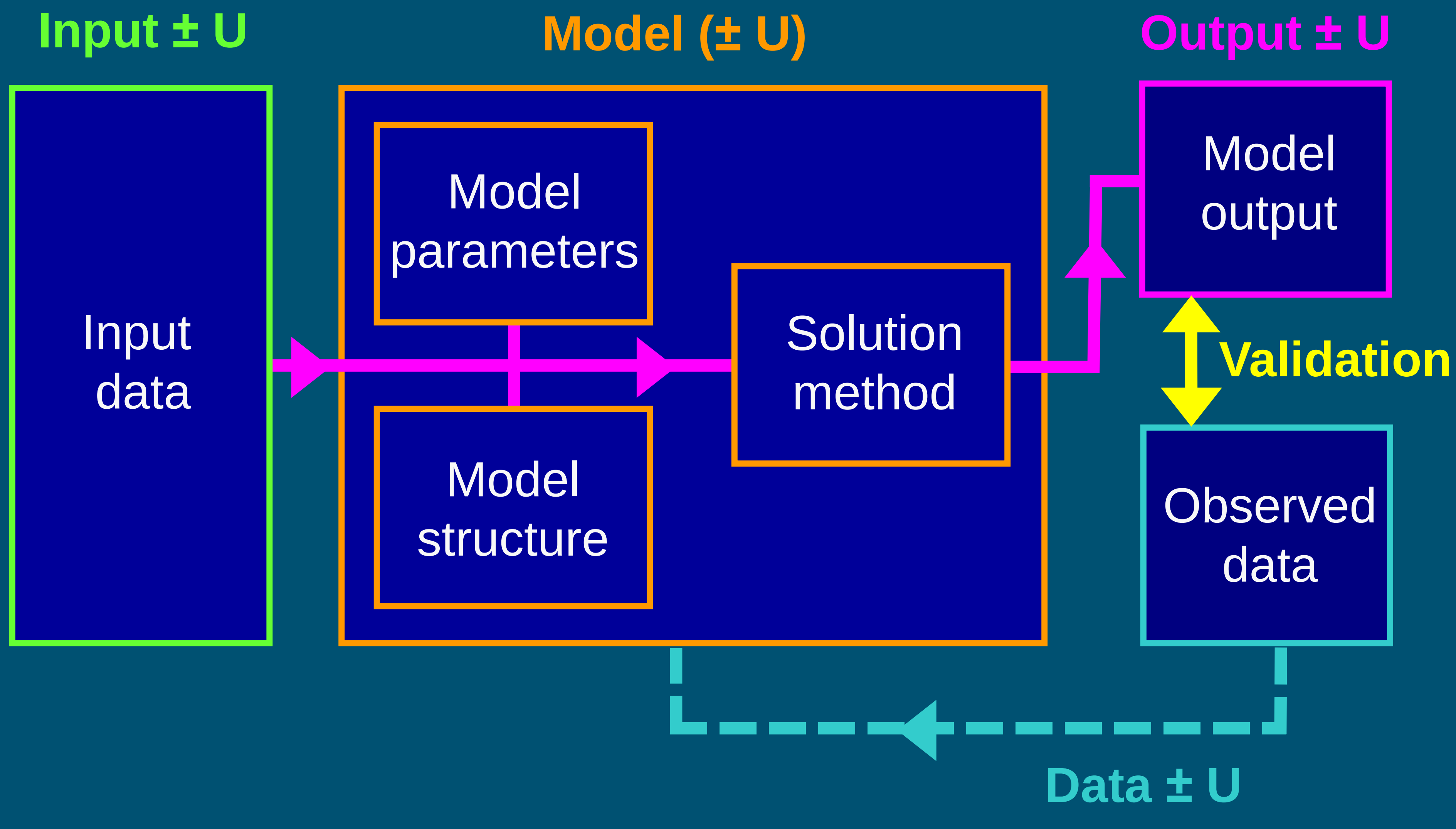
Uncertainty Analysis and Statistical Validation of Spatial Environmental Models

PE&RC Course 9-13 December 2024

Gerard Heuvelink and Sytze de Bruin

WAGENINGEN UNIVERSITY
WAGENINGEN UR

Uncertainty propagation and model validation overview



output = g(inputs, model parameters)

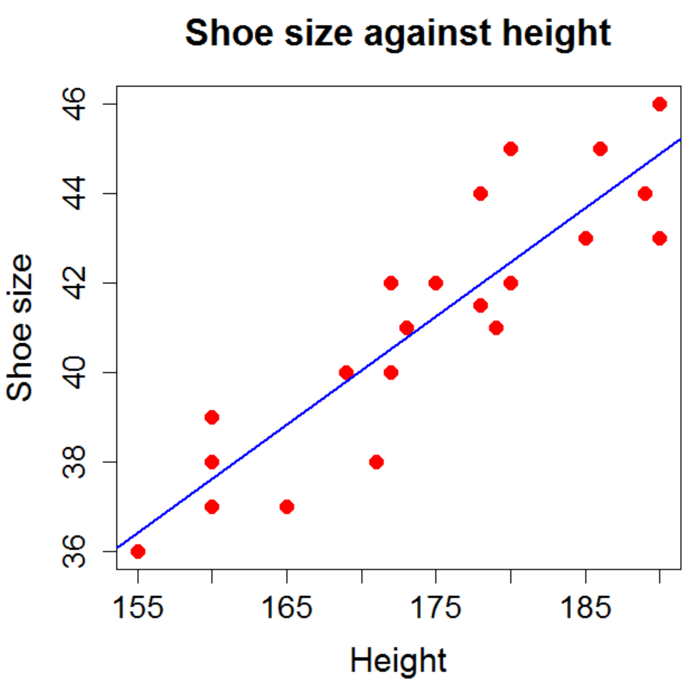
$$U = g(A_1, \dots, A_m, \theta)$$

For example:

$$shoe\ size = \beta_0 + \beta_1 \cdot height + \varepsilon$$

model structure error $N(0, \sigma^2)$

height	shoesize
172	40
179	41
178	41.5
160	38
180	42
180	45
189	44
180	42
186	45
169	40

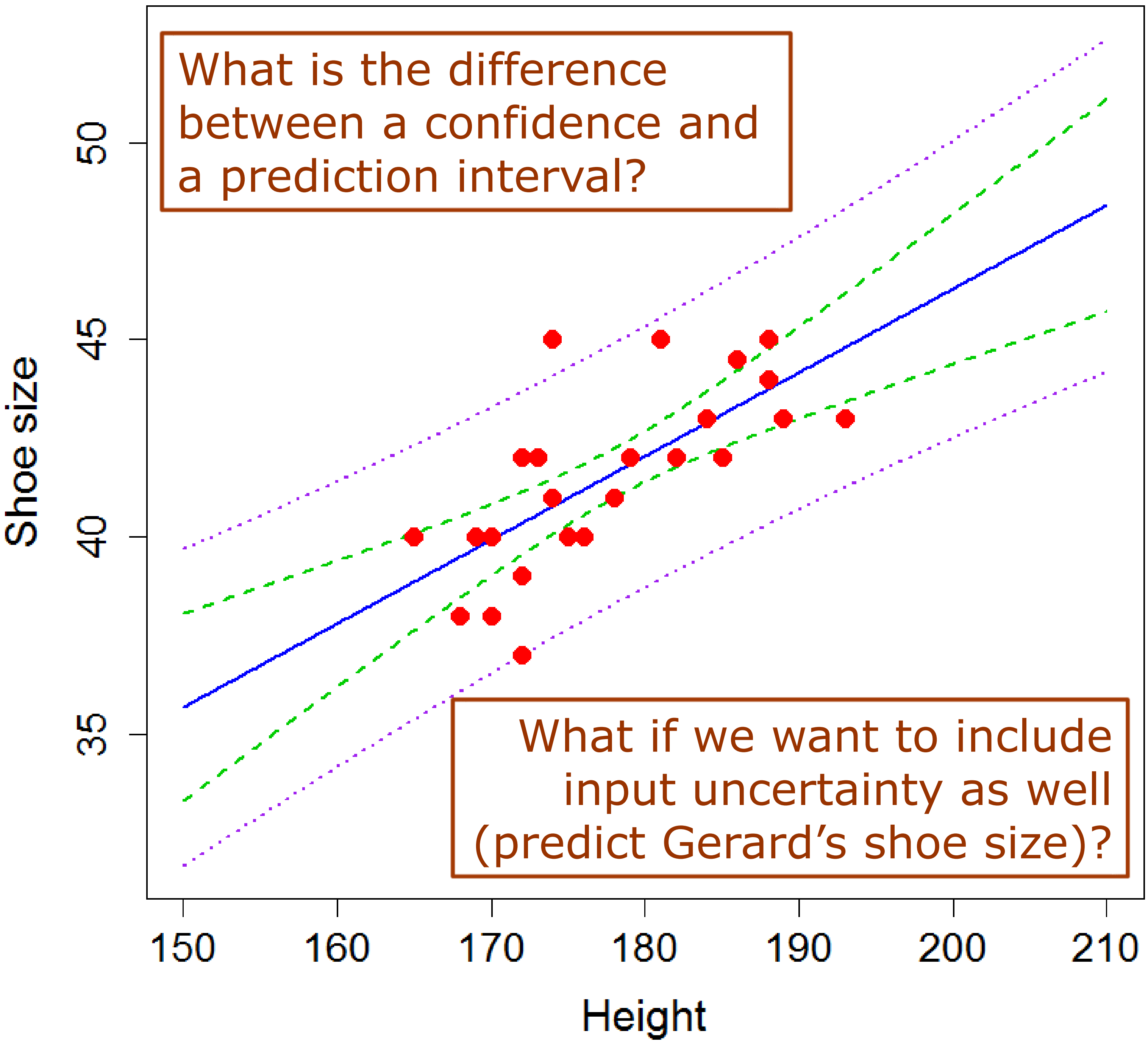


If we knew the joint probability distribution of the uncertain model parameters θ then uncertainty propagation would be easy

- For the linear regression model we have $\theta = (\beta_0, \beta_1, \sigma^2)$
- Since it is a simple linear model, the model parameter estimates and their uncertainty can be computed analytically
- Running function *lm* in R gives all required information: the estimate $\hat{\theta}$ as well as its uncertainty $Var(\hat{\theta} - \theta)$
- With this information we can calculate how model uncertainty propagates, for example using the Taylor series or Monte Carlo method

```
> slm = lm(shoesize~height, data =sl)
> summary(slm)

Call:
lm(formula = shoesize ~ height, data = sl)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.2851 -0.6338 -0.4053  0.9113  2.5326 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.18000    4.49417  -0.263    0.795    
height       0.24249    0.02575   9.416 5.49e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.271 on 21 degrees of freedom
Multiple R-squared:  0.8085,	Adjusted R-squared:  0.7994 
F-statistic: 88.67 on 1 and 21 DF,  p-value: 5.494e-09
```

```
> vcov(slm)
            (Intercept)        height
(Intercept)    20.19757 -0.1155300376
height         -0.11553  0.0006631372
```


What is the difference between a confidence and a prediction interval?
What if we want to include input uncertainty as well (predict Gerard's shoe size)?
Shoe size
35
40
45
50
Height
150
160
170
180
190
200
210

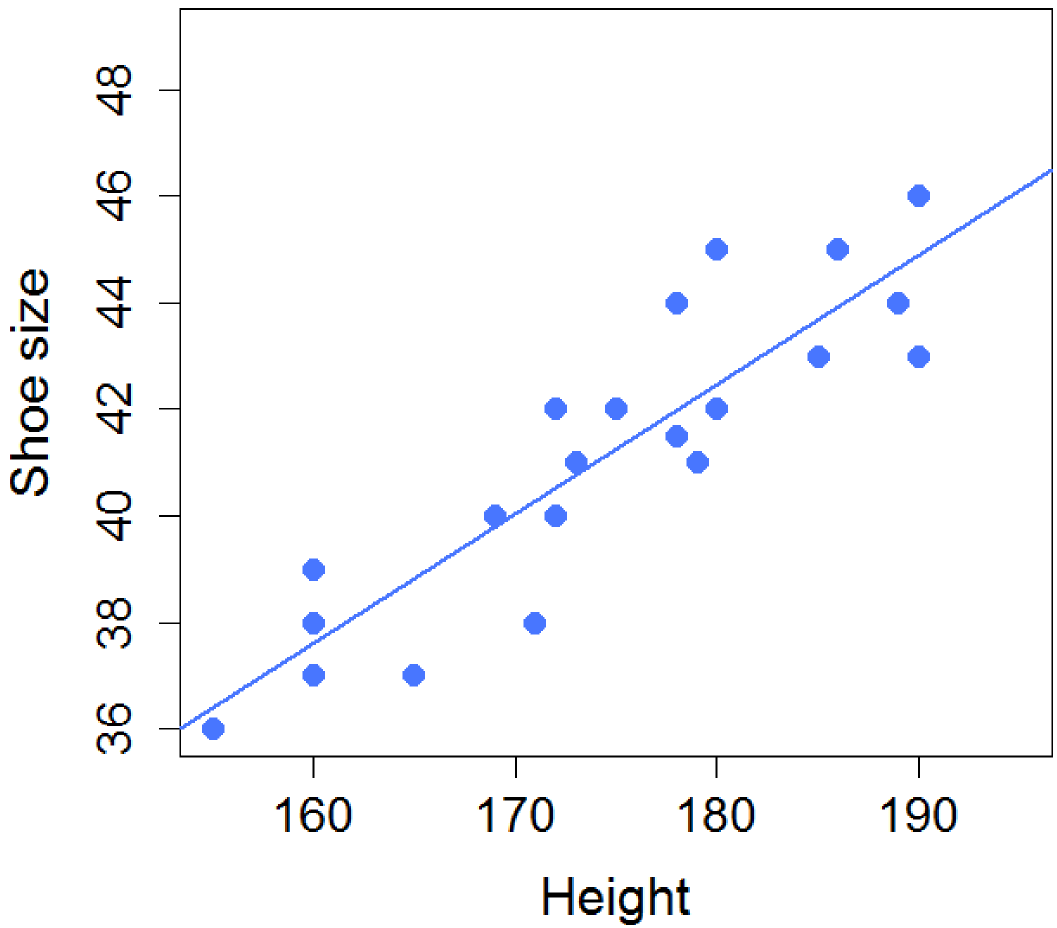
For complex models, quantification of model uncertainty is much more difficult

- Important to recognise that model uncertainty is case-dependent, there is no universal model uncertainty. The same model may perform well in one area and much worse in another
- In other words, we need observations of the model output to assess the model uncertainty for a particular case (unless we trust that experts can quantify model parameter and model structure uncertainty for a given case study)
- For linear regression we just learnt that this can be done analytically, but for more complex models (e.g. erosion, nitrate leaching, groundwater flow) analytical solutions usually do not exist
- Alternative is to use Bayesian calibration: works in almost all cases but is computationally demanding

Bayesian calibration

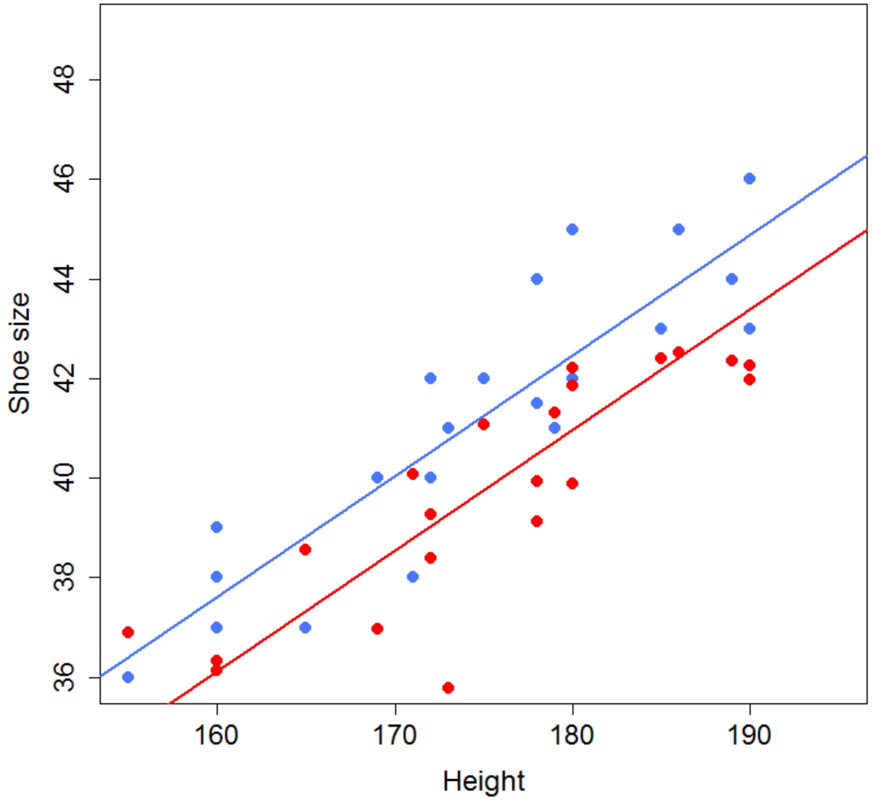
- Looks for $p(\theta|data)$, the probability distribution of the parameters conditional to the 'data' (= observations of the model output)
- It starts with an initial guess $p(\theta)$ and then uses the data to update knowledge about θ by computing $p(\theta|data)$
- $p(\theta|data)$ is called the posterior distribution, while $p(\theta)$ is called the prior distribution
- Values for θ that correspond well with the observations get a larger probability (density) than values for θ that do not correspond well with the observations
- Note that the result is not a 'point estimate' of the model parameters, but instead their full (joint) probability distribution

Consider the shoe size regression model, the data (blue dots) tell us which parameter values are likely and which are not

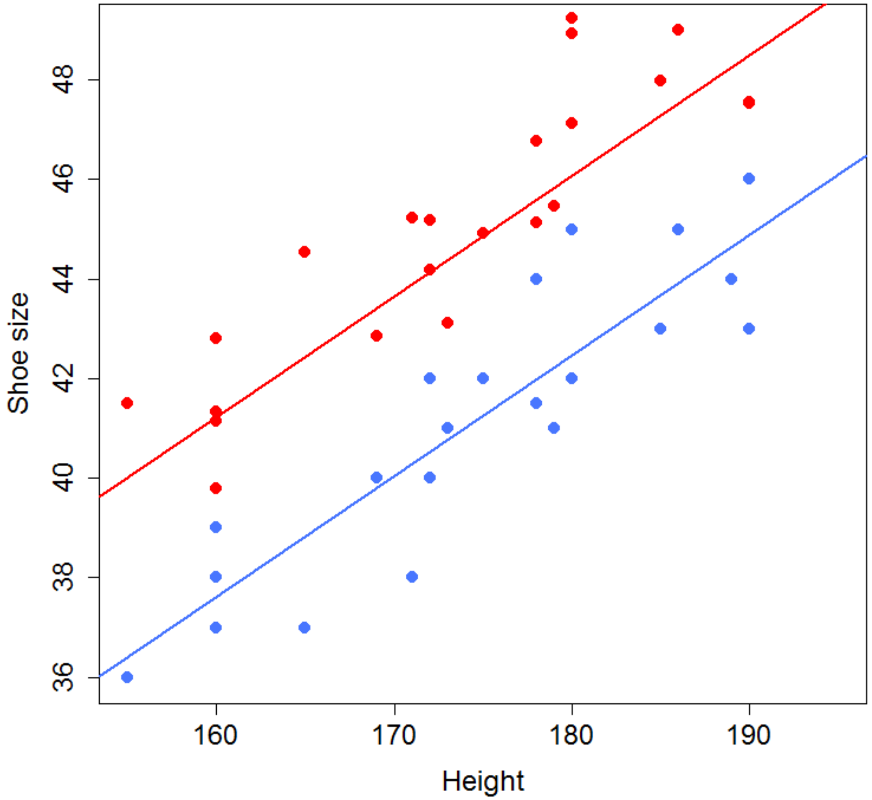


Unlikely values for the intercept β_0

too small

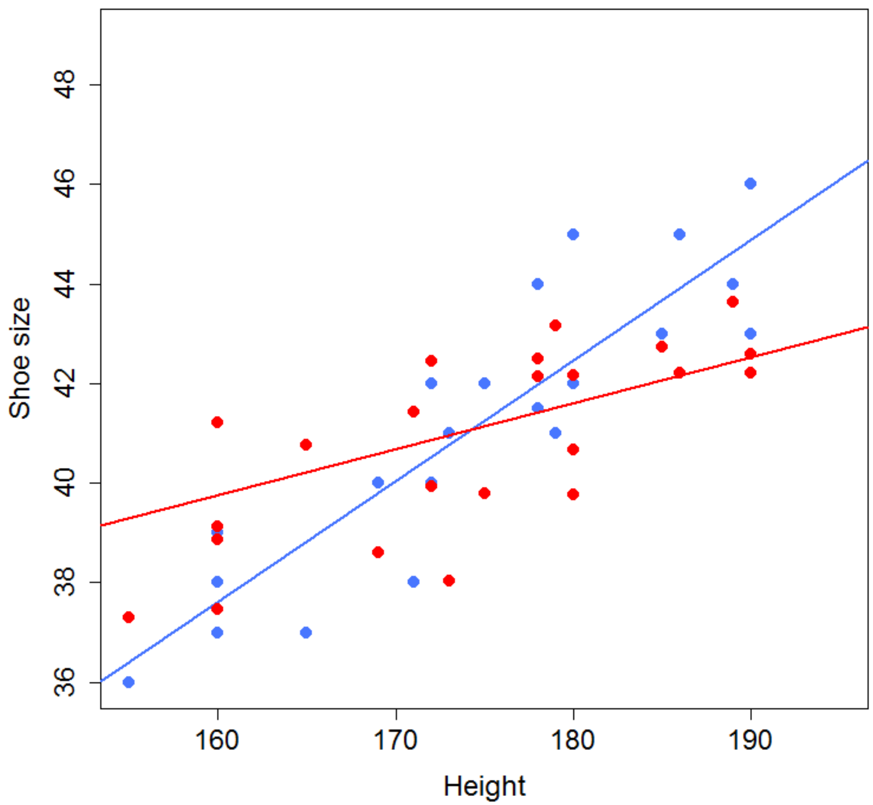


too large

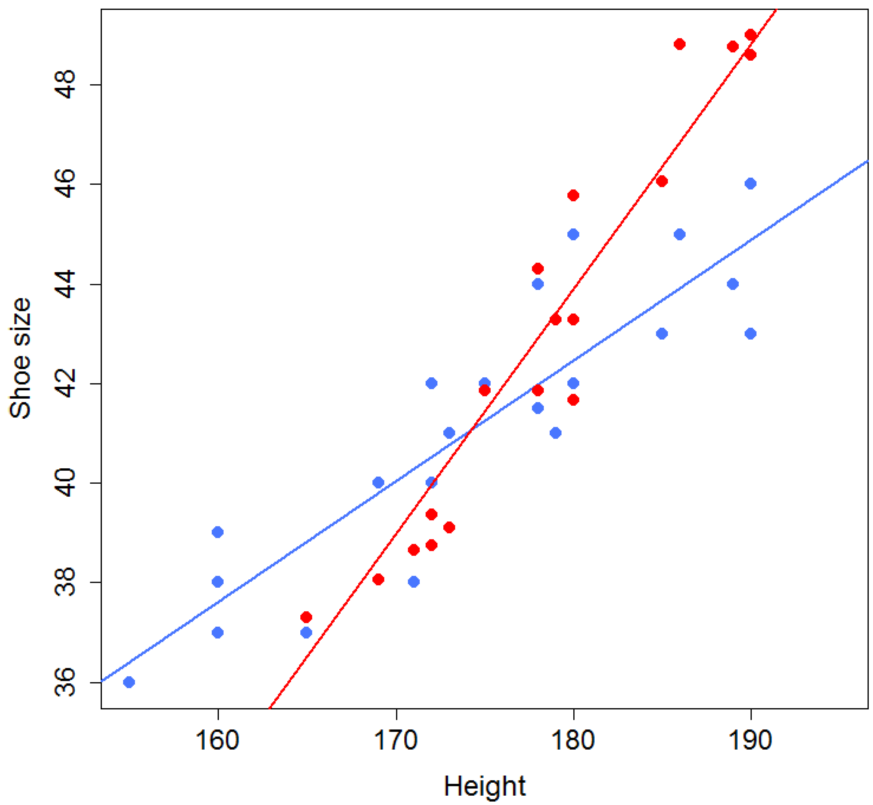


Unlikely values for the slope β_1

too gentle

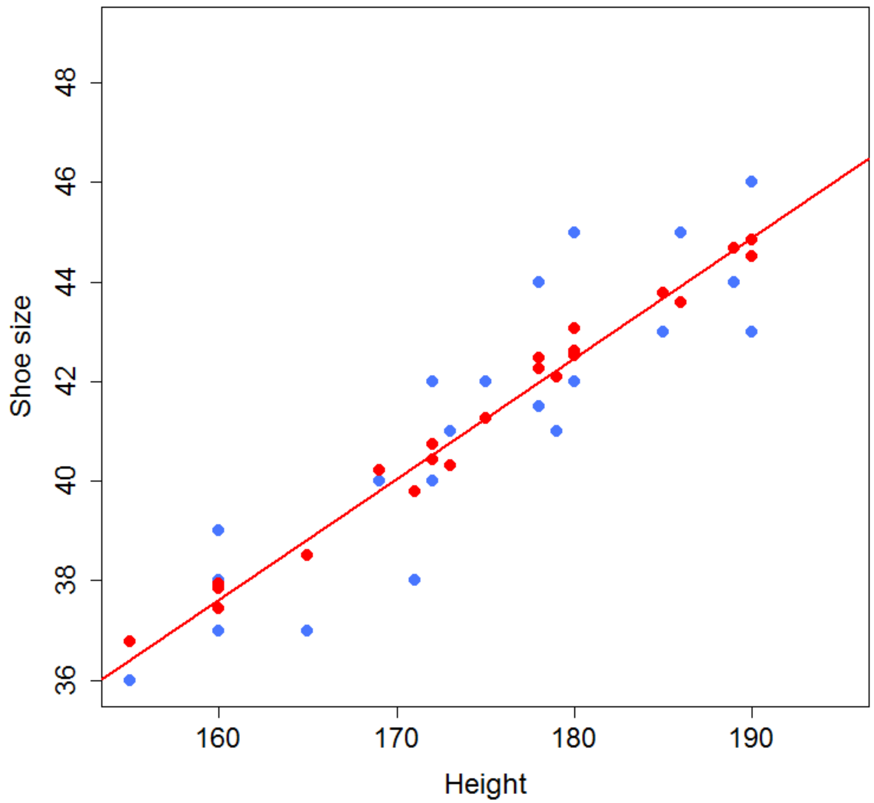


too steep

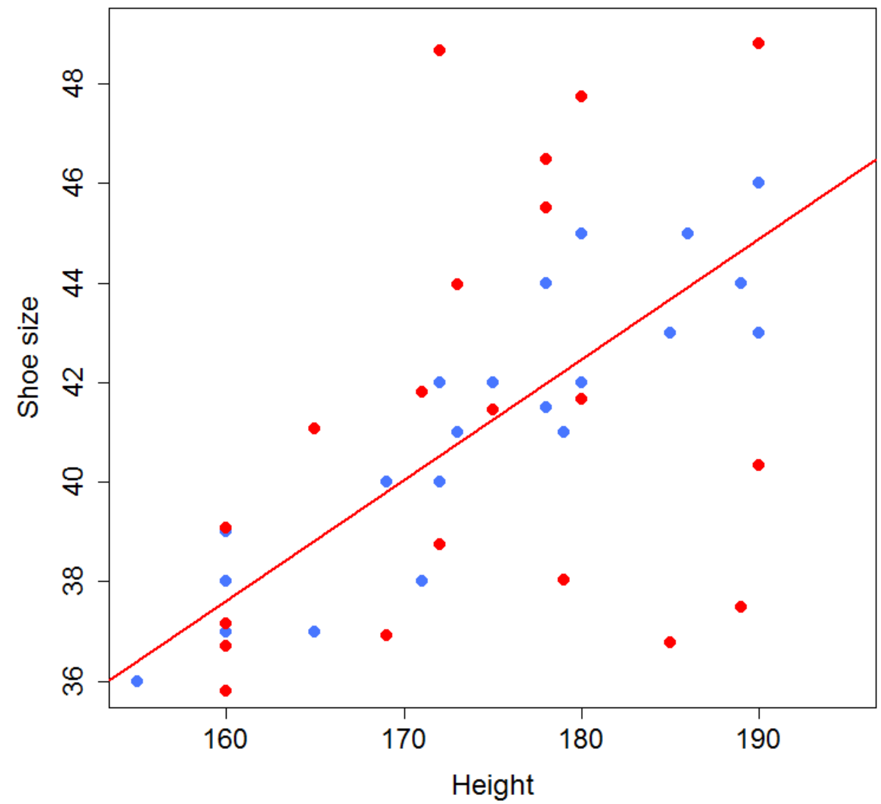


Unlikely values for the residual variance σ^2

too small



too large



Calculating the conditional probability

- But calculating $p(\theta|data)$ is not easy, it would be so much easier to calculate $p(data|\theta)$. [Why? How?]
- Therefore make use of Bayes' law:

$$P(A|B) = \frac{P(A)}{P(B)} \cdot P(B|A)$$

- In our case:

$$p(\theta|data) = \frac{p(\theta)}{p(data)} \cdot p(data|\theta)$$

Exercise 1

Try Bayes' law yourself (tip: make also use of the 'Law of Total Probability'):

- Thunderstorms happen on average 2 out of 100 days (2%)
- The probability of rain in case of a thunderstorm is 80%
- The probability of rain in case of no thunderstorm is 5%
- What is the probability of thunderstorm in case of rain?

For more insight into Bayes' law, check out this 15 minute video:

https://www.youtube.com/watch?v=HZGCoVF3YvM

The posterior is determined by the prior and the likelihood

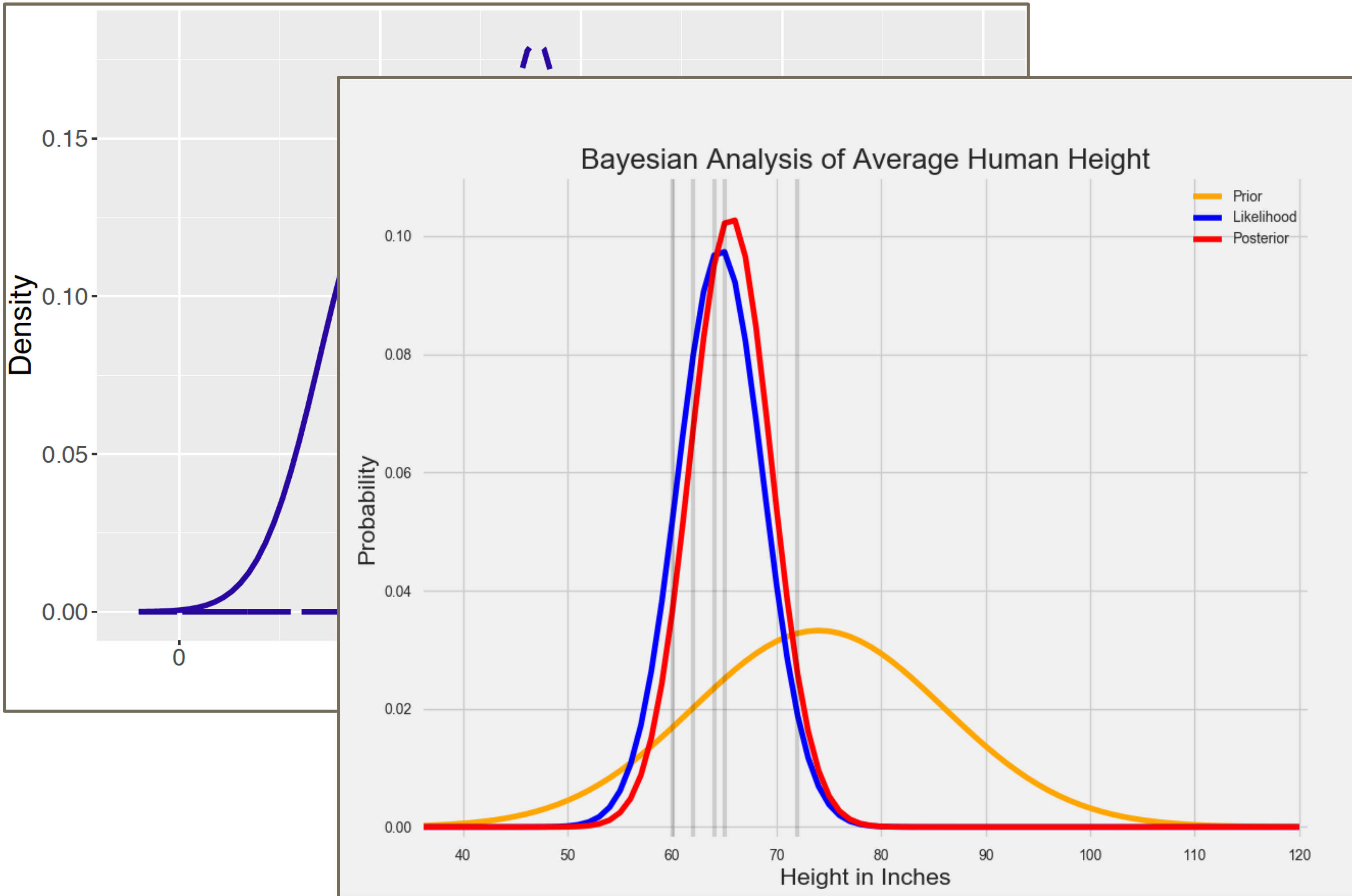
$$p(\theta|data) = \frac{p(\theta)}{p(data)} \cdot p(data|\theta)$$

For a given dataset $p(data)$ is constant, so we can also write:

$$p(\theta|data) = c \cdot p(\theta) \cdot p(data|\theta)$$

$$p(\theta|data) \propto p(\theta) \cdot p(data|\theta)$$

Some examples (you will also compute some in the computer practical)



$$p(\theta|data) = \frac{p(\theta)}{p(data)} \cdot p(data|\theta)$$

- The likelihood $p(data|\theta)$ is easy because the probability distribution of the model output follows directly from the model once the model parameters (and inputs) are known, since: $data = g(inputs, \theta)$
- The prior $p(\theta)$ is derived from expert judgement, and if experts have no idea we can use a flat or uninformative prior (very wide distribution)
- The difficulty is with $p(data)$, but note from the law of total probability that we can rewrite the equation as:

$$p(\theta|data) = \frac{p(\theta) \cdot p(data|\theta)}{\int_{all\ \theta} p(\theta) \cdot p(data|\theta) d\theta}$$

$$p(\theta|data) = \frac{p(\theta) \cdot p(data|\theta)}{\int_{all\ \theta} p(\theta) \cdot p(data|\theta) d\theta}$$

- However, this is not a very practical solution because the multidimensional integral in the denominator is very hard to compute (analytical solutions do not exist in general, and multi-dimensional numerical integration is a lot of work)
- A better alternative is to make use of ratios:

$$\frac{p(\theta_1|data)}{p(\theta_2|data)} = \frac{p(\theta_1) \cdot p(data|\theta_1)}{p(\theta_2) \cdot p(data|\theta_2)}$$

$$\frac{p(\theta_1|data)}{p(\theta_2|data)} = \frac{p(\theta_1) \cdot p(data|\theta_1)}{p(\theta_2) \cdot p(data|\theta_2)}$$

- We can calculate this ratio for all combinations of θ_1 and θ_2
- If we could generate a large set (sample) of θ values such that the relative frequencies of simulated θ values satisfy the ratio given by the equation above then we would have a sample from the posterior distribution
- This is achieved with Markov chain Monte Carlo

Markov chain Monte Carlo (MCMC)

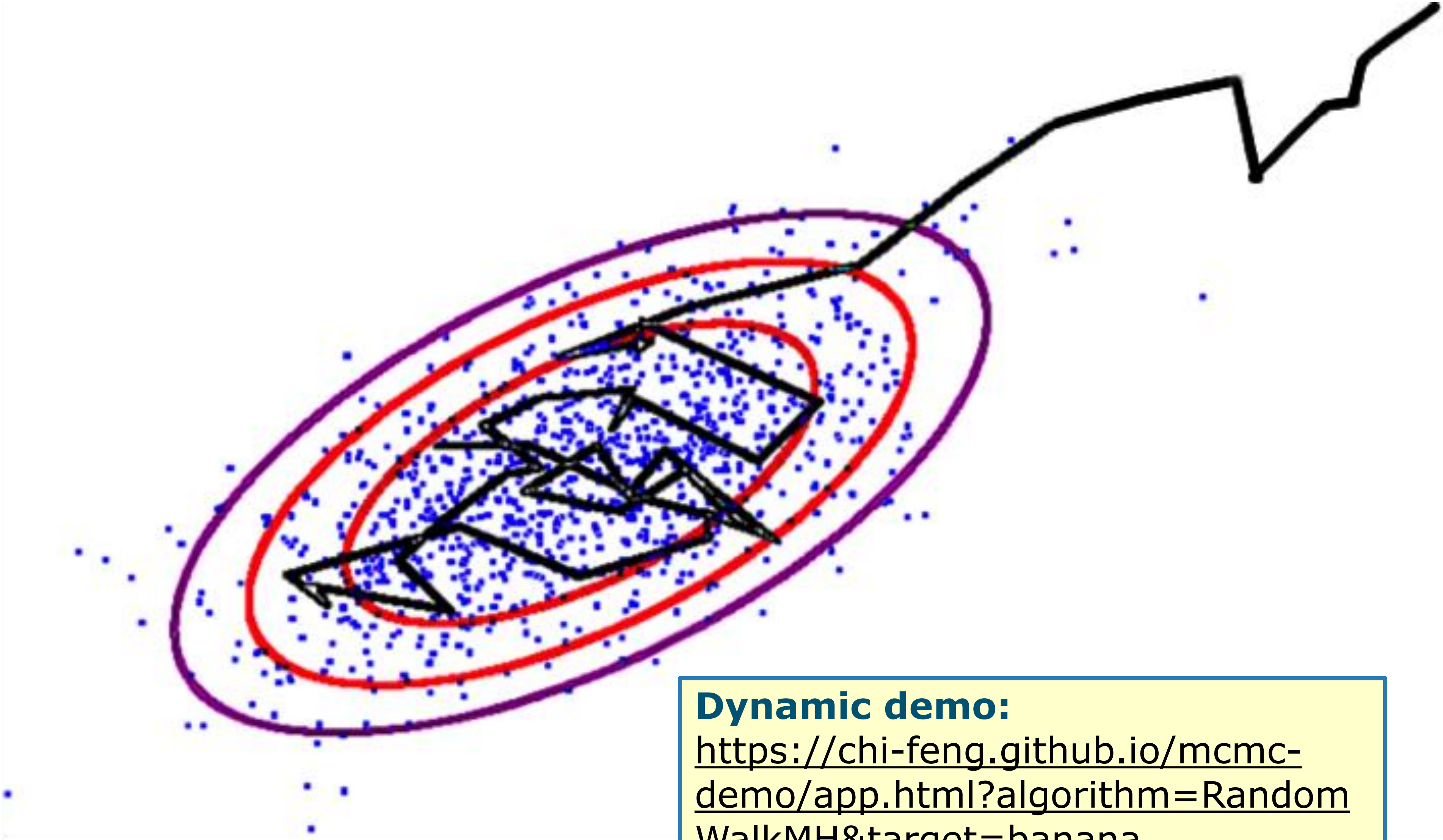
- Similar to Monte Carlo simulation: generate realisations by drawing from a probability distribution
- Difference is that in MCMC next realisation depends on the previous: it is a chain of realisations (see next slide)
- Apply burn-in and thinning to obtain independent realisations from the posterior distribution
- Cross-correlations between parameters also assessed
- If applied for linear regression with uninformative priors it will reproduce the analytical result
- But MCMC also works for parameter calibration (with uncertainty quantification) for models that have no analytical solution
- Disadvantage as before the computational load, even more so than ordinary Monte Carlo because typically tens of thousands of simulations needed
- Heavy computation of MCMC can be avoided using INLA: check literature folder (e.g. Gómez-Rubio et al., 2020)

Markov chain Monte Carlo algorithm

1. Start with an initial vector of model parameters θ_1
2. Let the parameter vector at step k be θ_k. Define $\theta' = \theta_k + \delta$ where δ is a random disturbance, simulated from a jumping distribution. The jumping distribution must be symmetric and centred around zero
3. Calulate $r = \dfrac{p(\theta')\cdot p(data|\theta')}{p(\theta_k)\cdot p(data|\theta_k)}$
4. If $r \geq 1$ accept θ' and set $\theta_{k+1} = \theta'$. If $r < 1$ accept θ' with probability r. If θ' rejected, set $\theta_{k+1} = \theta_k$
5. Increase k with one and go back to step 2
6. Stop when k reaches a pre-determined value (e.g. 10,000).

This is the Metropolis-Hastings algorithm, but there are many more, such as the Gibbs sampler. Some of these have much better convergence statistics, e.g. DREAM (see literature)

2D parameter space illustration of MCMC



Dynamic demo:
https://chi-feng.github.io/mcmc-demo/app.html?algorithm=RandomWalkMH&target=banana

Exercise 2

1. Open script 'mcmc.r' in RStudio and run it. This creates an MCMC sample from the bivariate normal distribution. How many runs are needed to obtain a stable result?
2. Check what happens if you use a different starting point.
3. Check what happens if you increase the jump size.
4. Check what happens if you decrease the jump size.
5. Include a correlation of 0.95 between the two variables and verify that MCMC can also reproduce this distribution.
6. Can you think of a distribution that would be difficult to reproduce?

In summary: incorporating model uncertainties

- Recall that we calculate output U from input A using model g, where A can consist of multiple inputs, uncertainty about A is defined by a probability distribution $p(A)$
- Let ξ represent the (vector of) model parameters and let σ^2 be the variance of the model structural error (i.e. assume additive noise), so that we can write $U = g(A, \xi, \sigma^2)$
- If we knew the (joint) probability distribution of A, ξ and σ^2 then uncertainty propagation was easy, but we only know $p(A)$
- $p(\xi, \sigma^2)$ is typically derived from the data (i.e. from the comparison of model outputs with independent observations, while taking input and observation error into account). Reason: model uncertainty is not universal but case-dependent
- We tackled this problem using a Bayesian calibration approach

The Bayesian framework

uncertainty of output given available information

because $U = g(A, \xi, \sigma^2)$ with known g

$$p(U|data) \Leftarrow p(A, \xi, \sigma^2|data) =$$

$$= \frac{p(A, \xi, \sigma^2)}{p(data)} \cdot p(data|A, \xi, \sigma^2) \propto p(A) \cdot p(\xi, \sigma^2) \cdot p(data|A, \xi, \sigma^2)$$

Bayes' law

input uncertainty, known

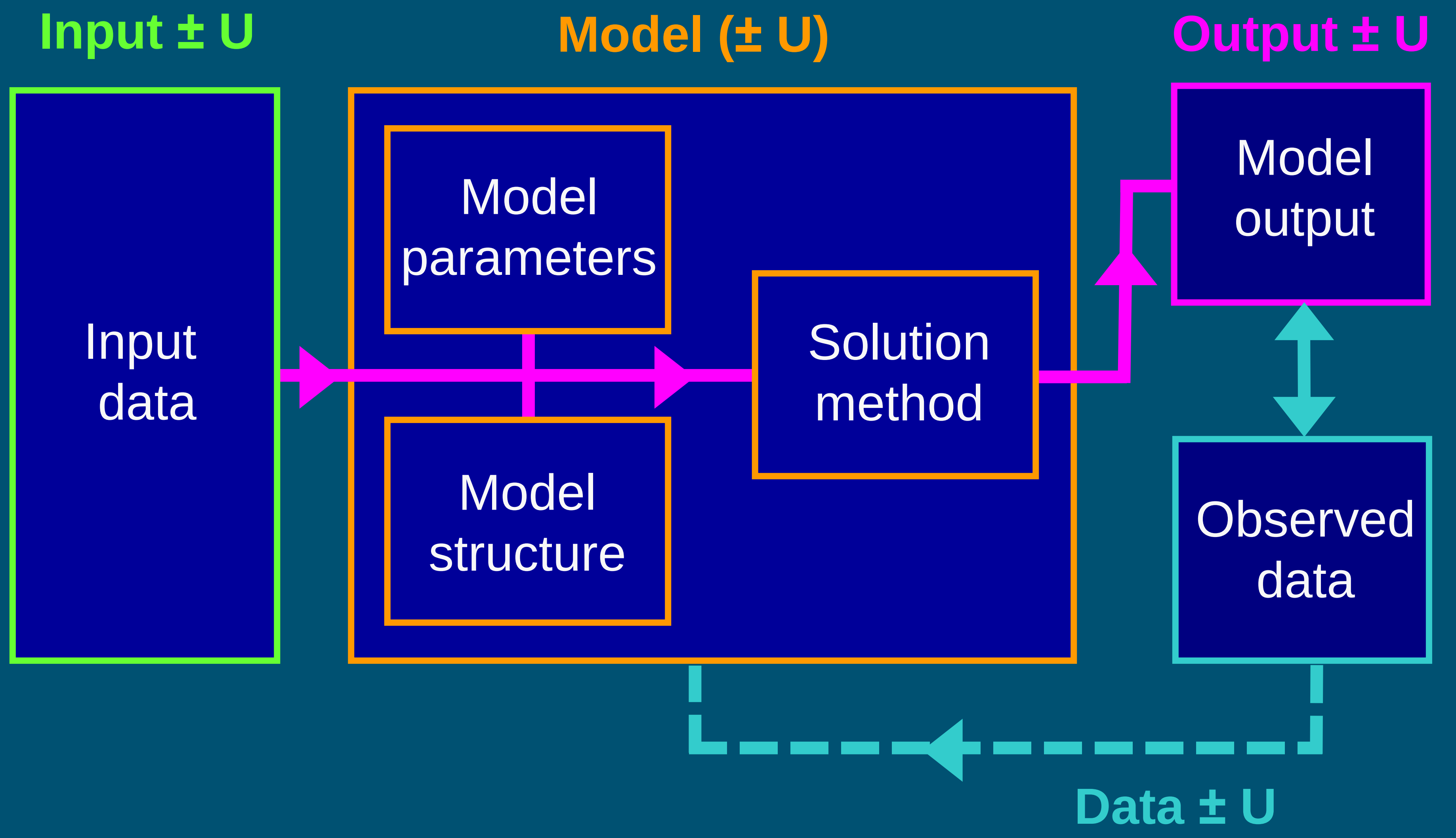
(flat) priors

easy, determined by g and observation error

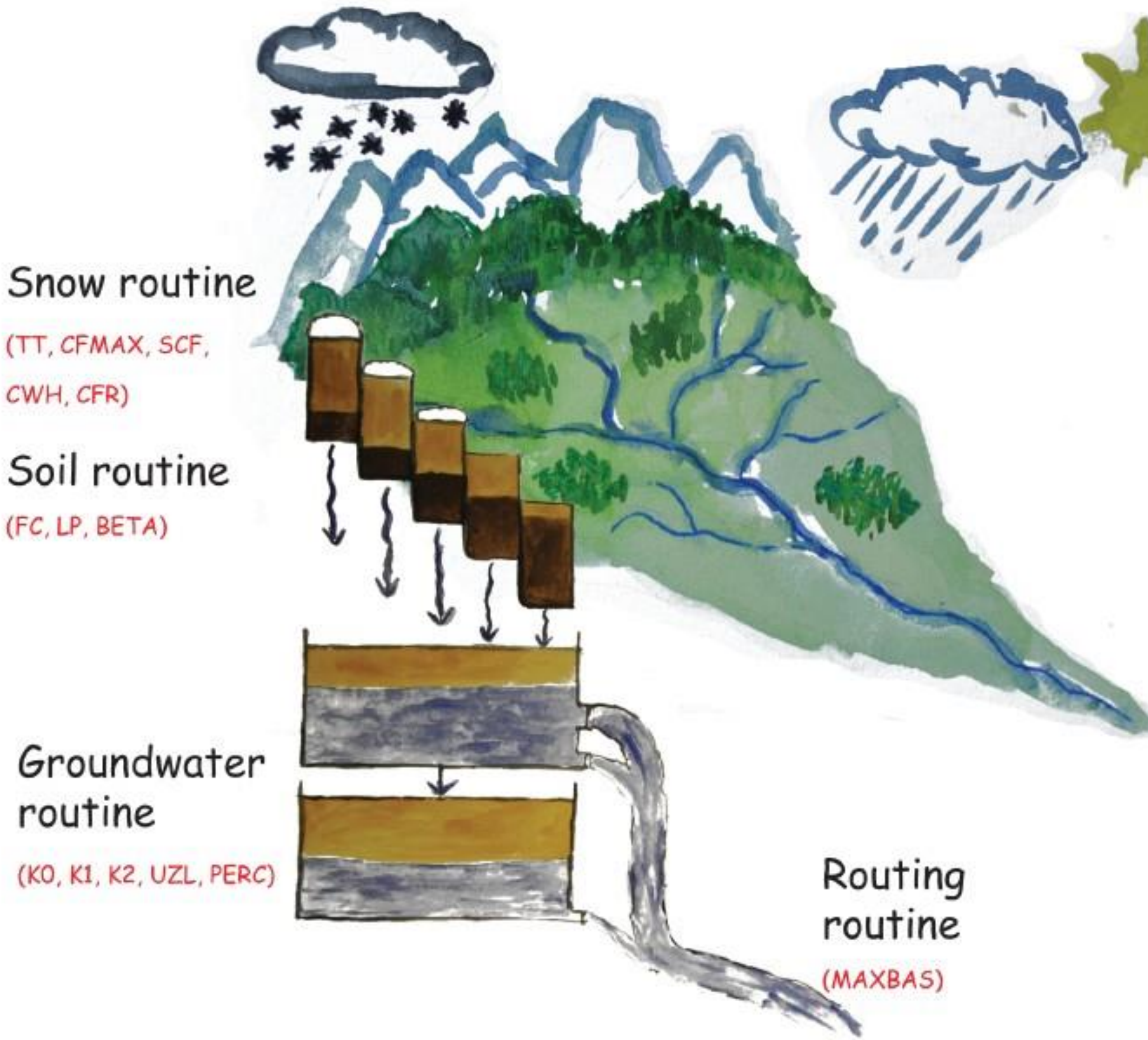
Propagation of all three uncertainty sources

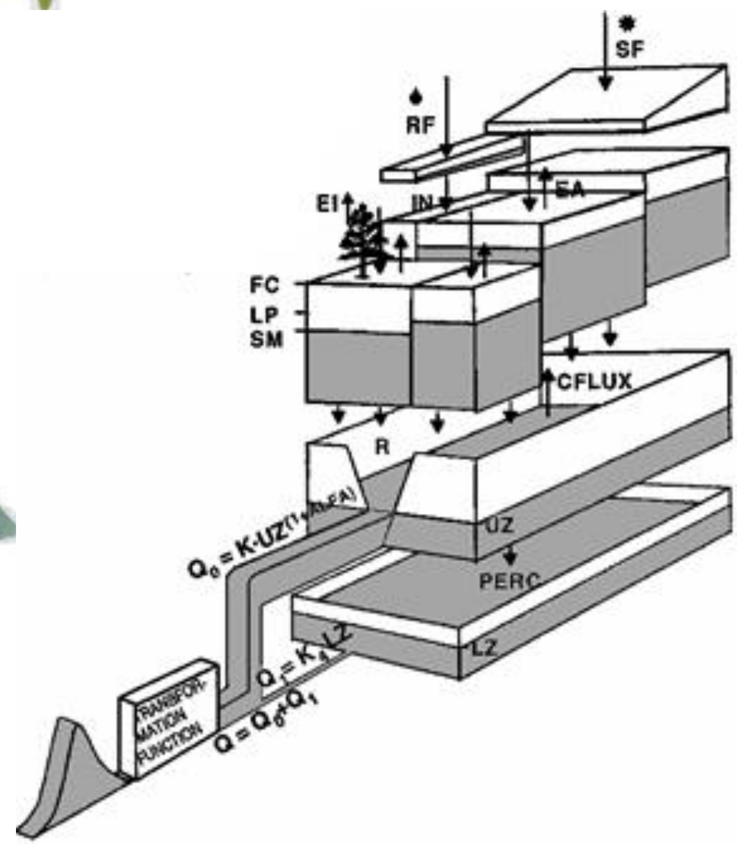
- The MCMC procedure generates a sample from the joint pdf of the model parameters ξ and from the variance σ^2 of the model structural uncertainty
- Calculation of this pdf was done such that it took into account that the discrepancy between model output and independent observations is partly caused by input uncertainty and observation uncertainty
- We can use this sample, together with a sample generated from the pdf of the input A, to analyse how all three uncertainty sources propagate through the model, using a straightforward Monte Carlo analysis
- This also allows us to compute the uncertainty contributions of each of the three sources

How nice: we can deal with all sources of uncertainty!



Introduction to the computer practical: Bayesian calibration of the TUWmodel (lumped rainfall-runoff model)





More detailed explanation of the model this afternoon

- The model has many parameters, we limit the Bayesian calibration to seven:
  - $lsuz$ = threshold storage of the upper reservoir before excess storage is reached
  - k_1 = how much of the excess storage of the upper reservoir reaches the outlet during a single time step (fast flow compared to k_2, slow compared to k_2)
  - $cperc$ = percolation rate from upper to lower reservoir
  - $croute$ = width of delay triangle used in the routing routine
  - *two* parameters that characterise model structural uncertainty and *one* parameter that characterises discharge observation uncertainty (see next slide)

Model structural uncertainty and discharge observation uncertainty

- Assume that these sources of uncertainty are multiplicative:

$$Y = H \cdot \boxed{e^{\varepsilon}} \cdot \boxed{e^{\eta}}$$

where Y is the measured discharge, H is the TUWmodel output and the means of e^{ε} and e^{η} are forced to one.

- Log-transformation gives:

$$\log(Y) = \log(H) + \varepsilon + \eta$$

where:

$$\varepsilon(t) = \beta_0 + \beta_1 \cdot \varepsilon(t-1) + \delta(t)$$

$$\delta(t) \sim N\left(0, \sigma_\delta^2\right)$$

$$\eta(t) \sim N(\mu_\eta, \sigma_\eta^2)$$

$$\beta_0 = -\frac{1}{2} \cdot \frac{1-\beta_1}{1-\beta_1^2} \cdot \sigma_\delta^2$$

$$\mu_\eta = -\frac{1}{2}\sigma_\eta^2$$

For more details see Wadoux et al. (2020) in the literature folder