Datasets Description:

airports.dat:

Airport ID	Unique OpenFlights identifier for this airport.
Name	Name of airport. May or may not contain the **City** name.
City	Main city served by airport. May be spelled differently from **Name**.
Country	Country or territory where airport is located. See countries.dat to cross-reference to ISO 3166-1 codes.
IATA	3-letter IATA code. Null if not assigned/unknown.
ICAO	4-letter ICAO code. Null if not assigned.
Latitude	Decimal degrees, usually to six significant digits. Negative is South, positive is North.
Longitude	Decimal degrees, usually to six significant digits. Negative is West, positive is East.
Altitude	In feet.
Timezone	Hours offset from UTC. Fractional hours are expressed as decimals, eg. India is 5.5.
DST	Daylight savings time. One of E (Europe), A (US/Canada), S (South America), O (Australia), Z (New Zealand), N (None) or U (Unknown). *See also: Help: Time*
Tz database time zone	Timezone in "tz" (Olson) format, eg. "America/Los_Angeles".
Type	Type of the airport. Value "airport" for air terminals, "station" for train stations, "port" for ferry terminals and "unknown" if not known. *In airports.csv, only type=airport is included.*
Source	Source of this data. "OurAirports" for data sourced from OurAirports, "Legacy" for old data not matched to OurAirports (mostly DAFIF), "User" for unverified user contributions. *In airports.csv, only source=OurAirports is included.*

Sample entries:

```
507,"London Heathrow Airport","London","United Kingdom","LHR","EGLL",51.4706,-
0.461941,83,0,"E","Europe/London","airport","OurAirports"
26,"Kugaaruk Airport","Pelly Bay","Canada","YBB","CYBB",68.534401,-89.808098,56,-
7,"A","America/Edmonton","airport","OurAirports"
3127,"Pokhara
Airport","Pokhara","Nepal","PKR","VNPK",28.200899124145508,83.98210144042969,2712,5
.75,"N","Asia/Katmandu","airport","OurAirports"
8810,"Hamburg
Hbf","Hamburg","Germany","ZMB",\N,53.552776,10.006683,30,1,"E","Europe/Berlin","sta
tion","User"
```

airlines.dat:

Airline ID	Unique OpenFlights identifier for this airline.
Name	Name of the airline.
Alias	Alias of the airline. For example, All Nippon Airways is commonly known as "ANA".
IATA	2-letter IATA code, if available.
ICAO	3-letter ICAO code, if available.
Callsign	Airline callsign.
Country	Country or territory where airline is incorporated.
Active	"Y" if the airline is or has until recently been operational, "N" if it is defunct. This field is *not* reliable: in particular, major airlines that stopped flying long ago, but have not had their IATA code reassigned (eg. Ansett/AN), will incorrectly show as "Y".

Sample entries:

```
324,"All Nippon Airways","ANA All Nippon Airways","NH","ANA","ALL
NIPPON","Japan","Y"
412,"Aerolineas Argentinas",\N,"AR","ARG","ARGENTINA","Argentina","Y"
413,"Arrowhead Airways",\N,"","ARH","ARROWHEAD","United States","N"
```

route.dat:

Airline	2-letter (IATA) or 3-letter (ICAO) code of the airline.
Airline ID	Unique OpenFlights identifier for airline (see Airline).
Source airport	3-letter (IATA) or 4-letter (ICAO) code of the source airport.
Source airport ID	Unique OpenFlights identifier for source airport (see Airport)
Destination airport	3-letter (IATA) or 4-letter (ICAO) code of the destination airport.
Destination airport ID	Unique OpenFlights identifier for destination airport (see Airport)
Codeshare	"Y" if this flight is a codeshare (that is, not operated by *Airline*, but another carrier), empty otherwise.
Stops	Number of stops on this flight ("0" for direct)
Equipment	3-letter codes for plane type(s) generally used on this flight, separated by spaces

Sample entries:

```
BA,1355,SIN,3316,LHR,507,,0,744 777
BA,1355,SIN,3316,MEL,3339,Y,0,744
TOM,5013,ACE,1055,BFS,465,,0,320
```

Reading the datasets:

```
import pandas as pd
airports = pd.read_csv("C:\\Users\\rameshragala\\Desktop\\DataVisualization\\Lab\\working\\airports.dat", header= None, dtype=str)
airports.columns = ["id", "name", "city", "country", "code", "icao", "latitude", "longitude", "altitude", "offset", "dst", "timezone", "Type", "Source"]

airlines = pd.read_csv("C:\\Users\\rameshragala\\Desktop\\DataVisualization\\Lab\\working\\airlines.dat", header= None, dtype=str)
airlines.columns = ["id", "name", "alias", "iata", "icao", "callsign", "country", "active"]

routes = pd.read_csv("C:\\Users\\rameshragala\\Desktop\\DataVisualization\\Lab\\working\\routes.dat", header= None, dtype=str)
routes.columns = ["airline", "airline_id", "source", "source_id", "dest", "dest_id", "codeshare", "stops", "equipment"]
```

bit of data cleaning:

```
routes = routes[routes["airline_id"] != "\\N"]
```

This line ensures that we have only numeric data in the `airline_id` column.

Problem:
Draw a histogram showing the distribution of route lengths by airlines.
Inference:
which airlines fly more shorter routes or more longer routes

Code for finding average route length:

```
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

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
route_lengths = routes.apply(calc_dist, axis=1)
```

Histogram Design:

```
import matplotlib.pyplot as plt
plt.hist(route_lengths, bins=20)
plt.show()
```

Exercise:

1. Identify the airports which are fly short routes using histogram
2. Draw a histogram showing the distribution of route lengths by airlines using ggplot in python.
3. Draw a histogram which fly short routes with airports names using ggplot in python