Python Data Analytics Day 1

Environment & Setup

- Anaconda
- Jupyter notebooks
- Command line

```
python --version
pip --version
pip list
conda --version
```

Python syntax overview

Data types

- bool
- string
- int
- float
- list
- tuple
- dict
- Variables are objects with methods.
    - For example, for strings: `.format()` `.split()`, .. etc
    - For lists, `.append()`, `.insert()`, `.pop()` ... etc.

Slicing

Loops

- while
- for .. in (strings are also iterable!)
- List comprehension (compose lists, repeat commands)

if .. elif .. else

import .. as, from ... import .. as

- Example: `numpy`, `np.random.randint()`, `.rand()`
- Check: what's the object type returned by `rand()`? Is it the same as a list?
- Calculations with lists vs. np arrays

LAB 1:

Write a dice game, where we throw 2 dice (i.e. 2 random numbers from 1 to 6).

- Display the dice readings + sum. For example *Dice: [4, 5]. Sum: 9*
- If same number on both dice: display *BINGO!*
- Else if sum > 8, display *WINNER!*
- Otherwise, display *LOOSER!*

Functions

- Specifying arguments and defaults
- Returning values
- lambda functions

LAB 2:

Write a `throwDice(numDice=1, numThrows=1)` function.

- It should return a 2D array indicating the results.
- For example: `throwDice(2, 3)` should return something like `[[1, 4], [3, 5], [5, 6]]` (reading of the 2 dice when throwed 3 times)
- Defaults call `throwDice()` should return something like `[[3]]`

If we `throwDice(2, 1000)` (2 dice for 1000 times), print out the frequency of the sum of each throw as a `dict`. For example: { 2: 50, 3: 44, 4: 37, ... }

Does the above result comply with probability of sums of 2 dice?

Matplotlib

Hello charts!

`plt.hist()`

Example problem:

Create a histogram containing 2 data series:

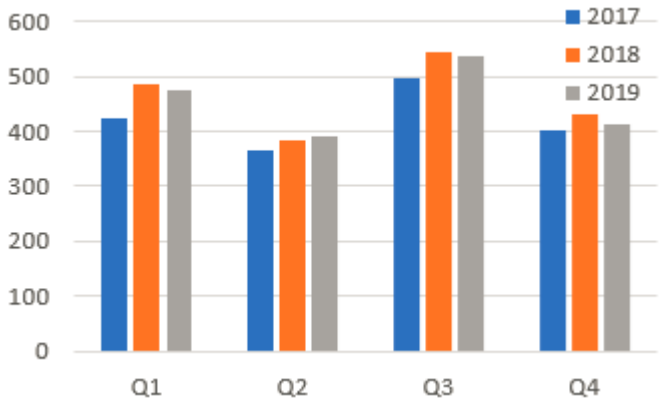
- 1 million normally distributed random values.
- 1 million uniformly distributed random values, with range from -4 to 4
- Add a legend to the chart to show which color is normally distributed data and which is uniformly distributed data.

LAB 3:

Given the following quarterly sales data of Widgets Corp.:

```
      Q1    Q2    Q3    Q4
2018  425   366   498   403
2019  486   385   546   432
2020  474   390   538   412
```

- Plot a bar chart, with quarters on the X-axis and sales on the Y-axis, and include a legend, like this:



- Plot a ***stacked barchart*** with years on the horizontal axis, and sales on the vertical axis. Each year's bar should be a stack of Q1, Q2, Q3 and Q4 sales, with a legend indicating quarters.
- TIP: refer to matplotlib gallery for help! https://matplotlib.org/gallery.html

Subplots

LAB 4

For the same data from LAB 3 above, use subplots to display a *dashboard* of 2x2 tiles containing the following:

- `tile[0,0]`: line chart of each sales figure (X-axis: Q1 2018, Q2 2018, ... Q4 2020, and Y-axis showing corresponding sales).
- `tile[0,1]`: Displaying text: **2019 Sales** and below it the total sales of 2019 in large font.
- `tile[1,0]`: Pie chart of 2019 sales by quarter.
- `tile[1,1]`: the stacked barchart from LAB 3.