

Appendix A: Python commands

Starting and quitting Python

ipython notebook	starts an ipython notebook session
ctrl+enter	executes a command in cell. Note! If the output writes Out[*], Python is still working on it.
shift+enter	runs the cell and creates a new cell
ctrl+m s	saves the notebook
ipython	starts the terminal version of python
ctrl+C	abort command (in the terminal)
ctrl+D	quit Python (in the terminal)

Python as a calculator

x+z	addition
x-z	subtraction
x*z	multiplication
x/z	division. Note! Float division is not the default in Python (3/2 is not the same as 3/2.0). So we must remember to import the module division.
x**n	power
math.sqrt(x)	square root (import the module Math)
math.exp(x)	exponential function
math.log(x)	logarithm, base e. log10() has base 10
math.factorial(n)	factorial, n!

Modules

from __future__ import division	Float division (instead of typing 3/2.0)
import math	Useful for mathematical calculations. For details, check: https://docs.python.org/2/library/math.html
import numpy as np	Useful for dealing with N-dimensional array object
import pandas as pd	easy-to-use data structures and data analysis tool
import scipy.stats as stats	Statistical functions
import matplotlib.pyplot as plt	Plotting figures
from scipy.stats import hypergeom	Module for hypergeometric distribution
from scipy.stats import binom	Module for binomial distribution
from scipy.stats import poisson	Module for poisson distribution
from scipy.stats import geom	Module for geometric distribution
from scipy.stats import chi2	Module for chi square test
from pylab import *	Takes all definitions from the module and places them into your current namespace
from numpy import *	

Navigation in the notebook and terminal

home / end	jumps to the start / end of a line
ctrl+end+downarrow	In an active cell, this takes you to the next cell. Opposite: home + uparrow.
alt+enter	run cell, insert below
ctrl+m x	remove cell
alt+tab (ctrl+tab in the browser)	switch between active windows (works generally)
uparrow / downarrow (works in the terminal)	display previous / next command. If you write a character, then uparrow, you'll jump to the previous command starting with that character.
tab (in the terminal)	completes an object or file name

Vector and matrix operations

x = np.array([0,1,2,3,4,5,6,7,8,9])	create a vector, named x
y = np.arange(0, 11, 2)	sequence 0, 2, up to 10 (10 is included)
y = np.array([2,3]*100)	repeat 2,3 hundred times
x[0]	first element in vector x
x[-2]	second last element in vector x
M = np.arange(25).reshape((5,5))	create a 5 by 5 matrix, named M Alternatively, for y: M = y.reshape((10,20))
y = M[0, :]	first row in matrix M, stored as y
z = M[:, 2]	column 3 in matrix M, stored as z
M[M > 2]	all elements greater than 2 in M (array)
M[1, 3]	select the element in row 2 and column 4 of M
M = np.vstack((z,x))	combine x (in row 1) and z (in row 2)

np.round(z,2)	display two decimals of a float object
sum(z)	sum of all elements in x
max(z)	maximum of all elements in x
min(z)	minimum of all elements in x
df = pd.DataFrame(x)	Satistical calculation becomes easier when defining a dataframe. Here we defined array in to dataframe, df
df.count()	Number of non-null observation in df
df.mean()	mean (1 st moment)
df.median()	median
df.mode()	mode (most frequent number)
df.abs()	absolute value
df.std()	standard deviation
df.var()	variance (2 nd moment)
df.sem()	standard error of the mean
df.skew()	skewness (3 rd moment)
df.quantile(%)	sample quantile (value in %)
df.cumsum()	cumulative sum
df.cummax()	cumulative maximum
df.cummin()	cumulative minimum
df.kurt()	kurtosis (4 th moment)
M.shape	display dimension of matrix M
M.T	transpose matrix
M.sum()	sum of all elements in M
m.sum(axis = 1)	sum of each row in M
np.sum(M ,1)	sum of each row in M
np.sum(M ,0)	sum of each column in M
M.sum(axis = 0)	sum of each column in M
! pwd	shows which directory you'er in (where you save files).
import os	module for operating system change to you home
os.chdir("Documents")	directory. Check file path with ! pwd
dir()	lists all objects in the work space
del z	removes object z
P = read_table("name.csv", sep=",")	read a text file. sep="," when comma separated
p.head()	display first 6 rows of object
p.tail()	display last 6 rows of object
p.columns = ['col 1', 'col 2']	give name to columns <i>Note: It will replace the value in first row</i>
pd.read_table("name.csv", header=None, names=['col 1', 'col2'])	Assigning the columns name without replacing the value in first row
np.isnan(p).sum()	number of NAN values in matrix P
pd.concat([df 1, df2],axis =1)	merging two dataframes
plt.plot(x,z);plt.show()	plot x (x axis) and z (y axis)as line plot
plt.scatter(x,z);plt.show()	plot x (x axis) and z (y axis)as line plot
plt.plot(x,z,color="green",lin ewidth=3.0,linestyle="dashed") ;plt.show()	linestyle = solid for solid line: try different by changing color and linewidth
plt.ylim(-2, 10)	Y-limits from -2 to 10
plt.xlim(-2, 10)	X-axis limit from -2 to 10
plt.xlabel('xlabel')	x-axis label
plt.ylabel('ylabel')	Y-axis label
plt.title('title')	Title of the plot
plt.legend([plot1,plot2])	Legend to the plot
plt.hist(x, bins = 2)	histogram with two bins
plt.hist(x, bins = 10, normed = True)	relative histogram, ten bins
Plt.hist(x ,bins = 10, normed = True, cumulative = True)	cumulative relative histogram

[illegible]