

VISUALISING DATA

⚙️ Install plotly

- 🔍 **What it is?** An external library for data visualisation.
- 💡 **Why do I need it?** It makes interactive visualisations.
- 🔗 **Where do I get it?** Into your Terminal, type `pip install plotly`

⚙️ Install nbformat

- 🔍 **What it is?** An external library for data visualisation in Jupyter notebooks.
- 💡 **Why do I need it?** Allows plotly to display interactive plots in notebooks.
- 🔗 **Where do I get it?** Into your Terminal, type `pip install nbformat`

💡 Types of Plots in plotly

🔗 bar chart

```
fig = px.bar(  
    [dataframe],  
    x=[col with vals for x axis],  
    y=[col with vals for y axis],  
    title=[title for whole graph],  
    color=[col to be distinguished by colour],  
)  
  
fig.show()
```

🔗 line chart

```
fig = px.line(  
    [dataframe],  
    x=[col on x],  
    y=[col on y],  
    color=[col name], ## optional; creates separate lines for each unique val in col  
    title=[title for whole graph],  
    labels={ [col name]: [label on chart], [col name]: [label on chart] },  
    markers=True ## add markers to the line  
)  
  
fig.show()
```

🔗 scatter plot

```
fig = px.scatter(  
    [dataframe],  
    x=[col on x],  
    y=[col on y],  
    title=[title for whole graph],  
    labels={ [col name]: [label on chart], [col name]: [label on chart] },  
    color=[col], ## colours points based on unique vals in col  
    hover_data=[list of col names] ## provides additional data  
                                from each of the listed cols upon hover  
)  
  
fig.show()
```

🔗 pie chart


```
fig = px.pie(  
    [dataframe], ## rows in this table represent pie slices  
    values=[col], ## values represent size of each pie slice  
    names=[col], ## label for each pie slice  
    title=[title for whole graph], ## label for the chart  
    hole=0, ## replace w a bigger value (e.g., 0.4) for a donut chart  
    hover_data=[list of col names] ## additional info on hover  
)  
  
fig.show()
```

This handout only contains a tiny fraction of what plotly can do, but you can always learn more [here](#) 🔗

DOWNLOADING DATABASES



Fetching Databases from Kaggle

- 1 Install the Kaggle library by running `pip install kaggle` in your Terminal
- ↓
- 2 Make yourself a Kaggle account [here](#) 
- ↓
- 3 On [kaggle.com](#), go to 'settings' (profile picture → 'settings')
- ↓
- 4 Under API, hit 'create token' (this will download a `.json` file)
- ↓
- 5 In your local user folder (`C:\Users\username`), create a folder named `.kaggle` (with a dot)
- ↓
- 6 move your `kaggle.json` file inside this folder
- ↓
- 7 go on [kaggle.com](#), hit 'datasets' (left bar), browse the available datasets, and find one you like
- ↓
- 8 copy the directory of your chosen dataset (`username/dataset_name`, at the end of the URL)
- ↓
- 9 In your Terminal, run `kaggle datasets download -d username/dataset_name`
- ↓
- 10 Find the `.zip` file the above command downloaded and open it in File Explorer (right click + 'Reveal in File Explorer' OR `shift + alt + r`)
- ↓
- 11 Right click on the `.zip` file in File Explorer and hit 'Extract All'
- ↓
- 12 Copy the `.csv` file inside the extracted folder and paste it in the folder of your Jupyter notebook