THỰC HÀNH: PHÂN TÍCH KHÁM PHÁ DỮ LIỆU

1.1. THỐNG KÊ MÔ TẢ

1.1.1. Ôn tập lý thuyết

1.1.2. Bài làm mẫu

1.1.3. Bài tập thực hành 1

Thực hiện thống kê mô tả trên tập dữ liệu về phân loại chất lượng rượu đỏ. Dữ liệu lấy tại *https://www.kaggle.com/code/eisgandar/red-wine-quality-eda-classification*

Nhiệm vụ 1: Khám phá dữ liệu chất lượng rượu vang đỏ

```
Mean: 10.422983114446529
Median: 10.2
Mode: 9.5 (count: 139 )
Variance: 1.1356473950004693
Standard Deviation: 1.0656675818473926
Max: 14.9 Min: 8.4
60th Percentile: 10.5
Q3 (0.75 quantile): 11.1
Interquartile Range (IQR): 1.5999999999999996
```

Nhiệm vụ 2: Loại bỏ dữ liệu trùng lặp

Out[4]:

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67.0	0.99680	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54.0	0.99700	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60.0	0.99800	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34.0	0.99780	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...
1594	6.2	0.600	0.08	2.0	0.090	32.0	44.0	0.99490	0.58	10.5	5
1595	5.9	0.550	0.10	2.2	0.062	39.0	51.0	0.99512	0.76	11.2	6
1596	6.3	0.510	0.13	2.3	0.076	29.0	40.0	0.99574	0.75	11.0	6
1597	5.9	0.645	0.12	2.0	0.075	32.0	44.0	0.99547	0.71	10.2	5
1598	6.0	0.310	0.47	3.6	0.067	18.0	42.0	0.99549	0.66	11.0	6

1599 rows × 11 columns

Nhiệm vụ 3: Thay thế dữ liệu và thay đổi đinh dang của dữ liệu

In [5]:

```
# thay giá trị 3,4,5 của quality thành low, low, medium
wine_data['quality_replaced'] = wine_data['quality'].replace([3,4,5], ['low','low','medium'])
# điền giá trị 0 cho các ô trống trong cột alcohol
wine_data['alcohol'] = wine_data['alcohol'].fillna(0)
# chuyển kiểu dữ liệu alcohol sang int
wine_data['alcohol_changed'] = wine_data['alcohol'].astype(int)
```

Nhiệm vụ 4: Xử lý dữ liệu thiếu

In [6]:
```
# đếm số giá trị thiếu của từng cột
wine_data.isnull().sum()
# xóa các hàng có giá trị thiếu
wine_data_withoutna = wine_data.dropna(how='any')
# xem kích thước dữ liệu sau khi xóa
wine_data_withoutna.shape
```

Out[6]: (1599, 14)

1.1.4. Bài tập thực hành 2

Thực hiện thống kê mô tả trên tập dữ liệu về bệnh tiểu đường. Dữ liệu lấy tại https://www.kaggle.com/code/vincentlugat/pima-indians-diabetes-eda-prediction-0-906

Nhiệm vụ 1: Khám phá dữ liệu về bệnh tiểu đường

```
Glucose - Mean: 120.89453125
Glucose - Median: 117.0
Glucose - Mode: 99 (count: 17 )
Glucose - Variance: 1022.2483142519557
Glucose - Std: 31.97261819513622
Glucose - Min: 0 Max: 199 Range: 199
Glucose - 60th percentile: 125.0
Glucose - Q1: 99.0 Q3: 140.25 IQR: 41.25
```

Nhiệm vụ 2: Loại bỏ dữ liệu trùng lặp

Out[8]:

	Pregnancies	Glucose	BloodPressure	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	0	33.6	0.627	50	1
1	1	85	66	0	26.6	0.351	31	0
2	8	183	64	0	23.3	0.672	32	1
3	1	89	66	94	28.1	0.167	21	0
4	0	137	40	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...
763	10	101	76	180	32.9	0.171	63	0
764	2	122	70	0	36.8	0.340	27	0
765	5	121	72	112	26.2	0.245	30	0
766	1	126	60	0	30.1	0.349	47	1
767	1	93	70	0	30.4	0.315	23	0

768 rows × 8 columns

Nhiệm vụ 3: Thay thế dữ liệu và thay đổi định dạng của dữ liệu

In [9]:
```
# Thay giá trị 0/1 của 'Outcome' thành chuỗi 'no_diabetes'/'diabetes'
diabetes['Outcome_replaced'] = diabetes['Outcome'].replace([0,1], ['no_diabetes','diabetes'])
# Điền giá trị thiếu trong cột 'Insulin' bằng 0
diabetes['Insulin'] = diabetes['Insulin'].fillna(0)
# Chuyển kiểu dữ liệu cột 'Insulin' sang int
diabetes['Insulin_changed'] = diabetes['Insulin'].astype(int)
```

Nhiệm vụ 4: Xử lý dữ liệu thiếu

Out[10]: (768, 11)

1.2. XỬ LÝ VÀ TRỰC QUAN HÓA DỮ LIỆU

1.2.1. Ôn tập lý thuyết

1.2.2. Bài làm mẫu

1.2.3 Bài tập thực hành 1

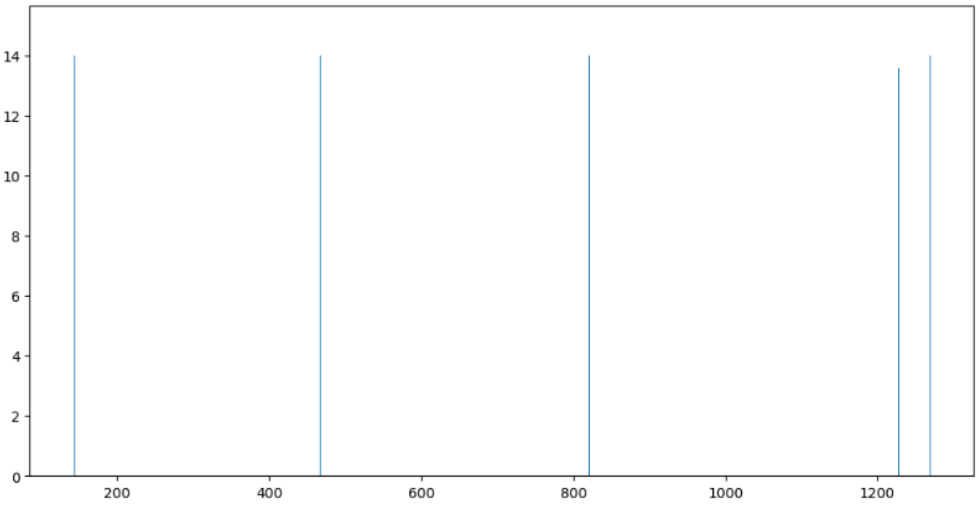
+ Thực hiện trực quan hóa dữ liệu trên tập dữ liệu về phân loại chất lượng rượu đỏ.
Dữ liệu lấy tại https://www.kaggle.com/code/eisgandar/red-wine-quality-eda-classification

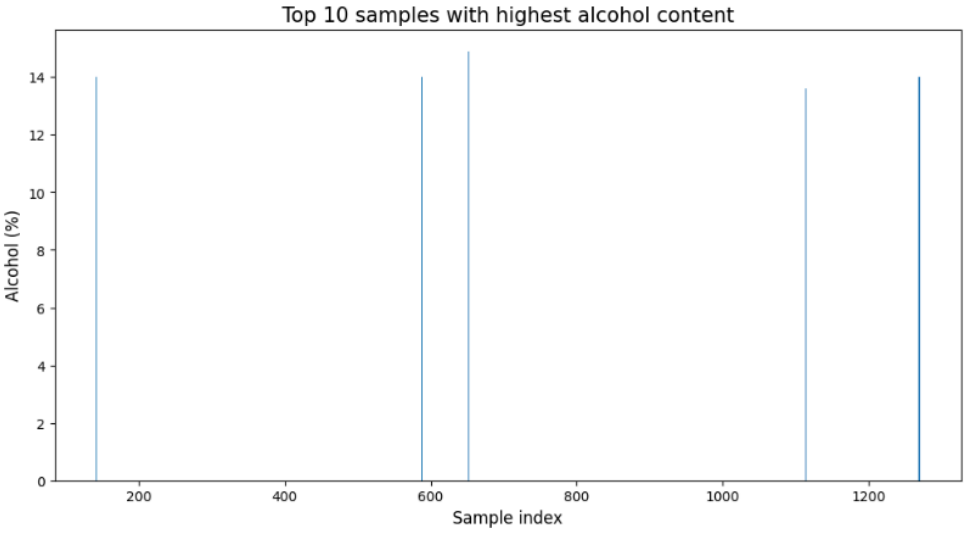
Nhiệm vụ 1: Chuẩn bị dữ liệu cho trực quan hóa dữ liệu

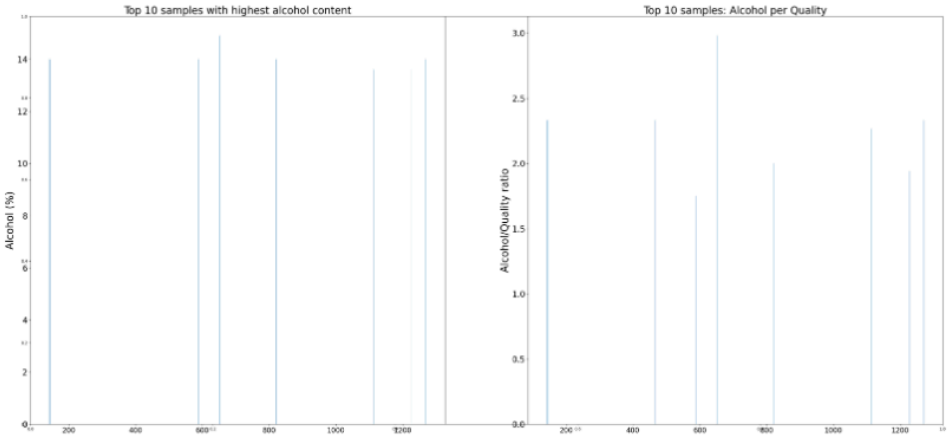
1. Chuẩn bị dữ liệu cho trực quan hóa dữ liệu

In [11]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Đọc file rượu vang đỏ (đặt file csv cùng thư mục làm việc hoặc thay đường dẫn cho đúng)
wine_data = pd.read_csv("winequality-red.csv")

# Chọn các cột cần thiết: alcohol (nồng độ cồn) và quality (điểm chất lượng)
wine_data = wine_data[['alcohol', 'quality']]

# Tạo biến alcohol_per_quality (nồng độ cồn chia cho điểm chất lượng) để so sánh
wine_data['alcohol_per_quality'] = wine_data['alcohol'] / wine_data['quality']
```

Nhiệm vụ 2: Trực quan hóa dữ liệu với thư viện Matplotlib

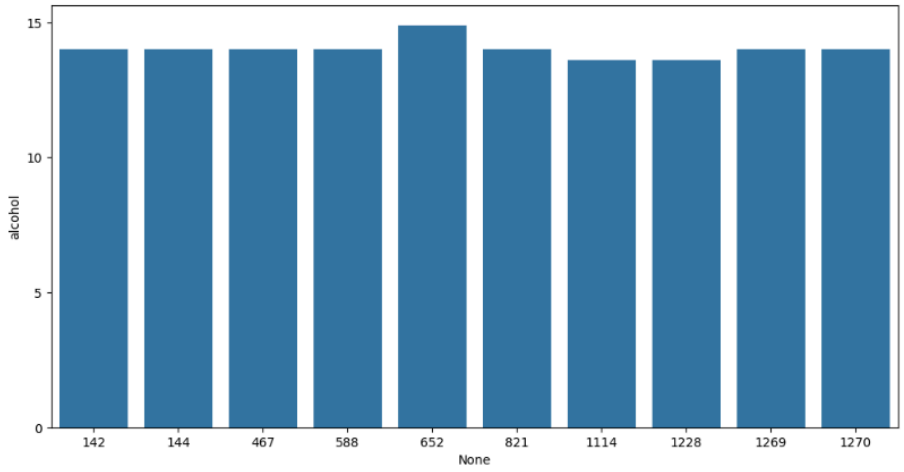

0
2
4
6
8
10
12
14
200
400
600
800
1000
1200

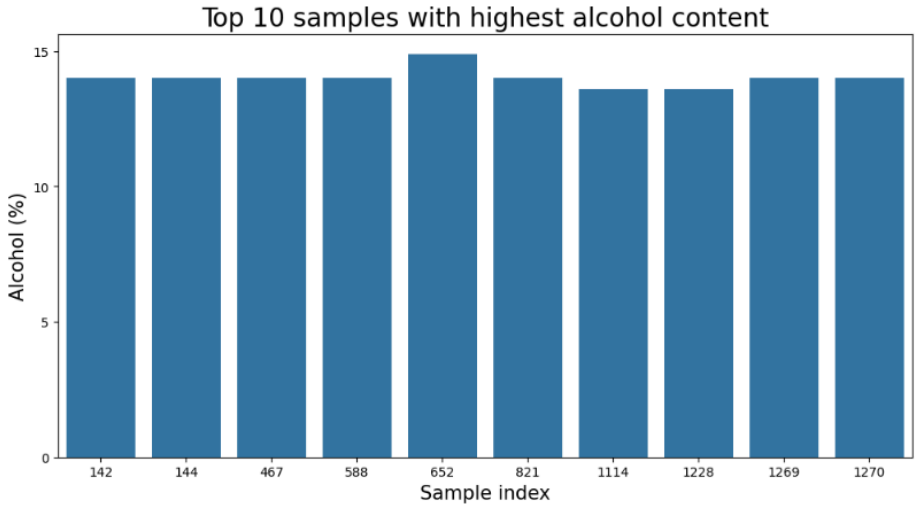

Top 10 samples with highest alcohol content
Alcohol (%)
Sample index
0
2
4
6
8
10
12
14
200
400
600
800
1000
1200


Top 10 samples with highest alcohol content
Alcohol (%)
Top 10 samples: Alcohol per Quality
Alcohol/Quality ratio

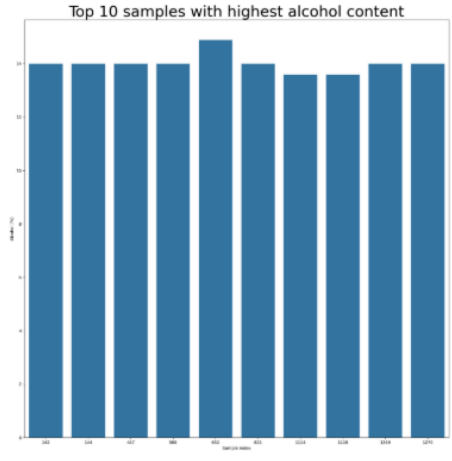
Nhiệm vụ 3: Trực quan hóa dữ liệu với thư viện Seaborn

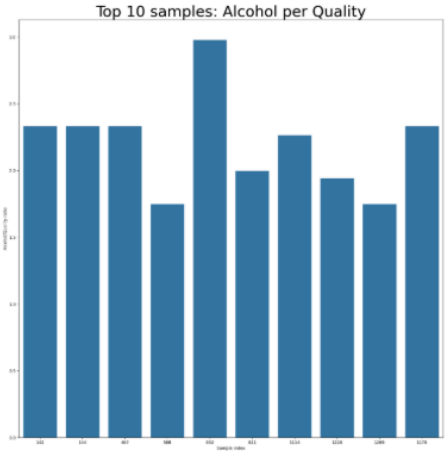
```
Out[13]:  Text(0.5, 1.0, 'Top 10 samples: Alcohol per Quality')
```





Sample index





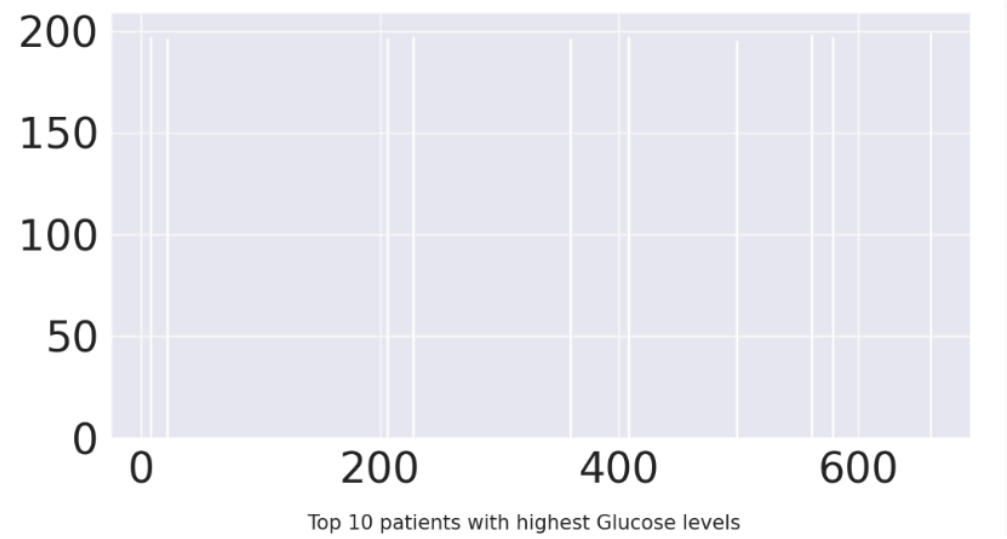
1.2.3.Bài tập thực hành 2

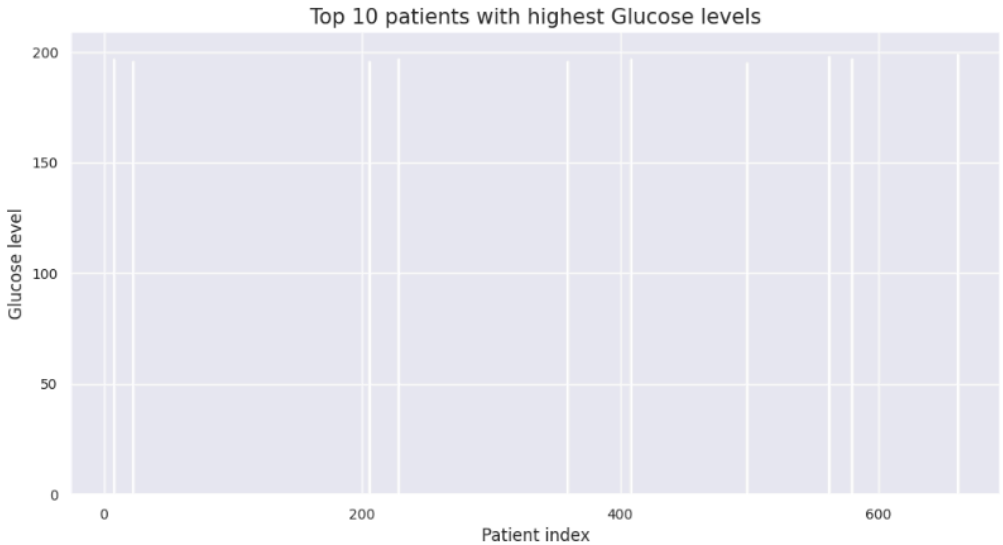
+ Thực hiện trực quan hóa dữ liệu trên tập dữ liệu về bệnh tiểu đường. Dữ liệu lấy tại https://www.kaggle.com/code/vincentlugat/pima-indians-diabetes-eda-prediction-0-906

Nhiệm vụ 1: Chuẩn bị dữ liệu cho trực quan hóa dữ liệu

In [14]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Đọc file dữ liệu bệnh tiểu đường (đặt file 'diabetes.csv' cùng thư mục làm việc)
diabetes_data = pd.read_csv("diabetes.csv")

# Chọn các cột quan trọng: Glucose và BMI
diabetes_data = diabetes_data[['Glucose', 'BMI']]

# Tạo biến Glucose_per_BMI (tỉ lệ Glucose/BMI) để so sánh
diabetes_data['Glucose_per_BMI'] = diabetes_data['Glucose'] / diabetes_data['BMI']
```

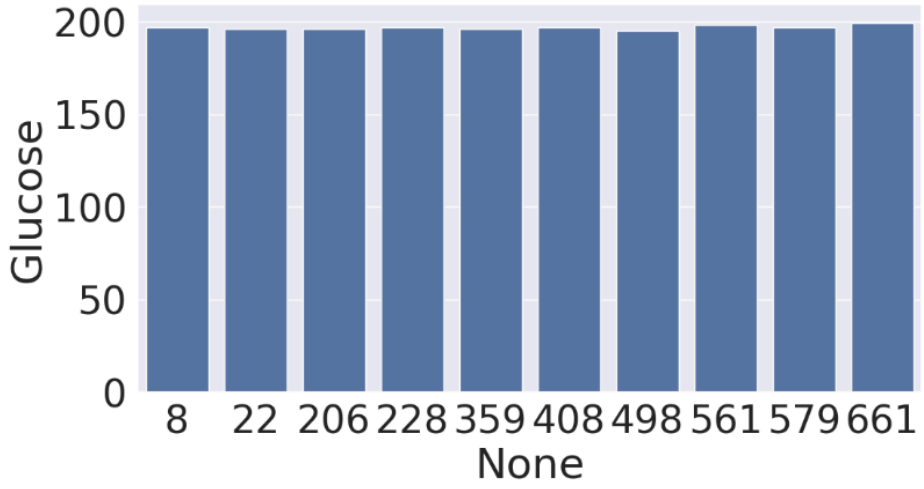
Nhiệm vụ 2. Trực quan hóa dữ liệu với thư viện Matplotlib

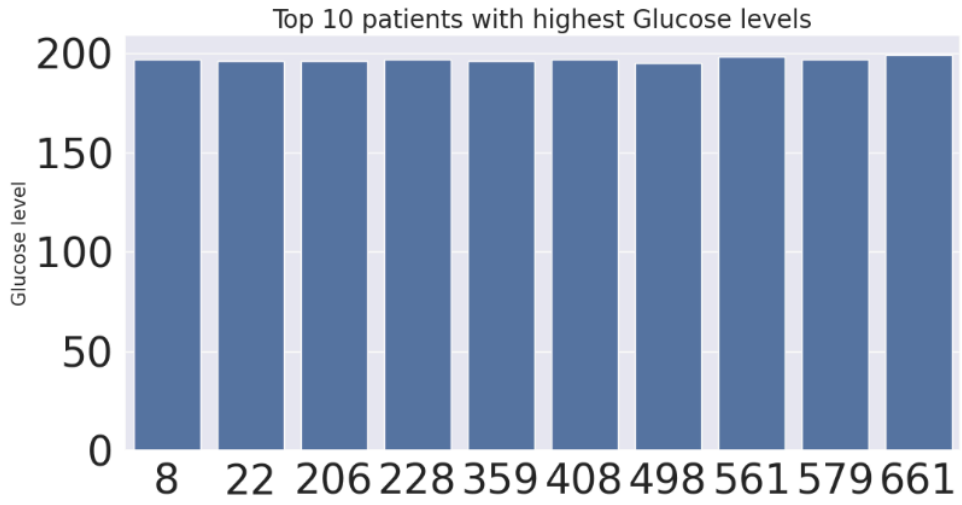


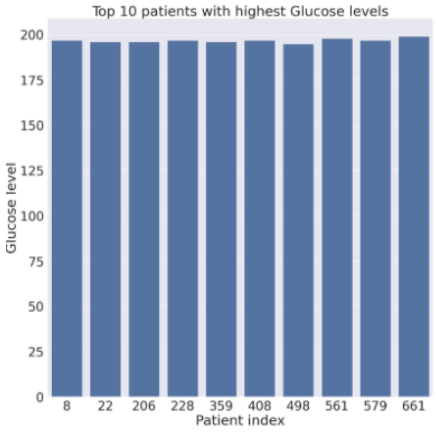


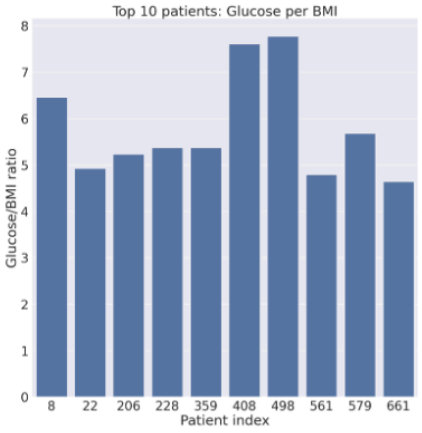
3. Trực quan hóa dữ liệu với thư viện Seaborn

```
Out[16]: Text(0.5, 1.0, 'Top 10 patients: Glucose per BMI')
```





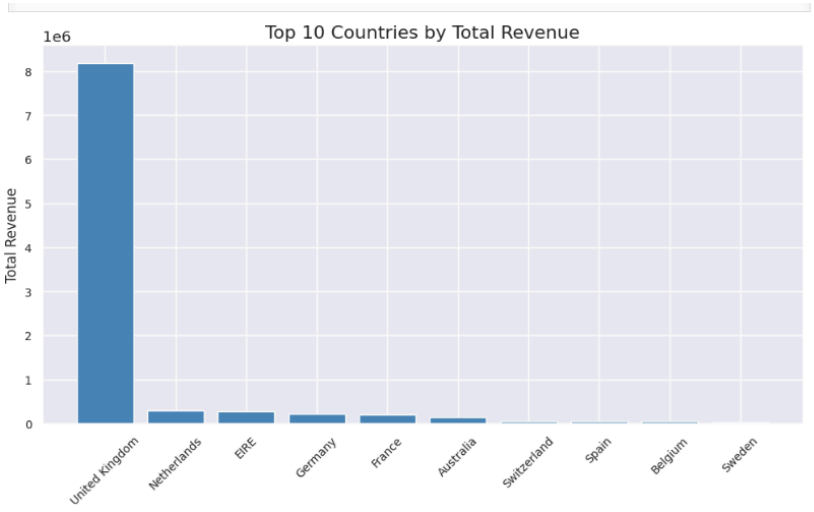


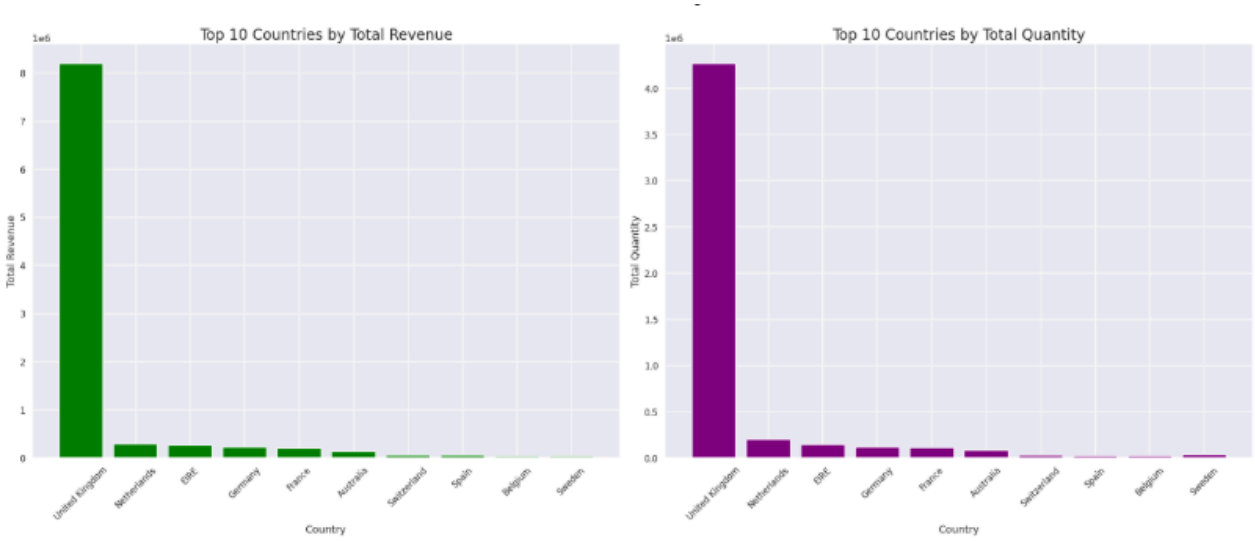


+ Thực hiện EDA trên tập dữ liệu mua sắm tại siêu thị. Tập dữ liệu lấy từ https://www.kaggle.com/code/rajatkumar30/eda-online-retail

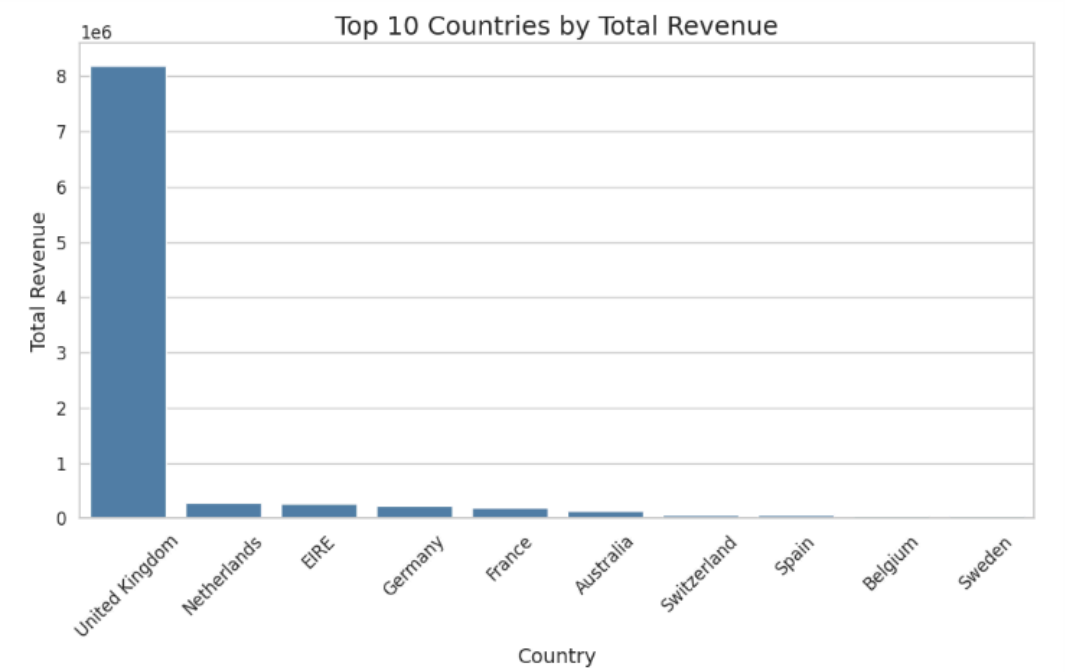
```python
# 1. Chuẩn bị dữ liệu cho trực quan hóa
# ======================================
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Đọc dữ liệu Excel
df = pd.read_excel("Online Retail.xlsx")

# Lọc các cột quan trọng
retail_data = df[['Country', 'Quantity', 'UnitPrice']].copy()

# Tạo cột doanh thu = Quantity * UnitPrice
retail_data['TotalPrice'] = retail_data['Quantity'] * retail_data['UnitPrice']

# Gom nhóm theo quốc gia để tính tổng doanh thu và tổng số lượng
country_revenue = retail_data.groupby('Country', as_index=False)['TotalPrice'].
sum()
country_revenue = country_revenue.sort_values('TotalPrice', ascending=False)

country_quantity = retail_data.groupby('Country', as_index=False)['Quantity'].
sum()
country_quantity = country_quantity.sort_values('Quantity', ascending=False)
```

2. Trực quan hóa dữ liệu với Matplotlib

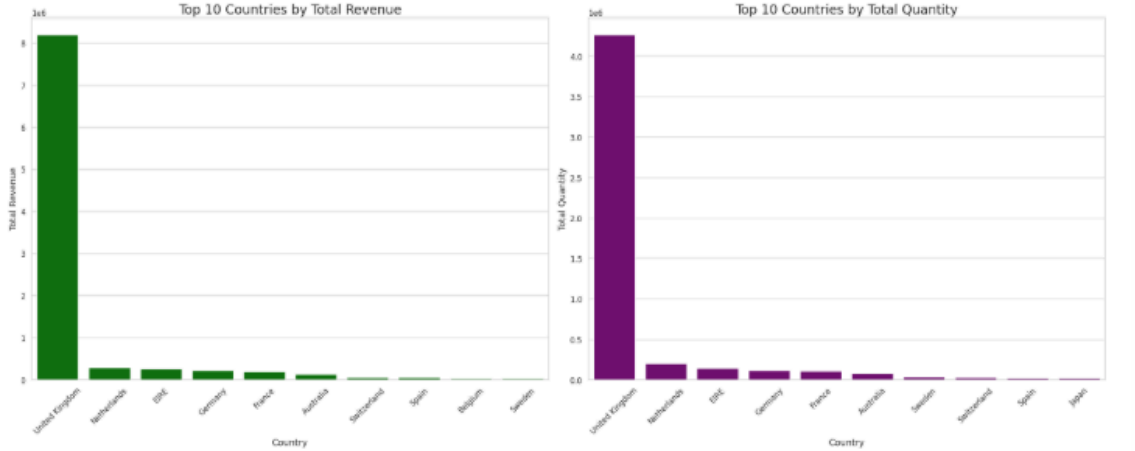




3. Trực quan hóa dữ liệu với Seaborn



```
/tmp/ipython-input-3807761161.py:25: UserWarning: set_ticklabels() should only be used with a fixed number of ticks, i.e. a
fter set_ticks() or using a FixedLocator.
  ax1.set_xticklabels(ax1.get_xticklabels(), rotation=45, fontsize=12)
/tmp/ipython-input-3807761161.py:31: UserWarning: set_ticklabels() should only be used with a fixed number of ticks, i.e. a
fter set_ticks() or using a FixedLocator.
  ax2.set_xticklabels(ax2.get_xticklabels(), rotation=45, fontsize=12)
```



1.3. Phân tích đơn biến và hai biến

1.3.1. Ôn lý thuyết

1.3.2. Bài làm mẫu

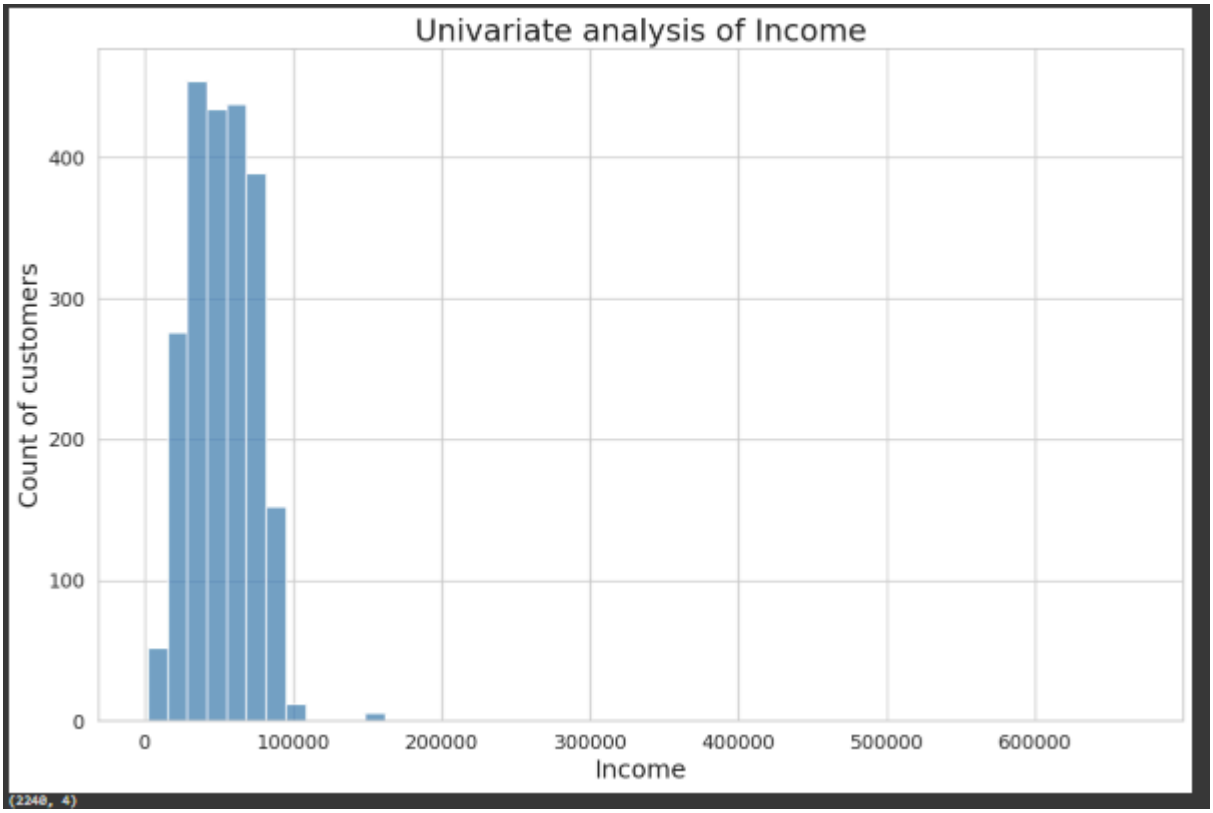
1.3.3. Bài tập thực hành 1

Tìm hiểu các tính năng và cách sử dụng sản phẩm SweetViz (https://pypi.org/project/sweetviz) áp dụng trên tập dữ liệu Marketing Campaign

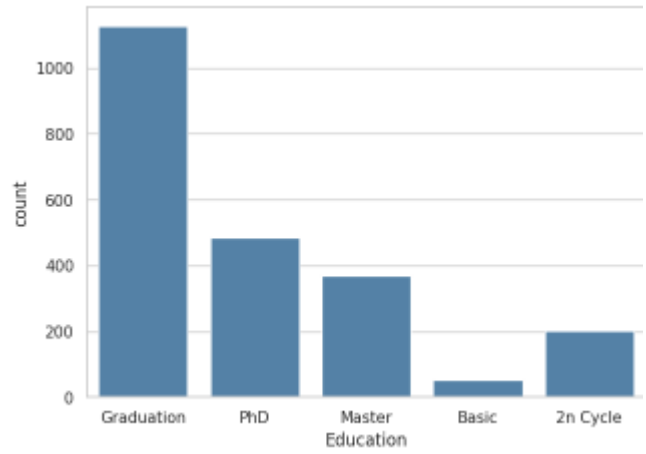
```
# Đọc file (dataset Kaggle dùng tab '\t' làm separator)
marketing_data = pd.read_csv("marketing_campaign.csv", sep="\t")

# Chọn một số cột tiêu biểu để phân tích
marketing_data = marketing_data[['Education','Income','Kidhome','Teenhome']]
marketing_data.head()
```

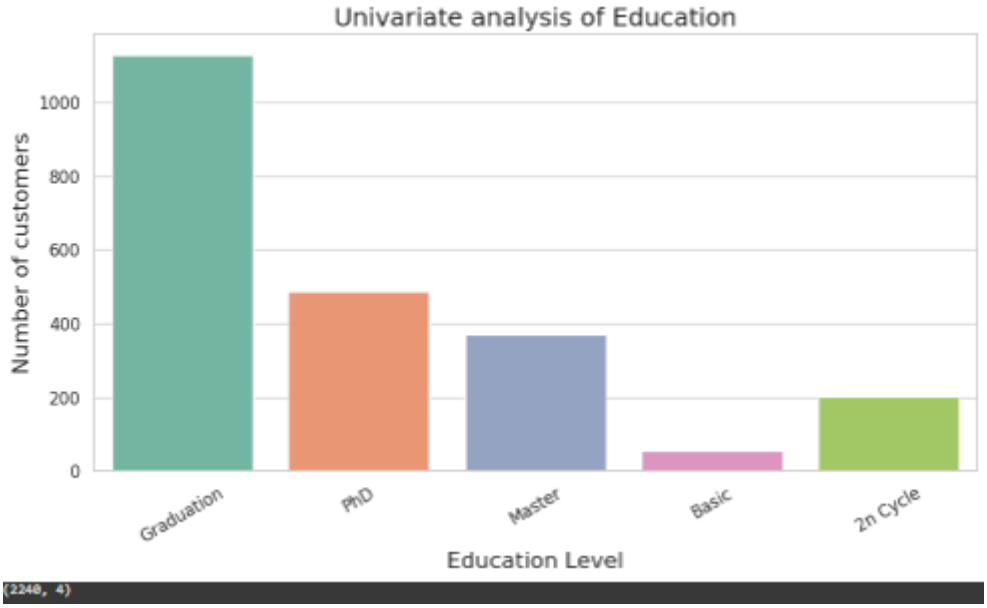
	Education	Income	Kidhome	Teenhome
0	Graduation	58138.0	0	0
1	Graduation	46344.0	1	1
2	Graduation	71613.0	0	0
3	Graduation	26646.0	1	0
4	PhD	58293.0	1	0

Phân tích đơn biến bằng Histogram (liên tục – Biến thu nhập)

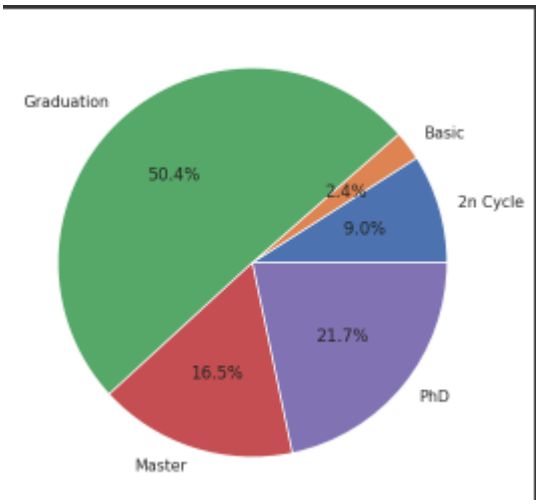


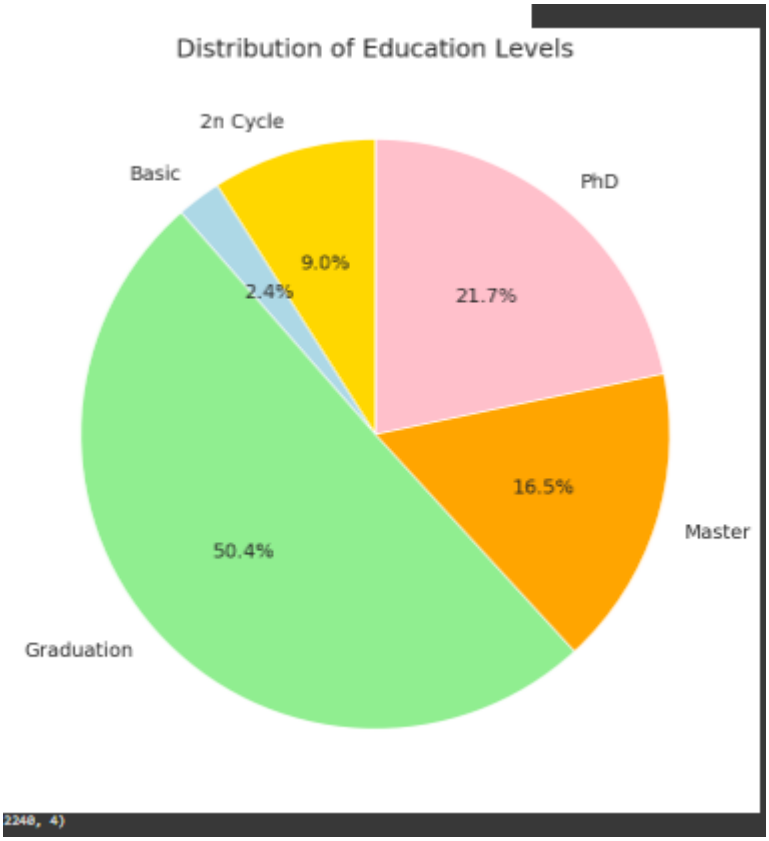
Phân tích đơn biến bằng Bar Chart (rời rạc – Biến education)





Phân tích đơn biến bằng Biểu đồ tròn (Pie Chart – Education)


Graduation
Basic
50.4%
2.4%
2n Cycle
9.0%
21.7%
16.5%
PhD
Master


Distribution of Education Levels
2n Cycle
Basic
PhD
9.0%
2.4%
21.7%
16.5%
Master
50.4%
Graduation

2240, 4)

[28] 0 giây

```
display(marketing_data.describe())
```

Kích thước dữ liệu: (2240, 5)

	Education	Income	Kidhome	Teenhome	Recency
0	Graduation	58138.0	0	0	58
1	Graduation	46344.0	1	1	38
2	Graduation	71613.0	0	0	26
3	Graduation	26646.0	1	0	26
4	PhD	58293.0	1	0	94

Số giá trị thiếu mỗi cột:

```
 Education     0
Income        24
Kidhome        0
Teenhome       0
Recency        0
dtype: int64
```

	Income	Kidhome	Teenhome	Recency
count	2216.000000	2240.000000	2240.000000	2240.000000
mean	52247.251354	0.444196	0.506250	49.109375
std	25173.076661	0.538398	0.544538	28.962453
min	1730.000000	0.000000	0.000000	0.000000
25%	35303.000000	0.000000	0.000000	24.000000
50%	51381.500000	0.000000	0.000000	49.000000
75%	68522.000000	1.000000	1.000000	74.000000

Customer Personality Analysis - Profiling Report

Overview Variables Interactions Correlations Missing values Sample

Overview

Brought to you by YData

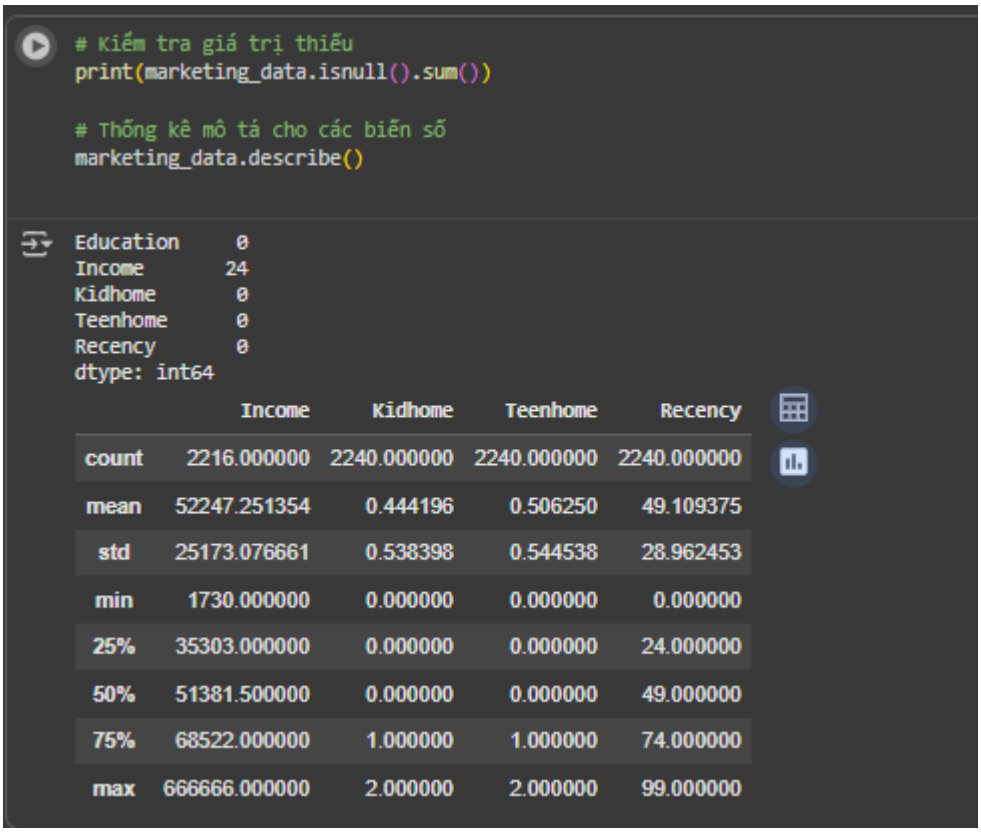
Overview Alerts 30 Reproduction

Dataset statistics

Number of variables	29
Number of observations	2240
Missing cells	24
Missing cells (%)	< 0.1%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	830.5 KiB

Variable types

Numeric	15
Categorical	13
DateTime	1



```
# Kiểm tra giá trị thiếu
print(marketing_data.isnull().sum())

# Thống kê mô tả cho các biến số
marketing_data.describe()
```

```
Education     0
Income       24
Kidhome       0
Teenhome      0
Recency       0
dtype: int64
```

	Income	Kidhome	Teenhome	Recency
count	2216.000000	2240.000000	2240.000000	2240.000000
mean	52247.251354	0.444196	0.506250	49.109375
std	25173.076661	0.538398	0.544538	28.962453
min	1730.000000	0.000000	0.000000	0.000000
25%	35303.000000	0.000000	0.000000	24.000000
50%	51381.500000	0.000000	0.000000	49.000000
75%	68522.000000	1.000000	1.000000	74.000000
max	666666.000000	2.000000	2.000000	99.000000

# Kiểm tra giá trị thiếu
print(marketing_data.isnull().sum())

# Thống kê mô tả cho các biến số
marketing_data.describe()

Tìm hiểu các tính năng và cách sử dụng sản phẩm SweetViz

1. Import thư viện và nạp dữ liệu

+ Mã + Văn bản

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Đọc file (dataset Kaggle dùng tab '\t' làm separator)
marketing_data = pd.read_csv("marketing_campaign.csv", sep="\t")

# Chọn một số cột tiêu biểu để phân tích
marketing_data = marketing_data[['Education','Income','Kidhome','Teenhome']]
marketing_data.head()
```

	Education	Income	Kidhome	Teenhome
0	Graduation	58138.0	0	0
1	Graduation	46344.0	1	1
2	Graduation	71613.0	0	0
3	Graduation	26646.0	1	0
4	PhD	58293.0	1	0

Các bước tiếp theo: Tạo mã bằng marketing_data | New interactive sheet

Nhiệm vụ 1: Tạo báo cáo tự động với pandas_profiling / ydata_profiling

Bước 1 – Cài đặt thư viện

[2] 51 giây

```
!pip install ydata-profiling
import pandas as pd
from ydata_profiling import ProfileReport

# Đọc dữ liệu Customer Personality Analysis
marketing_data = pd.read_csv("marketing_campaign.csv", sep="\t")

# Tạo báo cáo EDA tự động
profile = ProfileReport(marketing_data,
                        title="Customer Personality Analysis - Profiling
                        Report",
                        explorative=True)

# Xuất báo cáo ra file HTML
profile.to_file("profile_output.html")
```

Hiện kết quả đã ẩn

```
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->mat
Upgrade to ydata-sdk
Improve your data and profiling with ydata-sdk, featuring data quality scoring, redundancy detection, outlier identification, text validation, and s
Summarize dataset: 100% 263/263 [00:26<00:00,  6.14it/s, Completed]

  0%|          | 0/29 [00:00<?, ?it/s]
 17%|▌         | 5/29 [00:00<00:00, 34.22it/s]
100%|██████████| 29/29 [00:00<00:00, 76.56it/s]
Generate report structure: 100% 1/1 [00:06<00:00,  6.75s/it]
Render HTML: 100% 1/1 [00:01<00:00,  1.52s/it]
Export report to file: 100% 1/1 [00:00<00:00, 27.45it/s]
```

1.3.4. Bài tập thực hành

2. Tìm hiểu các tính năng và cách sử dung sản phẩm AutoViz (https://pypi.org/project/autoviz) áp dung trên tập dữ liệu Marketing Campaign

#Cài đặt & import

#Nạp dữ liệu

```
 #   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
```

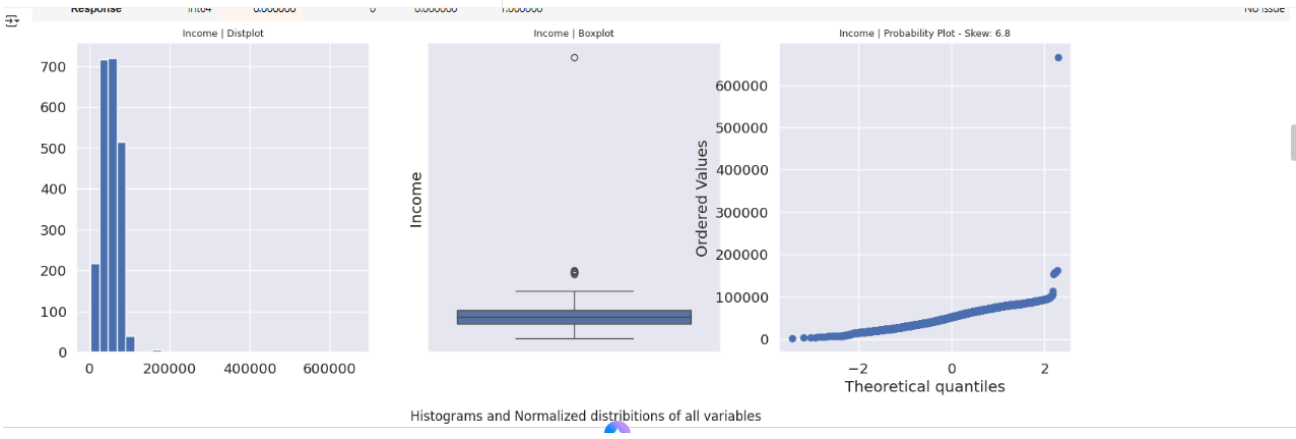
```
 26  Z_CostContact        2240 non-null   int64
 27  Z_Revenue            2240 non-null   int64
 28  Response             2240 non-null   int64
dtypes: float64(1), int64(25), object(3)
memory usage: 507.6+ KB
Shape of your Data Set loaded: (2240, 29)
################################################################################
###################### C L A S S I F Y I N G  V A R I A B L E S  ################
################################################################################
Classifying variables in data set...
    Number of Numeric Columns =  1
    Number of Integer-Categorical Columns =  15
    Number of String-Categorical Columns =  2
    Number of Factor-Categorical Columns =  0
    Number of String-Boolean Columns =  0
    Number of Numeric-Boolean Columns =  7
    Number of Discrete String Columns =  1
    Number of NLP String Columns =  0
    Number of Date Time Columns =  0
    Number of ID Columns =  1
    Number of Columns to Delete =  2
    29 Predictors classified...
        3 variable(s) removed since they were ID or low-information variables
        List of variables removed: ['ID', 'Z_CostContact', 'Z_Revenue']
To fix these data quality issues in the dataset, import FixDQ from autoviz...
    All variables classified into correct types.
                         Data       Missing       Unique       Minimum       Maximum
                         Type       Values%      Values%         Value         Value
```

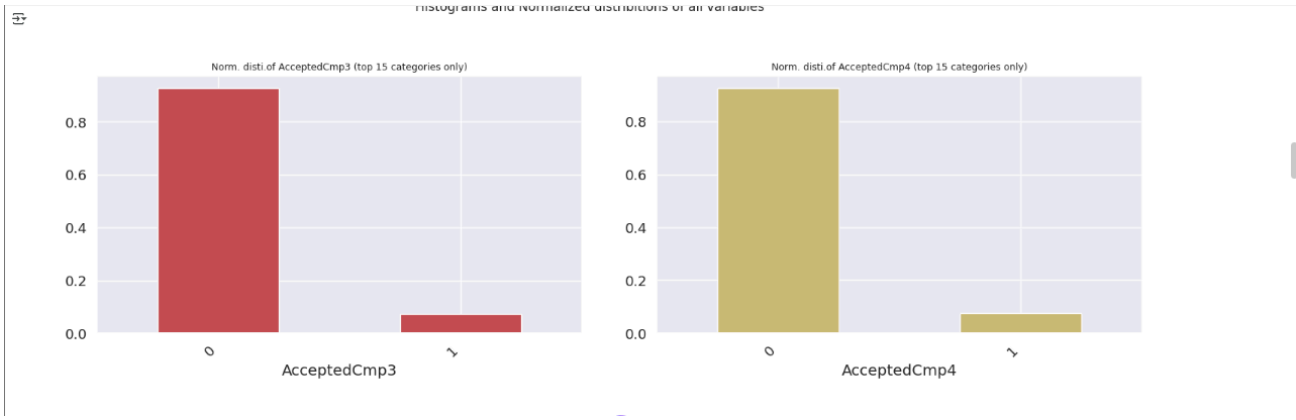
```
        3 variable(s) removed since they were ID or low-information variables
        List of variables removed: ['ID', 'Z_CostContact', 'Z_Revenue']
To fix these data quality issues in the dataset, import FixDQ from autoviz...
    All variables classified into correct types.
```

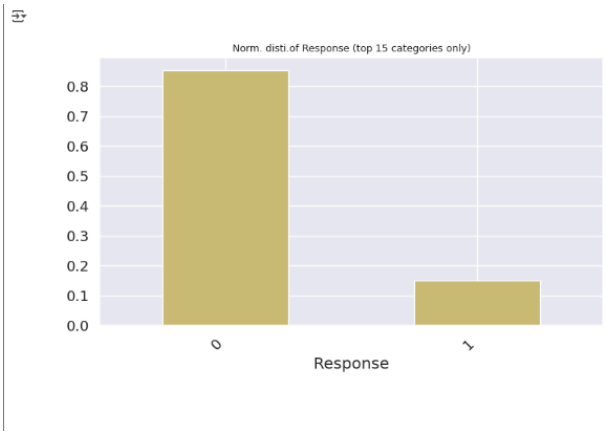
	Data Type	Missing Values%	Unique Values%	Minimum Value	Maximum Value	DQ Issue
ID	int64	0.000000	100	0.000000	11191.000000	Possible ID column: drop before modeling step.
Year_Birth	int64	0.000000	2	1893.000000	1996.000000	Column has 3 outliers greater than upper bound (2004.00) or lower than lower bound(1932.00). Cap them or remove them.
Education	object	0.000000	0			No issue
Marital_Status	object	0.000000	0			3 rare categories: ['Alone', 'Absurd', 'YOLO']. Group them into a single category or drop the categories.
Income	float64	1.071429	NA	1730.000000	666666.000000	24 missing values. Impute them with mean, median, mode, or a constant value such as 123., Column has 8 outliers greater than upper bound (118350.50) or lower than lower bound(-14525.50). Cap them or remove them.
Kidhome	int64	0.000000	0	0.000000	2.000000	No issue
Teenhome	int64	0.000000	0	0.000000	2.000000	No issue
Dt_Customer	object	0.000000	29			Possible high cardinality column with 663 unique values: Use hash encoding or text embedding to reduce dimension.
Recency	int64	0.000000	4	0.000000	99.000000	No issue
MntWines	int64	0.000000	34	0.000000	1493.000000	Column has 35 outliers greater than upper bound (1225.00) or lower than lower bound(-697.00). Cap them or remove them.
MntFruits	int64	0.000000	7	0.000000	199.000000	Column has 227 outliers greater than upper bound (81.00) or lower than lower bound(-47.00). Cap them or remove them.

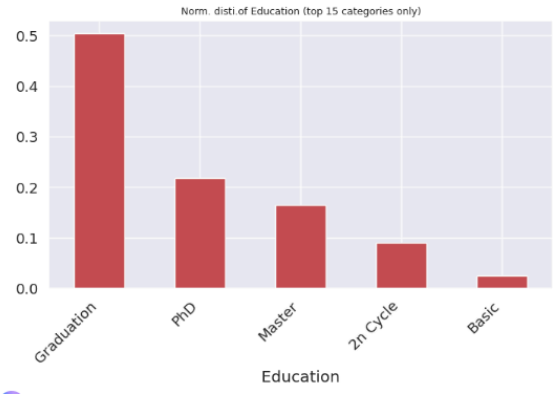
Cửa sổ dòng lệnh

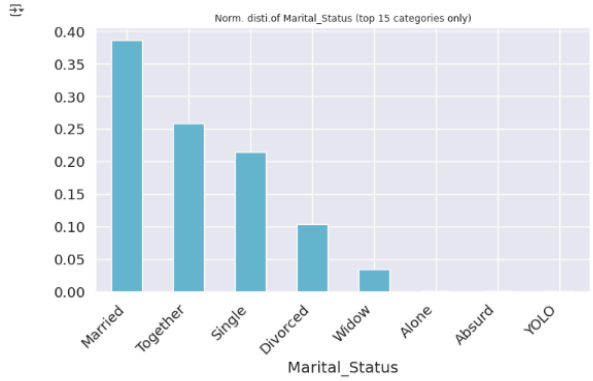
#Sinh báo cáo tự động

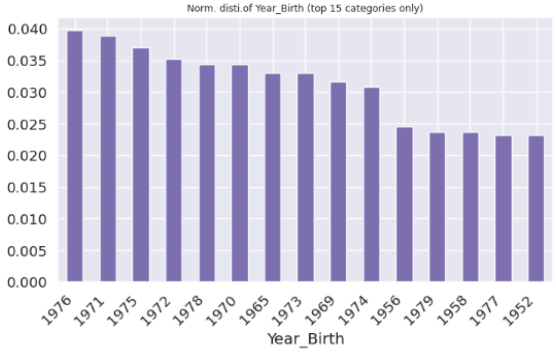


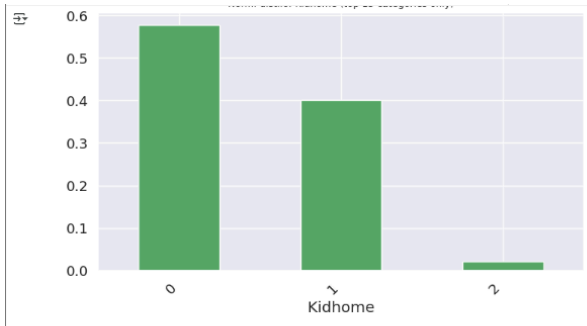


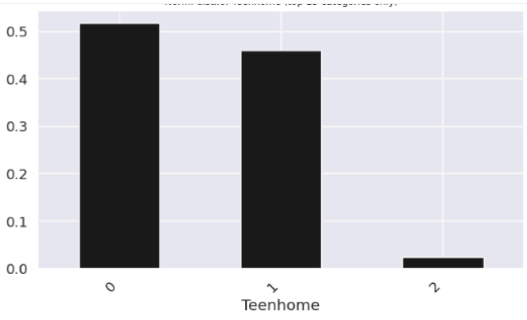

Norm. disti.of Response (top 15 categories only)
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
Response

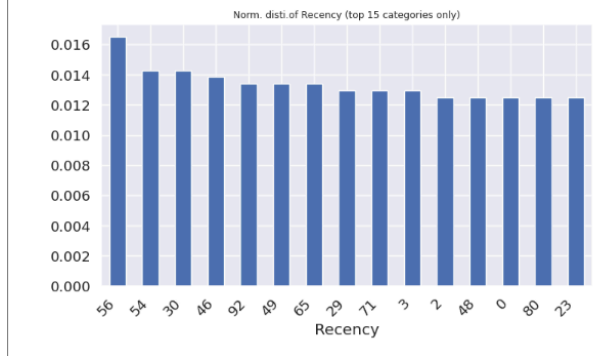

Norm. disti.of Education (top 15 categories only)
0.5
0.4
0.3
0.2
0.1
0.0
Graduation
PhD
Master
2n Cycle
Basic
Education

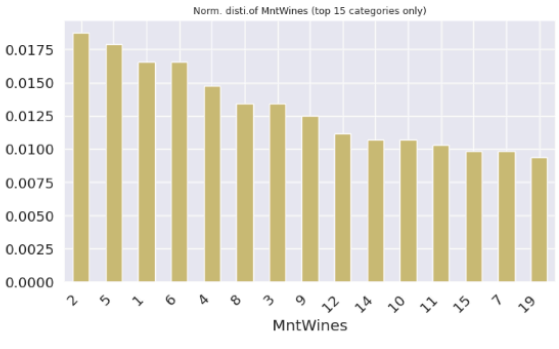

Norm. disti.of Marital_Status (top 15 categories only)
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Married
Together
Single
Divorced
Widow
Alone
Absurd
YOLO
Marital_Status

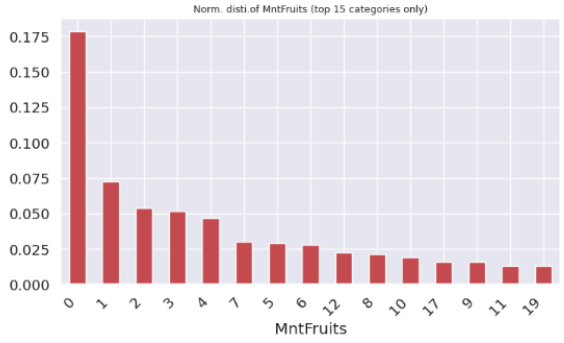

Norm. disti.of Year_Birth (top 15 categories only)
0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
1976
1971
1975
1972
1978
1970
1965
1973
1969
1974
1956
1979
1958
1977
1952
Year_Birth

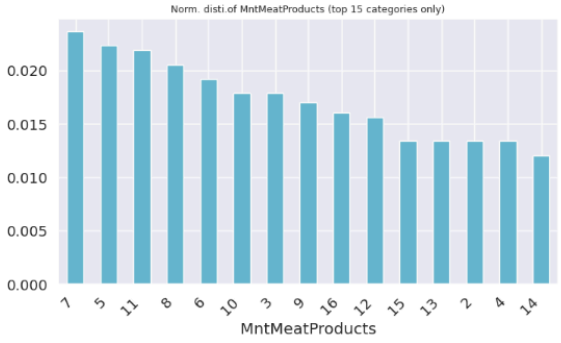

0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
Kidhome

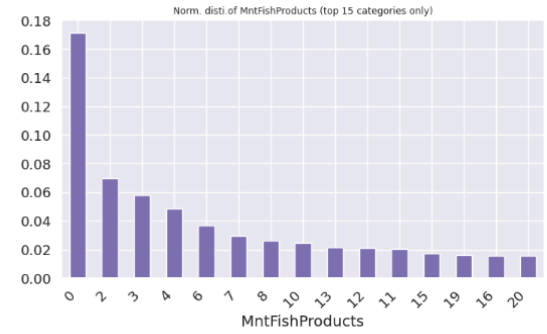

0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
Teenhome

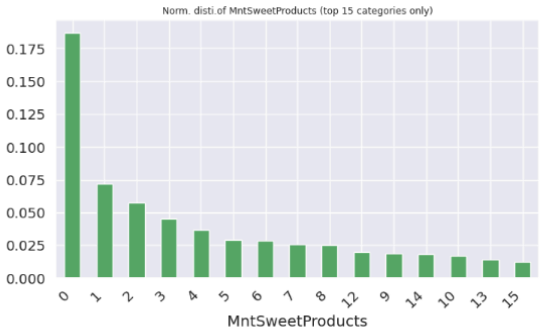

Norm. disti.of Recency (top 15 categories only)
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
56
54
30
46
92
49
65
29
71
3
2
48
0
80
23
Recency

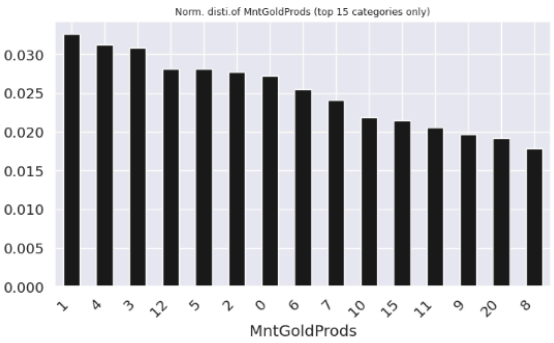

Norm. disti.of MntWines (top 15 categories only)
0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
2
5
1
6
4
8
3
9
12
14
10
11
15
7
19
MntWines

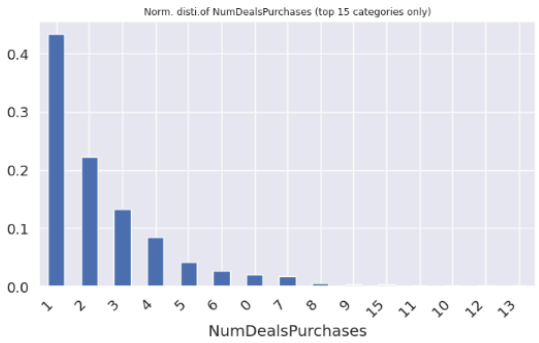

Norm. disti.of MntFruits (top 15 categories only)
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0 1 2 3 4 7 5 6 12 8 10 17 9 11 19
MntFruits

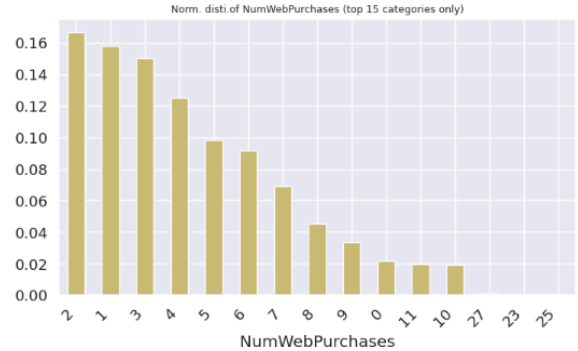

Norm. disti.of MntMeatProducts (top 15 categories only)
0.020
0.015
0.010
0.005
0.000
7 5 11 8 6 10 3 9 16 12 15 13 2 4 14
MntMeatProducts

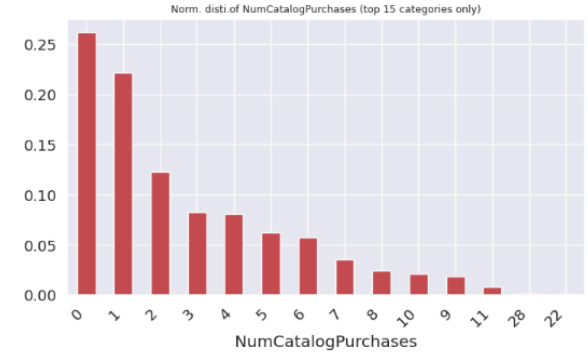

Norm. disti.of MntFishProducts (top 15 categories only)
0.18
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0 2 3 4 6 7 8 10 13 12 11 15 19 16 20
MntFishProducts

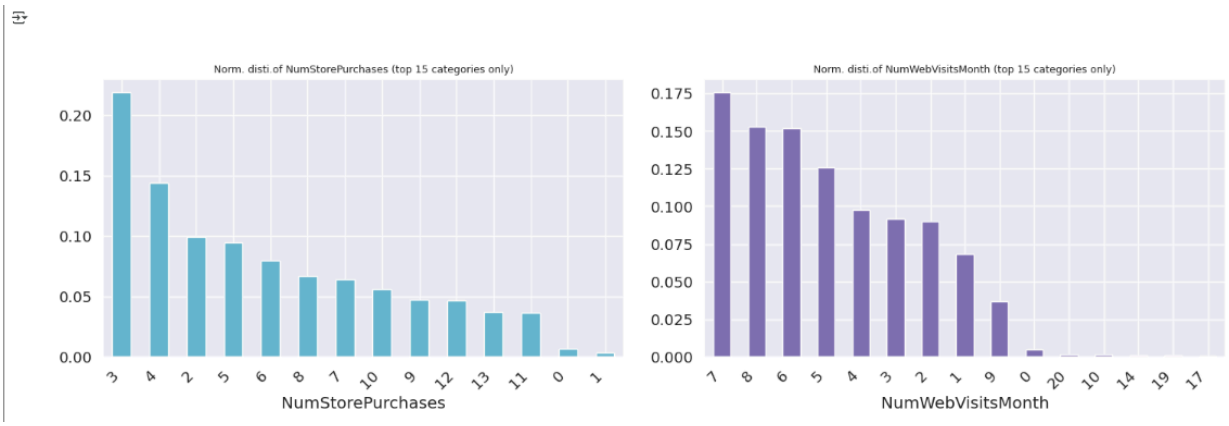

Norm. disti.of MntSweetProducts (top 15 categories only)
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0 1 2 3 4 5 6 7 8 12 9 14 10 13 15
MntSweetProducts


Norm. disti.of MntGoldProds (top 15 categories only)
0.030
0.025
0.020
0.015
0.010
0.005
0.000
1 4 3 12 5 2 0 6 7 10 15 11 9 20 8
MntGoldProds

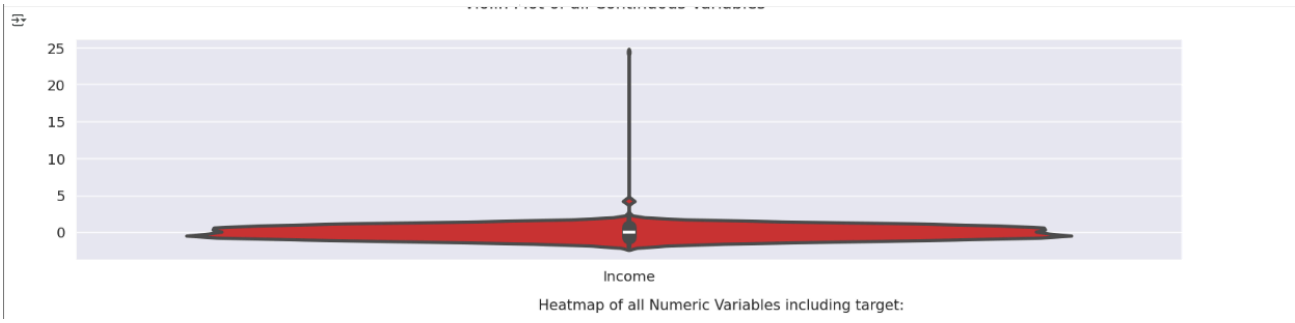

Norm. disti.of NumDealsPurchases (top 15 categories only)
0.4
0.3
0.2
0.1
0.0
1 2 3 4 5 6 0 7 8 9 15 11 10 12 13
NumDealsPurchases


Norm. disti.of NumWebPurchases (top 15 categories only)
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
2 1 3 4 5 6 7 8 9 0 11 10 27 23 25
NumWebPurchases

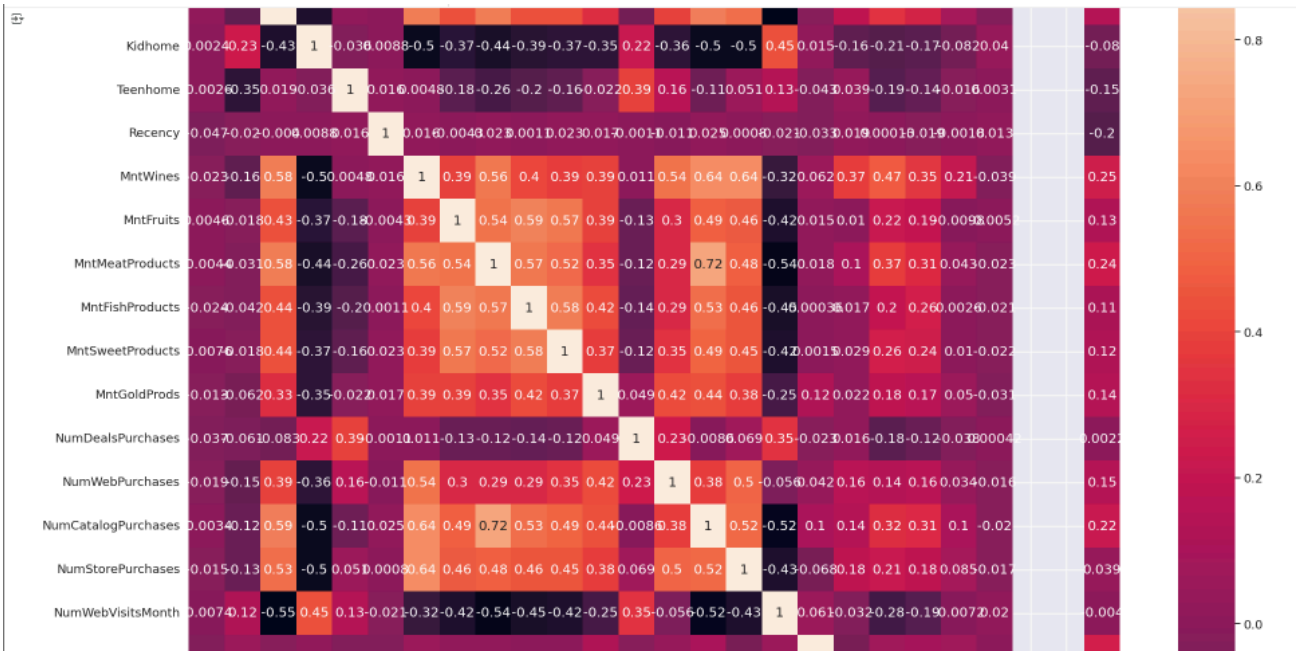

Norm. disti.of NumCatalogPurchases (top 15 categories only)
0.25
0.20
0.15
0.10
0.05
0.00
0 1 2 3 4 5 6 7 8 10 9 11 28 22
NumCatalogPurchases

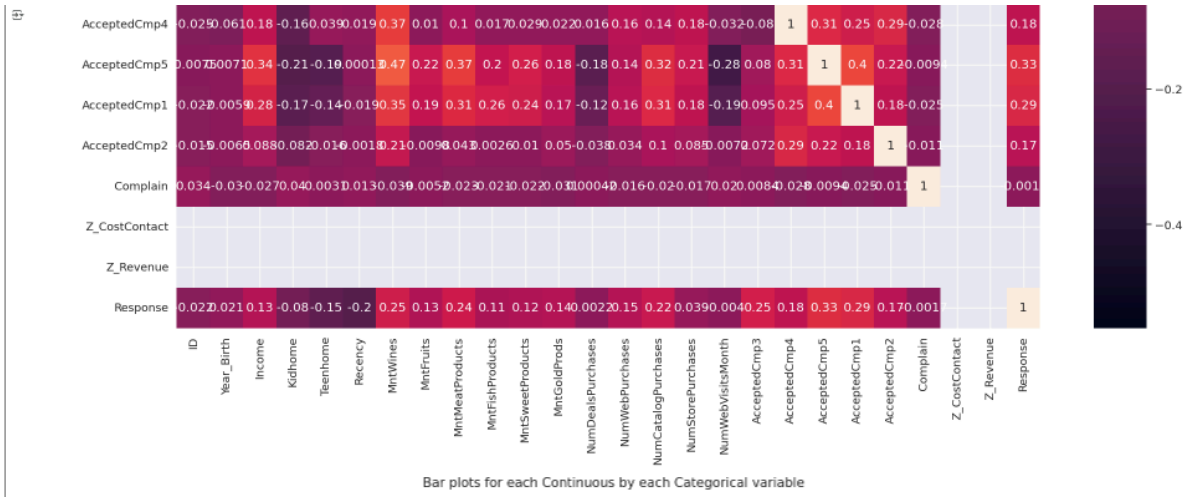


#Lưu thêm biểu đồ quan trọng



#Đồ thị heatmap





Bar plots for each Continuous by each Categorical variable