Project 2: Late Flights and Missing Data

Data Science Programming CSE 250

Travis Smith

Project Summary

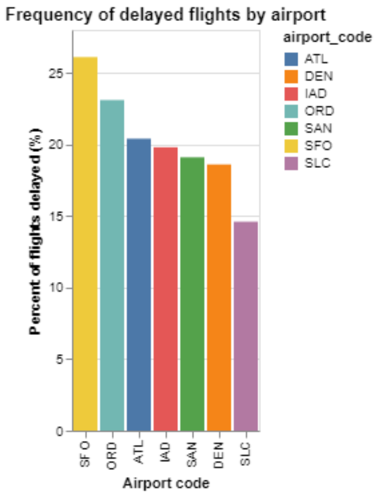
We are exploring a json file to look at flights in corelation to delays. We are looking at different types of delays and their duration. We can also see which airports are performing the best. The goal of this project is to understand all the methods to filter and get the data that we want to use.

Technical Details

Grand Question 1:

Which airport has the worst delays? Discuss how you chose to define "worst". Your answer should also include a table that lists (for each airport) the total number of flights, total number of delayed flights, proportion of delayed flights, and average delay time in hours.

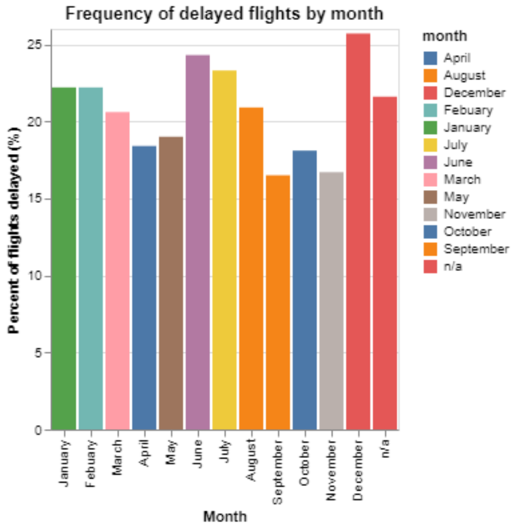
The airport with the most delays in total was Altanta (ALT); however, the airport with the most delays in proportion to the number of flights is San Fransico (SFO). So if you want a lower chance of a delayed flight I would avoid San Fransisco (SFO)



Grand Question 2:

What is the best month to fly if you want to avoid delays of any length? Discuss your answer. Include one chart to help support your answer, with the x-axis ordered by month. (To answer this question, you will need to remove any rows that are missing the Month variable.)

As you may have guessed, the worst months to travel in are December, June, and July. So the best month to travel in would be September and November to beat the crowds.



Grand Question 3:

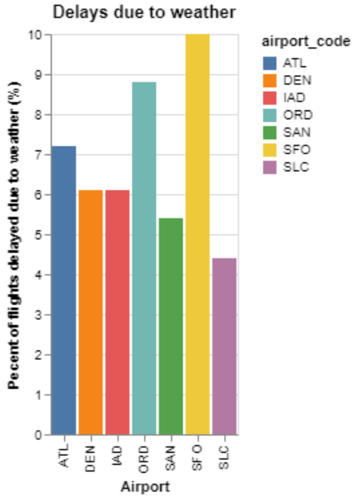
According to the BTS website, the “Weather” category only accounts for severe weather delays. Mild weather delays are not counted in the “Weather” category, but are actually included in both the “NAS” and “Late-Arriving Aircraft” categories. Your job is to create a new column that calculates the total number of flights delayed by weather (both severe and mild). You will need to replace all the missing values in the Late Aircraft variable with the mean. Show your work by printing the first 5 rows of data in a table. Use these three rules for your calculations:

```
a. 100% of delayed flights in the Weather category are due to weather.
b. 30% of delayed flights in the Late-Arriving category are due to weather.
c. From April to August, 40% of delayed flights in the NAS category are due to weather. The rest of the months, the proportion
rises to 65%.
```

	month	num_of_delays_weather	num_of_delays_late_aircraft	num_of_delays_nas	num_of_delays_total	all_weather_delays	num_of_flights_total	ai
0	January	448.0	1018.0	4598.0	8355.0	3742.0	35
1	January	233.0	928.0	935.0	3153.0	1119.0	12
2	January	61.0	1058.0	895.0	2430.0	960.0	12
3	January	306.0	2255.0	5415.0	9178.0	4502.0	28
4	January	56.0	680.0	638.0	1952.0	675.0	72

Grand Quesiton 4:

Using the new weather variable calculated above, create a barplot showing the proportion of all flights that are delayed by weather at each airport. Discuss what you learn from this graph.



So it seems like the San Fransisco Airport has the most delays due to weather.

Grand Question 5:

Fix all of the varied missing data types in the data to be consistent (all missing values should be displayed as “NaN”). In your report include one record example (one row) from your new data, formatted as a JSON file. You example should have at least one missing value.

Example from the first line of the JSON: Before fixing error codes: 1500+ After fixing error codes: NaN

Appendix A

```
import pandas as pd
import altair as alt
import numpy as np
flights = pd.read_json('flights_missing.json')
#flight = pd.read_json('https://raw.githubusercontent.com/byuidatascience/data4missing/master/data-
raw/flights_missing/flights_missing.json')
flights.head(8)
### Grand question 1:

q1 = (flights
    .filter(items=['airport_code', 'num_of_delays_total', 'num_of_flights_total'])
    .groupby('airport_code')
    .agg('sum')
    .assign(perc_delayed= lambda x: x.num_of_delays_total / x.num_of_flights_total * 100).round(1)
    .sort_values(by='perc_delayed', ascending=False)
    .reset_index())
```

```
# print(q1.filter(items='num_of_delays_total').max())
print(q1)
chart_q1 = (alt.Chart(q1, title='Frequency of delayed flights by airport')
    .mark_bar()
    .encode(
        x = alt.X('airport_code', axis = alt.Axis(title = "Airport code"), sort=alt.EncodingSortField(field="Letters", 
op="count", order='ascending')),
        y = alt.Y('perc_delayed', axis = alt.Axis(title = "Percent of flights delayed (%)")),
        color = 'airport_code'))

chart_q1
### Grand question 2:

q2 = (flights
    .filter(items=['month', 'num_of_delays_total', 'num_of_flights_total'])
    .groupby('month')
    .agg('sum')
    .assign(perc_delayed= lambda x: x.num_of_delays_total / x.num_of_flights_total * 100).round(1)
    .sort_values(by='perc_delayed', ascending=False)
    .reset_index())

# print(q1.filter(items='num_of_delays_total').max())
q2.head(12)
chart_q2 = (alt.Chart(q2, title='Frequency of delayed flights by month')
    .mark_bar()
    .encode(
        x = alt.X('month', axis = alt.Axis(title = "Month"), sort=['January', 'Febuary', 'March', 'April', 'May', 'June', 
'July', 'August', 'September', 'October', 'November', 'December', 'n/a']),
        y = alt.Y('perc_delayed', axis = alt.Axis(title = "Percent of flights delayed (%)")),
        color = 'month'))

chart_q2

### Grand question 3:

avg_late = flights.num_of_delays_late_aircraft.mean()

flights.replace([-999], avg_late, inplace=True)

q3 = flights.assign(
    weather1 = flights.num_of_delays_weather,
    weather2 = flights.num_of_delays_late_aircraft * 0.3,
    weather3 = np.where(flights.month.isin(['April', 'June', 'July', 'August']), 0.4 * flights.num_of_delays_nas, 0.65 * 
flights.num_of_delays_nas),
    all_weather_delays = lambda x: x.weather1 + x.weather2 + x.weather3).round(0)

q3.filter(items=['month', 'num_of_delays_weather', 'num_of_delays_late_aircraft', 'num_of_delays_nas', 'num_of_delays_total', 
'all_weather_delays', 'num_of_flights_total', 'airport_code']).head()
### Grand Question 4:

q4 = (q3
    .filter(items=['month', 'num_of_delays_weather', 'num_of_delays_late_aircraft', 'num_of_delays_nas', 'num_of_delays_total', 
'all_weather_delays', 'num_of_flights_total', 'perc_delayed', 'airport_code'])
    .groupby('airport_code')
    .agg('sum')
    .assign(perc_delayed= lambda x: x.all_weather_delays / x.num_of_flights_total * 100).round(1)
    .sort_values(by='perc_delayed', ascending=False)
    .reset_index())

# q4.filter(items=['airport_code', 'perc_delayed']).head()

chart_q4 = (alt.Chart(q4, title='Delays due to weather')
    .mark_bar()
    .encode(
        x = alt.X('airport_code', axis = alt.Axis(title = "Airport")),
        y = alt.Y('perc_delayed', axis = alt.Axis(title = "Pecent of flights delayed due to weather (%)")),
        color = 'airport_code'))

chart_q4
### Grand Question 5:

print('Before fixing error codes:', flights.loc[0].num_of_delays_carrier)

flights.replace(-999, 'NaN', inplace=True)
flights.replace('1500+', 'NaN', inplace=True)
flights.replace('', 'NaN', inplace=True)
flights.replace('n/a', 'NaN', inplace=True)

print('After fixing error codes: ', flights.loc[0].num_of_delays_carrier)
```