



**Библиотеки визуализации данных**  
**Matplotlib, Seaborn, Altair.**

# Библиотеки визуализации (Visualization Libraries)

Способность создавать легко понятные, но сложные графики жизненно важна для того, чтобы стать успешным специалистом по анализу данных. Создание визуализаций - отличный способ рассказать историю, лежащую в основе ваших данных. Визуализации выделяют взаимосвязи в данных и раскрывают информацию, видимую человеческому глазу, которую нельзя передать только числами и цифрами.

Визуализации можно создавать несколькими способами:

- Matplotlib
- Pandas (через Matplotlib)
- Seaborn
- Statistically-focused plotting methods
- Global preferences incorporated by Matplotlib
- Plotly, Plotly Express и др.

# Basic Scatter Plots with Matplotlib

Scatter plots могут быть созданы из Pandas Series (Scatter plots can be created from Pandas Series)

## Code

```
Import matplotlib.pyplot as plt
```

```
plt.plot(data.sepal_length,  
         data.sepal_width,  
         ls=' ', marker='o')
```

## Output

# Basic Scatter Plots with Matplotlib

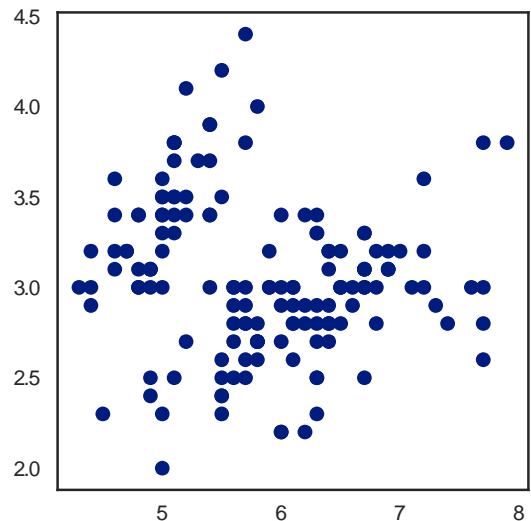
Scatter plots могут быть созданы из Pandas Series (Scatter plots can be created from Pandas Series)

## Code

```
Import matplotlib.pyplot as plt
```

```
plt.plot(data.sepal_length,  
         data.sepal_width,  
         ls='', marker='o')
```

## Output



# Basic Scatter Plots with Matplotlib

Также можно добавить несколько слоев данных (Multiple layers of data can also be added)

## Code

```
plt.plot(data.sepal_length,  
         data.sepal_width,  
         ls='', marker='o',  
         label='sepal')  
  
plt.plot(data.petal_length,  
         data.petal_width,  
         ls='', marker='o',  
         label='petal')
```

## Output

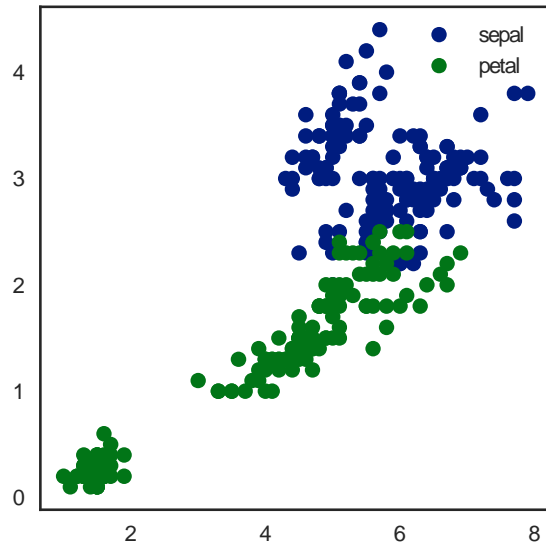
# Basic Scatter Plots with Matplotlib

Также можно добавить несколько слоев данных (Multiple layers of data can also be added)

## Code

```
plt.plot(data.sepal_length,  
         data.sepal_width,  
         ls='', marker='o',  
         label='sepal')  
  
plt.plot(data.petal_length,  
         data.petal_width,  
         ls='', marker='o',  
         label='petal')
```

## Output



# Histograms with Matplotlib

Создание гистограмм (Histograms can be created from Pandas Series)

Code

```
plt.hist(data.sepal_length, bins=25)
```

Output



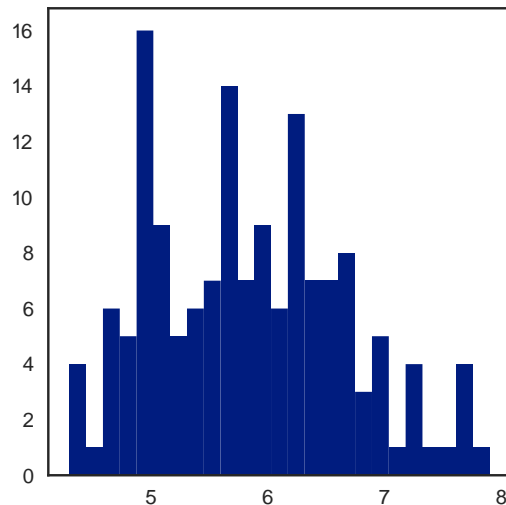
# Histograms with Matplotlib

Histograms can be created from Pandas Series

## Code

```
plt.hist(data.sepal_length, bins=25)
```

## Output



# Настройка (Customizing) Matplotlib Plots

Каждую функцию графиков Matplotlib можно настроить (Every feature of Matplotlib plots can be customized)

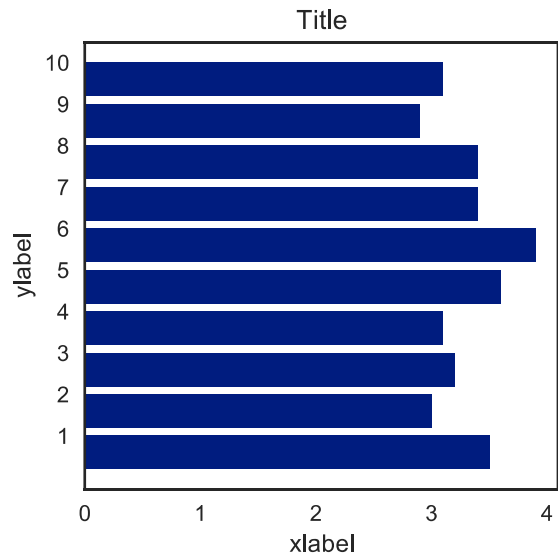
## Code

```
fig, ax = plt.subplots()

ax.barh(np.arange(10),
        data.sepal_width.iloc[:10])

# Set position of ticks and tick labels
ax.set_yticks(np.arange(0.4, 10.4, 1.0))
ax.set_yticklabels(np.arange(1, 11))
ax.set(xlabel='xlabel', ylabel='ylabel',
       title='Title')
```

## Output



# Использование статистических вычислений (Incorporating Statistical Calculations)

Статистические расчеты могут быть включены с помощью методов Pandas  
(Statistical calculations can be included with Pandas methods)

Code

```
(data
 .groupby('species')
 .mean()
 .plot(color=['red', 'blue',
             'black', 'green'],
       fontsize=10.0, figsize=(4,4)))
```

Output

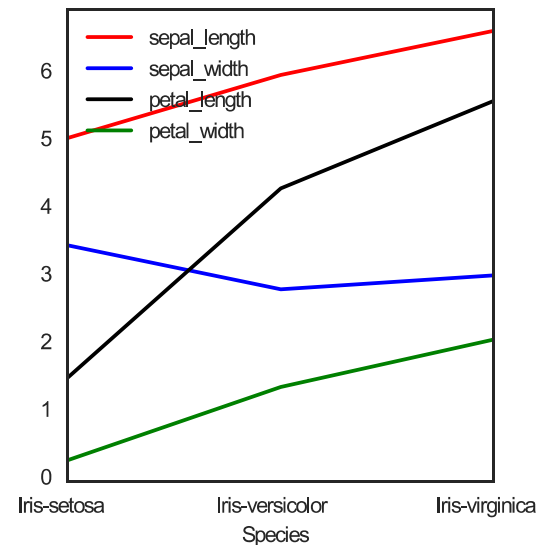
# Incorporating Statistical Calculations

Statistical calculations can be included with Pandas methods

## Code

```
(data
.groupby('species')
.mean()
.plot(color=['red','blue',
            'black','green'],
      fontsize=10.0, figsize=(4,4)))
```

## Output



# Statistical Plotting with Seaborn

Создание визуализаций совместного распределения и диаграмм рассеяния в seaborn (Joint distribution and scatter plots can be created)

Code

```
import seaborn as sns
```

```
sns.jointplot(x='sepal_length',  
             y='sepal_width',  
             data=data, size=4)
```

Output

# Statistical Plotting with Seaborn

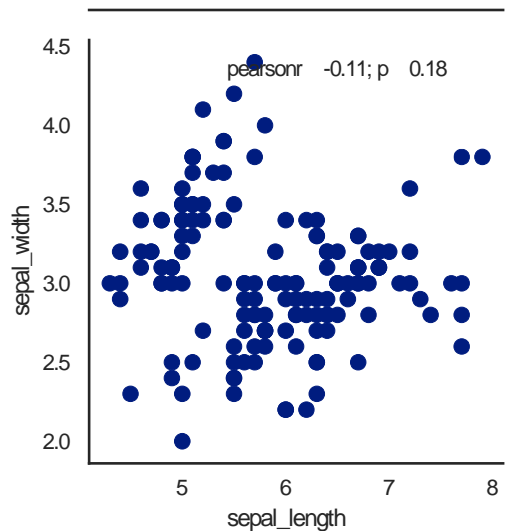
Создание визуализаций совместного распределения и диаграмм рассеяния в seaborn (Joint distribution and scatter plots can be created)

## Code

```
import seaborn as sns
```

```
sns.jointplot(x='sepal_length',  
              y='sepal_width',  
              data=data, size=4)
```

## Output



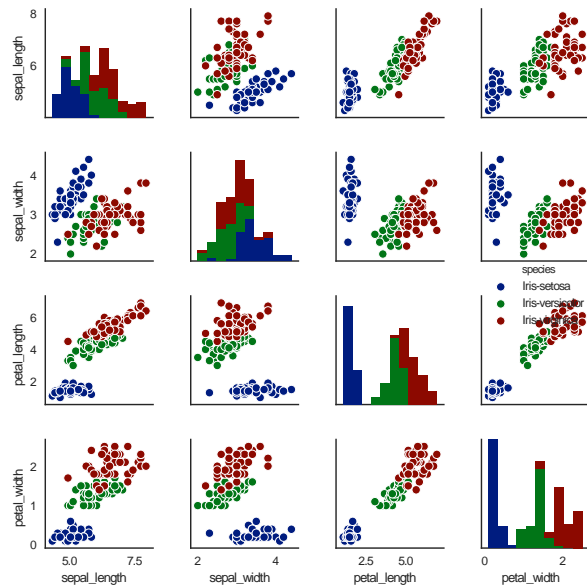
# Statistical Plotting with Seaborn

Графики попарной корреляции всех переменных в Seaborn (Correlation plots of all variable pairs can also be made with Seaborn)

## Code

```
sns.pairplot(data, hue='species', size=3)
```

## Output





# Altair

Altair - это библиотека Python, предназначенная для статистической визуализации. Оно носит декларативный характер

Как специалисту по анализу данным, Altair позволит вам сосредоточить свое время и больше усилий на ваших данных - на их понимании, анализе и визуализации, а не на коде, необходимом для этого. Это означает, что вы можете определить данные и результат, который вы ожидаете увидеть (как должна выглядеть визуализация в конце), и Альтаир автоматически выполнит необходимые манипуляции за вас.

Mark Name	Method	Description
area	<code>mark_area()</code>	A filled area plot.
bar	<code>mark_bar()</code>	A bar plot.
circle	<code>mark_circle()</code>	A scatter plot with filled circles.
geoshape	<code>mark_geoshape()</code>	A geographic shape
image	<code>mark_image()</code>	A scatter plot with image markers.
line	<code>mark_line()</code>	A line plot.
point	<code>mark_point()</code>	A scatter plot with configurable point shapes.
rect	<code>mark_rect()</code>	A filled rectangle, used for heatmaps
rule	<code>mark_rule()</code>	A vertical or horizontal line spanning the axis.
square	<code>mark_square()</code>	A scatter plot with filled squares.
text	<code>mark_text()</code>	A scatter plot with points represented by text.
tick	<code>mark_tick()</code>	A vertical or horizontal tick mark.



# Altair in Python: Data Visualizations

Визуализации данных в Python с помощью Altair

## Code

```
!pip install altair vega_datasets
import pandas as pd
import altair as alt

# Importing the Vega Dataset
from vega_datasets import data as vega_data

movies_df = pd.read_json(vega_data.movies.url)

# Checking the type of data that we get
print("movies_df is of the type: ", type(movies_df))

print("movies_df: ",
      movies_df.shape)
```

## Output

```
movies_df is of the type: <class
'pandas.core.frame.DataFrame'>
```

```
movies_df: (3201, 16)
```

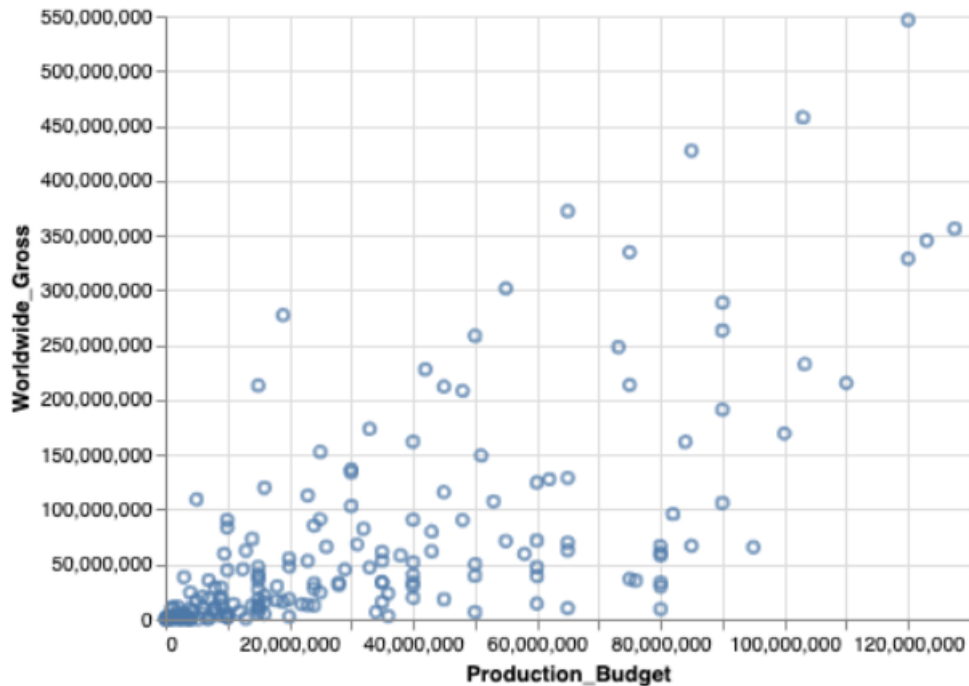
# Altair in Python: Data Visualizations

Визуализации данных в Python с помощью Altair

## Code

```
alt.Chart(movies_2000).mark_  
point().encode(  
  alt.X('Production_Budget'),  
  alt.Y('Worldwide_Gross') )
```

## Output



# Altair in Python: Data Visualizations

Визуализации данных в Python с помощью Altair

Code

```
alt.Chart(movies_2000).mark_point(
    filled=True).encode(
    alt.X('Production_Budget'),
    alt.Y('Worldwide_Gross'),
    alt.Size('US_Gross'),
    alt.Color('Major_Genre'),
    alt.OpacityValue(0.7), tooltip =
    [alt.Tooltip('Title'),
    alt.Tooltip('Production_Budget'),
    alt.Tooltip('Worldwide_Gross'),
    alt.Tooltip('US_Gross') ]
).interactive()
```

Output

