# Categorical_Plots_2

June 8, 2024

## 1 Categorical Data Plots

Now let's discuss using seaborn to plot categorical data! There are a few main plot types for this:

- factorplot
- boxplot
- violinplot
- stripplot
- swarmplot
- barplot
- countplot

Let's go through examples of each!

[24]:
```
import seaborn as sns
%matplotlib inline
```

[25]:
```
tips = sns.load_dataset('tips')
tips.head()
```

[25]:

| | 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 |

### 1.1 barplot and countplot

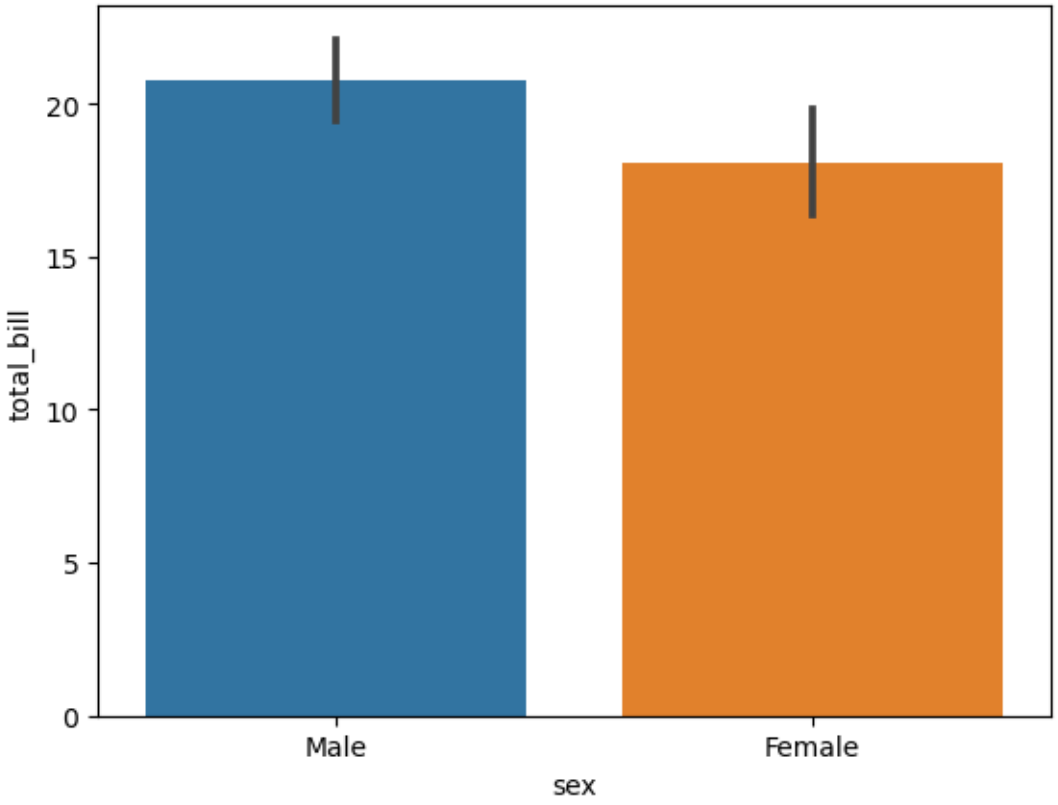
These very similar plots allow you to get aggregate data off a categorical feature in your data. **barplot** is a general plot that allows you to aggregate the categorical data based off some function, by default the mean:

[26]:
```
sns.barplot(x='sex',y='total_bill',data=tips)
```

```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:641:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
```

```
behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```

[26]:
```
<Axes: xlabel='sex', ylabel='total_bill'>
```



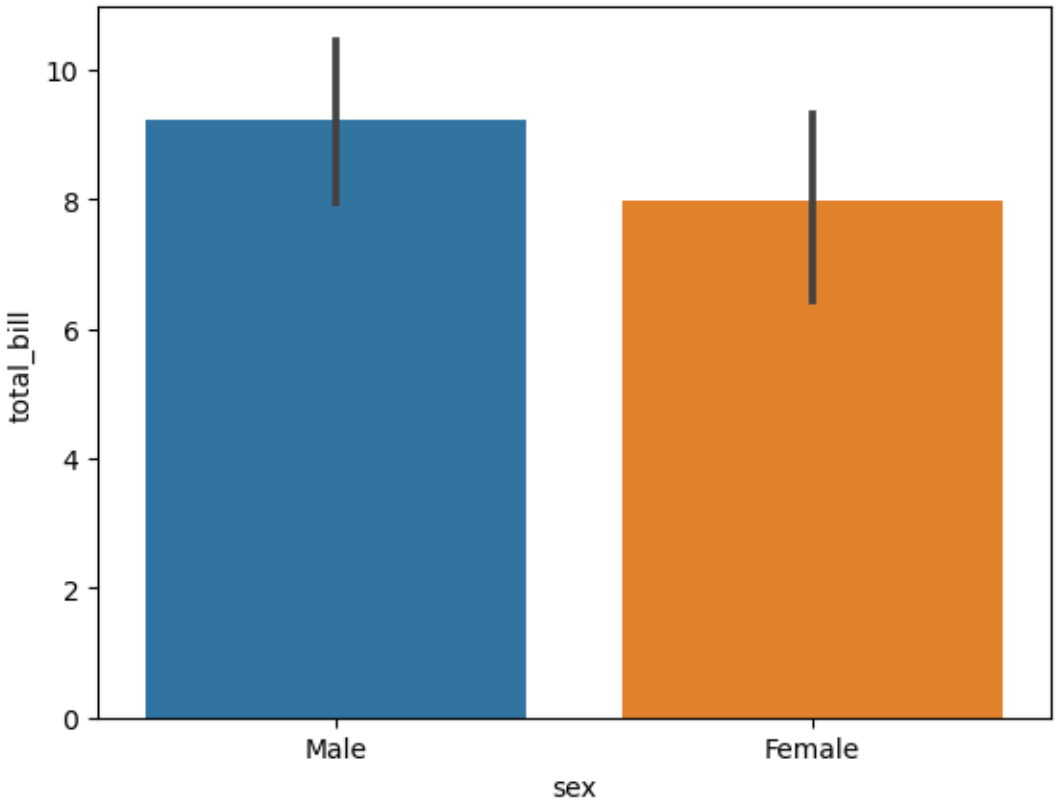

[4]:
```
import numpy as np
```

You can change the estimator object to your own function, that converts a vector to a scalar:

[27]:
```
sns.barplot(x='sex',y='total_bill',data=tips,estimator=np.std)
```

```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:641:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```

[27]:
```
<Axes: xlabel='sex', ylabel='total_bill'>
```

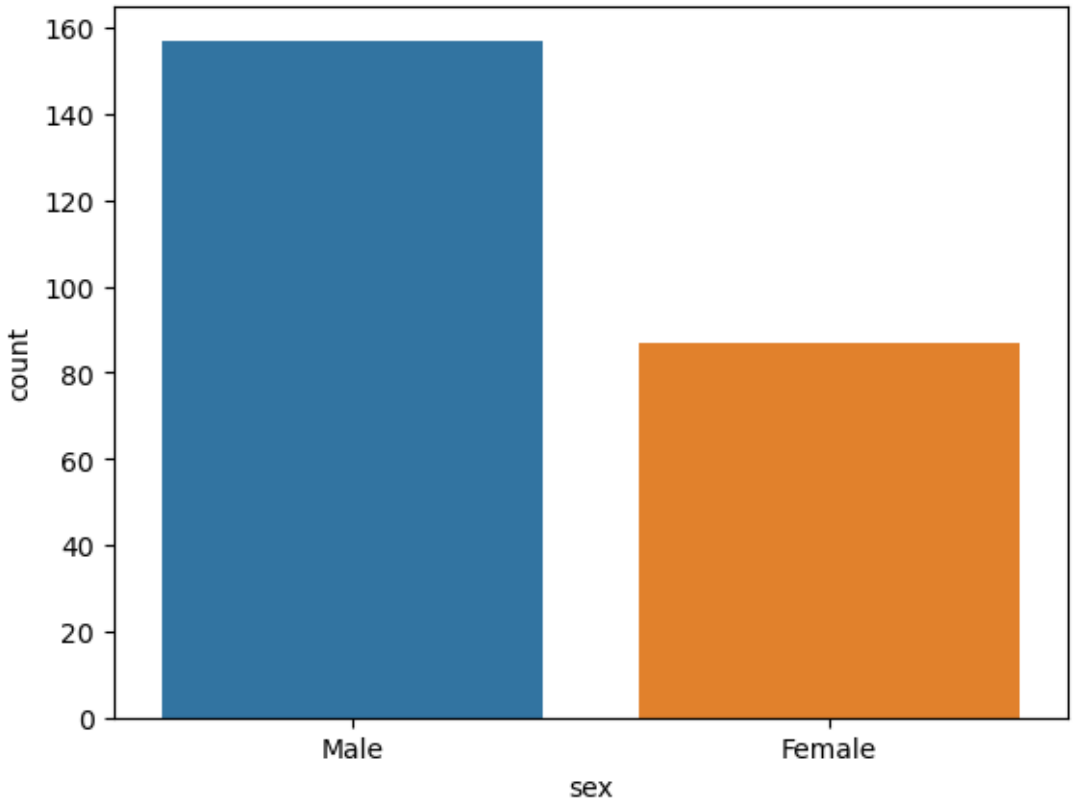



### 1.1.1 countplot

This is essentially the same as barplot except the estimator is explicitly counting the number of occurrences. Which is why we only pass the x value:

[28]:
```
sns.countplot(x='sex',data=tips)
```

[28]:
```
<Axes: xlabel='sex', ylabel='count'>
```



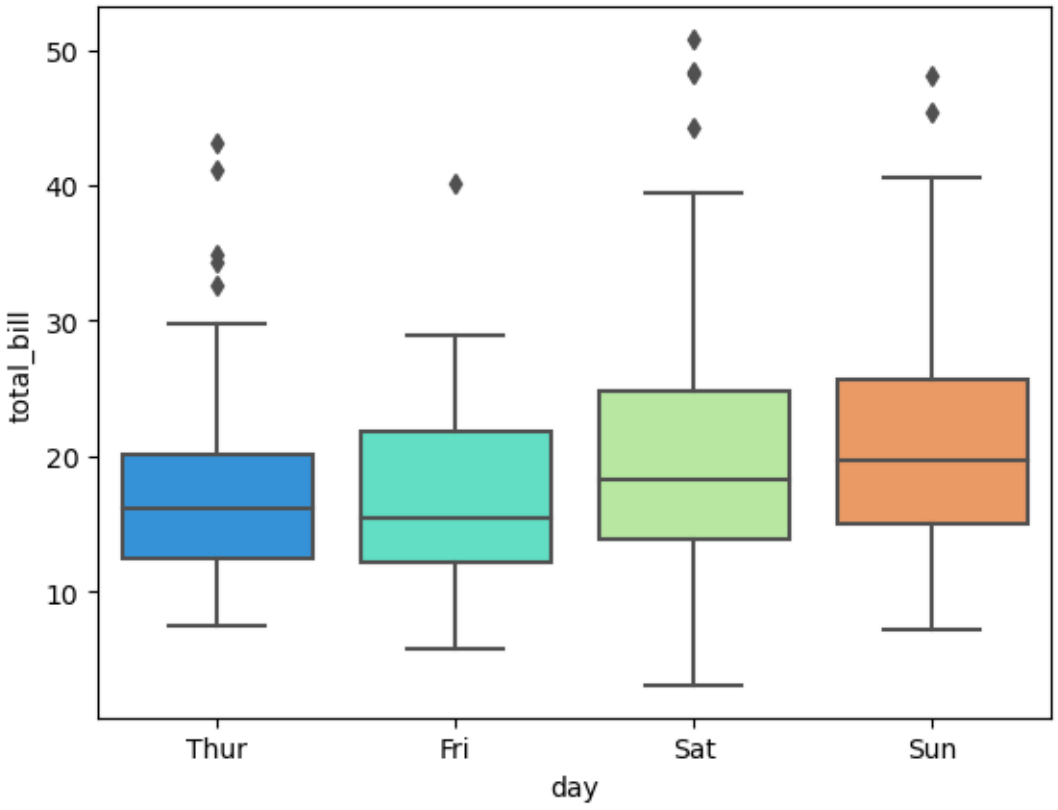

### 1.2 boxplot and violinplot

boxplots and violinplots are used to shown the distribution of categorical data. A box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable. The box shows the quartiles of the dataset while the whiskers extend to show the rest of the distribution, except for points that are determined to be "outliers" using a method that is a function of the inter-quartile range.

[29]:
```
sns.boxplot(x="day", y="total_bill", data=tips,palette='rainbow')
```

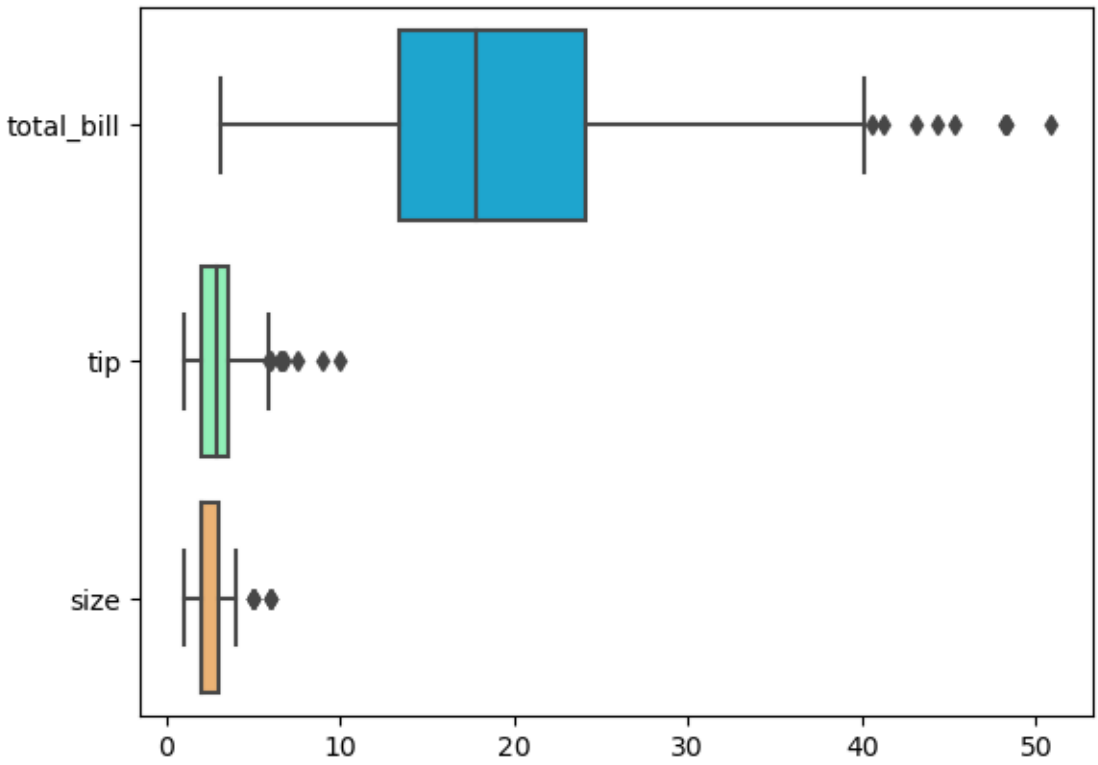
```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:641:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```

[29]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```




[30]:
```
# Can do entire dataframe with orient='h'
sns.boxplot(data=tips,palette='rainbow',orient='h')
```

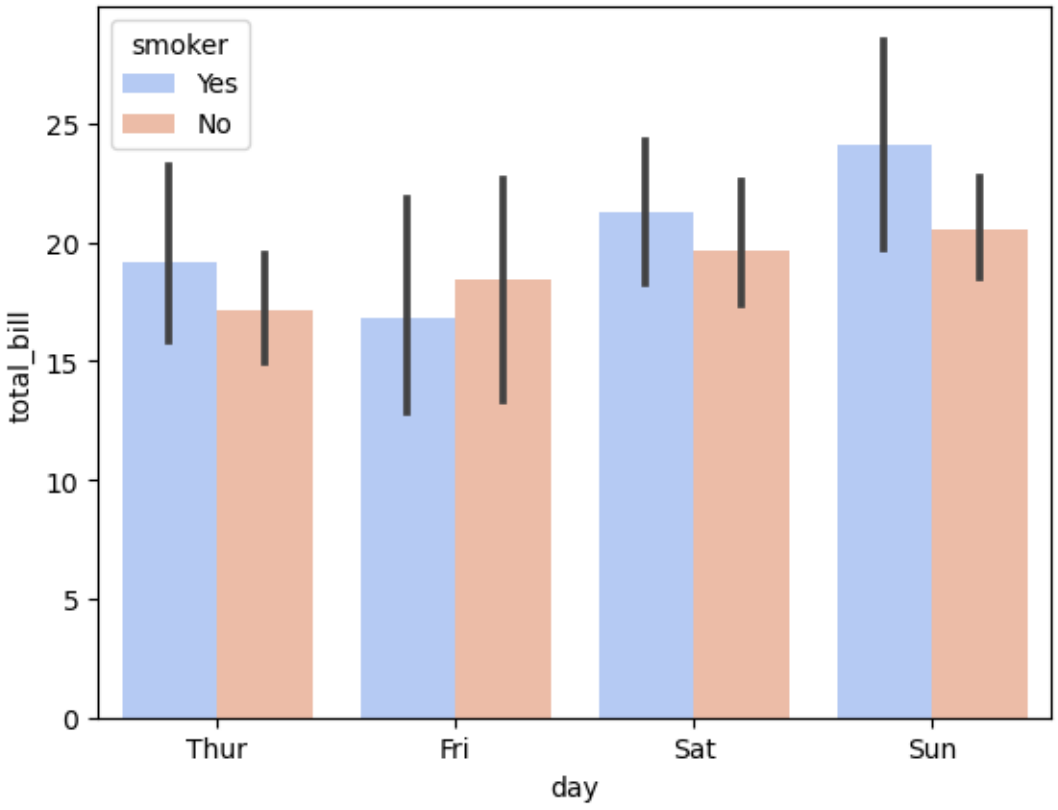
[30]: <Axes: >




[32]:
```
sns.barplot(x="day", y="total_bill", hue="smoker",data=tips, palette="coolwarm")
```

```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:641:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```

[32]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```



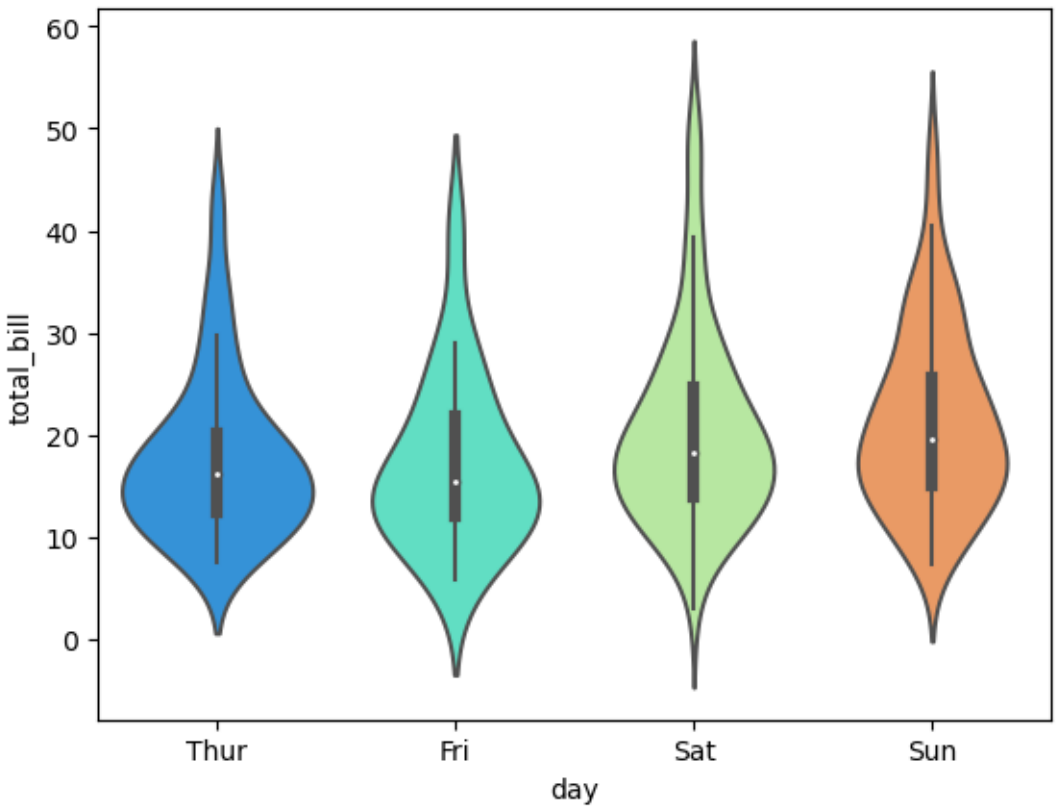

### 1.2.1 violinplot

A violin plot plays a similar role as a box and whisker plot. It shows the distribution of quantitative data across several levels of one (or more) categorical variables such that those distributions can be compared. Unlike a box plot, in which all of the plot components correspond to actual datapoints, the violin plot features a kernel density estimation of the underlying distribution.

[33]:
```
sns.violinplot(x="day", y="total_bill", data=tips,palette='rainbow')
```

```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:641:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```

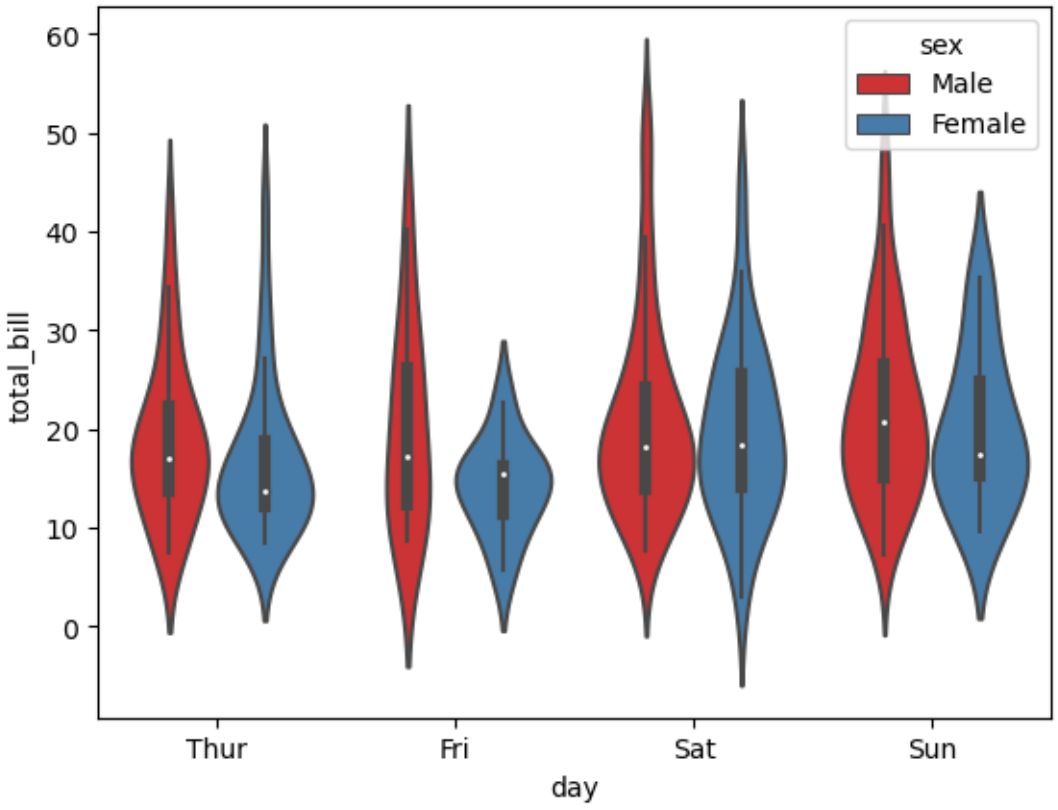
[33]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```




[34]:
```
sns.violinplot(x="day", y="total_bill", data=tips,hue='sex',palette='Set1')
```

```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:641:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```

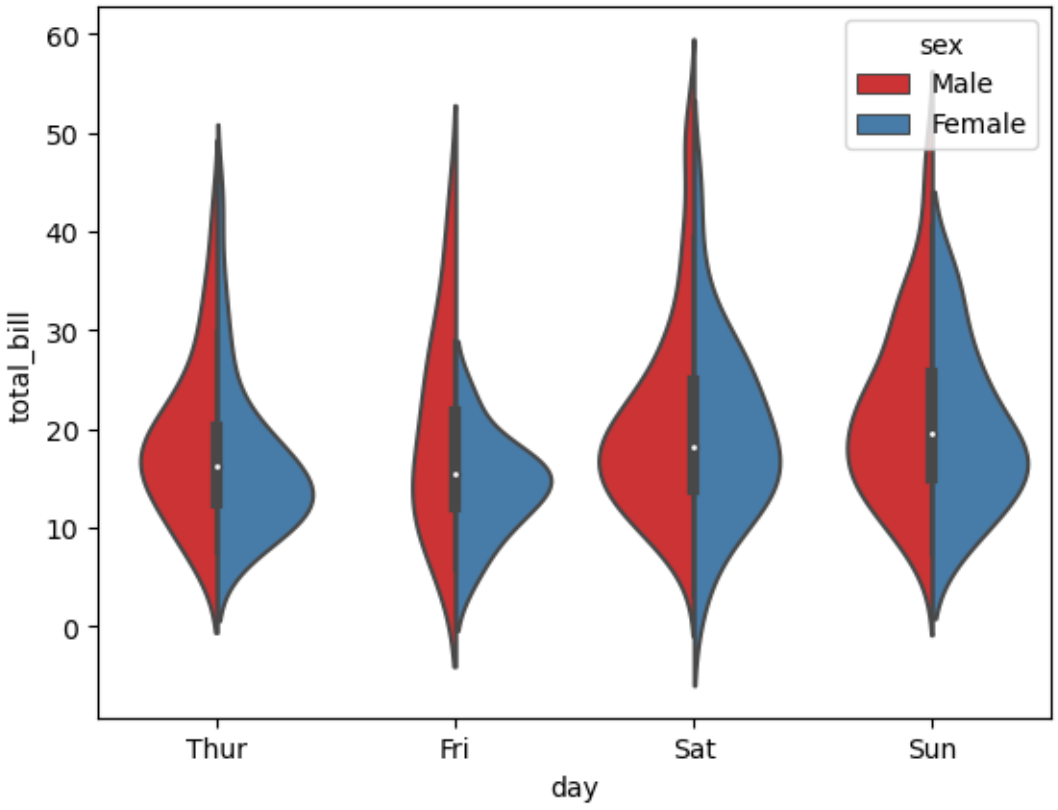
[34]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```




[35]:
```
sns.violinplot(x="day", y="total_bill",␣
↪data=tips,hue='sex',split=True,palette='Set1')
```

```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:641:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```

[35]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```




### 1.3 stripplot and swarmplot

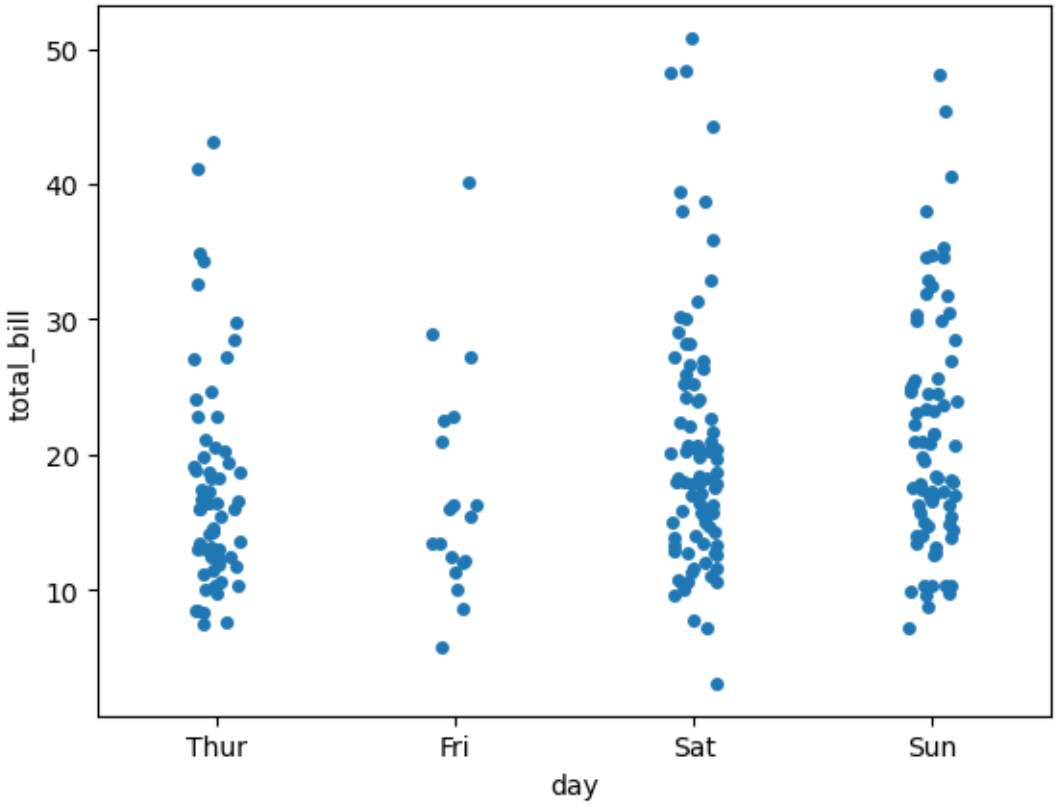
The stripplot will draw a scatterplot where one variable is categorical. A strip plot can be drawn on its own, but it is also a good complement to a box or violin plot in cases where you want to show all observations along with some representation of the underlying distribution.

The swarmplot is similar to stripplot(), but the points are adjusted (only along the categorical axis) so that they don't overlap. This gives a better representation of the distribution of values, although it does not scale as well to large numbers of observations (both in terms of the ability to show all the points and in terms of the computation needed to arrange them).

[13]:
```
sns.stripplot(x="day", y="total_bill", data=tips)
```

```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

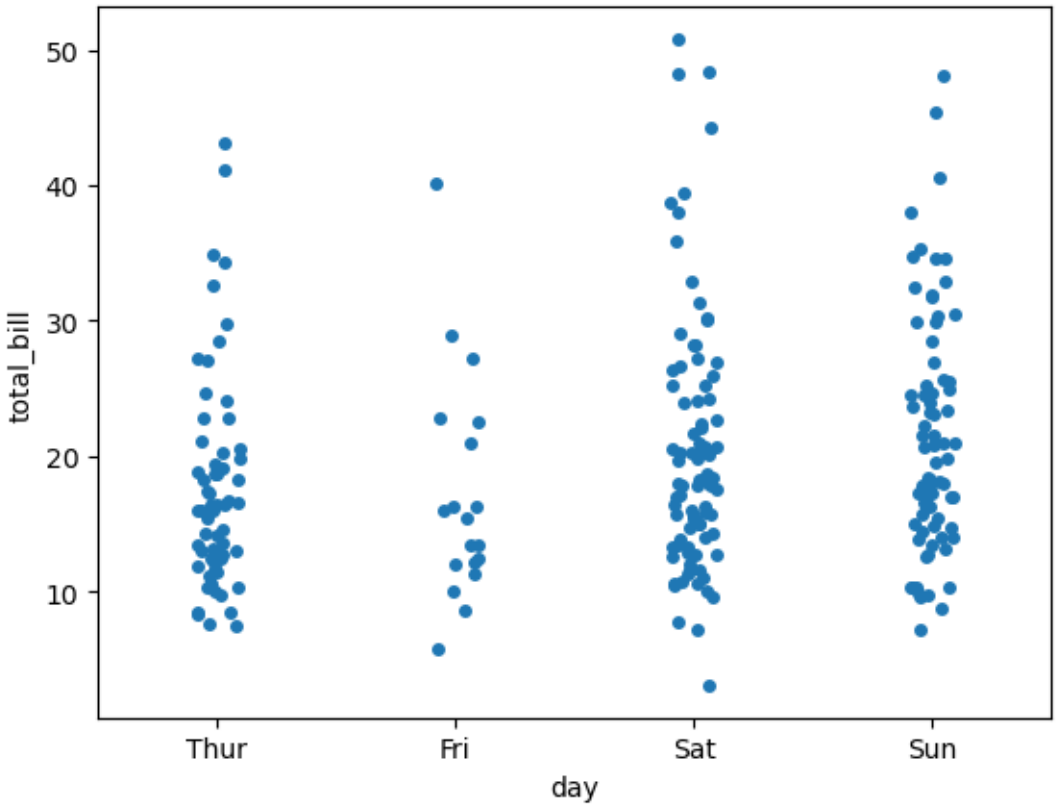
[13]: <Axes: xlabel='day', ylabel='total_bill'>




[36]:
```
sns.stripplot(x="day", y="total_bill", data=tips,jitter=True)
```

```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

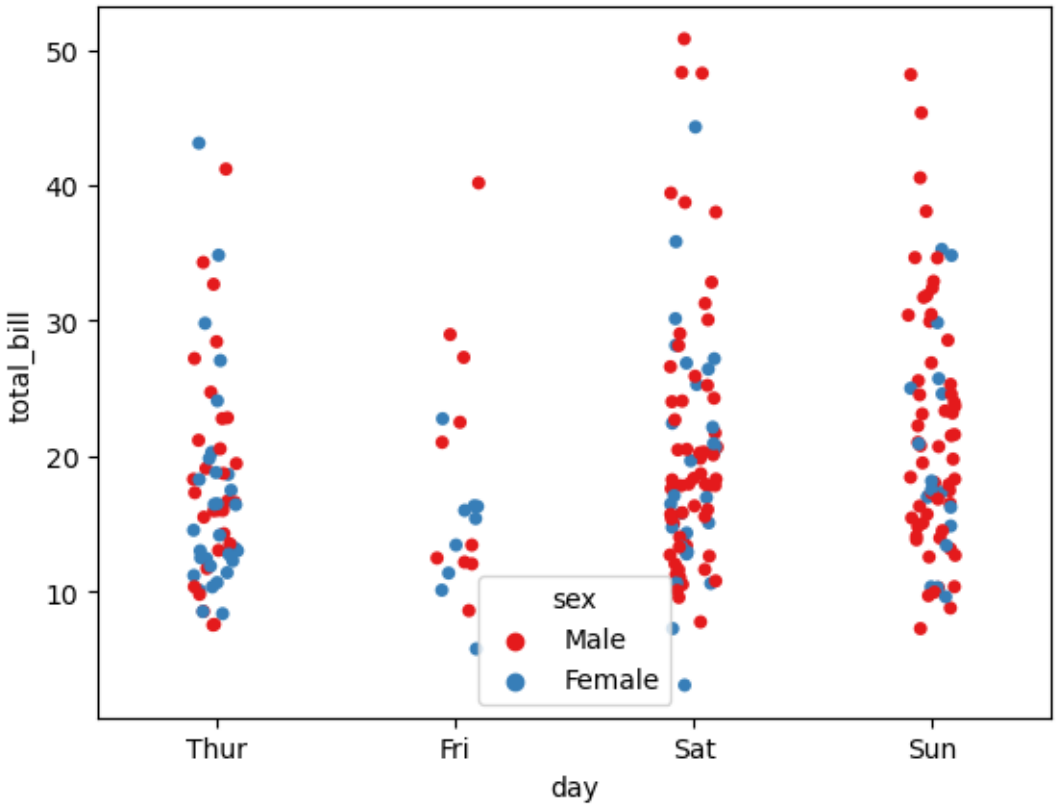
[36]: <Axes: xlabel='day', ylabel='total_bill'>




[37]:
```
sns.stripplot(x="day", y="total_bill",␣
↪data=tips,jitter=True,hue='sex',palette='Set1')
```

```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

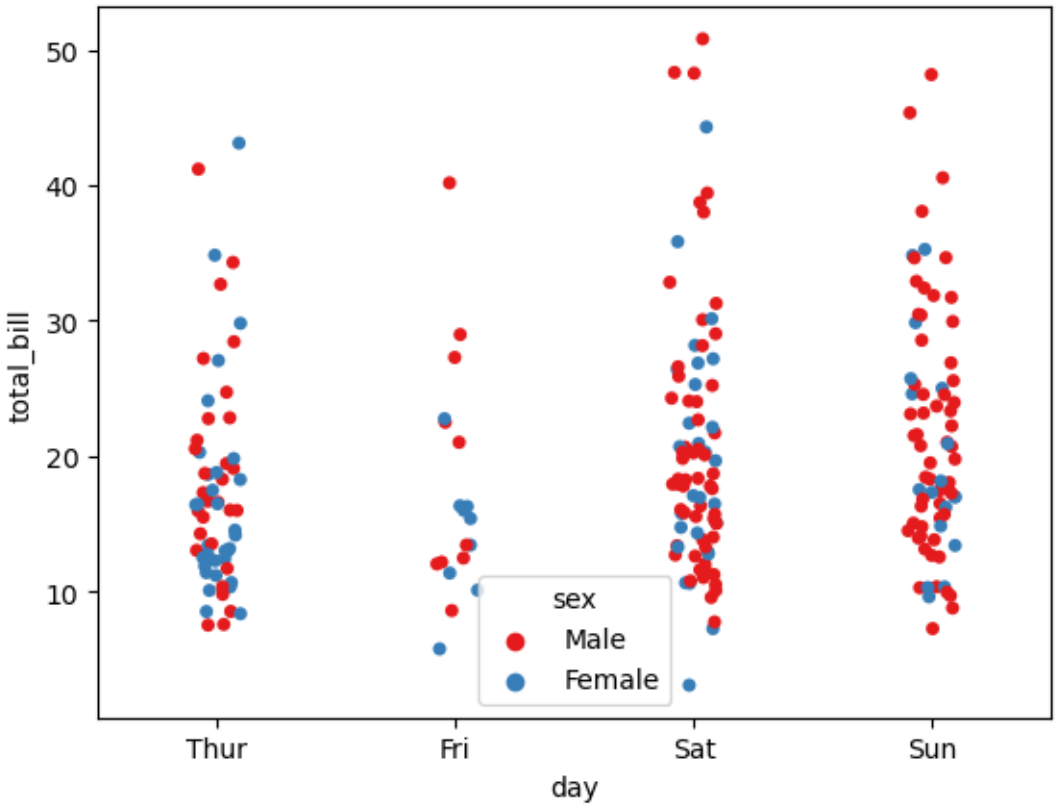
[37]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```




[38]:
```
sns.stripplot(x="day", y="total_bill",␣
↪data=tips,jitter=True,hue='sex',palette='Set1')
```

```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

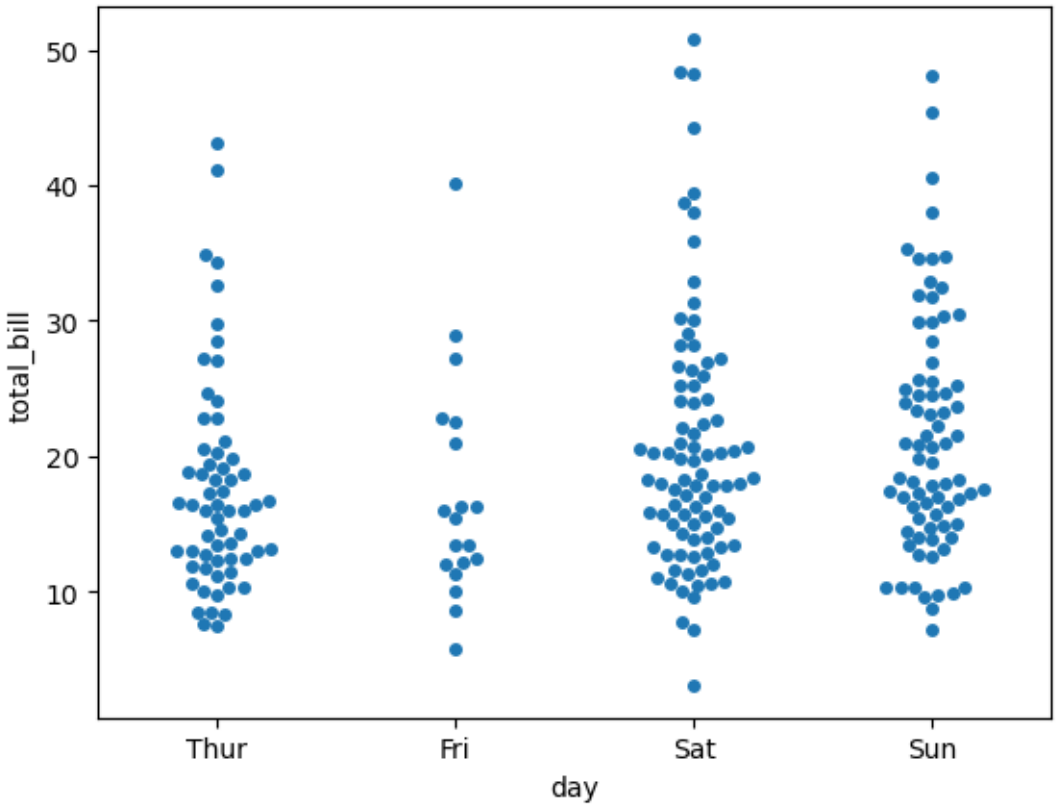
[38]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```




[39]:
```
sns.swarmplot(x="day", y="total_bill", data=tips)
```

```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

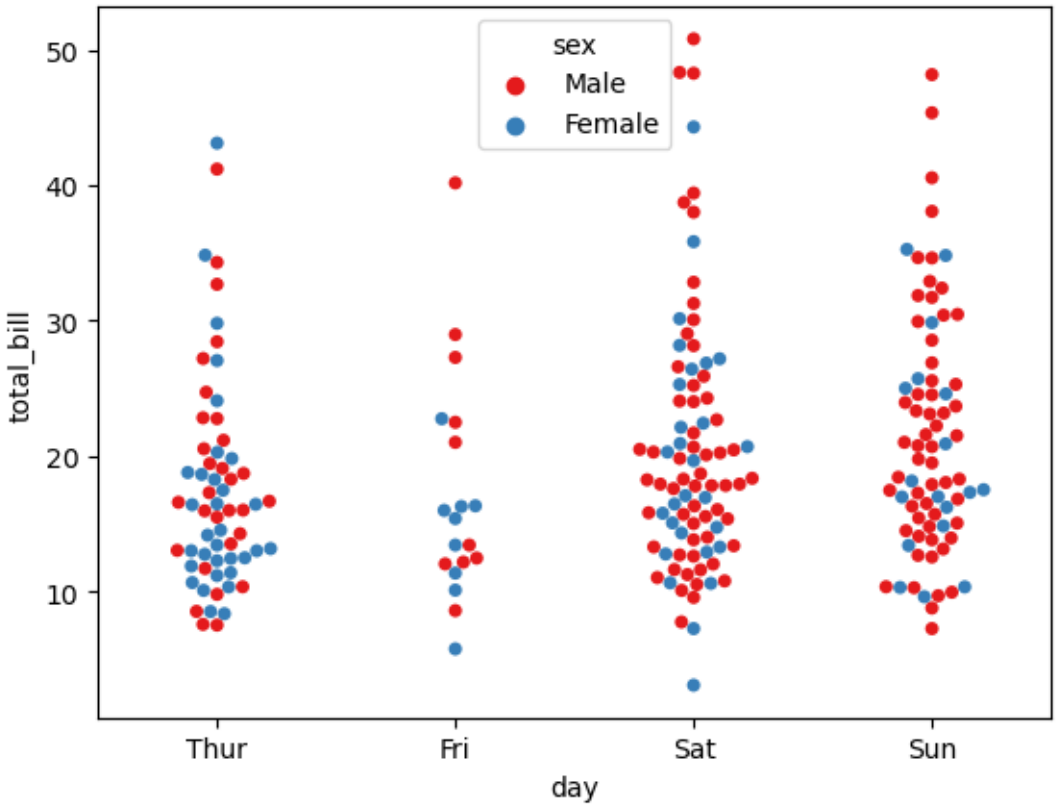
[39]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```




[40]:
```
sns.swarmplot(x="day", y="total_bill",hue='sex',data=tips, palette="Set1")
```

```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

[40]:
```
<Axes: xlabel='day', ylabel='total_bill'>
```

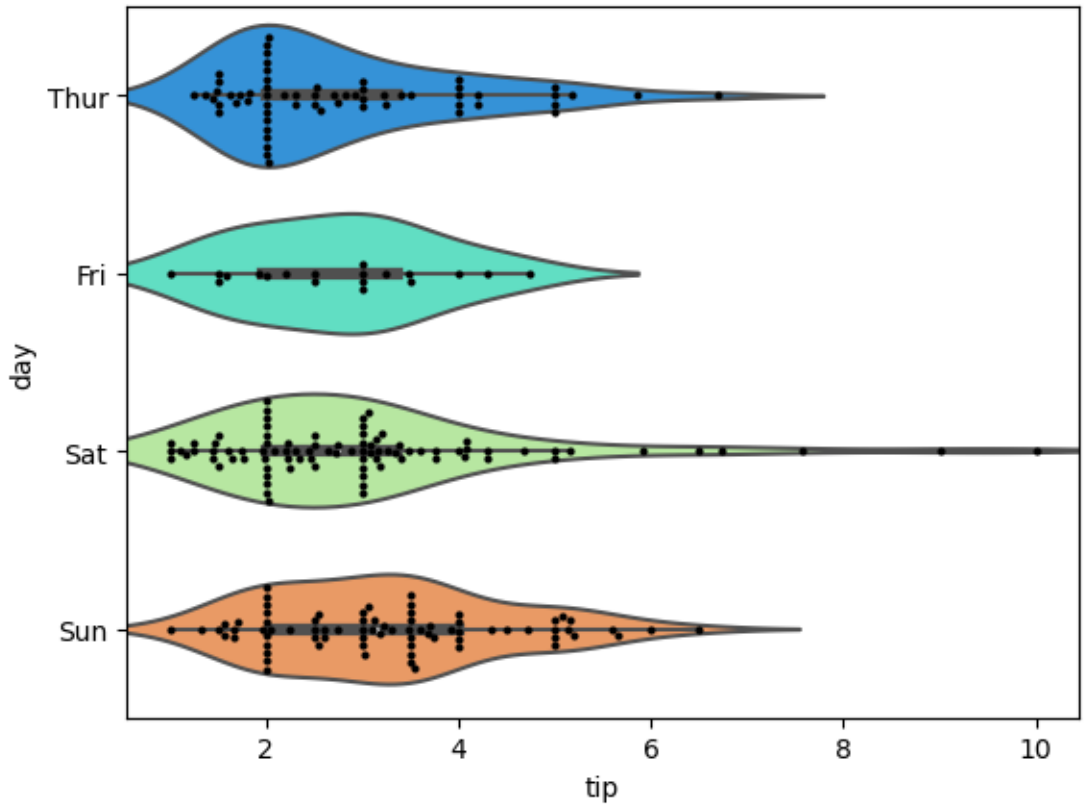



### 1.3.1 Combining Categorical Plots

[42]:
```
sns.violinplot(x="tip", y="day", data=tips,palette='rainbow')
sns.swarmplot(x="tip", y="day", data=tips,color='black',size=3)
```

```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:641:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

[42]: <Axes: xlabel='tip', ylabel='day'>




### 1.4 factorplot

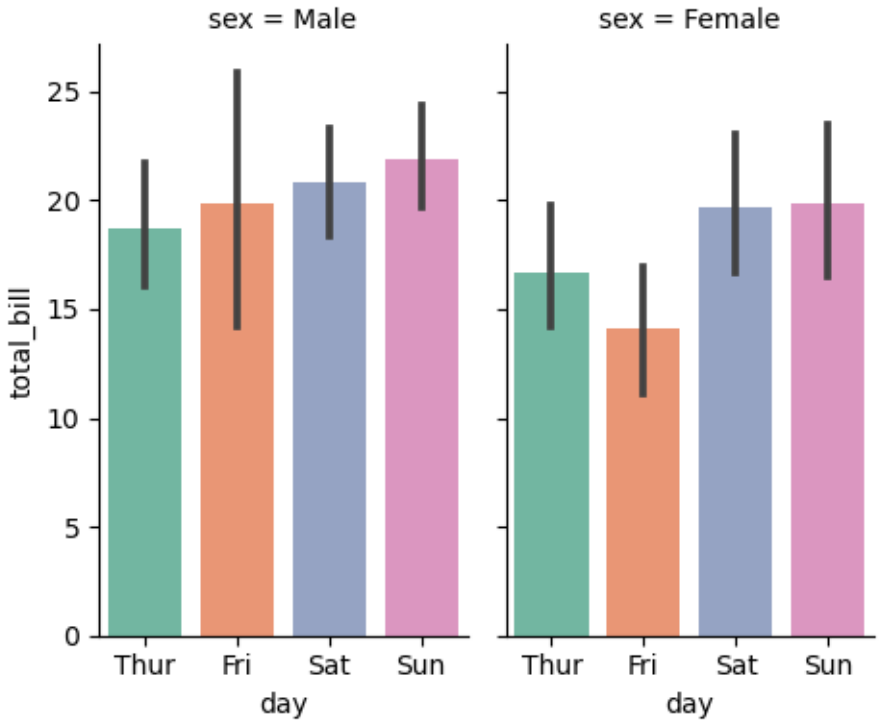
factorplot is the most general form of a categorical plot. It can take in a **kind** parameter to adjust the plot type:

[50]:
```
sns.catplot(
    data=tips, x="day", y="total_bill", col="sex",
    kind="bar", height=4, aspect=.6,palette='Set2')
```

```
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:641:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:641:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
/home/fischer/anaconda3/lib/python3.11/site-packages/seaborn/categorical.py:641:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
```

```
behavior or observed=True to adopt the future default and silence this warning.
  grouped_vals = vals.groupby(grouper)
```

[50]: <seaborn.axisgrid.FacetGrid at 0x7f90678cd210>




[46]:
```
sns.color_palette()
```

[46]:
```
[(0.12156862745098039, 0.4666666666666667, 0.7058823529411765),
 (1.0, 0.4980392156862745, 0.054901960784313725),
 (0.17254901960784313, 0.6274509803921569, 0.17254901960784313),
 (0.8392156862745098, 0.15294117647058825, 0.1568627450980392),
 (0.5803921568627451, 0.403921568627451, 0.7411764705882353),
 (0.5490196078431373, 0.33725490196078434, 0.29411764705882354),
 (0.8901960784313725, 0.4666666666666667, 0.7607843137254902),
 (0.4980392156862745, 0.4980392156862745, 0.4980392156862745),
 (0.7372549019607844, 0.7411764705882353, 0.13333333333333333),
 (0.09019607843137255, 0.7450980392156863, 0.8117647058823529)]
```

[ ]: