Client Report - Late flights and missing data (JSON files)

See code

Course DS 250

AUTHOR
Bracken Sant

Elevator pitch

Using pandas and altair I was able to go through flight data, from a json file, to look at different data points, such as: which airport has the worst delays, the best months to fly, and how many weather delays there actually are. I was also able to sort some of the data and replace missing values.

▼ Read and format project data

```
al.data_transformers.enable('json')
data = pd.read_json("flights_missing.json")
```

Highlight the grand questions

GRAND QUESTION 1

Which airport has the worst delays?

The airport with the worst delays is San Francisco CA: San Francisco International. For my metric I chose to use a proportion of number of delays to number of flights for each airport and to compare them. If any were to have a similar enough proportion then I would compare the average delay in hours and see if there was a sizable enough difference to change it over to the ther one. San Francisco CA had the worst by .03 and when comparing the average delay the difference was about 5 and a half minutes, which isn't enough to change it from being the worst airport.

▼ Read and format data

```
GQ1 = (data.groupby("airport_code", as_index=False)
       .agg({"airport_code" : 'first', "airport_name": 'last', "num_of_flights_total" : 'sum',
         "num_of_delays_total" : 'sum', "minutes_delayed_total" : 'sum'}))
GQ1["average_delay_in_hours"] = (GQ1["minutes_delayed_total"] / 60 /
         GQ1["num_of_delays_total"]).map('{:.2f}'.format)
GQ1.drop('minutes_delayed_total', inplace=True, axis = 1)
GQ1.drop("airport_code", inplace=True, axis = 1)
GQ1['proportions_of_delays'] = GQ1["num_of_delays_total"] / GQ1["num_of_flights_total"]
print(GQ1.to_markdown(index=False))
```

airport_name	num_of_flights_total	num_of_delays_total	average_delay_in_hours	proportions_of_delays
Atlanta, GA: Hartsfield-Jackson Atlanta International	4430047	902443	1	0.20371
Denver, CO: Denver International	2513974	468519	0.9	0.186366
Washington, DC: Washington Dulles International	851571	168467	1.02	0.197831
Chicago, IL: Chicago O'Hare International	3597588	830825	1.13	0.230939
San Diego, CA: San Diego International	917862	175132	0.79	0.190804
San Francisco, CA: San Francisco International	1630945	425604	1.04	0.260955
Salt Lake City, UT: Salt Lake City International	1403384	205160	0.82	0.146189

From the table above you can see that the proportion of delays is greatest in the San Francisco, Ca: San Francisco International with a proportion of delays to flights at around .26 which is 3 percent over any other airport. The average delay in hours is close enough to the airport with the second highest proportion for it to be negligable, so the decision will not change based on that.

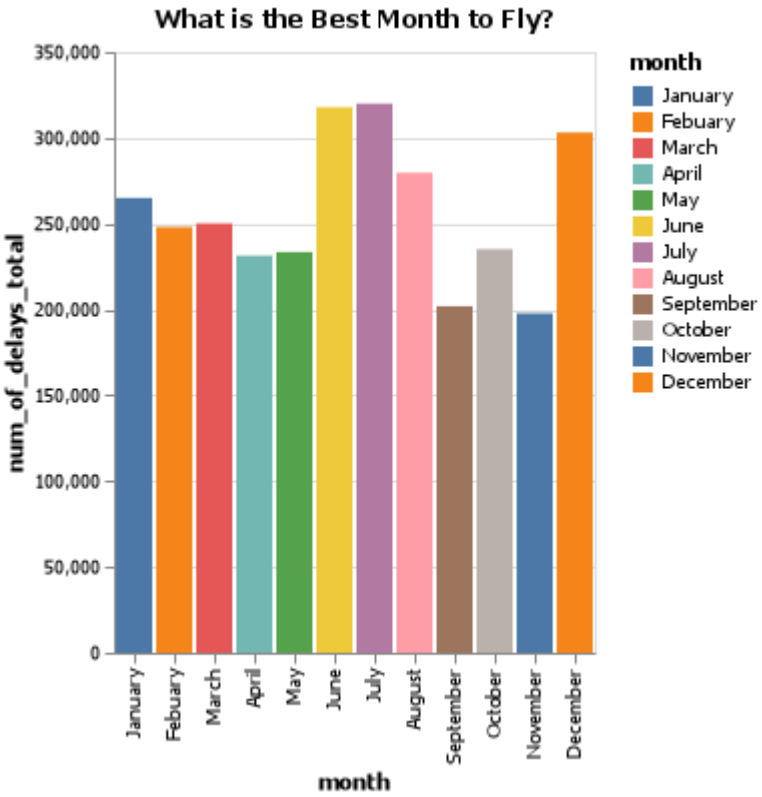
GRAND QUESTION 2

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

In the question it said of any length so I chose to look at the months and the total number of delays to check which month was best. The month with the least amount of total delays was November with September following close behind it.

▼ Read and format data

```
GQ2 = data.groupby("month", as_index=False).agg({"num_of_delays_total" : 'sum'})
GQ2 = GQ2.query("month != 'n/a'")
months = ("January", "Febuary", "March", "April", "May", "June", "July", "August", "September",
          "October", "November", "December")
chart = al.Chart(GQ2).mark_bar().encode(x=al.X('month', sort=months), y='num_of_delays_total',
          color = al.X("month", sort=months)).properties(title = "What is the Best Month to
          Fly?")
```



On the barplot you can see that November has the smallest amount of total delays.

GRAND QUESTION 3

Your job is to create a new column that calculates the total number of flights delayed by weather (both severe and mild).

For the late-arriving category I decided to do a mean from all the data. Since we don't know if those values could have been outliers from the rest of the data, I beleive that it is best to represent them all as the same.

▼ Read and format data

```
data_copy = data.copy()
count = 924-data_copy["num_of_delays_late_aircraft"].value_counts()[-999]
late_aircraft = int((((data_copy.query("num_of_delays_late_aircraft != -999"))
        ["num_of_delays_late_aircraft"].sum()))/count)
data_copy["num_of_delays_late_aircraft"] =
        data_copy["num_of_delays_late_aircraft"].replace(-999, late_aircraft)
weather_months = ("April", "May", "June", "July", "August")
other_months = ("January", "Febuary", "March", "September", "October", "November", "December")
data_copy["num_of_delays_all_weather"] = data_copy["num_of_delays_weather"]
data_copy["num_of_delays_all_weather"] += (data_copy["num_of_delays_late_aircraft"] * .3)
for month in months:
      if month in weather_months:
            data_copy.loc[data_copy["month"] == month, "num_of_delays_all_weather"] +=
        (data_copy.query("month in @weather_months")["num_of_delays_nas"] * .4)
      else:
```

```
            data_copy.loc[data_copy["month"] == month, "num_of_delays_all_weather"] +=
        (data_copy.query("month in @other_months")["num_of_delays_nas"] * .65)
print(data_copy[["airport_code", "month",
        "num_of_delays_all_weather"]].head(5).to_markdown(index=False))
```

airport_code	month	num_of_delays_all_weather
ATL	January	3769.4
DEN	January	1119.15
IAD	January	960.15
ORD	January	4502.25
SAN	January	674.7

I named the column with the total amount of weather delays: num of delays all weather

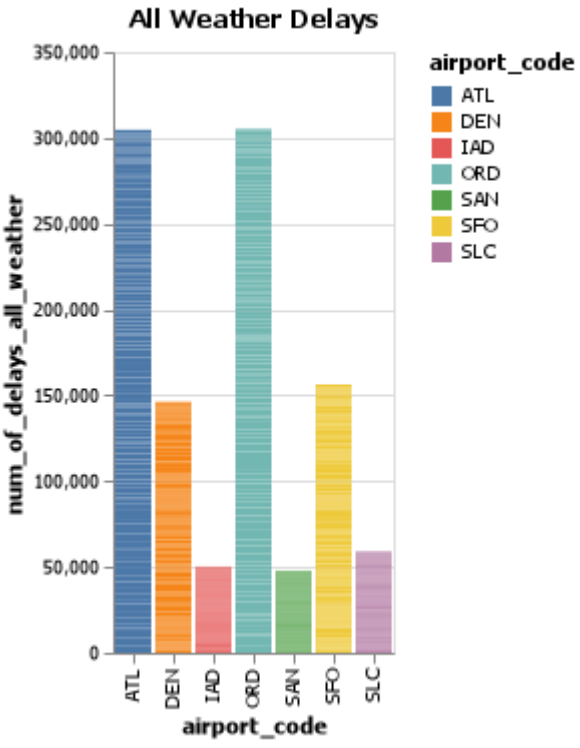
GRAND QUESTION 4

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

From the graph you can learn which airports get weather delays, which cannot be prevented. So you can know which airports to avoid bad weather seasons are around.

▼ Read and format data

```
chart2 = al.Chart(data_copy).mark_bar().encode(x="airport_code", y='num_of_delays_all_weather',
        color = "airport_code").properties(title = "All Weather Delays")
```



All Weather Delays

From the barplot you can see the airports with the most weather delays are Chicago and Atlanta.

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 the code below it is sorting through various columns and replacing their different missing values with "NaN"

▼ Read and format data

```
replace_all_missing_data = data.copy()
replace_all_missing_data["num_of_delays_late_aircraft"] =
        replace_all_missing_data["num_of_delays_late_aircraft"].replace(-999, "NaN")
replace_all_missing_data["airport_name"] = replace_all_missing_data["airport_name"].replace("",
        "NaN")
replace_all_missing_data["minutes_delayed_nas"] =
        replace_all_missing_data["minutes_delayed_nas"].replace(-999, "NaN")
replace_all_missing_data["month"] = replace_all_missing_data["month"].replace("n/a", "NaN")
print(replace_all_missing_data.iloc[13].to_json())
```

{"airport_code":"SLC","airport_name":"NaN","month":"Febuary","year":2005.0,"num_of_flights_total":12404,"num_of_delays_carrier":"645","num_of_delays_late_aircraft":463,"num_of_delays_nas":752,"num_of_delays_security":10,"num_of_delays_weather":79,"num_of_delays_total":1947,"minutes_delayed_carrier":32336.0,"minutes_delayed_late_aircraft":23087,"minutes_delayed_nas":24544.0,"minutes_delayed_security":293,"minutes_delayed_weather":4614,"minutes_delayed_total":84874}

Here you can see that in airport_name the "" has been replaced with "NaN"