Lesson 23 - Confidence intervals for regression

We will use the bear dataset Import statsmodels, seaborn and the other libraries so we can use them in our code:

import numpy as np import matplotlib.pyplot as plt import pandas as pd import seaborn as sns import statsmodels.formula.api as smf, and type: %matplotlib inline

We will look at the relationship between male bears' weight and their length, and the relationship between maternal age and birth weight.

The columns are:

- age: male bear age in months
- month: the month of measurement (ex: 1 = January)
- Headlen: male bear head length in inches
- Headwidth: male bear head width in inches
- Neck: male bear neck circumference in inches
- Length: male bear body length in inches
- Chest: male bear distance around the chest in inches
- Weight: male bear weight in pounds

First, let's import the necessary libraries.

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import statsmodels.formula.api as smf
%matplotlib inline
```

Loading and cleaning the data

Read the CSV file into the dataframe babies . As in Lab 16, we need to add the parameter sep = "\s+" since the columns are separated by whitespaces instead of commas. sep = "\s+" defines separator as being one single white space or more.

In [3]:
```
bears = pd.read_csv("Male_Bear_Measurements.csv")
```

Display your babies dataframe below to check it was created properly.

In [30]:
```
bears.head()
```

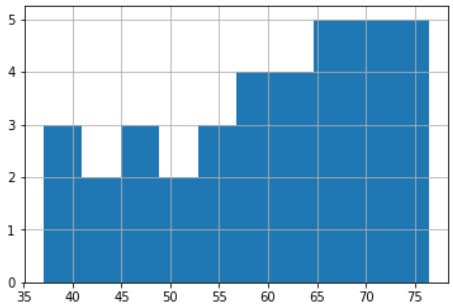
Out[30]:

	AGE	MONTH	HEADLEN	HEADWDTH	NECK	LENGTH	CHEST	WEIGHT	VAR WEIGHT
0	8	8	9.0	4.5	13.0	37.0	19.0	34	567.952381
1	9	9	10.0	4.0	13.0	40.0	23.0	40	567.952381
2	9	9	10.0	4.0	13.5	43.0	23.0	46	567.952381
3	16	4	10.0	5.0	15.0	41.0	26.0	64	1184.800000
4	16	4	10.0	4.0	15.5	48.0	26.0	60	1184.800000

Plot a histogram of the gestation column.

In [12]:
```
bears["LENGTH"].hist()
```

Out[12]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f11aefd7048>
```



What do you notice about the histogram?

Hint:

Drop any row with an NaN value (also done in Lab 18 and 22):

Pattern:

Let's check the unknown values have been successfully dropped by plotting a histogram of the weight or gestation column again.

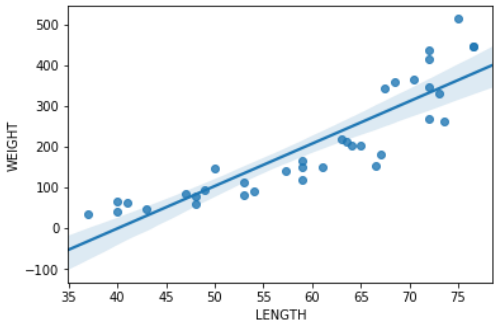
Relationship between weight and length of bears

In Lab 16 we looked at the relationship between birth weight and the number of gestational days (how long the baby was in the womb). We will look at this relationship again using regression.

First, we'll visualize the data. Use the Seaborn library (Lab 21) to plot a scatter plot with the regression line, where fish length, Length1 are on the x axis and fish weight, Weight is on the y axis.

In [13]:
```
sns.regplot(x = "LENGTH", y = "WEIGHT", data = bears)
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x7f11aefc9128>



Pattern:

Looking at the graph, do you think there is a linear relationship between fish length and their weight?

There appears to be a strong linear relationship between fish lengths and their weight. Let's compute the correlation matrix to get the correlation between the two variables:

In [14]:
```
bears.corr()
```

Out[14]:

	AGE	MONTH	HEADLEN	HEADWDTH	NECK	LENGTH	CHEST	WEIGHT	VAR WEIGHT
AGE	1.000000	0.062475	0.733517	0.751356	0.828331	0.724881	0.758577	0.831194	0.540300
MONTH	0.062475	1.000000	0.194382	0.015284	0.145123	0.131160	0.213384	0.153028	-0.211541
HEADLEN	0.733517	0.194382	1.000000	0.746273	0.929750	0.935881	0.874867	0.845835	0.565236
HEADWDTH	0.751356	0.015284	0.746273	1.000000	0.810770	0.730884	0.747917	0.749715	0.530826
NECK	0.828331	0.145123	0.929750	0.810770	1.000000	0.939277	0.952432	0.949077	0.637237
LENGTH	0.724881	0.131160	0.935881	0.730884	0.939277	1.000000	0.908492	0.889286	0.553897
CHEST	0.758577	0.213384	0.874867	0.747917	0.952432	0.908492	1.000000	0.965454	0.602114
WEIGHT	0.831194	0.153028	0.845835	0.749715	0.949077	0.889286	0.965454	1.000000	0.630056
VAR WEIGHT	0.540300	-0.211541	0.565236	0.530826	0.637237	0.553897	0.602114	0.630056	1.000000

Pattern:

What's the correlation between fish length, Length1 and their weight, Weight ? The correlation is .92 which is high

Next let's calculate this slope by computing the linear model, and then displaying the model summary:

In [16]:
```
lm = smf.ols(formula = "WEIGHT ~ LENGTH", data = bears).fit()
lm.summary()
```

Out[16]:

OLS Regression Results

Dep. Variable:	WEIGHT	**R-squared:**	0.791
Model:	OLS	**Adj. R-squared:**	0.785
Method:	Least Squares	**F-statistic:**	128.5
Date:	Sun, 15 Nov 2020	**Prob (F-statistic):**	4.26e-13
Time:	17:38:24	**Log-Likelihood:**	-199.88
No. Observations:	36	**AIC:**	403.8
Df Residuals:	34	**BIC:**	406.9
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-416.8303	56.016	-7.441	0.000	-530.669	-302.991
LENGTH	10.4081	0.918	11.338	0.000	8.542	12.274

Omnibus:	0.849	**Durbin-Watson:**	1.019
Prob(Omnibus):	0.654	**Jarque-Bera (JB):**	0.866
Skew:	0.205	**Prob(JB):**	0.649
Kurtosis:	2.360	**Cond. No.**	320.

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Pattern:

What's the slope of the regression line?

Remember that this relationship is only based on a sample of data. A different sample would give a different regression line and slope. To understand how much the slope could change depending on the sample, we will compute the 95% confidence interval for the slope.

First, we will create a function for computing the regression and extracting the slope from it:

def slope(df, x, y): formula_string = y + " ~ " + x # create a string containing the formula in advance lm = smf.ols(formula = formula_string, data = df).fit() return lm.params[1]

In [22]:
```
# df is the name of the dataframe
# x is the name of the column for the independent variable
# y is the name of the column for the dependent variable
def slope(bears, x, y):
    formula_string = y + " ~ " + x    # create a string containing the formula in advance
    lm = smf.ols(formula = formula_string, data = bears).fit()
    return lm.params[1]
```

Let's try calling (running) this function on our data:

In [21]:
```
slope(bears,"LENGTH","WEIGHT")
```

Out[21]: 10.408055193094903

Was the slope the same as your previous computation?

To find the 95% confidence interval, we will:

- create an empty list to store the slopes
- take 1000 bootstrap samples the same size as the original data
- for each sample, compute the slope of the regression line using our function and save it in our list

The pseudo-code is:
slopes = []
loop 1000 times:
    take a bootstrap sample (take a sample with replacement the same size as the original data)
    compute the slope of the regression line of the sample
    add the sample to slopes

Try writing the actual code below:

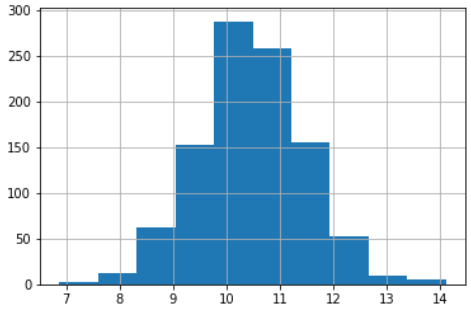
In [35]:
```
slopes = []
for i in range(1000):
    sample = bears.sample(36,replace = True)
    sample_slope = slope(sample,"LENGTH","WEIGHT")
    slopes.append(sample_slope)
```

Answer:

Plot the histogram of the slopes of the samples:

In [32]:
```
pd.Series(slopes).hist()
```

Out[32]: <matplotlib.axes._subplots.AxesSubplot at 0x7f11aee5f470>



In [36]:
```
pd.Series(slopes).mean()
```

Out[36]: 10.455039716105468

What is the mean of the slopes? 10.46 **
****How does this compare to the slope of the actual data? 10.46 > 10.41 but not by much!**

Were any of the sample slopes 0? Does this suggest a different sample of the data could have slope 0? Yes if the length of a bear is approx. less than 6.7 inches which is probably the size of a bear featus.

Finally, let's compute the 95% confidence interval by computing the 0.025 quantile (2.5 percentile) and 0.975 quantile (97.5 percentile). We can compute the 0.025 quantile with the code `pd.Series(slopes).quantile(0.025)`

In [37]:
```
pd.Series(slopes).quantile(0.025)
```

Out[37]: 8.568009269250641

In [38]:
```
pd.Series(slopes).quantile(0.975)
```

Out[38]: 12.336277890769356

What's your 95% confidence interval?

It should be close to (8.57,12.34).

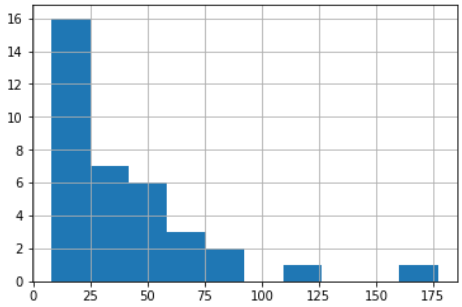
Male Bears' Age and their Weight

Let's look at whether there is a relationship between male bears' age and their weight. That is, can the age of bears in any way predict their weight?

Plot a histogram of the male bears' ages:

In [39]:
```
bears["AGE"].hist()
```

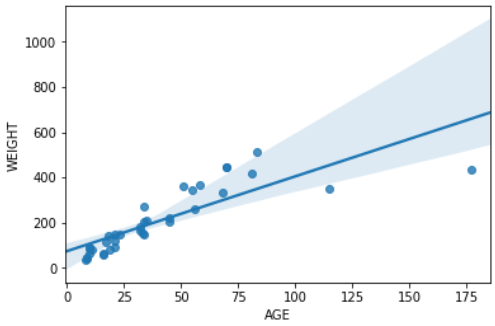
Out[39]: <matplotlib.axes._subplots.AxesSubplot at 0x7f11aed5c780>



First plot a scatter plot and regression line of bears' age (x) vs. their weight (y).

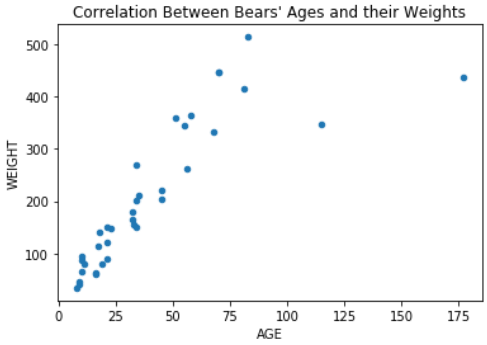
In [42]:
```
sns.regplot(x = "AGE", y = "WEIGHT", data = bears)
```

Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x7f11aec64d30>



In [43]:
```
bears.plot.scatter(x = "AGE", y = "WEIGHT")
plt.title("Correlation Between Bears' Ages and their Weights")
```

Out[43]: Text(0.5,1,"Correlation Between Bears' Ages and their Weights")



In [45]:
```
bears.corr()
```

Out[45]:

	AGE	MONTH	HEADLEN	HEADWDTH	NECK	LENGTH	CHEST	WEIGHT	VAR WEIGHT
AGE	1.000000	0.062475	0.733517	0.751356	0.828331	0.724881	0.758577	0.831194	0.540300
MONTH	0.062475	1.000000	0.194382	0.015284	0.145123	0.131160	0.213384	0.153028	-0.211541
HEADLEN	0.733517	0.194382	1.000000	0.746273	0.929750	0.935881	0.874867	0.845835	0.565236
HEADWDTH	0.751356	0.015284	0.746273	1.000000	0.810770	0.730884	0.747917	0.749715	0.530826
NECK	0.828331	0.145123	0.929750	0.810770	1.000000	0.939277	0.952432	0.949077	0.637237
LENGTH	0.724881	0.131160	0.935881	0.730884	0.939277	1.000000	0.908492	0.889286	0.553897
CHEST	0.758577	0.213384	0.874867	0.747917	0.952432	0.908492	1.000000	0.965454	0.602114
WEIGHT	0.831194	0.153028	0.845835	0.749715	0.949077	0.889286	0.965454	1.000000	0.630056
VAR WEIGHT	0.540300	-0.211541	0.565236	0.530826	0.637237	0.553897	0.602114	0.630056	1.000000

In [46]:
```
slope(bears,"AGE","WEIGHT")
```

Out[46]: 3.309577163892131

Does the slope look close to 0? If the slope is 0 it would indicate no relationship. However, for any particular sample of data, the slope is unlikely to be exactly 0. Therefore, we want to construct the 95% confidence interval for the slope and check if it contains 0.

We will now compute the 95% confidence interval.

First, compute 1000 bootstrap samples, saving the slope of the regression line of each sample.

In [49]:
```
slopes1 = []
for i in range(1000):
    sample1 = bears.sample(36,replace = True)
    sample_slope1 = slope(sample1,"AGE","WEIGHT")
    slopes1.append(sample_slope1)
```

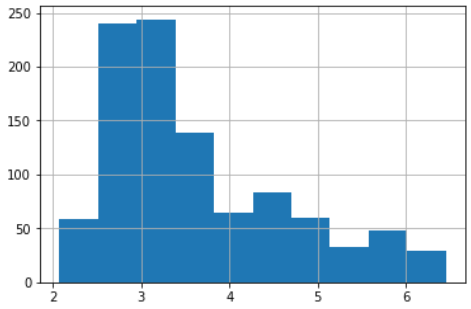
In [58]:
```
pd.Series(slopes1).mean()
```

Out[58]: 3.6293779874536183

Plot a histogram of the slopes:

In [60]:
```
pd.Series(slopes1).hist()
```

Out[60]: <matplotlib.axes._subplots.AxesSubplot at 0x7f11aeae72e8>



Does it look like the 95% confidence interval for the slope will contain 0?

Let's formally calculate the 95% confidence interval for the slope by computing the 0.025 and 0.975 quantiles:

In [56]:
```
pd.Series(slopes1).quantile(0.025)
```

Out[56]: 2.400273750193928

In [57]:
```
pd.Series(slopes1).quantile(0.975)
```

Out[57]: 6.033003825276708

What's the 95% confidence interval for the slope? Does it contain the slope of the regression line for bears age vs weight?

In [2]:
```
print("the confidence interval is (2.4, 6.03) which contains the slope of the regression line for bears' age vs. weight.  Thus we can be 95% confident that the true slope
```

```
the confidence interval is (2.4, 6.03) which contains the slope of the regression line for bears' age vs. weight.  Thus we can be 95% confident that the true slope is wi
thin this confidence interval. Another way to look at it is we're 95% confident that the true regression line lies somewhere in that blue zone.
```

In [ ]: