In [1]:
```
import csv
import json
neos_path = "./data/neos.csv"
cad_path = "./data/cad.json"
```

In [2]:
```
def createNeosKeys(path):
    """Takes a csv file path and returns a list of feature names
    aka labels to be used as python dictionary keys"""
    try:
        f = open(path, 'r')
        neos_labels = next(f).split(',') # first line of the file

    finally:
        f.close()
        return neos_labels[:-1]
```

In [3]:
```
def createNeosValues(path):
    """Takes a csv file path and returns a list of the
    feature values to be used as python dictionary values"""
    neos_elements = []

    with open(path) as infile:
        reader = csv.reader(infile)
        next(reader)
        for row in reader:
            neos_elements.append(row)
        return neos_elements
```

In [4]:
```
def loadCad(path):
    """Takes a json file path and returns a dictionary"""
    with open(path) as f:
        return json.load(f)
```

Create NEOS dictionary

In [5]:
```
neos_keys = createNeosKeys(neos_path)
```

In [6]:
```
neos_values = createNeosValues(neos_path)
```

In [7]:
```
neos_dict = {}
neos_dict[neos_keys[0]] = [i[0] for i in neos_values]
```

In [8]:
```
neos_dict = {}
for feat in range(len(neos_keys)):
    neos_dict[neos_keys[feat]] = [i[feat] for i in neos_values]
```

In [9]:
```
neos_dict = {neos_keys[feat]:[i[feat] for i in neos_values] for feat in range(len(neos
```

Create CAD dictionary

In [10]:
```
cad_data = loadCad(cad_path)
```

In [11]:
```
# Create cad dictionary: The official
cad_keys = cad_data['fields']
cad_values = cad_data['data']
cad_dict = {}
for feat in range(len(cad_keys)):
    cad_dict[cad_keys[feat]] = [i[feat] for i in cad_values]
```

In [12]:
```
# Alternative way to create the cad dictionary: The official
cad_dictionary = {}
cad_dictionary['des'] = [i[0] for i in cad_data['data']]
cad_dictionary['orbit_id'] = [i[1] for i in cad_data['data']]
cad_dictionary['jd'] = [i[2] for i in cad_data['data']]
cad_dictionary['cd'] = [i[3] for i in cad_data['data']]
cad_dictionary['dist'] = [i[4] for i in cad_data['data']]
cad_dictionary['dist_min'] = [i[5] for i in cad_data['data']]
cad_dictionary['dist_max'] = [i[6] for i in cad_data['data']]
cad_dictionary['v_rel'] = [i[7] for i in cad_data['data']]
cad_dictionary['v_inf'] = [i[8] for i in cad_data['data']]
cad_dictionary['t_sigma_f'] = [i[9] for i in cad_data['data']]
cad_dictionary['h'] = [i[10] for i in cad_data['data']]
```

In [13]:
```
# yet another way to create the cad dictionary
another_cad_dict = {cad_keys[feat]:[i[feat] for i in cad_values] for feat in range(len
```

In [14]:
```
# Wonder if it would be possible to do this using lambda and map
# list(map(lambda a: cad_dict['data'][a][0], cad_dict['data']))
```

1. How many NEOs are in the `neos.csv` data set?

In [15]:
```
# I need to check the length of neos_values list
len(neos_values)
```

Out[15]:
```
23967
```

2. What is the primary designation of the first Near Earth Object in the neos.csv data set?

Hint: Look at the first row of the CSV, under the header "pdes"

In [16]:
```
# I look at the first row of the CSV, under the header "pdes"
neos_dict['pdes'][0]
```

Out[16]:
```
'433'
```

3. What is the diameter (in kilometers) of the NEO whose name is "Apollo"?

Hint: Look for the row of the CSV containing the name "Apollo" in the "name" column, and find the entry under the "diameter" column.

In [17]:
```
# check if "Apollo" is in the name field
print("Apollo" in neos_dict['name'])
# save the index of the "Apollo" name in the 'name' list
apollo_index = neos_dict['name'].index("Apollo")
# Print out the diameter of the Apollo index
neos_dict['diameter'][apollo_index]
```

```
True
```

Out[17]:
```
'1.5'
```

4. How many NEOs have IAU names in the data set?

Hint: Count the number of rows that have nonempty entries in the "name" column.

In [18]:
```
# Check the number of empty indeces in the "name" column
empty_name_indices = [i for i, x in enumerate(neos_dict['name']) if x == ""]
# find the difference
len(neos_values) - len(empty_name_indices)
```

Out[18]:
```
343
```

5. How many NEOs have diameters in the data set?

Hint: Count the number of rows that have nonempty entries in the "diameter" column.

In [19]:
```
# Check the number of rows that have nonempty entries in the "diameter" column.
empty_diameter_indices = [i for i, x in enumerate(neos_dict['diameter']) if x == ""]
# find the difference
len(neos_values) - len(empty_diameter_indices)
```

Out[19]:
```
1268
```

6. How many close approaches are in the cad.json data set?

Hint: Instead of manually counting the entries, you can use the value of the "count" key.

In [20]:
```
# check cad_data keys to have a first glimpse of our data
print(cad_data.keys())
# I use the value of the "count" key
cad_data['count']
```

```
dict_keys(['signature', 'count', 'fields', 'data'])
```

Out[20]:
```
'406785'
```

7. On January 1st, 2000, how close did the NEO whose primary designation is "2015 CL" pass by Earth?

Hint: Find entries whose date starts with '2000-Jan-01'. One of the lists represents the close approach of the NEO "2015 CL". What is the value corresponding to the distance from Earth?

In [21]:
```
idx_2015_cl = cad_dict['des'].index("2015 CL")
```

In [22]:
```
cad_data['data'][idx_2015_cl]
```

Out[22]:
```
['2015 CL',
 '7',
 '2415739.441927107',
 '1901-Dec-20 22:36',
 '0.101258799371365',
 '0.100350977515448',
 '0.102168118307516',
 '12.4965470610001',
 '12.4944412172315',
 '03:16',
 '25.3']
```

8. On January 1st, 2000, how fast did the NEO whose primary designation is "2002 PB" pass by Earth?

Hint: Find entries whose date starts with '2000-Jan-01'. One of the lists represents the close approach of the NEO "2002 PB". What is the value corresponding to the velocity relative to Earth?

In [23]:
```
idx_2002_cl = cad_dict['des'].index("2002 PB")
cad_data['data'][idx_2002_cl]
```

Out[23]:
```
['2002 PB',
 '22',
 '2415054.516136742',
 '1900-Feb-04 00:23',
 '0.199796539822994',
 '0.199791629100154',
 '0.199801451243355',
 '21.4737681744826',
 '21.4731471304202',
 '00:14',
 '20.5']
```