ML for natural and physical scientists 2023 3

models - linear regression - uncertainties

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

this slide deck:

https://slides.com/federicabianco/mlpns23_3

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

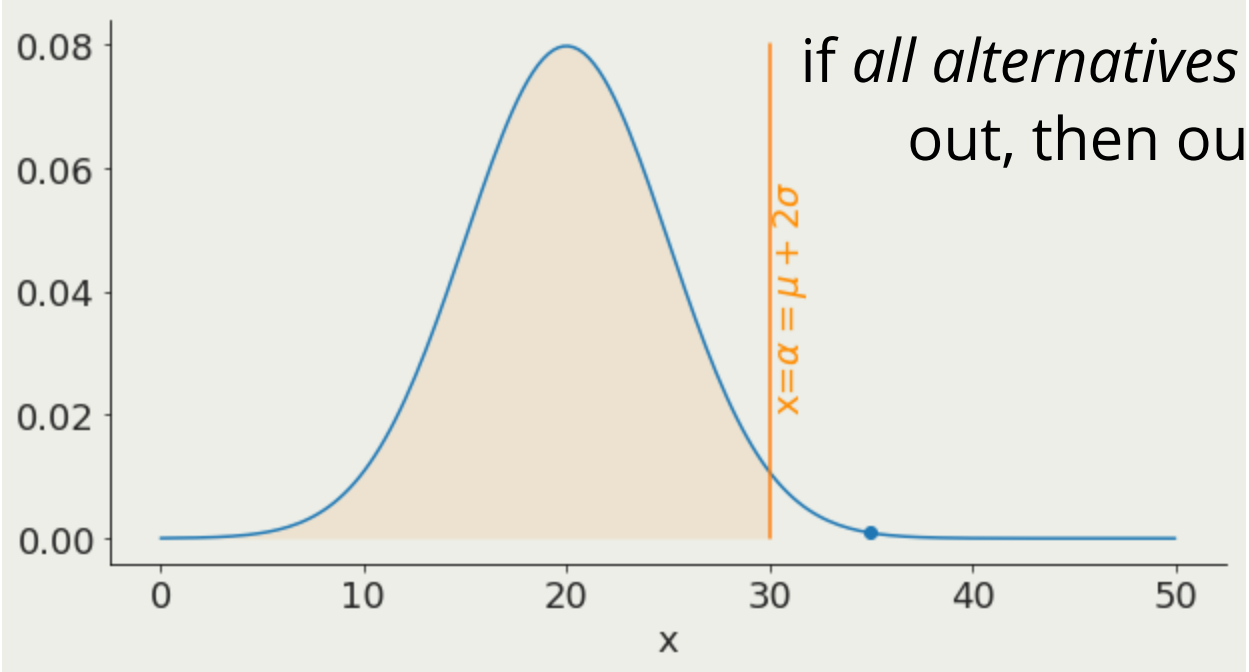
calculate probability of value obtained for the pivotal quantity under the Null

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

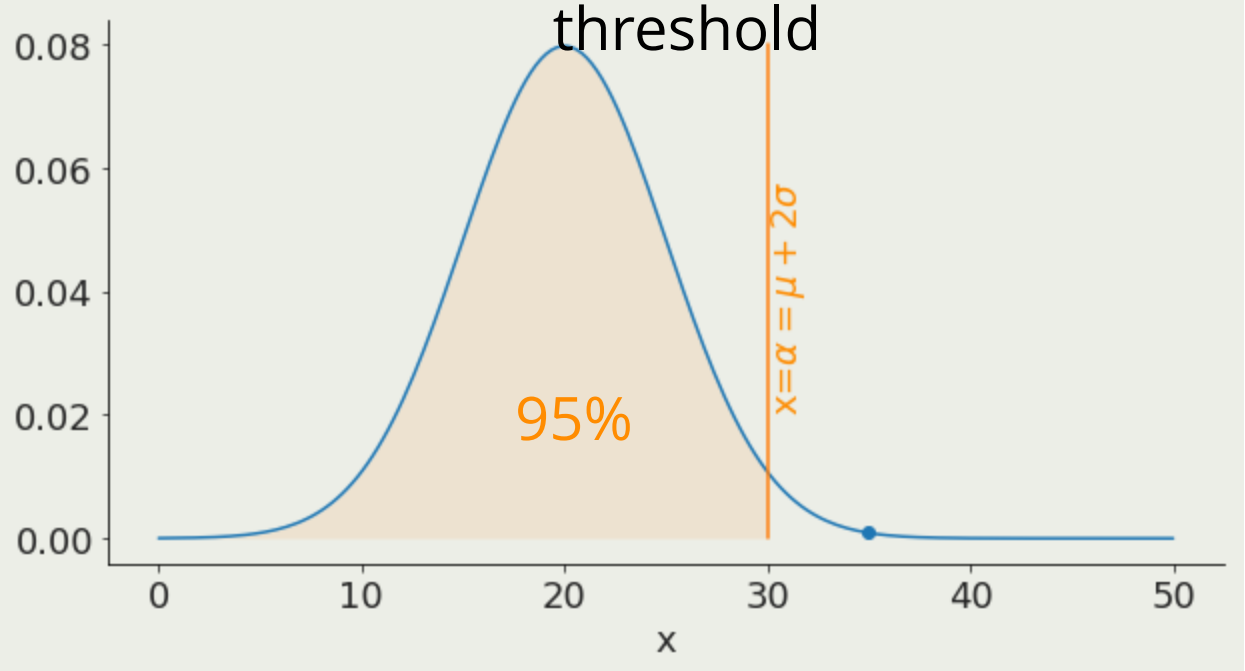
identify all alternative outcomes

Alternative Hypothesis

Null
Hypothesis
Rejection
Testing

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 what is machine learning

what is machine learning?

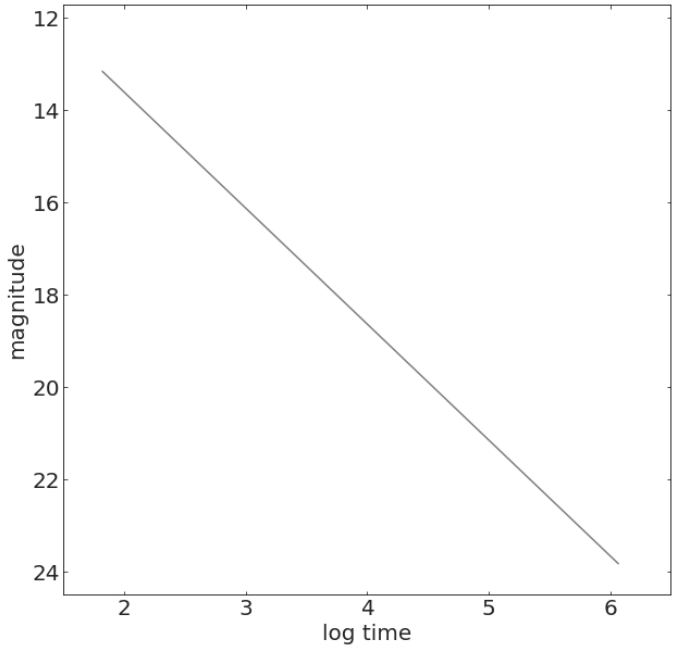
[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.

Arthur Samuel, 1959

what is 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

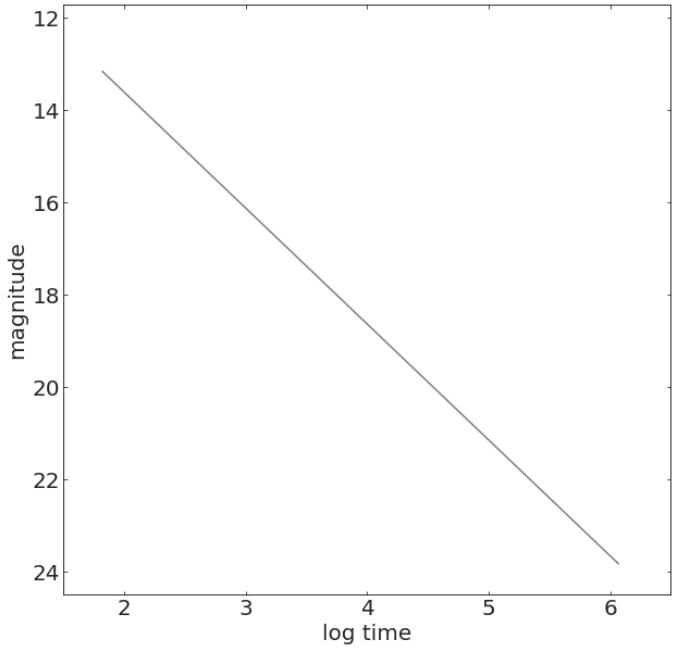
model $y = ax + b$

parameters: slope (a), intercept (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)

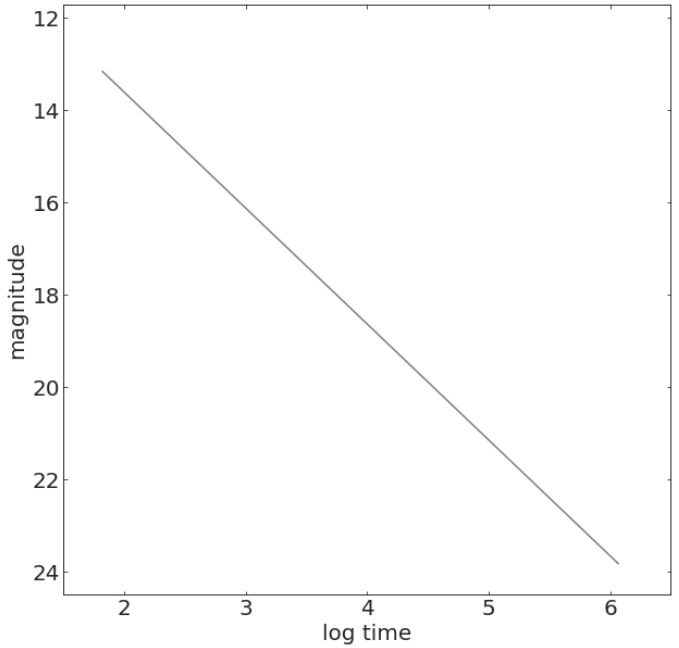
$$y = ax + b$$

model 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

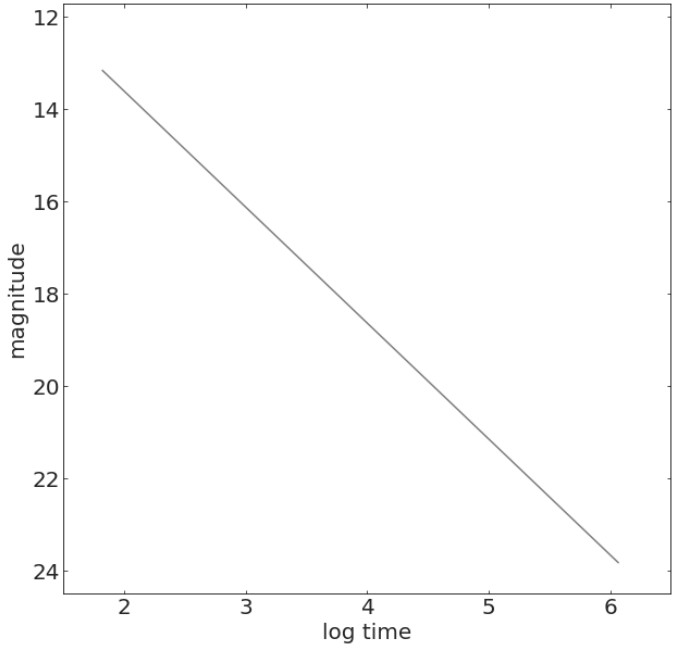
$$y = a_1 x_1 + a_1 x_2 + b$$

model 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

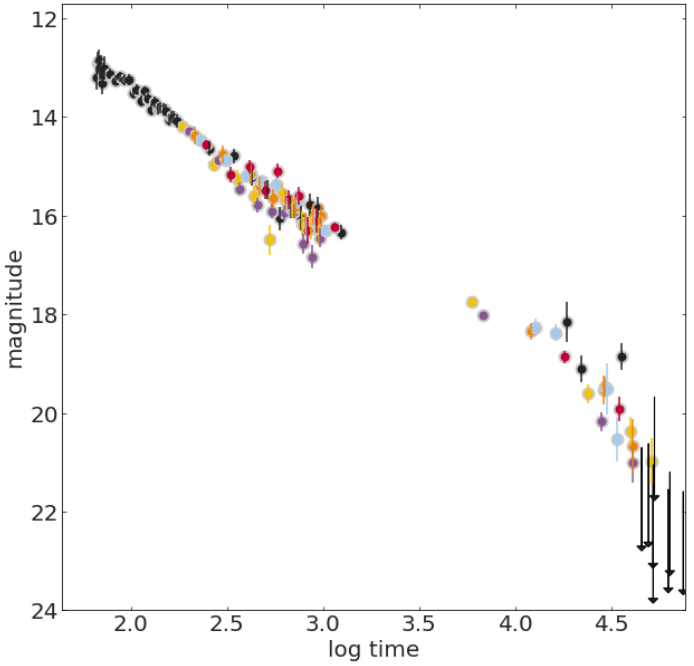


Model:

a mathematical formula with parameters

Data:

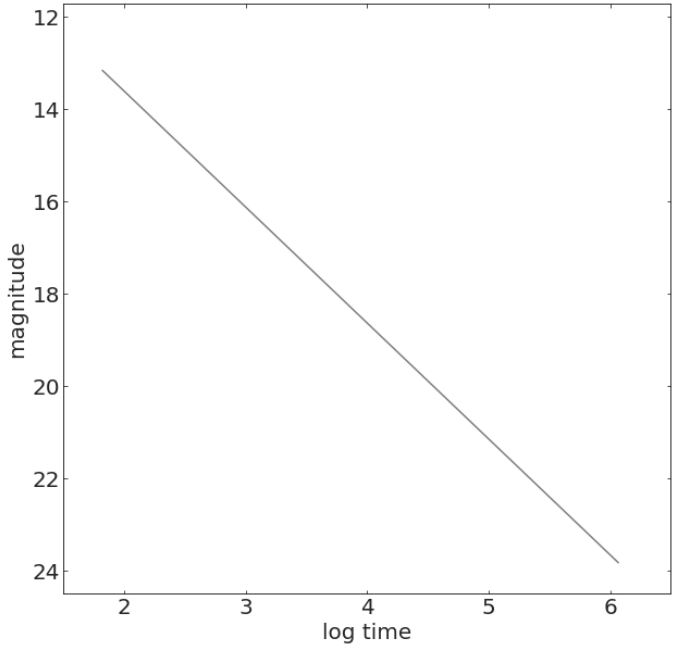
a set of observations



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

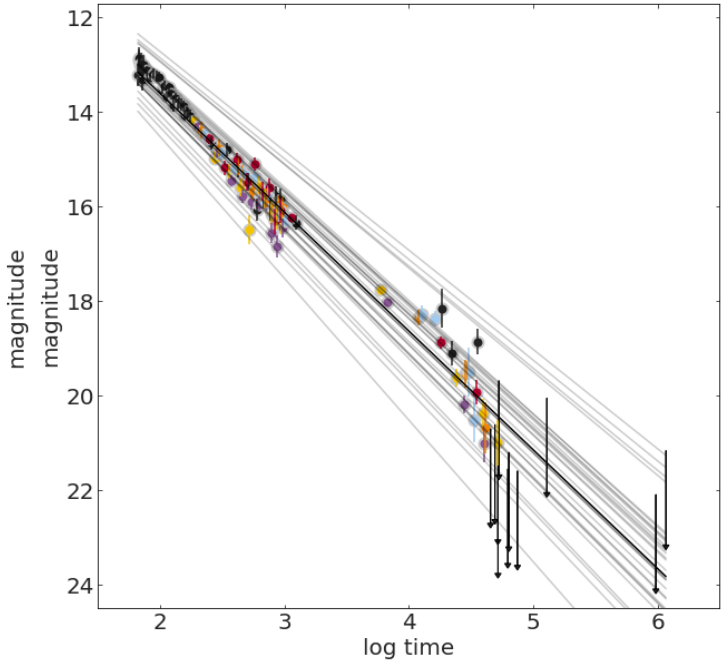


Model:

a mathematical formula with parameters

Data:

a set of observations

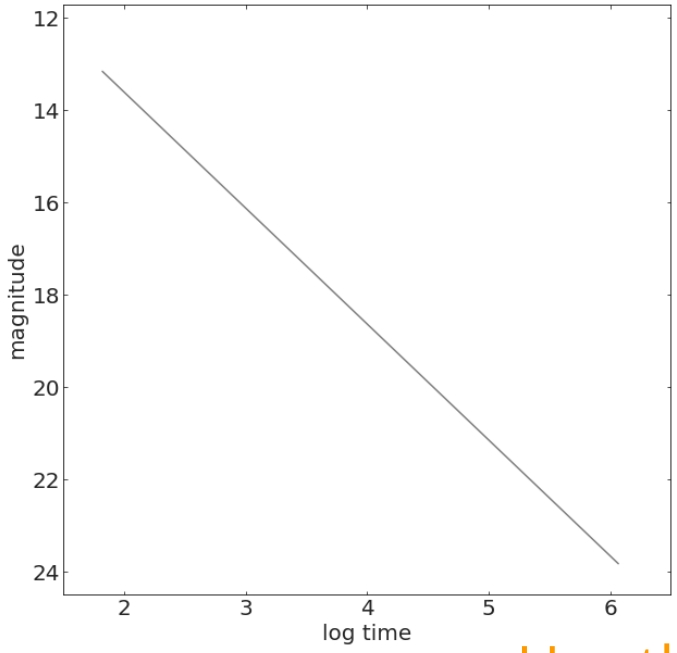


for every parameter there are an infinity of models

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

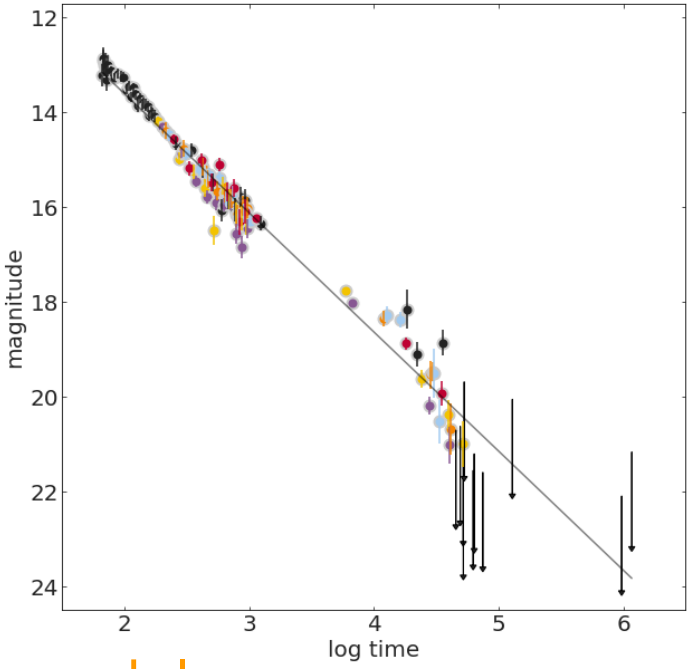


Model:

a mathematical formula with parameters

Data:

a set of observations



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

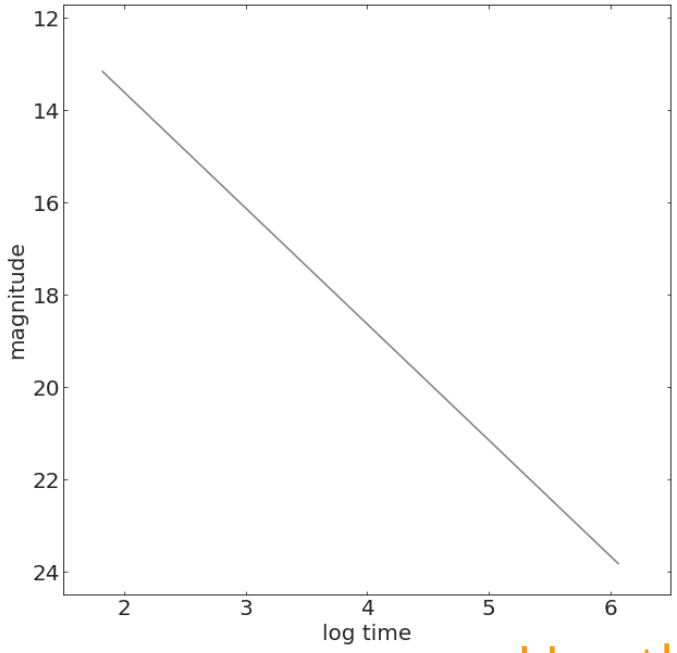
what is machine learning?

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

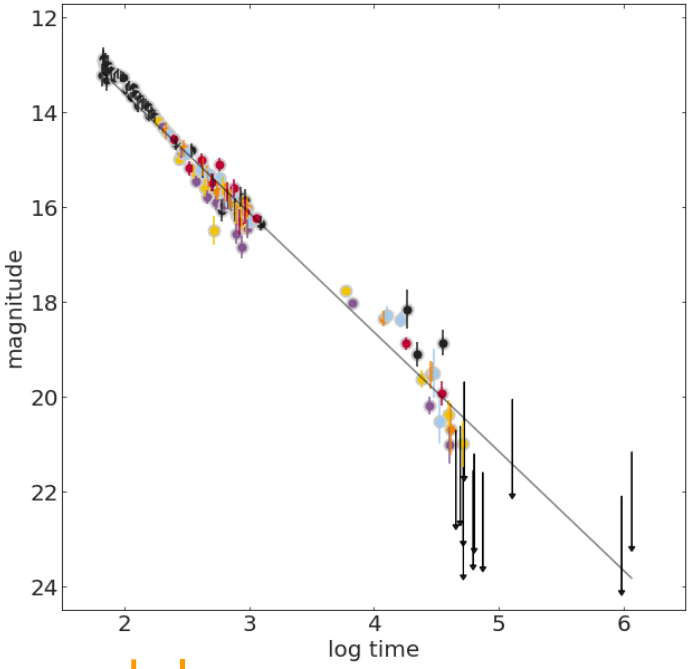
Machine Learning is the disciplines that conceptualizes, studies, and applies those models.

Key Concept

the best way to think about it *in the ML context:*



a model is a low dimensional representation of a higher dimensionality dataset



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

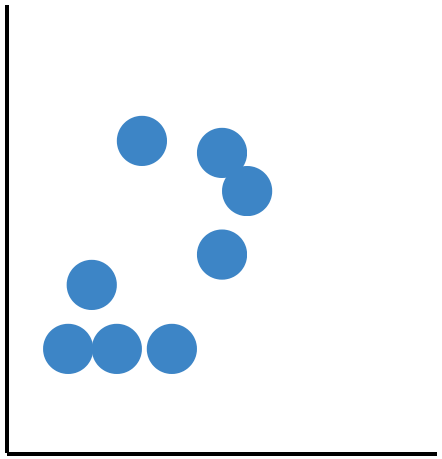
1.2 what is machine learning

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)

unsupervised vs supervised learning



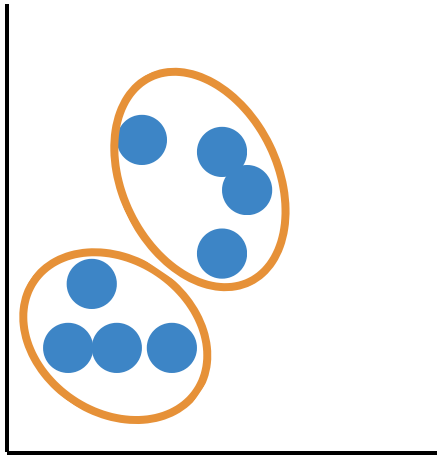
Clustering

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

unsupervised vs supervised learning

Unsupervised learning

- understanding structure
- anomaly detection
- dimensionality reduction



Clustering

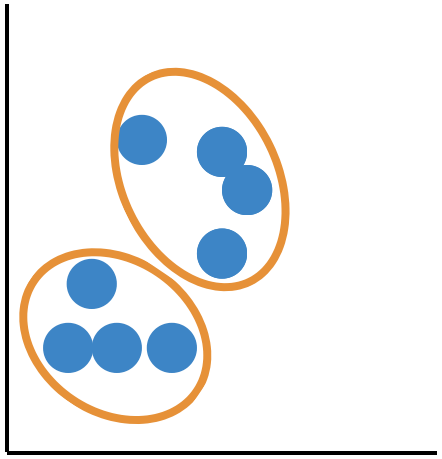
partitioning the feature space so that the existing data is grouped (according to some target function!)

All features are observed for all datapoints

unsupervised vs supervised learning

Unsupervised learning

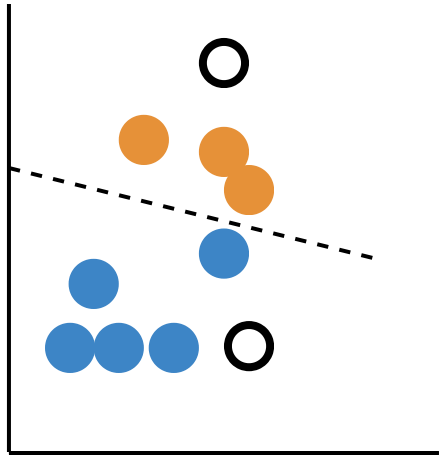
- understanding structure
- anomaly detection
- dimensionality reduction



Clustering

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

All features are observed for all datapoints



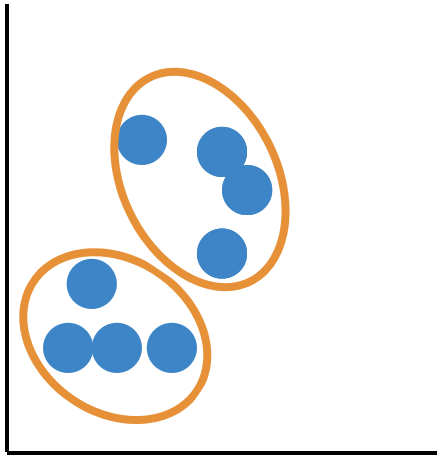
Classifying & regression

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

unsupervised vs supervised learning

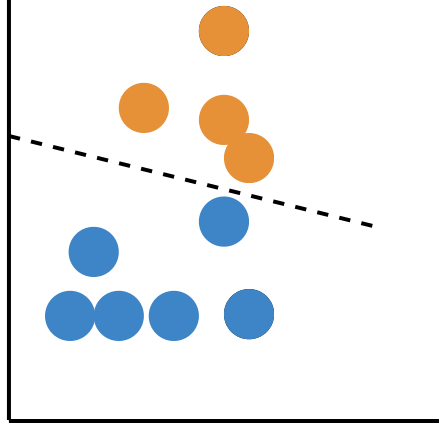
Unsupervised learning

- understanding structure
- anomaly detection
- dimensionality reduction



Clustering

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



Classifying & regression

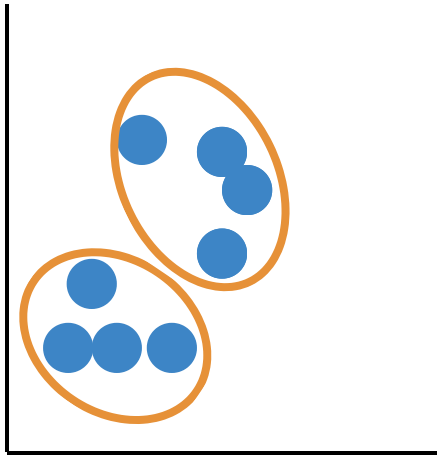
finding functions of the variables that allow to predict unobserved properties of new observations

All features are observed for all datapoints

unsupervised vs supervised learning

Unsupervised learning

- understanding structure
- anomaly detection
- dimensionality reduction



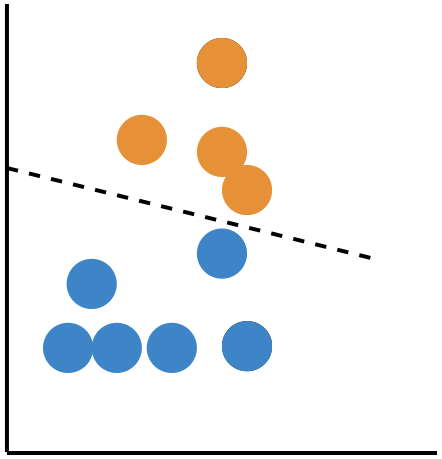
Clustering

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

All features are observed for all datapoints

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.

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.

13

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

We will talk more about this later...

but for now focus on

regression performance metrics

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

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

$$R^2 = 1 - rMSE$$

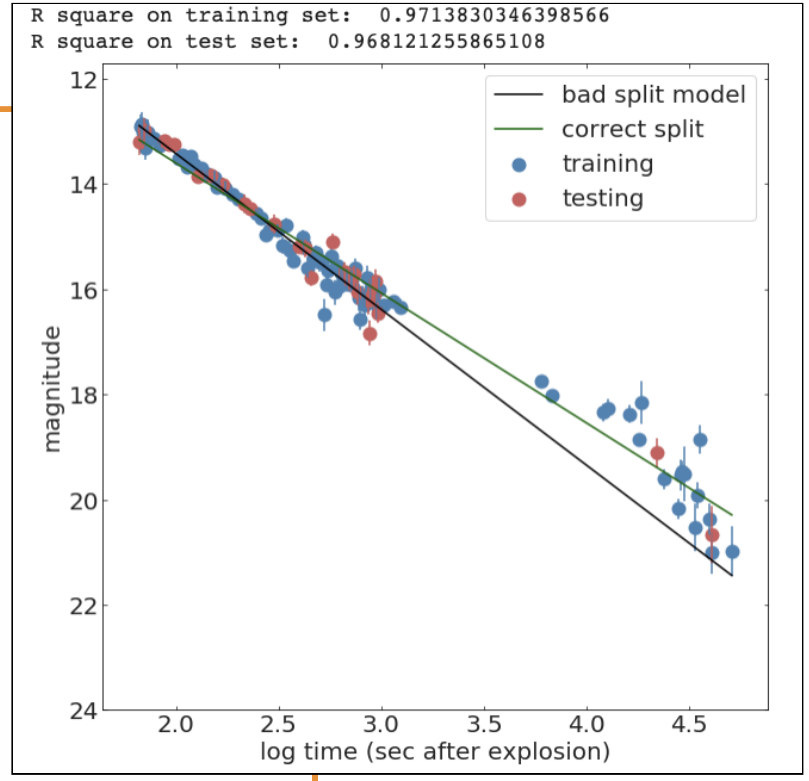
Split the sample in test and training sets

Train on the training set

Test (measure accuracy) on the test set

validating a model

```
1  from sklearn.model_selection import train_test_split
2
3  def line(x, intercept, slope):
4      return slope * x + intercept
5
6  def chi2(args, x, y, s):
7      a, b = args
8      return sum((y - line(x, a, b))**2 / s)
9
10 x_train, x_test, y_train, y_test, s_train, s_test = train_test_split(
11      x, y, s, test_size=0.25, random_state=42)
12
13 initialGuess = (10, 1)
14
15 chi2Solution_goodsplit = minimize(chi2, initialGuess,
16         args=(x_train, y_train, s_train))
17
18 print("best fit parameters from the minimization of the chi squared: " +
19       "slope {:.2f}, intercept {:.2f}".format(*chi2Solution_goodsplit.x))
20
21 print("R square on training set: ", Rsquare(chi2Solution_goodsplit.x, x_train, y_train))
22 print("R square on test set: ", Rsquare(chi2Solution_goodsplit.x, x_test, y_test))
```



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.

Intermission:
Gamma Ray Bursts

TL;DR: Gamma-ray bursts (GRBs) are bright X-ray and gamma-ray flashes observed in the sky, emitted by distant extragalactic sources. They are associated with the creation or merging of neutron stars or black holes.

Reprocessed X- and gamma-ray emission is visible in optical wavelengths, the "afterglow"

We measure a quantity named magnitude over time, which is an inverse logaritmic measure of brightness of the GRB, and which is expected to change it roughly linearly with the logarithm of time.

Details: The light that we measure from these explosions changes over time, so we can study its **time series or lightcurve**.

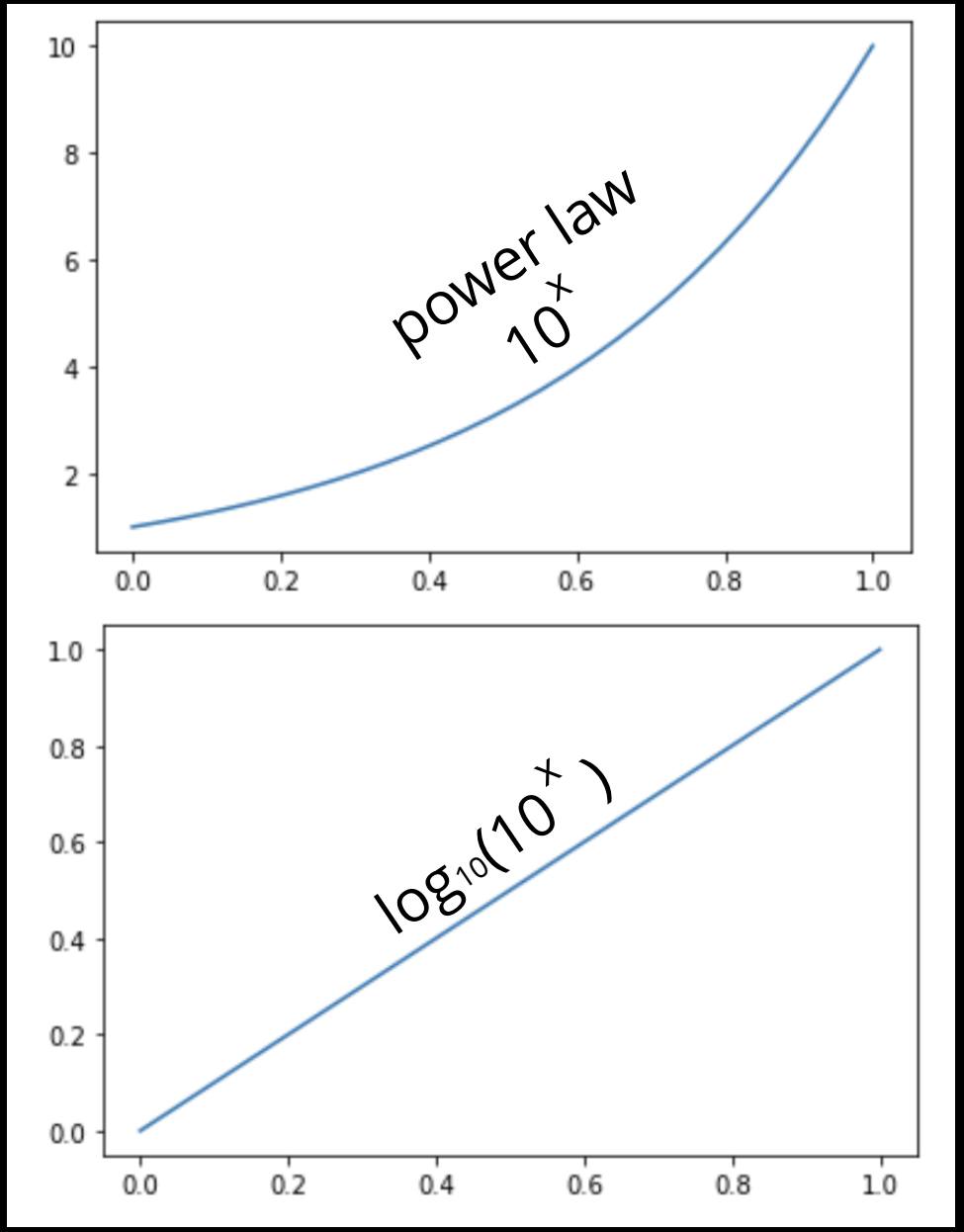
The GRB afterglow is generated by a power-law process.

The change in light follows a power law, it's not linear, but

the logarithm of a power law is a line.

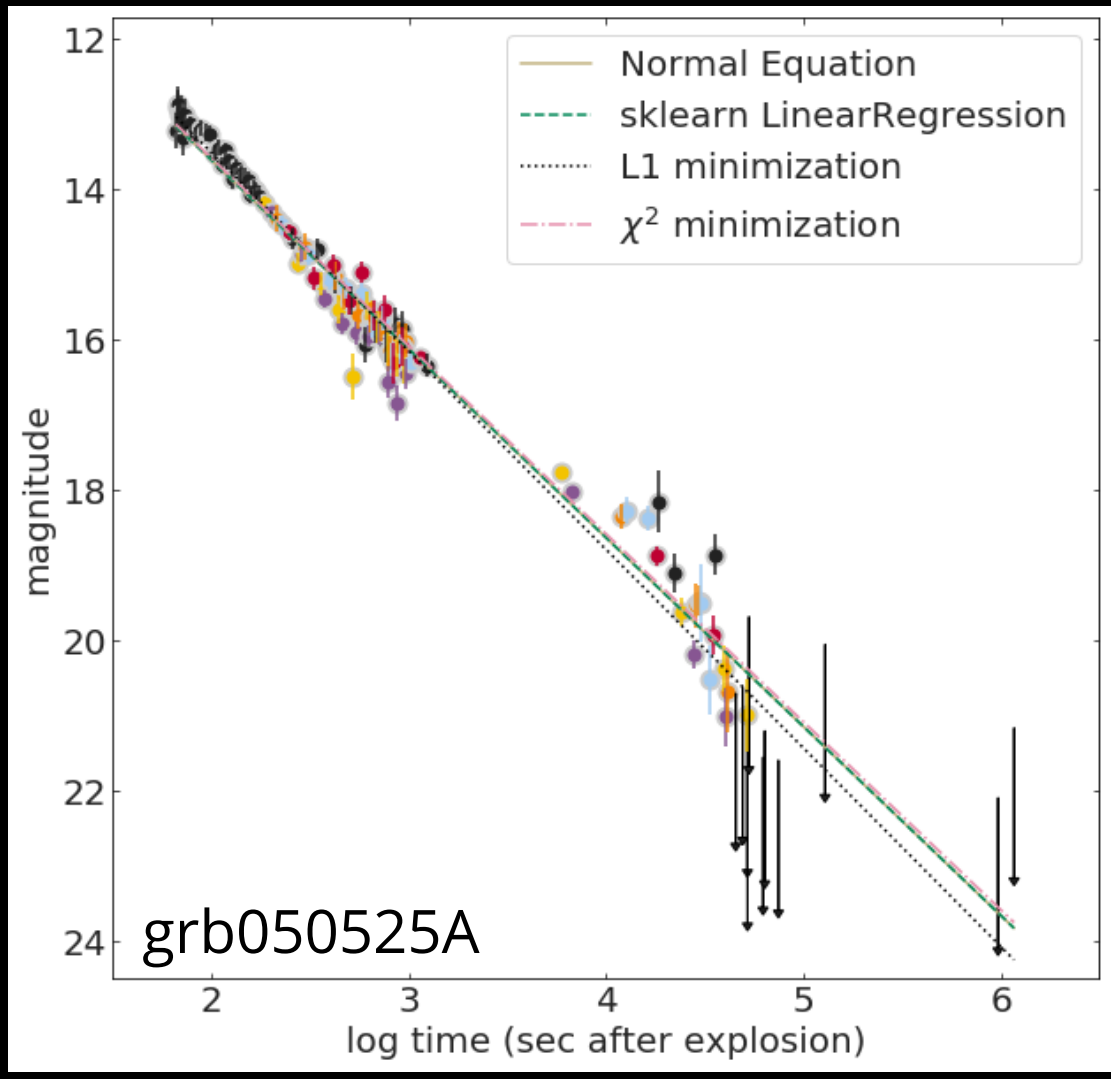
The logarithm (base 10) of the light flux is called **magnitude** in astronomy.

$m = 25 - 2.5\log_{10}(\text{flux})$



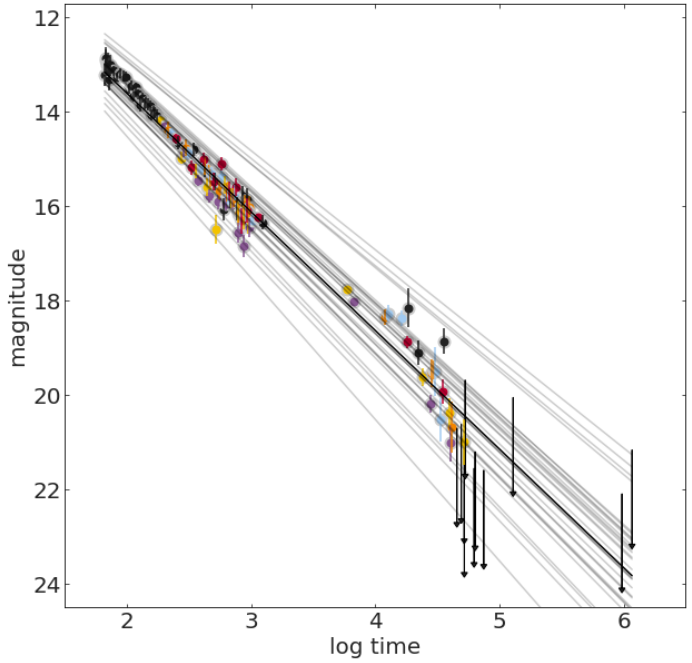
More Details: It is believed that the afterglow originates in the external shock produced as the blast wave from the explosion collides with and sweeps up material in the surrounding interstellar medium. The emission is synchrotron emission produced when electrons are accelerated in the presence of a magnetic field. The successive afterglows at progressively lower wavelengths (X-ray, optical, radio) result naturally as the expanding shock wave sweeps up more and more material causing it to slow down and lose energy.

*X-ray afterglows have been observed for all GRBs, but only about 50% of GRBs also exhibit afterglows at optical and radio wavelengths [ref]



2 Linear Regression

Regression



Fitting a line

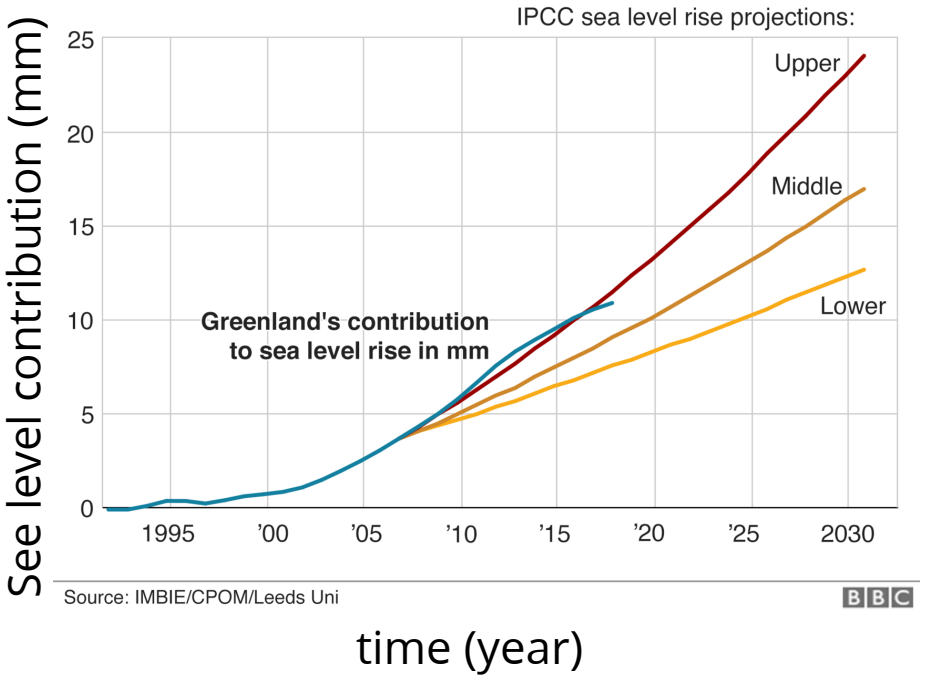
$ax+b$

to data y

WHY?

Regression

To predict and forecast



https://desdemonadespair.net/2019/12/greenland-losing-ice-seven-times-faster-than-in-the-1990s-sea-level-rise-from-greenland-melt-tracking-highest-climate-projections.html

Fitting a line

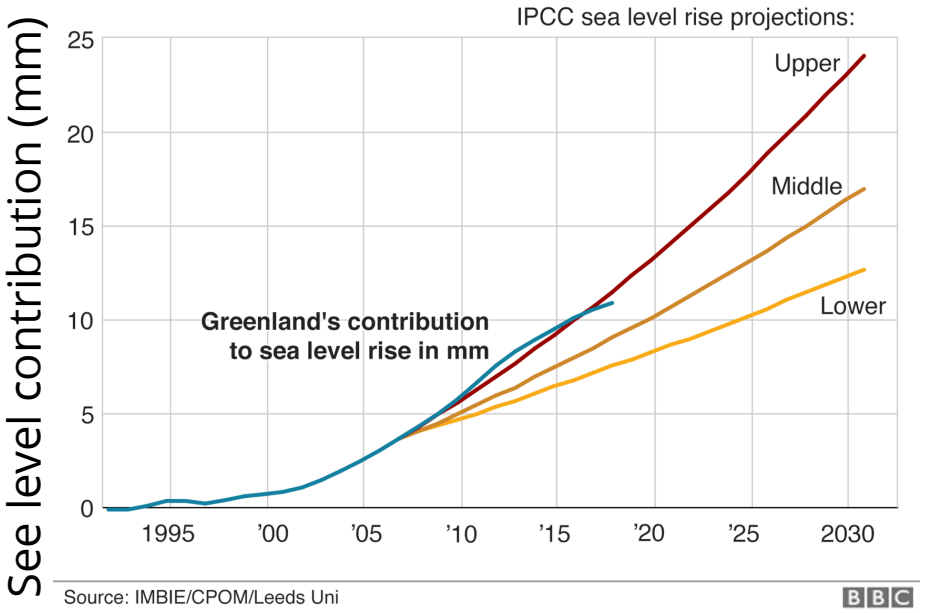
$ax+b$

to data y

WHY?

Regression

To predict and forecast



time (year)

https://desdemonadespair.net/2019/12/greenland-losing-ice-seven-times-faster-than-in-the-1990s-sea-level-rise-from-greenland-melt-tracking-highest-climate-projections.html

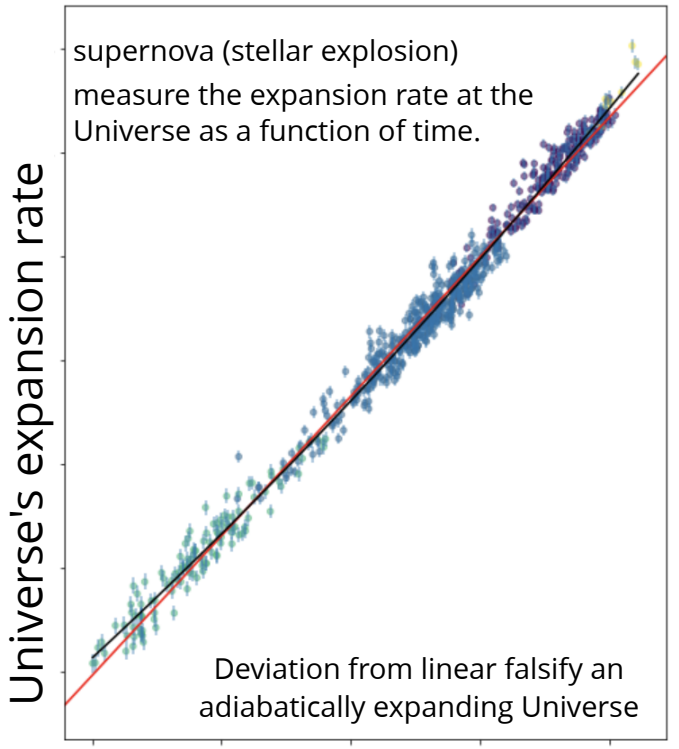
Fitting a line

$ax+b$

to data y

WHY?

To explain



https://github.com/fedhere/DSPS/blob/master/HW6/SNdataLineFit_solution.ipynb

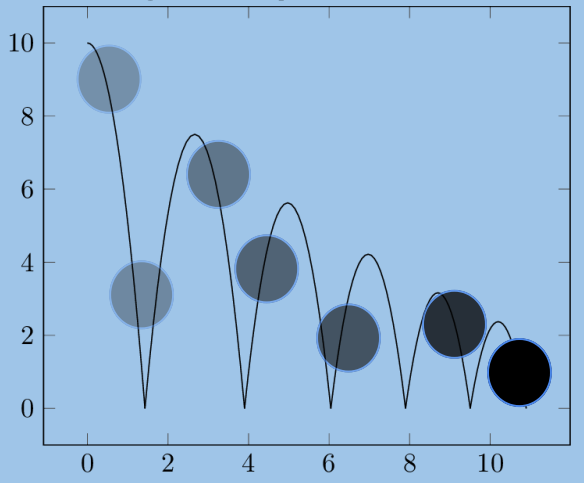
Regression & Model Fitting

We fit models to data in order to:

Predict and forecast: predict the value of the *endogenous* (dependent) variable at locations of the *exogenous* (independent, time) variable where we have no observations. This can be within the observed range, or outside of the range, which in time-series means predict the future (*forecast*)

Explain: relate observed behavior to first principles or behavior of possibly variables to explain the evolution and assess causality.

E.g. fitting a parabola to a bouncing ball demonstrates that gravity (and initial velocity) explains the behavior



Key Concept

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:

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

```
1 x = grbAG[grbAG.upperlimit == 0].logtime.values
2 y = grbAG.loc[grbAG.upperlimit == 0].mag.values
3
4 X = np.c_[np.ones((len(x), 1)), x]
5
6
7 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 = grbAG[grbAG.upperlimit == 0].logtime.values
2 y = grbAG.loc[grbAG.upperlimit == 0].mag.values
3
4 X = np.c_[np.ones((len(x), 1)), x]
5
6
7 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:

$$\underset{2\text{xN}}{(X^T} \cdot \underset{\text{Nx2}}{X)^{-1}} \cdot \underset{2\text{xN}}{X^T} \cdot \underset{\text{Nx1}}{\vec{y}} = \underset{2\text{x1}}{\begin{pmatrix} a \\ b \end{pmatrix}}$$

```
x = grbAG[grbAG.upperlimit == 0].logtime.values
y = grbAG.loc[grbAG.upperlimit == 0].mag.values

X = np.c_[np.ones((len(x), 1)), x]


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

independent variable

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

Linear Regression
Normal Equation

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

$$\underset{2\text{xN}}{(X^T} \cdot \underset{\text{Nx2}}{X)^{-1}} \cdot \underset{2\text{xN}}{X^T} \cdot \underset{\text{Nx1}}{\vec{y}} = \underset{2\text{x1}}{\begin{pmatrix} a \\ b \end{pmatrix}}$$

```
x = grbAG[grbAG.upperlimit == 0].logtime.values
y = grbAG.loc[grbAG.upperlimit == 0].mag.values

X = np.c_[np.ones((len(x), 1)), x]


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

independent variable

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

Linear Regression Normal Equation

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

$$\underset{2\text{xN}}{(X^T} \cdot \underset{\text{Nx2}}{X)^{-1}} \cdot \underset{2\text{xN}}{X^T} \cdot \underset{\text{Nx1}}{\vec{y}} = \underset{2\text{x1}}{\begin{pmatrix} a \\ b \end{pmatrix}}$$

```
x = grbAG[grbAG.upperlimit == 0].logtime.values
y = grbAG.loc[grbAG.upperlimit == 0].mag.values

X = np.c_[np.ones((len(x), 1)), x]


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 = grbAG[grbAG.upperlimit == 0].logtime.values
y = grbAG.loc[grbAG.upperlimit == 0].mag.values

X = np.c_[np.ones((len(x), 1)), x]

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:

$$\underset{2\text{xN}}{(X^T} \cdot \underset{\text{Nx2}}{X)^{-1}} \cdot \underset{2\text{xN}}{X^T} \cdot \underset{\text{Nx1}}{\vec{y}} = \underset{2\text{x1}}{\begin{pmatrix} a \\ b \end{pmatrix}}$$

```
x = grbAG[grbAG.upperlimit == 0].logtime.values
y = grbAG.loc[grbAG.upperlimit == 0].mag.values

X = np.c_[np.ones((len(x), 1)), x]


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 = grbAG[grbAG.upperlimit == 0].logtime.values
y = grbAG.loc[grbAG.upperlimit == 0].mag.values

X = np.c_[np.ones((len(x), 1)), x]

lr.fit(X, y)
lr.coef_, lr.intercept_
```

2.2 Regression

objective function

Objective Function

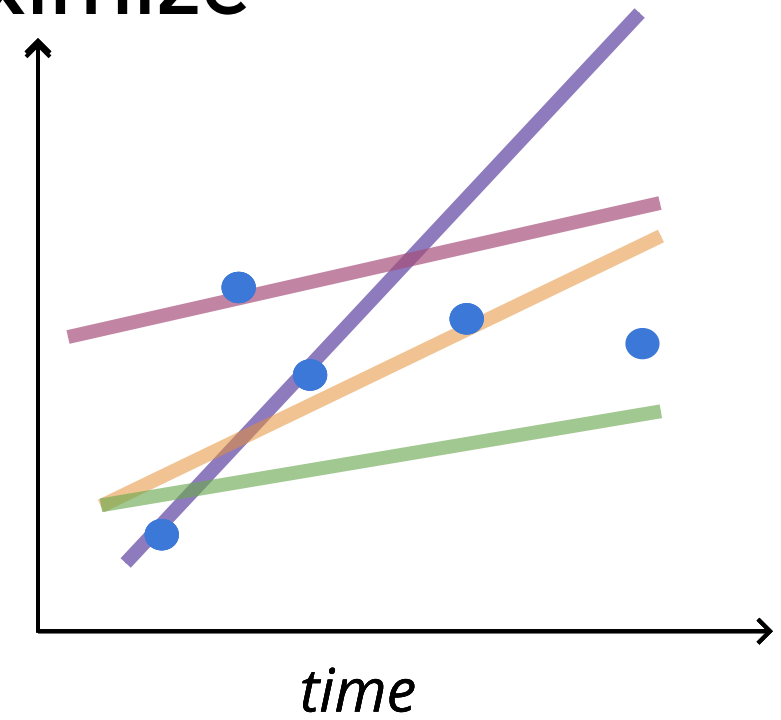
If there is no analytical solution

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



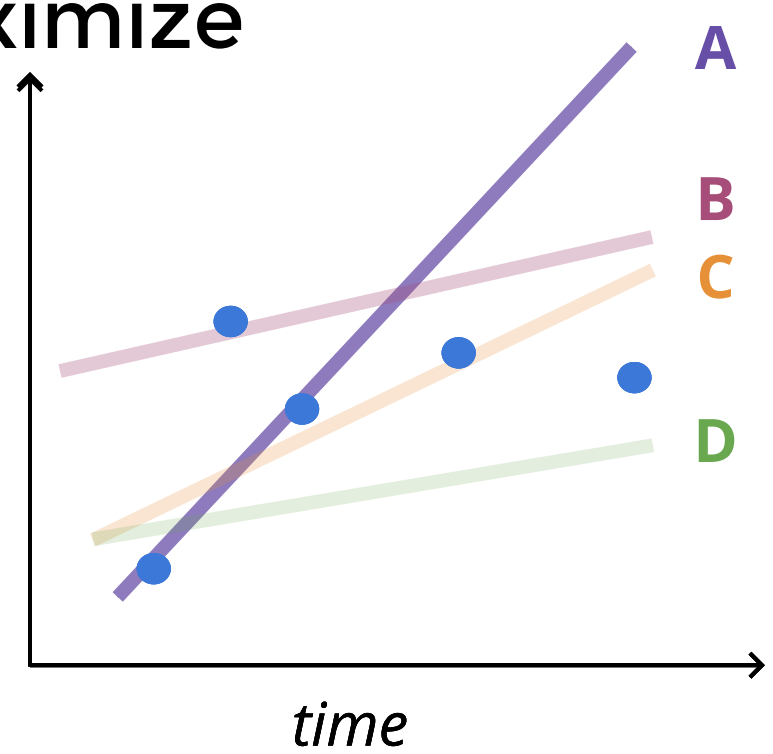
Objective Function

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?



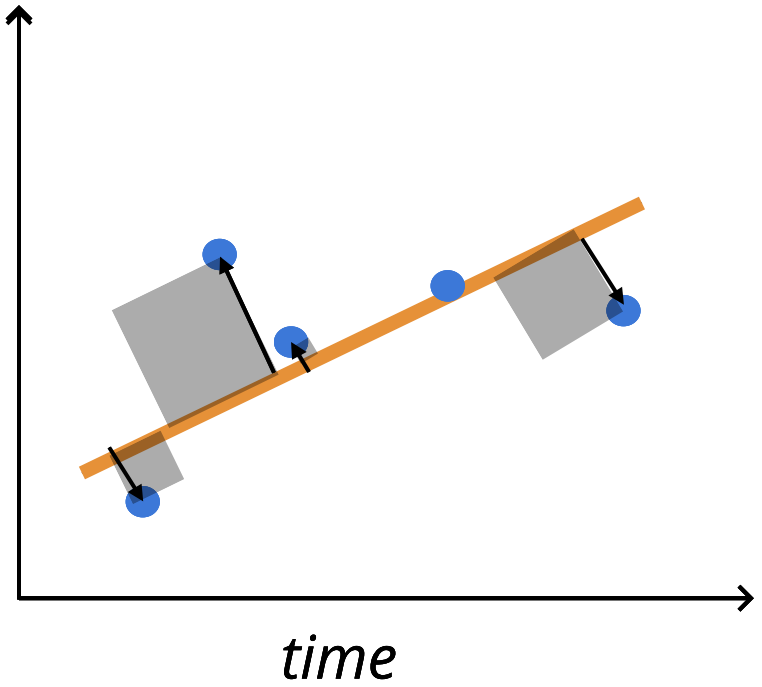
Objective Function

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(a, b, x):
    return a * x + b
def fitfunc(args, x, y):
    a, b = args
    return np.abs((y - line(a, b, x)))

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

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

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

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

```
from scipy.optimize import minimize
def line(a, b, x):
    return a * x + b
def fitfunc(args, x, y):
    a, b = args
    return np.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(a, b, x):
 3     return a * x + b
 4 def chi2(args, x, y, s):
 5     a, b = args
 6     return np.sum((y - line(a, b, x))**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))
```

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

Data analysis recipes: Fitting a model to data

Intro and Chapter 1; pages 1-8

D. Hogg et al. https://arxiv.org/abs/1008.4686

Lots of details about how to properly treat outliers, uncertainties, assumptions in fitting a line to data. Witty comments make it entertaining. Exercise it make it very helpful

Reading

Data analysis recipes:
Fitting a model to data*

David W. Hogg
Center for Cosmology and Particle Physics, Department of Physics, New York University
Max-Planck-Institut für Astronomie, Heidelberg

Jo Bovy
Center for Cosmology and Particle Physics, Department of Physics, New York University

Dustin Lang
Department of Computer Science, University of Toronto
Princeton University Observatory

Abstract

We go through the many considerations involved in fitting a model to data, using as an example the fit of a straight line to a set of points in a two-dimensional plane. Standard weighted least-squares fitting is only appropriate when there is a dimension along which the data points have negligible uncertainties, and another along which all the uncertainties can be described by Gaussians of known variance; these conditions are rarely met in practice. We consider cases of general, heterogeneous, and arbitrarily covariant two-dimensional uncertainties, and situations in which there are bad data (large outliers), unknown uncertainties, and unknown but expected intrinsic scatter in the linear relationship being fit. Above all we emphasize the importance of having a "generative model" for the data, even an approximate one. Once there is a generative model, the subsequent fitting is non-arbitrary because the model permits direct computation of the likelihood of the parameters or the posterior probability distribution. Construction of a posterior probability distribution is indispensible if there are "nuisance parameters" to marginalize away.

8 ways to do linear regression and measure their speed

Data analysis recipes: Fitting a model to data

D. Hogg et al. https://arxiv.org/abs/1008.4686 - lots of details about how to properly treat outliers, uncertainties, assumptions in fitting a line to data. Witty comments make it entertaining. Exercise it make it very helpful

AstroML Chapter 10 - Intro

HOMLwSKLKerasTF Chapter 4 pages 111-117

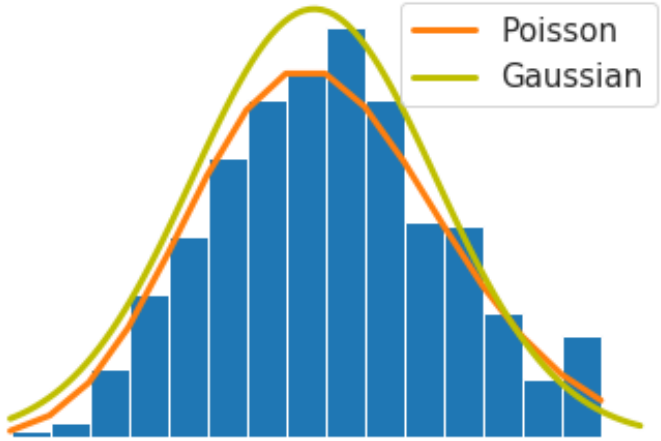
Elements of Statistical Learning Chapter 3 Section 1 and 2

References

4 uncertainties in measurements

1 uncertainty in measurements

2 stochasticity in nature



3 combining uncertainties

4 MonteCarlo methods

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

5 Dark matter

4. systematic uncertainties

stochastic or random errors

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

stochastic or random 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 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 errors

every measurement will be a bit different

2.0 +/- ε cm, ε > 0.1 cm

stochastic or random errors

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

Stochastic processes can be *completely random*: the probability of any event is disjoint from that of the previous one

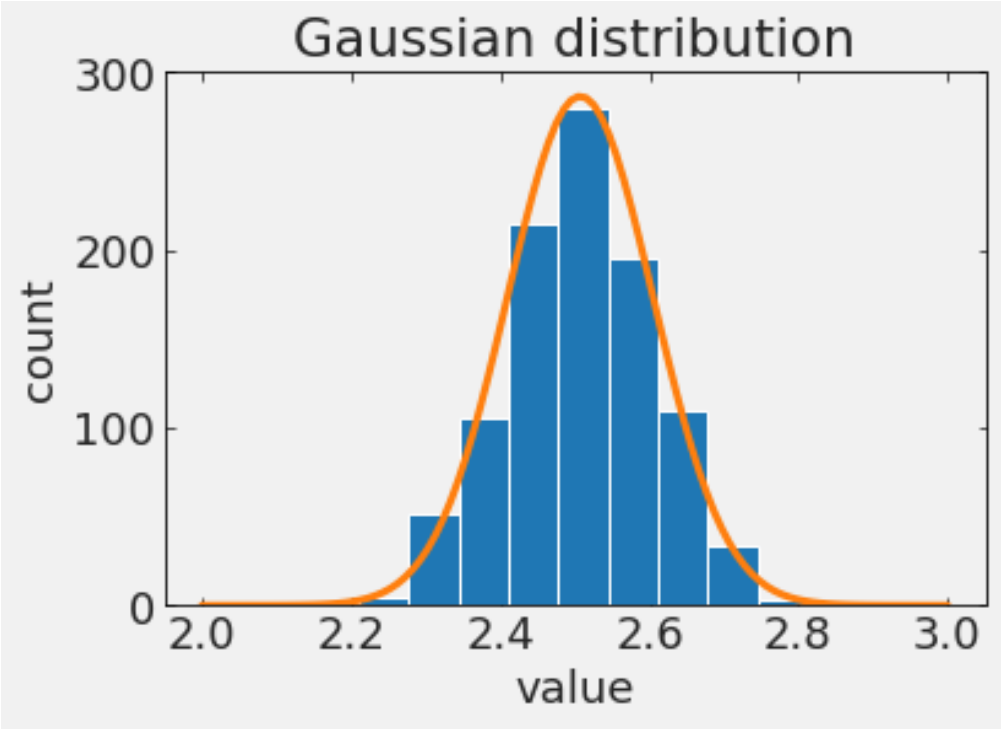
stochastic or random errors

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

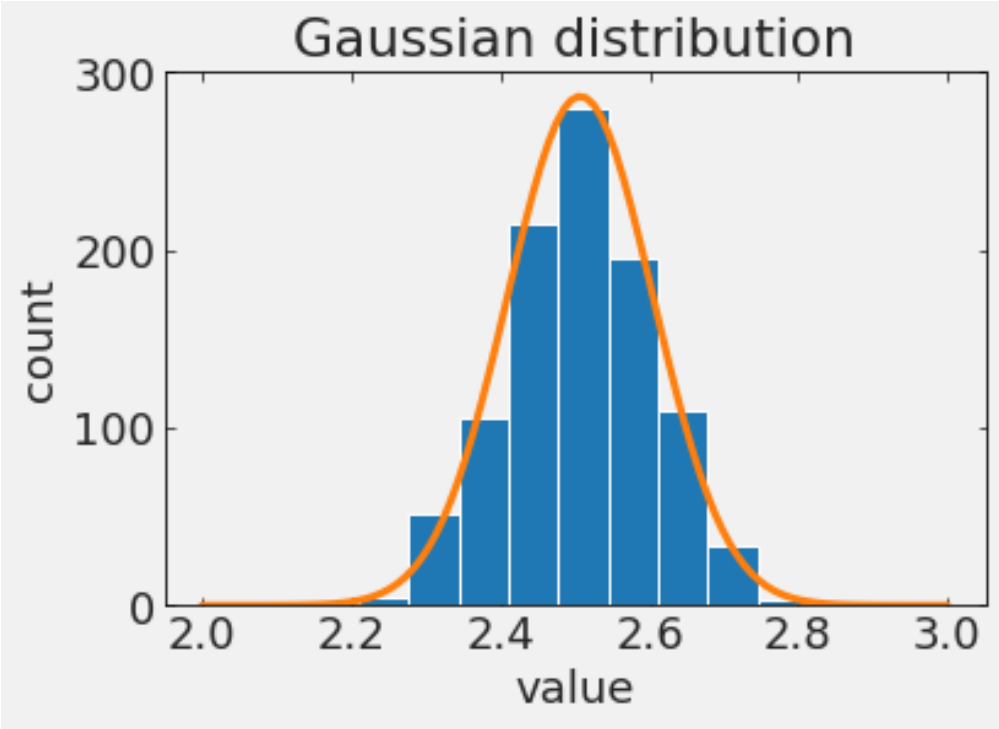
stochastic or random errors



2.0 +/- ε cm, ε > 0.1 cm

stochastic or random errors

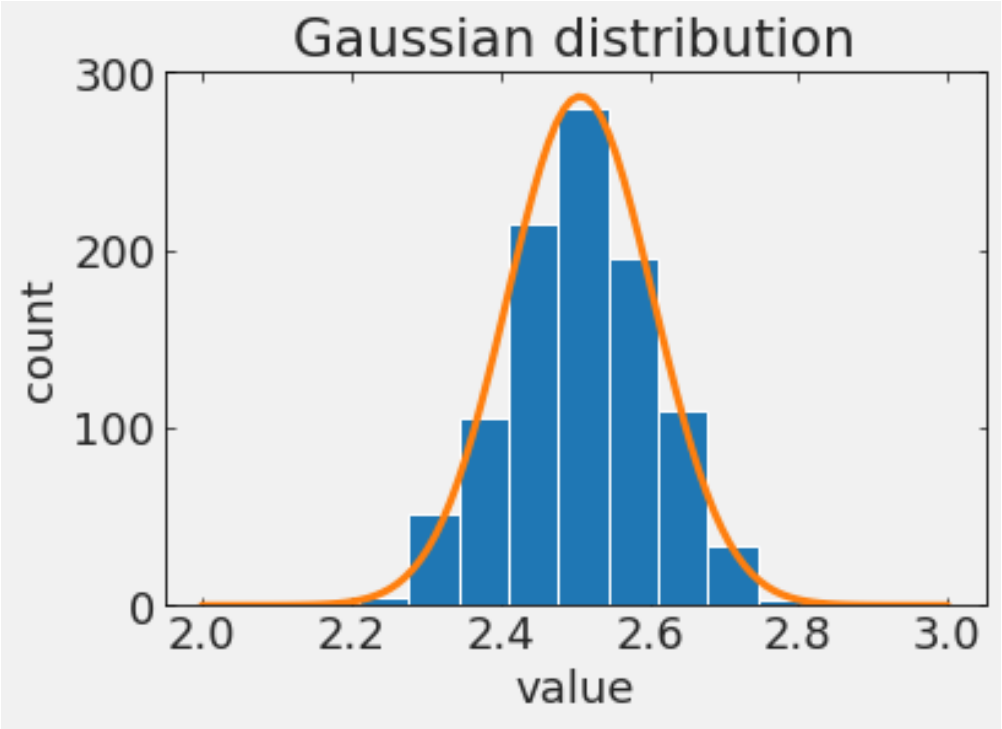
$$p(y_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp - \frac{(y_i - (mx_i + b))^2}{2\sigma_i^2}$$



stochastic or random errors

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

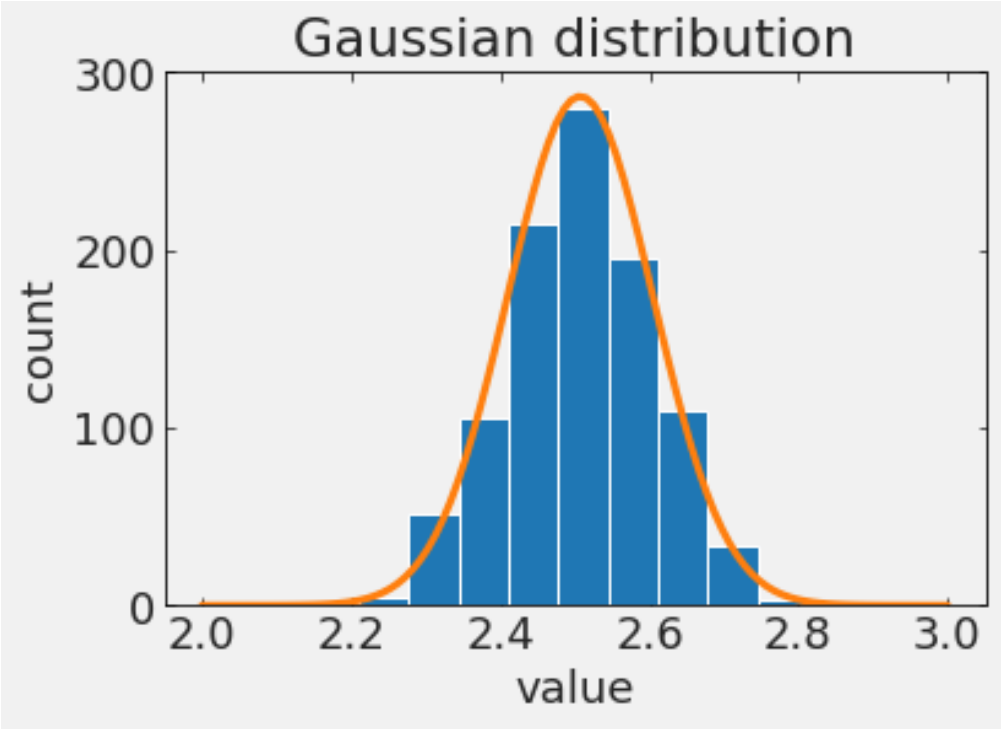
- symmetric



stochastic or random errors

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

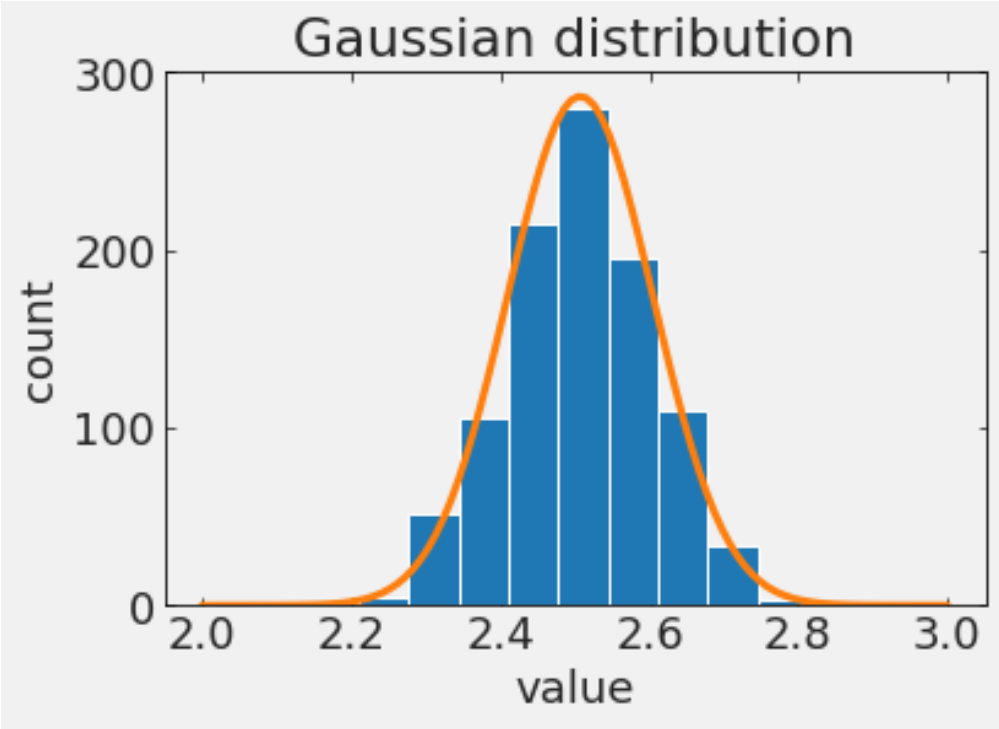
- symmetric
- max at $y_i = (mx_i + b)$



stochastic or random 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 errors

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

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

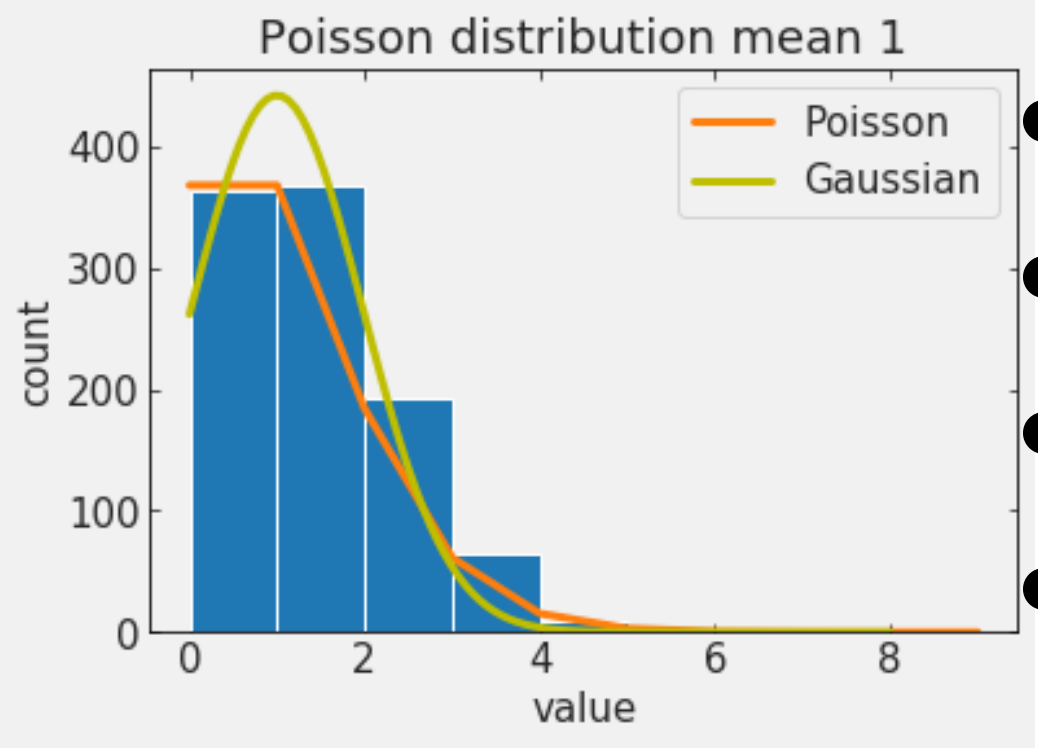
occurring in a fixed period of time if these events occur with a known average rate

and independently of the time since the last event.

stochastic or random errors

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

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

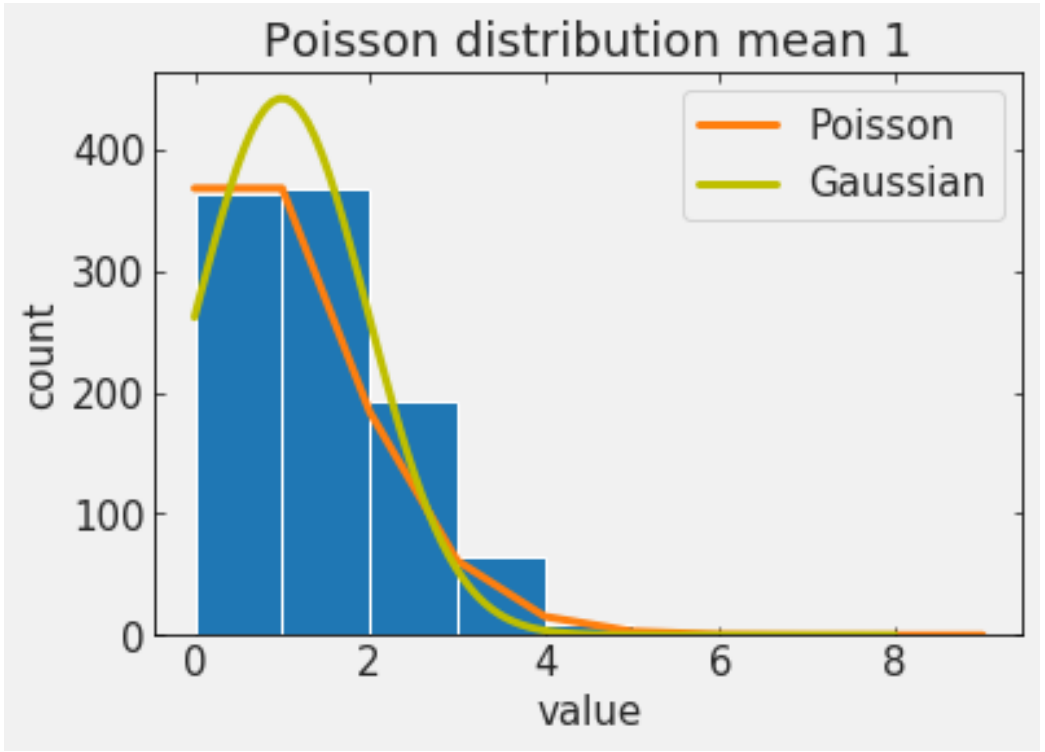


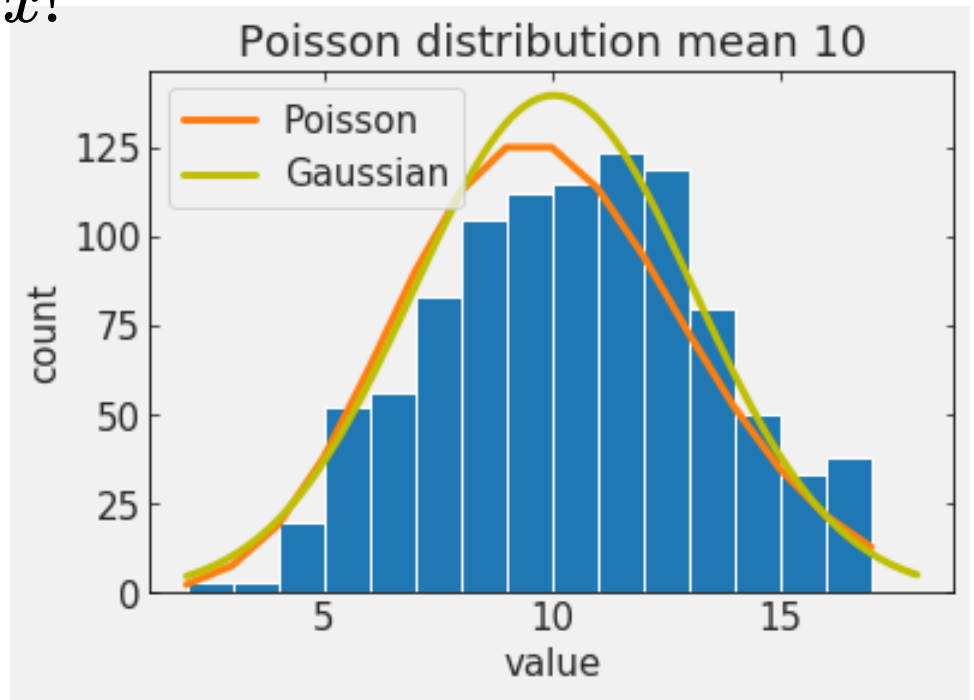
- asymmetric
- integer support
- support >0
- mean and stdev are related: $\mu : \lambda$ $\sigma : \sqrt{\lambda}$

stochastic or random errors

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

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

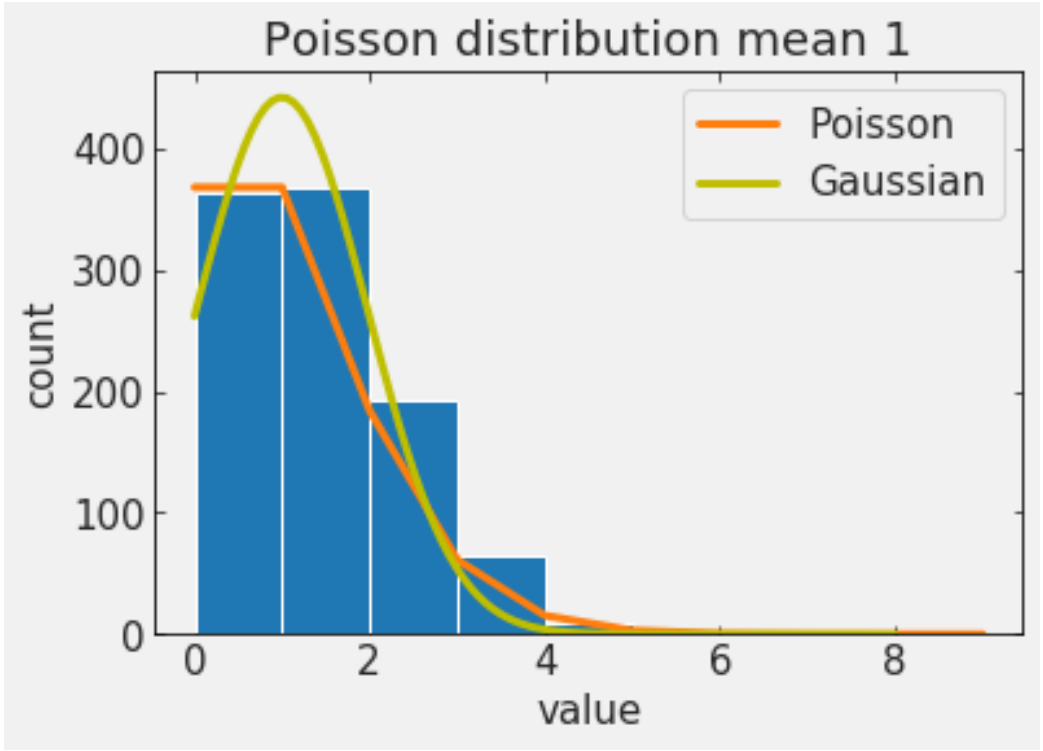




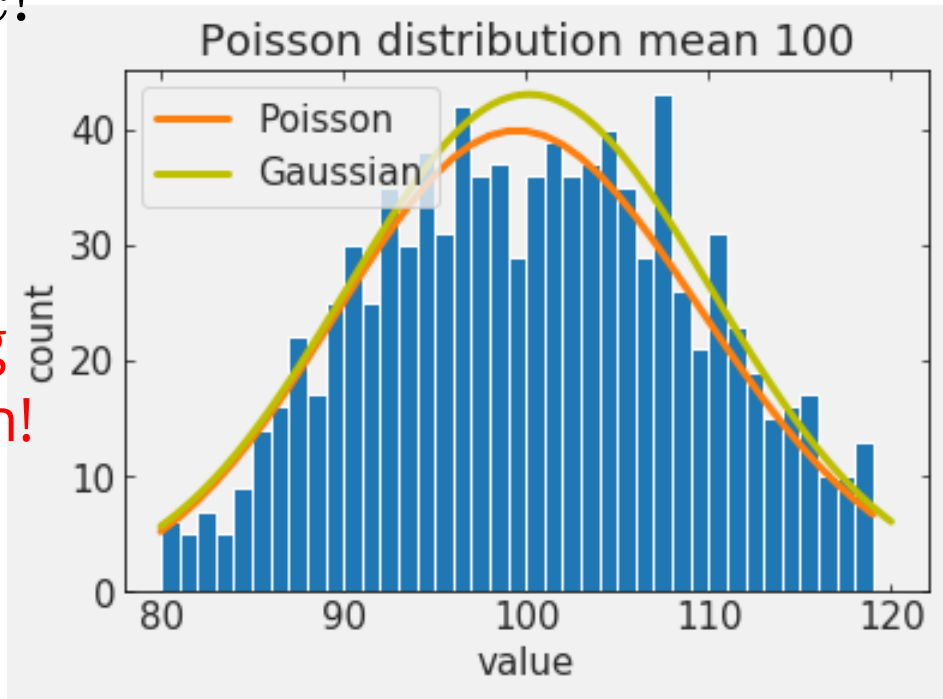
stochastic or random errors

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

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



as the mean increases it starts looking like a Gaussian!



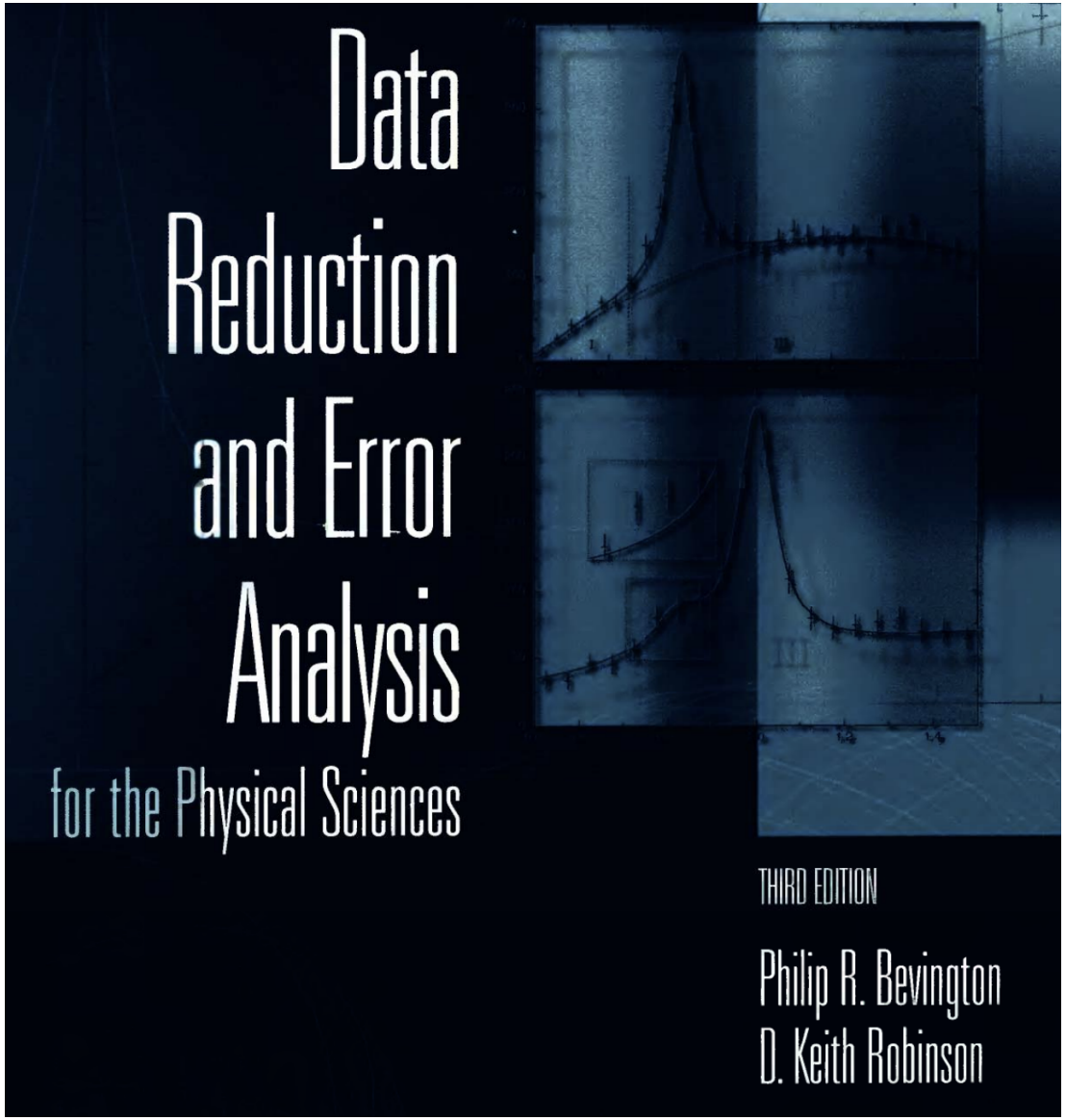
42

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.

~~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

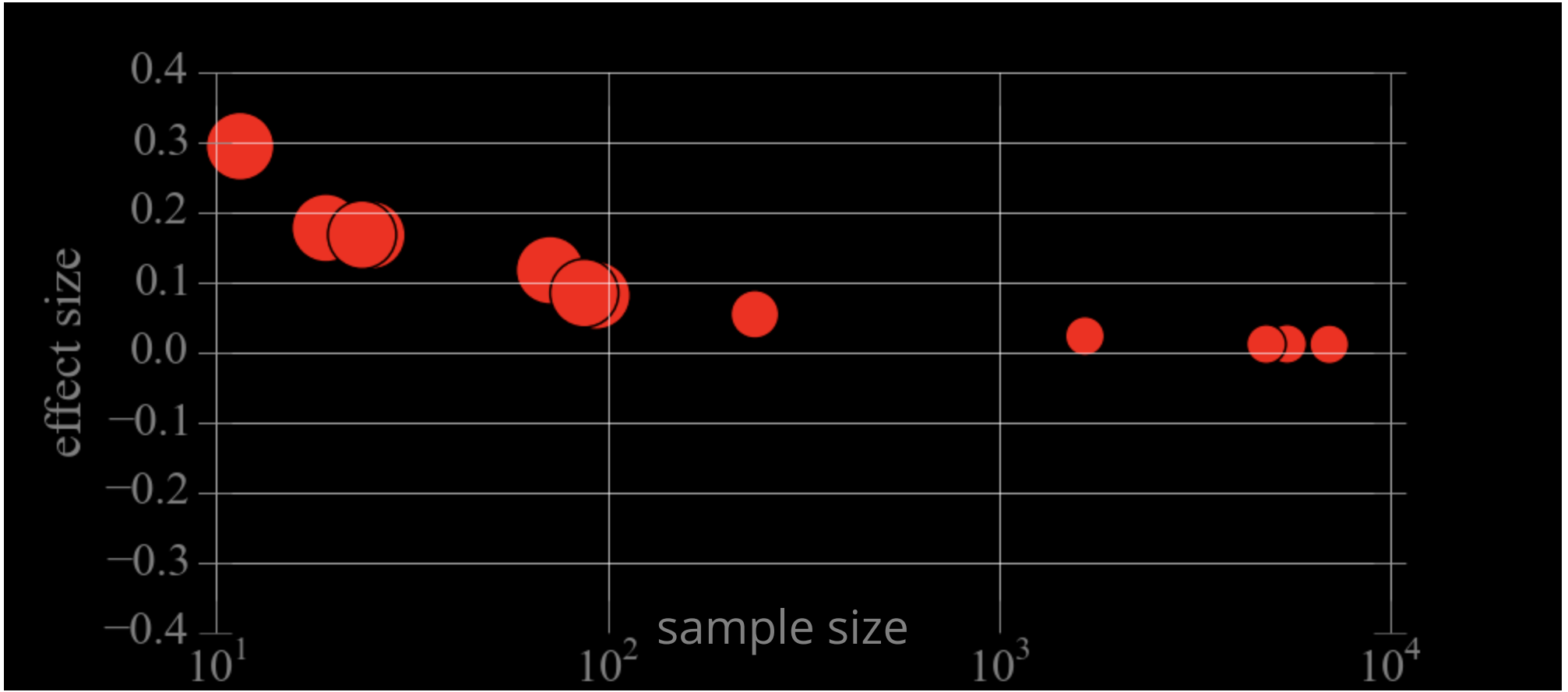
His team investigated the fate of 221 sociological studies conducted between 2002 and 2012, which were recorded by Time-sharing Experiments for the Social Sciences (TESS), a US project that helps social scientists to carry out large-scale surveys of people's views.

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.

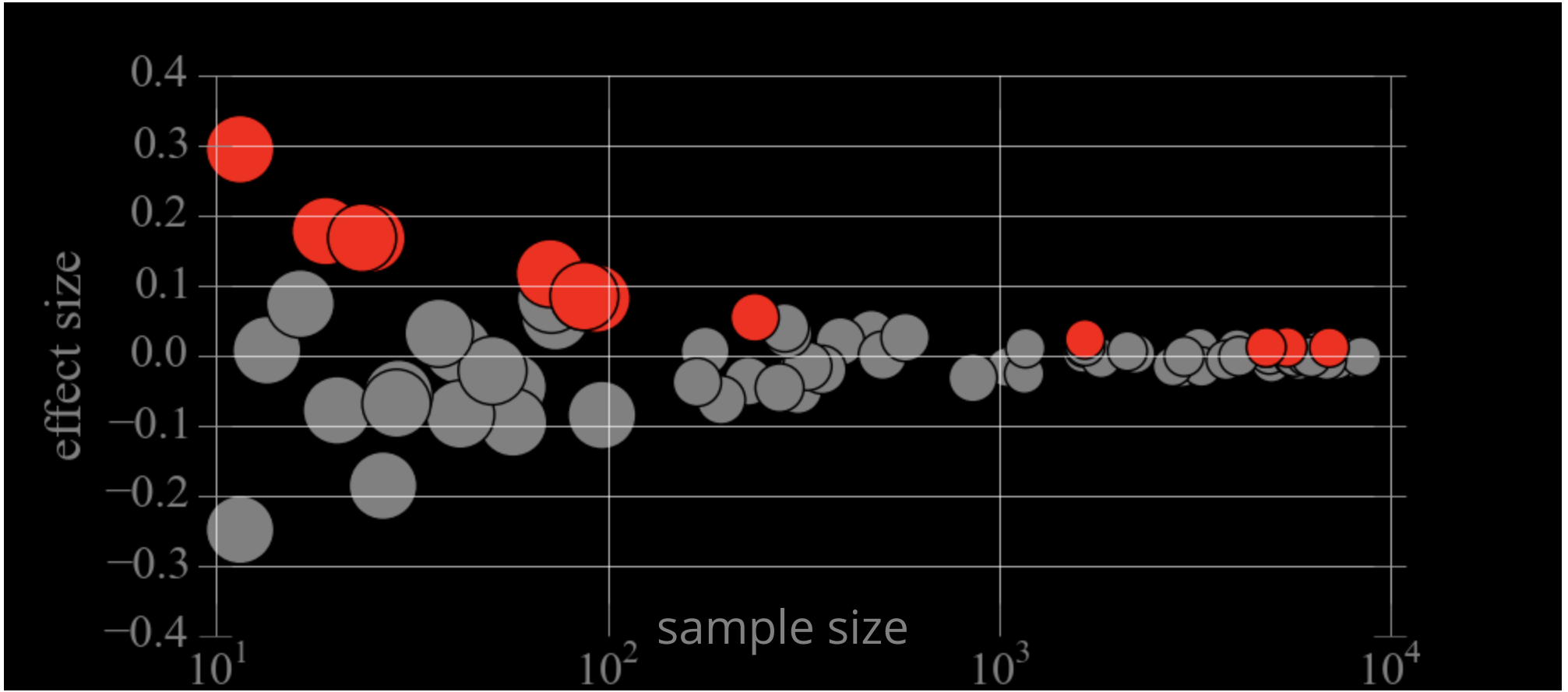
Bias in measurements: know your data

Publication Bias



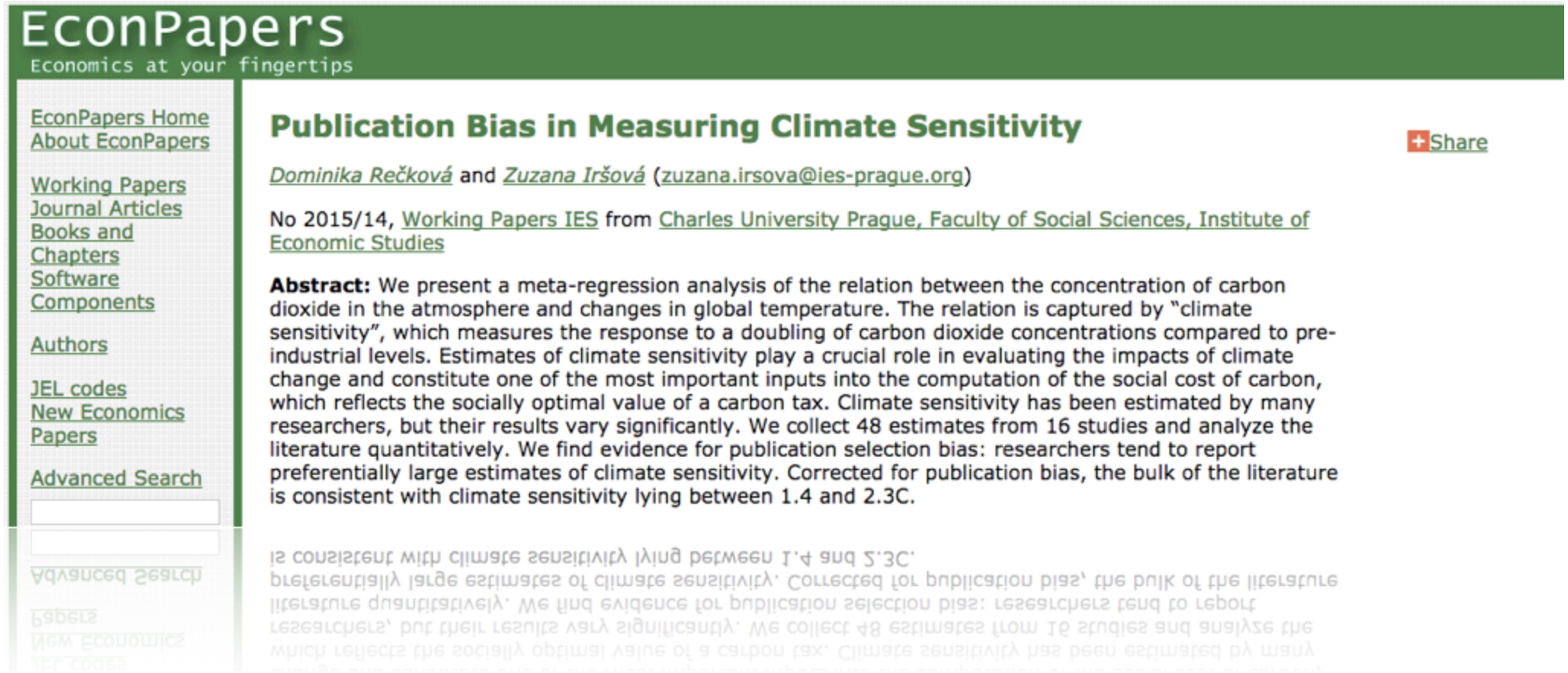
Bias in measurements: know your data

Publication Bias



Bias in measurements: know your data

Publication Bias



EconPapers
Economics at your fingertips

EconPapers Home
About EconPapers

Working Papers
Journal Articles
Books and Chapters
Software Components

Authors

JEL codes
New Economics Papers

Advanced Search

Publication Bias in Measuring Climate Sensitivity

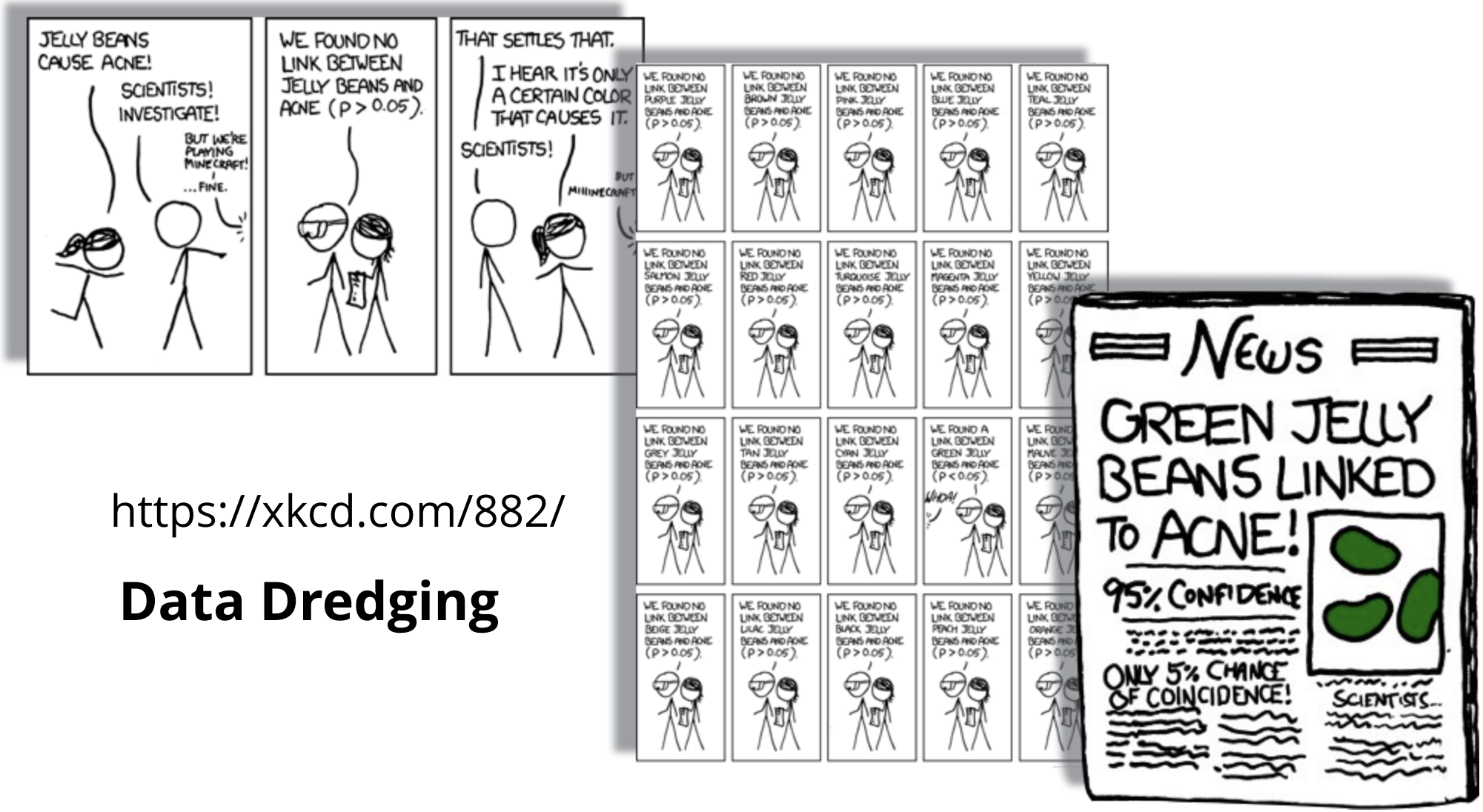
Share

Dominika Rečková and *Zuzana Iršová* (zuzana.irsova@ies-prague.org)

No 2015/14, Working Papers IES from Charles University Prague, Faculty of Social Sciences, Institute of Economic Studies

Abstract: We present a meta-regression analysis of the relation between the concentration of carbon dioxide in the atmosphere and changes in global temperature. The relation is captured by "climate sensitivity", which measures the response to a doubling of carbon dioxide concentrations compared to pre-industrial levels. Estimates of climate sensitivity play a crucial role in evaluating the impacts of climate change and constitute one of the most important inputs into the computation of the social cost of carbon, which reflects the socially optimal value of a carbon tax. Climate sensitivity has been estimated by many researchers, but their results vary significantly. We collect 48 estimates from 16 studies and analyze the literature quantitatively. We find evidence for publication selection bias: researchers tend to report preferentially large estimates of climate sensitivity. Corrected for publication bias, the bulk of the literature is consistent with climate sensitivity lying between 1.4 and 2.3C.

Bias in measurements: know your data



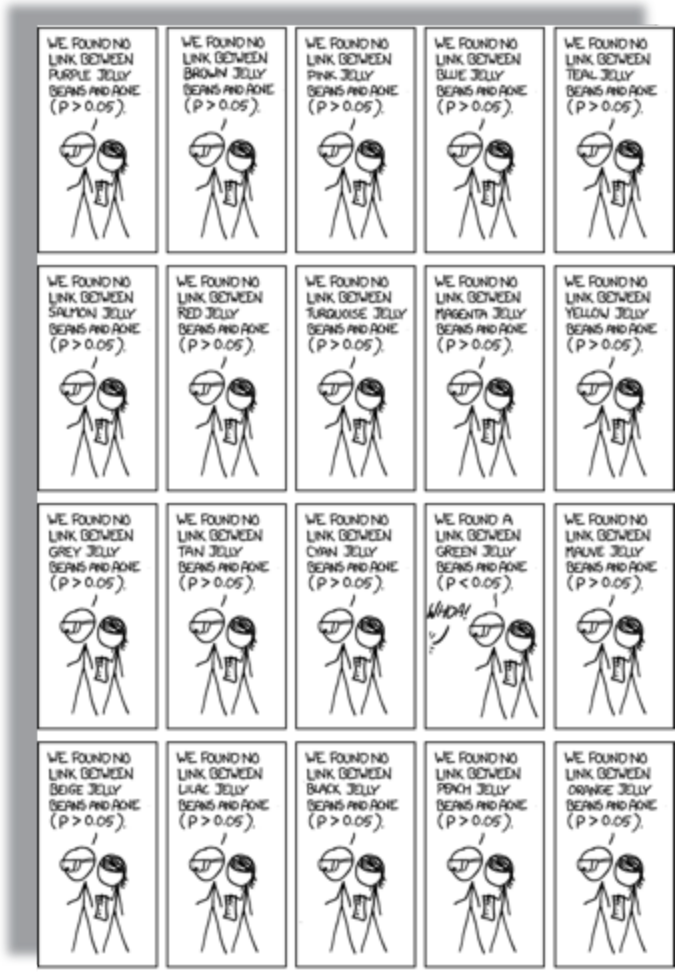
https://xkcd.com/882/

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

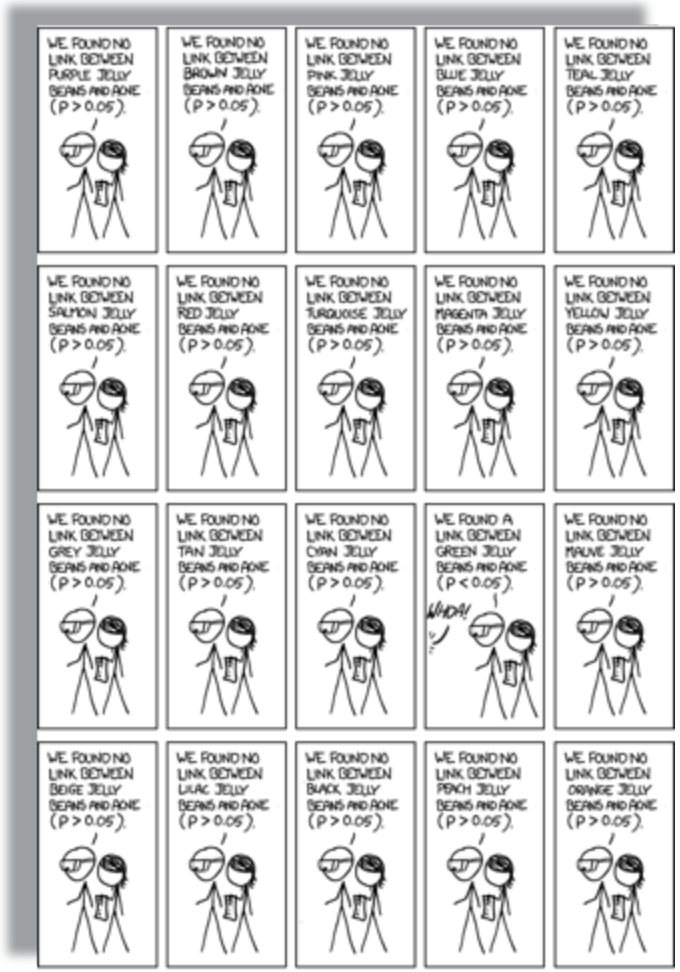
significance 95%

20 tests are preformed

assume independence:

if pi = 0.05 for each i=1..20

Data Dredging

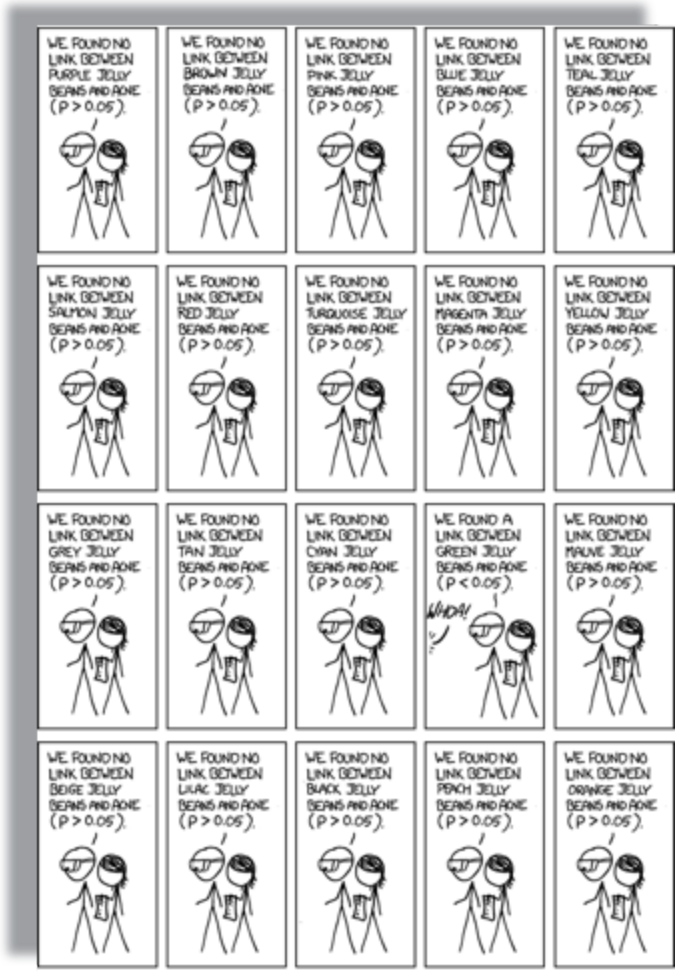


Bias in measurements: know your data

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

20 tests are preformed

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



Data Dredging

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

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

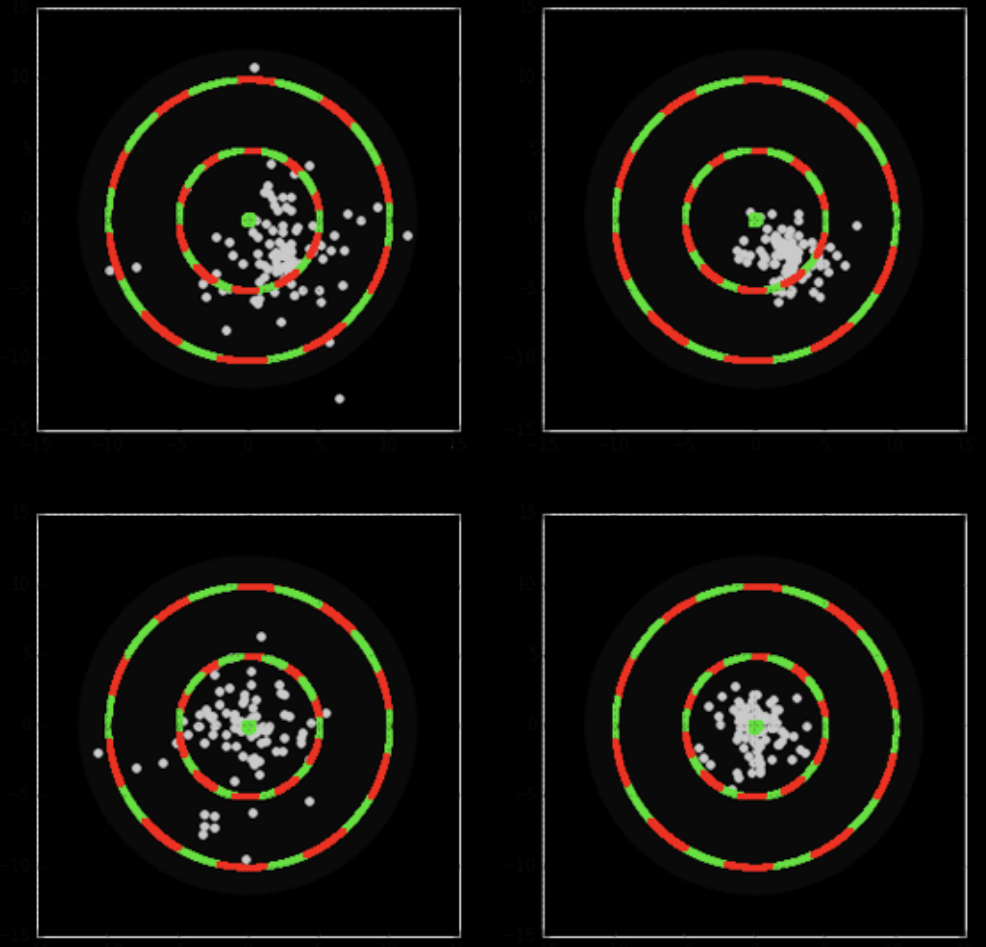
Stochastic vs Systematics

Systematic	**Statistical**
Biases the measurement *in one direction*	No preferred direction
Affects the sample regardless of the size	Shrinks with the sample size (typically as N)
Any distribution (usually we use Gaussian though)	Typically Gaussian or Poisson

Precision vs Accuracy

Precision

Accuracy



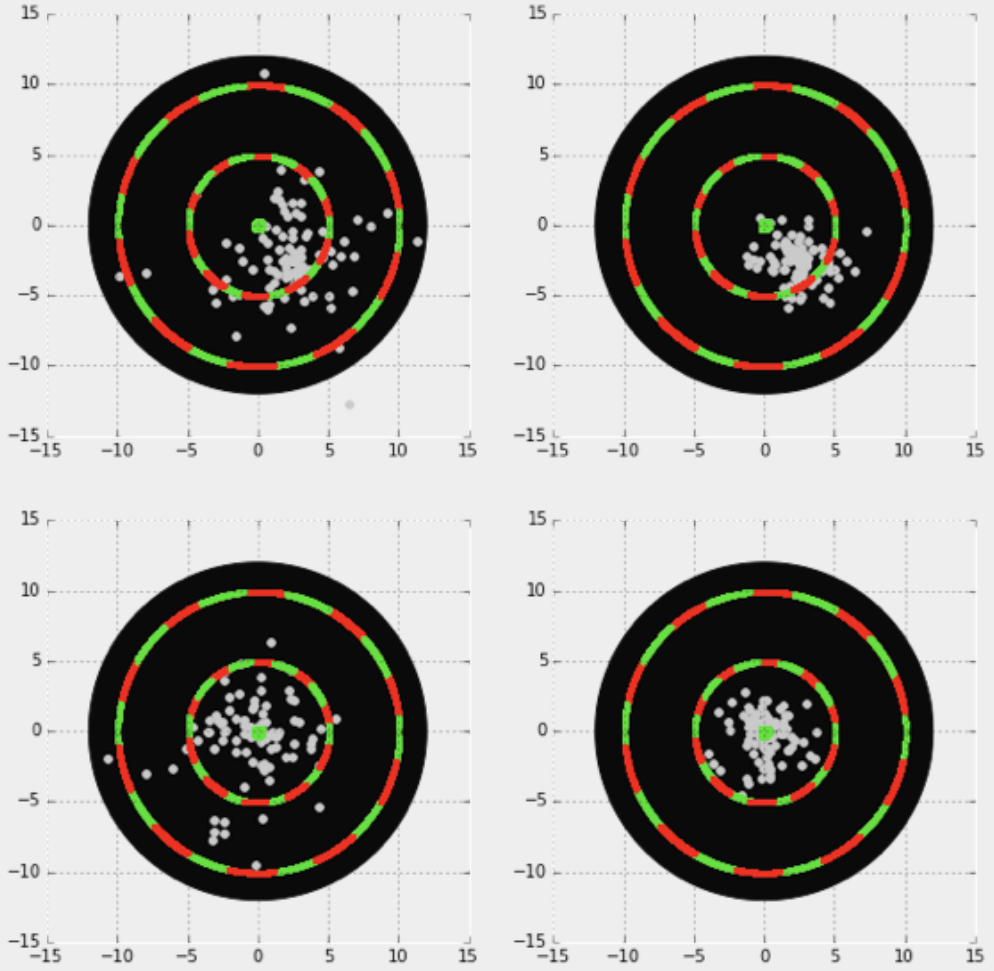
Precision vs Accuracy

which relates to stochastic errors,

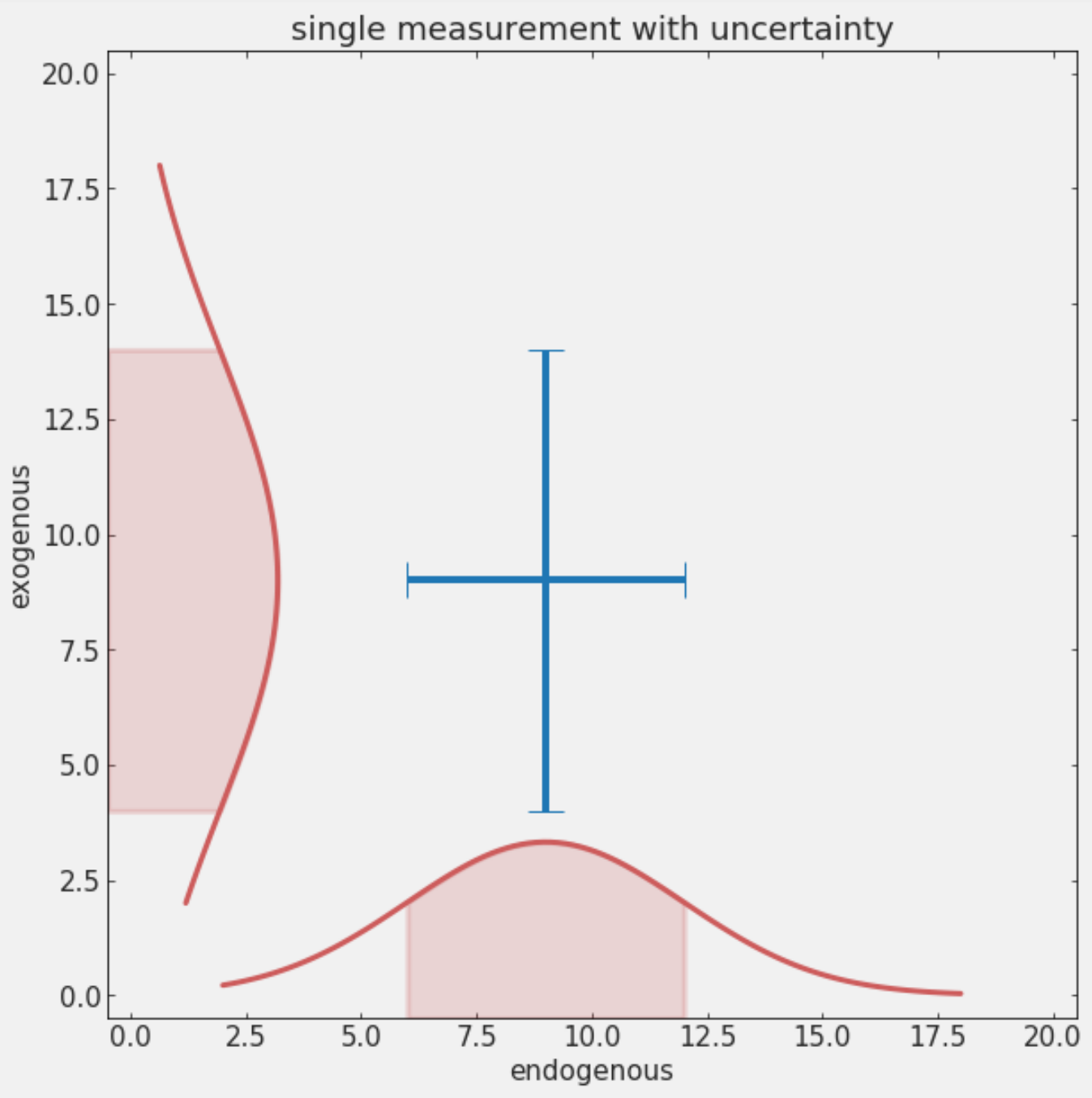
which to stsystematic?

Precision

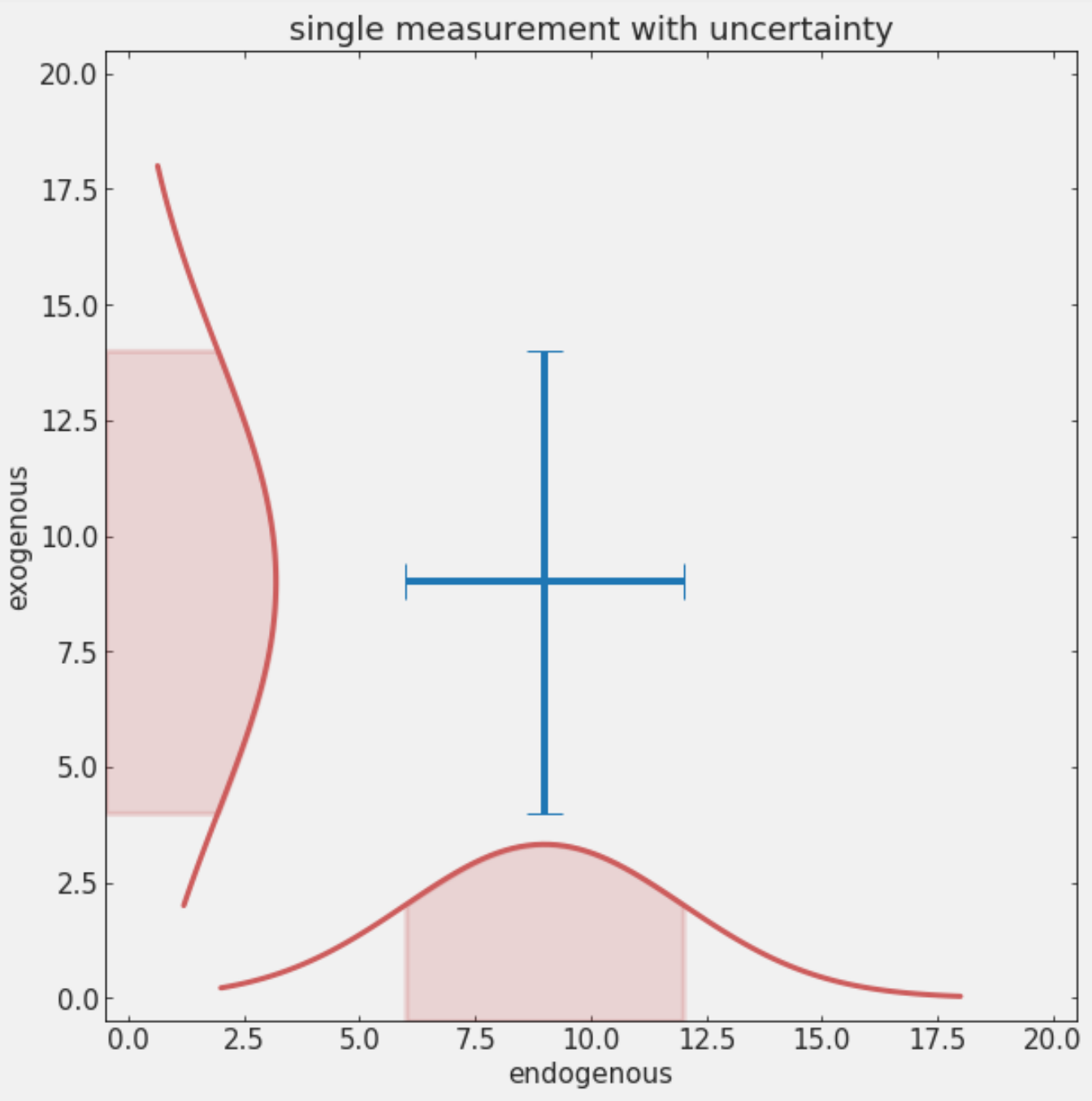
Accuracy



the meaning of "uncertainty"



the meaning of "uncertainty" when reporting a result



the meaning of
"uncertainty"
when reporting
a result

CMB cosmology

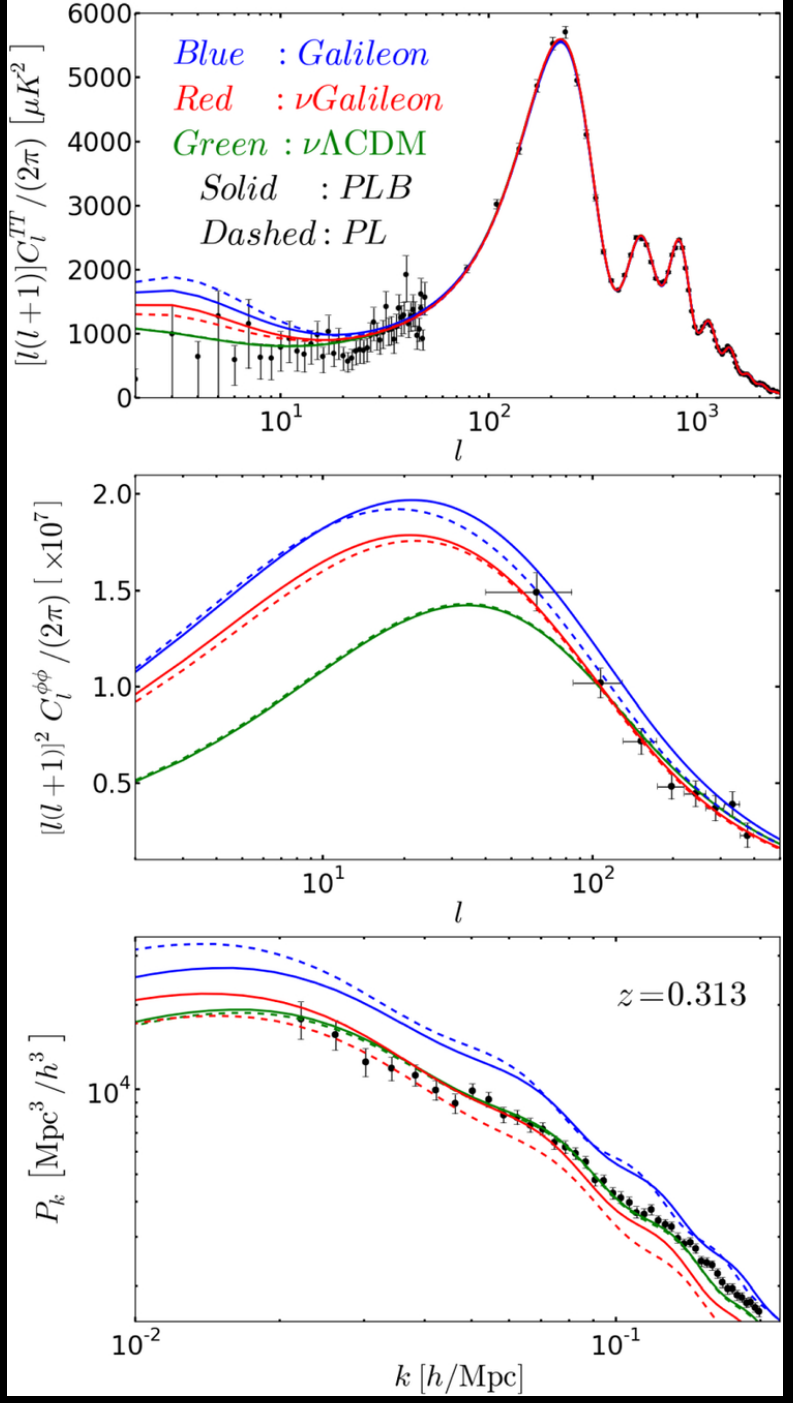
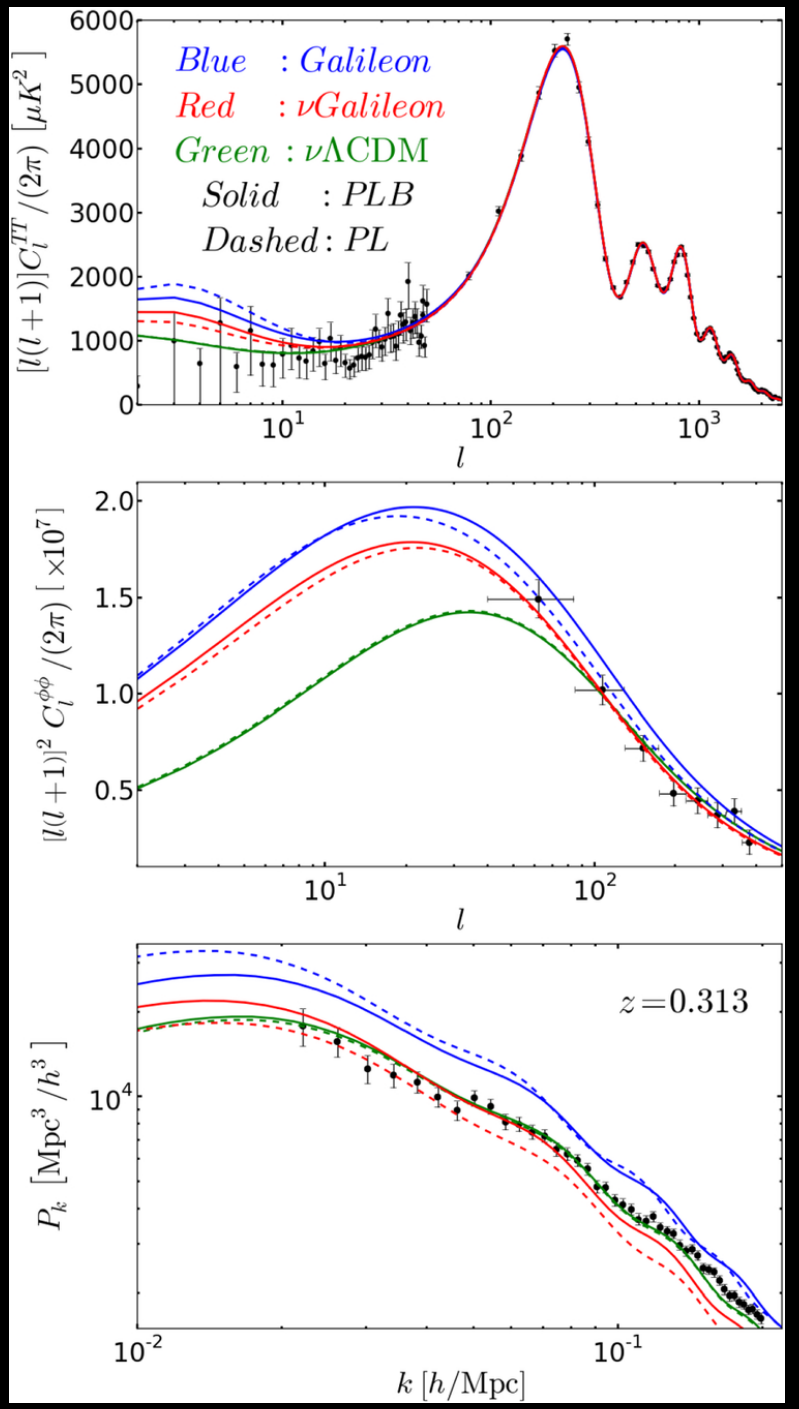


Fig 4: These plots illustrate the differences between ΛCDMΛCDM and Galileon models (see Sect. 7.3.1), with and without massive neutrinos. The Galileon models have background Friedmann equations that contain a scalar-field energy density contribution that generates late time cosmic acceleration and has an evolution consistent with observations and thus similar to that of a ΛCDMΛCDM 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).

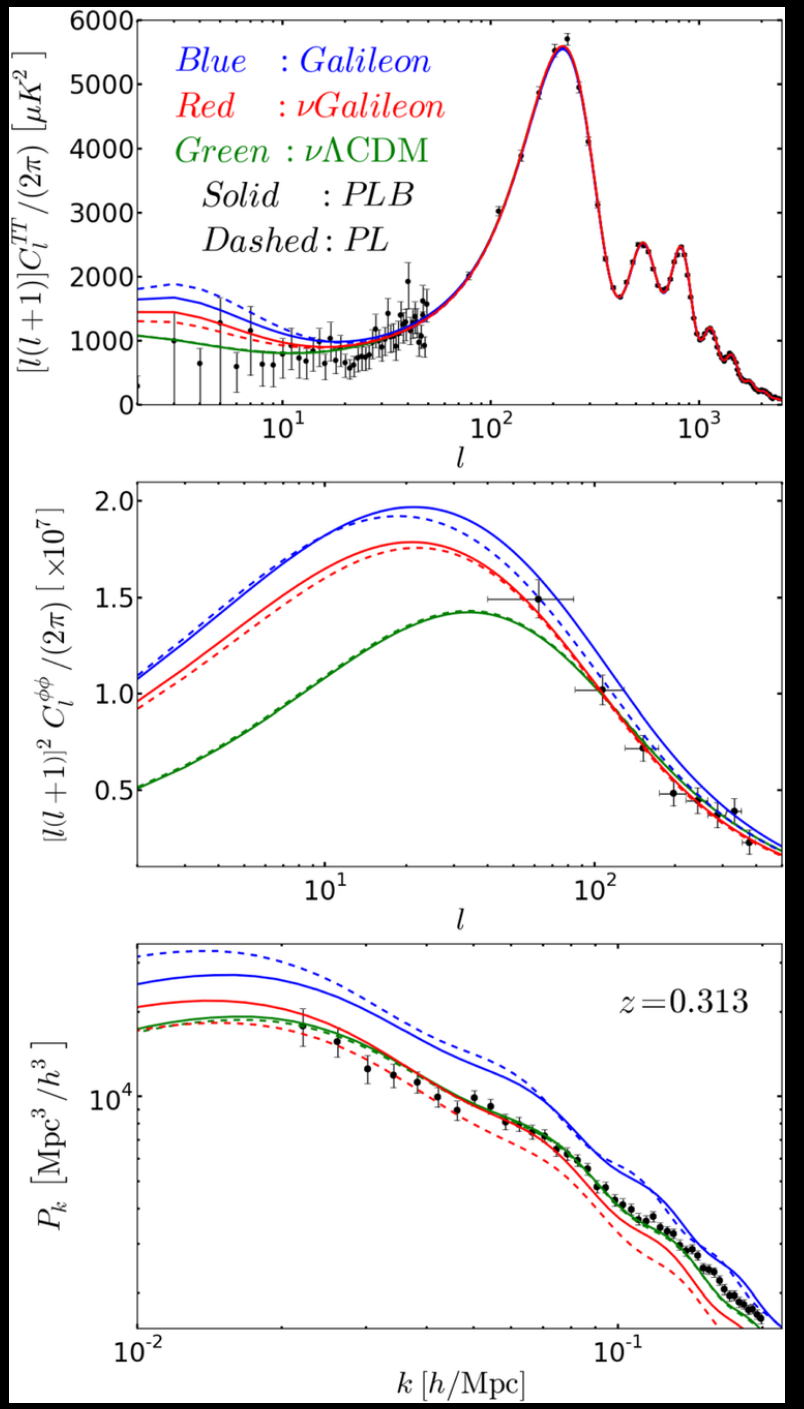
https://link.springer.com/article/10.1007/s41114-018-0017-4

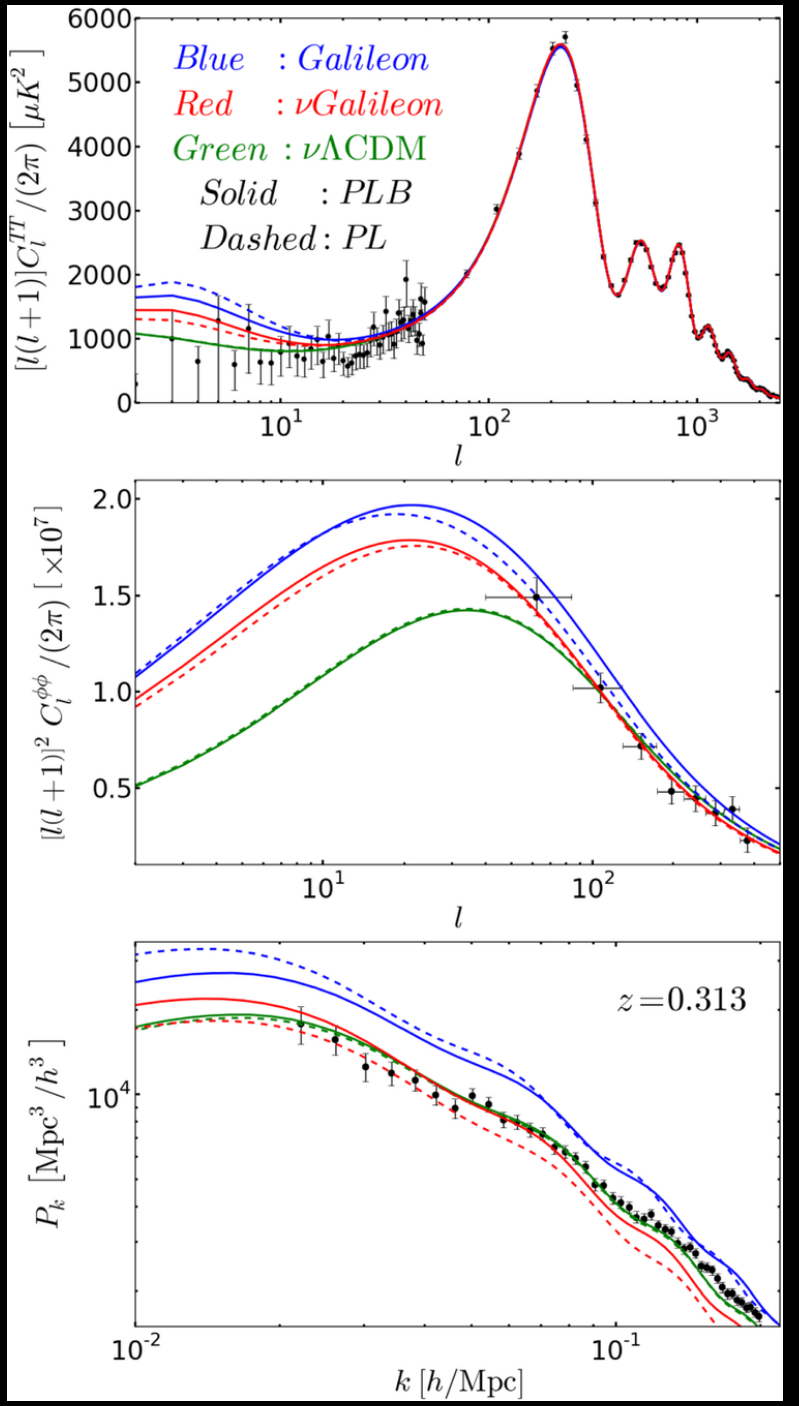


They note these as PL models. The solid lines indicate their best fits to CMB data (i.e., PL) plus BAO measurements from 6dF, SDSS DR7 and BOSS DR9. They note these as PLB models. The models correspond to best-fitting base Galileon modified gravity model (in blue), νGalileonνGalileon (in red) and $\nu\Lambda$CDM$\nu\Lambda$CDM (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 νGalileonνGalileon (PLB) model (Barreira et al. 2014a). We refer to this figure from various parts of the text

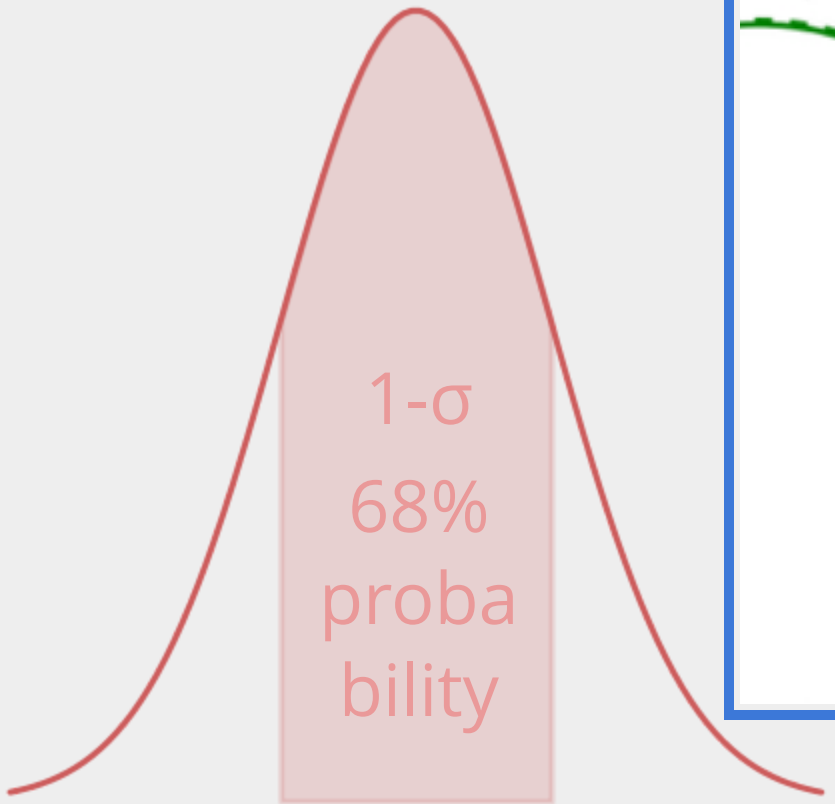
Image from Barreira et al. (2014).

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

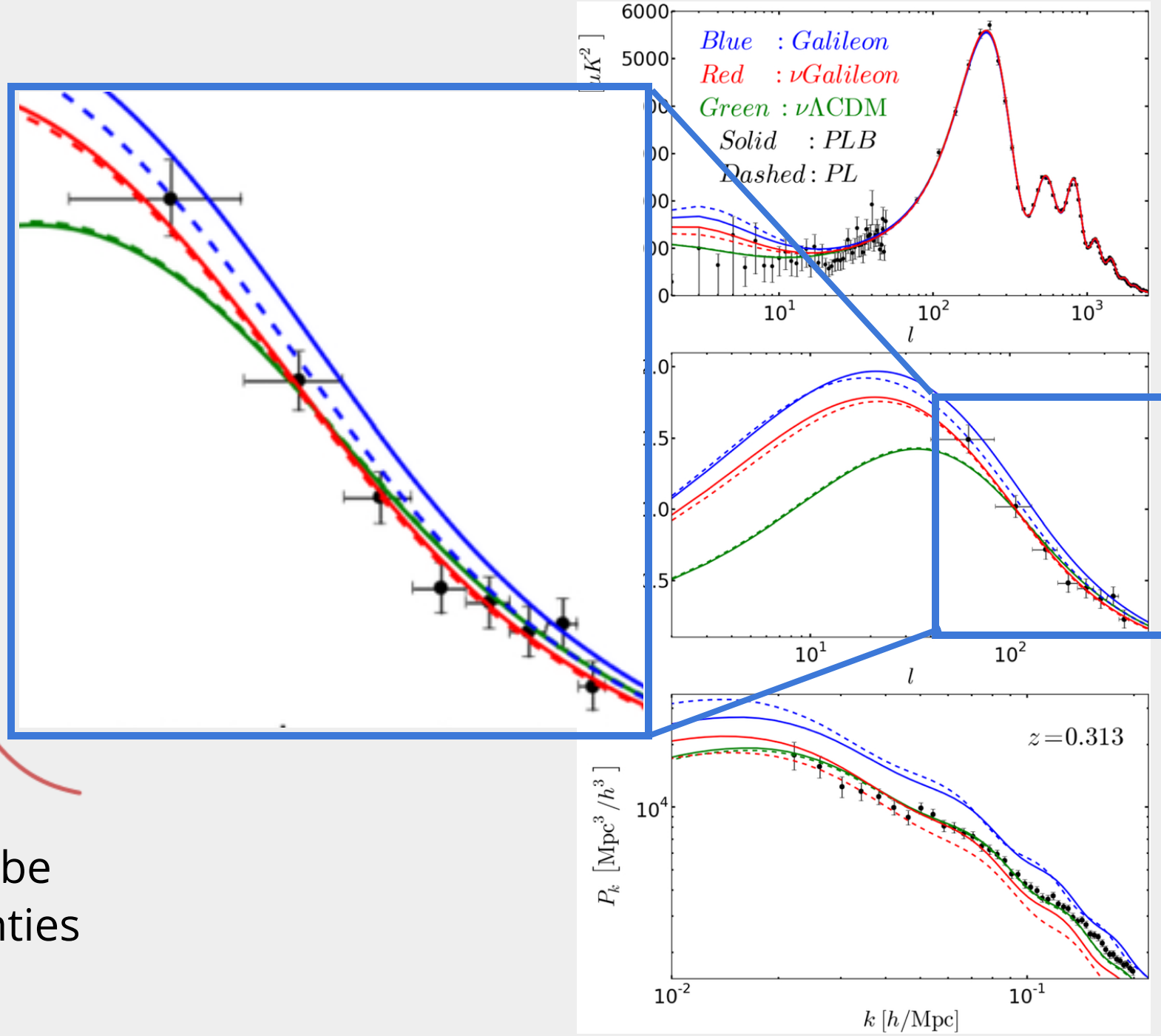



Blue : Galileon
Red : νGalileon
Green : νΛCDM
Solid : PLB
Dashed : PL
$[l(l+1)]C_l^{TT}/(2\pi)\ [\mu K^2]$
l
$[l(l+1)]^2\ C_l^{\phi\phi}/(2\pi)\ [\times 10^7]$
l
$z=0.313$
$P_k\ [\mathrm{Mpc}^3/h^3]$
$k\ [h/\mathrm{Mpc}]$

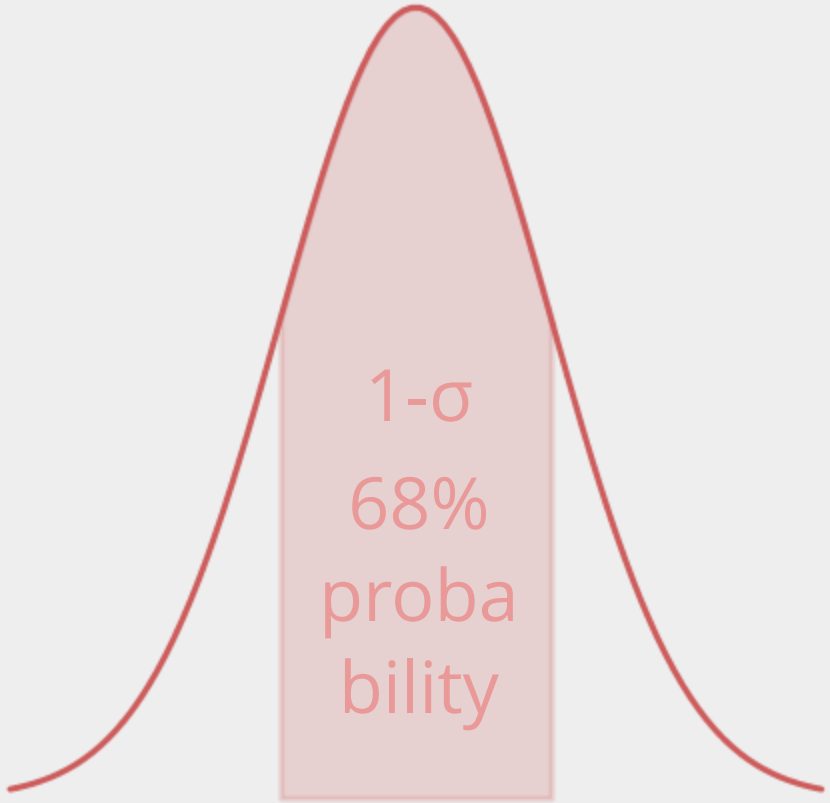
uncertainties: 1-σ



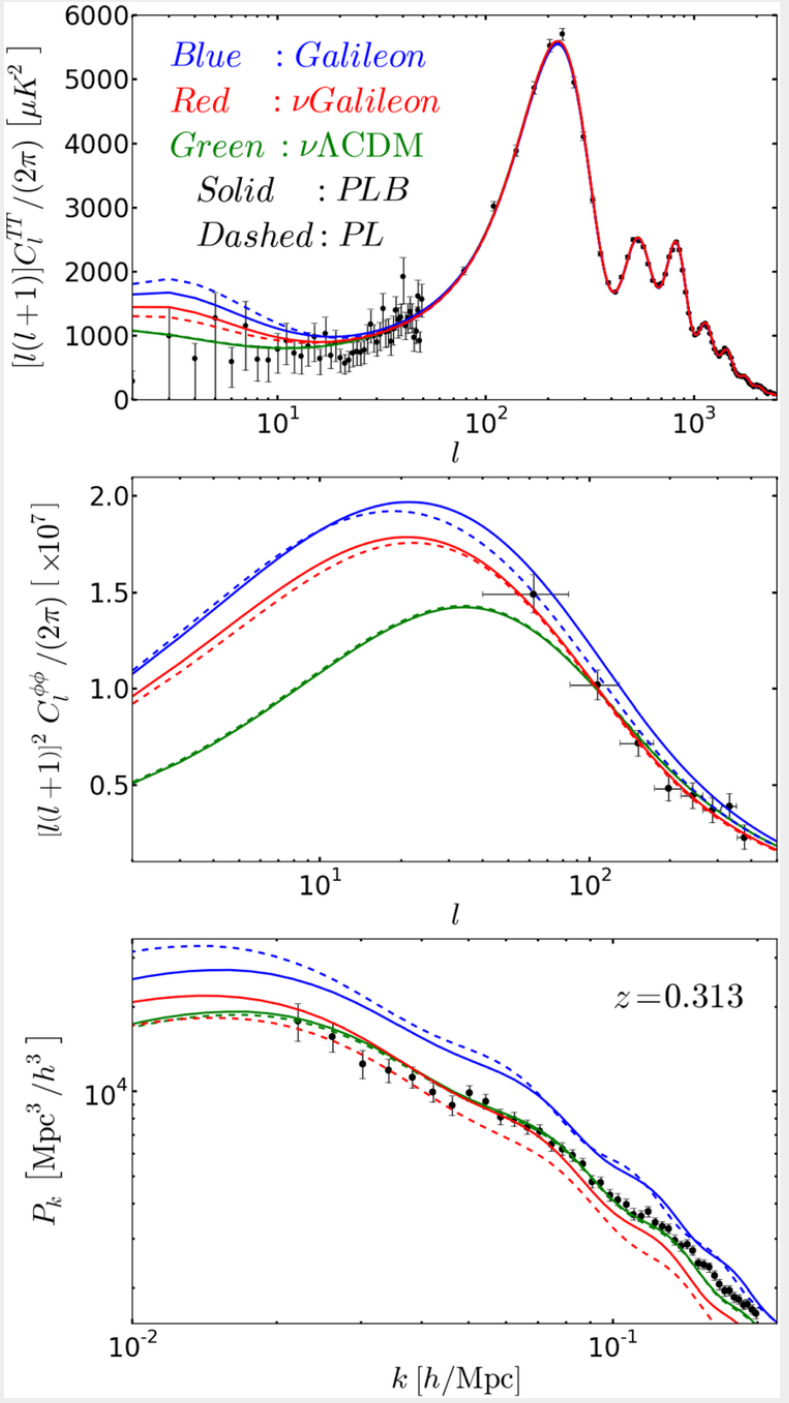
3 points out of 10 can be outside of the uncertainties



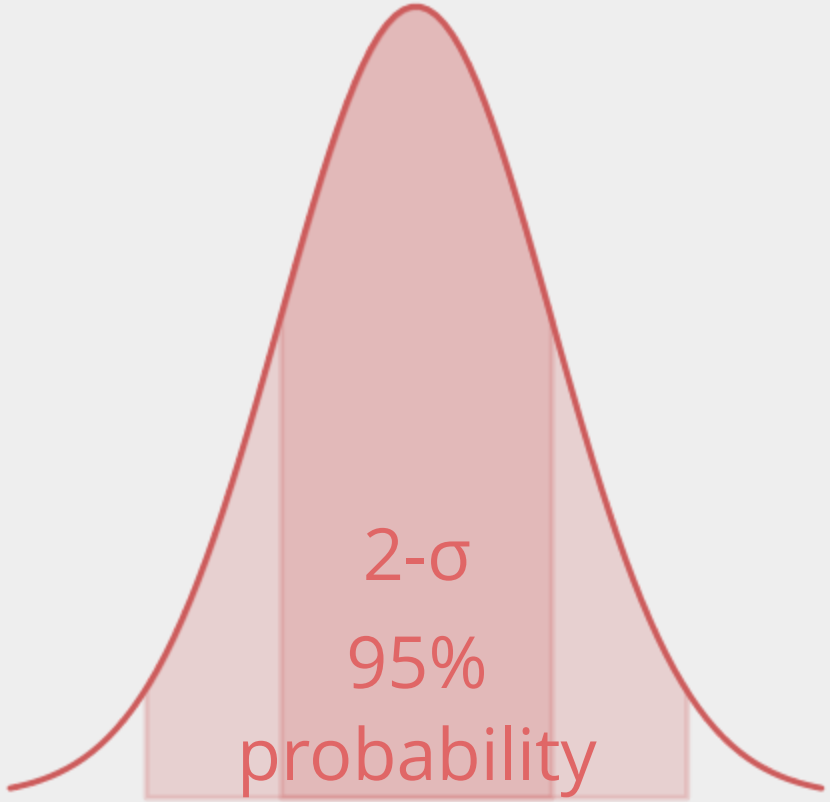
uncertainties: 1-σ



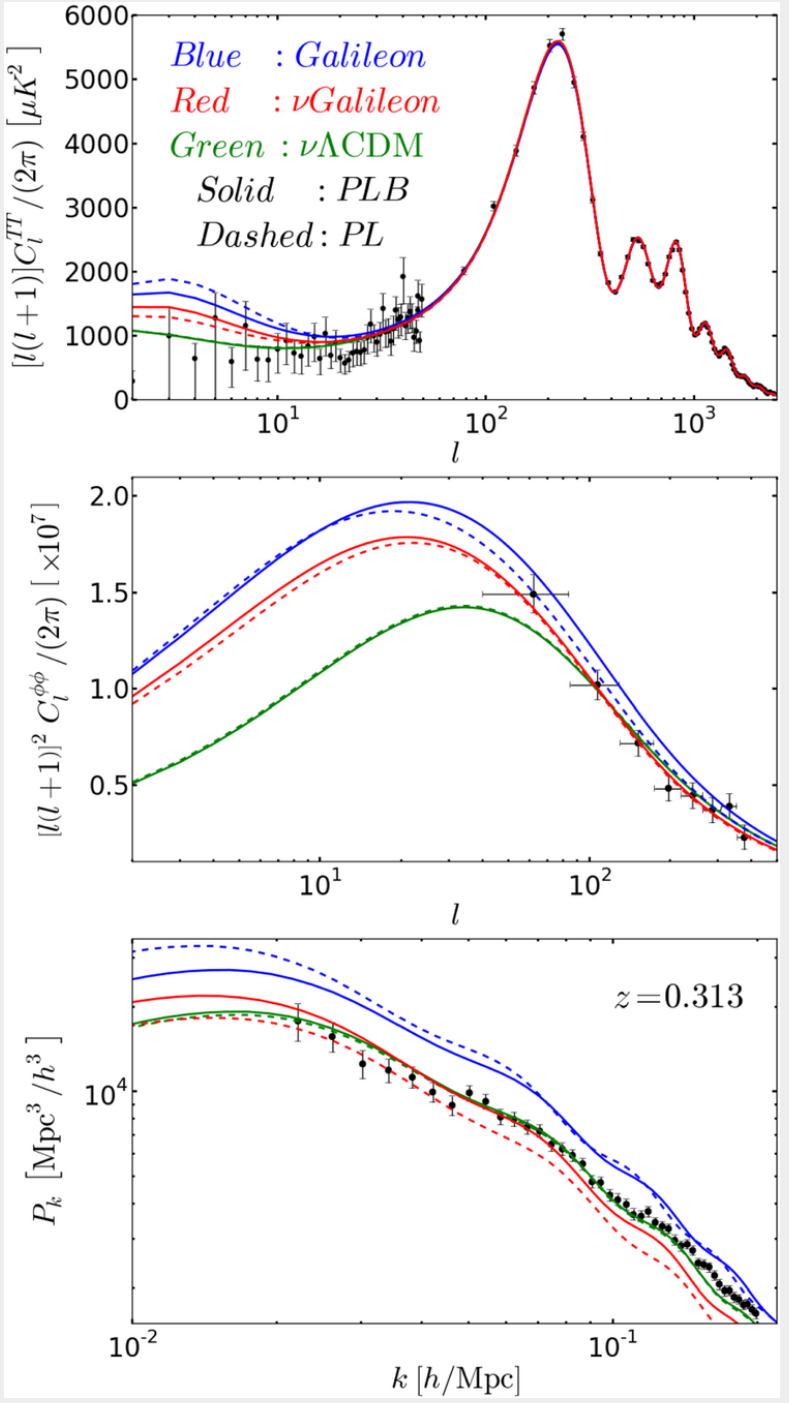
3 points out of 10 can be outside of the uncertainties



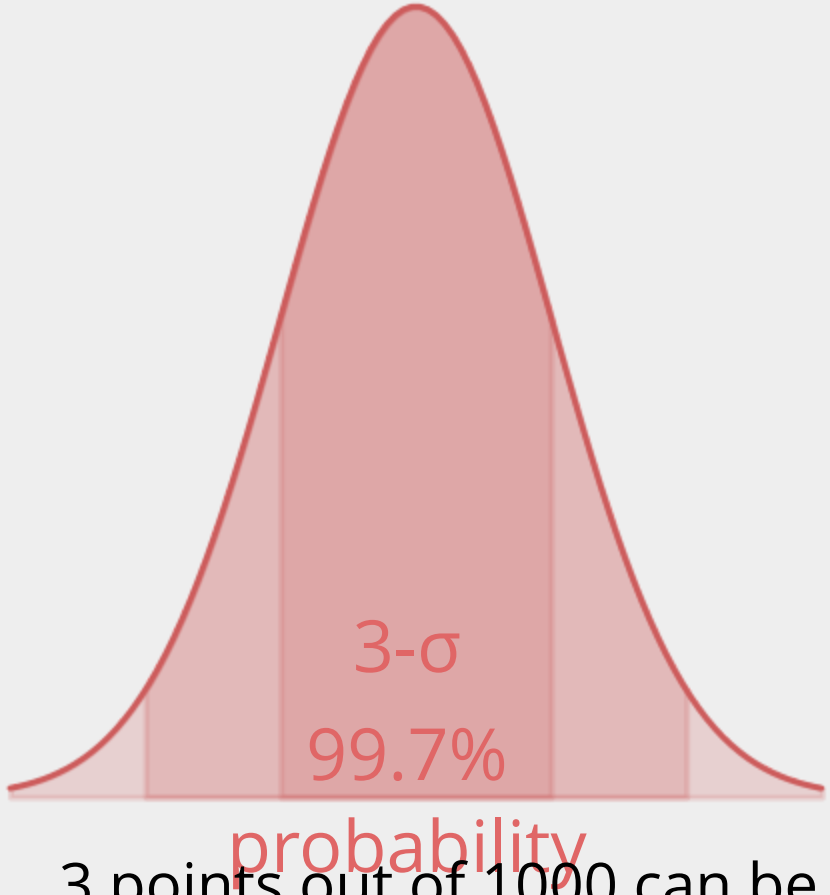
uncertainties: 1-σ



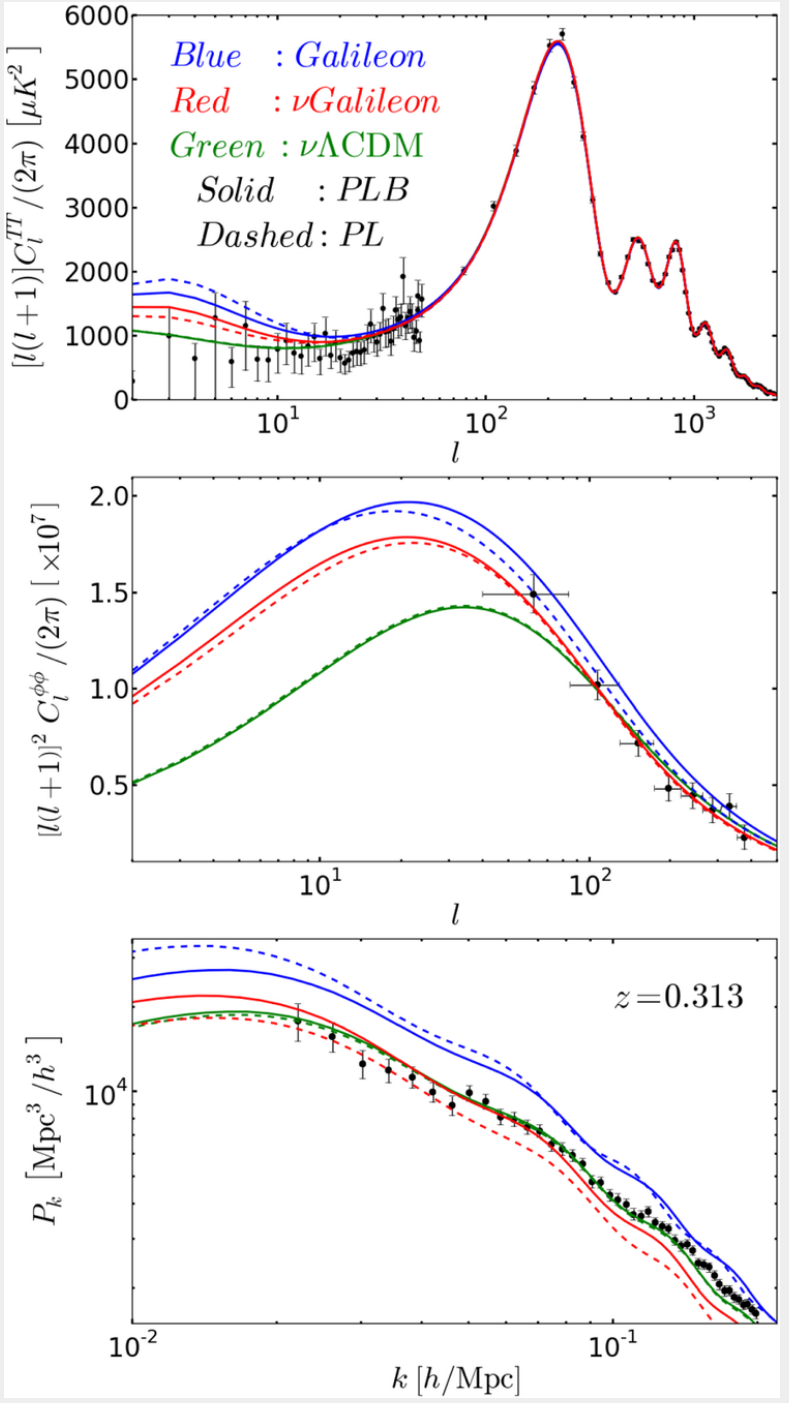
5 points out of 100 can be outside of the uncertainties



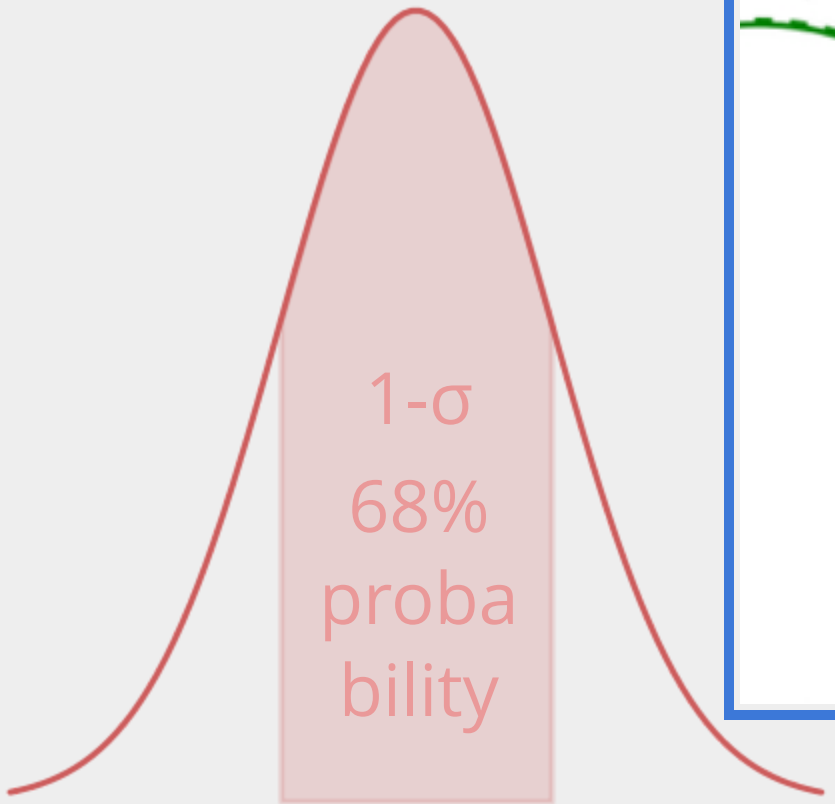
uncertainties: 1-σ



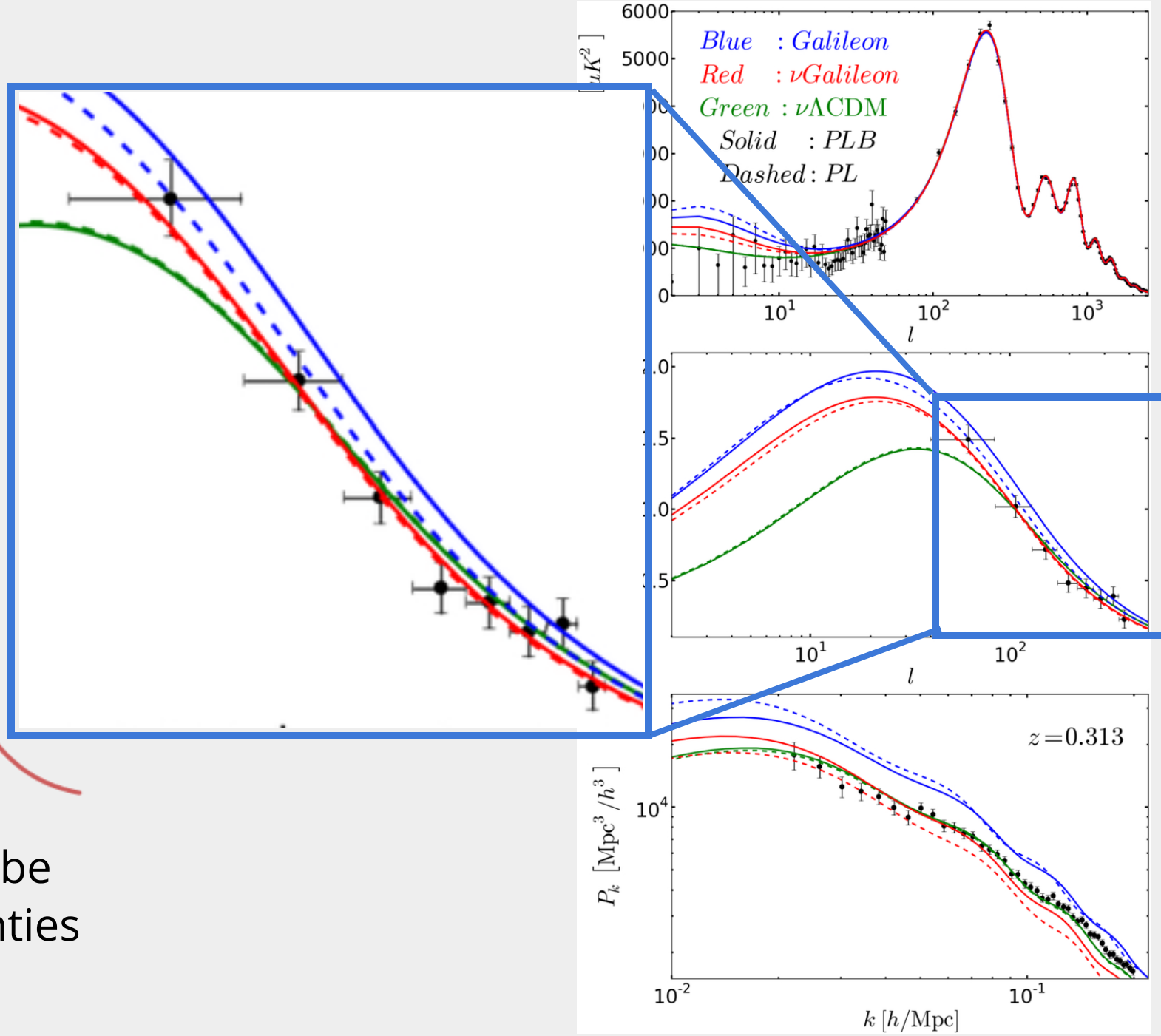
3 points out of 1000 can be outside of the uncertainties



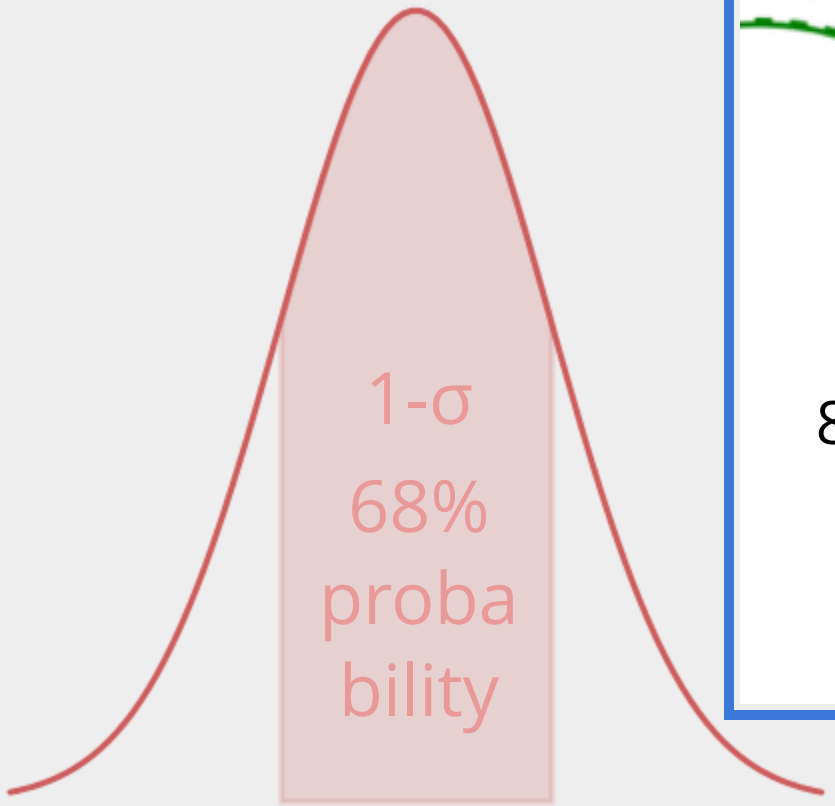
uncertainties: 1-σ



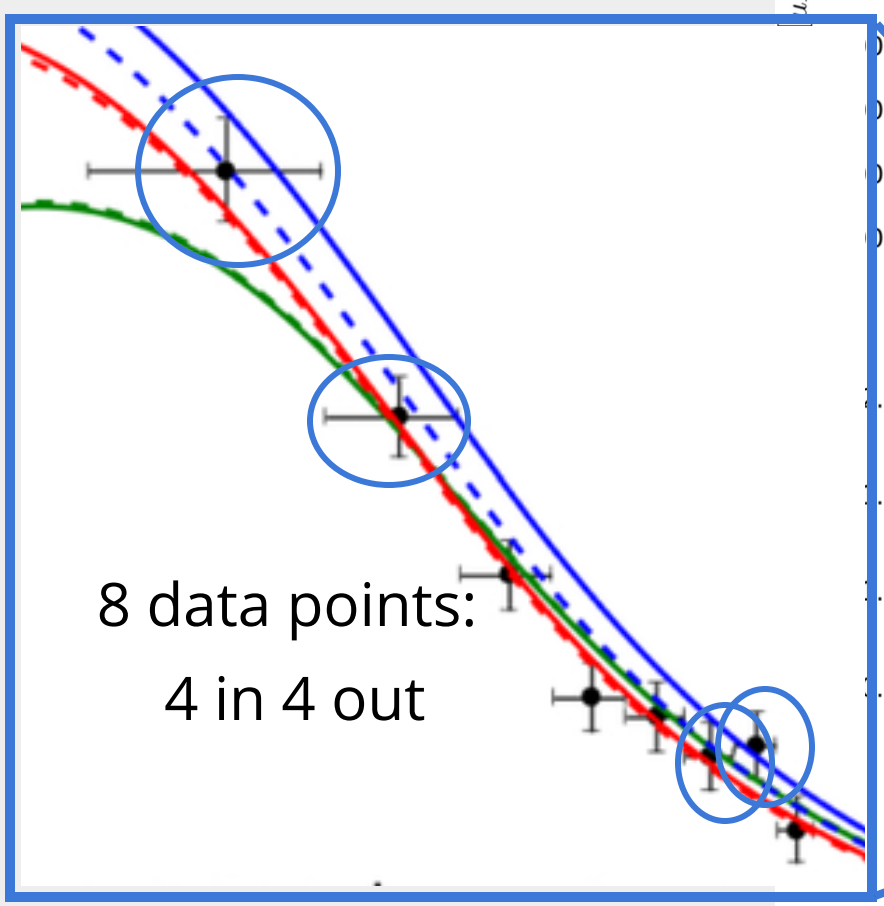
3 points out of 10 can be outside of the uncertainties



uncertainties: 1-σ

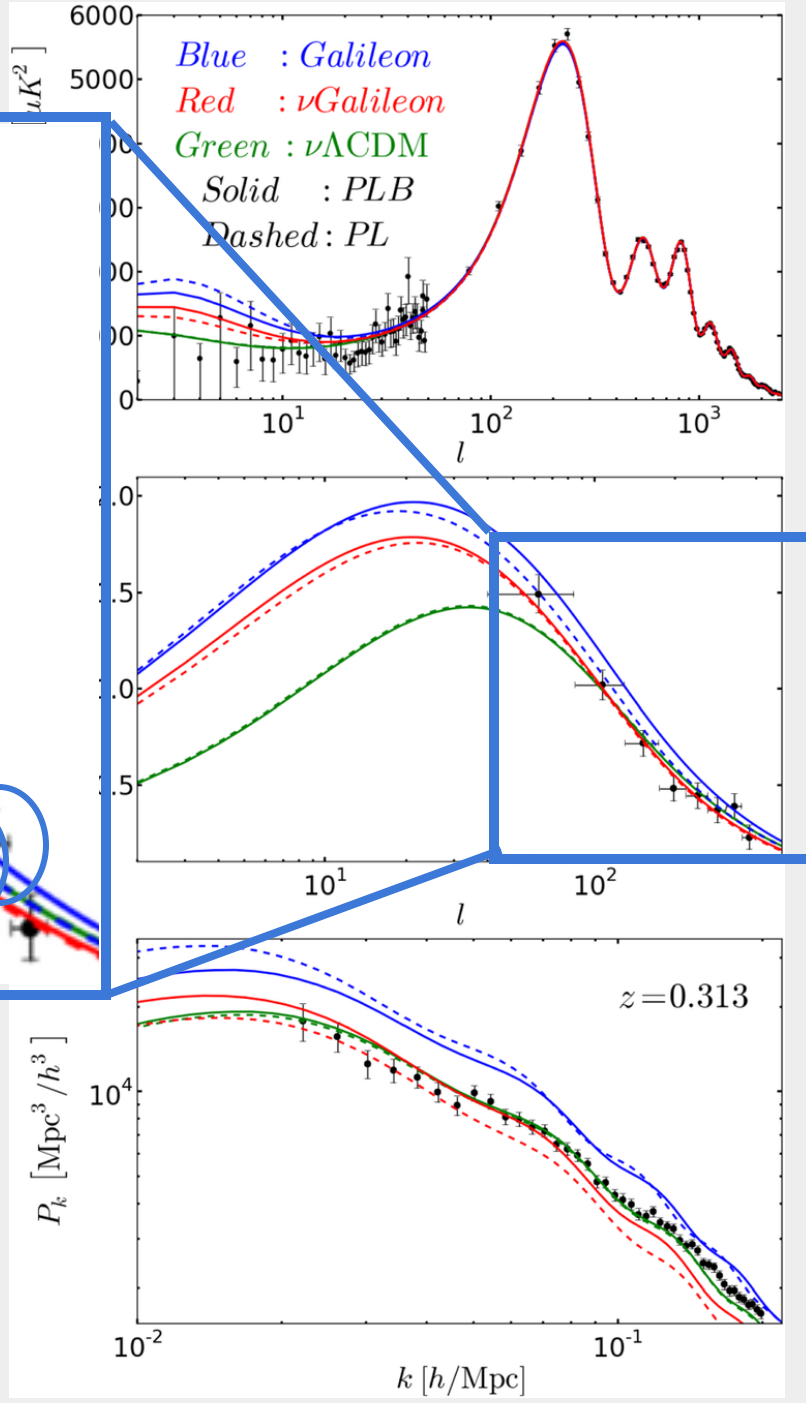


7 points out of 10 should be inside the errorbar

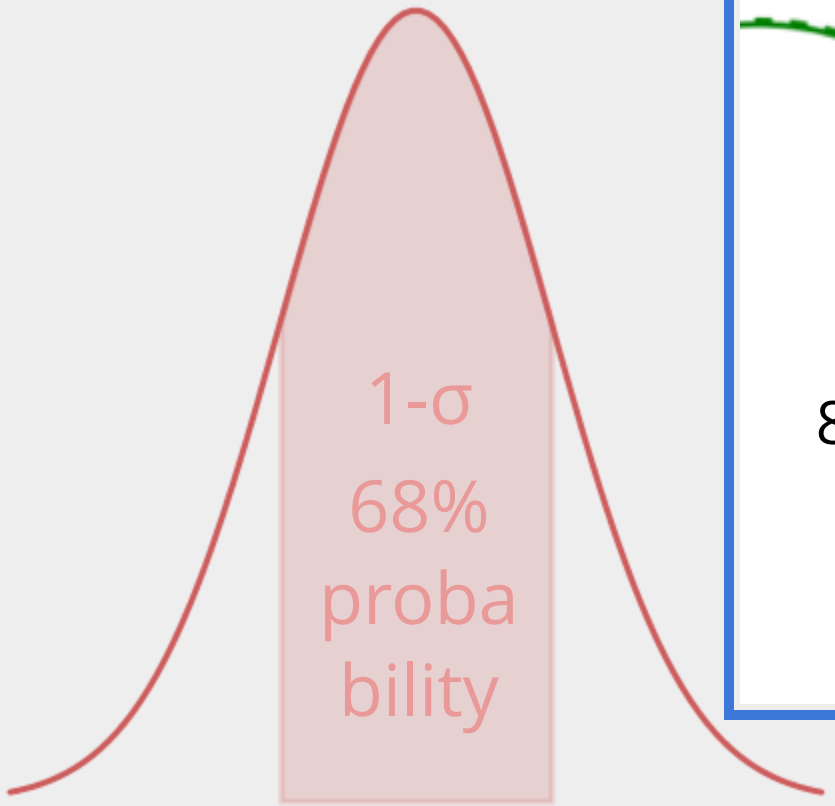


8 data points:

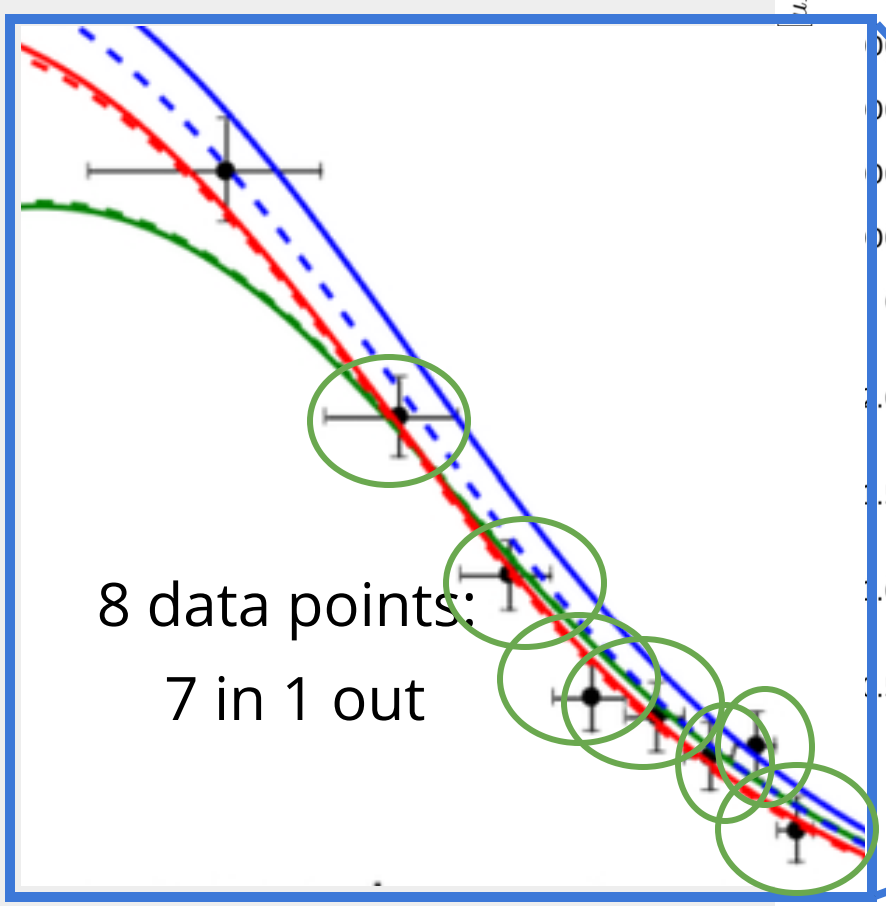
4 in 4 out



uncertainties: 1-σ

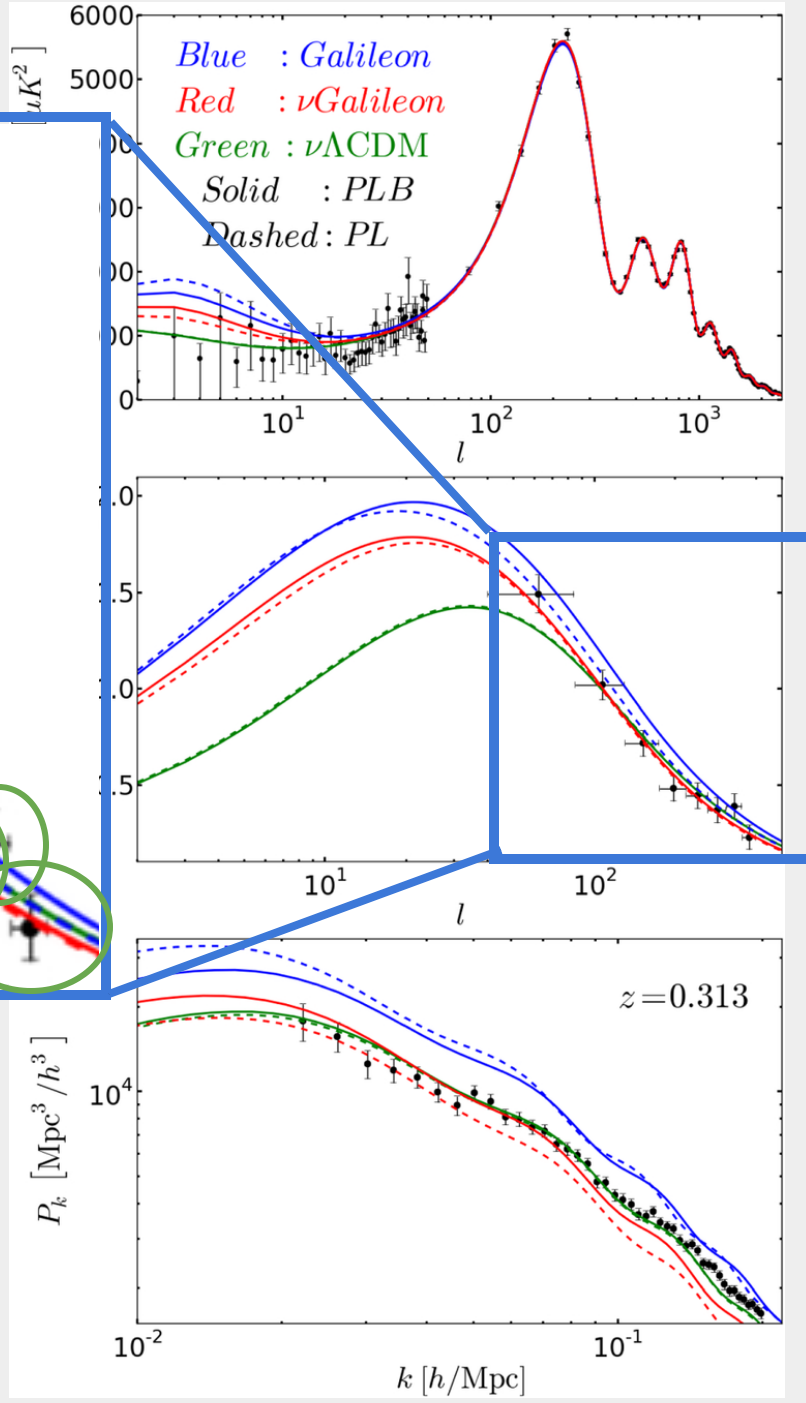


7 points out of 10 should be inside the errorbar

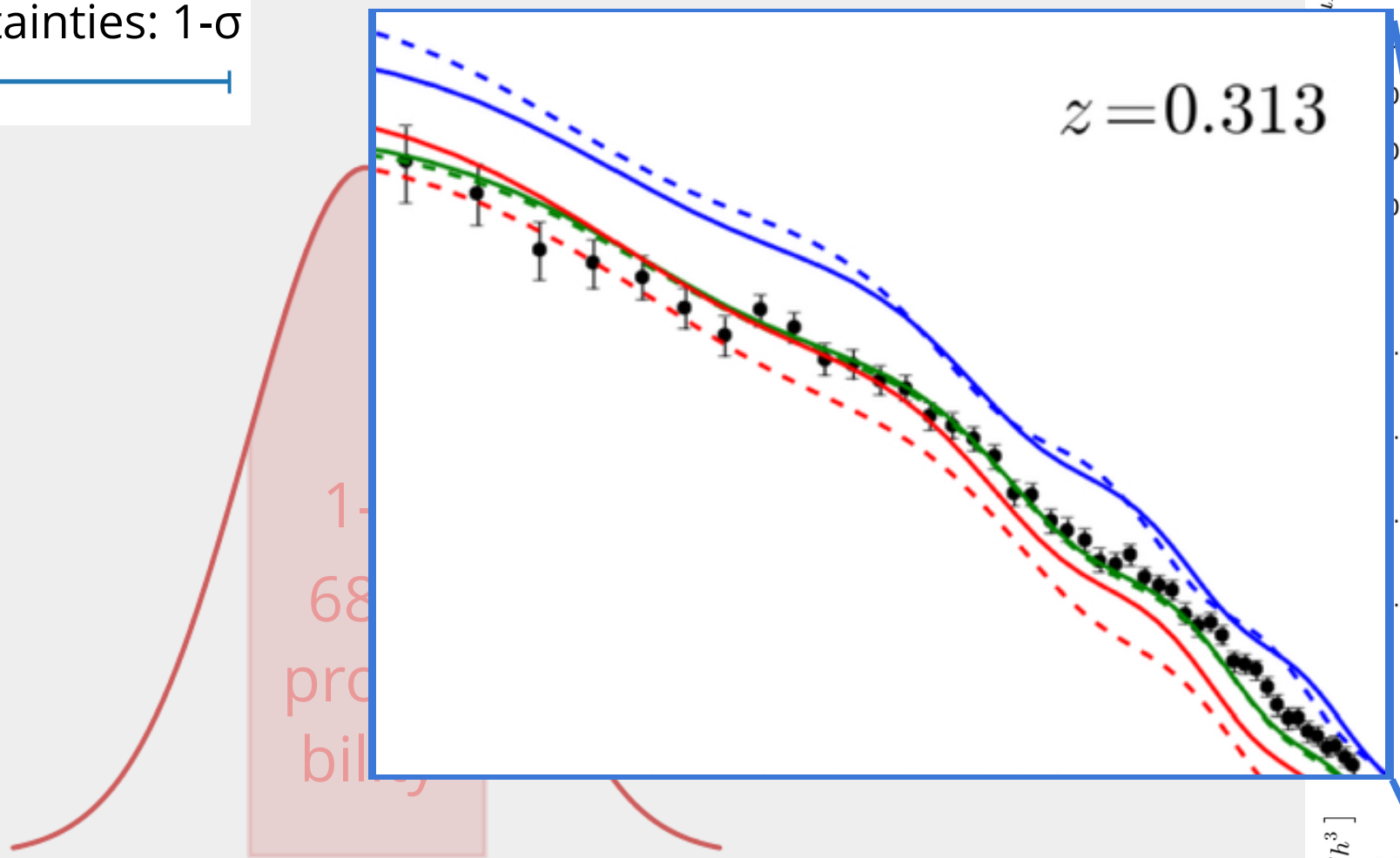


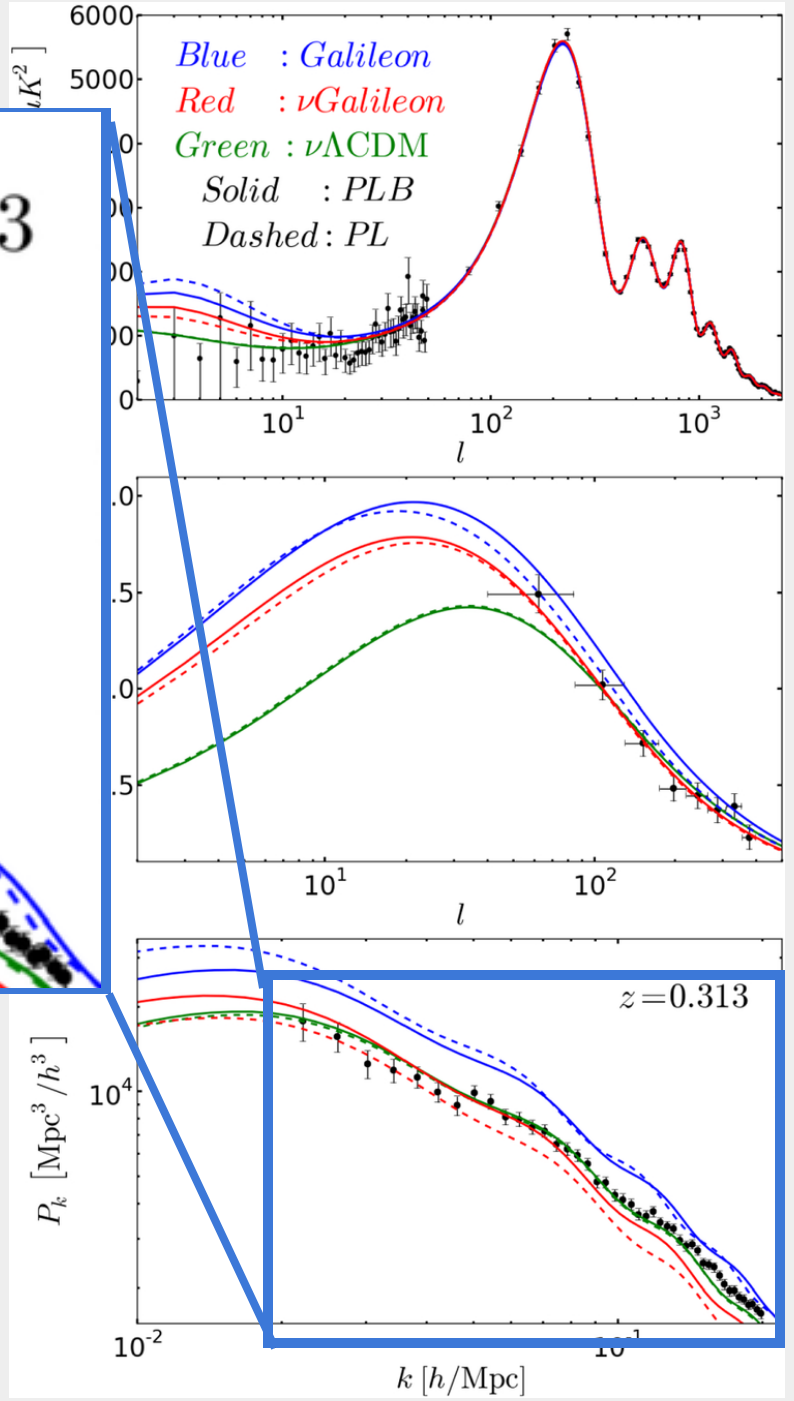
8 data points:

7 in 1 out



uncertainties: 1-σ





7 points out of 10 should be inside the errorbar

5

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:

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

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

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

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

6

Model fit:

Optimization and Likelihood

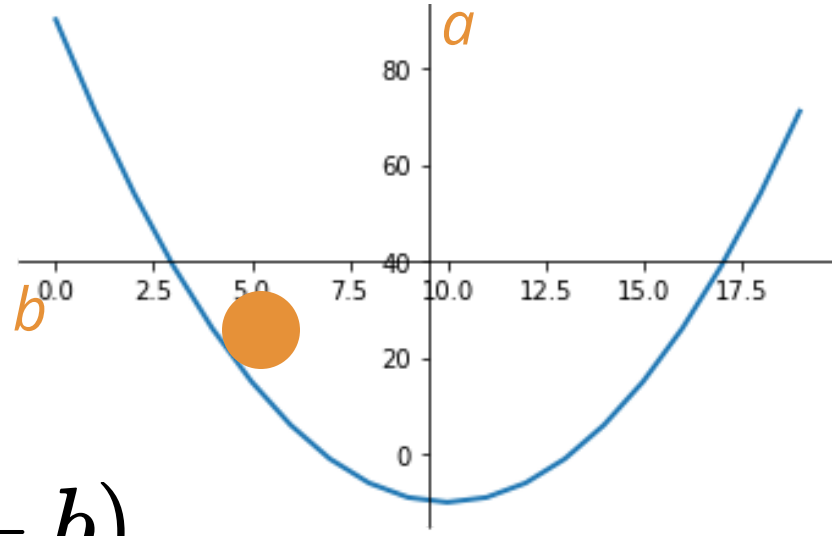
Optimization & Likelihood

change *a*, *b* to minimize χ2

Linear regression $y_i = (a\, x_i + b)$

Target Funcion $\chi^2 = \sum_{i=1}^{N} \left(\frac{y_i - (a\, x_i + b))^2}{\sigma^2} \right)$

Optimization & Likelihood



Univariate Linear regression $y_i = (a\, x_i + b)$

Target Funcion $\chi^2 = \sum_{i=1}^{N} \left(\frac{y_i - (a\, x_i + b))^2}{\sigma^2} \right)$

Data analysis recipes:
Fitting a model to data*

David W. Hogg
Center for Cosmology and Particle Physics, Department of Physics, New York University
Max-Planck-Institut für Astronomie, Heidelberg

Jo Bovy
Center for Cosmology and Particle Physics, Department of Physics, New York University

Dustin Lang
Department of Computer Science, University of Toronto
Princeton University Observatory

Gaussian

$$N \sim \frac{1}{\sqrt{2\pi\sigma}} exp \frac{-(x-\mu)^2}{2\sigma^2}$$

https://arxiv.org/pdf/1008.4686.pdf

In the case of the straight line fit in the presence of known, Gaussian uncertainties in one dimension, one can create this generative model as follows: Imagine that the data *really do* come from a line of the form $y = f(x) = m\,x + b$, and that the only reason that any data point deviates from this perfect, narrow, straight line is that to each of the true y values a small y-direction offset has been added, where that offset was drawn from a Gaussian distribution of zero mean and known variance σ_y^2. In this model, given an independent position x_i, an uncertainty σ_{yi}, a slope m, and an intercept b, the frequency distribution $p(y_i|x_i, \sigma_{yi}, m, b)$ for y_i is

$$p(y_i|x_i, \sigma_{yi}, m, b) = \frac{1}{\sqrt{2\,\pi\,\sigma_{yi}^2}} \exp\left(-\frac{[y_i - m\,x_i - b]^2}{2\,\sigma_{yi}^2}\right) \quad , \qquad (9)$$

where this gives the expected frequency (in a hypothetical set of repeated experiments[13]) of getting a value in the infinitesimal range $[y_i, y_i + \mathrm{d}y]$ per unit $\mathrm{d}y$.

The generative model provides us with a natural, justified, scalar objective: We seek the line (parameters m and b) that maximize the probability of the observed data given the model or (in standard parlance) the *likelihood of the parameters.*[14] In our generative model the data points are independently drawn (implicitly), so the likelihood $\mathscr{L}$ is the product of conditional probabilities

$$\mathscr{L} = \prod_{i=1}^{N} p(y_i|x_i, \sigma_{yi}, m, b) \quad . \qquad (10)$$

Likelihood

Probability of data given a model, e.g. Gaussian

unknown variables

$$P(\vec{x}|\mu,\sigma) \sim \frac{1}{\sqrt{2\pi}\sigma} exp \frac{-(\vec{x}-\mu)^2}{2\sigma^2}$$

Likelihood: the probability of the model given the data. Same Gaussian assumptions

$$L(\mu,\sigma|\vec{x}) \sim \frac{1}{\sqrt{2\pi}\sigma} exp \frac{-(\vec{x}-\mu)^2}{2\sigma^2}$$

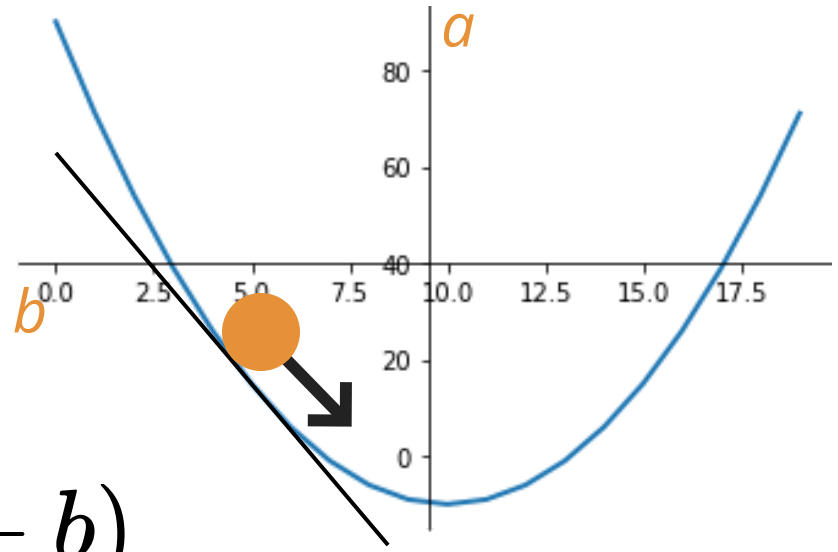
unknown variables

Maximizing the likelihood we seek the parameters that maximize the probability of the *observed* data under the chosen model

Optimization

Gradient descent is an iterative algorithm, that starts from a random point on a function and travels down its slope in steps until it reaches the lowest point of that function.

(*Toward data science*)



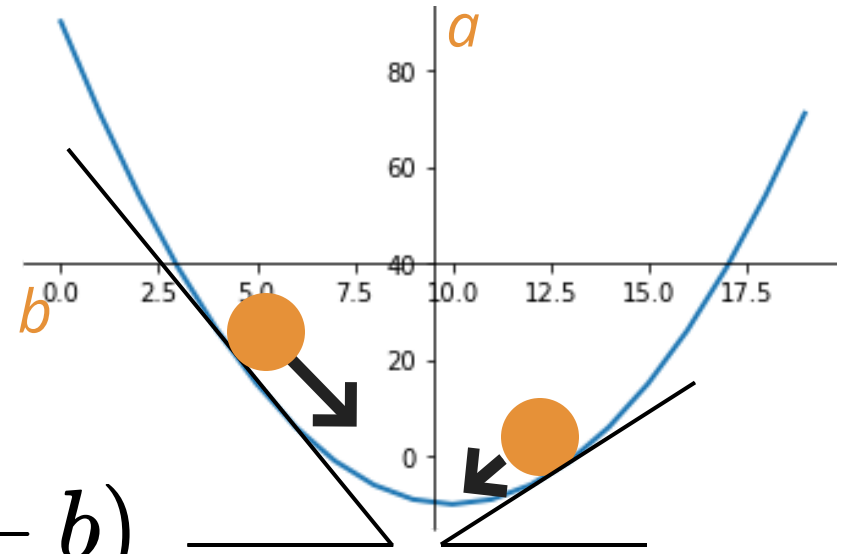
Univariate Linear regression $y_i = (a\, x_i + b)$

Target Funcion $\chi^2 = \sum_{i=1}^{N} \left(\frac{(y_i - (a\, x_i + b))^2}{\sigma_i^2} \right)$

Optimization

Gradient Descent

$$p_{\text{new}} := p_{\text{old}} - \eta \nabla Q(p)$$



Univariate Linear regression $y_i = (a\, x_i + b)$

Target Funcion $\chi^2 = \sum_{i=1}^{N} \left(\frac{(y_i - (a\, x_i + b))^2}{\sigma_i^2} \right)$

Optimization

Q : target function

p : parameters

η : learningrate

ϵ : tolerance

$$p_{\text{new}} := p_{\text{old}} - \eta \nabla Q(p)$$

"convergence" is reached when the gradient is

~0: with ε tollrance

$$\Delta Q(p_{\text{final}}) \in [-\epsilon, \epsilon]$$

gradient descent algorithm

1. Choose a target function ***Q(p)*** of the parameters ***p***
2. Choose a (random) initial value for the parameters: (e.g. $p_0 = (a_0, b_0)$)
3. Choose a learning rate ***η*** (this could be a multidimensional vector η_i setting a different learning rate for different features)

   Repeat steps 4, 5, 6 until "convergence":
4. Calculate the gradient ***Q'*** of the target function for the current parameter values ***calculating it over all observations in the training set***
5. Calculate the next step sizes for each feature : **stepsize = *Q'(p_now) * η***
6. Calculate the new parameters ***p_new*** as : ***p_new = p_now - stepsize***

Optimization

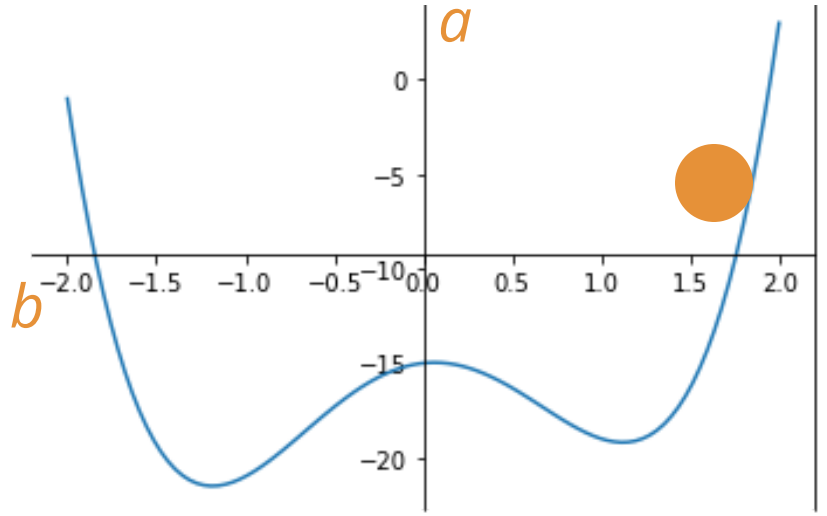
Gradient Descent

$$p_{\text{new}} = p_{\text{old}} - \sum_j \eta_j \sum_{i=1}^{N} \frac{df'(x_{i,j})}{dx_{i,j}}$$

Univariate Linear regression $y_i = (a\, x_i + b)$

Target Funcion $\chi^2 = \sum_{i=1}^{N} \left(\frac{(y_i - (a\, x_i + b))^2}{\sigma_i^2} \right)$

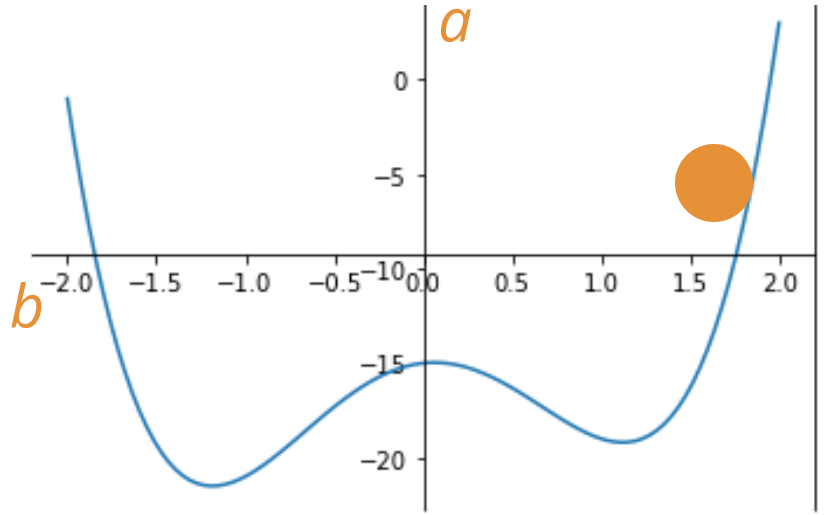
Optimization



Target Funcion $\chi^2 = \sum_{i=1}^{N} \left(\frac{(y_i - (a\, x_i + b))^2}{\sigma_i^2} \right)$

Optimization

Add stochasticity to avoid getting stuck in a local minimum



Target Funcion $\chi^2 = \sum_{i=1}^{N} \left(\frac{(y_i - (a\, x_i + b))^2}{\sigma_i^2} \right)$

Optimization

Q : target function

p : parameters

η : learning rate

ϵ : tolerance

$$p_{\text{new}} := p_{\text{old}} - \eta \nabla Q(p)$$

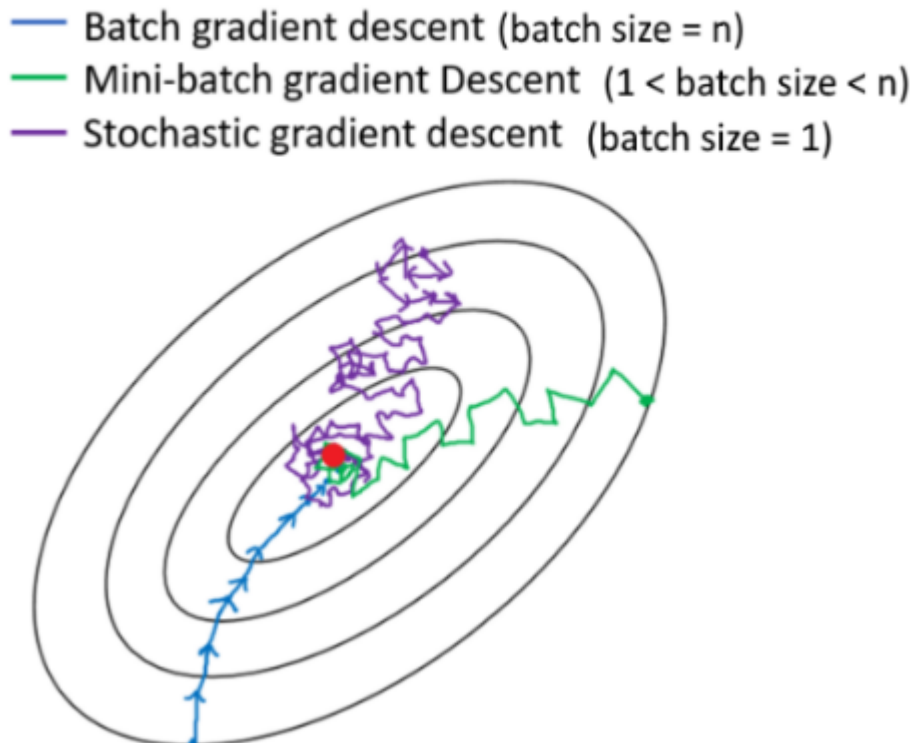
"convergence" is reached when the gradient is

~0: with ε tollrance

$$\Delta Q(p_{\text{final}}) \in [-\epsilon, \epsilon]$$

stochastic gradient descent algorithm

https://datascience.stackexchange.com/questions/52884/possible-for-batch-size-of-neural-network-to-be-too-small



Optimization

Q : target function

p : parameters

η : learning rate

ϵ : tolerance

$$p_{\text{new}} := p_{\text{old}} - \eta \nabla Q(p)$$

"convergence" is reached when the gradient is

~0: with ε tollrance

$$\Delta Q(p_{\text{final}}) \in [-\epsilon, \epsilon]$$

stochastic gradient descent algorithm

1. Choose a target function ***Q(p)*** of the parameters ***p***
2. Choose a (random) initial value for the parameters: (e.g. $p_0 = (a_0, b_0)$)
3. Choose a learning rate ***η*** (this could be a multidimensional vector η_i setting a different learning rate for different features)

   Repeat steps 4, 5, 6 until "convergence":
4. Calculate the gradient ***Q'*** of the target function for the current parameter values ***on a subset of the observations (extreme: size=1)***
5. Calculate the next step sizes for each feature : **stepsize = *Q'(p_now) * η***
6. Calculate the new parameters ***p_new*** as : ***p_new = p_now - stepsize***

Optimization

gradient descent algorithm

1. Choose a target function ***Q(p)*** of the parameters ***p***
2. Choose a (random) initial value for the parameters: (e.g. $p_0 = (a_0, b_0)$)
3. Choose a learning rate η (this could be a multidimensional vector η_i setting a different learning rate for different features)

   Repeat steps 4, 5, 6 until "convergence":

4. Calculate the gradient ***Q'*** of the target function for the current parameter values ***calculating it over all observations in the training set***
5. Calculate the next step sizes for each feature :
   stepsize = *Q'(p_now) * η*
6. Calculate the new parameters ***p_new*** as :
   p_new = p_now - stepsize

stochastic gradient descent algorithm

1. Choose a target function ***Q(p)*** of the parameters ***p***
2. Choose a (random) initial value for the parameters: (e.g. $p_0 = (a_0, b_0)$)
3. Choose a learning rate η (this could be a multidimensional vector η_i setting a different learning rate for different features)

   Repeat steps 4, 5, 6 until "convergence":

4. Calculate the gradient ***Q'*** of the target function for the current parameter values ***on a subset of the observations (extreme: size=1)***
5. Calculate the next step sizes for each feature :
   stepsize = *Q'(p_now) * η*
6. Calculate the new parameters ***p_new*** as :
   p_new = p_now - stepsize

Optimization

Stochastic Gradient Descent

$$p_{\text{new}} = p_{\text{old}} - \sum_j \eta_j \frac{df'(x_{i,j})}{dx_{i,j}}_{i \in N}$$

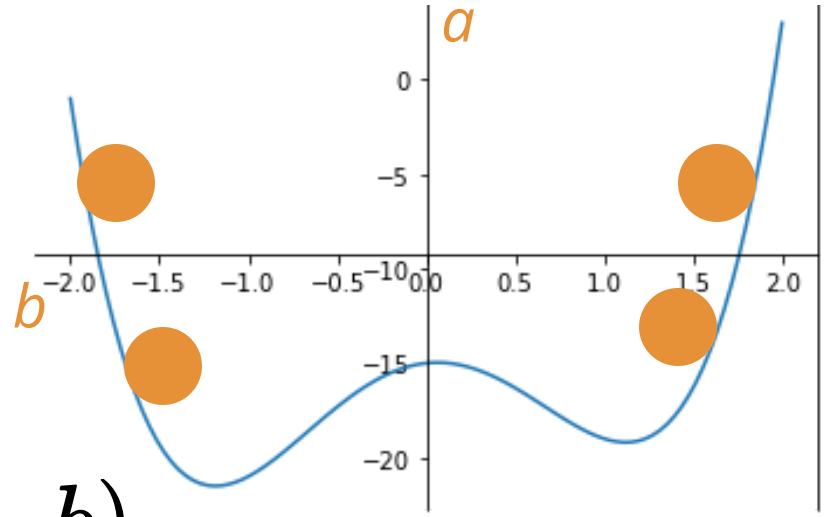
where *i* 1 elements of the full N-dimensional observation set (a subset)

Univariate Linear regression $y_i = (a\, x_i + b)$

Target Funcion $\chi^2 = \sum_{i=1}^{N} \left(\frac{(y_i - (a\, x_i + b))^2}{\sigma_i^2} \right)$

Optimization

idea 2. start a bunch of parallel optimizations



Univariate Linear regression $y_i = (a\, x_i + b)$

Target Funcion $\chi^2 = \sum_{i=1}^{N} \left(\frac{(y_i - (a\, x_i + b))^2}{\sigma_i^2} \right)$

Bayes theorem

$$P(\text{model}|\text{data})P(\text{data}) = P(\text{data}|\text{model})P(\text{model})$$

$$P(\text{model}|\text{data}) = \frac{P(\text{data}|\text{model})P(\text{model})}{P(\text{data})}$$

Bayes theorem

$$P(\theta|\mathrm{D}) = \frac{P(\mathrm{D}|\theta)P(\theta)}{P(\mathrm{D})}$$

posterior: joint probability distributin of a parameter set (θ, e.g. (m, b)) condition upon some data D and a model hypothesys f

prior: "intellectual" knowledge about the model parameters condition on a model hypothesys *f. This should come from domain knowledge or knowledge of data that is not the dataset under examination*

evidence: marginal likelihood of data under the model

in reality all of these quantities are conditioned on the shape of the model: this is a model fitting, not a model selection methodology $P(D|f) = \int_{-\infty}^{\infty} P(D|\theta, f)P(\theta|f)d\theta$

6 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

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

Canfield Solitaire

"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**"

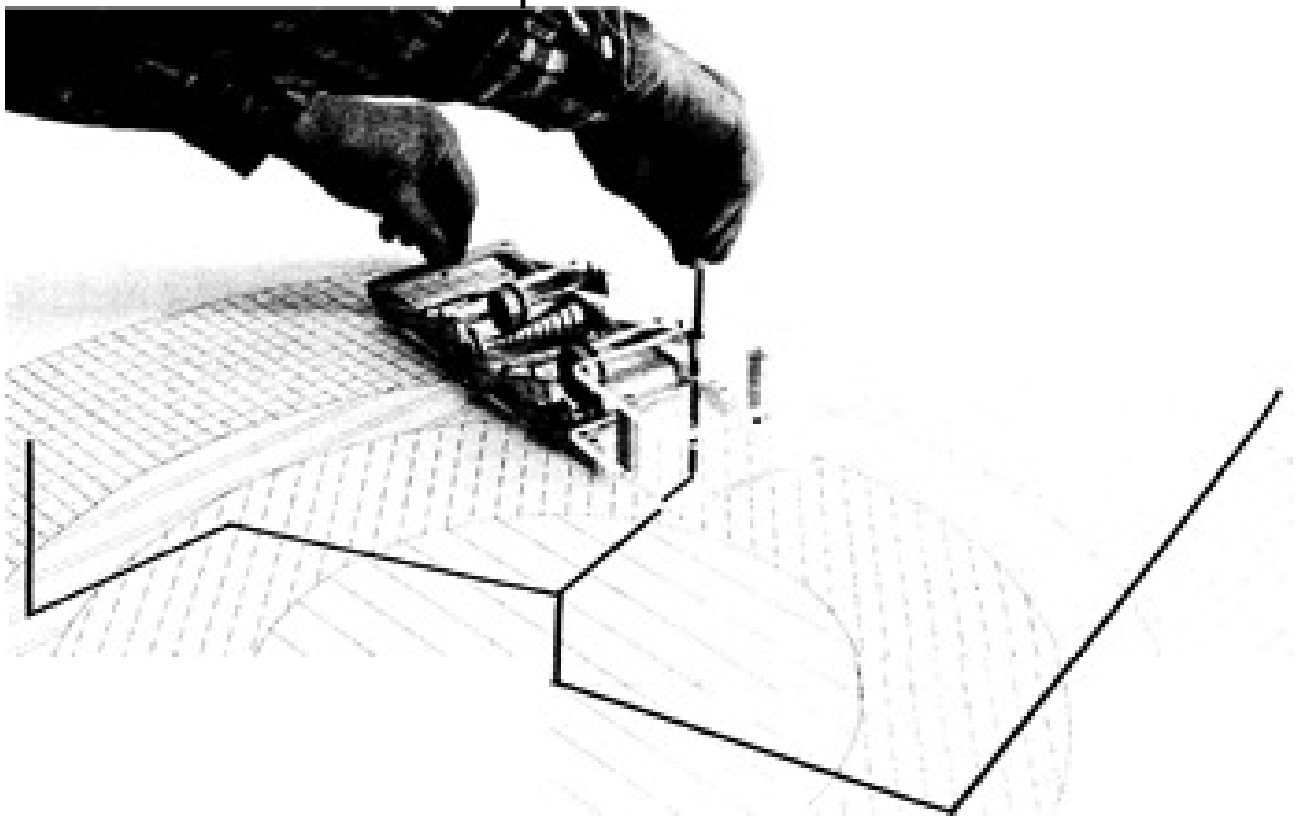
history of MC

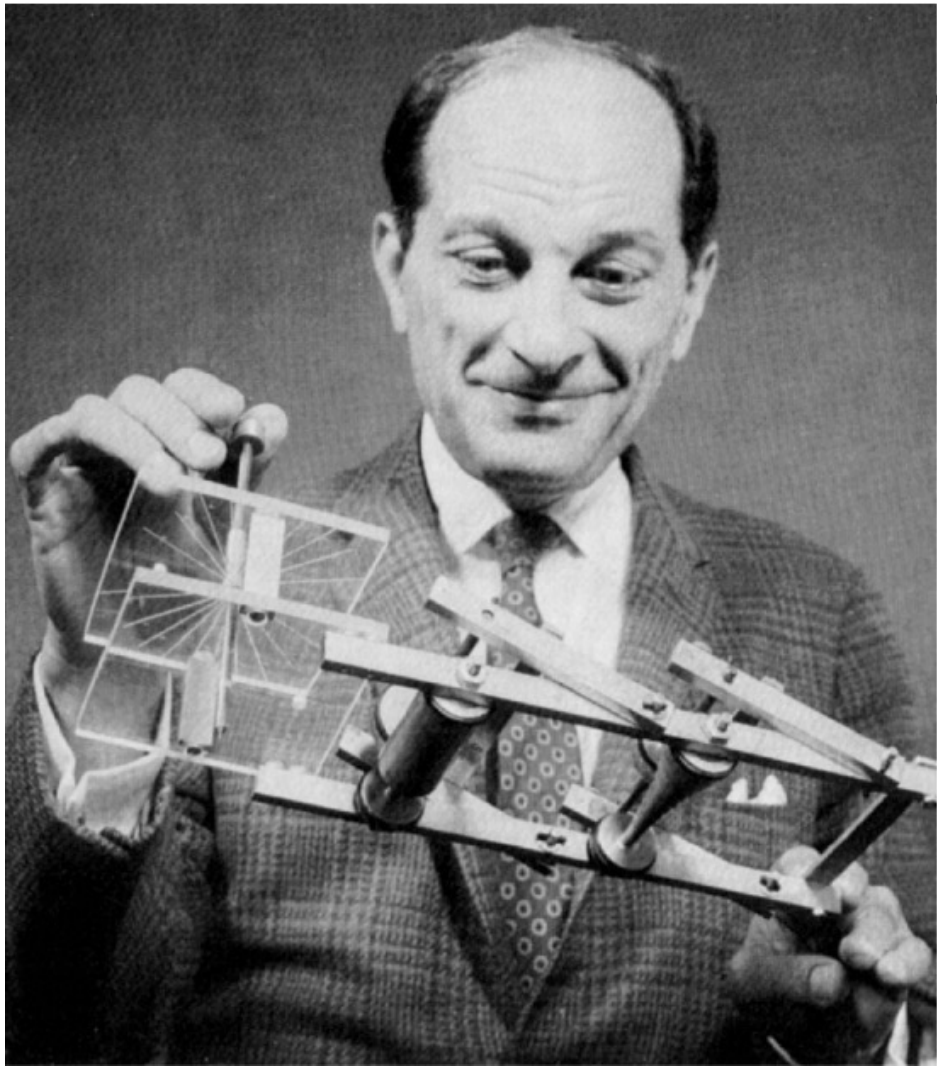
The number of different games is 52! = 52x51x52...x3x2x1 ~ 8x1067

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...



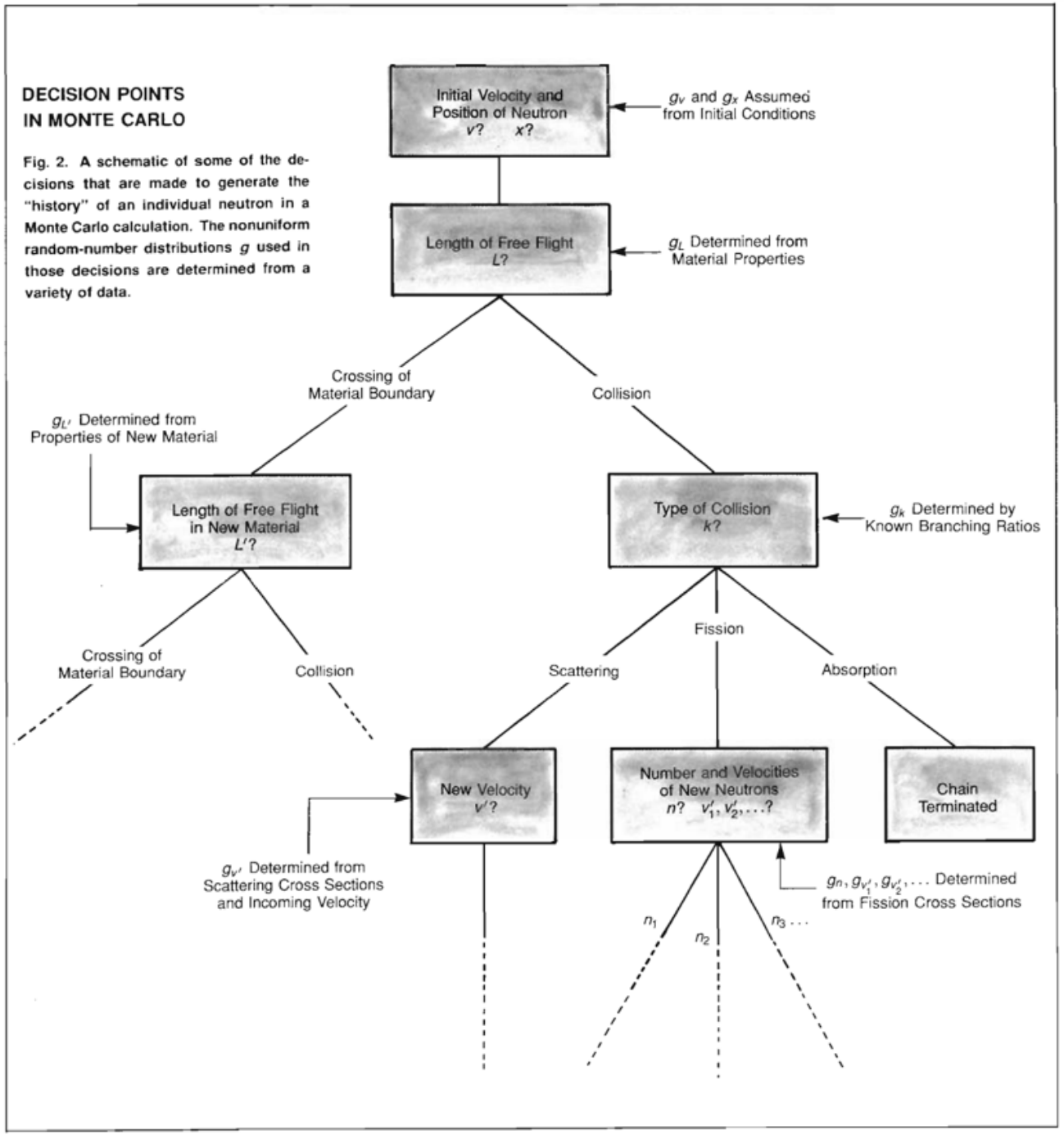


history of MC

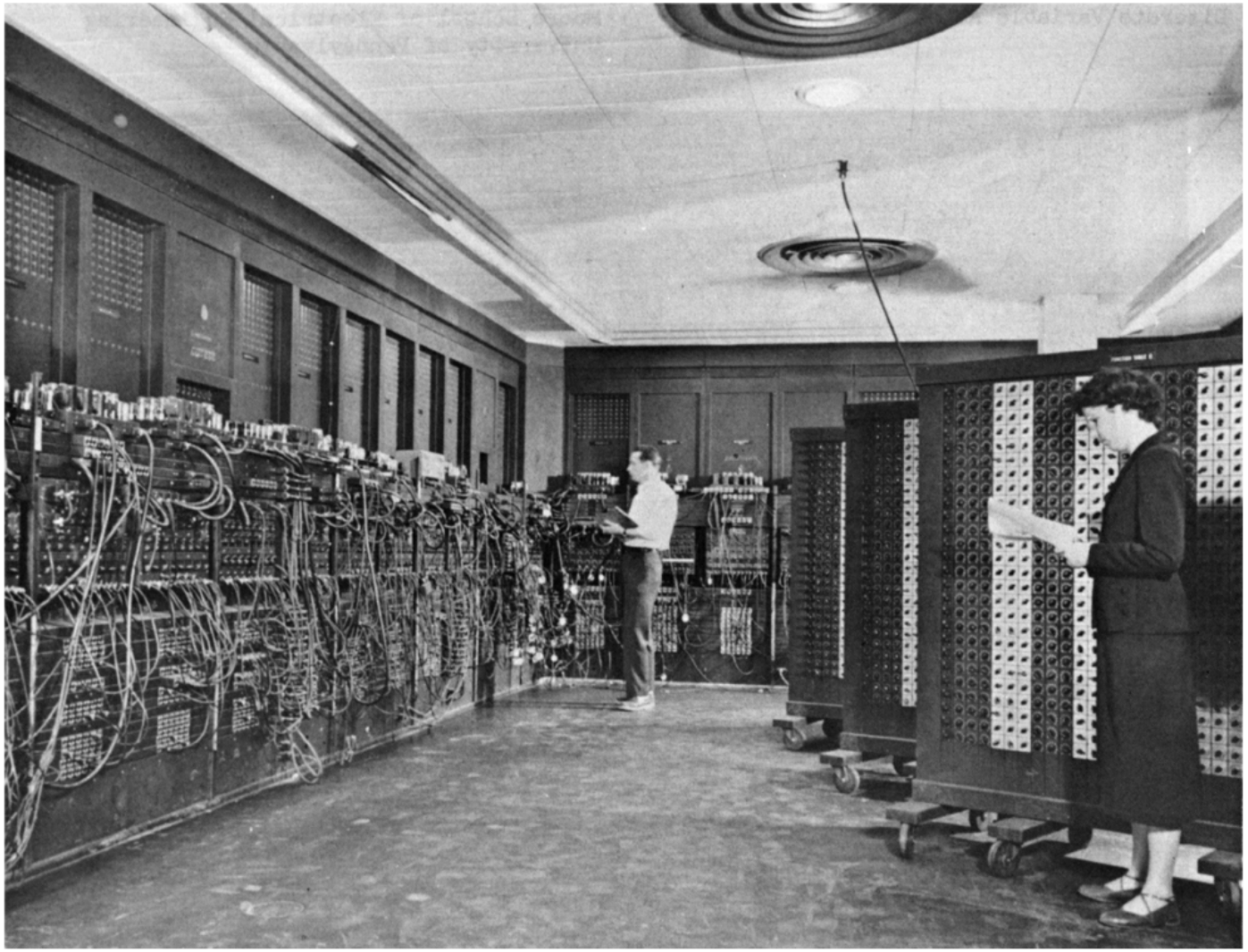
"What are the chances that a Canfield solitaire laid out with 52 cards will come out successfully?

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

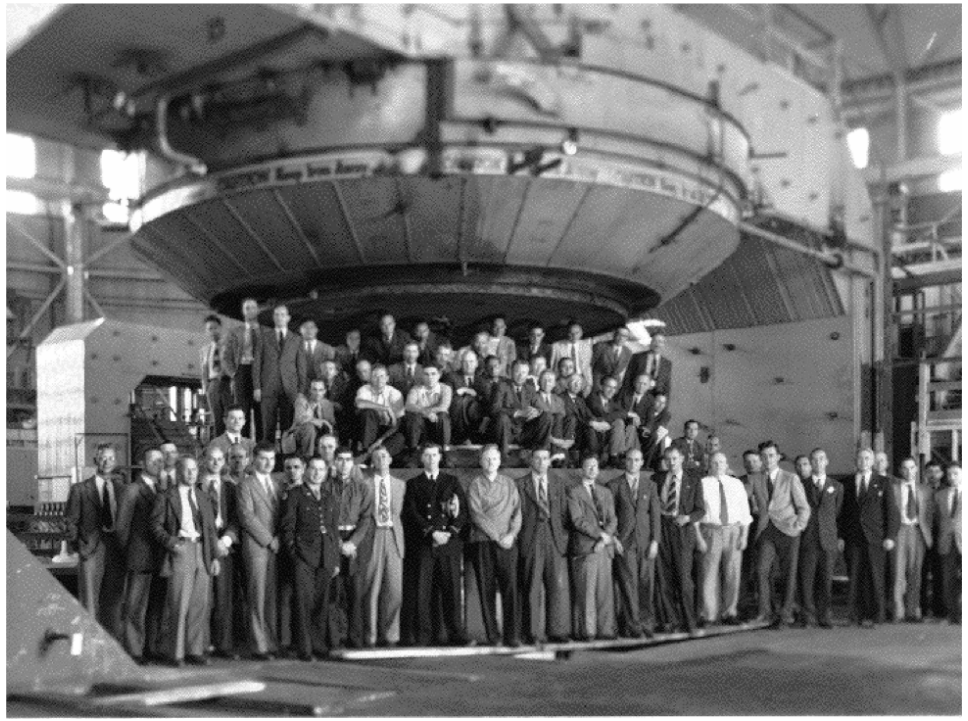


history of MC



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

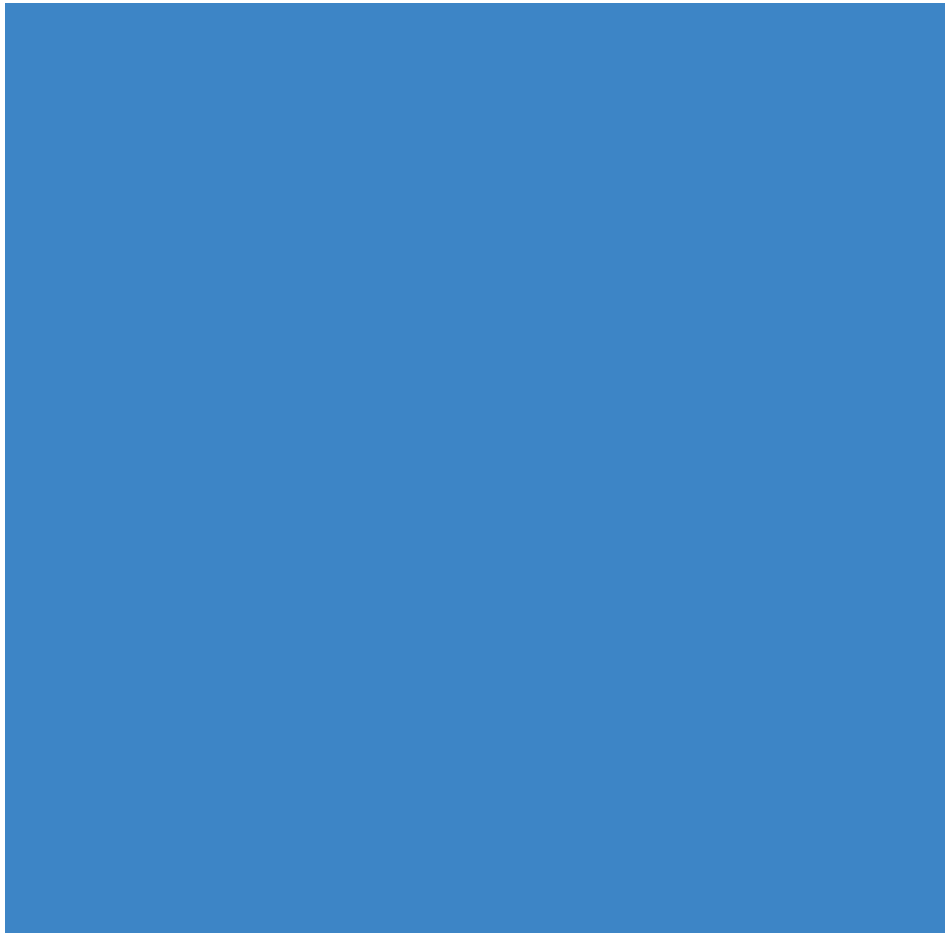
http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-88-9068



The Manhattan Project

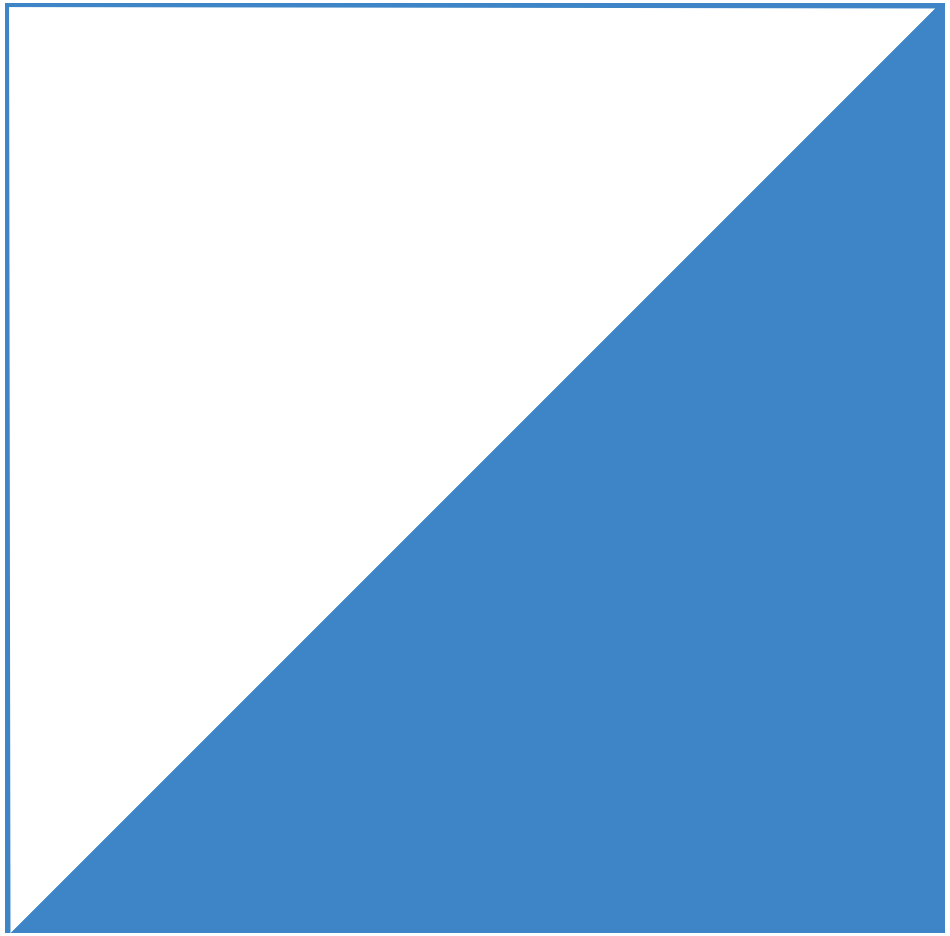
The advent of computing allowed for major innovation in the realm of simulation. Metropolis led a group that developed the Monte Carlo method, which simulates the results of an experiment by using a broad set of random numbers

simple example



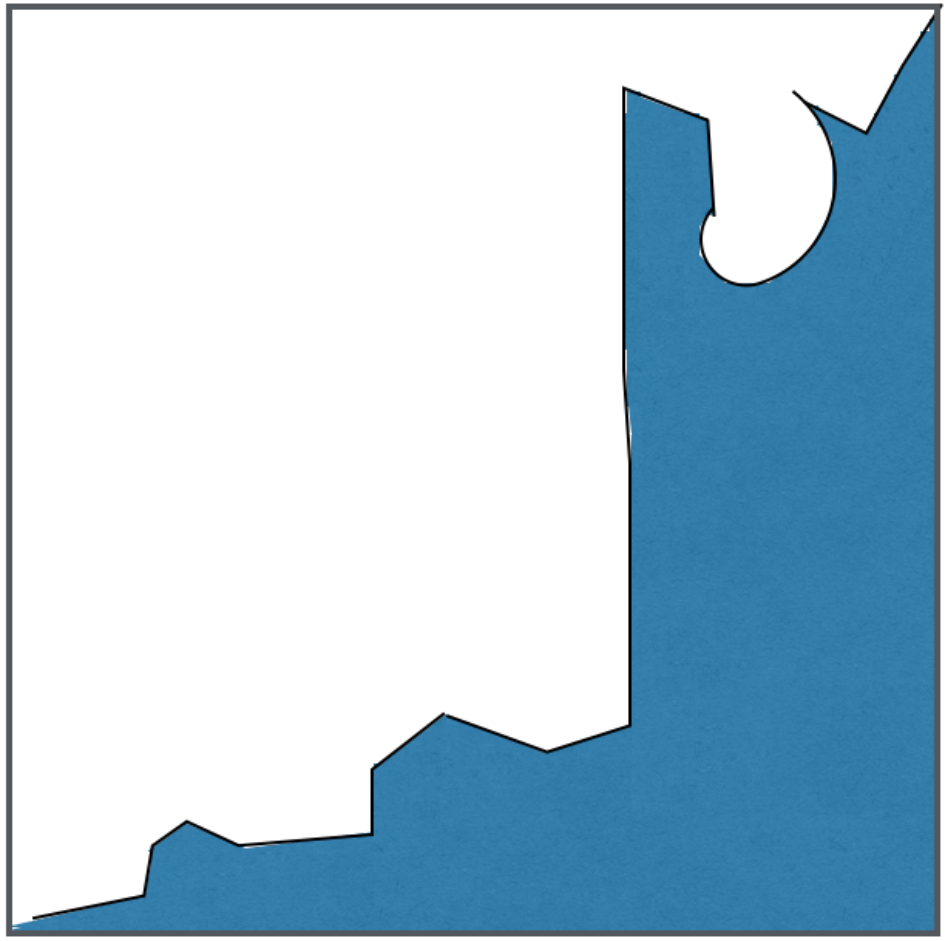
$$\text{Area} = \text{Base x Height}$$

simple example



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

simple example

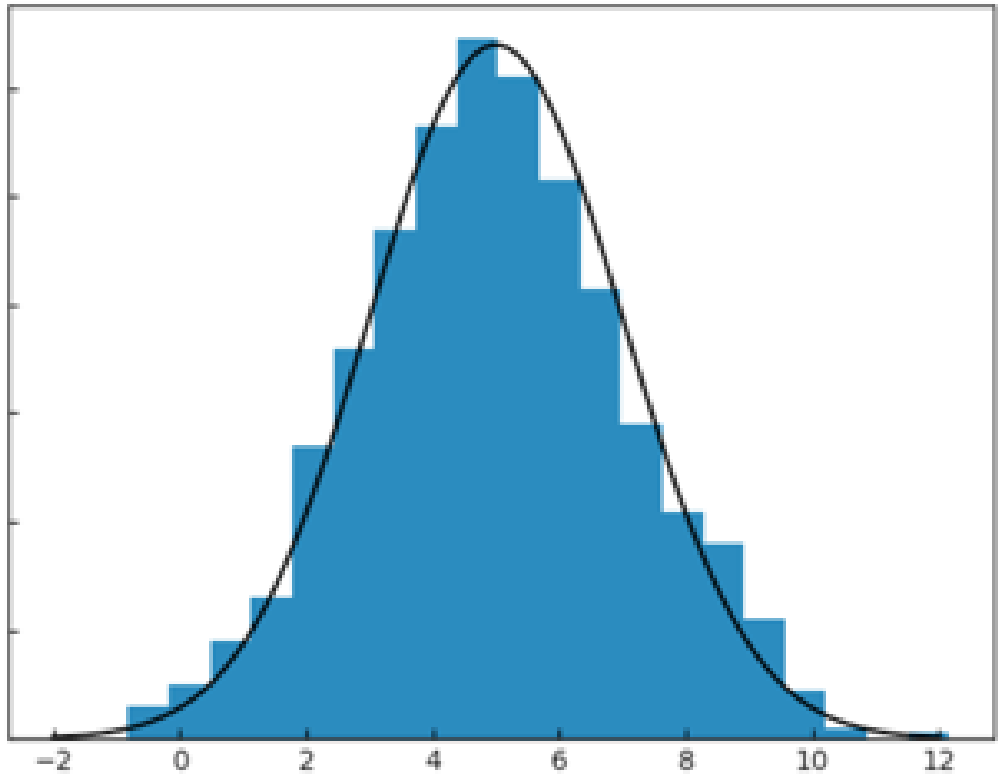


MCArea.ipynb

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

simple example

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



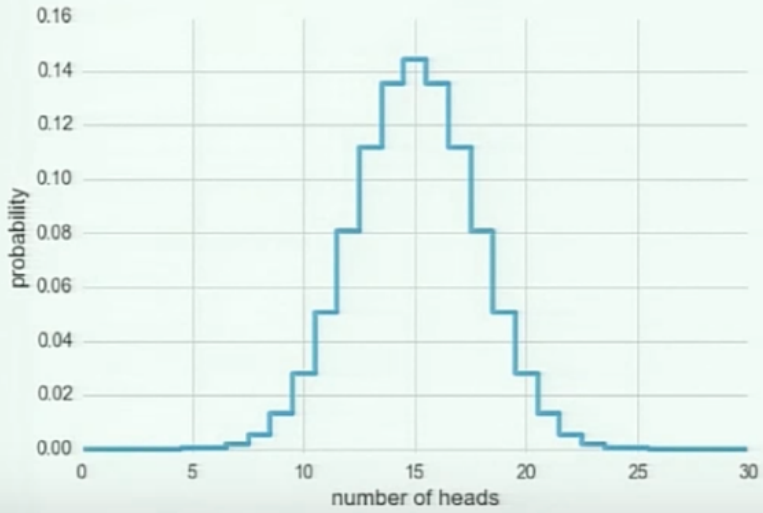
$$mean(X) = E[X] = \int X f(X) dX$$

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

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!

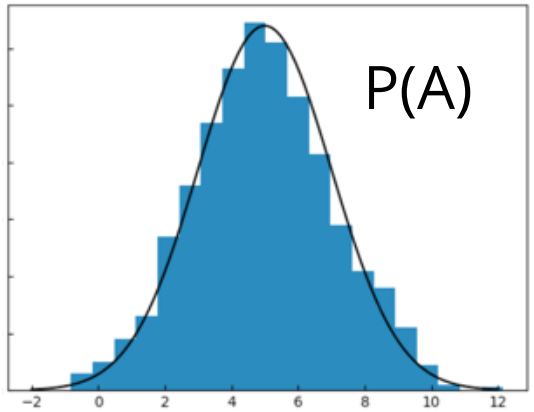
```
M = 0
for i in range(10000):
    trials = randint(2, size=30)
    if (trials.sum() >= 22):
        M += 1
p = M / 10000 # 0.008149
```

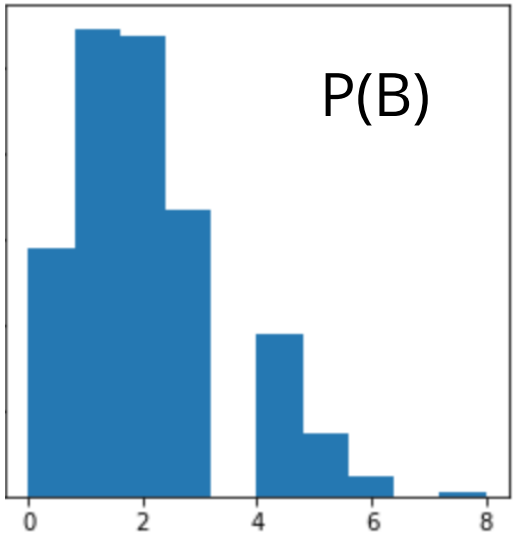
→ reject fair coin at p = 0.008

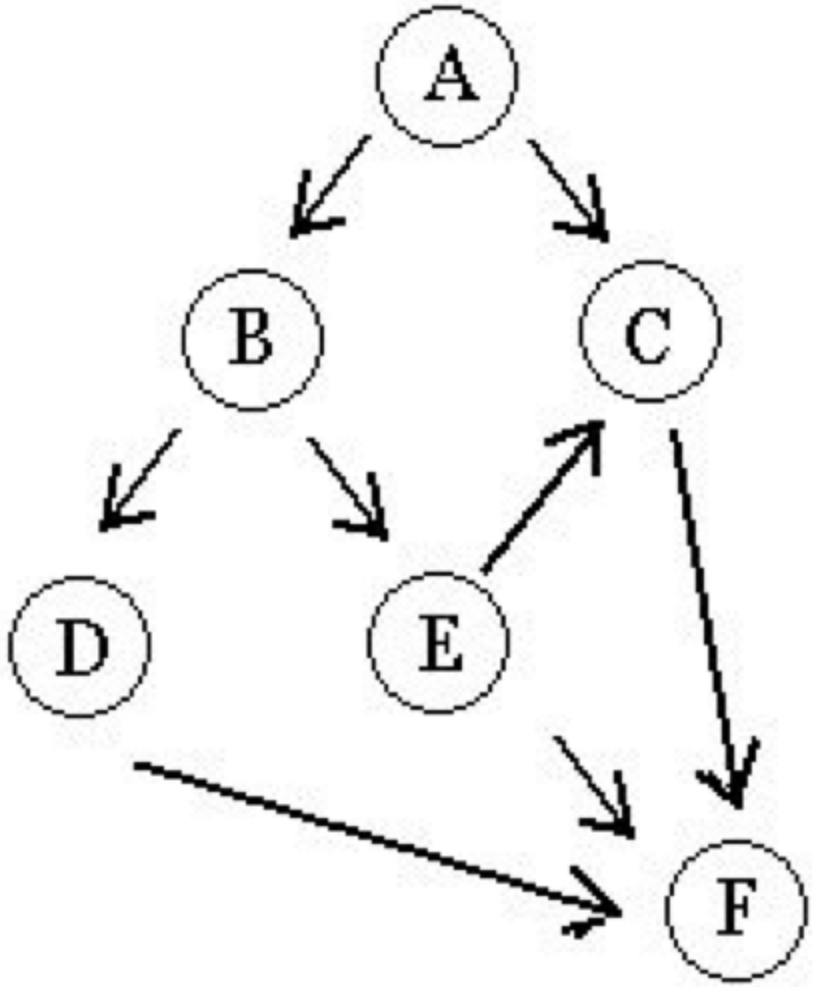
Statistics for Hackers, Jake Vanderplas PyCon16

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

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







$A \sim P(A)$

$B \sim P(B|A)$

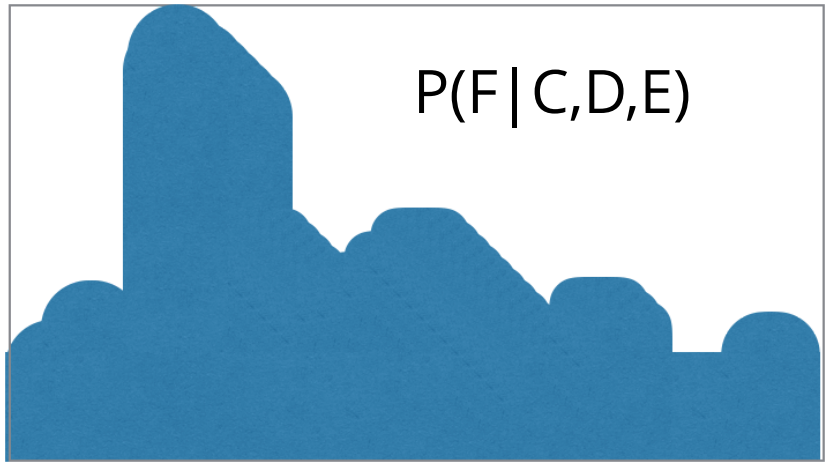
$C \sim P(C|A,E)$

$D \sim P(D|B)$

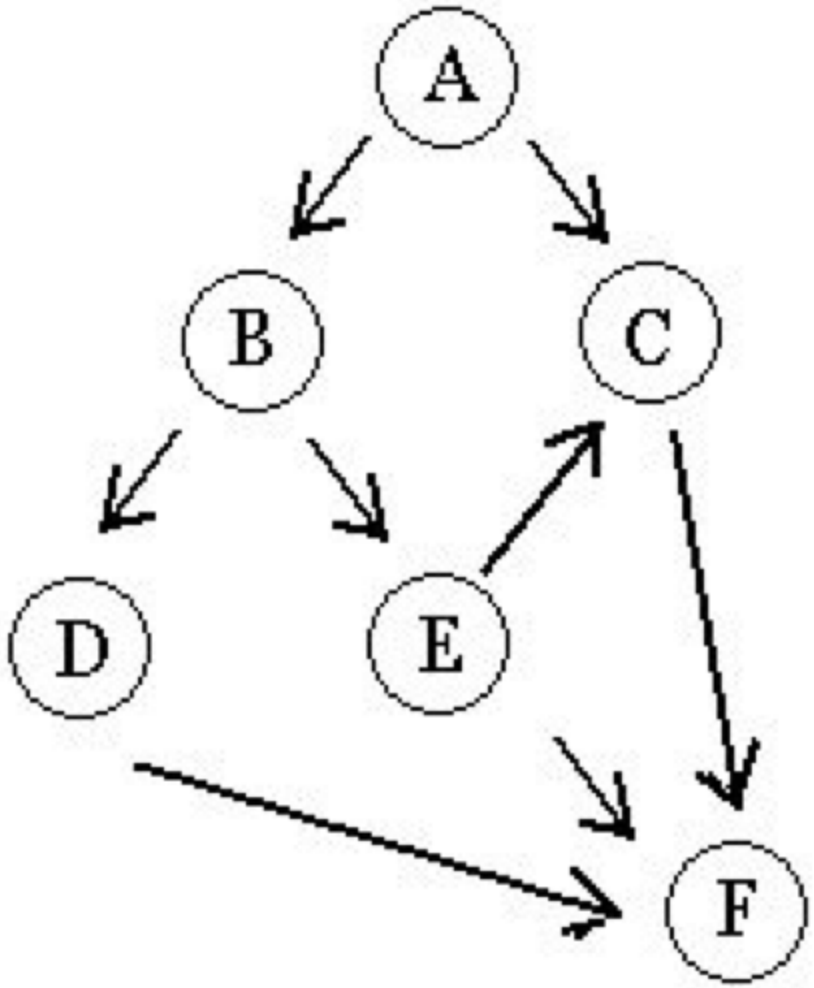
$E \sim P(E|B)$

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

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



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

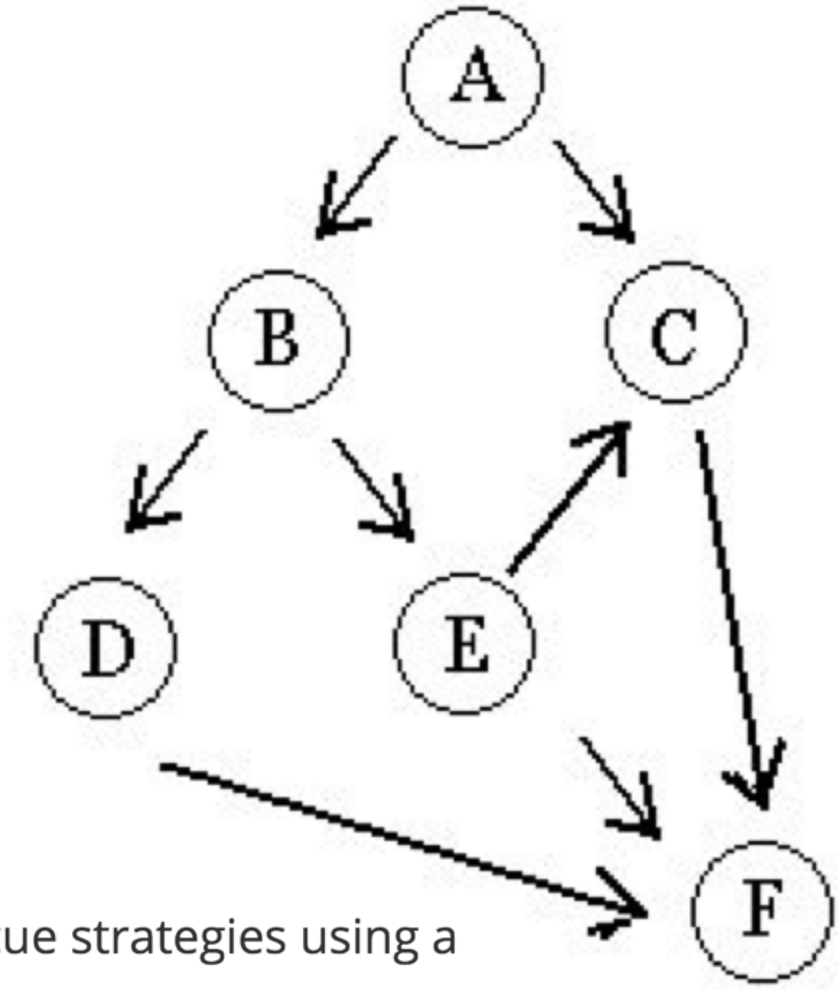
B prob to find them at time t

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

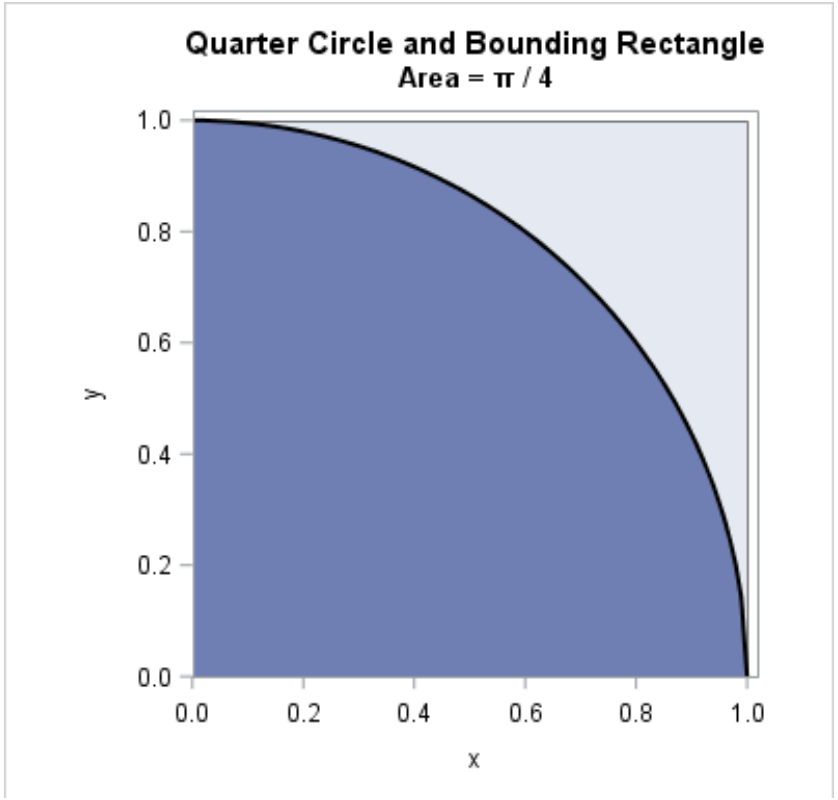
Published: May 3, 2017 • https://doi.org/10.1371/journal.pone.0175877

https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0175877

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

Calculate Pi

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



JOURNAL ARTICLE

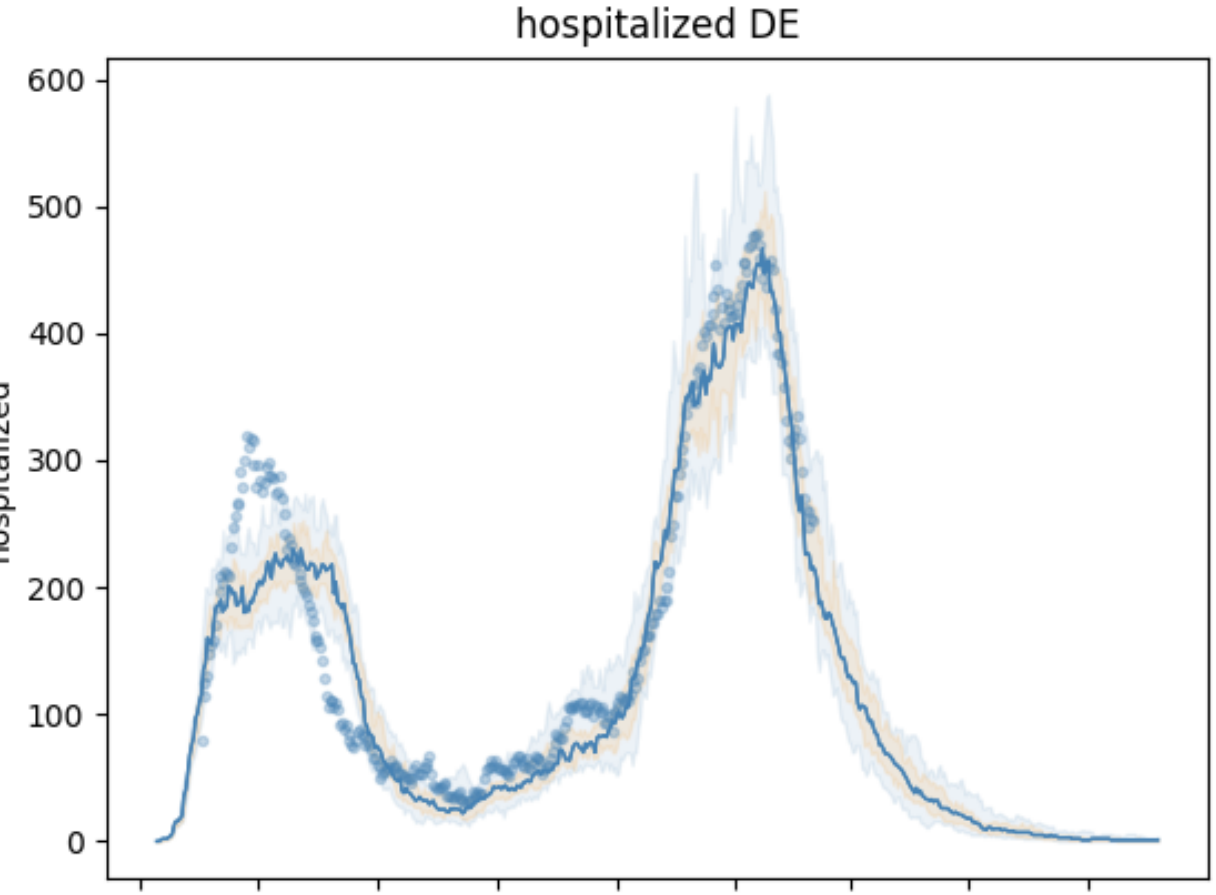
Determining Sample Sizes for Monte Carlo Integration

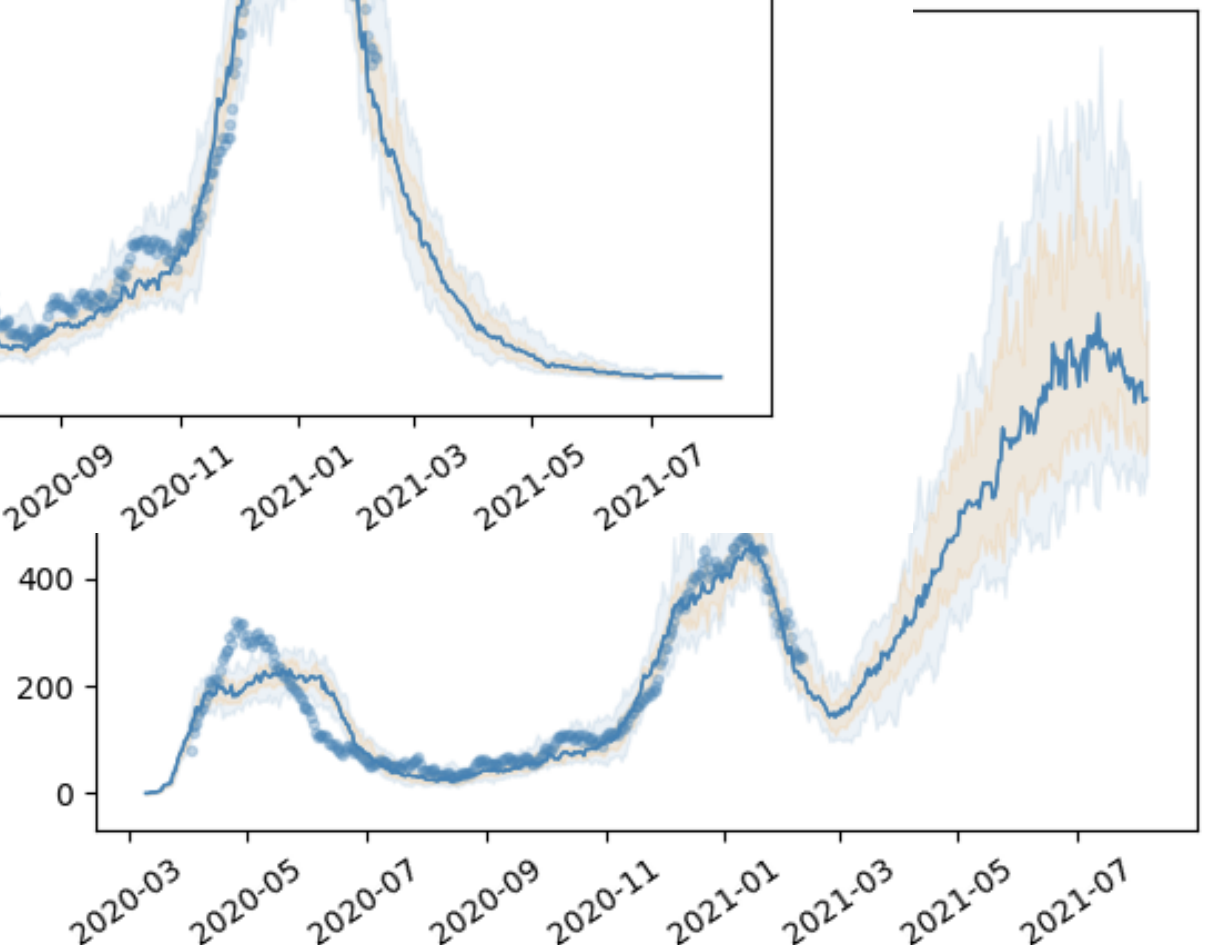
David Neal

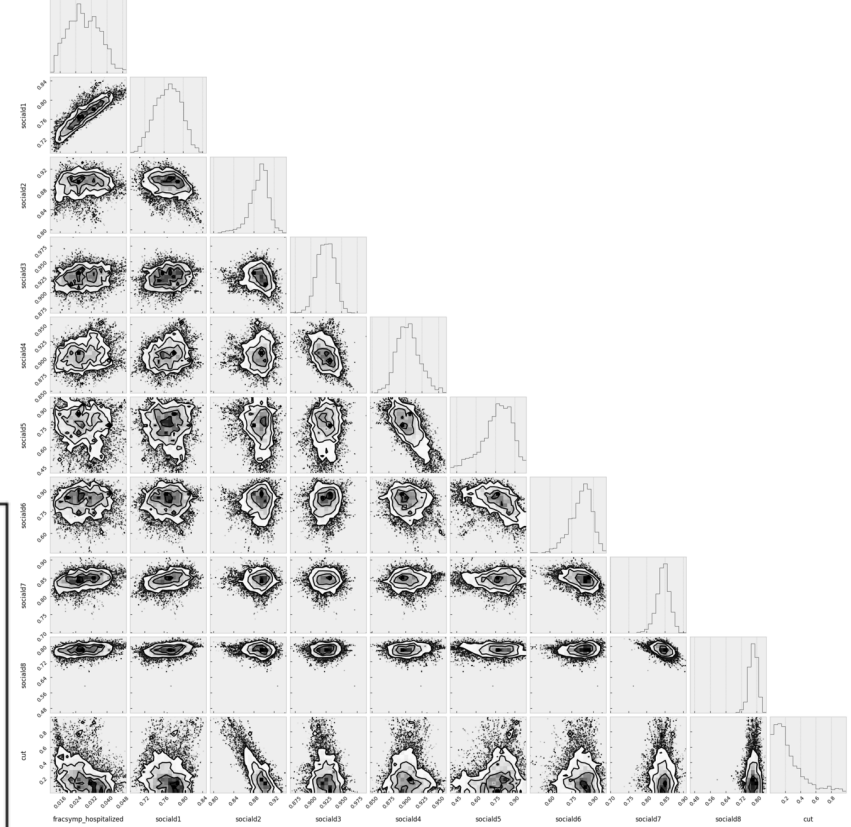
The College Mathematics Journal
Vol. 24, No. 3 (May, 1993), pp. 254-259
(6 pages)
Published By: Taylor & Francis, Ltd.

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

7 Model Optimization with MCMC


hospitalized DE
hospitalized
600
500
400
300
200
100
0
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
2021-03
2021-05
2021-07


400
200
0
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
2021-03
2021-05
2021-07


fracsymp_hospitalized
socialdl
socialdl2
socialdl3
socialdl4
socialdl5
socialdl6
socialdl7
socialdl8
cut

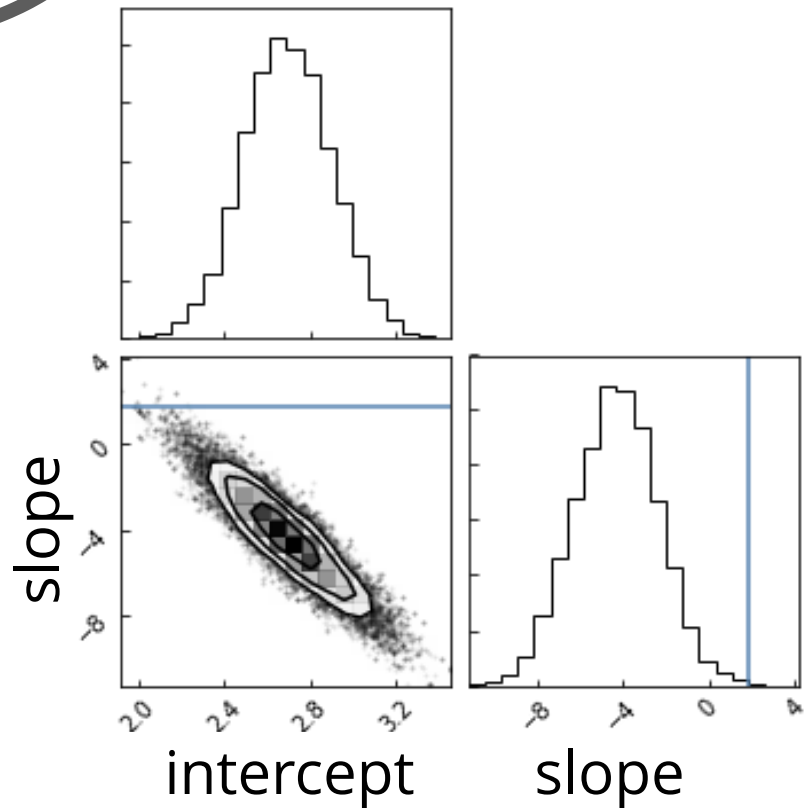

Monte Carlo
Markov
Chain
stochastic
"markovian"
sequence

MCMC

$$P(\theta|D,f) \propto P(D|\theta,f)\,P(\theta,f)$$

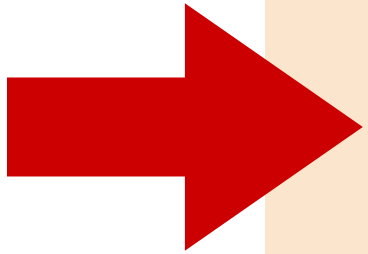
Goal: sample the posterior distribution

posterior: joint probability distributin of a parameter set (*m, b*) conditioned upon some data *D* and a model hypothesys *f*



MCMC

Goal: sample the posterior distribution



choose a starting point in the parameter space: ***current* = $\theta_0 = (m_0, b_0)$**
WHILE convergence criterion is met:
    calculate the current posterior $p_{curr} = P(D|\theta_0 f)$
    //proposal
    choose a new set of parameters ***new* = $\theta_{new} = (m_{new}, b_{new})$**
    calculate the proposed posterior $p_{new} = P(D|\theta_{new} f)$
    IF $p_{new}/p_{curr} > 1$:
        current = new
    ELSE:
        //probabilistic step: accept with probability p_{new}/p_{curr}
        draw a random number r ϵ$U[0,1]$
        IF $p_{new}/p_{curr} > r$:
            current = new
        ELSE:
            pass // do nothing

stochasticity allows us to explore the whole surface but spend more time in interesting spots

MCMC

Goal: sample the posterior distribution

choose a starting point in the parameter space: ***current* = $\theta_0 = (m_0, b_0)$**

WHILE convergence criterion is met:

calculate the current posterior $p_{curr} = P(D|\theta_0 f)$

//proposal

choose a new set of parameters ***new* = $\theta_{new} = (m_{new}, b_{new})$**

calculate the proposed posterior $p_{new} = P(D|\theta_{new} f)$

IF $p_{new}/p_{curr} > 1$:

current = new

ELSE:

MCMC

choose a starting point in the parameter space: ***current* = $\theta_0 = (m_0, b_0)$**

WHILE convergence criterion is met:

calculate the current posterior $p_{curr} = P(D|\theta_0 f)$

//proposal

choose a new set of parameters ***new* = $\theta_{new} = (m_{new}, b_{new})$**

calculate the proposed posterior $p_{new} = P(D|\theta_{new} f)$

IF $p_{new}/p_{curr} > 1$:

current = new

ELSE:

//probabilistic step: accept with probability p_{new}/p_{curr}

draw a random number r ϵ$U[0,1]$

IF $p_{new}/p_{curr} > r$:

current = new

ELSE:

pass // do nothing

Questions:

1. how do I choose the next point?

Any *Markovian ergodic* process

A Markovian Process

A process is Markovian if the next state of the system is determined stochastically as a perturbation of the current state of the system, and *only* the current state of the system, i.e. the system has no memory of earlier states (a *memory-less* process).

Definition

Ergodic Process

(given enough time) the entire parameter space would be sampled.

Detailed Balance is a sufficient condition
for ergodicity
Metropolis Rosenbluth Rosenbluth Teller 1953 - Hastings 1970

At equilibrium, each elementary process should be equilibrated by its reverse process.

reversible Markov process

$$\pi(x_1)P(x_2|x_1) = \pi(x_2)P(x_1|x_2)$$

Definition

it can be shown that

If the chains are a Markovian Ergodic process

the algorithm is guaranteed to explore the entire likelihood surface given infinite time

This is in contrast to gradient descent,
which can get stuck in local minima or in
local saddle points.

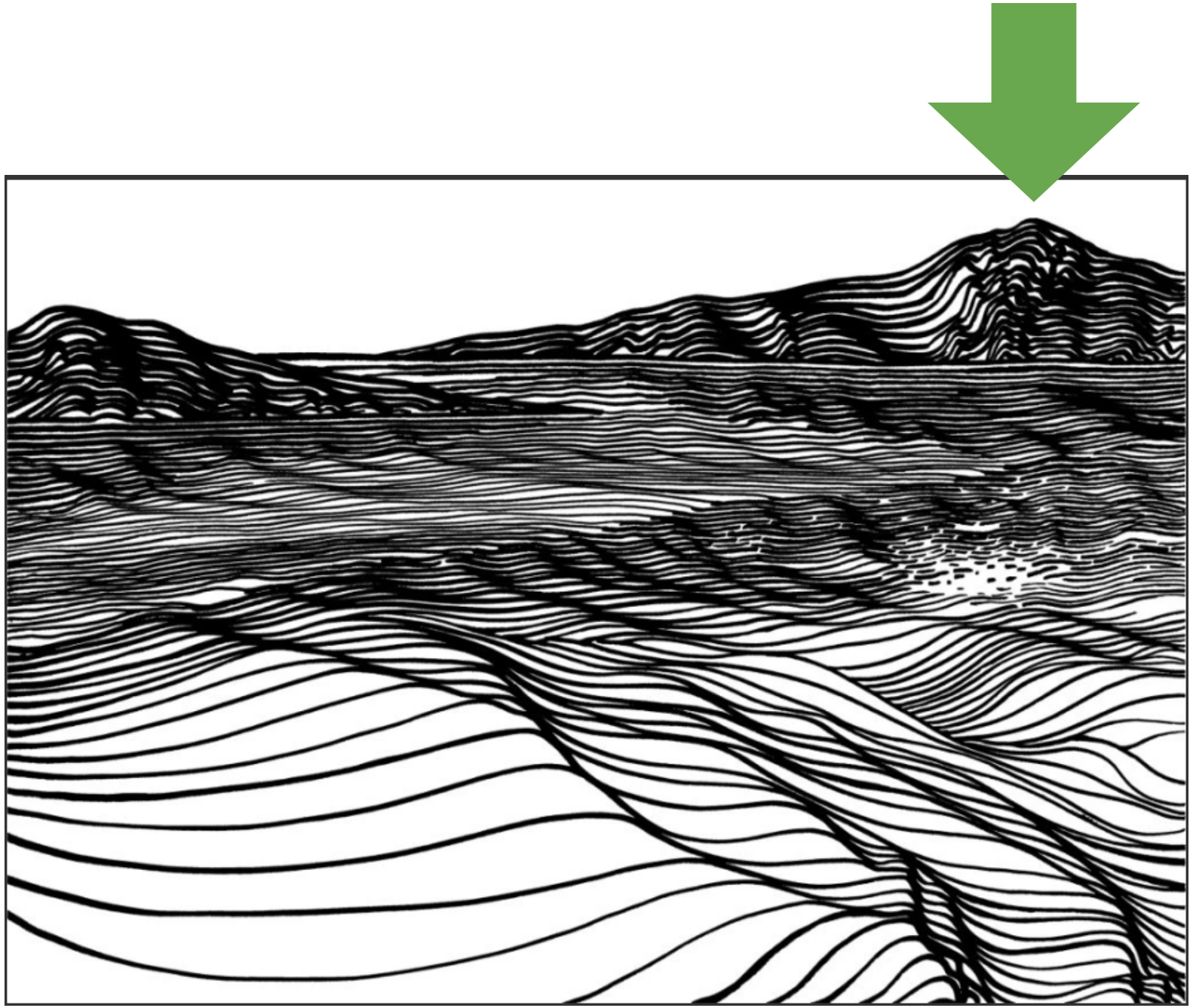
The problem of fitting models to data reduces to finding the

maximum *likelihood*

of the data given the model

This is effectively done by finding the **minimum** of the

-log(*likelihood*)

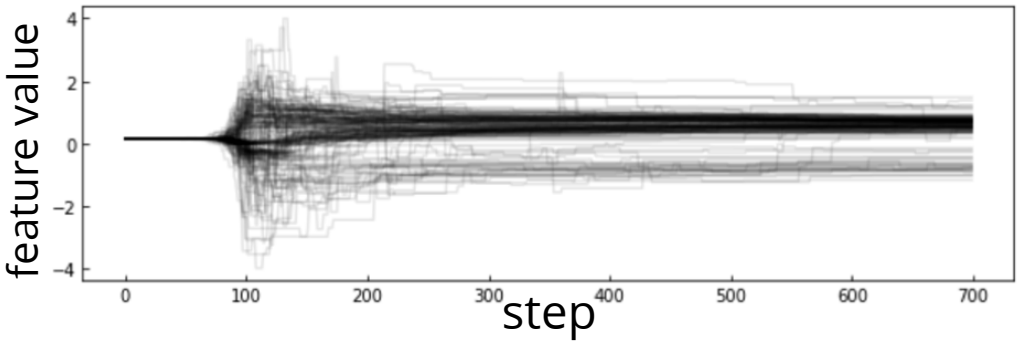


MCMC

The chains: *the algorithm creates a "chain" (a random walk) that "explores" the likelihood surface.*

More efficient is to run many parallel chains - each exploring the surface, an "ensemble"

The path of the chains can be shown along each feature



choose a starting point in the parameter space: ***current = θ_0 = (m_0,b_0)***
WHILE convergence criterion is met:
    calculate the current posterior ***p_{curr} = P(D|θ_0,f)***
    //proposal
    choose a new set of parameters ***new = θ_{new} = (m_{new},b_{new})***
    calculate the proposed posterior ***p_{new} = P(D|θ_{new},f)***
    IF $p_{new}/p_{curr} > 1$:
        current = new
    ELSE:
        //probabilistic step: accept with probability p_{new}/p_{curr}
        draw a random number r ϵU[0,1]
        IF $p_{new}/p_{curr} > r$:
            current = new
        ELSE:
            pass // do nothing

MCMC

how to choose the next point

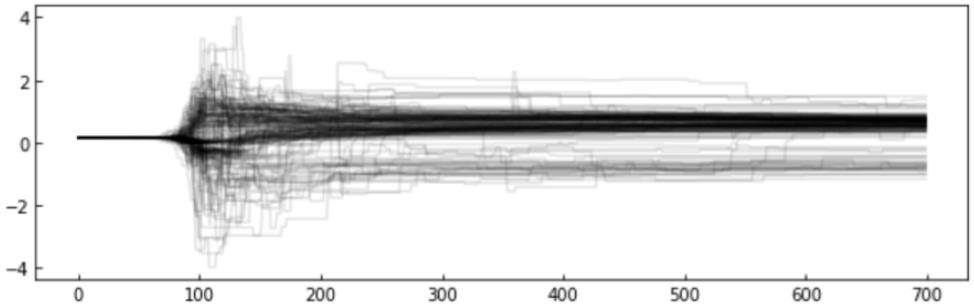
how you make this decision names the algorithm

simulated annealing (good for multimodal)

parallel tempering (good for multimodal)

differential evolution (good for covariant spaces)

Gibbs sampling (move in along one variable at a time)



choose a starting point in the parameter space: ***current* = $\theta_0 = (m_0, b_0)$**

WHILE convergence criterion is met:

calculate the current posterior ***$p_{curr} = P(D|\theta_0 f)$***

//proposal

choose a new set of parameters ***new* = $\theta_{new} = (m_{new}, b_{new})$**

calculate the proposed posterior ***$p_{new} = P(D|\theta_{new} f)$***

IF $p_{new}/p_{curr} > 1$:

current = new

ELSE:

//probabilistic step: accept with probability p_{new}/p_{curr}

draw a random number r $\in U[0,1]$

IF $p_{new}/p_{curr} > r$:

current = new

ELSE:

pass // do nothing

MCMC

how to choose the next point

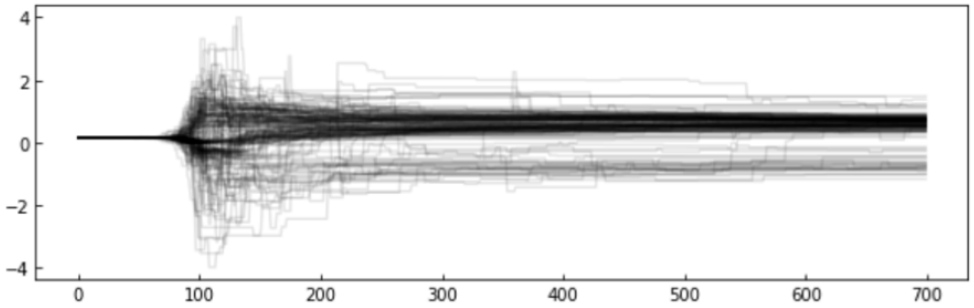
how you make this decision names the algorithm

simulated annealing (good for multimodal)

parallel tempering (good for multimodal)

differential evolution (good for covariant spaces)

Gibbs sampling (move in along one variable at a time)



Annealing Markov Chain Monte Carlo With Applications to Ancestral Inference

Charles J. GEYER and Elizabeth A. THOMPSON*

https://www.jstor.org/stable/2291325?seq=1

Markov chain Monte Carlo (MCMC) in the form of the Metropolis–Hastings algorithm (Hastings 1970; Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller 1953) and its special case the Gibbs sampler (Geman and Geman 1984) has been used in recent years to attack a wide variety of intractable statistical problems. (See, for example, Besag and Green 1993, Geyer 1992, Geyer and Thompson 1992, Smith and Roberts 1993, Tierney 1994, and the accompanying discussions and references.) MCMC simulates realizations from probability distributions whose densities are known up to a normalizing factor. If $h(x)$ is a nonnegative integrable function on the sample space, then the Metropolis–Hastings algorithm simulates a Markov chain whose equilibrium distribution is proportional to $h(x)$ using only evaluations of $h(x)$.

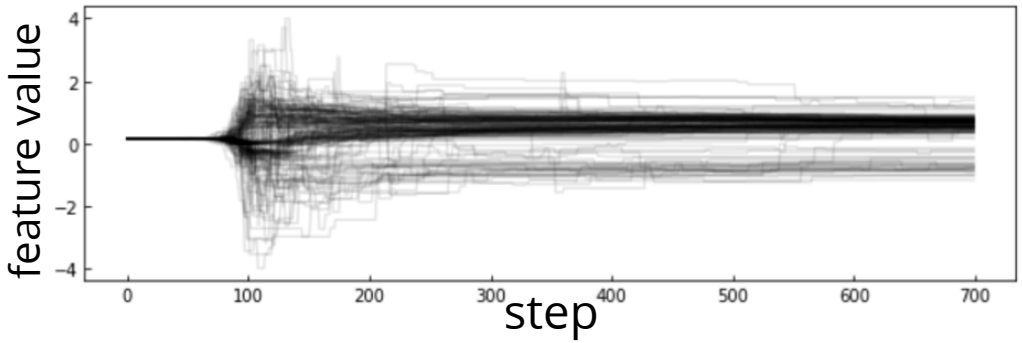
If the chain is irreducible, then time averages over the chain converge to expectations with respect to the stationary distribution as the Monte Carlo sample size goes to infinity; but if the chain is slowly mixing, then it may take astronomically large sample sizes to get accurate estimates. Slow mixing typically occurs in problems where the sample space has high dimension. For samplers that update one variable at a time, like the Gibbs sampler, the mixing time can be exponential in the number of variables. Thus, to do MCMC on high-

MCMC

MCMC animations
https://www.youtube.com/watch?v=J6FrNf5__G0&list=PLgArfv_fOU5dwjeP_57NO_jnRJ7cWCe6J

Examples of how to choose the next point

affine invariant : EMCEE package



choose a starting point in the parameter space: ***current* = $\theta_0 = (m_0, b_0)$**

WHILE convergence criterion is met:

calculate the current posterior ***p_{curr} = $P(D|\theta_0, f)$***

//proposal

choose a new set of parameters ***new* = $\theta_{new} = (m_{new}, b_{new})$**

calculate the proposed posterior ***p_{new} = $P(D|\theta_{new}, f)$***

IF $p_{new}/p_{curr} > 1$:

current = new

ELSE:

//probabilistic step: accept with probability p_{new}/p_{curr}

draw a random number r $\in U[0,1]$

IF $p_{new}/p_{curr} > r$:

current = new

ELSE:

pass // do nothing

MCMC convergence

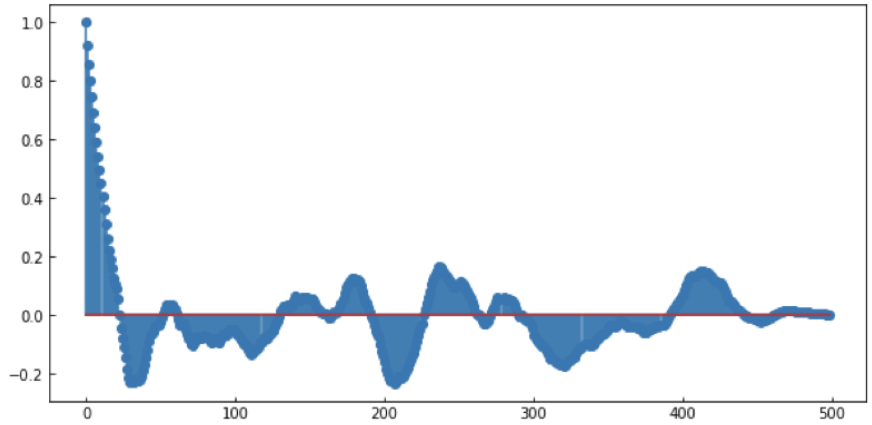
Goal: sample the posterior distribution

Questions:

1. how do I choose the next point?

2. when have I sampled the posterior adequatly?
   has your MCMC *converged* ?

MCMC convergence



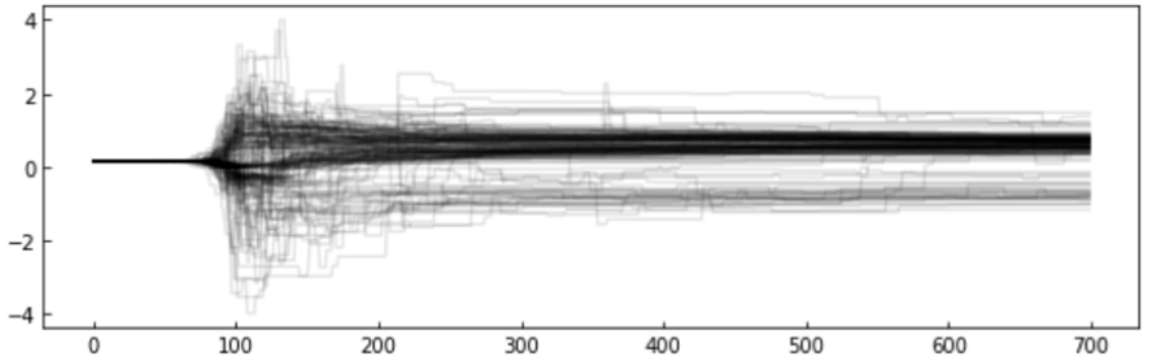
Goal: sample the posterior distribution

Questions:

1. how do I choose the next point?
2. when have I sampled the posterior adequatly?
   has your MCMC *converged*?

1. **check autocorrelation within a chain (*Raftery*)**
2. check that all chains coverged to same region (a stationary distribution *GelmanRubin*)
3. mean at beginning = mean at end (*Geweke*)
4. check that entire chain reached stationary distribution (or a final fraction of the chain, *Heidelberg-Welch* using Cramer-von-Mises statistic)

MCMC convergence



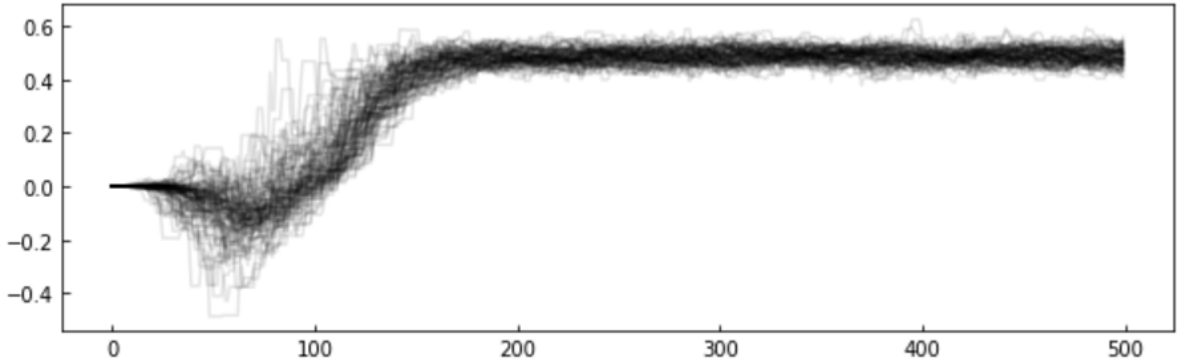
Goal: sample the posterior distribution

Questions:

1. how do I choose the next point?
2. when have I sampled the posterior adequatly?
   has your MCMC *converged*?

1. check autocorrelation within a chain (*Raftery*)
2. **check that all chains coverged to same region (a stationary distribution *GelmanRubin*)**
3. mean at beginning = mean at end (*Geweke*)
4. check that entire chain reached stationary distribution (or a final fraction of the chain, *Heidelberg-Welch* using Cramer-von-Mises statistic)

MCMC convergence



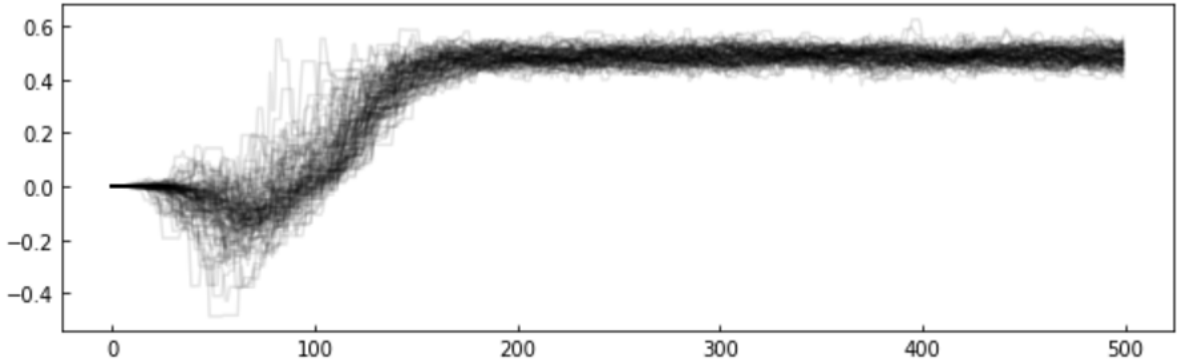
Goal: sample the posterior distribution

Questions:

1. how do I choose the next point?
2. when have I sampled the posterior adequatly?
   has your MCMC *converged*?

1. check autocorrelation within a chain (*Raftery*)
2. check that all chains coverged to same region (a stationary distribution *GelmanRubin*)
3. **mean at beginning = mean at end (*Geweke*)**
4. **check that entire chain reached stationary distribution (or a final fraction of the chain, *Heidelberg-Welch* using Cramer-von-Mises statistic)**

MCMC convergence



Goal: sample the posterior distribution

Questions:

1. how do I choose the next point?
2. when have I sampled the posterior adequatly?
   has your MCMC *converged*?
3. how can it be-the samples are *not independent!*

good point!…

1. check autocorrelation within a chain (*Raftery*)
2. check that all chains coverged to same region (a stationary distribution *GelmanRubin*)
3. mean at beginning = mean at end (*Geweke*)
4. check that entire chain reached stationary distribution (or a final fraction of the chain, *Heidelberg-Welch* using Cramer-von-Mises statistic)

8

Model Selection

Principles

what model should I choose?

No matter what anyone tells you an answer to
this question cannot be given in the abstract case:
it is a domain specific question!

except:

the principle of parsimony

principle of parsimony

or Ockham's razor

Pluralitas non est ponenda sine neccesitate

William of Ockham (logician and Franciscan friar) 1300ca
but probably to be attributed to John Duns Scotus (1265–1308)

“Complexity needs not to be postulated without a need for it”

principle of parsimony

the earth is round,

and it orbits around the sun

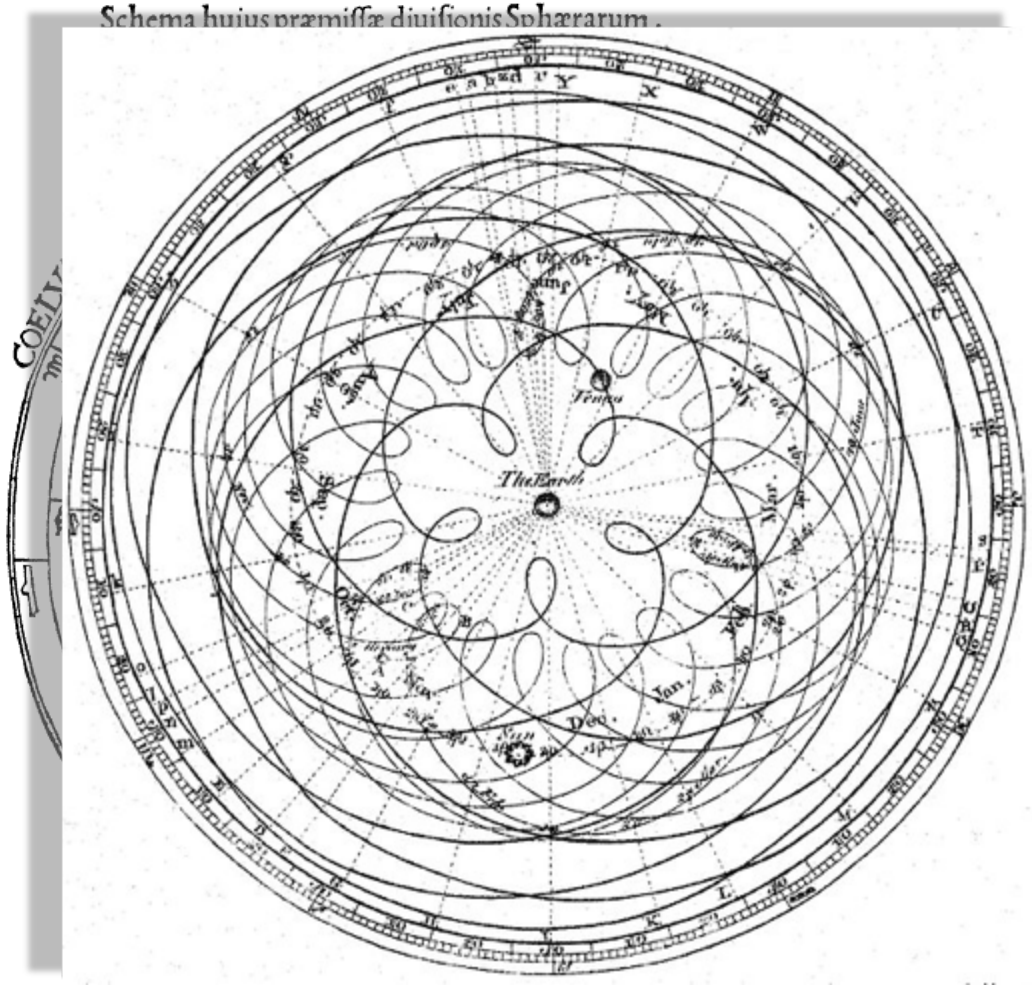
Geocentric models are intuitive:

from our perspective we see the Sun moving, while we stay still

Peter Apian, *Cosmographia*, Antwerp, 1524 from Edward Grant,

"Celestial Orbs in the Latin Middle Ages", *Isis*, Vol. 78, No. 2. (Jun., 1987).

principle of parsimony



Encyclopaedia Brittanica 1st Edition

Dr Long's copy of Cassini, 1777

the earth is round,

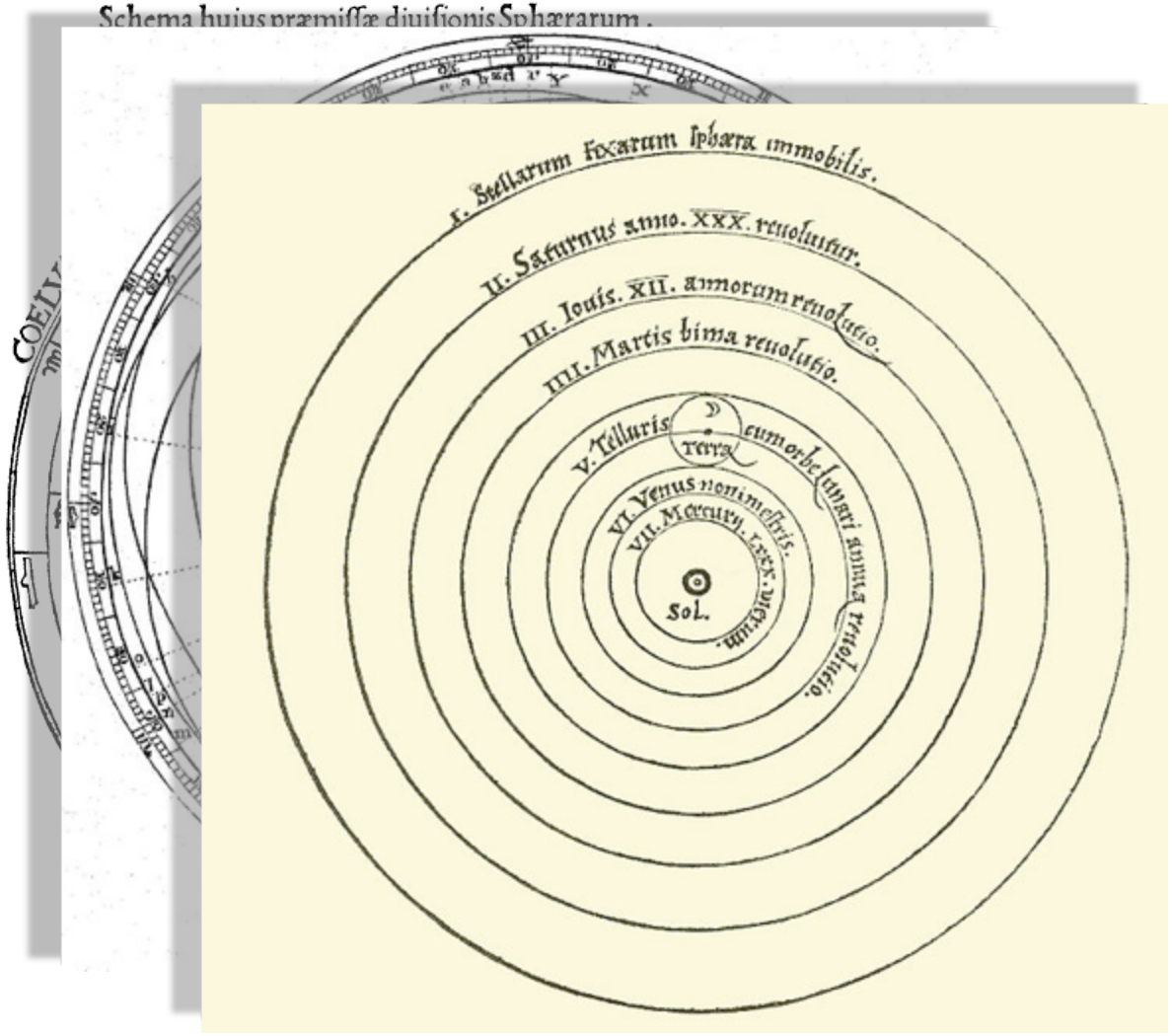
and it orbits around the sun

As observations improve

this model can no longer fit the data!

not easily anyways...

principle of parsimony



the earth is round,

~~and it orbits around the sun~~

A new model that is much simpler fit the data just as well

(perhaps though only until better data comes...)

Heliocentric model from Nicolaus Copernicus' *De revolutionibus orbium coelestium*.

principle of parsimony

or Ockham's razor

Pluralitas non est ponenda sine neccesitate

William of Ockham (logician and Franciscan friar) 1300ca
but probably to be attributed to John Duns Scotus (1265–1308)

“Complexity needs not to be postulated without a need for it”
“Between 2 theories that perform similarly choose the ***simpler one***”

principle of parsimony

or Ockham's razor

Pluralitas non est ponenda sine neccesitate

William of Ockham (logician and Franciscan friar) 1300ca
but probably to be attributed to John Duns Scotus (1265–1308)

“Complexity needs not to be postulated without a need for it”
“Between 2 theories that perform similarly choose the ***simpler one***”

the principle of parsimony

or Ockham's razor

Between 2 theories that perform similarly choose the ***simpler one***

In the context of model selection simpler means "with fewer parameters"

Key Concept

principle of parsimony

Science and Statistics George E. P. Box (1976)

Journal of the American Statistical Association, Vol. 71, No. 356, pp. 791-799.

Since all models are wrong the scientist cannot obtain a "correct" one by excessive elaboration. On the contrary following William of Ockham he should seek an economical description of natural phenomena

Since all models are wrong the scientist must be alert to what is importantly wrong.

principle of parsimony

For a dissenting argument

https://www.theatlantic.com/science/archive/2016/08/occams-razor/495332/

SCIENCE

The Tyranny of Simple Explanations

The history of science has been distorted by a longstanding conviction that correct theories about nature are always the most elegant ones.

PHILIP BALL AUGUST 11, 2016

But they should resist it. The value of keeping assumptions to a minimum is cognitive, not ontological: It helps you to think. A theory is not “better” if it is simpler—but it might well be more useful, and that counts for much more.

81 model selection methodology

AIC BIC MLD

HOW DO I CHOOSE A MODEL?

Given two models which is preferable?

A *rigorous* answer (in terms of NHST) can be obtained for ***2 nested* models**

This directly answers the question:

"is my more complex model *overfitting* the data?"

The LR statistics is expected to follow a χ^2 distrbution under the *Null Hypothesis* that the ***simpler model is preferable***

NESTED MODELS : one model contains the other one, e.g.

y= ***mx + I***

is contained in

*y=ax**2+* ***mx + I***

Likelihood-ratio tests

likelihood ratio statistics LR

$$LR = -2log_e \frac{L(\text{complex model})}{L(\text{simple model})}$$

statsmodels.model.compare_lr_ratio()

HOW DO I CHOOSE A MODEL?

Given two models which is preferable?

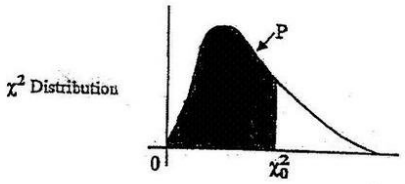
Likelihood-ratio tests

likelihood ratio statistics LR

$$LR = -2log_e \frac{L(\text{complex model})}{L(\text{simple model})}$$

statsmodels.model.compare_lr_ratio()

The LR statistics is expected to follow a χ^2 distrbution under the *Null Hypothesis* that the ***simpler model is preferable***



The table below gives the value x_0^2 for which $P[x^2 < x_0^2] = P$ for a given number of degrees of freedom and a given value of P.

Degrees of Freedom	Values of P									
	0.005	0.010	0.025	0.050	0.100	0.900	0.950	0.975	0.990	0.995
1	---	---	0.001	0.004	0.016	2.706	3.841	5.024	6.635	7.879
2	0.01	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345	12.838
4	0.207	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277	14.860
5	0.412	0.554	0.831	1.145	1.610	9.236	11.070	12.833	15.086	16.750
6	0.676	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812	18.548
7	0.989	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475	20.278
8	1.344	1.646	2.180	2.733	3.490	13.362	15.507	17.535	20.090	21.955
9	1.735	2.088	2.700	3.325	4.168	14.684	16.919	19.023	21.666	23.589
10	2.156	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209	25.188
11	2.603	3.053	3.816	4.575	5.578	17.275	19.675	21.920	24.725	26.757
12	3.074	3.571	4.404	5.226	6.304	18.549	21.026	23.337	26.217	28.300
13	3.565	4.107	5.009	5.892	7.042	19.812	22.362	24.736	27.688	29.819
14	4.075	4.660	5.629	6.571	7.790	21.064	23.685	26.119	29.141	31.319
15	4.601	5.229	6.262	7.261	8.547	22.307	24.996	27.488	30.578	32.801
16	5.142	5.812	6.908	7.962	9.312	23.542	26.296	28.845	32.000	34.267
17	5.697	6.408	7.564	8.672	10.085	24.769	27.587	30.191	33.409	35.718
18	6.265	7.015	8.231	9.390	10.865	25.989	28.869	31.526	34.805	37.156
19	6.844	7.633	8.907	10.117	11.651	27.204	30.144	32.852	36.191	38.582
20	7.434	8.260	9.591	10.851	12.443	28.412	31.410	34.170	37.566	39.997

MLTSA:

model selection

model selection is also based on the minimization of a quantity. Several quantities are suitable:

AIC Optimism and likelihood maximization on the training set

BIC Bayese theorem

MLD Shannon 1948: A Mathematical Theory of Communication

a theory to find fundamental limits on signal processing and communication operations such as data compression

MLTSA:

AIC, BIC, & MDL

Likelihood: **Model Performance**.

$$\mathrm{AIC} = -\frac{2}{N}\log(L) + \frac{2}{N}k$$

number of parameters:

Model Complexity

Akaike information criterion (AIC) .

Based on $\lim_{N\to\infty}(-2E(\log Pr_{\hat{\theta}}(Y))) = -\frac{2}{N}E\ \log(L) + d\frac{2}{N}$

where $Pr_{\hat{\theta}}(Y)$ is a family of function (=densities) containing the correct (=true) function and $\hat{\theta}$ is the set of parameters that maximized the likelihood *L*

***L* is the likelihood of the data, *k* is the number of parameters,**

***N the number of variables*.**

MLTSA:

AIC, BIC, & MDL

Likelihood: **Model Performance**.

$$\mathrm{AIC} = -\frac{2}{N}\log(L) + \frac{2}{N}k$$

number of parameters:

Model Complexity

Akaike information criterion (AIC) .

Based on $\lim_{N\to\infty}(-2E(\log Pr_{\hat{\theta}}(Y))) = -\frac{2}{N}E\ \log(L) + d\frac{2}{N}$

where $Pr_{\hat{\theta}}(Y)$ is a family of function (=densities) containing the correct (=true) function and $\hat{\theta}$ is the set of parameters that maximized the likelihood *L*

***L* is the likelihood of the data, *k* is the number of parameters,**

***N the number of variables*.**

"-" sign in front of the log-likelihood: AIC shrinks for better models,

AIC ~ k => is linearly proportional to the number of parameters

MLTSA:

AIC, BIC, & MDL

Likelihood: **Model Performance**.

$$\mathrm{BIC} = -2\log(L) + \log(N)k$$

number of parameters:
Model Complexity

Bayesian information criterion (BIC) .

***L* is the likelihood of the data, *k* is the number of parameters,**

***N the number of variables*.**

stronger penalization of complexity (as long as N> e^2*)*

Bayes Factor

The derivation is very different:

$$\frac{P(M_m|D)}{P(M_l|D)} = \frac{P(M_m)}{P(M_l)} \cdot \frac{P(D|M_m)}{P(D|M_l)}$$

MLTSA:

AIC, BIC, & MDL

$$\mathrm{MDL} = -\log(L(\theta)) - \log(L(y|X,\theta))$$

Minimum Description Length (MDL) .

negative log-likelihood of the model parameters (*θ*) and the negative log-likelihood of the target values (*y*) given the input values (*X*) and the model parameters (*θ*).

also: log(L(θ)): number of bits required to represent the model,

log(L(y | X,θ)): number of bits required to represent the predictions on observations

minimize the encoding of the model and its predictions

derived from Shannon's theorem of information

MLTSA:

AIC, BIC, & MDL

$$\text{AIC} = -\frac{2}{N}\log(L) + \frac{2}{N}k$$

$$\text{BIC} = -2\log(L) + \log(N)k$$

$$\text{MDL} = -\log(L(\theta)) - \log(L(y|X,\theta))$$

Mathematically similar, though derived from different approaches. All used the same way: the preferred model is the model that minimized the estimator

implementation https://machinelearningmastery.com/probabilistic-model-selection-measures/

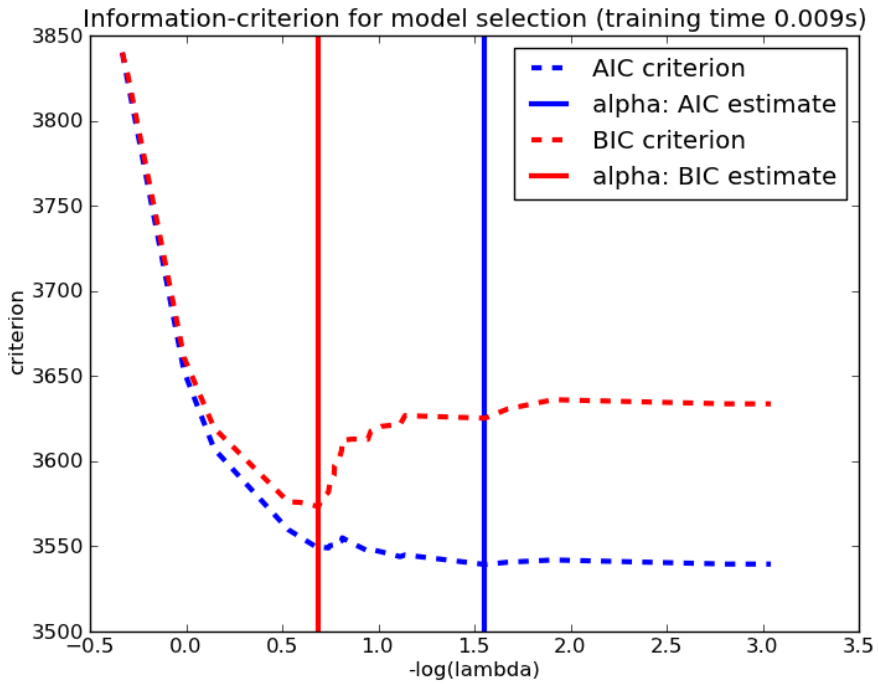
MLTSA:

AIC, BIC, & MDL

$$\mathrm{AIC} = -\frac{2}{N}\log(L) + \frac{2}{N}k$$

$$\mathrm{BIC} = -2\log(L) + \log(N)k$$

$$\mathrm{MDL} = -\log(L(\theta)) - \log(L(y|X,\theta))$$



Data Analysis: A Bayesian Tutorial

https://www.amazon.com/Data-Analysis-Bayesian-Devinderjit-Sivia-ebook/dp/B01BHLXKEI

Elements of Statistical Learning Chapter 7

https://web.stanford.edu/~hastie/Papers/ESLII.pdf

Sorry ARIMA, but I'm Going Bayesian

https://towardsdatascience.com/implementing-facebook-prophet-efficiently-c241305405a3

emcee: The MCMC Hammer

https://arxiv.org/abs/1202.3665

Forecasting at scale
(the facebook Prophet paper)

https://peerj.com/preprints/3190

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

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.

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