

14.3) The Regression Line

Vitor Kamada

December 2019

Tables, Graphics, and Figures from

**Computational and Inferential Thinking:
The Foundations of Data Science**

Adhikari & DeNero (2019): Ch 15.2 The
Regression Line

<https://www.inferentialthinking.com/>

Galton's data

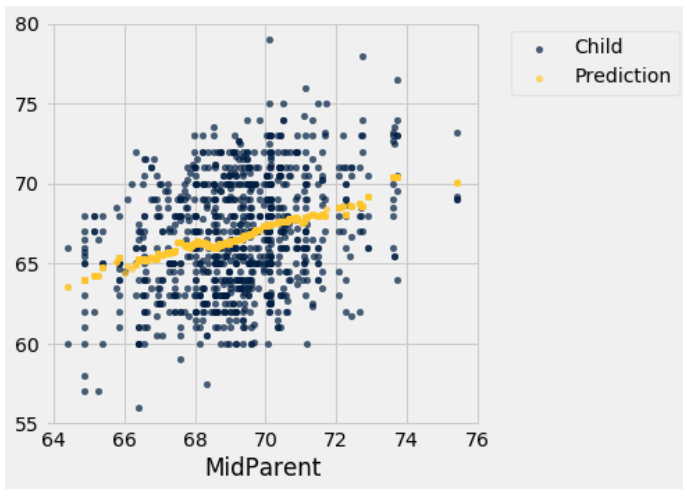
```
from datascience import *
import numpy as np

path_data = 'https://github.com/data-8/textbook/raw/gh-pages/data/'
galton = Table.read_table(path_data + 'galton.csv')
heights = Table().with_columns(
    'MidParent', galton.column('midparentHeight'),
    'Child', galton.column('childHeight'))

def predict_child(mpht):
    close_points = heights.where('MidParent',
                                are.between(mpht-0.5, mpht + 0.5))
    return close_points.column('Child').mean()

heights_with_predictions = heights.with_column(
    'Prediction', heights.apply(predict_child, 'MidParent'))
```

```
%matplotlib inline
import matplotlib.pyplot as plots
plots.style.use('fivethirtyeight')
heights_with_predictions.scatter('MidParent')
```



```
def standard_units(xyz):  
    "Convert any array of numbers to standard units."  
    return (xyz - np.mean(xyz))/np.std(xyz)
```

```
heights_SU = Table().with_columns(  
    'MidParent SU', standard_units(heights.column('MidParent'))),  
    'Child SU', standard_units(heights.column('Child')))
```

MidParent SU	Child SU
3.45465	1.80416
3.45465	0.686005
3.45465	0.630097

```
sd_midparent = np.std(heights.column(0))
```

1.8014050969207571

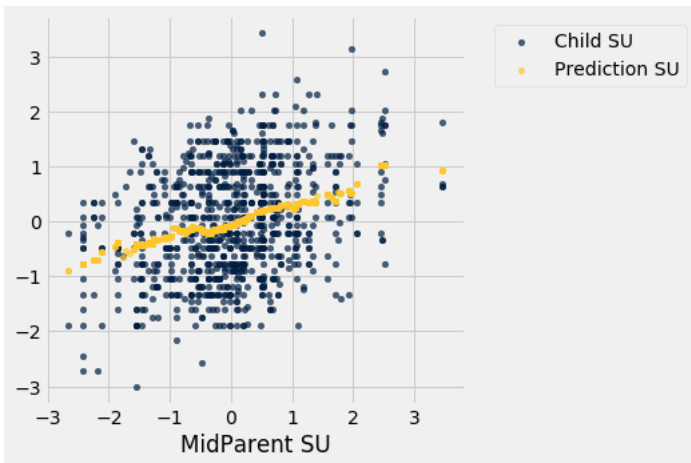
```
0.5/sd_midparent
```

0.277561110965367

```
def predict_child_su(mpht_su):  
    close = 0.5/sd_midparent  
    close_points = heights_SU.where('MidParent SU',  
                                     are.between(mpht_su-close, mpht_su + close))  
    return close_points.column('Child SU').mean()
```

```
heights_with_su_predictions = heights_SU.with_column(  
    'Prediction SU', heights_SU.apply(predict_child_su, 'MidParent SU'))
```

```
heights_with_su_predictions.scatter('MidParent SU')
```

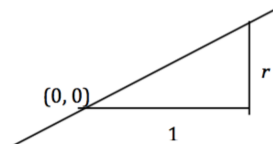


The Equation of the Regression Line

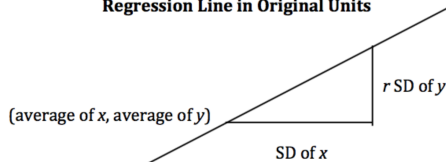
$$\hat{y} = r \cdot x$$

$$\frac{\text{estimate of } y - \text{average of } y}{\text{SD of } y} = r \times \frac{\text{the given } x - \text{average of } x}{\text{SD of } x}$$

Regression Line in Standard Units



Regression Line in Original Units



$$\text{Slope} = r \cdot \frac{\text{SD of } y}{\text{SD of } x}$$

$$\text{Intercept} = (\text{average of } y) - \text{slope} \cdot (\text{average of } x)$$


```
def correlation(t, label_x, label_y):
    return np.mean(standard_units(t.column(label_x))\
                    *standard_units(t.column(label_y)))

galton_r = correlation(heights, 'MidParent', 'Child')
```

0.32094989606395924

```
def slope(t, label_x, label_y):
    r = correlation(t, label_x, label_y)
    return r*np.std(t.column(label_y))/np.std(t.column(label_x))

def intercept(t, label_x, label_y):
    return np.mean(t.column(label_y)) - \
           slope(t, label_x, label_y)*np.mean(t.column(label_x))

galton_slope = slope(heights, 'MidParent', 'Child')
galton_intercept = intercept(heights, 'MidParent', 'Child')
galton_slope, galton_intercept
```

(0.637360896969479, 22.63624054958975)

```
galton_slope*70.48 + galton_intercept
```

67.55743656799862

```
heights_with_predictions.where('MidParent',  
                                are.equal_to(70.48)).show(3)
```

MidParent	Child	Prediction
-----------	-------	------------

70.48	74	67.6342
-------	----	---------

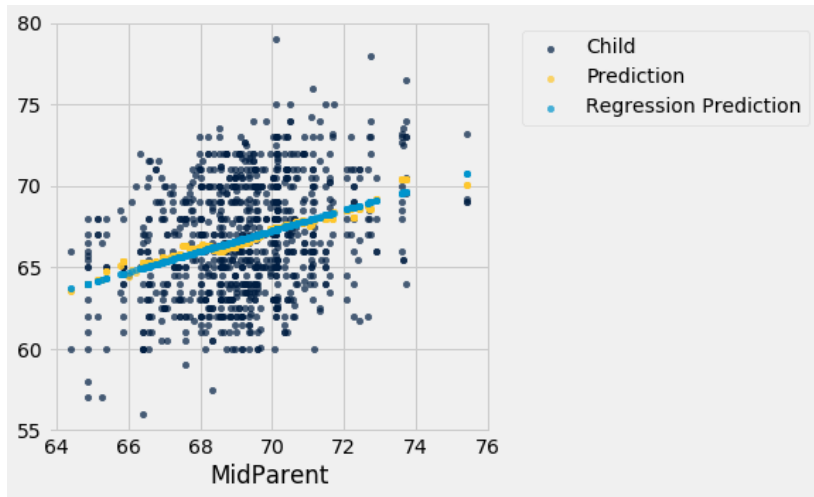
70.48	70	67.6342
-------	----	---------

70.48	68	67.6342
-------	----	---------

```
heights_with_predictions = heights_with_predictions.with_column(  
    'Regression Prediction',  
    galton_slope*heights.column('MidParent') + galton_intercept)
```

MidParent	Child	Prediction	Regression Prediction
75.43	73.2	70.1	70.7124
75.43	69.2	70.1	70.7124
75.43	69	70.1	70.7124
75.43	69	70.1	70.7124
73.66	73.5	70.4158	69.5842

```
heights_with_predictions.scatter('MidParent')
```



```
def fit(table, x, y):  
    a = slope(table, x, y)  
    b = intercept(table, x, y)  
    return a * table.column(x) + b
```

```
heights.with_column('Fitted', fit(heights,  
    'MidParent', 'Child')).scatter('MidParent')
```

