data science for (physical) scientists 4

from uncertainties to MCMC

dr.federica bianco | *fbb.space* |  *fedhere* |  *fedhere*

this slide deck

https://slides.com/federicabianco/dsps_4

NHRT

- *Theories* should be *falsifiable* (= make predictions)
- *Analysis* should be *reproducible* (share result, share raw data, share code to get result from raw data)

1 formulate your prediction (NH)

2 identify all alternative outcomes (AH)

3 set confidence threshold (*p*-value)

4 find a measurable quantity which under the Null has a known distribution (pivotal quantity)

5 calculate the pivotal quantity

6 calculate probability of value obtained for the pivotal quantity under the Null

Key Slide

if probability < *p*-value : reject Null

Null
Hypothesis
Rejection
Testing

$$P(A) + P(\bar{A}) = 1$$

if *all alternatives* to our model are ruled out, then our model must hold

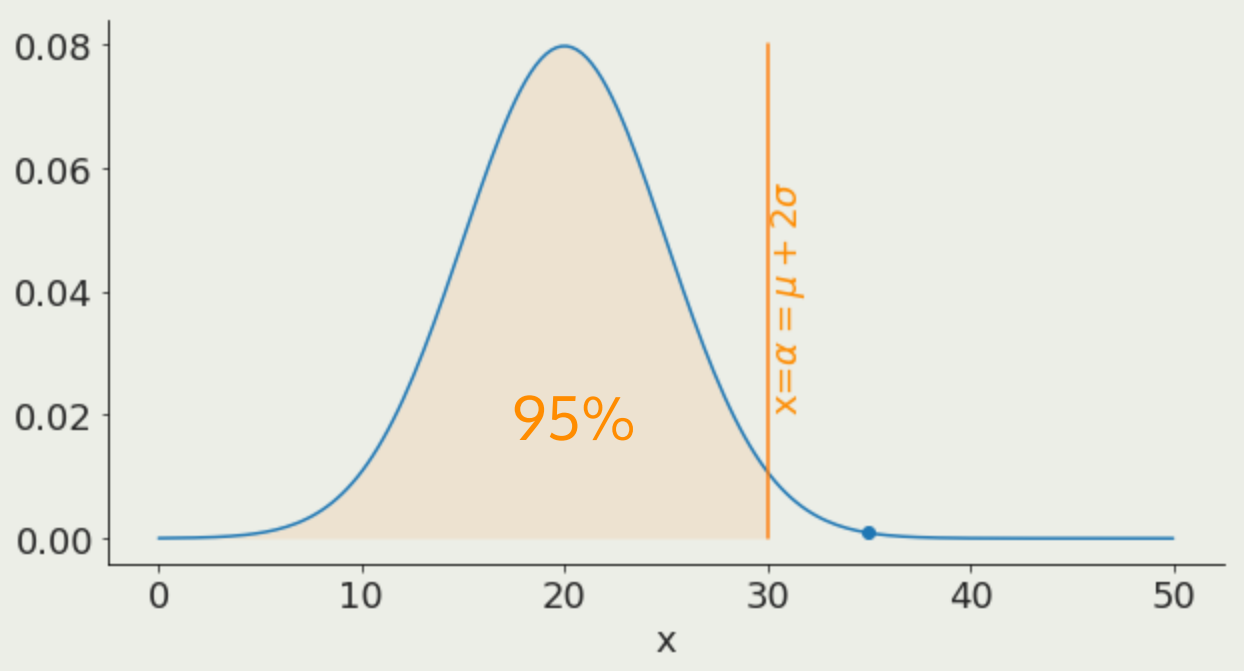
2 identify all alternative outcomes

Alternative Hypothesis

Null
Hypothesis
Rejection
Testing

$p - value$ what is α?

α is the x value corresponding to a chosen threshold



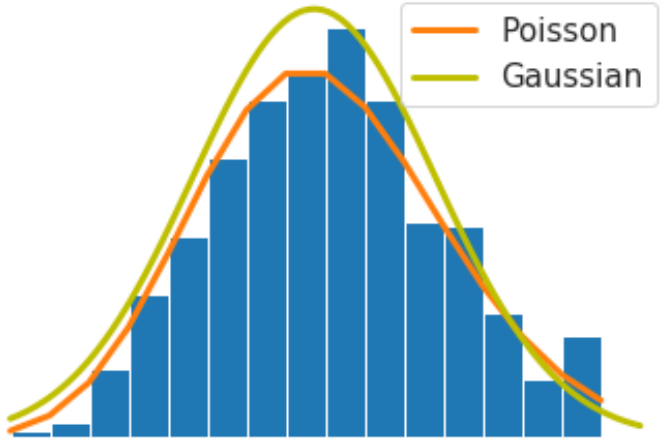
it represent the probability to get a result at least as extreme just by chance

6

test data against
alternative outcomes

1 uncertainty in measurements

2 stochasticity in nature



3 combining uncertainties

4 MonteCarlo methods

1 uncertainties in measurements

2 types of uncertainties

Statistical

Systematic

1.1 statistical uncertainties

stochastic or random or statistical errors

unpredictable uncertainty in a measurement due to lack of sensitivity in the measurement or to stochasticity in a process

Compact and clear pamphlet on errors from a physics perspective from Edo University (Nigeria)
https://www.edouniversity.edu.ng/oerrepository/articles/phy_119_general_physics_practical_20182019.pdf

stochastic or random or statistical errors

unpredictable uncertainty in a measurement
due to lack of sensitivity in the measurement or
to stochasticity in a process

2.5 +/- 0.1 cm

stochastic or random or statistical errors

unpredictable uncertainty in a measurement
due to lack of sensitivity in the measurement or
to stochasticity in a process

2.0 +/- ε cm, ε > 0.1 cm

stochastic or random or statistical errors

every measurement will be a bit different

2.0 +/- ε cm, ε > 0.1 cm

stochastic or random or statistical errors

Deterministic systems have no randomness in their evolution. *Chaos* is deterministic.

definition: a chaotic system loses memory of the initial conditions exponentially

(*it's hard to* predict)

Stochastic processes can be

completely random:

the probability of any event is disjoint from that of the previous one(s)

(*cannot* be predicted)

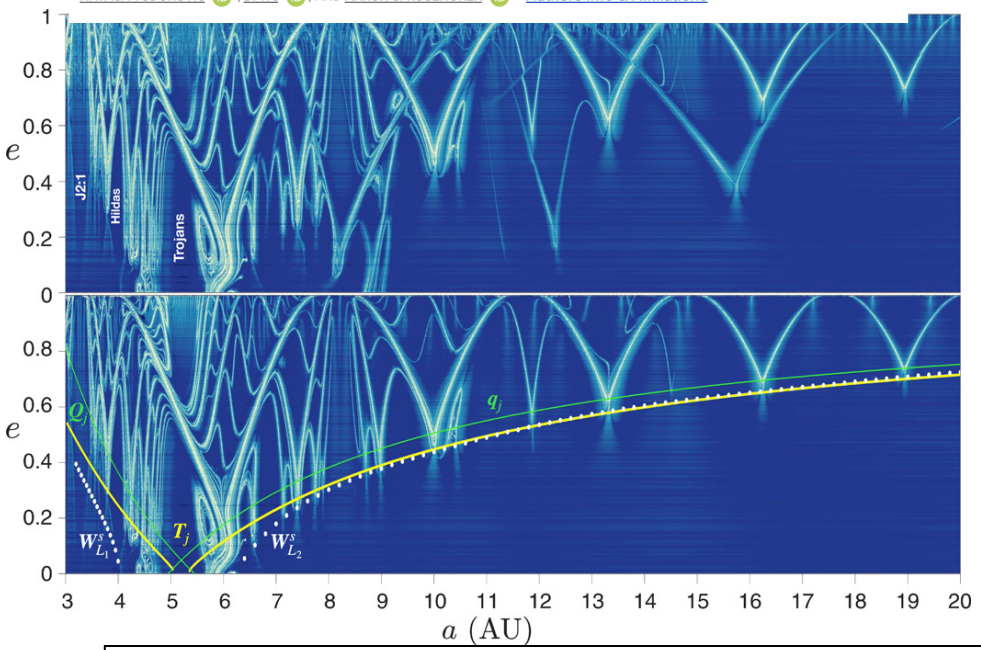
ScienceAdvances Current Issue First release papers

HOME > SCIENCE ADVANCES > VOL. 6, NO. 48 > THE ARCHES OF CHAOS IN THE SOLAR SYSTEM

RESEARCH ARTICLE | ASTRONOMY

The arches of chaos in the Solar System

NATAŠA TODOROVIĆ, DI WU, AND AARON J. ROSENGREN Authors Info & Affiliations



Here, we reveal a notable and hitherto undetected ornamental structure of manifolds, connected in a series of arches that spread from the asteroid belt to Uranus and beyond

chaotic diffusion from orbital resonances

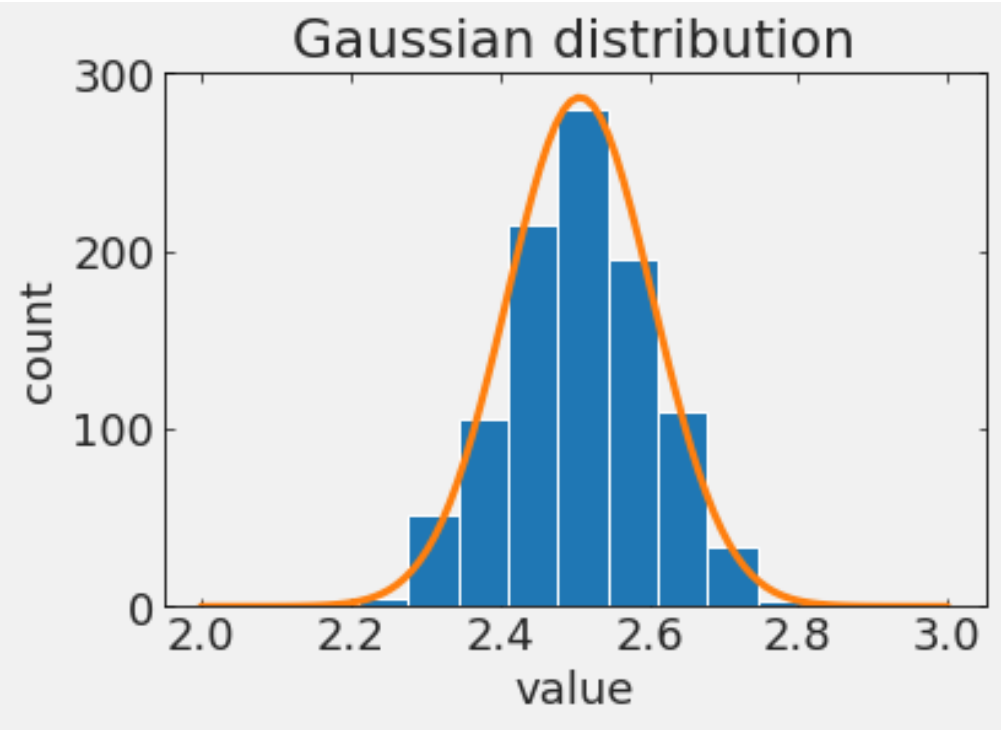
stochastic or random or statistical errors

every measurement will be a bit different

```
2.4, 2.6, 2.5, 2.3, 2.4,
2.7, 2.3, 2.5, 2.6, 2.4
```

2.0 +/- ε cm, ε > 0.1 cm

stochastic or random or statistical errors

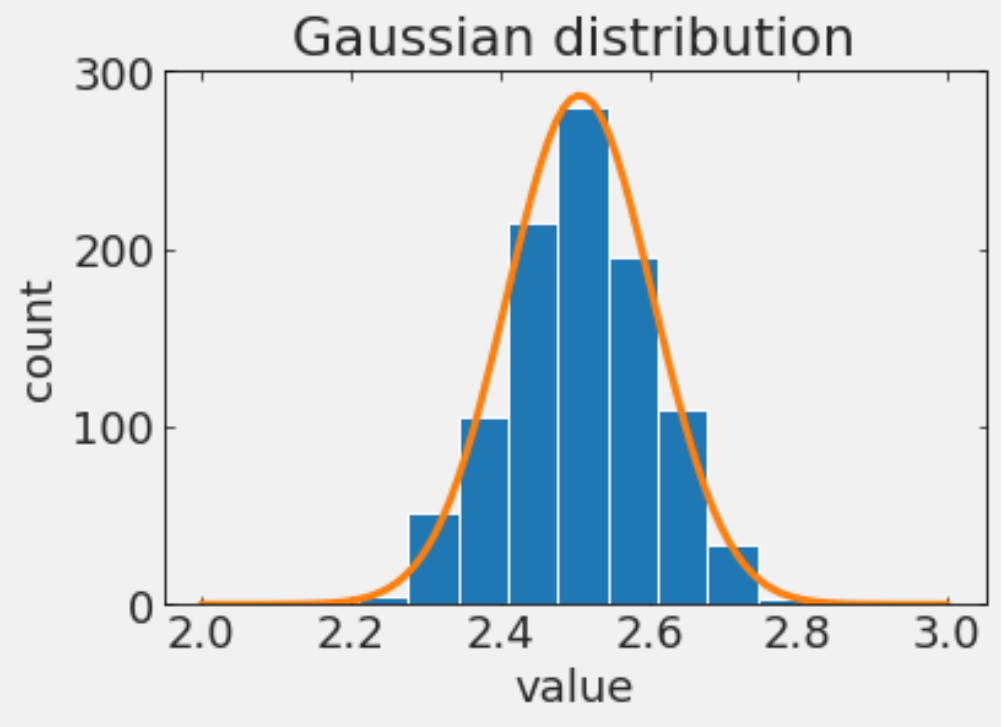


2.0 +/- ε cm, ε > 0.1 cm

stochastic or random or statistical errors

$$p(y_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp - \frac{(y_i - (mx_i + b))^2}{2\sigma_i^2}$$

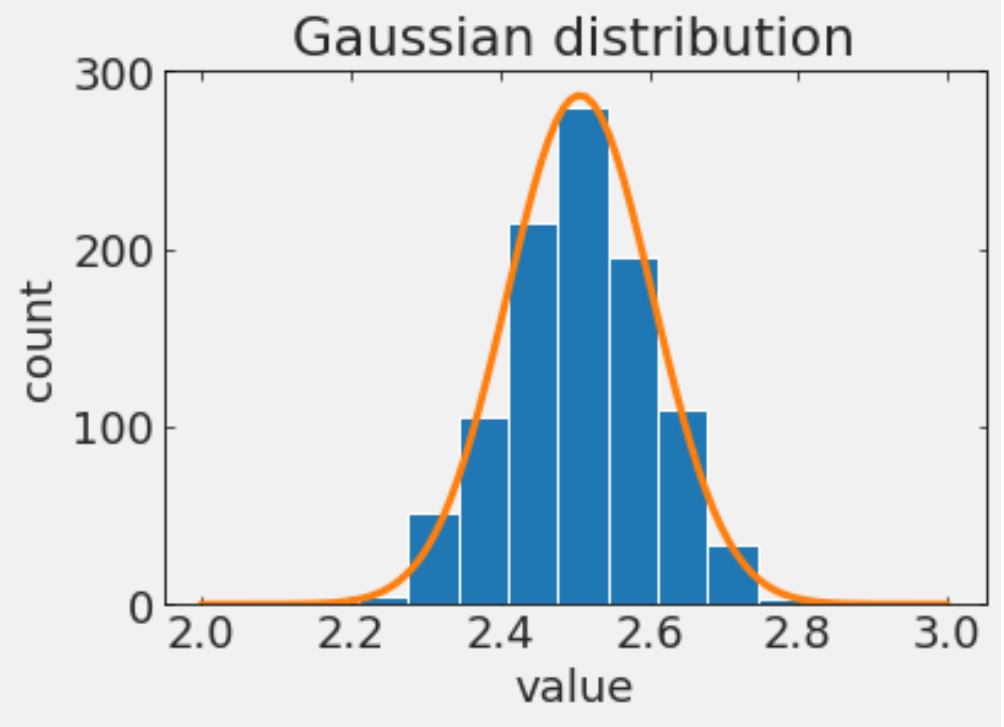
if I'm lucky, the uncertainty is consistent with a Gaussian distribution



stochastic or random or statistical errors

$$p(y_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp - \frac{(y_i - (mx_i + b))^2}{2\sigma_i^2}$$

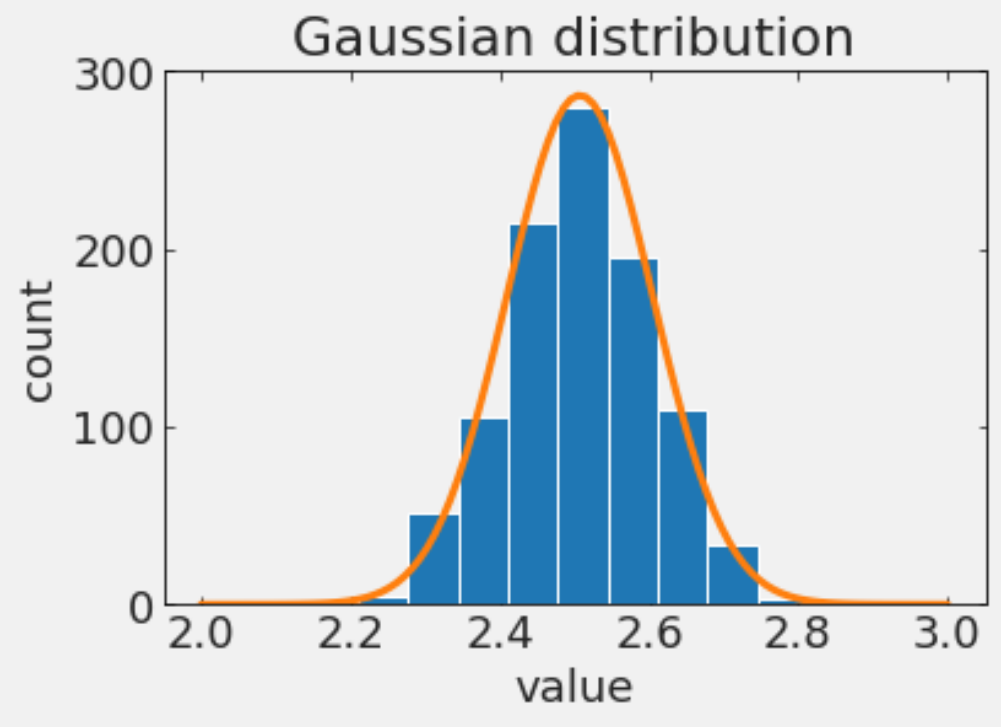
- symmetric



stochastic or random or statistical errors

$$p(y_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp - \frac{(y_i - (mx_i + b))^2}{2\sigma_i^2}$$

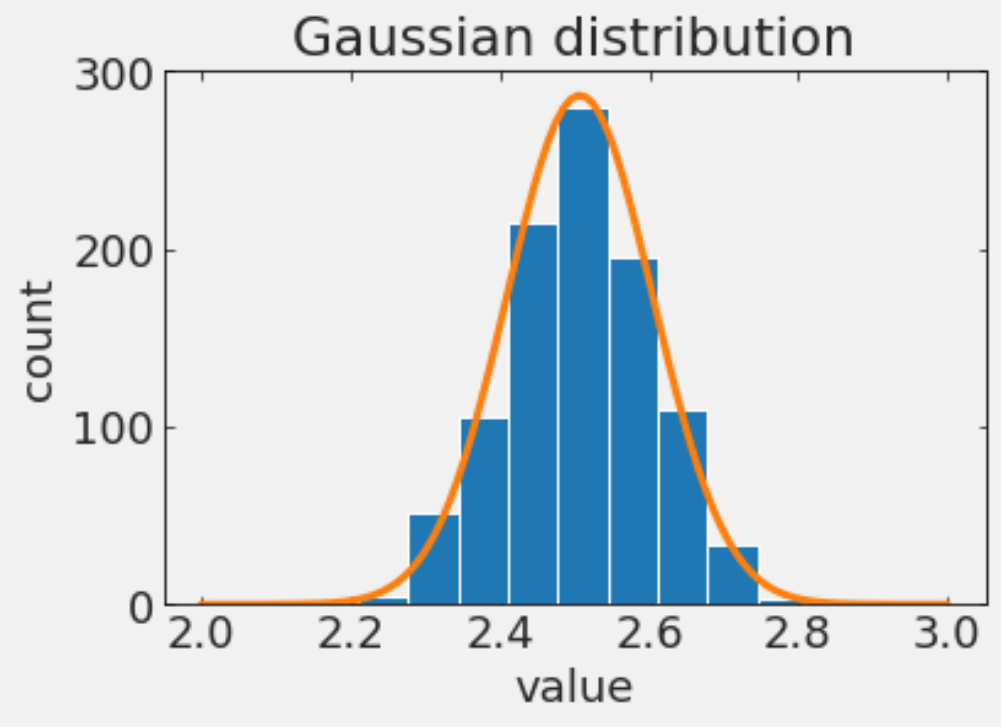
- symmetric
- max at $y_i = (mx_i + b)$



stochastic or random or statistical errors

$$p(y_i) = \frac{1}{\sigma_i \sqrt{2\pi}} \exp - \frac{(y_i - (mx_i + b))^2}{2\sigma_i^2}$$

- symmetric
- max at $y_i = (mx_i + b)$
- bell shaped



stochastic or random or statistical errors

of particular interest are ***Poisson processes***

A discrete distribution that expresses the probability of a number of events

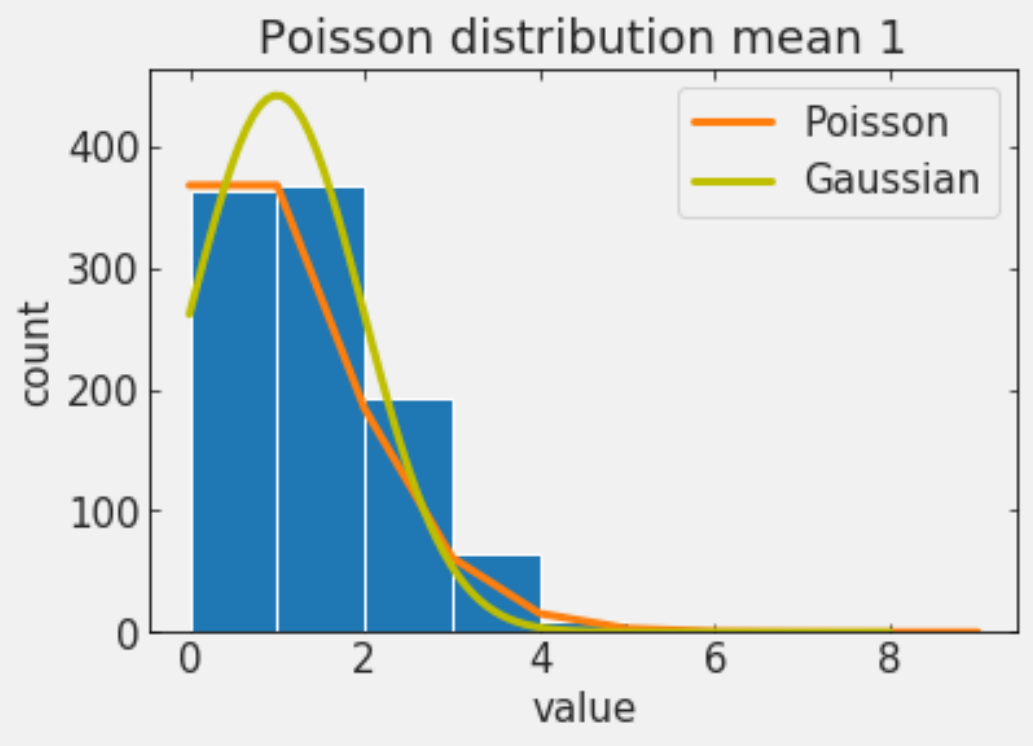
occurring in a fixed period of time if these events occur with a known average rate

and independently of the time since the last event.

stochastic or random or statistical errors

of particular interest are ***Poisson processes***

$$P(x) = \frac{e^{-\lambda}\lambda^{x}}{x!}$$



- asymmetric
- integer support
- support >0
- mean and stdev are related:

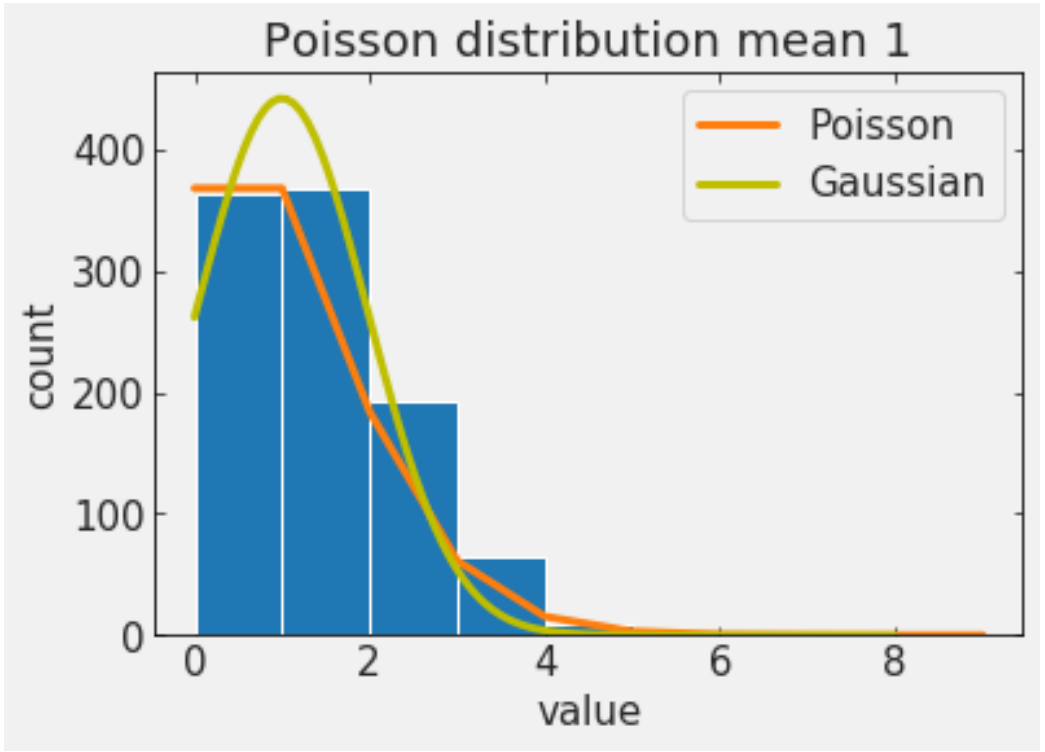
$$\mu : \lambda$$

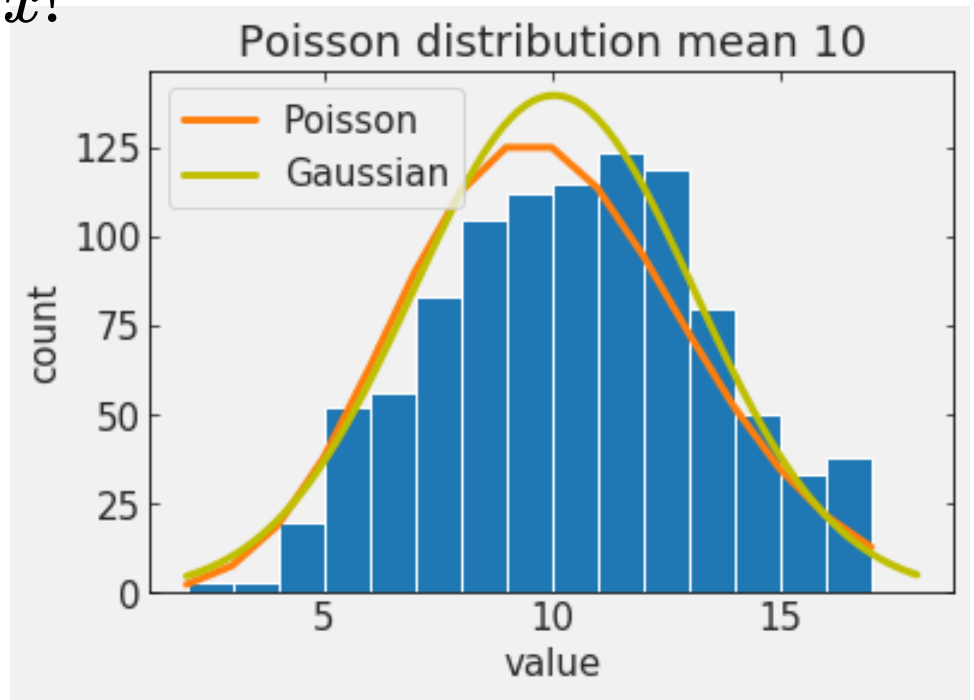
$$\sigma : \sqrt{\lambda}$$

stochastic or random or statistical errors

of particular interest are ***Poisson processes***

$$P(x) = \frac{e^{-\lambda}\lambda^{x}}{x!}$$

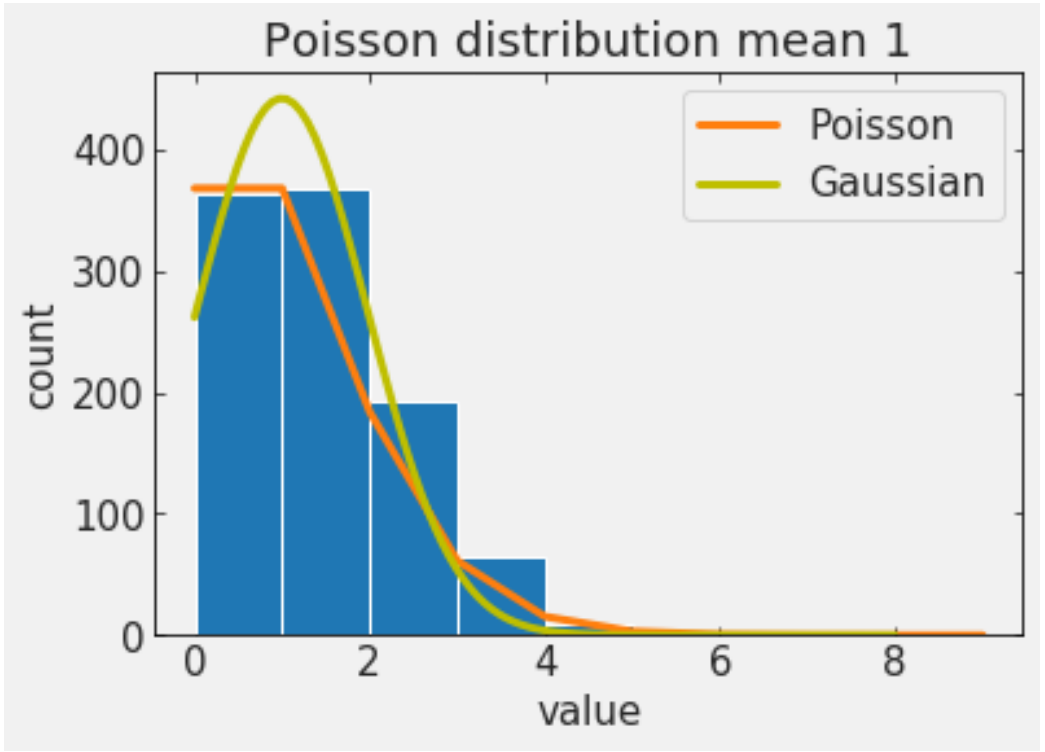




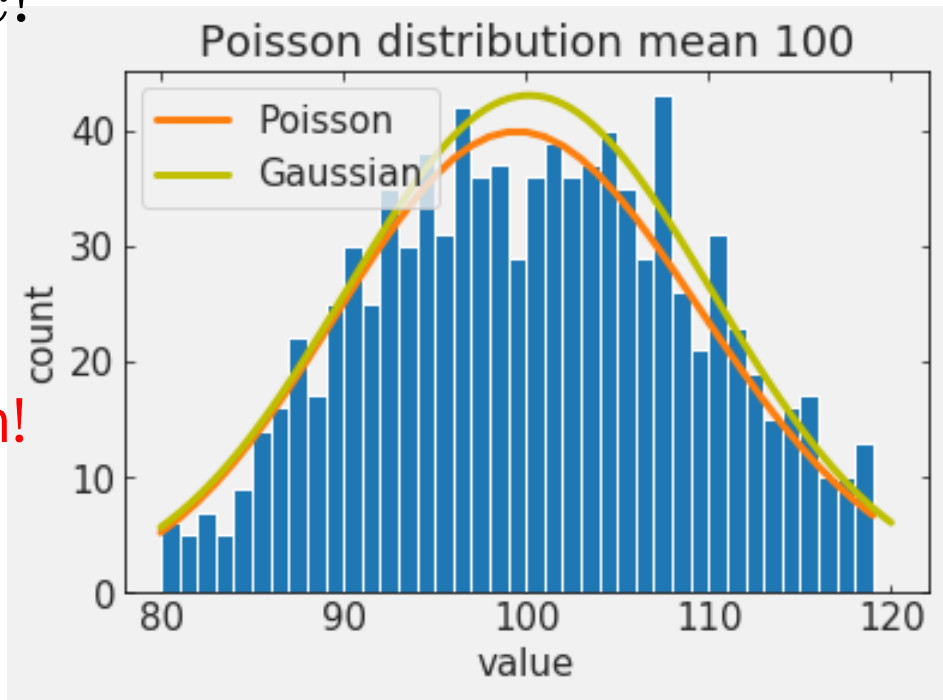
stochastic or random or statistical errors

of particular interest are ***Poisson processes***

$$P(x) = \frac{e^{-\lambda}\lambda^{x}}{x!}$$



as the mean increases it starts looking like a Gaussian!



stochastic or random or statistical errors

1) As the size of a sample tends to infinity the mean of the sample tends to the mean of the population (we already know that!!)

2) Count statistics follow a Poisson distribution with mean λ = N number of counts

if λ is large [redacted] Poisson ~ Gaussian

$$\hat{X} \sim N(\lambda, \sqrt{\lambda})$$

For inherently random phenomena that involve counting individual events or occurrences, we measure only a single number N. This kind of measurement is relevant to counting the number of radioactive decays in a specific time interval from a sample of material. It is also relevant to counting the number of Lutherans in a random sample of the population. The (absolute) uncertainty of such a single measurement, N, is estimated as the square root of N.

As example, if we measure 50 radioactive decays in 1 second we should present the result as 50±7 decays per second.

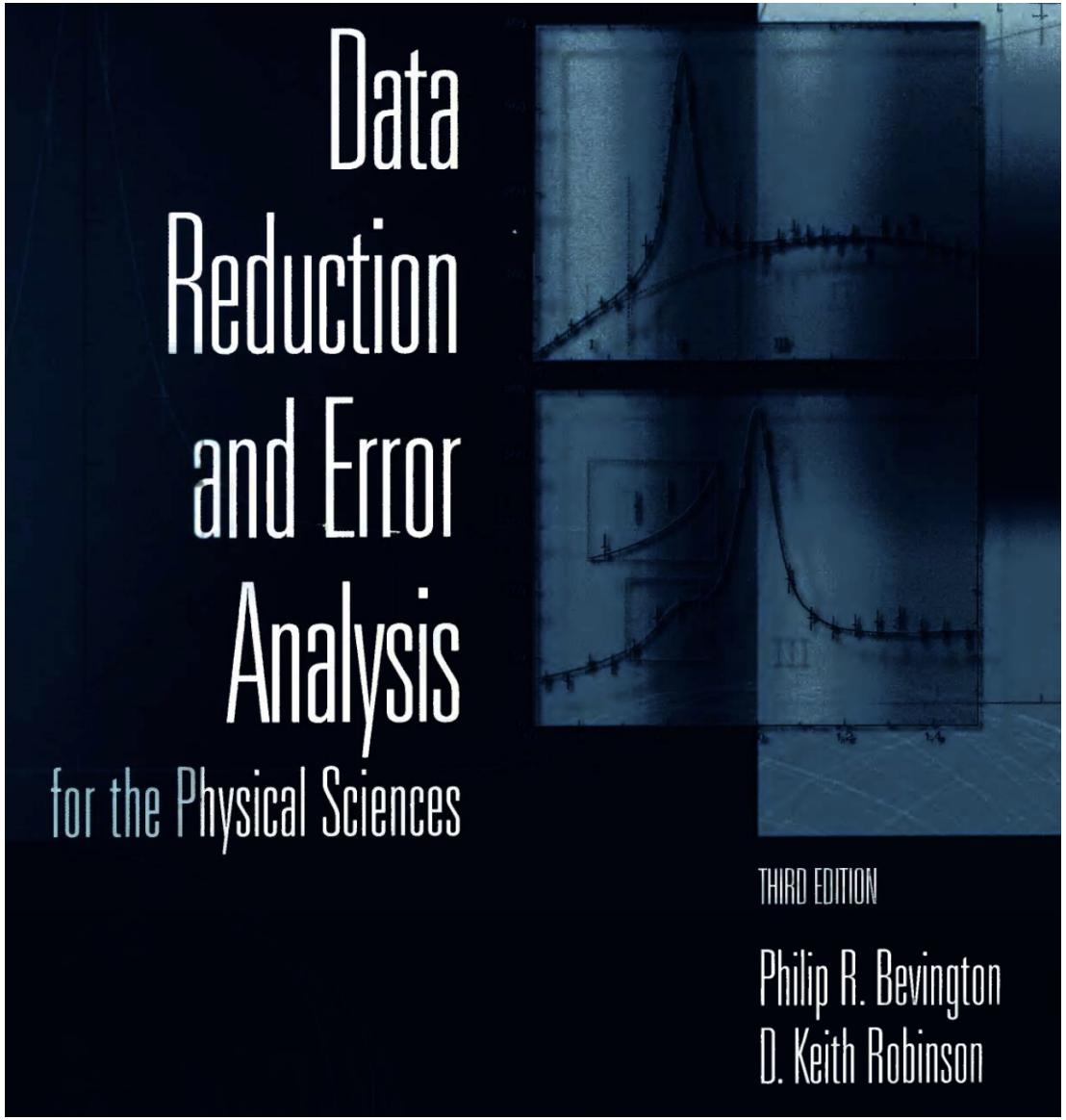
https://www.edouniversity.edu.ng/oerrepository/articles/phy_119_general_physics_practical_20182019.pdf

12 systematic uncertainties

systematic errors

reproducible inaccuracy introduced by faulty equipment, calibration, or technique.

http://hosting.astro.cornell.edu/academics/courses/astro3310/Books/Bevington_opt.pdf



systematic errors

reproducible inaccuracy introduced by faulty equipment, calibration, or technique.

~~2.5~~ 2.7 => 2.5 + 0.2 +/- 0.1

systematic errors

reproducible inaccuracy introduced by faulty equipment, calibration, or technique.

background, scanning efficiency, energy resolution, variation of counter efficiency with beam position, and energy, dead time, ...

~~2.5~~ 2.7 => 2.5 + 0.2 +/- 0.1

systematic errors

reproducible inaccuracy introduced by faulty equipment, calibration, or technique.

- Measurements are taken at 22 C with a steel rule calibrated at 15 C. This is a systematic *bias* and not a systematic *uncertainty*

~~2.5~~ 2.7 => 2.5 + 0.2 +/- 0.1

systematic errors

reproducible inaccuracy introduced by faulty equipment, calibration, or technique.

~~2.5~~ 2.7 => 2.5 + 0.2 +/- 0.1

systematic errors

reproducible inaccuracy introduced by faulty equipment, calibration, or technique.

~~2.5~~ 2.7 => 2.5 + 0.2 +/- 0.1

systematic errors

reproducible inaccuracy introduced by faulty equipment, calibration, or technique.

- Measurements are taken at 22 C with a steel rule calibrated at 15 C. This is a systematic *bias* and not a systematic *uncertainty*
- *Brightness* is known, distance is estimated accordingly. In space interstellar dust can make sources dimmer, but not brighter.
  systematic *uncertainty*

~~2.5~~ 2.7 => 2.5 + ? +/- 0.1

systematic errors

inaccuracy introduced by faulty equipment, calibration, or technique.

https://cuspuo.github.io/docs/dobler_urban_observatory.pdf

UO
CENTER FOR URBAN SCIENCE AND PROGRESS

systematic errors

inaccuracy introduced by faulty equipment,
calibration, or technique.

https://cuspuo.github.io/docs/dobler_urban_observatory.pdf

UO
CENTER FOR URBAN SCIENCE AND PROGRESS

systematic errors

inaccuracy introduced by faulty equipment,
calibration, or technique.

https://cuspuo.github.io/docs/dobler_urban_observatory.pdf

UO
CENTER FOR URBAN SCIENCE AND PROGRESS

Bias in measurements: know your data

Undercoverage bias

the surveyed segment of the population is lower in a sample than it is in the population. This can happen because the frame used to obtain the sample is incomplete or not representative of the population.

Bias in measurements: know your data

Publication Bias

Bias in measurements: know your data

Publication Bias

His team investigated the fate of 221 sociological studies conducted between 2002 and 2012, which were recorded by Time-sharing Experiments for the Social Sciences (TESS), a US project that helps social scientists to carry out large-scale surveys of people's views.

Only 48% of the completed studies had been published. So the team contacted the remaining authors to find out whether they had written up their results, or submitted them to a journal or conference. They also asked whether the results supported the researchers' original hypothesis.

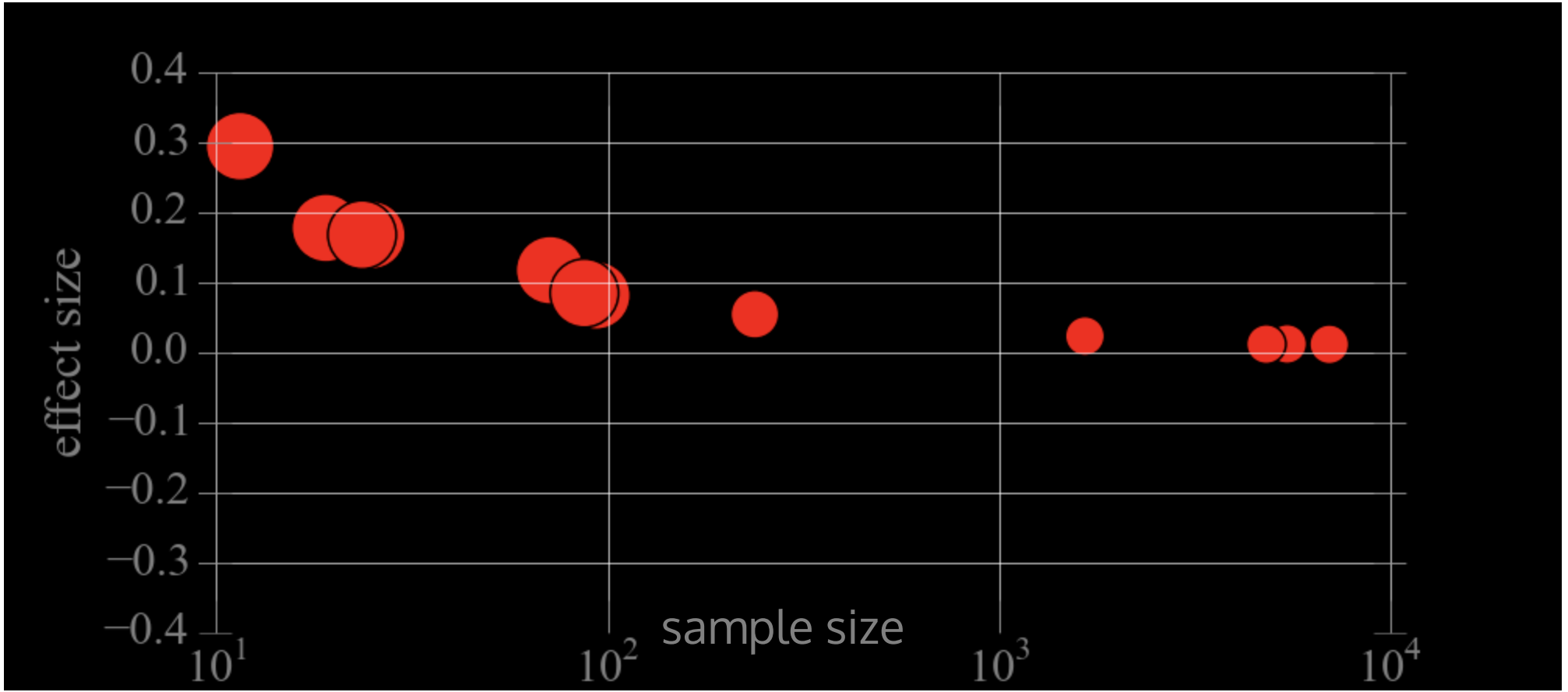
Of all the null studies, just 20% had appeared in a journal, and 65% had not even been written up. By contrast, roughly 60% of studies with strong results had been published. Many of the researchers contacted by Malhotra's team said that they had not written up their null results because they thought that journals would not publish them, or that the findings were neither interesting nor important enough to warrant any further effort.

Bias in measurements: know your data

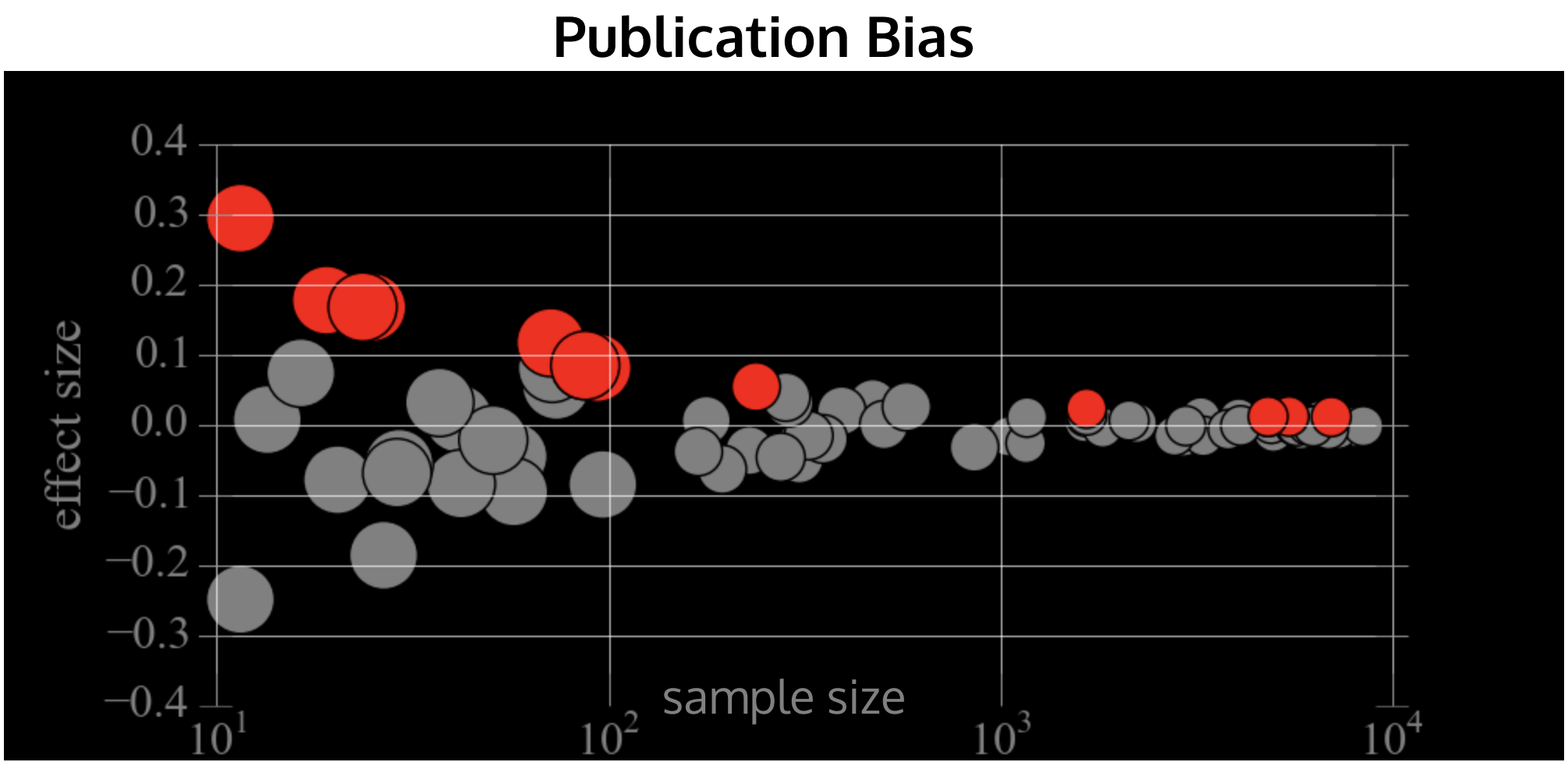
What is effect size? magnitude of the experimental effect.

e.g. difference between means

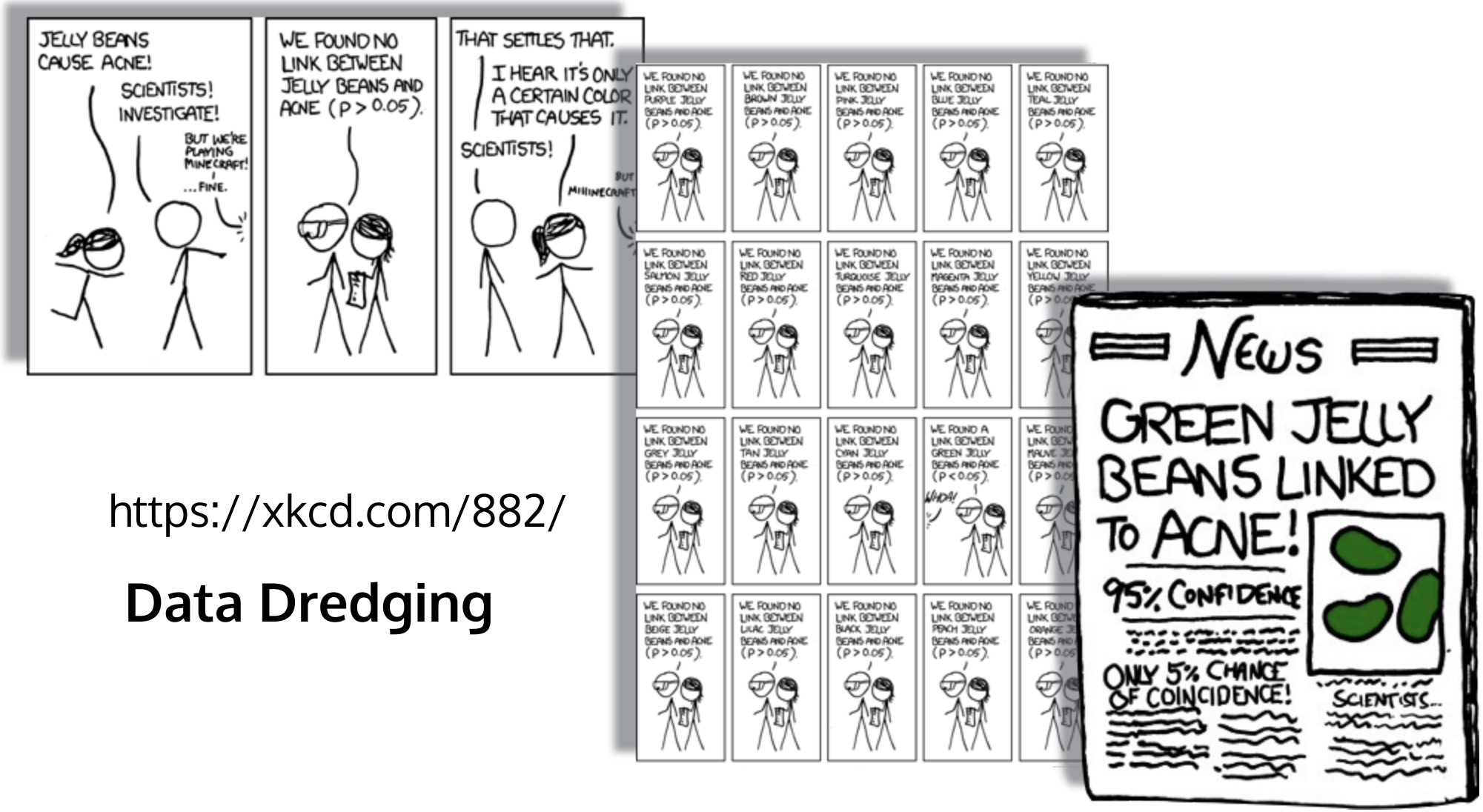
Publication Bias



Bias in measurements: know your data



Bias in measurements: know your data



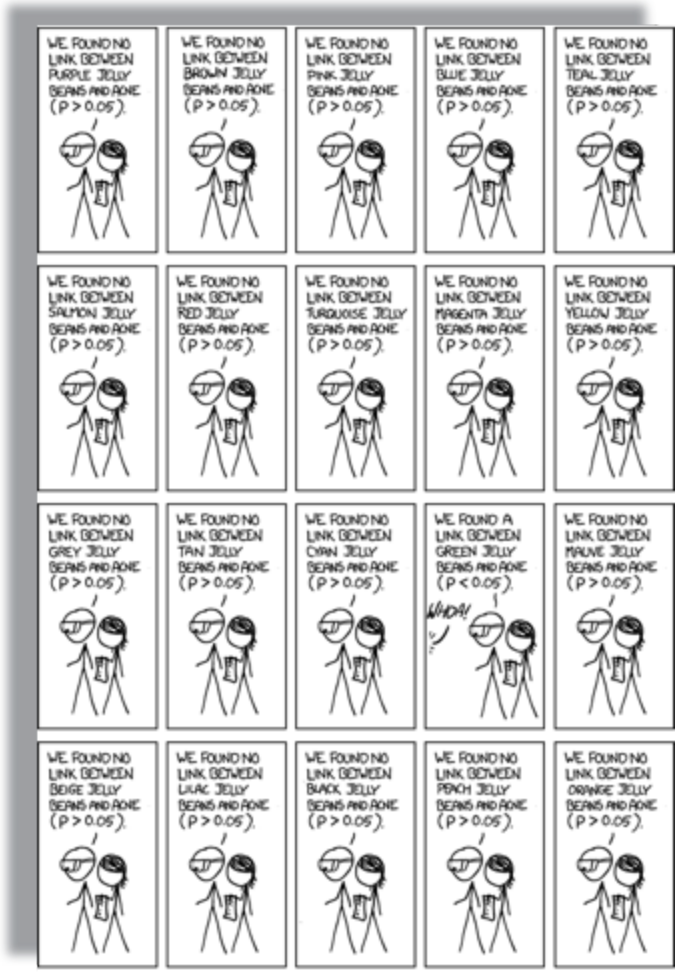
https://xkcd.com/882/

Data Dredging

Bias in measurements: know your data

https://xkcd.com/882/

Data Dredging



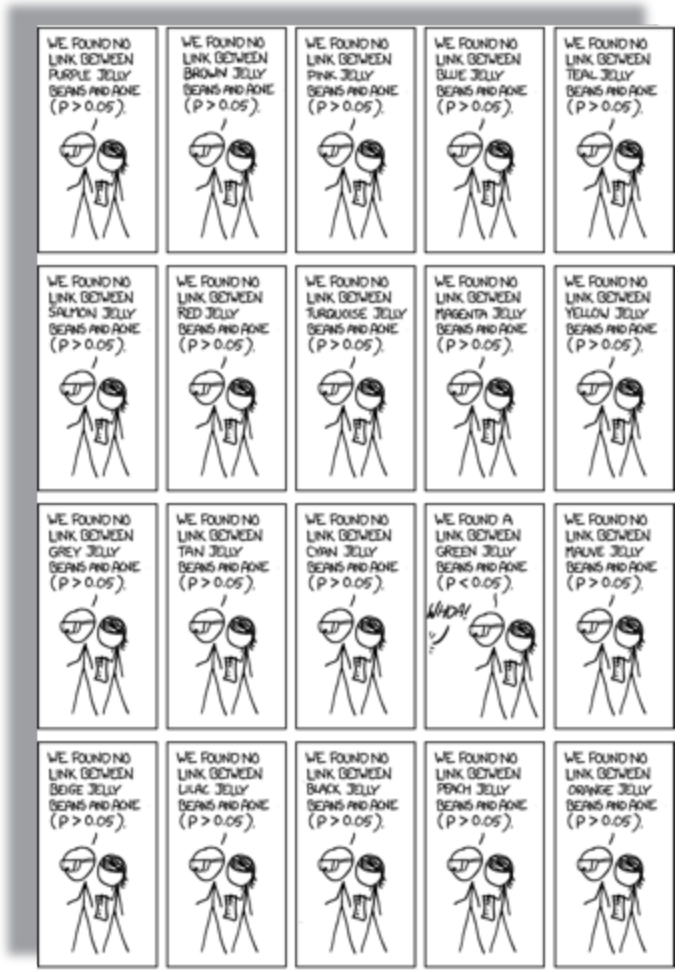
Bias in measurements: know your data

each test has a
probability p<=0.05
of Type I error
significance 95%

20 tests are preformed

assume independence:
if pi = 0.05 for each i=1..20

Data Dredging

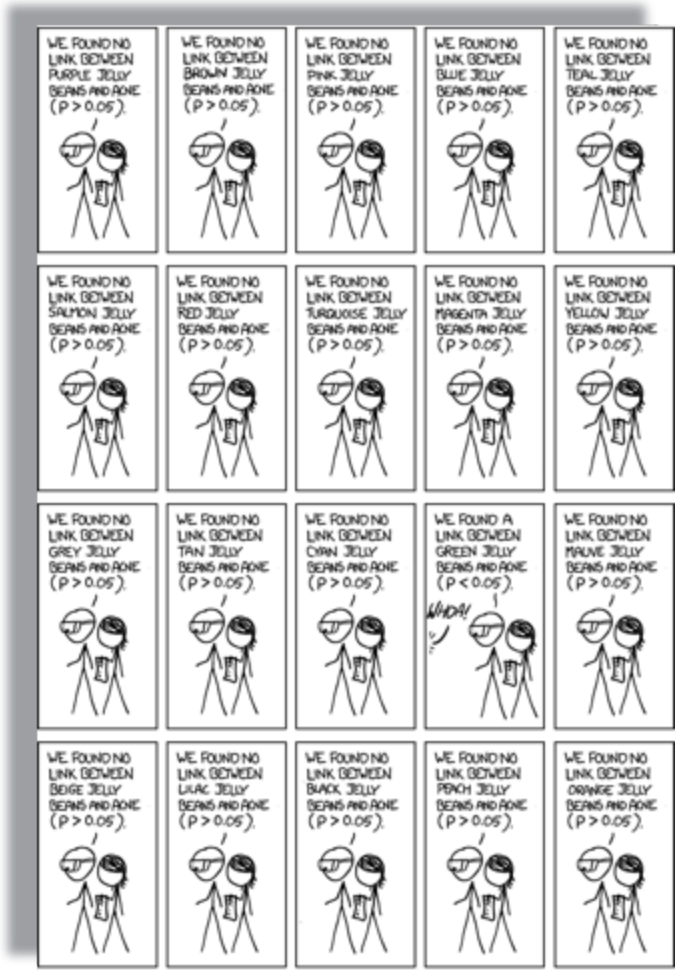


Bias in measurements: know your data

each test has a
probability p<=0.05
of Type I error
significance 95%

20 tests are preformed

assume independence:
if p_i = 0.05 for each i=1..20



Data Dredging
(also called *p*-hacking)

$$p_{tot} = \sum_i p_i$$

$$p_{tot} = 20 * p_i = 1$$

Stochastic vs Systematics

Statistical	**Systematic**
No preferred direction	Biases the measurement *in one direction*

Stochastic vs Systematics

Statistical	**Systematic**
No preferred direction	Biases the measurement *in one direction*

How can you mitigate these errors??

Stochastic vs Systematics

Statistical	**Systematic**
No preferred direction	Biases the measurement *in one direction*
Shrinks with the sample size (typically as $\sqrt{N}$)	Affects the sample regardless of the size
Typically Gaussian or Poisson	Who knows??? But we often pretend its gaussian...

Stochastic vs Systematics

Statistical	**Systematic**
No preferred direction	Biases the measurement *in one direction*
Shrinks with the sample size (typically as $\sqrt{N}$)	Affects the sample regardless of the size
Typically Gaussian or Poisson	Who knows??? But we often pretend its gaussian...

but.... WHY??

8.044 | Spring 2013 | Undergraduate

Statistical Physics I

Sums of a Random Variables

4 Sums of Random Variables

To find the probability density for the sum of two statistically independent random variables one can multiply the Fourier transforms of the individual probability densities and take the inverse transform of the product. As a practical application and example one can show that the sums of Gaussians are Gaussian, sums of Poisson variables are Poisson, and sums of Lorentzians are Lorentzian.

Gaussian

$$\frac{1}{\sqrt{2\pi\sigma^2}}\exp[-\frac{(x-E)^2}{2\sigma^2}] \leftrightarrow \exp[ikE-\frac{\sigma^2k^2}{2}]$$

Poisson

$$\sum_{n=0}^{\infty}\frac{1}{n!}\lambda^n e^{-\lambda}\delta(x-n) \leftrightarrow \exp[\lambda(e^{ik}-1)]$$

Lorentzian

$$\frac{\Gamma}{\pi}\frac{1}{(x-m)^2+\Gamma^2} \leftrightarrow \exp[imk-|k\Lambda|]$$

Each of these three transforms $F(k)$ has the property that a product of similar functions preserves the functional form; only the parameters change. For example if $p_G(S)$ represents the sum of two Gaussians, then

$$p_G(S) \leftrightarrow \exp[ikE_1-\frac{\sigma_1^2k^2}{2}]\exp[ikE_2-\frac{\sigma_2^2k^2}{2}]$$

$$\leftrightarrow \exp[ik(E_1+E_2)-\frac{(\sigma_1^2+\sigma_2^2)k^2}{2}]$$

The last expression is the Fourier transform of a Gaussian of mean E_1+E_2 and variance $\sigma_1^2+\sigma_2^2$. Check to see that the transforms for the Poisson and Lorentzian behave in a similar manner.

2 combining uncertainties

combining uncertainties

If ***x,...,w*** are measured with ***independent*** and ***random*** uncertainties

Δx,...,Δw the uncertainty in a linear combination of ***x,...,w*** is the quadratic sum:

combining uncertainties

If ***x,…,w*** are measured with ***independent*** and ***random*** uncertainties

Δx,…,Δw the uncertainty in a linear combination of ***x,…,w*** is the quadratic sum:

Addition/Subtraction	$z = x \pm y$	$\Delta z = \sqrt{(\Delta x)^2 + (\Delta y)^2}$
Multiplication	$z = xy$	$\Delta z = \lvert xy \rvert \sqrt{\left(\frac{\Delta x}{x}\right)^2 + \left(\frac{\Delta y}{y}\right)^2}$
Division	$z = \frac{x}{y}$	$\Delta z = \left\lvert \frac{x}{y} \right\rvert \sqrt{\left(\frac{\Delta x}{x}\right)^2 + \left(\frac{\Delta y}{y}\right)^2}$
Power	$z = x^n$	$\Delta z = \lvert n \rvert x^{n-1} \Delta x$
Multiplication by a Constant	$z = cx$	$\Delta z = \lvert c \rvert \Delta x$
Function	$z = f(x, y)$	$\Delta z = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 (\Delta x)^2 + \left(\frac{\partial f}{\partial y}\right)^2 (\Delta y)^2}$

combining uncertainties

If ***x,y,..,w*** are measured with ***independent*** and ***random*** uncertainties

Δx,Δy,...,Δw the uncertainty in a linear combination of ***x,y,...,w*** is the quadratic sum:

$$f(x, y, \ldots w) :$$

$$\Delta_f = \sqrt{\left(\frac{\partial f}{\partial x}^2\right) \Delta_x^2 + \left(\frac{\partial f}{\partial y}^2\right) \Delta_y^2 + \ldots + \left(\frac{\partial f}{\partial w}^2\right) \Delta_w^2}$$

http://ipl.physics.harvard.edu/wp-uploads/2013/03/PS3_Error_Propagation_sp13.pdf

combining uncertainties

derivation

$$f_k = \sum_{i=1}^{n} A_{ki} x_i \text{ or } \mathrm{f} = \mathrm{Ax}$$

$$\Sigma^x = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} & \cdots \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} & \cdots \\ \sigma_{13} & \sigma_{23} & \sigma_3^2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} = \begin{pmatrix} \Sigma_{11}^x & \Sigma_{12}^x & \Sigma_{13}^x & \cdots \\ \Sigma_{12}^x & \Sigma_{22}^x & \Sigma_{23}^x & \cdots \\ \Sigma_{13}^x & \Sigma_{23}^x & \Sigma_{33}^x & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

combining uncertainties

derivation

$$f_k = \sum_{i=1}^{n} A_{ki} x_i \text{ or } \mathrm{f} = \mathrm{Ax}$$

$$\Sigma^x = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} & \cdots \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} & \cdots \\ \sigma_{13} & \sigma_{23} & \sigma_3^2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} = \begin{pmatrix} \Sigma_{11}^x & \Sigma_{12}^x & \Sigma_{13}^x & \cdots \\ \Sigma_{12}^x & \Sigma_{22}^x & \Sigma_{23}^x & \cdots \\ \Sigma_{13}^x & \Sigma_{23}^x & \Sigma_{33}^x & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$\Sigma_{ij}^f = \sum_{k}^{n} \sum_{\ell}^{n} A_{ik} \Sigma_{k\ell}^x A_{j\ell}$$

combining uncertainties

derivation

$$f_k = \sum_{i=1}^{n} A_{ki} x_i \text{ or } \mathrm{f} = \mathrm{Ax}$$

$$\Sigma^x = \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} & \cdots \\ \sigma_{12} & \sigma_2^2 & \sigma_{23} & \cdots \\ \sigma_{13} & \sigma_{23} & \sigma_3^2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} = \begin{pmatrix} \Sigma_{11}^x & \Sigma_{12}^x & \Sigma_{13}^x & \cdots \\ \Sigma_{12}^x & \Sigma_{22}^x & \Sigma_{23}^x & \cdots \\ \Sigma_{13}^x & \Sigma_{23}^x & \Sigma_{33}^x & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$\Sigma = \mathrm{A} \Sigma^x \mathrm{A}^\top$$

combining uncertainties

derivation

$$f_k = \sum_{i=1}^{n} A_{ki} x_i \text{ or } \mathrm{f} = \mathrm{Ax}$$

$$\Sigma^x = \begin{pmatrix} \sigma_1^2 & 0 & 0 & \cdots \\ 0 & \sigma_2^2 & 0 & \cdots \\ 0 & 0 & \sigma_3^2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} = \begin{pmatrix} \Sigma_{11}^x & 0 & 0 & \cdots \\ 0 & \Sigma_{22}^x & 0 & \cdots \\ 0 & 0 & \Sigma_{33}^x & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

$$\Sigma_{ij}^f = \sum_{k}^{n} A_{ik} \Sigma_k^x A_{jk}$$

sum in quadrature:

$$\Delta_f = \sqrt{\left(\frac{\partial f}{\partial x}^2\right) \Delta_x^2 + \left(\frac{\partial f}{\partial y}^2\right) \Delta_y^2 + \ldots}$$

combining uncertainties

derivation

$$f_k = \sum_{i=1}^{n} A_{ki} x_i \text{ or } \mathrm{f} = \mathrm{Ax}$$

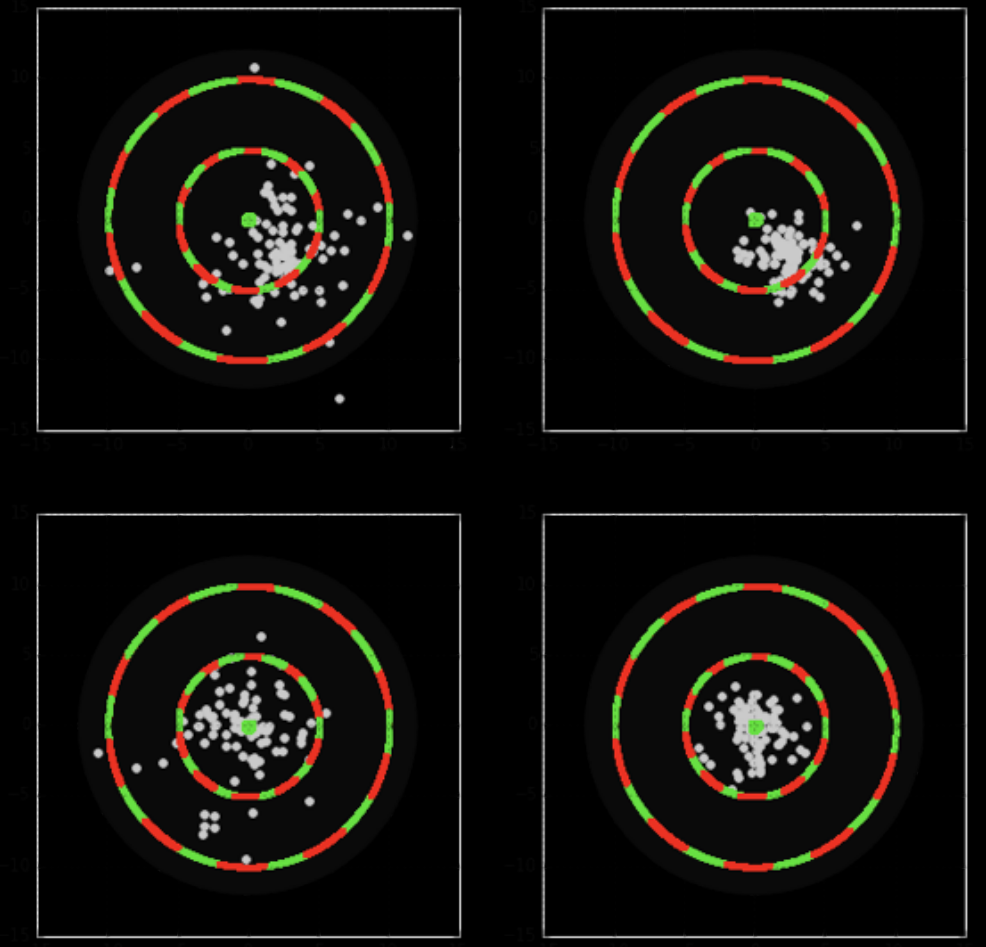
$$f = \sum_{i}^{n} a_i x_i : f = \mathrm{a}x$$

$$\sigma_f^2 = \sum_{i}^{n} \sum_{j}^{n} a_i \Sigma_{ij}^x a_j = \mathrm{a} \Sigma^x \mathrm{a}^\top$$

Precision vs Accuracy

Precision

Accuracy



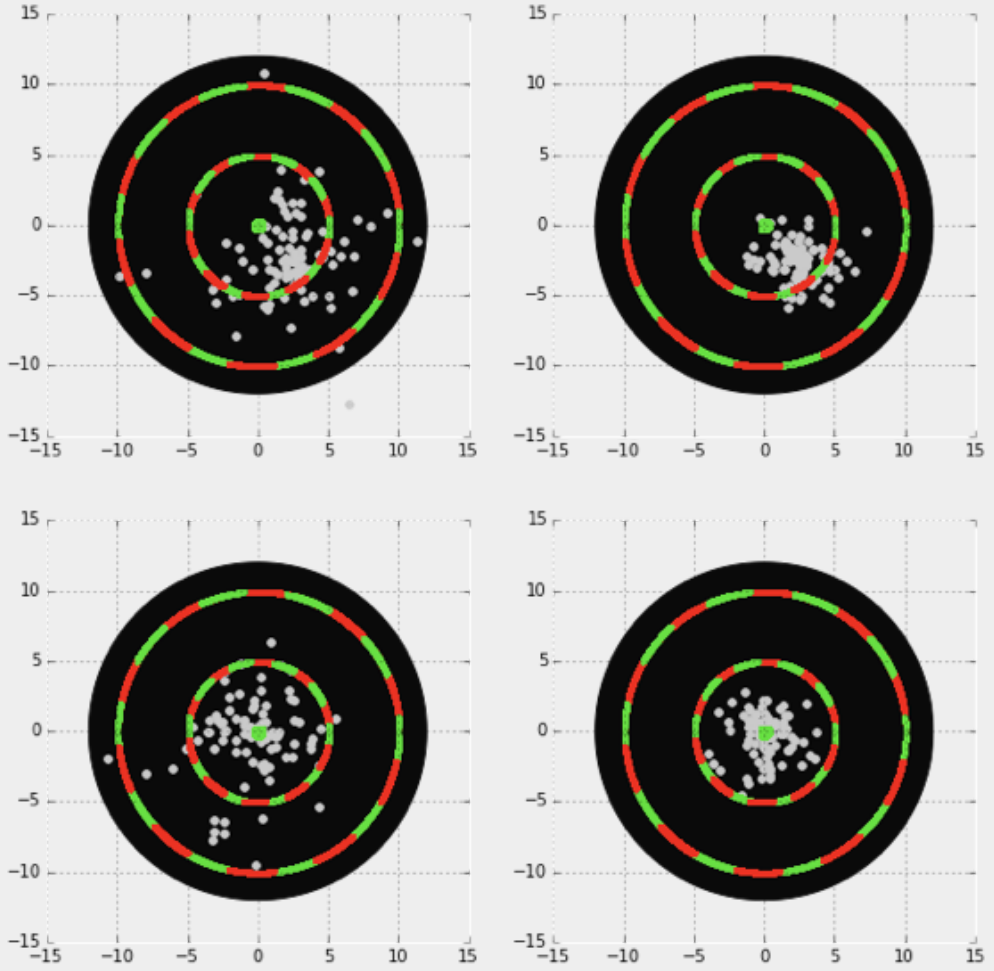
Precision vs Accuracy

which relates to stochastic errors,

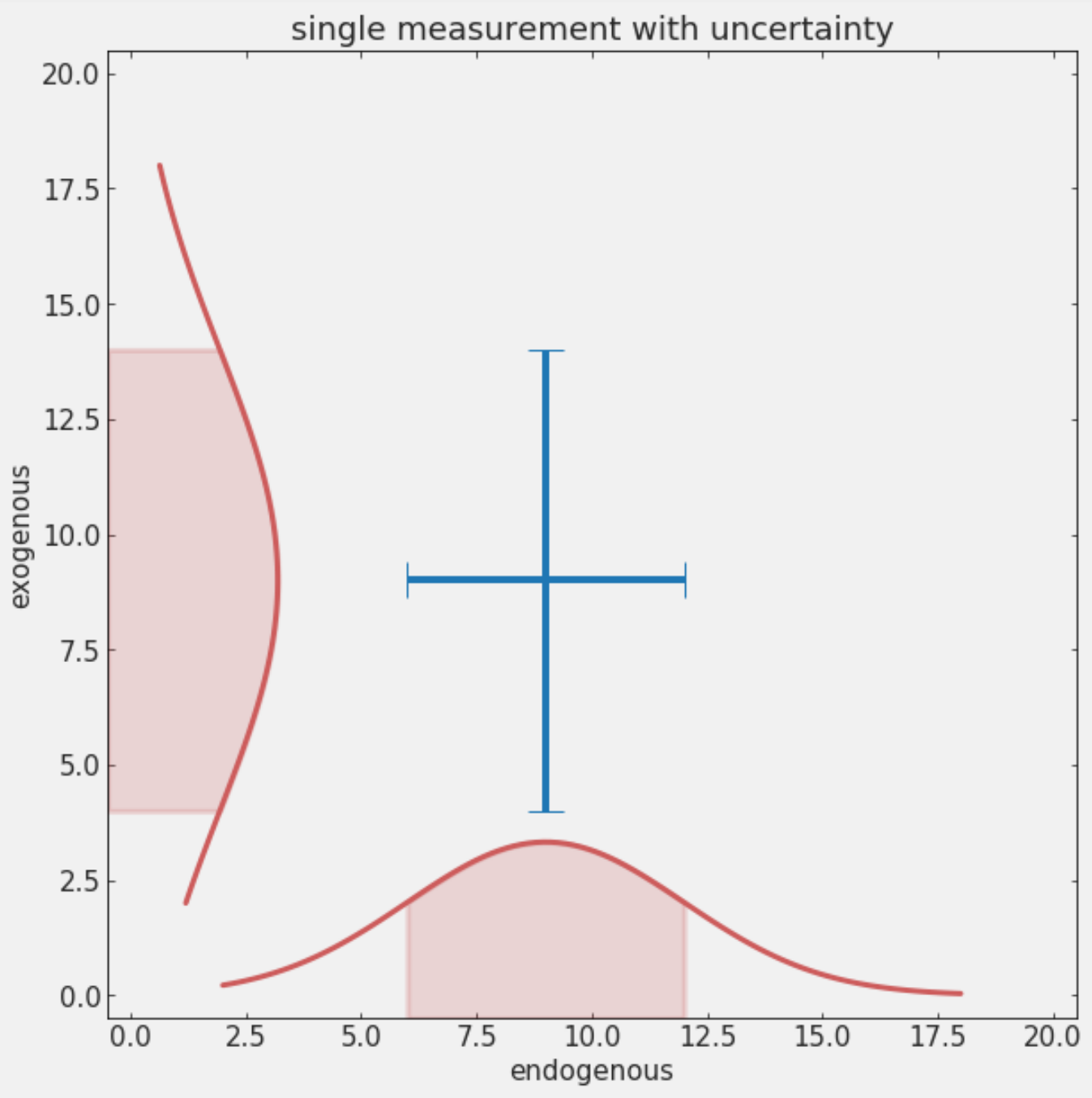
which to stsystematic?

Precision

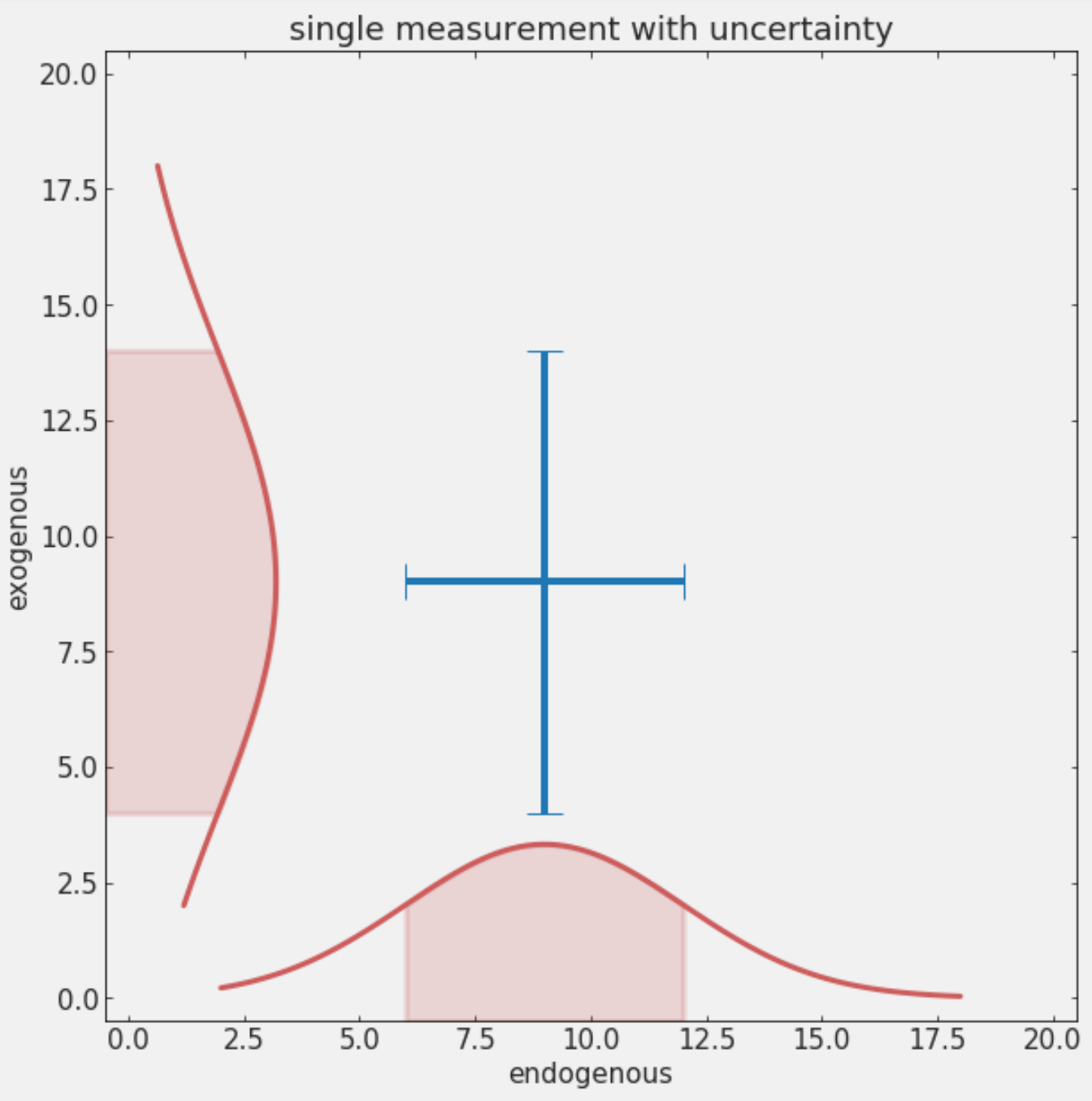
Accuracy



the meaning of "uncertainty"



the meaning of "uncertainty" when reporting a result



the meaning of "uncertainty" when reporting a result

CMB cosmology

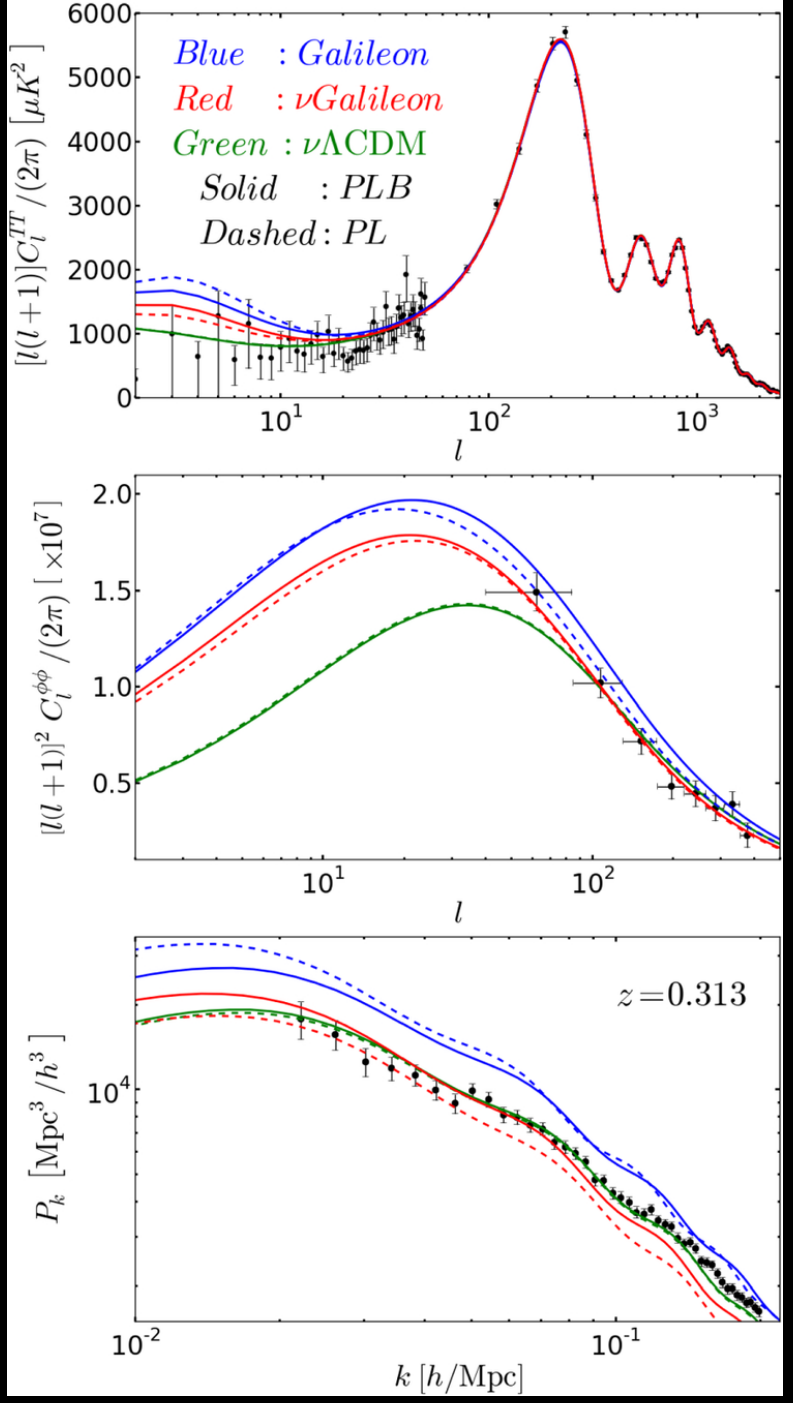
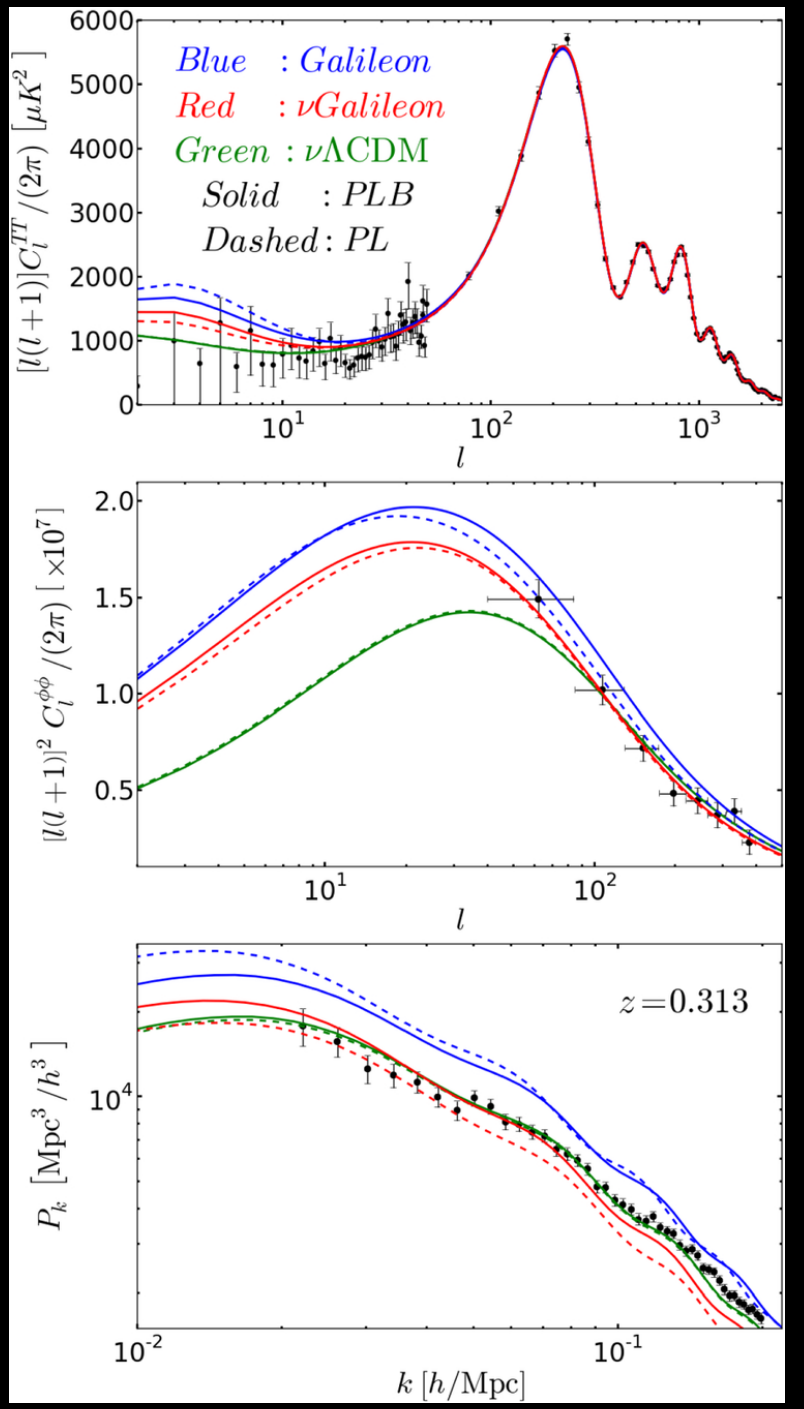


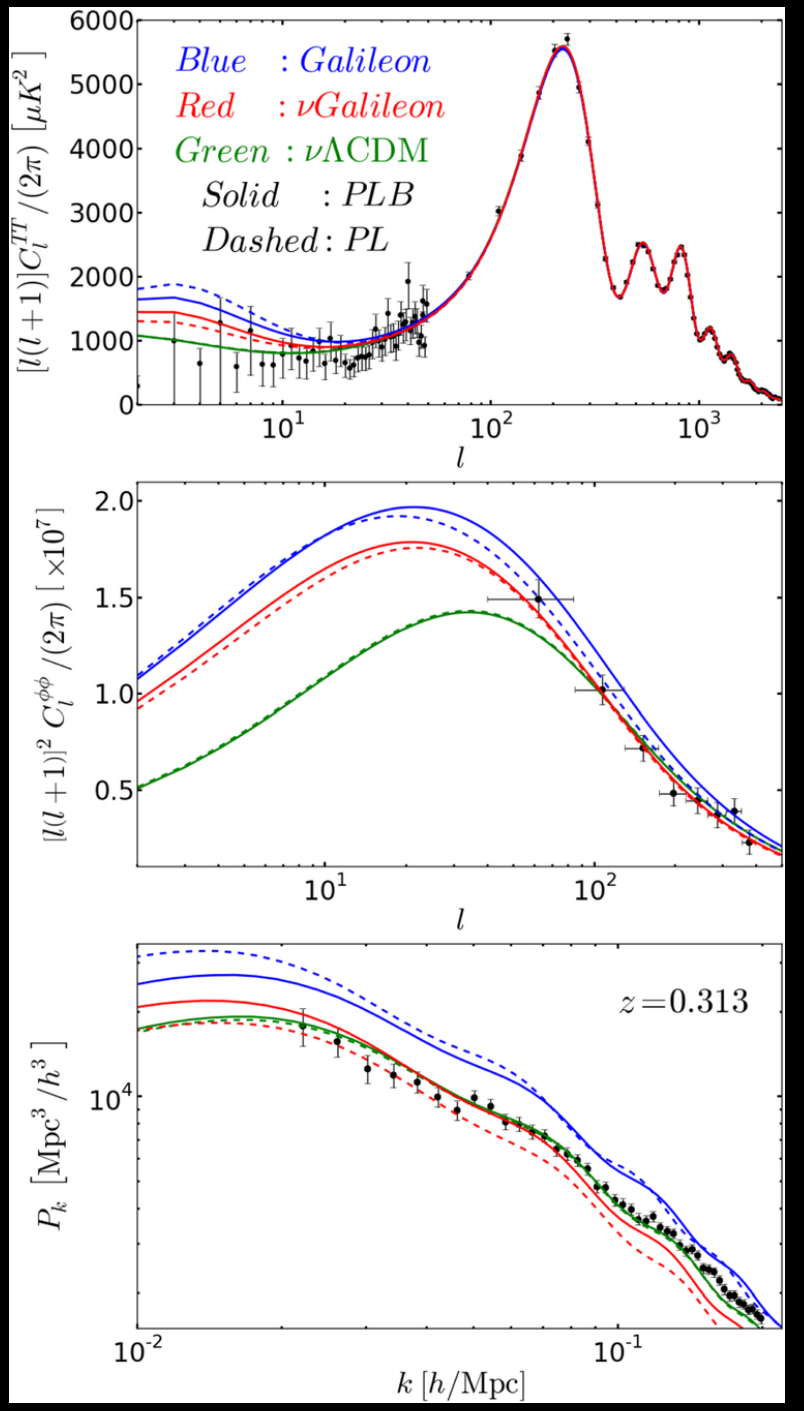
Fig 4: These plots illustrate the differences between ΛCDM and Galileon models (see Sect. 7.3.1), with and without massive neutrinos. The Galileon models have background Friedmann equations that contain a scalar-field energy density contribution that generates late time cosmic acceleration and has an evolution consistent with observations and thus similar to that of a ΛCDM model. The Galileon scalar field here also affects linear perturbations and is not coupled to matter. The effect of the Galileon field considered here is focused on large-scale structure. The Top: CMB temperature power spectra showing the ISW effect at low multipoles. Middle: CMB lensing potential spectra. Bottom: linear matter power spectra. The models plotted in dashed lines indicate their best fit models to Ade et al. (2014c) temperature data, WMAP9 polarization data (Hinshaw et al. 2013), and Planck-2013 CMB lensing (Ade et al. 2014d). https://link.springer.com/article/10.1007/s41114-018-0017-4

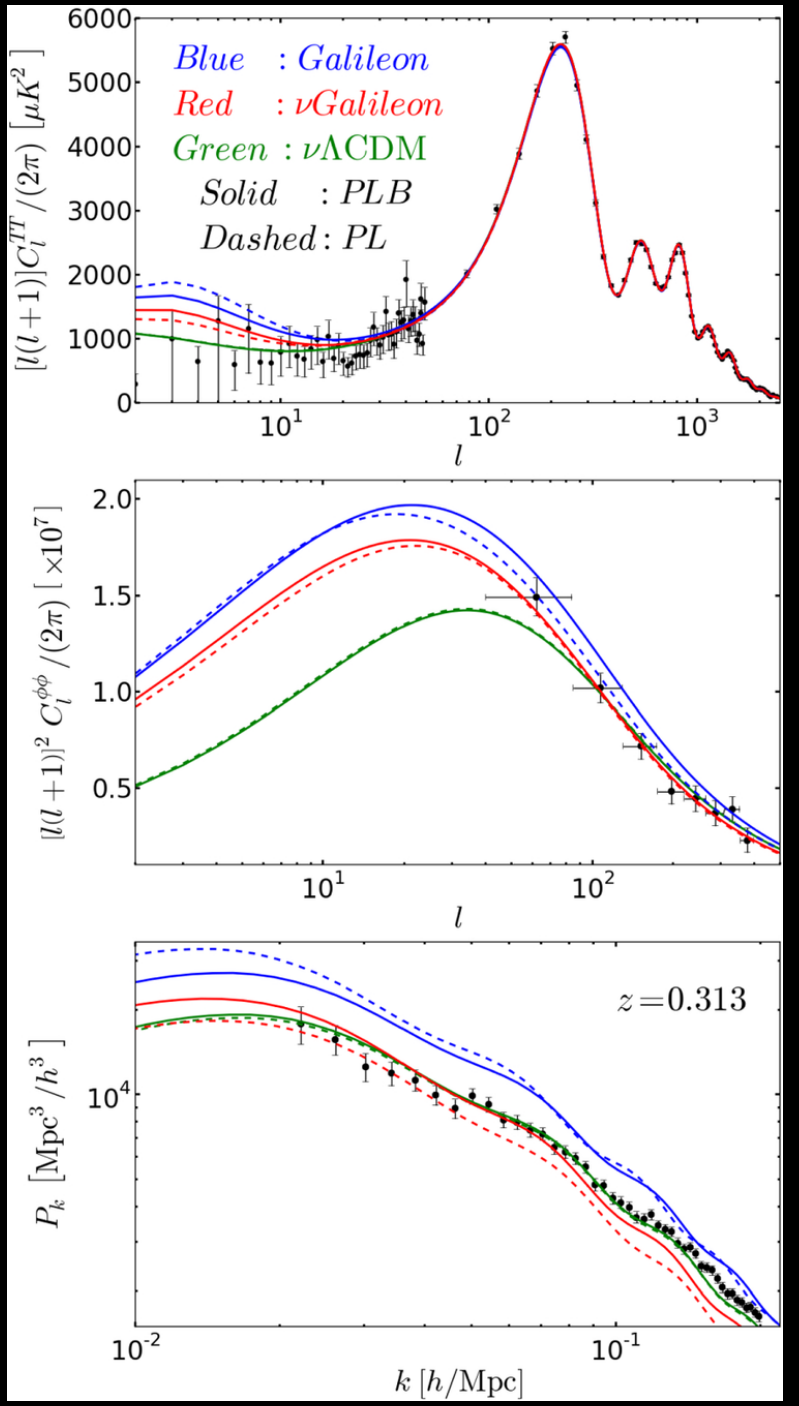


They note these as PL models. The solid lines indicate their best fits to CMB data (i.e., PL) plus BAO measurements from 6dF, SDSS DR7 and BOSS DR9. They note these as PLB models. The models correspond to best-fitting base Galileon modified gravity model (in blue), vGalileon (in red) and vΛCDM (in green). For the last two models, the authors added massive neutrino. In the upper and middle panels, the data points show the power spectrum measured by the Planck satellite (Ade et al. 2014c). In the lower panel, the data points show the SDSS-DR7 Luminous Red Galaxy power spectrum of Reid et al. (2010), but scaled down to match the amplitude of the best-fitting vGalileon (PLB) model (Barreira et al. 2014a). We refer to this figure from various parts of the text

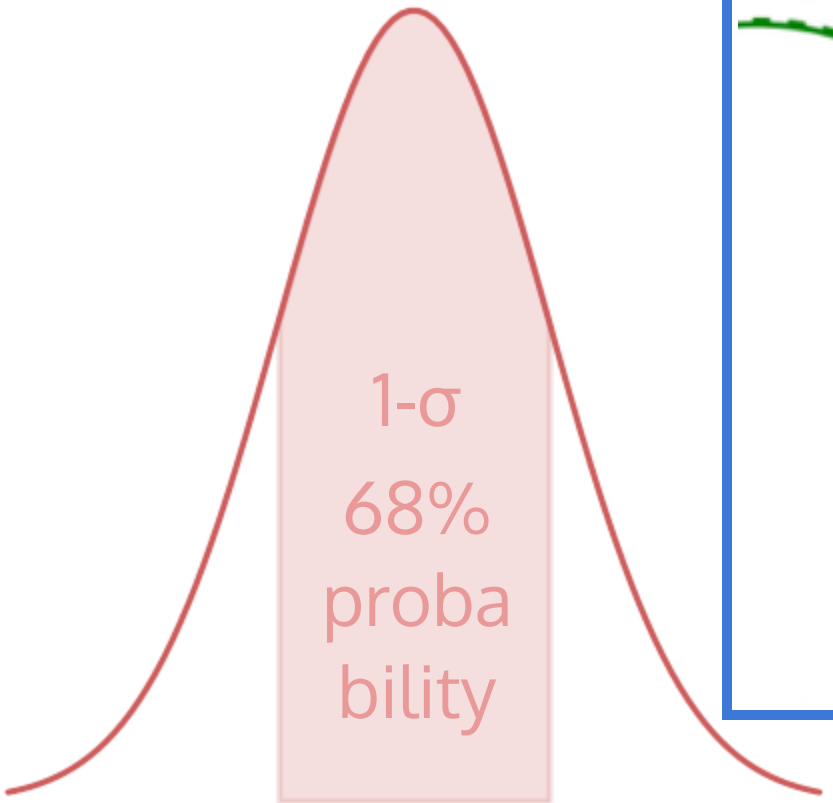
Image from Barreira et al. (2014).

https://link.springer.com/article/10.1007/s41114-018-0017-4

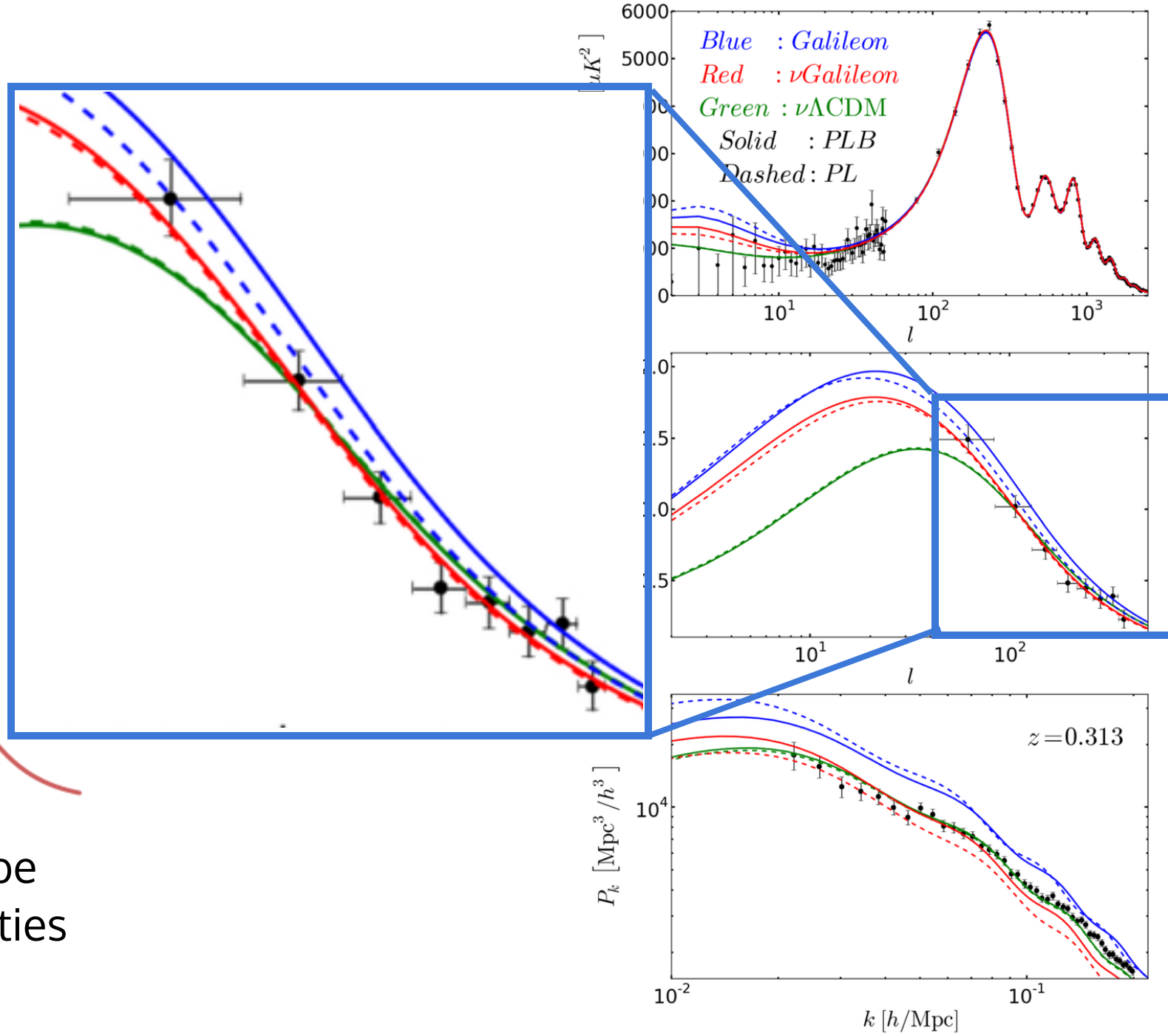



Blue : Galileon
Red : νGalileon
Green : νΛCDM
Solid : PLB
Dashed : PL
$[l(l+1)]C_l^{TT}/(2\pi)\ [\mu K^2]$
l
$[l(l+1)]^2\ C_l^{\phi\phi}/(2\pi)\ [\times 10^7]$
l
$z=0.313$
$P_k\ [\mathrm{Mpc}^3/h^3]$
$k\ [h/\mathrm{Mpc}]$

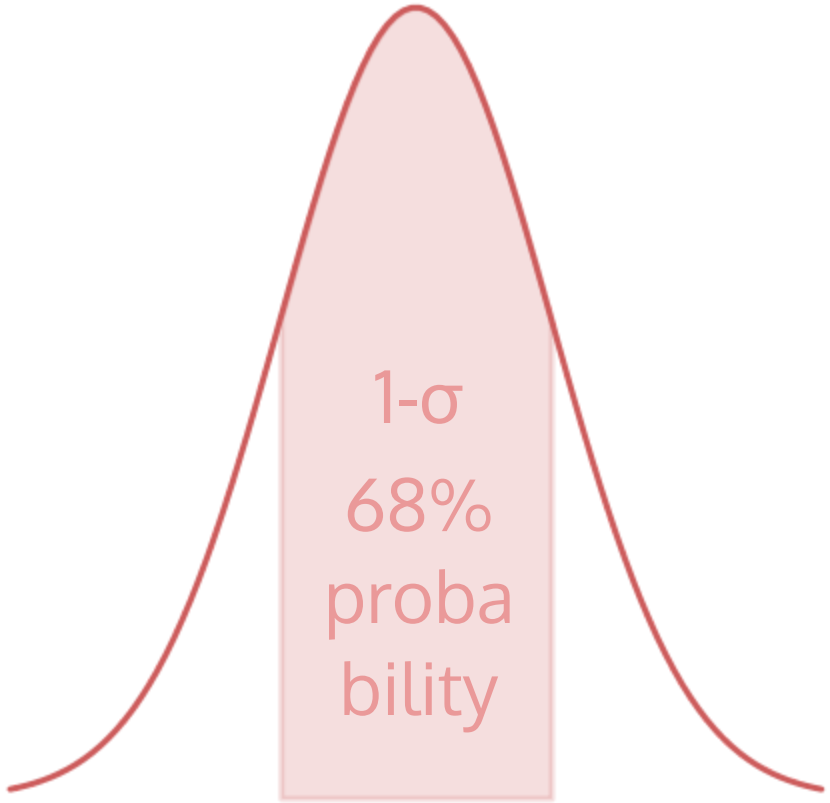
uncertainties: 1-σ



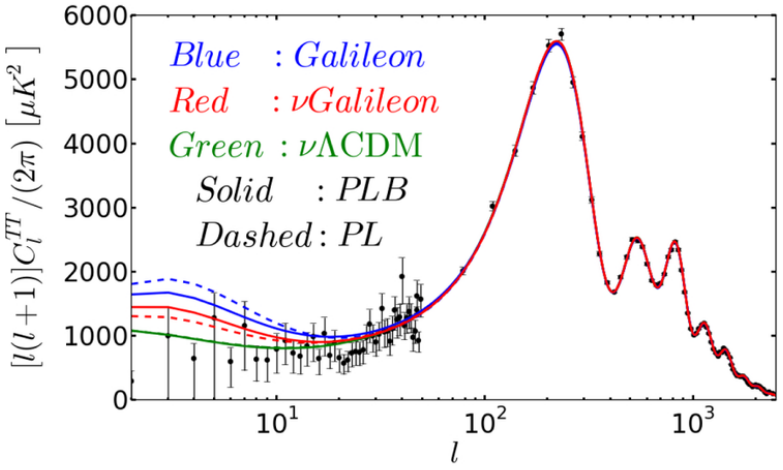
3 points out of 10 can be outside of the uncertainties

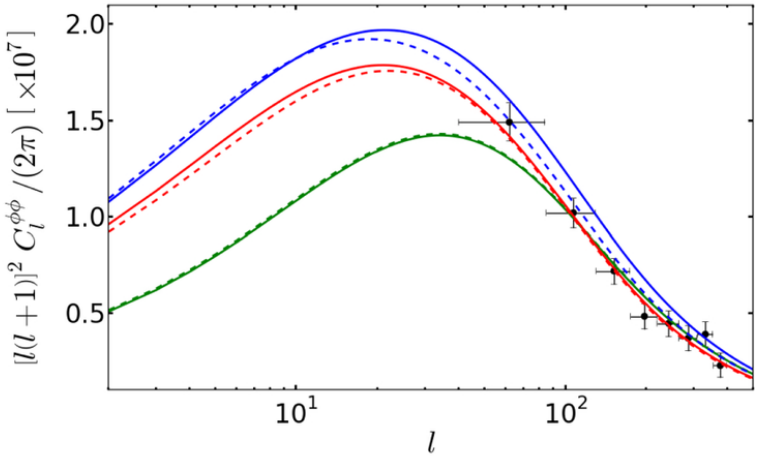


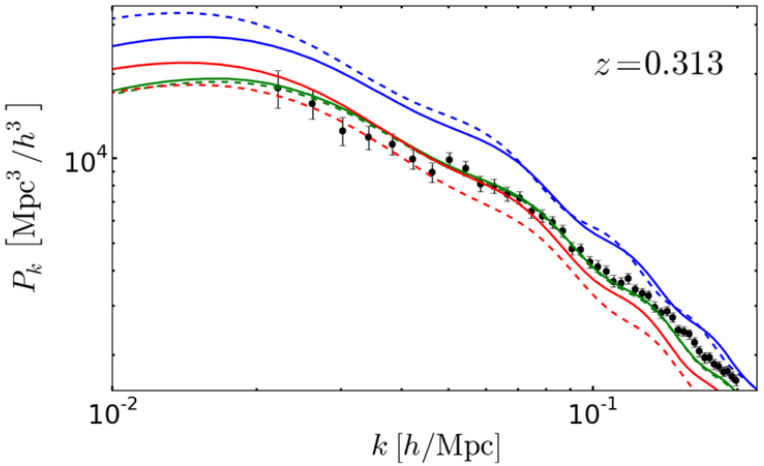
uncertainties: 1-σ



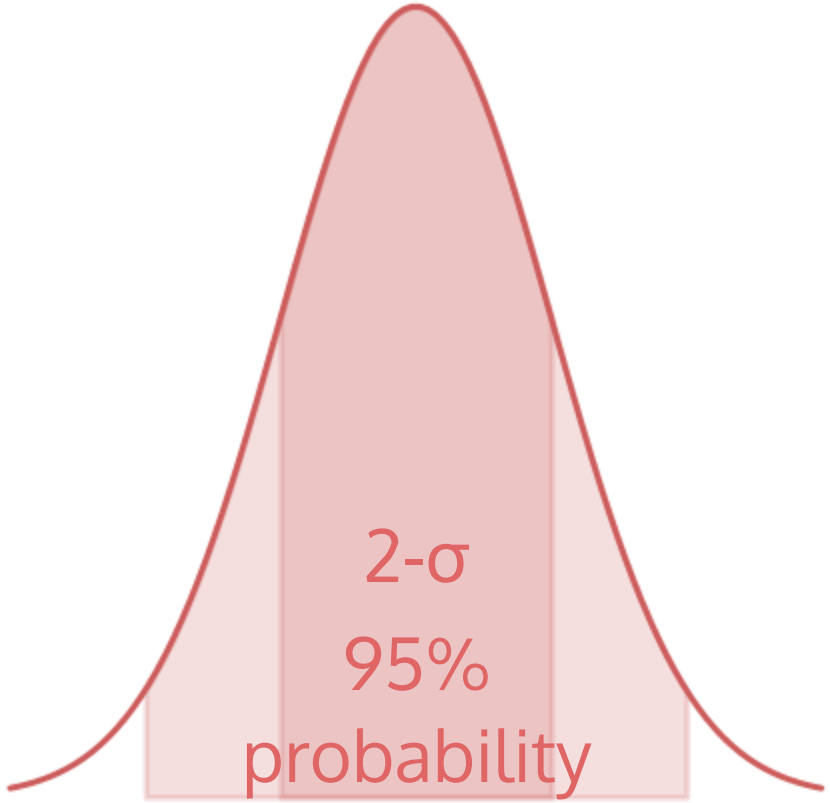
3 points out of 10 can be outside of the uncertainties



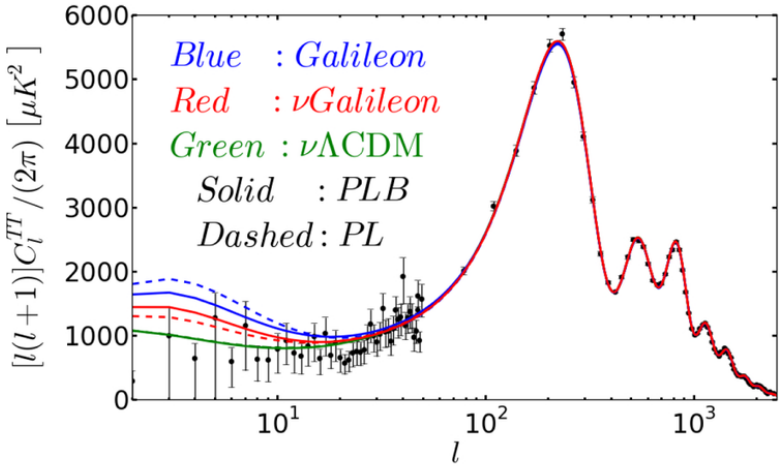


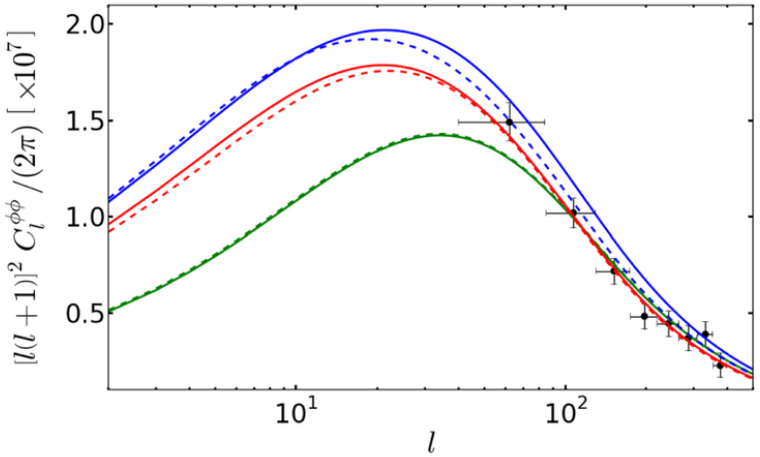


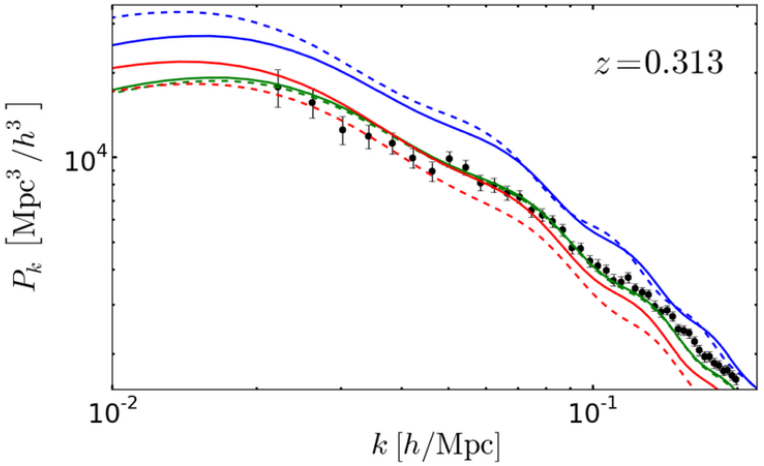
uncertainties: 1-σ



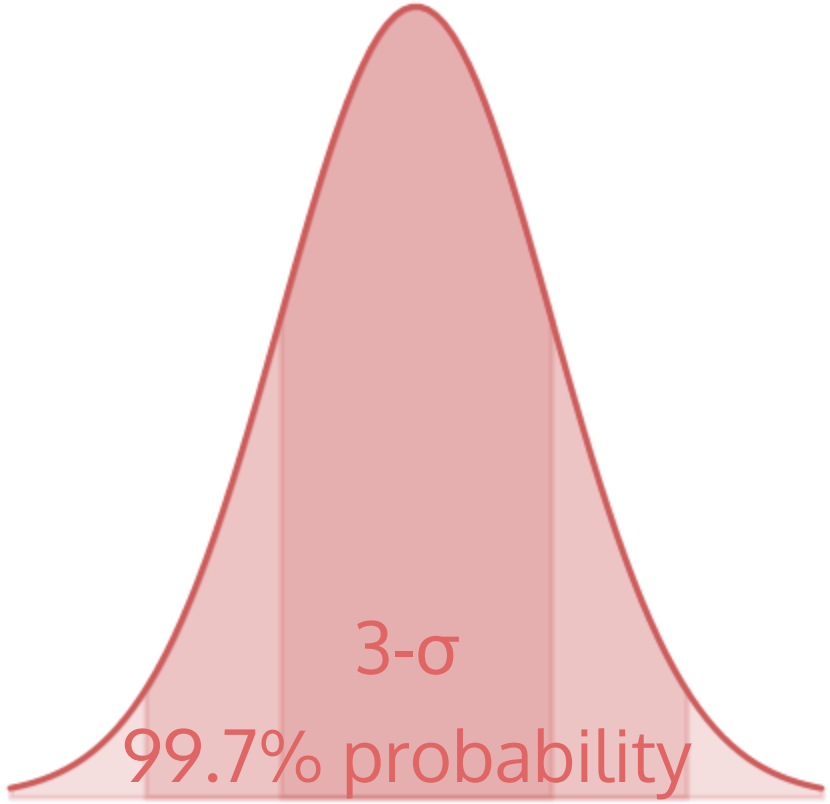
5 points out of 100 can be outside of the uncertainties



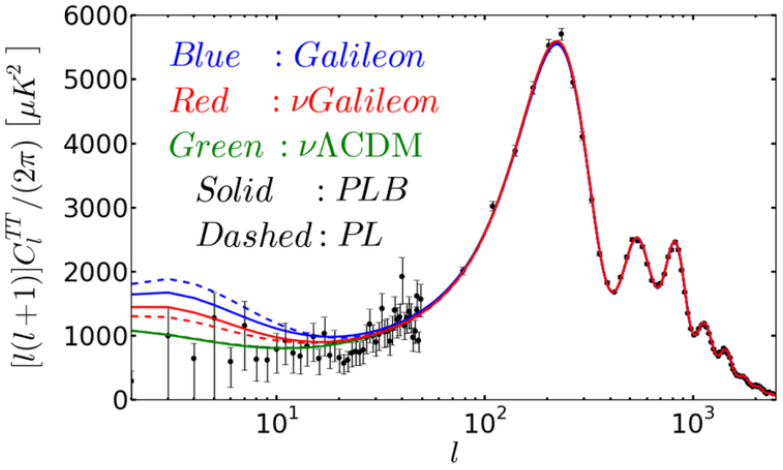


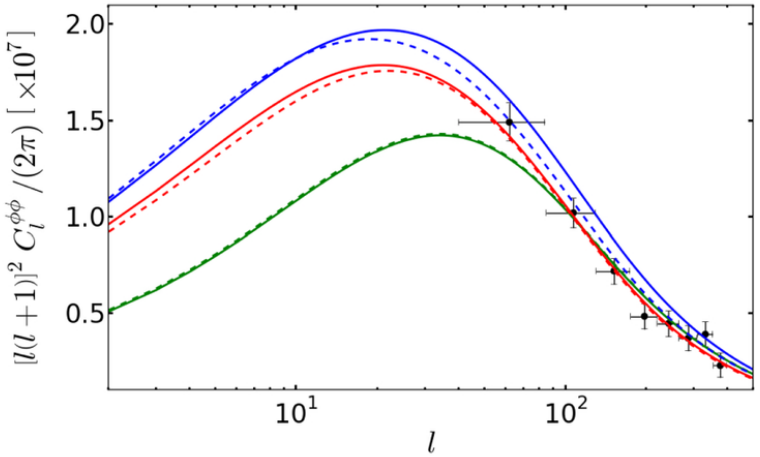


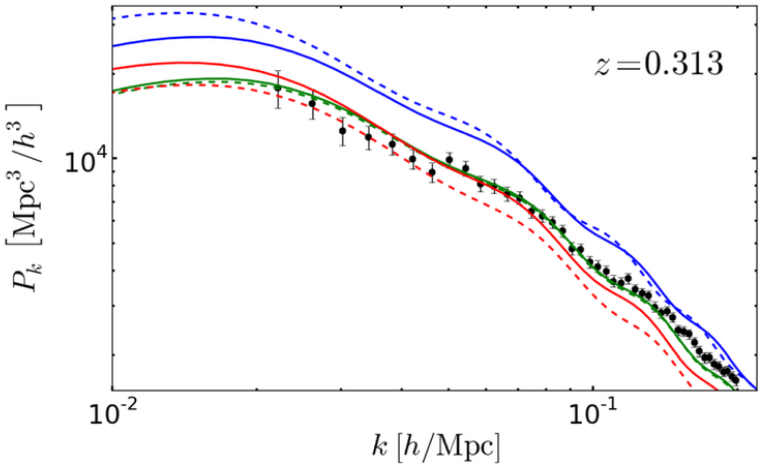
uncertainties: 1-σ



3 points out of 1000 can be outside of the uncertainties

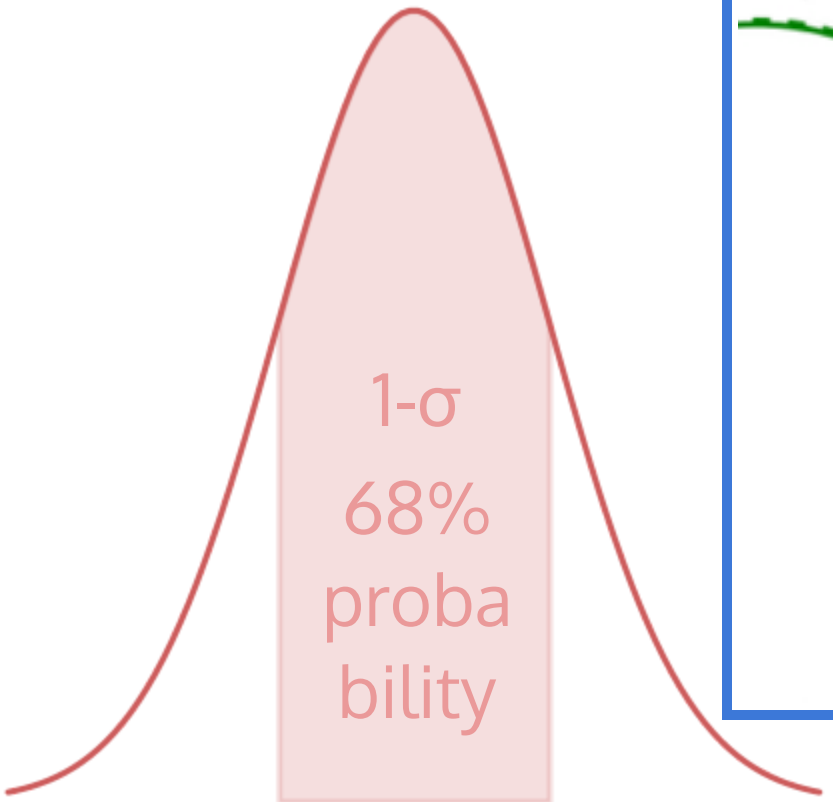




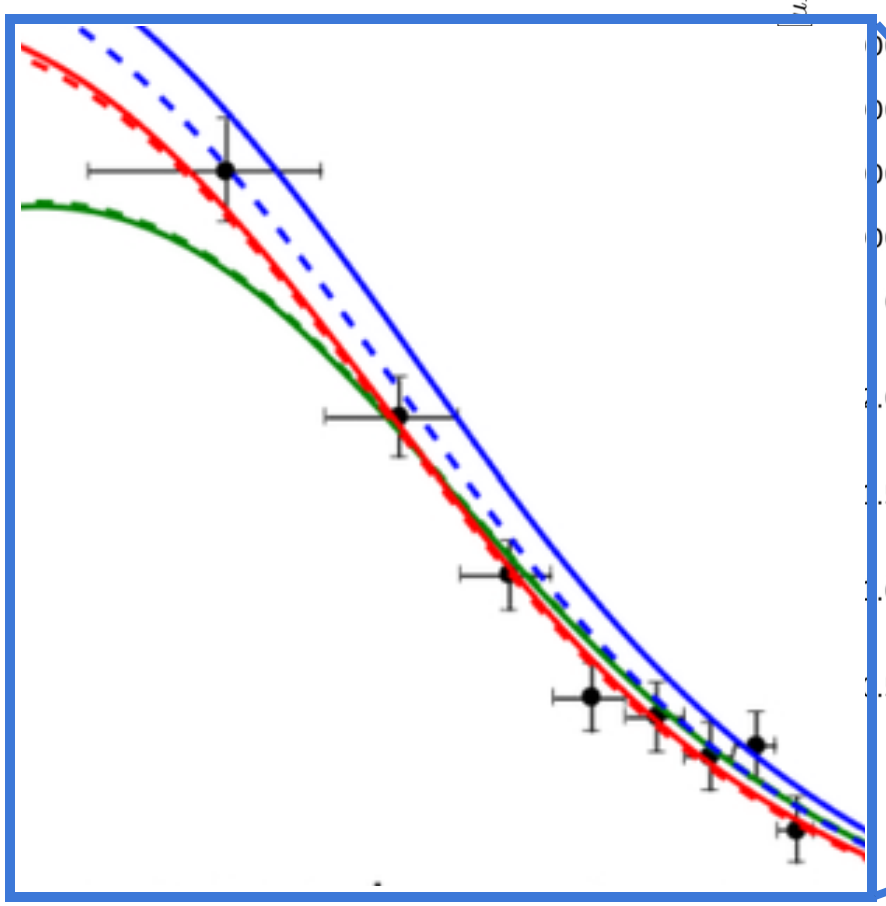


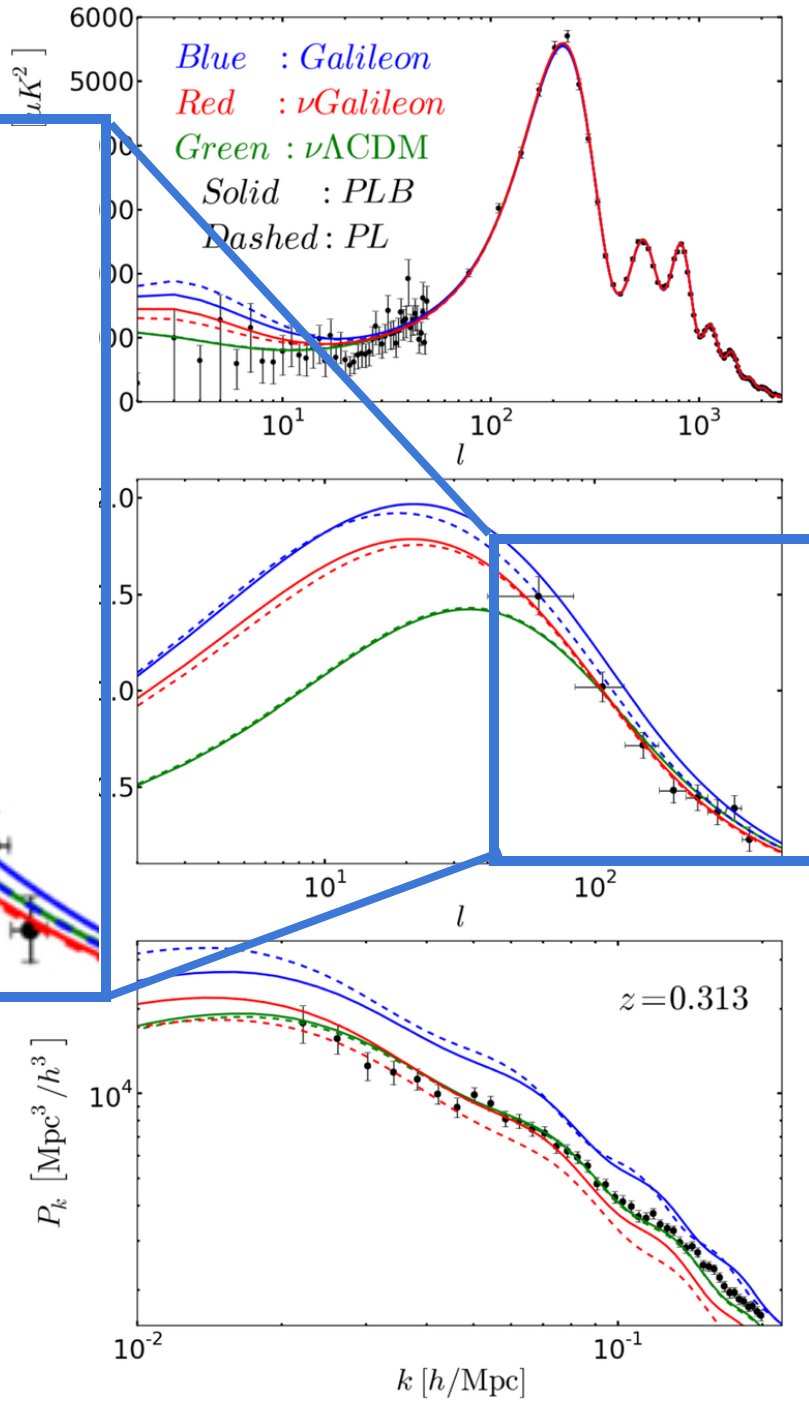
which model can be rejected at 1σ?

uncertainties: 1-σ

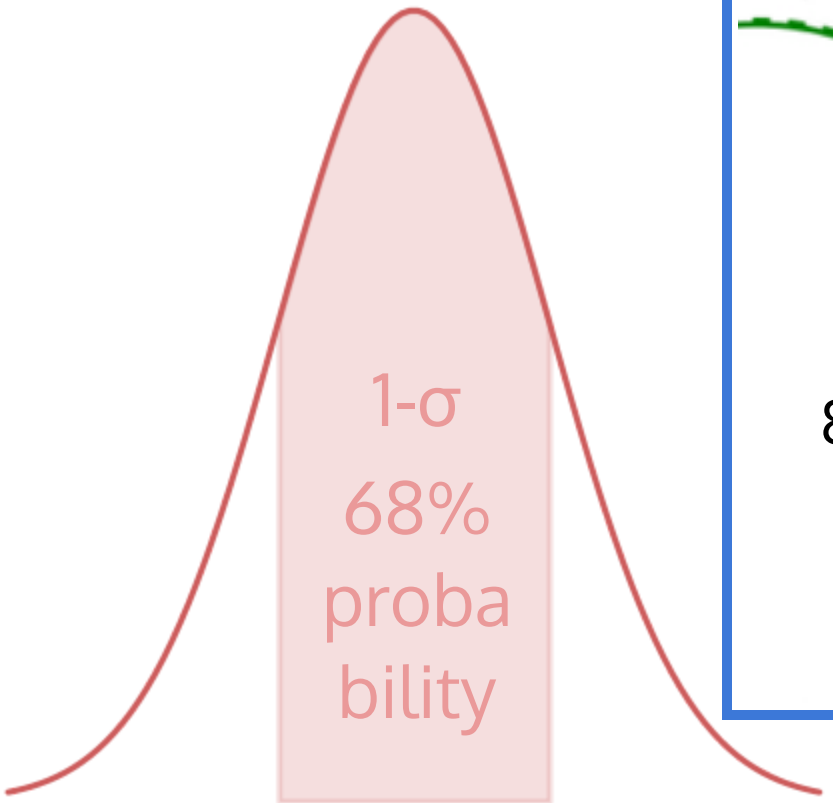


3 points out of 10 can be outside of the uncertainties





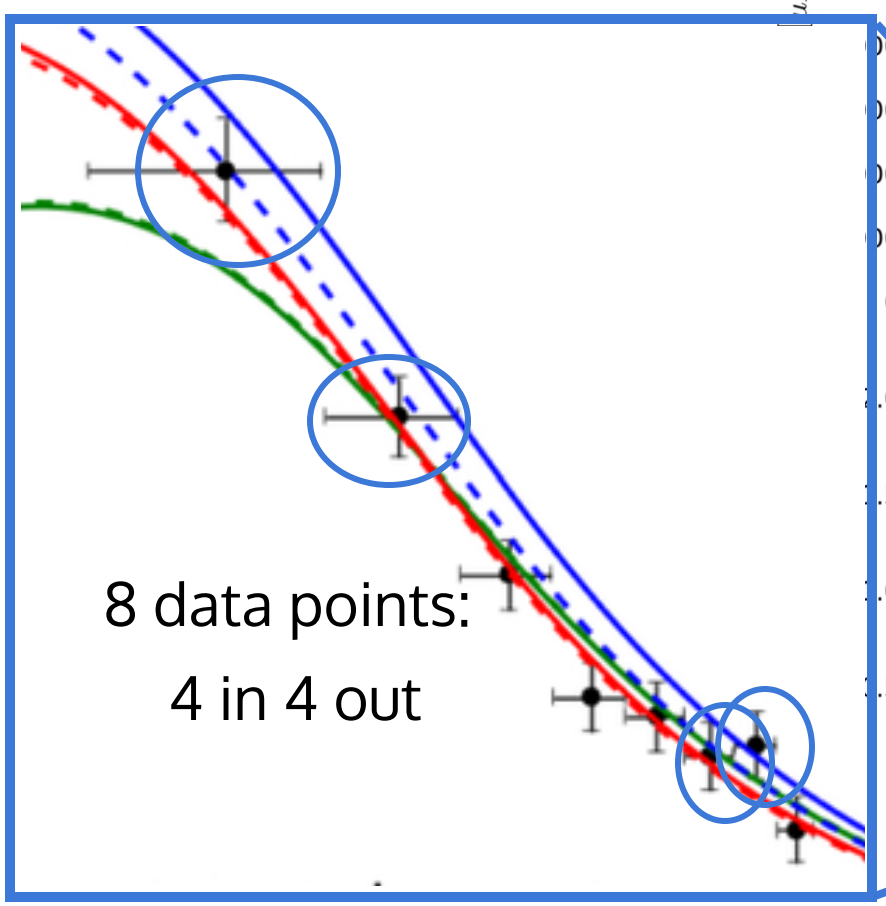
uncertainties: 1-σ

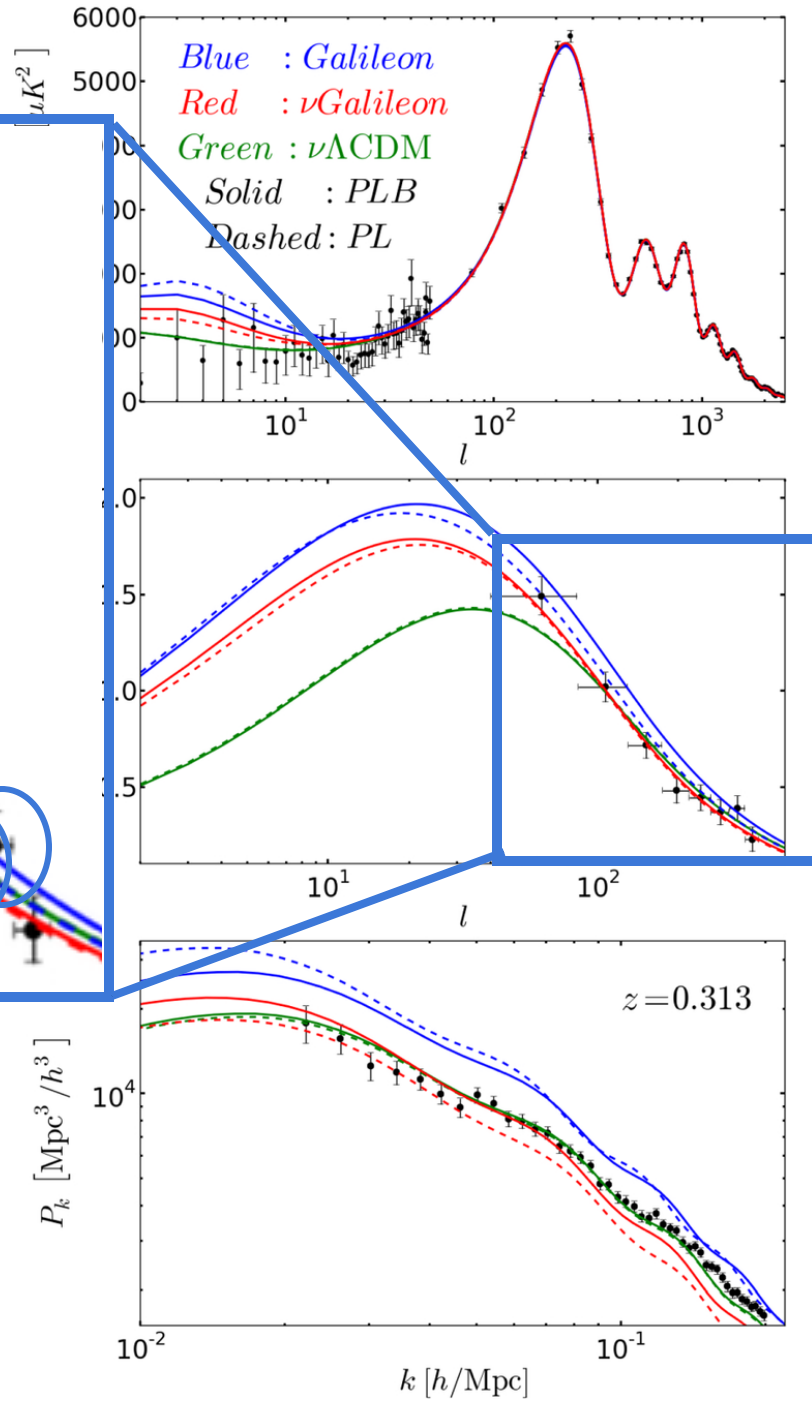


7 points out of 10 should be inside the errorbar

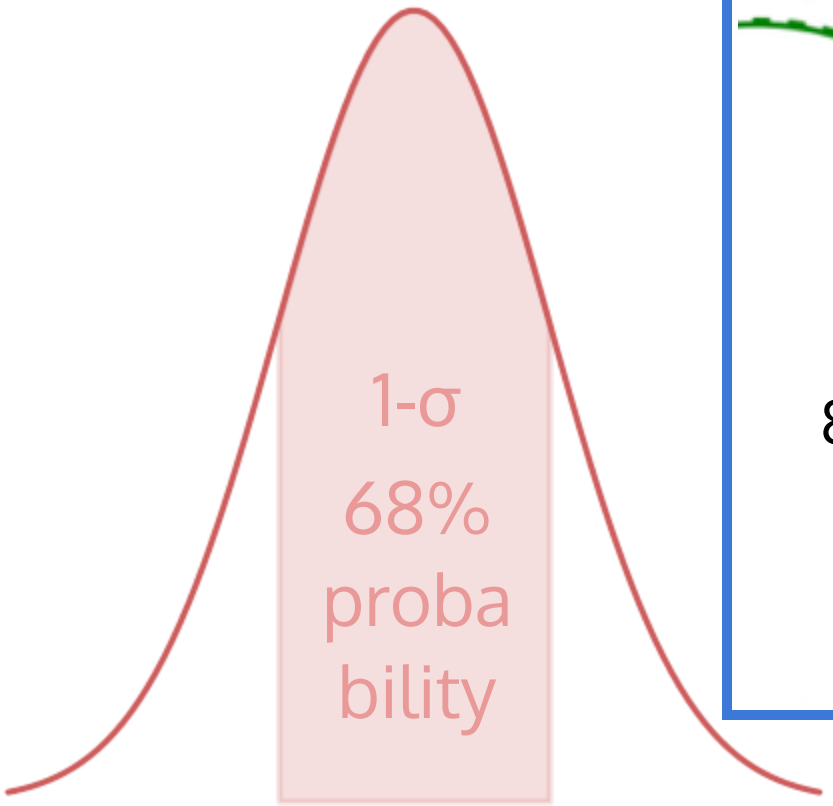
8 data points:

4 in 4 out





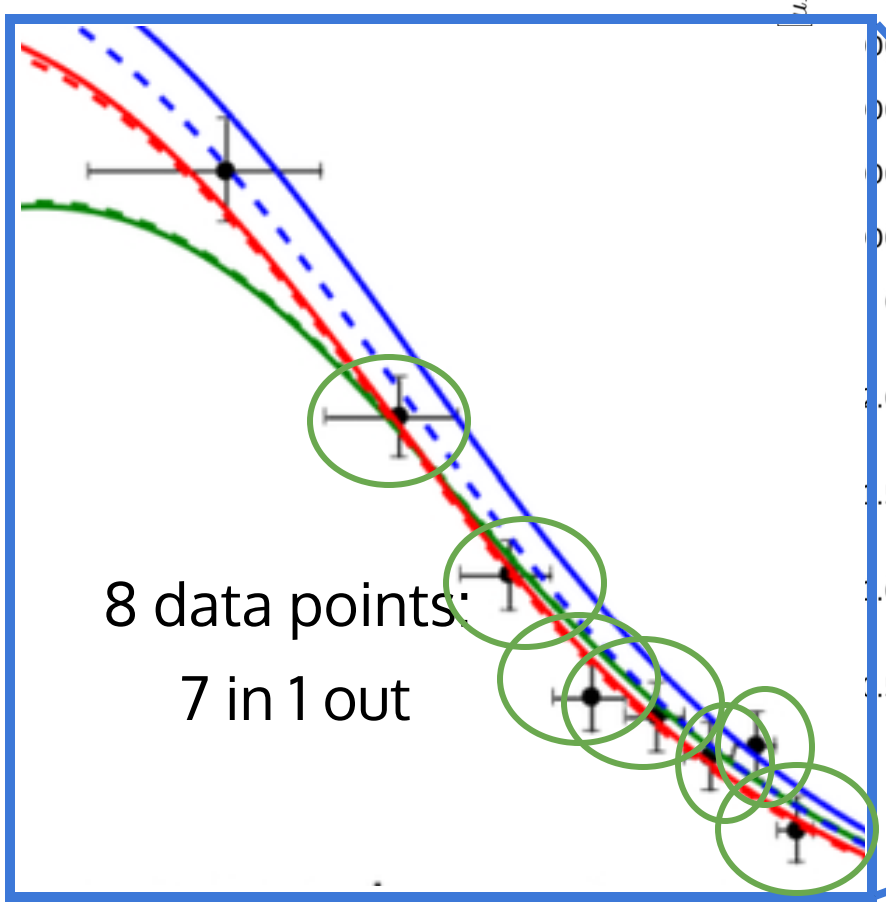
uncertainties: 1-σ

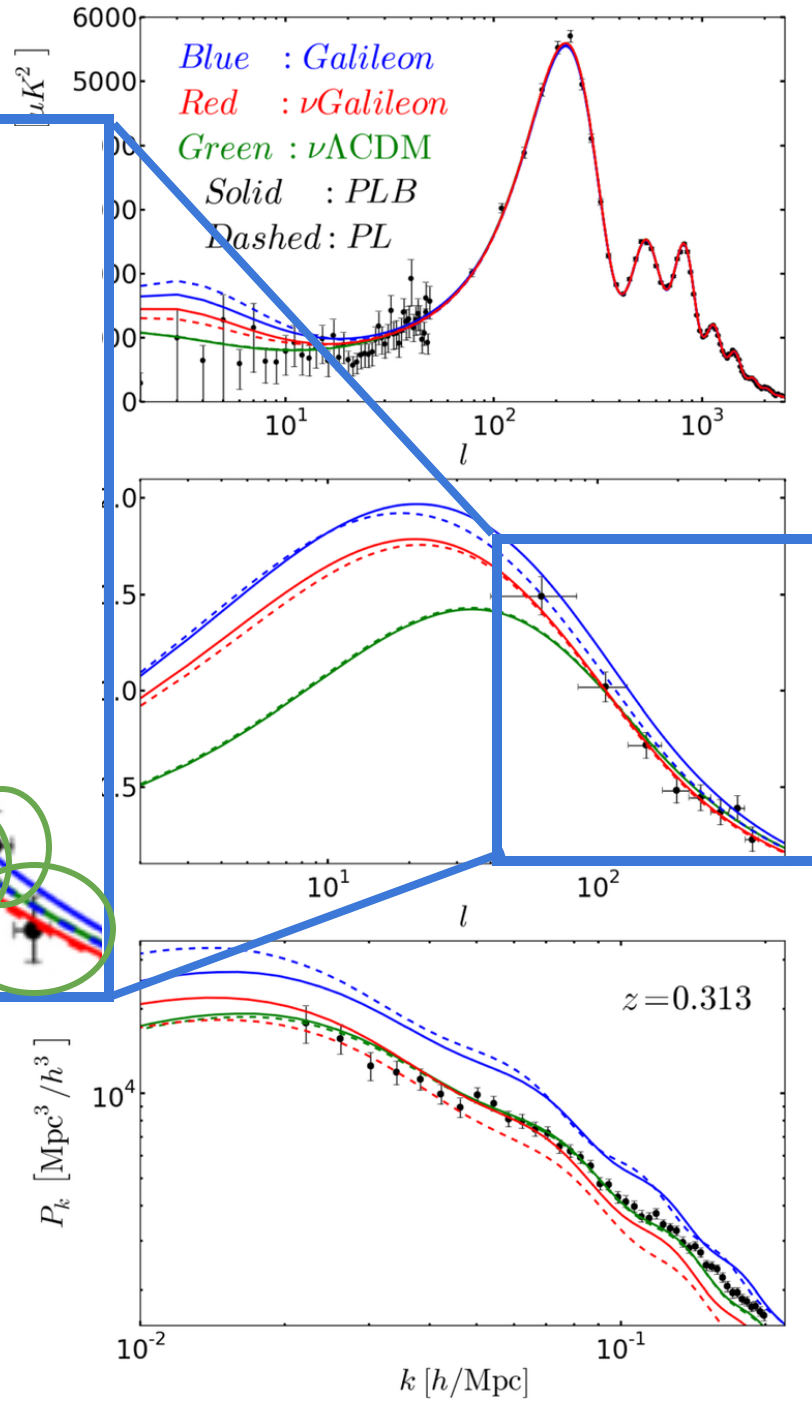


7 points out of 10 should be inside the errorbar

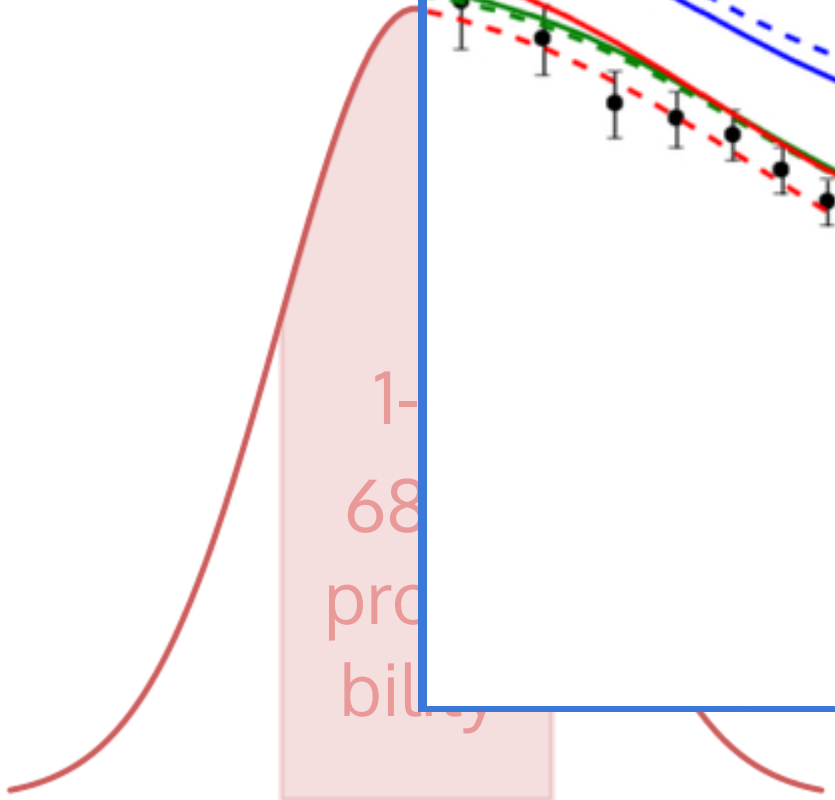
8 data points:

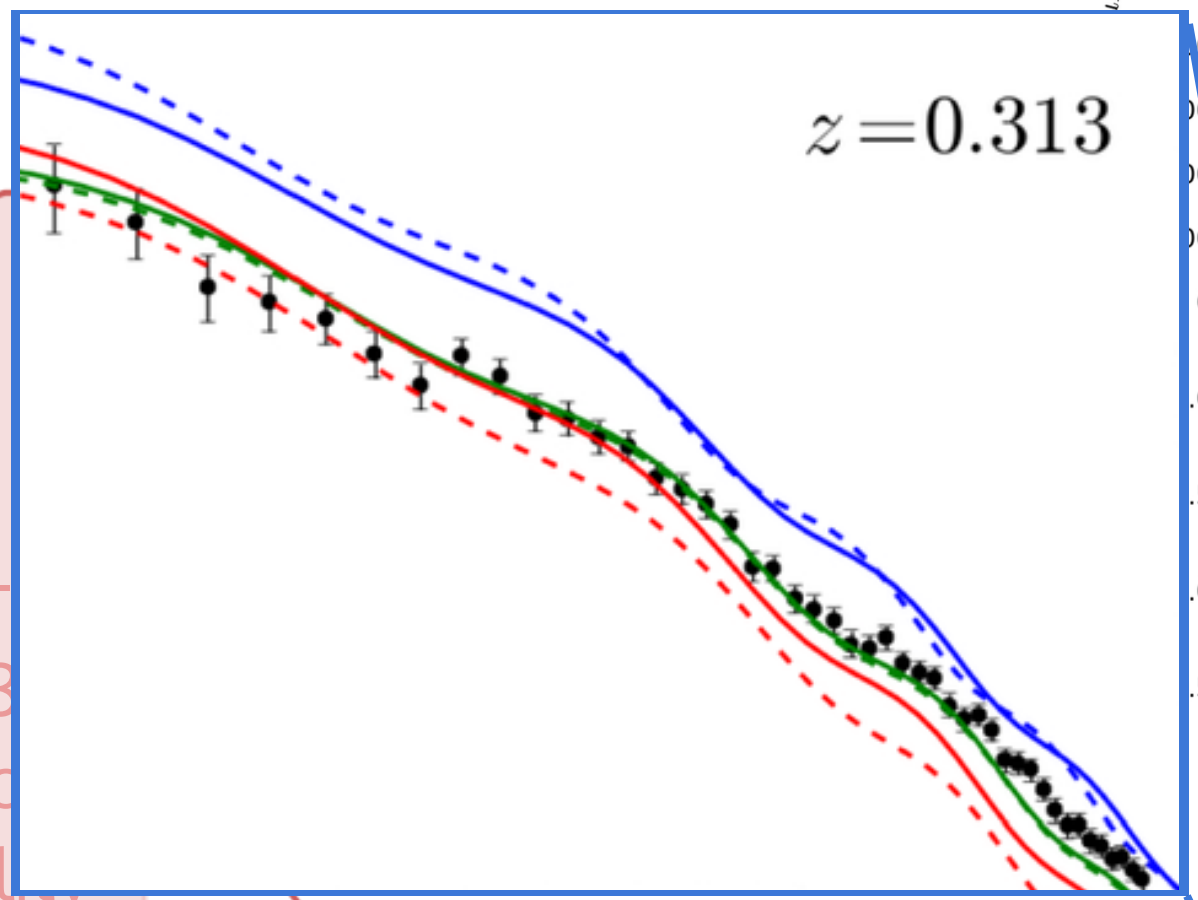
7 in 1 out

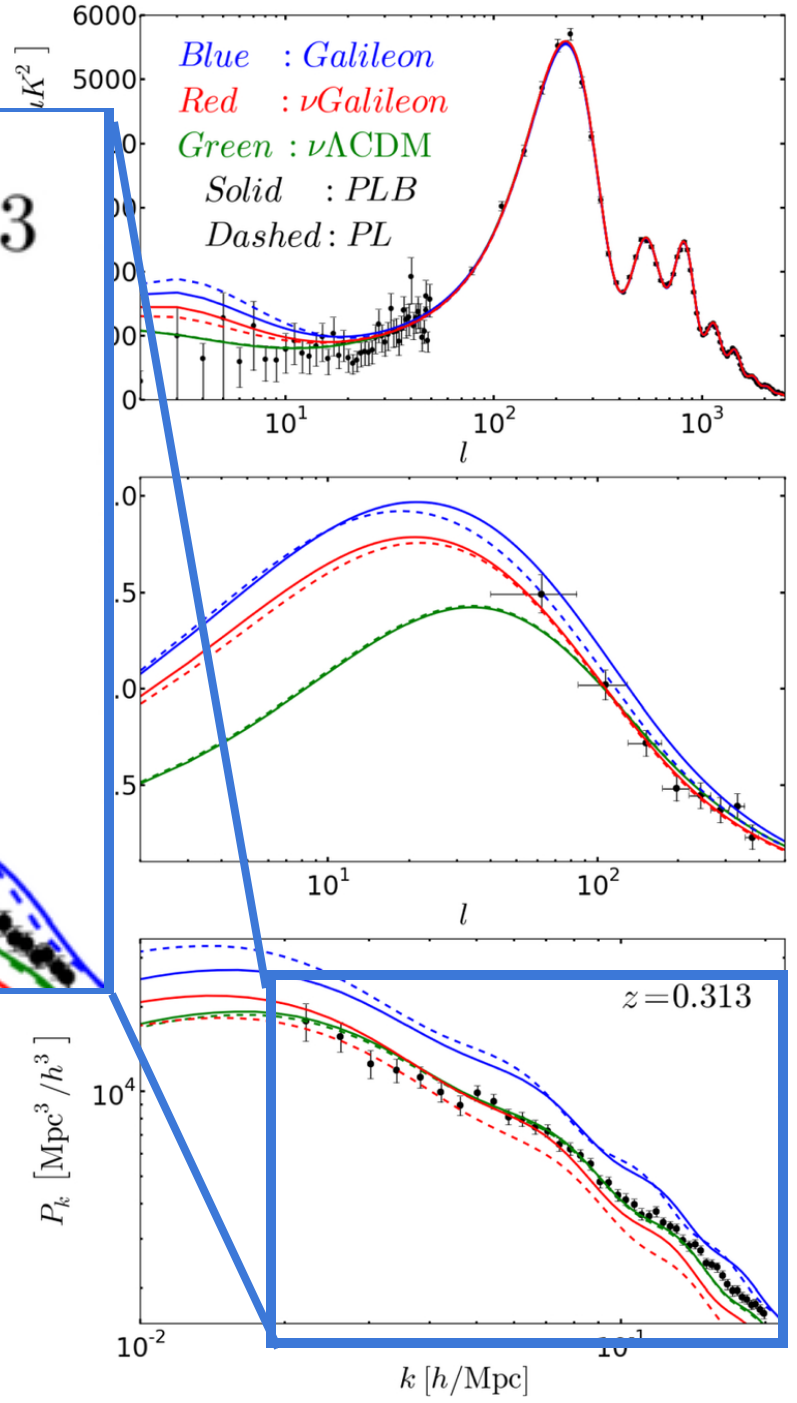




uncertainties: 1-σ

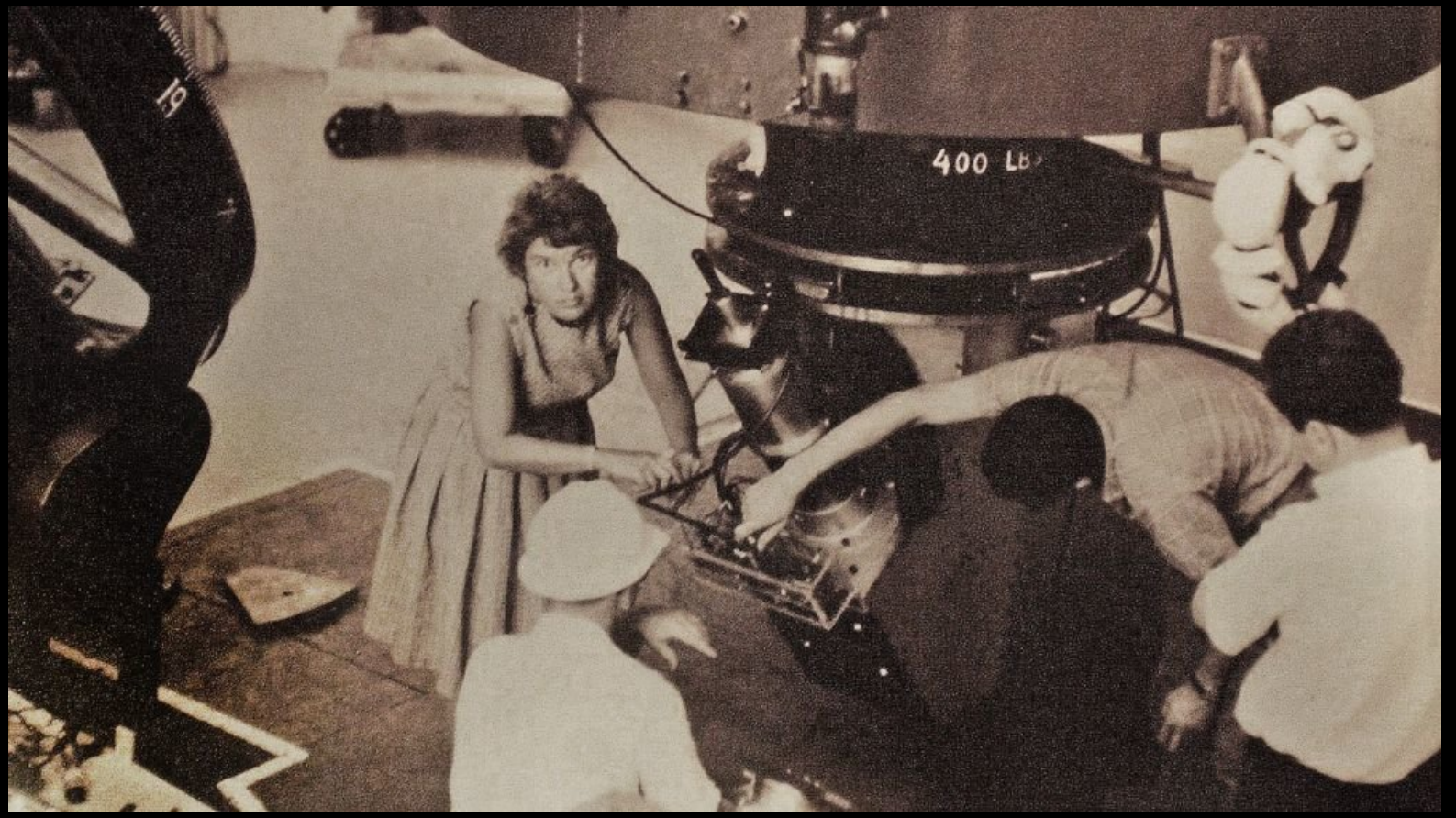






7 points out of 10 should be inside the errorbar

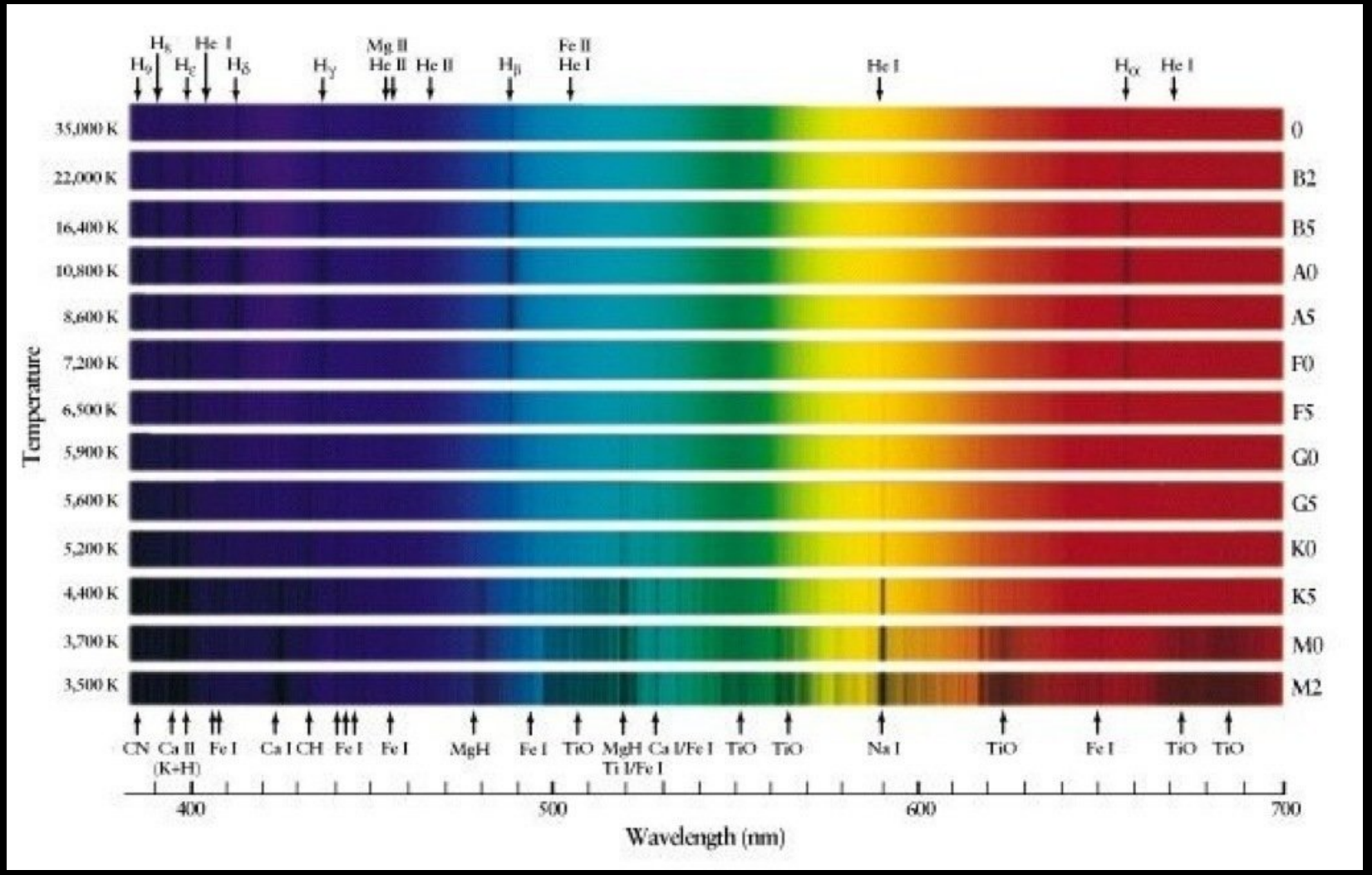
4 Dark Matter



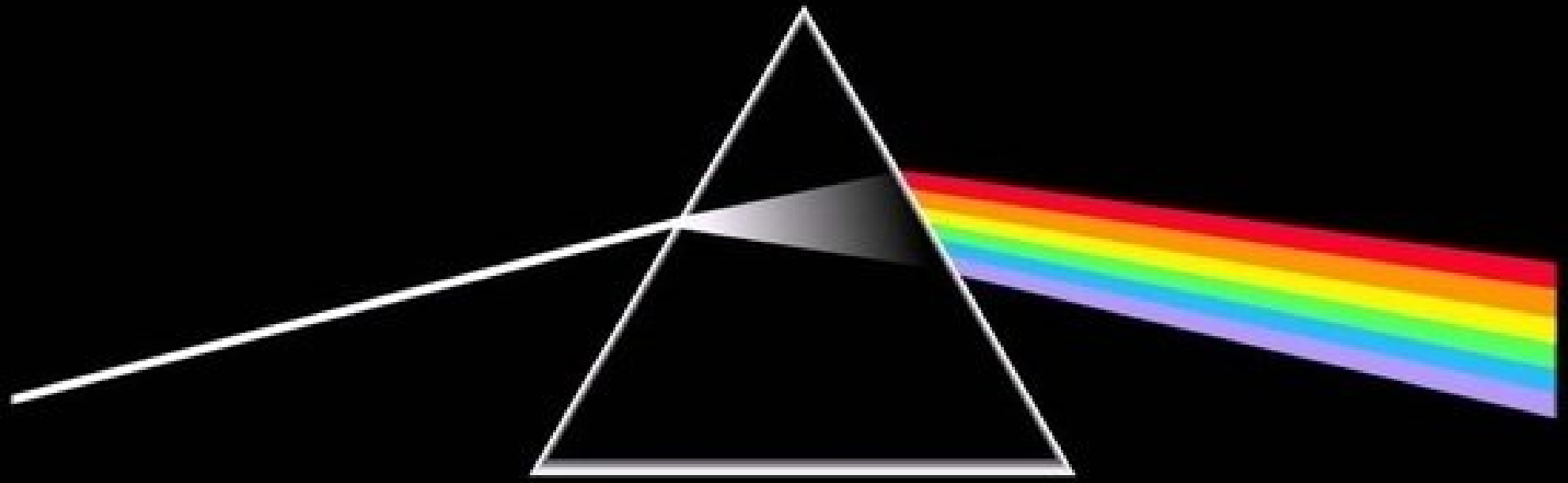
Vera Rubin at work at the Lowell Observatory in Flagstaff, AZ in 1965.

(Image: © Washington Times/Zuma)

Spectra

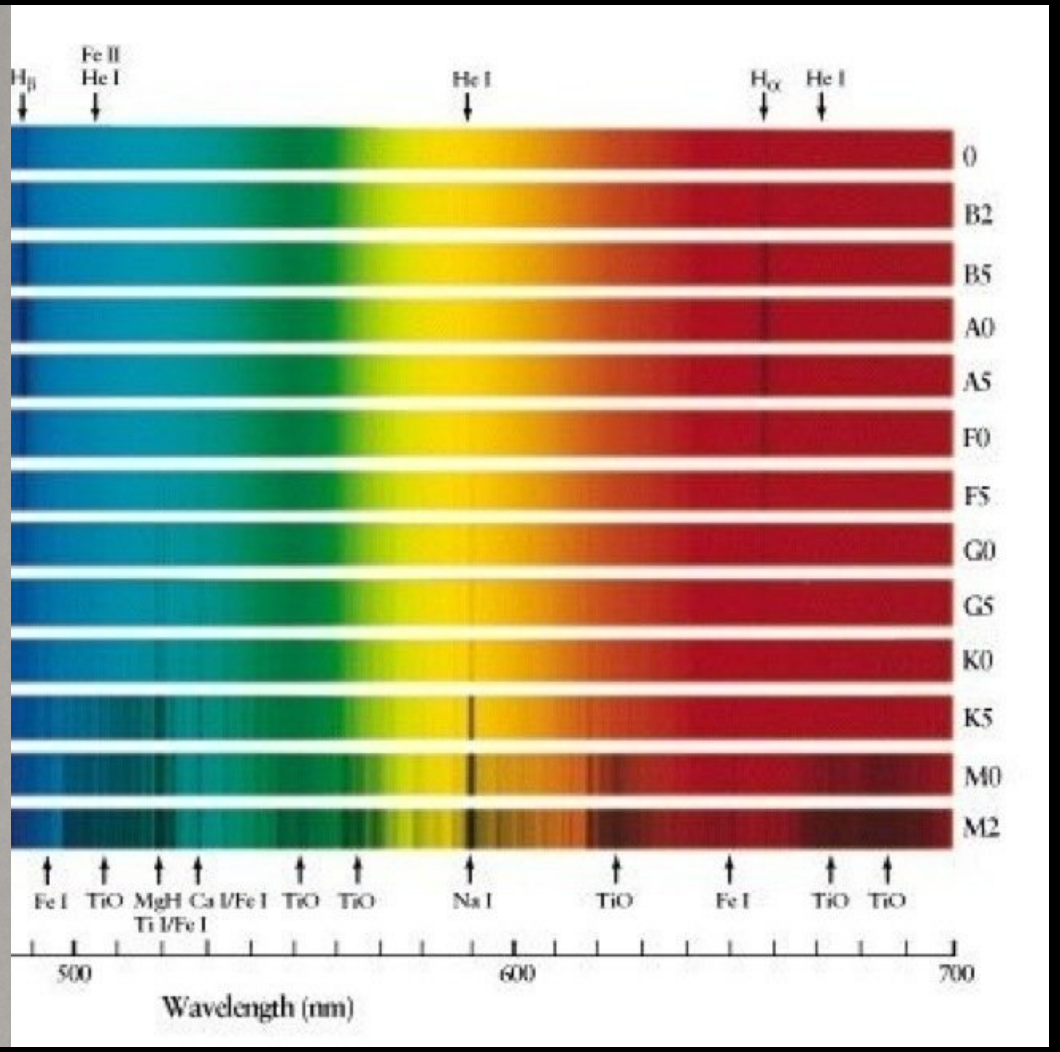


Stellar classification based on the strength of H lines



Spectra

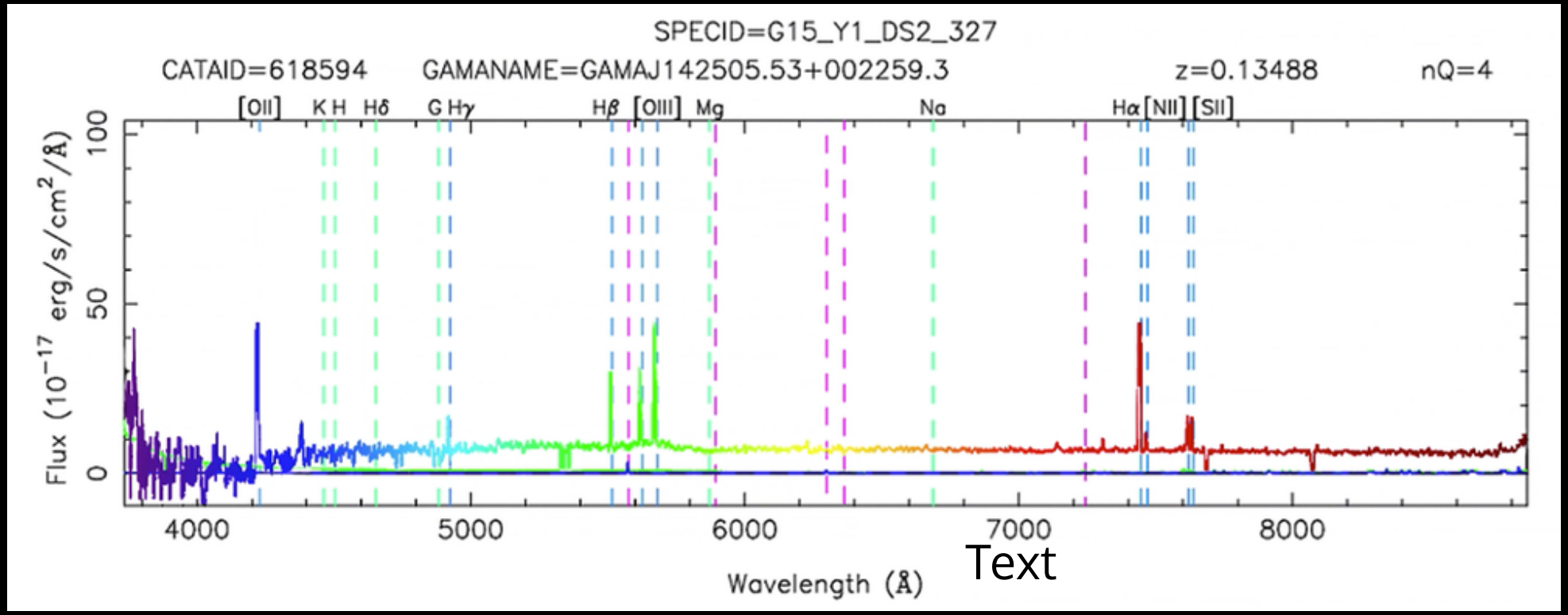
Pink Floyd's Album **Dark side of the moon**



Spectra

The Harvard computers

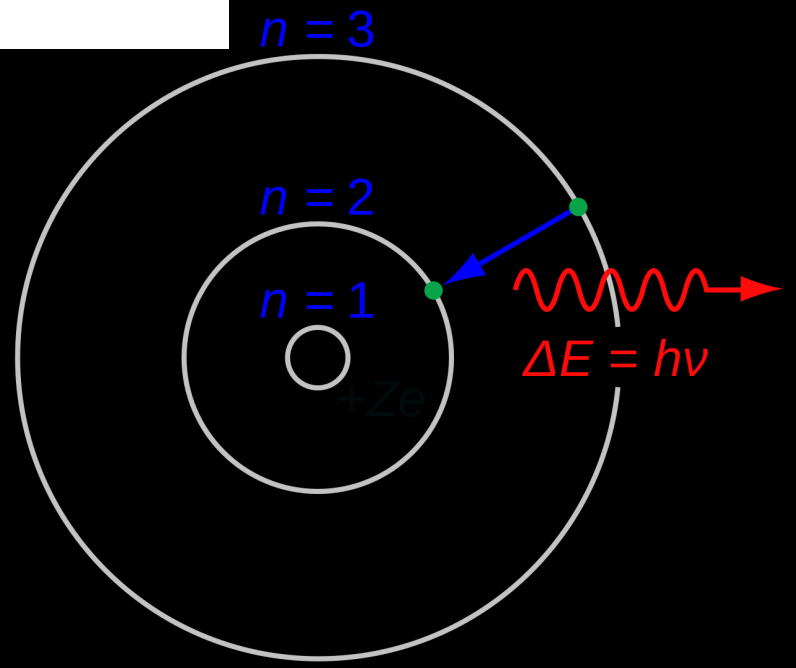
Williamina Fleming, Annie Jump Cannon, Antonia Maury, Henrietta Swan Leavitt and Cecilia Payne-Gaposchkin.



Text

The wavelength of the lines is fixed by atomic physics:

Spectra



https://www.youtube.com/embed/a3RfULw7aAY?enablejsapi=1

Spectra

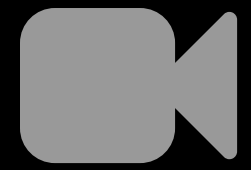
Doppler effect

https://www.youtube.com/embed/-BuwWtMygxU?enablejsapi=1

Spectra

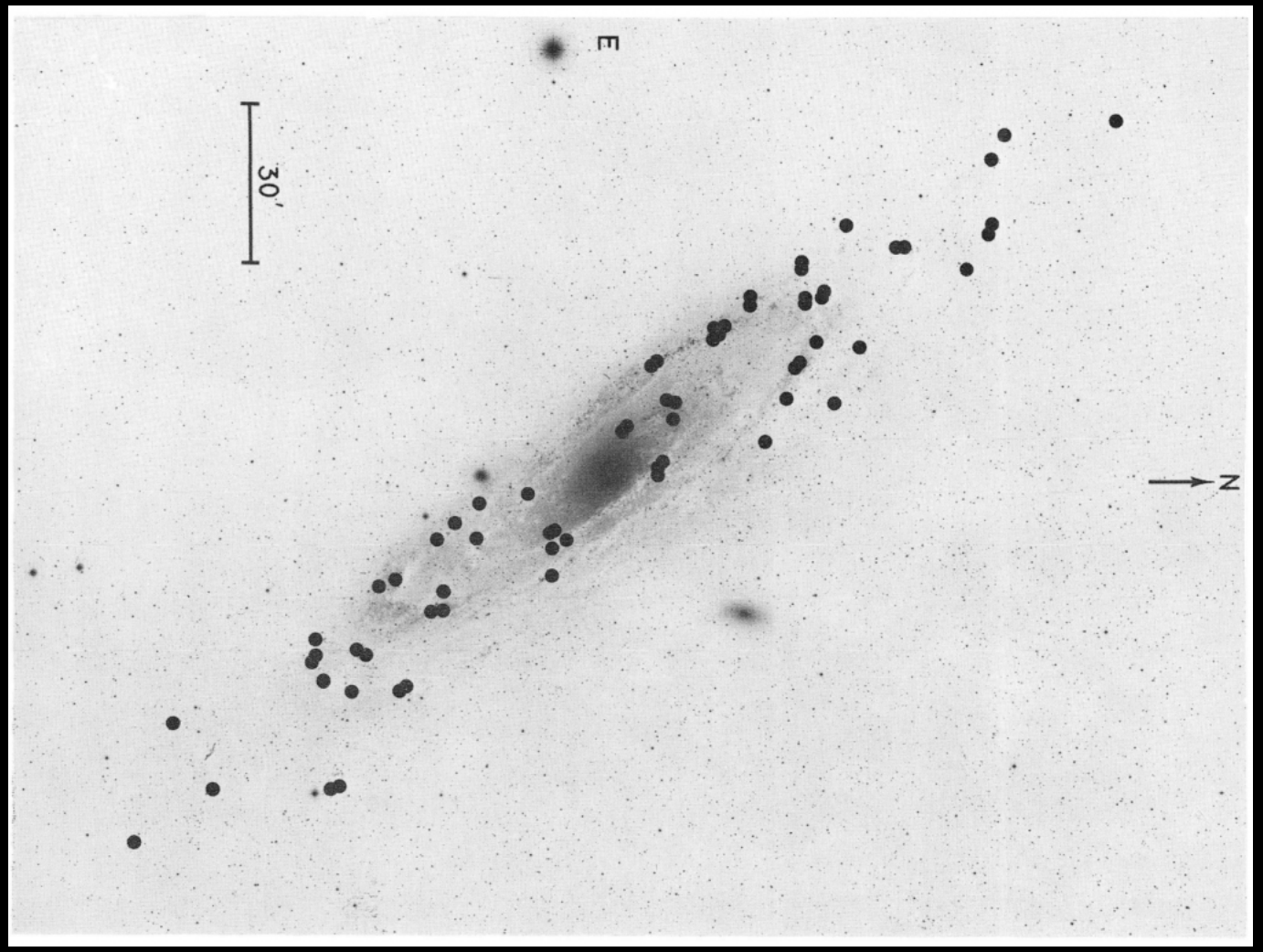
Doppler effect

Spectra



Doppler effect

Galaxies rotational velocity

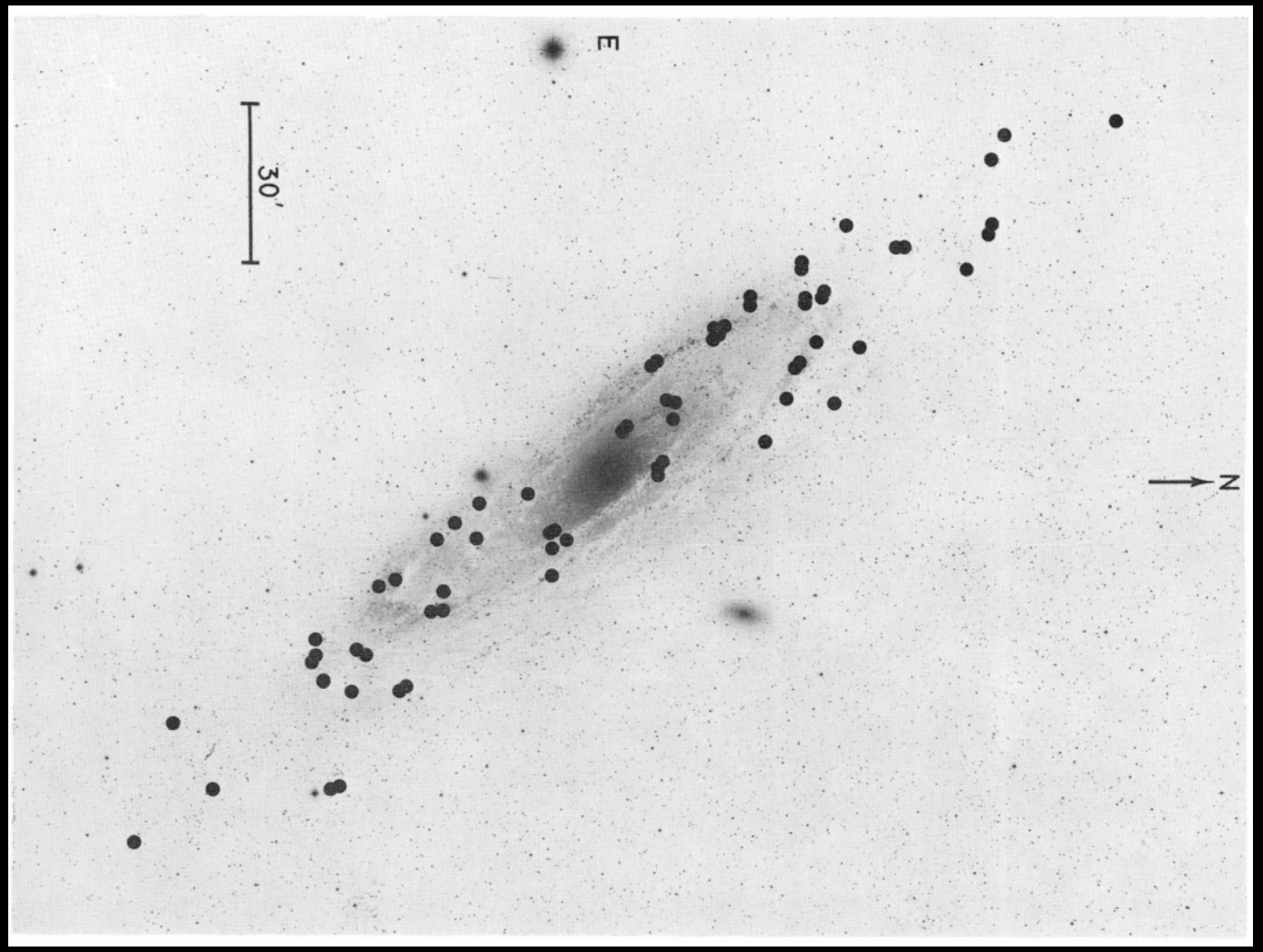


Rotation of the Andromeda Nebula from a Spectroscopic Survey of Emission Regions

Vera C. Rubin and W. Kent Ford, Jr. **1970**

Newton's gravity

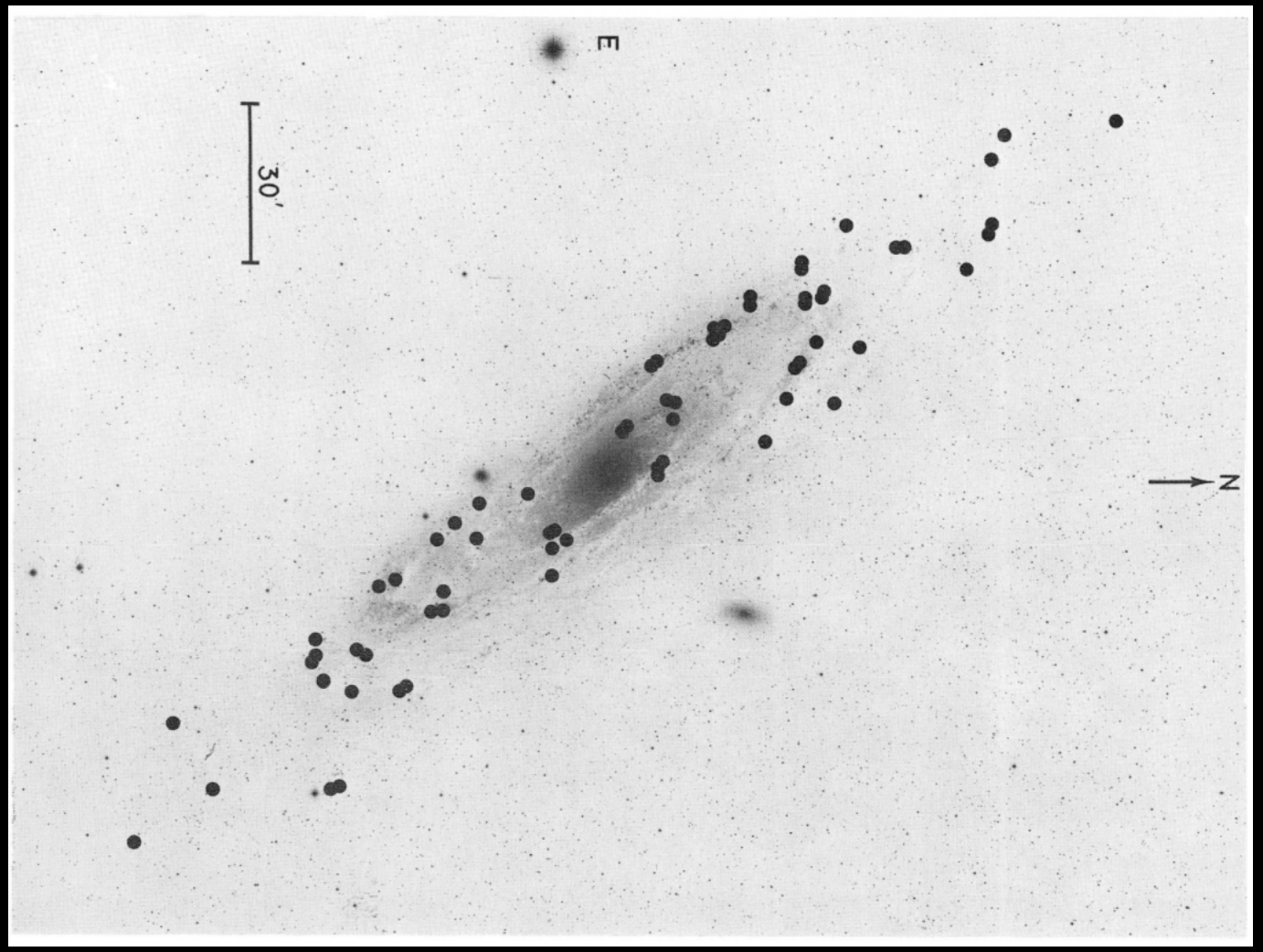
$$\frac{GMm}{r^2} = \frac{m\,v^2}{r}$$



Rotation of the Andromeda Nebula from a Spectroscopic Survey of Emission Regions
Vera C. Rubin and W. Kent Ford, Jr. **1970**

Newton's gravity

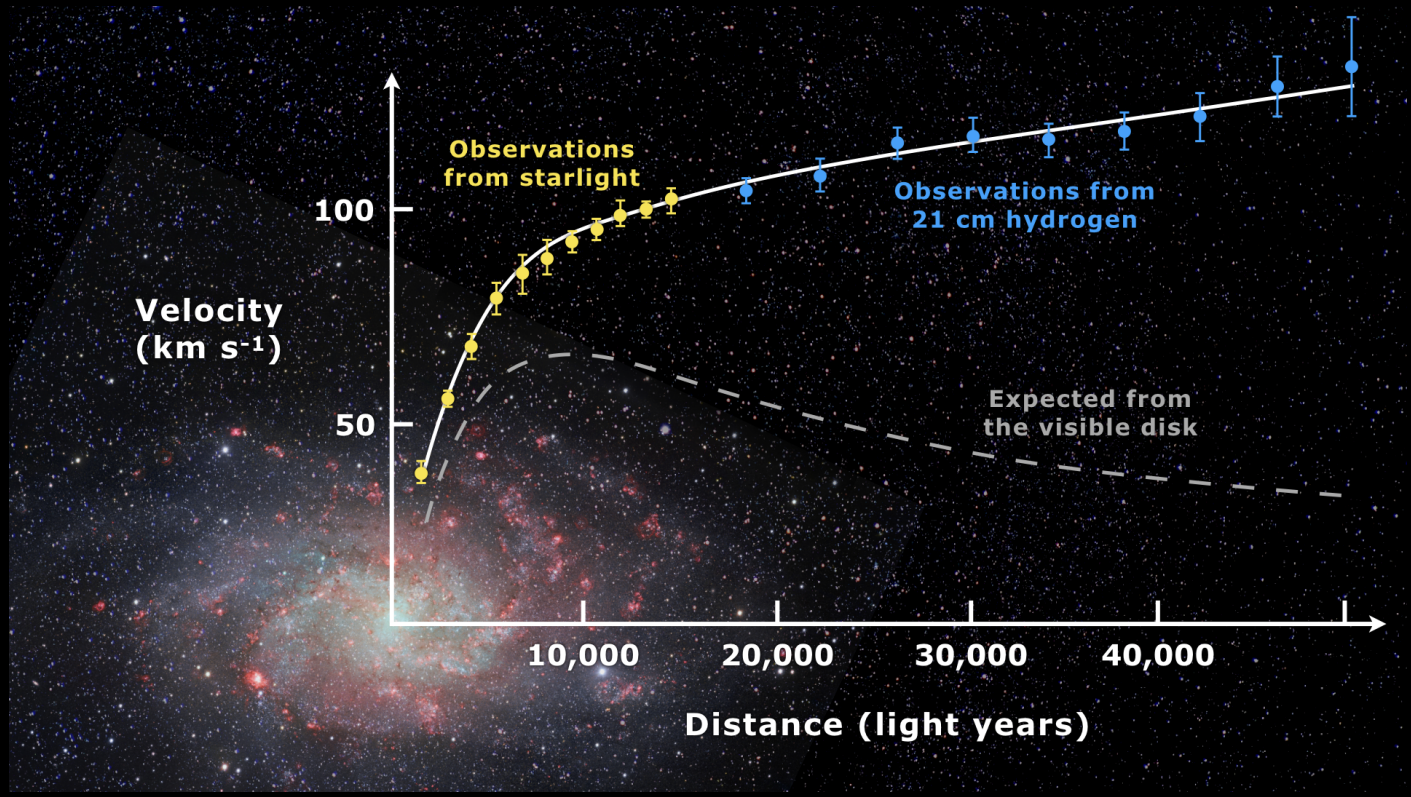
$$v = \frac{\text{const}}{r}$$



Rotation of the Andromeda Nebula from a Spectroscopic Survey of Emission Regions
Vera C. Rubin and W. Kent Ford, Jr. **1970**

Dark Matter

https://en.wikipedia.org/wiki/Galaxy_rotation_curve



$$v \neq \frac{\text{const}}{r}$$

Rotation of the Andromeda Nebula from a Spectroscopic Survey of Emission Regions

Vera C. Rubin and W. Kent Ford, Jr. **1970**

Galaxies rotational velocity

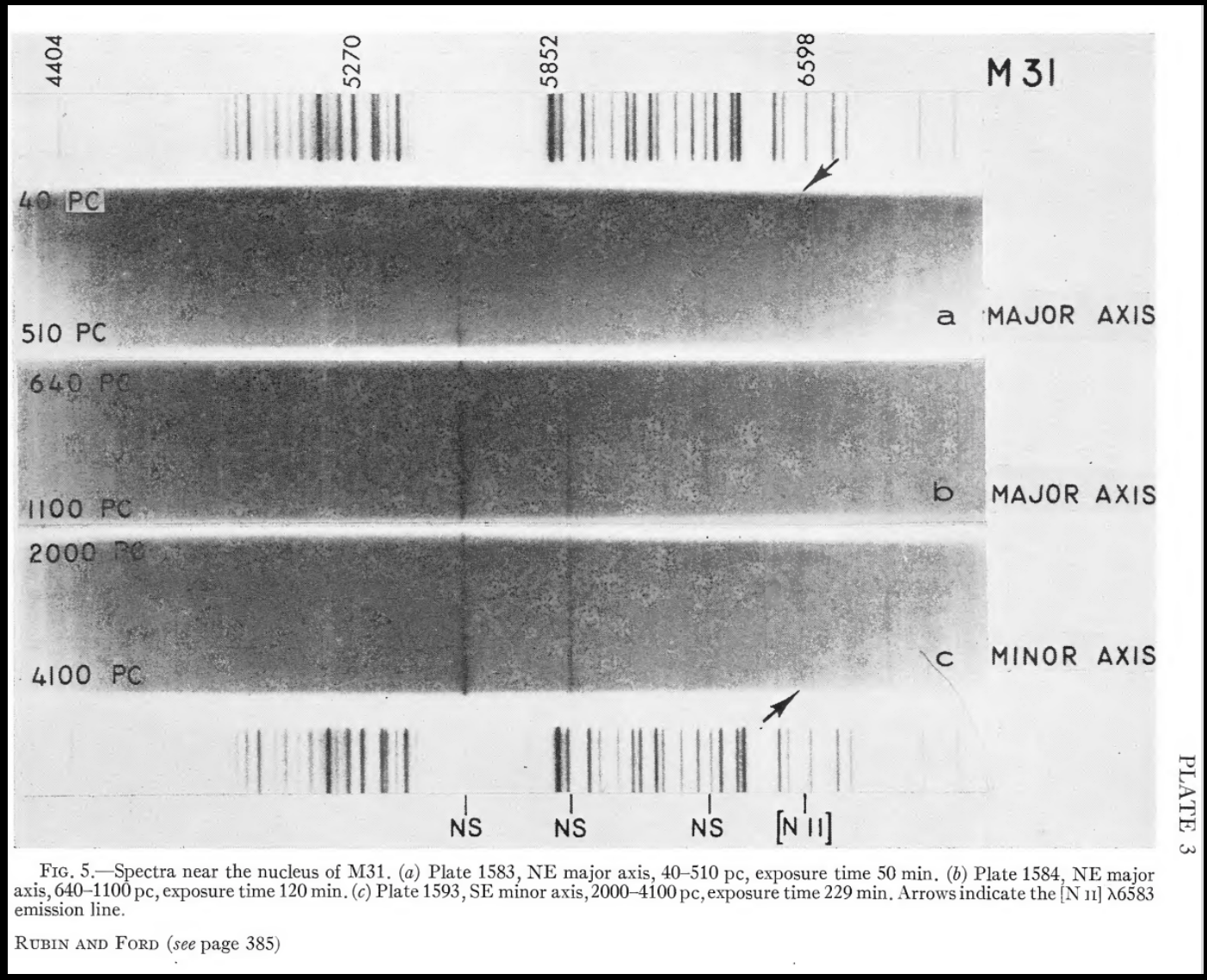


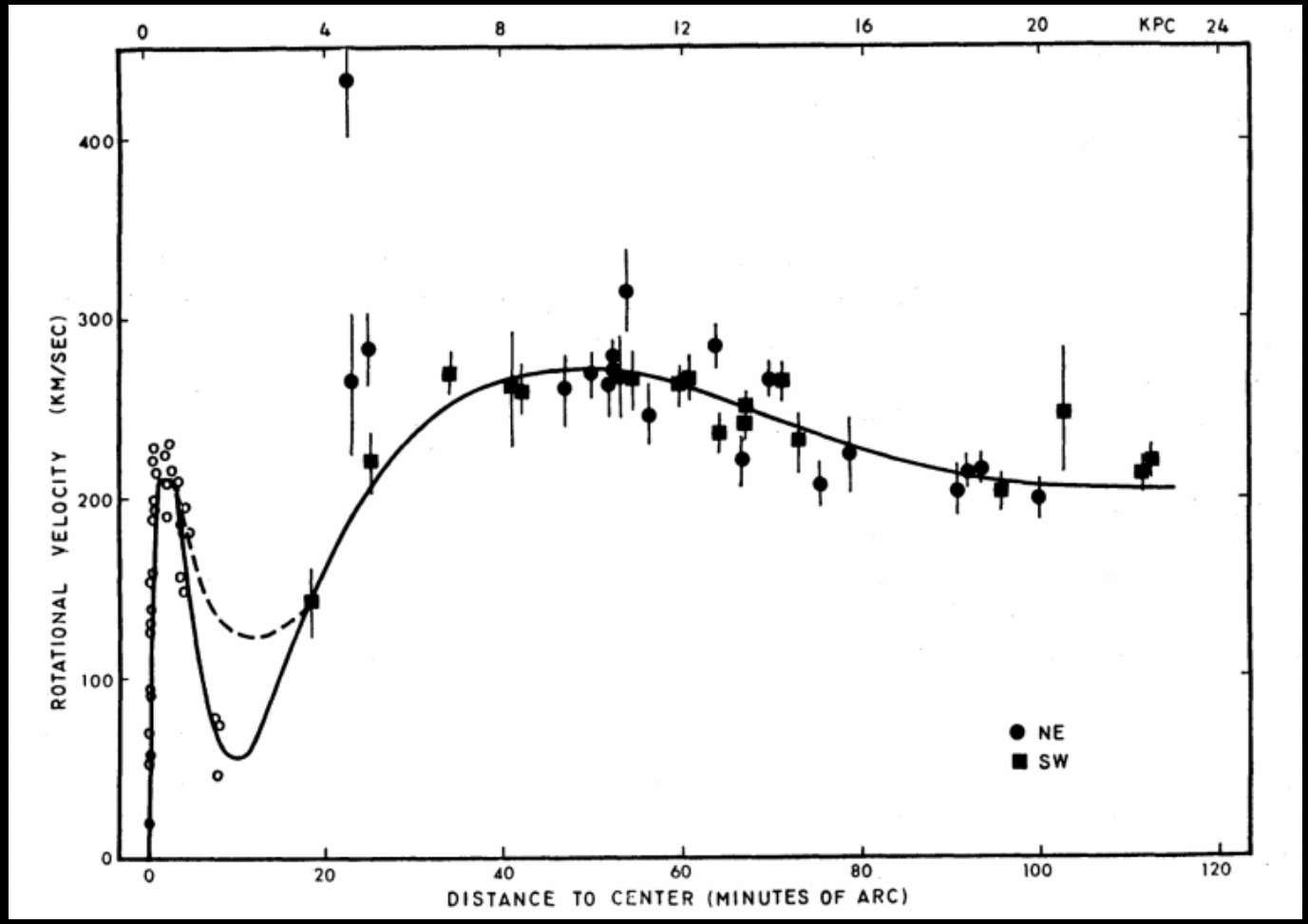
FIG. 5.—Spectra near the nucleus of M31. (*a*) Plate 1583, NE major axis, 40–510 pc, exposure time 50 min. (*b*) Plate 1584, NE major axis, 640–1100 pc, exposure time 120 min. (*c*) Plate 1593, SE minor axis, 2000–4100 pc, exposure time 229 min. Arrows indicate the [N II] λ6583 emission line.

RUBIN AND FORD (*see* page 385)

Rotation of the Andromeda Nebula from a Spectroscopic Survey of Emission Regions

Vera C. Rubin and W. Kent Ford, Jr. **1970**

Galaxies rotational velocity



Rotation of the Andromeda Nebula from a Spectroscopic Survey of Emission Regions

Vera C. Rubin and W. Kent Ford, Jr. **1970**

1. See instructions in HW4 for galaxy rotational velocity notebook

2. 617 and extra credit for 417

Problem 2 in https://arxiv.org/pdf/1710.06068.pdf

homework

https://www.slac.stanford.edu/econf/C030908/papers/TUAT004.pdf

PHYSTAT2003, SLAC, Stanford, California, September 8-11, 2003

Definition and Treatment of Systematic Uncertainties in High Energy Physics and Astrophysics

Pekka K. Sinervo
Department of Physics, University of Toronto, Toronto, ON M5S 1A7, CANADA

Systematic uncertainties in high energy physics and astrophysics are often significant contributions to the overall uncertainty in a measurement, in many cases being comparable to the statistical uncertainties. However, consistent definition and practice is elusive, as there are few formal definitions and there exists significant ambiguity in what is defined as a systematic and statistical uncertainty in a given analysis. I will describe current practice, and recommend a definition and classification of systematic uncertainties that allows one to treat these sources of uncertainty in a consistent and robust fashion. Classical and Bayesian approaches will be contrasted.

1. INTRODUCTION TO SYSTEMATIC UNCERTAINTIES

Most measurements of physical quantities in high energy physics and astrophysics involve both a statistical uncertainty and an additional "systematic" uncertainty. Systematic uncertainties play a key role in

include uncertainties that arise from the calibration of the measurement device, the probability of detection of a given type of interaction (often called the "acceptance" of the detector), and parameters of the model used to make inferences that themselves are not precisely known. The definition of such uncertainties is often ad hoc in a given measurement, and there are

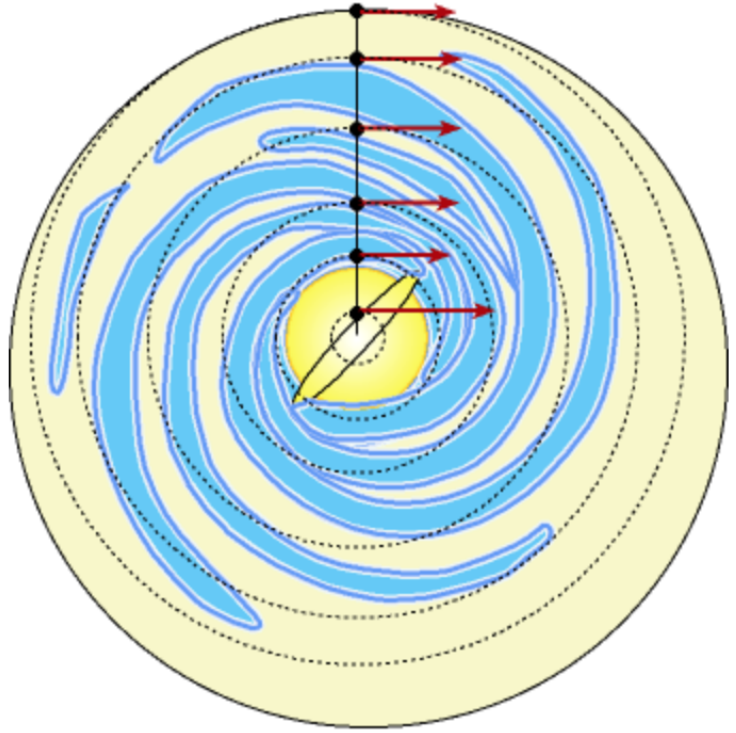
https://www.youtube.com/watch?v=1uPyq63aRvg

https://www.youtube.com/embed/1uPyq63aRvg?enablejsapi=1

references

Deriving the Galactic Mass from the Rotation Curve

http://www.astronomynotes.com/ismnotes/s7.htm



References

Rotation of the Andromeda Nebula from a Spectroscopic Survey of Emission Regions

Vera C. Rubin and W. Kent Ford, Jr. 1970

references

4 MonteCarlo methods

MonteCarlo method

results are computed based on repeated random sampling and statistical analysis

- History of Monte Carlo Methods
- Application of MC to probabilistic inference
- A simple MC simulation
- Rejection & Importance Sampling

- Markovian Processes and Markov chains
- Bayes theorem and the posterior distribution
- Metropolis-Hastings (and Gibbs sampling) MCMC
- Affine Invariant MCMC
- convergence criteria

41 MC history

history of MC

"What are the chances that a Canfield solitaire laid out with 52 cards will come out successfully?

Stanislav Ulam

Canfield Solitaire

The number of different games is 52! = 52x51x52...x3x2x1 ~ 8x1067

http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-88-9068

"What are the chances that a Canfield solitaire laid out with 52 cards will come out successfully?

After spending a lot of time trying to estimate them by pure combinatorial calculations, I wondered whether **a more practical method than abstract thinking** might not be to **lay it out say one hundred times and simply observe and count the number of successful play**"

history of MC

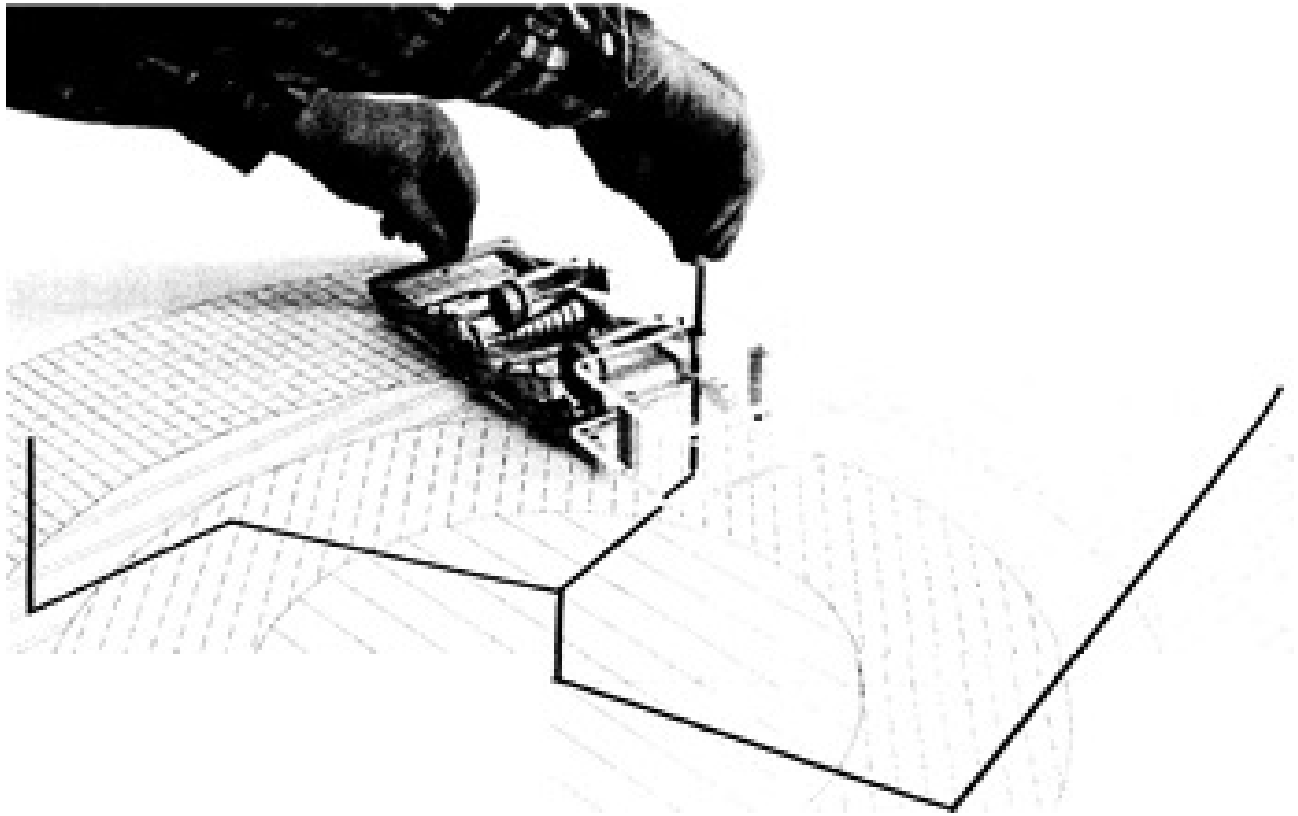
Canfield Solitaire

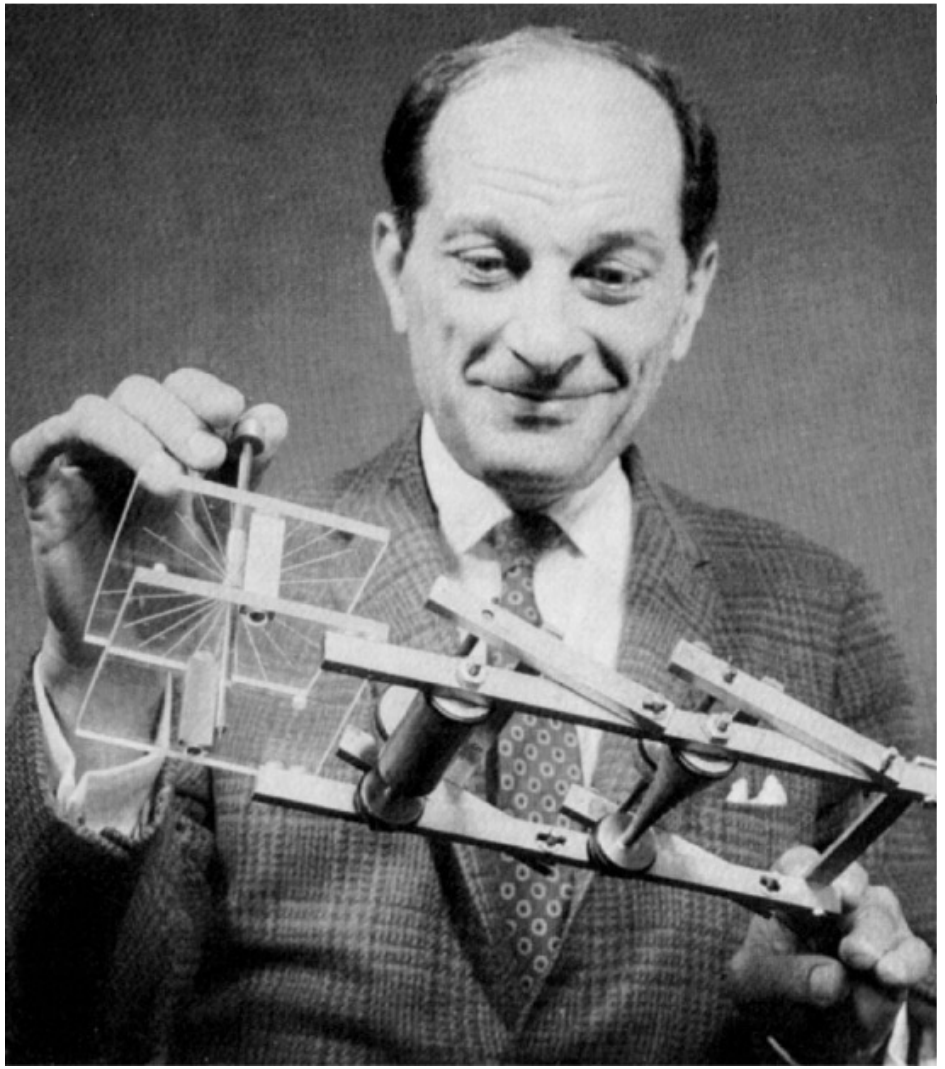
The number of different games is 52! = 52x51x52...x3x2x1 ~ 8x1067

http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-88-9068

The Fermiac or Monte Carlo trolley

Enrico Fermi looked really smart with his predictions...



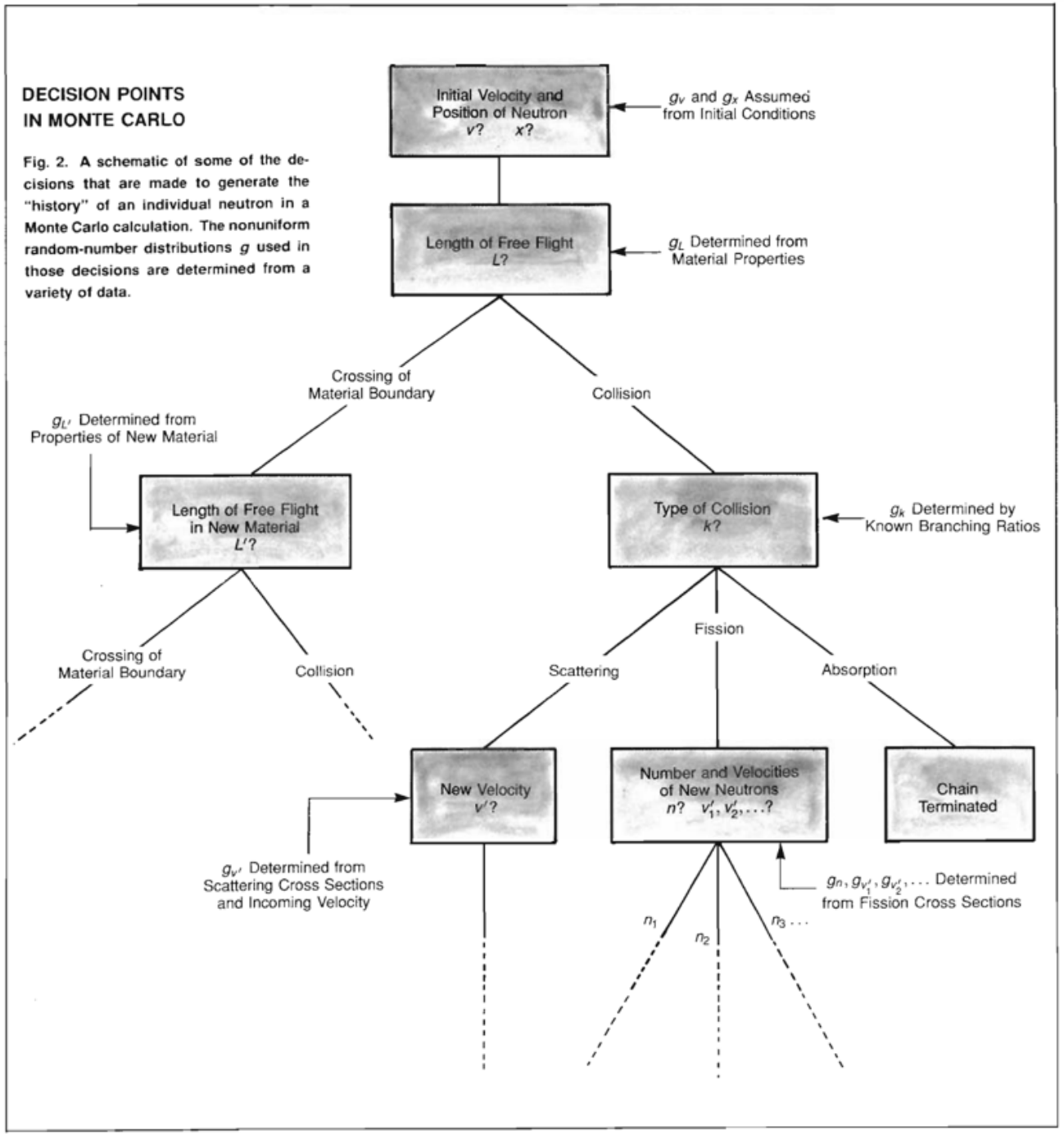


history of MC

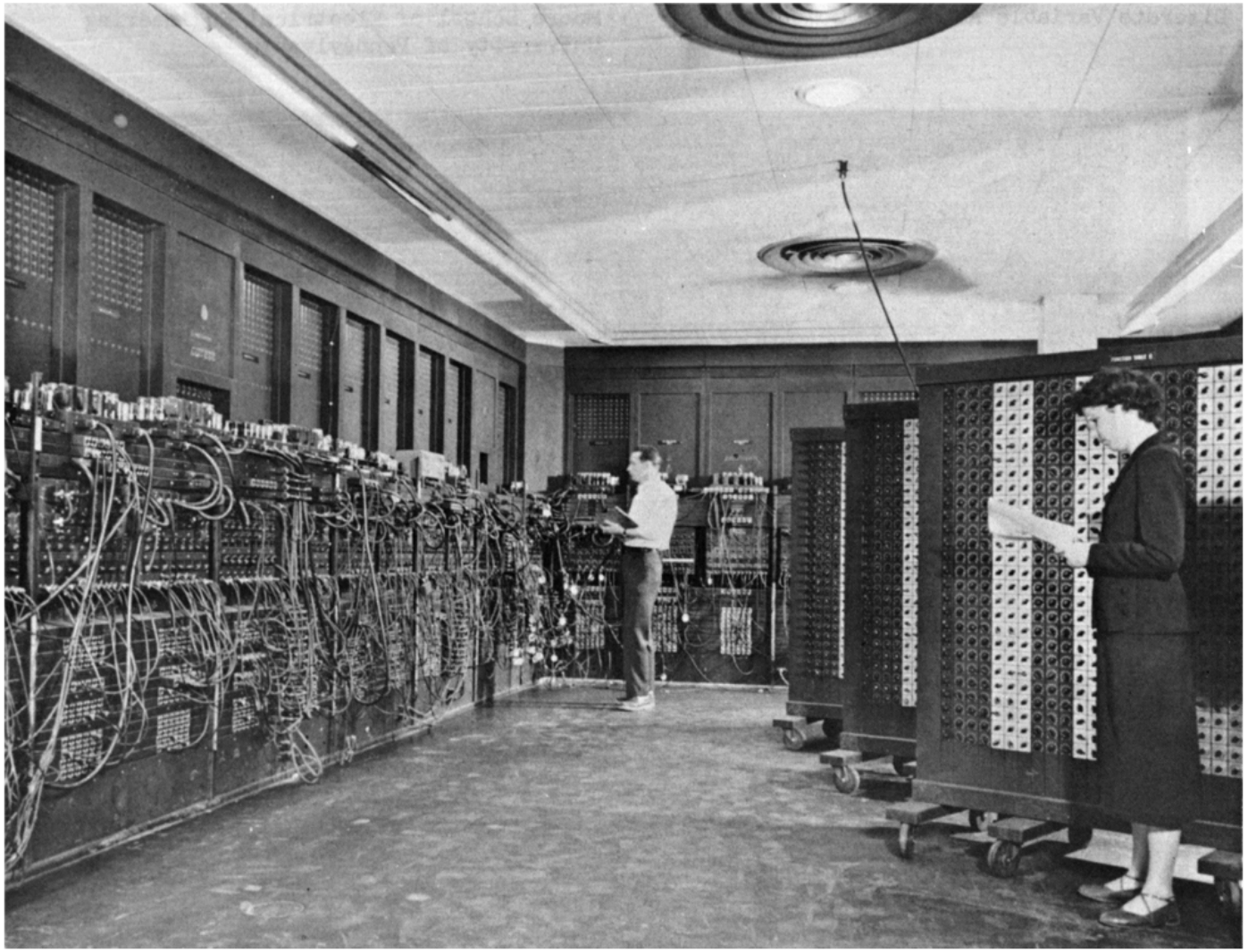
"What are the chances that a Canfield solitaire laid out with 52 cards will come out successfully?

After spending a lot of time trying to estimate them by pure combinatorial calculations, I wondered whether **a more practical method than abstract thinking** might not be to **lay it out say one hundred times and simply observe and count the number of successful play**"

history of MC

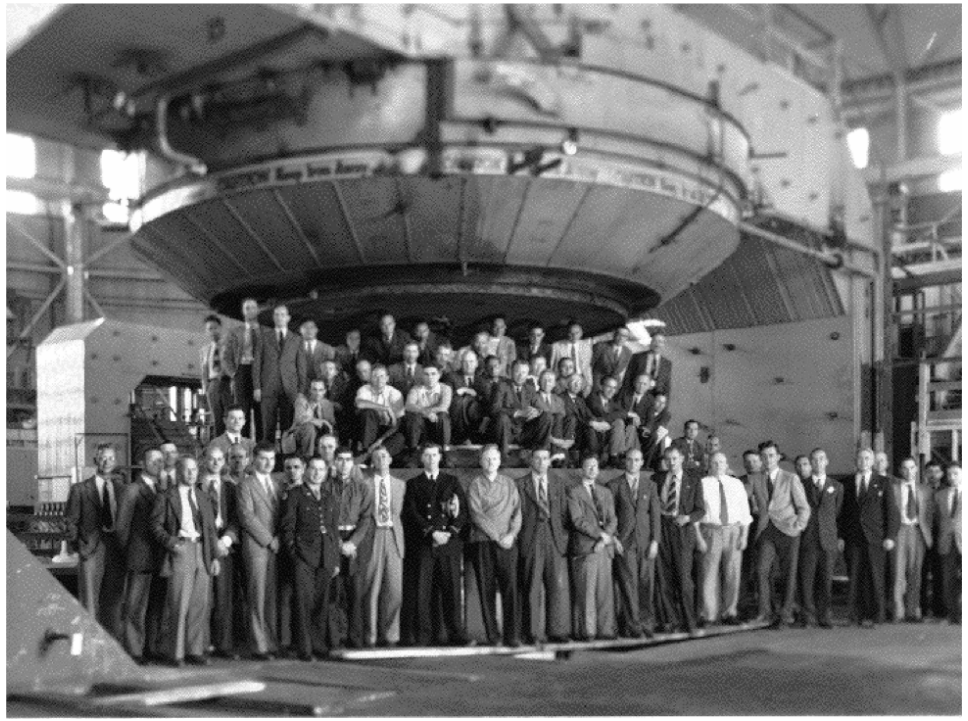


history of MC



ENIAC It weighed more than 30 short tons (27 t), was roughly 2.4 m × 0.9 m × 30 m (8 × 3 × 100 feet) in size, occupied 167 m^2 (1,800 ft^2), consumed 150 kW of electricity.
500FLOPS vs today's Macbook pro ~1TeraFLOP

http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-88-9068

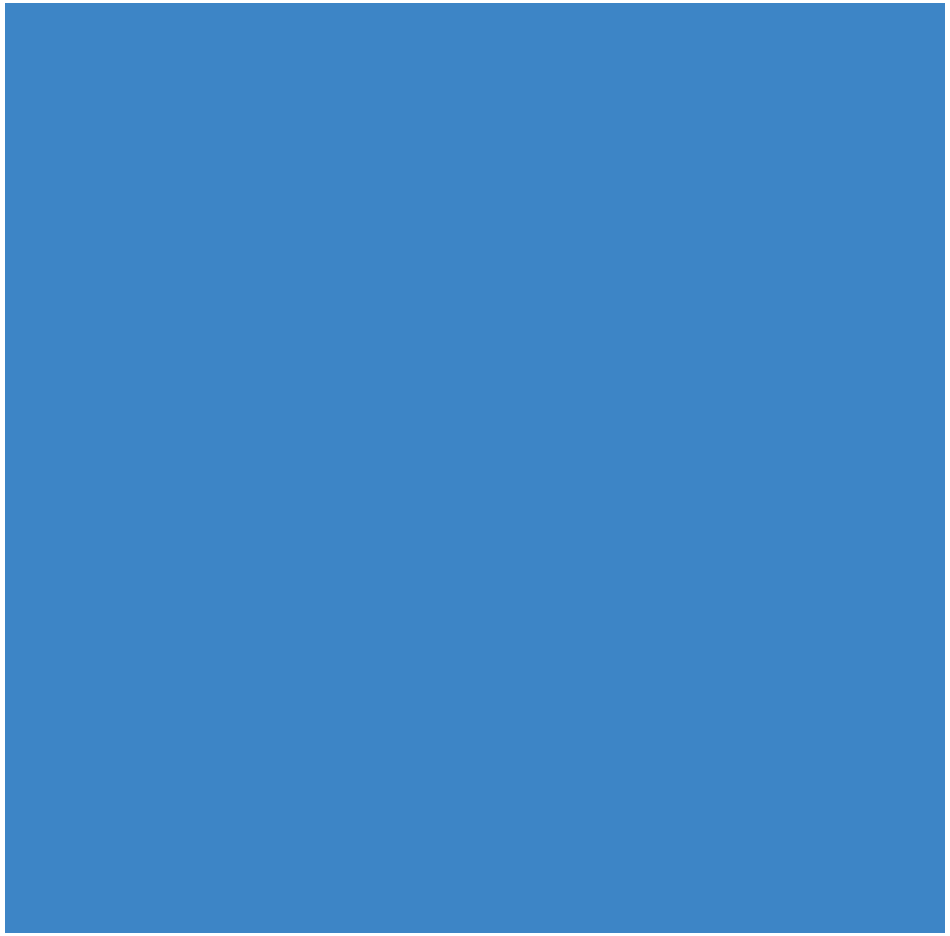


The Manhattan Project

The advent of computing allowed for major innovation in the realm of simulation. Metropolis led a group that developed the Monte Carlo method, which simulates the results of an experiment by using a broad set of random numbers

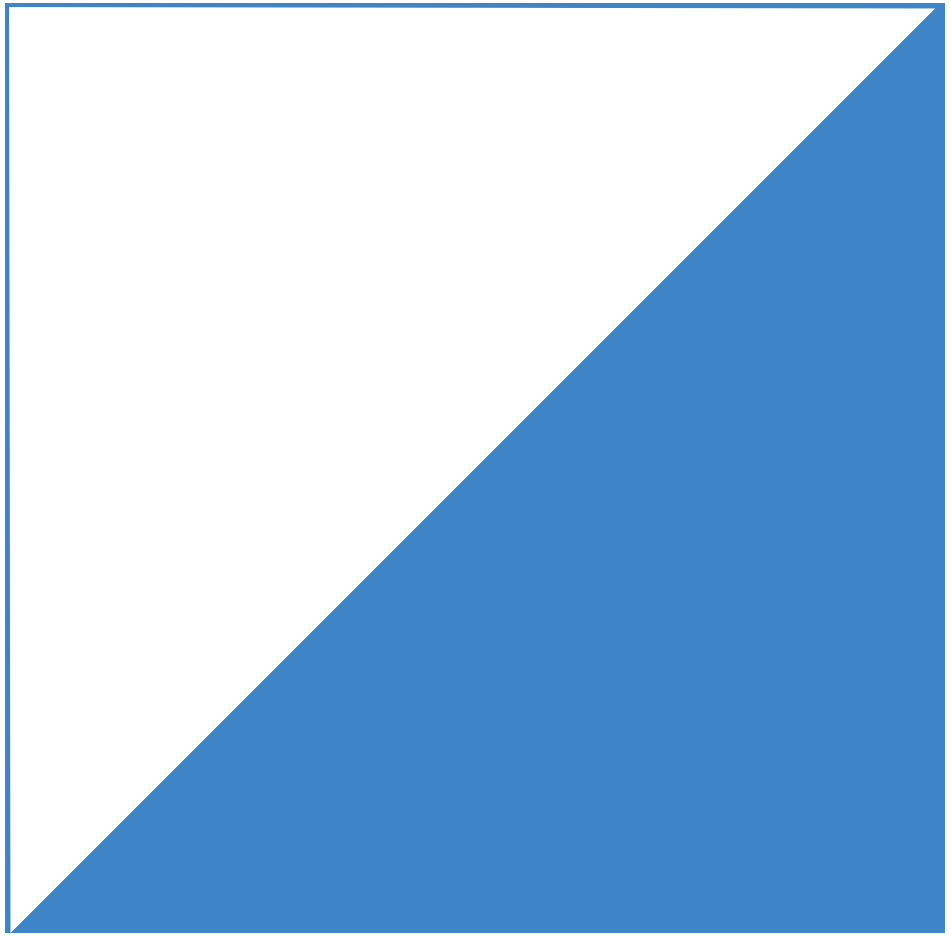
42

simple example



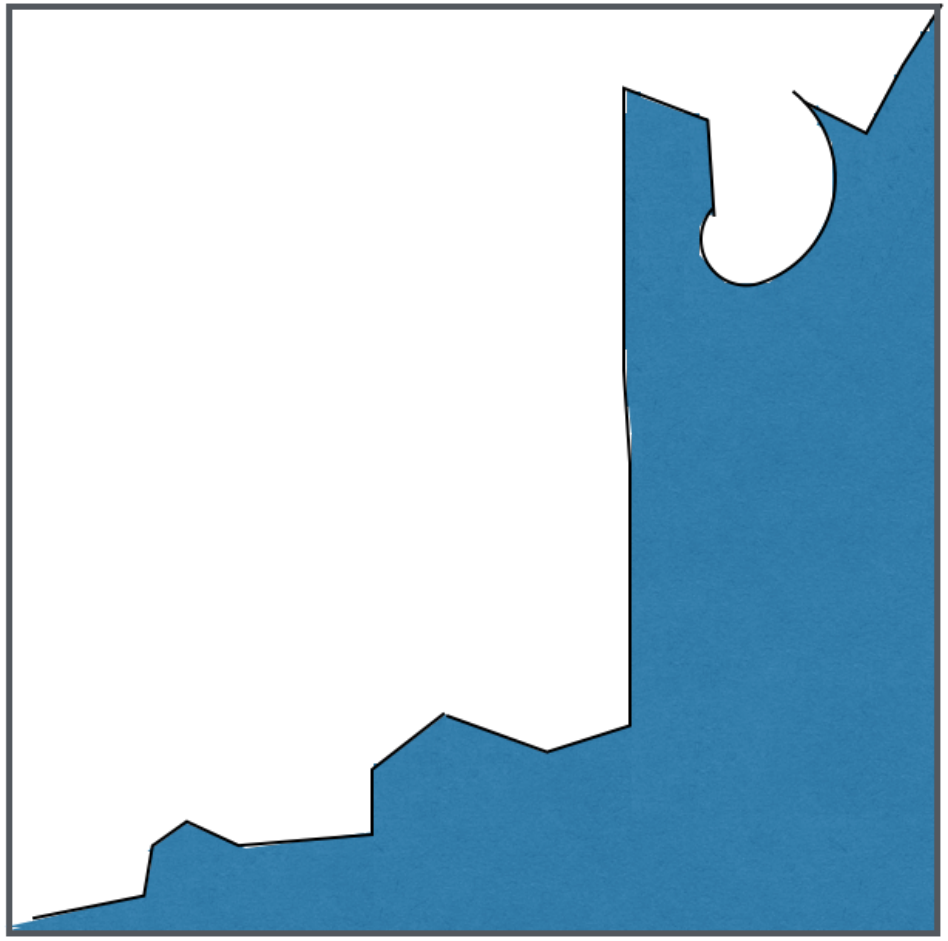
$$\text{Area} = \text{Base x Height}$$

simple example



$$\text{Area} = \frac{\text{Base x Height}}{2}$$

simple example

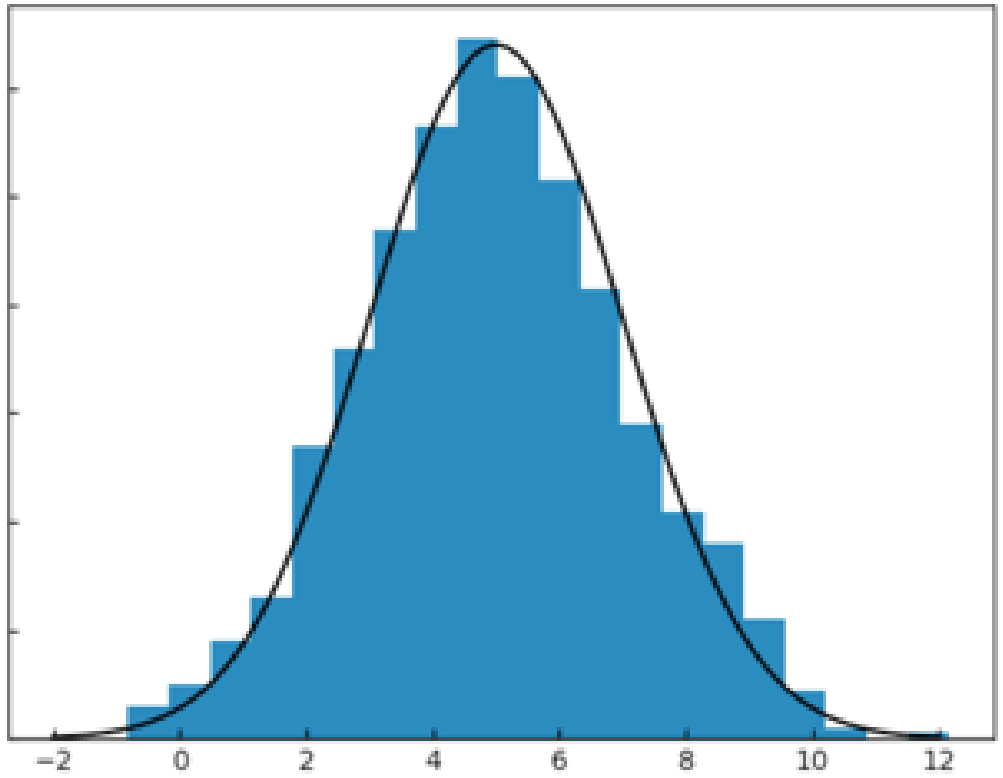


MCArea.ipynb

$$\text{Area} = ???$$

simple example

Why am I bothering with areas? - Expectation values are related to areas



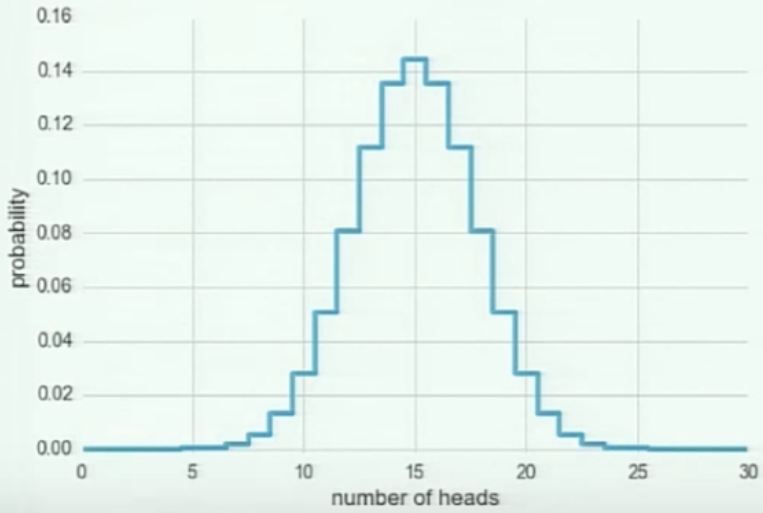
$$mean(X) = E[X] = \int X f(X) dX$$

$$Var(X) = E[X^2] - (E[X])^2.$$

Classic Method:

$N_H = 22,\ N_T = 8$

$$P(N_H, N_T) = \binom{N}{N_H}\left(\frac{1}{2}\right)^{N_H}\left(1-\frac{1}{2}\right)^{N_T}$$



Easier Method:

Just simulate it!

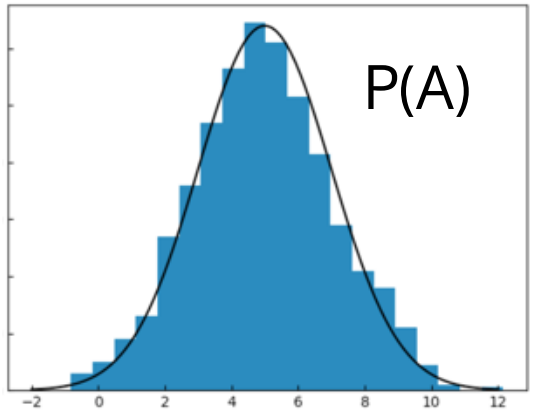
```
M = 0
for i in range(10000):
    trials = randint(2, size=30)
    if (trials.sum() >= 22):
        M += 1
p = M / 10000 # 0.008149
```

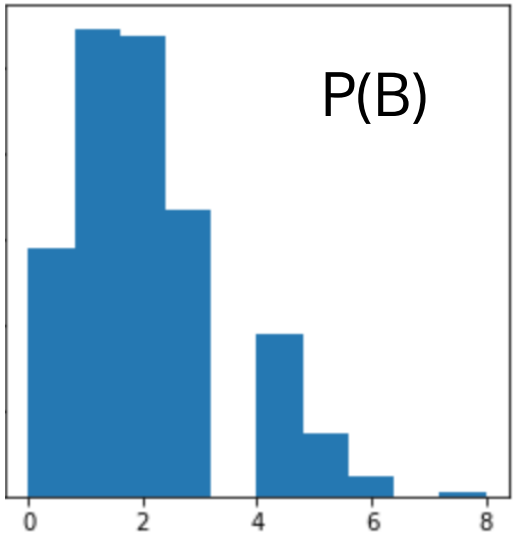
→ reject fair coin at p = 0.008

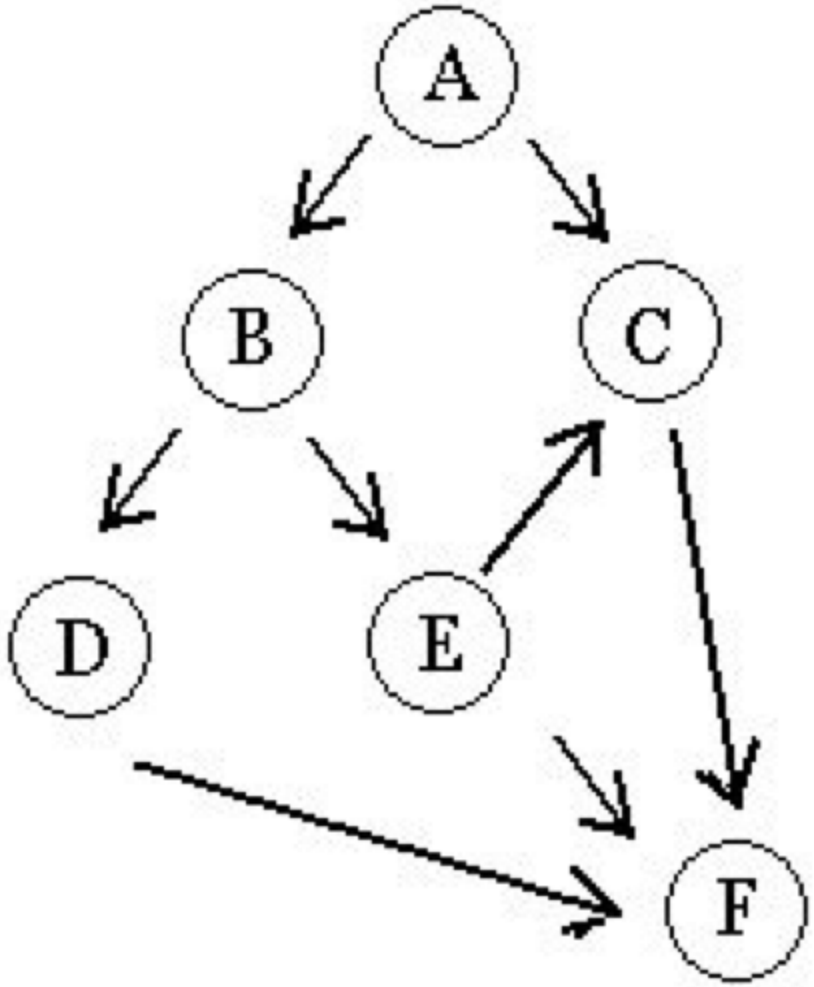
Statistics for Hackers, Jake Vanderplas PyCon16

https://www.youtube.com/watch?v=Iq9DzN6mvYA

Why am I bothering with areas? - Expectation values are related to areas







A ~ P(A)

B ~ P(B|A)

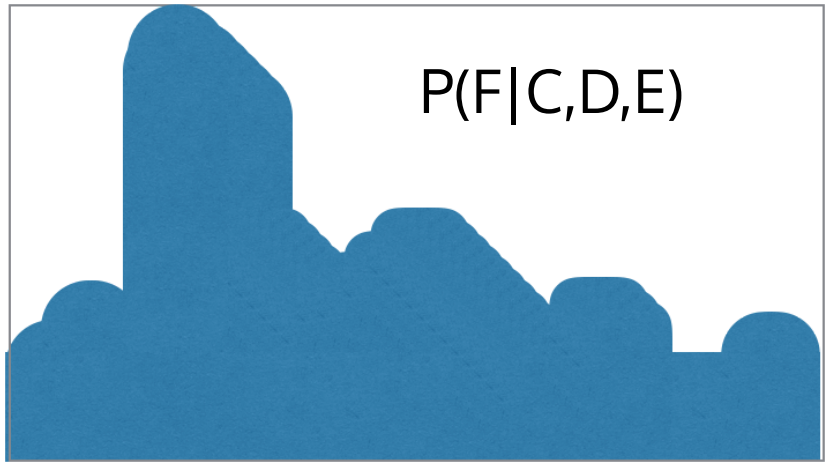
C ~ P(C|A,E)

D ~ P(D|B)

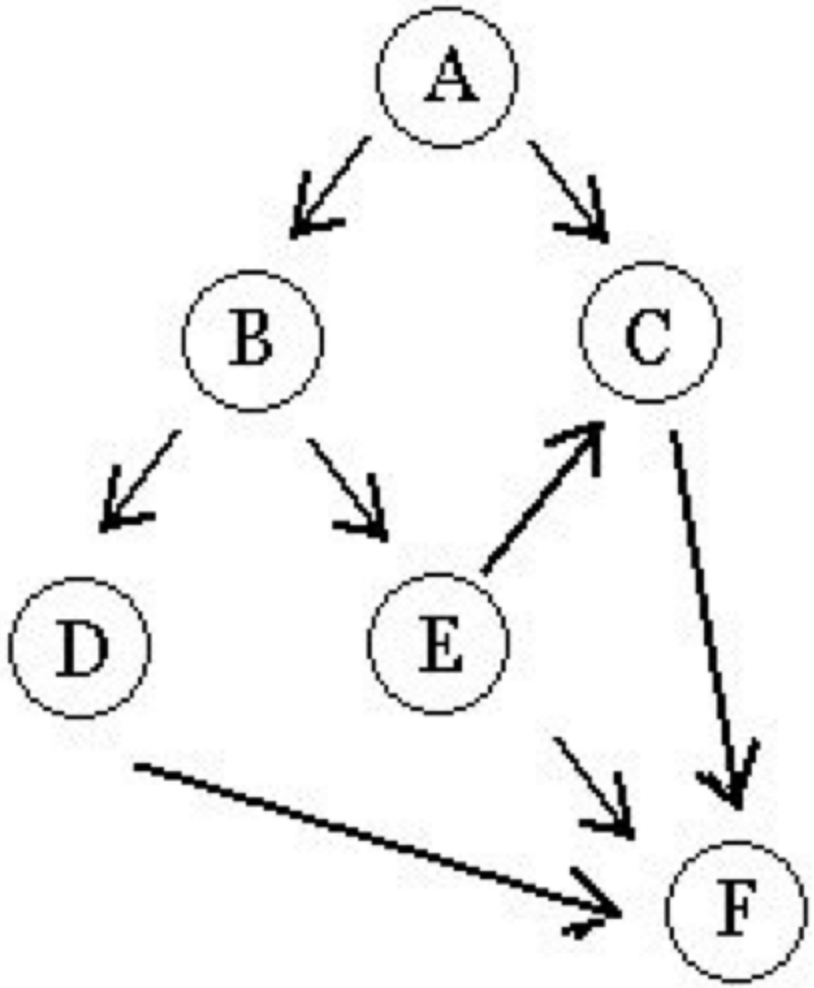
E ~ P(E|B)

F ~ P(F|C,D,E)

Why am I bothering with areas? - Expectation values are related to areas



The final probability is likely very complicated (especially if this is a complex system with feedback loops as many physics systems, e.g. radiative transfer!). It may not be tractable analytically but can be simulated



$A \sim P(A)$

$B \sim P(B|A)$

$C \sim P(C|A,E)$

$D \sim P(D|B)$

$E \sim P(E|B)$

$F \sim P(F|C,D,E)$

Why am I bothering with areas? - Expectation values are related to areas

A person's depth

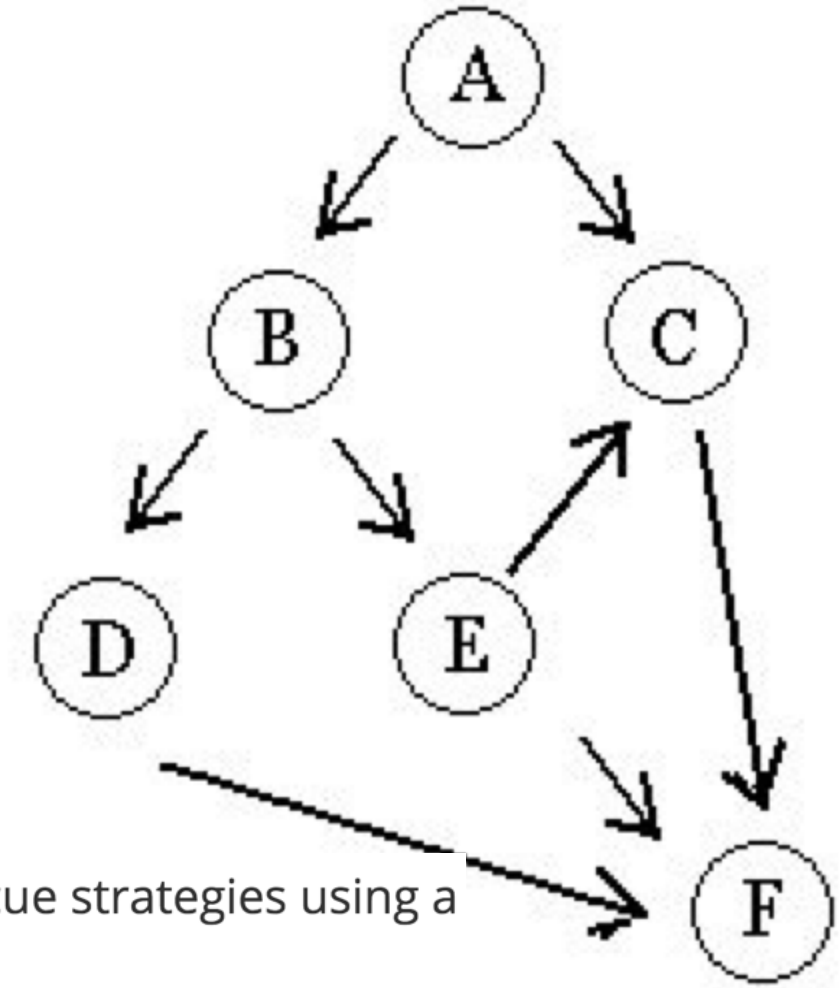
B prob to find them at time t

C they survive the avalange

D they are still alive at t

E can be resuscitated at time t

F person survives



$$A \sim P(A)$$
$$B \sim P(B|A)$$
$$C \sim P(C|A,E)$$
$$D \sim P(D|B)$$
$$E \sim P(E|B)$$
$$F \sim P(F|C,D,E)$$

A concept for optimizing avalanche rescue strategies using a Monte Carlo simulation approach

Ingrid Reiweger, Manuel Genswein, Peter Paal, Jürg Schweizer

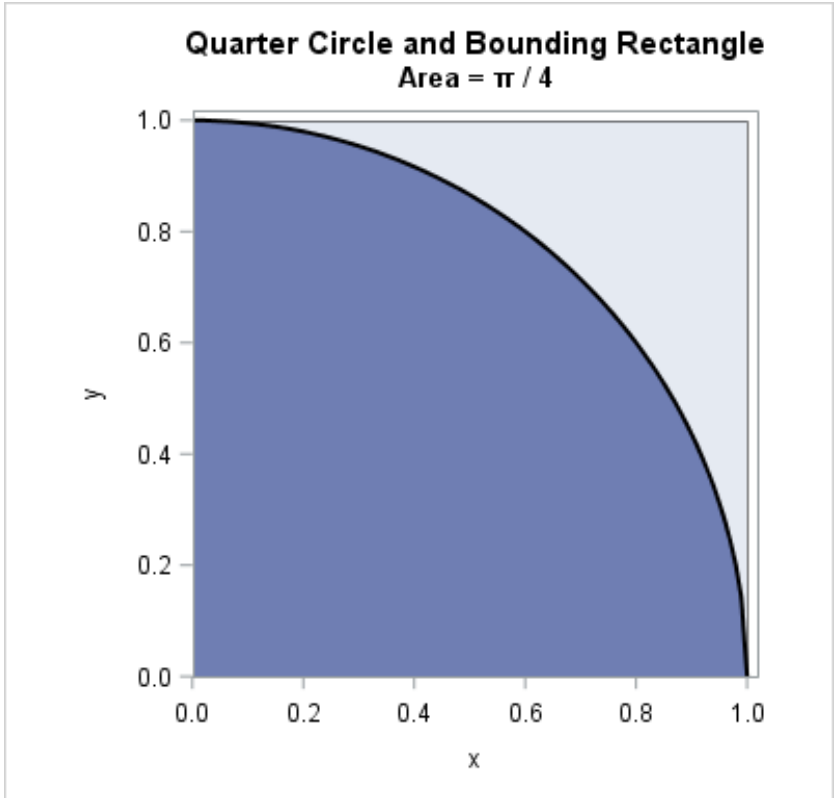
Published: May 3, 2017 • https://doi.org/10.1371/journal.pone.0175877

Why am I bothering with areas? - Expectation values are related to areas

The ratio of the area of the circle to the area of the square is π / 4.

Calculate Pi

https://github.com/fedhere/DSPS_FBianco/tree/master/montecarlo



JOURNAL ARTICLE

Determining Sample Sizes for Monte Carlo Integration

David Neal

The College Mathematics Journal
Vol. 24, No. 3 (May, 1993), pp. 254-259
(6 pages)
Published By: Taylor & Francis, Ltd.

https://doi.org/10.2307/2686489
https://www.jstor.org/stable/2686489

https://www.jstor.org/stable/2686489?seq=1

https://arxiv.org/pdf/1710.06068.pdf

read section 1 & 2

DATA ANALYSIS RECIPES: USING MARKOV CHAIN MONTE CARLO*

David W. Hogg[1,2,3,4] and Daniel Foreman-Mackey[1,5]

ABSTRACT

Markov Chain Monte Carlo (MCMC) methods for sampling probability density functions (combined with abundant computational resources) have transformed the sciences, especially in performing probabilistic inferences, or fitting models to data. In this primarily pedagogical contribution, we give a brief overview of the most basic MCMC method and some practical advice for the use of MCMC in real inference problems. We give advice on method choice, tuning for performance, methods for initialization, tests of convergence, troubleshooting, and use of the chain output to produce or report parameter estimates with associated uncertainties. We argue that autocorrelation time is the most important test for convergence, as it directly connects to the uncertainty on the sampling estimate of any quantity of interest. We emphasize that sampling is a method for doing integrals; this guides our thinking about how MCMC output is best used.

1 what is machine learning

what is machine learning?

[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.

Arthur Samuel, 1959

what is a model?

a *mathematical* representastion of reality

In applying mathematics to subjects such as physics or statistics we make tentative assumptions about the real world which we know are false but which we believe may be useful nonetheless.

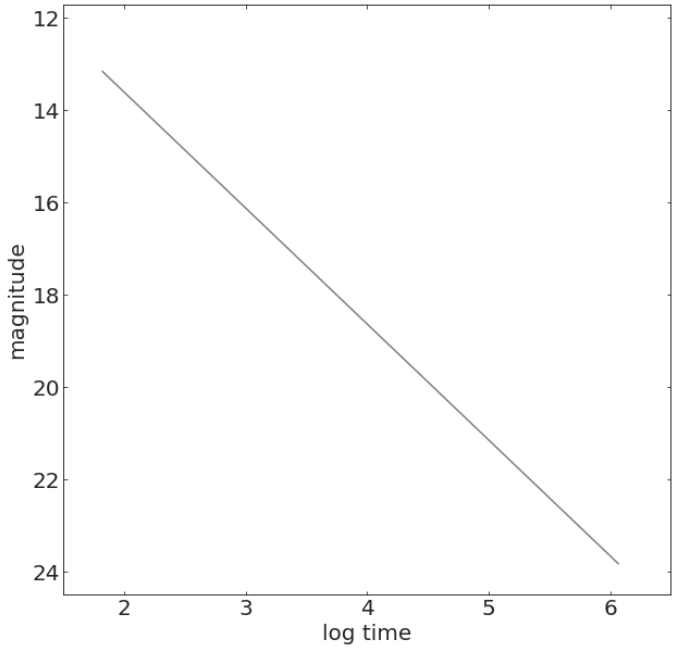
George Box, 1976

- no model is right
- some models are useful

what is machine learning?

[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.

Arthur Samuel, 1959



Model:

a mathematical formula
with parameters

model

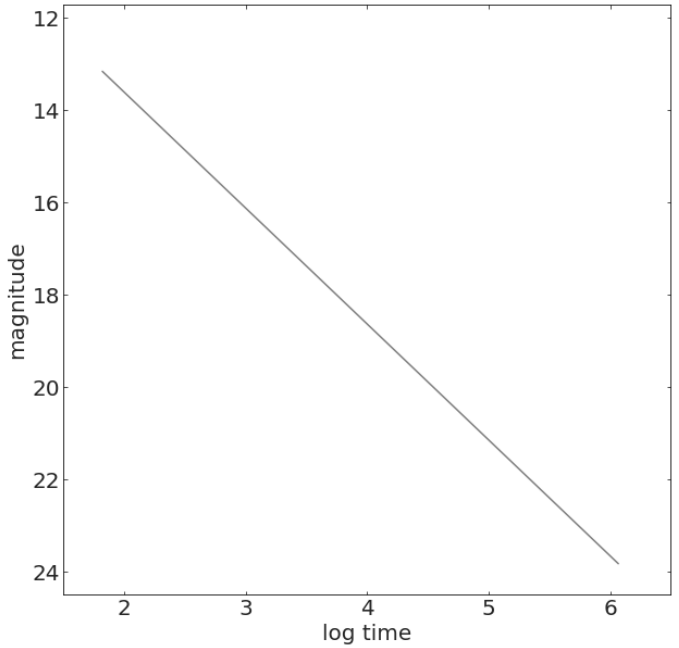
parameters: slope (*a*), intercept (*b*)

$$y = ax + b$$

what is machine learning?

[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.

Arthur Samuel, 1959



Model:

a mathematical formula
with parameters

model

parameters: slope (*a*), intercept (*b*)

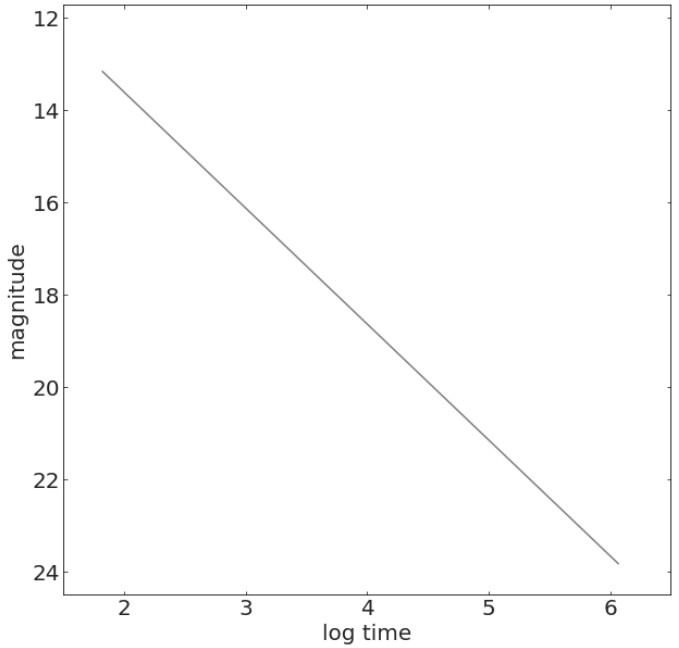
$$y = ax + b$$

variable: x - for example time, location, energy

what is machine learning?

[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.

Arthur Samuel, 1959



Model:

a mathematical formula
with parameters

model

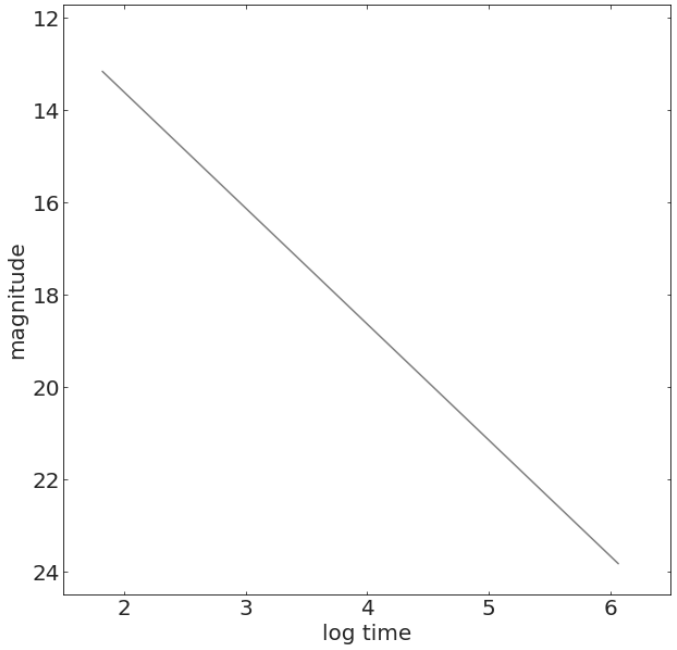
$$y = a_1 x_1 + a_2 x_2 + b$$

variable: x_1, x_2 - for example time and location

what is machine learning?

[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.

Arthur Samuel, 1959

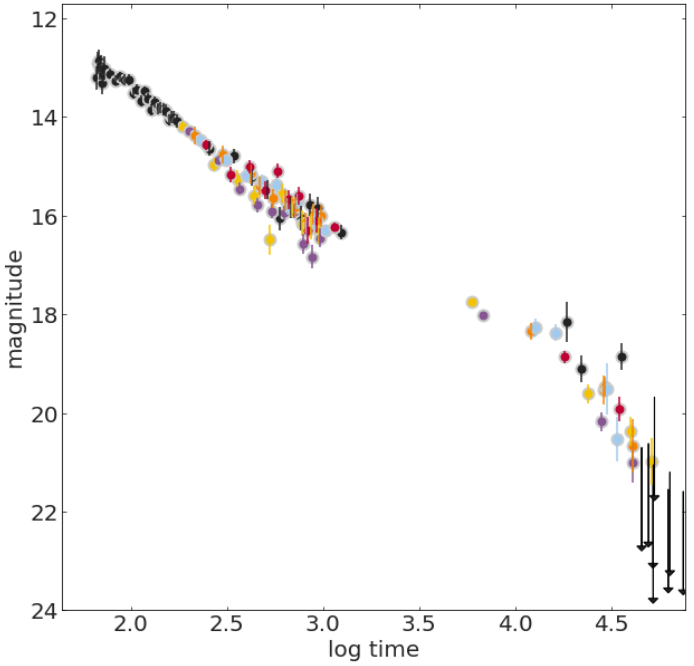


Model:

a mathematical formula with parameters

Data:

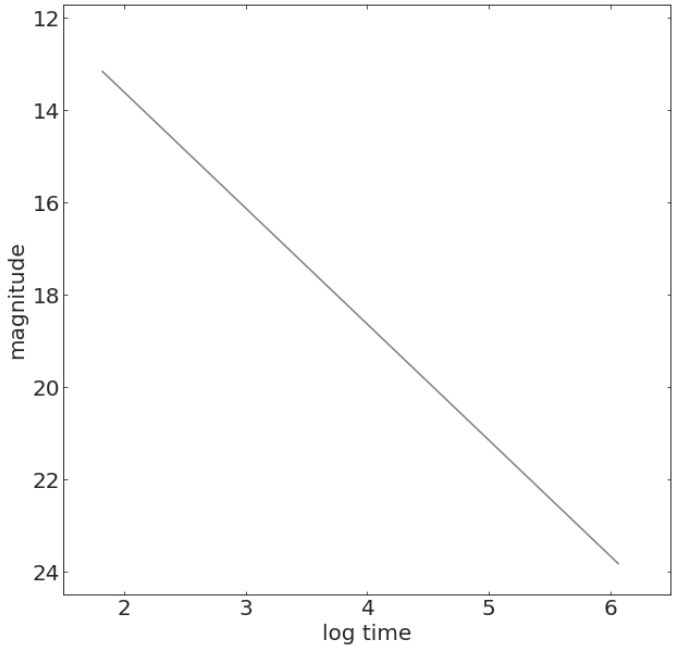
a set of observations



what is machine learning?

[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.

Arthur Samuel, 1959

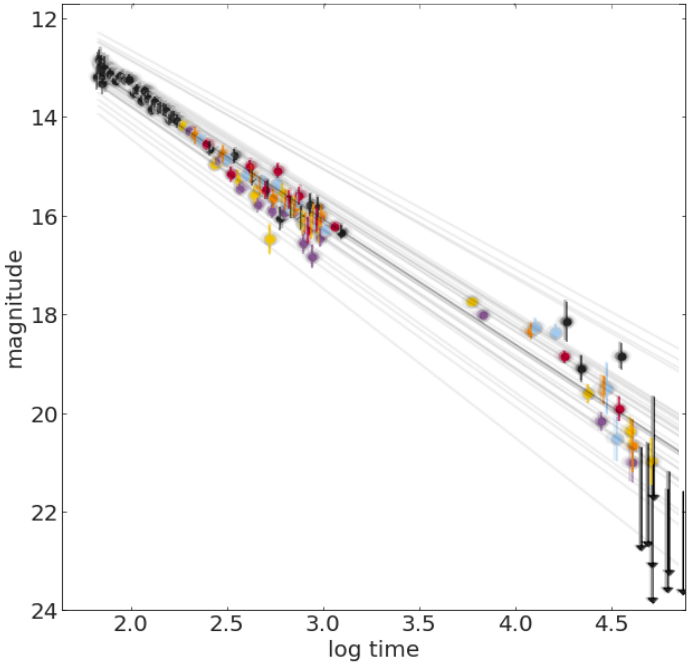


Model:

a mathematical formula
with parameters

Data:

a set of
observations

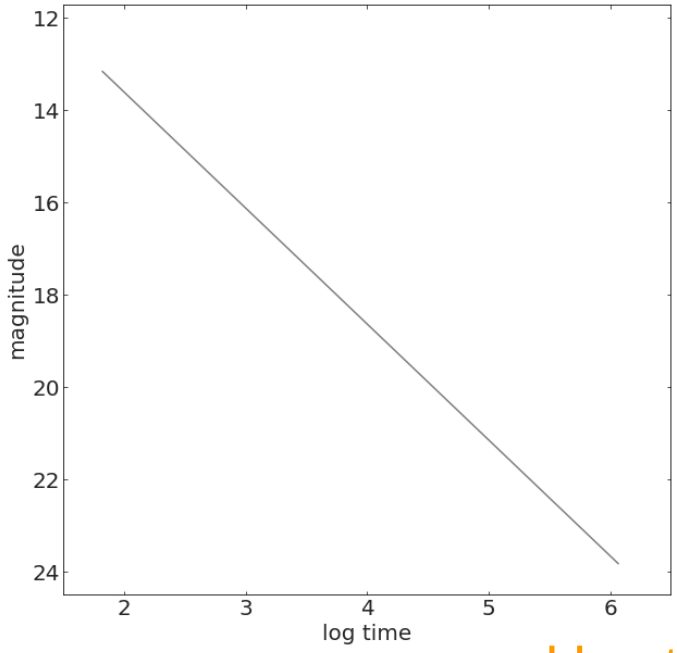


for every parameter there are an infinity of models

what is machine learning?

[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.

Arthur Samuel, 1959

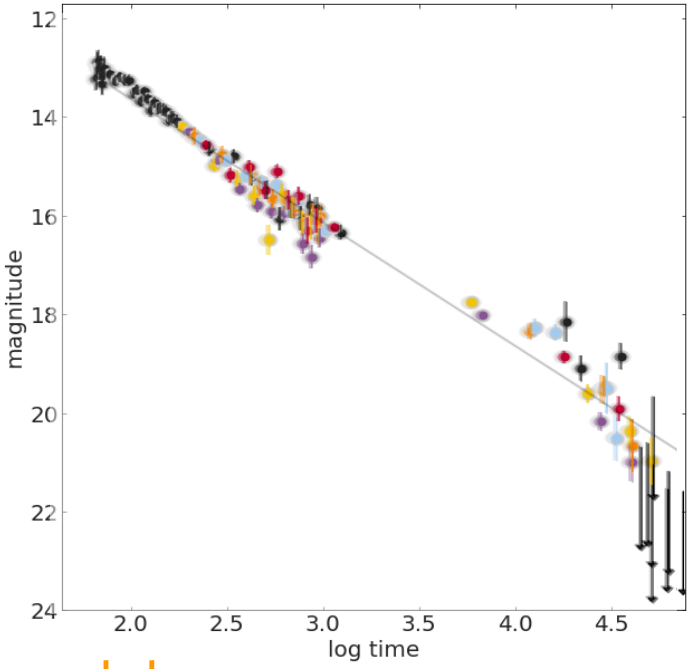


Model:

a mathematical formula with parameters

Data:

a set of observations



Use the data to *learn* the parameters of the model

what is machine learning?

Machine Learning models are parametrized representation of "reality" where the parameters are learned from finite sets of realizations of that reality

Machine Learning is the disciplines that conceptualizes, studies, and applies those models.

Key Concept

1.2

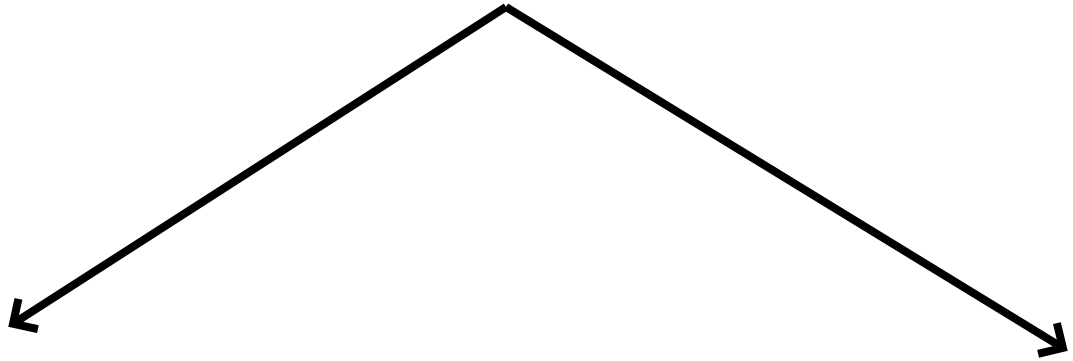
what is machine learning

unsupervised vs supervised learning

used to:

- understand structure of feature space
- classify based on examples,
- regression (classification with infinitely small classes)
- understand which features are important in prediction (to get close to causality)

what is machine learning?



supervised learning

extract features and create models that allow prediction where the correct answer is known for a subset of the data

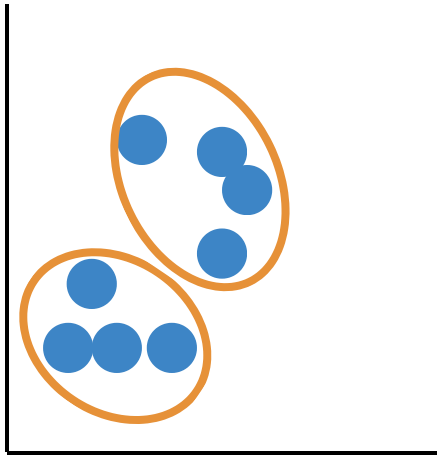
unsupervised learning

identify features and create models that allow to understand structure in the data

unsupervised vs supervised learning

Unsupervised learning

- understanding structure
- anomaly detection
- dimensionality reduction



Clustering

partitioning the feature space so that the existing data is grouped (according to some target function!)

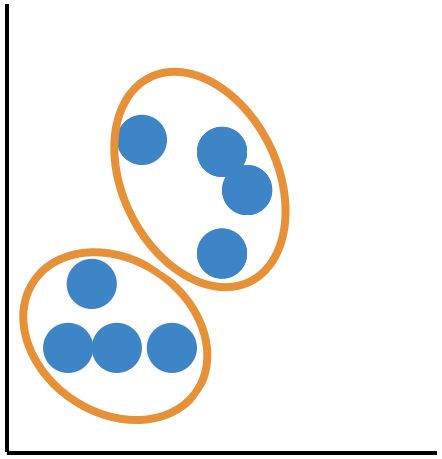
All features are observed for all datapoints

unsupervised vs supervised learning

prediction and classification based on examples

Unsupervised learning

- understanding structure
- anomaly detection
- dimensionality reduction



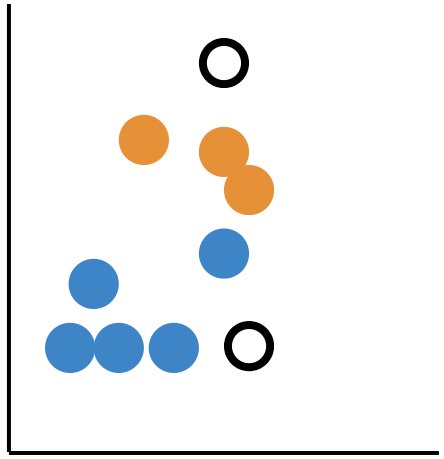
Clustering

partitioning the feature space so that the existing data is grouped (according to some target function!)

All features are observed for all datapoints

Supervised learning

- classification
- prediction
- feature selection



Classifying & regression

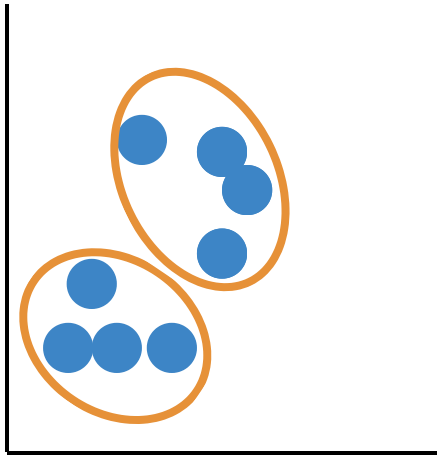
finding functions of the variables that allow to predict unobserved properties of new observations

unsupervised vs supervised learning

prediction and classification based on examples

Unsupervised learning

- understanding structure
- anomaly detection
- dimensionality reduction



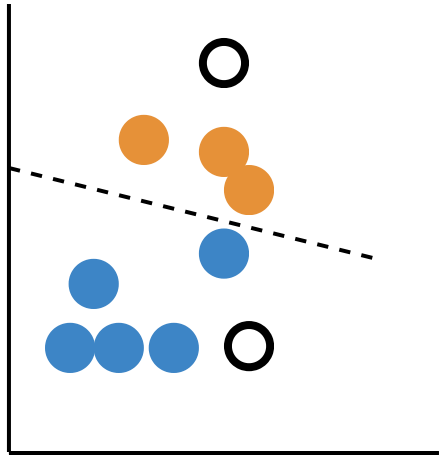
Clustering

partitioning the feature space so that the existing data is grouped (according to some target function!)

All features are observed for all datapoints

Supervised learning

- classification
- prediction
- feature selection



Classifying & regression

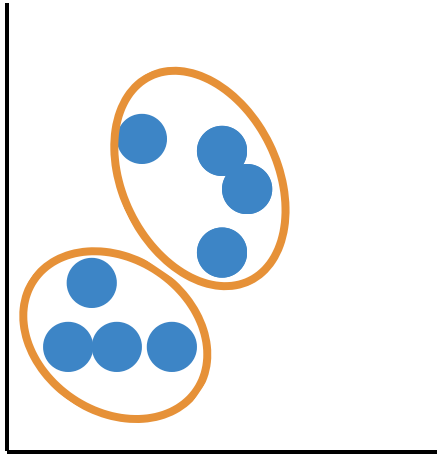
finding functions of the variables that allow to predict unobserved properties of new observations

unsupervised vs supervised learning

prediction and classification based on examples

Unsupervised learning

- understanding structure
- anomaly detection
- dimensionality reduction



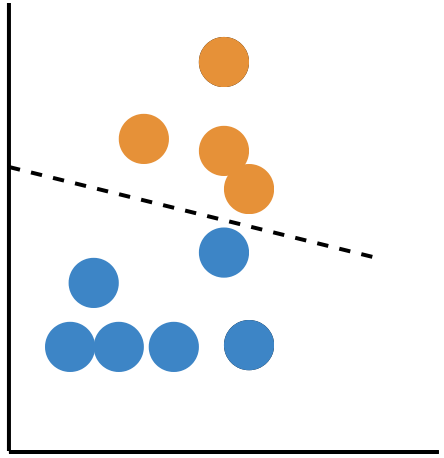
Clustering

partitioning the feature space so that the existing data is grouped (according to some target function!)

All features are observed for all datapoints

Supervised learning

- classification
- prediction
- feature selection



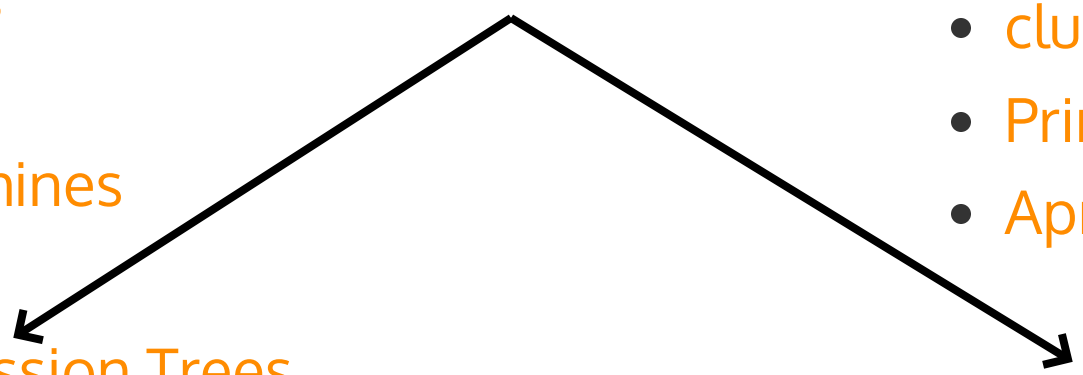
Classifying & regression

finding functions of the variables that allow to predict unobserved properties of new observations

Some features are not observed for some data points we want to predict them.

what is machine learning?

- k-Nearest Neighbors
- Regression
- Support Vector Machines
- Neural networks
- Classification/Regression Trees



- clustering
- Principle Component Analsysis
- Apriori (association rule)

supervised learning

extract features and create models that allow prediction where the correct answer is known for a subset of the data

unsupervised learning

identify features and create models that allow to understand structure in the data

unsupervised vs supervised learning

Unsupervised learning

All features are observed for all datapoints

and we are looking for structure in the feature space

Supervised learning

Some features are not observed for some data points we want to predict them.

The datapoints for which the target feature is observed are said to be "*labeled*"

also...

Semi-supervised learning

A small amount of labeled data is available. Data is cluster and clusters inherit labels

Active learning

The code can interact with the user to update labels and update model.

2.1

Linear Regression

analytical solution

Linear Regression Normal Equation

It can be shown that the optimal parameters for a line fit to data without uncertainties is:

$$(X^T \cdot X)^{-1} \cdot X^T \cdot \vec{y} = \begin{pmatrix} a \\ b \end{pmatrix}$$

```
X = np.c_[np.ones((len(grbAG) - grbAG.upperlimit.sum(), 1)),
          grbAG[grbAG.upperlimit == 0].logtime]
y = grbAG.loc[grbAG.upperlimit == 0].mag

theta_best = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(y)
```

Linear Regression Normal Equation

It can be shown that the optimal parameters for a line fit to data without uncertainties is:

$$(X^T \cdot X)^{-1} \cdot X^T \cdot \vec{y} = \begin{pmatrix} a \\ b \end{pmatrix}$$

```
1 X = np.c_[np.ones((len(grbAG) - grbAG.upperlimit.sum(), 1)),
2         grbAG[grbAG.upperlimit == 0].logtime]
3 y = grbAG.loc[grbAG.upperlimit == 0].mag
4
5 theta_best = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(y)
```

Linear Regression Normal Equation

It can be shown that the optimal parameters for a line fit to data without uncertainties is:

$$(X^T \cdot X)^{-1} \cdot X^T \cdot \vec{y} = \begin{pmatrix} a \\ b \end{pmatrix}$$

2xN Nx2 2xN Nx1

$$x = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix}$$

```
X = np.c_[np.ones((len(grbAG) - grbAG.upperlimit.sum(), 1)),
          grbAG[grbAG.upperlimit == 0].logtime]
y = grbAG.loc[grbAG.upperlimit == 0].mag

theta_best = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(y)
```

Linear Regression Normal Equation

It can be shown that the optimal parameters for a line fit to data without uncertainties is:

$$(X^T \cdot X)^{-1} \cdot X^T \cdot \vec{y} = \begin{pmatrix} a \\ b \end{pmatrix}$$

2xN Nx2 2xN Nx1

$$x = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{pmatrix}$$

```
X = np.c_[np.ones((len(grbAG) - grbAG.upperlimit.sum(), 1)),
          grbAG[grbAG.upperlimit == 0].logtime]
y = grbAG.loc[grbAG.upperlimit == 0].mag

theta_best = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(y)
```

Linear Regression Normal Equation

It can be shown that the optimal parameters for a line fit to data without uncertainties is:

$$(X^T \cdot X)^{-1} \cdot X^T \cdot \vec{y} = \begin{pmatrix} a \\ b \end{pmatrix}$$

2xN Nx2 2xN Nx1 2x1

```
X = np.c_[np.ones((len(grbAG) - grbAG.upperlimit.sum(), 1)),
          grbAG[grbAG.upperlimit == 0].logtime]
y = grbAG.loc[grbAG.upperlimit == 0].mag

theta_best = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(y)
```

We can let sklearn solve the equation for us:

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()

X = np.c_[np.ones((len(grbAG) -
          grbAG.upperlimit.sum(), 1)),
          grbAG[grbAG.upperlimit == 0].logtime]
y = grbAG.loc[grbAG.upperlimit == 0].mag
lr.fit(X, y)
lr.coef_, lr.intercept_
```

Linear Regression Normal Equation

It can be shown that the optimal parameters for a line fit to data without uncertainties is:

$$(X^T \cdot X)^{-1} \cdot X^T \cdot \vec{y} = \begin{pmatrix} a \\ b \end{pmatrix}$$

2xN Nx2 2xN Nx1 2x1

```
X = np.c_[np.ones((len(grbAG) - grbAG.upperlimit.sum(), 1)),
          grbAG[grbAG.upperlimit == 0].logtime]
y = grbAG.loc[grbAG.upperlimit == 0].mag

theta_best = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(y)
```

We can let sklearn solve the equation for us:

```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()

X = np.c_[np.ones((len(grbAG) -
          grbAG.upperlimit.sum(), 1)),
          grbAG[grbAG.upperlimit == 0].logtime]
y = grbAG.loc[grbAG.upperlimit == 0].mag
lr.fit(X, y)
lr.coef_, lr.intercept_
```

2.2 Regression

objective function

Objective Function

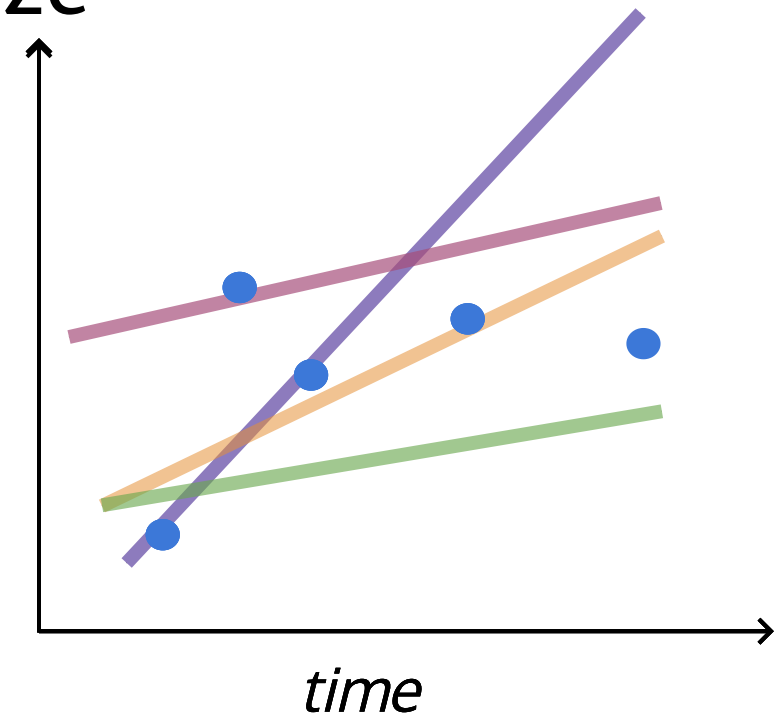
If there is no analytical solution

to select the "best" set of parameters we need a plan: we need to choose a function of the parameters to minimize or maximize

Objective Function

If there is no analytical solution

to select the best fit parameters we define a function of the parameters to minimize or maximize



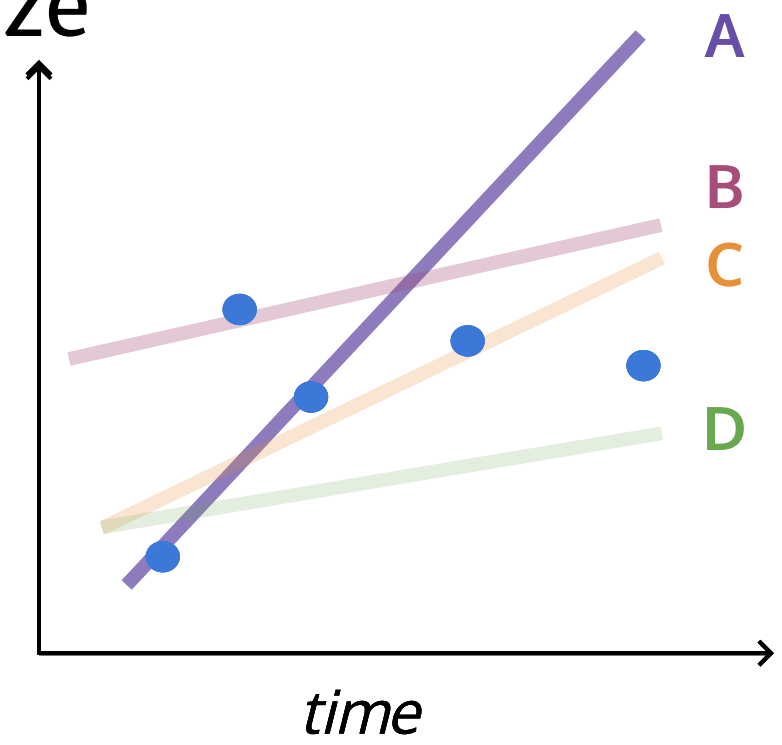
Objective Function

If there is no analytical solution

to select the best fit parameters we define a function of the parameters to minimize or maximize

which is the "best fit" line?

A , B, C, D?



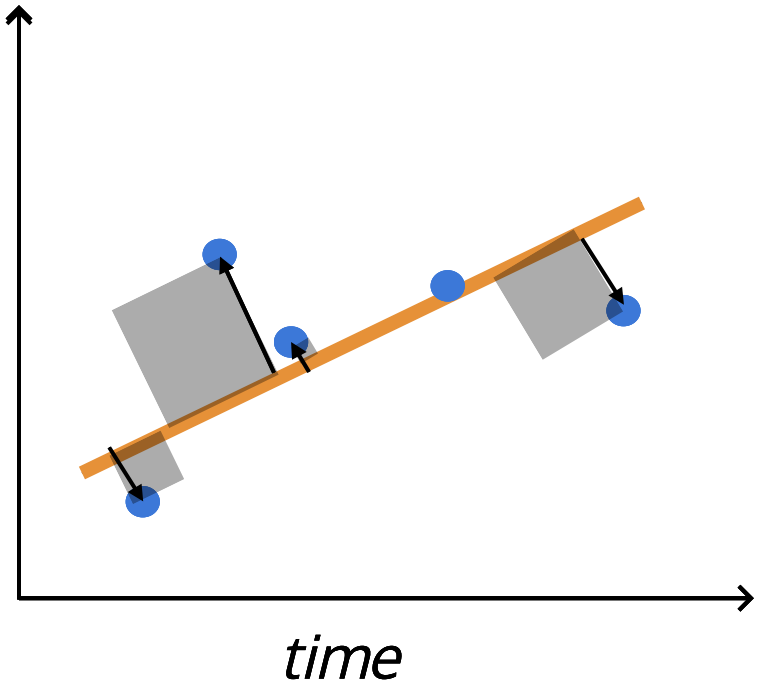
Objective Function

If there is no analytical solution

to select the best fit parameters we define a function of the parameters to minimize or maximize

$$L_1 = \sum_{i=1}^{N} |f(x) - y|$$

$$L_2 = \sum_{i=1}^{N} (f(x) - y)^2$$



Objective Function

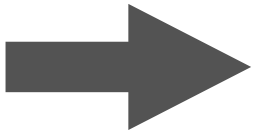
If there is no analytical solution

to select the best fit parameters we define a function of the parameters to minimize or maximize

$$L_1 = \sum_{i=1}^{N} |f(x) - y|$$

$$L_2 = \sum_{i=1}^{N} (f(x) - y)^2$$

$$\chi^2 = \sum_{i=1}^{N} \frac{(f(x)-y)^2}{\sigma^2}$$



chi square: relates to the likelihood if the distribution is Gaussian

Objective Function

If there is no analytical solution

to select the "best" set of parameters we need a plan:
we need to choose a function of the parameters to minimize or maximize

$$L_1 = \sum_{i=1}^{N} |f(x) - y|$$

$$L_2 = \sum_{i=1}^{N} (f(x) - y)^2$$

$$\chi^2 = \sum_{i=1}^{N} \frac{(f(x) - y)^2}{\sigma^2}$$

```
from scipy.optimize import minimize
def line(x, b, a):
    return a * x + b
def fitfunc(args, x, y):
    a, b = args
    return sum((y - line(a, b, x))**2)

x = grbAG[grbAG.upperlimit == 0].logtime.values
y = grbAG.loc[grbAG.upperlimit == 0].mag.values
initialGuess = (10, 1)

fitfunc(initialGuess, x, y)
solution = minimize(fitfunc, initialGuess, args=(x, y))
```

Objective Function

If there is no analytical solution

to select the "best" set of parameters we need a plan:
we need to choose a function of the parameters to minimize or maximize

$$L_1 = \sum_{i=1}^{N} |f(x) - y|$$

$$L_2 = \sum_{i=1}^{N} (f(x) - y)^2$$

$$\chi^2 = \sum_{i=1}^{N} \frac{(f(x)-y)^2}{\sigma^2}$$

```
1  from scipy.optimize import minimize
2  def line(x, b, a):
3      return a * x + b
4  def chi2(args, x, y, s):
5      a, b = args
6      return sum((y - line(x, b, a))**2 / s**2)
7
8  x = grbAG[grbAG.upperlimit == 0].logtime.values
9  y = grbAG.loc[grbAG.upperlimit == 0].mag.values
10 s = grbAG.loc[grbAG.upperlimit == 0].magerr.values
11 initialGuess = (10, 1)
12
13 fitfunc(initialGuess, x, y)
14 solution = minimize(chi2, initialGuess, args=(x, y, s))
15 solution
```

What is Machine Learning? Machine Learning models are parametrized representations of "reality" where the parameters are learned from finite sets of realizations of that reality. Machine Learning is the discipline that conceptualizes, studies, and applies those models.

Model selection: Choosing a model i.e. a mathematical formula which we expect to be a simplified representation of our observations.

Model fitting: Determining the best set of parameters to fit the observations within a chosen model.

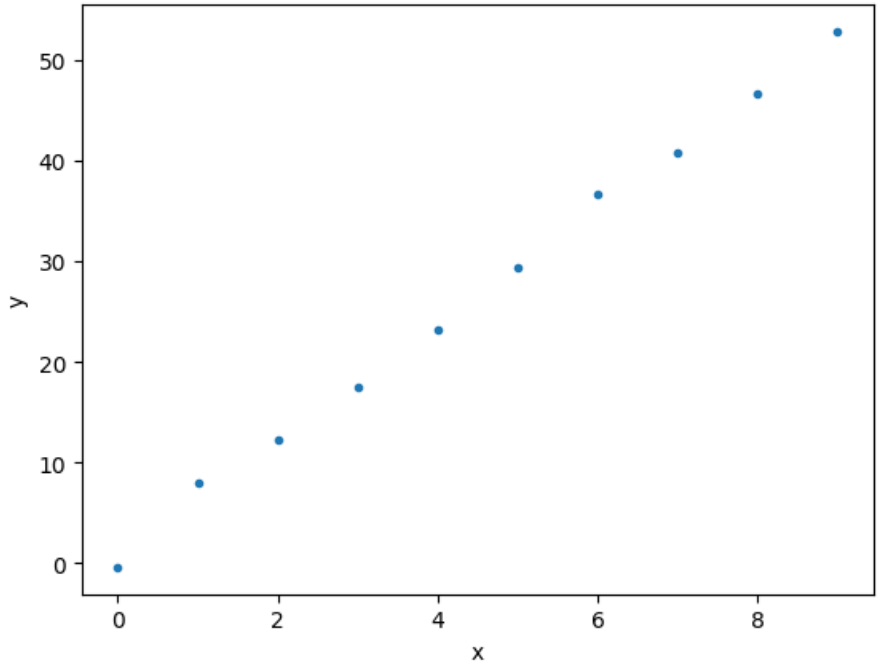
Objective Functions and optimization: To find the best model parameters we define a function of the data and parameters *f(data, parameters)* to be minimized or maximized.

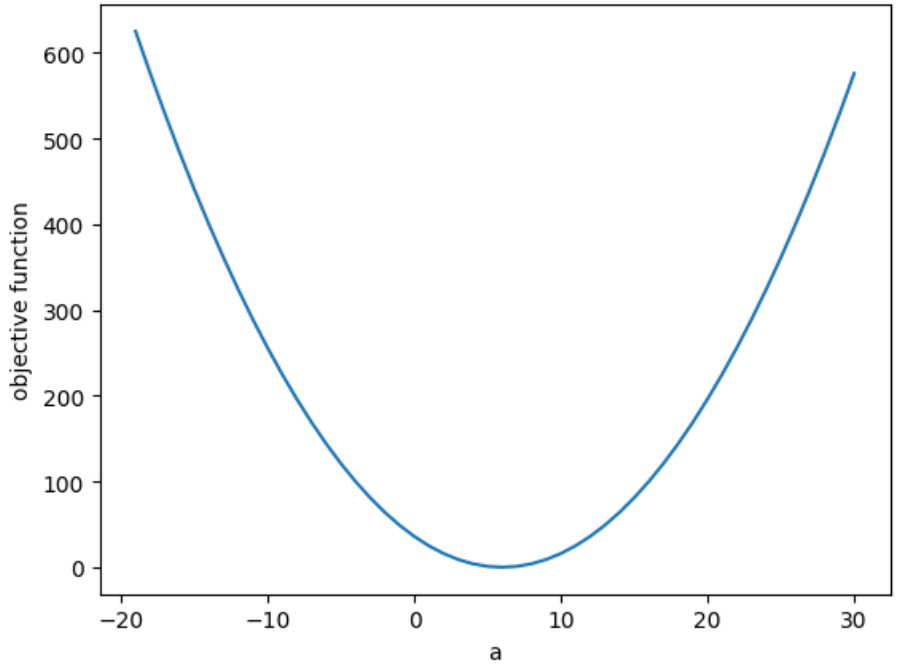
KeyConcepts

the algorithm: ~~Stochastic~~ Gradient Descent (SGD)

assume a simpler line model $y = ax$

($b = 0$) so we only need to find the "best" parameter a

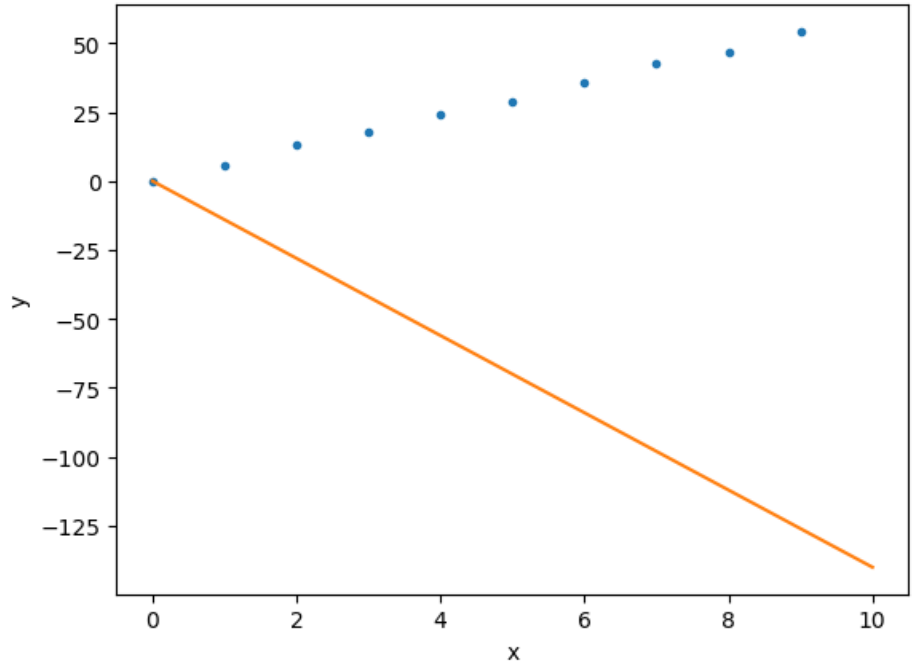


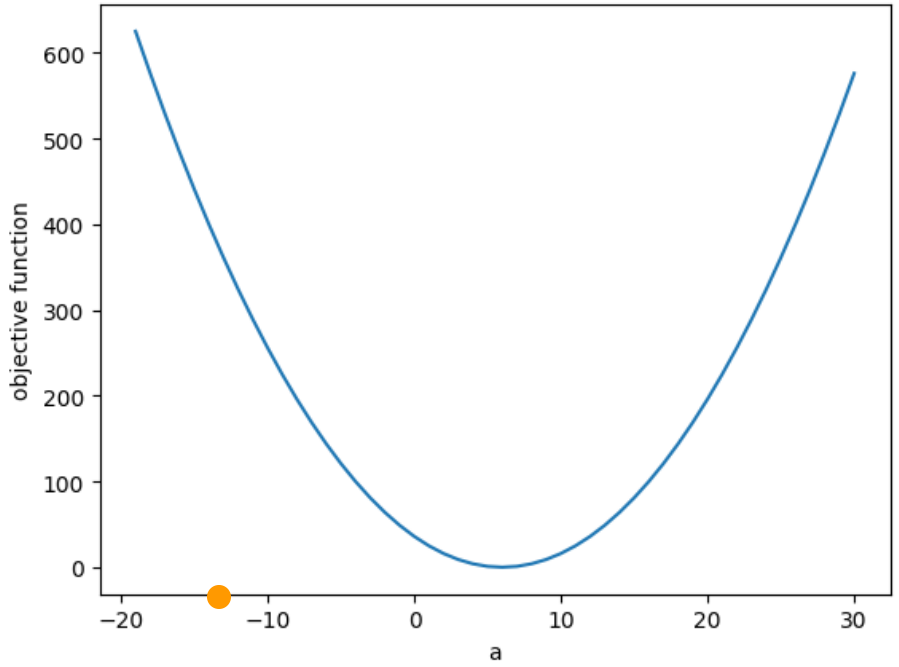


the algorithm: ~~Stochastic~~ Gradient Descent (SGD)

assume a simpler line model $y = ax$

($b = 0$) so we only need to find the "best" parameter a





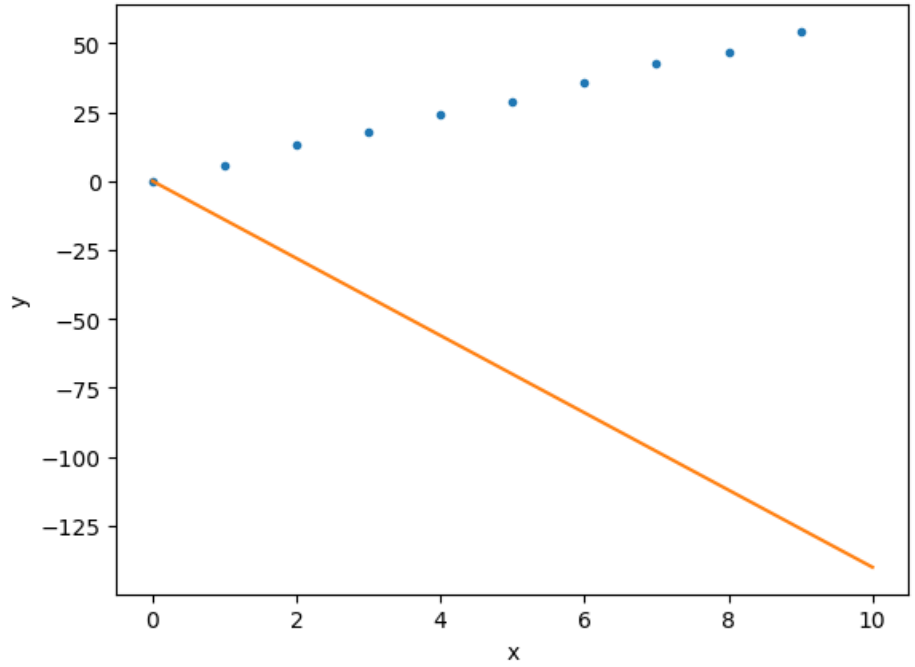
1. choose initial value for *a*

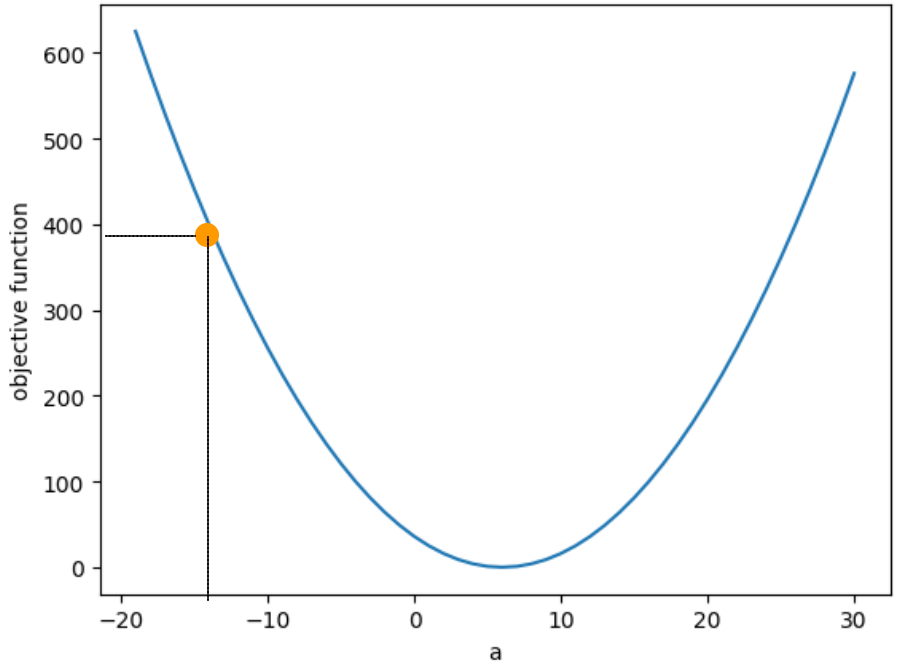
proposal a = -14

the algorithm: ~~Stochastic~~ Gradient Descent (SGD)

assume a simpler line model $y = ax$

($b = 0$) so we only need to find the "best" parameter a





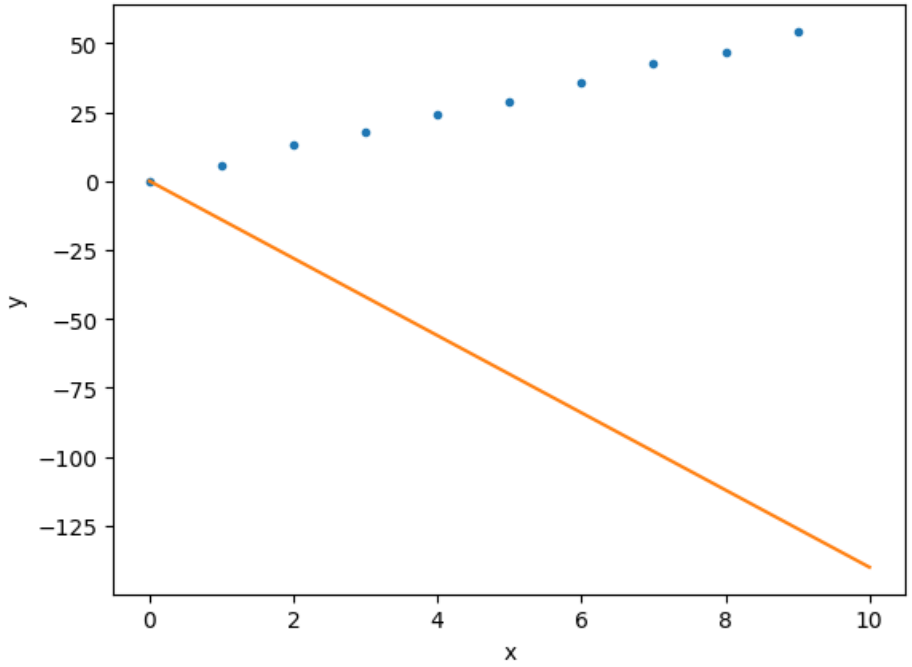
1. choose initial value for *a*
2. calculate objective function

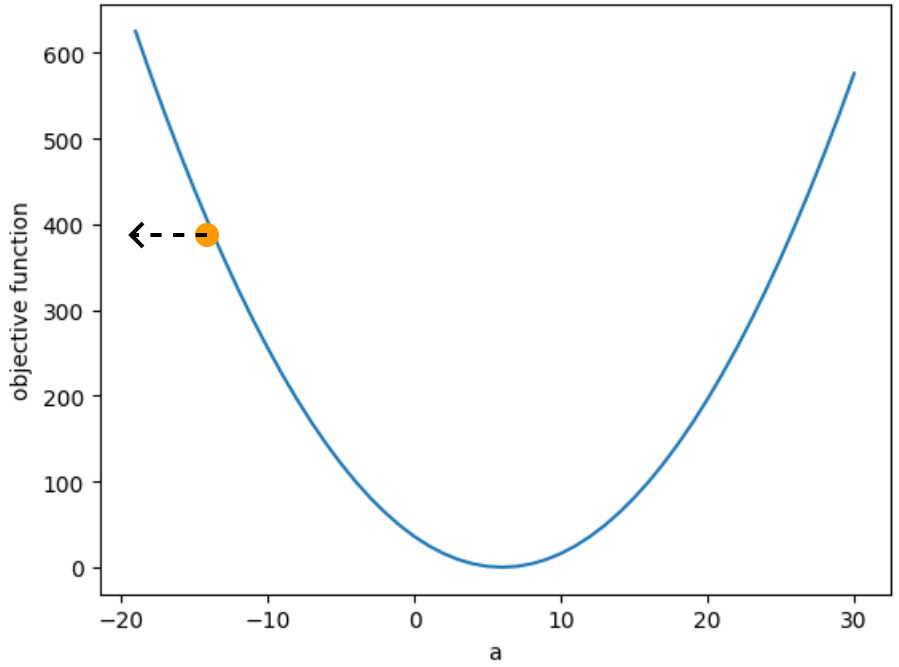
objective function value = 395

the algorithm: ~~Stochastic~~ Gradient Descent (SGD)

assume a simpler line model $y = ax$

($b = 0$) so we only need to find the "best" parameter a





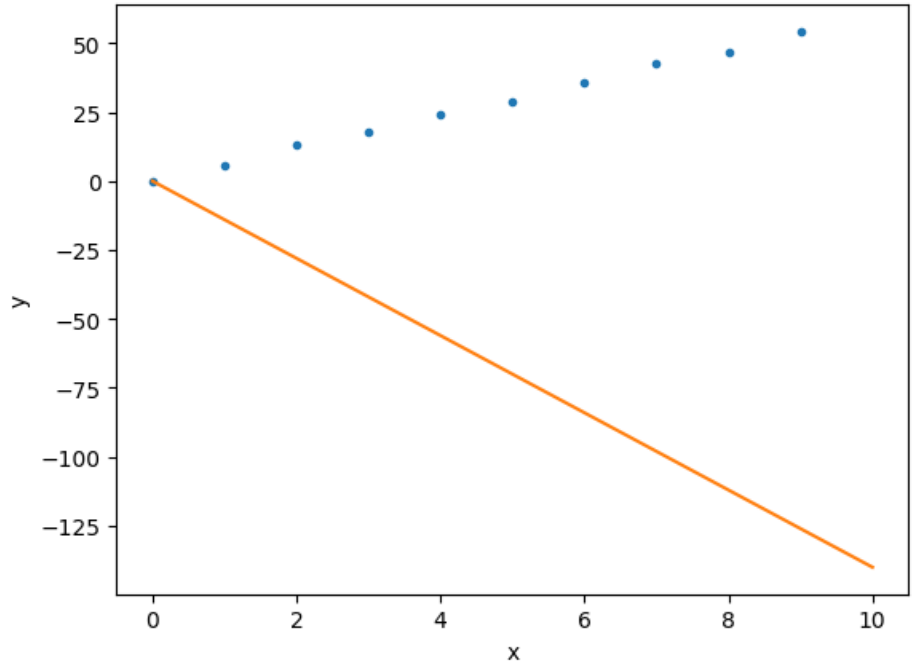
1. choose initial value for *a*
2. calculate objective function
3. find most promising direction to change *a*

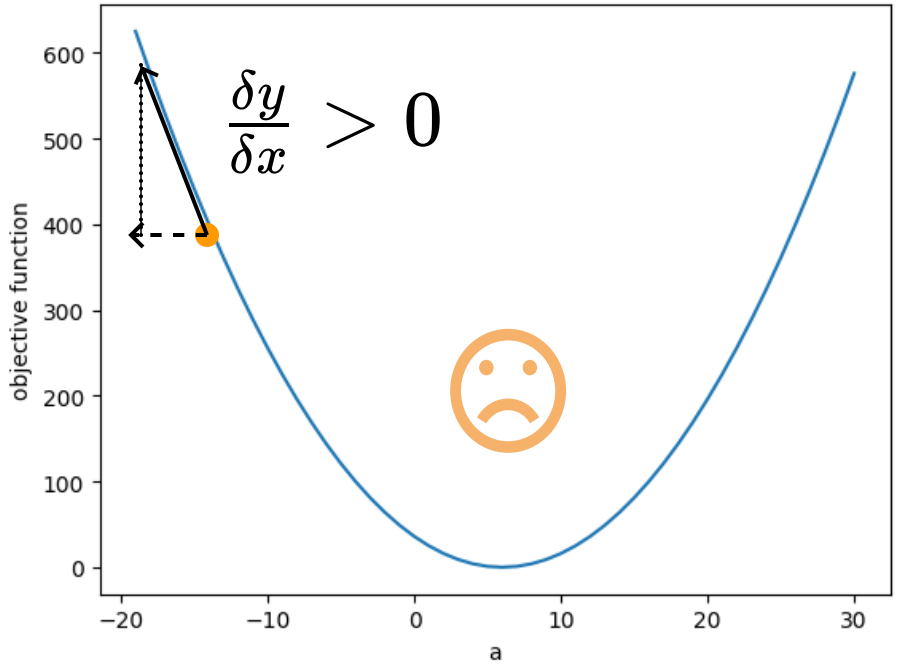
objective function value = 395

the algorithm: ~~Stochastic~~ Gradient Descent (SGD)

assume a simpler line model $y = ax$

($b = 0$) so we only need to find the "best" parameter a





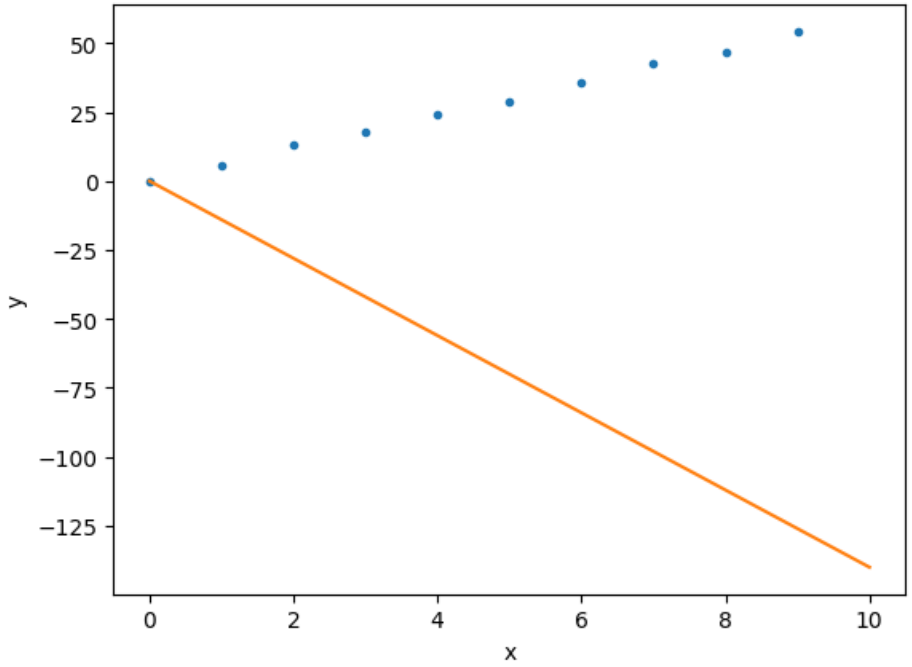
1. choose initial value for *a*
2. calculate objective function
3. find most promising direction to change *a*

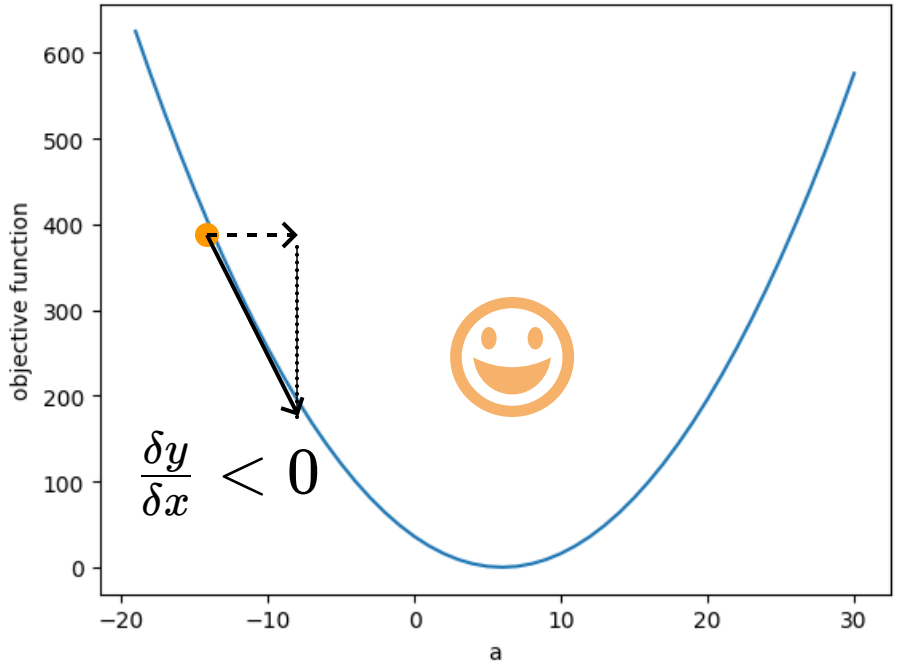
objective function value = 395

the algorithm: ~~Stochastic~~ Gradient Descent (SGD)

assume a simpler line model $y = ax$

($b = 0$) so we only need to find the "best" parameter a





1. choose initial value for *a*
2. calculate objective function
3. find most promising direction to change *a*

proposal a = -8

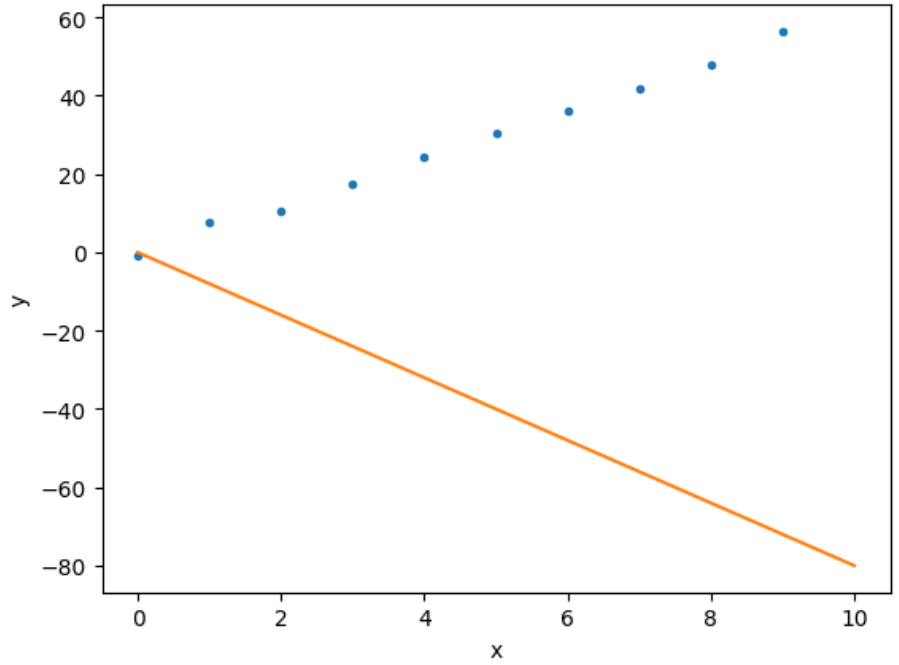
4. move in the best direction

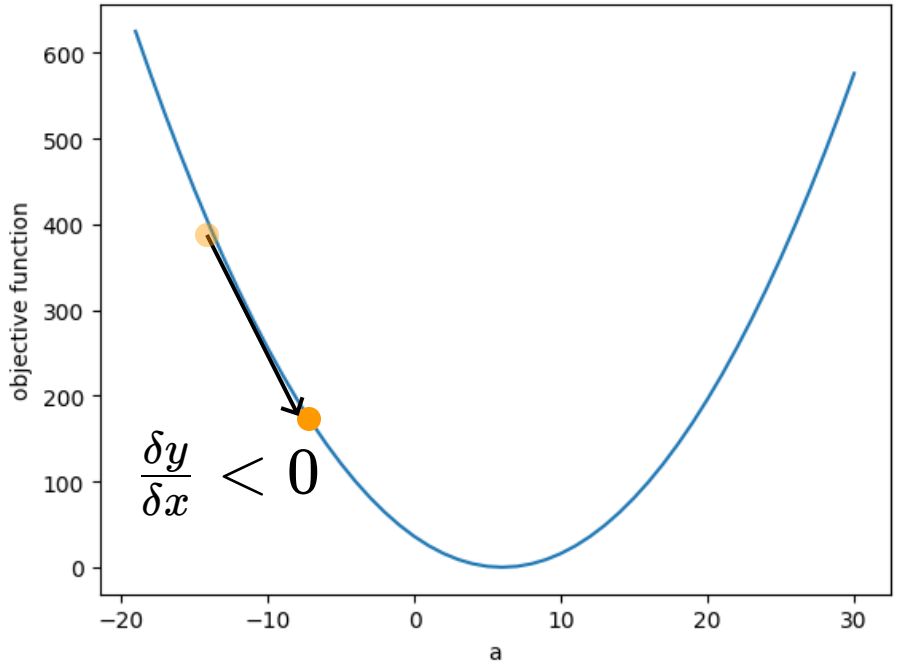
REPEAT 2,3,4

the algorithm: ~~Stochastic~~ Gradient Descent (SGD)

assume a simpler line model $y = ax$

($b = 0$) so we only need to find the "best" parameter a





1. choose initial value for *a*
2. calculate objective function
3. find most promising direction to change *a*

proposal a = -8

o.f. = 180

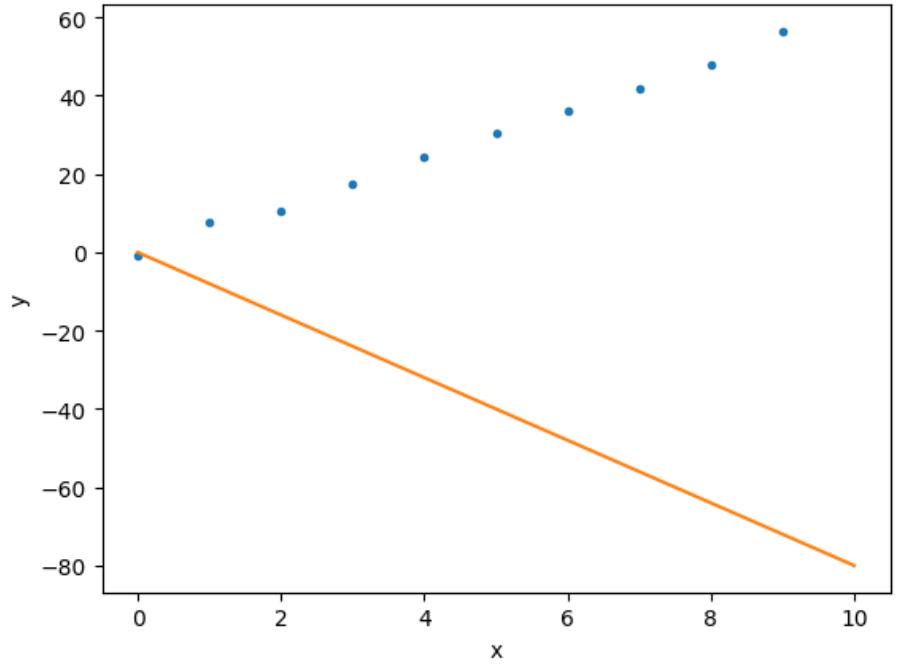
4. move in the best direction

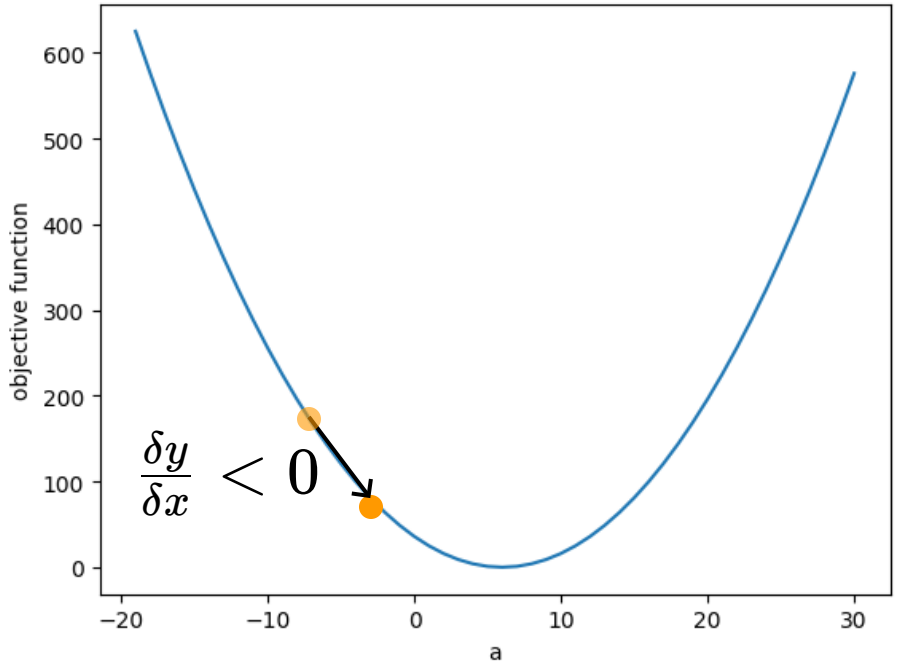
REPEAT 2,3,4

the algorithm: ~~Stochastic~~ Gradient Descent (SGD)

assume a simpler line model $y = ax$

($b = 0$) so we only need to find the "best" parameter a





1. choose initial value for *a*
2. calculate objective function
3. find most promising direction to change *a*

proposal a = -2

o.f. = 70

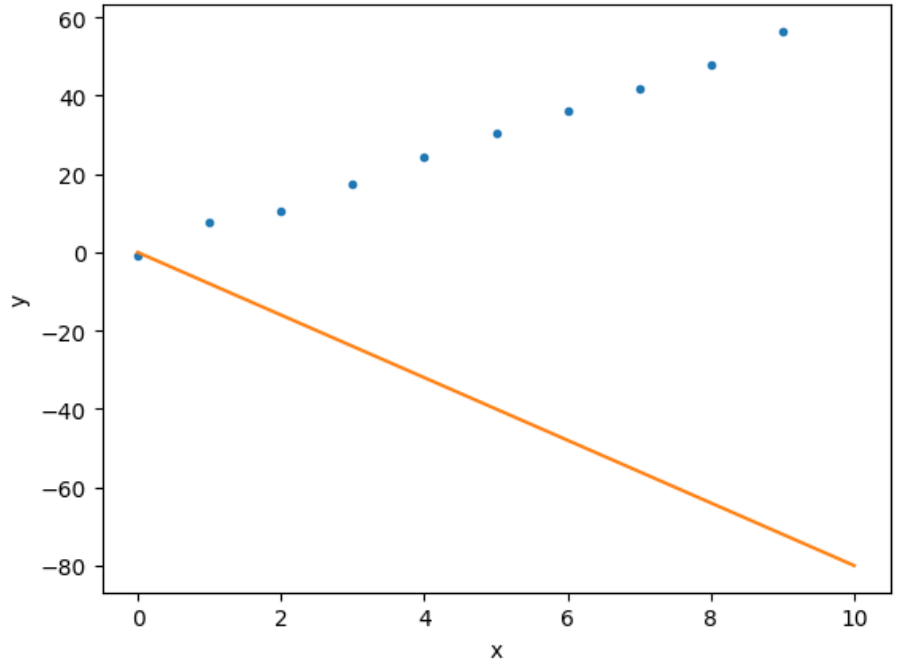
4. move in the best direction

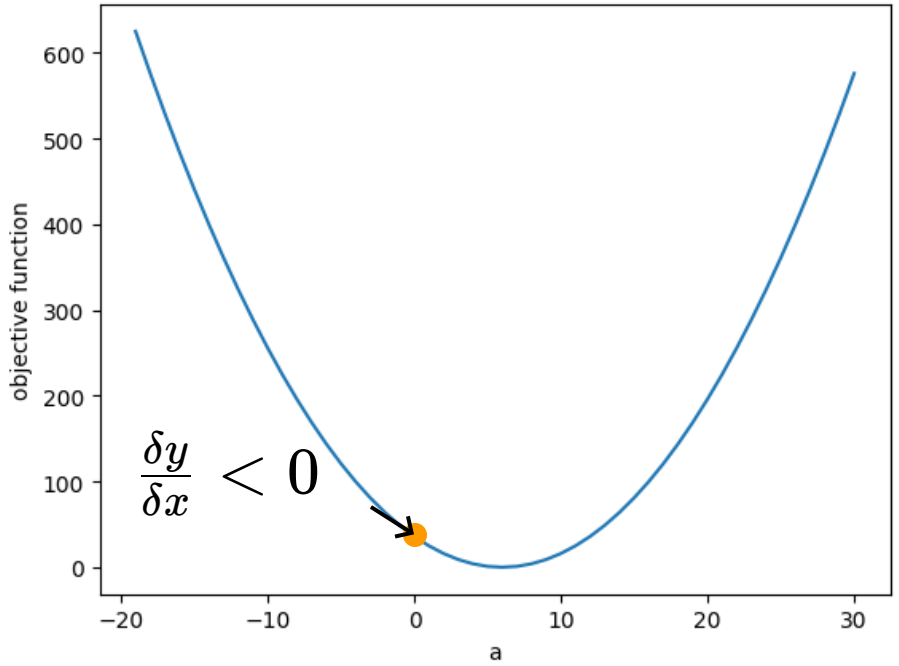
REPEAT 2,3,4

the algorithm: ~~Stochastic~~ Gradient Descent (SGD)

assume a simpler line model $y = ax$

($b = 0$) so we only need to find the "best" parameter a





1. choose initial value for *a*
2. calculate objective function
3. find most promising direction to change *a*

proposal a = 0

o.f. = 30

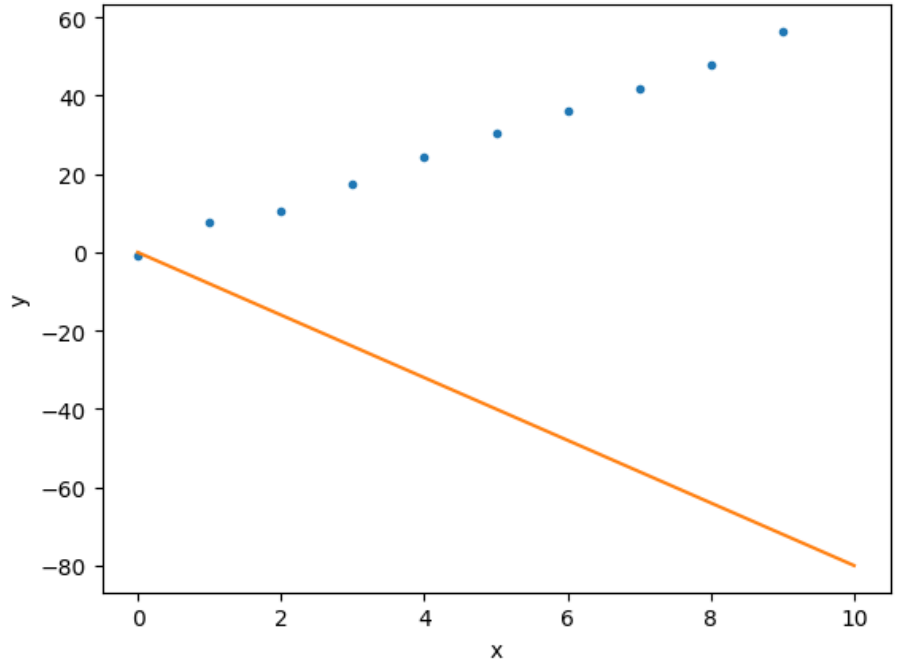
4. move in the best direction

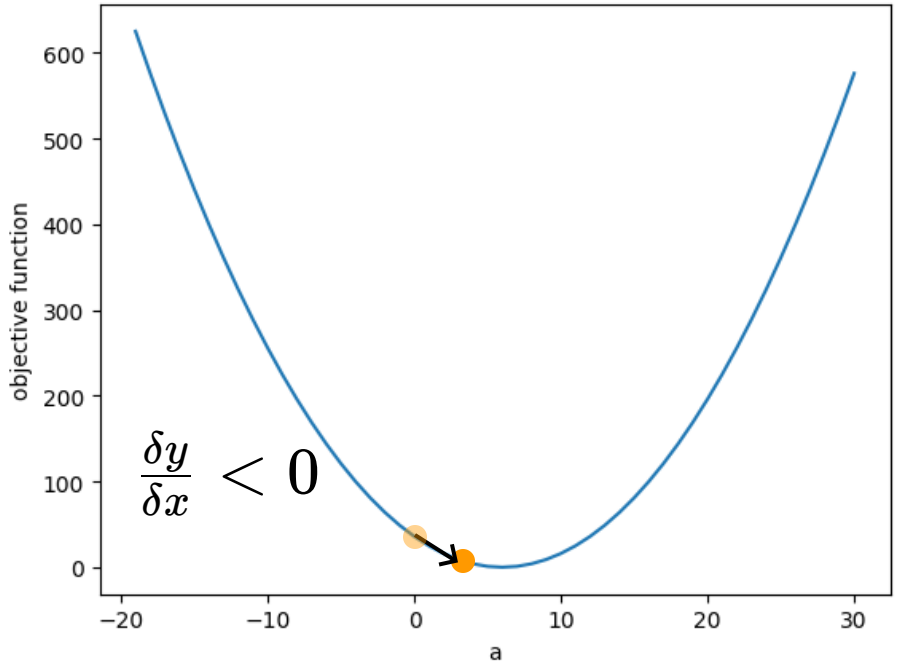
REPEAT 2,3,4

the algorithm: ~~Stochastic~~ Gradient Descent (SGD)

assume a simpler line model $y = ax$

($b = 0$) so we only need to find the "best" parameter a





1. choose initial value for *a*
2. calculate objective function
3. find most promising direction to change *a*

proposal a = 3

o.f. = 10

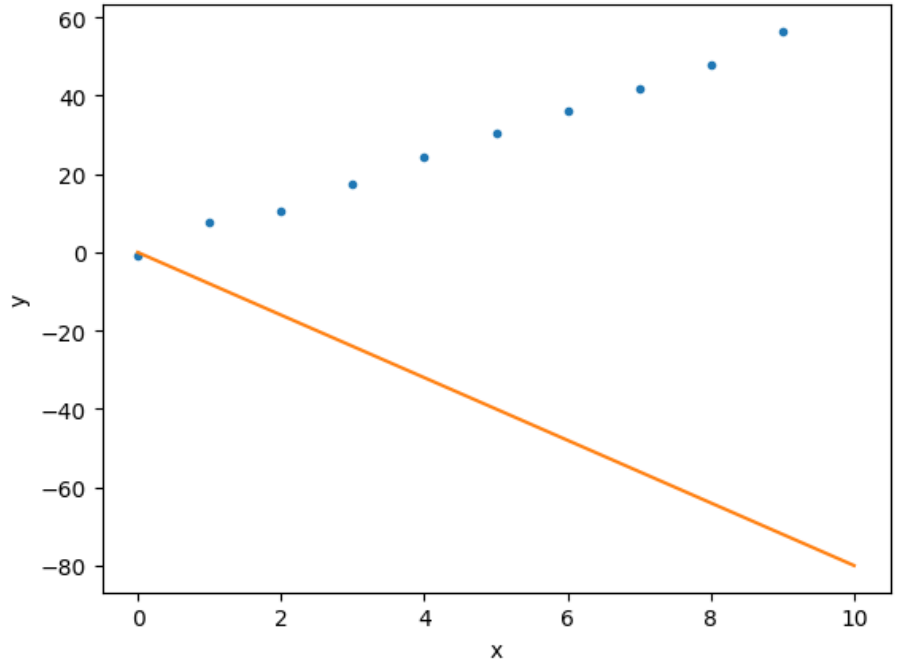
4. move in the best direction

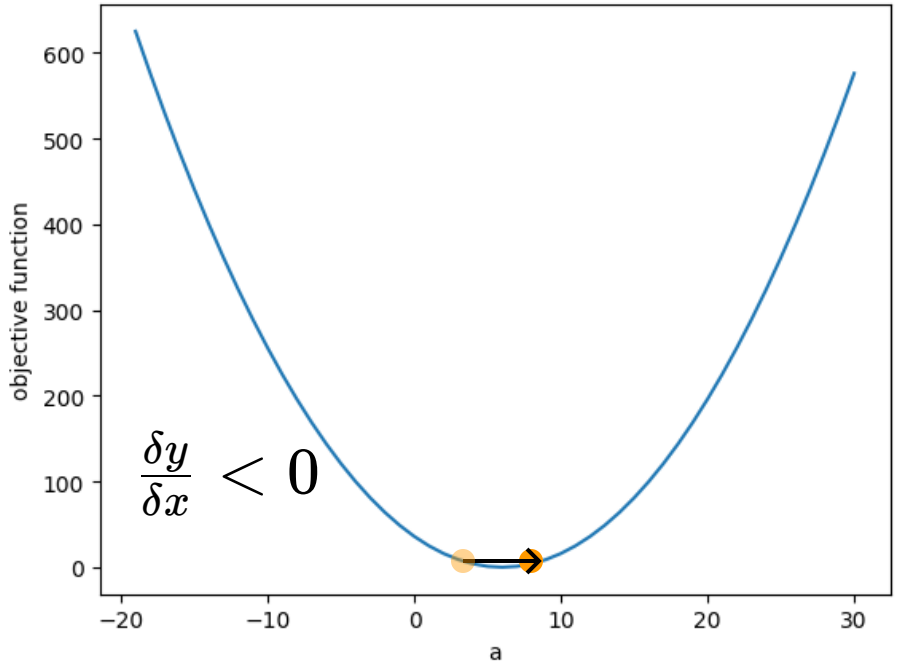
REPEAT 2,3,4

the algorithm: ~~Stochastic~~ Gradient Descent (SGD)

assume a simpler line model $y = ax$

($b = 0$) so we only need to find the "best" parameter a





1. choose initial value for *a*
2. calculate objective function
3. find most promising direction to change *a*

proposal a = 8

o.f. = 9

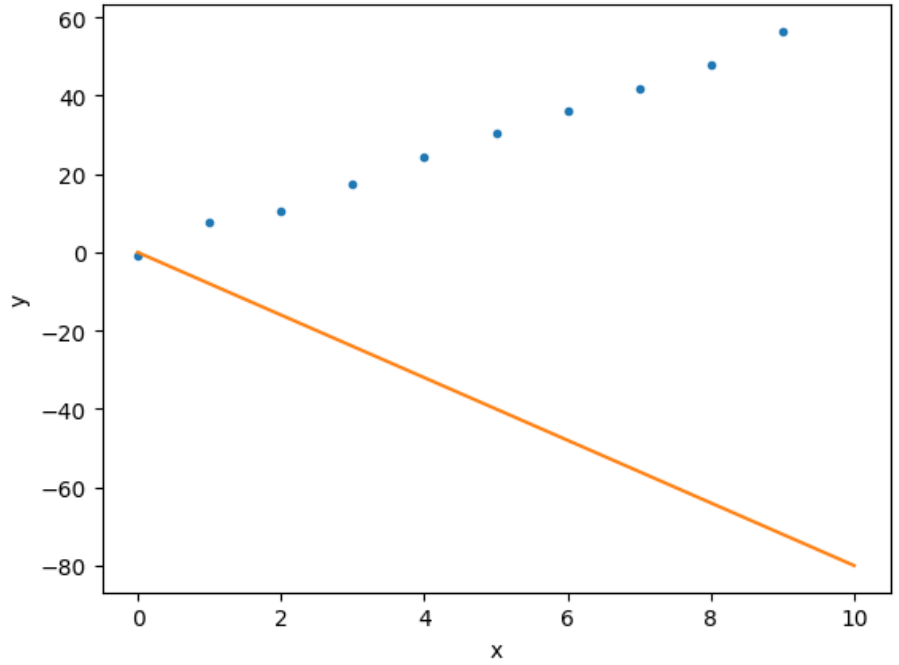
4. move in the best direction

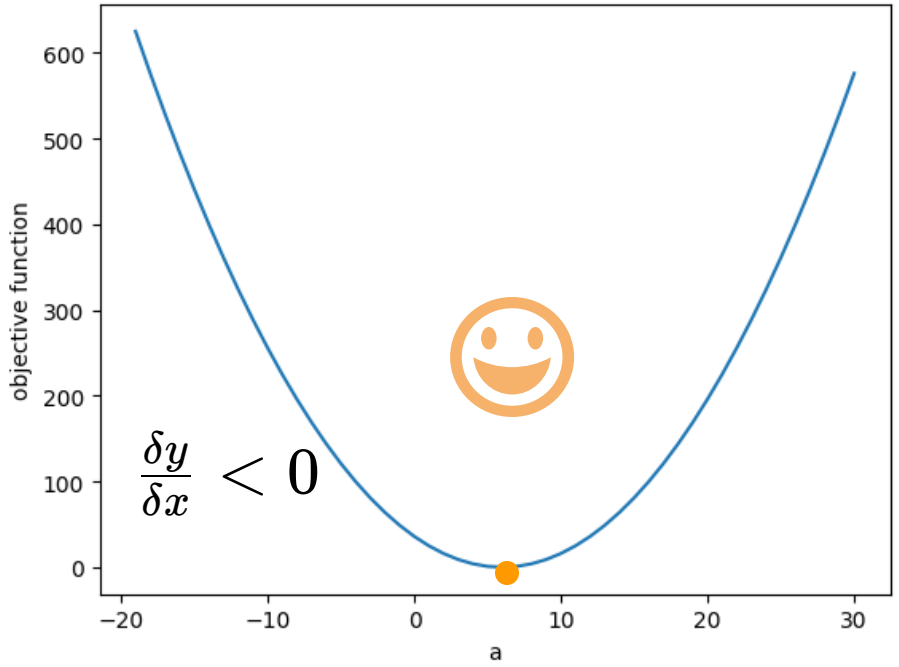
REPEAT 2,3,4

the algorithm: ~~Stochastic~~ Gradient Descent (SGD)

assume a simpler line model $y = ax$

($b = 0$) so we only need to find the "best" parameter a





1. choose initial value for *a*
2. calculate objective function
3. find most promising direction to change *a*

proposal a = 6

o.f. = 0

4. move in the best direction

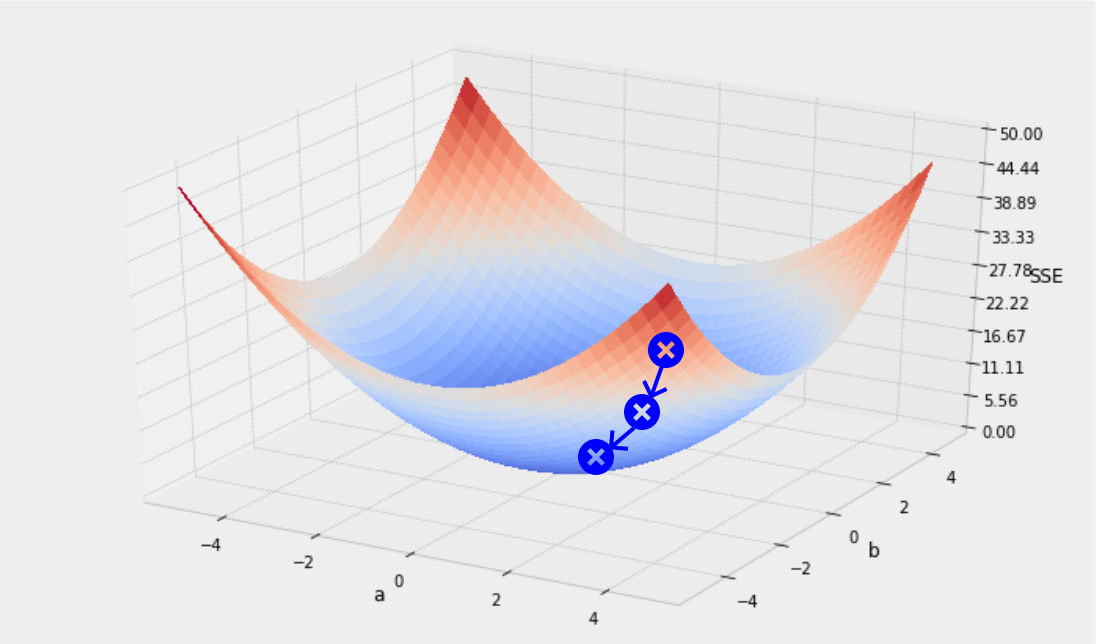
REPEAT 2,3,4

the algorithm: ~~Stochastic~~ Gradient Descent

for a line model $y = ax + b$

we need to find the "best" parameters a and b

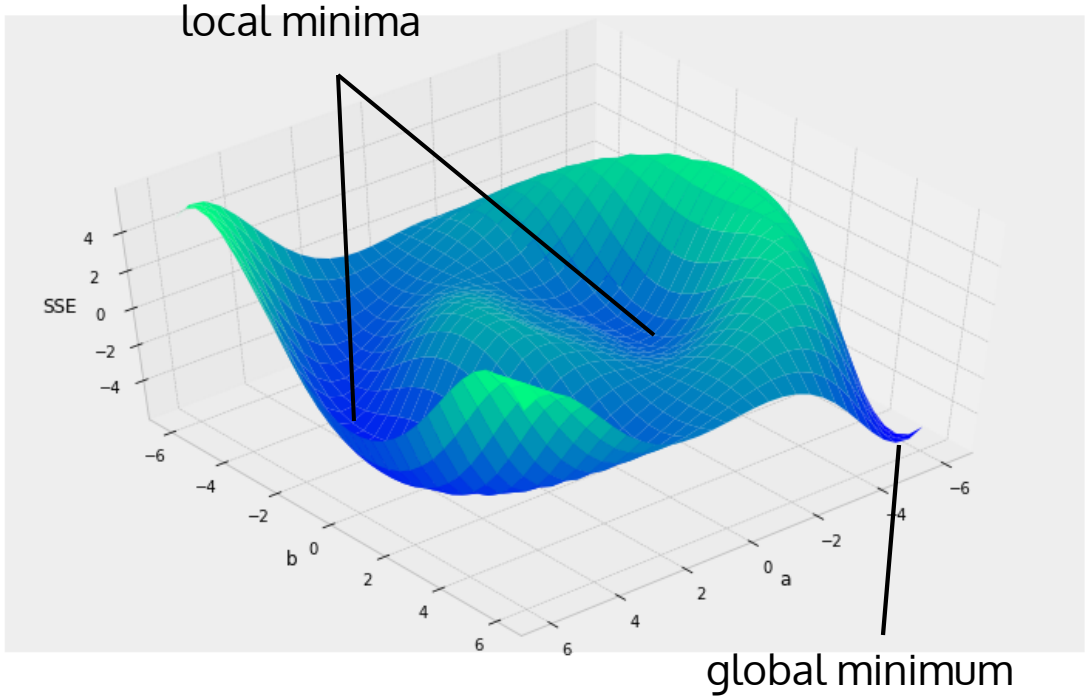
1. choose initial value for a & b
2. calculate the SSE
3. calculate best direction to go to decrease the SSE
4. step in that direction
5. go back to step 2 and repeat



the algorithm: ~~Stochastic~~ Gradient Descent

DANGER!!

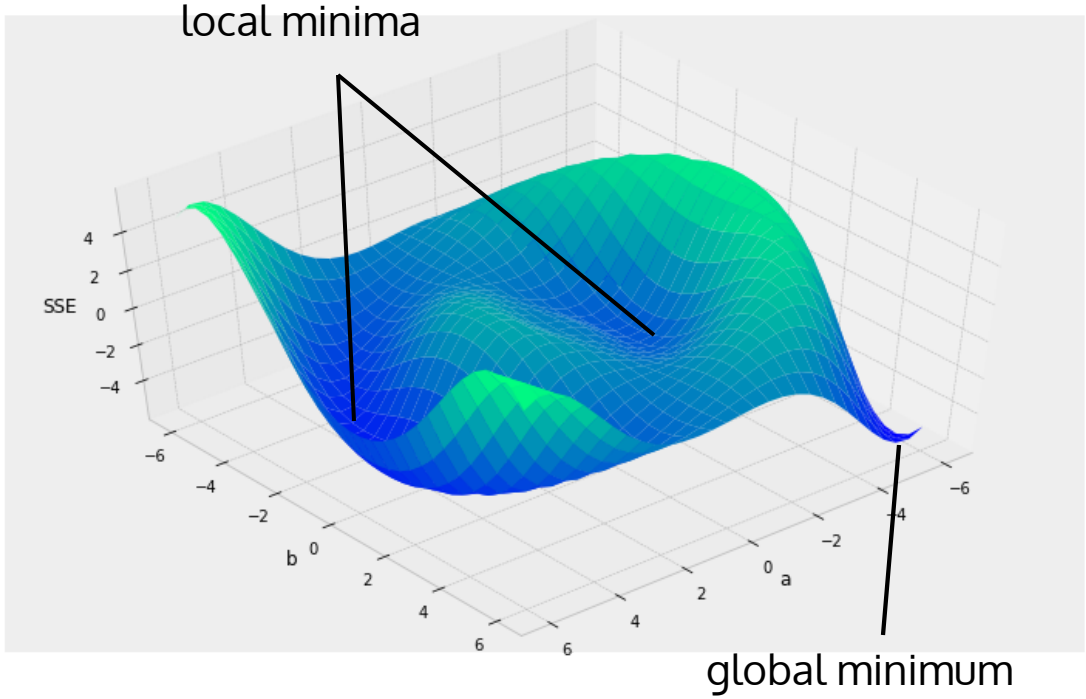
- local vs. global minima



the algorithm: ~~Stochastic~~ Gradient Descent

DANGER!!

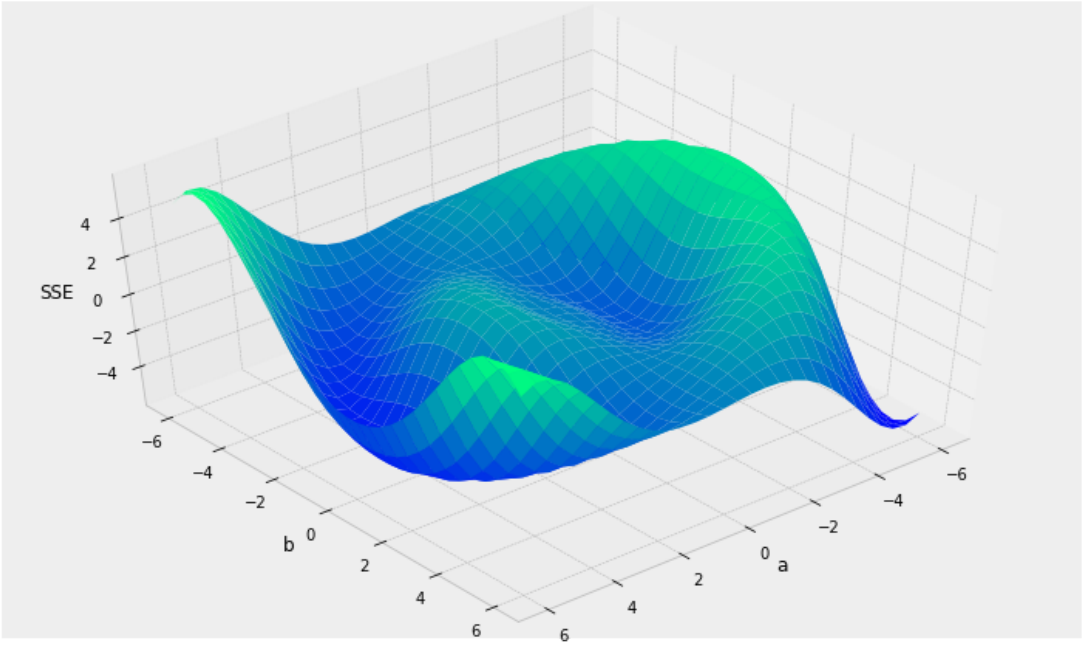
- local vs. global minima



the algorithm: **Stochastic Gradient Descent**

Things to consider:

- local vs. global minima
- initialization: choosing starting spot?
- learning rate: how far to step?
- stopping criterion: when to stop?



<u>Stochastic</u> Gradient Descent (SGD): use a different (random) sub-sample of the data at each iteration

train, test, and validate

validating a model

How do we measure if a model is good?

- Accuracy
- Precision
- Recall
- ROC
- AOC

We will talk more about this later...

but for now focus on

regression performance metrics

validating a model

How do we measure if a model is good? $\epsilon_i = y_i - f(t_i)$

Accuracy

Precision

Recall

ROC

AOC

We will talk more about this later...

but for now focus on

regression performance metrics

$$AE = \sum_i |\epsilon_i|$$ **Absolute error**

$$SE = \sum_i \epsilon_i^2$$ **Squared error**

$$MSE = \frac{1}{N} SE$$ **Mean squared error**

$$RMSE = \sqrt{MSE}$$ **Root mean squared error**

$$rMSE = \frac{MSE}{\sigma^2}$$ **Relative mean squared error**

$$R^2 = 1 - rMSE$$ **R squared**

validating a model

How do we measure if a model is good? $\epsilon_i = y_i - f(t_i)$

Accuracy

Precision

Recall

ROC

AOC

We will talk more about this later...

but for now focus on

regression performance metrics

do you recognize these??

$$AE = \sum_i |\epsilon_i|$$ **Absolute error**

$$SE = \sum_i \epsilon_i^2$$ **Squared error**

$$MSE = \frac{1}{N} SE$$ **Mean squared error**

$$RMSE = \sqrt{MSE}$$ **Root mean squared error**

$$rMSE = \frac{MSE}{\sigma^2}$$ **Relative mean squared error**

$$R^2 = 1 - rMSE$$ **R squared**

validating a model

How do we measure if a model is good? $\epsilon_i = y_i - f(t_i)$

Accuracy

Precision

Recall

ROC

AOC

$$AE = \sum_i |\epsilon_i| \equiv L_1$$ **Absolute error**

$$SE = \sum_i \epsilon_i^2 \equiv L_2$$ **Squared error**

$$MSE = \frac{1}{N} SE$$ **Mean squared error**

$$RMSE = \sqrt{MSE}$$ **Root mean squared error**

$$rMSE = \frac{MSE}{\sigma^2}$$ **Relative mean squared error**

$$R^2 = 1 - rMSE$$ **R squared**

We will talk more about this later...

but for now focus on

regression performance metrics

validating a model

How do we measure if a model is good? $\epsilon_i = y_i - f(t_i)$

Accuracy

Precision

Recall

ROC

AOC

We will talk more about this later...

but for now focus on

regression performance metrics $R^2 = 1 - rMSE$

Split the sample in test and training sets

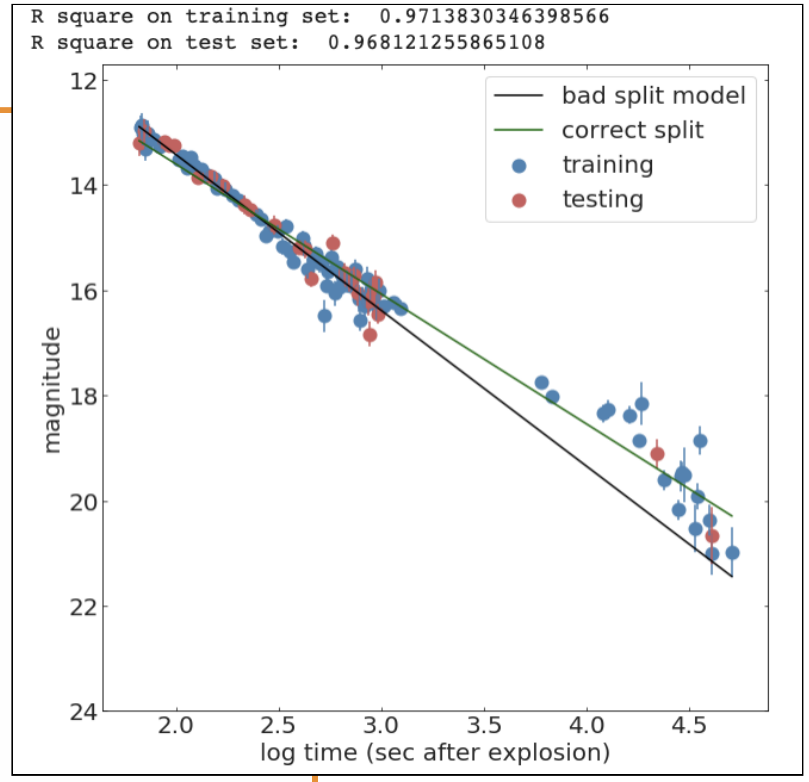
Train on the training set

Test (measure accuracy) on the test set

validating a model

```
from sklearn.model_selection import train_test_split

def line(x, intercept, slope):
    return slope * x + intercept

def chi2(args, x, y, s):
    a, b = args
    return sum((y - line(x, a, b))**2 / s)

x_train, x_test, y_train, y_test, s_train, s_test = train_test_split(
     x, y, s, test_size=0.25, random_state=42)

initialGuess = (10, 1)

chi2Solution_goodsplit = minimize(chi2, initialGuess,
        args=(x_train, y_train, s_train))

print("best fit parameters from the minimization of the chi squared: " +
       "slope {:.2f}, intercept {:.2f}".format(*chi2Solution_goodsplit.x))

print("R square on training set: ", Rsquare(chi2Solution_goodsplit.x, x_train, y_train))
print("R square on test set: ", Rsquare(chi2Solution_goodsplit.x, x_test, y_test))
```

```
R square on training set:  0.9713830346398566
R square on test set:  0.968121255865108
```



In ML models need to be "validated":

1. split the data into a training and a test set (typical split 70/30).
2. learn the model parameters by "training" the model on the training set
3. "test" the model on the test set: measure the accuracy of the prediction (e.g. as the distance between the prediction and the test data).

The performance on the model is the performance achieved on the test set.

a significance performance degradation on the test compared to training set indicates that the model is "overtrained" and does not generalize well.

An upgrade on this workflow is to create a training, a test, and a validation test. Iterate between training and test to achieve optimal performance, then measure accuracy on the validation set.This is because you can use the test set performance to tune the model hyperparameters (model selection) but then you would report a performance that is tuned on the test set.