Understanding a distribution of measurements

Let’s say you have a repeated measurements of some value. How do we estimate the best value and uncertainty.

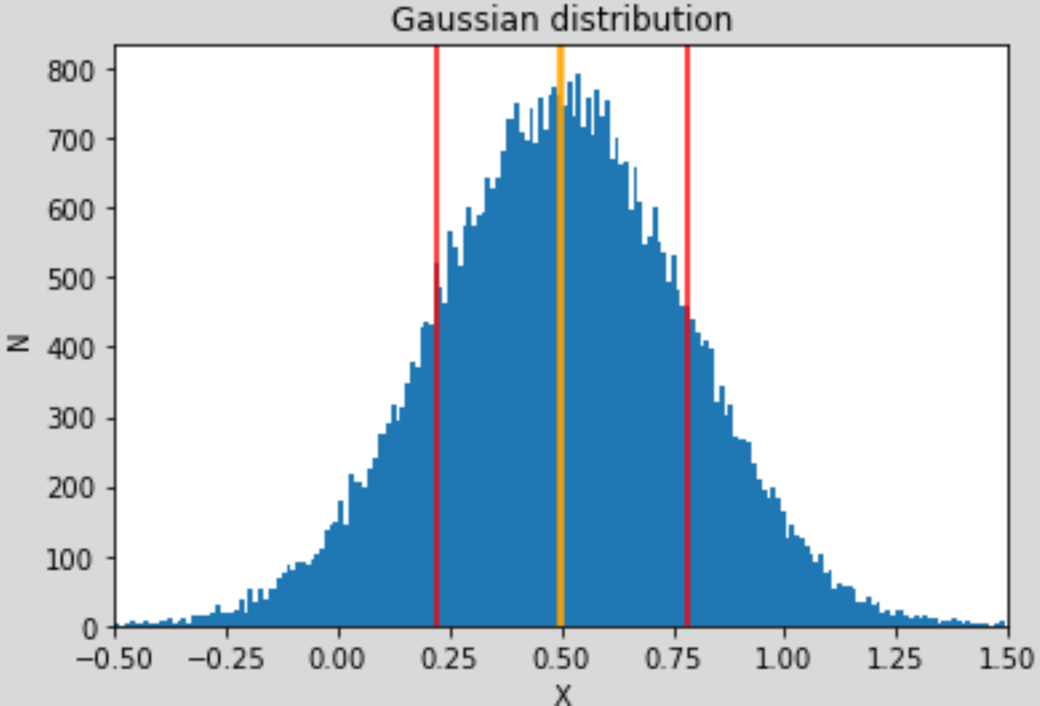
If your errors are independent and follow a Gaussian distribution:

- measure mean and standard deviation $(\bar{x}, \sigma)$
- “standard error in the mean” is given by $\sigma/\sqrt{N}$

```
mean     = np.average(data)
stdev    = np,std(data)
mean_err = stdev/np.sqrt(len(data))
```

Is this a good assumption? Take a distribution of 50,000 measurements with $\bar{x}, \sigma$ = 0.5, 0.28, look at distribution.

(yellow: mean, red: mean +/- 1σ)



Understanding a distribution of measurements

Let's say you have a repeated measurements of some value. How do we estimate the best value and uncertainty.

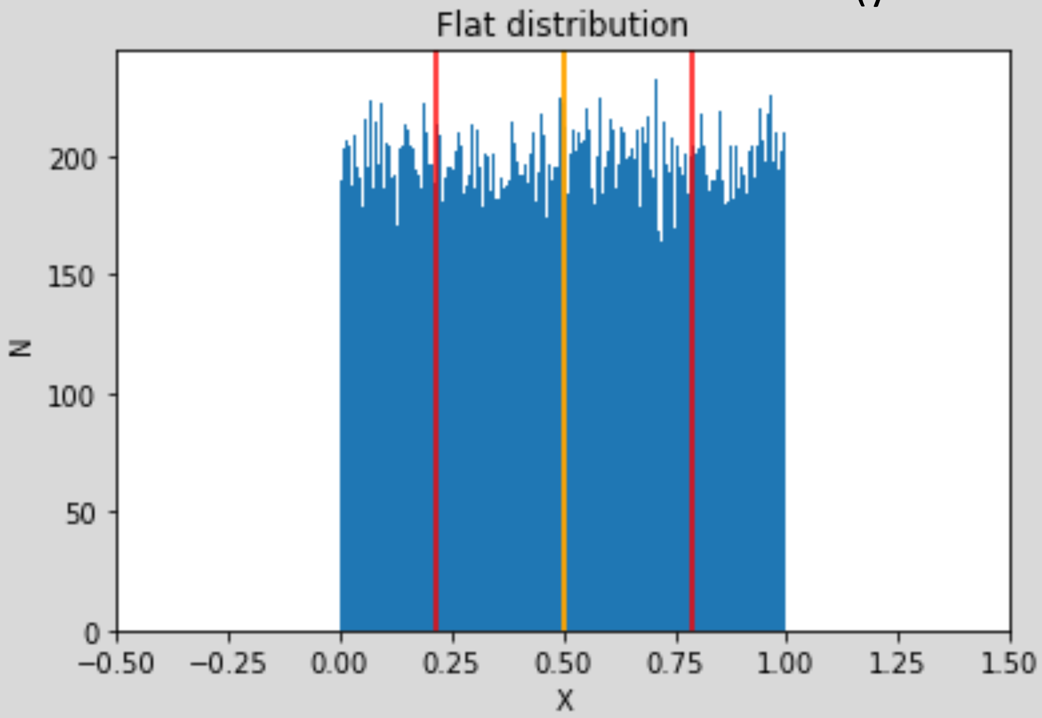
If your errors are independent and follow a Gaussian distribution:

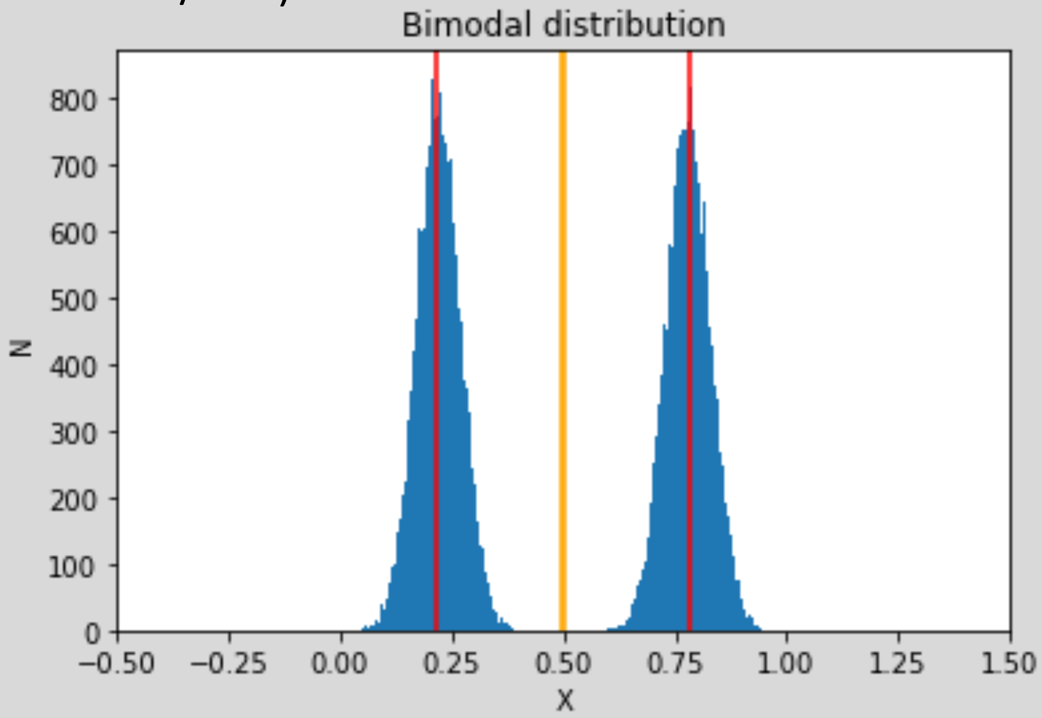
- measure mean and standard deviation $(\bar{x}, \sigma)$
- "error in the mean" is given by $\sigma/\sqrt{N}$

```
mean     = np.average(data)
stdev    = np,std(data)
mean_err = stdev/np.sqrt(len(data))
```

But other distributions can mimic the same answer, and may or may not be meaningful!

(yellow: mean, red: mean +/- 1σ)





Understanding a distribution of measurements

Let’s say you have a repeated measurements of some value. How do we estimate the best value and uncertainty.

If your errors are independent and follow a Gaussian distribution:

- measure mean and standard deviation $(\bar{x}, \sigma)$
- “error in the mean” is given by $\sigma/\sqrt{N}$

```
mean     = np.average(data)
stdev    = np,std(data)
mean_err = stdev/np.sqrt(len(data))
```

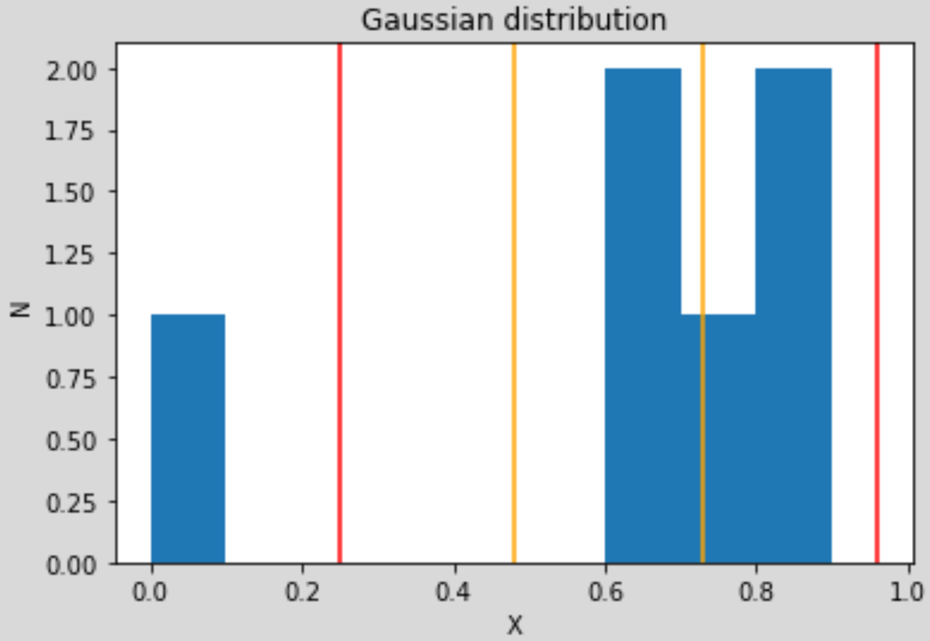
And when the amount of data is small, it can be hard to tell if these are good estimates!

(yellow: mean, red: mean +/- 1σ)

Moral of the story:

Gotta look at your data!

Do a `plt.hist(data)` to be sure.



Characterizing a linear (or linearized) relationship:

- Dataset of N points: $(x_i\ , y_i)$
- Fit a line to data: $y = mx + b$
- Calculate **slope**, **intercept**, and their **uncertainties**: $m \pm \sigma_m, b \pm \sigma_b$
- Calculate root-mean-square (RMS) **scatter** around the fit: $\sigma_{RMS}^2 \equiv \frac{1}{N}\sum\left(y_i - y_{fit}\right)^2$

The importance of scatter

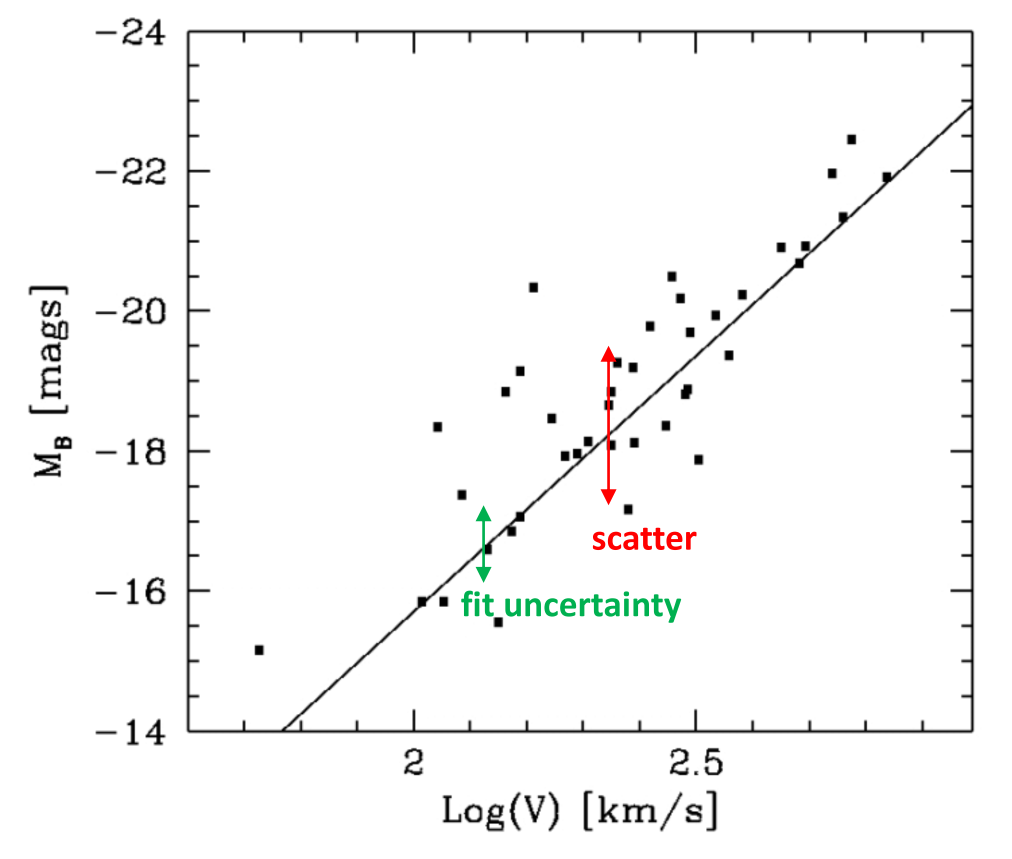
The uncertainties on the fit tell you how well-determined the fit parameters are.

The scatter of the fit tells you how well, on average, individual data points obey the relationship.

Example: Tully Fisher Relationship ⇒

Lower fit uncertainties mean that the TF relationship is better-determined.

Large scatter means any one galaxy may not perfectly obey TF.



Characterizing a linear (or linearized) relationship (least squares fitting, assuming Gaussian statistics):

```
# make a linear fit, and calculate uncertainty and scatter

good = <some criterion> # dont want to include bad data

coeff, cov = np.polyfit(x[good],y[good],1,cov=True)

coeff_err = np.sqrt(np.diag(cov))

print(' slope = {:.3f} +/- {:.3f}'.format(coeff[0],coeff_err[0]))

print('intercept = {:.3f} +/- {:.3f}'.format(coeff[1],coeff_err[1]))

polynomial=np.poly1d(coeff)

xfit=np.linspace(x.min(),x.max())

plt.plot(xfit,polynomial(xfit),color='green',lw=3)

print(' scatter = {:.3f}'.format(np.std(y[good]-polynomial(x[good]))))
```

Linearization

Sometimes you will need to fit a power law, or a sinusoid, or an exponential. These are non-linear models, but can be made linear.

Power Law: $\mathrm{y} = x^{\alpha}$, fit for α

Linearize it: $\log(y) = \log(x^{\alpha}) = \alpha \log(x)$, so fit a straight line to log(y) versus log(x), then the slope is alpha

Sine function: $\mathrm{y} = A \sin x + B$, fit for A and B.

Linearize it: it is already linear if you fit a straight line to y vs sin(x) rather than y vs x.

Exponential: $Y = e^{-x/h}$, fit for h

Linearize it: $\ln y = \ln\left(e^{-x/h}\right) = \frac{-1}{h} x$, so fit a straight line to ln(y) vs x and then h is -1/slope.

Simple Gaussian Propagation of Errors: Adding, Subtracting, Averaging

If you are adding or subtracting two things with uncertainties, the total uncertainty is the **quadrature sum** of the individual uncertainties:

$$z = x \pm y$$

$$\sigma_z^2 = \sigma_x^2 + \sigma_y^2$$

I you are averaging many data values together ($x_i \pm \sigma_{x_i}$) to get a final “best estimate” of what’s being measured, the uncertainty on that estimate is given by the **standard error of the mean**:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$\sigma_{\bar{x}} = \frac{1}{\sqrt{N}}\sigma_{x_i}$$

Simple Gaussian Propagation of Errors: Using a linear (or linearized) function

$$y = f(x) = ax + b$$

Propagate errors using the gradient method, adding in quadrature the error due to each of a, x, and b :

$$\sigma_y^2 = \left[\left(\frac{\partial f}{dx}\right)\sigma_x\right]^2 + \left[\left(\frac{\partial f}{da}\right)\sigma_a\right]^2 + \left[\left(\frac{\partial f}{db}\right)\sigma_b\right]^2$$

so

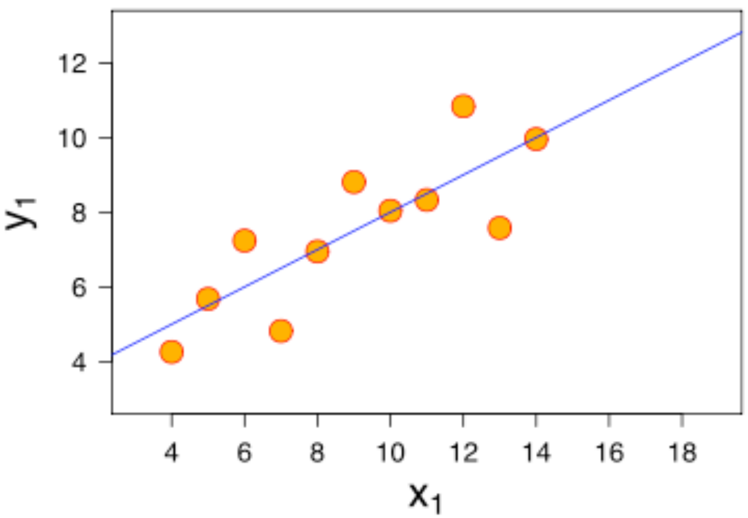
$$\sigma_y^2 = [a\sigma_x]^2 + [x\sigma_a]^2 + [\sigma_b]^2$$

Total uncertainty

Uncertainty due to the uncertainties in a and b

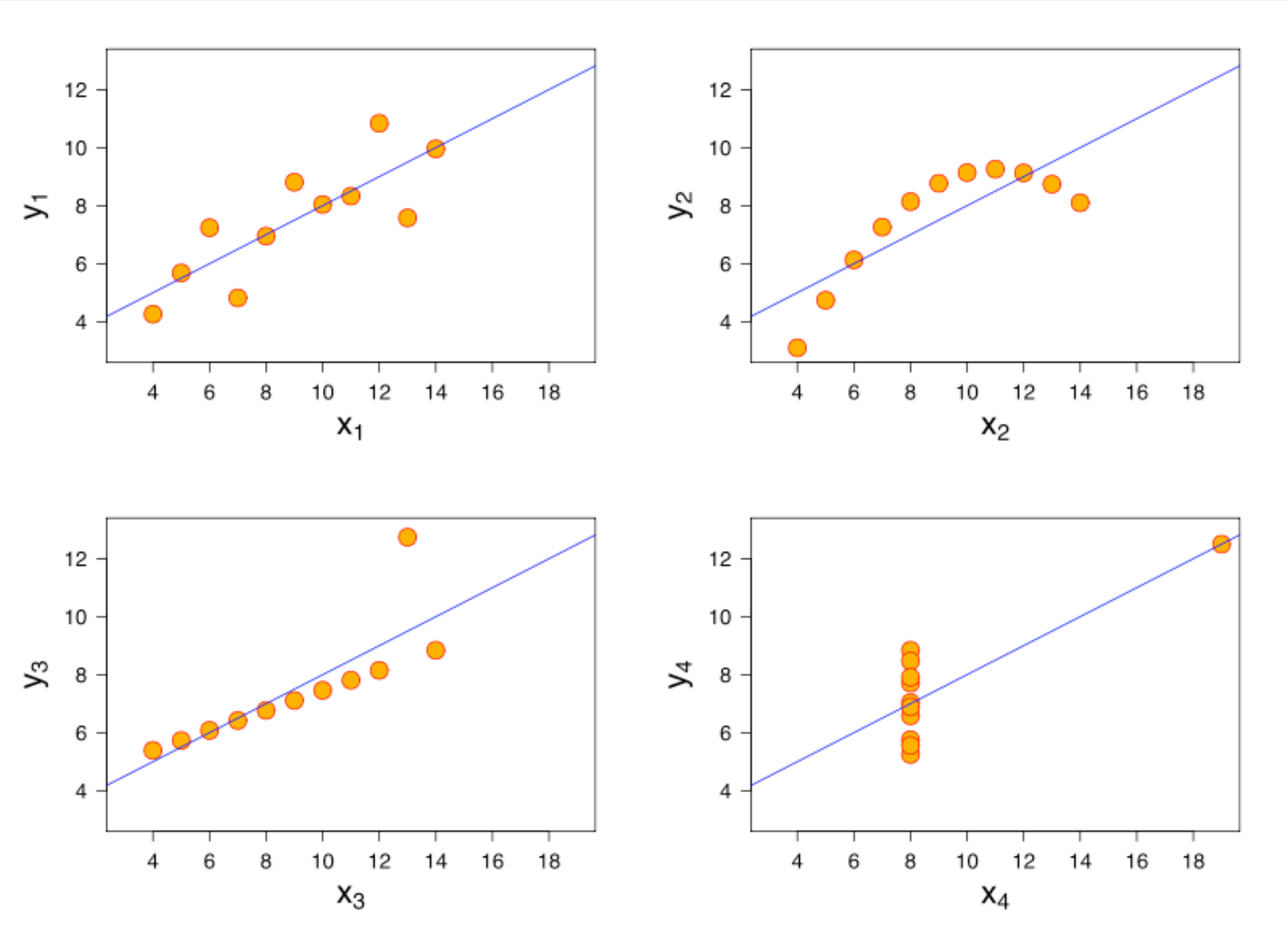
Uncertainty due to the measurement of x

But be careful with fits...



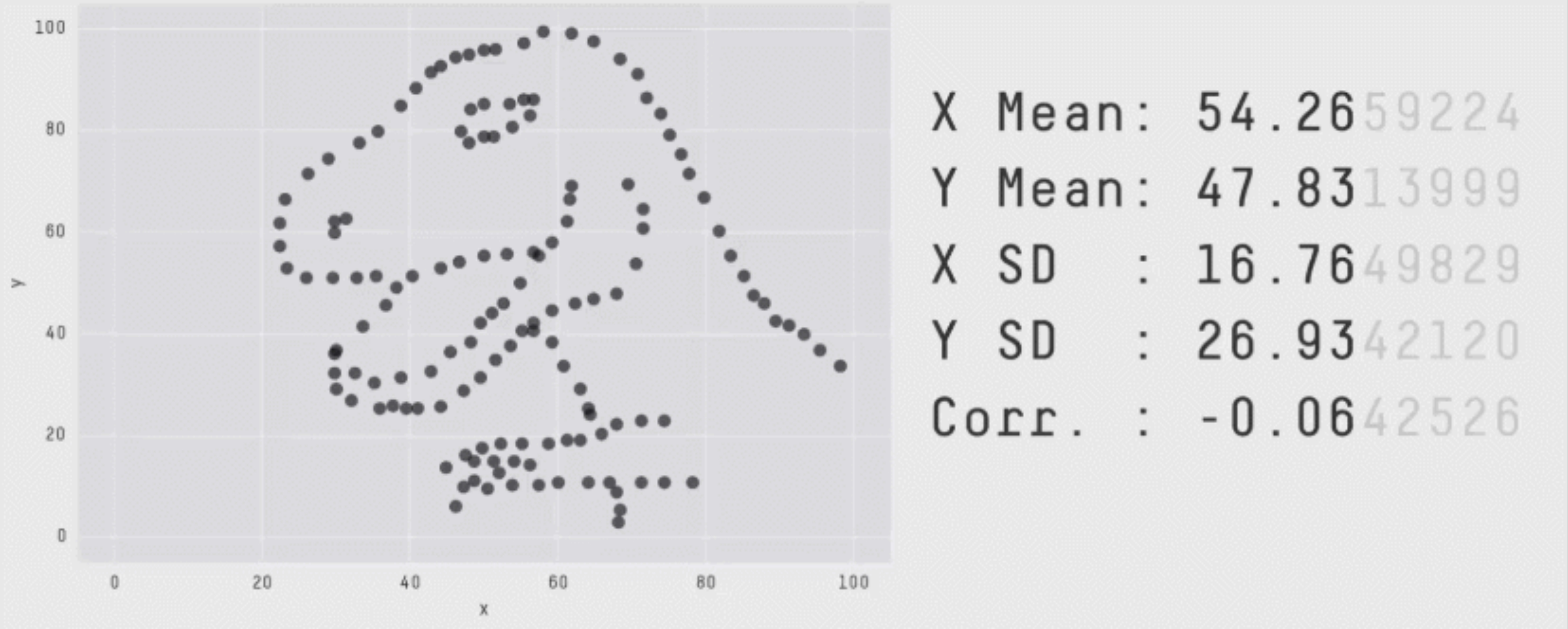
Anscombe's quartet: Fit y=mx+b and get the same r (correlation coefficient), m, b, σ_m, σ_b, σ_{RMS}

But be careful with fits...



Anscombe's quartet: Fit y=mx+b and get the same r (correlation coefficient), m, b, σ_m, σ_b, σ_{RMS}

Beware the datasaurus!



Moral of the story: ALWAYS PLOT YOUR DATA AND ALWAYS OVERPLOT YOUR FITS!