

# Визуализация данных в Python

---

# Варианты библиотек

---

**matplotlib**  
make easy things easy and hard things possible

**Seaborn**  
complex statistical visualizations

# matplotlib



***Python Pandas***



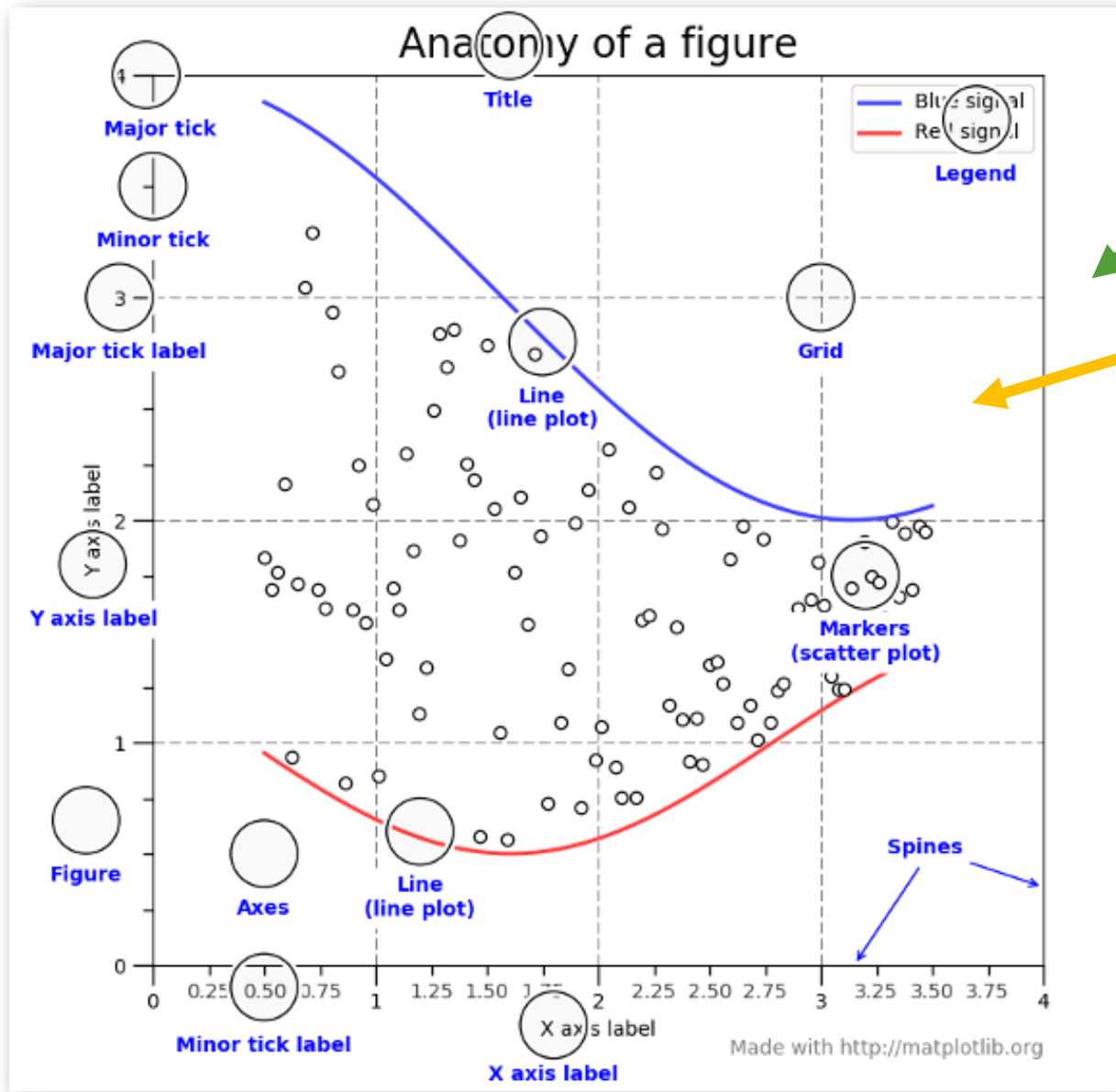
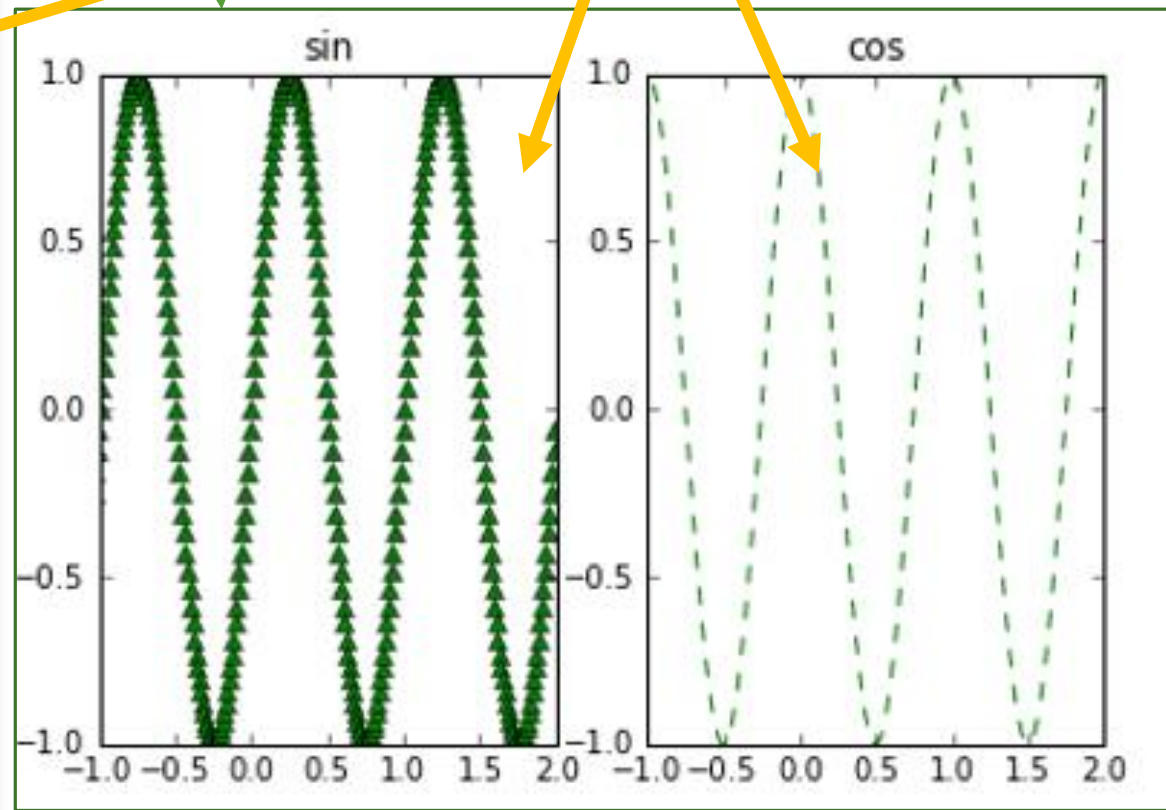


Figure - поле, область с графиками  
 Axes - сами графики



# Как прочитать данные?

## Pandas:

```
import pandas as pd
```

1. Файл *.xlsx* (MS Excel):

```
data = pd.read_excel('test.xlsx')
```

2. Файл *.csv*:

```
data = pd.read_csv('test.csv', sep = ';')
```

```
data = data.sort_values(by='Sales')↑
```

```
data = data.sort_values(by='Sales', ascending=False)↓
```

	Company	Sales	Quantity
0	Kulas Inc	137351.96	94
1	White-Trantow	135841.99	86
2	Trantow-Barrows	123381.38	94
3	Jerde-Hilpert	112591.43	89
4	Fritsch, Russel and Anderson	112214.71	81
5	Barton LLC	109438.50	82
6	Will LLC	104437.60	74
7	Koepp Ltd	103660.54	82
8	Frami, Hills and Schmidt	103569.59	72
9	Keeling LLC	100934.30	74

# Построение графиков

```
import matplotlib.pyplot as plt
```

```
fig, ax = plt.subplots() #1 поле с 1 графиком || plt.figure()
```

```
ax.barh(data.Company, data.Sales)
```

Чтобы напечатать  
названия всех  
столбцов, вызовите  
`print(data.columns)`

plot – линия

scatter – точечный график

bar – столбчатая диаграмма

barh – горизонтальная столбчатая диаграмма

boxplot – ящик с усами

fill\_between – заливка между 2 линиями

pie – круговая диаграмма

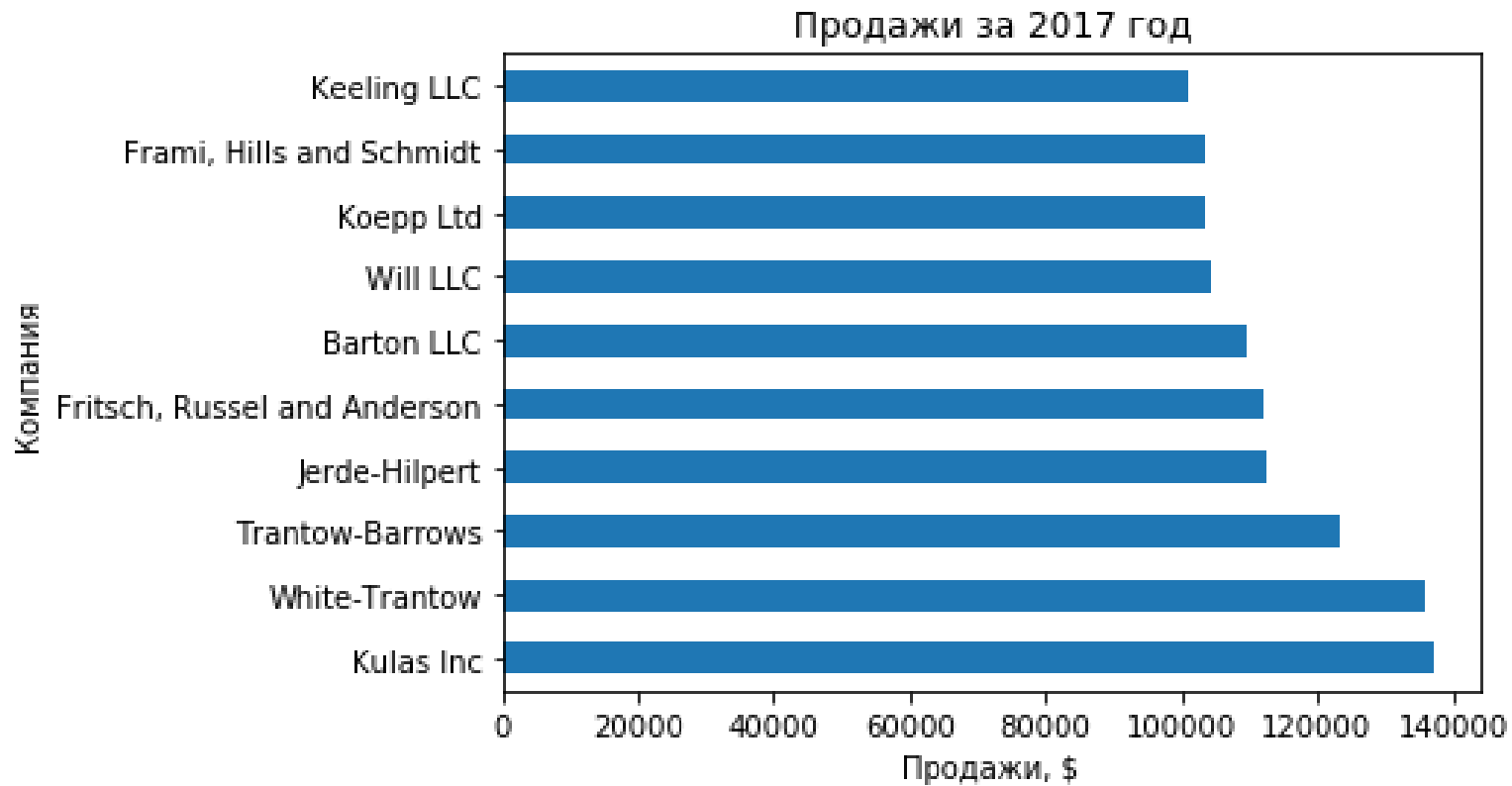
hist - гистограмма

hexbin – гексогональная 2-d гистограмма

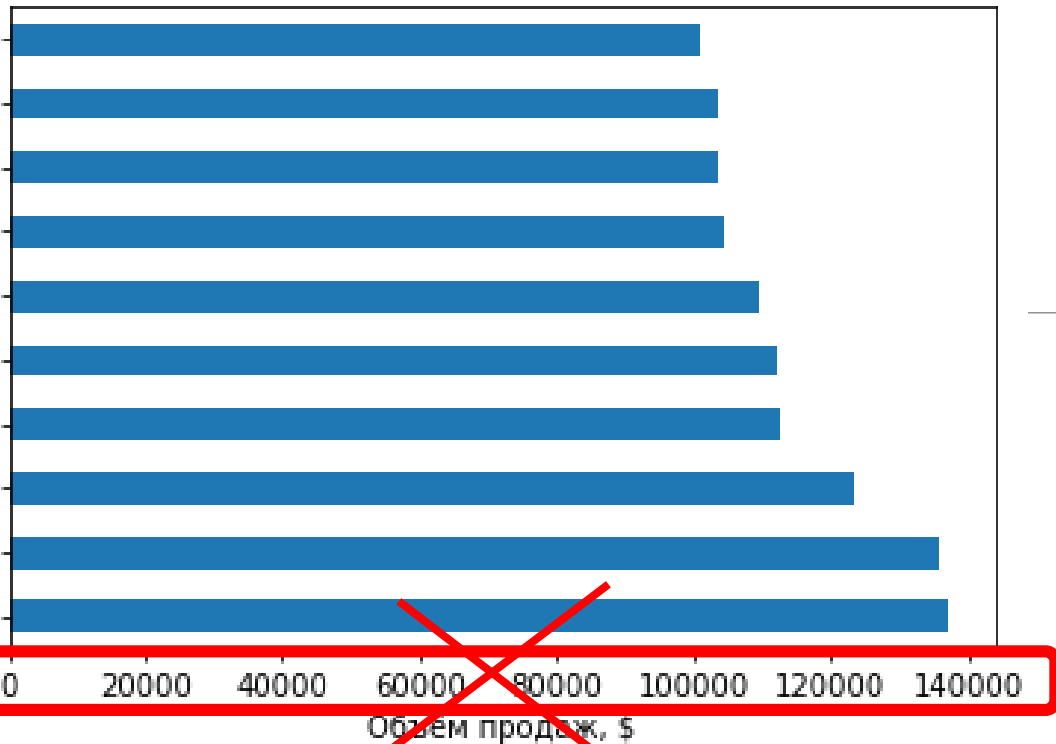
# Название графика и осей. Легенда

```
ax.set(title='Продажи за 2017 год',  
        xlabel = 'Объём продаж, $', ylabel='Компания')
```

```
ax.legend().set_visible(False) #скрываем легенду
```



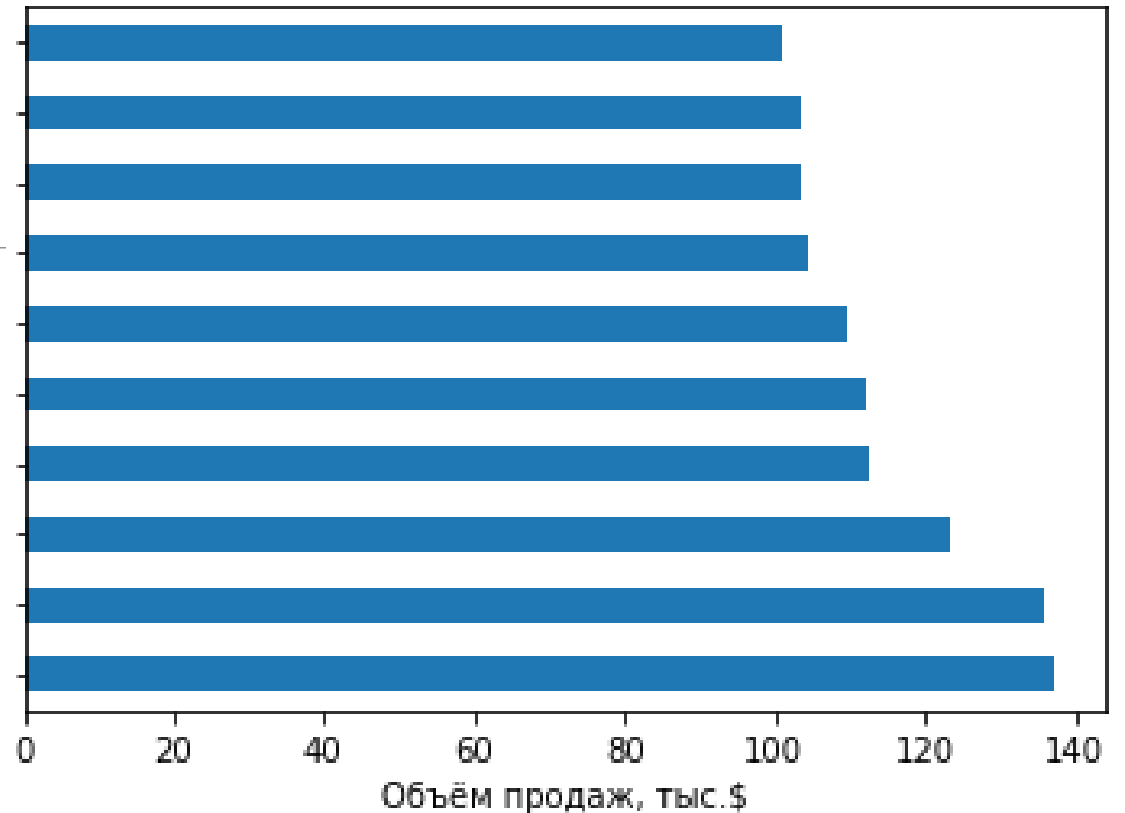
Продажи за 2017 год



```

from matplotlib.ticker import FuncFormatter
def changeMoney(x, pos):
    #return '${:1.0f}K'.format(x*1e-3)
    return '{:1.0f}'.format(x*1e-3)
....
formatter = FuncFormatter(changeMoney)
ax.xaxis.set_major_formatter(formatter)
    
```

Продажи за 2017 год





# Добавление линий

```
avg = data['Sales'].mean()
```

```
competitors = data[data.Sales > avg]
```

```
ax.axhline(y=len(competitors)-0.5, color='r', label='Average', linestyle='--', linewidth=2)
```

```
# добавляем линию, отсекая всех, кто набрал больше среднего
```




# Несколько графиков

```
fig, ax = plt.subplots(nrows=1, ncols=2, sharey=True, figsize=(7, 4))
```

 количество строк

 количество столбцов

 одна подпись для всех осей Y

 размер поля  
для графиков

```
ax[0].barh(data.Company, data.Sales)
```

```
ax[0].set(ylabel='Продавец', xlabel = 'Выручка', xlim = [-10000, 140000])
```

....

```
ax[1].barh(data.Company, data.Quantity)
```

```
ax[1].set(xlabel = 'Количество товаров', xlim = [0, 110])
```

```
fig.suptitle('Анализ продаж 2017', fontsize=14, fontweight='bold');
```

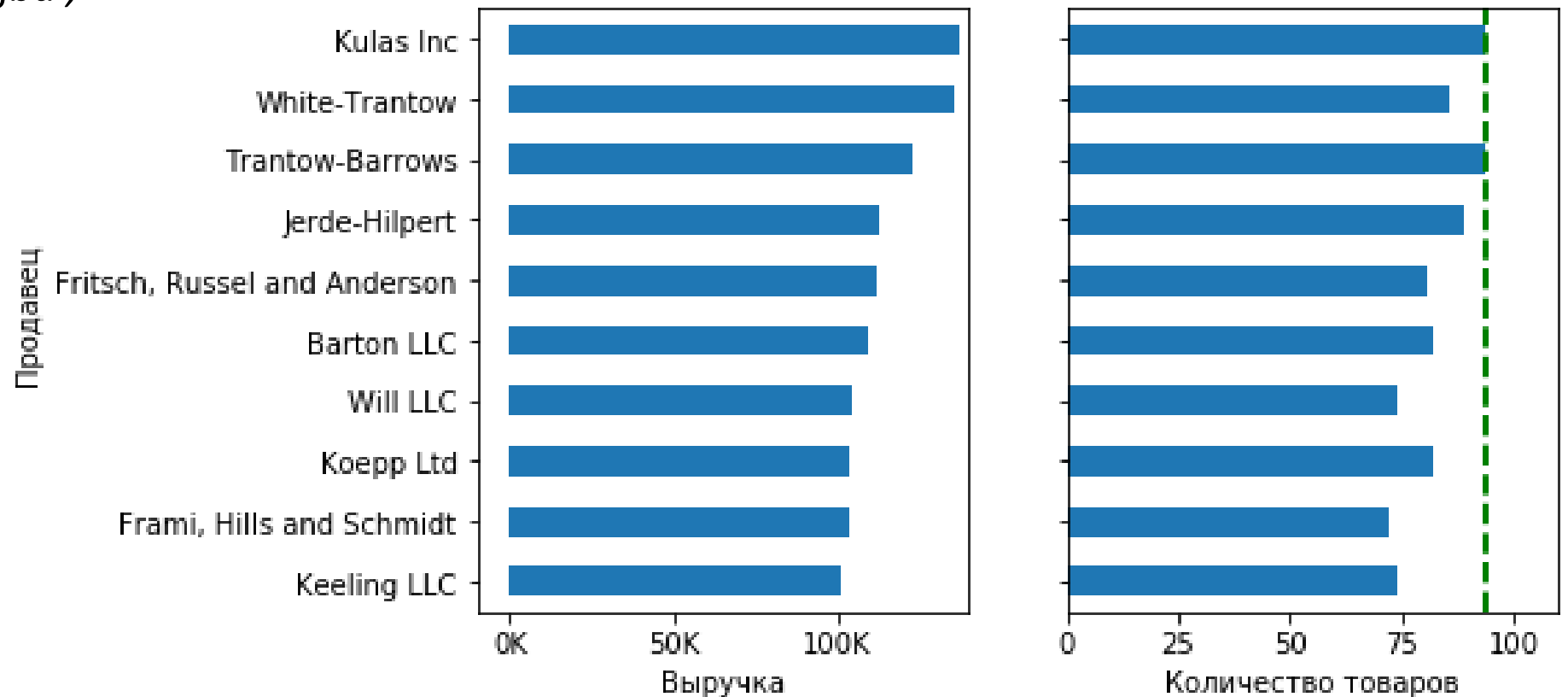
# Сохранение

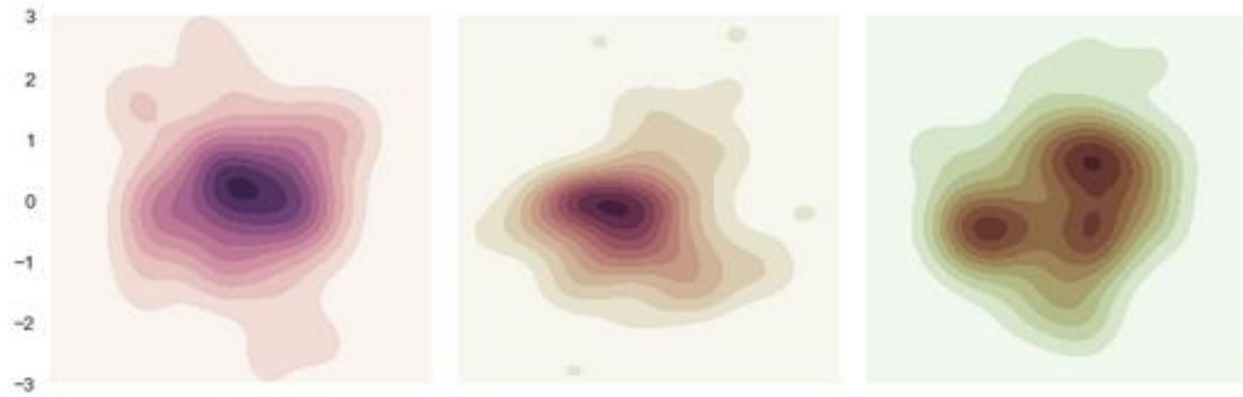
`fig.savefig('sales.png',...)`

`'eps', 'jpeg', 'jpg', 'pdf',  
'pgf', 'png', 'ps', 'raw', 'rgba',  
'svg', 'svgz', 'tif', 'tiff'`



## Анализ продаж 2017



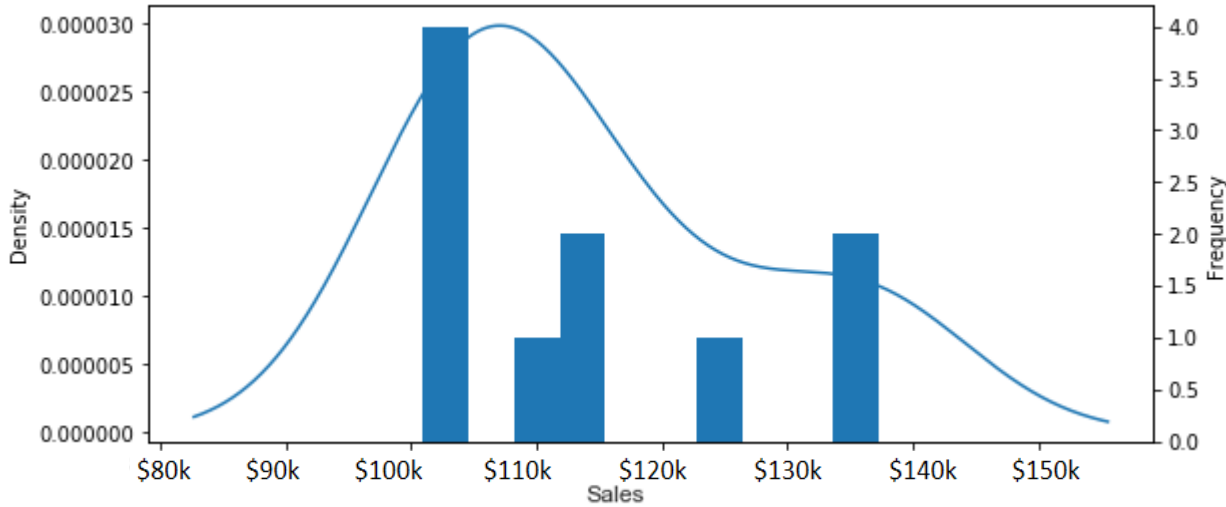


# Seaborn



# Matplotlib vs Seaborn

+ **pandas.plot**

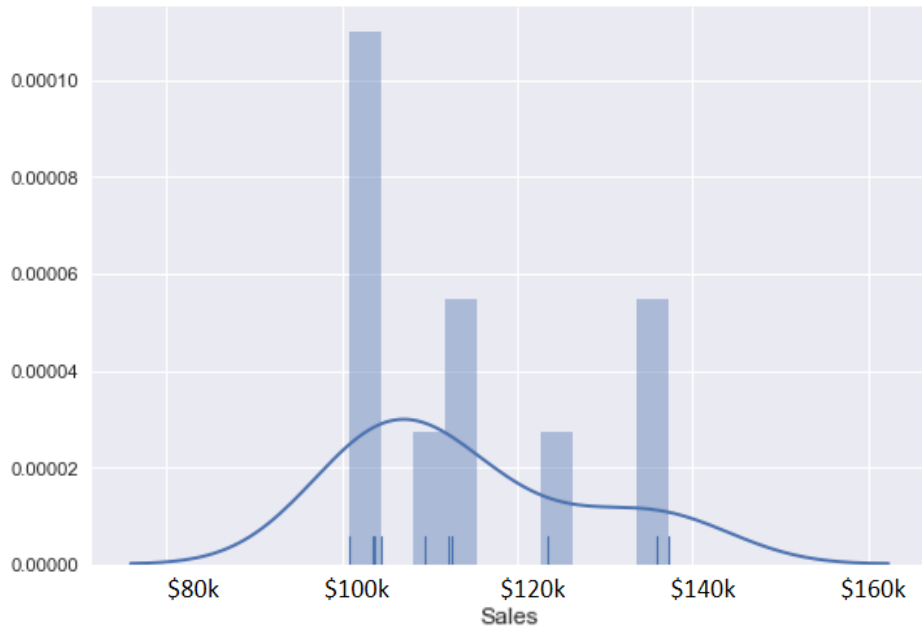


*kde только в pandas.plot:*

```
data.Sales.plot.kde(ax = ax)
```

```
ax1 = ax.twinx()
```

```
ax1.hist(data.Sales, bins = 50)
```



```
fig, ax = plt.subplots(figsize = (9,4))
```

```
ax = sns.distplot(data['Sales'], kde=True, rug=True, bins = 10)
```

# Scatter plot

---

```
import seaborn as sns
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
data = pd.read_csv("testChina.csv", sep = ',')
```

```
ax = sns.regplot(x = 'Sales', y = 'Quantity', data = data, marker = "x", color = 'r',  
                fit_reg = False)
```

```
ax.set(title = "Взаимосвязь объёма продаж и выручки")
```

```
[или: ax.set_title("Взаимосвязь объёма продаж и выручки") ]
```

# Scatter plot

**fit\_reg = False**

Взаимосвязь объёма продаж и выручки



**fit\_reg = True**

Взаимосвязь объёма продаж и выручки



# Scatter plot. 3D

---

```
data = pd.read_csv("testChina.csv", sep = ',')
```

```
ax = sns.regplot(x = 'Sales', y = 'Quantity', data = data, marker = "x", color = 'r',  
                fit_reg = False)
```

```
ax.set(title = "Взаимосвязь объёма продаж и выручки")
```

или

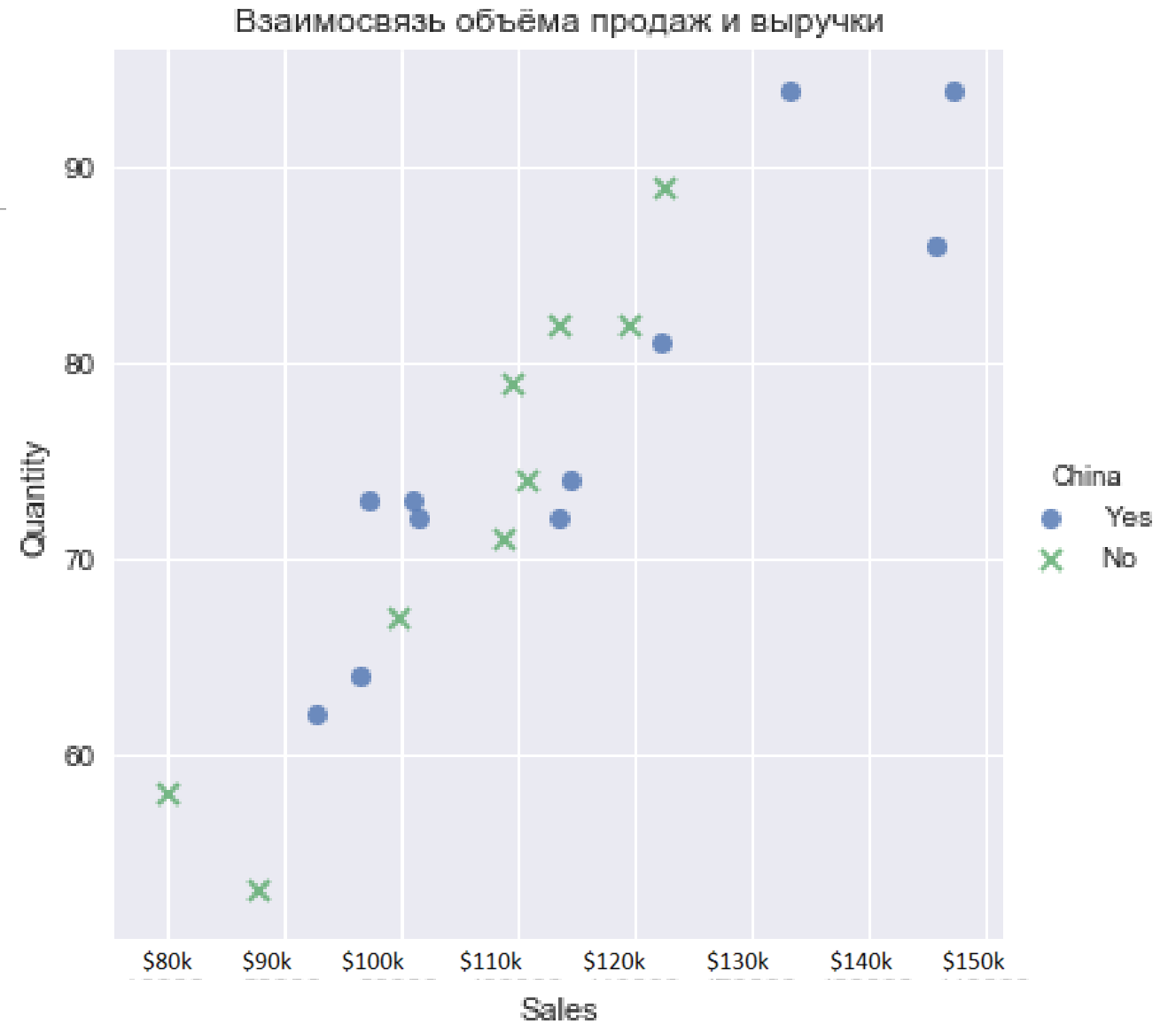
```
fg = sns.lmplot(x = 'Sales', y = 'Quantity', hue = 'China', data = data, markers = ['o', 'x'],  
               fit_reg = False)
```

```
fg.ax.set(title = "Взаимосвязь объёма продаж и выручки")
```



# Scatter plot. 3D

---



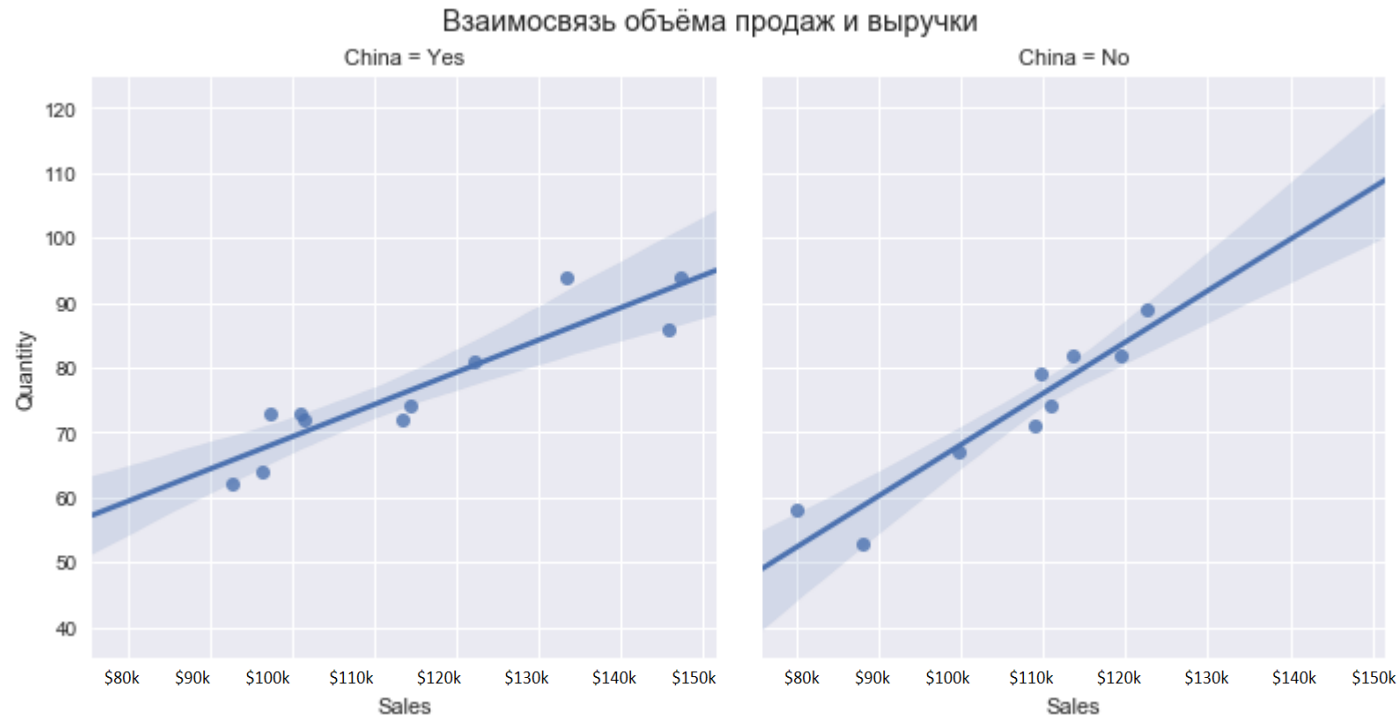
# Implot

```
fg = sns.lmplot(x = 'Sales', y = 'Quantity', hue = 'China', data = data)
```

ИЛИ

```
fg = sns.lmplot(x = "Sales", y = 'Quantity', col='China', data=data)
```

```
fg.fig.suptitle("Взаимосвязь объёма продаж и выручки")
```



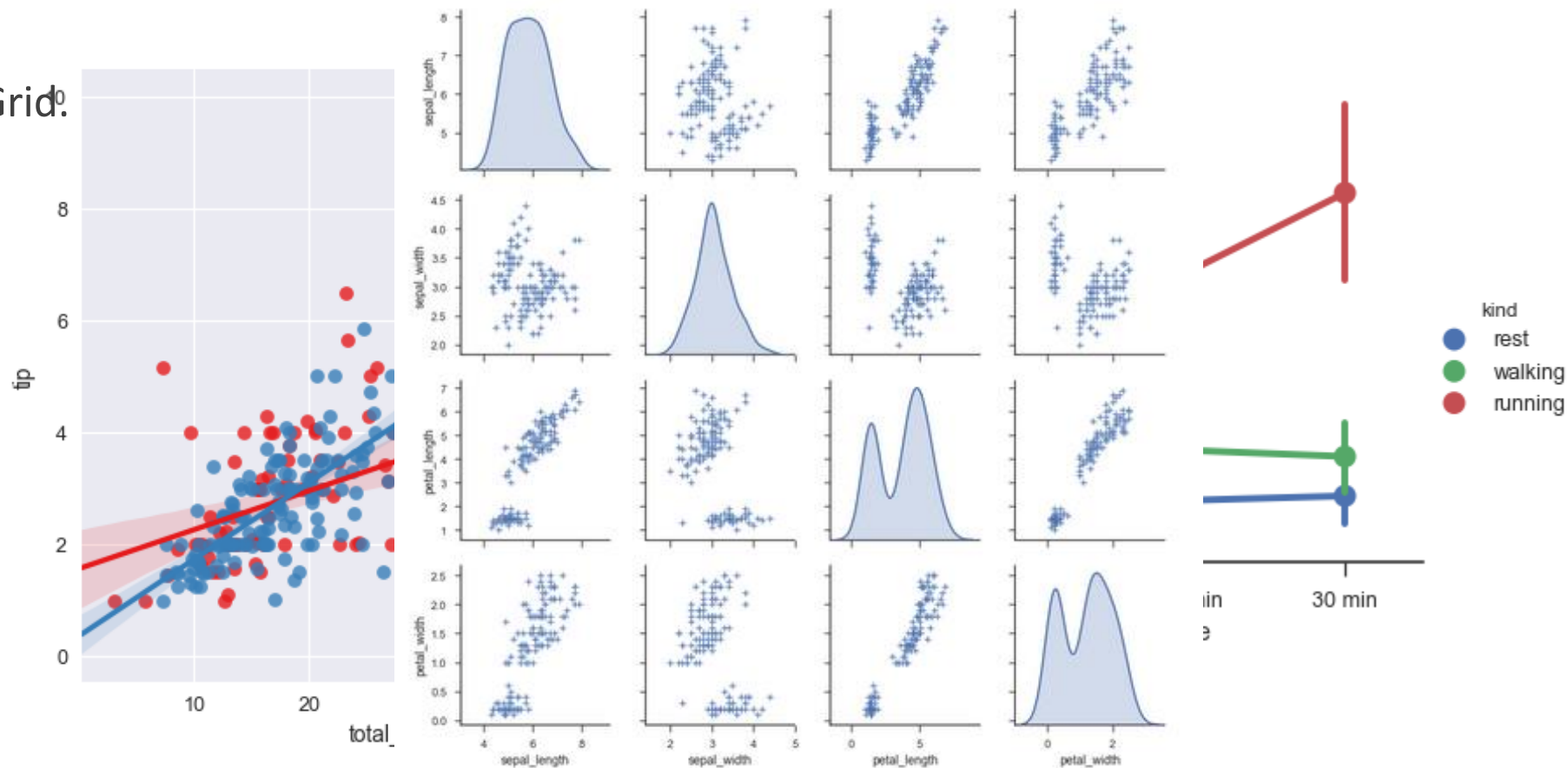
# Возвращаемые значения в Seaborn

Обычно: Axes.

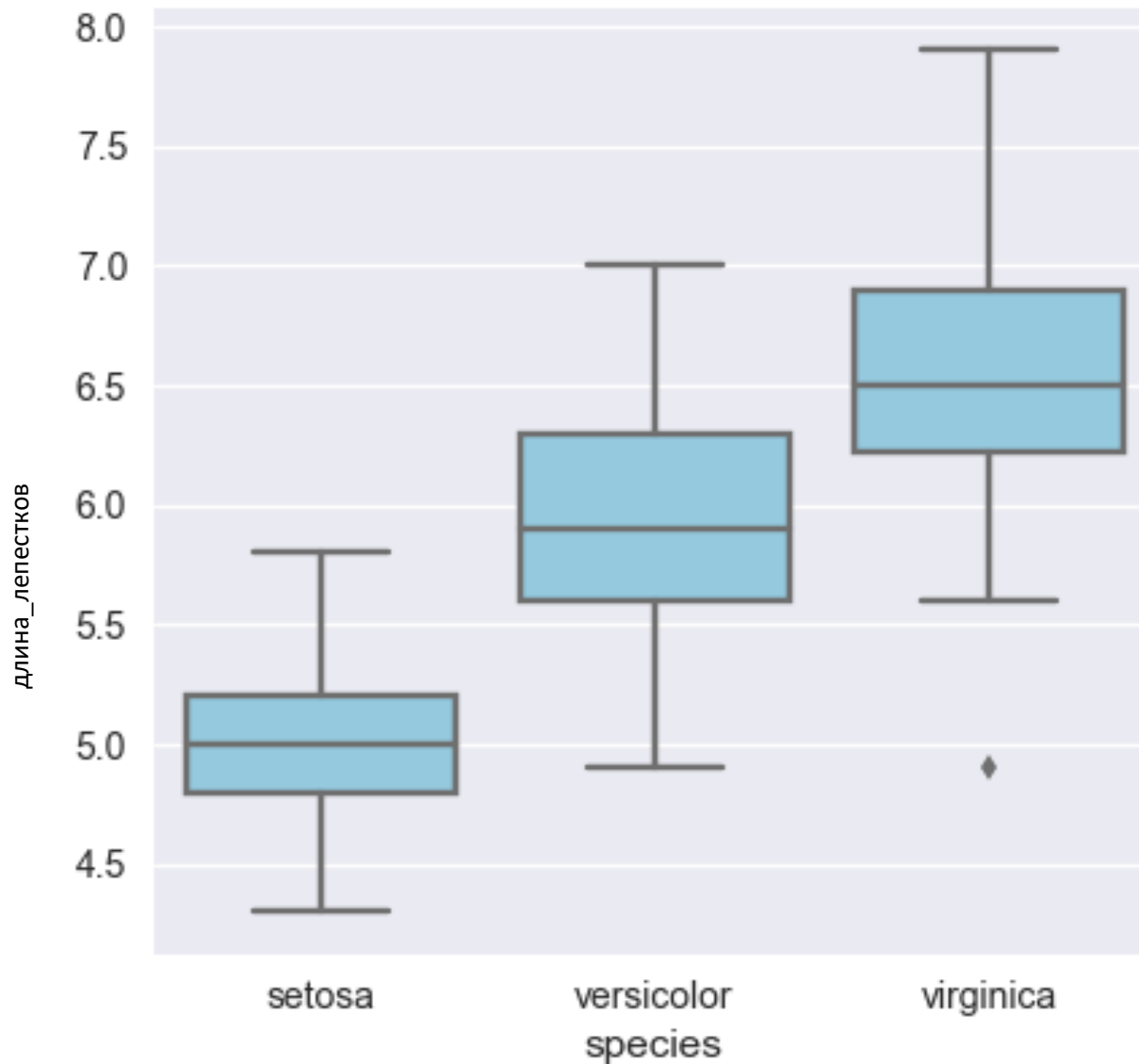
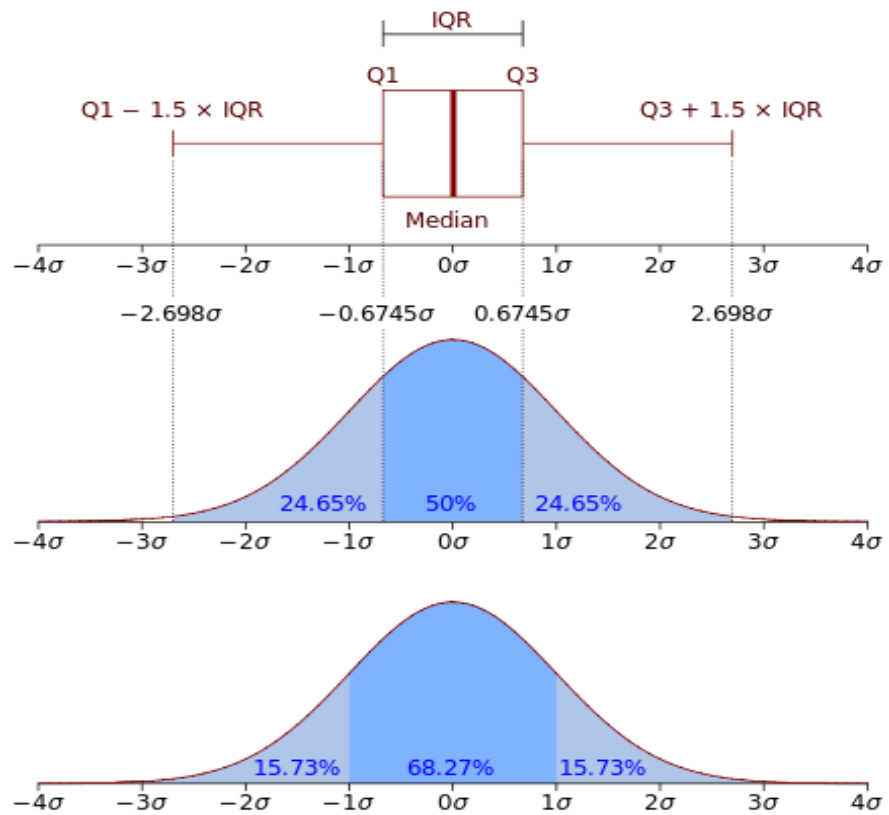
Для **Implot**, **factorplot**: FacetGrid.

Для **jointplot**: JointGrid.

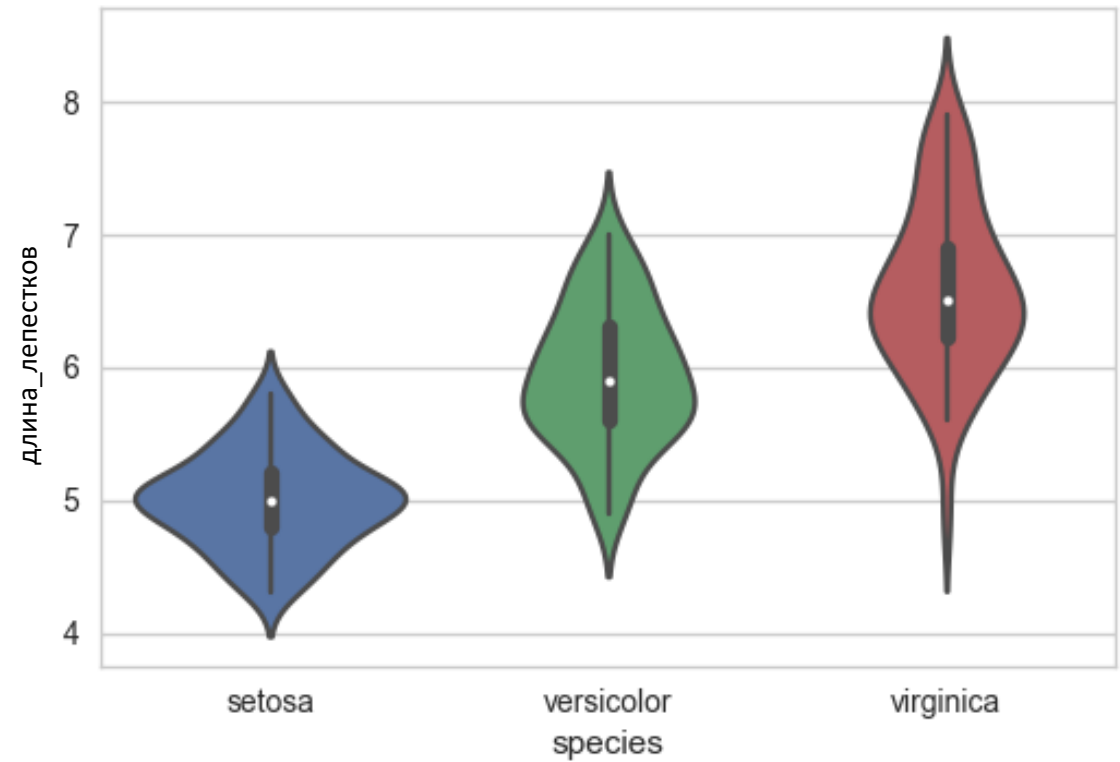
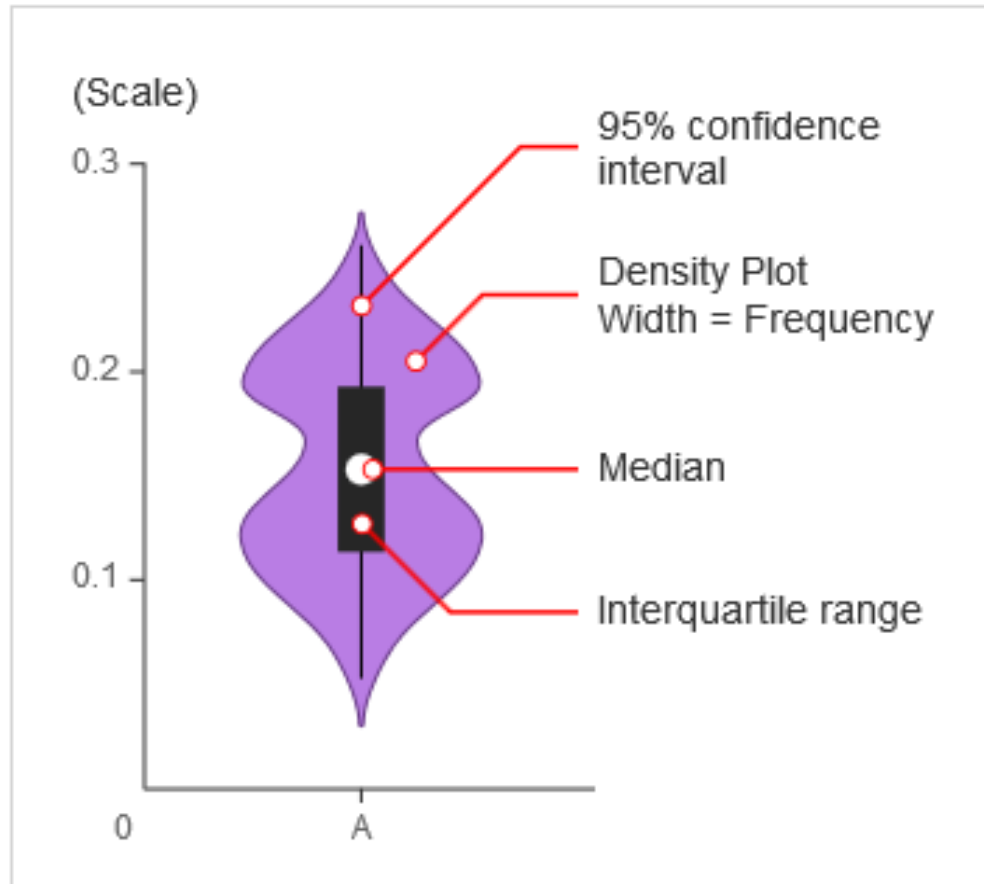
Для **pairplot**: PairGrid.



# Boxplot



# Violinplot



# Возможно полезные ссылки

---

<http://prog.tversu.ru/winter3.html> -- задание и данные.

[https://matplotlib.org/3.1.1/api/as\\_gen/matplotlib.pyplot.html](https://matplotlib.org/3.1.1/api/as_gen/matplotlib.pyplot.html) -- все возможности matplotlib

<https://matplotlib.org/3.1.1/gallery/index.html> -- примеры графиков matplotlib

<https://pandas.pydata.org/pandas-docs/stable/api.html#api-dataframe-plotting> – справочник по графикам в pandas с использованием Matplotlib

<https://pandas.pydata.org/pandas-docs/stable/visualization.html> -- примеры графиков

<https://seaborn.pydata.org/api.html> -- справочник по возможностям Seaborn

<https://seaborn.pydata.org/examples/index.html> -- примеры графиков Seaborn

Русскоязычные источники:

<http://malev.ru/анализ-данных-при-помощи-python-графики-в-pandas/> -- что-то про Matplotlib

<https://habrahabr.ru/company/ods/blog/323210/> -- немного про Seaborn