DSFA

Spring 2018

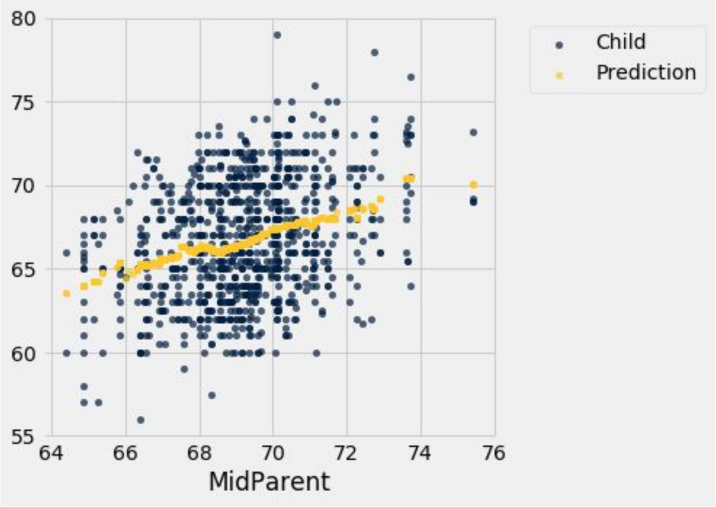
Lecture 31

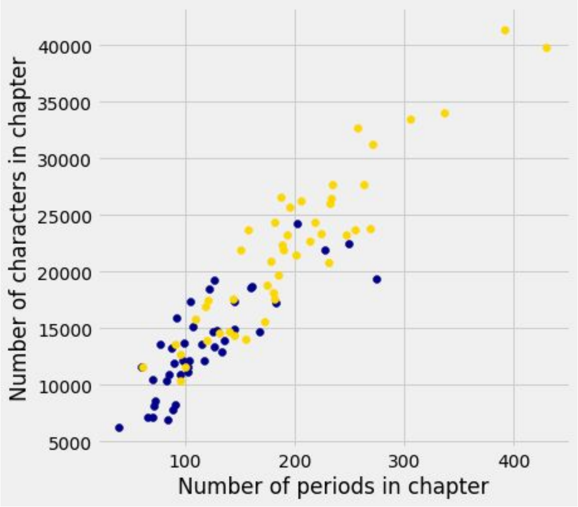
Least Squares

Announcements

Prediction

If we have a line describing the relation between two variables, we can make predictions



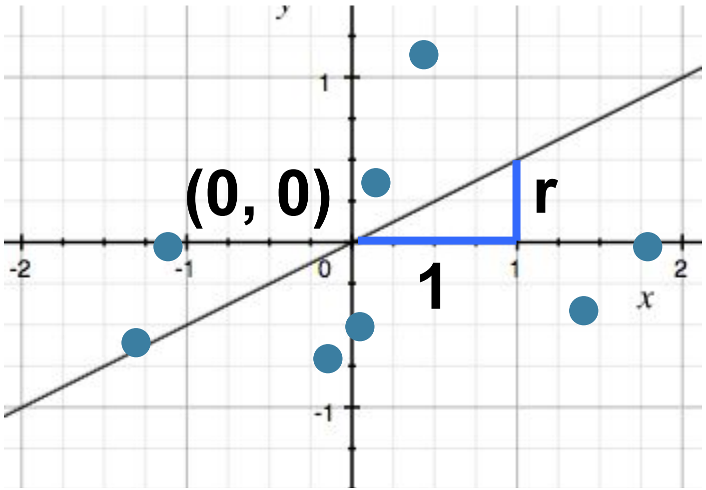


Prediction

- **Problem:** given a known *x* value, predict *y*, where both are in standard units
- **Solution:**
  - Compute correlation coefficient *r*
  - Predict that $y = r * x$
- Why is that a line? (slope = *r*, intercept = 0)
- Why use that line?
  - It is a version of the graph of averages, smoothed to a line

Regression Line Equation

In standard units, the equation of the regression line is:



Fitted value

Observed value

$$y_{(\text{su})} = r \times x_{(\text{su})}$$

Correlation coefficient

Regression Line Equation

In original units, the regression line has this equation:

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

y in standard units

x in standard units

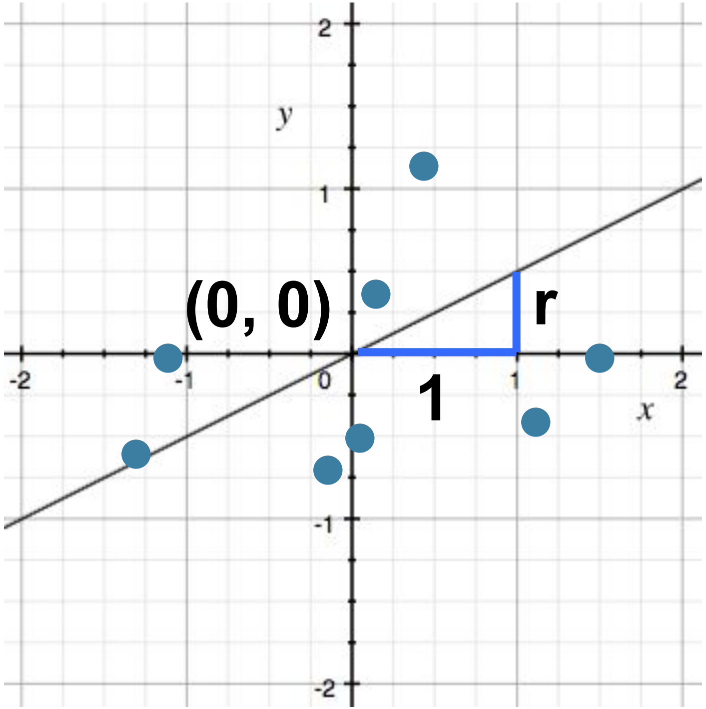
$$y = \text{slope} \times x + \text{intercept}$$

$$\textbf{slope of the regression line} \;=\; r \cdot \frac{\text{SD of } y}{\text{SD of } x}$$

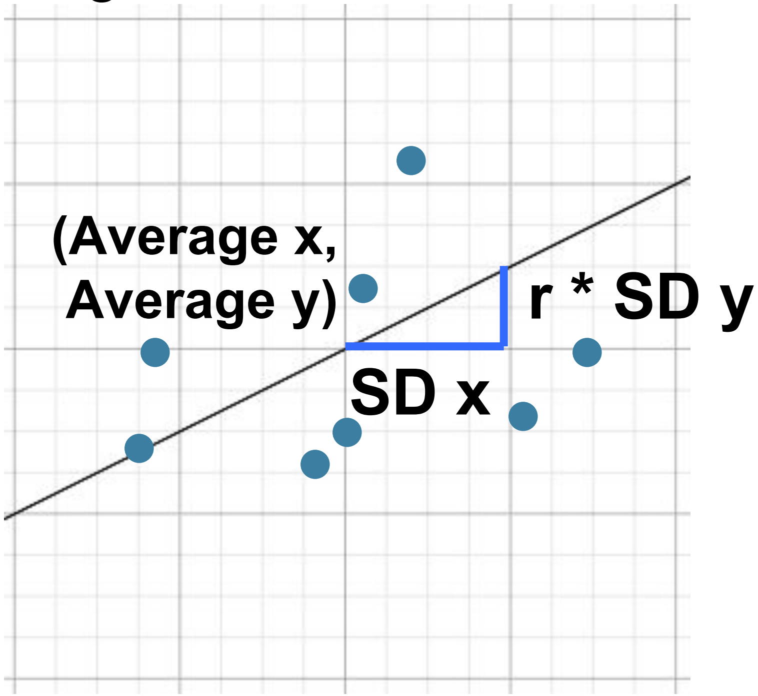
$$\textbf{intercept of the regression line} \;=\; \text{average of } y \;-\; \text{slope} \cdot \text{average of } x$$

Regression Line

Standard Units



Original Units



(Demo)

Abuses of *r*

- Summarizing non-linear data with *r*
- Eliminating outliers to “improve” *r*
- Drawing conclusions about individuals based on data about groups (*ecological* correlations)
- Jumping to conclusions about causality

Correlation is not causation

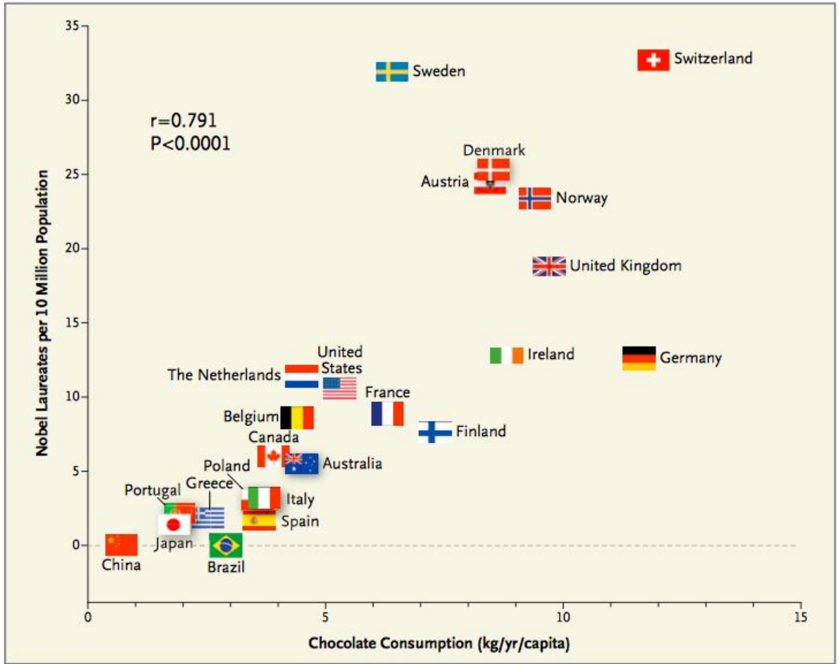
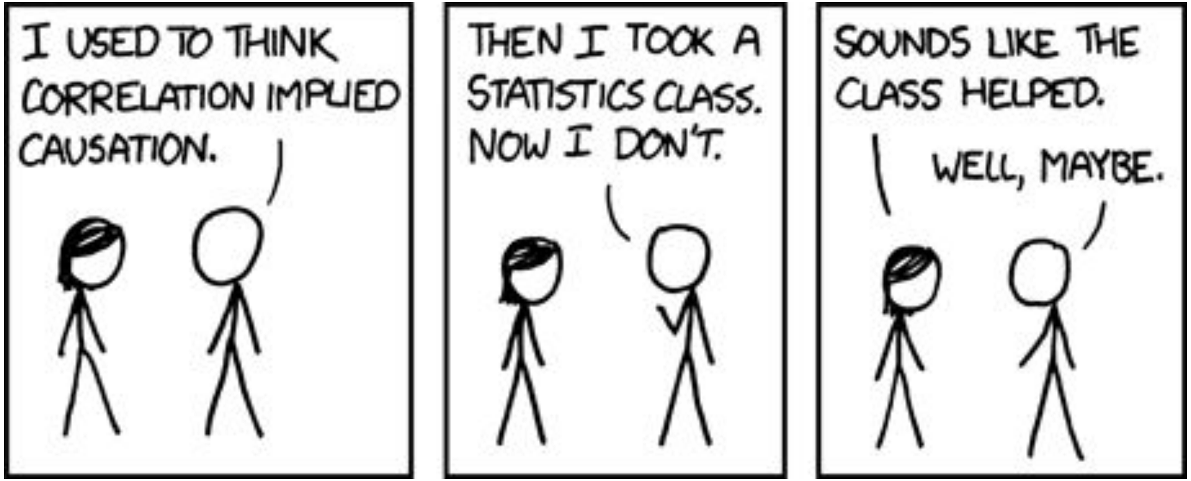


Figure 1. Correlation between Countries' Annual Per Capita Chocolate Consumption and the Number of Nobel Laureates per 10 Million Population.

Correlation is not causation



Quantifying Error

Error in Prediction

- How good is the regression line at making predictions?
  - Hard to say for unknown data
  - But easy for data we already have

- **error = actual value − prediction**

(Demo)

Error in Prediction

- How good is the regression line at making predictions?
  - Hard to say for unknown data
  - But easy for data we already have

- **error = actual value − prediction**
- RMSE = root mean square error
  4 3 2 1
- RMSE = root mean square of deviation from prediction
  5 4 3 2 1

(Demo)

RMSE

RMSE = root mean square error

RMSE = std(y) * sqrt(1 - r^2)

- If r = 1, what is RMSE? 0
- If r = 0, what is RMSE? std(y)

Compare regression line to other lines using RMSE...

(Demo)

Line with smallest RMSE?

- SciPy function `minimize(f)` returns the value `x` that produces the minimum output `f(x)` from `f`
- Also works for functions that make multiple arguments
- How to use to find best line:
  - Write function `rmse(a, b)` that returns the RMSE for line with slope `a` and intercept `b`
  - Call `minimize(rmse)` and get output array $[a_0, b_0]$
  - a_0 is slope and b_0 intercept of line that minimizes RMSE

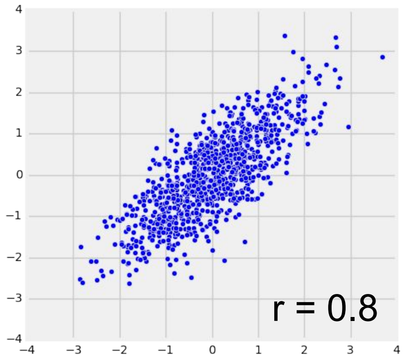
(Demo)

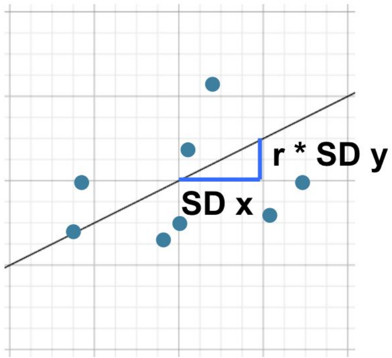
Regression line

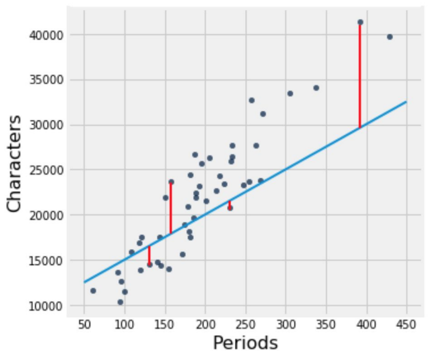
- Regression line has the minimum RMSE of all lines
- Names:
  - Regression line
  - Least squares line
  - “Best fit” line

What we can learn from *r*

- How clustered points are around a line
- How *y* depends on *x*
- How accurate linear regression predictions will be







Non-linear regression

- Minimization technique works to fit curves as well as lines

(Demo)