

- ▼ Import Libraries to perform specific task such as data manipulation, data analysis, numerical computations, creating various types of plots, statistical visualizations of plots.

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

- ▼ Load Data (Customer address) and save in variable callad "ca".

```
ca = pd.read_csv("/content/Customer_address.csv")
ca
```

	customer_id	address	postcode	state	country	property_valuation	
0	1	060 Morning Avenue	2016	New South Wales	Australia	10	
1	2	6 Meadow Vale Court	2153	New South Wales	Australia	10	
2	4	0 Holy Cross Court	4211	QLD	Australia	9	
3	5	17979 Del Mar Point	2448	New South Wales	Australia	4	
4	6	9 Oakridge Court	3216	VIC	Australia	9	
...	
3994	3999	1482 Hauk Trail	3064	VIC	Australia	3	

- ▼ Some basic Operations.



```
# Following command gives whole information of given dataset.
```

```
ca.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3999 entries, 0 to 3998
Data columns (total 6 columns):
#   Column                Non-Null Count  Dtype
---  -
0   customer_id           3999 non-null   int64
1   address               3999 non-null   object
2   postcode              3999 non-null   int64
3   state                 3999 non-null   object
4   country               3999 non-null   object
5   property_valuation    3999 non-null   int64
dtypes: int64(3), object(3)
memory usage: 187.6+ KB
```



```
# Following command gives Statistical information for only integer column of given dataset.
```

```
ca.describe()
```

	customer_id	postcode	property_valuation	
count	3999.000000	3999.000000	3999.000000	
mean	2003.987997	2985.755939	7.514379	
std	1154.576912	844.878364	2.824663	
min	1.000000	2000.000000	1.000000	
25%	1004.500000	2200.000000	6.000000	
50%	2004.000000	2768.000000	8.000000	
75%	3003.500000	3750.000000	10.000000	
max	4003.000000	4883.000000	12.000000	


Following command gives first 5 columns of given dataset.

```
ca.head()
```

	customer_id	address	postcode	state	country	property_valuation	
0	1	060 Morning Avenue	2016	New South Wales	Australia	10	
1	2	6 Meadow Vale Court	2153	New South Wales	Australia	10	
2	4	0 Holy Cross Court	4211	QLD	Australia	9	

Following command gives last 5 columns of given dataset.

```
ca.tail()
```

	customer_id	address	postcode	state	country	property_valuation	
3994	3999	1482 Hauk Trail	3064	VIC	Australia	3	
3995	4000	57042 Village Green Point	4511	QLD	Australia	6	
3996	4001	87 Crescent Oaks Alley	2756	NSW	Australia	10	

Following command calculate missing values for all columns of given dataset.

```
ca.isnull().sum()
```

```
customer_id      0
address          0
postcode         0
state            0
country          0
property_valuation  0
dtype: int64
```

Following command calculate if there is any duplicate values in given dataset.

```
ca.duplicated().sum()
```

```
0
```

Following command gives the name for all columns of given dataset.

```
ca.columns
```

```
Index(['customer_id', 'address', 'postcode', 'state', 'country',
      'property_valuation'],
      dtype='object')
```

Following command caluculate order pair of given dataset.

```
ca.shape
```

```
(3999, 6)
```

▼ Finding and replacing values

Following command calculate values of "country" column of given dataset.

```
ca.country.value_counts()
```

```
Australia      3999
Name: country, dtype: int64
```

Following command calculate values of "state" column of given dataset.
Before replacing

```
state = ca.groupby(['state'])
state.size()
```

```
state
NSW      2054
New South Wales    86
QLD      838
VIC      939
Victoria    82
dtype: int64
```

```
# Following command replace values in "state" column of given dataset.

ca['state'].replace({'NSW': 'New South Wales', 'VIC': 'Victoria', 'QLD': 'Queensland'}, inplace=True)
```

```
# Following command calculate values of "state" column of given dataset.
# After replacing

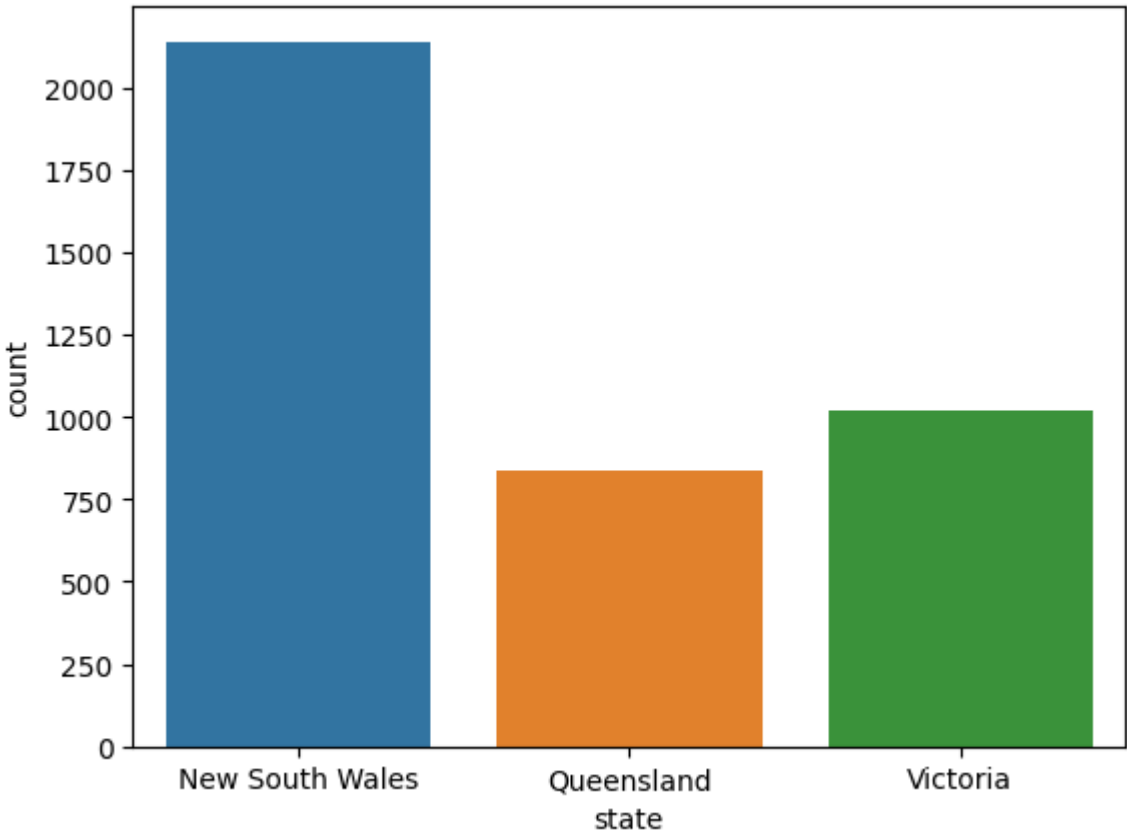
state = ca.groupby(['state'])
state.size()
```

```
state
New South Wales    2140
Queensland         838
Victoria          1021
dtype: int64
```

▼ Graphical View

```
# Following command plot "State" column of given dataset.

sns.countplot(x = 'state', data = ca)
plt.xlabel('state')
plt.show()
```



```
# Following command calculate values for "Property_Valuation" column of given dataset.

ca.property_valuation.value_counts()
```

```
9      647
8      646
10     577
7      493
11     281
6      238
5      225
4      214
12     195
3      186
1      154
2      143
Name: property_valuation, dtype: int64
```

```
# Following command Plot the relationship between two columns of given dataset.

sns.catplot( x= "state",y = "property_valuation", data = ca, kind = "box", aspect = 1.5)
plt.title("Relationship between State of Australia and Property_valuation")
plt.show()
```



Following command plot "property_valuation" column of given dataset.

```
plt.boxplot(ca['property_valuation'])
plt.title("Box Plot of property_valuation")
plt.show()
```

