# LinearRegression

December 20, 2016

## 1 Linear Regression

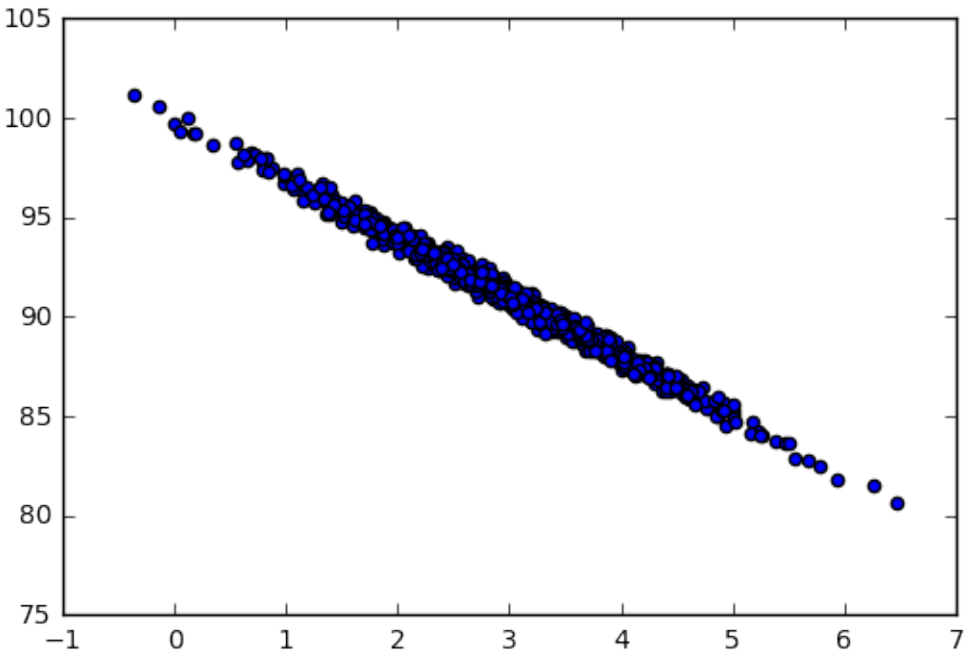
Let's fabricate some data that shows a roughly linear relationship between page speed and amount purchased:

In [1]:
```
%matplotlib inline
import numpy as np
from pylab import *

pageSpeeds = np.random.normal(3.0, 1.0, 1000)
purchaseAmount = 100 - (pageSpeeds + np.random.normal(0, 0.1, 1000)) * 3

scatter(pageSpeeds, purchaseAmount)
```

Out[1]:
```
<matplotlib.collections.PathCollection at 0x5cb8a50>
```




As we only have two features, we can keep it simple and just use scipy.state.linregress:

```
In [3]: from scipy import stats

        slope, intercept, r_value, p_value, std_err = stats.linregress(pageSpeeds, purchaseAmoun
```

Not surprisngly, our R-squared value shows a really good fit:

```
In [4]: r_value ** 2

Out[4]: 0.98898447900188802
```

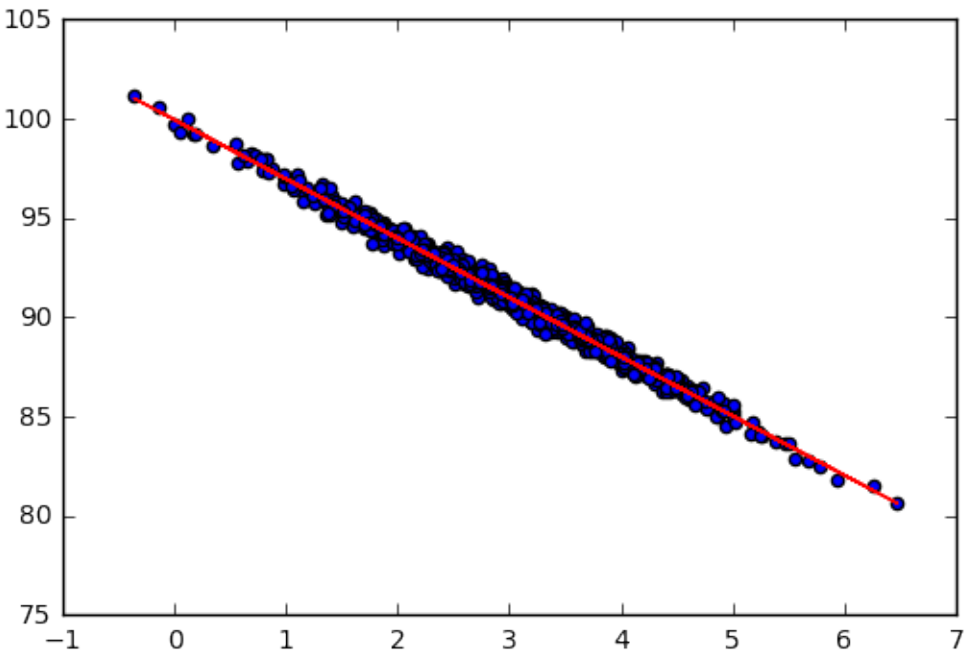
Let's use the slope and intercept we got from the regression to plot predicted values vs. observed:

```
In [5]: import matplotlib.pyplot as plt

        def predict(x):
            return slope * x + intercept    #Y = mX + C

        fitLine = predict(pageSpeeds)      #Y

        plt.scatter(pageSpeeds, purchaseAmount)
        plt.plot(pageSpeeds, fitLine, c='r')
        plt.show()
```



### 1.1 Activity

Try increasing the random variation in the test data, and see what effect it has on the r-squared error value.

```
In [ ]:
```