In [2]:

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

In [3]:

```
%matplotlib inline
```

In [4]:

```
df = pd.read_csv('Ecommerce Customers')
```

In [5]:

```
df.columns
```

Out[5]:

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

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

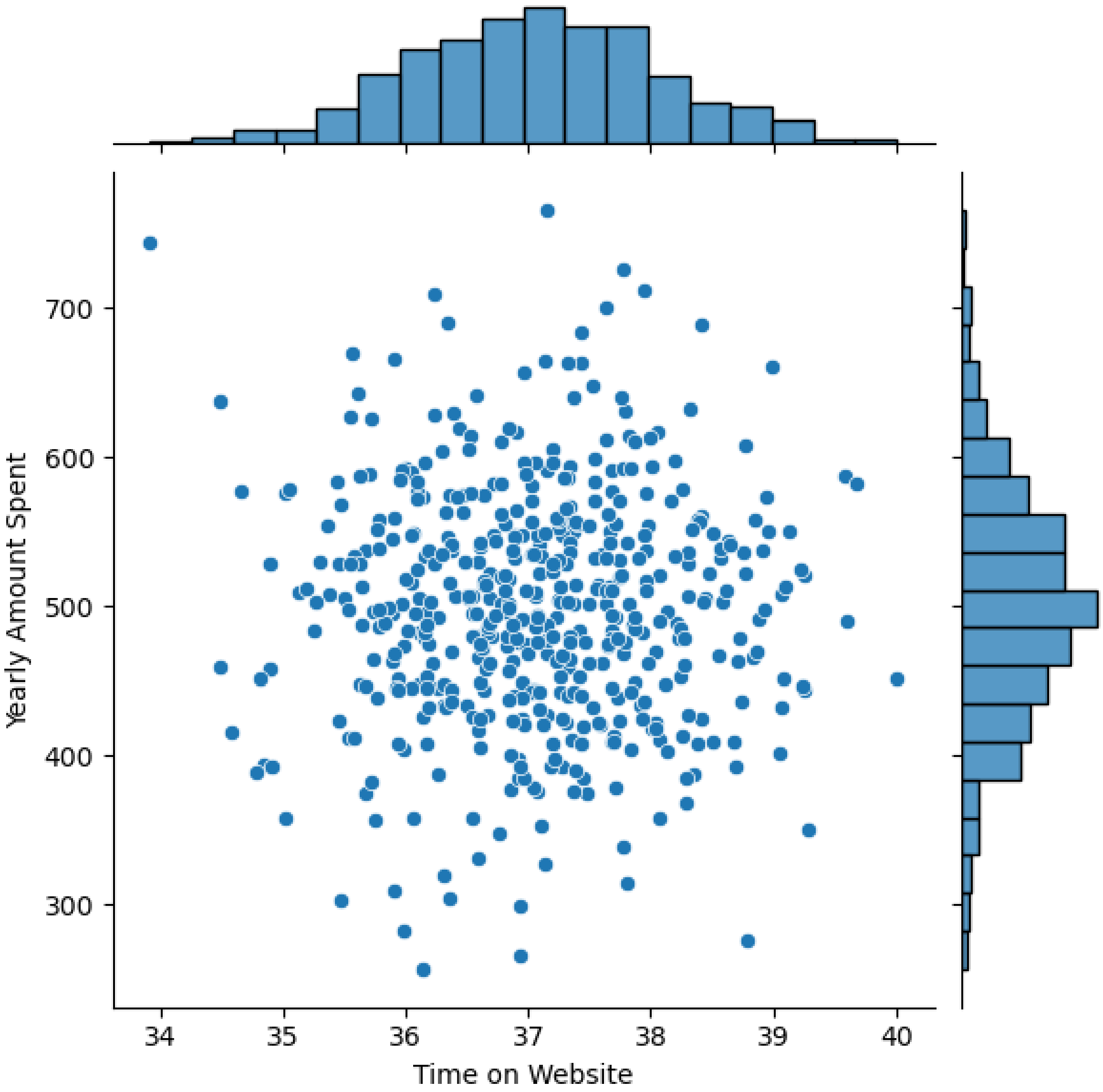
In [7]:

```
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
```

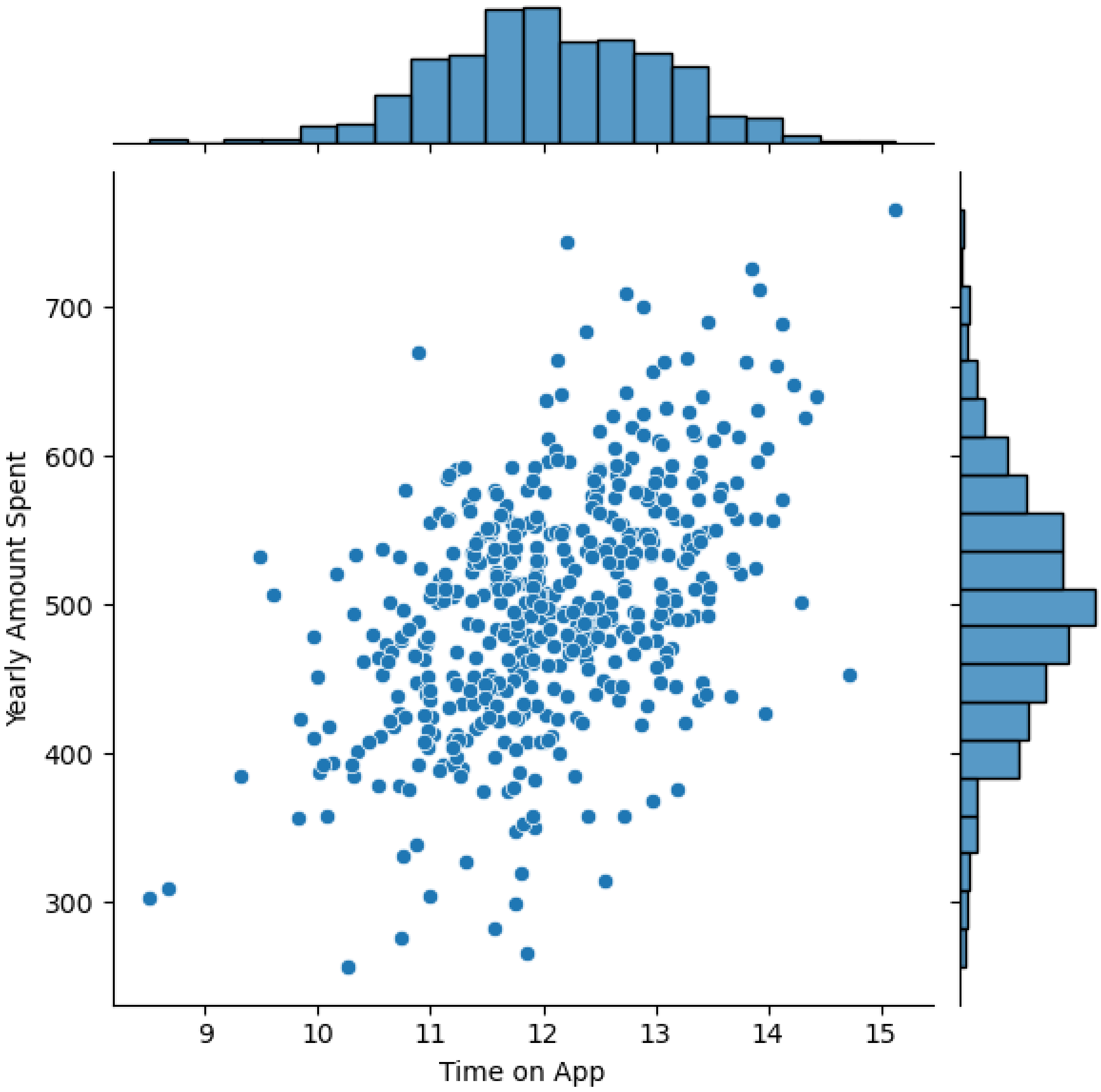
```
In [8]: sns.jointplot(x= df['Time on Website'], y= df['Yearly Amount Spent'])
```

```
Out[8]: <seaborn.axisgrid.JointGrid at 0x1de536bd430>
```



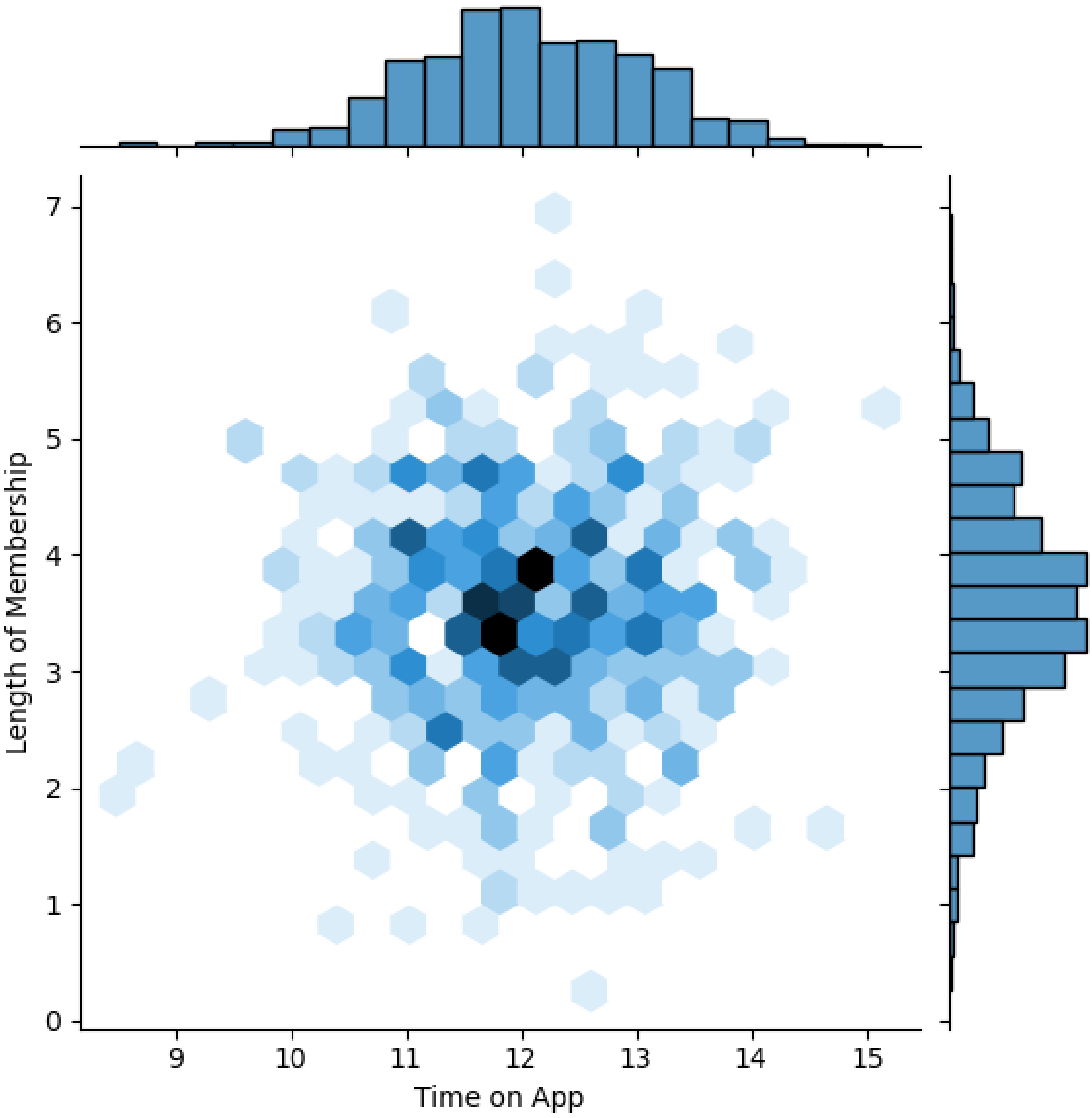
```
In [9]: sns.jointplot(x= df['Time on App'], y=df['Yearly Amount Spent'])
```

```
Out[9]: <seaborn.axisgrid.JointGrid at 0x1de53a5a400>
```



```
In [10]: sns.jointplot( x=df['Time on App'], y= df['Length of Membership'], kind= 'hex')
```

```
Out[10]: <seaborn.axisgrid.JointGrid at 0x1de53d26d90>
```



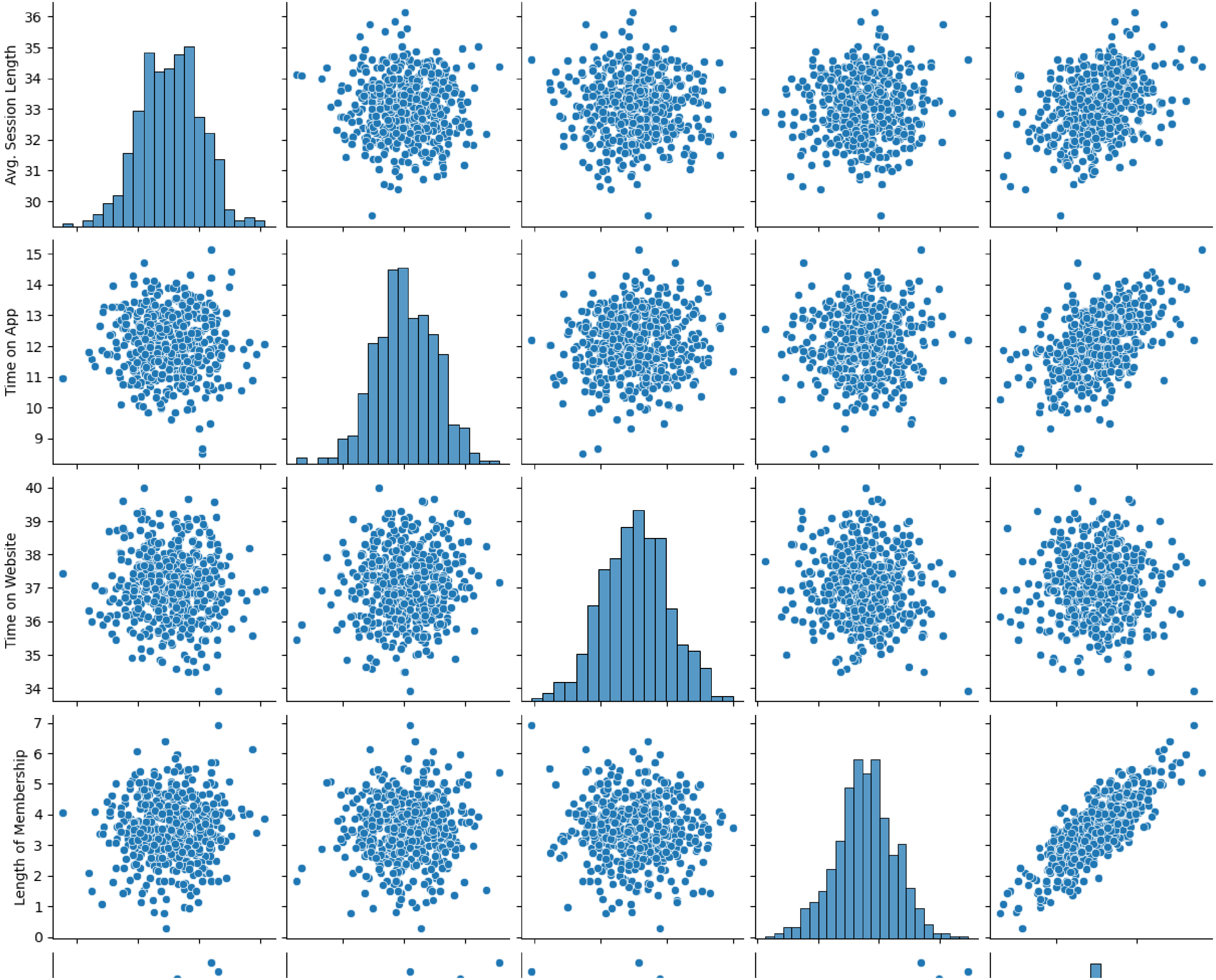
```
In [11]: df = df[['Avg. Session Length', 'Time on App',
                'Time on Website', 'Length of Membership', 'Yearly Amount Spent']]
```

In [12]:

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

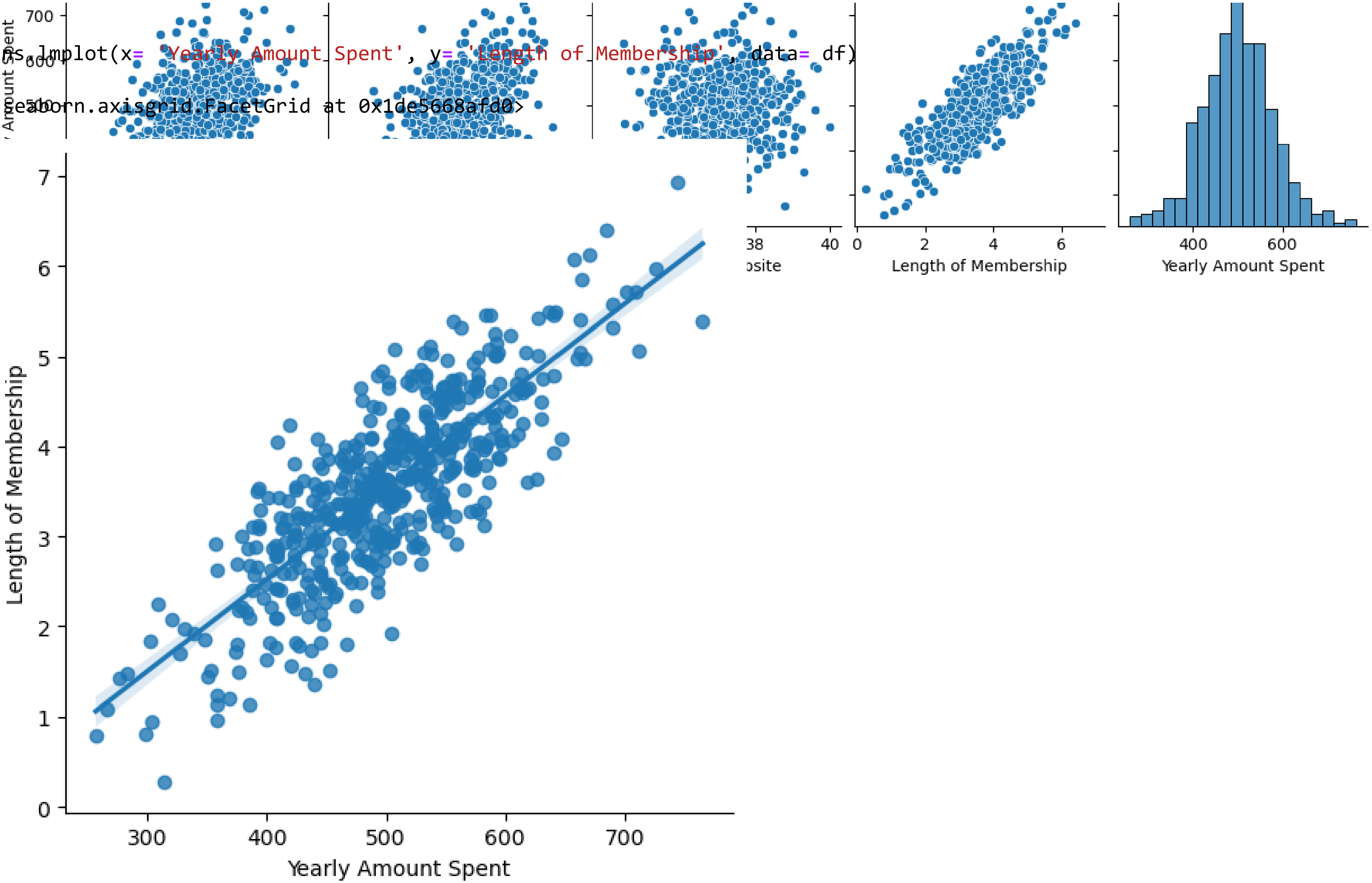
Out[12]:

```
<seaborn.axisgrid.PairGrid at 0x1de509ac1f0>
```


Avg. Session Length
36
35
34
33
32
31
30
Time on App
15
14
13
12
11
10
9
Time on Website
40
39
38
37
36
35
34
Length of Membership
7
6
5
4
3
2
1
0

In [13]:
```
sns.lmplot(x= 'Yearly Amount Spent', y= 'Length of Membership', data= df)
```

Out[13]:
```
<seaborn.axisgrid.FacetGrid at 0x1de5668afd0>
```



In [14]:
```
df.columns
```

Out[14]:
```
Index(['Avg. Session Length', 'Time on App', 'Time on Website',
       'Length of Membership', 'Yearly Amount Spent'],
      dtype='object')
```

In [15]:
```
X = df[['Avg. Session Length', 'Time on App', 'Time on Website',
       'Length of Membership']]
```

In [16]:
```
y = df['Yearly Amount Spent']
```

In [17]:
```
from sklearn.model_selection import train_test_split
```

In [18]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

In [19]:
```
from sklearn.linear_model import LinearRegression
```

In [20]:
```
lm = LinearRegression()
```

In [21]:
```
lm.fit(X_train, y_train)
```

Out[21]:
```
▾ LinearRegression
LinearRegression()
```

In [22]:
```
print(lm.intercept_)
```

```
-1047.9327822502394
```

In [23]:
```
lm.coef_
```

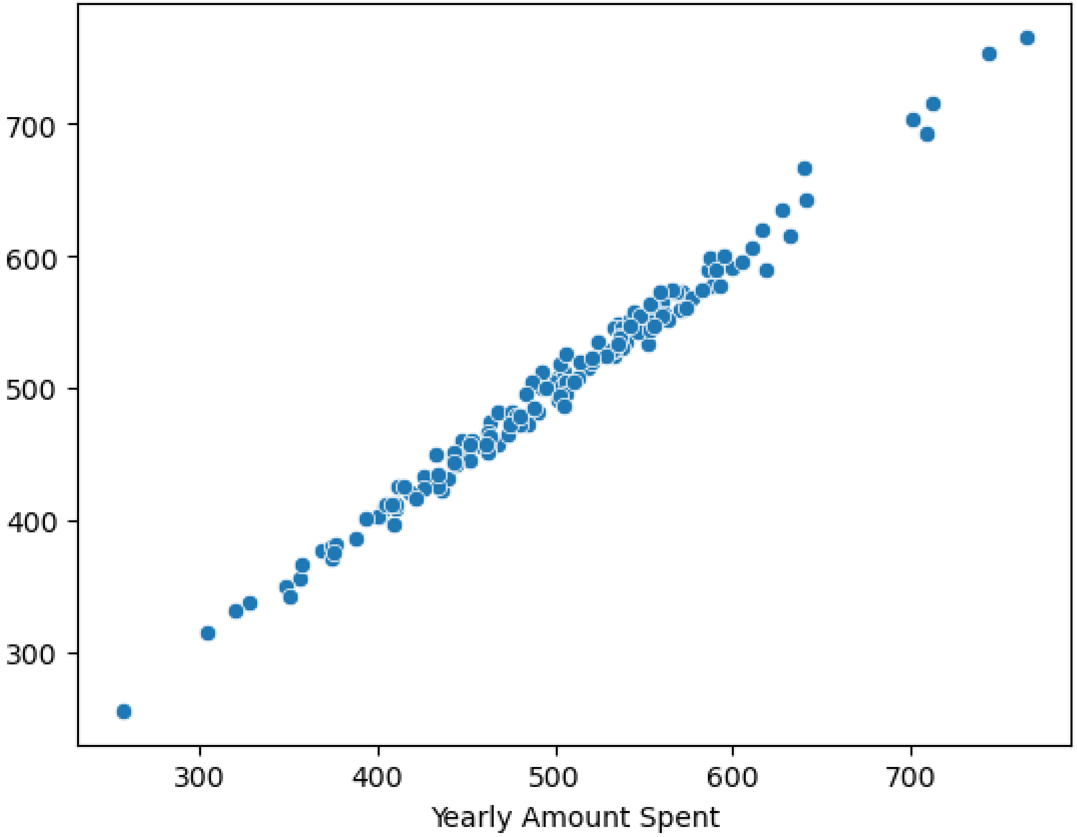
Out[23]:
```
array([25.98154972, 38.59015875,  0.19040528, 61.27909654])
```

```
In [24]: xd = pd.DataFrame (lm.coef_, X.columns, columns = ['Coeficiente'])
         xd
```

```
Out[24]:
                          Coeficiente
          Avg. Session Length   25.981550
                  Time on App   38.590159
              Time on Website    0.190405
         Length of Membership   61.279097
```

```
In [25]: predictions = lm.predict(X_test)
In [26]: sns.scatterplot(x = y_test, y= predictions)

Out[26]: <AxesSubplot:xlabel='Yearly Amount Spent'>
```



```
In [27]: from sklearn import metrics
```

```
In [28]: metrics.mean_absolute_error(y_test, predictions)

Out[28]: 7.2281486534308295
```

In [29]:

```
metrics.mean_squared_error(y_test, predictions)
```

Out[29]:

```
79.81305165097444
```

In [30]:

```
np.sqrt(metrics.mean_squared_error(y_test, predictions))
```

Out[30]:

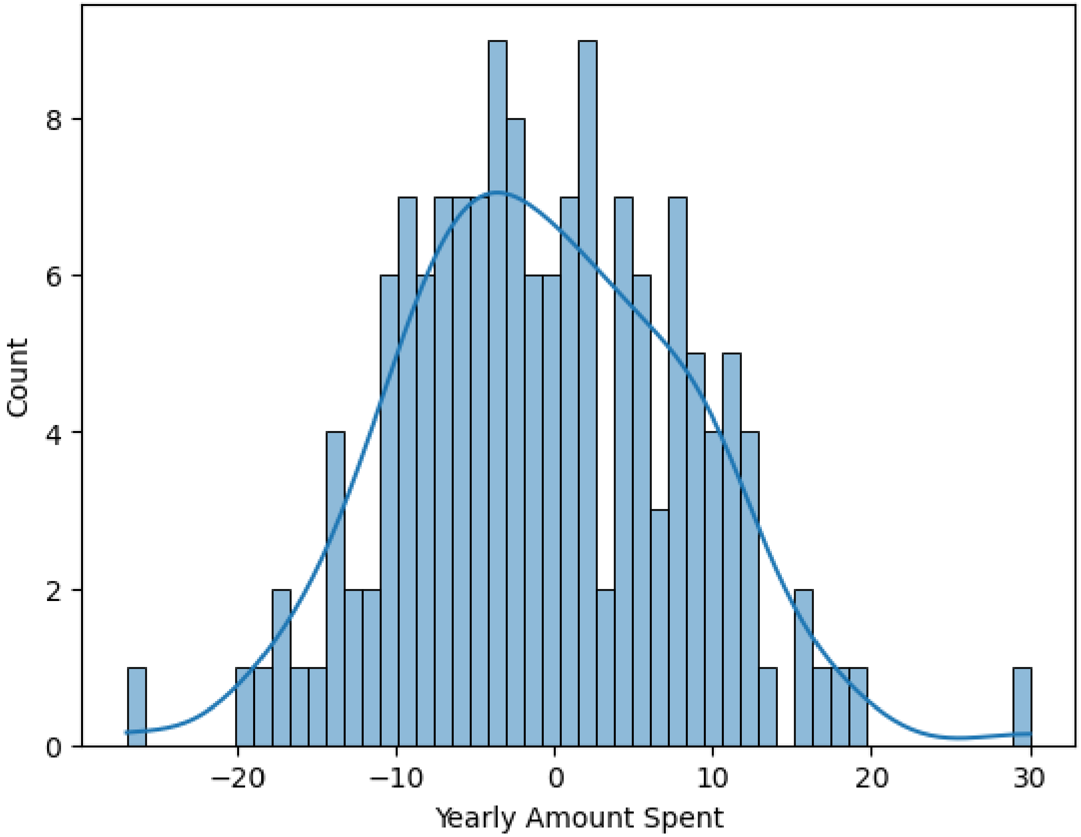
```
8.933815066978633
```

In [31]:

```
sns.histplot(y_test-predictions, bins= 50, kde = True)
```

Out[31]:

```
<AxesSubplot:xlabel='Yearly Amount Spent', ylabel='Count'>
```



In [32]:

```
xd
```

Out[32]:

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

In [ ]:

```

```