Министерство образования Республики Беларусь

Учреждение образования

“Брестский государственный университет”

Кафедра ИИТ

Лабораторная работа №5

По дисциплине: “ЯП”

Тема: “ Python. Основы Pandas ”

Выполнил:

Студент 2-го курса

Группы ПО-7

Практика Д.О

Проверил:

Бойко Д.О.

Брест, 2021

**Вариант 5**

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

**Список заданий:**

1. Загрузить датасет в pandas и проверить на доступность

df = pd.read\_csv("marketing\_campaign.csv", delimiter="\t")  
df.sample(15)

1. Вывести общую информацию о датасете

df.info()

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

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

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

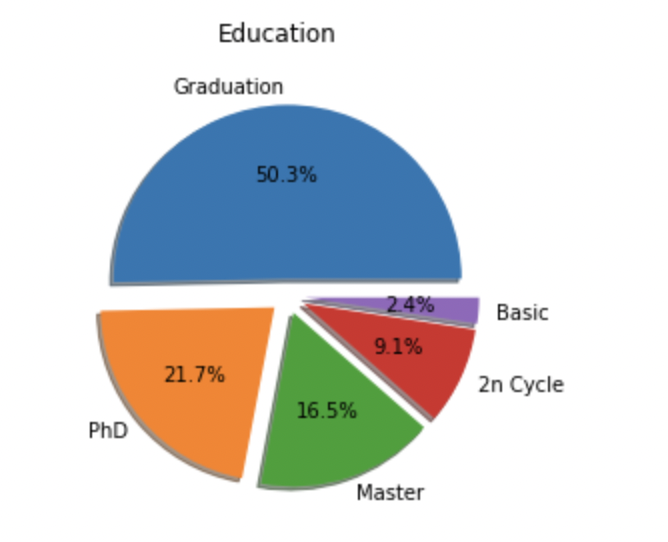
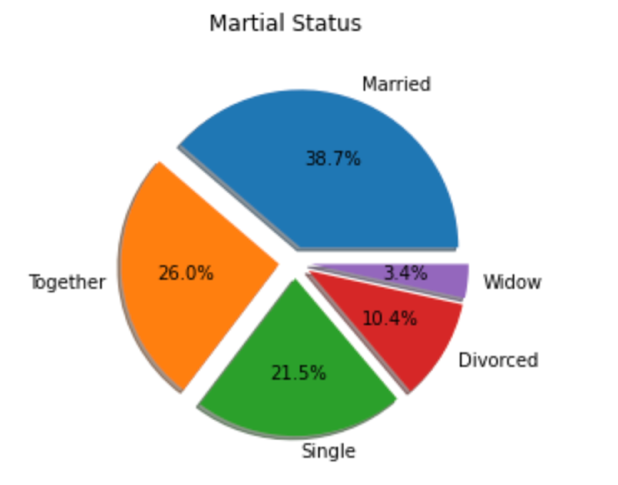
df.drop(["Z\_CostContact", "Z\_Revenue"], axis=1, inplace=**True**)

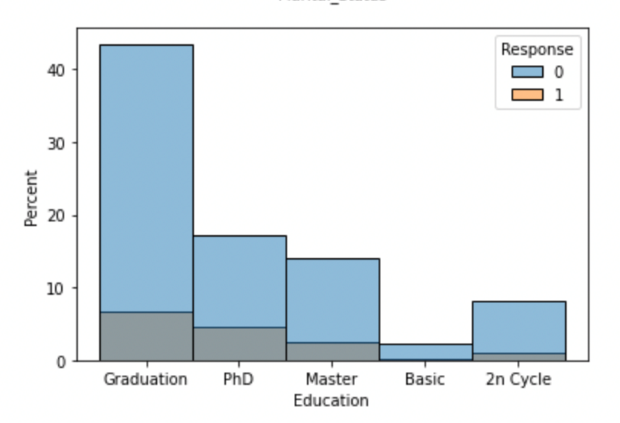
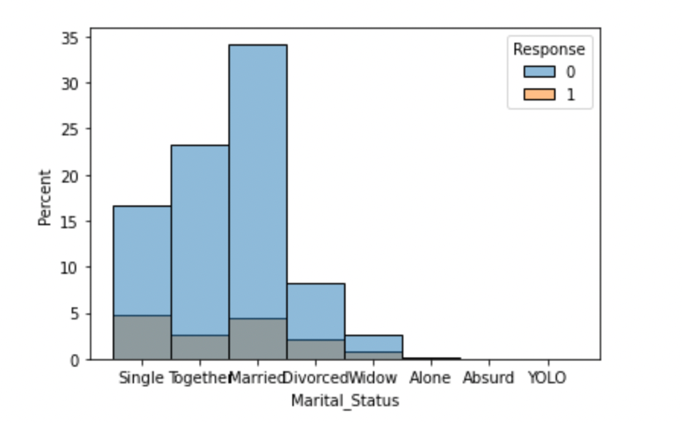
1. Переименовать колонку "Year\_Birth" в "Age"

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

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

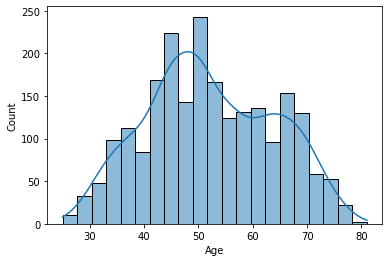
ms\_df\_c = df["Marital\_Status"].value\_counts().to\_frame()  
ms\_df = ms\_df\_c[ms\_df\_c["Marital\_Status"] > 50]  
ed\_df = df["Education"].value\_counts().to\_frame()  
  
ms\_df[ms\_df["Marital\_Status"] > 50].plot(  
 kind="pie",  
 autopct="%1.1f%%",  
 legend=**None**,  
 ylabel="",  
 subplots=**True**,  
 shadow=**True**,  
 explode=(0.1, 0.1, 0.1, 0.1, 0.1),  
 title="Martial Status",  
)  
plt.show()  
  
ed\_df.plot(  
 kind="pie",  
 autopct="%1.1f%%",  
 legend=**None**,  
 ylabel="",  
 subplots=**True**,  
 shadow=**True**,  
 explode=(0.1, 0.1, 0.1, 0.1, 0.1),  
 title="Education",  
)  
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()





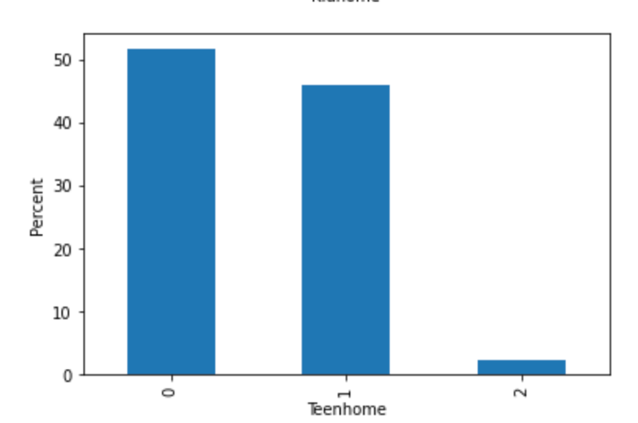
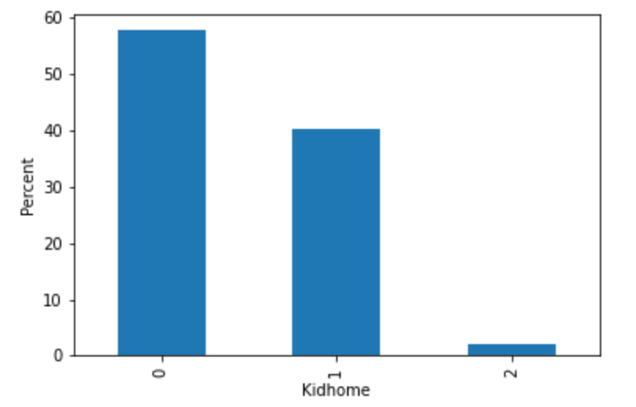
1. Создать гистаграмму по колонке "Age" и оценить на распределение по Гауссу.

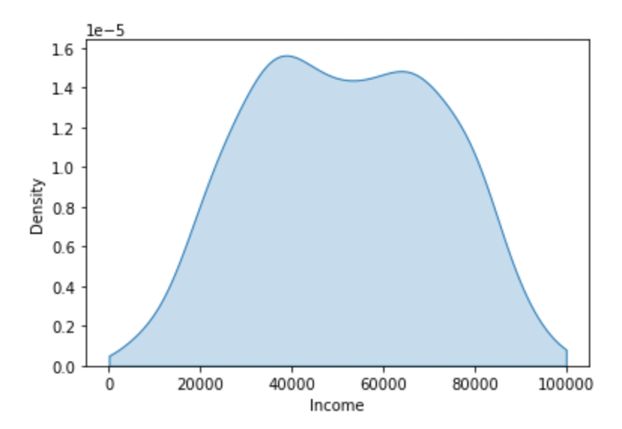
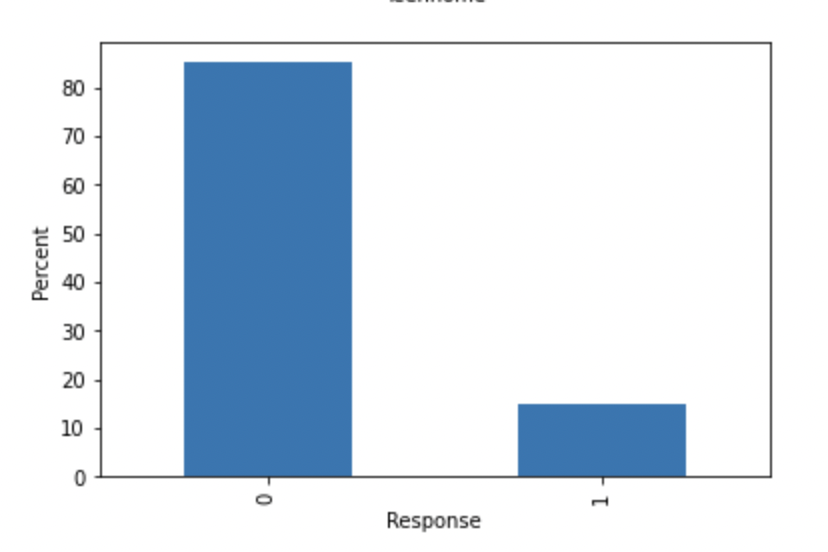
**def** calculate\_age(born) -> int:  
 **return** int(datetime.today().strftime("%Y")) - int(born)  
  
df["Age"] = df["Age"].apply(calculate\_age)  
filtered\_df = df[df["Age"] < 100]  
sns.histplot(x=filtered\_df["Age"], kde=**True**)  
plt.show()

**

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

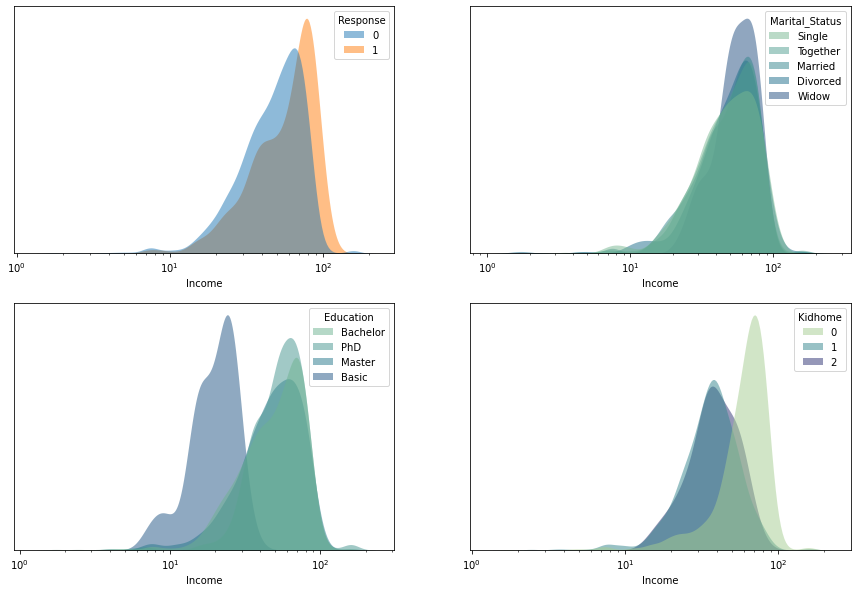
**def** show\_info(data\_name: str) -> **None**:  
 df[data\_name].value\_counts(normalize=**True**).mul(100).to\_frame().plot(  
 kind="bar",  
 legend=**None**,  
 xlabel=data\_name,  
 ylabel="Percent"  
 )  
 plt.show()  
  
  
show\_info ("Kidhome")  
show\_info ("Teenhome")  
show\_info ("Response")  
sns.kdeplot(df["Income"], shade=**True**, clip=(10 \*\* 0, 10 \*\* 5))  
plt.show()

**

**

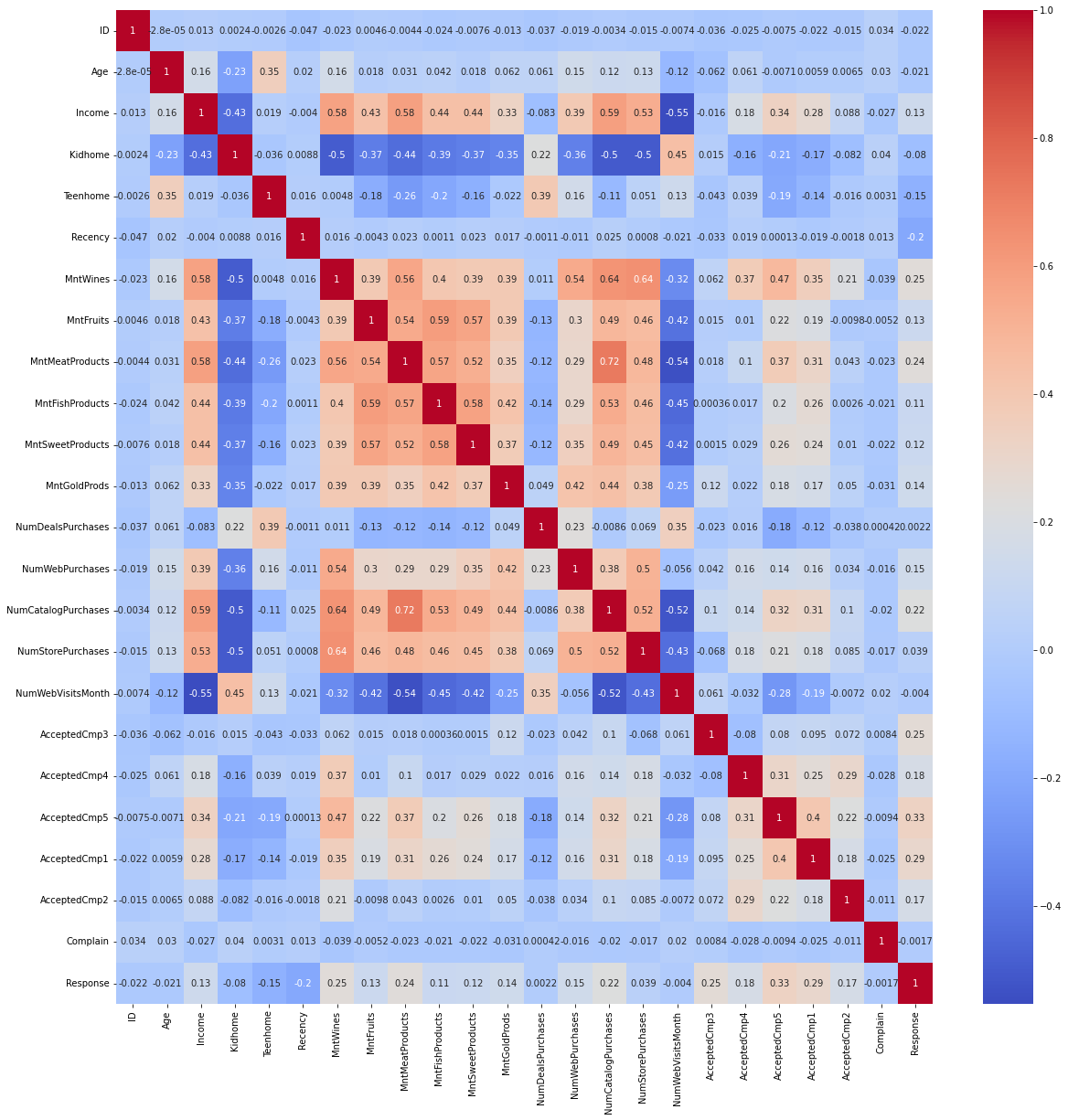
1. Построить графики "Response", "Marital\_Status", "Education" и "Kidhome" по образцу:

**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")



1. Построить heatmap для всех числовых колонок:

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**)  
plt.show()

**

**Вывод:** Входе лабораторной работы изучил основные способы построения диаграмм, гистограмм и heatmap. И полностью осознал всю важность работы с pandas