Project 2

Alice Pao

Project Summary

This porject focuses on analysiing total number of flights and delayed flights of seven airports from 2005-2015. It points out the airport having the worst delays, the best month to avoid delays, total number of weather delays, and replace the missing data into the official form.

Technical Detials

Grand Question 1

Which airport has the worst delays?

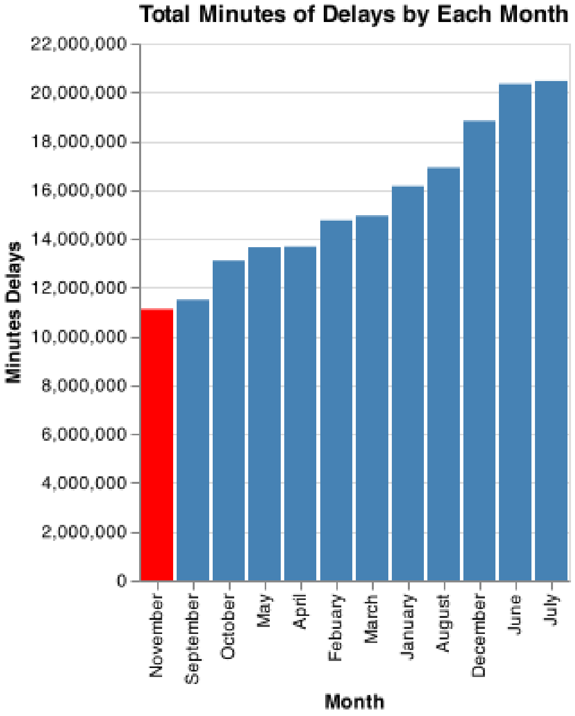
In the following table, SFO has the worst delay because of its highest proportion of delays. This means that out of all its total flights and total delayed flights from 2005 to 2015, SFO has 26% of delays.

airport_code	total_flights	total_delayed_flights	hours_delayed_average	proportion_of_delays
SFO	1630945	425604	3352.33	0.260955
ORD	3597588	830825	7115.67	0.230939
ATL	4430047	902443	6816.15	0.20371
IAD	851571	168467	1298.42	0.197831
SAN	917862	175132	1044.98	0.190804
DEN	2513974	468519	3178.46	0.186366
SLC	1403384	205160	1278.2	0.146189

Grand Question 2

What is the best month to fly if you want to avoid delays of any length?

The best month to avoid delays is in November. As shown in the following bar chart, November has the least total of delayed minutes compared to other 11 months.



Grand Question 3

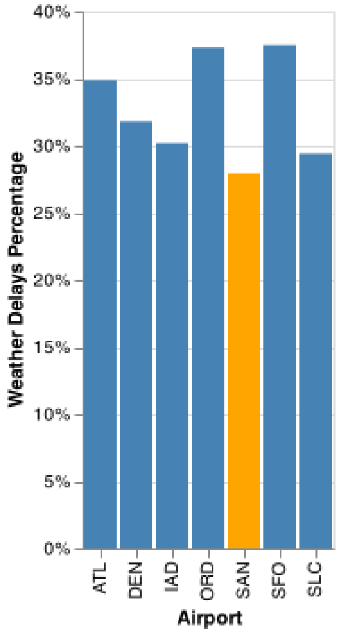
Create a new column that calculates the total number of flights delayed by weather. (both severe and mild)

airport_code	month	severe	mild_late	mild_nas	total_weather	num_of_delays_total
ATL	January	448	332.731	2988.7	3769.43	8355
DEN	January	233	278.4	607.75	1119.15	3153
IAD	January	61	317.4	581.75	960.15	2430
ORD	January	306	676.5	3519.75	4502.25	9178
SAN	January	56	204	414.7	674.7	1952

Grand Question 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.

From the bar chart, it shows that SAN has the least weather delays. On the other hand, ORD and SFO have the most flight delays becuase of the 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”)

The row I chose to present here is the airport_code: ATL, month: April, year: 2008

```
[
{
"airport_code": "ATL",
"airport_name": null,
"month": "April",
"year": 2008.0,
"num_of_flights_total": 34623,
"num_of_delays_carrier": null,
"num_of_delays_late_aircraft": null,
"num_of_delays_nas": 3123,
"num_of_delays_security": 5,
"num_of_delays_weather": 422,
"num_of_delays_total": 7157,
"minutes_delayed_carrier": 129813.0,
"minutes_delayed_late_aircraft": 109124,
"minutes_delayed_nas": 140254.0,
"minutes_delayed_security": 505,
"minutes_delayed_weather": 35650,
"minutes_delayed_total": 415346
}
]
```

Apendix A

```
#%%
import pandas as pd
import altair as alt
import numpy as np
import json

# import data
flights = pd.read_json('https://github.com/byuidatascience/data4missing/raw/master/data-raw/flights_missing/flights_

delayed = (flights.filter(['airport_code', 'num_of_flights_total', 'num_of_delays_total', 'minutes_delayed_total'])
    .assign(percent_delay = lambda x: x.num_of_delays_total/x.num_of_flights_total
    , ave_delay_hours = lambda x: x.minutes_delayed_total/x.num_of_flights_total/60))
delayed.groupby('airport_code')

# %%

# Grand Question 1:
# create a datafrom with a new column for hours_delayed_total by using minutes_delayed_total divided by 60
flights1 =  (flights.filter(['airport_code', 'num_of_flights_total', 'num_of_delays_total', 'minutes_delayed_total']
            .assign(hours_delayed_total = lambda x: x.minutes_delayed_total / 60))

#%%
# create a table with total_flights, total_delayed_flights, proportion_of_delays, and average_hours_delayed columns
table_one = (flights1.filter(['airport_code', 'num_of_flights_total', 'num_of_delays_total', 'hours_delayed_total',
    # group by different airport; turn the airport_code into a column
    .groupby('airport_code', as_index= False)
    # calculate total_flights, total_delayed_flights (sum), and hours_delayed_average (average)
    .agg({'num_of_flights_total':'sum', 'num_of_delays_total':'sum', 'hours_delayed_total':'mean'}).rename(columns={
    # create new column named proportion_of_delays
    .assign(proportion_of_delays = lambda x: x.total_delayed_flights / x.total_flights)
    # sort rows with proportion_of_delays value
    .sort_values(by='proportion_of_delays', ascending=False)
    .reset_index(drop=True)
    .to_markdown(index=False)
)
table_one

# issue for this question: 1. round 2. percentage sign 3. mark_down table
#%%

# Grand Question 2:
# filter out the missing value in month column
flights_month_filter = flights.query('month != "n/a"').filter(['month', 'minutes_delayed_total', 'year'])
# create a datafram to sum up all months of minutes_delayed_total
flights2 = (flights_month_filter.groupby('month')
    .agg({'minutes_delayed_total': 'sum'})
    )
# create a bar chart
chart2 = (alt.Chart(flights2.reset_index(), title = 'Total Minutes of Delays by Each Month')
    .mark_bar()
    .encode(
        # sort function means to arrange the bars with ascending order
        x = alt.X('month', axis = alt.Axis(title = 'Month'), sort = 'y'),
        y = alt.Y('minutes_delayed_total', axis = alt.Axis(title = 'Minutes Delays')),
        # highlight the month with the least minutes delayed in total
        color = alt.condition(alt.datum.month == 'November', alt.value('red'), alt.value('steelblue'))
        )
```

```
)
chart2.save('question_2.png')

# Grand Question 3:
# create a new column that calculates the total number of flights delayed by severe & mild wether
# replace the missing value (-999) from late_aircraft column with the average number
mean_of_late_aircraft = flights.num_of_delays_late_aircraft.replace(-999, np.nan).mean()
flights3 = flights.copy()
flights3.num_of_delays_late_aircraft = flights3.num_of_delays_late_aircraft.replace(-999, mean_of_late_aircraft)

months = ['April', 'May', 'June', 'July', 'August']
weather = flights.assign(
    severe = 1 * flights3.num_of_delays_weather,
    mild_late = .3 * flights3.num_of_delays_late_aircraft,
    mild_nas = np.where(flights3.month.isin(months), .4 * flights3.num_of_delays_nas, .65 * flights3.num_of_delays_r
    total_weather = lambda x: x.severe + x.mild_late + x.mild_nas
).filter(['airport_code','month','severe','mild_late','mild_nas',
    'total_weather', 'num_of_delays_total'])

# create the first 5 rows for markdown
print(weather.head(5).to_markdown(index=False))

# Grand Question 4:
# caulculate proportion of weather delays
weather3 = (weather.groupby('airport_code')
            .agg({'total_weather': 'sum', 'num_of_delays_total': 'sum'})
            .assign(proportion_weather_delays = lambda x: x.total_weather / x.num_of_delays_total)
            .reset_index()
)
# create barplot
chart4 = (alt.Chart(weather3)
    .mark_bar()
    .encode(
        # sort function means to arrange the bars with ascending order
        x = alt.X('airport_code', axis = alt.Axis(title = 'Airport')),
        y = alt.Y('proportion_weather_delays', axis = alt.Axis(title = 'Weather Delays Percentage', format = '%')),
        color = alt.condition(alt.datum.airport_code == 'SAN', alt.value('orange'), alt.value('steelblue')))
)
chart4.save('question_4.png')

# Grand Question 5:
# use replace() to change every missing value to NaN
flights_new = flights.replace('', np.nan).replace('n/a', np.nan).replace('1500+', np.nan).replace(-999, np.nan)
# save it to a new json file
flights_new.to_json('new_flights_data.json')
# select a specific row with missing data value
flights5_missing = flights_new.query('airport_code == "ATL" & month == "April" & year == 2008')
# print out a row
json_data = flights5_missing.to_json(orient="records")
json_object = json.loads(json_data)
json_formatted_str = json.dumps(json_object, indent = 4)
print(json_formatted_str)
```