Linear Regression Project

CaseStudy -: You just got some contract work with an Ecommerce company based in New York City that sells clothing online but they also have in-store style and clothing advice sessions. Customers come in to the store, have sessions/meetings with a personal stylist, then they can go home and order either on a mobile app or website for the clothes they want.

The company is trying to decide whether to focus their efforts on their mobile app experience or their website. They've hired you on contract to help them figure it out! Let's get started!

Just follow the steps below to analyze the customer data (it's fake, don't worry I didn't give you real credit card numbers or emails).

Imports

Q.Import pandas, numpy, matplotlib,and seaborn. Then set %matplotlib inline (You'll import sklearn as you need it.)

In [1]:

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

Get the Data

We'll work with the Ecommerce Customers csv file from the company. It has Customer info, suchas Email, Address, and their color Avatar. Then it also has numerical value columns:

- Avg. Session Length: Average session of in-store style advice sessions.
- Time on App: Average time spent on App in minutes
- Time on Website: Average time spent on Website in minutes
- Length of Membership: How many years the customer has been a member.

Q.Read in the Ecommerce Customers csv file as a DataFrame called customers.

In [2]:
```
df=pd.read_csv('Ecommerce Customers')
df.head()
```

Out[2]:

	Email	Address	Avatar	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655651	39.577668	4.082621	587.951054
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109461	37.268959	2.664034	392.204933
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.330278	37.110597	4.104543	487.547505
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.717514	36.721283	3.120179	581.852344
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.795189	37.536653	4.446308	599.406092

In [3]:
```

```

Q.Check the head of customers, and check out its info() and describe() methods.

In [4]:
```
df.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
```

In [5]:

```

```

In [6]:

```
df.describe()
```

Out[6]:

	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
count	500.000000	500.000000	500.000000	500.000000	500.000000
mean	33.053194	12.052488	37.060445	3.533462	499.314038
std	0.992563	0.994216	1.010489	0.999278	79.314782
min	29.532429	8.508152	33.913847	0.269901	256.670582
25%	32.341822	11.388153	36.349257	2.930450	445.038277
50%	33.082008	11.983231	37.069367	3.533975	498.887875
75%	33.711985	12.753850	37.716432	4.126502	549.313828
max	36.139662	15.126994	40.005182	6.922689	765.518462

In [7]:

```

```

Exploratory Data Analysis

Let's explore the data!

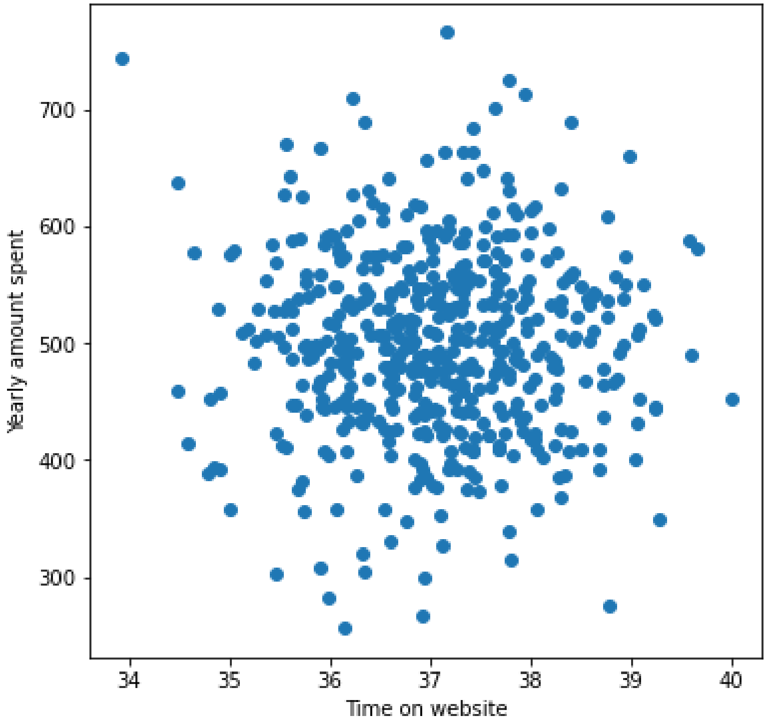
For the rest of the exercise we'll only be using the numerical data of the csv file.

Q. Use seaborn to create a jointplot to compare the Time on Website and Yearly Amount Spent columns. Does the correlation make sense?

In [8]:

```
plt.figure(figsize=(6,6))
plt.xlabel('Time on website')
plt.ylabel('Yearly amount spent')
plt.scatter(x=df['Time on Website'],y=df['Yearly Amount Spent'])

plt.show()
```



In [9]:

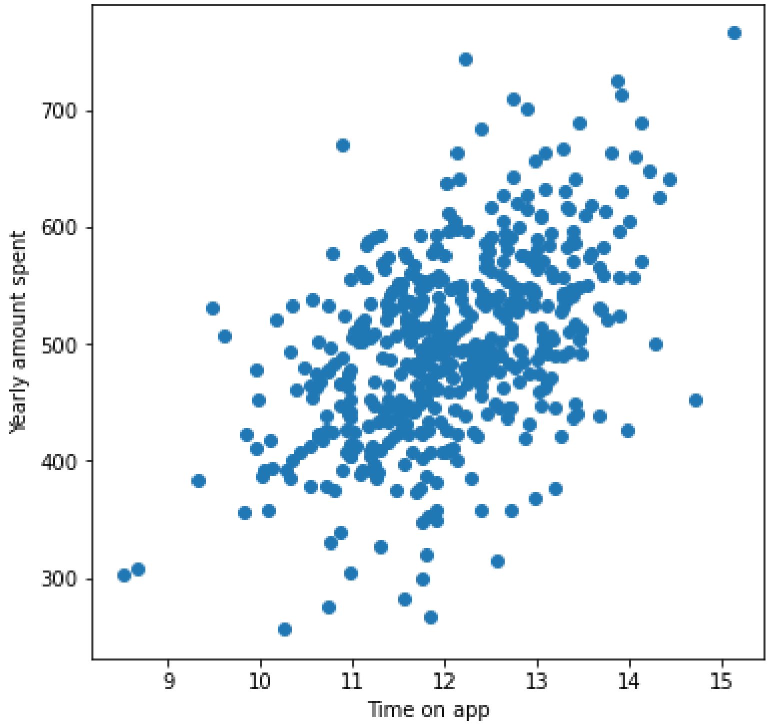
```
```

Q.Do the same but with the Time on App column instead.

In [10]:

```
plt.figure(figsize=(6,6))
plt.xlabel('Time on app')
plt.ylabel('Yearly amount spent')
plt.scatter(x=df['Time on App'],y=df['Yearly Amount Spent'])
```

```
plt.show()
```



In [11]:

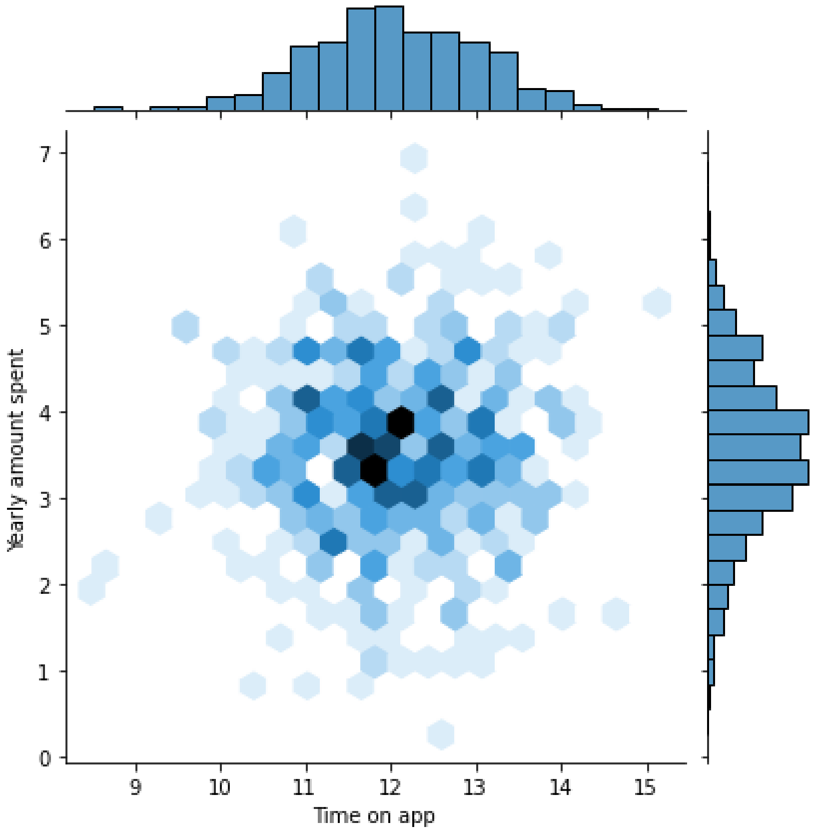
```

```

Q.Use jointplot to create a 2D hex bin plot comparing Time on App and Length of Membership.

In [12]:

```
sns.jointplot(x = df["Time on App"], y = df['Length of Membership'],
              kind = "hex", data = df)
plt.xlabel('Time on app')
plt.ylabel('Yearly amount spent')
plt.show()
```



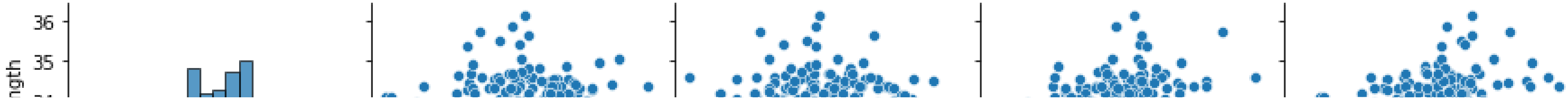
In [13]:

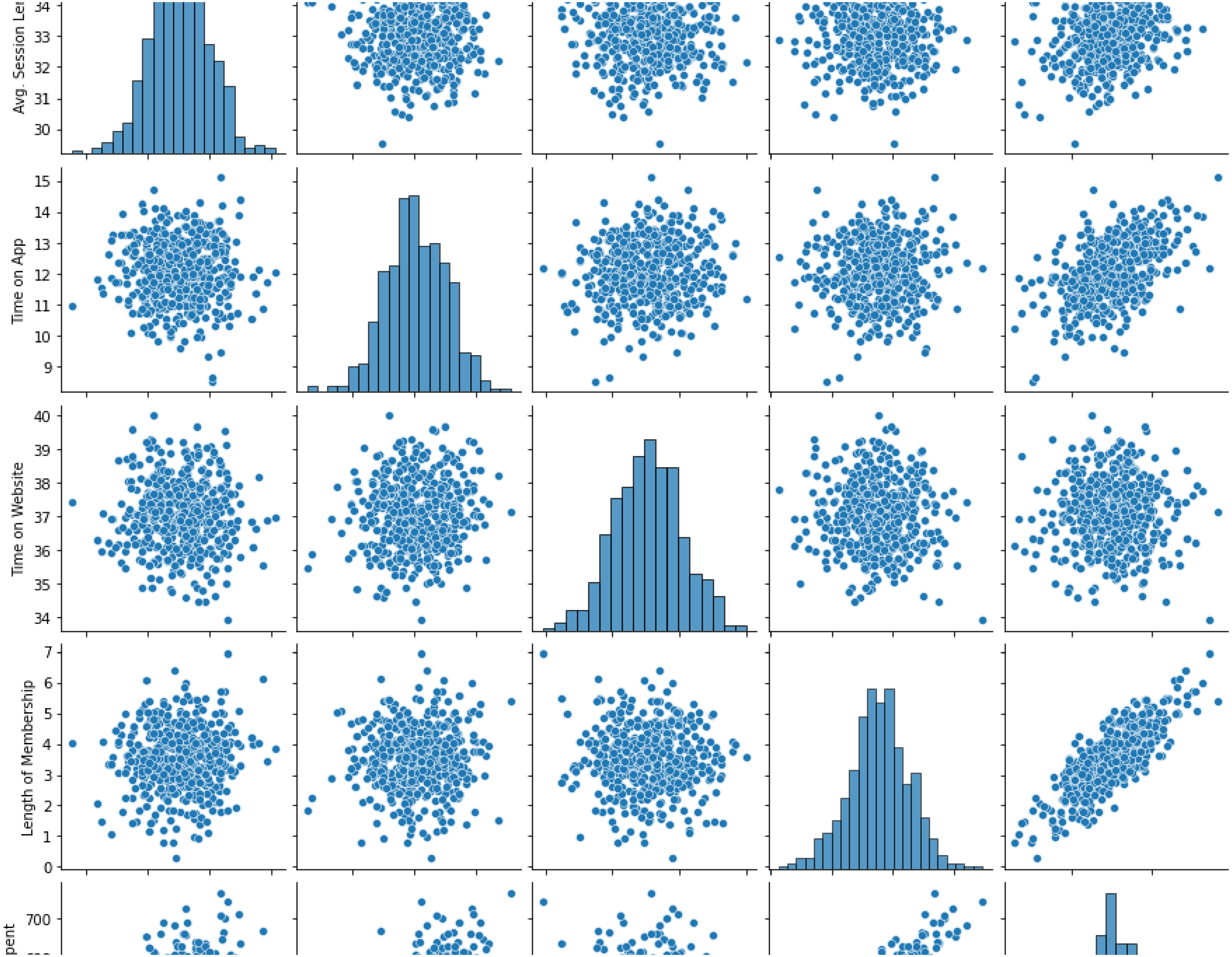
Q.Let's explore these types of relationships across the entire data set. Use pairplot to recreate the plot below.(Don't worry about the the colors)

In [14]:

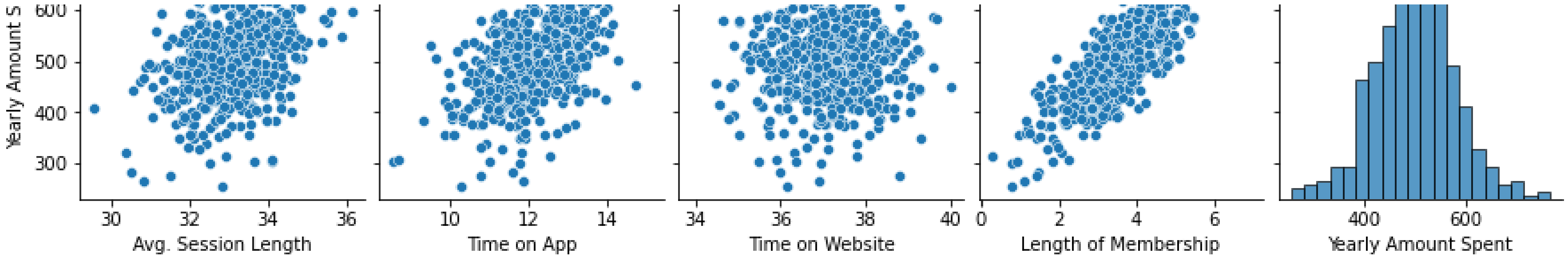
```
sns.pairplot(df)
```

Out[14]: <seaborn.axisgrid.PairGrid at 0x13d61d71a30>





Avg. Session Len
Time on App
Time on Website
Length of Membership
pent



In [15]:

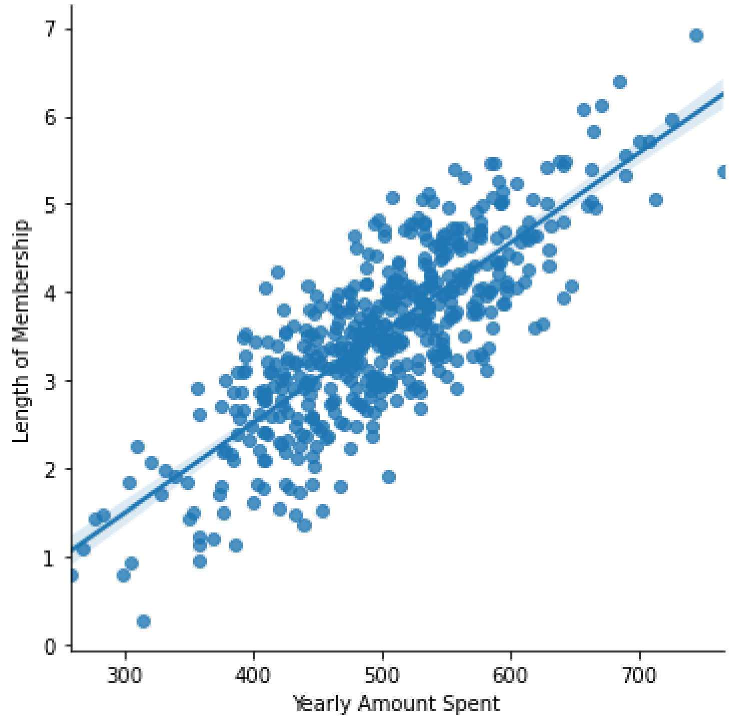
Q.Based ON this plot which column looks the most correlated feature with Yearly Amount Spent, write your anwser below?

-according to above pairplot length of membership is the most co-related feature with yearly amount spent

Q.Create a linear model plot (using seaborn's lmplot) of Yearly Amount Spent vs. Length of Membership.

In [16]:

```
g= sns.lmplot(x = "Yearly Amount Spent", y = 'Length of Membership', data = df)
```



In [17]:

```

```

Training and Testing Data

Now that we've explored the data a bit, let's go ahead and split the data into training and testing sets. **Set a variable X equal to the numerical features of the customers and a variable y equal to the "Yearly Amount Spent" column.**

In [18]:

```
df=df.select_dtypes(float)
df
```

Out[18]:

	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
0	34.497268	12.655651	39.577668	4.082621	587.951054

	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
1	31.926272	11.109461	37.268959	2.664034	392.204933
2	33.000915	11.330278	37.110597	4.104543	487.547505
3	34.305557	13.717514	36.721283	3.120179	581.852344
4	33.330673	12.795189	37.536653	4.446308	599.406092
...	...	...	...	...	...
495	33.237660	13.566160	36.417985	3.746573	573.847438
496	34.702529	11.695736	37.190268	3.576526	529.049004
497	32.646777	11.499409	38.332576	4.958264	551.620145
498	33.322501	12.391423	36.840086	2.336485	456.469510
499	33.715981	12.418808	35.771016	2.735160	497.778642

500 rows × 5 columns

In [19]:

```
```

In [20]:

```
x =df.iloc[:,:-1]
y =df.iloc[:,-1]
x
```

Out[20]:

	Avg. Session Length	Time on App	Time on Website	Length of Membership
0	34.497268	12.655651	39.577668	4.082621
1	31.926272	11.109461	37.268959	2.664034
2	33.000915	11.330278	37.110597	4.104543
3	34.305557	13.717514	36.721283	3.120179
4	33.330673	12.795189	37.536653	4.446308
...	...	...	...	...

	Avg. Session Length	Time on App	Time on Website	Length of Membership
495	33.237660	13.566160	36.417985	3.746573
496	34.702529	11.695736	37.190268	3.576526
497	32.646777	11.499409	38.332576	4.958264
498	33.322501	12.391423	36.840086	2.336485
499	33.715981	12.418808	35.771016	2.735160

500 rows × 4 columns

In [21]:

```
y
```

Out[21]:

```
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
```

Q. Split the data into training and testing sets. Set test_size=0.3 and random_state=101

In [22]:

```
from sklearn.model_selection import train_test_split as t
```

In [23]:

```
xtrain,xtest,ytrain,ytest=t(x,y,test_size=0.3,random_state=101)
```

Training the Model

Now its time to train our model on our training data!

Step1-: import model

In [24]:
```
from sklearn.linear_model import LinearRegression as linreg
```

Step2-: create an object for model

In [25]:
```
lr=linreg()
```

Step3-: train the model

In [26]:
```
lr.fit(xtrain,ytrain)
ypred=lr.predict(xtest)
lr.intercept_
```

Out[26]:
```
-1047.932782250239
```

Q.Print out the coefficients of the model

In [27]:
```
lr.coef_
```

Out[27]:
```
array([25.98154972, 38.59015875,  0.19040528, 61.27909654])
```

In [28]:
```

```

Predicting Test Data

Now that we have fit our model, let's evaluate its performance by predicting off the test values!

Step 4-: predict the test set and save it in ypred and print ypred

In [29]:
```
ypred
```

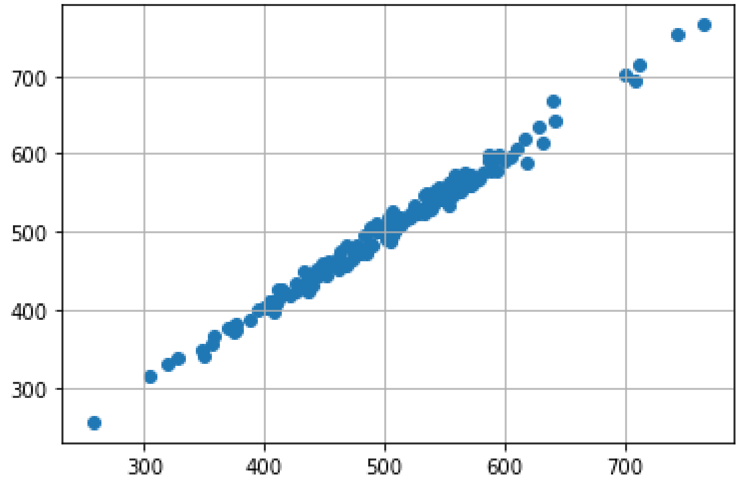
Out[29]:
```
array([456.44186104, 402.72005312, 409.2531539 , 591.4310343 ,
       590.01437275, 548.82396607, 577.59737969, 715.44428115,
       473.7893446 , 545.9211364 , 337.8580314 , 500.38506697,
       552.93478041, 409.6038964 , 765.52590754, 545.83973731,
       693.25969124, 507.32416226, 573.10533175, 573.2076631 ,
       397.44989709, 555.0985107 , 458.19868141, 482.66899911,
       559.2655959 , 413.00946082, 532.25727408, 377.65464817,
       535.0209653 , 447.80070905, 595.54339577, 667.14347072,
       511.96042791, 573.30433971, 505.02260887, 565.30254655,
       460.38785393, 449.74727868, 422.87193429, 456.55615271,
       598.10493696, 449.64517443, 615.34948995, 511.88078685,
       504.37568058, 515.95249276, 568.64597718, 551.61444684,
       356.5552241 , 464.9759817 , 481.66007708, 534.2220025 ,
       256.28674001, 505.30810714, 520.01844434, 315.0298707 ,
       501.98080155, 387.03842642, 472.97419543, 432.8704675 ,
       539.79082198, 590.03070739, 752.86997652, 558.27858232,
       523.71988382, 431.77690078, 425.38411902, 518.75571466,
       641.9667215 , 481.84855126, 549.69830187, 380.93738919,
       555.18178277, 403.43054276, 472.52458887, 501.82927633,
       473.5561656 , 456.76720365, 554.74980563, 702.96835044,
       534.68884588, 619.18843136, 500.11974127, 559.43899225,
       574.8730604 , 505.09183544, 529.9537559 , 479.20749452,
       424.78407899, 452.20986599, 525.74178343, 556.60674724,
       425.7142882 , 588.8473985 , 490.77053065, 562.56866231,
       495.75782933, 445.17937217, 456.64011682, 537.98437395,
       367.06451757, 421.12767301, 551.59651363, 528.26019754,
       493.47639211, 495.28105313, 519.81827269, 461.15666582,
       528.8711677 , 442.89818166, 543.20201646, 350.07871481,
       401.49148567, 606.87291134, 577.04816561, 524.50431281,
       554.11225704, 507.93347015, 505.35674292, 371.65146821,
       342.37232987, 634.43998975, 523.46931378, 532.7831345 ,
       574.59948331, 435.57455636, 599.92586678, 487.24017405,
       457.66383406, 425.25959495, 331.81731213, 443.70458331,
       563.47279005, 466.14764208, 463.51837671, 381.29445432,
       411.88795623, 473.48087683, 573.31745784, 417.55430913,
       543.50149858, 547.81091537, 547.62977348, 450.99057409,
       561.50896321, 478.30076589, 484.41029555, 457.59099941,
       411.52657592, 375.47900638])
```

In [30]:
```

```

Q.Create a scatterplot of the real test values versus the predicted values.

In [31]:
```
plt.scatter(ytest,ypred)
plt.grid()
```



In [32]:
```

```

Evaluating the Model

Let's evaluate our model performance by calculating the residual sum of squares and the explained variance score (R^2).

Q.Calculate the Mean Absolute Error, Mean Squared Error, and the Root Mean Squared Error. Refer to the lecture or to Wikipedia for the formulas

In [33]:
```
from sklearn.metrics import mean_absolute_error as mae ,mean_squared_error as mse,r2_score as r
```

In [34]:
```
print(f'MAE  : {mae(ytest,ypred)}')
print(f'MSE  : {mse(ytest,ypred)}')
print(f'RMSE : {np.sqrt(mse(ytest,ypred))}')
```

```
MAE  : 7.228148653430806
MSE  : 79.8130516509741
```

```
RMSE : 8.933815066978614
```

In [35]:

```

```

Residuals

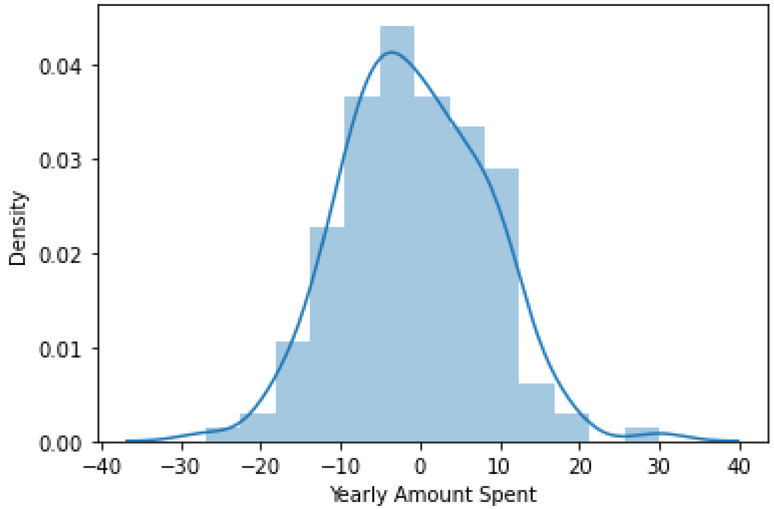
You should have gotten a very good model with a good fit. Let's quickly explore the residuals to make sure everything was okay with our data.

Q.Plot a histogram of the residuals and make sure it looks normally distributed. Use either seaborn distplot, or just plt.hist().

In [36]:

```
sns.distplot(ytest-ypred)
plt.show()
```

```
C:\Users\sharma17\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [37]:

```

```

Conclusion

We still want to figure out the answer to the original question, do we focus our efforst on mobile app or website development? Or maybe that doesn't even really matter, and Membership Time is what is really important. Let's see if we can interpret the coefficients at all to get an idea.

Q.Recreate the dataframe below.

In [38]:

```
coefficient=pd.DataFrame(lr.coef_,x.columns,columns=['Coefficient'])
coefficient
```

Out[38]:

	Coefficient
Avg. Session Length	25.981550
Time on App	38.590159
Time on Website	0.190405
Length of Membership	61.279097

In [39]:

```
```

Q.How can you interpret these coefficients? write as a markdown below

- eg.seting all other x constant, if we increase **col_name** by 1 unit, the **target_col** will increase by **coefficient_value** dollars

write your answers all columns here

setting all other x constant, if we increase Avg. Session Length by 1 unit, the Yearly Amount Spent will increase by 25.981550 dollars

setting all other x constant, if we increase Time on App by 1 unit, the Yearly Amount Spent will increase by 38.590159 dollars

setting all other x constant, if we increase Time on Website by 1 unit, the Yearly Amount Spent will increase by 0.190405 dollars

Length of Membership

setting all other x constant, if we increase Length of Membership by 1 unit, the Yearly Amount Spent will increase by 61.279097 dollars

Q.Do you think the company should focus more on their mobile app or on their website?

By the Interpretation we can conclude that company should focus more on the app instead of website as it generating more revenue

and company shoulde include more feature on the website as it is generating least revenue or neglect the website instead of website by adding some feature in the application will boast the revenue by App

THE END