Python for Data Science
Seaborn: Statistical Data Visualization
Cheat Sheet

f616 *adapted from datacamp.com*

Introductory Note

This document is an adaption of the original datacamp.org cheat sheet.

- https://www.datacamp.com/resources/cheat-sheets/python-seaborn-statistical-data-visualization
- https://github.com/f616/Python-Seaborn-Cheat-Sheet

Statistical Data Visualization With Seaborn

The Python visualization library Seaborn is based on matplotlib and provides a high-level interface for drawing attractive statistical graphics.

Make use of the following aliases to import the libraries:

```python
import matplotlib.pyplot as plt
import seaborn as sns
```

The basic steps to creating plots with Seaborn are:

1. Prepare some data
2. Control figure aesthetics
3. Plot with Seaborn
4. Further customize your plot
5. Show your plot

```python
import matplotlib.pyplot as plt
import seaborn as sns
tips = sns.load_dataset('tips')  #Step 1
sns.set_style('whitegrid')  #Step 2
g = sns.lmplot(x='tip',  #Step 3
               y='total_bill',
               data=tips,
               aspect=2)
g = (g.set_axis_labels('Tip', 'Total bill(USD)').set(xlim=(0,10),ylim=(0,100)))
plt.title('title')  #Step 4
plt.show(g)  #Step 5
```

1 Data

```python
import pandas as pd
import numpy as np
uniform_data = np.random.rand(10, 12)
data = pd.DataFrame({'x':np.arange(1,101),
                     'y':np.random.normal(0,4,100)})
```

Seaborn also offers built-in data sets:

```python
titanic = sns.load_dataset('titanic')
iris = sns.load_dataset('iris')
```

2 Figure Aesthetics

```python
f, ax = plt.subplots(figsize=(5,6))  #Create a figure and one subplot
```

Seaborn Styles

```python
sns.set()  #(Re)set the seaborn default
sns.set_style('whitegrid')  #Set the matplotlib parameters
sns.set_style('ticks',  #Set the matplotlib parameters
              {'xtick.major.size':8,
               'ytick.major.size':8})
#Return a dict of params or use with with to temporarily set the style
ns.axes_style('whitegrid')
```

Context Functions

```python
sns.set_context('talk')  #Set context to "talk"
sns.set_context('notebook',  #Set context to "notebook",
                font_scale=1.5,  #Scale font elements and
                rc={'lines.linewidth':2.5})  #override param mapping
```

Color Palette

```python
sns.set_palette('husl',3)  #Define the color palette
ns.color_palette('husl')  #Use with with to temporarily set palette
flatui = ['#9b59b6' '#3498db' '#95a5a6' '#e74c3c' '#34495e' '#2ecc71']
sns.set_palette(flatui)  #Set your own color palette
```

3 Plotttting With Seaborn

Axis Grids

```python
g = sns.FacetGrid(titanic,
                  col='survived",
                  row='sex')  #Subplot grid for plotting conditional relationships
g = g.map(plt.hist,'age')
sns.factorplot(x='pclass',
               y='survived',
               hue='sex',
               data=titanic)  #Draw a categorical plot onto a Facetgrid
h = sns.PairGrid(iris)  #Subplot grid for plotting pairwise relationships
h = h.map(plt.scatter)
sns.pairplot(iris)  #Plot pairwise bivariate distributions
i = sns.JointGrid(x='x',
                  y='y',
                  data=data)  #Grid for bivariate plot with marginal univariate plots
i = i.plot(sns.regplot,sns.distplot)
sns.jointplot('sepal_length',  #Plot bivariate distribution
              'sepal_width',
              data=iris,
              kind='kde')
```

4 Further Customizations

Axisgrid Objects

```python
g.despine(left=True)  #Remove left spine
g.set_ylabels('Survived')  #Set the labels of the y-axis
g.set_xticklabels(rotation=45)  #Set the tick labels for x
g.set_axis_labels('Survived',
                  'Sex')  #Set the axis labels
h.set(xlim=(0,5),
      ylim=(0,5),
      xticks=[0,2.5,5],
      yticks=[0,2.5,5])  #Set the limit and ticks of the x-and y-axis
```

Plot

```
plt.title('A Title')  #Add plot title
plt.ylabel('Survived')  #Adjust the label of the y-axis
plt.xlabel('Sex')  #Adjust the label of the x-axis
plt.ylim(0,100)  #Adjust the limits of the y-axis
plt.xlim(0,10)  #Adjust the limits of the x-axis
plt.setp(ax,yticks=[0,5])  #Adjust a plot property
plt.tight_layout()  #Adjust subplot params
```

Regression Plots

```
#Plot data and a linear regression model fit
sns.regplot(x='sepal_width',
            y='sepal_width',
            data=iris,
            ax=ax)
```

Distribution Plots

```
#Plot univariate distribution
plot = sns.distplot(data.y,
                    kde=False,
                    color='b')
```

Matrix Plots

```
ns.heatmap(uniform_data,vmin=0,vmax=1)  #Heatmap
```

Categorical Plots

Scatterplot

```
#Scatterplot with one categorical variable
sns.stripplot(x='species',
              y='petal_length',
              data=iris)

#Categorical scatterplot with non-overlapping points
sns.swarmplot(x='species',
              y='petal_length',
              data=iris)
```

Bar Chart

```
#Show point estimates & confidence intervals with scatterplot glyphs
sns.barplot(x='sex',
            y='survived',
            hue='class',
            data=titanic)
```

Count Plot

```
#Show count of observations
sns.countplot(x='deck',
              data=titanic,
              palette='Greens_d')
```

Point Plot

```
#Show point estimates & confidence intervals as rectangular bars
sns.pointplot(x='class',
              y='survived',
              hue='sex',
              data=titanic,
              palette={'male':'g',
                       'female':'m'},
              markers=['^','o'],
              linestyles=['-','--'])
```

Boxplot

```
sns.boxplot(x='alive',
            y='age',
            hue='adult_male'
            data=titanic)  #Boxplot

sns.boxplot(data=iris,orient='h')#Boxplot with wide-form data
```

Violinplot

```
#Violin plot
sns.violinplot(x='age',
               y='sex',
               hue='survived',
               data=titanic)
```

5 Show or Save Plot

```
plt.show()  #Show the plot
plt.savefig('foo.png')  #Save figure
plt.savefig('foo.png',  #Save transparent figure
            transparent=True)
```

6 Close & Clear

```
plt.cla()  #Clear an axis
plt.clf()  #Clear an entire figure
plt.close()  #Close a window
```