In today's world, cars play a crucial role in providing us with freedom and control over our daily lives. Whether it's commuting to work, running errands, or visiting loved ones, cars are indispensable for many people. Research indicates that a significant majority, over 76%, feel restricted in their travel options without access to a car. People typically choose their cars based on their specific needs and preferences.

Recognizing the importance of understanding customer preferences and purchasing behavior in the US market, Austo, a UK-based automotive company, has enlisted the help of a consulting firm. This firm has conducted various surveys to compile a dataset encompassing three primary types of cars widely used in the US. This dataset contains valuable information about car owners, offering insights into the US automobile market.

Austo's management team aims to grasp the demand trends in the US market and gain insights into buyer behavior. They seek to develop customer profiles through analysis, identifying new opportunities for car purchases. This understanding will enable them to adapt their business strategy and production to meet market demands effectively. Additionally, it will provide valuable insights into navigating a new market landscape.

As a Data Scientist at the consulting firm working with Austo, you're tasked with creating buyer profiles for different car types using the available data. Furthermore, you're expected to provide recommendations to Austo based on your analysis, aiding them in expanding their business successfully.

- Age: Age of the customer
- Gender: Gender of the customer
- Profession: Indicates whether the customer is a salaried or business person
- Marital_status: Marital status of the customer
- Education: Refers to the highest level of education completed by the customer
- No_of_dependents: Number of dependents(partner/children/spouse) of the customer
- Personal_loan: Indicates whether the customer availed a personal loan or not
- House_loan: Indicates whether the customer availed house loan or not
- Partner_working: Indicates whether the customer's partner is working or not
- Salary: Annual Salary of the customer
- Partner_salary: Annual Salary of the customer's partner
- Total_salary: Annual household income (Salary + Partner_salary) of the customer's family
- Price: Price of the car
- Make: Car type (Hatchback/Sedan/SUV)

```
import numpy as np
import pandas as pd

# import libraries for data visualization
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
df = pd.read_csv('/content/austo_automobile.csv')
df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 1581,\n  \"fields\":
[\n    {\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 9,\n        \"min\": 22,\n
\"max\": 60,\n        \"num_unique_values\": 39,\n        \"samples\":
[\n          45,\n          50,\n          36\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Gender\",\n      \"properties\":
{\n        \"dtype\": \"category\",\n        \"num_unique_values\":
2,\n        \"samples\": [\n          \"Female\",\n          \"Male\"\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Profession\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Business\",\n          \"Salaried\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Marital_status\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n
\"Single\",\n          \"Married\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Education\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n
\"Graduate\",\n          \"Post Graduate\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"No_of_Dependents\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0,\n        \"min\": 0,\n        \"max\": 4,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          3,\n
1\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Personal_loan\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Yes\",\n          \"No\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"House_loan\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 2,\n        \"samples\": [\n          \"No\",\n
\"Yes\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Partner_working\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"No\",\n          \"Yes\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Salary\",\n      \"properties\":
```

{\n   \"dtype\": \"number\",\n   \"std\": 14278,\n
\"min\": 30000,\n   \"max\": 90000,\n
\"num_unique_values\": 61,\n   \"samples\": [\n   52000,\n
48000\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Partner_salary\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 19480,\n   \"min\": 0,\n
\"max\": 80000,\n   \"num_unique_values\": 19,\n
\"samples\": [\n   25000,\n   50000\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Total_salary\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
24855,\n   \"min\": 30000,\n   \"max\": 158000,\n
\"num_unique_values\": 123,\n   \"samples\": [\n   89000,\
n   39000\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Price\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 13633,\n   \"min\": 18000,\n   \"max\": 70000,\n
\"num_unique_values\": 53,\n   \"samples\": [\n   23000,\n
47000\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Make\",\n   \"properties\": {\n   \"dtype\": \"category\",\n
\"num_unique_values\": 3,\n   \"samples\": [\n
\"Hatchback\",\n   \"SUV\"\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   }\n]\n}","type":"dataframe","variable_name":"df"}

```
df.shape
```

```
(1581, 14)
```

```
df.info()
df.dtypes
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1581 entries, 0 to 1580
Data columns (total 14 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Age               1581 non-null   int64
 1   Gender            1581 non-null   object
 2   Profession        1581 non-null   object
 3   Marital_status    1581 non-null   object
 4   Education         1581 non-null   object
 5   No_of_Dependents  1581 non-null   int64
 6   Personal_loan     1581 non-null   object
 7   House_loan        1581 non-null   object
 8   Partner_working   1581 non-null   object
 9   Salary            1581 non-null   int64
 10  Partner_salary    1581 non-null   int64
```

```
 11  Total_salary      1581 non-null   int64
 12  Price             1581 non-null   int64
 13  Make              1581 non-null   object
dtypes: int64(6), object(8)
memory usage: 173.0+ KB
```

```
Age                   int64
Gender               object
Profession           object
Marital_status       object
Education            object
No_of_Dependents      int64
Personal_loan        object
House_loan           object
Partner_working      object
Salary                int64
Partner_salary        int64
Total_salary          int64
Price                 int64
Make                 object
dtype: object
```

###observation: there are 1581 rows and 14 column in thee data set

```
df['Age'].unique()
df['Gender'].unique()
df['Profession'].unique()
df['Marital_status'].unique()
df['Education'].unique()
df['No_of_Dependents'].unique()
df['Personal_loan'].unique()
df['Salary'].unique()
df['Partner_salary'].unique()
df['Total_salary'].unique()
df['Price'].unique()
df['Make'].unique()
```

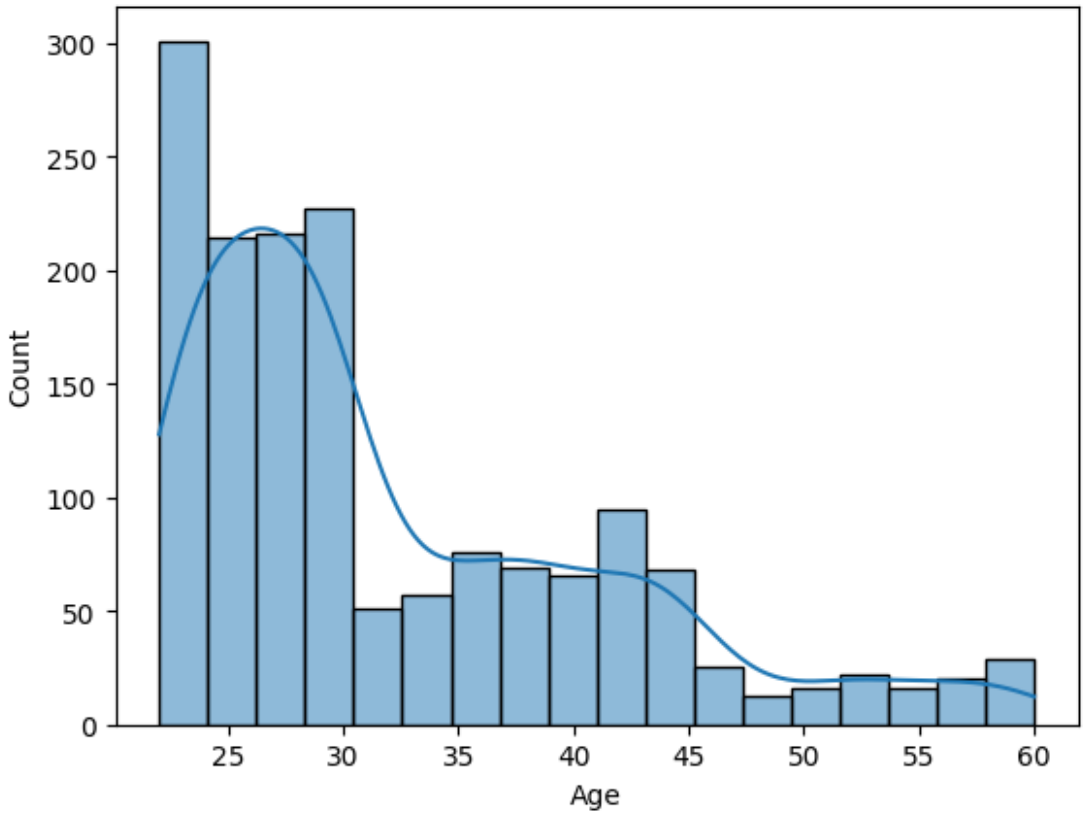
```
array(['Hatchback', 'SUV', 'Sedan'], dtype=object)
```

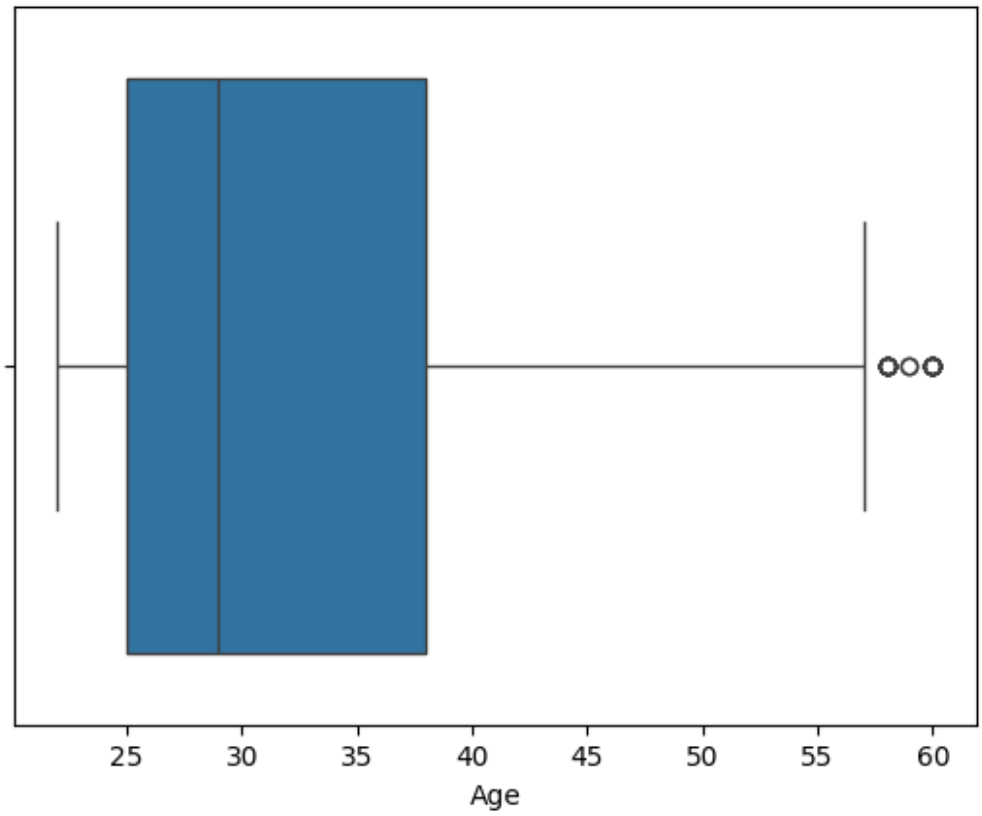
Univariate Analysis

Question: Explore all the variables and provide observations on the distributions of all the relevant variables in the dataset. (10 marks)

```
sns.histplot(data=df,x='Age',kde='True')
plt.show()
sns.boxplot(data=df,x='Age')
plt.show()
print(df['No_of_Dependents'].min())
```

```
print(df['No_of_Dependents'].max())
print(df['No_of_Dependents'].mean())
```



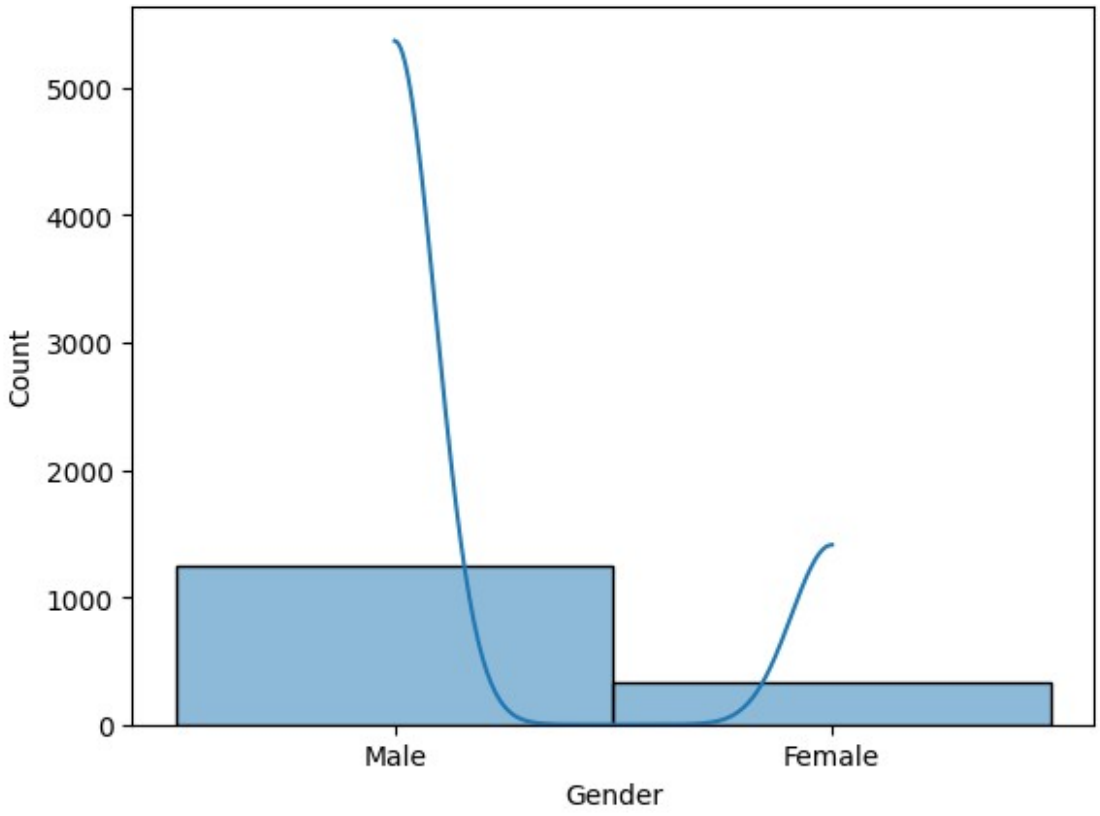



```
0
4
2.4579380139152436
```

###observation:

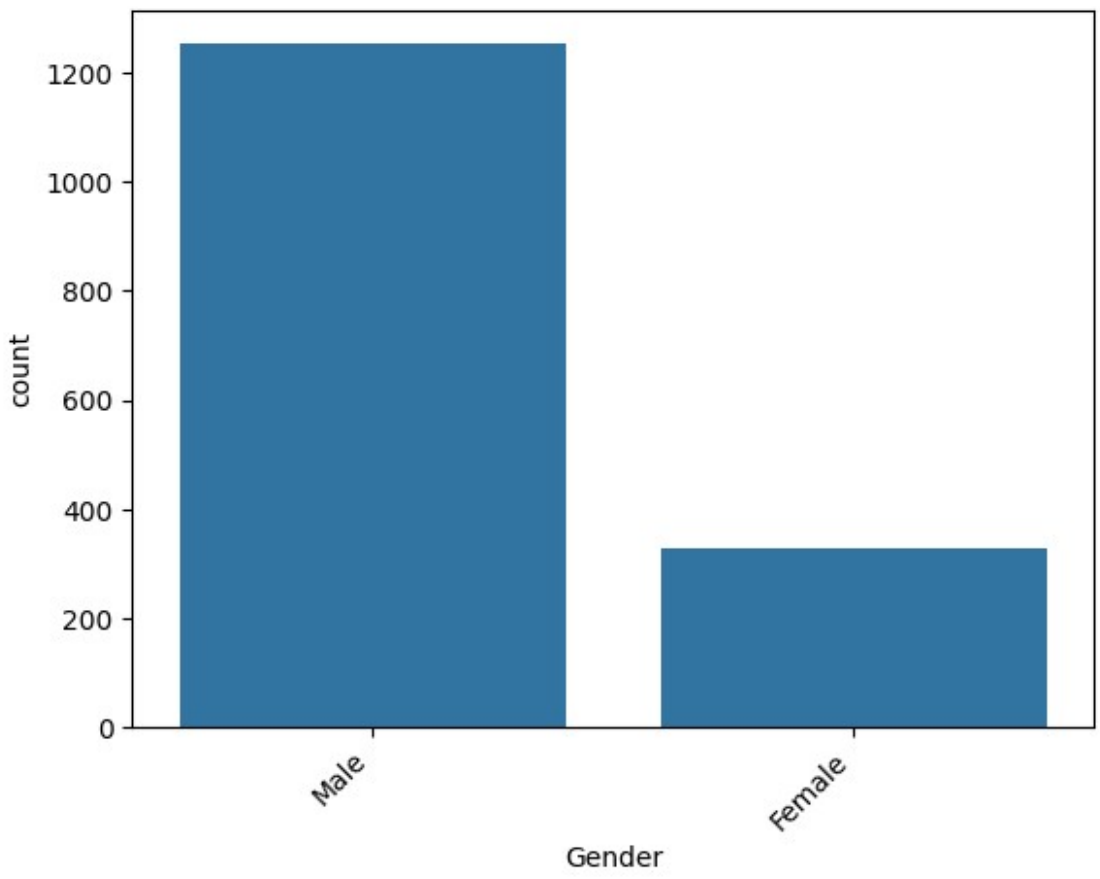
- more than 300 people belong to age group less than 25
- min dep=0
- max dep=4

```
sns.histplot(data=df,x='Gender',kde='True')
plt.show()
chart=sns.countplot(data=df,x='Gender')
chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
```




```
<ipython-input-8-378dc8a8b882>:4: UserWarning: FixedFormatter should
only be used together with FixedLocator
  chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')

[Text(0, 0, 'Male'), Text(1, 0, 'Female')]
```




```
personal_loan_df = df[df['Personal_loan'] == 'Yes']
age_group_counts = personal_loan_df['Age'].value_counts()
min_age_group = age_group_counts.idxmin()

print("Minimum age group of people taking a personal loan:",
min_age_group)
```

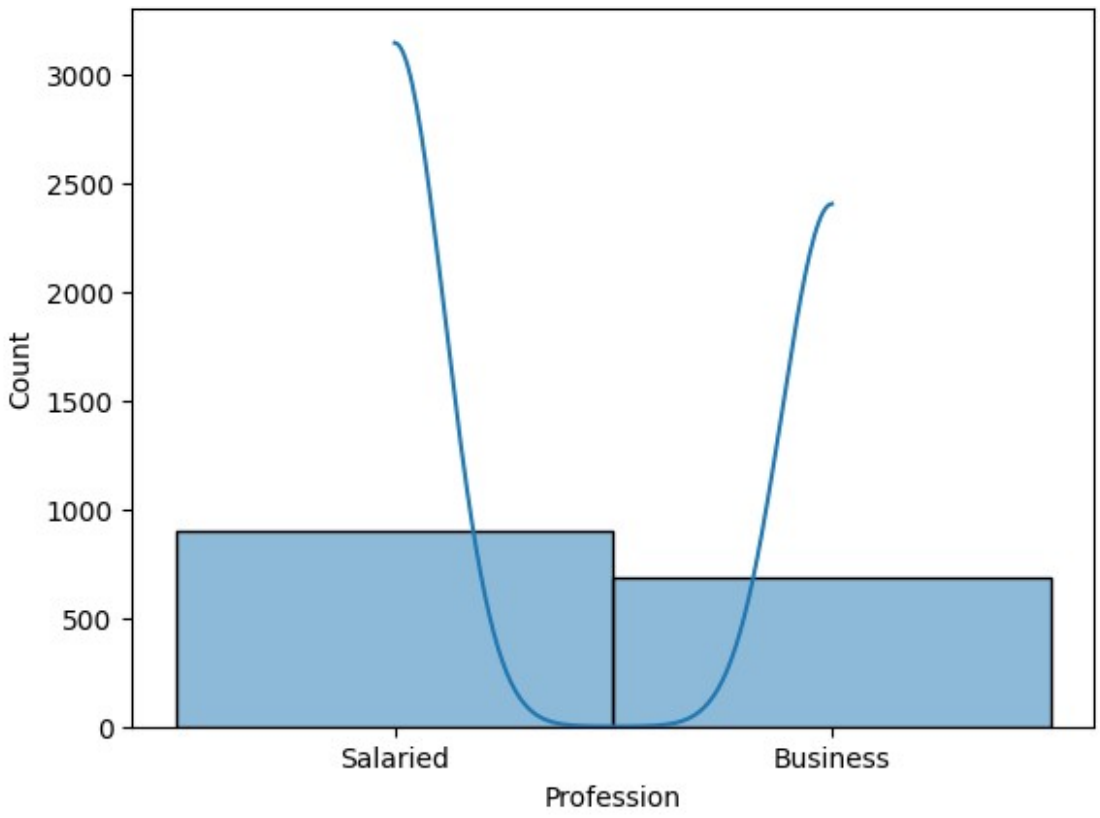
```
Minimum age group of people taking a personal loan: 55
```

###observations: number of male is more than number of females

```
df['Profession'].unique()
```

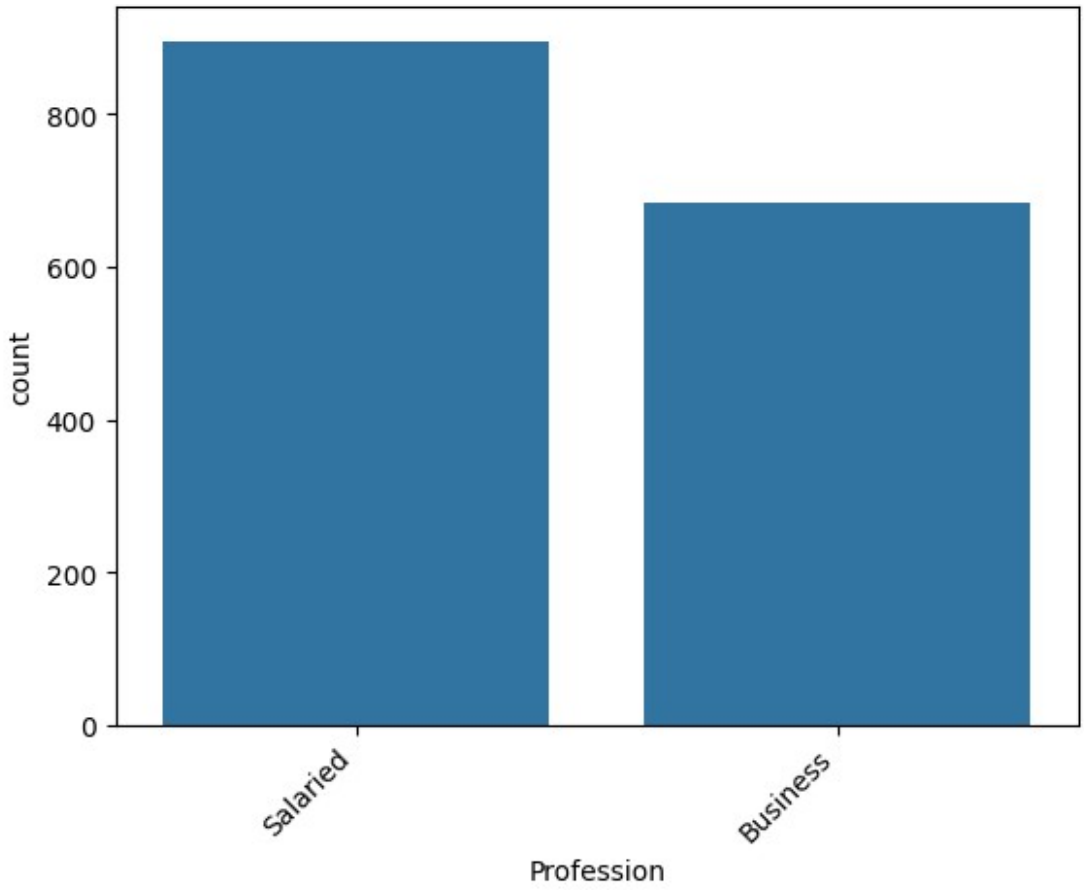
```
array(['Salaried', 'Business'], dtype=object)
```

```
sns.histplot(data=df,x='Profession',kde='True')
plt.show()
chart=sns.countplot(data=df,x='Profession')
chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
```




```
<ipython-input-11-2a40c32ba654>:4: UserWarning: FixedFormatter should
only be used together with FixedLocator
  chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')

[Text(0, 0, 'Salaried'), Text(1, 0, 'Business')]
```

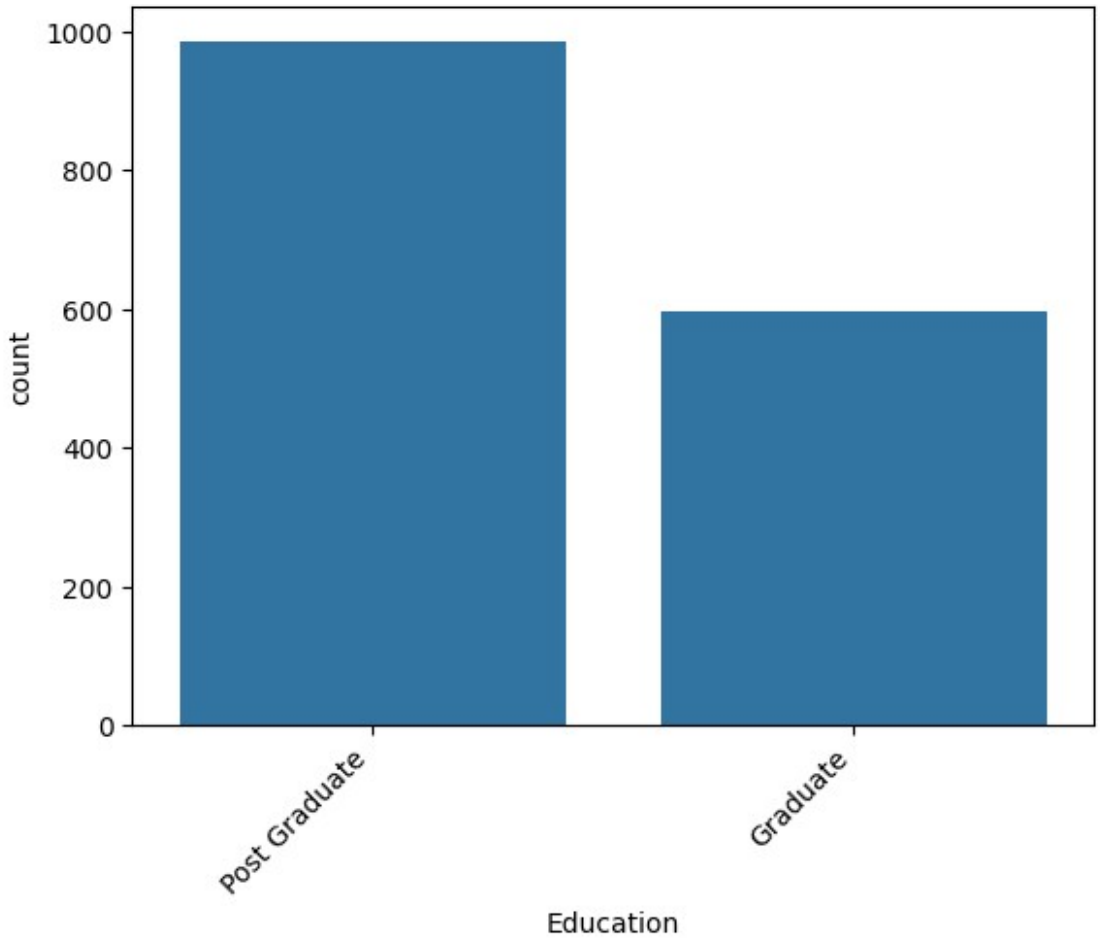


###observation: number of salaried persons are more than business person

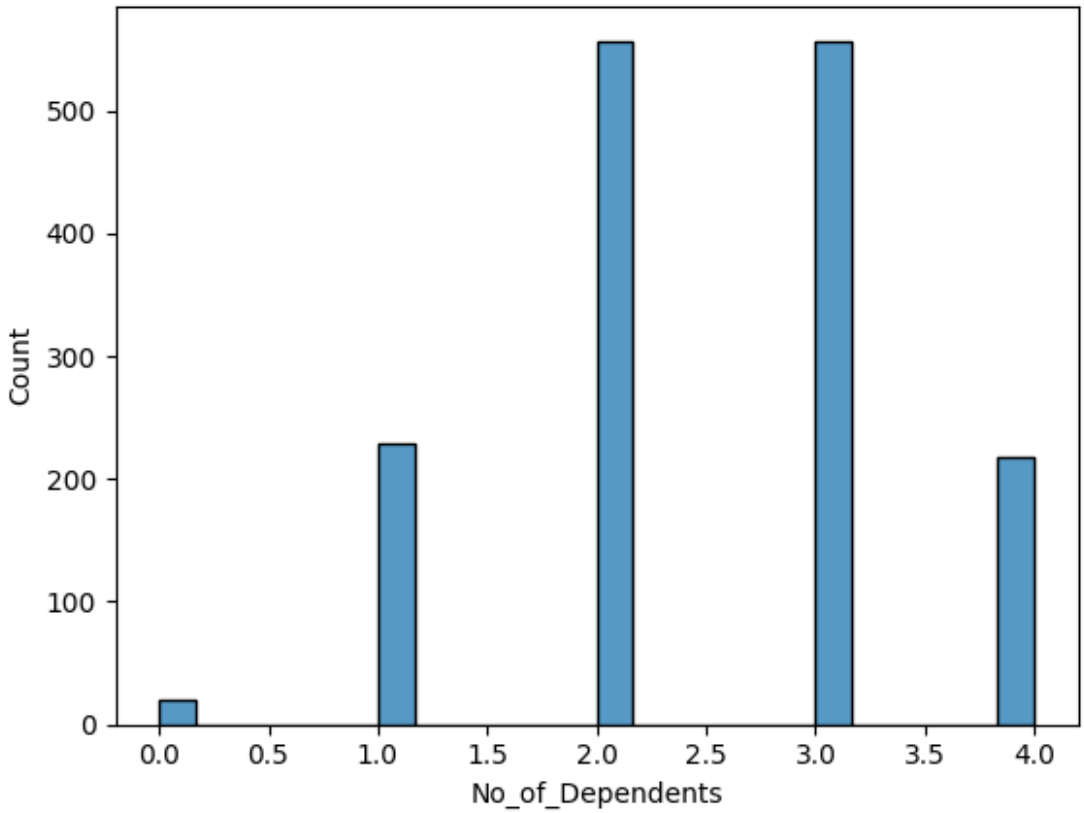
```
chart=sns.countplot(data=df,x='Education')
chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
```

```
<ipython-input-12-da20f3e47816>:2: UserWarning: FixedFormatter should
only be used together with FixedLocator
  chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
```

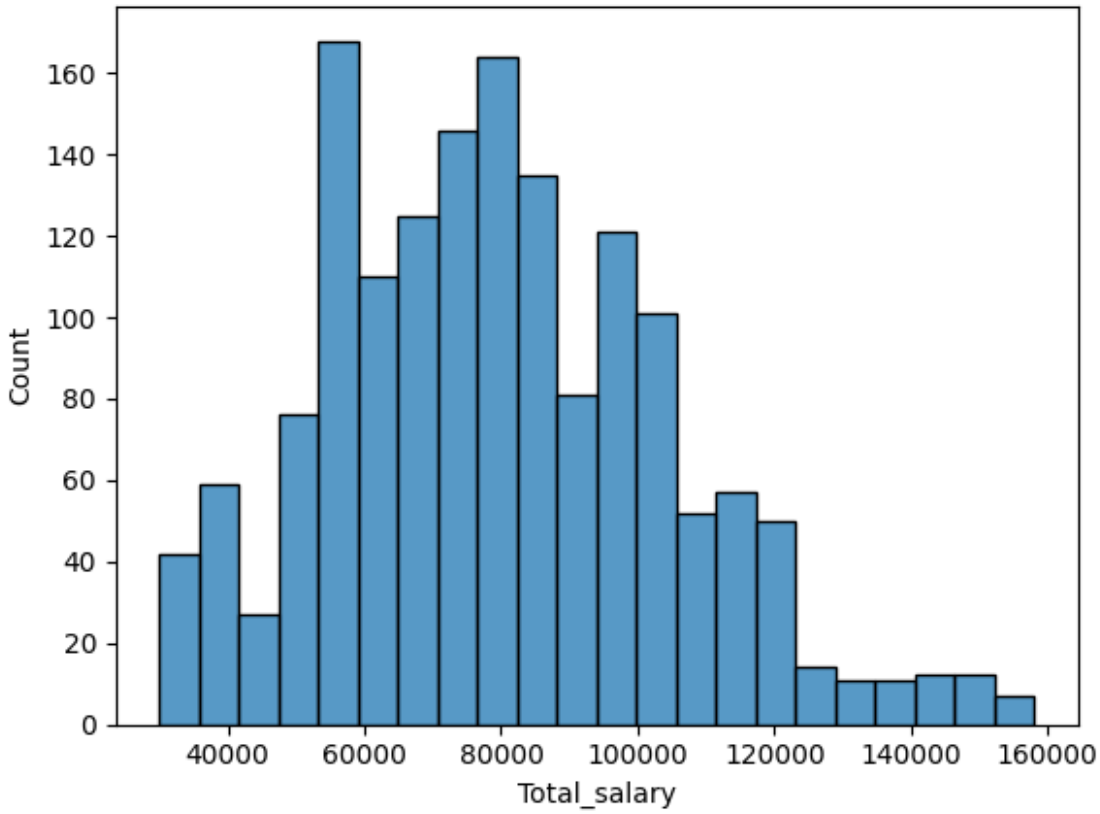
```
[Text(0, 0, 'Post Graduate'), Text(1, 0, 'Graduate')]
```



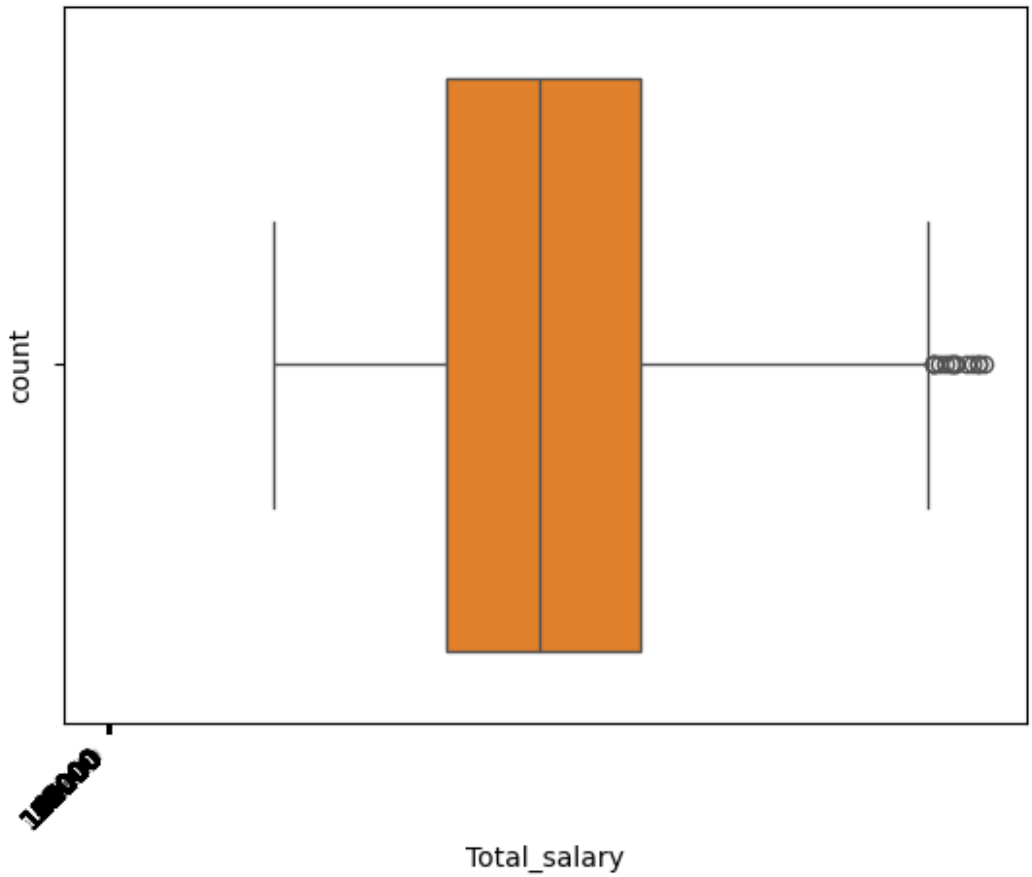

```
sns.histplot(data=df,x='No_of_Dependents')
plt.show()
```




```
sns.histplot(data=df,x='Total_salary')
plt.show()
chart=sns.countplot(data=df,x='Total_salary')
chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
sns.boxplot(data=df,x='Total_salary')
plt.show()
```




```
<ipython-input-14-89a233335d59>:4: UserWarning: FixedFormatter should
only be used together with FixedLocator
  chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
```



bivariate Analysis

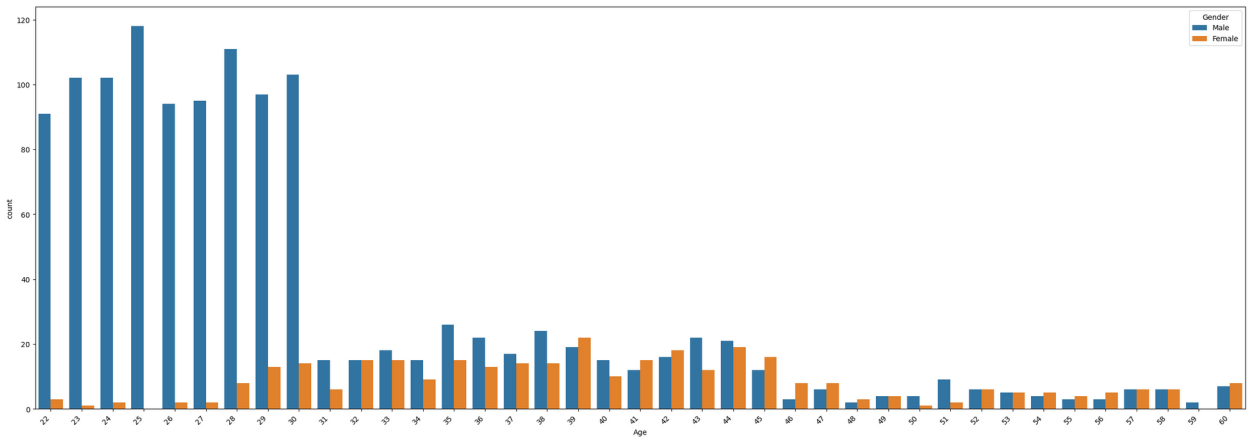
Question: Perform bivariate/multivariate analysis to explore relationships between the important variables in the dataset.

```
plt.figure(figsize=(30,10))
chart=sns.countplot(data=df,x='Age',hue='Gender')
chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
```

```
<ipython-input-15-3789816a73d9>:3: UserWarning: FixedFormatter should
only be used together with FixedLocator
  chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
```

```
[Text(0, 0, '22'),
 Text(1, 0, '23'),
 Text(2, 0, '24'),
 Text(3, 0, '25'),
 Text(4, 0, '26'),
 Text(5, 0, '27'),
 Text(6, 0, '28'),
```

```
Text(7, 0, '29'),
Text(8, 0, '30'),
Text(9, 0, '31'),
Text(10, 0, '32'),
Text(11, 0, '33'),
Text(12, 0, '34'),
Text(13, 0, '35'),
Text(14, 0, '36'),
Text(15, 0, '37'),
Text(16, 0, '38'),
Text(17, 0, '39'),
Text(18, 0, '40'),
Text(19, 0, '41'),
Text(20, 0, '42'),
Text(21, 0, '43'),
Text(22, 0, '44'),
Text(23, 0, '45'),
Text(24, 0, '46'),
Text(25, 0, '47'),
Text(26, 0, '48'),
Text(27, 0, '49'),
Text(28, 0, '50'),
Text(29, 0, '51'),
Text(30, 0, '52'),
Text(31, 0, '53'),
Text(32, 0, '54'),
Text(33, 0, '55'),
Text(34, 0, '56'),
Text(35, 0, '57'),
Text(36, 0, '58'),
Text(37, 0, '59'),
Text(38, 0, '60')]
```

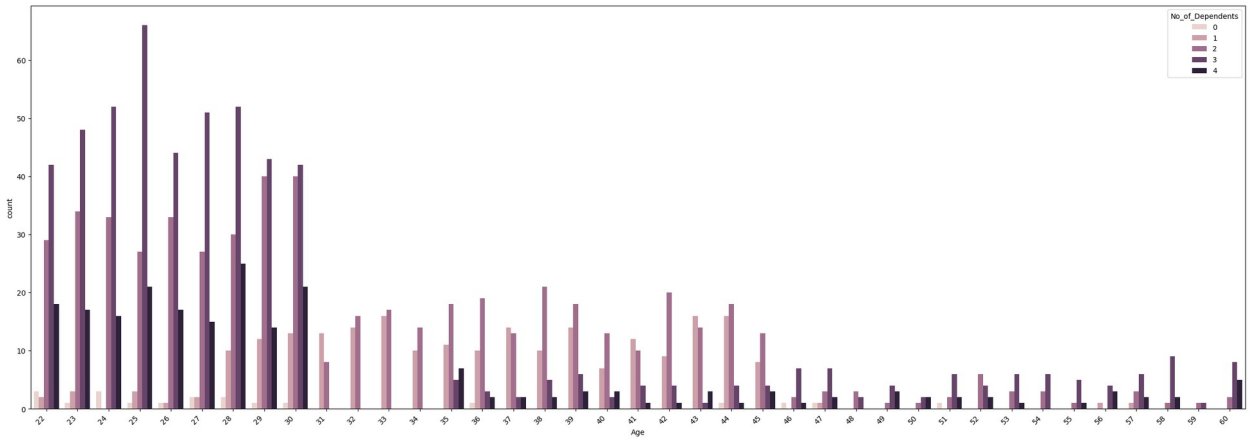


###Observation: count of male age within 30 years is grater than female count

```
plt.figure(figsize=(30,10))
chart=sns.countplot(data=df,x='Age',hue='No_of_Dependents')
chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
```

```
<ipython-input-16-75588947ffc3>:3: UserWarning: FixedFormatter should
only be used together with FixedLocator
  chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
```

```
[Text(0, 0, '22'),
 Text(1, 0, '23'),
 Text(2, 0, '24'),
 Text(3, 0, '25'),
 Text(4, 0, '26'),
 Text(5, 0, '27'),
 Text(6, 0, '28'),
 Text(7, 0, '29'),
 Text(8, 0, '30'),
 Text(9, 0, '31'),
 Text(10, 0, '32'),
 Text(11, 0, '33'),
 Text(12, 0, '34'),
 Text(13, 0, '35'),
 Text(14, 0, '36'),
 Text(15, 0, '37'),
 Text(16, 0, '38'),
 Text(17, 0, '39'),
 Text(18, 0, '40'),
 Text(19, 0, '41'),
 Text(20, 0, '42'),
 Text(21, 0, '43'),
 Text(22, 0, '44'),
 Text(23, 0, '45'),
 Text(24, 0, '46'),
 Text(25, 0, '47'),
 Text(26, 0, '48'),
 Text(27, 0, '49'),
 Text(28, 0, '50'),
 Text(29, 0, '51'),
 Text(30, 0, '52'),
 Text(31, 0, '53'),
 Text(32, 0, '54'),
 Text(33, 0, '55'),
 Text(34, 0, '56'),
 Text(35, 0, '57'),
 Text(36, 0, '58'),
 Text(37, 0, '59'),
 Text(38, 0, '60')]
```



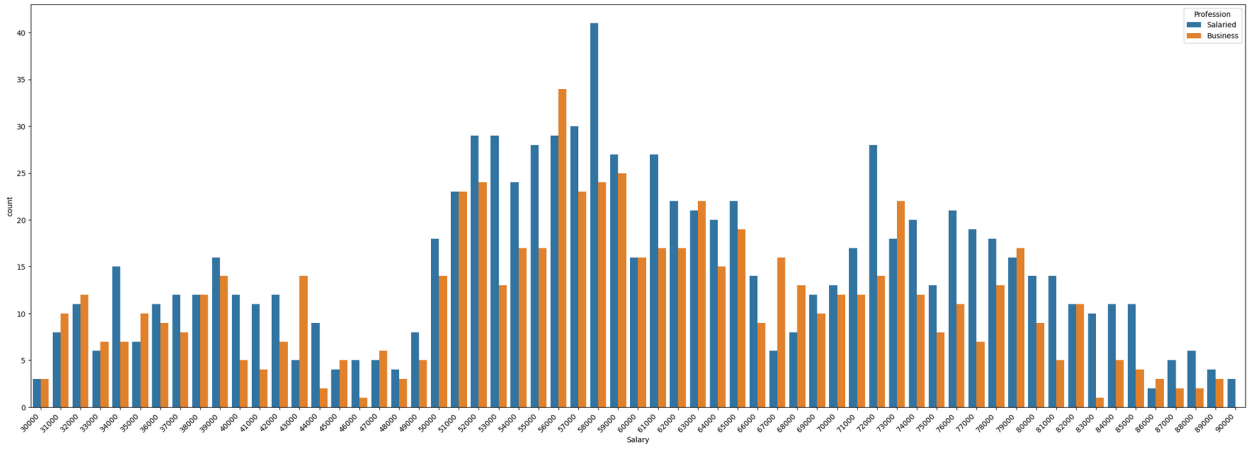
###observation: age group 31 to 34 have 1 or 2 dependents and age group 25 have highest number of dependents

```
plt.figure(figsize=(30,10))
chart=sns.countplot(data=df,x='Salary',hue='Profession')
chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
```

```
<ipython-input-17-dd9e1514d98b>:3: UserWarning: FixedFormatter should
only be used together with FixedLocator
  chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
```

```
[Text(0, 0, '30000'),
 Text(1, 0, '31000'),
 Text(2, 0, '32000'),
 Text(3, 0, '33000'),
 Text(4, 0, '34000'),
 Text(5, 0, '35000'),
 Text(6, 0, '36000'),
 Text(7, 0, '37000'),
 Text(8, 0, '38000'),
 Text(9, 0, '39000'),
 Text(10, 0, '40000'),
 Text(11, 0, '41000'),
 Text(12, 0, '42000'),
 Text(13, 0, '43000'),
 Text(14, 0, '44000'),
 Text(15, 0, '45000'),
 Text(16, 0, '46000'),
 Text(17, 0, '47000'),
 Text(18, 0, '48000'),
 Text(19, 0, '49000'),
 Text(20, 0, '50000'),
 Text(21, 0, '51000'),
 Text(22, 0, '52000'),
```

```
Text(23, 0, '53000'),
Text(24, 0, '54000'),
Text(25, 0, '55000'),
Text(26, 0, '56000'),
Text(27, 0, '57000'),
Text(28, 0, '58000'),
Text(29, 0, '59000'),
Text(30, 0, '60000'),
Text(31, 0, '61000'),
Text(32, 0, '62000'),
Text(33, 0, '63000'),
Text(34, 0, '64000'),
Text(35, 0, '65000'),
Text(36, 0, '66000'),
Text(37, 0, '67000'),
Text(38, 0, '68000'),
Text(39, 0, '69000'),
Text(40, 0, '70000'),
Text(41, 0, '71000'),
Text(42, 0, '72000'),
Text(43, 0, '73000'),
Text(44, 0, '74000'),
Text(45, 0, '75000'),
Text(46, 0, '76000'),
Text(47, 0, '77000'),
Text(48, 0, '78000'),
Text(49, 0, '79000'),
Text(50, 0, '80000'),
Text(51, 0, '81000'),
Text(52, 0, '82000'),
Text(53, 0, '83000'),
Text(54, 0, '84000'),
Text(55, 0, '85000'),
Text(56, 0, '86000'),
Text(57, 0, '87000'),
Text(58, 0, '88000'),
Text(59, 0, '89000'),
Text(60, 0, '90000')]
```



###observation: salaried person

```
df.groupby(['Profession'])
['Personal_loan'].count().sort_values(ascending =
False).reset_index().head(1)
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 1,\n  \"fields\": [\n
{\n      \"column\": \"Profession\",\n      \"properties\": {\n
\"dtype\": \"string\",\n          \"num_unique_values\": 1,\n
\"samples\": [\n          \"Salaried\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Personal_loan\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
null,\n        \"min\": 896,\n        \"max\": 896,\n
\"num_unique_values\": 1,\n        \"samples\": [\n          896\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe"}
```

```
df.groupby(['No_of_Dependents'])['Age'].count().sort_values(ascending
= False).reset_index().head()
```

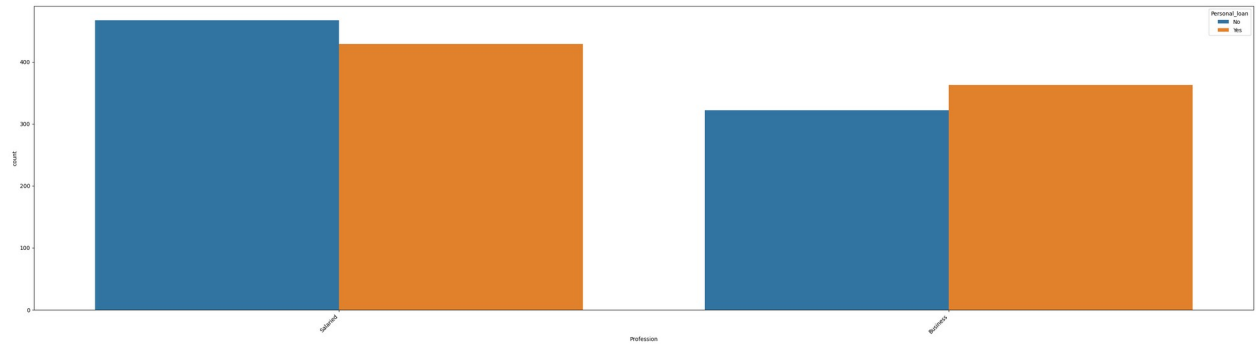
```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 5,\n  \"fields\": [\n
{\n      \"column\": \"No_of_Dependents\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 1,\n        \"min\": 0,\n
\"max\": 4,\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          3,\n          0,\n          1\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Age\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 235,\n        \"min\": 20,\n
\"max\": 557,\n        \"num_unique_values\": 4,\n        \"samples\":
[\n          229,\n          20,\n          557\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe"}
```

```
plt.figure(figsize=( 40,10))
chart=sns.countplot(data=df,x='Profession',hue='Personal_loan')
```

```
chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
```

```
<ipython-input-23-705f981fd546>:3: UserWarning: FixedFormatter should
only be used together with FixedLocator
  chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')

[Text(0, 0, 'Salaried'), Text(1, 0, 'Business')]
```

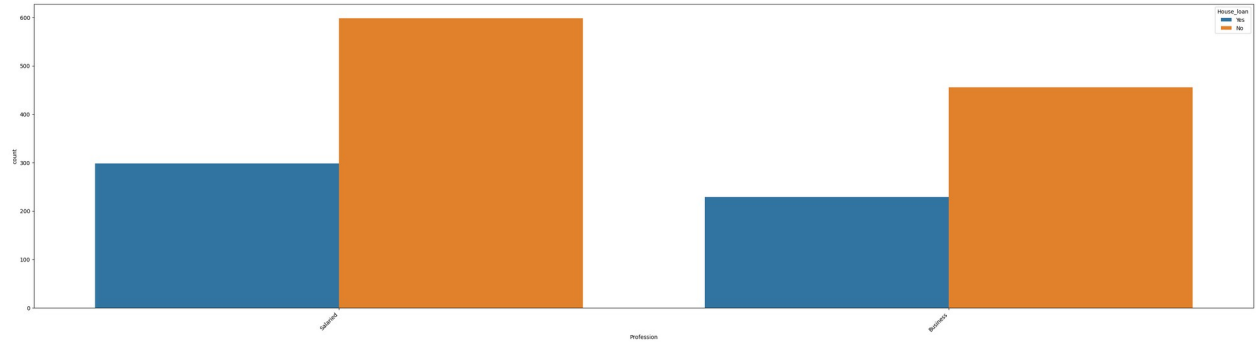


###Observation: more number of salaried person have not taken personal loan

```
plt.figure(figsize=(40,10))
chart=sns.countplot(data=df,x='Profession',hue='House_loan')
chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')
```

```
<ipython-input-24-e8fa15bbf83d>:3: UserWarning: FixedFormatter should
only be used together with FixedLocator
  chart.set_xticklabels(chart.get_xticklabels(), rotation=45,
horizontalalignment='right')

[Text(0, 0, 'Salaried'), Text(1, 0, 'Business')]
```



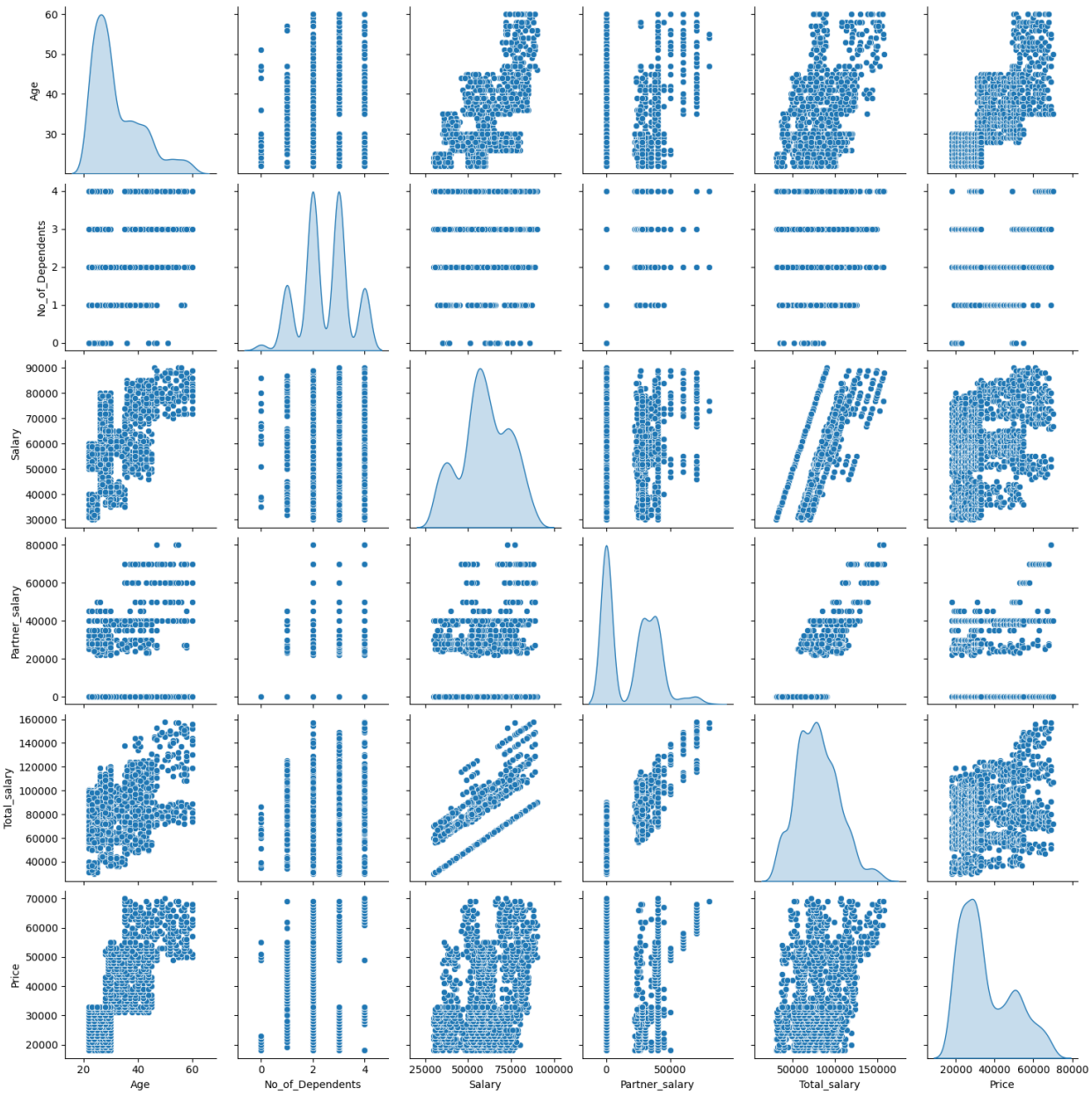

```
df.groupby(['Make'])['Price'].count().sort_values(ascending =
False).reset_index().head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 3,\n  \"fields\": [\n
{\n      \"column\": \"Make\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 3,\n
\"samples\": [\n          \"Hatchback\",\n          \"Sedan\",\n
\"SUV\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Price\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 328,\n        \"min\": 237,\n        \"max\": 884,\n
\"num_unique_values\": 3,\n        \"samples\": [\n          884,\n
460,\n          237\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

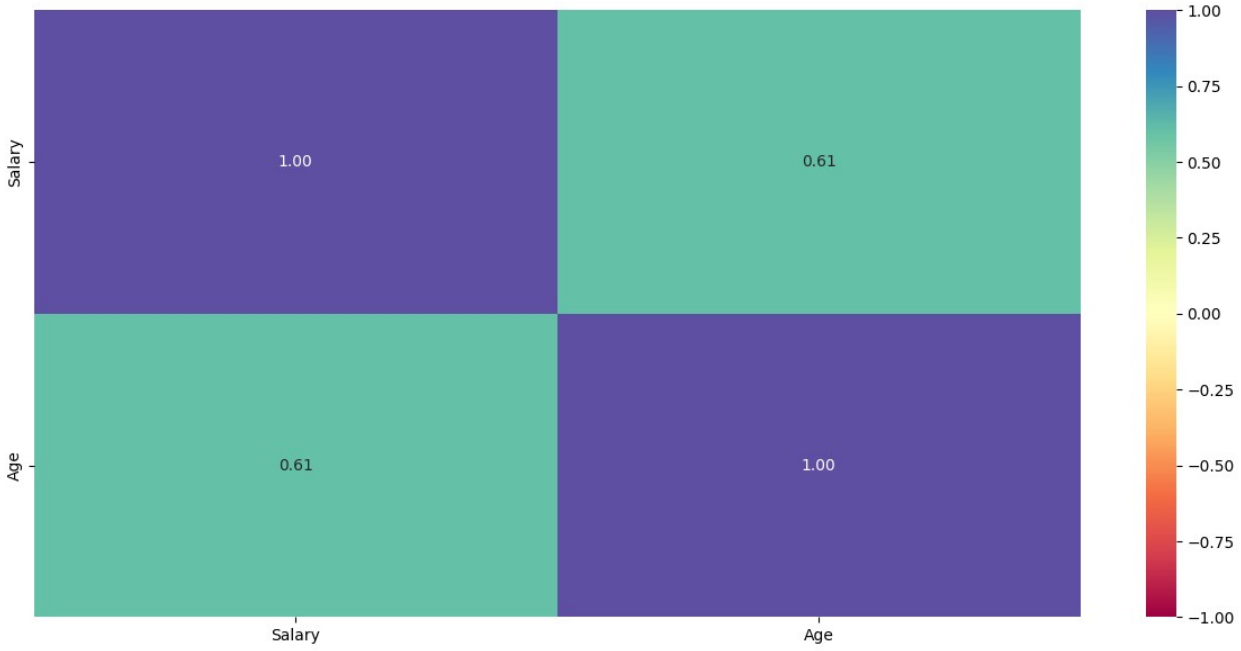

Multivariate Analysis

```
sns.pairplot(df, diag_kind="kde")
```

```
<seaborn.axisgrid.PairGrid at 0x780f998a2290>
```

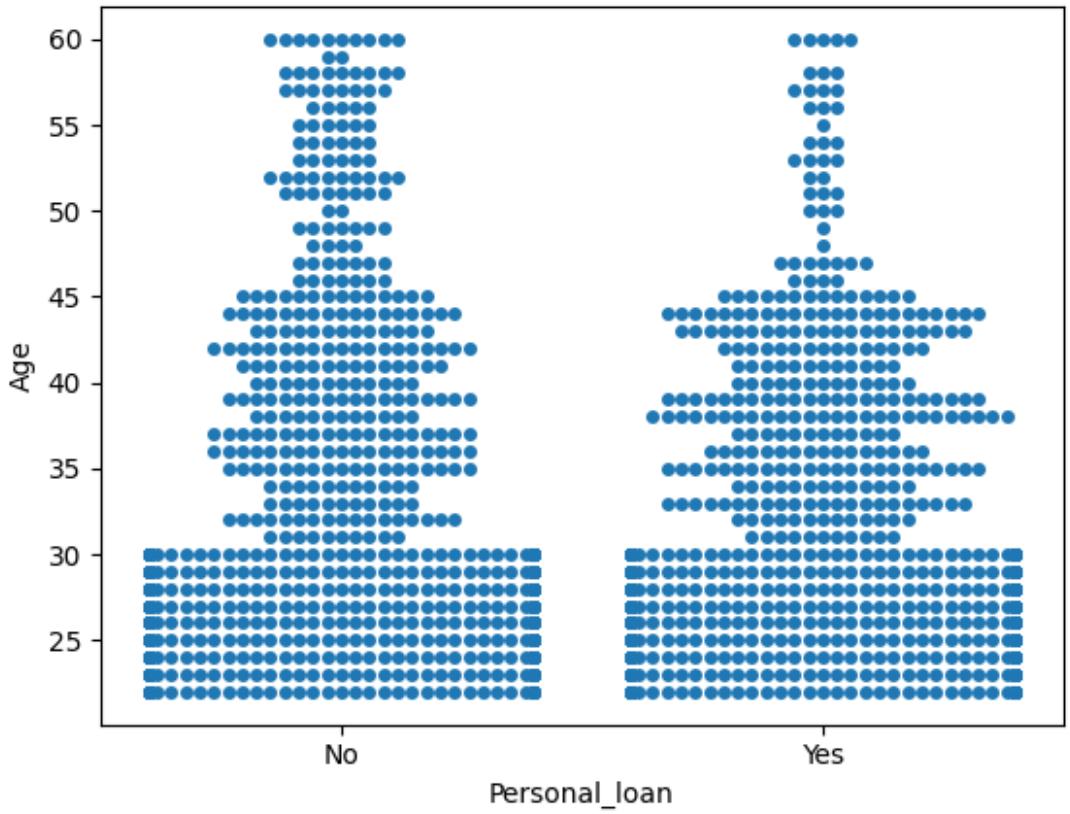



```
lis = ['Salary','Age']
corr = df[lis].corr()
plt.figure(figsize=(15, 7))
sns.heatmap(corr, annot=True, vmin=-1, vmax=1, fmt=".2f",
cmap="Spectral")
plt.show()
```




```
sns.swarmplot(x="Personal_loan", y="Age", data=df);
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 6.2% of the points cannot be placed; you may want to
decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 6.9% of the points cannot be placed; you may want to
decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 28.4% of the points cannot be placed; you may want to
decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.10/dist-packages/seaborn/categorical.py:3398:
UserWarning: 31.3% of the points cannot be placed; you may want to
decrease the size of the markers or use stripplot.
  warnings.warn(msg, UserWarning)
```



###Obsevations:

- number of salaried person in more than Business
- there are Hatchback 884, Sedan 460, SUV 237
- age group 21 to 25 have more 4 number of dependents
- 557 people have 2 and 557 people have 3 dependents

###Recamendation

- production of more Hatchback increases sales
- there is more chance that a salaried person taking house loan than a buiseness man