Question 1

In [41]:
```
import numpy as np
import pandas as pd

import scipy
from scipy.cluster.hierarchy import dendrogram,linkage
from scipy.cluster.hierarchy import fcluster
from scipy.cluster.hierarchy import cophenet
from scipy.spatial.distance import pdist

import matplotlib.pyplot as plt
from pylab import rcParams
import seaborn as sb

import sklearn
from sklearn import datasets
from sklearn.cluster import AgglomerativeClustering
import sklearn.metrics as sm
from sklearn.preprocessing import scale
```

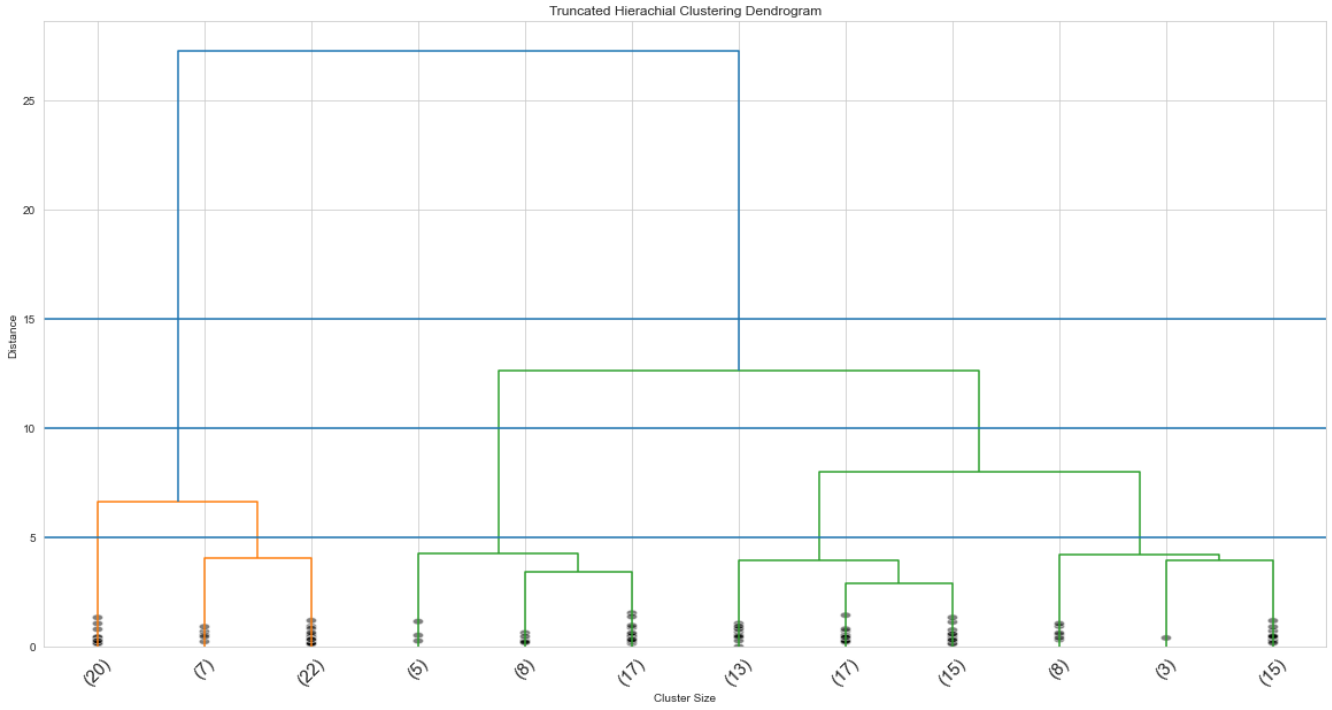
In [42]:
```
# i)
```

In [43]:
```
np.set_printoptions(precision=4,suppress=True)
%matplotlib inline
rcParams["figure.figsize"] =20,10
sb.set_style("whitegrid")
```

In [44]:
```
iris = datasets.load_iris()
data = scale(iris.data)
target = pd.DataFrame(iris.target)
variable_names = iris.feature_names
data[0:10]
```

Out[44]:
```
array([[-0.9007,  1.019 , -1.3402, -1.3154],
       [-1.143 , -0.132 , -1.3402, -1.3154],
       [-1.3854,  0.3284, -1.3971, -1.3154],
       [-1.5065,  0.0982, -1.2834, -1.3154],
       [-1.0218,  1.2492, -1.3402, -1.3154],
       [-0.5372,  1.9398, -1.1697, -1.0522],
       [-1.5065,  0.7888, -1.3402, -1.1838],
       [-1.0218,  0.7888, -1.2834, -1.3154],
       [-1.7489, -0.3622, -1.3402, -1.3154],
       [-1.143 ,  0.0982, -1.2834, -1.4471]])
```

In [45]:
```
z = linkage(data,"ward")
dendrogram(z,truncate_mode= "lastp", p =12, leaf_rotation=45,leaf_font_size=15, show_cor
plt.title("Truncated Hierachial Clustering Dendrogram")
plt.xlabel("Cluster Size")
plt.ylabel("Distance")
plt.axhline(y=15)
plt.axhline(5)
plt.axhline(10)
plt.show()
```



In [46]:
```
# ii)
```

In [47]:
```
k =3
HClustering = AgglomerativeClustering(n_clusters=k , affinity="euclidean",linkage="ward"
HClustering.fit(data)
sm.accuracy_score(target,HClustering.labels_)
```

Out[47]: 0.013333333333333334

In [48]:
```
k =3
HClustering = AgglomerativeClustering(n_clusters=k , affinity="euclidean",linkage="compl
HClustering.fit(data)
sm.accuracy_score(target,HClustering.labels_)
```

Out[48]: 0.013333333333333334

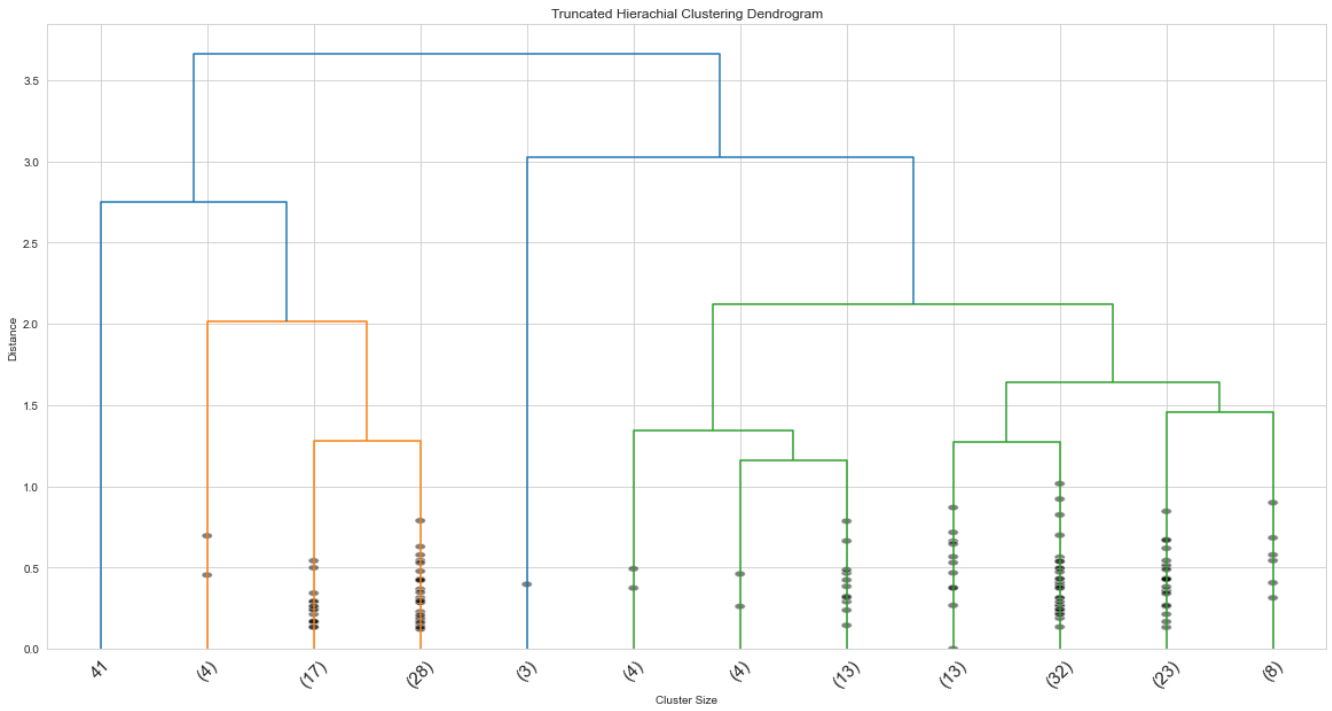
In [49]:
```
k =3
HClustering = AgglomerativeClustering(n_clusters=k , affinity="euclidean",linkage="avera
HClustering.fit(data)
sm.accuracy_score(target,HClustering.labels_)
```

Out[49]: 0.6866666666666666

In [50]:
```
k =3
HClustering = AgglomerativeClustering(n_clusters=k , affinity="euclidean",linkage="singl
HClustering.fit(data)
sm.accuracy_score(target,HClustering.labels_)
```

Out[50]: 0.0

In [51]:
```
z = linkage(data,"average")
dendrogram(z,truncate_mode= "lastp", p =12, leaf_rotation=45,leaf_font_size=15, show_cor
plt.title("Truncated Hierachial Clustering Dendrogram")
plt.xlabel("Cluster Size")
plt.ylabel("Distance")
plt.axhline(y=15)
plt.axhline(5)
plt.axhline(10)
plt.show()
```



Question 2

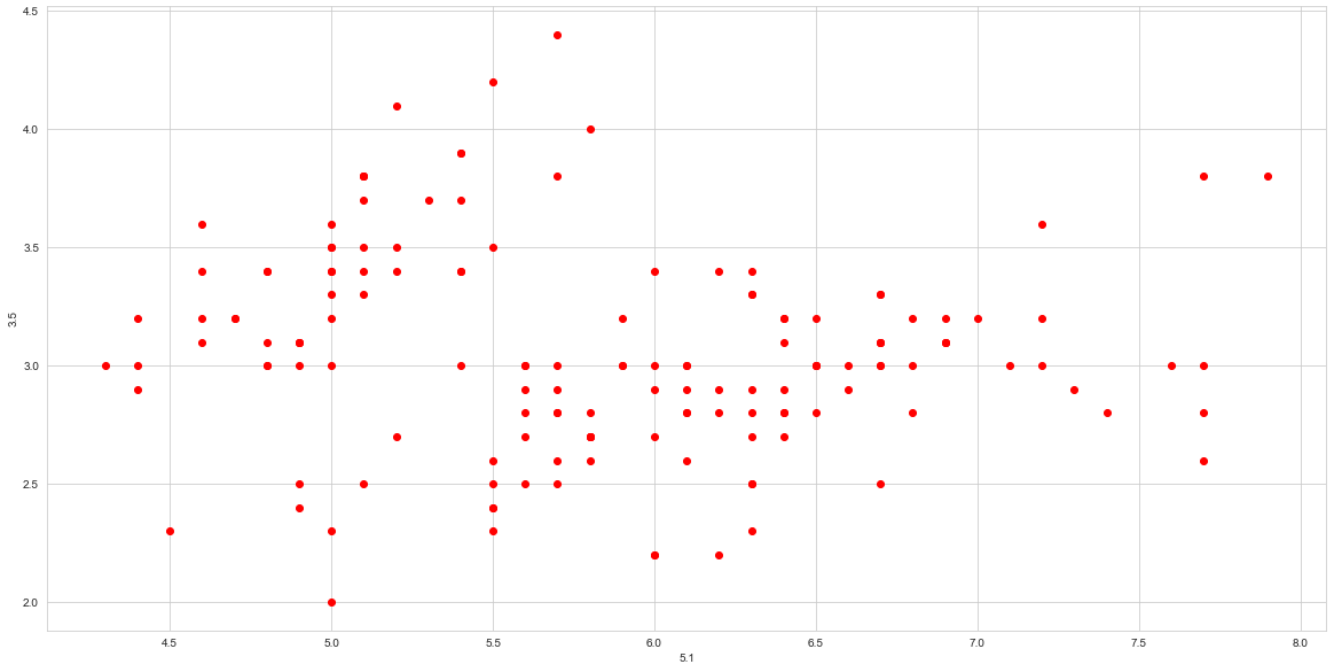
In [52]:
```
import random as rd
```

In [53]:
```
data = pd.read_csv('iris.csv')
data.head()
```

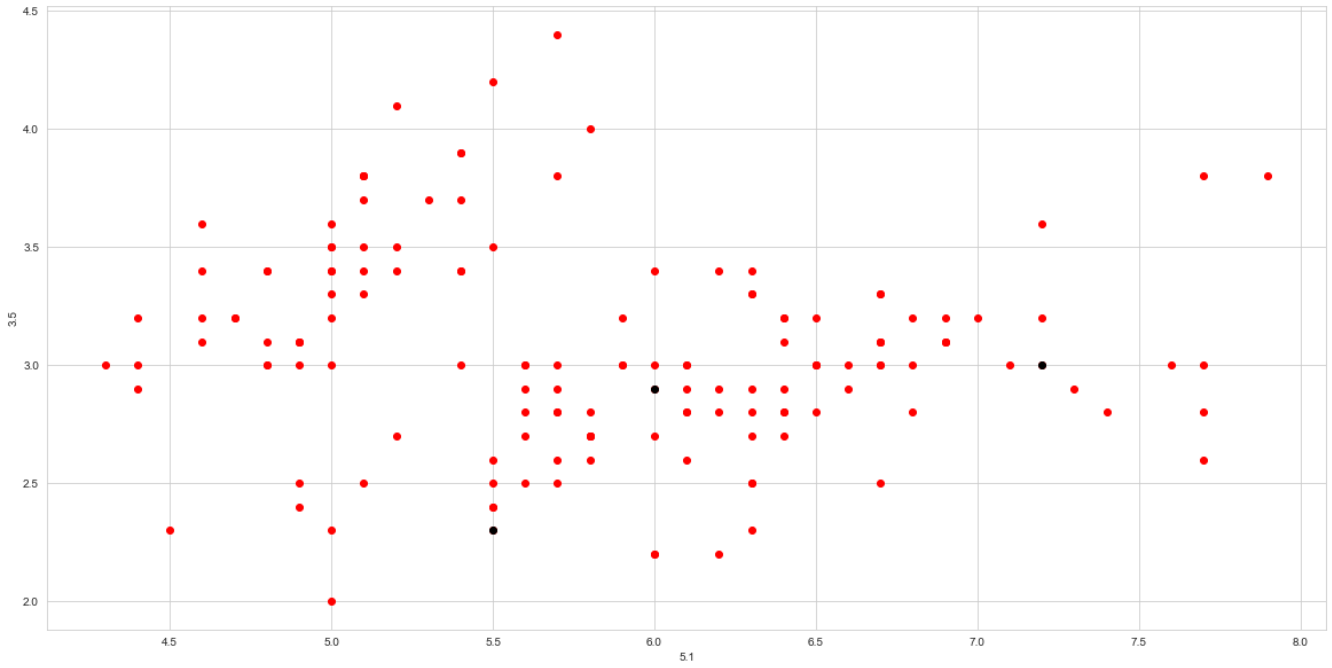
Out[53]:

	5.1	3.5	1.4	0.2	Iris-setosa
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

In [54]:
```
X = data[["5.1","3.5"]]
plt.scatter(X["5.1"],X["3.5"],c='red')
plt.xlabel('5.1')
plt.ylabel('3.5')
plt.show()
```



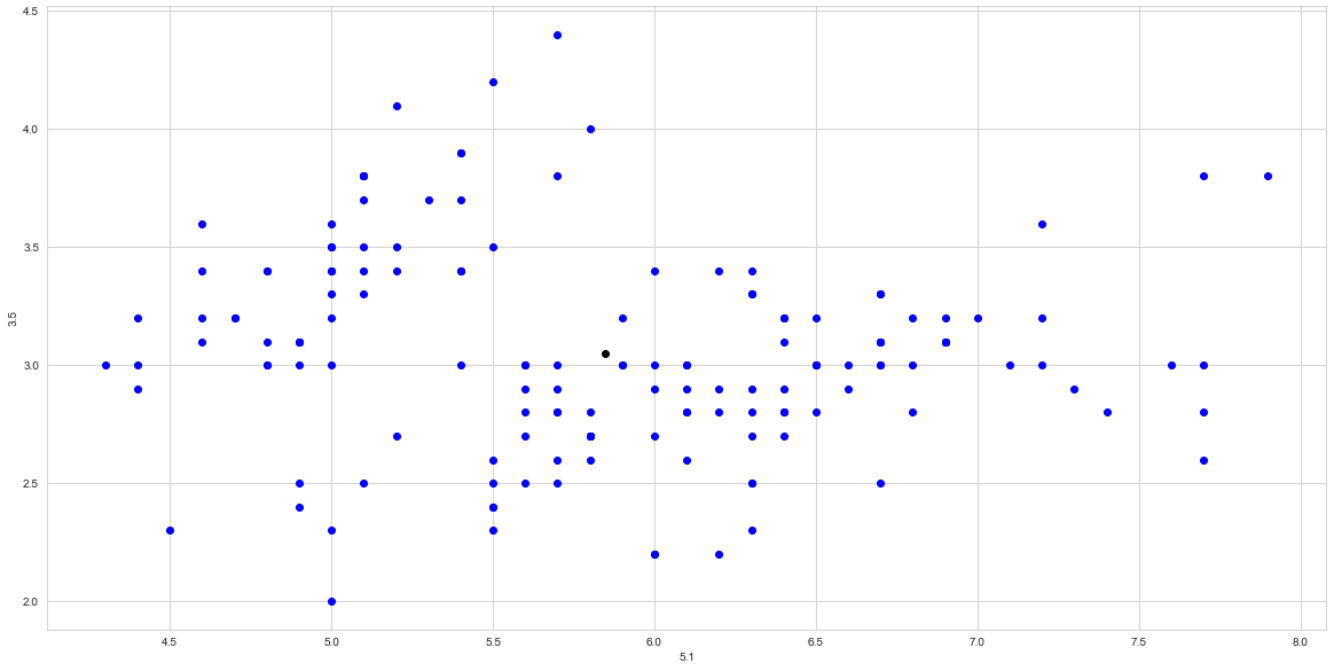
In [55]:
```
K=3
Centroids = (X.sample(n=K))
plt.scatter(X["5.1"],X["3.5"],c='red')
plt.scatter(Centroids["5.1"],Centroids["3.5"],c='black')
plt.xlabel('5.1')
plt.ylabel('3.5')
plt.show()
```



In [56]:
```
diff = 1
j=0
while(diff!=0):
    XD=X
    i=1
    for index1,row_c in Centroids.iterrows():
        ED=[]
        for index2,row_d in XD.iterrows():
            d1=(row_c["5.1"]-row_d["3.5"])**2
            d2=(row_c["3.5"]-row_d["3.5"])**2
            d=np.sqrt(d1+d2)
            ED.append(d)
        X[i]=ED
        i=i+1

    C=[]
    for index,row in X.iterrows():
        min_dist=row[1]
        pos=1
        for i in range(K):
            if row[i+1] < min_dist:
                min_dist = row[i+1]
                pos=i+1
        C.append(pos)
    X["Cluster"]=C
    Centroids_new = X.groupby(["Cluster"]).mean()[["3.5","5.1"]]
    if j == 0:
        diff=1
        j=j+1
    else:
        diff = (Centroids_new['3.5'] - Centroids['3.5']).sum() + (Centroids_new['5.1'] -
        print(diff.sum())
    Centroids = X.groupby(["Cluster"]).mean()[["3.5","5.1"]]
```

```
<ipython-input-56-b7e5d0e14beb>:13: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  X[i]=ED

-1.3997767123287677
0.7718232158988254
-0.2518906531347169
0.0
```

In [57]:
```
color=['blue','green','red']
for k in range(K):
    data=X[X["Cluster"]==k+1]
    plt.scatter(data["5.1"],data["3.5"],c=color[k])
plt.scatter(Centroids["5.1"],Centroids["3.5"],c='black')
plt.xlabel('5.1')
plt.ylabel('3.5')
plt.show()
```



Question 3

In [58]:
```
# 3.1
```

In [59]:
```
import nltk
from nltk.corpus import stopwords

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer

import pandas as pd
import string
import seaborn as sns
```

In [60]:
```
df = pd.read_csv("SMSSpamCollection", names=["label","message"])
df.head(2)
```

Out[60]:

	label	message
0	ham\tGo until jurong point	crazy.. Available only in bugis n great world...
1	ham\tOk lar... Joking wif u oni...	NaN

In [61]:
```
df.info()
df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5574 entries, 0 to 5573
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   label    5574 non-null   object
 1   message  1318 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
```

Out[61]:

	label	message
count	5574	1318
unique	4969	1153
top	ham\tSorry	I'll call later
freq	52	30

In [62]:
```
df.groupby('label').describe()
```

Out[62]:

	message			
	count	unique	top	freq
label				
ham\t	4	4		1
ham\t <#> in mca. But not conform.	0	0	NaN	NaN
ham\t <#> mins but i had to stop somewhere first.	0	0	NaN	NaN
ham\t <DECIMAL> m but its not a common car here so its better to buy from china or asia. Or if i find it less expensive. I.ll holla	0	0	NaN	NaN
ham\t and picking them up from various points	0	0	NaN	NaN
...	...	...	...	...
spam\ttodays vodafone numbers ending with 0089(my last four digits) are selected to received a £350 award. If your number matches please call 09063442151 to claim your £350 award	0	0	NaN	NaN
spam\tu r a winner U ave been specially selected 2 receive £1000 cash or a 4* holiday (flights inc) speak to a live operator 2 claim 0871277810710p/min (18)	0	0	NaN	NaN
spam\tu r subscribed 2 TEXTCOMP 250 wkly comp. 1st wk?s free question follows	1	1	subsequent wks charged@150p/msg.2 unsubscribe...	1
spam\twamma get laid?want real doggin locations sent direct to your mobile? join the UKs largest dogging network. txt dogs to 69696 now!nyt. ec2a. 3lp £1.50/msg.	0	0	NaN	NaN
spam\twe tried to contact you re your response to our offer of a new nokia fone and camcorder hit reply or call 08000930705 for delivery	0	0	NaN	NaN

4969 rows × 4 columns

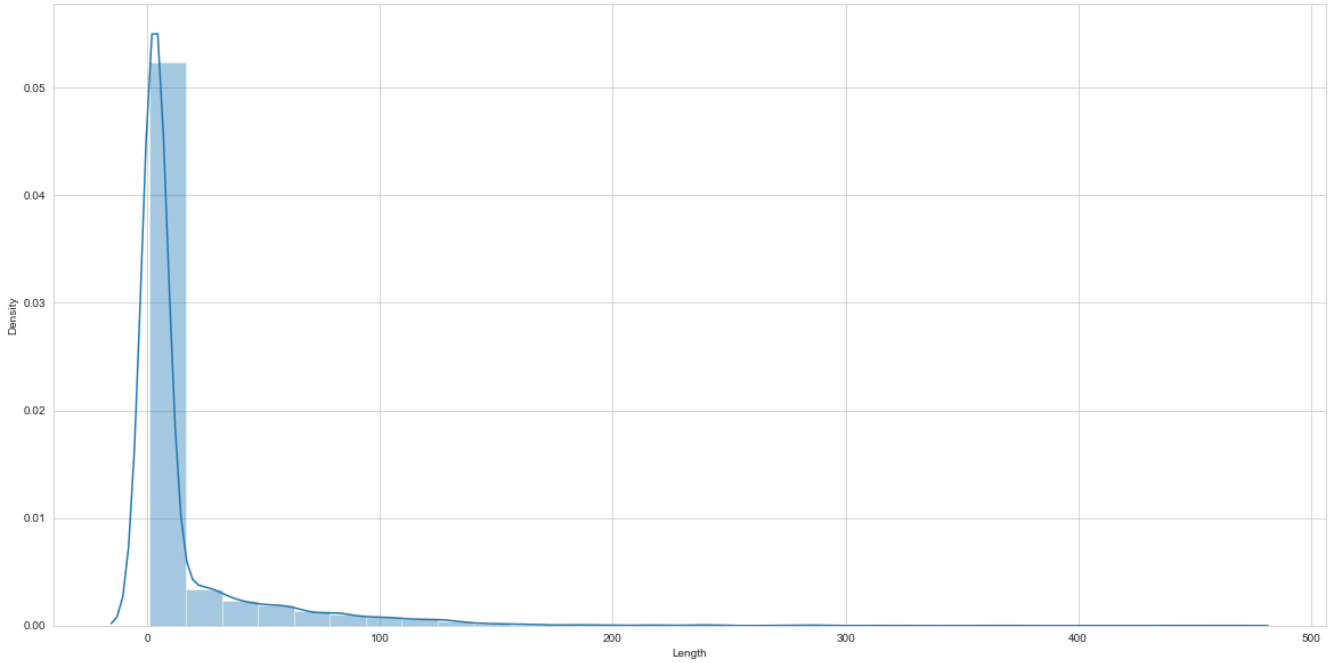
In [63]:
```
df['message'] = df['message'].apply(str)
```

In [64]:
```
df["Length"] = df["message"].apply(len)
```

In [65]:
```
sns.distplot(df["Length"], bins=30)
```

```
C:\Users\Nupur goel\anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[65]: <AxesSubplot:xlabel='Length', ylabel='Density'>



In [66]:
```
df["Length"].max()
```

Out[66]: 465

In [67]:
```
df[df["Length"]==465]["message"].iloc[0]
```

Out[67]: " hope you are having a nice day. I wanted to bring it to your notice that I have been late in paying rent for the past few months and have had to pay a $ <#> charge. I felt it would be inconsiderate of me to nag about something you give at great cost to yourself and that's why i didnt speak up. I however am in a recession and wont be able to pay the charge this month hence my askin well ahead of month's end. Can you please help. Thank you for everything."

In [68]:
```
df[df["Length"] == df["Length"].min()]["message"].iloc[0]
```

Out[68]: ' '

In [69]:
```
df.head(1)
```

Out[69]:

	label	message	Length
0	ham\tGo until jurong point	crazy.. Available only in bugis n great world...	89