In [18]:
```
# import the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [19]:
```
# upload data
df= pd.read_csv("F:\\datasets folder\\austo_automobile.csv")
```

In [20]:
```
# see the heads
df.head(5)
```

Out[20]:

	Age	Gender	Profession	Marital_status	Education	No_of_Dependents	Personal_loan	House_loan
0	53	Male	Business	Married	Post Graduate	4	No	No
1	53	Femal	Salaried	Married	Post Graduate	4	Yes	No
2	53	Female	Salaried	Married	Post Graduate	3	No	No
3	53	Female	Salaried	Married	Graduate	2	Yes	No
4	53	Male	Salaried	Married	Post Graduate	3	No	No

In [21]:
```
# see the shape of data
df.shape
```

Out[21]: (1581, 14)

In [22]:
```
# get information
df.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        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    1475 non-null   float64
 11  Total_salary      1581 non-null   int64
```

```
 12  Price             1581 non-null   int64
 13  Make              1581 non-null   object
dtypes: float64(1), int64(5), object(8)
memory usage: 173.0+ KB
```

In [23]:
```
# get all the columns
df.columns
```

Out[23]:
```
Index(['Age', 'Gender', 'Profession', 'Marital_status', 'Education',
       'No_of_Dependents', 'Personal_loan', 'House_loan', 'Partner_working',
       'Salary', 'Partner_salary', 'Total_salary', 'Price', 'Make'],
      dtype='object')
```

In [24]:
```
# correct the spell errors in data
df.Gender = df.Gender.str.replace('Femal', 'Female')
df
```

Out[24]:

	Age	Gender	Profession	Marital_status	Education	No_of_Dependents	Personal_loan	House_l
0	53	Male	Business	Married	Post Graduate	4	No	
1	53	Female	Salaried	Married	Post Graduate	4	Yes	
2	53	Femalee	Salaried	Married	Post Graduate	3	No	
3	53	Femalee	Salaried	Married	Graduate	2	Yes	
4	53	Male	Salaried	Married	Post Graduate	3	No	
...	...	...	...	...	...	...	...	
1576	22	Male	Salaried	Single	Graduate	2	No	
1577	22	Male	Business	Married	Graduate	4	No	
1578	22	Male	Business	Single	Graduate	2	No	
1579	22	Male	Business	Married	Graduate	3	Yes	
1580	22	Male	Salaried	Married	Graduate	4	No	

1581 rows × 14 columns

In [25]:
```
# see the statistical summary of data
df.describe()
```

Out[25]:

	Age	No_of_Dependents	Salary	Partner_salary	Total_salary	Price
count	1581.000000	1581.000000	1581.000000	1475.000000	1581.000000	1581.000000
mean	31.922201	2.457938	60392.220114	20225.559322	79625.996205	35597.722960
std	8.425978	0.943483	14674.825044	19573.149277	25545.857768	13633.636545
min	22.000000	0.000000	30000.000000	0.000000	30000.000000	18000.000000
25%	25.000000	2.000000	51900.000000	0.000000	60500.000000	25000.000000
50%	29.000000	2.000000	59500.000000	25600.000000	78000.000000	31000.000000

	Age	No_of_Dependents	Salary	Partner_salary	Total_salary	Price
75%	38.000000	3.000000	71800.000000	38300.000000	95900.000000	47000.000000
max	54.000000	4.000000	99300.000000	80500.000000	171000.000000	70000.000000

In [26]:
```
# correction spells in data:
df.Gender = df.Gender.str.replace('Femalee', 'Female')
df
```

Out[26]:

	Age	Gender	Profession	Marital_status	Education	No_of_Dependents	Personal_loan	House_lo
0	53	Male	Business	Married	Post Graduate	4	No	
1	53	Female	Salaried	Married	Post Graduate	4	Yes	
2	53	Female	Salaried	Married	Post Graduate	3	No	
3	53	Female	Salaried	Married	Graduate	2	Yes	
4	53	Male	Salaried	Married	Post Graduate	3	No	
...	...	...	...	...	...	...	...	
1576	22	Male	Salaried	Single	Graduate	2	No	
1577	22	Male	Business	Married	Graduate	4	No	
1578	22	Male	Business	Single	Graduate	2	No	
1579	22	Male	Business	Married	Graduate	3	Yes	
1580	22	Male	Salaried	Married	Graduate	4	No	

1581 rows × 14 columns

In [27]:
```
# correction of spells:
df.Gender = df.Gender.str.replace('Femle', 'Female')
df
```

Out[27]:

	Age	Gender	Profession	Marital_status	Education	No_of_Dependents	Personal_loan	House_lo
0	53	Male	Business	Married	Post Graduate	4	No	
1	53	Female	Salaried	Married	Post Graduate	4	Yes	
2	53	Female	Salaried	Married	Post Graduate	3	No	
3	53	Female	Salaried	Married	Graduate	2	Yes	
4	53	Male	Salaried	Married	Post Graduate	3	No	
...	...	...	...	...	...	...	...	
1576	22	Male	Salaried	Single	Graduate	2	No	

	Age	Gender	Profession	Marital_status	Education	No_of_Dependents	Personal_loan	House_lo
1577	22	Male	Business	Married	Graduate	4	No	
1578	22	Male	Business	Single	Graduate	2	No	`
1579	22	Male	Business	Married	Graduate	3	Yes	`
1580	22	Male	Salaried	Married	Graduate	4	No	

1581 rows × 14 columns

In [28]:
```
# see the null values
df.isnull()
```

Out[28]:

	Age	Gender	Profession	Marital_status	Education	No_of_Dependents	Personal_loan	House_l
0	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	F
...	...	...	...	...	...	...	...	
1576	False	False	False	False	False	False	False	F
1577	False	False	False	False	False	False	False	F
1578	False	False	False	False	False	False	False	F
1579	False	False	False	False	False	False	False	F
1580	False	False	False	False	False	False	False	F

1581 rows × 14 columns

In [29]:
```
# there is null values present in data
df.isnull().sum().any()
```

Out[29]: True

In [30]:
```
# drop the values
df.dropna(axis=0, inplace=True)
df.head()
```

Out[30]:

	Age	Gender	Profession	Marital_status	Education	No_of_Dependents	Personal_loan	House_loan
0	53	Male	Business	Married	Post Graduate	4	No	No
1	53	Female	Salaried	Married	Post Graduate	4	Yes	No
2	53	Female	Salaried	Married	Post Graduate	3	No	No

	Age	Gender	Profession	Marital_status	Education	No_of_Dependents	Personal_loan	House_loan
3	53	Female	Salaried	Married	Graduate	2	Yes	No
4	53	Male	Salaried	Married	Post Graduate	3	No	No

In [31]:

```
# fill the null values with mean
df['Partner_salary']=
df['Partner_salary'].fillna(df['Partner_salary'].mean())
df
```

Out[31]:

	Age	Gender	Profession	Marital_status	Education	No_of_Dependents	Personal_loan	House_lo
0	53	Male	Business	Married	Post Graduate	4	No	
1	53	Female	Salaried	Married	Post Graduate	4	Yes	
2	53	Female	Salaried	Married	Post Graduate	3	No	
3	53	Female	Salaried	Married	Graduate	2	Yes	
4	53	Male	Salaried	Married	Post Graduate	3	No	
...	...	...	...	...	...	...	...	
1574	22	Male	Salaried	Married	Graduate	3	Yes	
1575	22	Male	Salaried	Married	Graduate	3	Yes	
1576	22	Male	Salaried	Single	Graduate	2	No	
1578	22	Male	Business	Single	Graduate	2	No	
1580	22	Male	Salaried	Married	Graduate	4	No	

1425 rows × 14 columns

In [32]:

```
# Now, no values here
df.isnull().sum()
```

Out[32]:

```
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
```

In [33]:
```
# see the value counts
df.value_counts(ascending=True)
```

Out[33]:
```
Age  Gender  Profession  Marital_status  Education      No_of_Dependents  Personal_loa
n  House_loan  Partner_working  Salary  Partner_salary  Total_salary  Price  Make
22   Female  Salaried    Married         Graduate       3                 No
No          Yes              34800   28100.0         62900          31000  Hatchback
1
35   Female  Business    Married         Post Graduate  1                 No
No          No               63700   0.0             63700          37000  Sedan
1
                                         Graduate       4                 No
No          Yes              53200   25800.0         79000          66000  SUV
1
34   Male    Salaried    Married         Post Graduate  2                 Yes
No          Yes              58900   40900.0         99800          43000  Sedan
1
                                                                          No
Yes         Yes              61100   38300.0         99400          43000  Sedan
1

..
27   Male    Business    Married         Graduate       3                 No
No          Yes              59200   40600.0         99800          26000  Hatchback
1

52000   30100.0         82100         27000  Sedan        1

51200   30100.0         81300         24000  Sedan        1
                                         Post Graduate  3                 No
No          Yes              61300   30400.0         91700          24000  Hatchback
1
54   Male    Salaried    Married         Post Graduate  3                 No
No          Yes              82200   60900.0         143100         54000  SUV
1
Length: 1425, dtype: int64
```

In [34]:
```
# see unique values
df.nunique()
```

Out[34]:
```
Age                  33
Gender                2
Profession            2
Marital_status        2
Education             2
No_of_Dependents      5
Personal_loan         2
House_loan            2
Partner_working       2
Salary              526
Partner_salary      147
Total_salary        726
Price                53
Make                  3
dtype: int64
```

C. Explore all the features of the data separately by using appropriate visualizations and draw insights that can be utilized by the business.

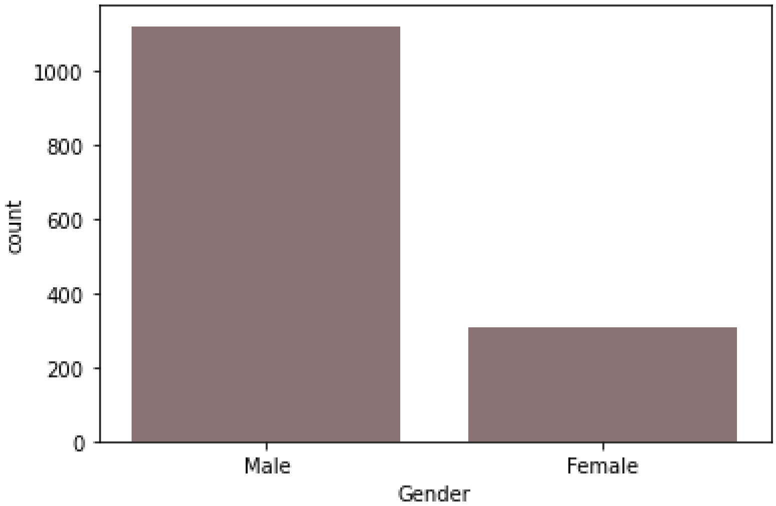
In [17]:
```
# blank cell
```

In [40]:
```
# blank cell
```

In [35]:

```
# make count plot of an each category to get insights of data
# there is number of male are more than female who uses cars, men are using more cars as
# compare to women.
sns.countplot(x='Gender',data=df, color='red', saturation=0.1)
```

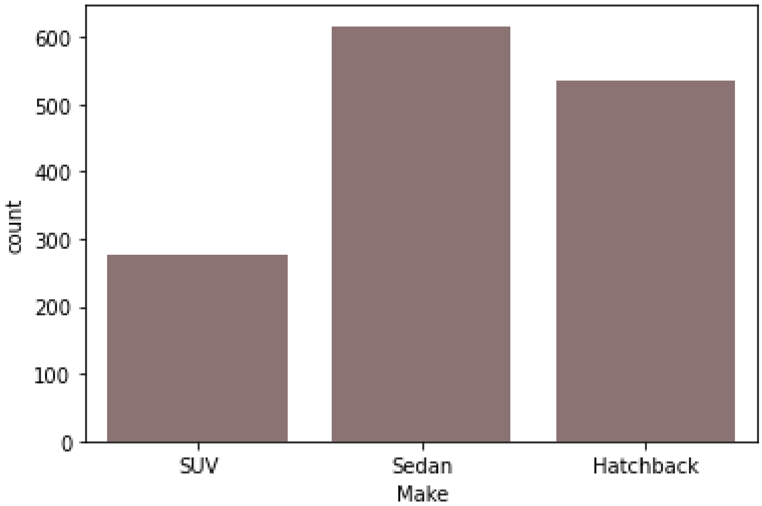
Out[35]: <AxesSubplot:xlabel='Gender', ylabel='count'>



In [42]:

```
# In this plot how many suv , sedan , hatchback cars are using by the customers, it shows the
# count of an each category of cars.
sns.countplot(x='Make',data=df, color='red', saturation=0.1)
```

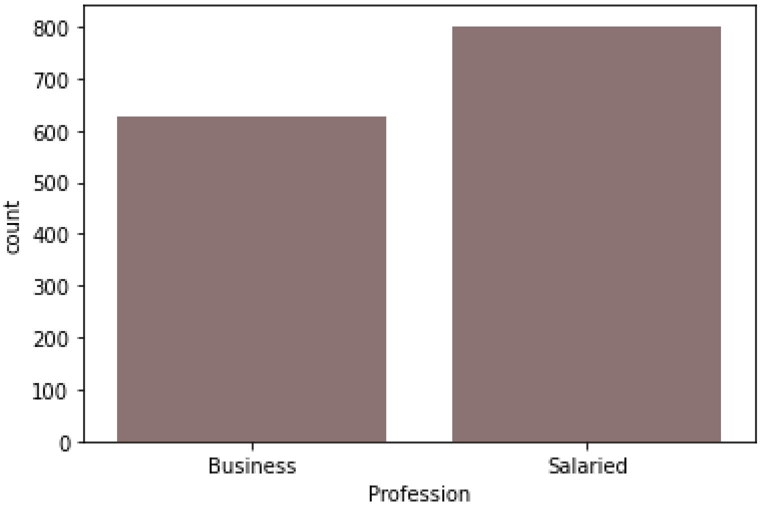
Out[42]: <AxesSubplot:xlabel='Make', ylabel='count'>



In [43]:

```
#This plot shows the count of the professional persons, in this plot salaried persons are higher
# than business persons.
sns.countplot(x='Profession',data=df, color='red', saturation=0.1)
```

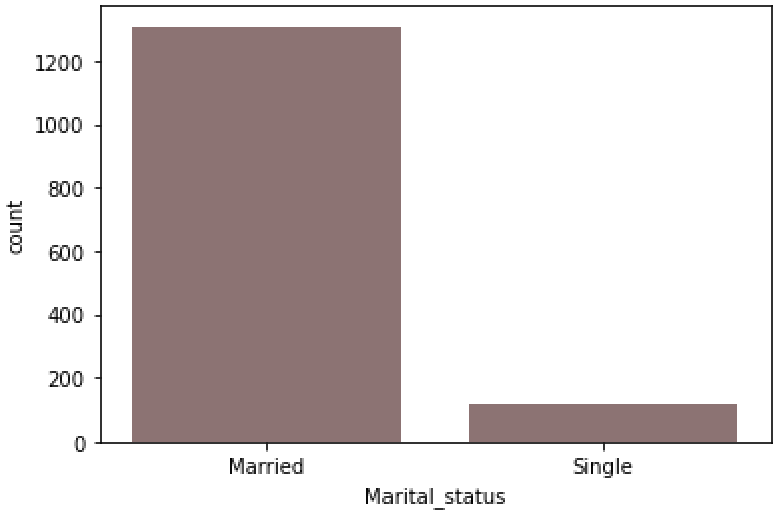
Out[43]: <AxesSubplot:xlabel='Profession', ylabel='count'>



In [44]:

```
# By this plot we getting the information about Martial Status of customers who are buying cars.
sns.countplot(x='Marital_status',data=df, color='red', saturation=0.1)
```

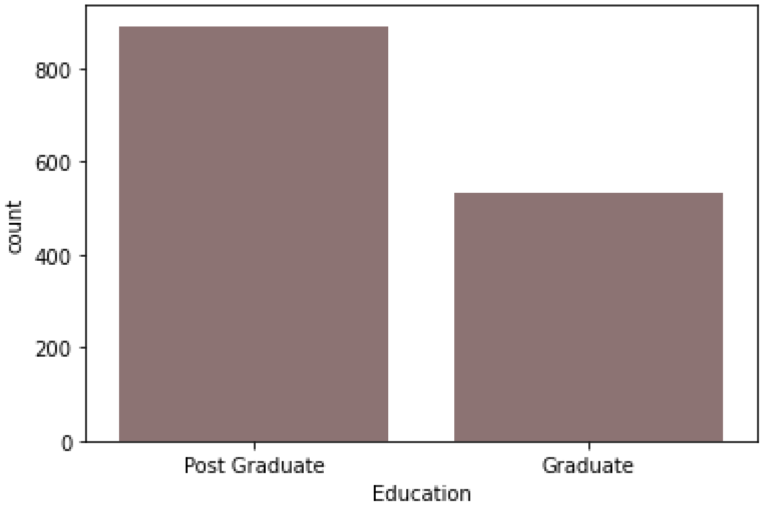
Out[44]: <AxesSubplot:xlabel='Marital_status', ylabel='count'>



In [45]:

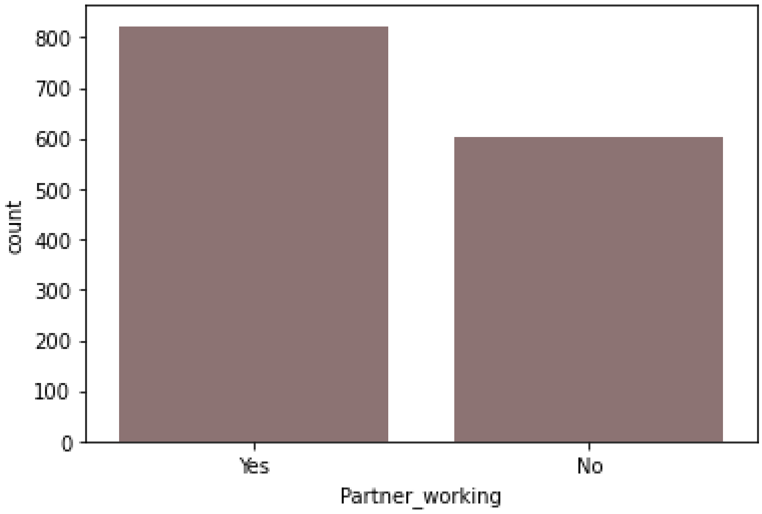
```
# This plot shows an education of person who are buying cars , post graduates are buying more cars
# as comparative to graduate people.
sns.countplot(x='Education',data=df, color='red', saturation=0.1)
```

Out[45]: <AxesSubplot:xlabel='Education', ylabel='count'>



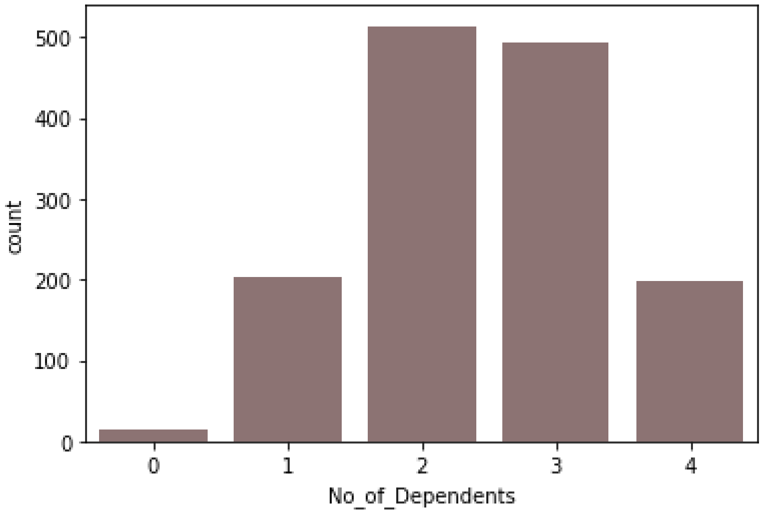
In [46]:
```
# In this plot count of working partners is showing , means who buys cars they have working partner
# , lesser people dont have...
sns.countplot(x='Partner_working',data=df, color='red', saturation=0.1)
```

Out[46]: <AxesSubplot:xlabel='Partner_working', ylabel='count'>



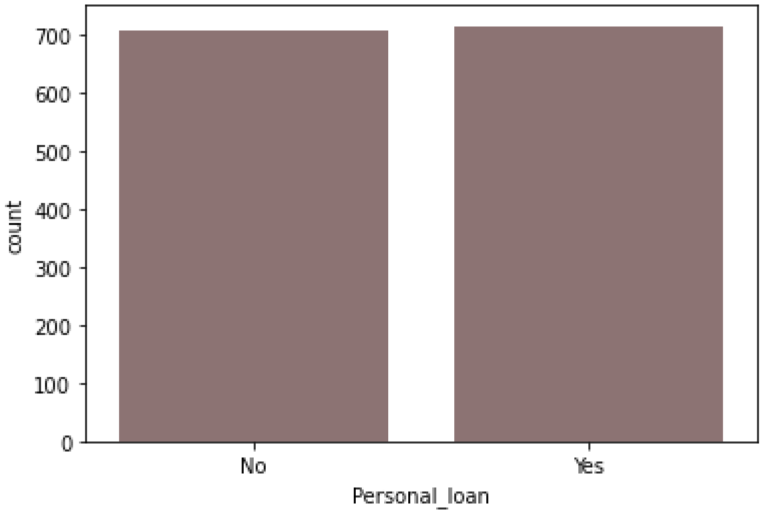
In [47]:
```
# it shows how many dependents they have who are buying cars...
sns.countplot(x='No_of_Dependents',data=df, color='red', saturation=0.1)
```

Out[47]: <AxesSubplot:xlabel='No_of_Dependents', ylabel='count'>



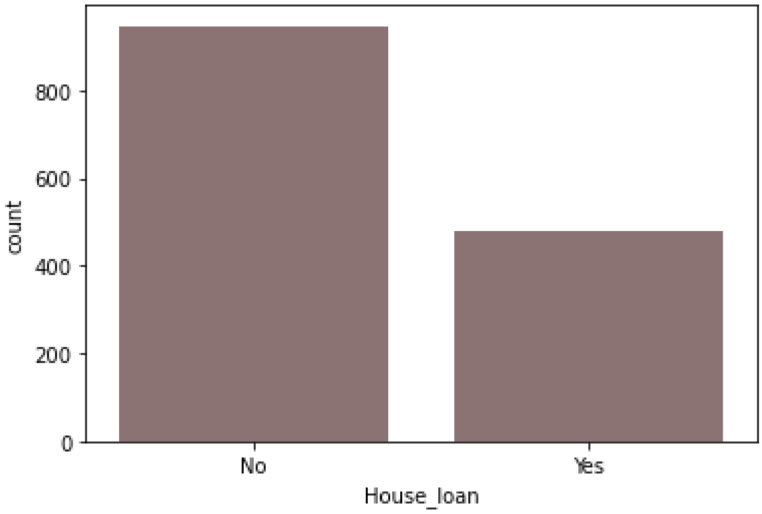
In [48]:
```
# by this plot we are getting how amny people get personal loan , plot shows equal count for them.
sns.countplot(x='Personal_loan',data=df, color='red', saturation=0.1)
```

Out[48]: <AxesSubplot:xlabel='Personal_loan', ylabel='count'>



In [49]:
```
# This plot shows the count of house loan , how many people have taken house loan who are buying cars.
sns.countplot(x='House_loan',data=df, color='red',saturation=0.1)
```

Out[49]: <AxesSubplot:xlabel='House_loan', ylabel='count'>



In [ ]:

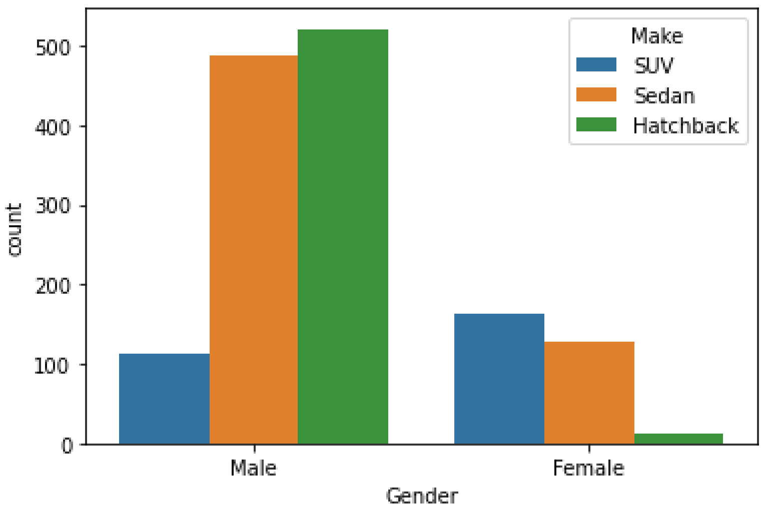
```
Blank cell
```

D. Understanding the relationships among the variables in the dataset is crucial for every analytical project. Perform analysis on the data fields to gain deeper insights. Comment on your understanding of the data.

In [51]:

```
# Make countplot of categories with 'Make' column taken as hue in this plot
# This plot shows how many number of male and female buying SUV, Sedan and Hatchback, it shows count
# of the cars categories which were bought by the male ans female.
sns.countplot(data=df, x="Gender", hue="Make")
```

Out[51]: <AxesSubplot:xlabel='Gender', ylabel='count'>

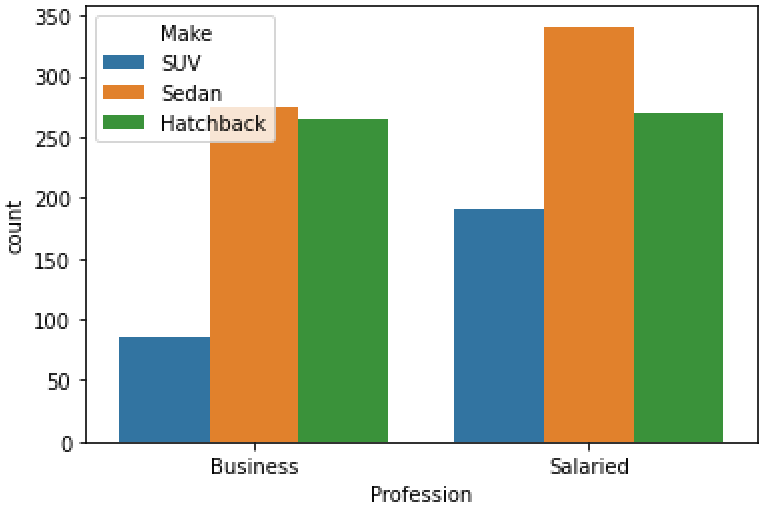


In [52]:

```
# Make a countplot of categories of cars using by the customers who are in what kind of profession they
# are in , here, it is showing business people are using Sedan cars more as compare to suv and hatchback
# and salaried people are also using more sedan cars rather tah using suv and hatchback , so sedan cars
```

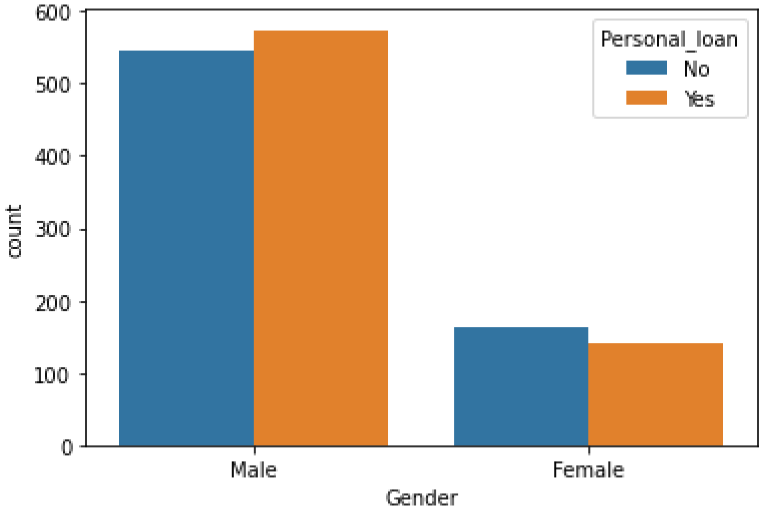
```
# sells is greater than other cars's category.
sns.countplot(data=df, x="Profession", hue="Make")
```

Out[52]: <AxesSubplot:xlabel='Profession', ylabel='count'>



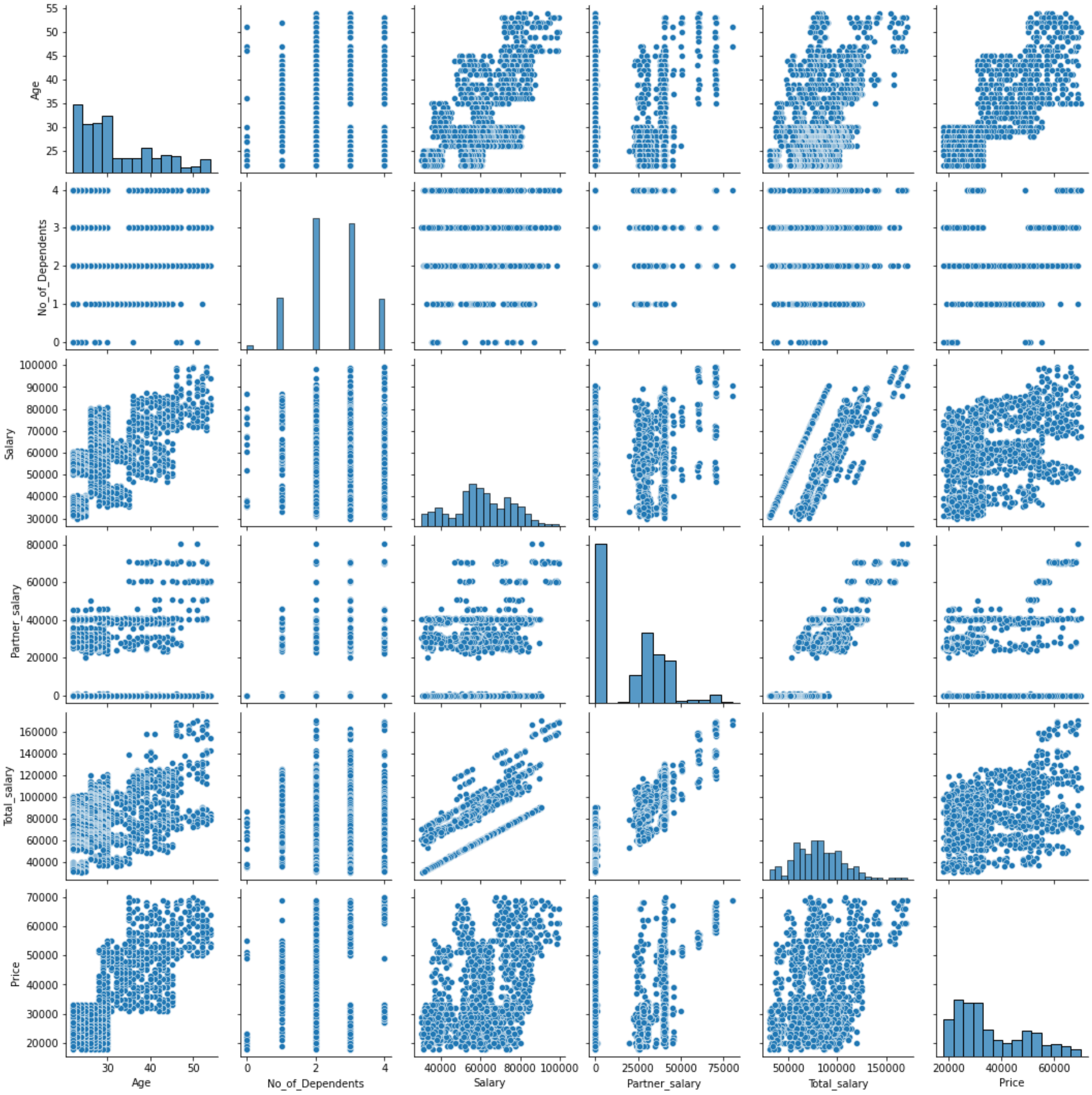
In [53]:
```
# Make a countplot of Gender with personal loan as hue , wherein, male takes more personal loan as comnpare
# to female, this plot shows female has less personal loan.
sns.countplot(data=df, x="Gender", hue="Personal_loan")
# female takes less personal loan as compared to male.
```

Out[53]: <AxesSubplot:xlabel='Gender', ylabel='count'>



In [54]:
```
# Make a pairplot of data which shows the relationship between all the categories , how they are
# related to each other
sns.pairplot(df)
```

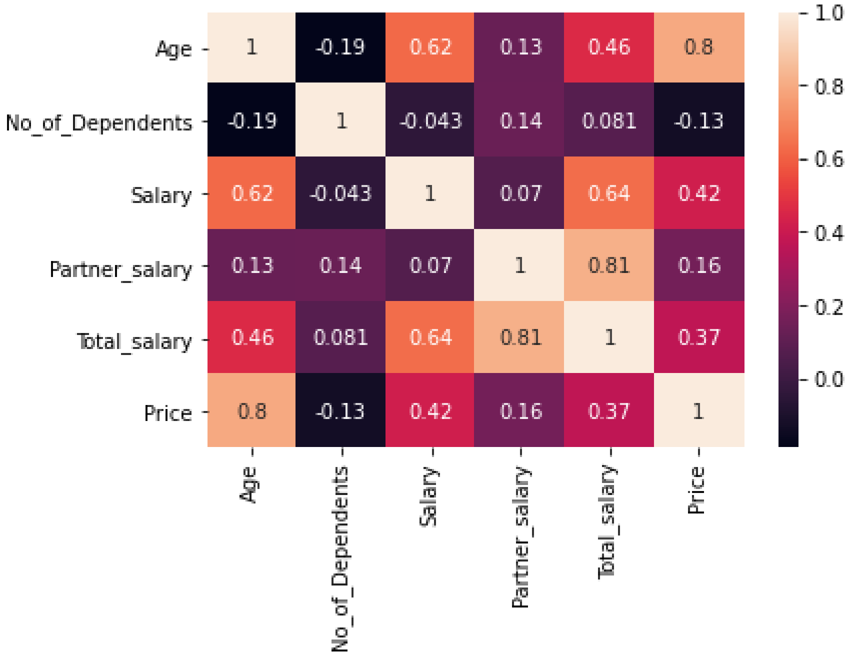
Out[54]: <seaborn.axisgrid.PairGrid at 0x21eeef7f220>



In [55]:
```python
# Correlation between different variables...
corr = df.corr()
```

In [56]:
```python
# Draw the heatmap which shows the relationship between the varaibles how they are related to each other,
# dark blocks shows higher values , medium color shows medium values and light color blocks shows lesser
# values..

sns.heatmap(corr, annot=True)
```

Out[56]: <AxesSubplot:>

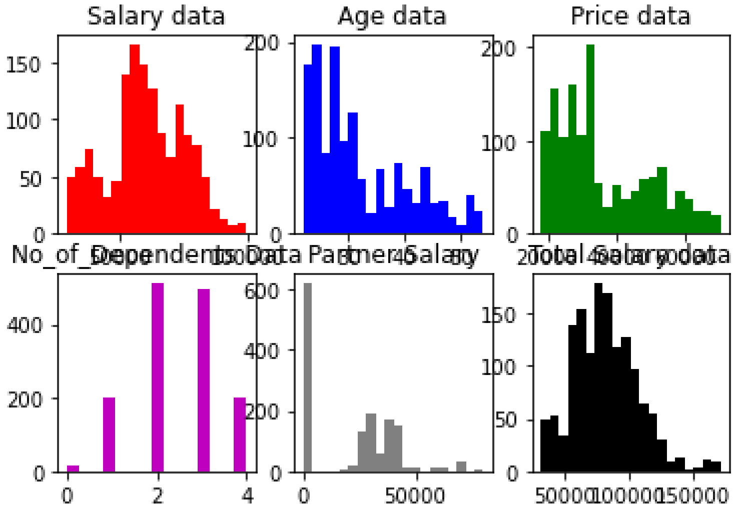


In [57]:

```
# draw hisplots of each category for doing comparisons, this plot shows the comparisons between the
# variables , which shows the salary data, age data, price data,partner salary, no of dependents and
# total salary data.
x  = df['Salary']

plt.subplot(2,3,1)
plt.hist(df['Salary'], bins = 20,color='r')
plt.title('Salary data')

plt.subplot(2,3,2)
plt.hist(df['Age'], bins = 20,color='b')
plt.title('Age data')

plt.subplot(2,3,3)
plt.hist(df['Price'], bins = 20,color='g')
plt.title('Price data')

plt.subplot(2,3,4)
plt.hist(df['No_of_Dependents'], bins = 15,color='m')
plt.title('No_of_Dependents Data')

plt.subplot(2,3,5)
plt.hist(df['Partner_salary'], bins = 20, color='grey')
plt.title('Partner Salary')
```

```python
plt.subplot(2,3,6)
plt.hist(df['Total_salary'], bins = 20, color='black')
plt.title('Total Salary data')
```

Out[57]: `Text(0.5, 1.0, 'Total Salary data')`



E. Employees working on the existing marketing campaign have made the following remarks. Based on the data and your analysis state whether you agree or disagree with their observations. Justify your answer Based on the data available. E1) Steve Roger says “Men prefer SUV by a large margin, compared to the women” E2) Ned Stark believes that a salaried person is more likely to buy a Sedan. E3) Sheldon Cooper does not believe any of them; he claims that a salaried male is an easier target for a SUV sale over a Sedan Sale.

In [59]:
```python
#E1) Steve Roger says “Men prefer SUV by a large margin, compared to the women”
# False, from the below graph, it clearly shows that Women prefer more SUV than men.

df.groupby('Gender').count()
```

Out[59]:

Gender	Age	Profession	Marital_status	Education	No_of_Dependents	Personal_loan	House_loan	Pa
Female	305	305	305	305	305	305	305	
Male	1120	1120	1120	1120	1120	1120	1120	

In [60]:
```python
# groupby of 'make' with count
df.groupby('Make').count()
```

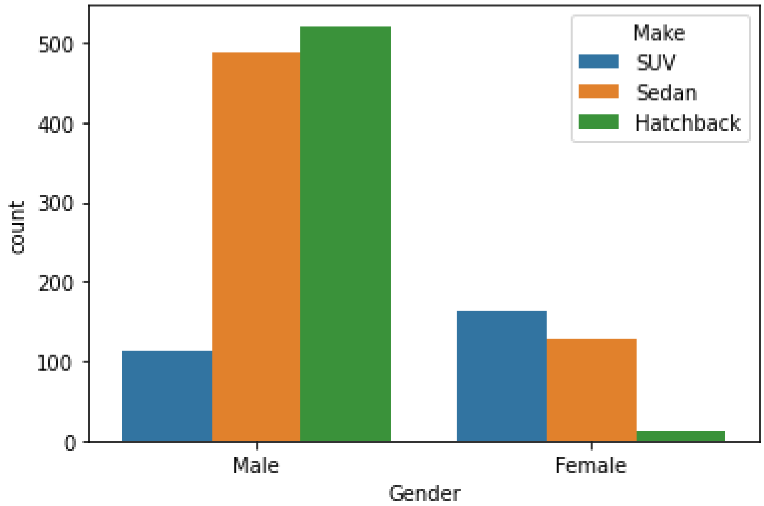
Out[60]:

Make	Age	Gender	Profession	Marital_status	Education	No_of_Dependents	Personal_loan	Ho
Hatchback	534	534	534	534	534	534	534	
SUV	275	275	275	275	275	275	275	

	Age	Gender	Profession	Marital_status	Education	No_of_Dependents	Personal_loan	Ho
Make								
Sedan	616	616	616	616	616	616	616	

In [61]:
```
# It shows the count of gender with make as hue , which tells us how many male and female using which
# category of cars ...
sns.countplot(data=df, x="Gender", hue="Make")
```

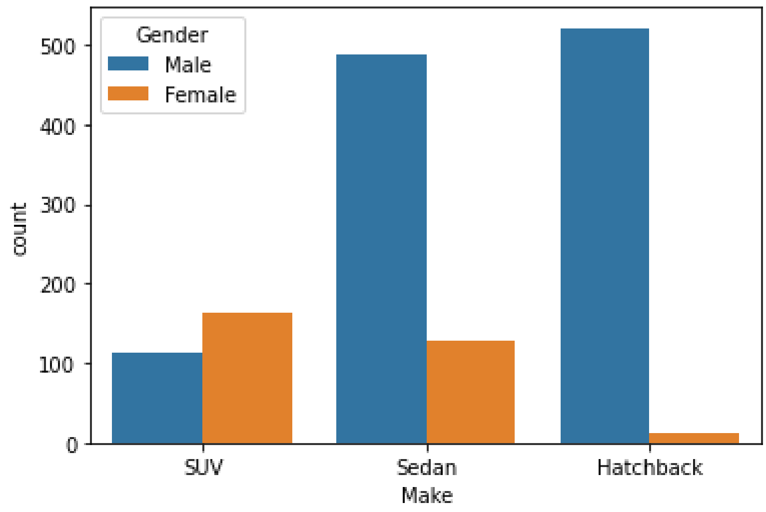
Out[61]: <AxesSubplot:xlabel='Gender', ylabel='count'>



E2) Ned Stark believes that a salaried person is more likely to buy a Sedan. E3) Sheldon Cooper does not believe any of them; he claims that a salaried male is an easier target for a SUV sale over a Sedan Sale.

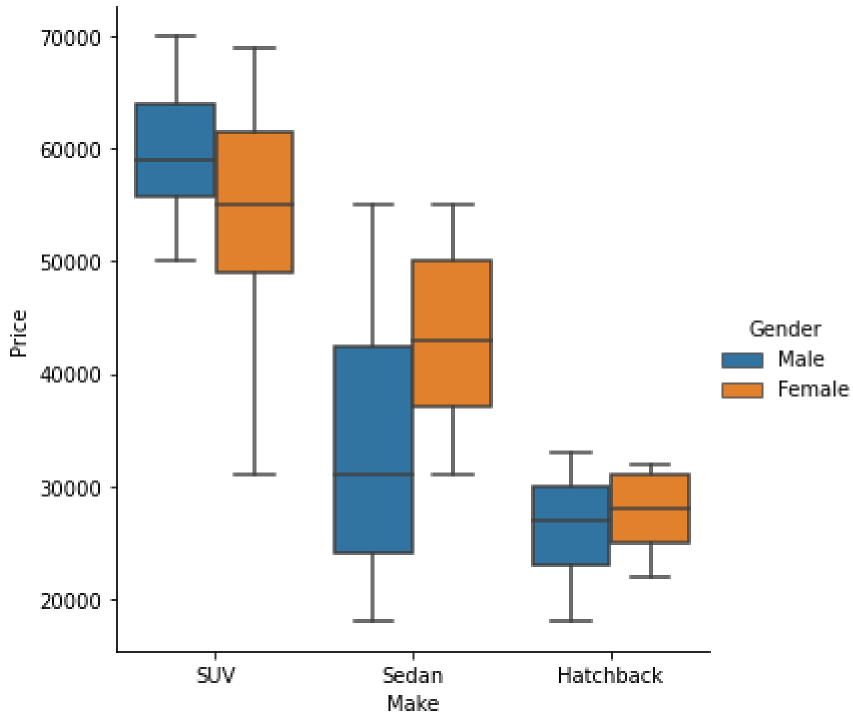
In [62]:
```
# make a counplot of make column with hue as gender,which tells us how many male and female buy which
# cars....
sns.countplot(data=df, x="Make", hue="Gender")
# E2: True, Ned Stark believes that a salaried person is more likely to buy a Sedan not SUV.
# E3: False, it concludes that Ned Stark believes that salaried person is more likely to buy a SUV over a Sudan.
```

Out[62]: <AxesSubplot:xlabel='Make', ylabel='count'>



In [63]:
```
# It shows male are buying more sedan as compare to suv and hatchback
with low prices however ,
# female are buying sedans with higher prices as compare to male.
sns.catplot(data=df, x= "Make",y="Price", hue= 'Gender',kind="box")
#  it concludes men buy more sedan by the visualisation...
```

Out[63]: <seaborn.axisgrid.FacetGrid at 0x21ef06e2fd0>

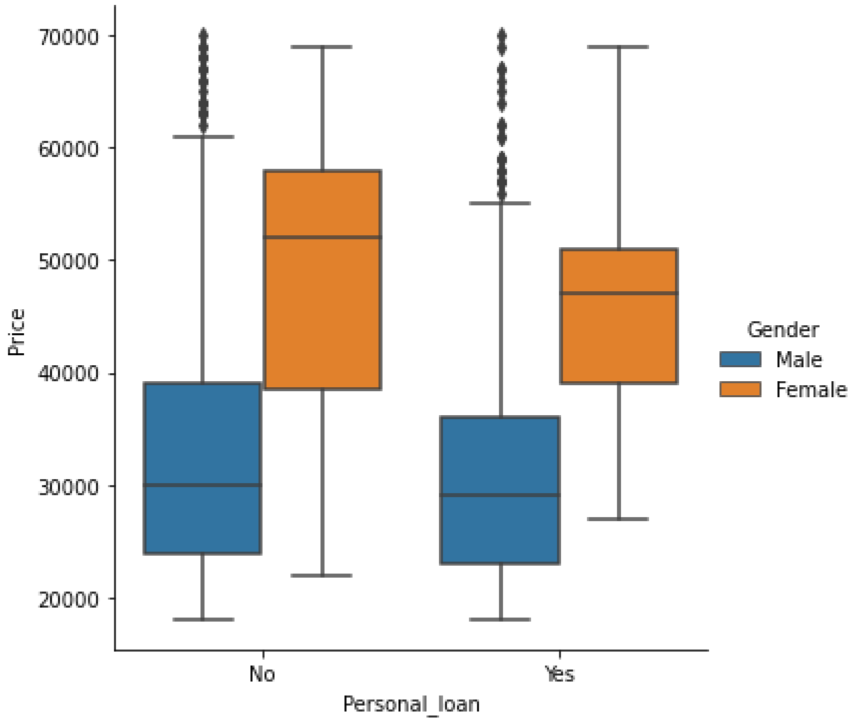


F. From the given data, comment on the amount spent on purchasing automobiles across the following categories. Comment on how a Business can utilize the results from this exercise. Give justification along with presenting metrics/charts used for arriving at the conclusions. Give justification along with presenting metrics/charts used for arriving at the conclusions. F1) Gender F2) Personal_loan

In [64]:
```
# F1 : Gender
# F2 : Personal_loan
# This Boxplot shows how many male and female have taken personal loan,
```

```
male taken more loan while female
# taken less ,however , female buys sedan on higher without taking
personal loan and male buys sedan on
# less price with taking personal loan.
sns.catplot(data=df, x= "Personal_loan",y="Price", hue=
"Gender",kind="box")
# Female buy more priced car when personal loan in not taken
```

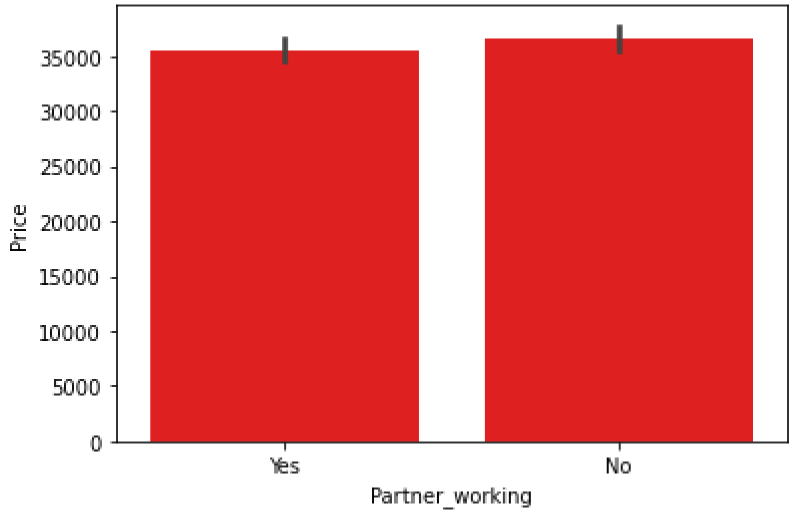
Out[64]: <seaborn.axisgrid.FacetGrid at 0x21ef017a970>



G. From the current data set comment if having a working partner leads to the purchase of a higher-priced car.

In [65]:
```
# This plot shows nothing related between the working partner and the
prices of the cars ,partner is
# working or not prices of the cars does not influenced ,so
# False, if partner is not working , still higher priced cars are also
selling.
sns.barplot(data=df, x= "Partner_working", y="Price", color='r')
```

Out[65]: <AxesSubplot:xlabel='Partner_working', ylabel='Price'>



In [66]:

```
#gender n make groupby with their mean value:
df.groupby(['Gender','Make']).mean()
```

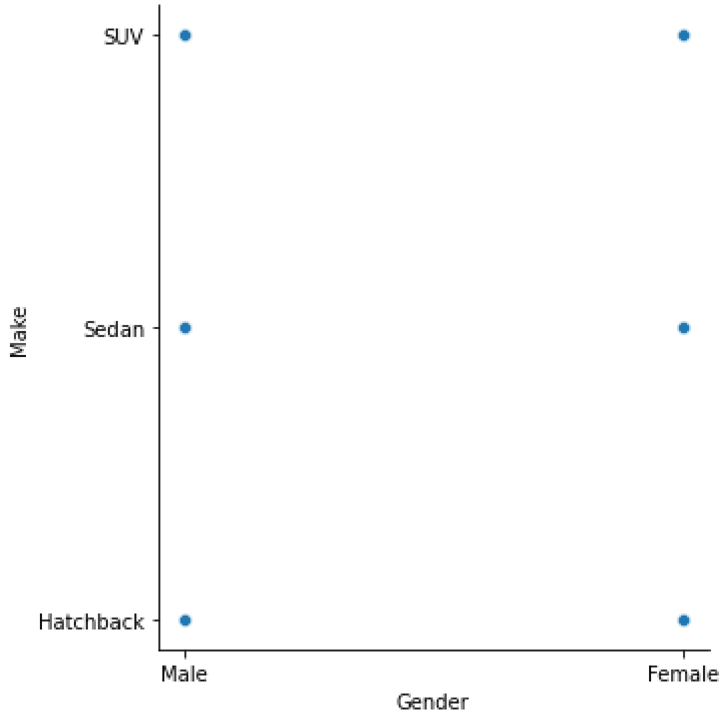
Out[66]:

Gender	Make	Age	No_of_Dependents	Salary	Partner_salary	Total_salary	
Female	**Hatchback**	26.384615	2.923077	58553.846154	19523.076923	78076.923077	27615.38
	SUV	42.950920	2.374233	70485.276074	25292.638037	95777.914110	53159.50
	Sedan	36.930233	1.542636	62393.798450	21154.263566	83548.062016	43139.53
Male	**Hatchback**	25.725528	2.798464	55006.333973	17105.566219	72111.900192	26550.86
	SUV	45.375000	2.937500	75226.785714	25429.464286	100656.250000	59473.21
	Sedan	30.926078	2.254620	59576.180698	21313.347023	80889.527721	33371.66

In [67]:

```
# Draw relatinal plot of gender and make which tells as the male and female who are buying which category
# of cars.
sns.relplot(data= df, x="Gender", y="Make")
```

Out[67]: <seaborn.axisgrid.FacetGrid at 0x21ef1a447f0>



In [68]:
```
# pivot table
pd.pivot_table(data=df, index='Gender',
               columns='Make',
               aggfunc=sum,
               fill_value=0).apply(lambda x: x*100/sum(x))
```

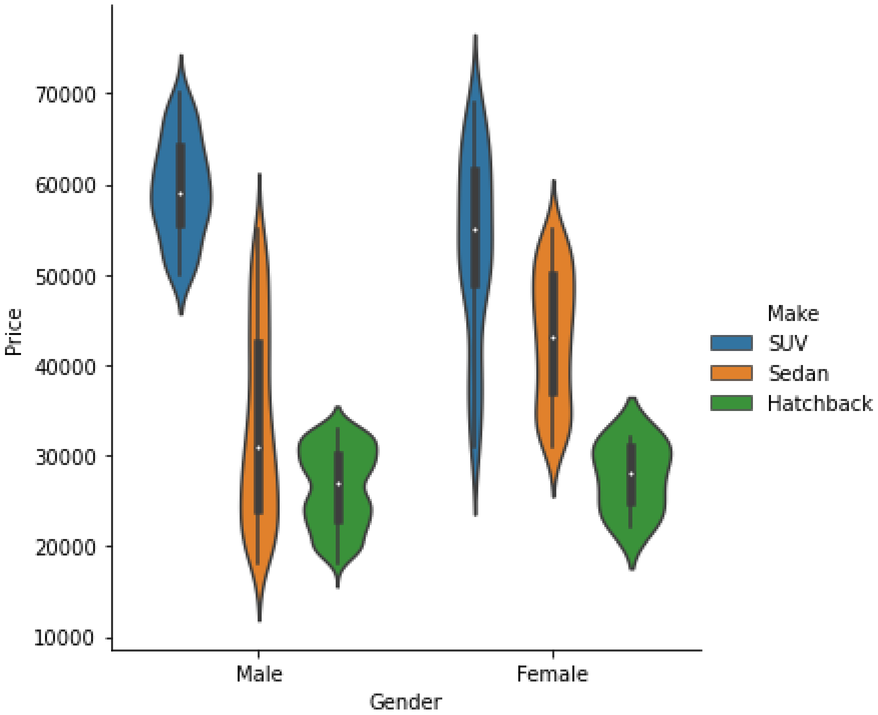
Out[68]:

	Age			No_of_Dependents			Partner_sa		
Make	**Hatchback**	**SUV**	**Sedan**	**Hatchback**	**SUV**	**Sedan**	**Hatchback**	**SUV**	**Se**
Gender									
Female	2.495271	57.940909	24.030265	2.540107	54.050279	15.343099	2.768989	59.142423	20.8
Male	97.504729	42.059091	75.969735	97.459893	45.949721	84.656901	97.231011	40.857577	79.1

other relational v plots below for deeper insights:

In [69]:
```
# Draw a categorical plots which are telling us the how many male and
female buying a which category of
# cars, with higher or lower prices.Which gender are buying which car and
on what prices ?
sns.catplot(data=df, x= 'Gender', y="Price", hue='Make',kind="violin")
```

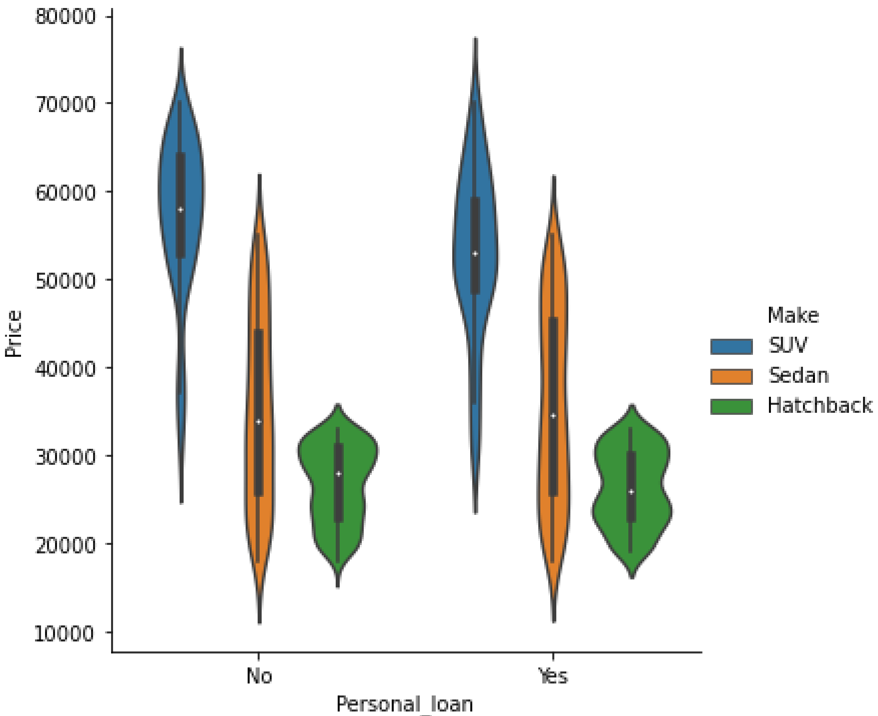
Out[69]: <seaborn.axisgrid.FacetGrid at 0x21ef1c313d0>



In [70]:

```
# This plot shows the personal loan taken by them or not....
sns.catplot(data=df, x= 'Personal_loan', y="Price",
hue='Make',kind="violin")
```

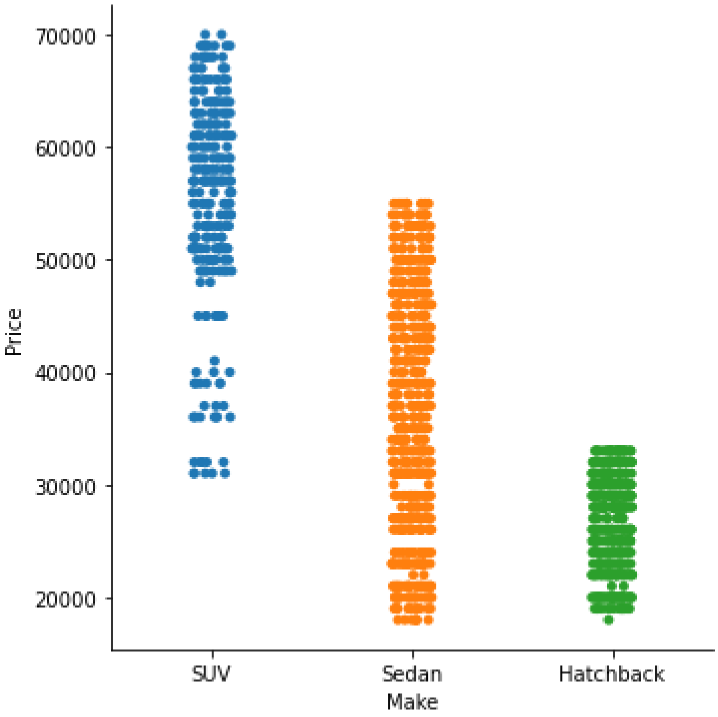
Out[70]: <seaborn.axisgrid.FacetGrid at 0x21ef1cdec70>



In [71]:

```
# This plot shows prices with the categories of cars of column "Make"...
sns.catplot(data=df, x= "Make",y="Price",kind="strip")
```

Out[71]: <seaborn.axisgrid.FacetGrid at 0x21ef1d53dc0>



In [1]:

```
H. The main objective of this analysis is to devise an improved marketing strategy to send
targeted information to different groups of potential buyers present in the data.
For the current analysis use the Gender and Marital_status - fields to arrive at groups
with similar purchase history.
```

In [74]:

```
# pivot table describes the groupby of gender and marital status..
df.groupby(['Gender',"Marital_status"]).describe().unstack()
```

Out[74]:

										A
	count		mean		std		min		2	
Marital_status	**Married**	**Single**	**Married**	**Single**	**Married**	**Single**	**Married**	**Single**	**Married**	**Sin**
Gender										
Female	286.0	19.0	39.643357	40.526316	7.486976	5.059298	22.0	29.0	34.0	3
Male	1022.0	98.0	30.005871	29.387755	7.476110	7.838094	22.0	22.0	25.0	2

2 rows × 96 columns

In [75]:

```
# Pivot table with columns gender, price, marital status, and Make....
table = pd.pivot_table(data=df,index=
['Gender','Price','Marital_status','Make'])
table
```

Out[75]:

Gender	Price	Marital_status	Make	Age	No_of_Dependents	Partner_salary	Salary
Female	**22000**	**Married**	**Hatchback**	27.000000	3.0	28700.000000	66800.000000
	23000	**Married**	**Hatchback**	24.000000	2.5	30600.000000	64300.000000
	25000	**Married**	**Hatchback**	30.000000	3.0	15300.000000	72300.000000
	27000	**Married**	**Hatchback**	23.000000	2.0	32000.000000	59100.000000
	28000	**Married**	**Hatchback**	22.000000	4.0	32900.000000	56600.000000
...	**...**	**...**	**...**	...	...	...	...
Male	**67000**	**Single**	**SUV**	49.000000	2.0	0.000000	77600.000000
	68000	**Married**	**SUV**	44.666667	3.0	26933.333333	79933.333333
		Single	**SUV**	44.000000	2.0	0.000000	70100.000000
	69000	**Married**	**SUV**	40.750000	3.5	40300.000000	67875.000000
	70000	**Married**	**SUV**	42.500000	4.0	20450.000000	69800.000000

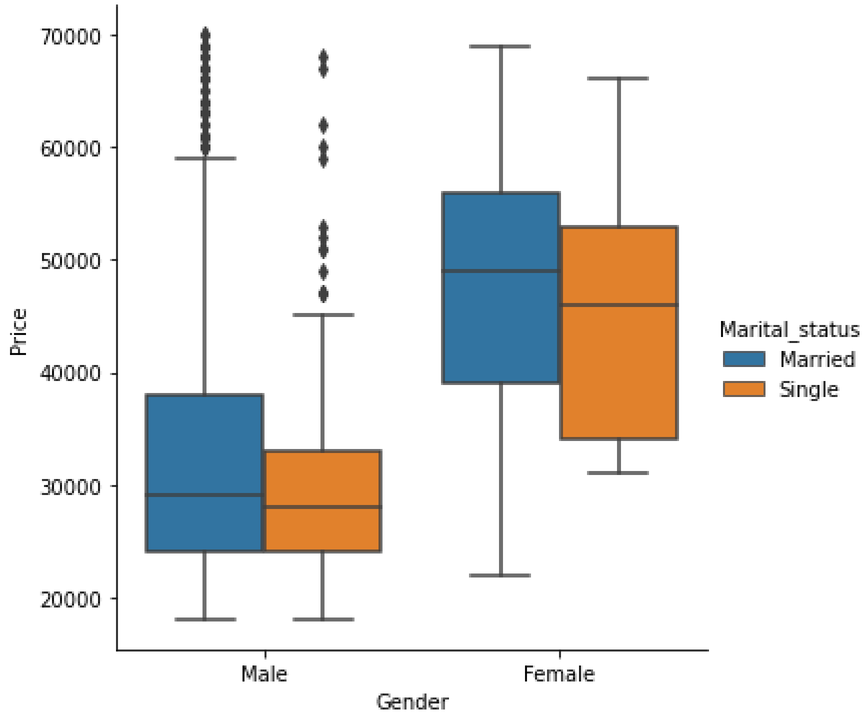
187 rows × 5 columns

#it concludes that gender and marital_status of the customer with respect to price of the cars.

In [76]:

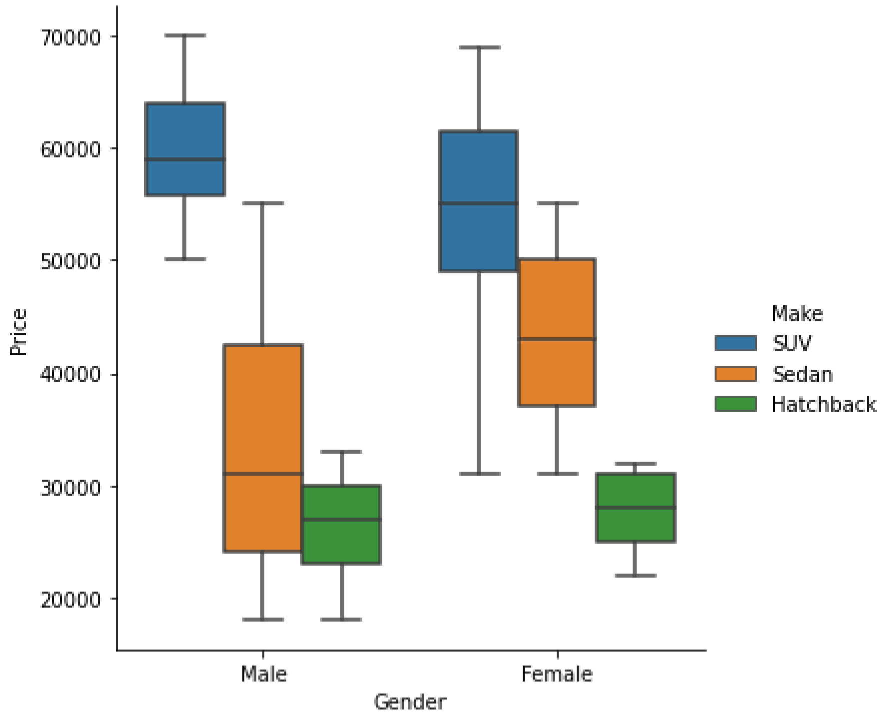
```
# This plot shows the Gender and prices with marital status as hue, which are telling us single female
# are buying cars on high prices however, single males are buying on low price...
sns.catplot(data=df, x= "Gender",y="Price", hue='Marital_status',kind="box")
```

Out[76]: <seaborn.axisgrid.FacetGrid at 0x21ef09d0ca0>



In [77]:
```
# Tghis boxplot shows the gender and price with make column as hue, which are telling us the sedan are
# buying by male and female both as compare to other categories of cars.
sns.catplot(data=df, x= "Gender",y="Price", hue='Make',kind="box")
```

Out[77]: <seaborn.axisgrid.FacetGrid at 0x21ef1c28e80>



conclusion: Sedan cars are selling more the the other cars by both the gender with different prices , personal loans taken by the male or female , but male ratio is higher than male, single male bought sedan category of cars at lower prices while single female bought same car with higher price but have not taken loan, salaried people are more and post graduated bought more cars than graduated poeple, working partner does not effect the prices of the cars.

In [ ]:

```
```