

# Announcements

COMP1730/COMP6730  
Programming for Scientists

Data analysis and visualisation

- ★ Please fill out **mid-semester survey on Wattle** *once you finished your lab this week!* It's open until the end of semester break (17 Sept) and will help us to identify areas for improvement in the 2nd half of the course!

## Recap of 1st half and outline for 2nd half

So far:

- ★ Functional decomposition
- ★ Types and expressions
- ★ Branching, `if else`
- ★ Iteration, `while` & `for` loop
- ★ Sequence, `list`, `tuple`, `str`
- ★ Code quality
- ★ Debugging & testing
- ★ Data analysis & visualisation

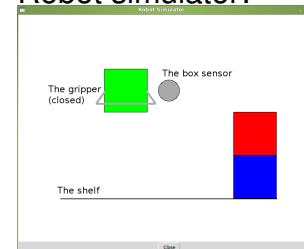
What's next?

- ★ Numpy arrays
- ★ Files, Input/Output
- ★ Dictionaries and sets
- ★ Exception handling
- ★ Complexity, big-O notation
- ★ Dynamic programming
- ★ Computational Science
- ★ Another advanced topic or 2

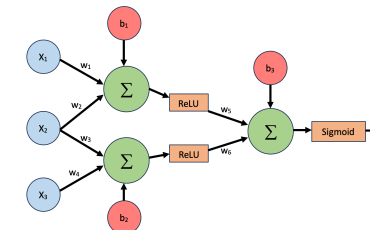
Many, if not most, concepts also apply to other programming languages, not just Python!

## Many scientific applications

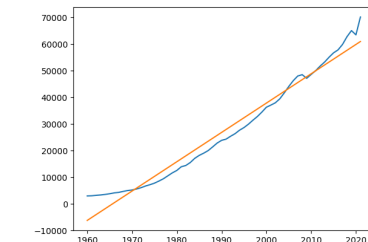
Robot simulator:



Neural network:



Linear regression:



Bioinformatics:

AGAGACCCCT	
AGA	
GAG	
AGA	
GAC	
ACC	
CCC	
CCC	
CCC	
CCT	

→ k=3:

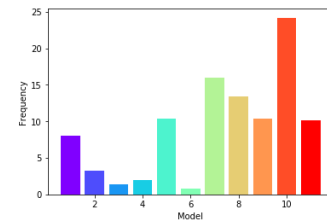
AGA	2
GAG	1
GAC	1
ACC	1
CCC	3
CCT	1

## Data science

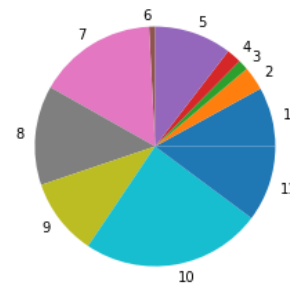
How-to:

- ★ Represent 2-dimensional data?
- ★ Read and write data?
- ★ Analyse and visualise data?
- ★ Interpret data?

Barplot:



Piechart:



## A working example

COVID-19 cases until 25th March 2022 (Source: Johns Hopkins University)

FIPS	Admin2	Province_State	Country_Region	Last_Update	Lat	Long_	Confirmed	Deaths	Recovered	Active	Combined_Key	Incident_Rate	Case_Fatality_Ratio
2			Alghanistan	26/3/22 4:20	33.93911	67.709953	177321	7657			Alghanistan	455.506183	4.31815747
3			Albania	26/3/22 4:20	41.1533	20.1683	273318	3490			Albania	9497.46394	1.2769009
4			Algeria	26/3/22 4:20	28.0339	1.6596	265612	6873			Algeria	605.714213	2.58760899
5			Andorra	26/3/22 4:20	42.5063	1.5218	39713	153			Andorra	51398.434	0.38526427
6			Angola	26/3/22 4:20	-11.2027	17.8739	99102	1900			Angola	301.531041	1.91721661
7			Antarctica	26/3/22 4:20	-71.9499	23.347	11	0			Antarctica		0
8			Antigua and Barbuda	26/3/22 4:20	17.0608	-61.7964	7482	135			Antigua and Barbuda	7640.30716	1.80433039
9			Argentina	26/3/22 4:20	-38.4161	-63.6167	9023812	127846			Argentina	19966.0513	1.41676267
10			Armenia	26/3/22 4:20	40.0691	45.0382	422423	8607			Armenia	14255.4722	2.0375311
11			Australian Capital Territory	26/3/22 4:20	-35.4735	149.0124	72571	39			Australian Capital Territory	16951.8804	0.05374047
12			New South Wales	26/3/22 4:20	-33.8688	151.2093	1715381	2055			New South Wales	21130.5864	0.11979846
13			Northern Territory	26/3/22 4:20	-12.4634	130.8456	47660	33			Northern Territory	19405.5375	0.06924045
14			Queensland	26/3/22 4:20	-27.4698	153.0251	721628	717			Queensland	14106.6953	0.09935867
15			South Australia	26/3/22 4:20	-34.9285	138.6007	227182	246			South Australia	12933.7888	0.10828323
16			Tasmania	26/3/22 4:20	-42.8821	147.3272	78805	29			Tasmania	14716.1531	0.0367997
17			Victoria	26/3/22 4:20	-37.8136	144.9631	1233174	2722			Victoria	18598.3499	0.22073122
18			Western Australia	26/3/22 4:20	-31.9505	115.8605	132060	34			Western Australia	5020.14749	0.02574587
19			Austria	26/3/22 4:20	47.5162	14.5501	3665003	15619			Austria	40693.3181	0.42616609
20			Azerbaijan	26/3/22 4:20	40.1431	47.5769	791654	9675			Azerbaijan	7807.87391	1.22212482
21			Bahamas	26/3/22 4:20	25.025885	-78.035889	33242	788			Bahamas	8453.18984	2.37049516
22			Bahrain	26/3/22 4:20	26.0275	50.55	549718	1468			Bahrain	32306.2701	0.26704601

## Data files

- ★ Many data file formats (e.g., excel, csv, json, binary). We'll use the following csv file.

```
FIPS,Admin2,Province_State,Country_Region,Last_Update,Lat,Long_,Confirmed,Deaths,Recovered,Active,Combined_Key,Incident_Rate,Case_Fatal
,,Afghanistan,2022-03-26 04:20:23,33.93911,67.709953,177321,7657,,Alghanistan,455.50618250081607,4.318157465838789
,,Albania,2022-03-26 04:20:23,41.1533,20.1683,273318,3490,,Albania,9497.463340051429,1.2769008993187423
,,Algeria,2022-03-26 04:20:23,28.0339,1.6596,265612,6873,,Algeria,605.7142130005892,2.587608993569568
,,Andorra,2022-03-26 04:20:23,42.5063,1.5218,39713,153,,Andorra,51398.43396104316,0.38526427114546874
,,Angola,2022-03-26 04:20:23,-11.2027,17.8739,99102,1900,,Angola,301.5310408836196,1.917216605113923
,,Antarctica,2022-03-26 04:20:23,-71.9499,23.346999999999998,11,0,,Antarctica,,0
,,Antigua and Barbuda,2022-03-26 04:20:23,17.0608,-61.7964,7482,135,,Antigua and Barbuda,7640.3071644473475,1.8043303929430634
,,Argentina,2022-03-26 04:20:23,-38.4161,-63.6167,9023812,127846,,Argentina,19966.05125297436,1.4167626719173672
,,Armenia,2022-03-26 04:20:23,40.0691,45.0382,422423,8607,,Armenia,14255.472238677698,2.0375311003425476
,,Australian Capital Territory,Australia,2022-03-26 04:20:23,-35.4735,149.0124,72571,39,,Australian Capital Territory, Australia",16951.8804005374047
,,New South Wales,Australia,2022-03-26 04:20:23,-33.8688,151.2093,1715381,2055,,New South Wales, Australia",21130.58635131806,0.11979846
,,Northern Territory,Australia,2022-03-26 04:20:23,-12.4634,130.8456,47660,33,,Northern Territory, Australia",19405.53745928339,0.06924045
,,Queensland,Australia,2022-03-26 04:20:23,-27.4698,153.0251,721628,717,,Queensland, Australia",14106.6953769915,0.09935867233533067
,,South Australia,Australia,2022-03-26 04:20:23,-34.9285,138.6007,227182,246,,South Australia, Australia",12933.788784514658,0.10828323
,,Tasmania,Australia,2022-03-26 04:20:23,-42.8821,147.3272,78805,29,,Tasmania, Australia",14716.153127917834,0.0367996545079627
,,Victoria,Australia,2022-03-26 04:20:23,-37.8136,144.9631,1233174,2722,,Victoria, Australia",18598.349899696823,0.2207312187898869
,,Western Australia,Australia,2022-03-26 04:20:23,-31.9505,115.8605,132060,34,,Western Australia, Australia",5020.147494868091,0.02574587
,,Austria,2022-03-26 04:20:23,47.5162,14.5501,3665003,15619,,Austria,40693.3180849174,0.4261660904586763
,,Azerbaijan,2022-03-26 04:20:23,40.1431,47.5769,791654,9675,,Azerbaijan,7807.873914790897,1.2221248171549692
,,Bahamas,2022-03-26 04:20:23,25.025885,-78.035889,33242,788,,Bahamas,8453.189844576451,2.370495156729439
,,Bahrain,2022-03-26 04:20:23,26.0275,50.55,549718,1468,,Bahrain,32306.270102604463,0.267046026828665
,,Bangladesh,2022-03-26 04:20:23,23.685,90.3563,1951174,29118,,Bangladesh,1184.7600400567412,1.4923323086510993
,,Barbados,2022-03-26 04:20:23,13.1939,-59.5432,58270,330,,Barbados,20276.92425470907,0.5663291573708598
,,Belarus,2022-03-26 04:20:23,53.7898,27.9534,957888,6767,,Belarus,10128.643105679232,0.7070405229195226
,,Antwerp,Belgium,2022-03-26 04:20:23,51.2195,4.4024,592524,0,,Antwerp, Belgium",31896.660101852216,0.0
,,Brussels,Belgium,2022-03-26 04:20:23,50.8503,4.3517,424772,0,,Brussels, Belgium",35147.475222209905,0.0
```

Which data type can we use to represent tables?

## Representing tables

- ★ Lists are 1-dimensional, but a list can contain values of any type, including lists.
- ★ A table can be stored as a list of lists, by row, for example:

```
data[i] # i:th row
data[i][j] # j:th column of i:th row
```

- ★ Indexing (and slicing) are operators
- ★ Indexing (and slicing) associate to the left:

```
data[i][j] == (data[i])[j]
```

## Reading data files

- \* Use a python module that helps with reading the file format:

---

```
import csv
with open("filename.csv") as csvfile:
    reader = csv.reader(csvfile)
    next(reader) # skip the header
    data = [ row for row in reader ] # reader is an iterable
```

---

- \* More about (reading and writing) files later in the course.

## How to select a column of the table?

- \* List comprehension:

---

```
first_col = [ row[0] for row in data ]
last_two_cols = [ row[-2:] for row in data ]
```

---

- \* Equivalent to:

---

```
first_col = []
for row in data:
    first_col.append(row[0])
```

---

## Select rows satisfying some conditions?

- \* Syntax:

---

```
[ expression for item in iterable if condition ]
```

---

- \* Example: select rows where column-1 is  $> 10$

---

```
sel_rows = [ row for row in data if int(row[1]) > 10 ]
```

---

- \* Equivalent to:

---

```
sel_rows = []
for row in data:
    if int(row[1]) > 10:
        sel_rows.append(row)
```

---

## How to sort rows by some keys?

- \* `sorted(seq)` returns a list with values in `seq` sorted in default order (`<`).
  - We can sort the rows in a table.
  - Reminder: comparison of sequences is lexicographic.
- \* `sorted(seq, key=fun)` sorts value `x` by `fun(x)`.

---

```
def new_order(row):
    return -row[-1] # decreasing on last col
```

---

```
sd = sorted(data, key=new_order)
```

---

## Descriptive statistics

- \* `min(seq)`;
- \* `max(seq)`;
- \* `mean(sum(seq) / len(seq))`;
- \* `variance`.
- \* No built-in function for median.

---

```
def median(seq):  
    if len(seq) % 2 == 1:  
        return sorted(seq)[len(seq) // 2]  
    else:  
        return sum(sorted(seq)[(len(seq)//2-1):(len(seq)//2+1)])/2
```

---

## Visualisation

- \* The purpose of visualisation is to see or show information – pretty pictures are only of secondary importance!
- \* Different kinds of plots show different things:
  - barplot
  - pie-chart
  - histogram or cumulative distribution
  - scatterplot
  - line and area plot
- \* Use one that best makes the point!
- \* Choose your dimensions carefully.
- \* Label axes, lines, etc.

## Matplotlib

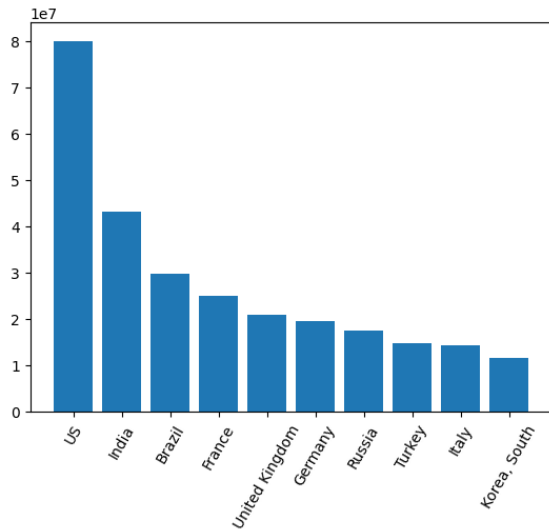
- \* Matplotlib is a Python 2D plotting library, which produces publication quality figures.
- \* “*Matplotlib makes easy things easy and hard things possible*”.
- \* Documentation: [matplotlib.org](https://matplotlib.org)

Programming problem:

- \* How many COVID-19 positive cases worldwide until 2022-03-25?
- \* How many COVID-19 deaths worldwide until 2022-03-25?
- \* What are the top-10 countries with the most cases until 2022-03-25?
- \* How to visualise this result?

(added after lecture)

The code was live demo in the lecture. And visualisation with barplot:



## Take home message

- \* Python is powerful in data analysis.
- \* Think carefully about visualisation: How can people quickly interpret the results?
- \* We have only scratched the surface of Matplotlib. Extensive documentation: <https://matplotlib.org> or just **google it!**