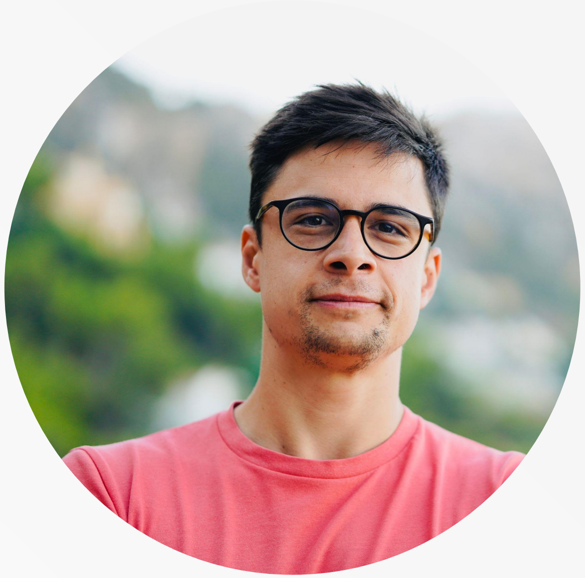
Bayesian Modeling Introduction

Alexandre Andorra
@alex_andorra

PyMC Labs

Alexandre Andorra

- Principal DS & Co-founder of PyMC Labs
- PyMC & ArviZ author
- Host & Creator of the "Learning Bayesian Statistics" podcast
- Teacher in the Intuitive Bayes galaxy



And of course, a football prodigy, since 1990

Thomas Wiecki

- PhD on computational psychiatry, Brown University
- Bayesian modeler & PyMC co-creator
- CEO & Founder of PyMC Labs
- Twitter: @twiecki

PyMC Labs – The Bayesian Consultancy

Inventors of PyMC, the leading platform for statistical data science

Decades of experience building Bayesian models

World-class Industry Expertise

Team of

PhDs | **Mathematicians** | **Neuroscientists**

Computer Scientists | **Industry Experts**

Alex Andorra

Ben Vincent

Bill Engels

Thomas Wiecki

Luciano Paz

Maxim Kochurov

Ricardo Vieira

Tomi Capretto

Example: Estimating Treatment Effect

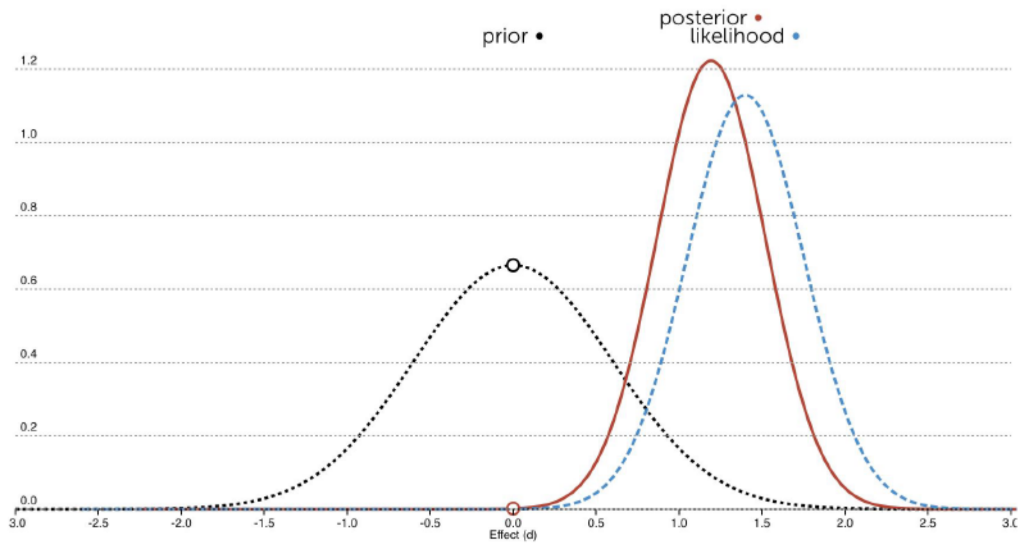
- Assume we have a new crop type and run a field trial
- Observe: Yield
- Question: How effective is our new crop?

How does it work in theory?

Updating beliefs

1. Starting belief: **Prior**
2. Observe data: **Likelihood**
3. Update belief: **Posterior**

$$\overset{\text{Posterior}}{p(\theta \mid \text{data})} = \frac{\overset{\text{Likelihood}}{p(\text{data} \mid \theta)} \cdot \overset{\text{Prior}}{p(\theta)}}{\underset{\text{Normalization}}{p(\text{data})}}$$



Demo

Go to: https://rpsychologist.com/d3/bayes/

How does it work in practice?

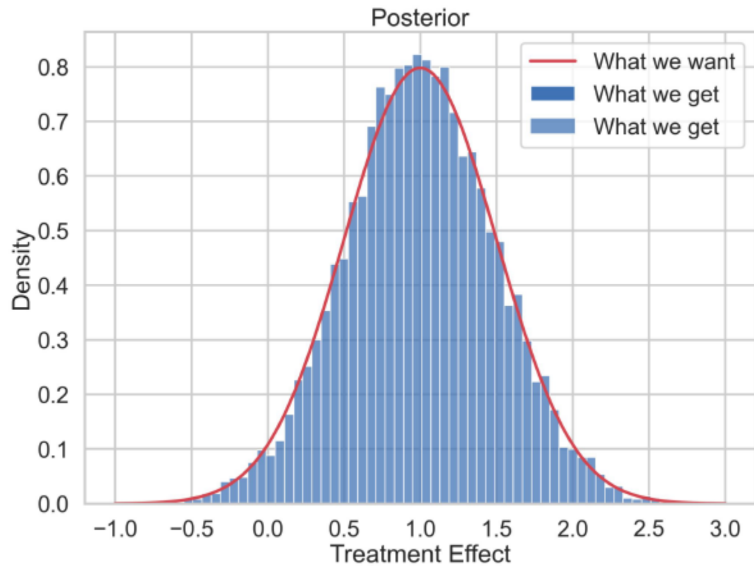
Normalization constant is (usually) impossible to compute analytically.

Instead: Use approximation method called **Markov Chain Monte Carlo (MCMC)** that *draws samples from the posterior.*

AKA... **The Magic Machine** 🪄

Offers great flexibility in model creation

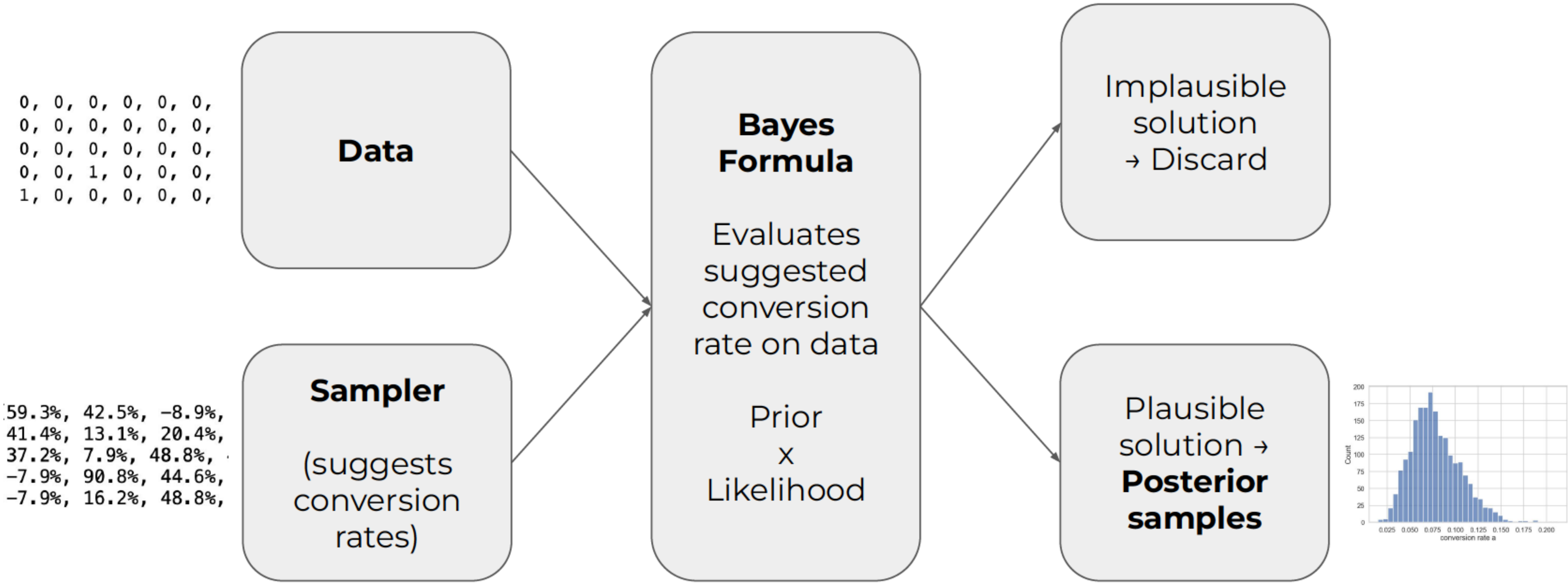
$$\overbrace{p(\theta \mid \text{data})}^{\text{Posterior}} = \frac{\overbrace{p(\text{data} \mid \theta)}^{\text{Likelihood}} \cdot \overbrace{p(\theta)}^{\text{Prior}}}{\underbrace{\cancel{p(\text{data})}}_{\text{Normalization}}}$$



How can we build this magic machine??

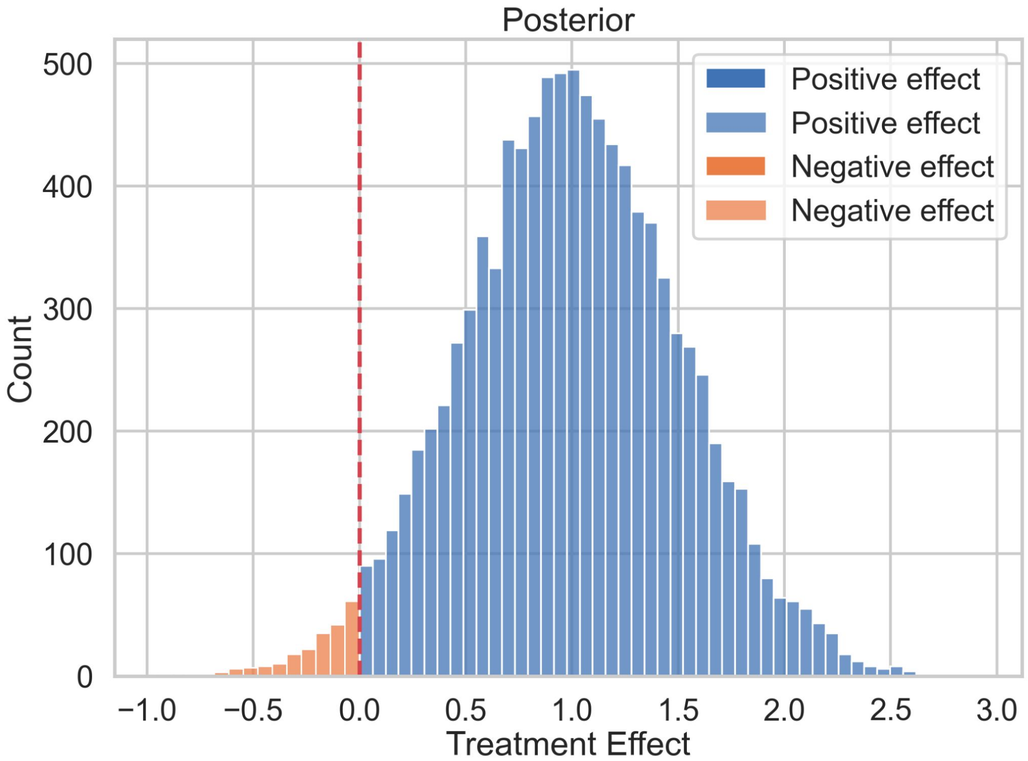
https://www.elmhurst.edu/blog/thomas-bayes/

Sampling to Get Our Posterior Belief



When statistics becomes counting

Probability of positive treatment effect?



→ Probability of positive treatment effect is 98%.

Que piola.

But it gets better, che.

Pre-packaged vs Composable

Frequentist statistics & Machine Learning

vs

Bayesian

Blackbox Machine Learning

- Only prediction - not inference
- Blackbox: Not good at conveying what was learned
- Down-side of automation: Purely data-driven, cannot incorporate pre-existing knowledge about problem

Frequentist Statistics

Data → Hypothesis Test → p-value

- Gray box - **implicit assumptions** (e.g. normality)
- Only use pre-packaged tests, or derive new estimator
- Assumes data has randomness, parameters are fixed
- p-values are **not** the probability we care about

Bayesian Modeling

- **Transparent**: Open box models
- **Composable**: Can tailor model to specific problem
- **Causal**: Incorporate domain knowledge
- **Probabilistic**: Assumes data is fixed, parameters have randomness
- **Code-friendly**: Build custom model in code, **not** math.

Priors & the Illusion of Objectivity

- A frequent concern about Bayesian modelling is that the scientist or analyst has to represent their **beliefs of the effect** they're studying into the prior

- However: it can be argued this is actually a positive effort to **explicit underlying assumptions**

- Many subjective decisions go into an experiment and its analysis:
    - Experimental design
    - Metrics
    - Model
    - Decision rules (what is a hit?)

PyMC: Code powerful models in Python

Modern: MCMC (NUTS) and variational inference (ADVI).

User friendly: Friendly Python syntax.

Fast: Compiles to C, supports GPU.

Batteries included: Distributions, Gaussian Processes

Community focused: Discourse, MeetUps, Twitter

```python
import pymc

with pymc.Model():
    effect = pm.Normal("effect", mu=0, sigma=1)
    noise = pm.HalfNormal("noise")
    obs = pm.Normal("obs", mu=effect, sigma=noise,
                    observed=[0.1, -.3, ...])

    posterior = pm.sample()
```

When does Bayesian modeling work best?

When does Bayesian modeling work best?

Not a simple prediction problem

Gain insight into your data

Structured data (hierarchical, time series...)

Integrate domain knowledge in models

Uncertainty plays an important role

Make decisions with real-world consequences

Sparse, noisy, unbalanced, or missing data

THE
POSSIBILITIES
ARE
ENDLESS

Our Clients

Build state-of-the-art MMM

- Beats FB Robyn and Uber's OKR in hold-out
- Used to optimize total international marketing budget

Automate MMM pipeline

- Before: Hundreds of analysts built custom MMM pipelines
- Now: Handful of analysts run automated pipelines that determine best MMM configuration

Improve AB testing

- Before: Used inflexible and limited frequentist framework
- Now: Uses modern Bayesian AB testing framework that scales to Twitch-size data
- Allows to make more accurate product-decisions faster

Develop state-of-the-art model for estimating differences in rat development

- Rats are monitored in cages over several months
- Model increases sensitivity and specificity over previous approach
- Upcoming Nature Methods submission

Estimating effects of different crop-types on agricultural fields

- Reliable estimation of treatment effects with spatial confounders
- Directly informs which crops are being marketed and sold

Cognitive training for kids with ADHD

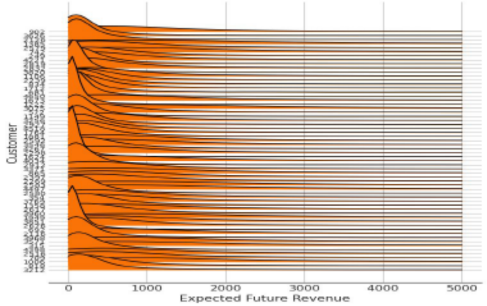
- Estimation of cognitive processes based on performance on cognitive training game
- Only model to reveal training effects with real-world carry-over

We work with	Data Science Teams	Marketing Analytics	Finance Teams	Growth Teams	Pharma	Biotech	Health Care

Examples from our client work

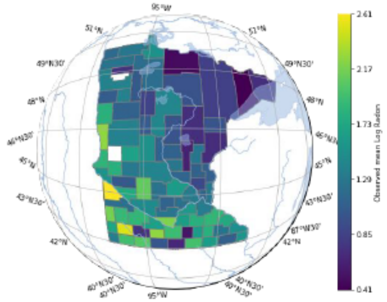
A

Expected Future Revenue per Customer?

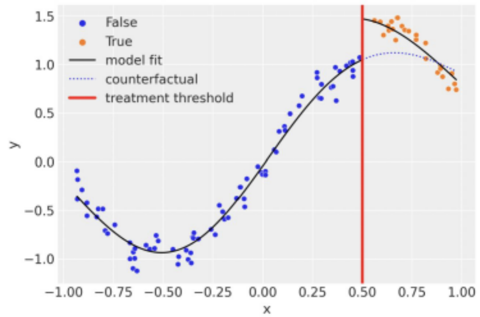


B

What is the spatial distribution of my data?

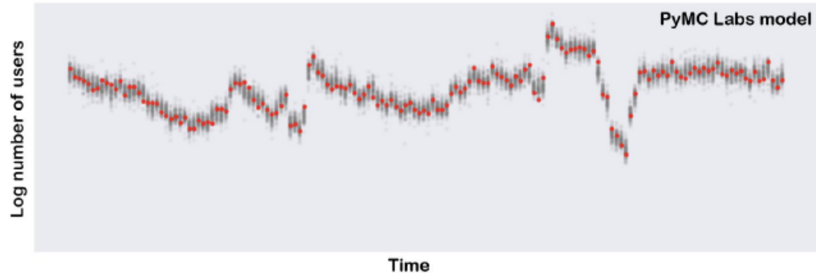


C



How big is the causal impact of my marketing campaign?

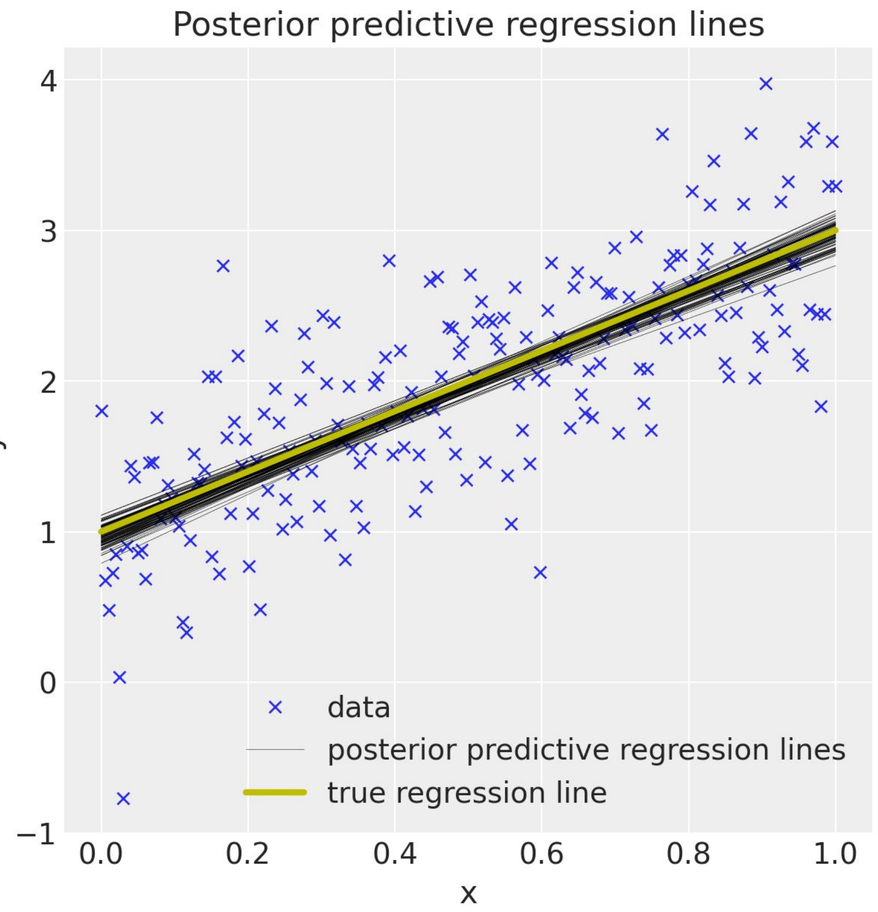
D



How many new users?

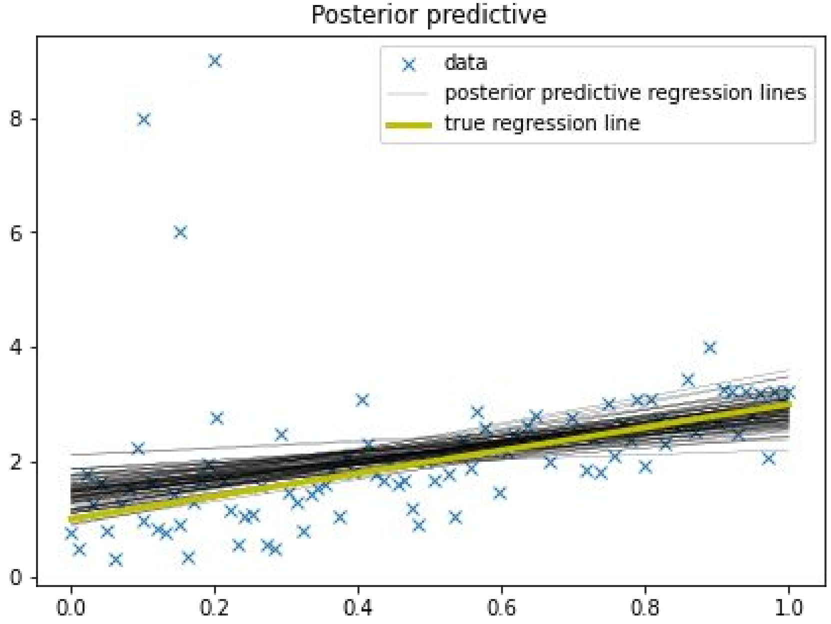
Linear regression

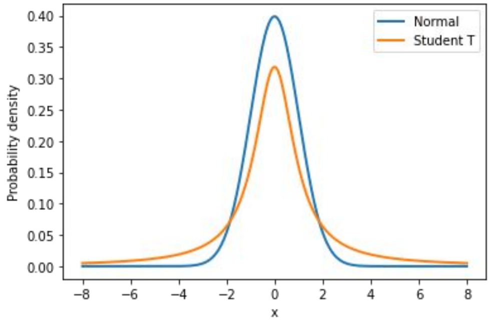
```python
import pymc3 as pm

X, y = linear_training_data()
with pm.Model() as linear_model:
    weights = pm.Normal("weights", mu=0, sigma=1)
    noise = pm.Gamma("noise", alpha=2, beta=1)
    y_observed = pm.Normal(
        "y_observed",
        mu=X @ weights,
        sigma=noise,
        observed=y,
    )
```



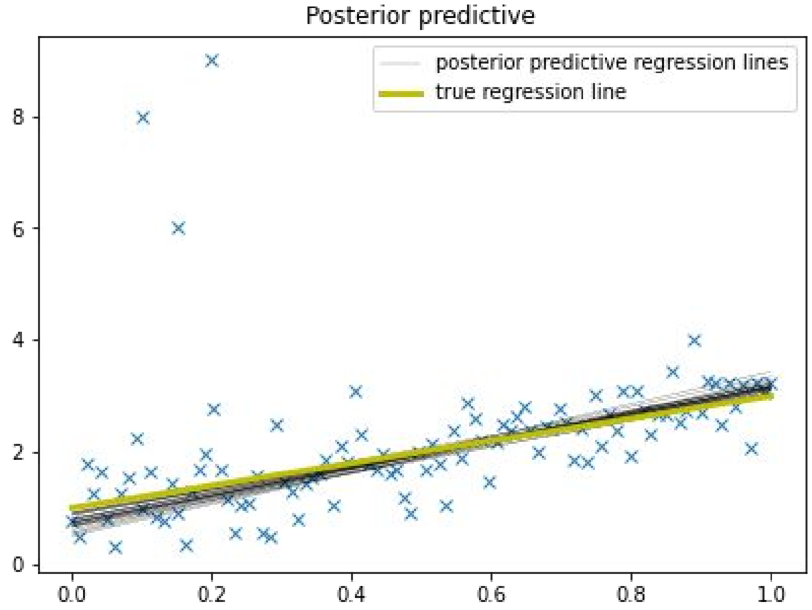
Robust linear regression

Normal likelihood



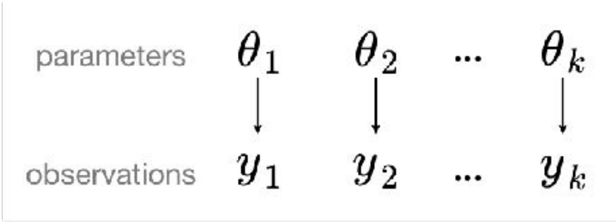


Student-T likelihood

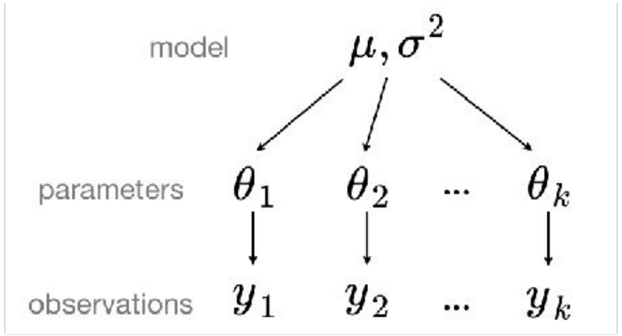


Hierarchical models for nested data

Single subject model



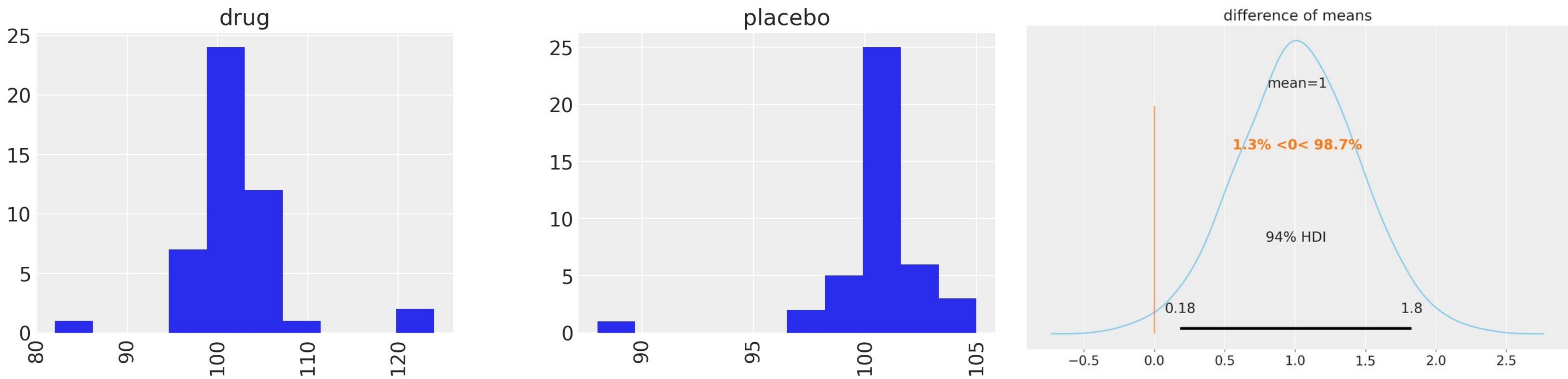
Hierarchical model



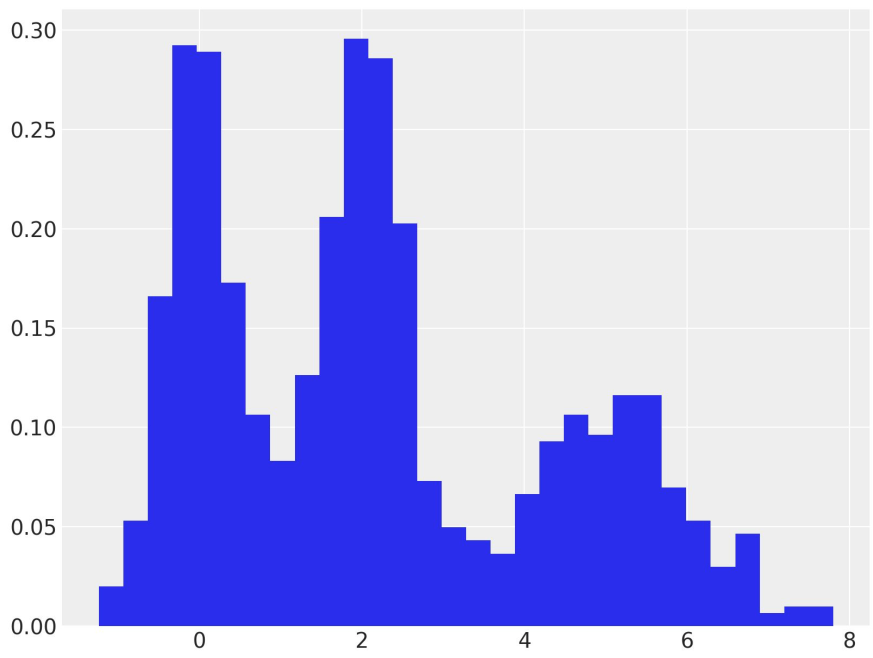
Useful specifically when you have multiple subjects.

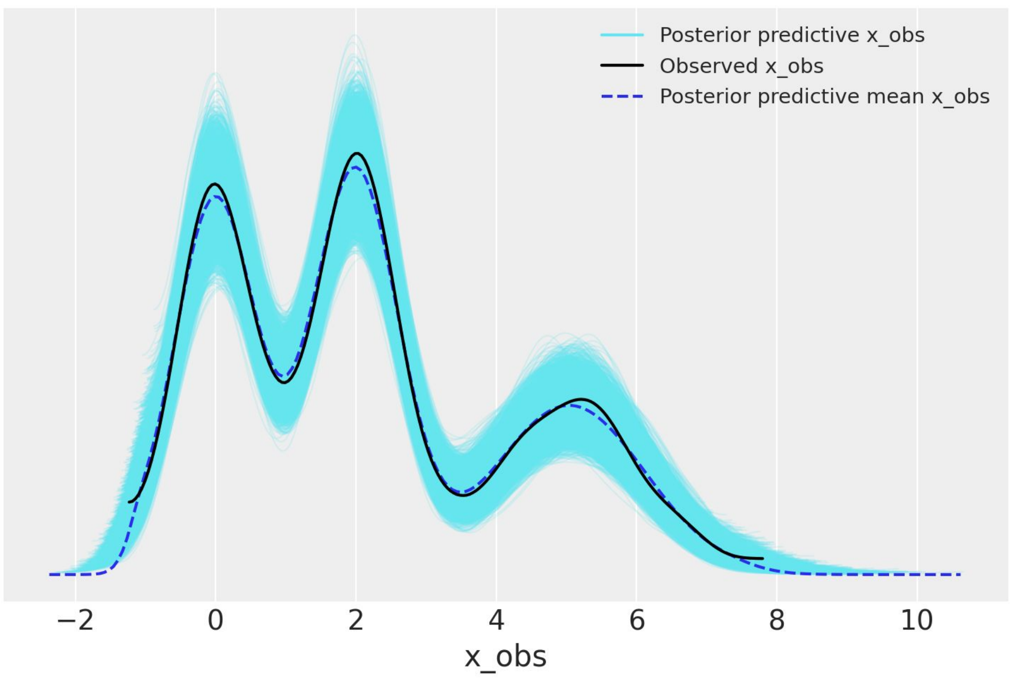
You can model differences *and* similarities.

Bayesian Estimation Supersedes the T-Test

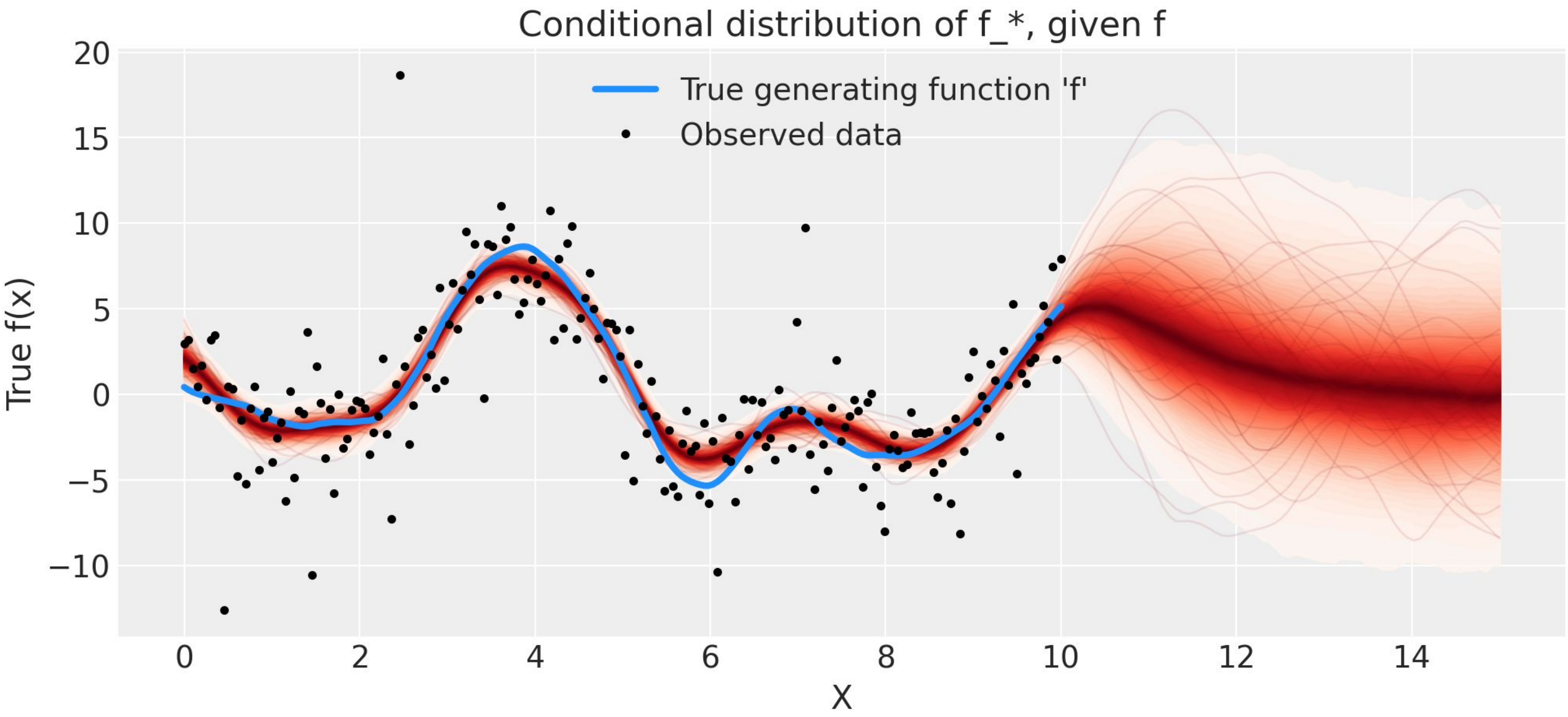


Mixture Models

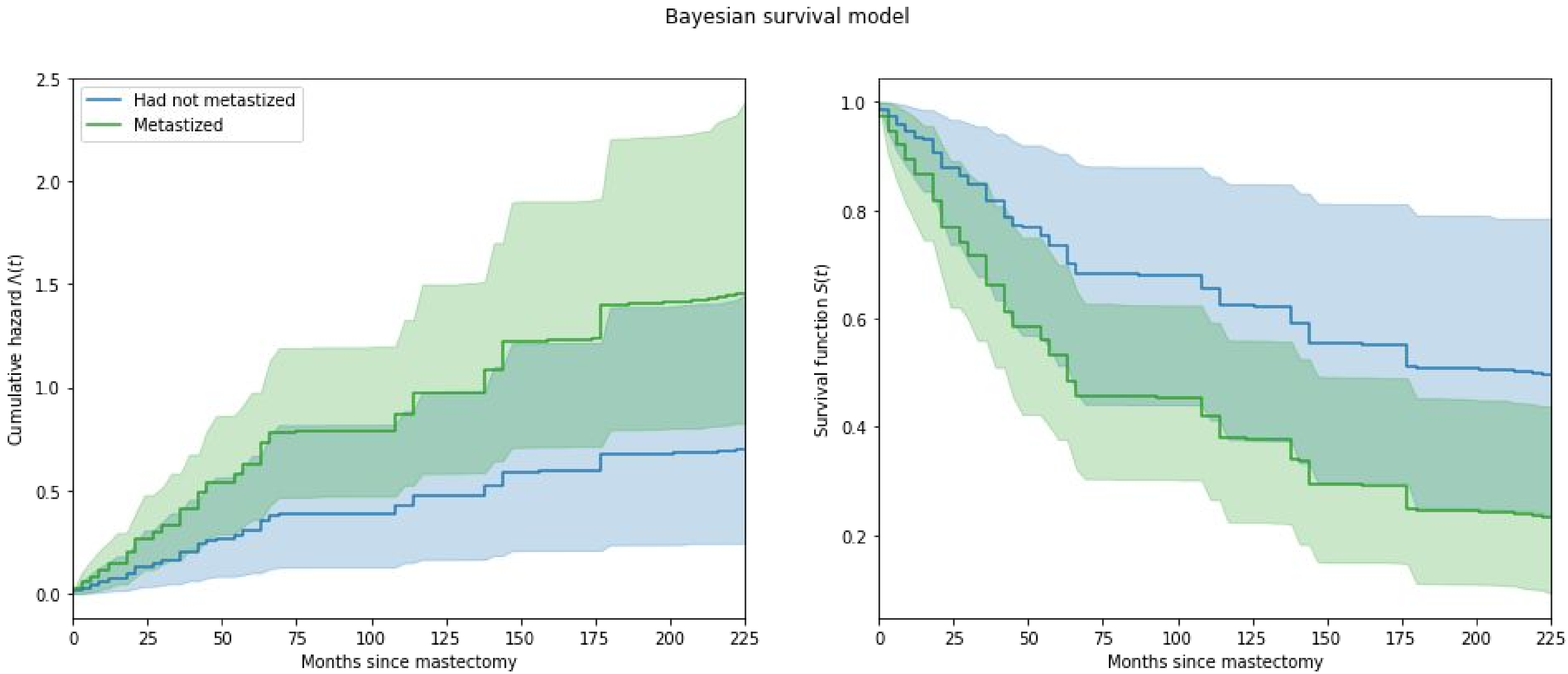




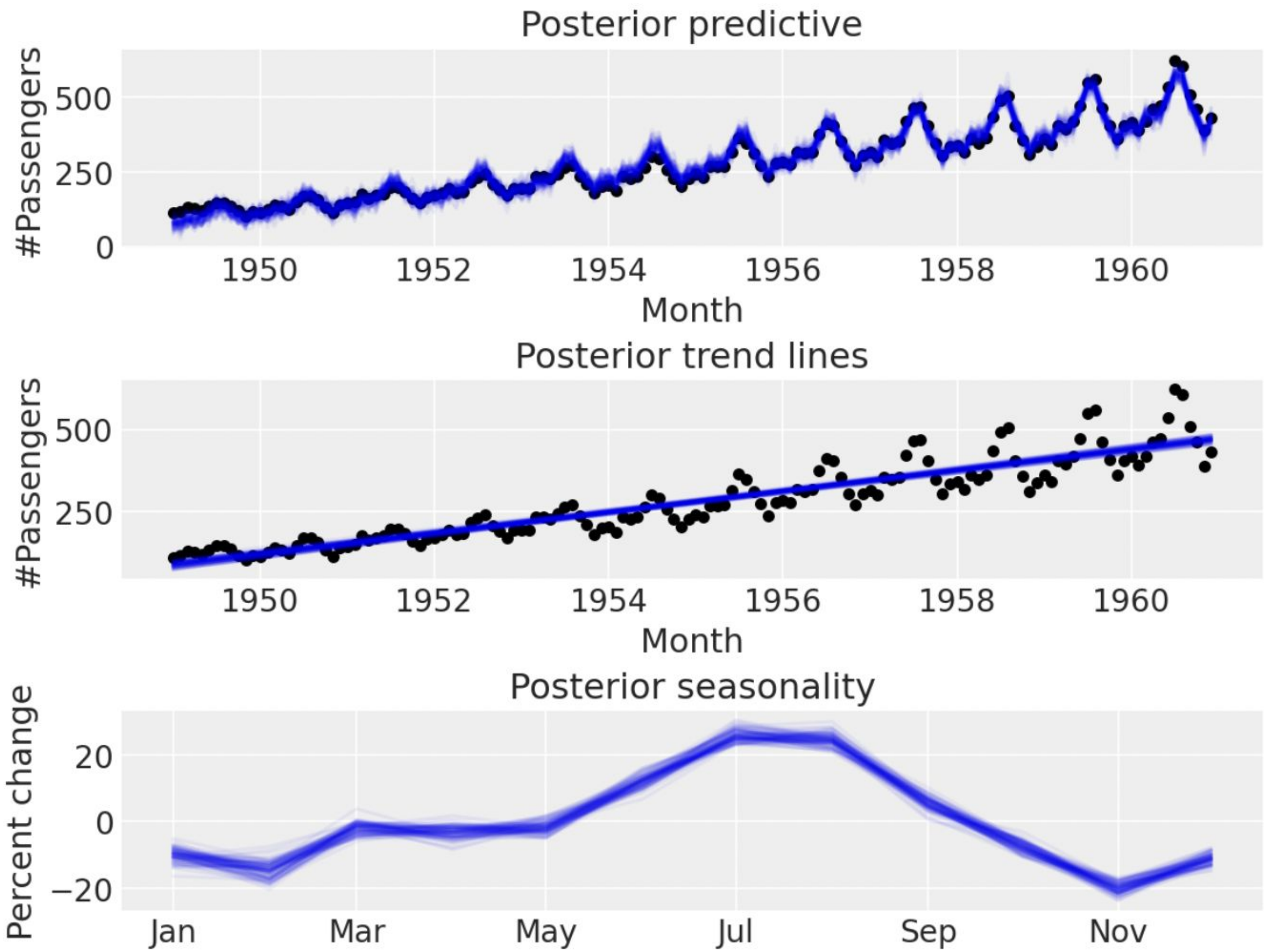
Gaussian Processes



Survival Analysis



Bayesian Time Series



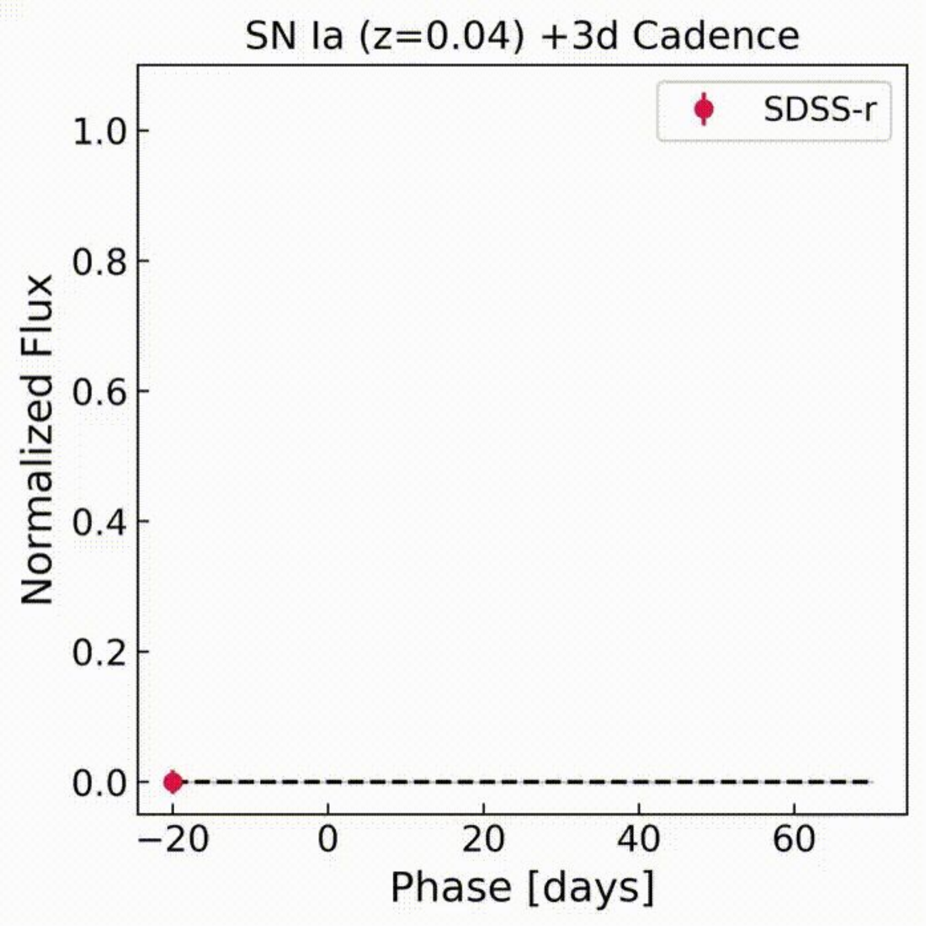
Wait.
Did you say Gaussian Process?

What do these have in common?

All special cases of Gaussian processes

Linear Regression Polynomial Regression Spline Regression	Time series (AR/ARMA) Neural Networks

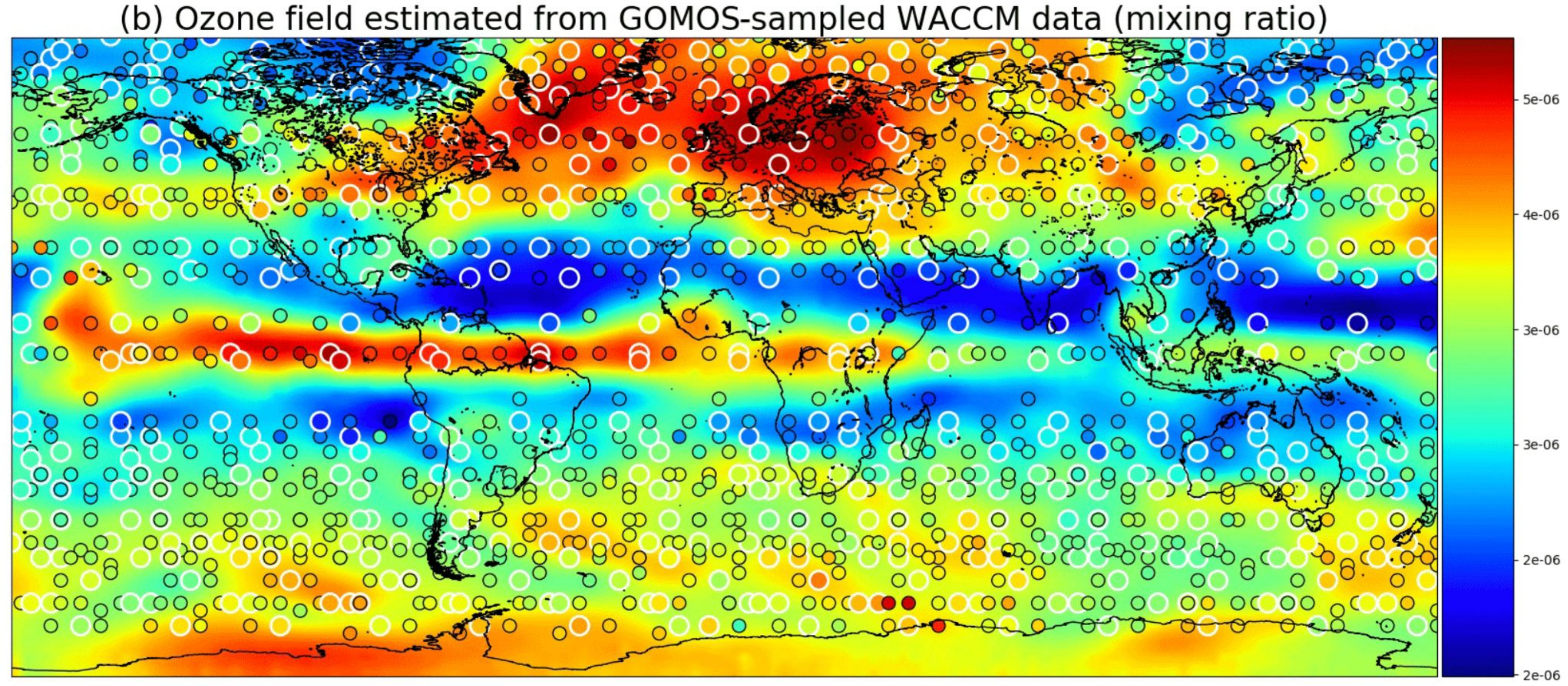
GPs can flexibly fit data



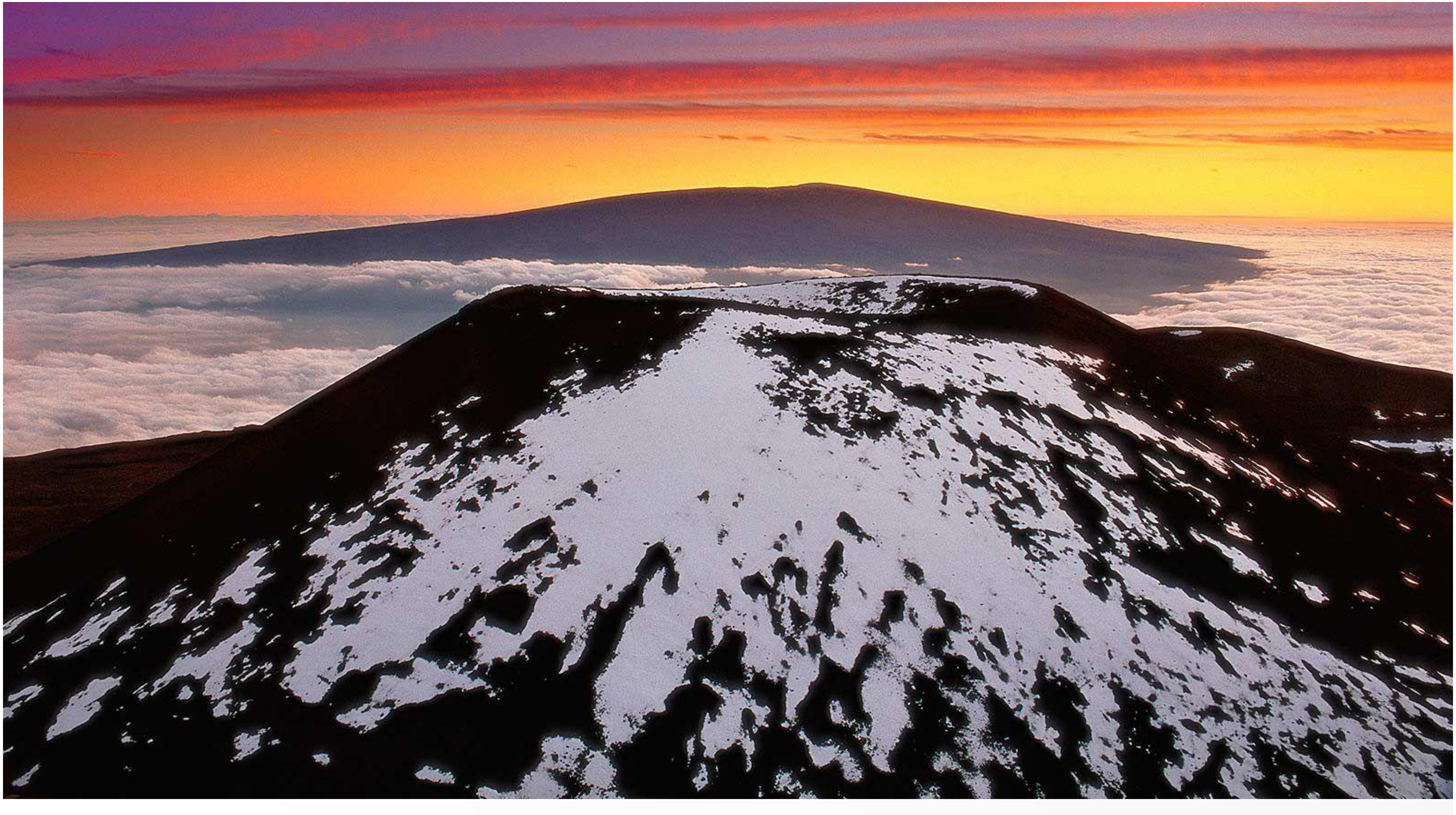
Using GP's to forecast Supernova Ia lightcurves,

by [Andy Tzanidakis]

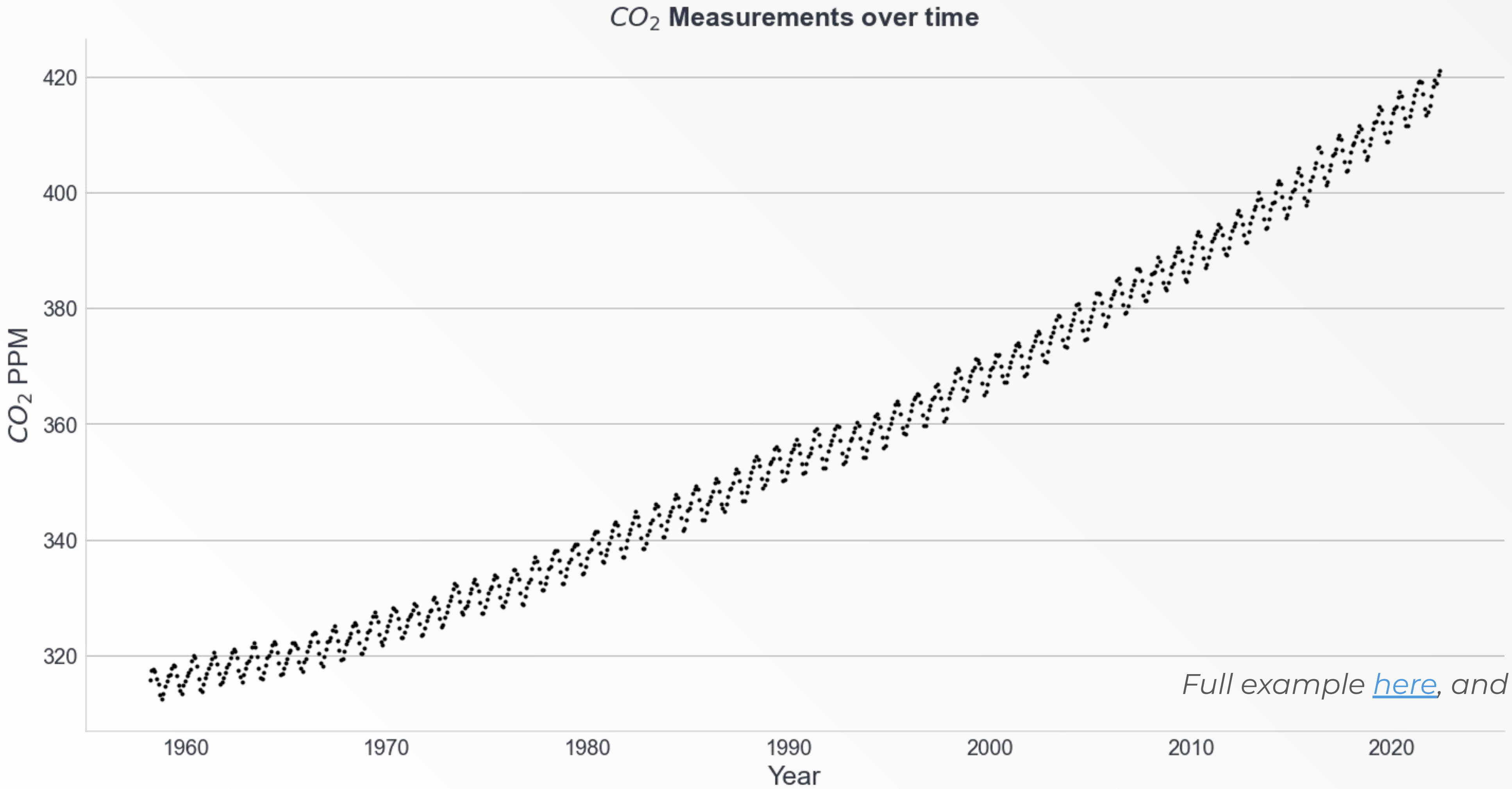
Geospatial Modeling



[Susiluoto et. al., 2020. Efficient multi-scale Gaussian process regression for massive remote sensing data with satGP v0.1.2]

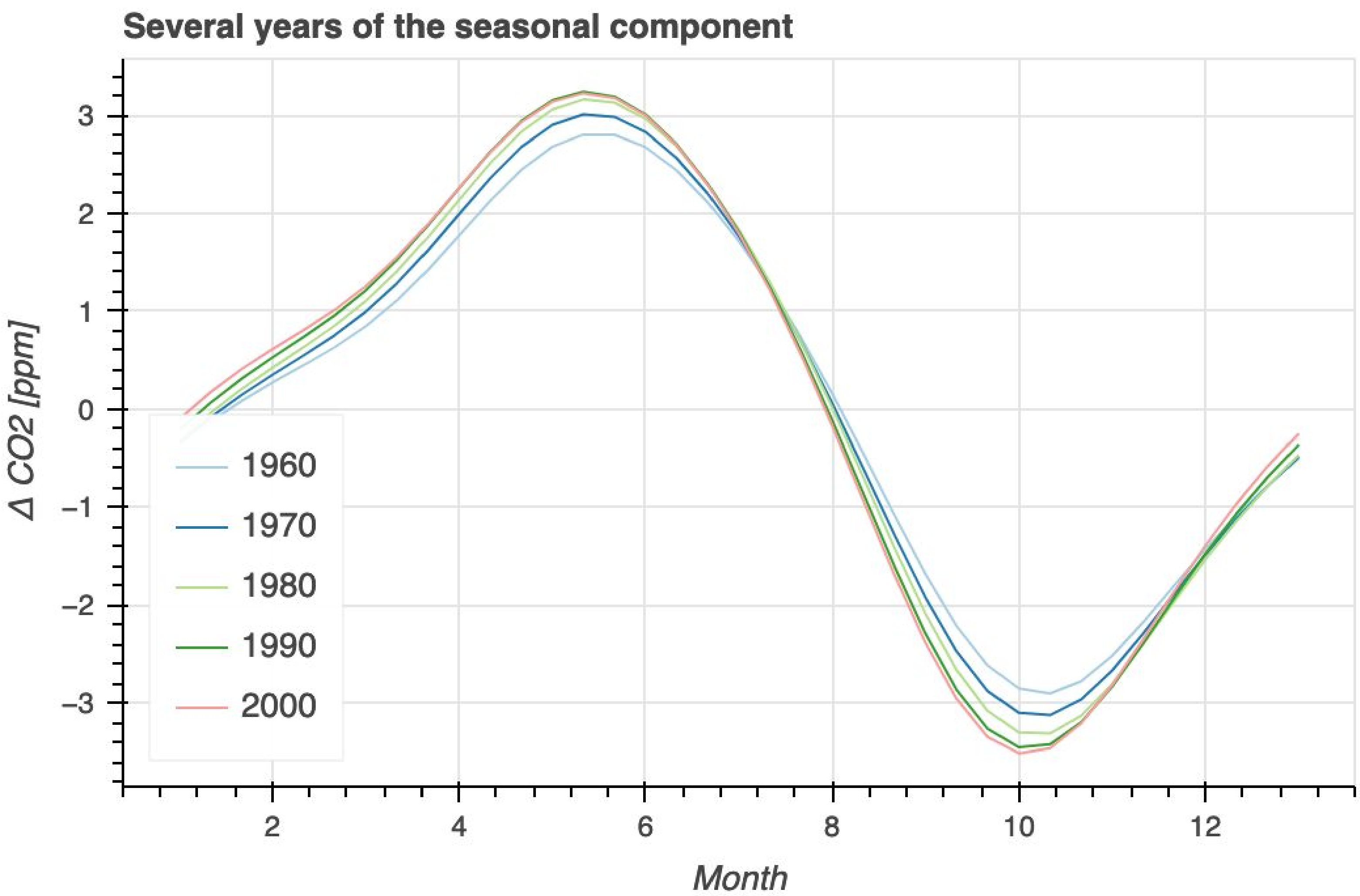


Show me the data



Full example [here](), and [there]()

Seasoning


Several years of the seasonal component
Δ CO2 [ppm]
3
2
1
0
−1
−2
−3
2
4
6
8
10
12
Month
1960
1970
1980
1990
2000

Modeling

The prior on CO_2 as a function of time is,

$$f(t) \sim \mathcal{GP}_{\text{slow}}(0,\ k_1(t, t')) + \mathcal{GP}_{\text{med}}(0,\ k_2(t, t')) + \mathcal{GP}_{\text{per}}(0,\ k_3(t, t')) + \mathcal{GP}_{\text{noise}}(0,\ k_4(t, t'))$$

PyMCing – Part 1

```
# long term trend
η_trend = pm.HalfCauchy("η_trend", beta=2, testval=2.0)
ℓ_trend = pm.Gamma("ℓ_trend", alpha=4, beta=0.1)
cov_trend = η_trend**2 * pm.gp.cov.ExpQuad(1, ℓ_trend)
gp_trend = pm.gp.Marginal(cov_func=cov_trend)
```

PyMCing – Part 2

```python
# small/medium term irregularities
η_med = pm.HalfCauchy("η_med", beta=0.5, testval=0.1)
ℓ_med = pm.Gamma("ℓ_med", alpha=2, beta=0.75)
α = pm.Gamma("α", alpha=5, beta=2)
cov_medium = η_med**2 * pm.gp.cov.RatQuad(1, ℓ_med, α)
gp_medium = pm.gp.Marginal(cov_func=cov_medium)
```

PyMCing – Part 3

```
# yearly periodic component x long term trend
η_per = pm.HalfCauchy("η_per", beta=2, testval=1.0)
ℓ_pdecay = pm.Gamma("ℓ_pdecay", alpha=10, beta=0.075)
period = pm.Normal("period", mu=1, sigma=0.05)
ℓ_psmooth = pm.Gamma("ℓ_psmooth ", alpha=4, beta=3)
cov_seasonal = (
    η_per**2 * pm.gp.cov.Periodic(1, period, ℓ_psmooth) * pm.gp.cov.Matern52(1, ℓ_pdecay)
)
gp_seasonal = pm.gp.Marginal(cov_func=cov_seasonal)
```

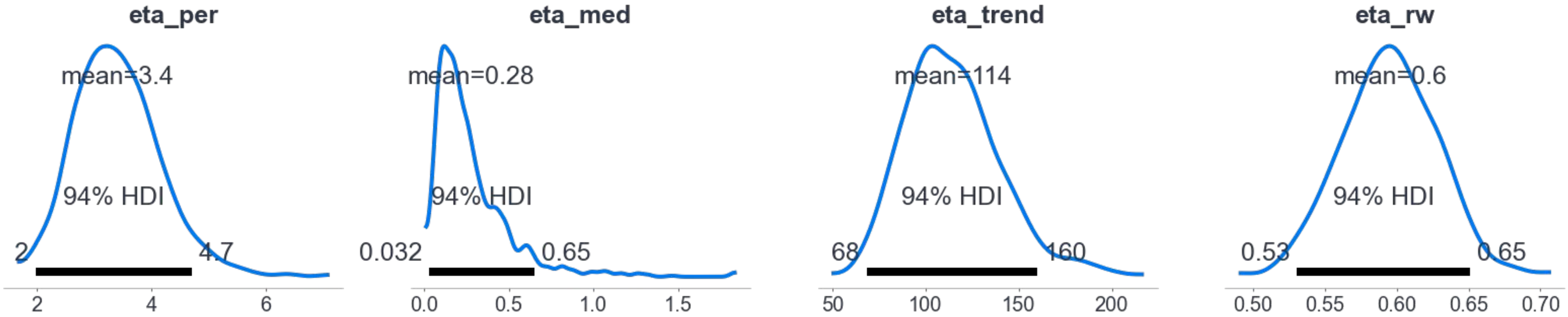
PyMCing – Part 4

```
# noise model
η_noise = pm.HalfNormal("η_noise", sigma=0.5, testval=0.05)
ℓ_noise = pm.Gamma("ℓ_noise", alpha=2, beta=4)
σ = pm.HalfNormal("σ", sigma=0.25, testval=0.05)
cov_noise = η_noise**2 * pm.gp.cov.Matern32(1, ℓ_noise) + pm.gp.cov.WhiteNoise(σ)
```

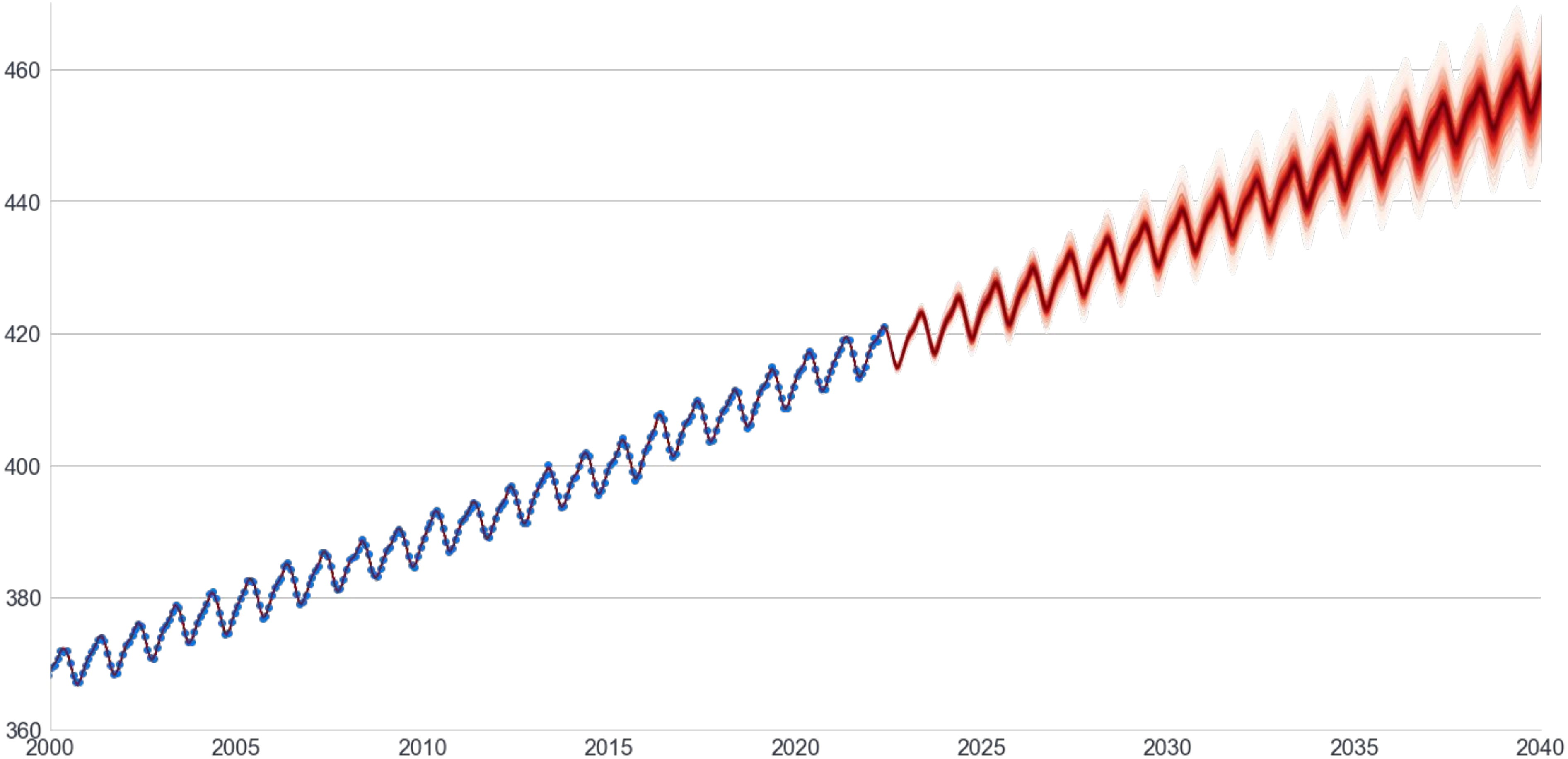
PyMCing – Part 5

```
# The Gaussian process is a sum of these three components
gp = gp_seasonal + gp_medium + gp_trend

# Since the normal noise model and the GP are conjugates, we
y_ = gp.marginal_likelihood("y", X=t, y=y, noise=cov_noise)
```

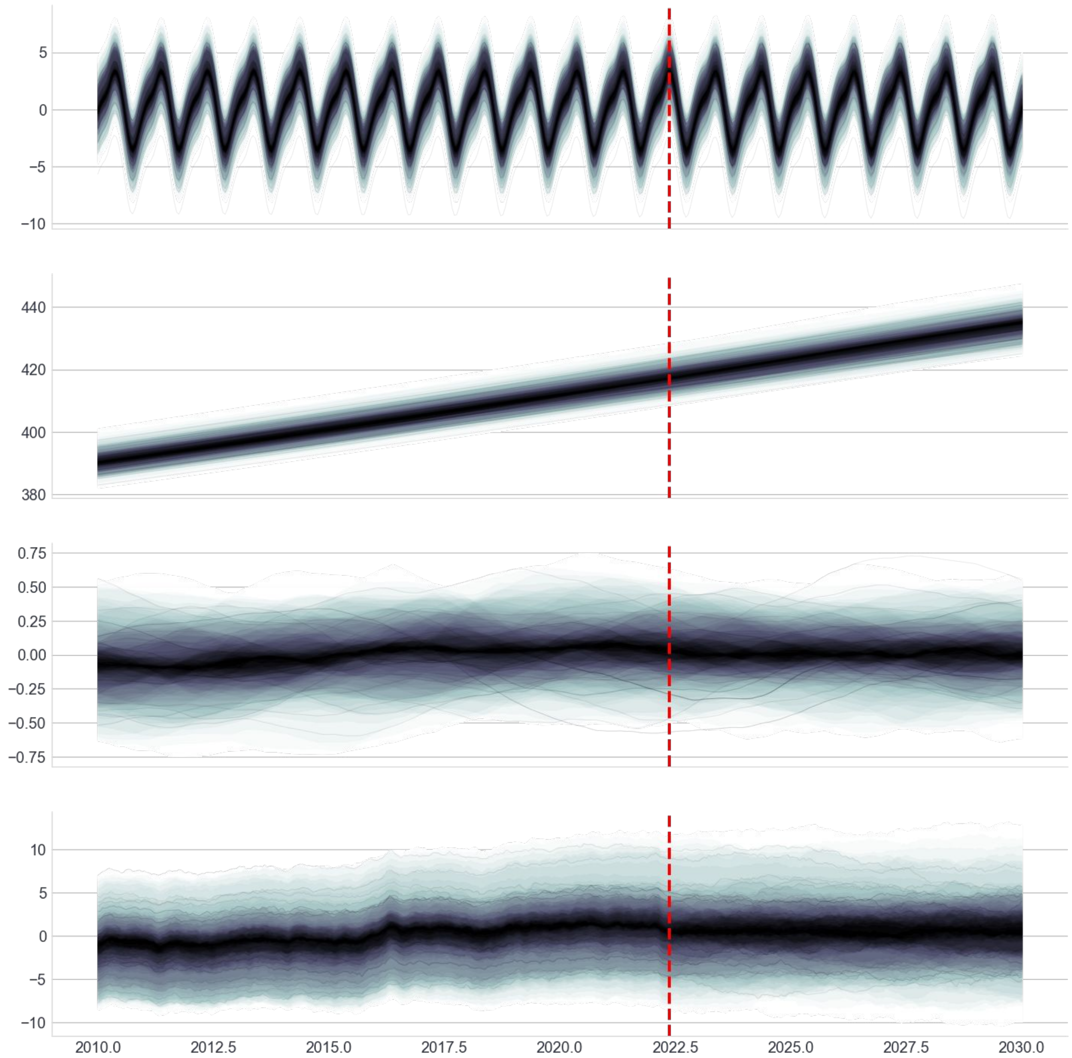
Show me the results!



Predicting the future



Decomposing the GP's elements



Statisticians just wanna leeeeeaaarn!

Intuitive Bayes
Introductory Course

'Learn Bayes Stats'
podcast

PyMC Docs

PyMC Educational Resources

PyMC Discourse

www.intuitivebayes.com

Gaussian Processes

INTUITIVE BAYES

- Bill Engels
- Ravin Kumar

Introductory Course

INTUITIVE BAYES

- Ravin Kumar
- Alex Andorra
- Thomas Wiecki

www.intuitivebayes.com

Advanced Regression

INTUITIVE BAYES

- Alex Andorra
- Tomas Capretto
- Ravin Kumar

Congreso Bayesiano Plurinacional

4 y 5 de agosto de 2023 - Santiago del Estero, Argentina

El Congreso Bayesiano Plurinacional tiene por objetivo reunir a estudiantes, docentes, investigadores, practicantes y expertos que utilicen, desarrollen o implementen métodos Bayesianos, en la academia o industria.

http://bayesdelsur.com.ar/

Need custom solutions to your most challenging data science problems?

Contact us @

alex.andorra@pymc-labs.io

Further info about PyMC

More details about what we do

The Bayesian super power in audio format