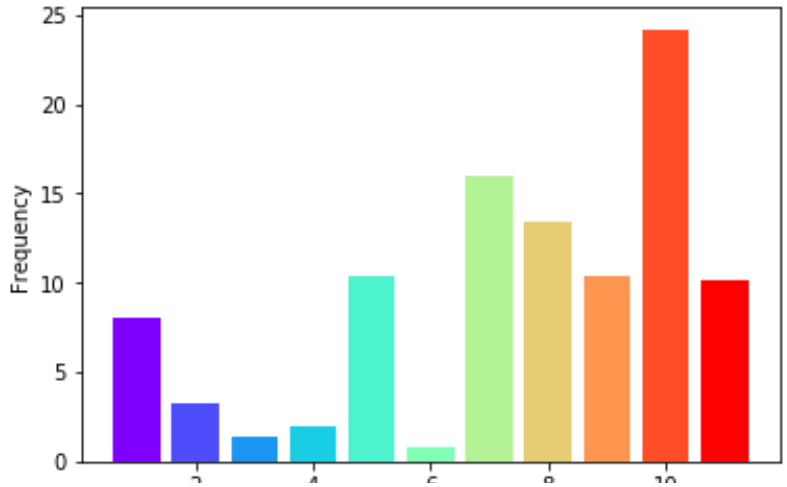
COMP1730/COMP6730
Programming for Scientists

Data science

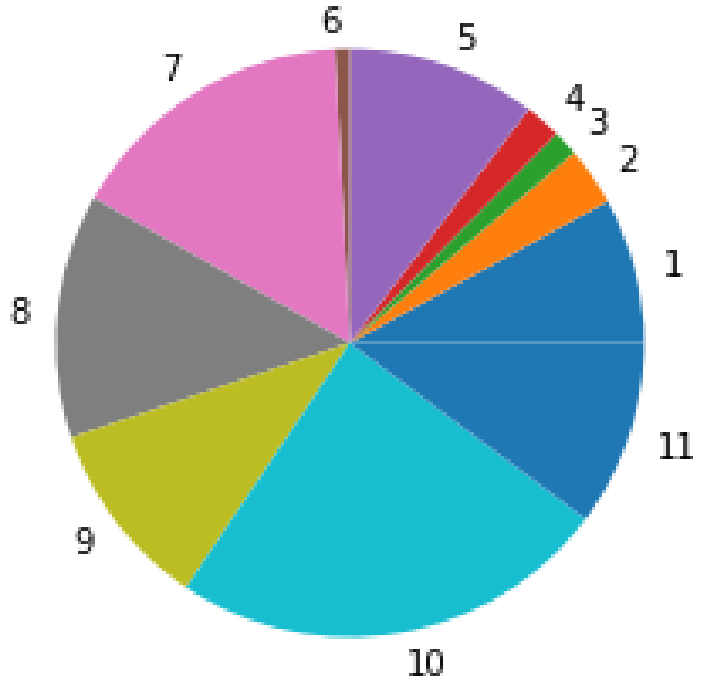
Announcements

* Example questions for mid-semester exam are available on the course web site.
* Example solutions for selected lab exercises also available online.
* Homework 3 will be checked in this week lab.
* Homework 4 is now available and due on Monday of week 8 (instead of week 7).
* There will be a lecture tomorrow on **Debugging**.

Data analysis



* Reading data files
* Representing tables
* Working with data: selecting, visualising
* Interpretation



Working example

Table shows how often each model fits best to each test data set. We want to answer: Which model is the best?

Question: Which Python data type can we use to process tables?

Model	test1	test2	test3	test4	test5	test6	test7	test8
1	40	571	353	9	95	41	1428	350
2	16	200	108	2	495	434	88	0
3	7	352	216	9	1201	1897	9	0
4	10	187	202	280	704	215	47	0
5	52	616	204	2	47	17	122	5
6	4	147	146	0	3646	536	0	0
7	80	914	373	4	45	2	161	60
8	67	406	778	1	9	2	3	30
9	52	635	303	1	5	0	5	860
10	121	712	595	0	19	0	1	53
11	51	1914	449	0	29	18	4	50

NumPy Arrays

* `numpy.ndarray` is sequence type, and can also represent *n*-dimensional arrays (tables).
* fast math operations on arrays/matrices;
* plotting (via `matplotlib`).
* All values in an array must be of the same type.
* Element-wise operators, functions on arrays.
* Read/write functions for some file formats.

Data files

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

```
Model,test1,test2,test3,test4,test5,test6,test7,test8
1,40,571,353,9,95,41,1428,350
2,16,200,108,2,495,434,88,0
3,7,352,216,9,1201,1897,9,0
4,10,187,202,280,704,215,47,0
5,52,616,204,2,47,17,122,5
6,4,147,146,0,3646,536,0,0
7,80,914,373,4,45,2,161,60
8,67,406,778,1,9,2,3,30
9,52,635,303,1,5,0,5,860
10,121,712,595,0,19,0,1,53
11,51,1914,449,0,29,18,4,50
```

Reading data files with NumPy

```
import numpy as np
data = np.genfromtxt("data.csv",
       dtype=int, delimiter=",",
       skip_header=1)
```

* **NOTE**: NumPy's `int` is more limited (64-bit) than Python's `int`
* More about (reading and writing) files later in the course.

Array operations

* A table is stored as a 2-dimensional array:

```
data[i]    # row i
data[i][j] # row i, column j
data[i,j]  # row i, column j
```

* Indexing an n-d array returns an $(n-1)$-d array.
* `data.ndim` returns number of dimensions.
* `data.shape` returns the dimensions of the array as a tuple.
* `data.dtype` returns the type of elements.

Slicing

```
data[i,:]       # row i
data[:,k]       # column k
data[i:j]       # rows i,i+1,...,j-1
data[:,k:l]     # columns k,k+1,...,l-1
data[i:j,k:l]   # rows i,...,j-1 and
                # columns k,...,l-1
```

* NOTE: Slicing returns a **view** (reference) of data. What happens with?

```
b = data[1:4,3:5]
b[0,0] = -100
b[:]   = -100
```

* To copy the data: `b = data.copy()`

Descriptive statistics

* `np.min(`*`data`*`)` or `data.min()`
* `np.max(`*`data`*`)` or `data.max()`
* `np.mean(`*`data`*`)` or `data.mean()`
* `np.median(`*`data`*`)`
* Column-wise statistics: `np.min(`*`data`*`,0)` or `data.min(0)`
* Row-wise statistics: `np.min(`*`data`*`,1)` or `data.min(1)`

Visualisation

* The purpose of visualisation is to see or show information – not drawing pretty pictures!
* 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`

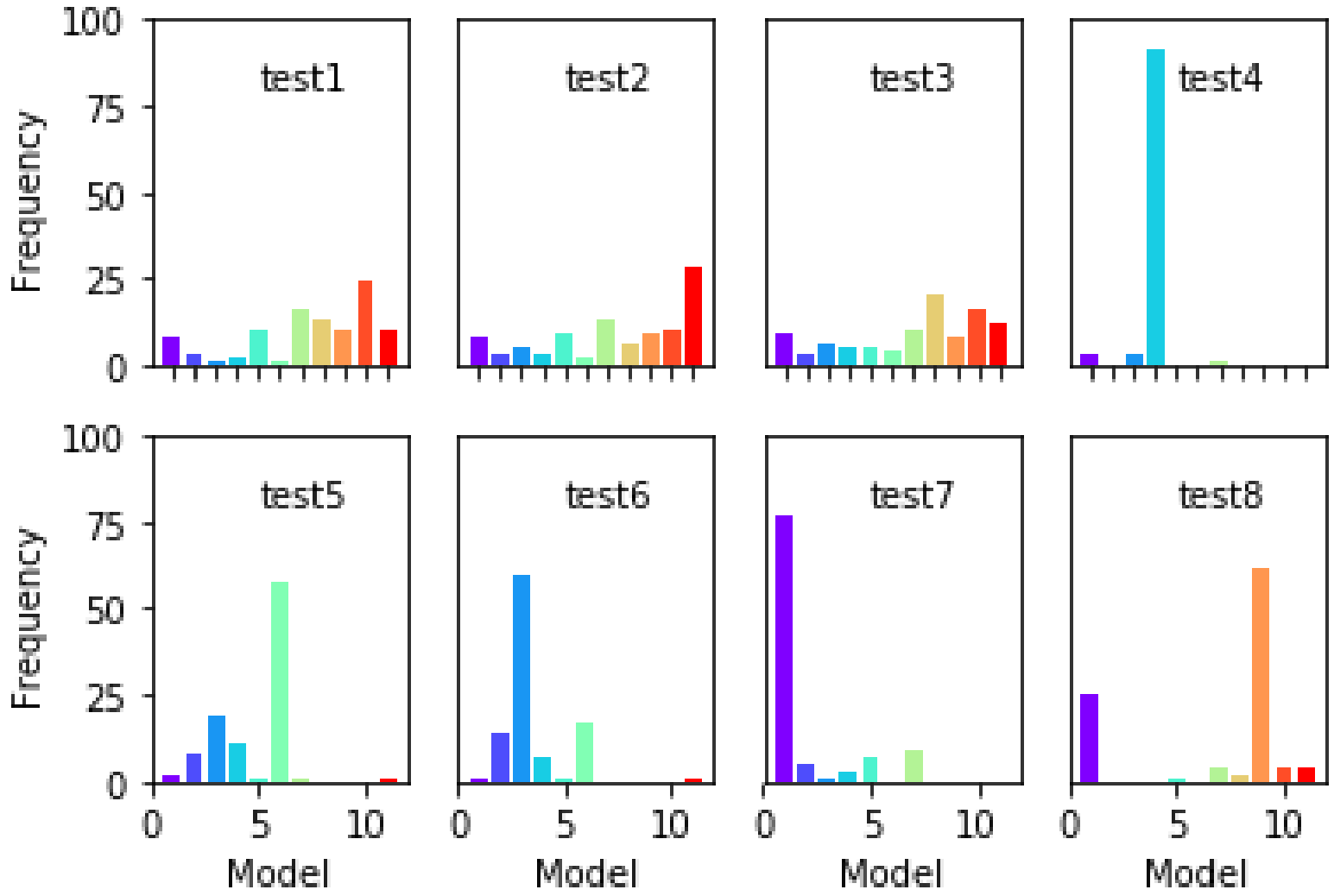
Using `matplotlib`

```
import matplotlib.pyplot as plot

plot.bar(data[:,0],data[:,1])
plot.xlabel("Model")
plot.ylabel("Best frequency")
plot.show()

plot.pie(data[:,1], labels=data[:,0],
autopct='%1.1f%%')
plot.show()
```

* Documentation: `matplotlib.org`

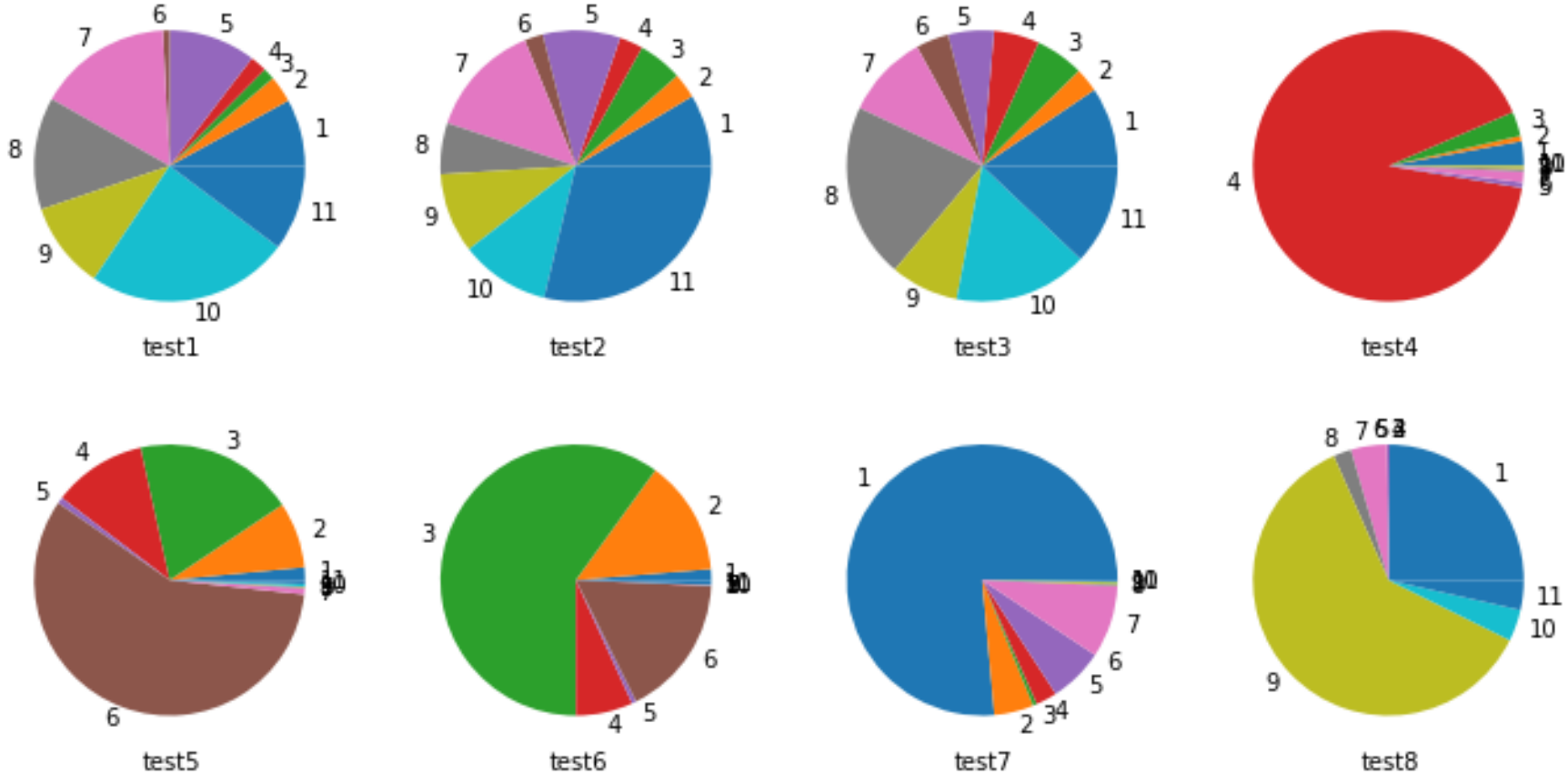
Interpretation

What is this telling us?



Interpretation

What is this telling us?



Takehome 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 NumPy and Matplotlib. Extensive documentation: `https://www.numpy.org` and `https://matplotlib.org`.
* Just **google it**!