

# Библиотека seaborn

## Получение доступных стилей

```
In import matplotlib.pyplot as plt
print(plt.style.available) # вызовем разные наборы цветов
```

## Выполнить построения с применением стиля

```
In with plt.style.context('seaborn-pastel'):
    # ваш код
```

## Использовать стиль для всех графиков

```
In plt.style.use('ggplot') # здесь выбран стиль ggplot
```

## Построение joint plot

```
In import seaborn as sns
import pandas as pd

sns.jointplot(
    x="column1",
    y="column2",
    data=df,
    kind='reg', # дополнительное построение плотности распределения и регрессии
    color='blue' # цвет графика
```

## Просмотр цветовой палитры

```
In current_palette = sns.color_palette("coolwarm", 20)
print(sns.palplot(current_palette))
```

## Установка цветовой палитры

```
In sns.set_palette('dark')
```

## Установка стиля

```
In sns.set_style("dark")

# Варианты параметра:
# 'darkgrid' (по умолчанию);
# 'dark', 'white' или 'ticks' (не рисуют сетку);
# 'whitegrid' (подойдёт для сложных графиков).
```

## Загрузка встроенного набора данных

```
In import seaborn as sns
iris = sns.load_dataset("iris")
print(iris.head())
```

## Построение столбчатой диаграммы

```
In import seaborn as sns
ax = sns.barplot(x="column1", y="column2", data=df, estimator=median)
# estimator – метод агрегации данных
```

### Построение ящика с усами

```
In import seaborn as sns

ax = sns.boxplot(x="column1", y="column2", hue="diet", data=df)
# hue – дополнительный параметр, "третье" измерение
```

### Построение линейного графика и гистограмму плотности распределения

```
In import seaborn as sns

sns.distplot(df['column'], bins=10)
# bins – число корзин
```

### График попарного распределения

```
In sns.pairplot(df, hue="column3")
# hue – дополнительный параметр, "третье" измерение
```

### Построение Violin Plot

```
In sns.violinplot(x="column1", y="column2", data=df, palette='rainbow')
# palette – стиль графика
```

### Построение strip plot - диаграммы рассеяния для каждой категории

```
In sns.stripplot(x="column1", y="column2", data=df)
```