Министерство образования Республики Беларусь

УО «Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №8

По дисциплине: “Языки программирования”

Тема: «Python. Основы Pandas»

Выполнила:

Савицкая А.Ю., ПО-7

Проверил:

Бойко Д.О.

Брест,2021

### Цель: 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

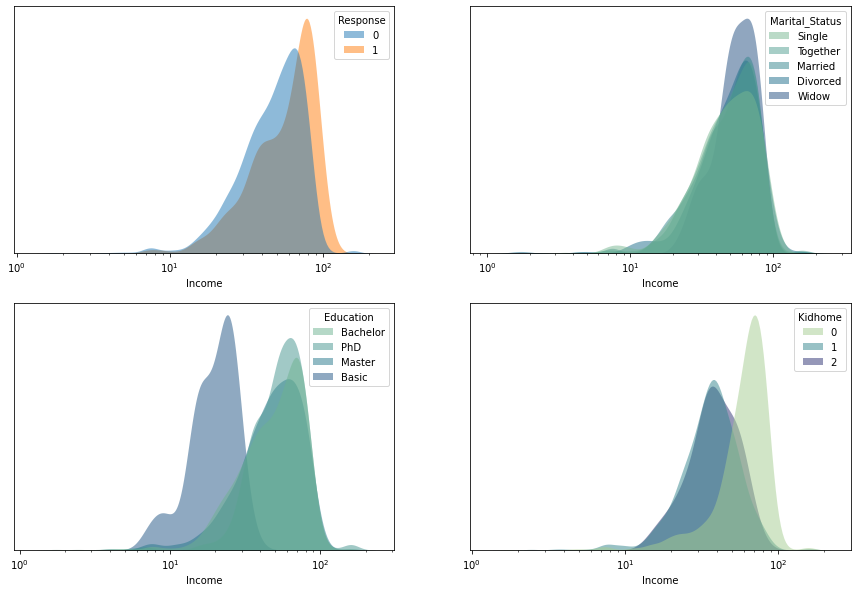
3. Проверка наличия NULL-данных. При их наличии вывести на экран

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
4. Построить heatmap для всех числовых колонок:

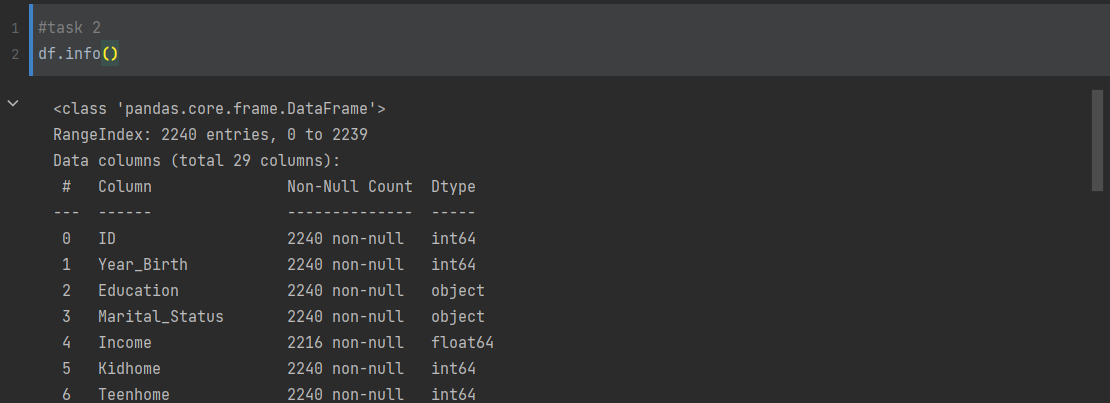


**Задание 1:**

df = pd.read\_csv("dataset.csv", delimiter="\t")

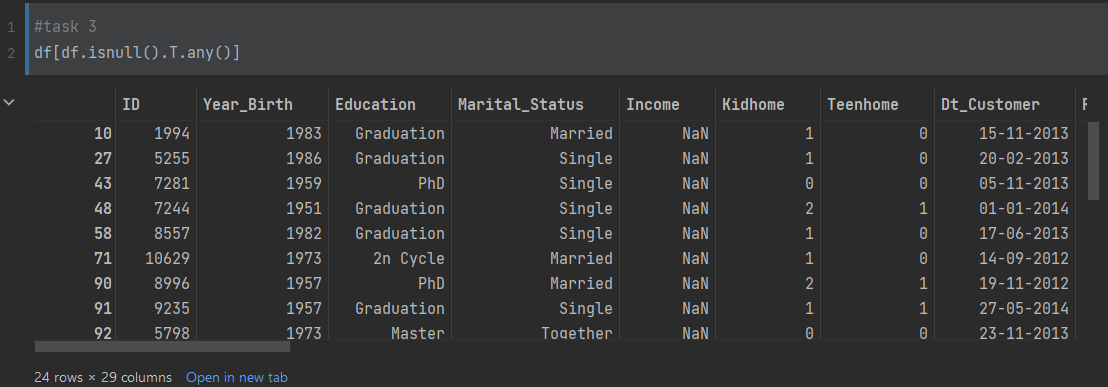
**Задание 2:**

df.info()

****

**Задание 3:**

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



**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)

)

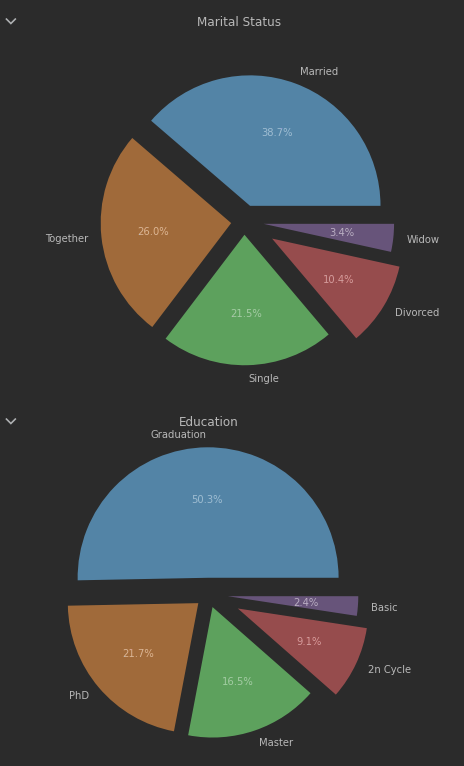
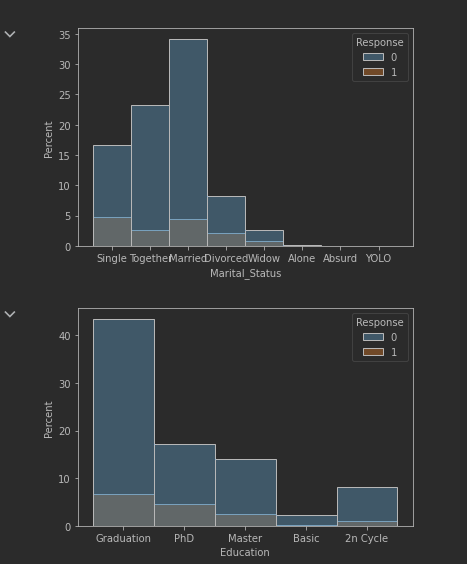
plt.show()

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:

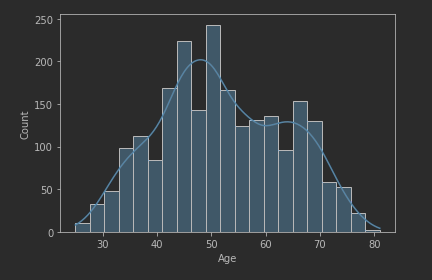
return int(datetime.today().strftime("%Y")) - int(born)

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()

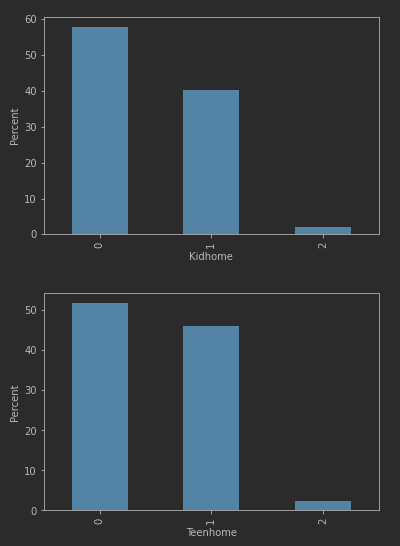
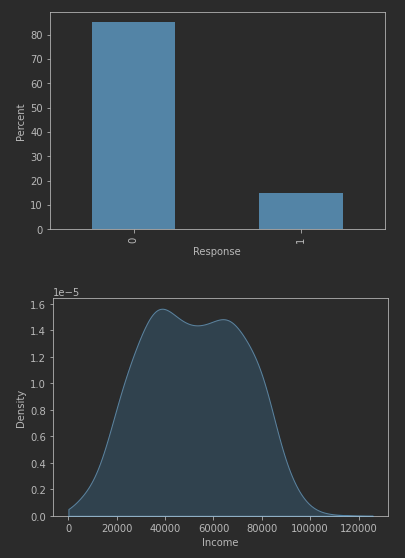
show\_bar("Kidhome")

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)

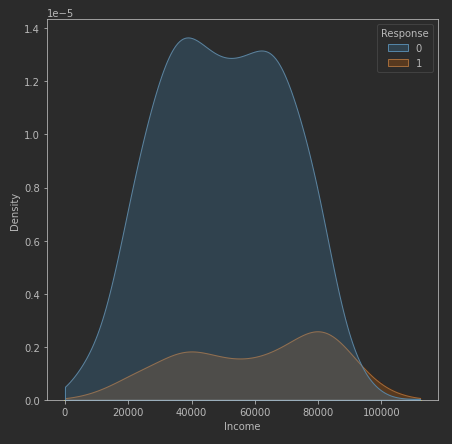
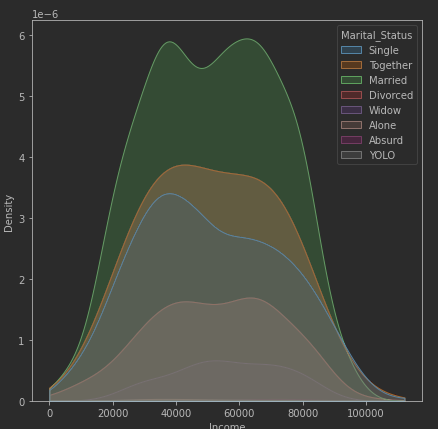
plt.show()

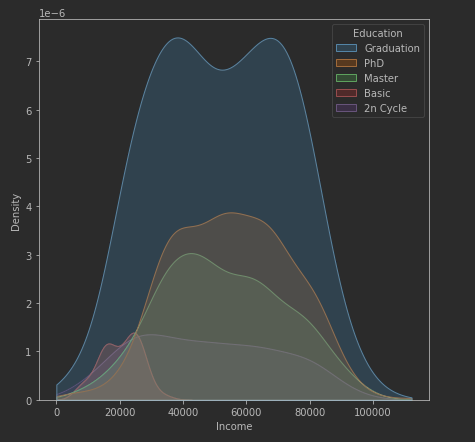
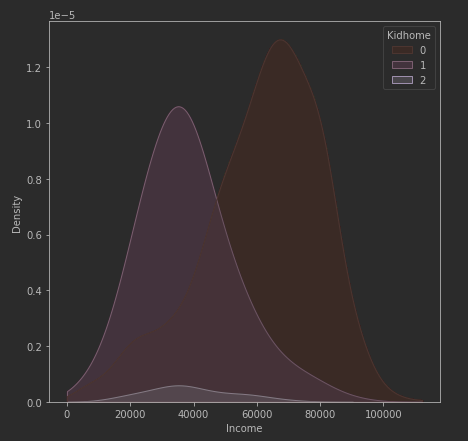
show\_kde("Response")

show\_kde("Marital\_Status")

show\_kde("Education")

show\_kde("Kidhome")



****

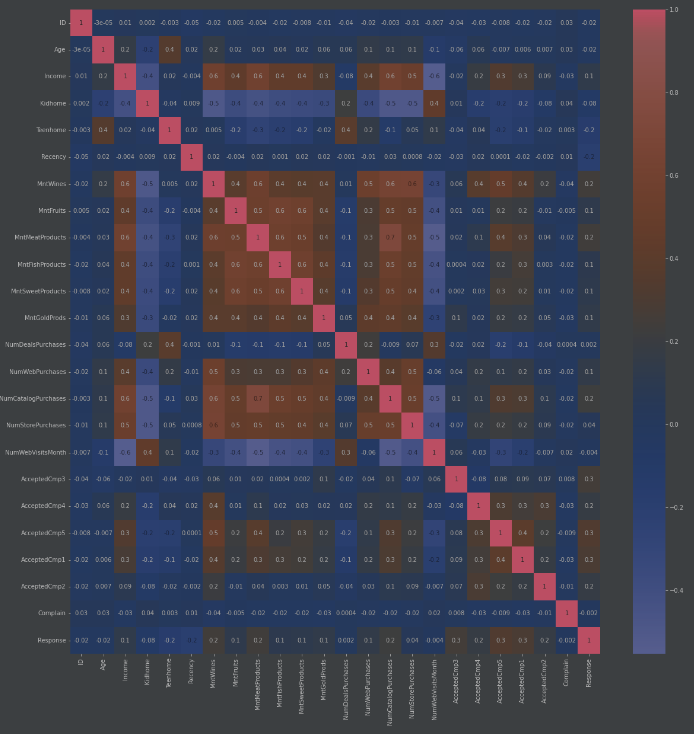
**Задание 10:**

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()

**Код программы:**

**import** numpy **as** np

**import** pandas **as** pd

**from** pandas **import** DataFrame

**import** matplotlib.pyplot **as** plt

**import** seaborn **as** sns

**from** scipy **import** stats

**from** datetime **import** datetime

In [2]:

*#task 1*

df **=** pd**.**read\_csv("marketing\_campaign.csv", delimiter**=**"\t")

In [3]:

df**.**sample(10)

Out[3]:

|  | **ID** | **Year\_Birth** | **Education** | **Marital\_Status** | **Income** | **Kidhome** | **Teenhome** | **Dt\_Customer** | **Recency** | **MntWines** | **...** | **NumWebVisitsMonth** | **AcceptedCmp3** | **AcceptedCmp4** | **AcceptedCmp5** | **AcceptedCmp1** | **AcceptedCmp2** | **Complain** | **Z\_CostContact** | **Z\_Revenue** | **Response** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **211** | 6521 | 1954 | Graduation | Together | 77972.0 | 0 | 0 | 18-03-2014 | 18 | 613 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **2194** | 7118 | 1957 | Graduation | Married | 73803.0 | 0 | 1 | 01-08-2012 | 61 | 833 | ... | 6 | 1 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 1 |
| **1907** | 10680 | 1986 | Graduation | Together | 33235.0 | 1 | 0 | 05-02-2014 | 71 | 45 | ... | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **1355** | 3834 | 1962 | Graduation | Single | 69627.0 | 0 | 1 | 17-03-2013 | 35 | 231 | ... | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **2025** | 9423 | 1979 | Master | Married | 32765.0 | 1 | 0 | 23-02-2014 | 49 | 13 | ... | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **1522** | 1998 | 1976 | Graduation | Single | 37697.0 | 1 | 0 | 07-02-2014 | 82 | 34 | ... | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **422** | 6528 | 1982 | Master | Together | 18492.0 | 1 | 0 | 05-06-2014 | 75 | 2 | ... | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **1455** | 4399 | 1969 | Graduation | Together | 68695.0 | 0 | 0 | 25-06-2014 | 3 | 458 | ... | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **387** | 6202 | 1960 | Graduation | Together | 63381.0 | 0 | 1 | 05-10-2012 | 78 | 571 | ... | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **1239** | 286 | 1952 | Graduation | Single | 44213.0 | 1 | 1 | 29-11-2013 | 48 | 95 | ... | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |

10 rows × 29 columns

In [4]:

*#task 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

7 Dt\_Customer 2240 non-null object

8 Recency 2240 non-null int64

9 MntWines 2240 non-null int64

10 MntFruits 2240 non-null int64

11 MntMeatProducts 2240 non-null int64

12 MntFishProducts 2240 non-null int64

13 MntSweetProducts 2240 non-null int64

14 MntGoldProds 2240 non-null int64

15 NumDealsPurchases 2240 non-null int64

16 NumWebPurchases 2240 non-null int64

17 NumCatalogPurchases 2240 non-null int64

18 NumStorePurchases 2240 non-null int64

19 NumWebVisitsMonth 2240 non-null int64

20 AcceptedCmp3 2240 non-null int64

21 AcceptedCmp4 2240 non-null int64

22 AcceptedCmp5 2240 non-null int64

23 AcceptedCmp1 2240 non-null int64

24 AcceptedCmp2 2240 non-null int64

25 Complain 2240 non-null int64

26 Z\_CostContact 2240 non-null int64

27 Z\_Revenue 2240 non-null int64

28 Response 2240 non-null int64

dtypes: float64(1), int64(25), object(3)

memory usage: 507.6+ KB

In [5]:

*#task 3*

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

Out[5]:

|  | **ID** | **Year\_Birth** | **Education** | **Marital\_Status** | **Income** | **Kidhome** | **Teenhome** | **Dt\_Customer** | **Recency** | **MntWines** | **...** | **NumWebVisitsMonth** | **AcceptedCmp3** | **AcceptedCmp4** | **AcceptedCmp5** | **AcceptedCmp1** | **AcceptedCmp2** | **Complain** | **Z\_CostContact** | **Z\_Revenue** | **Response** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **10** | 1994 | 1983 | Graduation | Married | NaN | 1 | 0 | 15-11-2013 | 11 | 5 | ... | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **27** | 5255 | 1986 | Graduation | Single | NaN | 1 | 0 | 20-02-2013 | 19 | 5 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **43** | 7281 | 1959 | PhD | Single | NaN | 0 | 0 | 05-11-2013 | 80 | 81 | ... | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **48** | 7244 | 1951 | Graduation | Single | NaN | 2 | 1 | 01-01-2014 | 96 | 48 | ... | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **58** | 8557 | 1982 | Graduation | Single | NaN | 1 | 0 | 17-06-2013 | 57 | 11 | ... | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **71** | 10629 | 1973 | 2n Cycle | Married | NaN | 1 | 0 | 14-09-2012 | 25 | 25 | ... | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **90** | 8996 | 1957 | PhD | Married | NaN | 2 | 1 | 19-11-2012 | 4 | 230 | ... | 9 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **91** | 9235 | 1957 | Graduation | Single | NaN | 1 | 1 | 27-05-2014 | 45 | 7 | ... | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **92** | 5798 | 1973 | Master | Together | NaN | 0 | 0 | 23-11-2013 | 87 | 445 | ... | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **128** | 8268 | 1961 | PhD | Married | NaN | 0 | 1 | 11-07-2013 | 23 | 352 | ... | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **133** | 1295 | 1963 | Graduation | Married | NaN | 0 | 1 | 11-08-2013 | 96 | 231 | ... | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **312** | 2437 | 1989 | Graduation | Married | NaN | 0 | 0 | 03-06-2013 | 69 | 861 | ... | 3 | 0 | 1 | 0 | 1 | 0 | 0 | 3 | 11 | 0 |
| **319** | 2863 | 1970 | Graduation | Single | NaN | 1 | 2 | 23-08-2013 | 67 | 738 | ... | 7 | 0 | 1 | 0 | 1 | 0 | 0 | 3 | 11 | 0 |
| **1379** | 10475 | 1970 | Master | Together | NaN | 0 | 1 | 01-04-2013 | 39 | 187 | ... | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **1382** | 2902 | 1958 | Graduation | Together | NaN | 1 | 1 | 03-09-2012 | 87 | 19 | ... | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **1383** | 4345 | 1964 | 2n Cycle | Single | NaN | 1 | 1 | 12-01-2014 | 49 | 5 | ... | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **1386** | 3769 | 1972 | PhD | Together | NaN | 1 | 0 | 02-03-2014 | 17 | 25 | ... | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **2059** | 7187 | 1969 | Master | Together | NaN | 1 | 1 | 18-05-2013 | 52 | 375 | ... | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **2061** | 1612 | 1981 | PhD | Single | NaN | 1 | 0 | 31-05-2013 | 82 | 23 | ... | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **2078** | 5079 | 1971 | Graduation | Married | NaN | 1 | 1 | 03-03-2013 | 82 | 71 | ... | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **2079** | 10339 | 1954 | Master | Together | NaN | 0 | 1 | 23-06-2013 | 83 | 161 | ... | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **2081** | 3117 | 1955 | Graduation | Single | NaN | 0 | 1 | 18-10-2013 | 95 | 264 | ... | 7 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |
| **2084** | 5250 | 1943 | Master | Widow | NaN | 0 | 0 | 30-10-2013 | 75 | 532 | ... | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 3 | 11 | 1 |
| **2228** | 8720 | 1978 | 2n Cycle | Together | NaN | 0 | 0 | 12-08-2012 | 53 | 32 | ... | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 3 | 11 | 0 |

24 rows × 29 columns

In [6]:

*#task 4*

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

In [7]:

*#task 5*

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

In [8]:

*#task 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**=False**,

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**=False**,

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()

In [9]:

*#task 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()

In [10]:

*#task 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()

In [11]:

*#task 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")

C:\Users\wwwmi\AppData\Roaming\Python\Python39\site-packages\seaborn\distributions.py:316: UserWarning: Dataset has 0 variance; skipping density estimate. Pass `warn\_singular=False` to disable this warning.

warnings.warn(msg, UserWarning)

In [12]:

*#task 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 ознакомилась с основами библиотеки pandas и научилась строить графики с использованием библиотек matplotlib.pyplot и seaborn.