Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №11

По дисциплине: «Языки программирования»

Тема: **«Python. Основы Pandas»**

Вариант №3

Выполнил:

Студент 2-го курса

Группы ПО-7

Гункевич И.А.

Проверила:

Дряпко А.В.

Брест 2021

**Цель работы:** ознакомиться с основами библиотеки pandas и научиться строить графики с использованием библиотек matplotlib.pyplot и seaborn.

**Ход работы:**

1. Загрузить датасет в pandas и проверить на доступность

2. Вывести общую информацию о датасете

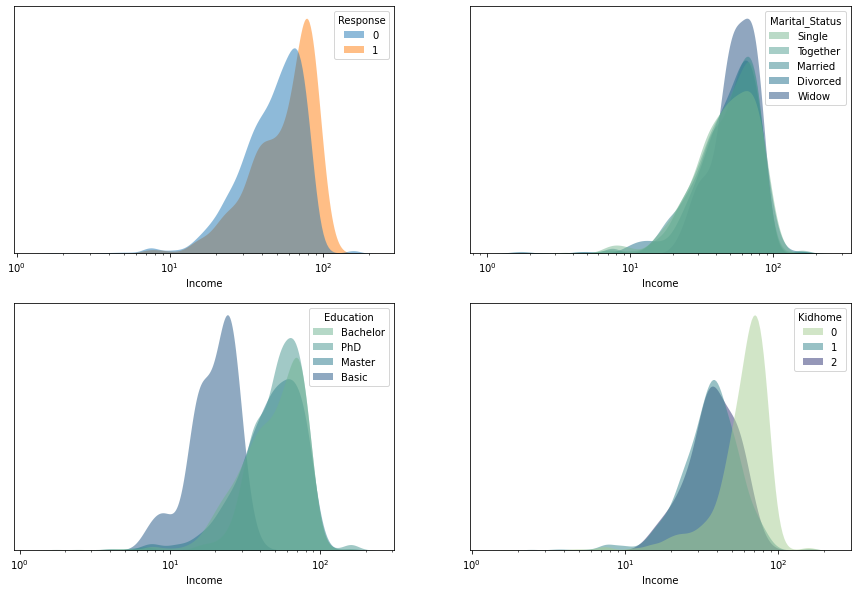
3. Проверка наличия NULL-данных. При их наличии вывести на экран

4. Удалить колонки "Z\_CostContact", "Z\_Revenue"

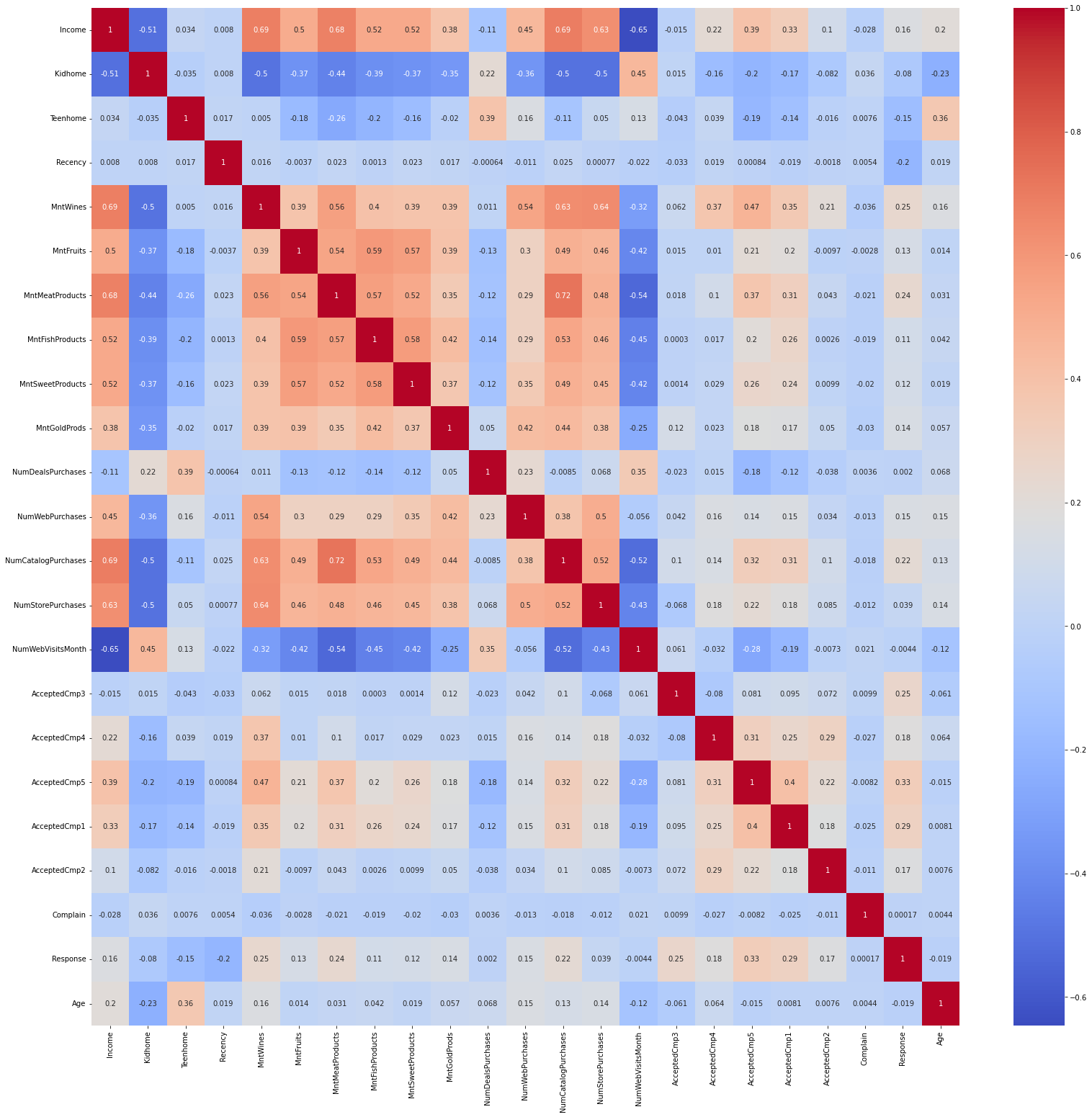
5. Переименовать колонку "Year\_Birth" в "Age"

6. Оценить состояние колонок "Marital\_Status", "Education". Построить информативные диаграммы и гистограммы для каждой.

1. Создать гистаграмму по колонке "Age" и оценить на распределение по Гауссу.
2. Оценка полей "Kidhome" и "Teenhome", "Response" и "Income" (диаграммы и гистограммы)
3. Построить графики "Response", "Marital\_Status", "Education" и "Kidhome" по образцу:



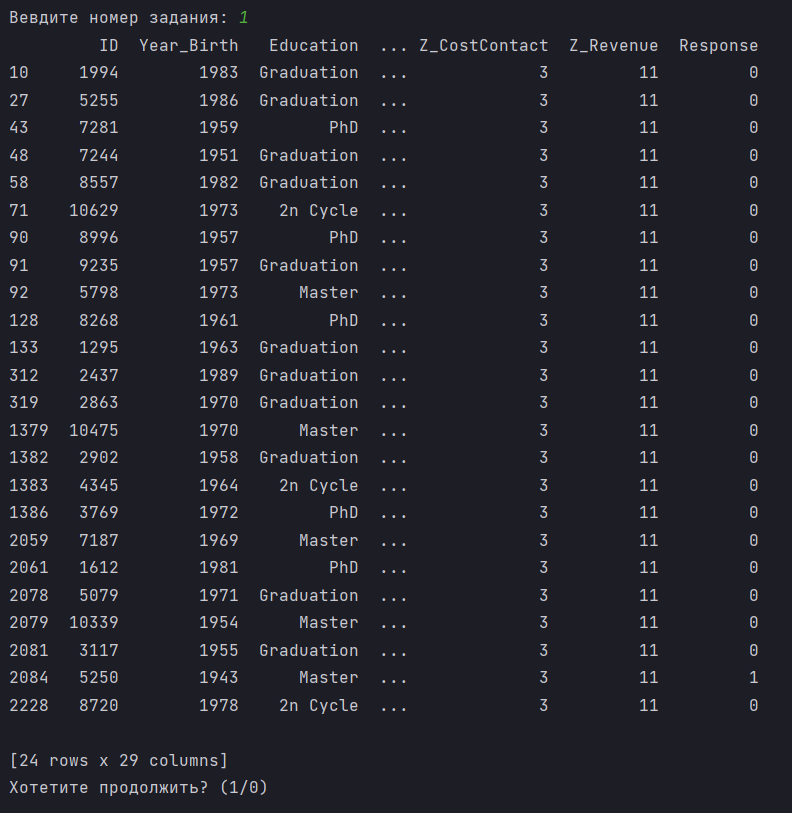
1. Построить heatmap для всех числовых колонок:

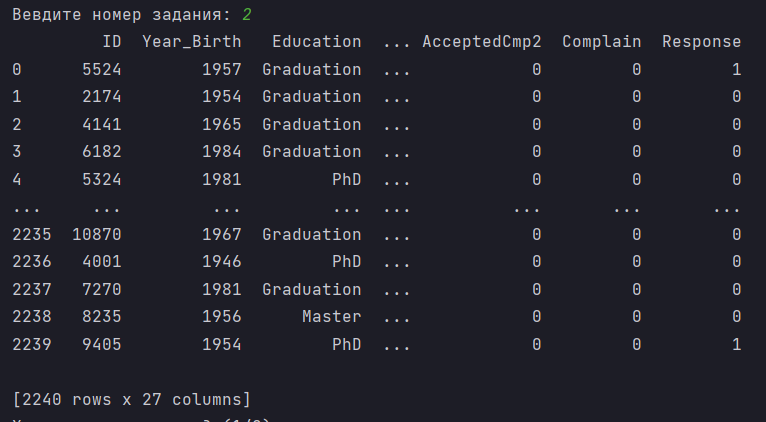


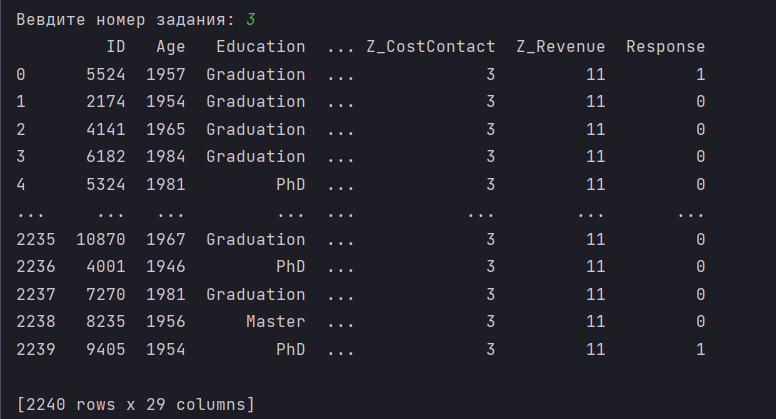
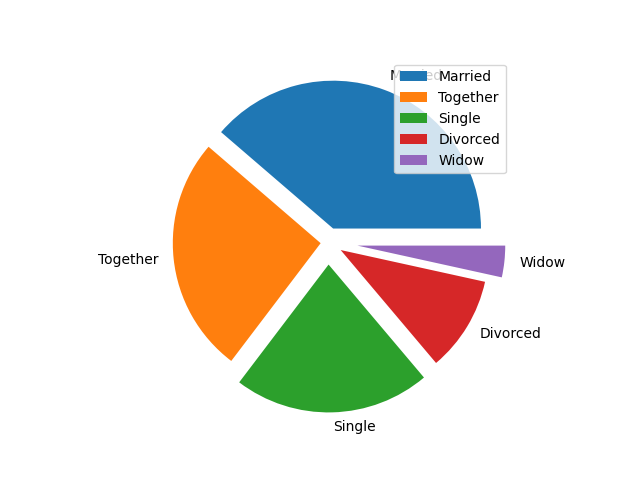
**Код программы:**

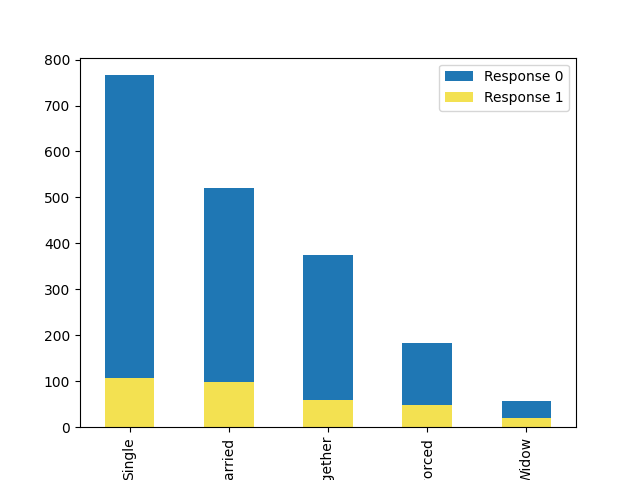
import pandas as pd  
import seaborn as sns  
import plotly.figure\_factory as ff  
import matplotlib.pyplot as plt  
from sklearn import preprocessing  
from scipy.stats import norm  
  
def main():  
 df = pd.read\_csv('marketing\_campaign.csv'**,** delimiter='\t')  
 if not df.empty:  
 print("DataFrame is accessible")  
 num\_task = -1  
 while (num\_task != 0):  
 print("|0| Выход")  
 print("|1| Задание 1")  
 print("|2| Задание 2")  
 print("|3| Задание 3")  
 print("|4| Задание 4")  
 print("|5| Задание 5")  
 print("|6| Задание 6")  
 print("|7| Задание 7")  
 print("|8| Задание 8")  
 num\_task = int(input("Вевдите номер задания: "))  
 if num\_task == 1:  
 print(df[df.isnull().any(axis=1)])  
 elif num\_task == 2:  
 print(df.drop(columns = ['Z\_CostContact'**,** 'Z\_Revenue']))  
 elif num\_task == 3:  
 df.rename(columns={'Year\_Birth': 'Age'}**,** inplace=True)  
 print(df)  
 elif num\_task == 4:  
 explode = (0.1**,** 0.05**,** 0.15**,** 0.1**,** 0.2)  
 df["Marital\_Status"].value\_counts(normalize=True).head().plot.pie(explode=explode)  
 plt.ylabel('')  
 plt.legend()  
 plt.show()  
  
 df.loc[df['Response'] == 0]["Marital\_Status"].value\_counts().head().plot.bar(label="Response 0")  
 df.loc[df['Response'] != 0]["Marital\_Status"].value\_counts().head().plot.bar(color="#f3e151"**,** label="Response 1")  
 plt.xlabel('Marital\_Status')  
 plt.legend()  
 plt.show()  
  
 df["Education"].value\_counts(normalize=True).head().plot.pie(explode=explode)  
 plt.ylabel('')  
 plt.legend()  
 plt.show()  
  
 df.loc[df['Response'] == 0]["Education"].value\_counts().head().plot.bar(label="Response 0")  
 df.loc[df['Response'] != 0]["Education"].value\_counts().head().plot.bar(color="#f3e151"**,** label="Response 1")  
 plt.xlabel('Education')  
 plt.legend()  
 plt.show()  
 elif num\_task == 5:  
 df.rename(columns={'Year\_Birth': 'Age'}**,** inplace=True)  
 data=2021-df['Age']  
 ages=data  
 mean\_plus = ages.mean()  
 mean\_minus = mean\_plus  
 sns.distplot(ages**,** fit=norm**,** kde = False)  
 plt.plot([mean\_plus**,** mean\_plus]**,**[0.0**,**norm.pdf(mean\_plus**,** ages.mean()**,** ages.std())]**,** color='black')  
 for i in range(4):  
 mean\_plus += ages.std()  
 mean\_minus -= ages.std()  
 plt.plot([mean\_plus**,** mean\_plus]**,**[0.0**,**norm.pdf(mean\_plus**,** ages.mean()**,** ages.std())]**,** color='black')  
 plt.plot([mean\_minus**,** mean\_minus]**,**[0.0**,**norm.pdf(mean\_minus**,** ages.mean()**,** ages.std())]**,** color='black')  
 plt.legend()  
 plt.show()  
 elif num\_task == 6:  
 df['Kidhome'].value\_counts().head().plot.bar()  
 plt.legend()  
 plt.show()  
 df['Teenhome'].value\_counts().head().plot.bar()  
 plt.legend()  
 plt.show()  
 df['Response'].value\_counts().head().plot.bar()  
 plt.legend()  
 plt.show()  
 df['Income'].value\_counts().head().plot.bar()  
 plt.legend()  
 plt.show()  
 df['Kidhome'].value\_counts().head().plot()  
 plt.legend()  
 plt.show()  
 df['Teenhome'].value\_counts().head().plot()  
 plt.legend()  
 plt.show()  
 df['Response'].value\_counts().head().plot()  
 plt.legend()  
 plt.show()  
 sns.kdeplot(data=df['Income']**,** shade=True)  
 plt.legend()  
 plt.show()  
 elif num\_task == 7:  
 sns.displot(df**,** x='Income'**,** hue='Response'**,** kind='kde'**,** fill=True)  
 plt.legend()  
 plt.show()  
 sns.displot(df**,** x='Income'**,** hue='Marital\_Status'**,** kind='kde'**,** fill=True)  
 plt.legend()  
 plt.show()  
 sns.displot(df**,** x='Income'**,** hue='Education'**,** kind='kde'**,** fill=True)  
 plt.legend()  
 plt.show()  
 sns.displot(df**,** x='Income'**,** hue='Kidhome'**,** kind='kde'**,** fill=True)  
 plt.legend()  
 plt.show()  
 elif num\_task == 8:  
 sns.heatmap(df.corr()**,** annot=False**,** vmin=-1**,** vmax=1**,** center=0**,** cmap='coolwarm'**,** fmt='.lg')  
 plt.legend()  
 plt.show()  
 else:  
 print("Вы ввели неправильный номер задания")   
  
 if (input("Хотетите продолжить? (1/0)\n") == '0'):   
 break  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

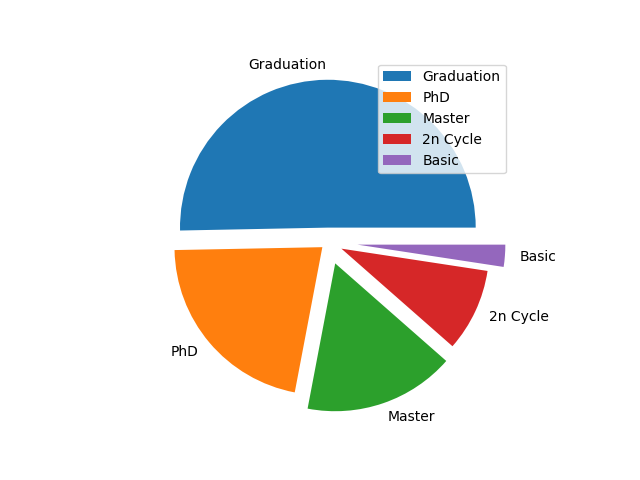
**Результаты:**

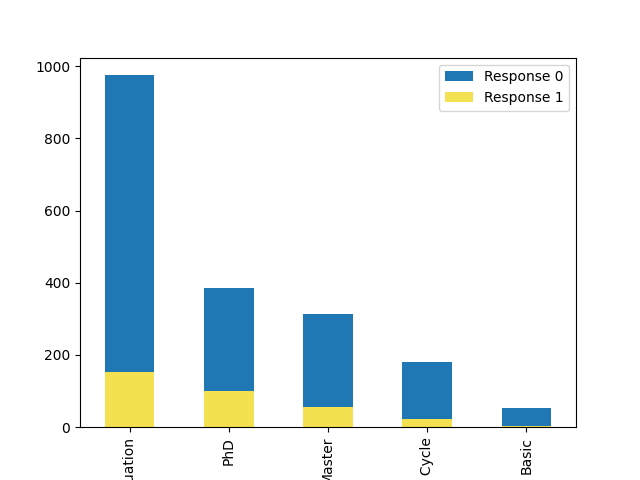


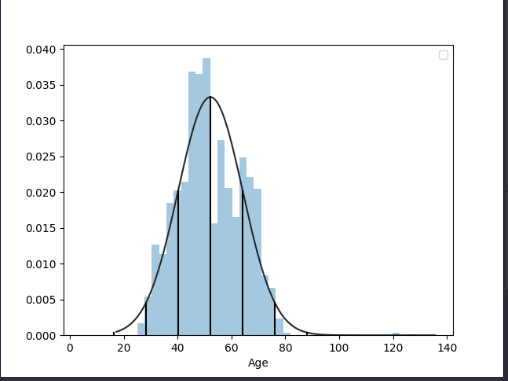


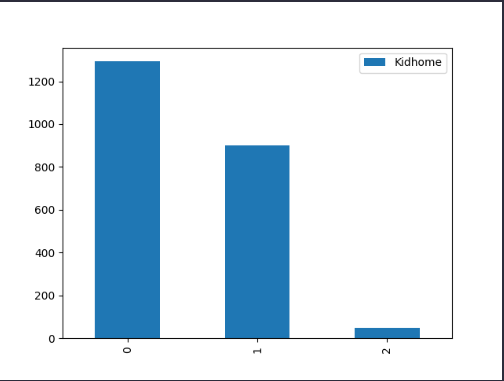
 ****

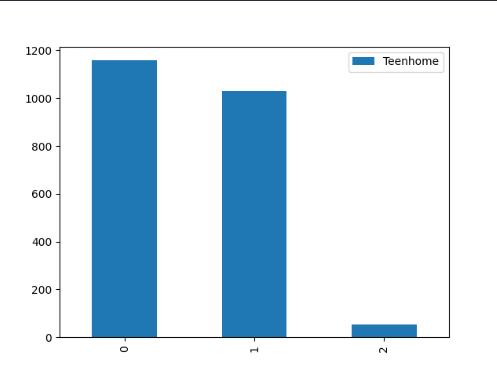
****

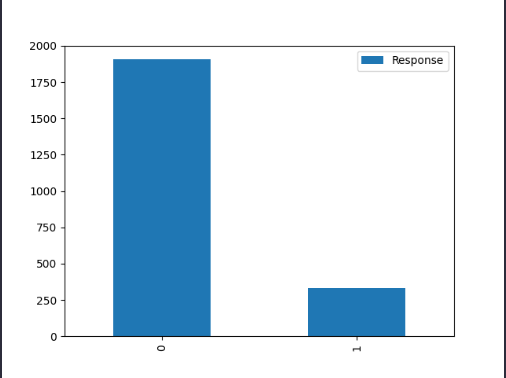
****

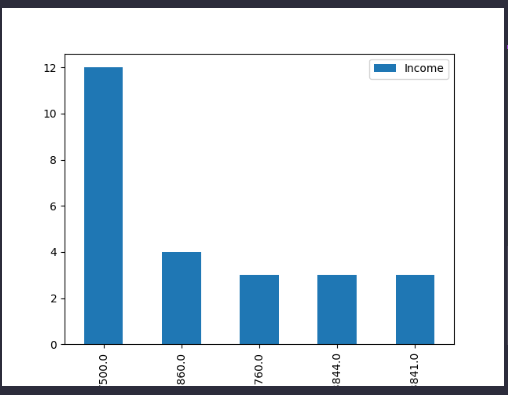
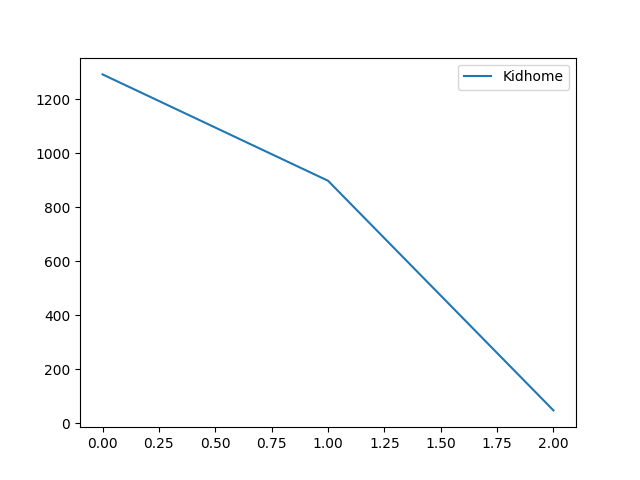
****

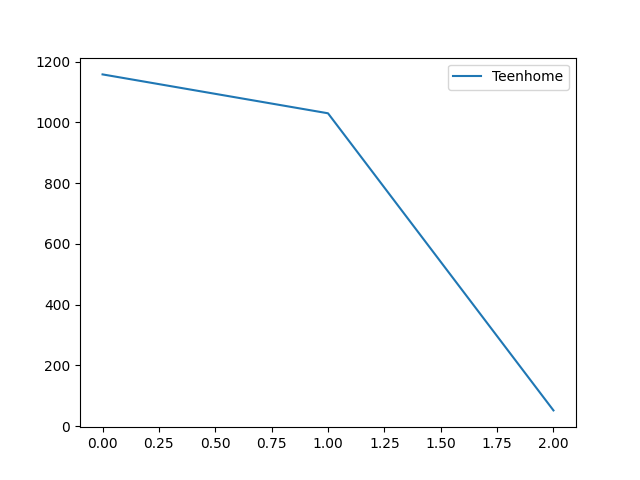


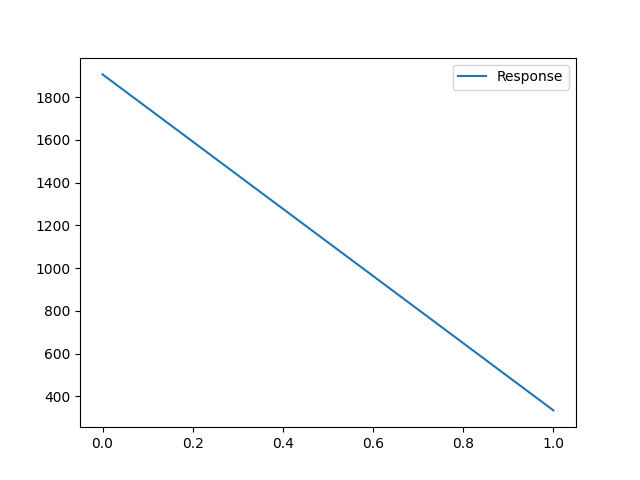
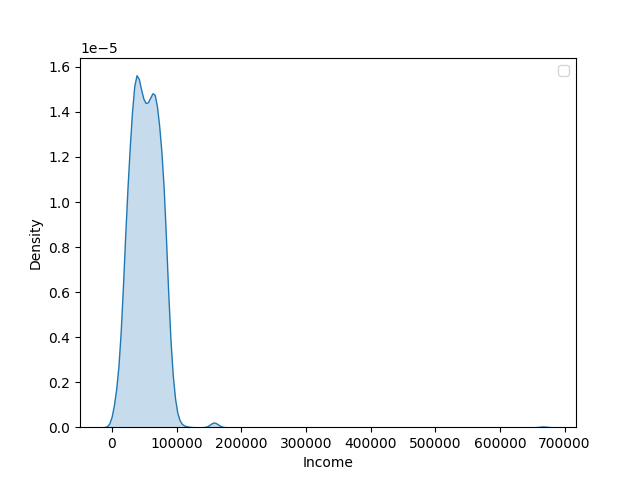


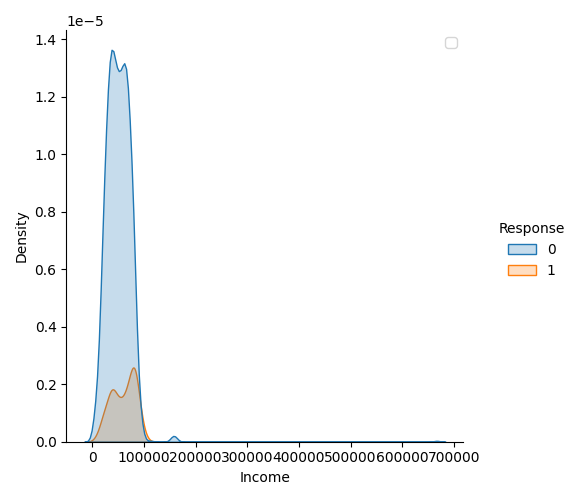
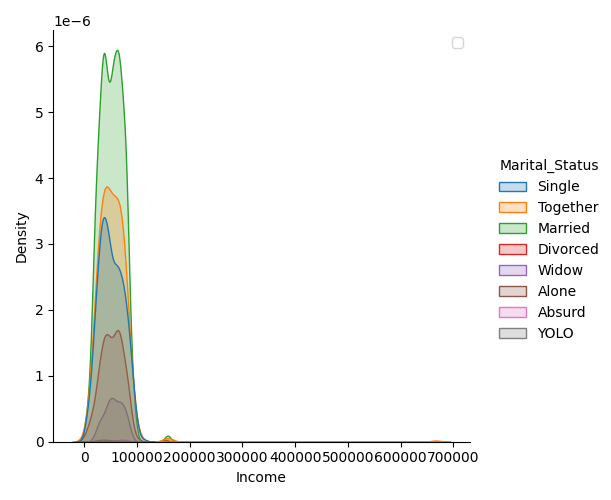


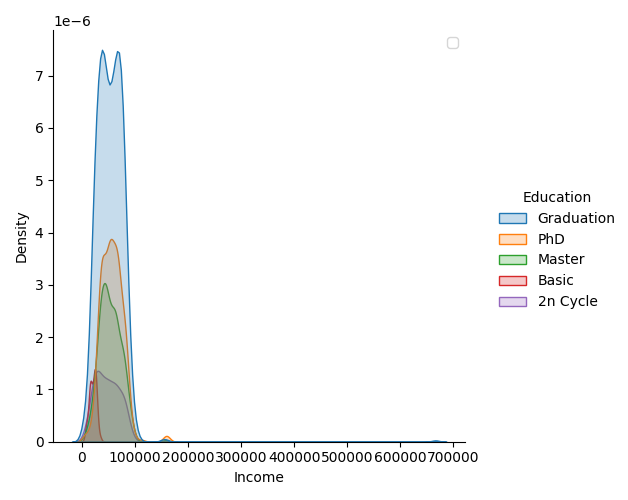
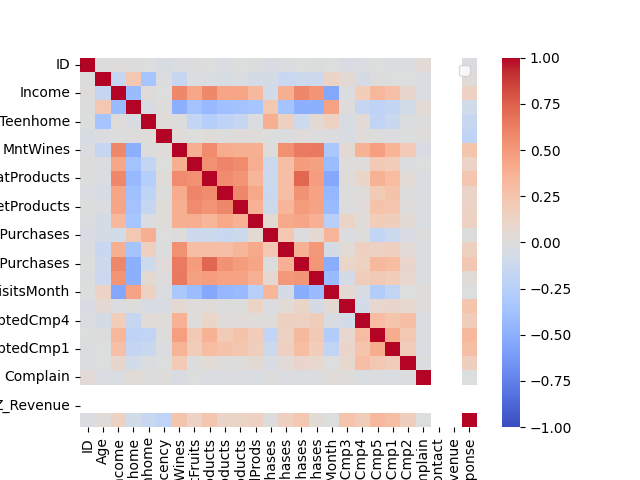


 ****

****

**** ****

**** ****

**** ****

**Вывод:** я ознакомился с основами библиотеки pandas и научился строить графики с использованием библиотек matplotlib.pyplot и seaborn.