lec09

September 20, 2021

[1]:
```
from datascience import *
import numpy as np

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

0.1 Lecture 9

0.2 Functions

[2]:
```
def double(x):
    return x * 2
```

[3]:
```
double(7)
```

[3]:
```
14
```

[4]:
```
double(15/3)
```

[4]:
```
10.0
```

[5]:
```
my_number = 12
```

[6]:
```
double(my_number)
```

[6]:
```
24
```

[7]:
```
double(my_number / 8)
```

[7]:
```
3.0
```

[8]:
```
double(make_array(3, 4, 5))
```

[8]:
```
array([ 6,  8, 10], dtype=int64)
```

[9]:
```
double('data')
```

```
[9]: 'datadata'
```

```
[10]: x
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-10-6fcf9dfbd479> in <module>
----> 1 x

NameError: name 'x' is not defined
```

```
[11]: x = 17
```

```
[12]: double(2)
```

```
[12]: 4
```

```
[13]: x
```

```
[13]: 17
```

```
[14]: double(x)
```

```
[14]: 34
```

```
[15]: x
```

```
[15]: 17
```

```
[16]: def percents(values):
          return np.round(values / sum(values) * 100, 2)
```

```
[17]: percents(make_array(1, 2, 3, 4))
```

```
[17]: array([10., 20., 30., 40.])
```

```
[18]: percents(make_array(1, 4, 30))
```

```
[18]: array([ 2.86, 11.43, 85.71])
```

```
[19]: def percents(values, places):
          return np.round(values / sum(values) * 100, places)
```

```
[20]: percents(make_array(1, 4, 30), 1)
```

```
[20]: array([ 2.9, 11.4, 85.7])
```

0.3 Apply

[21]:
```
ages = Table().with_columns(
    'Person', make_array('A', 'B', 'C', 'D'),
    'Age', make_array(63, 110, 99, 102)
)
ages
```

[21]:
```
Person | Age
A      | 63
B      | 110
C      | 99
D      | 102
```

[22]:
```
def cut_off_at_100(z):
    return min(z, 100)
```

[23]:
```
cut_off_at_100(3)
```

[23]:
```
3
```

[24]:
```
cut_off_at_100(107)
```

[24]:
```
100
```

[25]:
```
cut_age_array = ages.apply(cut_off_at_100, 'Age')
cut_age_array
```

[25]:
```
array([ 63, 100,  99, 100], dtype=int64)
```

[26]:
```
ages.with_column('Cut off ages', cut_age_array)
```

[26]:
```
Person | Age  | Cut off ages
A      | 63   | 63
B      | 110  | 100
C      | 99   | 99
D      | 102  | 100
```

[27]:
```
type(cut_off_at_100)
```

[27]:
```
function
```

0.4 Prediction

[28]:
```
galton = Table.read_table('galton.csv')
galton
```

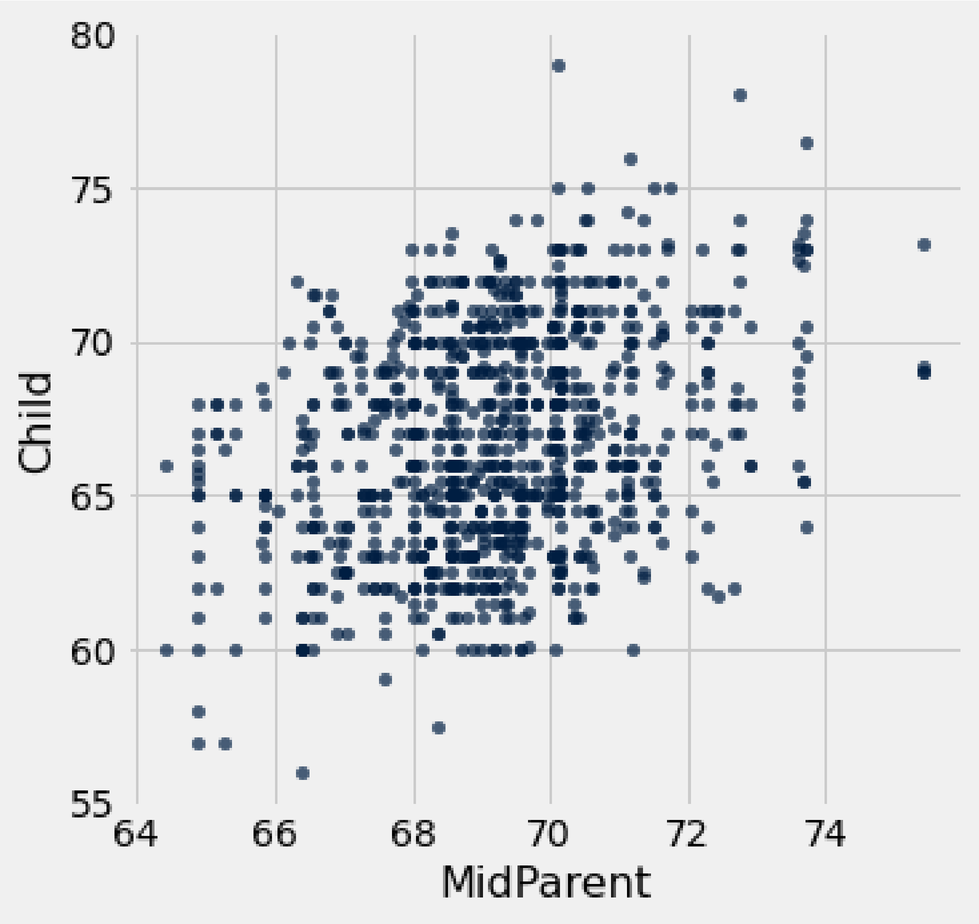
```
[28]: family | father | mother | midparentHeight | children | childNum | gender |
      childHeight
      1      | 78.5   | 67     | 75.43           | 4        | 1        | male   | 73.2
      1      | 78.5   | 67     | 75.43           | 4        | 2        | female | 69.2
      1      | 78.5   | 67     | 75.43           | 4        | 3        | female | 69
      1      | 78.5   | 67     | 75.43           | 4        | 4        | female | 69
      2      | 75.5   | 66.5   | 73.66           | 4        | 1        | male   | 73.5
      2      | 75.5   | 66.5   | 73.66           | 4        | 2        | male   | 72.5
      2      | 75.5   | 66.5   | 73.66           | 4        | 3        | female | 65.5
      2      | 75.5   | 66.5   | 73.66           | 4        | 4        | female | 65.5
      3      | 75     | 64     | 72.06           | 2        | 1        | male   | 71
      3      | 75     | 64     | 72.06           | 2        | 2        | female | 68
      … (924 rows omitted)
```

```
[29]: heights = galton.select(3, 7).relabeled(0, 'MidParent').relabeled(1, 'Child')
```

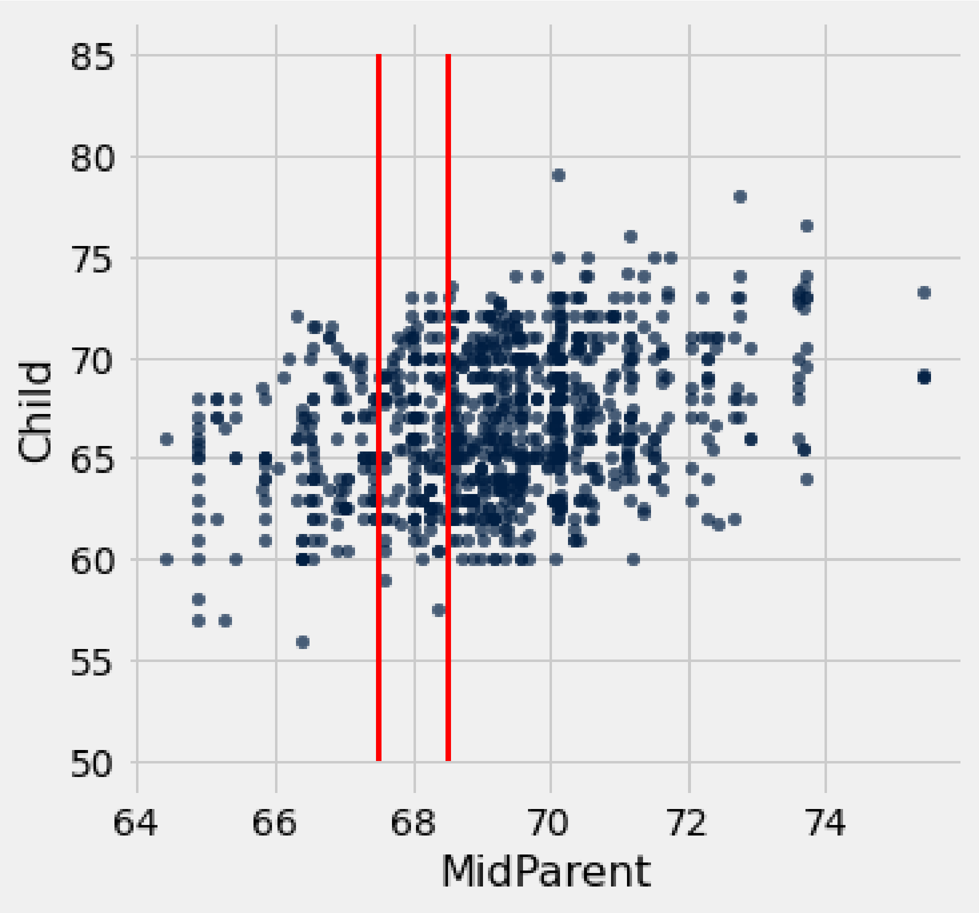
```
[30]: heights
```

```
[30]: MidParent | Child
      75.43     | 73.2
      75.43     | 69.2
      75.43     | 69
      75.43     | 69
      73.66     | 73.5
      73.66     | 72.5
      73.66     | 65.5
      73.66     | 65.5
      72.06     | 71
      72.06     | 68
      … (924 rows omitted)
```

```
[31]: heights.scatter('MidParent', 'Child')
```



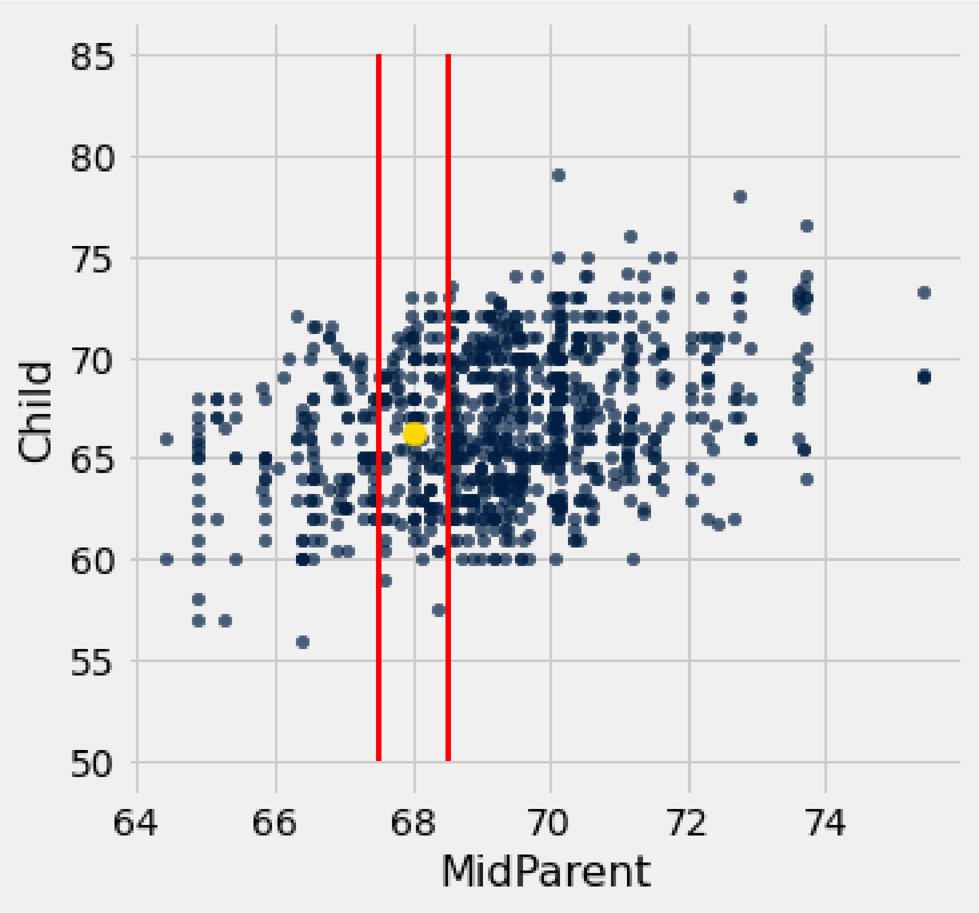
```
[32]: heights.scatter('MidParent', 'Child')
      plots.plot([67.5, 67.5], [50, 85], color='red', lw=2)
      plots.plot([68.5, 68.5], [50, 85], color='red', lw=2);
```



[33]:
```
nearby = heights.where('MidParent', are.between(67.5, 68.5))
nearby.column('Child').mean()
```

[33]:
```
66.24045801526718
```

[34]:
```
heights.scatter('MidParent', 'Child')
plots.plot([67.5, 67.5], [50, 85], color='red', lw=2)
plots.plot([68.5, 68.5], [50, 85], color='red', lw=2)
plots.scatter(68, 66.24, color='gold', s=75);
```



```
[35]: def predict_child(h):
          nearby = heights.where('MidParent', are.between(h-0.5, h+0.5))
          return nearby.column('Child').mean()
```

```
[36]: predict_child(68)
```

```
[36]: 66.24045801526718
```

```
[37]: predict_child(65)
```

```
[37]: 64.22962962962963
```

```
[38]: predictions = heights.apply(predict_child, 'MidParent')
```

```
[39]: heights = heights.with_column('Child Prediction', predictions)
```

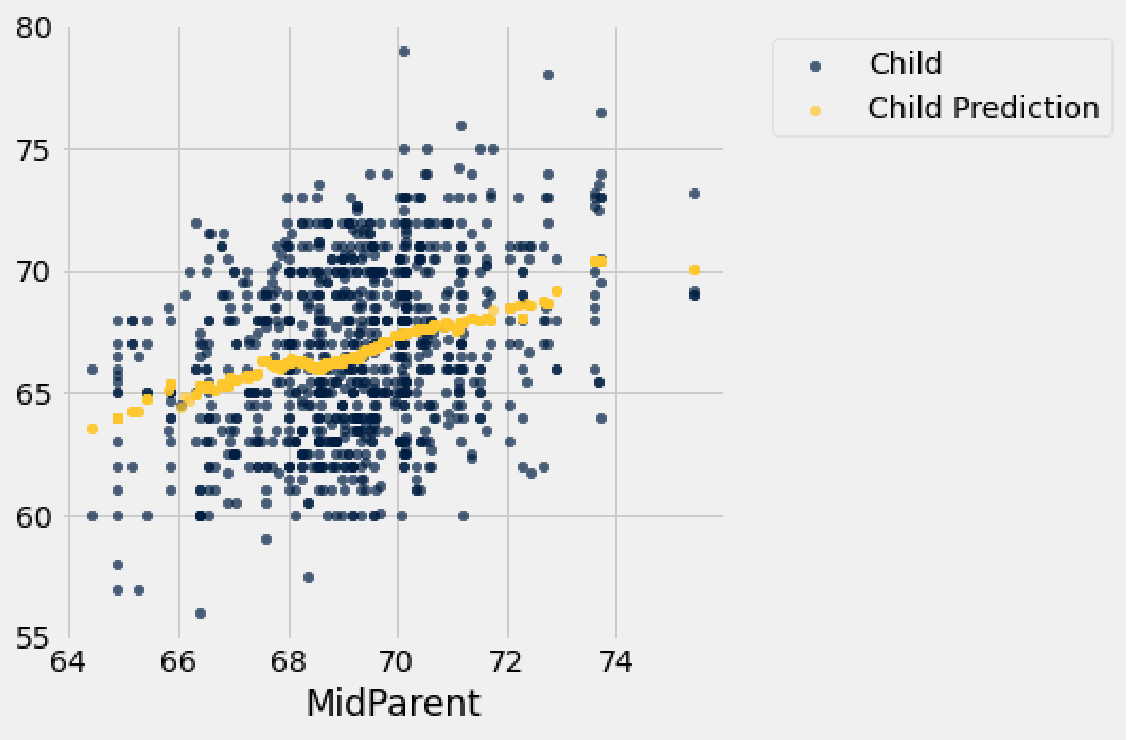
```
[40]: heights
```

[40]:

```
MidParent | Child | Child Prediction
75.43     | 73.2  | 70.1
75.43     | 69.2  | 70.1
75.43     | 69    | 70.1
75.43     | 69    | 70.1
73.66     | 73.5  | 70.4158
73.66     | 72.5  | 70.4158
73.66     | 65.5  | 70.4158
73.66     | 65.5  | 70.4158
72.06     | 71    | 68.5025
72.06     | 68    | 68.5025
… (924 rows omitted)
```

[41]:

```
heights.scatter('MidParent')
```



[ ]:

[ ]: