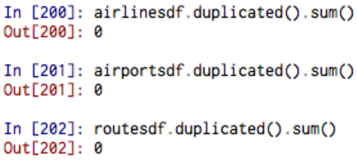
Anamitra Bhattacharyya
Predict 420-DL, Section 55
Assignment 2 (April 24, 2016)

1) Python code:

```
import pandas as pd #import panda
import numpy as np
#all files are in the current working directory

import cPickle as pickle

#read-in each pickle files into a separate panda dataframe
airlinesdf = pd.read_pickle('airlineslist.p')
airportsdf = pd.read_pickle('airportslist.p')
routesdf = pd.read_pickle('routeslist.p')

#Identify duplicates in airlinesdf, airportsdf and routesdf
airlinesdf.duplicated().sum()#returns a Boolean, 'True' if duplicated and 'False' if not
airportsdf.duplicated().sum()#for all DFs returns a sum=0, so no duplicates
routesdf.duplicated().sum()

#data types of all columns in each of the three dataframes
airlinesdf.dtypes
airportsdf.dtypes
routesdf.dtypes

#inspect first 10 indexes of all three dataframes
airlinesdf.ix[:10]
airportsdf.ix[:10]
routesdf.ix[:10]

#Number of defunct airlines (filter ‘active’ column in airlinesdf, find where its 'N')
print(airlinesdf.loc[airlinesdf['active']=="N"])

#Flights from nowhere. This code counts no. of blank entries in the srcAirport column
routesdf['srcAirport'].isnull().sum()#Zero flights from nowhere all return bool of'False'


#Pickling Airlines, Airports and Routes data frames again
import cPickle as pickle
pickle.dump(airlinesdf,open('airlineslist.p', 'wb'))
pickle.dump(airportsdf,open('airportslist.p', 'wb'))
pickle.dump(routesdf,open('routeslist.p', 'wb'))
```

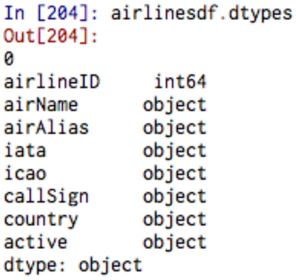
Output:

a) Duplicates



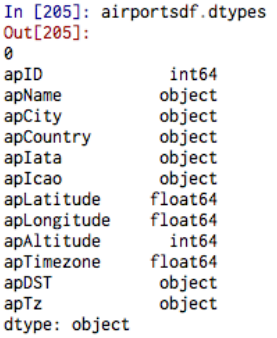
```
In [200]: airlinesdf.duplicated().sum()
Out[200]: 0

In [201]: airportsdf.duplicated().sum()
Out[201]: 0

In [202]: routesdf.duplicated().sum()
Out[202]: 0
```

b) airlinesdf data types



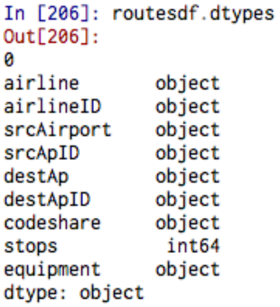
```
In [204]: airlinesdf.dtypes
Out[204]:
0
airlineID     int64
airName      object
airAlias     object
iata         object
icao         object
callSign     object
country      object
active       object
dtype: object
```

c) airportsdf data types



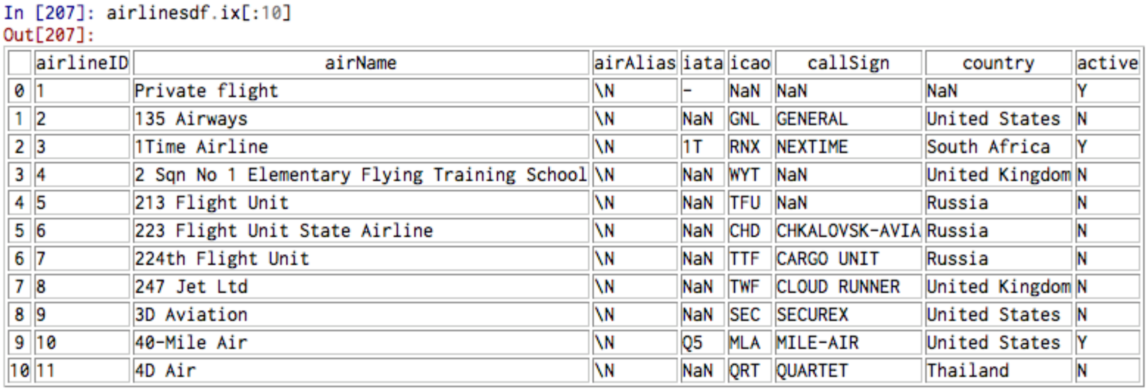
```
In [205]: airportsdf.dtypes
Out[205]:
0
apID              int64
apName           object
apCity           object
apCountry        object
apIata           object
apIcao           object
apLatitude      float64
apLongitude     float64
apAltitude        int64
apTimezone      float64
apDST            object
apTz             object
dtype: object
```

d) routesdf data types



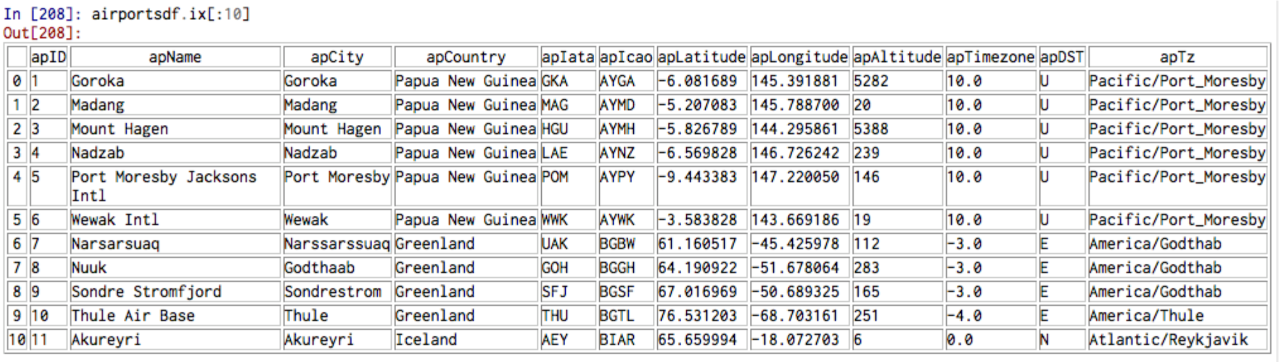
```
In [206]: routesdf.dtypes
Out[206]:
0
airline        object
airlineID      object
srcAirport     object
srcApID        object
destAp         object
destApID       object
codeshare      object
stops           int64
equipment      object
dtype: object
```

e) Inspect first 10 indexes of all three data frames



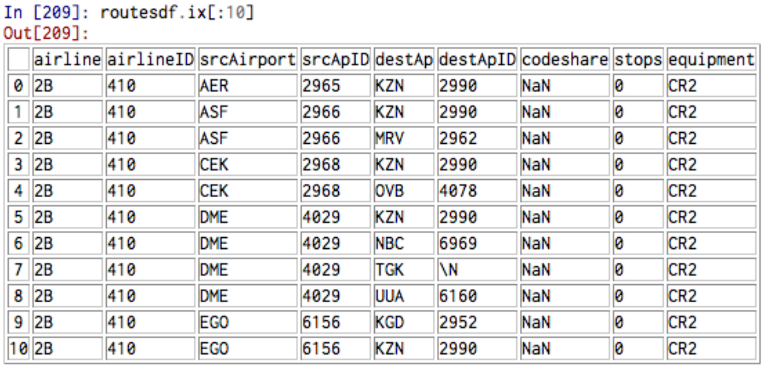
```
In [207]: airlinesdf.ix[:10]
Out[207]:
```

	airlineID	airName	airAlias	iata	icao	callSign	country	active
0	1	Private flight	\N	-	NaN	NaN	NaN	Y
1	2	135 Airways	\N	NaN	GNL	GENERAL	United States	N
2	3	1Time Airline	\N	1T	RNX	NEXTIME	South Africa	Y
3	4	2 Sqn No 1 Elementary Flying Training School	\N	NaN	WYT	NaN	United Kingdom	N
4	5	213 Flight Unit	\N	NaN	TFU	NaN	Russia	N
5	6	223 Flight Unit State Airline	\N	NaN	CHD	CHKALOVSK-AVIA	Russia	N
6	7	224th Flight Unit	\N	NaN	TTF	CARGO UNIT	Russia	N
7	8	247 Jet Ltd	\N	NaN	TWF	CLOUD RUNNER	United Kingdom	N
8	9	3D Aviation	\N	NaN	SEC	SECUREX	United States	N
9	10	40-Mile Air	\N	Q5	MLA	MILE-AIR	United States	Y
10	11	4D Air	\N	NaN	QRT	QUARTET	Thailand	N



```
In [208]: airportsdf.ix[:10]
Out[208]:
```

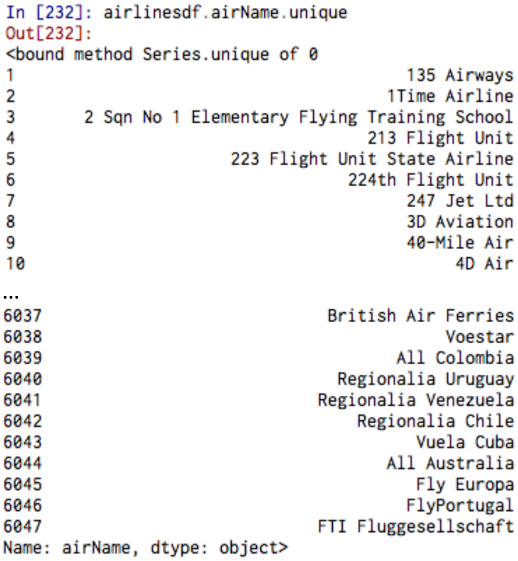
	apID	apName	apCity	apCountry	apIata	apIcao	apLatitude	apLongitude	apAltitude	apTimezone	apDST	apTz
0	1	Goroka	Goroka	Papua New Guinea	GKA	AYGA	-6.081689	145.391881	5282	10.0	U	Pacific/Port_Moresby
1	2	Madang	Madang	Papua New Guinea	MAG	AYMD	-5.207083	145.788700	20	10.0	U	Pacific/Port_Moresby
2	3	Mount Hagen	Mount Hagen	Papua New Guinea	HGU	AYMH	-5.826789	144.295861	5388	10.0	U	Pacific/Port_Moresby
3	4	Nadzab	Nadzab	Papua New Guinea	LAE	AYNZ	-6.569828	146.726242	239	10.0	U	Pacific/Port_Moresby
4	5	Port Moresby Jacksons Intl	Port Moresby	Papua New Guinea	POM	AYPY	-9.443383	147.220050	146	10.0	U	Pacific/Port_Moresby
5	6	Wewak Intl	Wewak	Papua New Guinea	WWK	AYWK	-3.583828	143.669186	19	10.0	U	Pacific/Port_Moresby
6	7	Narsarsuaq	Narssarssuaq	Greenland	UAK	BGBW	61.160517	-45.425978	112	-3.0	E	America/Godthab
7	8	Nuuk	Godthaab	Greenland	GOH	BGGH	64.190922	-51.678064	283	-3.0	E	America/Godthab
8	9	Sondre Stromfjord	Sondrestrom	Greenland	SFJ	BGSF	67.016969	-50.689325	165	-3.0	E	America/Godthab
9	10	Thule Air Base	Thule	Greenland	THU	BGTL	76.531203	-68.703161	251	-4.0	E	America/Thule
10	11	Akureyri	Akureyri	Iceland	AEY	BIAR	65.659994	-18.072703	6	0.0	N	Atlantic/Reykjavik



```
In [209]: routesdf.ix[:10]
Out[209]:
```

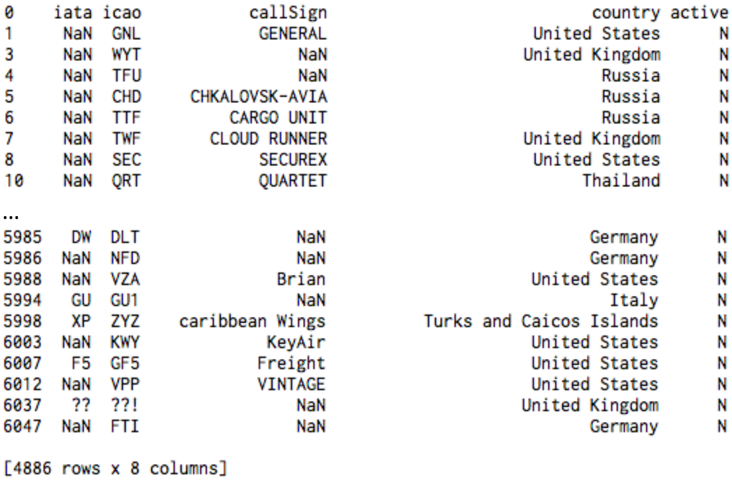
	airline	airlineID	srcAirport	srcApID	destAp	destApID	codeshare	stops	equipment
0	2B	410	AER	2965	KZN	2990	NaN	0	CR2
1	2B	410	ASF	2966	KZN	2990	NaN	0	CR2
2	2B	410	ASF	2966	MRV	2962	NaN	0	CR2
3	2B	410	CEK	2968	KZN	2990	NaN	0	CR2
4	2B	410	CEK	2968	OVB	4078	NaN	0	CR2
5	2B	410	DME	4029	KZN	2990	NaN	0	CR2
6	2B	410	DME	4029	NBC	6969	NaN	0	CR2
7	2B	410	DME	4029	TGK	\N	NaN	0	CR2
8	2B	410	DME	4029	UUA	6160	NaN	0	CR2
9	2B	410	EGO	6156	KGD	2952	NaN	0	CR2
10	2B	410	EGO	6156	KZN	2990	NaN	0	CR2

f) Number of airlines (unique) in airlines data: 6048 indexes in airlinesdf, so 6048 unique airlines in the data.



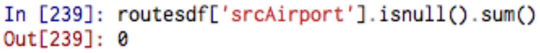
```
In [232]: airlinesdf.airName.unique
Out[232]:
<bound method Series.unique of 0
1                                  135 Airways
2                                1Time Airline
3       2 Sqn No 1 Elementary Flying Training School
4                              213 Flight Unit
5                 223 Flight Unit State Airline
6                             224th Flight Unit
7                                  247 Jet Ltd
8                                  3D Aviation
9                                  40-Mile Air
10                                      4D Air
...
6037                        British Air Ferries
6038                                    Voestar
6039                               All Colombia
6040                         Regionalia Uruguay
6041                       Regionalia Venezuela
6042                           Regionalia Chile
6043                                 Vuela Cuba
6044                              All Australia
6045                                 Fly Europa
6046                                FlyPortugal
6047                      FTI Fluggesellschaft
Name: airName, dtype: object>
```

g) Defunct airlines (4886 airlines out of 6048 total in airlinesdf)



```
0     iata icao          callSign                    country active
1      NaN  GNL           GENERAL              United States      N
3      NaN  WYT               NaN             United Kingdom      N
4      NaN  TFU               NaN                     Russia      N
5      NaN  CHD     CHKALOVSK-AVIA                    Russia      N
6      NaN  TTF        CARGO UNIT                     Russia      N
7      NaN  TWF      CLOUD RUNNER             United Kingdom      N
8      NaN  SEC           SECUREX              United States      N
10     NaN  QRT           QUARTET                   Thailand      N

...
5985    DW  DLT               NaN                    Germany      N
5986   NaN  NFD               NaN                    Germany      N
5988   NaN  VZA             Brian              United States      N
5994    GU  GU1               NaN                      Italy      N
5998    XP  ZYZ   caribbean Wings   Turks and Caicos Islands      N
6003   NaN  KWY            KeyAir              United States      N
6007    F5  GF5           Freight              United States      N
6012   NaN  VPP           VINTAGE              United States      N
6037    ??  ??!               NaN             United Kingdom      N
6047   NaN  FTI               NaN                    Germany      N

[4886 rows x 8 columns]
```

h) Flights from nowhere. There are no flights from nowhere, where ‘nowhere’ is defined as a source airport in the routes data frame, which has a blank or empty column entry.



```
In [239]: routesdf['srcAirport'].isnull().sum()
Out[239]: 0
```

Extra Credit code:

```
#import geopy package
#Use geopy package to calculate distance from 2 sets of longi-latitude data
import geopy as geo
from geopy.distance import VincentyDistance
x = (41.9742, 87.9073)#coordinates of Chicago (ORD)
y = (35.0433, 106.6129)#example coordinates of ABQ (Alberquerque)
print(VincentyDistance(x, y).miles)#prints distance ORD -> ABQ

#Create a dataframe for all routes from ORD to other airports
ordRoutesdf = routesdf[routesdf.srcAirport=="ORD"]

#rename apID column in airportsdf to be same as ordRoutesdf column
#('destApID' for merging), so that df contains lat/long coordinates
airportsdf.rename(columns={'apID':'destApID'}, inplace=True)
#df now contains all ordRoutesdf, airport and location data
df = ordRoutesdf.merge(airportsdf, on='destApID', how='left')

#Need to iterate through df to pass lat/long coordinates to distance function.
#Got up to here and got stuck! For some reason the coordinates appear as 'NaN'!
#Keep x coordinates constant and for y use coordinates extracted from destApID,
#pass coordinates to distance function, sort in descending order to get the top 10
#routes.
```