How to learn MATLAB:

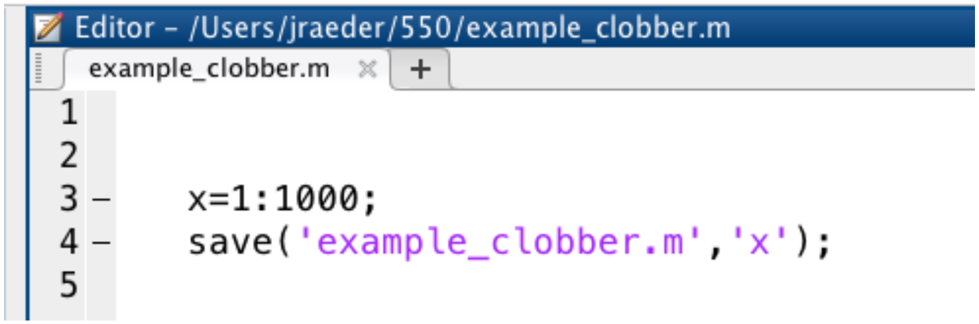
```
while ~understood

  pain();

  ((lookup_Google) || (ask_classmate) ...
    || (ask_TA) || (ask_prof)) ...
    && pray && (try_again)

   if code_runs(); break; end
   take_break(5);

end

printf("YEAH !!!!\n";

function take_break(minutes);
  make_coffee();
  for i=1:minutes;
      sip_coffee();
  end
end
```

Announcements

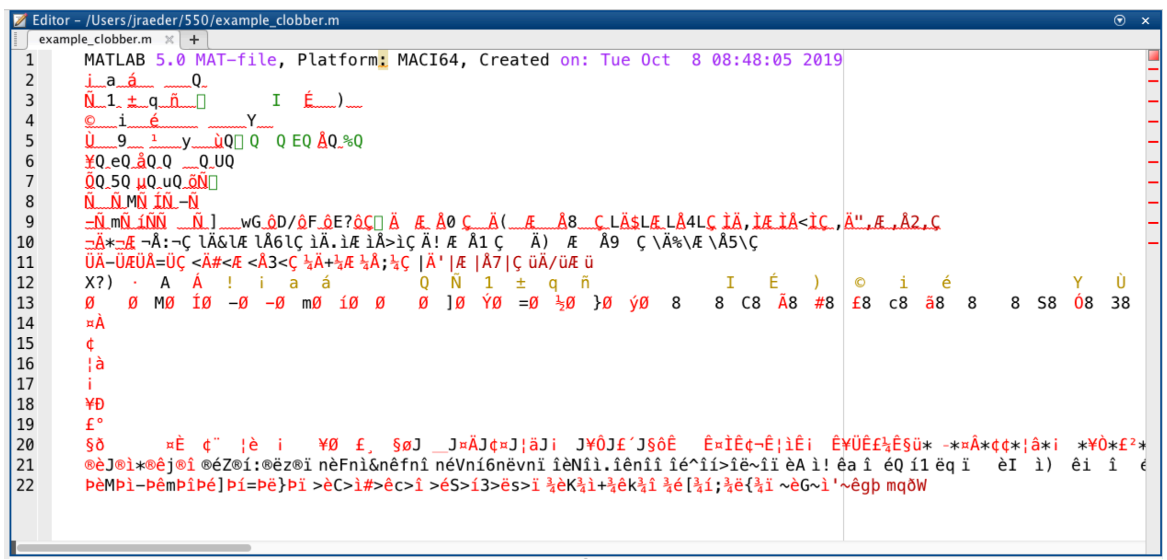
- It’s half time! (#13 out of 26 classes)
- Midterm:
  - Next week Thursday
  - During class time
  - 1h in-class exam
  - No books, computers, notes, etc.
  - In N108 and probably another room
- A new HW (way easier!) this Thursday.

A particularly bad experience

Clobber your script by saving data:



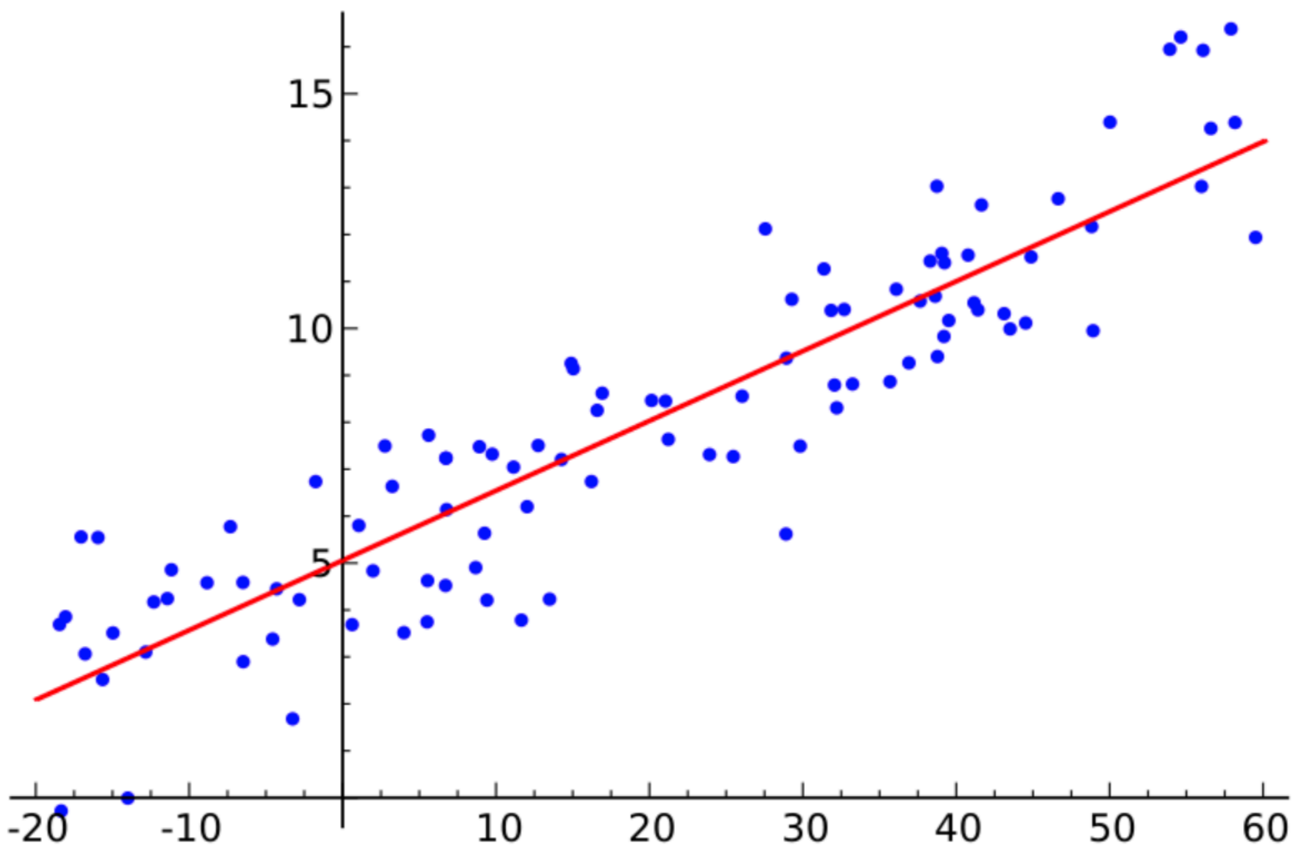
Nothing bad seems to happen when I run it (MATLAB should give a warning), but if I load the script again I get this (gibberish from a binary file):



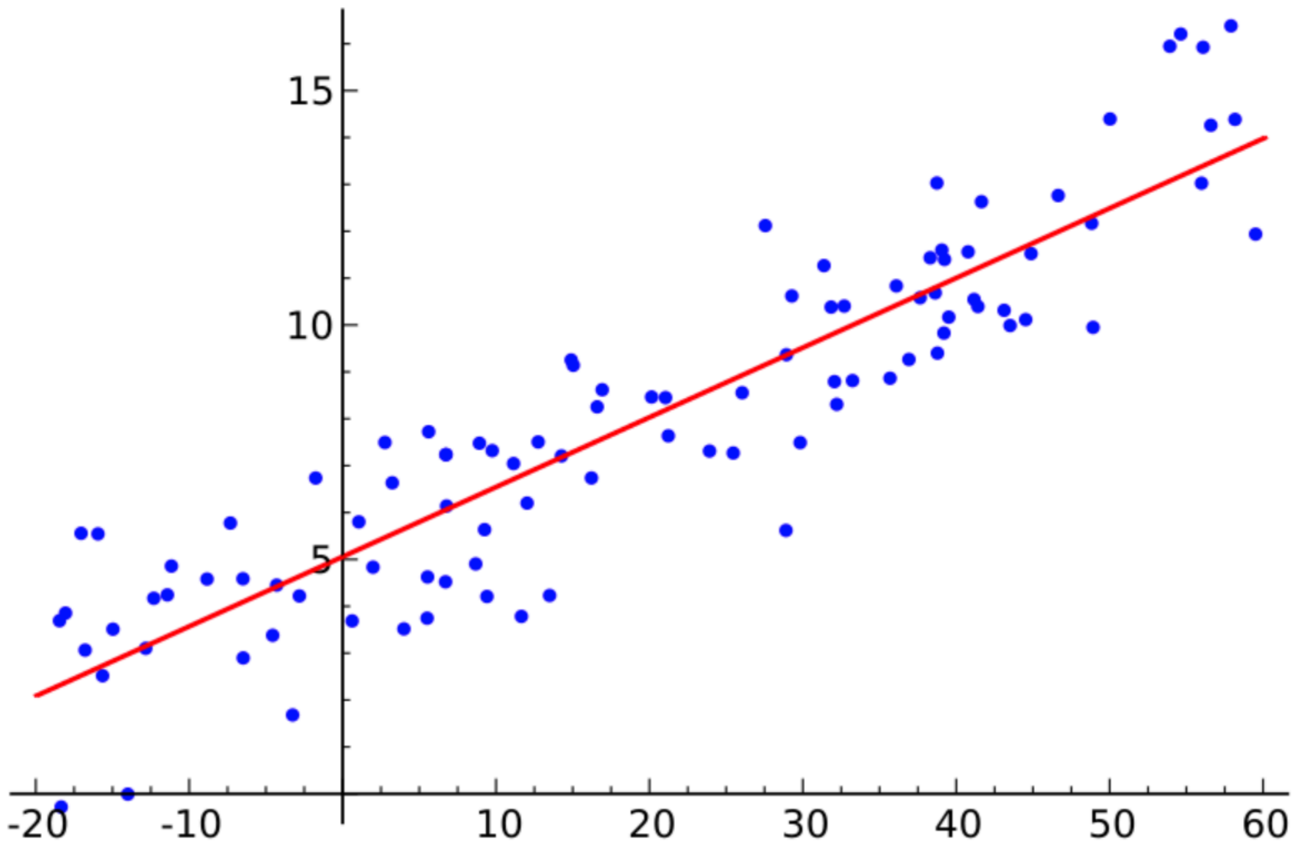
Use a distinct name for your data files, and never give it a '.m' extension. With no extension given in the save() command, MATLAB will add '.mat'

Linear regression

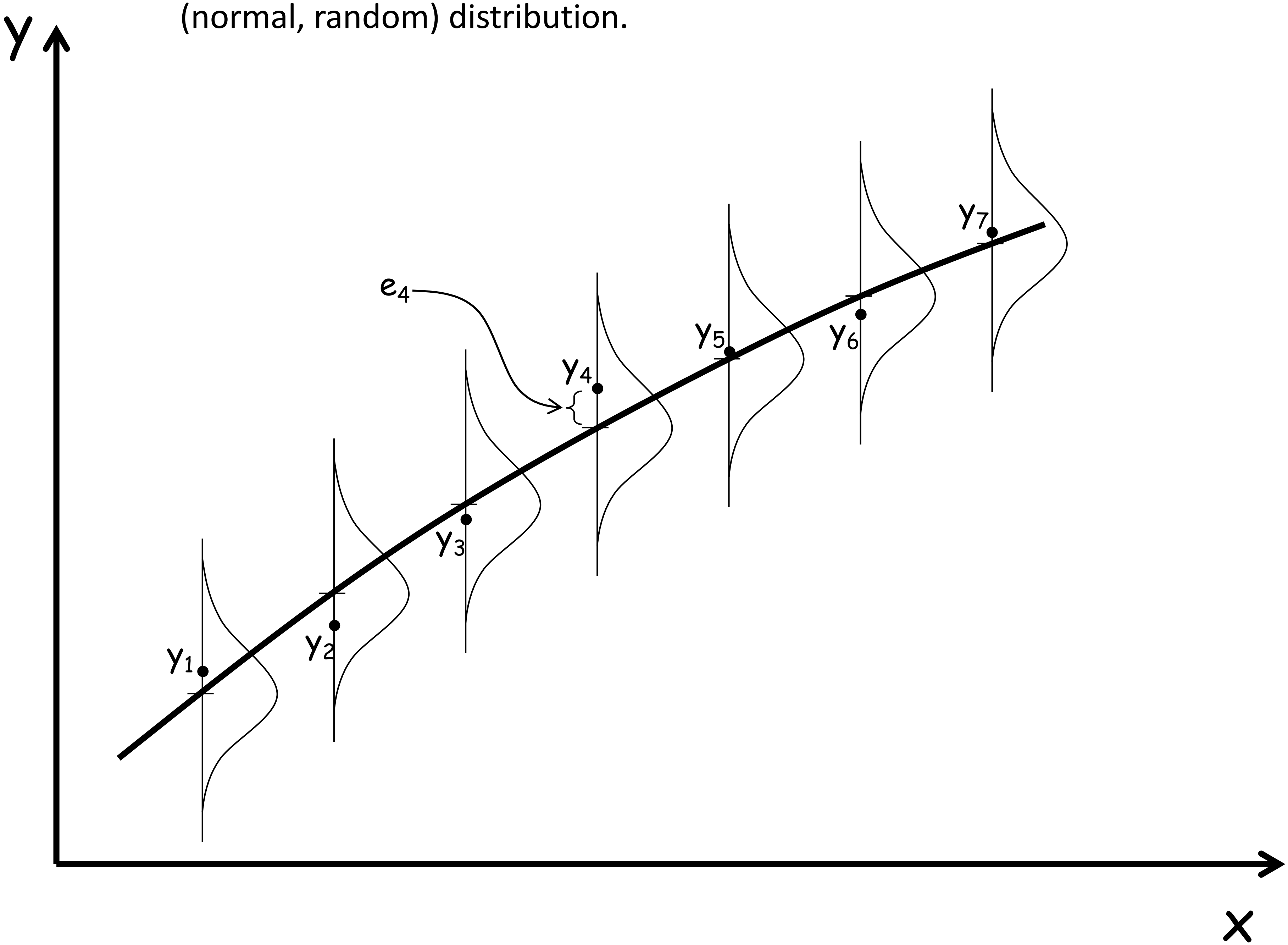
IAM 550, Fall 2019, lecture 13, 10/8/2019, J. Raeder



- Given a data set of one independent and one dependent variable.
- For example: position vs time x(t), stock price P(t), deflection versus force F(x), temperature versus time, etc.
- There are reasons to believe that there is a linear relationship between the dependent and independent variable: $y(x)=a_0+a_1x$
- But the dependent variable data are noisy.
- We want to find the 'best' fit to the data: linear regression, linear fit, or trend line.



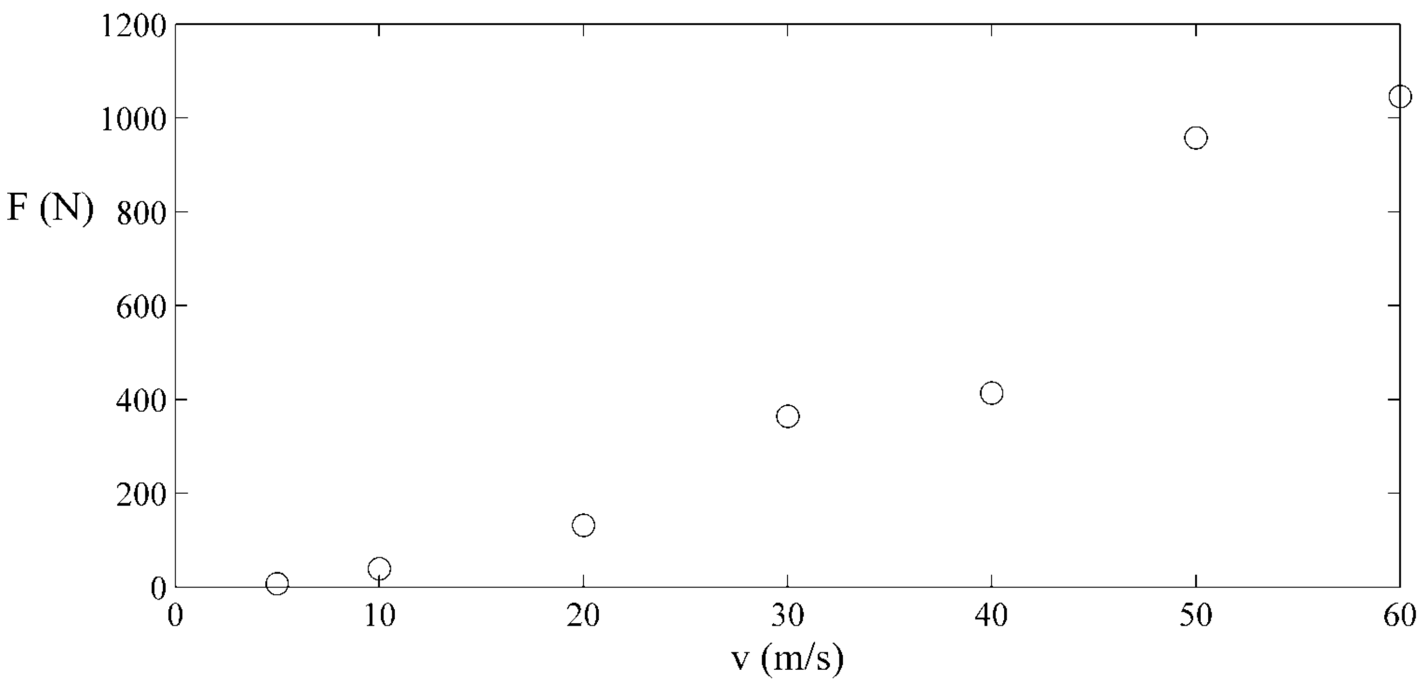
The noise is only in the y data. We also assume that the noise has a Gaussian (normal, random) distribution.



Observations of the force of air resistance on a skier in a wind tunnel

http://www.firsttracksonline.com/2011/10/11/canadian-ski-racers-train-in-wind-tunnel/

$$F = c_d v^2$$



Do the data match our expectation?

What is the coefficient of drag?

Linear Least-Squares Regression

Objective: fit a curve (a line) to our data

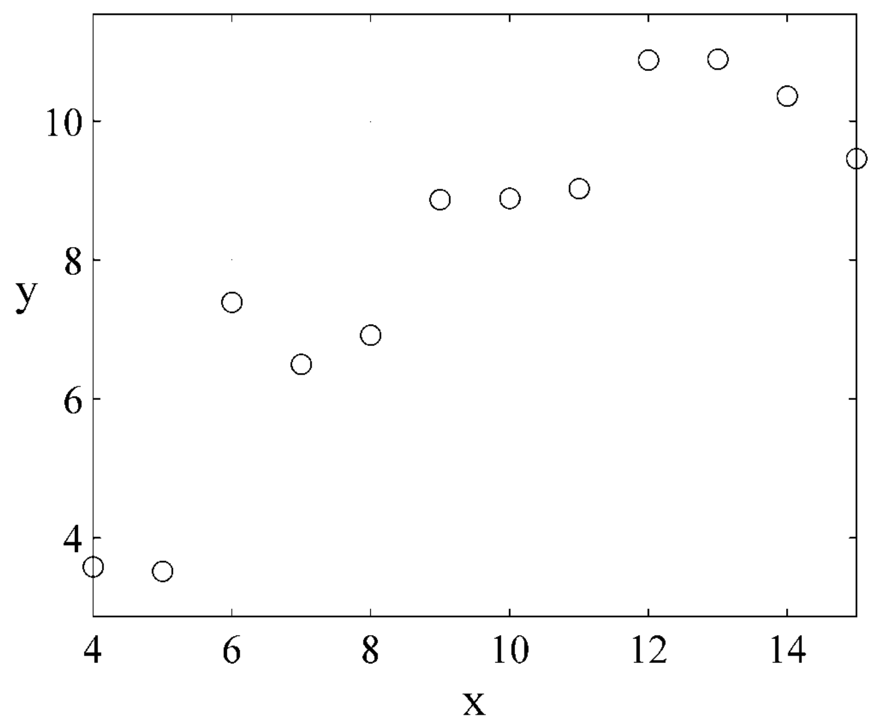
x	y
4.00	3.58
5.00	3.52
6.00	7.39
7.00	6.50
8.00	6.92
9.00	8.87
10.00	8.89
11.00	9.03
12.00	10.88
13.00	10.89
14.00	10.36
15.00	9.46

unknowns

$$y = a_o + a_1 x + e$$

observations

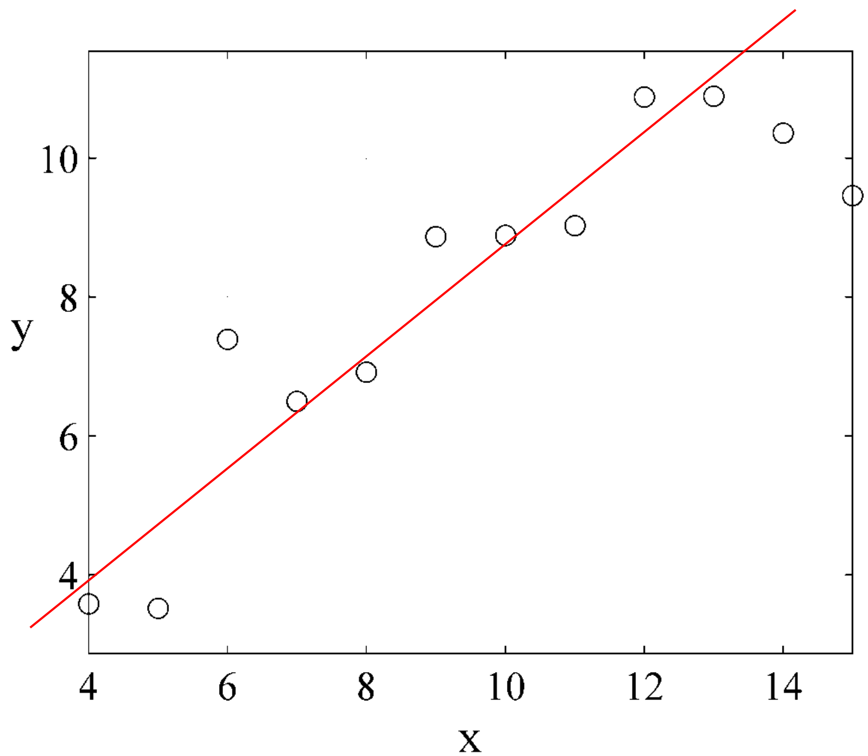
error in our observations



Linear Least-Squares Regression

Objective: fit a curve (a line) to our data

$$y = a_o + a_1 x + e \xrightarrow{\text{rearrange}} e = y - a_o - a_1 x$$



Goal: find a_o and a_1 such that e is minimized.

Note: a_o is the y-intercept and a_1 is the slope for the 'best-fit' line through the data.

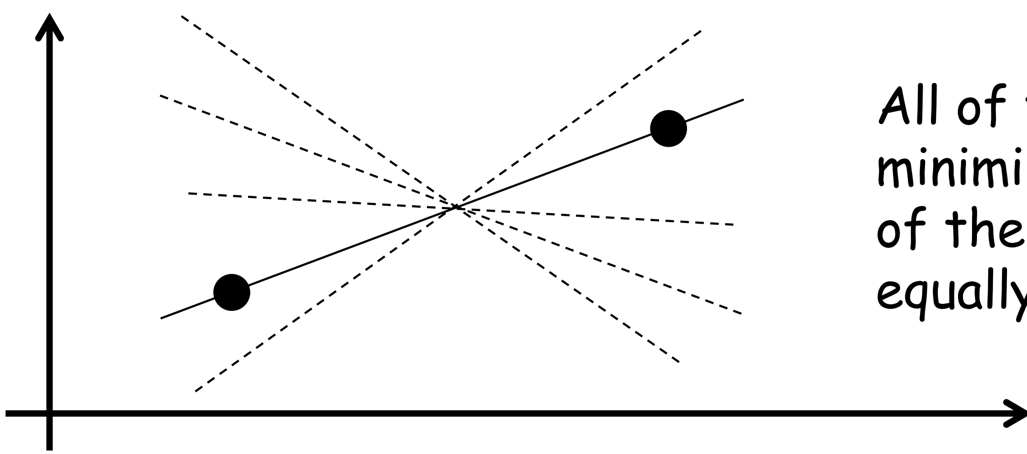
Linear Least-Squares Regression

Objective: fit a curve (a line) to our data

$$e = y - a_o - a_1 x$$

Strategy 1: Minimize the sum of the errors

$$\sum_{i=1}^{N} e_i = \sum_{i=1}^{N} (y_i - a_0 - a_1 x)$$



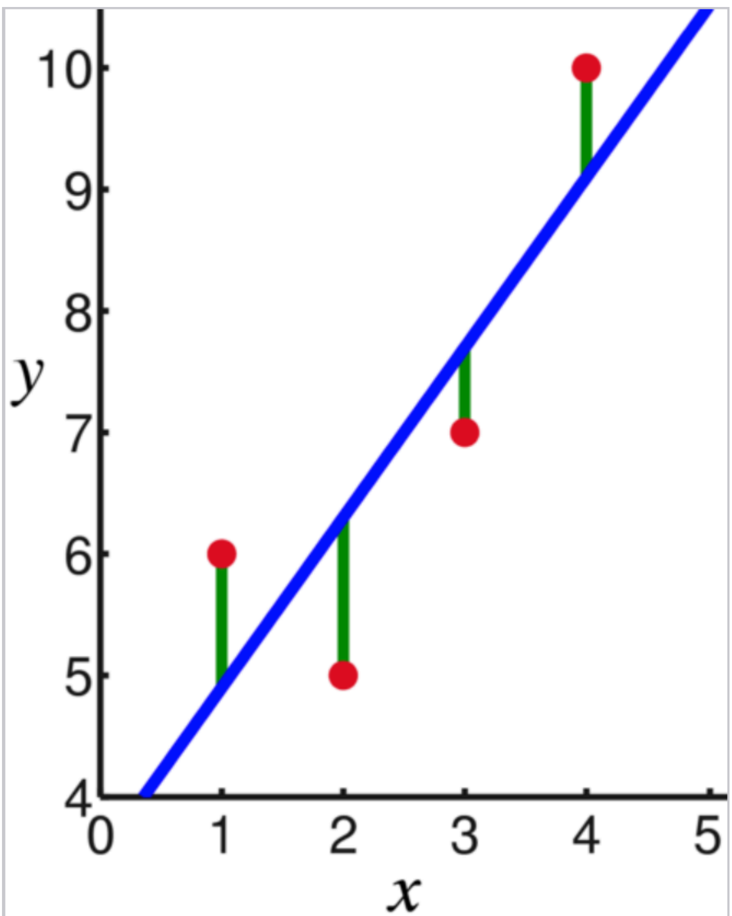
All of these lines minimize the sum of the errors equally!

Strategy 1 doesn't work!

Linear Least-Squares Regression

Strategy 2: Minimize the sum of squares of the errors

$$\sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (y_i - a_0 - a_1 x)^2$$



Squaring the errors removes the sign difference – it not longer matters whether the line fit is above or below the data – just how far away it is.

Strategy 2 should work better

Linear Least-Squares Regression

Objective: fit a curve (a line) to our data

<u>To find the best-fit line</u>: figure out what a_o and a_1 such that the sum of the squared errors is its smallest (least) value:

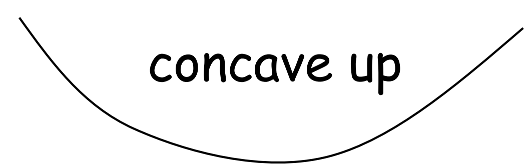
$$\sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (y_i - a_0 - a_1 x)^2$$

expand

$$F(a_0, a_1) =$$

$$(y_i - a_0 - a_1 x)^2 = y_i^2 + a_o^2 + a_1^2 x^2 - 2y_i a_o + 2a_o a_1 x_i - 2a_1 y_i x_i$$

The 2nd derivative of this entire eqn w/respect to a_o and a_1 will always be positive



We need to find this minimum point

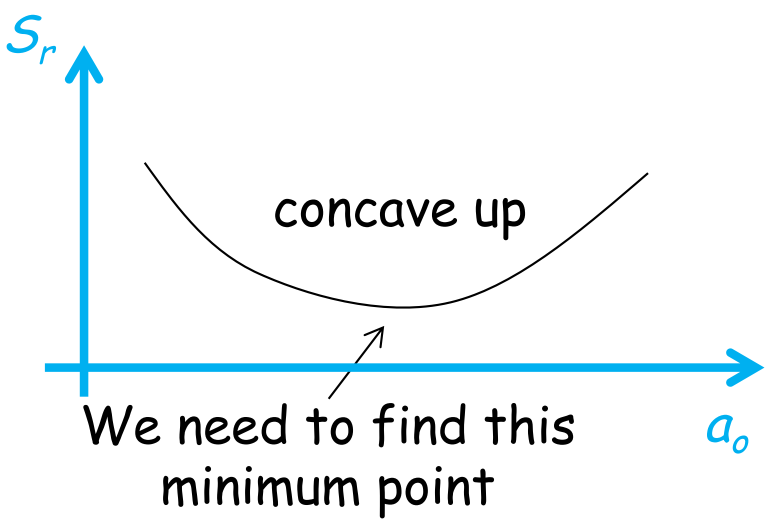
Linear Least-Squares Regression

Objective: fit a curve (a line) to our data

<u>To find the best-fit line</u>: figure out what a_o and a_1 such that the sum of the squared errors is its smallest (least) value:

$$\sum_{i=1}^{N} e_i^2 = \sum_{i=1}^{N} (y_i - a_0 - a_1 x)^2$$

Define: $$s_r = \sum_{i=1}^{N} (y_i - a_0 - a_1 x)^2$$



$$\frac{\partial s_r}{\partial a_0} = \sum_{i=1}^{N} [2a_0 - 2y_i + 2a_1 x] = 0$$

Linear Least-Squares Regression

Objective: fit a curve (a line) to our data

Look for minimum of Sr w/respect to a_o:
(previous slide)

$$\frac{\partial s_r}{\partial a_0} = \sum_{i=1}^{N} [2a_0 - 2y_i + 2a_1 x_i] = 0$$

$$(N)a_0 + \left(\sum_{i=1}^{N} x_i\right) a_1 = \sum_{i=1}^{N} y_i$$

2 eqns, 2 unknowns

Look for minimum of Sr w/respect to a_1:
(similar to previous slide)

$$\frac{\partial s_r}{\partial a_1} = \sum_{i=1}^{N} [2a_0 x_i - 2y_i x_i + 2a_1 x_i^2] = 0$$

$$\left(\sum_{i=1}^{N} x_i\right) a_0 + \left(\sum_{i=1}^{N} x_i^2\right) a_1 = \sum_{i=1}^{N} x_i y_i$$

Linear Least-Squares Regression

Objective: fit a curve (a line) to our data

Solve for a_o and a_1

$$(N)a_0 + \left(\sum_{i=1}^{N} x_i\right)a_1 = \sum_{i=1}^{N} y_i$$

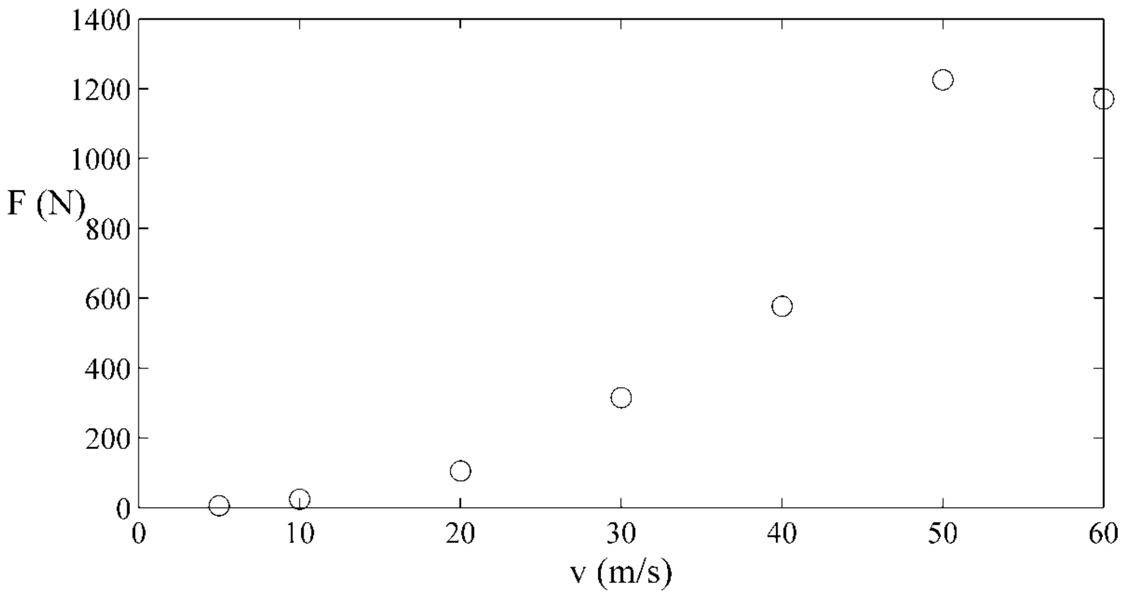
$$\left(\sum_{i=1}^{N} x_i\right)a_0 + \left(\sum_{i=1}^{N} x_i^2\right)a_1 = \sum_{i=1}^{N} x_i y_i$$

$$a_1 = \frac{N\sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}$$

$$a_o = \frac{1}{N}\sum_{i=1}^{N} y_i - \frac{a_1}{N}\sum_{i=1}^{N} x_i$$

Linear Least-Squares Regression

Example



Velocity (m/s)	Force (N)
5.00	5.66
10.00	24.16
20.00	105.53
30.00	315.51
40.00	577.42
50.00	1225.83
60.00	1170.81

http://www.firsttracksonline.com/2011/10/11/canadian-ski-racers-train-in-wind-tunnel/

Objective: Find a best-fit line to these data.

Linear Least-Squares Regression

Example

Velocity (m/s)	Force (N)
5.00	5.66
10.00	24.16
20.00	105.53
30.00	315.51
40.00	577.42
50.00	1225.83
60.00	1170.81

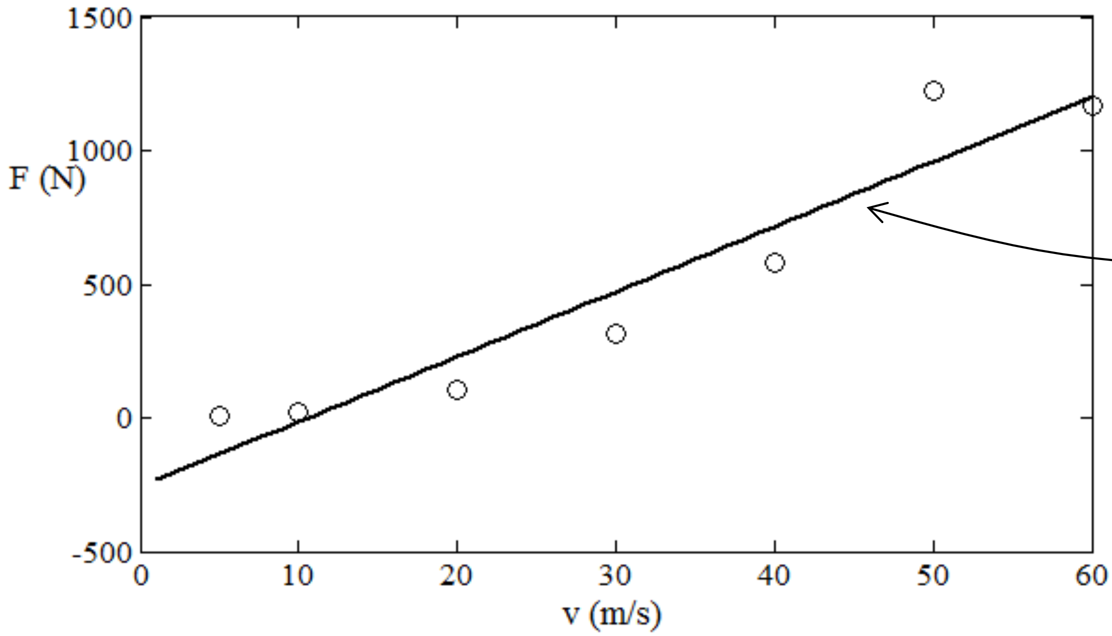
x→ Velocity (m/s)
y → Force (N)
N = 7

$$a_1 = \frac{N\sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2} = \frac{1165376.8 - 736357.8}{63875.0 - 46225.0} = 24.3$$

$$a_o = 489.27 — 746.6 = -257.3$$

Linear Least-Squares Regression

Example



$$y = a_o + a_1 x$$

$$a_1 = 24.3$$

$$a_o = -257.3$$

Question: how good is our fit?

Linear Least-Squares Regression

Goodness of fit

Recall that we had previously defined our 'residuals' about the line fit:

$$s_r = \sum_{i=1}^{N} (\underbrace{y_i}_{\text{The data}} - \underbrace{a_0 - a_1 x}_{\text{Our best-fit line}})^2$$

We could also simply examine the variance of the data:

$$\hat{\sigma}_y^2 = \frac{1}{N-1} \sum_{i=1}^{N} (y_i - \bar{y})^2$$

Key point: We can fit a line to any set of data, whether there is a linear trend or not. If there is a linear trend, then we would expect there to be a difference between the residuals about the line fit and the variance of the data.

Normalized goodness of fit:

$$r^2 = \frac{\hat{\sigma}_y^2 - s_r}{\hat{\sigma}_y^2}$$

Linear Least-Squares Regression

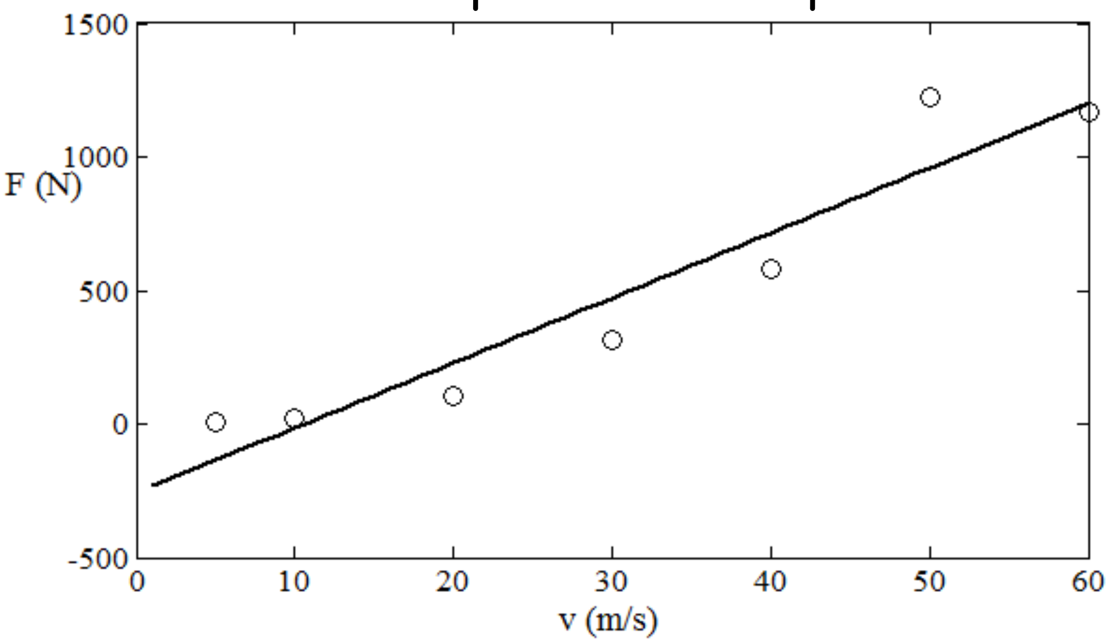
Goodness of fit

coefficient of determination: $$r^2 = \frac{\hat{\sigma}_y^2 - s_r}{\hat{\sigma}_y^2}$$ $$0 \leq r^2 \leq 1$$

A perfect fit

Note: we call r the correlation coefficient

Our previous example



$$y = a_o + a_1 x$$

$$a_1 = 24.3$$

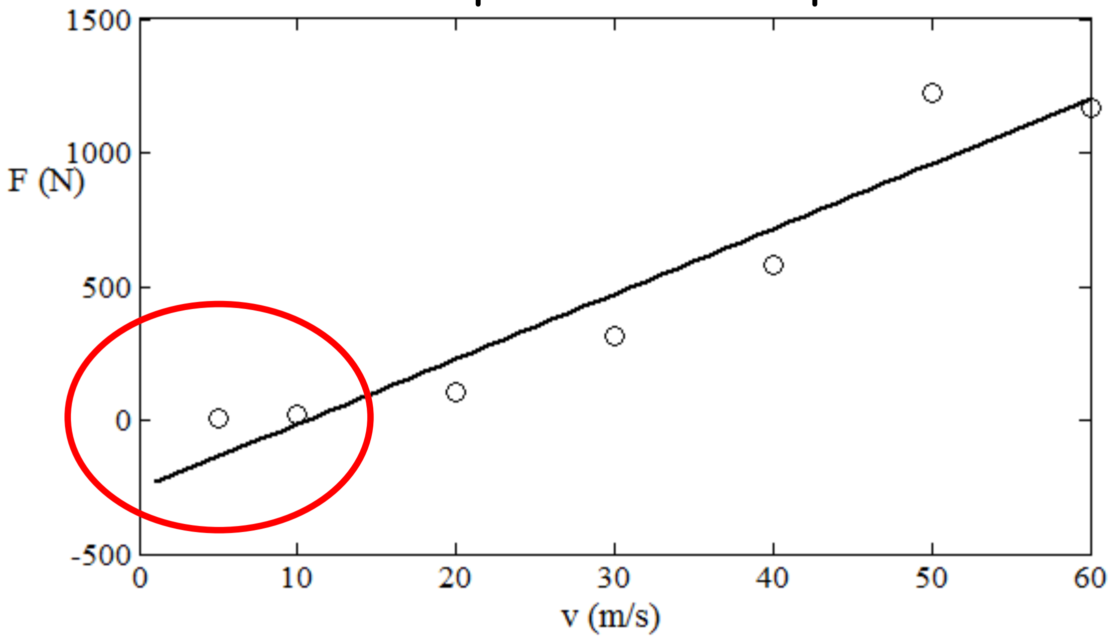
$$a_o = -257.3$$

r^2=0.44

Linear Least-Squares Regression

What if we want to fit a different curve (i.e. something that is not a line)

Our previous example



$$y = a_o + a_1 x$$

$$a_1 = 24.3$$

$$a_o = -257.3$$

r^2=0.44

Does a negative drag force make sense?

http://www.firsttracksonline.com/2011/10/11/canadian-ski-racers-train-in-wind-tunnel/

Should we really be trying to fit a line to our data?

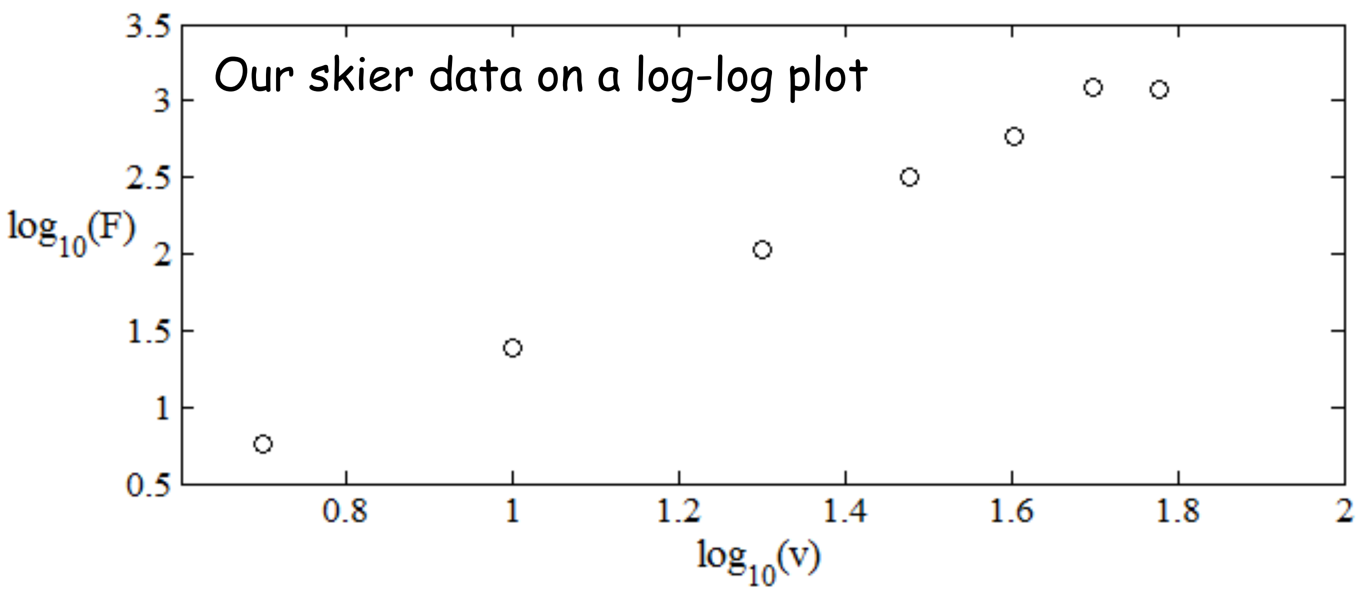
$$F = c_d v^2$$

Linear Least-Squares Regression

What if we want to fit a different curve (i.e. something that is not a line)

Take the base-10 log of this equation: $F = c_d v^2$

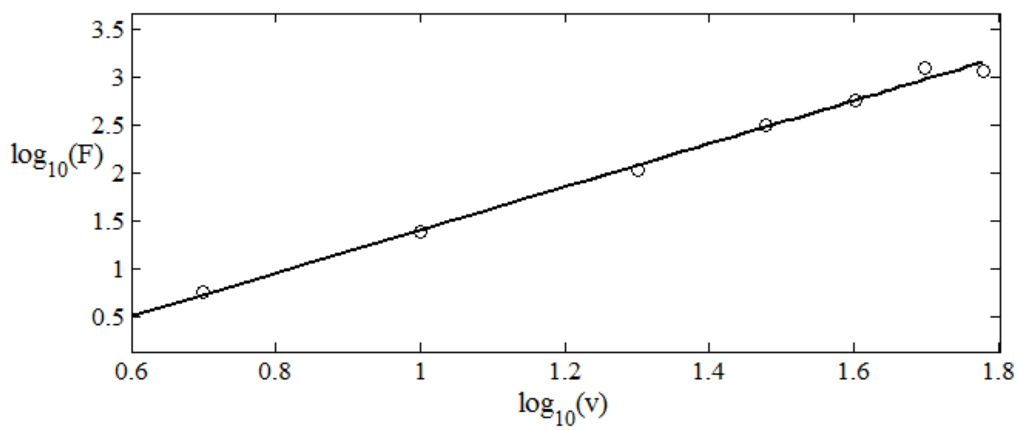
$$\underbrace{log_{10}[F]}_{y} = log_{10}[c_d v^2] = \underbrace{log_{10}[c_d]}_{a_o} + \underbrace{2log_{10}[v]}_{a_1 x}$$



We can use the same framework we already defined for linear least squares, we just take the base-10 log of x and y first.

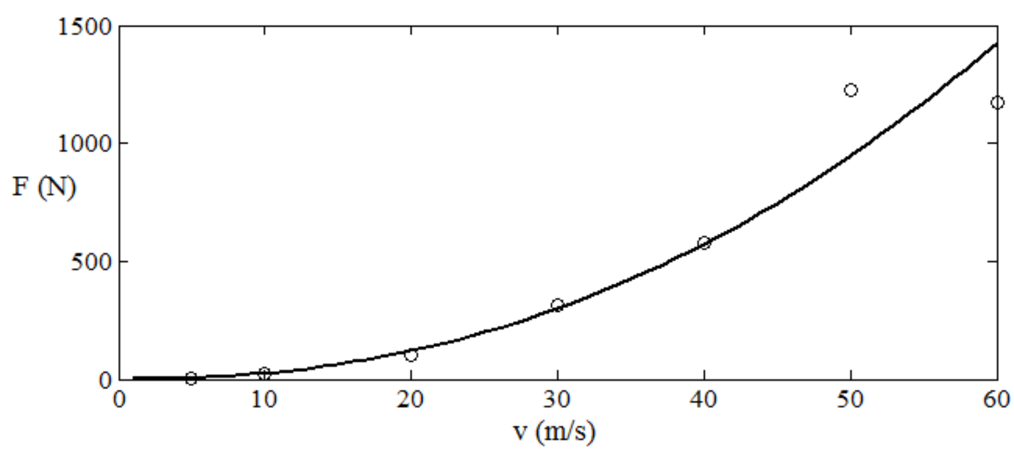
Linear Least-Squares Regression

What if we want to fit a different curve (i.e. something that is not a line)



Still the skier's data: much better fit. Also, it's not really the second power of the velocity, but v to the 2.25 power!

$a_1 = 2.25$
$a_o = -0.84$
$r^2 = 0.97$



The more general case of a power law:

$$y = Ax^B$$

$$A = 10^{a_o}$$

$$B = a_1$$

More generalizations

- Simple linear relation:

$$y(x) = a_0 + a_1 x$$

- A power law: take log of x and y first, then fit the line:

$$y(x) = a_0 x^{a_1} \rightarrow log(y(x)) = log(a_0) + a_1 log(x)$$

- A exponential law: take log first, then fit the line:

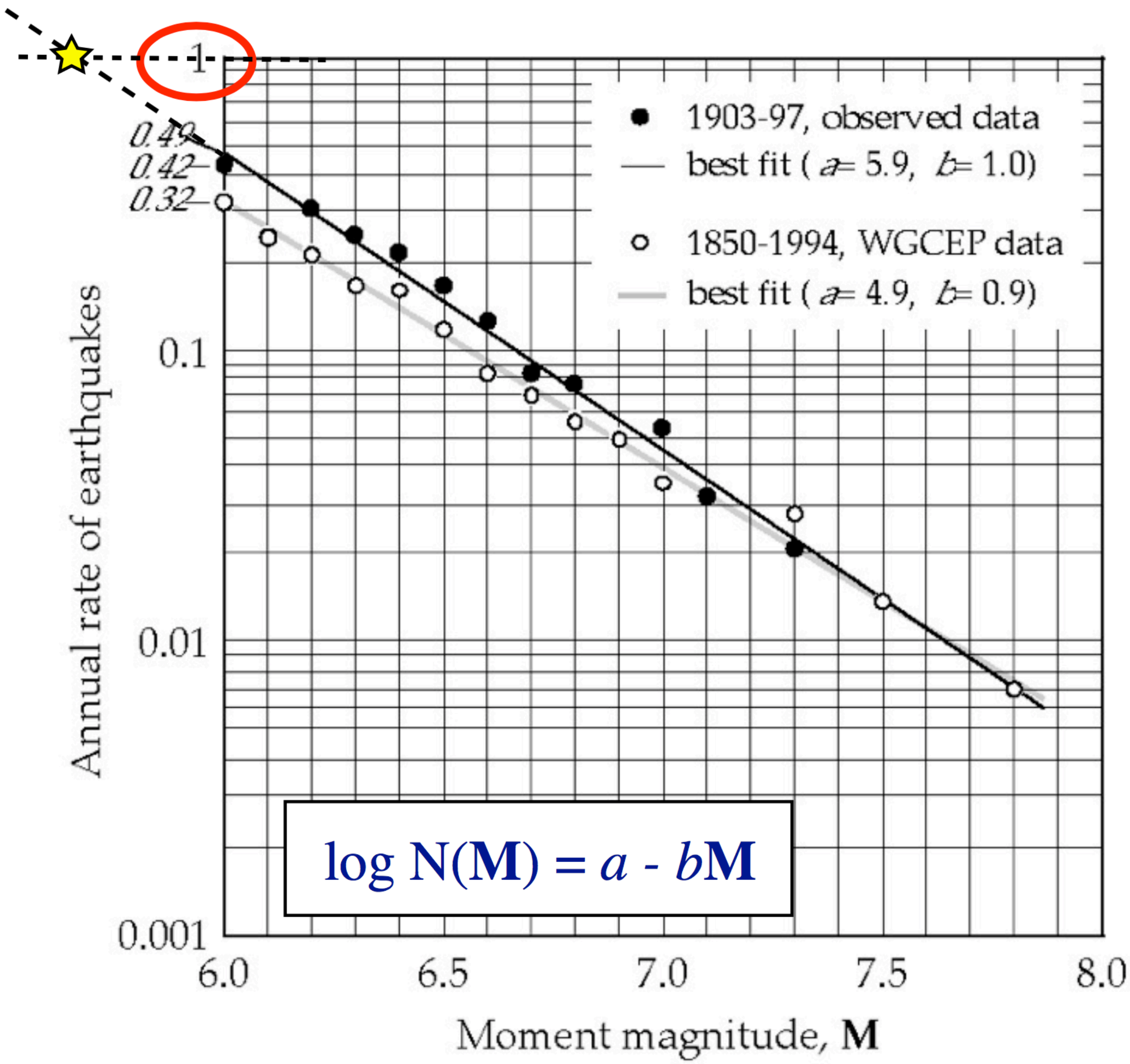
$$y(x) = a_0 e^{a_1 x} \rightarrow log(y(x)) = log(a_0) + a_1 x$$

- Multi-dimensional regression, multiple independent variables:

$$y(x_1, x_2, ..., x_N) = a_0 + a_1 x_1 + a_2 x_2 + ... + a_N x_N$$

Same approach, but leads to system of N linear equations → later.

Example: The Gutenberg-Richter law



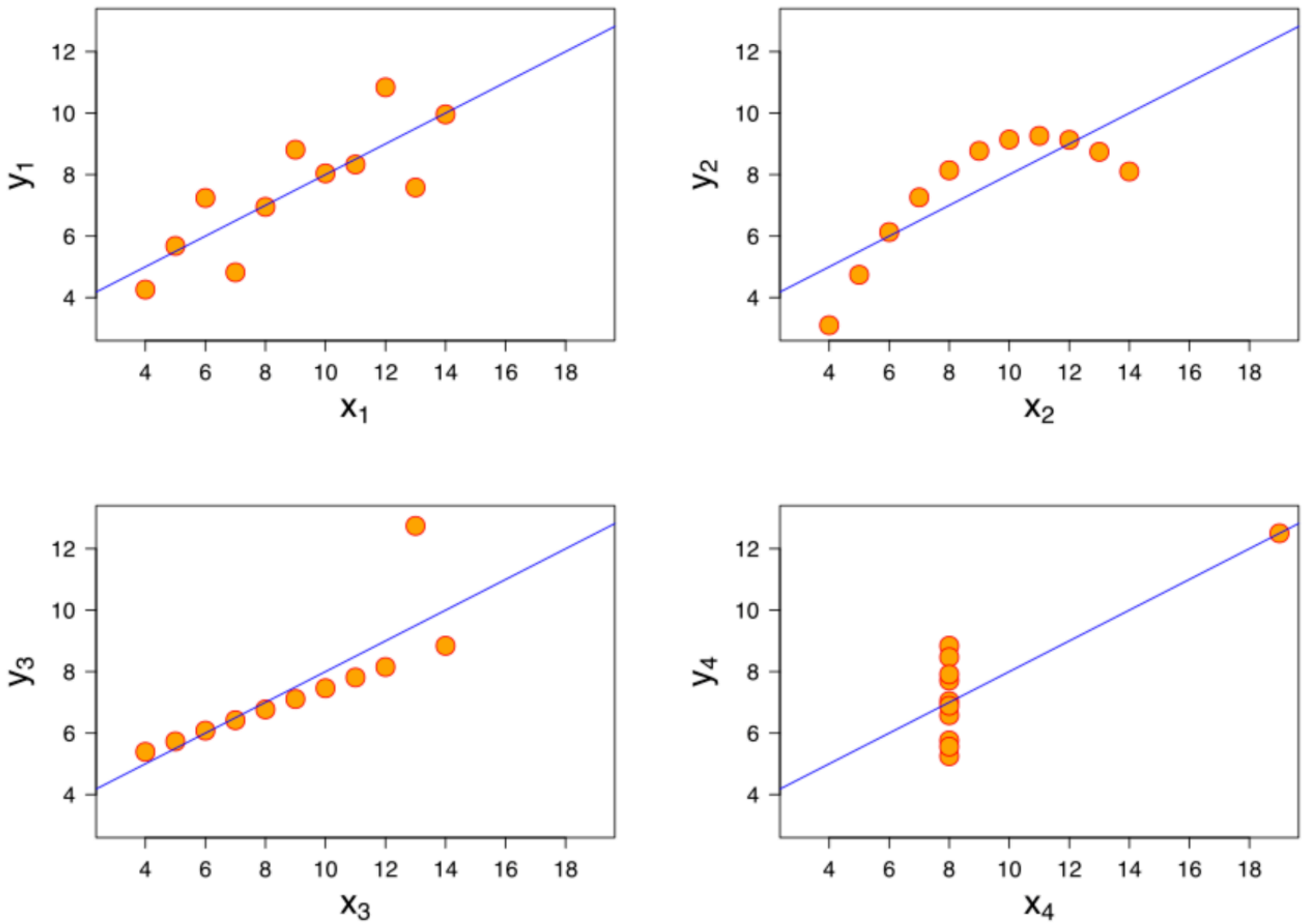
Southern California earthquake data R. Stein and T. Hanks, USGS

Here, note that M = **a/b** is off to the left (these data are for quakes that happen less than once per year).

What's the magnitude of the once per year quake for each dataset?

Use of these plots: predicting how often big ones occur (we need to know the maximum size)

Pitfalls



The data sets in the Anscombe's quartet are designed to have approximately the same linear regression line (as well as nearly identical means, standard deviations, and correlations) but are graphically very different. This illustrates the pitfalls of relying solely on a fitted model to understand the relationship between variables.