Problem 1.

https://github.com/meesuny/lab2

Problem 2.

In [1]:
```
!pip install openpyxl  # for Excel
```

```
Collecting openpyxl
  Downloading openpyxl-3.0.10-py2.py3-none-any.whl (242 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 242.1/242.1 kB 4.3 MB/s eta 0:00:00a 0:00:01
Collecting et-xmlfile
  Downloading et_xmlfile-1.1.0-py3-none-any.whl (4.7 kB)
Installing collected packages: et-xmlfile, openpyxl
Successfully installed et-xmlfile-1.1.0 openpyxl-3.0.10
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a
 virtual environment instead: https://pip.pypa.io/warnings/venv
```

In [2]:
```
!pip install pyreadstat  # for SPSS
```

```
Collecting pyreadstat
  Downloading pyreadstat-1.1.9-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.7 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.7/2.7 MB 10.1 MB/s eta 0:00:0000:0100:01
Requirement already satisfied: pandas>=1.2.0 in /usr/local/lib/python3.10/site-packages (from pyreadstat) (1.4.4)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/site-packages (from pandas>=1.2.0->pyreadstat) (1.23.3)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/site-packages (from pandas>=1.2.0->pyreadstat) (2022.4)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/site-packages (from pandas>=1.2.0->pyreadstat) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/site-packages (from python-dateutil>=2.8.1->pandas>=1.2.0->pyreadstat) (1.16.0)
Installing collected packages: pyreadstat
Successfully installed pyreadstat-1.1.9
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a
 virtual environment instead: https://pip.pypa.io/warnings/venv
```

In [3]:
```
!pip install pystata
```

```
Collecting pystata
  Downloading pystata-0.0.1-py3-none-any.whl (21 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/site-packages (from pystata) (1.4.4)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/site-packages (from pandas->pystata) (2022.4)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/site-packages (from pandas->pystata) (2.8.2)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/site-packages (from pandas->pystata) (1.23.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/site-packages (from python-dateutil>=2.8.1->pandas->pystata) (1.16.0)
Installing collected packages: pystata
Successfully installed pystata-0.0.1
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a
 virtual environment instead: https://pip.pypa.io/warnings/venv
```

In [4]:

```python
import pandas as pd
import numpy as np
import requests
import json
import os
import openpyxl
from bs4 import BeautifulSoup
import collections
collections.Callable = collections.abc.Callable
```

In [5]:

```python
data1 = pd.read_csv('data/data1.csv', header=2)
data1
```

Out[5]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices	Explained by: Generosity	Explained by: Perceptions of corruption
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681	0.192	0.393
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686	0.286	0.340
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868	0.683	0.284	0.408
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914	0.677	0.353	0.138
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660	0.256	0.357
...	...	...	...	...	...	...	...	...	...	...	...
151	Yemen	3.355	3.448	3.262	1.106	0.442	1.073	0.343	0.244	0.083	0.064
152	Tanzania	3.303	3.414	3.193	0.628	0.455	0.991	0.381	0.481	0.270	0.097
153	South Sudan	3.254	3.385	3.123	1.691	0.337	0.608	0.177	0.112	0.224	0.106
154	Central African Republic	3.083	3.227	2.939	2.487	0.024	0.000	0.010	0.305	0.218	0.038
155	Burundi	2.905	3.074	2.735	1.752	0.091	0.627	0.145	0.065	0.149	0.076

156 rows × 11 columns

In [6]:
```
data2 = pd.read_csv('data/data2.txt', header=2, comment = '/')
data2
```

Out[6]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices	Explained by: Generosity	Explained by: Perceptions of corruption
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681	0.192	0.393
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686	0.286	0.340
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868	0.683	0.284	0.408
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914	0.677	0.353	0.138
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660	0.256	0.357
...	...	...	...	...	...	...	...	...	...	...	...
151	Yemen	3.355	3.448	3.262	1.106	0.442	1.073	0.343	0.244	0.083	0.064
152	Tanzania	3.303	3.414	3.193	0.628	0.455	0.991	0.381	0.481	0.270	0.097
153	South Sudan	3.254	3.385	3.123	1.691	0.337	0.608	0.177	0.112	0.224	0.106
154	Central African Republic	3.083	3.227	2.939	2.487	0.024	0.000	0.010	0.305	0.218	0.038
155	Burundi	2.905	3.074	2.735	1.752	0.091	0.627	0.145	0.065	0.149	0.076

156 rows × 11 columns

In [7]:
```
data3 = pd.read_csv('data/data3.txt', header=2, sep='\t')
data3
```

Out[7]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices	Explained by: Generosity	Explained by: Perceptions of corruption
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681	0.192	0.393
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686	0.286	0.340
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868	0.683	0.284	0.408
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914	0.677	0.353	0.138
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660	0.256	0.357
...	...	...	...	...	...	...	...	...	...	...	...
151	Yemen	3.355	3.448	3.262	1.106	0.442	1.073	0.343	0.244	0.083	0.064
152	Tanzania	3.303	3.414	3.193	0.628	0.455	0.991	0.381	0.481	0.270	0.097
153	South Sudan	3.254	3.385	3.123	1.691	0.337	0.608	0.177	0.112	0.224	0.106
154	Central African Republic	3.083	3.227	2.939	2.487	0.024	0.000	0.010	0.305	0.218	0.038
155	Burundi	2.905	3.074	2.735	1.752	0.091	0.627	0.145	0.065	0.149	0.076

156 rows × 11 columns

In [8]:
```
data4 = pd.read_csv('data/data4.txt', header=None, sep='$')
data4

# Add headers on this one?
```

Out[8]:

	0	1	2	3	4	5	6	7	8	9	10
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681	0.192	0.393
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686	0.286	0.340
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868	0.683	0.284	0.408
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914	0.677	0.353	0.138
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660	0.256	0.357
...	...	...	...	...	...	...	...	...	...	...	...
151	Yemen	3.355	3.448	3.262	1.106	0.442	1.073	0.343	0.244	0.083	0.064
152	Tanzania	3.303	3.414	3.193	0.628	0.455	0.991	0.381	0.481	0.270	0.097
153	South Sudan	3.254	3.385	3.123	1.691	0.337	0.608	0.177	0.112	0.224	0.106
154	Central African Republic	3.083	3.227	2.939	2.487	0.024	0.000	0.010	0.305	0.218	0.038
155	Burundi	2.905	3.074	2.735	1.752	0.091	0.627	0.145	0.065	0.149	0.076

156 rows × 11 columns

In [9]:

```
data5 = pd.read_csv('data/data5.csv', nrows=156)
data5
```

Out[9]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices	Explained by: Generosity	Explained by: Perceptions of corruption
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681	0.192	0.393
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686	0.286	0.340
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868	0.683	0.284	0.408
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914	0.677	0.353	0.138
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660	0.256	0.357
...	...	...	...	...	...	...	...	...	...	...	...
151	Yemen	3.355	3.448	3.262	1.106	0.442	1.073	0.343	0.244	0.083	0.064
152	Tanzania	3.303	3.414	3.193	0.628	0.455	0.991	0.381	0.481	0.270	0.097
153	South Sudan	3.254	3.385	3.123	1.691	0.337	0.608	0.177	0.112	0.224	0.106
154	Central African Republic	3.083	3.227	2.939	2.487	0.024	0.000	0.010	0.305	0.218	0.038
155	Burundi	2.905	3.074	2.735	1.752	0.091	0.627	0.145	0.065	0.149	0.076

156 rows × 11 columns

In [10]:
```
data6 = pd.read_csv('data/data6.dat', na_values = '999.000')
data6
```

Out[10]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices	Explained by: Generosity	Explained by: Perceptions of corruption
0	Finland	7.632	7.695	7.569	2.595	NaN	NaN	NaN	0.681	0.192	0.393
1	Norway	7.594	7.657	7.530	NaN	NaN	1.582	NaN	0.686	0.286	0.340
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	NaN	0.683	0.284	0.408
3	Iceland	7.495	7.593	NaN	2.426	1.343	1.644	0.914	0.677	0.353	NaN
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660	0.256	0.357
...	...	...	...	...	...	...	...	...	...	...	...
151	NaN	3.355	3.448	3.262	1.106	0.442	1.073	0.343	0.244	NaN	0.064
152	Tanzania	NaN	NaN	3.193	0.628	NaN	0.991	0.381	0.481	0.270	0.097
153	South Sudan	3.254	NaN	3.123	1.691	0.337	NaN	0.177	0.112	0.224	0.106
154	Central African Republic	3.083	3.227	2.939	2.487	0.024	0.000	0.010	0.305	NaN	0.038
155	Burundi	2.905	3.074	NaN	1.752	0.091	NaN	0.145	0.065	0.149	0.076

156 rows × 11 columns

In [11]:

```
data7 = pd.read_excel('data/data7.xlsx', sheet_name = "Data")
data7

# Had to pip install openpyxl in order for this to run.
# Collecting openpyxl
#  Downloading openpyxl-3.0.10-py2.py3-none-any.whl (242 kB)
#     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 242.1/242.1 kB 367.7 kB/s eta 0:00:00a 0:00:01
# Collecting et-xmlfile
#  Downloading et_xmlfile-1.1.0-py3-none-any.whl (4.7 kB)
# Installing collected packages: et-xmlfile, openpyxl
# Successfully installed et-xmlfile-1.1.0 openpyxl-3.0.10
# WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a
```

Out[11]:

	Country	Happiness score	Whisker-high	Whisker-low	Dystopia (1.92) + residual	Explained by: GDP per capita	Explained by: Social support	Explained by: Healthy life expectancy	Explained by: Freedom to make life choices	Explained by: Generosity	Explained by: Perceptions of corruption
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681	0.192	0.393
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686	0.286	0.340
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868	0.683	0.284	0.408
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914	0.677	0.353	0.138
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660	0.256	0.357
...	...	...	...	...	...	...	...	...	...	...	...
151	Yemen	3.355	3.448	3.262	1.106	0.442	1.073	0.343	0.244	0.083	0.064
152	Tanzania	3.303	3.414	3.193	0.628	0.455	0.991	0.381	0.481	0.270	0.097
153	South Sudan	3.254	3.385	3.123	1.691	0.337	0.608	0.177	0.112	0.224	0.106
154	Central African Republic	3.083	3.227	2.939	2.487	0.024	0.000	0.010	0.305	0.218	0.038
155	Burundi	2.905	3.074	2.735	1.752	0.091	0.627	0.145	0.065	0.149	0.076

156 rows × 11 columns

In [12]:

```
data8 = pd.read_stata('data/data8.dta')
data8

# STATA pip install
```

Out[12]:

	country	happinessscore	whiskerhigh	whiskerlow	dystopia192residual	explainedbygdppercapita	explainedbysocialsupport	explainedbyhealthylifeexpectancy	explainedbyfreedomtor
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868	
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914	
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	
...	...	...	...	...	...	...	...	...	
151	Yemen	3.355	3.448	3.262	1.106	0.442	1.073	0.343	
152	Tanzania	3.303	3.414	3.193	0.628	0.455	0.991	0.381	
153	South Sudan	3.254	3.385	3.123	1.691	0.337	0.608	0.177	
154	Central African Republic	3.083	3.227	2.939	2.487	0.024	0.000	0.010	
155	Burundi	2.905	3.074	2.735	1.752	0.091	0.627	0.145	

156 rows × 11 columns

In [13]:

```
data9 = pd.read_spss('data/data9.sav')
data9
```

Out[13]:

	country	happiness	whiskerhigh	whiskerlow	dystopia	gdpPC	socsupport	lifeexp	lifechoice	generous	corrupt
0	Finland	7.632	7.695	7.569	2.595	1.305	1.592	0.874	0.681	0.192	0.393
1	Norway	7.594	7.657	7.530	2.383	1.456	1.582	0.861	0.686	0.286	0.340
2	Denmark	7.555	7.623	7.487	2.370	1.351	1.590	0.868	0.683	0.284	0.408
3	Iceland	7.495	7.593	7.398	2.426	1.343	1.644	0.914	0.677	0.353	0.138
4	Switzerland	7.487	7.570	7.405	2.320	1.420	1.549	0.927	0.660	0.256	0.357
...	...	...	...	...	...	...	...	...	...	...	...
151	Yemen	3.355	3.448	3.262	1.106	0.442	1.073	0.343	0.244	0.083	0.064
152	Tanzania	3.303	3.414	3.193	0.628	0.455	0.991	0.381	0.481	0.270	0.097
153	South Sudan	3.254	3.385	3.123	1.691	0.337	0.608	0.177	0.112	0.224	0.106
154	Central African Republic	3.083	3.227	2.939	2.487	0.024	0.000	0.010	0.305	0.218	0.038
155	Burundi	2.905	3.074	2.735	1.752	0.091	0.627	0.145	0.065	0.149	0.076

156 rows × 11 columns

In [14]:
```
data10 = pd.read_sas('data/data10.xpt')
data10
```

Out[14]:

	COUNTRY	HAPPINES	WHISKERH	WHISKERL	DYSTOPIA	EXPLAINE	EXPLAIN2	EXPLAIN3	EXPLAIN4	EXPLAIN5	EXPLAIN6
0	b'Finland'	7.632	7.695	7.569	2.595	1.305	1.592000e+00	0.874	0.681	0.192	0.393
1	b'Norway'	7.594	7.657	7.530	2.383	1.456	1.582000e+00	0.861	0.686	0.286	0.340
2	b'Denmark'	7.555	7.623	7.487	2.370	1.351	1.590000e+00	0.868	0.683	0.284	0.408
3	b'Iceland'	7.495	7.593	7.398	2.426	1.343	1.644000e+00	0.914	0.677	0.353	0.138
4	b'Switzerland'	7.487	7.570	7.405	2.320	1.420	1.549000e+00	0.927	0.660	0.256	0.357
...	...	...	...	...	...	...	...	...	...	...	...
151	b'Yemen'	3.355	3.448	3.262	1.106	0.442	1.073000e+00	0.343	0.244	0.083	0.064
152	b'Tanzania'	3.303	3.414	3.193	0.628	0.455	9.910000e-01	0.381	0.481	0.270	0.097
153	b'South Sudan'	3.254	3.385	3.123	1.691	0.337	6.080000e-01	0.177	0.112	0.224	0.106
154	b'Central African Republic'	3.083	3.227	2.939	2.487	0.024	5.397605e-79	0.010	0.305	0.218	0.038
155	b'Burundi'	2.905	3.074	2.735	1.752	0.091	6.270000e-01	0.145	0.065	0.149	0.076

156 rows × 11 columns

Problem 3.

In [15]:
```
nasatoken = os.environ['nasatoken']
```

In [16]:
```
useragent_url = 'https://httpbin.org/user-agent'
r = requests.get(useragent_url)
useragent = json.loads(r.text)['user-agent']
```

In [17]:
```
headers = {'X-API-Key': nasatoken,
           'User-Agent': useragent,
           'From': 'tby8aj@virginia.edu'}

root = 'https://api.nasa.gov'

endpoint = '/neo/rest/v1/feed?start_date=2022-10-01&end_date=2022-10-08&api_key=nasatoken'

r = requests.get(root + endpoint,
```

```
                  headers = headers)
r
```

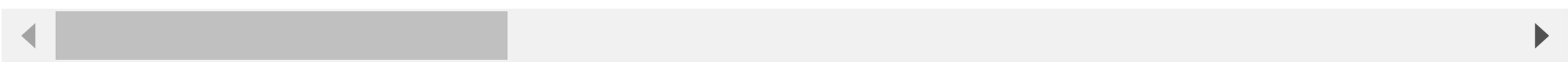
Out[17]:
```
<Response [200]>
```

In [18]:
```
myjson = json.loads(r.text)
near_earth_objects1 = pd.json_normalize(myjson, record_path = ['near_earth_objects','2022-10-01'])
near_earth_objects2 = pd.json_normalize(myjson, record_path = ['near_earth_objects','2022-10-02'])
near_earth_objects3 = pd.json_normalize(myjson, record_path = ['near_earth_objects','2022-10-03'])
near_earth_objects4 = pd.json_normalize(myjson, record_path = ['near_earth_objects','2022-10-04'])
near_earth_objects5 = pd.json_normalize(myjson, record_path = ['near_earth_objects','2022-10-05'])
near_earth_objects6 = pd.json_normalize(myjson, record_path = ['near_earth_objects','2022-10-06'])
near_earth_objects7 = pd.json_normalize(myjson, record_path = ['near_earth_objects','2022-10-07'])
near_earth_objects_all = pd.concat([near_earth_objects1, near_earth_objects2, near_earth_objects3, near_earth_objects4,
                                    near_earth_objects5, near_earth_objects6, near_earth_objects7], ignore_index=True)
#near_earth_objects_all

# This adds "close_approach_date" as the very last column of the resulting data frame.
df = pd.DataFrame(data=near_earth_objects_all)
df['close_approach_date'] = [d[0]['close_approach_date'] for d in df['close_approach_data']]

near_earth_objects_all
```

Out[18]:

	id	neo_reference_id	name	nasa_jpl_url	absolute_magnitude_h	is_potentially_hazardous_asteroid	close_approach_data	is_sentry_object	
0	3429684	3429684	(2008 TZ)	http://ssd.jpl.nasa.gov/sbdb.cgi?sstr=3429684	25.40	False	[{'close_approach_date': '2022-10-01', 'close_...	False	http://api.nasa.gov/neo
1	3548667	3548667	(2010 TX54)	http://ssd.jpl.nasa.gov/sbdb.cgi?sstr=3548667	20.98	False	[{'close_approach_date': '2022-10-01', 'close_...	False	http://api.nasa.gov/neo
2	3650847	3650847	(2013 TN127)	http://ssd.jpl.nasa.gov/sbdb.cgi?sstr=3650847	26.20	False	[{'close_approach_date': '2022-10-01', 'close_...	False	http://api.nasa.gov/neo
3	3772755	3772755	(2017 FT90)	http://ssd.jpl.nasa.gov/sbdb.cgi?sstr=3772755	26.20	False	[{'close_approach_date': '2022-10-01', 'close_...	False	http://api.nasa.gov/neo
4	3782015	3782015	(2017 SZ20)	http://ssd.jpl.nasa.gov/sbdb.cgi?sstr=3782015	24.50	False	[{'close_approach_date': '2022-10-01', 'close_...	False	http://api.nasa.gov/neo
...	...	...	...	...	...	...	...	...	
75	3342645	3342645	(2006 SG7)	http://ssd.jpl.nasa.gov/sbdb.cgi?sstr=3342645	22.90	False	[{'close_approach_date': '2022-10-07', 'close_...	False	http://api.nasa.gov/neo
76	3384488	3384488	(2007 RD20)	http://ssd.jpl.nasa.gov/sbdb.cgi?sstr=3384488	21.00	False	[{'close_approach_date': '2022-10-07', 'close_...	False	http://api.nasa.gov/neo
77	3625706	3625706	(2013 BD74)	http://ssd.jpl.nasa.gov/sbdb.cgi?sstr=3625706	24.20	False	[{'close_approach_date': '2022-10-07', 'close_...	False	http://api.nasa.gov/neo
78	3648879	3648879	(2013 TJ6)	http://ssd.jpl.nasa.gov/sbdb.cgi?sstr=3648879	25.20	False	[{'close_approach_date': '2022-10-07', 'close_...	False	http://api.nasa.gov/neo
79	54055076	54055076	(2020 TE2)	http://ssd.jpl.nasa.gov/sbdb.cgi?sstr=54055076	26.30	False	[{'close_approach_date': '2022-10-07', 'close_...	False	http://api.nasa.gov/neo/

80 rows × 18 columns



In [ ]:

Problem 4.

a) Zillow.com disallows the following pages and subpages to be crawled, which suggests that they do not allow pulling the current listed prices for houses for sale in Charlottesville from zillow.com.

- Disallow: /homes/comps_for_sale/
- Disallow: /homes/comps/
- Disallow: /homes/for_sale/*_agent_list/
- Disallow: /homes/for_sale/1_favorite_list/1_rs/1_fr/

b) Google.com allows both /m/finance and /finance to be crawled, so scraping the current stock prices off of google.com/finance is permitted.

c) Twitter disallows /*/followers to be crawled, so we wouldn't be able to copy the list of twitter accounts that each NBA player follows.

d) Genius.com disallows crawling /songs/*/metadata, and metadata presumably includes the lyrics, so it doesn't allow getting lyrics from their site. This means that we can't get the lyrics to Lizzo's Good as Hell from https://genius.com/Lizzo-good-as-hell-lyrics.

In [ ]: