**Introduction to categorical plots using Seaborn**

Creating and updating categories is only part of using categorical data. Let's start working on building visualizations that use categorical data.

**Our third dataset**

In this chapter, we will use a new dataset, the Las Vegas TripAdvisor reviews dataset. This dataset contains information on 504 reviews from 21 hotels in Las Vegas collected in 2015.

**Las Vegas reviews**

Using the dot-info method on our dataset, we can see that we have information on the hotel guest, such as their home country and traveler type, as well as information on the hotel, such as if it has a pool, gym, tennis court, or other amenities.

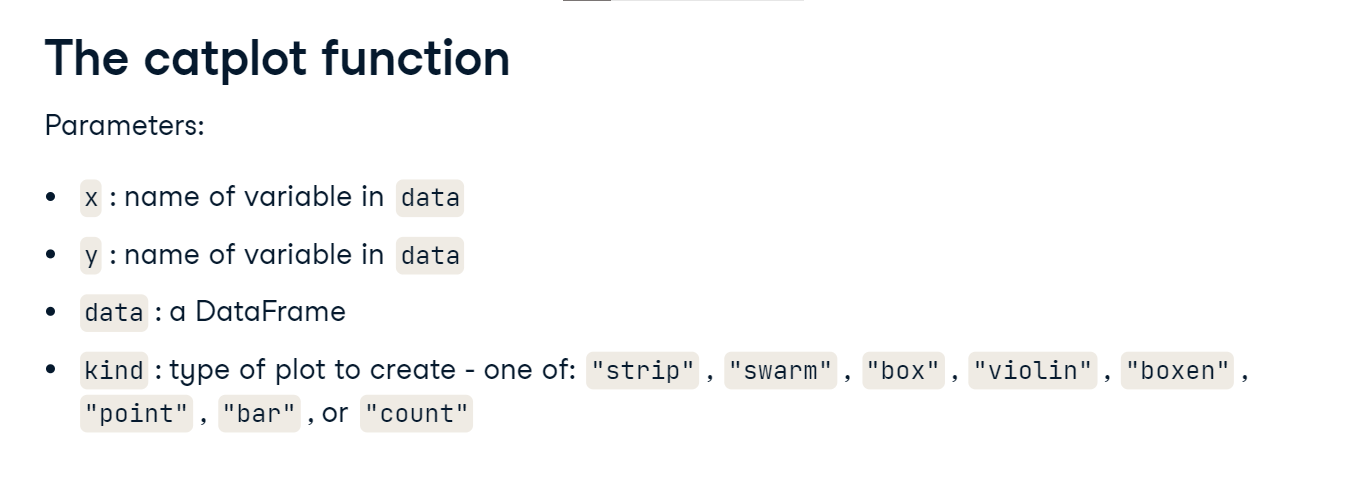
1. 1 https://www.kaggle.com/crawford/las-vegas-tripadvisor-reviews

**Seaborn**

To visualize this data, we are going to use the Python library seaborn, which we have loaded as sns. Datacamp offers two great courses on seaborn - if you'd like additional practice, definitely give those courses a try. For our purposes, we will focus solely on categorical plots using seaborn, and more specifically, the catplot function. Note that Seaborn is based off of the Python library matplotlib, so we have loaded matplotlib's pyplot as plt. Depending on the environment you are coding in, you may need to run plt-dot-show after creating your graphic for it to display.

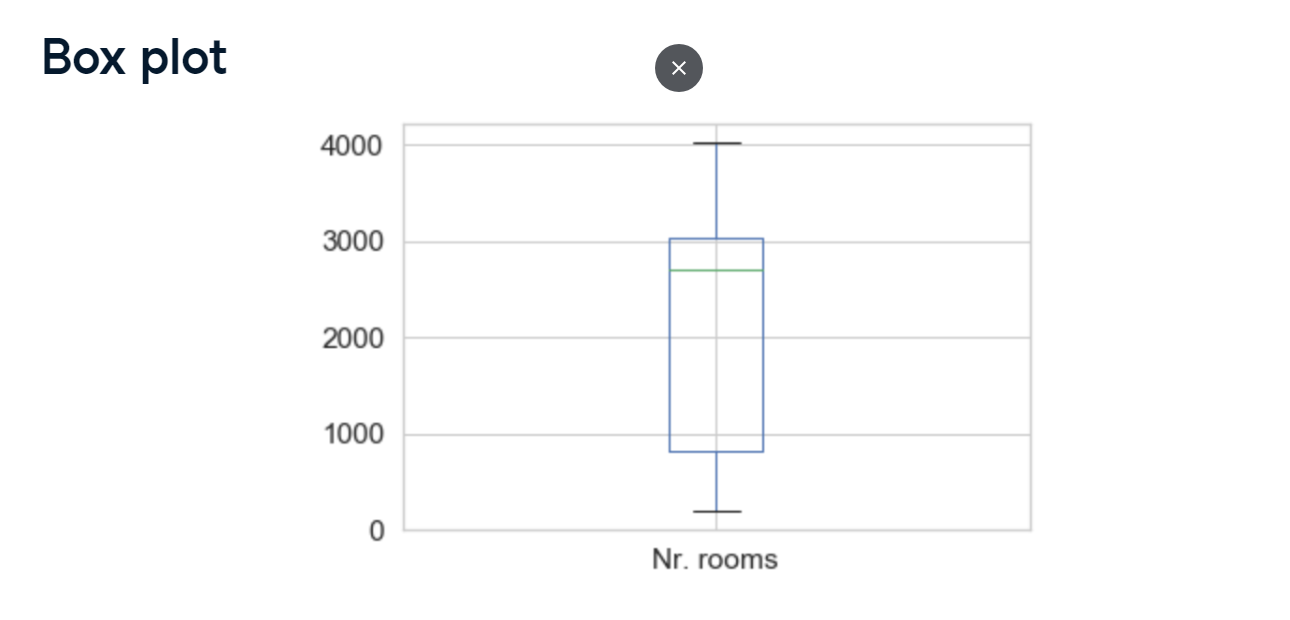
**The catplot function**

Whether we are creating scatterplots, distribution plots, or just counting the number of responses, the catplot function is capable of handling the task. Let's look at the common parameters of a catplot. Both the x and y parameter are names of variables found in the DataFrame being used, while the kind parameter specifies the type of graphic to create. In this chapter we will cover several uses of the kind parameter.



**Box plot**

One type of plot that catplot can create is a box plot. As a reminder, a box plot shows information on the quartiles of numerical data. In this example, we are looking at the number of rooms in hotels. The middle line of the box shows the median of the data, which is hovering around 2,800 beds. The bottom and top of the box show the 25th and 75th quartiles and look to be around 800 and 3000 beds respectively. Consult the linked wikipedia page if you need a refresher on the other elements of a box plot.

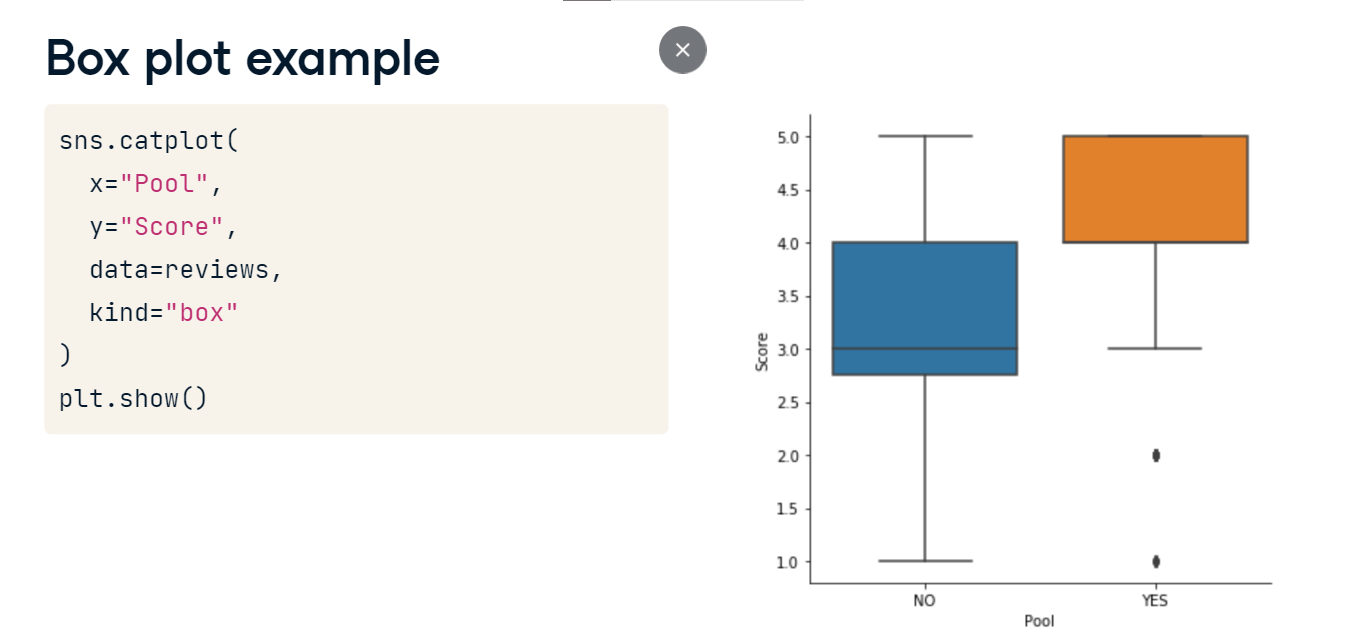


**Review score**

Before we look at an example, let's understand the numerical column we are going to explore. The review score is a value between one and five, and is the rating of the hotel given by the person doing the review. Most scores are four or five, but there are a few that are three and below.

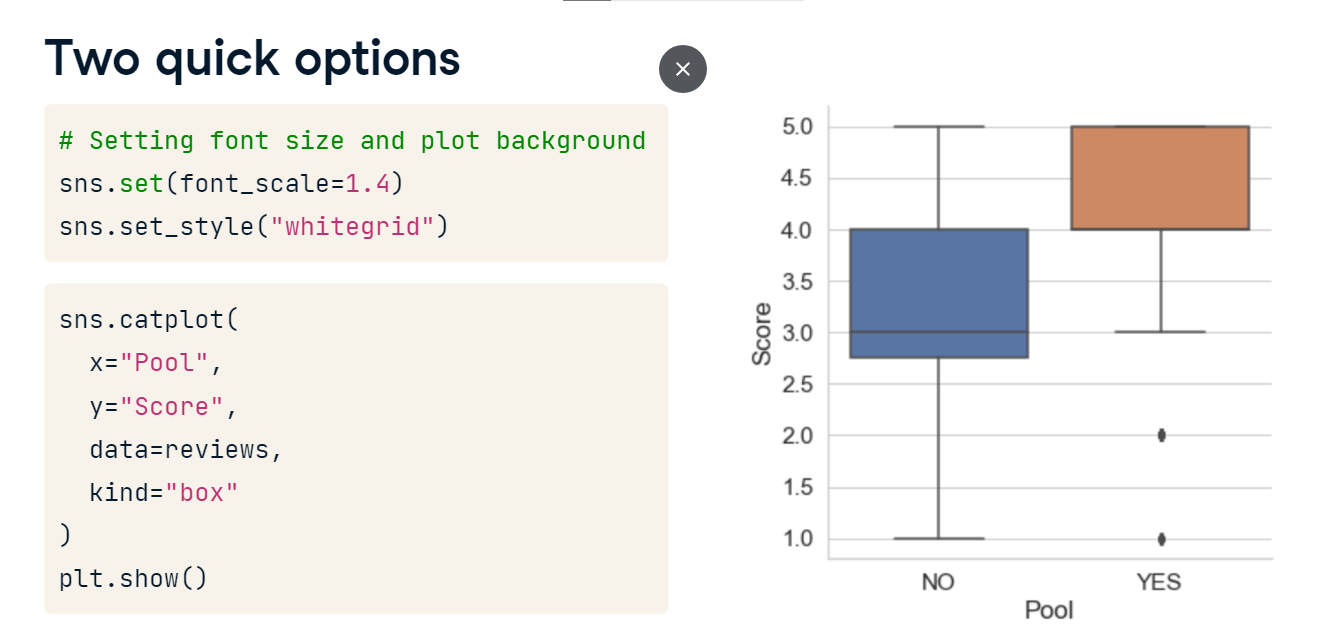
**Box plot example**

Let's look at the review score, across the categorical variable Pool, using box plots. This means that we can check the distribution of score given the hotel has a pool or not. Notice that for each category in Pool, a box plot for responses that match that category has been created. Two other things you may notice are that the text is tiny, and that it's hard to tell where the two outliers are - those tiny black dots under the orange box.



**Two quick options**

We can fix both of these issues using sns-dot-set and sns-dot-set-style. First, we increase the font size using font-scale, and then we add gridlines to the plot by specifying whitegrid for the style. The new graphic is easier to read, and we can tell that there are outliers at 2 and 1. It looks like a couple guests who stayed at a hotel with a pool did not like their experience.

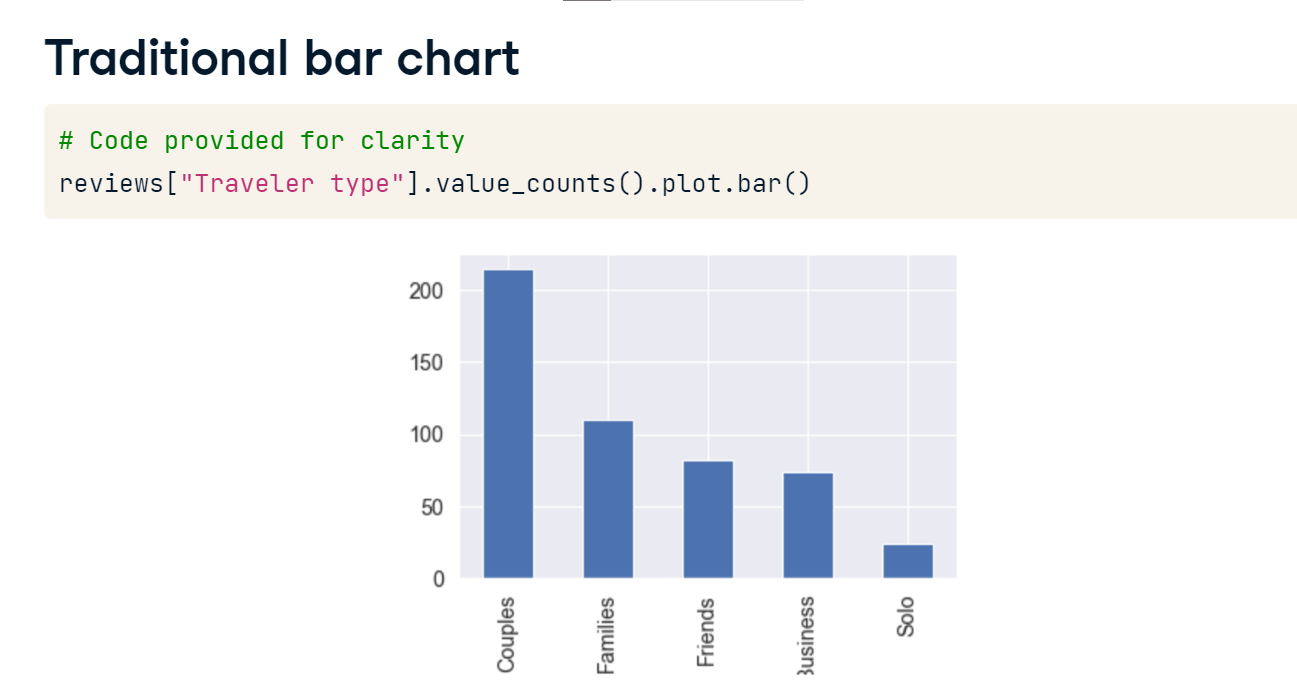


**Seaborn bar plots**

Bar charts may be one of the most common visualizations, but creating them using a categorical Series can be a whole new animal.

**Traditional bar chart**

Let's look at a typical bar chart. We have provided the code for clarity, but this type of visual creation will not be covered in this course. This bar chart shows the number of reviews in the dataset by the traveler type. Couples was the most common, with over 200 reviews, while solo travelers had the least, with fewer than 30 reviews. This is a great simple summarization of this variable. However, Seaborn bar charts serve a different purpose. Our goal is to summarize a numerical variable across the different levels of a categorical variable.



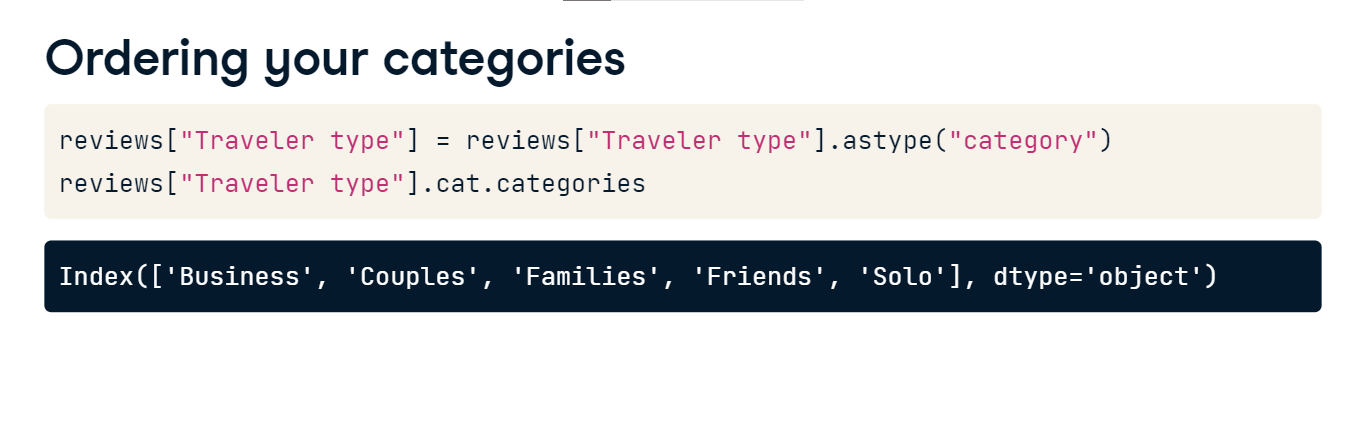
**The syntax**

It's probably no surprise, but the syntax is almost identical to that of creating a boxplot. The only difference is that the kind parameter is set to bar, instead of box. The resulting bar chart looks a little different, and has some funny black lines. The height of each bar is a point estimate for the mean of the data, while the black band represents a confidence interval for that value. Confidence intervals are common in statistics and in this context, the intervals roughly represent a range of values for which we are 95% confident the true mean of the data will fall within. For example, if we looked at a distribution of the Score among those with a traveler type of friends, **the estimated mean of that data would be just above four, and the confidence interval would be quite small. While the solo mean is below four and has a fairly large confidence interval.**



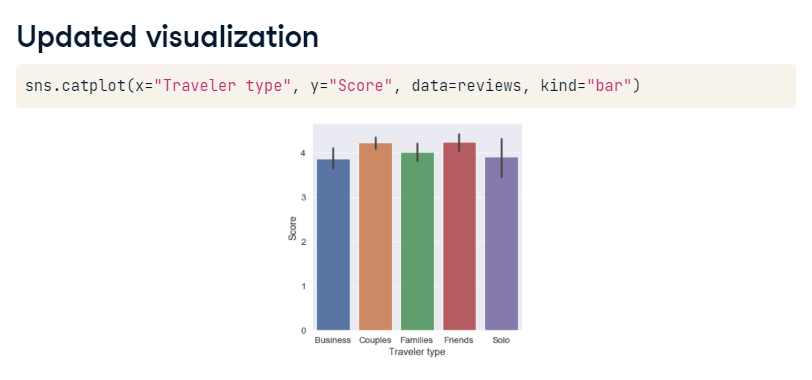
**Ordering your categories**

We previously learned how to create categorical Series, and we can use this to our advantage when creating visualizations. If we set the data type of the traveler type Series to category, the order of the categories displayed in our visualization will be updated. Note that the traveler type categories have been placed in alphabetical order.



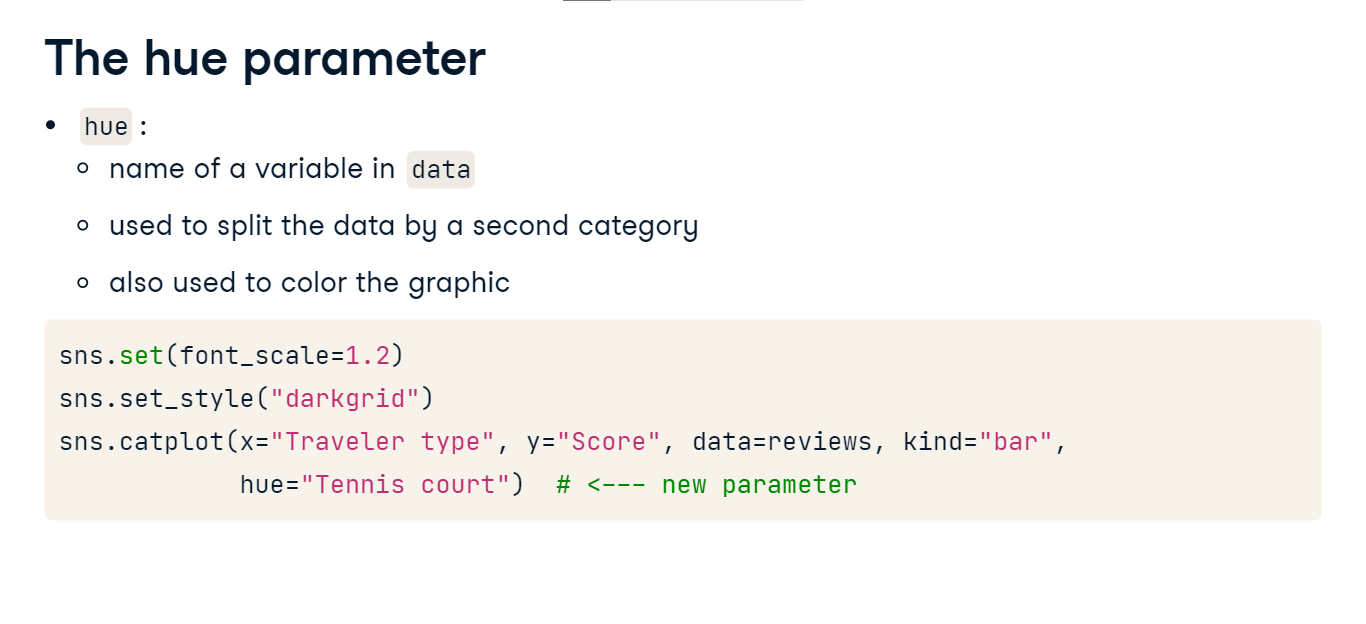
**Updated visualization**

Here is the same visualization, but with traveler type displayed in the order of the categories. Note that the catplot function has a parameter called order, but not all visualization methods have this parameter. It's best practice for us to order our category outside of the catplot function so that all of our visualizations are the same.



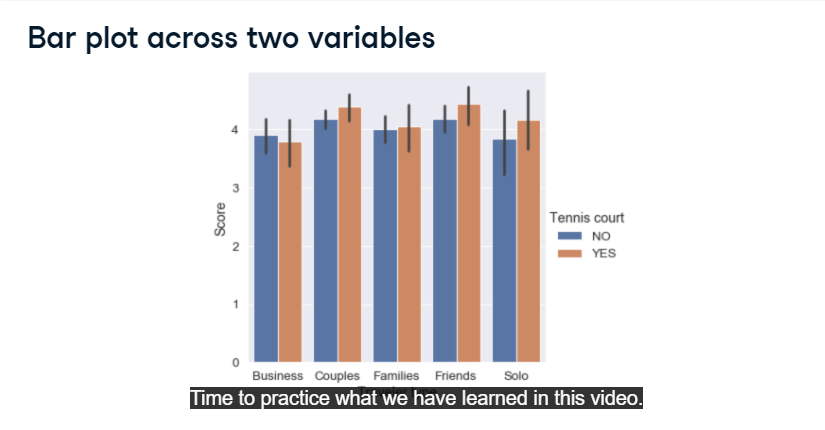
**The hue parameter**

Sometimes visualizing the data across one variable isn't enough, and we want to split the data a second time. The hue parameter can be used for this. Hue is set to a variable in the dataset and is used the split the data a second time. It also tells Seaborn to color the graphic by this variable. In this example we want to look at the Score variable across traveler type and tennis court.



**Bar plot across two variables**

It looks like tennis courts may not persuade business travelers to give a high rating, but the majority of all other traveler type categories gave hotels higher reviews if they had tennis courts. We generally won't use a barplot to compare distributions across categories like we would using a box plot, but this approach gives us a quick way to compare where the estimated mean of the Score is among different categories.

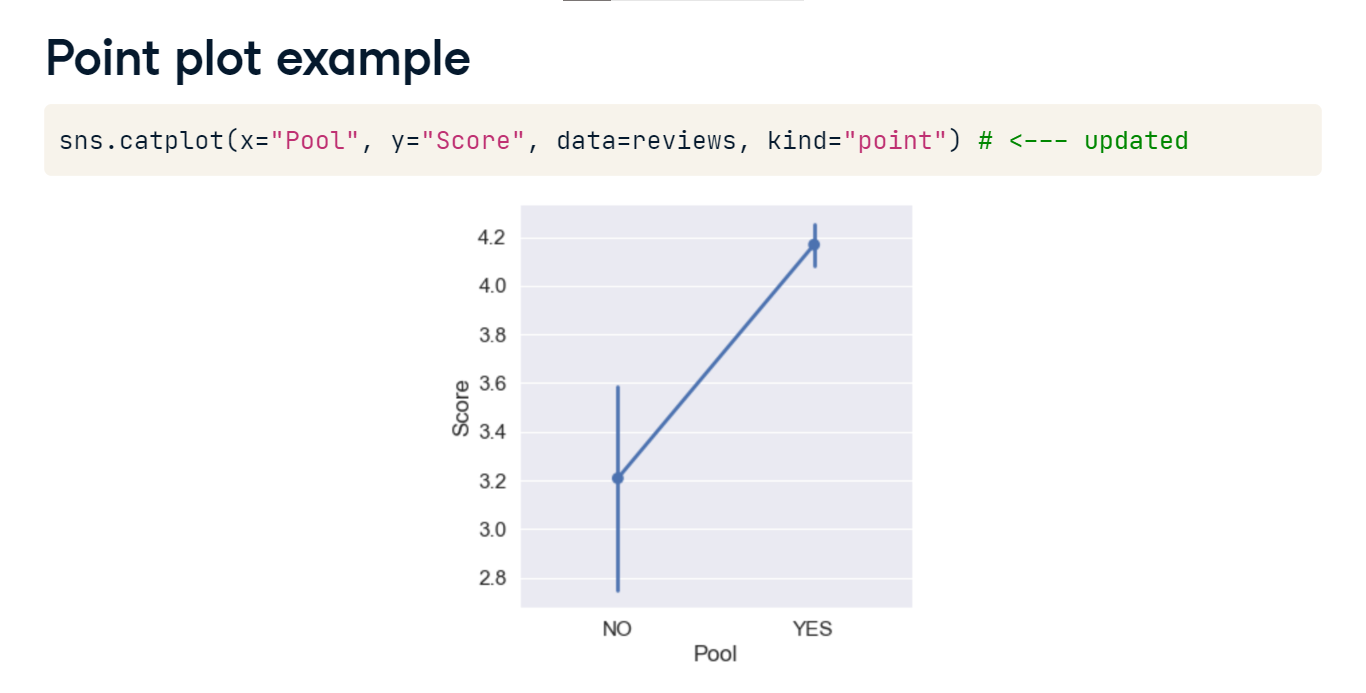


**Point and count plots**

Let's take a look at a couple more catplot visualizations, the point and count plots.

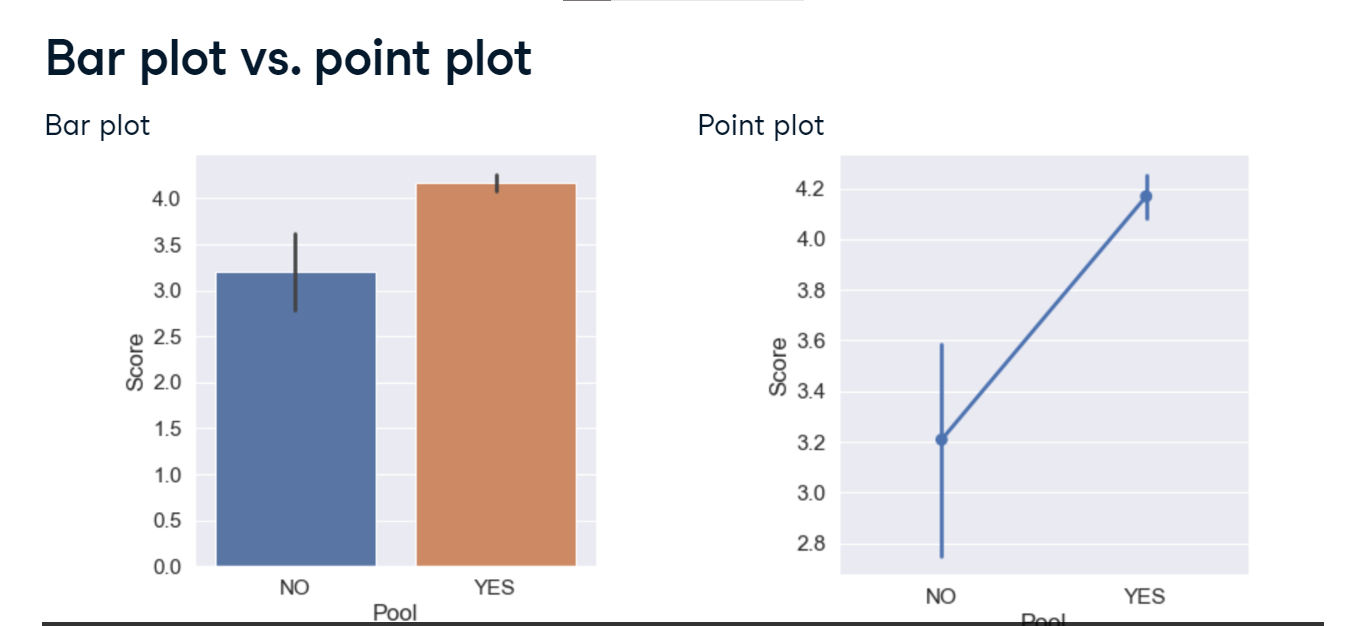
**Point plot example**

Seaborn's catplot is a great starting point for building visualizations based on categorical variables. For this next plot, we again only need to update the kind parameter. Using the reviews dataset one more time, let's look at the review scores given whether the hotel has a pool or not. The point plot shows the mean of the reviewer score just as a bar plot does. However, the diagonal line that connects the means helps users see the difference between them and is often more helpful when there are more categories displayed. The points show means of the score across the categories of pool, while the blue bands that go through the mean represent the confidence intervals. Reviews for hotels without pools hover around 3-point-2, while reviews for hotels with pools hover around 4-point-2.



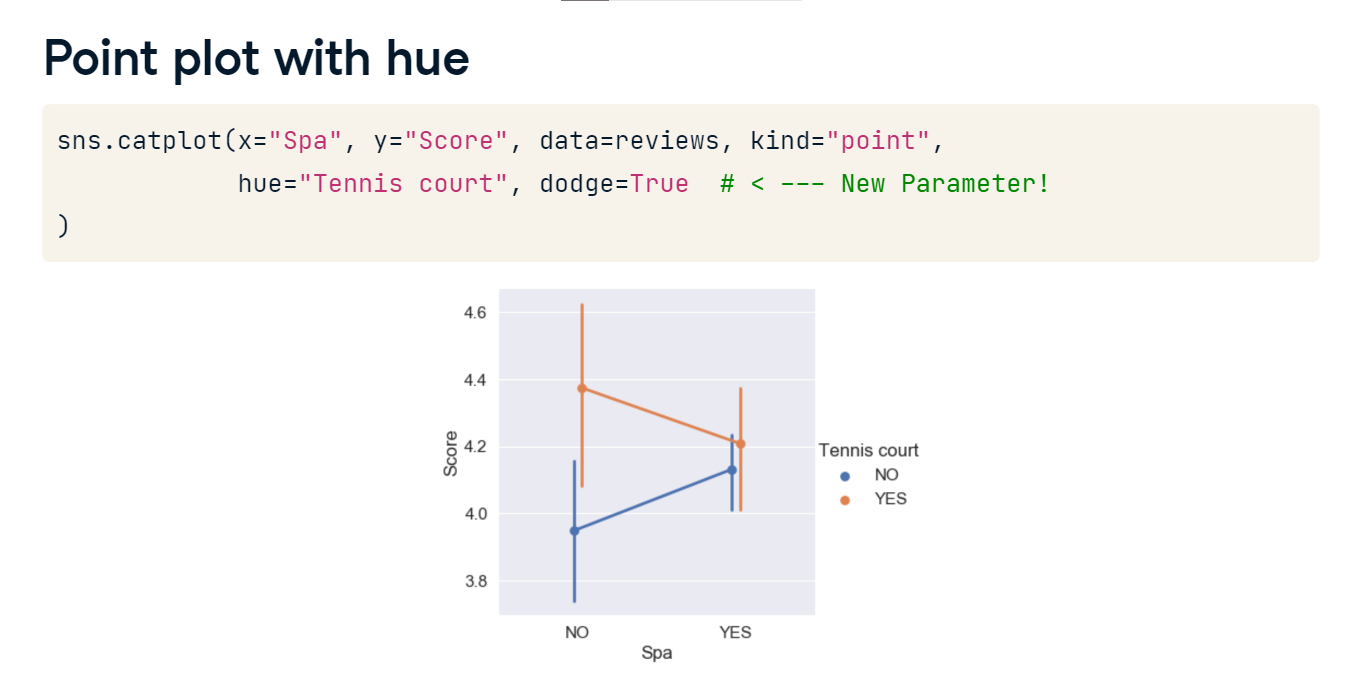
**Bar plot vs. point plot**

When creating visualizations using only one categorical variable, the two visualizations are quite similar. However, the point plot may help users focus on the different values across the categories by adding a connecting line across the points, while the y-axis is changed to better focus on the points. The bar plot, however, may have a more familiar look and does provide color differences even if only one categorical variable is used. The y-axis for this visual defaults to starting at 0. Let's take a look at one more plot that uses the hue parameter.



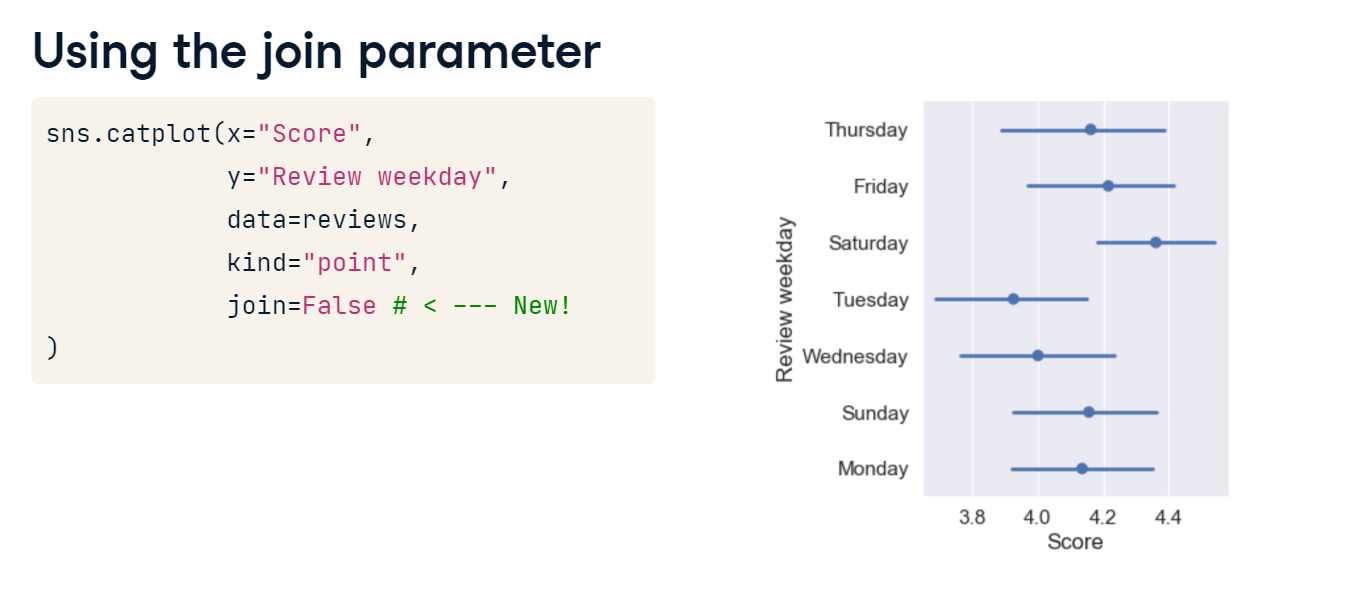
**Point plot with hue**

Notice that the two colored lines go in opposite directions. Having a tennis court or not makes a big difference for hotels with no Spa, but hotels with Spas seem to receive similar reviews regardless of if they have a tennis court. In this visualization, we have set the dodge parameter to true. This offsets the lines so that they don't overlap and makes it easy for users to see where the mean and confidence intervals fall.



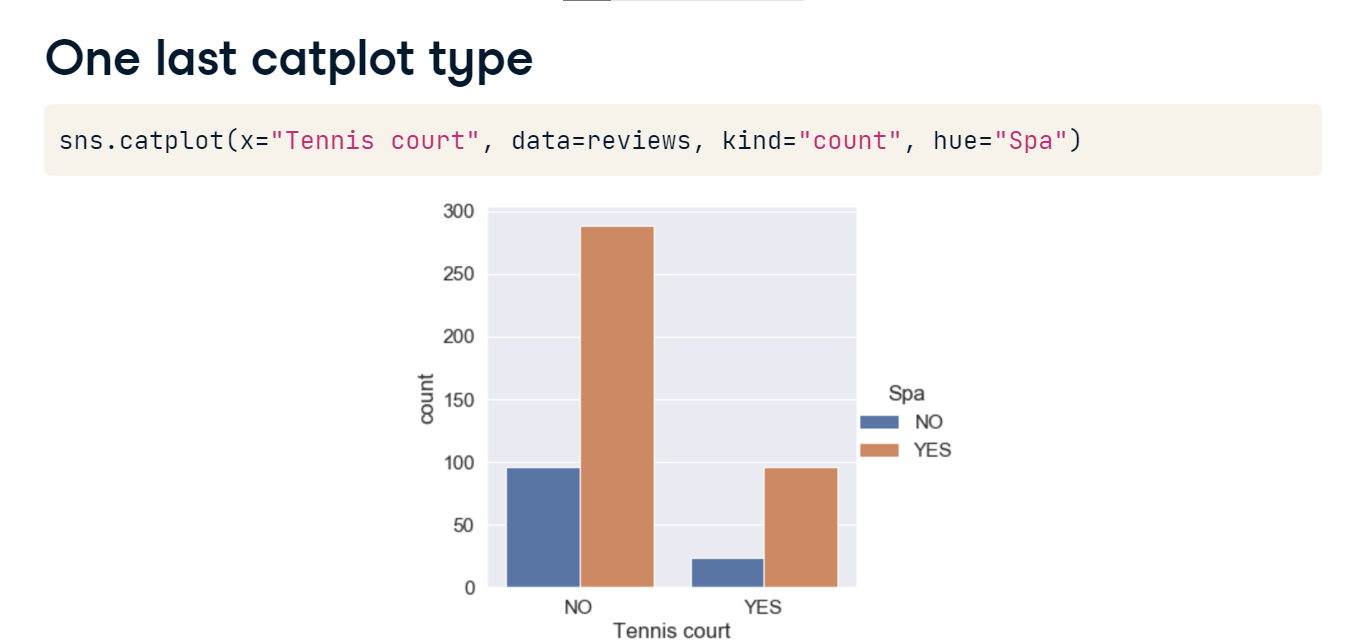
**Using the join parameter**

Sometimes we might not want to join the estimators of each line. We can turn this off using the join parameter. By setting join equal to false, the lines are no longer connected. In this visual, we put the score along the x-axis, and the review weekday along the y-axis. Did you notice the terrible order of the weekdays? That's because we didn't set the order of the review weekday variable! We'll fix that in one of the exercises.



**One last catplot type**

Throughout this course, we have used value-counts to view frequency tables. Value counts are usually shown using a bar plot, but we have already discussed what the seaborn catplot method does when specifying bar. Instead, the catplot method uses the count plot to display frequencies. All of the other visuals we have looked at have been aggregating a numerical variable across a categorical variable. This plot simply counts the number of occurrences of the categorical variables specified in the x or y and hue parameters. It may be a little odd, but the catplot count plot is just a typical bar graph. Try not to get the bar catplot and the count catplot mixed up.

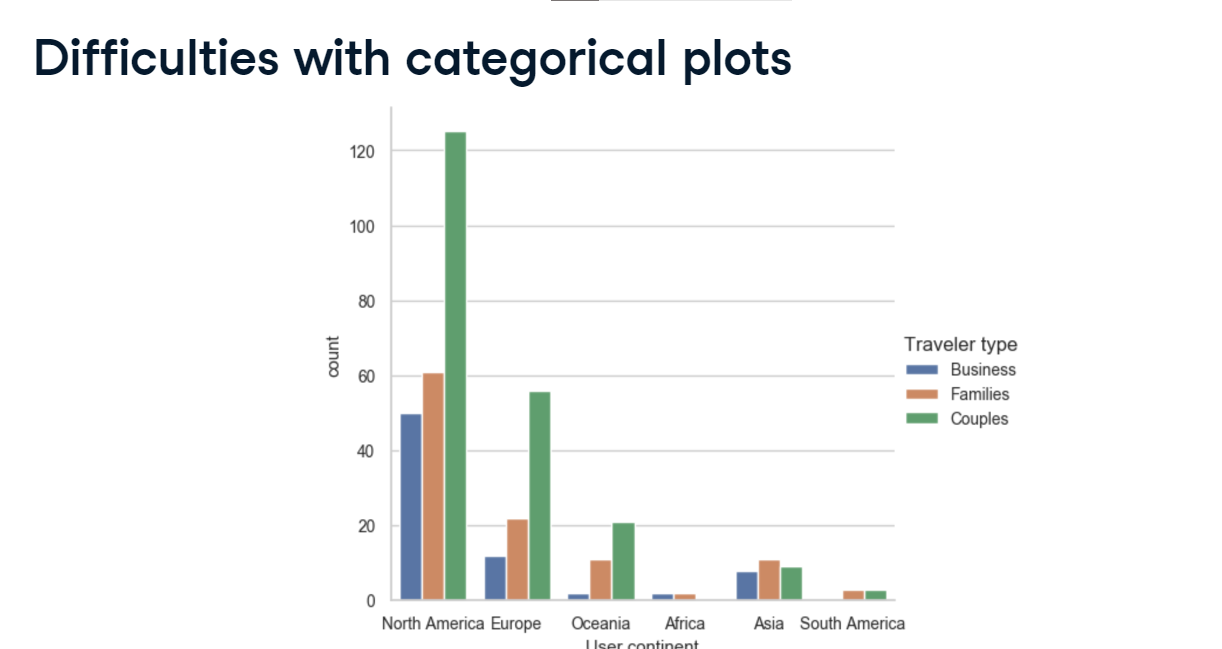


**Additional catplot() options**

To take the visualizations we have worked on a step further, we will update their titles, axes, and colors, as well as create multiple plots at the same time.

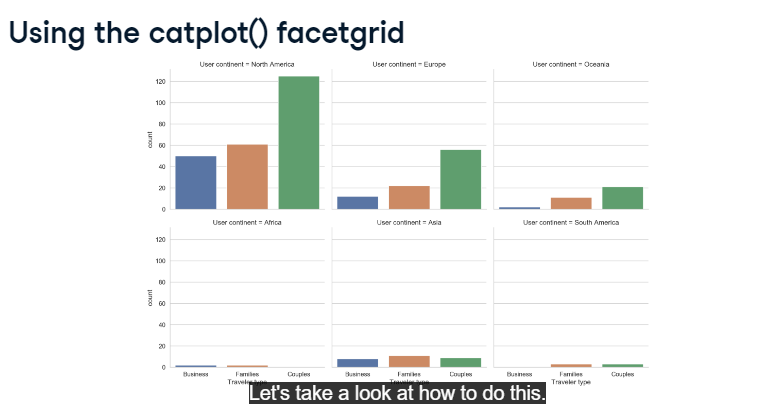
**Difficulties with categorical plots**

Summarizing data by category is a great start, but trying to visualize multiple categories can be difficult. Take this example, which counts the number of hotels in our Las Vegas reviews dataset by traveler type and user continent. Although there is nothing overtly wrong with the graphic, it pushes users to compare the count across continents, when the more interesting information may be the count within the continents. Instead of creating six different plots, one for each continent, we need a better option.



**Using the catplot() facetgrid**

We can create a better visualization by replicating the same graphic several times in a single plot. Before we look at the code, let's explore the output. Here we see the same information for number of hotel reviews by traveler type and user continent, but the emphasis is on the within continent information. Let's take a look at how to do this.



**Using different arguments**

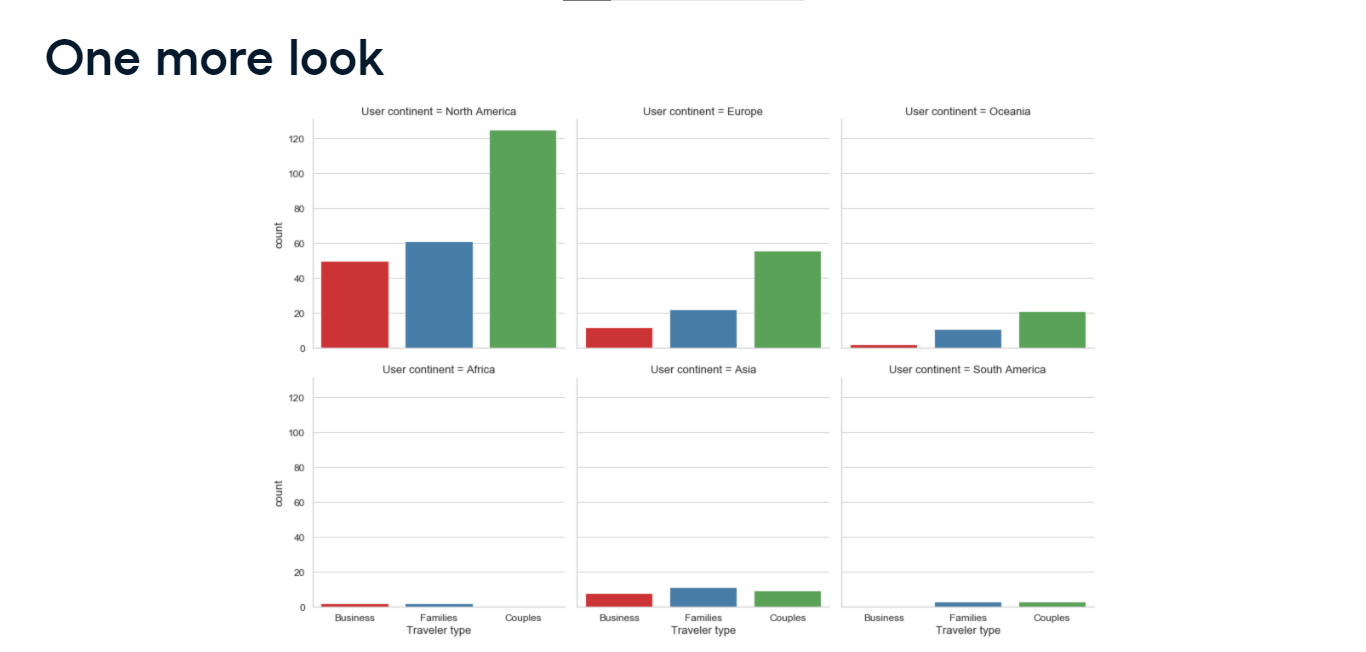
The catplot we just saw was created using the following code. We have specified the x-axis to be based on the categorical column traveler type and have set the type of the graphic to count. We have set the parameter col to be user continent, which tells Seaborn to create a catplot of each category in user continent. We also set col-wrap to be 3, which makes the visualization go to the next line, or wrap itself, after every 3rd graphic. And finally, we went ahead and changed the colors of the graph using Seaborn's color-palette function. Here we have used Set1 as the color palette, although Seaborn has a ton of different options that can be found using the link on your screen. Some of the most commonly used are set2, tab10, and paired.

1. 1 <http://seaborn.pydata.org/tutorial/color_palettes.html>



**One more look**

Let's take a look at this graphic one more time. Again, notice the emphasis is on the counts within each category of user continent and we have six clean graphics, all in one plot. The text may be a little small, but we already know how to fix that issue!



**Updating plots**

To really finish off creating our graphics, we need to learn how to adjust some basic options. To do this, we need some help from matplotlib. We will start by saving our seaborn graphic as an object, ax, and then using matplotlib to update specific items of the object. We can add a title by using fig-dot-suptitle on our saved object. We can add axis labels using set-axis-labels. The x axis label comes first, followed by the y axis label. In some cases, the title is displayed over top the graphic. We can fix this by setting the top of the actual plot to be at 90% of the full graphic. This is achieved using matplotlib's subplots-adjust function. Here is the finalized code. We have saved our graphic as an object, set a title and axis labels, and also fixed the height of the plot, so the title is displayed properly.



**Finished product**

And here is the final graphic. We have broken down the number of reviews by the user continent and reviewer type. Although a small sample size, Asia - the bottom middle graphic - had more reviews from traveling families than for couples. This is quite different than continents such as North America, Europe, or Oceania.

le