In [ ]:
```
Analysis of company's ideal customers by Chinenye
```

In [ ]:
```
1.1 Context
1.1.1 Problem Statement
Customer Personality Analysis is a detailed analysis of a company's ideal cust
a business to better understand its customers and makes it easier for them to
according to the specific needs, behaviors and concerns of different types of
Customer personality analysis helps a business to modify its product based on
from different types of customer segments. For example, instead of spending mo
new product to every customer in the company's database, a company can analyze
segment is most likely to buy the product and then market the product only on
segment.
```

In [ ]:
```
1.2 Content
1.2.1 Attributes
People
• ID: Customer's unique identifier
• Year_Birth: Customer's birth year
• Education: Customer's education level
• Marital_Status: Customer's marital status
• Income: Customer's yearly household income
• Kidhome: Number of children in customer's household
• Teenhome: Number of teenagers in customer's household
• Dt_Customer: Date of customer's enrollment with the company
• Recency: Number of days since customer's last purchase
• Complain: 1 if the customer complained in the last 2 years, 0 otherwise
Products
• MntWines: Amount spent on wine in last 2 years
• MntFruits: Amount spent on fruits in last 2 years
1
• MntMeatProducts: Amount spent on meat in last 2 years
• MntFishProducts: Amount spent on fish in last 2 years
• MntSweetProducts: Amount spent on sweets in last 2 years
• MntGoldProds: Amount spent on gold in last 2 years
Promotion
• NumDealsPurchases: Number of purchases made with a discount
• AcceptedCmp1: 1 if customer accepted the offer in the 1st campaign, 0 otherw
• AcceptedCmp2: 1 if customer accepted the offer in the 2nd campaign, 0 otherw
• AcceptedCmp3: 1 if customer accepted the offer in the 3rd campaign, 0 otherw
• AcceptedCmp4: 1 if customer accepted the offer in the 4th campaign, 0 otherw
• AcceptedCmp5: 1 if customer accepted the offer in the 5th campaign, 0 otherw
• Response: 1 if customer accepted the offer in the last campaign, 0 otherwise
Place
• NumWebPurchases: Number of purchases made through the company's website
• NumCatalogPurchases: Number of purchases made using a catalogue
• NumStorePurchases: Number of purchases made directly in stores
• NumWebVisitsMonth: Number of visits to company's website in the last month
```

In [ ]:
```
1.3 Target
Need to perform clustering to summarize customer segments.
```

In [108]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
plt.style.use('seaborn-v0_8')
```

In [9]:
```
df=pd.read_excel(r"C:\Users\Chinenye Claire\Desktop\Hamoye Internship\Part B\ma
```

In [12]:
```
data=df.rename(columns={'NumWebPurchases': "Web",'NumCatalogPurchases':'Catalo
```

In [15]:
```
data=data.rename(columns={'MntWines': "Wines",'MntFruits':'Fruits','MntMeatPro
```

In [14]:
```
data['Education']=data['Education'].replace({'Basic':'Undergraduate','2n Cycle
```

In [106]:
```
data['Marital_Status']=data['Marital_Status'].replace({'Divorced':'Alone','Sing
```

In [107]:
```
data.head()
```

Out[107]:

	ID	Year_Birth	Age	Age_Range	Education	Marital_Status	Income	Income_Range	Kidh
0	5524	1957	66	Aged	Postgraduate	Alone	58138.0	C	
1	2174	1954	69	Aged	Postgraduate	Alone	46344.0	B	
2	4141	1965	58	Adult	Postgraduate	In couple	71613.0	C	
3	6182	1984	39	Adult	Postgraduate	In couple	26646.0	B	
4	5324	1981	42	Adult	Postgraduate	In couple	58293.0	C	

5 rows × 34 columns

In [91]:
```
data.tail()
```

Out[91]:

	ID	Year_Birth	Age	Age_Range	Education	Marital_Status	Income	Income_Range
2235	10870	1967	56	Adult	Postgraduate	In couple	61223.0	C
2236	4001	1946	77	Aged	Postgraduate	In couple	64014.0	C
2237	7270	1981	42	Adult	Postgraduate	Alone	56981.0	C
2238	8235	1956	67	Aged	Postgraduate	In couple	69245.0	C
2239	9405	1954	69	Aged	Postgraduate	In couple	52869.0	C

5 rows × 34 columns

In [92]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2240 entries, 0 to 2239
Data columns (total 34 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   ID                  2240 non-null   int64
 1   Year_Birth          2240 non-null   int64
 2   Age                 2240 non-null   int64
 3   Age_Range           2240 non-null   object
 4   Education           2240 non-null   object
 5   Marital_Status      2240 non-null   object
 6   Income              2216 non-null   float64
 7   Income_Range        2240 non-null   object
 8   Kidhome             2240 non-null   int64
 9   Teenhome            2240 non-null   int64
 10  Dt_Customer         2240 non-null   datetime64[ns]
 11  Recency             2240 non-null   int64
 12  Wines               2240 non-null   int64
 13  Fruits              2240 non-null   int64
 14  Meat                2240 non-null   int64
 15  Fish                2240 non-null   int64
 16  Sweets              2240 non-null   int64
 17  Gold                2240 non-null   int64
 18  Deals               2240 non-null   int64
 19  Web                 2240 non-null   int64
 20  Catalog             2240 non-null   int64
 21  Store               2240 non-null   int64
 22  NumWebVisitsMonth   2240 non-null   int64
 23  AcceptedCmp1        2240 non-null   int64
 24  AcceptedCmp2        2240 non-null   int64
 25  AcceptedCmp3        2240 non-null   int64
 26  AcceptedCmp4        2240 non-null   int64
 27  AcceptedCmp5        2240 non-null   int64
 28  Response            2240 non-null   int64
 29  Complain            2240 non-null   int64
 30  Z_CostContact       2240 non-null   int64
 31  Z_Revenue           2240 non-null   int64
 32  Children            2240 non-null   object
 33  Has_child           2240 non-null   object
dtypes: datetime64[ns](1), float64(1), int64(26), object(6)
memory usage: 595.1+ KB
```

In [93]:
```
data.describe()
```

Out[93]:

	ID	Year_Birth	Age	Income	Kidhome	Teenhome	Rec
count	2240.000000	2240.000000	2240.000000	2216.000000	2240.000000	2240.000000	2240.00
mean	5592.159821	1968.805804	54.194196	52247.251354	0.444196	0.506250	49.10
std	3246.662198	11.984069	11.984069	25173.076661	0.538398	0.544538	28.96
min	0.000000	1893.000000	27.000000	1730.000000	0.000000	0.000000	0.00
25%	2828.250000	1959.000000	46.000000	35303.000000	0.000000	0.000000	24.00
50%	5458.500000	1970.000000	53.000000	51381.500000	0.000000	0.000000	49.00
75%	8427.750000	1977.000000	64.000000	68522.000000	1.000000	1.000000	74.00
max	11191.000000	1996.000000	130.000000	666666.000000	2.000000	2.000000	99.00

8 rows × 27 columns

In [94]:
```
data.isnull().sum()
```

Out[94]:
```
ID                   0
Year_Birth           0
Age                  0
Age_Range            0
Education            0
Marital_Status       0
Income              24
Income_Range         0
Kidhome              0
Teenhome             0
Dt_Customer          0
Recency              0
Wines                0
Fruits               0
Meat                 0
Fish                 0
Sweets               0
Gold                 0
Deals                0
Web                  0
Catalog              0
Store                0
NumWebVisitsMonth    0
AcceptedCmp1         0
AcceptedCmp2         0
AcceptedCmp3         0
AcceptedCmp4         0
AcceptedCmp5         0
Response             0
Complain             0
Z_CostContact        0
Z_Revenue            0
Children             0
Has_child            0
dtype: int64
```

In [95]:
```
df=data.dropna()
```

In [96]:
```
df.isnull().sum().any()
```

Out[96]:
```
False
```

In [97]:
```
#grouping total number of customers based on their level of education: over pos
df.value_counts("Education")
```

Out[97]:
```
Education
Postgraduate     1962
Undergraduate     254
dtype: int64
```

In [98]:
```
#grouping total number of customers based on their marital status: more than ha
df.value_counts("Marital_Status")
```

Out[98]:
```
Marital_Status
In couple    1430
Alone         786
dtype: int64
```

In [99]:
```
#grouping customers based on income: most of them earn over 50000 annually
# A means <=25000
# B means <=50000
#C means >50000
df.value_counts("Income_Range")
```

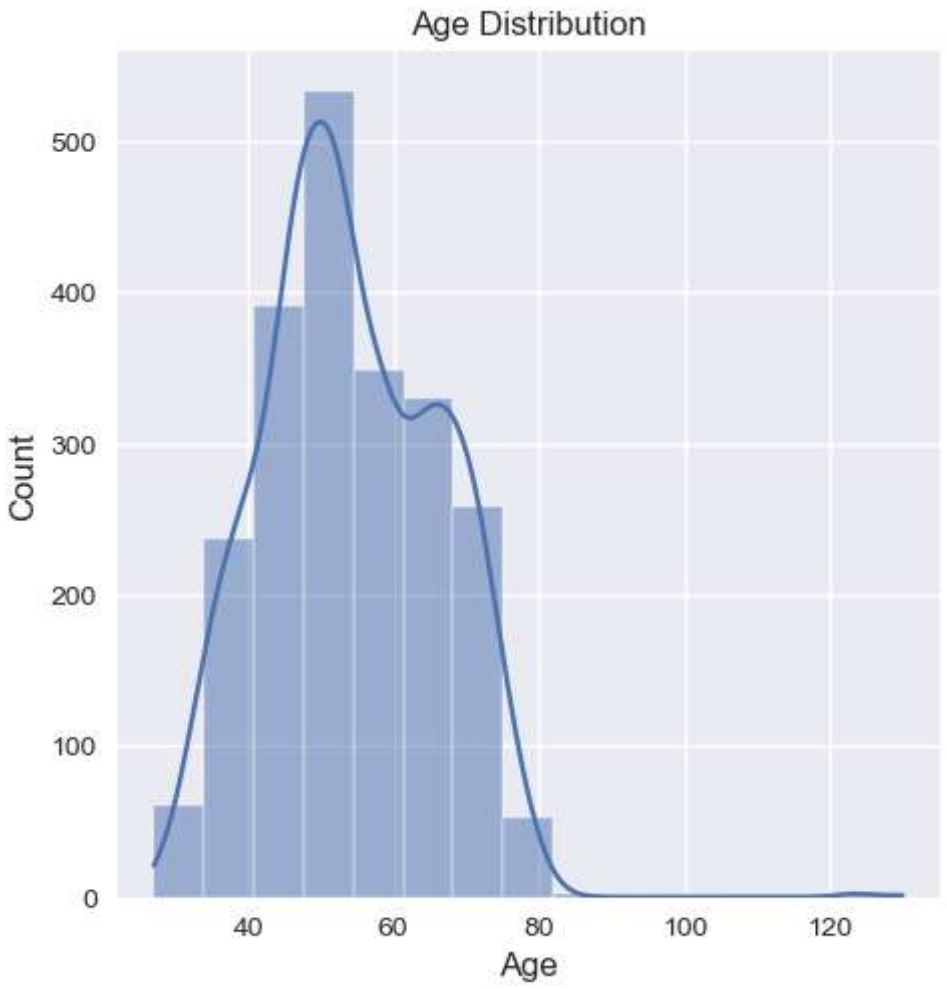
Out[99]:
```
Income_Range
C    1156
B     818
A     242
dtype: int64
```

In [100]:
```
#grouping customers by age: over 65% are Adults
#Youth means <30 years
#Adult means >30<=60 years
#Aged means >60 years
df.value_counts("Age_Range")
```

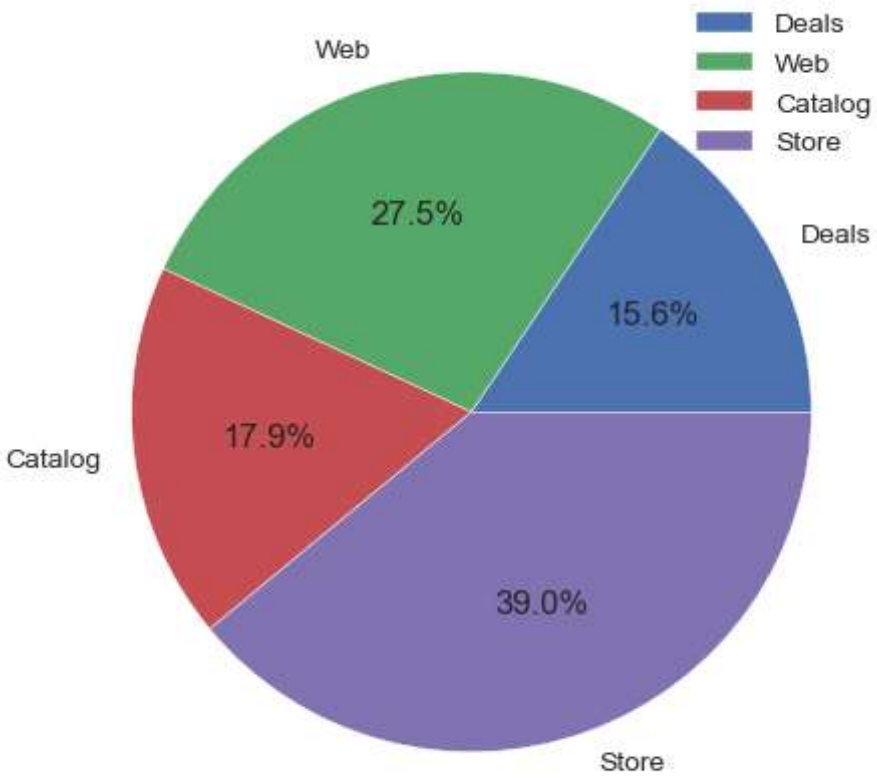
Out[100]:
```
Age_Range
Adult    1511
Aged      690
Youth      15
dtype: int64
```

In [159]:
```
sns.displot(df['Age'], kde=True, bins=15)
plt.title('Age Distribution')
```

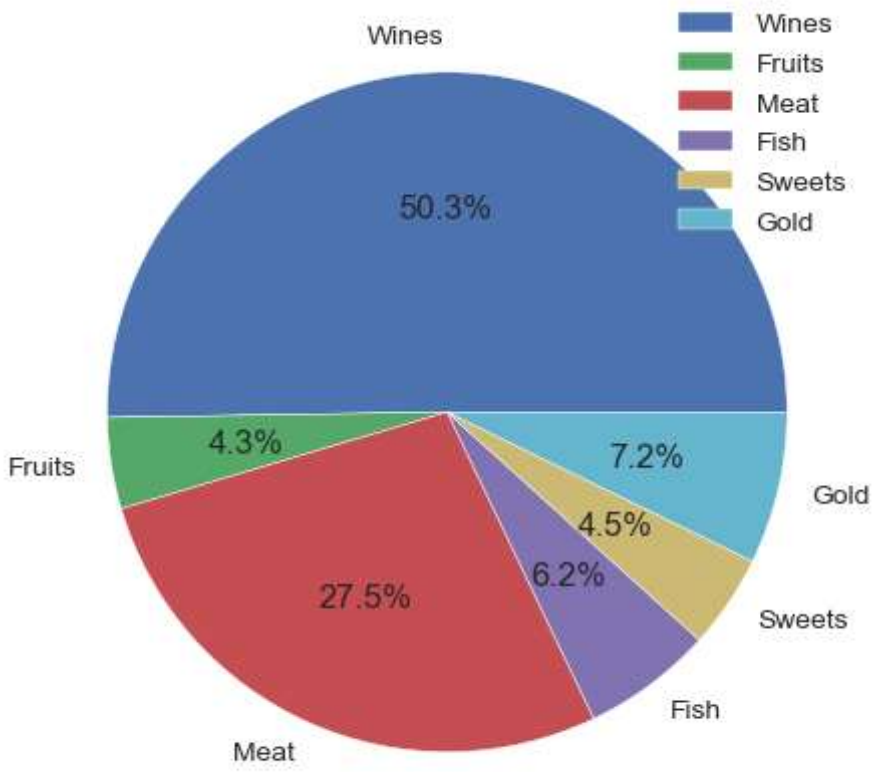
Out[159]: Text(0.5, 1.0, 'Age Distribution')



In [101]:
```
df[["Deals","Web","Catalog","Store"]].sum()
y=df[["Deals","Web","Catalog","Store"]].sum()
mylabels=["Deals","Web","Catalog","Store"]
plt.pie(y, labels=mylabels, autopct='%1.1f%%')
plt.legend()
plt.show()
```

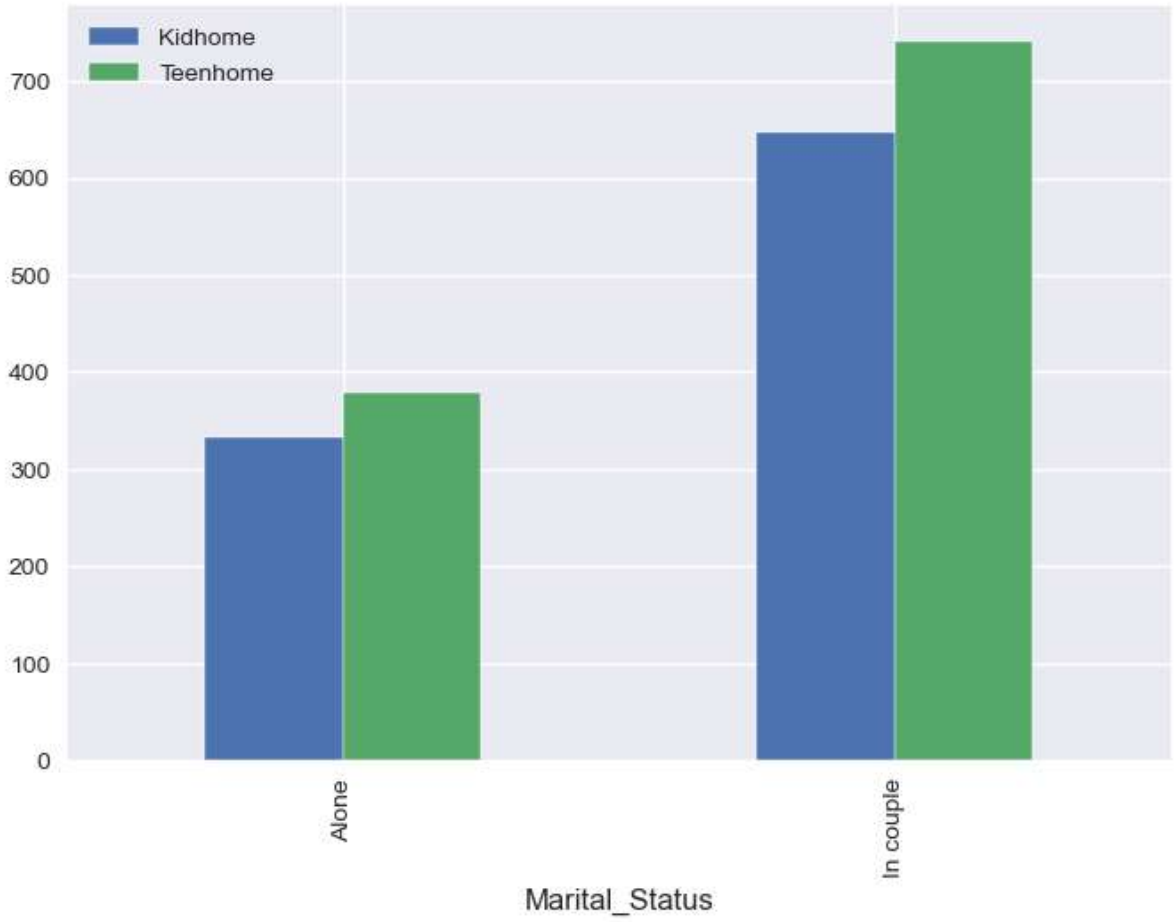


In [102]:
```
df[["Wines","Fruits","Meat","Fish","Sweets","Gold"]].sum()
z=df[["Wines","Fruits","Meat","Fish","Sweets","Gold"]].sum()
mylabels=["Wines","Fruits","Meat","Fish","Sweets","Gold"]
plt.pie(z, labels=mylabels, autopct='%1.1f%%')
plt.legend()
plt.show()
```



In [137]:
```
#considering customers that have kids and teens:married ones have the most numb
tab=df.groupby("Marital_Status")["Kidhome","Teenhome"].sum()
tab.plot(kind='bar')
```

Out[137]: <Axes: xlabel='Marital_Status'>



In [136]:
```
#Grouping customers based on level of education to show the company's most frequ
#undergraduates used all routes the most
df.groupby("Education")["Deals","Web","Catalog","Store"].sum()
```

Out[136]:

Education	Deals	Web	Catalog	Store
Postgraduate	4600	8198	5422	11589
Undergraduate	549	855	497	1266

In [138]:
```
df.groupby("Marital_Status")["Deals","Web","Catalog","Store"].sum()
```

Out[138]:

Marital_Status	Deals	Web	Catalog	Store
Alone	1773	3201	2130	4544
In couple	3376	5852	3789	8311

In [139]:
```
df.groupby("Income_Range")["Deals","Web","Catalog","Store"].sum()
```

Out[139]:

Income_Range	Deals	Web	Catalog	Store
A	536	493	127	662
B	1974	2286	704	3062
C	2639	6274	5088	9131

In [140]:
```
df.groupby("Age_Range")["Deals","Web","Catalog","Store"].sum()
```

Out[140]:

Age_Range	Deals	Web	Catalog	Store
Adult	3521	5863	3589	8293
Aged	1611	3135	2257	4464
Youth	17	55	73	98

In [145]:
```
df.groupby("Education")["Wines","Fruits","Meat","Fish","Sweets","Gold"].sum()
```

Out[145]:

Education	Wines	Fruits	Meat	Fish	Sweets	Gold
Postgraduate	635523	51933	342429	72876	52297	86818
Undergraduate	40560	6472	27634	10529	7599	10609

In [146]:
```
df.groupby("Marital_Status")["Wines","Fruits","Meat","Fish","Sweets","Gold"].s
```

Out[146]:

Marital_Status	Wines	Fruits	Meat	Fish	Sweets	Gold
Alone	242392	21812	136801	30627	21939	35954
In couple	433691	36593	233262	52778	37957	61473

In [147]:
```
df.groupby("Age_Range")["Wines","Fruits","Meat","Fish","Sweets","Gold"].sum()
```

Out[147]:

Age_Range	Wines	Fruits	Meat	Fish	Sweets	Gold
Adult	406924	38119	233062	52144	38742	62367
Aged	263802	19637	131874	29855	20463	34018
Youth	5357	649	5127	1406	691	1042

In [148]:
```
df.groupby("Income_Range")["Wines","Fruits","Meat","Fish","Sweets","Gold"].sum
```

Out[148]:

Income_Range	Wines	Fruits	Meat	Fish	Sweets	Gold
A	2688	1475	5251	1910	1531	4563
B	69017	5782	30936	9537	5892	18342
C	604378	51148	333876	71958	52473	74522

In [161]:
```
df.groupby("Age_Range")["AcceptedCmp1","AcceptedCmp2","AcceptedCmp3","Accepted
```

Out[161]:

Age_Range	AcceptedCmp1	AcceptedCmp2	AcceptedCmp3	AcceptedCmp4	AcceptedCmp5
Adult	84	17	123	95	103
Aged	56	12	38	68	55
Youth	2	1	2	1	4

In [162]:
```
df.groupby("Education")["AcceptedCmp1","AcceptedCmp2","AcceptedCmp3","Accepted
```

Out[162]:

Education	AcceptedCmp1	AcceptedCmp2	AcceptedCmp3	AcceptedCmp4	AcceptedCmp5
Postgraduate	128	28	142	155	152
Undergraduate	14	2	21	9	10

In [163]:
```
df.groupby("Income_Range")["AcceptedCmp1","AcceptedCmp2","AcceptedCmp3","Accep
```

Out[163]:

Income_Range	AcceptedCmp1	AcceptedCmp2	AcceptedCmp3	AcceptedCmp4	AcceptedCmp5
A	0	0	20	0	0
B	3	4	69	17	0
C	139	26	74	147	162

In [164]:
```
df.groupby("Marital_Status")["AcceptedCmp1","AcceptedCmp2","AcceptedCmp3","Acc
```

Out[164]:

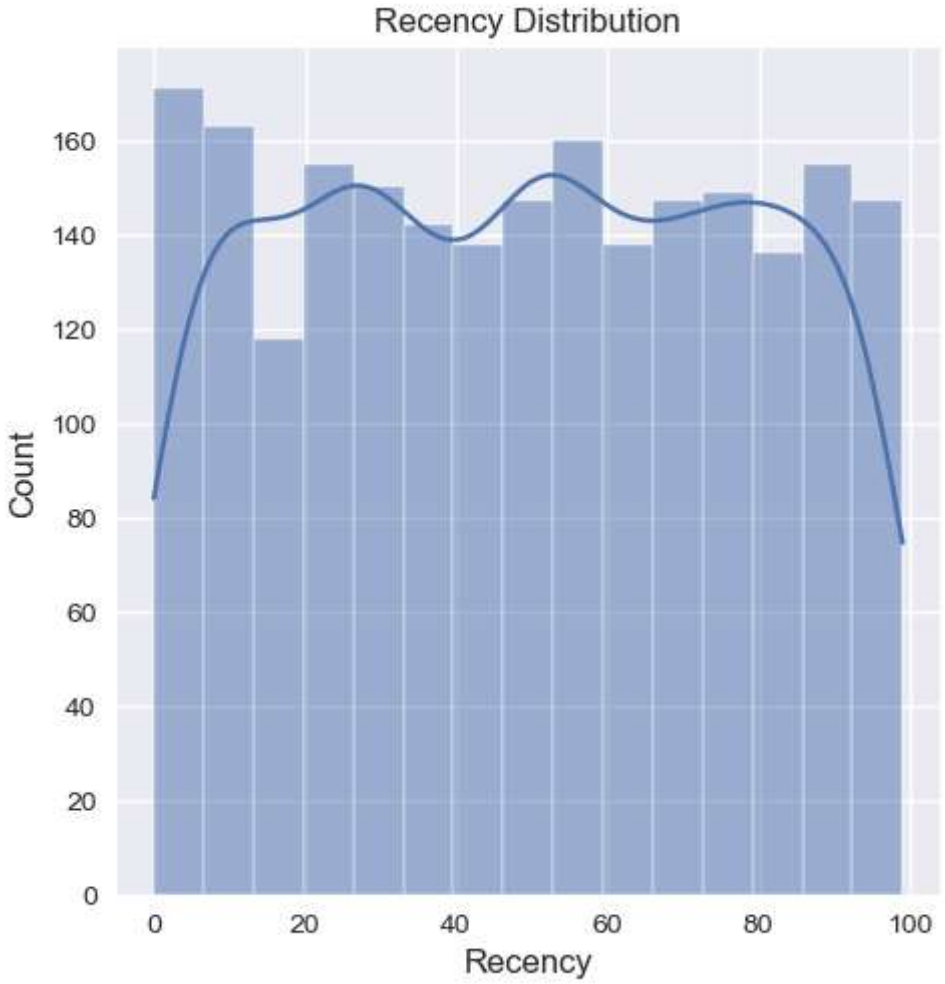
Marital_Status	AcceptedCmp1	AcceptedCmp2	AcceptedCmp3	AcceptedCmp4	AcceptedCmp5
Alone	48	11	63	60	52
In couple	94	19	100	104	110

In [156]:
```
df['Recency'].mean()
```

Out[156]:
```
49.01263537906137
```

In [158]:
```
sns.displot(df['Recency'], kde=True, bins=15)
plt.title('Recency Distribution')
```

Out[158]:
```
Text(0.5, 1.0, 'Recency Distribution')
```



In [160]:
```
#to find the correlation among the columns using pearson method
print(df.corr(method = 'pearson'))
```

```
                          ID  Year_Birth       Age    Income   Kidhome  \
ID                  1.000000    0.002355 -0.002355  0.013095  0.001736
Year_Birth          0.002355    1.000000 -1.000000 -0.161791  0.233615
Age                -0.002355   -1.000000  1.000000  0.161791 -0.233615
Income              0.013095   -0.161791  0.161791  1.000000 -0.428669
Kidhome             0.001736    0.233615 -0.233615 -0.428669  1.000000
Teenhome           -0.003030   -0.350791  0.350791  0.019133 -0.039869
Recency            -0.044376   -0.016295  0.016295 -0.003970  0.011492
Wines              -0.021084   -0.159451  0.159451  0.578650 -0.497336
Fruits              0.007326   -0.017747  0.017747  0.430842 -0.373396
Meat               -0.005902   -0.033697  0.033697  0.584633 -0.439261
Fish               -0.023992   -0.040425  0.040425  0.438871 -0.388884
Sweets             -0.005936   -0.020204  0.020204  0.440744 -0.378026
Gold               -0.011172   -0.064208  0.064208  0.325916 -0.355029
Deals              -0.040612   -0.058668  0.058668 -0.083101  0.216913
Web                -0.018476   -0.153051  0.153051  0.387878 -0.371977
Catalog            -0.002274   -0.121764  0.121764  0.589162 -0.504501
Store              -0.013070   -0.127891  0.127891  0.529362 -0.501349
NumWebVisitsMonth  -0.007794    0.123904 -0.123904 -0.553088  0.447477
```

In [ ]:
```
#output and overall analysis conducted on this data science project on customer
The biggest customers are: postgraduates who are mostly above 60 years of age,
Store and web purchases make up over 50% of the routes frequently used by cust
Wines are the most consumed products.
Customer Average visit in the last 2 years is less than 3 months.
Participation in most campaigns were by couples, Postgraduates, high income ear
```