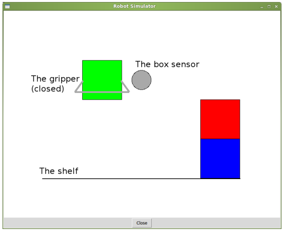
Australian National University

COMP1730/COMP6730
Programming for Scientists

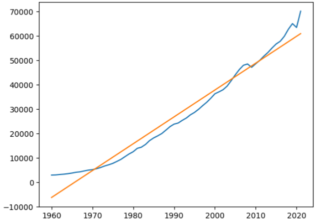
Data analysis and visualisation

Many scientific applications

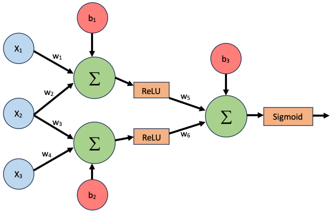
Robot simulator:



Linear regression:



Neural network:



Bioinformatics:

```
AGAGACCCCCT
AGA
 GAG
  AGA
   GAC
    ACC
     CCC
      CCC
       CCC
        CCT
```

$\rightarrow$

k=3:

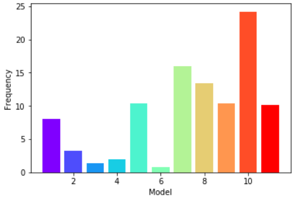
AGA	2
GAG	1
GAC	1
ACC	1
CCC	3
CCT	1

Data science

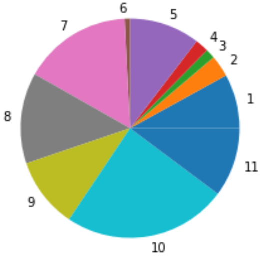
How-to:

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

Barplot:



Piechart:



Data file

- Many data file formats (e.g., excel, csv, tsv, json, binary). We'll use the following csv (comma-separated value) file: iris.csv.
- The Iris flower data set or Fisher's Iris data set is a multivariate data set used and made famous by the British statistician and biologist Ronald Fisher in 1936.
- It is representative of a typical experimental dataset, with individual sampling units (flowers) along the rows, with several measured variables along the columns.
- The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

Representing 2D tables

* Lists are 1-dimensional, but a list can contain values of any type, including lists and so, as we have seen, a 2D table could be stored as a list of lists.
* In this lecture we will describe the pandas module, which is designed for such data science applications.
* The pandas module builds on numpy arrays to implement the 2D DataFrame data structure (which is based on the R syntax).
* For further details see:
* `https://pandas.pydata.org/docs/user_guide/10min.html#min` or `https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf`
* To get started with pandas, you will need to get comfortable with its two data structures: Series and DataFrame.

Australian National University

Series

- A Series is a one-dimensional vector capable of holding any data type (integers, strings, floating point numbers, Python objects, etc.) and an associated array of data labels, called its index. If no index is passed, numbers will be assigned as labels automatically starting from 0.
- A Series (given this name as it often represents a time series or other series of measured values) is built on the Numpy vector class and can be indexed similarly. It mainly just adds the data labels which are useful in data analysis.

```
from pandas import Series, DataFrame

s =Series([4,7,-5,3])
s
s.values   # extract underlying Numpy vector
s = Series(range(5), index=list('abcde'))
s
s.mean()
s+s    # acts in a vectorised fashion, like the underlying  Numpy vecto
```

DataFrame

- A DataFrame represents a tabular 2 dimensional spreadsheet-like data structure containing an ordered collection of columns, each of which can be a different value type.
- Allowing columns to have different types, as is needed for e.g. measured variables on an experimental sample, is the main difference with Numpy arrays which have a single (homogeneous) type.
- Consider the following example of analysis comparing 2 genes in 3 different experiments:

```
df = DataFrame([[20,20],[21,30],[19,40]],columns = ['gene_experimental',
print(df)
print(df.dtypes)  # In a DataFrame columns can have different types
```

Reading data files

* Pandas makes it easy to read in csv and other data files.

```
df2 = pd.read_csv('iris.csv', header=0)
print(df2.dtypes)
print(df2.columns)
print(df2.head())
```

How to select a row and column of the table?

```
df[0:1]  # select first row
df["Species"] # select Species column (returns a Series)
df.iloc[0:2, 0:1]  # slicing like list, using indexing by position
df.loc[2,"Sepal.Length"] # indexing by row and column name
```

Select rows satisfying some conditions?

- Example: get rows where Sepal.Width $>$ 3.0
- To do this, we can index the DataFrame with a boolean array.

```
df[df["Sepal.Width"] > 3.0]
```

- Example: Select only rows of iris data where Species is recorded

```
df2[df2["Species"].notna()]
```

Descriptive statistics

* min
* max
* mean
* variance.

```
df["Sepal.Length"].mean()
df["Sepal.Length"].var()    # variance
df["Sepal.Length"].max()
df["Sepal.Length"].median()
```

Visualisation

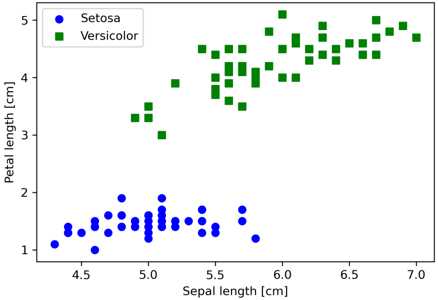
- The purpose of visualisation is to see or show information – pretty pictures are only of secondary importance!
- Different kinds of plots show different things:
  - barplot
  - pie-chart
  - histogram or cumulative distribution
  - scatterplot
  - line and area plot
- Use one that best makes the point.
- 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`

Programming problem:

* Show scatter plot of relationship between petal and sepal length.
* Select only setosa and versicolor species.
* Show different species in different colours.
* How to visualise this result?


Setosa
Versicolor
Petal length [cm]
Sepal length [cm]
1
2
3
4
5
4.5
5.0
5.5
6.0
6.5
7.0

Take home message

- Python with pandas, numpy and Matplotlib is powerful in data analysis.
- Think carefully about visualisation: How can people quickly interpret the results?
- We have only scratched the surface of Matplotlib.
- For more examples of using Matplotlib with pandas: `https://pandas.pydata.org/docs/getting_started/intro_tutorials/04_plotting.html`
- Extensive documentation: `https://matplotlib.org` or just **google it**