

Matplotlib und Seaborn

Matplotlib ist ein ausgezeichnetes 2D-Visualisierungsmodul in Python. Seaborn ist eine auf Statistik spezialisierte Erweiterung von Matplotlib.

Imports

Zusammen mit Matplotlib und Seaborn werden häufig NumPy und Pandas zur Datenbearbeitung vor der Visualisierung verwendet.

import matplotlib.pyplot as plt  
import seaborn as sns  
import numpy as np  
import pandas as pd

Datenvorbereitung

x = np.linspace(0, 5, 50)  
y = np.cos(x)  
z = np.sin(x)  
a, b = np.mgrid[-1:1:50j, -1:1:50j]  
data = pd.DataFrame({'x': np.arange(-10,40),  
                      'y': np.random.normal(-1,3,50)})  
img = plt.imread("bild.jpg")

Figure erzeugen

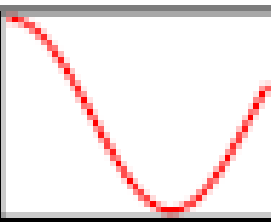
fig ist eine darzustellende Graphik **Figure**, die mindestens eine Instanz von **Axes** (Achsen) enthält.  
fig, axes = plt.subplots()  
Mehrere **Axes** sind indizierbar mit [Zeile, Spalte]:  
fig, four\_axes = plt.subplots(nrows=2,ncols=2)

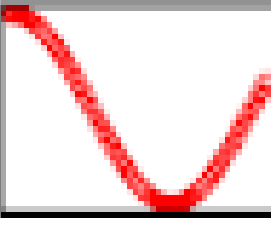
Layout

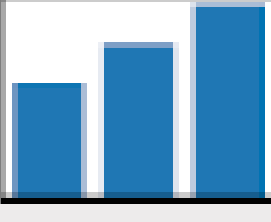
axes.margins(x=0.9,y=0.9)      Padding  
                                    Gleiches  
axes.axis('equal')            Seitenverhältnis  
                                    Achsen  
axes.set(                      Achsen  
    xlim=[-1,8.6],ylim=[-2.5,1])      begrenzen  
axes.set(title='Name',      Plot und Achsen  
      ylabel='Y', xlabel='X')      beschriften  
axes.legend(loc='best')      Legende erzeugen  
axes.xaxis.set(              Manuelle  
    ticks=range(2,10),      Marker  
    ticklabels=['test',6.1,-2,1])  
axes.tick\_params(            Marker  
    axis='y', direction='inout'      Layout  
    length=15)  
fig.subplots\_adjust(      Achsenabstände  
    wspace=0.3, hspace=0.2,  
    left=0.4, right=0.4,  
    top=0.9, bottom=0.2)  
fig.tight\_layout()      Achsen zu **Figure**  
                            skalieren  
axes.spines['bottom'] \      Unsichtbare  
    .set\_visible(False)      Achsenbegrenzung  
axes.spines['top'].set \      Achsenbegrenzung  
    \_position(('outward',5))      verschieben


Matplotlib Plotting


Plottingvarianten

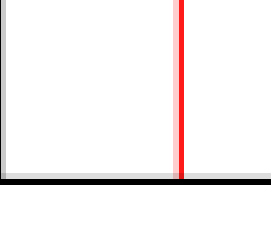
 Linie  
axes.plot(x,y)

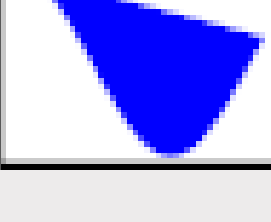
 Scatterplot  
axes.scatter(x,y)

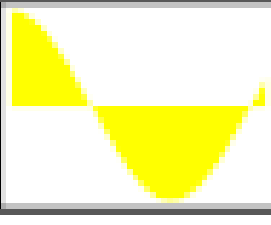
 Vertikale Balken  
axes.bar(4,0,3], [1,6,5])

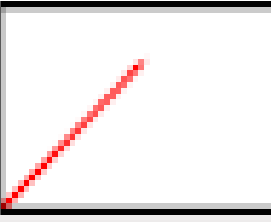
 Horizontale Balken  
axes.barh(4,0,3], [1,6,5])


 Horizontale Linie  
axes.axhline(0.35)

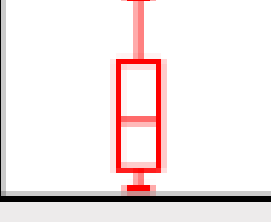
 Vertikale Linie  
axes.avhline(0.69)


 Fülle als Polygon  
axes.fill(x, y)

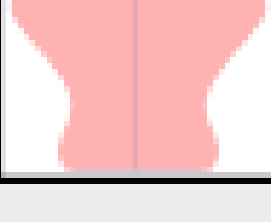
 Fülle zu y  
axes.fill\_between(x,y,color='yellow')

 Trendpfeil an der Position  
axes.arrow(0,0,0.5,0.7)

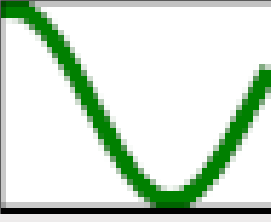
 Pfeile entlang Daten  
axes.quiver(y,z)

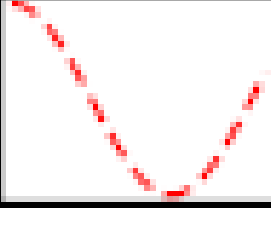
 Boxplot  
axes.boxplot(y)


 Histogramm  
axes.hist(y, color="b")

 Violinplot  
axes.violinplot(z)

Linientyp

 plt.setp(lines,color='g',linewidth=25)  
axes.plot(x,y)

 axes.plot(x,y,ls='-.')

 axes.plot(x,y,'-',x\*\*2,y\*\*2,'-.-')

Annotierung

axes.text(1, 0.6, 'Beispiel', style='italic')  
axes.annotate("Wichtig", textcoords='data',  
              xy=(2, -0.5), xycoords='data', xytext=(0, -0.5),  
              arrowprops=dict(arrowstyle=->))  
plt.suptitle(r'\$min\_y=1\$', fontsize=20,)

Seaborn

Datensätze

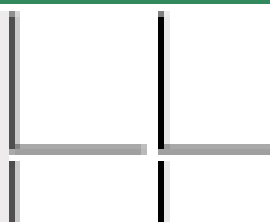
Seaborn beinhaltet Beispieldatenätze, unter anderem  
titanic = sns.load\_dataset("titanic")  
iris = sns.load\_dataset("iris")

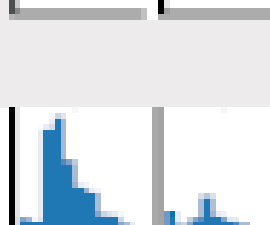
Darstellung

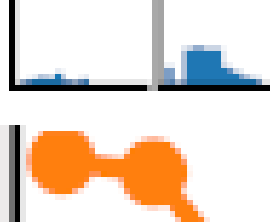
sns.set\_style("darkgrid")      Dunkler  
                                    Hintergrund  
sns.set\_style("ticks",      Markergrösse  
    {"xtick.major.size": 10,  
     "ytick.major.size": 4})  
sns.set\_context("talk",      Layoutklasse,  
    font\_scale=1.3      Linien- und  
                            Schriftgrösse  
    rc={"lines.linewidth":2.8})      Farbpalette  
                                    mit Anzahl  
                                    Farben  
sns.set\_palette("pastel",2)      Eigene  
                                    Palette

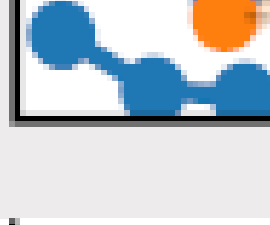
palette = ["#aaaaaa",  
          "#bbbbbb"]  
sns.set\_palette(palette)

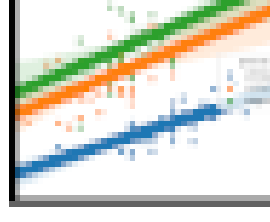
Achsenraster

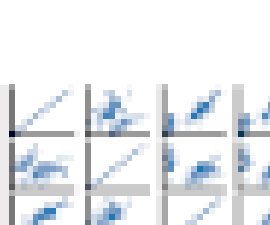
 Achsenraster  
p = sns.FacetGrid(titanic,  
                  col="survived", row="sex")


 Ein Subplot pro Kategorie  
p.map(plt.hist,"age")


 Alle Kategorien in einem Plot  
sns.factorplot(x="pclass",  
              data=titanic)

 Scatterplot mit Regression  
sns.lmplot(x="sepal\_width",  
          y="sepal\_length",  
          hue="species", data=iris)

 Paarweise Abhängigkeiten  
t = sns.PairGrid(iris)  
t = t.map(plt.scatter)

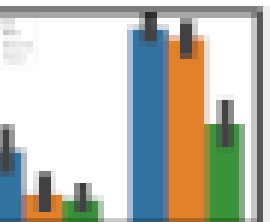
 Paarweise bivariate Verteilungen  
sns.pairplot(iris)


 Bivariater Plot mit univariatem Rand  
v = sns.JointGrid(x="x", y="y",  
                  data=data)  
v = v.plot(sns.regplot, sns.distplot)

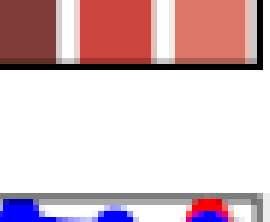
 Bivariate Verteilung  
sns.jointplot("sepal\_length",  
              "sepal\_width", data=iris,  
              kind='resid')


Weiteres

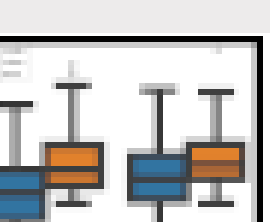
Kategorieplots


 Barplot mit Konfidenzintervall  
sns.barplot(x="sex",  
          y="survived", hue="class",  
          data=titanic)

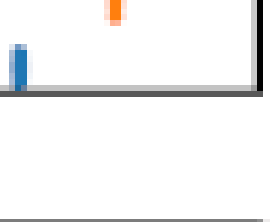
 Anzahl Datenpunkte  
sns.countplot(data=titanic,  
              x="pclass", palette="Reds\_d")

 Punktplot mitKonfidenzintervall  
sns.pointplot(x="class",  
              y="survived", hue="sex"  
              data=titanic)


 Boxplot mitKonfidenzintervall  
sns.boxplot(x="alive", y="age",  
             hue="adult\_male",  
             data=titanic)


 Scatterplot  
sns.stripplot(x="species",  
              y="petal\_length", data=iris)

 Scatterplot ohne Überlapp  
sns.swarmplot(x="species",  
              y="petal\_length", data=iris)

 Violinenplot  
sns.violinplot(x="age", y="sex",  
              hue="survived", data=titanic)

Weitere Plots

 Univariate Verteilung  
plot = sns.distplot(data.y, )  
      kde=False)

 Regressionsplot  
sns.regplot(x="sepal\_width",  
          y="sepal\_length", data=iris,  
          ax=axes[0,0])

Anzeigen

plt.show()

Speichern

plt.savefig('name.png')

Schließen

plt.cla()      Achsen schließen  
plt.clf()      Figure schließen  
plt.close()    Plot schließen