

```
#Ecommerce data set
```

```
ecommerce = pd.read_csv("/content/Ecommerce Customers.csv")
ecommerce
```



	Email	Address	Avatar	Avg. Session Length	Time
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.6554
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.1094
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.3304
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.7174
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.7954
...	...	...	...	...	...
495	lewisjessica@craig-evans.com	4483 Jones Motorway Suite 872\nLake Jamiefurt,...	Tan	33.237660	13.5664
496	katrina56@gmail.com	172 Owen Divide Suite 497\nWest Richard, CA 19320	PaleVioletRed	34.702529	11.6954

```
ecommerce.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Email                  500 non-null   object
1   Address                500 non-null   object
2   Avatar                 500 non-null   object
3   Avg. Session Length    500 non-null   float64
4   Time on App            500 non-null   float64
5   Time on Website        500 non-null   float64
6   Length of Membership    500 non-null   float64
7   Yearly Amount Spent    500 non-null   float64
dtypes: float64(5), object(3)
memory usage: 31.4+ KB
```

```
ecommerce.columns
```

```
Index(['Email', 'Address', 'Avatar', 'Avg. Session Length', 'Time on App',
       'Time on Website', 'Length of Membership', 'Yearly Amount Spent'],
      dtype='object')
```

```
# to know the numeric columns
```

```
num_col = ecommerce.select_dtypes(exclude= ["object"])
num_col.columns
```

```
Index(['Avg. Session Length', 'Time on App', 'Time on Website',
       'Length of Membership', 'Yearly Amount Spent'],
      dtype='object')
```

```
# to know oject columns
```

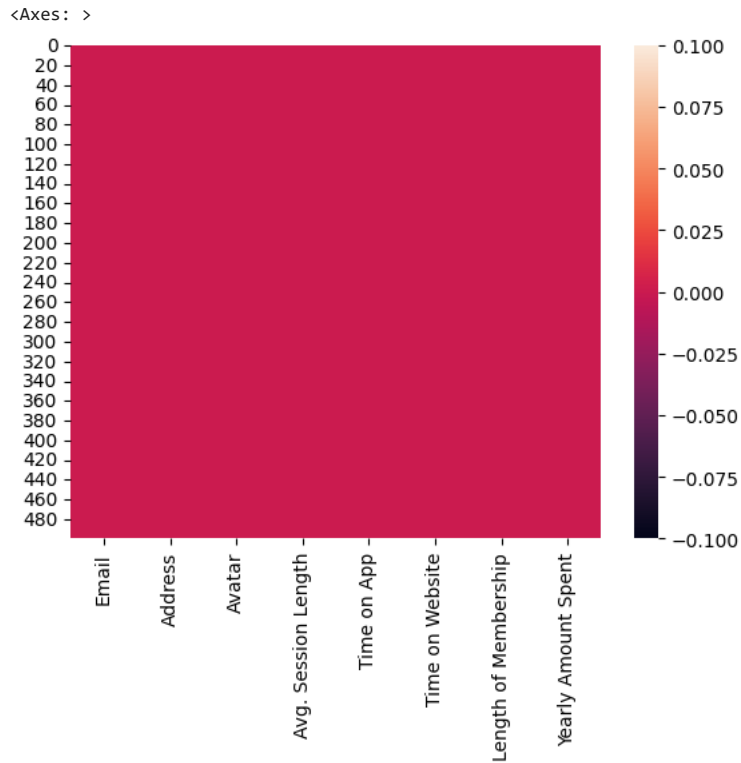
```
non_num_col = ecommerce.select_dtypes(include = ["object"])
non_num_col.columns
```

```
Index(['Email', 'Address', 'Avatar'], dtype='object')
```

```
#to know it is a categorical or not
non_num_col.nunique()
```

```
Email      500
Address     500
Avatar      138
dtype: int64
```

```
sns.heatmap(ecommerce.isnull())
```



```
ecommerce.isnull().sum()
```

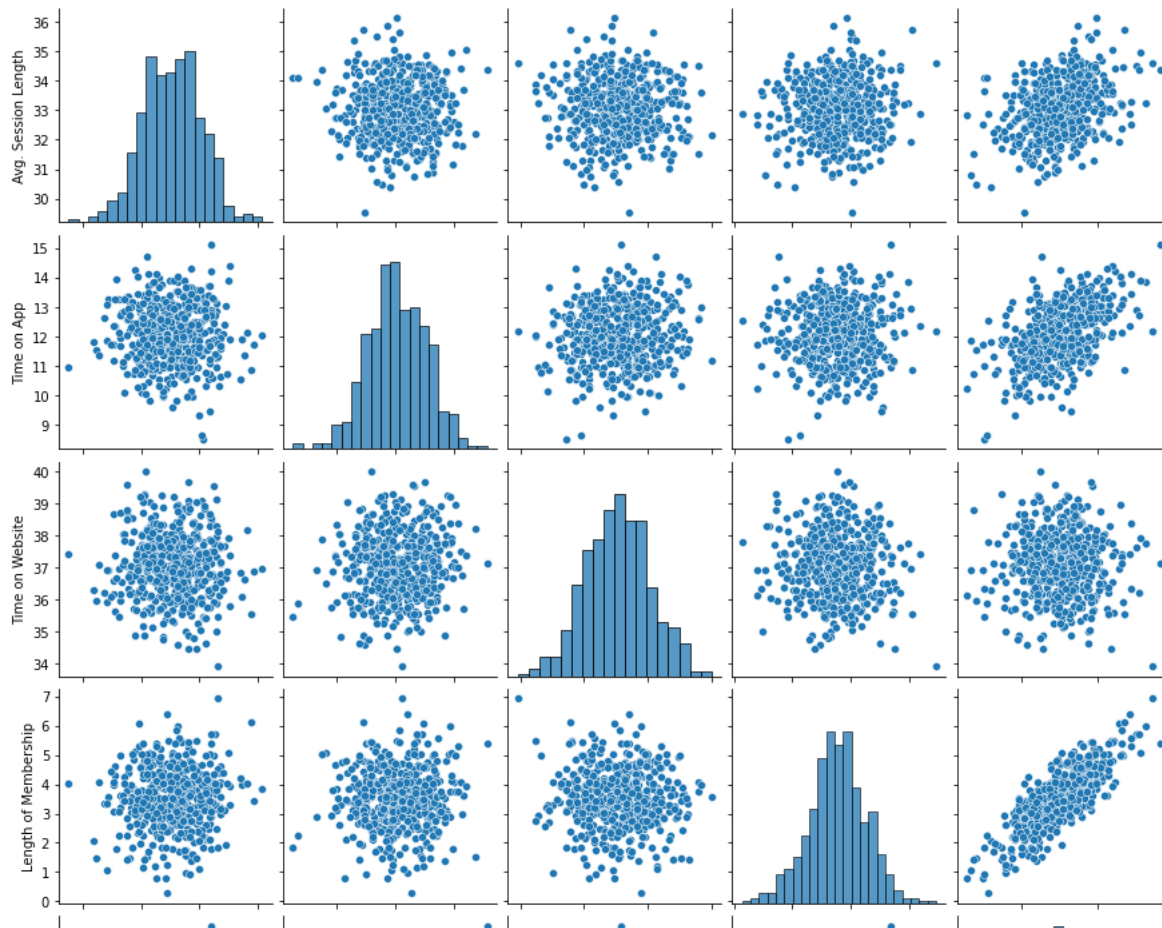
```
Email      0
Address     0
Avatar      0
Avg. Session Length  0
Time on App  0
Time on Website  0
Length of Membership  0
Yearly Amount Spent  0
dtype: int64
```

```
import pandas as pd
import matplotlib.pyplot as plt

import seaborn as sns
%matplotlib inline
```

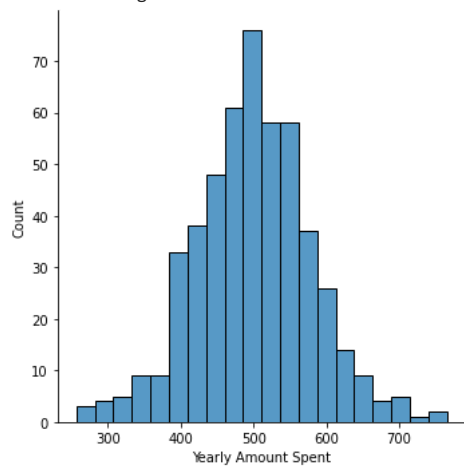
```
sns.pairplot(ecommerce)
```

<seaborn.axisgrid.PairGrid at 0x7f1b5bf49670>



```
sns.displot(ecommerce['Yearly Amount Spent'])
```

<seaborn.axisgrid.FacetGrid at 0x7f1b6afc7730>



```
X = ecommerce[['Avg. Session Length' , 'Time on App','Time on Website','Length of Membership']]
Y = ecommerce['Yearly Amount Spent']
X
Y
```

```
0      587.951054
1      392.204933
2      487.547505
3      581.852344
4      599.406092
...
495     573.847438
496     529.049004
497     551.620145
498     456.469510
```

499 497.778642  
Name: Yearly Amount Spent, Length: 500, dtype: float64

```
import sklearn
from sklearn.model_selection import (train_test_split)
X_train , X_test , Y_train , Y_test = train_test_split(X,Y,test_size = 0.4)
X_train
```

	Avg. Session Length	Time on App	Time on Website	Length of Membership
189	32.200799	12.276982	38.232606	3.316465
47	32.726785	12.988510	36.462003	4.113226
370	33.705113	10.163179	37.763041	4.778974
70	33.154176	11.887494	36.265001	2.602287
385	33.236266	10.972554	34.574028	2.931620
...	...	...	...	...
408	33.000850	11.230743	36.995290	3.781704
206	32.295248	11.031358	38.252978	3.107469
333	33.485520	11.887345	35.862447	3.206757
350	33.304337	12.692661	37.333591	3.827376
229	33.946241	10.983977	37.951489	3.050713

300 rows × 4 columns

Y\_train

351 533.396554  
418 475.725068  
274 446.418673  
300 473.949857  
414 532.724805  
...  
354 508.771907  
415 275.918421  
390 546.556667  
430 556.186369  
290 503.217393

Name: Yearly Amount Spent, Length: 300, dtype: float64

X\_test

	Avg. Session Length	Time on App	Time on Website	Length of Membership
417	32.215527	12.216855	36.953960	2.910531
199	30.836433	13.100110	35.907721	3.361613
58	32.843930	11.832286	36.814011	3.471919
433	34.278248	11.822722	36.308545	2.117383
121	33.701605	11.564022	37.673210	4.716105
...	...	...	...	...
88	31.956301	12.828893	36.951617	4.571213
349	32.008505	12.095889	36.377509	3.178952
416	33.454302	11.016756	37.637311	4.137000
355	33.344509	10.969803	35.974578	2.627625
152	32.510218	10.984836	37.396497	5.391275

200 rows × 4 columns

Y\_test

417 438.417742  
199 467.501900  
58 496.650708

```

433    434.144202
121    576.802547
...
88     547.125932
349    443.197221
416    511.038786
355    403.766902
152    555.892595
Name: Yearly Amount Spent, Length: 200, dtype: float64

```

```

from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(X_train , Y_train)

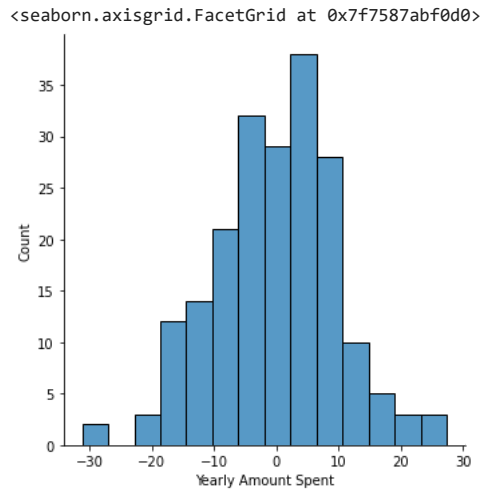
```

```
LinearRegression()
```

```

#evaluation of model
prediction = lm.predict(X_test)
sns.displot(prediction - Y_test)

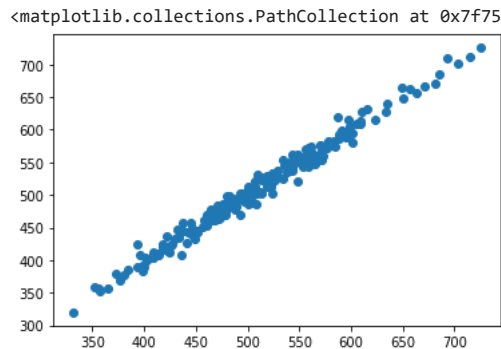
```



```

#evaluation of model
prediction = lm.predict(X_test)
plt.scatter(prediction , Y_test)

```



```

coeff = pd.DataFrame(
lm.coef_ , X.columns)
coeff

```

	0
Avg. Session Length	25.510207
Time on App	38.361057
Time on Website	0.684656
Length of Membership	61.729212

```
from sklearn import metrics
print(metrics.mean_absolute_error(prediction , Y_test))
```

7.797508251452291

```
from sklearn import metrics
print(metrics.mean_squared_error(Y_test , prediction))
```

97.49237965378856

```
import numpy as np
print(np.sqrt(metrics.mean_squared_error(Y_test , prediction)))
```

9.873822950295825