

Визуализация данных в Python

Варианты библиотек

matplotlib
make easy things easy and hard things possible

Seaborn
complex statistical visualizations

matplotlib



Python Pandas



Как прочитать данные?

Pandas:

```
import pandas as pd
```

1. Файл *.xlsx* (MS Excel):

```
data = pd.read_excel('test1.xlsx')
```

2. Файл *.csv*:

```
data = pd.read_csv('test2.csv', sep = ';')
```

```
data = data.sort_values(by='Sales')[ :10] ↑
```

```
data = data.sort_values(by='Sales', ascending=False)[ :10] ↓
```

	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

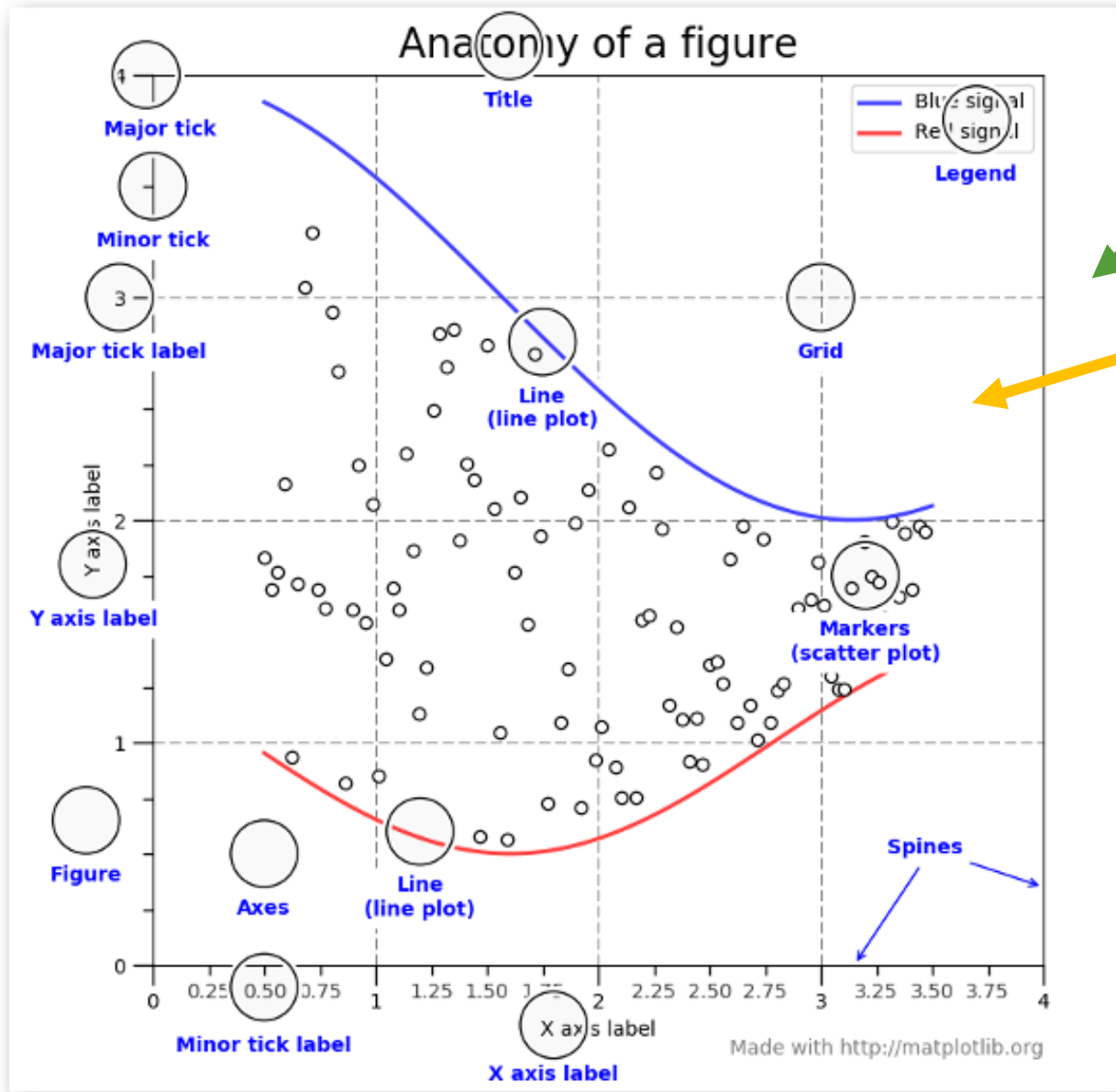
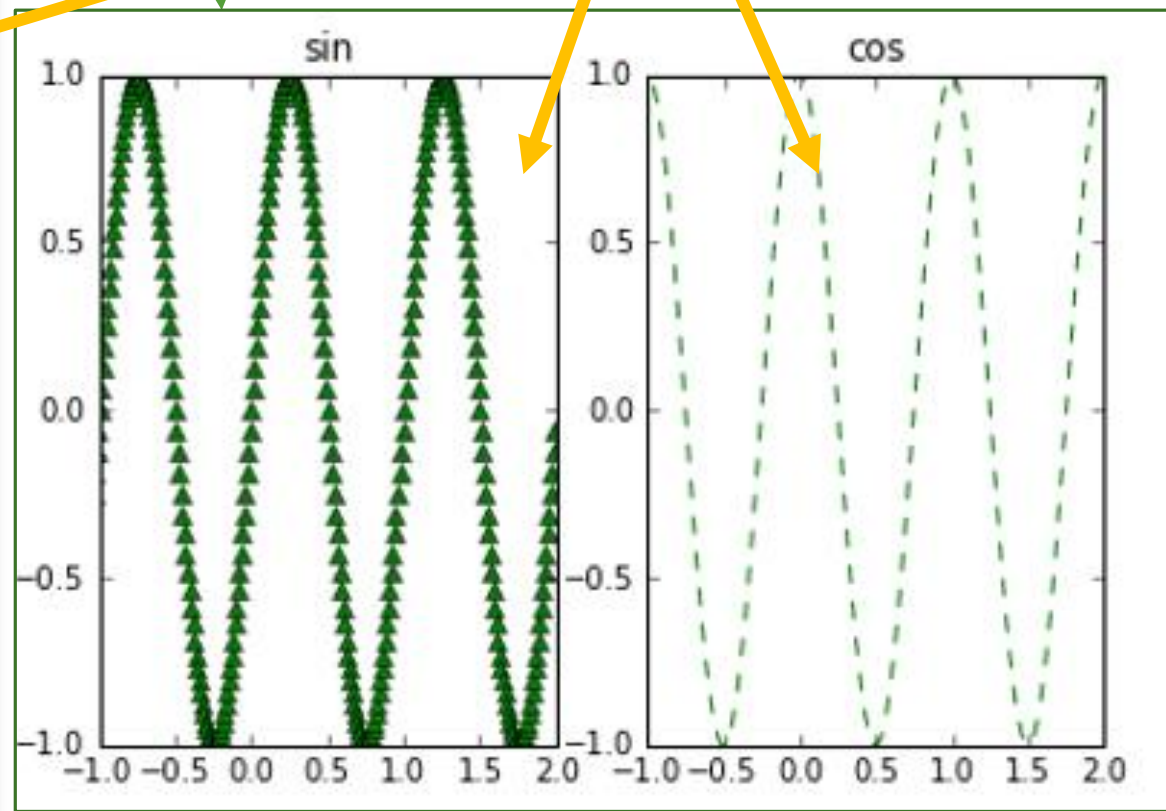


Figure – поле, область с графиками
 Axes – сами графики



Построение графиков

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

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

```
data.plot(kind='barh', x="Company", y="Sales", ax = ax)
```

line – линия

bar – столбчатая диаграмма

barh – горизонтальная столбчатая диаграмма

hist - гистограмма

box – ящик с усами

kde – график плотности распределения

density – аналогично kde

area – с областями

pie – круговая диаграмма

scatter – точечная диаграмма

hexbin – гексогональная 2-d гистограмма

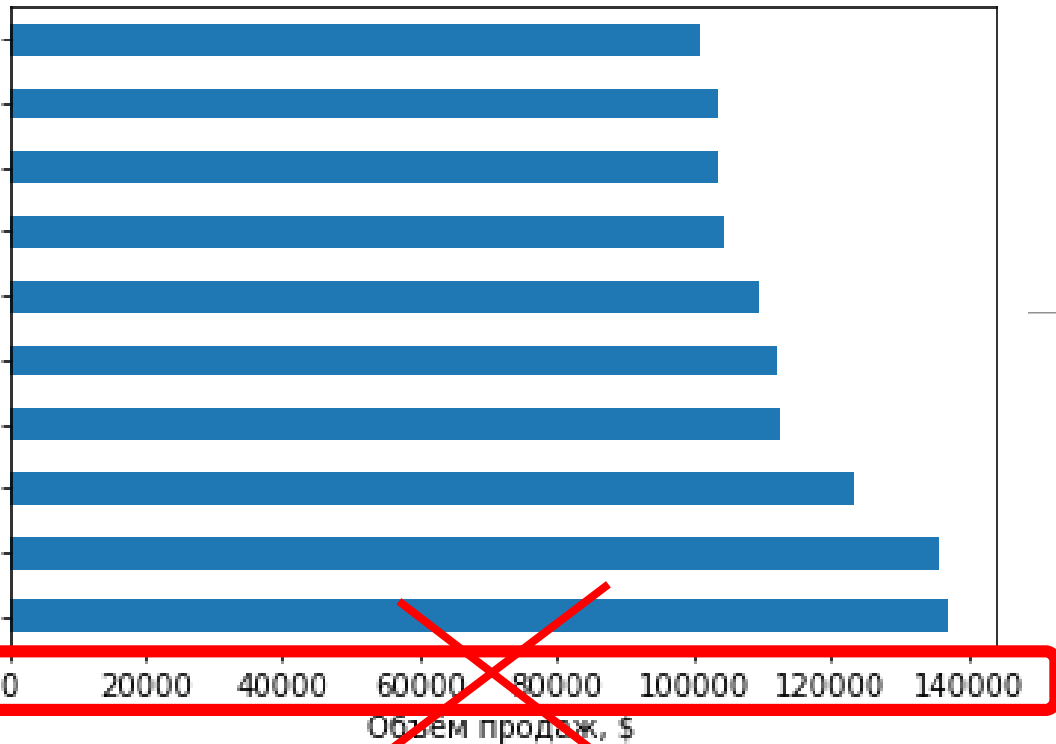
Название графика и осей. Легенда

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

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



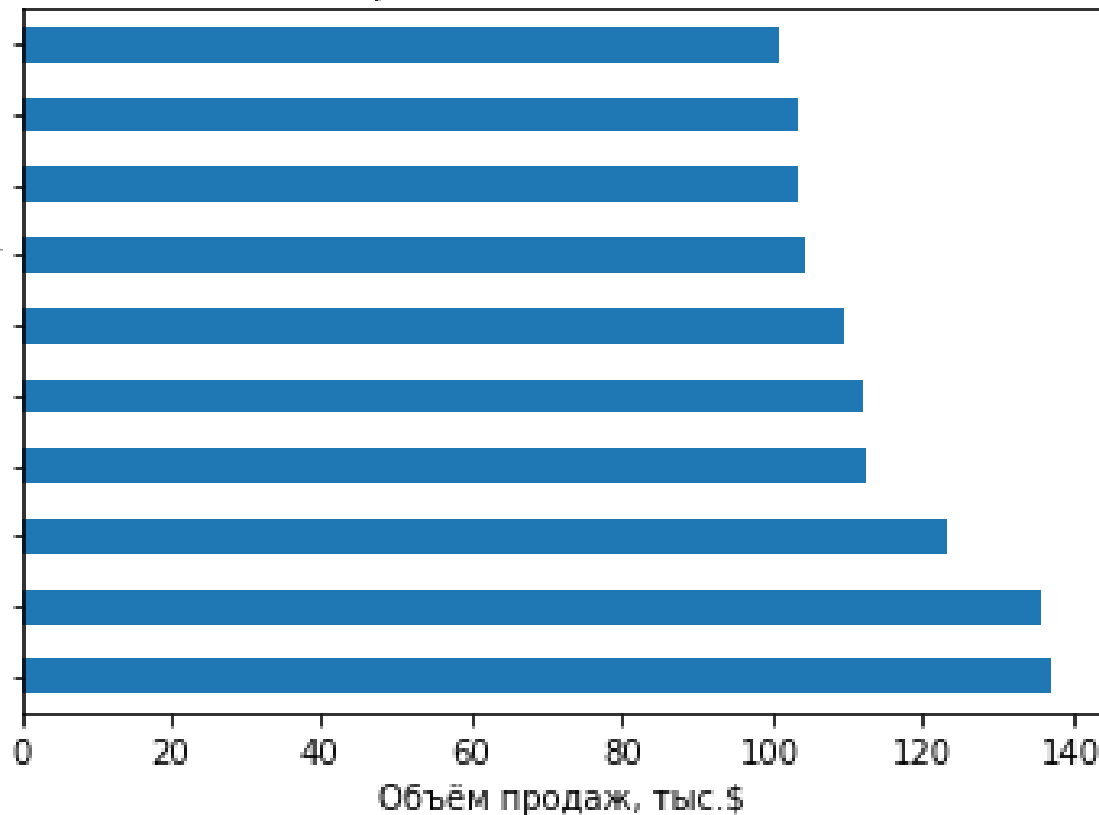
Продажи за 2017 год



```

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

```
data.plot(kind='barh', y="Sales", x="Company", ax=ax[0])
```

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

....

```
data.plot(kind='barh', y="Quantity", x="Company", ax=ax[1])
```

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

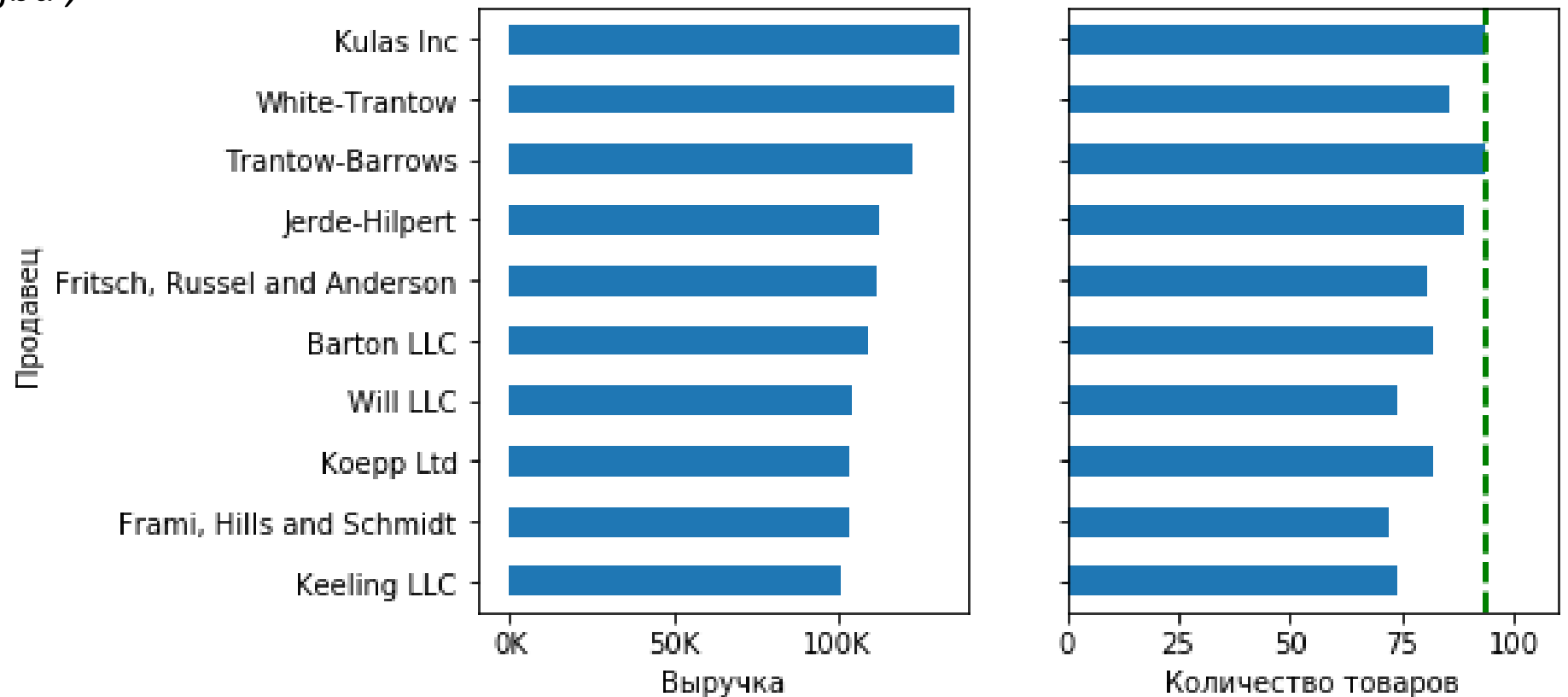
```
fig.suptitle('Анализ продаж 2017', fontsize=14, fontweight='bold');
```

Сохранение

```
fig.savefig('sales.png', transparent=False, dpi=80, bbox_inches="tight")
```

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

Анализ продаж 2017



Возможно полезные ссылки

<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

Русскоязычные источники:

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