

Jupyter Notebooks

June 30, 2020

1 Plotting Graphs in Python

This is a brief tutorial on how to plot graphs with python, we will be using the “D_mel_atlas.csv” dataset which is attached with the email containing the link to this notebook.

First we import the necessary libraries, which will include “seaborn”, “matplotlib” and “pandas”

```
[1]: import pandas as pd # for dataset manipulation
import matplotlib.pyplot as plt
import seaborn as sns

from mpl_toolkits import mplot3d #a necessary import for plotting 3D graphs

[2]: sns.set(style="white")
sns.set(style="whitegrid", color_codes=True) #setting stylistic choices for the
↳graphs

[3]: df = pd.read_csv("D_mel_atlas.csv") #importing our dataset
```

We will begin with plotting a distribution plot for the eve gene.

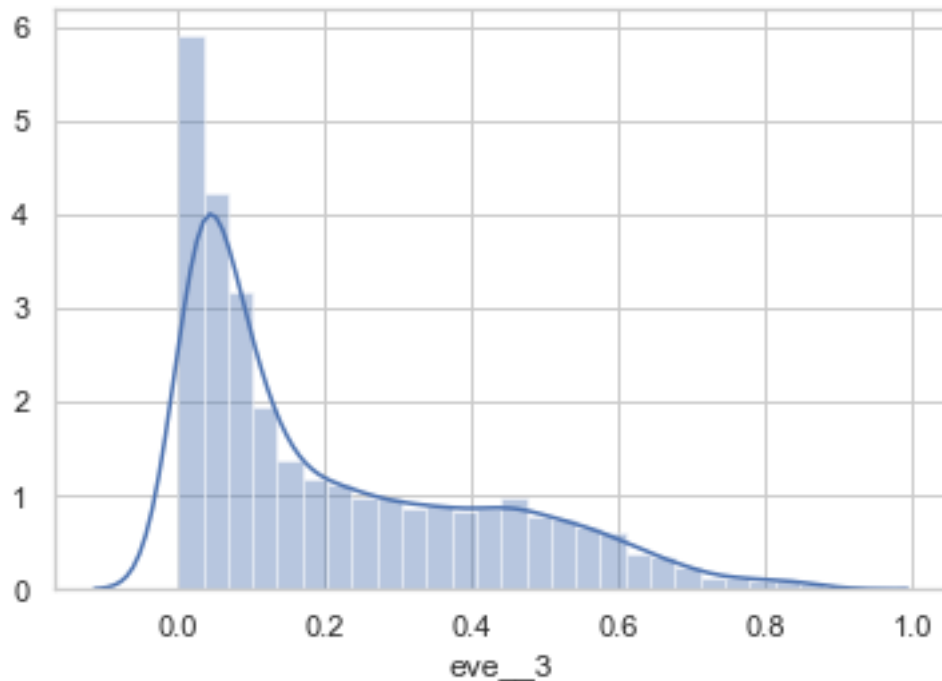
```
[4]: columns = df.columns.to_list()
print(columns) #always look at how the genes are named in the dataset before
↳starting

['id', 'x__1', 'y__1', 'z__1', 'Nx__1', 'Ny__1', 'Nz__1', 'CG31607__1',
'Cyp310a1__1', 'D__1', 'Kr__1', 'Traf1__1', 'bcd__1', 'cad__1', 'croc__1',
'eve__1', 'fkh__1', 'ftz__1', 'gt__1', 'h__1', 'hb__1', 'hkb__1', 'kni__1',
'knrl__1', 'noc__1', 'odd__1', 'prd__1', 'rho__1', 'slp1__1', 'slp2__1',
'sna__1', 'tll__1', 'trn__1', 'twi__1', 'zen__1', 'KrP__1', 'bcdP__1', 'gtP__1',
'hbP__1', 'x__2', 'y__2', 'z__2', 'Nx__2', 'Ny__2', 'Nz__2', 'CG10924__2',
'CG31607__2', 'D__2', 'Kr__2', 'Traf1__2', 'bcd__2', 'brk__2', 'cad__2',
'croc__2', 'eve__2', 'fkh__2', 'ftz__2', 'gt__2', 'h__2', 'hb__2', 'hkb__2',
'kni__2', 'knrl__2', 'noc__2', 'odd__2', 'prd__2', 'rho__2', 'slp1__2',
'slp2__2', 'sna__2', 'tll__2', 'trn__2', 'tsh__2', 'twi__2', 'zen__2', 'KrP__2',
'bcdP__2', 'gtP__2', 'hbP__2', 'x__3', 'y__3', 'z__3', 'Nx__3', 'Ny__3',
'Nz__3', 'CG10924__3', 'CG17786__3', 'CG4702__3', 'Cyp310a1__3', 'D__3',
'Dfd__3', 'Doc2__3', 'Kr__3', 'Traf1__3', 'brk__3', 'bun__3', 'cad__3',
'cnc__3', 'croc__3', 'emc__3', 'eve__3', 'fj__3', 'fkh__3', 'ftz__3', 'gt__3',
```

'h__3', 'hb__3', 'hkb__3', 'kni__3', 'knrl__3', 'oc__3', 'odd__3', 'path__3',
 'prd__3', 'rho__3', 'sala__3', 'slp1__3', 'slp2__3', 'sna__3', 'sob__3',
 'srp__3', 'term__3', 'tll__3', 'trn__3', 'tsh__3', 'twi__3', 'zen__3', 'KrP__3',
 'bcdP__3', 'gtP__3', 'hbP__3', 'x__4', 'y__4', 'z__4', 'Nx__4', 'Ny__4',
 'Nz__4', 'Alh__4', 'Ama__4', 'Ance__4', 'Blimp-1__4', 'Bsg25A__4', 'Btk29A__4',
 'CG10479__4', 'CG10924__4', 'CG11208__4', 'CG14427__4', 'CG17724__4',
 'CG17786__4', 'CG31607__4', 'CG31670__4', 'CG8147__4', 'CG8965__4',
 'Cyp310a1__4', 'D__4', 'Dfd__4', 'Doc2__4', 'Doc3__4', 'Esp__4', 'Ilp4__4',
 'ImpE2__4', 'Kr__4', 'MESR3__4', 'Mdr49__4', 'Mes2__4', 'Nek2__4', 'NetA__4',
 'Traf1__4', 'aay__4', 'apt__4', 'bmm__4', 'brk__4', 'bun__4', 'cad__4',
 'chrb__4', 'cnc__4', 'comm2__4', 'croc__4', 'dan__4', 'danr__4', 'disco__4',
 'dpn__4', 'emc__4', 'eve__4', 'fj__4', 'fkh__4', 'ftz__4', 'gk__4', 'gt__4',
 'h__4', 'hb__4', 'hkb__4', 'htl__4', 'jumu__4', 'ken__4', 'kni__4', 'knrl__4',
 'mfas__4', 'nub__4', 'numb__4', 'oc__4', 'odd__4', 'path__4', 'peb__4',
 'prd__4', 'pxb__4', 'rho__4', 'rib__4', 'sala__4', 'slp1__4', 'slp2__4',
 'sna__4', 'sob__4', 'srp__4', 'tkv__4', 'tll__4', 'toc__4', 'trn__4', 'tsh__4',
 'twi__4', 'zen__4', 'KrP__4', 'gtP__4', 'hbP__4', 'x__5', 'y__5', 'z__5',
 'Nx__5', 'Ny__5', 'Nz__5', 'Alh__5', 'Ama__5', 'Ance__5', 'Antp__5',
 'Blimp-1__5', 'Bsg25A__5', 'Btk29A__5', 'CG10479__5', 'CG10924__5',
 'CG13333__5', 'CG14427__5', 'CG17724__5', 'CG31670__5', 'CG4702__5',
 'CG8147__5', 'CG8965__5', 'Cyp310a1__5', 'D__5', 'Dfd__5', 'Esp__5', 'HLHm5__5',
 'Ilp4__5', 'ImpE2__5', 'ImpL2__5', 'Kr__5', 'MESR3__5', 'Mdr49__5', 'Nek2__5',
 'NetA__5', 'Traf1__5', 'aay__5', 'apt__5', 'bmm__5', 'brk__5', 'bun__5',
 'cad__5', 'cenG1A__5', 'chrb__5', 'cnc__5', 'comm2__5', 'croc__5', 'dan__5',
 'danr__5', 'disco__5', 'dpn__5', 'edl__5', 'emc__5', 'eve__5', 'fj__5',
 'fkh__5', 'ftz__5', 'gk__5', 'gt__5', 'h__5', 'hb__5', 'hkb__5', 'jumu__5',
 'kni__5', 'knrl__5', 'lok__5', 'mfas__5', 'noc__5', 'nub__5', 'numb__5',
 'oc__5', 'odd__5', 'path__5', 'peb__5', 'prd__5', 'pxb__5', 'rho__5', 'rib__5',
 'slp1__5', 'slp2__5', 'sna__5', 'sob__5', 'srp__5', 'term__5', 'tkv__5',
 'tll__5', 'toc__5', 'trn__5', 'tsh__5', 'twi__5', 'zen__5', 'KrP__5', 'gtP__5',
 'hbP__5', 'x__6', 'y__6', 'z__6', 'Nx__6', 'Ny__6', 'Nz__6', 'Ama__6',
 'Antp__6', 'Blimp-1__6', 'Bsg25A__6', 'Btk29A__6', 'CG10479__6', 'CG11208__6',
 'CG13333__6', 'CG14427__6', 'CG17724__6', 'CG17786__6', 'CG31607__6',
 'CG31670__6', 'CG8147__6', 'CG8965__6', 'Cyp310a1__6', 'D__6', 'Dfd__6',
 'Doc2__6', 'Doc3__6', 'Esp__6', 'HLHm5__6', 'Ilp4__6', 'ImpE2__6', 'ImpL2__6',
 'Kr__6', 'MESR3__6', 'Mdr49__6', 'Mes2__6', 'Nek2__6', 'NetA__6', 'Traf1__6',
 'aay__6', 'apt__6', 'bmm__6', 'brk__6', 'bun__6', 'cad__6', 'cenG1A__6',
 'cnc__6', 'comm2__6', 'croc__6', 'dan__6', 'danr__6', 'dpn__6', 'edl__6',
 'emc__6', 'eve__6', 'fj__6', 'fkh__6', 'ftz__6', 'gk__6', 'gt__6', 'h__6',
 'hb__6', 'hkb__6', 'jumu__6', 'ken__6', 'kni__6', 'knrl__6', 'lok__6',
 'mfas__6', 'noc__6', 'nub__6', 'numb__6', 'oc__6', 'odd__6', 'peb__6', 'prd__6',
 'pxb__6', 'rho__6', 'sala__6', 'slp1__6', 'slp2__6', 'sna__6', 'sob__6',
 'srp__6', 'tkv__6', 'tll__6', 'toc__6', 'trn__6', 'tsh__6', 'twi__6', 'zen__6',
 'KrP__6', 'gtP__6', 'hbP__6', 'Unnamed: 411', 'Unnamed: 412', 'Unnamed: 413',
 'Unnamed: 414', 'Unnamed: 415', 'Unnamed: 416', 'Unnamed: 417', 'Unnamed: 418',
 'Unnamed: 419']

```
[5]: sns.distplot(df["eve__3"]) # we will be using timepoint 3 for this example
```

```
[5]: <matplotlib.axes._subplots.AxesSubplot at 0x11a26e67e48>
```



As we can see, most of the cells have an eve expression ranging from 0.0 to ~0.2. As eve is expressed only in 7 stripes we expect only a small number of cells to have a high eve concentration. Therefore looking at the distribution plot, we can infer that the activation threshold should range around ~0.2 to ~0.4.

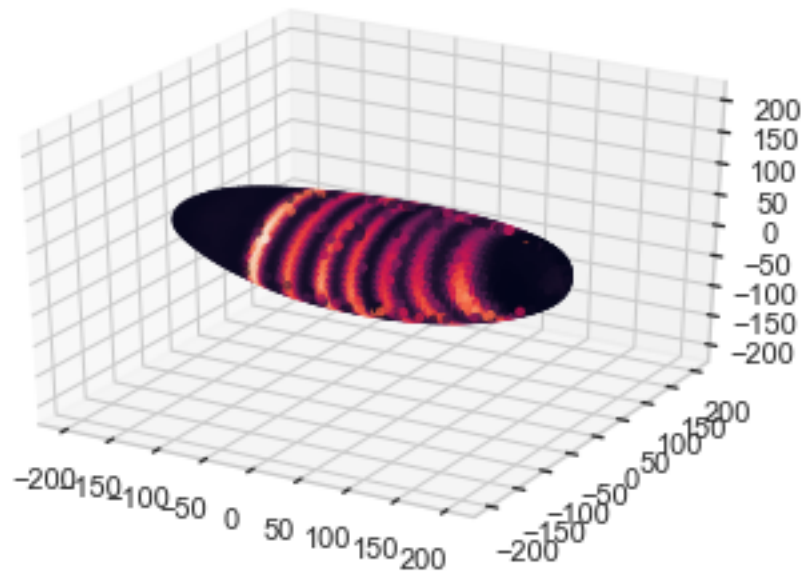
Look at the seaborns documentation for more customisation purposes and to expose yourself to the different types of graphs available for plotting.

Now, we will plot a 3D model of the Drosophila embryo to visualise eve expression by using eve to colorise our points.

```
[9]: ax = plt.axes(projection="3d") # prepares the ax variable for 3D plotting

#Now we set the limits of our plots
ax.set_xlim([-220,220])
ax.set_ylim([-220,220])
ax.set_zlim([-220,220])

ax.scatter(df["x__3"],df["y__3"],df["z__3"], c=df["eve__3"], s=30, marker=".")
plt.show()
```



Cells with high concentration of eve can be seen as red, producing a 3D graph with the distinct 7 stripes.

Now on your own, plot eve from the 6 different time points provided in the dataset: (i) Use a “for” loop to automate the process (ii) Set the angle so that graph outputs are consistent (iii) Remove the tick values (iv) Try using a different colormap outside of the default.