```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
df = pd.read_csv('clean_df.csv')
df.head()
```

	Unnamed: 0	symboling	normalized-losses	make	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	width	height	curb-weight
0	0	3	122	alfa-romero	two	convertible	rwd	front	88.6	0.811148	0.890278	0.816054	2548
1	1	3	122	alfa-romero	two	convertible	rwd	front	88.6	0.811148	0.890278	0.816054	2548
2	2	1	122	alfa-romero	two	hatchback	rwd	front	94.5	0.822681	0.909722	0.876254	2823
3	3	2	164	audi	four	sedan	fwd	front	99.8	0.848630	0.919444	0.908027	2337
4	4	2	164	audi	four	sedan	4wd	front	99.4	0.848630	0.922222	0.908027	2824

```
df.dtypes
```

```
Unnamed: 0             int64
symboling              int64
normalized-losses      int64
make                  object
num-of-doors          object
body-style            object
drive-wheels          object
engine-location       object
wheel-base           float64
```

```
length               float64
width                float64
height               float64
curb-weight            int64
engine-type           object
num-of-cylinders      object
engine-size            int64
fuel-system           object
bore                 float64
stroke               float64
compression-ratio    float64
horsepower           float64
peak-rpm             float64
city-mpg               int64
highway-mpg            int64
price                float64
horsepower-binned     object
diesel                 int64
gas                    int64
aspiration-std         int64
aspiration-turbo       int64
dtype: object
```

```
df.describe()
```

	Unnamed: 0	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore
count	201.000000	201.000000	201.00000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000	201.000000

```
df.describe(include=['object'])
```

	make	num-of-doors	body-style	drive-wheels	engine-location	engine-type	num-of-cylinders	fuel-system	horsepower-binned
count	201	201	201	201	201	201	201	201	200
unique	22	2	5	3	2	6	7	8	3
top	toyota	four	sedan	fwd	front	ohc	four	mpfi	Low
freq	32	115	94	118	198	145	157	92	115

```
df.describe(exclude=[np.number])
```

	make	num-of-doors	body-style	drive-wheels	engine-location	engine-type	num-of-cylinders	fuel-system	horsepower-binned
count	201	201	201	201	201	201	201	201	200
unique	22	2	5	3	2	6	7	8	3
top	toyota	four	sedan	fwd	front	ohc	four	mpfi	Low
freq	32	115	94	118	198	145	157	92	115

```
df.describe(include='all')
```

	Unnamed: 0	symboling	normalized-losses	make	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	length	width	he
count	201.000000	201.000000	201.00000	201	201	201	201	201	201.000000	201.000000	201.000000	201.00
unique	NaN	NaN	NaN	22	2	5	3	2	NaN	NaN	NaN	
top	NaN	NaN	NaN	toyota	four	sedan	fwd	front	NaN	NaN	NaN	
freq	NaN	NaN	NaN	32	115	94	118	198	NaN	NaN	NaN	
mean	100.000000	0.840796	122.00000	NaN	NaN	NaN	NaN	NaN	98.797015	0.837102	0.915126	0.89
std	58.167861	1.254802	31.99625	NaN	NaN	NaN	NaN	NaN	6.066366	0.059213	0.029187	0.04
min	0.000000	-2.000000	65.00000	NaN	NaN	NaN	NaN	NaN	86.600000	0.678039	0.837500	0.79
25%	50.000000	0.000000	101.00000	NaN	NaN	NaN	NaN	NaN	94.500000	0.801538	0.890278	0.86
50%	100.000000	1.000000	122.00000	NaN	NaN	NaN	NaN	NaN	97.000000	0.832292	0.909722	0.90

```
df['drive-wheels'].value_counts()
```

```
fwd    118
rwd     75
4wd      8
Name: drive-wheels, dtype: int64
```

```
drive_wheels_counts = df['drive-wheels'].value_counts().to_frame()
drive_wheels_counts.rename(columns={'drive-wheels': 'value_counts'}, inplace=True)
drive_wheels_counts
```

	value_counts
fwd	118
rwd	75
4wd	8

```
df['drive-wheels'].unique()

     array(['rwd', 'fwd', '4wd'], dtype=object)
```

```
df_group_one=df[['drive-wheels','body-style','price']]
```

Calculate the average price for each of the different categories of data of grouped data

```
df_group_one=df_group_one.groupby(['drive-wheels'],as_index= False).mean()
df_group_one
```

	drive-wheels	price
0	4wd	10241.000000
1	fwd	9244.779661
2	rwd	19757.613333

Group with multiple variables

```
df_gptest=df[['drive-wheels','body-style','price']]
grouped_test1=df_gptest.groupby(['drive-wheels','body-style'],as_index= False).mean()
grouped_test1
```

	drive-wheels	body-style	price
0	4wd	hatchback	7603.000000
1	4wd	sedan	12647.333333
2	4wd	wagon	9095.750000
3	fwd	convertible	11595.000000
4	fwd	hardtop	8249.000000
5	fwd	hatchback	8396.387755
6	fwd	sedan	9811.800000
7	fwd	wagon	9997.333333
8	rwd	convertible	23949.600000

Coversion to Pivot table

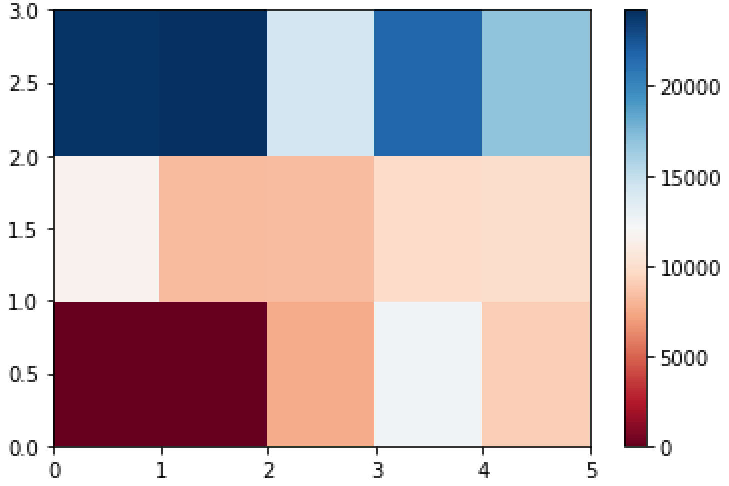
```
grouped_pivot=grouped_test1.pivot(index='drive-wheels',columns='body-style')
grouped_pivot
```

	price				
body-style	convertible	hardtop	hatchback	sedan	wagon
drive-wheels					
4wd	NaN	NaN	7603.000000	12647.333333	9095.750000
fwd	11595.0	8249.000000	8396.387755	9811.800000	9997.333333
rwd	23949.6	24202.714286	14337.777778	21711.833333	16994.222222

```
grouped_pivot=grouped_pivot.fillna(0) #fill missing values with 0
grouped_pivot
```

	price				
body-style	convertible	hardtop	hatchback	sedan	wagon
drive-wheels					
4wd	0.0	0.000000	7603.000000	12647.333333	9095.750000
fwd	11595.0	8249.000000	8396.387755	9811.800000	9997.333333
rwd	23949.6	24202.714286	14337.777778	21711.833333	16994.222222

```
plt.pcolor(grouped_pivot,cmap='RdBu')
plt.colorbar()
plt.show()
```



```
row_labels1=grouped_pivot.columns.levels[1]
row_labels1
```

```
Index(['convertible', 'hardtop', 'hatchback', 'sedan', 'wagon'], dtype='object', name='body-style')
```

```
#Graphical analysis
fig,ax=plt.subplots()
```

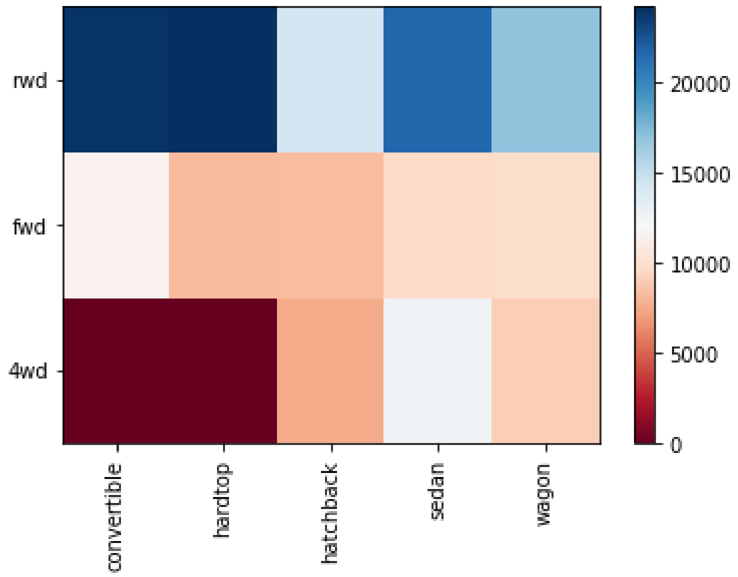
```
im=ax.pcolor(grouped_pivot, cmap='RdBu')

row_labels=grouped_pivot.columns.levels[1]
col_labels=grouped_pivot.index

ax.set_xticks(np.arange(grouped_pivot.shape[1])+0.5, minor=False)            #
ax.set_yticks(np.arange(grouped_pivot.shape[0])+0.5, minor=False)

ax.set_xticklabels(row_labels, minor=False)     #  labelling  xlabel
ax.set_yticklabels(col_labels, minor=False)

plt.xticks(rotation=90)
fig.colorbar(im)
plt.show()
```



Q Use the "groupby" function to find the average "price" of each car based on "body-style" ?

```
df_group_body_price = df[['price', 'body-style']]
```

```
df_group_3 =df_group_body_price.groupby(['body-style'],as_index= False).mean()
df_group_3
```

	body-style	price
0	convertible	21890.500000
1	hardtop	22208.500000
2	hatchback	9957.441176
3	sedan	14459.755319
4	wagon	12371.960000

Continuous numerical variables:

```
df.corr()
```

	Unnamed: 0	symboling	normalized-losses	wheel-base	length	width	height	curb-weight	engine-size	bore	s
Unnamed: 0	1.000000	-0.162764	-0.241092	0.125517	0.161848	0.043976	0.252015	0.064820	-0.047764	0.244734	-0.1
symboling	-0.162764	1.000000	0.466264	-0.535987	-0.365404	-0.242423	-0.550160	-0.233118	-0.110581	-0.140019	-0.0
normalized-losses	-0.241092	0.466264	1.000000	-0.056661	0.019424	0.086802	-0.373737	0.099404	0.112360	-0.029862	0.0
wheel-base	0.125517	-0.535987	-0.056661	1.000000	0.876024	0.814507	0.590742	0.782097	0.572027	0.493244	0.1
length	0.161848	-0.365404	0.019424	0.876024	1.000000	0.857170	0.492063	0.880665	0.685025	0.608971	0.1
width	0.043976	-0.242423	0.086802	0.814507	0.857170	1.000000	0.306002	0.866201	0.729436	0.544885	0.1
height	0.252015	-0.550160	-0.373737	0.590742	0.492063	0.306002	1.000000	0.307581	0.074694	0.180449	-0.0
curb-weight	0.064820	-0.233118	0.099404	0.782097	0.880665	0.866201	0.307581	1.000000	0.849072	0.644060	0.1
engine-size	-0.047764	-0.110581	0.112360	0.572027	0.685025	0.729436	0.074694	0.849072	1.000000	0.572609	0.2
bore	0.244734	-0.140019	-0.029862	0.493244	0.608971	0.544885	0.180449	0.644060	0.572609	1.000000	-0.0
stroke	-0.162490	-0.008153	0.055045	0.158018	0.123952	0.188822	-0.060663	0.167438	0.205928	-0.055390	1.0
compression-ratio	0.144301	-0.182196	-0.114713	0.250313	0.159733	0.189867	0.259737	0.156433	0.028889	0.001263	0.1
horsepower	-0.022474	0.075819	0.217299	0.371147	0.579821	0.615077	-0.087027	0.757976	0.822676	0.566936	0.0

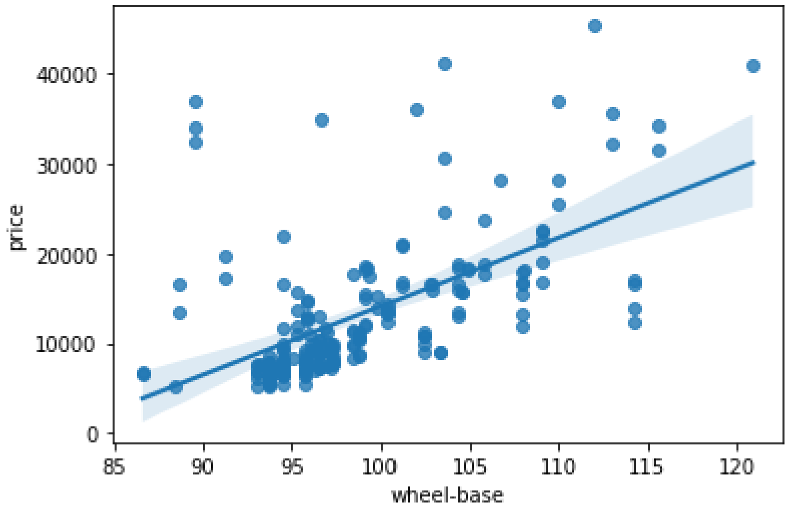
```
df[['wheel-base', 'price']].corr()
```

	wheel-base	price
wheel-base	1.000000	0.584642
price	0.584642	1.000000

diesel	0.121454	-0.196735	-0.101546	0.307237	0.211187	0.244356	0.281578	0.221046	0.070779	0.054458	0.2

```
sns.regplot(x="wheel-base", y="price", data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8ac21263d0>
```



```
df[['city-mpg', 'price']].corr()
```

	city-mpg	price
city-mpg	1.000000	-0.686571
price	-0.686571	1.000000

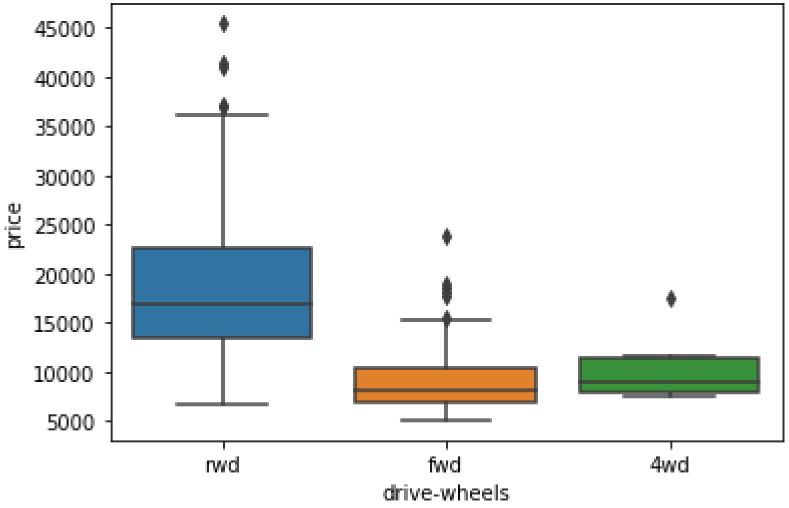
```
sns.regplot(x="city-mpg", y="price", data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8ac20959d0>
```

Categorical variables

```
sns.boxplot(x="drive-wheels", y="price", data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8ac1b10990>
```


price
rwd
fwd
4wd
drive-wheels