

## 1. Basic IPython usage

1. Try the 2 iPython modes
2. Create new cells above and below your current one
3. Try the following:
  - (a) In the 1st cell, set the variable `name` to your name. Execute the cell.
  - (b) In the 2nd cell, execute `print name`. Execute the cell.
  - (c) In the 3rd cell, redefine `name` to something else. Execute the cell.
  - (d) What will happen if you now rerun 2nd cell? Why? Would this be different if the notebook was a regular Python script?

## 2. DataFrames

Create a dummy dataframe:

```
df = pd.DataFrame('cov':[632, 1638, 569, 115, 433, 1130, 754, 555, 345], 'sample':['A', 'A', 'A', 'B', 'B', 'B', 'C', 'C'])
```

1. Use `head`, `tail` and `describe`
2. Use the `loc` method to...
  - (a) select rows 1-3.
  - (b) select all rows of the sample column.
  - (c) select rows 1-3 of chrom and cov.
3. Use the `iloc` method to select the first 5 rows and columns

## 3. Basic DataFrames

1. Load the quasar dataset into a pandas DataFrame using `read_csv`
2. Try out what happens if you load the DataFrame without using the correct `sep` argument
3. Try again the row/column selection you used on the dummy DataFrame
4. Save a subset of `df` into `df2` and compare the outputs of `describe`.

## 4. Counting and Boolean Indexing

1. How many different genes are there in the dataset?
2. Use boolean indexing to select all rows with `FAO > 50`.
3. How many rows are there with `FAO > 500`? And for `FAO > 1500`?
4. How many rows have an effect of either `'STOP_GAINED'` or `'FRAME_SHIFT'`? Hint: this is possible in one command by using the `.isin(list)` method.

## 5. Column creation

1. Replace the "Effect\_Impact" column with a lowercase version of itself. Hint: the `.str` method exposes a column as a string so you can use all the standard Python string functions.
2. Create a new column called DP as the sum of FAO and FDP.
3. Create a new column called len as the combined length of ref and alt.
4. Write a function that parses the Effect from a row and returns "strong" if it's in 'STOP\_GAINED' or 'FRAME\_SHIFT', otherwise "weak". Test the function using a dictionary (to simulate a row).
5. Create a new column "Binary\_Impact" by applying the function using `df.apply(function, axis=1)`.