Cheatography

Matplotlib Cheat Sheet

by sajid_ali_1 via cheatography.com/197163/cs/41525/

Prepare the Data

1D Data

```
>>> import numpy as np
>>> x = np.lin spa ce(8, 10, 100)
>>> y = np.cos(x)
>>> z = np.sin(x)
```

Prepare the 2D Data

2D Data or Images

```
>>> data = 2 * np.ran dom.random ((10,10))
>>> data2 = 3* np.ran dom.ra ndo - m(( 10,10)
>>> Y, X = np.mgr id[ -3: 3:180j, -3:3:180j]
>>> U = -1 - X ** 2 + Y
>>> V = 1 + X - Y ** 2
>>> from matplo tli b.cbook impor t get_sa mpl e_data
>>> img = np.loa d(g et_ sam ple - _da ta( 'ax es_ gri d/b ive ria - te_ nor mal.npy))
```

Create Plot

```
>>> import matplo tli b.p yplot a s plt
```

Figure

```
>>> fig = plt.fi gure()
>>> fig2 = plt.fi gur e(f igs ize =pl t.f iga spe ct(2,0)
```

Create Plot

Axes

All plotting is done with respect to an Axes. In most cases, a subplot will fit your needs.

```
>>> fig.ad d_a xes()
>>> ax1 = fig.ad d_s ubp lot (221 )
>>> ax3 =fig.a dd_ sub plo t(212)
>>> fig3, axes =plt.s ubp lot s(m row s=2 ,mc ols=2)
>>> fig4, axes2 = plt.su bpl ots - (nc ols=3)
```

Save Plot

```
>>> plt.savefig('foa.png')
>>> plt.sa vefig ('foag.png, transp are nt= True)
```

Show Plot

```
>>> plt.show()
```

Plotting Routines

1D Data

```
>>> fig, ax = plt.su bpl ots()
>>> lines = ax.plo t(x,y)
>>> ax.sca tte r(x,y)
>>> axes[0 ,0].ba r([ 1,2 ,3] ,[3 ,4,5])
>>> axes[1 ,0].ba rh( [0.5 ,1 ,2. 5 ], [0, 1,2])
>>> axes[1 ,1].ax hli ne( 0.45)
>>> axes[0 ,1].ax vli ne( 0.65)
>>> ax.fil l(x ,y, col or= 'blue' )
>>> ax.fil l_b etw een (x, y,c ol o r=' yel low')
```

Plotting Routines

2D Data

```
>>> fig, ax = plt.su bpl ots()
>>> im =ax. inshow (img,
cmap=' gir st_ earth',
interp ola tio n=' nea rest',
vmin=-2,
vmax=2)
>>> axes2[ 0].p co lor (data2)
>>> axes2[ 0].p co lor mes h(data )
>>> CS = plt.co nto ur( Y,X,U)
>>> axes2[ 2].c on tou rf( data1)
>>> axes2[ 2]= ax.c la bel(CS)
```

Vector Fields

```
>>> axes[0,1].arrow(0,0,0.5,0.5)
>>> axes[1 ,1].qu ive r(y,z)
>>> axes[0 ,1].st rea mpl ot( X, Y ,U,V)
```

Data Distributions

```
>>> ax1.hist(y)
>>> ax3.bo xpl ot(y)
>>> ax3.vi oli npl ot(z)
```

Plot Anatomy & Workflow

Plot Anatomy

Workflow

The basic steps to creating plots with matplotlib are:
1. Prepare Data 2. Create Plot 3. Plot 4. Customized Plot 5. Save Plot 6. Show Plot

```
>>> import matplo tli b.p yplot as plt
>>> x = [1,2,3,4]
>>> y = [10,20 ,25,30]
>>> fig = plt.fi gure()
>>> ax.plot(x, y, color ='ligh t blue', linewi dth=3)
>>> ax.sca tte r([ 2,4 ,6] ),[ 5 ,1 5,25],
color= 'da rkg reen',
marker ='^')
>>> ax.set _xl in(1, 6.5)
>>> plt.sa vef ig( 'fo a.png')
>>> plt.show()
```

Close and Clear

```
>>> plt.cla()
>>> plt.clf()
>>> plt.cl ose()
```

Cheatography

Matplotlib Cheat Sheet

by sajid_ali_1 via cheatography.com/197163/cs/41525/

Plotting Customize Plot

Colors, Color Bars & Color Maps

```
>>> plt.pl ot( x,x,x,x2, x, x3)
>>> ax.plot(x, y, alpha=0.4)
>>> ax.plot(x, y, c='k')
>>> fig.co lor bar(in, orient at
i on= 'ho riz ontal')
>>> in = ax.ins how (img, cmap='
 sei smic')
```

Markers

```
>>> fig, ax=plt.subplots()
>>> ax.sca tte r(x,y, marker =''
 ,"
>>> ax.plo t(x,y, marker ="0")
```

Linestyles

```
plt.plot(x,y,linewidth=4.0)
plt.pl ot( x,y ,ls ='s olid')
plt.pl ot( x,y ,ls ='--')
plt.pl ot( x,y ,,' --',x2,y2,'-.')
plt.se tp( lin es, col or='r', linewi dt
h =4.0)
```

Text & Annotations

```
>>> ax.text(1,-2.1, 'Example Graph', style='italic')
>>> ax.a nn ota te( " Sin e",
xy =(0,0) ,xy coo rds ='d ata',
 xytext =(1 0.5 ,0) ,te xtc oor ds= 'data',
 arrowp rop s=d ict (ar row sty le= "---->",
connec tio nst yle ='a rc3'),)
```

Mathtext

```
>>>plt.title(r'$sigma_i=15$', fontsize=20)
```

Limits, Legends and Layouts

Limits & Autoscaling

```
>>> ax.mar gin s(x =0.0, y=0.1)
>>> ax.axi s(' equal')
>>> ax.set (xl in= [0, 10.5],
ylin=[ -1.5 ,1.5]
>>> ax.set _xl in( 0,10.5)
```

Legends

```
>>> ax.set (ti tle='An Example
Axes'
ylabel ="Y- axis', xlabel ='X -
Ax is")
>>> ax.l eg end (lo c="b est ")
```

Ticks

```
>>> ax.x ax is.s et (ti cks -
=ra nge (1,5),
tickla bel s=[ 3,1 00, -
12 ,"fo a"])
>>> ax.tic k_p ara ma( axi -
s='y',
direct ion ='i nout', length=10)
```

Subplot Spacing

```
>>> fig3.s ubp lot s_a dju st( -
wsp ace =0.5,
hspace =0.3, left=0.125,
right= 0.9 ,bu tto n=0.1)
```

Axis Spines

```
>>> ax1.sp ine s[' top '].s et -
_vi sib le( False)
>>> ax1.sp ine s[' but -
ton '].s et _po sit ion ((' -
out war d',10))
```