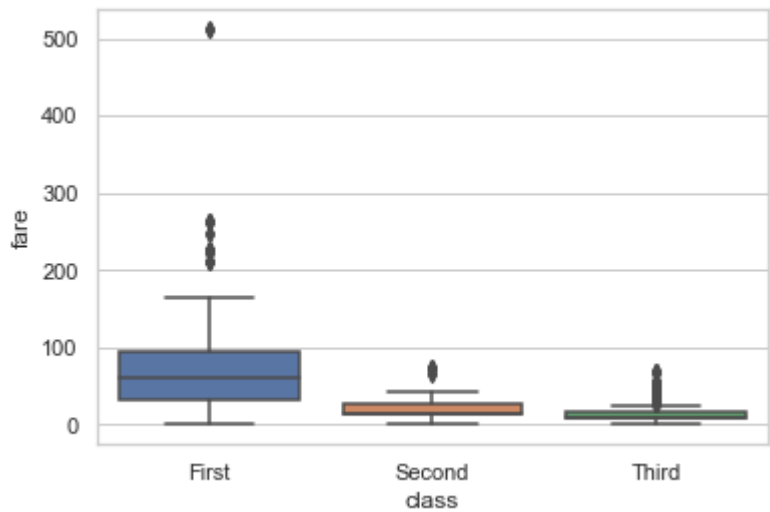


BOX Plots

```
In [9]: #import Libraries
import seaborn
#canvas (Baloon Board)
seaborn.set(style='whitegrid')
#Load dataset
boat = sns.load_dataset("titanic")
boat
#Draw a line plot
seaborn.boxplot(x="class", y="fare", data=boat)
```

Out[9]: <AxesSubplot:xlabel='class', ylabel='fare'>



Tips

```
In [10]: #import Libraries
import seaborn
#canvas (Baloon Board)
seaborn.set(style='whitegrid')
tip = seaborn.load_dataset('tips')
tip
```

Out[10]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
...
239	29.03	5.92	Male	No	Sat	Dinner	3

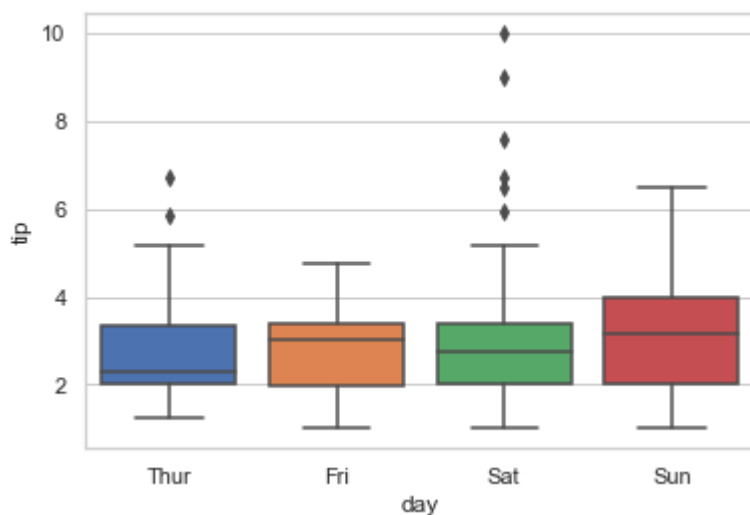
	total_bill	tip	sex	smoker	day	time	size
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

Simple Disk grapgh

```
In [11]: #import libraries
import seaborn
#canvas (Baloon Board)
seaborn.set(style='whitegrid')
tip = seaborn.load_dataset('tips')
tip
seaborn.boxplot(x='day', y='tip', data=tip, saturation=1.1)
```

Out[11]: <AxesSubplot:xlabel='day', ylabel='tip'>



Pandas

```
In [16]: #import libraries
import seaborn as sns
import pandas as pf
import numpy as np
#canvas (Baloon Board)

tip = seaborn.load_dataset('tips')
tip.describe() #descripe all element in data
```

```
Out[16]:
```

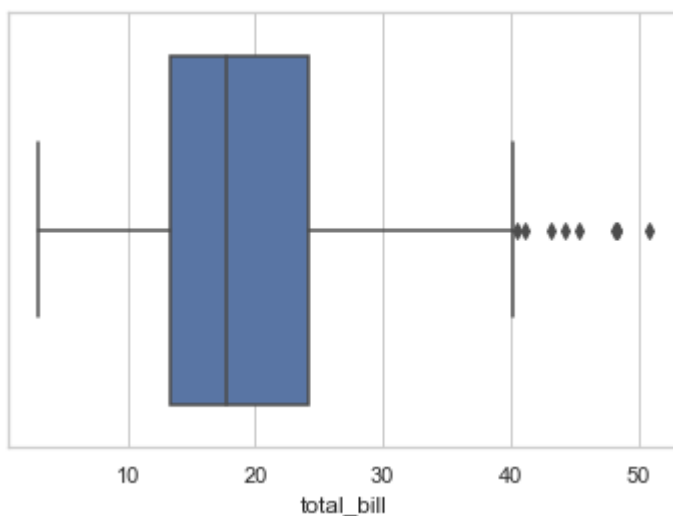
	total_bill	tip	size
count	244.000000	244.000000	244.000000

	total_bill	tip	size
mean	19.785943	2.998279	2.569672
std	8.902412	1.383638	0.951100
min	3.070000	1.000000	1.000000
25%	13.347500	2.000000	2.000000
50%	17.795000	2.900000	2.000000
75%	24.127500	3.562500	3.000000
max	50.810000	10.000000	6.000000

One Things Boxplots

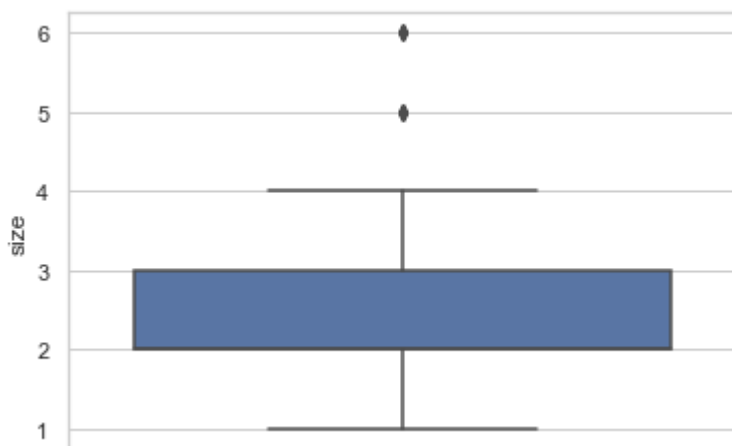
```
In [18]: #import Libraries
import seaborn
#canvas (Baloon Board)
seaborn.set(style='whitegrid')
#Loading DataSets
tip = seaborn.load_dataset('tips')
seaborn.boxplot(x= tip['total_bill'])
```

Out[18]: <AxesSubplot:xlabel='total_bill'>



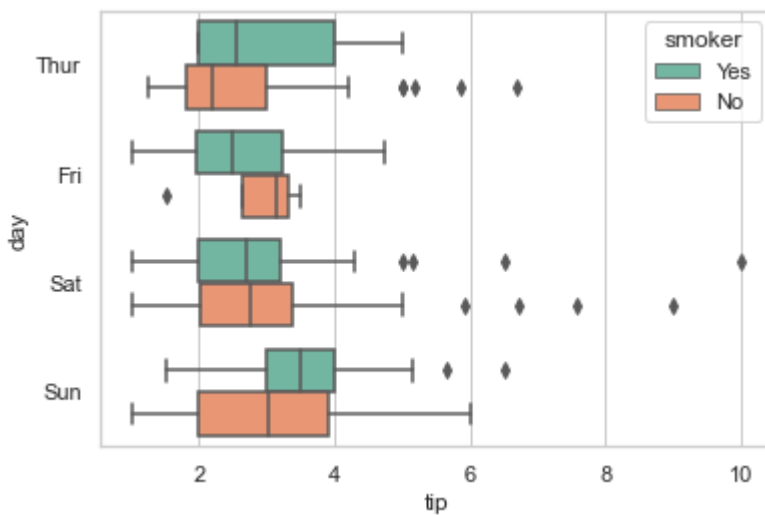
```
In [22]: #import Libraries
import seaborn
#canvas (Baloon Board)
seaborn.set(style='whitegrid')
#Loading DataSets
tip = seaborn.load_dataset('tips')
seaborn.boxplot(y= tip['size'])
```

Out[22]: <AxesSubplot:ylabel='size'>



```
In [35]: #import Libraries
import seaborn
#canvas (Baloon Board)
seaborn.set(style='whitegrid')
#Loading DataSets
tip = seaborn.load_dataset('tips')
seaborn.boxplot(x='tip', y='day', hue='smoker', data=tip, palette="Set2", dodge=True,
```

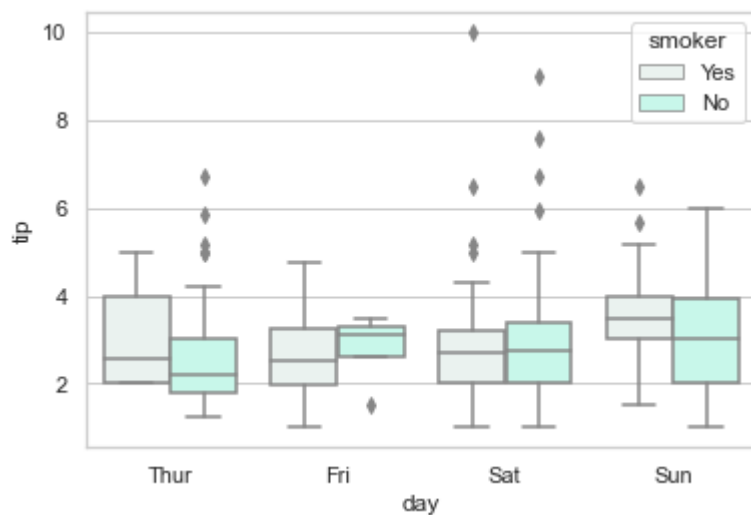
```
Out[35]: <AxesSubplot:xlabel='tip', ylabel='day'>
```



Color Pickers

```
In [50]: #import Libraries
import seaborn
#canvas (Baloon Board)
seaborn.set(style='whitegrid')
#Loading DataSets
tip = seaborn.load_dataset('tips')
seaborn.boxplot(x='day', y='tip', hue="smoker", data=tip, color= "#c0ffee" )
```

```
Out[50]: <AxesSubplot:xlabel='day', ylabel='tip'>
```



Customizing Plots

```
In [51]: import seaborn as sns
import pandas as pd
import numpy as np

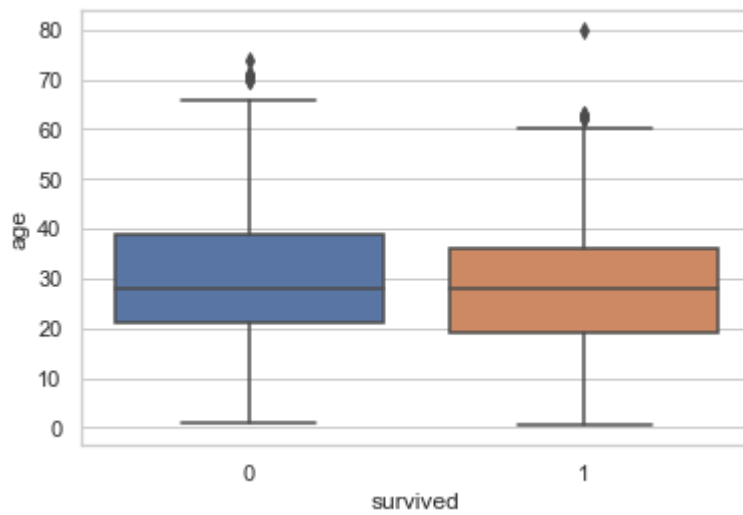
boat = sns.load_dataset("titanic")
boat.head()
```

```
Out[51]:
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	e
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	5
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	5
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	5
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	5
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	5

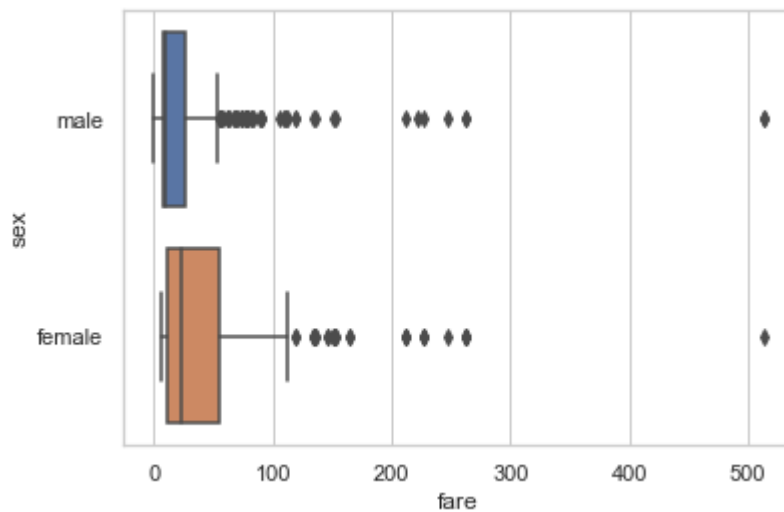
```
In [54]: sns.boxplot(x="survived", y="age", data=boat)
```

```
Out[54]: <AxesSubplot:xlabel='survived', ylabel='age'>
```



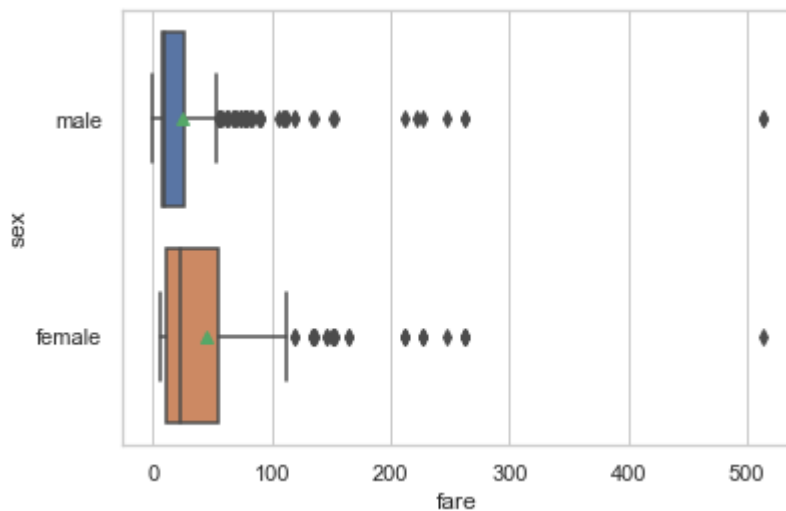
```
In [58]: sns.boxplot(x="fare", y="sex", data=boat)
```

```
Out[58]: <AxesSubplot:xlabel='fare', ylabel='sex'>
```



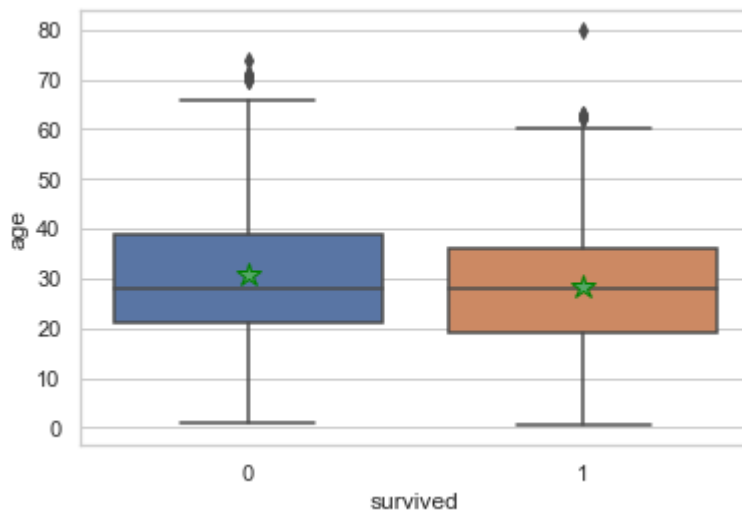
```
In [59]: sns.boxplot(x="fare", y="sex", showmeans=True, data=boat)
```

```
Out[59]: <AxesSubplot:xlabel='fare', ylabel='sex'>
```



```
In [70]: sns.boxplot(x="survived", y="age", showmeans=True, meanprops={"marker": "*", "markersize": 100})
```

```
Out[70]: <AxesSubplot:xlabel='survived', ylabel='age'>
```



FaceGrid and FaceWrap

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. Seaborn helps resolve the two major problems faced by Matplotlib; the problems are ?

Default Matplotlib parameters Working with data frames As Seaborn compliments and extends Matplotlib, the learning curve is quite gradual. If you know Matplotlib, you are already half way through Seaborn.

`seaborn.FacetGrid()` : `FacetGrid` class helps in visualizing distribution of one variable as well as the relationship between multiple variables separately within subsets of your dataset using multiple panels. A `FacetGrid` can be drawn with up to three dimensions ? row, col, and hue. The first two have obvious correspondence with the resulting array of axes; think of the hue variable as a third dimension along a depth axis, where different levels are plotted with different colors. `FacetGrid`

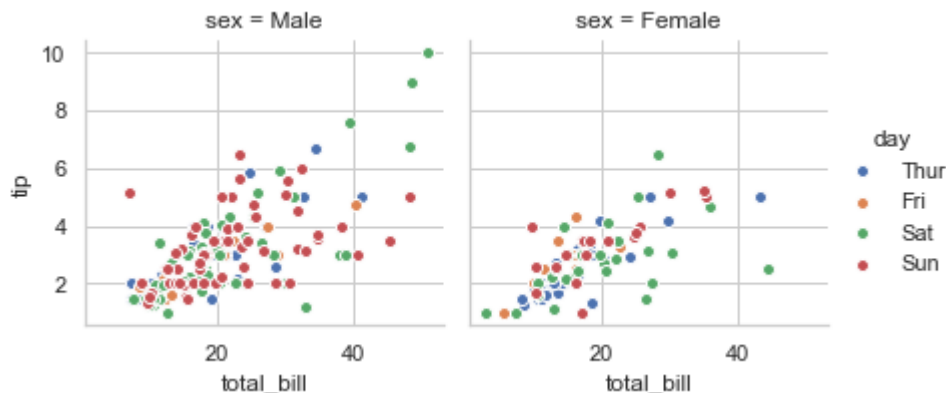
object takes a dataframe as input and the names of the variables that will form the row, column, or hue dimensions of the grid. The variables should be categorical and the data at each level of the variable will be used for a facet along that axis.

```
In [77]: # importing packages
import seaborn
import matplotlib.pyplot as plt

# Loading of a dataframe from seaborn
df = seaborn.load_dataset('tips')

##### Main Section #####
# Form a facetgrid using columns with a hue
graph = seaborn.FacetGrid(df, col = "sex", hue = "day")
# map the above form facetgrid with some attributes
graph.map(plt.scatter, "total_bill", "tip", edgecolor = "w").add_legend()
# show the object
plt.show()

# This code is contributed by Deepanshu Rustagi.
```



```
In [79]: # importing packages
import seaborn
import matplotlib.pyplot as plt

# Loading of a dataframe from seaborn
df = seaborn.load_dataset('tips')

##### Main Section #####
# Form a facetgrid using columns with a hue
graph = seaborn.FacetGrid(df, col = "smoker", hue = "time")
# map the above form facetgrid with some attributes
graph.map(plt.hist, 'total_bill', bins = 15, color = 'orange')
# show the object
plt.show()

# This code is contributed by Deepanshu Rustagi.
```