_subset`. Further, the variables that are considered when constructing the predictive model to identify factors that might influence tip as a percentage of total fare are labeled in the vector `varsConsidered` as shown below:

```
varsConsidered <- c("tip_percent", "Pickup_longitude", "Pickup_latitude",
                    "Dropoff_longitude","Dropoff_latitude",
                    "Trip_distance", "Hour", "Payment_type", "Fare_amount")
```

To caution against multicollinearity in model construction, preliminary analysis was done to discard highly correlated variables in order to minimize the chance of multicollinearity. In particular, a simple tool is checking the simple linear correlation between variables. Table 2 and Table 3 provide the correlation table for all variables to be considered in the analysis.

Table 2: The first half of the correlation table for the variables to be considered in our model.

	tip_percent	Pickup_longitude	Pickup_latitude	Dropoff_longitude
tip_percent	1	-0.191	-0.117	-0.181
Pickup_longitude	-0.191	1	0.240	0.701
Pickup_latitude	-0.117	0.240	1	0.128
Dropoff_longitude	-0.181	0.701	0.128	1
Dropoff_latitude	-0.119	0.199	0.843	0.131
Trip_distance	0.072	0.016	-0.053	0.093
Hour	0.020	-0.020	-0.006	-0.027
Payment_type	-0.810	0.187	0.092	0.183
Fare_amount	0.069	-0.003	-0.067	0.046

Table 3: The second half of the correlation table for the variables to be considered in our model.

	Dropoff_latitude	Trip_distance	Hour	Payment_type	Fare_amount
tip_percent	-0.119	0.072	0.020	-0.810	0.069
Pickup_longitude	0.199	0.016	-0.020	0.187	-0.003
Pickup_latitude	0.843	-0.053	-0.006	0.092	-0.067
Dropoff_longitude	0.131	0.093	-0.027	0.183	0.046
Dropoff_latitude	1	-0.115	-0.008	0.100	-0.116
Trip_distance	-0.115	1	-0.036	-0.151	0.921
Hour	-0.008	-0.036	1	-0.006	-0.021
Payment_type	0.100	-0.151	-0.006	1	-0.154
Fare_amount	-0.116	0.921	-0.021	-0.154	1

From this table, we want to identify any correlation value that is large positively or negatively (note that a correlation value is between -1 and 1). For example, we see that each of the longitude and latitude variables are highly correlated with one another and that trip distance and fare amount are highly correlated. Hence, I have selected the variable that is most highly correlated with the variable of interest, tip percent. From here we have the variables selected,

```
varsUsed <- c("tip_percent", "Pickup_longitude",
              "Dropoff_latitude", "Trip_distance",
              "Payment_type", "Hour")
```

to create the dataset `trip_subset`.

Model 1

Now, we consider methods to model tip percent. First, we consider a commonly used linear regression model. This method for modeling is both simple to implement and interpret.

The linear regression model is represented by

```
fit <- lm(tip_percent ~ . , data = trip_subset)
```

Another consideration for linear regression is variable selection. Due to time constraints we use the variable selection tool readily available in R for regression, namely the function `stepAIC()` in the `MASS` package in R. This function can be used to do forward selection and backward elimination for variable selection as follows:

```
step <- stepAIC(fit, direction="both")
step$anova # display results
```

Not surprisingly (due to the large sample size), the selection process suggests that all the variables considered are significant and should be retained in the model. The residual standard error for the linear regression model with all variables considered is 0.068, which is the lowest value for all model combinations. The corresponding multiple R-squared value of 0.663. This suggests that 66% of the variability in tip as a percentage of total fare can be explained collectively by pickup longitude, drop-off latitude, trip distance, payment type and hour. It is noteworthy that if we exclude the variable Payment type the residual standard error increases and multiple R-squared value decreases, suggesting a worse model fit. Further exploration of this model is needed (i.e. cross-validation) to make sure we have not overfit the model.

Model 2

Looking further into the data suggests that many of the variables do not have a linear relationship with tip percentage. Thus, another method that I have

considered is a multivariable fractional polynomial variable selection [3]. At a basic level fractional polynomial regression looks at different degrees of the variables considered and attempts to minimize the deviance of the overall model. The code that I have used to conduct this is as follows:

```
f <- mfp(tip_percent ~ fp(Trip_distance) + fp(Pickup_longitude) +
fp(Dropoff_latitude) + Payment_type + Hour, data = trip_subset,
 family = gaussian, alpha=0.05,
         verbose = TRUE, na.action = na.exclude)
summary(f)
```

This method is more computationally intensive and requires longer computation time than the multiple linear regression. As with the forward and backward selection, this method has selected leaving all variables in the model, but it has also brought in a few more complex variables for trip distance, pickup longitude, and drop-off latitude. Thus, I would advise to proceed with extreme caution when thinking about prediction due to the over-fitting of the model. Hence, as noted above cross-validation is necessary with this model.

Lastly, we compare model performance in terms of predicting tip percentage via comparison of the AIC value. In the case of comparing these two models the AIC for the linear regression model is smaller than the value for the fractional polynomial model. Therefore, for this fact and interpretability I would select the multivariable linear regression model.

Question 5

I have chosen to answer Option C because I wanted to show my ability to construct valid functions in R that can reproduce results. In particular, I have constructed a function that enables one to find how many origination points took place at a particular location based on the longitude and latitude specified. In addition, this function allows the user the option to specify a time point. Currently, the function only works for Hours because for simplicity I just used the variable created in Question 3, Hour. However, this

[3] A better understanding of the fractional polynomial method is explained in a journal article at this link https://www.ncbi.nlm.nih.gov/pubmed/18058845 and the implementation in R can be found at https://cran.r-project.org/web/packages/mfp/vignettes/mfp_vignette.pdf

can easily be extended to accommodate other units of time. The function arguments that need to be specified are `dat, lat, lon, miles`. There is also the optional `time` function argument. Below is the function I have constructed with some additional notes included.

```
locFunc <- function(dat, lon, lat, miles, time = "Not Specified"){

  # Making a circumference around the specified longitude and latitude values
  # The 69 and 54.6 came from the internet on the amount of miles between one
  # degree of longitude and latitude values, respectively. The radius is the
  # specified number of miles that you want to consider.
  lonCutoff <- miles/69
  lowerLon  <- lon - lonCutoff
  upperLon  <- lon + lonCutoff
  latCutoff <- miles/54.6
  lowerLat  <- lat - latCutoff
  upperLat  <- lat + latCutoff

  # If you do not specify a longitude and latitude value
  # within the range of the data an error message will be
  # returned
  if(lon < min(dat$Pickup_longitude) |
       lon > max(dat$Pickup_longitude) |
       lat < min(dat$Pickup_latitude) |
       lat > max(dat$Pickup_latitude)){
    stop("The specified Longitude and Latitude do not fall in the range of
         the data")
  }

  if(time == "Not Specified"){
  # Now we only count the number of people within the radius specified
  # using the function which()
  NumberInRange <- length(which(dat$Pickup_longitude >= lowerLon &
                                dat$Pickup_longitude <= upperLon &
                                dat$Pickup_latitude >= lowerLat &
                                dat$Pickup_latitude <= upperLat))
  }
```

```
  else{
    if(time < 0 | time > 23)stop("A Proper Hour for time was not specified")
  NumberInRange <- length(which(dat$Pickup_longitude >= lowerLon &
                                  dat$Pickup_longitude <= upperLon &
                                  dat$Pickup_latitude >= lowerLat &
                                  dat$Pickup_latitude <= upperLat &
                                  as.numeric(dat$Hour) == time))
    }

  return(NumberInRange)

}
```

The function `locFunc` above takes the longitude, latitude and radius (in miles) that the user wants to consider as inputs and returns the number of parties that were picked up within the circular area (computed based on the radius). This function can be implemented as follows:

```
locFunc(dat = test_dat, lon = -73.9, lat = 40.7, miles = 1)
```

This returns 2,286, which indicates that 2,286 parties are within a one mile radius of the longitude (-73.9) and latitude (40.7) values. Note that this implementation does not consider the time aspect. An implementation which considers the time aspect is shown below:

```
locFunc(dat = test_dat, lon = -73.9, lat = 40.7, miles = 1, time = 7)
```

This returns 34, which indicates that 34 pickups happened during the 7:00am hour throughout the month of September at that particular longitude and latitude value.

This function seems relevant to green taxi drivers as it is an efficient, computationally fast way to provide an idea of passenger volume picked up based on a specified location. More complexities such as the number of taxis in that particular area could further improve the applicability of this function. Nevertheless, currently the functionality can provide information to drivers to assess the potential gain of traveling farther from a specified location. It also provides beneficial information to taxi drivers on the time and locations to be close to in order to maximize the number of pickups.

Moreover, this function could serve as a basis to further build on to identify ideal starting areas for taxi drivers. In particular, this could be done via

an optimization algorithm. Finally, I want to note that this function would be made more user-friendly if one could input a location name, instead of longitude and latitude.