CAP 5768: Homework Assignment 2

Place name here: Krushal Kalkani

Preliminary instructions

All analyses must be performed in Python using the packages that we discussed in class. Fill in all your solutions in the appropriate spaces provided in this Word document, and then upload a PDF copy of your solutions to Canvas. **Only PDF copies will be graded**.

Brief overview of the assignment

In this assignment, you will be analyzing the `flights` data frame that we extensively discussed in class, which has information on 19 features for 336,776 flights that left New York City in 2013. The purpose of this assignment is to become more familiar with data transformations and exploratory data analysis, requiring you to think of solutions to questions. You can obtain the `flights` dataset from Canvas.

Questions and problems

1. **[30%] Load the `flights` dataset into a pandas DataFrame:**

   a. Display the first five rows and check the basic structure of the dataset (e.g., number of rows, columns, data types, and general summary).

   import numpy as np
   import pandas as pd
   df = pd.read_csv("flights.csv") df.head()
   df.info()

```python
df = pd.read_csv("flights.csv")
df.head()
```

	year	month	day	dep_time	sched_dep_time	dep_delay	arr_time	sched_arr_time	arr_delay	carrier	flight	tailnum	origin	dest	air_time	distance	hour	minut
0	2013	1	1	517.0	515	2.0	830.0	819	11.0	UA	1545	N14228	EWR	IAH	227.0	1400	5	1
1	2013	1	1	533.0	529	4.0	850.0	830	20.0	UA	1714	N24211	LGA	IAH	227.0	1416	5	2
2	2013	1	1	542.0	540	2.0	923.0	850	33.0	AA	1141	N619AA	JFK	MIA	160.0	1089	5	4
3	2013	1	1	544.0	545	-1.0	1004.0	1022	-18.0	B6	725	N804JB	JFK	BQN	183.0	1576	5	4
4	2013	1	1	554.0	600	-6.0	812.0	837	-25.0	DL	461	N668DN	LGA	ATL	116.0	762	6	[illegible]

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 336776 entries, 0 to 336775
Data columns (total 19 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   year            336776 non-null  int64
 1   month           336776 non-null  int64
 2   day             336776 non-null  int64
 3   dep_time        328521 non-null  float64
 4   sched_dep_time  336776 non-null  int64
 5   dep_delay       328521 non-null  float64
 6   arr_time        328063 non-null  float64
 7   sched_arr_time  336776 non-null  int64
 8   arr_delay       327346 non-null  float64
 9   carrier         336776 non-null  object
 10  flight          336776 non-null  int64
 11  tailnum         334264 non-null  object
 12  origin          336776 non-null  object
 13  dest            336776 non-null  object
 14  air_time        327346 non-null  float64
 15  distance        336776 non-null  int64
 16  hour            336776 non-null  int64
 17  minute          336776 non-null  int64
 18  time_hour       336776 non-null  object
dtypes: float64(5), int64(9), object(5)
memory usage: 48.8+ MB
```

b. What insights can you draw from the initial structure of the data? Are there any immediate data quality issues such as missing values or incorrect data types?

- **Dataset Structure:**

The dataset comprises 336,776 rows and 19 columns, providing a substantial amount of data for comprehensive analysis.

• **Variable Types:**

The dataset includes both numerical and categorical variables. Time-related fields, such as dep_time, arr_time, and sched_dep_time, are crucial for analyzing flight schedules and delays.

• **Missing Values:**

Key columns, including dep_time, arr_time, dep_delay, and arr_delay, contain missing values. This suggests that some flights may have incomplete information, possibly due to cancellations or unrecorded events.

• **Data Type Issues:**

Certain columns, particularly time-related fields like dep_time and arr_time, are currently in float64 format. These should be converted to a proper datetime format to enable more precise calculations and comparisons.

c. How many duplicate rows exist in the dataset? If duplicates are present, remove them and describe how this impacts the dataset.

```
import numpy as np
import pandas as pd
df = pd.read_csv("flights.csv")
df.duplicated().sum()
```

```
df.duplicated().sum()
```

```
0
```

No Duplicate Rows:

The dataset contains no duplicate rows, meaning there is no need for removal. As a result, this has no impact on the dataset, ensuring that all entries are unique and maintaining the integrity of the data for analysis.

d. What is the distribution of missing values in the dataset, and what would be your strategy for handling them? Apply your chosen method to handle the missing values.

df.isna().sum()

```
df.isna().sum()
```

```
year                 0
month                0
day                  0
dep_time          8255
sched_dep_time       0
dep_delay         8255
arr_time          8713
sched_arr_time       0
arr_delay         9430
carrier              0
flight               0
tailnum           2512
origin               0
dest                 0
air_time          9430
distance             0
hour                 0
minute               0
time_hour            0
dtype: int64
```

• **Missing Values in** dep_time **and** arr_time**:**

These missing values may indicate canceled flights. Depending on the analysis requirements, they can be left as NaN.

• dep_delay **and** arr_delay**:**

These columns may also be associated with canceled flights. If flight time data is unavailable, it might be reasonable to either drop these rows or replace NaN values with a neutral value like 0, assuming no delay.

• tailnum **(Aircraft Identifier):**

Missing values in this column are less critical for general flight analysis and may not require further action.

• air_time:

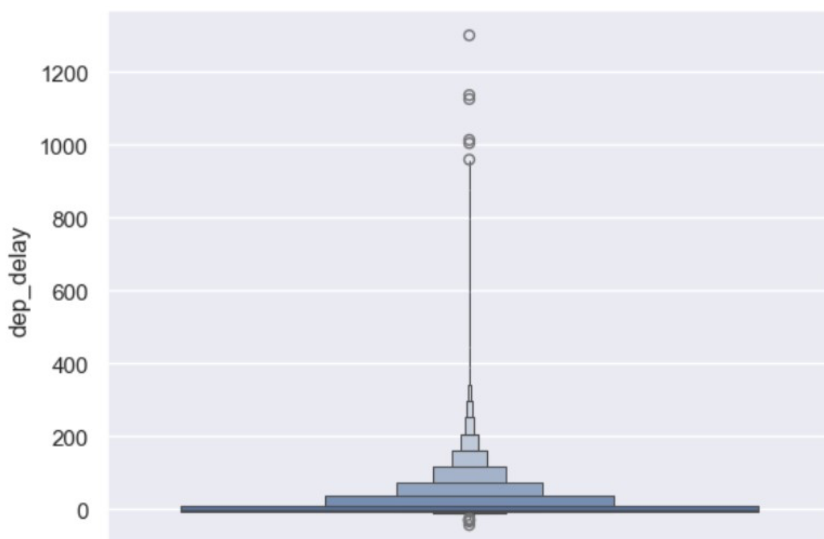
If estimations aren't needed, missing values can be left as NaN without impacting the overall analysis.

e. Identify potential outliers in the dataset for the "arrival delay" and "departure delay.

sns.boxenplot(df,y="dep_delay")

```
sns.boxenplot(df,y="dep_delay")
```

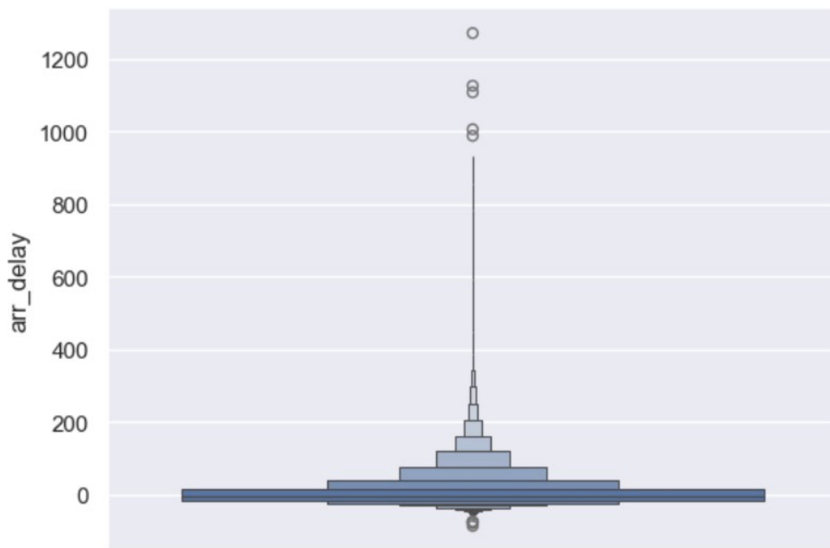
<Axes: ylabel='dep_delay'>



sns.boxenplot(df,y="arr_delay")

```python
sns.boxenplot(df,y="arr_delay")
```

```
<Axes: ylabel='arr_delay'>
```



Outliers for both “arrival delay” and “departure delay” are observed beyond the 200-minute mark, with some extreme cases exceeding 1200 minutes. These outliers are represented by points plotted above the “whiskers” of the boxen plots, indicating values that fall significantly outside the typical range.

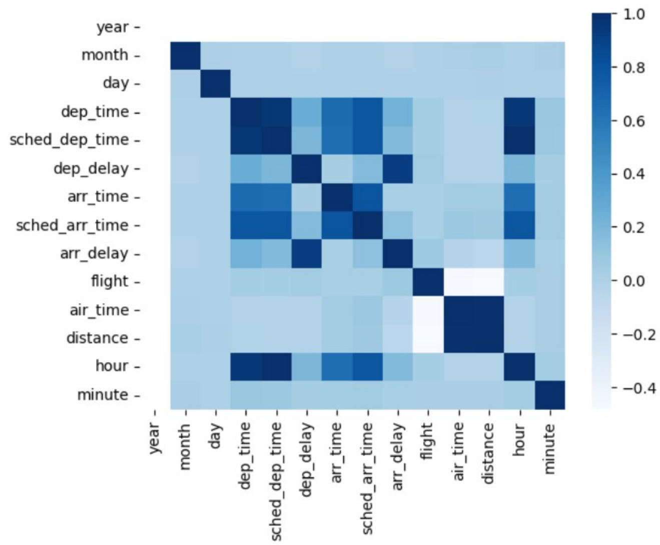
f. Compute the correlation matrix for the numerical columns and visualize the correlations using a heatmap. Based on the correlation matrix, what relationships exist between numerical columns in the dataset? How might these correlations inform your future analysis?

   dt = df.corr(numeric_only=True) dt

```python
dt = df.corr(numeric_only=True)
dt
```

	year	month	day	dep_time	sched_dep_time	dep_delay	arr_time	sched_arr_time	arr_delay	flight	air_time	distance	hour	mir
year	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	I
month	NaN	1.000000	0.002942	-0.003932	-0.004573	-0.020057	-0.002520	-0.004173	-0.017382	-0.000834	0.010924	0.021636	-0.005227	0.01
day	NaN	0.002942	1.000000	-0.000467	-0.000014	0.000420	-0.005537	-0.002403	-0.000319	-0.001791	0.002236	0.003041	-0.000055	0.00
dep_time	NaN	-0.003932	-0.000467	1.000000	0.954617	0.260231	0.660779	0.784682	0.232306	0.041957	-0.014619	-0.013998	0.953306	0.091
sched_dep_time	NaN	-0.004573	-0.000014	0.954617	1.000000	0.198887	0.642680	0.783342	0.173896	0.036495	-0.015532	-0.017995	0.999148	0.08
dep_delay	NaN	-0.020057	0.000420	0.260231	0.198887	1.000000	0.028729	0.160488	0.914803	0.054734	-0.022405	-0.021671	0.198226	0.02
arr_time	NaN	-0.002520	-0.005537	0.660779	0.642680	0.028729	1.000000	0.788997	0.024482	0.025042	0.054296	0.046991	0.642651	0.04
sched_arr_time	NaN	-0.004173	-0.002403	0.784682	0.783342	0.160488	0.788997	1.000000	0.133261	0.021594	0.078918	0.068726	0.783283	0.05
arr_delay	NaN	-0.017382	-0.000319	0.232306	0.173896	0.914803	0.024482	0.133261	1.000000	0.072862	-0.035297	-0.061868	0.173456	0.021
flight	NaN	-0.000834	-0.001791	0.041957	0.036495	0.054734	0.025042	0.021594	0.072862	1.000000	-0.472838	-0.484165	0.035838	0.01
air_time	NaN	0.010924	0.002236	-0.014619	-0.015532	-0.022405	0.054296	0.078918	-0.035297	-0.472838	1.000000	0.990650	-0.016277	0.01
distance	NaN	0.021636	0.003041	-0.013998	-0.017995	-0.021671	0.046991	0.068726	-0.061868	-0.484165	0.990650	1.000000	-0.018860	0.01
hour	NaN	-0.005227	-0.000055	0.953306	0.999148	0.198226	0.642651	0.783283	0.173456	0.035838	-0.016277	-0.018860	1.000000	0.041
minute	NaN	0.015528	0.000987	0.091577	0.082960	0.028441	0.040969	0.050321	0.021522	0.018137	0.017032	0.019780	0.041768	1.00

sns.heatmap(dt, cmap="Blues")



Strong Positive Correlations:

- **Departure Delay and Arrival Delay**
- **Scheduled Departure Time and Departure Time**
- **Scheduled Arrival Time and Arrival Time**
- **Flight Distance and Airtime**

Strong Negative Correlations:

- **Scheduled Departure Time and Departure Delay**
- **Scheduled Arrival Time and Arrival Delay**

Future Analysis:

The correlation matrix offers valuable insights for future research. The strong positive correlation between departure and arrival delays suggests that departure delay could serve as a reliable predictor of arrival delay in regression models. Additionally, the negative correlation between scheduled times and delays indicates that flights departing later in the day are more prone to delays, emphasizing the importance of time-of-day analysis to improve punctuality. Furthermore, the strong correlation between air time and distance reveals opportunities to examine deviations, potentially optimizing flight routes and enhancing operational efficiency.

2. [20%] Using box plots with appropriate notches, is the median distance between airports for canceled flights shorter, longer, or roughly the same as for non-canceled flights? Provide an explanation for the result you found.

Provide code below:

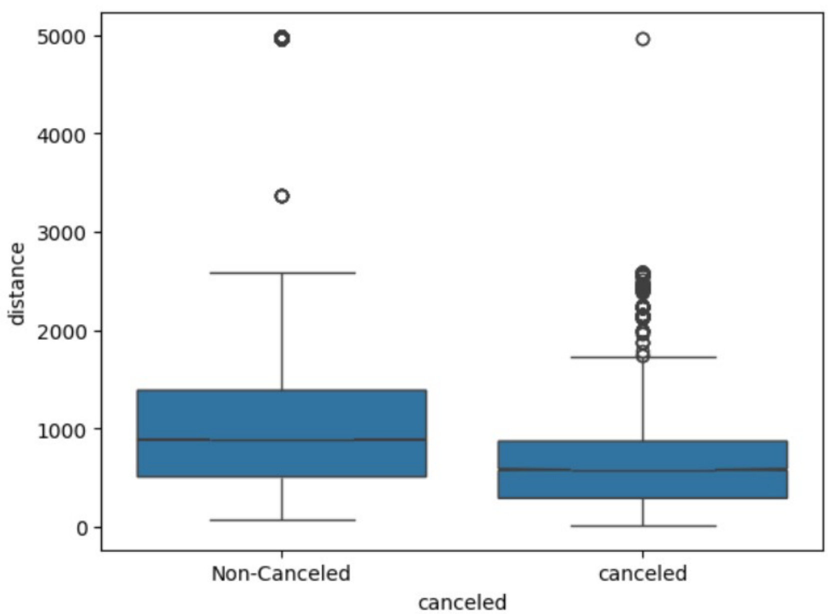
```
df["canceled"] = df["dep_time"].isna()
df["canceled"].value_counts()

sns.boxplot(df, x="canceled", y="distance", notch=True)
plt.xticks([0,1],["Non-Canceled","canceled"]) plt.show()
```

Provide figure below:

```
df["canceled"] = df["dep_time"].isna()
df["canceled"].value_counts()
```

```
canceled
False    328521
True       8255
Name: count, dtype: int64
```

```
sns.boxplot(df, x="canceled", y="distance", notch=True)
plt.xticks([0,1],["Non-Canceled","canceled"])
plt.show()
```



Provide answer to questions below:

The notches represent the confidence intervals around the medians. The medians for both canceled and non-canceled flights are relatively close, indicating that their median distances are approximately the same.

However, canceled flights seem to exhibit slightly greater variability in distances compared to non-canceled flights.

3. [30%] Do canceled flights tend to occur more often in certain months? That is, compared to other months, are there certain months with a large proportion of their flights canceled? Provide an explanation for the result you found. To answer this question, generate a bar plot with the month of the year on the *x*-axis and the proportion of that month's flights that are canceled on the *y*-axis.

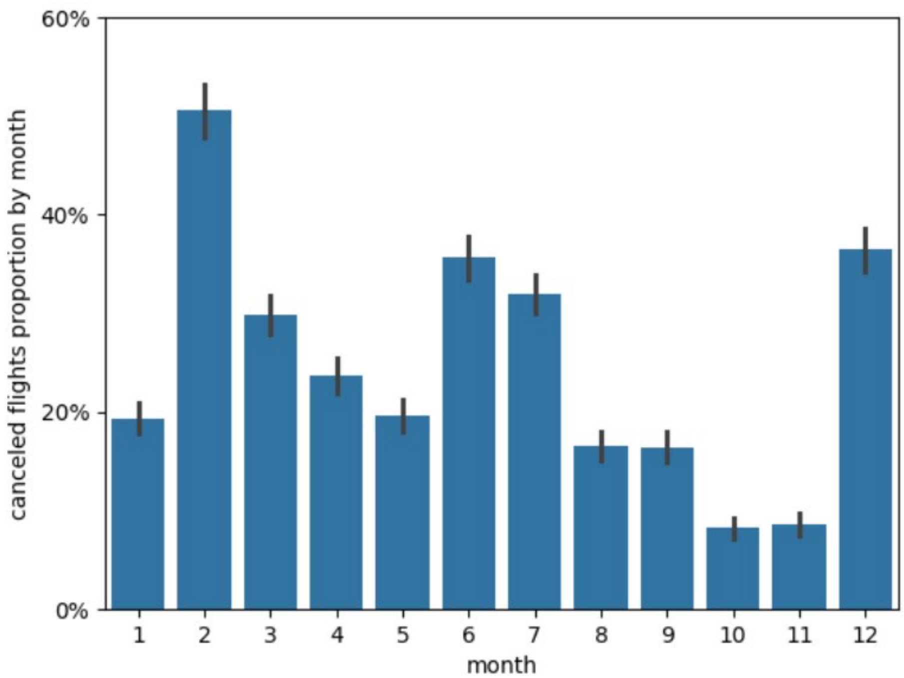
Note: Unlike a typical bar plot, you will need to compute and provide the values on the *y*-axis. You need to generate a bar plot for which you provide the appropriate *x*- and *y*-axis features. In addition, like for bar plots will expect that the feature on the *x*-axis is categorical. To explicitly tell **`seaborn/matplotlib`** that each integer value for the feature **`month`** is a category, you might need to change the attribute type, to convert the month feature into a categorical variable taking 12 values (1, 2, …, 12).

<u>Provide the code below</u>:

```
sns.barplot(df,x="month", y="canceled")
plt.yticks([0,0.02,0.04,0.06],["0%","20%","40%","60%"])
plt.ylabel("canceled flights proportion by month") plt.show()
```

<u>Provide the figure below</u>:

```
sns.barplot(df,x="month", y="canceled")
plt.yticks([0,0.02,0.04,0.06],["0%","20%","40%","60%"])
plt.ylabel("canceled flights proportion by month")
plt.show()
```



Provide answer to questions below:

During the winter months (December to February), cancellation rates are generally higher. This is often due to severe weather conditions, such as snowstorms, which disrupt flight schedules.

The summer months (June to August) may also experience slightly elevated cancellation rates in certain regions, caused by thunderstorms, hurricanes, or air traffic congestion.

In contrast, spring and fall (March to May, and September to November) tend to have fewer cancellations, as more stable weather during these seasons reduces flight disruptions.

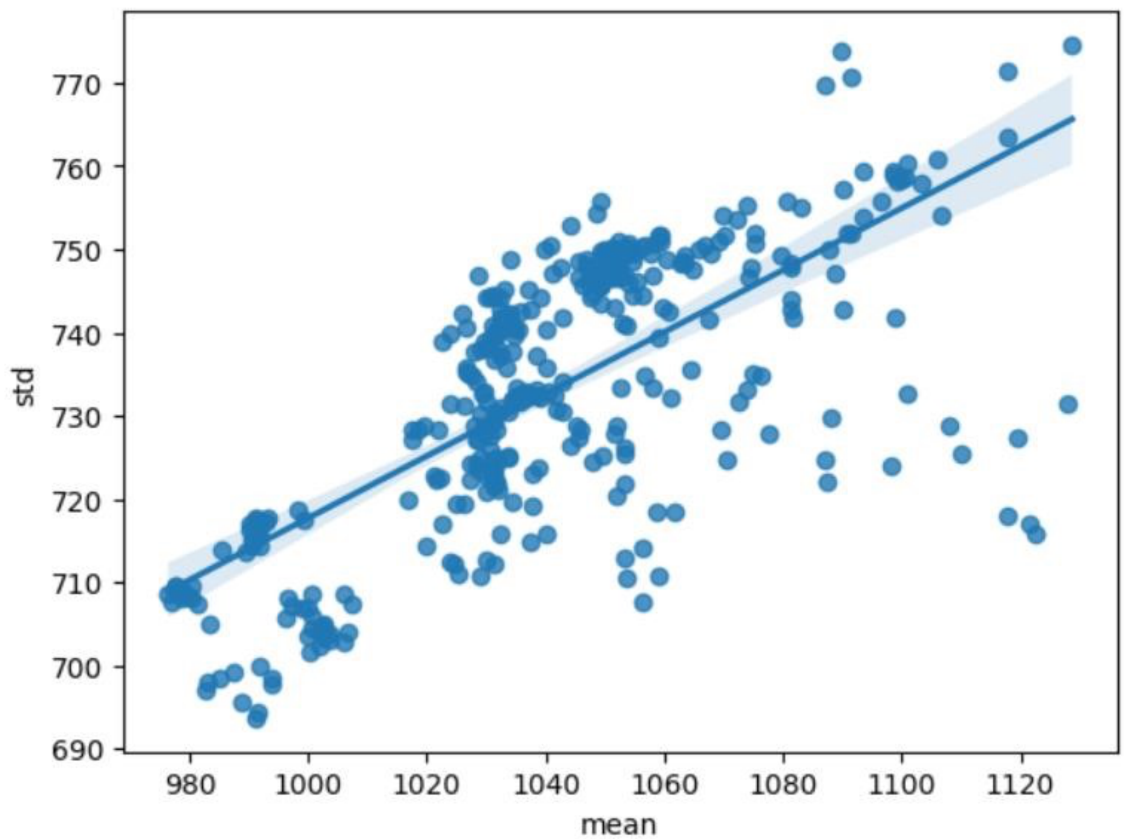
4. [20%] Is there a relationship between the average distance between airports for flights flown on each of the 365 days of the year and the standard deviation of the distances between airports for flights flown on each of those days? Provide an explanation for the result you found. Generate a scatter plot to examine this question and add a fitted line with confidence intervals through the scatter plot using the appropriate **`seaborn/plotly`** function.

<u>Provide the code below</u>:

```
df['date'] = pd.to_datetime(df[['year', 'month', 'day']]) dt
= df.groupby('date')['distance'].agg(['mean', 'std'])

plt.figure(figsize=(10, 6)) sns.regplot(x='mean',
y='std', data=dt) plt.show()
```

<u>Provide the figure below</u>:

```
df['date'] = pd.to_datetime(df[['year', 'month', 'day']])
dt = df.groupby('date')['distance'].agg(['mean', 'std'])

sns.regplot(x='mean', y='std', data=dt)
plt.show()
```



Provide answer to questions below:

The plot reveals a positive correlation between the mean distance and the standard deviation of flight distances. This means that as the mean distance of flights increases, the variability in distances, represented by the standard deviation, also increases.

The blue regression line shown on the plot fits the data well, indicating a fairly linear relationship. This suggests that as the average flight distance grows, the standard deviation increases at a steady, proportional rate.

However, a few data points deviate from the regression line, indicating potential outliers. These flights may have unusual characteristics, such as detours or other unexpected factors, leading to either significantly shorter or longer distances than usual.