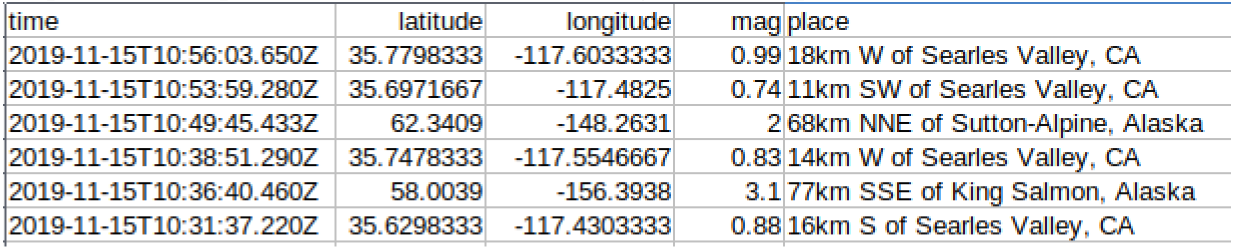
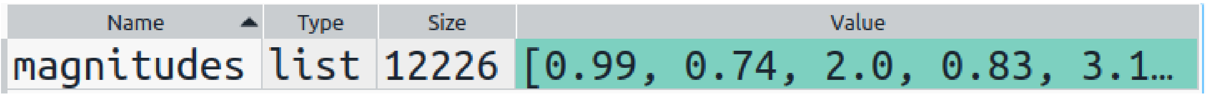
Exercise 1: Histogram, Box Plot and Violin Plot of the Earthquake Magnitudes

The file `“earthquakes_2019.csv”` (available from Moodle) contains the details of earthquakes over a period of a month. The following is an excerpt:



time	latitude	longitude	mag	place
2019-11-15T10:56:03.650Z	35.7798333	-117.6033333	0.99	18km W of Searles Valley, CA
2019-11-15T10:53:59.280Z	35.6971667	-117.4825	0.74	11km SW of Searles Valley, CA
2019-11-15T10:49:45.433Z	62.3409	-148.2631	2	68km NNE of Sutton-Alpine, Alaska
2019-11-15T10:38:51.290Z	35.7478333	-117.5546667	0.83	14km W of Searles Valley, CA
2019-11-15T10:36:40.460Z	58.0039	-156.3938	3.1	77km SSE of King Salmon, Alaska
2019-11-15T10:31:37.220Z	35.6298333	-117.4303333	0.88	16km S of Searles Valley, CA

The Python program `mpl_earthquakes.py` retrieves the data from the file and returns a list of the earthquake magnitudes:



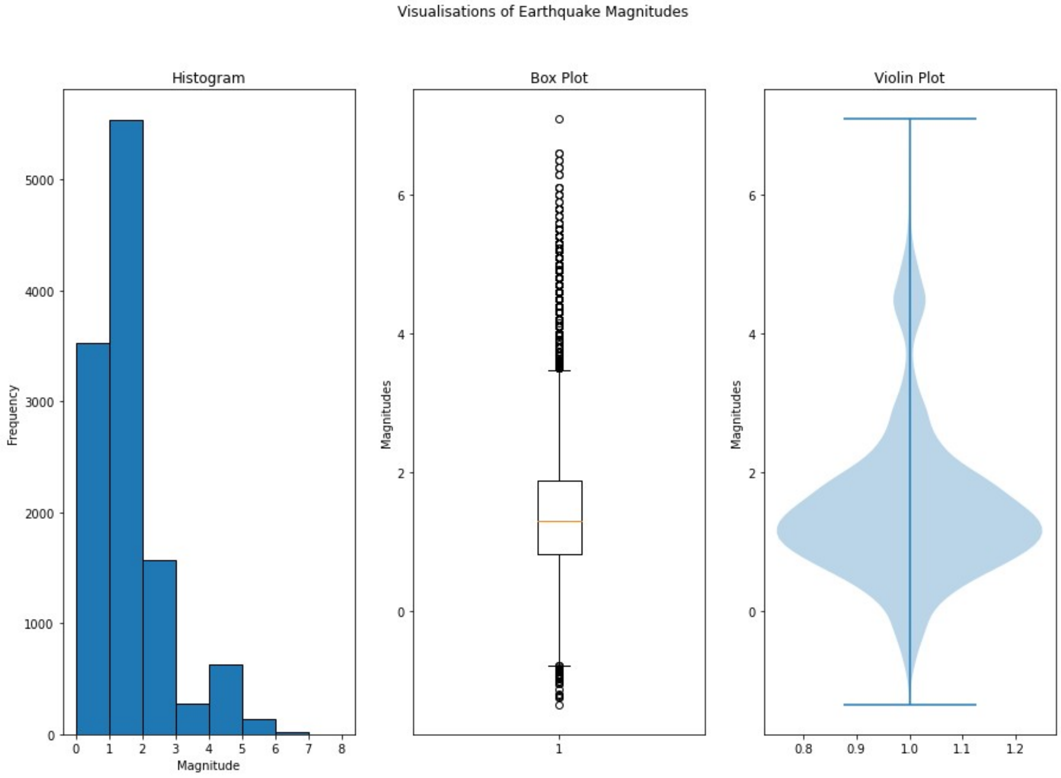
Name	Type	Size	Value
magnitudes	list	12226	[0.99, 0.74, 2.0, 0.83, 3.1…

Modify the program to display the following:

- A histogram
- A box plot
- A violin plot

of the earthquake magnitudes.

Sample Visualisation



Guidelines

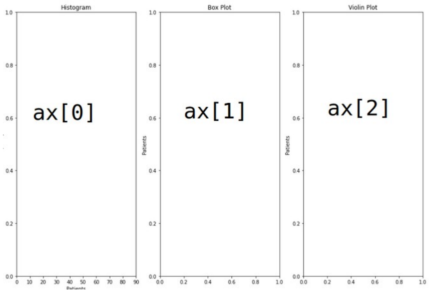
Create a Figure `fig` and Axes `axs` which can show the three plots in a row:

```
    fig, axs = plt.subplots(1,3, figsize=(15,10))
```

`(1,3)` means 1 row and 3 columns. If you want 3 rows and 1 column, use `subplots(3)`

You can set a title for the figure using `fig.suptitle(`*`text`*`)`

You'll refer to the 3 plots as `axs[0]` `axs[1]` and `axs[2]`



For the box plot, use the function `boxplot()`
with keyword arguments `showmeans` and `meanline` set to `True`

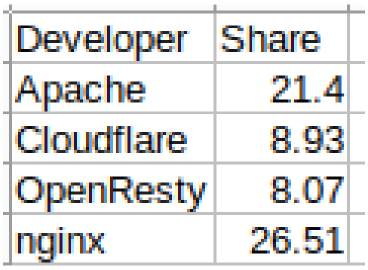
For the violin plot, use the function `violinplot()`
with keyword argument `showmeans` to `True`

For the histogram, you need to specify the groups for the bars, called *bins*:

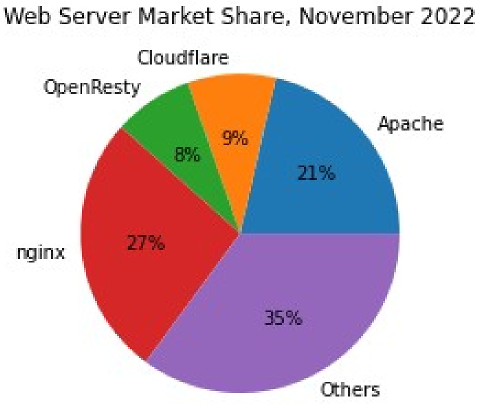
1. Create a list for the `bins` using `range(int(max(magnitudes)+2)`
2. Set the `xticks` to the `bins` list
3. Display the histogram using the function `hist` with `magnitudes` and `bins`.
   For greater visibility of the bars, use `ec="black"` (`ec` stands for edge colour).

Exercise 2: Web Server Market Share

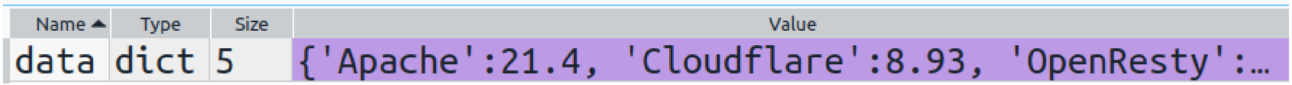
The file `webservers_202211.csv` contains information on the market share of the most popular web servers from November 2022 (https://news.netcraft.com/archives/category/web-server-survey/)



Developer	Share
Apache	21.4
Cloudflare	8.93
OpenResty	8.07
nginx	26.51



The program `mpl_webservers.py` retrieves the information from the file and provides the data in the following dictionary:



Name ▲	Type	Size	Value
data	dict	5	{'Apache':21.4, 'Cloudflare':8.93, 'OpenResty':…

Modify the program so that it creates a Pie Chart to visualise the data, as follows:

1) Import `matplotlib.pyplot` as `plt`

2) Create a Figure `fig` and Axes `ax` using `plt.subplots()`.

3) Set the title, for the Axes `ax`.

4) Create the pie chart, including the dictionary keys as labels and percentage share on each slice:

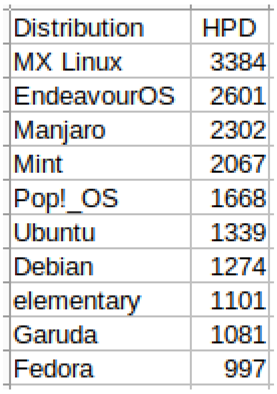
```
    ax.pie(dict.values(),labels=dict.keys(),autopct="%.f%%")
```

5) Show the plot (if not using Spyder)

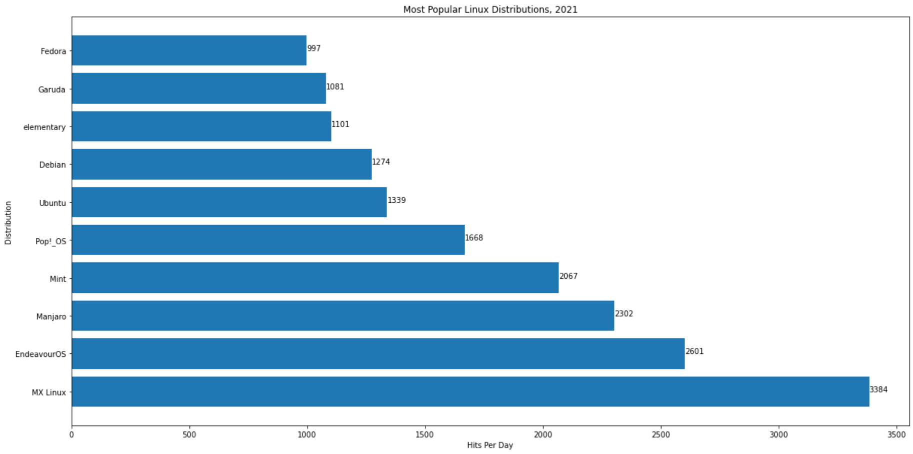
6) Save the figure and upload it to Moodle.

Exercise 3: Popularity of Linux Distributions

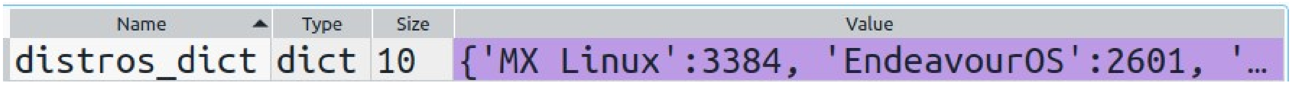
The file `distrowatch_2021.csv` contains information on the Top 10 most popular Linux Distributions from 2021 (https://distrowatch.com/index.php?dataspan=2021)



Distribution	HPD
MX Linux	3384
EndeavourOS	2601
Manjaro	2302
Mint	2067
Pop!_OS	1668
Ubuntu	1339
Debian	1274
elementary	1101
Garuda	1081
Fedora	997



The program `mpl_distrowatch.py` retrieves the information from the file and provides the data in the following dictionary:



Name ▲	Type	Size	Value
distros_dict	dict	10	{'MX Linux':3384, 'EndeavourOS':2601, '…

Modify the program so that it creates a Horizontal Bar Chart to visualise the data, as follows:

1) Import `matplotlib.pyplot` as `plt`

2) Create a Figure `fig` and Axes `ax` using `plt.subplots()`.

3) Set the title, x-axis label and y-axis label for the Axes `ax`.

4) Create the bar chart, using the dictionary keys as labels:

```
ax.barh(list(dict.keys()),dict.values())
```

5) Display the values at the end of the bars:

```
for index, value in enumerate(dict.values()):
    ax.text(value,index,str(value))
```

6) Show the plot (if not using Spyder)

7) Save the figure and upload it to Moodle.