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 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

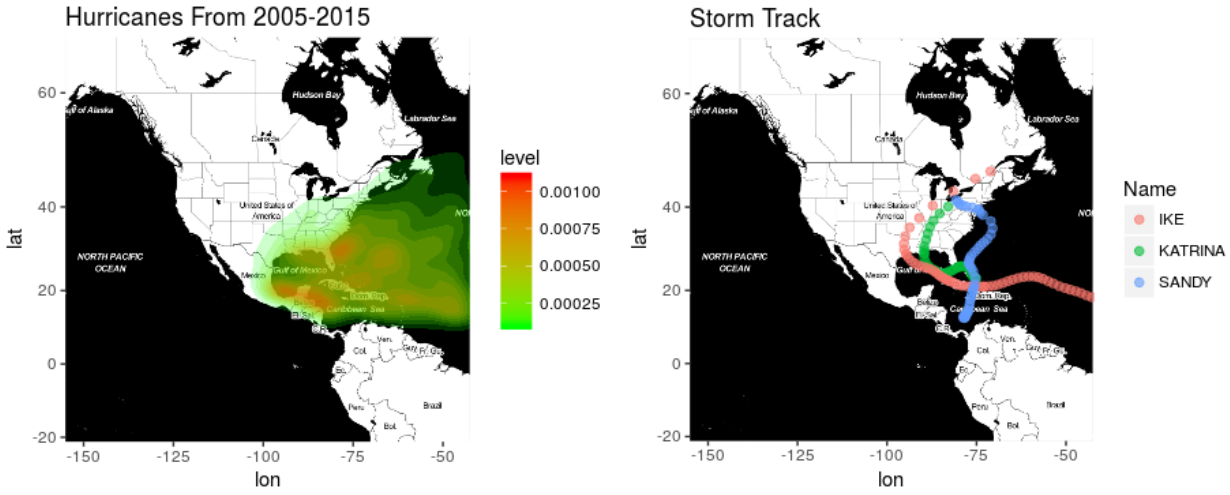

Figure 1: Left: Heatmap of Hurricanes from 2005-2015. Right: Storm track for the three storms studied.

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

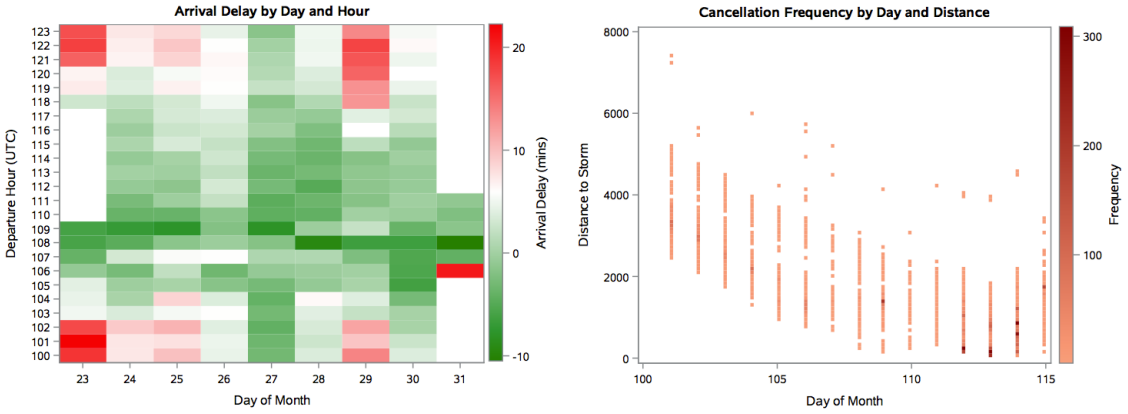

Figure 2: Exploring Delays in Katrina

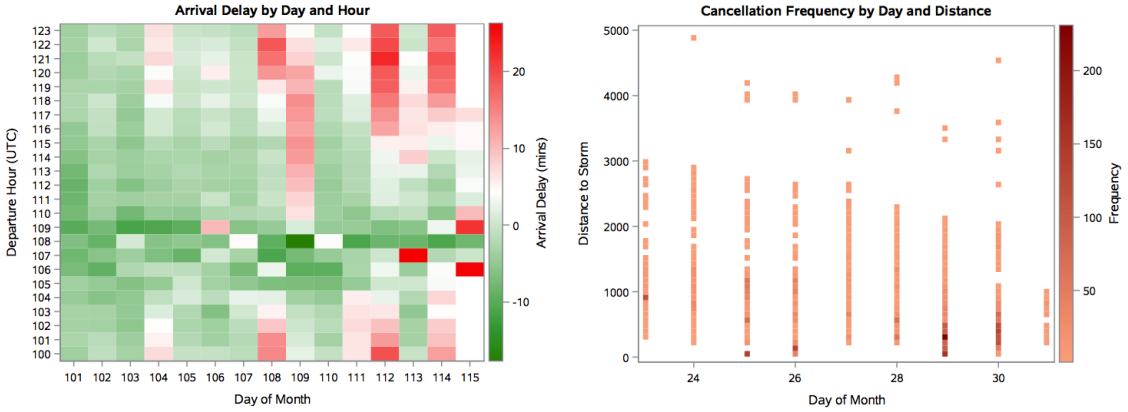

Figure 3: Exploring Delays in Ike

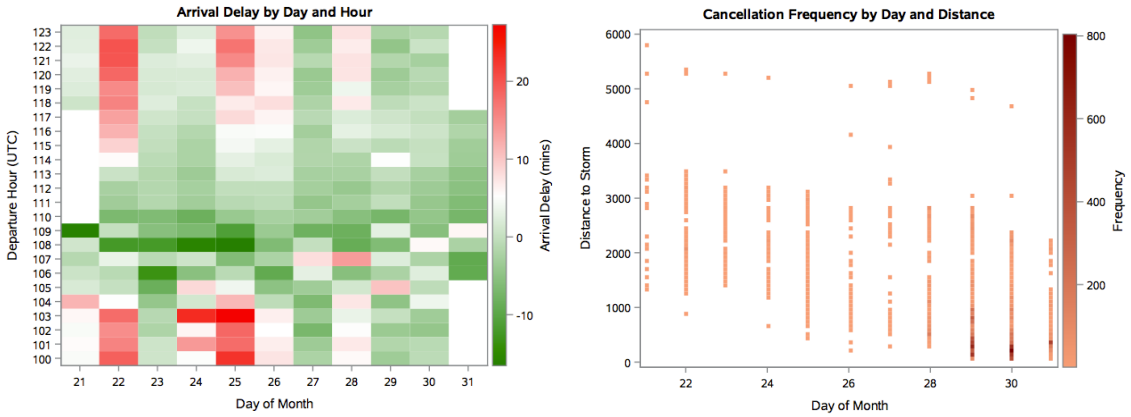

Figure 4: Exploring Delays in Sandy

3.2.3 Exploring the Relationship Between Ontime Performance and Storm

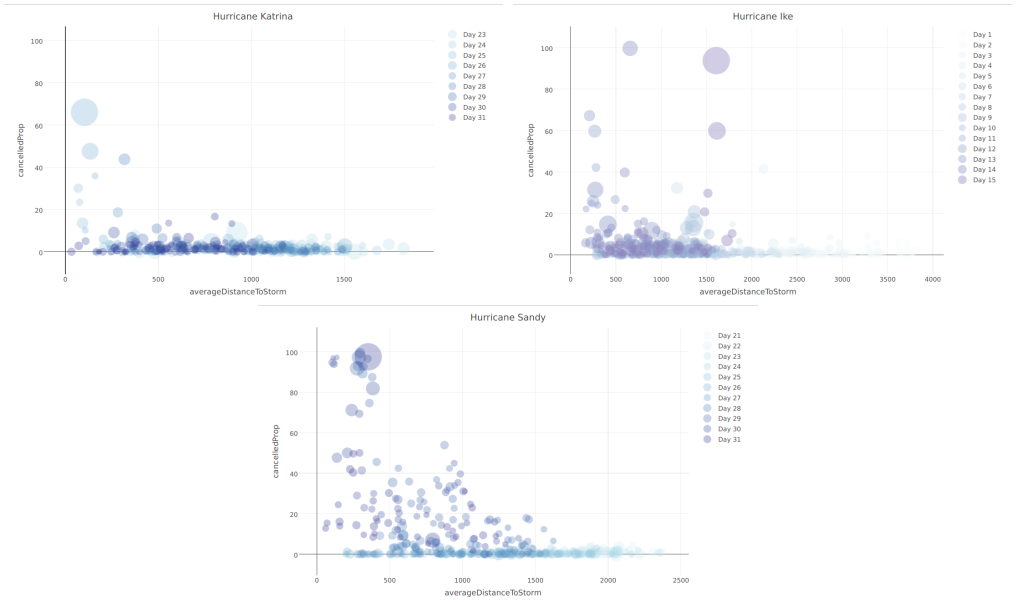

Figure 5: Bubble Plots of Katrina, Ike, Sandy

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. 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.

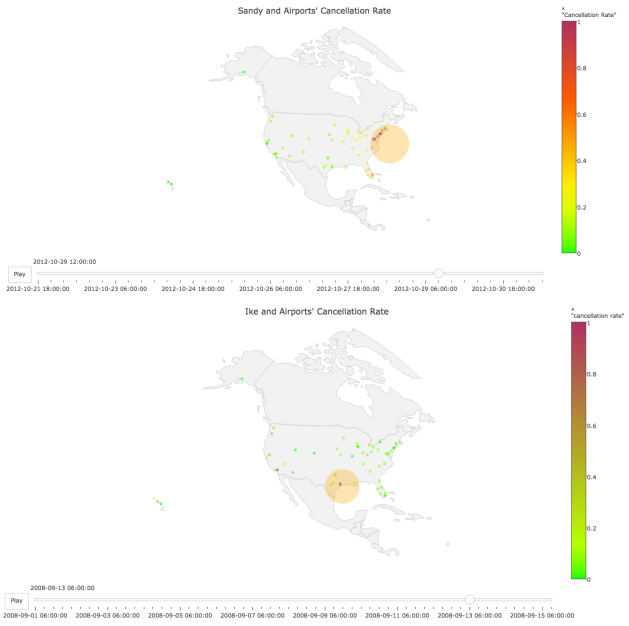

Figure 6: Animated Map of Sandy and Ike

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 since 2008.

When Sandy hits the eastern shores, we can see the 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.

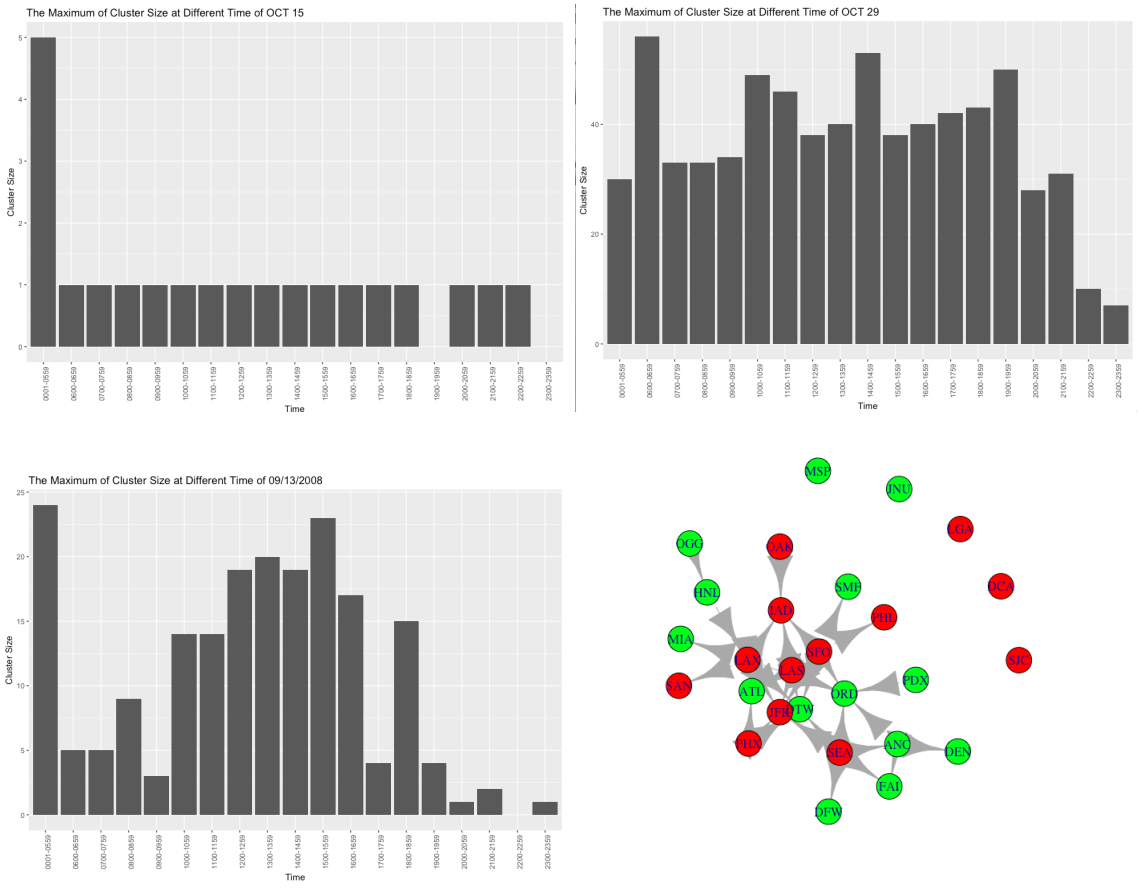

Figure 7: Top Left: Cancellation Cluster sizes on a regular day for Sandy. Top Right: Cancellation Cluster sizes on the day Sandy hit East Coast shores. Bottom Left: Cancellation Cluster sizes on the day Ike hit Central shores. Bottom Right: Igraph of Network Clusters

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 delayed 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%.

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. For example, we have read about the spatial and autoregressive models in determining the time lag effects and spatial effect of the dynamic hurricane. But learning about the knowledge in these analyses is beyond the scope of this project under the time constraint. We also didn't spend time tuning model parameters and doing cross-validation, since the bulk of our project was 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.

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-define project to work with.

7 Code Appendix

```
read_data_from_BTS = function(datadir){
  data = read.csv(datadir, stringsAsFactors = FALSE)
  colWanted = c("Year","Quarter","Month","DayofMonth","DayOfWeek","FlightDate",
                "UniqueCarrier","TailNum", "Origin","OriginCityName","OriginState",
                "Dest","DestCityName","DestState","DepTime","CRSDepTime","DepDelay",
                "DepDelayMinutes","DepDel15","DepTimeBlk","TaxiOut","WheelsOff","WheelsOn",
                "TaxiIn","CRSArrTime","ArrTime","ArrDelay","ArrDelayMinutes","ArrDel15",
                "ArrTimeBlk","CRSElapsedTime","ActualElapsedTime","Distance","DistanceGroup","Cancelled","CancellationCode",
                "CarrierDelay","WeatherDelay","NASDelay","SecurityDelay","LateAircraftDelay","Diverted")
  data = data[,colWanted]

  #  data = data[data$Origin %in% allAirports &
  #            data$Dest %in% allAirports,]
  data
}

#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`)
```

```
  levels(finaldf$`Origin Timezone`) = c("US/Hawaii","US/Eastern","US/Central",
    "US/Mountain","US/Pacific","US/Alaska")
  finaldf = finaldf[!is.na(finaldf$`Dest Timezone`) & !is.na(finaldf$`Origin
    Timezone`),]
  return(finaldf)
}

library(lubridate)
convertToUTC = function(row){
  row[2] = as.numeric(row[2])
  if(nchar(row[2]) == 3)
    row[2] = paste0("0",row[2])
  if(nchar(row[2]) == 2)
    row[2] = paste0("00",row[2])
  if(nchar(row[2]) == 1)
    row[2] = paste0("000",row[2])
  date = paste(row[1:2], collapse = " ")
  return(date)
}

getDistanceFromLatLonInKm = function (lat1,lon1,lat2,lon2) {
  R = 6371;
  dLat = deg2rad(lat2-lat1)
  dLon = deg2rad(lon2-lon1)
  a =
    sin(dLat/2) * sin(dLat/2) +
    cos(deg2rad(lat1)) * cos(deg2rad(lat2)) *
    sin(dLon/2) *sin(dLon/2)

  c = 2 * atan2(sqrt(a), sqrt(1-a))
  d = R * c * 0.621371
  return(d)
}

deg2rad = function (deg) {
  return(deg * (pi/180))
}

#Input DataFile
processStorm = function(dataFile) {
  stormdf = read_data_from_BTS(dataFile)

  #Identify Lon and Lat as well as timezone.
  df= getGpsLocation(stormdf)

  df$DepDateTime = apply(df[,c("FlightDate","CRSDepTime")], 1,convertToUTC)

  cols =  paste0("UTC",c("Year", "Month","DayofMonth",
                         "DayOfWeek","DepHour","CRSDepTime"))
```

```
  df[cols] = NA

  for(timezone in c("US/Hawaii","US/Eastern","US/Central",
                                                    "US/Mountain","US/Pacific","
    US/Alaska")){
    date = strptime(df$DepDateTime[df$`Origin Timezone` == timezone],
                    format="%Y-%m-%d %H:%M", tz = timezone)
    date = with_tz(date, tzone = 'UTC')
    newdf = data.frame(Year = year(date),Month = month(date), Day = day(date),
              DayOfWeek = weekdays(date), Hour = hour(date), Time = strftime(
    date, format="%Y-%m-%d %H:%M:%S"))
    newdf$Time = as.character(newdf$Time)
    newdf$DayOfWeek =as.character(newdf$DayOfWeek)
    df[df$`Origin Timezone` == timezone,cols] = newdf
  }

  #Put time in blocks
  utcTimes = strptime(df$UTCCRSDepTime, format="%Y-%m-%d %H:%M:%S", tz = 'UTC'
    )
  SixHourBlock = cut(utcTimes, breaks = "6 hour")
  df$DepSixHourBlock = SixHourBlock
  return(df)
}
```

Listing 1: Data Merging in R

```
#Animation Plot
plot_geo(locationmode = "USA-states") %>%
  add_markers(
    data = cancel_rate_ike_hubs, x = ~Lon, y = ~Lat, text = ~Name, frame = ~
    Group.2,
    hoverinfo = "text", alpha = 0.5, color = ~x, colors = c("green","yellow","
    orange","red","maroon")
  ) %>%
  add_markers(
    data = points_ike, x = ~LON, y = ~LAT, frame = ~ISO_time, alpha = 0.3,
    size = ~INTENSITY,
    color = ~'cancellation rate'
  ) %>%
  layout(
    title = paste("Ike and Airports' Cancellation Rate"), geo = geo,
    showlegend = FALSE
  ) %>%
  animation_opts(500, easing = "elastic",redraw = T
  ) %>%
  animation_slider(
    currentvalue = list(prefix = "Time Block ", font = list(color="Black"))
  )

#Clustering
```

```
cluster = sapply(1:19,function(i){
  routes[[i]]$Origin = factor(routes[[i]]$Origin)
  routes[[i]]$Dest = factor(routes[[i]]$Dest)
  airports = data.frame(name = delay_airport[[i]]$Group.1,congested = delay_
    airport[[i]]$congested)
  paths = data.frame(from = routes[[i]]$Origin, to = routes[[i]]$Dest)
  paths = paths[which(paths[,2] %in% airports$name),]
  paths = paths[which(paths[,1] %in% airports$name),]
  g = graph_from_data_frame(paths,vertices = airports)
  V(g)$color = ifelse(airports$congested == 1,"red","green")
  g = induced.subgraph(g,V(g)[V(g)$color %in% c("red")])
  plot(g)
  max(clusters(g)$csize)
})

p = ggplot(data = data15,aes(x = unique.cancel_rate_15.Group.2., y = cluster_
    15)) +
    geom_bar(stat = "identity") + ggtitle("The Maximum of Cluster Size at
    Different Time of OCT 15") +
    xlab("Time") + ylab("Cluster Size") + theme(axis.text.x = element_text(
    angle = 90, hjust = 1))
p
```

Listing 2: Animation Plot in R

```
#Modeling
extractFeatures = function(data){
  features = c("UniqueCarrier",
               "DepDistanceToStorm",
               "ArrDistanceToStorm",
               "UTCDayofMonth",
               "UTCDepHour",
               "Nature",
               "Wind.WMO.",
               "Pres.WMO.")
  fea = data[,features]
  fea$UniqueCarrier = as.factor(fea$UniqueCarrier)
  fea$Nature = as.factor(fea$Nature)
  return(fea)
}
set.seed(123)
modrf = randomForest(extractFeatures(train),as.factor(train$Cancelled),ntree =
    1000,importance = T)
test$predict = predict(modrf,extractFeatures(test))
```

Listing 3: Random Forest in R

Listings