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

Лабораторная работа №11

По дисциплине: «Языки программирования»

Тема: «Python. Основы Pandas»

Выполнила:
Студентка 2 курса
Группы ПО-7 (2)
Фурсевич Д.С.
Проверил:
Бойко Д.О.

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

```
[6]: df = pd.read_csv(r'C:\Users\Марина\Desktop\Univers\3 semestr\YaP\lab_8 pandas\marketing_campaign.csv')
```

```
[7]: df
```

[7]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	NumWebVisitsMonth	AcceptedCmp3	AcceptedCmp4	AcceptedCmp5	AcceptedCmp1	AcceptedCmp
0	5524	1957	Graduation	Single	58138.0	0	0	2012-09-04	58	635	...	7	0	0	0	0	
1	2174	1954	Graduation	Single	46344.0	1	1	2014-03-08	38	11	...	5	0	0	0	0	
2	4141	1965	Graduation	Together	71613.0	0	0	2013-08-21	26	426	...	4	0	0	0	0	
3	6182	1984	Graduation	Together	26646.0	1	0	2014-02-10	26	11	...	6	0	0	0	0	
4	5324	1981	PhD	Married	58293.0	1	0	2014-01-19	94	173	...	5	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2235	10870	1967	Graduation	Married	61223.0	0	1	2013-06-13	46	709	...	5	0	0	0	0	
2236	4001	1946	PhD	Together	64014.0	2	1	2014-06-10	56	406	...	7	0	0	0	1	
2237	7270	1981	Graduation	Divorced	56981.0	0	0	2014-01-25	91	908	...	6	0	1	0	0	
2238	8235	1956	Master	Together	69245.0	0	1	2014-01-24	8	428	...	3	0	0	0	0	
2239	9405	1954	PhD	Married	52869.0	1	1	2012-10-15	40	84	...	7	0	0	0	0	

2240 rows × 29 columns

2. Вывести общую информацию о датасете

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	NumWebVisitsMonth	AcceptedCmp3	AcceptedCmp4	AcceptedCmp5	AcceptedCmp1	AcceptedCmp
0	5524	1957	Graduation	Single	58138.0	0	0	2012-09-04	58	635	...	7	0	0	0	0	
1	2174	1954	Graduation	Single	46344.0	1	1	2014-03-08	38	11	...	5	0	0	0	0	
2	4141	1965	Graduation	Together	71613.0	0	0	2013-08-21	26	426	...	4	0	0	0	0	
3	6182	1984	Graduation	Together	26646.0	1	0	2014-02-10	26	11	...	6	0	0	0	0	
4	5324	1981	PhD	Married	58293.0	1	0	2014-01-19	94	173	...	5	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2235	10870	1967	Graduation	Married	61223.0	0	1	2013-06-13	46	709	...	5	0	0	0	0	
2236	4001	1946	PhD	Together	64014.0	2	1	2014-06-10	56	406	...	7	0	0	0	1	
2237	7270	1981	Graduation	Divorced	56981.0	0	0	2014-01-25	91	908	...	6	0	1	0	0	
2238	8235	1956	Master	Together	69245.0	0	1	2014-01-24	8	428	...	3	0	0	0	0	
2239	9405	1954	PhD	Married	52869.0	1	1	2012-10-15	40	84	...	7	0	0	0	0	

2240 rows × 29 columns

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

```
[8]: df.isnull().values.any()
```

```
[8]: True
```

```
[9]: df.isnull().sum()
```

```
[9]: ID                    0
     Year_Birth            0
     Education             0
     Marital_Status        0
     Income               24
     Kidhome               0
     Teenhome              0
     Dt_Customer           0
     Recency               0
     MntWines              0
     MntFruits             0
     MntMeatProducts       0
     MntFishProducts       0
     MntSweetProducts      0
     MntGoldProds          0
     NumDealsPurchases     0
     NumWebPurchases       0
     NumCatalogPurchases   0
     NumStorePurchases     0
     NumWebVisitsMonth     0
     AcceptedCmp3          0
     AcceptedCmp4          0
     AcceptedCmp5          0
     AcceptedCmp1          0
     AcceptedCmp2          0
     Complain              0
     Z_CostContact         0
     Z_Revenue             0
     Response              0
     dtype: int64
```

```
df[df['Income'].isnull()]
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	NumWebVisitsMonth	AcceptedCmp3	AcceptedCmp4	AcceptedCmp5	AcceptedCmp1	AcceptedCmp
10	1994	1983	Graduation	Married	NaN	1	0	2013-11-15	11	5	...	7	0	0	0	0	
27	5255	1986	Graduation	Single	NaN	1	0	2013-02-20	19	5	...	1	0	0	0	0	
43	7281	1959	PhD	Single	NaN	0	0	2013-11-05	80	81	...	2	0	0	0	0	
48	7244	1951	Graduation	Single	NaN	2	1	2014-01-01	96	48	...	6	0	0	0	0	
58	8557	1982	Graduation	Single	NaN	1	0	2013-06-17	57	11	...	6	0	0	0	0	
71	10629	1973	2n Cycle	Married	NaN	1	0	2012-09-14	25	25	...	8	0	0	0	0	
90	8996	1957	PhD	Married	NaN	2	1	2012-11-19	4	230	...	9	0	0	0	0	
91	9235	1957	Graduation	Single	NaN	1	1	2014-05-27	45	7	...	7	0	0	0	0	
92	5798	1973	Master	Together	NaN	0	0	2013-11-23	87	445	...	1	0	0	0	0	
128	8268	1961	PhD	Married	NaN	0	1	2013-07-11	23	352	...	6	0	0	0	0	
133	1295	1963	Graduation	Married	NaN	0	1	2013-08-11	96	231	...	4	0	0	0	0	
312	2437	1989	Graduation	Married	NaN	0	0	2013-06-03	69	861	...	3	0	1	0	1	
319	2863	1970	Graduation	Single	NaN	1	2	2013-08-23	67	738	...	7	0	1	0	1	
1379	10475	1970	Master	Together	NaN	0	1	2013-04-01	39	187	...	5	0	0	0	0	
1382	2902	1958	Graduation	Together	NaN	1	1	2012-09-03	87	19	...	5	0	0	0	0	
1383	4345	1964	2n Cycle	Single	NaN	1	1	2014-01-12	49	5	...	7	0	0	0	0	
1386	3769	1972	PhD	Together	NaN	1	0	2014-03-02	17	25	...	7	0	0	0	0	
2059	7187	1969	Master	Together	NaN	1	1	2013-05-18	52	375	...	3	0	0	0	0	
2061	1612	1981	PhD	Single	NaN	1	0	2013-05-31	82	23	...	6	0	0	0	0	
2078	5079	1971	Graduation	Married	NaN	1	1	2013-03-03	82	71	...	8	0	0	0	0	
2079	10339	1954	Master	Together	NaN	0	1	2013-06-23	83	161	...	6	0	0	0	0	
2081	3117	1955	Graduation	Single	NaN	0	1	2013-10-18	95	264	...	7	0	0	0	0	
2084	5250	1943	Master	Widow	NaN	0	0	2013-10-30	75	532	...	1	0	0	1	0	
2228	8720	1978	2n Cycle	Together	NaN	0	0	2012-08-12	53	32	...	0	0	1	0	0	

24 rows × 29 columns

4. Удалить колонки "Z_CostContact", "Z_Revenue"

```
df.drop(['Z_CostContact', 'Z_Revenue'], axis=1, inplace=True)
```

```
df
```

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth	AcceptedCmp3	AcceptedCmp4	A
0	5524	1957	Graduation	Single	58138.0	0	0	2012-09-04	58	635	...	10	4	7	0	0	
1	2174	1954	Graduation	Single	46344.0	1	1	2014-03-08	38	11	...	1	2	5	0	0	
2	4141	1965	Graduation	Together	71613.0	0	0	2013-08-21	26	426	...	2	10	4	0	0	
3	6182	1984	Graduation	Together	26646.0	1	0	2014-02-10	26	11	...	0	4	6	0	0	
4	5324	1981	PhD	Married	58293.0	1	0	2014-01-19	94	173	...	3	6	5	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2235	10870	1967	Graduation	Married	61223.0	0	1	2013-06-13	46	709	...	3	4	5	0	0	
2236	4001	1946	PhD	Together	64014.0	2	1	2014-06-10	56	406	...	2	5	7	0	0	
2237	7270	1981	Graduation	Divorced	56981.0	0	0	2014-01-25	91	908	...	3	13	6	0	1	
2238	8235	1956	Master	Together	69245.0	0	1	2014-01-24	8	428	...	5	10	3	0	0	
2239	9405	1954	PhD	Married	52869.0	1	1	2012-10-15	40	84	...	1	4	7	0	0	

2240 rows × 27 columns

5. Переименовать колонку "Year_Birth" в "Age"

```
df = df.rename(columns={'Year_Birth': 'Age'})
```

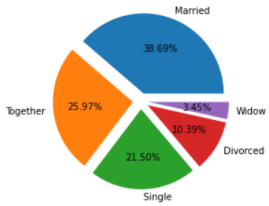
```
df
```

	ID	Age	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWines	...	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth	AcceptedCmp3	AcceptedCmp4	Accep
0	5524	1957	Graduation	Single	58138.0	0	0	2012-09-04	58	635	...	10	4	7	0	0	
1	2174	1954	Graduation	Single	46344.0	1	1	2014-03-08	38	11	...	1	2	5	0	0	
2	4141	1965	Graduation	Together	71613.0	0	0	2013-08-21	26	426	...	2	10	4	0	0	
3	6182	1984	Graduation	Together	26646.0	1	0	2014-02-10	26	11	...	0	4	6	0	0	
4	5324	1981	PhD	Married	58293.0	1	0	2014-01-19	94	173	...	3	6	5	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2235	10870	1967	Graduation	Married	61223.0	0	1	2013-06-13	46	709	...	3	4	5	0	0	
2236	4001	1946	PhD	Together	64014.0	2	1	2014-06-10	56	406	...	2	5	7	0	0	
2237	7270	1981	Graduation	Divorced	56981.0	0	0	2014-01-25	91	908	...	3	13	6	0	1	
2238	8235	1956	Master	Together	69245.0	0	1	2014-01-24	8	428	...	5	10	3	0	0	
2239	9405	1954	PhD	Married	52869.0	1	1	2012-10-15	40	84	...	1	4	7	0	0	

2240 rows × 27 columns

6. Оценить состояние колонок "Marital_Status", "Education". Построить информативные диаграммы и гистограммы для каждой.

```
labels = 'Married ','Together','Single ','Divorced', 'Widow'
sizes = [864,580,480,232,77]
explode = (0.1,0.1,0.1,0.1,0.1)
h = plt.pie(sizes,labels = labels,explode = explode,autopct = '%1.2f%%')
```

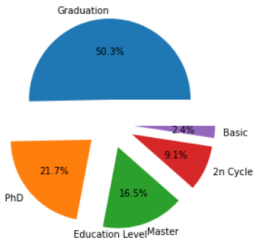



```
df.Marital_Status.value_counts()
```

```
Married     864
Together    580
Single      480
Divorced    232
Widow        77
Alone         3
Absurd        2
YOLO          2
Name: Marital_Status, dtype: int64
```

```
explode = (0.3,0.3,0.3,0.3,0.3)
h = df.Education.value_counts().plot.pie(autopct='%1.1f%%', explode = explode, ylabel='')
h.set_xlabel('Education Level')
```

```
Text(0.5, 0, 'Education Level')
```

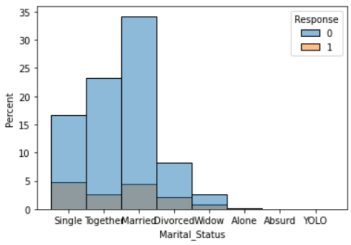



```
df.Education.value_counts()
```

```
Graduation    1127
PhD            486
Master         370
2n Cycle       203
Basic           54
Name: Education, dtype: int64
```

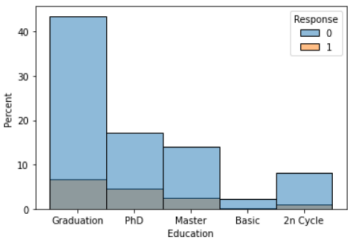
```
sns.histplot(data=df, x="Marital_Status", stat='percent', hue="Response")
```

```
<AxesSubplot:xlabel='Marital_Status', ylabel='Percent'>
```



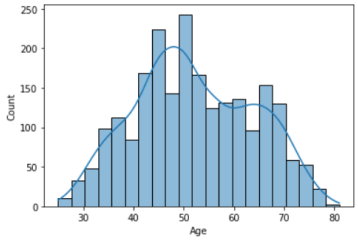

```
sns.histplot(data=df, x="Education", stat='percent', hue="Response" )
```

```
<AxesSubplot:xlabel='Education', ylabel='Percent'>
```



7. Создать гистаграмму по колонке "Age" и оценить на распределение по Гауссу.

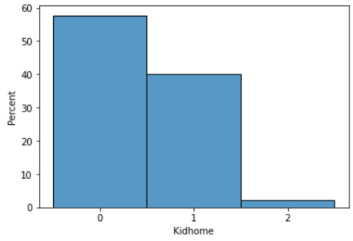
```
def calculate_age(born) -> int:
    return int(pd.to_datetime("today").strftime("%Y")) - int(born)

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



8. Оценка полей "Kidhome" и "Teenhome", "Response" и "Income" (диаграммы и гистограммы)

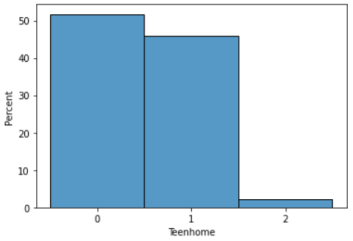
```
index = [0, 1, 2]
sns.histplot(data=df.Kidhome, discrete=True, stat='percent' )
plt.xticks(index)
```

```
([<matplotlib.axis.XTick at 0x1652b052310>,
  <matplotlib.axis.XTick at 0x1652b052160>,
  <matplotlib.axis.XTick at 0x1652b006430>],
 [Text(0, 0, ''), Text(0, 0, ''), Text(0, 0, '')])
```



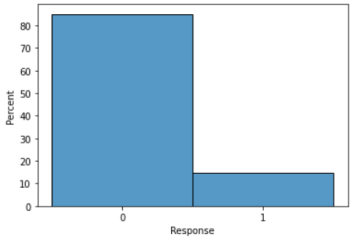

```
index = [0, 1, 2]
sns.histplot(data=df.Teenhome , discrete=True, stat='percent')
plt.xticks(index)
```

```
([<matplotlib.axis.XTick at 0x1652b021130>,
  <matplotlib.axis.XTick at 0x1652b021100>,
  <matplotlib.axis.XTick at 0x1652b338e20>],
 [Text(0, 0, ''), Text(0, 0, ''), Text(0, 0, '')])
```



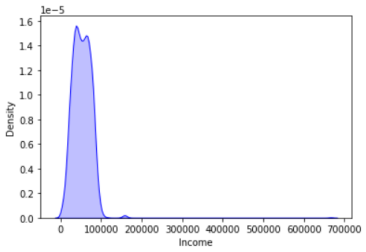

```
index = [0, 1]
sns.histplot(data=df.Response , discrete=True, stat='percent')
plt.xticks(index)
```

```
([<matplotlib.axis.XTick at 0x1652b364d60>,
  <matplotlib.axis.XTick at 0x1652b364d30>],
 [Text(0, 0, ''), Text(0, 0, '')])
```



```
sns.kdeplot(df['Income'], color='b', shade=True)
```

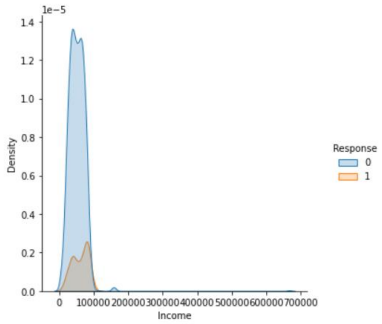
```
<AxesSubplot:xlabel='Income', ylabel='Density'>
```



9. Построить графики "Response", "Marital_Status", "Education" и "Kidhome" по образцу

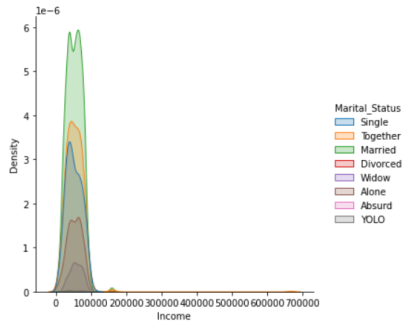
```
sns.displot(x="Income",
            hue="Response",
            kind="kde",
            fill=True,
            data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x1652d433610>
```



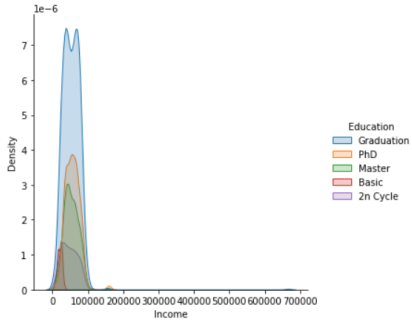

```
sns.displot(x="Income",
            hue="Marital_Status",
            kind="kde",
            fill=True,
            warn_singular=False,
            data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x1652ed71f70>
```



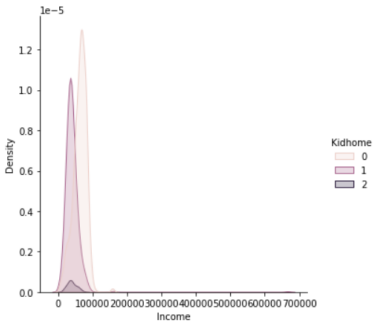
```
sns.displot(x="Income",
            hue="Education",
            kind="kde",
            fill=True,
            data=df)
```

```
<seaborn.axisgrid.FacetGrid at 0x1652d433a00>
```



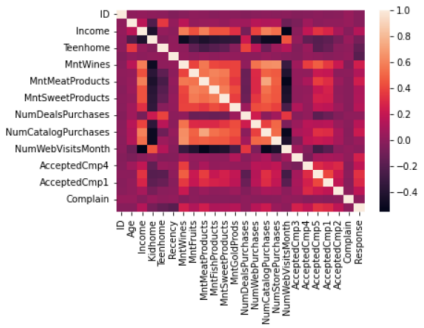

```
sns.displot(x="Income",
            hue="Kidhome",
            kind="kde",
            fill=True,
            data=df)
```

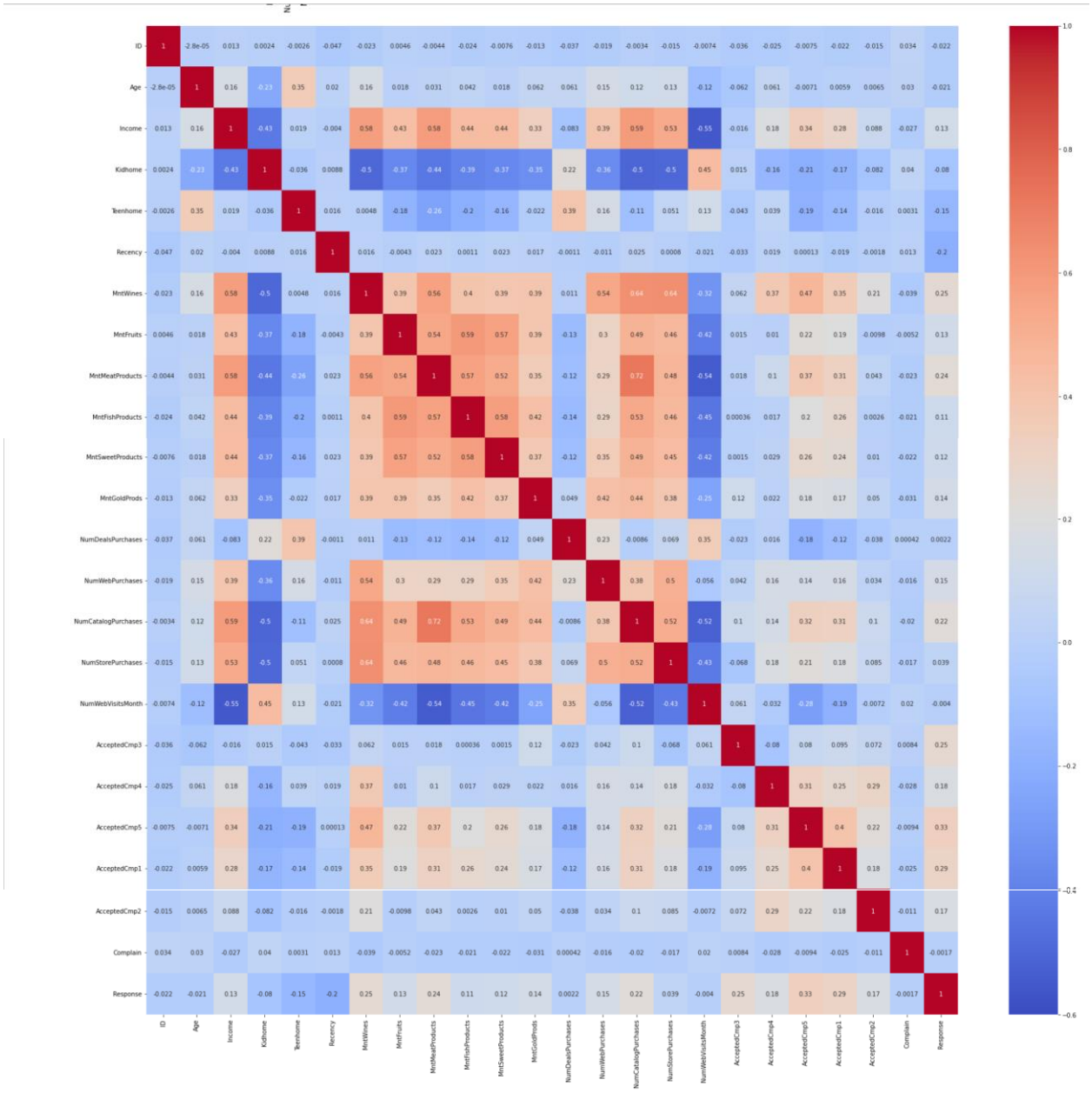
```
<seaborn.axisgrid.FacetGrid at 0x1652ed2e9a0>
```



10. Построить heatmap для всех числовых колонок

```
sns.heatmap(df.corr())
heatmap = plt.figure(figsize=(30, 30))
heatmap = sns.heatmap(df.corr(), vmin=-0.6, vmax=1, annot=True,  cmap='coolwarm')

# vmin, vmax — устанавливают диапазон значений, которые служат основой для цветовой карты (colormap)
# annot — при значении True числовые значения корреляции отображаются внутри ячеек
```





Вывод: в ходе выполнения данной лабораторной работы ознакомились с основами библиотеки pandas и научились строить графики с использованием библиотек matplotlib.pyplot и seaborn.