Capital One Data Science Challenge

Yuchong Zhang

December 15, 2017

I have worked through all the questions in the Capital One Data Science Challenge. For fast knit purpose, I have commented out several codes and instead loaded previously saved data from my computer for some steps. However, I have put all the executable code in a seperate R file.

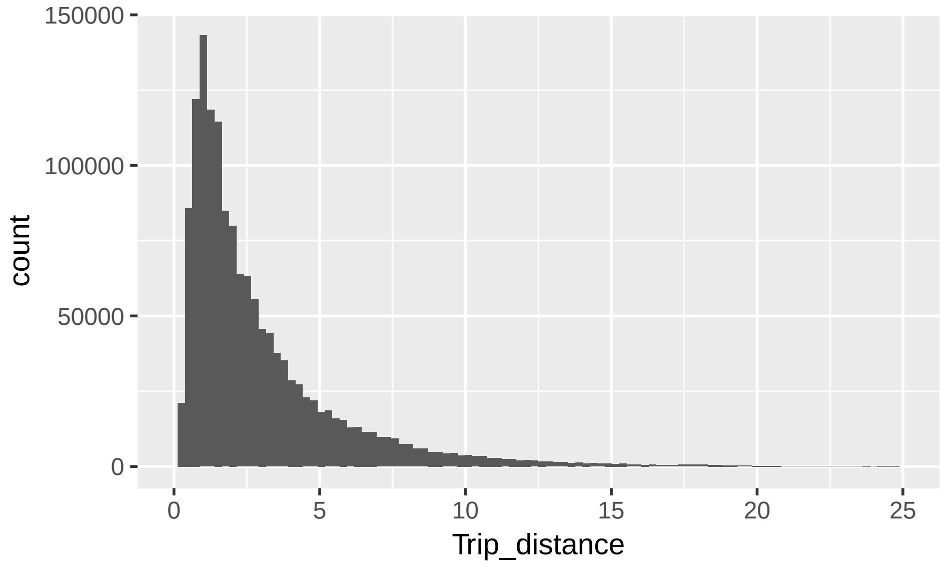
Question 1

```
#require(data.table)
#green_trip<-fread("https://s3.amazonaws.com/nyc-tlc/trip+data/green_tripdata_2015-09.csv")
setwd("/Users/Yuchong/Documents/capital one")
green_trip= as.data.frame(read.csv(file="green_tripdata_2015-09.csv",header=TRUE,stringsAsFactors = FALSE))
nrow(green_trip)

## [1] 1494926

ncol(green_trip)

## [1] 21
```

The data frame has 1494926 rows and 21 columns.

Question 2

```
require(dplyr)
require(ggplot2)
ggplot(green_trip, aes(Trip_distance)) +
  geom_histogram(bins = 100)+
  xlim(0,25)

## Warning:  Removed 1225 rows containing non-finite values (stat_bin).
```



```
summary(green_trip$Trip_distance)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.100   1.980   2.968   3.740 603.100
```

From the histogram plot, we find that the distribution of Trip_distance is right skewed. Combining the knowledge from the summary, we conclude that most of the trips are quite short, perhaps for a long travel the taxi fares become unaffordable. The 3rd quantile is 3.740, therefore the histogram truncated between 0 and 25 is adeqaute to show the whole picture of the distribution. This distribution struture may come from lognormal distribution or gamma distribution.

Question 3

```
green_trip %>%
mutate(pickup_hour=as.integer(substr(lpep_pickup_datetime, 12, 13)))->green_trip_2
green_trip_2$pickup_hour<-as.factor(green_trip_2$pickup_hour)
green_trip_2%>%
  group_by(pickup_hour)->green_trip_2_grouped
print(summarise(green_trip_2_grouped,distance_mean=mean(Trip_distance),distance_median=median(Trip_distance)),n=24)

## # A tibble: 24 x 3
##    pickup_hour distance_mean distance_median
##         <fctr>         <dbl>           <dbl>
##  1           0      3.115276            2.20
##  2           1      3.017347            2.12
##  3           2      3.046176            2.14
##  4           3      3.212945            2.20
##  5           4      3.526555            2.36
##  6           5      4.133474            2.90
##  7           6      4.055149            2.84
##  8           7      3.284394            2.17
##  9           8      3.048450            1.98
## 10           9      2.999105            1.96
## 11          10      2.944482            1.92
## 12          11      2.912015            1.88
## 13          12      2.903065            1.89
## 14          13      2.878294            1.84
## 15          14      2.864304            1.83
## 16          15      2.857040            1.81
## 17          16      2.779852            1.80
## 18          17      2.679114            1.78
## 19          18      2.653222            1.80
## 20          19      2.715597            1.85
## 21          20      2.777052            1.90
## 22          21      2.999189            2.03
## 23          22      3.185394            2.20
## 24          23      3.191538            2.22
```

The mean and median trip distance grouped by hour of day is summarized as above.

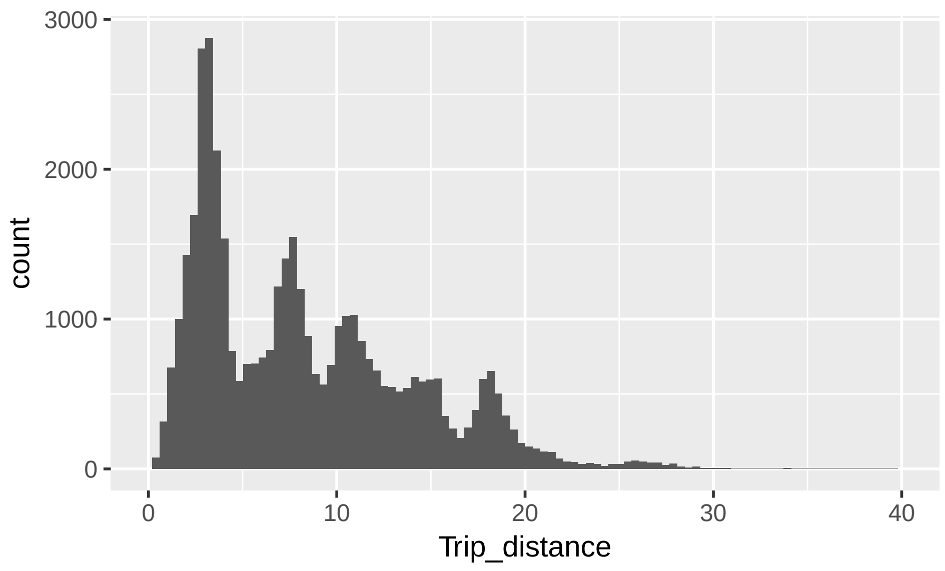
To identify trips that orginate or terminate at one of the NYC area airports, I use ggmap::geocode to locate the longitude and latitude of three main airports (JFK, EWR, LGA). Note that although EWR is in New Jersey, it is usually considered as one main airport in the New York area. From the data, we know the location (longitude and latitude) of pickup and dropoff points so we can calculate the distance from pickup or dropoff location to any of these airports to determine whether a trip originates or terminates at an airport. geosphere::distm is used return the shortest distance between two points on earth. The next question is how close that distance needs to be for us to consider it as actually being in the airport. I look up the walking distance (the walking route is more straight than the driving route and therefore more consistent with shortest distance as we calculate) between different terminals of JFK on google map and find 1 mile (1609m) can be a good threshold. I also look up the area of JFK, which is 2.62 mile * 2.62 mile by square. The half side length is approximately 1.31 mile. Again, 1 mile (1609m) looks like a good cutoff distance.

```
#require(ggmap)
#Airport_name <- c("John F. Kennedy International Airport,NY","Newark Liberty #International Airport,NY","LaGuardi
#Airport_location <- geocode(Airport_name)
#Airport_location£lon
#Airport_location£lat
#require(geosphere)
#JFK<-c(Airport_location£lon[1],Airport_location£lat[1])
#EWR<-c(Airport_location£lon[2],Airport_location£lat[2])
#LGA<-c(Airport_location£lon[3],Airport_location£lat[3])
#green_trip %>%
#  mutate(JFK_P=distm(cbind(Pickup_longitude,Pickup_latitude),JFK,fun=distHaversine#),JFK_D=distm(cbind(Dropoff_lon
#         EWR_P=distm(cbind(Pickup_longitude,Pickup_latitude),EWR,fun=distHaversine#),EWR_D=distm(cbind(Dropoff_lon
#         LGA_P=distm(cbind(Pickup_longitude,Pickup_latitude),LGA,fun=distHaversine#),LGA_D=distm(cbind(Dropoff_lon
#green_trip_3£JFK_P<-as.vector(green_trip_3£JFK_P)
#green_trip_3£JFK_D<-as.vector(green_trip_3£JFK_D)
#green_trip_3£EWR_P<-as.vector(green_trip_3£EWR_P)
#green_trip_3£EWR_D<-as.vector(green_trip_3£EWR_D)
#green_trip_3£LGA_P<-as.vector(green_trip_3£LGA_P)
#green_trip_3£LGA_D<-as.vector(green_trip_3£LGA_D)
load(file="/Users/Yuchong/Documents/capital one/green_trip_3.Rdata")
green_trip_3_airport<-filter(green_trip_3, JFK_P<=1609|JFK_D<=1609|EWR_P<=1609|EWR_D<=1609|LGA_P<=1609|LGA_D<=1609)
nrow(green_trip_3_airport)

## [1] 42572

nrow(green_trip_3_airport)/nrow(green_trip_3)

## [1] 0.02847766

summary(green_trip$Total_amount)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -475.00    8.16   11.76   15.03   18.30  581.30

summary(green_trip_3_airport$Total_amount)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -45.00   14.80   29.34   32.34   44.55  450.00

green_trip_3_airport_new<-filter(green_trip_3_airport, Total_amount>=0)
summary(green_trip_3_airport_new$Total_amount)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   14.80   29.34   32.36   44.59  450.00
```

There are 42572 trips associated with the airport and they constitute 2.85% of all the trips. The average fare (Here I refer to Total_amount) is $32.34, which almost doubles the average fare for all trips ($15.03). It is unexpecetd that the fare can be negative so I further remove observations that have negative Total_amount. The summary changes litte since these negative observations are minority and the average fare is slightly increased to $32.34.

We explore the trip distance in Question 2, so it is interesting to see what the distribution of trip distance for the airport subset.

```
ggplot(green_trip_3_airport, aes(Trip_distance)) +
  geom_histogram(bins = 100)+
  xlim(0,40)

## Warning:  Removed 10 rows containing non-finite values (stat_bin).
```



Unlike the single-peak distribution for all trips, the new distribution is multi-modal, peak height decaying from left to right. This may be a result of superposition of distributions of three individual airports or simply a result of lack of statistics. Given more time I would try to figure out why the structure is like this.

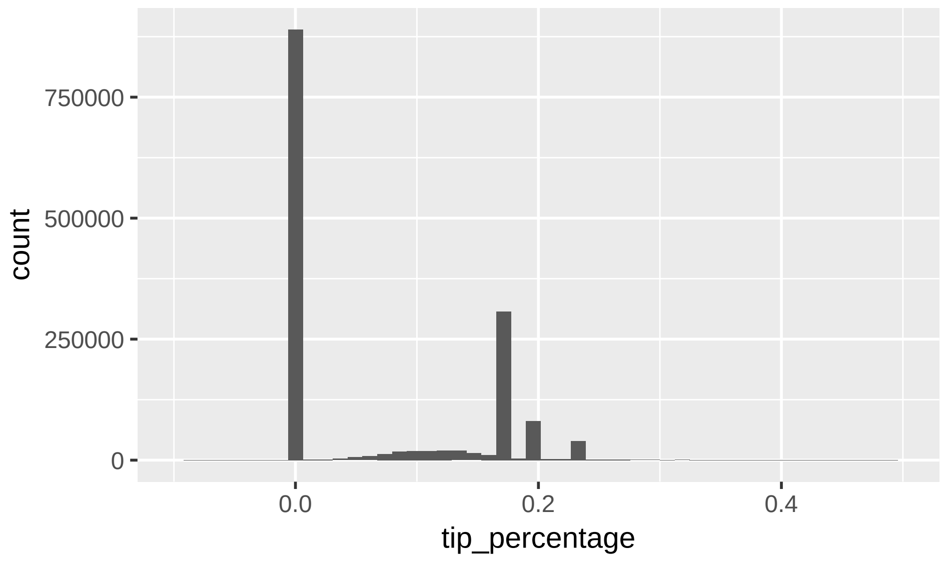
Another interesting thing to take a look is the contribution of trips from different airports.

```
green_trip_3_JFK<-filter(green_trip_3, JFK_P<=1609|JFK_D<=1609)
green_trip_3_EWR<-filter(green_trip_3, EWR_P<=1609|EWR_D<=1609)
green_trip_3_LGA<-filter(green_trip_3, LGA_P<=1609|LGA_D<=1609)
nrow(green_trip_3_JFK)

## [1] 12933

nrow(green_trip_3_EWR)

## [1] 714

nrow(green_trip_3_LGA)

## [1] 28990

n_2airports=nrow(green_trip_3_JFK)+nrow(green_trip_3_EWR)+nrow(green_trip_3_LGA)-nrow(green_trip_3_airport)
n_2airports

## [1] 65

n_2airports/nrow(green_trip_3_airport)

## [1] 0.001526825
```

We find that trips associated with JFK, EWR and LGA are 12933, 714 and 28990, respectively. It is quite unexpected that LGA ranks the first since it has the least passengers from web. One explanation is that taxi is more inevitable for LGA. After some study it is the case as there are AirTrain service for JFK and EWR, which can be cheaper but also fast and convenient. However, in LGA, public transport is poor and furthermore, LGA is more closer to Manhattan, so taxi is affordable. I am also curious about how many trips actually connect between two different airports since I have experiences that when booking flights there is warning that you need to go a nearby airport for connection. The number is 65, only making up 0.15% of all airport related trips.

Question 4

```
green_trip %>%
  mutate(tip_percentage=Tip_amount/Total_amount)->green_trip_4
ggplot(green_trip_4, aes(tip_percentage)) +
  geom_histogram(bins = 50)+
  xlim(-0.1,0.5)

## Warning:  Removed 6363 rows containing non-finite values (stat_bin).
## Warning:  Removed 1 rows containing missing values (geom_bar).
```

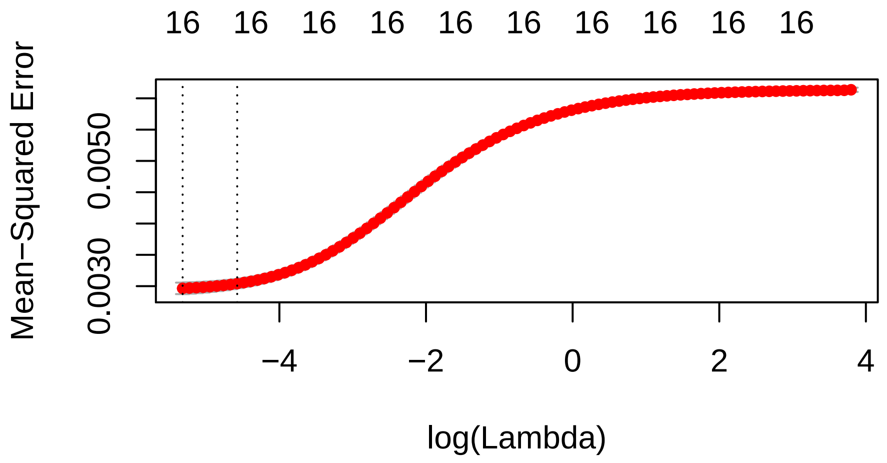


I create a new variable tip_percentage to represent tip as a percentage of the total fare. The histogram of tip_percentage shows some interesting features. The majority of data fall between the range of 0 and 0.25. There are several values of tip_percentage (approximately 0, 0.17, 0.2, 0.23) that are of high frequency. And other data points are weekly centered around value of approximately 0.12. From the description of varaibles, we know that only credit cards payments contribute to tips. So we can say that there are some contributions to the tip_percentage=0 from observations with cash payment. And the exsitence of several frequent values of tip_percentage are probably from the process of automatic tipping which provide several fixed percentage options (including no tips). Passengers who choose to tip manually contribute to tip_percentage of various values.

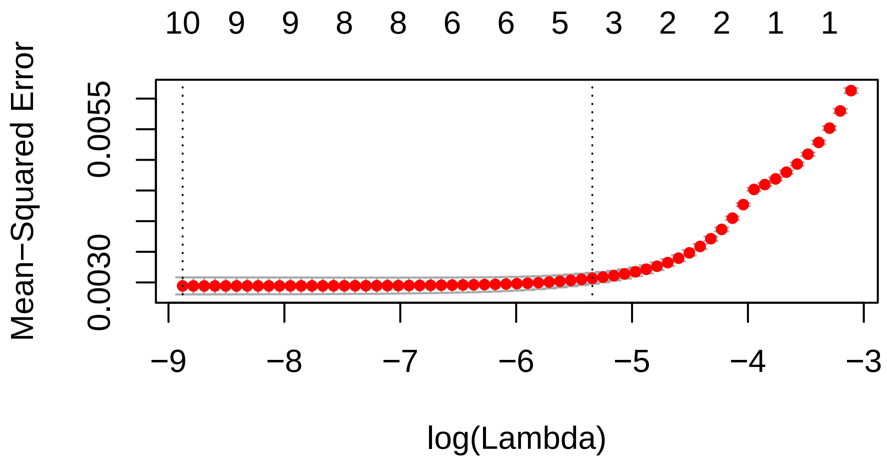
To predict tip_percentage, I firstly filter the observations through Payment_type, and only study the observations using credit card payment. For prediction purposes, the observations using other payment methods (such as cash) should automatically give a tip_percentage of 0. Then I exclude several variables from the data that are not useful for the prediction. They are the location information of pickup and dropoff points and Ehail_fee. I also remove rows with missing values and create a model matrix, which creates dummy variables for quatitative variables. I randomly split the model into 60% traing set and 40% testing set. For the training set, I tried ridge regression and lasso regression respectively and used cross-validation to get the best parameter. Then the fitted model is applied to the test data for a comparison of prediction accruacy. Here I use mean squared error as the criteria.

```r
green_trip_4 %>%
  filter(Total_amount>0&Payment_type==1)->green_trip_4_credit
vars <- names(green_trip_4_credit) %in% c("Ehail_fee")
green_trip_4_credit<-green_trip_4_credit[!vars]
green_trip_4_credit <- na.omit(green_trip_4_credit)
vars <- names(green_trip_4_credit) %in% c("lpep_pickup_datetime","Lpep_dropoff_datetime","tip_percentage","Pickup_l
green_trip_4_credit$VendorID<-as.factor(green_trip_4_credit$VendorID)
green_trip_4_credit$Store_and_fwd_flag<-as.factor(green_trip_4_credit$Store_and_fwd_flag)
green_trip_4_credit$RateCodeID<-as.factor(green_trip_4_credit$RateCodeID)
green_trip_4_credit$Trip_type<-as.factor(green_trip_4_credit$Trip_type)
green_trip_4_credit_X<-green_trip_4_credit[!vars]
green_trip_4_credit_Y<-green_trip_4_credit["tip_percentage"]

green_trip_4_credit_X<-model.matrix(~., green_trip_4_credit_X)[,-1]
smp_size <- floor(0.6 * nrow(green_trip_4_credit_X))

set.seed(1)
train_ind <- sample(seq_len(nrow(green_trip_4_credit_X)), size = smp_size)
xtrain<-green_trip_4_credit_X[train_ind,]
ytrain<-green_trip_4_credit_Y[train_ind,]
xtest<-green_trip_4_credit_X[-train_ind,]
ytest<-green_trip_4_credit_Y[-train_ind,]

require(glmnet)

#ridge
grid =10^seq(10,-2, length =100)
```

```r
ridge.mod =glmnet(xtrain,ytrain,alpha =0, lambda =grid)
cv.out =cv.glmnet (xtrain,ytrain,alpha =0)
plot(cv.out)
```



```r
bestlam_rigde =cv.out$lambda.min
ridge.pred=predict (ridge.mod ,s=bestlam_rigde ,newx=xtest)
mean((ridge.pred -ytest)^2)

## [1] 0.003030125

#lasso
grid =10^seq(10,-2, length =100)
lasso.mod =glmnet(xtrain,ytrain,alpha =1, lambda =grid)
cv.out =cv.glmnet (xtrain,ytrain,alpha =1)
plot(cv.out)
```



```r
bestlam_lasso =cv.out$lambda.min
lasso.pred=predict(lasso.mod ,s=bestlam_lasso,newx=xtest)
mean((lasso.pred -ytest)^2)

## [1] 0.003382463
```

```
#baseline
mean((mean(ytest) -ytest)^2)

## [1] 0.006144964
```

From the results we see that the MSE for ridge regression and lasso regression are 0.003030 and 0.003382 respectively, both better than the 0.006145 of the base model, which simply assigns the average tip_percentage for each prediction. Ridge performs slightly better than the lasso, so to build the predictive model, I use all the data to fit a ridge regression with the best parameter from CV.

```
ridge.mod_new =glmnet(green_trip_4_credit_X,green_trip_4_credit_Y[,],alpha =0, lambda =grid)
#ridge.pred=predict (ridge.mod_new ,s=bestlam_rigde ,newx="new test data")
```

Substitute the new data into the "new test data" (certainly after several manipulation of data) and the prediction can be obtained.

If given more time I will work on a function to automate the process (Payment_type recoginization, data tidying and prediction...). And I also want to try some other prediction methods (such as SVM) and ensemble learners (such as xgboosting) to see whether a better predictive model can be set up. Another idea is to use a hierarchical predictive model considering about the distribution of tip_percentage. That is, the first layer tells apart who use the fixed tiping percentage and who manually tip a certain amount. Then in each category, a different prediction model can be built to predict tip_percentage. Again, I have no time to further elaborate on this.

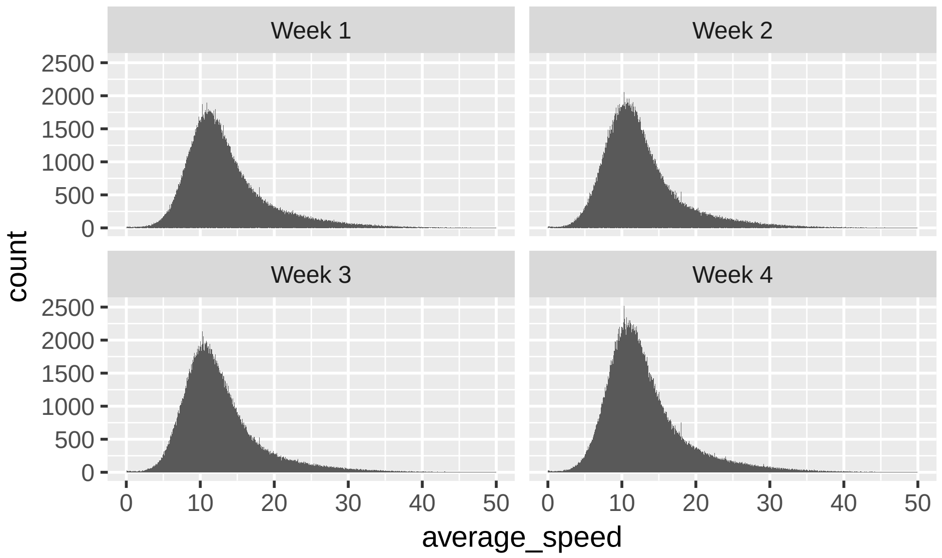
Question 5

I chose Option A: Distributions.

```
green_trip %>%
  filter(Trip_distance>0)%>%
  filter(substr(lpep_pickup_datetime, 9, 10)==substr(Lpep_dropoff_datetime, 9, 10))%>%
  mutate(time_elapsed=(as.integer(substr(Lpep_dropoff_datetime, 12, 13))-as.integer(substr(lpep_pickup_datetime, 12
  filter(time_elapsed>=60)%>%
  mutate(average_speed=Trip_distance/time_elapsed*3600)->green_trip_5
```

I create a variable average_speed to represent the average speed over the course of a trip. I have made two approximations. The first one is that I remove the trips which span two days, due to the complexity when calculating trip time and these trips are only a small part of all the trips. The second one is that I remove all the trips that are shorter than 1 min because these trip data are usually suspicious (there are even trips that last 0 second but have positive trip distance!) and again, these trips are only a small part of all the trips.

```
green_trip_5<-mutate(green_trip_5,date=as.integer(substr(lpep_pickup_datetime, 9, 10)))
green_trip_5$week<-cut(green_trip_5$date,
      breaks = c(1, 7, 14, 21, 30),
      labels = c("Week 1", "Week 2", "Week 3", "Week 4"),
      include.lowest = TRUE)
green_trip_5%>%
  group_by(week)->green_trip_5_grouped

print(summarise(green_trip_5_grouped,average_speed_mean=mean(average_speed)))

## # A tibble: 4 x 2
##     week average_speed_mean
##   <fctr>              <dbl>
## 1 Week 1           13.63274
## 2 Week 2           12.68480
## 3 Week 3           12.77174
## 4 Week 4           13.19728

ggplot(data = green_trip_5) +
  geom_histogram(aes(average_speed),bins=1000) +
  xlim(0,50)+
  facet_wrap(~week)

## Warning:  Removed 530 rows containing non-finite values (stat_bin).
```



From the numerical summary, the mean average trip speeds for different weeks are not the same. To prove average trip speeds are materially different in all weeks of September, I notice that the average_speed distribution is close to normal distribution. Therefore I can use t-test for hypothesis testing. For example, for Week 1 and Week 2, the null hypothesis is average trip speeds are the same for Week 1 and Week 2 and the alternative hypothesis is average trip speeds not the same for Week 1 and Week 2. I assume different variance for these two distributions.

```
green_trip_5_week1<-filter(green_trip_5,week=="Week 1")
green_trip_5_week2<-filter(green_trip_5,week=="Week 2")
green_trip_5_week3<-filter(green_trip_5,week=="Week 3")
green_trip_5_week4<-filter(green_trip_5,week=="Week 4")
#Week 1 and 2
t.test(green_trip_5_week1$average_speed, green_trip_5_week2$average_speed,paired = FALSE, var.equal = FALSE)

##
##  Welch Two Sample t-test
##
## data:  green_trip_5_week1$average_speed and green_trip_5_week2$average_speed
## t = 57.753, df = 660260, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.9157688 0.9801087
## sample estimates:
## mean of x mean of y
##  13.63274  12.68480
```

The p-value is smaller 2.2e-16, so we choose to believe tha alternative hypothesis. Similarly, we test for other week pairs.

```
#Week 1 and 3
t.test(green_trip_5_week1$average_speed, green_trip_5_week3$average_speed,paired = FALSE, var.equal = FALSE)

##
##  Welch Two Sample t-test
##
## data:  green_trip_5_week1$average_speed and green_trip_5_week3$average_speed
## t = 52.226, df = 663370, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.8286831 0.8933068
## sample estimates:
## mean of x mean of y
##  13.63274  12.77174

#Week 1 and 4
```

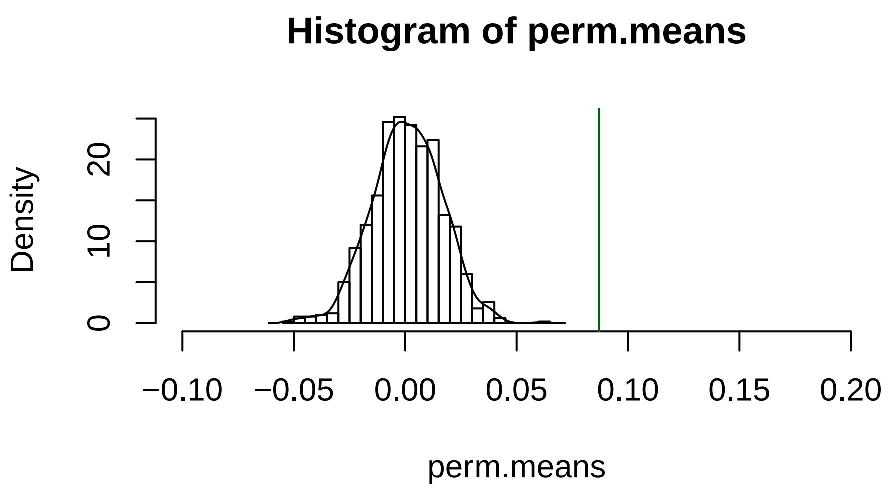
```r
t.test(green_trip_5_week1$average_speed, green_trip_5_week4$average_speed,paired = FALSE, var.equal = FALSE)

##
##  Welch Two Sample t-test
##
## data:  green_trip_5_week1$average_speed and green_trip_5_week4$average_speed
## t = 23.781, df = 741670, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.3995710 0.4713501
## sample estimates:
## mean of x mean of y
##  13.63274  13.19728

#Week 2 and 3
t.test(green_trip_5_week2$average_speed, green_trip_5_week3$average_speed,paired = FALSE, var.equal = FALSE)

##
##  Welch Two Sample t-test
##
## data:  green_trip_5_week2$average_speed and green_trip_5_week3$average_speed
## t = -5.6548, df = 696640, p-value = 1.561e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.11707858 -0.05680905
## sample estimates:
## mean of x mean of y
##  12.68480  12.77174

#Week 2 and 4
t.test(green_trip_5_week2$average_speed, green_trip_5_week4$average_speed,paired = FALSE, var.equal = FALSE)

##
##  Welch Two Sample t-test
##
## data:  green_trip_5_week2$average_speed and green_trip_5_week4$average_speed
## t = -29.592, df = 746740, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5464209 -0.4785355
## sample estimates:
## mean of x mean of y
##  12.68480  13.19728

#Week 3 and 4
t.test(green_trip_5_week3$average_speed, green_trip_5_week4$average_speed,paired = FALSE, var.equal = FALSE)

##
##  Welch Two Sample t-test
##
## data:  green_trip_5_week3$average_speed and green_trip_5_week4$average_speed
## t = -24.475, df = 749770, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.4596116 -0.3914572
## sample estimates:
## mean of x mean of y
##  12.77174  13.19728
```

All the tests show that the average trip speeds are different for all weeks.

Actually we can also ignore the normal assumption and use permutation test. With the same null and alternative hypothesis, I take Week 2 and Week 3 as an example, which look very close.

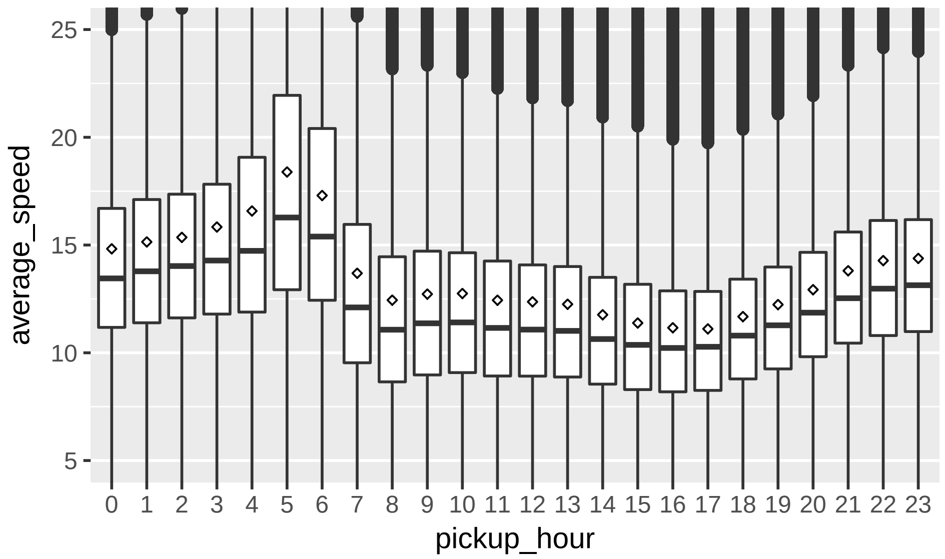
This code chunk follows and modifies that from http://t-redactyl.io/blog/2015/10/two-group-hypothesis-testing-permutation-tests.html

```
#Week2 and Week3
#green_trip_5_week23<-rbind(green_trip_5_week2,green_trip_5_week3)
#one.test <- function(grouping, variable) {
#  resampled.group <- sample(grouping)
#  mean(variable[resampled.group == "Week 3"]) -
#    mean(variable[resampled.group == "Week 2"])
#}
#set.seed(1)
#perm.means <- replicate(1000, one.test(green_trip_5_week23£week, green_trip_5_week#23£average_speed))
load(file="/Users/Yuchong/Documents/capital one/perm.means.Rdata")
mean(perm.means > mean(green_trip_5_week3$average_speed)-mean(green_trip_5_week2$average_speed))

## [1] 0

hist(perm.means,breaks='scott',freq=F,xlim=c(-0.1,0.2))
lines(density(perm.means))
abline(v=mean(green_trip_5_week3$average_speed)-mean(green_trip_5_week2$average_speed), col="darkgreen")
```



After 1000 permutations, the p-value is 0. So we have confidence saying the average trip speed for Week 2 and Week 3 are different. Similar tests can be done for other combinations but not shown here.

```
green_trip_5%>%
  mutate(pickup_hour=as.integer(substr(lpep_pickup_datetime, 12, 13)))->green_trip_5_time
green_trip_5_time$pickup_hour<-as.factor(green_trip_5_time$pickup_hour)
p <- ggplot(green_trip_5_time, aes(x=pickup_hour, y=average_speed)) +
  geom_boxplot()+coord_cartesian(ylim=c(5,25))
p + stat_summary(fun.y=mean, geom="point", shape=23, size=1)
```



From the boxplot (diamonds represent mean and bars represent median) we see that at night (20 to 7) the average_speed is greater and at daytime (8 to 19) the average_speed is smaller. So I will build up a hypothesis that the average trip speed is significantly greater at night than daytime. Further the change of average_speed at different time is similar to trigonometric function, so I propose the average_speed can be expressed as a function of time using sin function.

```
mean(green_trip_5_time$average_speed)

## [1] 13.06988

green_trip_5_time5<-filter(green_trip_5_time,pickup_hour==5)
mean(green_trip_5_time5$average_speed)

## [1] 18.39064

green_trip_5_time17<-filter(green_trip_5_time,pickup_hour==17)
mean(green_trip_5_time17$average_speed)

## [1] 11.10929

(mean(green_trip_5_time5$average_speed)-mean(green_trip_5_time17$average_speed))/2

## [1] 3.640675
```

The maximum of this sin function is 18.391 at 5 and the minimum is 11.109 at 17. The period is 24 hours and the amplitude is 3.641.