Aryaman Mishra

19BCE1027

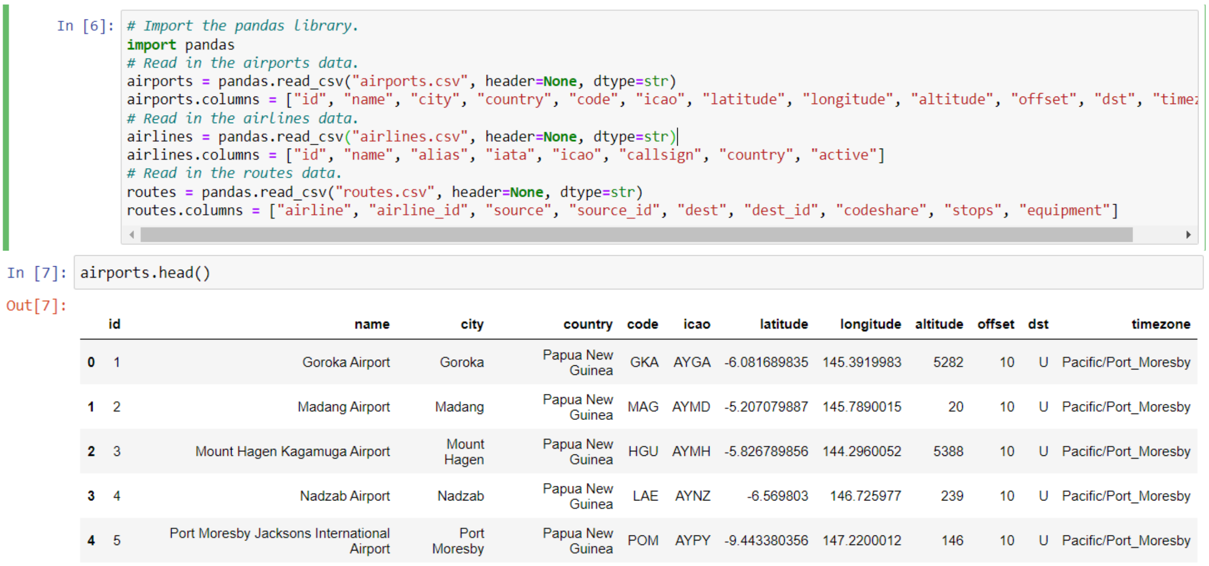
Exercises

Pre-requisite: We will assume you are moderately familiar with basic concepts in Python

Dataset : Airline, Airport and Route datasets

Reference : (https://www.dataquest.io/blog/python-data-visualization-libraries/)

Pre Processing : Assign Column Headers to the given datasets

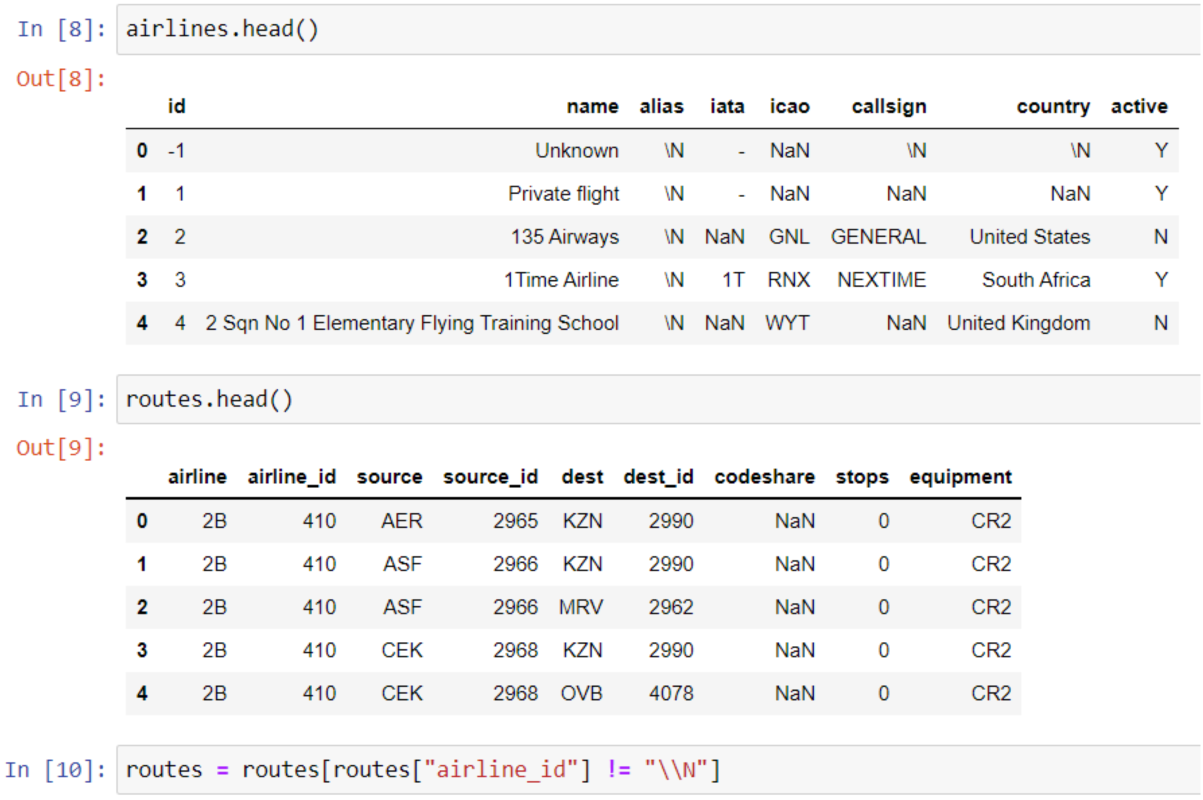


In [6]:
```
# Import the pandas library.
import pandas
# Read in the airports data.
airports = pandas.read_csv("airports.csv", header=None, dtype=str)
airports.columns = ["id", "name", "city", "country", "code", "icao", "latitude", "longitude", "altitude", "offset", "dst", "time
# Read in the airlines data.
airlines = pandas.read_csv("airlines.csv", header=None, dtype=str)
airlines.columns = ["id", "name", "alias", "iata", "icao", "callsign", "country", "active"]
# Read in the routes data.
routes = pandas.read_csv("routes.csv", header=None, dtype=str)
routes.columns = ["airline", "airline_id", "source", "source_id", "dest", "dest_id", "codeshare", "stops", "equipment"]
```

In [7]:
```
airports.head()
```

Out[7]:

	id	name	city	country	code	icao	latitude	longitude	altitude	offset	dst	timezone
0	1	Goroka Airport	Goroka	Papua New Guinea	GKA	AYGA	-6.081689835	145.3919983	5282	10	U	Pacific/Port_Moresby
1	2	Madang Airport	Madang	Papua New Guinea	MAG	AYMD	-5.207079887	145.7890015	20	10	U	Pacific/Port_Moresby
2	3	Mount Hagen Kagamuga Airport	Mount Hagen	Papua New Guinea	HGU	AYMH	-5.826789856	144.2960052	5388	10	U	Pacific/Port_Moresby
3	4	Nadzab Airport	Nadzab	Papua New Guinea	LAE	AYNZ	-6.569803	146.725977	239	10	U	Pacific/Port_Moresby
4	5	Port Moresby Jacksons International Airport	Port Moresby	Papua New Guinea	POM	AYPY	-9.443380356	147.2200012	146	10	U	Pacific/Port_Moresby



In [8]:
```
airlines.head()
```

Out[8]:

	id	name	alias	iata	icao	callsign	country	active
0	-1	Unknown	\N	-	NaN	\N	\N	Y
1	1	Private flight	\N	-	NaN	NaN	NaN	Y
2	2	135 Airways	\N	NaN	GNL	GENERAL	United States	N
3	3	1Time Airline	\N	1T	RNX	NEXTIME	South Africa	Y
4	4	2 Sqn No 1 Elementary Flying Training School	\N	NaN	WYT	NaN	United Kingdom	N

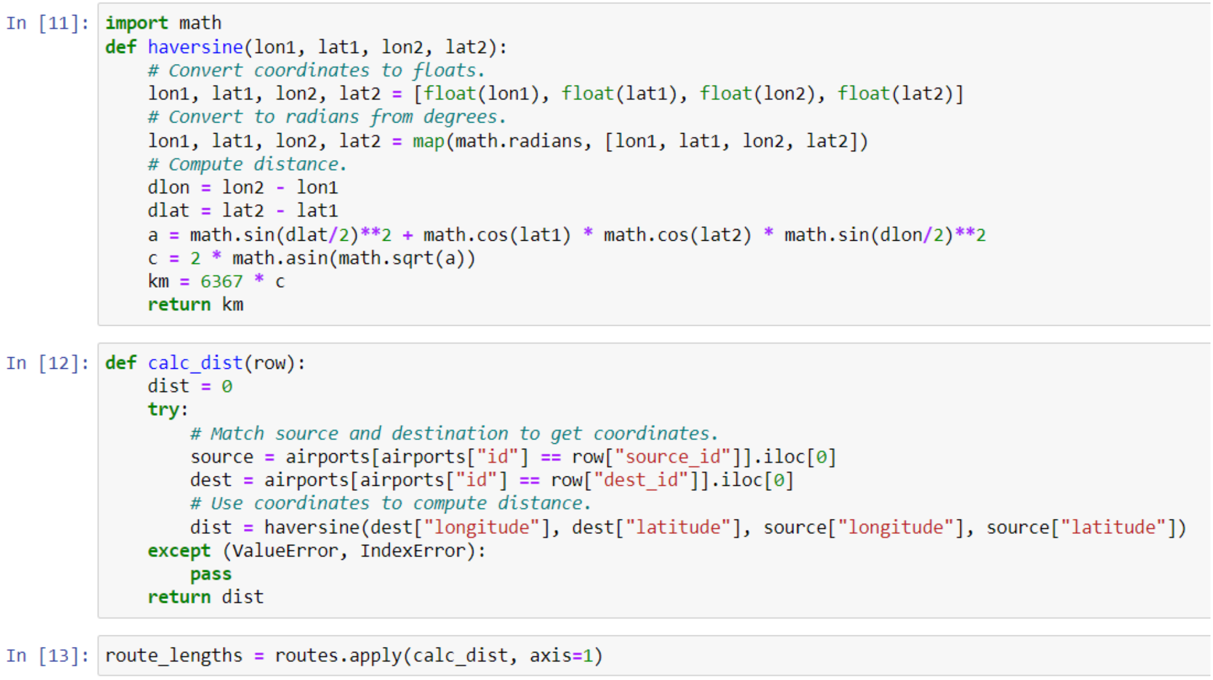
In [9]:
```
routes.head()
```

Out[9]:

	airline	airline_id	source	source_id	dest	dest_id	codeshare	stops	equipment
0	2B	410	AER	2965	KZN	2990	NaN	0	CR2
1	2B	410	ASF	2966	KZN	2990	NaN	0	CR2
2	2B	410	ASF	2966	MRV	2962	NaN	0	CR2
3	2B	410	CEK	2968	KZN	2990	NaN	0	CR2
4	2B	410	CEK	2968	OVB	4078	NaN	0	CR2

In [10]:
```
routes = routes[routes["airline_id"] != "\\N"]
```

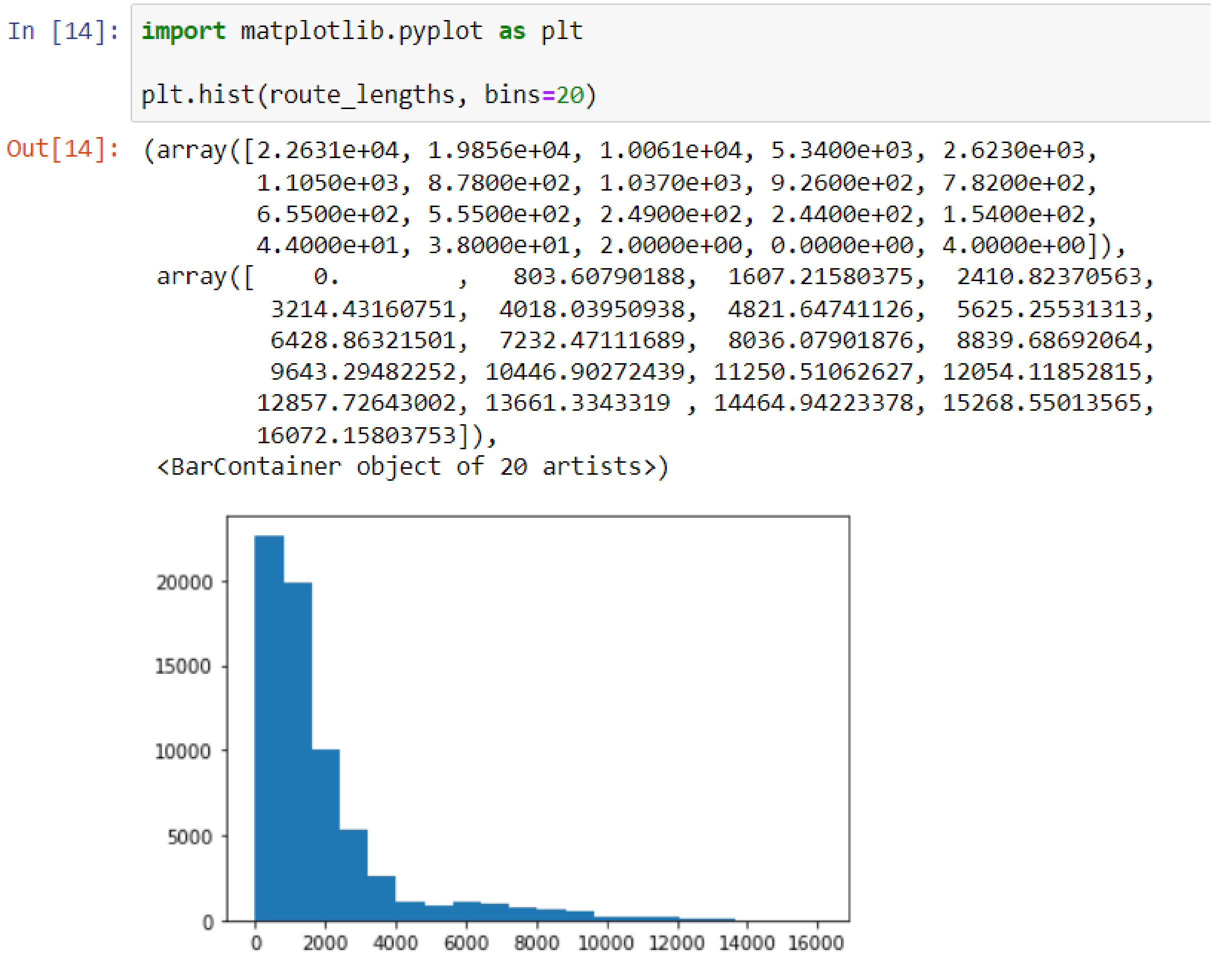
Make histogram for route length, bin the values into ranges and count how many routes fall into each range



```
In [11]: import math
         def haversine(lon1, lat1, lon2, lat2):
             # Convert coordinates to floats.
             lon1, lat1, lon2, lat2 = [float(lon1), float(lat1), float(lon2), float(lat2)]
             # Convert to radians from degrees.
             lon1, lat1, lon2, lat2 = map(math.radians, [lon1, lat1, lon2, lat2])
             # Compute distance.
             dlon = lon2 - lon1
             dlat = lat2 - lat1
             a = math.sin(dlat/2)**2 + math.cos(lat1) * math.cos(lat2) * math.sin(dlon/2)**2
             c = 2 * math.asin(math.sqrt(a))
             km = 6367 * c
             return km
```

```
In [12]: def calc_dist(row):
             dist = 0
             try:
                 # Match source and destination to get coordinates.
                 source = airports[airports["id"] == row["source_id"]].iloc[0]
                 dest = airports[airports["id"] == row["dest_id"]].iloc[0]
                 # Use coordinates to compute distance.
                 dist = haversine(dest["longitude"], dest["latitude"], source["longitude"], source["latitude"])
             except (ValueError, IndexError):
                 pass
             return dist
```

```
In [13]: route_lengths = routes.apply(calc_dist, axis=1)
```

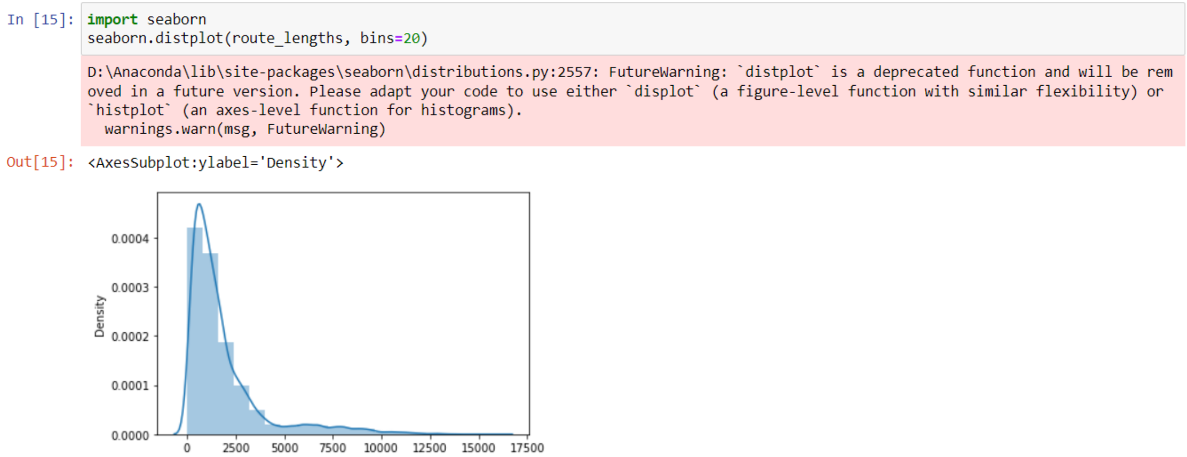


```
In [14]: import matplotlib.pyplot as plt

         plt.hist(route_lengths, bins=20)
```

```
Out[14]: (array([2.2631e+04, 1.9856e+04, 1.0061e+04, 5.3400e+03, 2.6230e+03,
                 1.1050e+03, 8.7800e+02, 1.0370e+03, 9.2600e+02, 7.8200e+02,
                 6.5500e+02, 5.5500e+02, 2.4900e+02, 2.4400e+02, 1.5400e+02,
                 4.4000e+01, 3.8000e+01, 2.0000e+00, 0.0000e+00, 4.0000e+00]),
          array([    0.        ,   803.60790188,  1607.21580375,  2410.82370563,
                  3214.43160751,  4018.03950938,  4821.64741126,  5625.25531313,
                  6428.86321501,  7232.47111689,  8036.07901876,  8839.68692064,
                  9643.29482252, 10446.90272439, 11250.51062627, 12054.11852815,
                 12857.72643002, 13661.3343319 , 14464.94223378, 15268.55013565,
                 16072.15803753]),
          <BarContainer object of 20 artists>)
```

Use seaborn for route dataset (route Length)

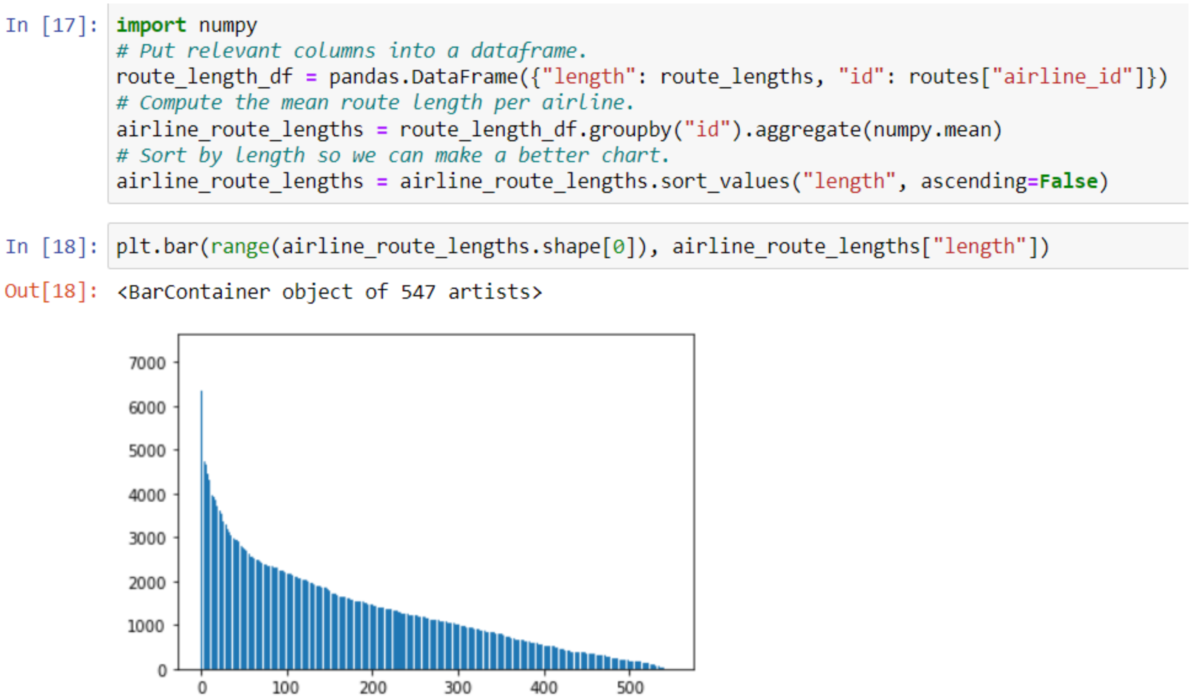


```
In [15]: import seaborn
         seaborn.distplot(route_lengths, bins=20)

D:\Anaconda\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be rem
oved in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or
`histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)

Out[15]: <AxesSubplot:ylabel='Density'>
```

Bar chart - plot each airline against the average route length each airline flies

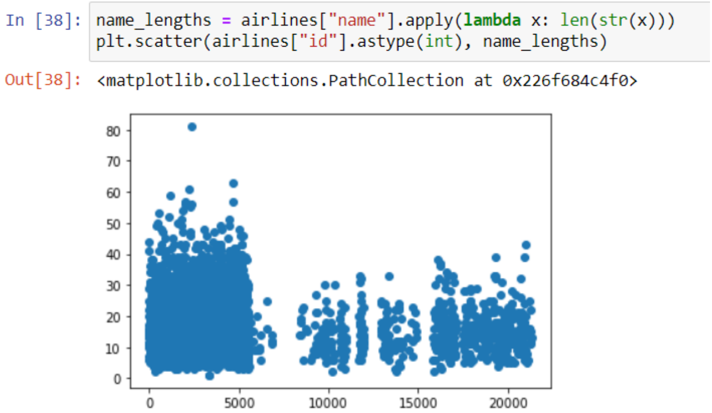


```
In [17]: import numpy
         # Put relevant columns into a dataframe.
         route_length_df = pandas.DataFrame({"length": route_lengths, "id": routes["airline_id"]})
         # Compute the mean route length per airline.
         airline_route_lengths = route_length_df.groupby("id").aggregate(numpy.mean)
         # Sort by length so we can make a better chart.
         airline_route_lengths = airline_route_lengths.sort_values("length", ascending=False)

In [18]: plt.bar(range(airline_route_lengths.shape[0]), airline_route_lengths["length"])

Out[18]: <BarContainer object of 547 artists>
```

Create a scatter plot comparing the airline ids to the name lengths



```
In [38]: name_lengths = airlines["name"].apply(lambda x: len(str(x)))
         plt.scatter(airlines["id"].astype(int), name_lengths)

Out[38]: <matplotlib.collections.PathCollection at 0x226f684c4f0>
```