**Introduction to relational plots and subplots**

Many questions in data science are centered around describing the relationship between two quantitative variables. Seaborn calls plots that visualize this relationship "relational plots".

**Questions about quantitative variables**

So far we've seen several examples of questions about the relationship between two quantitative variables, and we answered them with scatter plots. These examples include: "do taller people tend to weigh more?"

**Questions about quantitative variables**

"what's the relationship between the number of absences a student has and their final grade?"

**Questions about quantitative variables**

and "how does a country's GDP relate to the percent of the population that can read and write?" Because they look at the relationship between two quantitative variables, these scatter plots are all considered relational plots.

**Visualizing subgroups**

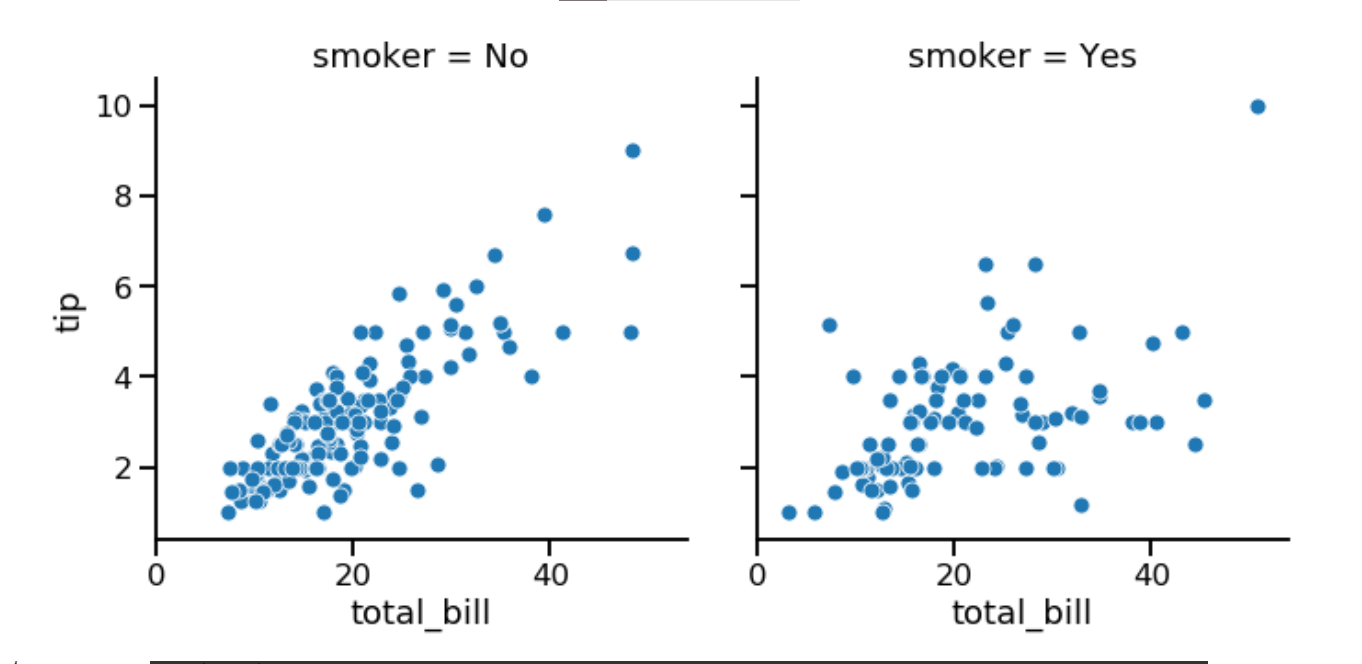
While looking at a relationship between two variables at a high level is often informative, sometimes we suspect that the relationship may be different within certain subgroups. In the last chapter, we started to look at subgroups by using the "hue" parameter to visualize each subgroup using a different color on the same plot.

1. 1 Waskom, M. L. (2021). seaborn: statistical data visualization. https://seaborn.pydata.org/

**Visualizing subgroups**

In this lesson, we'll try out a different method: creating a separate plot per subgroup.

1. 1 Waskom, M. L. (2021). seaborn: statistical data visualization. https://seaborn.pydata.org/

****

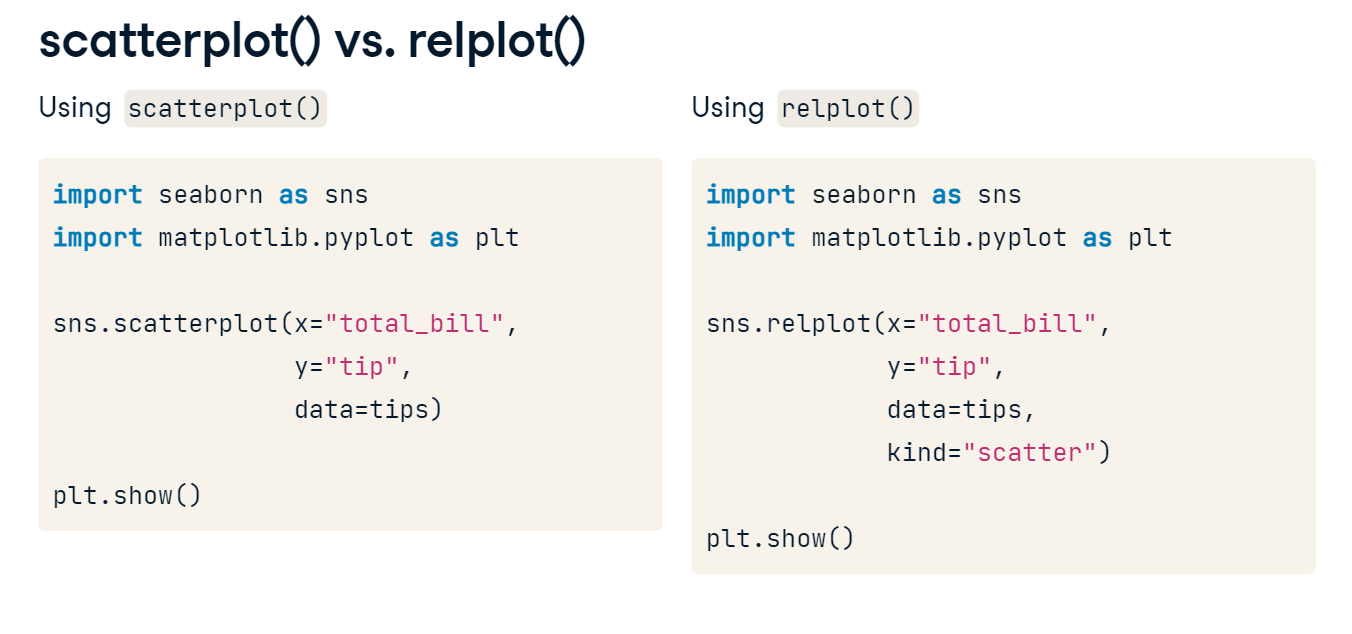
**Introducing relplot()**

To do this, we're going to introduce a new Seaborn function: "relplot()". "relplot()" stands for "relational plot" and enables you to visualize the relationship between two quantitative variables using either scatter plots or line plots. You've already seen scatter plots, and you'll learn about line plots later in this chapter. Using "relplot()" gives us a big advantage: the ability to create subplots in a single figure. Because of this advantage, we'll be using "relplot()" instead of "scatterplot()" for the rest of the course.

**scatterplot() vs. relplot()**

Let's return to our scatter plot of total bill versus tip amount from the tips dataset. On the left, we see how to create a scatter plot with the "scatterplot" function. To make it with "relplot()" instead, we change the function name to "relplot()" and use the "kind" parameter to specify what kind of relational plot to use - scatter plot or line plot. In this case, we'll set kind equal to the word "scatter".

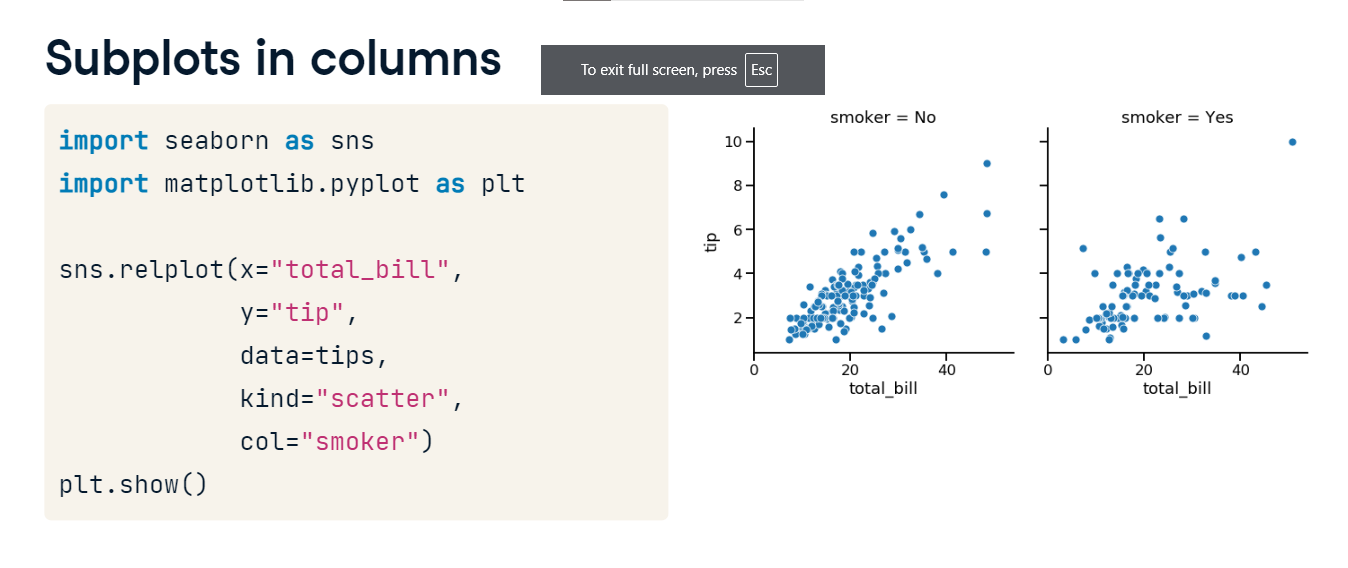
1. 1 Waskom, M. L. (2021). seaborn: statistical data visualization. https://seaborn.pydata.org/

****

**Subplots in columns**

By setting "col" equal to "smoker", we get a separate scatter plot for smokers and non-smokers, arranged horizontally in columns.

1. 1 Waskom, M. L. (2021). seaborn: statistical data visualization. https://seaborn.pydata.org/

****

**Subplots in rows**

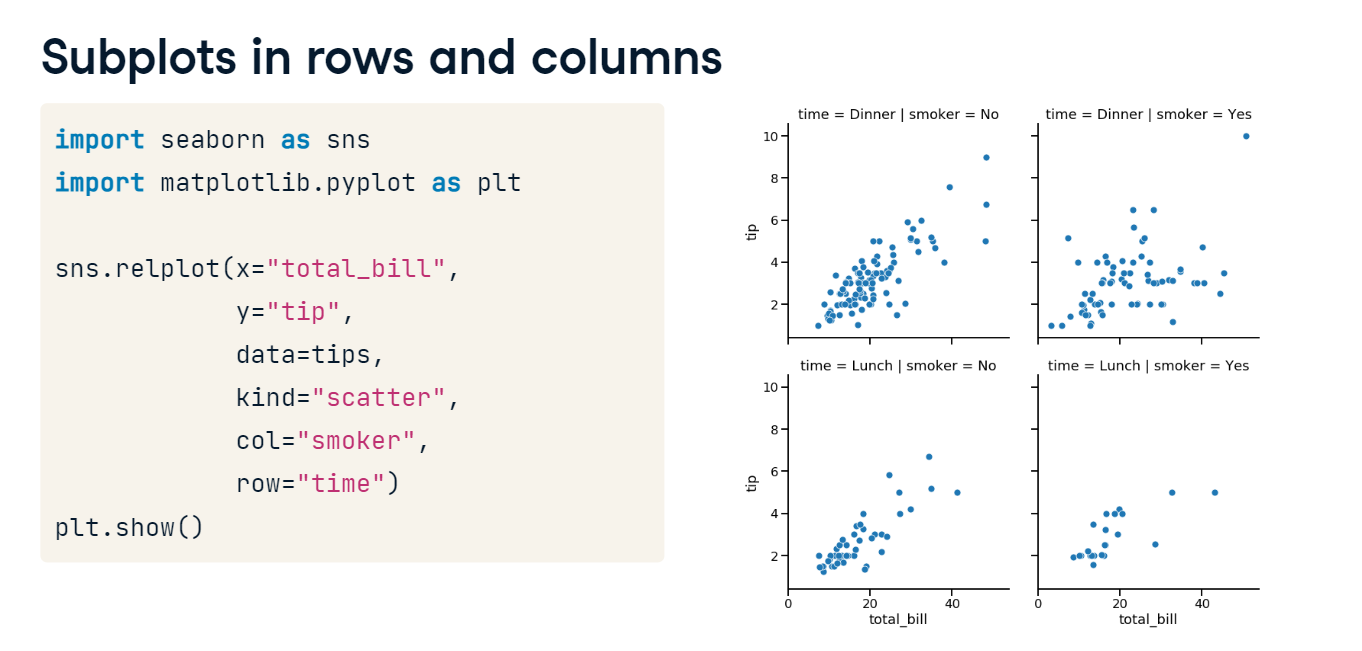
If you want to arrange these vertically in rows instead, you can use the "row" parameter instead of "col".

1. 1 Waskom, M. L. (2021). seaborn: statistical data visualization. https://seaborn.pydata.org/

**Subplots in rows and columns**

It is possible to use both "col" and "row" at the same time. Here, we set "col" equal to smoking status and "row" equal to the time of day (lunch or dinner). Now we have a subplot for each combination of these two categorical variables.

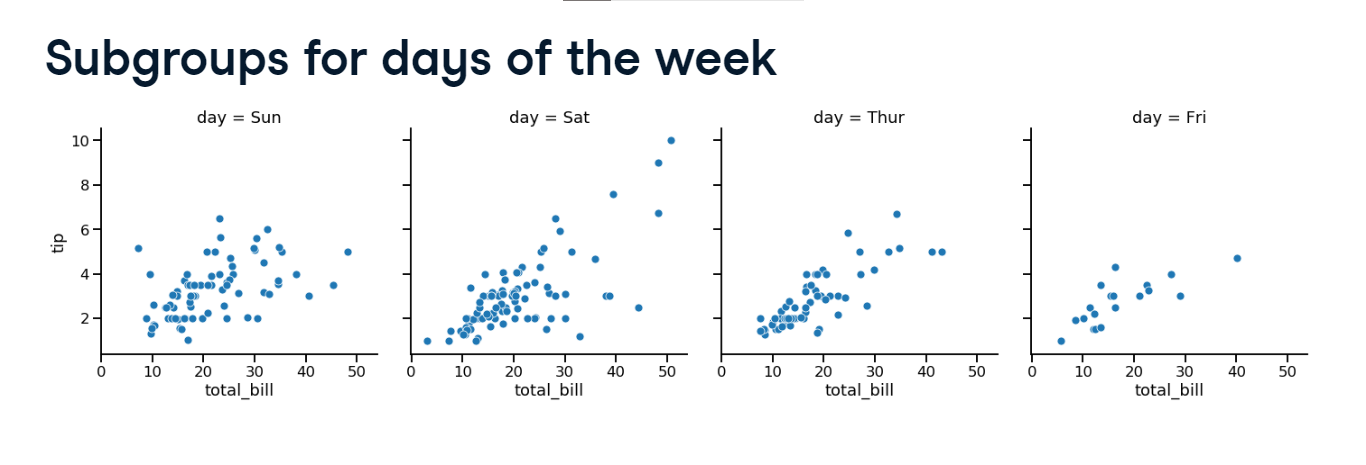
1. 1 Waskom, M. L. (2021). seaborn: statistical data visualization. https://seaborn.pydata.org/

****

**Subgroups for days of the week**

As another example, let's look at subgroups based on day of the week. There are four subplots here, which can be a lot to show in a single row. To address this, you can use the "col\_wrap" parameter to specify how many subplots you want per row.

1. 1 Waskom, M. L. (2021). seaborn: statistical data visualization. <https://seaborn.pydata.org/>



**Wrapping columns**

Here, we set "col\_wrap" equal to two plots per row.

1. 1 Waskom, M. L. (2021). seaborn: statistical data visualization. https://seaborn.pydata.org/

**Ordering columns**

We can also change the order of the subplots by using the "col\_order" and "row\_order" parameters and giving it a list of ordered values.

1. 1 Waskom, M. L. (2021). seaborn: statistical data visualization. https://seaborn.pydata.org/

