# NASA Near Earth Objects Data Analysis

## 1 Objectives

The goal of this project is to first collect data related to near-Earth objects and then evaluate the statistics associated with some properties of the retrieved objects. The information of the asteroids that are in their closest approach within a range of date will be retrieved using NASA API and stored in a database. Furthermore, the statistics of some data such as, diameter, distance and the relative velocity of the asteroids will be evaluated and visually presented. The analysis performed in this project includes calculating the mean, variance, coefficient of variation and entropy of the data together with plotting the histogram, kernel-estimated probability distribution function, box plot, scatter matrix and pie chart of the data.

## 2 Data

Data is obtained from NASA website, Near Earth Object Web Service (NeoWs), Neo-Feed data, that contains information about near earth Asteroids. We will be searching for Asteroids based on their closest approach to Earth from NeoWs. Below is the link that provides more information:

https://api.nasa.gov/api.html#neows-feed

In the code, users are asked to input the "start date" and "end date". Then, a list of Asteroids that are in their closest distance to Earth, within the requested date range, will be obtained in JSON format using NASA API for Neo dataset. The URL looks like the following:

https://api.nasa.gov/neo/rest/v1/feed?start_date=START_DATE&end_date=END_DATE&api_key=API_KEY

For example:

https://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-07&end_date=2015-09-08&api_key=DEMO_KEY

Each Asteroid contains information such as, ID, diameter, velocity, etc. Figure 1 exhibits an example of the retrieved data in JSON format.

```
{
  "links" : {
    "next" : "https://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-08&end_date=2015-09-09&detailed=false&api_key=DEMO_KEY",
    "prev" : "https://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-06&end_date=2015-09-07&detailed=false&api_key=DEMO_KEY",
    "self" : "https://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-07&end_date=2015-09-08&detailed=false&api_key=DEMO_KEY"
  },
  "element_count" : 20,
  "near_earth_objects" : {
    "2015-09-08" : [ {
      "links" : {
        "self" : "https://api.nasa.gov/neo/rest/v1/neo/3726710?api_key=DEMO_KEY"
      },
      "neo_reference_id" : "3726710",
      "name" : "(2015 RC)",
      "nasa_jpl_url" : "http://ssd.jpl.nasa.gov/sbdb.cgi?sstr=3726710",
      "absolute_magnitude_h" : 24.3,
      "estimated_diameter" : {
        "kilometers" : {
          "estimated_diameter_min" : 0.0366906138,
          "estimated_diameter_max" : 0.0820427065
        },
        "meters" : {
          "estimated_diameter_min" : 36.6906137531,
          "estimated_diameter_max" : 82.0427064882
        },
        "miles" : {
          "estimated_diameter_min" : 0.0227984834,
          "estimated_diameter_max" : 0.0509789586
        },
        "feet" : {
          "estimated_diameter_min" : 120.3760332259,
          "estimated_diameter_max" : 269.1689931548
        }
      },
      "is_potentially_hazardous_asteroid" : false,
      "close_approach_data" : [ {
        "close_approach_date" : "2015-09-08",
        "epoch_date_close_approach" : 1441695600000,
        "relative_velocity" : {
          "kilometers_per_second" : "19.4701053405",
          "kilometers_per_hour" : "70092.3792259649",
          "miles_per_hour" : "43552.6786362669"
        },
        "miss_distance" : {
          "astronomical" : "0.0269024393",
          "lunar" : "10.46504879",
          "kilometers" : "4024547.75",
          "miles" : "2500738"
        },
        "orbiting_body" : "Earth"
      } ]
```


Figure 1: Example of the retrieved data in JSON format

Following asteroids properties will be extracted from the retrieved data:

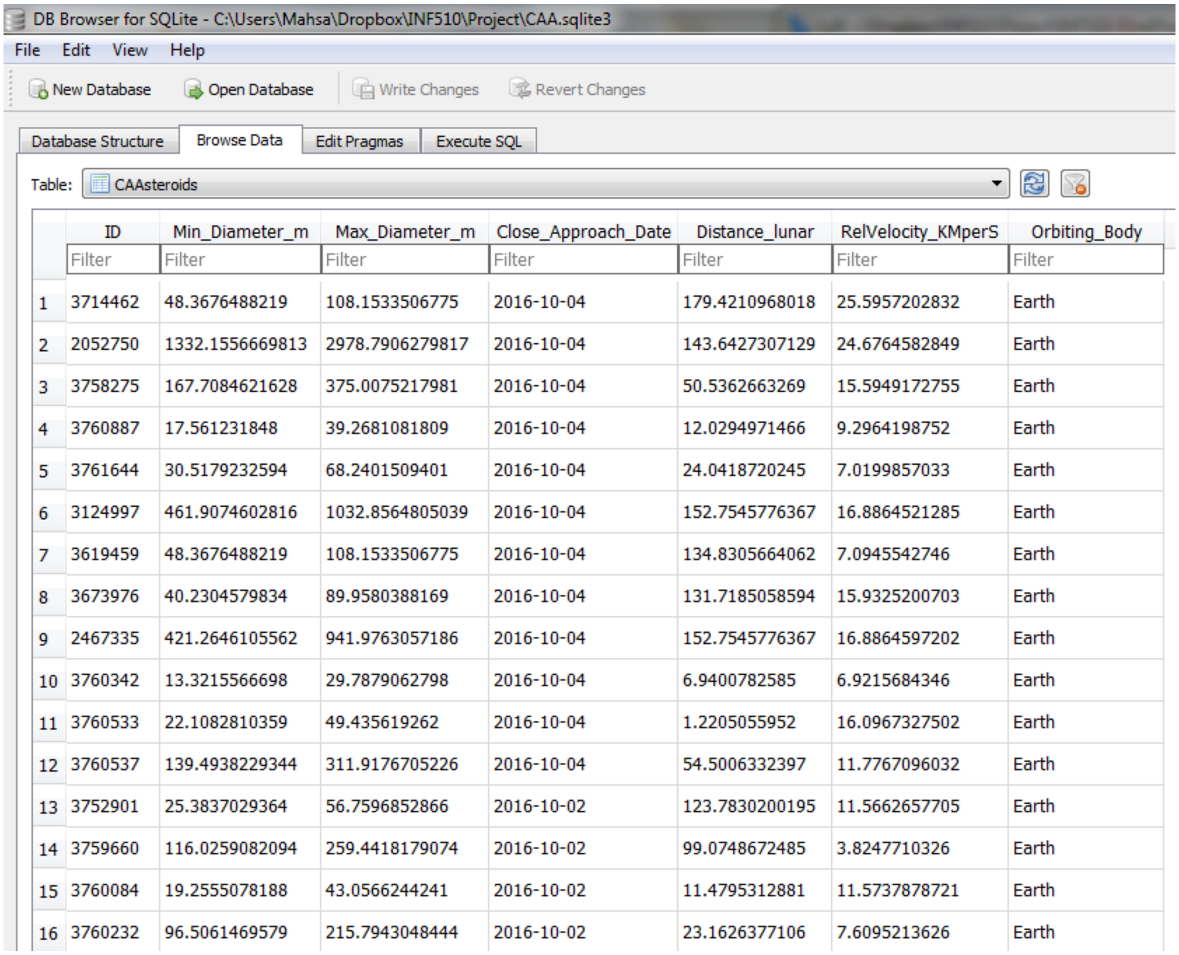
- Object ID
- Estimated Diameter_Min (m)
- Estimated Diameter_Max (m)
- Close Approach Date (YYYY-MM-DD)

- Distance (km):
- Relative Velocity (km/s)
- Orbiting Body

For instance:

- Object ID: 3763481
- Estimated Diameter_Min (m): 31.3584780571
- Estimated Diameter_Max (m): 70.1196886066
- Closest Approach Date (YYYY-MM-DD): 2016-11-03
- Distance (LD): 11.2066545486
- Relative Velocity (km/s): 27.6232344003
- Orbiting Body: Earth

Then, they will be inserted to the database using SQLite. Therefore, the table has the following columns: ID, Estimated_Diameter_Min, Estimated_Diameter_Max, Close_Approach_Date, Distance_lunar, RelVelocity_KMperS, and Orbiting_Body. The data associated with Object ID and Closest Approach Date will be used as composite primary key. The screenshot of the database stored in SQLite is presented in Figure 2.




| | ID | Min_Diameter_m | Max_Diameter_m | Close_Approach_Date | Distance_lunar | RelVelocity_KMperS | Orbiting_Body |
|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 3714462 | 48.3676488219 | 108.1533506775 | 2016-10-04 | 179.4210968018 | 25.5957202832 | Earth |
| 2 | 2052750 | 1332.1556669813 | 2978.7906279817 | 2016-10-04 | 143.6427307129 | 24.6764582849 | Earth |
| 3 | 3758275 | 167.7084621628 | 375.0075217981 | 2016-10-04 | 50.5362663269 | 15.5949172755 | Earth |
| 4 | 3760887 | 17.561231848 | 39.2681081809 | 2016-10-04 | 12.0294971466 | 9.2964198752 | Earth |
| 5 | 3761644 | 30.5179232594 | 68.2401509401 | 2016-10-04 | 24.0418720245 | 7.0199857033 | Earth |
| 6 | 3124997 | 461.9074602816 | 1032.8564805039 | 2016-10-04 | 152.7545776367 | 16.8864521285 | Earth |
| 7 | 3619459 | 48.3676488219 | 108.1533506775 | 2016-10-04 | 134.8305664062 | 7.0945542746 | Earth |
| 8 | 3673976 | 40.2304579834 | 89.9580388169 | 2016-10-04 | 131.7185058594 | 15.9325200703 | Earth |
| 9 | 2467335 | 421.2646105562 | 941.9763057186 | 2016-10-04 | 152.7545776367 | 16.8864597202 | Earth |
| 10 | 3760342 | 13.3215566698 | 29.7879062798 | 2016-10-04 | 6.9400782585 | 6.9215684346 | Earth |
| 11 | 3760533 | 22.1082810359 | 49.435619262 | 2016-10-04 | 1.2205055952 | 16.0967327502 | Earth |
| 12 | 3760537 | 139.4938229344 | 311.9176705226 | 2016-10-04 | 54.5006332397 | 11.7767096032 | Earth |
| 13 | 3752901 | 25.3837029364 | 56.7596852866 | 2016-10-02 | 123.7830200195 | 11.5662657705 | Earth |
| 14 | 3759660 | 116.0259082094 | 259.4418179074 | 2016-10-02 | 99.0748672485 | 3.8247710326 | Earth |
| 15 | 3760084 | 19.2555078188 | 43.0566244241 | 2016-10-02 | 11.4795312881 | 11.5737878721 | Earth |
| 16 | 3760232 | 96.5061469579 | 215.7943048444 | 2016-10-02 | 23.1626377106 | 7.6095213626 | Earth |

Figure 2: Example of the table stored in SQLite Database

## 3 Software Packages

Packages used in my code and a brief description of their functionality are listed below,

**urllib:** is a package for fetching data across the World Wide Web. It can open, read and generally access Universal Resource Locators (URLs). This package is used in the code to open and read NASA Neo URL and eventually retrieve data using NASA Neo API.

**json:** is used to load data, retrieved from NASA Neo API, in a JSON format.

**sqlite3:** is used to create a database and tables in "DB Browser for SQLite" and to store the data in the tables.

**pandas:** is a cross-section and time series data analysis tool that enables Python programming language users to construct data structures and perform data analysis. This package is used in the code to construct dataframes and perform analysis on dataframes. For examples, the data from SQLite table is inserted to a dataframe using pandas package.

**matplotlib.pyplot:** is a two-dimensional plotting package inside matplotlib library that is used in this code to visualize statistical analysis such as, plotting histogram, pie chart, scatter matrix.

**scatter_matrix:** is package in pandas (specifically from pandas.tools.plotting) that enables users to construct and visualize scatter plot matrix.

**statsmodels.api:** is a package in statsmodels that provides classes and functions for the estimation of statistical models. It is used in this code to obtain the non-parametric probability distribution function (PDF) of the variables (i.e. data such as diameter, distance and relative velocity) and consequently calculating the entropy associated with the data. Moreover, it is used to output a table in a txt file containing the statistics of the data.

## 4 Instructions, Inputs and Outputs

Users are asked to input the start and end dates. Then, a list of Asteroids with corresponding properties that are in their closest distance to Earth, within the requested date range, will be obtained in JSON format using NASA API for Neo dataset. Therefore, the code first asks the dates from users using raw_input command as shown in Figure 3. Users are allowed to input dates with maximum 7 days range in YYYY-MM-DD format. Otherwise, the code will output a message to guide users entering the right format.

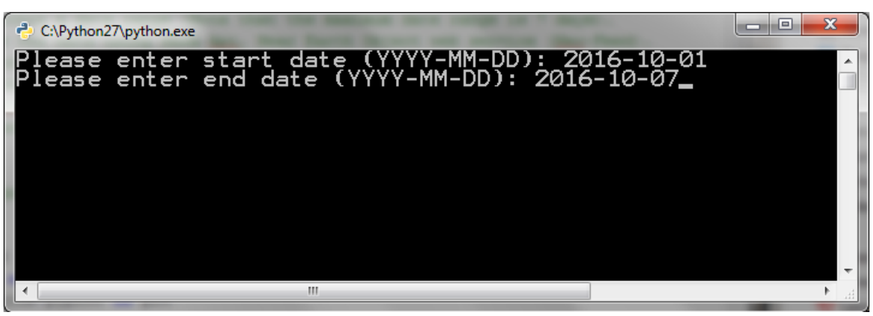



Figure 3: Example of the required inputs to run the code asked from users

Then, the URL will be constructed and printed which looks like the following:

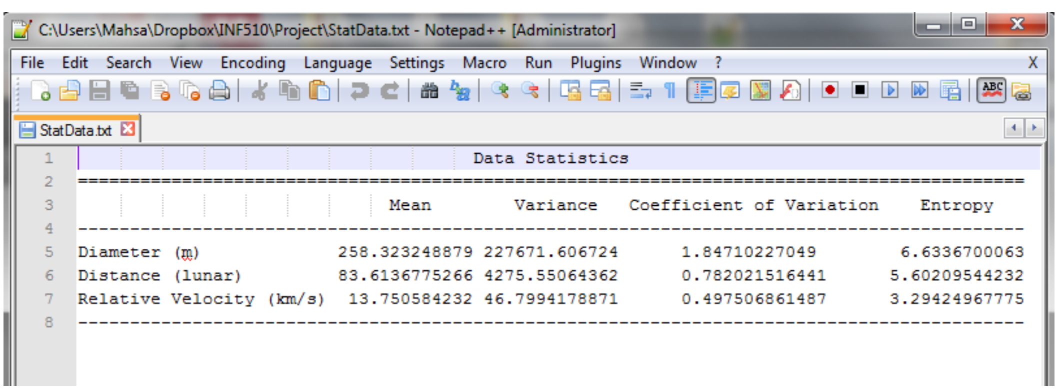
https://api.nasa.gov/neo/rest/v1/feed?start_date=START_DATE&end_date=END_DATE&api_key=API_KEY

For example:

https://api.nasa.gov/neo/rest/v1/feed?start_date=2015-09-07&end_date=2015-09-08&api_key=DEMO_KEY

Next, data will be retrieved from NASA Neo API. There is a comment in the code to print the JSON data (print json.dumps(js, indent=4)) which can be uncommented in case users are interested in observing the retrieved data. Then, some of the retrieved data will be collected to be stored in a database named "CloseApproachAsteroids.sqlite3" in a table named "CAAsteroids". The collected data are as follows: Object ID, Estimated Diameter_Min (m), Estimated Diameter_Max (m), Close Approach Date (YYYY-MM-DD), Distance (lunar), Relative Velocity (km/s) and Orbiting Body (please see Figure 2). Next, the table in the database will be inserted to a dataframe using pandas to perform the desired analysis. The analysis and the corresponding outputs are listed below:

- The diameter of each retrieved object (asteroid) will be estimated by calculating the average of "Estimated Diameter_Min" and "Estimated Diameter_Max" data for each row and a new column will be added to the dataframe, named Diameter_m.
- The statistics of the diameter (Diameter_m), distance (Distance_lunar) and relative velocity (RelVelocity_KMperS) of the asteroids will be evaluated and printed into a table in a text file, named "StatData.txt". The created table as shown in Figure 4 contains information about the mean, variance, coefficient of variation and entropy of the diameter, distance and relative velocity. For calculating entropy values, the estimation of non-parametric PDFs (kernel density estimation) is used.



Data Statistics

| | Mean | Variance | Coefficient of Variation | Entropy |
|---|---|---|---|---|
| Diameter (m) | 258.323248879 | 227671.606724 | 1.84710227049 | 6.6336700063 |
| Distance (lunar) | 83.6136775266 | 4275.55064362 | 0.782021516441 | 5.60209544232 |
| Relative Velocity (km/s) | 13.750584232 | 46.7994178871 | 0.497506861487 | 3.29424967775 |

Figure 4: Statictics of the asteroids data (output file named StatData.txt)

- To better visualize the statistics of the data, the histogram of the diameter, distance and relative velocity of the asteroids together with the corresponding Box plot will be plotted. Figure 5 shows an instance of the aforementioned plot.

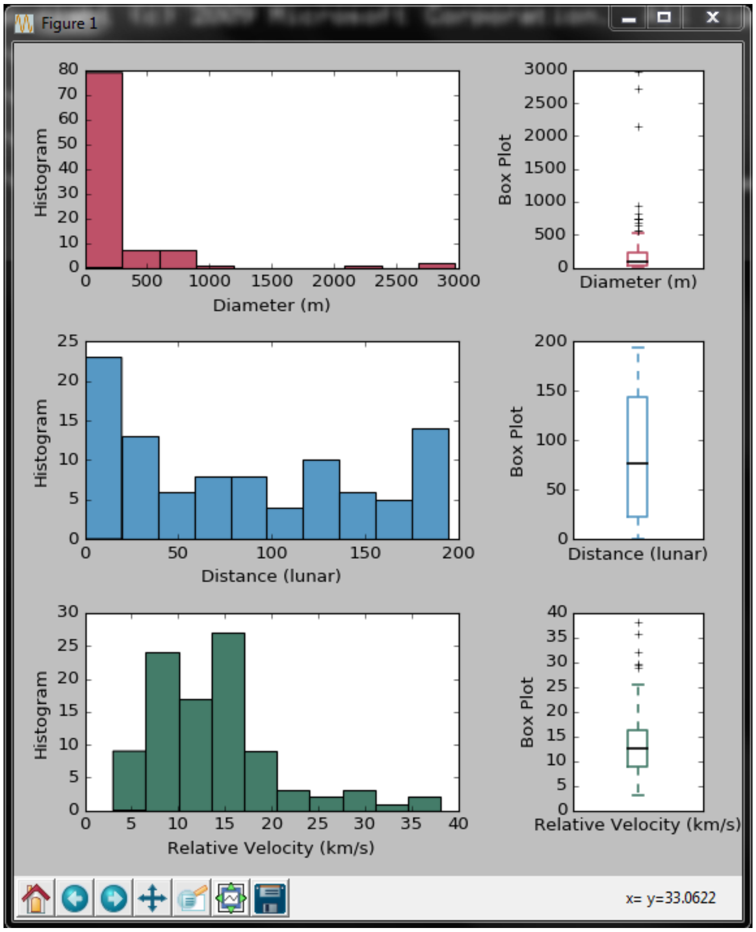



Figure 5: Histogram of the distance, diameter and the relative velocity of the asteroids.

- Scatter plot matrix for the distance, diameter and relative velocity of the asteroids will be plotted. The kernel-estimated PDF is also presented in the main diagonal of the scatter matrix. The scatter matrix reveals whether distance, diameter and relative velocity values are correlated. An example of the scatter matrix plot is shown in Figure 6.

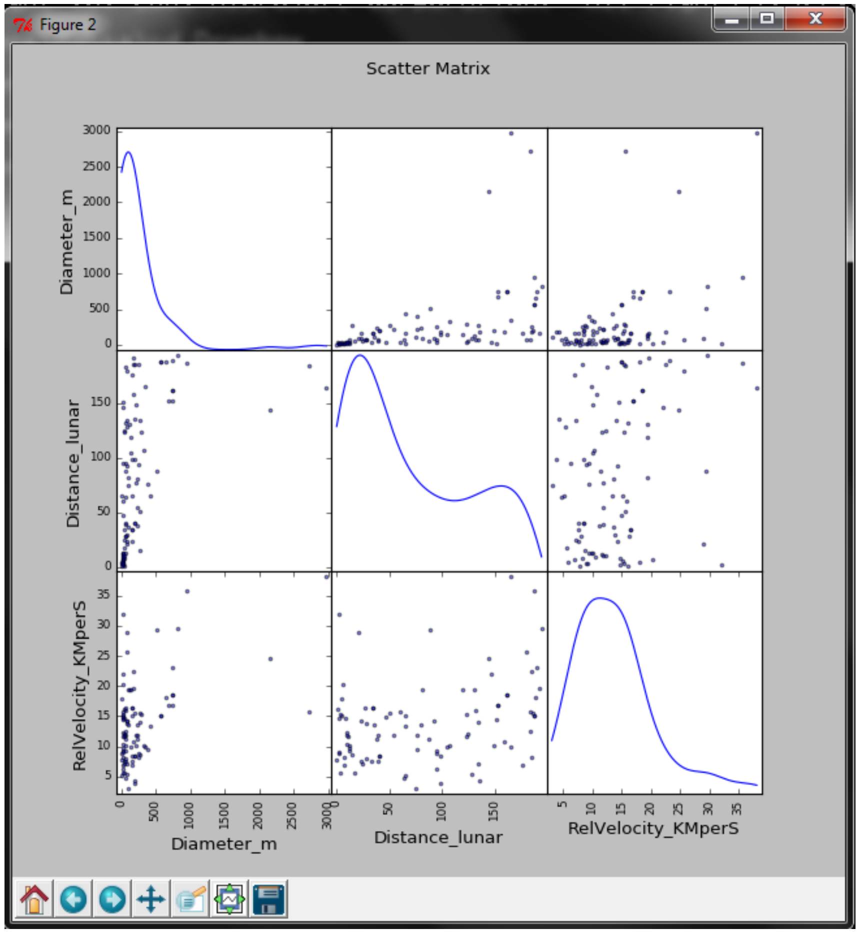



Figure 6: Scatter Matrix of the diameter, distance and relative velocity of the asteroids.

- Pie chart of the close approach dates of the asteroids within the requested range will also be plotted to visually present the number of asteroids in their closest approach for each day relative to the total number of retrieved asteroids. Figure 7 illustrates an example of the pie chart mentioned above.

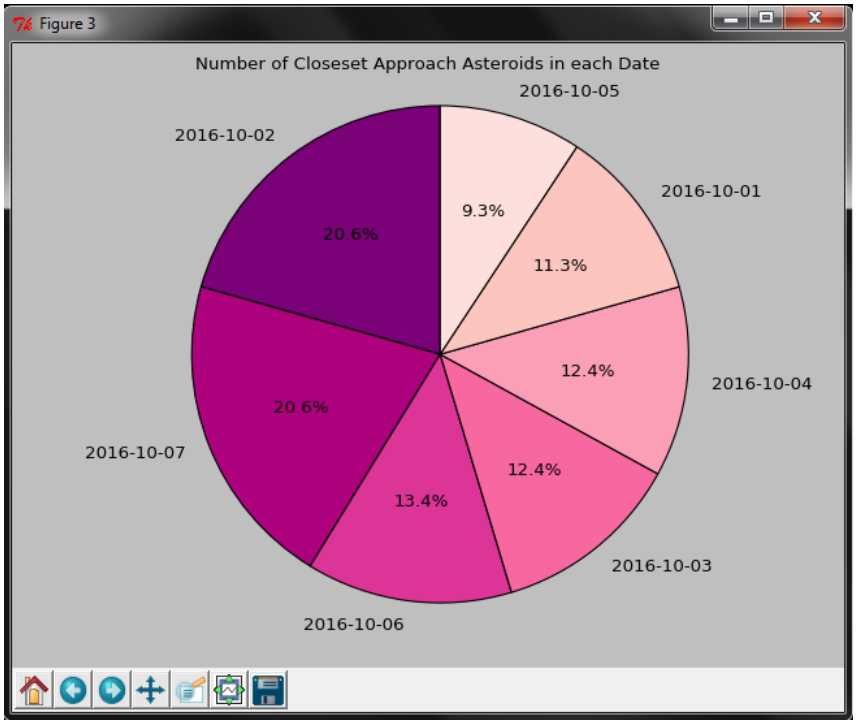



Figure 7: Pie chart for the number of closest approach asteroids for each date.