15.2) Visual Diagnostics

Vitor Kamada

December 2019

Reference

Tables, Graphics, and Figures from

Computational and Inferential Thinking: The Foundations of Data Science

Adhikari & DeNero (2019): Ch 15.5 Visual Diagnostics

https://www.inferentialthinking.com/

Galton's data

```
from datascience import *
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plots
plots.style.use('fivethirtyeight')
path_data = 'https://github.com/data-8/textbook/raw/gh-pages/data/'
galton = Table.read_table(path_data + 'galton.csv')
heights = galton.select('midparentHeight', 'childHeight')
heights = heights.relabel(0, 'MidParent').relabel(1, 'Child')
hybrid = Table.read_table(path_data + 'hybrid.csv')
```

Functions

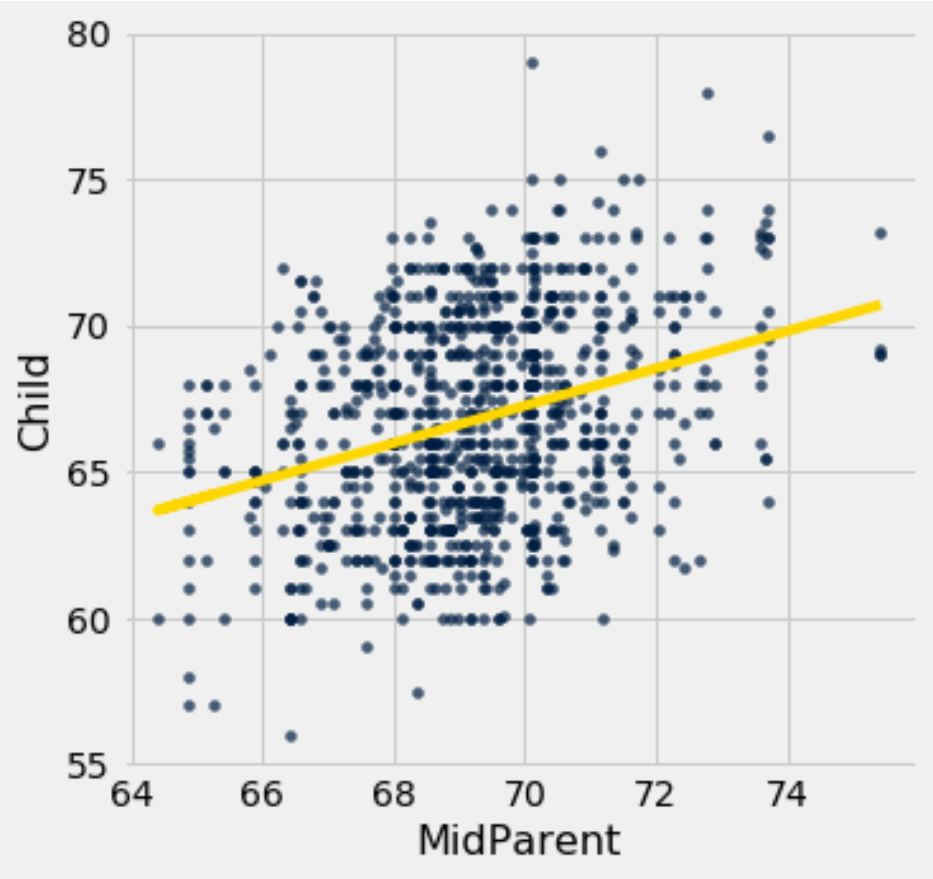
```
def standard_units(x):
    return (x - np.mean(x))/np.std(x)

def correlation(table, x, y):
    x_in_standard_units = standard_units(table.column(x))
    y_in_standard_units = standard_units(table.column(y))
    return np.mean(x_in_standard_units * y_in_standard_units)

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

```
def residual(table, x, y):
    return table.column(y) - fit(table, x, y)

heights = heights.with_columns(
        'Fitted Value', fit(heights, 'MidParent', 'Child'),
        'Residual', residual(heights, 'MidParent', 'Child'))
```

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

MidParent	Child	Fitted Value	Residual
75.43	73.2	70.7124	2.48763
75.43	69.2	70.7124	-1.51237
75.43	69	70.7124	-1.71237

```
def scatter_fit(table, x, y):
    table.scatter(x, y, s=15)
    plots.plot(table.column(x), fit(table, x, y), lw=4, color='gold')
    plots.xlabel(x)
    plots.ylabel(y)

scatter_fit(heights, 'MidParent', 'Child')
```



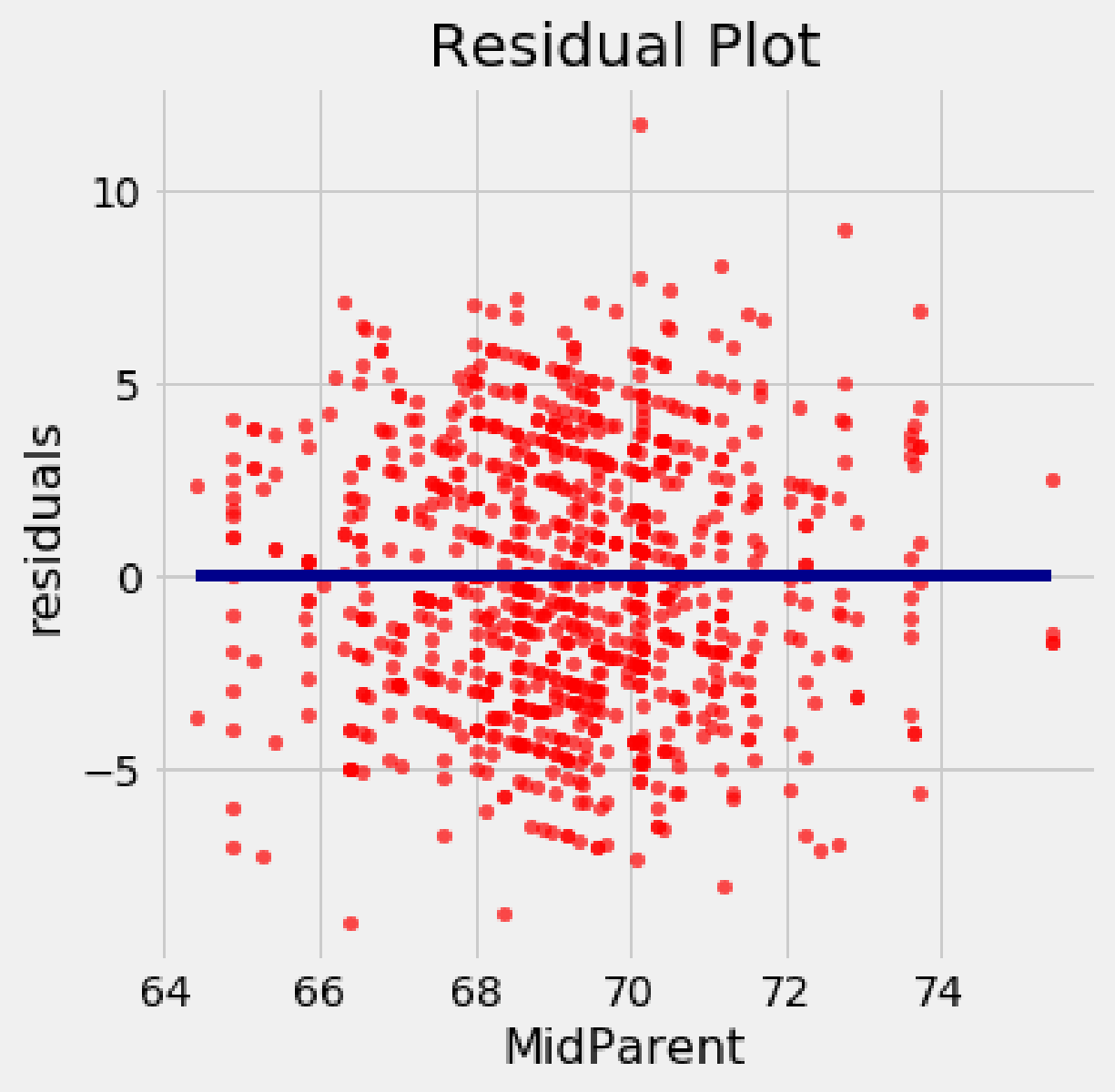
Residuals against the Predictor Variable

```
def residual_plot(table, x, y):
    x_array = table.column(x)
    t = Table().with_columns(
            x, x_array,
            'residuals', residual(table, x, y))
    t.scatter(x, 'residuals', color='r')
    xlims = make_array(min(x_array), max(x_array))
    plots.plot(xlims, make_array(0, 0), color='darkblue', lw=4)
    plots.title('Residual Plot')
```

```
correlation(heights, 'MidParent', 'Residual')
```

```
-2.719689807647064e-16
```

```
residual_plot(heights, 'MidParent', 'Child')
```



```
round(np.mean(heights.column('Residual')), 10)
```

```
0.0
```

$$\text{SD of residuals} = \sqrt{1 - r^2} \cdot \text{SD of } y$$

```
np.std(heights.column('Residual'))
```

```
3.3880799163953426
```

```
r = correlation(heights, 'MidParent', 'Child')
np.sqrt(1 - r**2) * np.std(heights.column('Child'))
```

```
3.388079916395342
```

Another Way to Interpret r

$$\frac{\text{SD of residuals}}{\text{SD of } y} = \sqrt{1 - r^2}$$

$$\frac{\text{variance of fitted values}}{\text{variance of } y} = r^2$$

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

```
0.32094989606395924
```

```
np.std(heights.column('Fitted Value'))/np.std(heights.column('Child'))
```

```
0.32094989606395957
```