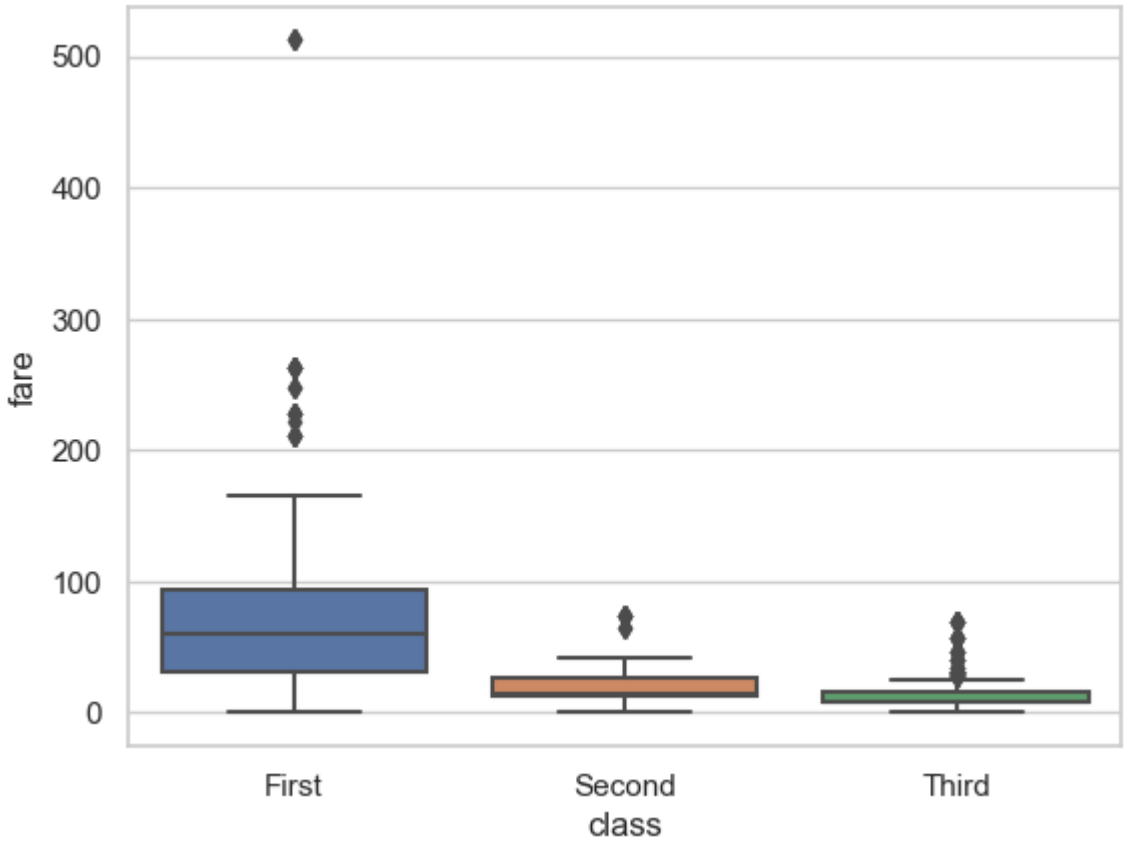
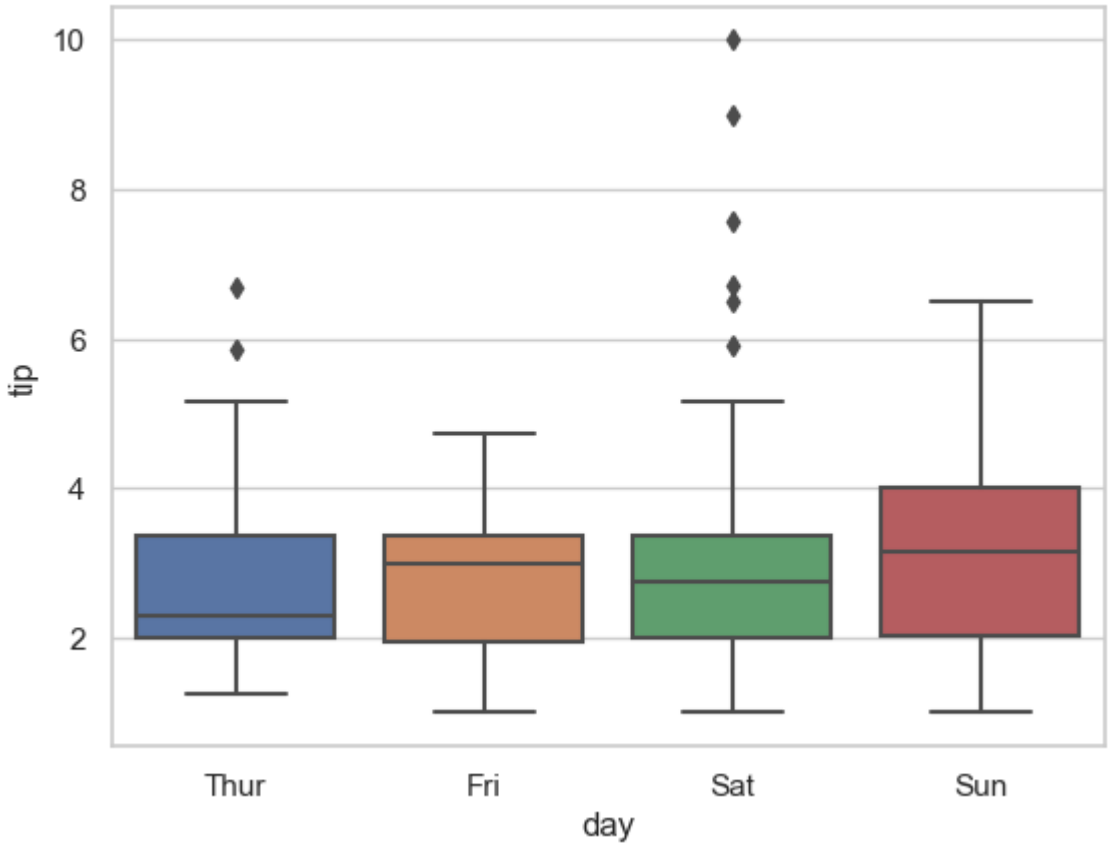
In [15]:
```
# import library
import seaborn as sns
# canvas(balloon board)
sns.set(style="whitegrid")

kashti=sns.load_dataset("titanic")
seaborn.boxplot(x="class", y="fare", data=kashti)
```

Out[15]:
```
<Axes: xlabel='class', ylabel='fare'>
```



In [23]:
```
import seaborn as sns

tip=sns.load_dataset("tips")
tip
sns.boxplot(x="day",y="tip", data=tip, )
```

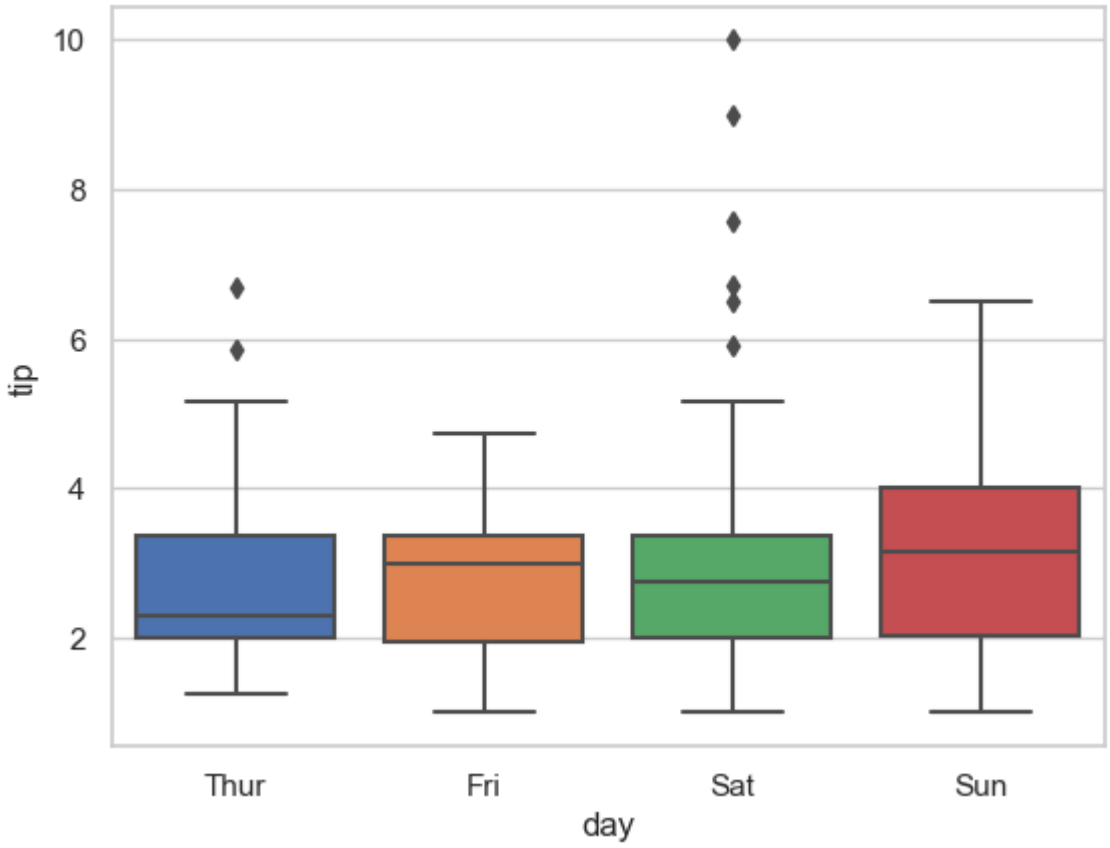
Out[23]:
```
<Axes: xlabel='day', ylabel='tip'>
```



saturation of boxplot

In [30]:
```
import seaborn as sns


tip=sns.load_dataset("tips")
tip
sns.boxplot(x="day",y="tip", data=tip,saturation=4 )
```

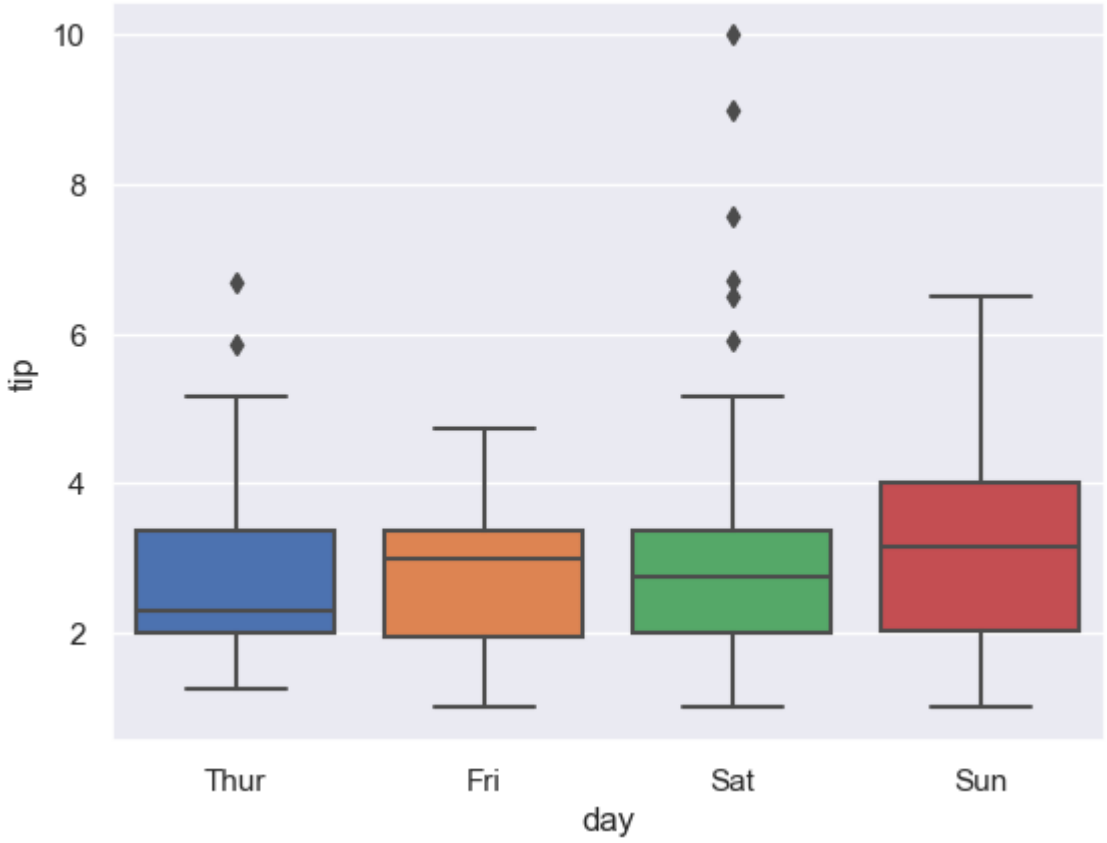
Out[30]:
```
<Axes: xlabel='day', ylabel='tip'>
```



add grid

In [33]:
```
import seaborn as sns

tip=sns.load_dataset("tips")
tip
# sns.set_style("dark") or sns.set(style="dark")
sns.set(style="darkgrid")
sns.boxplot(x="day",y="tip", data=tip, saturation=4 )
```

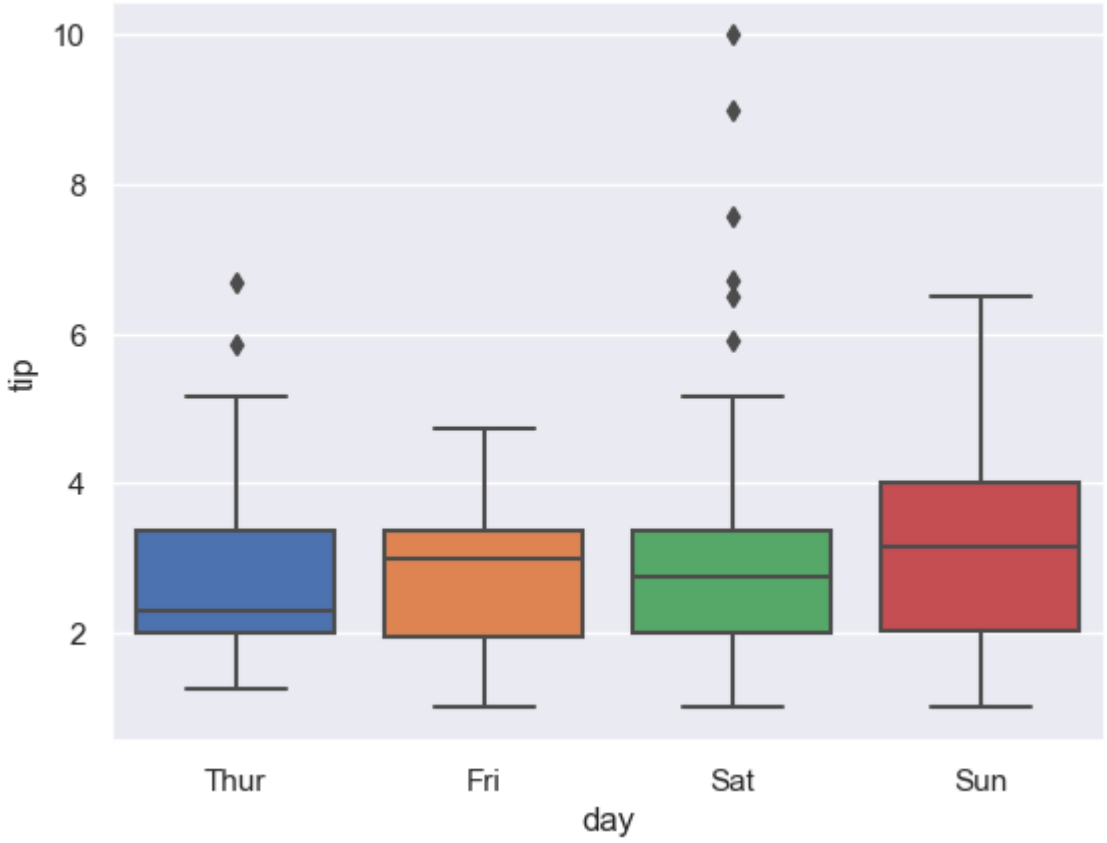
Out[33]:
```
<Axes: xlabel='day', ylabel='tip'>
```



add estimator but estimator is not work for boxplot

In [39]:
```
import seaborn as sns
# import numpy or from numpy import median/mean
from numpy import mean

tip=sns.load_dataset("tips")
tip
# sns.set_style("dark") or sns.set(style="dark")
sns.set(style="darkgrid")
sns.boxplot(x="day",y="tip", data=tip, saturation=4, )
```

Out[39]:
```
<Axes: xlabel='day', ylabel='tip'>
```



In [46]:
```
import seaborn as sns
import pandas as pf
import numpy as np

tip=sns.load_dataset("tips")
tip
```

Out[46]:

	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
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

In [47]:
```
tip.describe
```

Out[47]:
```
<bound method NDFrame.describe of      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
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 x 7 columns]>
```

In [49]:
```
tip.describe()
```

Out[49]:

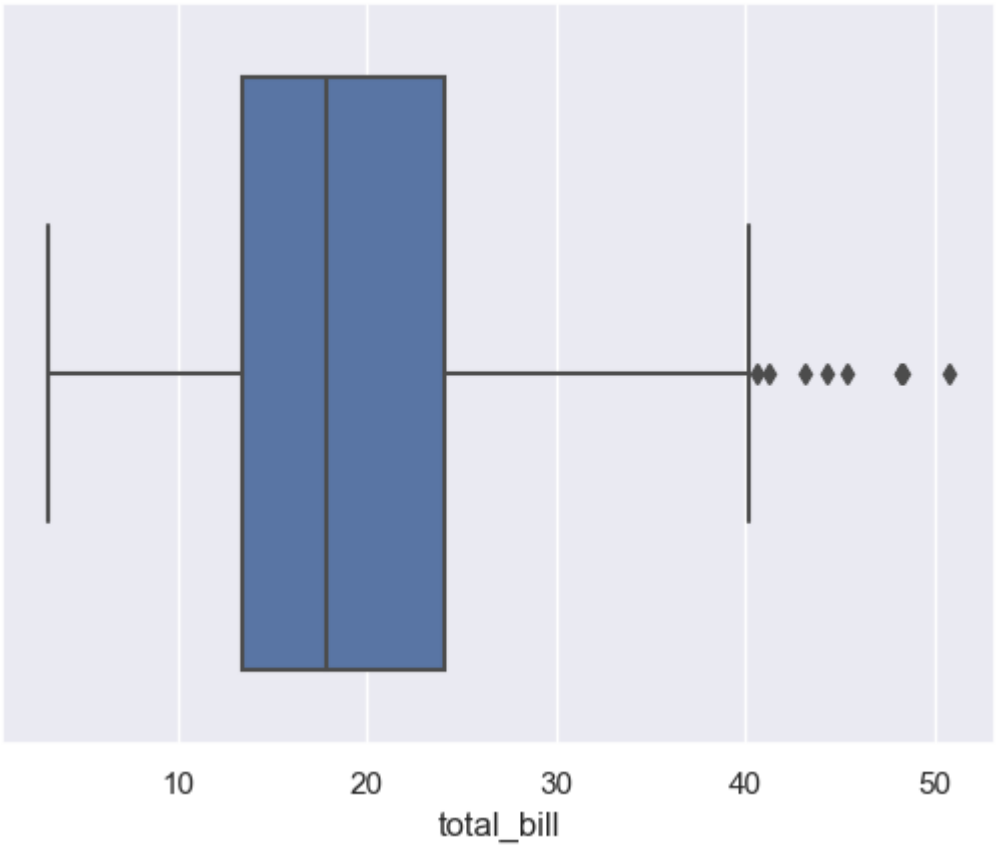
	total_bill	tip	size
count	244.000000	244.000000	244.000000
mean	19.785943	2.998279	2.569672
std	8.902412	1.383638	0.951100
min	3.070000	1.000000	1.000000
25%	13.347500	2.000000	2.000000
50%	17.795000	2.900000	2.000000
75%	24.127500	3.562500	3.000000
max	50.810000	10.000000	6.000000

- catagorical variables must be in x axis or in hue numerical variables must be in y axis numrical variables not in hue

importing the required module

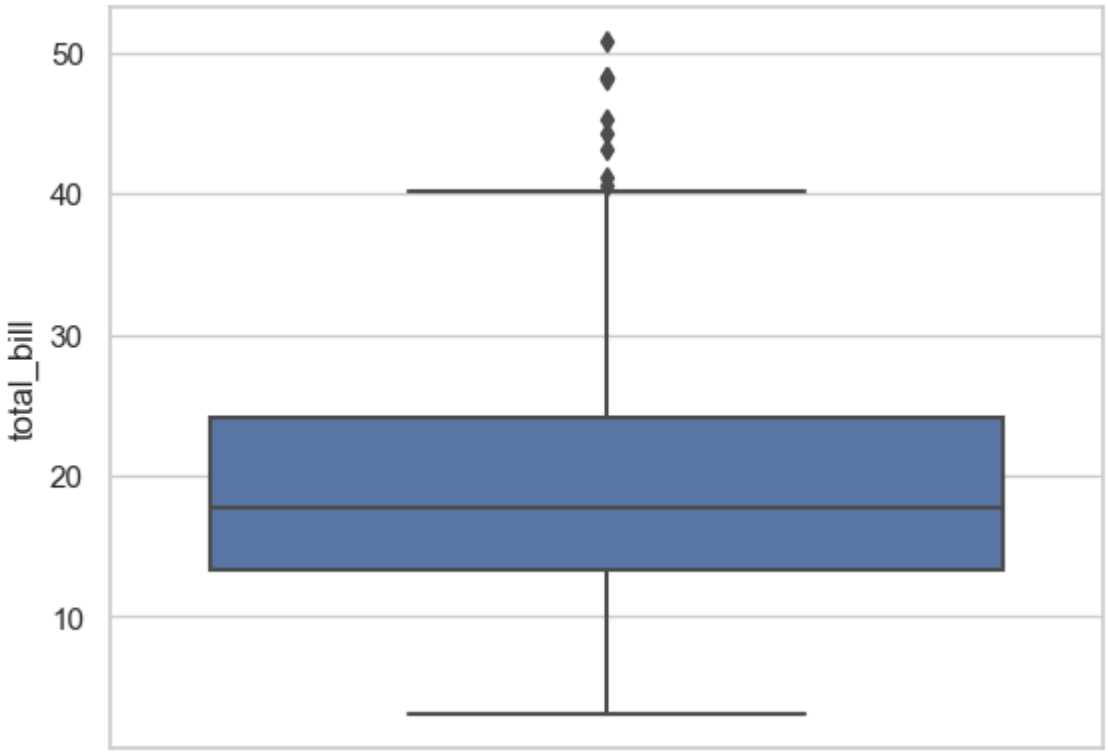
In [58]:
```
# importing the required module
import seaborn as sns

# use to set_style of background of plot
sns.set(style="darkgrid")

# loading dataset
tip=sns.load_dataset("tips")
tip

sns.boxplot(x=tip["total_bill"])
```

Out[58]:
```
<Axes: xlabel='total_bill'>
```



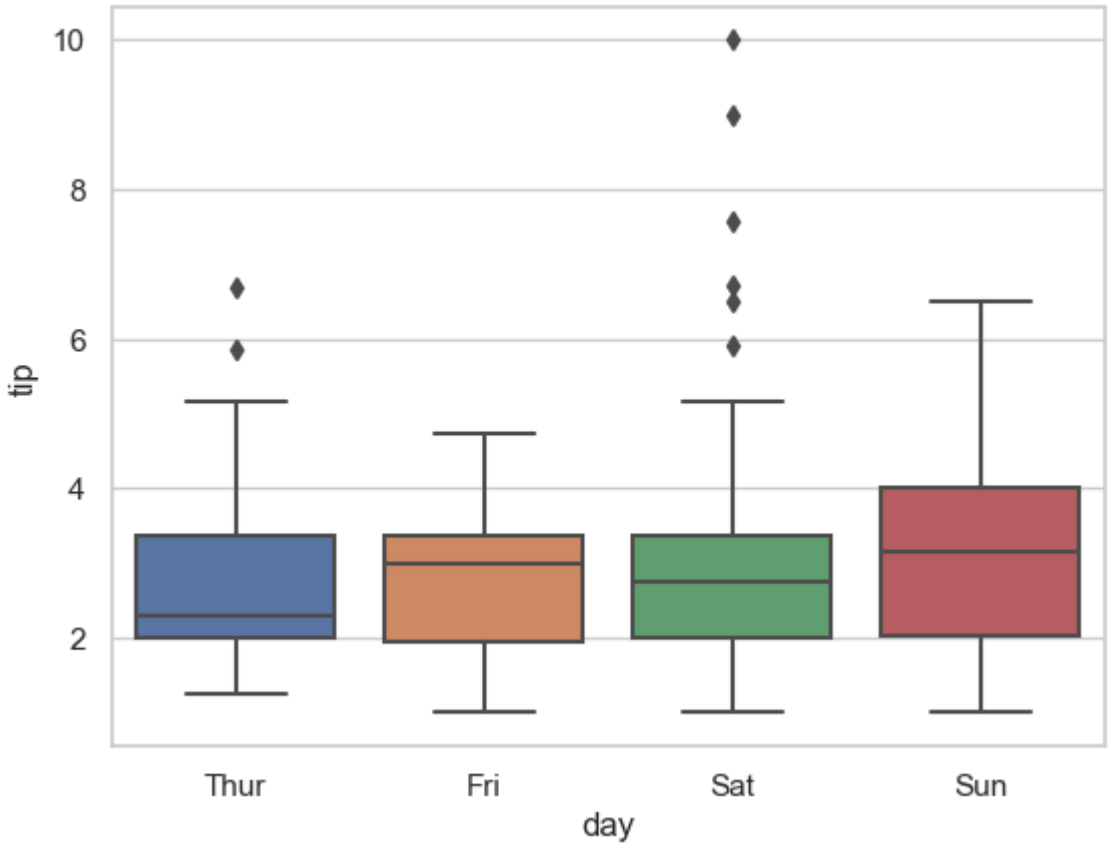
In [57]:
```
# importing the required module
import seaborn as sns

# use to set_style of background of plot
sns.set(style="whitegrid")

# loading dataset
tip=sns.load_dataset("tips")
tip

sns.boxplot(y=tip["total_bill"])
```

Out[57]:
```
<Axes: ylabel='total_bill'>
```



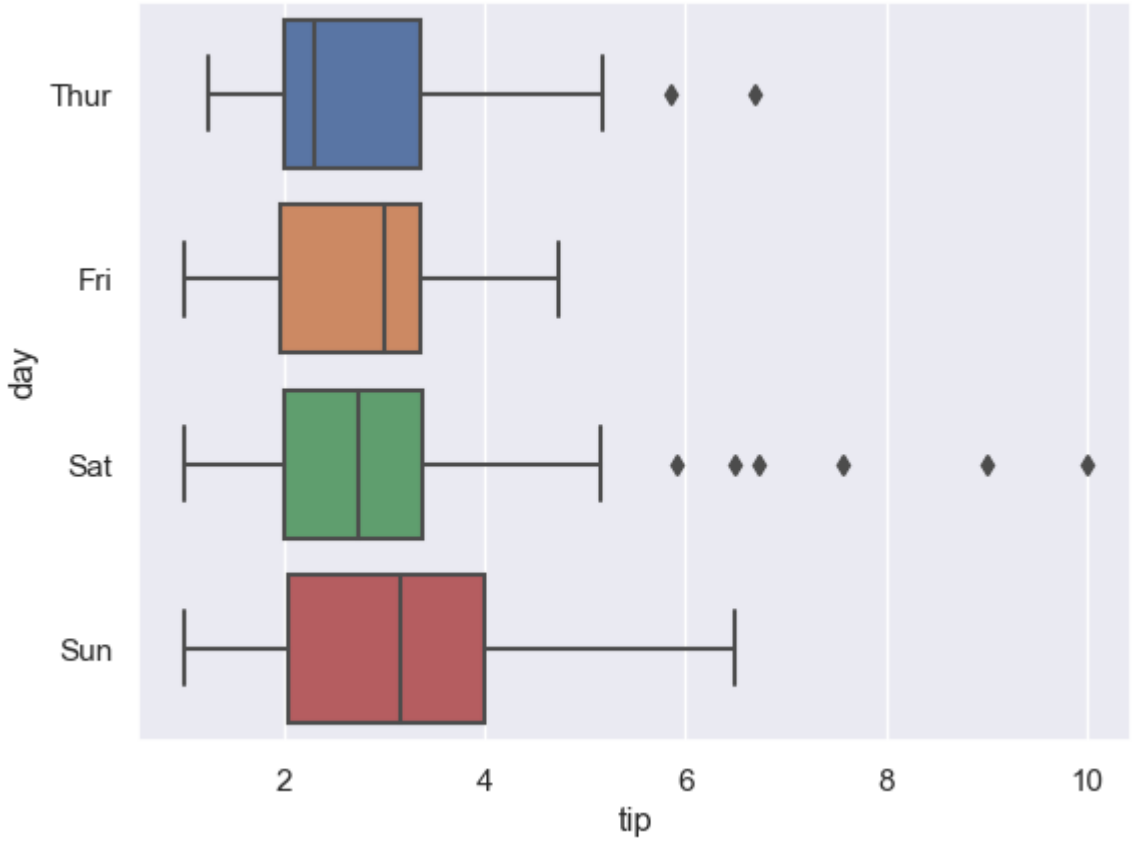
In [61]:
```
# importing the required module
import seaborn as sns

# use to set_style of background of plot
sns.set(style="whitegrid")

# loading dataset
tip=sns.load_dataset("tips")
tip

sns.boxplot(x="day", y="tip", data=tip)
```

Out[61]:
```
<Axes: xlabel='day', ylabel='tip'>
```



In [62]:
```
# importing the required module
import seaborn as sns

# use to set_style of background of plot
sns.set(style="darkgrid")

# loading dataset
tip=sns.load_dataset("tips")
tip

sns.boxplot(y="day", x="tip", data=tip)
```

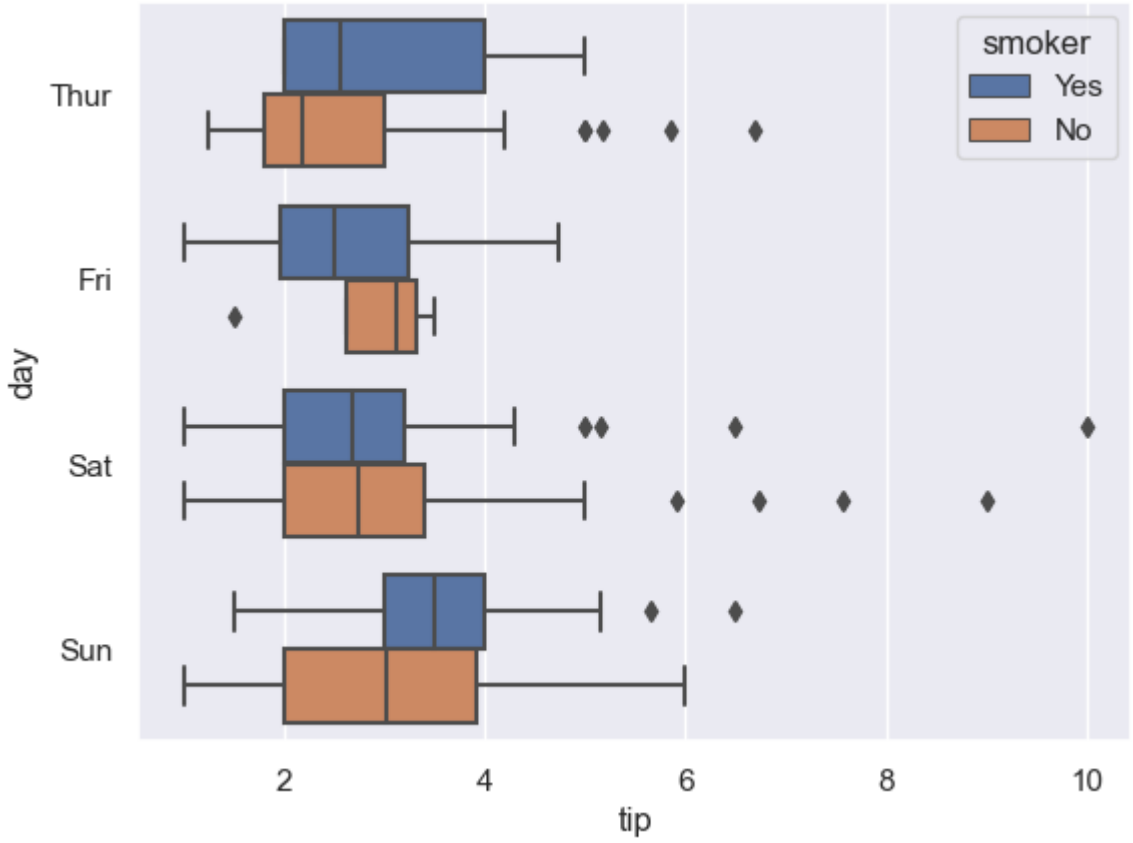
Out[62]:
```
<Axes: xlabel='tip', ylabel='day'>
```



add hue, palette, dodge, capital Set

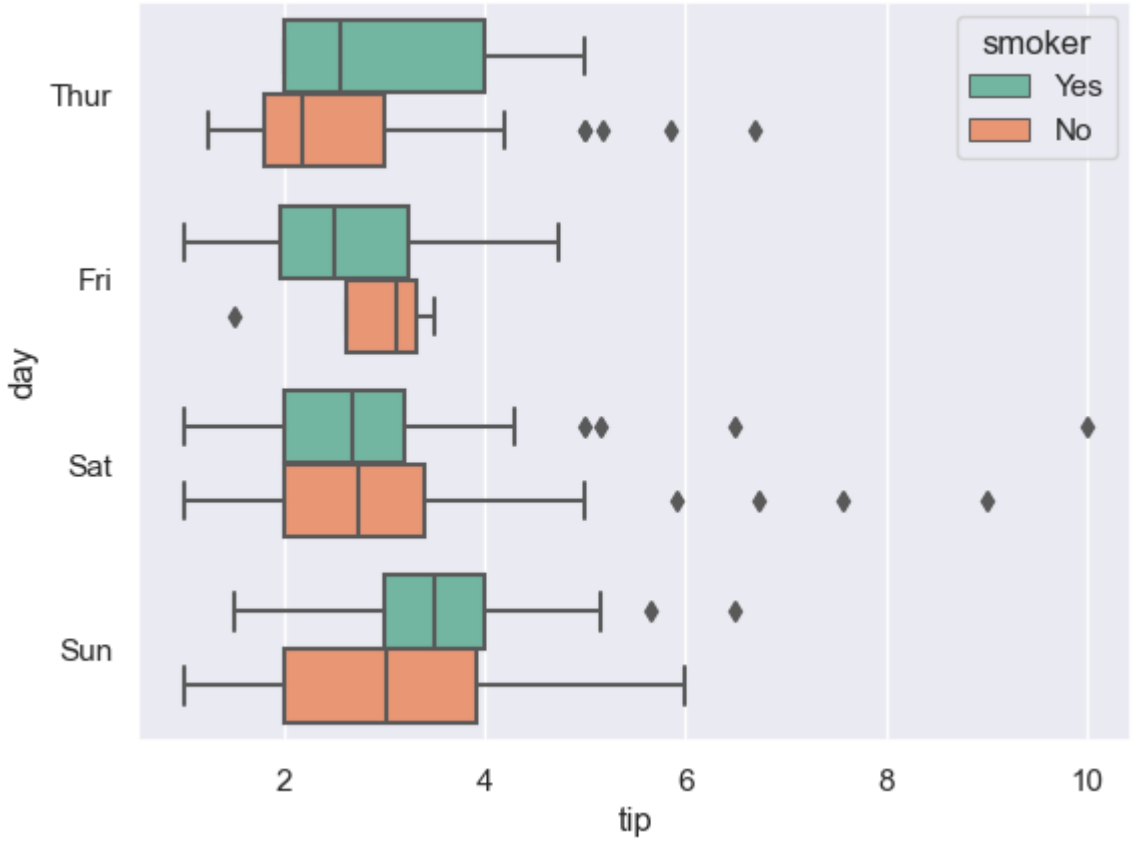
In [70]:
```
# importing the required module
import seaborn as sns

# use to set_style of background of plot
sns.set(style="darkgrid")

# loading dataset
tip=sns.load_dataset("tips")
tip

sns.boxplot(y="day", x="tip", hue="smoker", data=tip,
            )
```

Out[70]:
```
<Axes: xlabel='tip', ylabel='day'>
```



In [74]:
```
# importing the required module
import seaborn as sns

# use to set_style of background of plot
sns.set(style="darkgrid")

# loading dataset
tip=sns.load_dataset("tips")
tip

sns.boxplot(y="day", x="tip", hue="smoker", data=tip,
         palette="Set2"  )
```

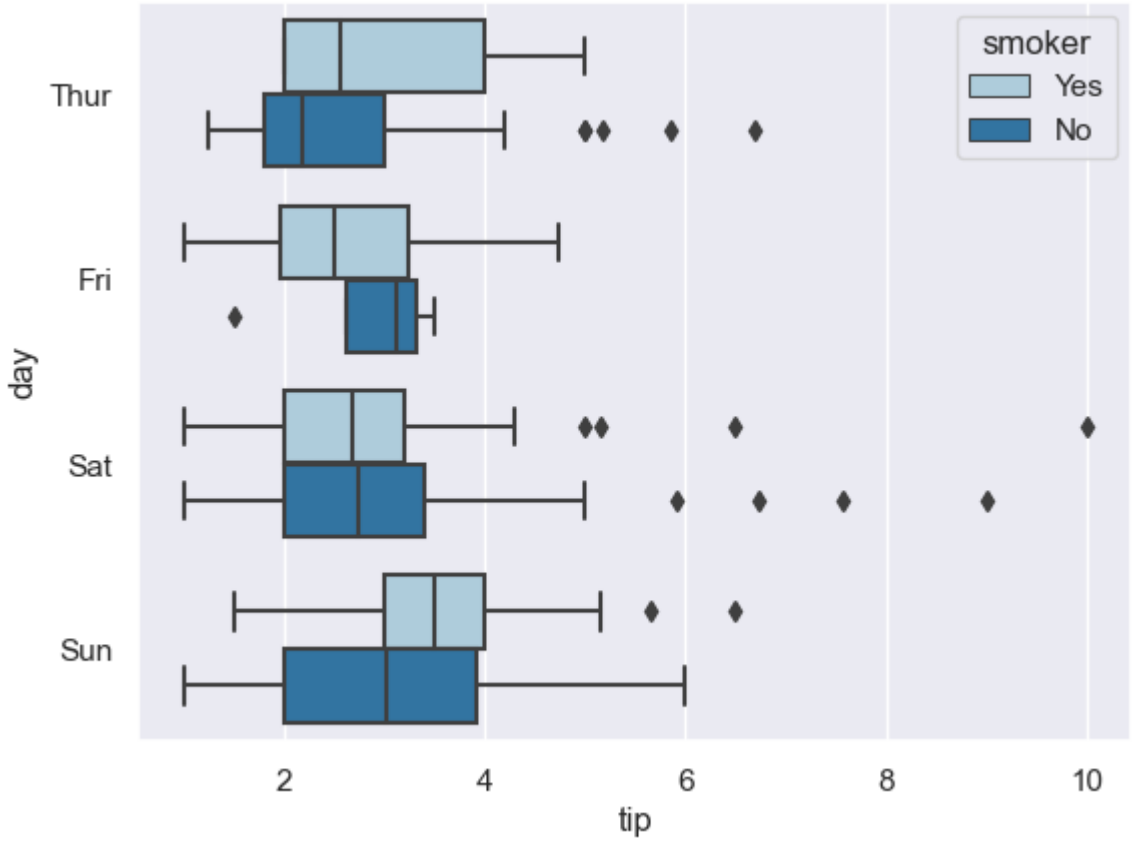
Out[74]:
```
<Axes: xlabel='tip', ylabel='day'>
```



In [ ]:
```
### Palette are the comlours but fisrt is capital
```

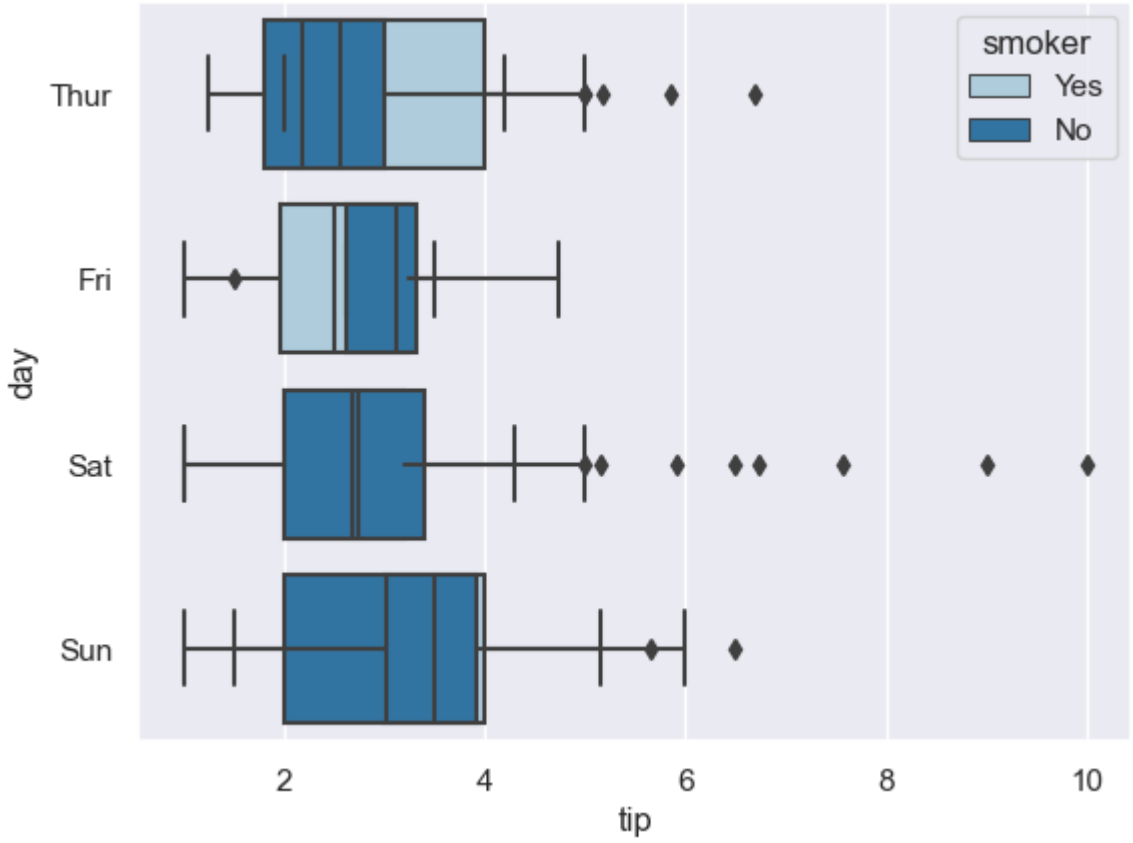
In [75]:
```
# importing the required module
import seaborn as sns

# use to set_style of background of plot
sns.set(style="darkgrid")

# loading dataset
tip=sns.load_dataset("tips")
tip

sns.boxplot(y="day", x="tip", hue="smoker", data=tip,
         palette="Paired"  )
```

Out[75]:
```
<Axes: xlabel='tip', ylabel='day'>
```



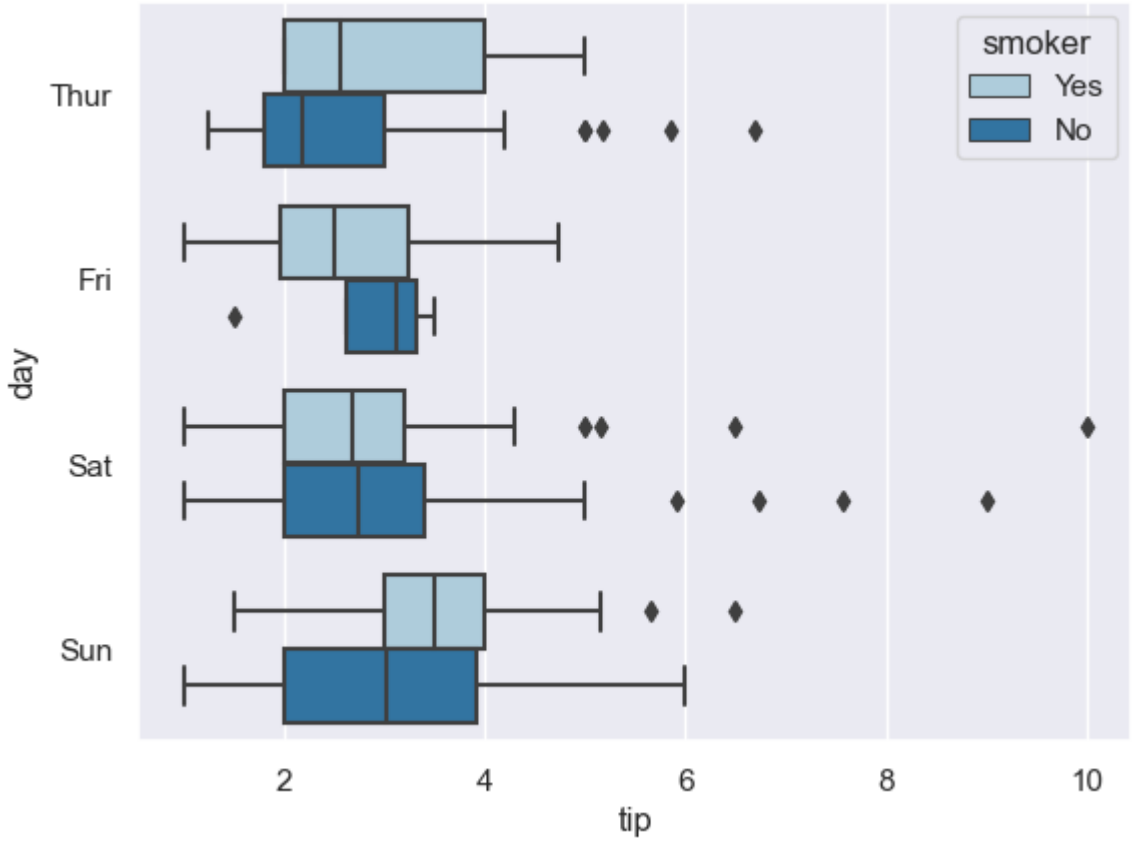
In [77]:
```
# importing the required module
import seaborn as sns

# use to set_style of background of plot
sns.set(style="darkgrid")

# loading dataset
tip=sns.load_dataset("tips")
tip

sns.boxplot(y="day", x="tip", hue="smoker", data=tip,
         palette="Paired", dodge=False )
```

Out[77]:
```
<Axes: xlabel='tip', ylabel='day'>
```



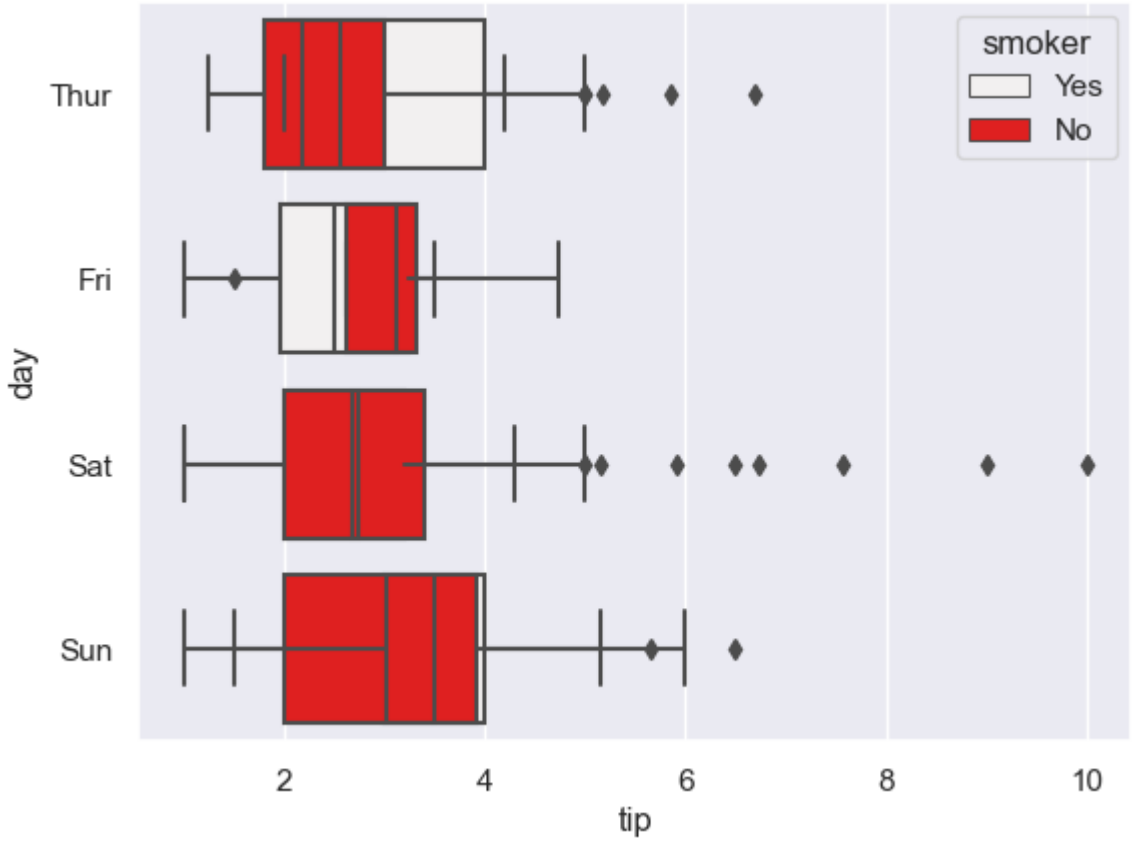
In [78]:
```
# importing the required module
import seaborn as sns

# use to set_style of background of plot
sns.set(style="darkgrid")

# loading dataset
tip=sns.load_dataset("tips")
tip

sns.boxplot(y="day", x="tip", hue="smoker", data=tip,
         palette="Paired", dodge=True )
```

Out[78]:
```
<Axes: xlabel='tip', ylabel='day'>
```



In [90]:
```
# Dodge color

# importing the required module
import seaborn as sns

# use to set_style of background of plot
sns.set(style="darkgrid")

# loading dataset
tip=sns.load_dataset("tips")
tip

sns.boxplot(y="day", x="tip", hue="smoker", data=tip,
        dodge=False, color="red")
```

Out[90]:
```
<Axes: xlabel='tip', ylabel='day'>
```



In [117…

```
# Remove Hue
# Dodge color (use color codes by )

# importing the required module
import seaborn as sns

# use to set_style of background of plot
sns.set(style="darkgrid")

# loading dataset
tip=sns.load_dataset("tips")
tip

sns.boxplot(y="day", x="tip",  data=tip,
          color="#690690")
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\seaborn\palettes.py:251, in color_palette(palette,
n_colors, desat, as_cmap)
    250     palette = map(mpl.colors.colorConverter.to_rgb, palette)
--> 251     palette = _ColorPalette(palette)
    252 except ValueError:

File ~\anaconda3\Lib\site-packages\matplotlib\colors.py:496, in to_rgb(c)
    495 """Convert *c* to an RGB color, silently dropping the alpha channel."""
--> 496 return to_rgba(c)[:3]

File ~\anaconda3\Lib\site-packages\matplotlib\colors.py:299, in to_rgba(c, alpha)
    298 if rgba is None:  # Suppress exception chaining of cache lookup failure.
--> 299     rgba = _to_rgba_no_colorcycle(c, alpha)
    300     try:

File ~\anaconda3\Lib\site-packages\matplotlib\colors.py:374, in _to_rgba_no_colorcycl
e(c, alpha)
    373         return c, c, c, alpha if alpha is not None else 1.
--> 374     raise ValueError(f"Invalid RGBA argument: {orig_c!r}")
    375 # turn 2-D array into 1-D array

ValueError: Invalid RGBA argument: 'No'

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
Cell In[117], line 14
     11 tip=sns.load_dataset("tips")
     12 tip
---> 14 sns.boxplot(y="day", x="tip",  data=tip,
     15           palette={"Yes", ".9", "No",".5"})

File ~\anaconda3\Lib\site-packages\seaborn\categorical.py:2231, in boxplot(data, x,
y, hue, order, hue_order, orient, color, palette, saturation, width, dodge, fliersiz
e, linewidth, whis, ax, **kwargs)
   2224 def boxplot(
   2225     data=None, *, x=None, y=None, hue=None, order=None, hue_order=None,
   2226     orient=None, color=None, palette=None, saturation=.75, width=.8,
   2227     dodge=True, fliersize=5, linewidth=None, whis=1.5, ax=None,
   2228     **kwargs
   2229 ):
-> 2231     plotter = _BoxPlotter(x, y, hue, data, order, hue_order,
   2232                           orient, color, palette, saturation,
   2233                           width, dodge, fliersize, linewidth)
   2235     if ax is None:
   2236         ax = plt.gca()

File ~\anaconda3\Lib\site-packages\seaborn\categorical.py:786, in _BoxPlotter.__init_
_(self, x, y, hue, data, order, hue_order, orient, color, palette, saturation, width,
dodge, fliersize, linewidth)
    781 def __init__(self, x, y, hue, data, order, hue_order,
    782              orient, color, palette, saturation,
    783              width, dodge, fliersize, linewidth):
    785     self.establish_variables(x, y, hue, data, orient, order, hue_order)
--> 786     self.establish_colors(color, palette, saturation)
    788     self.dodge = dodge
    789     self.width = width
```

```
File ~\anaconda3\Lib\site-packages\seaborn\categorical.py:696, in _CategoricalPlotte
r.establish_colors(self, color, palette, saturation)
    693             levels = self.hue_names
    694         palette = [palette[l] for l in levels]
--> 696     colors = color_palette(palette, n_colors)
    698 # Desaturate a bit because these are patches
    699 if saturation < 1:

File ~\anaconda3\Lib\site-packages\seaborn\palettes.py:253, in color_palette(palette,
n_colors, desat, as_cmap)
    251         palette = _ColorPalette(palette)
    252     except ValueError:
--> 253         raise ValueError(f"Could not generate a palette for {palette}")
    255 return palette

ValueError: Could not generate a palette for <map object at 0x000001A4B736A560>
```