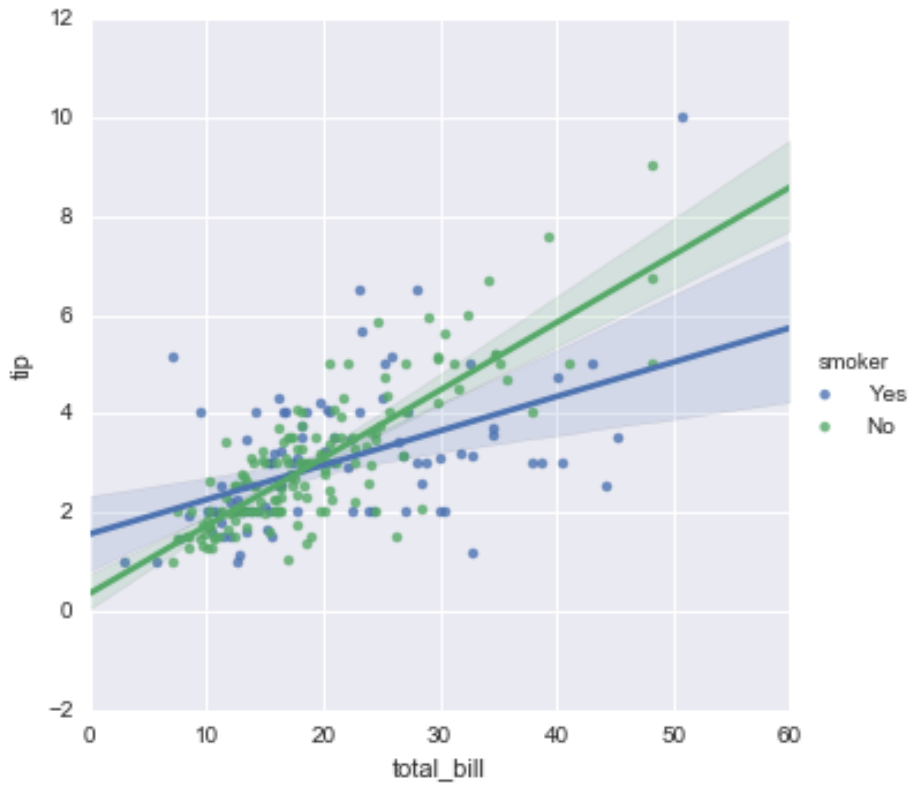
Conditioning on other variables

- The plots above show many ways to explore the relationship between a pair of variables.
- Often, however, a more interesting question is how does the relationship between these two variables change as a function of a third variable?
- This is where the difference between `regplot()` and `lmplot()` appears.
- While `regplot()` always shows a single relationsihp, `lmplot()` combines `regplot()` with FacetGrid to provide an easy interface to show a linear regression on faceted plots that allow you to explore interactions with up to three additional categorical variables.

Seaborn Workshop

The best way to separate out a relationship is to plot both levels on the same axes and to use color to distinguish them:

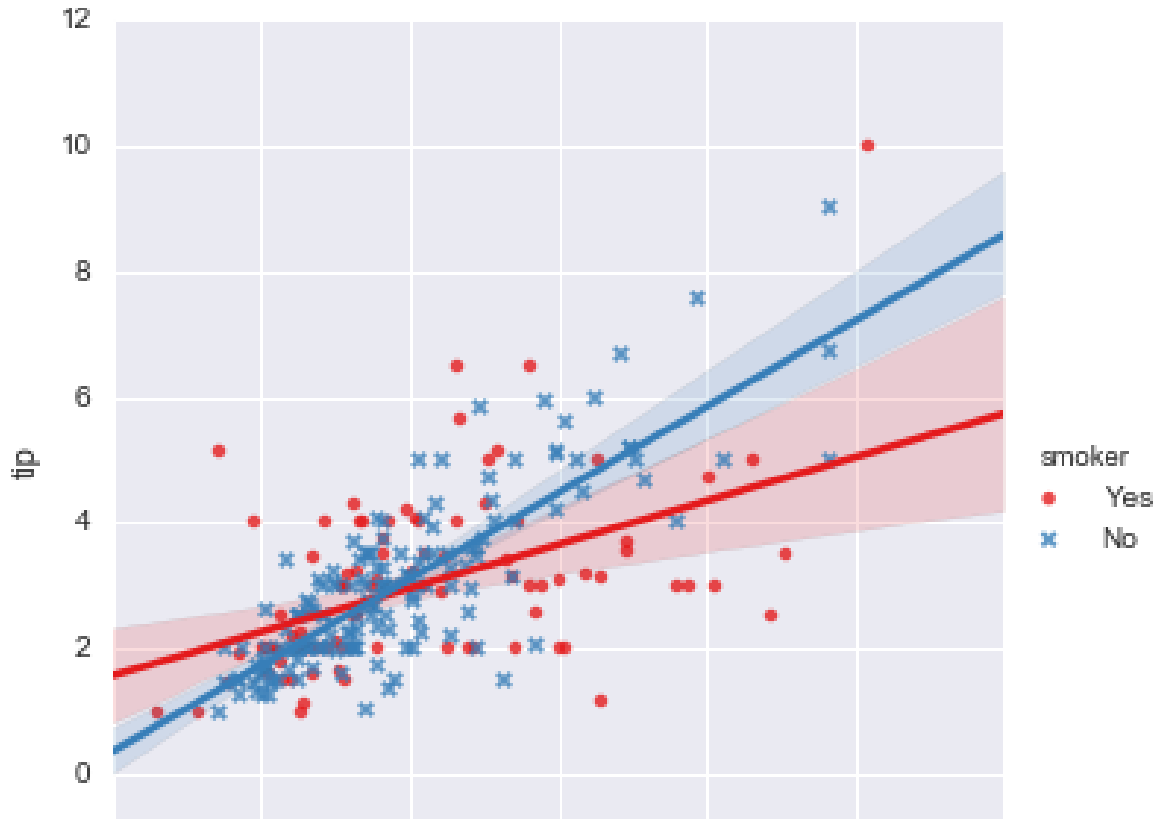
```
sns.lmplot(x="total_bill", y="tip",
           hue="smoker", data=tips);
```



Seaborn Workshop

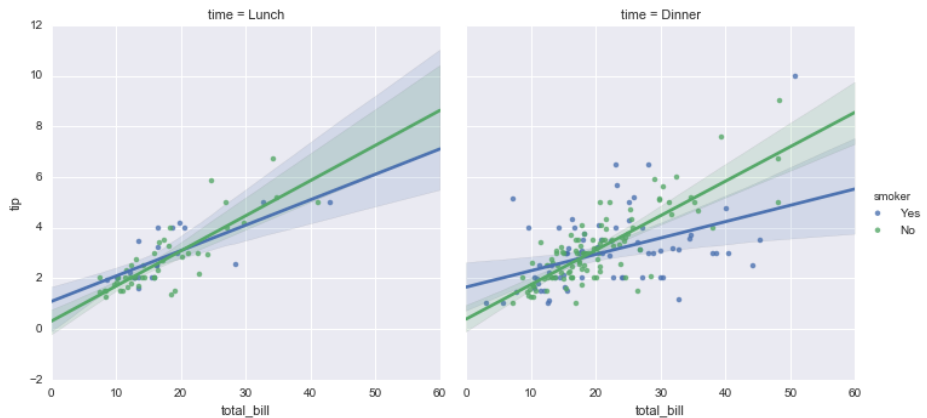
In addition to color, its possible to use different scatterplot markers to make plots the reproduce to black and white better. You also have full control over the colors used:

```
sns.lmplot(x="total_bill", y="tip", hue="smoker", data=
           markers=["o", "x"], palette="Set1");
```



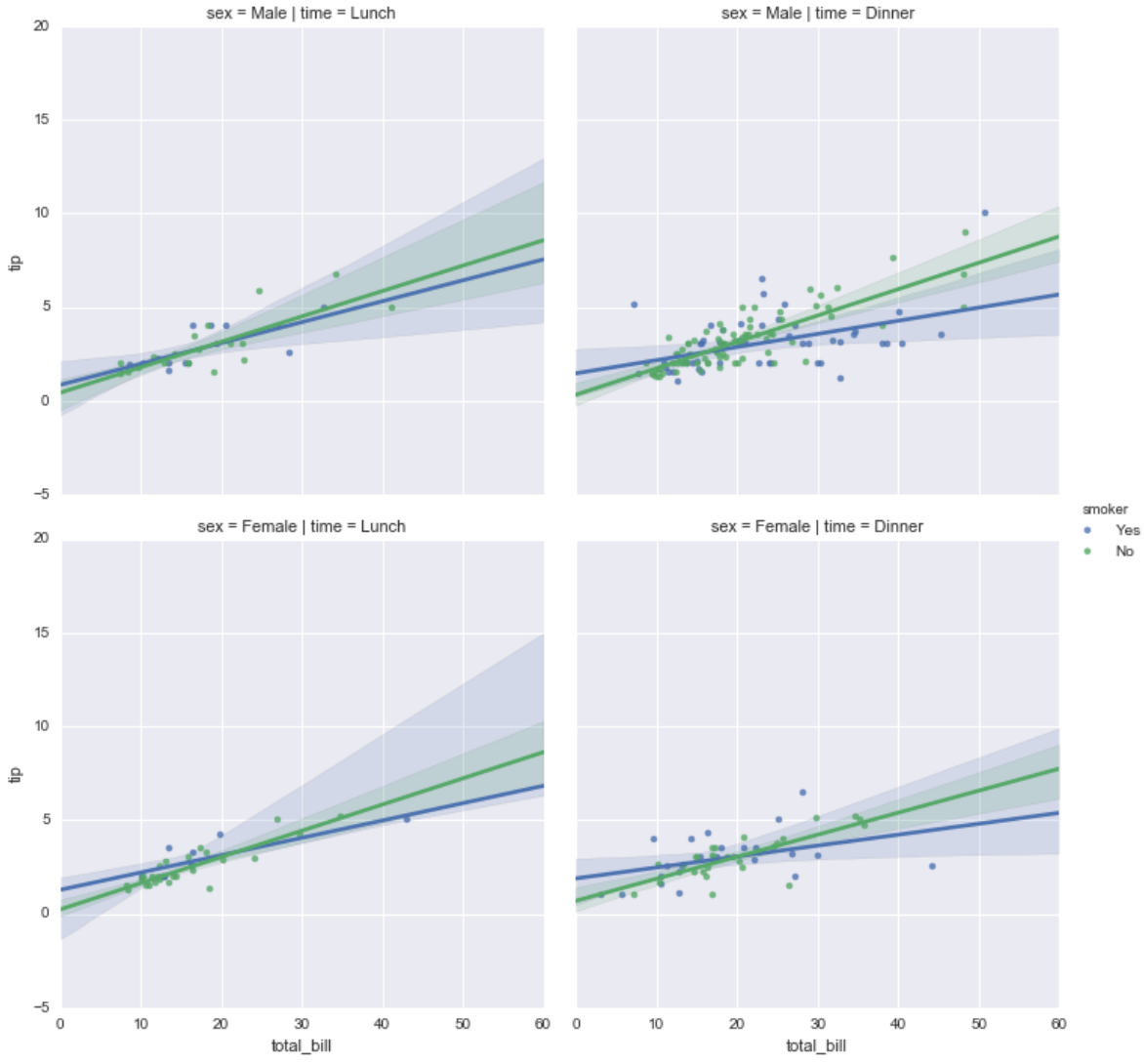
To add another variable, you can draw multiple facets which each level of the variable appearing in the rows or columns of the grid:

```
sns.lmplot(x="total_bill", y="tip",
           hue="smoker", col="time",
           data=tips);
```



Seaborn Workshop

```
sns.lmplot(x="total_bill", y="tip", hue="smoker",
           col="time", row="sex", data=tips);
```

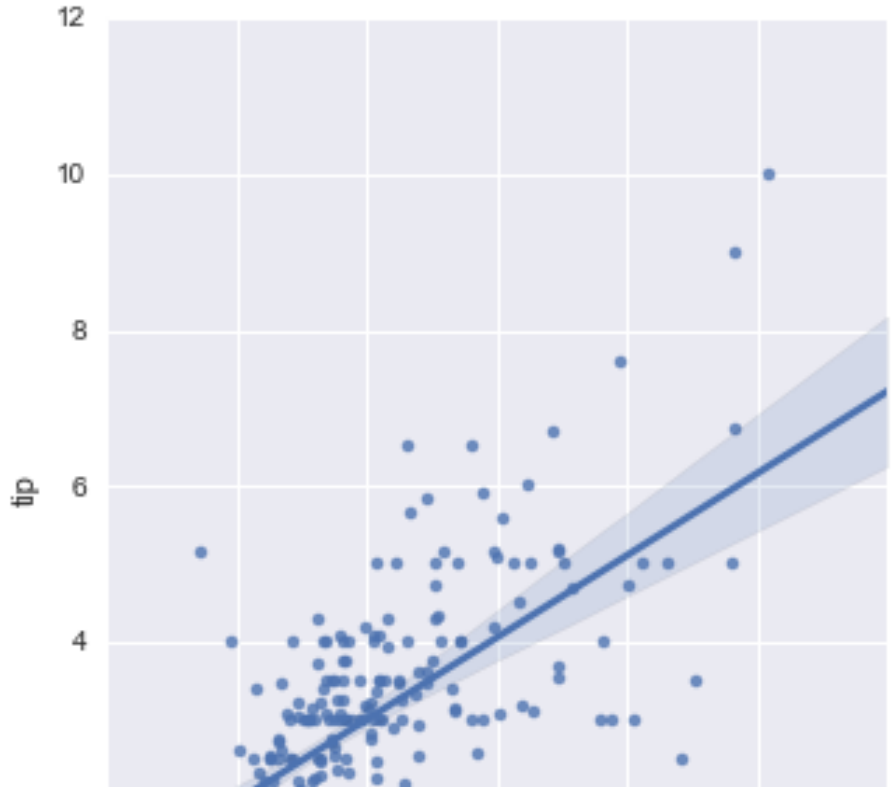


Seaborn Workshop

- Before we noted that the default plots made by `regplot()` and `lmplot()` look the same but on axes that have a different size and shape. This is because func:regplot is an axes-level function draws onto a specific axes.
- This means that you can make mutli-panel figures yourself and control exactly where the the regression plot goes.
- If no axes is provided, it simply uses the *currently active* axes, which is why the default plot has the same size and shape as most other matplotlib functions. To control the size, you need to create a figure object yourself.

Seaborn Workshop

```
f, ax = plt.subplots(figsize=(5, 6))
sns.regplot(x="total_bill", y="tip",
            data=tips, ax=ax);
```

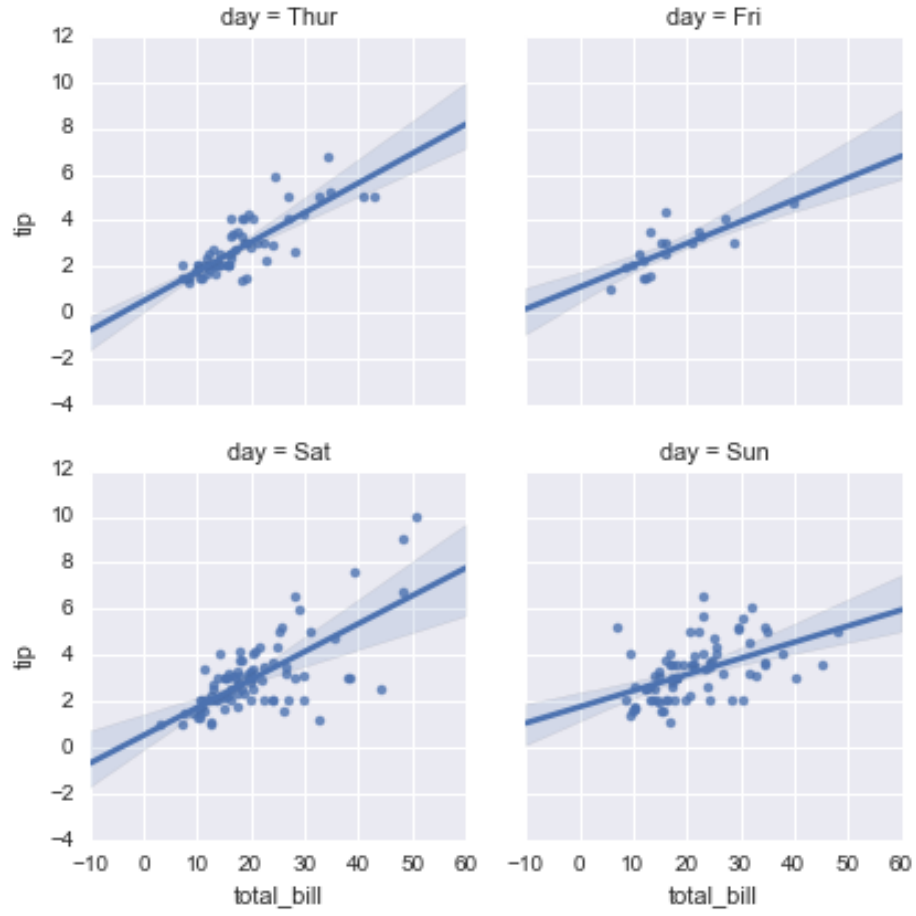


Seaborn Workshop

- In contrast, the size and shape of the `lmplot()` figure is controlled through the `FacetGrid` interface using the size and aspect parameters, which apply to each facet in the plot, not to the overall figure itself:

Seaborn Workshop

```
sns.lmplot(x="total_bill", y="tip", col="day", data=tip
col_wrap=2, size=3);
```



Seaborn Workshop

```
sns.lmplot(x="total_bill", y="tip", col="day", data=tip
aspect=.5);
```