15.1) Least Squares Regression

Vitor Kamada

December 2019

Reference

Tables, Graphics, and Figures from

Computational and Inferential Thinking: The Foundations of Data Science

Adhikari & DeNero (2019): Ch 15.3 The Method of Least Squares

https://www.inferentialthinking.com/

Novel "Little Women"

```
from datascience import *
import numpy as np
```

```
path_data = 'https://github.com/data-8/textbook/raw/gh-pages/data/'
little_women = Table.read_table(path_data + 'little_women.csv')
```

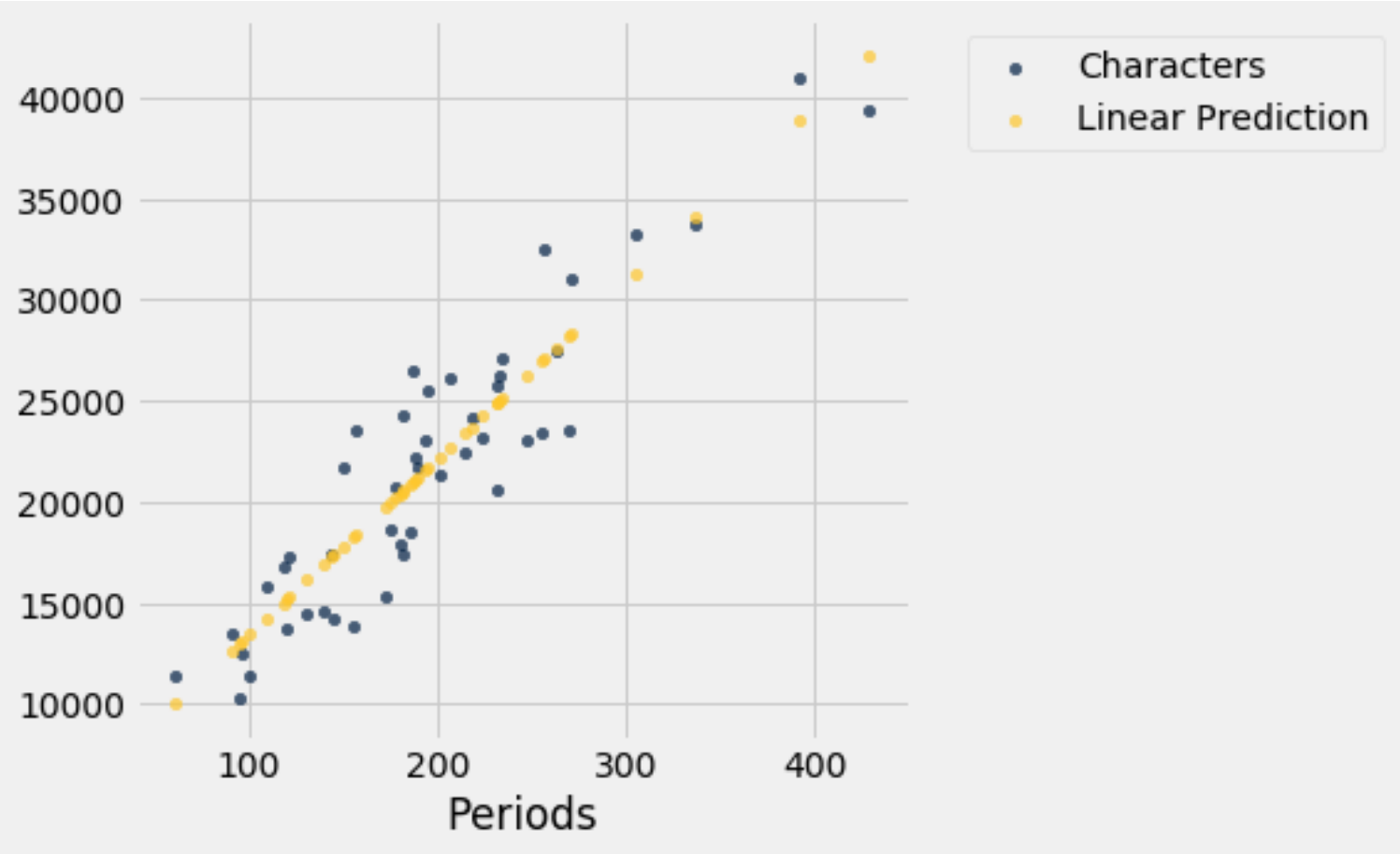
Characters	Periods
21759	189
22148	188
20558	231

One row for every chapter

Functions

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

def correlation(t, x, y):
    return np.mean(standard_units(t.column(x))*standard_units(t.column(y)))

def slope(table, x, y):
    r = correlation(table, x, y)
    return r * np.std(table.column(y))/np.std(table.column(x))

def intercept(table, x, y):
    a = slope(table, x, y)
    return np.mean(table.column(y)) - a * np.mean(table.column(x))

def fit(table, x, y):
    """Return the height of the regression line at each x value."""
    a = slope(table, x, y)
    b = intercept(table, x, y)
    return a * table.column(x) + b
```

```
%matplotlib inline
import matplotlib.pyplot as plots
plots.style.use('fivethirtyeight')
lw_with_predictions = little_women.with_column('Linear Prediction',
                        fit(little_women, 'Periods', 'Characters'))
lw_with_predictions.scatter('Periods')
```



```
actual = lw_with_predictions.column('Characters')
predicted = lw_with_predictions.column('Linear Prediction')
errors = actual - predicted
lw_with_predictions.with_column('Error', errors)
```

$$e = y - \hat{y}$$

Characters	Periods	Linear Prediction	Error
21759	189	21183.6	575.403
22148	188	21096.6	1051.38
20558	231	24836.7	-4278.67
25526	195	21705.5	3820.54

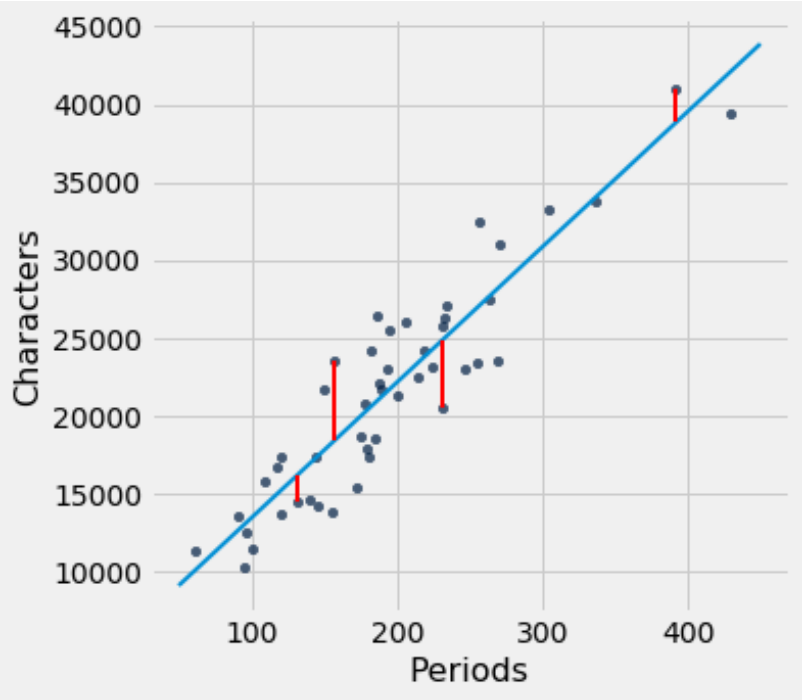
Plot Vertical Distance

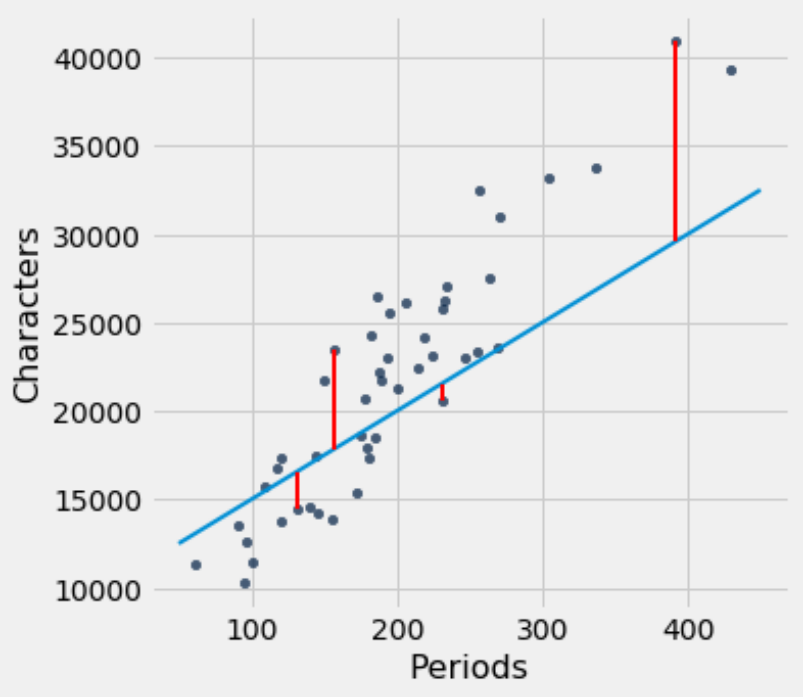
```
lw_reg_slope = slope(little_women, 'Periods', 'Characters')
lw_reg_intercept = intercept(little_women, 'Periods', 'Characters')
sample = [[131, 14431], [231, 20558], [392, 40935], [157, 23524]]
def lw_errors(slope, intercept):
    little_women.scatter('Periods', 'Characters')
    xlims = np.array([50, 450])
    plots.plot(xlims, slope * xlims + intercept, lw=2)
    for x, y in sample:
        plots.plot([x, x], [y, slope * x + intercept], color='r', lw=2)
```

```
print('Slope of Regression Line:    ',
      np.round(lw_reg_slope), 'characters per period')
print('Intercept of Regression Line:',
      np.round(lw_reg_intercept), 'characters')
lw_errors(lw_reg_slope, lw_reg_intercept)
```

```
Slope of Regression Line:      87.0 characters per period
Intercept of Regression Line: 4745.0 characters
```

```
lw_errors(50, 10000)
```





Root Mean Squared Error = $\sqrt{E(y - \hat{y})^2}$

```
def lw_rmse(slope, intercept):
    lw_errors(slope, intercept)
    x = little_women.column('Periods')
    y = little_women.column('Characters')
    fitted = slope * x + intercept
    mse = np.mean((y - fitted) ** 2)
    print("Root mean squared error:", mse ** 0.5)
```

```
lw_rmse(lw_reg_slope, lw_reg_intercept)
```

(87, 4745)

```
Root mean squared error: 2701.690785311856
```

```
lw_rmse(50, 10000)
```

```
Root mean squared error: 4322.167831766537
```

Regression line Minimizes Root Mean Squared Error

$$\sqrt{E(y - \hat{y})^2}$$

```
best = minimize(lw_mse)
```

```
print("slope from formula:          ", lw_reg_slope)
print("slope from minimize:         ", best.item(0))
print("intercept from formula:      ", lw_reg_intercept)
print("intercept from minimize:     ", best.item(1))
```

```
slope from formula:          86.97784125829821
slope from minimize:         86.97784116615884
intercept from formula:      4744.784796574928
intercept from minimize:     4744.784845352655
```