

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

1 Abstract	2
2 Introduction	2
3 Description of Data	2
3.1 Data Preprocessing	2
3.2 Data Exploration	2
3.2.1 Exploring the NOAA Storm Data	3
3.2.2 Exploring the Ontime Performance Data	4
3.2.3 Exploring the Relationship Between Ontime Performance and Storm	4
4 Model	7
4.1 Limitations	8
5 Conclusion	8
6 Project Reflections	8
7 Code Appendix	8

1 Abstract

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 we examined study in depth were Sandy (10/2012, Class: 3), Ike (09/2008, Class 4), and Katrina (08/2005, Class: 5). Other hurricanes were selected to build a random forest prediction model.

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 and seem reliable.

3.2.1 Exploring the NOAA Storm Data

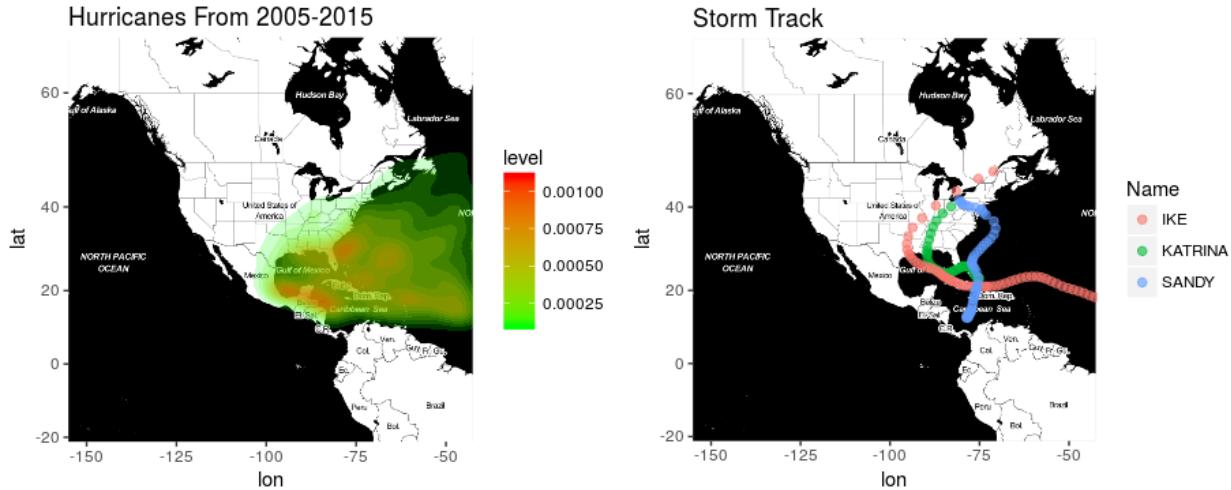


Figure 1: Left: Heatmap of Hurricanes from 2005-2015. Right: Storm track for the three storms studied.

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 Ontime Performance Data

3.2.3 Exploring the Relationship Between Ontime Performance and Storm

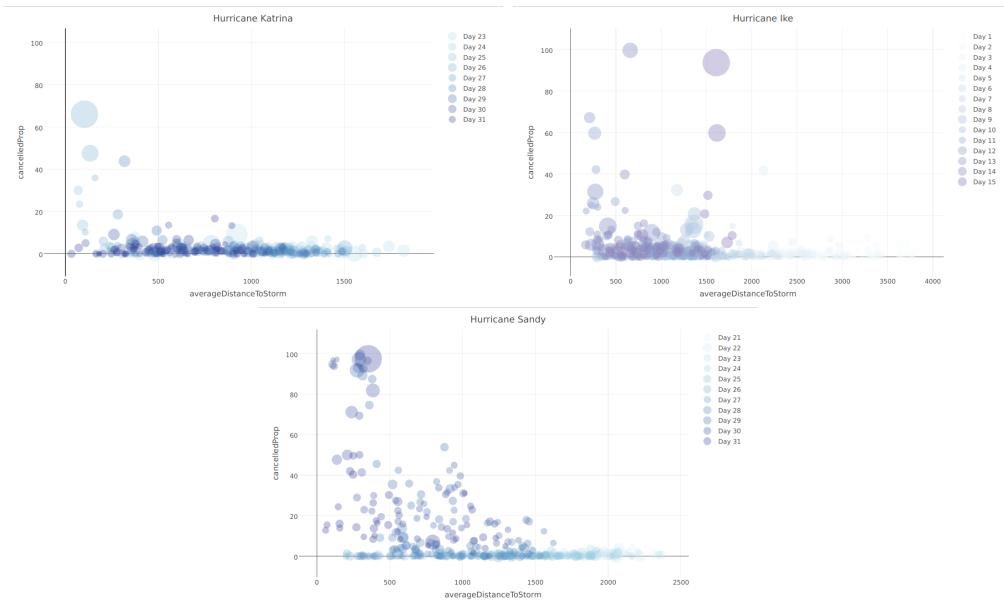


Figure 2: Bubble Plots of Katrina, Ike, Sandy

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. An interactive version of the plot allowing the option to hover over and see the actual values are available in the links below. 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 large 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.

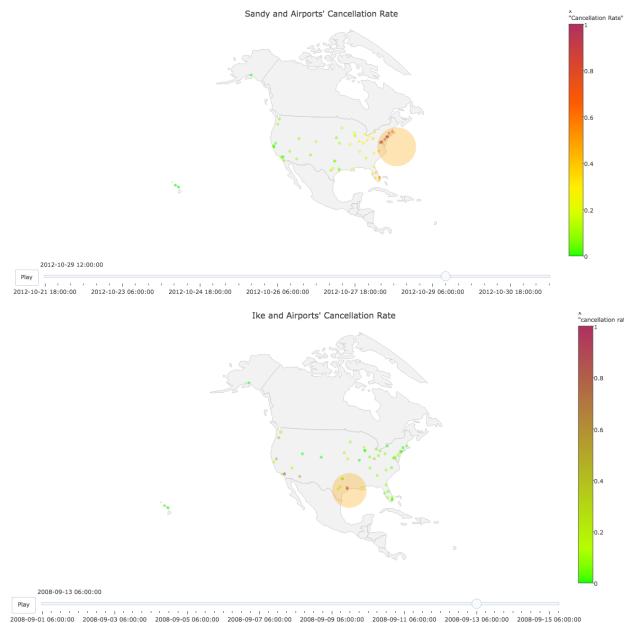


Figure 3: Animated Map of Sandy and Ike

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 since 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.

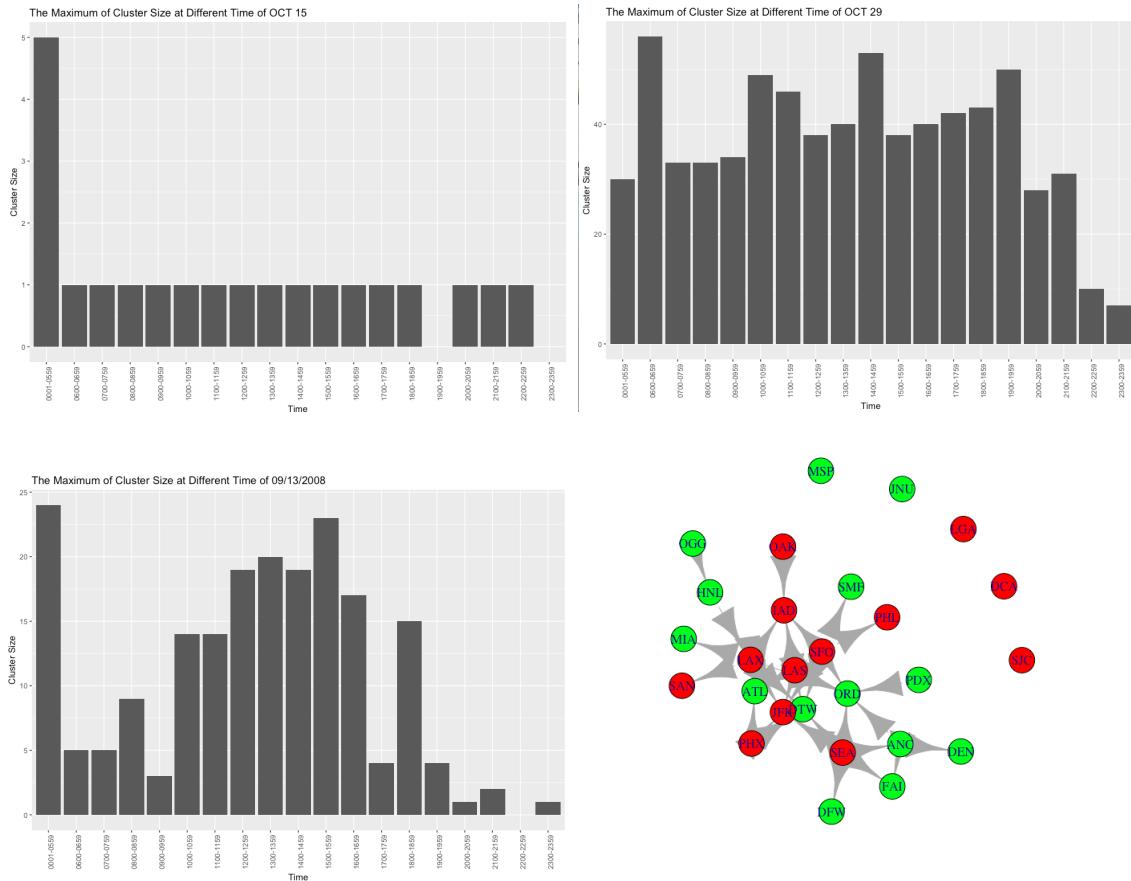


Figure 4: Top Left: Cluster sizes on a regular day. Top Right: Cluster sizes on the day Sandy hit East Coast shores. Bottom Left: Cluster sizes on the day Ike hit Central shores. Bottom Right: Network Clusters

4 Model

Our motivation for creating a prediction model was to be able to predict the percentage of cancellations at the next 6 hour interval of the hurricane based off the distance away from the hurricane center. We believe that such prediction will help airlines maybe further away to better allocate their resources and be prepared for the storm. (WHY? HOW?)

The most intuitive model we have for such problem was to use logistic regression adjusting for distance away from hurricane center, wind speed, atmospheric pressure, departure hour, and month of the year. However, we realize that a logistic model would not work well here due to the spatial and temporal dependence of the response variable, which violates the i.i.d assumption of the response variable.

We tried looking

4.1 Limitations

5 Conclusion

6 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-defined project to work with.

7 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   # data = data[data$Origin %in% allAirports &  
20   #           data$Dest %in% allAirports ,]  
21   data  
22 }  
23  
24 #Merge GPS Location  
25 getGpsLocation = function(data){  
26   airports = read.csv("~/media/sf_Windows/FlightData/airports.dat", header =  
27     FALSE,  
28     col.names = c("ID", "Name", "City", "Country", "IATA", "ICAO"  
29     ,
```

```
21             "Lat", "Lon", "Altitude", "Timezone", "DST", "
22             Tz", "Type", "Source"))
23 USairports = airports[airports$Country == 'United States', c("IATA", "Lat",
24             "Lon", "Timezone")]
25 finaldf = merge(data, USairports, by.x = 'Origin', by.y = 'IATA', all.x =
26             TRUE)
27 names(finaldf)[names(finaldf) %in% c('Lat', 'Lon', 'Timezone')] = paste("
28             Origin", c('Lat', 'Lon', 'Timezone'))
29 finaldf = merge(finaldf, USairports, by.x = 'Dest', by.y = 'IATA', all.x =
30             TRUE)
31 names(finaldf)[names(finaldf) %in% c('Lat', 'Lon', 'Timezone')] = paste("Dest"
32             , c('Lat', 'Lon', 'Timezone'))
33 # #Fix timezone. Convert UTC offset to US timezone.
34 finaldf$'Dest Timezone' = factor(finaldf$'Dest Timezone')
35 levels(finaldf$'Dest Timezone') = c("US/Hawaii", "US/Eastern", "US/Central",
36             "US/Mountain", "US/Pacific", "US/Alaska")
37 finaldf$'Origin Timezone' = factor(finaldf$'Origin Timezone')
38 levels(finaldf$'Origin Timezone') = c("US/Hawaii", "US/Eastern", "US/Central",
39             "US/Mountain", "US/Pacific", "US/Alaska")
40 finaldf = finaldf[!is.na(finaldf$'Dest Timezone') & !is.na(finaldf$'Origin
41             Timezone'),]
42 return(finaldf)
43 }
44
45 library(lubridate)
46 convertToUTC = function(row){
47     row[2] = as.numeric(row[2])
48     if(nchar(row[2]) == 3)
49         row[2] = paste0("0", row[2])
50     if(nchar(row[2]) == 2)
51         row[2] = paste0("00", row[2])
52     if(nchar(row[2]) == 1)
53         row[2] = paste0("000", row[2])
54     date = paste(row[1:2], collapse = " ")
55     return(date)
56 }
57
58 getDistanceFromLatLonInKm = function(lat1, lon1, lat2, lon2) {
59     R = 6371;
60     dLat = deg2rad(lat2 - lat1)
61     dLon = deg2rad(lon2 - lon1)
62     a =
63         sin(dLat/2) * sin(dLat/2) +
64         cos(deg2rad(lat1)) * cos(deg2rad(lat2)) *
65         sin(dLon/2) * sin(dLon/2)
66
67     c = 2 * atan2(sqrt(a), sqrt(1-a))
68     d = R * c * 0.621371
69     return(d)
```

```

61 }
62
63 deg2rad = function (deg) {
64   return(deg * (pi/180))
65 }
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$DepDateTime = apply(df[,c("FlightDate","CRSDepTime")], 1,convertToUTC)
75
76   cols = paste0("UTC",c("Year", "Month", "DayofMonth",
77                     "DayOfWeek", "DepHour", "CRSDepTime"))
78   df[ cols ] = NA
79
80   for(timezone in c("US/Hawaii","US/Eastern","US/Central",
81                     "US/Mountain","US/Pacific","US/Alaska")){
82     date = strptime(df$DepDateTime[df$'Origin Timezone' == timezone],
83                      format="%Y-%m-%d %H:%M", tz = timezone)
84     date = with_tz(date, tzone = 'UTC')
85     newdf = data.frame(Year = year(date),Month = month(date), Day = day(date),
86                         DayOfWeek = weekdays(date), Hour = hour(date), Time = strftime(
87                           date, format="%Y-%m-%d %H:%M:%S"))
88     newdf$Time = as.character(newdf$Time)
89     newdf$DayOfWeek =as.character(newdf$DayOfWeek)
90     df[df$'Origin Timezone' == timezone,cols] = newdf
91   }
92
93   #Put time in blocks
94   utcTimes = strptime(df$UTCCRSDepTime, format="%Y-%m-%d %H:%M:%S", tz = 'UTC')
95   )
96   SixHourBlock = cut(utcTimes, breaks = "6 hour")
97   df$DepSixHourBlock = SixHourBlock
98   return(df)
99 }
```

Listing 1: Data Merging in R

Listings