In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
```

In [2]:
```
data = pd.read_csv("Ecommerce.csv")
```

In [3]:
```
data
```

Out[3]:

	Email	Address	Avatar	Avg. Session Length	Time on App
0	mstephenson@fernandez.com	835 Frank Tunnel\r\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655651
1	hduke@hotmail.com	4547 Archer Common\r\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109461
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\r\nCobbborough,...	Bisque	33.000915	11.330278
3	riverarebecca@gmail.com	1414 David Throughway\r\nPort Jason, OH 22070-...	SaddleBrown	34.305557	13.717514
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\r\nPort Jacobville, PR...	MediumAquaMarine	33.330673	12.795189
...	...	...	...	...	...
495	lewisjessica@craig-evans.com	4483 Jones Motorway Suite 872\r\nLake Jamiefur...	Tan	33.237660	13.566160
496	katrina56@gmail.com	172 Owen Divide Suite 497\r\nWest Richard, CA ...	PaleVioletRed	34.702529	11.695736
497	dale88@hotmail.com	0787 Andrews Ranch Apt. 633\r\nSouth Chadburgh...	Cornsilk	32.646777	11.499409
498	cwilson@hotmail.com	680 Jennifer Lodge Apt. 808\r\nBrendachester, ...	Teal	33.322501	12.391423
499	hannahwilson@davidson.com	49791 Rachel Heights Apt. 898\r\nEast Drewboro...	DarkMagenta	33.715981	12.418808

500 rows × 8 columns

In [38]:
```
data = pd.read_csv("Ecommerce.csv")
```

In [39]:
```
ndata = data.head()
```

In [40]:

```
ndata
```

Out[40]:

	Email	Address	Avatar	Avg. Session Length	Time on App	T W
0	mstephenson@fernandez.com	835 Frank Tunnel\r\nWrightmouth, MI 82180-9605	Violet	34.497268	12.655651	39.
1	hduke@hotmail.com	4547 Archer Common\r\nDiazchester, CA 06566-8576	DarkGreen	31.926272	11.109461	37.
2	pallen@yahoo.com	24645 Valerie Unions Suite 582\r\nCobbborough,...	Bisque	33.000915	11.330278	37
3	riverarebecca@gmail.com	1414 David Throughway\r\nPort Jason, OH 22070-...	SaddleBrown	34.305557	13.717514	36.
4	mstephens@davidson-herman.com	14023 Rodriguez Passage\r\nPort Jacobville, PR...	MediumAquaMarine	33.330673	12.795189	37.

In [41]:

```
ndata.describe()
```

Out[41]:

	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
count	5.000000	5.000000	5.000000	5.000000	5.000000
mean	33.412137	12.321618	37.643032	3.683537	529.792386
std	1.043940	1.088170	1.121068	0.754170	88.957399
min	31.926272	11.109461	36.721283	2.664034	392.204933
25%	33.000915	11.330278	37.110597	3.120179	487.547505
50%	33.330673	12.655651	37.268959	4.082621	581.852344
75%	34.305557	12.795189	37.536653	4.104543	587.951054
max	34.497268	13.717514	39.577668	4.446308	599.406092

In [42]:
```
data.isnull()
```

Out[42]:

	Email	Address	Avatar	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
495	False	False	False	False	False	False	False	False
496	False	False	False	False	False	False	False	False
497	False	False	False	False	False	False	False	False
498	False	False	False	False	False	False	False	False
499	False	False	False	False	False	False	False	False

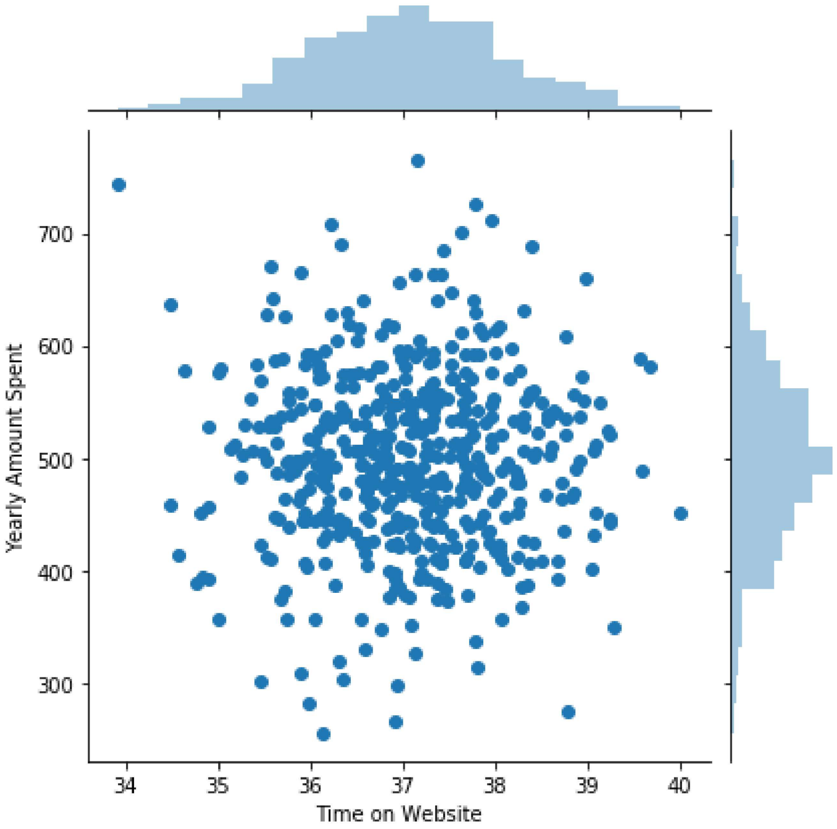
500 rows × 8 columns

In [43]:
```
data.info()
```

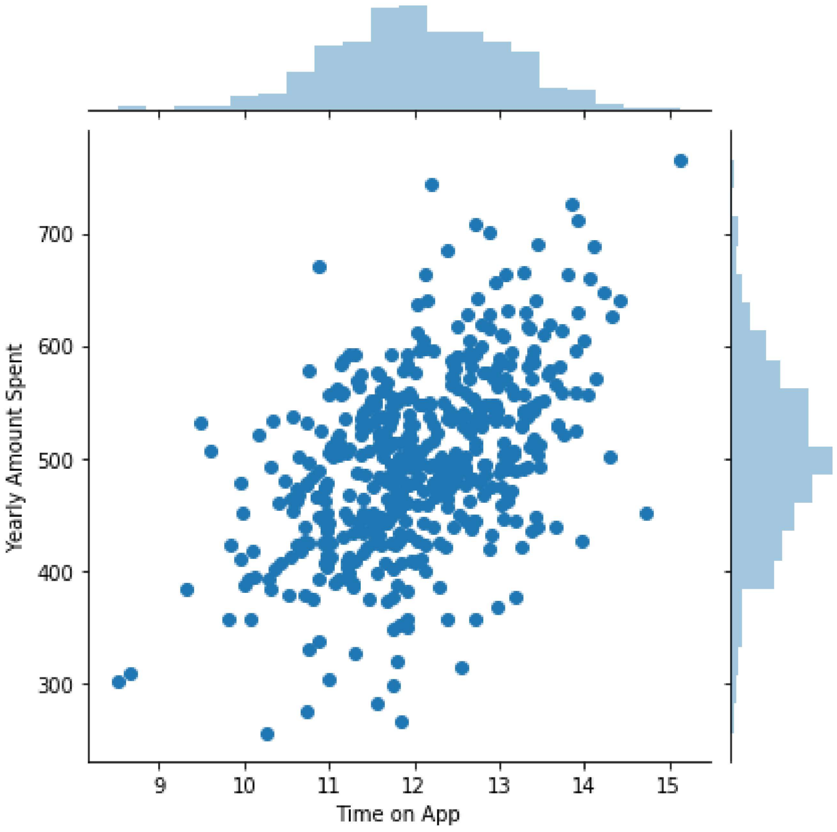
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 500 entries, 0 to 499
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Email                 500 non-null    object 
 1   Address               500 non-null    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 [44]:
```
import seaborn as sns
```

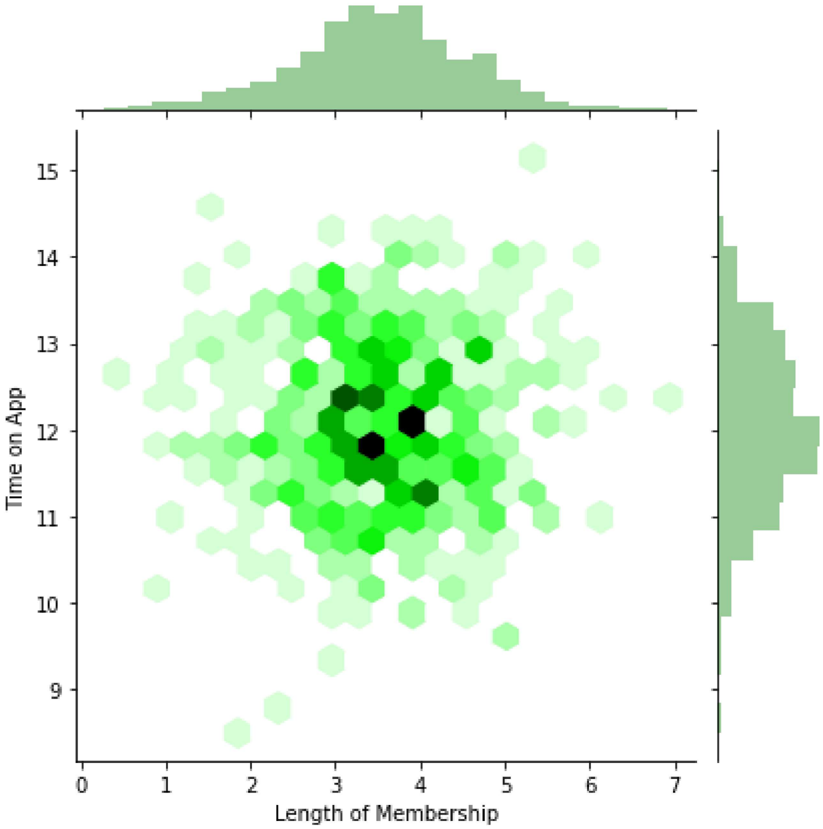
In [47]:
```
sns.jointplot(data = data, x="Time on Website", y = "Yearly Amount Spent")
plt.show()
```



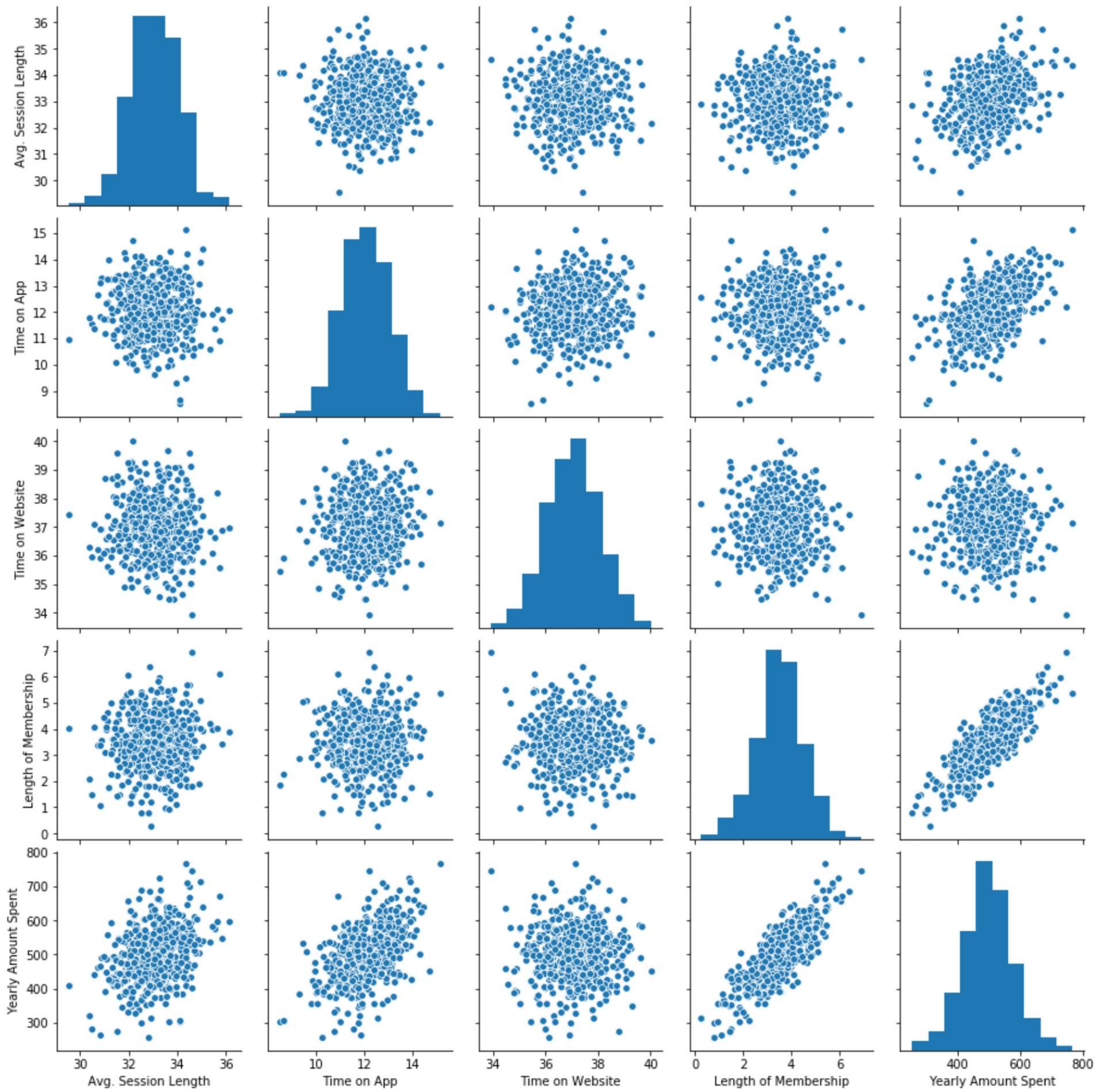
```
In [49]: sns.jointplot(data = data, x="Time on App", y = "Yearly Amount Spent")
         plt.show()
```



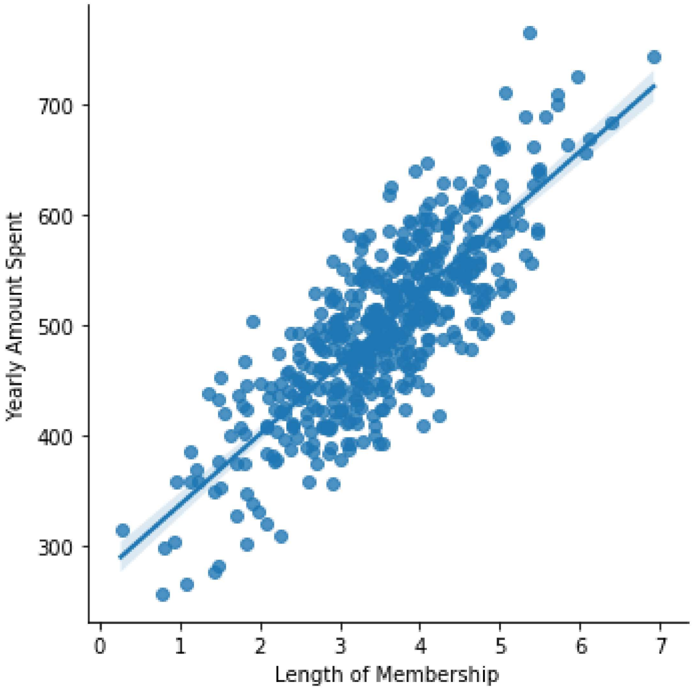
In [82]:
```
sns.jointplot(data = data, x="Length of Membership", y = "Time on App", kind = 'h
plt.show()
```



In [83]:
```
sns.pairplot(data)
plt.show()
```



In [113]:
```
sns.lmplot(data = data, x="Length of Membership", y = "Yearly Amount Spent")
plt.show()
```



In [114]:
```
data.columns
```

Out[114]:
```
Index(['Email', 'Address', 'Avatar', 'Avg. Session Length', 'Time on App',
       'Time on Website', 'Length of Membership', 'Yearly Amount Spent'],
      dtype='object')
```

In [115]:
```
X = data[['Avg. Session Length', 'Time on App', 'Time on Website', 'Length of Mem
```

In [116]:
```
X
```

Out[116]:

	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
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 [117]:
```
y = data['Yearly Amount Spent']
```

In [118]:
```
y
```

Out[118]:
```
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
```

In [119]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [120]:
```
model = LinearRegression()
```

In [121]:
```
model.fit(X, y)
```

Out[121]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [122]:
```
model.coef_
```

Out[122]:
```
array([-1.96909094e-14,  1.49880108e-14, -2.96095613e-15,  3.92307714e-14,
        1.00000000e+00])
```

In [123]:
```
model.intercept_
```

Out[123]:
```
6.821210263296962e-13
```

In [124]:
```
smallData = pd.DataFrame(data = model.coef_, index = X.columns, columns = ["Data
```

In [125]:
```
smallData
```

Out[125]:

	Data Coef
Avg. Session Length	-1.969091e-14
Time on App	1.498801e-14
Time on Website	-2.960956e-15
Length of Membership	3.923077e-14
Yearly Amount Spent	1.000000e+00

In [126]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, randor
```

In [127]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [128]:
```
X_train
```

Out[128]:

	Avg. Session Length	Time on App	Time on Website	Length of Membership	Yearly Amount Spent
202	31.525752	11.340036	37.039514	3.811248	443.965627
428	31.862741	14.039867	37.022269	3.738225	556.298141
392	33.258238	11.514949	37.128039	4.662845	549.131573
86	33.877779	12.517666	37.151921	2.669942	487.379306
443	33.025020	12.504220	37.645839	4.051382	561.516532
...	...	...	...	...	...
63	32.789773	11.670066	37.408748	3.414688	483.159721
326	33.217188	10.999684	38.442767	4.243813	505.230068
337	31.827979	12.461147	37.428997	2.974737	440.002748
11	33.879361	11.584783	37.087926	3.713209	522.337405
351	32.189845	11.386776	38.197483	4.808320	533.396554

350 rows × 5 columns

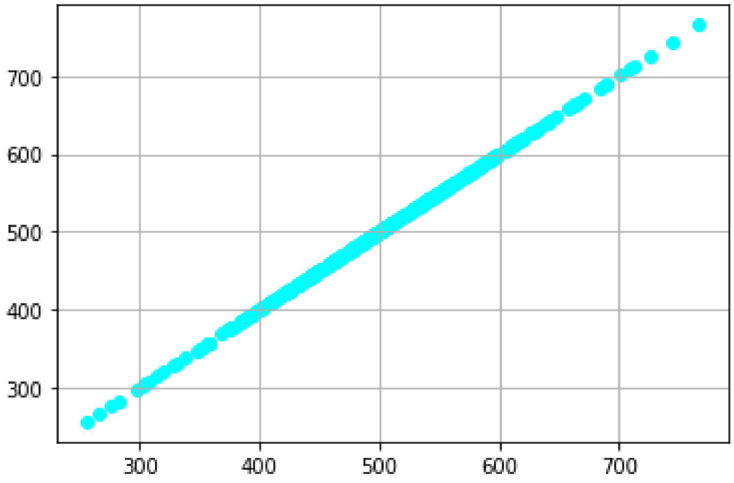
```
In [129]: model = LinearRegression()
```

```
In [130]: model.fit(X, y)
```

```
Out[130]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
In [131]: pre = model.predict(X)
```

```
In [132]: plt.scatter(y, pre, color = "cyan")
          plt.grid()
          plt.show()
```



```
In [107]: from sklearn import metrics
```

In [135]:
```
print('MAE: ', metrics.mean_absolute_error(y_test,pre))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-135-3f331ef597a1> in <module>
----> 1 print('MAE: ', metrics.mean_absolute_error(y_test,pre))

~\Anaconda3\lib\site-packages\sklearn\metrics\_regression.py in mean_absolute_e
rror(y_true, y_pred, sample_weight, multioutput)
    176     """
    177     y_type, y_true, y_pred, multioutput = _check_reg_targets(
--> 178         y_true, y_pred, multioutput)
    179     check_consistent_length(y_true, y_pred, sample_weight)
    180     output_errors = np.average(np.abs(y_pred - y_true),

~\Anaconda3\lib\site-packages\sklearn\metrics\_regression.py in _check_reg_targ
ets(y_true, y_pred, multioutput, dtype)
     82
     83     """
---> 84     check_consistent_length(y_true, y_pred)
     85     y_true = check_array(y_true, ensure_2d=False, dtype=dtype)
     86     y_pred = check_array(y_pred, ensure_2d=False, dtype=dtype)

~\Anaconda3\lib\site-packages\sklearn\utils\validation.py in check_consistent_l
ength(*arrays)
    210     if len(uniques) > 1:
    211         raise ValueError("Found input variables with inconsistent numbe
rs of"
--> 212                          " samples: %r" % [int(l) for l in lengths])
    213
    214

ValueError: Found input variables with inconsistent numbers of samples: [150, 5
00]
```

In [ ]: