Linear Regression Project

Imports

```
import pandas as pd
import numpy as np
import matplotlib as plt
import seaborn as sns
%matplotlib inline
```

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.

```
import os
print(os.listdir("../input/"))
```

```
['ecommerce-customers']
```

```
import os

# List files in the directory
directory_path = '/kaggle/input/ecommerce-customers'
files = os.listdir(directory_path)
print(files)
```

```
['Ecommerce Customers.csv']
```

```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("srolka/ecommerce-customers")

print("Path to dataset files:", path)
```

```
Path to dataset files: /kaggle/input/ecommerce-customers
```

```
import os

# List files in the directory
directory_path = '/kaggle/input/ecommerce-customers'
```

```
files = os.listdir(directory_path)
print(files)
```

```
['Ecommerce Customers.csv']
```

```
customers = pd.read_csv('/kaggle/input/ecommerce-customers/Ecommerce
Customers.csv')
```

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

```
customers.head()
```

```
                            Email  \
0      mstephenson@fernandez.com
1              hduke@hotmail.com
2               pallen@yahoo.com
3         riverarebecca@gmail.com
4  mstephens@davidson-herman.com

                                              Address             Avatar
\
0       835 Frank Tunnel\nWrightmouth, MI 82180-9605             Violet

1     4547 Archer Common\nDiazchester, CA 06566-8576          DarkGreen

2  24645 Valerie Unions Suite 582\nCobbborough, D...             Bisque

3   1414 David Throughway\nPort Jason, OH 22070-1220        SaddleBrown

4  14023 Rodriguez Passage\nPort Jacobville, PR 3...  MediumAquaMarine


   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

   Yearly Amount Spent
0           587.951054
1           392.204933
2           487.547505
```

```
3           581.852344
4           599.406092
```

```
customers.describe()
```

```
       Avg. Session Length  Time on App  Time on Website  \
count           500.000000   500.000000       500.000000
mean             33.053194    12.052488        37.060445
std               0.992563     0.994216         1.010489
min              29.532429     8.508152        33.913847
25%              32.341822    11.388153        36.349257
50%              33.082008    11.983231        37.069367
75%              33.711985    12.753850        37.716432
max              36.139662    15.126994        40.005182

       Length of Membership  Yearly Amount Spent
count             500.000000           500.000000
mean                3.533462           499.314038
std                 0.999278            79.314782
min                 0.269901           256.670582
25%                 2.930450           445.038277
50%                 3.533975           498.887875
75%                 4.126502           549.313828
max                 6.922689           765.518462
```

```
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
```

```
customers.shape
```

```
(500, 8)
```

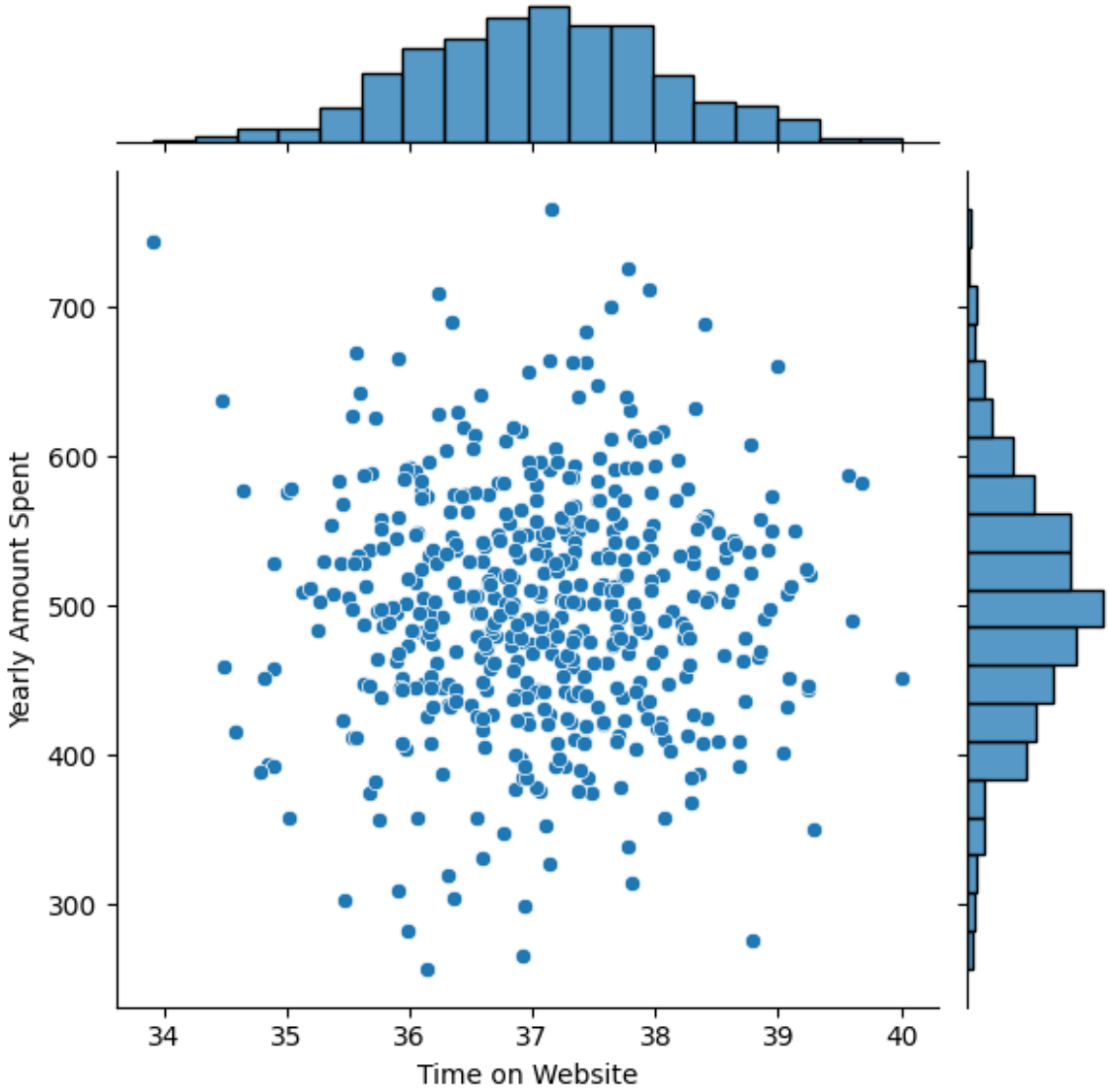
Exploratory Data Analysis

Let's explore the data!

```
sns.jointplot(x='Time on Website',y ='Yearly Amount Spent', data =
customers)
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

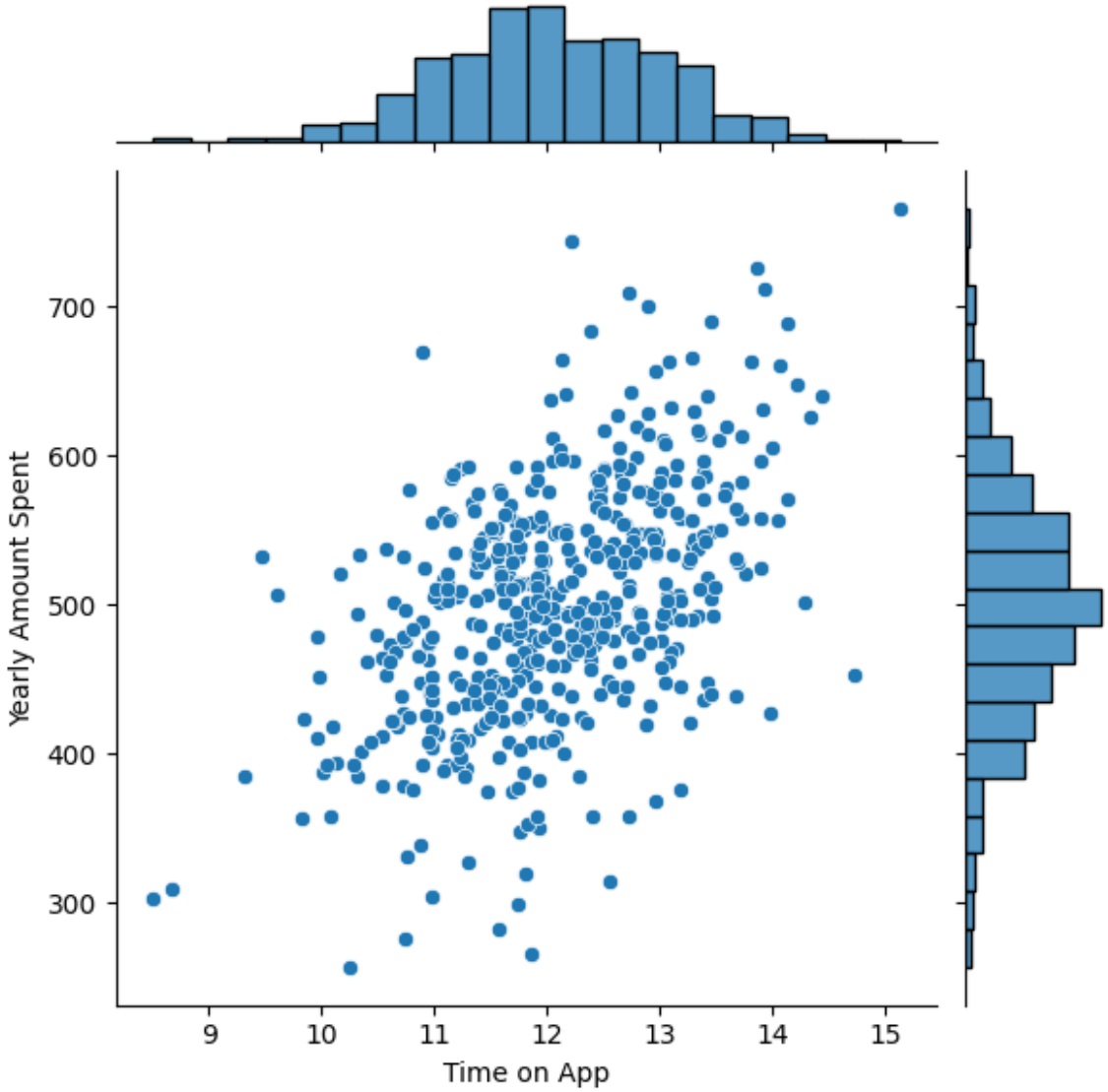
```
<seaborn.axisgrid.JointGrid at 0x7eff283c97b0>
```



```
sns.jointplot(x="Time on App", y="Yearly Amount Spent",
data=customers)
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

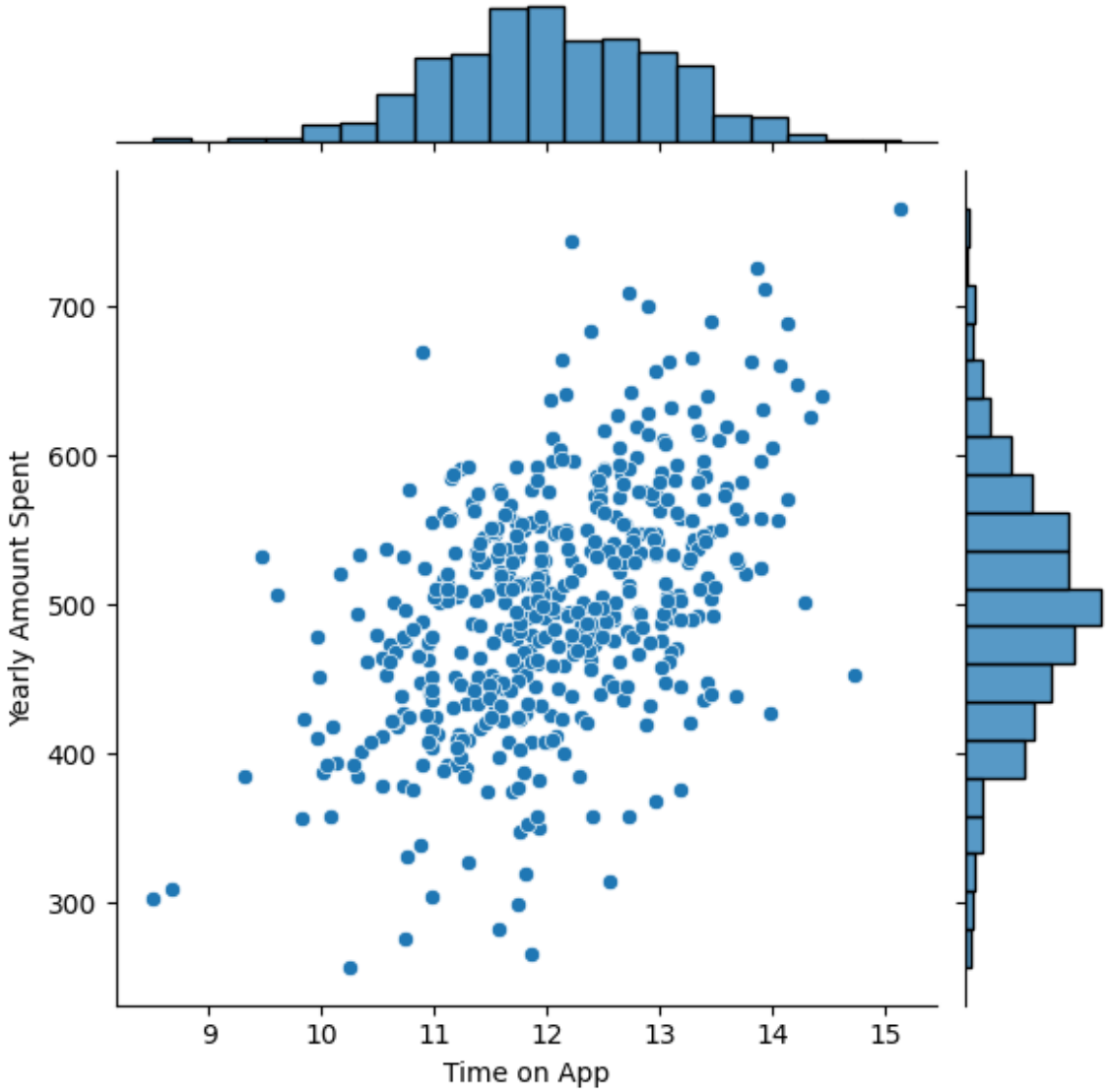
```
<seaborn.axisgrid.JointGrid at 0x7efeeff770a0>
```



```
sns.jointplot(x='Time on App',y ='Yearly Amount Spent', data =
customers)
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

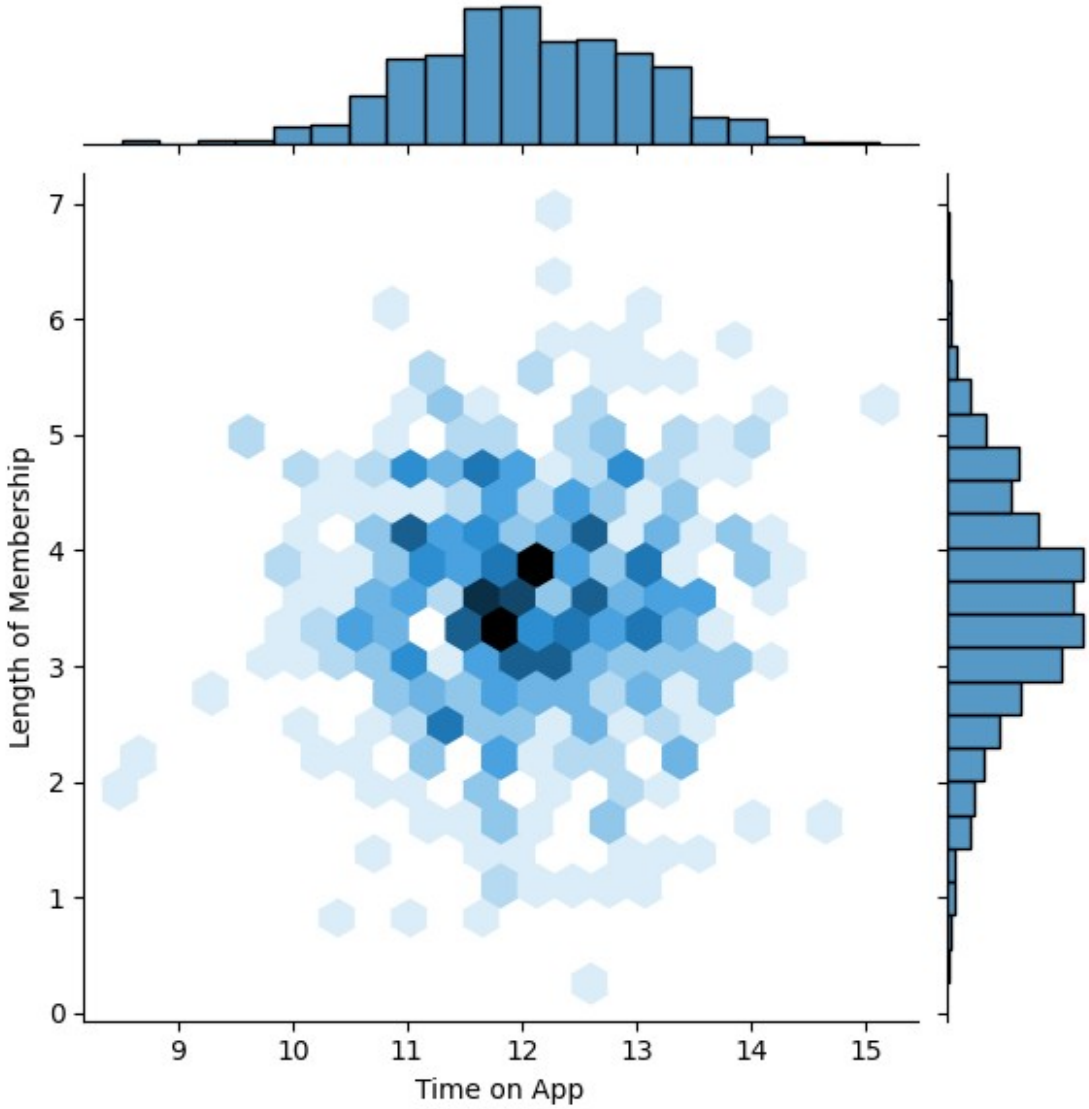
```
<seaborn.axisgrid.JointGrid at 0x7efeeff410c0>
```



```
sns.jointplot(x='Time on App',y ='Length of Membership', data =
customers, kind='hex')
```

```
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

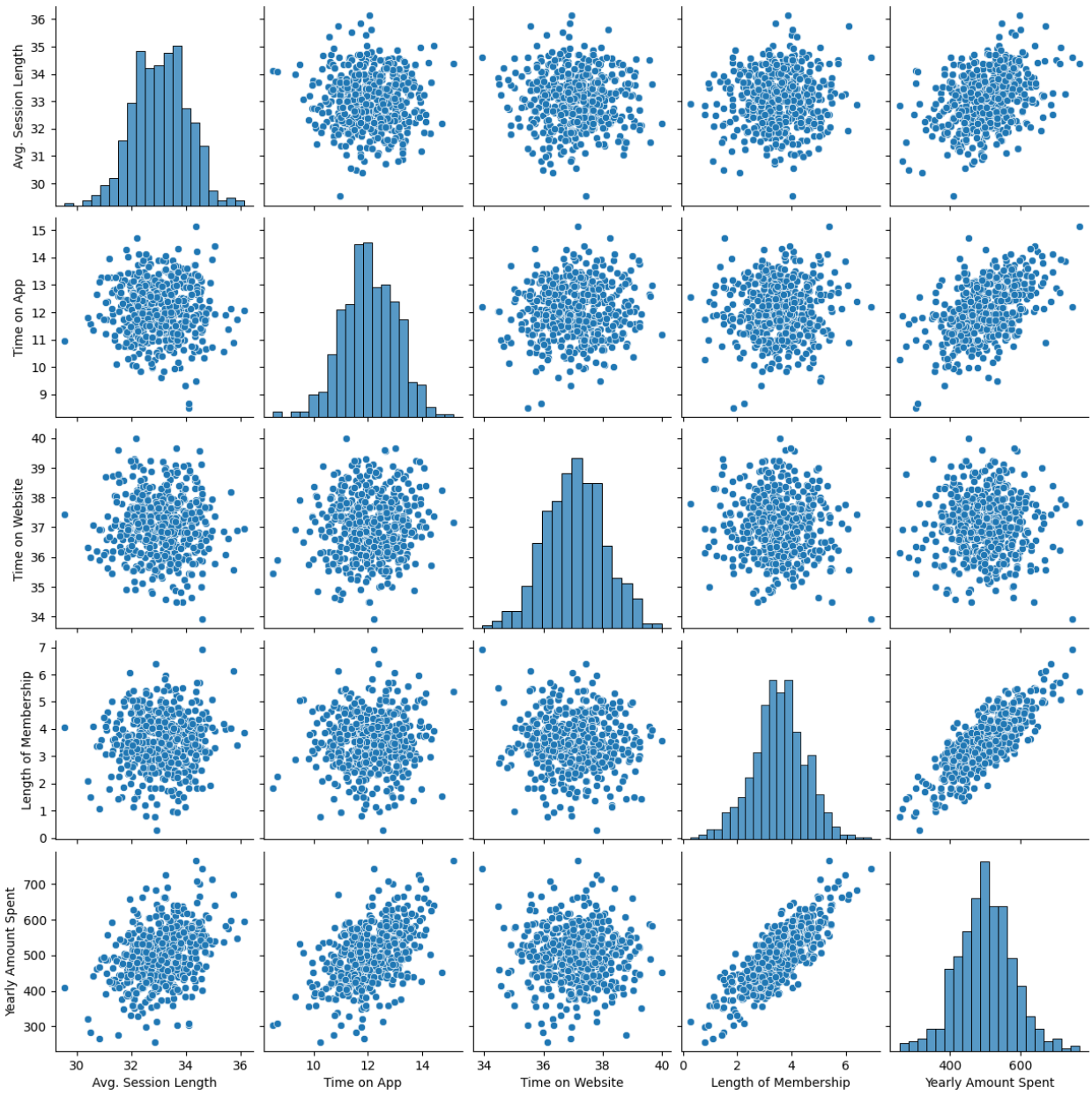
```
<seaborn.axisgrid.JointGrid at 0x7efeefccc7f0>
```



Let's explore these types of relationships across the entire data set. Use pairplot to recreate the plot below.

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

```
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):

<seaborn.axisgrid.PairGrid at 0x7efeefc20a90>
```



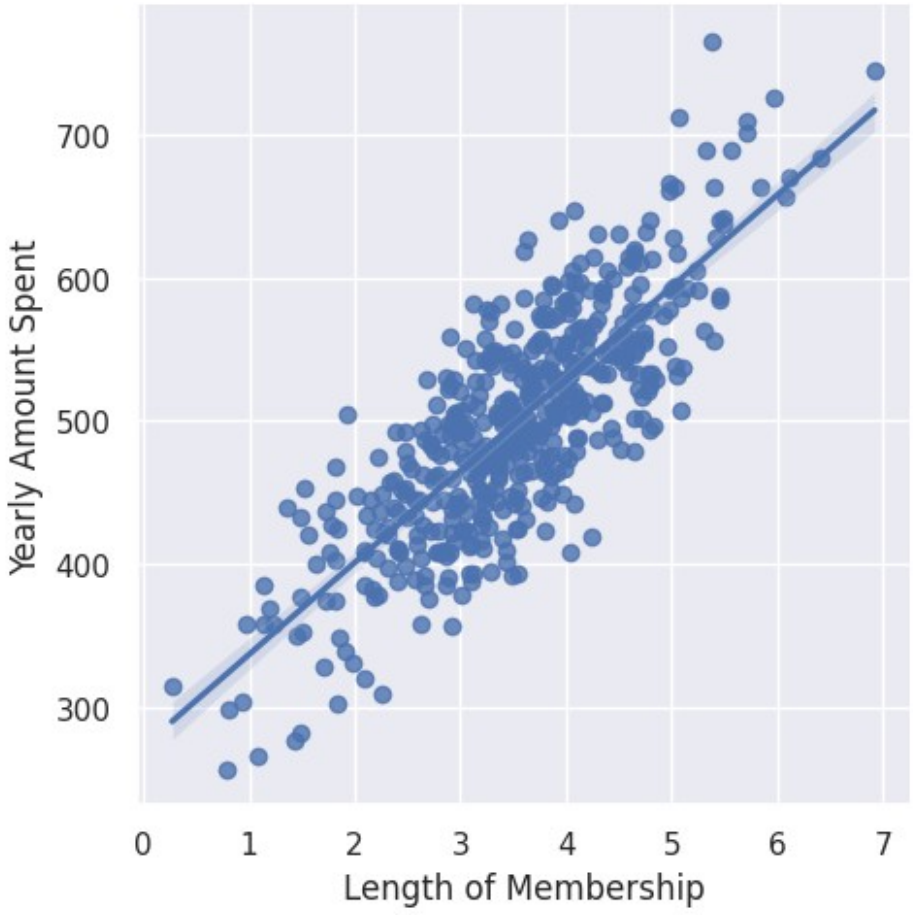
Based off this plot what looks to be the most correlated feature with Yearly Amount Spent?

```
print("Length of Membership")
```

```
Length of Membership
```

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

```
<seaborn.axisgrid.FacetGrid at 0x7efeee8ba8c0>
```



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. **

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

y= customers['Yearly Amount Spent']

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.3, random_state=101)
```

Training the Model

```
from sklearn.linear_model import LinearRegression

lm = LinearRegression()
```

** Train/fit lm on the training data.**

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

```
LinearRegression()
```

Print out the coefficients of the model

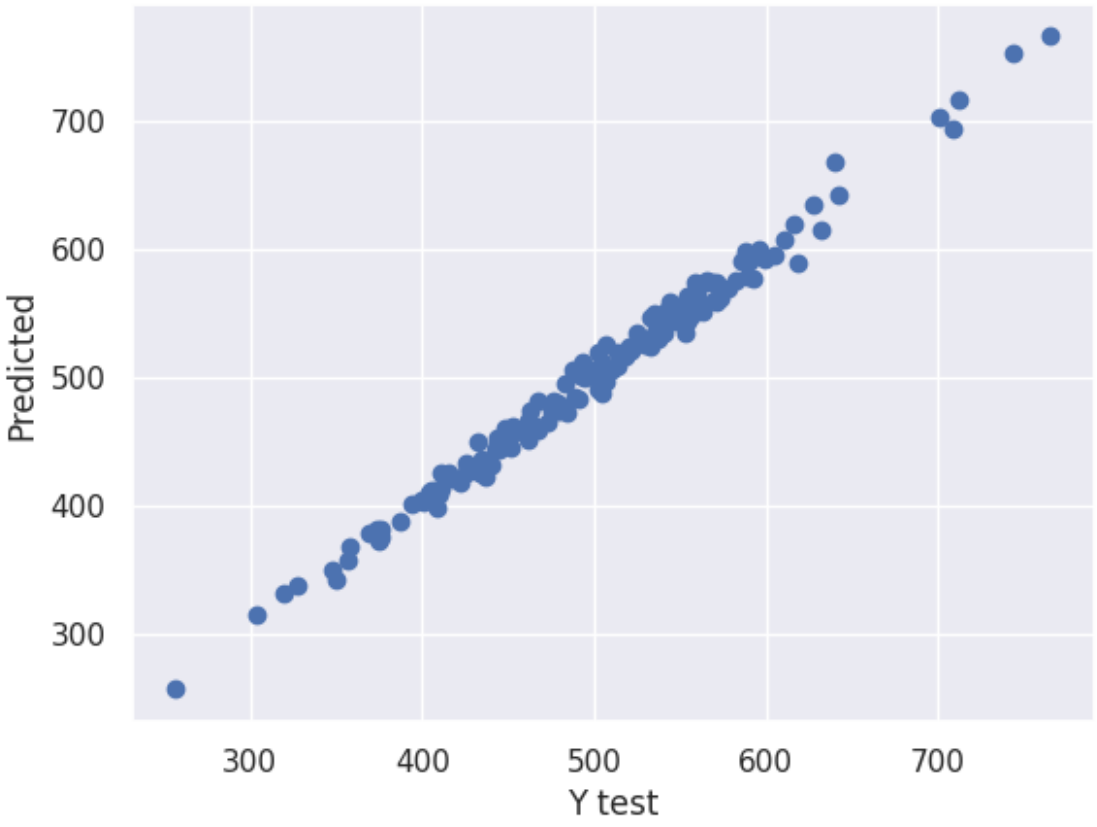
```
print(lm.coef_)
```

```
[25.98154972 38.59015875  0.19040528 61.27909654]
```

Predicting Test Data

```
predictions = lm.predict(X_test)

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

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



Evaluating the Model

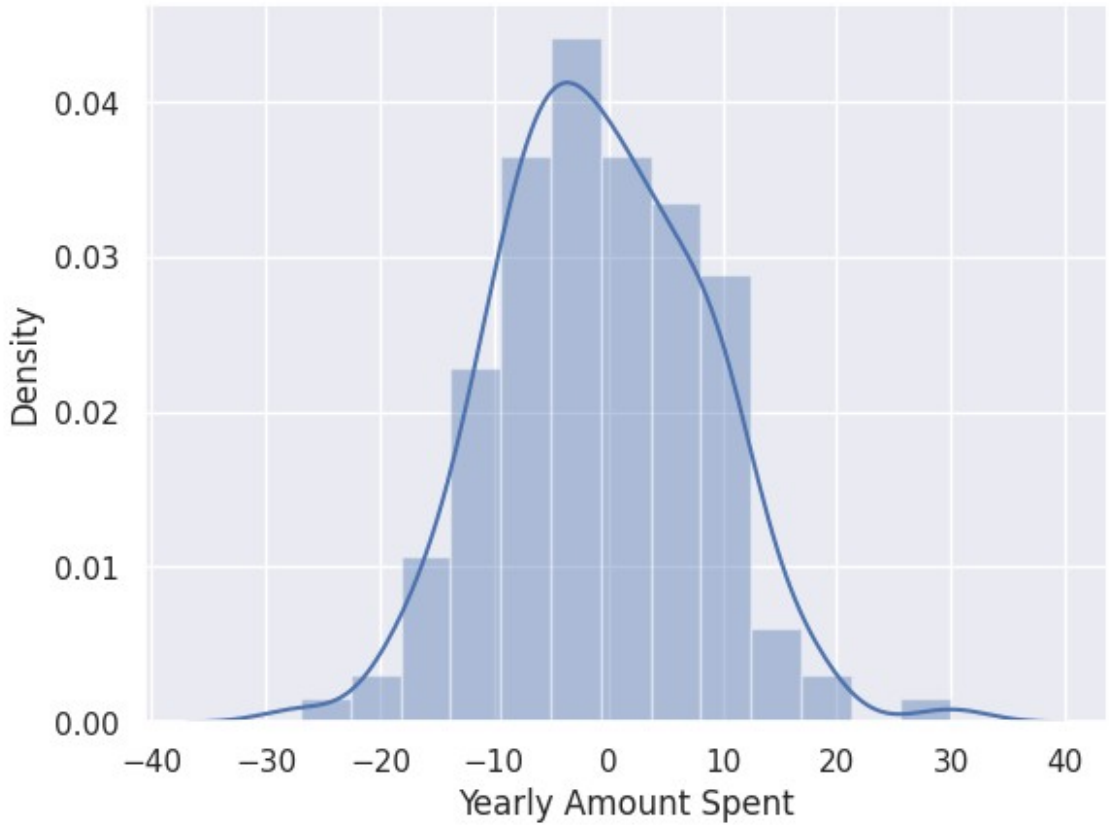
```
import sklearn.metrics as metrics
print('MAE: {}'.format(metrics.mean_absolute_error(y_test,
predictions)))
print('MSE: {}'.format(metrics.mean_squared_error(y_test,
predictions)))
print('RMSE: {}'.format(np.sqrt(metrics.mean_squared_error(y_test,
predictions))))
print('R2: {}'.format(metrics.r2_score(y_test, predictions)))
```

```
MAE: 7.228148653430826
MSE: 79.81305165097427
RMSE: 8.933815066978624
R2: 0.9890046246741234
```

Residuals

```
sns.distplot((y_test-predictions))
```

```
<ipython-input-30-550730dc5ec8>:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn
v0.14.0.

Please adapt your code to use either `displot` (a figure-level
function with
similar flexibility) or `histplot` (an axes-level function for
histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot((y_test-predictions))
/usr/local/lib/python3.10/dist-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed
in a future version. Convert inf values to NaN before operating
instead.
  with pd.option_context('mode.use_inf_as_na', True):

<Axes: xlabel='Yearly Amount Spent', ylabel='Density'>
```



Conclusion

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

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

The company should focus on the mobile app