### Import Libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

### Import CSV File of Transcation

```
Trans = pd.read_csv("/content/KPMG_Transcation.csv")
Trans
```

| tion_date | online_order | order_status | brand | product_line | product_class | product_size | list_ |
|---|---|---|---|---|---|---|---|
| 25/02/2017 | False | Approved | Solex | Standard | medium | medium | |
| 21/05/2017 | True | Approved | Trek Bicycles | Standard | medium | large | 20 |
| 16/10/2017 | False | Approved | OHM Cycles | Standard | low | medium | 17 |
| 31/08/2017 | False | Approved | Norco Bicycles | Standard | medium | medium | 11 |
| 1/10/2017 | True | Approved | Giant Bicycles | Standard | medium | large | 17 |
| ... | ... | ... | ... | ... | ... | ... | |
| 24/06/2017 | True | Approved | OHM Cycles | Standard | high | medium | 20 |
| 9/11/2017 | True | Approved | Solex | Road | medium | medium | 4 |
| 14/04/2017 | True | Approved | OHM Cycles | Standard | medium | medium | 16 |
| 3/7/2017 | False | Approved | OHM Cycles | Standard | high | medium | 2 |
| 22/09/2017 | True | Approved | Trek Bicycles | Standard | medium | small | 17 |

### Data Describe

```
Trans.describe()
```

| | transaction_id | product_id | customer_id | list_price | product_first_sold_date |
|---|---|---|---|---|---|
| **count** | 20000.000000 | 20000.00000 | 20000.000000 | 20000.000000 | 19803.000000 |
| **mean** | 10000.500000 | 45.36465 | 1738.246050 | 1107.829449 | 38199.776549 |
| **std** | 5773.647028 | 30.75359 | 1011.951046 | 582.825242 | 2875.201110 |
| **min** | 1.000000 | 0.00000 | 1.000000 | 12.010000 | 33259.000000 |
| **25%** | 5000.750000 | 18.00000 | 857.750000 | 575.270000 | 35667.000000 |
| **50%** | 10000.500000 | 44.00000 | 1736.000000 | 1163.890000 | 38216.000000 |
| **75%** | 15000.250000 | 72.00000 | 2613.000000 | 1635.300000 | 40672.000000 |
| **max** | 20000.000000 | 100.00000 | 5034.000000 | 2091.470000 | 42710.000000 |

### Data Info

```
Trans.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 13 columns):
 #   Column                  Non-Null Count  Dtype
```

```
---  ------                    --------------  -----
 0   transaction_id            20000 non-null  int64
 1   product_id                20000 non-null  int64
 2   customer_id               20000 non-null  int64
 3   transaction_date          20000 non-null  object
 4   online_order              19640 non-null  object
 5   order_status              20000 non-null  object
 6   brand                     19803 non-null  object
 7   product_line              19803 non-null  object
 8   product_class             19803 non-null  object
 9   product_size              19803 non-null  object
 10  list_price                20000 non-null  float64
 11  standard_cost             19803 non-null  object
 12  product_first_sold_date   19803 non-null  float64
dtypes: float64(2), int64(3), object(8)
memory usage: 2.0+ MB
```

### Data Isnull value check

```
Trans.isnull().sum()
```

```
transaction_id               0
product_id                   0
customer_id                  0
transaction_date             0
online_order               360
order_status                 0
brand                      197
product_line               197
product_class              197
product_size               197
list_price                   0
standard_cost              197
product_first_sold_date    197
dtype: int64
```

### Data Missiing Values Check

```
missing_values = Trans.isnull().sum()
print("Missing Values:\n", missing_values)
```

```
Missing Values:
 transaction_id               0
product_id                   0
customer_id                  0
transaction_date             0
online_order               360
order_status                 0
brand                      197
product_line               197
product_class              197
product_size               197
list_price                   0
standard_cost              197
product_first_sold_date    197
dtype: int64
```

### Data Dropna Value from Missing Values

```
Trans = Trans.dropna()
```

### Again Check Isnull Values of Data

```
Trans.isnull().sum()
```

```
transaction_id             0
product_id                 0
customer_id                0
transaction_date           0
online_order               0
order_status               0
brand                      0
product_line               0
product_class              0
product_size               0
list_price                 0
standard_cost              0
product_first_sold_date    0
dtype: int64
```

### Data Information

```
Trans.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 19445 entries, 0 to 19999
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   transaction_id           19445 non-null  int64  
 1   product_id               19445 non-null  int64  
 2   customer_id              19445 non-null  int64  
 3   transaction_date         19445 non-null  object 
 4   online_order             19445 non-null  object 
 5   order_status             19445 non-null  object 
 6   brand                    19445 non-null  object 
 7   product_line             19445 non-null  object 
 8   product_class            19445 non-null  object 
 9   product_size             19445 non-null  object 
 10  list_price               19445 non-null  float64
 11  standard_cost            19445 non-null  object 
 12  product_first_sold_date  19445 non-null  float64
dtypes: float64(2), int64(3), object(8)
memory usage: 2.1+ MB
```

### Print Columns From Data

```
columns = Trans.columns
print(columns)
```

```
Index(['transaction_id', 'product_id', 'customer_id', 'transaction_date',
       'online_order', 'order_status', 'brand', 'product_line',
       'product_class', 'product_size', 'list_price', 'standard_cost',
       'product_first_sold_date'],
      dtype='object')
```

### Check Duplicate Values

```
Trans.duplicated().sum()
```

```
0
```

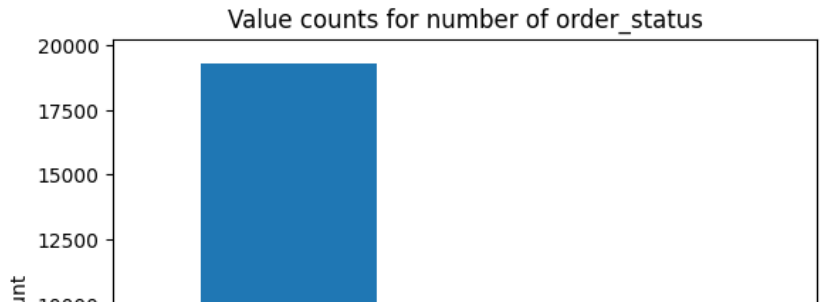
### GroupBy Data Order Status

```
order_status = Trans.groupby(['order_status'])
order_status.size()
```

```
order_status
Approved     19273
Cancelled      172
dtype: int64
```

### Check Bar Chart by Order Status

```
Trans.order_status.value_counts().plot(kind= "bar")
plt.title("Value counts for number of order_status")
plt.xlabel("Order_Status")
plt.xticks(rotation = 2)
plt.ylabel("Count")
plt.show()
```



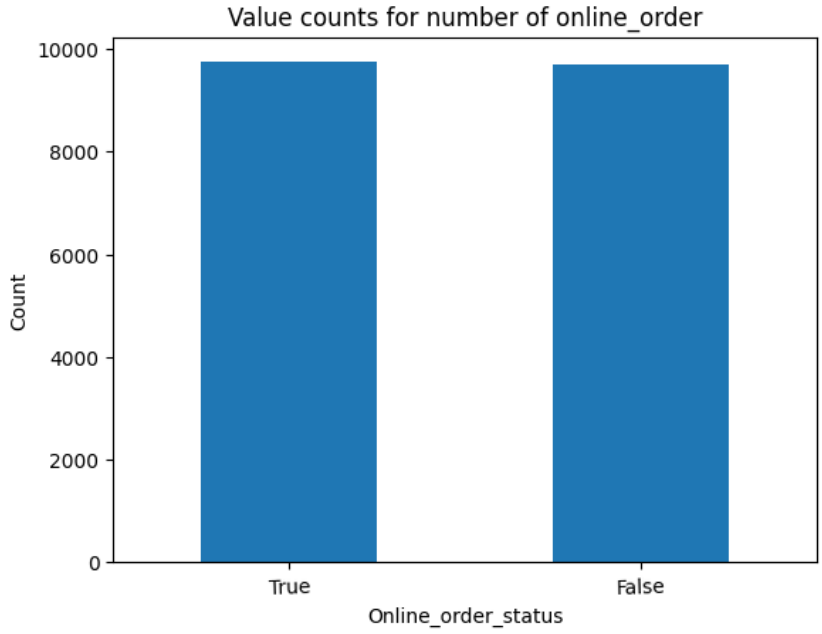

### Check Online Order by GroupBy

```
online_order = Trans.groupby(['online_order'])
online_order.size()
```

```
online_order
False    9706
True     9739
dtype: int64
```

### Make Online order status by Bar Chart

```
Trans.online_order.value_counts().plot(kind= "bar")
plt.title("Value counts for number of online_order")
plt.xlabel("Online_order_status")
plt.xticks(rotation =2)
plt.ylabel("Count")
plt.show()
```




### Product Line by GroupBy

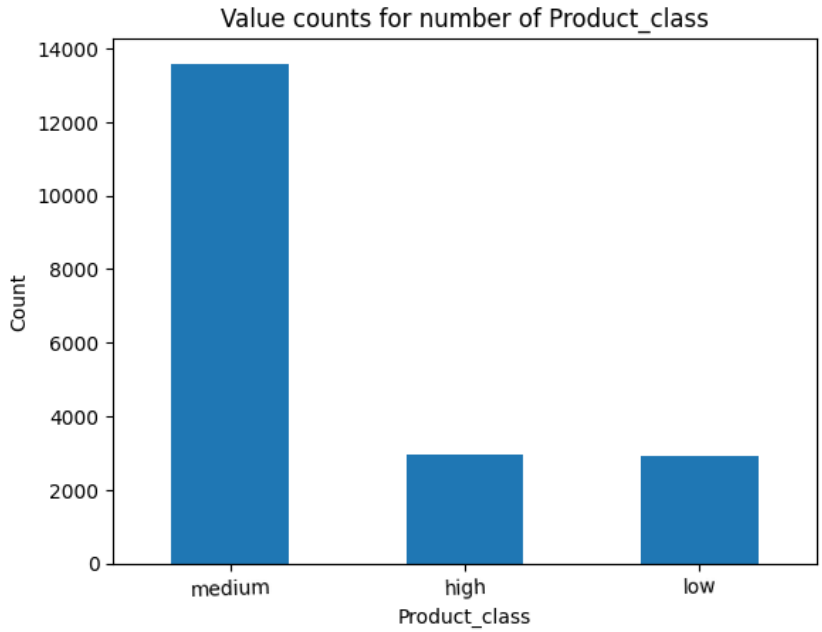
```
product_line = Trans.groupby(['product_line'])
product_line.size()
```

```
product_line
Mountain      418
Road         3894
Standard    13920
Touring      1213
dtype: int64
```

### Check Product Class by Bar Chart

```
Trans.product_class.value_counts().plot(kind ="bar")
plt.title("Value counts for number of Product_class")
plt.xlabel("Product_class")
plt.xticks(rotation = 2)
```

```
plt.ylabel("Count")
plt.show()
```



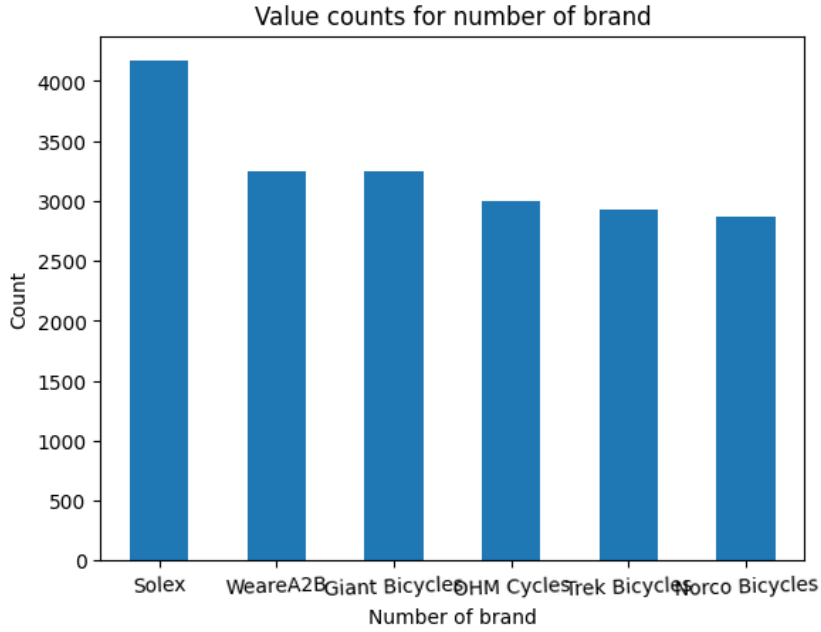

### GroupBy Brand

```
brand = Trans.groupby(['brand'])
brand.size()
```

```
brand
Giant Bicycles    3244
Norco Bicycles    2863
OHM Cycles        2993
Solex             4169
Trek Bicycles     2931
WeareA2B          3245
dtype: int64
```

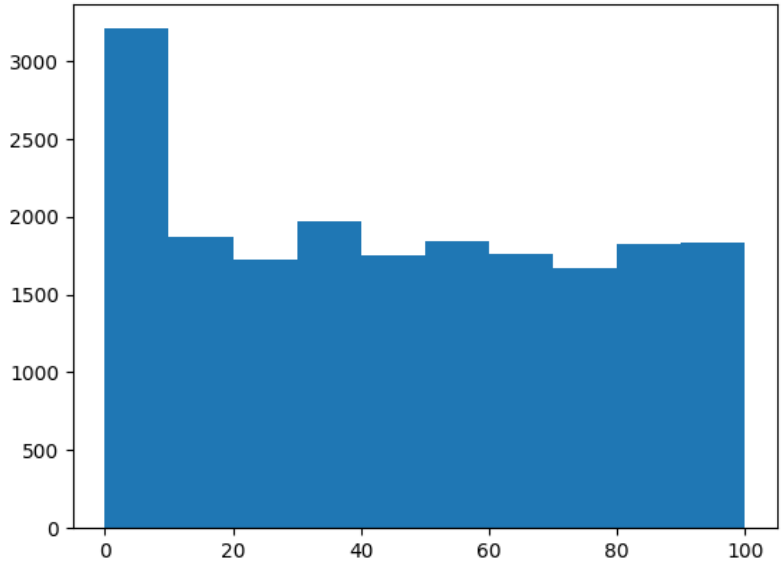
### Check Number of Brand by Bar Chart

```
Trans.brand.value_counts().plot(kind="bar")
plt. title("Value counts for number of brand")
plt.xlabel("Number of brand")
plt.xticks(rotation = 2)
plt.ylabel("Count")
plt.show()
```



### Check Prodcuts with Hist chart

```
plt.hist(Trans['product_id'], bins = 10)
plt.show()
```




### Remove A Dollar Sign '$'

```
# Remove dollar signs ($) from all columns
Trans['standard_cost'] = Trans['standard_cost'].str.replace('$', '', regex=False)
Trans
```

```
<ipython-input-30-9e8776c0aa73>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing
  Trans['standard_cost'] = Trans['standard_cost'].str.replace('$', '', regex=False)
```

| | transaction_id | product_id | customer_id | transaction_date | online_order | order_status | brand |
|---|---|---|---|---|---|---|---|
| **0** | 1 | 2 | 2950 | 25/02/2017 | False | Approved | Solex |
| **1** | 2 | 3 | 3120 | 21/05/2017 | True | Approved | Trek Bicycles |
| **2** | 3 | 37 | 402 | 16/10/2017 | False | Approved | OHM Cycles |
| **3** | 4 | 88 | 3135 | 31/08/2017 | False | Approved | Norco Bicycles |
| **4** | 5 | 78 | 787 | 1/10/2017 | True | Approved | Giant Bicycles |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **19995** | 19996 | 51 | 1018 | 24/06/2017 | True | Approved | OHM Cycles |
| **19996** | 19997 | 41 | 127 | 9/11/2017 | True | Approved | Solex |
| **19997** | 19998 | 87 | 2284 | 14/04/2017 | True | Approved | OHM Cycles |
| **19998** | 19999 | 6 | 2764 | 3/7/2017 | False | Approved | OHM Cycles |
| **19999** | 20000 | 11 | 1144 | 22/09/2017 | True | Approved | Trek Bicycles |

19445 rows × 13 columns