I: Learning the Data

Preface:

We have a dataset about fights among the states of USA in 2009. We want to know what information it contians.

Steps:

1. What is the size of the dataset?

It contains 529,269 rows and 35 different features.

```
pd_flights.shape
✓ 0.5s

(529269, 35)
```

2. What attributes does this dataset have?

```
pd_flights.columns
✓ 0.5s

Index(['YEAR', 'MONTH', 'DAY_OF_MONTH', 'DAY_OF_WEEK', 'UNIQUE_CARRIER',
       'AIRLINE_ID', 'CARRIER', 'TAIL_NUM', 'FL_NUM', 'ORIGIN',
       'ORIGIN_CITY_NAME', 'ORIGIN_STATE_ABR', 'ORIGIN_STATE_NM', 'ORIGIN_WAC',
       'DEST', 'DEST_CITY_NAME', 'DEST_STATE_ABR', 'DEST_STATE_NM', 'DEST_WAC',
       'CRS_DEP_TIME', 'DEP_TIME', 'DEP_DELAY', 'CRS_ARR_TIME', 'ARR_TIME',
       'ARR_DELAY', 'CANCELLED', 'CANCELLATION_CODE', 'DIVERTED', 'AIR_TIME',
       'DISTANCE', 'CARRIER_DELAY', 'WEATHER_DELAY', 'NAS_DELAY',
       'SECURITY_DELAY', 'LATE_AIRCRAFT_DELAY'],
      dtype='object')
```

3. Understand the specific attributes.

a) TAIL_NUM and FL_NUM: TAIL_NUM is the number appears in the airport, every airplane has its unique tial number to distinguish with other planes. While flight number is not unique, different planes can have the same flight number, it is a specific code that an airline assigns to a particular flight in its network.

b) ORIGIN_WAC and DEST_WAC: A World Aeronautical Chart (WAC) is a type of aeronautical chart used for navigation by pilots of moderate speed aircraft and aircraft at high altitudes. WACs show topographic information, airports and radio navigational aids. They are useful for strategic flight planning, where a view of the entire flight area is useful.

c) CRS_DEP_TIME and CRS_ARR_TIME: represent as schedual departure time and schedual arrive time.

d) DIVERTED: Diverted flight means a flight which is operated from the scheduled origin point to a point other than the scheduled destination point in the carrier's published schedule. For example, a carrier has a published schedule for a flight

from A to B to C. If the carrier were to actually fly an A to C operation, the A to B segment is a diverted flight, and the B to C segment is a cancelled flight.

e) NAS_DELAY: Delay that is within the control of the National Airspace System (NAS) may include: non-extreme weather conditions, airport operations, heavy traffic volume, air traffic contro

f) CANCELLATION_CODE: can't find information, only know we have 4 tpyes of cancellation

```
pd_flights['CANCELLATION_CODE'].unique()
✓ 0.1s
array([nan, 'B', 'A', 'C', 'D'], dtype=object)
```

4. Types of data and missing data. As we can see below, the integrity of this dataset is good. Not many missing data.

g) Rows from 30-34 have so many missing data is because when the fligh is not delay, they will not fill the data, so we can simply fill with 0.

h) There are around 500 missing data in TAIL_NUM rows. We find out that when there are flight cancelled, there will be no TAIL_NUM.

i) There are 14730 flights cacelled.

j) DEP_TIME, DEP_DELAY; ARR_TIME,ARR_DELAY might due to canclled and other unknown reason for missing.

```
pd_flights.info()
✓ 0.3s
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 529269 entries, 0 to 529268
Data columns (total 35 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   YEAR                 529269 non-null  int64
 1   MONTH                529269 non-null  int64
 2   DAY_OF_MONTH         529269 non-null  int64
 3   DAY_OF_WEEK          529269 non-null  int64
 4   UNIQUE_CARRIER       529269 non-null  object
 5   AIRLINE_ID           529269 non-null  int64
 6   CARRIER              529269 non-null  object
 7   TAIL_NUM             524711 non-null  object
 8   FL_NUM               529269 non-null  int64
 9   ORIGIN               529269 non-null  object
 10  ORIGIN_CITY_NAME     529269 non-null  object
 11  ORIGIN_STATE_ABR     529269 non-null  object
 12  ORIGIN_STATE_NM      529269 non-null  object
 13  ORIGIN_WAC           529269 non-null  int64
 14  DEST                 529269 non-null  object
 15  DEST_CITY_NAME       529269 non-null  object
 16  DEST_STATE_ABR       529269 non-null  object
 17  DEST_STATE_NM        529269 non-null  object
 18  DEST_WAC             529269 non-null  int64
 19  CRS_DEP_TIME         529269 non-null  int64
 20  DEP_TIME             515076 non-null  float64
 21  DEP_DELAY            515076 non-null  float64
 22  CRS_ARR_TIME         529269 non-null  int64
 23  ARR_TIME             514091 non-null  float64
 24  ARR_DELAY            513051 non-null  float64
 25  CANCELLED            529269 non-null  int64
 26  CANCELLATION_CODE    14730 non-null   object
 27  DIVERTED             529269 non-null  int64
 28  AIR_TIME             513051 non-null  float64
 29  DISTANCE             529269 non-null  int64
 30  CARRIER_DELAY        132020 non-null  float64
 31  WEATHER_DELAY        132020 non-null  float64
 32  NAS_DELAY            132020 non-null  float64
 33  SECURITY_DELAY       132020 non-null  float64
 34  LATE_AIRCRAFT_DELAY  132020 non-null  float64
```

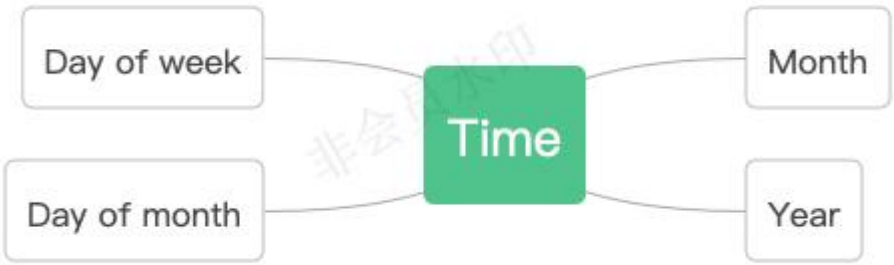
II: Come up with Meaningful Questions

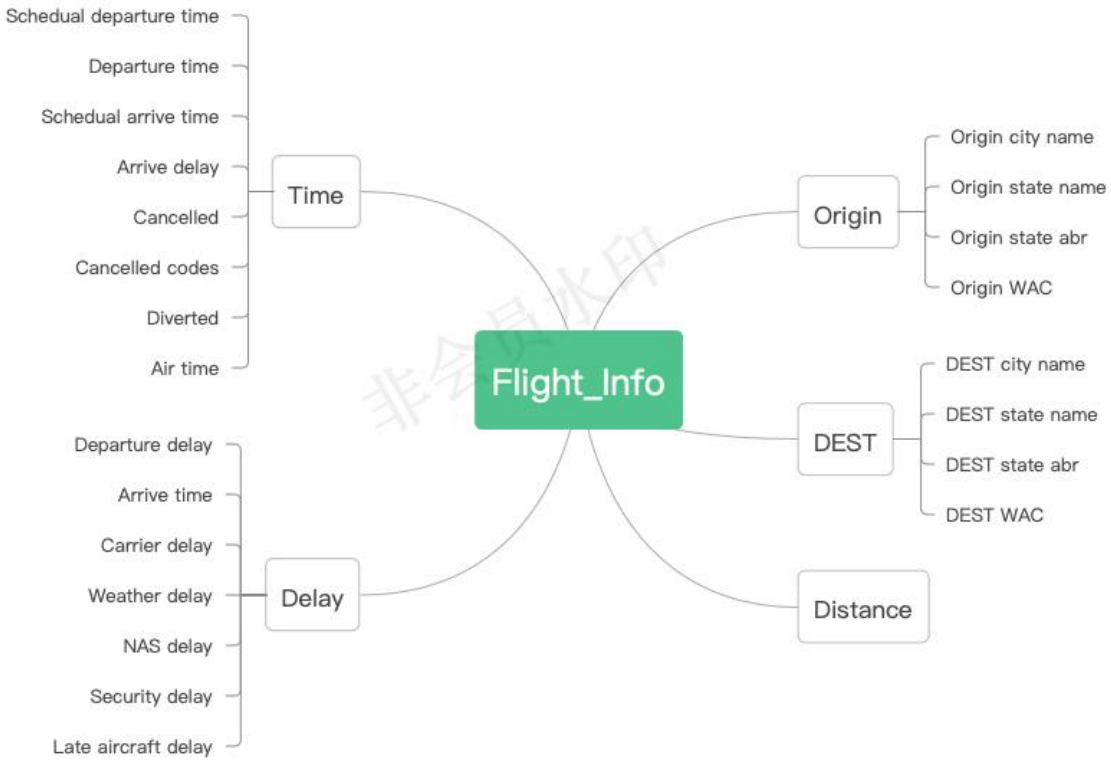
Preface:

In order to comp up with some meaningful quesitons, we would like to category these 35 features first and then find some possible interesting conncetions between the features.

Steps:

1. Category the attributes





2. Questions interesting in the begining :

a) In what day of week has the most flights?(show each day the number of flights)
b) How many carrier in USA and how many planes do they have?(use slider to show)
c) The percentage of each typt of delay, which is the key dealy?
d) Does ditance has relation with delay time? Like longer the distance the flight tend to delay?
e) Does diverted related to delay?
f) Show the flight during Christmas time. What are the most active cities? (map)
g) What carries love long distance flights, waht carreis have more short distance flights?
h) What states have most active flights(map)
i) More questions....

3. Finally decided question

a) What is the buiest day of the week?
b) What is the buiest carrier in US?
c) What are the top delay reasons?
d) What are the top delay reasons during christmas?
e) What are the different delay reasons of different carrier during Chirstmas holiday?
f) What are the different delay reasons of different states during Chirstmas holiday?
g) What are the most popular flights took of from New York in the December 25th in 2009?

III: Preprocessing and Transform Data

Preface:

In order to answer the questions and more easier to find the answers, we would like to proprocess the data and maybe transorm the data.

Steps:

1. Pie Graph data transform

a) What is the buiest day of the week? The create a empty list and sum up the times of each flights fly in the specific day and store in that list.

```
# Create a list to store the flights volume of each day of week
# Create a empty list
Day_of_Week_Volume = []
# Create a list to store the day
Day_of_Week = ['Mondy','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']
# for loop , use sum() to calculate the rows of each day
for i in range(1,8):
    Day_of_Week_Volume.append((df_flights['DAY_OF_WEEK'] == i).sum())
# combie tow list to form a pandas dataframe
# first create a dictionary to store two list
dictionary_day = {'Day_of_Week_Volume' : Day_of_Week_Volume, 'Day_of_Week': Day_of_Week}
df_propotion_day = pd.DataFrame(dictionary_day)
```

b) What is the buiest carrier in US? We create a empty list and sum up the times of each different carriers fly a plane and store in that list.

```
# Create a list to store the flights volume of carrier
# Create a empty list
Carrier_Volume = []
# Convert the
Carriers = (df_flights['UNIQUE_CARRIER'].unique()).tolist()
# for loop , use sum() to calculate the number of flights of each Carrier
for carrier in Carriers:
    Carrier_Volume.append((df_flights['UNIQUE_CARRIER'] == carrier).sum())
# combie tow list to form a pandas dataframe
# first create a dictionary to store two list
dictionary_Carriers = {'Carrier_Volume' : Carrier_Volume, 'Carriers': Carriers}
df_propotion_Carriers = pd.DataFrame(dictionary_Carriers)
```

c) What are the top delay reasons? First, we fill the empty entry to 0.

```
# fill the missing data
# We know that in the dalay columns, if the value in NaN, is because the flight didn't delay
# so we just simply fill it with 0
df_flights['CARRIER_DELAY'] = df_flights['CARRIER_DELAY'].fillna(0)
df_flights['WEATHER_DELAY'] = df_flights['WEATHER_DELAY'].fillna(0)
df_flights['NAS_DELAY'] = df_flights['NAS_DELAY'].fillna(0)
df_flights['SECURITY_DELAY'] = df_flights['SECURITY_DELAY'].fillna(0)
df_flights['LATE_AIRCRAFT_DELAY'] = df_flights['LATE_AIRCRAFT_DELAY'].fillna(0)
print()
```

Then we create a empty list and sum up the all the minutes it delay and store in the list.

```
# create a delay list
delay_type = ['CARRIER_DELAY','WEATHER_DELAY','NAS_DELAY','SECURITY_DELAY','LATE_AIRCRAFT_DELAY']
# initial a empty list to store the delay value
delay_volume = []
# Calculate the total delay time
for i in range(0,5):
    delay_volume.append((df_flights[delay_type[i]]).sum())
# create a dictionary to store two lists
dictionary_delay = {"delay_type" : delay_type, "delay_volume" : delay_volume}
# create a pandas dataframe
df_delay = pd.DataFrame(dictionary_delay)
print(delay_volume)
```

2. Animation Graph transform

d) What is the top delay reason during the christmas? We first need to select the day we want , in here is from December 22th to December 31th. And we sum up the minutes in each delay of that day and store in the list.

```
# build the suitable dataframe
# we want MONTH as animation_frame
df_delay_christmas = df_flights[['DAY_OF_MONTH','CARRIER_DELAY','WEATHER_DELAY','NAS_DELAY','SECURITY_DELAY','LATE_AIRCRAFT_DELAY']]
# create empty list to store the value
day_list = [22,23,24,25,26,27,28,29,30,31]
carrier_delay_list, weather_delay_list, nas_delay_list, security_delay_list, late_aircraft_delay_list = [],[],[],[],[]
# for each month, sum up all delay time in each type of delay
for day in range(22,32):
  carrier_delay_list.append((df_delay_christmas[df_delay_christmas['DAY_OF_MONTH'] == day]['CARRIER_DELAY']).sum())
  weather_delay_list.append((df_delay_christmas[df_delay_christmas['DAY_OF_MONTH'] == day]['WEATHER_DELAY']).sum())
  nas_delay_list.append((df_delay_christmas[df_delay_christmas['DAY_OF_MONTH'] == day]['NAS_DELAY']).sum())
  security_delay_list.append((df_delay_christmas[df_delay_christmas['DAY_OF_MONTH'] == day]['SECURITY_DELAY']).sum())
  late_aircraft_delay_list.append((df_delay_christmas[df_delay_christmas['DAY_OF_MONTH'] == day]['LATE_AIRCRAFT_DELAY']).sum())
# create dictionary
dictionary_delay_christmas = {'day':day_list, 'carrier_delay':carrier_delay_list, 'weather_delay':weather_delay_list,
                              'nas_delay':nas_delay_list,'security_delay':security_delay_list,'late_aircraft_delay':late_aircraft_delay}
# create pandas dataframe
df_delay_christmas_plot = pd.DataFrame(dictionary_delay_christmas)
```

And then we use melt function to convert the horizontal dataframe to a vertical dataframe in order to plot.

```
# change the horizontal df to vertical df
df_delay_christmas = pd.melt(df_delay_christmas_plot, id_vars=['day'], value_vars=['carrier_delay','weather_delay',
'nas_delay','security_delay','late_aircraft_delay'])
# df_delay_christmas
```

e) What is the different delay reason of different carrier during the christmas? Is the same as above, the difference is we select the data with different carrier.

```
# build the suitable dataframe
# we want MONTH as animation_frame
df_delay_christmas_carrier1= df_flights[['DAY_OF_MONTH','CARRIER_DELAY','WEATHER_DELAY','NAS_DELAY','SECURITY_DELAY','LATE_AIRCRAFT_DELAY','UNIQUE_CARRIER']]
# create empty list to store the value
day_list = [22,23,24,25,26,27,28,29,30,31]
carrier_list = (df_flights['UNIQUE_CARRIER'].unique()).tolist()
day_list_merge = []
carrier_list_merge = []
carrier_delay_list, weather_delay_list, nas_delay_list, security_delay_list, late_aircraft_delay_list = [],[],[],[],[]
# for each month, sum up all delay time in each type of delay
for day in range(22,32):
    for carrier in carrier_list:
        day_list_merge.append(day)
        carrier_list_merge.append(carrier)
        carrier_delay_list.append((df_delay_christmas_carrier1[(df_delay_christmas_carrier1['DAY_OF_MONTH'] == day) & (df_delay_christmas_carrier1['UNIQUE_CARRIER'] == carrier) ]['CARRIER_DELAY']).sum())
        weather_delay_list.append((df_delay_christmas_carrier1[(df_delay_christmas_carrier1['DAY_OF_MONTH'] == day) & (df_delay_christmas_carrier1['UNIQUE_CARRIER'] == carrier) ]['WEATHER_DELAY']).sum())
        nas_delay_list.append((df_delay_christmas_carrier1[(df_delay_christmas_carrier1['DAY_OF_MONTH'] == day) & (df_delay_christmas_carrier1['UNIQUE_CARRIER'] == carrier) ]['NAS_DELAY']).sum())
        security_delay_list.append((df_delay_christmas_carrier1[(df_delay_christmas_carrier1['DAY_OF_MONTH'] == day) & (df_delay_christmas_carrier1['UNIQUE_CARRIER'] == carrier) ]['SECURITY_DELAY']).sum())
        late_aircraft_delay_list.append((df_delay_christmas_carrier1[(df_delay_christmas_carrier1['DAY_OF_MONTH'] == day) & (df_delay_christmas_carrier1['UNIQUE_CARRIER'] == carrier)]['LATE_AIRCRAFT_DELAY']).sum())

# create dictionary
dictionary_delay_christmas_carrier1 = {'day':day_list_merge, 'carrier':carrier_list_merge, 'carrier_delay':carrier_delay_list, 'weather_delay':weather_delay_list,
                              'nas_delay':nas_delay_list,'security_delay':security_delay_list,'late_aircraft_delay':late_aircraft_delay_list}
# create pandas dataframe
df_delay_christmas_carrier = pd.DataFrame(dictionary_delay_christmas_carrier1)

# melt the dataframe
# change the horizontal df to vertical df
df_delay_christmas_carrier_plot = pd.melt(df_delay_christmas_carrier, id_vars=['day','carrier'], value_vars=['carrier_delay','weather_delay','nas_delay','security_delay','late_aircraft_delay'])
```

f) What are the different delay reason in each state during the christmas? The same as above, just slect the data with different state.

```
# build the suitable dataframe
# we want MONTH as animation_frame
df_delay_christmas_state1 = df_flights[['DAY_OF_MONTH','CARRIER_DELAY','WEATHER_DELAY','NAS_DELAY','SECURITY_DELAY','LATE_AIRCRAFT_DELAY','ORIGIN_STATE_NM']]
# create empty list to store the value
day_list = [22,23,24,25,26,27,28,29,30,31]
state_list = (df_flights['ORIGIN_STATE_NM'].unique()).tolist()
day_list_merge = []
state_list_merge = []
carrier_delay_list, weather_delay_list, nas_delay_list, security_delay_list, late_aircraft_delay_list = [],[],[],[],[]
# for each month, sum up all delay time in each type of delay
for day in range(22,32):
    for state in state_list:
        day_list_merge.append(day)
        state_list_merge.append(state)
        carrier_delay_list.append((df_delay_christmas_state1[(df_delay_christmas_state1['DAY_OF_MONTH'] == day) & (df_delay_christmas_state1['ORIGIN_STATE_NM'] == state) ]['CARRIER_DELAY']).sum())
        weather_delay_list.append((df_delay_christmas_state1[(df_delay_christmas_state1['DAY_OF_MONTH'] == day) & (df_delay_christmas_state1['ORIGIN_STATE_NM'] == state) ]['WEATHER_DELAY']).sum())
        nas_delay_list.append((df_delay_christmas_state1[(df_delay_christmas_state1['DAY_OF_MONTH'] == day) & (df_delay_christmas_state1['ORIGIN_STATE_NM'] == state)]['NAS_DELAY']).sum())
        security_delay_list.append((df_delay_christmas_state1[(df_delay_christmas_state1['DAY_OF_MONTH'] == day) & (df_delay_christmas_state1['ORIGIN_STATE_NM'] == state)]['SECURITY_DELAY']).sum())
        late_aircraft_delay_list.append((df_delay_christmas_state1[(df_delay_christmas_state1['DAY_OF_MONTH'] == day) & (df_delay_christmas_state1['ORIGIN_STATE_NM'] == state)]['LATE_AIRCRAFT_DELAY']).sum())

# create dictionary
dictionary_delay_christmas_state1 = {'day':day_list_merge, 'state':state_list_merge, 'carrier_delay':carrier_delay_list, 'weather_delay':weather_delay_list,
                                     'nas_delay':nas_delay_list,'security_delay':security_delay_list,'late_aircraft_delay':late_aircraft_delay_list}
# create pandas dataframe
df_delay_christmas_state = pd.DataFrame(dictionary_delay_christmas_state1)

# melt the dataframe
# change the horizontal df to vertical df
df_delay_christmas_state_plot = pd.melt(df_delay_christmas_state, id_vars=['day','state'], value_vars=['carrier_delay','weather_delay','nas_delay','security_delay','late_aircraft_delay'])
```

3. Cartographic Graph transform

g) What are the most popular flights took of from New York in the December 25th in 2009?

We search the longtitude and latitude of difference state in US and store them in the list.

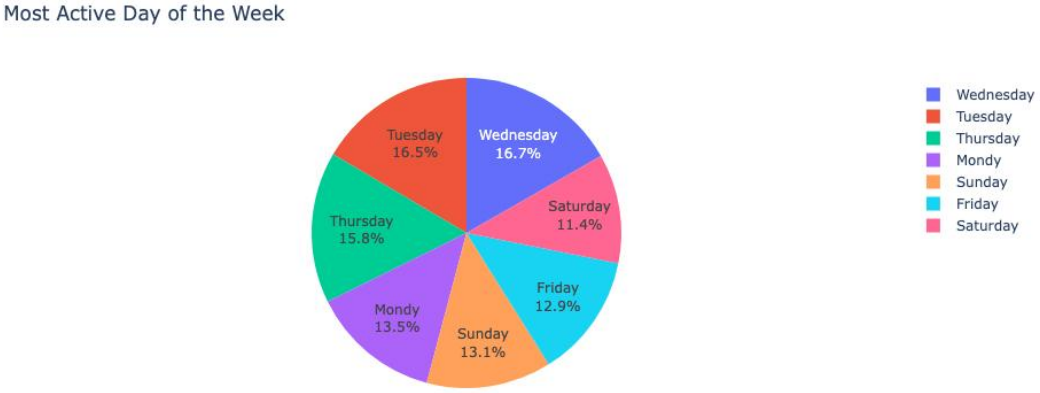
```
# create a list of longtitude and latitude of the state in the US
state_list = ['North Carolina', 'Virginia', 'Georgia', 'Mississippi',
       'Louisiana', 'Missouri', 'Wisconsin',
       'New York', 'Tennessee', 'Texas', 'Pennsylvania', 'South Carolina',
       'Kentucky', 'Ohio', 'Michigan', 'Illinois', 'New Jersey', 'Maine',
       'Oklahoma', 'Rhode Island', 'Florida', 'Minnesota', 'Nebraska',
       'Iowa', 'Kansas', 'Alabama', 'South Dakota', 'Arkansas', 'Indiana',
       'Vermont', 'North Dakota', 'Massachusetts', 'New Hampshire',
       'Connecticut', 'California', 'Nevada',
       'Washington', 'Utah', 'Colorado', 'Arizona', 'New Mexico',
       'Oregon', 'Maryland', 'Wyoming', 'Alaska',
       'Idaho', 'Montana', 'West Virginia', 'Hawaii']
state_latitude_list = [35.782169, 37.926868, 33.247875, 33.000000,
30.391830, 38.573936, 44.500000,
43.000000, 35.860119, 31.000000, 41.203323, 33.836082,
37.839333, 40.367474, 44.182205, 40.000000, 39.833851, 45.367584,
36.084621, 41.742325, 27.994402, 46.392410, 41.500000,
42.032974, 38.500000, 32.318230, 44.500000, 34.799999, 40.273502,
44.000000, 47.650589, 42.407211, 44.000000,
41.599998, 36.778259, 39.876019,
47.751076, 39.419220, 39.113014, 34.048927, 34.307144,
44.000000, 39.045753, 43.075970, 66.160507,
44.068203, 46.965260, 39.000000, 19.741755]
state_longitude_list = [-80.793457, -78.024902, -83.441162, -90.000000,
-92.329102, -92.603760, -89.500000,
-75.000000, -86.660156, -100.000000, -77.194527, -81.163727,
-84.270020, -82.996216, -84.506836, -89.000000, -74.871826, -68.972168,
-96.921387, -71.742332, -81.760254, -94.636230, -100.000000,
-93.581543, -98.000000, -86.902298, -100.000000, -92.199997, -86.126976,
-72.699997, -100.437012, -71.382439, -71.500000,
-72.699997, -119.417931, -117.224121,
-120.740135, -111.950684, -105.358887, -111.093735, -106.018066,
-120.500000, -76.641273, -107.290283, -153.369141,
-114.742043, -109.533691, -80.500000, -155.844437]
```

0.1s

IV: Visualiazation and Find Relations

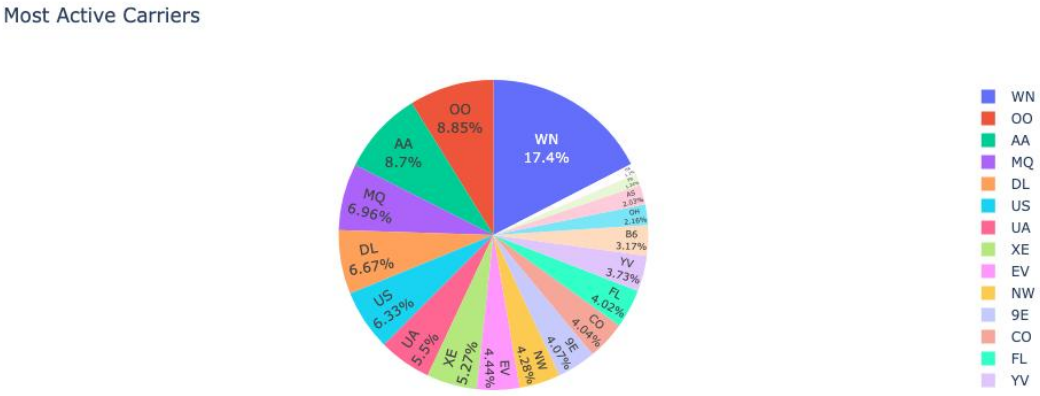
1. What is the buiest day of the week?

Wednesday is the busiest day, Saturday is the least busy day.
Tusday and Thusday are busier than Monday and Sunday, more than 2 percent.



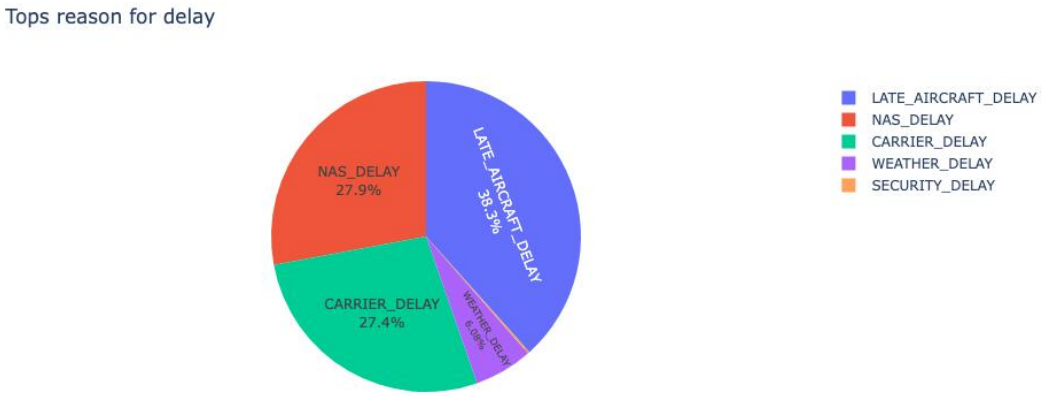
2. What is the buiest carrier in US?

WN(Southwest Airline) is the most active, occupy 17.4% of the flights.
OO(Skywest Airline) and AA(American Airline) cccupy around 9% of the flights
MQ(Envoy Air), DL(Delta Airline), US(can't find) occupy around 7% of the flights
UA(United Airline) and XE(can't find) occupy around 5%
Other compnies are less than 5%



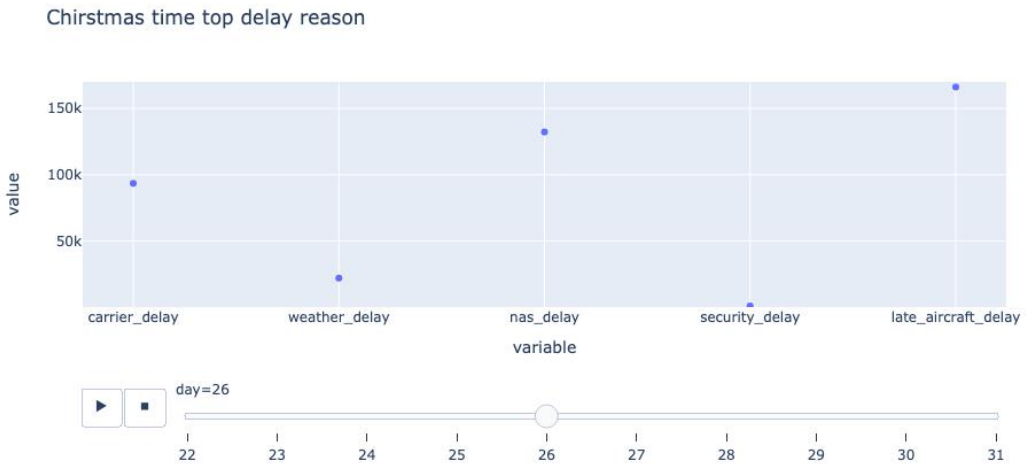
3. What are the top delay reasons?

The late aircraft delay is most significant reason of delay, occupy around 38%.
NAS delay and carrier delay are ohter 2 main reason of delay, around 27%.
weather delay sometime happen, and security delay is very rare,



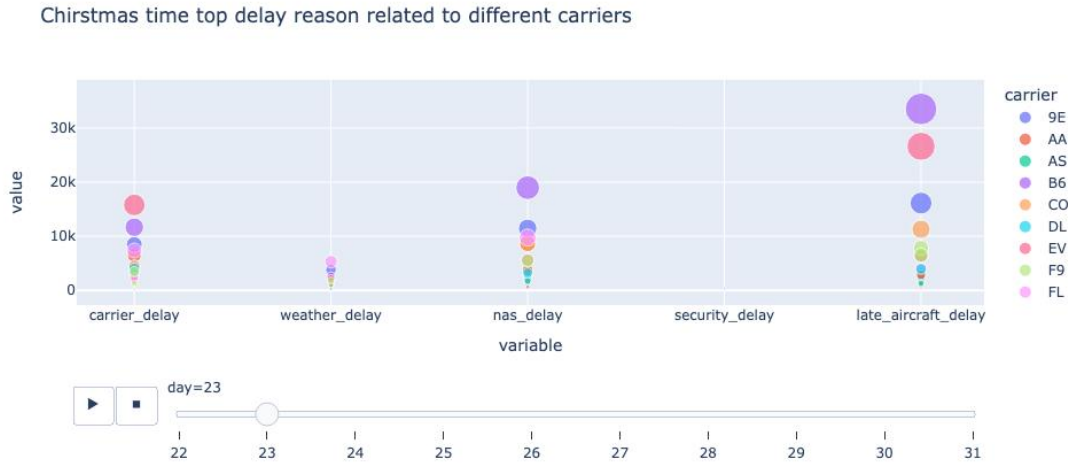
4. What are the top delay reasons during christmas?

During the animation, you can see the top delay reason of each day.
The late aircraft delay is the main reason for most of the time.



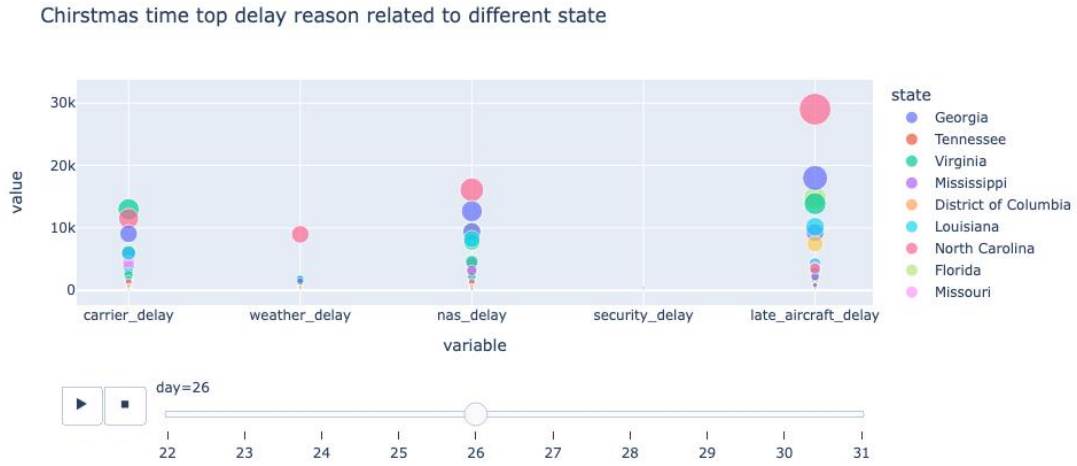
5. What are the different delay reasons of different carrier during Chirstmas holiday?

You can see the different carrier's different delay reason during the christmas holiday.
In this frame, the 9E carrier is high in late aircraft delay and nas dela. EV is high in carrier delay.



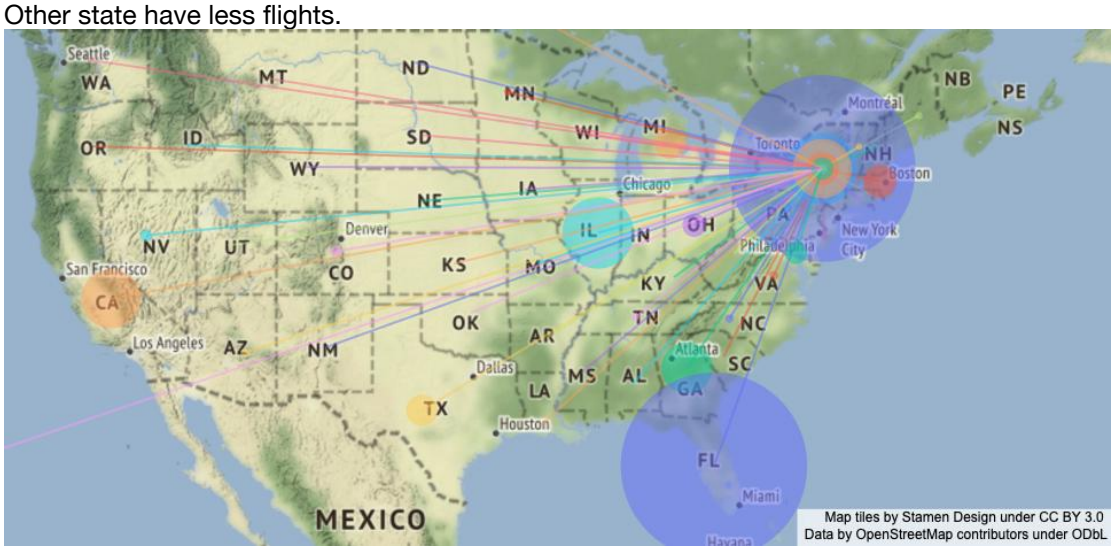
6. What are the different delay reasons of different states during Chirstmas holiday?

You can see the different state's different delay reason during the christmas.
In this frame, The Tennessee has high delay in all the category delay excpet security delay.
Virginia have high delay in carrier delay, middle delay in late aircraft delay.



7. What are the most popular flights took of from New York in the December 25th in 2009?

In this picture you can see the flight took off from New York in Christmas. The bigger the circle the more frequent the flight.
In Christmas, There are many flights fly to other cities of New York and FL. Also quit amount of flights fly to CA, TX, IL,MI and MA.
Other state have less flights.



V: Summary and Challenges

Challenges:

1. Some data can't interperate. Like WAC and Cancelled Codes. If we can understand the meaning behind the data, we might found more relation.
2. If we have more tables of data about the flight of USA in 2009, we certainly can gain more interesting information. For example, if we have a corresponding table about the ticket price and customs satisfaction. We might can help carriers to sale more tickets.
3. Didn't implement the search bar. It turns out the more common way to implement the search bar is using Dash. If we want to use python to implement the search bar, the method is use figurewidege. We will try next time.

Summary:

In this assignemnt , we dig into the database of flight of USA in December 2009. First we use some basic function in the pandas to have a genereal look at the dataset, and then we plot some 2D pie plot to find out what is the buiest day of the week, what is the buiest carrier in the US and what is the top reason to delay. And the delay reason is something we really interesting in. So we decide to dig more detail in the reason of delay. We use animation to plot the different carrier's different delay reason during christmas holiday. And also plot the different state's differernt delay reason during christmas holiday. Unfortunately, you can't interact with the plot in the pdf file, you need to import the Flight dataset and run the code to interact with the animation plot. Fianlly, we think New York is a city with lost of people who comes from other states. So we plot a map plot to show in the December 25th, the flights took off from New York.