Cheatography

Matplotlib, seaborn and plotly Cheat Sheet

by Anoikis via cheatography.com/74320/cs/18763/

Seaborn - 1 feature

`_, axes = plt.subplots(nrows=1, ncols=2, figsize=(12, 4))` `axis[0] = sns.xxx`	put a sns plot in a pyplot figure
`sns.distplot(df['feat']);`	histogram + density of a numeric feature's repartition
`sns.boxplot(data=df['feat']);`	simple boxplot
`sns.violinplot(data=df['feat']);`	simple violin plot
`sns.countplot(x='feat',data=df);`	repartition of a categorical feature

Seaborn - 2 features

`sns.heatmap(matrix)`	heatmap
`sns.jointplot(x=feat1, y=feat2, data=df, kind=['scatter', 'kde'])`	joint plot
`sns.pairplot(df[feats])`	scatterplots matrix
`sns.boxplot(x=feat1, y=feat2, data=df, ax=ax)`	boxplot for disjoint groups (x categorical)
`sns.violinplot(x=feat1, y=feat2, data=df, ax=ax)`	violin plot for disjoint groups

Seaborn - 2 features (cont)

`sns.countplot(x='feat1', hue='feat2', data=df);`	repartition of a categorical feature according to another one
`sns.lmplot(feat1, feat2, data=df, hue=feat3, fit_reg=False)`	scatterplot with color according to category
`sns.factorplot(x=cat1, y=numeric, col=cat2, data=df, kind="box", col_wrap=4, size=, aspect=);`	analyze a quantitative variable in 2 categorical dimensions at once

Dimensionallity reduction - t-SNE

```
from sklearn.manifold import TSNE
from sklearn.preprocessing
import StandardScaler

# Standardize data
scaler = StandardScaler()
tab_scaled = scaler.fit_transform(tab)

# Run t-SNE
tsne = TSNE(random_state=17)
tsne_repr = tsne.fit_transform(tab_scaled)

# Show results
plt.scatter(tsne_repr[:, 0], tsne_repr[:, 1],
c=df[feat].map({False: 'green', True: 'red'}));
```

Plotly

```
from plotly.offline import download_plotlyjs,
init_notebook_mode, plot, iplot
import plotly
import plotly.graph_objs as go
```

`trace0 = go.xxx`
`data = [trace0, trace1, ...]`
`layout = {'title': title, ..}`
`fig = go.Figure(data=data,`
`layout=layout)`
`iplot(fig, show_link=False)`
	general syntax

`trace = go.Scatter(x=feat1,` `y=feat2, name=)`	line
`trace = go.Bar(x=feat1, y=feat2 ,` `name=)`	barchart
`trace = go.Box(y=feat, name=genre)`	boxplot

`plotly.offline.plot(`
`fig,`
`filename='file.html',`
`show_link=False)`
	Save figure as an html file