Seaborn regplot

Tipping Dataset

regplot displays a scatterplot, a regression line and the 95% confidence interval around the regression line.

In [12]:

```
import pandas as pd
import seaborn as sns
```

In [13]:

```
tips = pd.read_csv('data/tips.csv')
tips.head(5)
```

Out[13]:

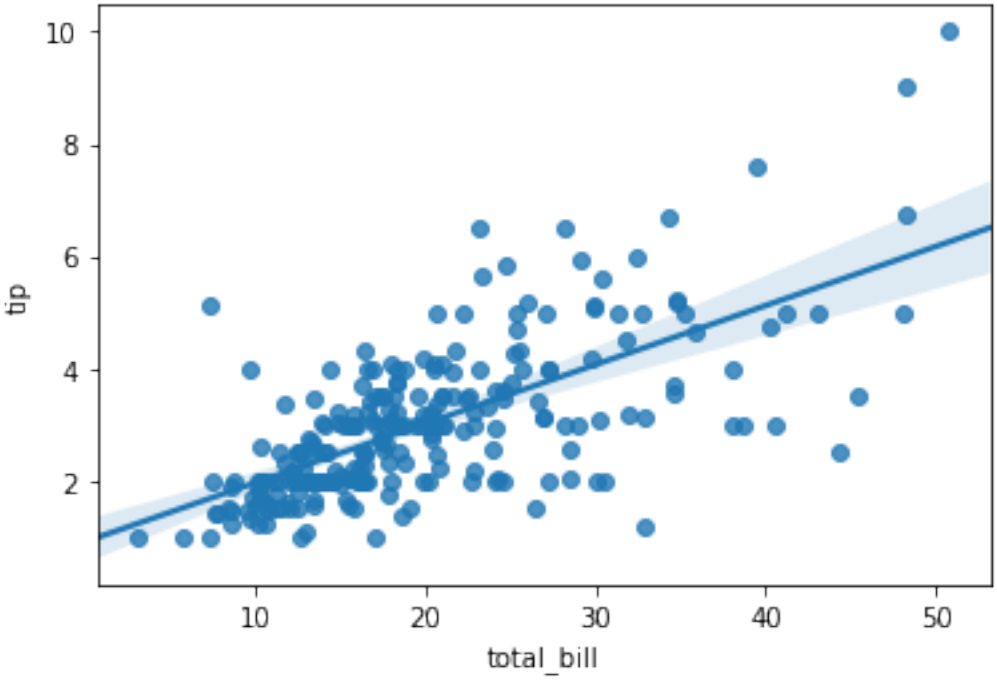
	total_bill	tip	smoker	day	time	size
0	16.99	1.01	No	Sun	Dinner	2
1	10.34	1.66	No	Sun	Dinner	3
2	21.01	3.50	No	Sun	Dinner	3
3	23.68	3.31	No	Sun	Dinner	2
4	24.59	3.61	No	Sun	Dinner	4

In [14]:

```
sns.regplot(x="total_bill", y="tip", data=tips)
```

Out[14]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a209ca8d0
>
```



Macrodata

In [15]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [16]:

```
macro = pd.read_csv('data/macrodata.csv')
macro.head()
```

Out[16]:

	year	quarter	realgdp	realcons	realinv	realgovt	realdpi	cpi	
0	1959.0	1.0	2710.349	1707.4	286.898	470.045	1886.9	28.98	1
1	1959.0	2.0	2778.801	1733.7	310.859	481.301	1919.7	29.15	1
2	1959.0	3.0	2775.488	1751.8	289.226	491.260	1916.4	29.35	1
3	1959.0	4.0	2785.204	1753.7	299.356	484.052	1931.3	29.37	1
4	1960.0	1.0	2847.699	1770.5	331.722	462.199	1955.5	29.54	1

m1 - the money supply is the entire stock of currency and other liquid instruments circulating in a country's economy as of a particular time. The money supply can include cash, coins and balances held in checking and savings accounts.

unemp - unemployment rate for the nation is the number of unemployed as a percentage of the labor force (the sum of the employed and unemployed).

In [17]:

```
data = macro[['m1', 'unemp']]
data.head()
```

Out[17]:

	m1	unemp
0	139.7	5.8
1	141.7	5.1
2	140.5	5.3
3	140.0	5.6
4	139.6	5.2

numpy.log() is the natural logarithm is logarithm in base e, element-wise.

In [18]:

```
np.log([1, np.e, np.e**2])
```

Out[18]:

```
array([0., 1., 2.])
```

In [19]:

```
trans_data = np.log(data).diff().dropna()
trans_data[-5:]
```

Out[19]:

	m1	unemp
198	0.045361	0.105361
199	0.066753	0.139762
200	0.010286	0.160343
201	0.037461	0.127339
202	0.012202	0.042560

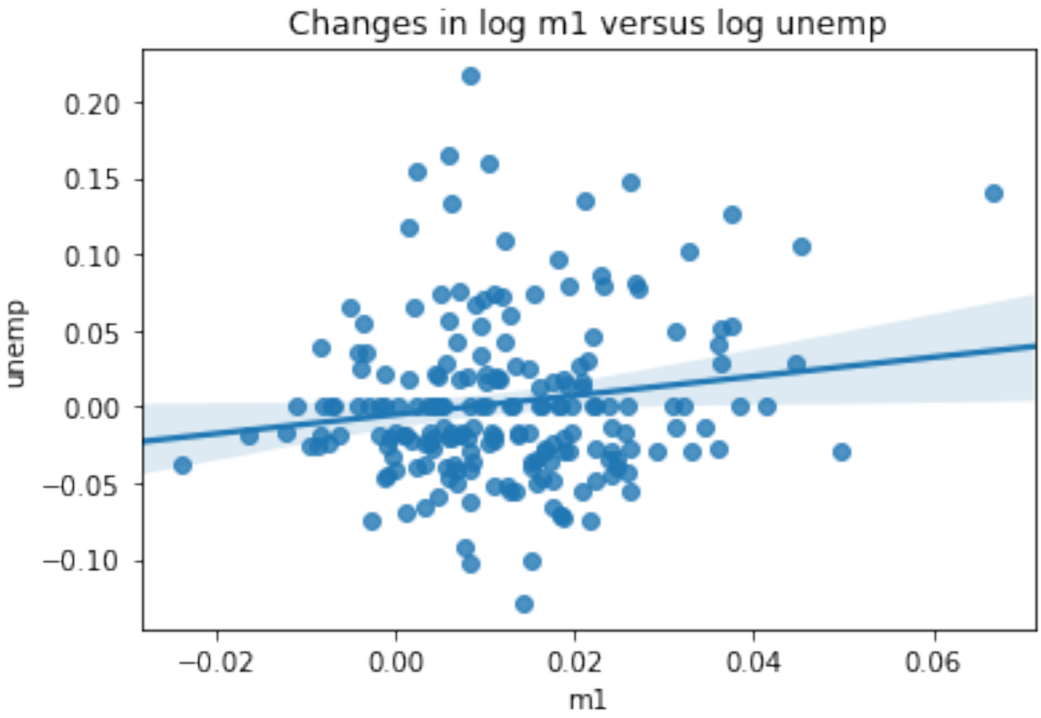
regplot draws a scatterplot of two variables, x and y, and then fit the regression model and plot the resulting regression line and a 95% confidence interval for that regression.

In [20]:

```
sns.regplot('m1', 'unemp', data=trans_data)
plt.title('Changes in log %s versus log %s' % ('m1', 'unemp'))
```

Out[20]:

```
Text(0.5, 1.0, 'Changes in log m1 versus log unemp')
```



Use of logarithms in economics: https://econbrowser.com/archives/2014/02/use-of-logarithms-in-economics (https://econbrowser.com/archives/2014/02/use-of-logarithms-in-economics)

What effect does the increase of the money supply have on the nation's monetary policy aimed at stimulating growth and reducing unemployment? https://www.enotes.com/homework-help/what-effect-does-increase-money-supply-have-479325 (https://www.enotes.com/homework-help/what-effect-does-increase-money-supply-have-479325)

Boston Housing Dataset

This example is from the books 'Beginning Data Analysis with Python and Jupyter' and 'Applied Deep Learning with Python' by Luis Capelo; Alex Galea. Published by Packt Publishing, 2018.

MEDV - Median House Value. How do we predict it?
RM - The number of rooms by house
LSTAT - % lower status of the population (lower class)

In [21]:

```
from sklearn import datasets
from sklearn.utils import Bunch

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
%matplotlib inline

boston = datasets.load_boston() #boston is sklearn.utils.Bunch
#Bunch?
df = pd.DataFrame(data = boston['data'], columns = boston['feature_
df['MEDV'] = boston['target']
y = df['MEDV'].copy()
del df['MEDV']
df = pd.concat((y, df), axis=1)
for col in ['ZN', 'NOX', 'RAD', 'PTRATIO', 'B']:
     del df[col]

cols = ['RM', 'AGE', 'TAX', 'LSTAT', 'MEDV']
df[cols].head()
```

Out[21]:

	RM	AGE	TAX	LSTAT	MEDV
0	6.575	65.2	296.0	4.98	24.0
1	6.421	78.9	242.0	9.14	21.6
2	7.185	61.1	242.0	4.03	34.7
3	6.998	45.8	222.0	2.94	33.4
4	7.147	54.2	222.0	5.33	36.2

Seaborn regplot generates:

- a scatterplot,
- the line of best fit linear model
- and 95% confidence interval

In [22]:

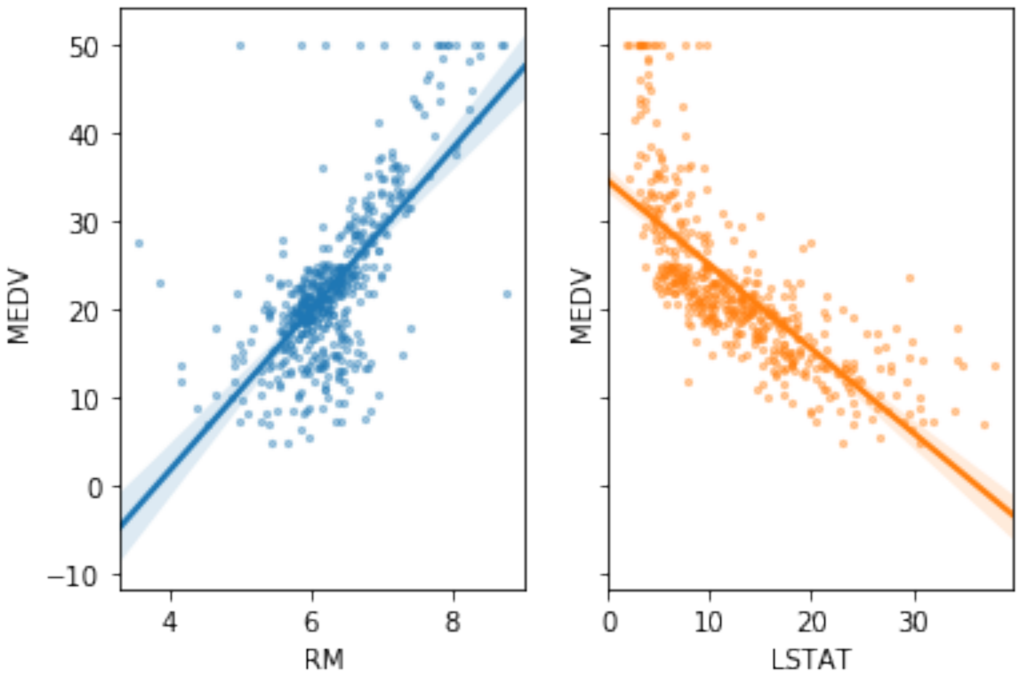
```
fig, ax = plt.subplots(1, 2, sharey=True)

sns.regplot('RM', 'MEDV', df, ax=ax[0], scatter_kws={'alpha': 0.4,

sns.regplot('LSTAT', 'MEDV', df, ax=ax[1], scatter_kws={'alpha': 0.
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a20c3bf60
>
```



The number of rooms per house (RM) and the % of the population that is lower class are highly correlated with the median house value (MDEV).

Learn more about regplot:
https://seaborn.pydata.org/generated/seaborn.regplot.html
(https://seaborn.pydata.org/generated/seaborn.regplot.html)
https://seaborn.pydata.org/tutorial/regression.html
(https://seaborn.pydata.org/tutorial/regression.html)