Class 5: Error Estimates

In this class we will review methods to determine statistical errors by re-sampling data, Monte Carlo simulations or error propagation

Class 5: Error Estimates

At the end of this class you should be able to …

- … understand the definition of an error range
- … generate errors through re-sampling data using bootstrap or jack-knife procedures
- … propagate errors in different quantities in linear or non-linear combinations
- … use Fisher matrices to forecast parameter errors
- … model errors using Monte Carlo simulations

What is an error?

- In science we all need to determine the **errors in our measurements**
- *What does a statement such as "$H_0 = 70 \pm 5\ km\ s^{-1}\ Mpc^{-1}$" mean?*
- It almost never means, "H_0 has a value between 65 and 75"
- It almost always means, "there is 68% probability that H_0 lies in the **confidence region** $65 < H_0 < 75$"
- It often means, "the probability distribution for H_0 is a Gaussian with mean $\mu = 70$ and std dev $\sigma = 5$

Statistical versus systematic errors

Image credit: ori.hhs.gov

- **Statistical (or random) errors** are due to noise fluctuations in our data, which are reduced by collecting more data
- **Systematic errors** are consistent offsets due to incorrect calibration of our measurements, which are not reduced by collecting more data
- We focus here on **estimating statistical errors** in data

Error estimation

- Often we have no analytic model for the statistics we estimate from our data. However, we can still determine errors in these statistics using **approximate sampling procedures**

- The basic idea is to build up many "statistical realizations" of the data by random re-sampling, and use the **scatter across these realizations** to estimate the error

- We will consider here the **jack-knife procedure**, **bootstrap procedure** and **Monte Carlo simulations**

Jack-knife errors

- The **jack-knife procedure** allows us to estimate the error in a statistic by re-sampling the data (without replacement)
- Given a dataset with N entries $(x_1, x_2, x_3, \ldots, x_N)$, we create $\boldsymbol{N}$ **separate datasets, deleting 1 entry in turn**

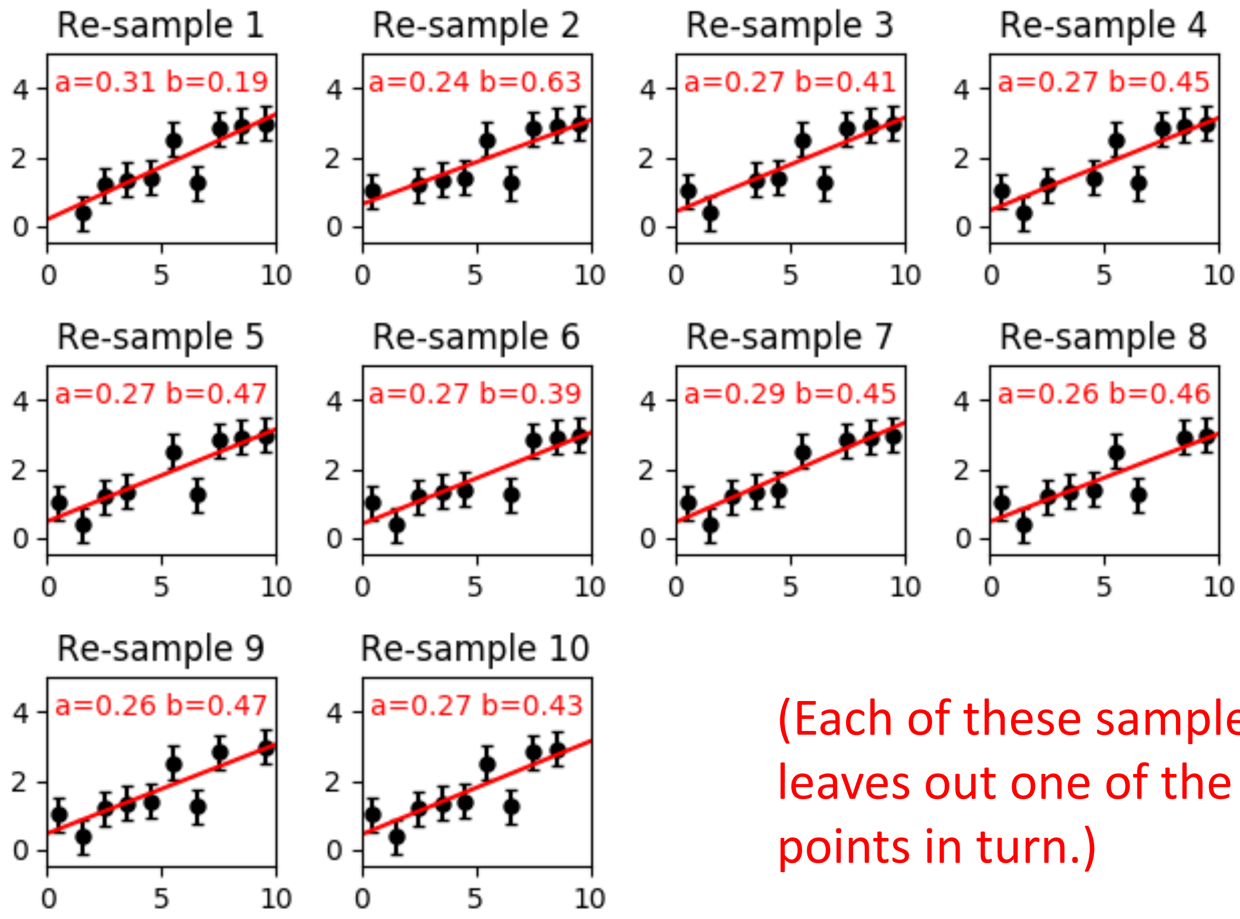
Dataset 1 – delete $\boldsymbol{x_1}$ – $\boldsymbol{D_1 = (x_2, x_3, \ldots, x_N)}$

Dataset 2 – delete $\boldsymbol{x_2}$ – $\boldsymbol{D_2 = (x_1, x_3, \ldots, x_N)}$

- We measure the statistic for each of these datasets $(D_1, D_2, \ldots, D_N)$, creating N measurements $(S_1, S_2, \ldots, S_N)$
- The error is given by: $\boxed{\text{JK error} = \sqrt{N-1} \times \text{std dev of } S_i}$
- [Small print: The factor $\sqrt{N-1}$ is required since the S_i are correlated with each other, given that the datasets D_i all significantly overlap.]

Jack-knife errors

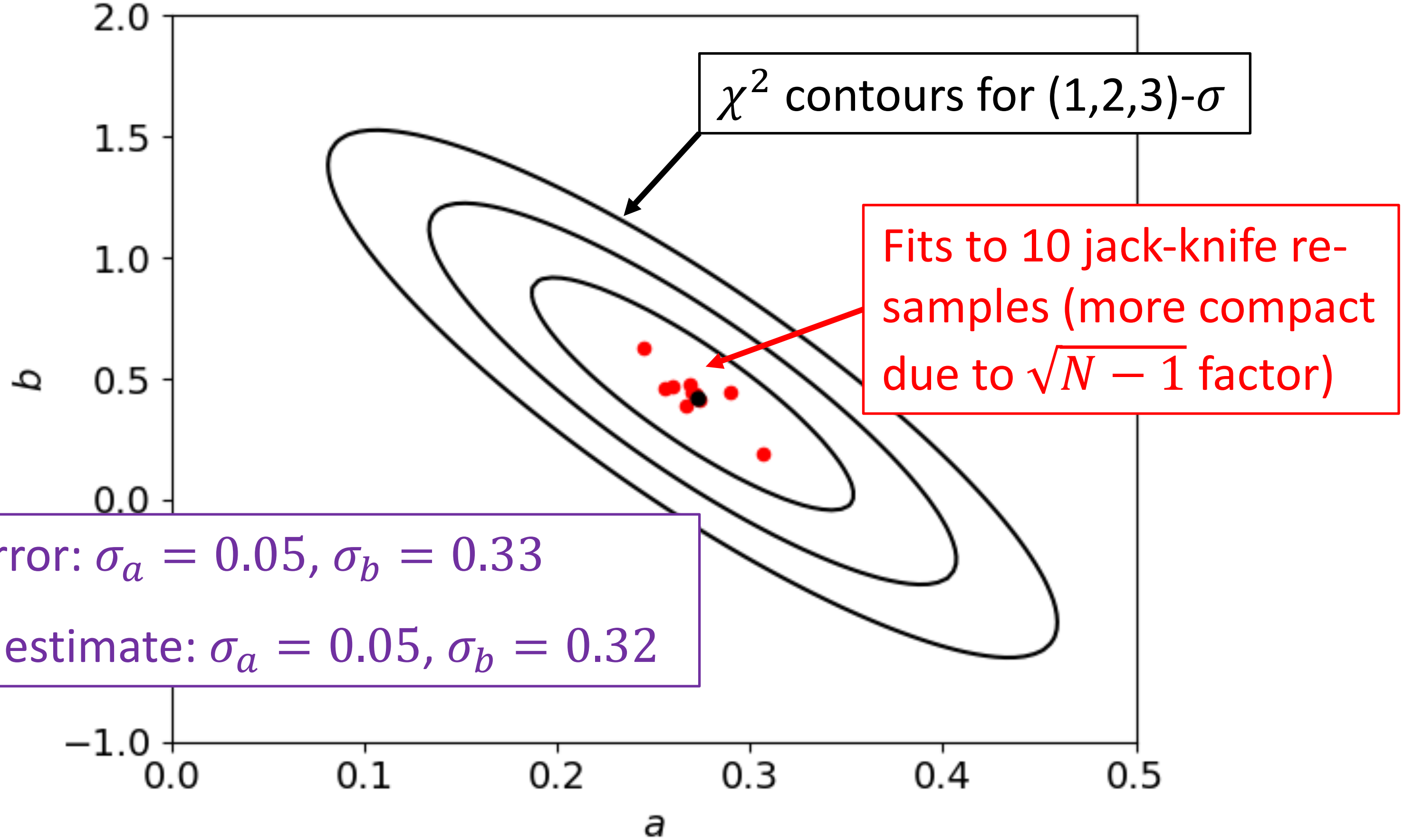
- *Let's apply this procedure to the problem of fitting the straight line from Class 3. Here are the 10 jack-knife samples:*



(Each of these samples leaves out one of the data points in turn.)

Jack-knife errors

- *Here are the best fits of (a, b) to each of those samples, compared to the original χ^2 contours:*

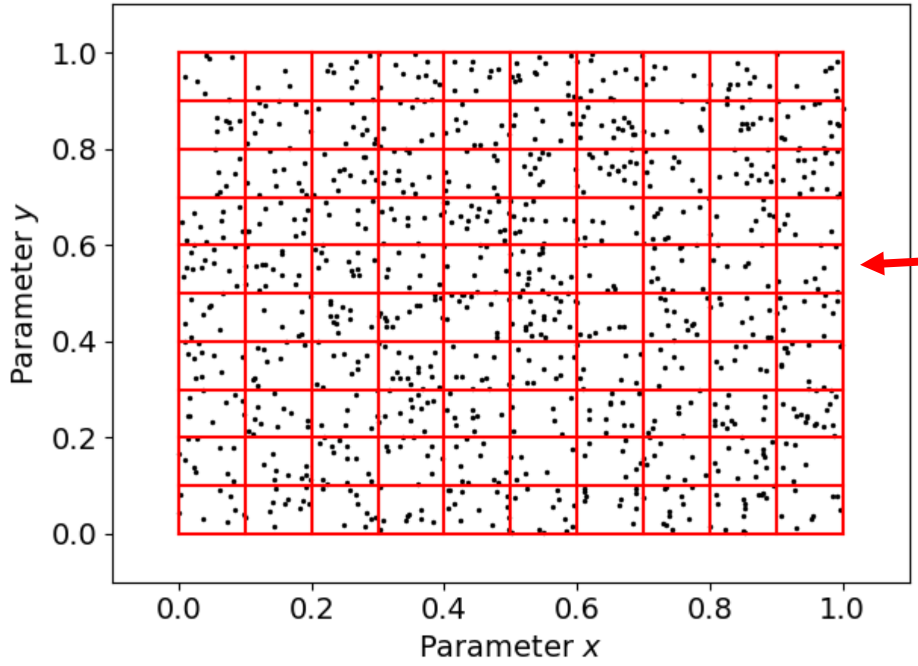


Original error: $\sigma_a = 0.05, \sigma_b = 0.33$

Jack-knife estimate: $\sigma_a = 0.05, \sigma_b = 0.32$

Jack-knife errors

- In some situations, the jack-knife samples could be created by deleting **regions** or **groups of points**, not individual points



100 jack-knife samples can be created by deleting the points inside each region in turn

- [Small print: this could be because each portion of the dataset, deleted in turn, should be statistically independent for a reliable jack-knife error.]

Bootstrap errors

- The **bootstrap procedure** is another method for estimating errors by re-sampling the data
- If we have N data points, the procedure is to **repeatedly draw samples of N points at random, with replacement**
- *Hence, the same point can appear multiple times in each bootstrap sample!*

Bootstrap re-samples of (0,1,2,3,4,5,6,7,8,9):

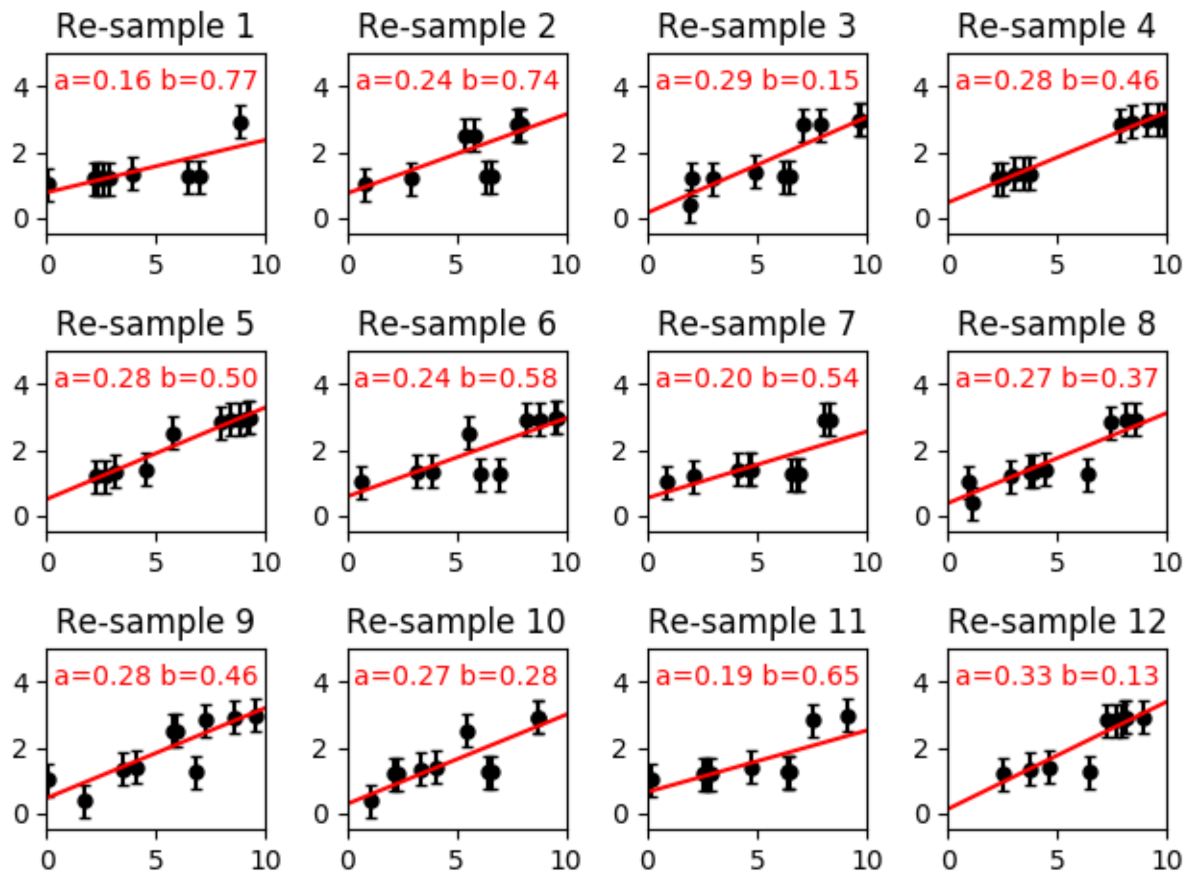
[9 2 6 0 9 8 7 9 2 8]
[7 1 3 4 3 5 9 2 5 7]
[7 9 1 4 9 5 1 3 8 4]
[9 2 2 0 9 2 0 5 0 2]
[8 6 9 0 0 5 6 5 6 6]
[1 8 3 0 0 5 8 8 2 6]
[4 1 9 4 6 6 3 0 4 5]
[0 8 7 6 7 6 6 3 3 7]
[1 2 7 7 5 7 5 0 2 7]
[5 4 9 3 7 0 5 1 2 7]

Bootstrap errors

- The **bootstrap procedure** is another method for estimating errors by re-sampling the data
- If we have N data points, the procedure is to **repeatedly draw samples of N points at random, with replacement**
- *Hence, the same point can appear multiple times in each bootstrap sample!*
- As with the jack-knife error, we measure the statistic for **each** of these bootstrap datasets (which can number $N_{\mathrm{samp}} \gg N$ this time), creating measurements S_i
- The bootstrap error = the standard deviation of S_i (no extra scaling factors this time)

Bootstrap errors

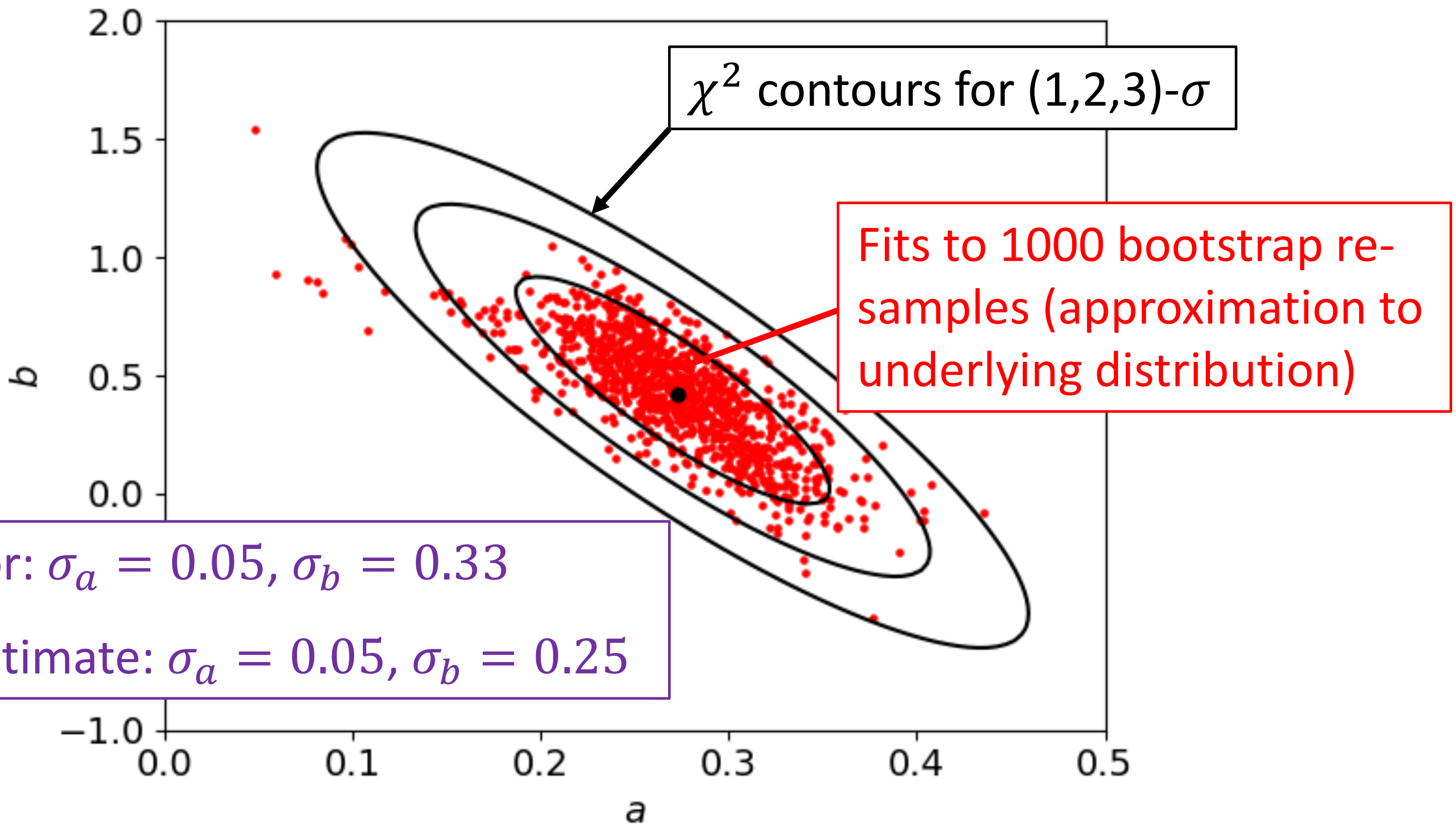
- Let's apply this approach to the straight-line fit. Here are the first 12 of 1000 bootstrap re-samples:



(I slightly offset the points, so you can see some are appearing more than once in each sample.)

Bootstrap errors

- Here are the best fits of (a, b) for each of the 1000 bootstrap re-samples, compared to the original χ^2 contours:



Original error: $\sigma_a = 0.05, \sigma_b = 0.33$

Bootstrap estimate: $\sigma_a = 0.05, \sigma_b = 0.25$

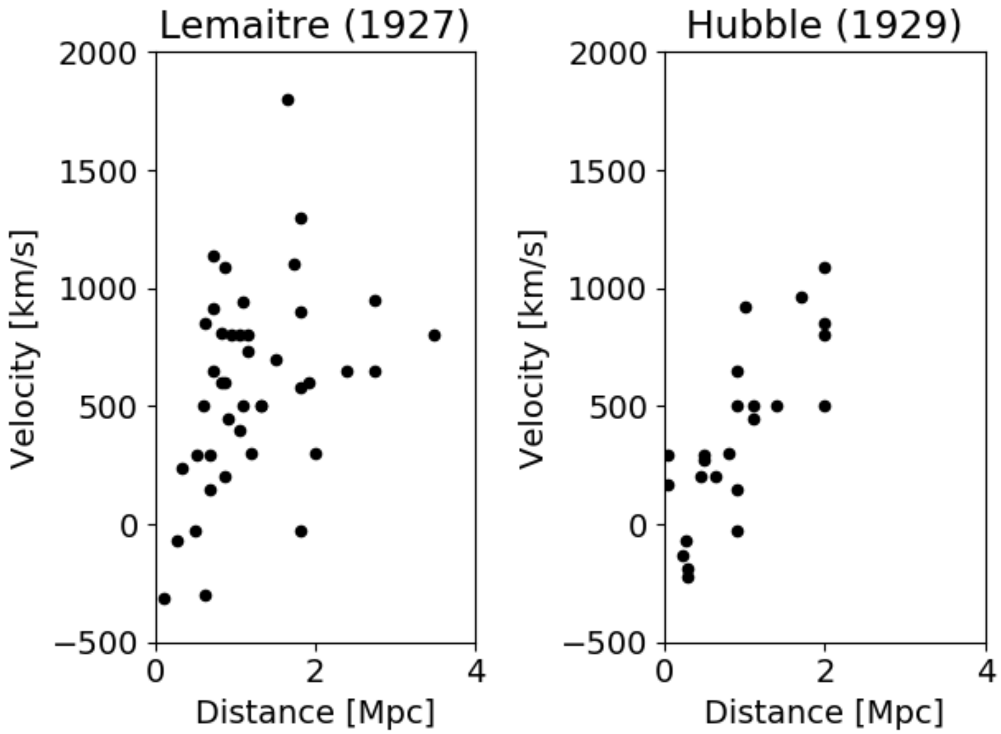
The Hubble parameter (continued)

- In this Activity we will return to our previous analyses of Hubble and Lemaitre's distance-velocity datasets and determine **bootstrap errors** on our measurements

- Find the bootstrap error in the **correlation coefficient**

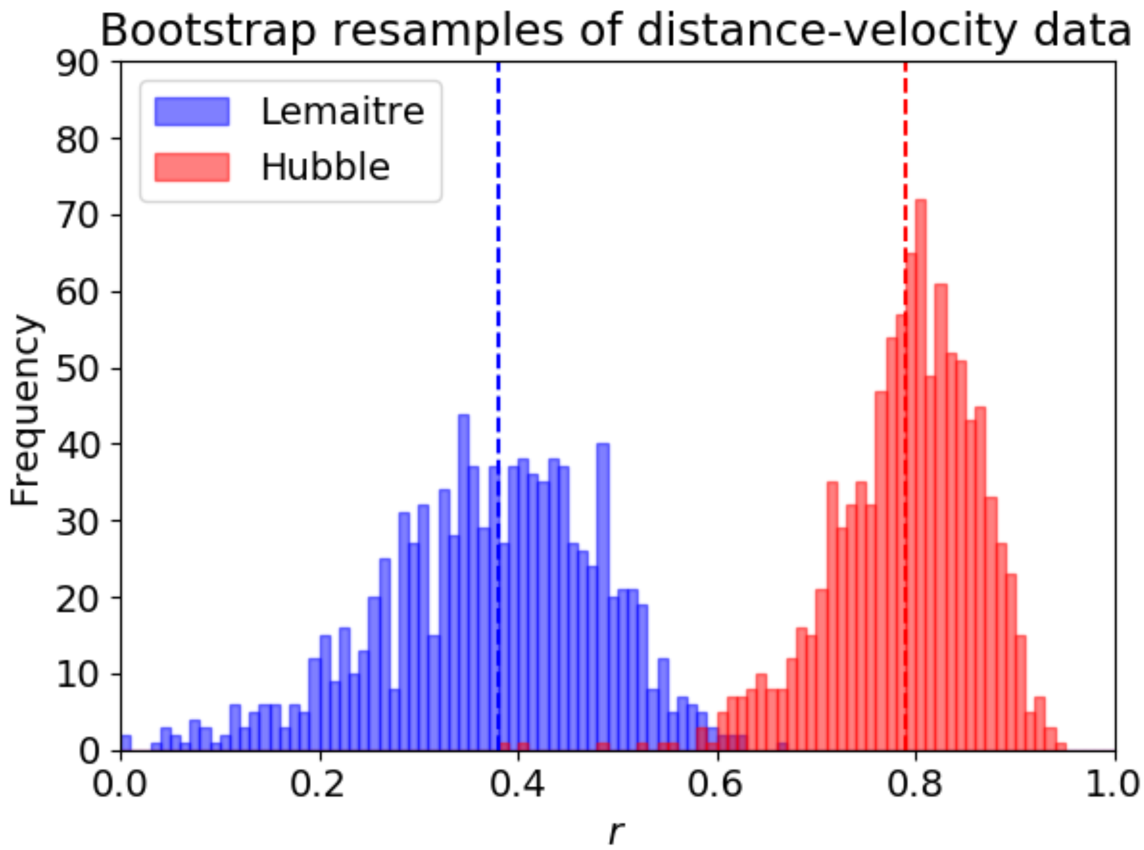
- Find the bootstrap error in the **slope** $\boldsymbol{H_0}$

- *How do these compare to your previous measurements?*



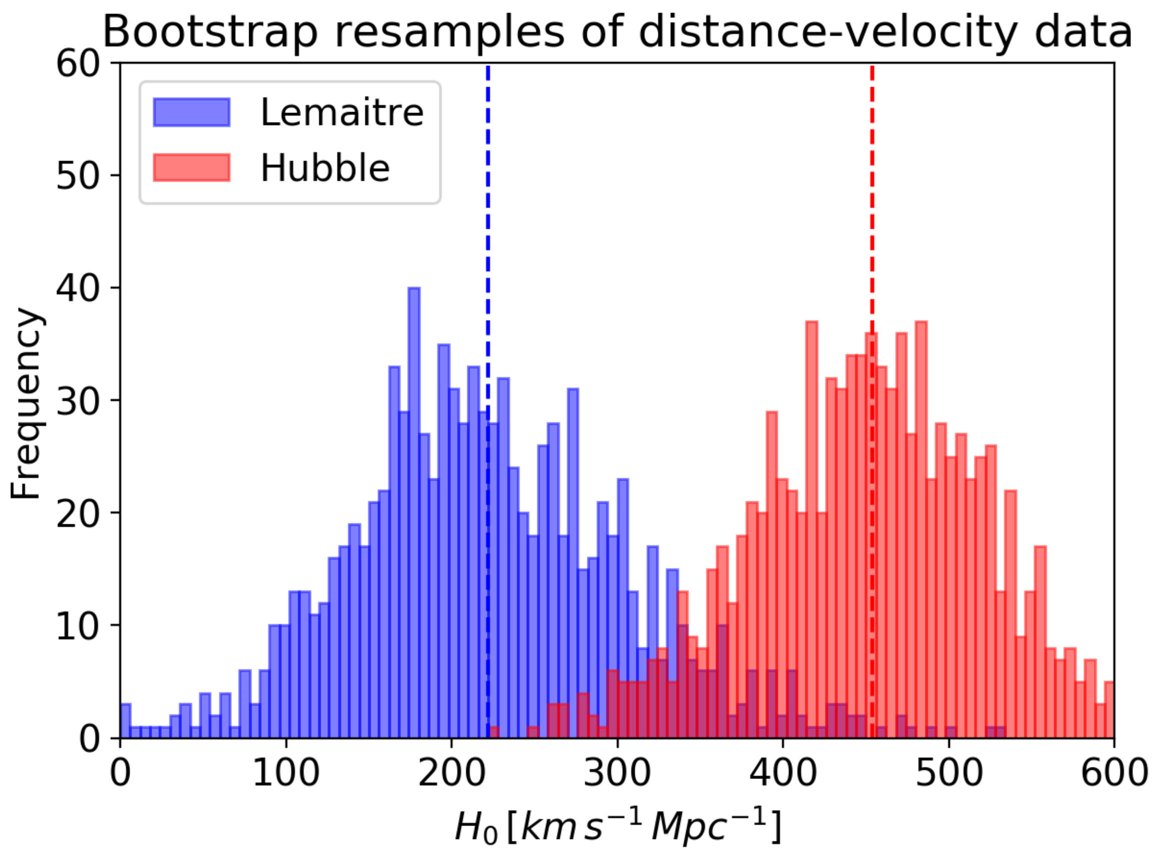
The Hubble parameter (continued)

- *Bootstrap determination of the correlation coefficient errors:*



The Hubble parameter (continued)

- *Bootstrap determination of the errors in the slope:*



Combining and propagating errors

- *A common situation in statistical analysis: we have measurements and errors of some variables. What is the error in a function of those variables?*

- First example: a **linear function of independent variables** (x, y) with coefficients (a, b):

$$z = a\,x + b\,y$$

- The variances combine as:

$$\mathrm{Var}(z) = a^2\,\mathrm{Var}(x) + b^2\,\mathrm{Var}(y)$$

- [**Why?** Consider $\mathrm{Var}(z) = \langle z^2 \rangle - \langle z \rangle^2 = \langle (ax + by)^2 \rangle - (a\langle x \rangle + b\langle y \rangle)^2 = a^2\langle x^2 \rangle + 2ab\langle x \rangle\langle y \rangle + b^2\langle y^2 \rangle - a^2\langle x \rangle^2 - 2ab\langle x \rangle\langle y \rangle - b^2\langle y \rangle^2 = a^2\,\mathrm{Var}(x) + b^2\,\mathrm{Var}(y)$]

Combining and propagating errors

- Now: a **non-linear function of a single variable** x:

$$z = f(x)$$

- An approximation of the propagated error at $x = x_0$ is:

$$\sigma_z = \left| \frac{df}{dx}(x = x_0) \right| \sigma_x$$

- [Small print: this approximation uses the chain rule and assumes the derivative df/dx is approximately constant across σ_x]

- A non-linear function of 2 independent variables, $z = f(x, y)$:

$$\mathrm{Var}(z) = \left(\frac{\partial f}{\partial x}\right)^2 \mathrm{Var}(x) + \left(\frac{\partial f}{\partial y}\right)^2 \mathrm{Var}(y)$$

Combining and propagating errors

- A galaxy of absolute magnitude $M = -20$ is observed to have an apparent magnitude $m = 20.0 \pm 0.2$. What is the luminosity distance D_L in Mpc, and its error?
[Assume $m - M = 5\log_{10} D_L + 25$]

- The total mass of a binary star system (in $M_{\odot}$) is given by Kepler's law $M = a^3/P^2$, where a is the mean separation (in A.U.) and P is the period (in years). The α Centauri system has a period of $P = 79.9 \pm 1.0$ years and mean separation $a = 23.7 \pm 1.0$ A.U. What is the total mass and error?

Fisher matrix

- The **Fisher matrix** is a mathematical technique to propagate the statistical errors in a dataset, to the errors in the model parameters describing the dataset. The matrix is given by:

$$\boxed{F_{ij} = \sum_k \frac{\partial m_k}{\partial p_i} \frac{1}{\sigma_k^2} \frac{\partial m_k}{\partial p_j}}$$

i, j label the model parameters p_i, so the Fisher matrix dimension is $N_{par} \times N_{par}$

k labels the N data points d_k, so this sum is over N terms

$\partial m_k / \partial p_i$ is the partial derivative of the model at point k, with respect to the parameter p_i

σ_k is the error in data point k

- [Small print: assumes Gaussian errors and uncorrelated data points.]

Fisher matrix

$$\boxed{F_{ij} = \sum_k \frac{\partial m_k}{\partial p_i} \frac{1}{\sigma_k^2} \frac{\partial m_k}{\partial p_j}}$$

- Let's evaluate this for the example of **fitting a straight line**, $y = ax + b$ at N positions x_k
- There are 2 parameters, $p_i = (a, b)$
- The model is $m_k = ax_k + b$, so $\frac{\partial m_k}{\partial p_1} = x_k$ and $\frac{\partial m_k}{\partial p_2} = 1$
- The Fisher matrix is hence:

$$\begin{pmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{pmatrix} = \begin{pmatrix} \sum_k \frac{x_k^2}{\sigma_k^2} & \sum_k \frac{x_k}{\sigma_k^2} \\ \sum_k \frac{x_k}{\sigma_k^2} & \sum_k \frac{1}{\sigma_k^2} \end{pmatrix}$$

Fisher matrix

- We have $x_k = (0.5, 1.5, \ \ldots\ , \ 9.5)$ and $\sigma_k = 0.5$, so we can evaluate the above matrix to $F = \begin{pmatrix} 1330 & 200 \\ 200 & 40 \end{pmatrix}$

- The **covariance matrix of the parameters is then given by the inverse of the Fisher matrix**:

$$\boxed{C = F^{-1}}$$

Error in $p_1 = \sqrt{C_{11}}$
Error in $p_2 = \sqrt{C_{22}}$

- For our example, $C = \begin{pmatrix} 0.003 & -0.015 \\ -0.015 & 0.101 \end{pmatrix}$ hence the forecast errors are $\sigma_a = \sqrt{C_{11}} = 0.05$ and $\sigma_b = \sqrt{C_{22}} = 0.32$

- *We have propagated the errors theoretically, without needing to perform any re-sampling*

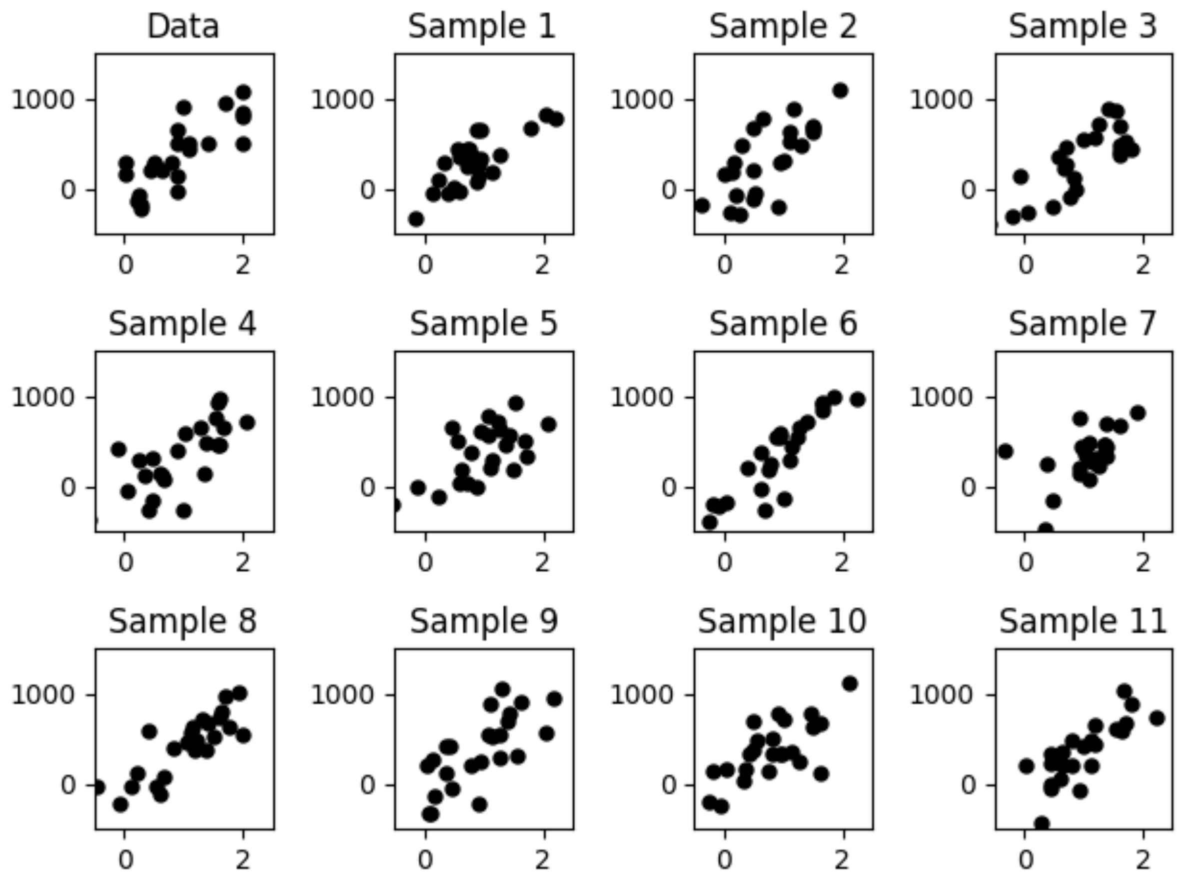
Monte Carlo simulations

- A **Monte Carlo simulation** is a computer model of an experiment in which many random realizations of the results are created and analysed like the real data
- This allows us to determine the errors in our measurements, as the *standard deviation of the fitted parameters over the realizations*

"many realizations of an experiment"

The Hubble parameter (continued)

- Run a **Monte Carlo simulation of Hubble's distance-redshift investigation**, and hence determine the error in H_0



Summary

At the end of this class you should be able to …

- … understand the definition of an error range
- … generate errors through re-sampling data using bootstrap or jack-knife approaches
- … propagate errors in different quantities in linear or non-linear combinations
- … use Fisher matrices to forecast parameter errors
- … model errors using Monte Carlo simulations