Data Analysis & Machine Learning

Checkpoint #4:

Simple Monte Carlo Generation of Pseudo Experiments (“toys”)

Peter Clarke

THE UNIVERSITY of EDINBURGH

Summary of the checkpoint

- Use the random number material to simulate the decay times of a set of 1000 muons in an experimental situation.

- Estimate the lifetime of the muon from the simulated data set and compare it to the true lifetime

- Repeat this process 500 times - i.e. create 500 distinct simulated data sets, where each data set represents 1000 muon decays.

- Plot the distribution of the estimated muon lifetime you get from each of the 500 data sets and interpret this with respect to the known true lifetime to answer:
  - How well (with what numerical precision) can you expect to measure the lifetime from any single experiment ?
  - Is the method biased ? What is the bias and precision on the bias ?

Aims

- This checkpoint will introduce the technique of Monte Carlo simulation.
- You are going to use the random number material to simulate the decay times of a set of 1000 muons in an experimental situation.
- You are going to estimate the lifetime of the muon from the simulated data set and compare it to the true lifetime. This is called a pseudo-experiment – or more often a "toy" experiment
- You are going to repeat this process 500 times - i.e. create 500 distinct simulated data sets, where each data set represents 1000 muon decays.
- You will plot the distribution of the estimated muon lifetime you get from each of the 500 "toy" data sets and interpret this
- You will then add components for background and see how it affects the above results
- You will then add a time acceptance and see how it affects the above results

Simulating 1000 muon decays

- We imagine we have a detector in the laboratory which traps (or stops) muons from the atmosphere

- Once trapped, these muons hang around, and then decay according to an exponential decay time distribution, with a characteristic lifetime τ, as is the case for all fundamental particles.

- This means that the distribution describing muon decay is an exponential function normalised to 1, where the muon lifetime is $\tau = 2.2$ microseconds.

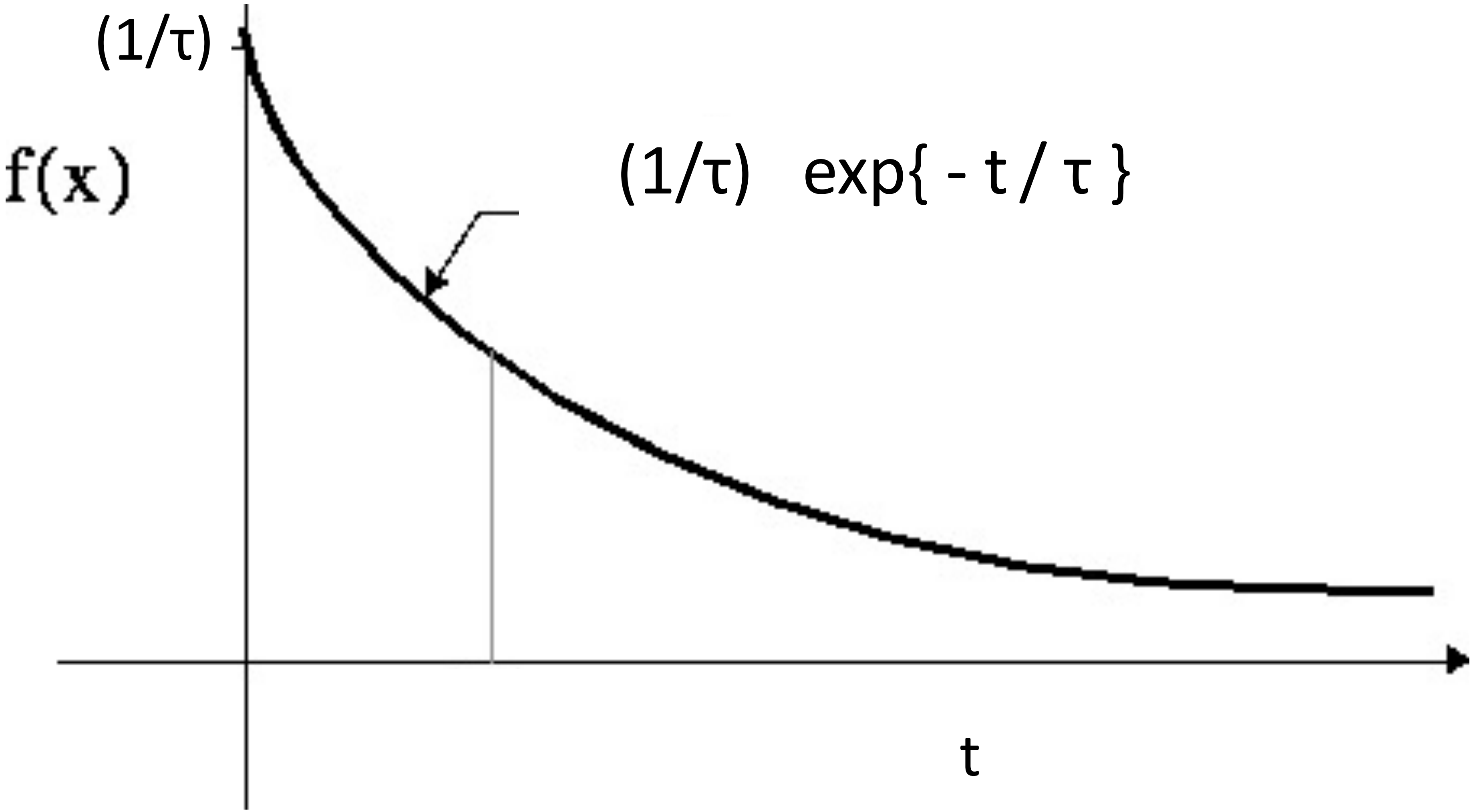
$$P(t_i \mid \tau) = (1/\tau) \exp\{-t_i / \tau\}$$

(we actually call this the "probability density function" but more of this in the next unit)

- You should write code to produce a set of 1000 random muon decay times drawn from such a distribution in the range $0 < t < 10$ microseconds.

- These numbers therefore represent the simulated decay times of a set of 1000 muons

- You should plot the values in a histogram and check that it has an exponential shape.



This function is normalised to 1, i.e. the integral 0->infinity = 1

Estimate the lifetime from the data from a single data set

- We now consider the data set to be as if we had observed it in the real laboratory experiment. This means we have recorded 1000 muon decay times, but of course we wouldn't know what the original muon lifetime was.

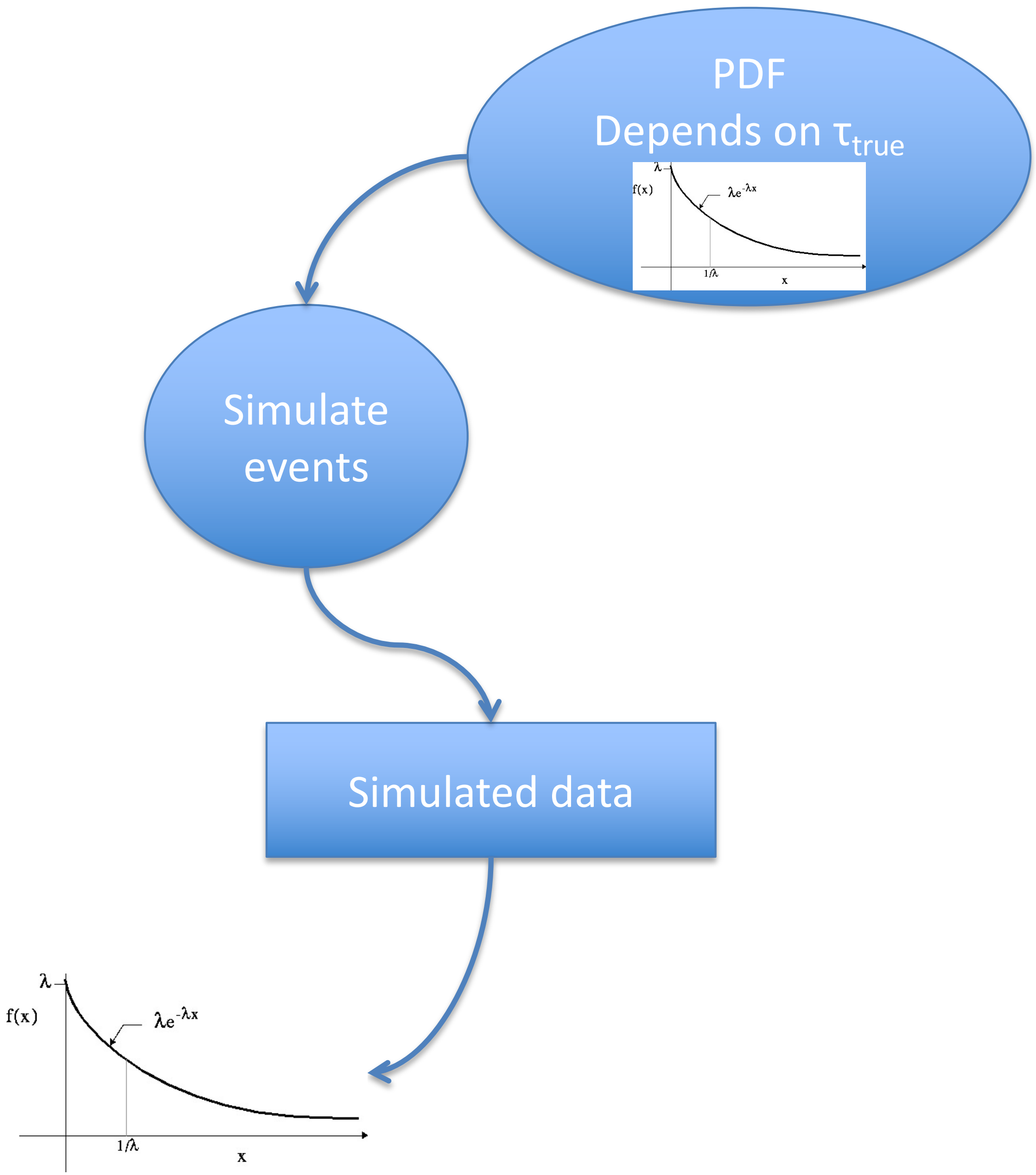
- You should now estimate the lifetime from this single data set . This is easy in the case of an exponential distribution. By definition (this is just maths)

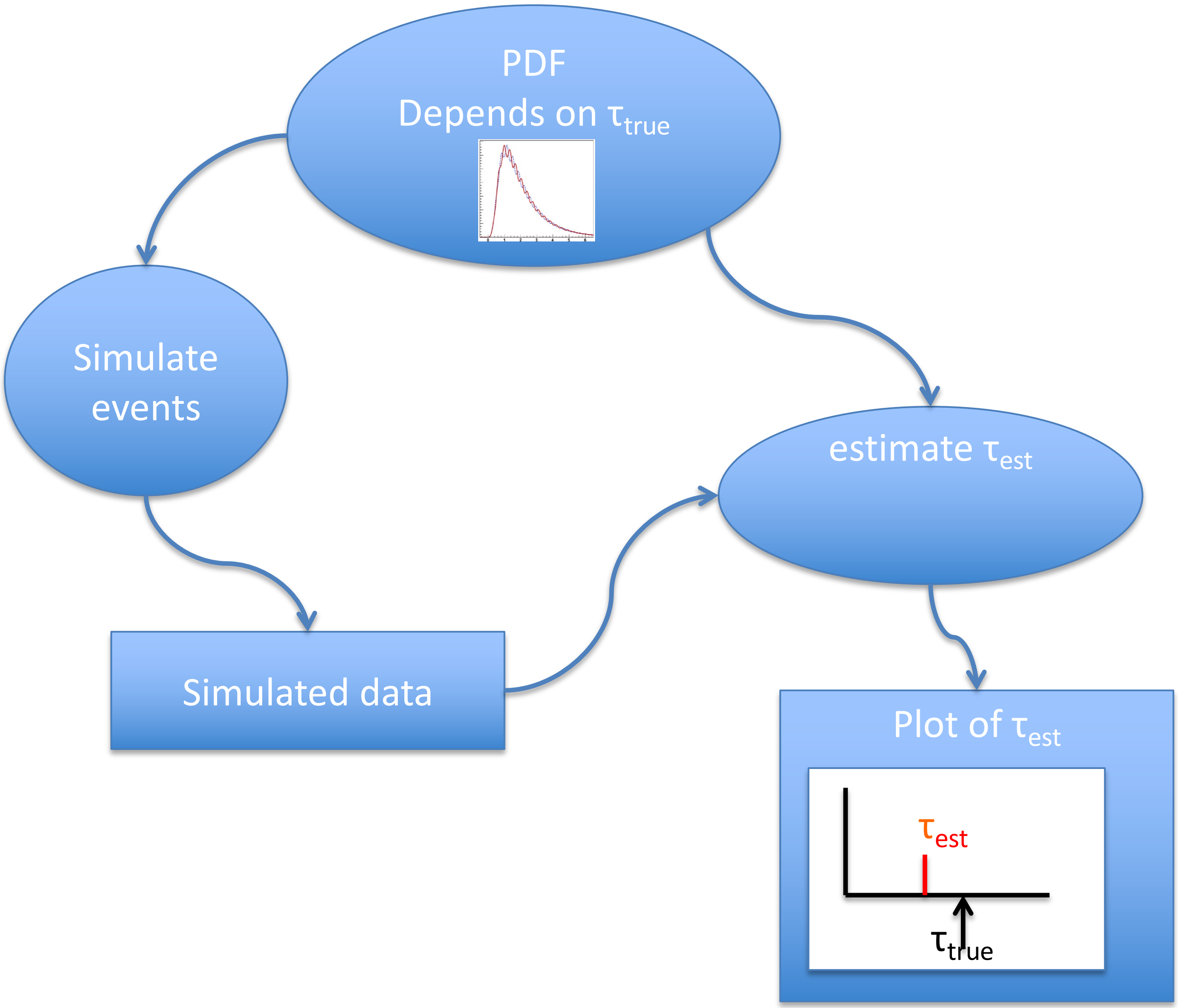
  estimate of τ = the average of all measured decay times

- Compare the value you find with the known original value of 2.2 microseconds.

- Have a think about why are they probably not exactly the same ? You will probably get a values a bit less than 2.2

This is the process in pictures



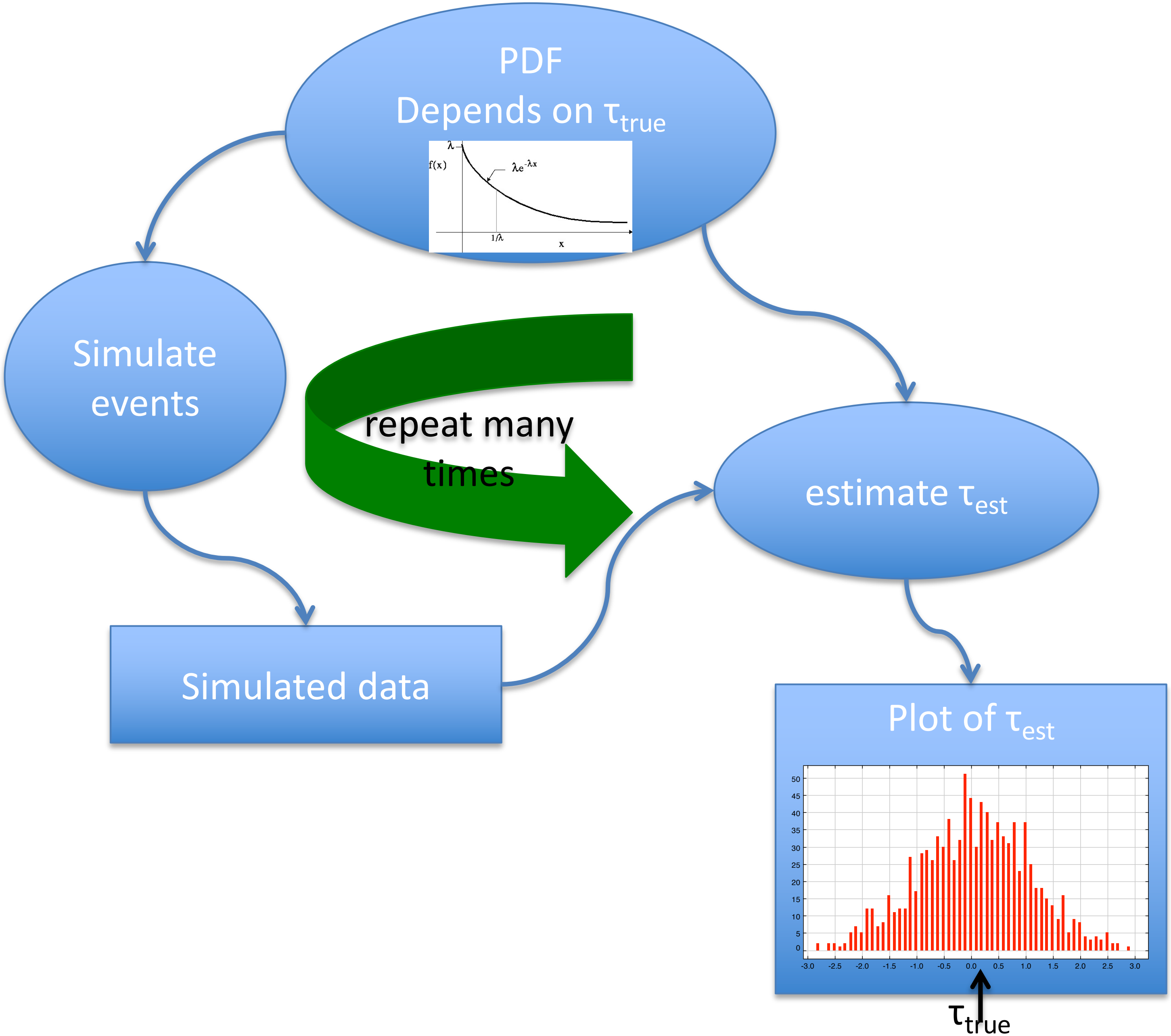

PDF
Depends on τ_{true}
Simulate events
Simulated data
estimate τ_{est}
Plot of τ_{est}
τ_{est}
τ_{true}

Repeat 500 times

- Repeat this whole process to simulate the whole experiment 500 times.
  - In each case obtain an estimate of τ .
  - Plot the distribution of all the values of τ which you obtain

- What do you learn from this distribution ?
  - How well (with what numerical precision) can you expect to estimate the true lifetime from any <u>single experiment</u> ?
  - Is the method biased ? what is the bias and precision on the bias
  - To answer this you need to understand the difference between the following quantities:
    - The mean of a Gaussian distribution
    - The standard deviation of the distribution (RMS deviation) and what it tells you
    - The error on the knowledge of the mean (hint is is NOT the standard deviation)

This is the process in pictures



Summary of the checkpoint

- Use the random number material to simulate the decay times of a set of 1000 muons in an experimental situation.

- Estimate the lifetime of the muon from the simulated data set and compare it to the true lifetime

- Repeat this process 500 times - i.e. create 500 distinct simulated data sets, where each data set represents 1000 muon decays.

- Plot the distribution of the estimated muon lifetime you get from each of the 500 data sets and interpret this with respect to the known true lifetime to answer:
  - How well (with what numerical precision) can you expect to measure the lifetime from any single experiment ?
  - Is the method biased ? What is the bias and precision on the bias ?

- Notes:
  - You are allowed to either write your own code to generate events, or use built in numpy functions.
  - You can use numpy (or others functions) as you wish to do statistical calculations
  - You cannot generate lifetimes up to infinity, so curtail the time range from 0 -> 10 times the muon lifetime

The following material is stretch material.
It is NOT part of the checkpoint assessment

This is simple Monte Carlo simulation

- What you will have done by this point is to make 500 **“Toy Monte Carlo”** simulations of the experiment.

- You have used these to estimate the “precision capability” of your detector and to see if your process is “biased”

- In this case the MC simulations are particularly simple.

- We could add all sorts of complexity to this :
  - Simulation of the imperfection of the detector
  - Simulation of geometry of detector causing some events to be lost
  - Simulation of efficiency of detectors (e.g a photo multiplier)
  - Addition of background noise
  - ..... and so on

Adding background

- The events were generated assuming a “perfect situation”

- In all real particle physics experiments you never have perfect data. You generally “select” some events using some criteria, but inevitably some fake events get included. We call these background.

- In this situation there are 2 sorts of likely background
  - Events that decayed at time=0 but due to finite detector resolution get measured to have a narrow Gaussian distribution centred on time=0
  - Events which decay with an exponential lifetime, but which are not muons, and so have a different lifetime

- Add each of these in turn to your function P(t) which you use to generate events. This means adapting your random generation to sample from a linear combination of two distributions. Typically you code this, for a fraction X, as

$$P(t) = X * P1(t) + (1-X)*P2(t))$$

where

P1(t) is the exponential you have already constructed

P2(t) is the background Gaussian/exponential

Adding background

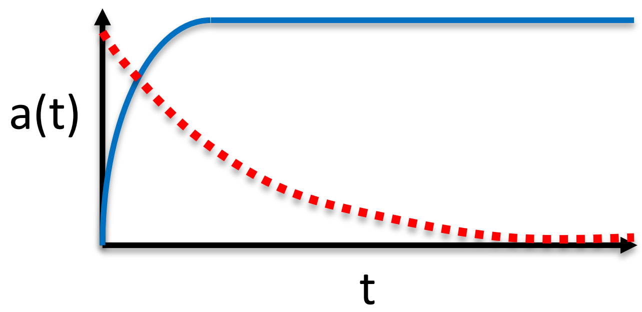
- Perform the Monte Carlo study again - where you still estimate the lifetime from the average (i.e. we are pretending there is a background in nature, but we don’t know about it when we are interpreting the data to measure the lifetime).

- In each case, determine again
  - The precision of measuring the lifetime
  - Is the method biased

Adding a time acceptance

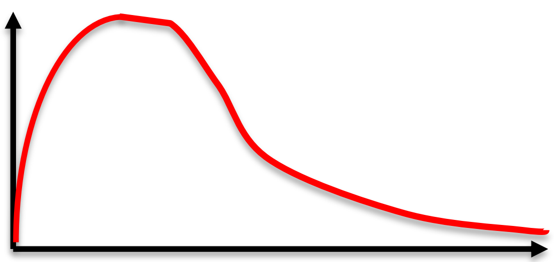
- The events were generated assuming a “perfect detector”
- In all real particle physics experiments you never have a perfect detector
- Al detectors (here of muons) will have some imperfect efficiency for detecting the muons as a function of time. Maybe the detector is poor at detecting decays near time=0, but gets better once the decay time is a bit longer.
- This is called a time efficiency, or more often a time acceptance a(t).
- Example:

$a(t) = 1 - \exp(-t/G)$ →



where G is just some short timescale compared to the Muon Lifetime

What you observe in the experiment is then the “product”



Adding a time acceptance

- To simulate this, you generate events from a modified function

  $P(t) \rightarrow P(t) * a(t)$

- Add this to your code - and choose a suitable value of G, say 10% of the muon lifetime

- Perform the Monte Carlo study again - where you still estimate the lifetime from the average (i.e. we are pretending there is a time acceptance in nature, but we don't know about it when we are interpreting the data to measure the lifetime).

- In each case, determine again
  - The precision of measuring the lifetime
  - Is the method biased