

# Отчёт по графикам

Гвоздев П.А.

23 Мая 2023

```
article ucs [T2A]fontenc [utf8x]inputenc [russian, english]babel listings graph-
icx amsmath Гвоздев П.А. SciPy 27 Мая 2023
article ucs [T2A]fontenc [utf8x]inputenc [russian, english]babel listings graph-
icx amsmath Гвоздев П.А. Отчёт "Визуализация данных" ФН1-21Б
xcolor
codegreenrgb0,0.6,0 codegrayrgb0.5,0.5,0.5 codepurplergb0.58,0,0.82 back-
colourrgb0.95,0.95,0.92 mystyle backgroundcolor=backcolour, commentstyle=codegreen,
keywordstyle=magenta, numberstyle=codegray, stringstyle=codepurple, basicstyle=, breakatwhitespace=false,
breaklines=true, captionpos=b, keepspaces=true, numbers=left, numbersep=5pt, showspaces=false,
showstringspaces=false, showtabs=false, tabsize=2 style=mystyle
```

Для обработки я выбрал датасет на тему заработных плат "Salary Data", включающий в себя такие данные как:

Возраст, пол, уровень образования, профессия, опыт работы, заработная плата.

Импортируем необходимые библиотеки, удалим лишнее, исправим опечатку, устанавливаем тему для графиков. [language=Python] import numpy as np  
import pandas as pd import matplotlib.pyplot as plt from mpl\_toolkits.mplot3d import Axes3D import seaborn

```
df=pd.read_csv('SalaryData.csv')df.dropna(inplace = True)df.drop(df.index[df['Gender'] == 'Other'],inplace = True)df['EducationLevel'] = df['EducationLevel'].apply(lambda x :  
x.replace('phD',' PhD'))sns.set_theme()
```

Далее построим график распределения заработной платы в зависимости от опыта работы.

```
[language=Python] plt.figure(figsize=(5,5)) sns.scatterplot(df,x='Years of Ex-  
perience',y='Salary') plt.show()  
[width=0.8]graph1.png
```

График зависимости заработной платы от возраста.

```
[language=Python] plt.figure(figsize=(5,5)) sns.lineplot(df,x='Age',y='Salary')  
plt.show()  
[width=0.8]graph2.png
```

Построим график распределения зарплат по уровням образования.

```
[language=Python] plt.figure(figsize=(5,5)) sns.stripplot(df,y='Education Level',x='Salary')
plt.show()
[width=0.8]graph3.png
```

Построим столбчатую диаграмму суммы зарплат всех специалистов с одним и тем же уровнем образования. Так как наименования столбцов не влезят сократим их названия.

```
[language=Python] df['Education Level']=df['Education Level'].apply(lambda
x: x.replace('High School','School')) df['Education Level']=df['Education Level'].apply(lambda
x: x.replace("Master's Degree",'Master D')) df['Education Level']=df['Education
Level'].apply(lambda x: x.replace("Master's",'Master')) df['Education Level']=df['Education
Level'].apply(lambda x: x.replace("Bachelor's Degree",'Bach D')) df['Education
Level']=df['Education Level'].apply(lambda x: x.replace("Bachelor's",'Bachelor'))
plt.figure(figsize=(5,5)) sns.barplot(df,x='Education Level',y='Salary') plt.show()
[width=0.8]graph4.png
```

Построим столбчатую диаграмму количества специалистов различных уровней образования.

```
[language=Python] plt.figure(figsize=(5,5)) sns.countplot(df,x='Education Level')  
plt.show()  
[width=0.8]graph5.png
```

Построим pairplot по гендерам.

```
[language=Python] plt.figure(figsize=(5,5)) sns.pairplot(df,hue='Gender')  
[width=0.8]graph6.png
```

Построим график эмпирической кумулятивной функции распределения опыта работы.

```
[language=Python] plt.figure(figsize=(5,5)) sns.ecdfplot(data=df, x="Years of Experience") plt.show()
```

[width=0.8]graph7.png

Построим усиковую диаграмму распределения зарплаты по уровням образования.

```
[language=Python] plt.figure(figsize=(5,5)) sns.boxplot(x='Education Level', y='Salary',  
data=df) plt.show()  
[width=0.8]graph8.png
```

Построим графики распределения зарплат в зависимости от пола и уровня образования.

```
[language=Python] plt.figure(figsize=(5,5)) sns.violinplot(x=df['Education Level'],  
y=df['Salary'], hue=df['Gender'], split=True) plt.show()  
[width=0.8]graph9.png
```



Построим histplot для самой популярной профессии- "Data Analyst".

```
[language=Python] plt.figure(figsize=(5,5)) df1=df[df['Job Title']=='Data Analyst'] sns.histplot(df1['Age'], kde=True, stat='count') plt.show()
```

[width=0.8]graph10.png