EDA on Vehicle Insurance Data

1. Import library

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as pyplot
```

2. Reading data

In [3]:

```
df1=pd.read_csv('customer_details.csv')
df2=pd.read_csv('customer_policy_details.csv')
```

In [4]:

```
df1.head()
```

Out[4]:

	0	1	2	3	4	5	6	7
0	1	Male	44	1	21	0	<1 yrs	Yes
1	2	Male	34	1	5	1	>2 yrs	No
2	3	Female	23	1	323	0	>3 yrs	Yes
3	4	Male	54	1	225	1	<1 yrs	Yes
4	5	Female	33	1	87	1	<2 yrs	No

In [5]:

```
df2.head()
```

Out[5]:

	0	1	2	3	4
0	1	23600	11	211	1
1	2	34246	11	112	0
2	3	32732	231	232	1
3	4	32754	122	211	0
4	5	24322	122	99	0

In [6]:

```
df1.columns=['customer_id','gender','age','dlp','region code','pi','vehicle a
```

In [7]:
```
df1
```

Out[7]:

	customer_id	gender	age	dlp	region code	pi	vehicle age	vehicle damage
0	1	Male	44	1	21	0	<1 yrs	Yes
1	2	Male	34	1	5	1	>2 yrs	No
2	3	Female	23	1	323	0	>3 yrs	Yes
3	4	Male	54	1	225	1	<1 yrs	Yes
4	5	Female	33	1	87	1	<2 yrs	No

In [8]:
```
df2.columns=['customer_id','annual premium(in rs)','sc code','vintage','respo
```

In [9]:
```
df2
```

Out[9]:

	customer_id	annual premium(in rs)	sc code	vintage	response
0	1	23600	11	211	1
1	2	34246	11	112	0
2	3	32732	231	232	1
3	4	32754	122	211	0
4	5	24322	122	99	0

3. A) Handling Missing data of df1

In [10]:
```
print('null values in customer_id',df1['customer_id'].isnull().sum()) # same
```

```
null values in customer_id 0
```

In [11]:
```
df1.isnull().sum() #null values for df1 in all columns
```

Out[11]:
```
customer_id       0
gender            0
age               0
dlp               0
region code       0
pi                0
vehicle age       0
vehicle damage    0
dtype: int64
```

In [12]:
```
df_ci=df1.dropna(subset=['customer_id'])
```

In [13]:

```
print('null values after dropping null values in customer_id',df1['customer_i
```

```
null values after dropping null values in customer_id 0
```

In [14]:

```
df_ci['gender']=df1['gender'].fillna(df1['gender'].mode()[0])
df_ci['age']=df1['age'].fillna(df1['age'].mean())
df_ci['dlp']=df1['dlp'].fillna(df1['dlp'].mode()[0])
df_ci['region code']=df1['region code'].fillna(df1['region code'].mode()[0])
df_ci['pi']=df1['pi'].fillna(df1['pi'].mode()[0])
df_ci['vehicle age']=df1['vehicle age'].fillna(df1['vehicle age'].mode()[0])
df_ci['vehicle damage']=df1['vehicle damage'].fillna(df1['vehicle damage'].mo
```

3. B) Handling Missing data of df2

In [15]:

```
print('null values in customer_id is',df2['customer_id'].isnull().sum())
```

```
null values in customer_id is 0
```

In [16]:

```
df2.isnull().sum()
```

Out[16]:

```
customer_id             0
annual premium(in rs)   0
sc code                 0
vintage                 0
response                0
dtype: int64
```

In [17]:

```
df_ci2=df2.dropna(subset=['customer_id'])
```

In [18]:

```
df_ci2['annual premium(in rs)']=df2['annual premium(in rs)'].fillna(df2['annu
df_ci2['sc code']=df2['sc code'].fillna(df2['sc code'].mode()[0])
df_ci2['vintage']=df2['vintage'].fillna(df2['vintage'].mean())
df_ci2['response']=df2['response'].fillna(df2['response'].mode()[0])
```

4. Outliers

In [19]:
```
df1.describe()
```

Out[19]:

	customer_id	age	dlp	region code	pi
count	5.000000	5.000000	5.0	5.000000	5.000000
mean	3.000000	37.600000	1.0	132.200000	0.600000
std	1.581139	11.802542	0.0	137.481635	0.547723
min	1.000000	23.000000	1.0	5.000000	0.000000
25%	2.000000	33.000000	1.0	21.000000	0.000000
50%	3.000000	34.000000	1.0	87.000000	1.000000
75%	4.000000	44.000000	1.0	225.000000	1.000000
max	5.000000	54.000000	1.0	323.000000	1.000000

In [20]:
```
q1=df1.describe().loc['25%','age']
q3=df1.describe().loc['75%','age']
```

In [21]:
```
iqr=q3-q1
```

In [22]:
```
hl=q1+1.5*iqr
ll=q1-1.5*iqr
```

In [23]:
```
print('oulier in higher limit:',df1.loc[df1['age']>hl,'age'].count())
```

```
oulier in higher limit: 1
```

In [24]:
```
print('outlier in lower limit:',df1.loc[df1['age']<(ll),'age'].count())
```

```
outlier in lower limit: 0
```

In [25]:
```
df1.loc[df1['age']>(hl),'age']=df1['age'].mean()
```

In [26]:
```
print('outlier after replacing by mean',df1.loc[df1['age']>(hl),'age'].count(
```

```
outlier after replacing by mean 0
```

In [27]:
```
df2.describe()
```

Out[27]:

	customer_id	annual premium(in rs)	sc code	vintage	response
count	5.000000	5.000000	5.000000	5.000000	5.000000
mean	3.000000	29530.800000	99.400000	173.000000	0.400000
std	1.581139	5127.762592	92.153676	62.381888	0.547723
min	1.000000	23600.000000	11.000000	99.000000	0.000000
25%	2.000000	24322.000000	11.000000	112.000000	0.000000
50%	3.000000	32732.000000	122.000000	211.000000	0.000000
75%	4.000000	32754.000000	122.000000	211.000000	1.000000
max	5.000000	34246.000000	231.000000	232.000000	1.000000

In [28]:
```
q1=df2.describe().loc['25%','annual premium(in rs)']
q3=df2.describe().loc['75%','annual premium(in rs)']
```

In [29]:
```
iqr=q3-q1
```

In [30]:
```
hl=q1+1.5*iqr
ll=q1-1.5*iqr
```

In [31]:
```
print('outlier in higher limit:',df2.loc[df2['annual premium(in rs)']>hl,'ann
```

```
outlier in higher limit: 0
```

In [32]:
```
print('outlier in lower limit:',df2.loc[df2['annual premium(in rs)']<ll,'annu
```

```
outlier in lower limit: 0
```

In [33]:
```
q1=df2.describe().loc['25%','vintage']
q3=df2.describe().loc['75%','vintage']
```

In [34]:
```
iqr=q3-q1
```

In [35]:
```
hl=q1+1.5*iqr
ll=q1-1.5*iqr
```

In [36]:
```
print('outlier in higher limit:',df2.loc[df2['vintage']>hl,'vintage'].count()
print('outlier in lower limit:',df2.loc[df2['vintage']<ll,'vintage'].count())
```

```
outlier in higher limit: 0
outlier in lower limit: 0
```

5. Whitespace in df1

In [37]:
```
df1['gender']=df1['gender'].str.strip()
df1['vehicle age']=df1['vehicle age'].str.strip()
df1['vehicle damage']=df1['vehicle damage'].str.strip()
```

6. Case Correction in df1

In [38]:
```
df1['gender']=df1['gender'].str.lower()
df1['vehicle age']=df1['vehicle age'].str.lower()
df1['vehicle damage']=df1['vehicle damage'].str.lower()
```

7. Conversion of categorical data in dummy data

In [39]:
```
gender_dummy=pd.get_dummies(df1['gender'])
vehicle_age_dummy=pd.get_dummies(df1['vehicle age'])
vehicle_damage_dummy=pd.get_dummies(df1['vehicle damage'])
dlp_dummy=pd.get_dummies(df1['dlp'])
region_code_dummy=pd.get_dummies(df1['region code'])
pi_dummy=pd.get_dummies(df1['pi'])
```

In [40]:
```
sc_code_dummy=pd.get_dummies(df2['sc code'])
response_dummy=pd.get_dummies(df2['response'])
```

8. Check Duplicate in df1 and df2

In [41]:
```
print('duplicate table in df1:',df1.duplicated().sum())
```

```
duplicate table in df1: 0
```

In [42]:
```
print('duplicate table in df2:',df2.duplicated().sum())
```

```
duplicate table in df2: 0
```

9. Create a Master table

In [43]:
```
master_df=pd.merge(df1,df2,on='customer_id')
```

In [44]:
```
master_df
```

Out[44]:

	customer_id	gender	age	dlp	region code	pi	vehicle age	vehicle damage	annual premium(in rs)	sc code	vintage	res
0	1	male	44.0	1	21	0	<1 yrs	yes	23600	11	211	
1	2	male	34.0	1	5	1	>2 yrs	no	34246	11	112	
2	3	female	23.0	1	323	0	>3 yrs	yes	32732	231	232	
3	4	male	37.6	1	225	1	<1 yrs	yes	32754	122	211	
4	5	female	33.0	1	87	1	<2 yrs	no	24322	122	99	

10. Average annual premium - genderwise,agewise,vehicleagewise

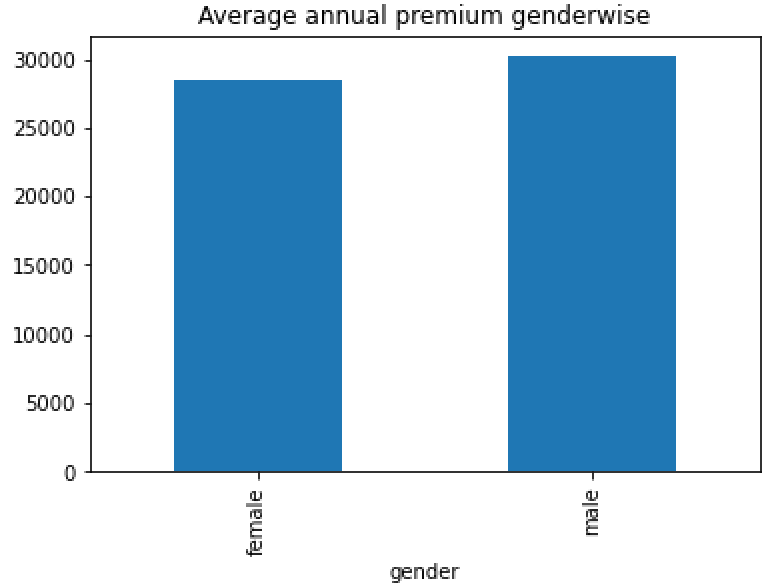
In [45]:
```
aap_gw=master_df.groupby(['gender'])['annual premium(in rs)'].mean()
aap_aw=master_df.groupby(['age'])['annual premium(in rs)'].mean()
aap_vaw=master_df.groupby(['vehicle age'])['annual premium(in rs)'].mean()
```

11. Visualization

In [46]:
```
aap_gw
```

Out[46]:
```
gender
female    28527.0
male      30200.0
Name: annual premium(in rs), dtype: float64
```

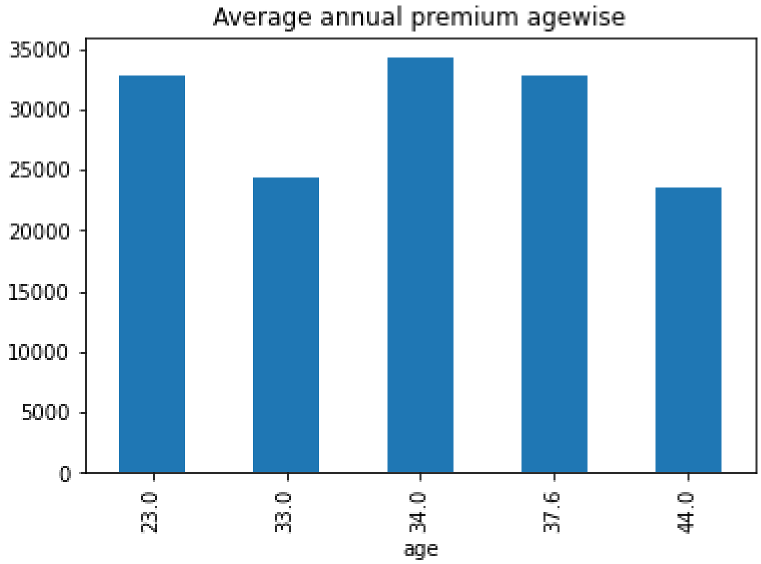
In [62]:
```
pyplot.title('Average annual premium genderwise')
aap_gw.plot.bar() #Average annual premium of male is high then female which i
pyplot.show()
```



In [54]:
```
aap_aw
```

Out[54]:
```
age
23.0    32732.0
33.0    24322.0
34.0    34246.0
37.6    32754.0
44.0    23600.0
Name: annual premium(in rs), dtype: float64
```

In [64]:
```
pyplot.title('Average annual premium agewise')
aap_aw.plot.bar() #Average annual premium at age 34 is high.
pyplot.show()
```



In [65]:

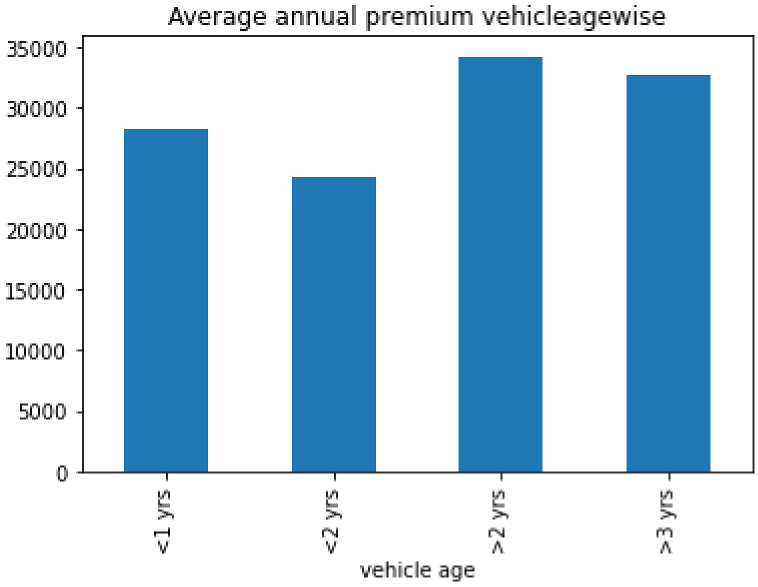
```
aap_vaw
```

Out[65]:

```
vehicle age
<1 yrs    28177.0
<2 yrs    24322.0
>2 yrs    34246.0
>3 yrs    32732.0
Name: annual premium(in rs), dtype: float64
```

In [67]:

```
pyplot.title('Average annual premium vehicleagewise')
aap_vaw.plot.bar() #Average annual premium of vehicleage is high which is gre
pyplot.show()
```



In [68]:

```
correction_coefficient=master_df['age'].corr(master_df['annual premium(in rs)
```

In [69]:

```
n=correction_coefficient
```

In [70]:

```
if n<-0.5:
    print('there is strong positive')
elif n>0.5:
    print('there is strong positive')
else:
    print('there is no relationship')
```

```
there is no relationship
```

In [ ]: