

Project 1. WEB SCRAPPING:

```
import requests
from bs4 import BeautifulSoup as bs
import pandas as pdr = requests.get(url)

soup = bs(r.text,"html.parser")

mt= soup.find("div", id="column-mobileMedian", class_ = "pure-u-1 pure-
u-lg-1-2 results-column mobileMedian-results")
ct= soup.find("div", id = "column-fixedMedian", class_="pure-u-1 pure-
u-lg-1-2 results-column fixedMedian-results")

m_table = mt.find("table", class_ = "list-results")
c_table = ct.find("table", class_ = "list-results")

headers = m_table.find_all("th")

titles = []
for i in headers:
    title = i.text
    titles.append(title)

titles.insert(1,"rank")

print(titles)

df = pd.DataFrame(columns = titles)
df1 = pd.DataFrame(columns = titles)

rows = m_table.find_all("tr")
rows1 = c_table.find_all("tr")

for i in rows[1:]:
    data = i.find_all("td")
    row_data = [td.text.strip() for td in data]
    if len(row_data) == len(df.columns):
        df.loc[len(df)] = row_data

for i in rows1[1:]:
    data1 = i.find_all("td")
    row_data1 = [td1.text.strip() for td1 in data1]
    if len(row_data1) == len(df1.columns):
        df1.loc[len(df1)] = row_data1

df.set_index("#", inplace= True)

df1.set_index("#", inplace= True)
```

```
df.to_csv("Mobile_Table1.csv")
df1.to_csv("Country_Table2.csv")
url = "https://www.speedtest.net/global-index"
r = requests.get(url)
```

PROJECT 2. Auto Claim Fraud Detection:

```
[ ] import pandas as pd
```

```
[ ] df= pd.read_csv("/content/AutoInsClaims.csv")
```

```
[ ] df.shape
```

```
(975, 38)
```

```
[ ] df.head()
```

	HOUSEHOLD_ID	DRIVER_ID	POLICY_ID	CLAIM_ID	INCIDENT_CAUSE	DESCRIPTION	CLAIM_STATUS	ODOMETER_AT_LOSS	LOSS_EVENT_TIME	CLAIM_INIT_TIME	...	GENDER	BIRTH
0	CH42335	XZJ2837	NW5567882	A-2017-UU907	3	NaN	1	157654.9	4/25/17	4/28/17	...	M	12/
1	IH49805	VVR6423	UR4864804	A-2018-Fl481	3	NaN	3	226164.5	8/26/18	8/31/18	...	F	9/
				A-2016-									

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 975 entries, 0 to 974
Data columns (total 38 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   HOUSEHOLD_ID                          975 non-null    object
1   DRIVER_ID                             975 non-null    object
2   POLICY_ID                             975 non-null    object
3   CLAIM_ID                              975 non-null    object
4   INCIDENT_CAUSE                        975 non-null    int64
5   DESCRIPTION                           0 non-null     float64
6   CLAIM_STATUS                          975 non-null    int64
7   ODOMETER_AT_LOSS                     975 non-null    float64
8   LOSS_EVENT_TIME                      975 non-null    object
9   CLAIM_INIT_TIME                      975 non-null    object
10  POLICE_REPORT                         975 non-null    int64
11  CLAIMS_AT_LOSS_DATE                  975 non-null    int64
12  LOSS_LOCATION_LAT                    975 non-null    float64
13  LOSS_LOCATION_LONG                   975 non-null    float64
14  CLAIM_AMOUNT                         975 non-null    float64
15  FLAG_FOR_FRAUD_INV                   975 non-null    int64
16  PRIMARY_DRIVER_ID                    975 non-null    object
17  START_DATE                           975 non-null    object
18  EXPIRY DATE                          975 non-null    object
```

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

```
HOUSEHOLD_ID      0
DRIVER_ID          0
POLICY_ID          0
CLAIM_ID           0
INCIDENT_CAUSE     0
DESCRIPTION        975
CLAIM_STATUS       0
ODOMETER_AT_LOSS   0
LOSS_EVENT_TIME    0
CLAIM_INIT_TIME    0
POLICE_REPORT      0
CLAIMS_AT_LOSS_DATE 0
LOSS_LOCATION_LAT  0
LOSS_LOCATION_LONG 0
CLAIM_AMOUNT       0
FLAG_FOR_FRAUD_INV 0
PRIMARY_DRIVER_ID  0
START_DATE         0
EXPIRY DATE        0
MODEL_YEAR         0
MAKE               0
MODEL              0
PLATE              0
COLOR              0
```

```
df.isna().any()
```

```
HOUSEHOLD_ID      False
DRIVER_ID         False
POLICY_ID         False
CLAIM_ID          False
INCIDENT_CAUSE     False
DESCRIPTION        True
CLAIM_STATUS       False
ODOMETER_AT_LOSS   False
LOSS_EVENT_TIME    False
CLAIM_INIT_TIME    False
POLICE_REPORT      False
CLAIMS_AT_LOSS_DATE False
LOSS_LOCATION_LAT  False
LOSS_LOCATION_LONG False
CLAIM_AMOUNT       False
FLAG_FOR_FRAUD_INV False
PRIMARY_DRIVER_ID  False
START_DATE         False
EXPIRY_DATE        False
MODEL_YEAR         False
MAKE              False
MODEL             False
```

```
t=df.dtypes
```

```
t
```

```
HOUSEHOLD_ID      object
DRIVER_ID         object
POLICY_ID         object
CLAIM_ID          object
INCIDENT_CAUSE     int64
DESCRIPTION        float64
CLAIM_STATUS       int64
ODOMETER_AT_LOSS   float64
LOSS_EVENT_TIME    object
CLAIM_INIT_TIME    object
POLICE_REPORT      int64
CLAIMS_AT_LOSS_DATE int64
LOSS_LOCATION_LAT  float64
LOSS_LOCATION_LONG float64
CLAIM_AMOUNT       float64
FLAG_FOR_FRAUD_INV int64
PRIMARY_DRIVER_ID  object
START_DATE         object
EXPIRY_DATE        object
MODEL_YEAR         int64
MAKE              object
MODEL             object
```

```
df1=df.drop("DESCRIPTION",axis=1)
```

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 975 entries, 0 to 974
Data columns (total 37 columns):
#   Column                Non-Null Count  Dtype
---  -
0   HOUSEHOLD_ID           975 non-null    object
1   DRIVER_ID              975 non-null    object
2   POLICY_ID              975 non-null    object
3   CLAIM_ID               975 non-null    object
4   INCIDENT_CAUSE         975 non-null    int64
5   CLAIM_STATUS           975 non-null    int64
6   ODOMETER_AT_LOSS       975 non-null    float64
7   LOSS_EVENT_TIME        975 non-null    object
8   CLAIM_INIT_TIME        975 non-null    object
9   POLICE_REPORT          975 non-null    int64
10  CLAIMS_AT_LOSS_DATE     975 non-null    int64
11  LOSS_LOCATION_LAT       975 non-null    float64
12  LOSS_LOCATION_LONG      975 non-null    float64
13  CLAIM_AMOUNT           975 non-null    float64
14  FLAG_FOR_FRAUD_INV      975 non-null    int64
```

```
df1['LOSS_EVENT_TIME'] = pd.to_datetime(df1['LOSS_EVENT_TIME'])
df1['EXPIRY_DATE'] = pd.to_datetime(df1['EXPIRY_DATE'])
df1['DRIVERS_LICENSE_EXPIRY'] = pd.to_datetime(df1['DRIVERS_LICENSE_EXPIRY'])
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 975 entries, 0 to 974
Data columns (total 37 columns):
#   Column                Non-Null Count  Dtype
---  -
0   HOUSEHOLD_ID           975 non-null    object
1   DRIVER_ID              975 non-null    object
2   POLICY_ID              975 non-null    object
3   CLAIM_ID               975 non-null    object
4   INCIDENT_CAUSE         975 non-null    int64
5   CLAIM_STATUS           975 non-null    int64
6   ODOMETER_AT_LOSS       975 non-null    float64
7   LOSS_EVENT_TIME        975 non-null    datetime64[ns]
8   CLAIM_INIT_TIME        975 non-null    object
9   POLICE_REPORT          975 non-null    int64
10  CLAIMS_AT_LOSS_DATE     975 non-null    int64
11  LOSS_LOCATION_LAT       975 non-null    float64
12  LOSS_LOCATION_LONG      975 non-null    float64
13  CLAIM_AMOUNT           975 non-null    float64
```

```
fraud1 = df1['EXPIRY_DATE'] - df1['LOSS_EVENT_TIME']
fraud1
```

```
0      147 days
1      -73 days
2      229 days
3     -140 days
4     -150 days
...
970    -141 days
971     -25 days
972      -2 days
973      87 days
974    -177 days
Length: 975, dtype: timedelta64[ns]
```

```
[ ] from datetime import timedelta
zero_days = timedelta(days=0)
```

```
[ ] df1['Fraud_Pol_Claim'] = fraud1 < zero_days
```

```
df1['Fraud_Pol_Claim'] = fraud1 < zero_days
df1
```

```
HOUSEHOLD_ID  DRIVER_ID  POLICY_ID  CLAIM_ID  INCIDENT_CAUSE  CLAIM_STATUS  ODOMETER_AT_LOSS  LOSS_EVENT_TIME  CLAIM_INIT_TIME  POLICE_REPORT  ...  BIRTHDATE
0      CH42335  XZJ2037  NW5567882  A-2017-  UU907          3          1          157654.9      2017-04-25      4/28/17          1  ...  12/25/7
1      IH49805  VVR6423  UR4864804  A-2018-  FI481          3          3          226154.5      2018-08-26      8/31/18          0  ...  9/14/4
2      AF28736  UQM2512  RR8595908  A-2016-  ZG694          1          1          83958.6      2016-01-07      1/11/16          0  ...  5/16/8
3      EF53594  YDT5591  RN5640634  A-2016-  NG783          1          1          309570.3      2016-12-11      12/18/16          0  ...  1/18/7
4      LD32277  ONM5465  YY1229530  A-2017-  ZO863          3          3          136633.9      2017-06-06      6/8/17          0  ...  5/1/5
```

```
fraud2 = df1['DRIVERS_LICENSE_EXPIRY'] - df1['LOSS_EVENT_TIME']
fraud2
```

```
0      481 days
1     -211 days
2     1412 days
3      886 days
4     1126 days
...
970     571 days
971    1097 days
972    1219 days
973    1332 days
974    1478 days
Length: 975, dtype: timedelta64[ns]
```

```
df1['Fraud_DL_claims'] = fraud2 < zero_days
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 975 entries, 0 to 974
Data columns (total 39 columns):
```

```
[ ] import numpy as np
```

```
df1['amount_fraud'] = np.where(df1['CLAIM_AMOUNT'] > 10000, True, False)
```

```
[ ] df1.head()
df1.columns
```

```
Index(['HOUSEHOLD_ID', 'DRIVER_ID', 'POLICY_ID', 'CLAIM_ID', 'INCIDENT_CAUSE',
      'CLAIM_STATUS', 'ODOMETER_AT_LOSS', 'LOSS_EVENT_TIME',
      'CLAIM_INIT_TIME', 'POLICE_REPORT', 'CLAIMS_AT_LOSS_DATE',
      'LOSS_LOCATION_LAT', 'LOSS_LOCATION_LONG', 'CLAIM_AMOUNT',
      'FLAG_FOR_FRAUD_INV', 'PRIMARY_DRIVER_ID', 'START_DATE', 'EXPIRY_DATE',
      'MODEL_YEAR', 'MAKE', 'MODEL', 'PLATE', 'COLOR', 'INITIAL_ODOMETER',
      'LOW_MILEAGE_USE', 'FIRST_NAME', 'LAST_NAME', 'GENDER', 'BIRTHDATE',
      'SSN', 'DRIVERS_LICENSE_ID', 'DRIVERS_LICENSE_EXPIRY',
      'DRIVERS_LICENSE_STATE', 'DATE_AT_CURRENT_ADDRESS', 'CONTACT_NUMBER',
      'EMAIL', 'COMMUTE_DISCOUNT', 'Fraud_Pol_Claim', 'Fraud_DL_claims',
      'amount_fraud'],
      dtype='object')
```

```
[ ] df1.head()
```

```
df1.head()
```

	HOUSEHOLD_ID	DRIVER_ID	POLICY_ID	CLAIM_ID	INCIDENT_CAUSE	CLAIM_STATUS	ODOMETER_AT_LOSS	LOSS_EVENT_TIME	CLAIM_INIT_TIME	POLICE_REPORT	...	DRIVERS_LICENSE
0	CH42335	XZJ2837	NW5567882	A-2017-UU907	3	1	157654.9	2017-04-25	4/28/17	1	...	X151-5
1	IH49805	VVR5423	UR4864804	A-2018-FI481	3	3	226154.5	2018-06-26	8/31/18	0	...	K702-7
2	AF28736	UQM2512	RR8595908	A-2016-ZG694	1	1	83988.6	2016-01-07	1/11/16	0	...	V549-5
3	EF53594	YDT5591	RN5640634	A-2016-NG783	1	1	309570.3	2016-12-11	12/18/16	0	...	Z350-2
4	LD32277	ONM5465	YY1228530	A-2017-ZO863	3	3	136633.9	2017-06-06	6/8/17	0	...	E912-8

5 rows x 40 columns

```
[ ] import seaborn as sns
import matplotlib.pyplot as plt
```

