



Data Glacier

Your Deep Learning Partner

Bank Customer Segmentation

Name: Madoka Fujii

Date: Jan 15, 2025

Agenda

Problem Description

Data Understanding

Data Cleaning

EDA for Business

Recommendations for Technical

Batch:LISUM39

Week11: EDA Presentation and proposed modeling technique

Project: Bank Customer Segmentation

Group name: Apple Analytics

Name: Madoka Fujii

Email address: mdkfji@gmail.com

Country: United States

Company: Omdena

Specialization: Data Analytics

Problem Description

XYZ Bank plans to enhance its marketing campaign as Christmas offers for its customers. However, instead of offering the same deal to all customers as generic, the bank wants to provide personalized offers to specific customer groups to fit their preferences. Identifying customer categories manually would be inefficient and fail to uncover hidden patterns in the data that could inform better segmentation. To address this, the bank has sought the assistance of Apple Analytics.

Additionally, the bank has specified that customer segmentation should result in no more than 5 groups to ensure the campaign's efficiency.

Data Understanding

Tabular data details: cust_seg.csv.zip

Total number of observations	1,000,000
Total number of files	1
Total number of features	48
Base format of the file	csv.zip
Size of the data	19,483KB

Data Understanding

There are missing values.

Check the unique values and frequencies.

	count	unique	top	freq
fecha_datos	1000000	2	2015-01-28	625457
ind_empleado	989218	5	N	988260
pais_residencia	989218	113	ES	982264
sexo	989214	2	V	562000
age	1000000	115	22	51017
fecha_alta	989218	6238	2013-10-14	3920
antiguedad	1000000	249	21	34320
ult_fec_cli_1t	1101	22	2015-07-01	97
tiprel_1mes	989218	3	A	547800
indresi	989218	2	S	982264
indext	989218	2	N	946328
conyuemp	178	2	N	176
canal_entrada	989139	156	KAT	313944
indfall	989218	2	N	986107
nomprov	982266	52	MADRID	360131

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 1000000 entries, 0 to 999999
```

```
Data columns (total 48 columns):
```

#	Column	Non-Null Count	Dtype
0	Unnamed: 0	1000000 non-null	int64
1	fecha_datos	1000000 non-null	object
2	ncodpers	1000000 non-null	int64
3	ind_empleado	989218 non-null	object
4	pais_residencia	989218 non-null	object
5	sexo	989214 non-null	object
6	age	1000000 non-null	object
7	fecha_alta	989218 non-null	object
8	ind_nuevo	989218 non-null	float64
9	antiguedad	1000000 non-null	object
10	indrel	989218 non-null	float64
11	ult_fec_cli_1t	1101 non-null	object
12	indrel_1mes	989218 non-null	float64
13	tiprel_1mes	989218 non-null	object
14	indresi	989218 non-null	object
15	indext	989218 non-null	object
16	conyuemp	178 non-null	object
17	canal_entrada	989139 non-null	object
18	indfall	989218 non-null	object
19	tipodom	989218 non-null	float64
20	cod_prov	982266 non-null	float64
21	nomprov	982266 non-null	object
22	ind_actividad_cliente	989218 non-null	float64
23	renta	824817 non-null	float64
24	ind_ahor_fin_ult1	1000000 non-null	int64
25	ind_aval_fin_ult1	1000000 non-null	int64
26	ind_cco_fin_ult1	1000000 non-null	int64
27	ind_cder_fin_ult1	1000000 non-null	int64
28	ind_cno_fin_ult1	1000000 non-null	int64
29	ind_ctju_fin_ult1	1000000 non-null	int64
30	ind_ctma_fin_ult1	1000000 non-null	int64
31	ind_ctop_fin_ult1	1000000 non-null	int64
32	ind_ctpp_fin_ult1	1000000 non-null	int64
33	ind_deco_fin_ult1	1000000 non-null	int64
34	ind_deme_fin_ult1	1000000 non-null	int64
35	ind_dela_fin_ult1	1000000 non-null	int64
36	ind_ecue_fin_ult1	1000000 non-null	int64
37	ind_fond_fin_ult1	1000000 non-null	int64
38	ind_hip_fin_ult1	1000000 non-null	int64

Data Understanding

Check the unique values each variables.

```
Unique values in fecha_dato are :
fecha_dato
2015-01-28    625457
2015-02-28    374543
Name: count, dtype: int64
*****

Unique values in ind_empleado are :
ind_empleado
N    988260
B      387
A      287
F      282
S        2
Name: count, dtype: int64
*****

Unique values in pais_residencia are :
pais_residencia
ES    982264
FR      546
AR      542
DE      487
GB      480
...
MM        2
ML        2
LV        2
BZ        2
AL        1
Name: count, Length: 113, dtype: int64
*****

Unique values in sexo are :
sexo
V    562000
H    427214
Name: count, dtype: int64
*****

Unique values in age are :
age
22    51017
23    45366
24    38992
21    34015
44    28800
...
110     14
115     12
...
```

```
Name: count, Length: 115, dtype: int64
*****

Unique values in fecha_alta are :
fecha_alta
2013-10-14    3920
2013-08-03    3738
2014-07-28    3285
2014-10-03    2861
2013-10-11    2686
...
2015-02-17        1
2010-11-20        1
2010-11-21        1
2012-05-12        1
2011-02-05        1
Name: count, Length: 6238, dtype: int64
*****

Unique values in antiguedad are :
antiguedad
21    34320
23    23122
24    20467
12    19155
20    18582
...
0        49
2        46
1        37
3        33
-999999    4
Name: count, Length: 249, dtype: int64
*****

Unique values in ult_fec_cli_it are :
ult_fec_cli_it
2015-07-01    97
2015-07-09    81
2015-07-06    76
2015-07-21    67
2015-07-07    63
2015-07-17    59
2015-07-28    57
2015-07-10    55
2015-07-15    54
2015-07-24    54
2015-07-20    53
2015-07-22    45
2015-07-03    42
```

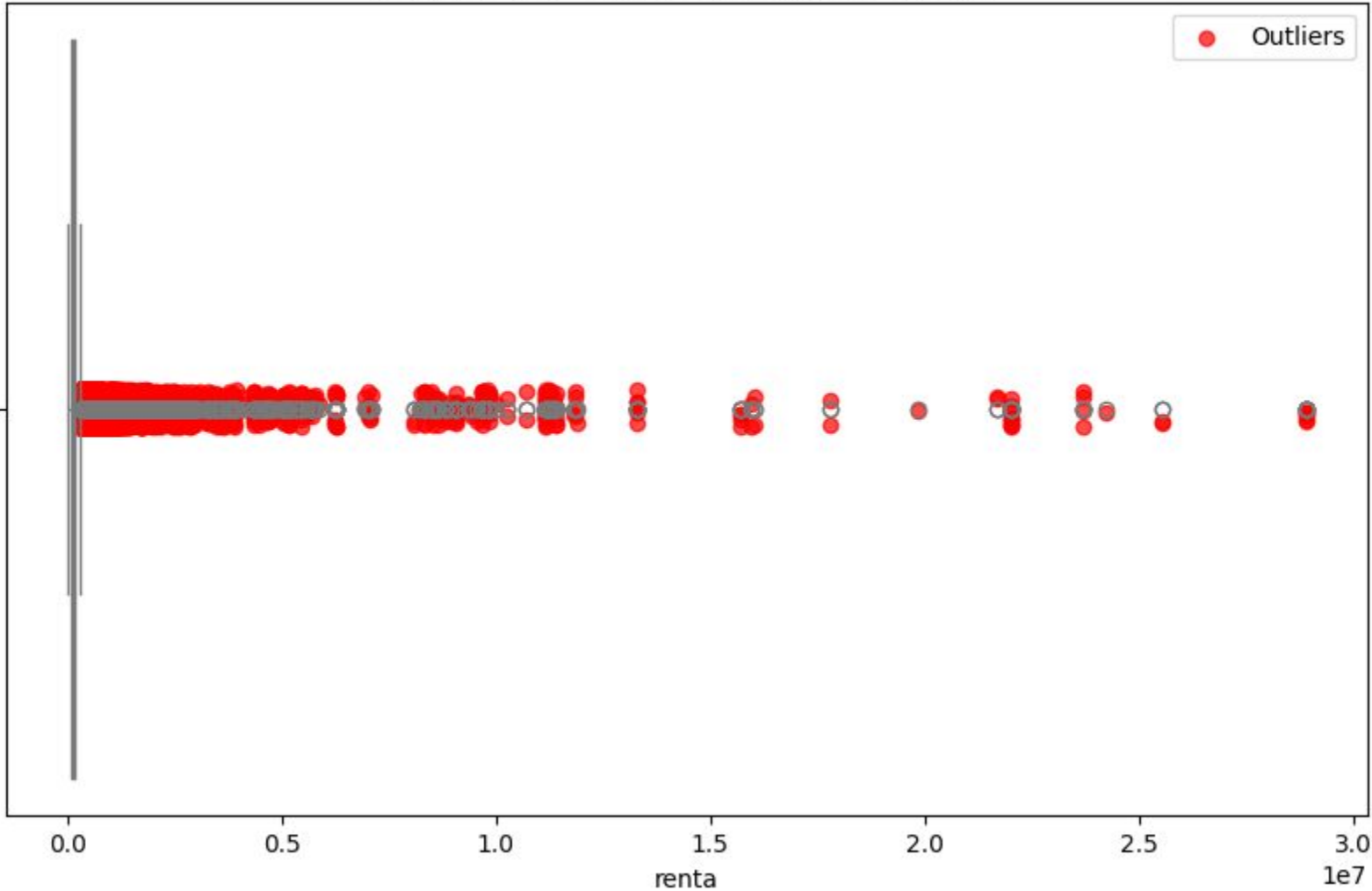

Data Understanding

Check how many percent each variable has the missing values.

	Column_Name	aggregate	percent
0	conyuemp	999822	0.999822
1	ult_fec_cli_1t	998899	0.998899
2	renta	175183	0.175183
3	cod_prov	17734	0.017734
4	nomprov	17734	0.017734
5	canal_entrada	10861	0.010861
6	sexo	10786	0.010786
7	pais_residencia	10782	0.010782
8	indresi	10782	0.010782
9	tiprel_1mes	10782	0.010782
10	indext	10782	0.010782
11	ind_empleado	10782	0.010782
12	indrel_1mes	10782	0.010782
13	indrel	10782	0.010782
14	ind_nuevo	10782	0.010782
15	fecha_alta	10782	0.010782
16	tipodom	10782	0.010782
17	indfall	10782	0.010782
18	ind_actividad_cliente	10782	0.010782
19	ind_nom_pens_ult1	5402	0.005402
20	ind_nomina_ult1	5402	0.005402
21	ncodpers	0	0.000000
22	Unnamed: 0	0	0.000000
23	age	0	0.000000

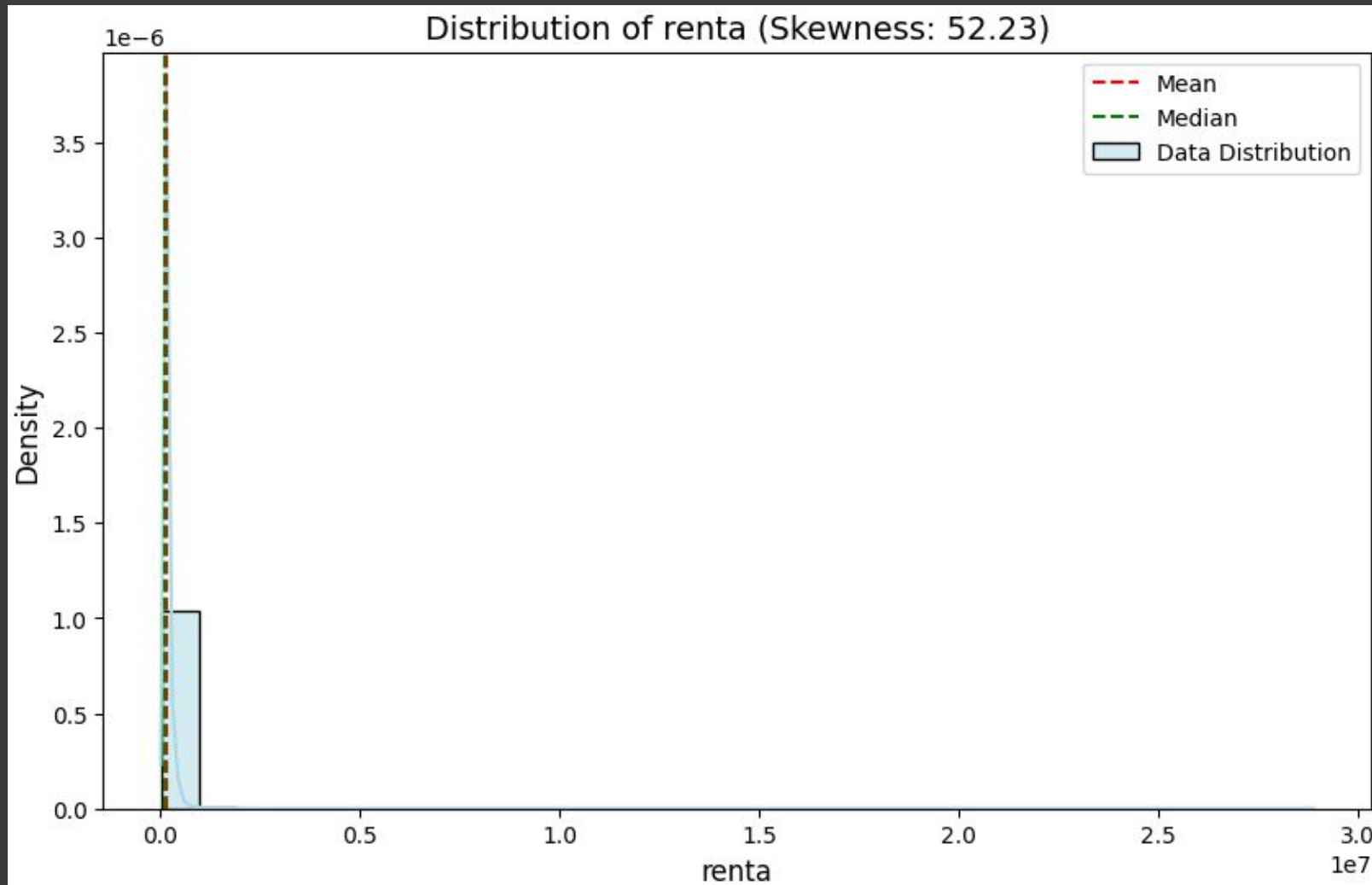
Data Understanding

Box Plot with Outliers Highlighted for renta



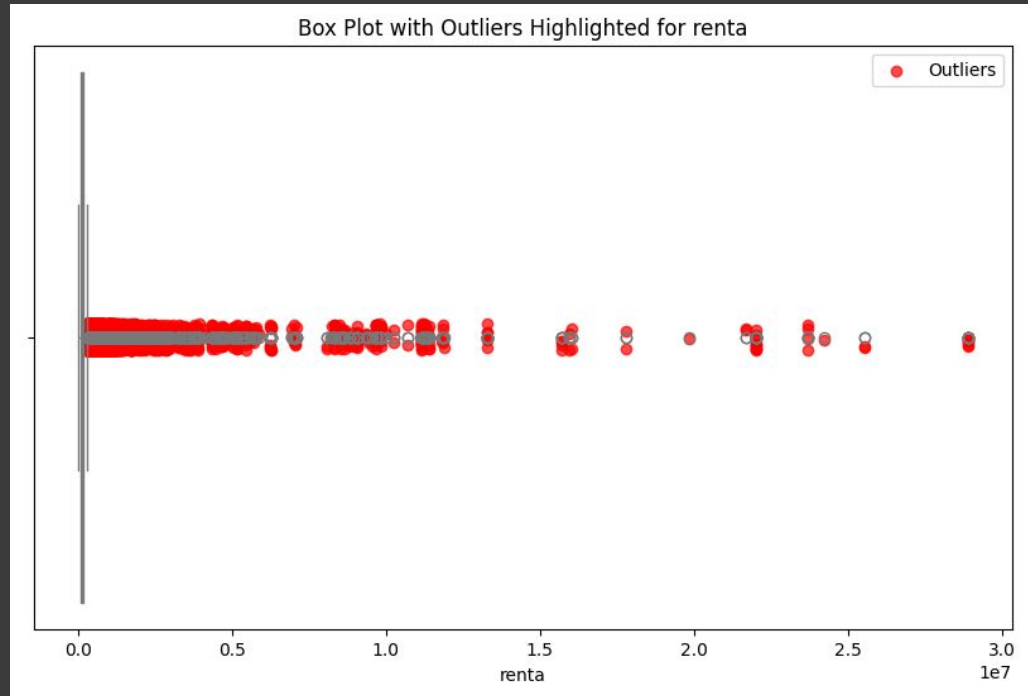
The variable 'renta' contains numerous outliers, with many of them representing individuals with high gross incomes.

Data Understanding



In the variable 'renta', Median and Mean is almost same point while there are many outliers.

Data Cleaning



Examined the outliers and skewness in the visualizations in the previous slide. It appears that the 'renta' variable, which represents the gross income of households, contains many outliers above the median and mean. These outliers correspond to high-income customers. Retaining these outliers may be beneficial for identifying specific trends within the high-income group, which could be useful for targeted campaigns. Therefore, while I am checking for outliers, I do not plan to omit them.

Data Cleaning

Impute Numerical missing values

1, fillna() median

Regarding **Renta**, the median and mean are almost same points and extremely skewed to right so applying **imputation with median**.

```
df['renta'] = df['renta'].fillna(df['renta'].median())
```

2, KNN Imputer method

Apply **KNN Imputer** to numerical values

Nearest Neighbor Imputation is a powerful technique that relies on the similarity between data points. It can provide more accurate imputations than simpler methods like mean or median imputation, especially when relationships between features are complex.

```
from sklearn.impute import KNNImputer

# Initialize KNNImputer with k=2 neighbors
imputer = KNNImputer(n_neighbors=2)

# Impute missing values
numeric_df = df.select_dtypes(include=[float, int]) #KNN impute only can apply for numerical values
imputed_data = imputer.fit_transform(numeric_df)

print(imputed_data)
```

```
[[0.000000e+00 1.375586e+06 0.000000e+00 ... 0.000000e+00 0.000000e+00
 0.000000e+00]
 [1.000000e+00 1.050611e+06 0.000000e+00 ... 0.000000e+00 0.000000e+00
```

Regarding 'renta', applied the method fillna() median.

Regarding other numerical missing values, applied KNN Imputation.

Data Cleaning

3, Impute Categorical missing values with fillna() mode

```
cols = df.select_dtypes(['object']).columns.tolist()
print(cols)
```

```
['fecha_datos', 'ind_employment', 'pais_residencia', 'sexo', 'age', 'fecha_alta', 'antiguedad', 'tiprel_1mes', 'indresi', 'index', 'canal_entrada', 'indfall', 'nomprov']
```

```
for i in cols:
    df[i] = df[i].astype('category')
```

```
cat_cols = df.select_dtypes(['category']).columns.tolist()
print(cat_cols)
```

```
['fecha_datos', 'ind_employment', 'pais_residencia', 'sexo', 'age', 'fecha_alta', 'antiguedad', 'tiprel_1mes', 'indresi', 'index', 'canal_entrada', 'indfall', 'nomprov']
```

```
for column in cat_cols:
    mode = df[column].mode()[0]
    df[column] = df[column].fillna(value=mode)
```

Regarding Categorical missing values, applied fillna() mode method.

Data Cleaning

Rename the column names to reader friendly

Drop the column with unnamed because it used be index

```
df.drop(columns=["Unnamed: 0"], inplace = True, axis = 1)
```

```
df.rename({'ancondpers': 'Customer_code', 'ind_empleado': 'Employee_index: A_active_B_ex_employed_F_filial_N_not_employee_P'}, inplace=True)
```

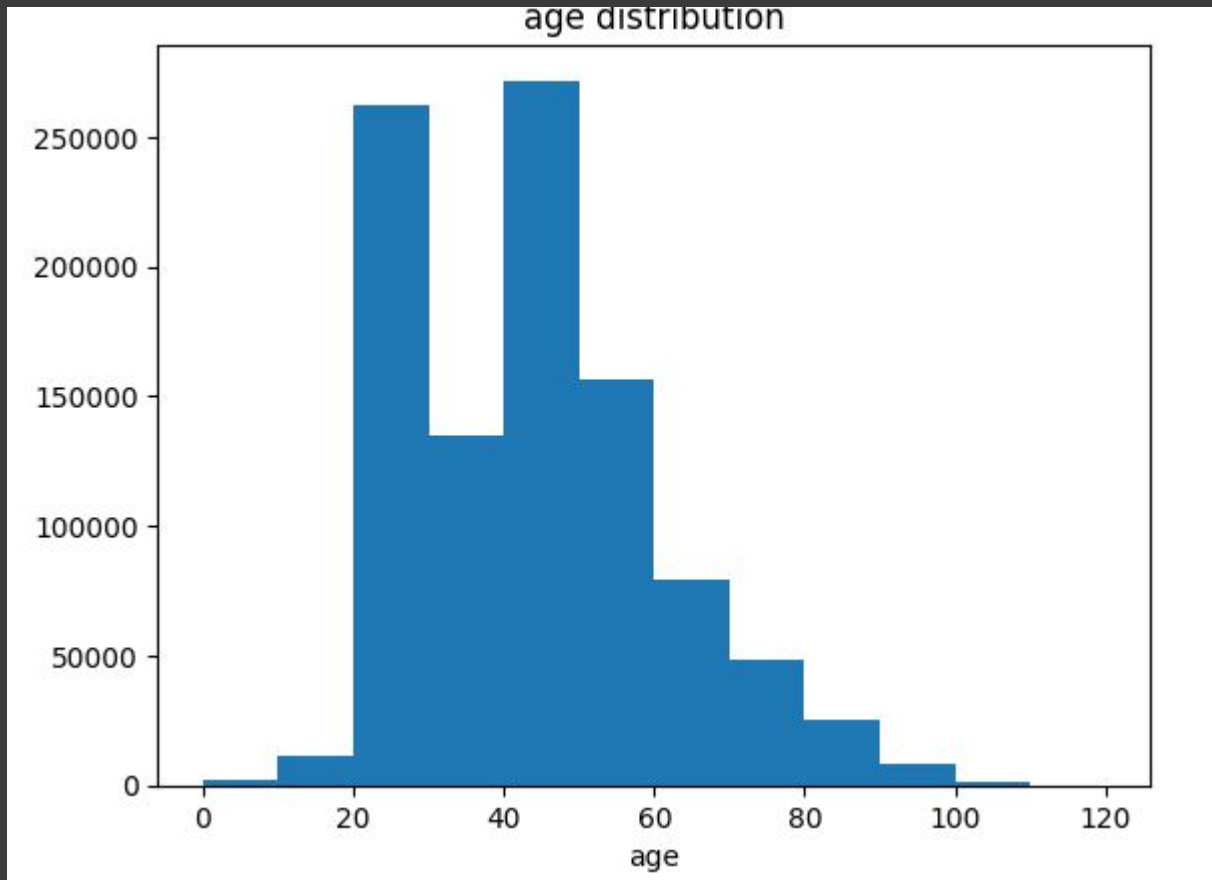
For readability in English,
renamed the column name if it is
hard to understand.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000000 entries, 0 to 999999
Data columns (total 45 columns):
 #   Column                                     Non-Null Count  Dtype
---  -
 0   fecha_dato                               1000000 non-null object
 1   customer_id                             1000000 non-null int64
 2   Employee_or_not                         1000000 non-null object
 3   Customer_s_Country_residence            1000000 non-null object
 4   sexo                                     1000000 non-null object
 5   age                                      1000000 non-null int64
 6   fecha_alta                              1000000 non-null object
 7   New_customer_Index                     1000000 non-null int64
 8   Customer_seniority                     1000000 non-null int64
 9   indrel                                  1000000 non-null int64
10   Customer_type_at_the_beginning_of_the_month 1000000 non-null int64
11   Customer_relation_type_at_the_beginning_of_the_month 1000000 non-null object
12   Residence_index                         1000000 non-null object
13   Foreigner_index                        1000000 non-null object
14   channel_used_by_the_customer_to_join    1000000 non-null object
15   Deceased_index                         1000000 non-null object
16   Address_type                           1000000 non-null int64
17   Province_code                          1000000 non-null int64
18   Province_name                          1000000 non-null object
19   Activity_index                         1000000 non-null int64
20   Gross_income_of_the_household          1000000 non-null float64
21   Saving_Account                         1000000 non-null int64
22   Guarantees                             1000000 non-null int64
23   Current_Accounts                       1000000 non-null int64
24   Derivada_Account                       1000000 non-null int64
25   Payroll_Account                        1000000 non-null int64
26   Junior_Account                         1000000 non-null int64
27   Más_particular_Account                  1000000 non-null int64
28   particular_Account                     1000000 non-null int64
29   particular_Plus_Account                 1000000 non-null int64
30   Short-term_deposits                     1000000 non-null int64
31   Medium-term_deposits                    1000000 non-null int64
32   Long-term_deposits                     1000000 non-null int64
33   e-account                              1000000 non-null int64
34   Funds                                  1000000 non-null int64
35   Mortgage                               1000000 non-null int64
36   Pensions                               1000000 non-null int64
37   Loans                                  1000000 non-null int64
38   Taxes                                  1000000 non-null int64
39   Credit_Card                            1000000 non-null int64
40   Securities                             1000000 non-null int64
```

Make sure the data type each variable and the number of data each variable.

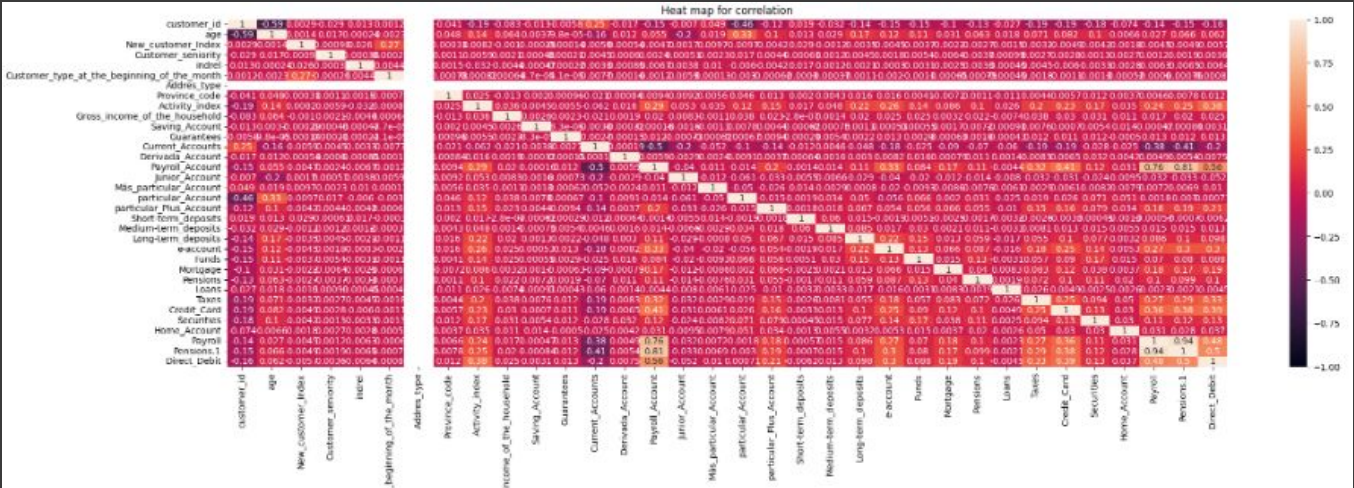
EDA

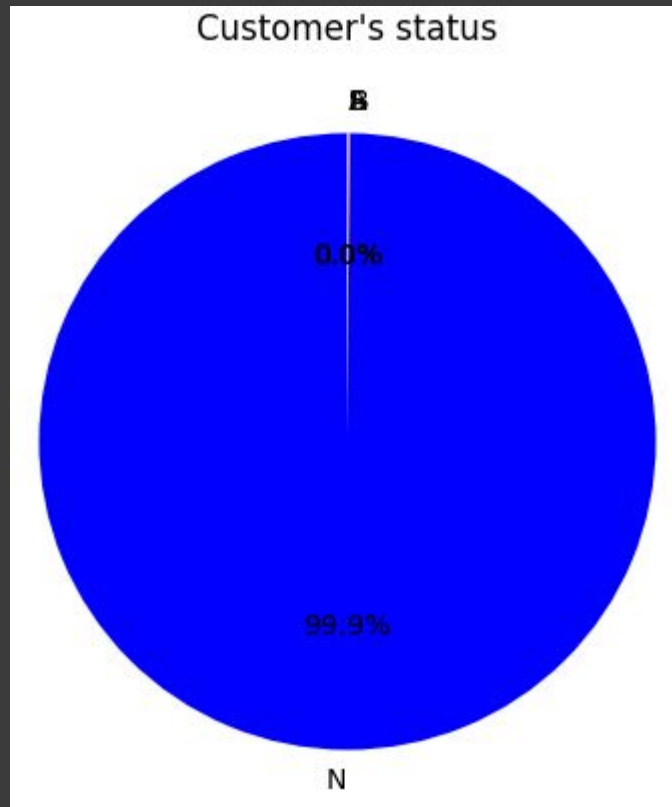


In the plot of Age Distribution, the majority of age group is from 20 years old to 60 years old.

EDA

There is no significantly related correlation.





Employee index:

A active

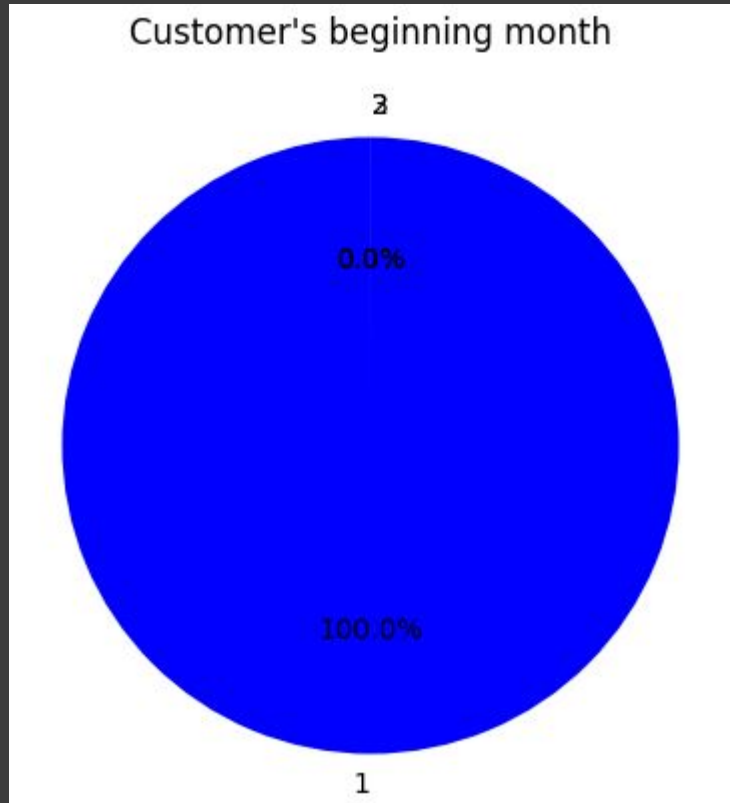
B ex employed

F filial

N not employee

P passive

Most of the customers are N as not employees.



Customer type at the beginning of the month

1 (First/Primary customer)

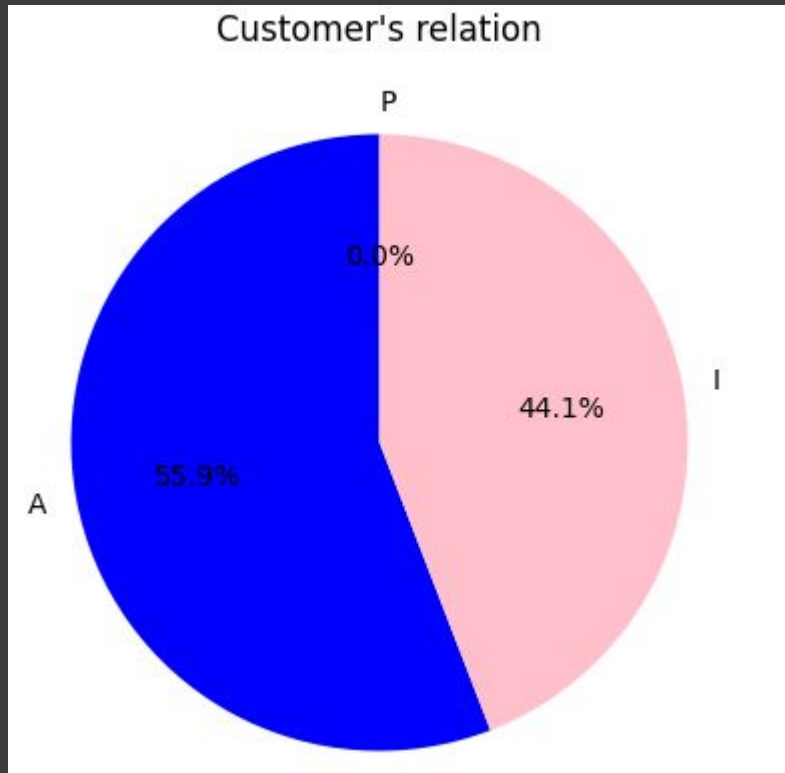
2 (co-owner)

P (Potential)

3 (former primary)

4(former co-owner)

Most of customers are 1 (First/Primary customer)



Customer relation type at the beginning of the month

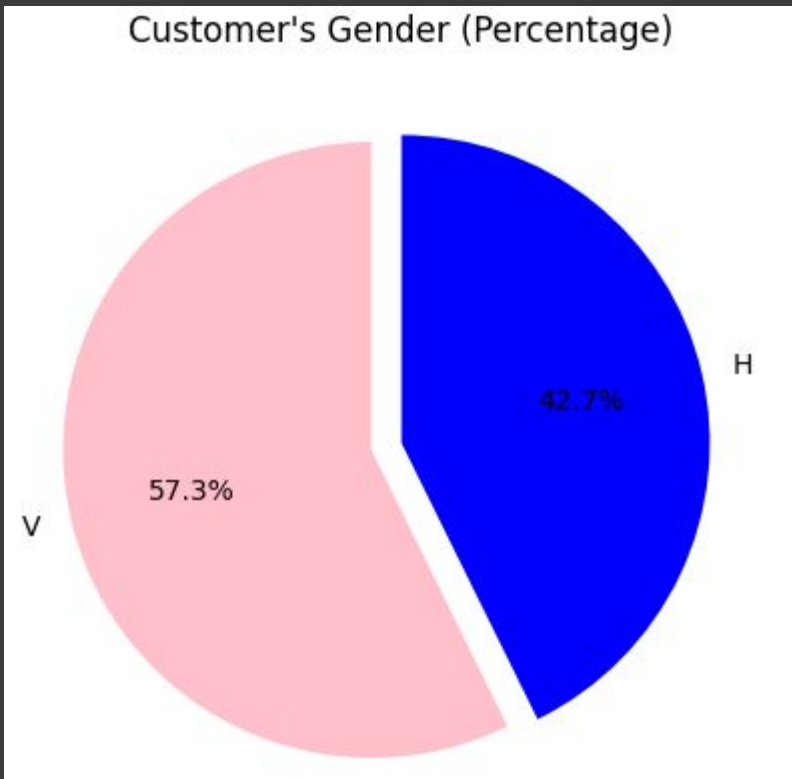
A (active)

I (inactive)

P (former customer)

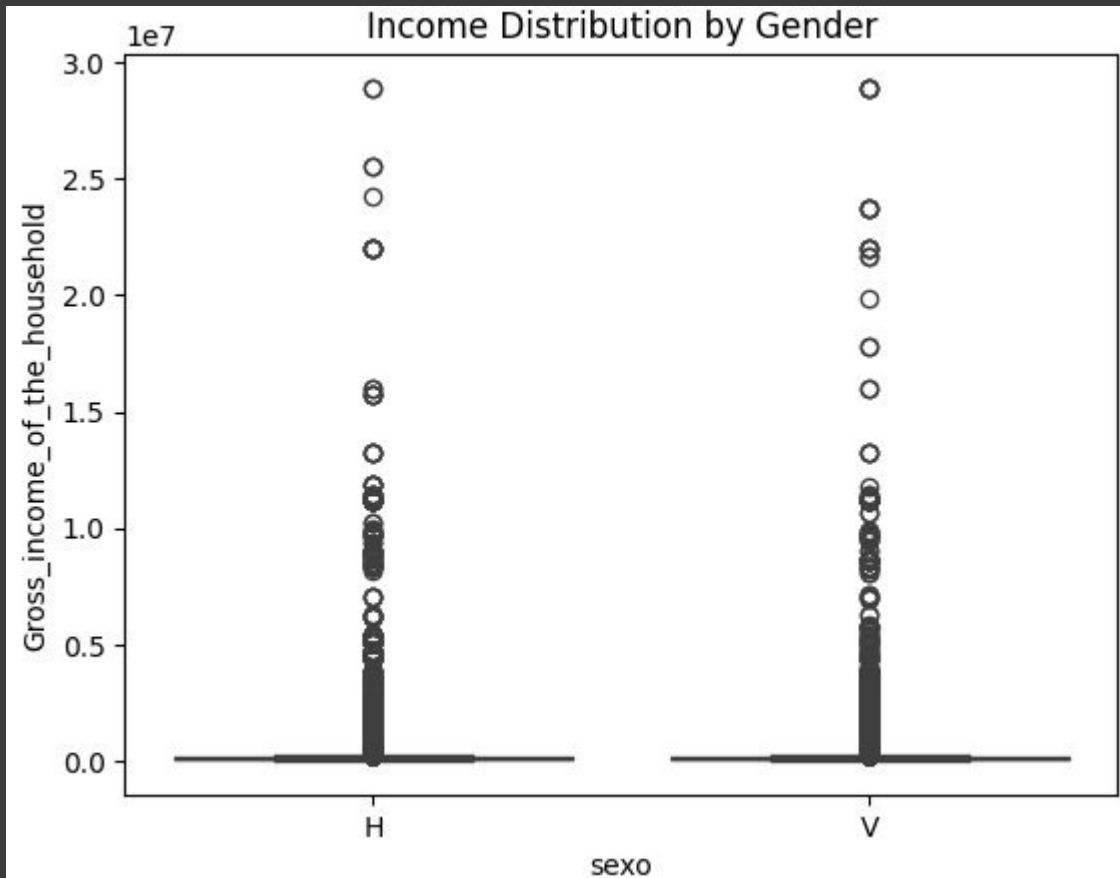
R (Potential)

In Customer's relation, 55.9% of customers are active while 44.1% of customers are inactive.



Age distribution of customer are
Male(H) shares 42.7% and Female(V)
shares 57.3%.

EDA

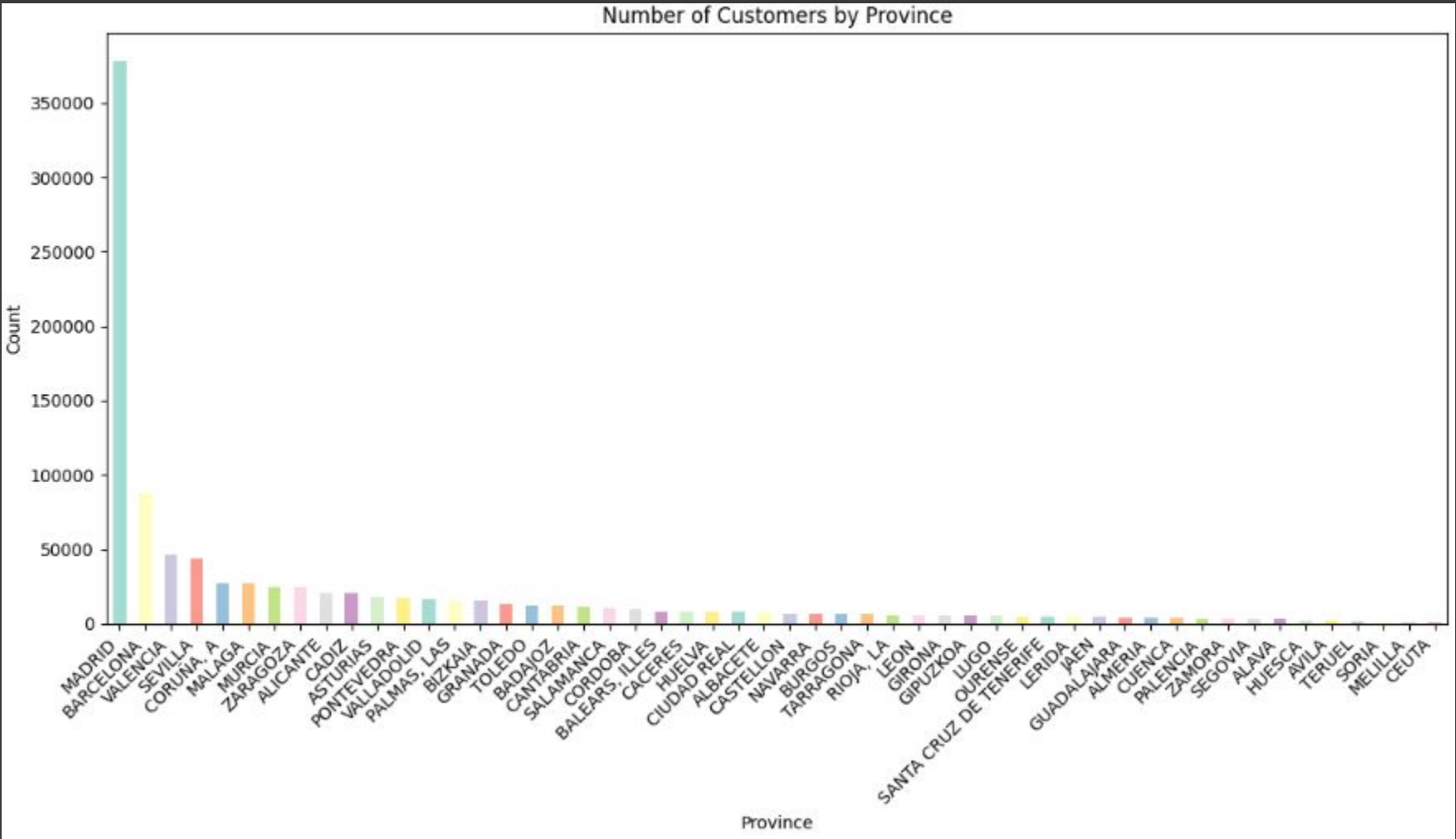


Checking the Outliers of Gender Distribution.

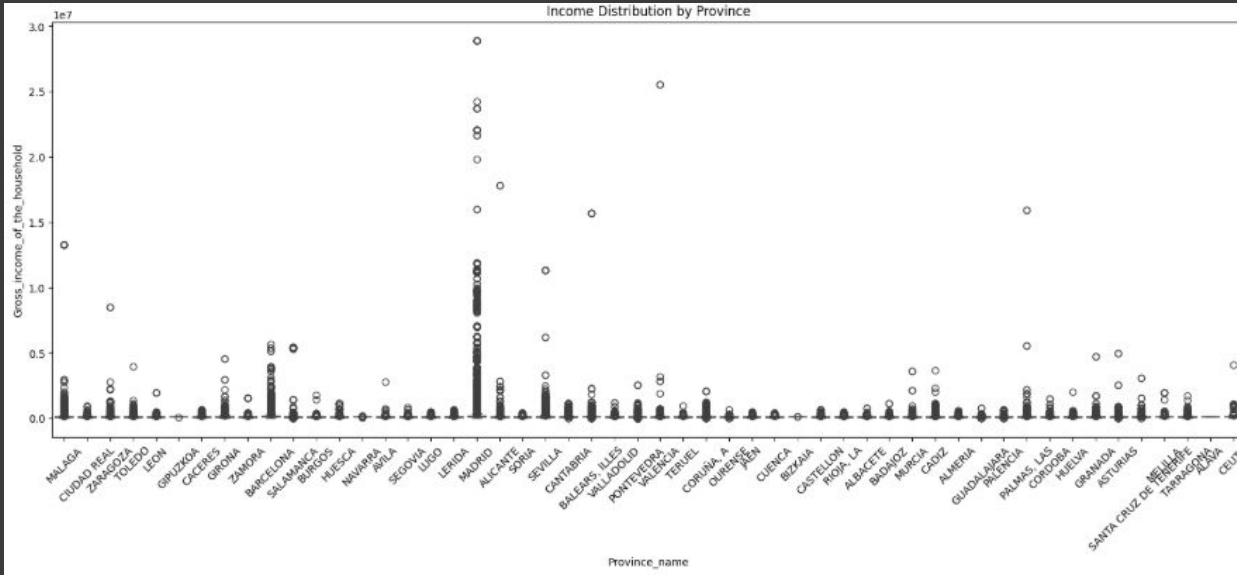
There are no significant differences in the distribution of gross income between Male(H) and Female(V). However, it seems slightly Female has more outliers concentrated above the median.



Madrid has the highest population of customers and the highest gross income household.



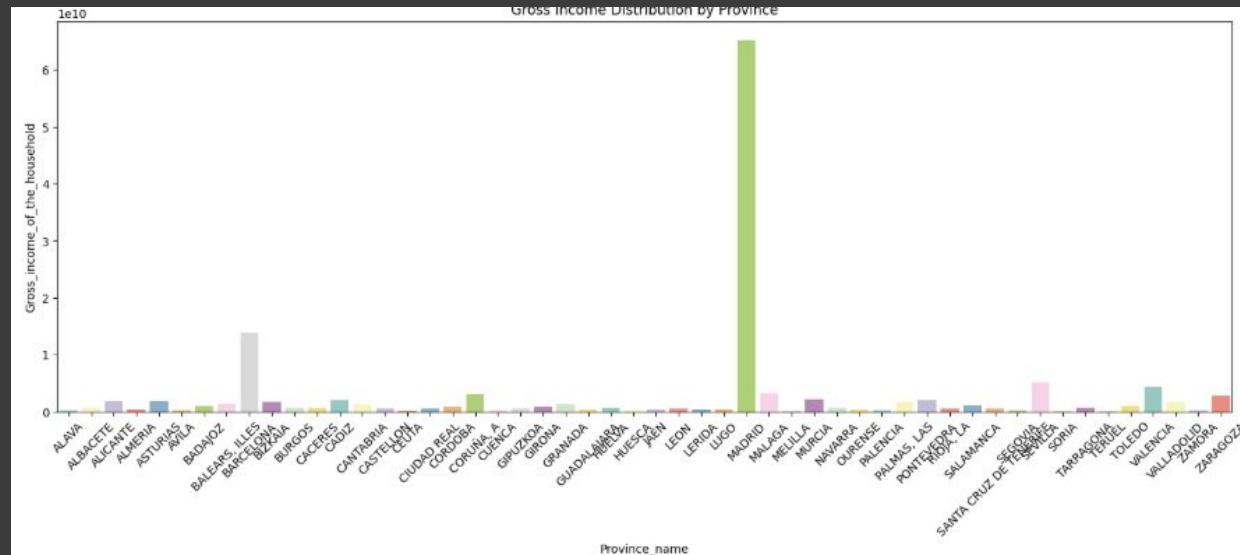
EDA



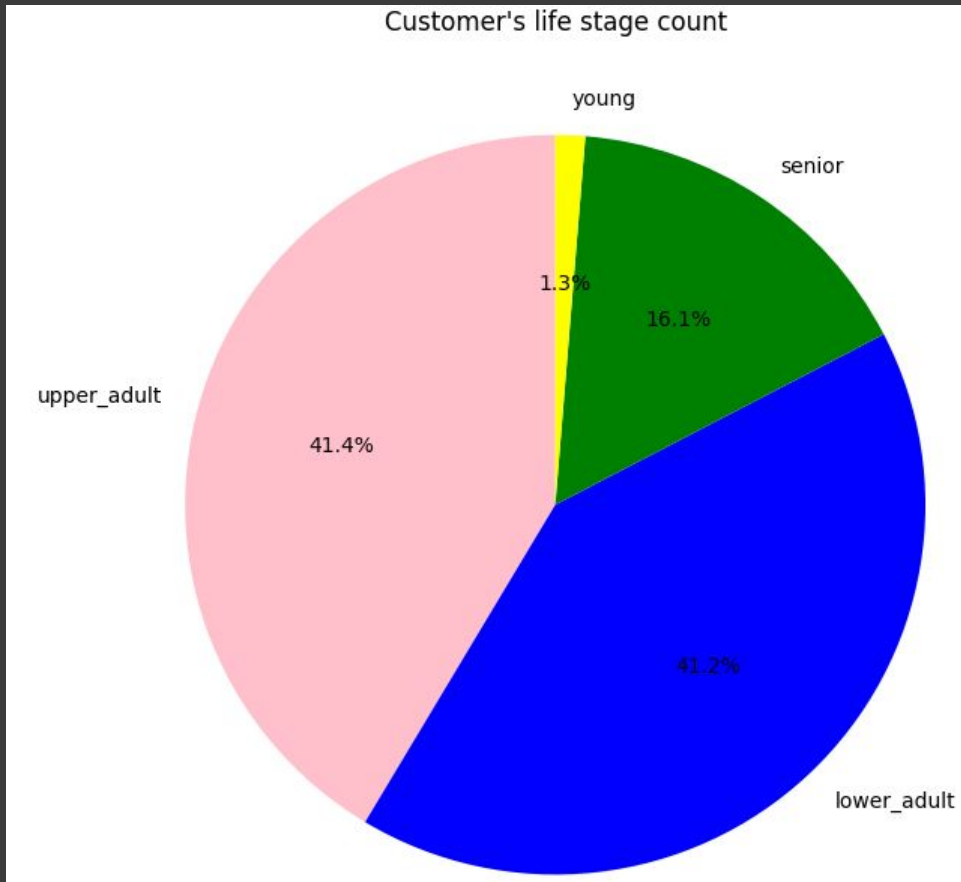
Regarding Income Distribution by Province in terms of outliers, Madrid has significant of amount of outliers.



Regarding Gross Income distribution by Province, Madrid has significant amount of income compared to other cities.



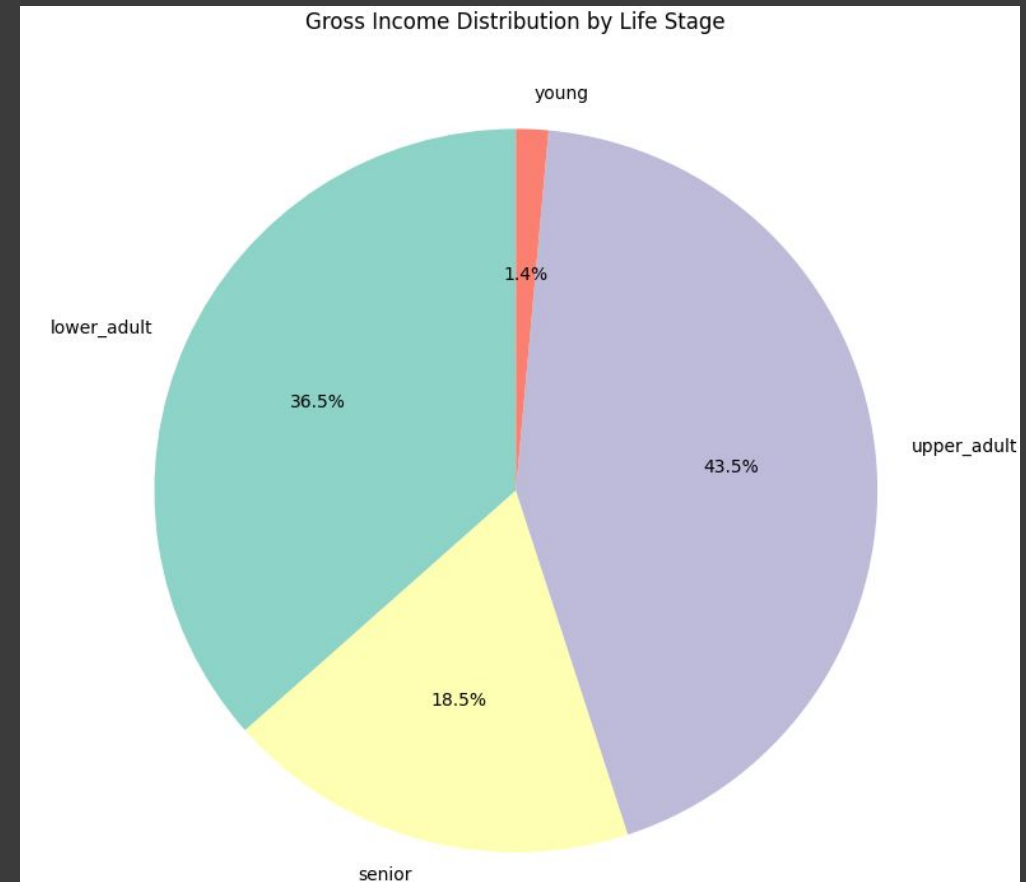
EDA



Majority of this bank customer is Adult(20-60 years old). On the other hand, Young(under 20 years old) is only 1.3% shared.

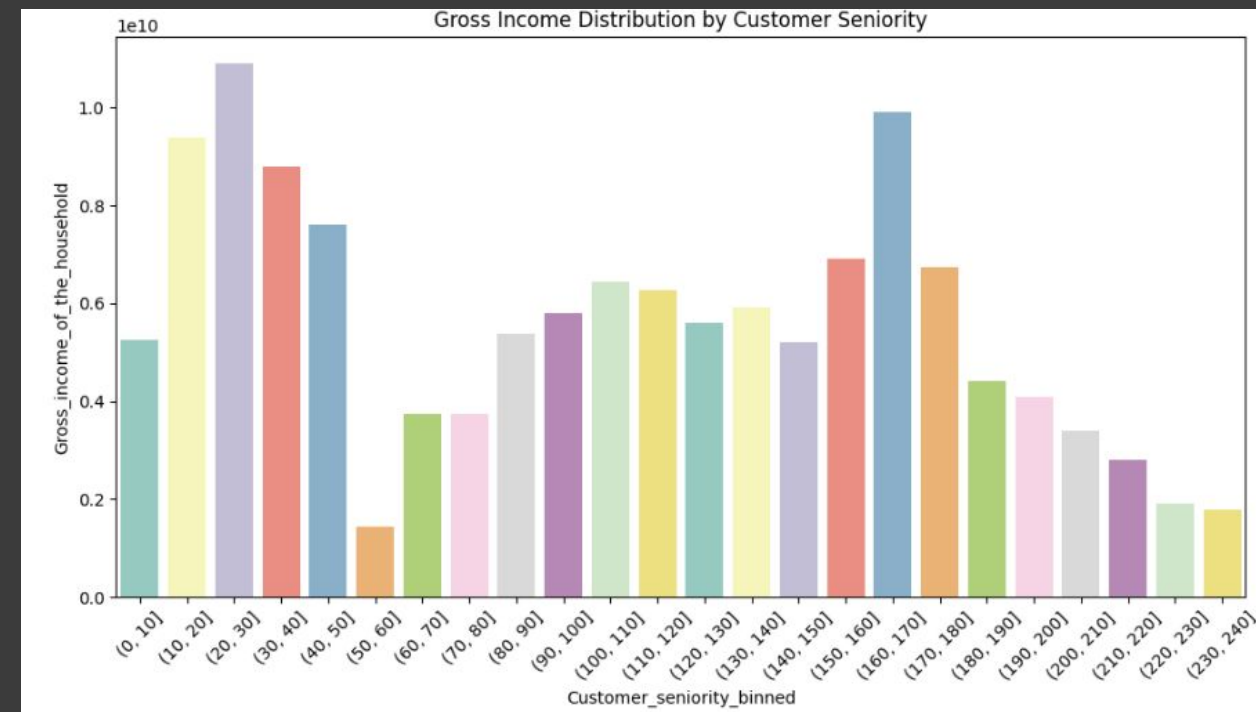


In term of the Gross Income Distribution by Life Stage, 43.5% is shared by Upper Adult and 36.5% is shared by Lower Adult.



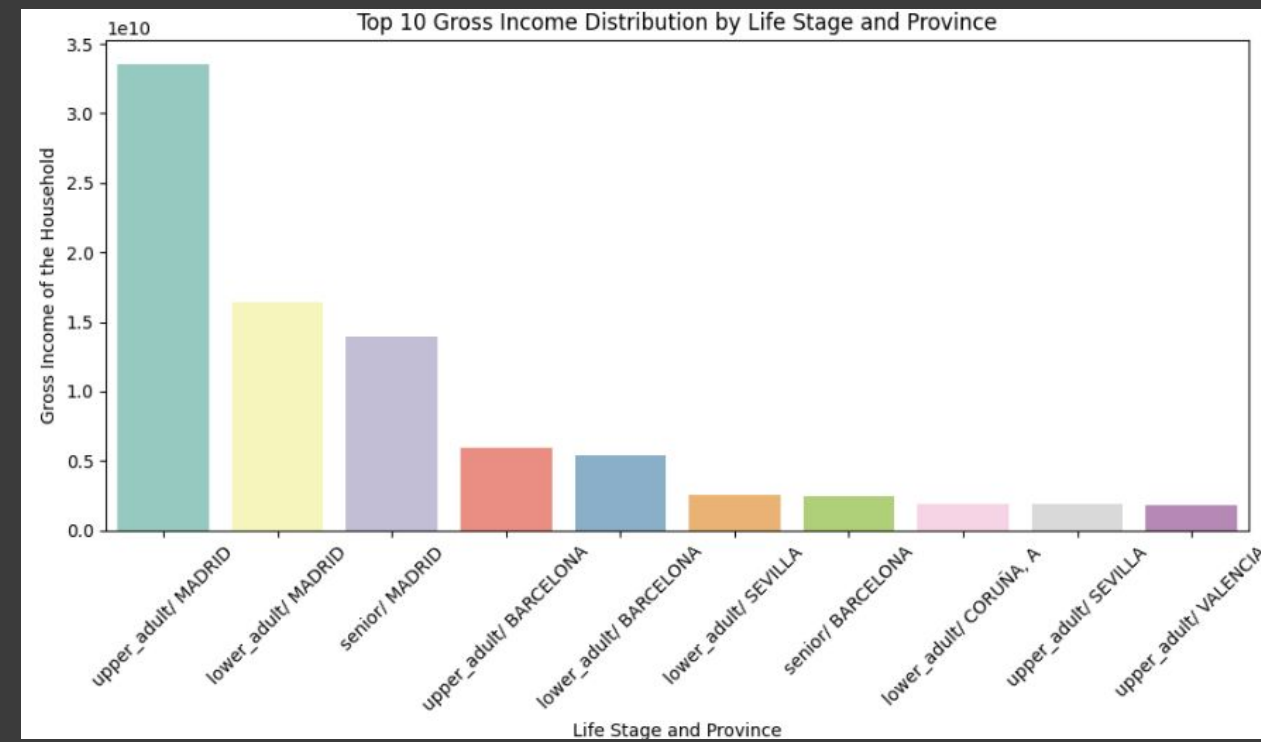
EDA

The highest group of customer seniority is 20-30 months. The next highest group is 160-170 months. The lowest group is 50-60 months. It may be some restrictions existing in this term such as some promotions being ended. From 180-240 months(15-20 years), there is a trend to decrease the number of seniority that indicates this bank cannot retain customers for the long term.



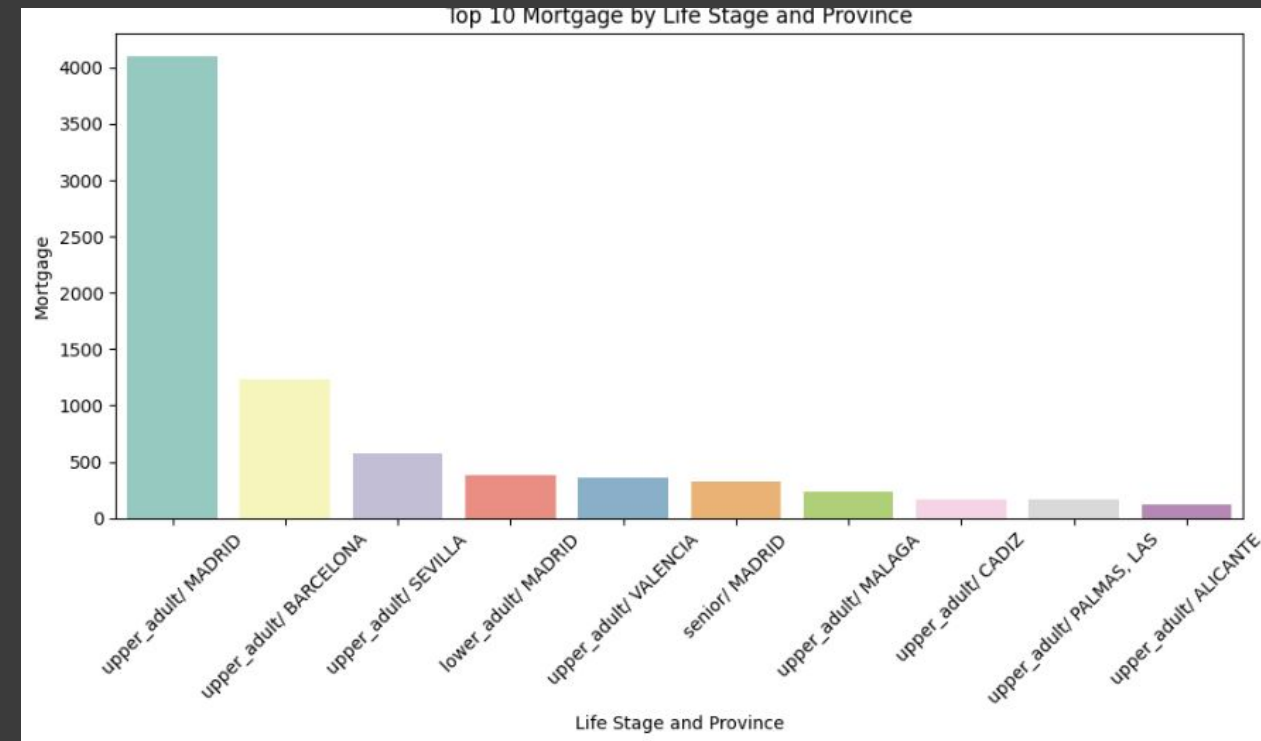


Regardless of the life stage group, customers who live in MADRID have the highest income. The customers who live in BARCELONA seem 2nd highest income group. But the life stage of the young is the minority in the any of provinces. There are no young life stage groups in this observation.



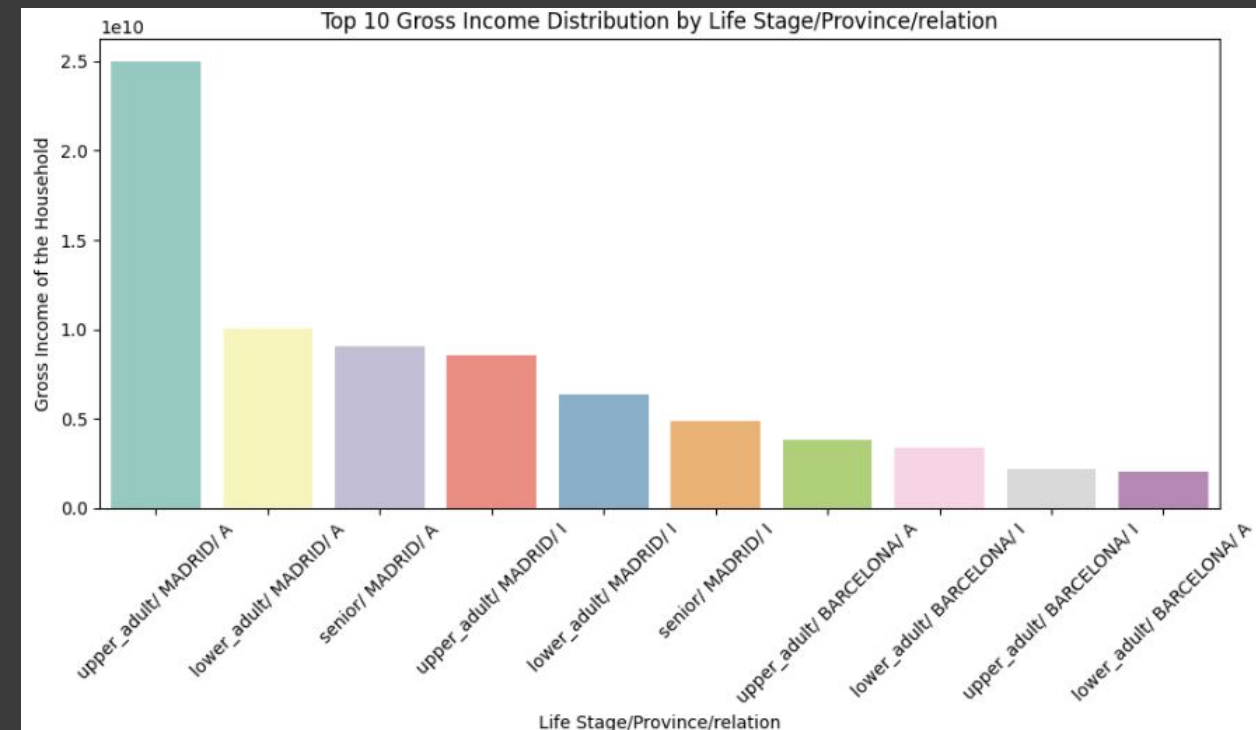


High income customers also have more Mortgage compared to low income customers. This plot relatively correlated to the plot of Gross Income Distribution by Life Stage and Province.



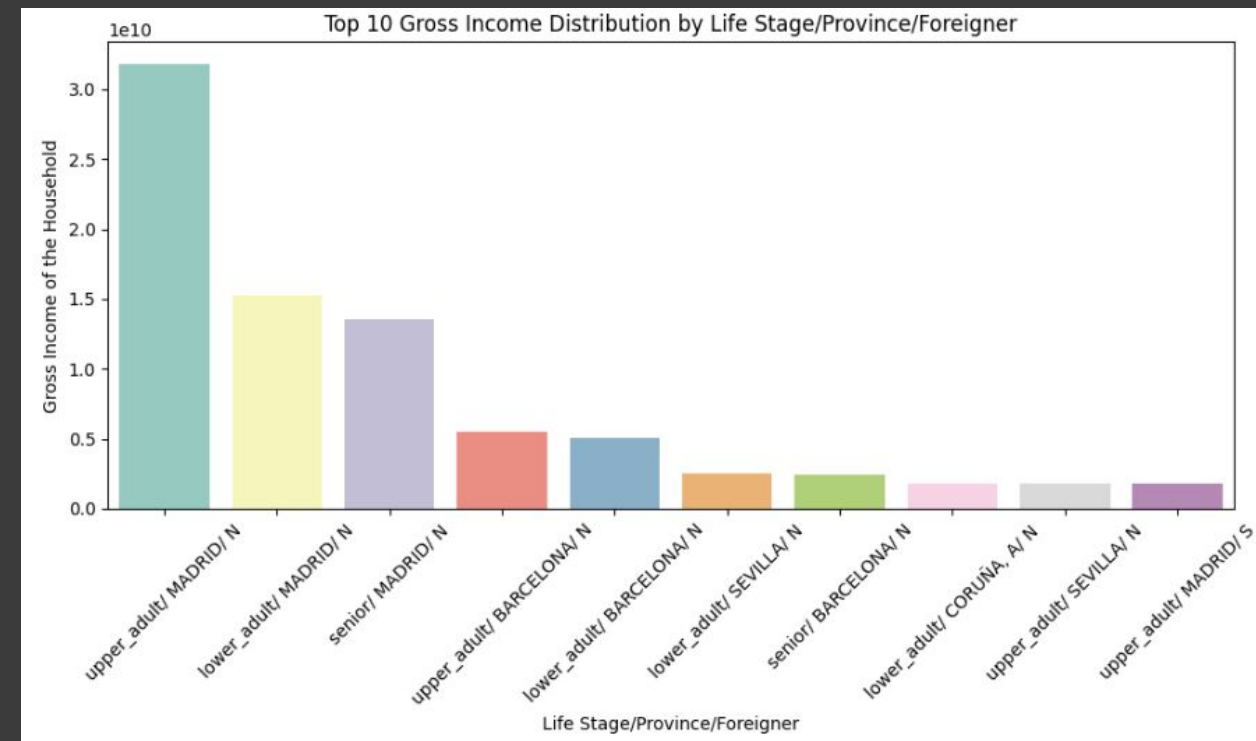


The group of Upper Adult/ Madrid is the most sharing the Active customers. Also, Upper Adult/ Madrid is the most sharing the Inactive customers.





The majority of high-income customers are from Residence (no foreigners) while upper_adult/MADRID/S shows they are foreigners.



Recommendation for technical users

K-Means Clustering Overview

K-Means Clustering is an unsupervised machine learning algorithm used to group data into **K clusters** based on their features. The algorithm iteratively assigns data points to clusters to minimize the variance within each cluster while maximizing the distance between clusters.

How It Works

1. **Initialization:**
 - Randomly select **KKK** initial cluster centroids (starting points for each cluster).
2. **Assignment:**
 - Assign each data point to the cluster whose centroid is closest based on a distance metric, typically **Euclidean distance**.
3. **Update:**
 - Recalculate the centroids as the mean of all data points assigned to each cluster.
4. **Repeat:**
 - Repeat the assignment and update steps until the centroids stabilize (i.e., no significant change in their positions) or a maximum number of iterations is reached.

Key Features

- **Number of Clusters:**
 - The number of clusters **KKK** must be specified beforehand. Techniques like the **Elbow Method** can help determine the optimal value of **KKK**.
- **Centroid-Based:**
 - Each cluster is represented by its centroid, which is the mean position of all points in the cluster.
- **Iterative Process:**
 - The algorithm continues to refine cluster assignments and centroids iteratively to minimize the **within-cluster sum of squares (WCSS)**.

Thank You