Assignment 04

Neha Patwardhan, Joy Machado

February 12, 2016

R Markdown - Assignment 04

This is a markdown report to predicts the prices of airlines in the given data set against distance or time using simple linear regression.

To predict prices for each airline, we have taken the output file from the Map Reduce program.

The Map Reduce program provides an output file consisting the carrier name, time, distance and the average price for each record. The output consisit of all the flights active in the year 2015 and is computed only for records from 2010 - 2014

On the output generated by the Map Reduce code we are trying to predict prices for airlines active in 2015 and for records from 2010 to 2014

The below is the R script to apply linear regression to predict prices against time or distance

```
require(dplyr)
```

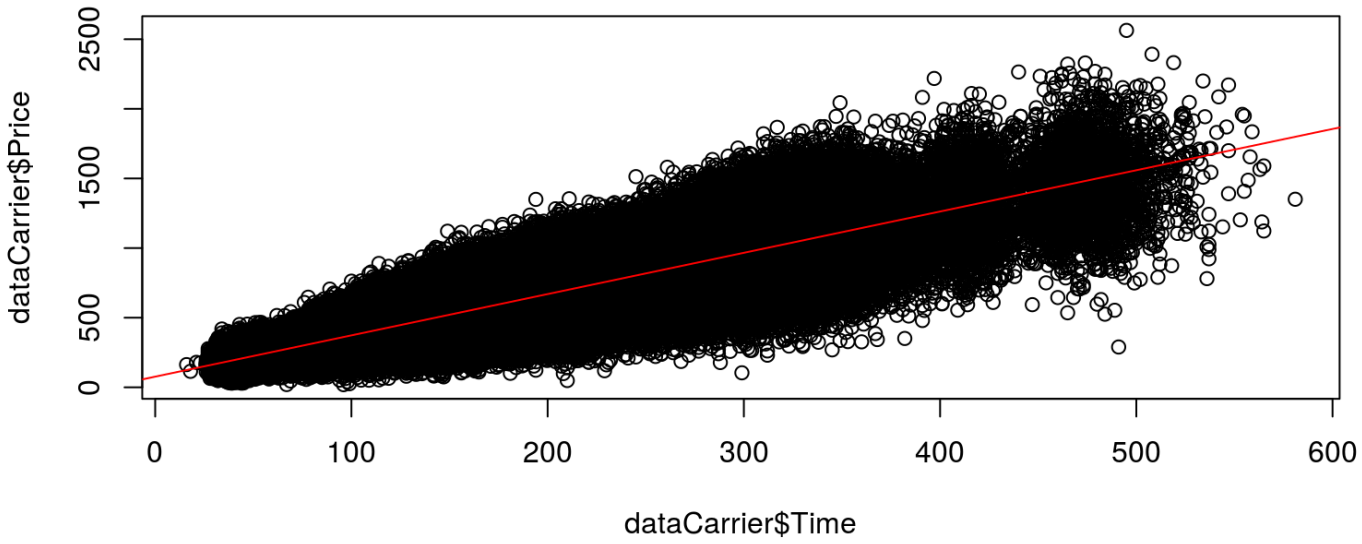
```
## Loading required package: dplyr
```

```
## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called 'dplyr'
```

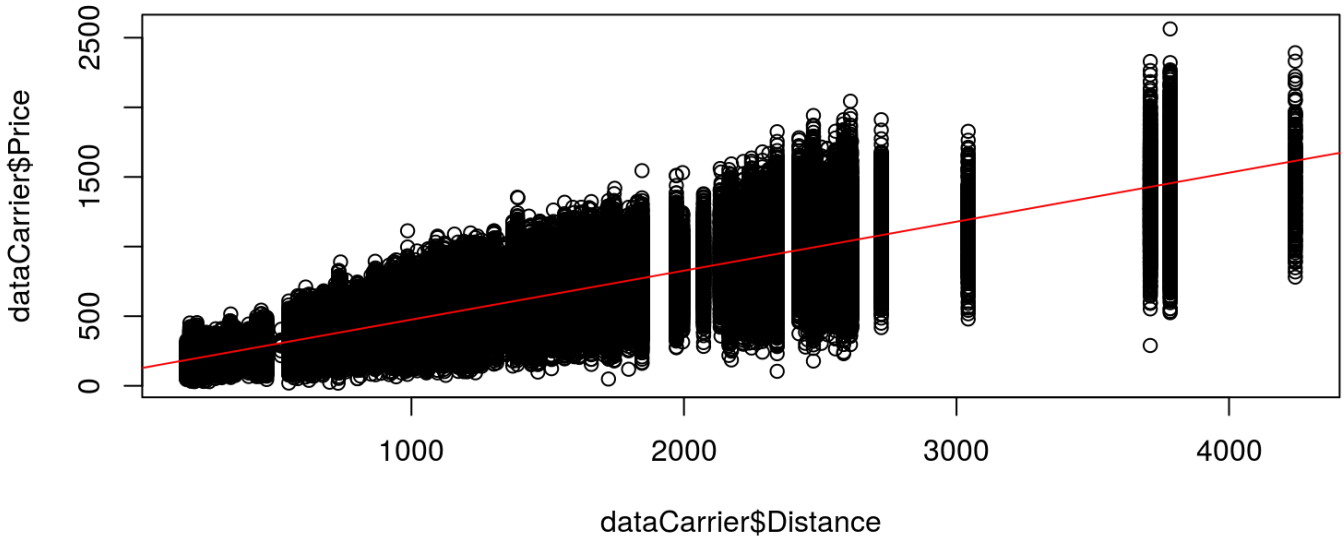
```
args <- commandArgs(TRUE)
srcFile <- "/home/shailesh/Downloads/part-r-00000"
ouputFile <- "/home/shailesh/Downloads"
data <- read.csv(srcFile,header=FALSE,sep="\t")
names(data) <- c("Carrier","Time","Distance","Price")
vectorT <- c()
vectorD <- c()
mapT <- new.env(hash=T, parent=emptyenv())
mapD <- new.env(hash=T, parent=emptyenv())
for(i in unique(data$Carrier)){
  #extract data for a particular carrier
  dataCarrier <- data[which(data$Carrier == i), ]

  #linear regression
  dataLRT <- lm(dataCarrier$Price ~ dataCarrier$Time)
  dataLRD <- lm(dataCarrier$Price ~ dataCarrier$Distance)

  #find mean, slope and intercept and get a value that you can compare to find cheapest a
irline
  meanTimeCarrier <- mean(data$Time)*dataLRT$coefficients[2] + dataLRT$coefficients[1]
  meanDistanceCarrier <- mean(data$Distance)* dataLRD$coefficients[2] + dataLRD$coefficie
nts[1]

  key <- i
  #time map
  mapT[[key]] = meanTimeCarrier
  #distance map
  mapD[[key]] = meanDistanceCarrier

  sm1 <- summary(dataLRT)
  sm2 <- summary(dataLRD)

  #find mean squared error
  t <- mean(sm1$residuals^2)
  d <- mean(sm2$residuals^2)

  #choosing whether we want to select distance or time as best measurement
  if(t < d){
    vectorT <- c(vectorT,t)
  }
  else{
    vectorD <- c(vectorD,d)
  }

  par(mfrow=c(2,1))

  plot(dataCarrier$Time,dataCarrier$Price,main = (paste("Time Vs Price for Carrier",sep
=",",i)))
  abline(dataLRT,col='red')
```

```
  plot(dataCarrier$Distance,dataCarrier$Price,main = (paste("Distance Vs Price for Carrie
r",sep=",",i)))
  abline(dataLRD,col='red')
}
```

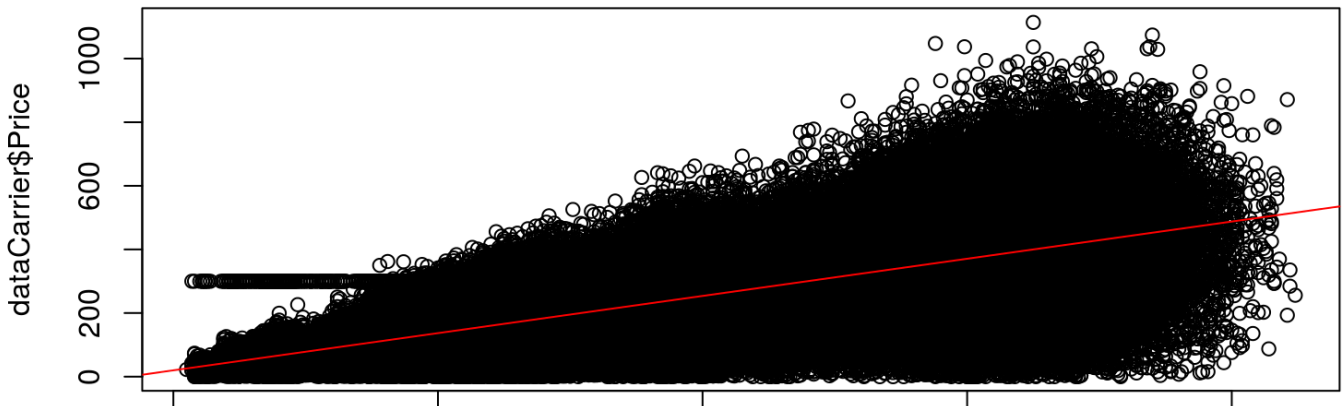
Time Vs Price for Carrier,AA

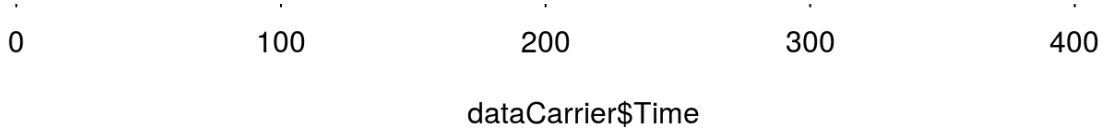


Distance Vs Price for Carrier,AA



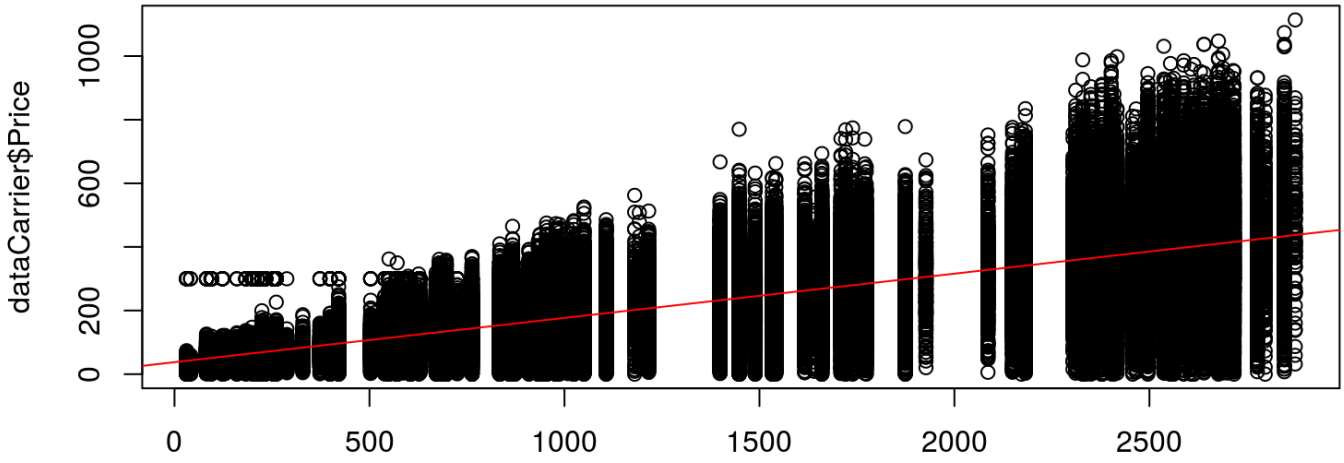
Time Vs Price for Carrier,AS





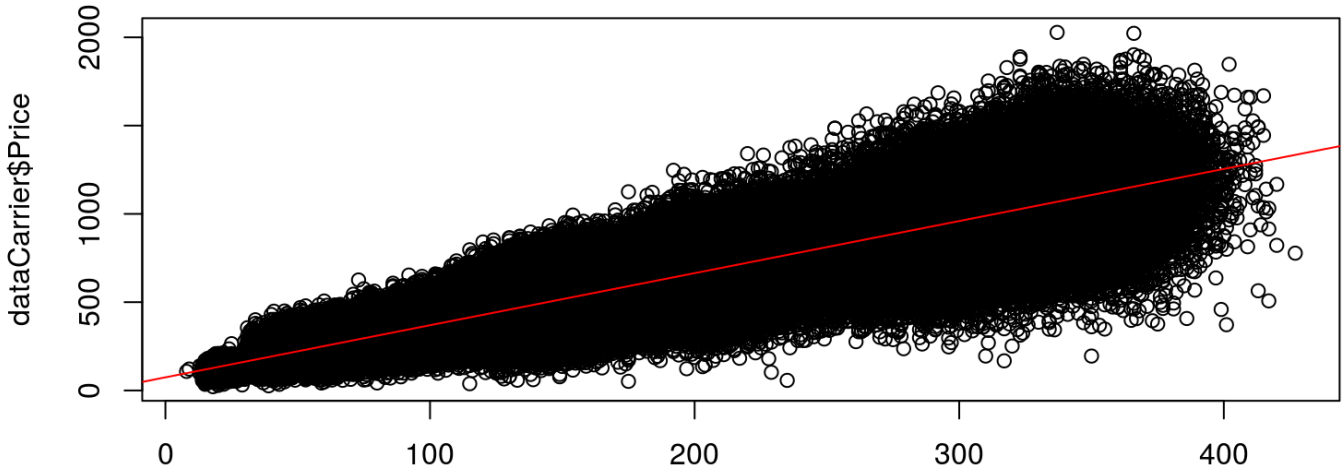
dataCarrier$Time

Distance Vs Price for Carrier,AS



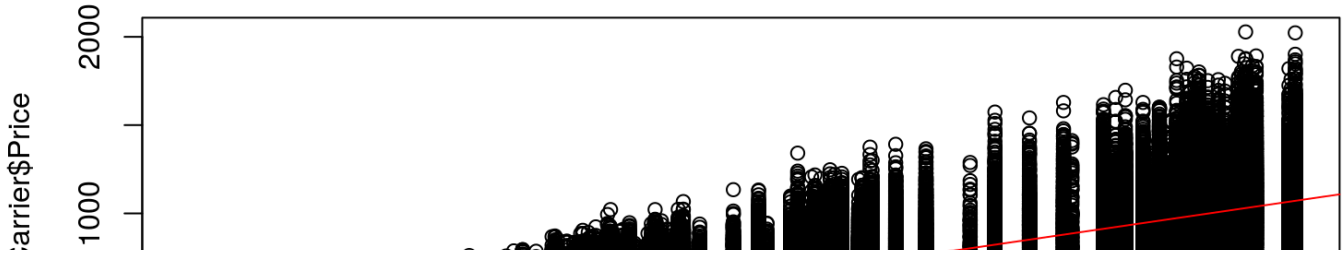
dataCarrier$Distance

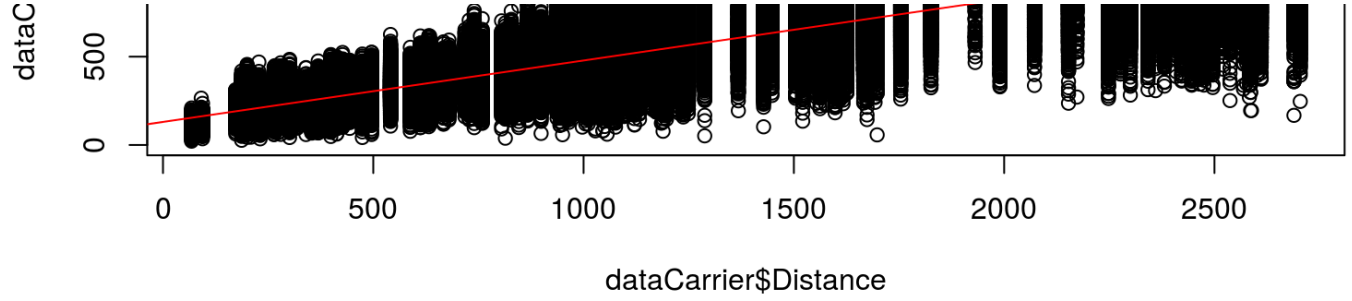
Time Vs Price for Carrier,B6



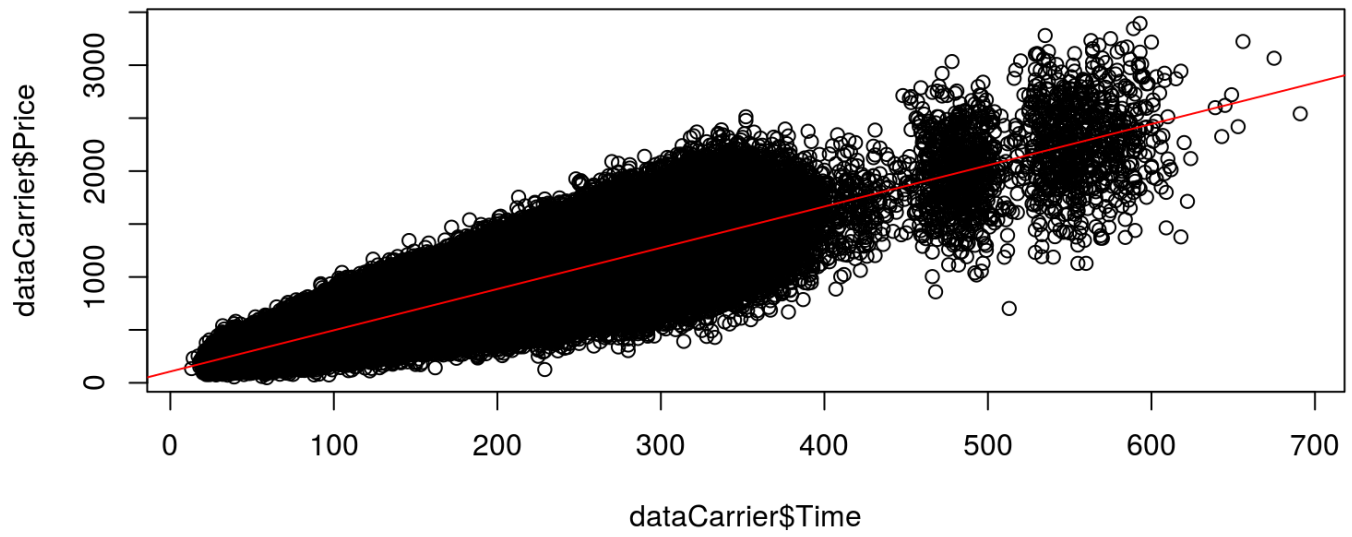
dataCarrier$Time

Distance Vs Price for Carrier,B6

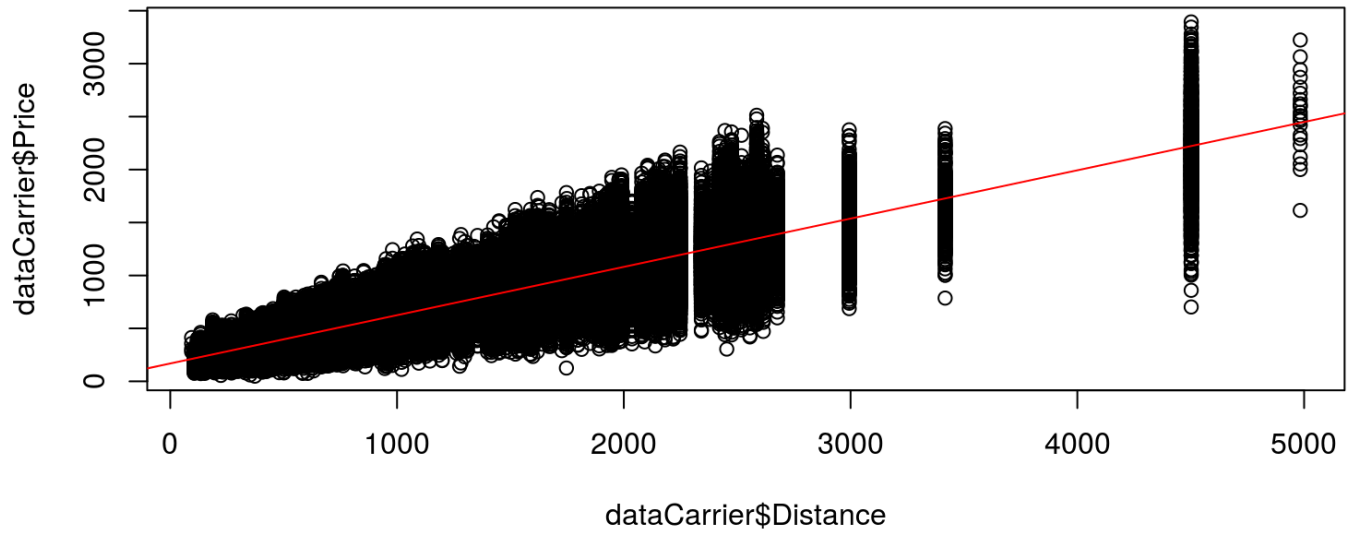




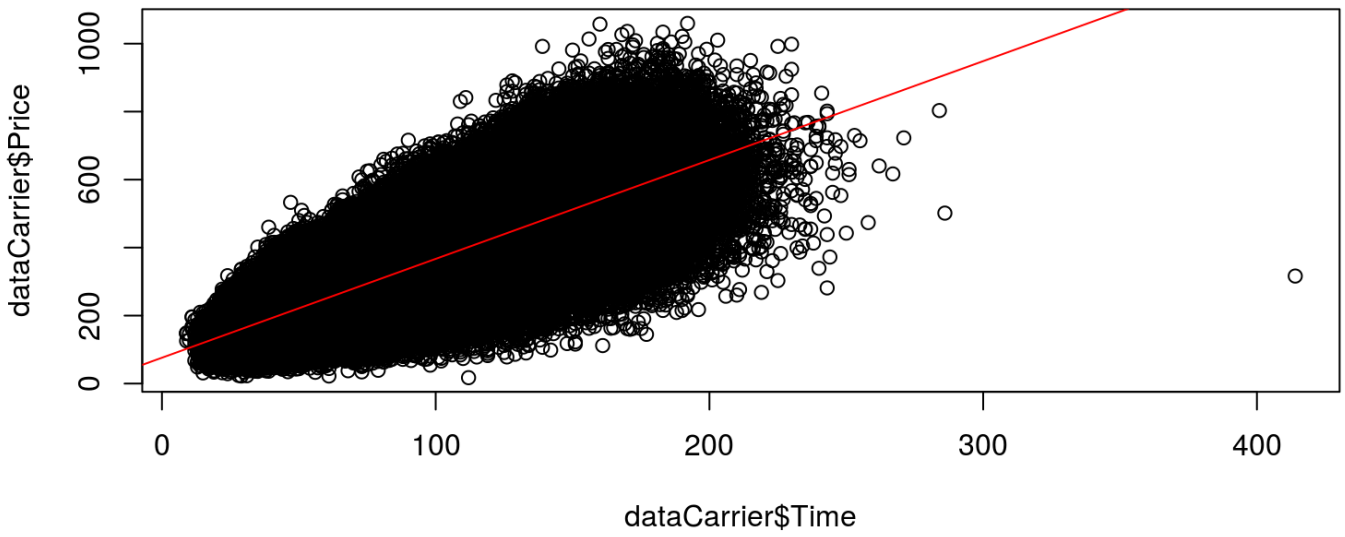
Time Vs Price for Carrier,DL



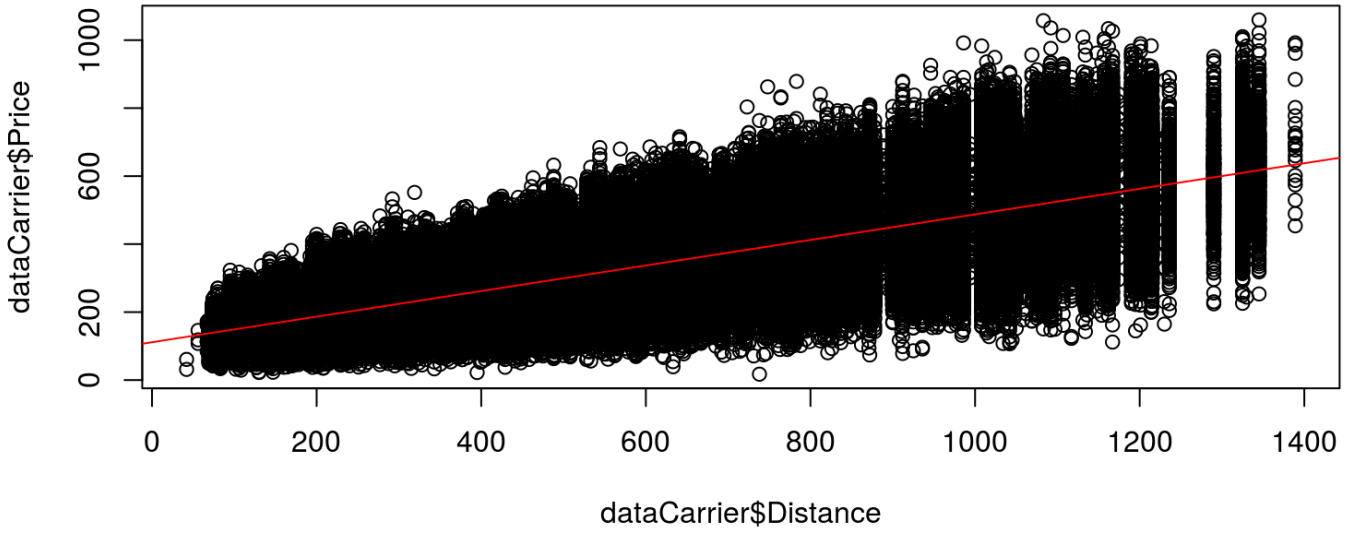
Distance Vs Price for Carrier,DL

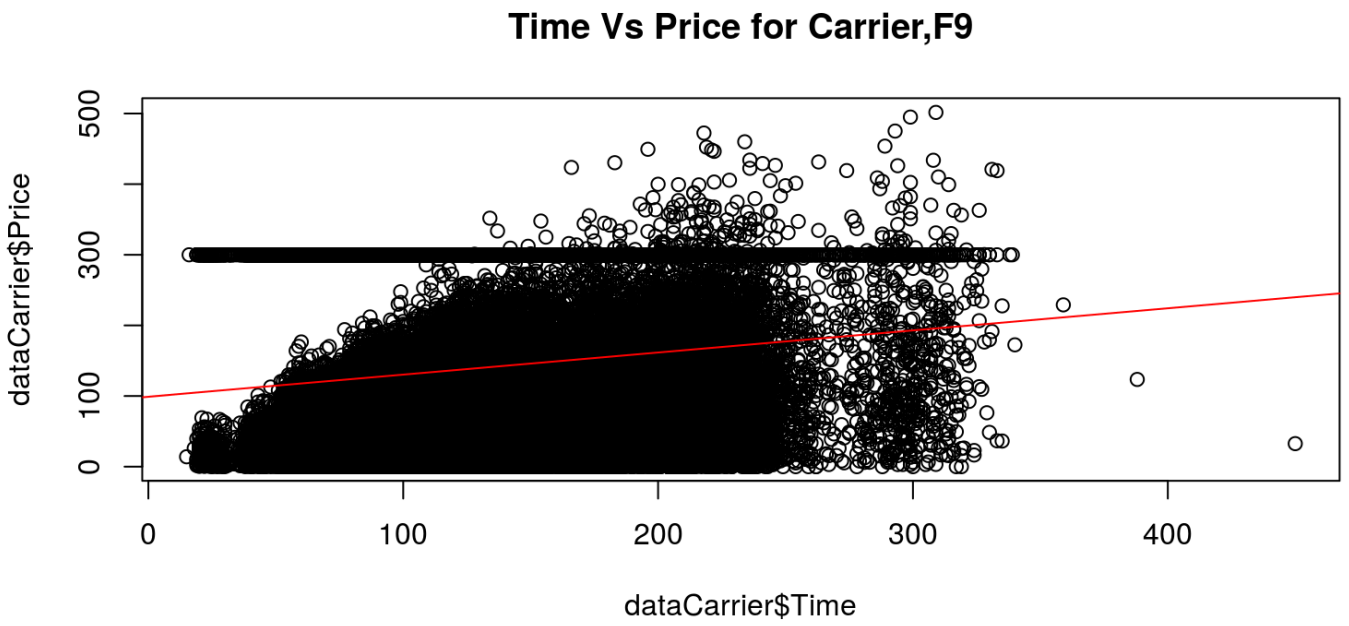


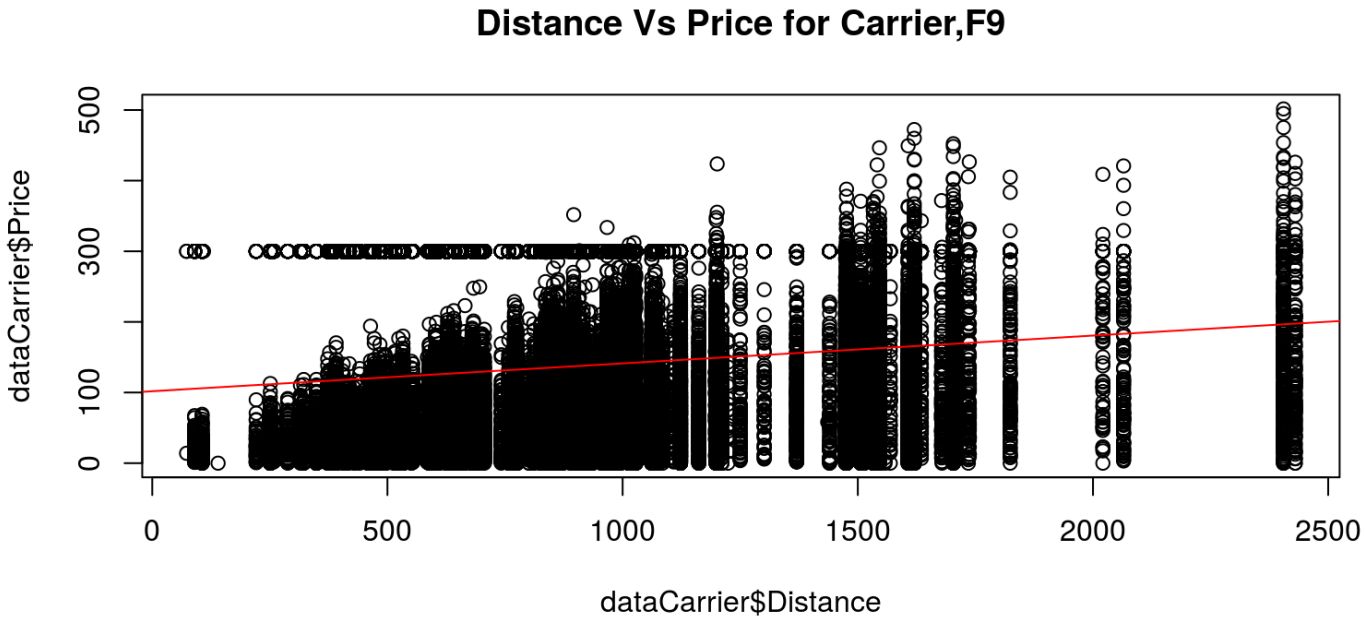
Time Vs Price for Carrier,EV

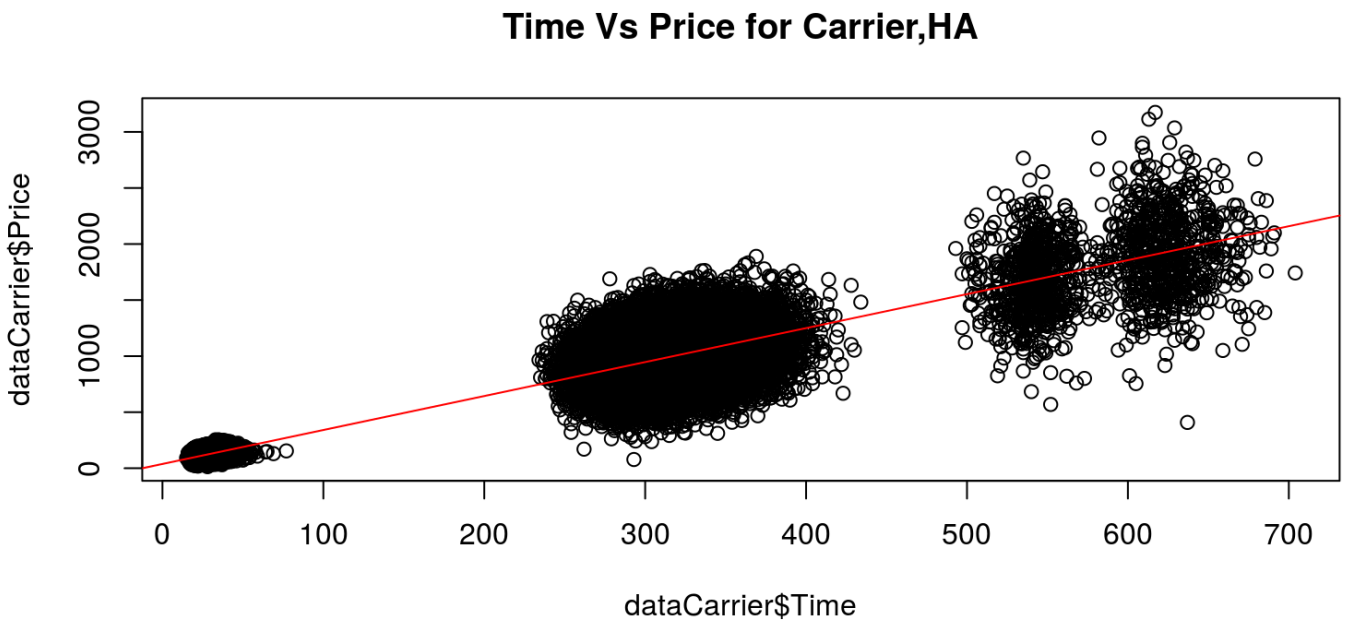


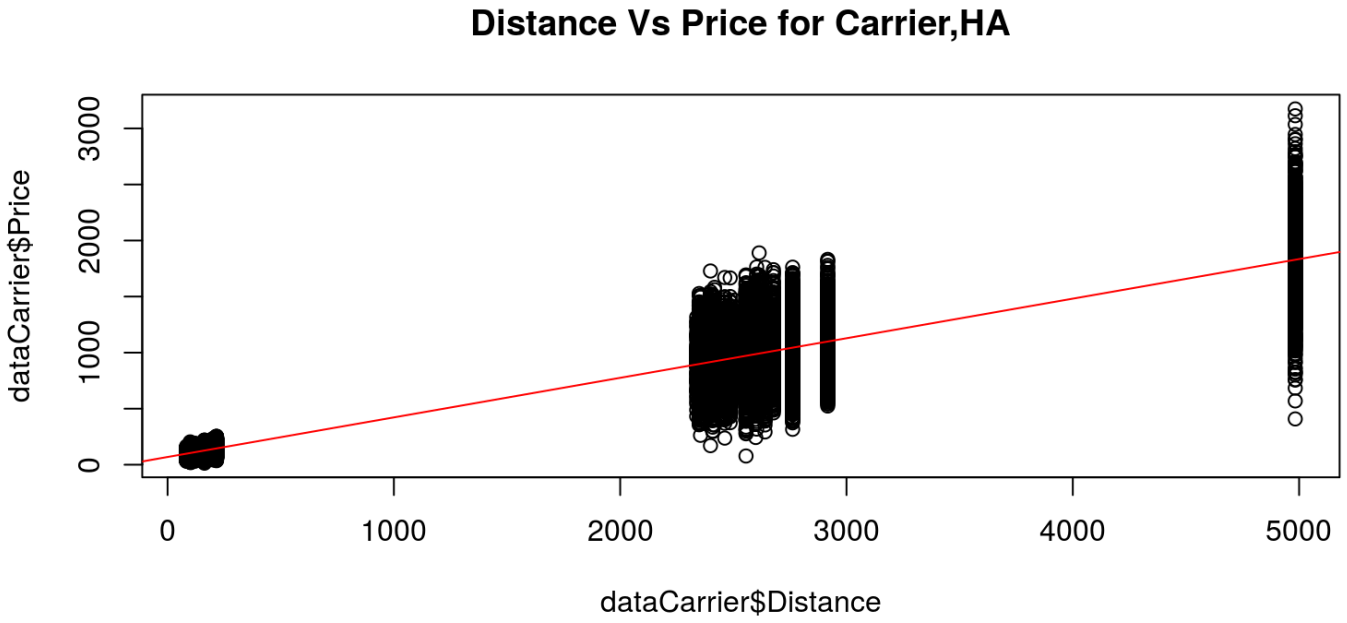
Distance Vs Price for Carrier,EV



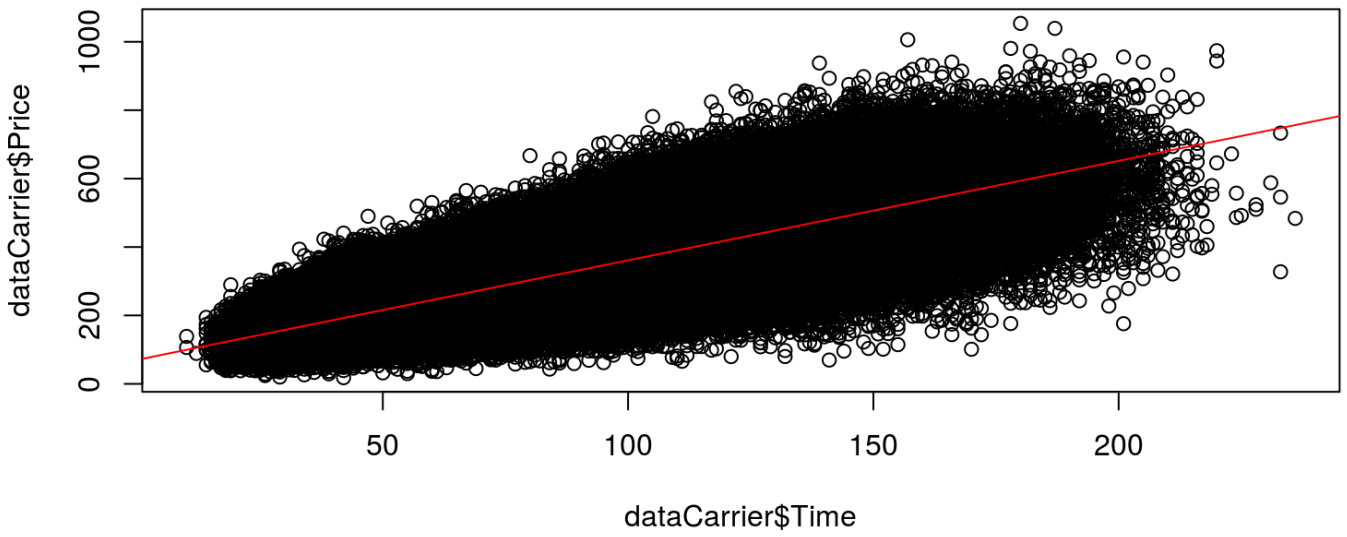

Time Vs Price for Carrier,F9
dataCarrier$Price
0
100
300
500
0
100
200
300
400
dataCarrier$Time


Distance Vs Price for Carrier,F9
dataCarrier$Price
0
100
300
500
0
500
1000
1500
2000
2500
dataCarrier$Distance

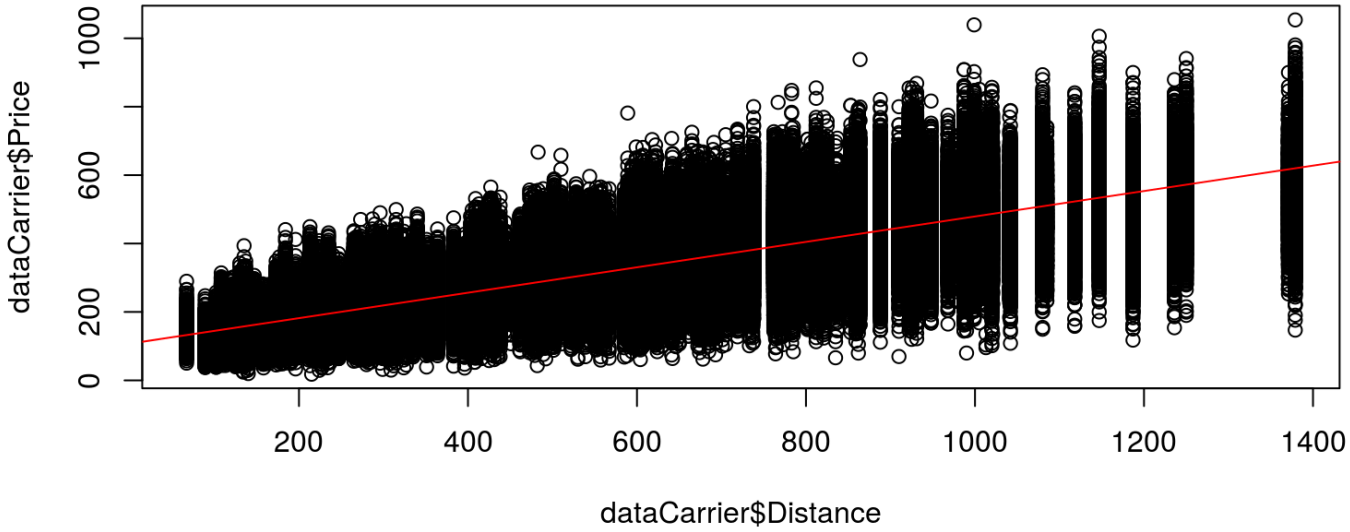

Time Vs Price for Carrier,HA
dataCarrier$Price
0
1000
2000
3000
0
100
200
300
400
500
600
700
dataCarrier$Time


Distance Vs Price for Carrier,HA
dataCarrier$Price
0
1000
2000
3000
0
1000
2000
3000
4000
5000
dataCarrier$Distance

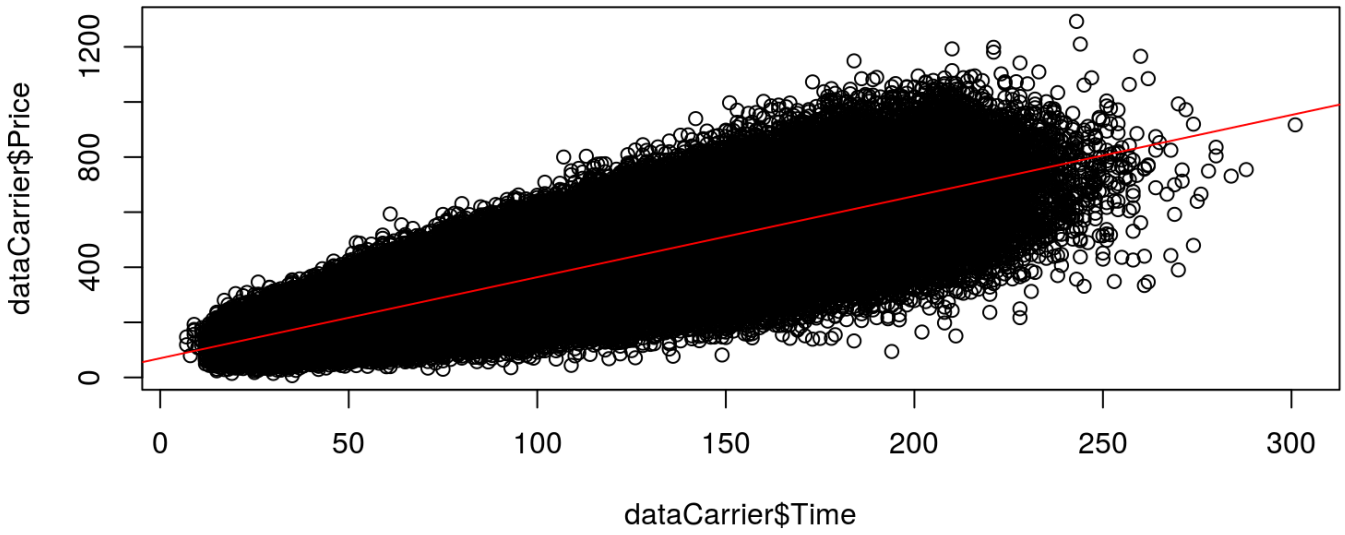
Time Vs Price for Carrier,MQ



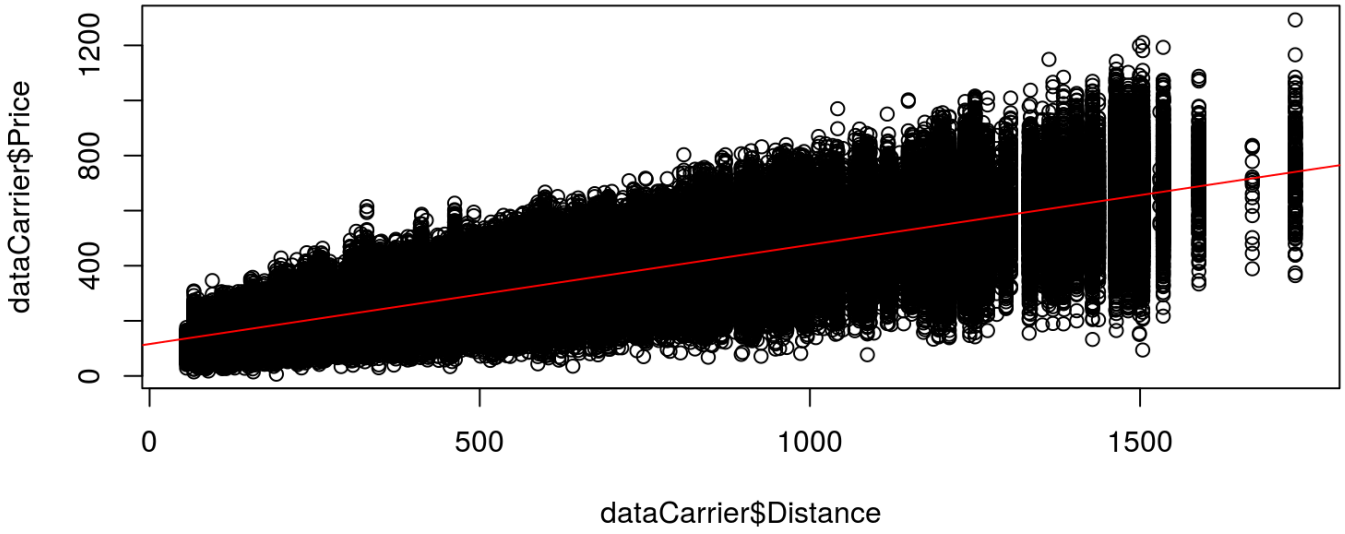
Distance Vs Price for Carrier,MQ



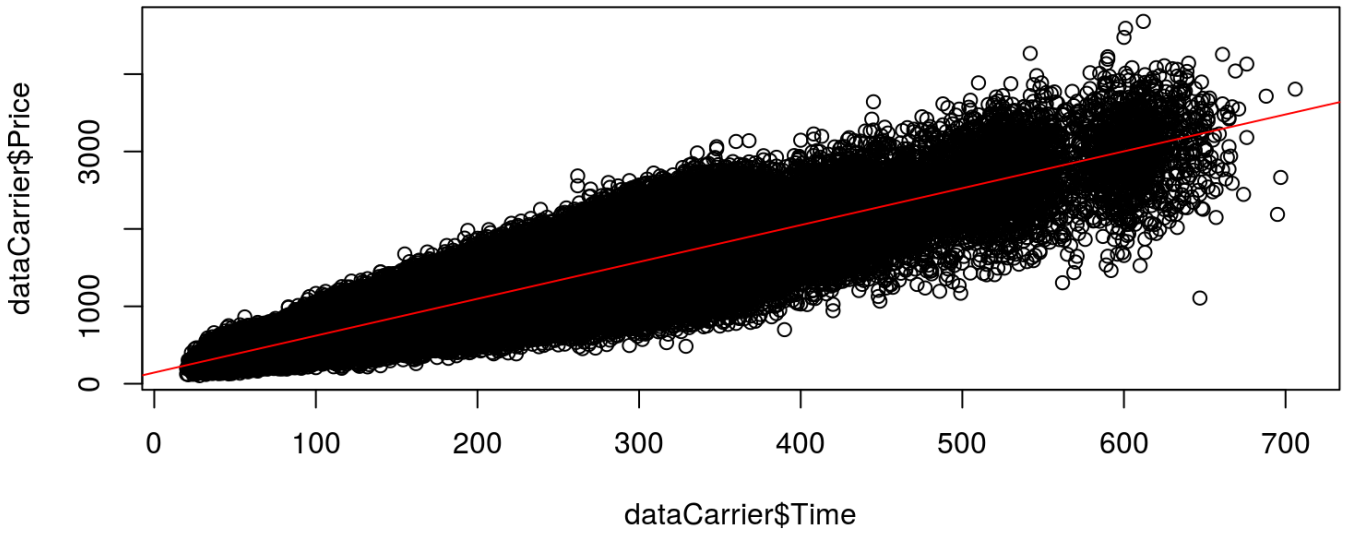
Time Vs Price for Carrier,OO



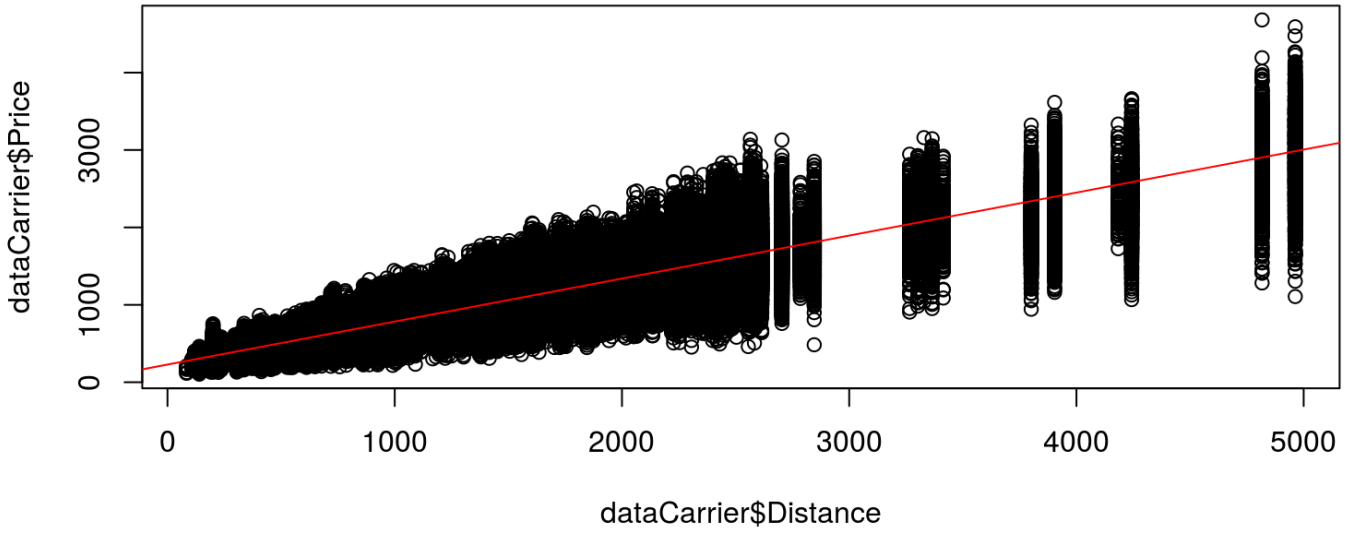
Distance Vs Price for Carrier,OO



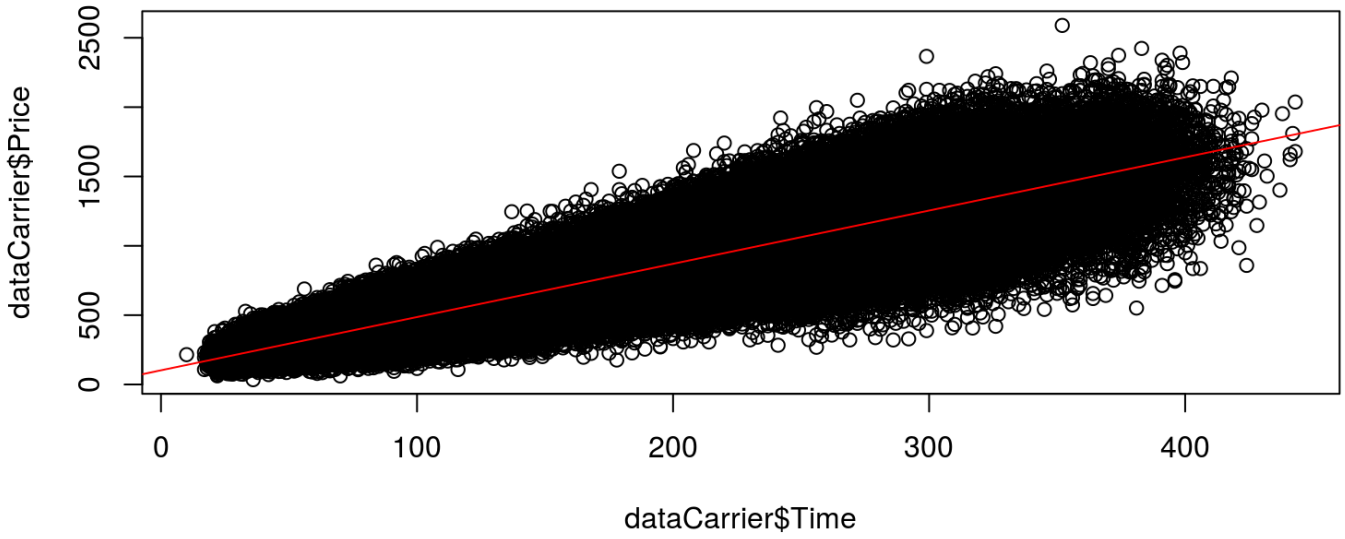
Time Vs Price for Carrier,UA



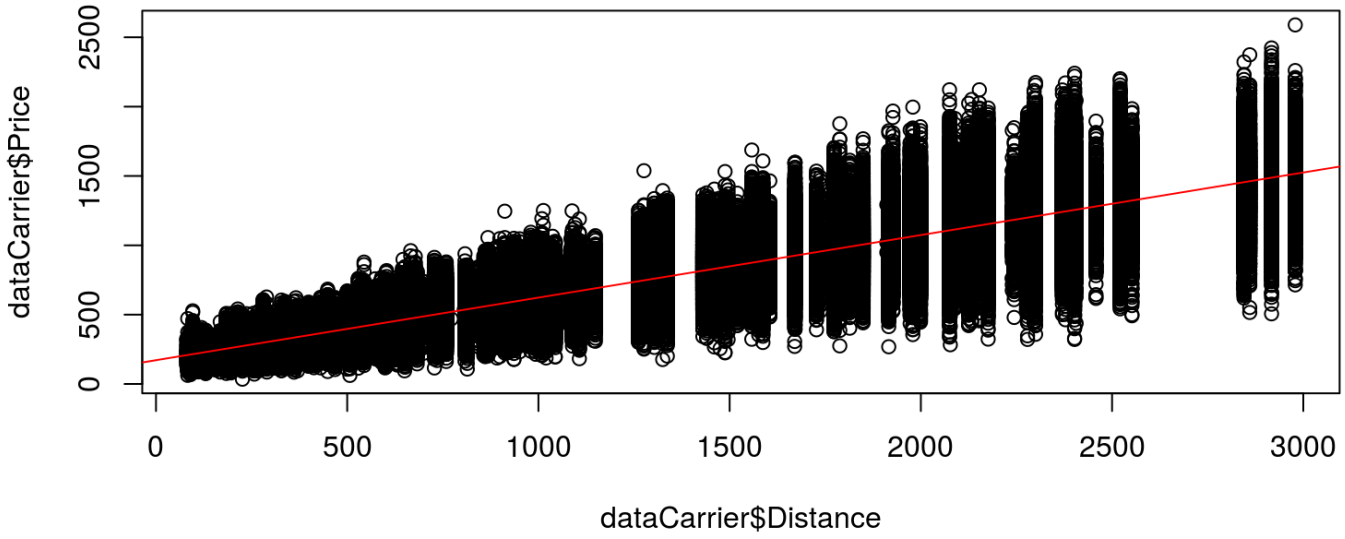
Distance Vs Price for Carrier,UA

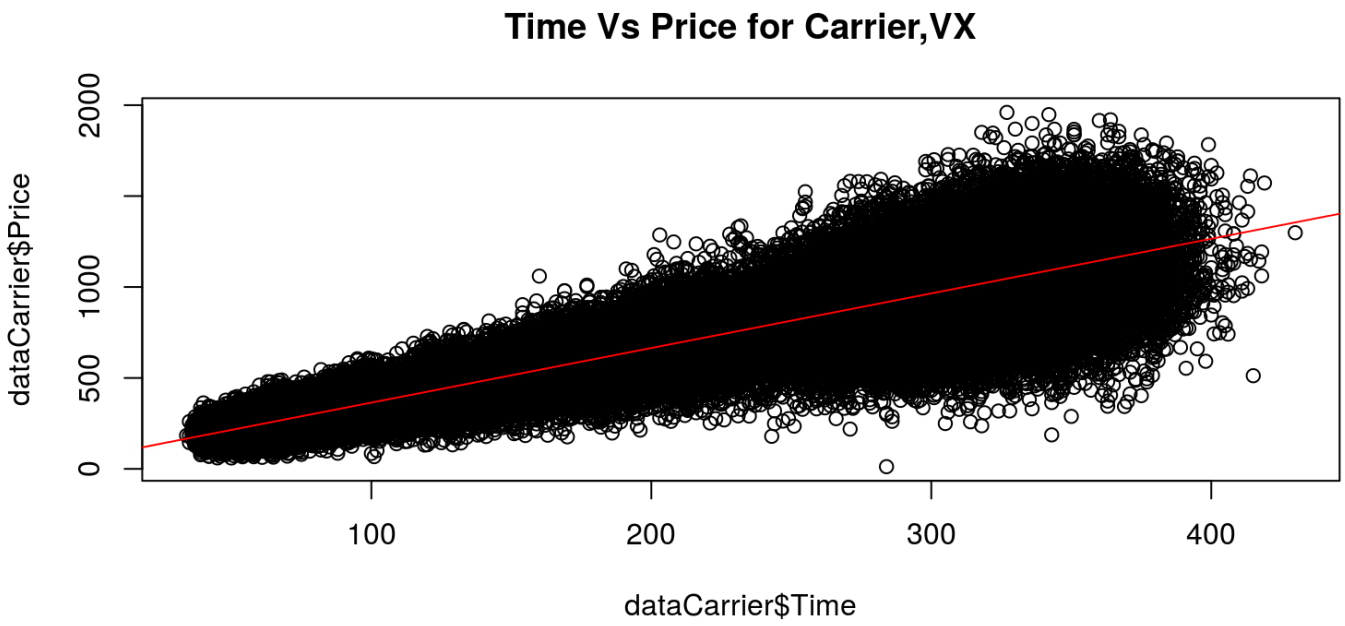


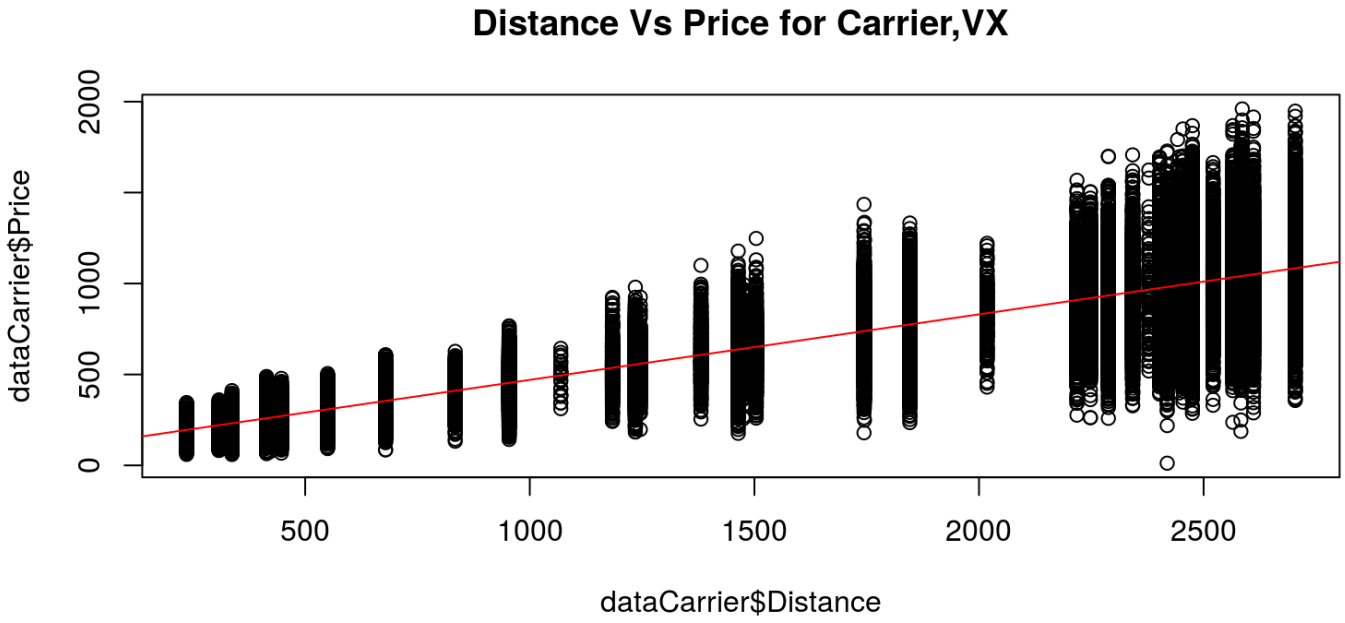
Time Vs Price for Carrier,US

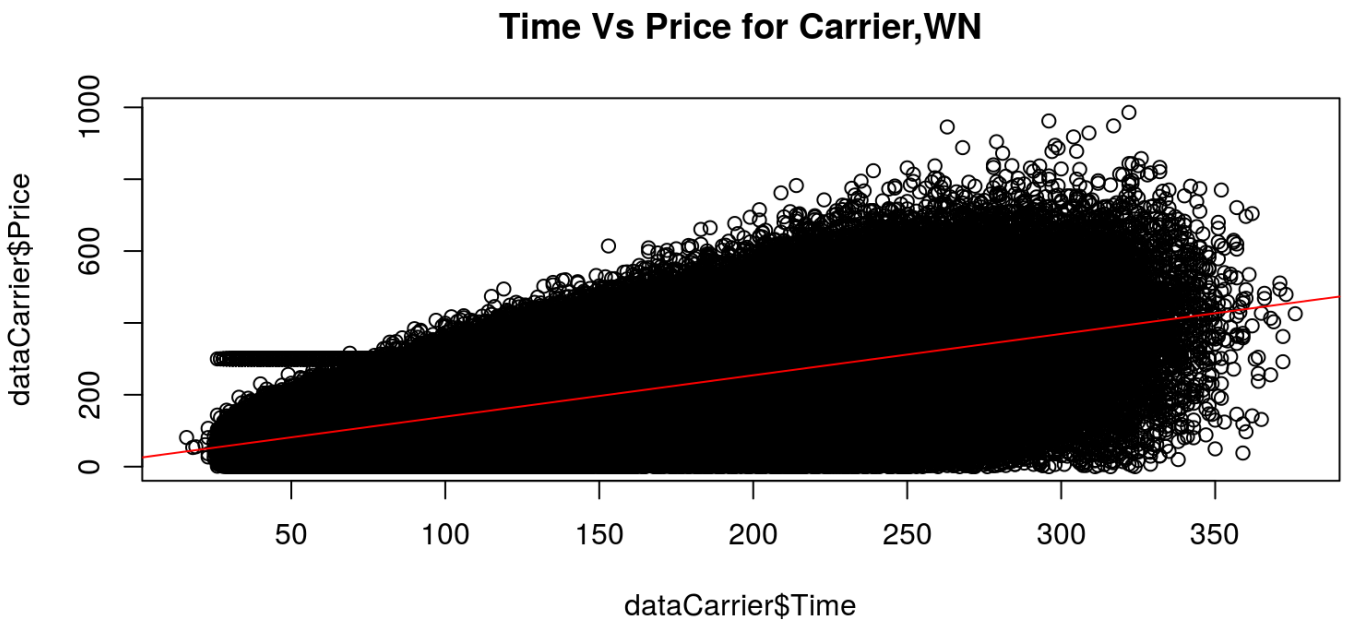


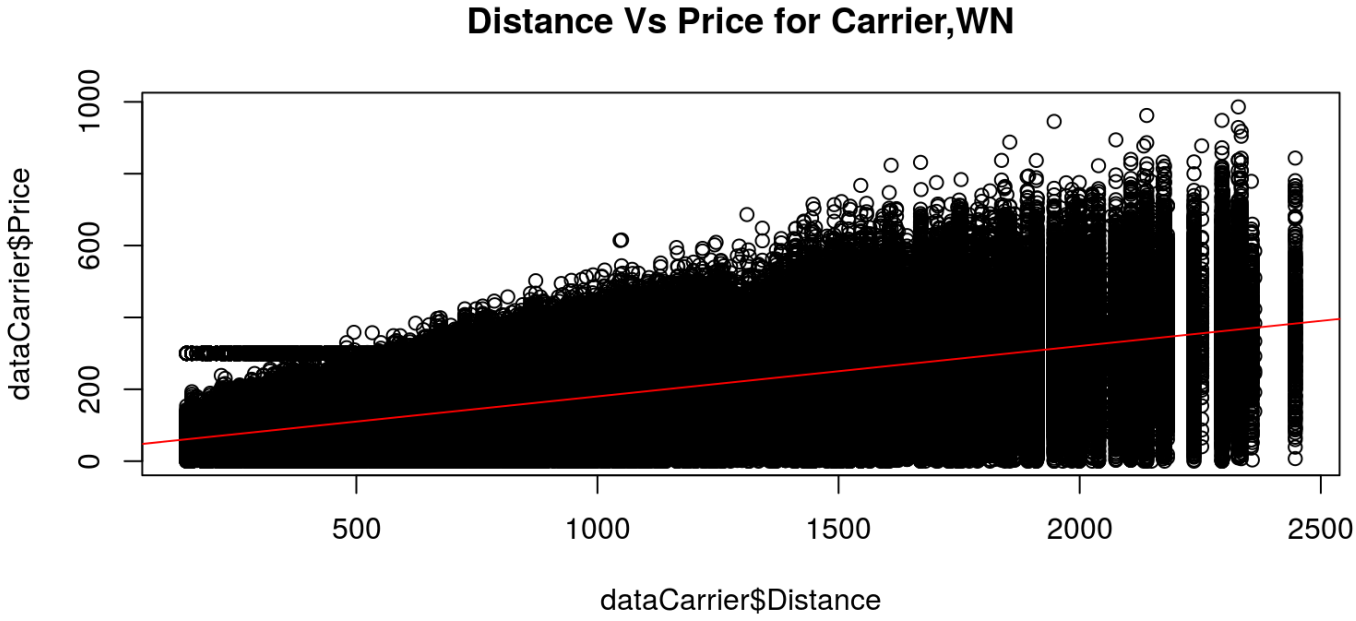
Distance Vs Price for Carrier,US




Time Vs Price for Carrier,VX
dataCarrier$Price
0
500
1000
2000
100
200
300
400
dataCarrier$Time


Distance Vs Price for Carrier,VX
dataCarrier$Price
0
500
1000
2000
500
1000
1500
2000
2500
dataCarrier$Distance


Time Vs Price for Carrier,WN
dataCarrier$Price
0
200
600
1000
50
100
150
200
250
300
350
dataCarrier$Time


Distance Vs Price for Carrier,WN
dataCarrier$Price
0
200
600
1000
500
1000
1500
2000
2500
dataCarrier$Distance

```
#search which is the cheapest airline
sorting <- c()
string <- ""
if(length(vectorT) > length(vectorD)){
  #time
  min <- 99999
  carrierName <- "none"
  for(i in unique(data$Carrier)){
    if(mapT[[i]] < min){
      min <- mapT[[i]]
      carrierName <- i
    }
    string <- paste(mapT[[i]],sep = ",",i)
    sorting <- c(sorting,string)
  }
}else{
  min <- 99999
  carrierName <- "none"
  for(i in unique(data$Carrier)){
    if(mapD[[i]] < min){
      min <- mapD[[i]]
      carrierName <- i
    }
    string <- paste(mapD[[i]],sep = ",",i)
    sorting <- c(sorting,string)
  }
}
sortedArray <- sort(sorting)
printArray <- paste(sortedArray,sep = ",",rank(sortedArray))
printArray
```

```
##  [1] "133.675422313118,F9,1"  "149.526289318186,AS,2"
##  [3] "151.667288523876,WN,3"  "373.971274561225,HA,4"
##  [5] "392.939282815709,MQ,5"  "396.226131863862,OO,6"
##  [7] "397.12937405042,VX,7"   "398.681588284215,EV,8"
##  [9] "401.532313917293,B6,9"  "405.427352808947,AA,10"
## [11] "528.159207055109,US,11" "539.016284707695,DL,12"
## [13] "672.852910696981,UA,13"
```

```
carrierName
```

```
## [1] "F9"
```

Linear Regression

We used the lm function to apply linear regression to predict price against time or distance.

The mean is computed for time and distance. Either one will be used to compute the predicted price.

The decision to use time or distance is made on the mean sqaure error. Smaller the error, better the accuracy for the linear regression

Analysis

Time OR Distance

For each airline we computed the mean sqaure error for both time and distance. If the mean sqaure error for time is smaller than the mean sqaure error for the same airline then add the mean sqaure error of time to the vector for time. Similarly with distance.

However we noticed that the vector for time had a greater size than distance hence we have chosen to predict prices against time instead of distance

Output:

14 Airlines active in 2015.

HA

EV

MQ

OO

US

B6

WN

UA

DL

NK

VX

AS

F9

AA

NK

From the recuder we computed values for all flights between 2010 to 2014 and displayed only those active in the year 2015 and found 13 of these airlines active.

HA

EV

MQ

OO

US

B6

WN

UA

DL

NK

VX

AS

F9

AA

From R when we ranked the airlines with the lowest possible prices as :

“133.675422313118,F9,1”

“149.526289318186,AS,2”

“151.667288523876,WN,3”

“373.971274561225,HA,4”

“392.939282815709,MQ,5”

“396.226131863862,OO,6”

“397.12937405042,VX,7”

“398.681588284215,EV,8”

“401.532313917293,B6,9”

“405.427352808947,AA,10”

“528.159207055109,US,11”

“539.016284707695,DL,12”

“672.852910696981,UA,13”

Least expensive airline : F9

Conclusion

We received 13(printed in the output) airlines from the reducer output which were active in 2015 and got values for the airlines from the year 2010 to 2014.

From R while doing linear regression, we drew the graphs for each airline against time and distance.

The mean sqaure errors was consitently better for time and hence we decided to pick time as the explanatory variable and tried to predict price based on time rather than distance.

Once we found the correct explanatory variable we calculated the predicted prices by picking a mean value in the data as the constanst value with which we can predict prices for all airlines and when we sorted

Once we found the mean, we multiplied the slope with the mean value and added the intercept. For the 13 airlines we plotted their predicted prices (showed in the output) we found F9 to have the least value.