

INFO5002: Intro to Python for Info Sys

Visualisation

DSfs 43-53



**Northeastern
University**

We want to see

- Currently your data is a bunch of numbers and strings which is not easily communicable.
- Data visualisation is the study of communicating data across visually.
 - Charts, tables, graphs, etc.

We will be using matplotlib

- Python library for creating visualisations.

```
pip install matplotlib
```

- Can then import it.

```
import matplotlib.pyplot as plt
```

General commands

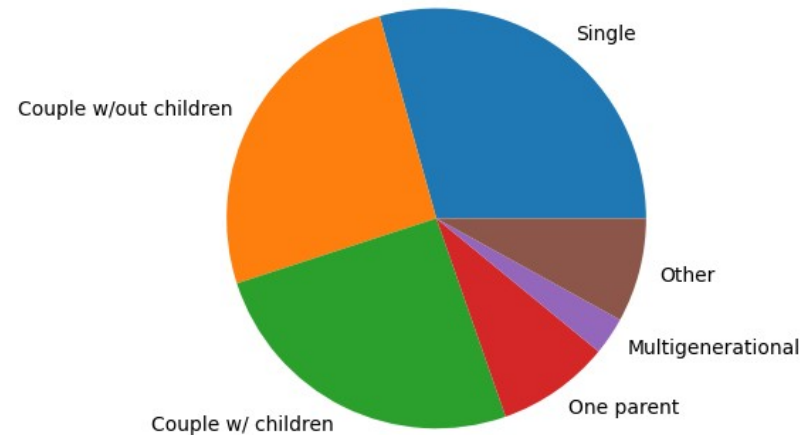
- Title: `plt.title("Title")`
- X-label: `plt.xlabel("Time")`
- Y-label: `plt.ylabel("Time")`
- Super title: `plt.suptitle("Super title")`
- Tight-layout: `plt.tight_layout()`
- Legend: `plt.legend()`

- Add grid lines: `plt.grid()`
- Set figure size: `plt.figure(figsize=(width, height))`
- X-axis range: `plt.xlim(min, max)`
- Y-axis range: `plt.ylim(min, max)`
- X-ticks: `plt.xticks()`
- Y-ticks: `plt.yticks()`
- Display: `plt.show()`
- Save to disk: `plt.savefig("filename.png")`

Pie Chart

- You give a 1D collection (x) where the proportion of each is computed as: `x / sum(x)`
- Can optional give a string list of labels.

```
plt.pie(x, labels=labels])
```

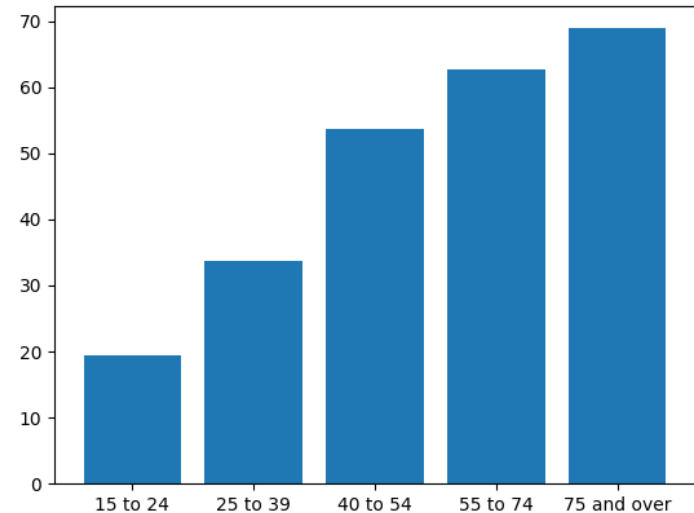


StatCan21 Household Distribution

Bar Charts

- You give the bar labels (x) and the height of each (height)
- Can optionally specify:
 - Width of each bar (width)
 - Bar alignment:
“center” or “edge”

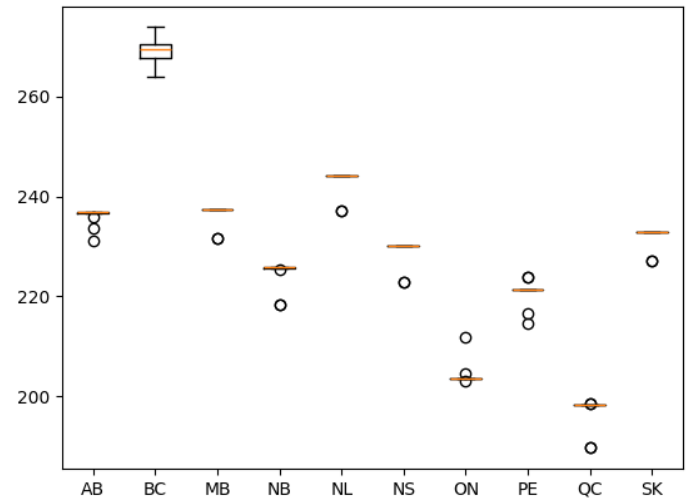
```
plt.bar(x, height,  
        width=0.8, align="center")
```



StatCan21 Toronto Household ownership rate by age.

Box Plot

- Give data as a 2D collection where each entry is a column and for each column you give all the raw data.
- Can optionally specify:
 - Labels (tick_labels)



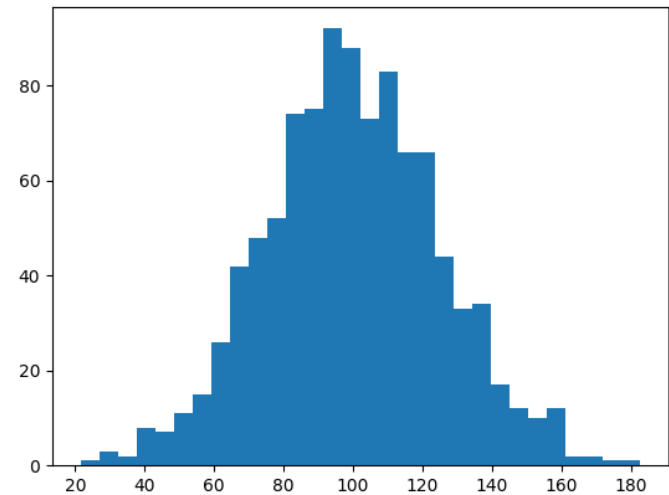
```
plt.boxplot(x, tick_labels=labels)
```

StatCan24 Average monthly egg prices per province for 2024.

Histogram

- Simply pass all of your data (x).
- Can optionally specify:
 - Number subdivisions (bins)

```
plt.hist(x, bins=10)
```

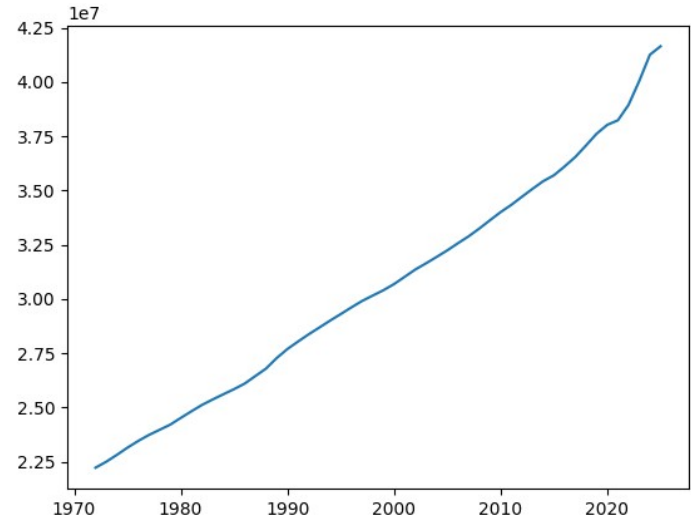


Histogram of a random distribution at mean 100 and std div of 25 with 30 subdivisions.

Line Charts

- Give your numerical x and y data.
- Can optionally pass:
 - Basic formatting
 - To scale x or y (scalex, scaley)

```
plt.plot(x, y, [fmt]  
         scalex=True, scaley=True)
```



Canada yearly population since StatCan.

[fmt]

- You first specify the colour, then the marker shape, then the line style.

r	red
g	green
b	blue
c	cyan
m	magenta
y	yellow
k	black
w	white

o	circle
*	star
.	point
,	pixel
x	x
+	plus
s	square
d	diamond

-	solid
:	dotted
--	dashed
-.	dotted-dashed

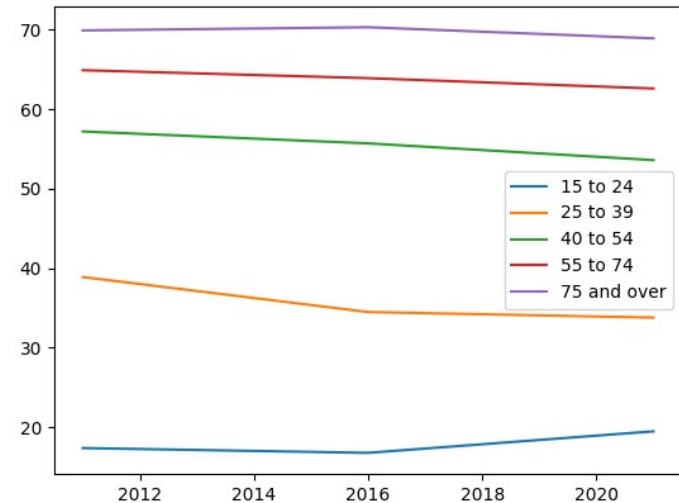
Examples:

- “bo”
- “c*--”

Multi-Line Charts

- Can call plot multiple times to place multiple lines on same chart.
- You can specify legend's label with label.

```
plt.plot(x1, y1, label="ax1")  
plt.plot(x2, y2, label="ax2")  
plt.plot(x3, y3, label="ax3")
```

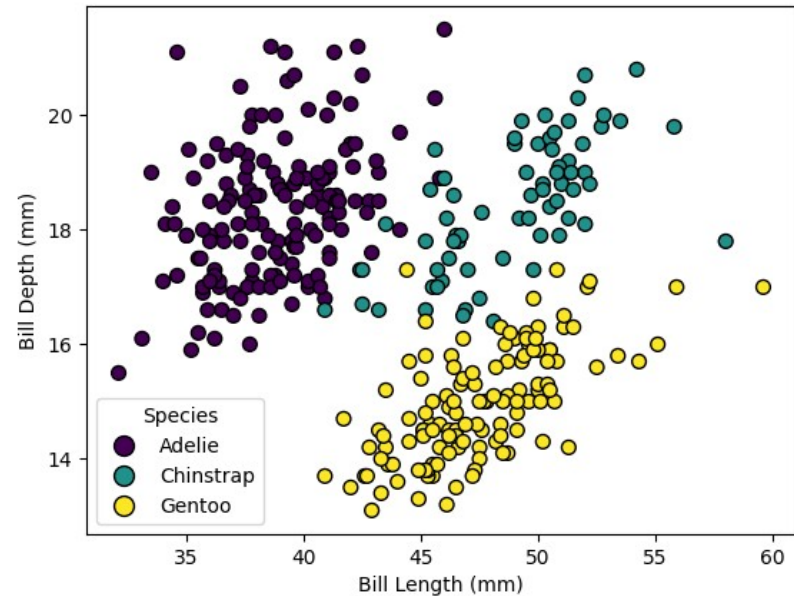


StatCan household ownership rates by age group 2011, 2016, and 2021.

Scatterplot

- Pass in the x and y.
- Can optionally specify:
 - Colour each gets (c) as a 1D collection for each entry.

```
plt.scatter(x, y, c=colours)
```



3 different peinguin samples Bill Length vs Bill Depth, Palmer Peinguis.

Use seaborn for complex viz

- Built on-top of matplotlib and allows for complex and prettier visualisations.

```
pip install seaborn
```

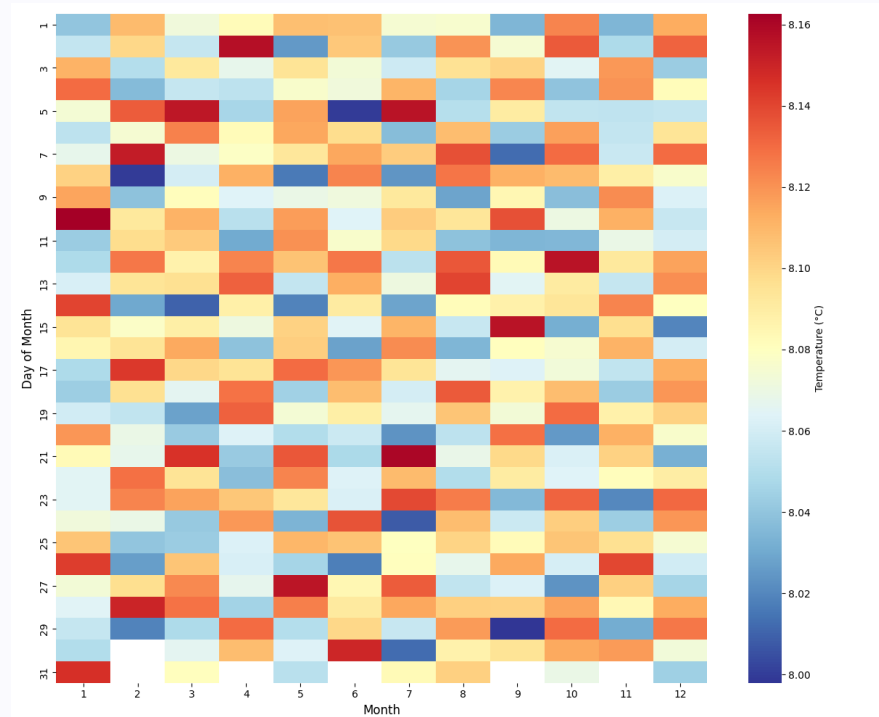
- Can then import it.

```
import seaborn as sns
```

Heatmap

- You pass a pandas DataFrame.
- Many optional arguments.

```
sns.heatmap(dataframe)
```



Average daily sea temperatures for each month sector 11417 off coast Vancouver 1.0m depth, ERDDAP

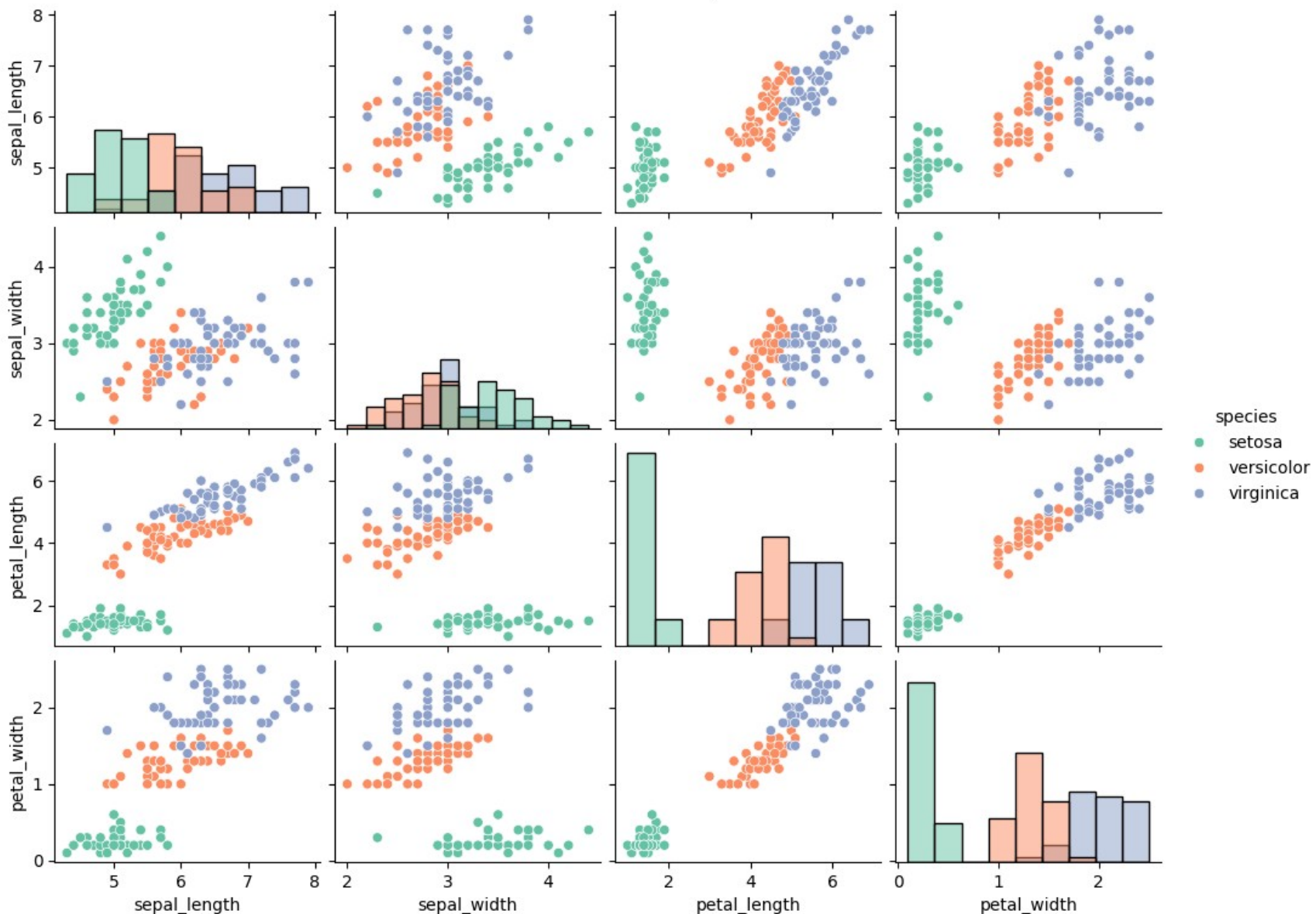
Pair Plot

- Pass in pandas DataFrame as first argument.
- Specify which variable should change colours (hue).

- Can optionally specify:

```
sns.pairplot(data,  
             hue=hue,  
             palette=palette,  
             vars=vars,  
             diag_kind="auto")
```

- palette: seaborn colour palette
- vars: limit which variables to use (otherwise all)
- diag_kind: "auto", "hist", "kde", None



Let's practice

In **Canvas** you will find a zipped file which has the energy production data of **each** of Canada's 10 provinces per month for the year 2024. Your job is to ***tell a story***.