

```
In [1]: import seaborn as sb
from matplotlib import pyplot as plt
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: import os
os.getcwd()
```

```
Out[2]: 'C:\\Users\\KHAN IZHAN'
```

```
In [3]: #Load the data set
df1=pd.read_csv("AS.csv")
```

```
In [4]: df1.head()
# first five roecod you can put the under the paranthesis 10 its show first ten rec
```

```
Out[4]:
```

	DATE	DESCRIPTION	UNIT	PLACE
0	Friday, April 1, 2022	Beef	1	Kolenchery
1	Friday, April 1, 2022	Fish	1	Kolenchery
2	Sunday, April 3, 2022	Beef	2	Tripunithura
3	Tuesday, April 5, 2022	Prawns	1	Bangalore
4	Wednesday, April 6, 2022	Dates	1	Mangalore

```
In [5]: df1.tail() #last five record its also same like tail(10)
```

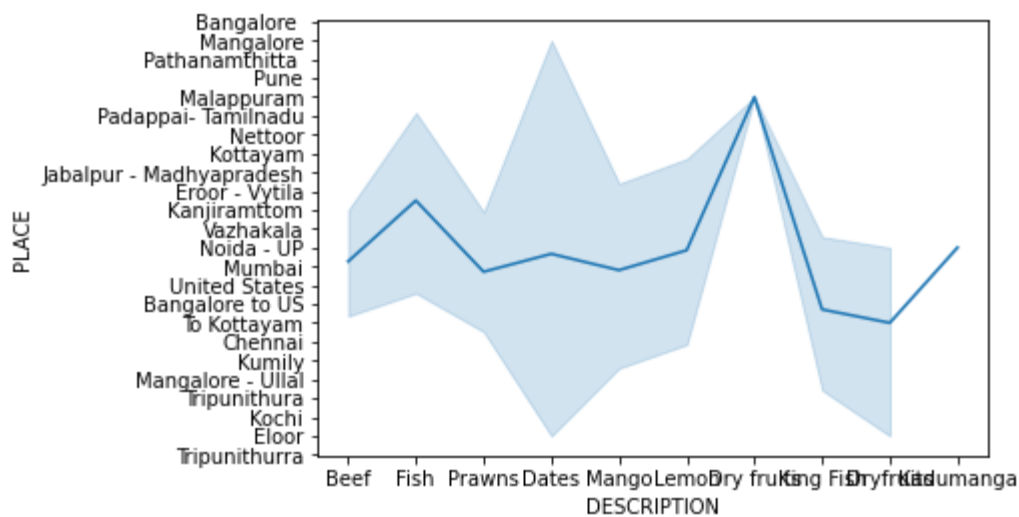
```
Out[5]:
```

	DATE	DESCRIPTION	UNIT	PLACE
106	Tuesday, July 5, 2022	Lemon	4	NaN
107	Friday, July 8, 2022	Beef	1	NaN
108	Friday, July 15, 2022	Beef	1	NaN
109	Friday, July 15, 2022	Prawns	1	NaN
110	Friday, July 15, 2022	Lemon	1	NaN

```
In [6]: df1.shape #its show the how many record have like row and column
```

```
Out[6]: (111, 4)
```

```
In [7]: #NOW CREATE A LINEPLOT
sb.lineplot(x="DESCRIPTION",y="PLACE",data=df1)
plt.figure(figsize=(150,70))# change the figure size
plt.show()
```

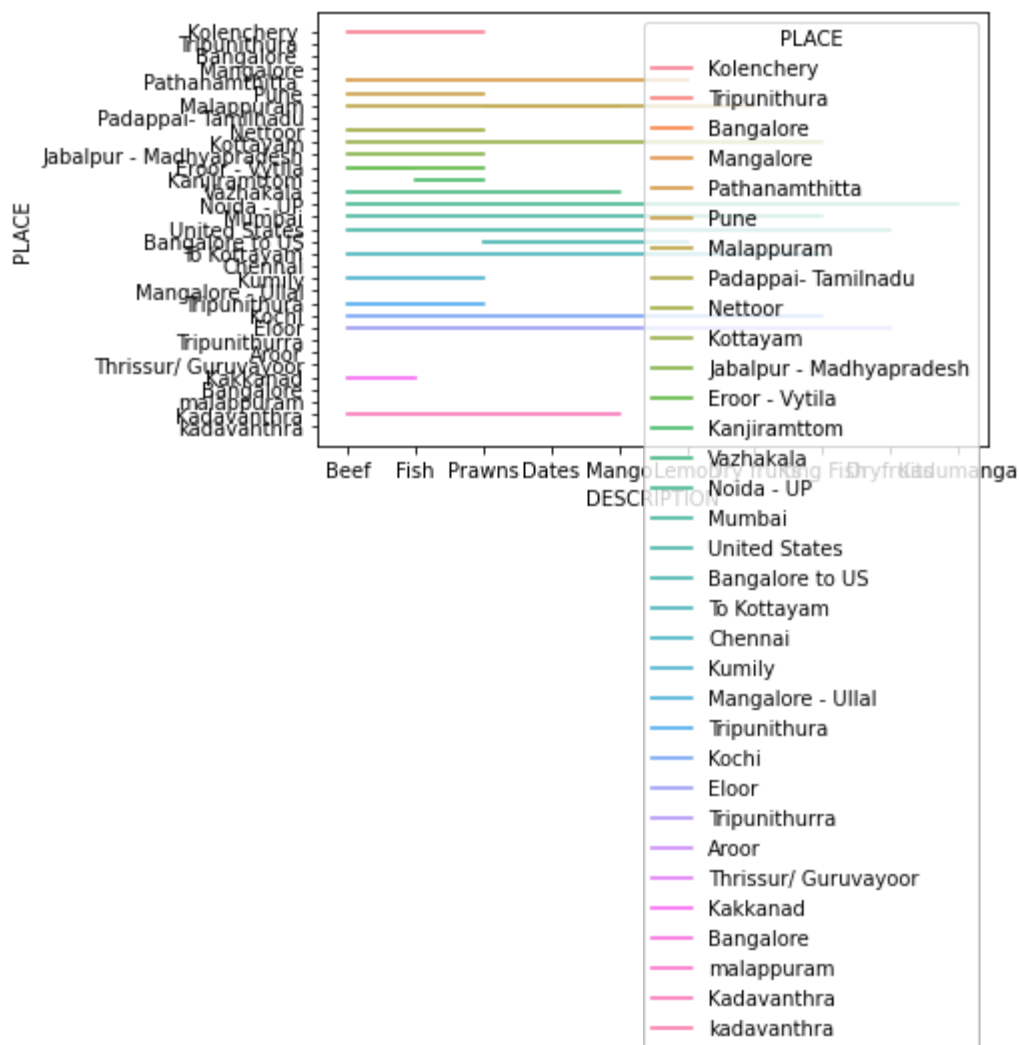


<Figure size 10800x5040 with 0 Axes>

```
In [8]: df1["DESCRIPTION"].value_counts()
```

```
Out[8]: Beef          32
Prawns             31
Fish               13
Mango              10
Lemon               9
King Fish          7
Dates               3
Dryfruits           3
Dry fruits          2
Kadumanga           1
Name: DESCRIPTION, dtype: int64
```

```
In [9]: #now use hue means show colour determine line
sb.lineplot(x="DESCRIPTION",y="PLACE",data=df1,hue="PLACE")
plt.figure(figsize=(180,70))
plt.show()
```



<Figure size 12960x5040 with 0 Axes>

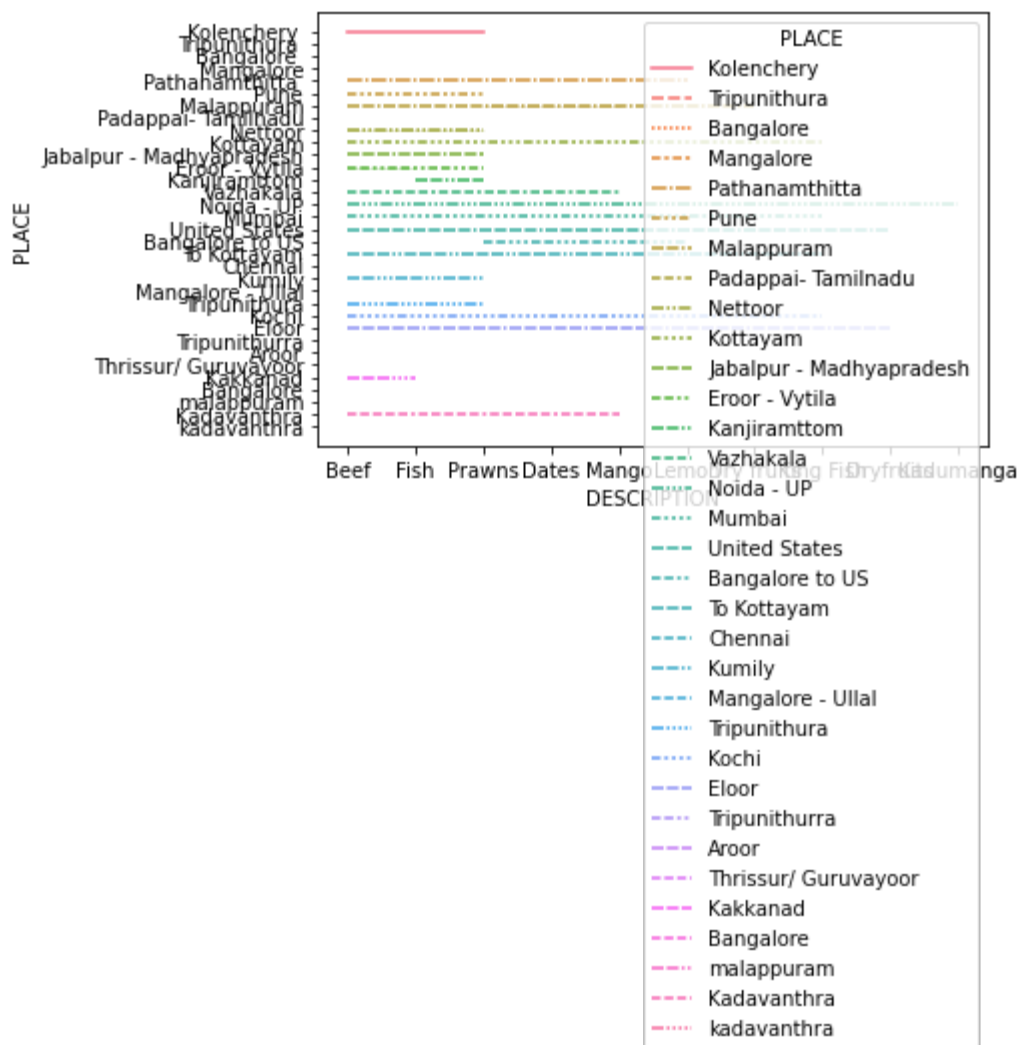
```
In [10]: df1["PLACE"].value_counts()
```

```
Out[10]:
```

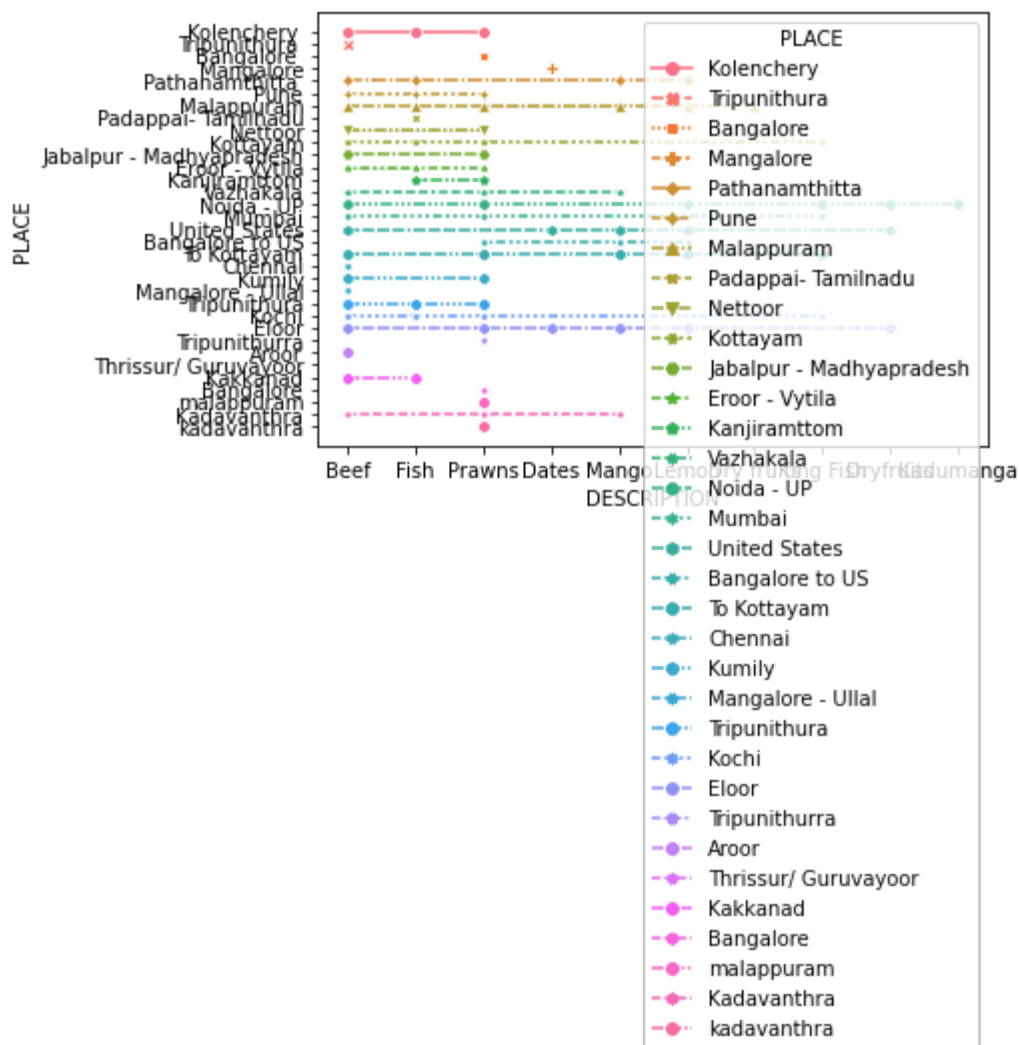
Malappuram	10
Noida - UP	6
Eloor	6
Kottayam	6
Mumbai	6
United States	5
Kochi	5
To Kottayam	5
Eroor - Vytila	5
Pathanamthitta	4
Nettoor	4
Bangalore to US	4
Kadavanthra	3
Tripunithura	3
Kolenchery	3
Vazhakala	3
Pune	3
Kanjiramttom	2
Jabalpur - Madhyapradesh	2
Chennai	2
Kumily	2
Mangalore - Ullal	2
Kakkanad	2
Thrissur/ Guruvayoor	1
Bangalore	1
malappuram	1
Bangalore	1
Mangalore	1
Aroor	1
Tripunithurra	1
Tripunithura	1
Padappai- Tamilnadu	1
kadavanthra	1

Name: PLACE, dtype: int64

```
In [32]: #style change the line style
sb.lineplot(x="DESCRIPTION",y="PLACE",data=df1,hue="PLACE",style="PLACE")
#plt.figure(figsize=(5,3))
plt.show()
```



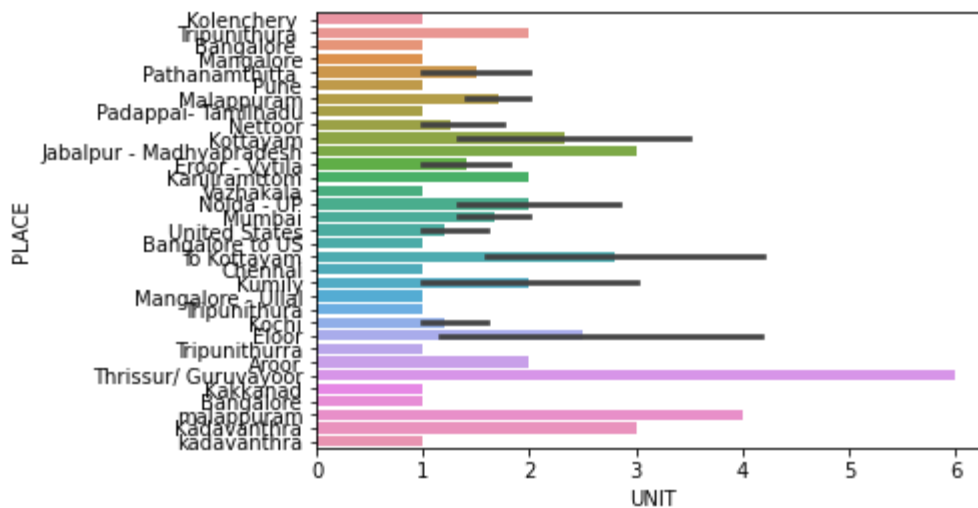
```
In [12]: sb.lineplot(x="DESCRIPTION",y="PLACE",data=df1,hue="PLACE",style="PLACE",markers=Tr
plt.figure(figsize=(150,70))
plt.show()
```



<Figure size 10800x5040 with 0 Axes>

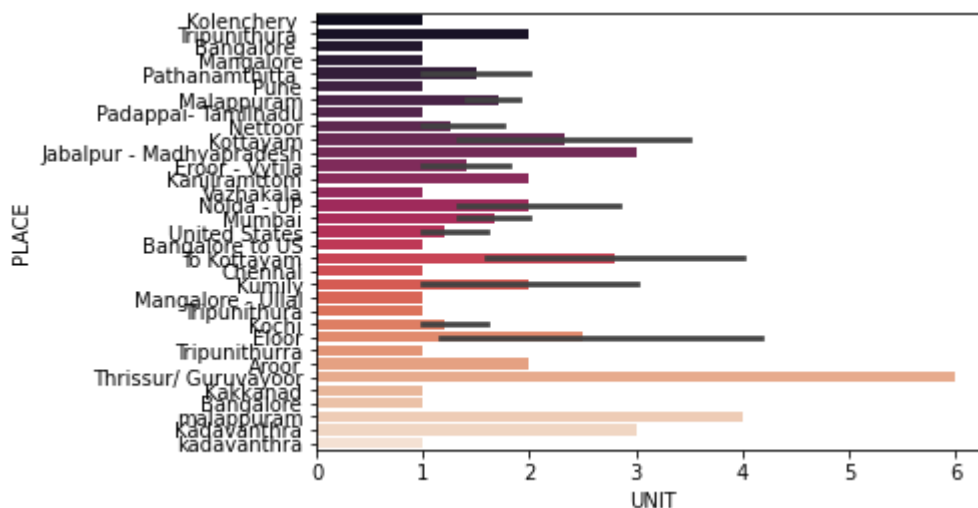
BAR PLOT

```
In [13]: sb.barplot(x="UNIT",y="PLACE",data=df1) #bar plot allow numerice values like catoga
plt.figure(figsize=(150,70))
plt.show()
```



<Figure size 10800x5040 with 0 Axes>

```
In [14]: sb.barplot(x="UNIT",y="PLACE",palette="rocket",data=df1)
plt.figure(figsize=(150,70))
plt.show()
```



<Figure size 10800x5040 with 0 Axes>

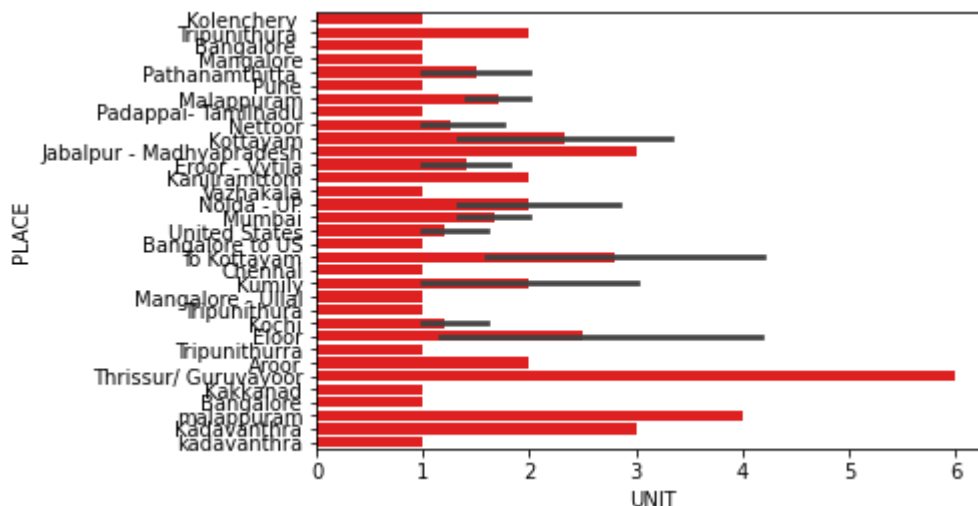
```
In [33]: df1["UNIT"].value_counts()
```

```
Out[33]: 1    63
         2    31
         3     7
         4     7
         6     2
         5     1
         Name: UNIT, dtype: int64
```

```
In [35]: df1["PLACE"].value_counts()
```

```
Out[35]: Malappuram          10
         Noida - UP          6
         Eloor              6
         Kottayam           6
         Mumbai             6
         United States      5
         Kochi              5
         To Kottayam        5
         Eroor - Vyttila    5
         Pathanamthitta     4
         Nettoor            4
         Bangalore to US    4
         Kadavanthra        3
         Tripunithura       3
         Kolenchery         3
         Vazhakala          3
         Pune               3
         Kanjiramttom       2
         Jabalpur - Madhya Pradesh 2
         Chennai           2
         Kumily             2
         Mangalore - Ullal  2
         Kakkanad           2
         Thrissur/ Guruvayoor 1
         Bangalore         1
         malappuram        1
         Bangalore         1
         Mangalore         1
         Aroor             1
         Tripunithurra     1
         Tripunithura      1
         Padappai- Tamilnadu 1
         kadavanthra       1
         Name: PLACE, dtype: int64
```

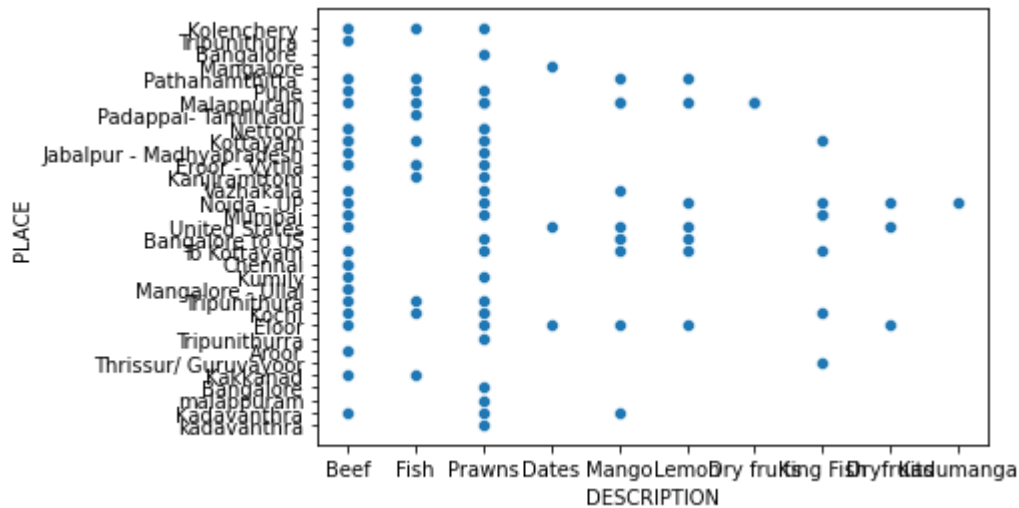
```
In [15]: #you can put your own colour choice you can use the color attributes all colours are
         sb.barplot(x="UNIT",y="PLACE",color="RED",data=df1)
         plt.figure(figsize=(150,70))
         plt.show()
```



<Figure size 10800x5040 with 0 Axes>

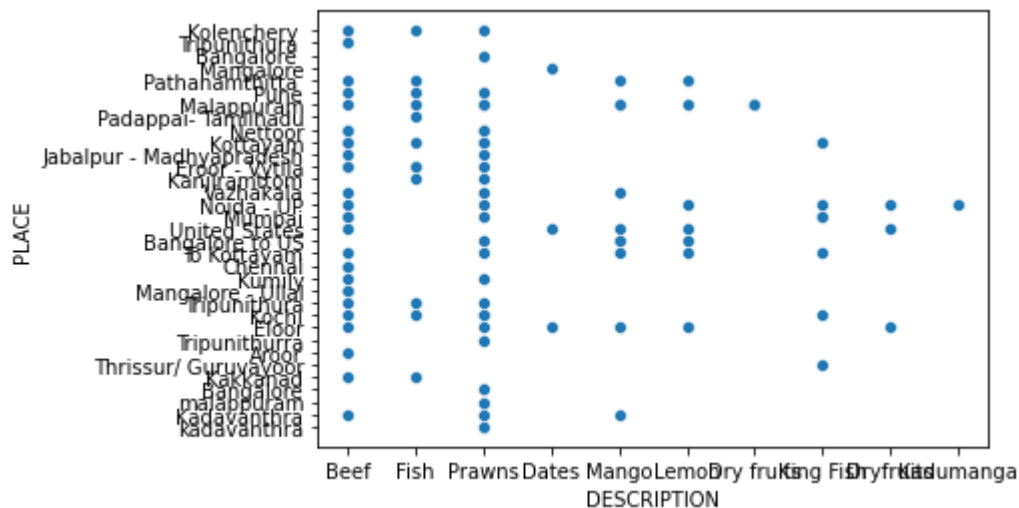
Scatterplot

```
In [16]: #you can check the relation between input and ouput Like UNIT AND PLACE
sb.scatterplot(x="DESCRIPTION",y="PLACE",data=df1)
plt.figure(figsize=(150,70))
plt.show()
```



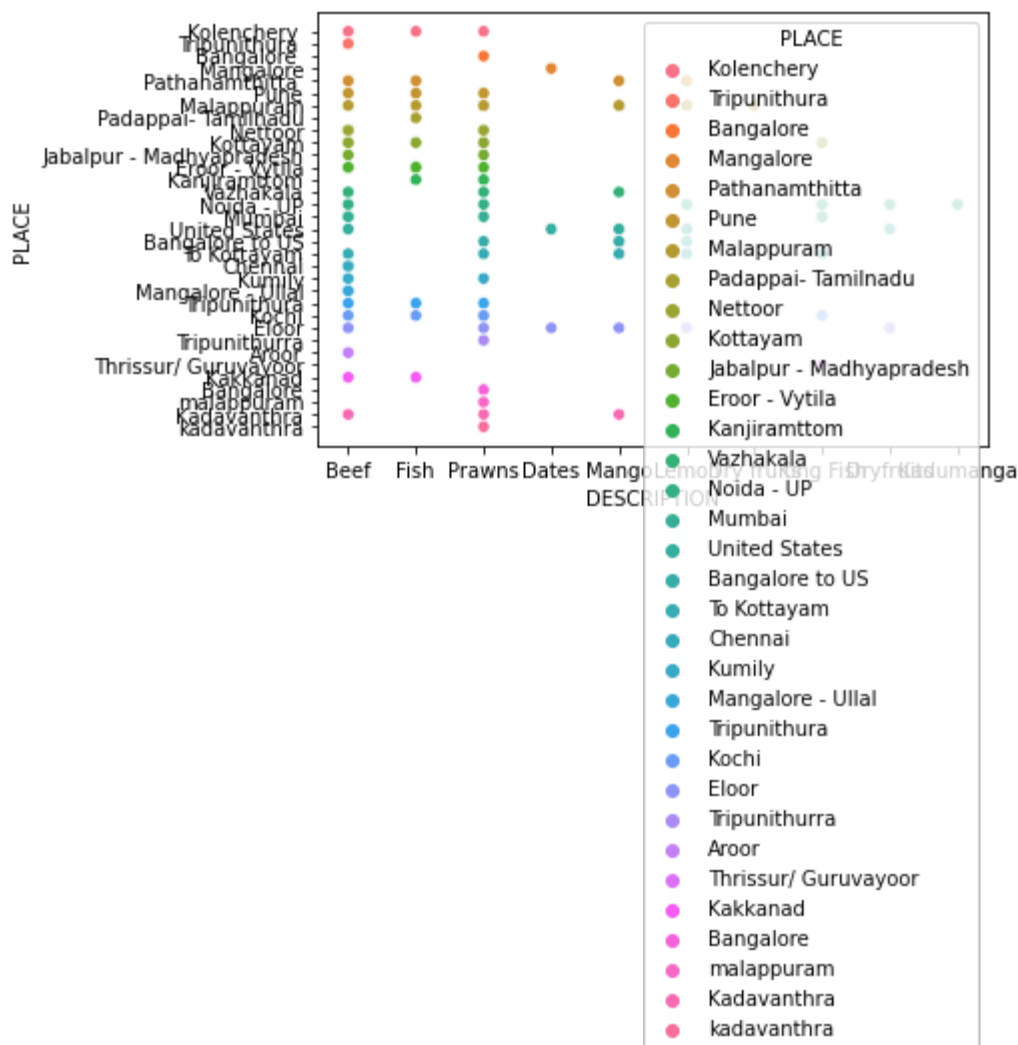
<Figure size 10800x5040 with 0 Axes>

```
In [17]: sb.scatterplot(x="DESCRIPTION",y="PLACE",data=df1)
plt.figure(figsize=(150,70))
plt.show()
```



<Figure size 10800x5040 with 0 Axes>

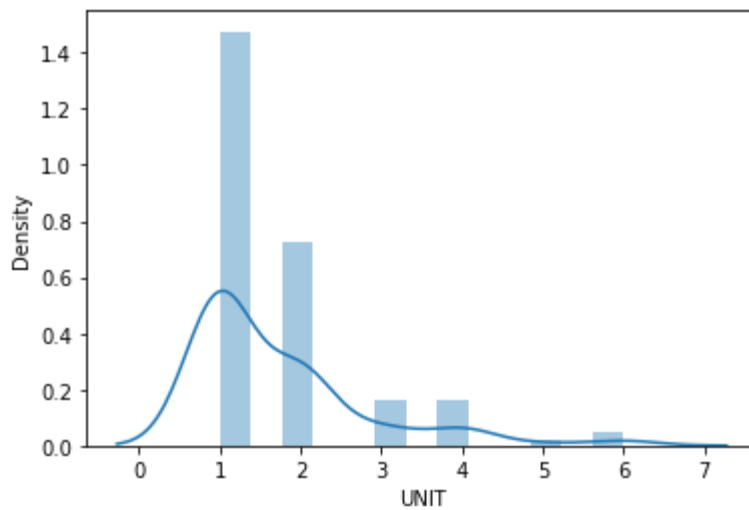
```
In [18]: sb.scatterplot(x="DESCRIPTION",y="PLACE",hue="PLACE",data=df1)
plt.figure(figsize=(150,70))
plt.show()
```



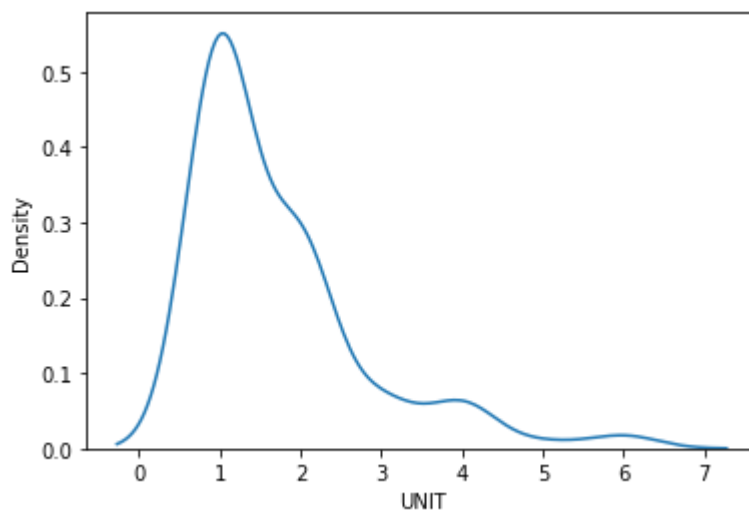
<Figure size 10800x5040 with 0 Axes>

DISTPLOT MEANS DISTRIBUTION PLOT ITS PART OF HISTOGRAM

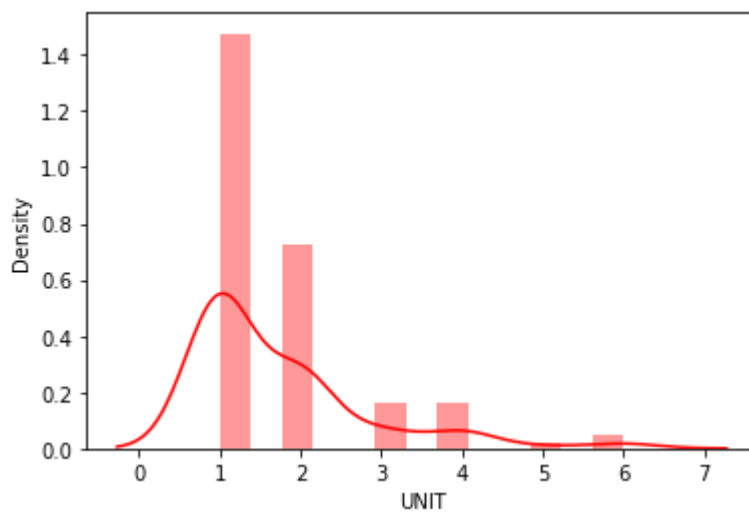
```
In [19]: sb.distplot(df1["UNIT"])
plt.show()
#histogram best for numericL Data
```



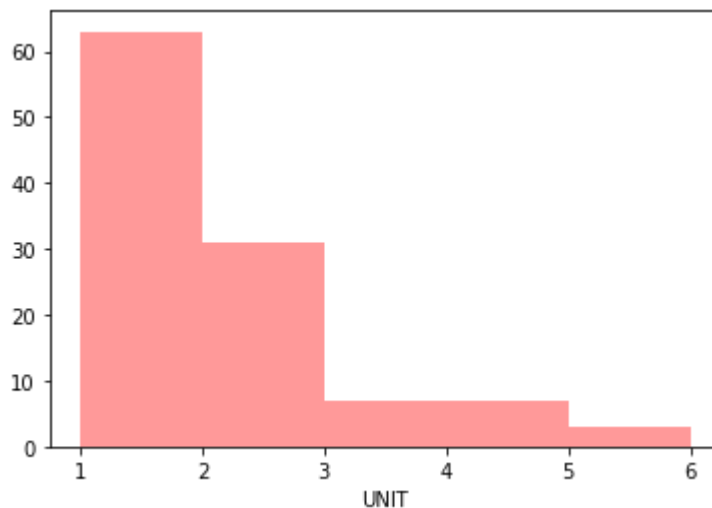
```
In [20]: #only show curve using hist method  
sb.distplot(df1["UNIT"],hist=False)  
plt.show()
```



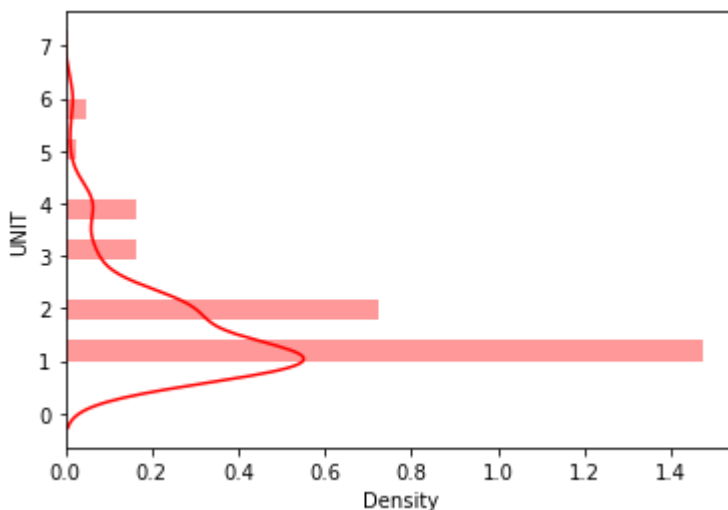
```
In [21]: #using color attributes  
sb.distplot(df1["UNIT"],color="red")  
plt.show()
```



```
In [22]: #using kde for removing curve line
#using bins means its show all big values like following chart
sb.distplot(df1["UNIT"],color="red",bins=5,kde=False)
plt.show()
```

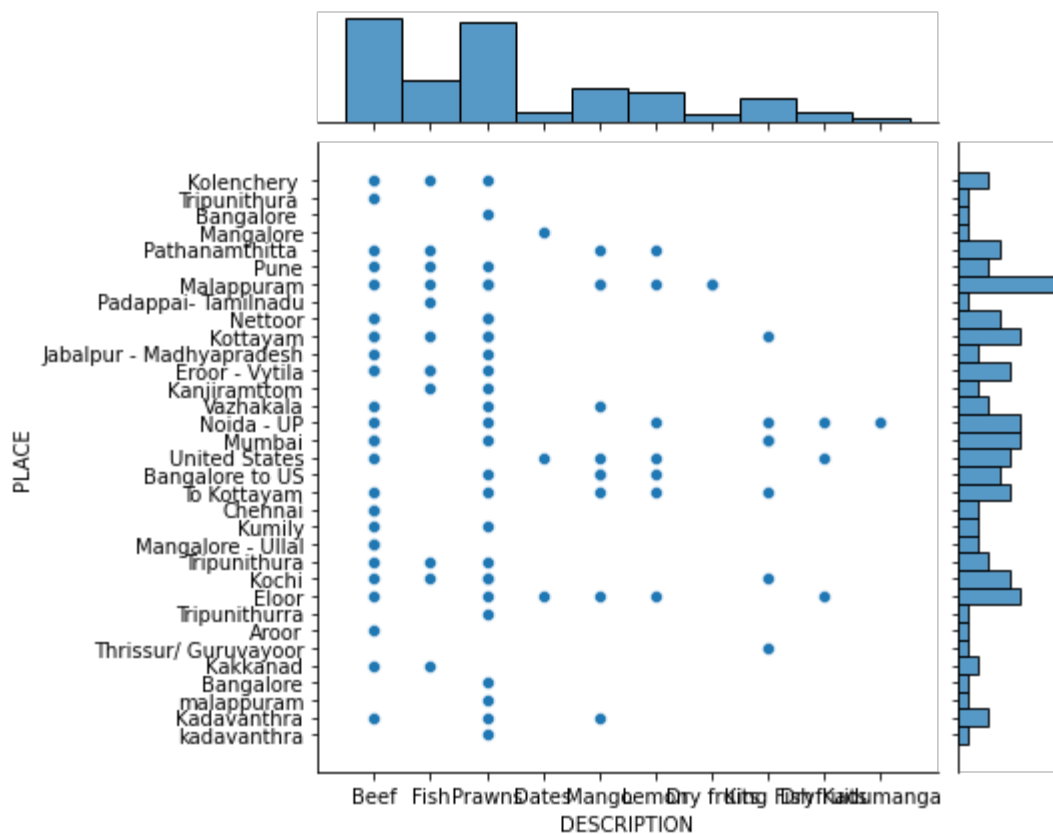


```
In [23]: #Now create a plot vertically
sb.distplot(df1["UNIT"],color="red",vertical=True)
plt.show()
#vertically means replace the x and y
```



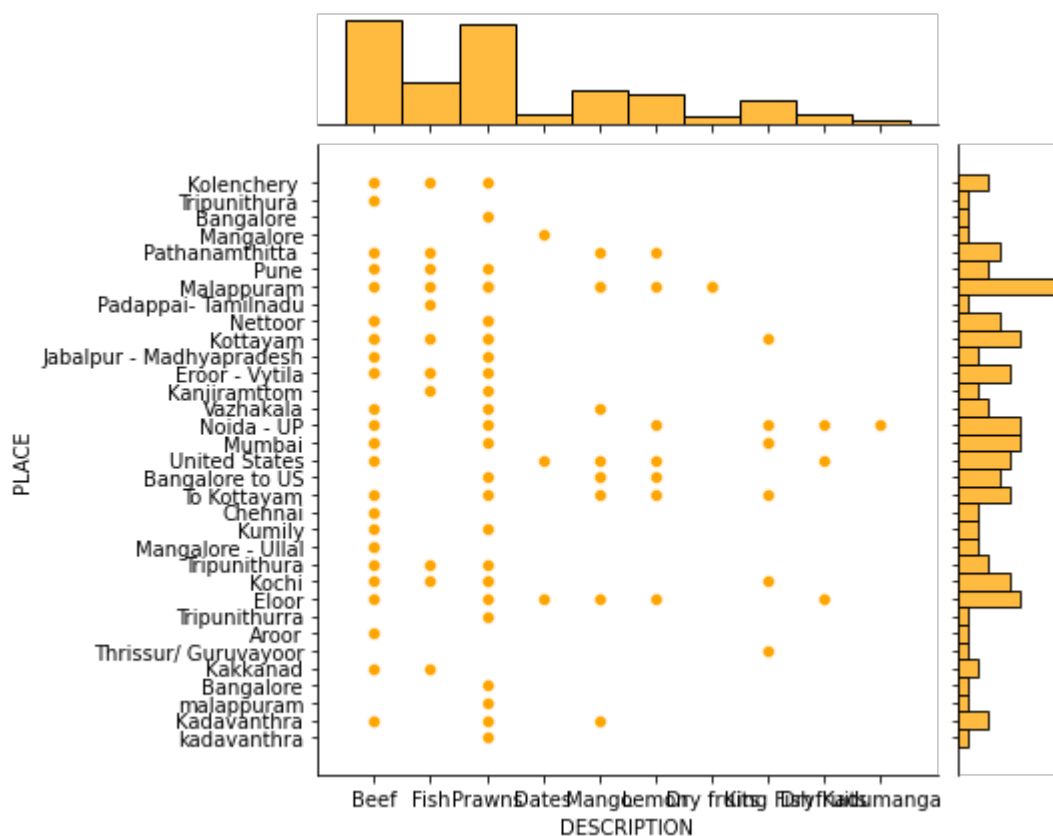
JointPLOT

```
In [24]: sb.jointplot(x="DESCRIPTION",y="PLACE",data=df1)
plt.figure(figsize=(150,70))
plt.show()
#joint plot have both plot histplot and scatterplot attributes
```



<Figure size 10800x5040 with 0 Axes>

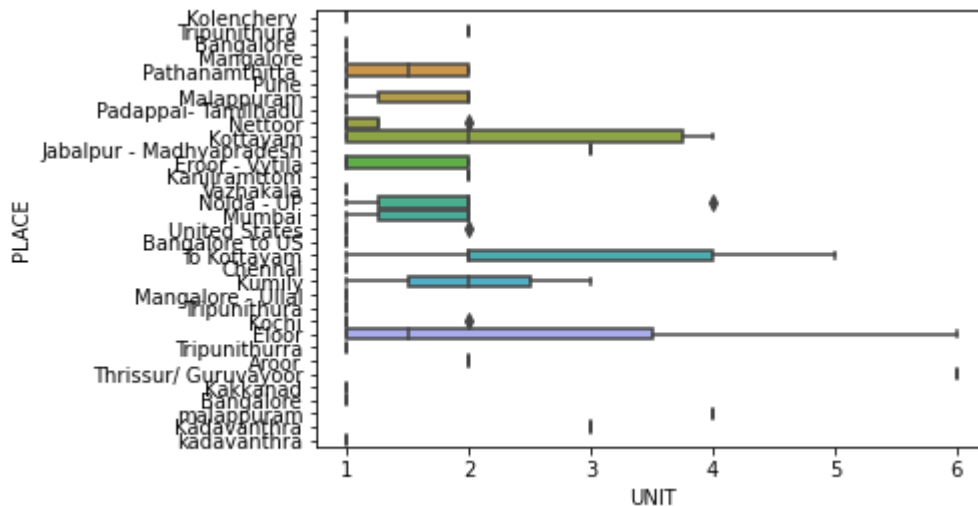
```
In [25]: sb.jointplot(x="DESCRIPTION",y="PLACE",data=df1,color="orange")#for change color
plt.figure(figsize=(150,70))
plt.show()
```



<Figure size 10800x5040 with 0 Axes>

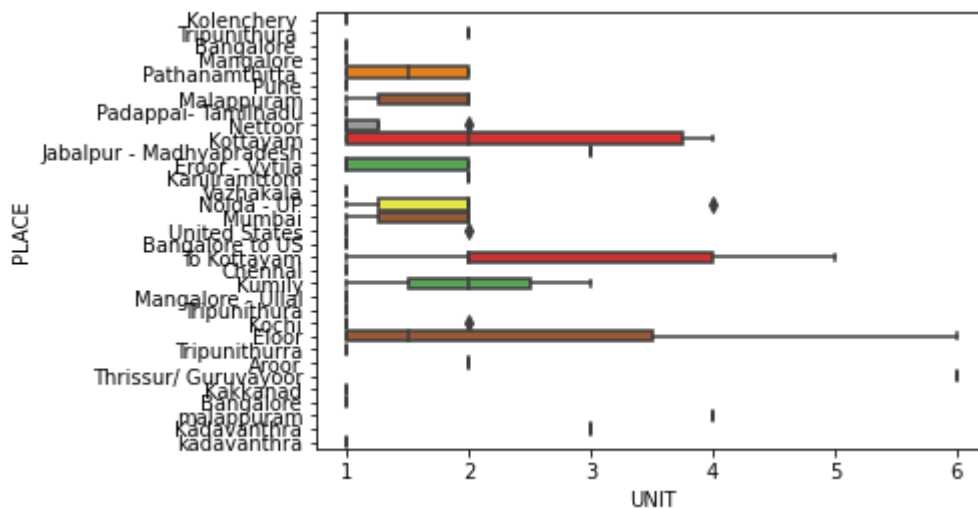
BOXPLOT

```
In [26]: sb.boxplot(x="UNIT",y="PLACE",data=df1)#check the relationship between UNIT AND PLA
plt.figure(figsize=(150,70))
plt.show()
```



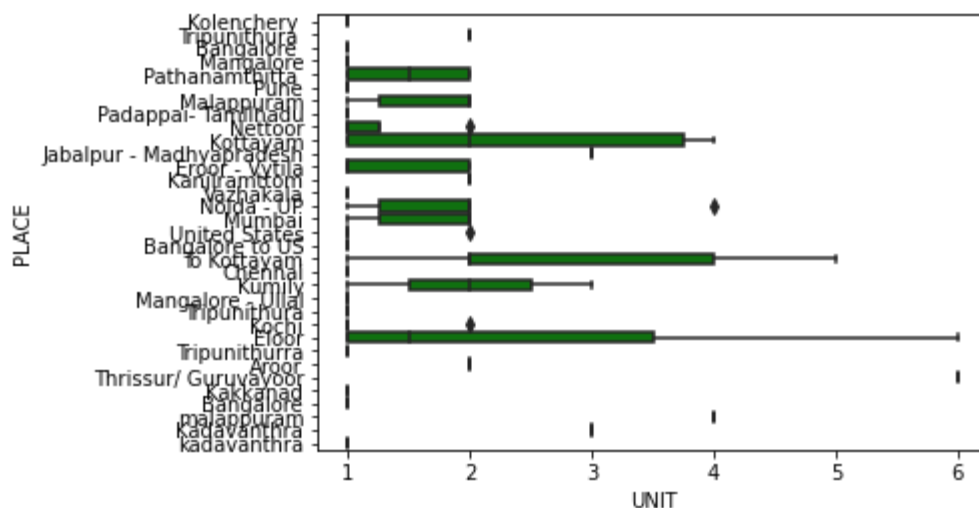
<Figure size 10800x5040 with 0 Axes>

```
In [27]: #boxplot have different palette like set1 etc. palette put the same colour on random
sb.boxplot(x="UNIT",y="PLACE",data=df1,palette="Set1")
plt.figure(figsize=(150,70))
plt.show()
```



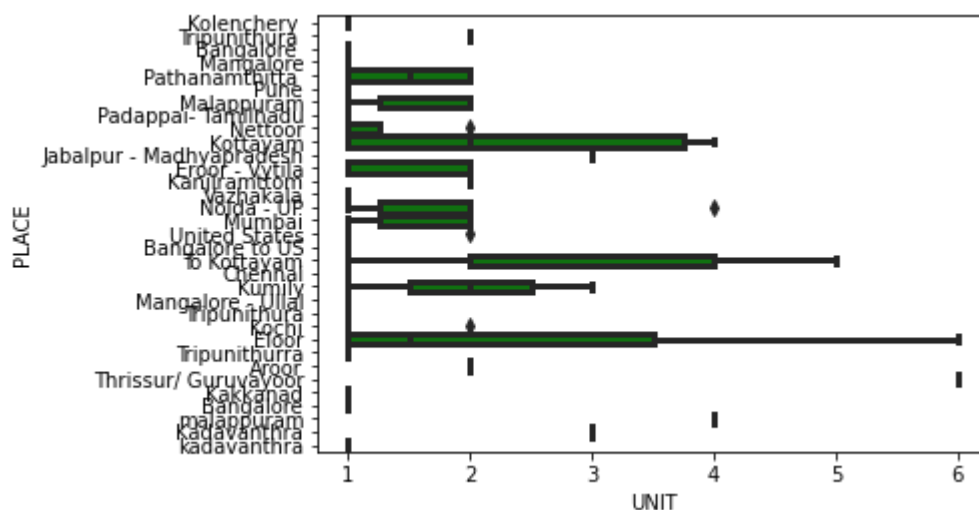
<Figure size 10800x5040 with 0 Axes>

```
In [28]: sb.boxplot(x="UNIT",y="PLACE",data=df1,color="green")#using color attributes from all
plt.figure(figsize=(150,70))
plt.show()
```



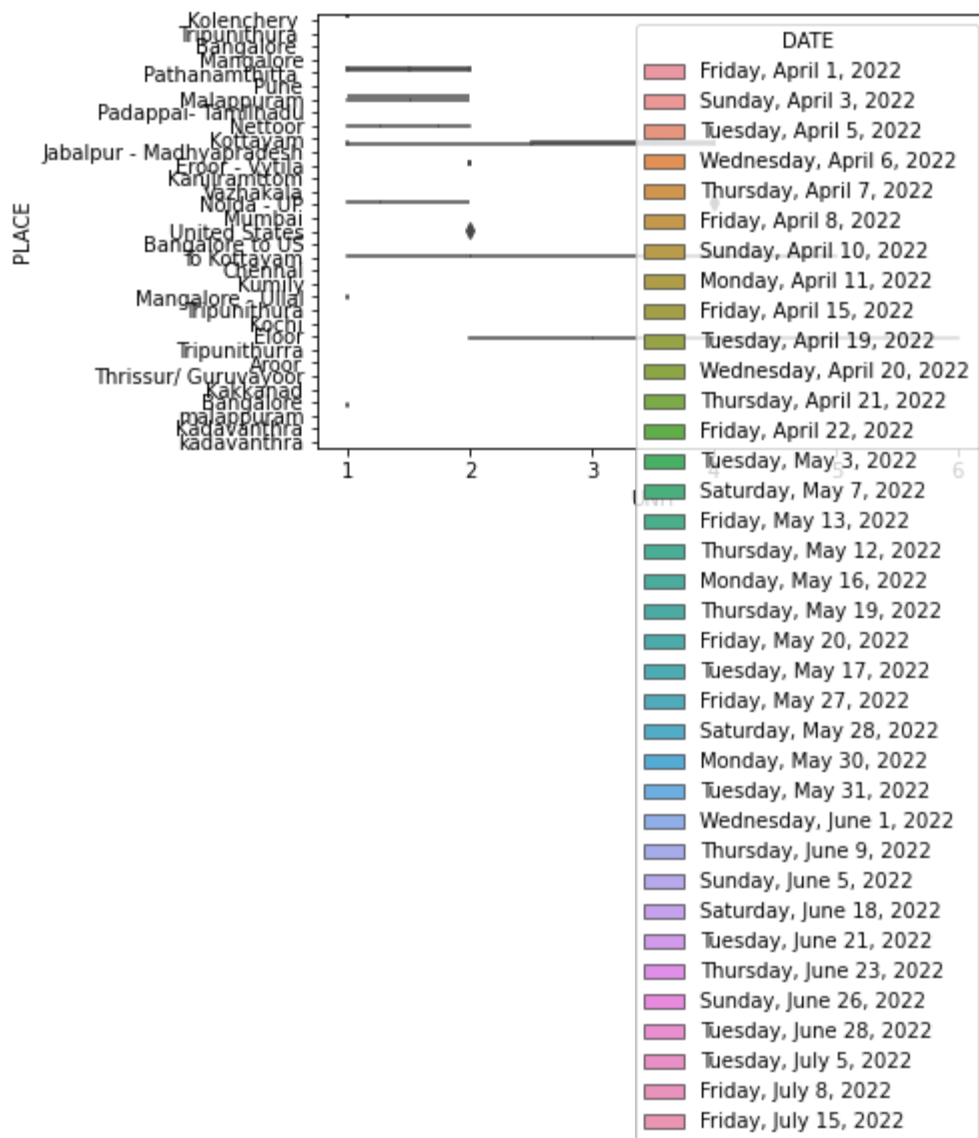
<Figure size 10800x5040 with 0 Axes>

```
In [29]: #change the thickness using linewidth method
sb.boxplot(x="UNIT",y="PLACE",data=df1,color="green",linewidth=3)
plt.figure(figsize=(150,70))
plt.show()
#now its difference between upper and lower image
```



<Figure size 10800x5040 with 0 Axes>

```
In [30]: sb.boxplot(x="UNIT",y="PLACE",data=df1,hue="DATE") #check relation with the date co
plt.figure(figsize=(150,70))
plt.show()
```



<Figure size 10800x5040 with 0 Axes>

pairplot

```
In [31]: sb.pairplot(df1, hue="PLACE",)
plt.show()
```

