```
import os, sys, os.path, shutil
import matplotlib as mpl

import xlrd, openpyxl
import hashlib
import pweave

mode_serv = False
for param in sys.argv[1:]:
    if param[:len("serv")]=="serv" and param[len("serv")] == "=":
        mode_serv = param[len("serv")+1:] in "1 y Y yes Yes YES"
#if not mode_serv:
#    mpl.use('TkAgg')
#else:
#    mpl.use('Agg')
if mode_serv:
    mpl.use('Agg')
try:
    import appnope
    appnope.nope()    # stop the apple power nap
except ImportError:
    pass

import pandas
from math import pi
import matplotlib.pyplot as plt
import matplotlib.colors as colors
import matplotlib.patches as mpatches
from matplotlib.collections import PatchCollection
import numpy as np
from numpy.random import rand
from pylab import pcolor, show, colorbar, xticks, yticks, pcolormesh, imshow
from random import gauss
from scipy.interpolate import interp2d
import threading, multiprocessing, time
from decimal import Decimal
import scipy

from scipy.optimize import curve_fit
mpl.rc('text', usetex=True)
#plt.rc('text', usetex=True)
mpl.rc('text.latex',preamble=r"\usepackage{amsmath} \usepackage{graphicx} \usepackage{ni

import scipy as sp
import subprocess
import datetime
from matplotlib.ticker import PercentFormatter
from mpl_toolkits.mplot3d import Axes3D

from mpl_toolkits.mplot3d.art3d import Poly3DCollection, Line3DCollection
import itertools

#vector of position P1 +1/-1 for right and left
v1=[0,0,0]
v2=[-1,0,0]
v3=[0,-1,0]
v4=[0,0,-1]
```

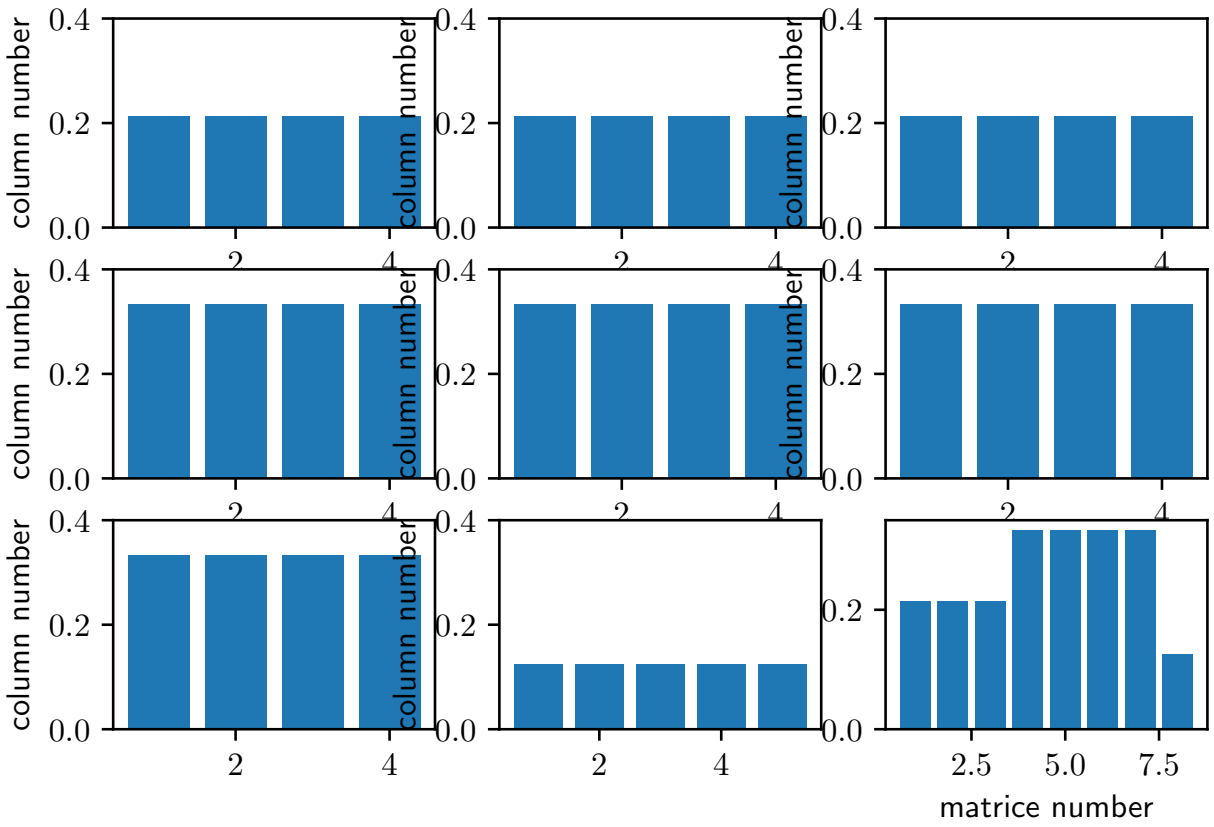
```
#factor matrix (either -1 or 1) for the five factor 6 sim as described in scriptum
m1 = np.mat([[-1,-1,-1,-1,-1],
       [1,-1,-1,-1,-1],
       [-1,1,-1,-1,-1],
       [-1,-1,1,-1,-1],
       [-1,-1,-1,1,-1],
       [-1,-1,-1,-1,1]])

m2 = np.mat([[1,1,1,1,1],
       [1,-1,-1,-1,-1],
       [-1,1,-1,-1,-1],
       [-1,-1,1,-1,-1],
       [-1,-1,-1,1,-1],
       [-1,-1,-1,-1,1]])

m3 = np.mat([[-1,-1,-1,-1,-1],
       [1,1,1,1,-1],
       [1,1,1,-1,1],
       [1,1,-1,1,1],
       [1,-1,1,1,1],
       [-1,1,1,1,1]])

m4 = np.mat([[-1,-1,-1,-1,-1],
       [1,1,-1,-1,-1],
       [-1,1,1,-1,-1],
       [-1,-1,1,1,-1],
       [-1,-1,-1,1,1],
       [1,-1,-1,-1,1]])

m5 = np.mat([[1,1,1,1,1],
       [1,1,-1,-1,-1],
       [-1,1,1,-1,-1],
       [-1,-1,1,1,-1],
       [-1,-1,-1,1,1],
       [1,-1,-1,-1,1]])

m6 = np.mat([[-1,-1,-1,-1,-1],
       [1,1,1,-1,-1],
       [-1,1,1,1,-1],
       [-1,-1,1,1,1],
       [1,-1,-1,1,1],
       [1,1,-1,-1,1]])

m7 = np.mat([[1,1,1,1,1],
       [1,1,1,-1,-1],
       [-1,1,1,1,-1],
       [-1,-1,1,1,1],
       [1,-1,-1,1,1],
       [1,1,-1,-1,1]])

H8 = np.mat(scipy.linalg.hadamard(8))
m8=H8[:,1:6]
print(m1)
print(m8)
```

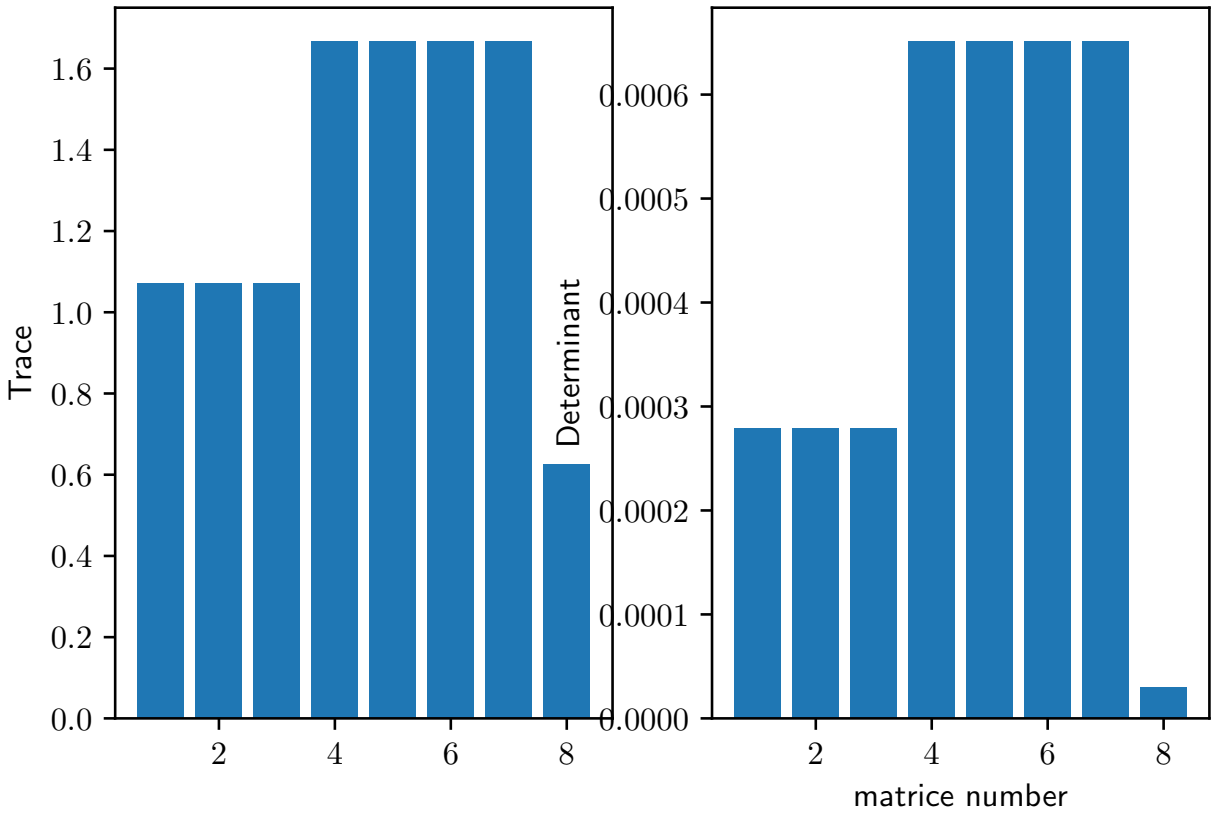
```
#dispersion matrix (X.T*X)^-1
d1= np.linalg.inv( np.matmul(m1.T,m1))
d2= np.linalg.inv( np.matmul(m2.T,m2))
d3= np.linalg.inv( np.matmul(m3.T,m3))
d4= np.linalg.inv( np.matmul(m4.T,m4))
d5= np.linalg.inv( np.matmul(m5.T,m5))
d6= np.linalg.inv( np.matmul(m6.T,m6))
d7= np.linalg.inv( np.matmul(m7.T,m7))
d8= np.linalg.inv( np.matmul(m8.T,m8))


print(d1)
print(d5)

print(d8)


fig = plt.figure()
ax = fig.add_subplot(331)
ax.bar(range(1,len(np.diag(d1))),np.diag(d1)[1:])
ax.set_ylim(0,0.4)
ax.set_ylabel("column number")
ax = fig.add_subplot(332)
ax.bar(range(1,len(np.diag(d2))),np.diag(d2)[1:])
ax.set_ylim(0,0.4)
ax.set_ylabel("column number")
ax = fig.add_subplot(333)
ax.bar(range(1,len(np.diag(d3))),np.diag(d3)[1:])
ax.set_ylim(0,0.4)
ax.set_ylabel("column number")
ax = fig.add_subplot(334)
ax.bar(range(1,len(np.diag(d4))),np.diag(d4)[1:])
ax.set_ylim(0,0.4)
ax.set_ylabel("column number")
ax = fig.add_subplot(335)
ax.bar(range(1,len(np.diag(d5))),np.diag(d5)[1:])
ax.set_ylim(0,0.4)
ax.set_ylabel("column number")
ax = fig.add_subplot(336)
ax.bar(range(1,len(np.diag(d6))),np.diag(d6)[1:])
ax.set_ylim(0,0.4)
ax.set_ylabel("column number")
ax = fig.add_subplot(337)
ax.bar(range(1,len(np.diag(d7))),np.diag(d7)[1:])
ax.set_ylim(0,0.4)
ax.set_ylabel("column number")
ax = fig.add_subplot(338)
ax.bar(range(1,len(np.diag(d1))+1),np.diag(d8))
ax.set_ylim(0,0.4)
ax.set_ylabel("column number")
ax = fig.add_subplot(339)
ax.bar(range(1,9),[d1[1,1],d2[1,1],d3[1,1],d4[1,1],d5[1,1],d6[1,1],d7[1,1],d8[1,1]])
ax.set_xlabel("matrice number")
c1=[d1[1,1],d2[1,1],d3[1,1],d4[1,1],d5[1,1],d6[1,1],d7[1,1],d8[1,1]]

fig = plt.figure()
ax = fig.add_subplot(121)
ax.bar(range(1,9),[np.trace(d1),np.trace(d2),np.trace(d3),np.trace(d4),np.trace(d5),np.t
```

```
ax.set_ylabel("Trace")
ax = fig.add_subplot(122)
ax.set_xlabel("matrice number")
ax.bar(range(1,9),[np.linalg.det(d1),np.linalg.det(d2),np.linalg.det(d3),np.linalg.det(d
ax.set_ylabel("Determinant")
ax.set_xlabel("matrice number")

c2=[np.trace(d1),np.trace(d2),np.trace(d3),np.trace(d4),np.trace(d5),np.trace(d6),np.tra
c3=[np.linalg.det(d1),np.linalg.det(d2),np.linalg.det(d3),np.linalg.det(d4),np.linalg.de

# vertices of a pyramid
# v = np.array([[-1, -1, -1], [1, -1, -1], [1, 1, -1],  [-1, 1, -1], [0, 0, 1]])
# v = np.array([v1,v2,v3,v4])
# ax.scatter3D(v[:, 0], v[:, 1], v[:, 2])

# generate list of sides' polygons of our pyramid
# verts = [ [v[0],v[1],v[2]], [v[0],v[2],v[3]], [v[1],v[2],v[3]]]
# verts = list(itertools.combinations(v,3))


# plot sides
# ax.add_collection3d(Poly3DCollection(verts,
 # facecolors=['r','g','b','y'], linewidths=1, edgecolors='r', alpha=.25))
# ax.set_xlim(-1,1)
# ax.set_ylim(-1,1)
# ax.set_zlim(-1,1)

fig = plt.figure()
df = pandas.DataFrame(data={'diag': c1, 'Trace': c2, 'Determinant': c3},index=['mat1','m
ax = plt.subplot(111, frame_on=False) # no visible frame
ax.xaxis.set_visible(False)  # hide the x axis
ax.yaxis.set_visible(False)  # hide the y axis

pandas.plotting.table(ax,df, loc='center')



plt.show()
```

```
[[-1 -1 -1 -1 -1]
 [ 1 -1 -1 -1 -1]
 [-1  1 -1 -1 -1]
 [-1 -1  1 -1 -1]
 [-1 -1 -1  1 -1]
 [-1 -1 -1 -1  1]]
[[ 1  1  1  1  1]
 [-1  1 -1  1 -1]
 [ 1 -1 -1  1  1]
 [-1 -1  1  1 -1]
 [ 1  1  1 -1 -1]
 [-1  1 -1 -1  1]
 [ 1 -1 -1 -1 -1]
 [-1 -1  1 -1  1]]
[[ 0.21428571 -0.03571429 -0.03571429 -0.03571429 -0.03571429]
 [-0.03571429  0.21428571 -0.03571429 -0.03571429 -0.03571429]
 [-0.03571429 -0.03571429  0.21428571 -0.03571429 -0.03571429]
 [-0.03571429 -0.03571429 -0.03571429  0.21428571 -0.03571429]
 [-0.03571429 -0.03571429 -0.03571429 -0.03571429  0.21428571]]
[[ 0.33333333 -0.16666667  0.08333333  0.08333333 -0.16666667]
```

```
 [-0.16666667  0.33333333 -0.16666667  0.08333333  0.08333333]
 [ 0.08333333 -0.16666667  0.33333333 -0.16666667  0.08333333]
 [ 0.08333333  0.08333333 -0.16666667  0.33333333 -0.16666667]
 [-0.16666667  0.08333333  0.08333333 -0.16666667  0.33333333]]
[[0.125 0.    0.    0.    0.   ]
 [0.    0.125 0.    0.    0.   ]
 [0.    0.    0.125 0.    0.   ]
 [0.    0.    0.    0.125 0.   ]
 [0.    0.    0.    0.    0.125]]
```





	diag	Trace	Determinant
mat1	0.2142857142857143	1.0714285714285714	0.00027901785714285686
mat2	0.2142857142857143	1.0714285714285714	0.00027901785714285686
mat3	0.2142857142857143	1.0714285714285714	0.00027901785714285686
mat4	0.33333333333333337	1.6666666666666665	0.0006510416666666665
mat5	0.33333333333333337	1.6666666666666665	0.0006510416666666665
mat6	0.33333333333333337	1.6666666666666665	0.0006510416666666665
mat7	0.33333333333333337	1.6666666666666665	0.0006510416666666665
Hadamard	0.125	0.625	3.0517578125000014e-05