

SUMMER BOOTCAMP PROJECT 2024

MRIDUL HEMRAJANI

2022511920

INDEX

S. No	Topic
1	List of Tables
2	List of Figures
3	Problem Statement
4	Necessary Libraries
5	Loading Dataset
6	Basic Exploration
7	Descriptive Statistics
8	Data Distribution
9	Correlation Analysis
10	Salary Analysis
11	Loan Status
12	Marital Status and Dependents
13	Partner Employment
14	Salary Comparison
15	House Loan Analysis
16	Salary Distribution
17	Automobile Make Analysis
18	Price Analysis
19	Marital status and loans
20	Educational Qualification Impact
21	Dependent Count Analysis
22	Gender and Salary
23	Loan Status Impact
24	Partner's Salary Contribution
25	Total Salary Distribution

LIST OF TABLES

Table Number	Cell Number
Table 1	3
Table 2	4
Table 3	10
Table 4	29
Table 5	31
Table 6	34
Table 7	37
Table 8	39
Table 9	41
Table 10	43
Table 11	45
Table 12	46
Table 13	48
Table 14	51
Table 15	55
Table 16	57
Table 17	59
Table 18	61
Table 19	63

LIST OF FIGURES

Figure Number	Cell Number
Figure 1	6
Figure 2	9
Figure 3	11
Figure 4	13
Figure 5	16
Figure 6	20
Figure 7	23
Figure 8	28
Figure 9	30
Figure 10	35
Figure 11	38
Figure 12	40
Figure 13	42
Figure 14	14
Figure 15	47
Figure 16	49
Figure 17	52
Figure 18	54
Figure 19	56
Figure 20	58
Figure 21	60
Figure 22	62
Figure 23	64
Figure 24	65

**PROBLEM STATEMENT

Bright Motor Company want to analyze the data to get a fair idea about the demand of customers which will help them in enhancing their customer experience. Suppose you are a Data Scientist at the company and the Data Science team has shared some of the key questions that need to be answered. Perform the data analysis to find answers to these questions that will help the company to improve the business.

IMPORTING THE NECESSARY LIBRARIES

```
In [10]: import pandas as pd  
import numpy as np
```

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

LOADING THE DATASET

```
In [11]: data=pd.read_csv(r'D:\Machine_Learning\Bootcamp ML\Datasets\5-bright_automotive_
```

BASIC EXPLORATION

1. FIRST 5 ROWS

```
In [12]: data.head()
```

	Age	Gender	Profession	Marital_status	Education	No_of_Dependents	Personal_loan
0	53	Male	Business	Married	Post Graduate	4	N
1	53	Femal	Salaried	Married	Post Graduate	4	Ye
2	53	Female	Salaried	Married	Post Graduate	3	N
3	53	Female	Salaried	Married	Graduate	?	Ye
4	53	Male	NaN	Married	Post Graduate	3	N

Table 1

	0	1	2	3	4
Age	53	53	53	53	53
Gender	Male	Femal	Female	Female	Male
Profession	Business	Salaried	Salaried	Salaried	NaN
Marital_status	Married	Married	Married	Married	Married
Education	Post Graduate	Post Graduate	Post Graduate	Graduate	Post Graduate
No_of_Dependents	4	4	3	?	3
Personal_loan	No	Yes	No	Yes	No
House_loan	No	No	No	No	No
Partner_working	Yes	Yes	Yes	Yes	Yes
Salary	99300.0	95500.0	97300.0	72500.0	79700.0
Partner_salary	70700.0	70300.0	60700.0	70300.0	60200.0
Total_salary	170000	165800	158000	142800	139900
Price	61000	61000	57000	61000	57000
Make	SUV	SUV	SUV	?	SUV

OBSERVATION

'No_of_Dependants' and 'Make' has value '?'. Need to check that.

'Profession' is NULL at one place.

2. LAST 5 ROWS

In [13]: `data.tail()`

	Age	Gender	Profession	Marital_status	Education	No_of_Dependents	Personal_loan
1576	22	Male	Salaried	Single	Graduate	2	
1577	22	Male	Business	Married	Graduate	4	
1578	22	Male	Business	Single	Graduate	2	
1579	22	Male	Business	Married	Graduate	3	
1580	22	Male	Salaried	Married	Graduate	4	

Table 2

	1576	1577	1578	1579	1580
Age	22	22	22	22	22
Gender	Male	Male	Male	Male	Male
Profession	Salaried	Business	Business	Business	Salaried
Marital_status	Single	Married	Single	Married	Married
Education	Graduate	Graduate	Graduate	Graduate	Graduate
No_of_Dependents	2	4	2	3	4
Personal_loan	No	No	No	Yes	No
House_loan	Yes	No	Yes	Yes	No
Partner_working	No	No	No	No	No
Salary	33300.0	32000.0	32900.0	32200.0	31600.0
Partner_salary	0.0	NaN	0.0	NaN	0.0
Total_salary	33300	32000	32900	32200	31600
Price	27000	31000	30000	24000	31000
Make	Hatchback	Hatchback	Hatchback	Hatchback	Hatchback

OBSERVATIONS

'Partner_Salary' is NULL at places.

3. Shape

In [14]: `data.shape`

Out[14]: (1581, 14)

There are 1581 rows and 14 columns in the given dataset.

4.Datatypes

In [15]: `data.dtypes`

```
Out[15]: Age           int64
          Gender        object
          Profession    object
          Marital_status object
          Education     object
          No_of_Dependents object
          Personal_loan  object
          House_loan     object
          Partner_working object
          Salary         float64
          Partner_salary float64
          Total_salary   int64
          Price          int64
          Make           object
          dtype: object
```

Figure 1

```
Age           int64
Gender        object
Profession    object
Marital_status object
Education     object
No_of_Dependents object
Personal_loan  object
House_loan     object
Partner_working object
Salary         float64
Partner_salary float64
Total_salary   int64
Price          int64
Make           object
dtype: object
```

OBSERVATIONS

Data Types of 'Total_Salary' and 'Price' is int, it should be float.

Data Type of 'No_Of_Dependants' is object.

In [16]: `data['No_of_Dependents']=data['No_of_Dependents'].replace('?', np.nan)`

```
In [17]: data['Total_salary']=data['Total_salary'].astype('float')
data['Price']=data['Price'].astype('float')
```

```
In [18]: data.info()
```

```
<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            1528 non-null    object  
 2   Profession        1575 non-null    object  
 3   Marital_status    1581 non-null    object  
 4   Education          1581 non-null    object  
 5   No_of_Dependents  1579 non-null    object  
 6   Personal_loan      1581 non-null    object  
 7   House_loan          1581 non-null    object  
 8   Partner_working    1581 non-null    object  
 9   Salary             1568 non-null    float64 
 10  Partner_salary     1475 non-null    float64 
 11  Total_salary       1581 non-null    float64 
 12  Price              1581 non-null    float64 
 13  Make               1581 non-null    object  
dtypes: float64(4), int64(1), object(9)
memory usage: 173.1+ KB
```

Figure 2

```
<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            1528 non-null    object  
 2   Profession        1575 non-null    object  
 3   Marital_status    1581 non-null    object  
 4   Education          1581 non-null    object  
 5   No_of_Dependents  1579 non-null    object  
 6   Personal_loan      1581 non-null    object  
 7   House_loan          1581 non-null    object  
 8   Partner_working    1581 non-null    object  
 9   Salary             1568 non-null    float64 
 10  Partner_salary     1475 non-null    float64 
 11  Total_salary       1581 non-null    float64 
 12  Price              1581 non-null    float64 
 13  Make               1581 non-null    object  
dtypes: float64(4), int64(1), object(9)
memory usage: 173.1+ KB
```

Observations

NULL values in several columns.

5. Statistical Summary

In [19]: `data.describe()`

Out[19]:

	Age	Salary	Partner_salary	Total_salary	Price
count	1581.000000	1568.000000	1475.000000	1581.000000	1581.000000
mean	31.952562	60276.913265	20225.559322	79625.996205	35948.170778
std	8.712549	14636.200199	19573.149277	25545.857768	21175.212108
min	14.000000	30000.000000	0.000000	30000.000000	58.000000
25%	25.000000	51900.000000	0.000000	60500.000000	25000.000000
50%	29.000000	59450.000000	25600.000000	78000.000000	31000.000000
75%	38.000000	71700.000000	38300.000000	95900.000000	47000.000000
max	120.000000	99300.000000	80500.000000	171000.000000	680000.000000

Table 3

	Age	Salary	Partner_salary	Total_salary	Price
count	1581.000000	1568.000000	1475.000000	1581.000000	1581.000000
mean	31.952562	60276.913265	20225.559322	79625.996205	35948.170778
std	8.712549	14636.200199	19573.149277	25545.857768	21175.212108
min	14.000000	30000.000000	0.000000	30000.000000	58.000000
25%	25.000000	51900.000000	0.000000	60500.000000	25000.000000
50%	29.000000	59450.000000	25600.000000	78000.000000	31000.000000
75%	38.000000	71700.000000	38300.000000	95900.000000	47000.000000
max	120.000000	99300.000000	80500.000000	171000.000000	680000.000000

6. NULL Values

In [20]: `data.isnull().sum()`

```
Out[20]: Age          0
          Gender       53
          Profession    6
          Marital_status 0
          Education      0
          No_of_Dependents 2
          Personal_loan    0
          House_loan       0
          Partner_working   0
          Salary          13
          Partner_salary    106
          Total_salary      0
          Price            0
          Make             0
          dtype: int64
```

Figure 3

Age	0
Gender	53
Profession	6
Marital_status	0
Education	0
No_of_Dependents	0
Personal_loan	0
House_loan	0
Partner_working	0
Salary	13
Partner_salary	106
Total_salary	0
Price	0
Make	0
dtype: int64	

7. Duplicated Values

```
In [21]: data.duplicated().sum()
```

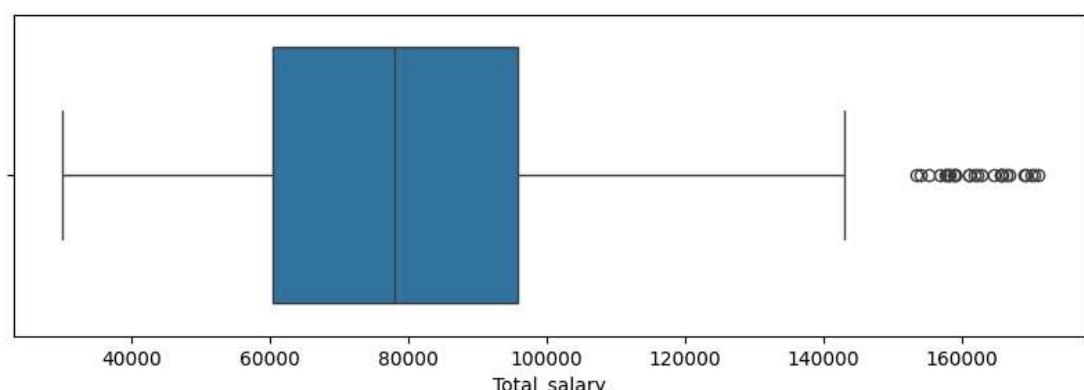
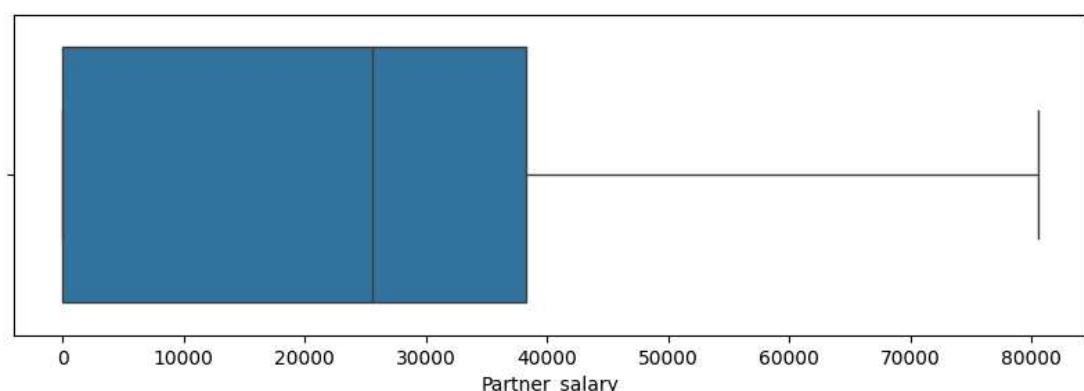
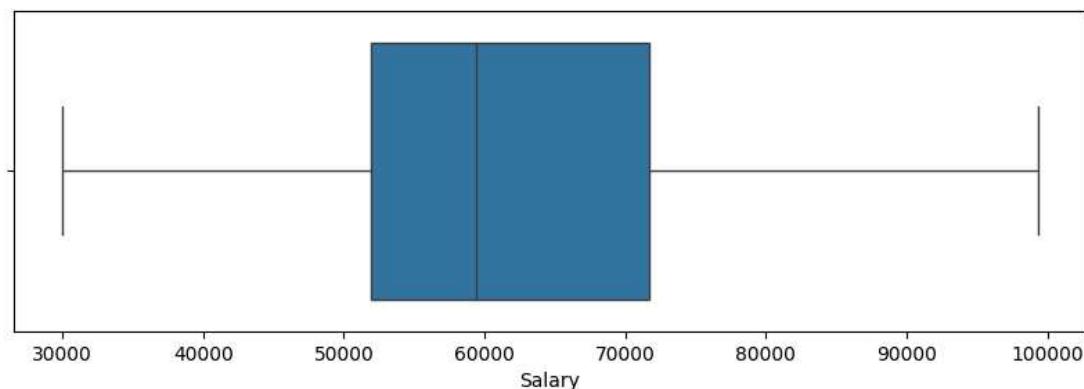
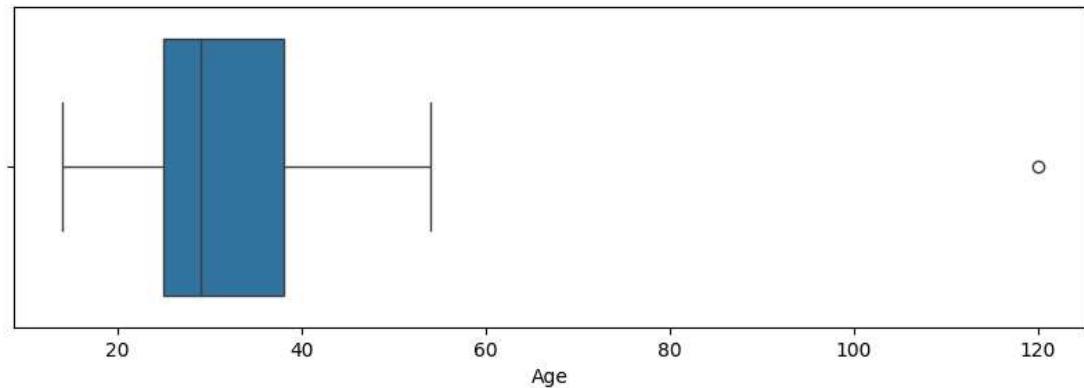
```
Out[21]: 0
```

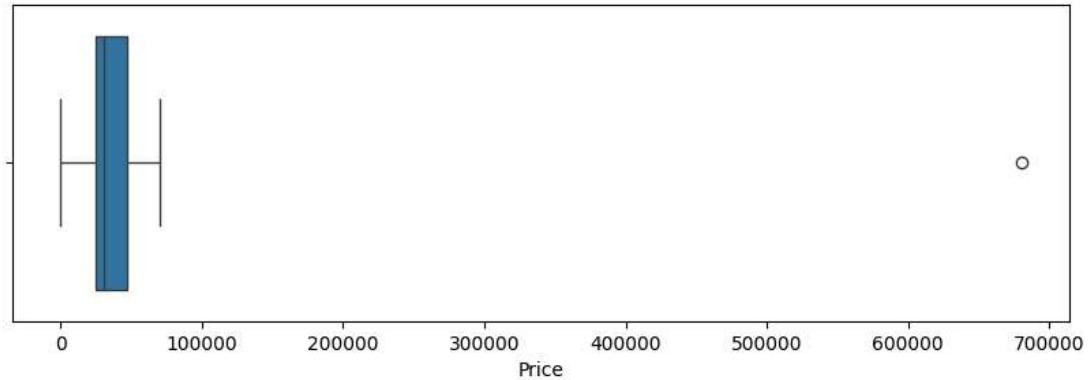
0 duplicated values in the dataset.

8. Outliers

```
In [ ]: for i in ['Age', 'Salary', 'Partner_salary', 'Total_salary', 'Price']:
    plt.figure(figsize=(10,3))
    sns.boxplot(data=data, x=i)
```

Figure 4





9. Data Cleaning

Filling NULL values in numerical columns with medians due to the presence of outliers.

```
In [ ]: data['No_of_Dependents']=data['No_of_Dependents'].replace('?', '9')
data['No_of_Dependents']=data['No_of_Dependents'].replace(np.nan, '9')
data['No_of_Dependents']=data['No_of_Dependents'].astype('int')
data['No_of_Dependents']=data['No_of_Dependents'].replace('9', np.nan)

In [ ]: value={'Age':data['Age'].median(),'Salary':data['Salary'].median(),'Partner_salary':data['Partner_salary'].median()}
data.fillna(value=value, inplace=True)

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

Figure 5

Age	0
Gender	53
Profession	6
Marital_status	0
Education	0
No_of_Dependents	0
Personal_loan	0
House_loan	0
Partner_working	0
Salary	0
Partner_salary	0
Total_salary	0
Price	0
Make	0
dtype:	int64

Treating anomalies present in the 'Gender','No_of_Dependents' and 'Make' columns of the dataset and dealing with their NULL values.

```
In [ ]: data['Gender'].unique()

In [ ]: data['Gender']=data['Gender'].replace('Femal','Female')
        data['Gender']=data['Gender'].replace('Femle','Female')
        data['Make']=data['Make'].replace('?',np.nan)

In [ ]: values={'Gender':data['Gender'].mode().values[0],'Profession':data['Profession']}
        data.fillna(value=values,inplace=True)

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

Figure 6

Age	0
Gender	0
Profession	0
Marital_status	0
Education	0
No_of_Dependents	0
Personal_loan	0
House_loan	0
Partner_working	0
Salary	0
Partner_salary	0
Total_salary	0
Price	0
Make	0
dtype:	int64

Dealing with outliers present in the data.

```
In [ ]: def remove_outliers(col):
            sorted(col)
            Q1,Q3=col.quantile([0.25,0.75])
            IQR=Q3-Q1
            lower_bound=Q1-1.5*IQR
            upper_bound=Q3+1.5*IQR
            return lower_bound,upper_bound

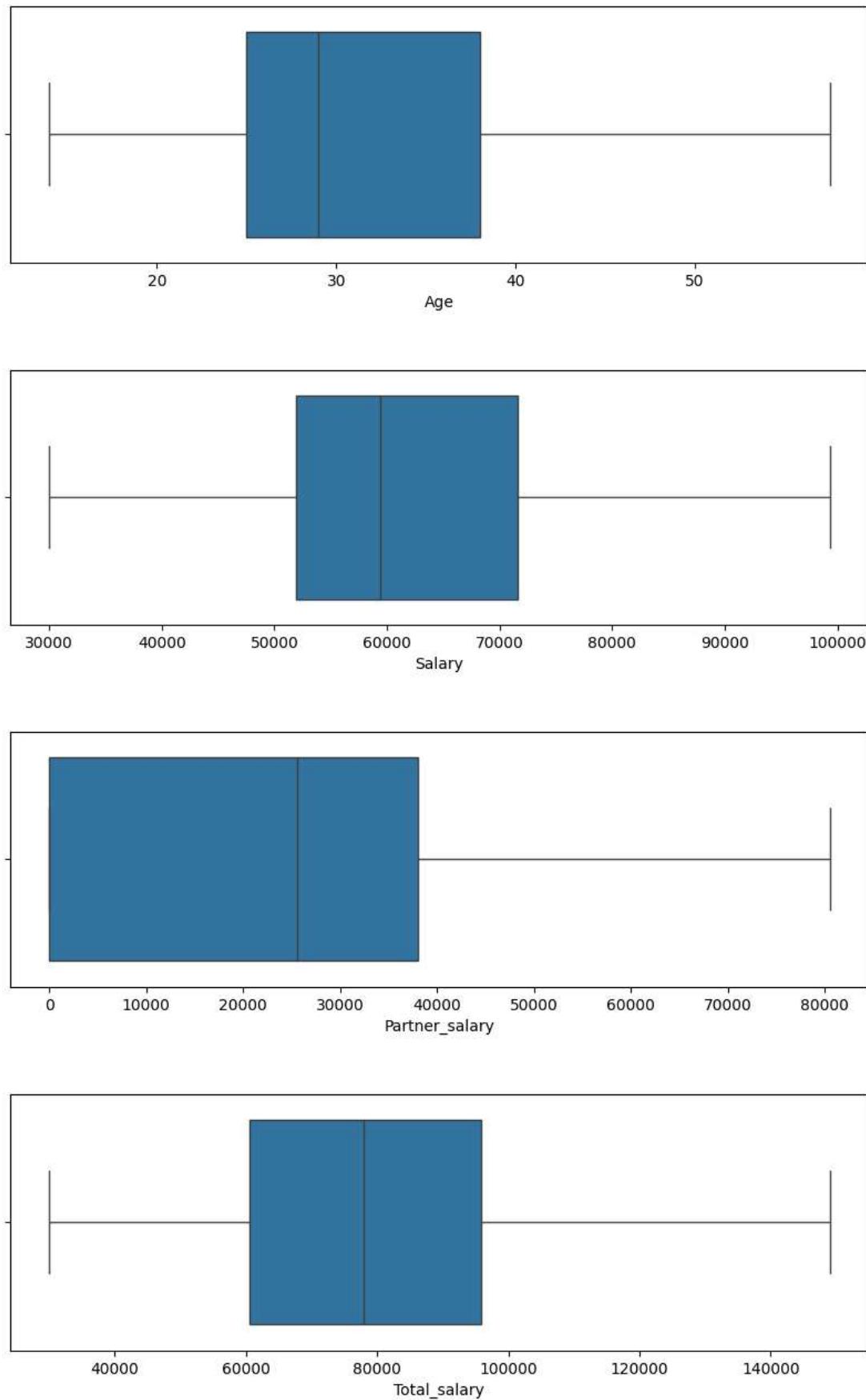
In [ ]: llts,ults=remove_outliers(data['Total_salary'])
        data['Total_salary']=np.where(data['Total_salary']>ults,ults,data['Total_salary'])
        data['Total_salary']=np.where(data['Total_salary']<llts,llts,data['Total_salary'])

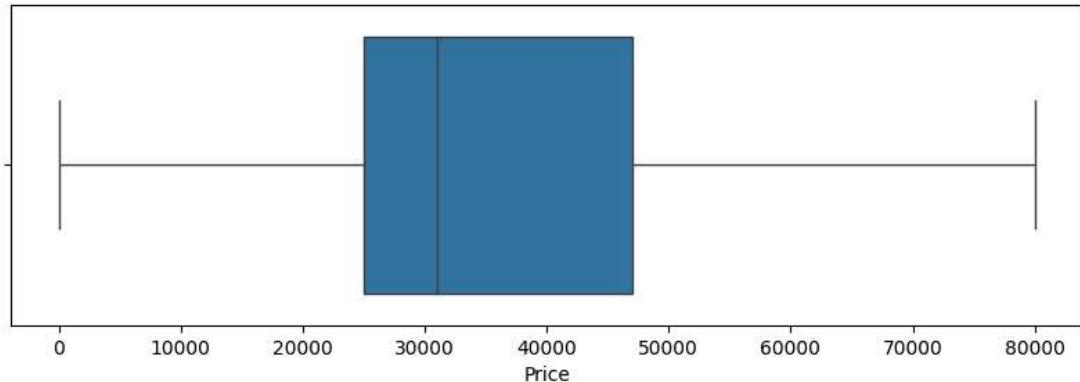
        llp,ulp=remove_outliers(data['Price'])
        data['Price']=np.where(data['Price']>ulp,ulp,data['Price'])
        data['Price']=np.where(data['Price']<llp,llp,data['Price'])

        lla,ula=remove_outliers(data['Age'])
        data['Age']=np.where(data['Age']>ula,ula,data['Age'])
        data['Age']=np.where(data['Age']<lla,lla,data['Age'])
```

```
In [ ]: for i in ['Age', 'Salary', 'Partner_salary', 'Total_salary', 'Price']:
    plt.figure(figsize=(10,3))
    sns.boxplot(data=data, x=i)
```

Figure 7





Descriptive Statistics

- o What are the mean, median, and standard deviation of the ages of individuals in the dataset?

```
In [ ]: data['Age'].mean()
```

The mean is 31.91302972802024

```
In [ ]: data['Age'].median()
```

The median is 29.0

```
In [ ]: data['Age'].std()
```

The standard deviation is 8.450649424059444

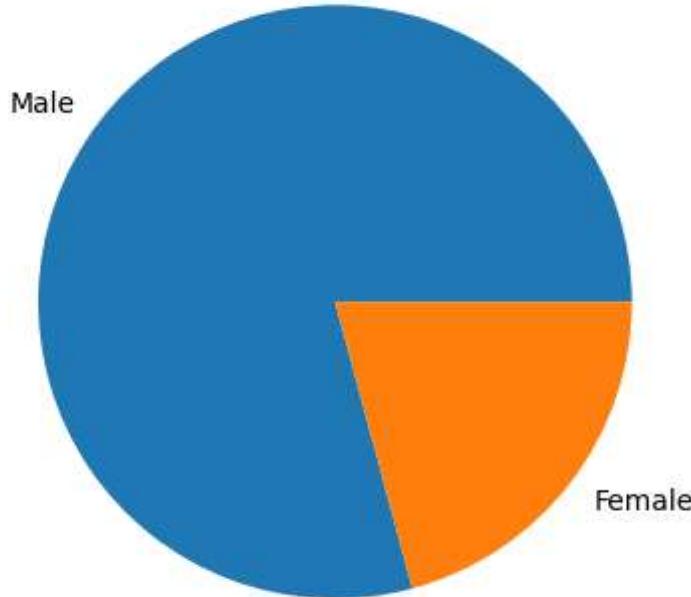
• Data Distribution

- o What is the distribution of gender in the dataset? Represent it using a pie chart.

```
In [ ]: data['Gender'].value_counts()
```

```
In [ ]: plt.pie(data=data,x=data['Gender'].value_counts().values,labels=data['Gender'].v
```

Figure 8



Correlation Analysis

- Is there a correlation between age and salary? Provide the correlation coefficient and interpret the result.

```
In [ ]: data[['Age', 'Salary']].corr()
```

Table 4

	Age	Salary
Age	1.000000	0.599922
Salary	0.599922	1.000000

The correlation coefficient is 0.599922.

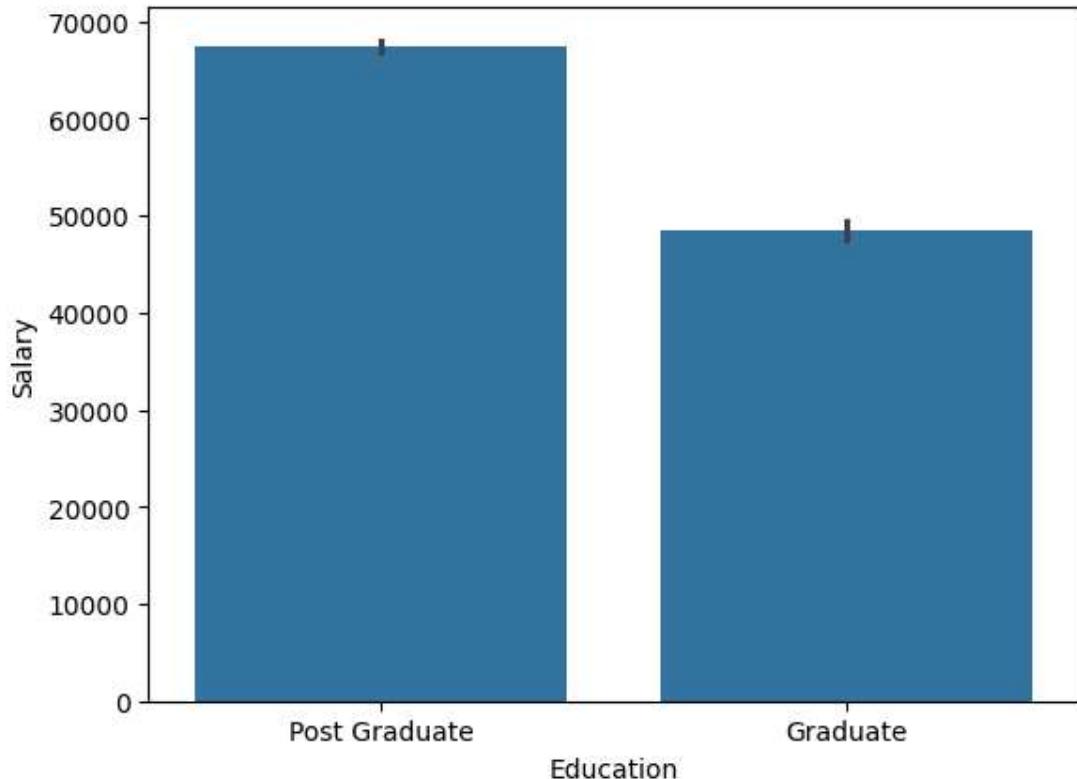
This means that Age and Salary are not related to each other significantly.

Salary Analysis

- What is the average salary for individuals based on their educational qualifications (Graduate vs. Post Graduate)?

```
In [ ]: sns.barplot(data=data,x=data['Education'],y=data['Salary'])
```

Figure 9



```
In [ ]: data.groupby('Education')['Salary'].mean()
```

Table 5

```
Education
Graduate      48514.597315
Post Graduate 67383.096447
Name: Salary, dtype: float64
```

The average salary of Post Graduates is higher than that of Graduates.

Loan Status

- o What percentage of individuals have taken a personal loan? How does this compare between males and females?

```
In [ ]: data['Personal_loan'].unique()
```

```
In [ ]: data['Personal_loan'][data['Personal_loan']=='Yes'].value_counts().sum()/len(data)
```

The percentage of people who have taken a personal loan is 50.094876660341555

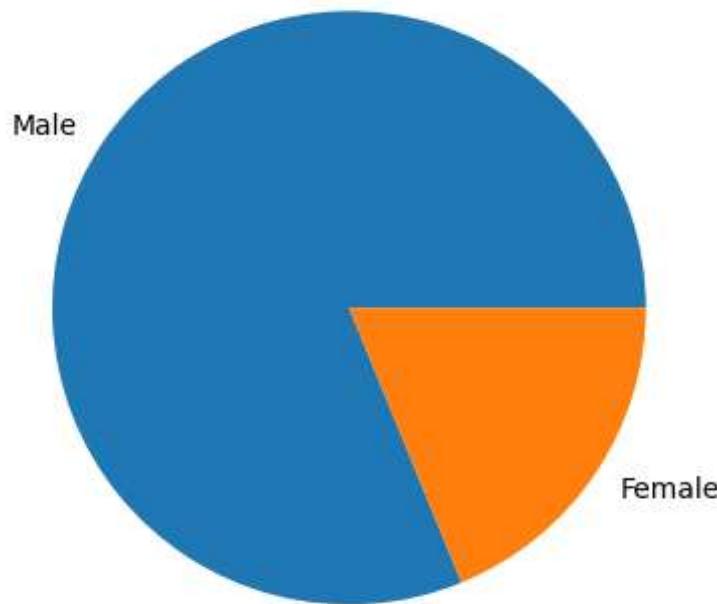
```
In [ ]: data.groupby('Gender')['Personal_loan'].value_counts()
```

Table 6

Gender	Personal_loan
Female	No
	Yes
Male	Yes
	No
Name: count, dtype: int64	

```
In [ ]: plt.pie(data=data,x=data['Gender'][data['Personal_loan']=='Yes'].value_counts(),
```

Figure 10



Males have taken more Personal Loans than Females.

Marital Status and Dependents

- o What is the average number of dependents for married individuals versus single individuals

```
In [ ]: data.columns
```

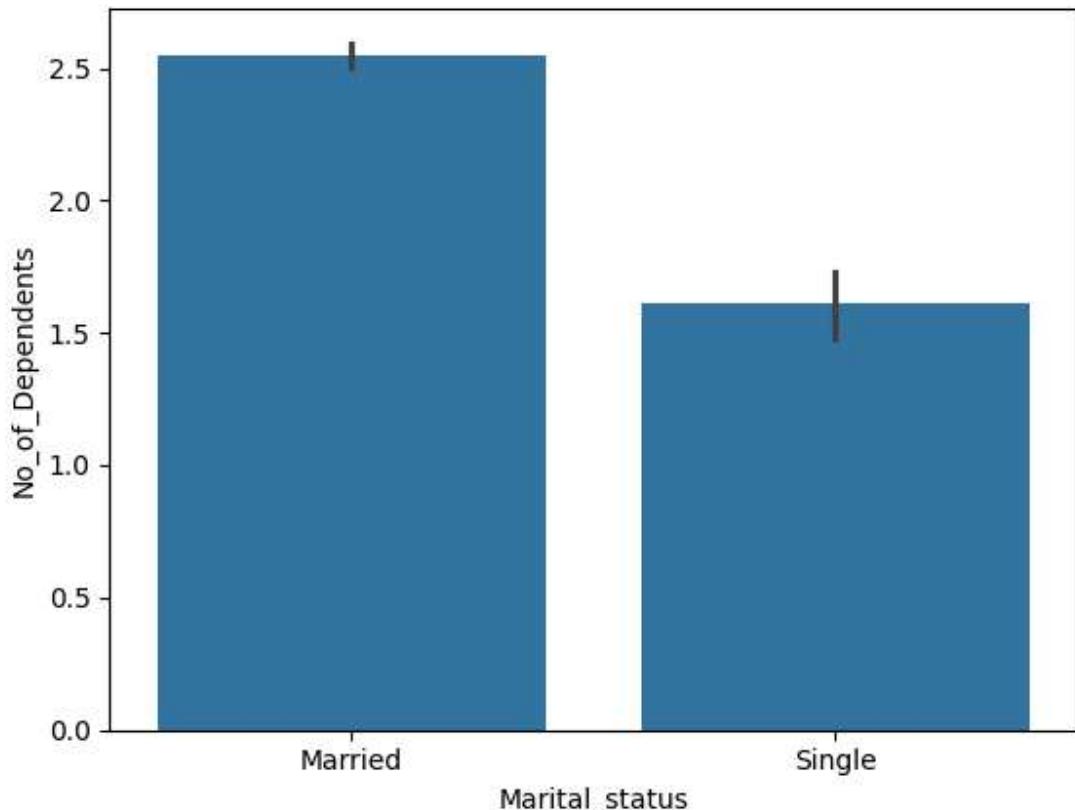
```
In [ ]: data.groupby('Marital_status')['No_of_Dependents'].mean()
```

Table 7

```
Marital_status
Married    2.547471
Single     1.608696
Name: No_of_Dependents, dtype: float64
```

```
In [ ]: sns.barplot(data=data,x=data[ 'Marital_status' ],y=data[ 'No_of_Dependents' ])
```

Figure 11



Partner Employment

- o How does the employment status of a partner affect the total combined salary?

```
In [ ]: data.groupby( 'Partner_working')[ 'Total_salary' ].mean()
```

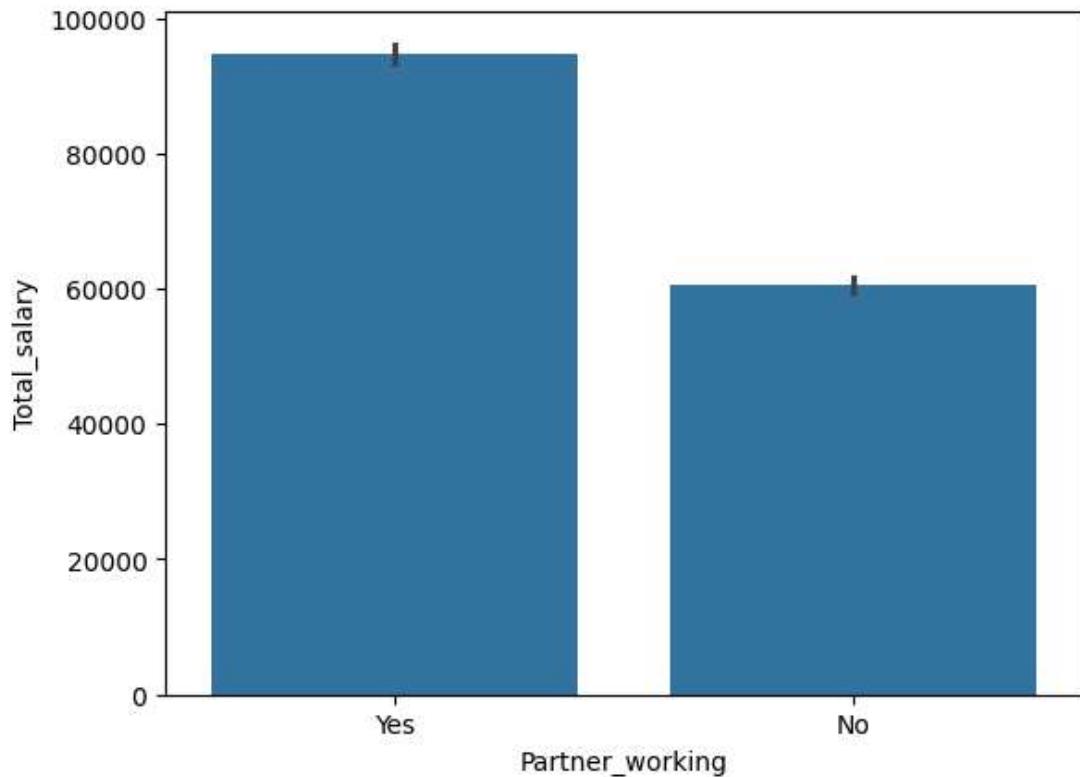
Table 8

```
Partner_working
No      60527.208976
Yes     94900.000000
Name: Total_salary, dtype: float64
```

The total salary is more in cases where the partner is working.

```
In [ ]: sns.barplot(data=data, x=data['Partner_working'], y=data['Total_salary'])
```

Figure 12



Salary Comparison

- o Compare the average salary of individuals whose partners are working versus those whose partners are not working.

```
In [ ]: data.groupby('Partner_working')['Salary'].mean()
```

Table 9

```
Partner_working
No      60256.451613
Yes     60281.336406
Name: Salary, dtype: float64
```

```
In [ ]: plt.figure(figsize=(15,2))
sns.barplot(data=data, y=data['Partner_working'], x=data['Salary'])
```

Figure 13



Average salary is slightly more in the case where the partner is working.

House Loan Analysis

- o What is the proportion of individuals with house loans based on their profession?

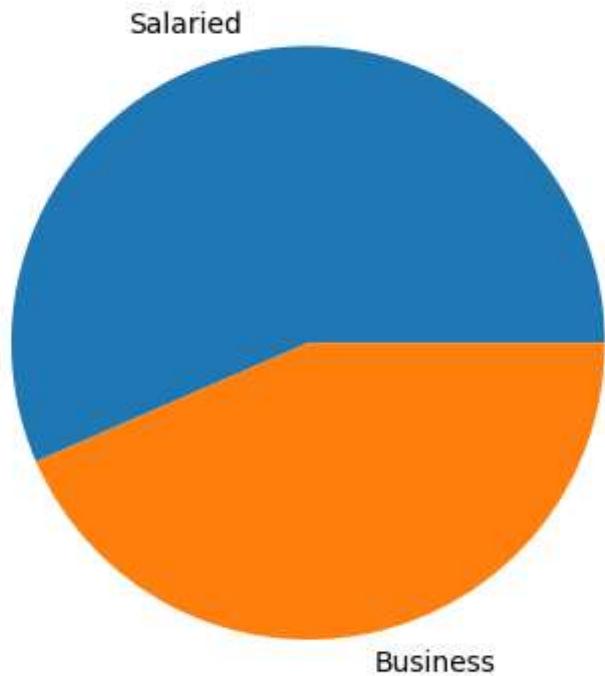
```
In [ ]: data['House_loan'].unique()  
data.groupby('Profession')['House_loan'].value_counts()
```

Table 10

Profession	House_loan	
Business	No	456
	Yes	229
Salaried	No	598
	Yes	298
Name: count, dtype: int64		

```
In [ ]: plt.pie(data=data, x=data['Profession'][data['House_loan']=='Yes'].value_counts()
```

Figure 14



People who are salaried have taken more House Loans than people who are salaried.

Salary Distribution

- o What is the distribution of salaries for individuals with personal loans versus those without personal loans? Represent it using a box plot.

```
In [ ]: data.groupby('Personal_loan')['Salary'].value_counts()
```

Table 11

Personal_loan	Salary	count
No	51400.0	6
	56000.0	6
	56400.0	6
	59450.0	6
	59900.0	6
	..	
Yes	95500.0	1
	97700.0	1
	98400.0	1
	98600.0	1
	99300.0	1

Name: count, Length: 817, dtype: int64

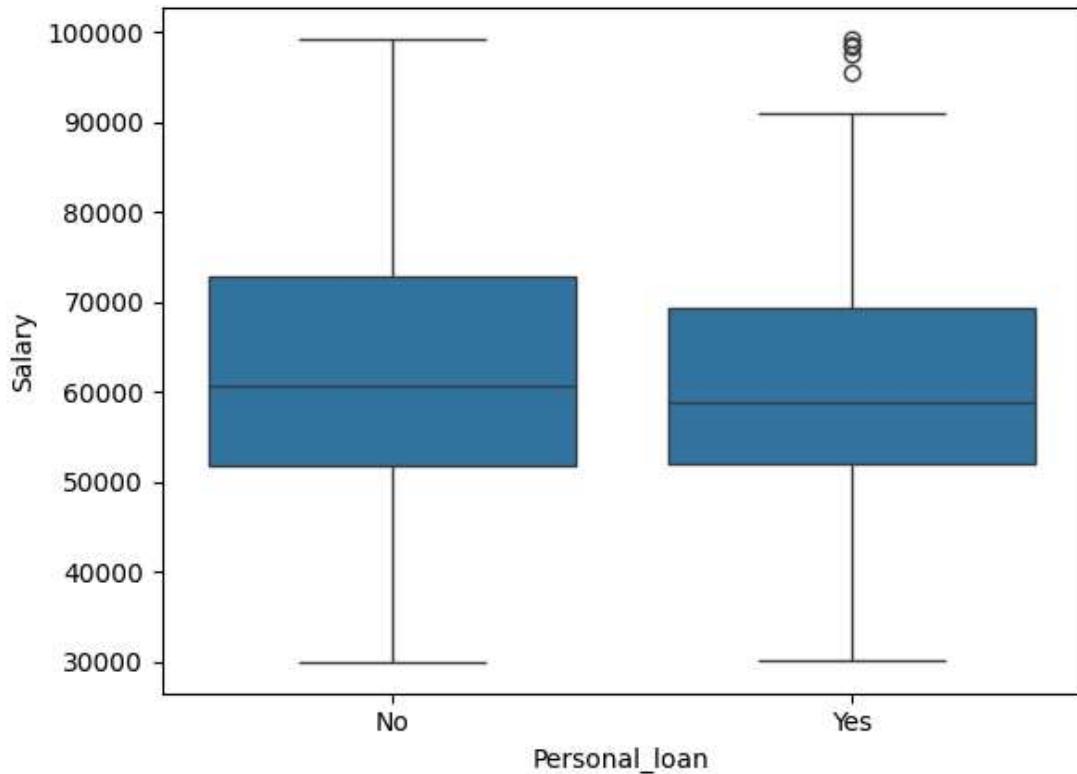
```
In [ ]: data.groupby('Personal_loan')['Salary'].mean()
```

Table 12

```
Personal_loan
No      61155.13308
Yes     59388.44697
Name: Salary, dtype: float64
```

```
In [ ]: sns.boxplot(data=data, x=data['Personal_loan'], y=data['Salary'])
```

Figure 15



Salary of people with no personal loan is on average greater than those with personal loans. Although there are a few people with high salaries who have a personal loan.

Automobile Make Analysis:

- o How does the type of automobile relate to the salary of the individuals? Provide insights based on the make of the automobile.

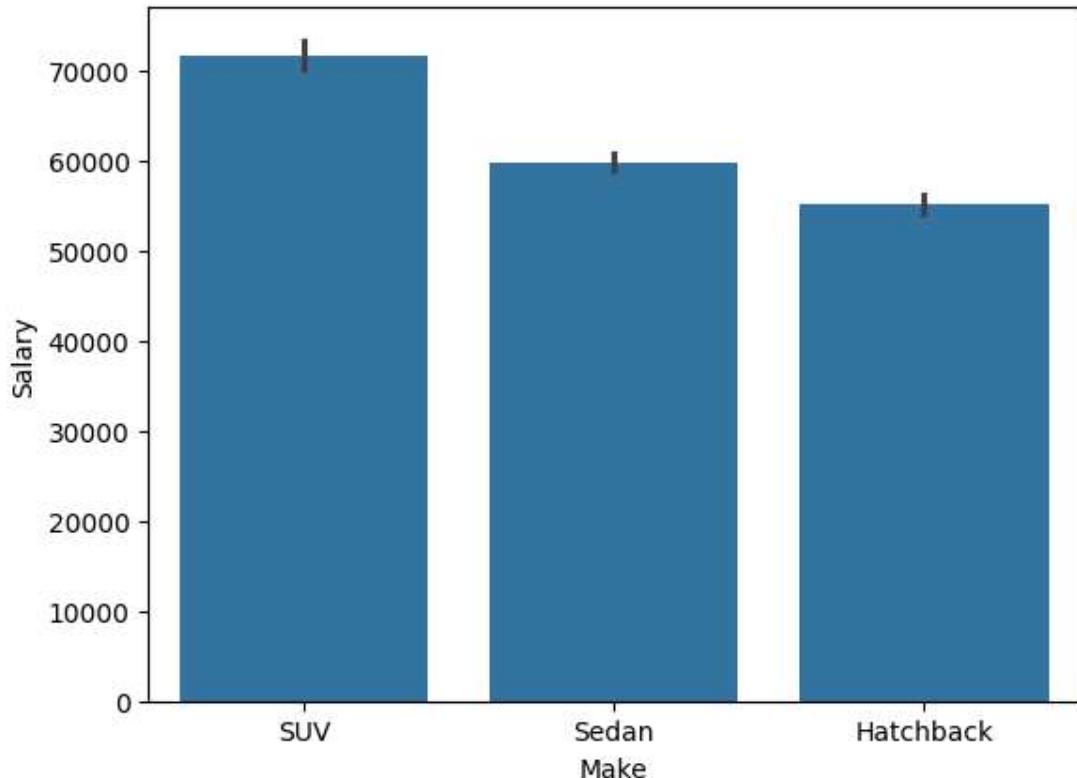
```
In [ ]: data.groupby('Make')['Salary'].mean()
```

Table 13

```
Make
Hatchback    55083.505155
SUV          71642.203390
Sedan        59792.613636
Name: Salary, dtype: float64
```

```
In [ ]: sns.barplot(data=data,x=data[ 'Make' ],y=data[ 'Salary' ])
```

Figure 16



People who have SUVs have the highest salaries on average. Sedan Makes are a second while Hatchbacks are the lowest.

Price Analysis

- o What is the average price of the product/service in the dataset? How does this price vary based on the individual's total salary?

```
In [ ]: data[ 'Price' ].mean()
```

The average price of the products is 35568.66413662239

```
In [ ]: data[ [ 'Price' , 'Total_salary' ] ].corr()
```

Table 14

	Price	Total_salary
Price	1.000000	0.358806
Total_salary	0.358806	1.000000

```
In [ ]: sns.heatmap(data[['Price', 'Total_salary']].corr())
```

Figure 17



The price of the products do not vary much(close to almost none) with the salaries of the individual. The correlation coefficient between them is 0.358806

Marital Status and Loans

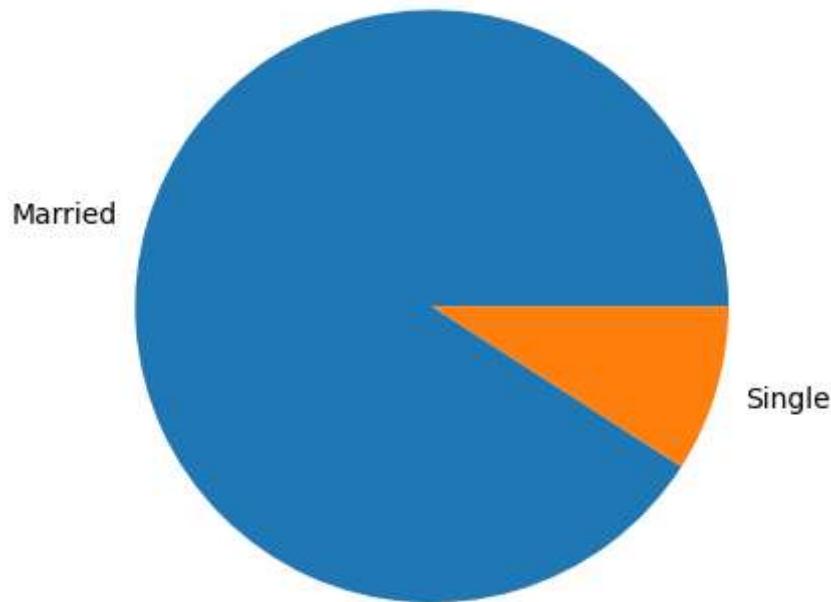
- o Is there a significant difference in the number of personal loans taken by married individuals compared to single individuals?

```
In [ ]: data['Marital_status'][data['Personal_loan']=='Yes'].value_counts()
```

Marital_status	
Married	720
Single	72
Name: count, dtype: int64	

```
In [ ]: plt.pie(data=data,x=data['Marital_status'][data['Personal_loan']=='Yes'].value_c
```

Figure 18



There is a significant difference. People who have married have taken more personal loans than people who are single.

Educational Qualification Impact

- o How does educational qualification impact the likelihood of taking a house loan?

```
In [ ]: data.groupby('Education')['House_loan'].value_counts()
```

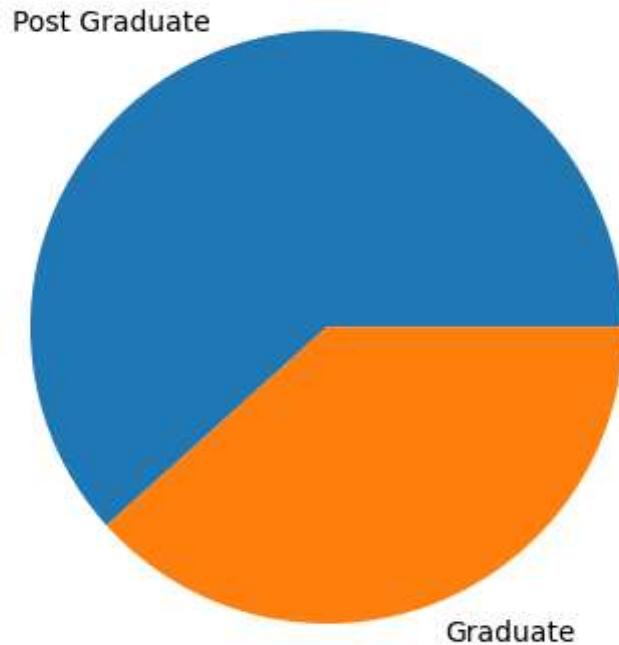
Table 15

Education	House_loan	
Graduate	No	394
	Yes	202
Post Graduate	No	660
	Yes	325

Name: count, dtype: int64

```
In [ ]: plt.pie(data=data,x=data['Education'][data['House_loan']=='Yes'].value_counts()
```

Figure 19



Post Graduates are more likely to take a house loan.

Dependent Count Analysis

- Analyze the number of dependents based on the profession of the individual. Which profession has the highest average number of dependents?

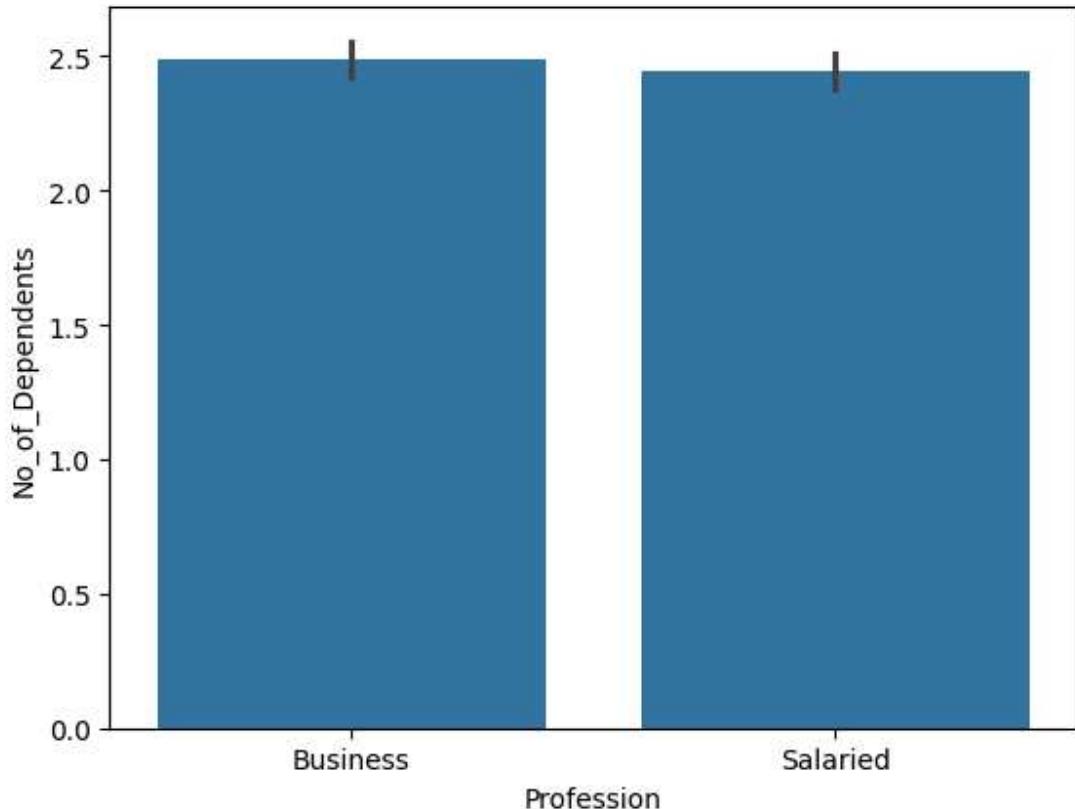
```
In [ ]: data.groupby('Profession')['No_of_Dependents'].mean()
```

Table 16

```
Profession
Business      2.490511
Salaried      2.446429
Name: No_of_Dependents, dtype: float64
```

```
In [ ]: sns.barplot(data=data,x=data['Profession'],y=data['No_of_Dependents'])
```

Figure 20



Gender and Salary

- Is there a significant difference in salaries between males and females? Provide statistical evidence.*

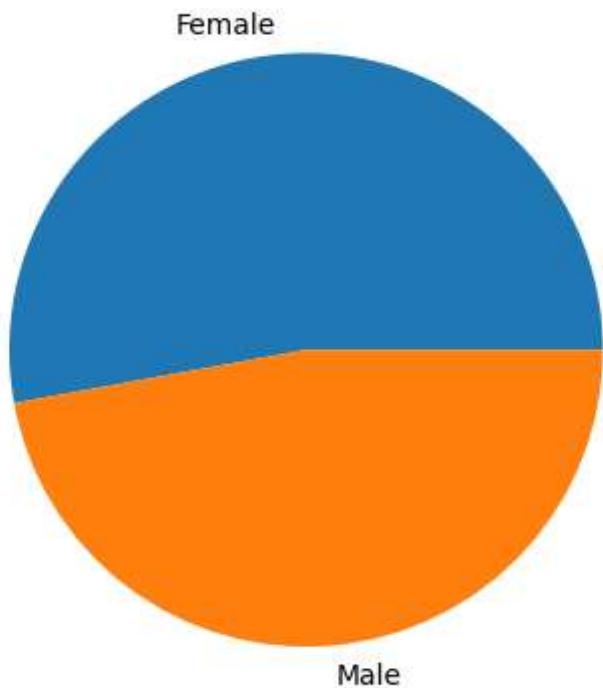
```
In [ ]: data.groupby('Gender')['Salary'].mean()
```

Table 17

```
Gender
Female    65948.024316
Male      58778.075080
Name: Salary, dtype: float64
```

```
In [ ]: plt.pie(data=data, x=data.groupby('Gender')['Salary'].mean().values, labels=data
```

Figure 21



Females have a higher average salary than males, although it is not very significant.

Loan Status Impact

- o How does having a personal loan affect the total combined salary of the individual and their partner?

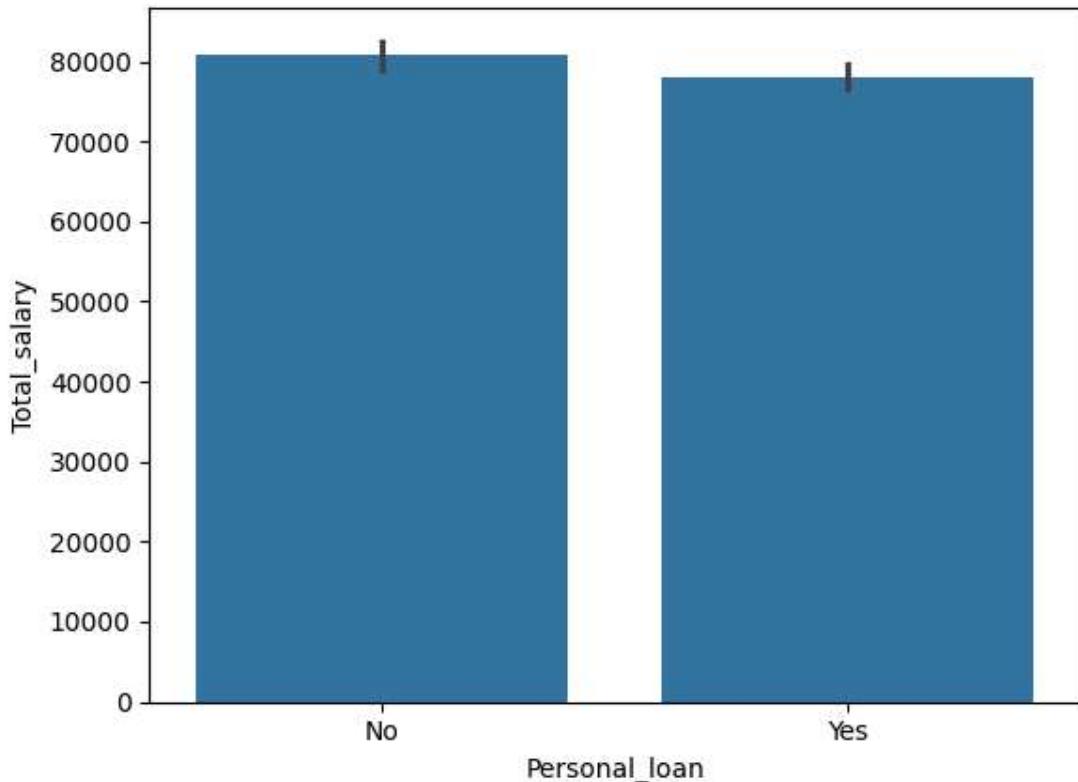
```
In [ ]: data.groupby('Personal_loan')['Total_salary'].mean()
```

Table 18

```
Personal_loan
No      80742.839037
Yes     78059.343434
Name: Total_salary, dtype: float64
```

```
In [ ]: sns.barplot(data=data,x=data['Personal_loan'],y=data['Total_salary'])
```

Figure 22



Partner's Salary Contribution

- o What is the average partner's salary for individuals with and without house loans?

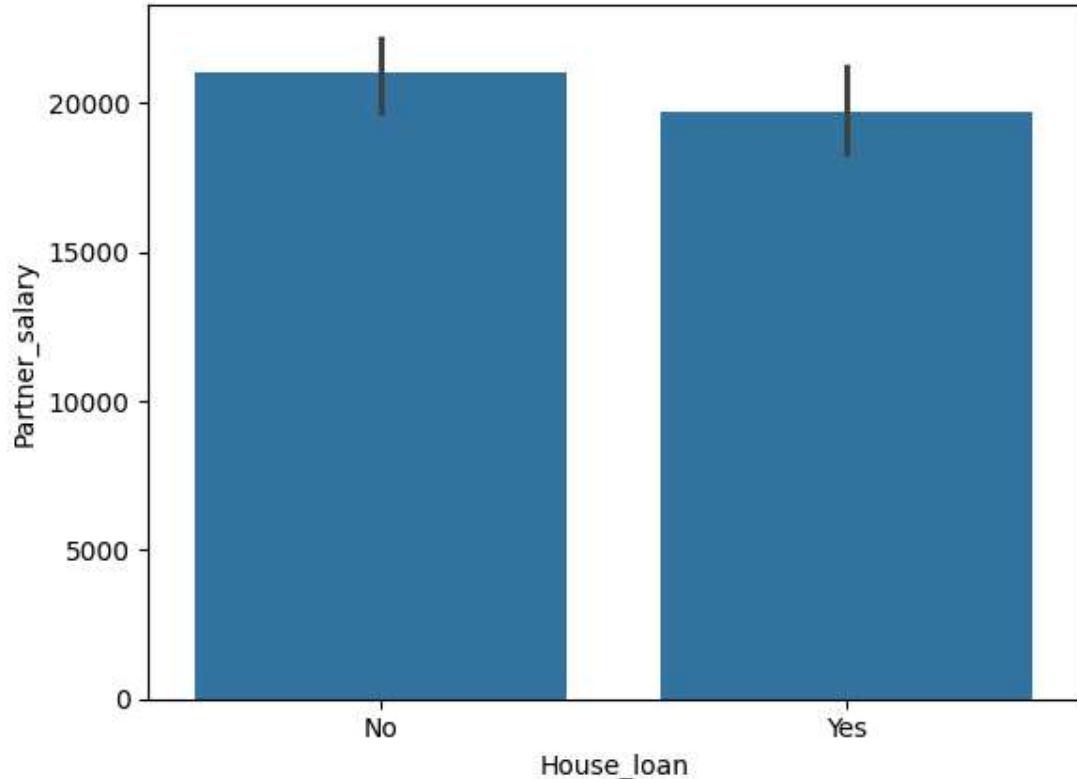
```
In [ ]: data.groupby('House_loan')[['Partner_salary']].mean()
```

Table 19

```
House_loan
No      21028.462998
Yes     19700.759013
Name: Partner_salary, dtype: float64
```

```
In [ ]: sns.barplot(data=data,x=data['House_loan'],y=data['Partner_salary'])
```

Figure 23

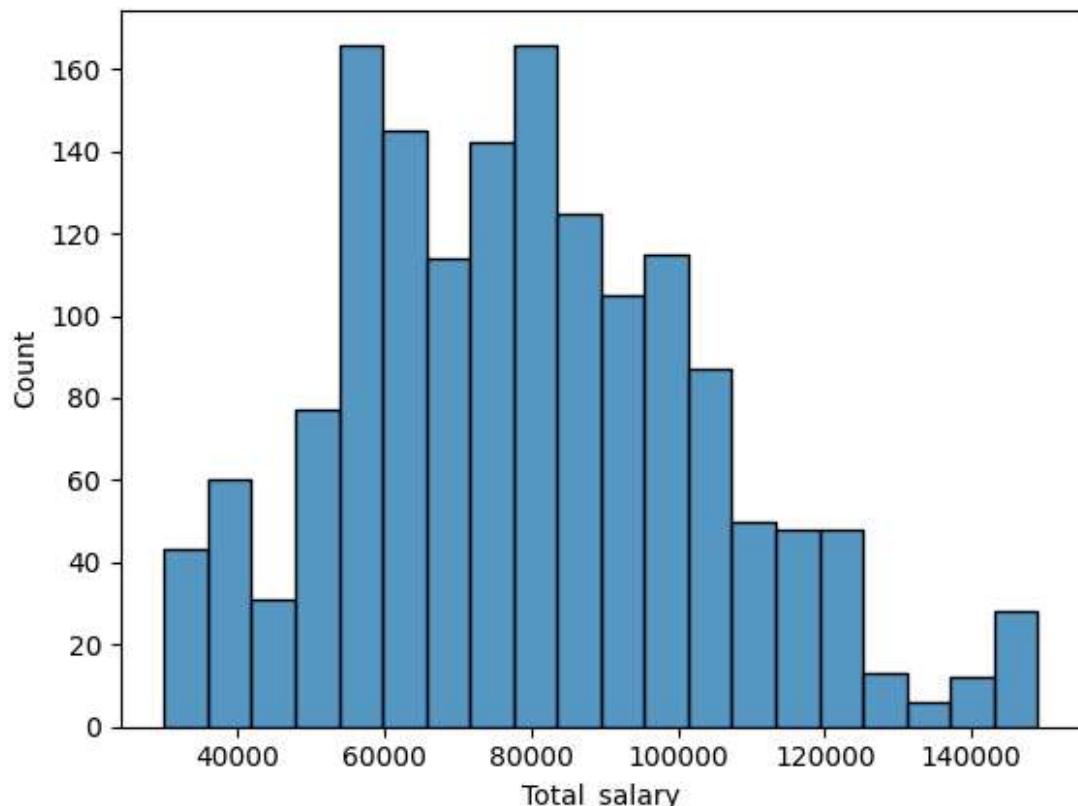


Total Salary Distribution

- Create a histogram showing the distribution of total combined salaries. Identify and discuss any skewness or outliers in the data.

```
In [ ]: sns.histplot(data=data,x=data['Total_salary'])
```

Figure 24



In []: