In [1]:
```
# The normal imports
import numpy as np
from numpy.random import randn
import pandas as pd

# Import the stats library from numpy
from scipy import stats

# These are the plotting modules adn libraries we'll use:
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns

# Command so that plots appear in the iPython Notebook
%matplotlib inline
```

In [2]:
```
# Now we'll learn how ot visualize multiple regression with lmplot()

# Luckily, Seaborn comes with an example dataset to use as a pandas DataFrame
tips = sns.load_dataset("tips")
```

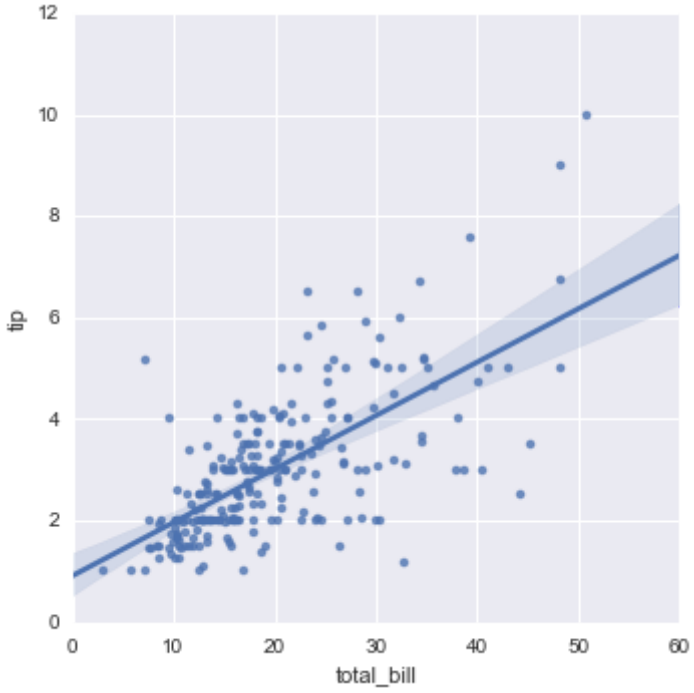
In [4]:
```
# Preview
tips.head()
```

Out[4]:

	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

In [5]:
```
# Let's use lmplot() to plot the total bill versus tips
sns.lmplot("total_bill","tip",tips)
```

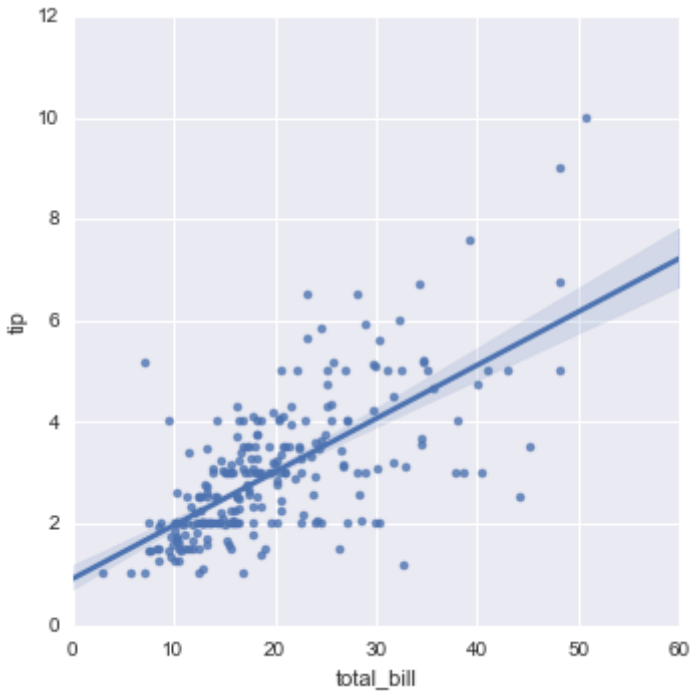
Out[5]: <seaborn.axisgrid.FacetGrid at 0x192a9c18>



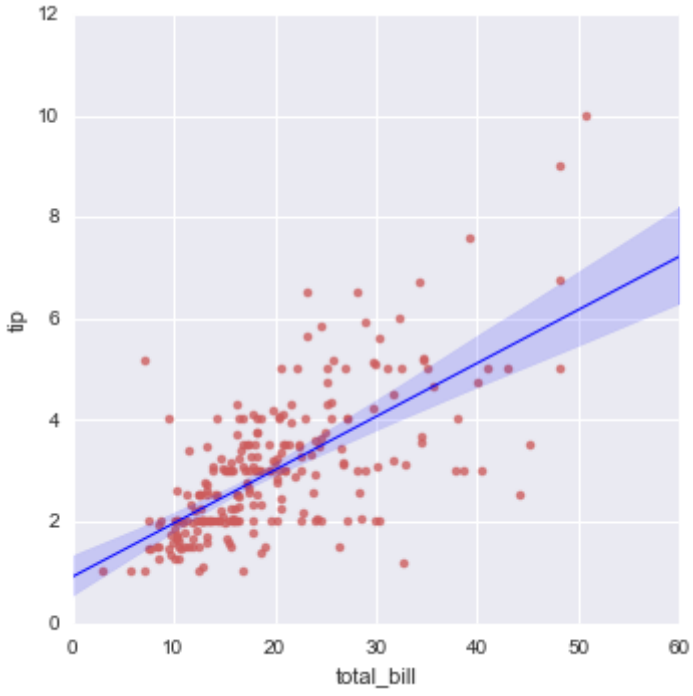
In [6]:
```
# First we can see a scatter plot of all the points, tip vs total_bill
# Then we see a linear regression line, which is an estimateed linear fit model t
```

In [8]:
```
# WE can also specify teh confidence interval to use for the linear fit

sns.lmplot("total_bill","tip",tips,ci=75) # 68% ci
```

Out[8]: <seaborn.axisgrid.FacetGrid at 0x19758160>

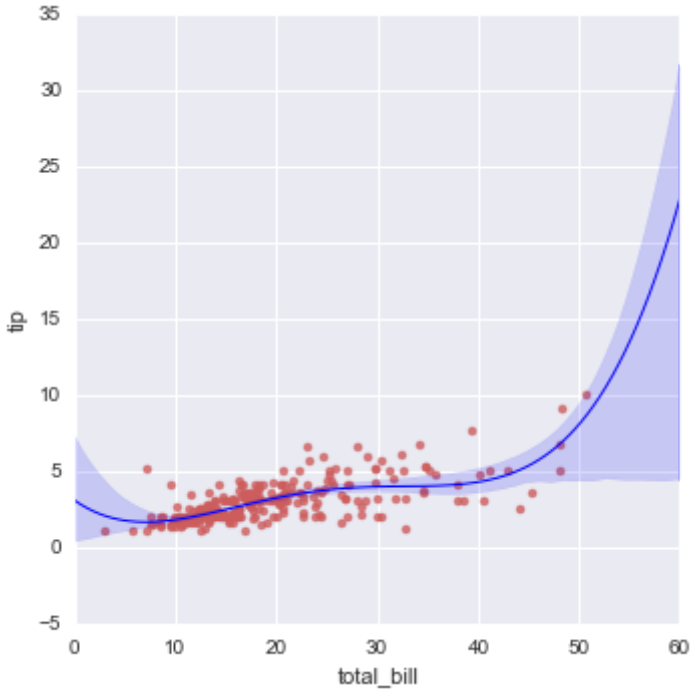


In [13]:
```
# Just like before, we can use dictionaries to edit individual parts of the plot

sns.lmplot("total_bill", "tip", tips,
           scatter_kws={"marker": "o", "color": "indianred"},
           line_kws={"linewidth": 1, "color": "blue"});
```



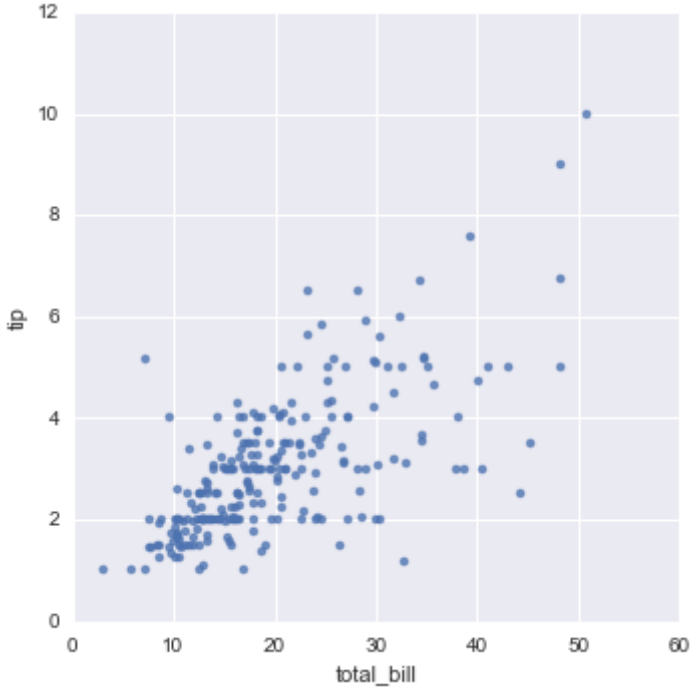
In [22]:
```
# WE can also check out higher-order trends
sns.lmplot("total_bill", "tip", tips,order=4,
           scatter_kws={"marker": "o", "color": "indianred"},
           line_kws={"linewidth": 1, "color": "blue"})
```

Out[22]: <seaborn.axisgrid.FacetGrid at 0x1f7c42e8>

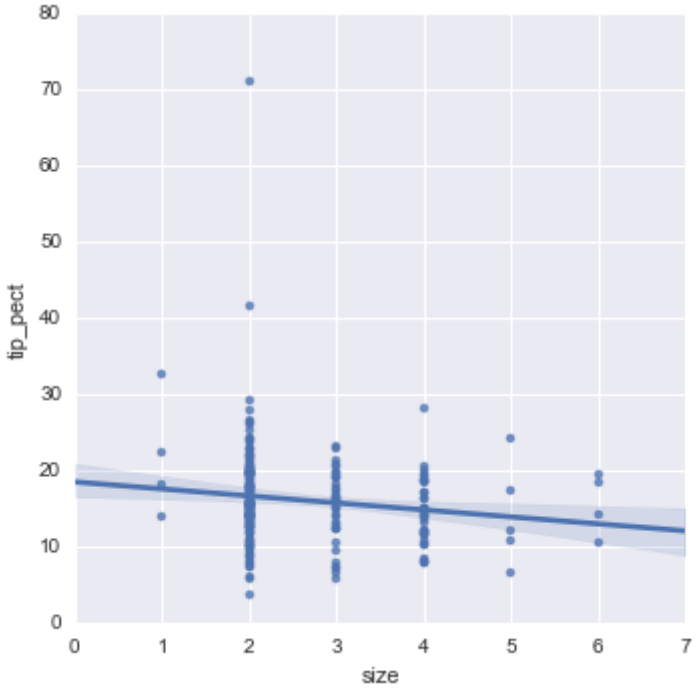


In [15]:
```
# We can also not fit a regression if desired
sns.lmplot("total_bill", "tip", tips,fit_reg=False)
```

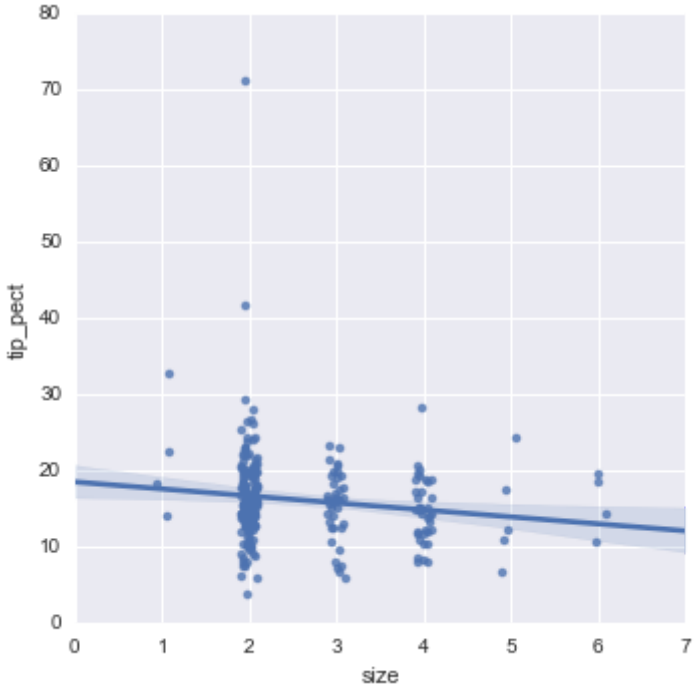
Out[15]: <seaborn.axisgrid.FacetGrid at 0x1c4a44a8>



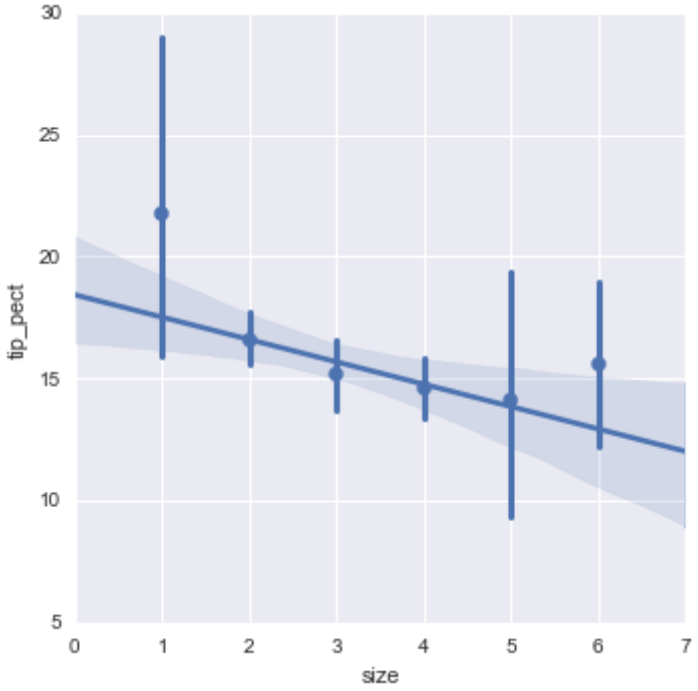
In [23]:
```
# lmplot() also works on discrete variables, such as the percentage of the tip

# Create a new column for tip percentage
tips["tip_pect"]=100*(tips['tip']/tips['total_bill'])

#plot
sns.lmplot("size", "tip_pect", tips);
```



In [26]:
```
# We can also add jitter to this

#Info link
url = "http://en.wikipedia.org/wiki/Jitter"

#plot
sns.lmplot("size", "tip_pect", tips,x_jitter=.1);
```



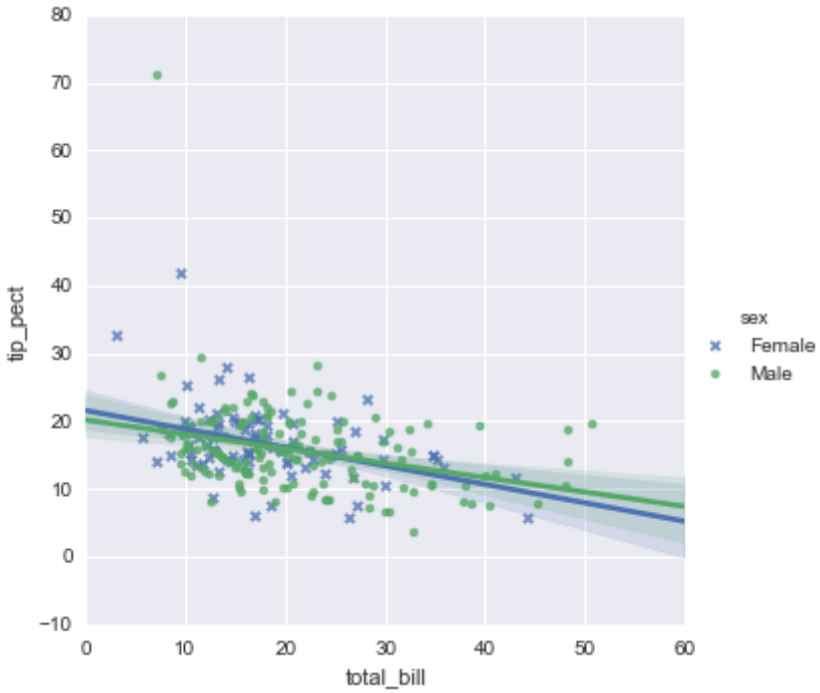
In [27]:
```
# We can also estimate the tendency of each bin (size of party in this case)
sns.lmplot("size", "tip_pect", tips, x_estimator=np.mean);
```



In [28]:
```
# Interesting, looks like there is more variance for party sizes of 1 then 2-4
```

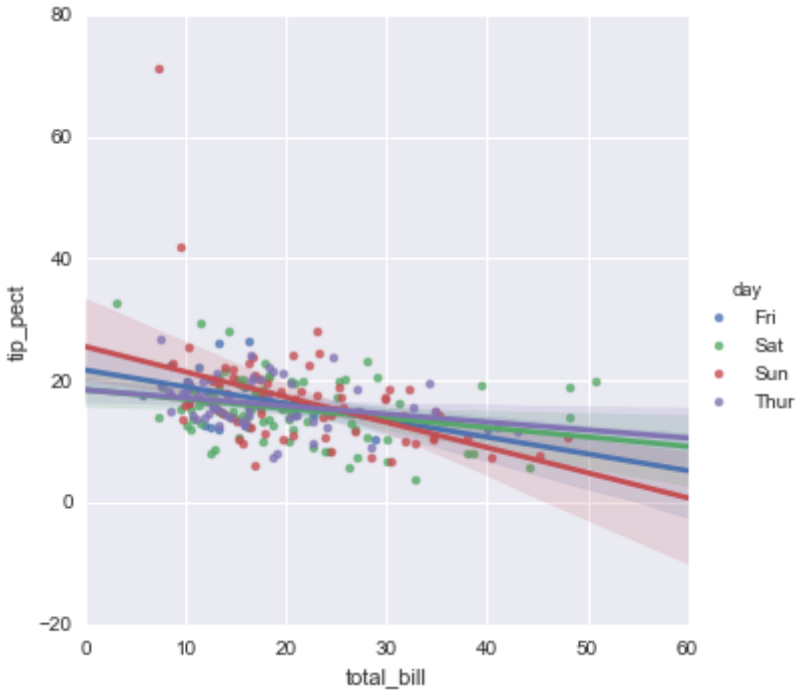
In [31]:
```
# We can use the hue facet to automatically define subsets along a column

# Plot, note the markers argument
sns.lmplot("total_bill", "tip_pect", tips, hue="sex",markers=["x","o"])
```

Out[31]: <seaborn.axisgrid.FacetGrid at 0x2044b390>

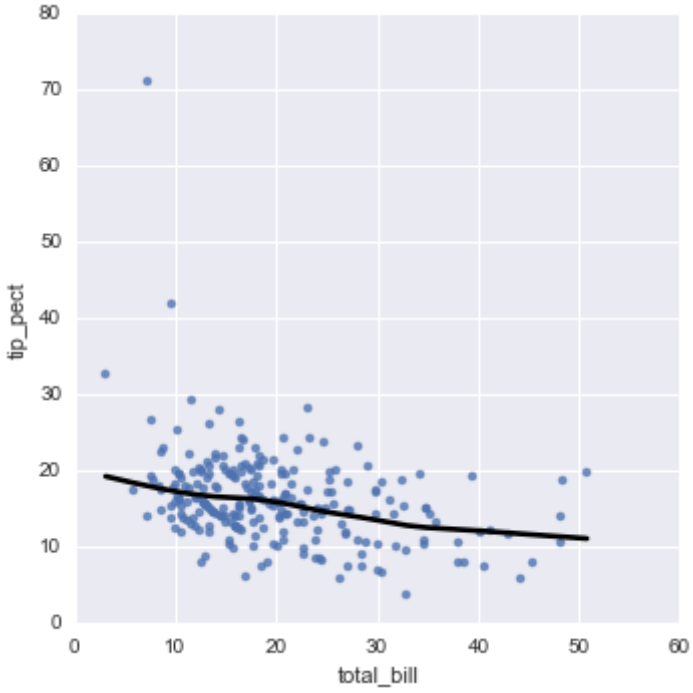


In [32]:
```
# Does day make a difference?
sns.lmplot("total_bill", "tip_pect", tips, hue="day")
```

Out[32]: <seaborn.axisgrid.FacetGrid at 0x2042bd30>

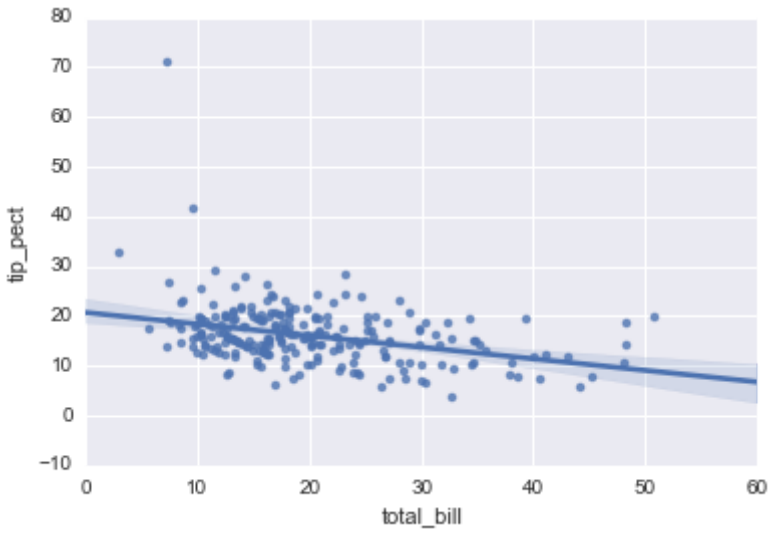


In [36]:
```
# Finally it should be noted that Seabron supports LOESS model fitting
url = 'http://en.wikipedia.org/wiki/Local_regression'

sns.lmplot("total_bill", "tip_pect", tips, lowess=True, line_kws={"color": 'black
```



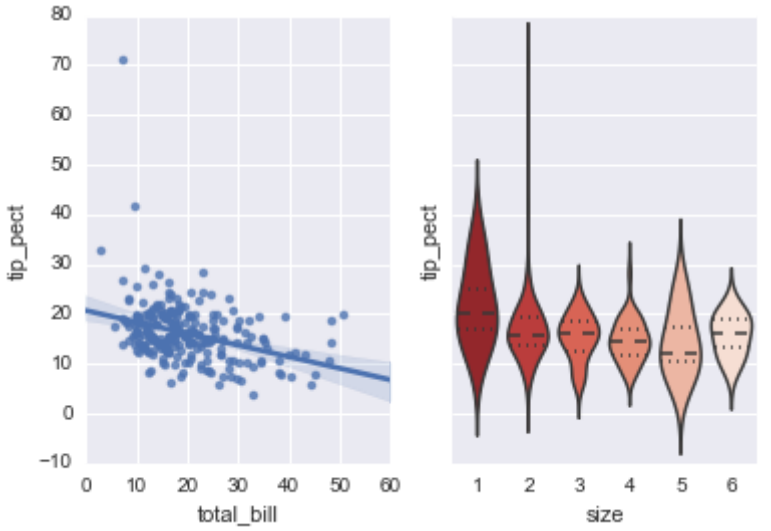
In [37]:
```
# The lmplot() we've been using is actually using a lower-level function, regplot

sns.regplot("total_bill","tip_pect",tips)
```

Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x1fdad710>

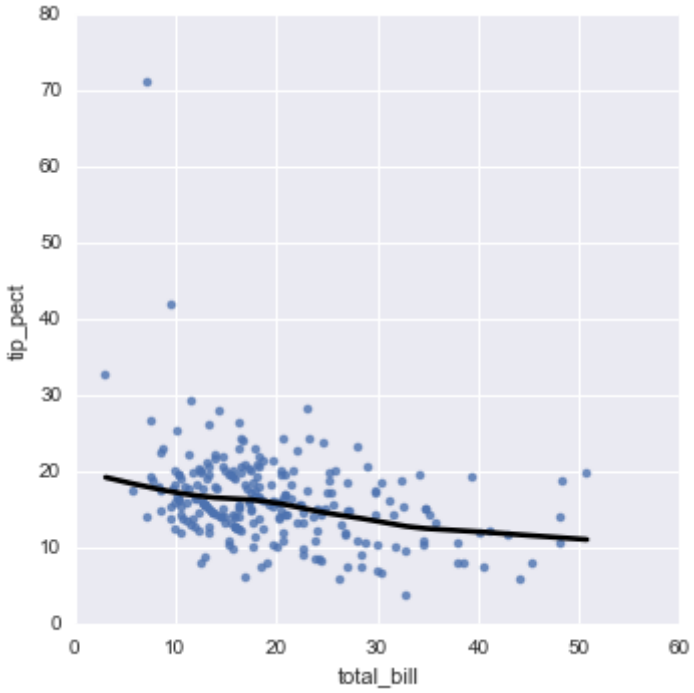


In [38]:
```
# reg_plot can be added to existing axes without modifying anything in the figure

# Create figure with 2 subplots
fig, (axis1,axis2) = plt.subplots(1,2,sharey =True)

sns.regplot("total_bill","tip_pect",tips,ax=axis1)
sns.violinplot(tips['tip_pect'],tips['size'],color='Reds_r',ax=axis2)
```

Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x1c46a080>



In [35]:
```
# Next up: We'll learn about
```



In [ ]:
```

```