In [85]:
```
#IMporting Datasets
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings(action ='ignore',category=FutureWarning)
```

In [86]:
```
dataset=pd.read_csv('Ecommerce Customers (1)')
```

In [87]:
```
dataset.head()
```

Out[87]:

	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 [88]:
```
dataset.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 [95]:
```
X=dataset.drop(['Email','Address','Avatar'],axis=1)
y=X.iloc[:,-1]
X=X.iloc[:,:-1]
```

In [97]:
```
X
```

Out[97]:

	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
...	...	...	...	...
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

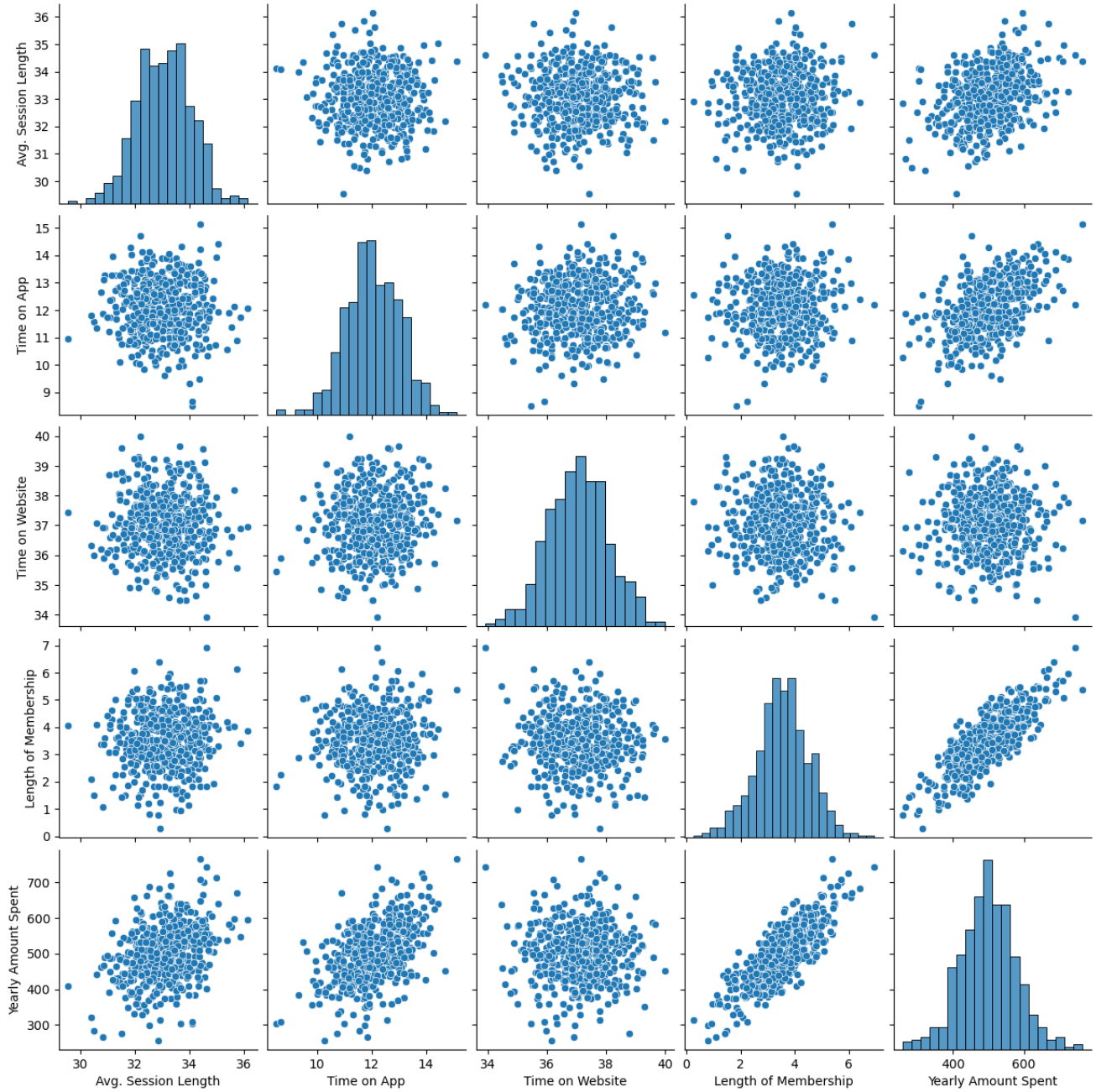
In [98]:
```
#Statical Summary
dataset.describe()
```

Out[98]:

	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 [107]:
```
#Figurign out the linear Relationship among the numerical varriables.
sns.pairplot(dataset,palette='cmap')
```

Out[107]: <seaborn.axisgrid.PairGrid at 0x1b2a5b07700>



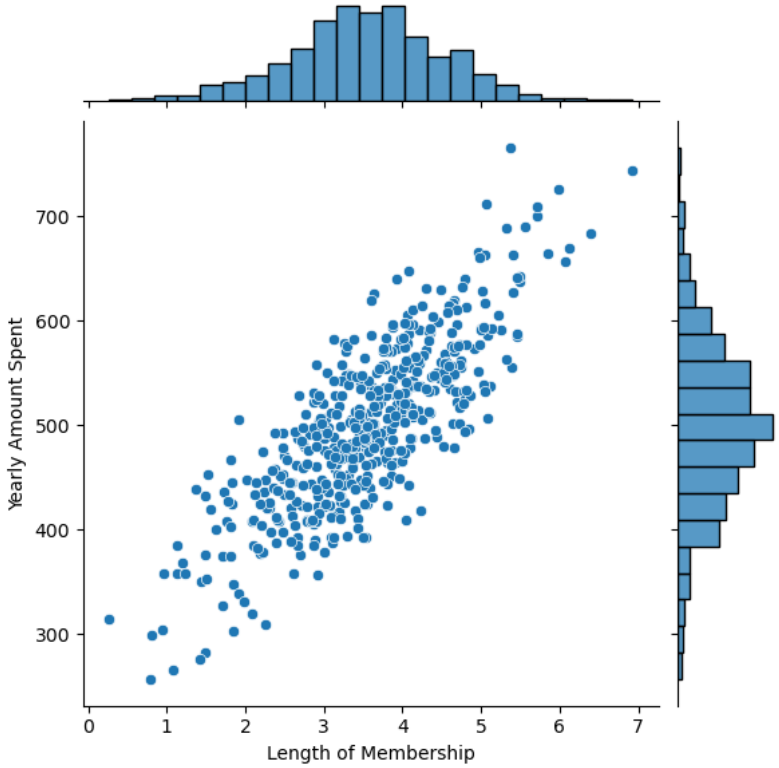
In [122]:
```
#From above data we can see the Length of Membership has strong positive relationship with Amount Spent on site by the customers.
```

In [101]:
```
dataset.columns
```

Out[101]:
```
Index(['Email', 'Address', 'Avatar', 'Avg. Session Length', 'Time on App',
       'Time on Website', 'Length of Membership', 'Yearly Amount Spent'],
      dtype='object')
```

In [111]:
```
sns.jointplot(x='Length of Membership',y='Yearly Amount Spent',data=dataset,palette='rainbow')
```

Out[111]:
```
<seaborn.axisgrid.JointGrid at 0x1b2a89d4670>
```



In [113]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=.3,random_state=42)
```

In [115]:
```
lm=LinearRegression()
lm.fit(X_train,y_train)
```

Out[115]:
```
LinearRegression()
```

In [116]:
```
prediction=lm.predict(X_test)
```

In [117]:
```
from sklearn.metrics import mean_squared_error, r2_score
```

```
Mean Squared Error of the linear model is: <function mean_squared_error at 0x000001B29FE85550>.
R Squared Error of the linear model is: <function r2_score at 0x000001B29FE858B0>.
```

In [121]:
```
print(f'Mean Squared Error of the linear model is: {round(mean_squared_error(y_test,prediction),2)}.')
print(f'R Squared Error of the linear model is: {round(r2_score(y_test,prediction),2)}.')
```

```
Mean Squared Error of the linear model is: 103.92.
R Squared Error of the linear model is: 0.98.
```

In [ ]:
```
```