To Airlines: Recruiting Employees may Make Money by Reducing Delay

Siwei Tang* Zeyuan Li* Shuyan Li*
Statistics Department, University of Michigan
{tangsw, zeyuanli, lishuyan}@umich.edu

February 2nd, 2023

1 Executive Summary

In 2021, US airlines' cost of delay has reached $8.3 billion while the passengers' cost of delay is over $18 billion. JetBlue Airways, one of the major discount airlines in the US, has performed throughout the 6 years worse than the rest of the airlines from 2016 - 2021.(Figure 6) As the founder of JetBlue, David Neeleman, once said that he wants JetBlue to be a customer service company that provides flights. The more delay one has, the less likely the customer is going to stay loyal to the such airline. If this situation proceeds to develop, JetBlue will potentially lose customers and reputation.

As we know, weather and regulatory delays are not under the control of airline companies. Thus, in this report, we try to give a feasible solution to reducing the delays that a company can proactively take action for (e.g. airline delays caused by long cleaning time, luggage off-loading time, or crew shortage). All these can be improved by increasing the labor force or enhancing employees' working efficiency after airplane takes off.

Our cost-benefit analysis shows that 1% of an increase in an employee's salary would justify 1.4% of the benefits resulting from a better on-time rate, provided by certain conditions. This will break even the cost and benefit. It's worth noticing that there are extra perks when airlines manage to decrease their delay ratio while offering employees bonuses. On the one hand, it can retain its customer base and build loyalty from the customer with fewer delay occasions. On the other hand, it can also create a better working culture, because workers don't need to extend their work time due to flight delays, and they will receive higher income after they help the company reach its target.

Further validation can be done using random sampling on the same airport during different dates in a week so that we can use A/B testing to verify the efficacy of raising bonuses to the maintenance crew. Thus, based on the test result, we can extrapolate to other airports and airline companies.

It's worth noticing that the same situation of high delay rate happened to our competitor Continental Airlines years ago. They address it by increasing bonuses once the delay rate is decreased to a certain target. When Gordon Bethune took the office of Continental Airlines as a CEO, the airline is ranked the last in on-time arrival rate. Once in an interview, he admitted that the high delay rate caused the company a severe loss of 6 million dollars per month. After giving 3 million bonuses to employees such as flight attendants, technicians, mechanics, and maintenance staff, Gordon successfully overturned the losing streak and Continental rises up to the 5th in terms of the on-time arrival rate. And the company was later acquired by United Airlines.

Besides, to provide bonuses to its employees, JetBlue can change airlines' schedules by cutting schedules is another approach. This is what JetBlue was deciding to do since April 2022, which play coincides with us. We suggest cutting 37 routes including their most important routes between NYC and South Florida. NYC and Florida are ranked the top two hubs in JetBlue's portfolio, in both in-flow and out-flow traffic. (Figure 1)

2 Non-Technical Solution Summary

Flight delays and cancellations are unavoidable issues for airline companies, which will cause customer loss and negative growth in revenue. The business question we try to address is how an airline company (JetBlue) can decrease its delay rate in order to avoid unnecessary expenditures such as delay compensation plus customers' potential loss.

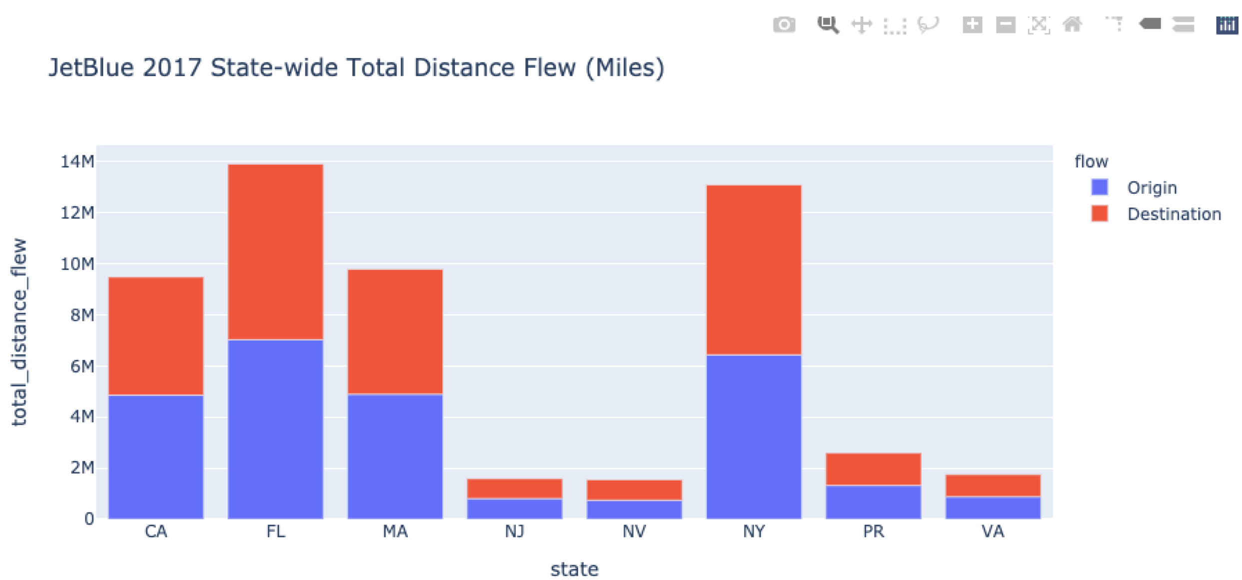

Figure 1: Origin Total Distance Flew

We find that Florida, New York, and Massachusetts are the three major states in terms of total distance flown. Because the load factor of JetBlue 2017 is 84%. Multiplying the total distance, and flight seats (roughly 200) will give the RPM of that state. Furthermore, if times by load factor 84%, this will give us the RPM of that state. Florida is undoubtedly the highest one as origin and destination at the same time. And most of the states are consistent for distance flown both as an original state and as a destination state

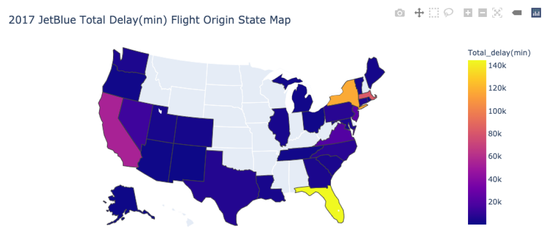

Figure 2: Total Delay Minutes Origin

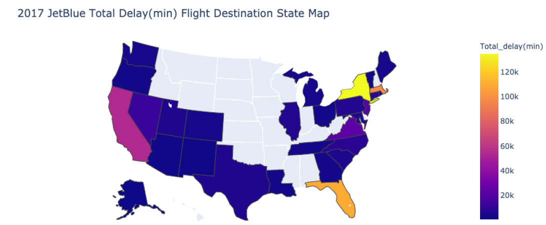

Figure 3: Total Delay Minutes Destination

However, this consistency has no longer existed when we examine the delay times and the delay per flight. Figure 2 has Florida as the highest delayed minutes departing state, while New York claim the lead in arriving state. As Florida turns from yeloow to orange, this shows its rank is decrease, and NY vice versa.

Combining the month level result in Figure 6 7 8, we can see from April to September, the entire industry including JetBlue would witness a delay rate spike. This is probably due to the travel season with leisure passengers and an increase of extreme weather such as flooding and hurricanes in the east coast every summer.

This is particularly important if we considering taking the second approach – cutting schedule. We might consider which state have the highest delay minutes per flight. Figure4 5 shows that NY and FL decrease its rank with respect to average delay minutes per flight. Two states outstands others, one is Colorado in origin which has 30.5 delay minutes per flight, and Hawaii in destination which has 31.3 delay minutes per flight. Judging from the data, if JetBlue wants to lower its average delay time for the company, it can consider cut schedule in HI and CO.

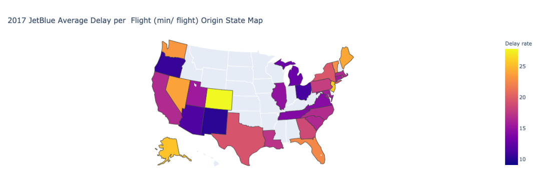

Figure 4: Total Delay Minutes Origin

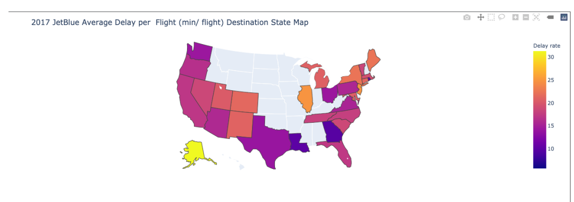

Figure 5: Total Delay Minutes Destination

The provided data set “4. Flight Delay 2016-2021.csv” includes the year of the data, the month of the data, the Abbreviation of the airline, the name of the airline, the number of flights arriving at the airport, the number of flights delayed due to air carrier, the number of flights delayed due to air carrier, the number of flights delayed due to National Aviation System, the number of flights canceled due to a security breach, the number of flights delayed as a result of another flight on the same aircraft delayed, number of canceled flights, and etc. We utilize the previous data set and classify delays caused by air carriers as controllable delays. And delays caused by weather, National Aviation System, security breaches, or another flight on the same aircraft delayed are classified as uncontrollable delays. Then we also sum up the number of canceled flights and all delays and name it as total delay and cancel flight number.

Next, we group the above data by year and month and get the sum of controllable delays, uncontrollable delays, total delayed and canceled flights, and the number of flights arriving at the airport in each month of each year. We can divide the above first three numbers by the sum of the number of flights arriving at the airport to get the controllable delay rate, uncontrollable delay rate, and total delay and cancel rate.

We find that JetBlue company's performance is significantly worse compared to the average performance of all airline companies. As we can see from the following Figure 6 7 8, no matter in which year or month, JetBlue's controllable delay rate, uncontrollable delay rate, and total delay and cancel rate are all above the average.

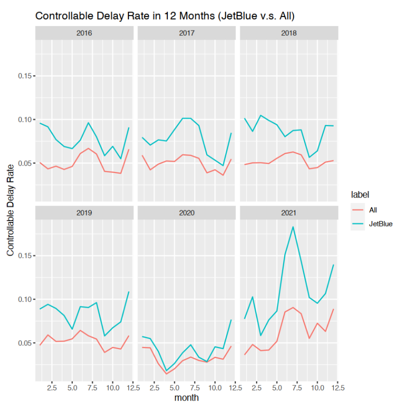

Figure 6: Controllable Delay Rate (JetBlue v.s. All)

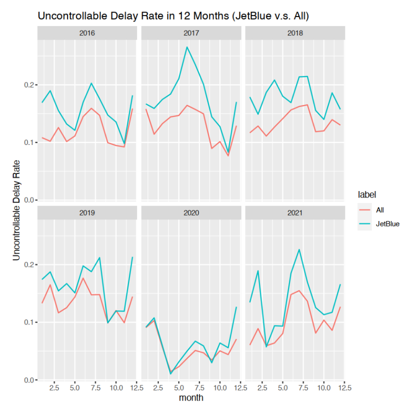

Figure 7: Uncontrollable Delay Rate (JetBlue v.s. All)

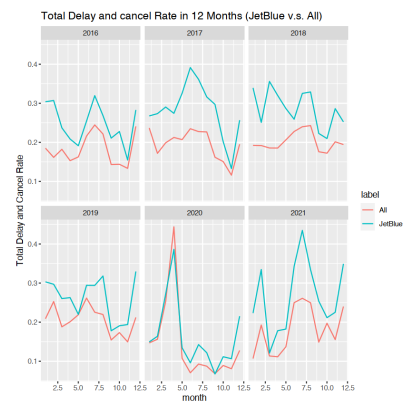

Figure 8: Total Delay and Cancel Rate (JetBlue v.s. All)

We next use data "7. Flight_Traffic_2017.csv" for detailed analysis of delay reasons of JetBlue Airline. This data includes the scheduled departure time, actual departure time, scheduled landing time, actual landing time, and the delay time for each of the five reasons for each flight for each day in 2017.

As mentioned before, JetBlue has a higher delay rate than the whole airline industry. We would like to analyze which specific delay reason accounts for a larger percentage. We get the pie plot in Figure 9

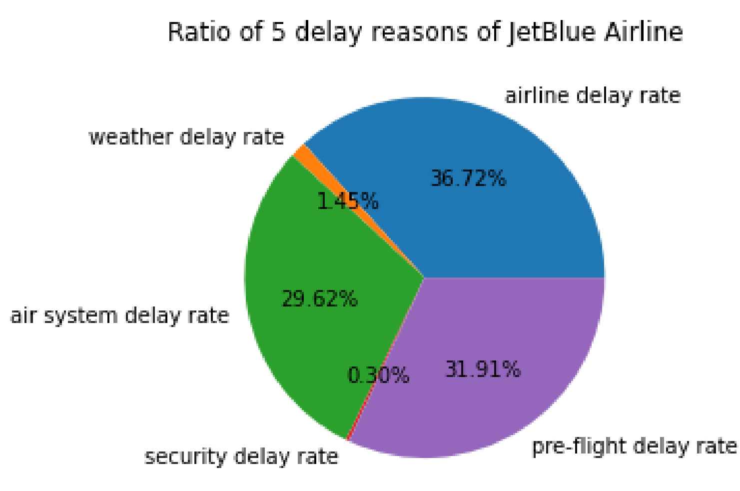

Figure 9: Ratio of 5 delay reasons of JetBlue Airline

We could conclude that airline delays, air system delays, and pre-flight delays are the three main reasons contributing to all delays, while the proportion of weather delays and security delays is very small. Notice that the cause of pre-flight delay must be caused by the other 4 causes, so if we want to mitigate the delays, then as soon as we mitigate the first 4 reasons for delays, we will naturally mitigate the pre-flight delay. So our research focus is methods to reduce airline delays and air system delays.

The reasons for the occurrence of airline delays tend to focus on 1. lack of ground crew 2. lack of crew members such as flight attendants 3. aircraft cleaning time is too long. We make a reasonable assumption that all of these reasons can be mitigated by hiring more employees. Recruiting more employees means paying more in wages, but the airlines can reduce the delay rate by a certain percentage, let's say x, then the company maybe benefit from the reduced delay rate. Our logic is shown in Figure 10.

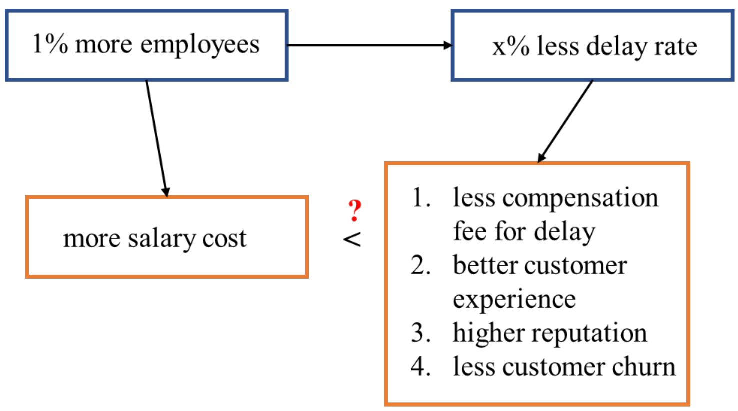

Figure 10: Ratio of 5 delay reasons of JetBlue Airline

The airlines can benefit from the reduced delay rate by 4 parts. The first obvious benefit is that the reduced delay rate means that the airline will pay less compensation fee to passengers for being late. Besides, there are three implicit benefits like better customer service, higher reputation for airlines and less customer churn.

We want to compute the additional salary cost and the reduced compensation fee, and compare what value of x can the airline make profits. Through publicly available external data, we estimate that hiring 1% more employees will cost $595,812 per month. What's more, assume that the loss of delay, including the compensation fee and implicit costs, is 100$/per passenger, then we get reduced monthly cost is 42400820 * x. In conclusion, if x is over 1.4%, then hiring more employees will make a profit.

3 Technical Exposition

We next use data "7. Flight_Traffic_2017.csv" for detailed analysis of delay reasons of JetBlue Airline. We use the formula

$$\text{delay\_rate\_of\_one\_reason} = \frac{\text{count\_of\_delay\_of\_one\_reason}}{\text{count\_of\_total\_flights}}$$

to compute the delay rate of one specific reason. After regularizing the rate to 1, we get the pie plot in Figure 9.

So our research focus is methods to reduce airline delays and air system delays. We make an assumption that hiring more employees can reduce airline delays. We say hiring 1% more staff will lead to x% reduce of delay rate.

We want to compute the additional salary cost and the reduced compensation fee, and compare what value of x can the airline make profits. Both of the two numbers are computed monthly through the following formula:

$$\begin{cases} \text{salary cost} = \text{hourly salary} * 8\,\text{hours per day} * 30\,\text{days} * 1\% * \text{origin employees} \\ \text{reduced compensation fee} = \text{seats per aircraft} * \text{loadfactor} * \text{compensation fee per person} * \\ \quad \text{flights per month} * x\% \end{cases}$$

Firstly, we compute salary cost. JetBlue airline just needs to hire more flight attendants, airport operations personnel, and technicians, but there is no need to hire more pilots. In Figure 11, it shows the number of employees of 6 kinds of jobs in JetBlue. Suppose JetBlue hire 1% more employees of flight attendants, airport operations personnel, and technicians, then JetBlue will hire additional 44 flight attendants, 36 airport operations personnel and 7 technicians.

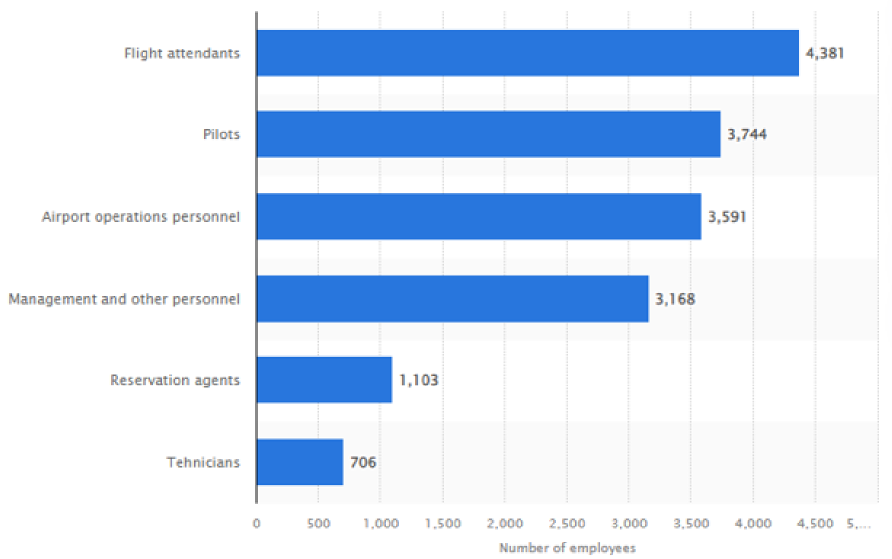

Figure 11: Number of JetBlue full-time equivalent employees in FY 2021. Source: `https://www.statista.com/statistics/529590/jetblue-number-of-employees-profession-breakdown/`

We also get the annual salary of these three jobs from Indeed (source: `https://www.indeed.com/cmp/Jetblue/salaries/Aircraft-Maintenance-Technician#:~:text=Average%20JetBlue%20Aircraft%20Maintenance%20T` echnician,43%25%20above%20the%20national%20average), and use it to compute the hourly salary. We finally got the hourly income in the following:

$$\begin{cases} \text{Flight attendant} = 25\$/\text{hour} \\ \text{airport operations personnel} = 40.89\$/\text{hour} \\ \text{technician} = 31.15\$/\text{hour} \end{cases}$$

With all the statistics above, we get the salary cost of 1% additional employees are

$$\text{salary cost} : (4381 * 25 + 3591 * 31.15 + 706 * 38.06) * 1\% * 8h * 30\text{days} = 595,812\$/\text{month}$$

Secondly, we compute the monthly reduced compensation fee. We make an assumption that each aircraft has fixed 200 seats. We get the load factor in 2017 as 0.847 from the Bureau of Transportation Statistics(source:

https://www.transtats.bts.gov/carriers.asp?20=E). The flight per month can be calculated from "7. Flight_Traffic_2017.csv" as 2503. The last number we need to estimate is the average compensation fee per passenger.

The compensation policy is shown in Figure 12 and the count of delayed flights in each bin is shown in Figure 13.

Delay time:	**Compensation amount:**
• 3 - 4:59 hours:	$100 credit
• 5 - 5:59 hours:	$175 credit
• 6 or more hours:	$250 credit

Figure 12: Compensation Policy of JetBlue

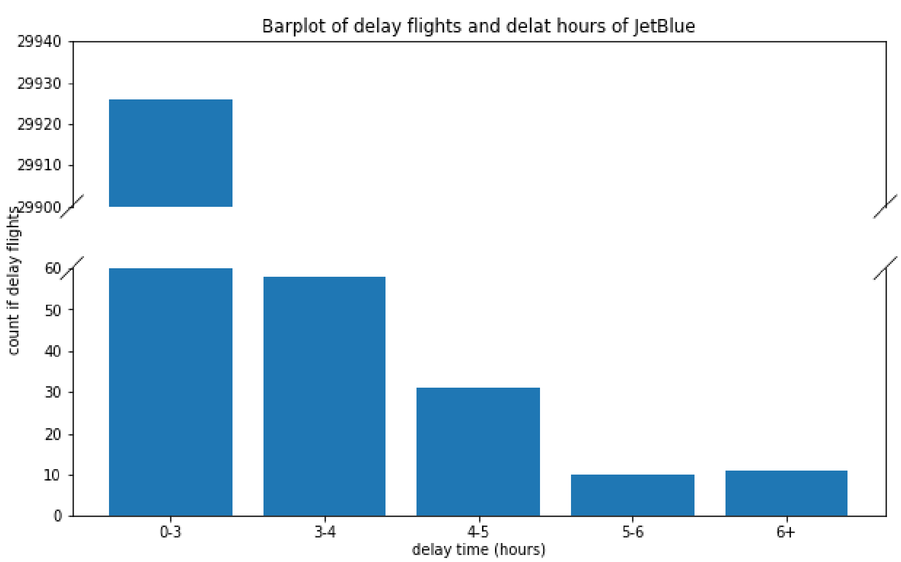

Figure 13: Barplot of delay flights and delay hours of JetBlue

Thus we get the compensation fee per passenger as

$$\text{average fee} = \frac{29926 * 0 + 451 * 50 + 146 * 125 + 110 * 200}{29926 + 451 + 146 + 110} = 2.09\$/\text{perperson}$$

This number is surprisingly small, but it's reasonable because more than 99.99% delays are within 3 hours and JetBlue will not pay compensation fee for this case. Finally, we get the reduced monthly compensation fee as 886526*x. If we want the reduced monthly compensation fee to cover the increased salary, then x should be over 67%.

A 1% increase in employees requires a 67% reduction in delay rate to be profitable, which is too unrealistic. However, The calculation only takes into account the reduction in explicit compensation fees, but does not consider the implicit benefits to the company from the reduced delay rate, like better customer service, higher reputation for airlines and less customer churn. In other words, the loss of being delay of a flight is not only limited to the compensation fee, but also a lot of implicit costs.

We make an assumption that the loss , including the compensation fee and implicit costs, is 100$/per passenger. Then the reduced monthly cost turns to 42400820 * x. And when x is over 1.4%, then increasing 1% employees would be profitable.

4 Conclusion

From our analysis, we draw the conclusion that

- airlines companies suffer huge financial loss every year due to delay $8.3 billion every year;
- JetBlue performs worse than the rest of the industry for 6 consecutive years ;
- although its prevailing states FL, NY, MA had the largest delay time in 2017, its delay time per flight is not the highest;
- one solution is to cut schedule, which is already taken by JetBlue last April
- another solution is to hire more employees to reduce airline delays. Suppose a 1% increase in staff will bring x% reduce of delay rate. Assume that the airline will lose $100 per passenger per delay, then JetBlue is profitable as long as x% is greater than 1.4%.
- x% is a hypothetical threshold value. We recommend the airline to do A/B test or gray test to accumulate the data and estimate x%.

5 Code

Listing 1: Python Code1

```
import numpy as np
import pandas as pd
import sklearn
```

```
import os
import glob
import plotly.express as px
import matplotlib.pyplot as plt

path = os.getcwd()
csv_files = glob.glob(os.path.join(path, "*.csv"))
# loop over the list of csv files
for f in csv_files:

   # read the csv file
   df = pd.read_csv(f)

   # print the location and filename
   print('Location:', f)
   print('File Name:', f.split("\\")[-1])

   print(df.columns)
   # print the content
   print('Content:')
   print(df.head())

   print('-' * 80)
df_flight_traffic = pd.read_csv('7. Flight_Traffic_2017.csv')
df_flight_traffic.head()
jet_blue_2017 = df_flight_traffic[df_flight_traffic['airline_id'] ==
   'B6']
print(jet_blue_2017.head())
# finding na
print('Printing null counts in every column', '\n',
   jet_blue_2017.isnull().sum())
# drop ones with actual_arrival/actual_elapsed == null
jet_blue_2017 = jet_blue_2017[(jet_blue_2017.actual_arrival.isnull()
   == False) & (jet_blue_2017.actual_elapsed.isnull() == False)]
# replace zero
jet_blue_2017 = jet_blue_2017.fillna(0)
# calculate total delay min
jet_blue_2017['Total_delay(min)'] = jet_blue_2017.iloc[:,
   -5:].sum(axis=1)
jet_blue_2017.head()
# without month grouping
(jet_blue_2017.groupby(['origin_airport','destination_airport','month'])
 .aggregate({'cancelled': 'count', 'diverted':'count',
    'Total_delay(min)': 'sum', 'distance':'sum'})
)
airport_distiance =
   (jet_blue_2017[['origin_airport','destination_airport','distance']]
```

```
                    .groupby(['origin_airport','destination_airport','distance'])
                    .size()
                    .reset_index()
                    .iloc[:,0:3])
airport_distiance
jet_blue_2017_grp =
    (jet_blue_2017.groupby(['origin_airport','destination_airport'])
 .aggregate({'cancelled': 'count', 'Total_delay(min)': 'sum'})
 .sort_values(by=['Total_delay(min)','cancelled'], ascending=False)
)
jet_blue_2017_grp.rename(columns={"cancelled":"flight count"},
    inplace=True)
jet_blue_2017_grp
# looking at ratio of total_delay per route
# because delay time can be large due to number of flights flying is
    large
jet_blue_2017_grp['Delay Ratio(min/flight)'] =
    jet_blue_2017_grp['Total_delay(min)'] / jet_blue_2017_grp['flight
    count']
jet_blue_2017_grp.sort_values(by='Delay Ratio(min/flight)',
    ascending=False)
total_delay_ratio = jet_blue_2017_grp['Total_delay(min)'].sum() /
    jet_blue_2017_grp['flight count'].sum()
jet_blue_2017_grp['Delay_Ratio_Relativity'] = jet_blue_2017_grp['Delay
    Ratio(min/flight)'] / total_delay_ratio
# jet_blue_2017_grp.sort_values(by='Delay_Ratio_Relativity',
    ascending=False)
jet_blue_2017_grp
total_delay_ratio
state_airport_df= pd.read_csv('5. Air_Traffic_2021.csv')
state_airport_df =
    state_airport_df.groupby(['ORIGIN','ORIGIN_STATE_ABR']).size().reset_index().ilo
jet_blue_2017_grp = jet_blue_2017_grp.reset_index()
state_airport_df
# merge it with groupby
plot_2017_df = jet_blue_2017_grp.merge(state_airport_df,
                              left_on='origin_airport',
                              right_on='ORIGIN')
# rename state name into origin_state
plot_2017_df = plot_2017_df.rename(columns={'ORIGIN_STATE_ABR':
    'origin_state'})
plot_2017_df.drop(columns=['ORIGIN'], inplace=True)
plot_2017_df = plot_2017_df.merge(state_airport_df,
                              left_on='destination_airport',
                              right_on='ORIGIN')
plot_2017_df = plot_2017_df.rename(columns={'ORIGIN_STATE_ABR':
    'destination_state'})
```

```
plot_2017_df.drop(columns=['ORIGIN'], inplace=True)
plot_2017_df['flights %'] = plot_2017_df['flight count'] /
   np.sum(plot_2017_df['flight count'])
plot_2017_df.head()
# aggregate into state level
(plot_2017_df.loc[:,['Total_delay(min)','flight count',
   'origin_state','destination_state']]
         .groupby(['origin_state','destination_state']).sum()
 )
 # state in flow and outflow
inflow_df = (plot_2017_df.loc[:,['Total_delay(min)','flight count',
   'origin_state']]
          .groupby(['origin_state'])
          .sum()
          .reset_index())
outflow_df = (plot_2017_df.loc[:,['Total_delay(min)','flight
   count','destination_state']]
          .groupby(['destination_state'])
          .sum()
          .reset_index())
inflow_df['flow'] = 'Origin'
outflow_df['flow'] = 'Destination'
state_innout_flow =
   pd.concat([inflow_df.rename(columns={'origin_state':'state'}),
                     outflow_df.rename(columns={'destination_state':'state'})])
state_innout_flow.sort_values('Total_delay(min)',ascending=False)
state_innout_flow['Delay rate'] =
   state_innout_flow['Total_delay(min)'] / state_innout_flow['flight
   count']
fig5 = plot_state_level(state_innout_flow, y='Delay rate', origin=True)
fig5.update_layout(title='2017 JetBlue {0} Flight (min/ flight) {1}
   State Map'.format('Average Delay per ', 'Origin'))
fig5.show()
## plot for stacked bar chart
fig =
   px.bar(state_innout_flow[state_innout_flow.state.isin(['NY','FL','MA','CA','NJ',
          x="state",
          y="Total_delay(min)",
          color="flow",
          title="JetBlue 2017 State-wide Delay Time (min)")
fig.show()
fig.write_html("JetBlue 2017 State-wide Delay Time (min).html")
## origin state aggregation
def plot_state_level(df, y, origin=True):
   """

   :param df: df
```

```
    :param y: string, is the values you want to plot
    :param origin: bool
    :return:
    """
    flow = 'Origin' if origin else 'Destination'
    fig = px.choropleth(data_frame=df.loc[df['flow']==flow,
       [y,'state']],
                     locations='state',
                     locationmode="USA-states",
                     color=y, scope="usa")
    return fig

fig1 = plot_state_level(state_innout_flow, y='Total_delay(min)',
    origin=True)
fig1.update_layout(title='2017 JetBlue {0} Flight {1} State
    Map'.format('Total Delay(min)', 'Origin'))
fig1.show()
fig2 = plot_state_level(state_innout_flow, y='Total_delay(min)',
    origin=False)
fig2.update_layout(title='2017 JetBlue {0} Flight {1} State
    Map'.format('Total Delay(min)', 'Destination'))
fig2.show()
airport_distiance.head()
plot_2017_df = plot_2017_df.merge(airport_distiance,
              on=['origin_airport','destination_airport'])
plot_2017_df['total_distance_flew'] = plot_2017_df['flight count'] *
    plot_2017_df['distance']
plot_2017_df.head()
# state in flow and outflow
inflow_df = (plot_2017_df.loc[:,['total_distance_flew',
    'origin_state']]
          .groupby(['origin_state'])
          .sum()
          .reset_index())
outflow_df =
    (plot_2017_df.loc[:,['total_distance_flew','destination_state']]
          .groupby(['destination_state'])
          .sum()
          .reset_index())
inflow_df['flow'] = 'Origin'
outflow_df['flow'] = 'Destination'
state_innout_flow =
    pd.concat([inflow_df.rename(columns={'origin_state':'state'}),
                        outflow_df.rename(columns={'destination_state':'state'})])
state_innout_flow.sort_values('total_distance_flew',ascending=False)
## plot for stacked bar chart
```

```
fig =
   px.bar(state_innout_flow[state_innout_flow.state.isin(['NY','FL','MA','CA','NJ',
         x="state",
         y="total_distance_flew",
         color="flow",
         title="JetBlue 2017 State-wide Total Distance Flew (Miles)")
fig.show()
fig.write_html("JetBlue 2017 State-wide Total Distance Flew.html")
fig3 = plot_state_level(state_innout_flow, y='total_distance_flew',
   origin=True)
fig3.update_layout(title='2017 JetBlue {0} Flight {1} State
   Map'.format('Total Distance Flew(miles)', 'Origin'))
fig3.show()
fig4 = plot_state_level(state_innout_flow, y='total_distance_flew',
   origin=True)
fig4.update_layout(title='2017 JetBlue {0} Flight {1} State
   Map'.format('Total Distance Flew(miles)', 'Destination'))
fig4.show()
temp =
   (jet_blue_2017.groupby(['origin_airport','destination_airport','month'])
 .aggregate({'cancelled': 'count', 'diverted':'sum',
    'Total_delay(min)': 'sum', 'distance':'sum'})
      .reset_index()
      .rename(columns={'cancelled':'flight count'})
      .merge(plot_2017_df[['origin_airport','destination_airport','origin_state','c
         on=['origin_airport','destination_airport'])
)
temp[temp['origin_state']=='NY'].groupby('month').sum()
```

Listing 2: Python Code2

```
%load_ext autoreload
%autoreload 2
from google.colab import drive
drive.mount('/content/drive')
import os
import shutil
import sys
assert sys.version_info[0]==3
assert sys.version_info[1] >= 5

GOOGLE_DRIVE_PATH_AFTER_MYDRIVE = 'Ross Datathon Competition/data'
GOOGLE_DRIVE_PATH = os.path.join('drive', 'MyDrive',
   GOOGLE_DRIVE_PATH_AFTER_MYDRIVE)
sys.path.append(GOOGLE_DRIVE_PATH)

print(os.listdir(GOOGLE_DRIVE_PATH))
import numpy as np
```

```
import pandas as pd

import os
import bz2
import pandas as pd
import re
import numpy as np
from sklearn.model_selection import train_test_split
from bs4 import BeautifulSoup
from textblob import TextBlob
import string
import time
import matplotlib.pyplot as plt
import seaborn as sns
from brokenaxes import brokenaxes
tock_price_2016_2021 = pd.read_csv(GOOGLE_DRIVE_PATH + '/2.
   US_Airlines_StockPrice_2016_2021.csv')
oil_price_2016_2021 = pd.read_csv(GOOGLE_DRIVE_PATH + '/3.
   Crude_Oil_Price_2016_2021.csv')
flight_delay_2016_2021 = pd.read_csv(GOOGLE_DRIVE_PATH + '/4.
   Flight_Delay_2016_2021.csv')
air_traffic_2021 = pd.read_csv(GOOGLE_DRIVE_PATH + '/5.
   Air_Traffic_2021.csv')
flight_traffic_2017 = pd.read_csv(GOOGLE_DRIVE_PATH + '/7.
   Flight_Traffic_2017.csv')
weather_summary_2017 = pd.read_csv(GOOGLE_DRIVE_PATH + '/8.
   Weather_Summary_2017.csv')
flight_traffic_2017 = flight_traffic_2017.fillna(0)

flight_traffic_2017 = flight_traffic_2017.fillna(0)
flight_traffic_2017['overall_delay_flag'] =
   np.where(flight_traffic_2017['airline_delay']+
                  flight_traffic_2017['weather_delay']+
                  flight_traffic_2017['air_system_delay']+
                  flight_traffic_2017['security_delay']+
                  flight_traffic_2017['aircraft_delay'] > 0,
                  1, 0
                 )
flight_traffic_2017['overall_delay_flag'].value_counts()
delay_rate =
   flight_traffic_2017['overall_delay_flag'].value_counts()[1] /
   sum(flight_traffic_2017['overall_delay_flag'].value_counts())
print('overall_delay_rate:{:.2f}%'.format(delay_rate*100))
flight_traffic_2017['airline_delay_flag'] =
   np.where(flight_traffic_2017['airline_delay'] > 0,
                  1, 0
                 )
```

```
flight_traffic_2017['weather_delay_flag'] =
   np.where(flight_traffic_2017['weather_delay'] > 0,
                  1, 0
                 )
flight_traffic_2017['air_system_delay_flag'] =
   np.where(flight_traffic_2017['air_system_delay'] > 0,
                  1, 0
                 )
flight_traffic_2017['security_delay_flag'] =
   np.where(flight_traffic_2017['security_delay'] > 0,
                  1, 0
                 )
flight_traffic_2017['aircraft_delay_flag'] =
   np.where(flight_traffic_2017['aircraft_delay'] > 0,
                  1, 0
                 )
airline_delay_rate =
   flight_traffic_2017['airline_delay_flag'].value_counts()[1]/sum(flight_traffic_2
weather_delay_rate =
   flight_traffic_2017['weather_delay_flag'].value_counts()[1]/sum(flight_traffic_2
air_system_delay_rate =
   flight_traffic_2017['air_system_delay_flag'].value_counts()[1]/sum(flight_traffi
security_delay_rate =
   flight_traffic_2017['security_delay_flag'].value_counts()[1]/sum(flight_traffic_
aircraft_delay_rate =
   flight_traffic_2017['aircraft_delay_flag'].value_counts()[1]/sum(flight_traffic_
flight_traffic_2017['aircraft_delay_flag'].value_counts()
print('airline_delay_rate:{:.2f}%'.format(airline_delay_rate*100))
print('weather_delay_rate:{:.2f}%'.format(weather_delay_rate*100))
print('air_system_delay_rate:{:.2f}%'.format(air_system_delay_rate*100))
print('security_delay_rate:{:.2f}%'.format(security_delay_rate*100))
print('aircraft_delay_rate:{:.2f}%'.format(aircraft_delay_rate*100))
JetBlue_2017 = flight_traffic_2017[flight_traffic_2017['airline_id']
   == 'B6']
JetBlue_2017['airline_delay_flag'] =
   np.where(JetBlue_2017['airline_delay'] > 0,
                  1, 0
                 )
JetBlue_2017['weather_delay_flag'] =
   np.where(JetBlue_2017['weather_delay'] > 0,
                  1, 0
                 )
JetBlue_2017['air_system_delay_flag'] =
   np.where(JetBlue_2017['air_system_delay'] > 0,
                  1, 0
                 )
JetBlue_2017['security_delay_flag'] =
```

```
   np.where(JetBlue_2017['security_delay'] > 0,
                  1, 0
                 )
JetBlue_2017['aircraft_delay_flag'] =
   np.where(JetBlue_2017['aircraft_delay'] > 0,
                  1, 0
                 )
airline_delay_rate =
   JetBlue_2017['airline_delay_flag'].value_counts()[1]/sum(JetBlue_2017['airline_d
weather_delay_rate =
   JetBlue_2017['weather_delay_flag'].value_counts()[1]/sum(JetBlue_2017['weather_d
air_system_delay_rate =
   JetBlue_2017['air_system_delay_flag'].value_counts()[1]/sum(JetBlue_2017['air_sy
security_delay_rate =
   JetBlue_2017['security_delay_flag'].value_counts()[1]/sum(JetBlue_2017['security
aircraft_delay_rate =
   JetBlue_2017['aircraft_delay_flag'].value_counts()[1]/sum(JetBlue_2017['aircraft
print('airline_delay_rate:{:.2f}%'.format(airline_delay_rate*100))
print('weather_delay_rate:{:.2f}%'.format(weather_delay_rate*100))
print('air_system_delay_rate:{:.2f}%'.format(air_system_delay_rate*100))
print('security_delay_rate:{:.2f}%'.format(security_delay_rate*100))
print('aircraft_delay_rate:{:.2f}%'.format(aircraft_delay_rate*100))
y = [airline_delay_rate, weather_delay_rate, air_system_delay_rate,
   security_delay_rate, aircraft_delay_rate]
y = y / np.sum(y)
plt.pie(y,
   labels=['airline delay rate','weather delay rate','air system delay
      rate','security delay rate', 'pre-flight delay rate'],
   autopct='%.2f%%')
plt.title("Ratio of 5 delay reasons of JetBlue Airline")
plt.show()
JetBlue_2017 = flight_traffic_2017[flight_traffic_2017['airline_id']
   == 'B6']
Delta_2017 = flight_traffic_2017[flight_traffic_2017['airline_id'] ==
   'DL']
JetBlue_2017['airline_delay_flag'] =
   np.where(JetBlue_2017['airline_delay'] < 180, 0,
                  np.where(JetBlue_2017['airline_delay'] < 240, 1,
                  np.where(JetBlue_2017['airline_delay'] < 300, 2,
                  np.where(JetBlue_2017['airline_delay'] < 360, 3,
                  4))))
JetBlue_2017['airline_delay_flag'] =
   np.where(JetBlue_2017['airline_delay'] < 60, 0,
                  np.where(JetBlue_2017['airline_delay'] < 120, 1,
                  np.where(JetBlue_2017['airline_delay'] < 180, 2, 3
                  )))
JetBlue_2017['airline_delay_flag'].value_counts()
```

```
# df = pd.DataFrame()
# df['flag'] = ['0-3', '3-4', '4-5', '5-6', '6+']
# df['delay count'] = [29926, 58, 31, 10, 11]
# plt.bar(data=df, x='flag', y='delay count')

x = [29926, 58, 31, 10, 11]
fig = plt.figure(figsize=(10,6))
bax = brokenaxes(ylims=((0, 60), (29900, 29940)), hspace=0.3,
    despine=False)
bax.bar(['0-3', '3-4', '4-5', '5-6', '6+'], x)
bax.set_xlabel('delay time (hours)')
bax.set_ylabel('count if delay flights')
bax.set_title('Barplot of delay flights and delay hours of JetBlue')


delay_per_year_JetBlue =
    JetBlue_2017['airline_delay_flag'].value_counts()
compensation_fee_list = np.array([0, 50, 125, 200])
compensation_fee = sum(delay_per_year_JetBlue * compensation_fee_list)
fee_per_person = compensation_fee / sum(delay_per_year_JetBlue)
fee_per_person
# delay_rate_decrease = 0.1
Loadfactor_2017 = 0.847
seats_per_airplane = 200
crew_salary_one_percent = 600000
#loss_per_person = 100
loss_per_person = fee_per_person
flight_2017 = sum(delay_per_year_JetBlue)

y = crew_salary_one_percent / (seats_per_airplane * Loadfactor_2017 *
    loss_per_person * flight_2017/12)
print(y)
```

Listing 3: R code

```
library(dplyr)
library(ggplot2)
library(readr)
# Read csv file
df = read.csv(file = file.path(pa, '4. Flight_Delay_2016_2021.csv'))
df = df %>%
   mutate(controllable_del=carrier_ct,
         uncontrollable_del=weather_ct+nas_ct+security_ct+late_aircraft_ct,
         total_del_can=carrier_ct+weather_ct+nas_ct+security_ct+late_aircraft_ct+ar
df_jetBlue = df %>% filter(carrier_name=="JetBlue Airways") %>%
  group_by(year, month) %>%
  summarise(control_del_rate=sum(controllable_del, na.rm =
     TRUE)/sum(arr_flights, na.rm = TRUE),
```

```
          uncontrol_del_rate=sum(uncontrollable_del, na.rm =
             TRUE)/sum(arr_flights, na.rm = TRUE),
          del_cancel_rate=sum(total_del_can, na.rm =
             TRUE)/sum(arr_flights, na.rm = TRUE))
df_jetBlue_can_del = df_jetBlue %>% mutate(label="JetBlue")
df_total_can_del = df_del_can %>% mutate(label="All")
df_can_del_combine = rbind(df_jetBlue_can_del, df_total_can_del)
ggplot(data = df_can_del_combine, aes(x = month, y = control_del_rate,
   color=label)) +
   geom_line() + ggtitle("Controllable Delay Rate in 12 Months
      (JetBlue v.s. All)") + ylab("Controllable Delay Rate") +
   facet_wrap(facets = vars(year))
ggplot(data = df_can_del_combine, aes(x = month, y =
   uncontrol_del_rate, color=label)) +
   geom_line() + ggtitle("Uncontrollable Delay Rate in 12 Months
      (JetBlue v.s. All)") + ylab("Uncontrollable Delay Rate") +
   facet_wrap(facets = vars(year))
ggplot(data = df_can_del_combine, aes(x = month, y = del_cancel_rate,
   color=label)) +
   geom_line() + ggtitle("Total Delay and cancel Rate in 12 Months
      (JetBlue v.s. All)") + ylab("Total Delay and Cancel Rate") +
   facet_wrap(facets = vars(year))
```