STATS 769 - Lab 01 - bole001

Bernard O'Leary

5 August 2019

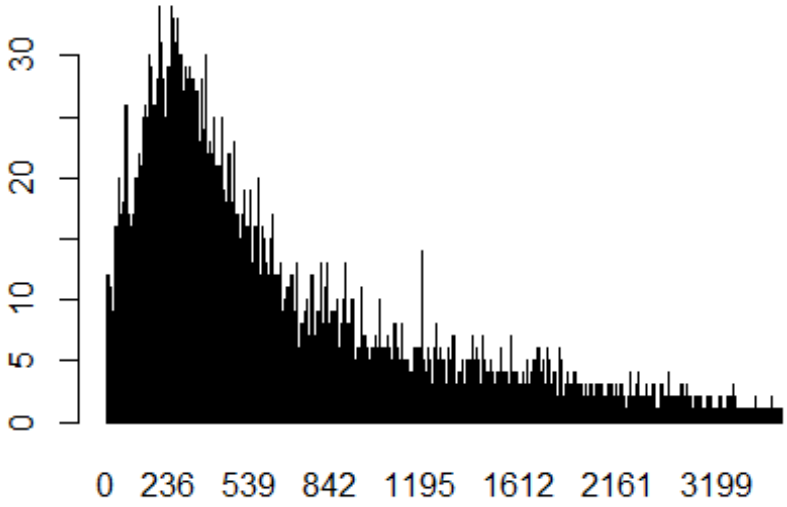
Read in data

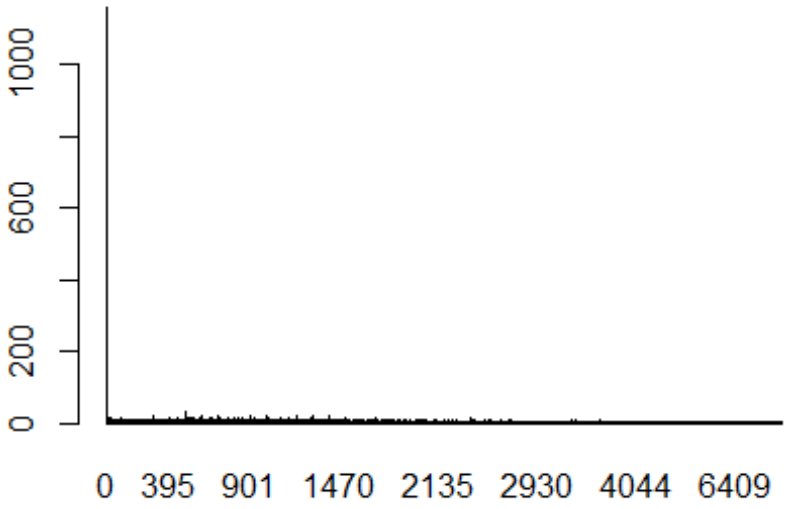
Date format is CSV (Comma Separated Values), reading into R using the read.csv function.

```
# import the data
trips201807 <- read.csv("trips-2018-7.csv")
trips201808 <- read.csv("trips-2018-8.csv")
trips201809 <- read.csv("trips-2018-9.csv")
trips <- rbind(trips201807, trips201808, trips201809)
```

Explore duration and distance

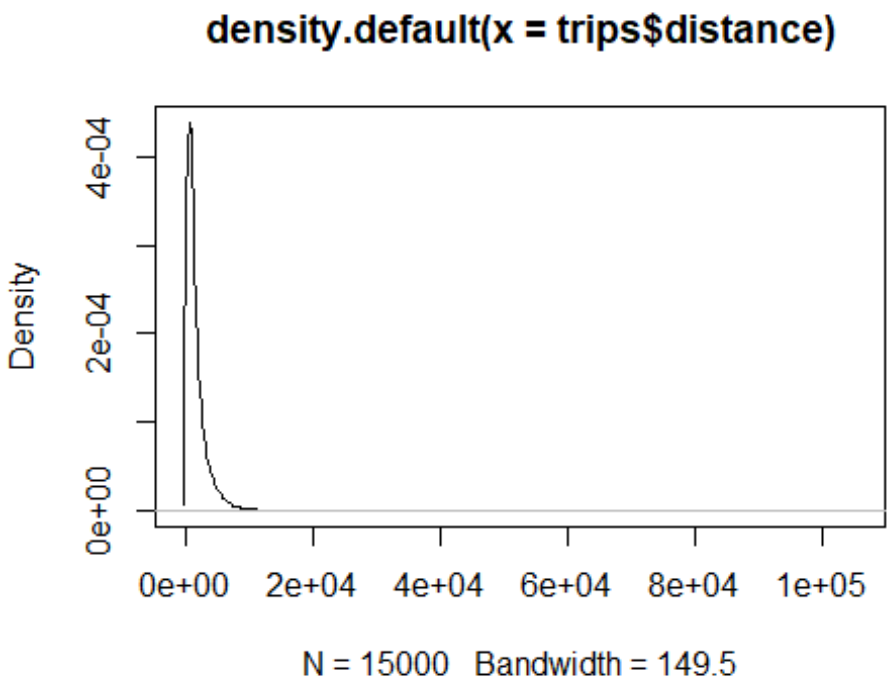
Visualise the distance (predictor) and duration (response) features by generating a barplot of their respective values. Both variables appear to be right skewed.

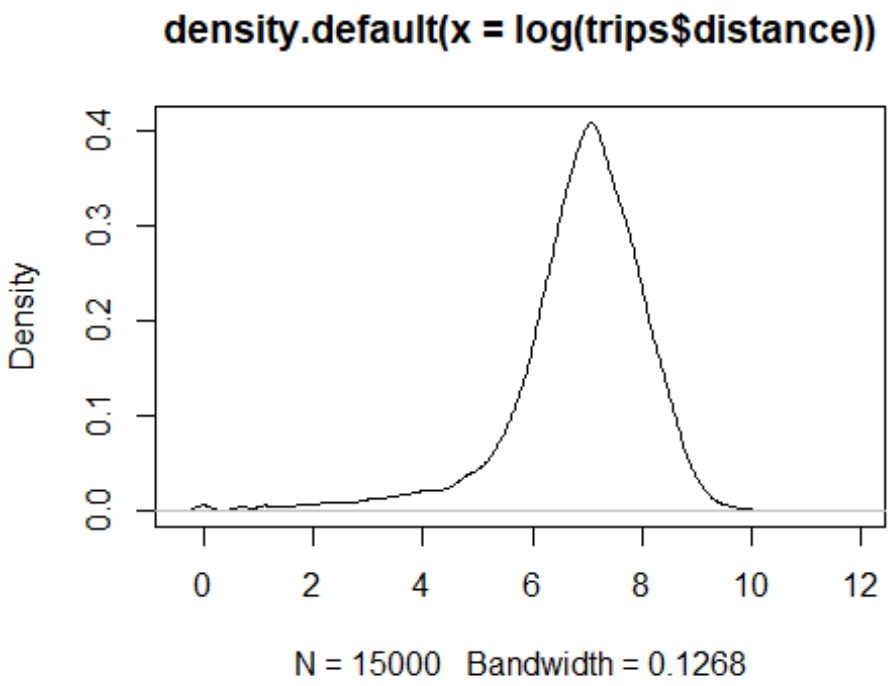

0
5
10
20
30
0
236
539
842
1195
1612
2161
3199


0
200
600
1000
0
395
901
1470
2135
2930
4044
6409

Explore duration and density further

To get a better understanding of the distance feature we can plot the logged output of the density function. This shows us the distribution of the variable. The logged distribution of the distance data appears to be left skewed.


density.default(x = trips$distance)
Density
0e+00
2e-04
4e-04
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
N = 15000 Bandwidth = 149.5


density.default(x = log(trips$distance))
Density
0.0
0.1
0.2
0.3
0.4
0
2
4
6
8
10
12
N = 15000 Bandwidth = 0.1268

Train model

Use the plyr package to split the records evenly by month (1000 of each) and take samples. Use month as divider and take a random sample of 0.8 of the dataset split evenly across the months for the training set. Capture the remainder (0.2) of the split in the second part of the output for the test set.

The plyr package can be found here:
https://www.rdocumentation.org/packages/plyr/versions/1.8.4/topics/dlply

The sample() function enables a random selection of data from an R dataframe.

```
# use the plyr library to split the records evenly and take samples
# use month as divider and take a random sample of 0.8 of the dataset split evenly across the months
# capture the remainder (0.2) of the split in the second part of the output
# https://www.rdocumentation.org/packages/plyr/versions/1.8.4/topics/dlply
# https://stackoverflow.com/questions/18258690/take-randomly-sample-based-on-groups
# https://stackoverflow.com/questions/18942792/training-and-test-set-with-respect-to-group-affiliation
library(plyr)

## Warning: package 'plyr' was built under R version 3.6.1

split_set = dlply(trips, .(month), function(.) { s = sample(1:nrow(.), trunc(nrow(.) * 0.8)); list(.[s, ], .[-s,]) } )

# train/test split
# https://www.rdocumentation.org/packages/plyr/versions/1.8.4/topics/ldply
training_set = ldply(split_set, function(.) .[[1]])
test_set = ldply(split_set, function(.) .[[2]])

# make the model
y_train = training_set$duration
x_train = training_set$distance
fit_mean = mean(y_train)
fit_lm <- lm(y ~ x, data.frame(y=y_train, x=x_train))

# test results
y_test <- test_set$distance
x_test <- test_set$duration

# make a vector that is populated with the fit_mean
pred_mean <- rep(fit_mean, length(y_test))

# get a vector of predictions based on test data
pred_lm <- predict(fit_lm, data.frame(x=x_test))
```

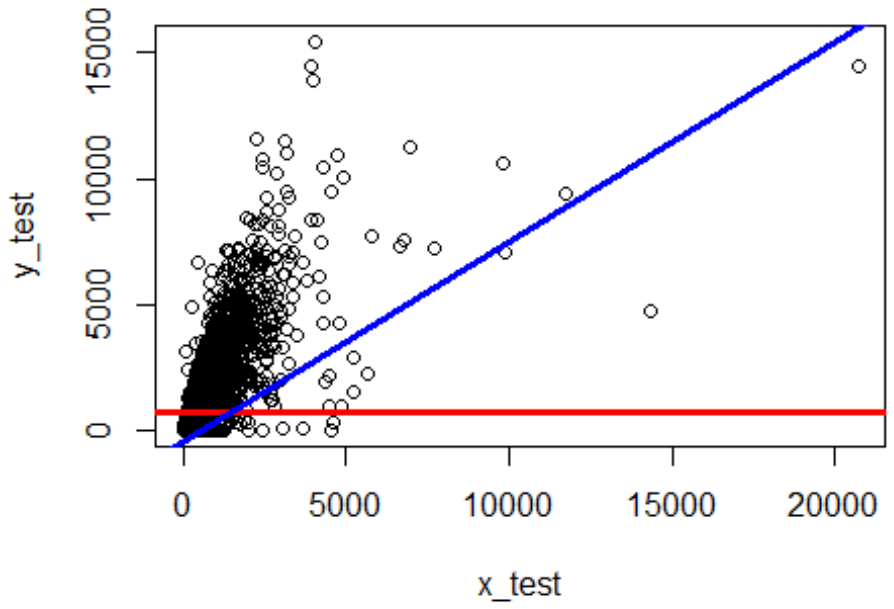
Define RMSE function and evaluate using test set

Take the RMSE (root mean square error) of the predicted mean and the predicted linear model. The linear model gives us a slightly higher RMSE than the predicted mean (the predicted mean is an average model with random noise).

```
# define RMSE function
RMSE <- function(m, o) {
    sqrt(mean((m - o)^2))
}

# get the RMSE
RMSE(pred_mean, y_test)

## [1] 1896.576

RMSE(pred_lm, y_test)

## [1] 1950.925
```

Visualise the model

Plot y_test vs x_test and overlay the average with random error model and our derived linear model. This is to help us visualise the data and the derived model as it applies to the test dataset.



Summary and analysis

Our model has not reduced the variability that we see by simply taking the average model with random error - in-fact it slightly increases the variablity. This is not a good model as it explains less of the variability in the data than the average model with random error. This is possibly because the shape of the data is not linear.