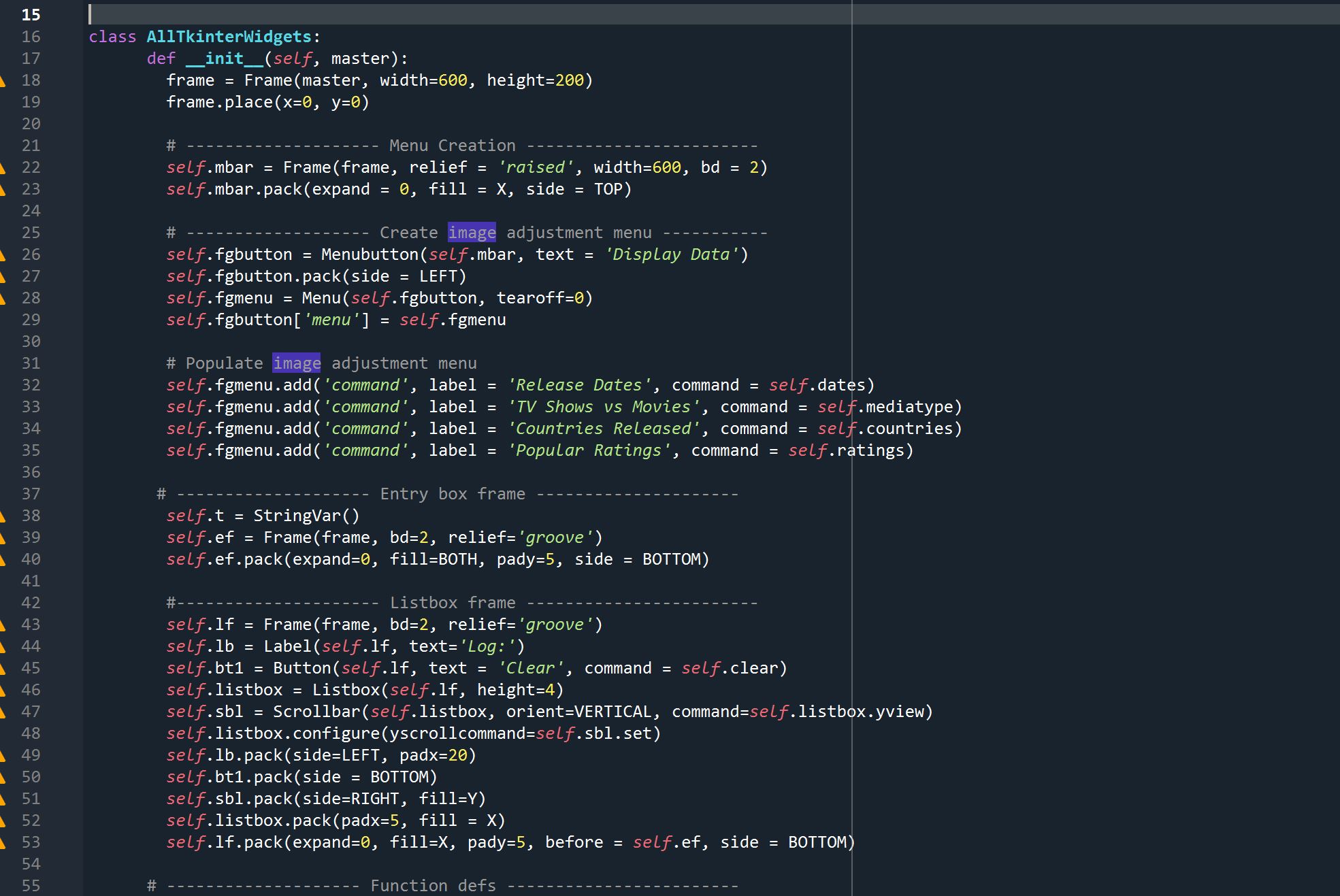
Code Explanation PDF

Eric Jones, Charlotte Komrosky-Licata

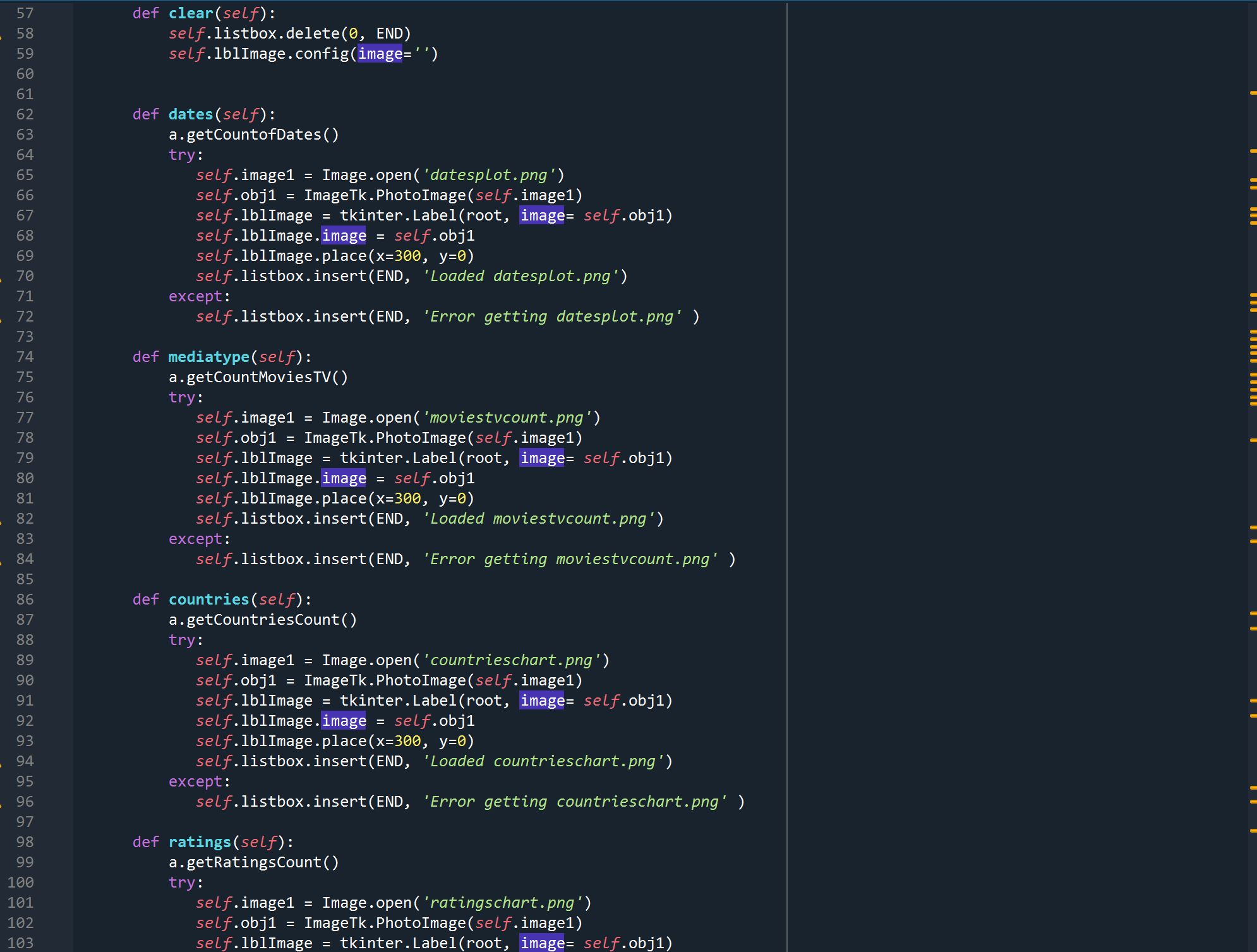
**Main.py**



Initially creates and displays a frame spanning across the entire screen.

Also creates and displays a menu with which the user can interact and choose four different charts to display from menu buttons called ‘Release Date’, ‘TV Shows vs Movies’, ‘Countries Released’, and ‘Popular Ratings’ which reference **dates(self), mediatype(self), countries(self)**, and **ratings(self)** as commands.

Users can also clear the log and the screen to choose a different chart using the ‘Clear’ button which is created and formatted in the listbox section along with a field called ‘Log’ which displays output in a text field.



Local Functions:

**clear(self)** is a command used in the listbox frame section of menu creation which clears the listbox and images from the frame.

**dates(self), mediatype(self), countries(self)**, and **ratings(self)** functions test to see whether the images created in their correlating functions from utilities.py will open, and load confirmation as a string in the log box. These functions are referenced as commands in the menu button section, and each has its own button so the user can choose which data to see.

**dates(self)** calls getCountofDates() from utilities.py to display the created image.

**mediatype(self)** calls getCountMoviesTV() from utilities.py to display the created image.

**countries(self)** calls getCountriesCount from utilities.py to display the created image.

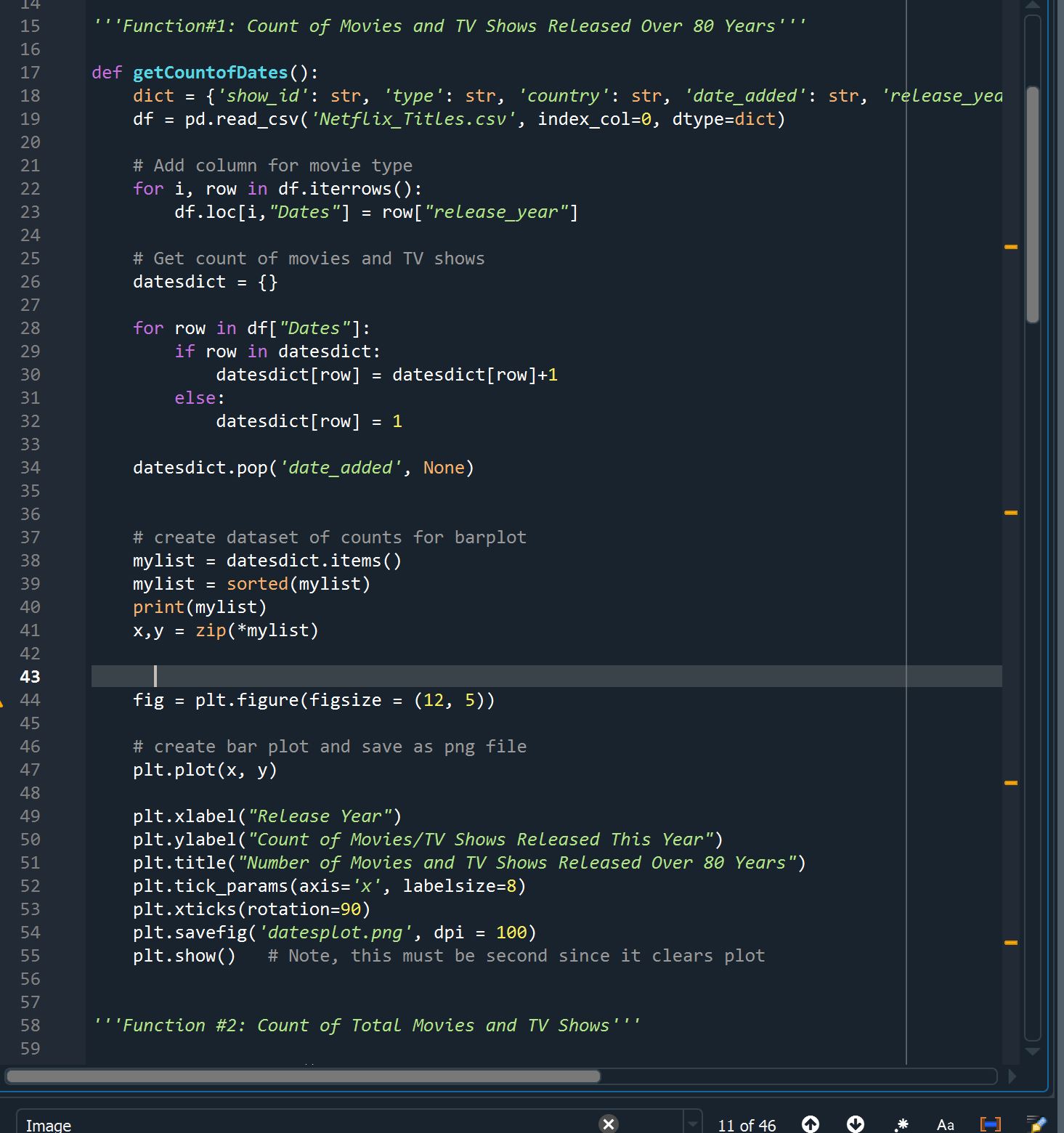
**ratings(self)** calls getRatingsCount from utilities.py to display the created image.

**#main**

Displays the functions, runs the tk object and alltkinterobjects class and initiates main.py

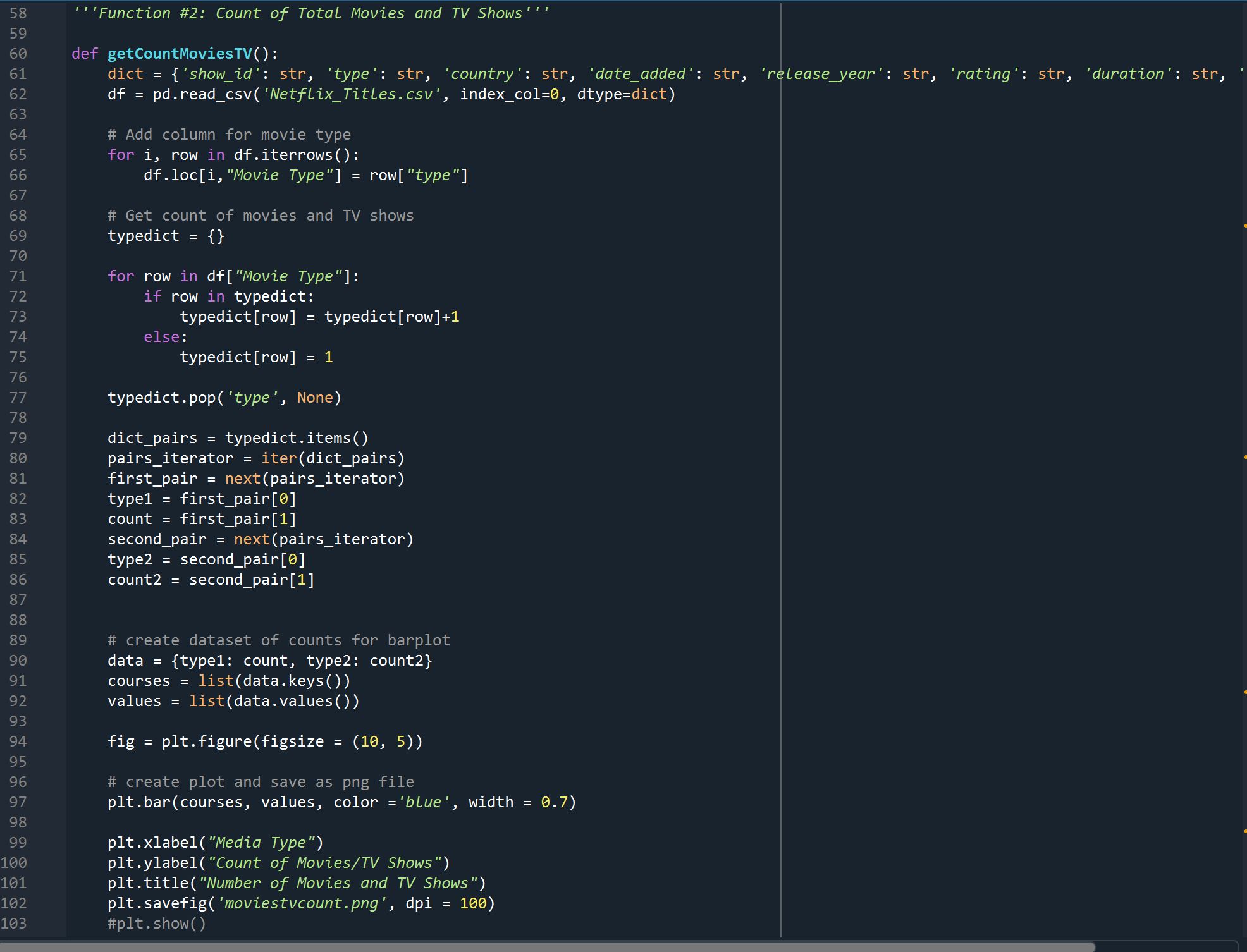
**Utilities.py**

**getCountsofDates()**



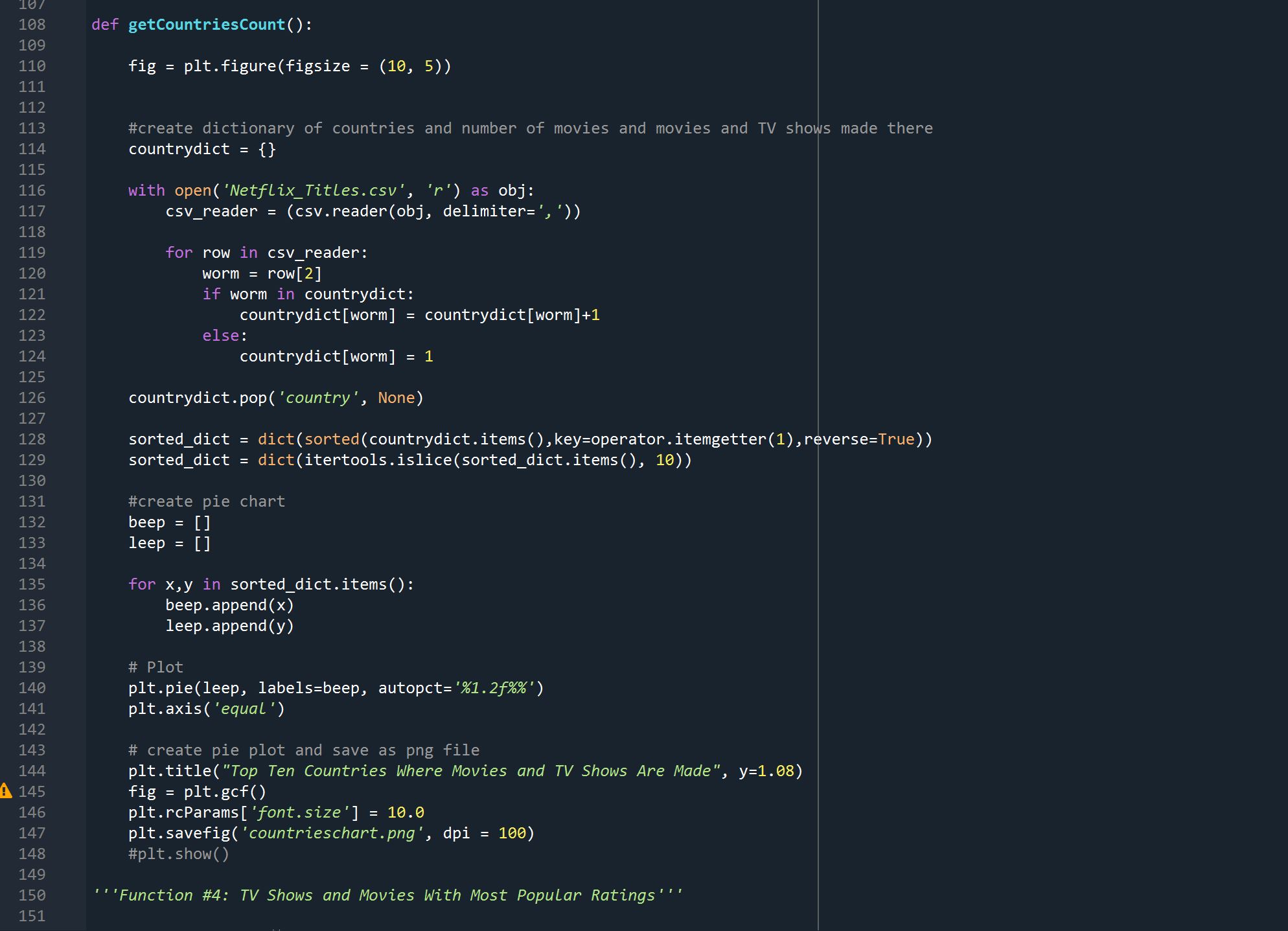
This function uses pandas to create a dictionary so that the Netflix csv file can be ordered by each column of the data. We then use a for loop to get all the data in the “release\_year” column. Then we use a for loop to get a count of the movies and tv shows. Then using our dictionary to see what years these movies and tv shows were added. Next we added a list to count the amount of movies and tv shows in each date\_added column. Then Matplotlib is used to create a bar chart showing our results.

**getCountMoviesTV()**



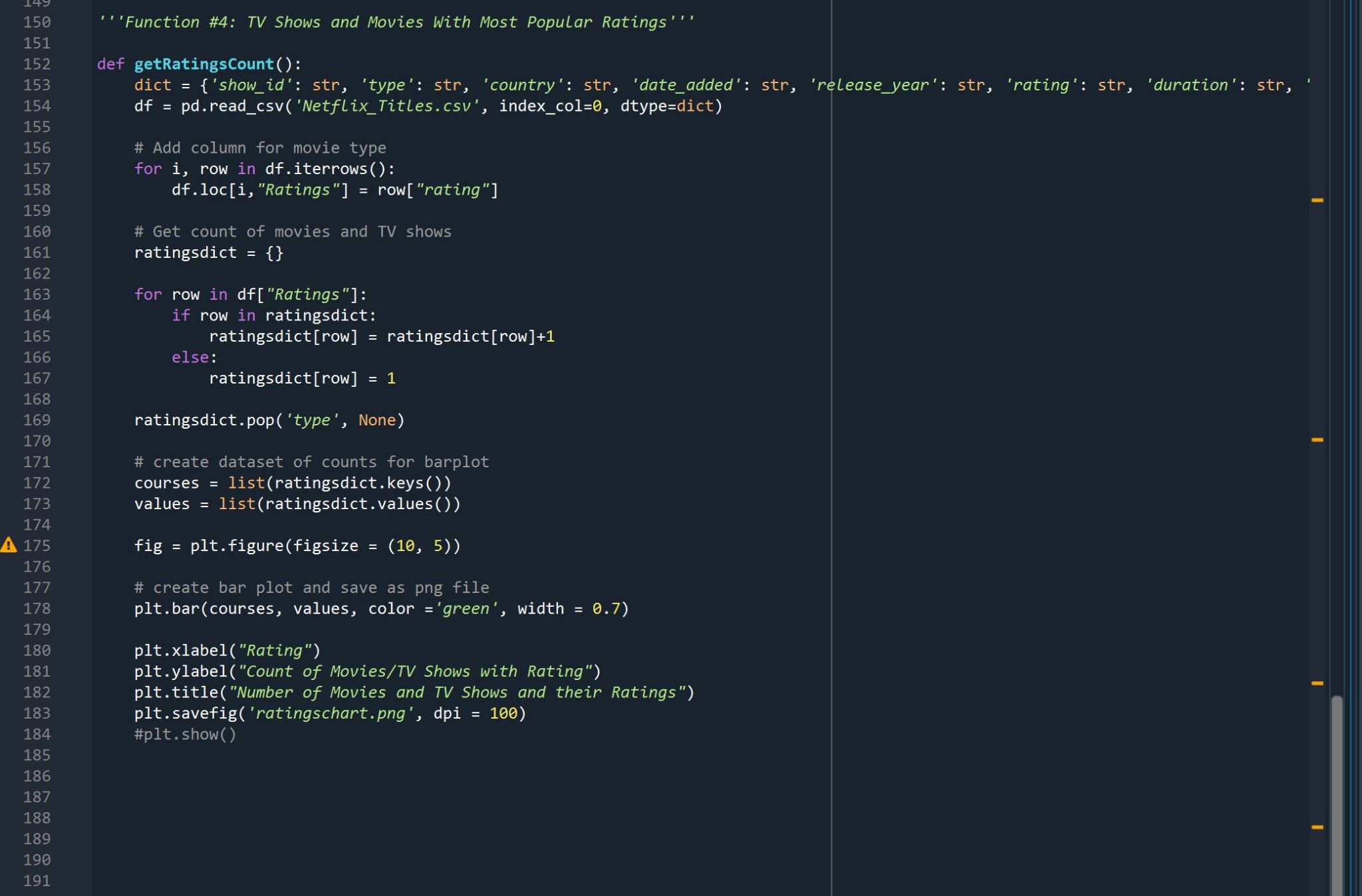
This function also uses pandas to create a dictionary to read the Netflix csv file. Then a for loop is used to select the type which specifies if each object is a movie or a tv show. The type is then used in a for loop to count the number of movies and tv shows. We then assign the data to type1 and type2 to iterate the movies and tv shows. Then using these types we create a dataset of counts for the barplot. Then the barlot is made using matplotlib and saved as an image.

**getCountriesCount()**



This function iterates over lines in the ‘Netflix\_Titles.csv’ file and designates country names as keys and the count of movies and TV shows made in each country as values. From there, the dictionary is sorted by value in reverse order (from most TV show/movies made in each country to least) and the dictionary is split so it only includes the top ten countries. Then, the pairs are split into two lists which are then used to create a pie chart. The chart is titled “Top Ten Countries Where Movies and TV Shows Are Made”, and the figure is saved as a png.

**getRatingsCount()**



This function creates a figure on which to put a bar chart, then iterates through the lines in the Netflix\_Titles.csv’ file and creates dictionary key, value pairs for ratings and a count of the number of movies and TV shows which have that rating. These dictionary pairs are then turned into two lists which are used and plotted as the x and y axes in a bar chart. The chart’s x-axis is then titled ‘Ratings, y-axis is titled “Count of Movies/TV Shows With Rating”, and the chart is then titled “Number of Movies and TV Shows and Their Ratings”. Finally, the function saves the chart as a png image.