

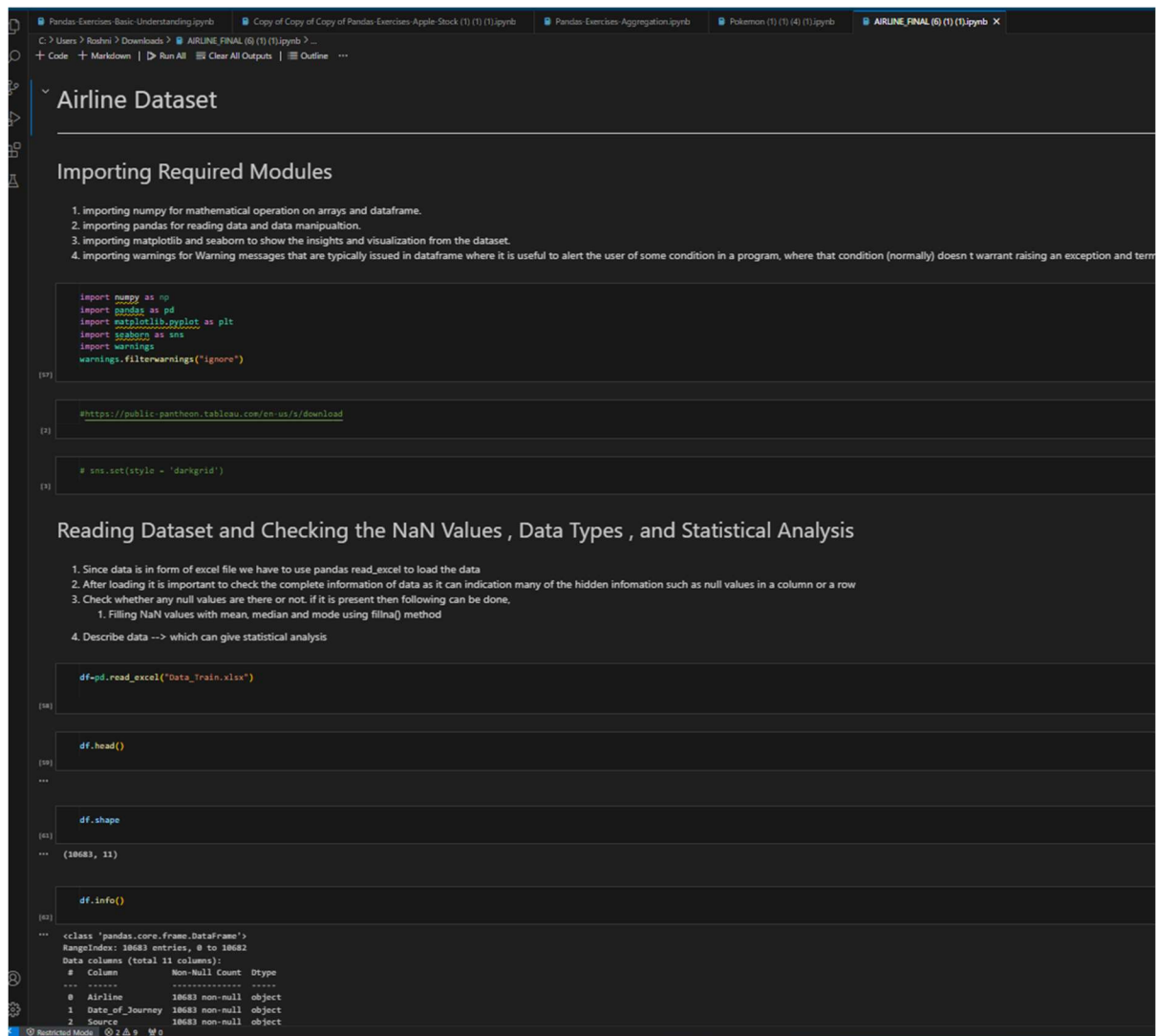
26-11-2024

Training Day – 50

November 26, Tuesday

- *Topic:* Final Review and Practice

- Revisited core concepts and practiced integrating analysis and visualization.
- Example: Created a summary report of the entire analysis workflow.



```
Pandas Exercises: Basic Understanding.ipynb | Copy of Copy of Copy of Pandas Exercises: Apple Stock (1) (1) (1).ipynb | Pandas Exercises: Aggregation.ipynb | Pokemon (1) (1) (4) (1).ipynb | AIRLINE_FINAL (6) (1) (1).ipynb X
C:\Users\> Rstudio\> Downloads > AIRLINE_FINAL (6) (1) (1).ipynb > ...
+ Code + Markdown | ▶ Run All | Clear All Outputs | Outline ...

Airline Dataset

Importing Required Modules

1. importing numpy for mathematical operation on arrays and dataframe.
2. importing pandas for reading data and data manipulation.
3. importing matplotlib and seaborn to show the insights and visualization from the dataset.
4. importing warnings for Warning messages that are typically issued in dataframe where it is useful to alert the user of some condition in a program, where that condition (normally) doesn't warrant raising an exception and term

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")

[07]

https://public.pantheon.tableau.com/en-us/s/download

[2]

sns.set(style = 'darkgrid')

[3]

Reading Dataset and Checking the NaN Values , Data Types , and Statistical Analysis

1. Since data is in form of excel file we have to use pandas read_excel to load the data
2. After loading it is important to check the complete information of data as it can indicate many of the hidden information such as null values in a column or a row
3. Check whether any null values are there or not. If it is present then following can be done.
    1. Filling NaN values with mean, median and mode using fillna() method
4. Describe data --> which can give statistical analysis

df=pd.read_excel("Data_train.xlsx")

[08]

df.head()

[09]

...

df.shape

[02]

...
(10683, 11)

df.info()

[03]

...
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Column                10683 non-null object
1   Airline               10683 non-null object
2   Date_of_Journey       10683 non-null object
3   Source               10683 non-null object
```

```
Pandas-Exercises-Basic-Understanding.ipynb | Copy of Copy of Copy of Pandas-Exercises-Apple-Stock (1) (1) (1).ipynb | Pandas-Exercises-Aggregation.ipynb | Pokemon (1) (1) (4) (1).ipynb | AIRL
C:\Users\Roshni\Downloads> AIRLINE_FINAL (6) (1) (1).ipynb > ...
+ Code + Markdown | Run All | Clear All Outputs | Outline ...

[63] df.describe()
...

[64] df.describe(include=object)
...

[65] df.isnull().sum()
...
Airline      0
Date_of_Journey  0
Source      0
Destination  0
Route       1
Dep_Time    0
Arrival_Time 0
Duration    0
Total_Stops  1
Additional_Info 0
Price       0
dtype: int64

[66] df['Route'].mode()
...
0    DEL -> BOM -> COK
Name: Route, dtype: object

[67] df['Route']=df['Route'].fillna(df['Route'].mode()[0])

[68] df['Total_Stops'].mode()
...
0    1 stop
Name: Total_Stops, dtype: object

df['Total_Stops']=df['Total_Stops'].fillna(df['Total_Stops'].mode()[0])
```

```
Pandas-Exercises-Basic-Understanding.ipynb Copy of Copy of Copy of Pandas-Exercises-Apple-Stock (1) (1).ipynb Pandas-Exercises-Aggregation.ipynb Pokemon (1) (1) (4) (1).ipynb AIRLINE_FINAL

C:\Users> Roshni> Downloads> AIRLINE_FINAL (6) (1) (1).ipynb> ...
+ Code + Markdown | Run All | Clear All Outputs | Outline ...

From df.info() we can see that Date_of_Journey is a object data type

1. Therefore, we have to convert this datatype into timestamp so that we can use that column properly to find the insights.
2. For this we require pandas to_datetime to convert object data type to datetime dtype.

df['Date_of_Journey']=pd.to_datetime(df['Date_of_Journey'])

[71]

df.head(2)

[72]

...

df.info()

[73]

...
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 11 columns):
# Column Non-Null Count Dtype
---
0 Airline 10683 non-null object
1 Date_of_Journey 10683 non-null datetime64[ns]
2 Source 10683 non-null object
3 Destination 10683 non-null object
4 Route 10683 non-null object
5 Dep_Time 10683 non-null object
6 Arrival_Time 10683 non-null object
7 Duration 10683 non-null object
8 Total_Stops 10683 non-null object
9 Additional_Info 10683 non-null object
10 Price 10683 non-null int64
dtypes: datetime64[ns](1), int64(1), object(9)
memory usage: 918.2+ KB

df['Total_Stops'].unique()

[74]

...
array(['non-stop', '2 stops', '1 stop', '3 stops', '4 stops'],
      dtype=object)
```

