МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа №1
по курсу «Методы машинного обучения»

Тема: «Разведочный анализ данных. Исследование и визуализация данных»

ИСПОЛНИТЕЛЬ: Ромичева Е.В.

группа ИУ5-22М _____
подпись

"__"_____2019 г.

ПРЕПОДАВАТЕЛЬ: Гапанюк Ю.Е.

_____
подпись

"__"_____2019 г.

Москва - 2019

Цель работы

Цель лабораторной работы: изучение различных методов визуализация данных.

Задание

1. Выбрать набор данных (датасет).
2. Создать ноутбук, который содержит следующие разделы:
   - Текстовое описание выбранного Вами набора данных.
   - Основные характеристики датасета.
   - Визуальное исследование датасета.
   - Информация о корреляции признаков.
3. Сформировать отчет и разместить его в своем репозитории на github.

Датасет BlackFriday – 550000 наблюдений за покупками, совершенными в ритейле в черную пятницу. Анализ покупок помогает выявить закономерности выбора товаров покупателями и спрогнозировать дальнейший спрос на товар.

Текстовое описание набора данных

Поле	Значение
User_ID	Покупатель
Product_ID	Товар
Gender	Пол
Age	Возраст
Occupation	Профессия
City_Category	Тип города
Stay_In_Current_City_Years	Срок проживания в городе
Marital_Status	Семейное положение
Product_Category_1	Категория товаров 1
Product_Category_2	Категория товаров 2
Product_Category_3	Категория товаров 3
Purchase	Сумма покупки

Первые 5 строк датасета

```
In [11]: data.head()
```

Out[11]:

	User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Produc
0	1000001	P00069042	F	0-17	10	A	2	0	3	NaN	
1	1000001	P00248942	F	0-17	10	A	2	0	1	6.0	
2	1000001	P00087842	F	0-17	10	A	2	0	12	NaN	
3	1000001	P00085442	F	0-17	10	A	2	0	12	14.0	
4	1000002	P00285442	M	55+	16	C	4+	0	8	NaN	

Размер датасета

```
In [16]: data.shape
Out[16]: (537577, 12)
```

Список колонок с типами данных

```
data.dtypes
```

```
User_ID                       int64
Product_ID                   object
Gender                       object
Age                          object
Occupation                    int64
City_Category                object
Stay_In_Current_City_Years   object
Marital_Status                int64
Product_Category_1            int64
Product_Category_2          float64
Product_Category_3          float64
Purchase                      int64
dtype: object
```

Проверим число уникальных значений. Цикл по колонкам датасета

```
for col in data.columns:
    print('{} unique element: {}'.format(col,data[col].nunique()))
```

```
User_ID unique element: 5891
Product_ID unique element: 3623
Gender unique element: 2
Age unique element: 7
Occupation unique element: 21
City_Category unique element: 3
Stay_In_Current_City_Years unique element: 5
Marital_Status unique element: 2
Product_Category_1 unique element: 18
Product_Category_2 unique element: 17
Product_Category_3 unique element: 15
Purchase unique element: 17959
```

Проверим наличие пустых значений. Цикл по колонкам датасета

```
for col in data.columns:
    # Количество пустых значений - все значения заполнены
    temp_null_count = data[data[col].isnull()].shape[0]
    print('{} - {}'.format(col, temp_null_count))

missing_ser_percentage = (data.isnull().sum()/data.shape[0]*100).sort_values(ascending=False)
missing_ser_percentage = missing_ser_percentage[missing_ser_percentage!=0].round(2)
missing_ser_percentage.name = 'missing values %'
print('\n<NaN ratio>')
print(missing_ser_percentage)
```

```
User_ID - 0
Product_ID - 0
Gender - 0
Age - 0
Occupation - 0
City_Category - 0
Stay_In_Current_City_Years - 0
Marital_Status - 0
Product_Category_1 - 0
Product_Category_2 - 166986
Product_Category_3 - 373299
Purchase - 0

<NaN ratio>
Product_Category_3    69.44
Product_Category_2    31.06
Name: missing values %, dtype: float64
```

Основные статистические характеристики набора данных

```
data2 = data
data2.drop(columns = ["User_ID","Occupa
data2.describe()
```

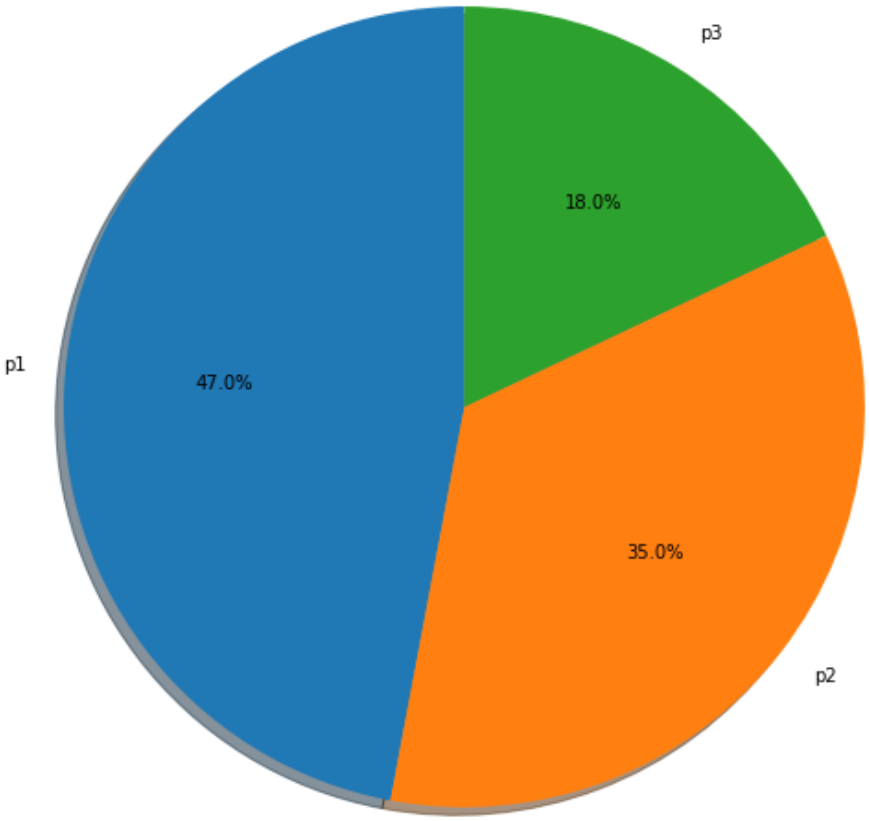
	Purchase
count	537577.000000
mean	9333.859853
std	4981.022133
min	185.000000
25%	5866.000000
50%	8062.000000
75%	12073.000000
max	23961.000000

В данном датасете статистический интерес представляет графа Потраченная сумма. Видно, что в среднем на покупки тратили 9000 долларов.

Какую категорию товаров больше всего покупали

```
p1 = data.groupby('Product_Category_1')['Purchase'].sum().sum()
p2 = data.groupby('Product_Category_2')['Purchase'].sum().sum()
p3 = data.groupby('Product_Category_3')['Purchase'].sum().sum()
products = [p1, p2, p3]
labels = ['p1', 'p2', 'p3']
fig, ax = plt.subplots(figsize=(10,10))
ax.pie(products, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
print(p1, p2, p3)
```

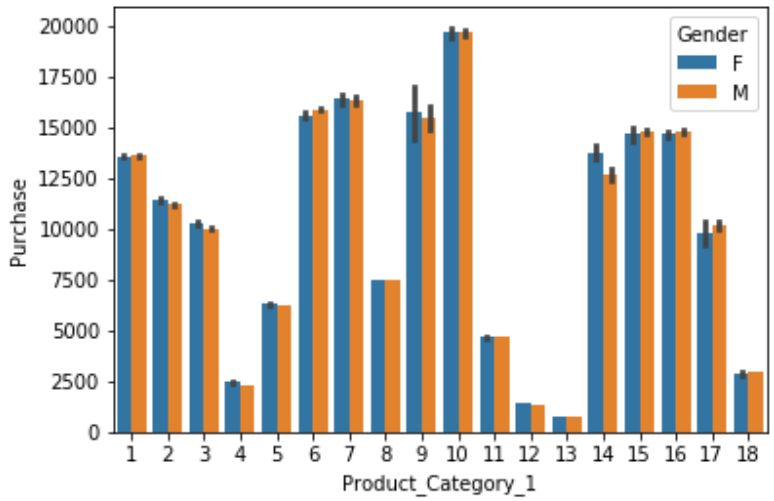
```
5017668378 3732568444 1915646035
```



Из диаграммы видно, что чаще всего покупали товары категории 1. Далее проведем более тщательное исследование именно этих товаров.

Распределение затрат на категорию товаров 1 по полу

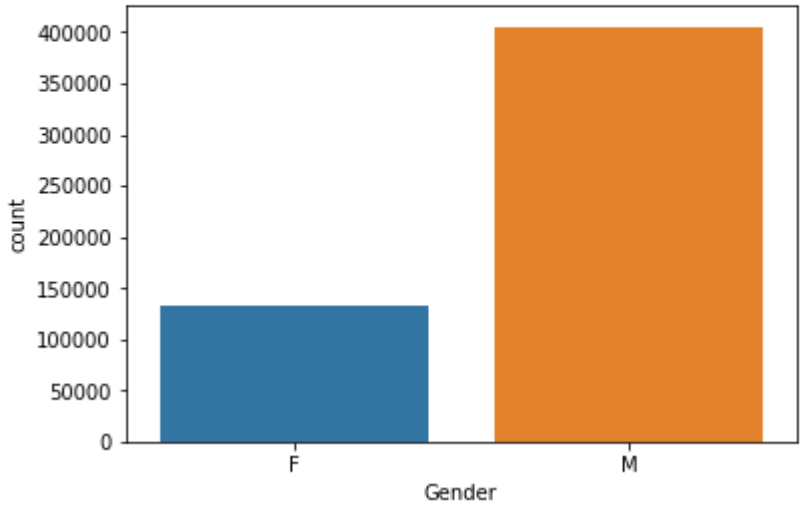
```
ax = sns.barplot(x="Product_Category_1", y="Purchase", hue="Gender", data=data)
```



Распределение покупок по полу

```
sns.countplot(data['Gender'])
```

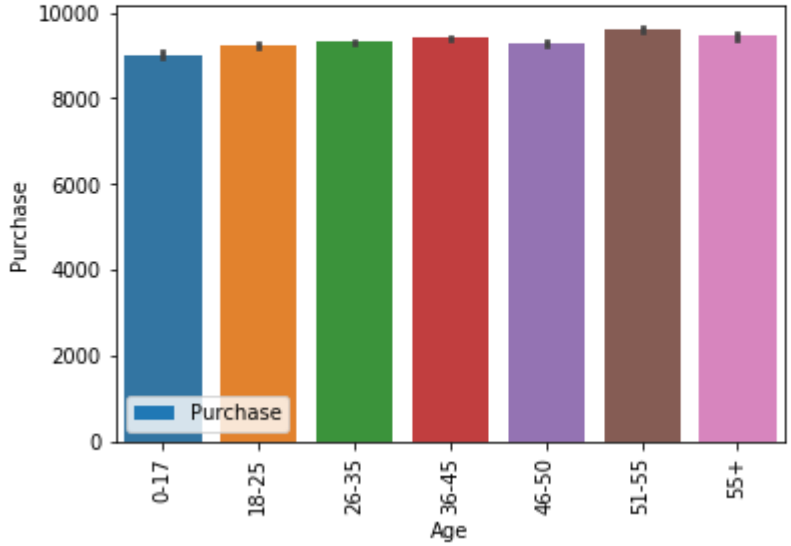
```
<matplotlib.axes._subplots.AxesSubplot at 0x27e36bb19e8>
```



Распределение затрат по возрасту

```
age_order = ['0-17','18-25','26-35','36-45','46-50','51-55','55+']
data[['Age','Purchase']].groupby('Age').mean().plot.bar()
sns.barplot('Age', 'Purchase', order=age_order, data = data)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x27e36e84278>
```

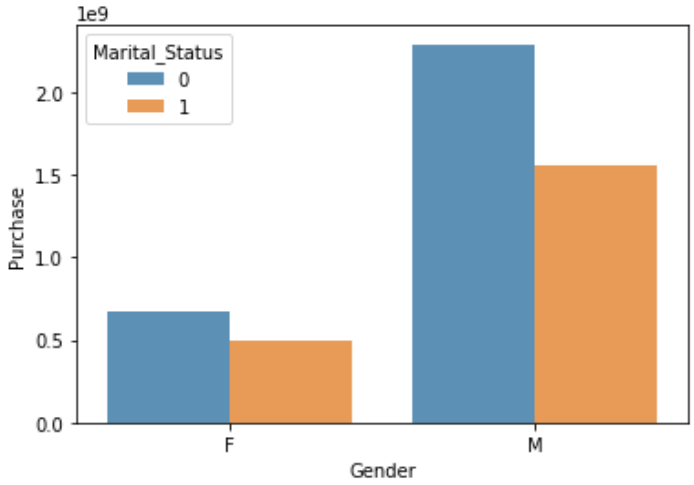


Больше всего покупали мужчины, а больше всего тратили люди старшего возраста. Можно сделать вывод, что с возрастом люди более финансово устойчивы и могут позволить себе дорогие покупки.

Распределение затрат на покупки по полу, семейному положению и возрасту

```
qq = data.groupby(['Gender','Marital_Status']).agg({'Purchase': sum}).reset_index()
sns.barplot('Gender','Purchase',hue='Marital_Status',data=qq, alpha = 0.8)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x205c665eac8>
```

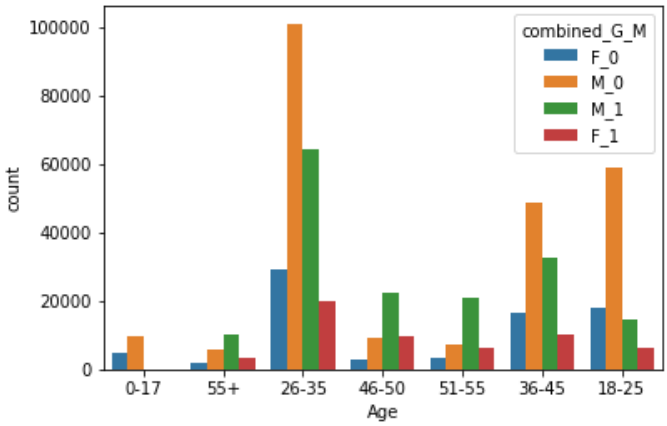


Из диаграммы видно, что незамужние люди тратят больше денег на покупки.

Распределение числа покупок по полу, семейному положению и возрасту

```
data['combined_G_M'] = data.apply(lambda x:'%s_%s' % (x['Gender'],x['Marital_Status']),axis=1)
sns.countplot(data['Age'],hue=data['combined_G_M'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x27e4bd36cf8>
```

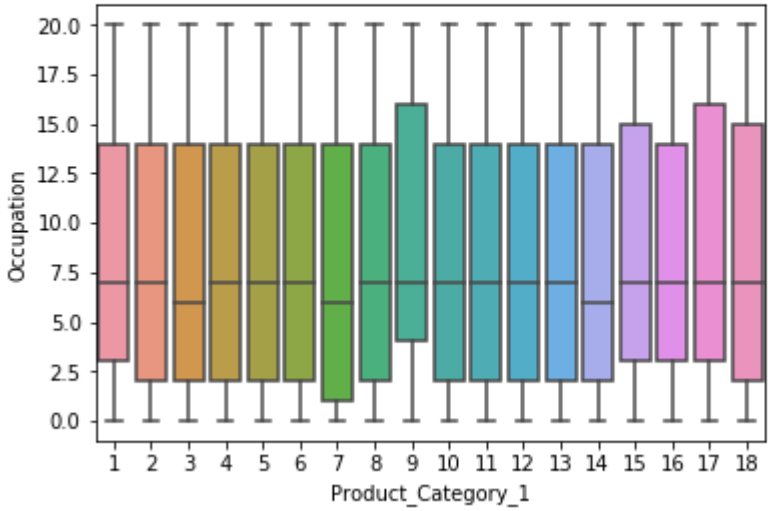


Из диаграммы видно, что большее число покупок совершили люди 26-35 лет.

Какая профессия у людей, купивших продукт категории 1

```
age_order = ['0-17','18-25','26-35','36-45','46-50','51-55','55+']
sns.boxplot(x=data['Product_Category_1'], y = data['Occupation'])
```

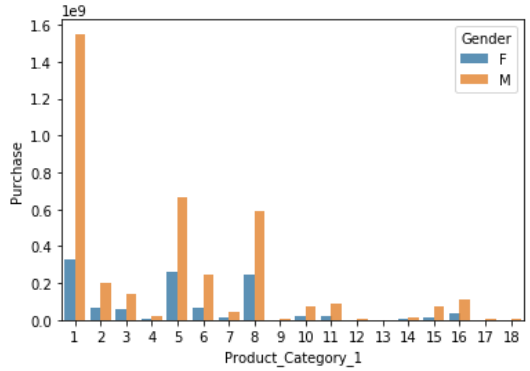
```
<matplotlib.axes._subplots.AxesSubplot at 0x17c4e0289b0>
```



Кто покупал больше категорию товаров 1 (м/ж)

```
df_Tpurchase_by_PC1_Gender = data.groupby(['Product_Category_1','Gender']).agg({'Purchase':np.sum}).reset_index()
sns.barplot('Product_Category_1','Purchase',hue='Gender',data=df_Tpurchase_by_PC1_Gender,alpha = 0.8)
```

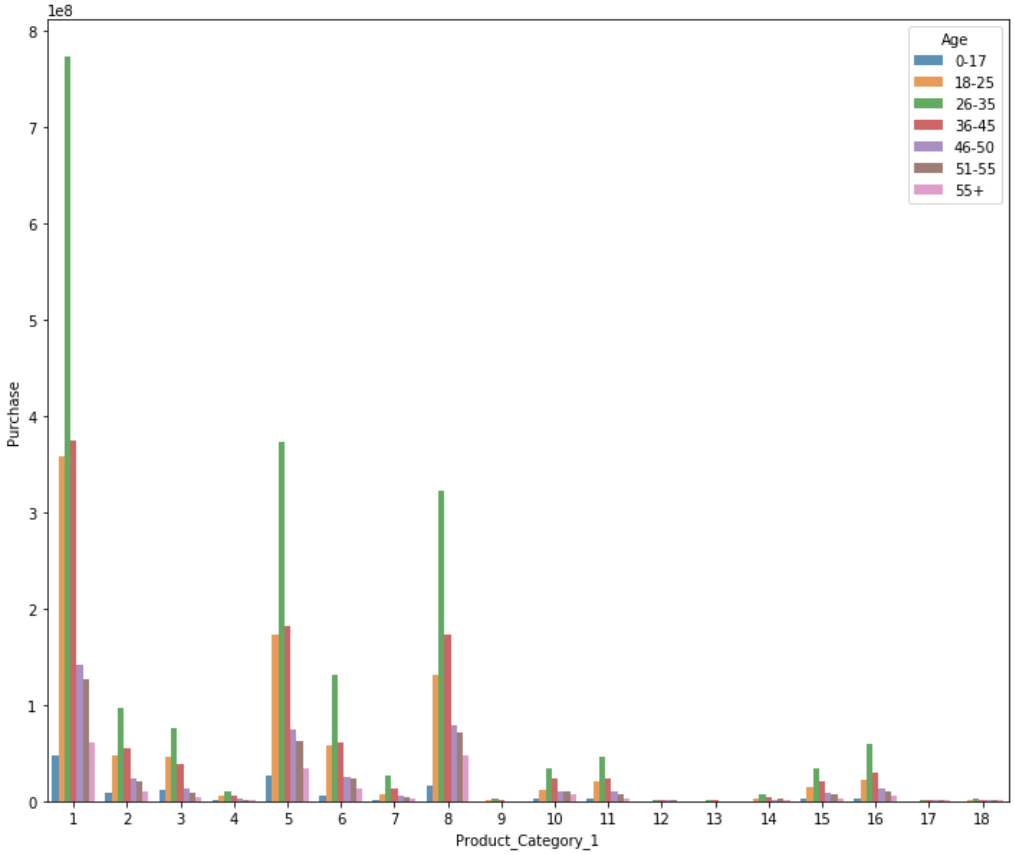
```
<matplotlib.axes._subplots.AxesSubplot at 0x27e373cabe0>
```



Кто покупал больше категорию товаров 1 (по возрасту)

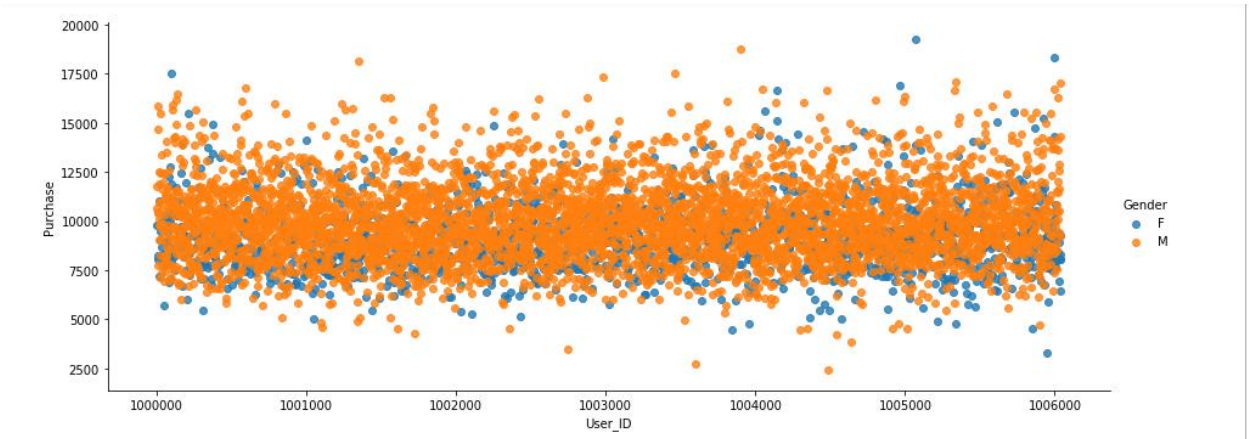
```
df_Tpurchase_by_PC1_Age = data.groupby(['Product_Category_1','Age']).agg({'Purchase':np.sum}).reset_index()
fig = plt.figure(figsize=(12,10))
sns.barplot('Product_Category_1','Purchase',hue='Age',data=df_Tpurchase_by_PC1_Age,alpha = 0.8)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x27e381412e8>
```

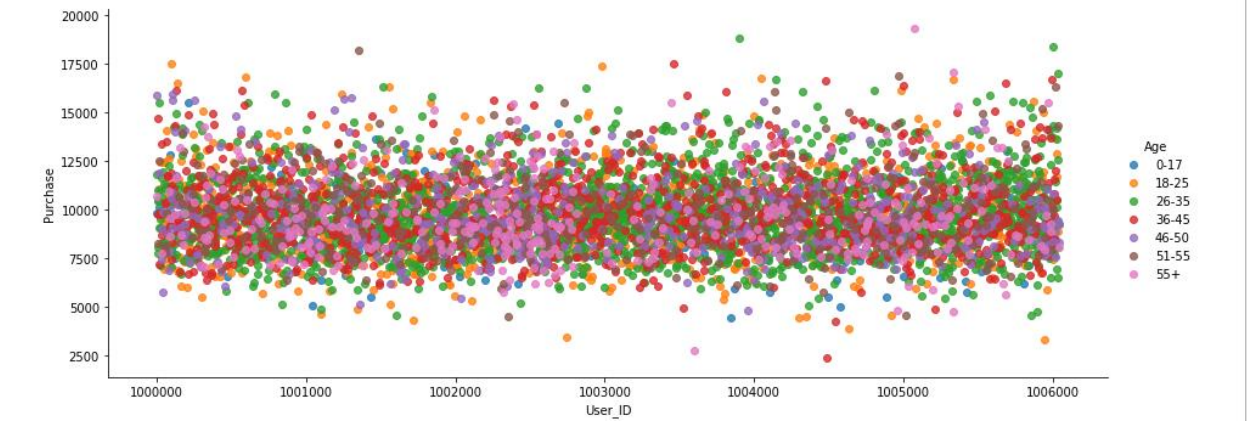


Средняя сумма покупок по полу и возрасту

```
Avgpurchase_by_UID_Gender = data.groupby(['User_ID','Gender']).agg({'Purchase':np.mean}).reset_index()
Avgpurchase_by_UID_Age = data.groupby(['User_ID','Age']).agg({'Purchase':np.mean}).reset_index()
age_order = ['0-17','18-25','26-35','36-45','46-50','51-55','55+']
plt.figure(figsize=(20,5))
sns.lmplot('User_ID','Purchase',data=Avgpurchase_by_UID_Gender,fit_reg=False,hue='Gender',aspect=2.5)
plt.figure(figsize=(20,5))
sns.lmplot('User_ID','Purchase',data=Avgpurchase_by_UID_Age,fit_reg=False,hue='Age',hue_order=age_order,aspect=2.5)
```



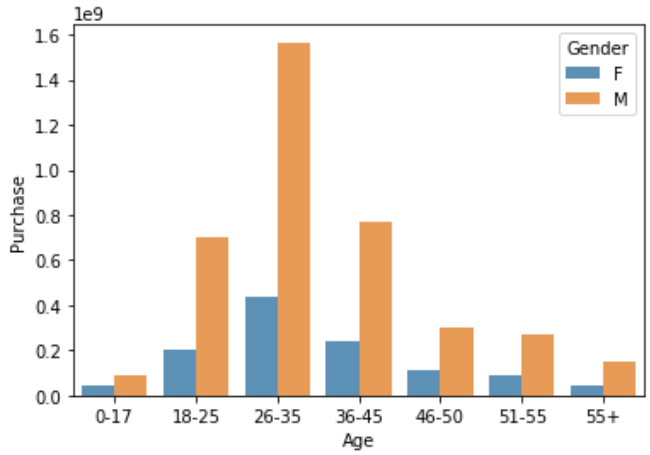

```
<Figure size 1440x360 with 0 Axes>
```



Затраты на покупки по полу и возрасту

```
age_order = ['0-17','18-25','26-35','36-45','46-50','51-55','55+']
df_Tpurchase_by_Age = data.groupby(['Age','Gender']).agg({'Purchase':np.sum}).reset_index()
sns.barplot('Age','Purchase',hue='Gender',data=df_Tpurchase_by_Age, alpha = 0.8)
```

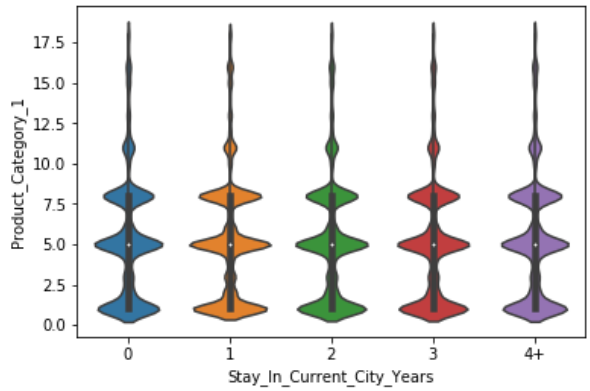
```
<matplotlib.axes._subplots.AxesSubplot at 0x205ca8da4a8>
```



Сколько лет жили в городе люди, купившие товары категории 1

```
stay_order = ['0','1','2','3','4+']
sns.violinplot(x=data['Stay_In_Current_City_Years'], y = data['Product_Category_1'], order = stay_order)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x17c4e8baf28>
```



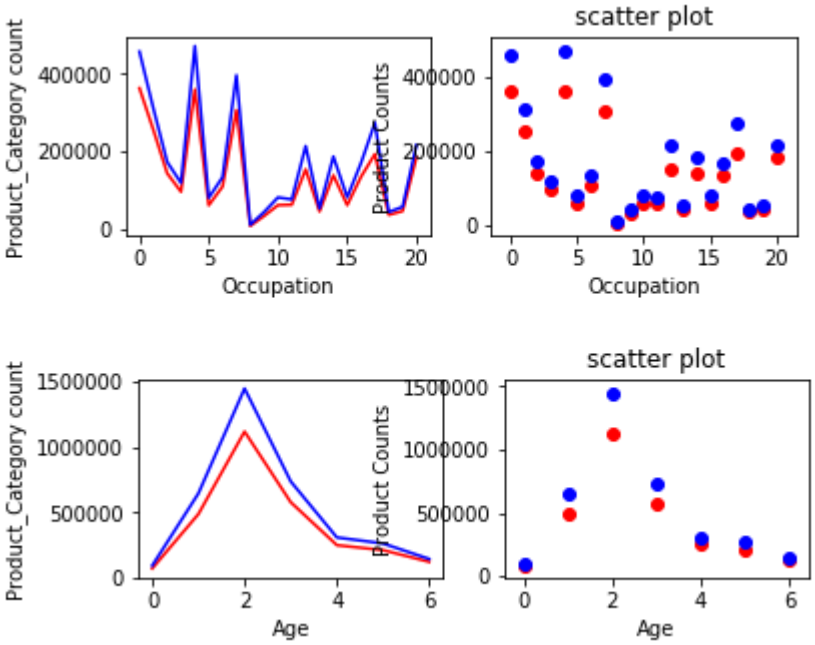
Сравнение покупок товаров категории 1 и 2

```
# draw line graph in subplot
plt.subplot(2,2,1)
plt.plot( dataByOcc.Occupation, dataByOcc.Product_Category_1, color="red", label="Product_Category_1" )
plt.plot( dataByOcc.Occupation, dataByOcc.Product_Category_2, color="blue", label="Product_Category_2" )
plt.xlabel("Occupation")
plt.ylabel("Product_Category count")

# draw scatter graph in subplot
plt.subplot(2,2,2)
plt.scatter(dataByOcc.Occupation ,dataByOcc.Product_Category_1, color="r", label="Product_Category_1")
plt.scatter(dataByOcc.Occupation ,dataByOcc.Product_Category_2, color="b", label="Product_Category_2")
plt.xlabel("Occupation")
plt.ylabel("Product Counts")
plt.title("scatter plot")
plt.show()

# draw line graph in subplot
plt.subplot(2,2,3)
plt.plot( dataByAge.AgeCategory, dataByAge.Product_Category_1, color="red", label="Product_Category_1" )
plt.plot( dataByAge.AgeCategory, dataByAge.Product_Category_2, color="blue", label="Product_Category_2" )
plt.xlabel("Age")
plt.ylabel("Product_Category count")

# draw scatter graph in subplot
plt.subplot(2,2,4)
plt.scatter(dataByAge.AgeCategory ,dataByAge.Product_Category_1, color="r", label="Product_Category_1")
plt.scatter(dataByAge.AgeCategory ,dataByAge.Product_Category_2, color="b", label="Product_Category_2")
plt.xlabel("Age")
plt.ylabel("Product Counts")
plt.title("scatter plot")
plt.show()
```



Корреляционные характеристики

```
# сделаем возраст числовым значением
def map_age(age):
    if age == '0-17':
        return 0
    elif age == '18-25':
        return 1
    elif age == '26-35':
        return 2
    elif age == '36-45':
        return 3
    elif age == '46-50':
        return 4
    elif age == '51-55':
        return 5
    else:
        return 6
data['Age'] = data['Age'].apply(map_age)

# сделаем город числовым значением
def map_city_categories(city_category):
    if city_category == 'A':
        return 2
    elif city_category == 'B':
        return 1
    else:
        return 0
data['City_Category'] = data['City_Category'].apply(map_city_categories)

# сделаем время проживания в городе числовым значением
def map_stay(stay):
    if stay == '4+':
        return 4
```

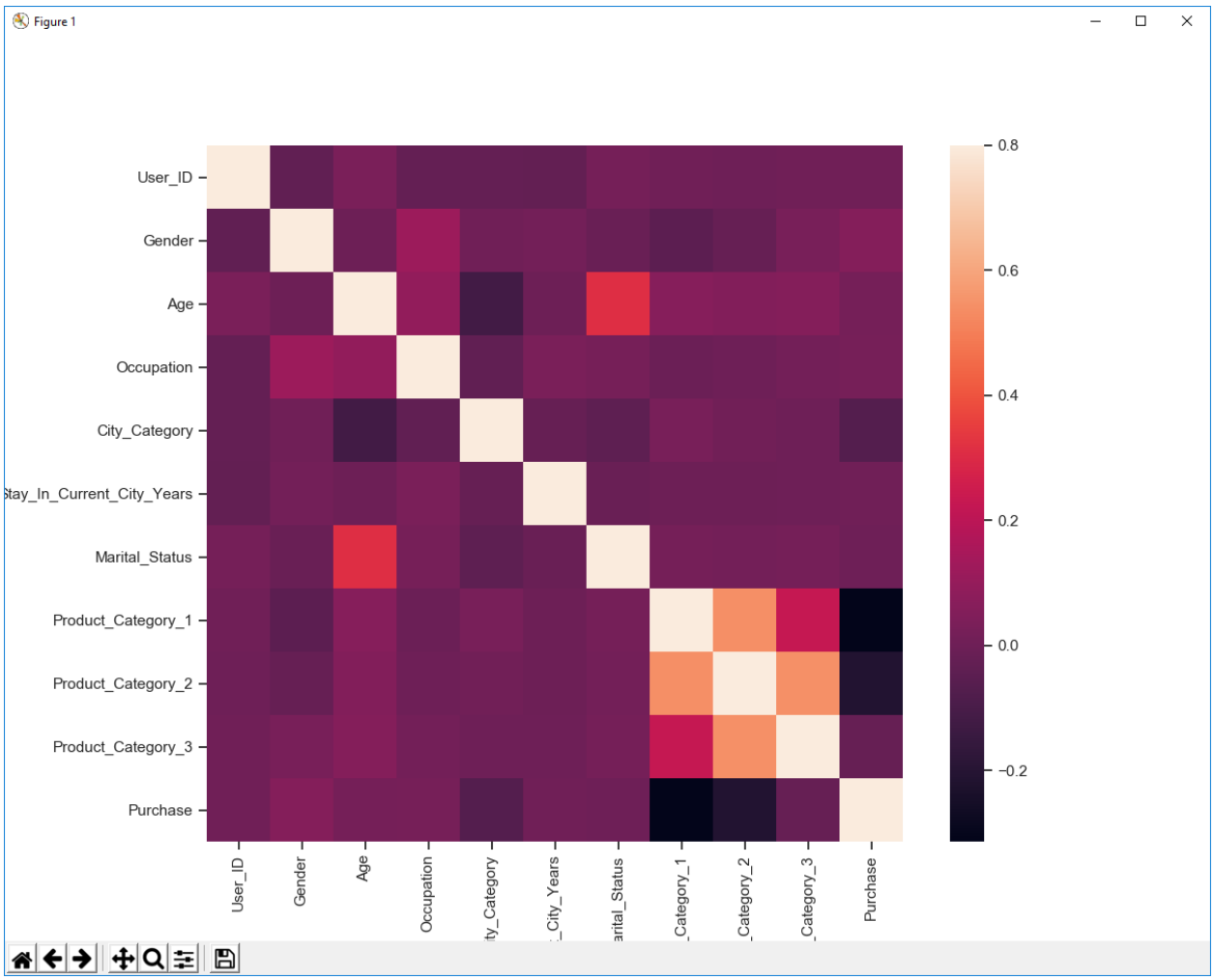
```
    else:
        return int(stay)
data['Stay_In_Current_City_Years'] = data['Stay_In_Current_City_Years'].apply(map_stay)

print (data.head())

corrmat = data.corr()
fig,ax = plt.subplots(figsize = (12,9))
sns.heatmap(corrmat, vmax=.8, square=True)
```

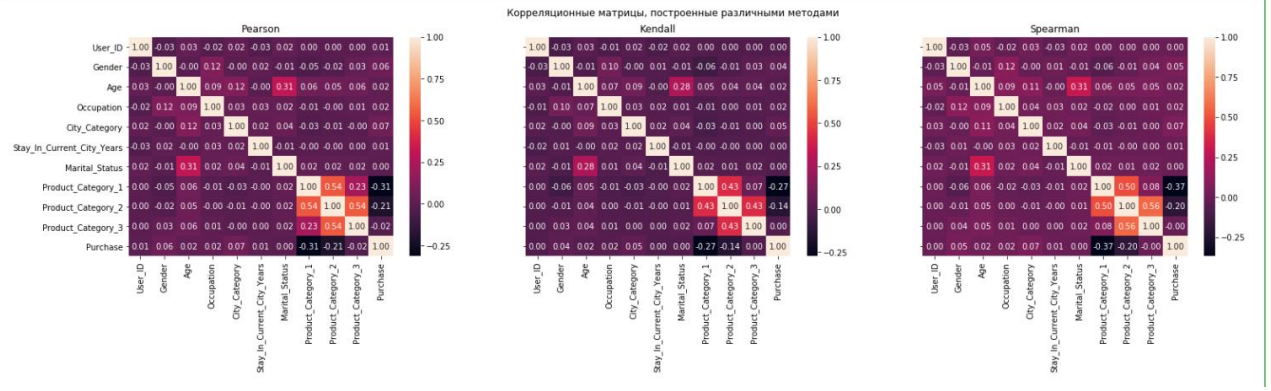
	User_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1
User_ID	1.000000	-0.031898	0.033359	-0.023024	0.024107	-0.030655	0.018732	0.003687
Gender	-0.031898	1.000000	-0.004413	0.117294	-0.004129	0.015392	-0.010379	-0.045661
Age	0.033359	-0.004413	1.000000	0.091898	0.122308	-0.004754	0.312079	0.061951
Occupation	-0.023024	0.117294	0.091898	1.000000	0.033781	0.031203	0.024691	-0.008114
City_Category	0.024107	-0.004129	0.122308	0.033781	1.000000	0.019948	0.040173	-0.027444
Stay_In_Current_City_Years	-0.030655	0.015392	-0.004754	0.031203	0.019948	1.000000	-0.012663	-0.004182
Marital_Status	0.018732	-0.010379	0.312079	0.024691	0.040173	-0.012663	1.000000	0.020546
Product_Category_1	0.003687	-0.045661	0.061951	-0.008114	-0.027444	-0.004182	0.020546	1.000000
Product_Category_2	0.001471	-0.018440	0.054762	-0.000031	-0.012120	-0.001781	0.015116	0.540423
Product_Category_3	0.004045	0.028403	0.057155	0.013452	-0.002672	0.002039	0.019452	0.229490
Purchase	0.005389	0.060086	0.017717	0.021104	0.068507	0.005470	0.000129	-0.314125

Product_Category_2	Product_Category_3	Purchase
0.001471	0.004045	0.005389
-0.018440	0.028403	0.060086
0.054762	0.057155	0.017717
-0.000031	0.013452	0.021104
-0.012120	-0.002672	0.068507
-0.001781	0.002039	0.005470
0.015116	0.019452	0.000129
0.540423	0.229490	-0.314125
1.000000	0.543544	-0.209973
0.543544	1.000000	-0.022257
-0.209973	-0.022257	1.000000



Из графика видно, что семейное положение зависит от возраста, однако невозможно выявить четкой корреляции между характеристиками покупателей и совершаемыми покупками.

```
fig, ax = plt.subplots(1, 3, sharex='col', sharey='row', figsize=(25,5))
sns.heatmap(data1.corr(method='pearson'), ax=ax[0], annot=True, fmt='.2f')
sns.heatmap(data1.corr(method='kendall'), ax=ax[1], annot=True, fmt='.2f')
sns.heatmap(data1.corr(method='spearman'), ax=ax[2], annot=True, fmt='.2f')
fig.suptitle('Корреляционные матрицы, построенные различными методами')
ax[0].title.set_text('Pearson')
ax[1].title.set_text('Kendall')
ax[2].title.set_text('Spearman')
```



Итак, по диаграммам можно определить основные гендерные и возрастные характеристики покупателей товаров в Черную Пятницу. Это поможет определить целевую аудиторию той или иной категории товара и в будущем спрогнозировать и спланировать продажи.