

Министерство образования Республики Беларусь  
УО «Брестский государственный технический университет»  
Кафедра ИИТ

**Лабораторная работа №8**

По дисциплине: “Языки программирования”

Тема: «Python. Основы Pandas»

**Вариант №9**

**Выполнил:** студент 2 курса группы ПО-7  
Крупенков Михаил Дмитриевич

**Проверила:** Дряпко А. В.

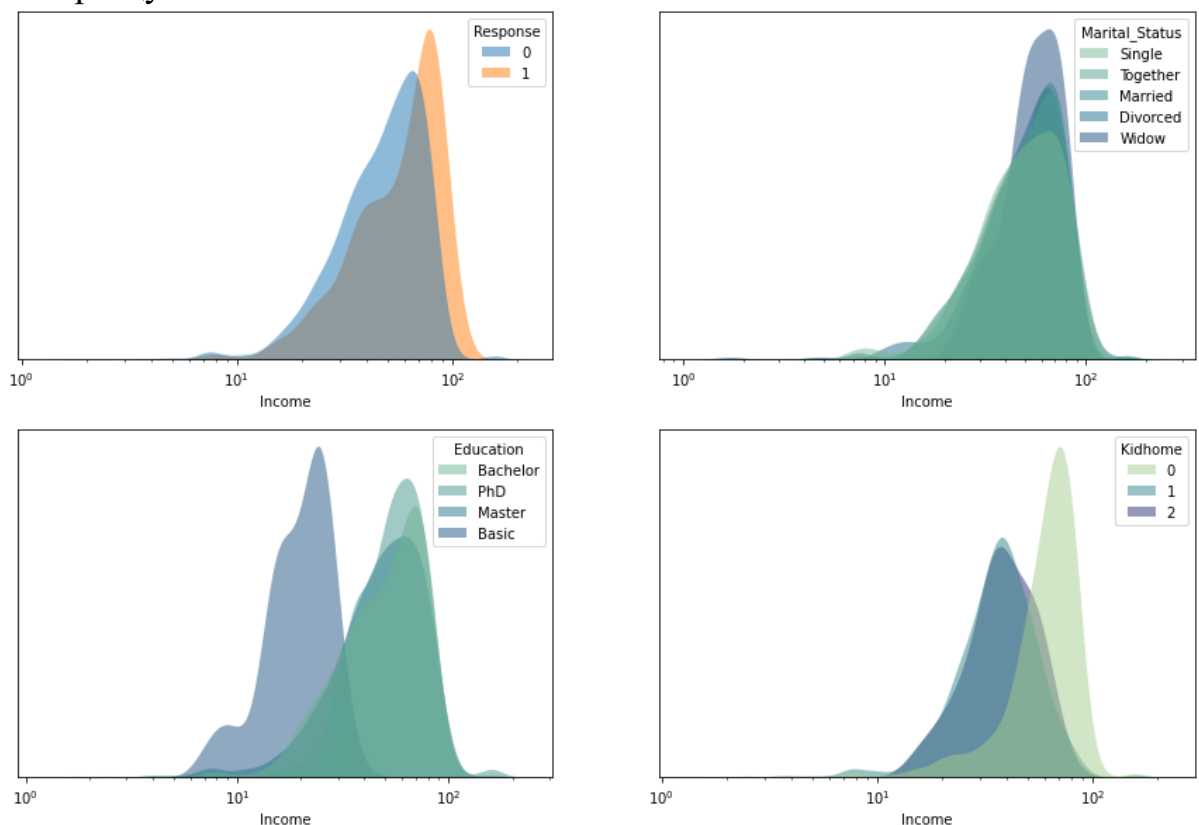
Брест 2021

## Цель работы

Ознакомиться с основами библиотеки pandas и научиться строить графики с использованием библиотек matplotlib.pyplot и seaborn.

## Общие требования

1. Загрузить датасет в pandas и проверить на доступность
2. Вывести общую информацию о датасете
3. Проверка наличия NULL-данных. При их наличии вывести на экран
4. Удалить колонки "Z\_CostContact", "Z\_Revenue"
5. Переименовать колонку "Year\_Birth" в "Age"
6. Оценить состояние колонок "Marital\_Status", "Education". Построить информативные диаграммы и гистограммы для каждой.
7. Создать гистограмму по колонке "Age" и оценить на распределение по Гауссу.
8. Оценка полей "Kidhome" и "Teenhome", "Response" и "Income" (диаграммы и гистограммы)
9. Построить графики "Response", "Marital\_Status", "Education" и "Kidhome" пообразцу



10. Построить heatmap для всех числовых колонок:

### Задание 1

```
df = pd.read_csv("dataset.csv", delimiter="\t")
```

### Задание 2

df.info()

```
1 #task 2
2 df.info()
```

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 2240 entries, 0 to 2239  
Data columns (total 29 columns):  
# Column Non-Null Count Dtype  
--- --  
0 ID 2240 non-null int64  
1 Year\_Birth 2240 non-null int64  
2 Education 2240 non-null object  
3 Marital\_Status 2240 non-null object  
4 Income 2216 non-null float64  
5 Kidhome 2240 non-null int64  
6 Teenhome 2240 non-null int64

### Задание 3

df[df.isnull().T.any()]

```
1 #task 3
2 df[df.isnull().T.any()]
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	F
10	1994	1983	Graduation	Married	NaN	1	0	15-11-2013	
27	5255	1986	Graduation	Single	NaN	1	0	20-02-2013	
43	7281	1959	PhD	Single	NaN	0	0	05-11-2013	
48	7244	1951	Graduation	Single	NaN	2	1	01-01-2014	
58	8557	1982	Graduation	Single	NaN	1	0	17-06-2013	
71	10629	1973	2n Cycle	Married	NaN	1	0	14-09-2012	
90	8996	1957	PhD	Married	NaN	2	1	19-11-2012	
91	9235	1957	Graduation	Single	NaN	1	1	27-05-2014	
92	5798	1973	Master	Together	NaN	0	0	23-11-2013	

24 rows × 29 columns [Open in new tab](#)

### Задание 4

df.drop(["Z\_CostContact", "Z\_Revenue"], axis=1, inplace=True)

### Задание 5

df.rename({"Year\_Birth": "Age"}, axis=1, inplace=True)

### Задание 6

ms\_df\_c = df["Marital\_Status"].value\_counts().to\_frame()

ms\_df = ms\_df\_c[ms\_df\_c["Marital\_Status"] > 50]

ed\_df = df["Education"].value\_counts().to\_frame()

ms\_df[ms\_df["Marital\_Status"] > 50].plot(

kind='pie',

autopct='%1.1f%%',

legend=None,

ylabel="",

subplots=True,

shadow=True,

explode=(0.13, 0.1, 0.1, 0.25, 0.15),

```

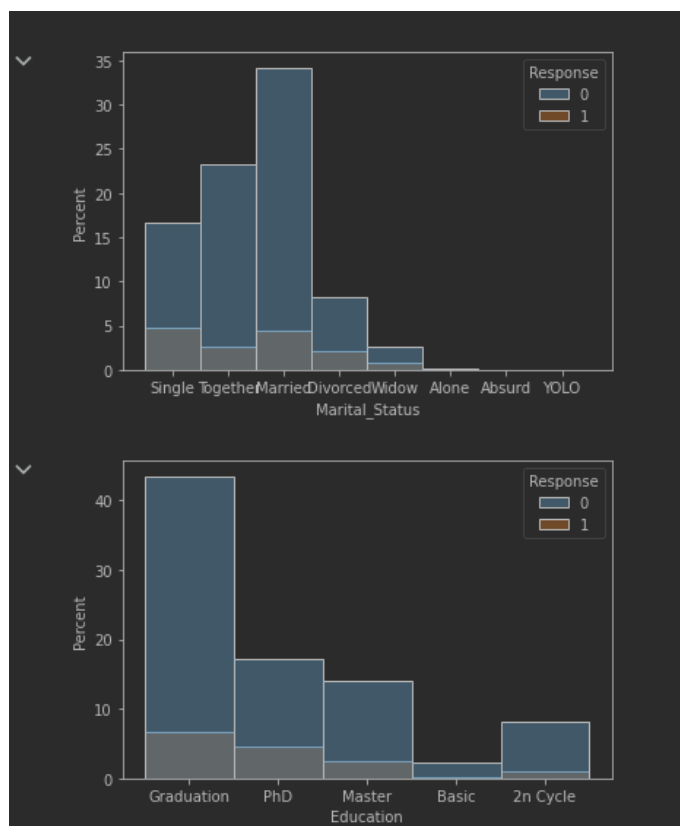
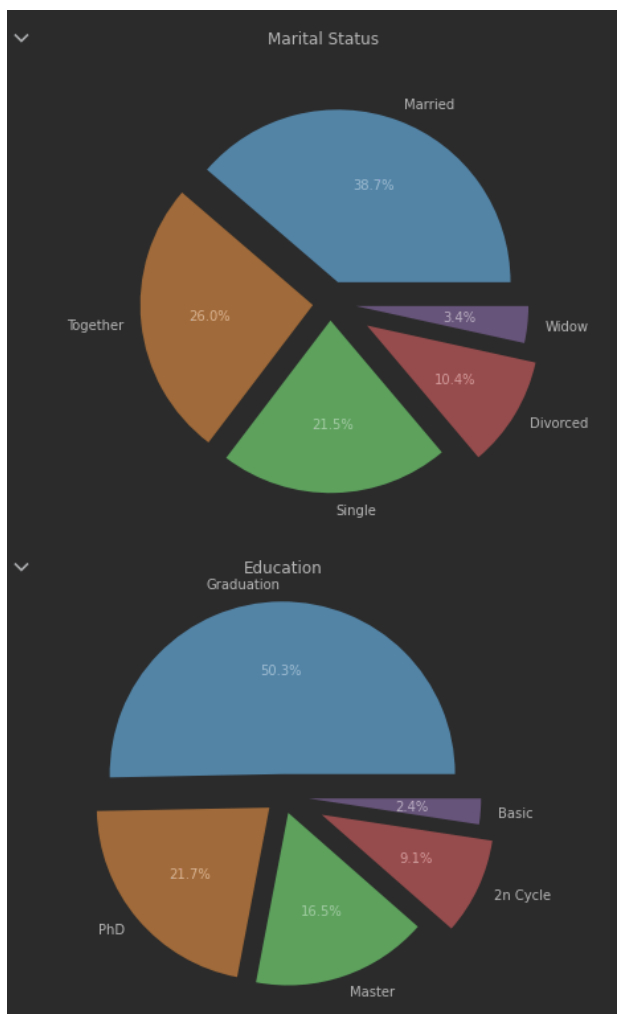
    title="Marital Status",
    figsize = (6,6)
)
plt.show()

ed_df.plot(
    y="Education",
    kind='pie',
    autopct='% 1.1f%%',
    legend=None,
    ylabel="",
    shadow=True,
    explode=(0.13, 0.1, 0.1, 0.25, 0.15),
    title="Education",
    figsize = (6,6)

)
plt.show()

sns.histplot(data=df, x="Marital_Status", stat='percent', hue="Response")
plt.show()
sns.histplot(data=df, x="Education", stat='percent', hue="Response")
plt.show()

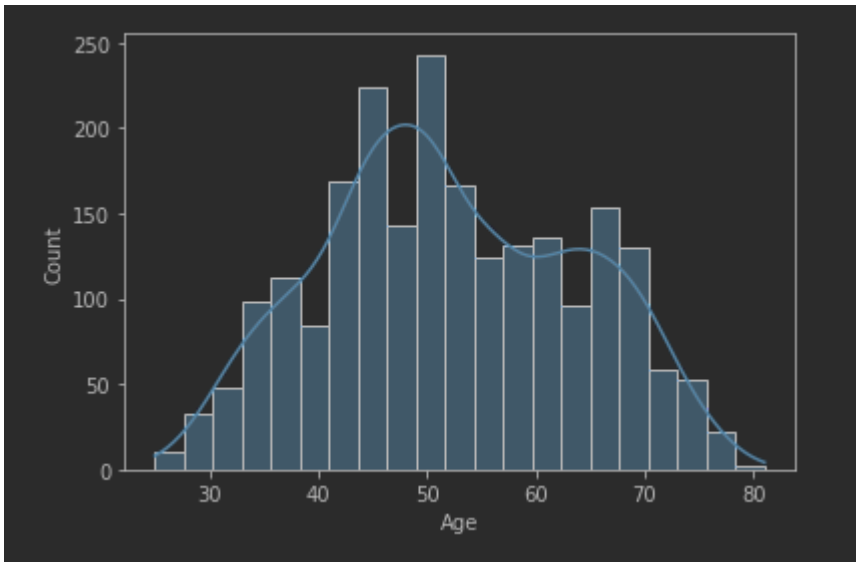
```



## Задание 7

```
def calculate_age(born) -> int:
    return int(datetime.today().strftime("%Y")) - int(born)

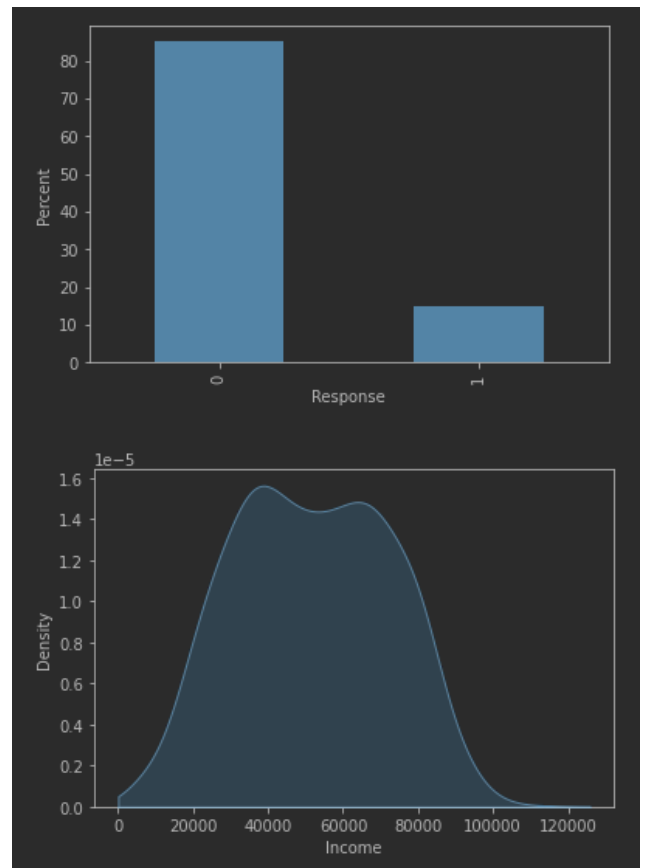
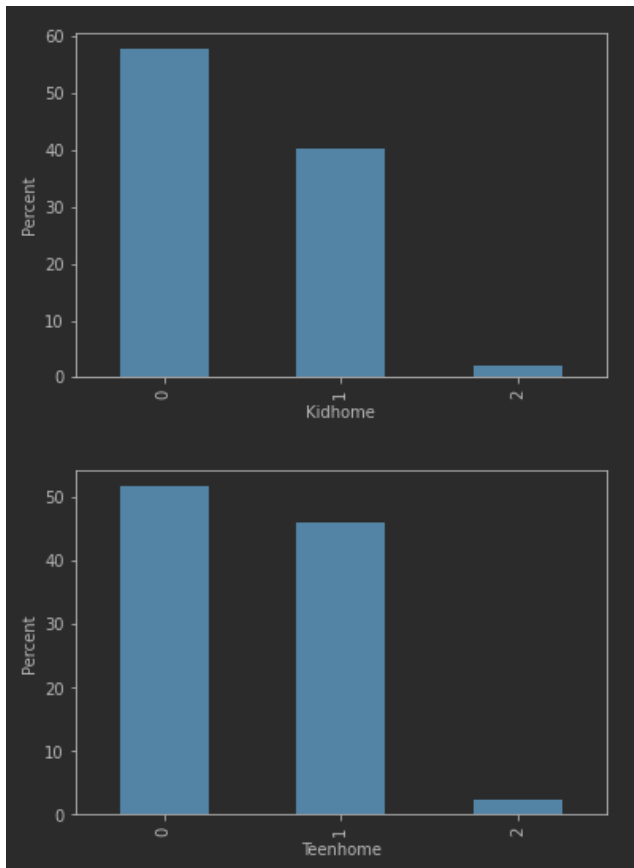
df["Age"] = df["Age"].apply(calculate_age)
filtered_df = df[df["Age"] < 100]
sns.histplot(x=filtered_df["Age"], kde=True)
plt.show()
```



## Задание 8

```
def show_bar(data_name: str) -> None:
    df[data_name].value_counts(normalize=True).mul(100).to_frame().plot(
        kind='bar',
        legend=None,
        xlabel=data_name,
        ylabel="Percent"
    )
    plt.show()

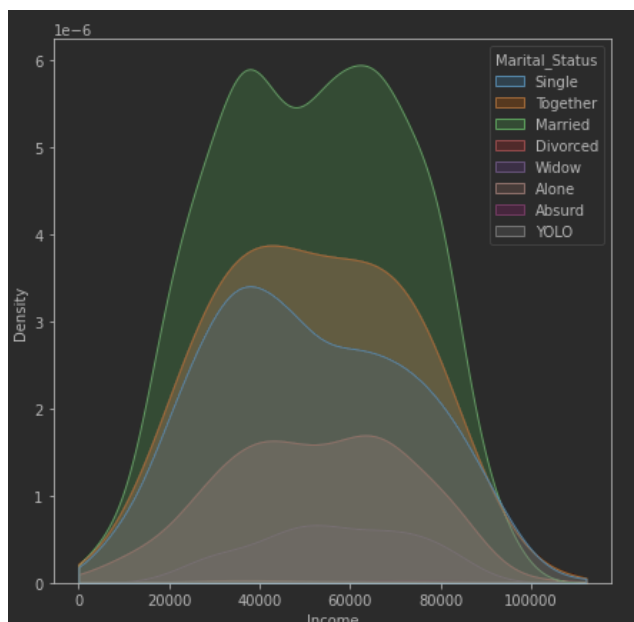
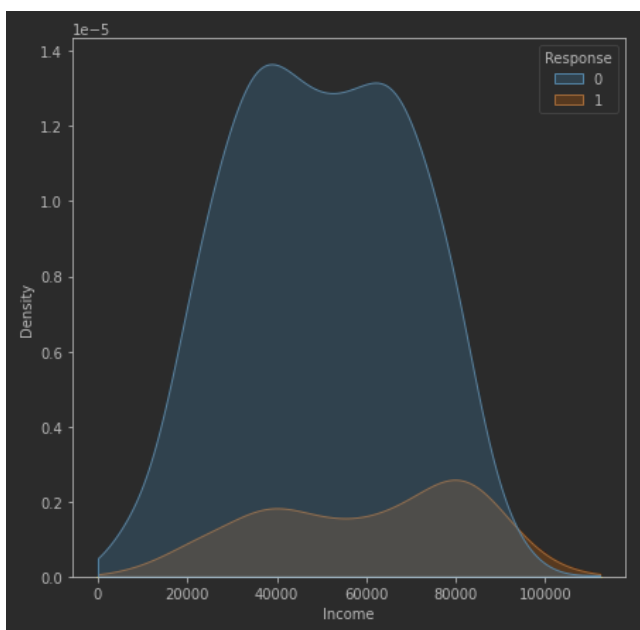
show_bar("Kidhome")
show_bar("Teenhome")
show_bar("Response")
sns.kdeplot(df["Income"], shade=True, clip=(10**0, 10**5.1))
plt.show()
```

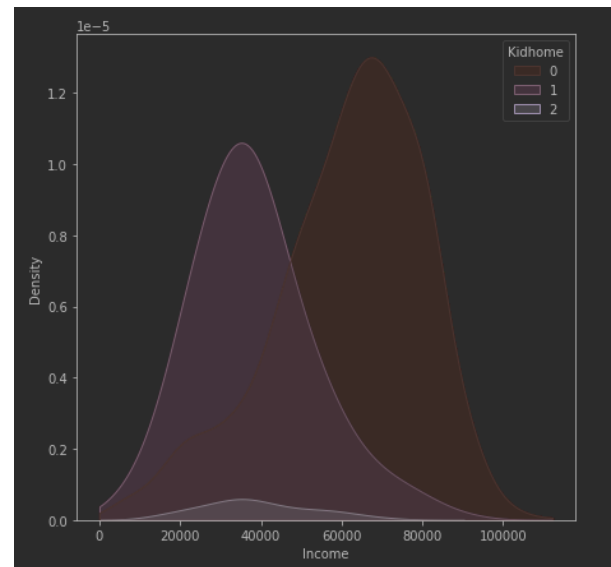
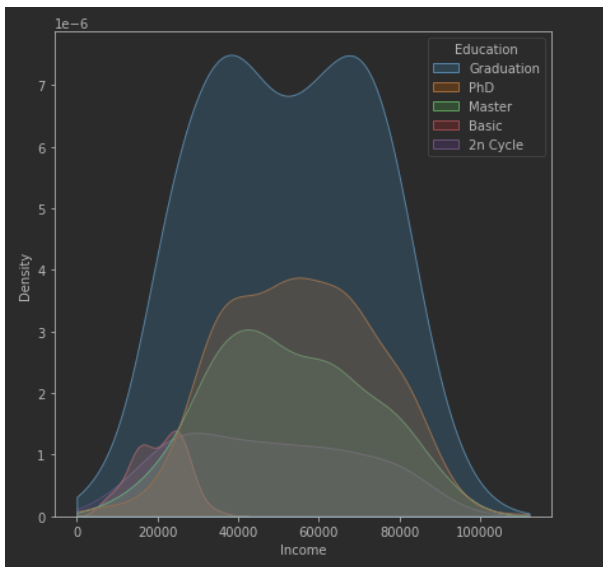


## Задание 9

```
def show_kde(hue: str, clip: tuple = (10**0, 10**5.05)):
    plt.figure(figsize = (7, 7))
    sns.kdeplot(data=df, x="Income", hue=hue, shade=True, clip=clip)
    plt.show()
```

```
show_kde("Response")
show_kde("Marital_Status")
show_kde("Education")
show_kde("Kidhome")
```



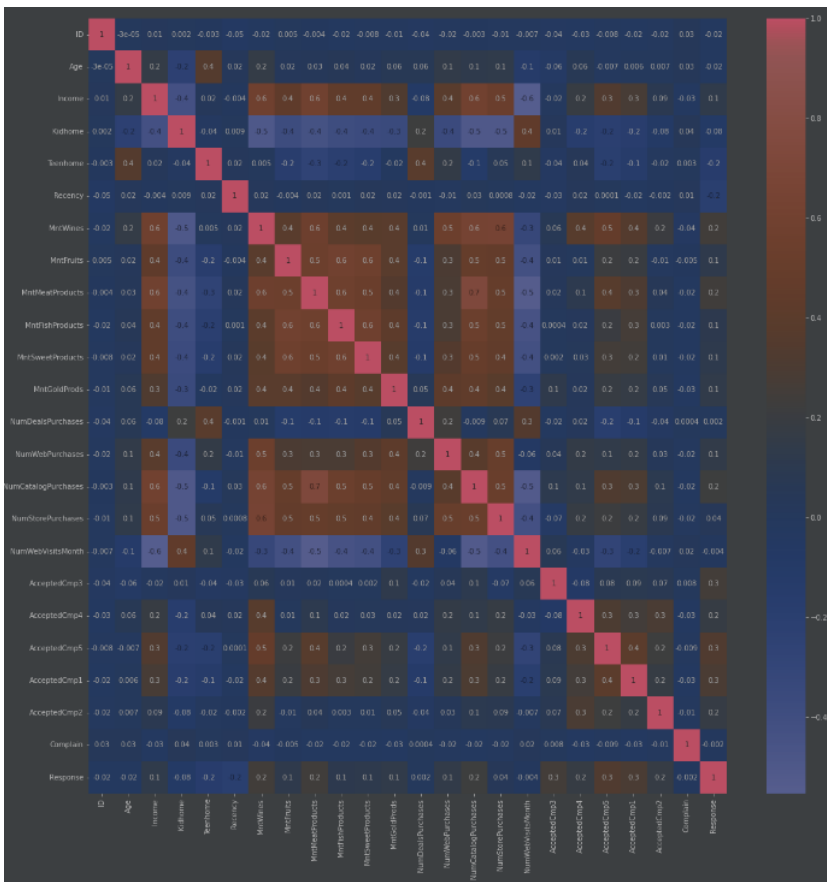


## Задание 10

```

numerics = ['int16', 'int32', 'int64', 'float16', 'float32', 'float64']
newdf = df.select_dtypes(include=numerics)
plt.figure(figsize = (20,20))
sns.heatmap(newdf.corr(), cmap='coolwarm', annot=True, fmt='.1g')
plt.show()

```



**Вывод:** Я преисполнился