wbs
WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

For the Change Makers

Programming for Data Analytics

Week 5: Data Processing
Information Systems and Management
Warwick Business School

Scatter

pyplot.scatter(x, y, *marker, c, s, alpha, data*)

- marker, the shape of marker
- c, the color of the marker
- s, the size of the marker,
- alpha, the transparency of the marker
- data, dataset if x, y are column names

```
plt.scatter(x, y, s=100, c='red', marker='>', alpha=
0.5)
```

Bar

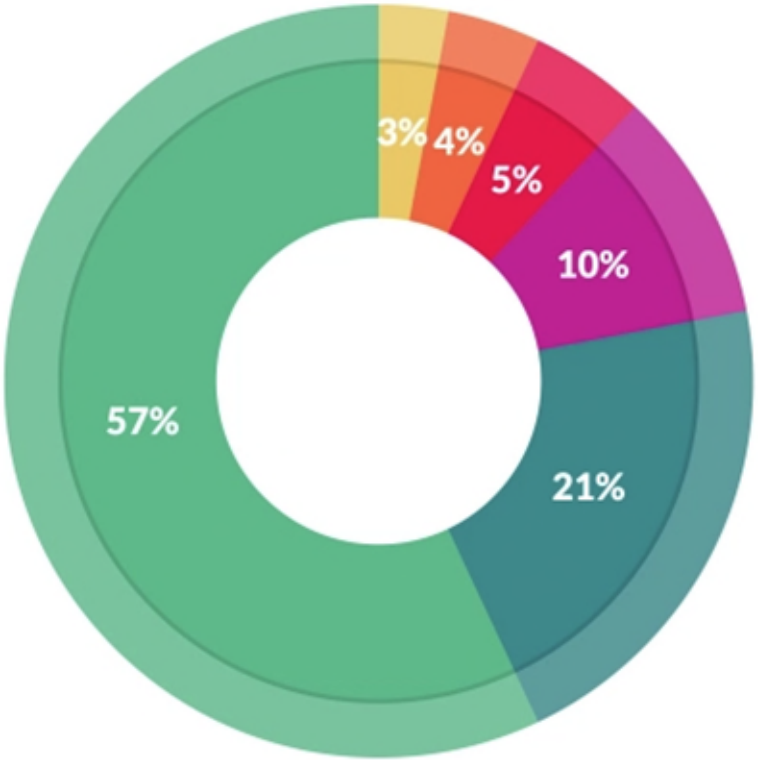
pyplot.bar(x, *height, width, data*)

- height, the height of the bar
- width, the width of the bar
- data, dataset if x, y are column names

Summary

- Matplotlib is designed based on OOP, and provides "command" functions through pyplot APIs.
- pyplot functions are easier to use and suit for most single plotting cases.
- Knowing OOP interface helps to understand the underlying operations done when calling pyplot functions.
- It also helps to tweak visualizations created by other libraries making use of Matplotlib.

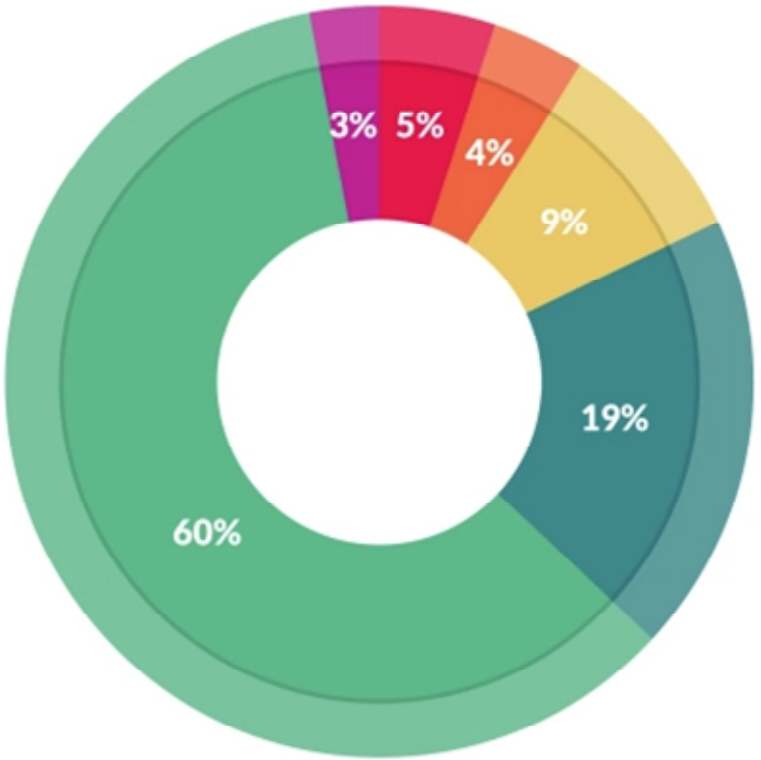
What's the least enjoyable part?



What's the least enjoyable part of data science?

- *Building training sets: 10%*
- *Cleaning and organizing data: 57%*
- *Collecting data sets: 21%*
- *Mining data for patterns: 3%*
- *Refining algorithms: 4%*
- *Other: 5%*

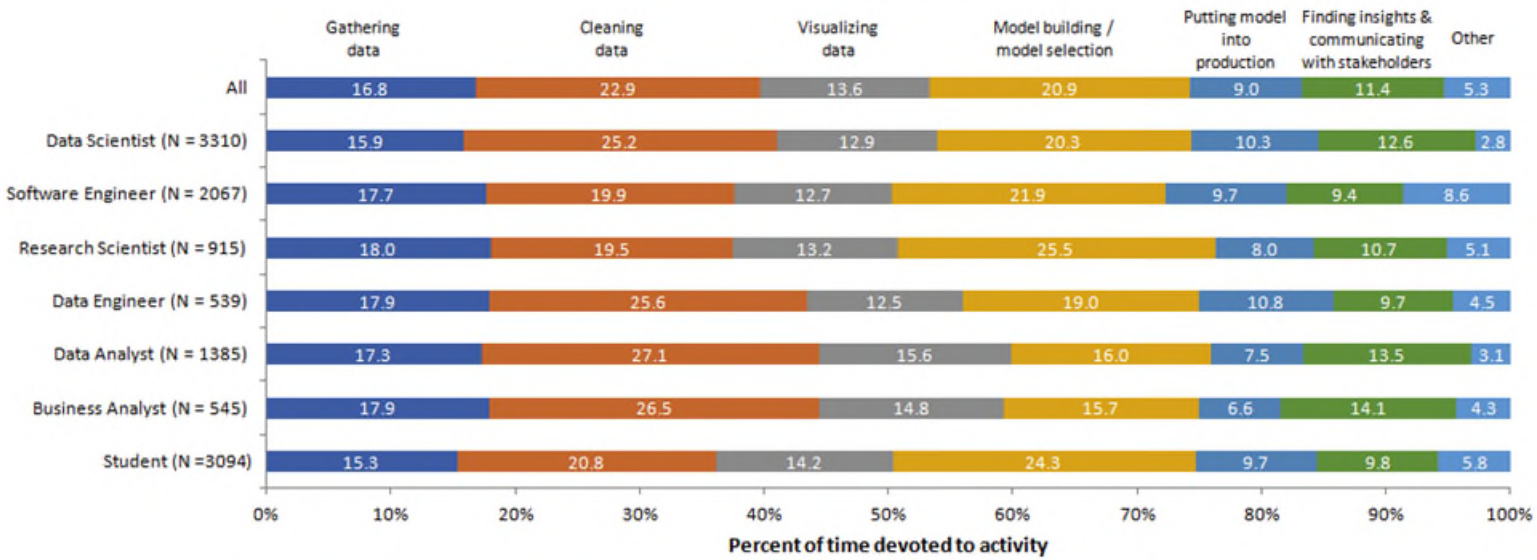
What data scientists spend the most time doing



What data scientists spend the most time doing

- *Building training sets: 3%*
- *Cleaning and organizing data: 60%*
- *Collecting data sets; 19%*
- *Mining data for patterns: 9%*
- *Refining algorithms: 4%*
- *Other: 5%*

During a typical data science project at work or school, approximately what proportion of your time is devoted to the following?



Note: Data are from the 2018 Kaggle ML and Data Science Survey. You can learn more about the study here: http://www.kaggle.com/kaggle/kaggle-survey-2018.
A total of 23859 respondents completed the survey; the percentages in the graph are based on a total of 15937 respondents who provided an answer to this question. Only selected job titles are presented.

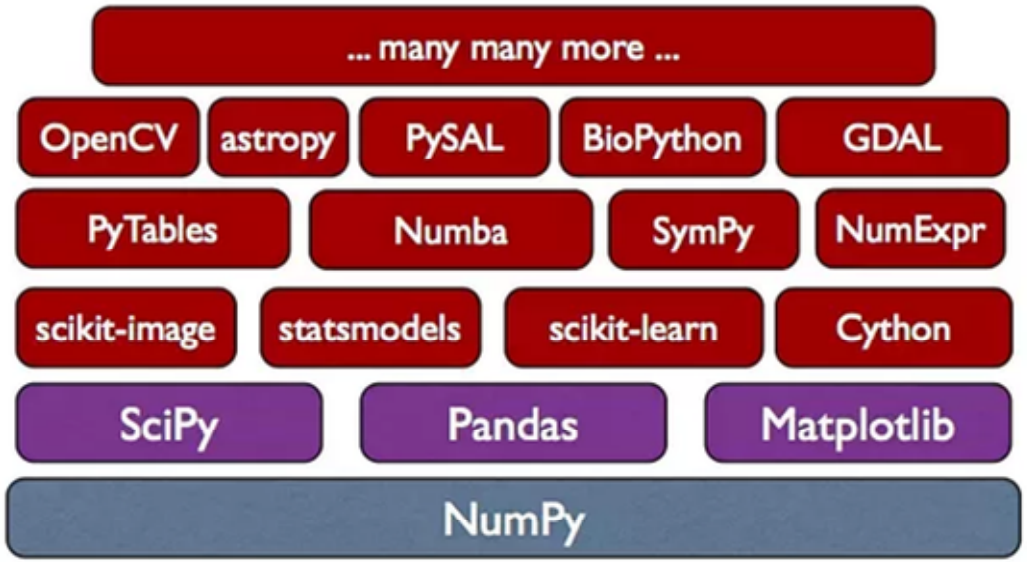
BUSINESS BROADWAY
DATA SCIENCE | CUSTOMER ANALYTICS | MACHINE LEARNING

Data processing

- Data cleaning, data wrangling, data preparation, data pre-processing, etc.
- Why?
- Data quality.
- Better performance.

NumPy

- NumPy stands for numerical Python.
- Foundation library for scientific computing in Python



https://www.quora.com/What-is-the-relationship-among-NumPy-SciPy-Pandas-and-Scikit-learn-and-when-should-I-use-each-one-of-them

Why Numpy?

- ndarray: A **multidimensional array** much faster and more efficient than those provided by the basic package of Python.
- Element-wise computation: A set of functions for performing this type of **calculation with arrays** and mathematical operations between arrays.
- Reading-writing datasets: A set of tools for reading and writing data stored in the hard disk.
- Integration with other languages such as C, C++, and FORTRAN: A set of tools to integrate code developed with these programming languages.

Installation

- NumPy should be installed as part of the Anaconda installation. If not or in a new environment:
- conda install numpy
- pip install numpy

Quick recall of Python data types

- Number
- String
- List
- Tuple
- Dictionary

NumPy array

- The NumPy library is based on one main object: **ndarray** (which stands for N-dimensional array).
- A new ndarray can be created by the array() function.

```
>>> a = np.array([1, 2, 3])
>>> a
array([1, 2, 3])
>>> type(a)
<type 'numpy.ndarray'>
```

Characteristics of ndarray

- ndarray is **homogeneous**: consisted of data in the same date type.
- ndarray is **size-fixed**: once created, the size of a ndarray cannot be changed.

NumPy data type dtype

bool_	Boolean (True or False) stored as a byte
int_	Default integer type (same as C long; normally either int64 or int32)
intc	Identical to C int (normally int32 or int64)
intp	Integer used for indexing (same as C ssize_t; normally either int32 or int64)
int8	Byte (-128 to 127)
int16	Integer (-32768 to 32767)
int32	Integer (-2147483648 to 2147483647)
int64	Integer (-9223372036854775808 to 9223372036854775807)
uint8	Unsigned integer (0 to 255)
uint16	Unsigned integer (0 to 65535)
uint32	Unsigned integer (0 to 4294967295)
uint64	Unsigned integer (0 to 18446744073709551615)
float_	Shorthand for float64.
float16	Half precision float: sign bit, 5 bits exponent, 10 bits mantissa
float32	Single precision float: sign bit, 8 bits exponent, 23 bits mantissa
float64	Double precision float: sign bit, 11 bits exponent, 52 bits mantissa
complex_	Shorthand for complex128.
complex64	Complex number, represented by two 32-bit floats
complex128	Complex number, represented by two 64-bit floats

Python	Numpy
int	int_
bool	bool_
float	float_
complex	cfloat
bytes	bytes_
str	unicode_

```
>>> a.dtype
dtype('int64')
```

Homogeneous

If you mix some strings with the numbers then all of the elements will get converted into a string type and we won't be able to perform most of the numpy operations on that array.

```
>>> a.dtype
dtype('int64')
>>>b = np.array(['a', 2, 3])
dtype('<U1')
```

Creating a Numpy Arrary

- A numpy array can be created in three ways:

1. From a Python list or tuple:

```
>>> larray =
np.array([1,2,3])
>>> tarray =
np.array((1,2,3))
>>> larray
array([1, 2, 3])
>>>tarray
array([1, 2, 3])
```

```
>>> marray = np.array([[1,2,3],
                       [2,3,4],
                       [3,4,5]])
>>>marray
array([[1, 2, 3],
       [2, 3, 4],
       [3, 4, 5]])
```

2. Using numpy functions:

- .arange(start, stop, step), similar to range().

```
>>>narray = np.arange(1,10,2)
>>>narray
array([1, 3, 5, 7, 9])
```

- .linspace(start, stop, num) creates arrays with a specified number of elements, and spaced equally between the start and stop.

```
>>>narray = np.linspace(1,10,5)
>>>narray
array([ 1.  ,  3.25,  5.5 ,  7.75, 10.  ])
```

`.zeros()` creates an array full of zeros,

`.ones()` creates an array full of ones,

`.empty()` creates an array whose initial content is random and depends on the state of the memory.

- You only need to specify shape of array as a list or tuple to be generated.

np.zeros([2,3])

- You can provide data type, dtype, as arguments to these functions.

Example

- Create a 3 by 2 matrix with all zeros with data type int16.

>>>zarray = np.zeros((3,2),dtype=np.int16)

Create a 2 by 3 by 4 array with all ones with data type float64.

>>>oarray = np.ones((2,3,4),dtype=np.float64)

Create an empty 3 by 3 array with data type

>>>earray = np.empty((3,3))

3. Reading data files.

import data in text (csv) file into numpy array with loadtxt() and genfromtxt().

- loadtxt() is a fast reader for simply formatted files with numeric values only.
- genfromtxt() provides more sophisticated handling of, e.g., lines with missing values, mixed data type.

```
>>>farray = np.getfromtxt('data.csv')
>>>farray = np.loadtxt('data.csv')
```

loadtxt()

- Required argument: file path
- Key optional arguments:
  - **delimiter**: define how to split each row
  - skiprows: the number of rows to skip, int, default is 0.
  - max_rows: the maximum number of rows to import, int, default is all.
  - usecols: the columns to import (starting with 0 for first column), int or a tuple of integars, default is all.
  - encoding: Encoding used to decode the inputfile, str, default is bytes.

```
>>>tips = np.loadtxt('tips.csv',skiprows=1,usecols=(0,1,6),max_rows=244,delimiter=',')
```

genfromtxt()

- Required argument: file path
- Key optional arguments:
  - **delimiter:** define how to split each row
  - **names:** True, or a command separated strings or a list, default is None.
  - dtype: non-numbers will be treated as nan. Set to None to detect automatically.
  - skip_header: the number of rows to skip, int, default is 0.
  - max_rows: the maximum number of rows to import, int, default is all.
  - missing_values: the values, such as "N/A" or "???", should be marked as missing.
  - filling_values: values, such as 0, should be filled for missing values.
  - usecols: the columns to import (starting with 0 for first column), int or a tuple of integars, default is all.
  - encoding: Encoding used to decode the inputfile, str, default is bytes.

```
Tips = np.genfromtxt('tips.csv', delimiter=',',usecols=[0,1], max_rows=100, names=True)
```

Loading and Saving Data in Binary Files

- Numpy ndarray dataset can also be saved to a binary file and then load. Similar as you save into Excel .xlsx file.
- To save your data into .npy file, use .save(file_name, data)

>>>np.save('np_data', data) #.npy extension will be added automatically.

- To load your .npy file into ndarray, use load(file_name)

>>> data_load = np.load('np_data.npy')#.npy extension should be manually added.

Exercise

- Can you load the all_game.csv file as a ndarray and then save it to .npy file.