notebook

March 31, 2023

1 MBML 2023

1.1 INIT

1.1.1 Load Packages

[ ]:
```
import pandas as pd
import kaggle
import os
import shutil
import requests
import urllib
from urllib.request import urlopen, urlretrieve
from io import BytesIO
from zipfile import ZipFile
import matplotlib.pyplot as plt
import plotly.express as px
from IPython.display import Image
import numpy as np


from src.data import extract, load, transform
```

1.1.2 Set Flags

[ ]:
```
pd.set_option('display.max_colwidth', None)
pd.set_option('display.max_columns', None)

DATA_DIR = "data/"
```

1.2 Data

1.2.1 Extract Data

[ ]:
```
# Download Fligt Delay Dataset form Kaggle
kaggle.api.authenticate()
kaggle.api.dataset_download_files(
    "robikscube/flight-delay-dataset-20182022",
```

```
    path=DATA_DIR,
    unzip=True,
)
for filename in os.listdir(DATA_DIR):
    f = os.path.join(DATA_DIR, filename)
    if f.endswith(".parquet") or filename == "Airlines.csv":
        pass
    else:
        if os.path.isfile(f):
            os.remove(f)
        else:
            shutil.rmtree(f)
```

[ ]:
```
# Download Location of airports
urlretrieve(
    "https://raw.githubusercontent.com/lxndrblz/Airports/main/airports.csv",
    DATA_DIR + "airports.csv"
    )
```

1.2.2 Transform Data

1.2.3 Load Data

[ ]:
```
main_df = extract.combine_parquet(data_path = "data/")
main_df['count'] = 1
airport_df = pd.read_csv('data/airports.csv')
airline_df = pd.read_csv('data/Airlines.csv')
```

1.2.4 Define display options for later export to pdf

[ ]:
```
pd.set_option('display.notebook_repr_html', True)

def _repr_latex_(self):
    return "\centering{%s}" % self.to_latex()

pd.DataFrame._repr_latex_ = _repr_latex_  # monkey patch pandas DataFrame
```

[ ]:
```
# silence future warnings
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

[ ]:
```
# set max number of rows of dataframe
pd.set_option('display.max_rows', 80)
```

1.2.5 Describe Data

Display two first rows of data

[ ]:
```
# transposing dataframe and displaying in two different cells for compatability␣
↪with pdf conversion
pd.DataFrame(main_df.iloc[:2,:30].T)
```

[ ]:

	0	1
FlightDate	2018-01-23 00:00:00	2018-01-24 00:00:00
Airline	Endeavor Air Inc.	Endeavor Air Inc.
Origin	ABY	ABY
Dest	ATL	ATL
Cancelled	False	False
Diverted	False	False
CRSDepTime	1202	1202
DepTime	1157.0	1157.0
DepDelayMinutes	0.0	0.0
DepDelay	-5.0	-5.0
ArrTime	1256.0	1258.0
ArrDelayMinutes	0.0	0.0
AirTime	38.0	36.0
CRSElapsedTime	62.0	62.0
ActualElapsedTime	59.0	61.0
Distance	145.0	145.0
Year	2018	2018
Quarter	1	1
Month	1	1
DayofMonth	23	24
DayOfWeek	2	3
Marketing_Airline_Network	DL	DL
Operated_or_Branded_Code_Share_Partners	DL_CODESHARE	DL_CODESHARE
DOT_ID_Marketing_Airline	19790	19790
IATA_Code_Marketing_Airline	DL	DL
Flight_Number_Marketing_Airline	3298	3298
Operating_Airline	9E	9E
DOT_ID_Operating_Airline	20363	20363
IATA_Code_Operating_Airline	9E	9E
Tail_Number	N8928A	N800AY

[ ]:
```
pd.DataFrame(main_df.iloc[:2,30:].T)
```

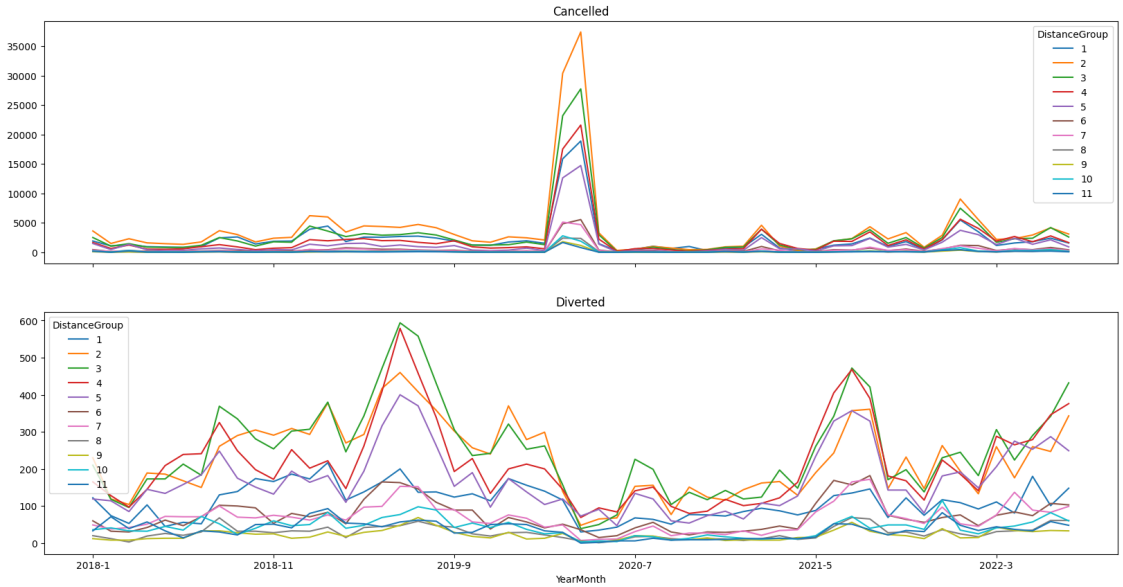
[ ]:

	0	1
Flight_Number_Operating_Airline	3298	3298
OriginAirportID	10146	10146
OriginAirportSeqID	1014602	1014602
OriginCityMarketID	30146	30146
OriginCityName	Albany, GA	Albany, GA
OriginState	GA	GA
OriginStateFips	13	13
OriginStateName	Georgia	Georgia
OriginWac	34	34
DestAirportID	10397	10397
DestAirportSeqID	1039707	1039707
DestCityMarketID	30397	30397
DestCityName	Atlanta, GA	Atlanta, GA
DestState	GA	GA
DestStateFips	13	13
DestStateName	Georgia	Georgia
DestWac	34	34
DepDel15	0.0	0.0
DepartureDelayGroups	-1.0	-1.0
DepTimeBlk	1200-1259	1200-1259
TaxiOut	14.0	13.0
WheelsOff	1211.0	1210.0
WheelsOn	1249.0	1246.0
TaxiIn	7.0	12.0
CRSArrTime	1304	1304
ArrDelay	-8.0	-6.0
ArrDel15	0.0	0.0
ArrivalDelayGroups	-1.0	-1.0
ArrTimeBlk	1300-1359	1300-1359
DistanceGroup	1	1
DivAirportLandings	0.0	0.0
count	1	1

[ ]:
```
# Plot cancellations and Diverted by length of flight
g_main_df = main_df.groupby(["Year","Month","DistanceGroup"]).sum().unstack().
  ↪reset_index()
g_main_df["YearMonth"] = g_main_df["Year"].astype(str)+"-"+g_main_df["Month"].
  ↪astype(str)

subplots = 2
fig, ax = plt.subplots(subplots,1,figsize = (20,10))
prediction_cols = ["Cancelled","Diverted"]
for i in range(subplots):
```

```
    g_main_df[[prediction_cols[i],"YearMonth"]].
↪plot(x="YearMonth",y=prediction_cols[i],kind = "line", legend = True,ax =␣
↪ax[i],sharex = True,title = prediction_cols[i])
```

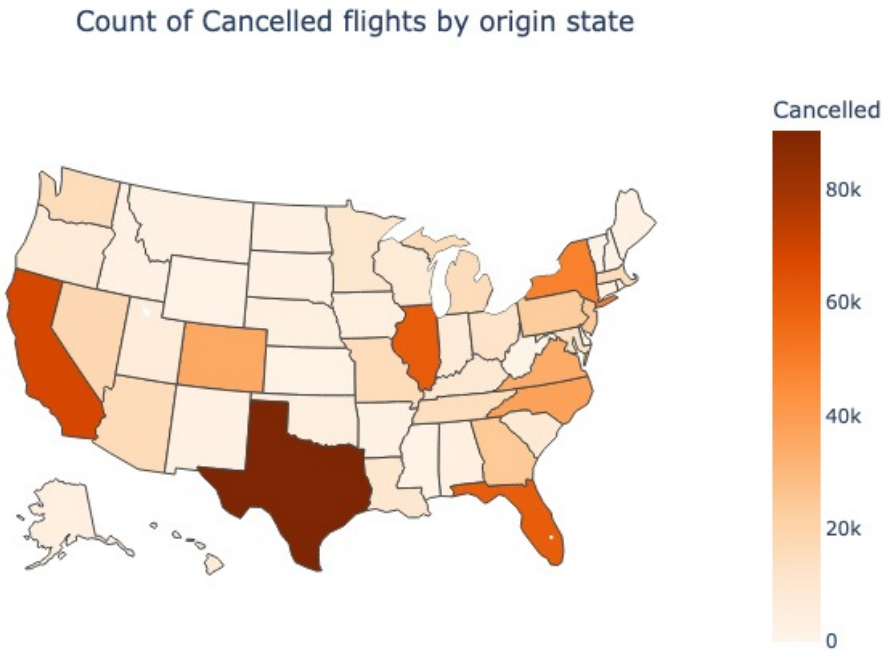


[ ]:
```
g_main_df = main_df.groupby(["OriginState"]).sum()
```

[ ]:
```
# plot a choropleth with color range by count per state
fig = px.choropleth(g_main_df['Cancelled'].reset_index(),
                    locations='OriginState',
                    locationmode="USA-states",
                    scope="usa",
                    color="Cancelled",
                    color_continuous_scale="Oranges",
                   )
# center the title
fig.update_layout(title_text='Count of Cancelled flights by origin state',␣
 ↪title_x=0.5)

# export plot to image to compatability with jupyter conversion
#fig.show()
im = fig.to_image("jpeg")
Image(im)
```

[ ]:

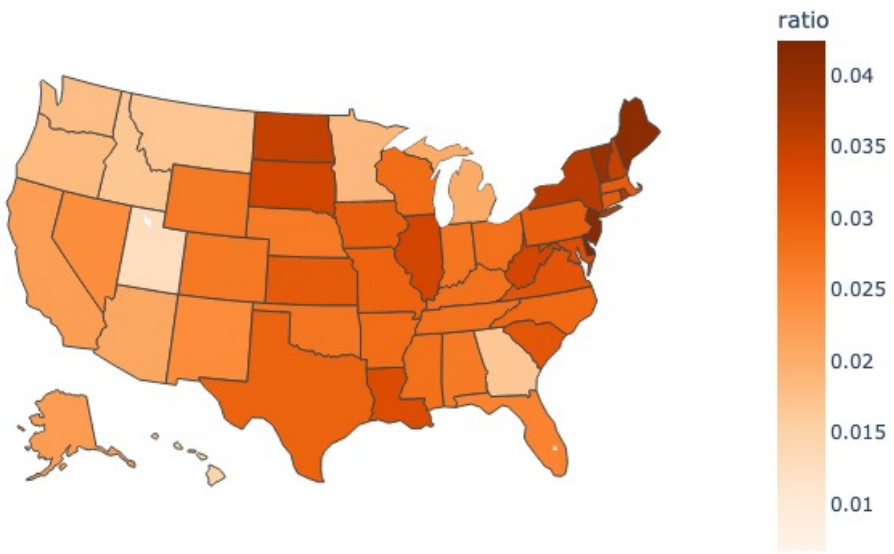


[ ]:
```
state_df = g_main_df[['Cancelled','count']].reset_index()
state_df["ratio"] = state_df['Cancelled']/state_df['count']

# plot a choropleth with color range by count per state
fig = px.choropleth(state_df,
                    locations='OriginState',
                    locationmode="USA-states",
                    scope="usa",
                    color="ratio",
                    color_continuous_scale="Oranges",
                   )
# center the title
fig.update_layout(title_text='Ratio of Cancelled flights by origin state',␣
 ↪title_x=0.5)

# export plot to image to compatability with pdf conversion
#fig.show()
im = fig.to_image("jpeg")
Image(im)
```

[ ]:


Ratio of Cancelled flights by origin state
ratio
0.04
0.035
0.03
0.025
0.02
0.015
0.01