

Python 数据科学 速查表

Seaborn

用 Seaborn 绘制统计型数据可视图

Seaborn 是基于 matplotlib 开发的高阶 Python 数据可视图库，用于绘制优雅、美观的统计图形。

使用下列别名导入该库：

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

使用 Seaborn 创建图形的基本步骤：

1. 准备数据
2. 设定画布外观
3. 使用 Seaborn 绘图
4. 自定义图形

```
>>> import matplotlib.pyplot as plt
>>> import seaborn as sns
>>> tips = sns.load_dataset("tips")
>>> sns.set_style("whitegrid")
>>> g = sns.lmplot(x="tip", 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")
>>> plt.show(g)
```

1 数据 [参阅列表、Numpy 及 Pandas](#)

```
>>> 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 提供了内置数据集：

```
>>> titanic = sns.load_dataset("titanic")
>>> iris = sns.load_dataset("iris")
```

2 画布外观

```
>>> f, ax = plt.subplots(figsize=(5,6))
```

创建画布与子图

Seaborn 样式

```
>>> sns.set()
>>> sns.set_style("whitegrid")
>>> sns.set_style("ticks", {"xtick.major.size":8, "ytick.major.size":8})
>>> sns.axes_style("whitegrid")
```

设置或重置 Seaborn 默认值
设置 matplotlib 参数

返回参数字典或用 with 设置临时样式

3 使用 Seaborn 绘图

坐标轴栅格

```
>>> g = sns.FacetGrid(titanic, col="survived", row="sex")
>>> g = g.map(plt.hist, "age")
>>> sns.factorplot(x="pclass", y="survived", hue="sex", data=titanic)
>>> sns.lmplot(x="sepal_width", y="sepal_length", hue="species", data=iris)
```

绘制条件关系的子图栅格

在分面栅格上绘制分类图

绘制适配分面栅格的数据与回归模型

各类图形

```
>>> sns.stripplot(x="species", y="petal_length", data=iris)
>>> sns.swarmplot(x="species", y="petal_length", data=iris)
```

含分类变量的散点图

不重叠分类散点图

条形图

```
>>> sns.barpplot(x="sex", y="survived", hue="class", data=titanic)
```

用散点图符号
显示点估计值和置信区间

计数图

```
>>> sns.countplot(x="deck", data=titanic, palette="Greens_d")
```

显示观测数量

点图

```
>>> sns.pointplot(x="class", y="survived", hue="sex", data=titanic, palette={"male":"g", "female":"m"}, markers=["^","o"], linestyle=["-","--"])
```

用柱状图
显示点估计和置信区间

箱型图

```
>>> sns.boxplot(x="alive", y="age", hue="adult_male", data=titanic)
>>> sns.boxplot(data=iris, orient="h")
```

箱形图

使用宽表数据的箱型图

小提琴图

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

小提琴图

```
>>> h = sns.PairGrid(iris)
>>> h = h.map(plt.scatter)
>>> sns.pairplot(iris)
>>> i = sns.JointGrid(x="x", y="y", data=data)
>>> i = i.plot(sns.regplot, sns.distplot)
>>> sns.jointplot("sepal_length", "sepal_width", data=iris, kind="kde")
```

绘制配对关系的子图栅格
绘制配对的双变量分布
绘制双变量图的边际单变量图栅格

绘制双变量分布

回归图

```
>>> sns.regplot(x="sepal_width", y="sepal_length", data=iris, ax=ax)
```

绘制与线性回归模型拟合的数据

分布图

```
>>> plot = sns.distplot(data.y, kde=False, color="b")
```

绘制单变量分布

矩阵图

```
>>> sns.heatmap(uniform_data, vmin=0, vmax=1)
```

热力图

4 深度自定义

[参阅 Matplotlib](#)

Axisgrid 对象

```
>>> g.despine(left=True)
>>> g.set_ylabels("Survived")
>>> g.set_xticklabels(rotation=45)
>>> g.set_axis_labels("Survived", "Sex")
>>> h.set(xlim=(0,5), ylim=(0,5), xticks=[0,2.5,5], yticks=[0,2.5,5])
```

移除左框
设置Y轴的标签
设置X轴刻度标签
设置坐标轴标签
设置X与Y轴的限制和刻度

图形

```
>>> plt.title("A Title")
>>> plt.ylabel("Survived")
>>> plt.xlabel("Sex")
>>> plt.ylim(0,100)
>>> plt.xlim(0,10)
>>> plt.setp(ax, yticks=[0,5])
>>> plt.tight_layout()
```

添加图形标题
调整y轴标签
调整x轴标签
调整y轴限制
调整x轴限制
调整图形属性
调整子图参数

5 显示或保存图形

[参阅 Matplotlib](#)

```
>>> plt.show()
>>> plt.savefig("foo.png")
>>> plt.savefig("foo.png", transparent=True)
```

显示图形
将画布保存为图形
保存透明画布

关闭与清除

[参阅 Matplotlib](#)

```
>>> plt.cla()
>>> plt.clf()
>>> plt.close()
```

清除坐标轴
清除画布
关闭窗口

原文作者

DataCamp
Learn Python for Data Science Interactively

