

UNIVERSITY OF CALIFORNIA DAVIS  
DEPARTMENT OF STATISTICS

---

## Practice in Statistical Data Science (STA-160)

---

### Analysis on the Effect of Atlantic Hurricanes on the US Airport Network

David Fung  
Yu Chan  
Jiahui Tan

## Contents

<b>1 Abstract</b>	<b>2</b>
<b>2 Introduction</b>	<b>2</b>
<b>3 Description of Data</b>	<b>3</b>
3.1 Data Preprocessing . . . . .	3
3.2 Data Exploration . . . . .	3
3.2.1 Exploring the NOAA Storm Data . . . . .	4
3.2.2 Exploring the Relationship Between Ontime Performance and Storm	5
<b>4 Model</b>	<b>10</b>
4.1 Motivation . . . . .	10
4.2 Findings . . . . .	10
4.3 Model Limitation . . . . .	11
<b>5 Conclusion</b>	<b>11</b>
<b>6 Bibliography</b>	<b>12</b>
<b>7 Project Reflections</b>	<b>12</b>
<b>8 Code Appendix</b>	<b>12</b>

## 1 Abstract

The airport system is characterized by a network structure connecting different operation centers. In this report, we study the performance and resilience of US airports reacting to the extreme weather condition of hurricanes, by measuring how delay and cancellation rate propagates throughout time when different Atlantic hurricanes hits the US coastals. We focused our study on hurricane Katrina, Sandy, and Ike, category three to five hurricanes that has directly hit US in the past resulting in costly damages. Our result suggests that when one of the major hubs is paralyzed, the effect can propagate, magnify and eventually involve a significant part of the network. Also, hurricanes striking along the eastern borders of the US will result in more cancellations in the airport network than those hitting the southern central areas, this is because more major hubs are concentrated in the East Coast. We also quantify the level of network congestion and cancellation rate and attempted at a prediction model of cancellations using the flight performance data. The random forest model results suggest that wind speed, followed by nature and distance were most indicative of flight cancellation.

## 2 Introduction

Atlantic hurricanes are tropical cyclones that form in the Atlantic Ocean between the months of June to November every year. These natural weather formations come in varying intensities and can lead to billion of dollars worth of damage to buildings and properties. The main goals of our project involve examining the effect of hurricanes falling into (3,4,5) categories of the Saffir-Simpson Hurricane Wind Scale on the US airport network through changes in flight cancellations and delays corresponding to changes in the weather and geographical distance away from the center of the hurricane. The hurricanes whose behavior and route we studied in depth were Sandy (10/2012, Class: 3), Ike (09/2008, Class 4), and Katrina (08/2005, Class: 5).

The stakeholders in our project include airport officials and airlines. For example, an airport might want to foresee the effect of an impending hurricane and how many flights may be canceled. An airline may also want to predict the probability of cancelling its flights based on historical data. This way, they can prepare and react to the situation, sending planes to the right locations to have the least impact on overall service. Consumers indirectly benefit from accurate prediction of flight cancellations since they can make changes to their travel plans earlier, diminishing the impact of such cancellation.

## 3 Description of Data

### 3.1 Data Preprocessing

For our project, we used data from three sources.

1. Bureau of Transportation Statistics (BTS) (Airline On-Time Performance Data)
2. National Oceanic and Atmospheric Administration (NOAA) (IBTrACS-WMO Data)
3. OpenFlights (airports.dat)

BTS data was used to get on-time performance of each flight. OpenFlights was used to get the longitude and latitude of each airport as well as the UTC offset. NOAA was used to get hurricane information. The NOAA storm information is given at every 6 hour intervals.

The primary cleaning task we performed was standardizing the departure time in BTS, which was presented as local time to the region, to UTC time with the help of OpenFlights. Then we computed the distance of the origin and destination coordinate to the center of the storm. Afterwards, we merged BTS and OpenFlights on origin and destination coordinates. Then we cut the departure time into the same 6 hour intervals as the NOAA storm data to do the final merge.

### 3.2 Data Exploration

Large and medium hubs are those as defined in [Wikipedia list of the busiest airports](#). The list is based off FAA standards since 2014 and seem reliable enough. An interactive version of the bubble plot allowing the option to hover over and see the actual values and the animation of the map are available in the first link of the Bibliography [6](#).

### 3.2.1 Exploring the NOAA Storm Data

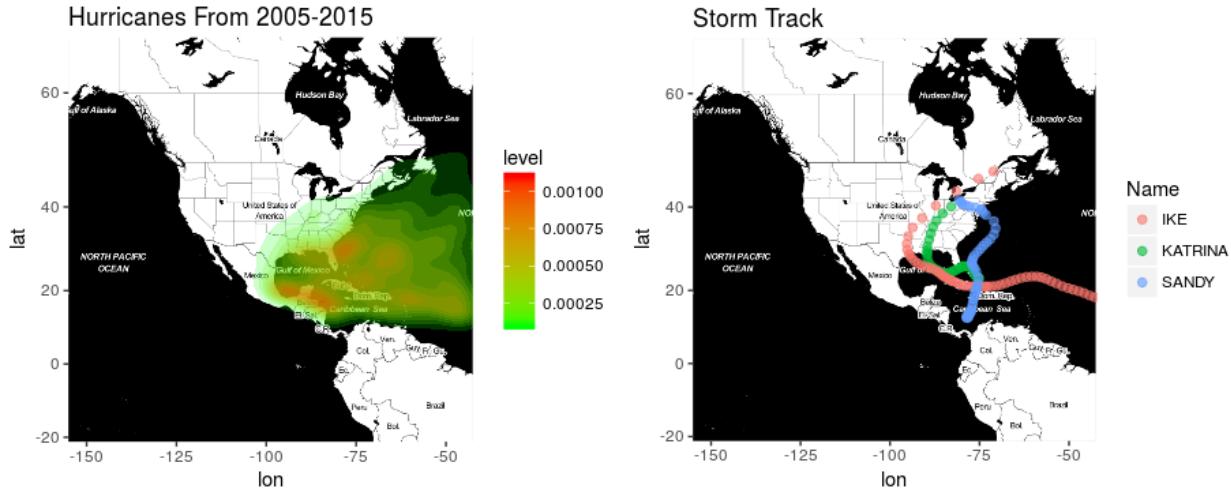


Figure 1: Left: Heatmap of Hurricanes from 2005-2015. Right: Storm track for the three storms studied.

From the figure above, we see that Atlantic hurricanes affect primarily the Central and Eastern parts of the US. Hurricane Katrina and Hurricane Ike went through Central US, while Hurricane Sandy trailed along the east coast.

Table 1: Hurricanes Characteristics by Year

Year	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
Num of Hurricanes	31	10	17	17	11	20	20	19	15	9	12
Max Wind Speed	160	105	150	135	115	135	120	100	80	125	135
Median Wind Speed	45	45	35	45	35	40	45	45	35	40	35

Table 2: Hurricane Characteristics by Month 2005-2015

Month	1	5	6	7	8	9	10	11	12
Num of Hurricanes	1	7	14	24	51	67	41	10	4
Max Wind Speed	55	65	80	140	150	155	160	125	75
Median Wind Speed	45	35	35	40	40	45	40	45	45

Between 2005 to 2015, there were 181 hurricanes recorded by NOAA. The year 2005 and months August to October has the most number of hurricane observations. From table

1,2005 has the highest windspeed probably from Hurricane Katrina. After 2005, most hurricanes falls under the category 4 (Windspeed 130-156mph) and 3 (Windspeed 111-129mph) region. From table 2, we see that not only do August to October witness the most hurricanes, but also more destructive hurricanes. From the median perspective, there is not too much fluctuation across years and months.

### 3.2.2 Exploring the Relationship Between Ontime Performance and Storm

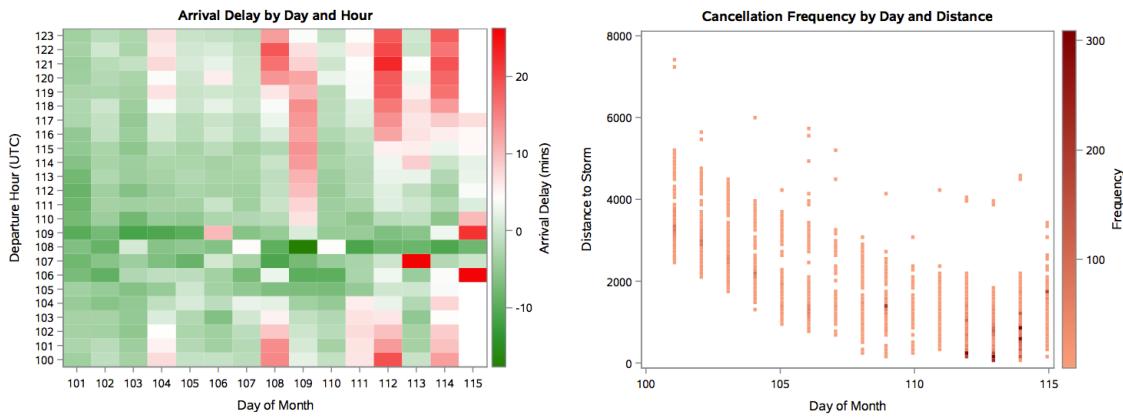


Figure 2: Exploring Delay and Cancellation for Katrina

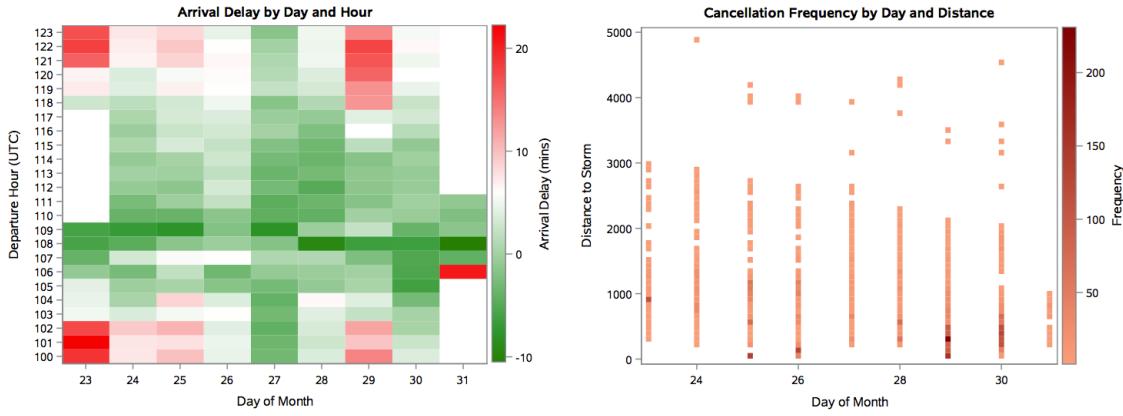


Figure 3: Exploring Delays and Cancellation for Ike

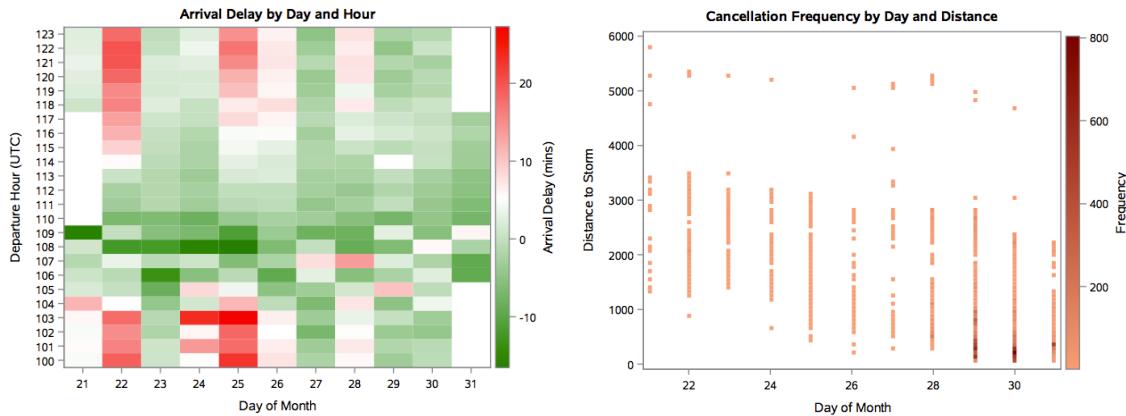


Figure 4: Exploring Delays and Cancellation for Sandy

To begin with, the tickmarks in some of the plots are weird because for some reason SaS weren't able to order the values. To get an ordering, we added 100 to the values. So 101 - 115 in days mean day 1-15 of the month and 100-123 in hours mean 0-23 in hour.

From these plots, we can see that delays and cancellations were consistent during Hurricane Katrina. However, in Ike and certainly Sandy, there were a lot less delays on days where there were a lot of cancellations. This would suggest that airlines are simply cancelling flights during the storm. In Sandy, the range of total cancelled flights went as high as 800 on the scale.

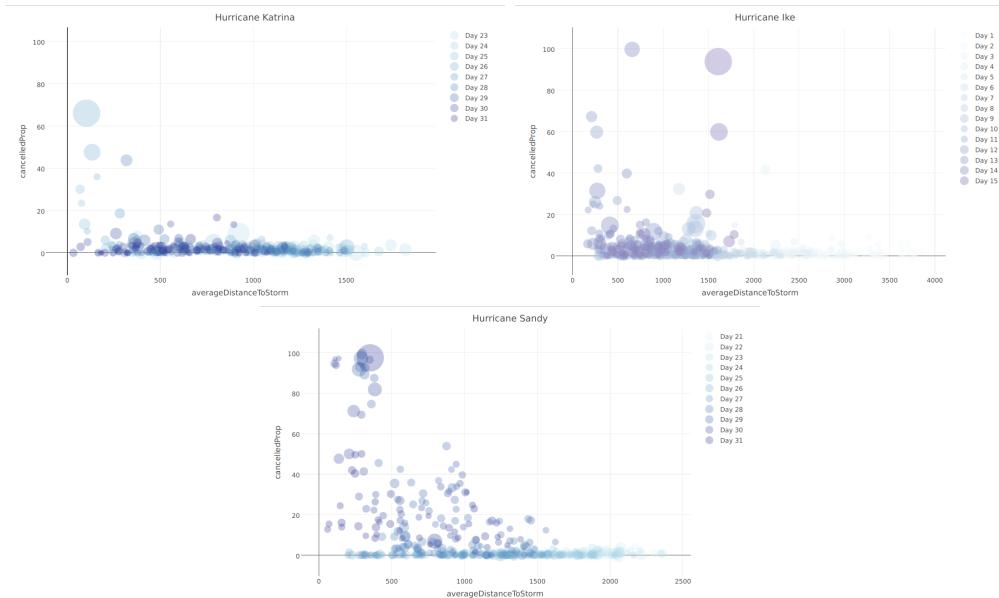


Figure 5: Bubble Plots of Katrina, Ike, Sandy

A bubble plot is plotted to explore the relationship between departure distance from storm center against percentage of cancellation. The size of the bubble measures the average delay

at different airport locations. Each color of the bubble represent different days of the storm available in the NOAA data. Only large and medium airport hubs in the Central and Eastern timezones have been used for the plots to avoid small airports with little flights showing up with high cancellation.

From the plots, we see that Sandy seem to have more hubs experiencing cancellations than Ike and Katrina. By the color of the bubble, we can identify the days that the hurricanes cause the highest cancellation with respect to the hub's position to the storm center. With the exception of HOU airport on day 15 for Ike, most airports experience higher cancellation when their proximity to hurricane center increase, as we would expect. The larger the bubble indicates higher average delay for the hub, but from the plot we see that most of the bubble do not grow in size with decrease in departure distance to the center of the storm. Also some airports with 100% cancellation weren't shown in the plot since they would have no delay information available to generate the size of the bubble, but we have noted them down in the table 3.

Table 3: Hurricanes 100% Cancellation

Hurricane	Hub(Days)
IKE	MSY(1,2,3), HOU(13,14), IAH (13)
KATRINA	MSY(29,30,31)
SANDY	EWR(30,31), JFK(30), LGA (30,31)

For Ike the cancellation that resulted on days 1-3 was due to another hurricane striking right before Ike at the end of August. Overall, from the bubble plot it seems like hurricane that strike along the East Coast will lead to more cancellations than hurricane moving inland from the southern tip as illustrated by the storm track of Ike and Katrina. Also, there is no noticeably bad delays resulting from hurricane weather. Most of the delays are under an hour on average. Furthermore, we see that most cancellation activity occur within 1000 miles of hurricane center.

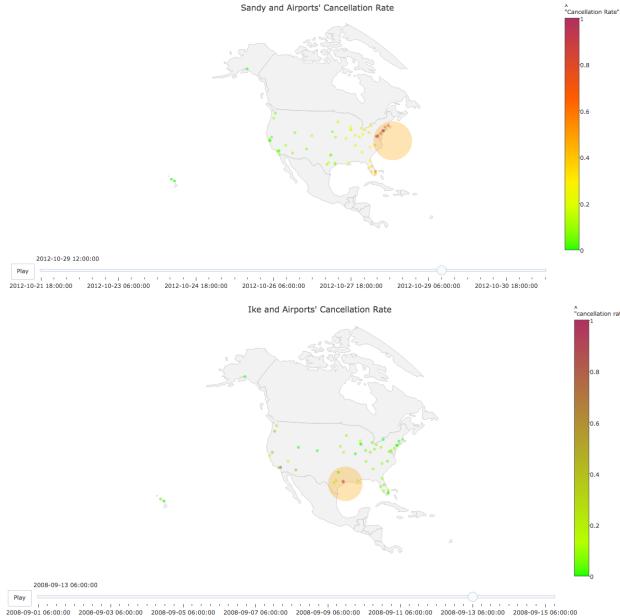


Figure 6: Animated Map of Sandy and Ike

Focusing only on the medium and large hubs as in the case of the bubble plots, we also animated the storm track through the US and see the percentage of cancellation rates. The plots shown in figure 3 shows the first time the hurricane comes into contact with land. Only Sandy and Ike were plotted because the shapefile data only has information starting from 2008.

When Sandy hits the eastern shores, we can see all the hubs in the area going toward red. However, when Ike hits, most of the East Coast remain green and we see some points in the West turning red. Our conjecture for the reason that the West have cancellation while the East remain green was due to the time difference. 6 AM UTC is 2 AM EST in the morning and around 11 PM in PST. From data exploration and research, we know that almost all the airports reduce or have no departures after 12 AM midnight to 5 AM in the morning, explaining why the East have little to no cancellations. However due to the storm, for flights departing from the West Coast going toward the East, they seem to end up having increased cancellations.

After studying the patterns throughout the animation, we also come to the same conclusion as the bubble plots in that hurricanes coming up from the South to Central US tend to have result in less cancellation than a hurricane striking the East Coast.

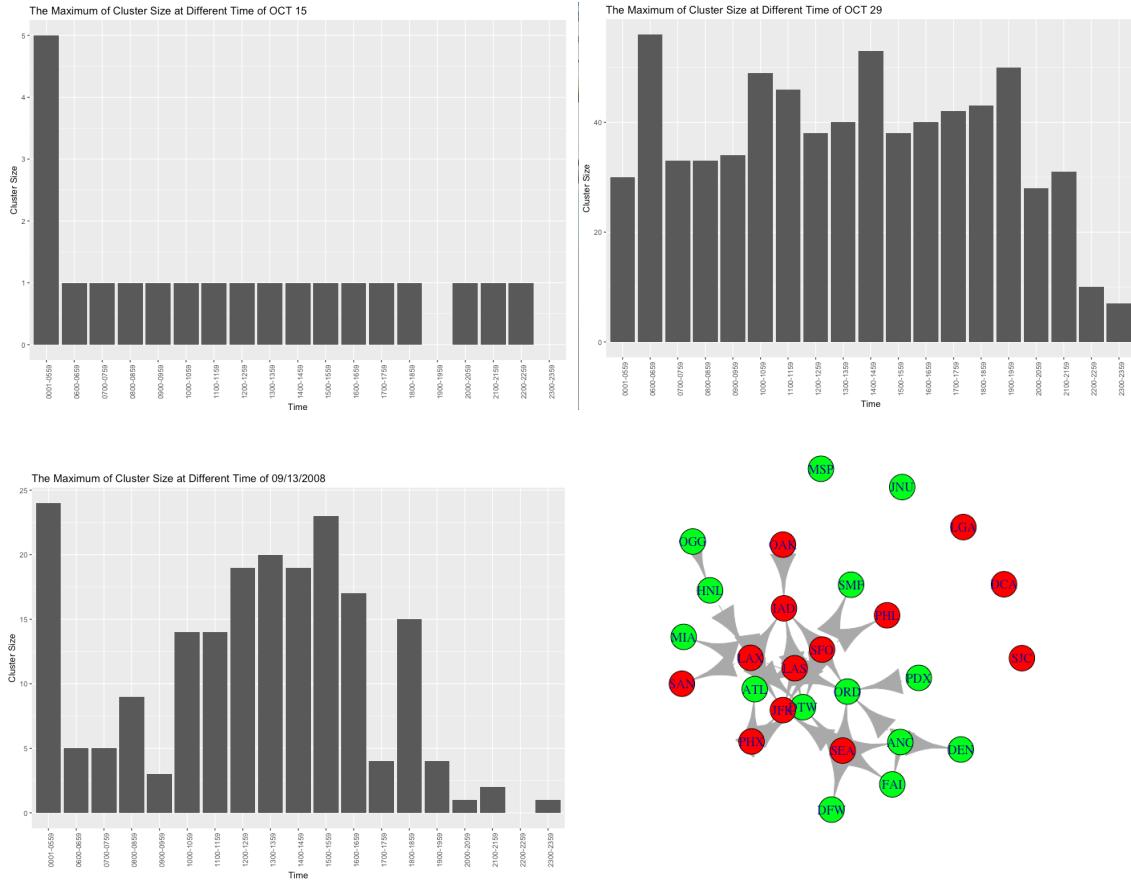


Figure 7: Top Left: Cancellation Cluster sizes on a regular day for Sandy. Top Right: Cancellation Cluster sizes on the day Sandy hit East Coast shores. Bottom Left: Cancellation Cluster sizes on the day Ike hit Central shores. Bottom Right: Igraph of Network Clusters

In the Igraph, the nodes are the airports and the edges are existing routes connecting them. By clustering the nodes, we can figure out how many airports with severe cancellation rate (more than 20%) are connected to each other. These clusterization helps us understand how the effect of cancellation from one airport propagates to the others. The barplots show the maximum sizes of clustering at different times of the day when Ike and Sandy hits the US costals, comparing to normal day in the Sandy time range. For Sandy, the clusters seem to be uniformly distributed as oppose to Ike, where seem more normally distributed. This is also consistent to our earlier claim that hurricane hitting the Eastern coastal will create a heavier effect on cancellation rates than hitting the Southern coastal regions.

## 4 Model

### 4.1 Motivation

In the following, we try to predict the probabilities that particular flights are going to be cancelled based on different factors, such as distances between the airports and the center of the storm, the carrier of the flights, and the wind speed and pressure of the hurricane. We thought such prediction is interesting given that if airports can forecast the proportion of cancellations prior to the next 6 hour hurricane measurement, maybe they will be more prepared and can better allocate their resources.

We use all data points from hurricane Katrina, Sandy, and Ike. We then divide it into 80% training data and 20% testing data.

Our first intuition is to use a logistic regression model. However, the result is not very satisfying. It is very conservative and poor at predicting cancelled flights. It only makes 21 attempts in guessing the flights are going to be cancelled. See top chart of table 4 for confusion matrix of results.

One reason that our model does very poor is because we violate the assumption of logistic regression that each observation in the data should be independent. However, as shown in the Igraph above, airports' cancellation rate are dependent on each other. For example, when the storm hits Atlanta and causing a lot of cancellation, flights coming from San Francisco are going to be cancelled too.

### 4.2 Findings

We then decided to use a random forest model, since requires no distribution assumptions or need to take into account spatial autocorrelations. We included airlines, arrival and departure distances from the storm, nature of the storm, wind speed and pressure of the storm, and departure hour in our model.

Our model becomes better at guessing the cancelled flights and we have a misclassification rate of only 1.87%. 8% of the flights are cancelled during days of the three hurricanes. See bottom chart of table 4 for confusion matrix of results.

In the end, we also test our data with hurricane Dennis, which our training dataset does not include. We still manage to maintain a misclassification rate of 2.48%. However, Hurricane Dennis itself contained a high proportion of non-cancellations and we didn't test on more hurricanes to verify if the random forest trained only on 3 storms will do well.

Test Data		Actual	
		Not Cancelled	Cancelled
Predicted	Not Cancelled	110942	0
	Cancelled	4659	21
Dennis		Actual	
		Not Cancelled	Cancelled
Predicted	Not Cancelled	110700	242
	Cancelled	1921	2759

Table 4: Confusion Matrix for Random Forest with Testing Data and Hurricane Dennis

### 4.3 Model Limitation

We realize that analyzing the data of hurricane can be very challenging and complicated. In order to achieve a more accurate result, we should look into spatial-temporal models or time series autoregressive models. Maybe it's possible for us to add in a time lag to improve model performance. But learning about the knowledge in these analyses is beyond the scope of this project under the time constraint. We also weren't able to spend time tuning model parameters and doing cross-validation, since the bulk of our project was spent on exploring the dataset and we kind throw in a model last minute to see if there is anything interesting. Furthermore, we only combined 3 hurricanes as our data and its not a random sample either, so that might be problematic. Next time, if we do want to proceed with such analysis, maybe include all the hurricane data information. However to get the data for all the flight information will be a pain, since we need to manually click and download each month. Overall, our model can only be seen as an exploratory effort than something solid people can apply.

## 5 Conclusion

The data sets for U.S. domestic flights are extremely large. Combining with hurricane data, it was quite a challenge to manage, subset, and analyze our data effectively. In addition, it was difficult to design a model that took into account the vast number of variables and spatial, temporal dependencies within our data. Overall, we can tell that hurricanes with a path in Northeastern United States will have a bigger effect on flights than a hurricane going through Southeastern United States. This makes sense as there are generally more flights heading to and taking off from the major hubs in the East Coast (such as New York City). With bigger hurricanes, we can also identify with high certainty whether a flight will

be cancelled. Smaller hurricanes present a challenge since they rarely hit mainland U.S., and even when they do, they rarely affect the flights. Overall, we would need to further explore the data with spatial analysis to create a more accurate model.

## 6 Bibliography

1. [Website with Animation and Interactive Plots](#)
2. [On-time Performance Data of Flights](#)
3. [Hurricane Data](#)
4. [Airport Location Information](#)
5. [System Delay Propagation in the US Airport Network](#)
6. [JetBlue Earliest to Cancel during Storms, Fewer Refunds Result](#)
7. [Applied Spatial Statistics in R](#)
8. [Assumptions of Logistic Regression](#)

## 7 Project Reflections

The biggest challenge we have for this project was actually identifying and clarifying the question we want to answer and why any of this matters. We spend basically 9 out of the 10 weeks changing and revising our project to try to make it meaningful. Even up to the last minute, changes are being made to the project and we are constantly fighting with the question 'why it matters and to who?'. So if we learn anything in these 10 weeks, it was how important it is to have a well-define project to work with. Since our project goals were so unclear, we weren't able to spent much on learning a new method and applying it. If we were to do a similar project in the future, we believe the goal should be to make sure everyone is on the same page with a well-define goal way early in the project so we can have as little revision as possible and something to really focus on throughout. Realistically speaking, having to define a problem that is interesting not only to you but to stakeholders is a big challenge itself. Even if we have access to the right data, but we don't have a good, solid questions, it is still difficult to go through with a project and be easily discouraged during the process. Therefore, we think the project design should always begin with the question then find the data for that question and not the other way around.

## 8 Code Appendix

```

1  read_data_from_BTS = function(datadir){
2    data = read.csv(datadir, stringsAsFactors = FALSE)
3    colWanted = c("Year", "Quarter", "Month", "DayofMonth", "DayOfWeek", "FlightDate"
4      ,
5        "UniqueCarrier", "TailNum", "Origin", "OriginCityName", "
6        OriginState",
7          "Dest", "DestCityName", "DestState", "DepTime", "CRSDepTime", "
8        DepDelay",
9          "DepDelayMinutes", "DepDel15", "DepTimeBlk", "TaxiOut", "WheelsOff"
10         , "WheelsOn",
11           "TaxiIn", "CRSArrTime", "ArrTime", "ArrDelay", "ArrDelayMinutes", "
12         ArrDel15",
13           "ArrTimeBlk", "CRSElapsedTime", "ActualElapsedTime", "Distance",
14             "DistanceGroup", "Cancelled", "CancellationCode",
15               "CarrierDelay", "WeatherDelay", "NASDelay", "SecurityDelay",
16                 "LateAircraftDelay", "Diverted")
17   data = data[, colWanted]
18
19
20
21
22
23
24
25
26
27
28
29
30
#Merge GPS Location
getGpsLocation = function(data){
  airports = read.csv("~/media/sf_Windows/FlightData/airports.dat", header =
    FALSE,
    col.names = c("ID", "Name", "City", "Country", "IATA", "ICAO"
      ,
      "Lat", "Lon", "Altitude", "Timezone", "DST", "
    Tz", "Type", "Source"))
  USairports = airports[airports$Country == 'United States', c("IATA", "Lat", "
    Lon", "Timezone")]
  finaldf = merge(data, USairports, by.x = 'Origin', by.y = 'IATA', all.x =
    TRUE)
  names(finaldf)[names(finaldf) %in% c('Lat', 'Lon', 'Timezone')] = paste("
    Origin", c('Lat', 'Lon', 'Timezone'))
  finaldf = merge(finaldf, USairports, by.x = 'Dest', by.y = 'IATA', all.x =
    TRUE)
  names(finaldf)[names(finaldf) %in% c('Lat', 'Lon', 'Timezone')] = paste("Dest"
    , c('Lat', 'Lon', 'Timezone'))
  # #Fix timezone. Convert UTC offset to US timezone.
  finaldf$'Dest Timezone' = factor(finaldf$'Dest Timezone')
  levels(finaldf$'Dest Timezone') = c("US/Hawaii", "US/Eastern", "US/Central", "
    US/Mountain", "US/Pacific", "US/Alaska")
  finaldf$'Origin Timezone' = factor(finaldf$'Origin Timezone')

```

```
31 levels(finaldf$'Origin Timezone') = c("US/Hawaii","US/Eastern","US/Central",
32 "US/Mountain","US/Pacific","US/Alaska")
33 finaldf = finaldf[!is.na(finaldf$"Dest Timezone") & !is.na(finaldf$"Origin
34 Timezone"),]
35 return(finaldf)
36 }
37
38 library(lubridate)
39 convertToUTC = function(row){
40   row[2] = as.numeric(row[2])
41   if(nchar(row[2]) == 3)
42     row[2] = paste0("0",row[2])
43   if(nchar(row[2]) == 2)
44     row[2] = paste0("00",row[2])
45   if(nchar(row[2]) == 1)
46     row[2] = paste0("000",row[2])
47   date = paste(row[1:2], collapse = " ")
48   return(date)
49 }
50
51 getDistanceFromLatLonInKm = function (lat1,lon1,lat2,lon2) {
52   R = 6371;
53   dLat = deg2rad(lat2-lat1)
54   dLon = deg2rad(lon2-lon1)
55   a =
56     sin(dLat/2) * sin(dLat/2) +
57     cos(deg2rad(lat1)) * cos(deg2rad(lat2)) *
58     sin(dLon/2) *sin(dLon/2)
59   c = 2 * atan2(sqrt(a), sqrt(1-a))
60   d = R * c * 0.621371
61   return(d)
62 }
63
64 deg2rad = function (deg) {
65   return(deg * (pi/180))
66 }
67 #Input DataFile
68 processStorm = function(dataFile) {
69   stormdf = read_data_from_BTS(dataFile)
70
71   #Identify Lon and Lat as well as timezone.
72   df= getGpsLocation(stormdf)
73
74   df$DepDateTime = apply(df[,c("FlightDate","CRSDepTime")], 1,convertToUTC)
75
76   cols = paste0("UTC",c("Year", "Month", "DayofMonth",
77   "DayOfWeek", "DepHour", "CRSDepTime"))
78 }
```

```
78 df[cols] = NA
79
80 for(timezone in c("US/Hawaii","US/Eastern","US/Central",
81                   "US/Mountain","US/Pacific",""
82                   "US/Alaska")){
83   date = strptime(df$DepDateTime[df$'Origin Timezone' == timezone],
84                   format="%Y-%m-%d %H%M", tz = timezone)
85   date = with_tz(date, tzone = 'UTC')
86   newdf = data.frame(Year = year(date), Month = month(date), Day = day(date),
87                      DayOfWeek = weekdays(date), Hour = hour(date), Time = strftime(
88                      date, format="%Y-%m-%d %H:%M%S"))
89   newdf$Time = as.character(newdf$Time)
90   newdf$DayOfWeek = as.character(newdf$DayOfWeek)
91   df[df$'Origin Timezone' == timezone, cols] = newdf
92 }
93
94 #Put time in blocks
95 utcTimes = strptime(df$UTCCRSDepTime, format="%Y-%m-%d %H:%M%S", tz = 'UTC')
96 SixHourBlock = cut(utcTimes, breaks = "6 hour")
97 df$DepSixHourBlock = SixHourBlock
98 return(df)
```

Listing 1: Data Merging in R

```
1 #Animation Plot
2 plot_geo(locationmode = "USA-states") %>%
3   add_markers(
4     data = cancel_rate_ike_hubs, x = ~Lon, y = ~Lat, text = ~Name, frame = ~
5       Group.2,
6     hoverinfo = "text", alpha = 0.5, color = ~x, colors = c("green","yellow",
7       "orange","red","maroon"))
8   ) %>%
9   add_markers(
10    data = points_ike, x = ~LONG, y = ~LAT, frame = ~ISO_time, alpha = 0.3,
11    size = ~INTENSITY,
12    color = ~'cancellation rate'
13  ) %>%
14  layout(
15    title = paste("Ike and Airports' Cancellation Rate"), geo = geo,
16    showlegend = FALSE
17  ) %>%
18  animation_opts(500, easing = "elastic", redraw = T
19  ) %>%
20  animation_slider(
21    currentvalue = list(prefix = "Time Block ", font = list(color="Black")))
22
23 #Clustering
```

```

21 cluster = sapply(1:19 ,function(i){
22   routes [[ i ]] $Origin = factor(routes [[ i ]] $Origin)
23   routes [[ i ]] $Dest = factor(routes [[ i ]] $Dest)
24   airports = data.frame(name = delay_airport [[ i ]] $Group.1 ,congested = delay_
25     airport [[ i ]] $congested)
26   paths = data.frame(from = routes [[ i ]] $Origin , to = routes [[ i ]] $Dest)
27   paths = paths[which(paths [,2] %in% airports$name) ,]
28   paths = paths[which(paths [,1] %in% airports$name) ,]
29   g = graph_from_data_frame(paths ,vertices = airports)
30   V(g)$color = ifelse(airports$congested == 1,"red","green")
31   g = induced.subgraph(g,V(g)[V(g)$color %in% c("red")])
32   plot(g)
33   max(clusters(g)$csize)
34 })
35 p = ggplot(data = data15 ,aes(x = unique.cancel_rate_15.Group.2. , y = cluster_
36   15)) +
37   geom_bar(stat = "identity") + ggtitle("The Maximum of Cluster Size at
38   Different Time of OCT 15") +
      xlab("Time") + ylab("Cluster Size") + theme(axis.text.x = element_text(
      angle = 90 , hjust = 1))
39 p

```

Listing 2: Animation Plot in R

```

1 #Modeling
2 extractFeatures = function(data){
3   features = c("UniqueCarrier",
4               "DepDistanceToStorm",
5               "ArrDistanceToStorm",
6               "UTCDayofMonth",
7               "UTCDepHour",
8               "Nature",
9               "Wind.WMO.",
10              "Pres.WMO.")
11 fea = data[,features]
12 fea$UniqueCarrier = as.factor(fea$UniqueCarrier)
13 fea$Nature = as.factor(fea$Nature)
14 return(fea)
15 }
16 set.seed(123)
17 modrf = randomForest(extractFeatures(train),as.factor(train$Cancelled),ntree =
18   1000,importance = T)
19 test$predict = predict(modrf,extractFeatures(test))

```

Listing 3: Random Forest in R

```

1 p1 <- plot_ly(result[result>Name == "KATRINA"], x = ~averageDistanceToStorm ,
2                 y = ~cancelledProp ,

```

```
2         type = 'scatter', mode = 'markers', size = ~averageDepDelay ,  
3         color = ~DayofMonthLabel ,  
4             sizes = c(10, 50) ,  
5             marker = list(opacity = 0.3, sizemode = 'diameter') ,  
6             colors= "RdYlBu" ,  
7             hoverinfo = 'text' ,  
8             text = ~paste('Airport:', ORIGIN,  
9                             '  
10                            Day of Month:', DayofMonth,  
11                            '  
12                            %of Cancelled Flights:', cancelledProp ,  
13                            '  
14                            Flights Scheduled for Airport:',  
15                            proportionFlights ,  
16                            '  
17                            Average Departure Delay:', averageDepDelay))  
18  
19 layout(title = 'Hurricane Katrina ')
```

Listing 4: Sample Code for Bubble Plots in R

## Listings

1	Data Merging in R . . . . .	12
2	Animation Plot in R . . . . .	15
3	Random Forest in R . . . . .	16
4	Sample Code forBubble Plots in R . . . . .	16