MTMG06 - Statistics for weather and climate science

Practical 1 – Descriptive Statistics (not assessed)

These exercises are intended to help you to understand the concepts introduced in the lectures as well as to learn how to conduct, interpret and present basic statistical analyses. They are intended to be completed in order (later exercises assume knowledge acquired in earlier ones) and each exercise assumes that you have studied the matching section of the lecture notes.

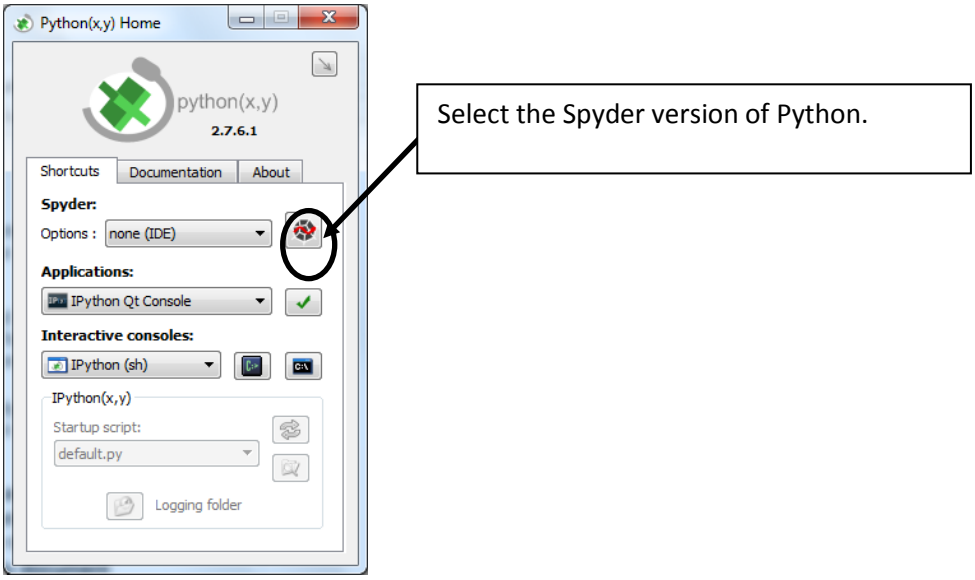
In this course we will use Python. Pyhton can be used for statistical computing and graphics. Python is free software available under the GNU General Public License (that means you can download it at home for free!). Python provides a wide array of statistical and graphical methods, which include (but are far from limited to) linear and nonlinear modelling, probability functions, statistical tests and time-series analysis, there are many different Python packages which can be imported to calculate your required statistics. There are many websites and books which provide useful help for Python if you get stuck; some of these are listed on the blackboard site under “Books and Links”.

In the practical notes, things you should see or type into the computer will be in a fixed width font (`like this`), and show the command prompt as seen in R. When text should be entered in place of filler, italics are used and placed within inverted commas (e.g. *“replace this”*). This method should clarify what is meant, this also applies during the lectures.

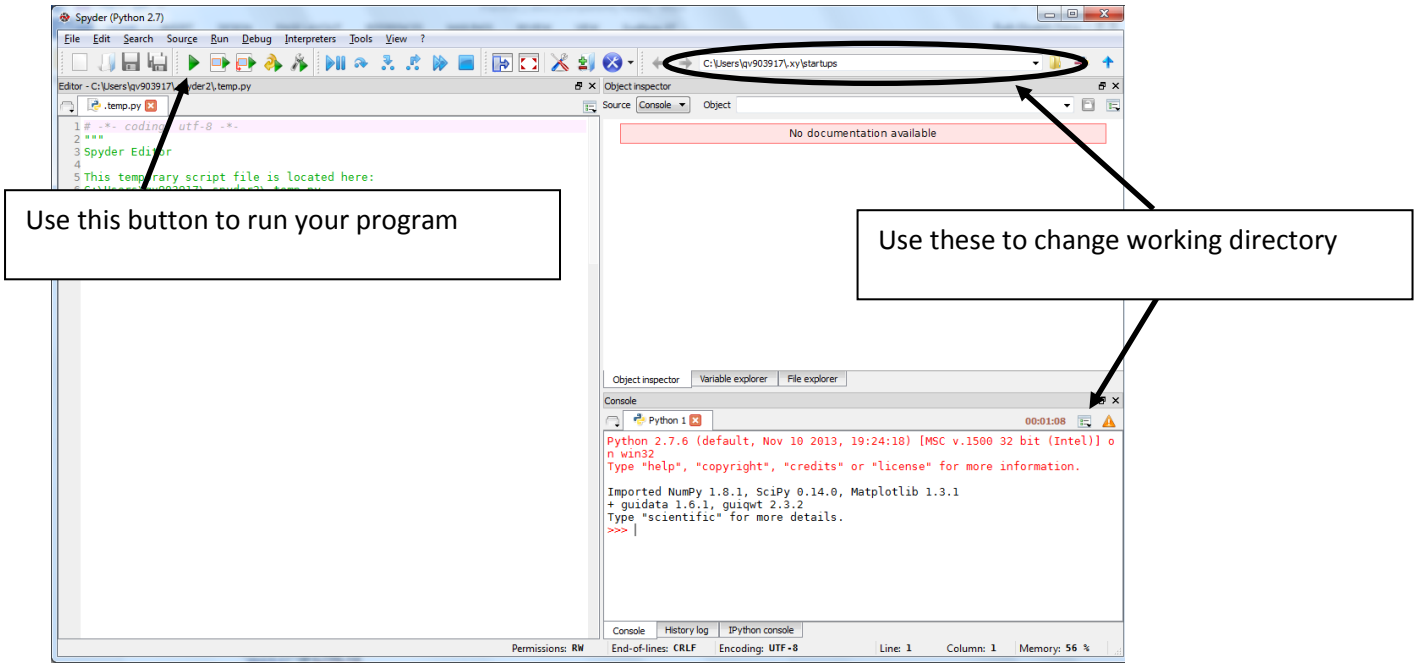
Part A – Getting started

1. Visit the blackboard site for this course – ensure you are enrolled correctly - if not tell a demonstrator ASAP to get you correctly enrolled. We will use blackboard to download data for the exercises.

2. Open Python, Go to the start menu, click on all programs and then click on Python(x,y). .



3. This will open an IDE session of Python, that looks like this:



From here you can type and execute commands directly in the bottom right hand pane. Alternatively you can make a Python program in the editor on the left. Files should be saved in the folder in which you are working, with the file extension "*.py*", for example "*practical1.py*". For example you should create a folder for the practical where you will store the data, the Python script, figures and report.

IMPORTANT: You will be expected to submit your Python code as well as your reports for the assessed exercises. Although this is not a numerical programming class a small number of marks will be awarded for the quality of the code that you submit. So remember that you should properly document your code, make proper use of functions and parameters, etc. The comment character for python is #, at the very top of your code you should write a brief description of the code, your name and other information you may find useful.

4. The "`help()`" command in python is extremely useful, try typing "`help(if)`", this will detail the syntax for an **if** loop in Python.

5. To exit the help menu type "`quit`" or press q. To exit Python completely type "`quit()`" or CTRL+d.

Part B: Statistics with Python

1. We will use some Python packages which have statistical routines built in (red commands from the lecture notes)
2. If running in interactive mode or at the start of a Python script you should import the packages that you will need. The ones that you will need for this practical are, matplotlib.pylab, numpy and scipy.stats to import these type:

```
>>> import matplotlib.pylab as plt
>>> import numpy as np
>>> import scipy.stats as sps
```

   The shorthand that is defined after as can be anything you choose, these are common shorthands but if you have a different preference you can use that.
   Using the shorthands means that instead of typing
   `>>> numpy.mean(data)`, you just need to type `>>> np.mean()`
3. From these packages you will be able to compute all the required statistics for this course.

Part C: Exercise 2: 0900 Temperatures

1. Download the file "temps_0910.csv" from blackboard; save this file in a directory where you will save work for this week's exercises. This file is of the temperatures recorded by the Reading University Atmospheric Obseratory observer from 2009 and 2010. The columns show the date (in decimal year), the dry bulb temperature at 0900 (in °C), the maximum daily temperature and minimum daily temperature (both in °C).

2. Read this into Python using (note you don't type the

   ```
   >>> temps=np.genfromtxt("temps_0910.csv", delimiter=",",skip_header=1)
   ```

   This loads the data into an array, the delimiter="," tells Python this has comma separations, in this simple case we have skipped reading in the header information. In this simple example we know that the columns are "Date", "Tdry", "Tmin" and "Tmax" Now to find the size of the array:

   ```
   >>> print np.shape(temps)
   ```

   This should tell you that you have an array with 730 rows and 4 columns.

   Allocate the Tdry data to a single numpy array for this exercise, recall that Python starts at counting from zero, the columns are date, Tdry, Tmin and Tmax, so date is column 0 etc.

3. Create a summary of the Tdry data.
   a. What is the problem?
   b. How would this affect your analysis?

4. The describe function cannot handle arrays with NaNs. Other python packages exist which can handle NaNs, such as pandas – if you are confident with Python you could try importing pandas as pd and using pd.describe.
   However for this exercise it will be simpler if we just remove the NaN and all the numpy and scipy.stats will work easily. You can do this with the command

   ```
   >>> y=x[np.isfinite(x)]
   ```

   Where `x` is the name of your original array and `y` is the new name – you should use sensible array names like tdry etc.

   Try the describe function again on your new array with the NaN removed.

5. Try calculating the mean and variance of the Tdry data set using `np.mean and np.var`.
   a. Do you still get the same value?
   b. Try using the commands `np.nanmean` and `np.nanvar`
   c. What is the standard deviation of the data?

6. Make a boxplot of the Tdry data.
   a. What is wrong with this plot?
   b. You can change the y-axis limits with the command `plt.ylim[(`y_{min}`,`y_{max}`)]`, where y_{min} is the lower limit and y_{max} is the upper limit, choose sensible upper and lower limits.
   c. Add a label to the y-axis with the command `plt.ylabel("Heights (cm)")`
   d. Change the x-tick label with the command `plt.xticks([1], ['Tdry'])`

7. Plot a histogram of the Tdry data with spacing of 5°C and then again with 1°C spacing. Python takes the number of bins as an argument to the histogram command `plt.hist(data,nbins)` in order to set the exact bin size you can instead use the command `plt.hist(data,bins=[-10,-5...30])` an easier way of doing this is to define the bins as an array first
   a. Use the command `np.linspace(-10,30,9)` to give you an array of elements starting at -10 and going to 30 in 9 steps which is the number you need increment by 5 at each step.

b. Define two arrays with sensible names and then use the command `plt.hist(data,bins=your_bin_array_name)`
c. Now plot the two different histograms one with the 5°C spacing and one with 1°C spacing.
d. Do these different bin sizes affect your view of the data?

8. What are the skewness and kurtosis of the distribution?

9. Plot the empirical cumulative distribution function for this data using the command `plt.hist(your_data_array_name, nbins, normed=1,histtype='step', cumulative=True)`. You will need to define a suitable number of nbins before using this command, try using different numbers of nbins.

    a. What is the probability that the temperature will be greater than 5°C?

10. If you have time using commands given in the lecture notes, order the data and calculate what the empirical probability that the Tdry will be greater than 5 °C?

---

11. Write a short report (max one page) describing the daily minimum and maximum temperature data in "temps_0910.csv". Use an approach similar to this (which is a good template for any exploratory data analysis).

    1. Ensure you understand the data and what it means.
    2. Decide what features merit investigation.
    3. Determine the statistics and plots to do this.
    4. Calculate those statistics and generate the plots.
    5. Interpret results with regard to the chosen aims.
    6. Consider if the results lead to further investigation?
       If so, repeat 2-6. Otherwise write up findings.

    Your report (as any) should contain
    - an explanation of the data,
    - a statement of the goals of your investigation,
    - a description of the methods used,
    - comments upon the results,
    - a summary of the conclusions of the analysis.

    The aim is to conduct statistical analysis, not overload the reader with graphs and tables, for this exercise only one or two figures should be used.

Further Exercises

12. Try producing a scatter plot, is there any relationship between the Tdry at 0900 and Tmax?

13. Investigate the use of quantiles, find what the 95% quantile of Tdry, Tmax and Tmin is.

14. Write some routines for the data transformations of centring and standardizing.

15. Perform a statistical analysis on the maximum and minimum temperatures.

An example report and sample code will be made available on blackboard before the next practical. Make sure you understand **all** the code and how to structure your report.