Analyzing historical airlines data with Destination to Chicago O’Hare Airport

Min-Huey Wang and Tomoaki Tanaka

Motivation

Flight delays impact the US economics according to the revised final report released on December 16, 2010 from National Center of Excellence for Aviation Operations Research (NEXTOR). The report entitled “Total Delay Impact Study” is a comprehensive assessment of the costs and impacts of flight delay in the United States during 2007. It concluded in 2007, domestic flight delays were found to cost the U.S. economy $31.2 billion. Among these loss $16.7 billion in direct costs to passengers. The Line Item cost component is shown in table 1.

Table 1 Line Item Cost Component from UA flight delay in 2007 (NEXTOR)

Line.Item.Cost.Component	Category	Billions
Flight Delay Against Schedule	Airlines	4.6
Intrinsic Flight Delay due to Schedule Buffer	Airlines	3.7
Excess Travel Time due to Schedule Buffer	Passengers	6.0
Passenger Delay Against Schedule: Delayed Flights	Passengers	4.7
Passenger Delay Against Schedule: Canceled Flights	Passengers	3.2
Passenger Delay Against Schedule: Missed Connections	Passengers	1.5
Capacity-Induced Schedule Delay	Passengers	0.7
Voluntary Early-Departure-Time Adjustment	Passengers	0.6
Welfare loss due to switch from air to automobile	Shared	2.0
Externality cost from increased road traffic	Shared	0.2
Forgone GDP	Shared	4.0
Total U.S. Cost	All	31.2

This makes the study of the delay of flight valuable. It’s worthy to study the delay pattern to have better understanding of the causes such that we can help either the airline companies or passengers to minimize the impacts.

State of problem

We hope from the historical data to build a model to predict the delay of the flight. The biggest challenge is how to choose the right features among 31 features to establish the right model. Can we make suggestion about which airline company is more reliable? Could we improve the cause of delay? i.e. reduce the delay rate. If delay is inevitable, could we estimate the delay time in advance such that either airline company or the customer can prepare for it.

Background of Dataset

The data set is download from the website of United States Department of Transportation.The data contain 4,780,904 recorders of arriving flight to Chicago O'Hare airport. The time span is from 1987 to 2013 including 23 carriers from 180 cities. The abbreviation namse of 23 carriers are 9E, AA, AS, B6, CO, DH, DL, EA, EV, F9, HP, MQ, NW, OH, OO, PA (1), PI, TW, UA, US, VX, XE, YV. Every recorder has 31 features. The names of features are : Year, Month, DayofMonth, DayOfWeek, DepTime, CRSDepTime, ArrTime, CRSArrTime, UniqueCarrier, FlightNum, TailNum, ActualElapsedTime, CRSElapsedTime, AirTime, ArrDelay, DepDelay, Origin, Dest, Distance, TaxiIn, TaxiOut, Cancelled, CancellationCode, Diverted, CarrierDelay, WeatherDelay, NASDelay, SecurityDelay, LateAircraftDelay, IsArrDelayed, and IsDepDelayed. The total counts of flight arriving O'Hare airport per year is shown in figure 1. Year 1987 and 2013 are not whole year count. There is a step increase after year 2000. A bar chart of count per carrier company per year is shown in figure 2. From the plot it shows the big airline company decreasing their flights to Chicago while local or small airline company takes the loads. Combining information shown in figure 1 and 2 it indicates that the market of flying to O'Hare was opened to more operators after year 2000 and the capacity is increased.

Flight variables

- Variables included in the flight and weather data:

```
## 'data.frame':    4738790 obs. of  56 variables:
##  $ Year                  : int  1987 1987 1987 1987 1987 1987 1987 1987 1987 1987 ...
##  $ Month                 : int  10 10 10 10 10 10 10 10 10 10 ...
##  $ DayofMonth            : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ DayOfWeek             : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ DepTime               : int  634 1155 710 1327 1556 1329 759 908 1324 840 ...
##  $ CRSDepTime            : int  635 1157 710 1330 1556 1330 800 910 1325 840 ...
##  $ ArrTime               : int  1226 1307 717 1541 1811 1956 1305 1035 1412 1022 ...
##  $ CRSArrTime            : int  1231 1307 718 1559 1749 1925 1308 1035 1417 1017 ...
##  $ UniqueCarrier         : Factor w/ 23 levels "9E","AA","AS",..: 2 19 17 19 19 2 19 19 2 2
 ...
##  $ FlightNum             : int  508 772 715 82 643 390 524 393 591 234 ...
##  $ TailNum               : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ ActualElapsedTime     : int  232 72 67 134 195 267 186 147 108 102 ...
##  $ CRSElapsedTime        : int  236 70 68 149 173 235 188 145 112 97 ...
##  $ AirTime               : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ ArrDelay              : int  -5 0 -1 -18 22 31 -3 0 -5 5 ...
##  $ DepDelay              : int  -1 -2 0 -3 0 -1 -1 -2 -1 0 ...
##  $ Origin                : Factor w/ 180 levels "ABE","ABQ","ALB",..: 156 119 98 11 65 123 69
 23 107 21 ...
##  $ Dest.x                : Factor w/ 1 level "ORD": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Distance              : int  1846 334 323 972 1182 1836 1498 867 594 409 ...
##  $ TaxiIn                : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ TaxiOut               : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Cancelled             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CancellationCode      : logi  NA NA NA NA NA NA ...
##  $ Diverted              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ CarrierDelay          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ WeatherDelay          : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ NASDelay              : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ SecurityDelay         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ LateAircraftDelay     : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ IsArrDelayed          : Factor w/ 2 levels "NO","YES": 1 1 1 1 2 2 1 1 1 2 ...
##  $ IsDepDelayed          : Factor w/ 2 levels "NO","YES": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ICAO                  : Factor w/ 1 level "KORD": 1 1 1 1 1 1 1 1 1 1 ...
##  $ MaxTempF              : int  72 72 72 72 72 72 72 72 72 72 ...
##  $ MeanTempF             : int  56 56 56 56 56 56 56 56 56 56 ...
##  $ MinTempF              : int  39 39 39 39 39 39 39 39 39 39 ...
##  $ MaxDewPointF          : int  49 49 49 49 49 49 49 49 49 49 ...
##  $ MeanDewPointF         : int  40 40 40 40 40 40 40 40 40 40 ...
##  $ Min.DewpointF         : int  34 34 34 34 34 34 34 34 34 34 ...
##  $ MaxHumidity           : int  93 93 93 93 93 93 93 93 93 93 ...
##  $ MeanHumidity          : int  58 58 58 58 58 58 58 58 58 58 ...
##  $ MinHumidity           : int  30 30 30 30 30 30 30 30 30 30 ...
##  $ MaxSeaLevelPressureIn : num  30.1 30.1 30.1 30.1 30.1 ...
##  $ MeanSeaLevelPressureIn: num  29.8 29.8 29.8 29.8 29.8 ...
##  $ MinSeaLevelPressureIn : num  29.5 29.5 29.5 29.5 29.5 ...
##  $ MaxVisibilityMiles    : int  15 15 15 15 15 15 15 15 15 15 ...
##  $ MeanVisibilityMiles   : int  14 14 14 14 14 14 14 14 14 14 ...
##  $ MinVisibilityMiles    : int  10 10 10 10 10 10 10 10 10 10 ...
##  $ MaxWindSpeedMPH       : int  29 29 29 29 29 29 29 29 29 29 ...
##  $ MeanWindSpeedMPH      : int  9 9 9 9 9 9 9 9 9 9 ...
##  $ MaxGustSpeedMPH       : int  38 38 38 38 38 38 38 38 38 38 ...
##  $ PrecipitationIn       : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ CloudCover            : int  4 4 4 4 4 4 4 4 4 4 ...
##  $ Events                : Factor w/ 19 levels "","Fog","Fog-Rain",..: 12 12 12 12 12 12 12 1
2 12 12 ...
##  $ WindDirDegrees        : int  223 223 223 223 223 223 223 223 223 223 ...
##  $ Dest.y                : Factor w/ 1 level "ORD": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ts                    : POSIXct, format: "1987-10-01 06:35:00" "1987-10-01 11:57:00" ...
```

- Total number of arrival flights excluding cancelled flights:

```
## [1] 4738178
```

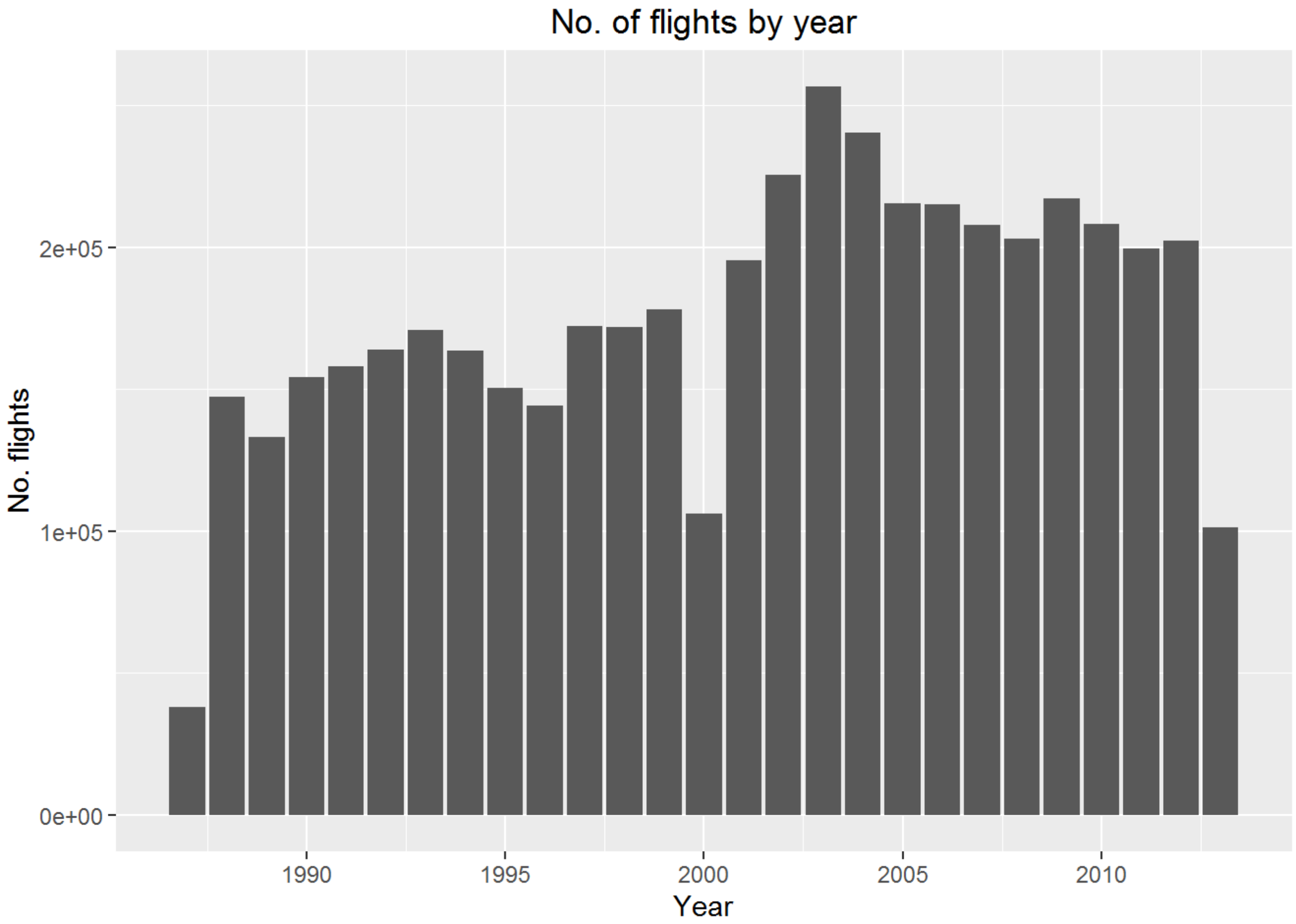

Figure 1. Historical total counts of flight arriving Chicago O’Hare airport.

```
ggplot(planes_year_arr_c, aes (Year, count,fill=factor(UniqueCarrier))  ) +
  geom_bar(stat="identity",position = "dodge")+               # width of bar
  xlab("Year")  +
  ylab("No. flights") +
  ggtitle("No. of flights by year")
```

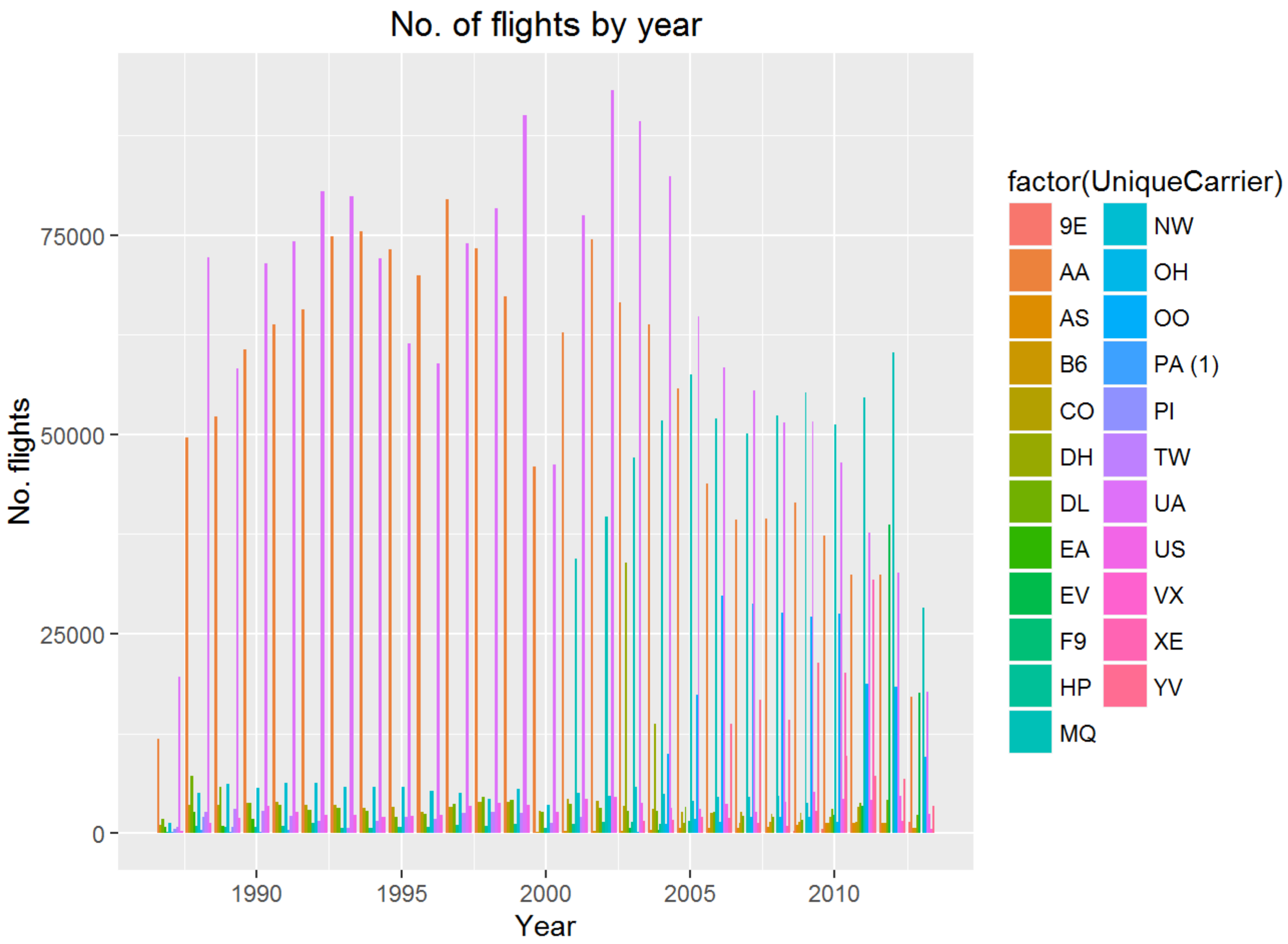

Figure 2 Bar chart of count of flight arriving O'Hare airport per carrier company per year.

To fair compare the delay events of every carrier we define the delay rate as delay time divided by traveling distance in unit of (min/mile). Which take into consideration of long travel distance is likely to have longer delay time. The historical delay rate of every carrier is shown in figure 3. We select five big airline companies AA, UA, DL, CO, and NW in U.S.A. for a close look to compare the delay rate. The delay rate of the five company is shown in figure 4. During the time span Delta and Northwest's operating certificates were merged on December 31, 2009. Northwest then ceased to exist as an independent carrier. That is why no more data for NW after 2009. The similar merger happened to UA and CO at later time on November 30, 2011. Among the five airlines NW has the worst delay rate and CO the second worst. After the merger the delay rate of DL jumped because the bad delay rate from NW. As for UA the delay rate seems not affect by CO. UA and AA have delay rate comparable. The delay rate has slow increase tendency.

```
planes_year_arr <- merged_data%>%
  group_by(UniqueCarrier,Year)%>%
  filter( ArrDelay>0, ArrDelay!=1)%>%
  summarise(count = n(), dist=mean(Distance, na.rm=TRUE),
            delay=mean(ArrDelay, na.rm=TRUE))

ggplot(planes_year_arr,
       aes(x=Year,y=delay/dist,color=UniqueCarrier))+geom_line()+geom_point()+ylab("delay/dist
 (min/mile)")
```

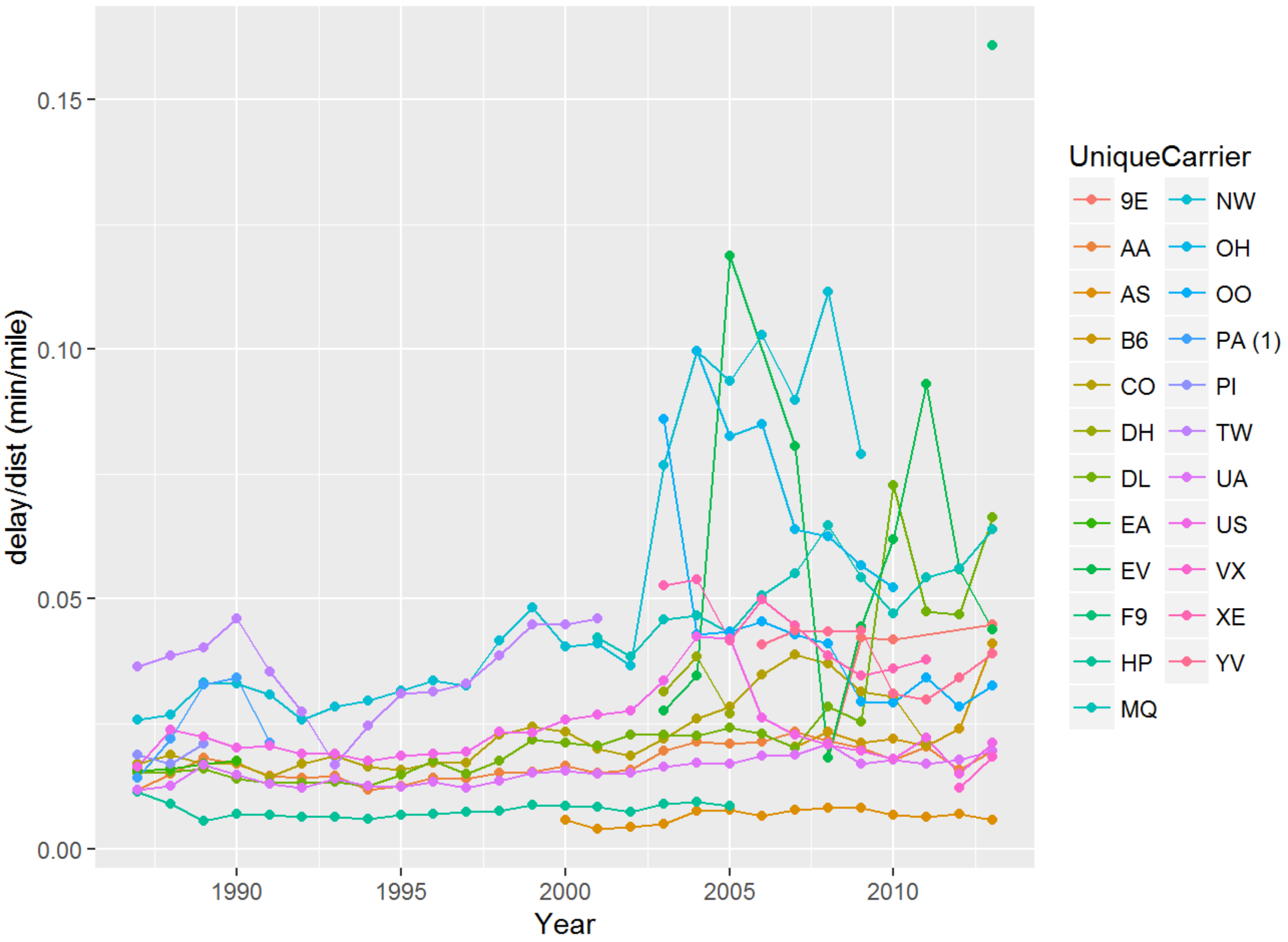

Figure 3. The historical delay rate of every carrier arriving O'Hare airport. The delay rate is defined as delay time divided by traveling distance in unit of (min/mile).

```
ggplot(subset(planes_year_arr,UniqueCarrier %in% c("AA","UA","DL","NW","CO")),
          aes(x=Year,y=delay/dist,color=UniqueCarrier))+geom_line()+geom_point()+ylab("delay/dis
t (min/mile)")
```

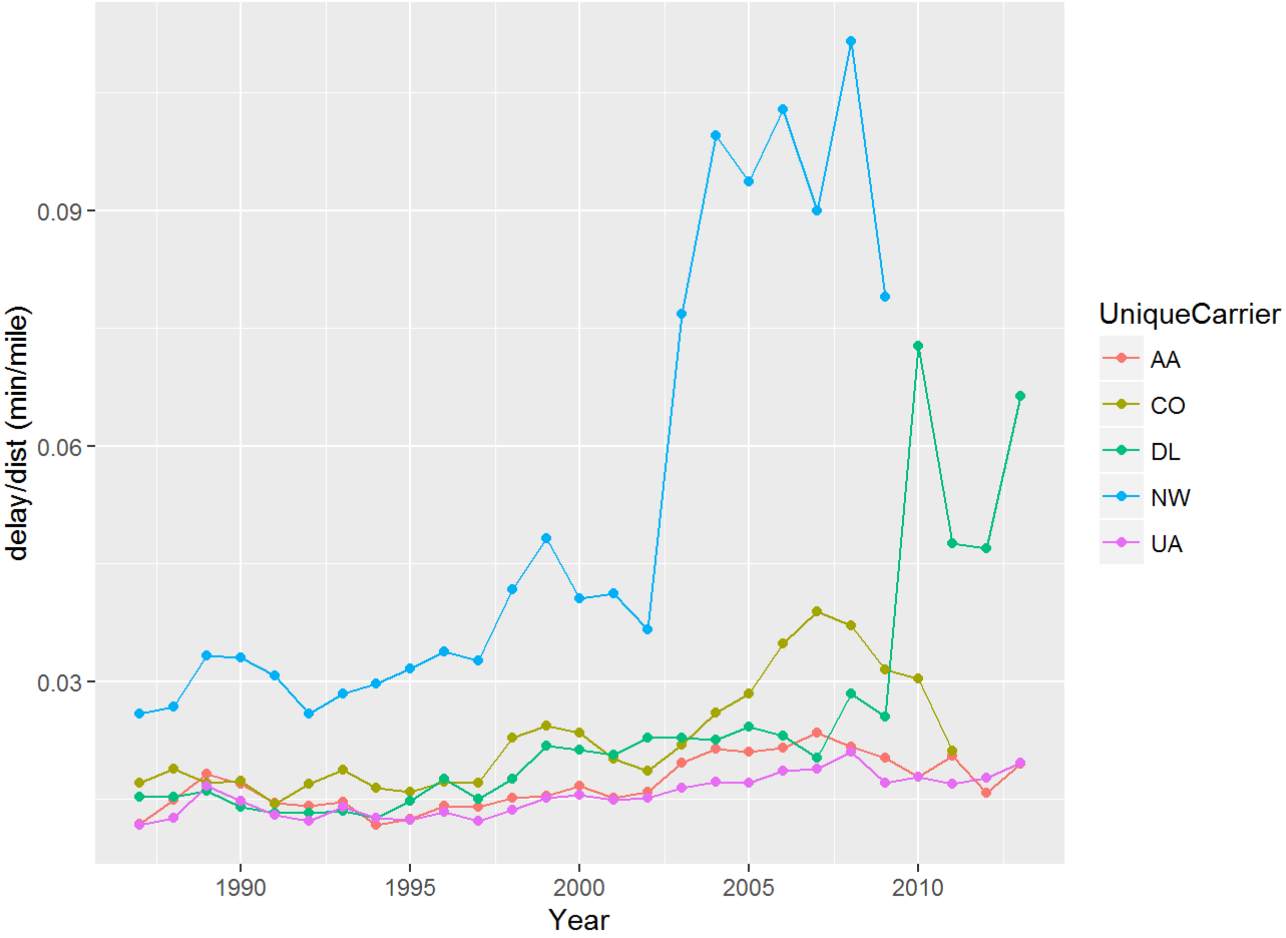

Figure 4. The historical delay rate of five USA big airline companies.

Weather variables

Weather data include the climate condition variables like temperature, pressure, humidity, precipitation, visibility, cloudcover etc. One weather condition per daya is stored in file. We merged the flight data and weather data to

- Number of delay by weather events:

```
## # A tibble: 18 x 2
##                       Events      n
##                       <fctr>  <int>
## 1                        Fog  63734
## 2                   Fog-Rain  76046
## 3  Fog-Rain-Hail-Thunderstorm    650
## 4              Fog-Rain-Snow  19035
## 5  Fog-Rain-Snow-Thunderstorm   1647
## 6       Fog-Rain-Thunderstorm  34311
## 7                   Fog-Snow  39963
## 8       Fog-Snow-Thunderstorm    288
## 9            Fog-Thunderstorm    420
## 10          Hail-Thunderstorm    104
## 11                      Rain 178403
## 12     Rain-Hail-Thunderstorm   1467
## 13                 Rain-Snow  21170
## 14     Rain-Snow-Thunderstorm   1809
## 15          Rain-Thunderstorm  84546
## 16                      Snow 105774
## 17          Snow-Thunderstorm    341
## 18               Thunderstorm   2100
```

```
ggplot(delay_w_events, aes(x=Events, fill=Events)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 90, hjust = -0.,vjust=0.2))
```

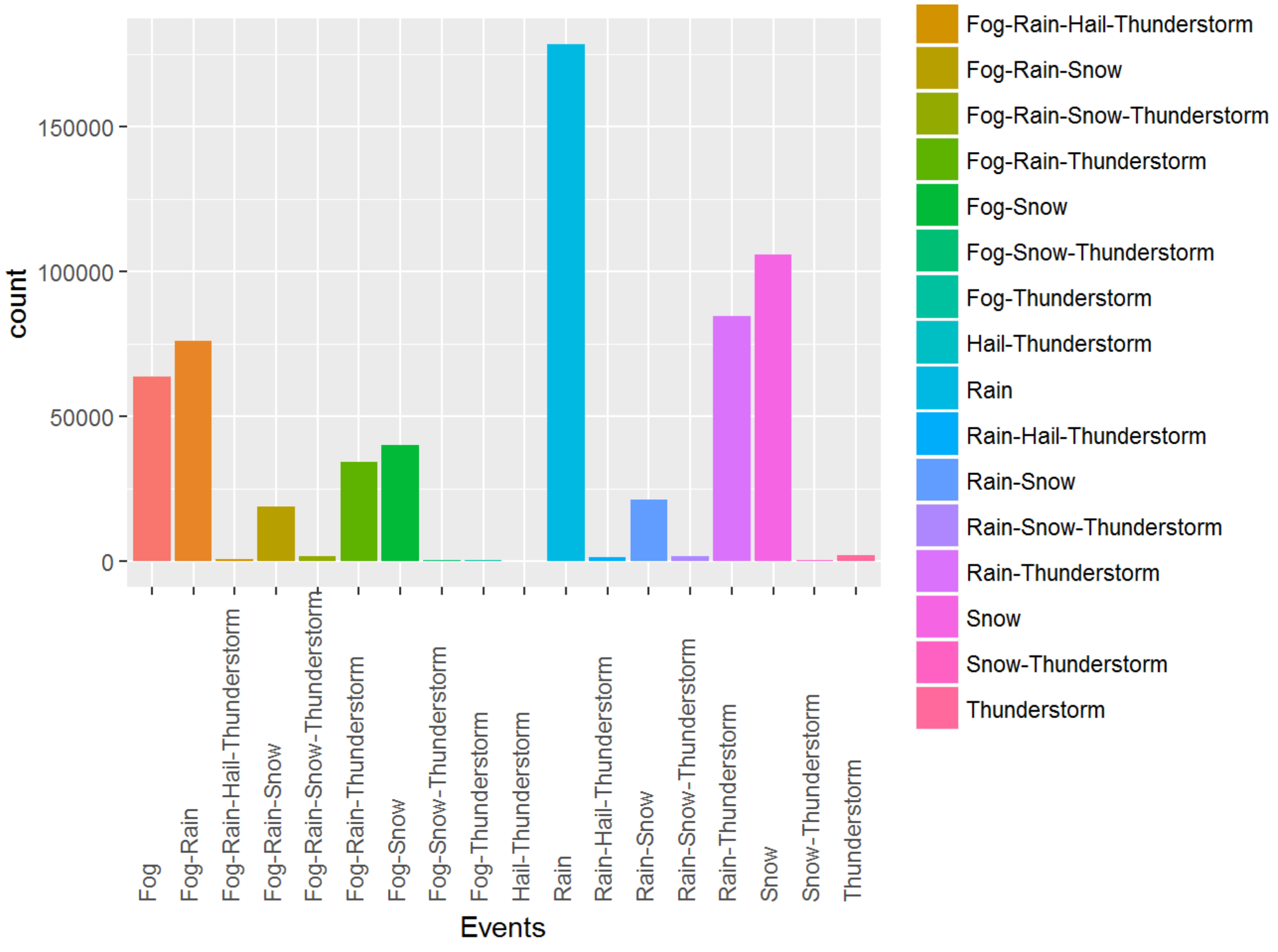

Figure 5 Plot for No. of delayed flight by weather event

Results of Logistic Regression for arrival delay

Logistic regression will provide probabilities in the form of P(Y=1|X). Our decision boundary will be 0.5. If P(y=1|X) > 0.5 then y = 1 otherwise y=0.

Logistic regression model for the arrival delay

Here we show the results of the logistic regression for arrival delays at O'Hare airport. We use four variables from the flight data: CRSElapsedTime, AirTime, Distance and ts (timestamp) and three variables from the weather data: Events, PrecipitationIn and CloudCover. One percent of original data is used for training for our logistic regression due to the memory limits.

```
arr.glm = glm(IsArrDelayed~CRSElapsedTime
                + AirTime
                +Distance
                +Events
                +PrecipitationIn +CloudCover
                ,data=train, family = binomial)
summary(arr.glm)
```

```
## 
## Call:
## glm(formula = IsArrDelayed ~ CRSElapsedTime + AirTime + Distance + 
##     Events + PrecipitationIn + CloudCover, family = binomial, 
##     data = train)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.3661  -0.7921  -0.5128   0.8220   3.0064  
## 
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      0.1285531  0.1656901   0.776  0.43783
## CRSElapsedTime                  -0.1014073  0.0030410 -33.346  < 2e-16
## AirTime                          0.1370203  0.0032963  41.568  < 2e-16
## Distance                        -0.0036737  0.0002463 -14.918  < 2e-16
## EventsFog-Rain                   0.0445470  0.1600050   0.278  0.78070
## EventsFog-Rain-Hail-Thunderstorm  0.1961907  0.8681515   0.226  0.82121
## EventsFog-Rain-Snow              0.3789516  0.2238820   1.693  0.09052
## EventsFog-Rain-Snow-Thunderstorm -0.2669552  0.7363537  -0.363  0.71695
## EventsFog-Rain-Thunderstorm      0.2471022  0.2083899   1.186  0.23571
## EventsFog-Snow                   0.3049030  0.1689946   1.804  0.07120
## EventsFog-Snow-Thunderstorm      2.5593729  1.3363019   1.915  0.05546
## EventsFog-Thunderstorm           3.0452000  1.4065103   2.165  0.03038
## EventsHail-Thunderstorm         -0.4426779  0.8311583  -0.533  0.59431
## EventsRain                      -0.0701753  0.1207552  -0.581  0.56115
## EventsRain-Hail-Thunderstorm     0.5293646  0.4778333   1.108  0.26793
## EventsRain-Snow                  0.2521763  0.1597848   1.578  0.11451
## EventsRain-Snow-Thunderstorm     0.5722242  0.4604357   1.243  0.21395
## EventsRain-Thunderstorm          0.0928735  0.1278660   0.726  0.46763
## EventsSnow                       0.2005404  0.1284867   1.561  0.11857
## EventsSnow-Thunderstorm         -0.1113030  0.8543138  -0.130  0.89634
## EventsThunderstorm               0.5328716  0.3261708   1.634  0.10232
## PrecipitationIn                  0.1636177  0.0763341   2.143  0.03208
## CloudCover                       0.0512873  0.0163026   3.146  0.00166
##                                     
## (Intercept)                         
## CRSElapsedTime                   ***
## AirTime                          ***
## Distance                         ***
## EventsFog-Rain                      
## EventsFog-Rain-Hail-Thunderstorm    
## EventsFog-Rain-Snow              .  
## EventsFog-Rain-Snow-Thunderstorm    
## EventsFog-Rain-Thunderstorm         
## EventsFog-Snow                   .  
## EventsFog-Snow-Thunderstorm      .  
## EventsFog-Thunderstorm           *  
## EventsHail-Thunderstorm             
## EventsRain                          
## EventsRain-Hail-Thunderstorm        
## EventsRain-Snow                     
## EventsRain-Snow-Thunderstorm        
## EventsRain-Thunderstorm
## EventsSnow
## EventsSnow-Thunderstorm
## EventsThunderstorm
## PrecipitationIn                  *
## CloudCover                       **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 14196  on 11198  degrees of freedom
## Residual deviance: 11249  on 11176  degrees of freedom
## AIC: 11295
##
## Number of Fisher Scoring iterations: 5
```

Prediction using testset

We predict arrival delay occured or not from our logistic regression. Test set include 240,720 rows for estimating the performance of our model.

```
fitted.results <- predict(arr.glm,newdata=test_arr,type='response')
```

Confusion matrix

```
table(test_arr$IsArrDelayed, fitted.results > 0.5)
```

```
##
##       FALSE  TRUE
##   NO  86234  5106
##   YES 17382 11262
```

We can estimate the accuracy of our model from the confusion matrix. You can calculate the accuracy of your model with:

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + TrueNegative + FalsePositive + FalseNegative}$$

The confusion matrix shows our logistic regression model using flight and weather data at O’Hare airport shows 81 % accuracy on the test set.

ROC curve

Receiver Operating Characteristic (ROC) curve summarizes the model’s performance by evaluating the trade offs between true positive rate (sensitivity) and false positive rate (1- specificity). In the ROC curve, the true positive rate is plotted in function of the false positive rate as shown below.

```
ROCRpred <- prediction(fitted.results, test_arr$IsArrDelayed)
ROCRperf <- performance(ROCRpred, 'tpr','fpr')
plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))
```

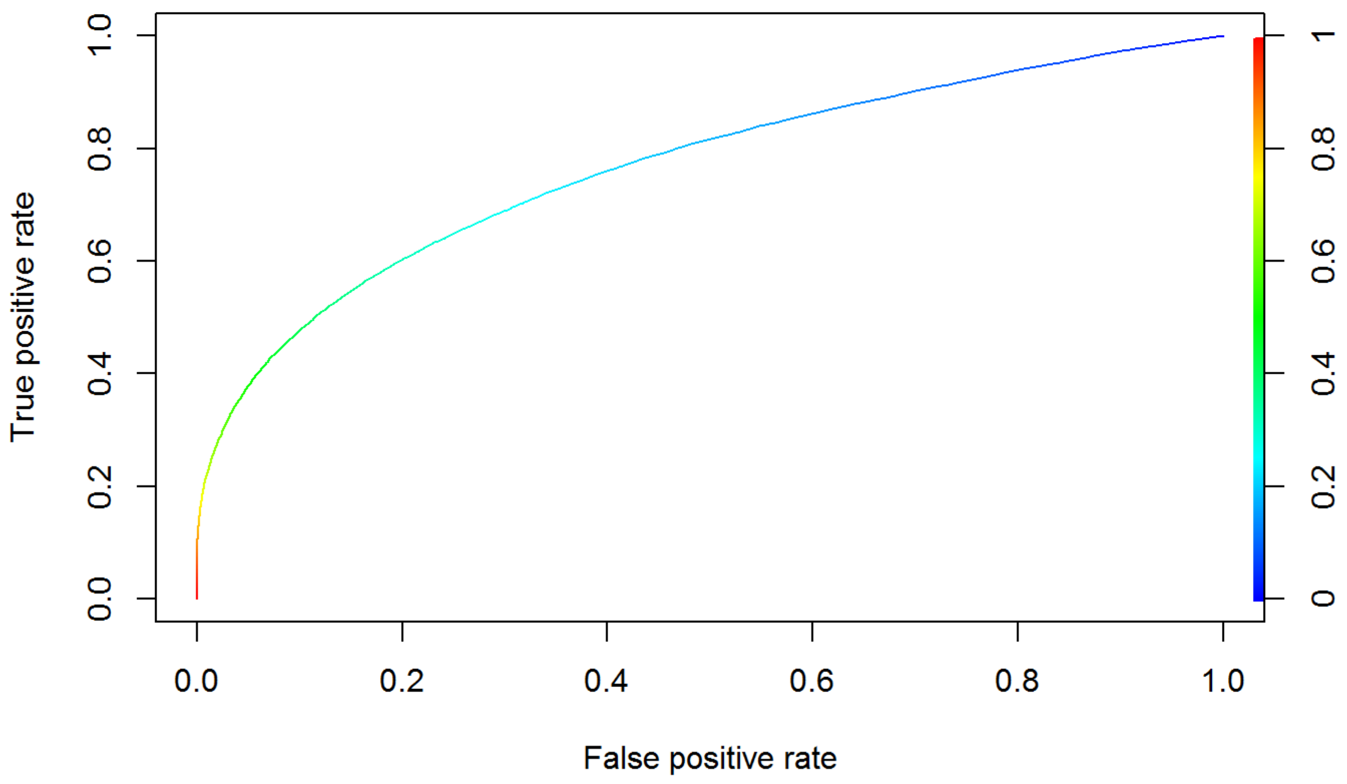

Figure 6 ROC curve for the arrival delay

Discussion and Conclusion

Our logistic regression model using the features: CRSElapsedTime, AirTime, Distance, ts (timestamp), weather Events, weather PrecipitationIn and weather CloudCover can predicts the delay occurrence up to 80% accuracy when verified with the test set. However, we only use 1 % of data for training due to hardware memory limits. It would get better accuracy if we could use more data for training of our model. There are some things we are interested to do but lack of data or time like: flight delay versus the age of the aircraft. It could provide suggestion to the airline company to decide when it’s more economic to replace the old aircraft. We notice that delay time can be negative that means flight can arrive earlier. It will be interested to know if the flight delay or flight arriving earlier related to the direction of the flight bound. From the course and exercise of this project we find data analysis can not only check the common acknowledge but also be a powerful tool to explore the unknown which beyond one’s expectation and imagination.