MatplotLib

X and y axis must have same first dimention

- from matplotlib import pyplot as plt
- plt.plot(x_axis, y_axis) ← Plotting data
- plt.show() ← showing a transformation on plt
- plt.title(' ') ← title on data visialization (give it a titl
- plt.Xlabel(' ') ← Label X_axis

two ways ← add legend

① add label for legand as a list of values and order that there were added to the plot

ex:- plt.legand(['All Devs', 'Python'])

Plot ترجم اوانت عالك Line أول واحد تخصيص بأول

② Pass a label argument to the plot methods

plt.plot(x_axis, y_axis, Label = 'All_Devs')

not Likely

① fmt = '[Marker][Line][Color]' ← Change formatting of the plot

↳ format string Consits of a Part for Color, marker & line

ex: plt.plot(ages_x, dev_y, 'k--', Label = 'All_Dev')

↳ [black Color -- dashed line style]

② another way

plt.plot(x_axis, y_axis, color = 'k', linestyle = '--', marker = '.', Label = 'All_Devs')

plt.bar(x_axis, y-axis) ← vertical bar chart
plt.barh(x_axis, yaxis) ← horizontal bar chart

to read from csv file

```
with open('data.csv') as csv_file:
    csv_reader = csv.DictReader(csv_file)
    language_Counter = Counter()
    For row in csv_reader:
        language_Counter.update(
            row['languagesworkedwith'].split(';'))
```

another method by Pandas:

```
data = pd.read_csv('data.csv')
ids = data['Responder-id']
Lang_responses = data['LanguagesWorkedWith']
Language_Counter = Counter()
for response in lang_responses:
    language_Counter.update(response.split(';'))
```

Video 3 MatPlotLib Corey Schafer ☑ Pie Chart

instead of plotting certain values for each Category you can think of Chart as a pie & each Category as a slice of this pie

---

→ plt.pie (List of values)

→ add Labels

plt. pie (List of values , List of labels

→ add separators between slices

Pass some wedge properties

plt.pie(list , list of labels , wedgeprops = {'edgeColor': 'bl

→ specify pie Colors

Create new list of Colors

& add this to plt.pie as (Colors = List of Colors

→ Pie Chart not good when you have much data

it is best for comparing 5 or 6 things with other

bar chart is better in this Case

explode slice from Pie يعني تخليها طايرة جزئية عن بقية ال Pie

دي اللي عايز تعملها explode

explode = [0, 0, 0.1, 0]

10% من الـ slice هيطلع برة explode

in plt.pie(explode = explode

→ passing shadow argument in plt.pie

```
plt.pie ( shadow = True )
```

→ rotate Pie

```
plt.pie ( startangle = 90)
```

→ display Percentage of each slice

```
plt.pie( autopct = '% 1.1f %%')
```

---

video 4 stacked plots (area chart) ✓

allow us see different proportions over time

see distribution of point in specifice minite

بستفيد منها إن انا بعرف مع الوقت الحاجة دى مشيت ازاى

```
plt.pie( [1,1] , labels = ["player 1", 'player 2']
```

pie chart

```
plt.stackplot( list contain (time) , player 1 , playerlist )
                minute / year          list      2
```

add labels as a list to the labels argument

add legand بتوضح ال total ايه وكل حاجة لوحدها خدت قد ايه

```
plt.legand ( loc = 'upper left')
plt.legand ( loc = (0.07 , 0.05))
```

تغير مجه

video 5 ☑ Filling Area on line plots

add fields to plots @ Conditional fields

Show what areas.. for a line plot
for above & below thrash hold

```
plt.fill_between (ages , py_salaries, overall_median = 57200
                  , where = (py_salaries > overall_median)
                  , interpolate = True , color = 'red', alpha = 0.25)
```

Histograms

بنرسم - Column واحد ونشوف عدد الـ frequency

List of

```
plt.hist (ages , edgeColor = 'black', bins = 8,
```

لو عندي قيم كبيرة أوي وقيم صغيرة أوي وعايز الـ data تبان
هبدأ أعمل scale لـ log

```
ages_column = df['Age']
plt.hist (ages_Columns, log = True)
```

- good for visualizing distribution data where The data falls within certain boundaries
- groups data up into bins instead of plotting each individual value

video 7 ☑ scatter plot

t.style.use ('seaborn') ← anothe style
plt.tight-layout ← add some automatic padding to plots

---

plt.scatter (x_list , y_list)

scatter nice for seeing different trends or outliers

since scatter plots random this mean there is no Correlation between two lists

بدل النقط هتبقى x

Customizations ↗ size ↗ color ↗
plt.scatter (x, y, s= 100 , c = 'green' , marker='x')

good to add edges to circle & give it alpha

plt.scatter(edgecolor = 'black', linewidth=1, alpha=0.75

* How colors & sizes can be mark basis

Colors =[7,5, 8, 9]
أول قيمة بال 2Lists اللي كاملة بيعم ال scatter → has a color of 7

Color map options to change colors →

Colormap it's argument is cmap

```
plt.scatter( cmap = 'Greens')
```

Capital I

Making Colorbar

add Label for Color map

```
Cbar = plt.Colorbar() ← method
Cbar.set-label ('Satisfaction')
```

ممكن تبعت ال size بتاع ال scatter plots عشان يكونوا List عشان يبقوا

كلهم نفس الحجم

use log scale for axis to make scatter plot look a bit better

```
plt.xscale('log')
plt.yscale('log')
```

Video 8 ☑ Plotting Time Series Data

plt.plot_date(dates, y)

dates → list of dates
y → y_axis list of random variables

```
dates = [
    dateTime(2019,5,24)
    dateTime(2019,5,25)
    ==
]
y = [0,1,3,4,6,5,7]
```

plot dates
↳ plt.plot_date(dates, y)

*Customizations
plt.plot_date(linestyle='solid')

* running auto format x date method will rotate date if it abit nicer & change thier alignments & things like that
get current figure
↳ plt.gcf().autofmt_xdate()
run auto format method
figuers & axis in subplot

* change format of the dates you need to → from matplotlib import dates as mpl_dates

From imported module you can use date formatter class & passing any format string that you can pass into strf time method from date/time class

date_format = mpl_dates.DateFormatter('%b,%d %Y')

→ grab the axis to run format method similar to

plt.gca().xaxis.set_major_formatter(date_format

get Current access

get Current figure

☆ Convert date column to a date time

data = pd.read_csv('data.csv')

data['Data'] = Pd.to_datetime(data['Date'])

data.sort_Values('Date', inplace=True)

To create Live graph 3 steps:

1- Initialize plot fig, ax = plt.subplots()

2- update Data:- update plot with new data points at regular intervals

3- Animate Plot;- use Animation technique to refresh the plot in real-time

From matplotLib import FuncAnimation

subplots :- Tool for Create & organize multiple plots

Syntax → fig, axes = plt.subplots(nrows , nCols

nrows : Number of rows of subplot
nCols: Number of Columns of subplots

fig → figure object represent entire figure

axes → single "Axes" object (nrows=1, nCols = 1)
↳ array of "Axes" object ((nrows , nCols) > 1)

fig, (ax1, ax2) = plt.subplots (nrows = 1,
nCols = 2,
figsize = (10, 4))