



2.8 Настройка графиков

Трёхмерные точки и линии

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

fig = plt.figure()
ax = plt.axes(projection="3d")

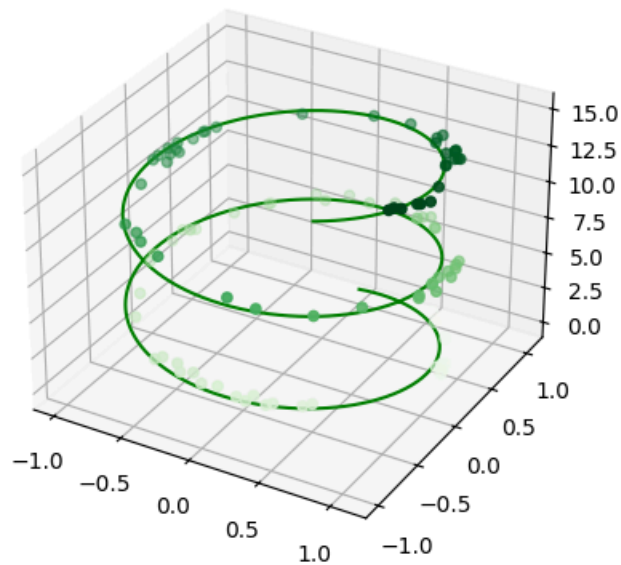
z1 = np.linspace(0, 15, 1000)
y1 = np.cos(z1)
x1 = np.sin(z1)

ax.plot3D(x1, y1, z1, "green")

z2 = 15 * np.random.random(100)
y2 = np.cos(z2) + 0.1 * np.random.random(100)
x2 = np.sin(z2) + 0.1 * np.random.random(100)

ax.scatter3D(x2, y2, z2, c=z2, cmap="Greens")

plt.savefig("img.png")
plt.show()
```



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

def f(x, y):
    return np.sin(np.sqrt(x ** 2 + y ** 2))

fig = plt.figure()
ax = plt.axes(projection="3d")

x = np.linspace(-6, 6, 30)
y = np.linspace(-6, 6, 30)

X, Y = np.meshgrid(x, y)
Z = f(X, Y)

ax.contour3D(X, Y, Z, 40, cmap="binary")

ax.set_xlabel("$X$")
ax.set_ylabel("$Y$")
```

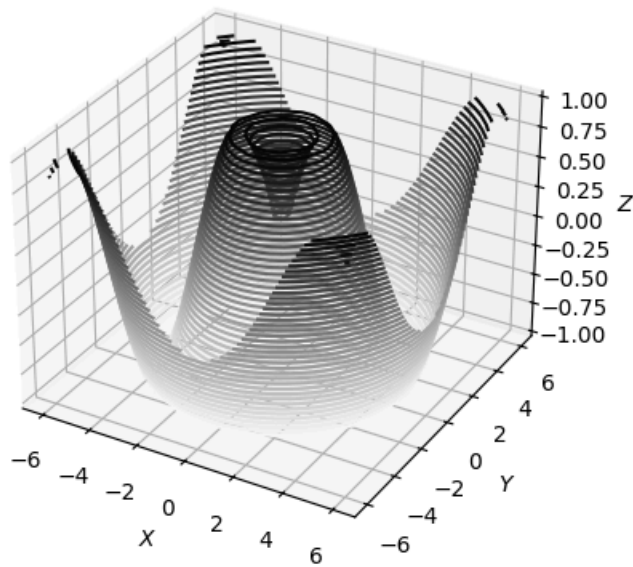
```

ax.set_zlabel("$Z$")

plt.savefig("img.png")

ax.view_init(45, 60)
plt.show()

```



Каркасные или поверхностные графики

```

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

fig = plt.figure()
ax = plt.axes(projection="3d")

def f(x, y):
    return np.sin(np.sqrt(x ** 2 + y ** 2))

```

```

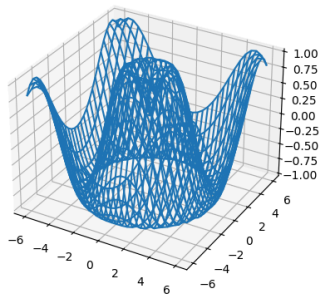
x = np.linspace(-6, 6, 30)
y = np.linspace(-6, 6, 30)

X, Y = np.meshgrid(x, y)
Z = f(X, Y)

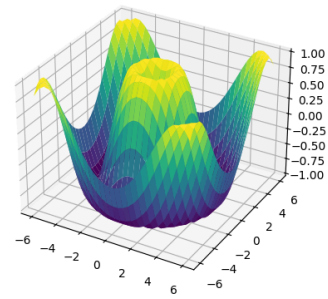
ax.plot_wireframe(X, Y, Z)
#ax.plot_surface(X, Y, Z, cmap="viridis")

plt.savefig("img.png")
plt.show()

```



ax.plot_wireframe(X, Y, Z)



ax.plot_surface(X, Y, Z, cmap="viridis")

Срез поверхностного графика

```

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

fig = plt.figure()
ax = plt.axes(projection="3d")

def f(x, y):
    return np.sin(np.sqrt(x ** 2 + y ** 2))

r = np.linspace(0, 6, 20)
theta = np.linspace(-0.9 * np.pi, 0.8 * np.pi, 40)

```

```

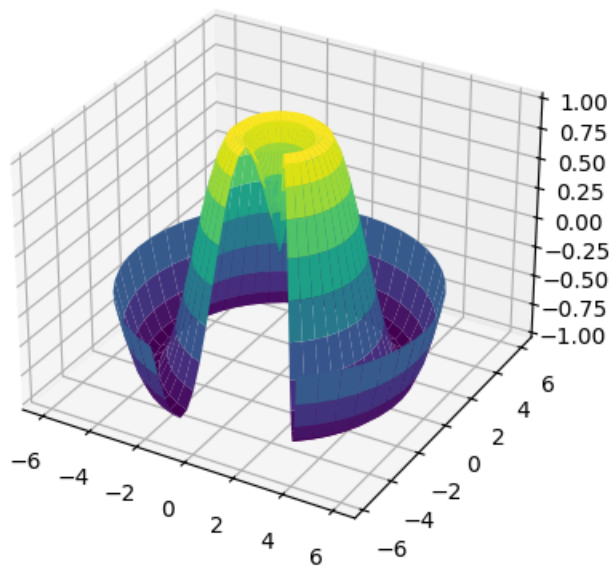
R, Theta = np.meshgrid(r, theta)

X = r * np.sin(Theta)
Y = r * np.cos(Theta)
Z = f(X, Y)

ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap="viridis", edgecolor="none")

plt.savefig("img.png")
plt.show()

```



Триангуляции поверхности

Триангуляция в Matplotlib — это процесс разбиения двумерной области на треугольники для визуализации данных, которые могут быть неравномерно распределены в пространстве. Триангуляция позволяет создать гладкие поверхности из дискретных точек, что делает данные более понятными и визуальнo привлекательными.

```

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

```

```

fig = plt.figure()
ax = plt.axes(projection="3d")

def f(x, y):
    return np.sin(np.sqrt(x ** 2 + y ** 2))

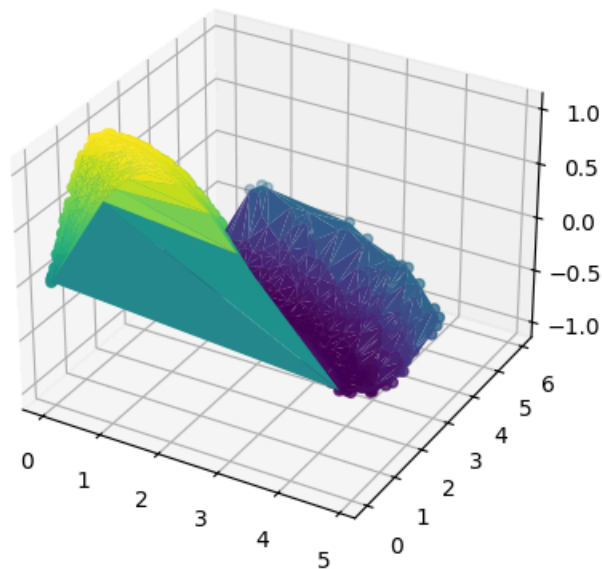
theta = 2 * np.pi + np.random.random(1000)
r = 6 * np.random.random(1000)

x = r * np.sin(theta)
y = r * np.cos(theta)
z = f(x, y)

ax.scatter(x, y, z, c=z, cmap="viridis")
ax.plot_trisurf(x, y, z, cmap="viridis")

plt.savefig("img.png")
plt.show()

```



Seaborn

- Работает с DataFrame (Matplotlib с Pandas)
- Более высокоуровневый интерфейс

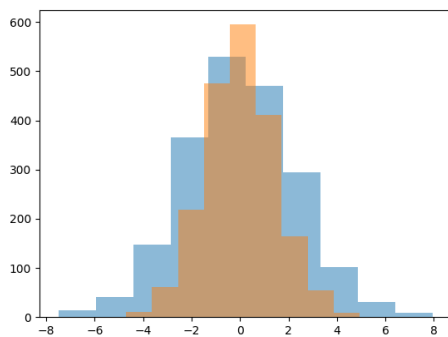
```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

data = np.random.multivariate_normal([0, 0], [[5, 2], [2, 2]], size=2000)
data = pd.DataFrame(data, columns=['X', 'Y'])

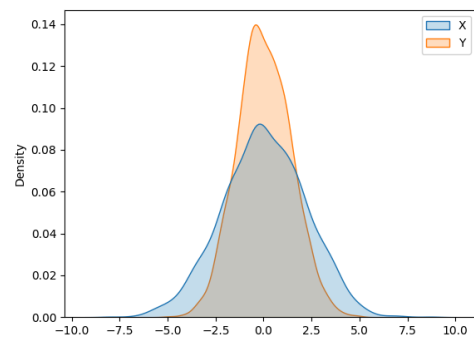
# Реализация через matplotlib
fig = plt.figure()
plt.hist(data['X'], alpha=0.5)
plt.hist(data['Y'], alpha=0.5)

# Реализация через seaborn
fig = plt.figure()
sns.kdeplot(data=data, shade=True)

plt.savefig("img.png")
plt.show()
```



matplotlib



seaborn

Seaborn DataSet

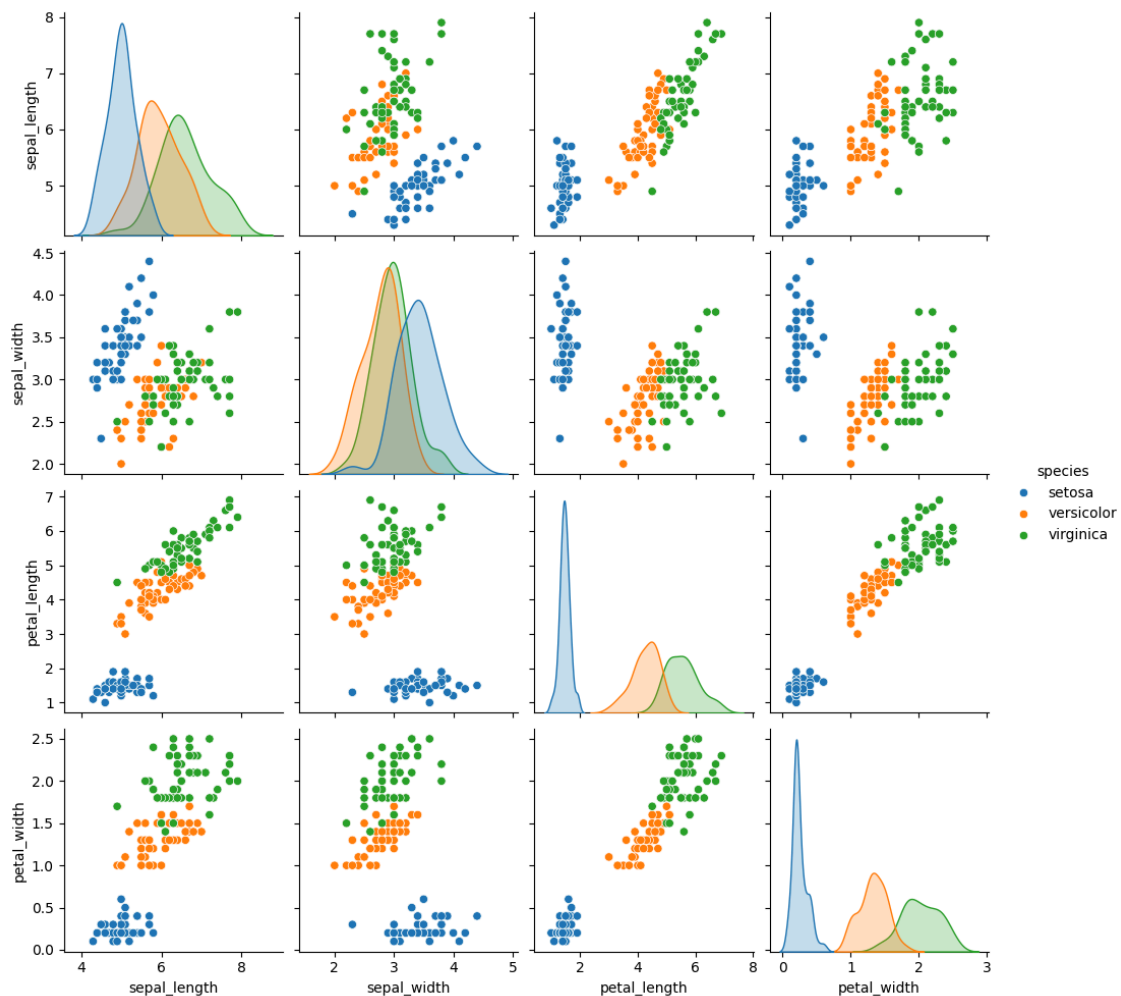
```
import numpy as np
import pandas as pd
import matplotlib as mpl
```

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

iris = sns.load_dataset("iris")

print(iris.head())
sns.pairplot(iris, hue="species")

plt.savefig("img.png")
plt.show()
```



Гистограммы подмножеств

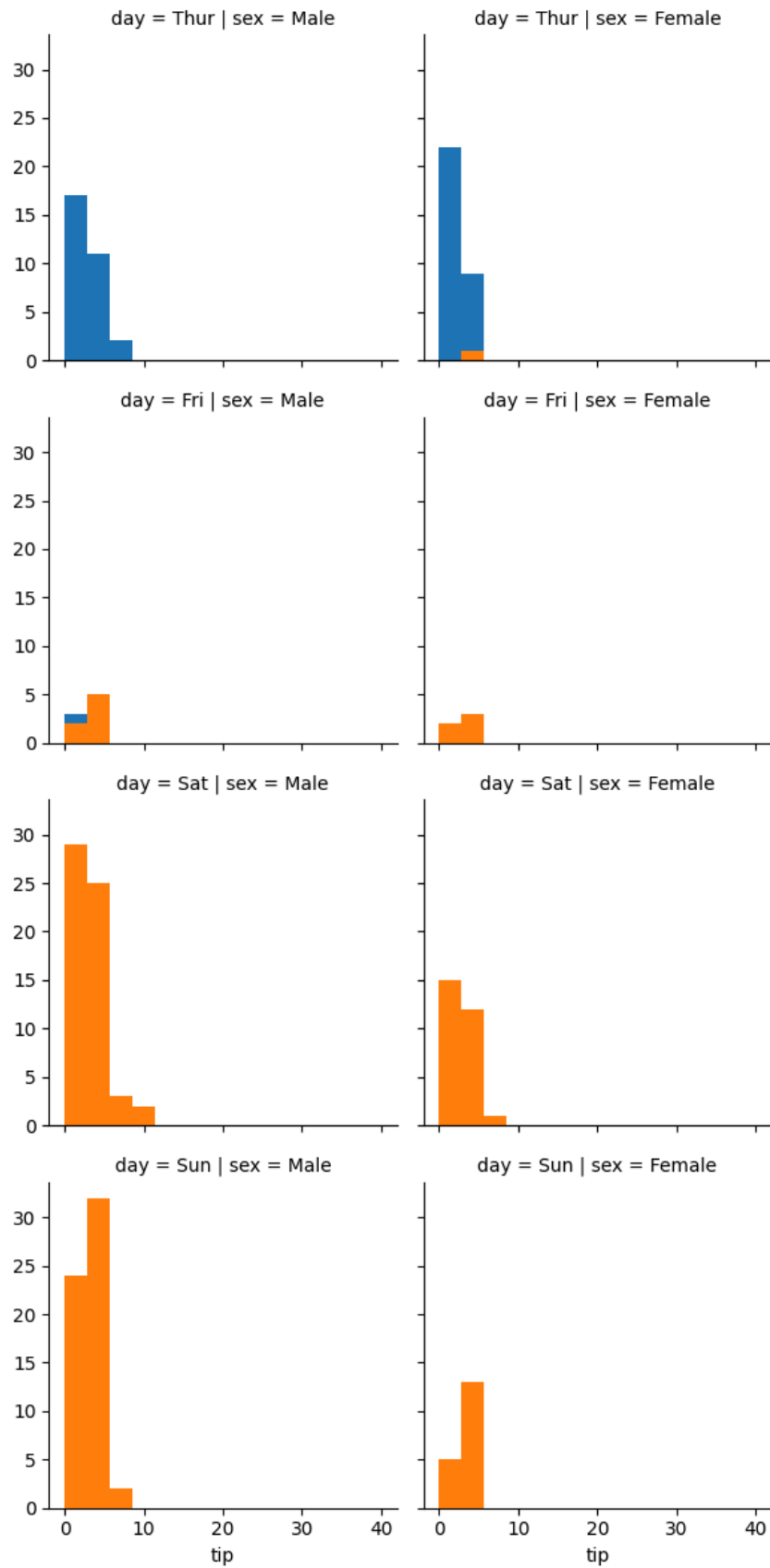
```
import numpy as np
import pandas as pd
import matplotlib as mpl
```



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

tips = sns.load_dataset("tips")
grid = sns.FacetGrid(tips, col="sex", row="day", hue="time")
grid.map(plt.hist, "tip", bins=np.linspace(0, 40, 15))

plt.savefig("img.png")
plt.show()
```



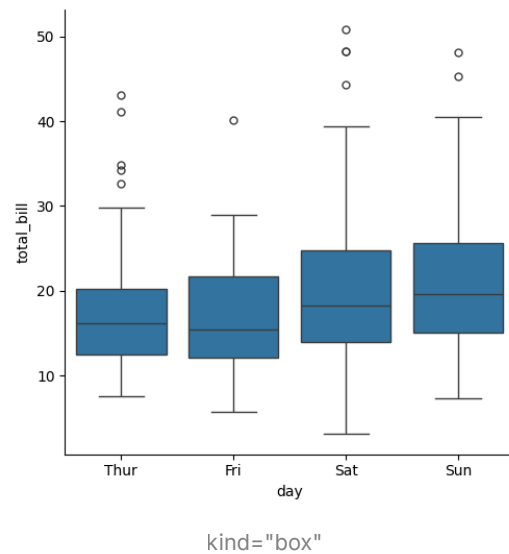
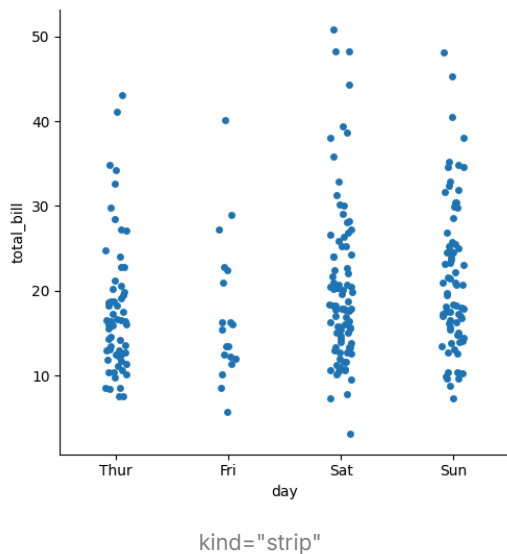
Графики факторов

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

tips = sns.load_dataset("tips")

sns.catplot(data=tips, x="day", y="total_bill", kind="strip")

plt.savefig("img.png")
plt.show()
```

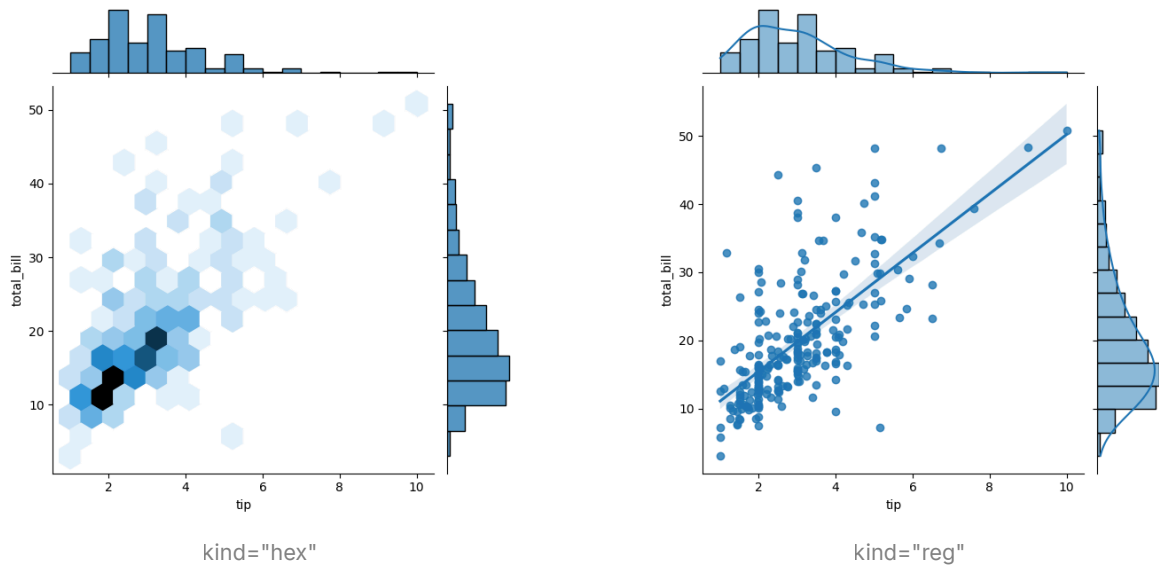


```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

tips = sns.load_dataset("tips")

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

```
plt.savefig("img.png")
plt.show()
```

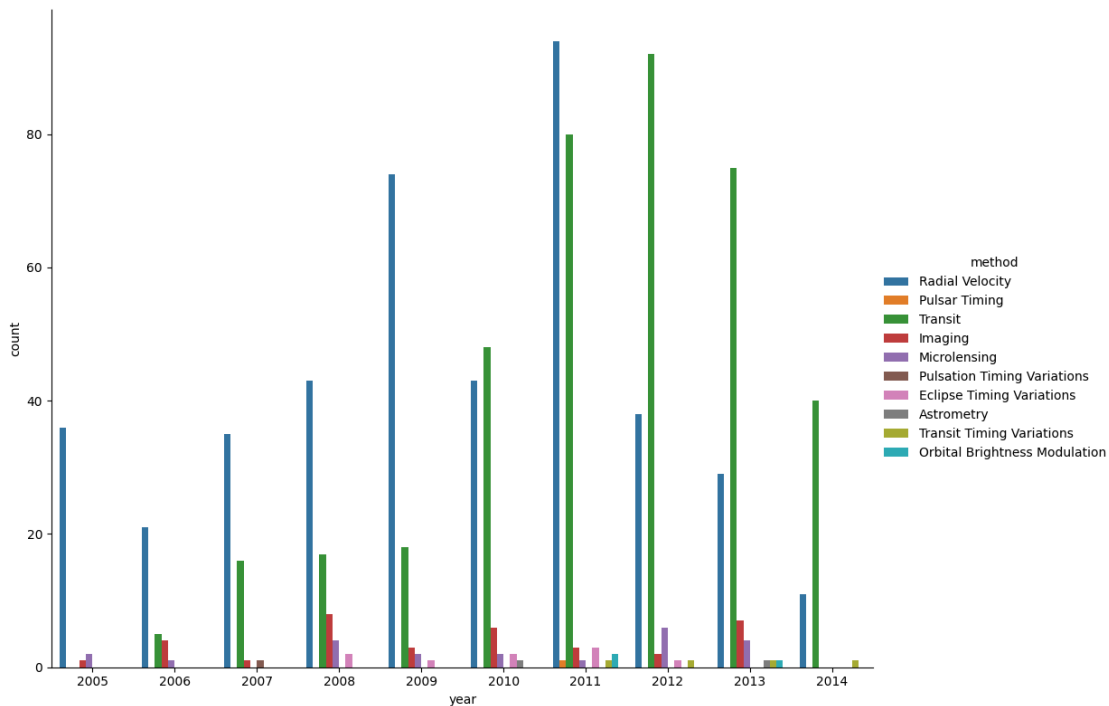


```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

planets = sns.load_dataset("planets")

sns.catplot(data=planets, x="year", kind="count", order=range(2005, 2015), \
            hue="method", height=8, aspect=1.25)

plt.savefig("img.png")
plt.show()
```



Диаграммы для изучения DataSet

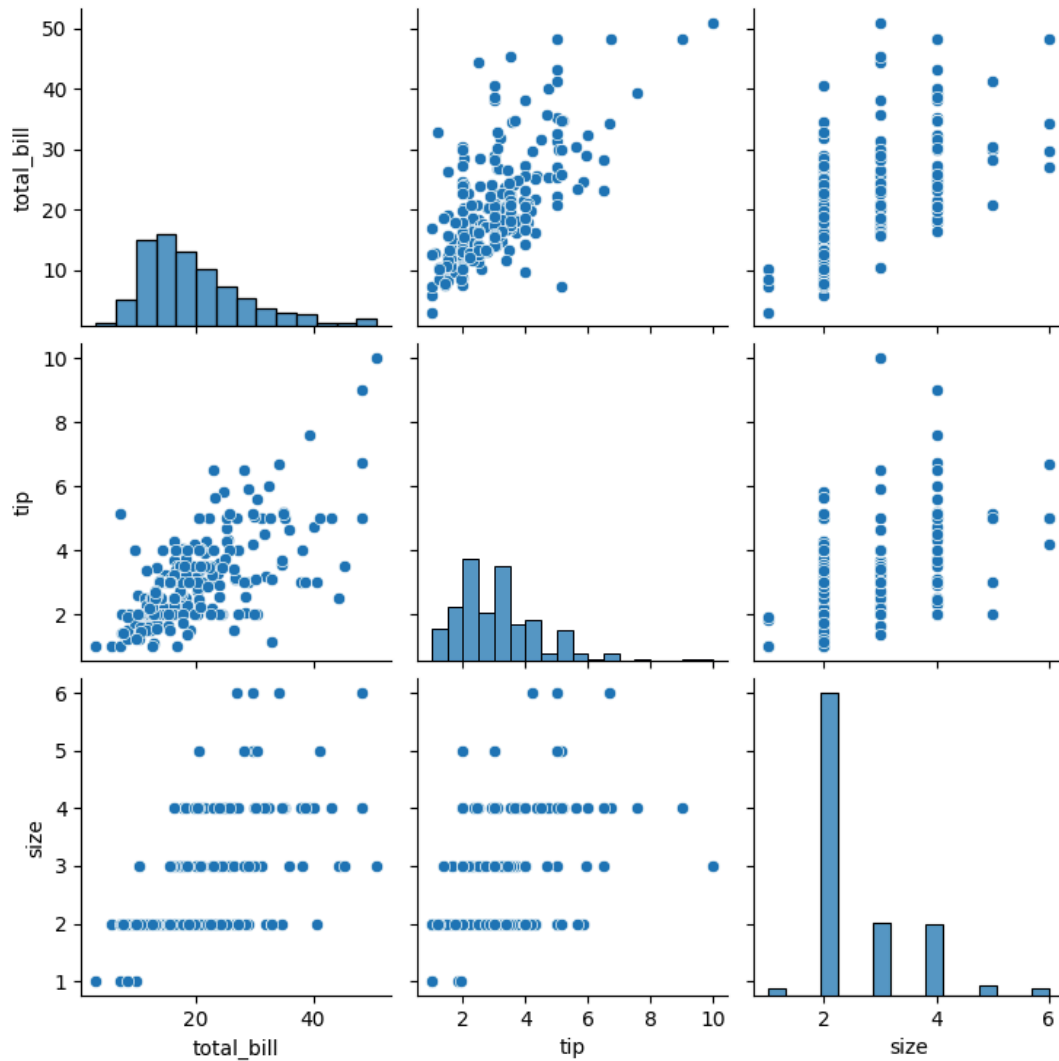
Сравнение числовых данных

Числовые пары

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

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

plt.savefig("img.png")
plt.show()
```



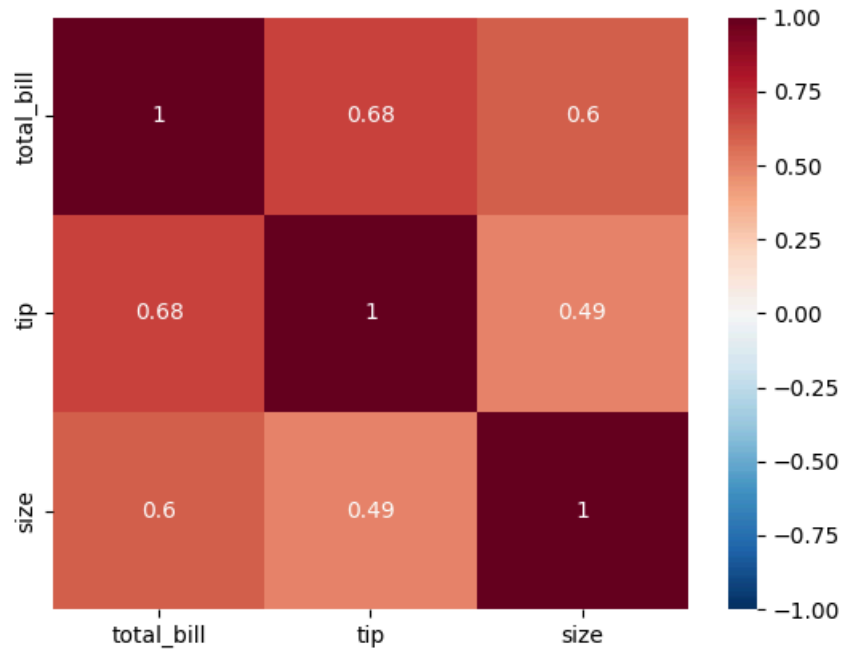
Тепловая карта

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

tips = sns.load_dataset("tips")
tips_corr = tips[["total_bill", "tip", "size"]]

sns.heatmap(tips_corr.corr(), cmap="RdBu_r", annot=True, vmin=-1, vmax=1)
```

```
plt.savefig("img.png")
plt.show()
```



0 - независимы

1 - положительная зависимость

-1 - отрицательная зависимость

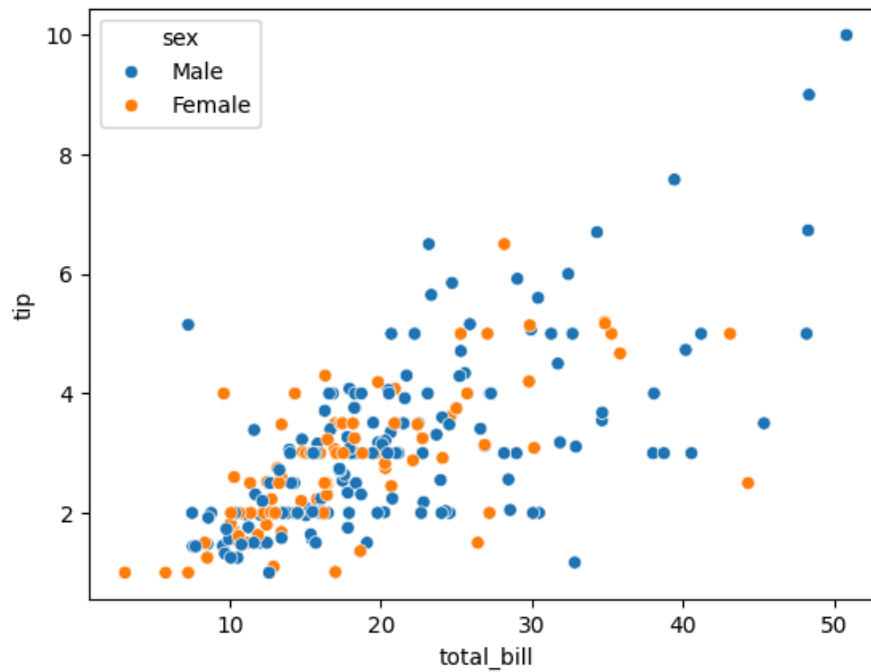
Диаграмма рассеяния

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

tips = sns.load_dataset("tips")

sns.scatterplot(data=tips, x="total_bill", y="tip", hue="sex")

plt.savefig("img.png")
plt.show()
```



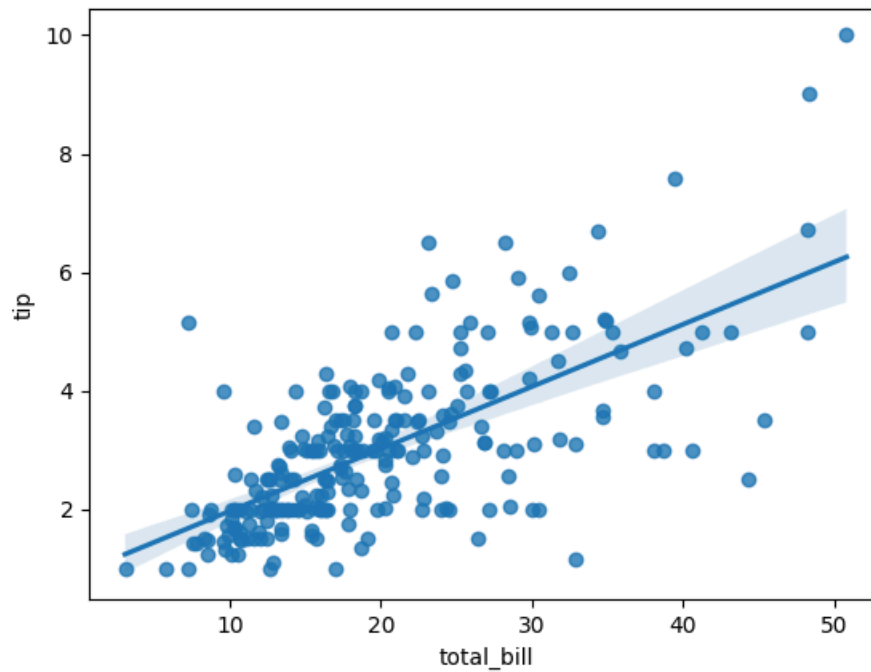
Линейная регрессия

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

tips = sns.load_dataset("tips")

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

plt.savefig("img.png")
plt.show()
```

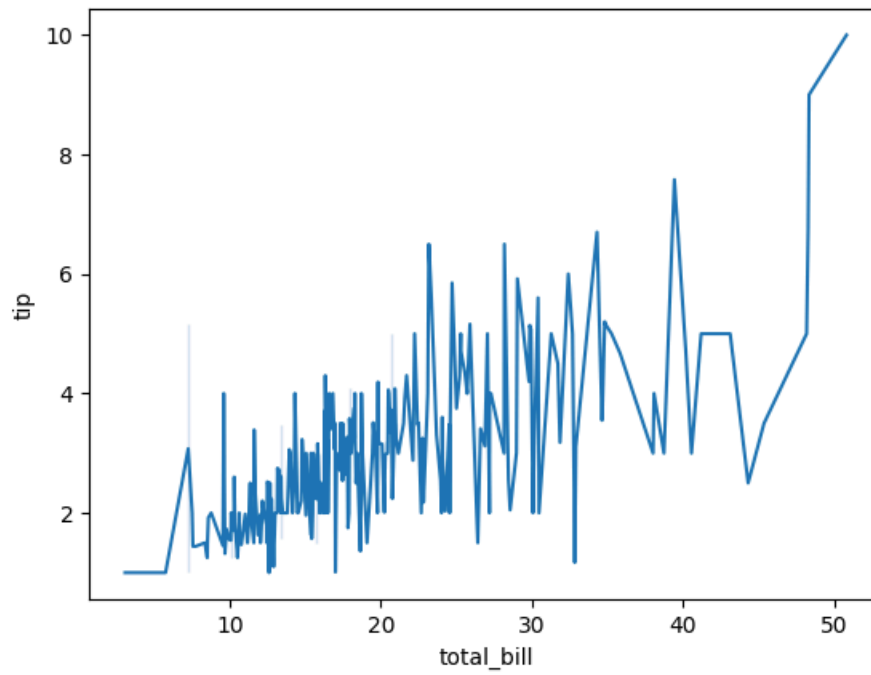
Линейный график

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

tips = sns.load_dataset("tips")

sns.lineplot(data=tips, x="total_bill", y="tip")

plt.savefig("img.png")
plt.show()
```



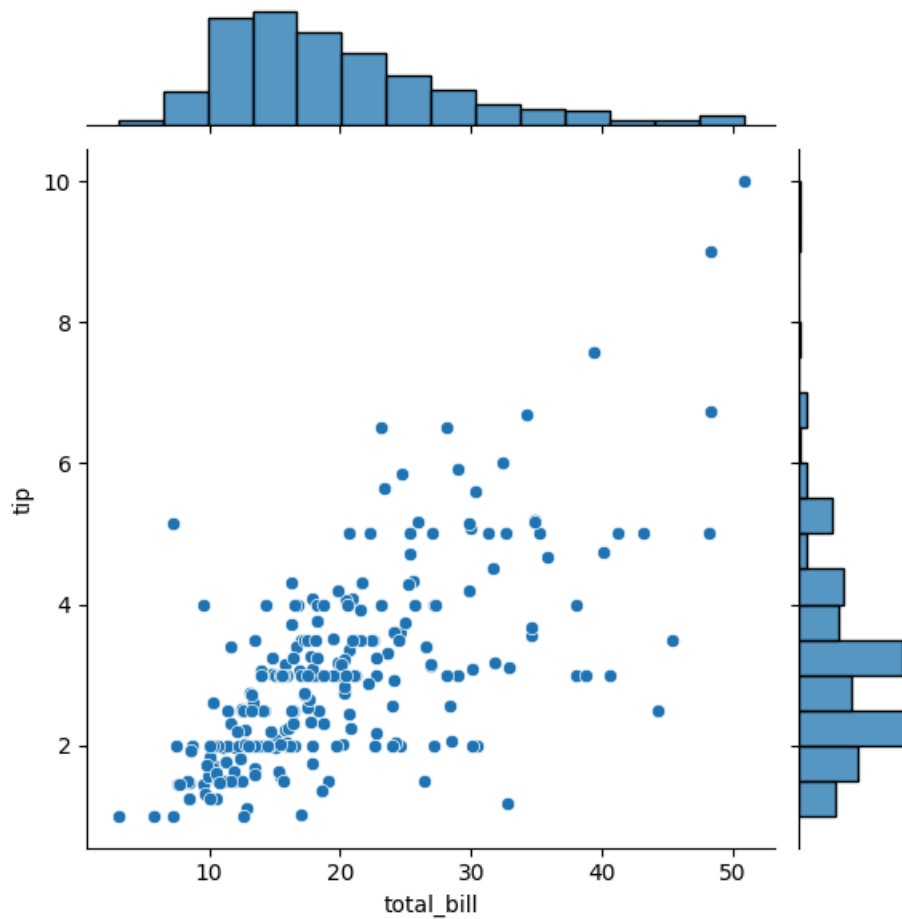
Сводная диаграмма

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

tips = sns.load_dataset("tips")

sns.jointplot(data=tips, x="total_bill", y="tip")

plt.savefig("img.png")
plt.show()
```



Сравнение числовых и категориальных данных

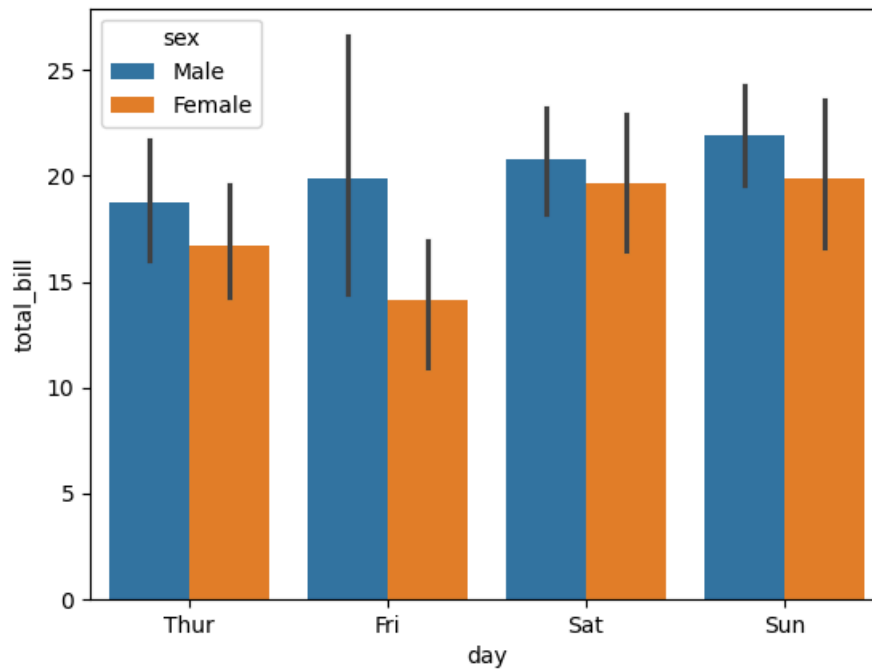
Гистограмма

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

tips = sns.load_dataset("tips")

sns.barplot(data=tips, x="day", y="total_bill", hue="sex")

plt.savefig("img.png")
plt.show()
```



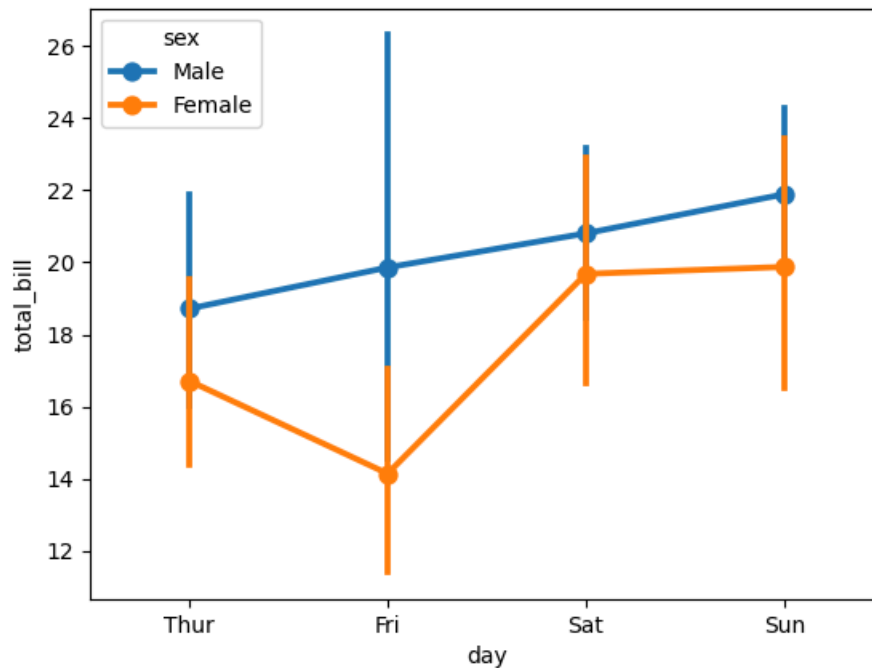
Точечная диаграмма

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

tips = sns.load_dataset("tips")

sns.pointplot(data=tips, x="day", y="total_bill", hue="sex")

plt.savefig("img.png")
plt.show()
```



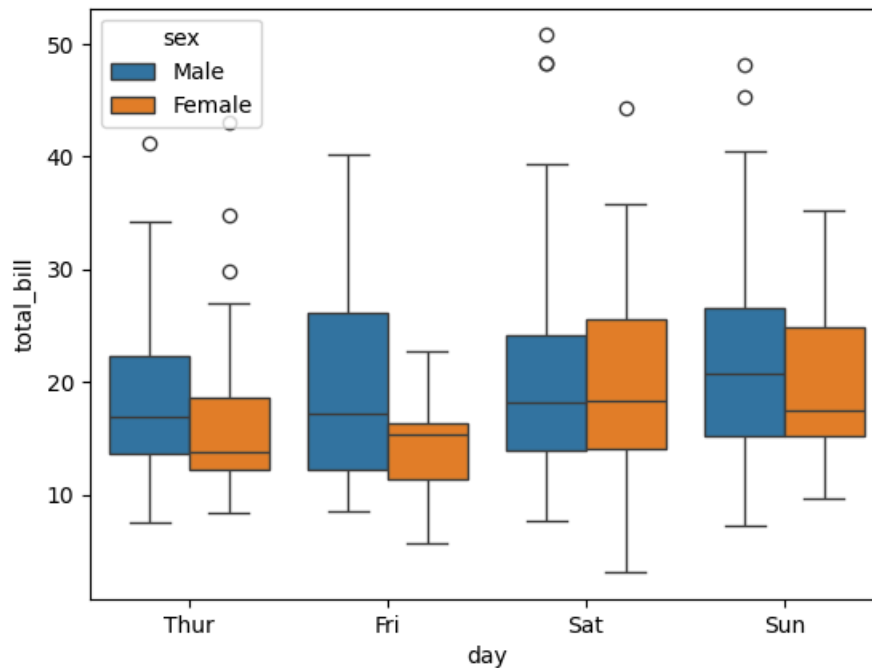
Ящик с усами

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

tips = sns.load_dataset("tips")

sns.boxplot(data=tips, x="day", y="total_bill", hue="sex")

plt.savefig("img.png")
plt.show()
```



Ящик	Медиана	Усы	Выбросы	Минимум и максимум
Представляет интерквартильный размах (IQR).	Линия внутри ящика.	Линии, простирающиеся от ящика.	Точки, находящиеся за пределами усов.	Обычно обозначаются концами усов, если они находятся в пределах 1.5 * IQR от квартилей.
Ограничен нижним квартилем (Q1) и верхним квартилем (Q3).	Делит ящик на две равные части.	Длина усов обычно равна 1.5 * IQR.	Обозначаются отдельными символами (например, кружками).	
Показывает диапазон средних 50% данных.	Представляет второй квартиль (Q2).	Показывают диапазон данных без выбросов.	Представляют собой значения, значительно отличающиеся от остальных.	

Скрипичная диаграмма

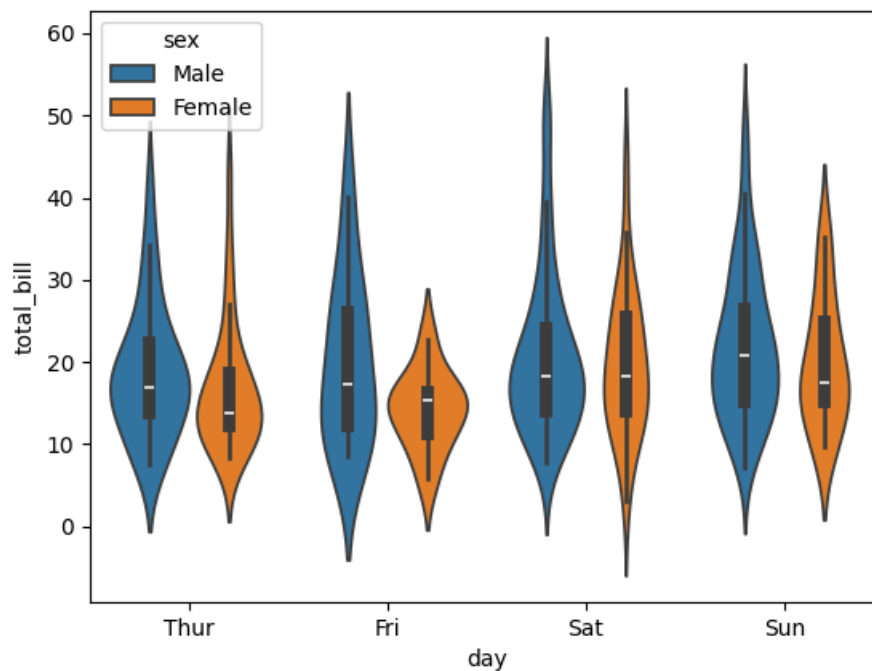
```
import numpy as np
import pandas as pd
import matplotlib as mpl
```

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

tips = sns.load_dataset("tips")

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

plt.savefig("img.png")
plt.show()
```



Одномерная диаграмма рассеяния

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

tips = sns.load_dataset("tips")

sns.stripplot(data=tips, x="day", y="total_bill", hue="sex")
```

```
plt.savefig("img.png")  
plt.show()
```

