LinearRegression_Basic

April 24, 2018

0.1 Basic Linear Regression

In [52]:
```
%matplotlib inline
from sklearn import linear_model
import pandas as pd
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="whitegrid", color_codes=True)
```

In [53]:
```
X_weight = [[700],[750],[760],[800],[820],[890],[900],[950],[970],[1000],[1090],[1100],
Y_speed = [[100],[120],[140],[160],[170],[190],[220],[250],[300],[320],[350],[380],[400
Y_speed = Y_speed[::-1]
```

In [54]:
```
for data in zip(Y_speed,X_weight):
    print(data[1], '>', data[0])
```

```
[700] > [450]
[750] > [420]
[760] > [400]
[800] > [380]
[820] > [350]
[890] > [320]
[900] > [300]
[950] > [250]
[970] > [220]
[1000] > [190]
[1090] > [170]
[1100] > [160]
[1200] > [140]
[1240] > [120]
[1290] > [100]
```

In [55]:
```
lr = linear_model.LinearRegression()
```

In [57]:
```
x_train,x_test,y_train,y_test = train_test_split(X_weight,Y_speed)
```

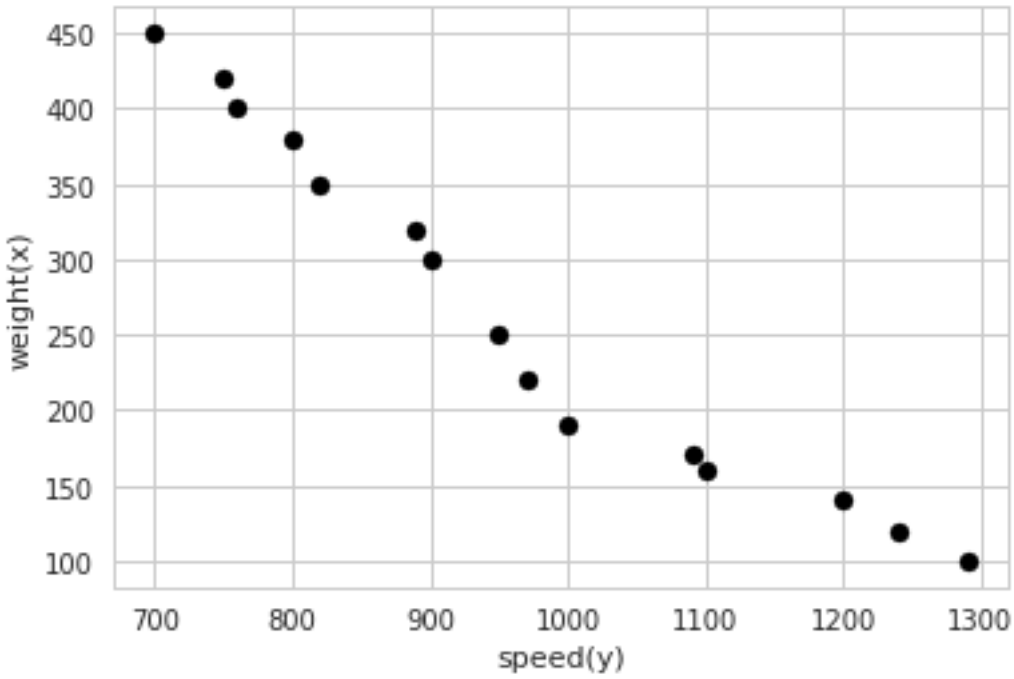
```
In [58]: lr.fit(x_train,y_train)

Out[58]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

In regression with multiple independent variables, the coefficient tells you how much the dependent variable is expected to increase when that independent variable increases by one, holding all the other independent variables constant. Remember to keep in mind the units which your variables are measured in.

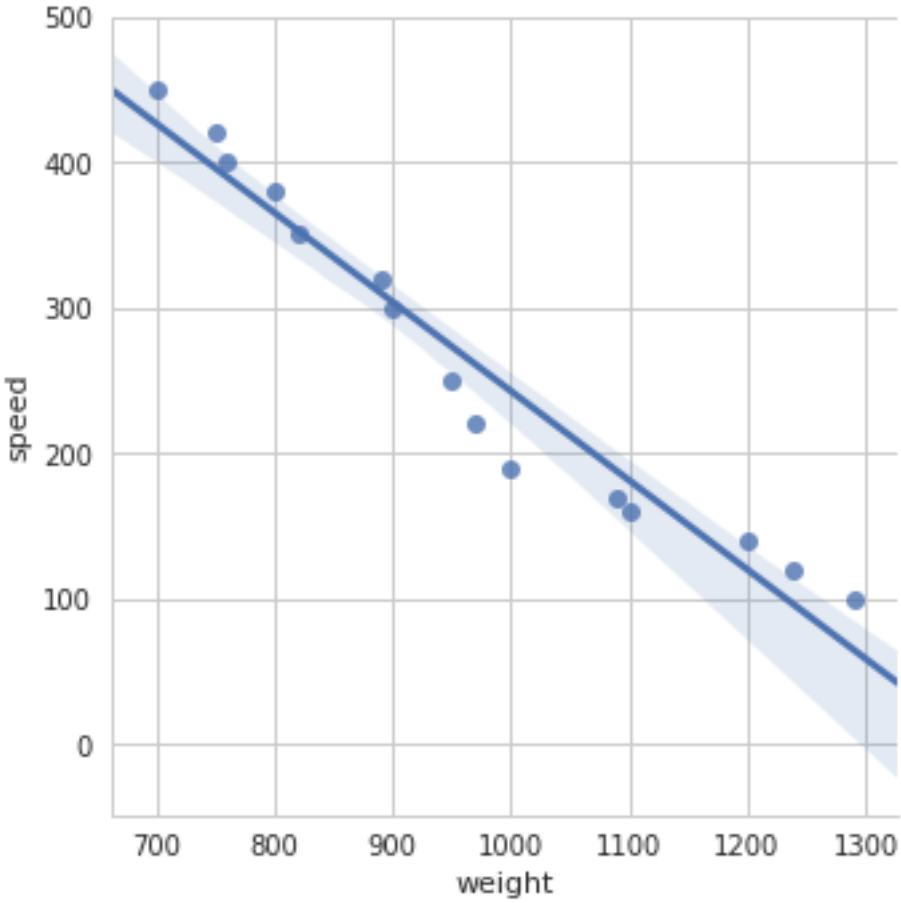
```
In [59]: lr.coef_

Out[59]: array([[-0.61847526]])
```

The intercept (often labeled the constant) is the expected mean value of Y when all X=0. Start with a regression equation with one predictor, X. If X sometimes = 0, the intercept is simply the expected mean value of Y at that value.

```
In [60]: lr.intercept_

Out[60]: array([861.78663588])

In [61]: x_test ,'>', y_test

Out[61]: ([[1200], [800], [970], [900]], '>', [[140], [380], [220], [300]])

In [62]: for i,j in zip(x_test, y_test):
                 print(j, '>', lr.predict(int(str(i).strip('[]'))))

[140] > [[119.61632487]]
[380] > [[367.00642854]]
[220] > [[261.86563448]]
[300] > [[305.15890262]]


In [63]: plt.scatter(X_weight,Y_speed,color='black')
         plt.xlabel('speed(y)')
         plt.ylabel('weight(x)')

Out[63]: Text(0,0.5,'weight(x)')
```




```
In [68]: x = [int(str(i).strip('[]')) for i in X_weight]
         y = [int(str(i).strip('[]')) for i in Y_speed]
         df = pd.DataFrame({'weight': x,'speed':y})
         sns.lmplot(x='weight',y='speed',data=df)

Out[68]: <seaborn.axisgrid.FacetGrid at 0x7f2d9a9e9b00>
```


500
400
300
200
100
0
speed
700
800
900
1000
1100
1200
1300
weight