Introduction to data science

Patrick Shafto

Department of Math and Computer Science

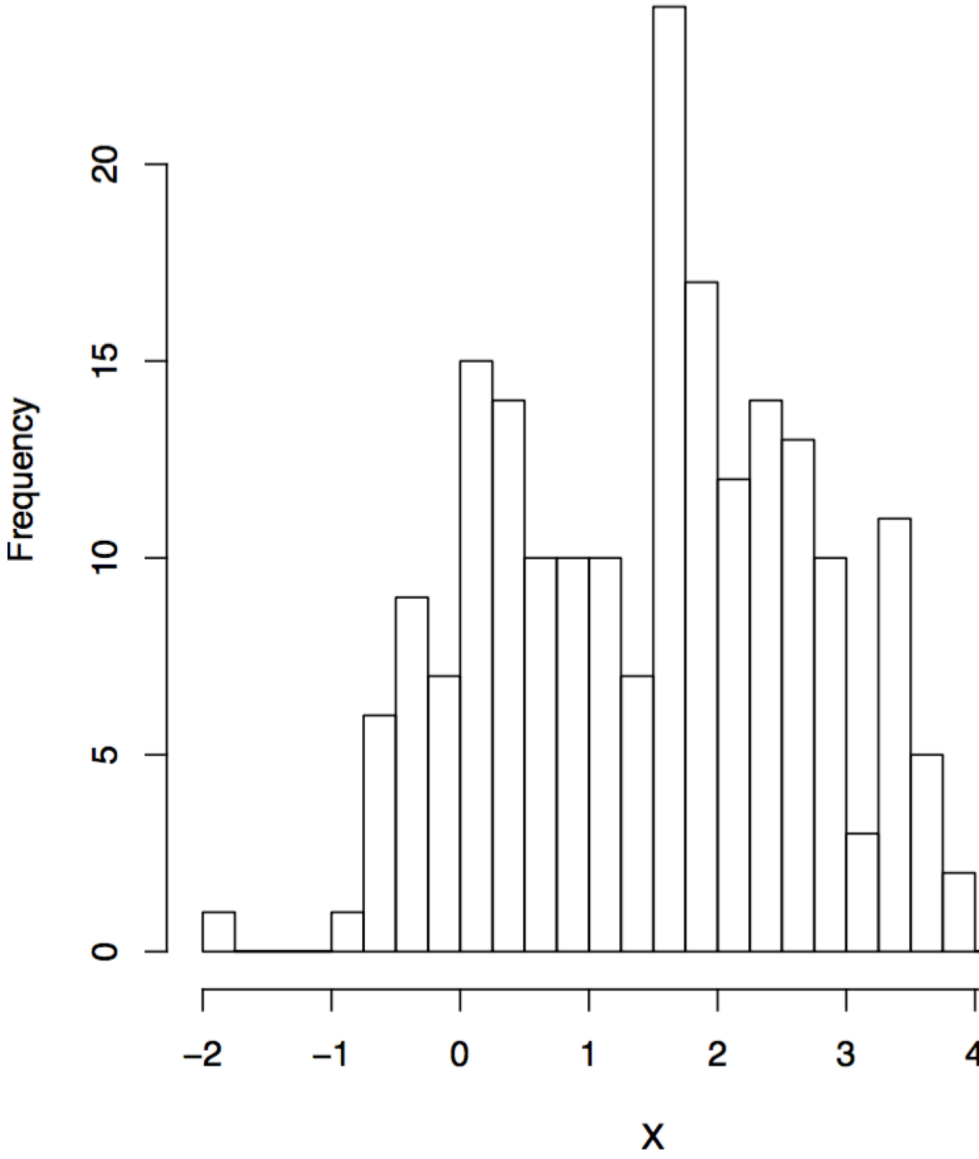
Plan for today

- Viz
- HW: Finish up data cleaning

Cleaning data / EDA

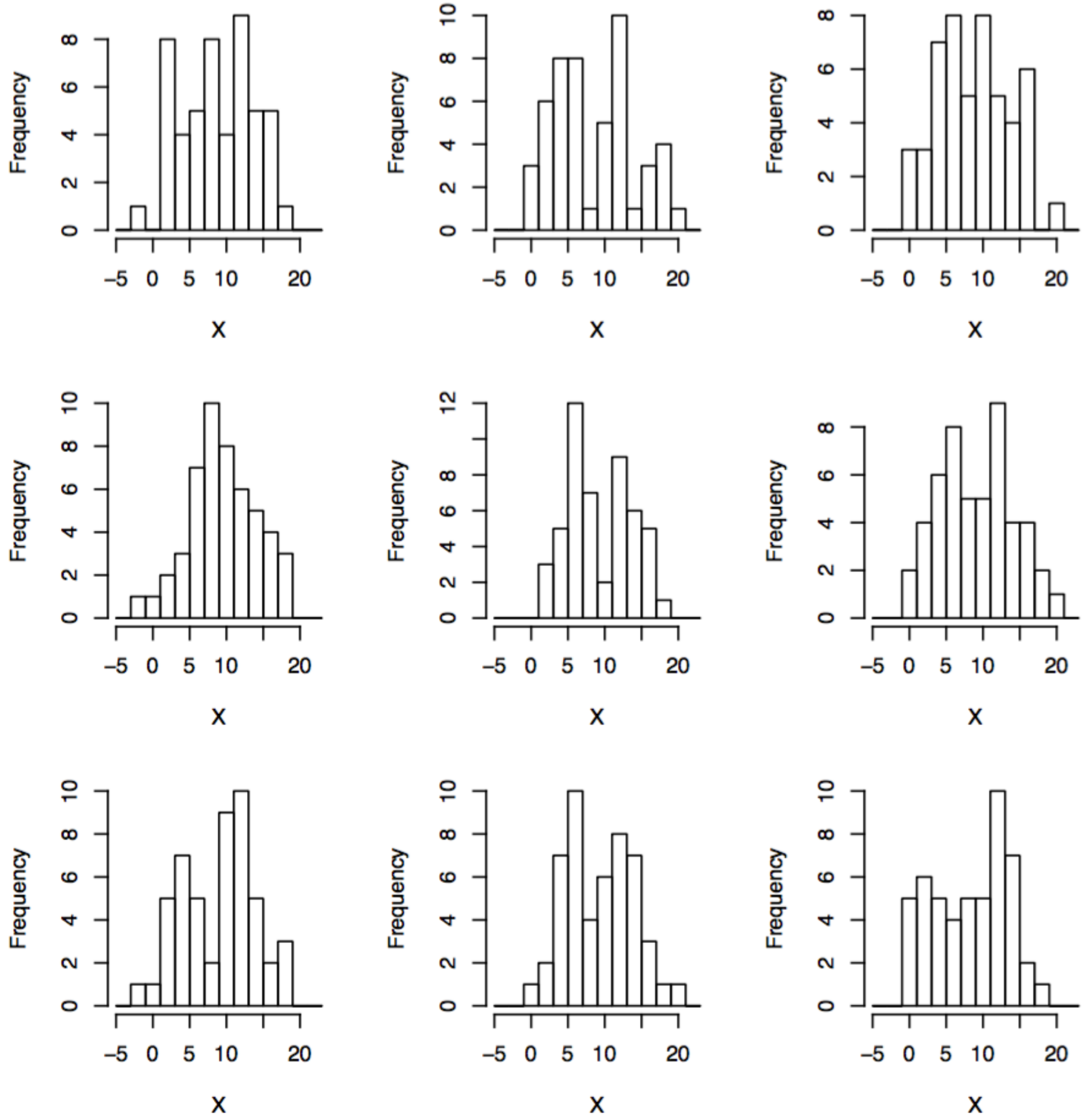
Univariate non-graphical EDA

- Characteristics of quantitative data
  - Histogram
  - Modes? Shape? Outliers?



Univariate graphical EDA

- Histogram
  - Variability is expected!

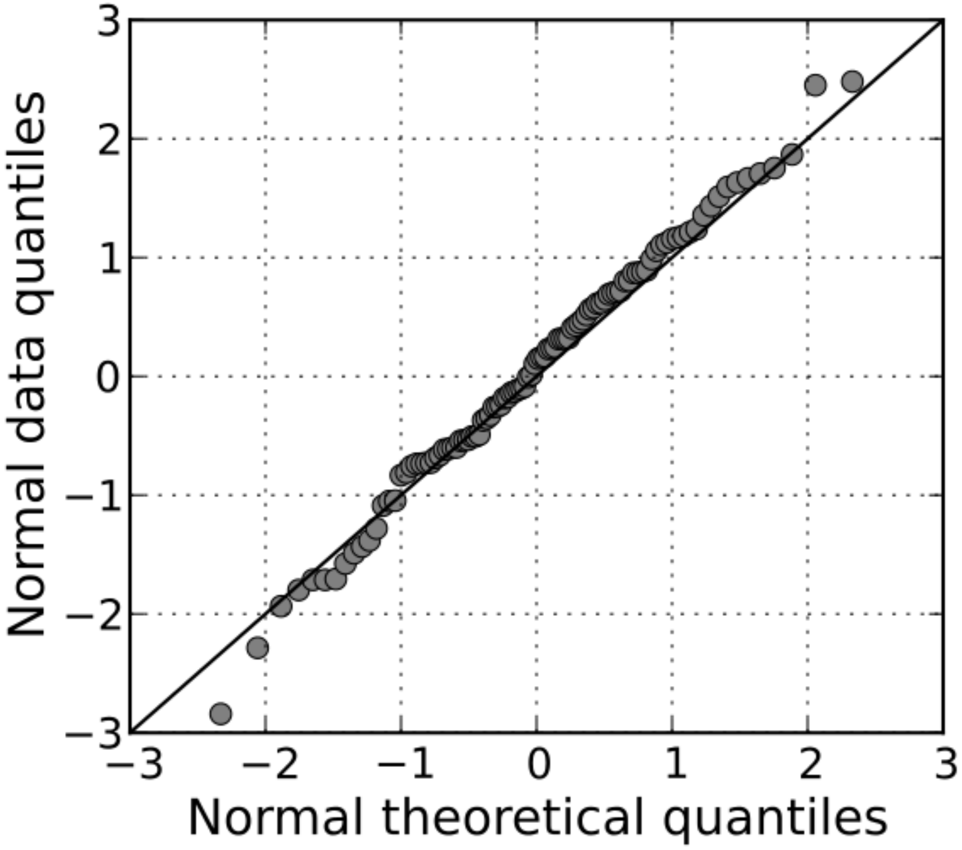


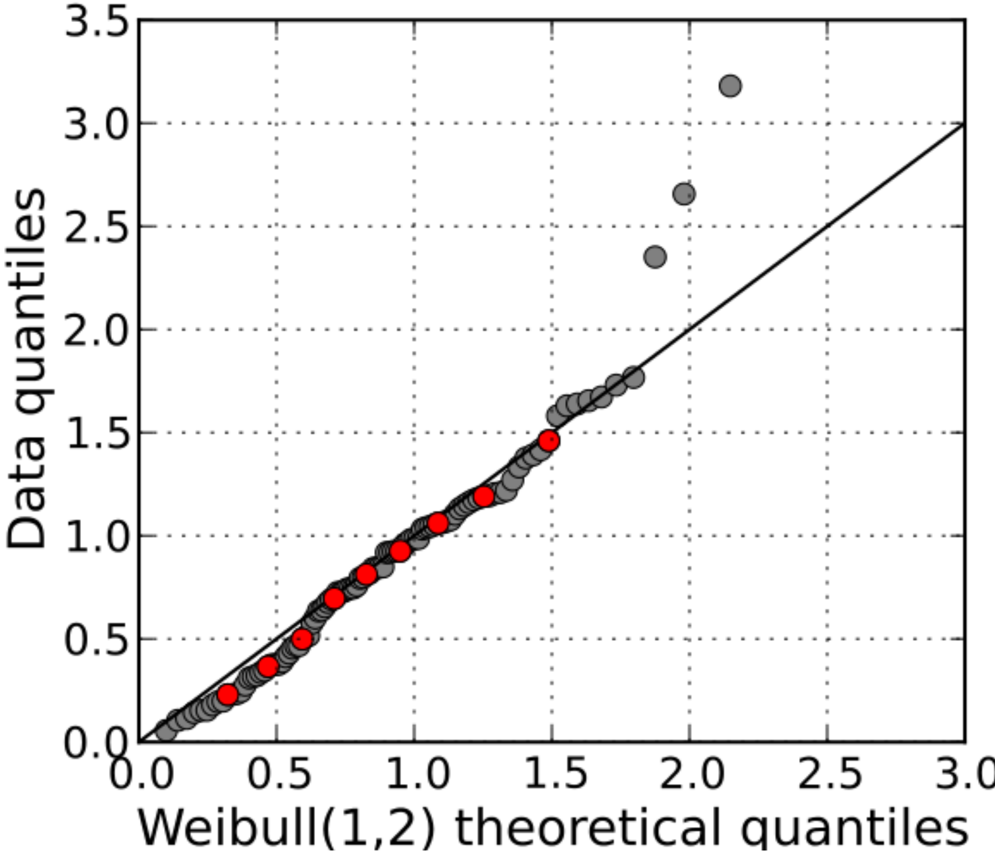
Univariate graphical EDA

- Also, box plots, violin plots

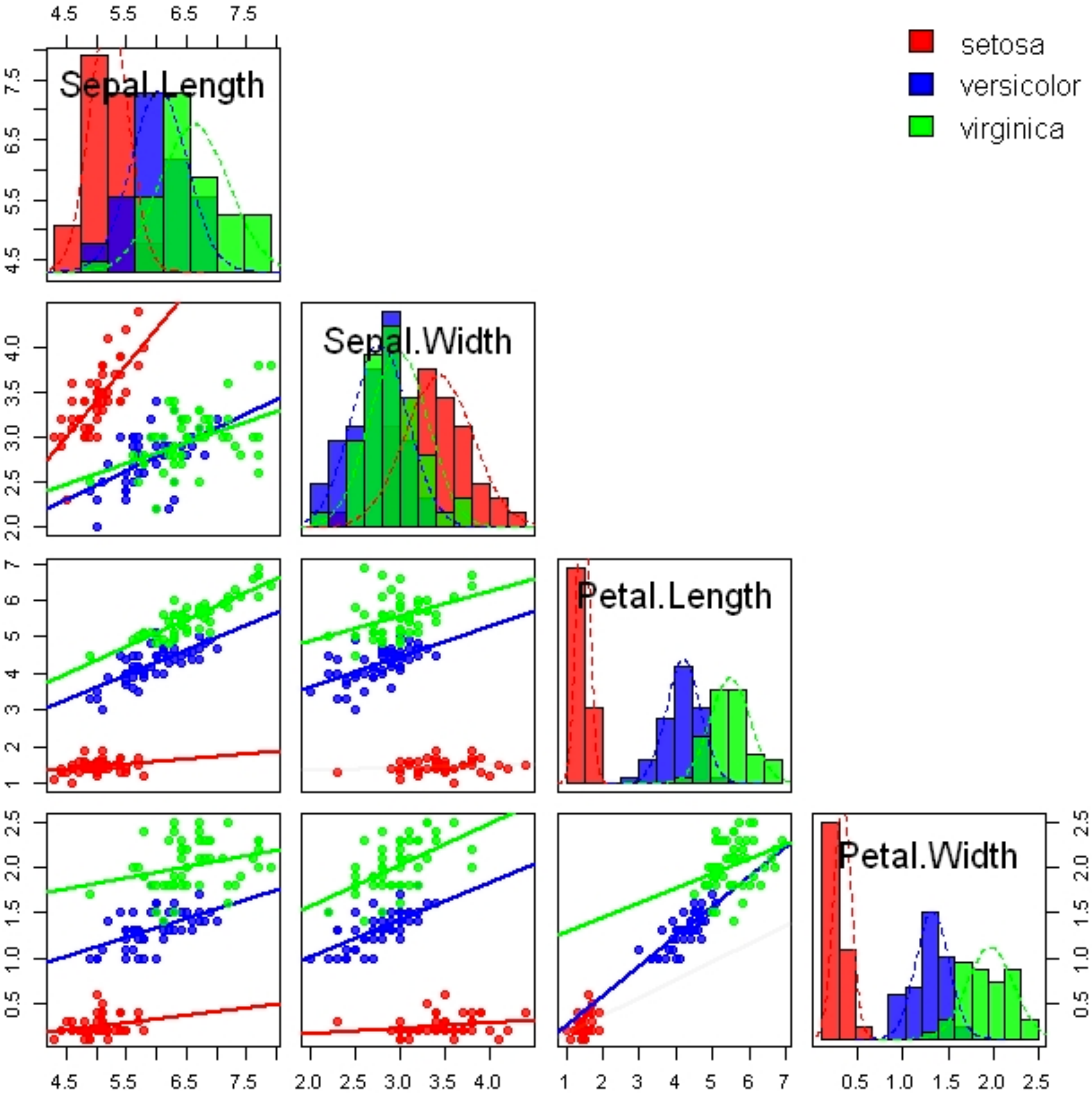
Univariate graphical EDA

- QQ plot:





Scatter matrix

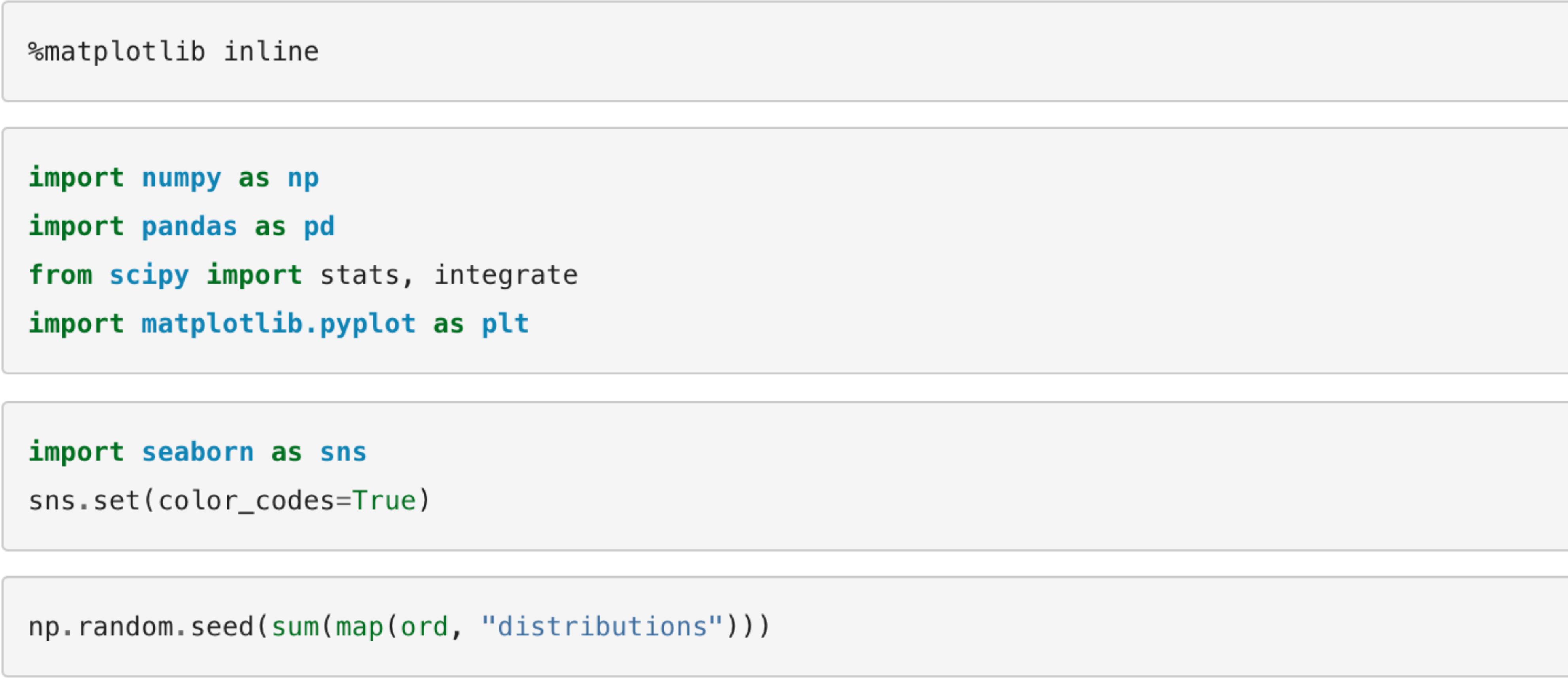


Visualization tools for python

- Seaborn
- Matplotlib
- Bokeh

Seaborn

- Visualizing distributions



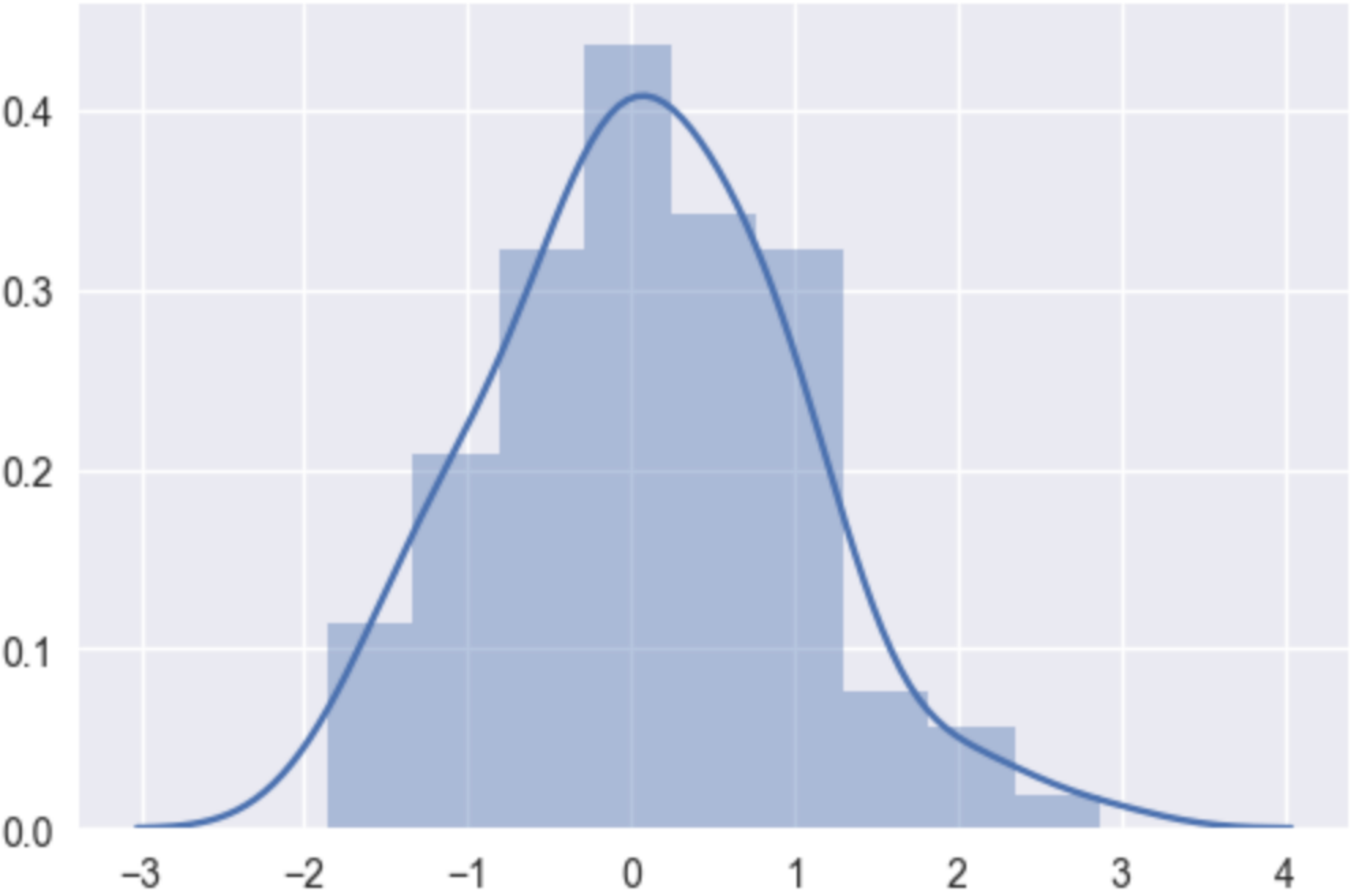
```
%matplotlib inline
```

```
import numpy as np
import pandas as pd
from scipy import stats, integrate
import matplotlib.pyplot as plt
```

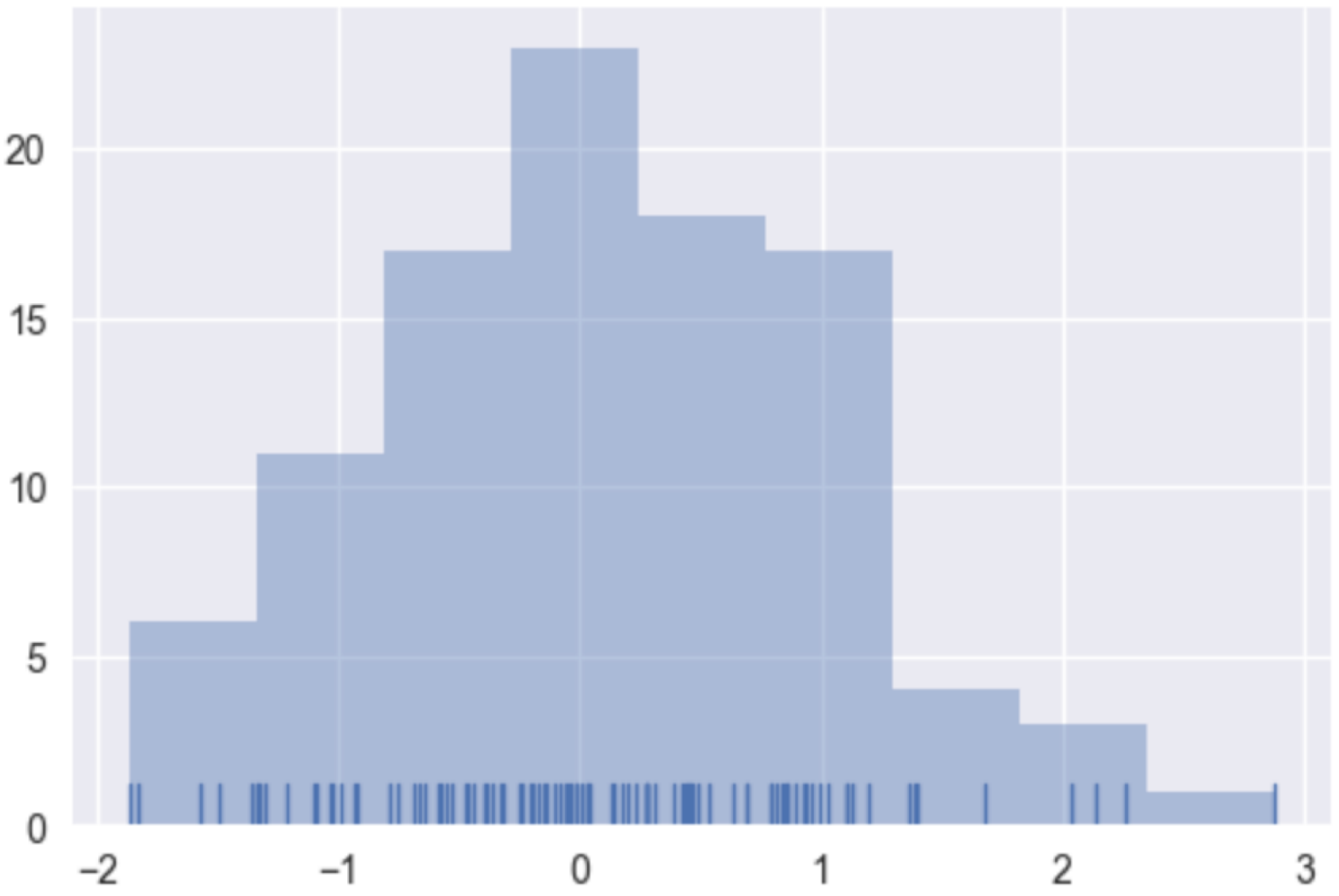
```
import seaborn as sns
sns.set(color_codes=True)
```

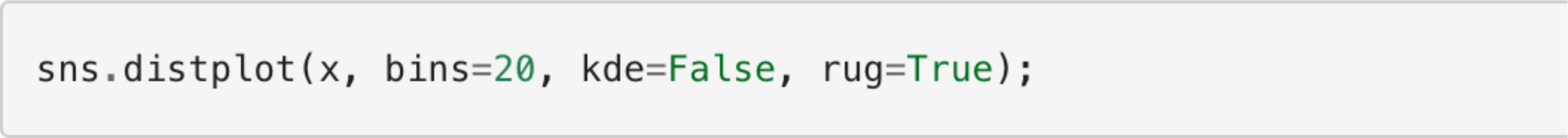
```
np.random.seed(sum(map(ord, "distributions")))
```

```
x = np.random.normal(size=100)
sns.distplot(x);
```

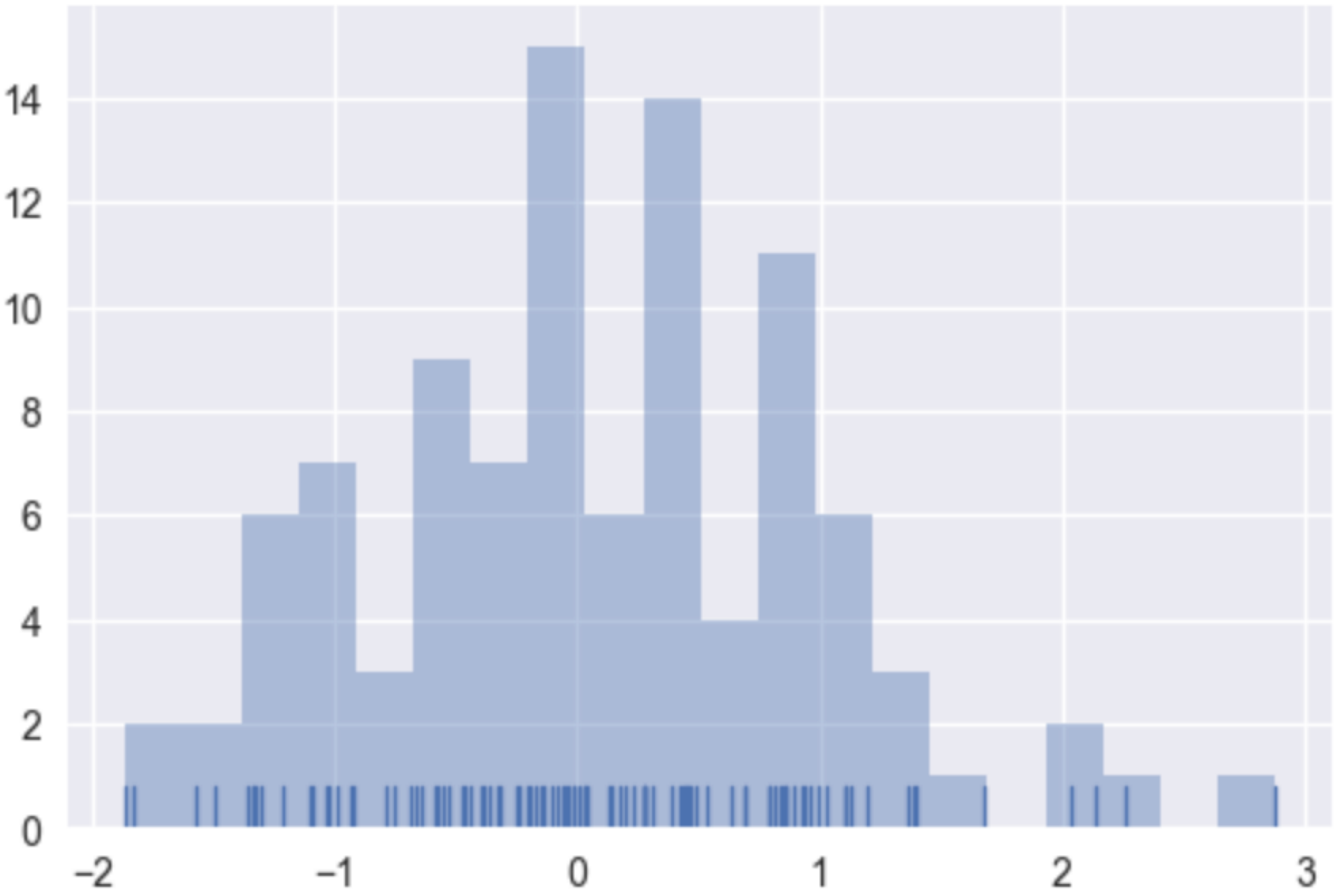


```
sns.distplot(x, kde=False, rug=True);
```

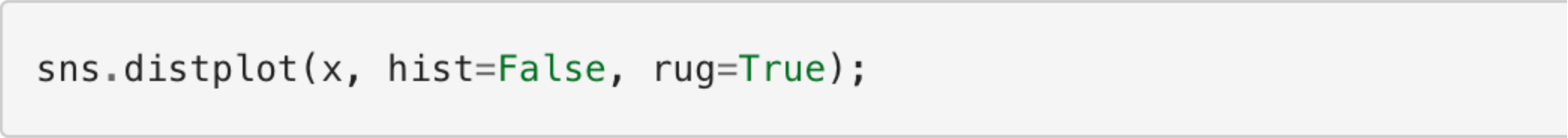




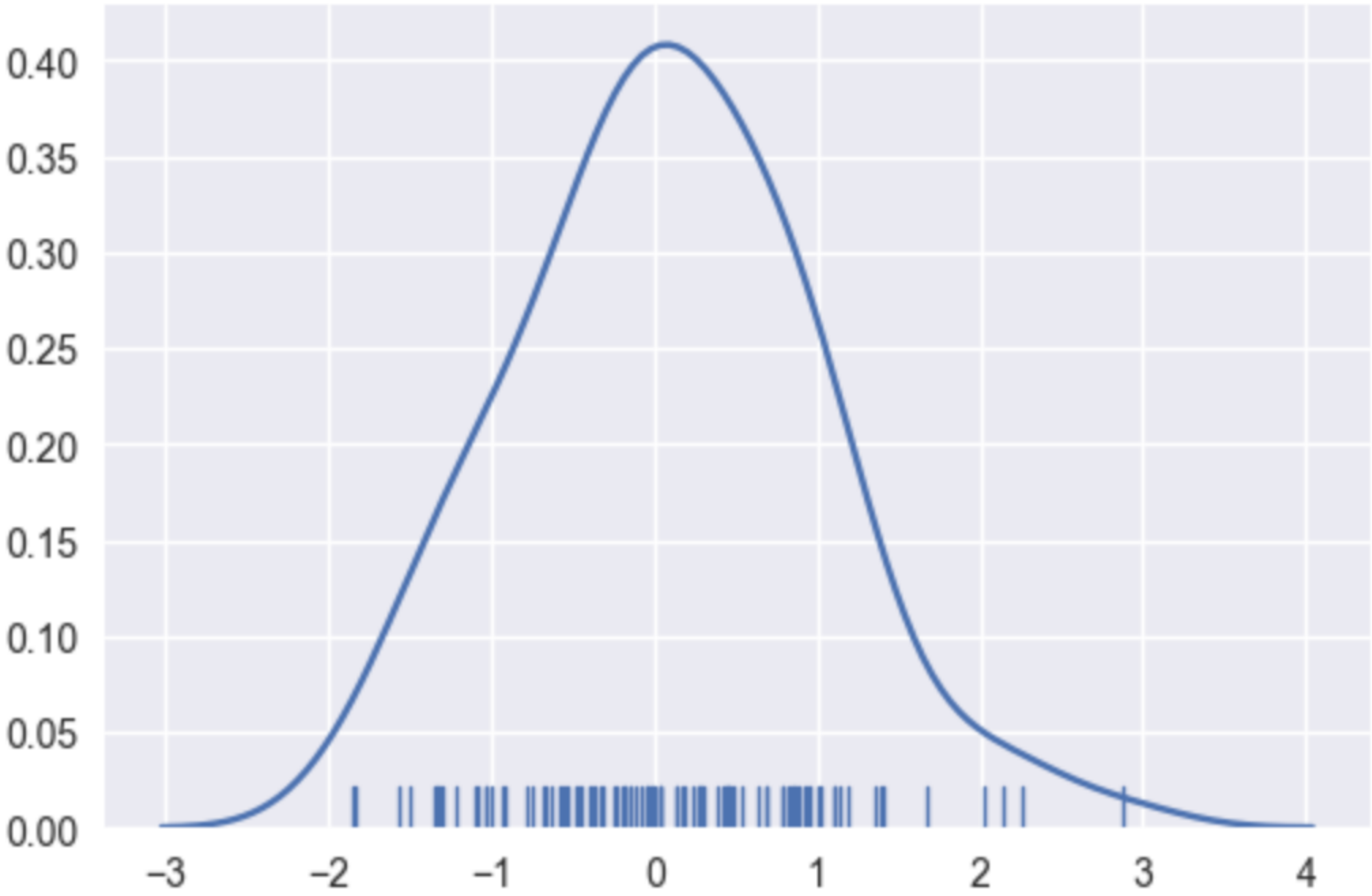
```
sns.distplot(x, bins=20, kde=False, rug=True);
```

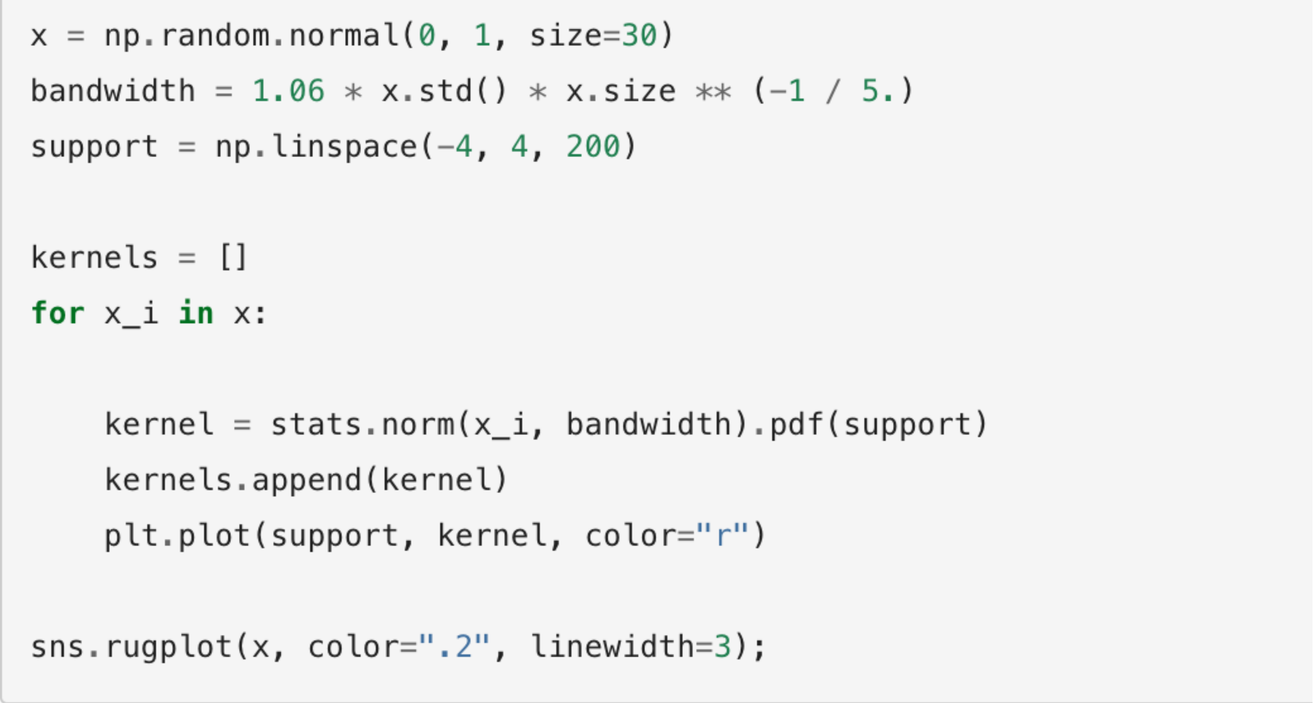


Kernel density estimation

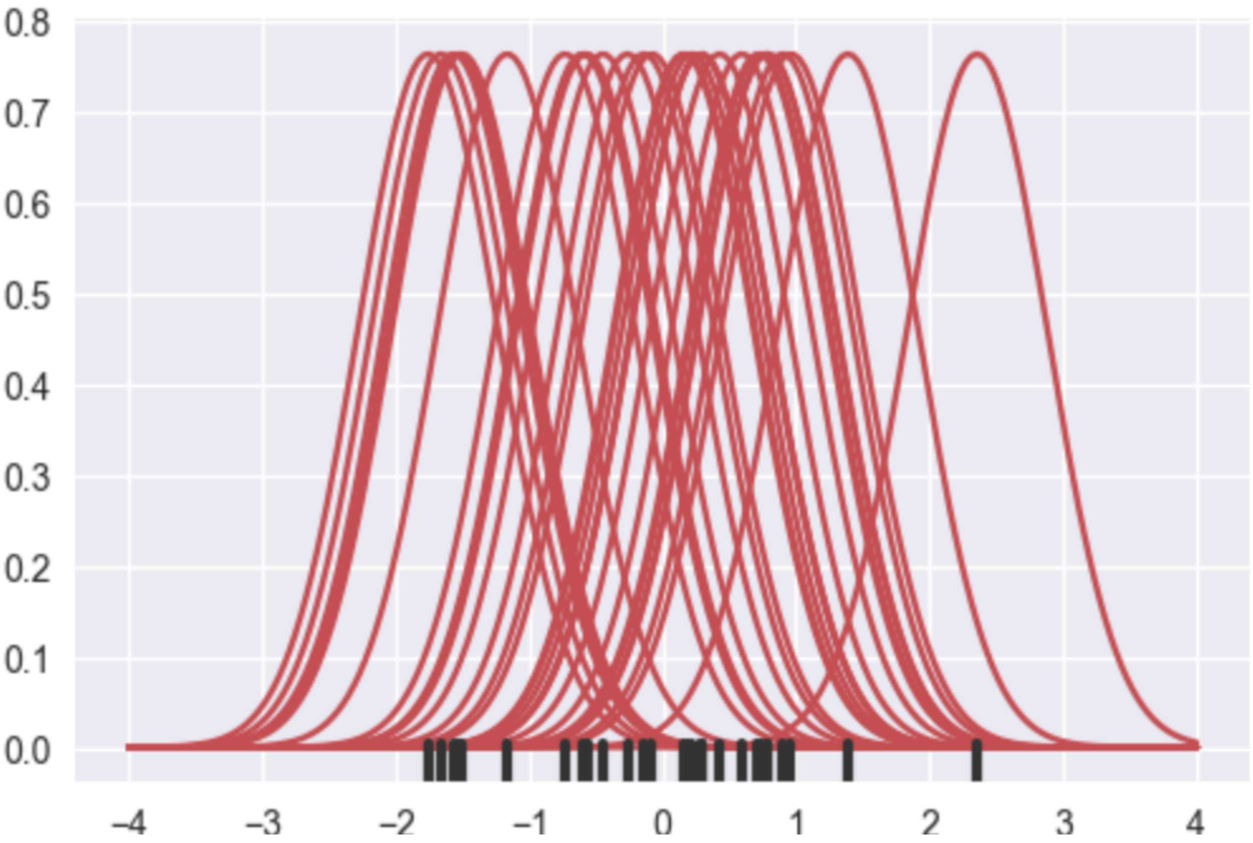


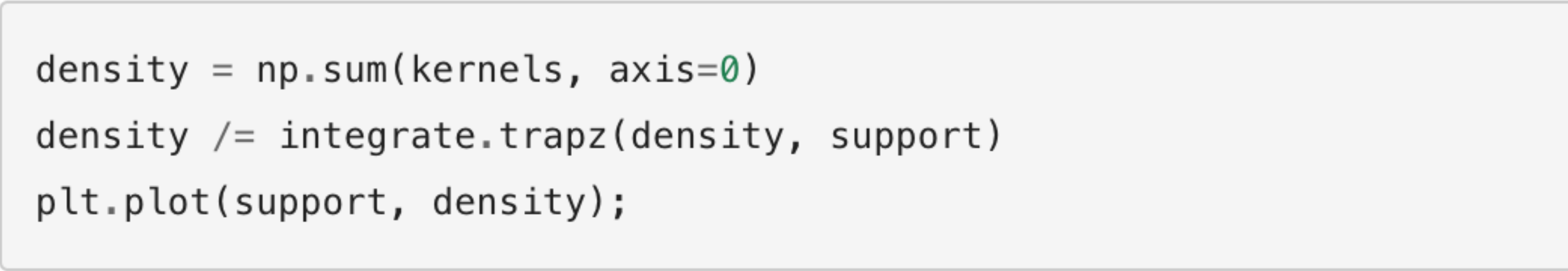
```
sns.distplot(x, hist=False, rug=True);
```



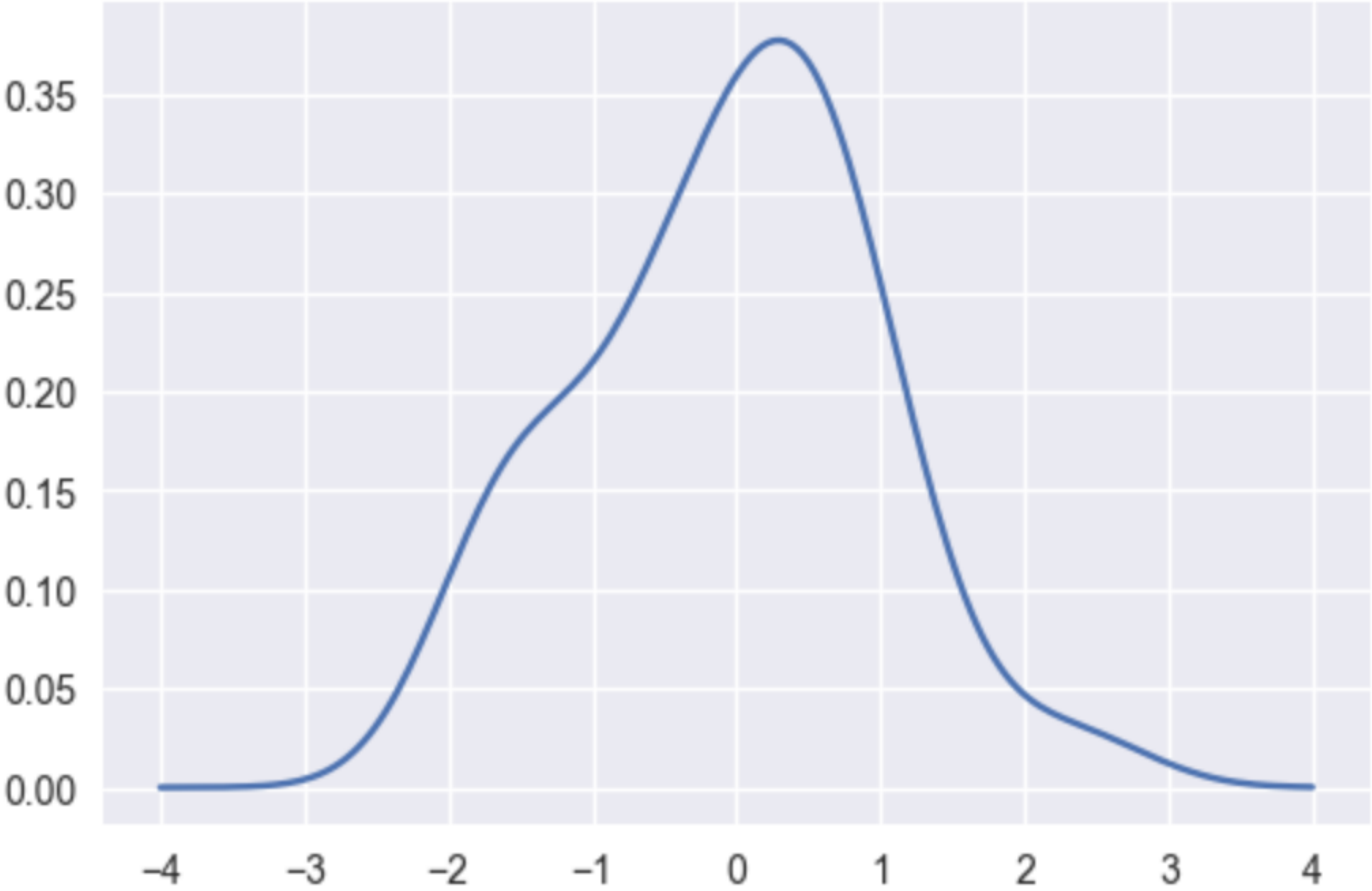


```
x = np.random.normal(0, 1, size=30)
bandwidth = 1.06 * x.std() * x.size ** (-1 / 5.)
support = np.linspace(-4, 4, 200)

kernels = []
for x_i in x:

    kernel = stats.norm(x_i, bandwidth).pdf(support)
    kernels.append(kernel)
    plt.plot(support, kernel, color="r")

sns.rugplot(x, color=".2", linewidth=3);
```

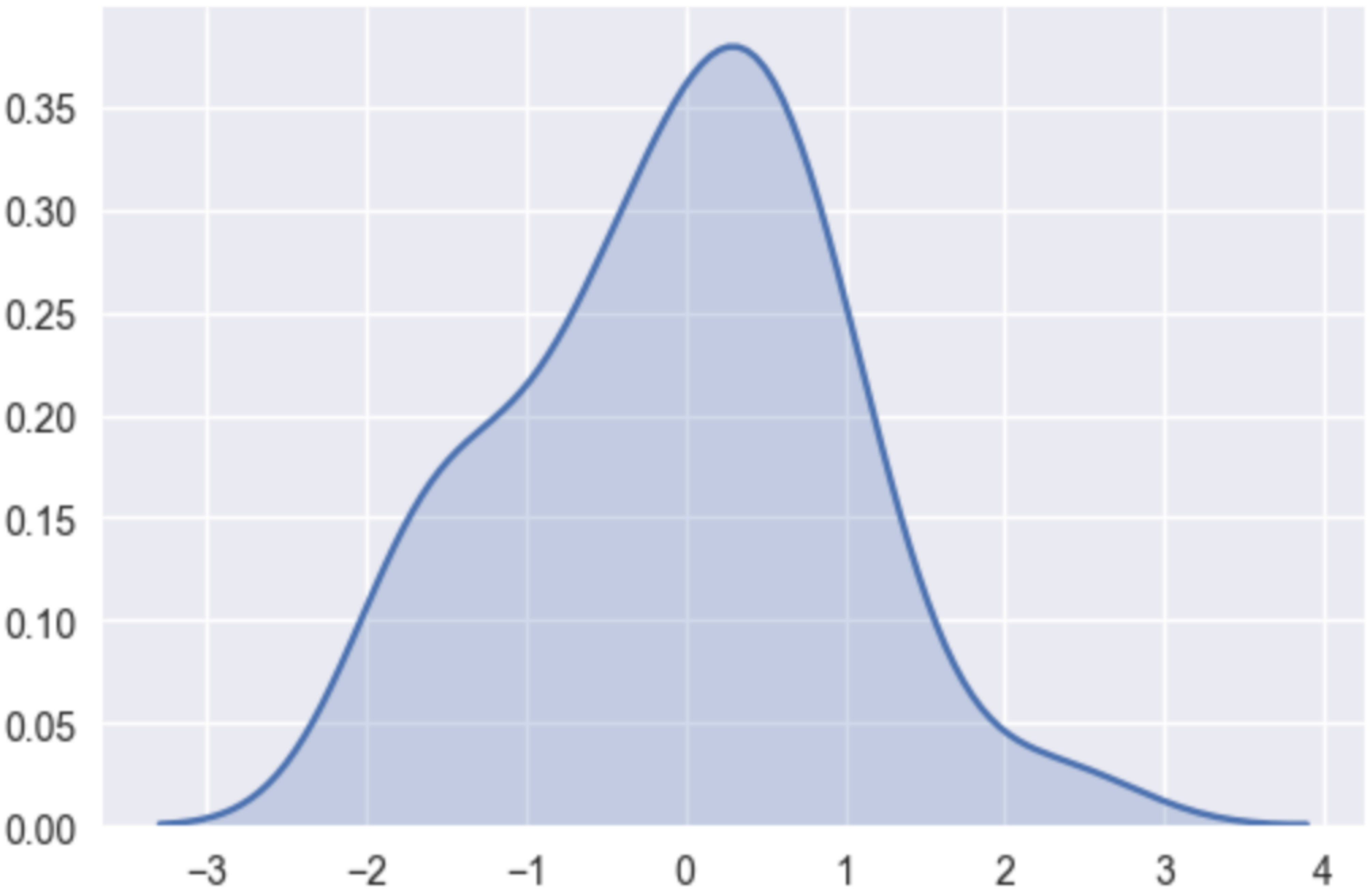


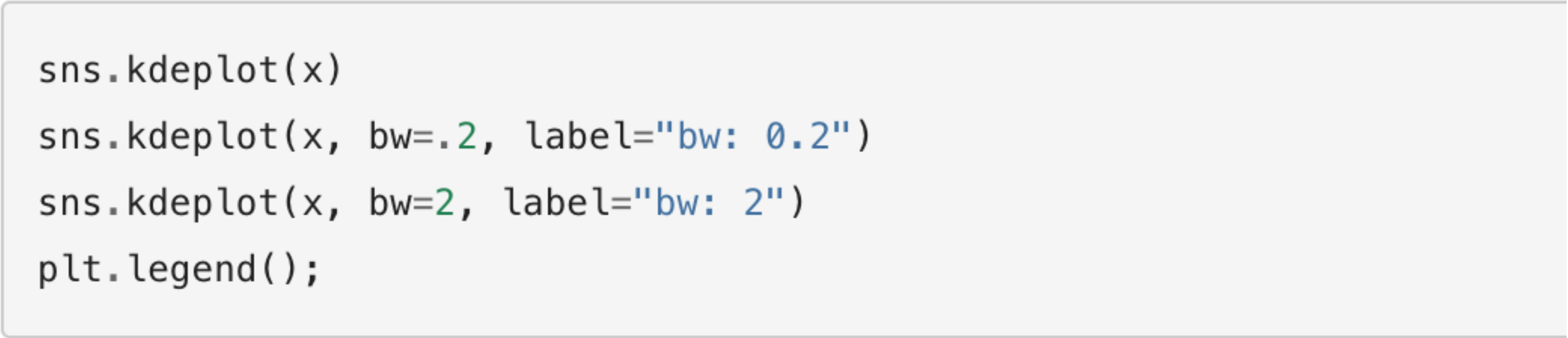


```
density = np.sum(kernels, axis=0)
density /= integrate.trapz(density, support)
plt.plot(support, density);
```

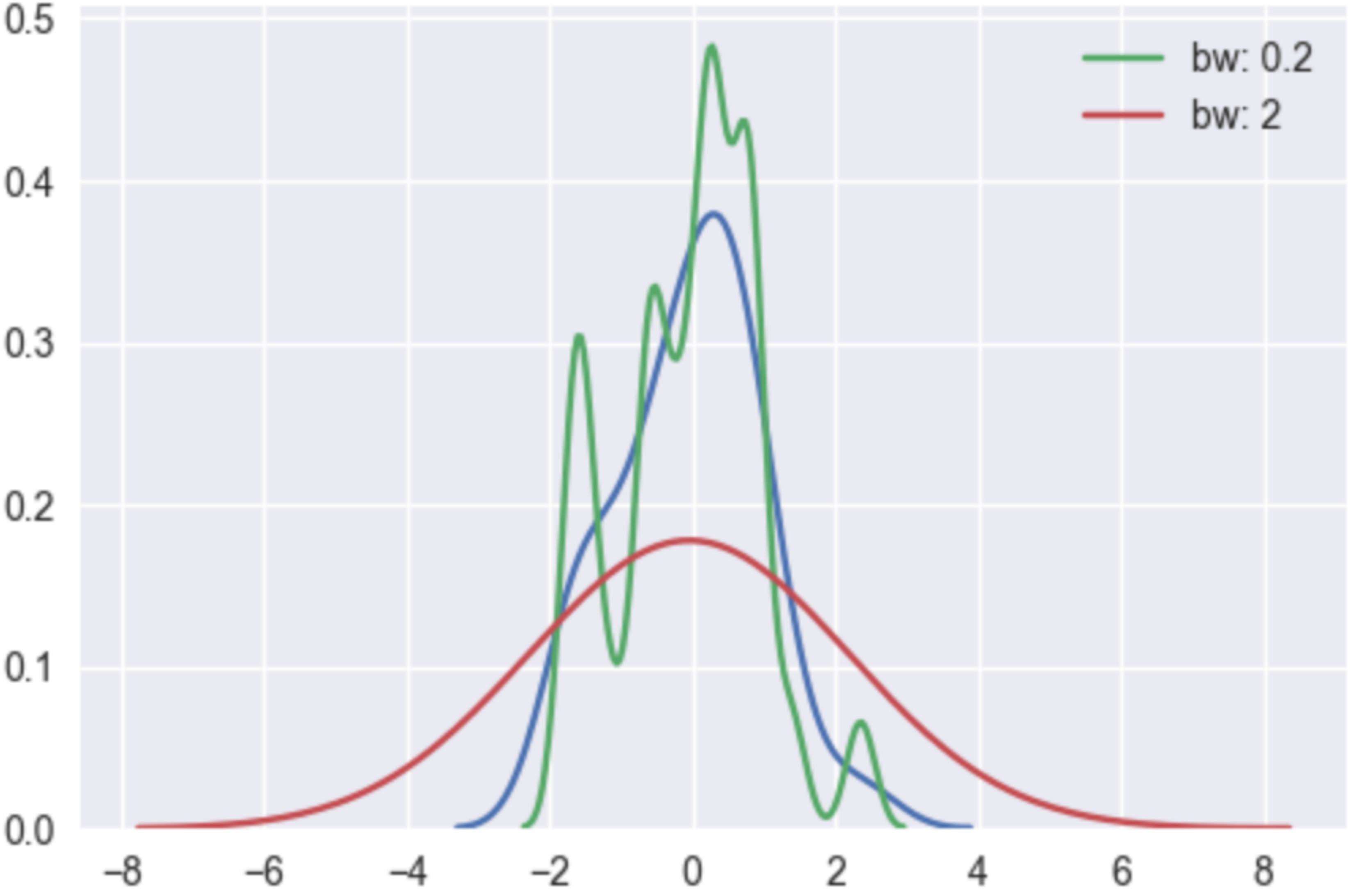


```
sns.kdeplot(x, shade=True);
```

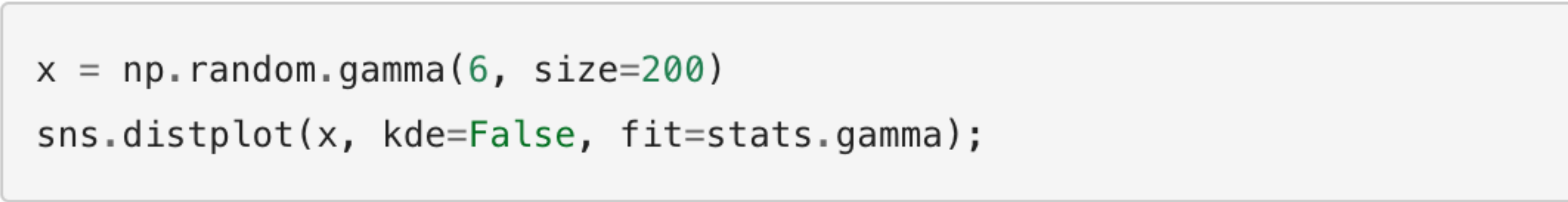




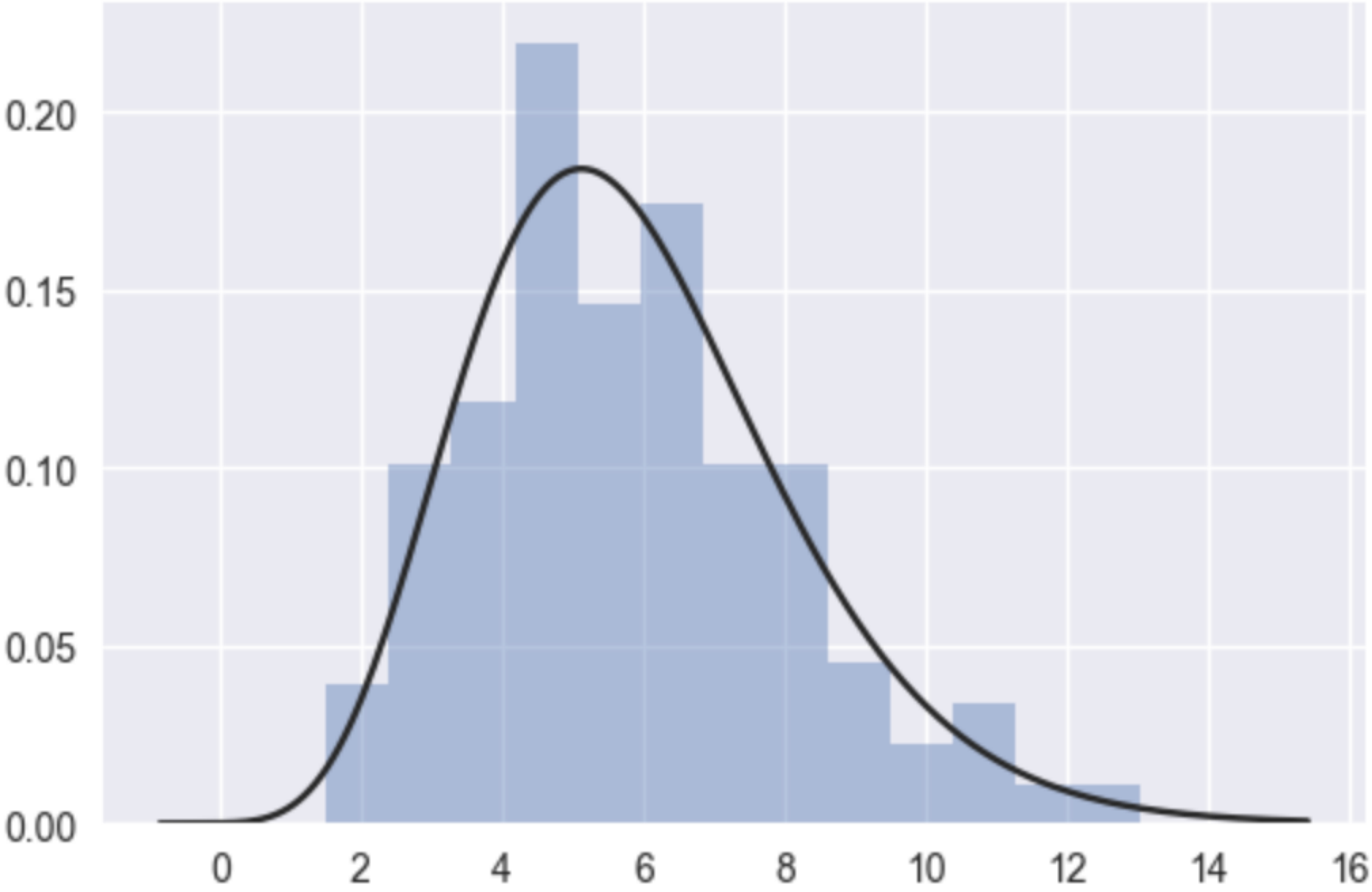
```
sns.kdeplot(x)
sns.kdeplot(x, bw=.2, label="bw: 0.2")
sns.kdeplot(x, bw=2, label="bw: 2")
plt.legend();
```



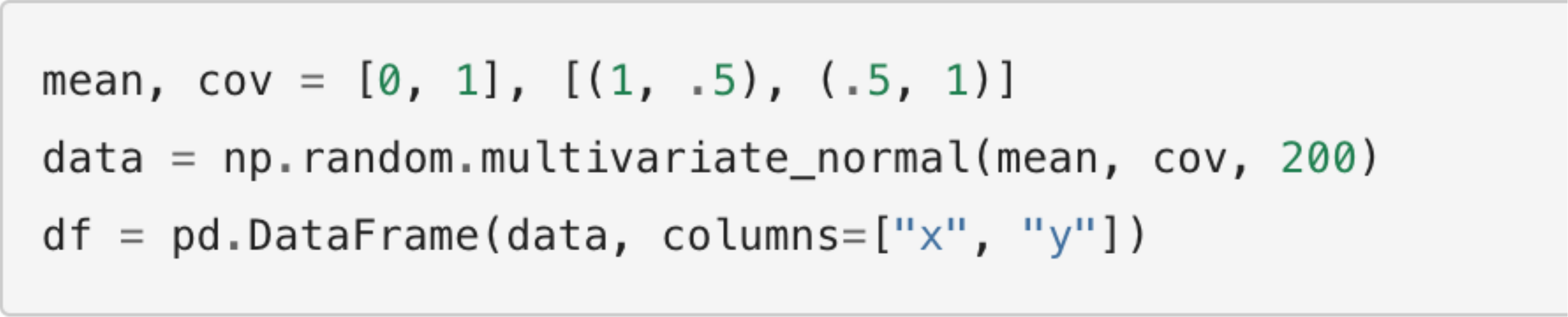
Fitting parametric distributions



```
x = np.random.gamma(6, size=200)
sns.distplot(x, kde=False, fit=stats.gamma);
```

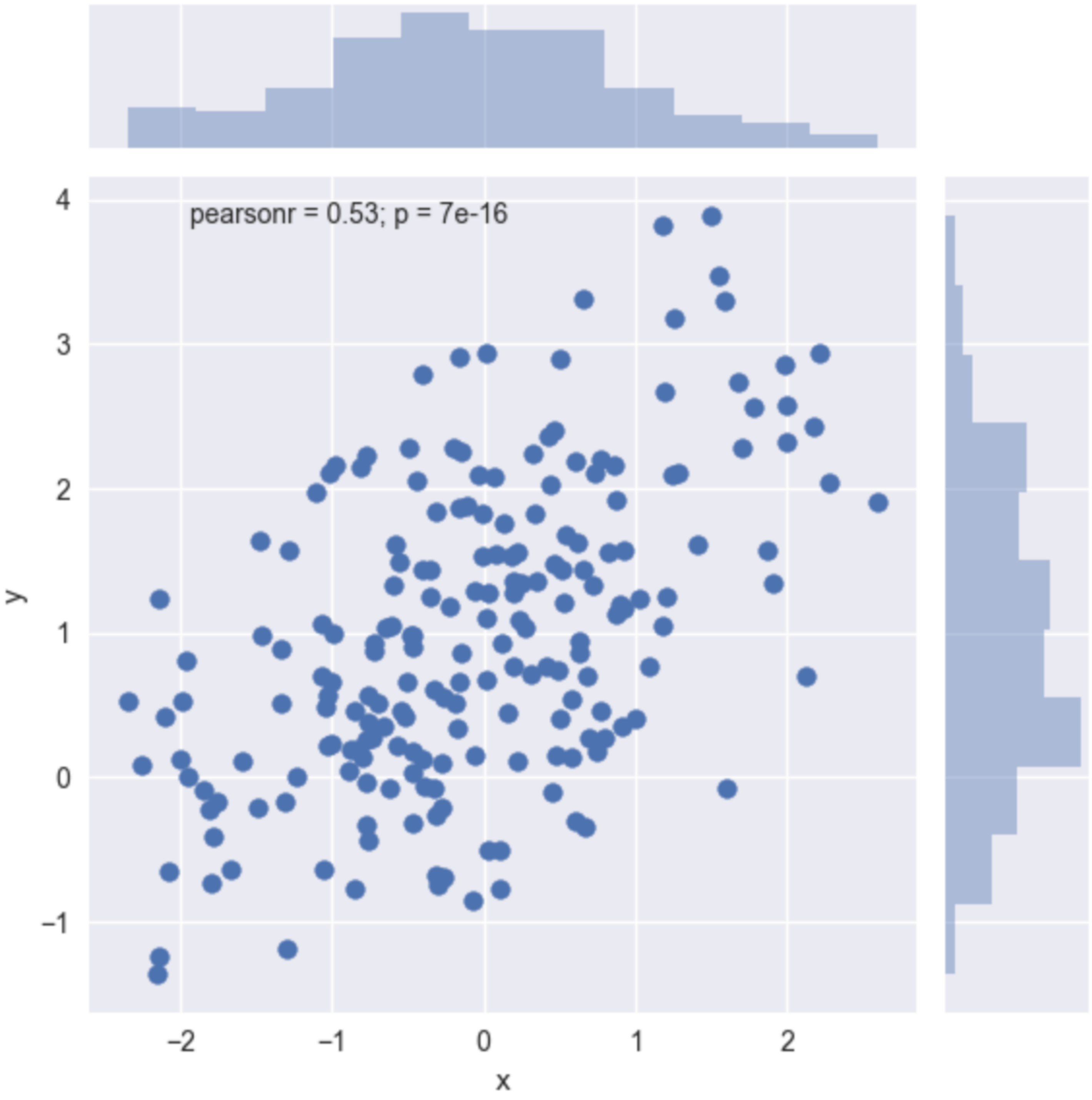


Plotting bivariate distributions

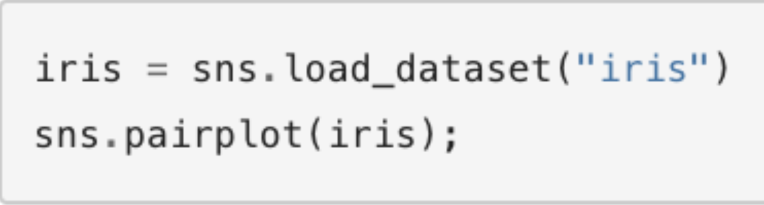


```
mean, cov = [0, 1], [(1, .5), (.5, 1)]
data = np.random.multivariate_normal(mean, cov, 200)
df = pd.DataFrame(data, columns=["x", "y"])
```

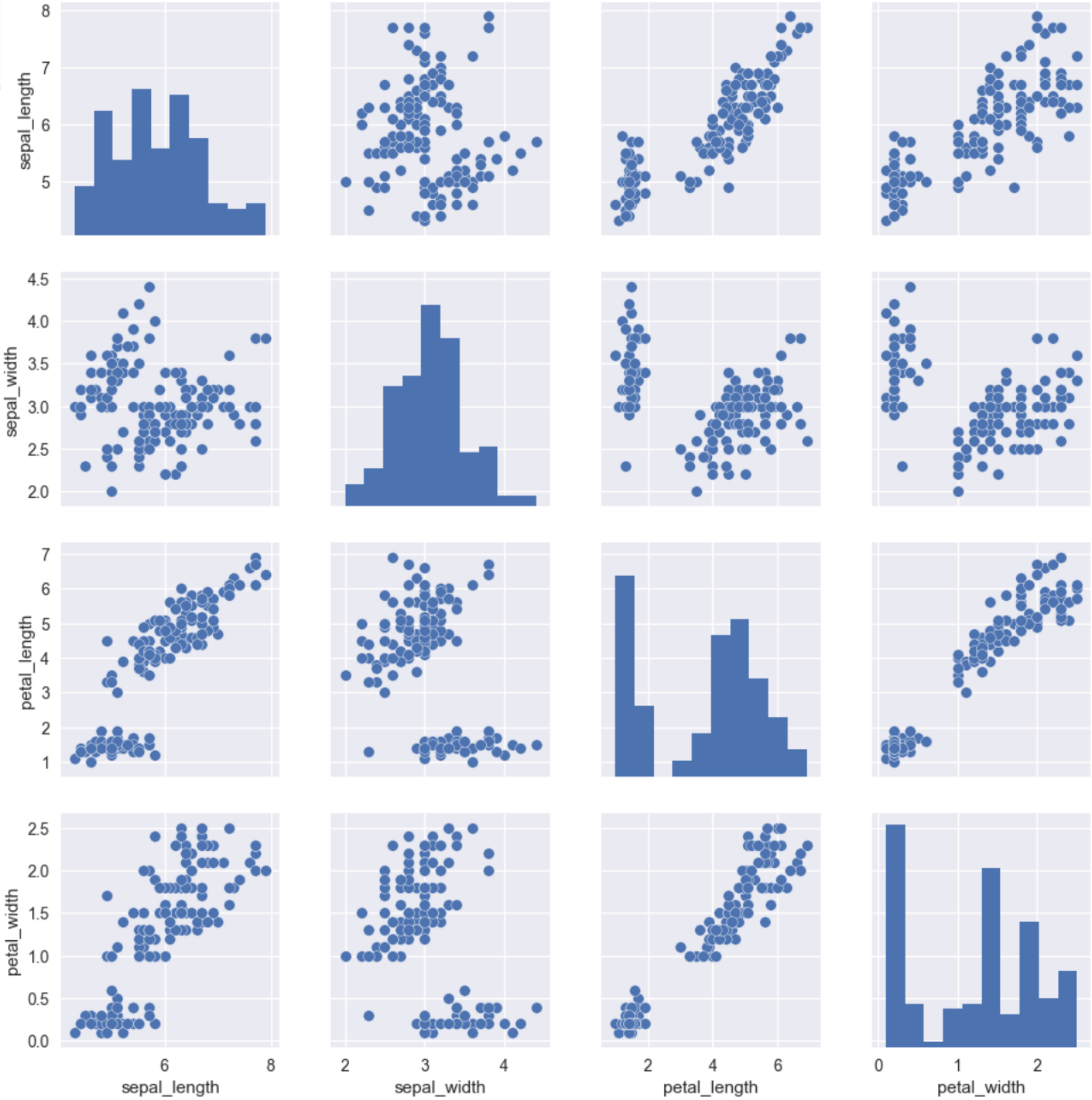
```
sns.jointplot(x="x", y="y", data=df);
```



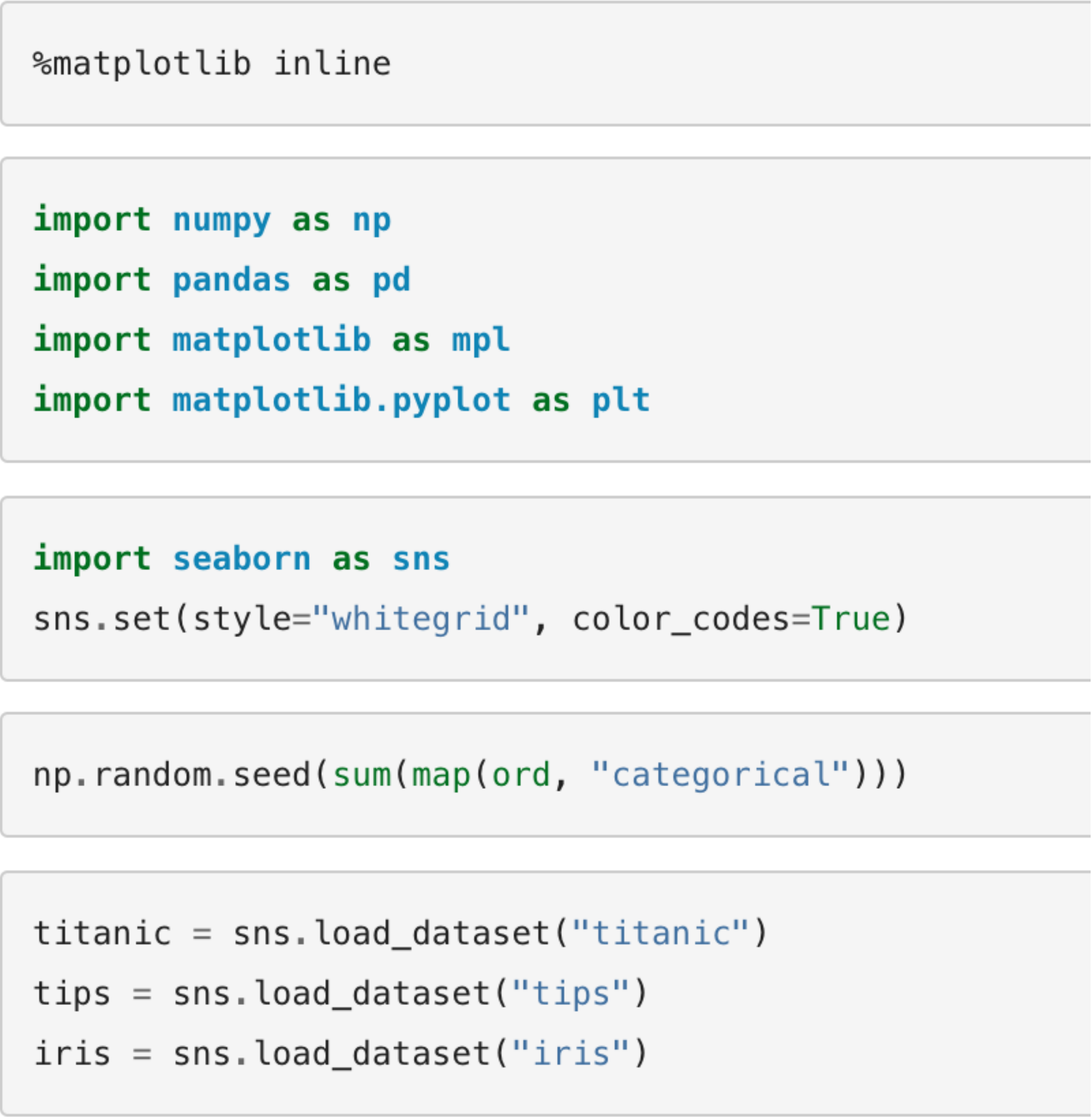
Visualizing pairwise relationships in a dataset



```
iris = sns.load_dataset("iris")
sns.pairplot(iris);
```



Visualizing categorical data



```
%matplotlib inline
```

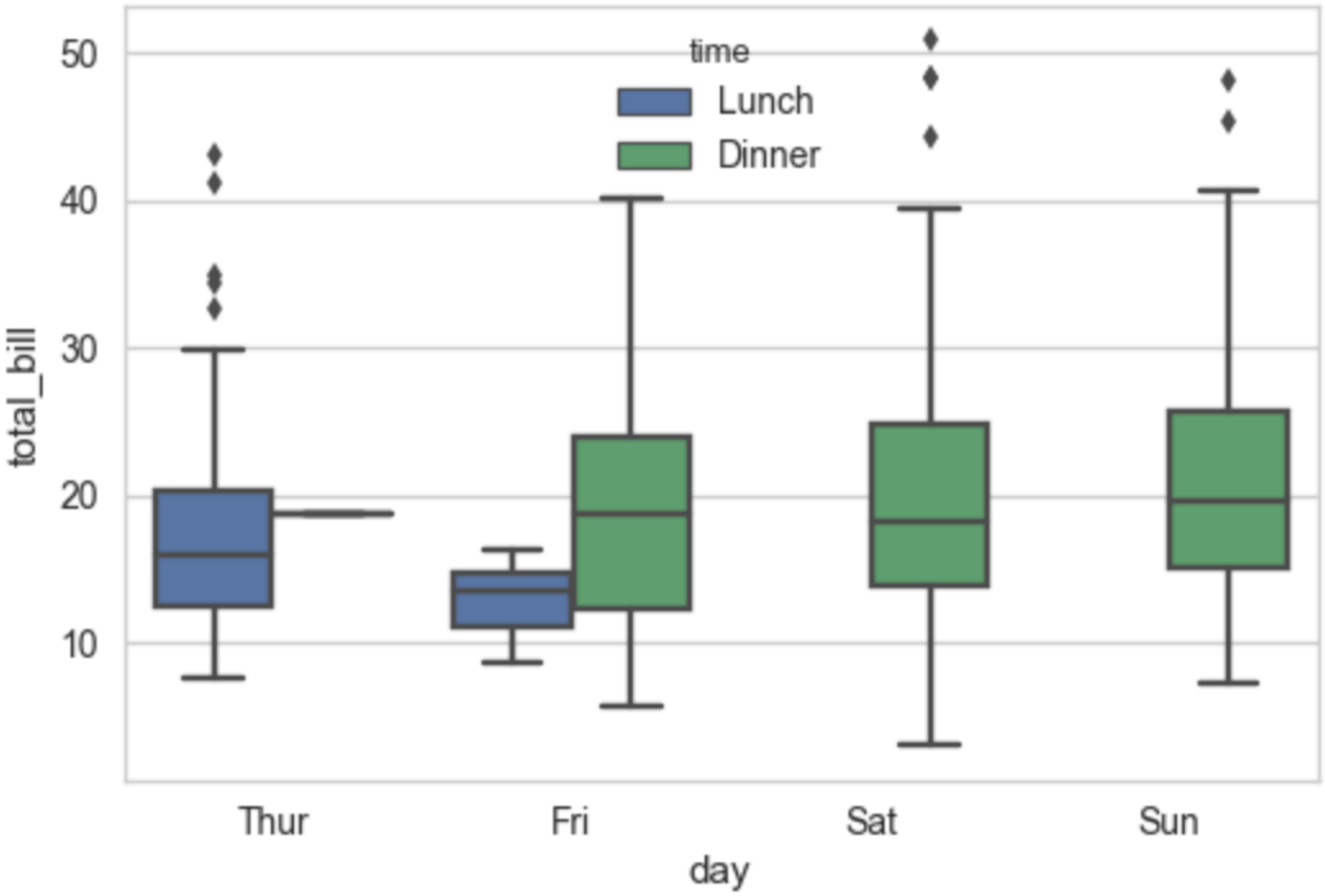
```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
```

```
import seaborn as sns
sns.set(style="whitegrid", color_codes=True)
```

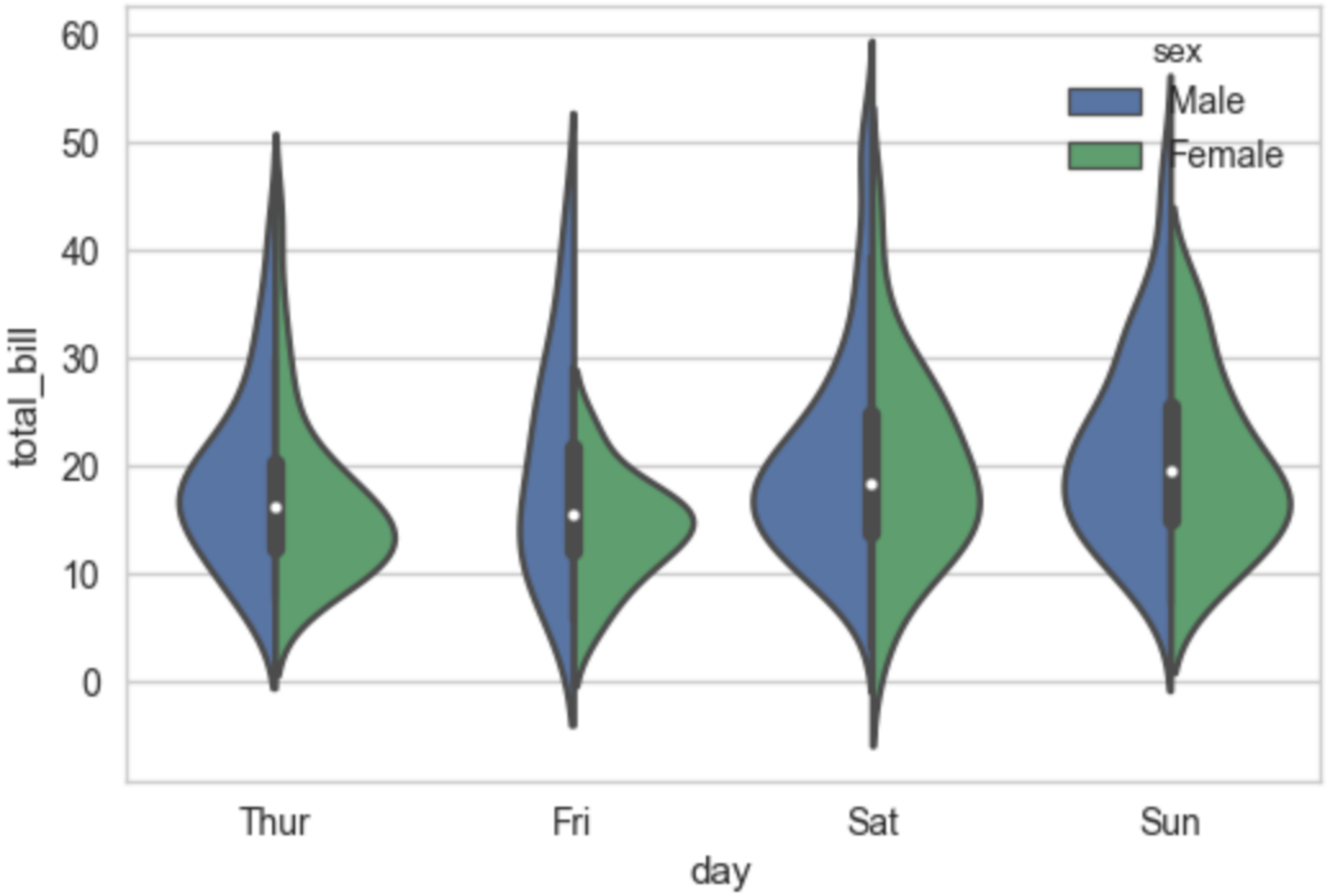
```
np.random.seed(sum(map(ord, "categorical")))
```

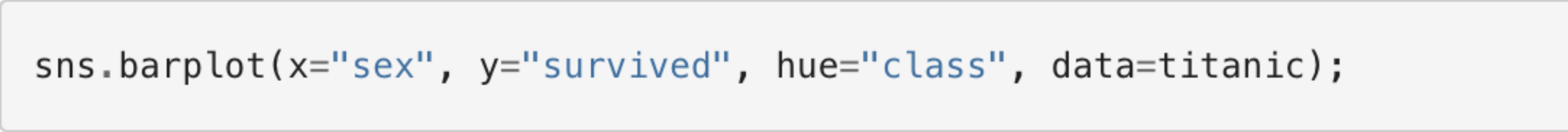
```
titanic = sns.load_dataset("titanic")
tips = sns.load_dataset("tips")
iris = sns.load_dataset("iris")
```

```
sns.boxplot(x="day", y="total_bill", hue="time", data=tips);
```

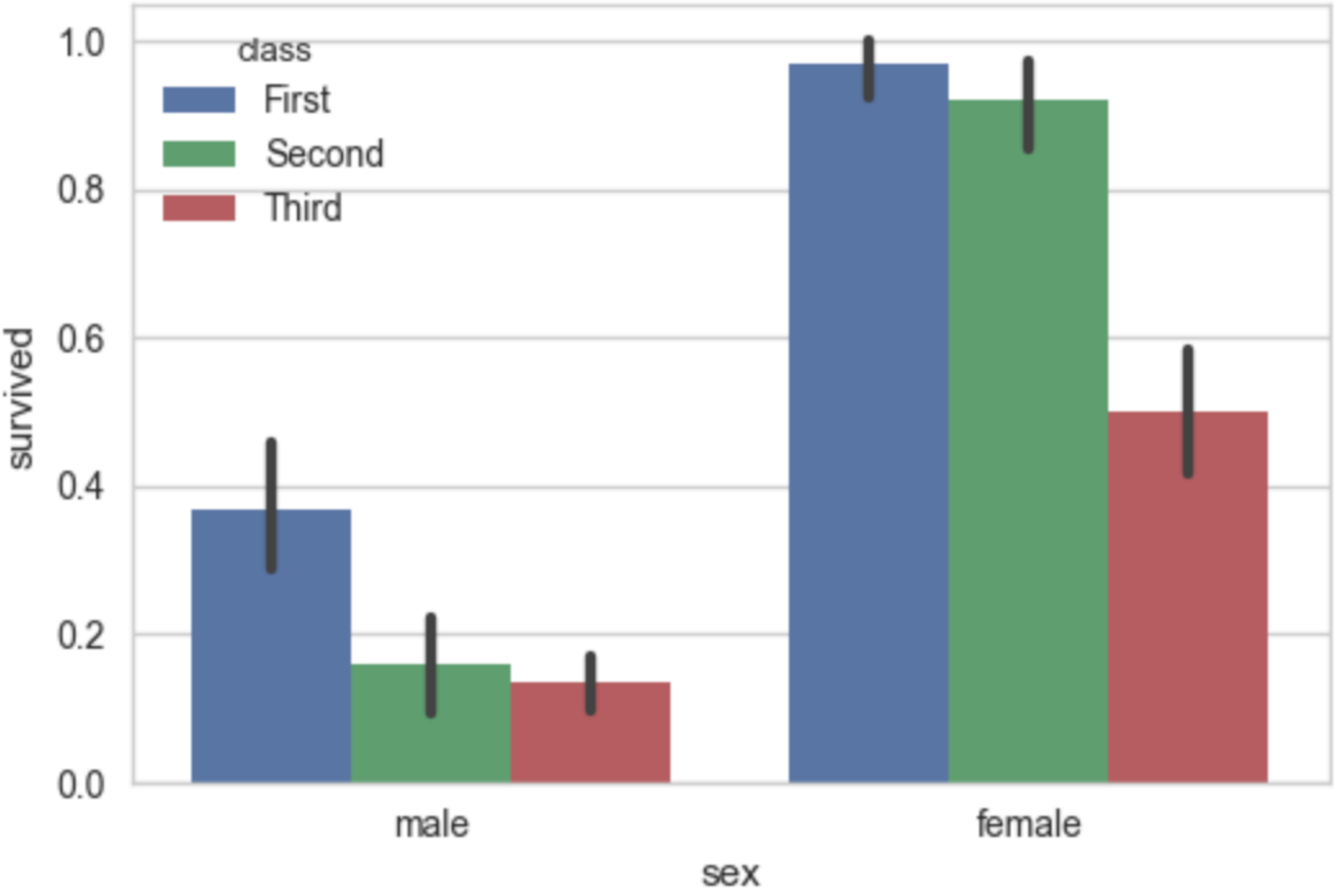


```
sns.violinplot(x="day", y="total_bill", hue="sex", data=tips, split=True);
```

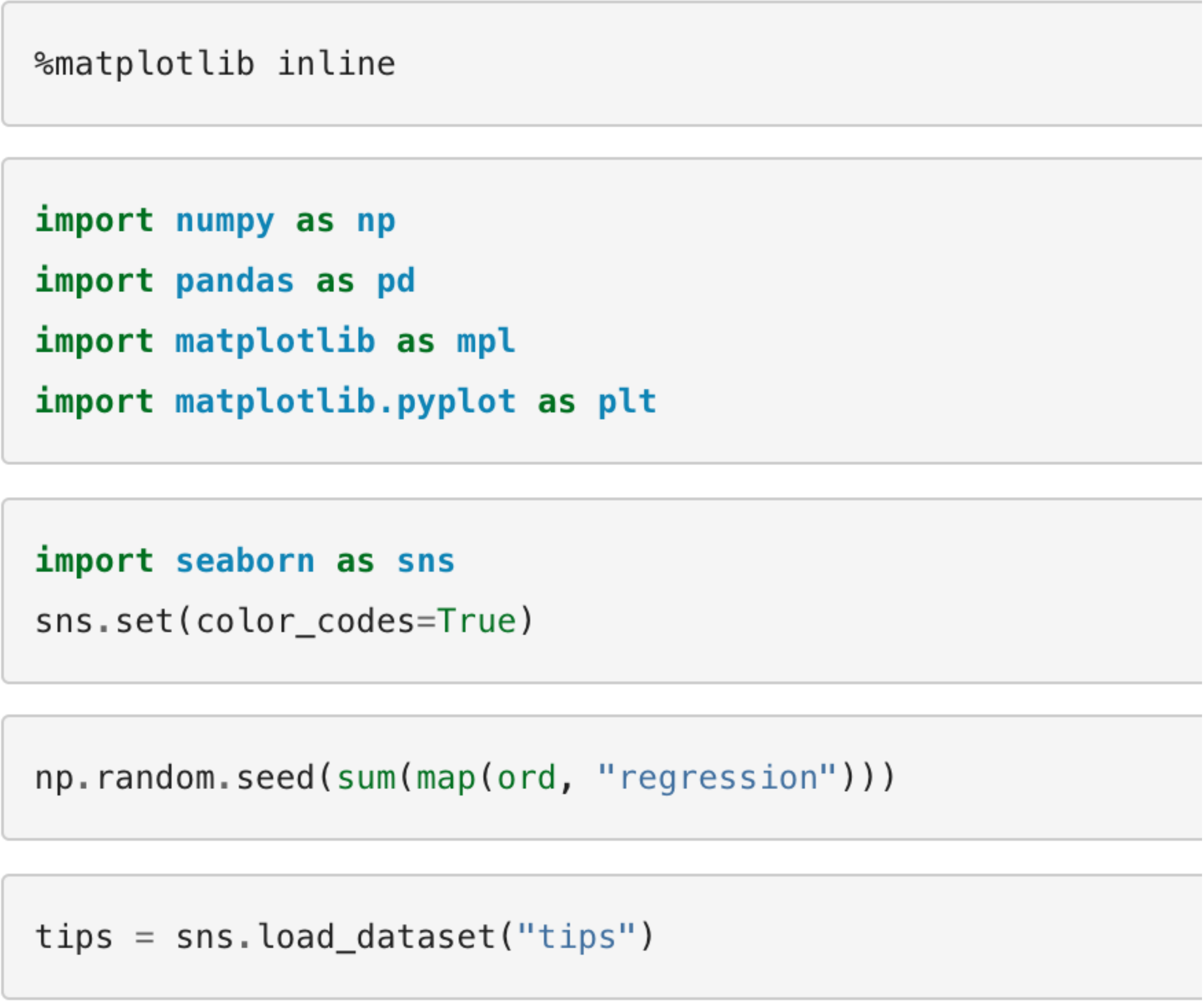




```
sns.barplot(x="sex", y="survived", hue="class", data=titanic);
```



Visualizing linear relationships



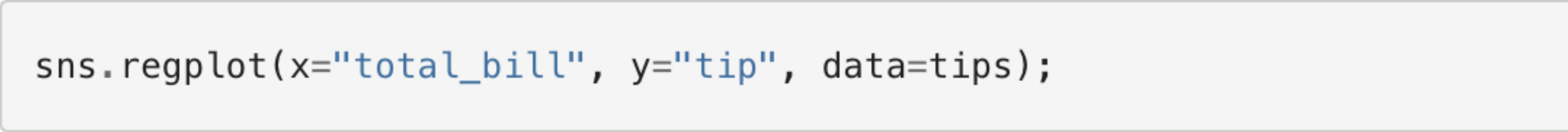
```
%matplotlib inline
```

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
```

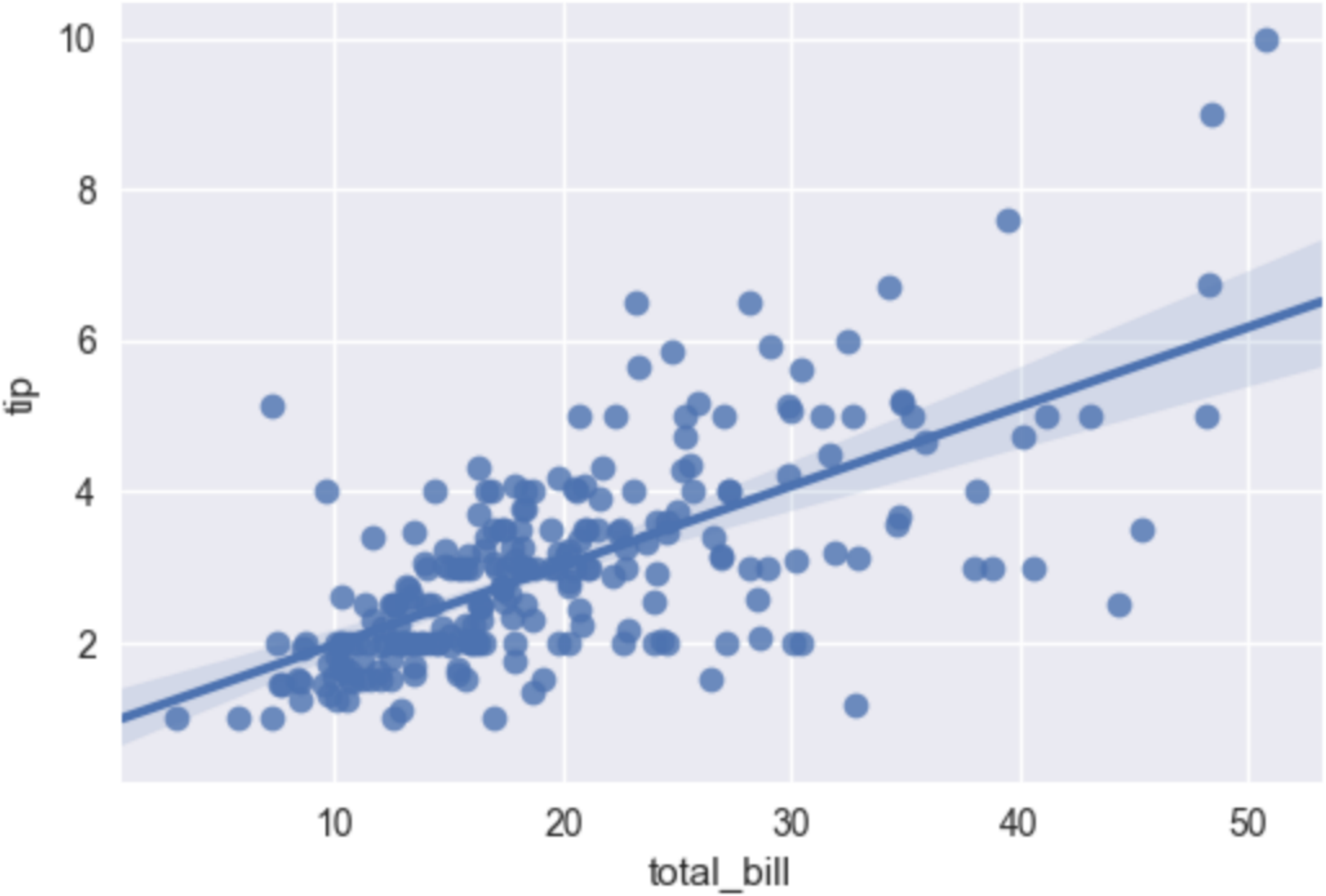
```
import seaborn as sns
sns.set(color_codes=True)
```

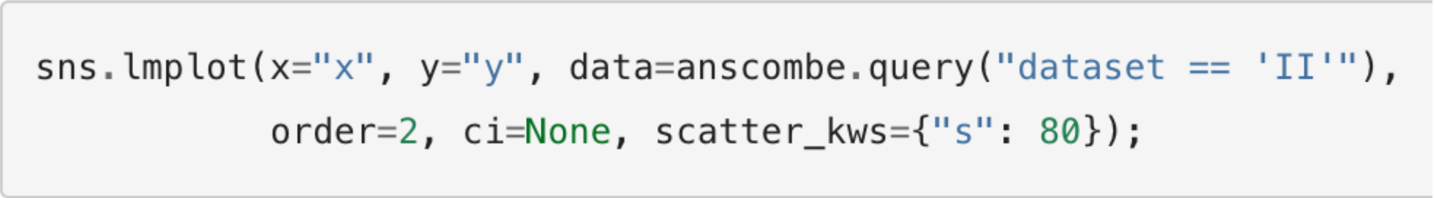
```
np.random.seed(sum(map(ord, "regression")))
```

```
tips = sns.load_dataset("tips")
```

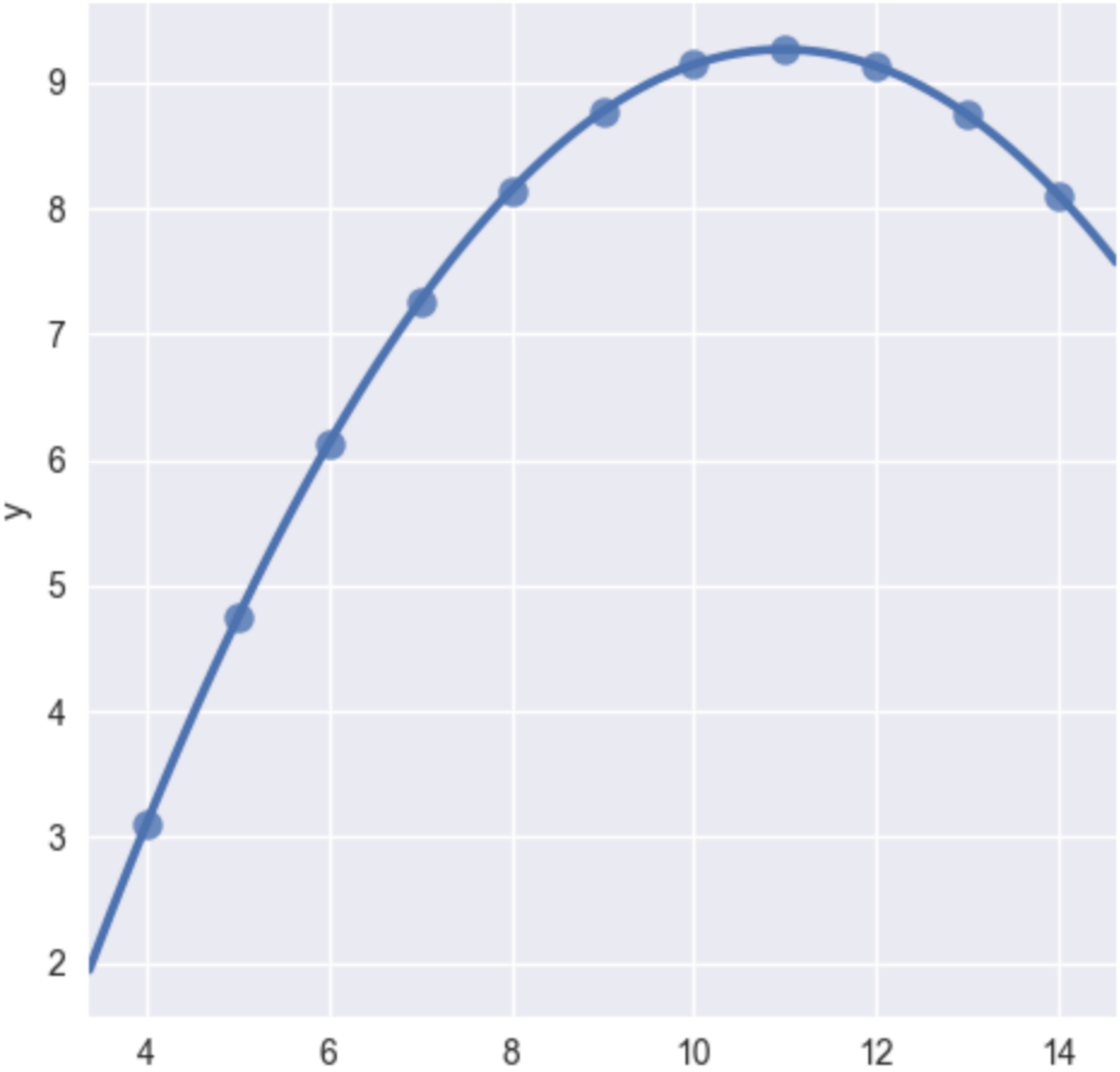


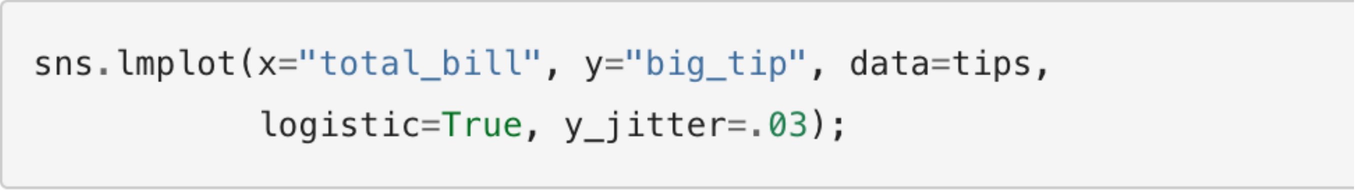
```
sns.regplot(x="total_bill", y="tip", data=tips);
```



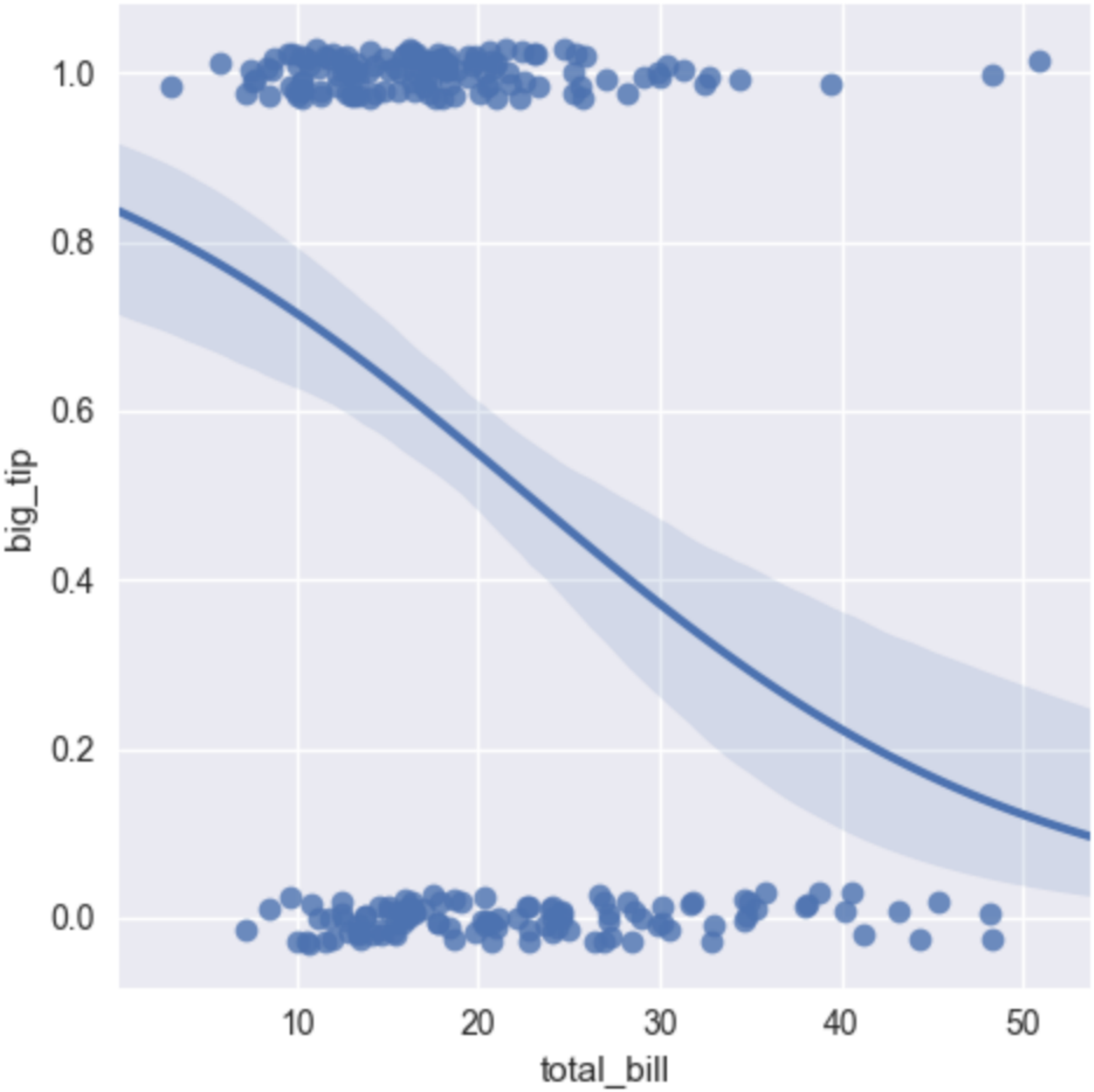


```
sns.lmplot(x="x", y="y", data=anscombe.query("dataset == 'II'"),
           order=2, ci=None, scatter_kws={"s": 80});
```

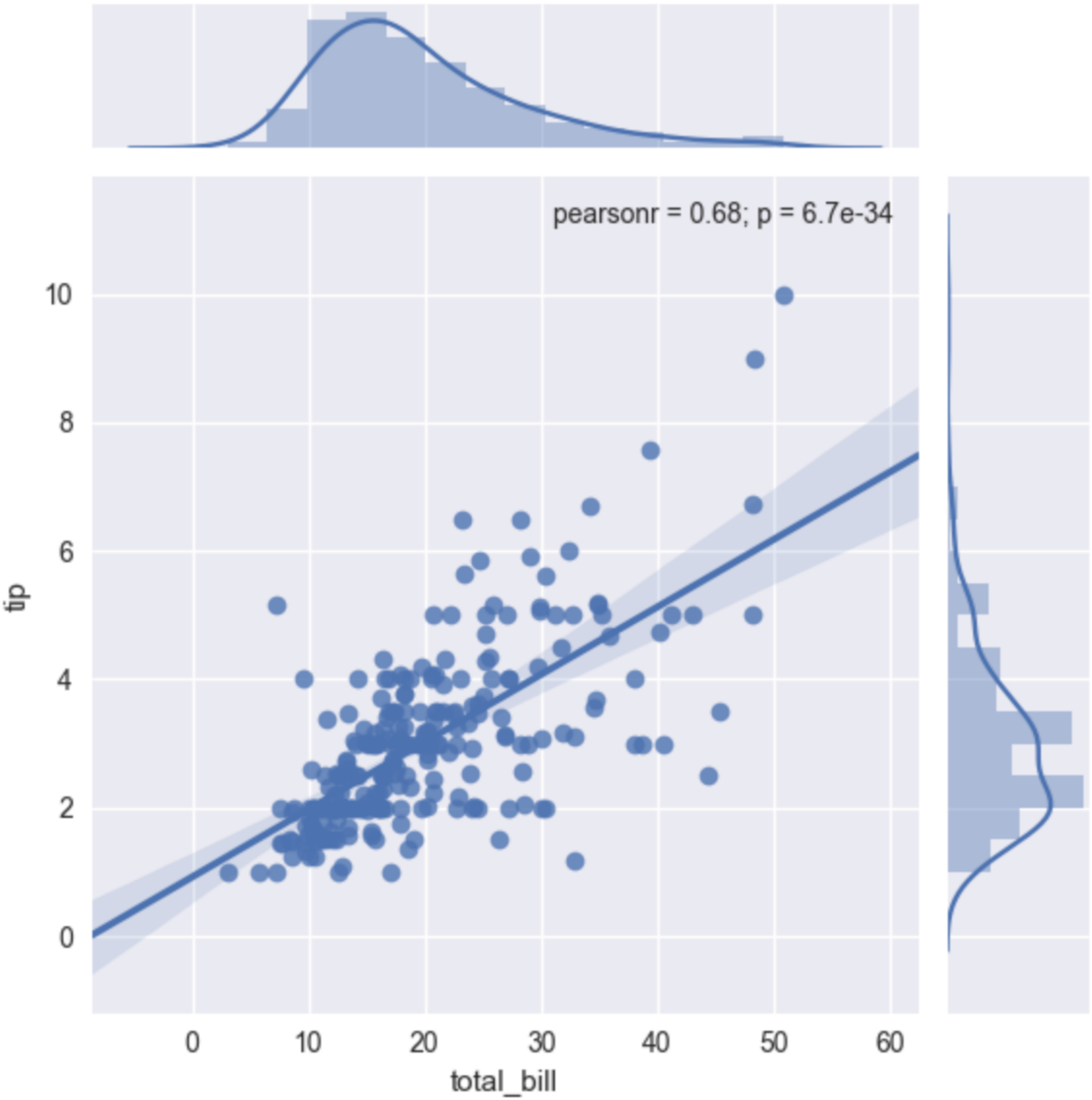




```
sns.lmplot(x="total_bill", y="big_tip", data=tips,
           logistic=True, y_jitter=.03);
```



```
sns.jointplot(x="total_bill", y="tip", data=tips, kind="reg");
```



- **HW: Pick one data set, write notebook that downloads and cleans the data (for general purpose analyzing)**

- Netflix data
    - https://www.kaggle.com/netflix-inc/netflix-prize-data/data
- Yahoo finance
    - https://pypi.python.org/pypi/yahoo-finance
- IMF data
    - https://briandew.wordpress.com/2016/05/01/machine-reading-imf-data-data-retrieval-with-python/
- NYC open data
    - https://opendata.cityofnewyork.us/data/#datasetscategory
    - Examples:
        - http://blog.nycdatascience.com/student-works/r-shiny/noise-coming-case-study-nycs-311-noise-complaints/
        - http://blog.nycdatascience.com/student-works/new-york-city/