Home Assignment 5. Correlation and linear regression

Probability and Statistics, Spring 2019
Salla Vesterinen Helsinki Metropolia University of Applied Sciences (https://www.metropolia.fi/en)

In [1]:
```
# Import libraries
%pylab inline
import pandas as pd
import seaborn as sns
from numpy import polyfit, polyval
from scipy.stats import linregress
```

```
Populating the interactive namespace from numpy and matplotlib
```

The following problems uses data from OpenIntro Statistics (https://www.openintro.org/stat/) Lab for Linear regression (https://www.openintro.org/stat/). The original R-data was read into R-studio and exported as csv-file.

NOTE! The data file can be found from the same OMA assignment where you downloaded this Notebook.

Problem 1.

- Read in the 'mlb11.csv' file using pandas read_csv function. Tip: Use the first column (Unnamed: 0) as index. See read_csv documentation for more details how to do that.
- Plot the relationship between the `runs` and `at_bats` variables. Use `at_bats` as predictor (https://en.wikipedia.org/wiki/Regression_analysis) and `runs` as dependent variable (https://en.wikipedia.org/wiki/Dependent_and_independent_variables).
- Does the relationship look linear?
- If you knew a team's `at_bats`, would you be comfortable using a linear model to predict the number of runs?

In [15]:
```
# Read the example data into Python
file=open("mlb11.csv", "r")
sep = ","
data = pd.read_csv(file, sep, index_col=0)
data.head()
```

Out[15]:

	team	runs	at_bats	hits	homeruns	bat_avg	strikeouts	stolen_bases	wins	new_onbase	new_slug	new_obs
1	Texas Rangers	855	5659	1599	210	0.283	930	143	96	0.340	0.460	0.800
2	Boston Red Sox	875	5710	1600	203	0.280	1108	102	90	0.349	0.461	0.810
3	Detroit Tigers	787	5563	1540	169	0.277	1143	49	95	0.340	0.434	0.773
4	Kansas City Royals	730	5672	1560	129	0.275	1006	153	71	0.329	0.415	0.744
5	St. Louis Cardinals	762	5532	1513	162	0.273	978	57	90	0.341	0.425	0.766

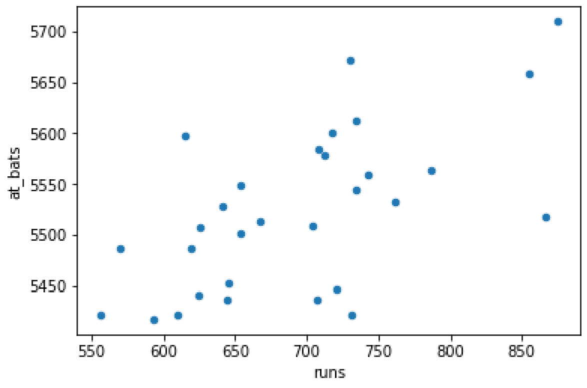
In [16]:
```
data[['runs', 'at_bats']].corr()
```

Out[16]:

	runs	at_bats
runs	1.000000	0.610627
at_bats	0.610627	1.000000

```
In [19]: data.plot.scatter(x = 'runs', y = 'at_bats')
```

```
Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0x26acb7684a8>
```



The relationship does not look extremely linear, since the correlation is only 61% and I wouldn't predict the number of runs based on it.

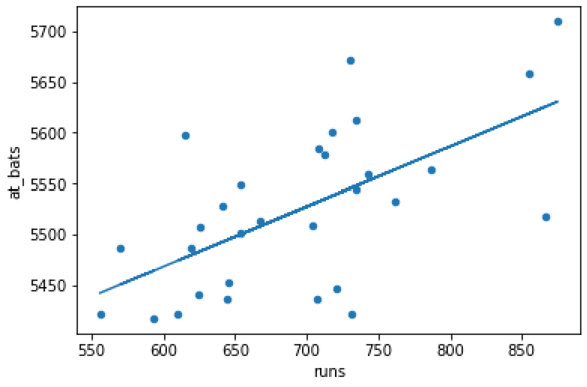
Problem 2.

- Looking at your plot from the previous problem, describe the relationship between these two variables. Make sure to discuss the form, direction, and strength of the relationship as well as any unusual observations.
- Using the `polyfit` function choose a line that fits best to the data in previous problem.
- Make a graph where you have drawn both the datapoints and the best fitted linear line between the two variables.

The relationship between these two variables is somewhat linear and positive but not strong since the correlation coefficient is low.
There are a few outliers

```
In [22]: x=data["runs"]
         y=data["at_bats"]
         data.plot.scatter(x = 'runs', y = 'at_bats')
         p = polyfit(x, y, 1)
         y = polyval(p, x)
         plot(x, y)
```

```
Out[22]: [<matplotlib.lines.Line2D at 0x26acb8247b8>]
```

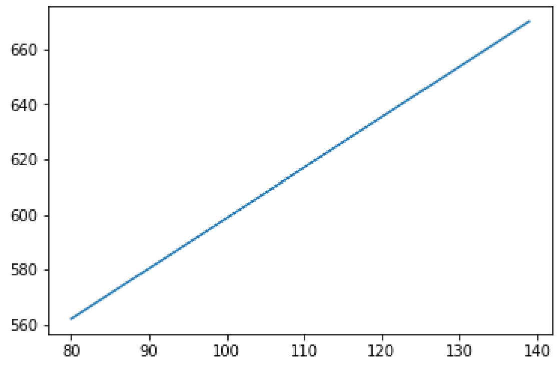


Problem 3.

- Fit a new model that uses `homeruns` to predict `runs`. Using the outputs of your code, write the equation of the regression line.
- Make a graph that combines both the datapoints and the regression line.
- What does the slope tells us about the relationship between success of a team and its home runs?

```
In [27]: x=data["homeruns"]
         y=data["runs"]
         p = polyfit(x, y, 1)
         print(p)
         x = np.arange(80, 140)
         y = p[0] * x + p[1]
         plot(x, y)
         show()
```

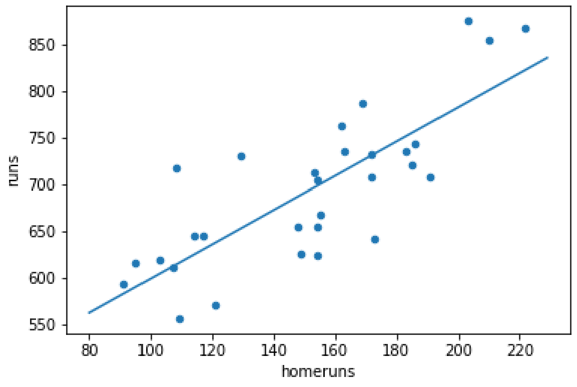
```
[  1.83454162 415.23888492]
```



In this case the line is:
y = 1.8345 * x + 415.239

```
In [31]: data.plot.scatter(x = 'homeruns', y = 'runs')
         x = np.arange(80, 230)
         y = p[0] * x + p[1]
         plot(x, y)
```

```
Out[31]: [<matplotlib.lines.Line2D at 0x26acbc33c18>]
```



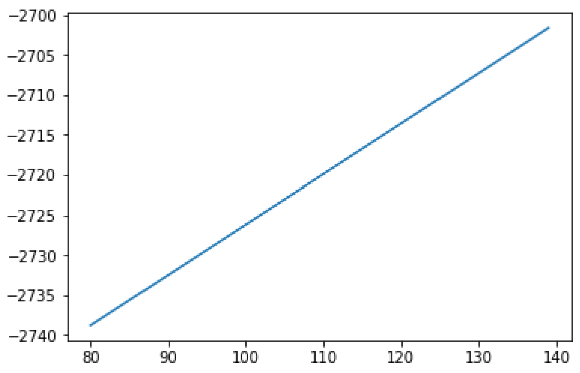
The slope tells us about the relationship between success of a team and its home runs, which in this case tells us that the more runs there are the more homeruns there are.
y=415.2389+1.8345∗homeruns

Problem 4. (Bonus)

- If a team manager saw the regression line and not the actual data (you could draw the line only), how many runs would he or she predict for a team with 5,578 at-bats?
- Is this an overestimate or an underestimate, and by how much? (In other words, what is the residual for this prediction?)

In [50]:
```
x=data["at_bats"]
y=data["runs"]
p = polyfit(x, y, 1)
print(p)
x = np.arange(80, 140)
y = p[0] * x + p[1]
plot(x, y)
show()
```

```
[ 6.30549993e-01 -2.78924289e+03]
```

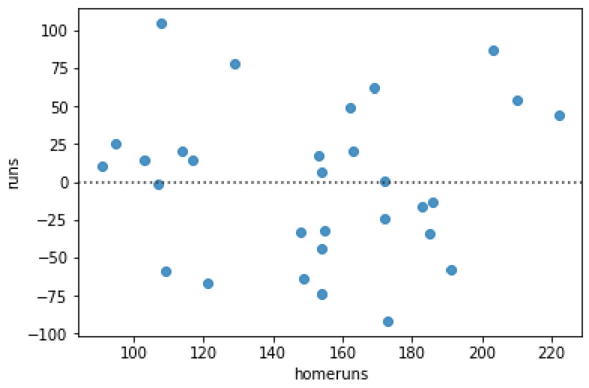


In [51]:
```
y = -2789.2429 + 0.6305 * 5578
print(y)
```

```
727.6860999999994
```

In [52]:
```
sns.residplot(data = data, x = 'homeruns', y = 'runs')
```

Out[52]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x26acbeb6fd0>
```



In [65]:
```
data[(data.at_bats>=5575)&(data.at_bats<=5580)]
```

Out[65]:

	team	runs	at_bats	hits	homeruns	bat_avg	strikeouts	stolen_bases	wins	new_onbase	new_slug	new_obs
16	Philadelphia Phillies	713	5579	1409	153	0.253	1024	96	102	0.323	0.395	0.717

It's an over estimation.
The closest to 5578 in the data is 5579 bats with 713 runs.
Comparing to the predicted one of 728, it is an overestimation of around 15.

P.S. It's worth to read at least the introduction of the original laboratory exercise (http://htmlpreview.github.io/?https://github.com/andrewpbray/oiLabs-base-R/blob/master/simple_regression/simple_regression.html). This assignment follows it's Creative Commons Attribution-ShareAlike 3.0 Unported (https://creativecommons.org/licenses/by-sa/3.0/) license policy.