STAT analysis basics

Yi Chen (MIT)
Jul 23, 2020, JETSCAPE summer school

Co-instructor: James Mulligan, Raymond Ehlers
And with help from many people

The MIT group's work is supported by US DOE NP
The JETSCAPE collaboration is supported by National Science Foundation under Cooperative Agreement ACI-1550300

Discussions + questions

This is the slack channel to ask questions and followups

#bayesian-chen

As was mentioned, if you see a problem you also are having, please press thumbs-up so that we know how many people have the same problem

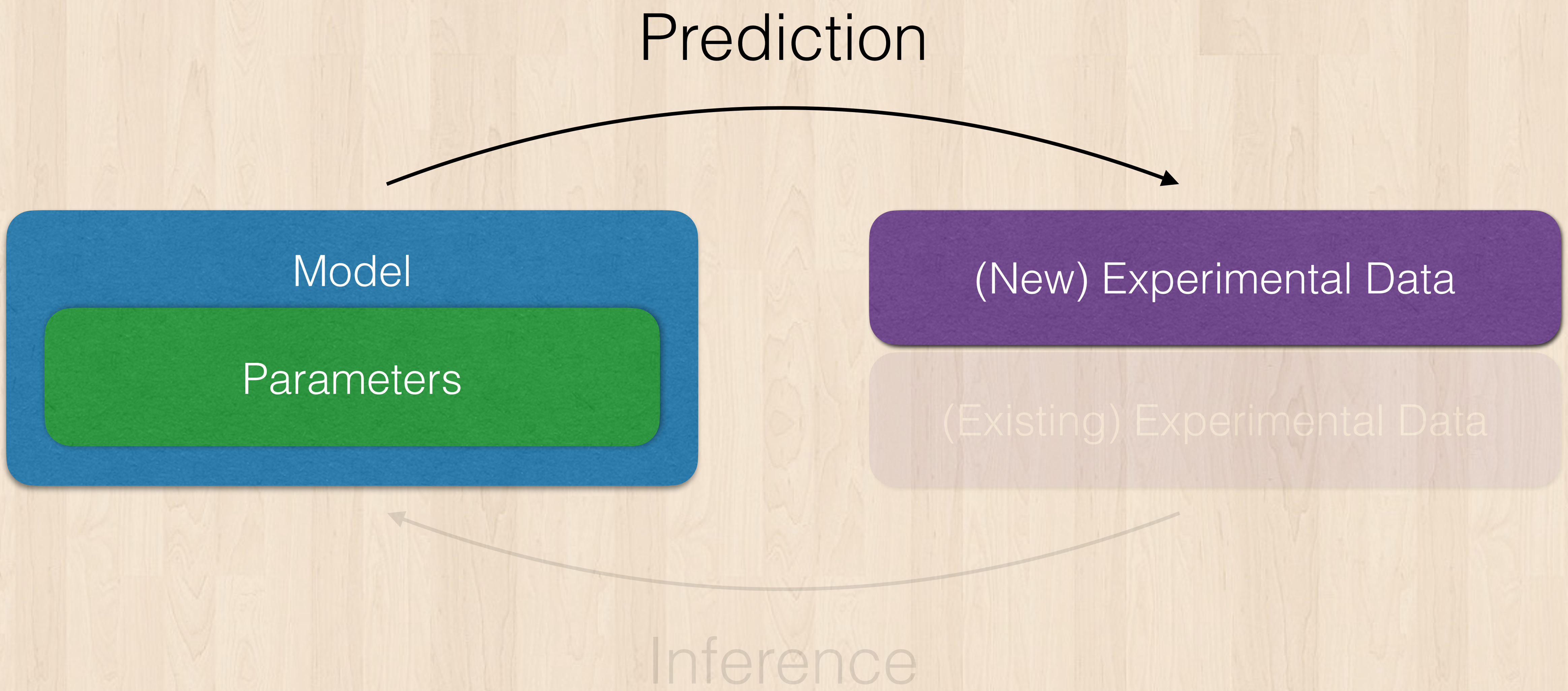
Outline

- What is the problem we want to solve?
- What are the **likelihood & posterior probability functions**
- **What can we do**, once we have the function
- **How can we obtain** the function?
- Putting things together! => Hands-on session
- **Dealing with data** (Friday)

- My first goal today is to give you **the big picture** with the statistical analysis
- Then we do an actual hands-on exercise to materialize all these using the JETSCAPE statistical analysis package

The goal

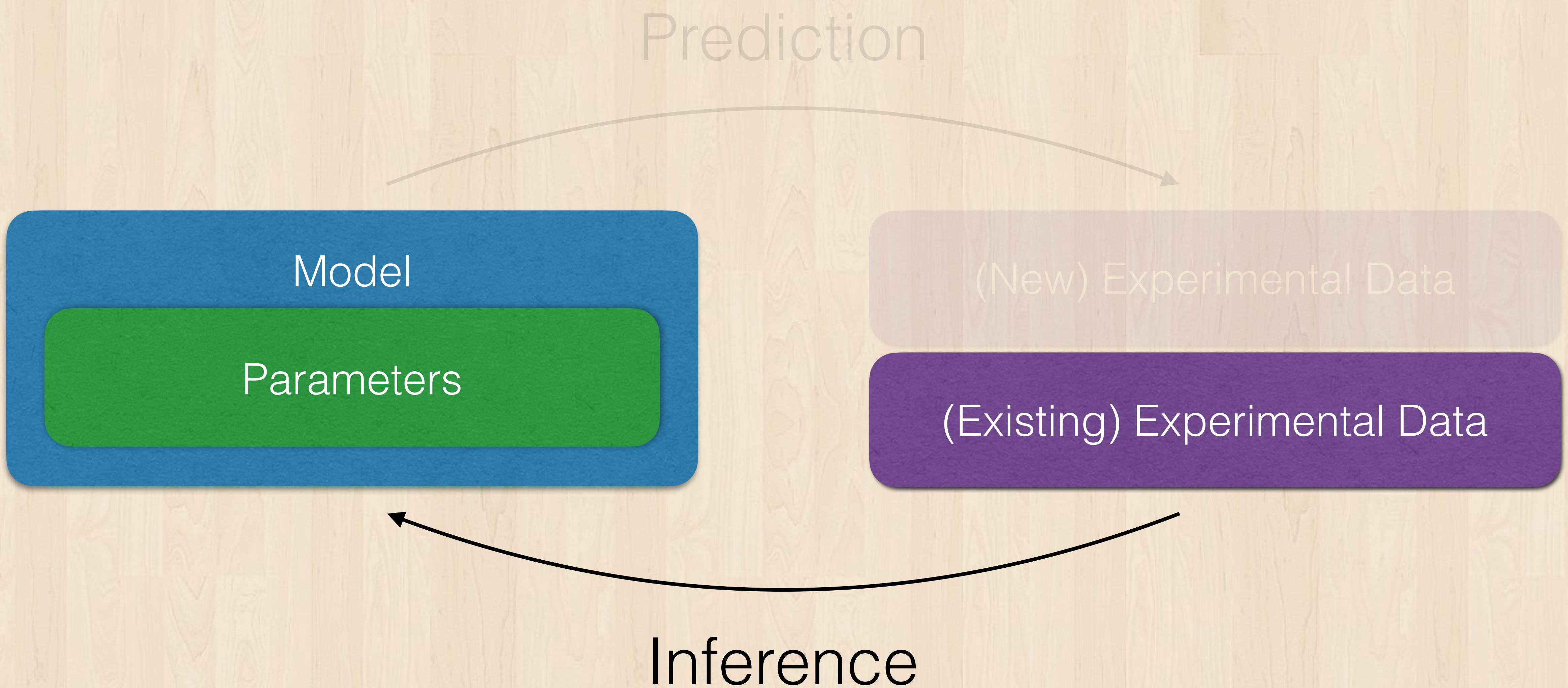
The goal



Given some models with some set of parameters, we can make predictions for experiments.

But what if we do not know the best parameters?

The goal



How can we infer what parameters fit the data best if we already have some experimental data?

The likelihood function

What is likelihood

Likelihood function = how “likely” a set of parameter is, relative to other parameter values, given observed data

$$\mathscr{L}(\overrightarrow{\theta} \mid \overrightarrow{x})$$

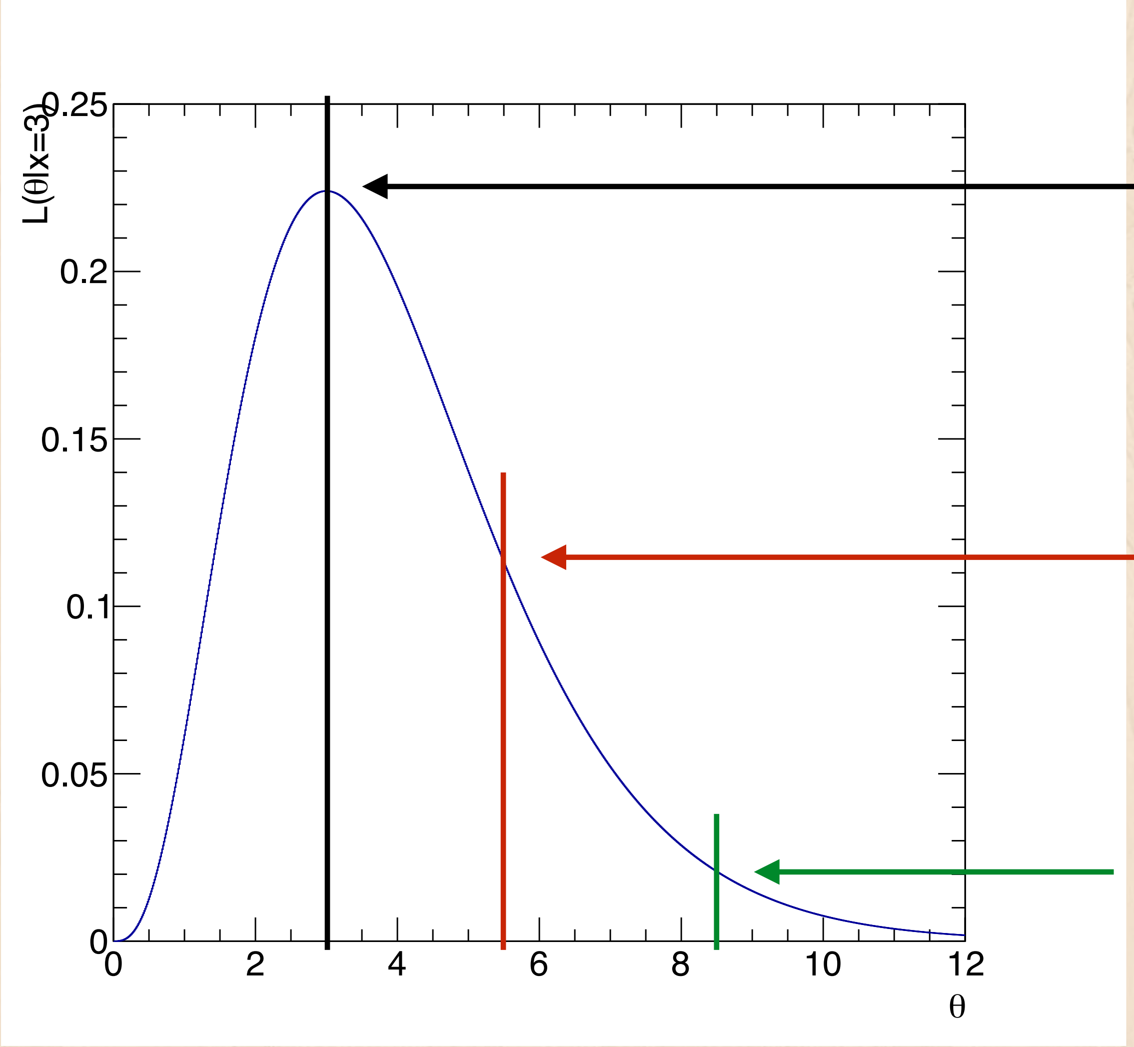
Parameters of interest   Observed data

It’s a function of the parameters of interest, conditioning on what we see in data

Example: counting experiment

Suppose we count number of events and data says 3

$$\mathcal{L}(\theta \,|\, x = 3)$$



Most likely true count is 3

5.5 is about “half as likely”

8.5 is about “10% as likely”

Example: counting experiment

If the expected count is θ, we can write the probability of observing a count of x as

$$P(x\,|\,\theta) = \text{Poisson}(x\,|\,\theta) = \frac{\theta^{x} e^{-\theta}}{x!}$$

$\boxed{\text{Function of x, given θ}}$

We write the likelihood function as

$$\mathscr{L}(\theta\,|\,x) \equiv \frac{\theta^{x} e^{-\theta}}{x!}$$

$\boxed{\text{Function of θ, given x}}$

More examples

$$\mathscr{L}(\vec{\theta}\,|\,\vec{x}) \quad \text{vs} \quad P(\vec{x}\,|\,\vec{\theta})$$

θ = theory parameter	**x = observed data**
Probability of head per flip	Heads/tail sequence of coin throw
Higgs boson cross section	Number of events around 125 GeV
Neutrino interaction cross section	Counts in water tank
Energy loss parameter for Partons	Hadron R_{AA}
……	……

Bayesian school of thinking

How to connect the two things?

$$\mathscr{L}(\overrightarrow{\theta} \mid \overrightarrow{x}) \text{ vs } P(\overrightarrow{x} \mid \overrightarrow{\theta})$$

One is function of θ, one is function of x

One way to go is through the Bayes' theorem with the posterior probability function *

$$P(\overrightarrow{\theta} \mid \overrightarrow{x})$$

* Likelihood and Bayesian posterior are not exactly the same thing; the distinction is beyond the scope of this lecture

Bayesian school of thinking

Starting from the probability equality

$$P(a,b) = P(a\,|\,b)P(b)$$

We can “write”

$$P(\vec{x}, \vec{\theta}) = P(\vec{x}\,|\,\vec{\theta})P(\vec{\theta}) = P(\vec{\theta}\,|\,\vec{x})P(\vec{x})$$

or

$$\boxed{P(\vec{\theta}\,|\,\vec{x}) = \frac{P(\vec{x}\,|\,\vec{\theta})P(\vec{\theta})}{P(\vec{x})}}$$

Bayes’ theorem

Bayesian school of thinking

Prior knowledge of how likely given parameter is true

Posterior probability

$$P(\vec{\theta} \mid \vec{x}) = \frac{P(\vec{x} \mid \vec{\theta})P(\vec{\theta})}{P(\vec{x})}$$

Probability of observing “x”, generally speaking

Since experiment is done, P(x) is a constant

$$P(\vec{\theta} \mid \vec{x}) \propto P(\vec{x} \mid \vec{\theta})P(\vec{\theta})\Big|_{\text{given fixed x}}$$

Prior knowledge

- The Bayesian formalism always involves a “prior” $P(\theta)$, which encodes our prior knowledge on how θ distributes
- It’s both a blessing and a curse — our outcome will always be “biased” by what we know before
- Setting $P(\theta) = 1$ gives us back to the simplest case
  - However $P(\theta) = 1$ does not mean unbiased (why not $P(\theta^2) = 1$? $P(\ln \theta) = 1$?)

Small Bayesian exercise

If it rains, the ground is wet 100% of the time

If it does not rain, the ground is wet 10% of the time

Forecast says 65% chance of rain right now

Given that we see the ground is wet, what is the probability that it is actually raining?

$$P(\vec{\theta} \mid \vec{x}) \propto P(\vec{x} \mid \vec{\theta}) P(\vec{\theta})$$

Press "**yes**" if you get it, "**no**" if you are not sure

Small Bayesian exercise

$$P(\text{wet} \mid \text{rain}) = 100\%$$
$$P(\text{wet} \mid \text{no rain}) = 10\%$$

$$P(\text{rain}) = 65\%$$

$$P(\text{rain} \mid \text{wet}) \propto P(\text{wet} \mid \text{rain})\, P(\text{rain}) = 0.65$$
$$P(\text{no rain} \mid \text{wet}) \propto P(\text{wet} \mid \text{no rain})\, [1 - P(\text{rain})] = 0.035$$

$$P(\text{rain} \mid \text{wet}) = 0.65 / (0.65 + 0.035) = 94.9\%$$

Small Bayesian exercise

P(wet | rain) = 100%
P(wet | no rain) = 10%

| What if forecast says 5%? | → P(rain) = 5%

$P(\text{rain} \mid \text{wet}) \propto P(\text{wet} \mid \text{rain})\, P(\text{rain}) = 0.05$

$P(\text{no rain} \mid \text{wet}) \propto P(\text{wet} \mid \text{no rain})\, [1 - P(\text{rain})] = 0.095$

$P(\text{rain} \mid \text{wet}) = 0.05 / (0.05 + 0.095) = 34.5\%$

Small Bayesian exercise

P(wet | rain) = 100%
P(wet | no rain) = 10%

What if forecast says 5%? → P(rain) = 5%

P(rain | wet) ∝ P(wet | rain) P(rain) = 0.05
P(no rain | wet) ∝ P(wet | no rain) [1 - P(rain)] = 0.095

Our view of what is happening is affected by prior knowledge. There is no “unbiased” P(rain)

Further reading: uninformative prior, Jeffreys prior

Updating knowledge

Another way to think about the Bayes' formalism is to refine our knowledge about the problem

$$P(\overrightarrow{\theta} \mid \overrightarrow{x}) \propto P(\overrightarrow{x} \mid \overrightarrow{\theta})P(\overrightarrow{\theta})$$

What we know afterwards

What we know before

$$P(\overrightarrow{\theta} \mid \overrightarrow{x}, \overrightarrow{x}_{\text{past}}) \propto P(\overrightarrow{x} \mid \overrightarrow{\theta}, \overrightarrow{x}_{\text{past}})P(\overrightarrow{\theta} \mid \overrightarrow{x}_{\text{past}})$$

Everything is conditioning on our past knowledge

Likelihood: recap

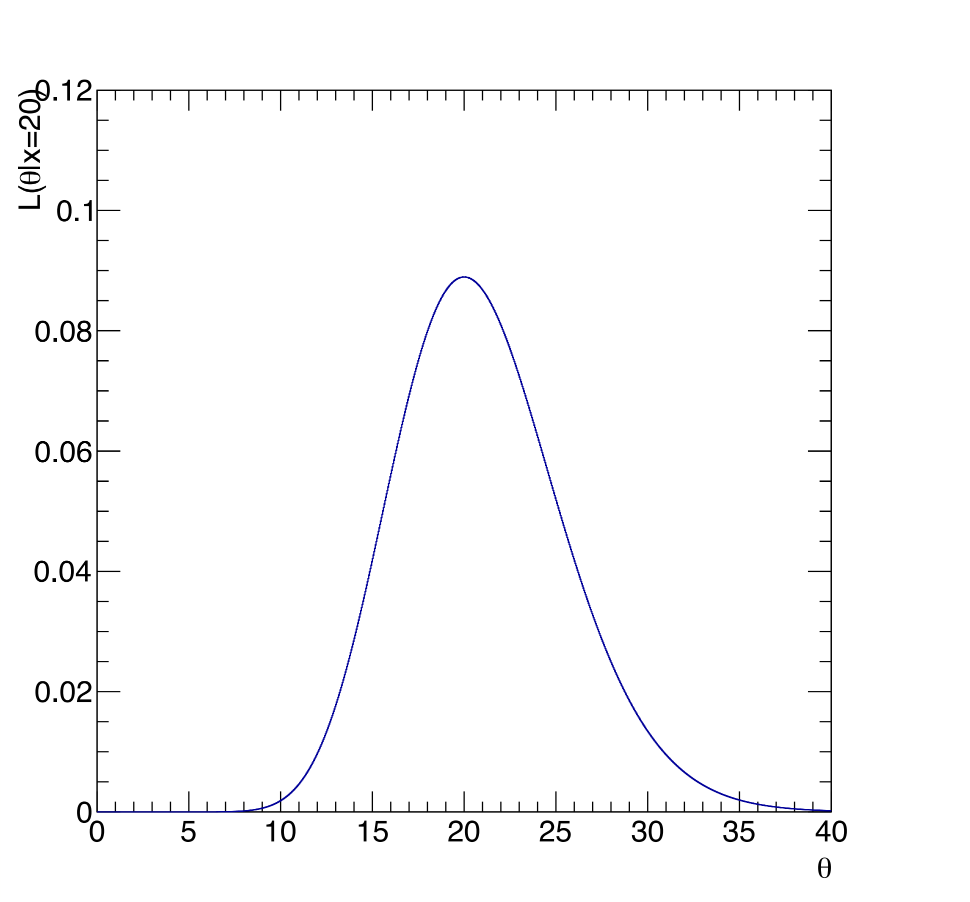
- The likelihood function $L(\theta|x)$ gives us the relative degree of likelihood as a function of θ, given observed data x
- We can write the posterior $P(\theta|x)$ in terms of the probability distribution $P(x|\theta)$, which is the probability of observing data x, given θ
  - With a prior term $P(\theta)$ — there is no “universally unbiased” choice

What can we do with the function?

"Description"

The simplest thing we can do is to **describe** the function

$P(\vec{\theta} \mid \vec{x}) \longrightarrow$ mean, RMS, most probable point, …



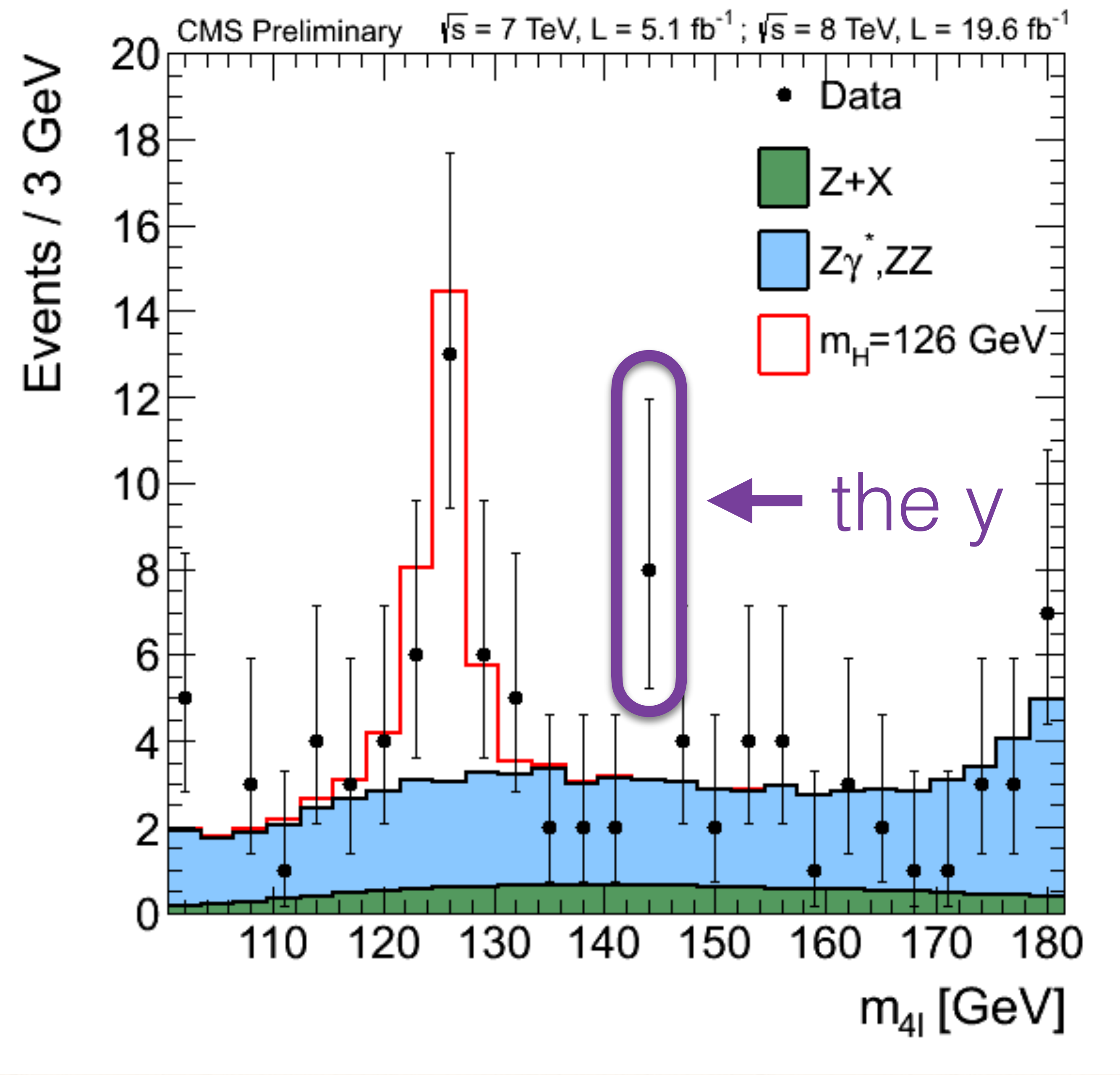
Example: counting, x = 20

What	Value
Most probable	20
Mean	20.99
RMS	4.57 (~sqrt(20))
Skewness	0.41

“Description”

- Uncertainty, too, is a description
- The big take home message: in experimental physics, **everything is always a distribution, and the numbers we quote are descriptions that characterizes the underlying function**
- When we say we measure 25 ± 5, we are **describing** the underlying function
    - For example it could mean that most probable value is 25, and the 68.27% most likely interval is [20, 30]
    - Or it could mean that the range [20, 30] has likelihood value above 1/e (~37%) of the peak value
    - Or that the RMS of the distribution is 5, …etc etc

Is this “θ” or “x”?

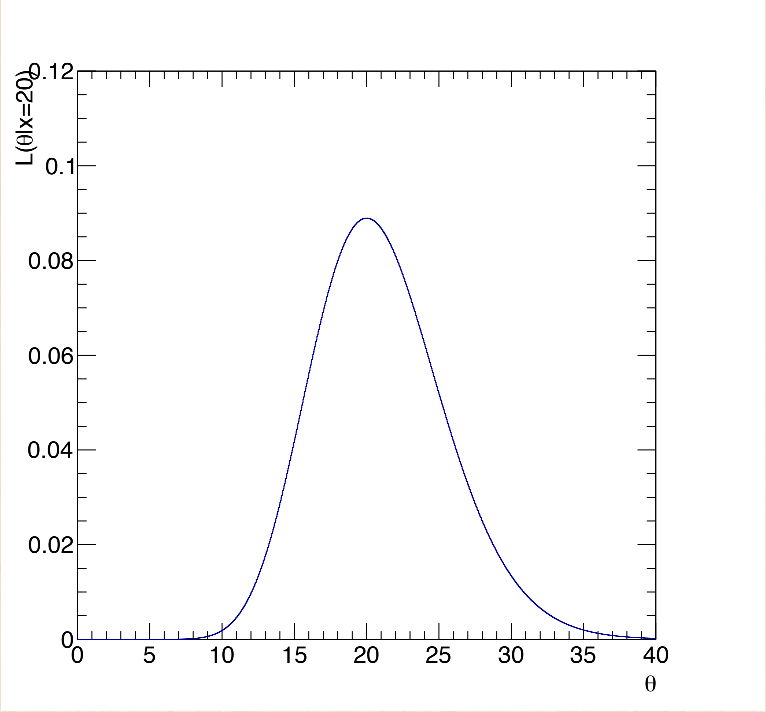


Press “**yes**” for θ
Press “**no**” for x

What else can we do?

- If we are lucky enough to write down the analytical form, things are easy — we can derive many things
- What if we have a large number of parameters? Or function evaluation is slow?
- We build the function **numerically**
  - For example to get the Higgs CP property, we have the Higgs mass shape, anomalous couplings, etc
  - CMS “MD” method in 4l: 12 observables, 10+ parameters
  - 12D integral, a few 12x12 Jacobian’s, etc. => O(1s) / evaluation
- One potential way is to **create a set of samples that distributes according to the posterior function**

Sampling



Sampling →

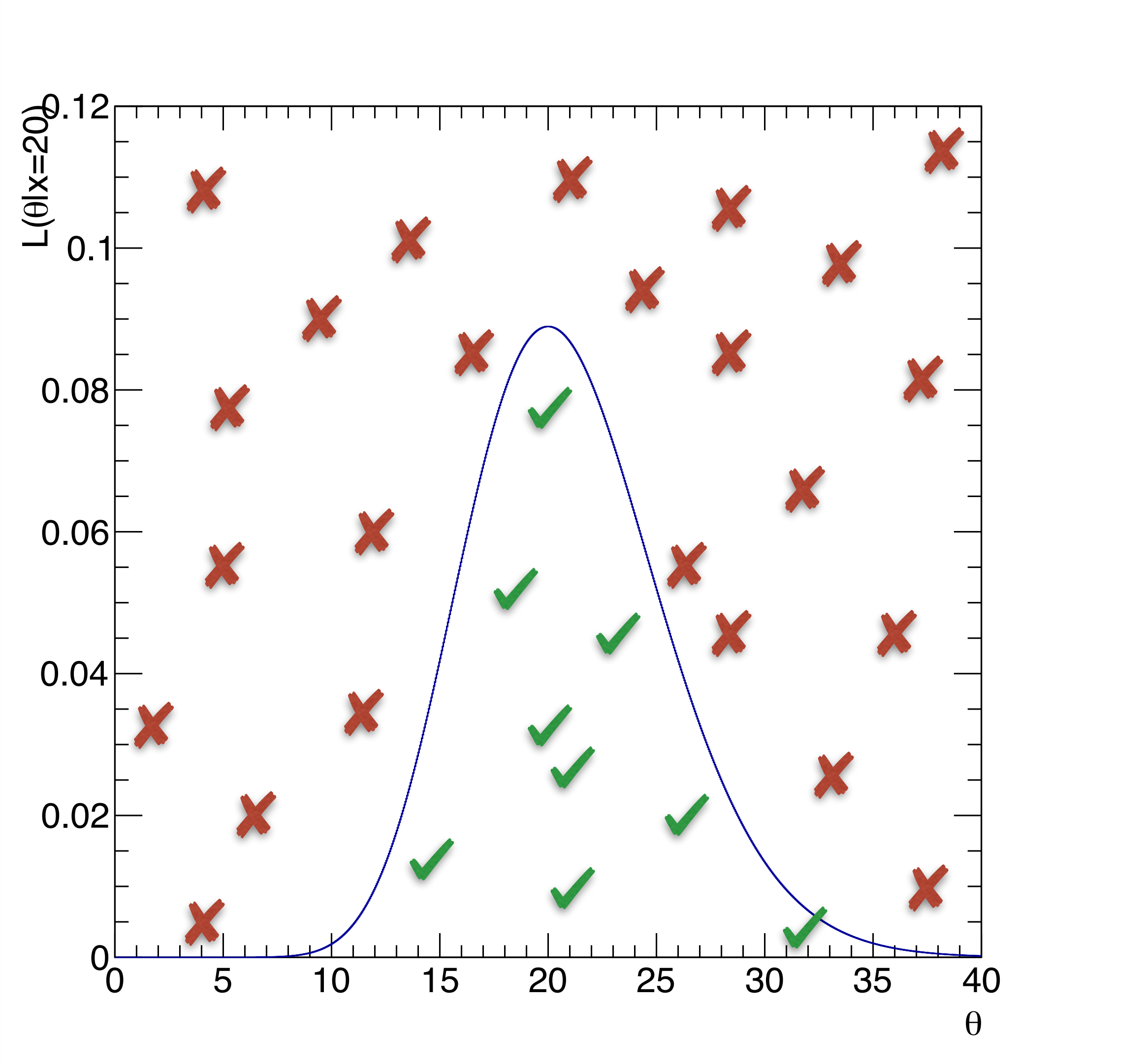
{20.1, 20.9, 17.5, 15.2, 30.8, 19.7, 20.3, …}

← Histogram

In other words, we create a large set of numbers, when plotted as a histogram, gives us back the function

Then we can analyze the samples without worrying too much about the costly calculations

Sampling: how?



Conceptually simplest way: shoot darts

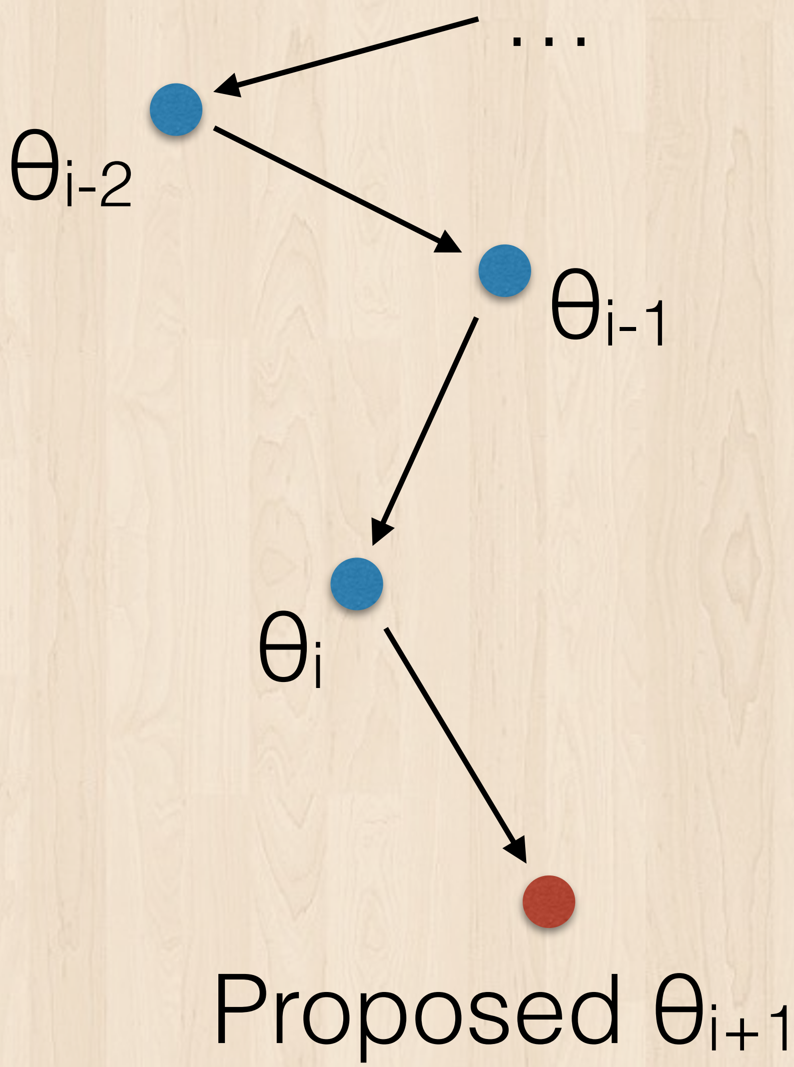
Randomly pick points on the plot, and collect the θ values of the ones falling under the curve

This would work — though not necessary efficient

MCMC

- Markov-Chain Monte-Carlo (MCMC) is another way to achieve the same thing
- It walks through the phase space with some special algorithm: θ_i —> θ_{i+1} —> θ_{i+2} —> …
  - Such that asymptotically, $\{\theta_i\}$ approaches $P(\theta|x)$
  - We call $\{\theta_i\}$ a “chain”. Ie., a chain of samples

ex. Metropolis algorithm



1. Pick a proposed location to move to
2. Evaluate likelihood $P(\theta_i)$ and P(proposal)
   A. If $P(\theta_i)$ < P(proposal), go
   B. Otherwise, throw a dice to see if we move — with probability P(proposal)/$P(\theta_i)$

Direct consequence: samples are very correlated!

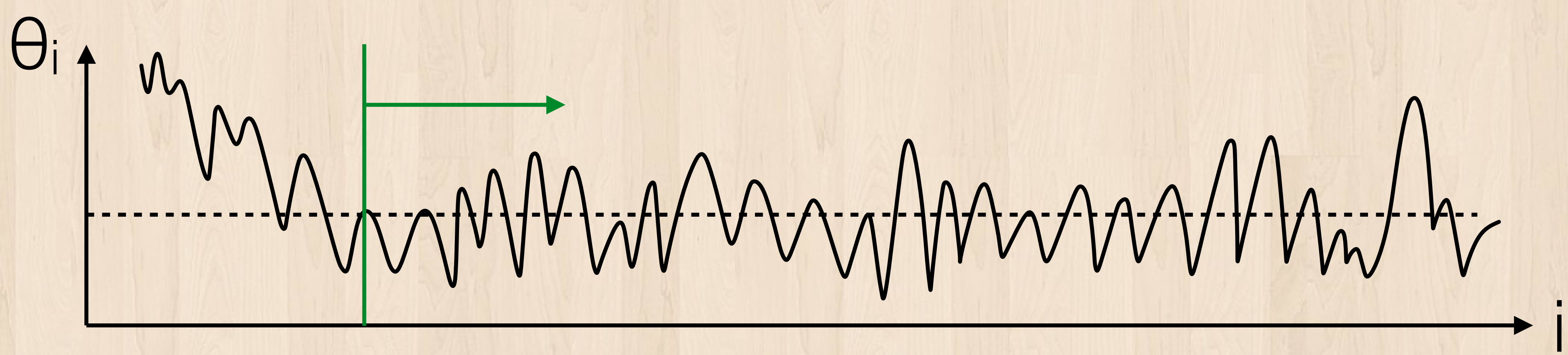
{…, 1, 1, 1, 2, 2, 5, 3, …}

“Burn-in”

- The MCMC will only approach the desired distribution <u>asymptotically</u>

- In other words, when we let the chain go on for a long time, eventually, it will approach the distribution

- This also means that the initial steps do not necessarily follow the posterior



Analyzing likelihood: recap

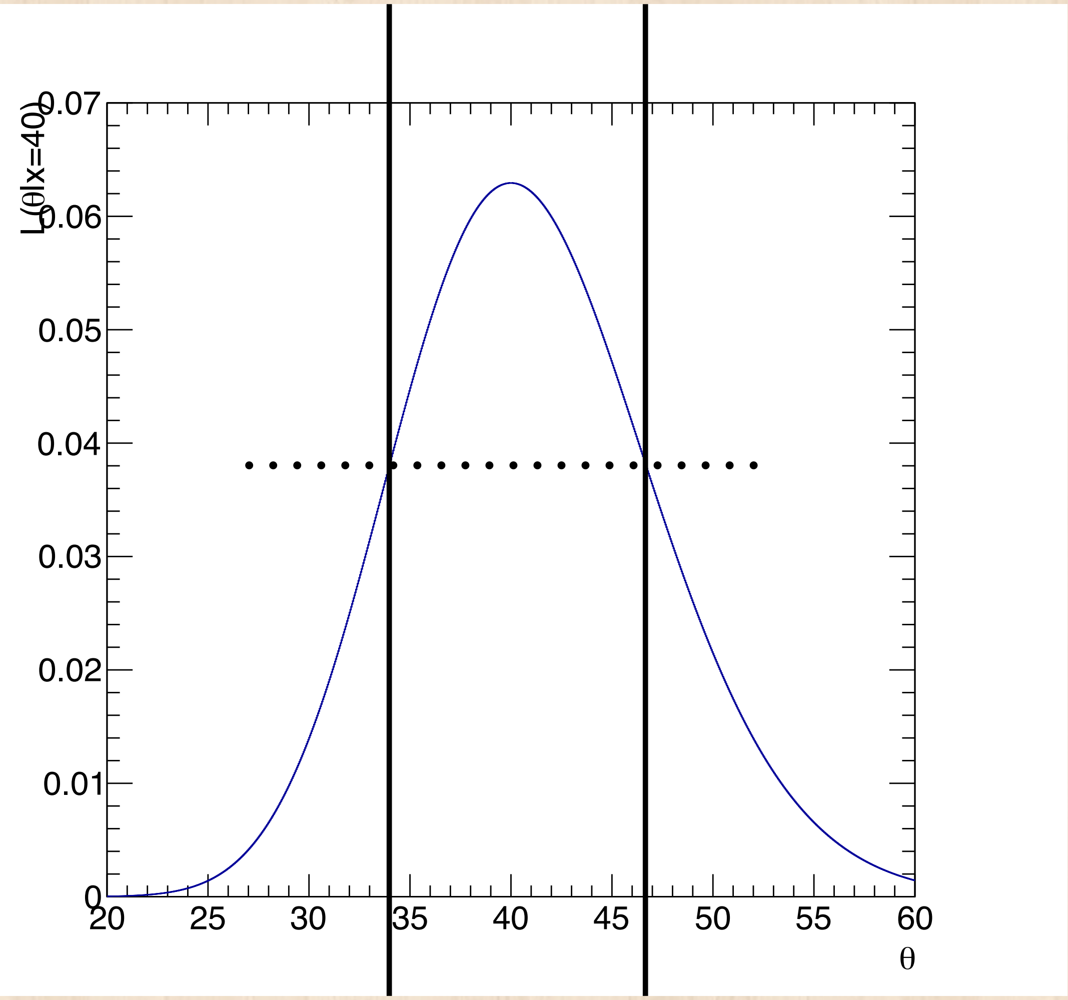
- Once we have built the posterior function, we can proceed to analyze it
- We can quote numbers to describe the function
  - **The numbers we quote in experimental physics are descriptions of the likelihood**
- We can also create samples and analyze them statistically
  - Throw darts, MCMC, …

How do you build the posterior function?

Simplest case: counting

- In counting experiments, $P(x|\theta) = \text{Poisson}(x|\theta)$
- So we can write down $P(\theta|x)$ analytically in a straightforward manner
- Then we just analyze the function

Example: "uncertainty" with x = 40



Let's define "uncertainty" to be the most likely region of θ that encloses 68.27% of the area of the curve *

We can do the math and conclude [33.99, 46.68]

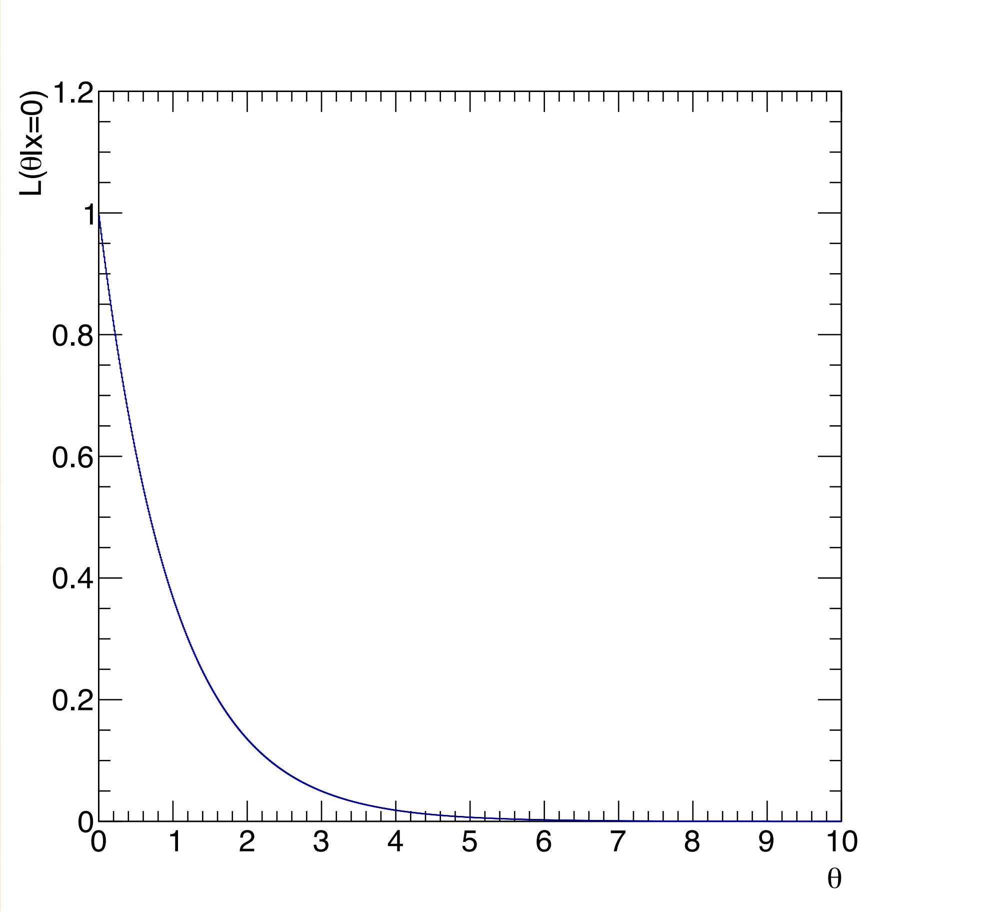
Exercise: what is the "uncertainty" with x = 0?

Press "**yes**" if you get it, "**no**" if you are not sure

* Note there are many many prescriptions to define "uncertainty"

Exercise: “uncertainty” on 0?

$$P(\vec{\theta} \mid \vec{x})\Big|_{x=0} = \frac{P(\vec{x} \mid \vec{\theta})P(\vec{\theta})}{P(\vec{x})}\Big|_{x=0} \propto \theta^x e^{-\theta}/x!\Big|_{x=0} = e^{-\theta}$$



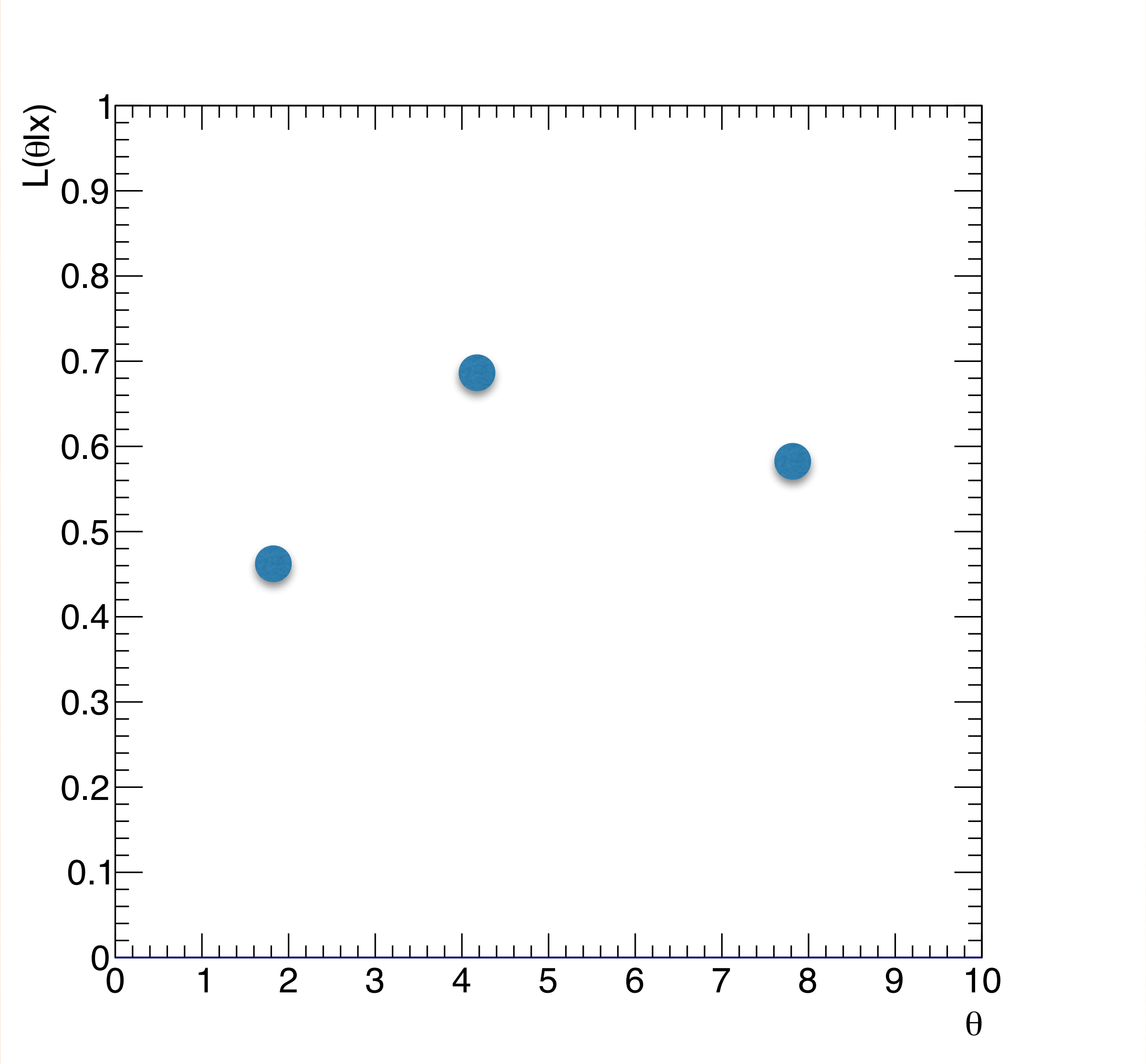
$$0.6827 = \frac{\int_0^{\theta_0} e^{-\theta} d\theta}{\int_0^{\infty} e^{-\theta} d\theta}$$

$$\downarrow$$

$$\theta_0 = ?$$

Computing-intensive case

What can we do, if likelihood on one point takes weeks-months to calculate on a computing cluster?

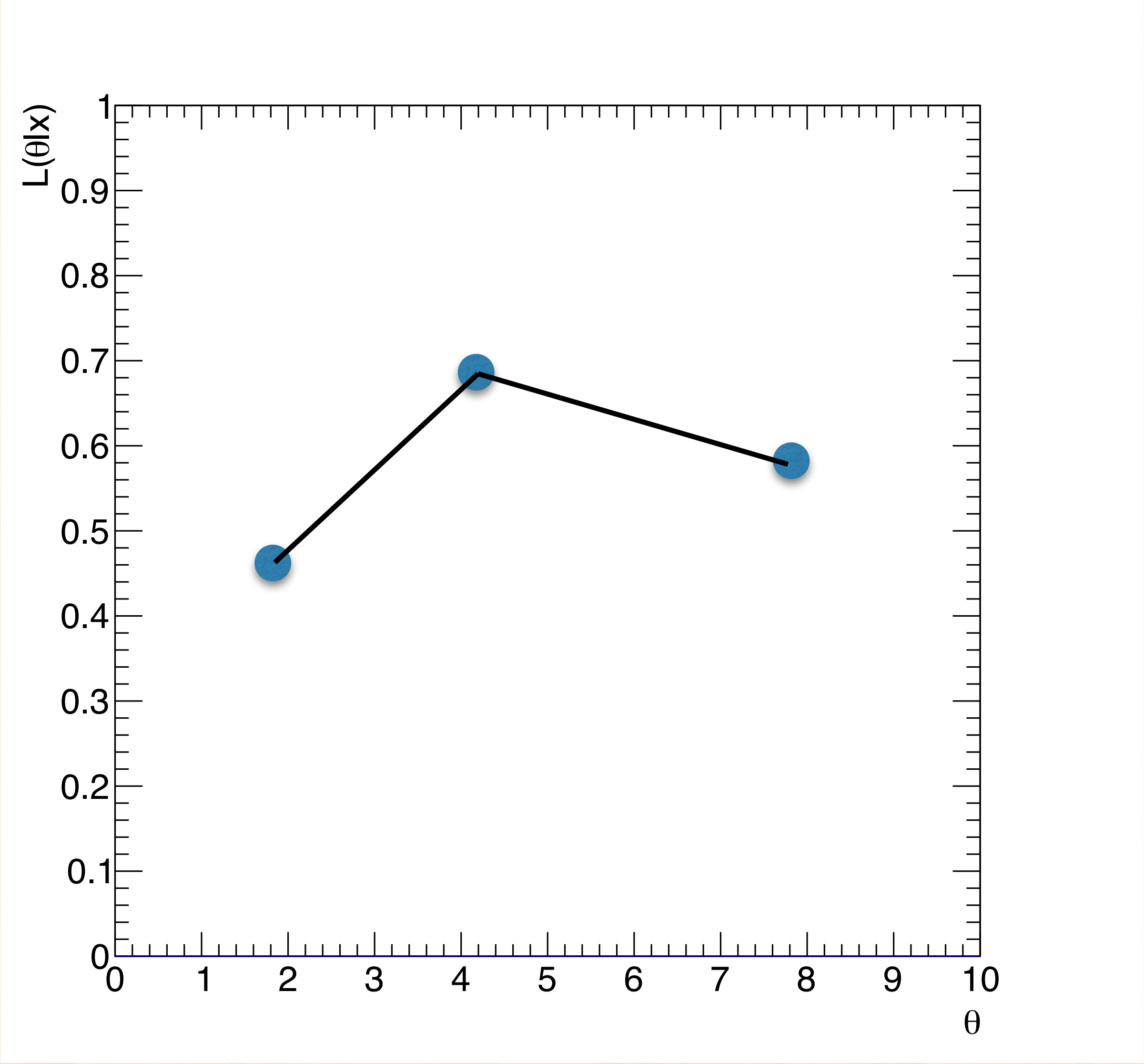


Each of these points take a month to get

MCMC won't work well with this latency

Computing-intensive case

We pick nicely spaced points ("design points"), evaluate the likelihood on those, and interpolate



If the points are picked well, the interpolated function should approach L(θ|x)

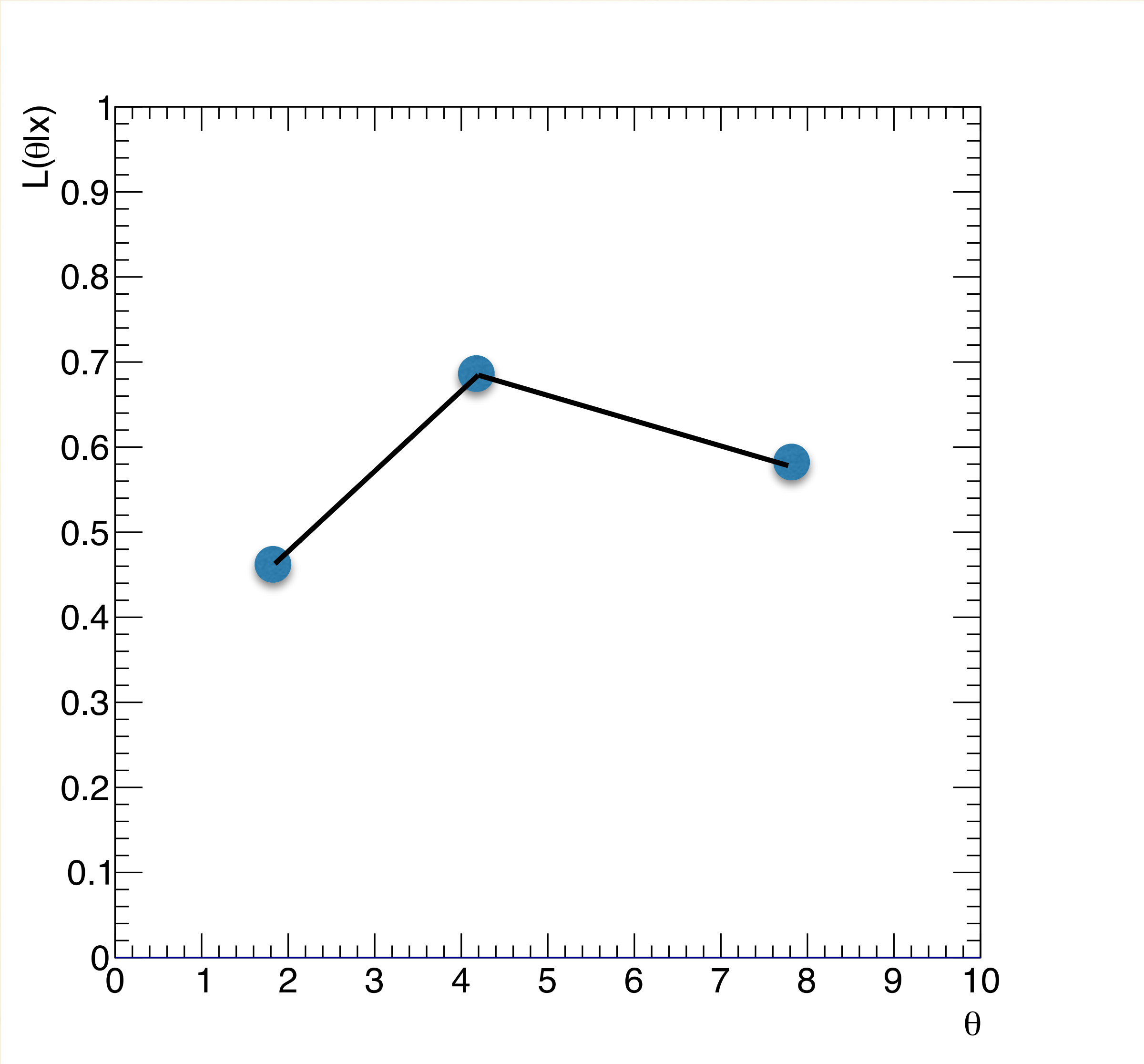
"Latin hypercube": an algorithm to sample N dimensional space ~uniformly

How to interpolate?

<u>Straight line / spline:</u>
works ~well for 1D

Generalization to more dimensions not straightforward

But not ideal because of kinks etc

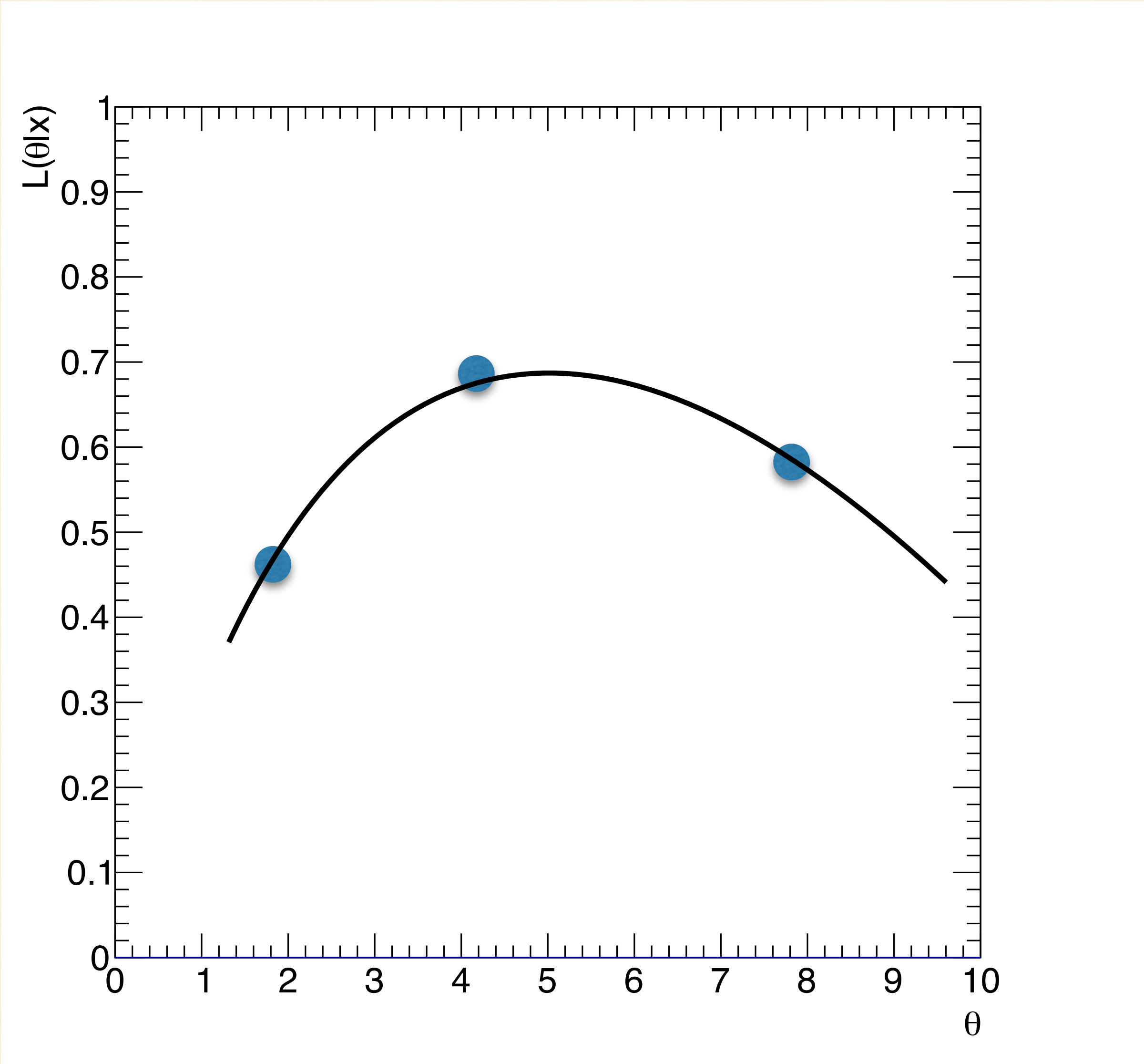


How to interpolate?

<u>Fit a function</u>

Good choice if there is a well-motivated functional form

The choice of function is too important, and can bias the result if chosen poorly



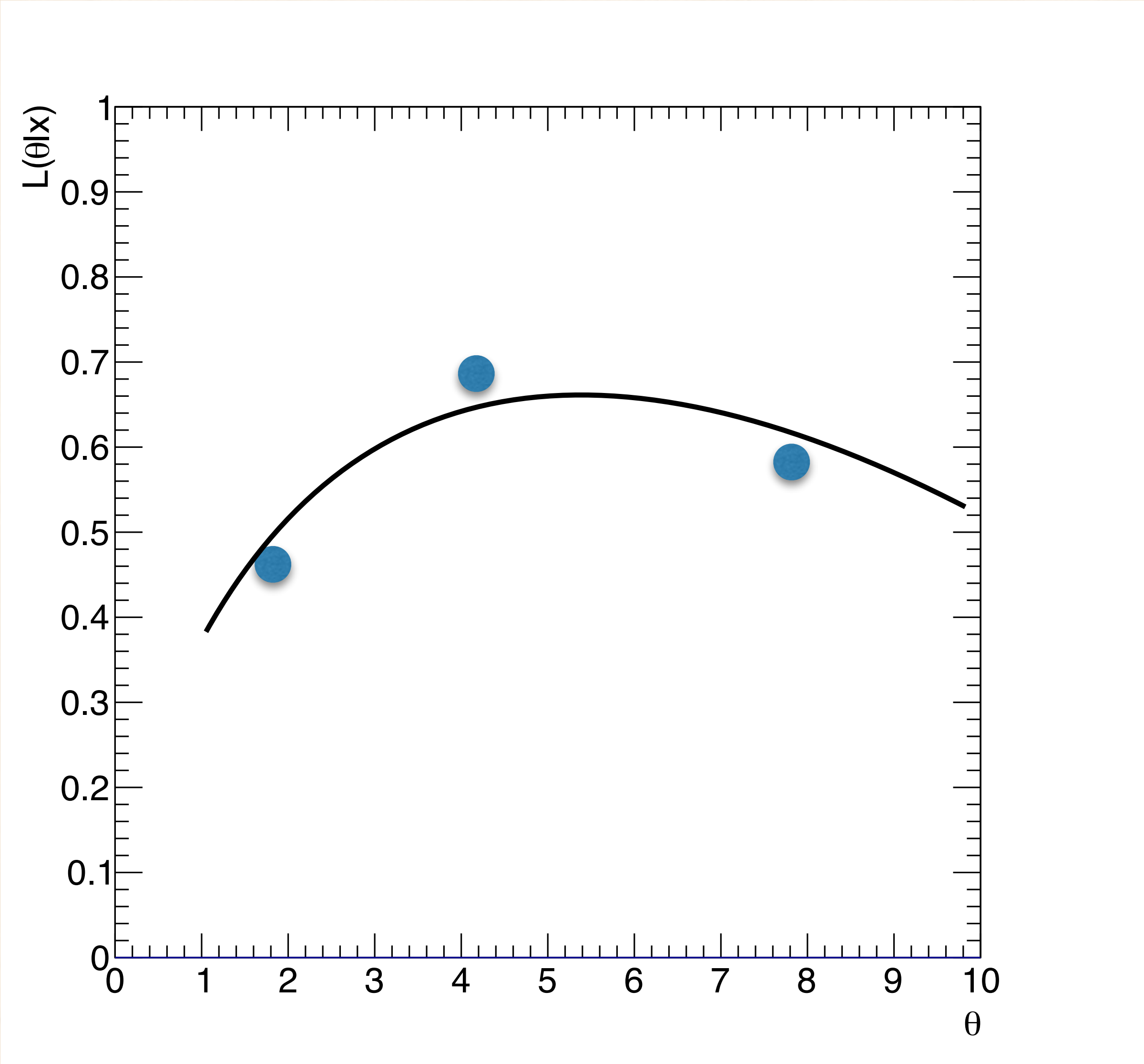
How to interpolate?

<u>Closest neighbor average</u>

Average of neighbors with weights depending on distance

Easily generalized to higher dimensions

Smooths the likelihood function — may not be ideal



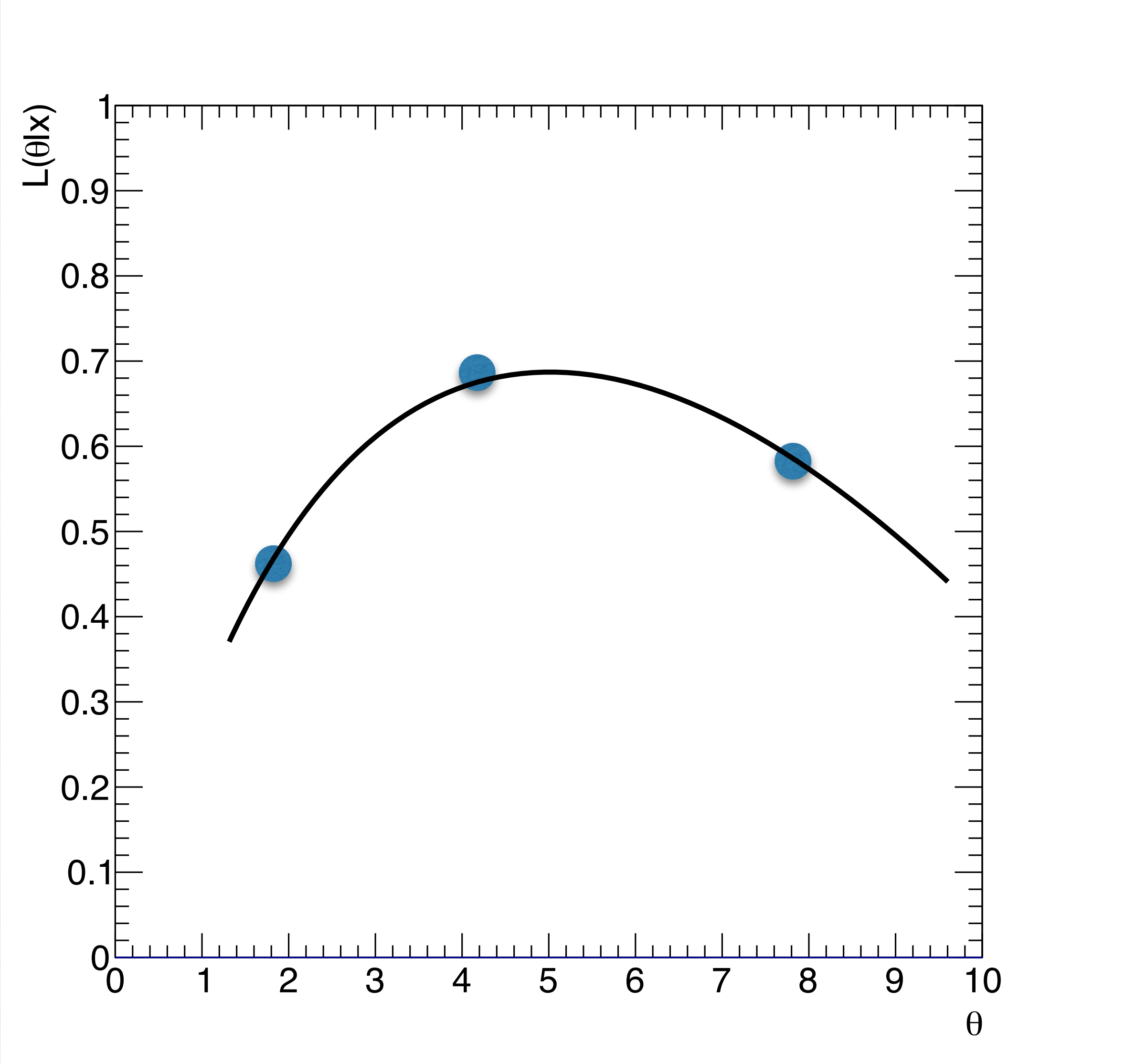
How to interpolate?

<u>"Gaussian process emulator" (GPE)</u>

Interpolates points without needing to assume a global functional form

Can easily be adapted to higher dimensions

Gives "interpolation error"

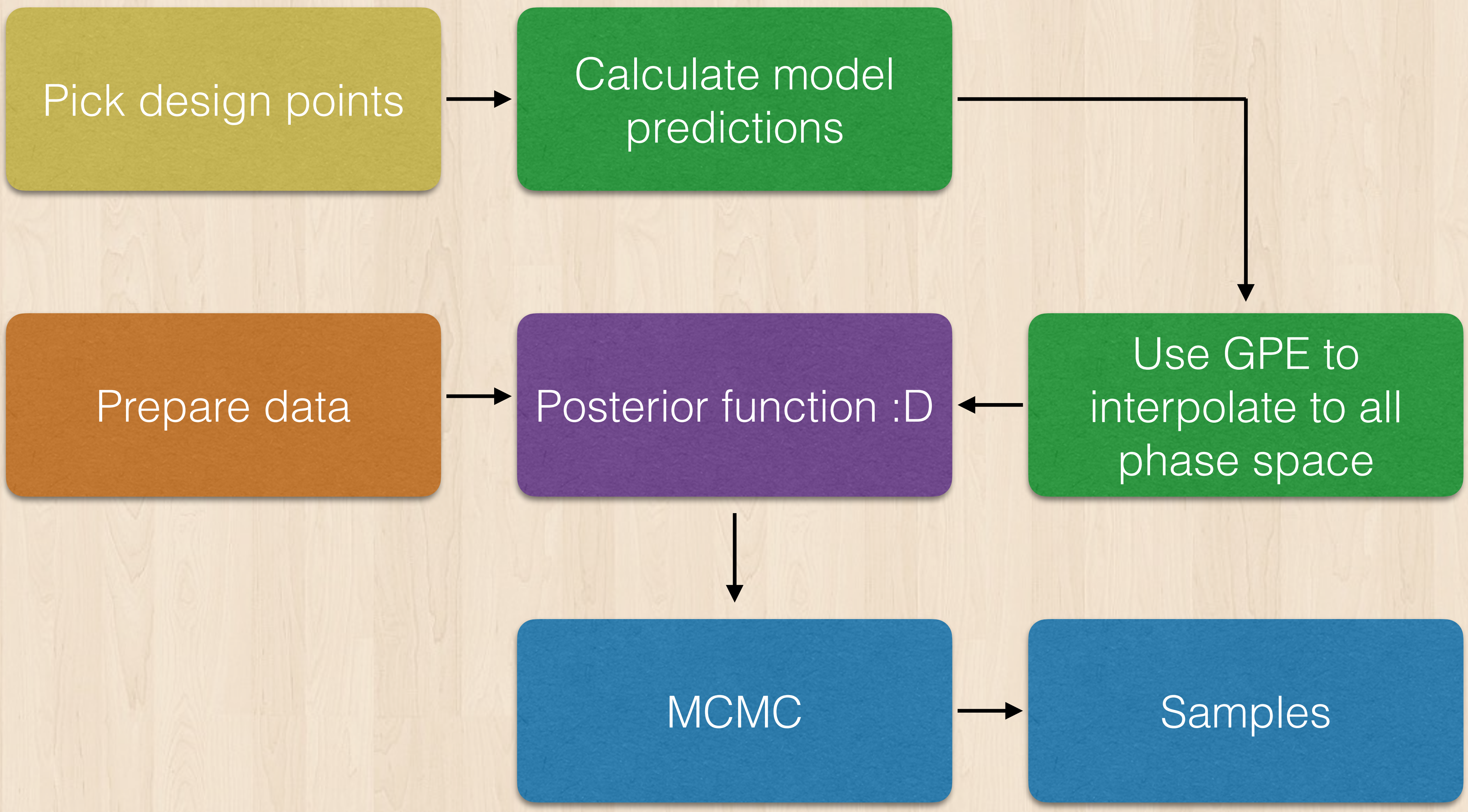


Build the likelihood: recap

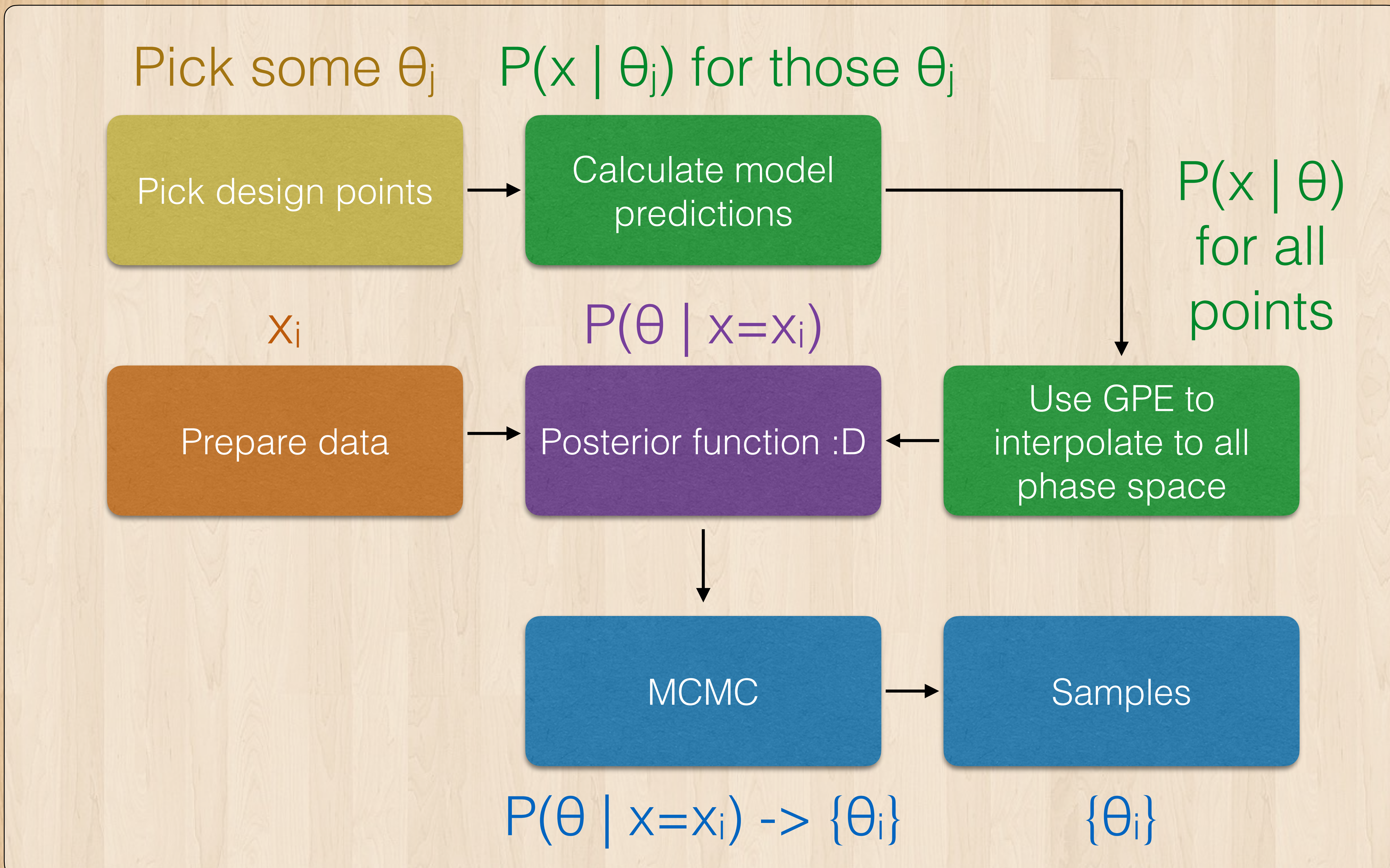
- There are many ways we can build the function
  - Analytical functions if we are extremely lucky
- When it becomes complicated, approximations have to be made
- For example in the case of computing-intensive calculations, we can pick points and interpolate
  - Gaussian process emulator (GPE) is one of the good ways to do this

Putting it all together

STAT analysis flow



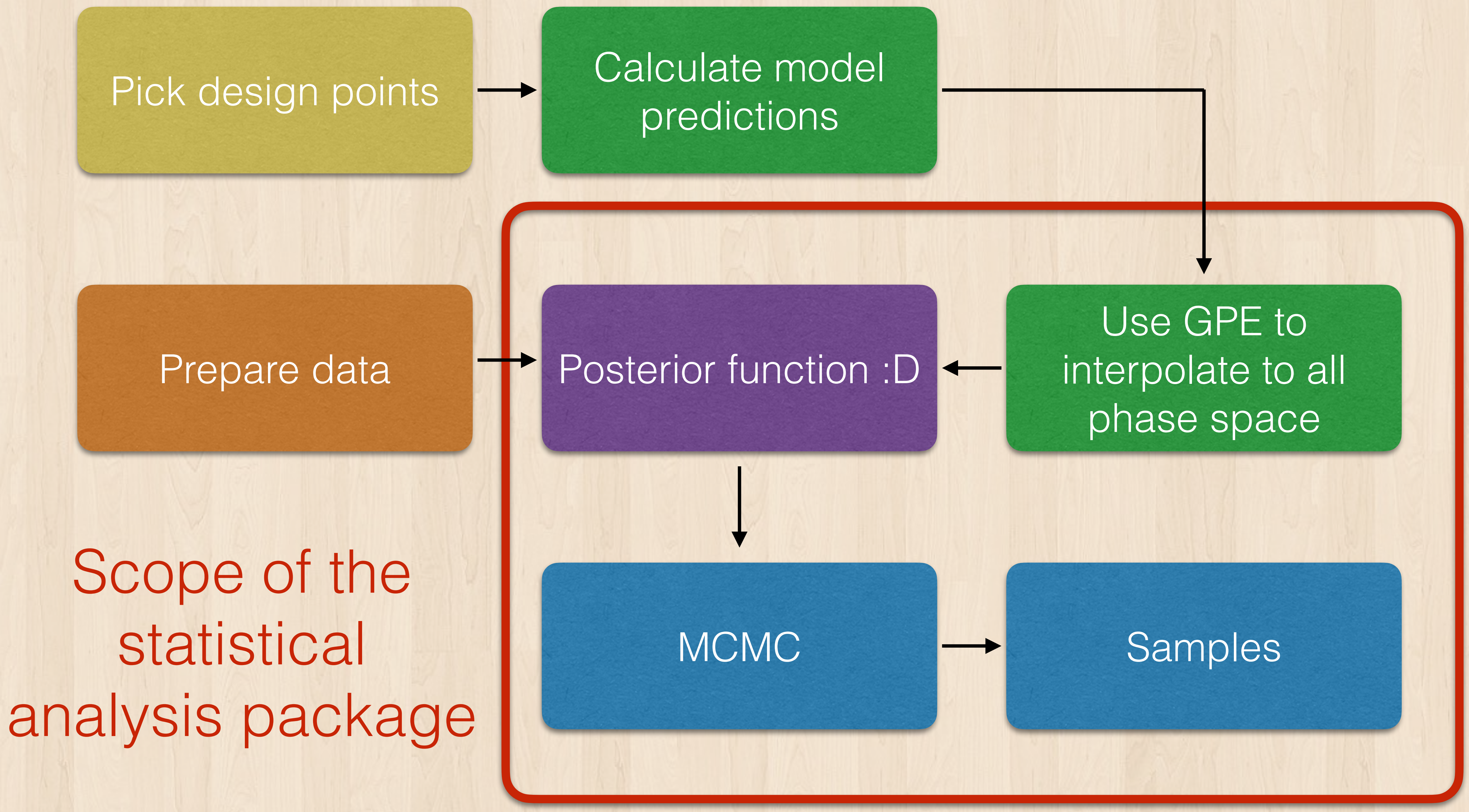
STAT analysis flow



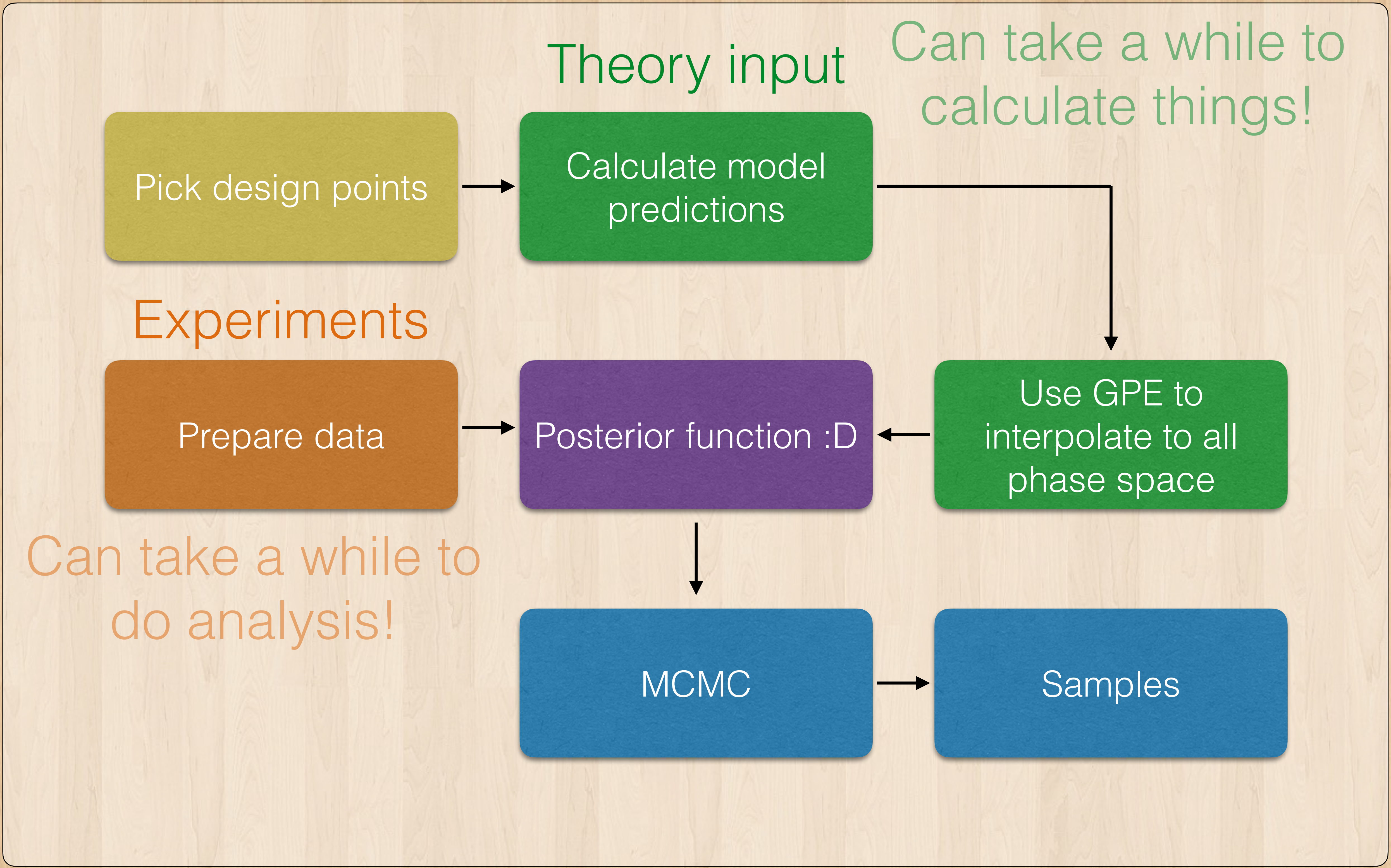
The STAT package

- Statistical analysis package for JETSCAPE
  - https://github.com/JETSCAPE/STAT
  - Evolved through many collaborators
- Python-based
- Simple to use
- Standardized input format — easy to share input files with colleagues

More practically...

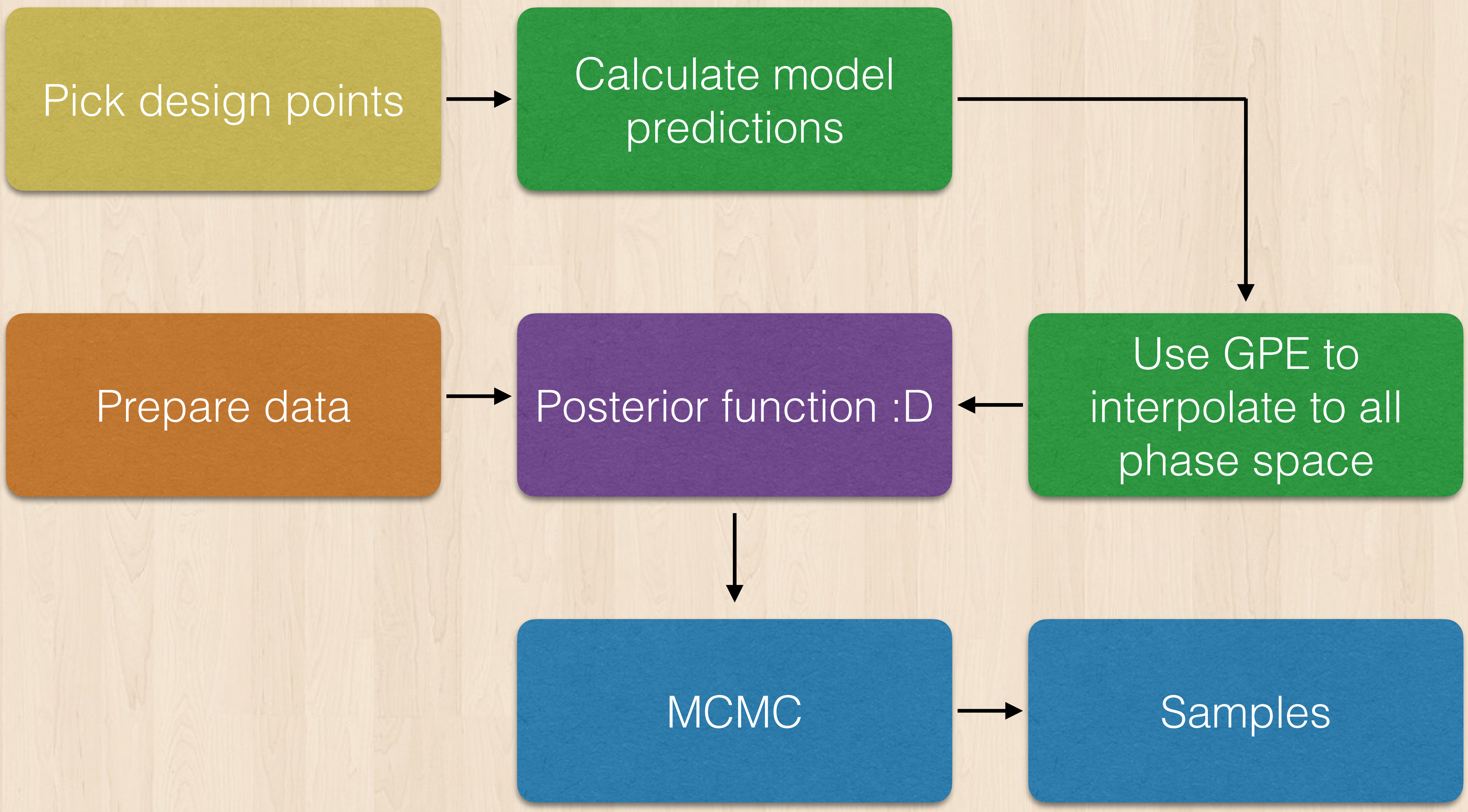


More practically…

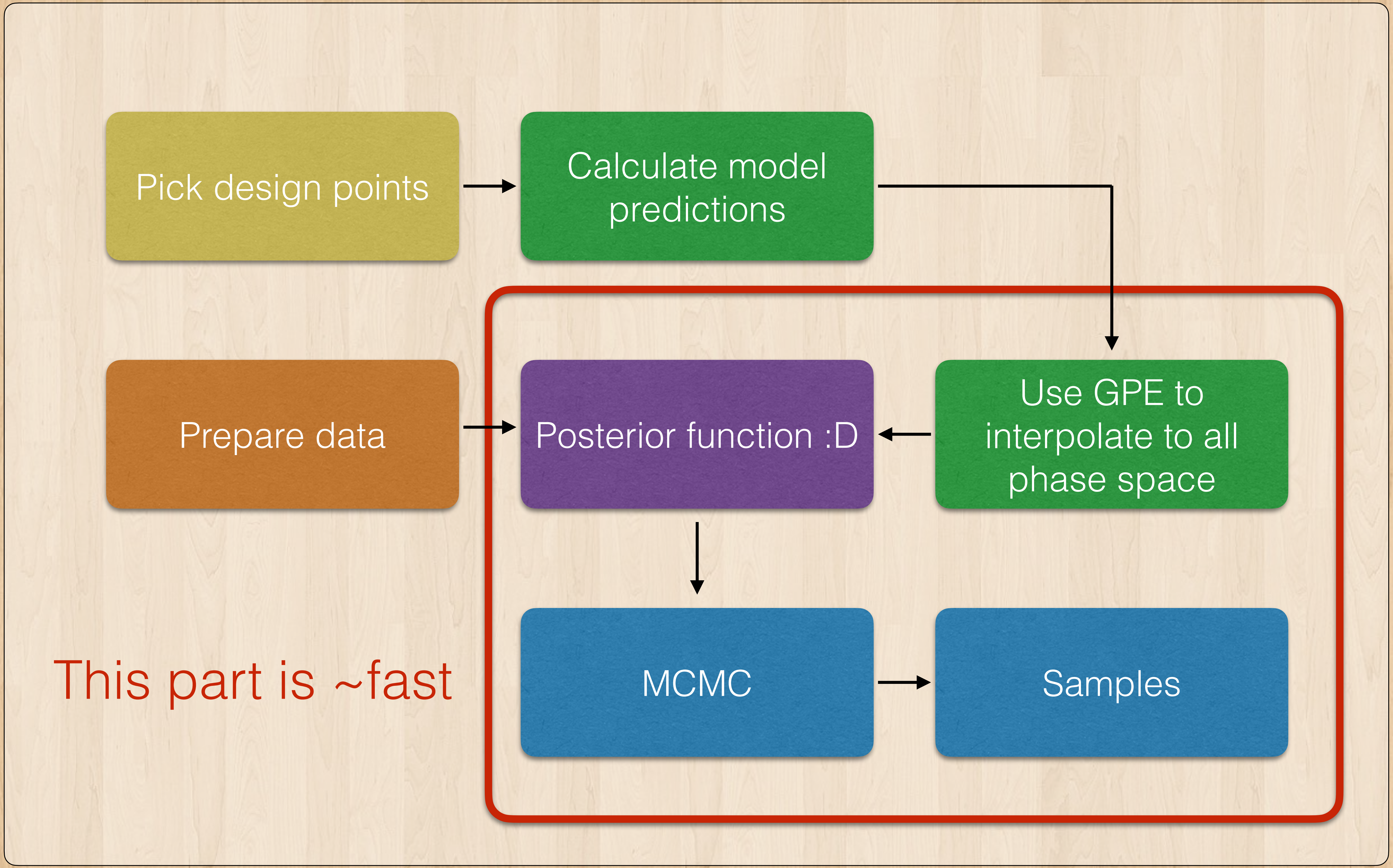


More practically...

May need to revisit if insufficient



More practically...



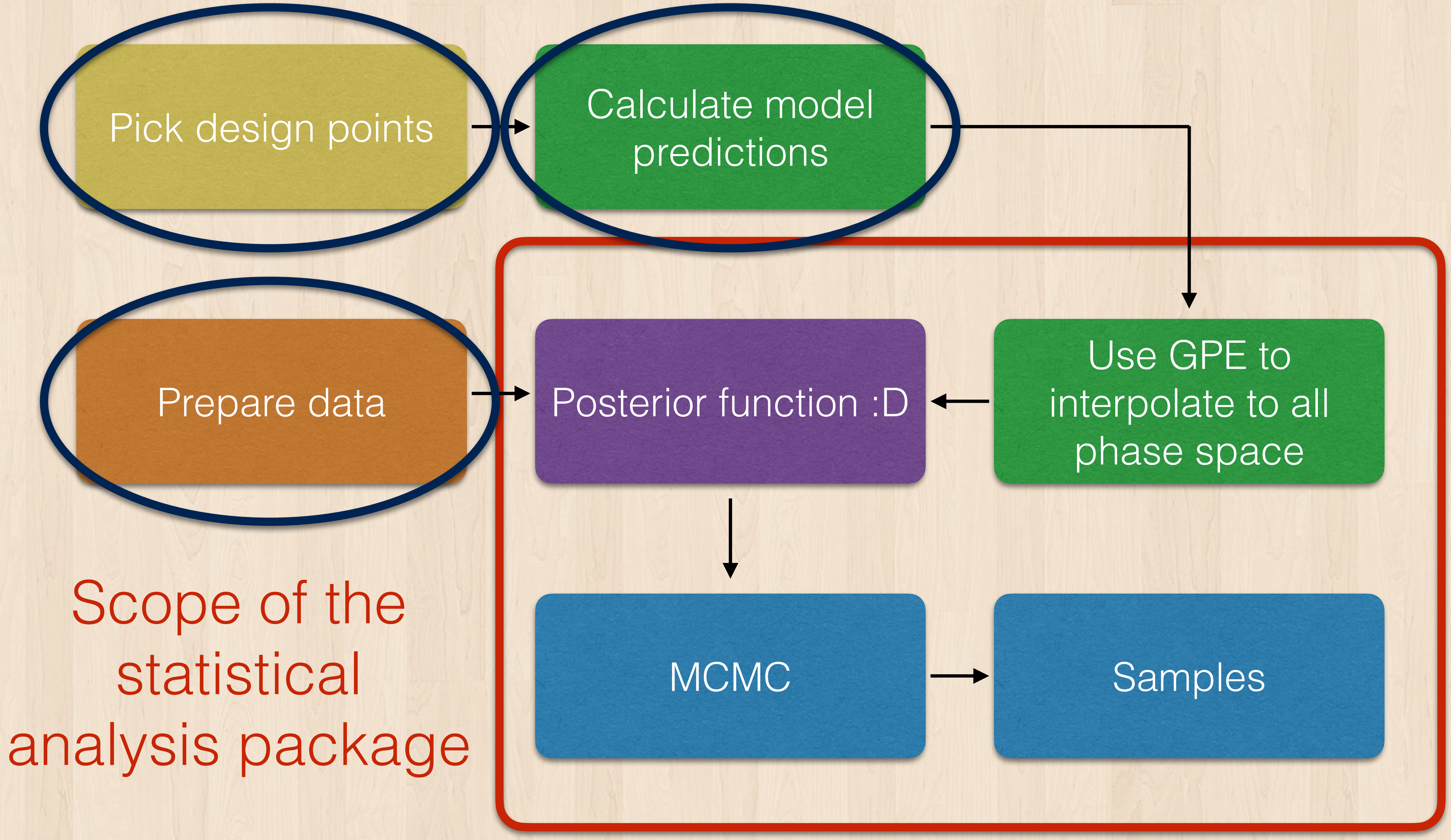
Hands-on session

Hands-on overview

- Today we will try to "learn" the parameters of a simple function
  - General procedure is very similar to the past two days
  - The primary goal is to understand the STAT package
- First we will go over the formats of the input files — you can find the files in the `input/SimpleExample` folder
- Then we will go through the `JetScapeSummerSchoolHandsOnSession` Jupyter notebook together

Input to the analysis

There are three types of input to be prepared



Input files — experimental data

Basic information

```
# Version 1.0
# DOI None
# Source None
# Experiment JetScapeRun1
# System PbPb5020
# Centrality 0to10
# XY X Y
# Label xmin xmax y stat,low stat,high sys,low sys,high
4.000e+01 5.000e+01 7.782090226582282222e-01 2.00000e-02 2.00000e-02 2.00000e-02 2.00000e-02
5.000e+01 6.000e+01 9.342818035113488184e-01 2.00000e-02 2.00000e-02 2.00000e-02 2.00000e-02
6.000e+01 7.000e+01 1.099344779825634166e+00 2.00000e-02 2.00000e-02 2.00000e-02 2.00000e-02
7.000e+01 8.000e+01 1.280022425780759310e+00 2.00000e-02 2.00000e-02 2.00000e-02 2.00000e-02
8.000e+01 9.000e+01 1.487233526614457402e+00 2.00000e-02 2.00000e-02 2.00000e-02 2.00000e-02
9.000e+01 1.000e+02 1.679563913274253029e+00 2.00000e-02 2.00000e-02 2.00000e-02 2.00000e-02
```

What the columns are

Each row is a data point

Input files — design points

The name of the parameters

```
# Version 1.0
# Parameter A B C
9.782411327236648635e-01 2.076838198651422829e-01 8.476964382665840292e-01
6.155842916764638906e-01 6.993495654476791223e-01 1.832904306159582886e-01
2.755050531222112964e-01 1.338337544465995066e-03 6.992493861834578883e-01
2.429636313141654291e-01 2.924101332840836065e-01 8.026584930879354651e-02
3.325660307041455876e-02 9.885420691136761473e-01 8.814541527616210903e-01
5.304160839555781548e-01 2.226059269751509140e-01 8.870357852154864275e-01
9.768419303075818183e-01 7.363279517998432278e-01 9.798007596704348954e-01
2.253429267015828463e-01 5.782081284312748926e-01 1.665769065869221466e-01
1.491366361275814345e-02 4.816583915667406179e-01 7.713447963424632237e-01
1.142979447757289657e-01 6.192934846748854305e-01 1.940153145482814701e-01
3.387106201193146315e-02 6.546859997917365837e-01 2.142190487123640796e-01
2.072812023378357571e-01 6.754897160180575177e-01 7.593847668854326605e-01
```

Each row is a design point

Input file — model prediction

What the prediction corresponds to: data, design

```
# Version 1.0
# Data Data_Selection1.dat
# Design Design.dat
1.243357380411961977e+00 9.674079083276510005e-01 4.177053057193712005e-01 3.908020257770337680e-01 6.565950001057970775
e-01 8.102134976005320732e-01 1.506599162450774188e+00 5.192684080795082480e-01 3.878572610771402474e-01 4.3226811407545
44016e-01 3.718591192824666769e-01 6.650269898402616509e-01 5.403501494739986200e-01 6.586968814432990760e-01 6.49892670
2474743244e-01 6.491529696474237499e-01 4.976822138859062217e-01 6.215422879006272661e-01 4.855757677346337897e-01 8.113
218068449465914e-01 6.837379415851289055e-01 6.443147103659520036e-01 1.223541684612486691e+00 6.573379283186966404e-01
1.045691015825165326e+00 1.180200125898649643e+00 6.732457639020236195e-01 3.853311948435006462e-01 1.285087684774989469
e+00 1.166868698351097855e+00 4.847523309162217187e-01 3.014234934841315550e-01 5.904982059703407504e-01 4.876474162087
59971e-01 4.269267081564170341e-01 1.349667994145692163e+00 1.209953749562795222e+00 7.012029154111967255e-01 8.50105649
7948069612e-01 1.257777553006125704e+00 6.194736969599161647e-01 9.358735819423303903e-01 1.131254941849516138e+00 8.966
324895860531274e-01 2.073836372279621587e-01 6.912776348515086156e-01 1.408307337848372809e+00 7.558610701970035484e-01
9.302139554485455708e-01 9.994242417380125865e-01 2.533677013664347166e-01 1.457273667231519854e+00 1.102735229192783661
e+00 1.314069699916649014e+00 1.148056552324210777e+00 2.370463546662085197e-01 7.066069517685329426e-01 1.170234940785
19404e+00 1.294440928905818300e+00 4.673126237364766400e-01 1.480355285411345811e-01 5.171661227564560148e-01 7.09778454
6392805760e-01 7.594988891016289934e-01 7.543110797481908936e-01 3.865789053549301135e-01 6.489350997061693604e-01 9.004
235324140148489e-01 7.799537807305327863e-01 1.439714656169180707e+00 3.354893593410366859e-01 8.275629594222200236e-01
1.073967355450783590e+00 1.159474583030086992e+00 1.057318996505542952e+00 7.083111414130465189e-01 5.389887113511117045
e-01 9.285769830518001422e-01 9.956371586610001101e-01 7.600191572879745339e-01 1.241932624816065367e+00 2.500420211447
85181e-01 5.696570599752873720e-01 7.720859096766425900e-01 3.347397167380940508e-01 1.098352350358581697e+00 9.9335678
8653876441e-01 6.325193108648247131e-01 6.875601703363792838e-01 1.177948099314045871e+00 1.086378427837187965e+00 7.542
245565248617556e-01 9.150656874123375140e-01 9.216295834203388493e-01 6.071512579621712868e-01 4.673438733167923909e-01
8.291494775945014162e-01 9.067873509335313553e-01 1.255307629695931571e+00 7.542059489318131416e-01
1.348895406225134819e+00 1.055671907934014886e+00 4.877640780921636554e-01 4.280696240363214833e-01 8.435946222933268235
e-01 9.211776688195958407e-01 1.678212033597802133e+00 5.937469115813279741e-01 5.131575798680606537e-01 5.135989939977
12027e-01 4.587496241328767876e-01 8.085144381306108574e-01 5.982190835744993773e-01 7.677755380935767926e-01 6.92500841
```

Each row = prediction for one data point from all design points

Input file format: recap

- For more information, you can find the specifications here
- All the input files needed for the exercise today is prepared in `input/SimpleExample`, so we do not have to worry about the details on that (for now)
- Let’s now move on to the hands-on analysis part

Analysis

Starting Jupyter notebook

- Go to the docker base directory
- Update the `STAT` directory
  - If you haven't checked it out yet, do `git clone https://github.com/JETSCAPE/STAT.git` in the base directory
  - If you have checked it out some time ago, do a `git pull` inside the `STAT` folder to make sure things are up to date
- In the `STAT` folder, switch to the summer school branch: `git checkout JetScapeSummerSchool2020`

Starting Jupyter notebook

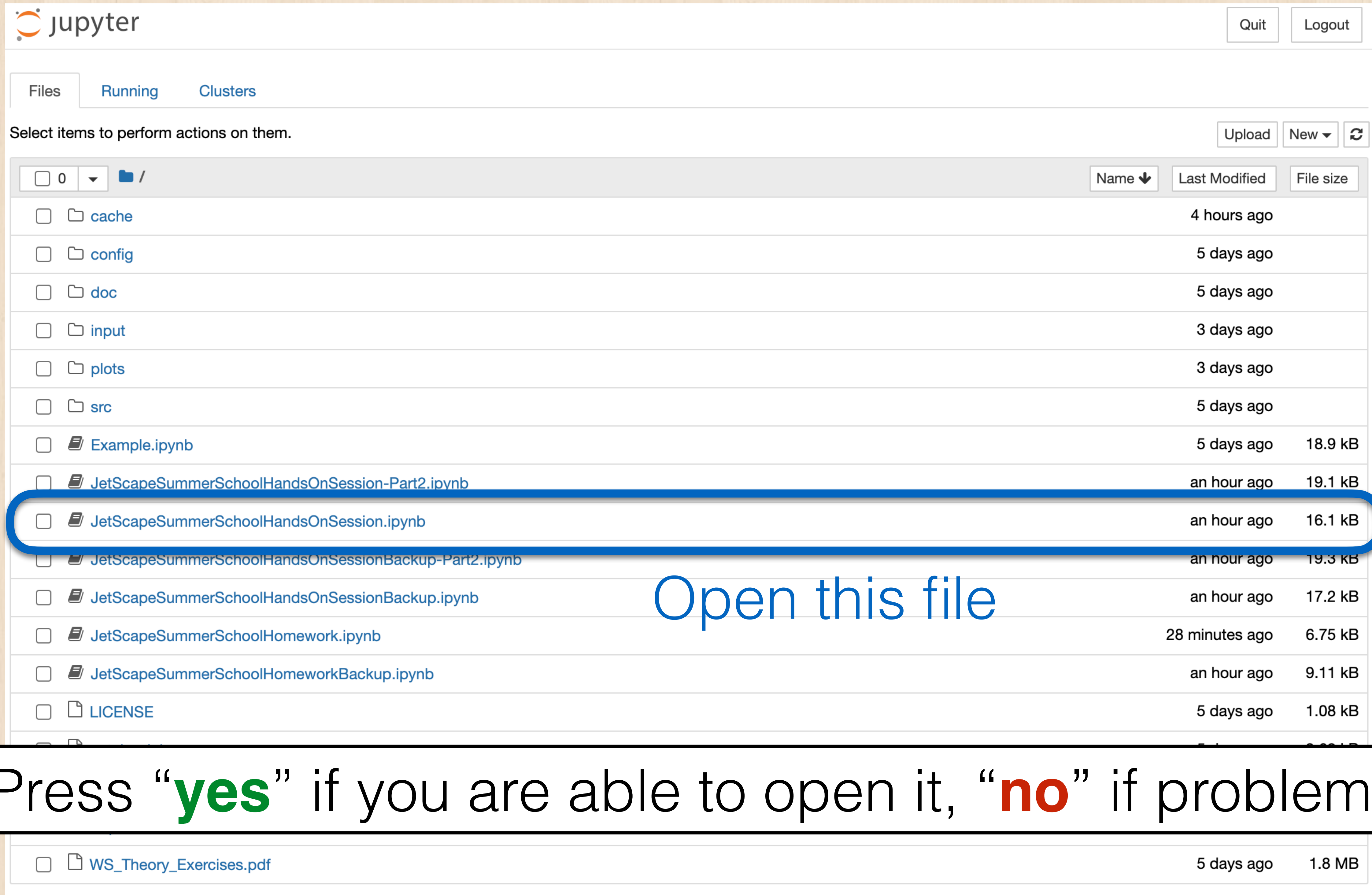
- In the base directory, start docker as `docker run --rm -it -p 8888:8888 -v `pwd`:/home/jetscape-user --name stat jetscape/base:v1.4` (add `--user $(id -u):$(id -g)` if on linux)
- In the container, enter the `STAT` directory, and start jupyter notebook as `jupyter-notebook --ip 0.0.0.0 --no-browser`

```
[I 09:57:59.319 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 09:57:59.332 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///home/jetscape-user/.local/share/jupyter/runtime/nbserver-9-open.html
    Or copy and paste one of these URLs:
     or http://127.0.0.1:8888/?token=fd91238546e5e215dba6db292c1b8ca064d4d23c3155b613
```

Copy-paste this URL into your browser

Jupyter notebook



Open this file

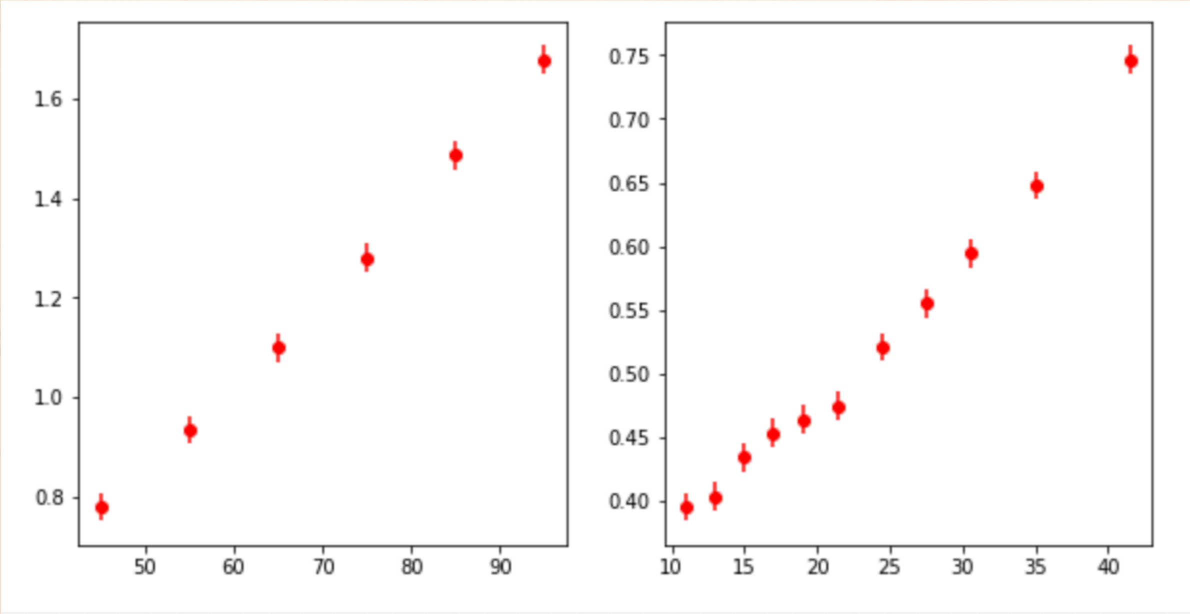
Press “**yes**” if you are able to open it, “**no**” if problems

Analysis setup

- We have the "truth" function as

$$y = A + B\frac{x}{100} + C\left(\frac{x}{100}\right)^2$$

- We have two measurements, one in high-x region, one in low-x region



- Let's try to learn A, B and C using the STAT package!

Homework

Homework

- We will attempt to learn something about jet energy loss from dijet imbalance data
- The data file is provided
- The homework objective is
  - Generate “model prediction” and design points
    - Follow the `JetScapeSummerSchoolHomework.ipynb`
  - Plug them into the analysis and see if you can learn something about the parameters
    - Use (or make a copy of) the part2 notebook for this purpose

Dealing with data

Uncertainty!?

- The central question to interpret data is “what do you mean by uncertainty!?”
- Recall that the **uncertainties are “descriptions” to the full likelihood function**, and one can only guess what the likelihood functions look like with some Ansätze
- One popular guess is a Gaussian function with mean and RMS which is equal to what is reported by experiments

Complications, correlations

- Even with the Gaussian approximation, there are a few complications to take into account
  - Correlation between different bins in the measurement?
  - Correlation between different systematic uncertainties?
- For experimental data to be most efficiently used, the ideal case is that experiments provide these correlations
- The more approximations one has to make, the worse the result will be

Example: charged hadron RAA

From HEPData

RE	PB PB --> CHARGED X
CENTRALITY	10-30%
TRACK ETA	-1.0 TO 1.0
SQRT(S)/NUCLEON	5020.0 GEV
PT [GEV]	**RAA**
0.7 - 0.8	0.345625 ±1.7e-05 stat ±0.027069 sys $^{+0.009331}_{-0.011751}$ sys,TAA ±0.007949 sys,lumi
0.8 - 0.9	0.369594 ±1.9e-05 stat ±0.02923 sys $^{+0.009979}_{-0.012566}$ sys,TAA ±0.0085 sys,lumi
0.9 - 1.0	0.404274 ±2.2e-05 stat ±0.032123 sys $^{+0.010915}_{-0.013745}$ sys,TAA ±0.009298 sys,lumi
1.0 - 1.1	0.425955 ±2.5e-05 stat ±0.036377 sys $^{+0.0115}_{-0.014482}$ sys,TAA ±0.009796 sys,lumi
1.1 - 1.2	0.445204 ±2.8e-05 stat ±0.037816 sys $^{+0.01202}_{-0.015136}$ sys,TAA ±0.010239 sys,lumi
1.2 - 1.4	0.466304 ±2.4e-05 stat ±0.045156 sys $^{+0.01259}_{-0.015854}$ sys,TAA ±0.010725 sys,lumi
1.4 - 1.6	0.487476 ±3.2e-05 stat ±0.050542 sys +0.013161 sys,TAA ±0.011212 sys,lumi

Example: charged hadron RAA

RE	PB PB --> CHARGED X
CENTRALITY	10-30%
TRACK ETA	-1.0 TO 1.0
SQRT(S)/NUCLEON	5020.0 GEV
PT [GEV]	RAA
0.7 - 0.8	0.345625 ±1.7e-05 stat ±0.027069 sys +0.009331 -0.011751 sys,TAA ±0.007949 sys,lumi
0.8 - 0.9	0.369594 ±1.9e-05 stat ±0.02923 sys +0.009979 -0.012566 sys,TAA ±0.0085 sys,lumi
0.9 - 1.0	0.404274 ±2.2e-05 stat ±0.032123 sys +0.010915 -0.013745 sys,TAA ±0.009298 sys,lumi
1.0 - 1.1	0.425955 ±2.5e-05 stat ±0.036377 sys +0.0115 -0.014482 sys,TAA ±0.009796 sys,lumi
1.1 - 1.2	0.445204 ±2.8e-05 stat ±0.037816 sys +0.01202 -0.015136 sys,TAA ±0.010239 sys,lumi
1.2 - 1.4	0.466304 ±2.4e-05 stat ±0.045156 sys +0.01259 -0.015854 sys,TAA ±0.010725 sys,lumi
1.4 - 1.6	0.487476 ±3.2e-05 stat ±0.050542 sys +0.013161 sys,TAA ±0.011212 sys,lumi

Stat: uncorrelated

Example: charged hadron RAA

RE	PB PB --> CHARGED X				
CENTRALITY	10-30%				
TRACK ETA	-1.0 TO 1.0				
SQRT(S)/NUCLEON	5020.0 GEV				
PT [GEV]	RAA				
0.7 - 0.8	0.345625	±1.7e-05 stat	±0.027069 sys	+0.009331 -0.011751 sys,TAA	±0.007949 sys,lumi
0.8 - 0.9	0.369594	±1.9e-05 stat	±0.02923 sys	+0.009979 -0.012566 sys,TAA	±0.0085 sys,lumi
0.9 - 1.0	0.404274	±2.2e-05 stat	±0.032123 sys	+0.010915 -0.013745 sys,TAA	±0.009298 sys,lumi
1.0 - 1.1	0.425955	±2.5e-05 stat	±0.036377 sys	+0.0115 -0.014482 sys,TAA	±0.009796 sys,lumi
1.1 - 1.2	0.445204	±2.8e-05 stat	±0.037816 sys	+0.01202 -0.015136 sys,TAA	±0.010239 sys,lumi
1.2 - 1.4	0.466304	±2.4e-05 stat	±0.045156 sys	+0.01259 -0.015854 sys,TAA	±0.010725 sys,lumi
1.4 - 1.6	0.487476	±3.2e-05 stat	±0.050542 sys	+0.013161 sys,TAA	±0.011212 sys,lumi

TAA: correlated across bins and experiments

Lumi: correlated across centrality and bin

Example: charged hadron RAA

RE	PB PB --> CHARGED X				
CENTRALITY	10-30%				
TRACK ETA	-1.0 TO 1.0				
SQRT(S)/NUCLEON	5020.0 GEV				
PT [GEV]	RAA				
0.7 - 0.8	0.345625	±1.7e-05 stat	±0.027069 sys	+0.009331 -0.011751 sys,TAA	±0.007949 sys,lumi
0.8 - 0.9	0.369594	±1.9e-05 stat	±0.02923 sys	+0.009979 -0.012566 sys,TAA	±0.0085 sys,lumi
0.9 - 1.0	0.404274	±2.2e-05 stat	±0.032123 sys	+0.010915 -0.013745 sys,TAA	±0.009298 sys,lumi
1.0 - 1.1	0.425955	±2.5e-05 stat	±0.036377 sys	+0.0115 -0.014482 sys,TAA	±0.009796 sys,lumi
1.1 - 1.2	0.445204	±2.8e-05 stat	±0.037816 sys	+0.01202 -0.015136 sys,TAA	±0.010239 sys,lumi
1.2 - 1.4	0.466304	±2.4e-05 stat	±0.045156 sys	+0.01259 -0.015854 sys,TAA	±0.010725 sys,lumi
1.4 - 1.6	0.487476	±3.2e-05 stat	±0.050542 sys	+0.013161 sys,TAA	±0.011212 sys,lumi

"other" systematics: we don't know the correlation :(

The corresponding header

```
# Version 1.0
# DOI http://dx.doi.org/10.1007/JHEP04(2017)039
# Source https://www.hepdata.net/download/table/ins1496050/Table14/yaml
# Experiment CMS
# System PbPb5020
# Centrality 10to30
# XY PT RAA
# Label xmin xmax y stat,low stat,high sys,low sys,high sys,TAA,low sys,TAA,high sys,lumi,low sys,lumi,high
```

All the different systematic sources

Experimentally…

- Knowing the exact likelihood is very tricky, even if you are the main analyzer for an experiment analysis
- Often we see things like “we vary X by y% and quote the difference as systematics”
  - What is actually meant is that, there is some underlying likelihood function — and this difference tells us something about that function
- It is good exercise to think about these when doing analysis
- The better we can pin this down, the more useful the data will be for the community

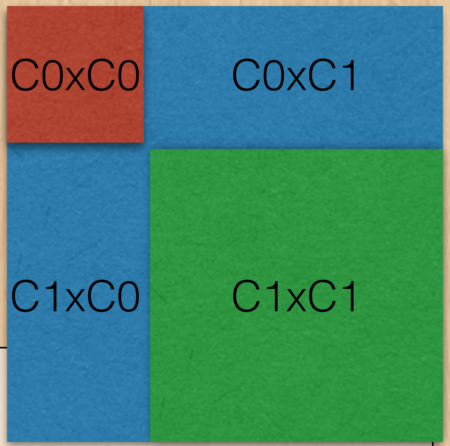
Ideally…

```
Covariance["PbPb5020"][("R_AA", "C0")][("R_AA", "C0")] = RawCov1["Matrix"]
Covariance["PbPb5020"][("R_AA", "C1")][("R_AA", "C1")] = RawCov2["Matrix"]
```

If an experiment provides covariance matrix directly, we can put it in directly!

This is the best-case scenario, but unfortunately we very rarely have them…

In practice…



Covariance between “C0” and “C1” measurements

```
Covariance["PbPb5020"][("R_AA", "C0")][("R_AA", "C1")] = \
   Reader.EstimateCovariance(RawData1, RawData2, SysLength = {"default": 0.2, "sys,lumi": 999})
Covariance["PbPb5020"][("R_AA", "C1")][("R_AA", "C0")] = \
   Reader.EstimateCovariance(RawData2, RawData1, SysLength = {"default": 0.2, "sys,lumi": 999})
```

If not explicitly listed, assume correlation length of 0.2 between bins (see next page)

Assume “lumi” systematics are fully correlated between the two measurements

The labels correspond to the “column name” in data

In practice...

This is source by source

1.9 for numerical stability

$$C_{ij} = \text{strength} \times \sigma_i \sigma_j \exp\left[-\left(\frac{x_i - x_j}{\text{length}}\right)^{1.9}\right]$$

Correlation strength:
"SysStrength"
defaults to 1

Correlation length:
"SysLength"

Check src/reader.py for complete information on this covariance matrix estimation

Summary

Summary

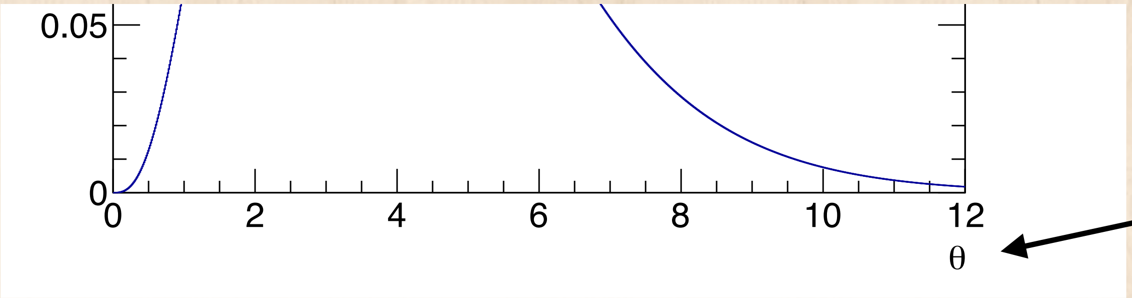
- Likelihood function is the key
    - **The numbers we quote in experimental physics are "descriptions" of the likelihood**
- We can analyze the posterior function by sampling (MCMC)
- We can build the function by "interpolation" (GPE), in case it is computing-intensive
- Systematic uncertainties are tricky: usually we don't have access to the full likelihood functions —> approximations

Backup slides ahead

Why sqrt(N) is not correct

$P(x\,|\,\theta)$ Variance on x = θ

$\mathcal{L}(\theta\,|\,x)$ Variance on θ ≠ x



θ

What we are quoting is a range on θ, not on x

$$P(x \mid \theta)$$

$$\mathscr{L}(\theta \mid x)$$

$$P(\theta \mid x)$$

STAT analysis flow