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!

In [1]:

```
import seaborn as sns
%matplotlib inline
```

In [2]:

```
tips = sns.load_dataset('tips')
tips.head()
```

Out[2]:

	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

barplot and countplot

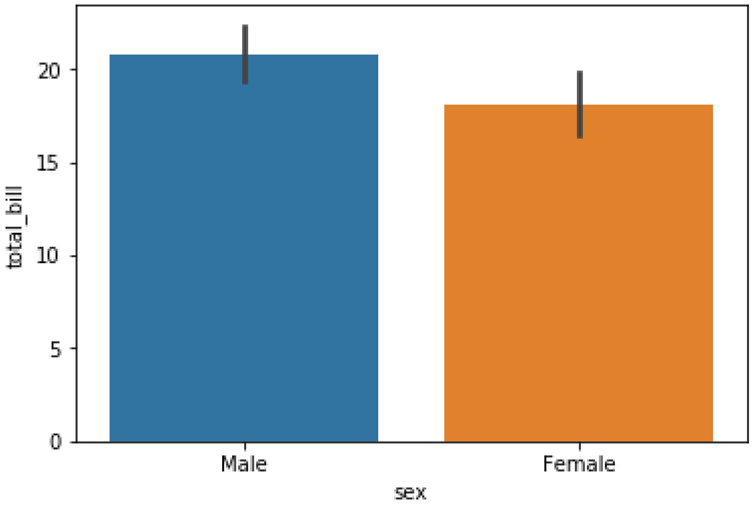
These very similar plots allow you to get aggregate data off a categorical feature in your data. **barplot** is a general plot that allows you to aggregate the categorical data based off some function, by default the mean:

In [4]:

```
sns.barplot(x='sex',y='total_bill',data=tips)
```

Out[4]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1559f13208>
```

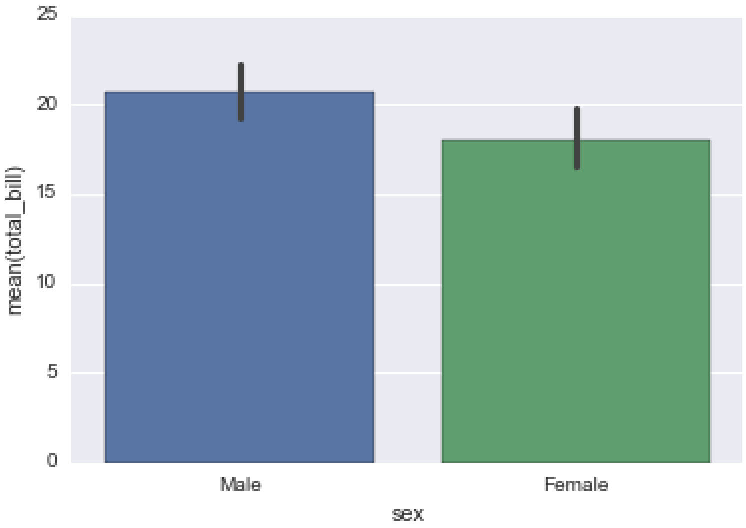


In [8]:

```
sns.barplot(x='sex',y='total_bill',data=tips)
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11c99b8d0>
```



In [6]:

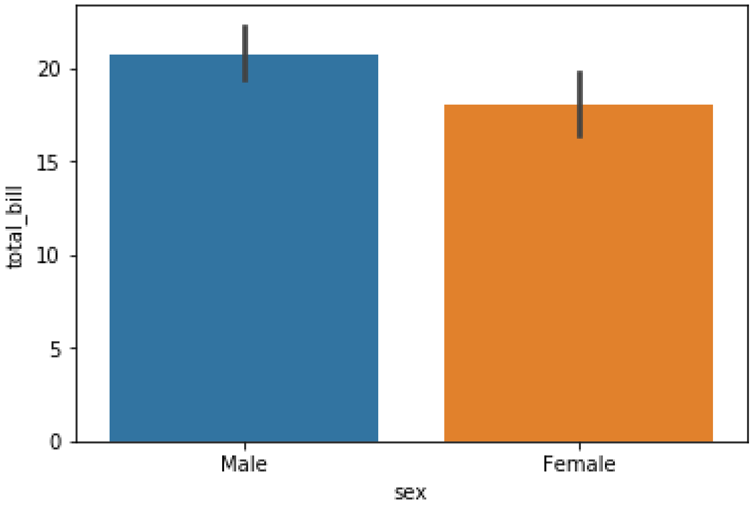
```
import numpy as np
```

In [8]:

```
sns.barplot(x='sex',y='total_bill',data=tips,estimator=np.mean)
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f155a3c7d68>
```

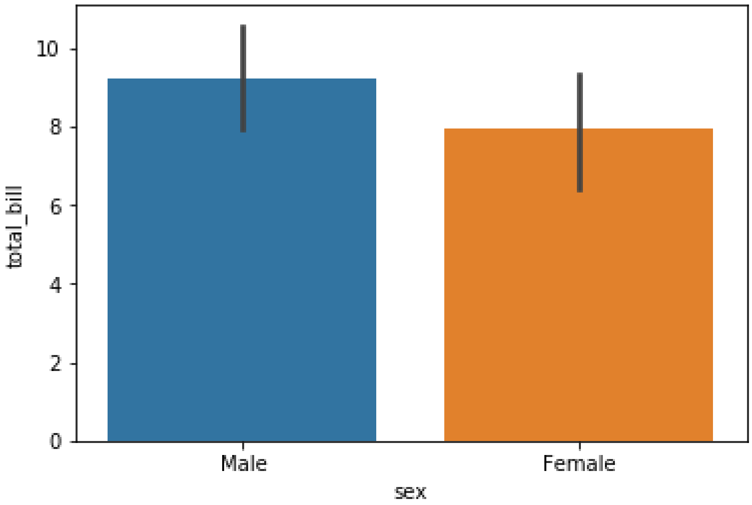


In [9]:

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

Out[9]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f155a3f24a8>
```



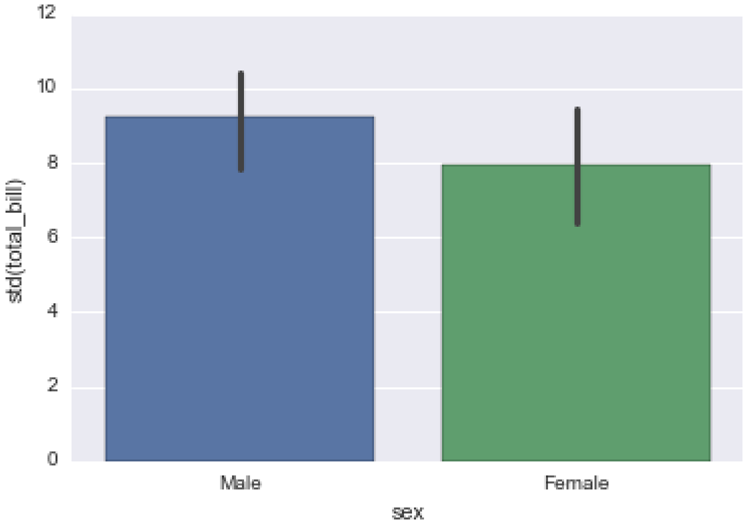
You can change the estimator object to your own function, that converts a vector to a scalar:

In [11]:

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

Out[11]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11c9b00b8>
```



countplot

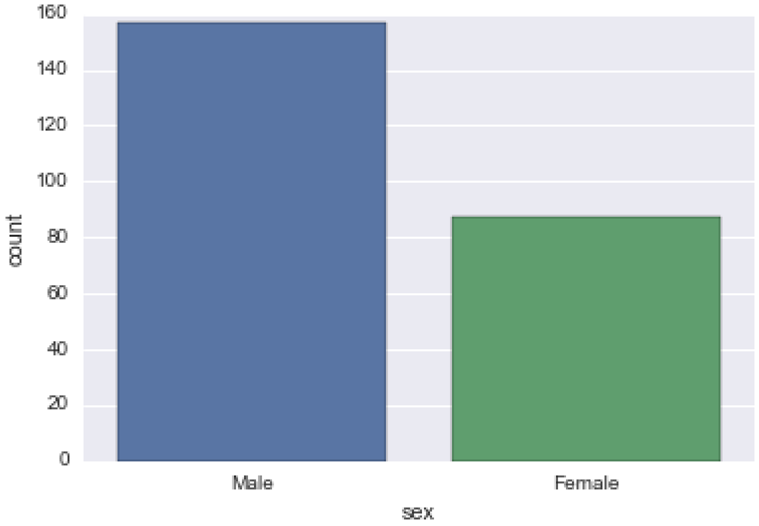
This is essentially the same as barplot except the estimator is explicitly counting the number of occurrences. Which is why we only pass the x value:

In [13]:

```
sns.countplot(x='sex',data=tips)
```

Out[13]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1153276d8>
```

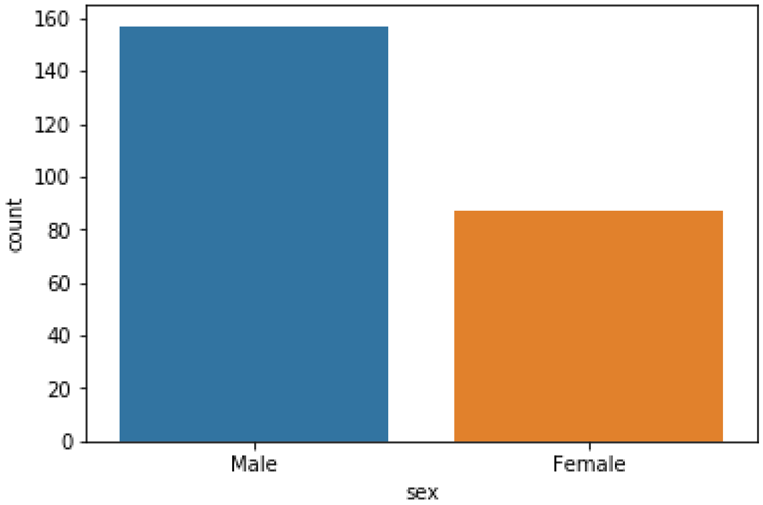


In [10]:

```
sns.countplot(x='sex',data=tips)
```

Out[10]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f155a1245c0>
```



boxplot and violinplot

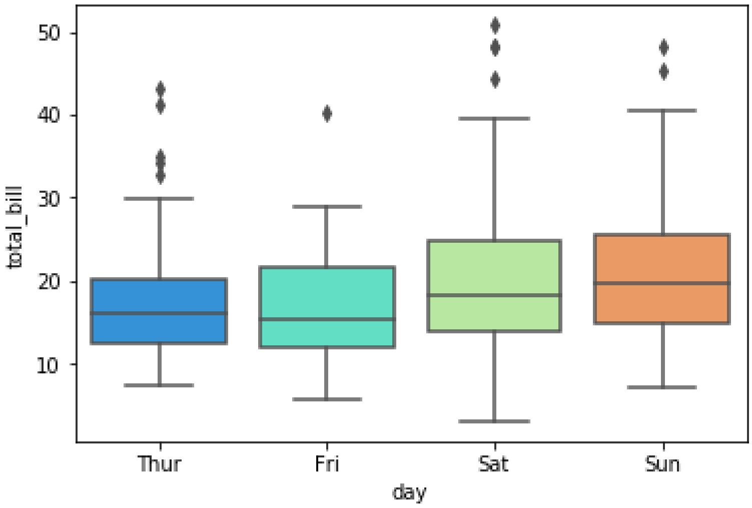
boxplots and violinplots are used to shown the distribution of categorical data. A box plot (or box-and-whisker plot) shows the distribution of quantitative data in a way that facilitates comparisons between variables or across levels of a categorical variable. The box shows the quartiles of the dataset 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.

In [12]:

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

Out[12]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f155a33ebe0>
```

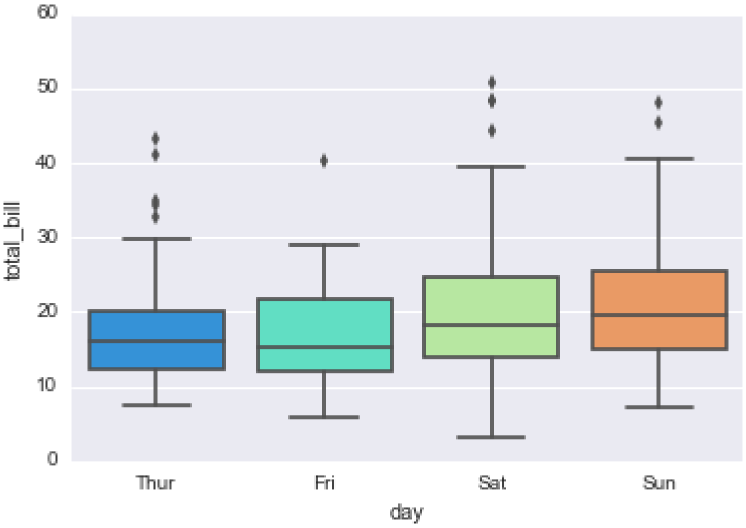


In [22]:

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

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11db81630>
```

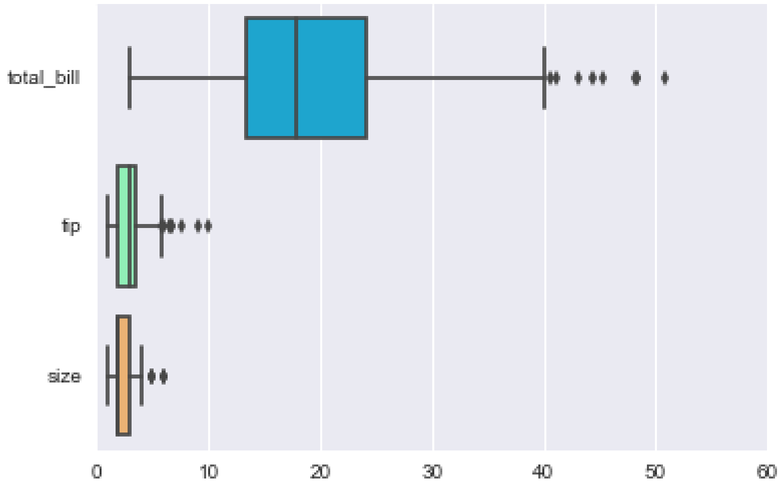


In [25]:

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

Out[25]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11e2c0b00>
```

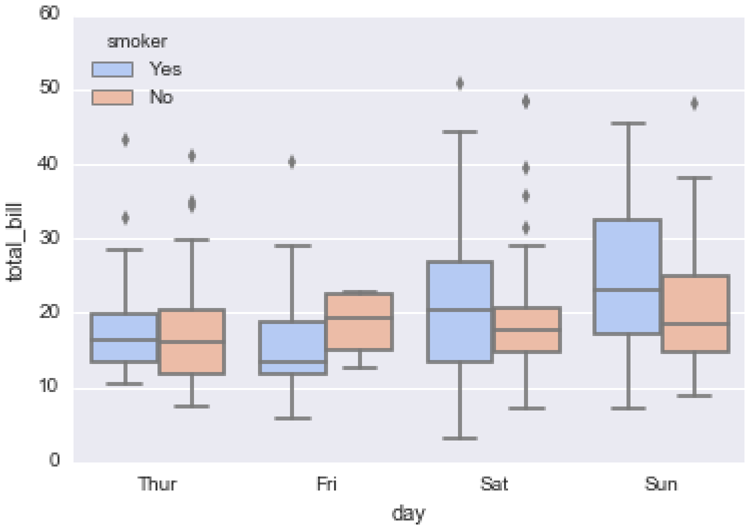


In [26]:

```
sns.boxplot(x="day", y="total_bill", hue="smoker",data=tips, palette="coolwarm"
```

Out[26]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11e2c77f0>
```



violinplot

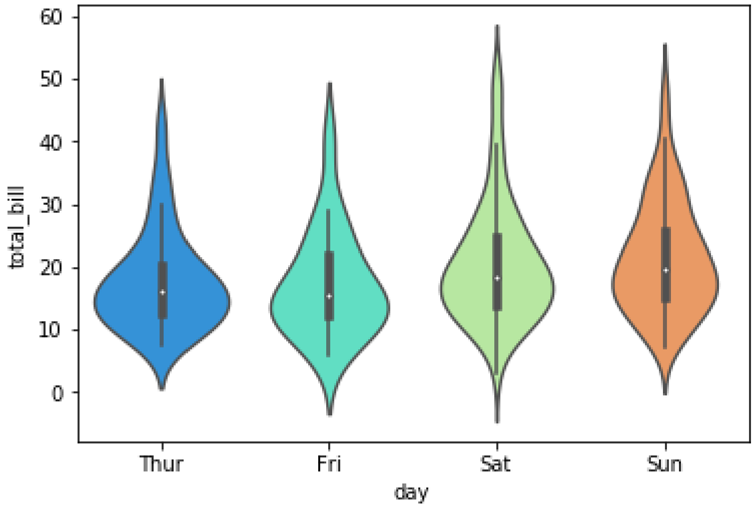
A violin plot plays a similar role as a box and whisker plot. It shows the distribution of quantitative data across several levels of one (or more) categorical variables such that those distributions can be compared. Unlike a box plot, in which all of the plot components correspond to actual datapoints,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

In [13]:

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

Out[13]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f155a46ada0>
```

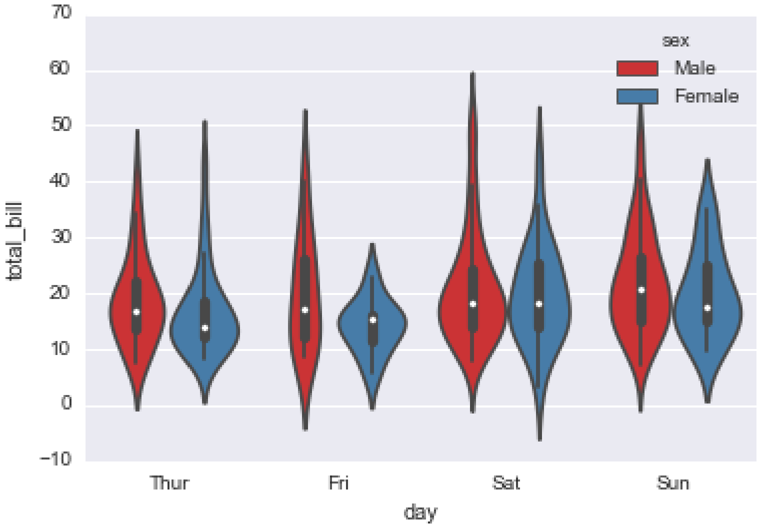


In [37]:

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

Out[37]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11f739dd8>
```

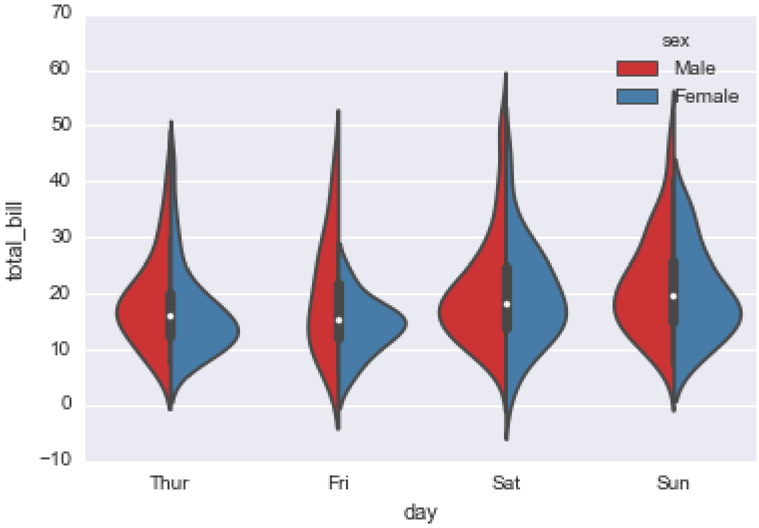


In [36]:

```
sns.violinplot(x="day", y="total_bill", data=tips,hue='sex',split=True,palette=
```

Out[36]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x11f4d0710>
```



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.

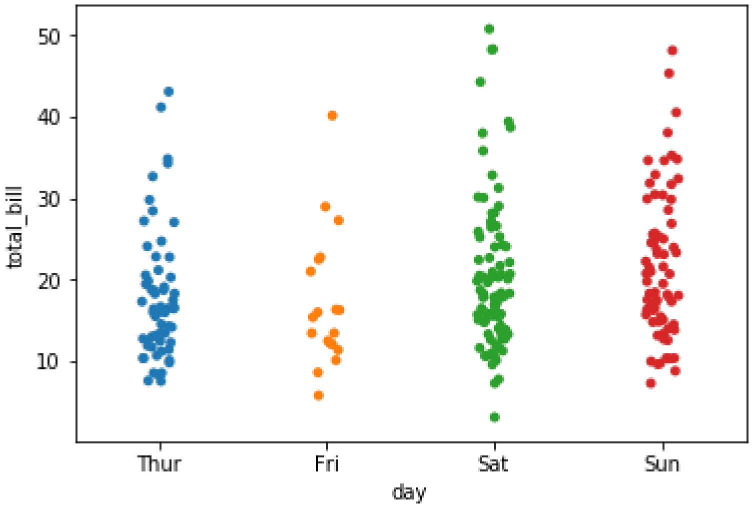
The swarmplot is similar to stripplot(), but the points are adjusted (only along the categorical axis) so that they don’t overlap. This gives a better representation of the distribution of values, although it does not scale as well to large numbers of observations (both in terms of the ability to show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

In [14]:

```
sns.stripplot(x="day", y="total_bill", data=tips)
```

Out[14]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f155a2243c8>
```

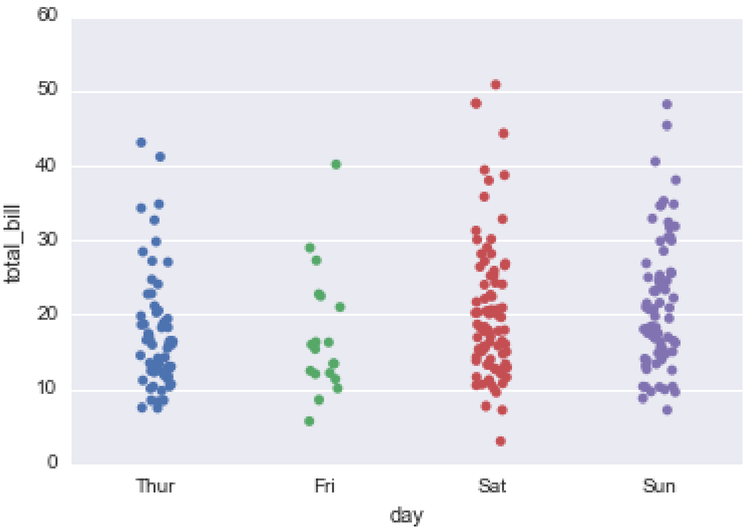


In [39]:

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

Out[39]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1203a8470>
```

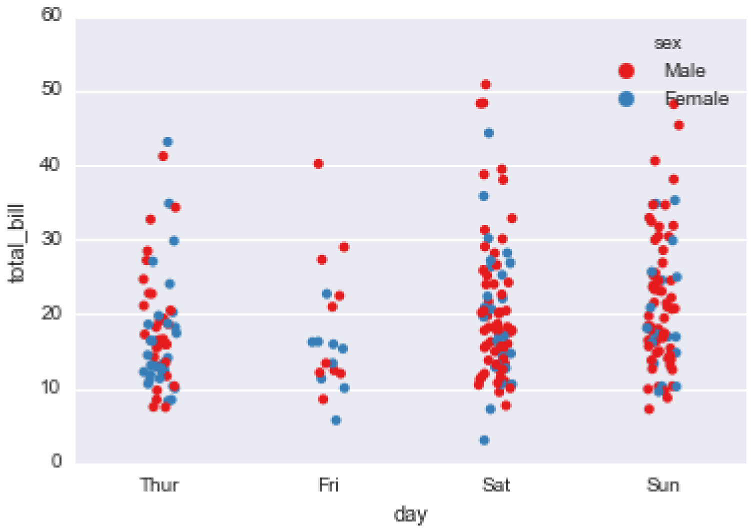


In [42]:

```
sns.stripplot(x="day", y="total_bill", data=tips,jitter=True,hue='sex',palette=
```

Out[42]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x12092e518>
```

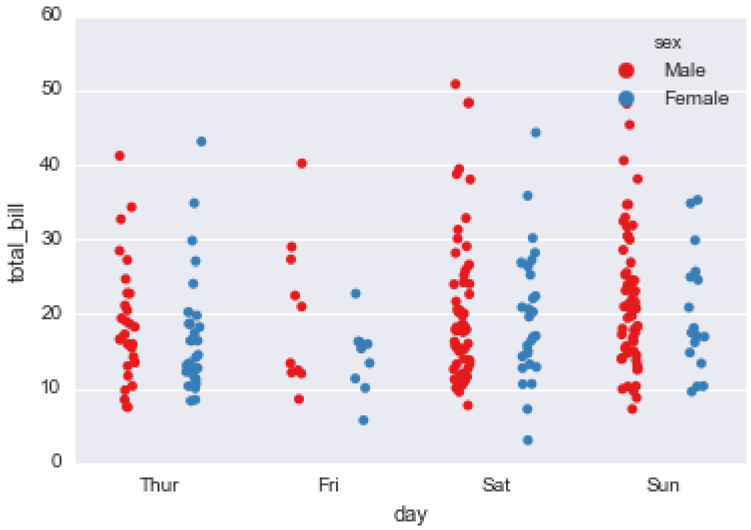


In [43]:

```
sns.stripplot(x="day", y="total_bill", data=tips,jitter=True,hue='sex',palette=
```

Out[43]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x12099db70>
```

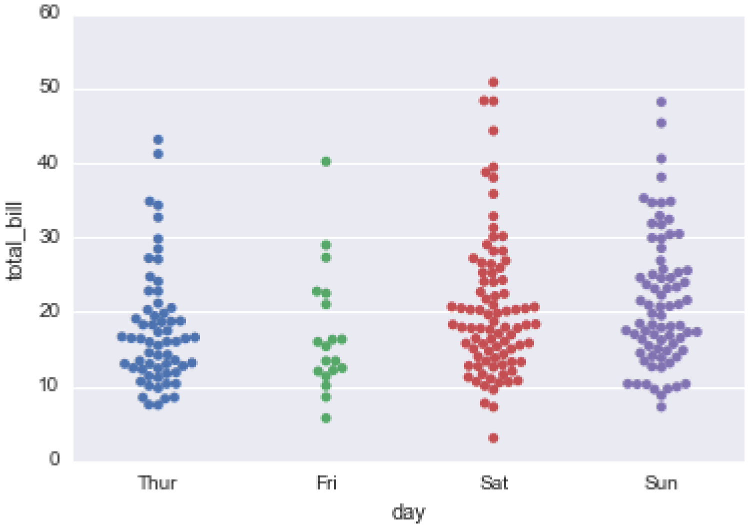


In [44]:

```
sns.swarmplot(x="day", y="total_bill", data=tips)
```

Out[44]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x120c463c8>
```

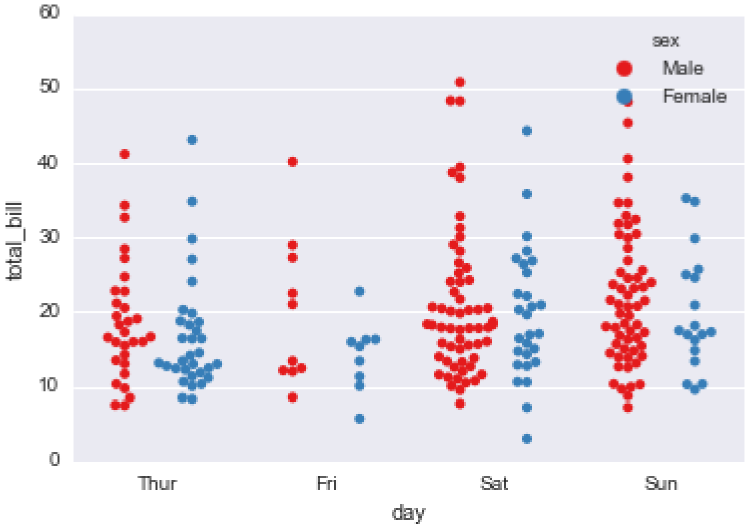


In [47]:

```
sns.swarmplot(x="day", y="total_bill",hue='sex',data=tips, palette="Set1", spli
```

Out[47]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1211b6da0>
```



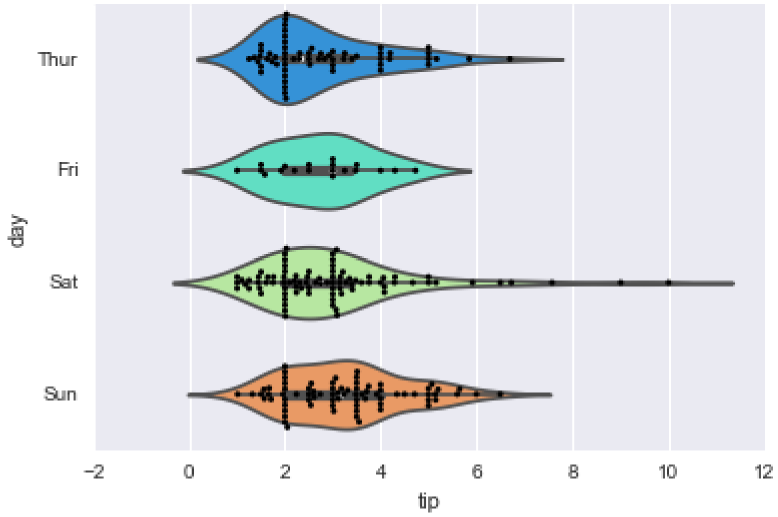
Combining Categorical Plots

In [61]:

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

Out[61]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1228af668>
```



factorplot

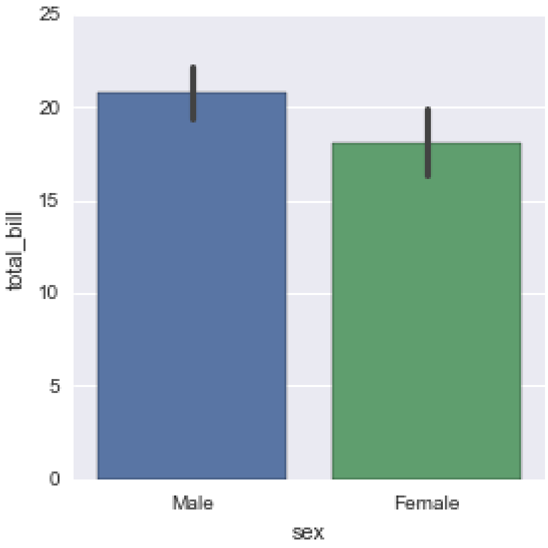
factorplot is the most general form of a categorical plot. It can take in a **kind** parameter to adjust the plot type:

In [15]:

```
sns.factorplot(x='sex',y='total_bill',data=tips,kind='bar')
```

Out[15]:

```
<seaborn.axisgrid.FacetGrid at 0x11d03a278>
```



Great Job!