2. Linear Regression

- Linear Regression is a supervised machine learning algorithm where the predicted output is continuous and has a constant slope.
- Simple linear regression is a type of regression analysis where the number of independent variables is one and there is a linear relationship between the independent(x) and dependent(y) variable
- Regression is a method of modelling a target value based on independent predictors.
- It’s used to predict values within a continuous range, (e.g. sales, price) rather than trying to classify them into categories (e.g. cat, dog)

Imports Library

In [ ]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

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

Import the dataset

In [ ]:

```
customers = pd.read_csv('Ecommerce Customers.csv')
```

Exploratory Data Analysis

In [ ]:

```
customers.head(10)
```

Out[ ]:

	Email	Address	Avatar	Avg. Session Length	Time on App
0	mstephenson@fernandez.com	835 Frank Tunnel\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655651
1	hduke@hotmail.com	4547 Archer Common\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109461
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\nCobbborough, D...	Bisque	33.000915	11.330278
3	riverarebecca@gmail.com	1414 David Throughway\nPort Jason, OH 22070-1220	SaddleBrown	34.305557	13.717514
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\nPort Jacobville, PR 3...	MediumAquaMarine	33.330673	12.795189
5	alvareznancy@lucas.biz	645 Martha Park Apt. 611\nJeffreychester, MN 6...	FloralWhite	33.871038	12.026925
6	katherine20@yahoo.com	68388 Reyes Lights Suite 692\nJosephbury, WV 9...	DarkSlateBlue	32.021596	11.366348
7	awatkins@yahoo.com	Unit 6538 Box 8980\nDPO AP 09026-4941	Aqua	32.739143	12.351959
8	vchurch@walter-martinez.com	860 Lee Key\nWest Debra, SD 97450-0495	Salmon	33.987773	13.386235
9	bonnie69@lin.biz	PSC 2734, Box 5255\nAPO AA 98456-7482	Brown	31.936549	11.814128

```
In [ ]:
```

```
customers.tail(10)
```

```
Out[ ]:
```

	Email	Address	Avatar	Avg. Session Length	Time Ap
490	brian28@sanchez.org	7446 Mary Ferry\nLake Sherryfurt, GA 49066-0207	GhostWhite	34.695591	11.6089
491	leonardhancock@hotmail.com	64147 Alexander Station Apt. 474\nEast Jasonvi...	SeaShell	34.343922	11.6930
492	davidsonkathleen@gmail.com	70128 Zimmerman Overpass\nRobertsshire, VA 59860	DarkBlue	33.680937	11.2015
493	nathan84@lowery.net	01242 Stephanie Ways Suite 003\nChurchville, M...	MediumSeaGreen	32.060914	12.6254
494	kellydeborah@chan.biz	354 Sanchez Wall Suite 884\nJuliabury, VI 39735	DarkTurquoise	33.431097	13.3506
495	lewisjessica@craig-evans.com	4483 Jones Motorway Suite 872\nLake Jamiefurt,...	Tan	33.237660	13.5661
496	katrina56@gmail.com	172 Owen Divide Suite 497\nWest Richard, CA 19320	PaleVioletRed	34.702529	11.6957
497	dale88@hotmail.com	0787 Andrews Ranch Apt. 633\nSouth Chadburgh, ...	Cornsilk	32.646777	11.4994
498	cwilson@hotmail.com	680 Jennifer Lodge Apt. 808\nBrendachester, TX...	Teal	33.322501	12.3914
499	hannahwilson@davidson.com	49791 Rachel Heights Apt. 898\nEast Drewboroug...	DarkMagenta	33.715981	12.4188

In [ ]:

```
customers.describe()
```

Out[ ]:

	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 [ ]:

```
customers.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
```

Data Visualization

In [ ]:

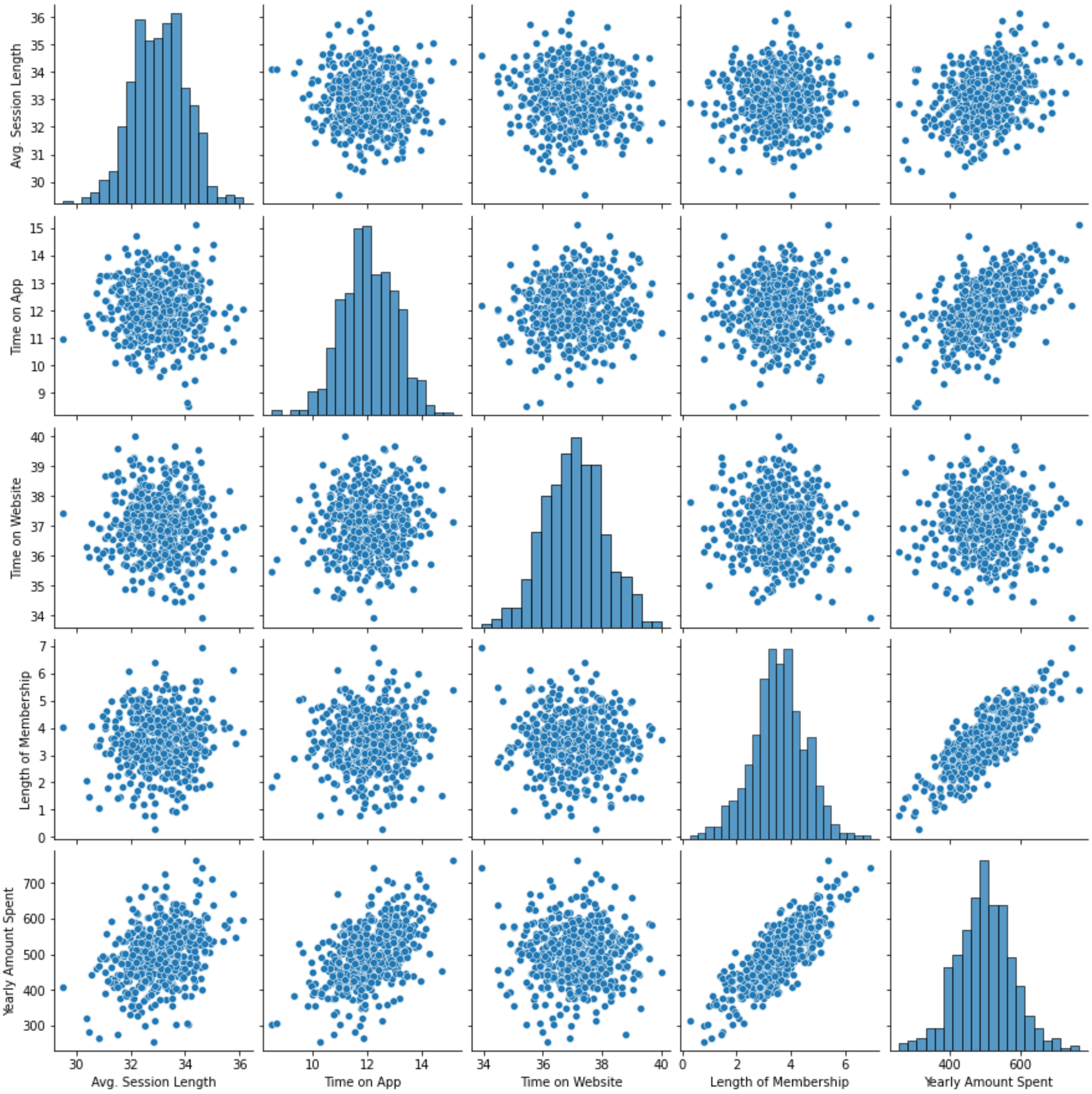
```
import seaborn as sns
```

In [ ]:

```
sns.pairplot(customers)
```

Out[ ]:

```
<seaborn.axisgrid.PairGrid at 0x7f6e9eacada0>
```



Based off this plot what looks to be the most correlated feature with Yearly Amount Spent?

In [ ]:

```
#Length of Membership
```

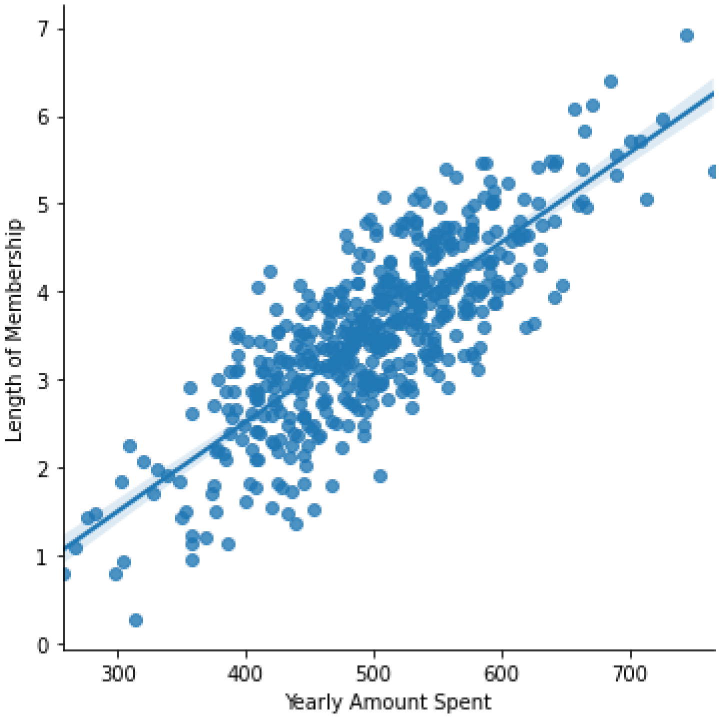
Create a linear model plot (using seaborn's lmplot) of Yearly Amount Spent vs. Length of Membership.

In [ ]:

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

Out[ ]:

```
<seaborn.axisgrid.FacetGrid at 0x7f6e8b63f8d0>
```



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 [ ]:

```
y = customers['Yearly Amount Spent']
```

In [ ]:

```
X = customers[['Avg. Session Length', 'Time on App','Time on Website', 'Length of Membe
rship']]
```

Split the dataset

Use model_selection.train_test_split from sklearn to split the data into training and testing sets. Set test_size=0.3 and random_state=101

In [ ]:

```
from sklearn.model_selection import train_test_split
```

In [ ]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=1
01)
```

Training the Model

In [ ]:

```
X_train
```

Out[ ]:

	Avg. Session Length	Time on App	Time on Website	Length of Membership
161	33.503705	12.399436	35.012806	0.968622
72	32.386252	10.674653	38.006583	3.401522
246	31.909627	11.347264	36.323652	5.314354
230	32.351478	13.105159	35.574842	3.641497
391	33.481931	11.918670	37.317705	3.336339
...	...	...	...	...
63	32.789773	11.670066	37.408748	3.414688
326	33.217188	10.999684	38.442767	4.243813
337	31.827979	12.461147	37.428997	2.974737
11	33.879361	11.584783	37.087926	3.713209
351	32.189845	11.386776	38.197483	4.808320

400 rows × 4 columns

In [ ]:

```
from sklearn.linear_model import LinearRegression
```

In [ ]:

```
lm = LinearRegression()
```

In [ ]:

```
lm.fit(X_train,y_train)
```

Out[ ]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=F
alse)
```

In [ ]:

```
print('Coefficients: \n', lm.coef_)
```

```
Coefficients: 
 [26.02948861 38.70983485  0.35618404 61.47280903]
```

Predicting Test Data

In [ ]:

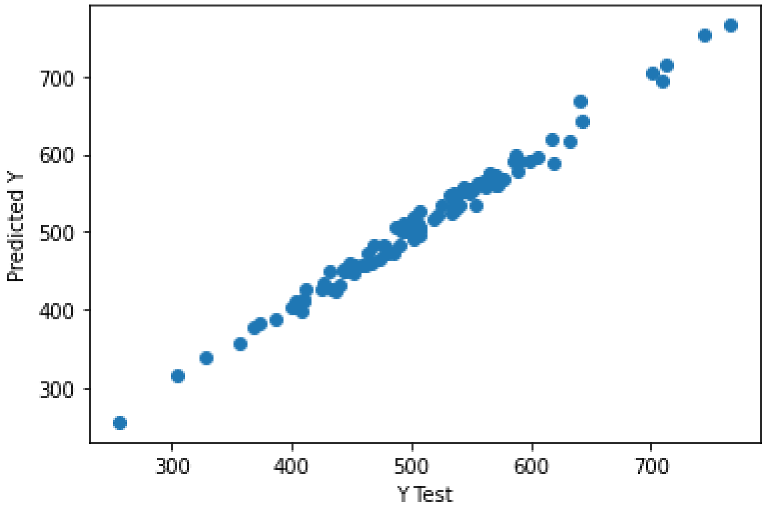
```
predictions = lm.predict(X_test)
```

In [ ]:

```
plt.scatter(y_test,predictions)
plt.xlabel('Y Test')
plt.ylabel('Predicted Y')
```

Out[ ]:

```
Text(0, 0.5, 'Predicted Y')
```



Evaluating the Model

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

Calculate the Mean Absolute Error, Mean Squared Error, and the Root Mean Squared Error.

In [ ]:

```
from sklearn import metrics

print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
MAE: 7.758907540457862
MSE: 91.82335857016922
RMSE: 9.582450551407465
```

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.

Recreate the dataframe below.

In [ ]:

```
coeffecients = pd.DataFrame(lm.coef_,X.columns)
coeffecients.columns = ['Coeffecient']
coeffecients
```

Out[ ]:

	Coeffecient
Avg. Session Length	26.029489
Time on App	38.709835
Time on Website	0.356184
Length of Membership	61.472809