Data Visualization with Seaborn: : CHEAT SHEET

Introductions

Seaborn is a library for making statistical graphics in Python. It builds on top of **matplotlib** and integrates closely with **pandas** data structures.

```
>>> import Mathplotlib.pylot as plt
>>> import seaborn as sns
```

Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

The basic steps of creating plots with Seaborn are:

1. Data preparation
2. Style control
3. Plot with Seaborn
4. Further customization

summary function

```
>>> df = sns.load_dataset( “tips” )
>>> sns.set_theme()
>>> g = sns.displot(data = df, x = “total_bill” ,col
= “time” )
>>> plt.title( “tips” )
```

Plt.show(g) shows the plot g
Plt.savefig(‘tips.png’ ,dpi = 1000) saves the last plot as file named ‘tips.png’ in working directory with 1000 dpi.

Data Preparation

In Seaborn, data can be created using pandas and numpy

```
>>> import pandas as pd
>>> import numpy as np
>>> df = pd.DataFrame({ ‘x’ :np.arrange(1,101),
 ‘y’ :np.random.normal(0,4,100)})
```

Seaborn also offers built-in data sets:

```
>>> planets = sns.load_dataset( ‘planets’ )
>>> iris = sns.load_dataset( ‘iris’ )
>>> Titanic = sns.load_dataset( ‘Titanic’ )
```

Controlling figure aesthetics

Seaborn styles and themes

```
>>> sns.set_theme()
>>> sns.set_style( ‘whitegride’ )
>>> sns.axes_style("whitegrid")
```

Scaling functions

Set different contexts and scale font elements

```
>>> sns.set_context( “paper” )
>>> sns. set_context( ‘notebook’ ,font_scale = 1.5,
rc = ( “lines.linewidth” :2.5))
```

Plots with Seaborn

Categorical data plotting

Seaborn provides three families of categorical plots: **Categorical scatterplots, Categorical distribution plots, Categorical estimate plots** and multiple relationships with facets

```
>>> sns.catplot(x="day", y="total_bill", data=tips);
>>> sns.catplot(x="total_bill", y="day", hue="sex",
kind="violin", data=tips)
>>> sns.catplot(y="deck", hue="class", kind="count",
palette="pastel", edgecolor=".6", data=titanic)
>>> sns.catplot(x="day", y="total_bill", hue="smoker",
col="time", aspect=.7, kind="swarm", data=tips)
```

Further Customizations

Set title, label of x-axis and y-axis and adjust subplots params

```
>>> plt.title("A Title")
>>> plt.xlabel("xlab")
>>> plt.ylabel("ylab")
>>> plt.tight_layout()
```

Show or save plots

```
>>> plt.show()
>>> plt.savefig( “temp.png “,
transparent=True
```

Plot Aesthetics

Choosing color palettes

Seaborn color palettes setting

```
>>> sns.set_palette( ‘magma’ ,3)
>>> sns.color_palette( ‘magma’ )
```

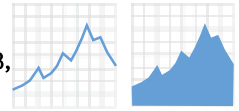
Setting you own color palette with Seaborn

```
>>> color_palate =
["#9b59b6","#3498db","#95a5a6","#e74c3c","#34495e","#2ecc71
"])
>>> sns.color_palette(color_palate)
```

Visualization statistical relationships

Relating variables with **scatter plots**, **line plots** and **facets**.

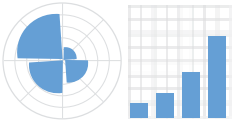
```
>>> sns.relplot(x="total_bill", y="tip", size="size", data=tips);
>>> sns.relplot(x="total_bill", y="tip", size="size",kind = "line",
data=tips);
>>> sns.relplot(x="timepoint", y="signal",
hue="subject",col="region", row="event", height=3,
kind="line", estimator=None, data=fmri);
```



Visualizing regression models

Seaborn provides regression plots and plots with different kinds of models

```
>>> sns.regplot(x="total_bill", y="tip", data=tips);
>>> sns.lmplot(x="x", y="y", data=anscombe.query("dataset ==
'II'"),order=2, ci=None, scatter_kws={"s": 80});
>>> sns.lmplot(x="total_bill", y="tip", col="day",
data=tips,aspect=.5);
```



Visualization distributions of data

Seaborn provides histograms, kernel density estimation, joint and marginal distributions plots

```
>>> sns.displot(penguins, x="flipper_length_mm", binwidth=3)
>>> sns.displot(penguins, x="flipper_length_mm", hue="species")
>>> sns.displot(penguins, x="flipper_length_mm", kind="kde “,
bw_adjust=.25)
>>> sns.jointplot(data=penguins, x="bill_length_mm",
y="bill_depth_mm")
```