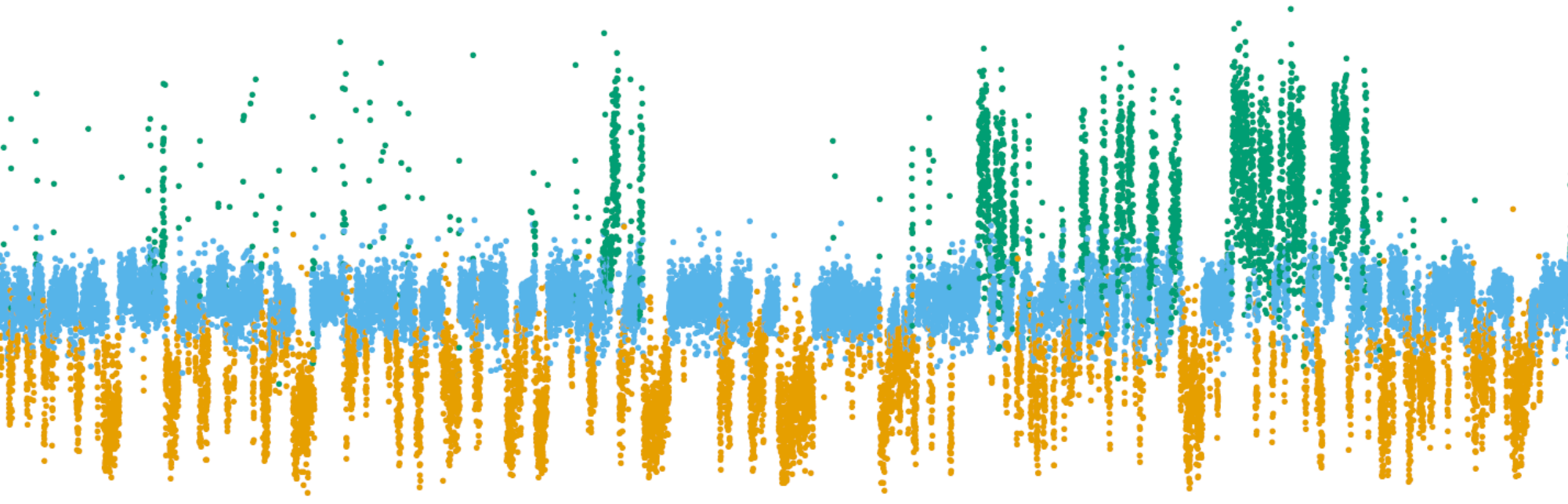
Hidden Markov Models (HMMs)

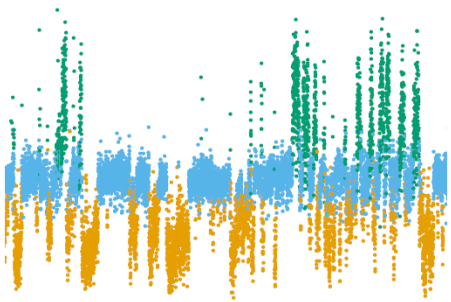
CUSO Summer School 2024

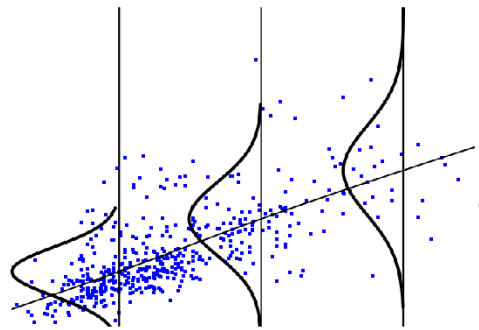
Roland Langrock, Bielefeld University

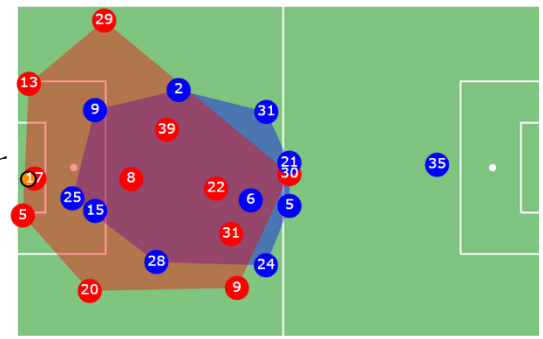


Who am I? (brief academic CV)

- born 1983 in Hannover, Germany
- studied mathematics 2003–2008 in Heidelberg
- PhD with focus on statistics 2008-2011 in Göttingen
- postdoc/lecturer 2011-2015 in St Andrews (statistics & biology)
- since 2015 professor of statistics and data analysis at Bielefeld University







Plan for the course

Part 1: What is an HMM? (33 slides)

Part 2: How do we fit an HMM to data? (46 slides)

Part 3: What else can we do with HMMs? (89 slides)

Part 1 — What is an HMM?

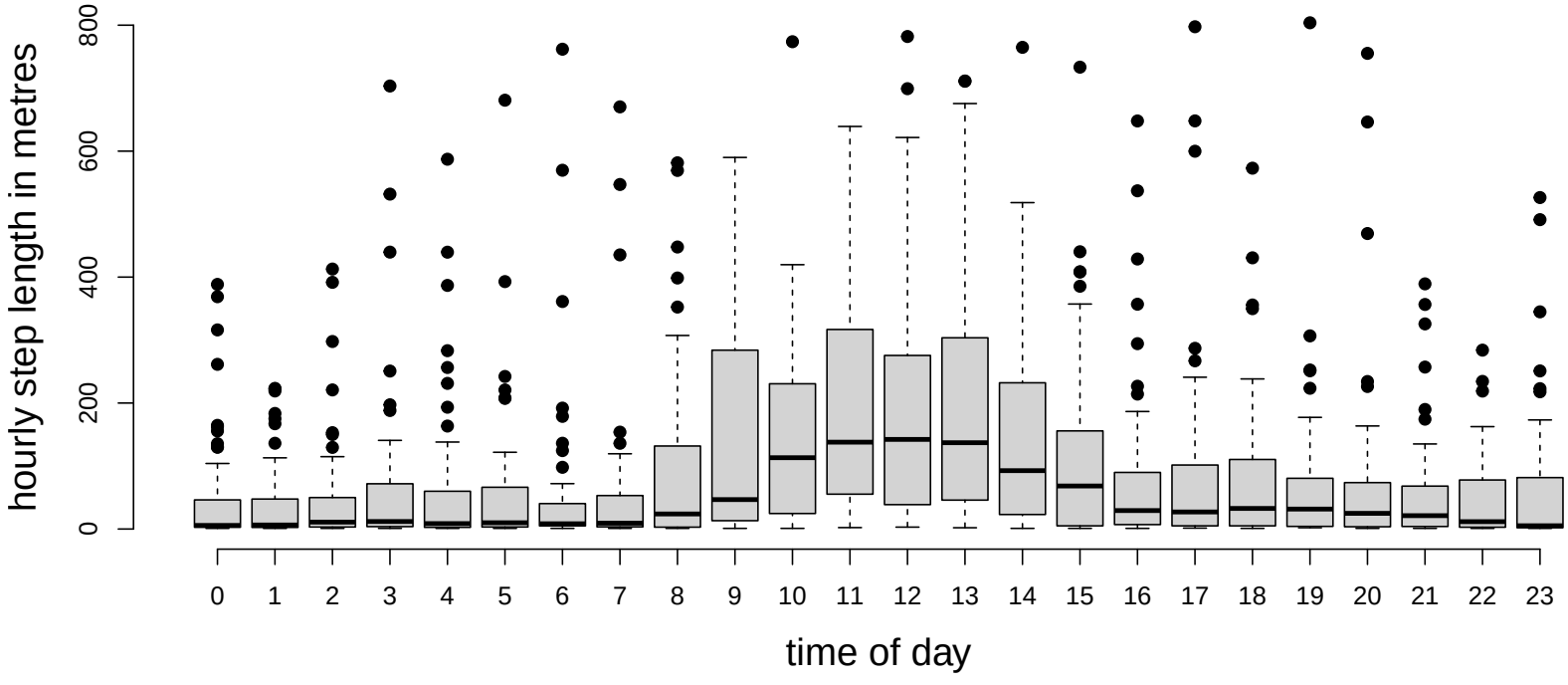
1.1 Motivating example


Greenland
Iceland
Norwegian Sea
Norway
North Sea
United Kingdom
Ireland
London
Netherlands
Germany
Belgium
Paris
Denmark
France
Austria

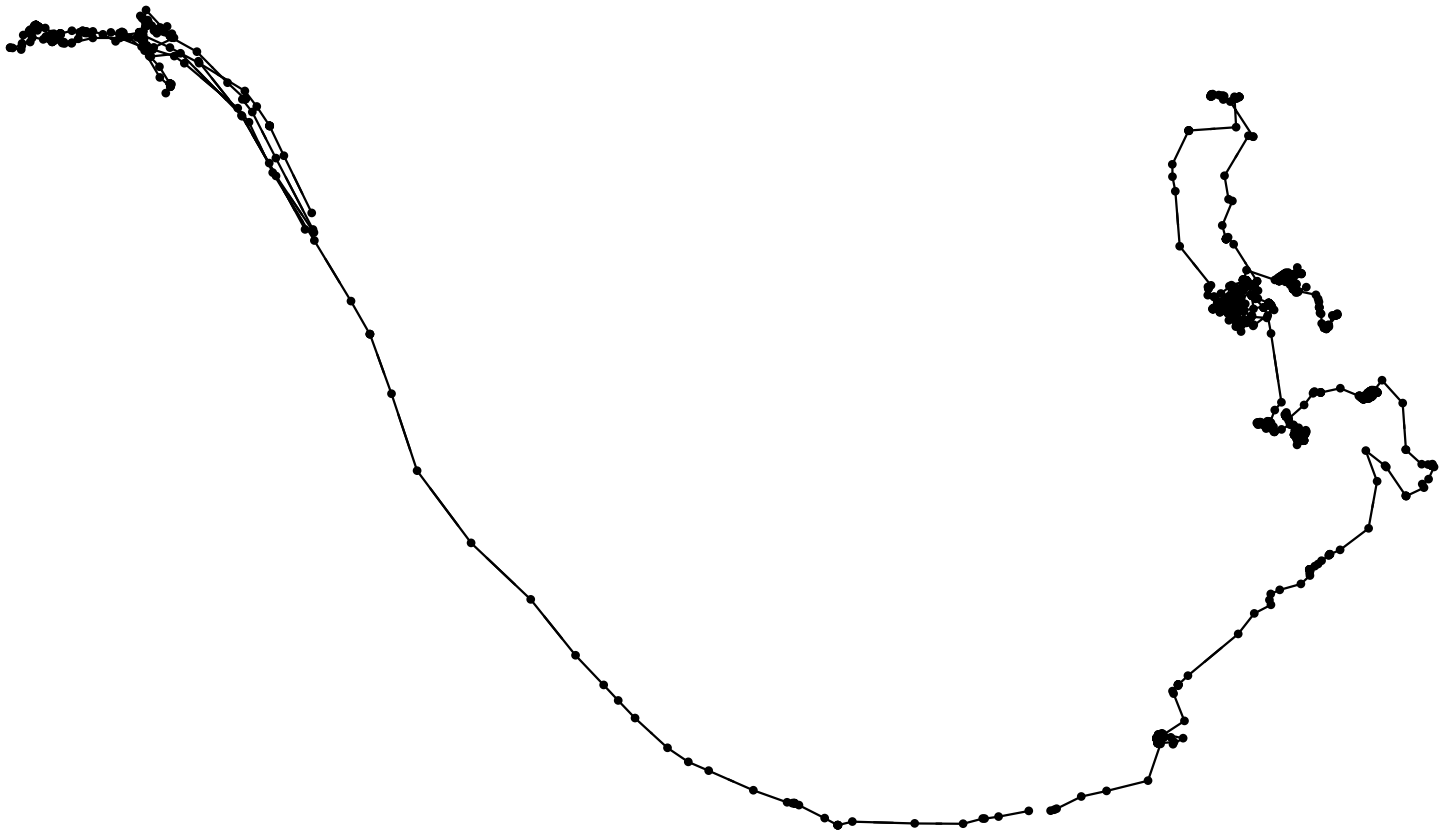

Photo:
Lars Holst Hansen/ARC-PIC.com

Why would we want to statistically model muskox movement?

- do individual characteristics like age, size & sex affect movement activity?
- what about external covariates like temperature, snow cover, etc.?
- how does the behaviour vary over the day? (see EDA below)



What are the main patterns in the movement data?



Deriving key movement metrics

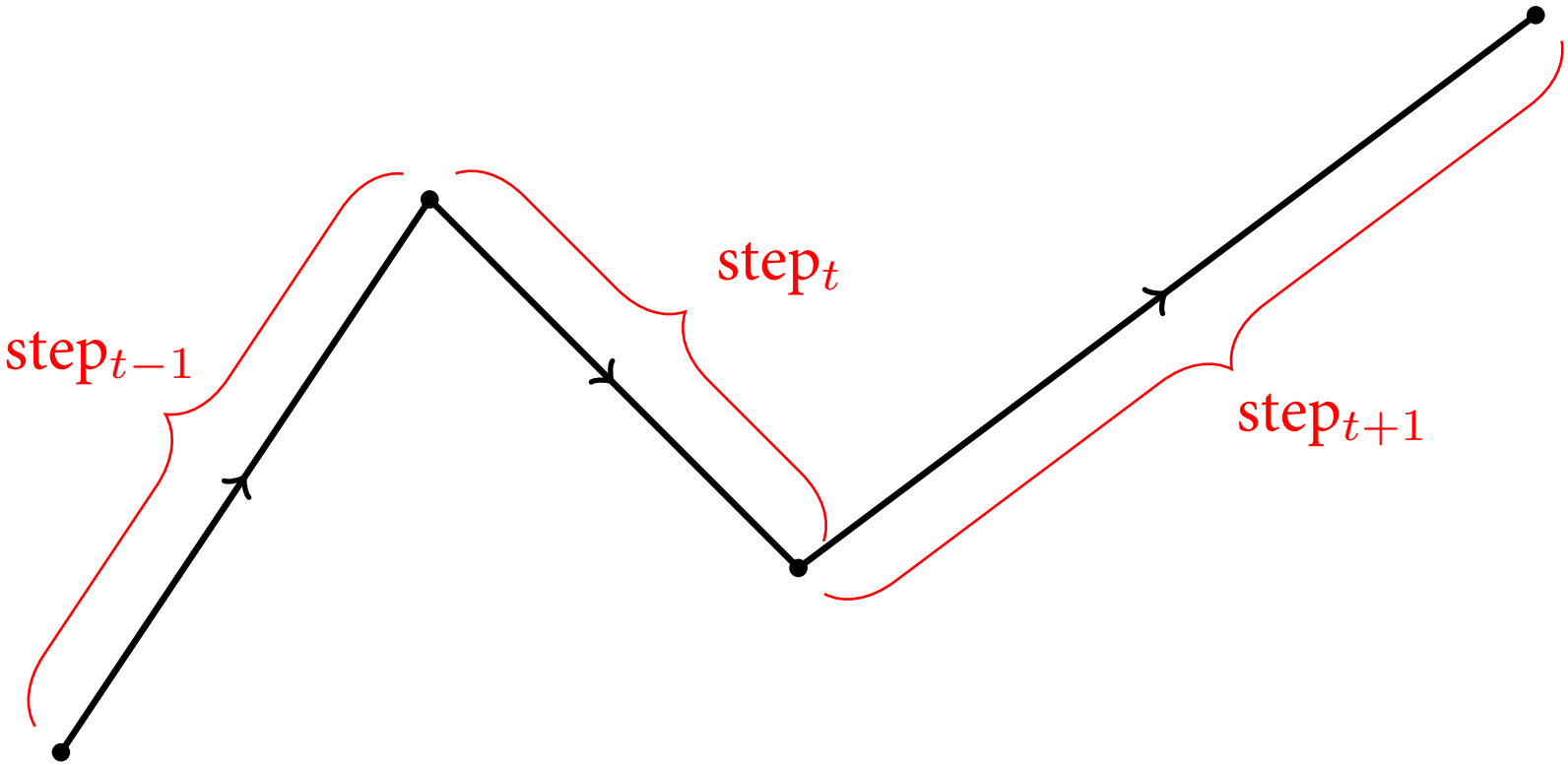


Figure: Calculating (Euclidean) step lengths between successive locations — later on, we will additionally consider the turning angles.

Load the data in R:

```
muskox <- read.csv("http://www.rolandlangrock.com//Misc//muskox.csv")
```

Calculate step lengths (and turning angles) using the `moveHMM` package:

```
install.packages("moveHMM") # if not already installed
library(moveHMM)
data <- prepData(muskox, type = "UTM")
```

Let's have a look at the first few rows:

```
> head(data, 12)
       ID      step        angle      x       y temp  altitude tod
1  track1 2046.5034           NA 517945 8262572   -7  2.364444  12
2  track1 2005.0249 -0.152212718 519221 8260972   -6  5.244889  13
3  track1 1988.6609  0.074810792 520219 8259233   -7 20.193333  14
4  track1        NA           NA 521335 8257587   -6 13.878333  15
5  track1        NA           NA     NA      NA   -6 13.878333  16
6  track1        NA           NA     NA      NA   -6 13.878333  17
7  track1        NA           NA     NA      NA   -6 13.878333  18
8  track1        NA           NA     NA      NA   -6 13.878333  19
9  track1        NA           NA     NA      NA   -6 13.878333  20
10 track1        NA           NA     NA      NA   -6 13.878333  21
11 track1 1860.8732           NA 521264 8258161   -5 15.695111  22
12 track1  967.3826 -0.006769389 520439 8259829   -2 35.833334  23
```

And you might want to try:

```
plot(data)
```

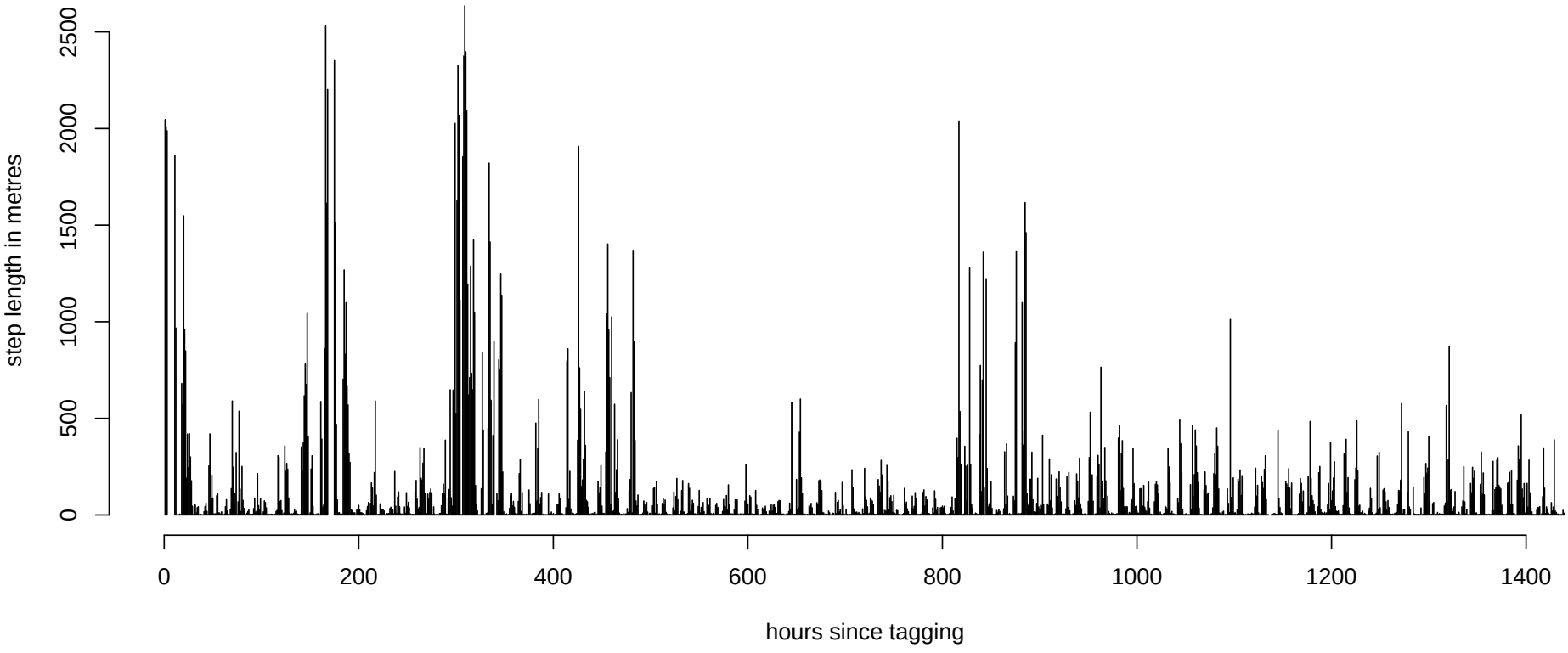


Figure: Time series of step lengths $x_1, \ldots, x_{1440}$ between successive locations.

Towards a statistical model for the muskox movement data

Exploratory data analysis reveals **three levels of (movement) activity**:

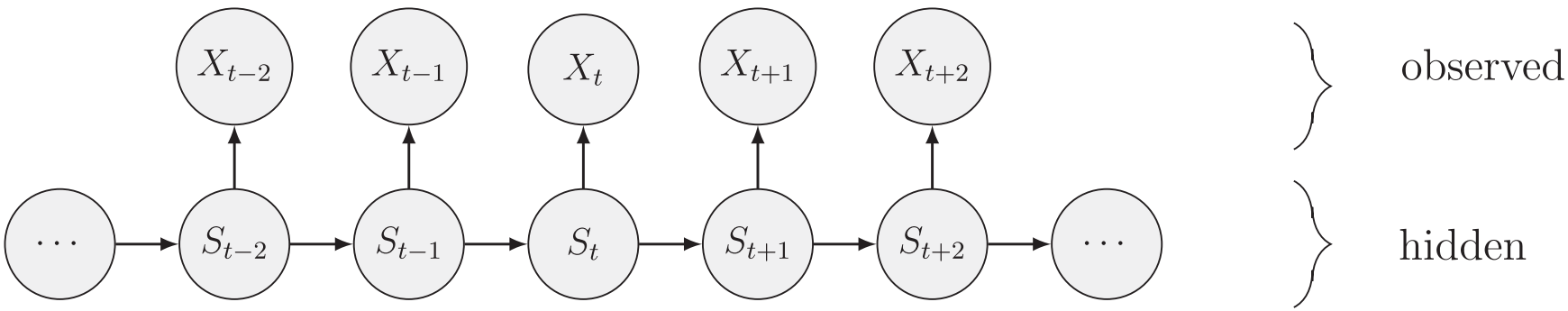
- none ($\rightsquigarrow$ resting)
- moderate ($\rightsquigarrow$ foraging/area-restricted search)
- high ($\rightsquigarrow$ travelling)

These different activity levels **occur in clusters**: when the animal is say travelling, then it tends to exhibit the same behaviour in subsequent time periods.

HMMs — sneak preview

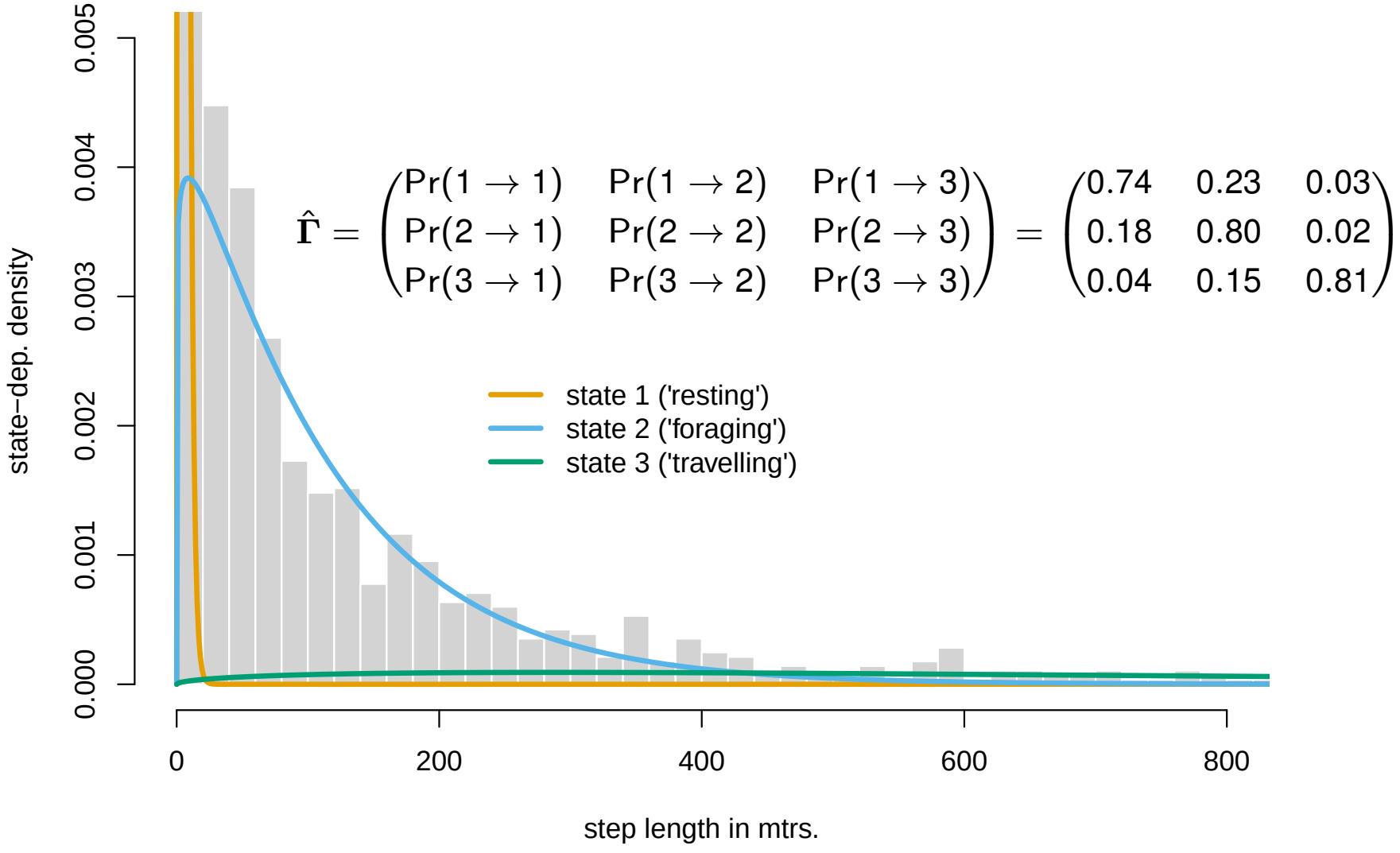
A **hidden Markov model (HMM)** involves two stochastic processes:

1. **observation process** (here: the time series of step lengths)
2. underlying hidden **state process** (here: the behavioural state)



- ⇝ each observation is generated by one of N possible distributions
- ⇝ the state process selects which of the N distributions is active at any time
- ⇝ the state at time t depends on the state at time $t-1$ (⇝ Markov chain)

Muskox step lengths — how the fitted HMM will look like



OK, fine, but what do people use HMMs for?

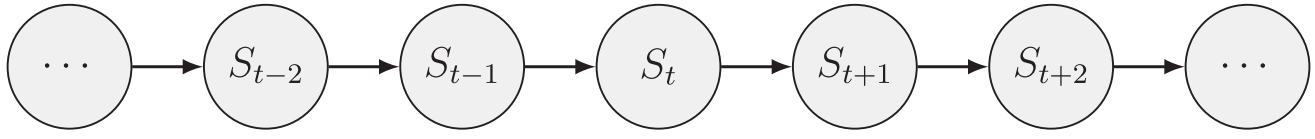
- decoding hidden states (medicine, recognition tasks, etc.)
- forecasting (mainly in econ/finance)
- to better understand the dynamics of a system (very common in ecology)

1.2 Definition of the basic HMM

A **Markov chain** is a sequence of random variables $S_1, S_2, \ldots$ such that

- $S_t \in \{1, \ldots, N\}$ for all t (i.e. there are N so-called "states")
- the **Markov property** holds:

$$\Pr(S_{t+1} = s_{t+1} \mid S_t = s_t, \ldots, S_1 = s_1) = \Pr(S_{t+1} = s_{t+1} \mid S_t = s_t)$$



The Markov property simply means that the state at time t completely determines the probabilities of the different states at time $t + 1$.

This dependence structure is *mathematically convenient* and often plausible.

Due to the Markov property, a Markov chain is fully characterised by

(i) the **initial state distribution**,

$$\boldsymbol{\delta}^{(1)} = \left(\delta_1^{(1)}, \ldots, \delta_N^{(1)}\right) = \left(\Pr(S_1 = 1), \ldots, \Pr(S_1 = N)\right),$$

(ii) and the (one-step) **state transition probabilities**[1],

$$\gamma_{ij} = \Pr(S_{t+1} = j \mid S_t = i),$$

which we summarise in the **transition probability matrix** (t.p.m.):

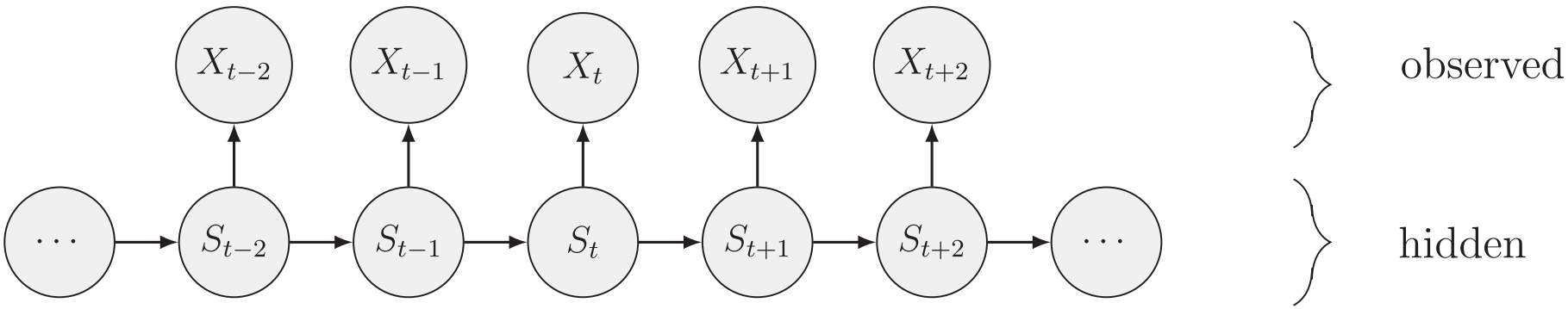
$$\boldsymbol{\Gamma} = \begin{pmatrix} \gamma_{11} & \cdots & \gamma_{1N} \\ \vdots & \ddots & \vdots \\ \gamma_{N1} & \cdots & \gamma_{NN} \end{pmatrix} \overset{\text{e.g.}}{=} \begin{pmatrix} 0.74 & 0.23 & 0.03 \\ 0.18 & 0.80 & 0.02 \\ 0.04 & 0.15 & 0.81 \end{pmatrix}$$

[1] here for simplicity assumed to be constant over time — this will later be relaxed

HMM formulation — some preliminary remarks

- the class of HMMs is immensely flexible and versatile
- in particular,
  - various different types of data can be considered — count data, continuous data, binary data, categorical data, univariate data, multivariate data, ... (you name it!)
  - various different dependence structures can be considered
- we start with the most basic formulation, later building up complexity
- main inferential tools are the same regardless of the specific formulation

Intuitive definition of the basic HMM



- $S_1, S_2, \ldots$ is an N–state Markov chain
- S_t selects which of N distributions is active at time t
- X_t is then generated by that distribution

More formal definition of the basic HMM

An N–state HMM is a doubly stochastic process in discrete time, with

- an unobserved **state process** $S_1, S_2, \ldots, S_T$ taking values in $\{1, \ldots, N\}$,
- and an observed **state-dependent process** $X_1, X_2, \ldots, X_T$[2],

such that

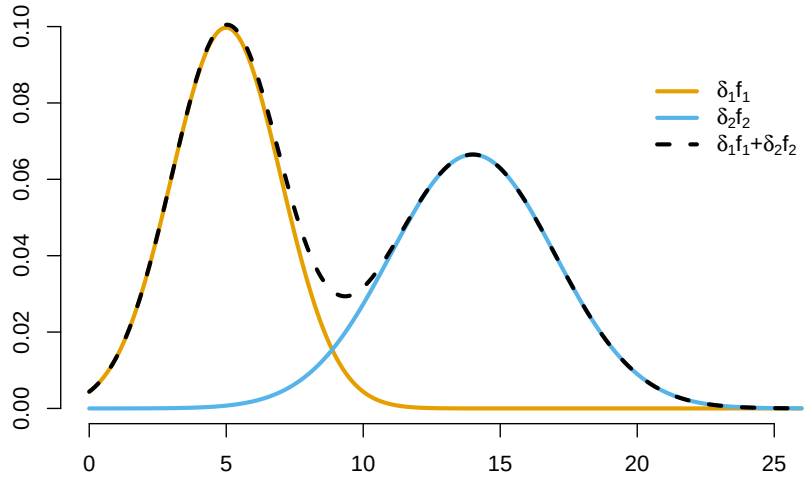
- $f(s_t \mid s_1, \ldots, s_{t-1}) = f(s_t \mid s_{t-1})$
  (Markov property)
- $f(x_t \mid s_1, \ldots, s_T, x_1, \ldots, x_{t-1}, x_{t+1}, \ldots, x_T) = f(x_t \mid s_t)$
  (conditional independence assumption)

Note this is a **general model for time series data**, which is useful when

- we are interested in how some process evolves over time...
- ...but we don't directly observe that process (instead just a proxy)

[2] where the X_t can also be vectors

Marginal distribution



If $\{S_t\}$ is stationary with stationary distribution $(\delta_1, \ldots, \delta_N)$, then the unconditional (marginal) distribution of X_t is

$$f(x_t) \overset{\text{tot. prob.}}{=} \sum_{j=1}^{N} \Pr(S_t = j) f_j(x_t) \overset{\text{stat.}}{=} \sum_{j=1}^{N} \delta_j f_j(x_t).$$

- thus, an HMM is a mixture model!
- crucially, an HMM is a **dependent mixture model**: which distribution is selected at time t does affect which one will be selected at time $t + 1$

A basic HMM is specified by

- the **initial state distribution** $\boldsymbol{\delta}^{(1)}$,
- the **transition probability matrix** $\boldsymbol{\Gamma}$, and
- the **state-dependent (component) distributions**, $f_j(x_t) = f(x_t \mid s_t = j)$

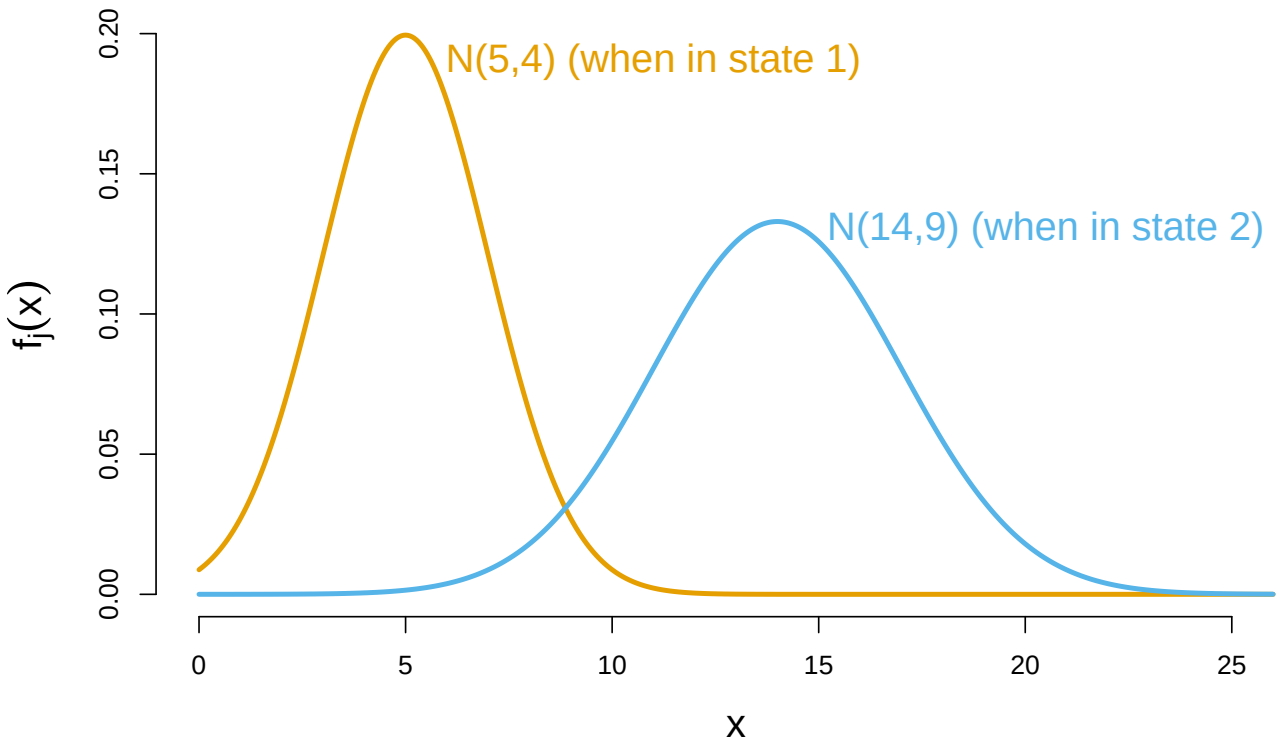
In particular, given these three components we can

- simulate data from the HMM
- calculate the likelihood $f(x_1, \ldots, x_T)$

1.3 Simulating data from an HMM

A concrete simulation example

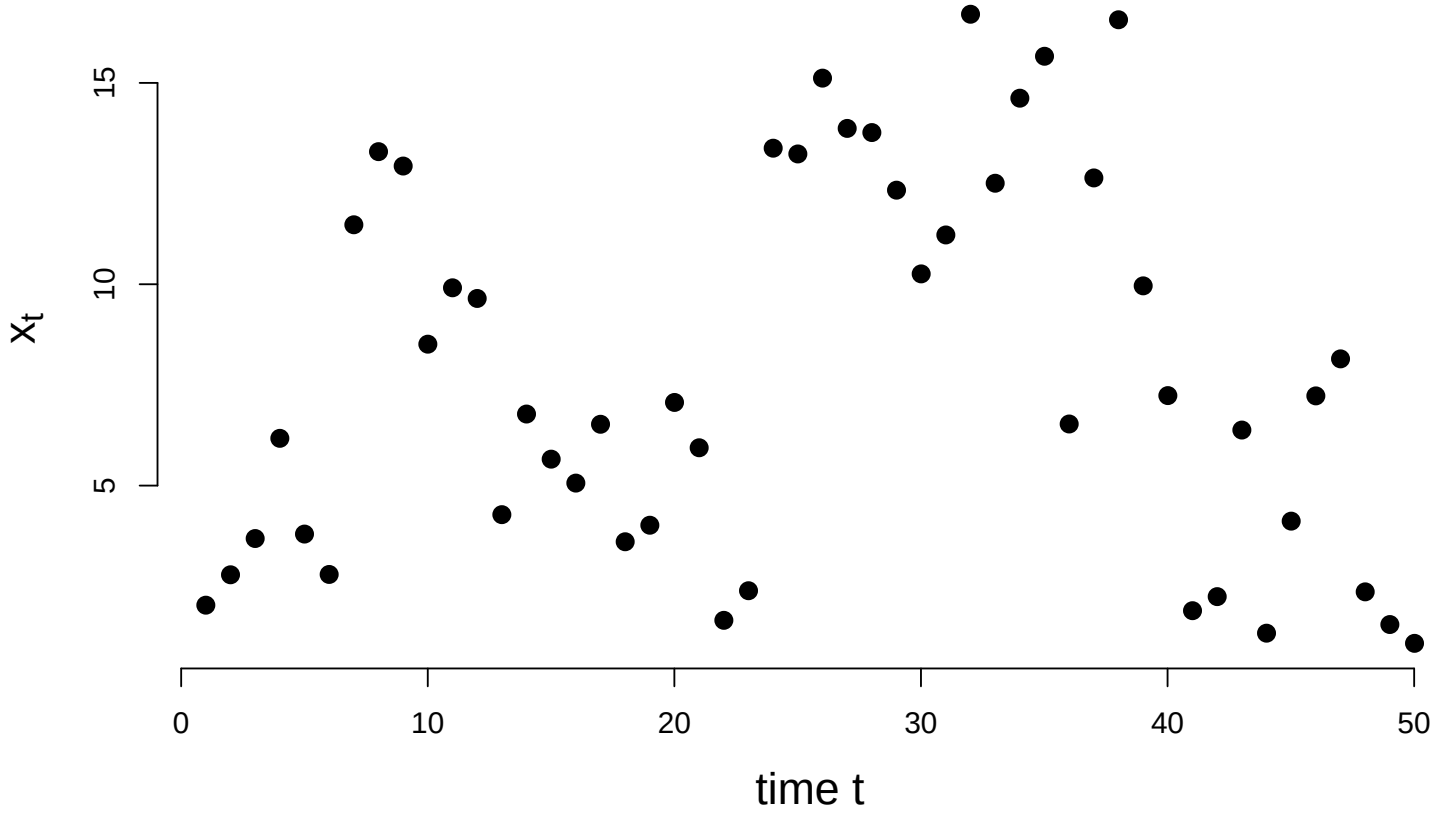
$$\boldsymbol{\delta}^{(1)} = (0.5, 0.5) \qquad \boldsymbol{\Gamma} = \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}$$



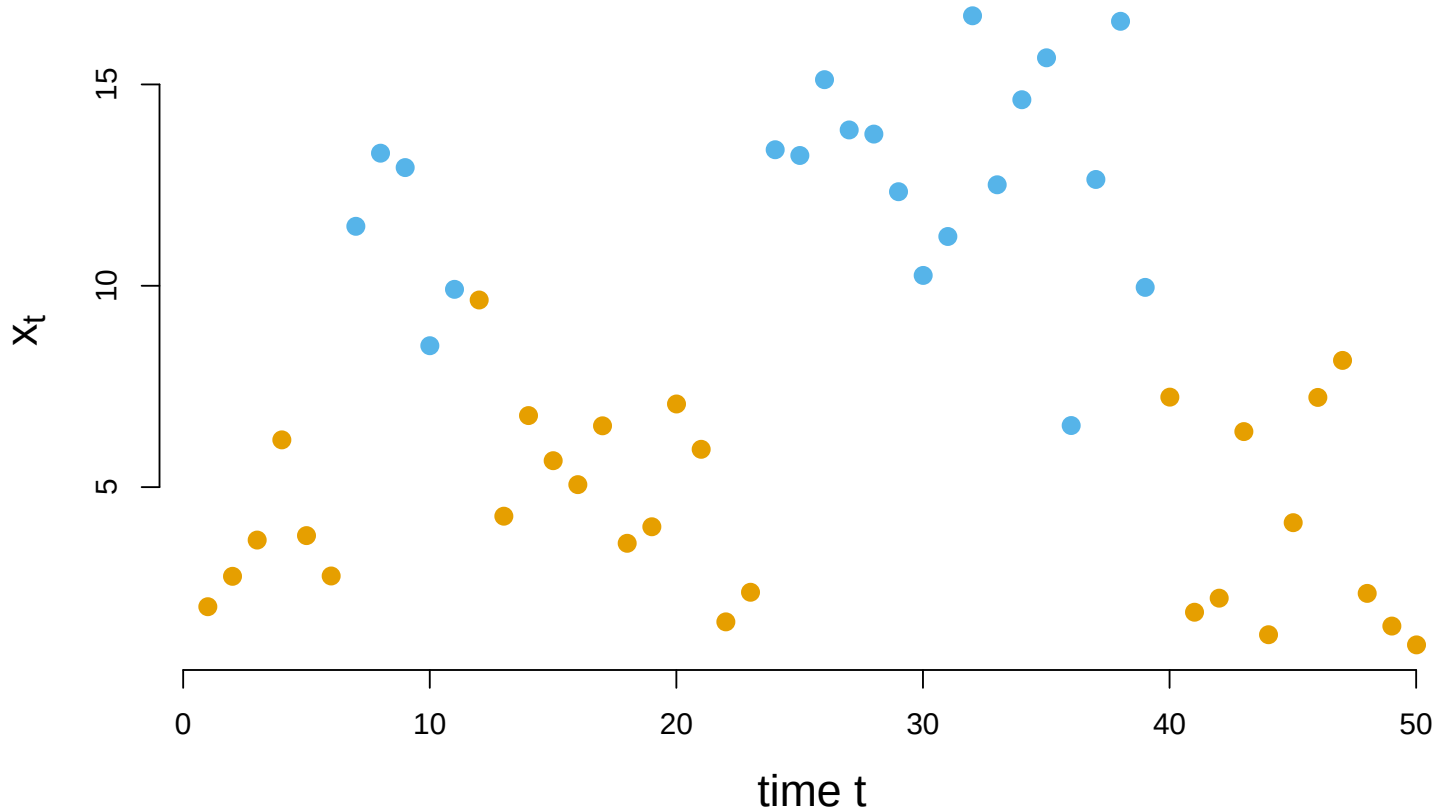
R code for simulating data from this HMM

```
n <- 50
x <- s <- rep(NA, n)
Gamma <- matrix(c(0.9, 0.1, 0.1, 0.9), nrow = 2)
delta <- c(0.5, 0.5)
mu <- c(5, 14)
sigma <- c(2, 3)

s[1] <- sample(1:2, size = 1, prob = delta)
x[1] <- rnorm(1, mu[s[1]], sigma[s[1]])

for (t in 2:50){
  s[t] <- sample(1:2, size = 1, prob = Gamma[s[t-1], ])
  x[t] <- rnorm(1, mu[s[t]], sigma[s[t]])
}
```

One example realisation



Can you guess the states?



These are the actual states (which in practice aren't observed).

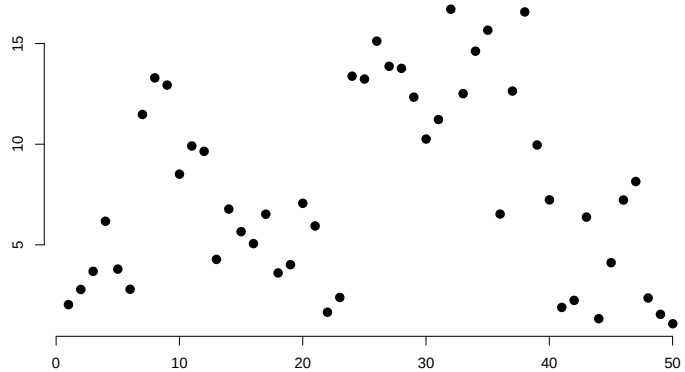
Some remarks on the hidden states

The example from the previous slide shows:

- the obs. will often give a good idea of what could be the underlying state
- however, when the component distributions overlap, we can never be sure
- it's crucial to take the time series nature of the observations into account

The time series $X_1, \ldots, X_T$ is a noisy observation of the state process $S_1, \ldots, S_T$, which we can use to learn something about the latter.

Cond. independence $\neq$ independence!



While the observations are **conditionally independent of each other** (given the states), they are **not independent of each other**:

- say that, in the simulation example, x_t is large (relative to the overall mean)
- then most likely this is because state 2 is active at time t
- with a prob. of 90% the process will then still be in state 2 at time $t + 1$
- and hence x_{t+1} will probably also be large

Thus, the Markov chain **induces dependence** in the state-dependent process — only *within* states the observations are independent of each other.

1.4 Some remarks

Areas of application of HMMs

Observations	Interpretation of states	Literature
Fourier transforms of recorded speech	sequence of phonemes	Juang and Rabiner (1991)
EEG measurements during sleep	REM and non-REM sleep states	Langrock *et al.* (2013)
multiple sclerosis lesion count data	disease states	Altman and Petkau (2005)
times between a neuron's firing events	states of neuron	Camproux *et al.* (1996)
DNA sequence of bases	homogeneous segments of DNA sequence	Churchill (1989)
share returns	market sentiment	Rydén *et al.* (1998)
retailer transaction records	customer's propensity to buy	Mark *et al.* (2013)
volcanic eruptions	activity level of the volcano	Bebbington (2007)
responses in a learning experiment	guessing state vs. learned state	Visser *et al.* (2002)

Terminology in the literature

Various other labels are used for what I call HMMs, e.g.

- Markov-switching/regime-switching models (especially in econometrics)
- state-switching models (e.g. in ecology)
- latent Markov models (especially in Italy!)
- latent transition/latent class models (especially in the social sciences)
- hidden Markov processes (especially in engineering)
- state-space models (in mathematical statistics)

These labels very often refer to the same mathematical object...

Rule-of-thumb: model with observations driven by underlying states & Markov assumption for states? $\rightsquigarrow$ most likely it's an HMM as discussed here.

More on HMMs in the literature (classification vs. general inference)

Roughly speaking, there are two branches of literature on HMMs:

1. **engineering/ML** literature, dealing with recognition/classification:
   - originally developed for speech recognition in the 1960s, HMMs have been applied in all sorts of recognition tasks
   - in those instances, training data, where the states are observed, are used to calibrate the model (supervised learning)
   - recognition/classification then involves decoding the latent states for new data

2. **statistical** literature, dealing with general inference:
   - here the main aim often is to learn something about the system considered
   - sometimes used also for forecasting, especially in economics/finance

Our focus is on 2., but we'll cover the main techniques used within 1. too.

Interpretation of HMM states in case of unsupervised learning

A trivial yet underappreciated HMM fact:

HMM states $\neq$ meaningful entities (e.g. behaviours)

Instead, the model states are only proxies for potentially meaningful entities.

Why?

- features of HMM states are (usually) data-driven ($\rightsquigarrow$ unsupervised learning)
- model picks up whatever pattern is strongest w.r.t. multimodality
- temporal resolution determines which states may be inferred at all

Part 2 — How do we fit an HMM to data?

2.1 Overview: how to fit an HMM to data

Workflow for finding “the right” HMM for given real data

1. formulate **candidate models** based on EDA, in particular
   - selecting N & the class of state-dependent distributions
   - potentially incorporating covariates (see later slides)

   ↓ ↓ ↓
2. **estimate** model parameters, for each candidate model

   ↓ ↓ ↓
3. **choose** between candidate models

   ↓ ↓ ↓
4. **check** the chosen model (and, if necessary, go back to 1.)

   ↓ ↓ ↓
5. conduct whatever **inference** is of interest

We will first focus on 2., which is usually the hardest part.

Parameter estimation — overview

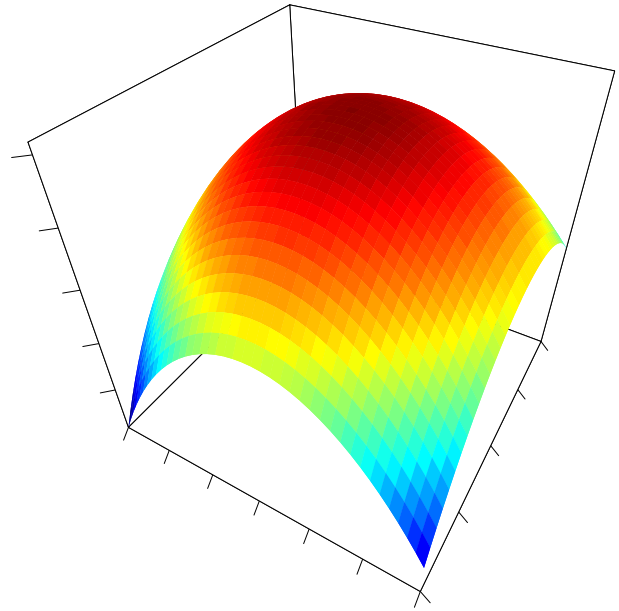
There are three main approaches to fitting HMMs:

1. direct numerical likelihood maximisation
2. likelihood maximisation using the EM algorithm
3. Bayesian inference/MCMC

While there are scenarios where either 2. or 3. can be preferable[3], in most cases my preferred option is 1., as it's simplest and often also fastest.

[3] in particular, Bayesian inference has some advantages when the model includes random effects

Maximum likelihood estimation



- maximum likelihood (ML) estimation is an approach for fitting a model to data
- key idea: good parameter estimates make the observed data look plausible
- ML estimation: select parameter $\boldsymbol{\theta}$ for which model has the highest likelihood

$$\mathcal{L}(\boldsymbol{\theta}) = f_{\boldsymbol{\theta}}(x_1, \ldots, x_T)$$

  of having generated the observed data

- ML estimation...
  - ...is intuitively appealing,
  - ...is practically feasible in many cases (including HMMs),
  - ...and has desirable theoretical properties
- but we do need to be able to calculate the likelihood!

2.2 Likelihood evaluation & optimisation

Likelihood evaluation — how?

The seemingly easiest way to calculate the likelihood is via summation over all possible state sequences (law of total probability):

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{s_1=1}^{N} \cdots \sum_{s_T=1}^{N} \left(\prod_{t=1}^{T} f_{s_t}(x_t) \right) \left(\delta_{s_1}^{(1)} \prod_{t=2}^{T} \gamma_{s_{t-1},s_t} \right)$$

- simple structure, and all components are directly available
- but N^T summands render the calculation infeasible in most cases
- note though that many calculations in the sum above are redundant
- for example, for $T = 20$ and $N = 2$, it is extremely inefficient to separately calculate the likelihood contribution of the two state sequences

1 1 1 2 2 2 2 1 1 1 1 1 1 2 2 2 2 1 1 1

and 1 1 1 2 2 2 2 1 1 1 1 1 1 2 2 2 2 1 1 2

Smarter way to do this: the forward algorithm

Consider the so-called **forward variables**,

$$\alpha_t(j) = f(x_1, \ldots, x_t, s_t = j), \quad \boldsymbol{\alpha}_t = \big(\alpha_t(1), \ldots, \alpha_t(N)\big)$$

At time t, these variables contain information on

- the likelihood of the observations up to time t
- the probabilities of being in the different states at time t

The **forward algorithm** is an efficient recursive scheme for calculating the forward variables:

$$\boldsymbol{\alpha}_1 = \boldsymbol{\delta}^{(1)}\mathbf{P}(x_1), \qquad \boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-1}\boldsymbol{\Gamma}\mathbf{P}(x_t) \text{ for } t = 2, \ldots, T,$$

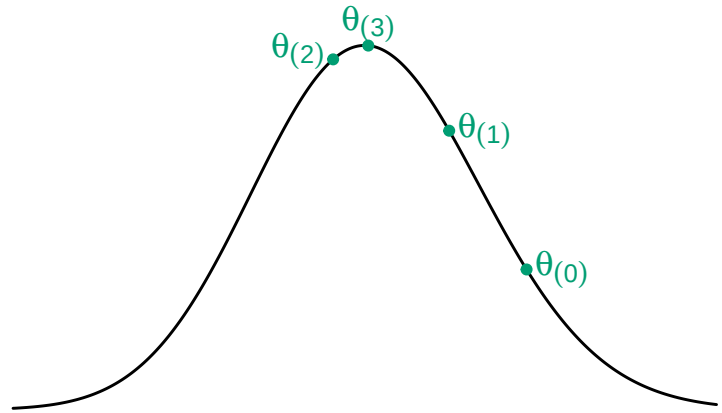
with $\mathbf{P}(x_t) = \text{diag}\big(f_1(x_t), \ldots, f_N(x_t)\big)$, initial distribution $\boldsymbol{\delta}^{(1)}$ and t.p.m. $\boldsymbol{\Gamma}$.

Using the forward algorithm to evaluate the likelihood

- the forward algorithm can be applied in order to first calculate $\boldsymbol{\alpha}_1$, then $\boldsymbol{\alpha}_2$ based on $\boldsymbol{\alpha}_1$, then $\boldsymbol{\alpha}_3$ based on $\boldsymbol{\alpha}_2$, etc., until one arrives at $\boldsymbol{\alpha}_T$
- the likelihood is then $\mathcal{L}(\boldsymbol{\theta}) = \sum_{j=1}^{N} f(x_1, \ldots, x_T, s_T = j) = \sum_{j=1}^{N} \alpha_T(j)$
- written in closed form, with $\mathbf{1} = (1, \ldots, 1) \in \mathbb{R}^N$:

$$\mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{\delta}^{(1)} \mathbf{P}(x_1) \boldsymbol{\Gamma} \mathbf{P}(x_2) \boldsymbol{\Gamma} \mathbf{P}(x_3) \ldots \boldsymbol{\Gamma} \mathbf{P}(x_T) \mathbf{1}^t$$

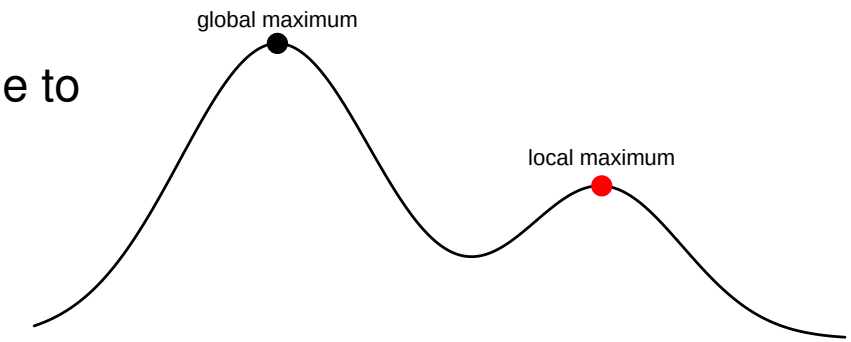
- comput. cost of evaluating $\mathcal{L}(\boldsymbol{\theta})$ is **linear** in the number of observations, T
- in practice, this means that the likelihood can be evaluated *in a fraction of a second* even for T in the thousands and moderate number of states N
- this opens up the way for a numerical maximisation of the likelihood



Numerical search algorithms are used to find the maximum likelihood estimate:

- guess the value of the parameter vector as $\boldsymbol{\theta}_{(0)}$ (initial value)
- obtain improved guess $\boldsymbol{\theta}_{(1)}$ based on (gradient in) $\boldsymbol{\theta}_{(0)}$
- obtain improved guess $\boldsymbol{\theta}_{(2)}$ based on (gradient in) $\boldsymbol{\theta}_{(1)}$
- ...
- terminate algorithm when changes in $\mathcal{L}(\boldsymbol{\theta})$ are negligible

WARNING: the algorithm might converge to a local rather than the global maximum!



2.3 Example: fitting HMMs to movement data

A distribution for modelling step lengths

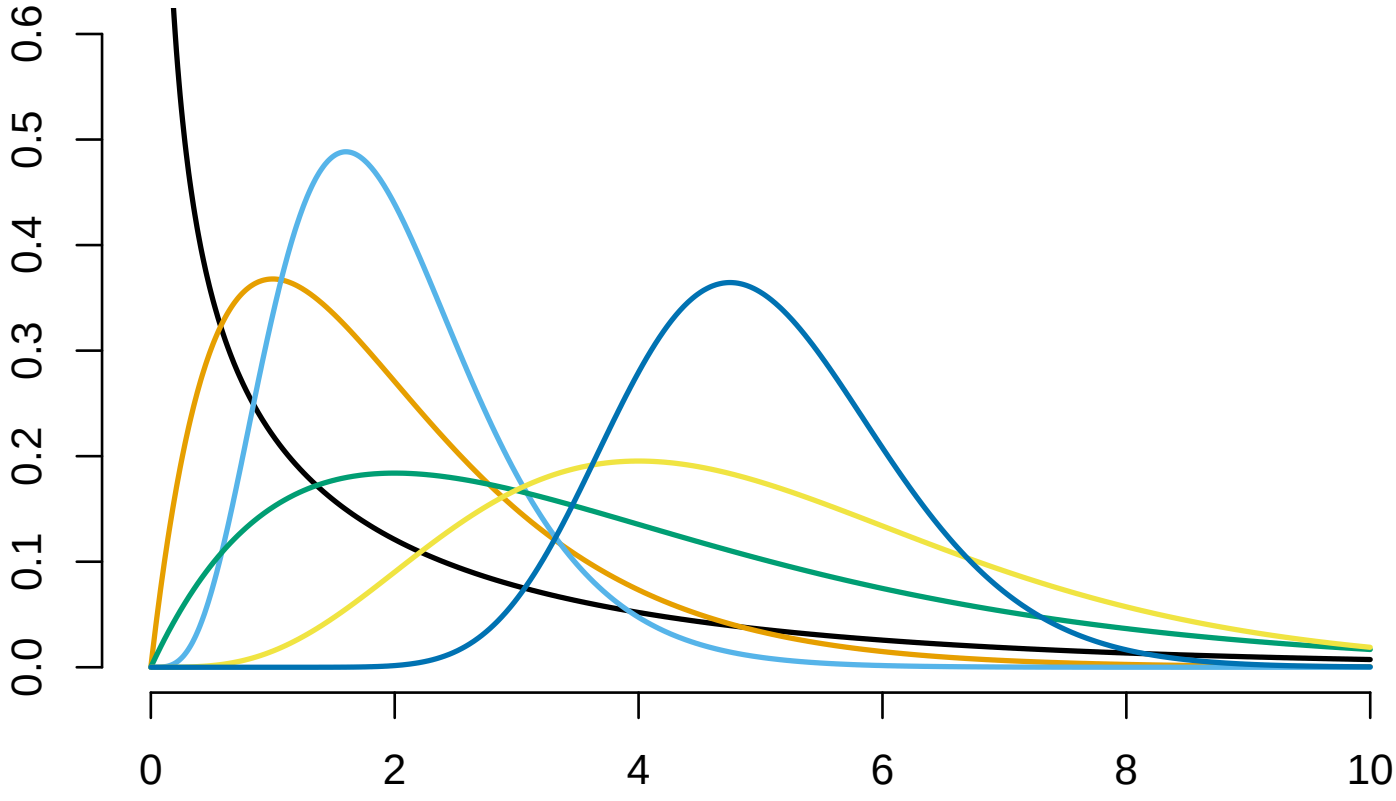


Figure: **Gamma distributions** with different mean (μ) and std. dev. (σ) parameters.

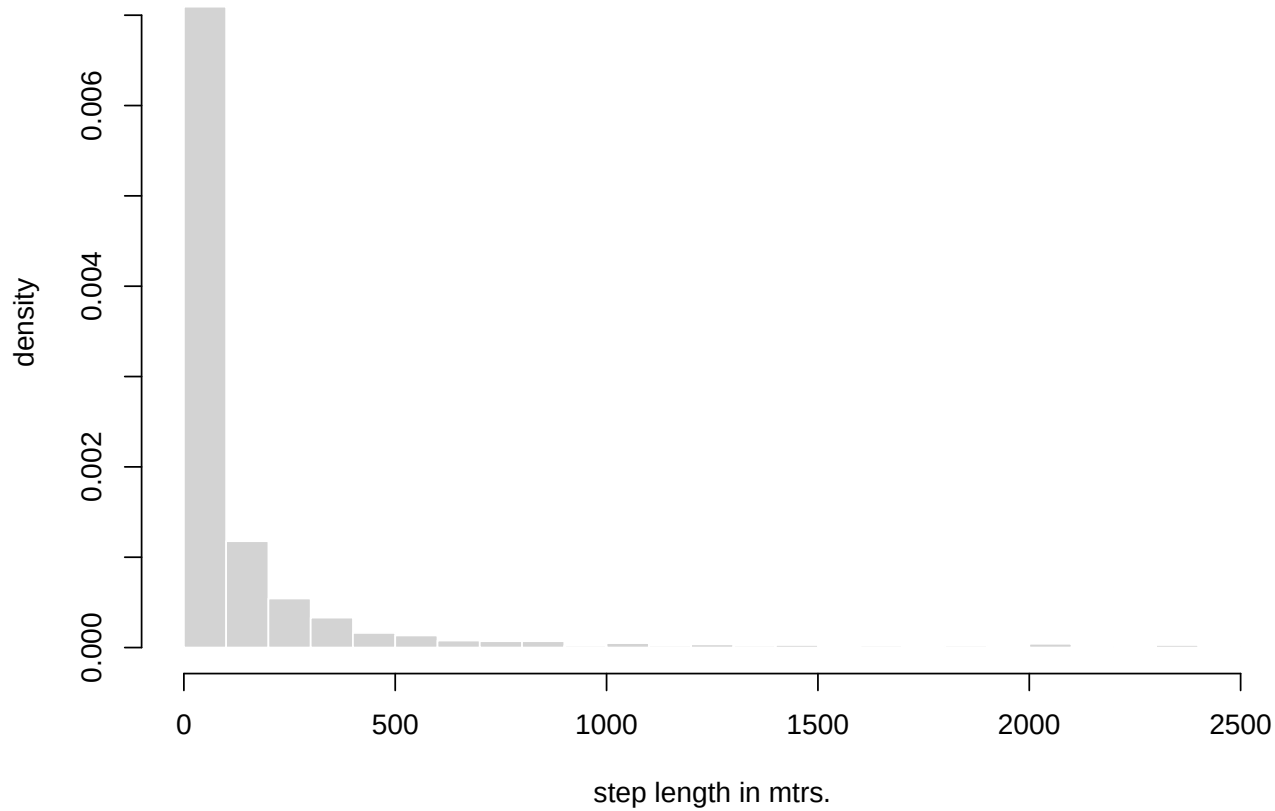


Figure: Histogram of the observed hourly step lengths.

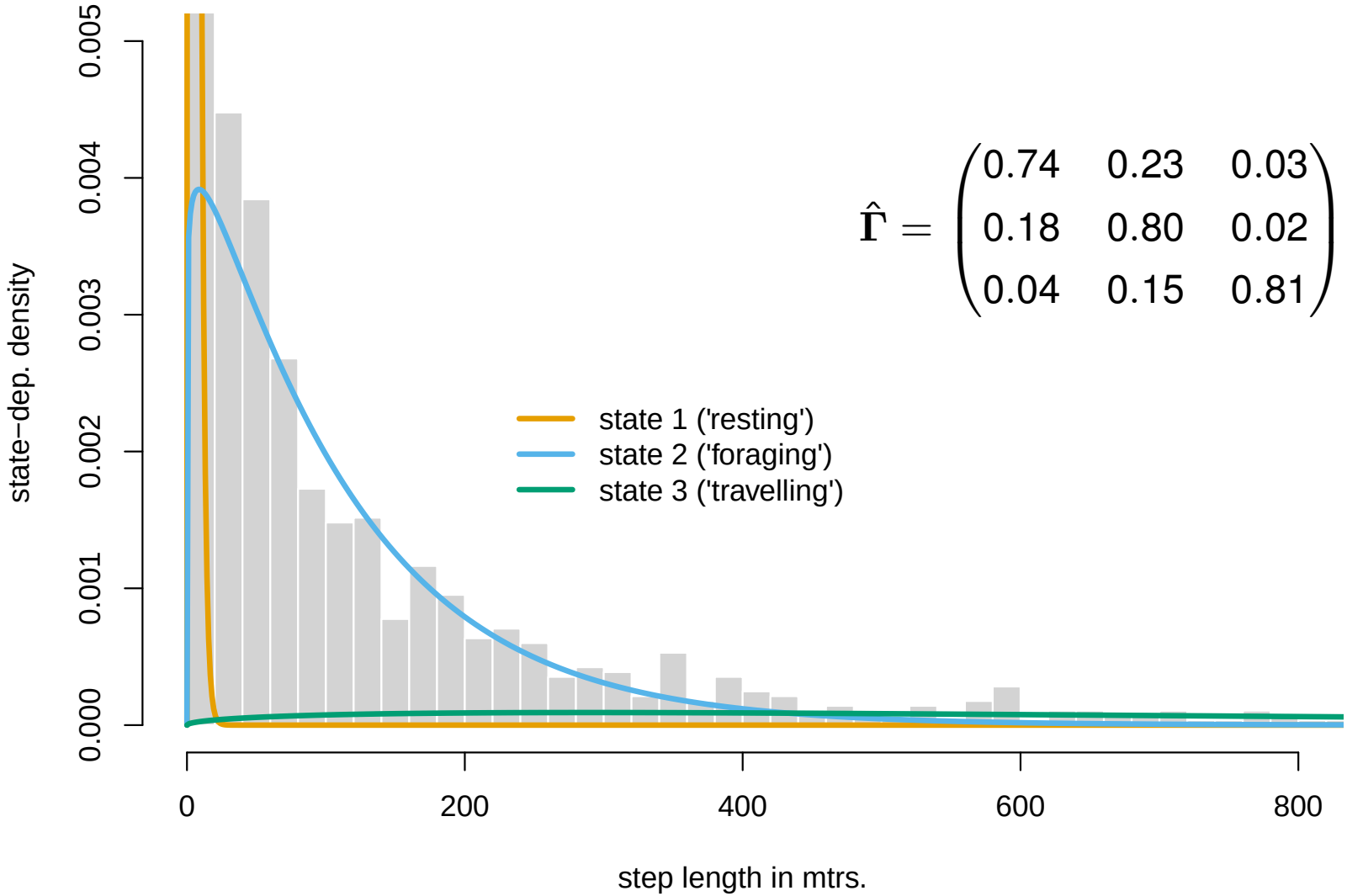
Eye-balling possible initial values for the gamma step length distributions:

- state-dep. means of (about) 5, 50 and 500 metres might be about right
- initial values for standard deviations can simply be taken to be a bit larger

Using `moveHMM` to fit a 3–state gamma HMM to the muskox data

```
muskox <- read.csv("http://www.rolandlangrock.com//muskox.csv")
# install.packages("moveHMM")
library(moveHMM)
data_muskox <- prepData(muskox, type="UTM")

# choose initial parameter values
stepMean0 <- c(5, 50, 500) # means of gamma distribution
stepSD0 <- c(10, 100, 1000) # standard deviations of gamma distribution
stepPar0 <- c(stepMean0, stepSD0)

s <- Sys.time()
muskoxhmm <- fitHMM(data=data_muskox, nbStates=3, stepPar0=stepPar0,
                    verbose=2, angleDist="none")
Sys.time()-s

muskoxhmm

plot(muskoxhmm)
```

Muskox step lengths — fitted 3–state gamma HMM



$$\hat{\mathbf{\Gamma}} = \begin{pmatrix} 0.74 & 0.23 & 0.03 \\ 0.18 & 0.80 & 0.02 \\ 0.04 & 0.15 & 0.81 \end{pmatrix}$$

Checking for local maxima

```
> llks <- rep(NA, 20)
> mods <- vector("list")
> for (k in 1:20){
+  step_mean0 <- runif(3, 0, 1000) # gamma means
+  step_sd0 <- runif(3, 0, 500) # gamma std. dev
+  step0 <- c(step_mean0, step_sd0)
+  mods[[k]] <- fitHMM(data, nbStates=3, stepPar0=step0,
+                      angleDist="none")
+  llks[k] <- -mods[[k]]$mod$minimum
+  print(llks)
+ }
 [1] -7388.818 -7193.963 -7812.318 -7388.818 -7193.963 -7812.318
 [7] -7388.818 -7812.318 -7388.818 -7388.818 -7388.818 -7388.818
[13] -7193.963 -7812.318 -7388.818 -7812.319 -7812.318 -7812.318
[19] -7193.963 -7193.963
```

Incorporating directionality

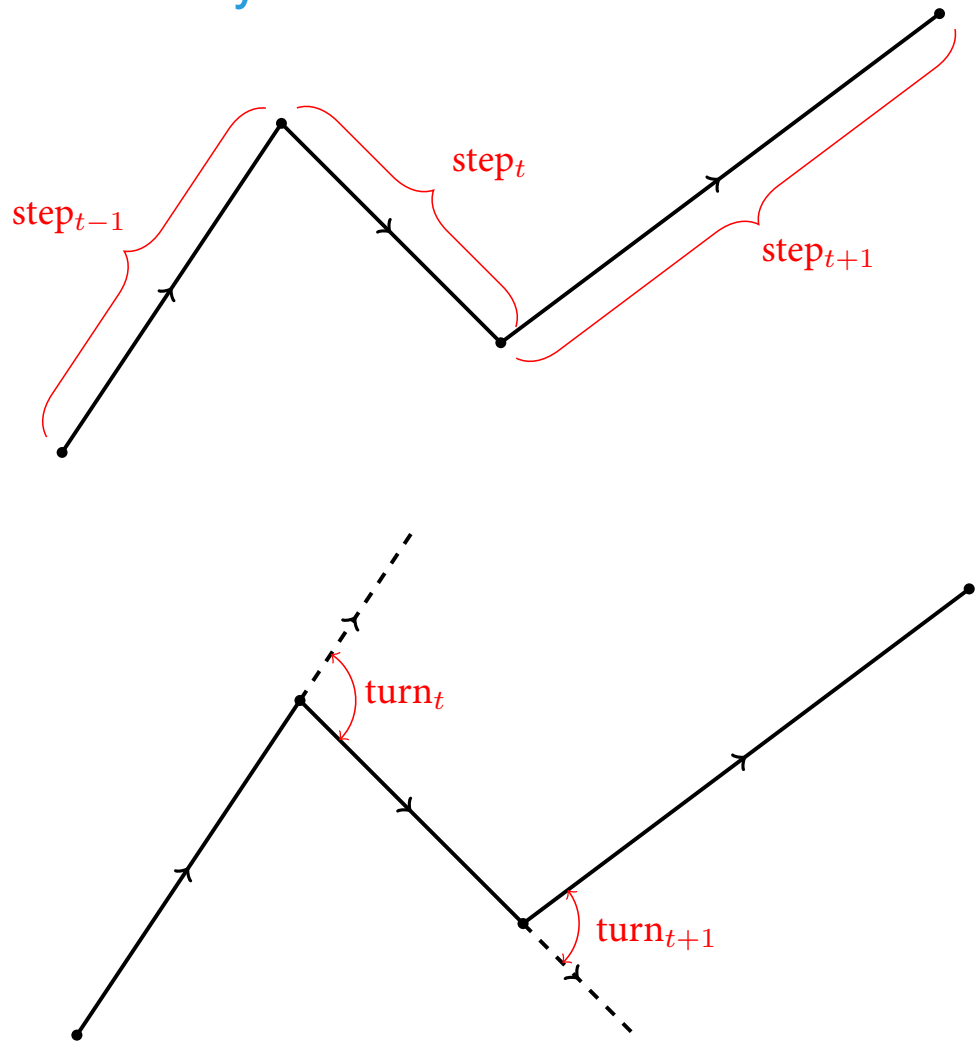


Figure: Representing location data in terms of step lengths (top) & turning angles (bottom).

HMM for bivariate time series of steps & turns

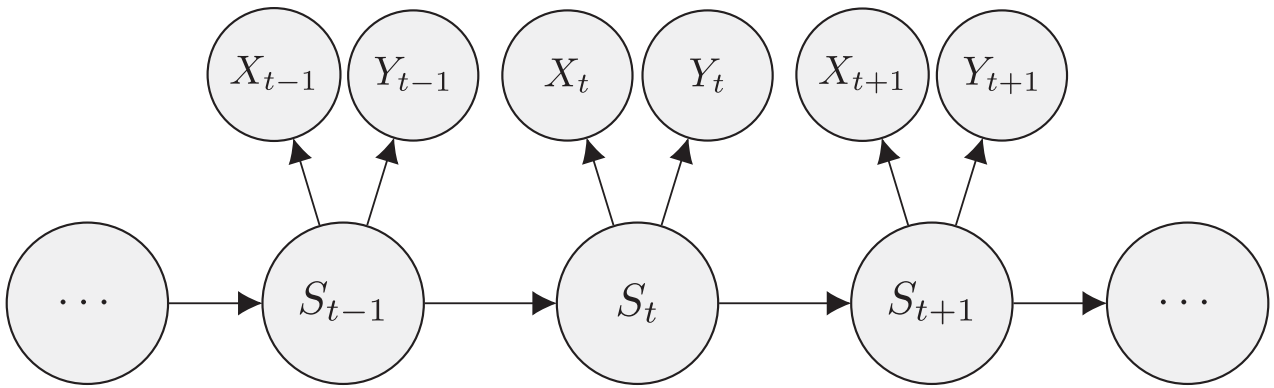


Figure: Simple extension of the basic HMM to accommodate bivariate observations.

- ⇝ we can again use gamma distributions for the steps X_t
- ⇝ but we need a special type of distribution for the turns Y_t

The von Mises distribution

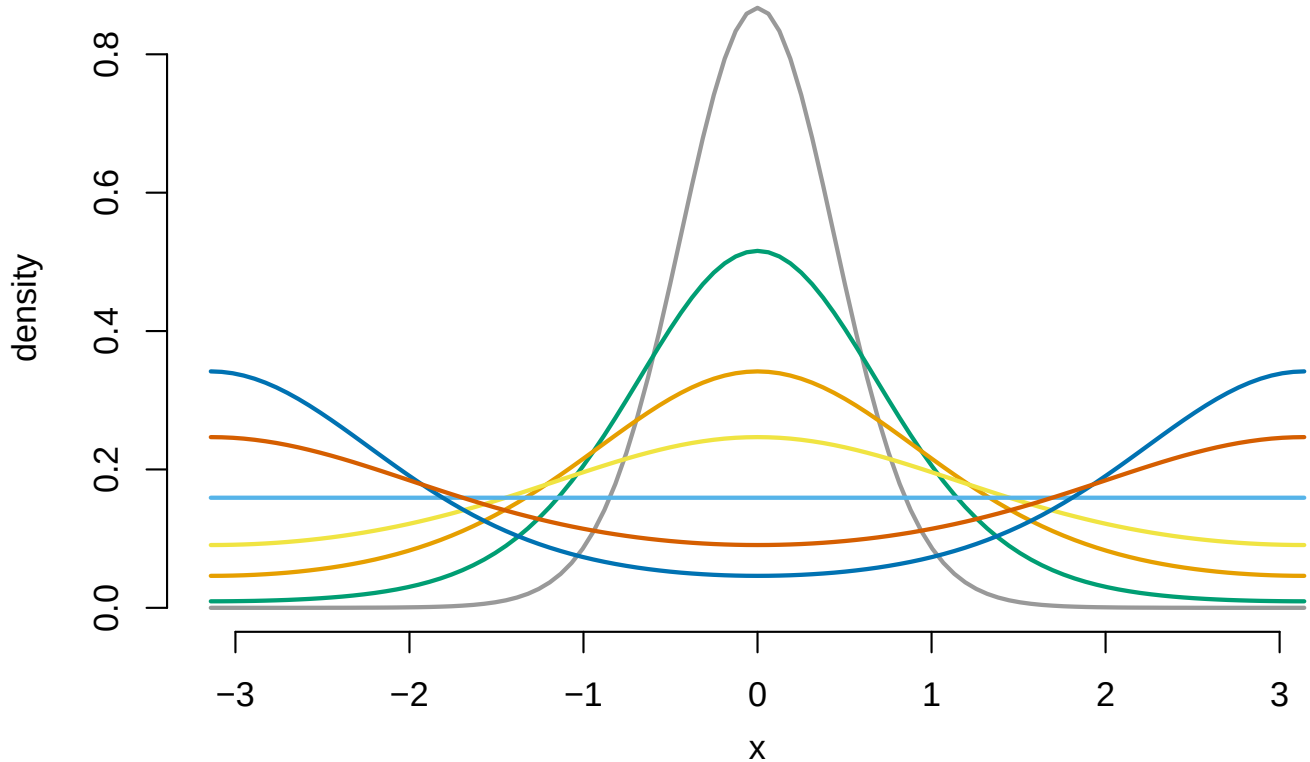
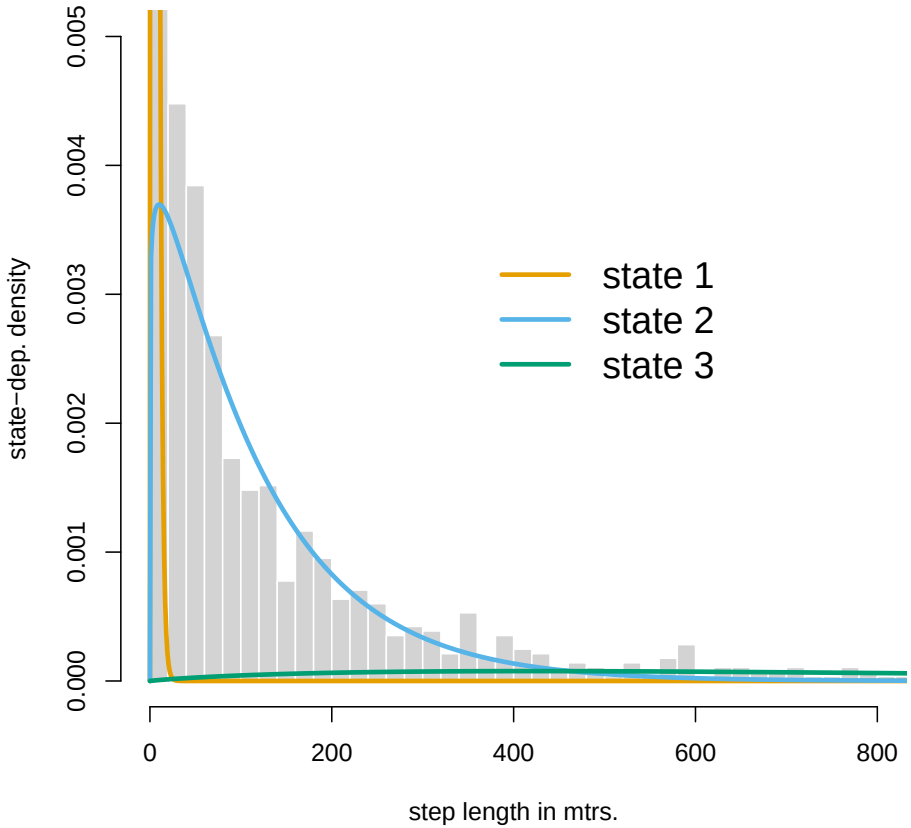


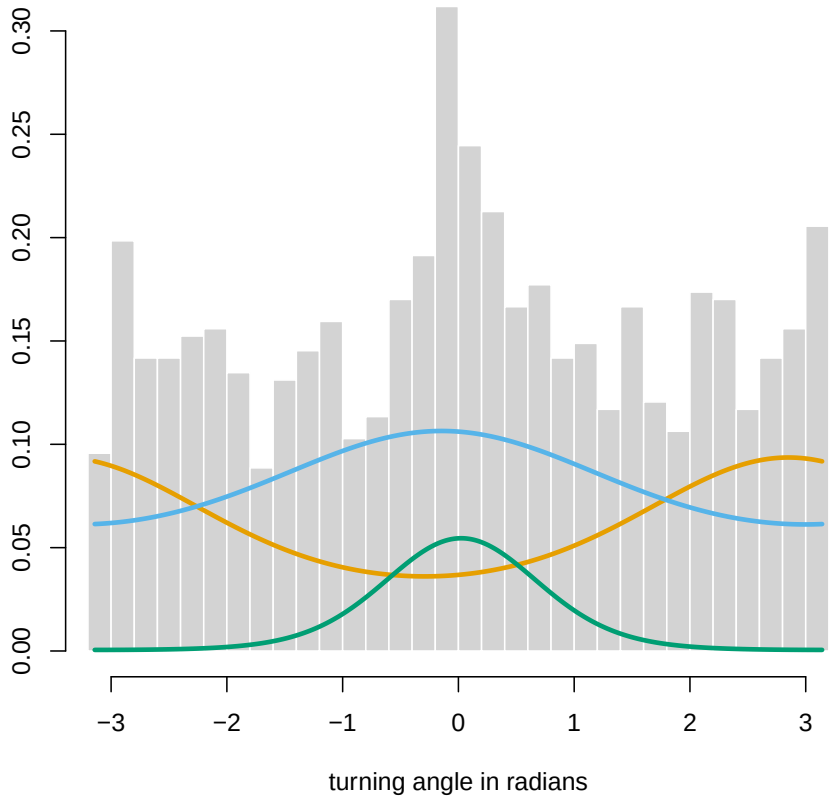
Figure: Example densities of **von Mises distributions** with different values of the mean (μ) and concentration (κ) parameters.

Using `moveHMM` to fit a 3–state gamma/von Mises HMM to the muskox data

```
muskox <- read.csv("http://www.rolandlangrock.com//muskox.csv")
library(moveHMM)
data_muskox <- prepData(muskox, type="UTM")

# choose initial parameter values
stepMean0 <- c(5, 40, 400) # means of gamma distribution
stepSD0 <- c(10, 50, 500) # standard deviations of gamma distribution
stepPar0 <- c(stepMean0, stepSD0)
angleMean0 <- c(pi, pi, 0) # means of von Mises   (turning angles)
angleCon0 <- c(0.5, 0.5, 1) # std. dev. of von Mises (turning angles)
anglePar0 <- c(angleMean0, angleCon0)

s<-Sys.time()
fullmuskoxhmm<-fitHMM(data=data_muskox, nbStates=3,
                  stepPar0=stepPar0, anglePar0=anglePar0, verbose=2)
Sys.time()-s

fullmuskoxhmm

plot(fullmuskoxhmm)
```

Muskox steps & turns — fitted 3–state gamma/von Mises HMM





$$\hat{\mathbf{\Gamma}} = \begin{pmatrix} 0.75 & 0.23 & 0.02 \\ 0.18 & 0.80 & 0.02 \\ 0.05 & 0.15 & 0.80 \end{pmatrix}$$

How do we fit *general* HMMs?

The `moveHMM` package was developed specifically for tracking data (steps & turns) and cannot deal with general types of data.

Options for fitting general HMMs:

- `momentuHMM` (an extension of `moveHMM` — much more flexible)
- `hmmTMB` (very good in particular for mixed models)
- `depmixS4` or `HiddenMarkov` (general HMM packages)
- write your own code (example implementation on the next slide)

Vanilla code for fitting a 3–state gamma HMM to the muskox data

```
muskox <- read.csv("http://www.rolandlangrock.com//Misc//muskox.csv")
library(moveHMM)
data <- prepData(muskox, type = "UTM")

mllk <- function(theta.star, x){
 Gamma <- diag(3)
 Gamma[!Gamma] <- exp(theta.star[1:6])
 Gamma <- Gamma / rowSums(Gamma)
 delta <- solve(t(diag(3) - Gamma + 1), rep(1, 3))
 mu <- exp(theta.star[7:9])
 sigma <- exp(theta.star[10:12])
 allprobs <- matrix(1, length(x), 3)
 ind <- which(!is.na(x))
 for (j in 1:3){
  allprobs[ind, j] <- dgamma(x[ind], shape = mu[j]^2 / sigma[j]^2,
                                     scale = sigma[j]^2 / mu[j])
 }
 foo <- delta %*% diag(allprobs[1, ])
 l <- log(sum(foo))
 phi <- foo / sum(foo)
 for (t in 2:length(x)){
  foo <- phi %*% Gamma %*% diag(allprobs[t, ])
  l <- l + log(sum(foo))
  phi <- foo / sum(foo)
 }
 return(-l)
}

theta.star <- c(rep(-2, 6), log(c(5, 50, 500)), log(c(10, 100, 800)))
nlm(mllk, theta.star, x = data$step, print.level = 2, iterlim = 10000)
```

2.4 Model selection

AIC and BIC

Aim: select the best HMM from a set of candidate models
(e.g. 2–state vs. 3–state or log-normal vs. gamma distributions)

$$\text{AIC} = -2 \log \mathcal{L} + 2 \cdot \#\text{parameters}$$

$$\text{BIC} = -2 \log \mathcal{L} + \log(T) \cdot \#\text{parameters}$$

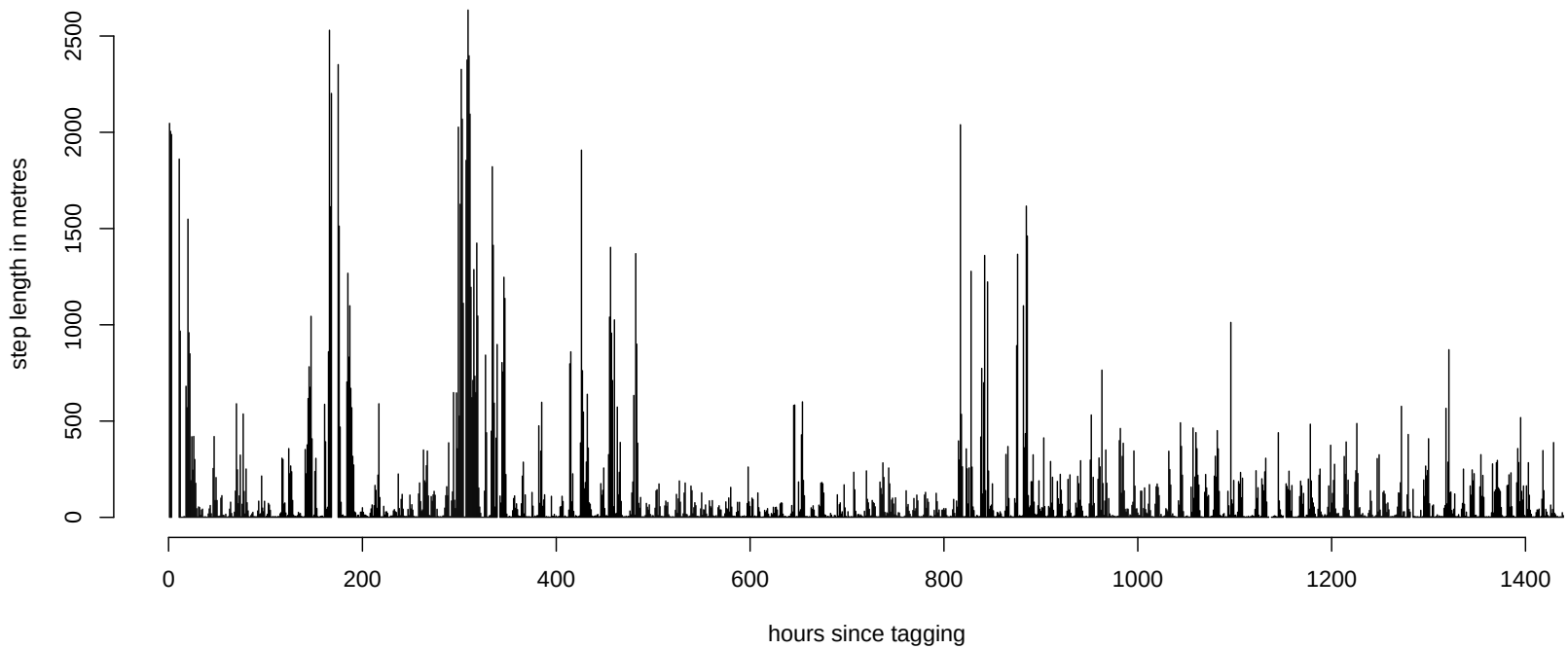
$\rightsquigarrow$ in either case, choose model with *smallest* value of the criterion applied.

- both criteria **reward goodness of fit** while **penalising complexity**
- AIC favours more complex models due to the smaller penalty (for $T \geq 8$)
- both provide a relative comparison of models from a suite of candidate models $\rightsquigarrow$ the selected model could still be a bad one!!

Illustration using the muskox data

In the following:

- we will illustrate the use of AIC and BIC for the muskox step lengths
- we fit N–state gamma HMMs with $N = 2, 3, 4, 5, 6$



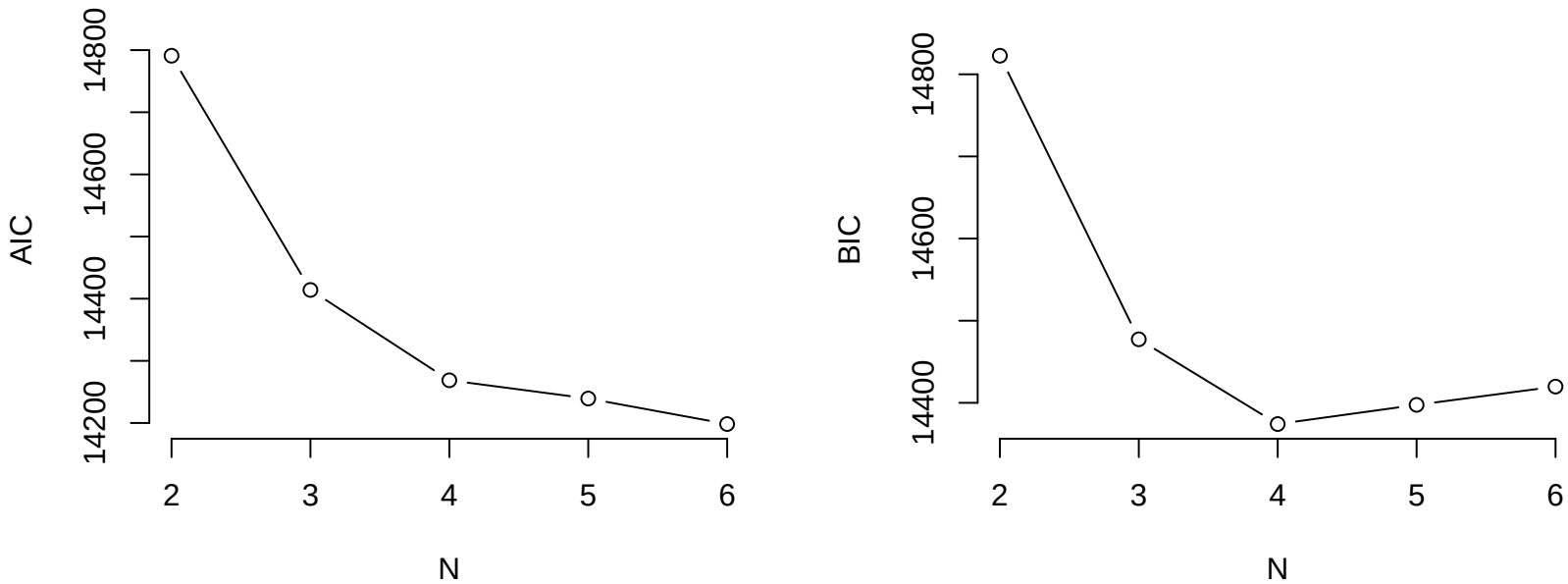


Figure: AIC & BIC values for gamma HMMs with different numbers of states.

- AIC selects the 6–state model[4]
- BIC selects the 4–state model
- biologist who collected the data thinks there should be 3 states ...

[4] and would probably go for even more states if we were to consider corresponding models...

Some remarks on model selection

Criteria like AIC or BIC are very appealing as they...

- have a theoretical foundation...
- ...yet are very easy to use,

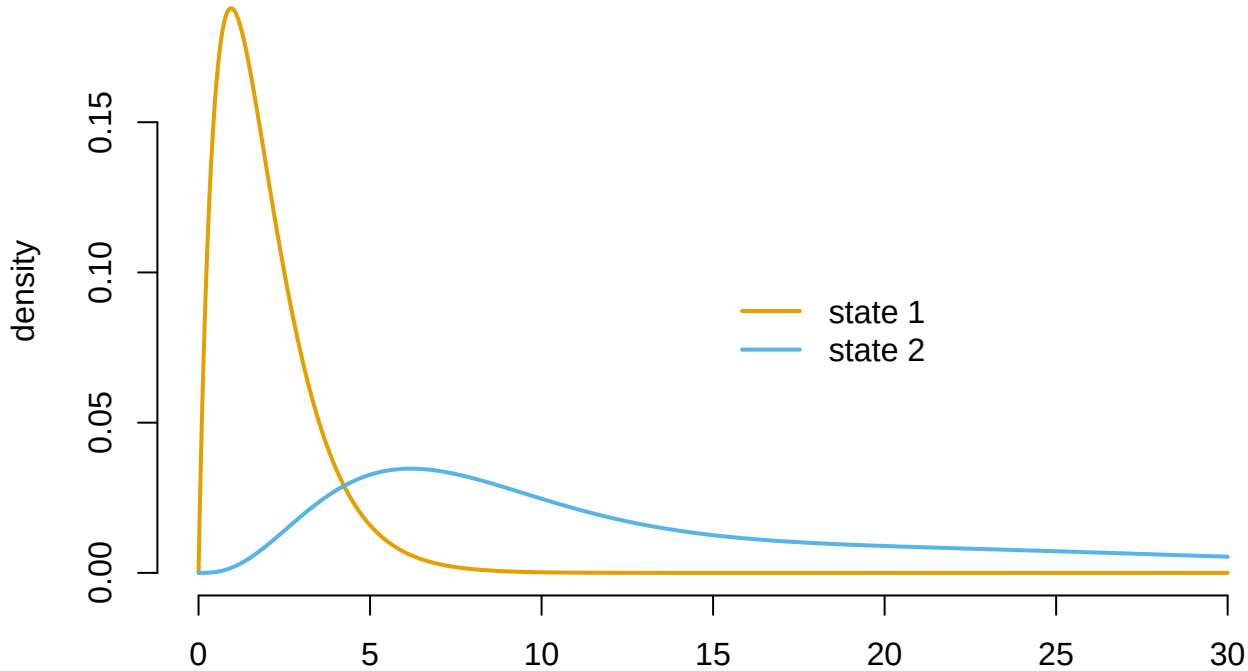
such that they appear to offer universally applicable solutions to model selection.

As a consequence, they are often applied **without any critical reasoning**.

However, both AIC and BIC have their shortcomings.

⇝ **don't blindly trust these criteria**, and instead only use them as guidance
(this in particular concerns the choice of the number of states!)

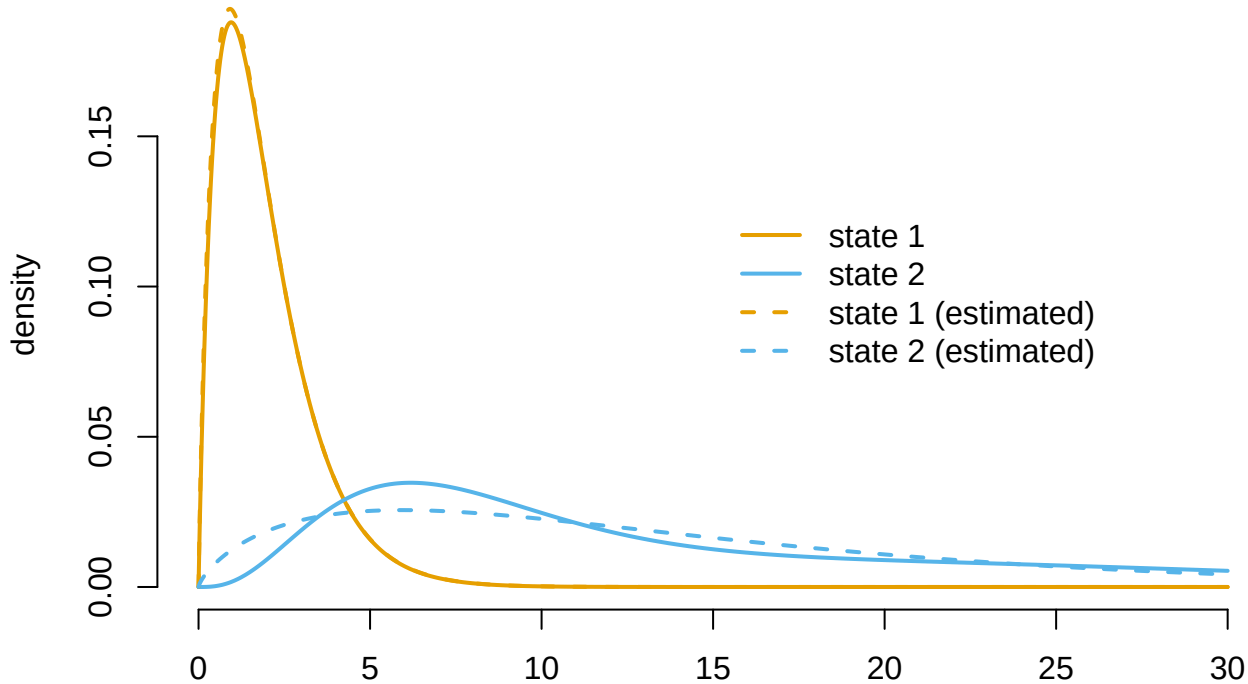
Illustrating example with simulated data



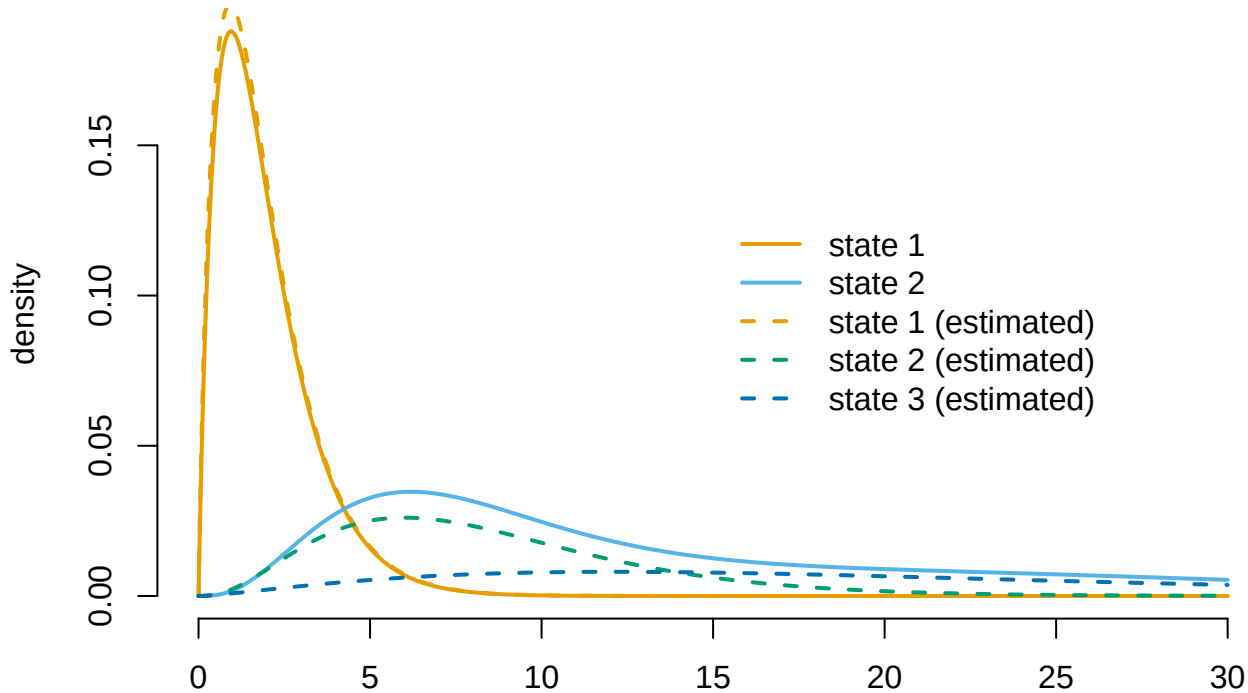
Observations were generated using the two state-dependent distributions displayed above and the t.p.m.

$$\Gamma = \begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}$$

If we fit gamma HMMs with $N = 2, 3$, then what N will be selected?



$\rightsquigarrow$ lack of fit of the distribution in state 2 due to insufficient flexibility



$\rightsquigarrow$ this (3–state) model is selected by both AIC and BIC

States mopping up structure

Model misspecification is often compensated by additional states, which “mop up” the neglected structure due to, e.g.,

- inadequate (parametric) state-dep. distributions
- outliers
- individual heterogeneity
- violations of Markov property or conditional independence assumption

⇝ for HMMs, AIC/BIC **tend to favour models with “too many” states**

Resulting models may fit the data better, but are often not interpretable:

- may not be a problem if model is used for forecasting
- but when inferring animal behaviours, then such an HMM can be useless

How to deal with this?

- ⇝ ideally, improve model formulation to account for such structure
- ⇝ often not feasible for complex data!
- ⇝ we may need to accept lack of fit and be pragmatic[5] about choosing N
- ⇝ in particular, I'd rather trust the ecologists' expertise than AIC/BIC

Bottom line: selecting N is **notoriously difficult**!!

[5] Pohle et al. (2017), Selecting the number of states in hidden Markov models: pragmatic solutions illustrated using animal movement, *JABES*.

2.5 Model checking

Model checking in HMMs

Main options to check if a fitted HMM is adequate:

1. graphical comparison of marginal distribution under fitted HMM and empirical distribution, to check adequacy of state-dep. distributions

2. simulate data from the fitted model, then compare the patterns found in the simulated data with those of the real data[6] $\rightsquigarrow$ informal, but useful strategy!

3. a residual analysis $\rightsquigarrow$ comprehensive formal check of the model

[6] patterns to look for: marginal distribution, autocorrelation, etc.

Marginal vs. empirical distribution

The marginal distribution of a fitted stationary HMM is

$$f(x) = \sum_{j=1}^{N} \delta_j f_j(x),$$

where δ_j is the stationary prob. of occupying state j. This can be compared to the empirical distribution of the data, as visualised e.g. using a histogram.

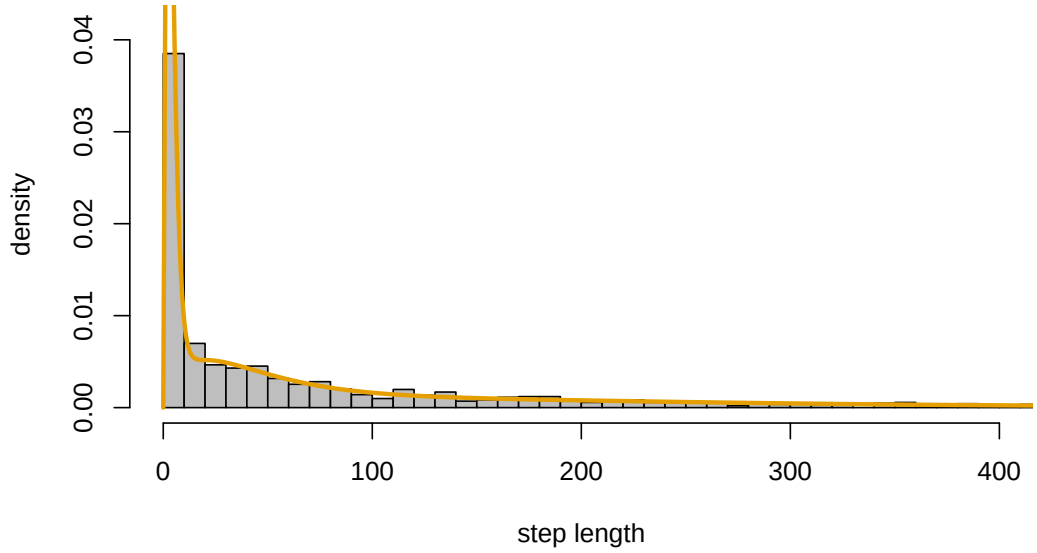


Figure: Histogram of the muskox' step lengths (truncated at 400 mtrs. for clarity) and marginal distribution under the fitted 4–state gamma HMM.

Simulation-based check

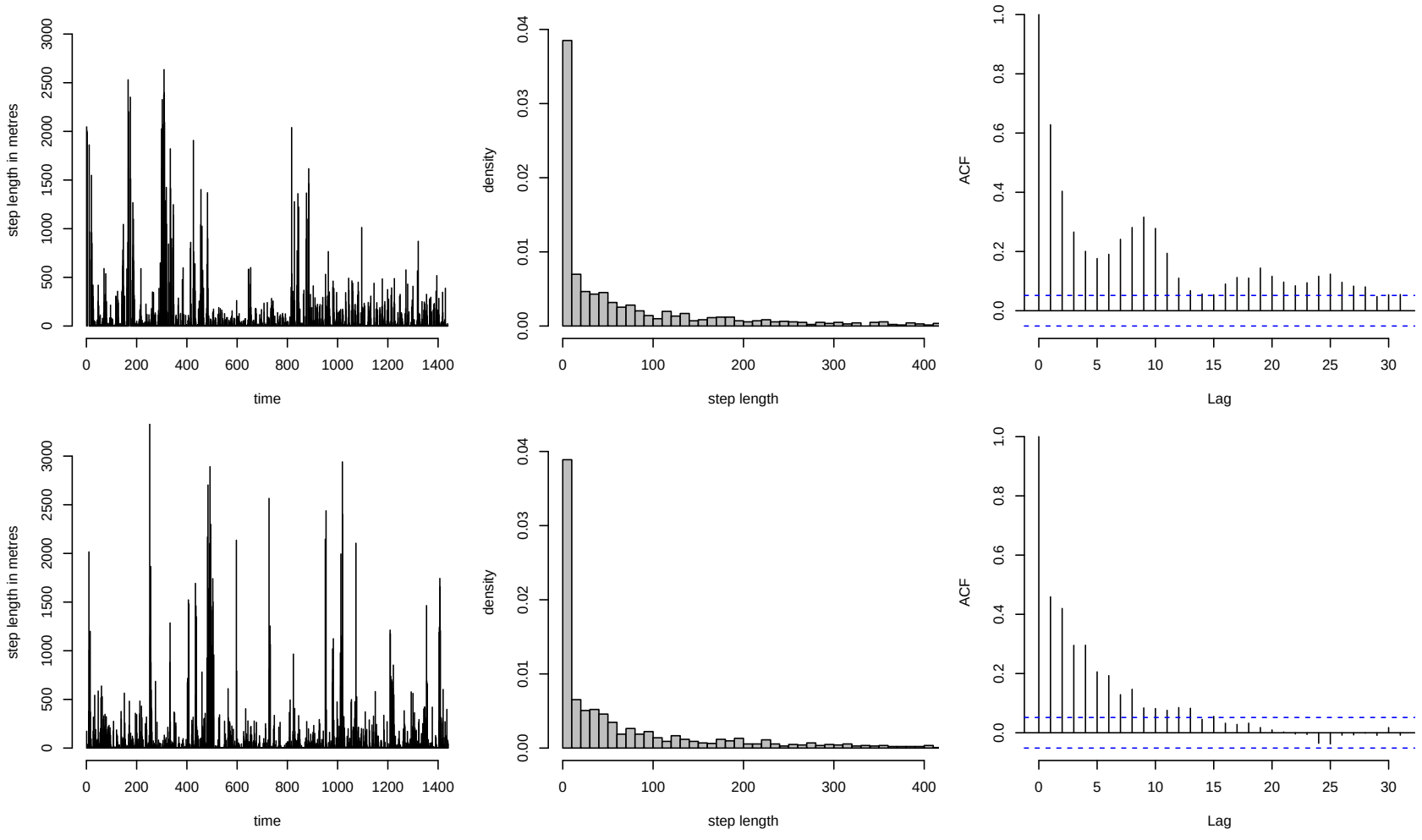


Figure: Comparison of the actual time series of muskox step lengths (top row) with a time series simulated from the fitted 4–state gamma HMM (bottom row).

Pseudo-residuals

Problem: due to the time series structure, each X_t has a different distribution, making it difficult to assess which observations are extreme *relative to the model*.

Trick: use **probability integral transform** to obtain a common scale.

- first convert X_t such that transformation is uniformly distributed:

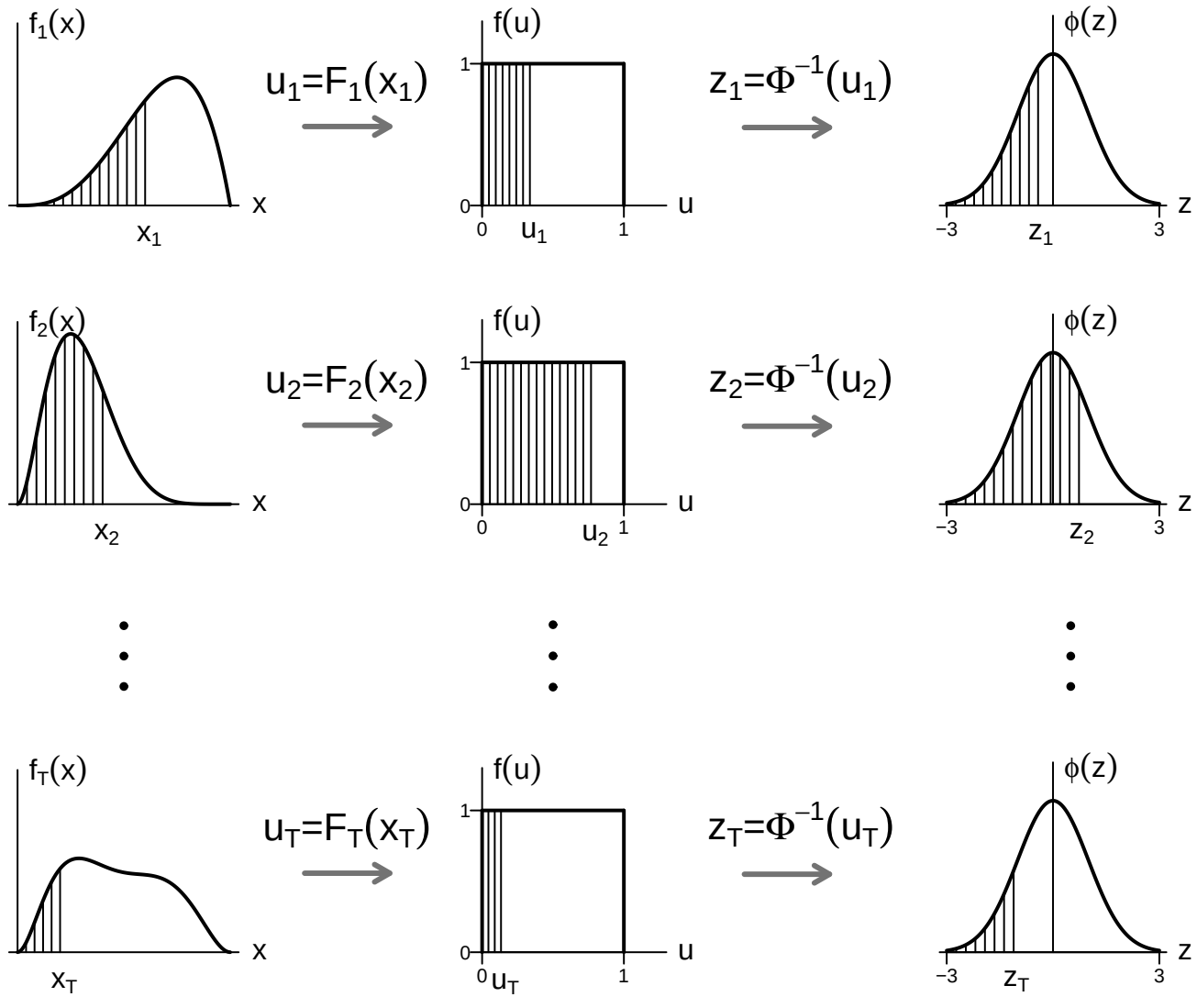
$$F_{X_t}(X_t) \sim \text{Uniform}[0,1]$$

- then convert once more to obtain standard normal distribution:

$$\Phi^{-1}(F_{X_t}(X_t)) \sim \mathcal{N}(0,1),$$

  which holds **if F_{X_t} is correct**

Construction of pseudo-residuals



Given a fitted HMM, we consider the conditional distribution:

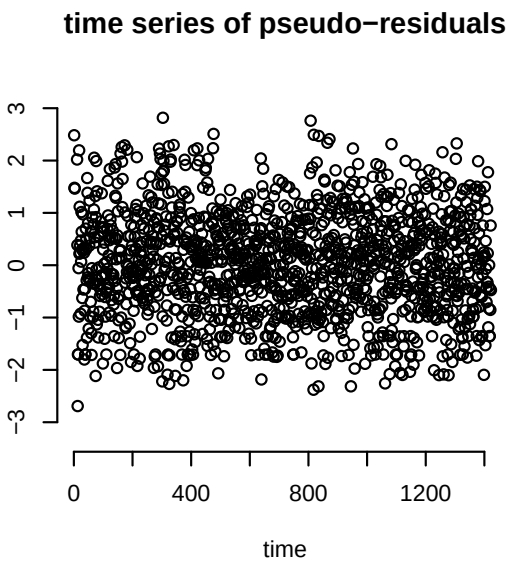
$$F_{X_t}(x_t) = \Pr(X_t \leq x_t \,|\, X_1 = x_1, \ldots X_{t-1} = x_{t-1}, X_{t+1} = x_{t+1}, \ldots X_T = x_T)$$

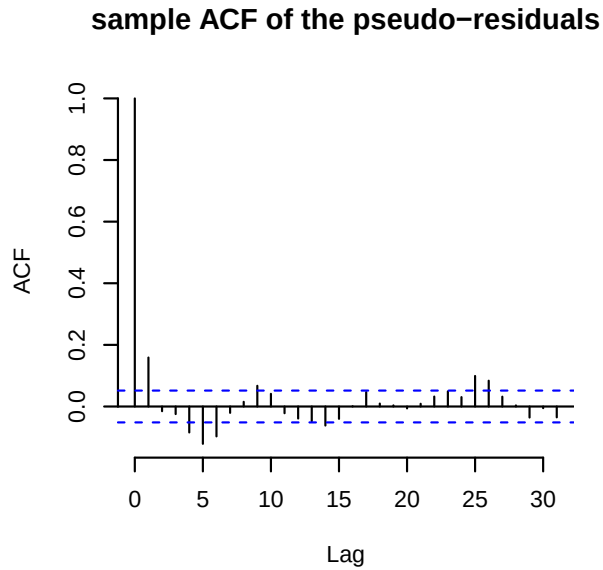
(these are obtained using the forward and the backward variables)

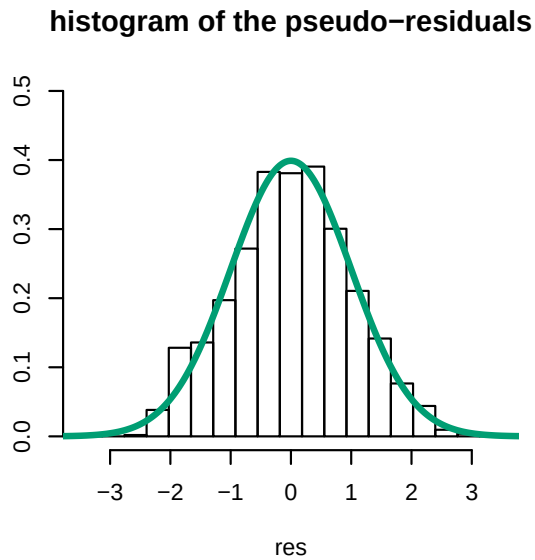
The quantities $\Phi^{-1}\big(F_{X_t}(X_t)\big)$ are called **pseudo-residuals**. If the model is correct, then these (random variables) are standard normally distributed.

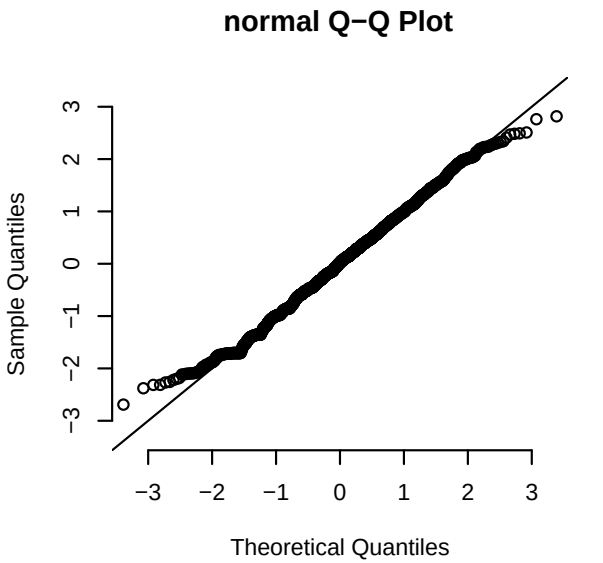
- we can use quantile-quantile-plots and/or hypothesis tests to check for normality: any indication of non-normality $\rightsquigarrow$ indication of a lack of fit!
- strong residual autocorrelation $\rightsquigarrow$ correlation structure not fully captured!

Pseudo-res. for the muskox data, obtained under fitted 4–state model









So when are we satisfied with a model?

1. it **helps to address the study aim**
   (e.g. forecasting, classification, obtaining overview of patterns)

2. it “outperforms” other reasonable candidate models

3. model checks don’t reveal any substantial lack of fit — in particular,
   - pseudo-residuals should be approx. normally distributed...
   - ...and exhibit little autocorrelation

2.6 State decoding

Given a fitted model, it is often of interest to **decode the hidden states** underlying the observed time series.

Global decoding: looks at the sequence as a whole

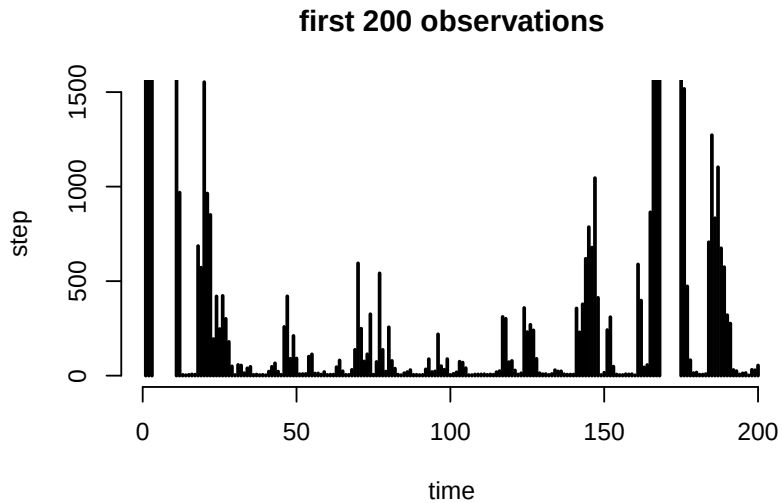
- consider $\Pr(S_1 = i_1, \ldots, S_T = i_T | x_1, \ldots, x_T)$
- most probable state sequence is maximum of the above over $(i_1, \ldots, i_T) \in \{1, \ldots, N\}^T$

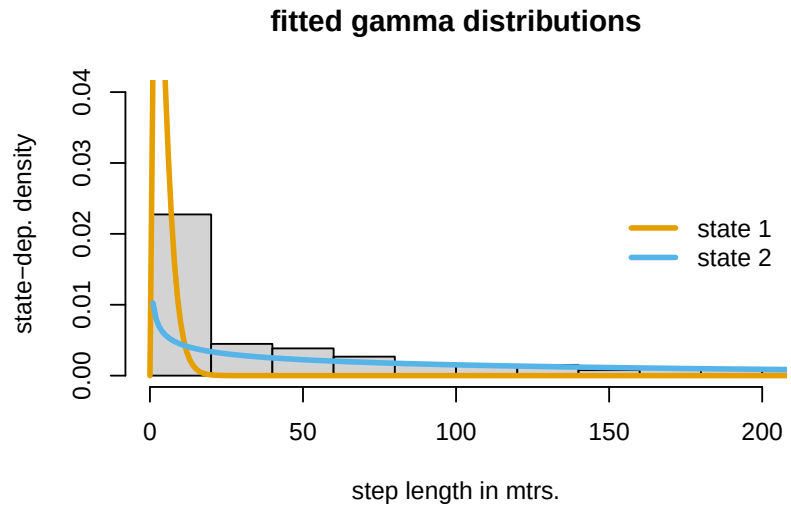
Local decoding: looks at each time point in isolation

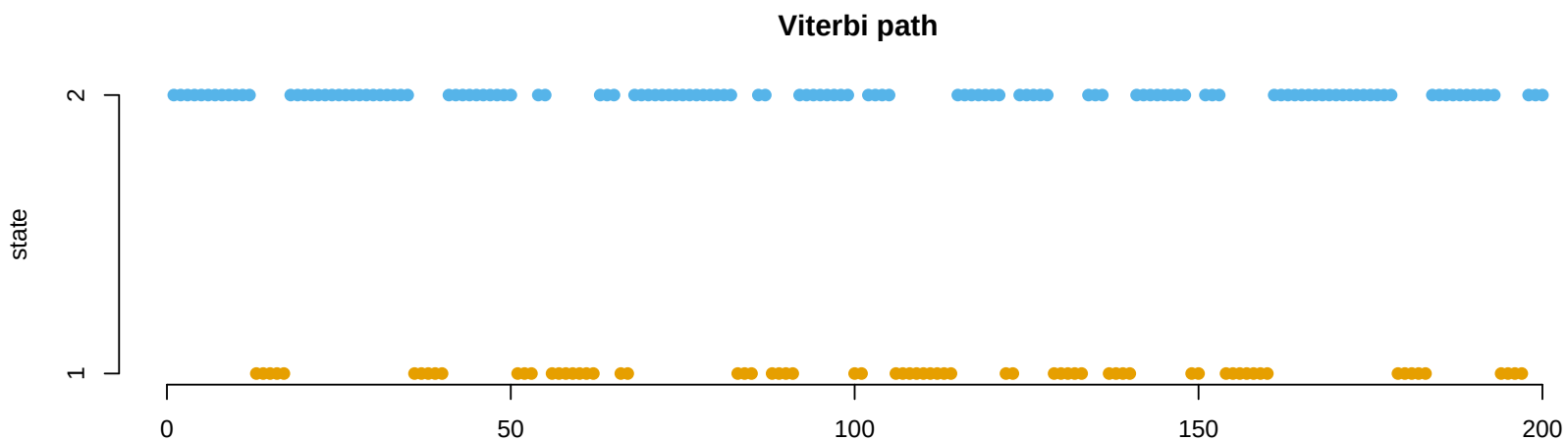
- consider $\Pr(S_t = i | x_1, \ldots, x_T)$
- most probable state at time t is maximum of the above over $i = 1, \ldots, N$

Usually the outcome is either identical or at least very similar. But local decoding additionally provides uncertainty information.

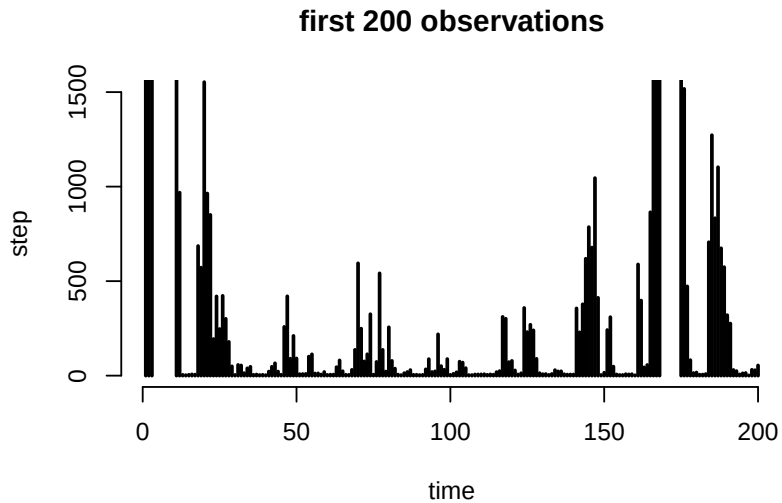
Global decoding (using the Viterbi algorithm) in the muskox example

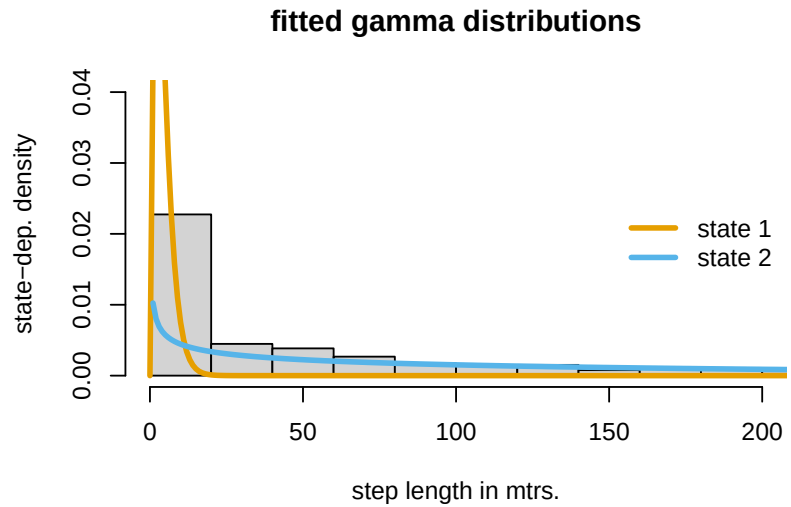


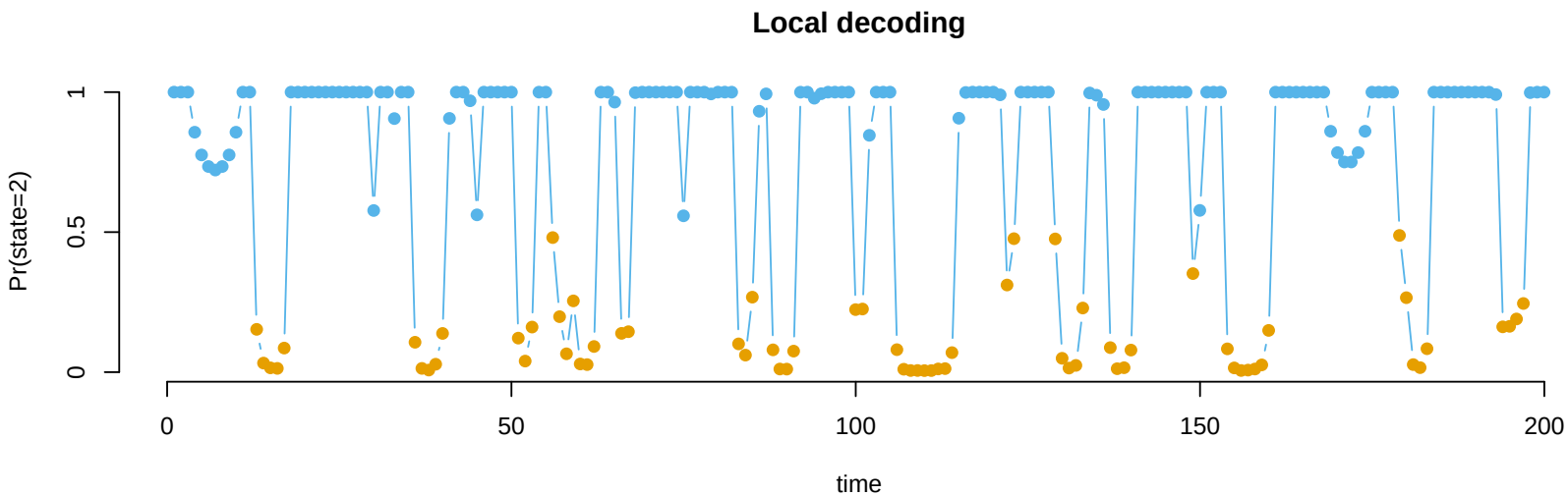




Local decoding (using the forward-backward algorithm) in the example







Part 3 — What else can we do with HMMs?

3.1 Covariates, seasonality & random effects

Why and how to include covariates in HMMs?

The inclusion of covariates allows to investigate how the dynamic system HMM responds to potential drivers:

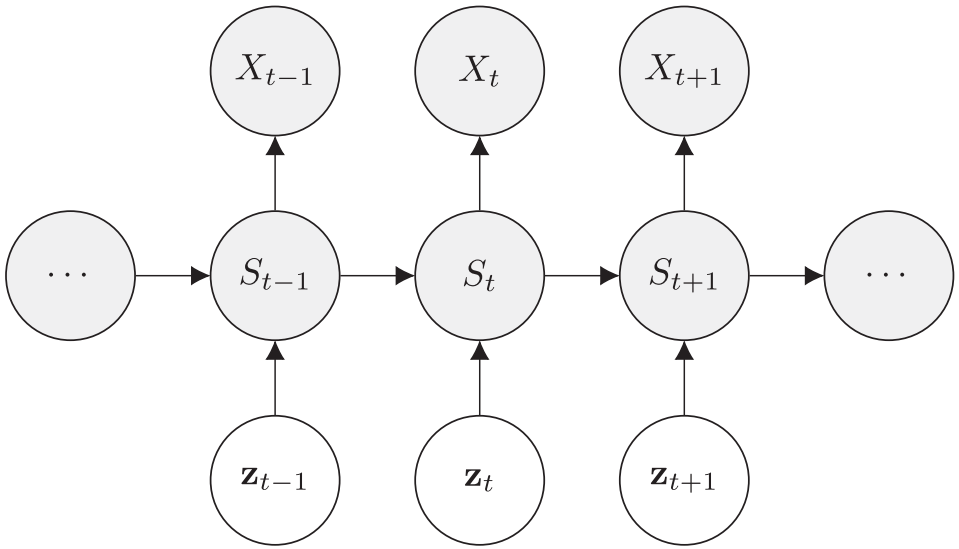
- how does animal behaviour change with varying temperatures?
- how does a cancer patient respond to medication?
- how does the financial market respond to changes in say the interest rate?

Technically it's straightforward to incorporate covariates in HMMs, in both state process and state-dep. process: the likelihood structure remains unaffected.

$$\mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{\delta}^{(1)}\mathbf{P}(x_1)\boldsymbol{\Gamma}\mathbf{P}(x_2)\boldsymbol{\Gamma}\mathbf{P}(x_3)\dots\boldsymbol{\Gamma}\mathbf{P}(x_T)\mathbf{1}^t$$

$\rightsquigarrow$ all that changes is that $\mathbf{P}(x_t)$ and/or $\boldsymbol{\Gamma}$ will depend on covariates

Covariates in the state process



We first consider the inclusion of **covariates in the state process**,

- expressing state transition probabilities as functions of covariates...
- ...to infer how state dynamics depend on external (or internal) factors

Muskox example with covariate influence in the state process

Consider a 2–state HMM for steps and turns, with temp_t affecting the transitions:

$$\mathbf{\Gamma}^{(t)} = \begin{pmatrix} 1-\gamma_{12}^{(t)} & \gamma_{12}^{(t)} \\ \gamma_{21}^{(t)} & 1-\gamma_{21}^{(t)} \end{pmatrix},$$

where, for $i \neq j$,

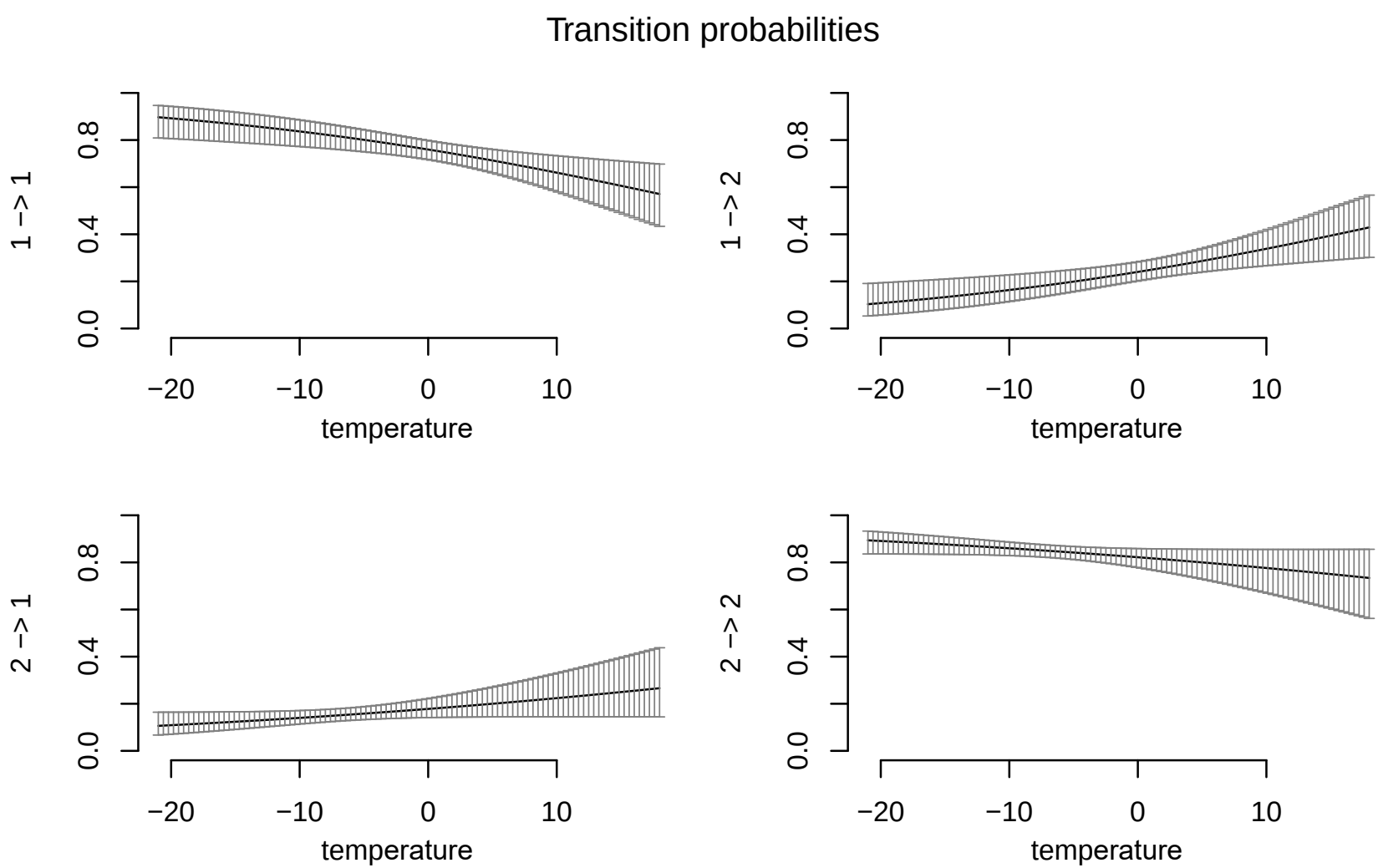
$$\gamma_{ij}^{(t)} = \text{logit}^{-1}\left(\beta_0^{(ij)} + \beta_1^{(ij)} \cdot \text{temp}_t\right) = \frac{e^{\beta_0^{(ij)} + \beta_1^{(ij)} \cdot \text{temp}_t}}{e^{\beta_0^{(ij)} + \beta_1^{(ij)} \cdot \text{temp}_t} + 1}$$

(extension to multiple covariates, quadratic effects, interactions etc. is obvious)

In `moveHMM`:

```
muskox <- read.csv("http://www.rolandlangrock.com//muskox.csv")
# install.packages("moveHMM")
library(moveHMM)
data_muskox <- prepData(muskox, type="UTM")

step_mean0 <- c(5, 200)
step_sd0 <- c(5, 100)
step0 <- c(step_mean0, step_sd0)
angle_mean0 <- c(pi, 0)
angle_con0 <- c(0.5, 1)
angle0 <- c(angle_mean0, angle_con0)

muskoxhmm <- fitHMM(data, nbStates = 2, stepPar0 = step0,
                    anglePar0 = angle0, formula = ~temp, verbose = 2)
muskoxhmm

plot(muskoxhmm, plotCI = TRUE)
```

```
> muskoxhmm
Value of the maximum log-likelihood: -9907.767

Step length parameters:
-----------------------
      state 1  state 2
mean 4.714617 229.1425
sd   3.323690 281.4776

Turning angle parameters:
-------------------------
                state 1     state 2
mean          2.8735816 -0.06313096
concentration 0.5299382  0.47877136

Regression coeffs for the transition probabilities:
---------------------------------------------------
               1 -> 2      2 -> 1
intercept -1.15149048 -1.52939023
temp       0.04811378  0.02857322

Initial distribution:
---------------------
[1] 9.458273e-06 9.999905e-01
```



(the colder it is, the longer the animal stays in a state — but this data set is small)

Another example: grey seal behaviour & commercial fishing

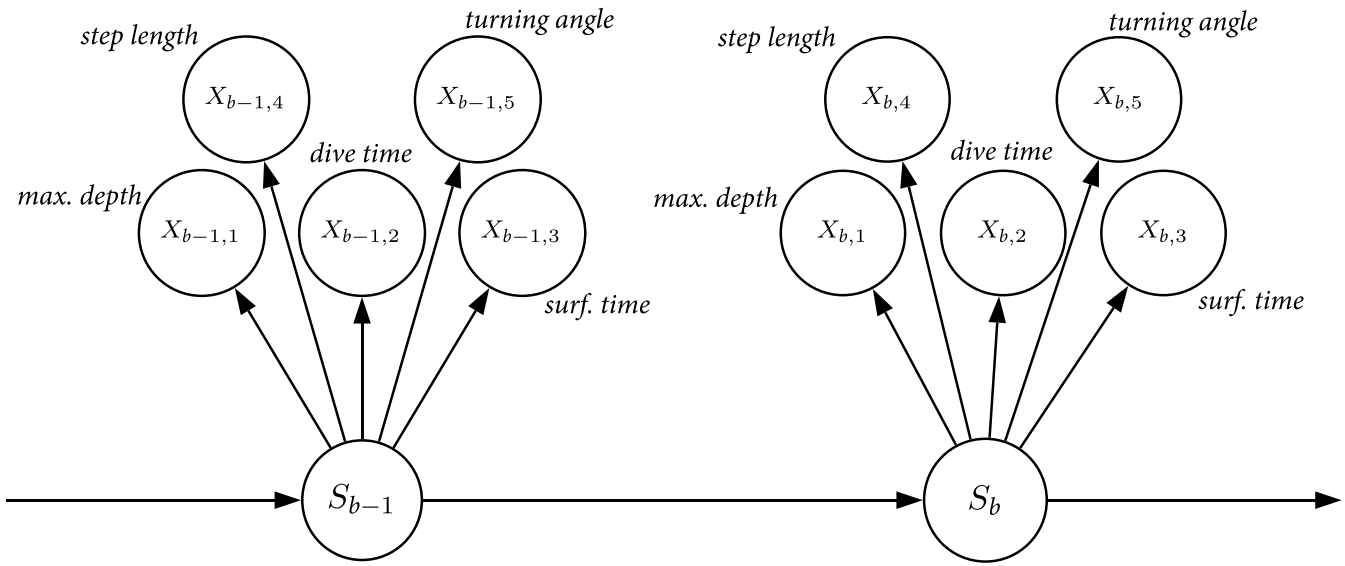


Figure: HMM for multivariate dive-by-dive data collected for grey seals in the Baltic Sea.

- 3–state model (“foraging”, “resting”, “travelling”) was deemed most adequate
- covariate effects on state process modelled using multinomial logit link
- main covariate of interest: **distance to nearest fishing net**
- additional covariates included: sediment type, sex, bathymetry, salinity

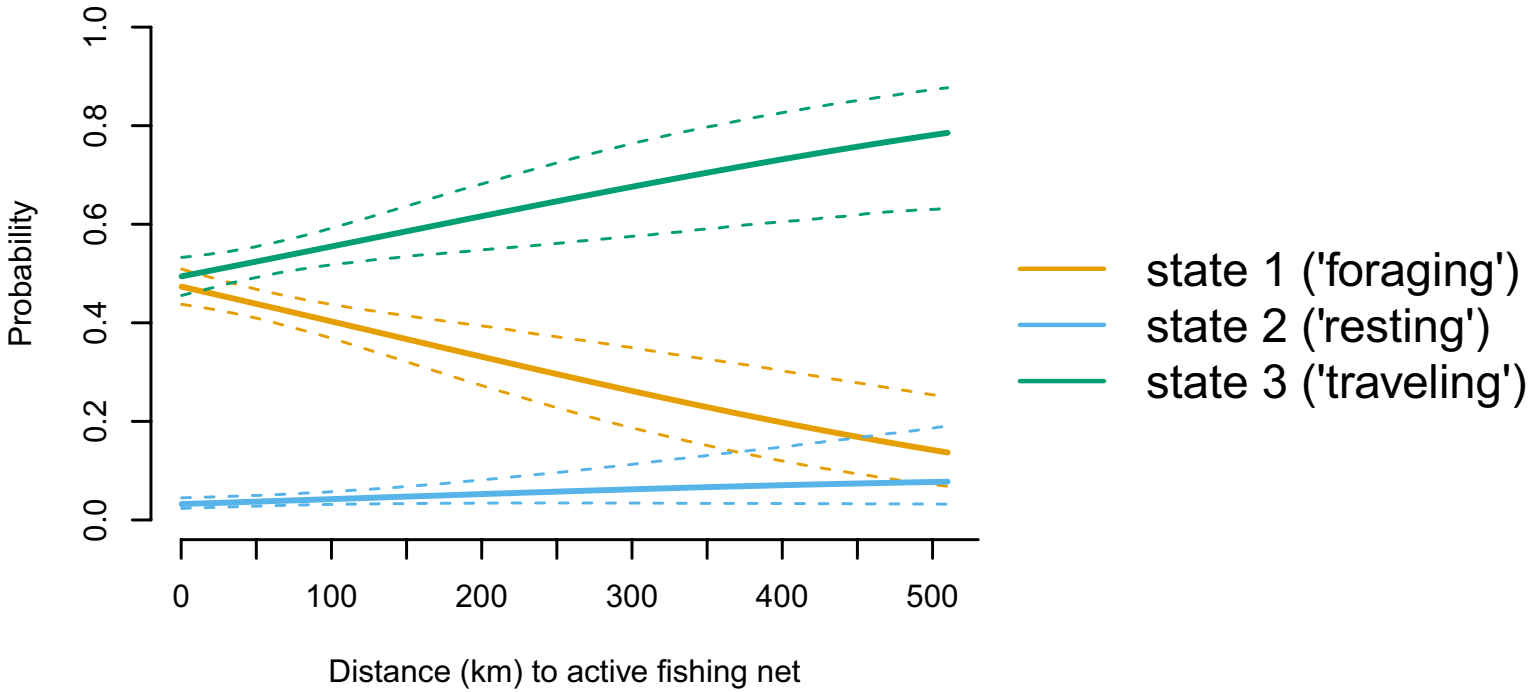


Figure: Estimated state occupancy as a function of the distance to the nearest fishing net.

In this case we have $\Delta\text{AIC} = 37.2$, indicating fairly strong correlation[7].

[7] but of course we're not estimating a causal effect here

Incorporating seasonality or diel patterns

In many real data applications, there is within-day or within-year variation.

In essence, this means that parameters depend on the covariate time_t.

However,

- predictors ought to return to where they started after completing a full cycle
- this can be achieved by using trigonometric functions
  (with period equal to day, year, or whatever the assumed cycle length)

For the muskox data, consider the 2–state HMM as before, but now with

$$\gamma_{ij}^{(t)} = \text{logit}^{-1}\left(\beta_0^{(ij)} + \beta_1^{(ij)} \sin\left(\frac{2\pi \cdot \text{time}_t}{24}\right) + \beta_2^{(ij)} \cos\left(\frac{2\pi \cdot \text{time}_t}{24}\right)\right)$$

Muskox example with within-day variation in the state process

In `moveHMM`:

```
muskox <- read.csv("http://www.rolandlangrock.com//muskox.csv")
# install.packages("moveHMM")
library(moveHMM)
data_muskox <- prepData(muskox, type="UTM")

step_mean0 <- c(5, 200)
step_sd0 <- c(5, 100)
step0 <- c(step_mean0, step_sd0)
angle_mean0 <- c(pi, 0)
angle_con0 <- c(0.5, 1)
angle0 <- c(angle_mean0, angle_con0)

muskoxhmm <- fitHMM(data, nbStates = 2, stepPar0 = step0,
                    anglePar0 = angle0, verbose = 2,
                    formula = ~sin(2*pi*tod/24)+cos(2*pi*tod/24))
muskoxhmm

plot(muskoxhmm, plotCI = TRUE)
plotStationary(muskoxhmm, plotCI = TRUE)
```

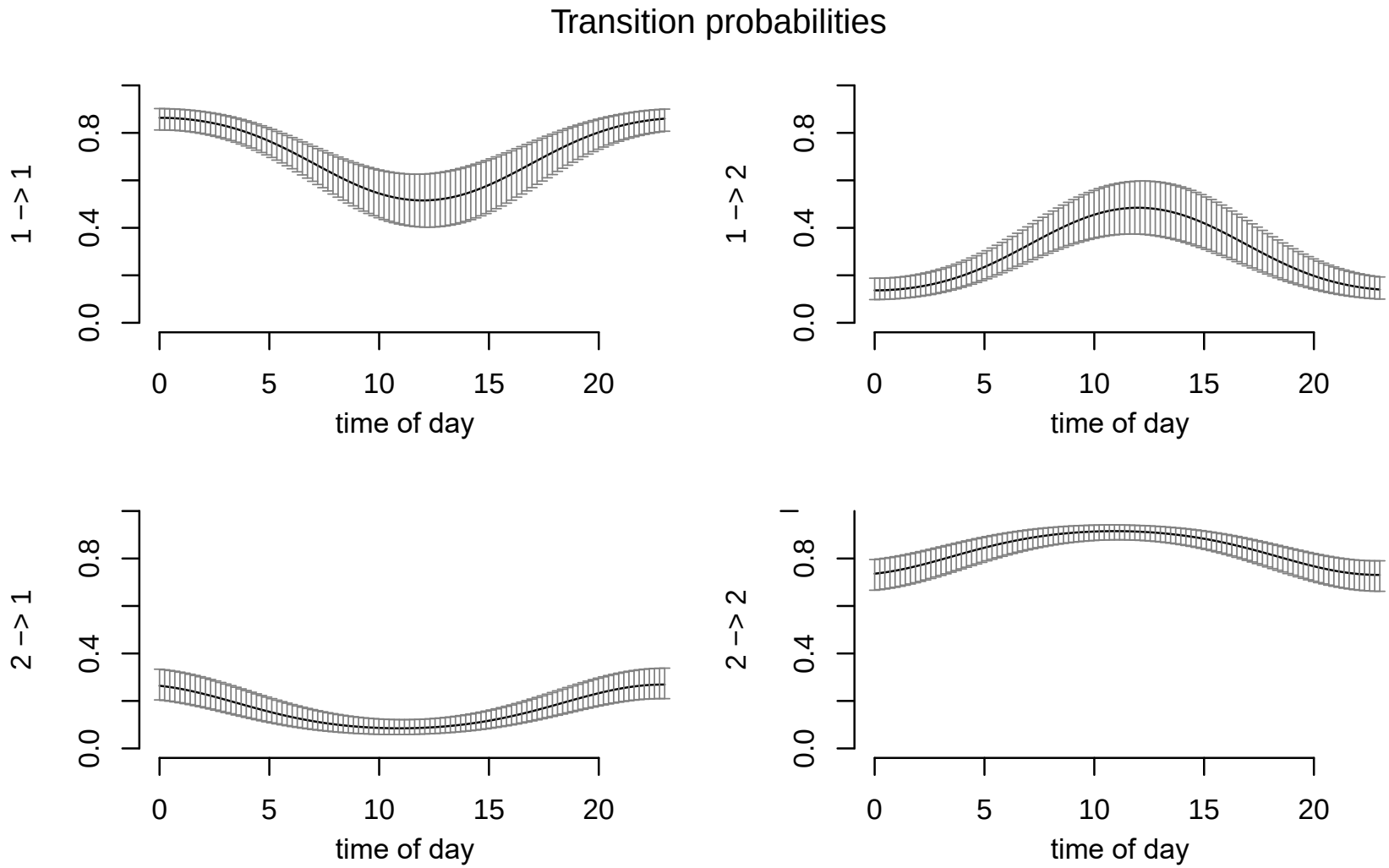
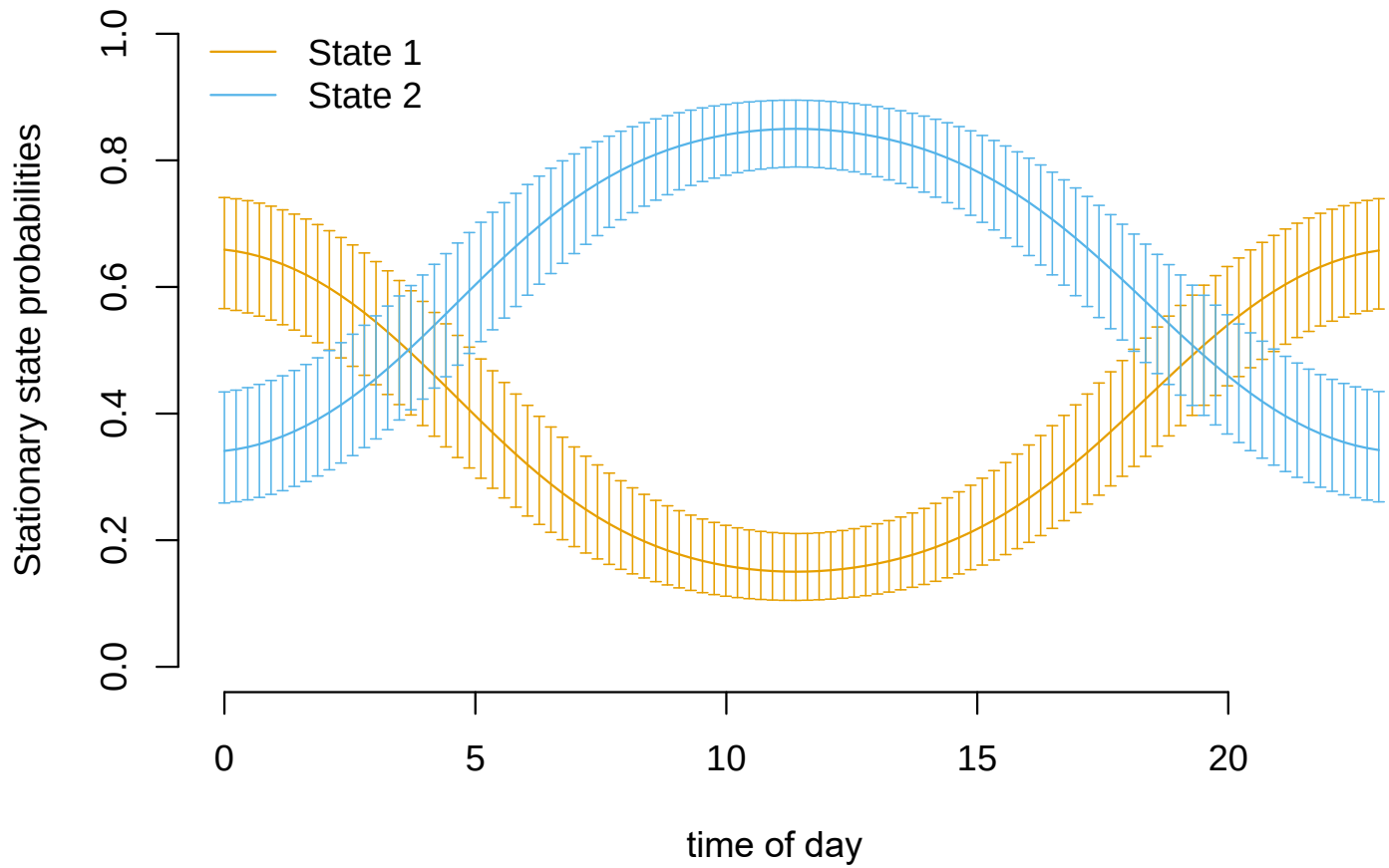
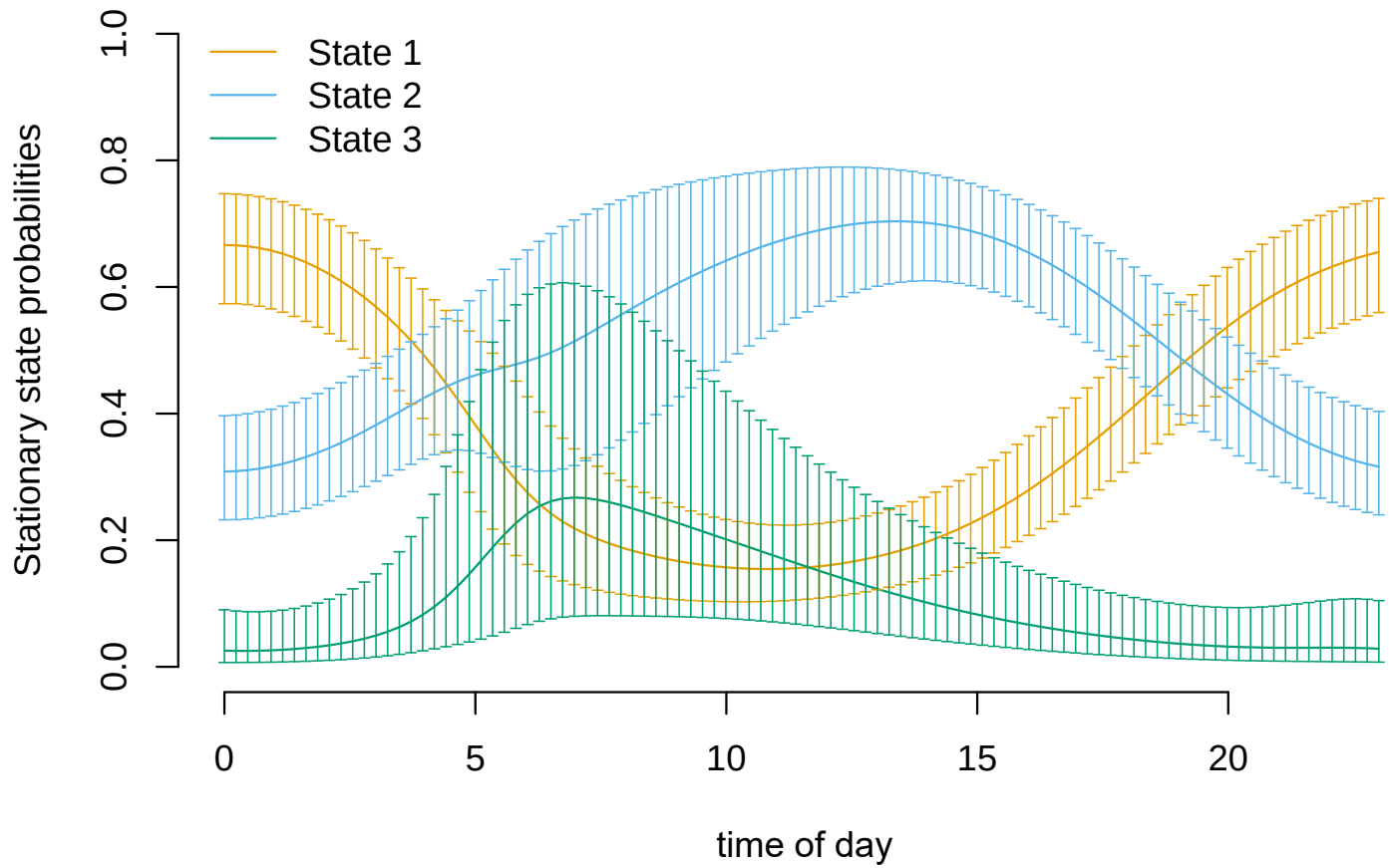


Figure: Time-of-day variation in the state process dynamics (2–state muskox model).

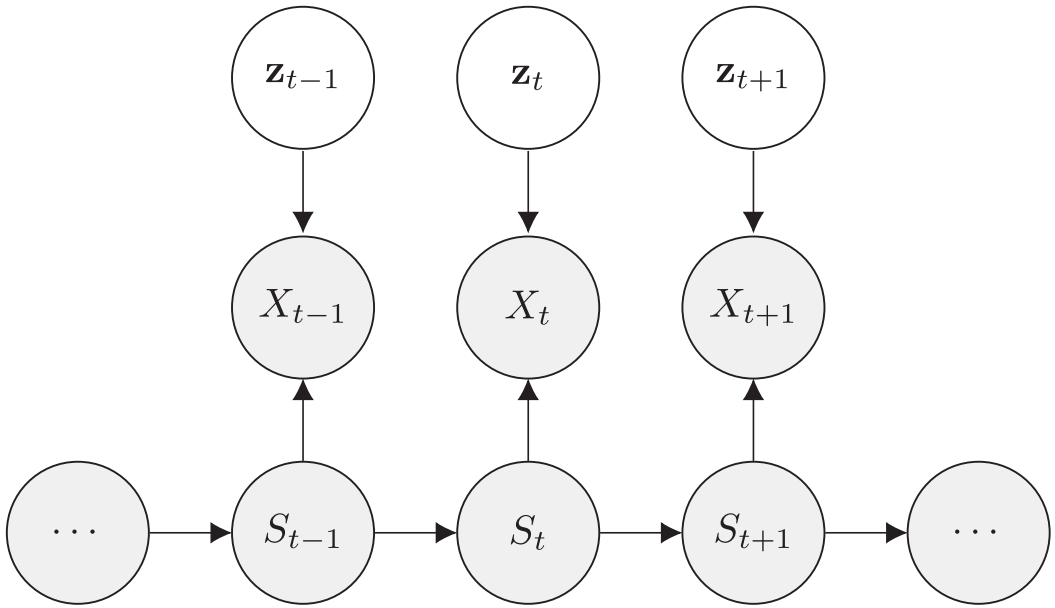
Resulting state occupancy (model with $N = 2$ states)



Resulting state occupancy (model with $N = 3$ states)



Covariates in the state-dependent process



- e.g., the state-dep. mean step length could depend on age or time of year
- as a consequence, meaning of the state may vary across covariate values
- easy to implement, but not as often seen in applications
- in econometrics commonly referred to as **Markov-switching regression**

Markov-switching regression — motivation from a regression perspective

Consider a regression scenario,

$$Y_t = \beta_0 + \beta_1 x_t + \epsilon_t,$$

where the pairs (Y_t, x_t) are observed *over time*, i.e. the index t refers to time[8].

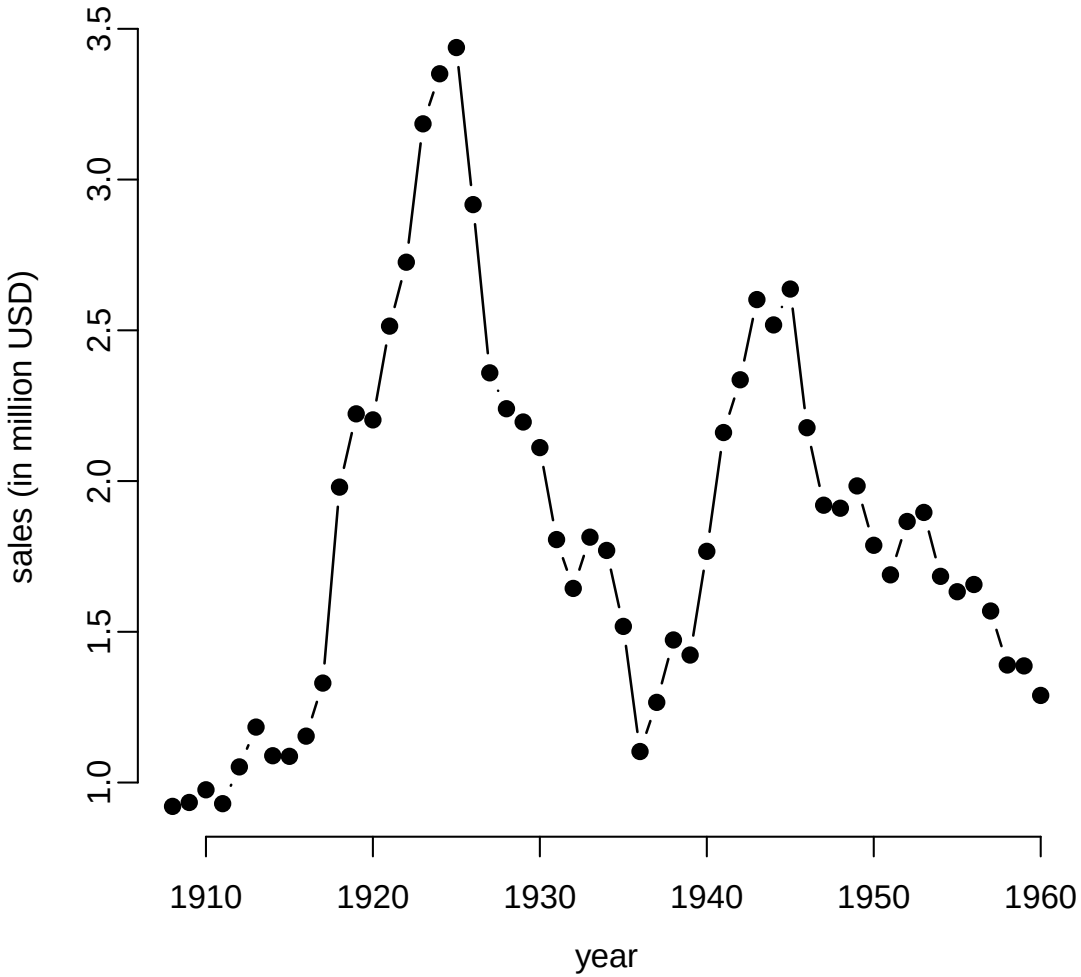
In such a context, there is often temporal correlation in the data, which can render simple regression models invalid ($\rightsquigarrow$ correlated errors).

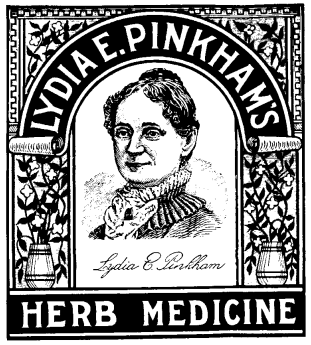
A possible reason: the regression coefficients, β_0 and β_1, might change over time.

Classic example: economic time series where the effect of an explanatory variable x_t may differ between times of high and low economic growth.

[8] e.g. days/months/years

Example Lydia Pinkham sales





Key features of the Lydia Pinkham data:

1. sales figures were strongly driven by advertising expenditure
2. sales figures in year t tended to be similar to those in year $t-1$
3. advertising strategy was changed several times (more on this later)

A simple regression model that takes 1. and 2. (but not 3.) into account:

$$\text{sales}_t = \beta_0 + \beta_1 \cdot \text{sales}_{t-1} + \beta_2 \cdot \text{advertising}_t + \sigma \cdot \epsilon_t, \quad \epsilon_t \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$$

Fitted linear model:

$$\text{sales}_t = 0.139 + 0.759 \cdot \text{sales}_{t-1} + 0.329 \cdot \text{advertising}_t + 0.225 \cdot \epsilon_t$$

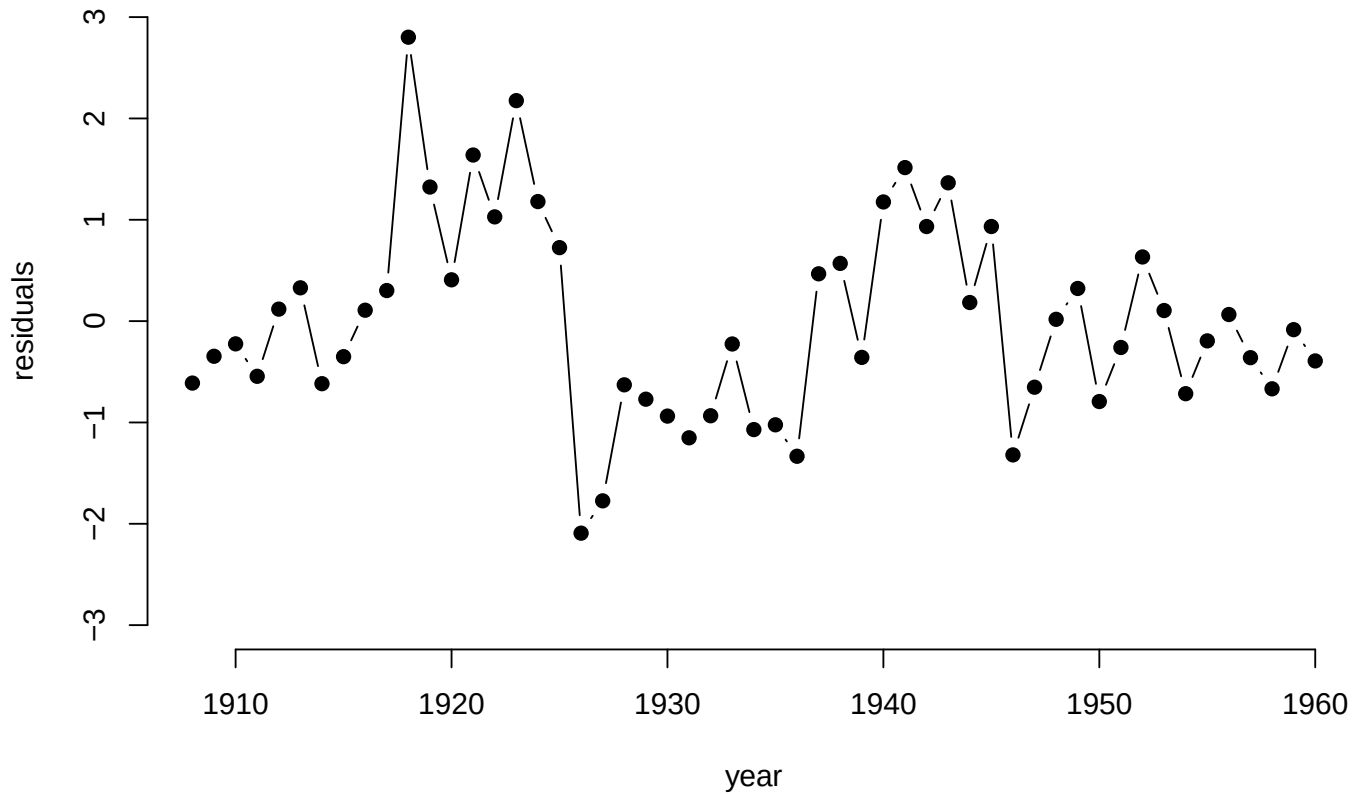


Figure: Residuals of the linear model plotted against time $\rightsquigarrow$ strongly serially correlated.

Unsurprisingly, the simple regression model fails to capture the temporal correlation of the observations.

In the example, there seem to be major structural breaks in the time series.

Thus, consider instead a **Markov-switching regression model**, where the linear model changes when there is a switch in an underlying state process:

$$\text{sales}_t = \beta_0^{(s_t)} + \beta_1^{(s_t)} \cdot \text{sales}_{t-1} + \beta_2^{(s_t)} \cdot \text{advertising}_t + \sigma^{(s_t)} \cdot \epsilon_t,$$

with s_t denoting the state of an unobserved 2–state Markov chain.

Fitted Markov-switching model:

$$\text{sales}_t = \begin{cases} 0.693 + 0.434 \cdot \text{sales}_{t-1} + 0.747 \cdot \text{advertising}_t + 0.121 \cdot \epsilon_t & \text{when } s_t = 1; \\ 0.309 + 0.562 \cdot \text{sales}_{t-1} + 0.397 \cdot \text{advertising}_t + 0.103 \cdot \epsilon_t & \text{when } s_t = 2. \end{cases}$$

$$\hat{\mathbf{\Gamma}} = \begin{pmatrix} 0.841 & 0.159 \\ 0.047 & 0.953 \end{pmatrix}$$

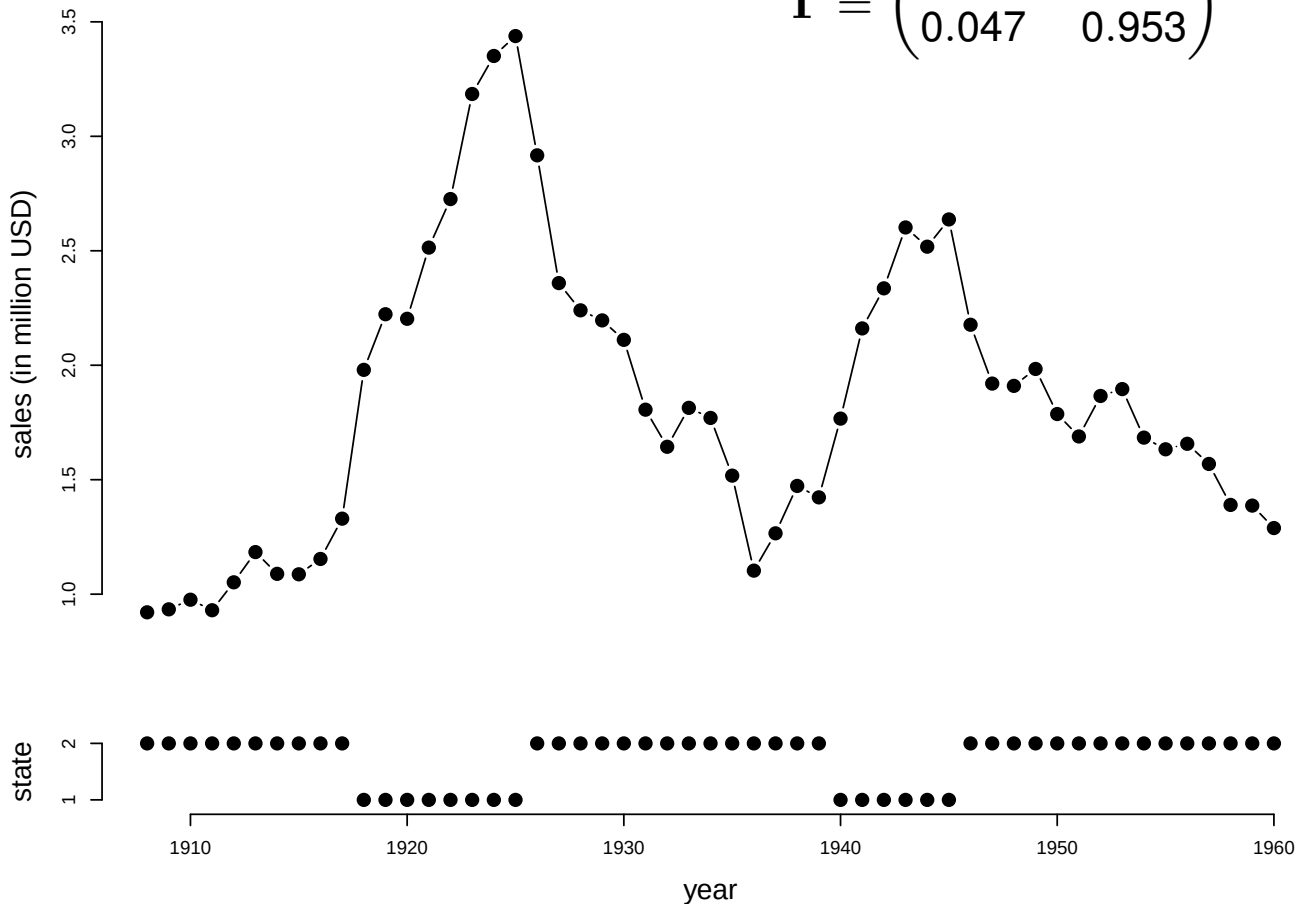


Figure: Sales figures and states decoded under the Markov-switching model.

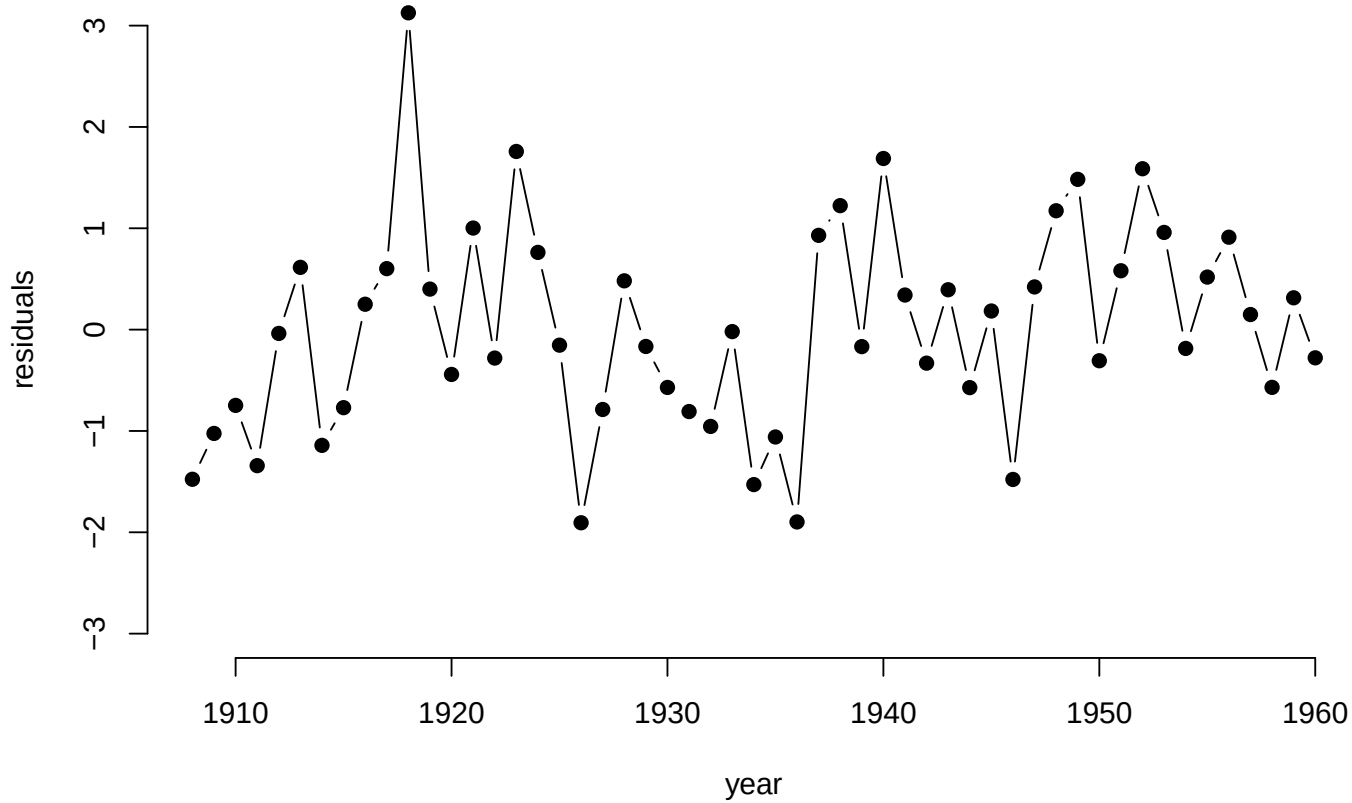


Figure: Residuals of the Markov-switching model $\rightsquigarrow$ only minor serial correlation.

Remarks on the fitted Markov-switching model:

- $\beta_2^{(1)} = 0.747$
- $\beta_2^{(2)} = 0.397$

$\rightsquigarrow$ advertising much more effective in state 1

- the actual history:
  - around 1915, beginning of successful marketing as “remedy for female troubles”
  - in 1925, a court ordered to stop this kind of advertising (“vegetable tonic” was the new, much less successful label being used in subsequent years)
  - from 1940, the previous marketing strategy was allowed to be used again

$\rightsquigarrow$ clear interpretation of the HMM states

$\rightsquigarrow$ Viterbi sequence nicely aligned with actual history of Lydia Pinkham

Some summary remarks on covariates in HMMs

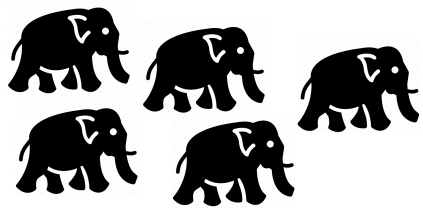
- in many real-data applications, the ability to include covariates into the model formulation is of crucial importance
- the flexibility of HMMs is here both a blessing[9] and a curse[10]
- in most applications, including covariates in the state process is preferable over the inclusion in the state-dep. process (better interpretability)

[9] lots of options to tailor the model to the particular data/question at hand

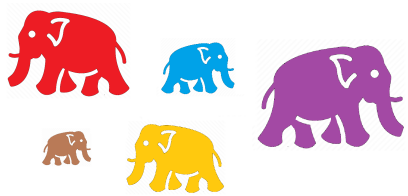
[10] it can be difficult to choose from the large variety of possible model formulations

Addressing heterogeneity when tracking multiple animals

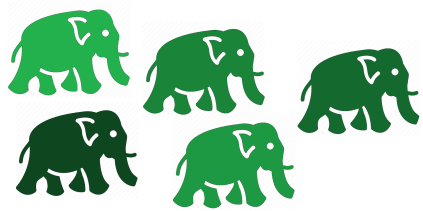
- **complete pooling**:
  - all individuals are assumed to follow the same data-generating process
  - ignores potential heterogeneity
  - can be misleading & can invalidate inference when individuals are very different



- **no pooling**:
  - model the individuals separately, with no parameters shared across them
  - less than ideal # observations / parameter ratio
  - resulting models most likely incommensurable



- **partial pooling**:
  - some — but not all — of the parameters are constant across individuals
  - compromise between the extremes above
  - individual-specific parameters modelled as functions of covariates or as random effects



Different strategies for partial pooling

(a) estimate, for each subject-specific parameter, one value for each individual
(easy to fit, but still relatively many & unstructured parameters)

(b) model parameters as functions of **individual-specific covariates** (e.g. age)
(easy to implement & helps to understand the source of the heterogeneity)

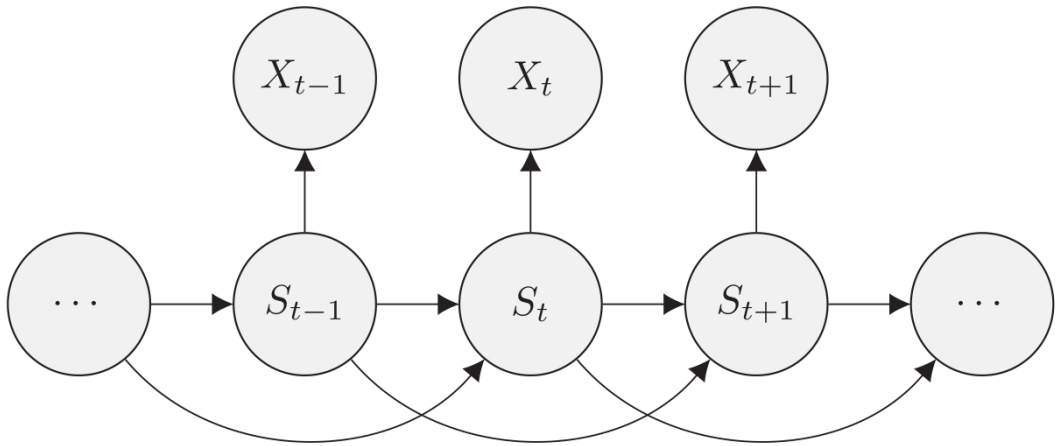
(c) assume parameters to be **random effects**, i.e. that they are drawn from a distribution common to all individuals, with one realisation per individual
(parsimonious in terms of numbers of parameters, but difficult to fit)

We often want to (or need to) implement c), but it's relatively hard!

3.2 Alternative & advanced dependence structures

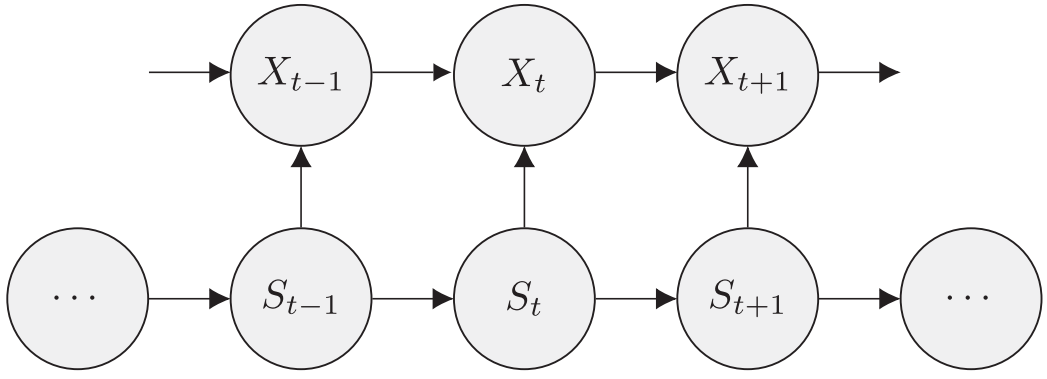
Higher-order state processes

- Markov property can be unrealistic in practice, a common criticism
- conceptually it's easy to allow for **higher-order Markov state processes**
- Markov chain of second order:



- model complexity and comp. effort increase very rapidly
- associated models are very difficult to interpret
- hardly ever used in applications

Autoregressive structures in the state-dependent process



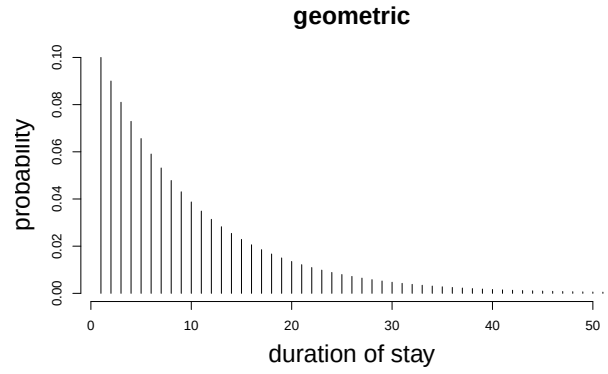
Potentially useful for high-resolution data, where conditional independence assumption will be violated — though practical relevance is unclear.

Creates no difficulties for inference: x_{t-1} is treated just like any other covariate in the state-dependent distribution of X_t.

Hidden semi-Markov models

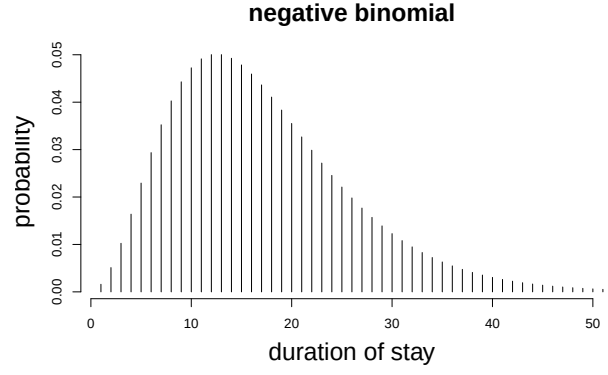
- in a basic HMM, the duration of a stay within a state is necessarily geometrically distributed

  $\rightsquigarrow$ mode is 1 (very often unrealistic!)



- **hidden semi-Markov models** relax this restrictive condition: any distribution on the positive integers can be modelled

  $\rightsquigarrow$ e.g. negative binomial



$\rightsquigarrow$ of interest primarily when focus lies on state-switching dynamics

An N–state hidden semi-Markov model is defined by specifying:

1. N distributions on the positive integers describing how long $\{S_t\}$ stays in any given state (called the state dwell-time distributions)
2. the conditional state trans. prob., given the current state is left:

$$\Pr\big(S_{t+1} = j \mid S_t = i, S_{t+1} \neq i\big), \quad i, j = 1, \ldots, N, \; i \neq j$$

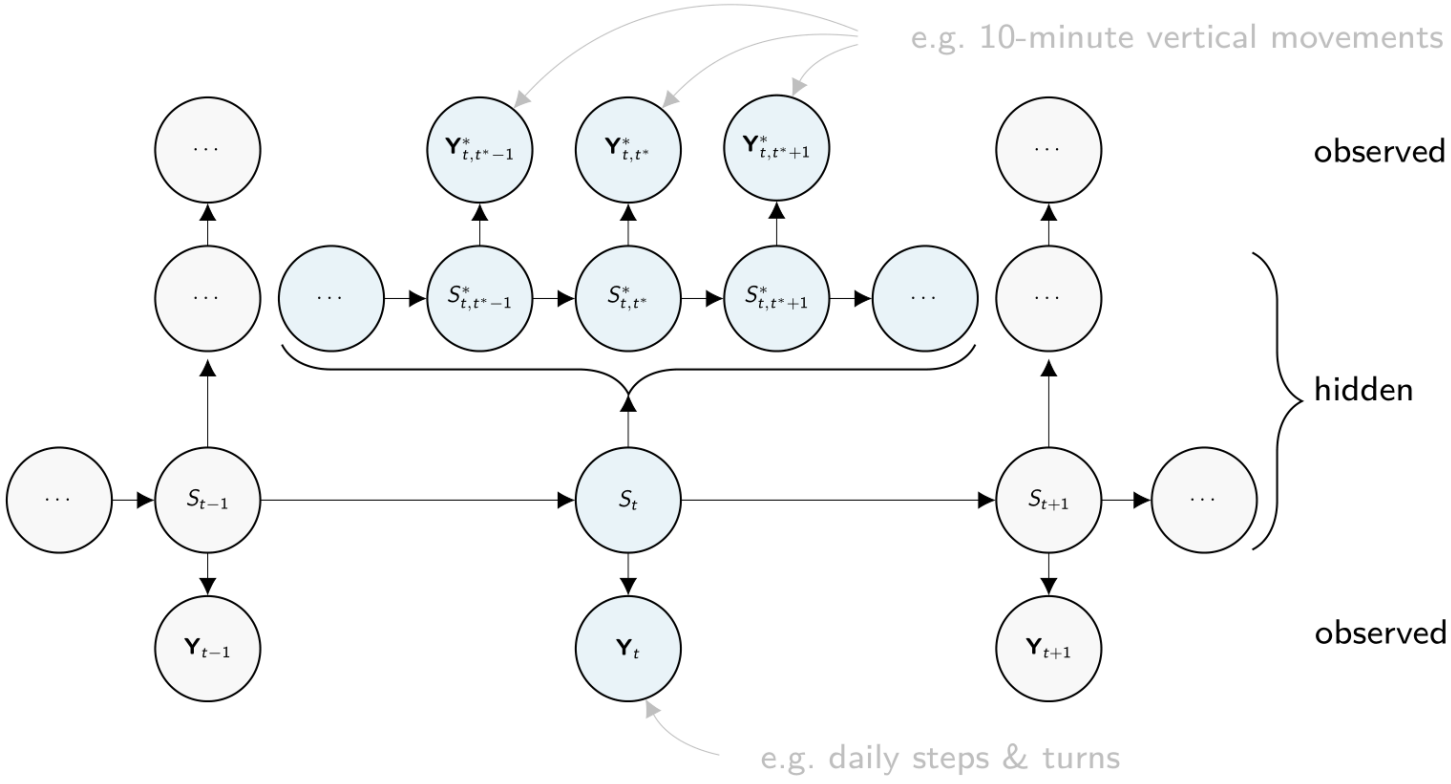
3. the state-dependent distributions

$\rightsquigarrow$ much more flexible than standard HMMs, yet very parsimonious in terms of the number of parameters (N additional parameters if neg. binom. is used)

$\rightsquigarrow$ of interest primarily when focus lies on state-switching dynamics

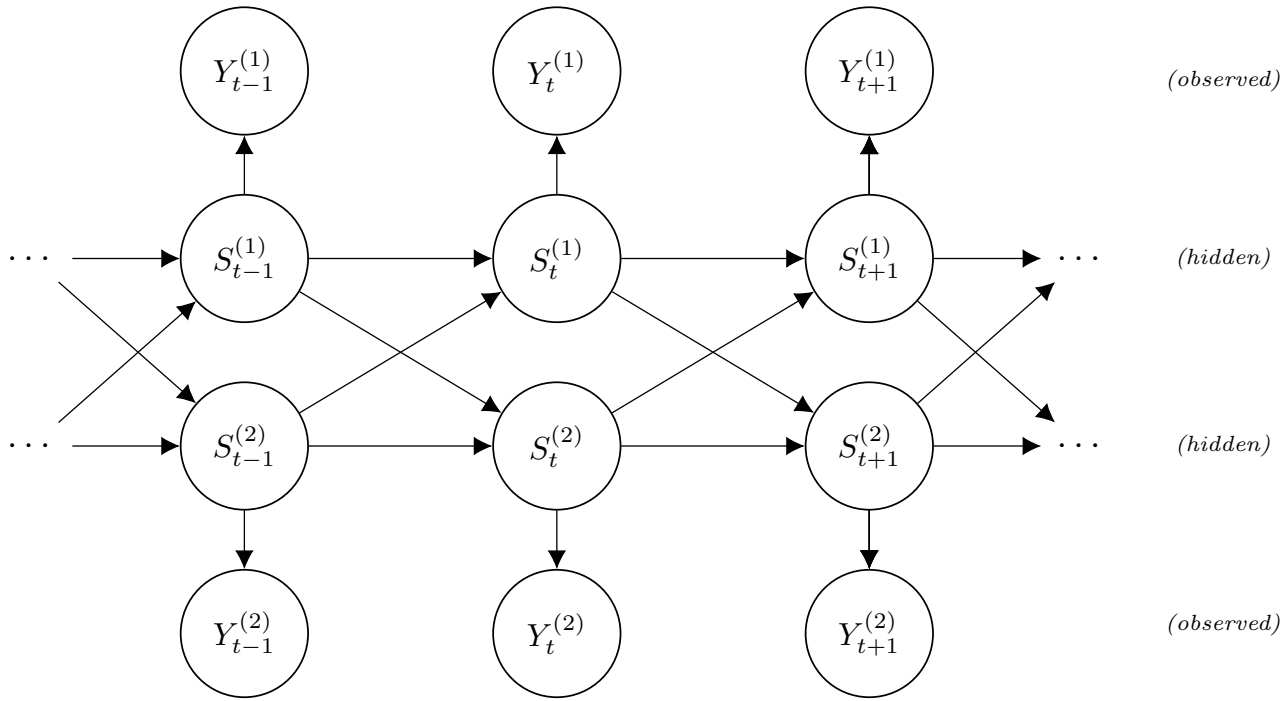
$\rightsquigarrow$ estimation is more challenging and takes some time to implement, by writing HSMM as an HMM with extended state space

Hierarchical HMMs



- allows joint modelling of multi-scale data
- statistical inference is relatively straightforward (in theory anyway...)
- lots of difficult modelling decisions, inference can be very unstable

Coupled HMMs



- ⇝ conceptually appealing for modelling interactions
- ⇝ relatively straightforward to implement
- ⇝ but number of parameters quickly explodes

3.3 Continuous-valued state processes

Motivation

- a defining property of HMMs is that their state process takes only a finite number of values
- in some cases,
  - the choice of the number of states is relatively straightforward...
  - ...and the interpretation of the states is intuitive
- however, in general, both can be difficult
- additional problem: the number of parameters increases rapidly as the number of states increases ($N^2 - N$ for the t.p.m. alone)

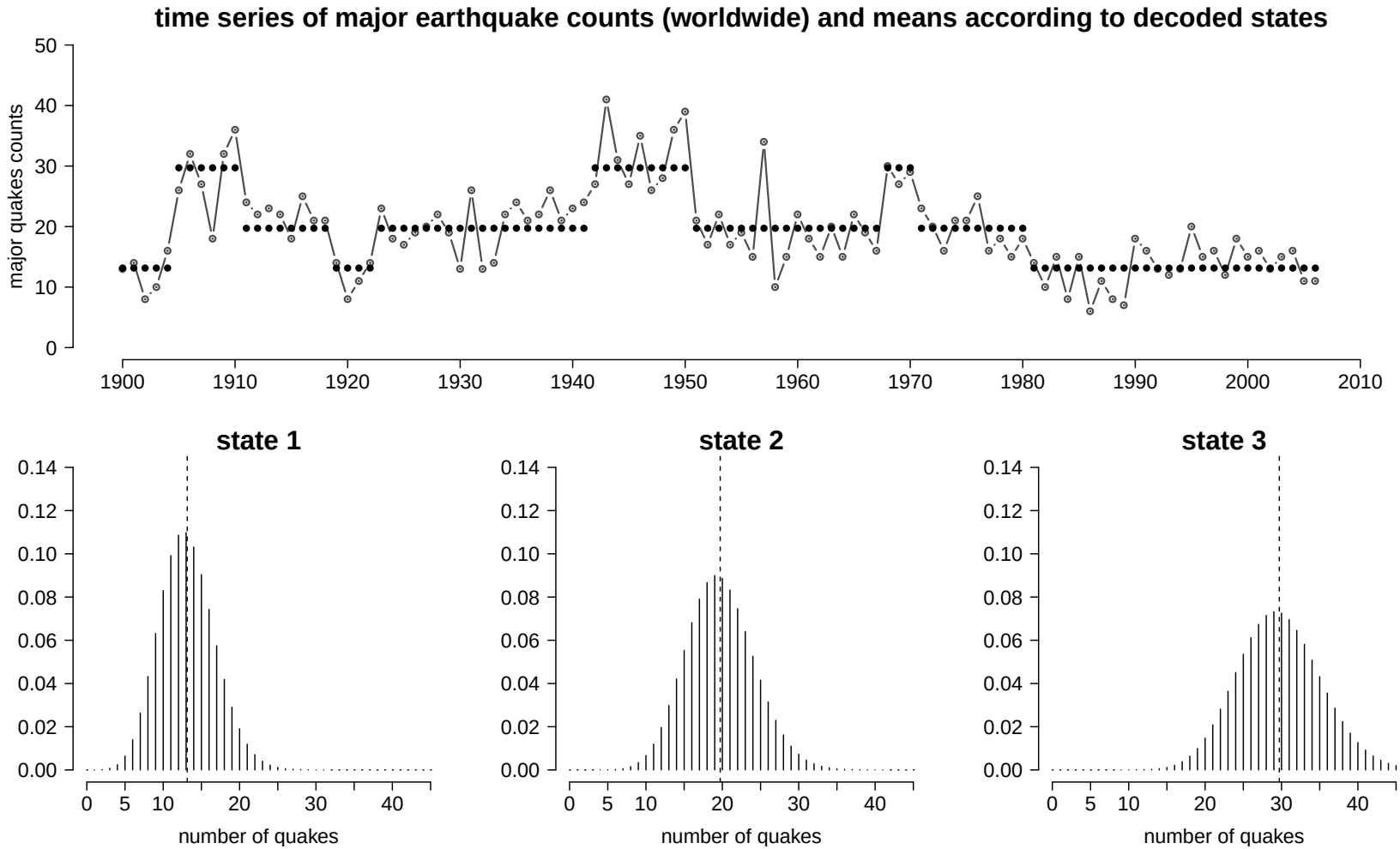
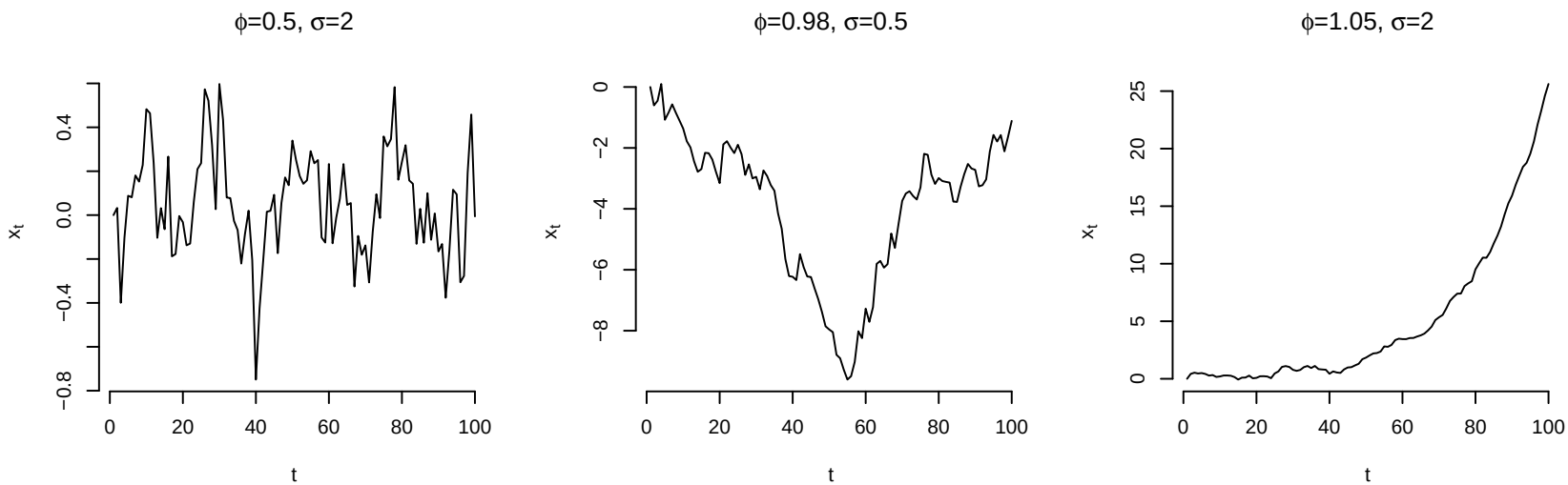


Figure: 3–state Poisson HMM fitted to time series of annual counts of major earthquakes.

- probably no seismological grounds for assuming finitely many states
- more intuitive: assume that the rate of occurrence of major earthquakes is continuous-valued, so that *gradual* change over the years is possible
- we could use an autoregressive process (of order 1) to model those rates:

$$S_t = \phi S_{t-1} + \sigma \eta_t,$$

with $|\phi| < 1$ (otherwise non-stationary), $\sigma > 0$, $\eta_t \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$



Continuous-valued state processes — a possible model specification

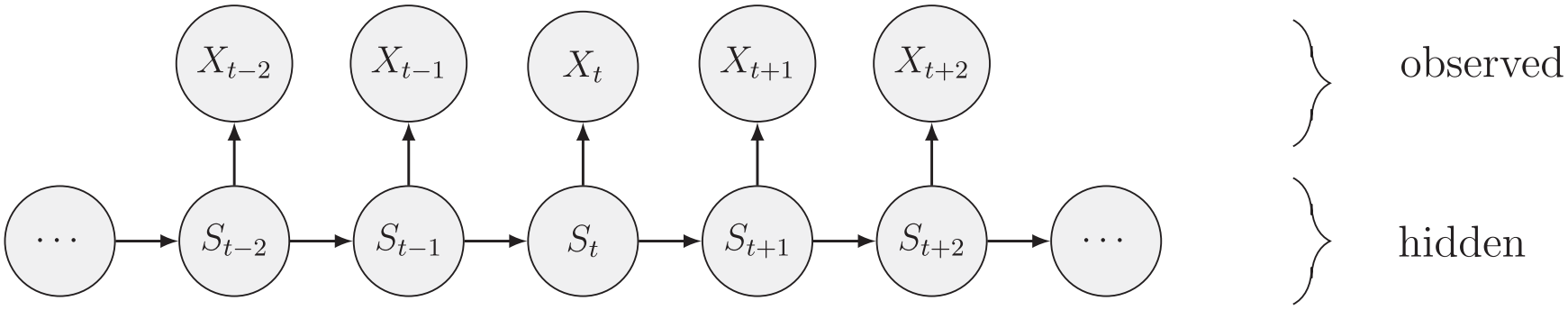
A simple model allowing for gradual change:

$$S_t = \phi S_{t-1} + \sigma \eta_t$$

$$X_t \sim \text{Poisson}\left(\beta e^{S_t}\right)$$

- $\{S_t\}$ determines occurrence rate of earthquakes (with only two parameters!)
- $\{S_t\}$ fluctuates around zero, such that the mean of $\{X_t\}$ fluctuates around β
- ϕ controls the strength of the mean-reverting effect
- σ controls the variability of the occurrence rates
- this is an example of a **state-space model** (SSM)

State-space models — general formulation



- an SSM is a doubly stochastic process in discrete time, with
  - an unobserved state process $S_1, S_2, \ldots, S_T$ (typically continuous-valued)
  - and an observed state-dependent process $X_1, X_2, \ldots, X_T$,
- such that
  - $f(x_t \mid s_1, \ldots, s_t, x_1, \ldots, x_{t-1}) = f(x_t \mid s_t)$
    (conditional independence assumption)
  - $f(s_t \mid s_1, \ldots, s_{t-1}) = f(s_t \mid s_{t-1})$
    (Markov property)
- an HMM is in fact a special case of an SSM where the state space is finite

State-space models — likelihood evaluation

For continuous-valued state processes, the likelihood is

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}) &= f(x_1, \ldots, x_T) \\
&= \int \ldots \int f(x_1, \ldots, x_T, s_1, \ldots, s_T) \, \mathrm{d}s_T \ldots \mathrm{d}s_1 \\
&= \int \ldots \int f(x_1, \ldots, x_T \mid s_1, \ldots, s_T) f(s_1, \ldots, s_T) \, \mathrm{d}s_T \ldots \mathrm{d}s_1 \\
&= \int \ldots \int f(s_1) f(x_1 \mid s_1) \prod_{t=2}^{T} f(s_t \mid s_{t-1}) f(x_t \mid s_t) \, \mathrm{d}s_T \ldots \mathrm{d}s_1
\end{aligned}$$

Analogous derivation and structure as for HMMs. But now we are dealing with T integrals instead of T sums...

Consider the innermost integral,

$$\int \underbrace{f(s_T \mid s_{T-1}) f(x_T \mid s_T)}_{\stackrel{\text{def}}{=} g(s_T)} \, \mathrm{d}s_T$$

A simple **midpoint quadrature** gives the approximation

$$\int g(s_T) \, \mathrm{d}s_T \approx \sum_{i=1}^{m} h \, g(b_i^*) = \sum_{i=1}^{m} h \, f(b_i^* \mid s_{T-1}) f(x_T \mid b_i^*),$$

where $b_1^*, \ldots, b_m^*$ are the midpoints of the intervals $[b_{i-1}, b_i]$, $i = 1, \ldots, m$, all of length $h = (b_m - b_0)/m$.

The approximation will be accurate if

- m is large

  and

- g is effectively zero outside the interval $[b_0, b_m]$.

By repeated application of midpoint quadrature, we obtain the following approximation of the likelihood:

$$\mathcal{L}(\boldsymbol{\theta}) = \int \dots \int f(s_1) f(x_1 \mid s_1) \prod_{t=2}^{T} f(s_t \mid s_{t-1}) f(x_t \mid s_t) \, \mathrm{d}s_T \dots \mathrm{d}s_1$$

$$\approx h^T \sum_{i_1=1}^{m} \dots \sum_{i_T=1}^{m} f(b^*_{i_1}) f(x_1 \mid b^*_{i_1}) \prod_{t=2}^{T} f(b^*_{i_t} \mid b^*_{i_{t-1}}) f(x_t \mid b^*_{i_t}) = \mathcal{L}_{\text{approx}}(\boldsymbol{\theta})$$

- ⇝ there are m^T summands (and both m and T are large!)
- ⇝ but the structure of $\mathcal{L}_{\text{approx}}(\boldsymbol{\theta})$ is *identical* to that of a standard HMM
- ⇝ numerical integration corresponds to **discretisation of the state space** — we're approximating the SSM by an HMM with very many (namely m) states

We want to apply the forward algorithm, so we define:

- the i–th component of the m–dimensional vector $\boldsymbol{\delta}^{(1)}$ to be $\delta_i = h\, f(b_i^*)$
  (the approximate probability of S_1 falling in the interval $[b_{i-1}, b_i]$)
- an $m \times m$ matrix $\boldsymbol{\Gamma} = \left(\gamma_{ij}\right)$ by specifying $\gamma_{ij} = h\, f(b_j^* | b_i^*)$
  (the approximate probability of S_t falling into the interval $[b_{j-1}, b_j]$, given that S_{t-1} is in the interval $[b_{i-1}, b_i]$)
- the diagonal matrix $\mathbf{P}(x_t)$ to be the $m \times m$ diagonal matrix with i–th diagonal entry equal to $f(x_t \mid b_i^*)$
  (an approximation of the density of x_t, given that $S_t \in [b_{i-1}, b_i]$)

Putting all the pieces together, we can rewrite the approximate likelihood as:

$$\mathcal{L}_{\text{approx}}(\boldsymbol{\theta}) = \boldsymbol{\delta}^{(1)} \mathbf{P}(x_1) \boldsymbol{\Gamma} \mathbf{P}(x_2) \boldsymbol{\Gamma} \mathbf{P}(x_3) \cdots \boldsymbol{\Gamma} \mathbf{P}(x_{T-1}) \boldsymbol{\Gamma} \mathbf{P}(x_T) \mathbf{1}^t$$

Overview and technical remarks

- numerical max. of $\mathcal{L}_{\text{approx}}(\boldsymbol{\theta})$ is feasible for large T and fairly large m
- the choices of m and $[b_0, b_m]$ control the accuracy of the approximation
- m needs to be large to provide a good approximation, and $[b_0, b_m]$ should be neither too narrow nor too wide
- note that the number of model parameters does not depend on m — the entries of the $m \times m$ matrix $\boldsymbol{\Gamma}$ depend only on the parameters of $\{S_t\}$
- all other HMM tools, e.g. Viterbi, are applicable

Earthquake example — implementation in R

```
mllk <- function(theta.star, x, m, bm){
 phi <- plogis(theta.star[1])
 sigma <- exp(theta.star[2])
 beta <- exp(theta.star[3])
 b <- seq(-bm, bm, length = m + 1)          # specify boundaries of m intervals
 h <- b[2] - b[1]                           # h is the length of each interval
 bstar <- (b[-1] + b[-(m + 1)]) * 0.5       # midpoints of the m intervals
 Gamma <- matrix(0, m, m)
 for (i in 1:m){
  Gamma[i, ] <- h * dnorm(bstar, phi * bstar[i], sigma)  # m*m t.p.m. of the approx. HMM
 }
 delta <- h * dnorm(bstar, 0, sigma / sqrt(1 - phi^2))   # stat. initial distribution
 foo <- delta * dpois(x[1], exp(bstar) * beta)
 l <- log(sum(foo))
 phi <- foo / sum(foo)
 for (t in 2:length(x)){
  foo <- phi %*% Gamma * dpois(x[t], exp(bstar) * beta)
  l <- l + log(sum(foo))
  phi <- foo / sum(foo)
 }
 return(-l)
}

quakes <- read.table("http://www.rolandlangrock.com/Misc/earthquakes.txt", header = TRUE)

theta.star <- c(qlogis(0.8), log(0.2), log(20))
mod <- nlm(mllk, theta.star, x = quakes$count, m = 200, bm = 1.5, print.level = 2)

c(plogis(mod$estimate[1]), exp(mod$estimate[2]), exp(mod$estimate[3]))
```

The code fits the following simple SSM to the series of earthquake counts:

$$S_t = \phi S_{t-1} + \sigma \eta_t$$

$$X_t \sim \text{Poisson}\left(\beta e^{S_t}\right)$$

- $m = 200$ and $[b_0, b_m] = [-1.5, 1.5]$ were used in the approximation
- maximum likelihood estimates:

$$\widehat{\phi} = 0.89, \quad \widehat{\sigma} = 0.14, \quad \widehat{\beta} = 17.8$$

- the variance of the stationary distribution of $\{S_t\}$ is $\widehat{\sigma}^2/(1-\widehat{\phi}^2)$, approximately 0.3^2, which indicates that $[b_0, b_m]$ is sufficiently wide
- estimation takes less than a second!![11]
- AIC$=$ 670.54 (AIC of 3–state Poisson HMM: 676.92)

[11] this is remarkable — alternative approaches for fitting SSMs tend to be magnitudes slower

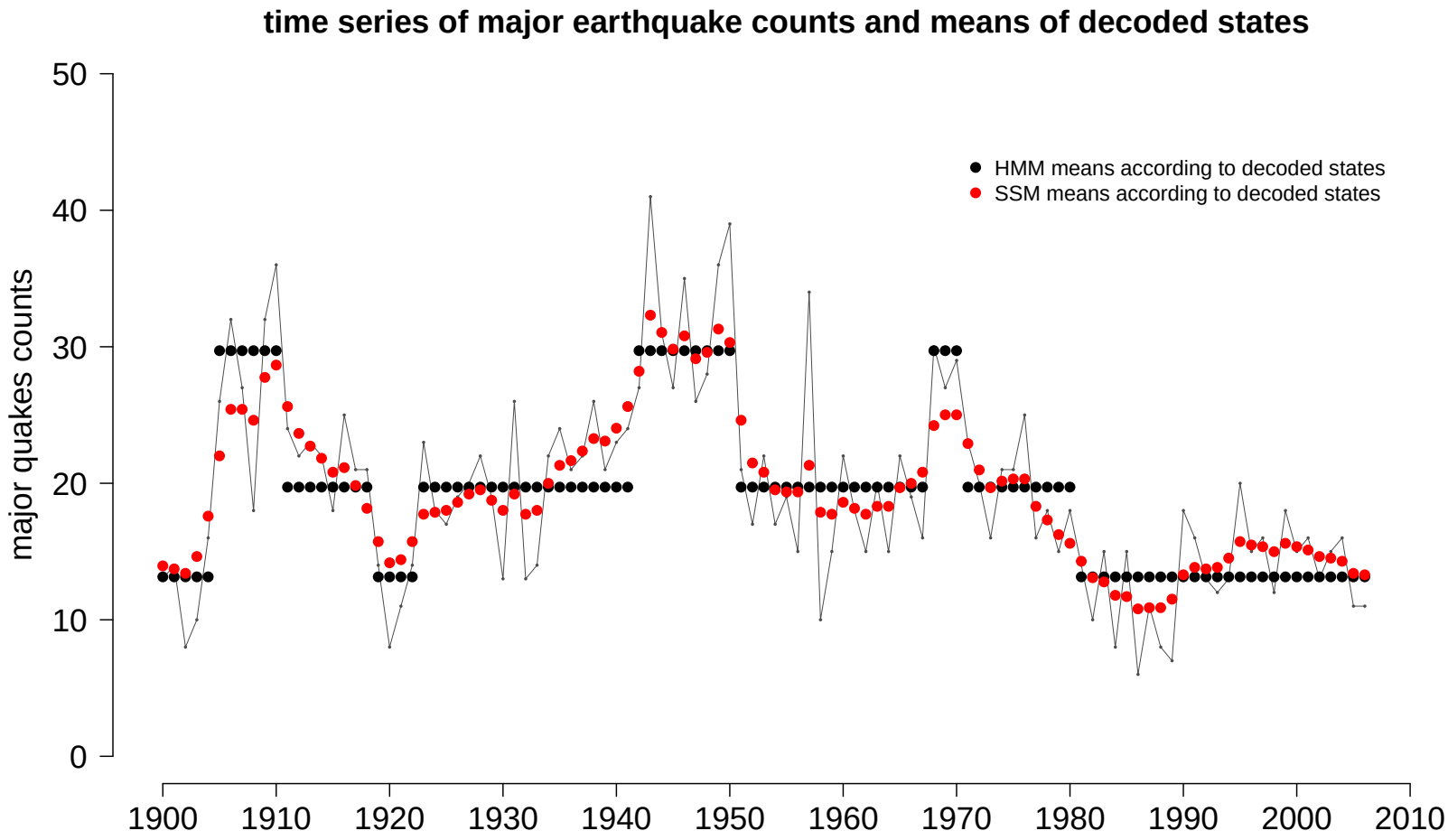
Illustration of the influence of m and $[b_0, b_m]$

Table: SSM fitted to earthquakes data: maximum log-likelihood values obtained for various values of m and $b_{\max}$, where $-b_0 = b_m = b_{\max}$.

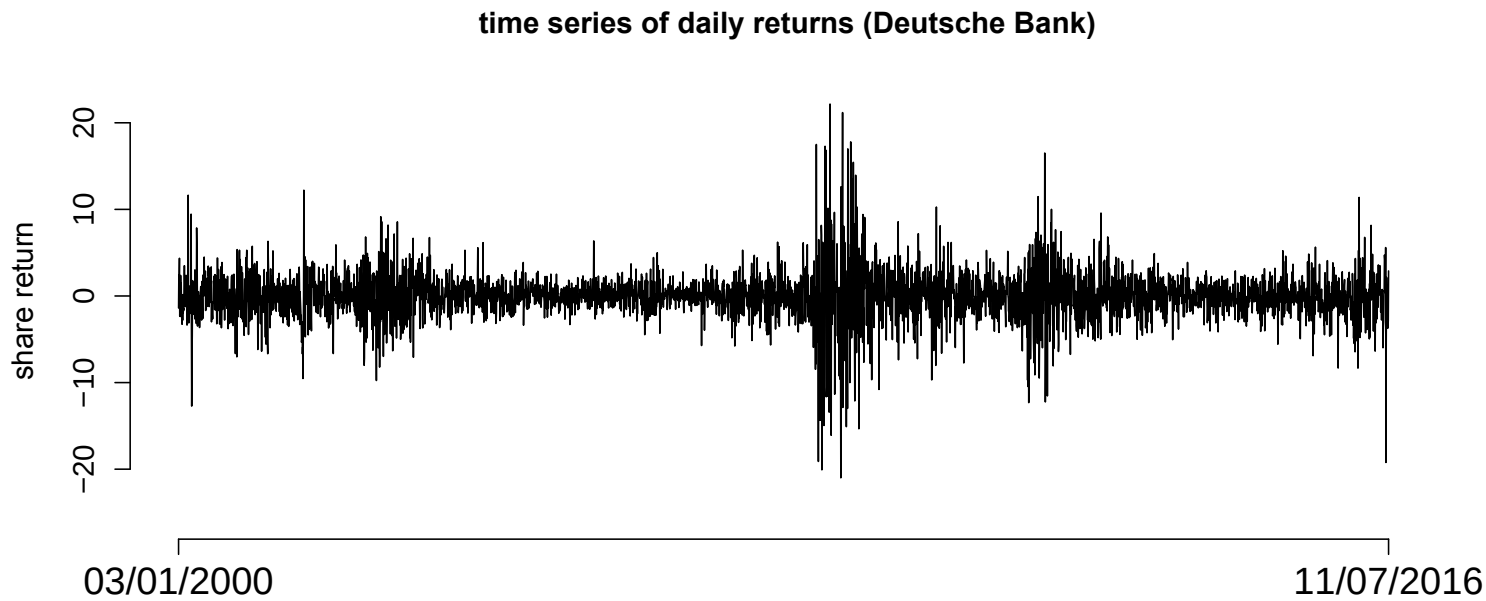
$b_{\max}$	$m = 20$	$m = 40$	$m = 70$	$m = 100$	$m = 200$
0.5	-337.61050	-337.66812	-337.68095	-337.68412	-337.68640
1	-332.26895	-332.26918	-332.26924	-332.26925	-332.26926
2	-332.21761	-332.26789	-332.26789	-332.26789	-332.26789
4	–	-332.21761	-332.26789	-332.26789	-332.26789

In this application, the likelihood approximation is virtually exact for $m = 40$, provided that the specified essential range is neither too small nor too large.

Viterbi output in the earthquake example



A second example: Deutsche Bank share returns



For modelling share returns using HMMs, we could assume that

$$X_t | S_t = j \sim \mathcal{N}(0, \sigma_j^2)$$

(such that states $\cong$ levels of nervousness/volatility of the market)

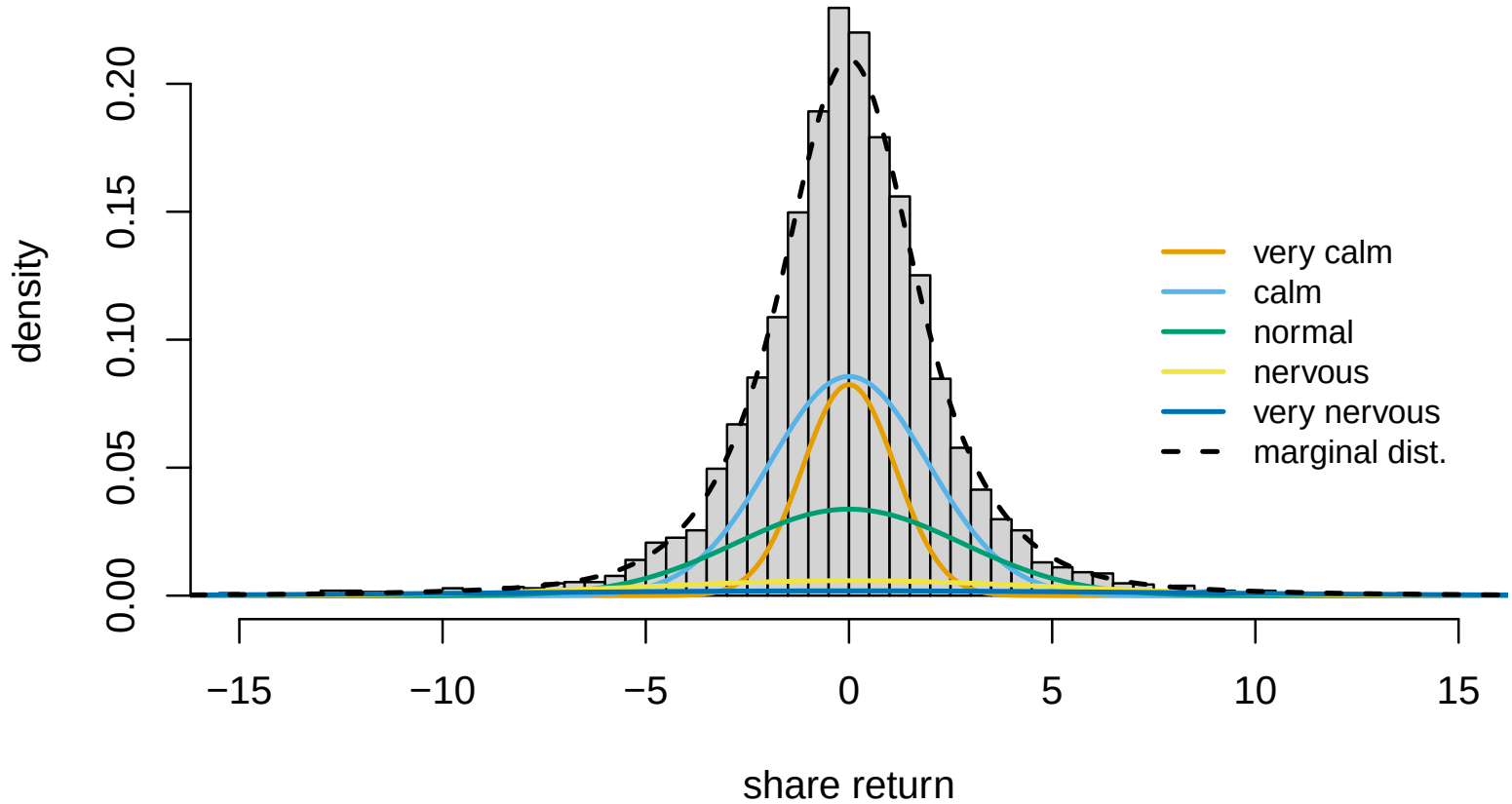


Figure: 5–state HMM fitted to Deutsche Bank share returns — displayed are the (weighted) state-dep. normal distributions, $\mathcal{N}(0, \sigma_j^2)$, for $j = 1, \ldots, 5$, and the marginal distribution.

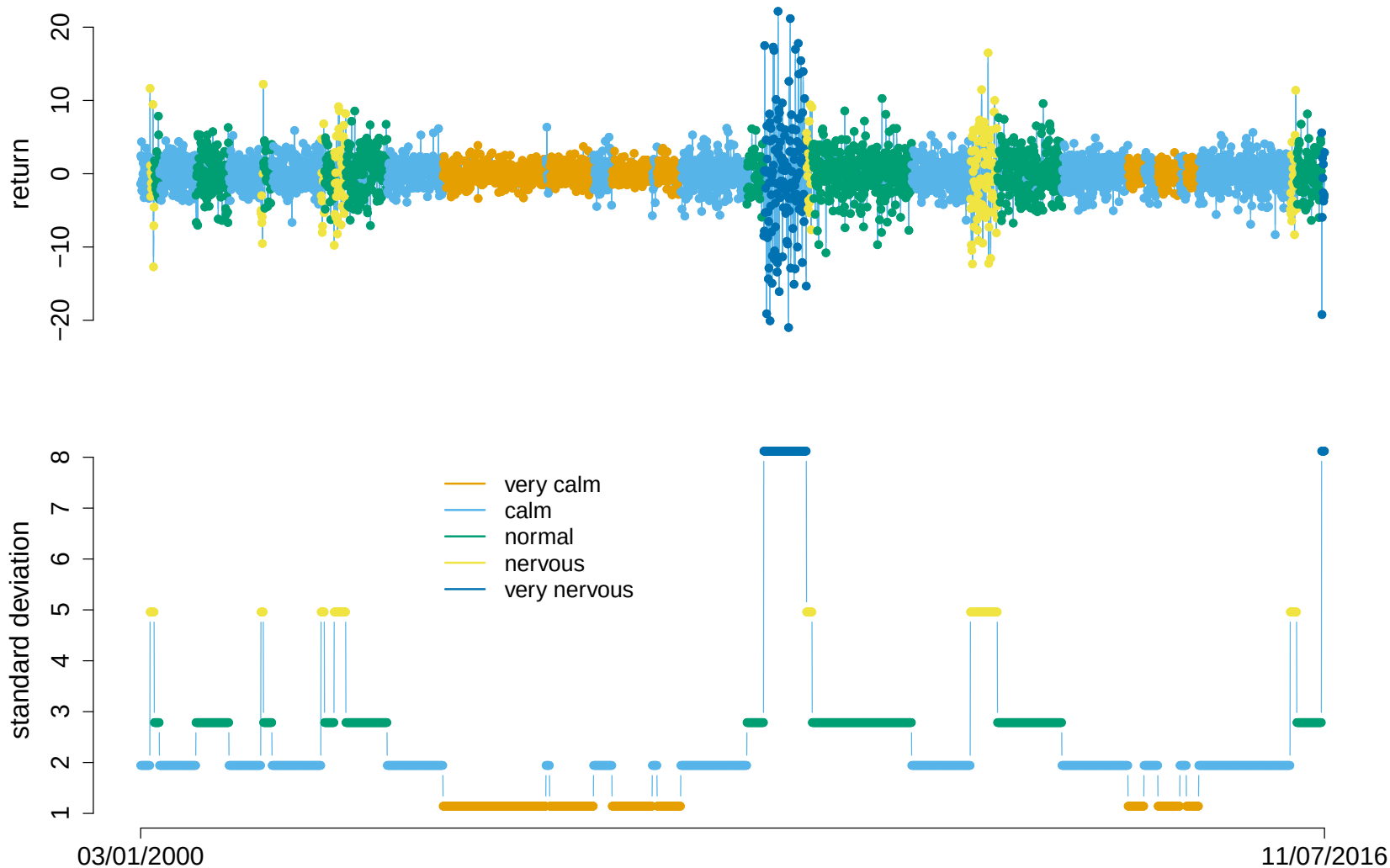


Figure: Time series of daily returns, colour-coded according to the Viterbi-decoded states (top plot), and associated “volatility” levels (bottom plot).

Continuous volatility

The assumption of discrete volatility levels seems rather unrealistic!

Idea: formulate a model where **market volatility is continuous-valued**.

We will again assume that the volatility at time t — now denoted g_t for consistency with the literature — depends only on the volatility at time $t-1$, g_{t-1}.

For example, we can again use an AR(1) process:

$$g_t = \phi g_{t-1} + \sigma \eta_t, \quad \eta_t \sim \mathcal{N}(0,1)$$

Basic **stochastic volatility (SV) model** for share returns:

$$y_t = \beta \epsilon_t e^{g_t/2}, \qquad g_t = \phi g_{t-1} + \sigma \eta_t$$

- y_t: share return on day t
- g_t: unobserved volatility, $\eta_t \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$
- $\epsilon_t \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$
- β is the baseline standard deviation of the returns (when g_t is in equilibrium)
- SV models capture most of the 'stylized facts' attributed to series of returns (no autocorrelation, correlation of squared returns, kurtosis > 3, etc.)

For the DB share returns, maximum likelihood estimation yields:

$$\hat{\beta} = 2.1164, \qquad \hat{\phi} = 0.9897, \qquad \hat{\sigma} = 0.1397$$

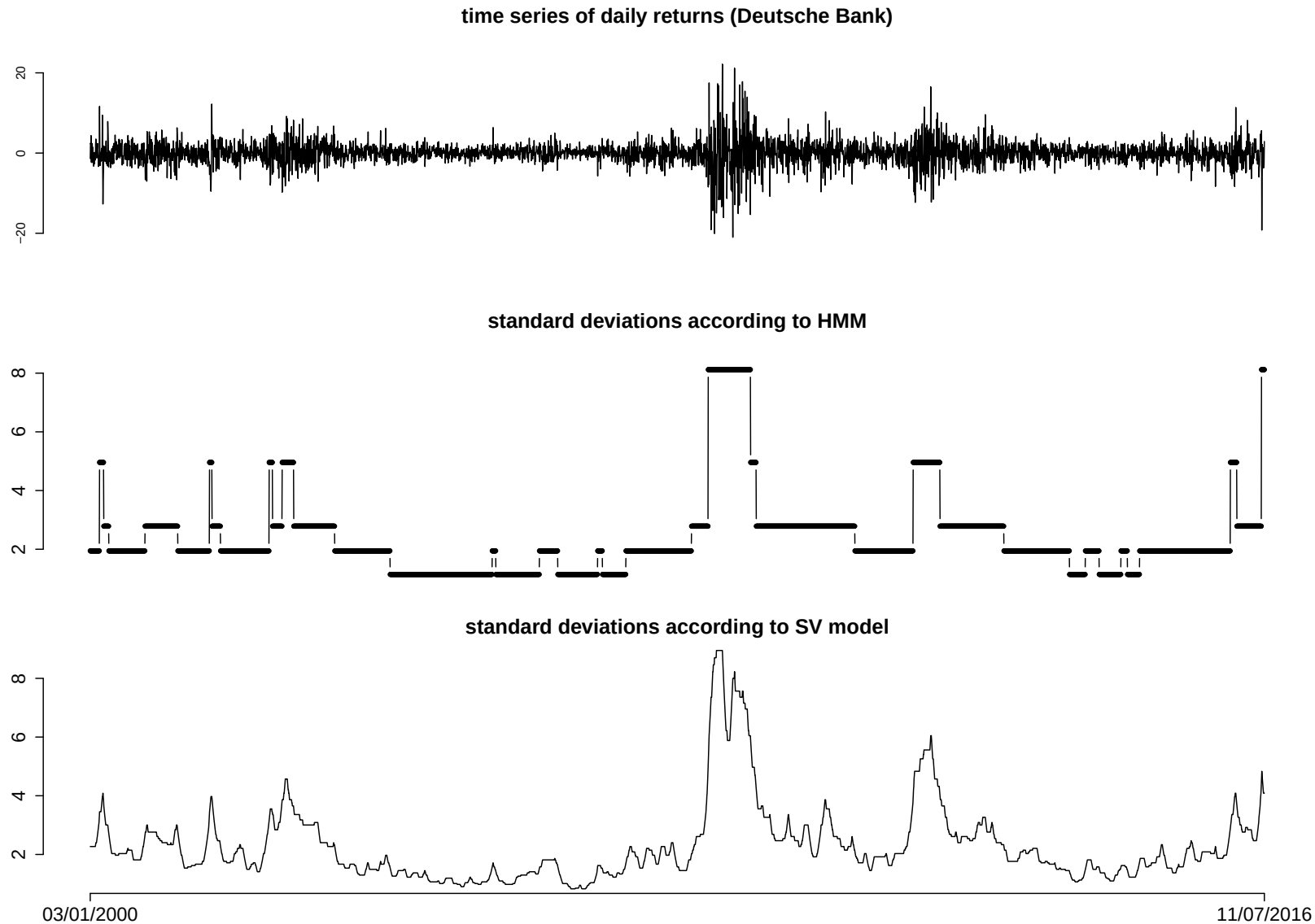


Figure: Comparison between fitted SV model and 5–state Gaussian HMM — displayed in the bottom two rows are the standard deviations conditional on the decoded states.

3.4 HMMs in continuous time

Sampling schemes and their relevance for HMMs

The **discrete-time nature** of HMMs means that there needs to be a **meaningful sampling unit** w.r.t. which t.p.m. and state-dep. distributions are interpreted:

- data collected at regular time intervals (hourly/daily/etc.) ✓
- dive-by-dive summary statistics for marine mammals ✓
- opportunistic data, e.g. collected whenever a patient visits their doctor X
- experience sampling methods, e.g. smartphone push notifications X

For irregular sampling, we may need model formulations in continuous time.

Motivating example: lung function measurements after lung transplantation

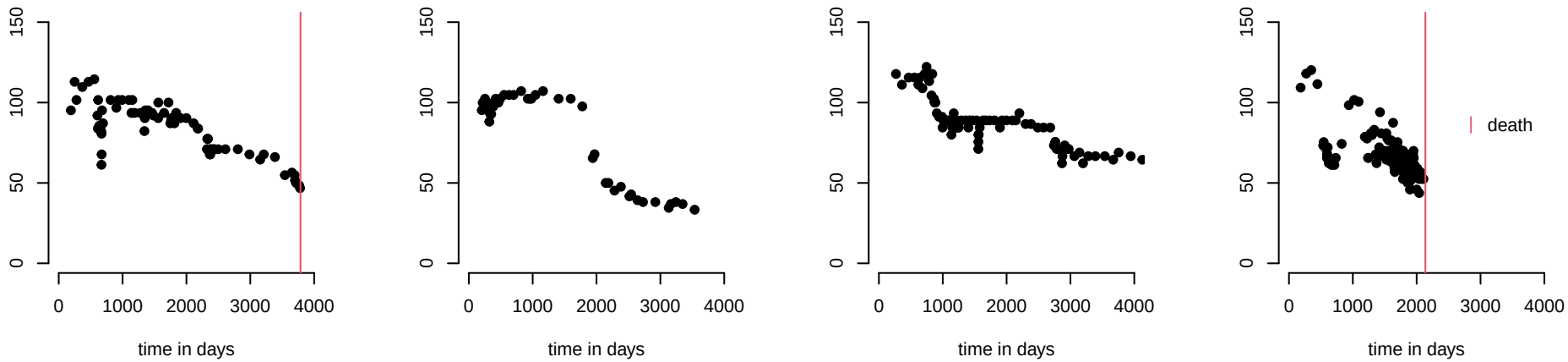


Figure: Time series for 4 out of 203 patients who had a lung transplantation — shown on the *y* axis are measurements of the forced expiratory volume (FEV).

- ⇝ doctor consultations irregularly spaced in time
- ⇝ disease progression hence needs to be modelled in continuous time
- ⇝ additional patterns:
  - deterioration of lung function largely irreversible
  - death as absorbing state

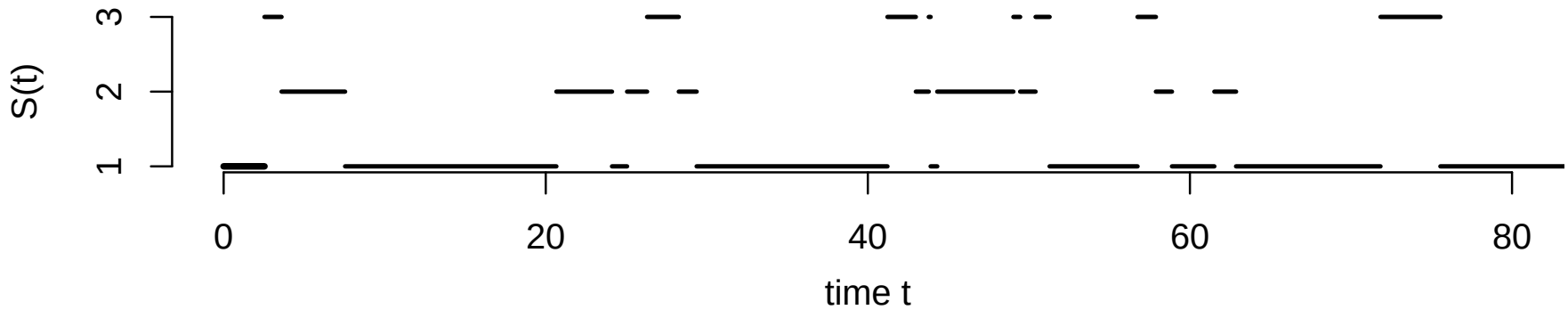
From discrete-time to continuous-time Markov chains

Discrete-time Markov chain $\{S_t, t = 1, 2, \ldots\}$ can be defined as follows:

- duration of a stay within a state is **geometrically distributed**
- at the end of such a stay, chain switches to a different state

Natural analogue $\{S_t, t \geq 0\}$ in continuous time:

- duration of a stay within a state is **exponentially distributed**
- at the end of such a stay, chain switches to a different state



Shown above is an example realisation from a continuous-time Markov chain with conditional transition probability matrix (given a state is left)

$$\boldsymbol{\Omega} = \begin{pmatrix} 0.0 & 0.3 & 0.7 \\ 0.8 & 0.0 & 0.2 \\ 0.5 & 0.5 & 0.0 \end{pmatrix}$$

and Exp(0.2), Exp(0.5) and Exp(1) dwell-time distrib. in states 1–3, respectively.

A **continuous-time Markov chain** $\{S_t, t \geq 0\}$ is a stochastic process such that

- $S_t \in \{1, \ldots, N\}$ for all $t \geq 0$ (i.e. there are N states)
- the duration of a stay in state i follows an $\text{Exp}(\lambda_i)$ distribution
- given a transition away from state i, the probability that the process enters state j is ω_{ij}, with $\omega_{ii} = 0$ and $\sum_{j \neq i} \omega_{ij} = 1$

This is one of several possible ways to define a continuous-time Markov chain — actually not the standard definition, but arguably the most intuitive one.

Infinitesimal generator matrix

- we can interpret λ_i as the *rate of transitions out of state i*
- out of those transitions, a proportion of ω_{ij} go to state j, such that $\lambda_i \cdot \omega_{ij}$ can be interpreted as the *rate of transitions from state i to state j*
- all these transition rates are summarised in the so-called **(infinitesimal) generator matrix** (also called transition intensity matrix):

$$\mathbf{Q} = \begin{pmatrix} -\lambda_1 & \lambda_1\omega_{12} & \lambda_1\omega_{13} & \dots \\ \lambda_2\omega_{21} & -\lambda_2 & \lambda_2\omega_{23} & \\ \lambda_3\omega_{31} & \lambda_3\omega_{32} & -\lambda_3 & \\ \vdots & & & \ddots \end{pmatrix} = \begin{pmatrix} q_{11} & q_{12} & q_{13} & \dots \\ q_{21} & q_{22} & q_{23} & \\ q_{31} & q_{32} & q_{33} & \\ \vdots & & & \ddots \end{pmatrix}$$

- the diagonal entries are calculated as $q_{ii} = -\sum_{j \neq i} q_{ij}$, with $q_{ij} \geq 0$ for $i \neq j$

⇝ from the transition rates q_{ij}, we can obtain both λ_i and ω_{ij}

⇝ generator matrix completely describes the dynamics of the state process

Deriving state transition probabilities from **Q**

Defining $\mathbf{P}(t_1, t_2)$ as the matrix containing the state transition probabilities over the period $[t_1, t_2]$[12], we have the important relation

$$\mathbf{P}(t_1, t_2) = e^{\mathbf{Q}\cdot(t_2 - t_1)},$$

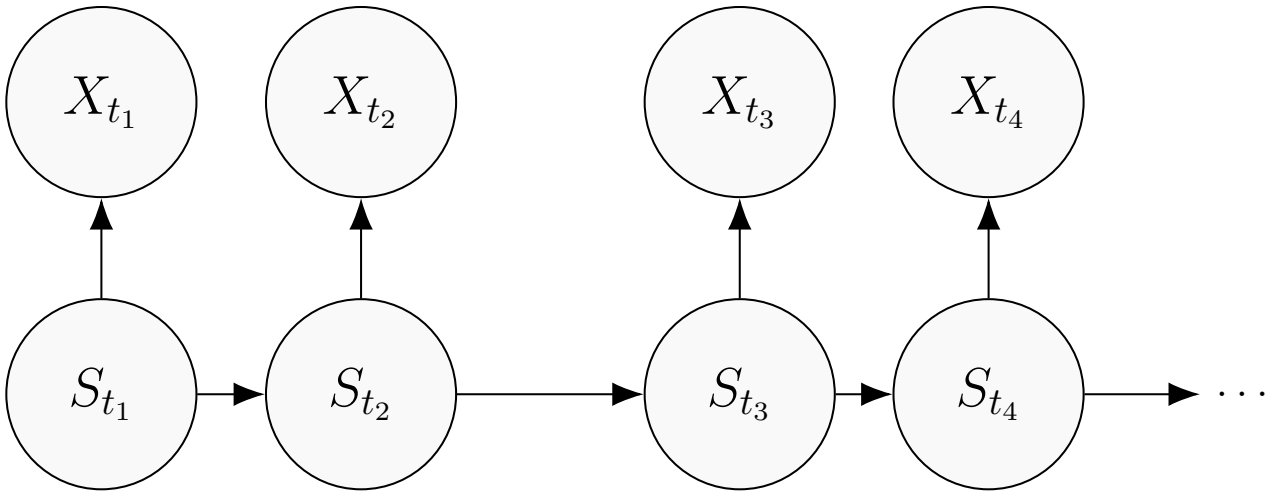
where $e^{\cdots}$ is the matrix exponential function.

Example:

$$\mathbf{Q} = \begin{pmatrix} -0.2 & 0.06 & 0.14 \\ 0.4 & -0.5 & 0.1 \\ 0.5 & 0.5 & -1 \end{pmatrix} \quad \rightsquigarrow \quad \mathbf{P}(0,5) = e^{\mathbf{Q}\cdot 5} = \begin{pmatrix} 0.70 & 0.18 & 0.12 \\ 0.65 & 0.24 & 0.11 \\ 0.66 & 0.22 & 0.12 \end{pmatrix}$$

[12] i.e. with $p_{ij}(t_1, t_2) = \Pr\left(S_{t_2} = j | S_{t_1} = i\right)$

Continuous-time HMMs



Assumptions are analogous to the discrete-time case:

- Markov property for the (continuous-time) state process
- observations are conditionally independent, given the states

Likelihood calculation using the forward algorithm

Suppose we want to calculate the likelihood of observations $x_{t_1}, x_{t_2}, \ldots, x_{t_T}$.

Consider the (continuous-time version of the) **forward variables**,

$$\alpha_z(j) = f(x_{t_1}, \ldots, x_{t_z}, s_{t_z} = j), \quad \boldsymbol{\alpha}_z = \big(\alpha_z(1), \ldots, \alpha_z(N)\big)$$

Forward algorithm (in continuous time):

$$\begin{aligned}
\boldsymbol{\alpha}_{t_1} &= \boldsymbol{\delta}^{(1)}\mathbf{P}(x_{t_1}) \\
\boldsymbol{\alpha}_{t_z} &= \boldsymbol{\alpha}_{t_{z-1}} e^{\mathbf{Q}\cdot(t_z - t_{z-1})}\mathbf{P}(x_{t_z}) \quad \text{for } z = 2, \ldots, T
\end{aligned}$$

Resulting closed-form expression for the likelihood:

$$\mathcal{L}(\boldsymbol{\theta}) = \boldsymbol{\delta}^{(1)}\mathbf{P}(x_{t_1})e^{\mathbf{Q}\cdot(t_2-t_1)}\mathbf{P}(x_{t_2})e^{\mathbf{Q}\cdot(t_3-t_2)}\mathbf{P}(x_{t_3}) \cdot \ldots \cdot e^{\mathbf{Q}\cdot(t_T-t_{T-1})}\mathbf{P}(x_{t_T})\mathbf{1}^t$$

Fitted 4–state CT-HMM in the lung transplantation example

$$\hat{\mathbf{Q}} = \begin{pmatrix} -0.0010 & 0.0009 & 0.0000 & 0.0001 \\ 0.0000 & -0.0012 & 0.0010 & 0.0002 \\ 0.0000 & 0.0000 & -0.0013 & 0.0013 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 \end{pmatrix}$$

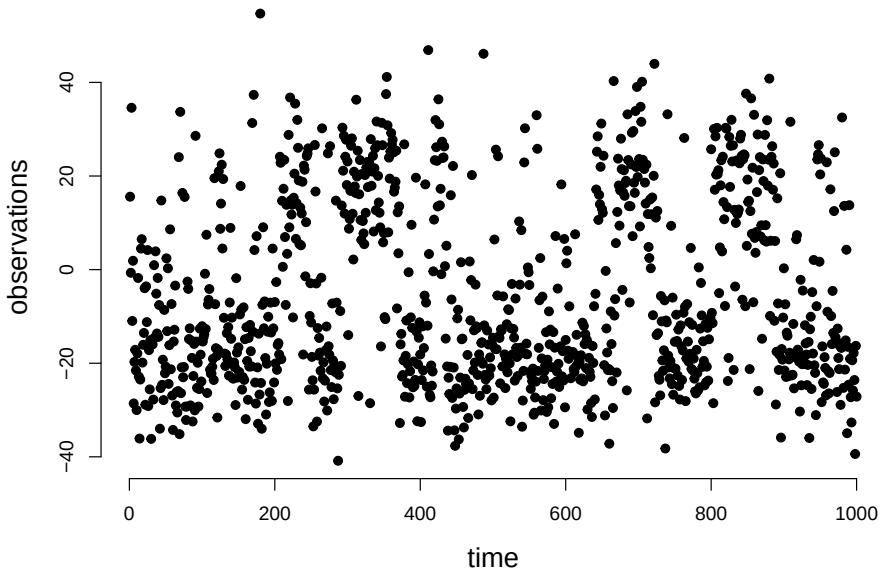
$$X_t \mid S_t = 1 \sim \mathcal{N}(103.9, 15.1)$$
$$X_t \mid S_t = 2 \sim \mathcal{N}(74.7, 12.4)$$
$$X_t \mid S_t = 3 \sim \mathcal{N}(39.6, 11.6)$$

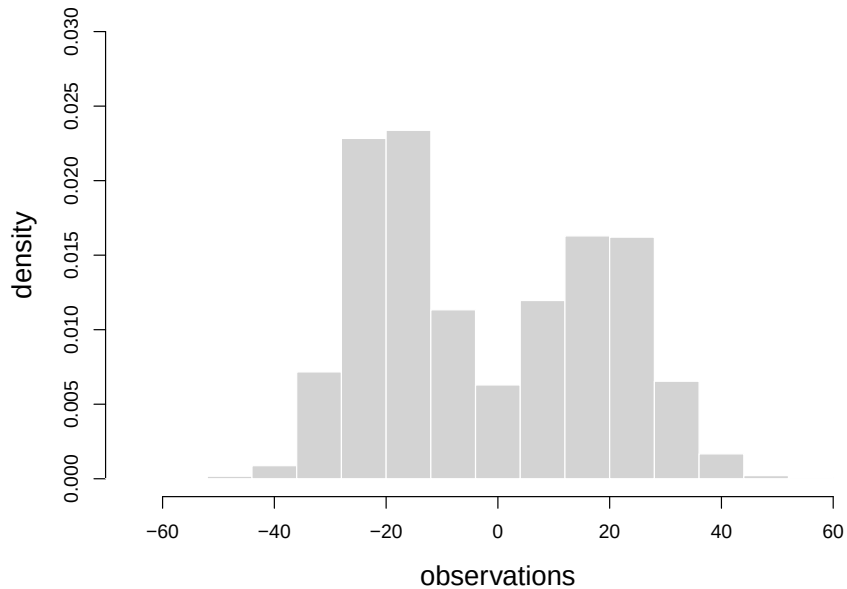
(state 4 is absorbing and indicates death)

3.5 Nonparametric inference in HMMs

Why nonparametrically estimate the state-dependent distributions?

- state-dependent distributions usually from a class of parametric distributions
- finding the “right” distributional family, or even a suitable one, can be difficult

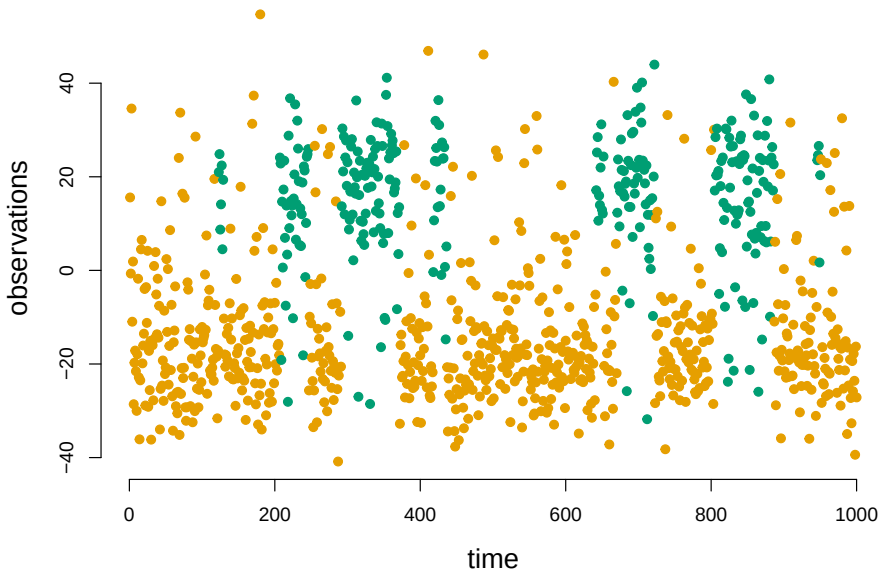


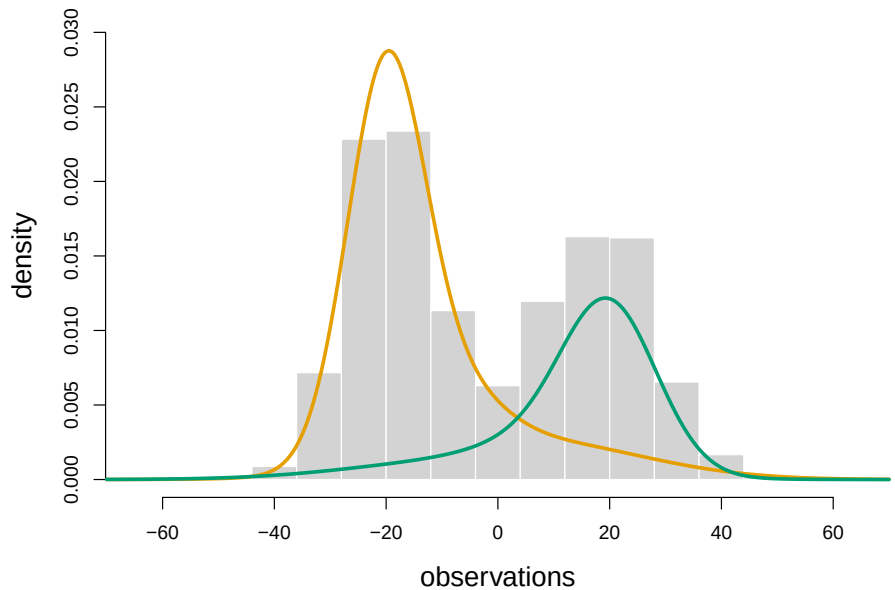


Based on this EDA, what family of distributions would you use?

Why nonparametrically estimate the state-dependent distributions?

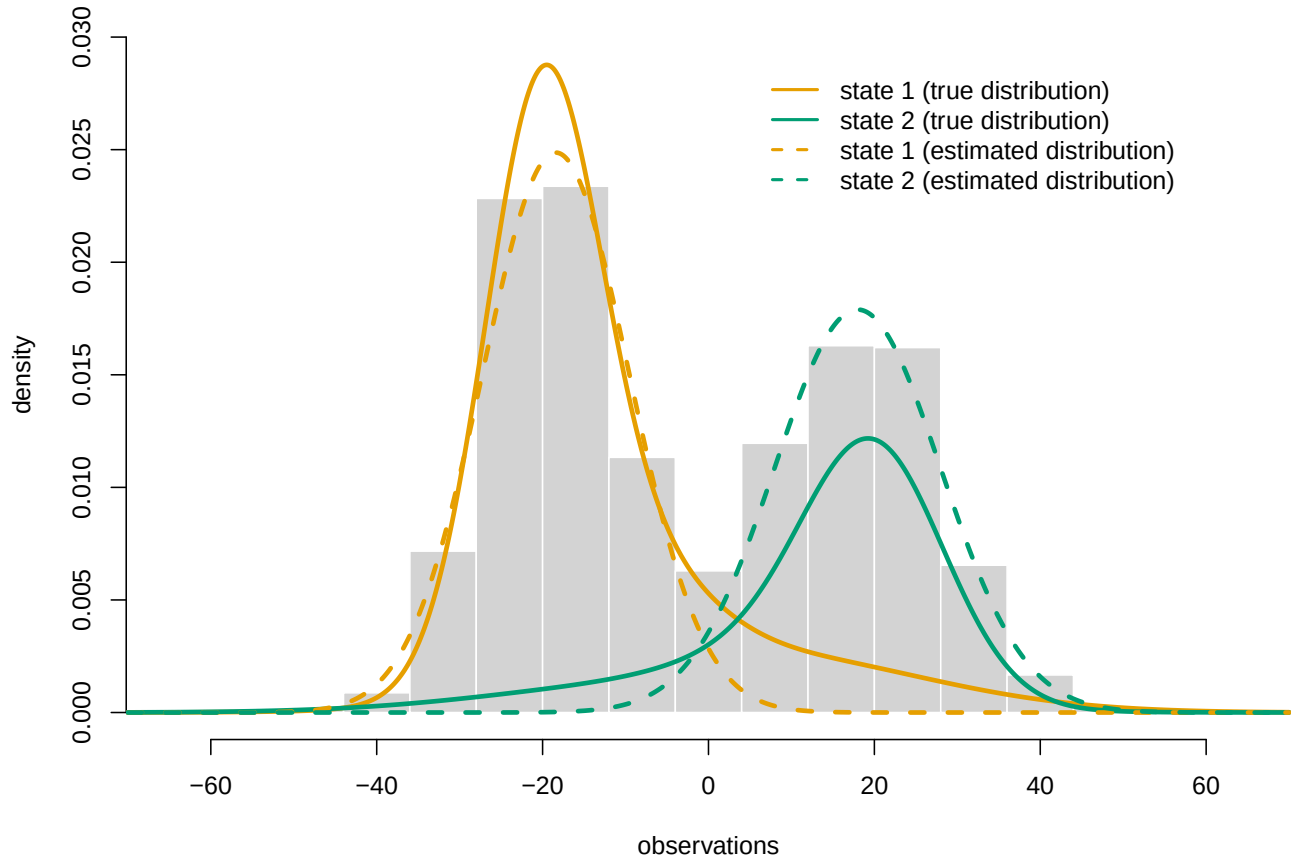
- state-dependent distributions usually from a class of parametric distributions
- finding the “right” distributional family, or even a suitable one, can be difficult





The true model: highly **skewed** distributions in both states.

Consequences of fitting a (misspecified) Gaussian HMM



Observations from the extreme tails of either of the distributions will then obviously be allocated to the incorrect state.

An unfortunate choice of the parametric family can lead to...

- ...a poor fit and hence poor predictive power
- ...a mismatch between model states and “true” states
- ...a bad performance of the state decoding
- ...invalid inference e.g. on the number of states

Alternative *nonparametric* estimation based on P–splines

Represent (state-dependent) densities using **B–spline densities**:

$$f(x_t|S_t = i) = \sum_{k=-K}^{K} \omega_{k,i}\phi_k(x_t)$$

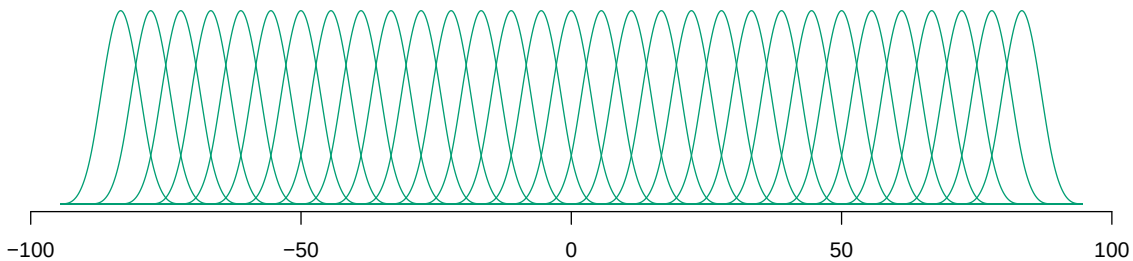


Figure: Possible set of B–spline densities $\phi_{-K}, \ldots, \phi_K$ to be used as basis functions.

Transform constrained parameters $\omega_{-K,i}, \ldots, \omega_{K,i}$ to ensure f is a density:

$$\omega_{k,i} = \frac{\exp(\beta_{k,i})}{\sum_{j=-K}^{K} \exp(\beta_{j,i})} \quad \text{with} \quad \beta_{0,i} = 0$$

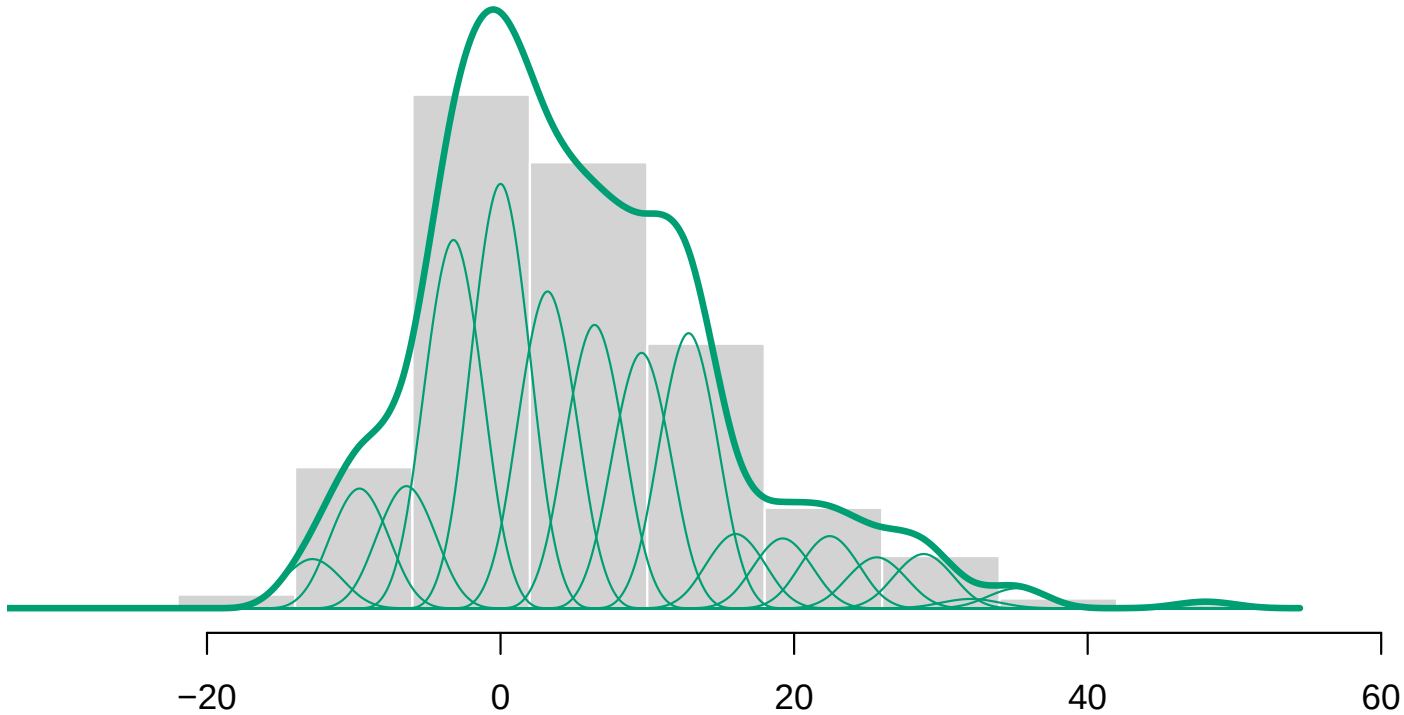


Figure: **Unpenalised estimation** of a density as linear combination of basis functions.

Smoothness selection

Tackling the bias-variance trade-off by selecting K is tedious.

Instead, we use a fairly large K (e.g. $K = 25$), to obtain virtually unlimited flexibility for capturing complex distributional shapes.

Then we numerically maximise the **penalised log-likelihood**:

$$l_p(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \log\big(\mathcal{L}(\boldsymbol{\theta})\big) - \left[\sum_{i=1}^{N} \lambda_i \sum_{k=-K+2}^{K} \big(\Delta^2 \omega_{k,i}\big)^2\right]$$

This penalty approximates the integrated squared second derivatives.

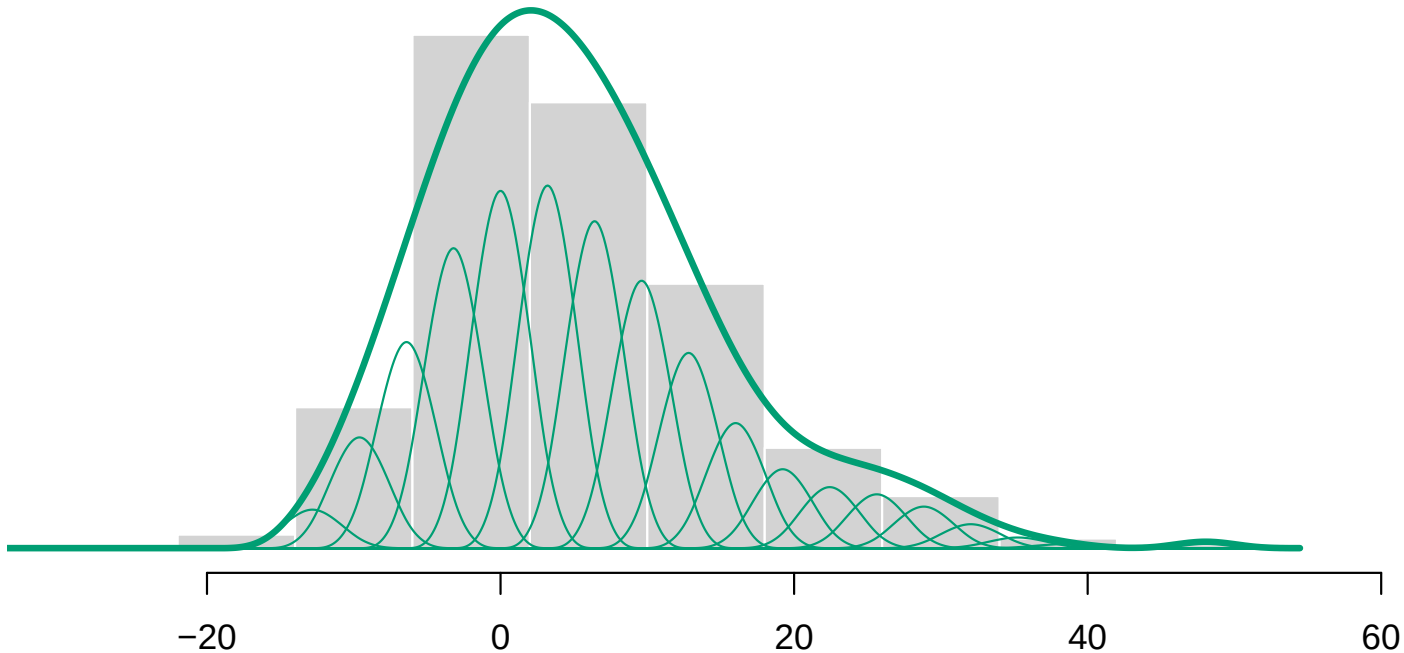


Figure: **Penalised estimation** of a density as linear combination of basis functions.

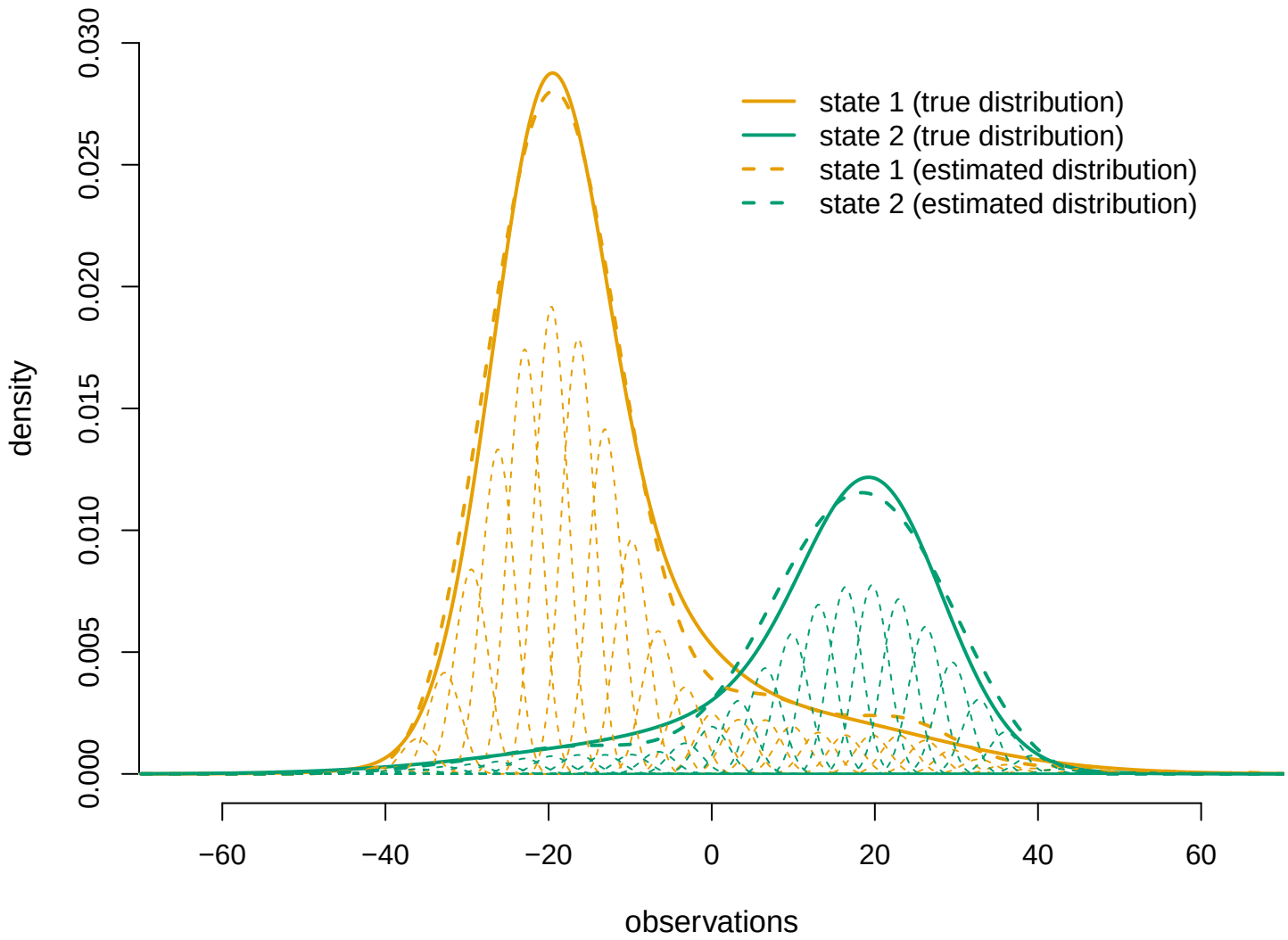
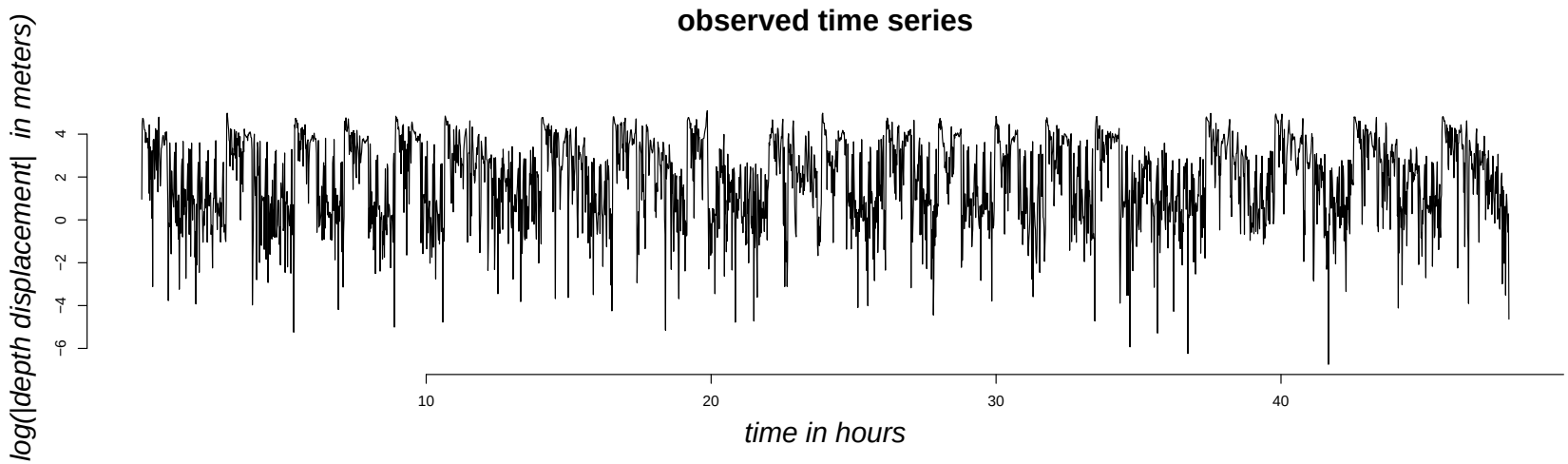
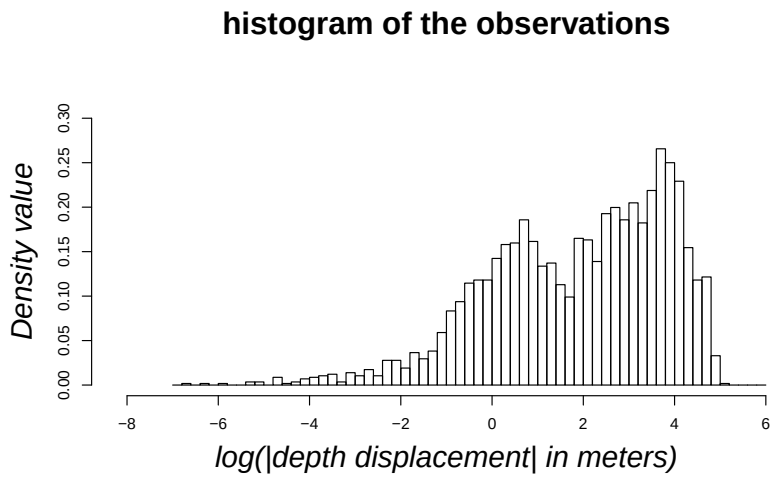
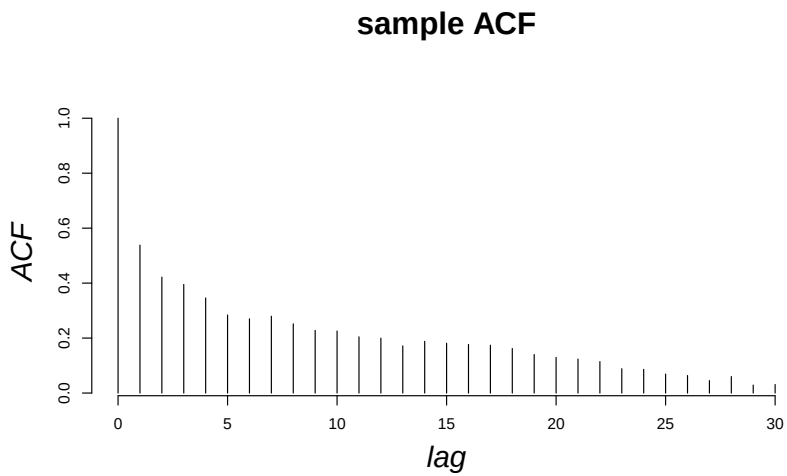


Figure: 2–state nonparametric HMM fitted to the simulated data shown earlier.

Real-data example: beaked whale dive data





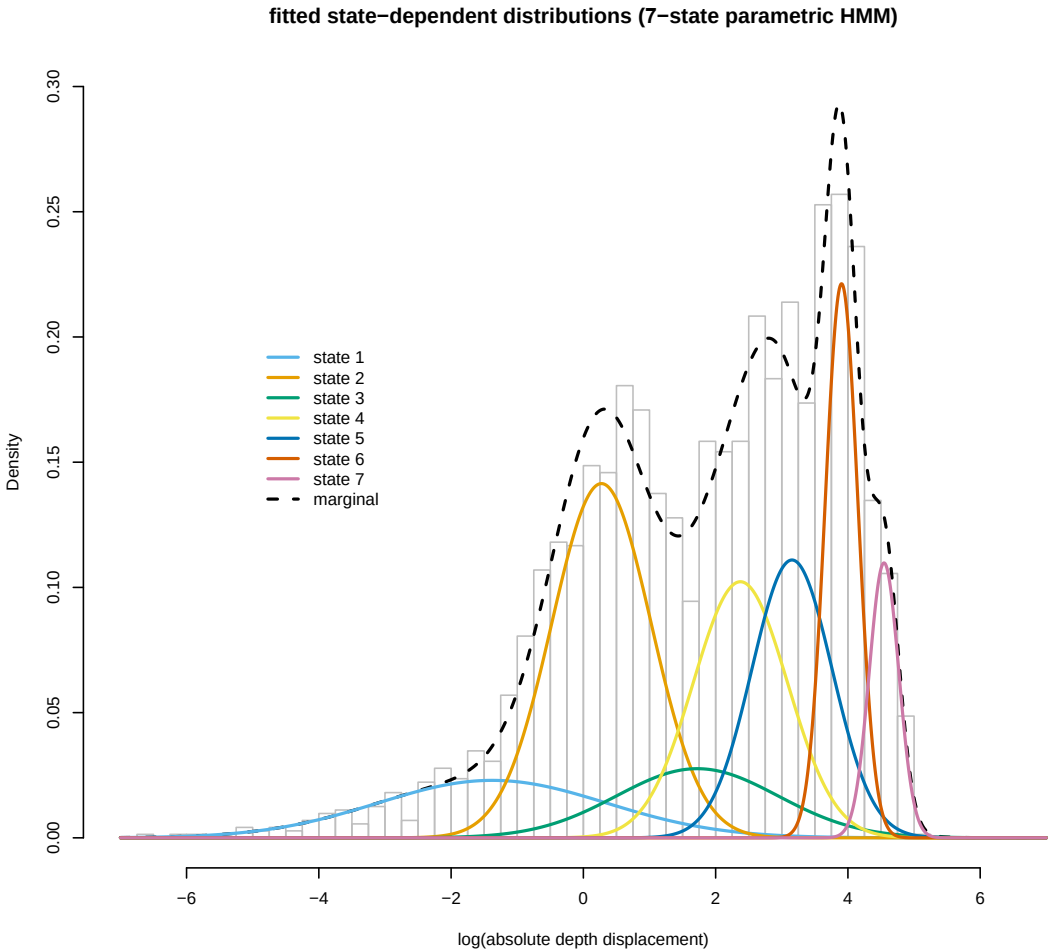


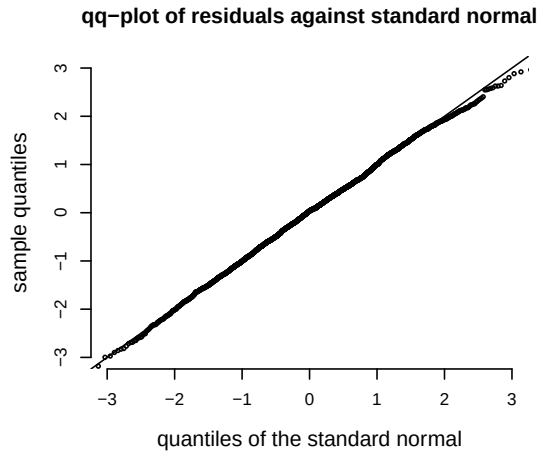
Real-data example: beaked whale dive data

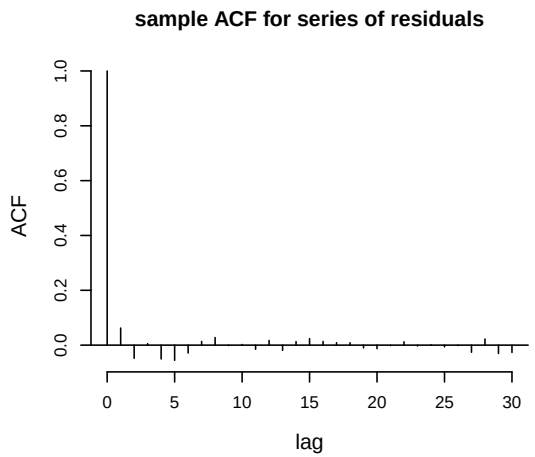
Table: Results of fitting **HMMs with normal state-dependent distributions**.

#states	#param.	AIC	BIC
3	12	9784.00	9855.59
4	20	9498.16	9617.47
5	30	9400.30	9579.27
6	42	9294.88	9545.43
7	56	9208.04	**9542.11**
8	72	9129.15	9558.67
9	90	9090.98	9627.87
10	110	**9064.53**	9720.74

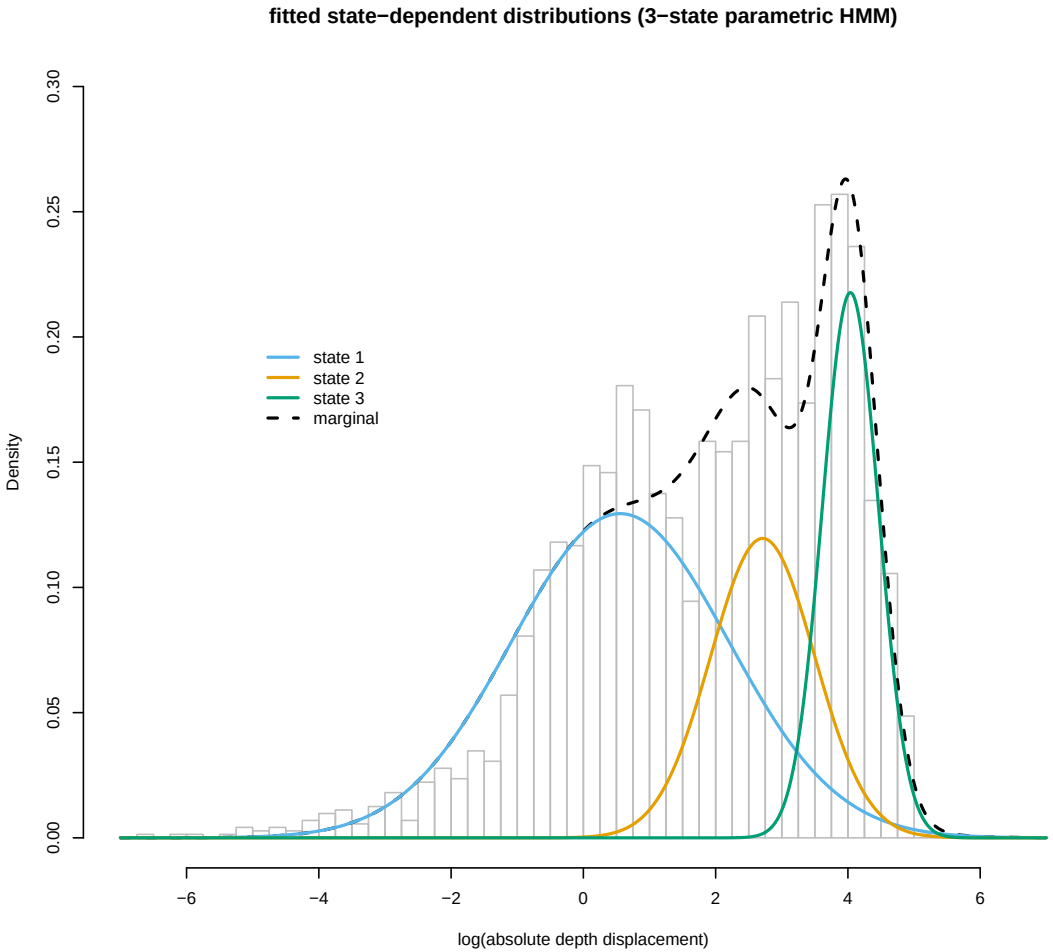
Beaked whale data — fitted parametric HMM with $N = 7$

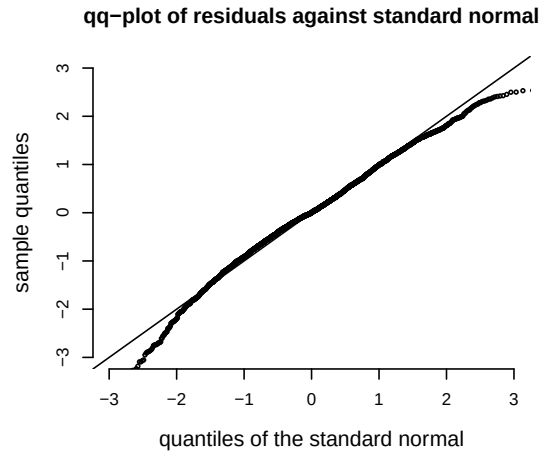


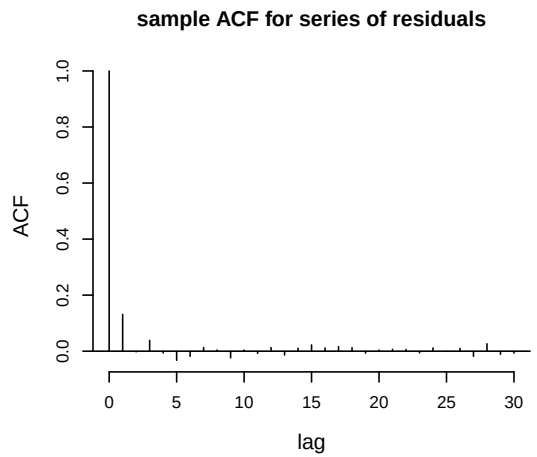




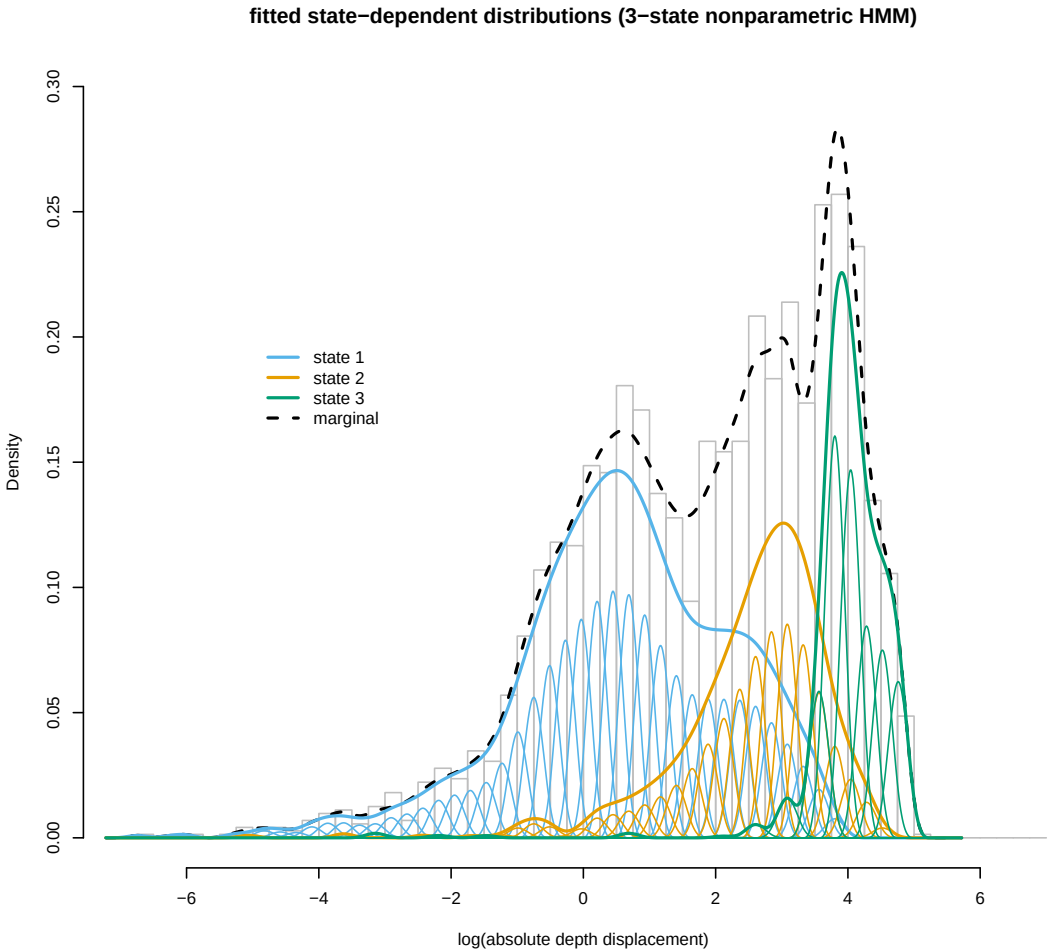
Beaked whale data — parametric HMM, $N = 3$

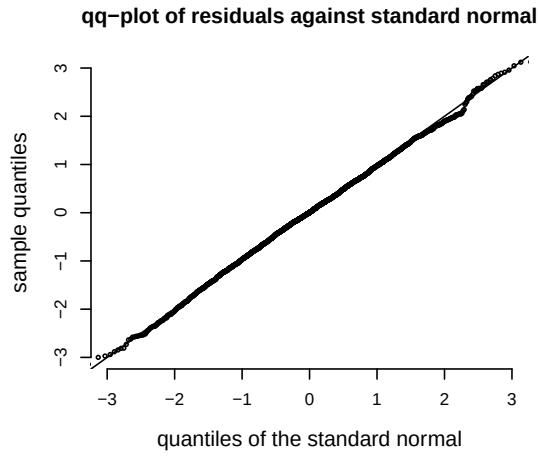


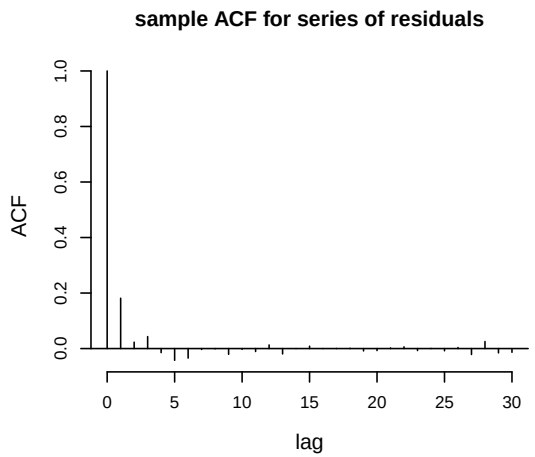




Beaked whale data — nonparametric HMM with $N = 3$







Nonparametric Markov-switching regression

Consider a general GLM–like Markov-switching regression model

$$g\Big(\underbrace{\mathbb{E}(Y_t \mid s_t, \mathbf{x}_{\cdot t})}_{\mu_t^{(s_t)}}\Big) = \eta^{(s_t)}(\mathbf{x}_{\cdot t}),$$

where...

- ...Y_t follows some distribution from the exponential family
- ...$\mathbf{x}_{\cdot t} = (x_{1t}, \ldots, x_{Pt})$ is the covariate vector at time t
- ...g is a suitable link function
- ...$\eta^{(s_t)}$ is the predictor function given state s_t
  (the form of which will be specified shortly)

Nonparametric modelling of the predictor

Now consider a GAM–type predictor:

$$\eta^{(s_t)}(\mathbf{x}_{\cdot t}) = \beta_0^{(s_t)} + f_1^{(s_t)}(x_{1t}) + f_2^{(s_t)}(x_{2t}) + \ldots + f_P^{(s_t)}(x_{Pt})$$

We represent each $f_p^{(i)}$ as a linear combination of B-spline basis functions,

$$f_p^{(i)}(x) = \sum_{k=1}^{K} \gamma_{ipk} B_k(x),$$

and numerically maximise the penalised log-likelihood:

$$l_p(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \log\big(\mathcal{L}(\boldsymbol{\theta})\big) - \sum_{i=1}^{N} \sum_{p=1}^{P} \lambda_{ip} \sum_{k=3}^{K} (\Delta^2 \gamma_{ipk})^2$$

- inference analogous as for nonparametric HMMs
- notably, parametric models are nested special cases (for $\lambda \to \infty$)

Example Spanish energy price ∼ exchange rate & market state

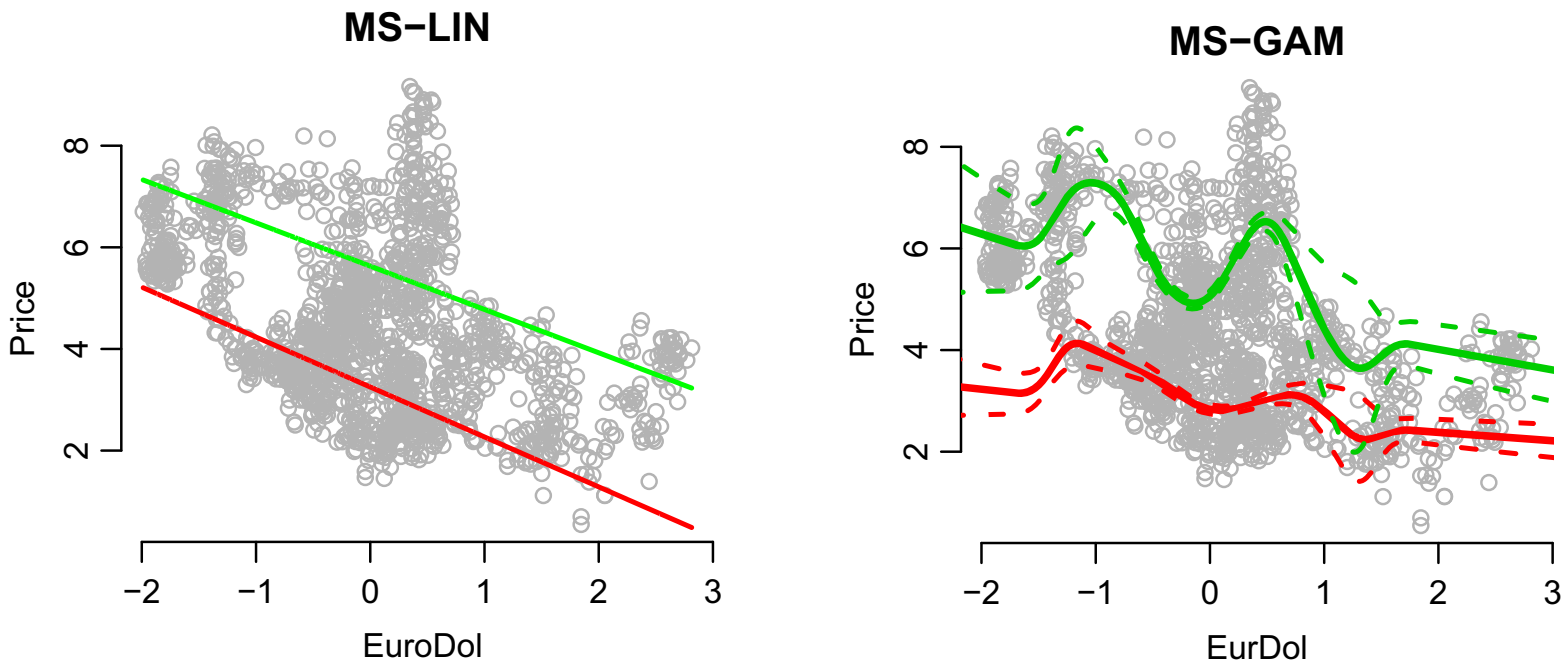


Figure: Fitted Markov-switching regression models, with linear predictor (left panel) and with nonparametric effect modelling (right panel), and colours indicating different states.

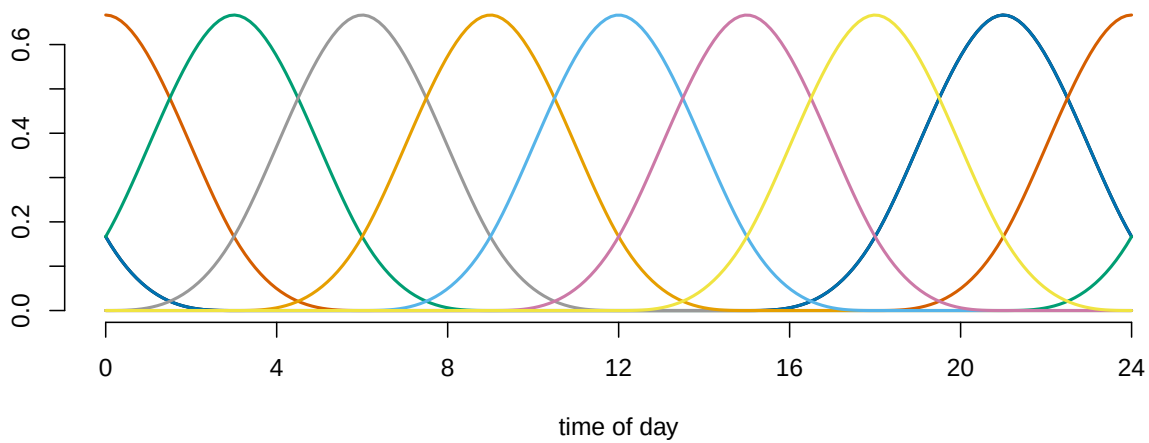
Flexible modelling of periodic variation in the state process

General parametric model for periodic variation:

$$\text{transition probabilities} \sim \mathbf{z}_t'\boldsymbol{\beta}^{(ij)} + \sum_{k=1}^{K} \omega_k^{(ij)} \sin\left(\frac{2\pi kt}{24}\right) + \sum_{k=1}^{K} \psi_k^{(ij)} \cos\left(\frac{2\pi kt}{24}\right)$$

More flexible **spline**-based model:

$$\text{transition probabilities} \sim \mathbf{z}_t'\boldsymbol{\beta}^{(ij)} + \sum_{q=1}^{Q} a_q^{(ij)} B_q(t \bmod 24)$$



Example: common fruit fly activity data

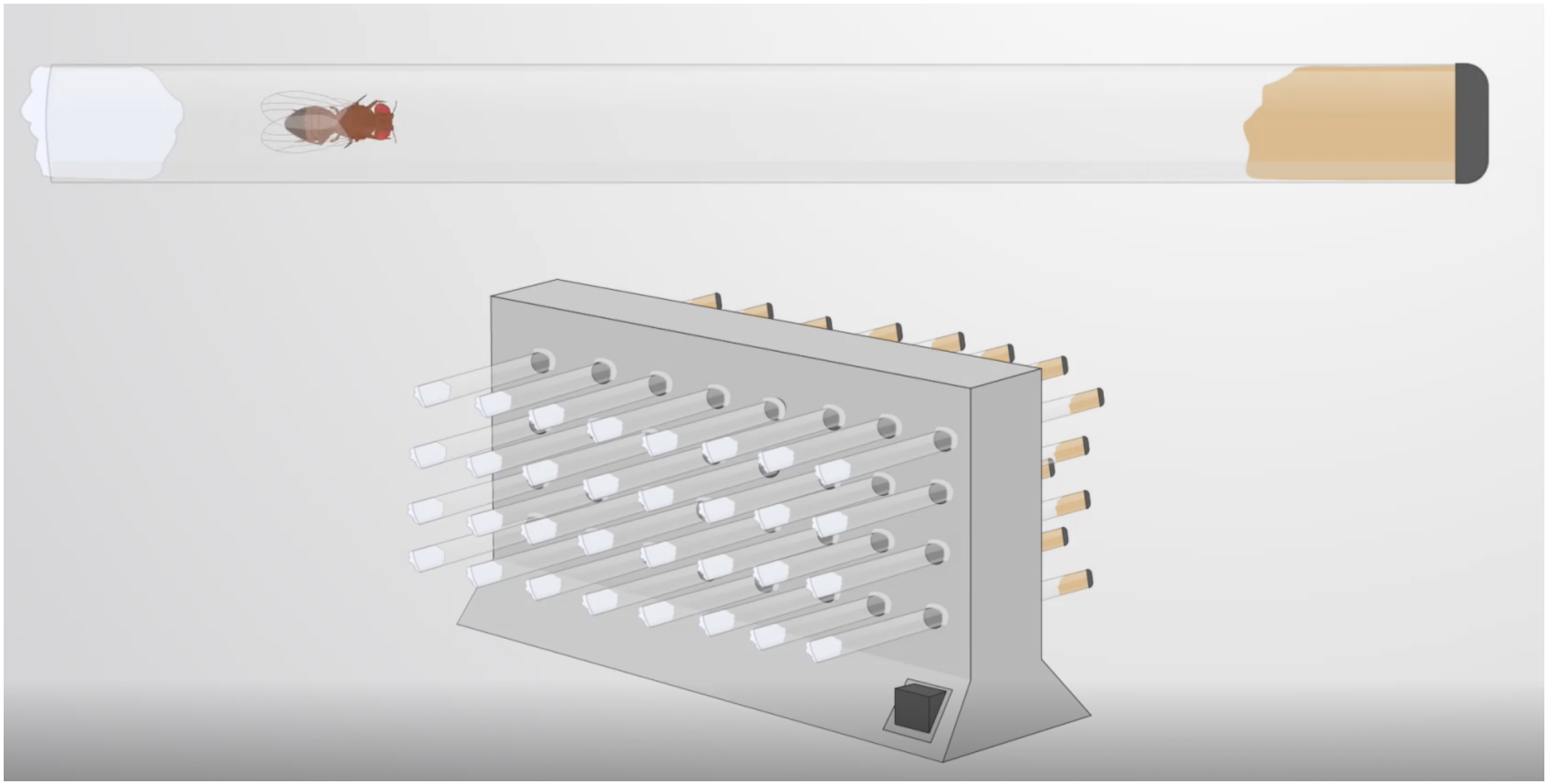


Figure: Locomotor tubes measuring fly activity as counts of passes of infrared beam.

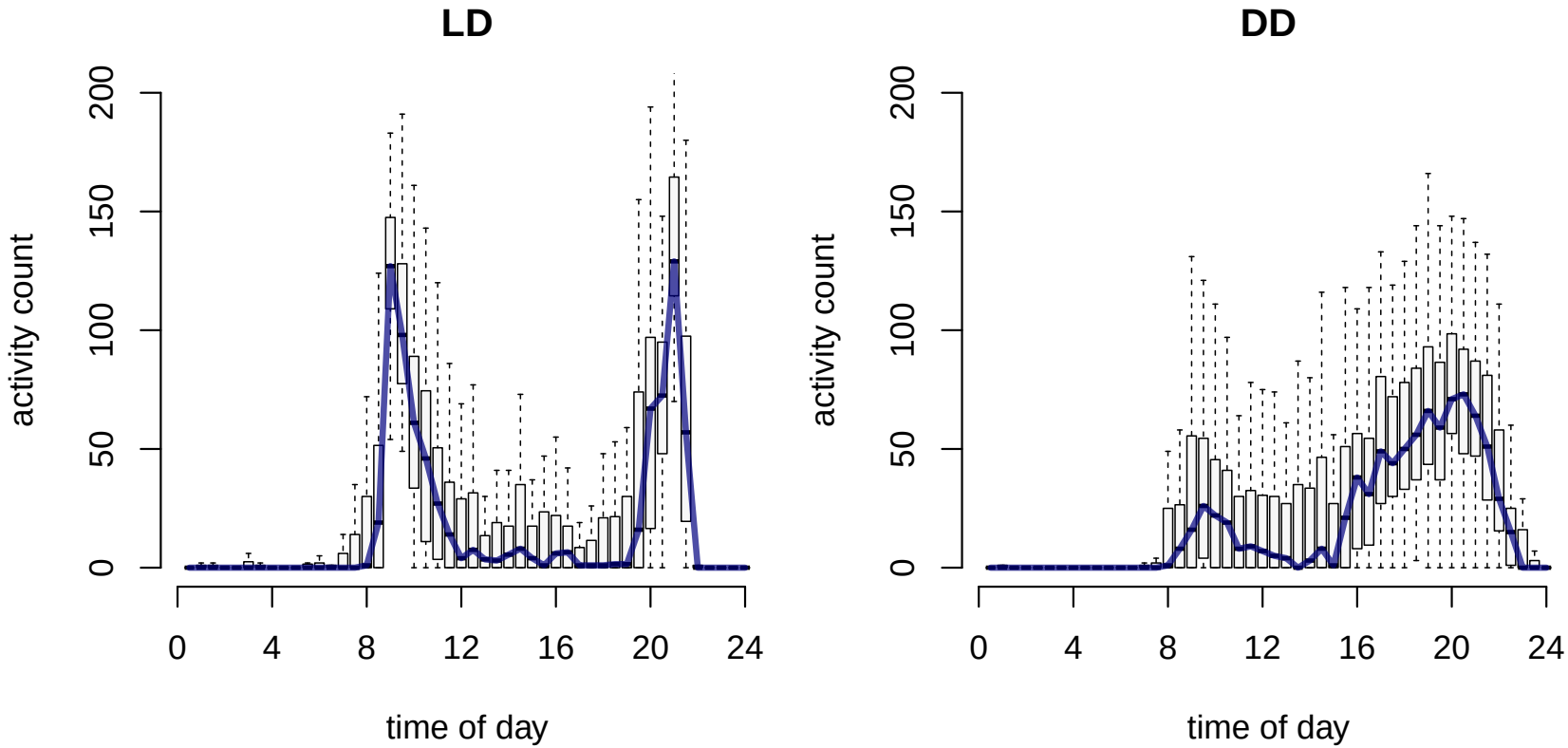


Figure: Boxplots of the flies' activity counts over the day, under two different light conditions.

2–state negative binomial HMM fitted to the fruitfly data

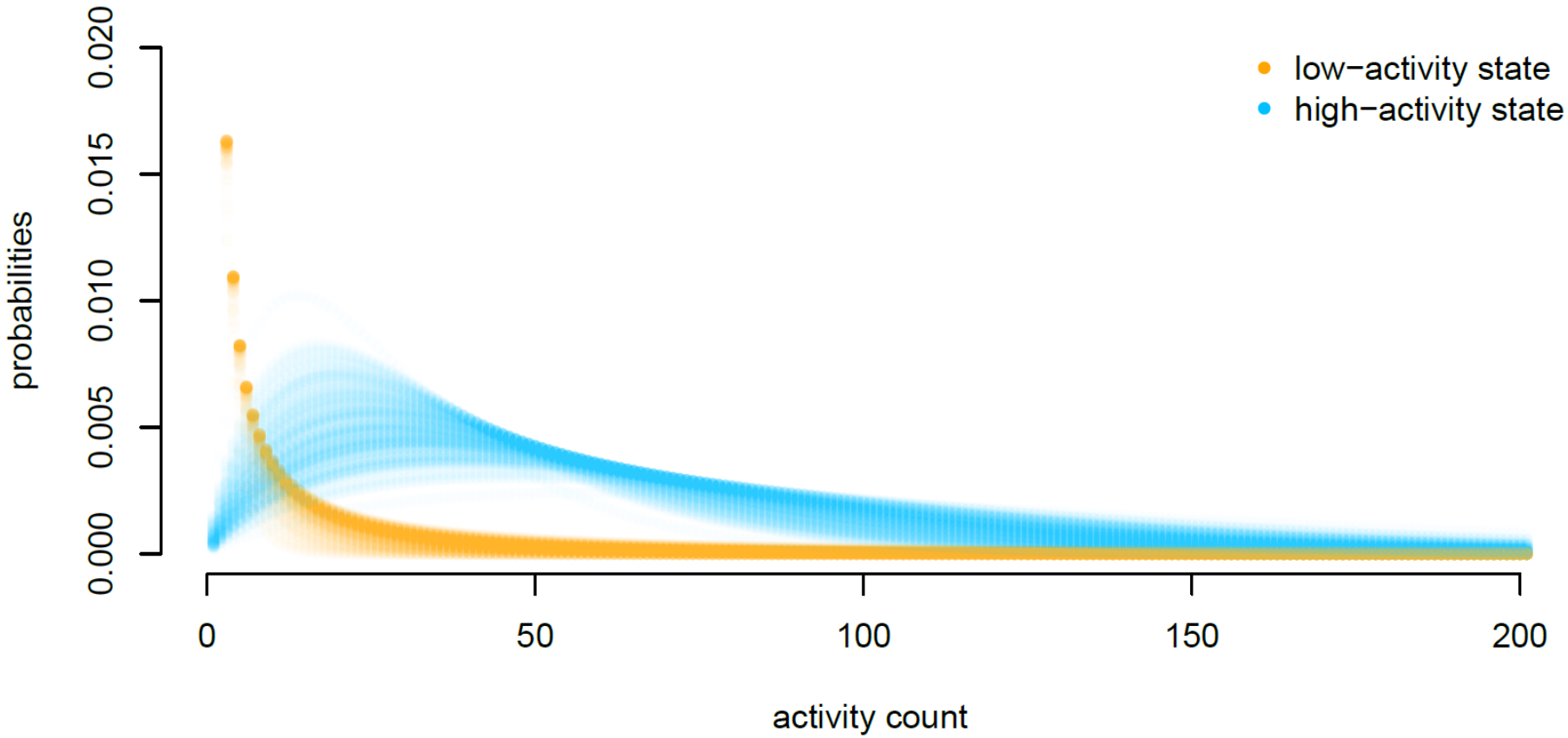


Figure: Estimated state-dependent distribution (multiple lines to indicate random effect modelling of heterogeneity across flies).

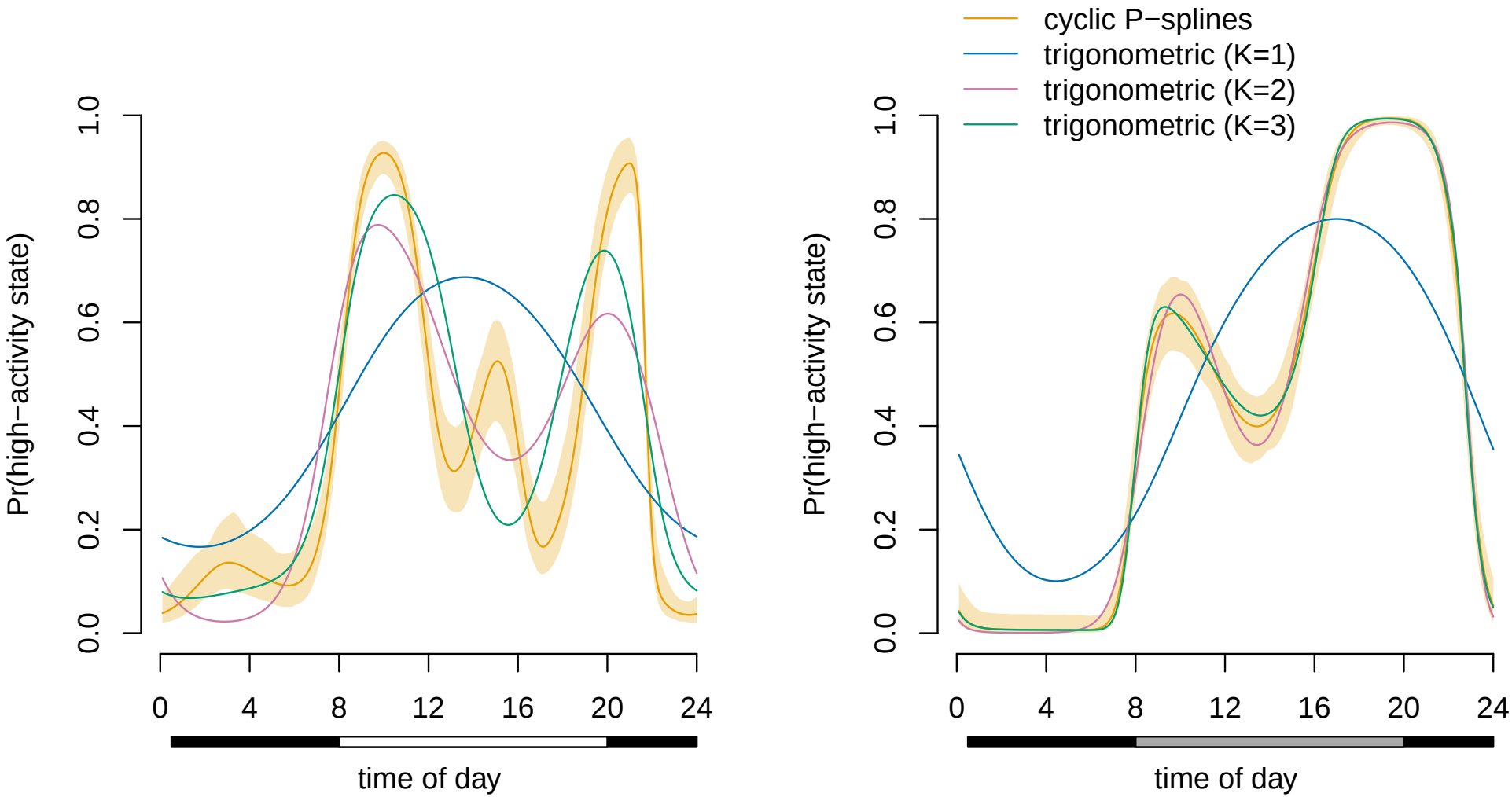
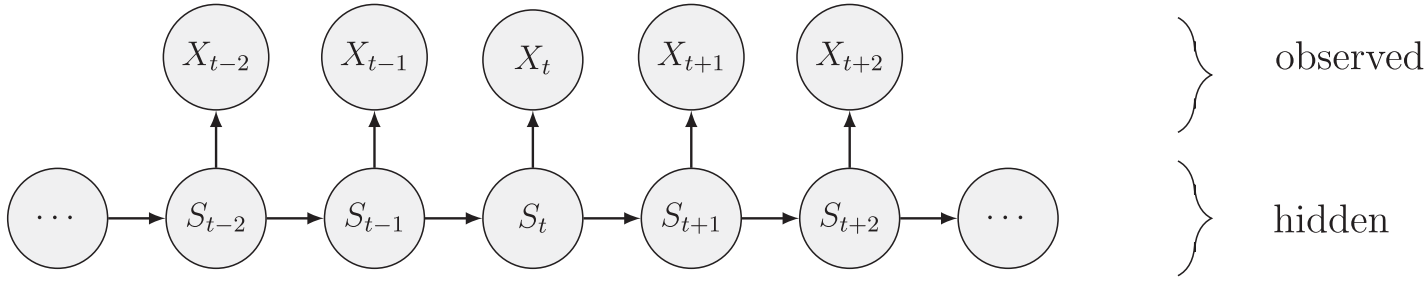


Figure: Estimated time-varying probability of being in the active state, under different specifications of the predictor for the transition probabilities (LD on the left, DD on the right).

Very quick summary at the end



HMMs constitute a flexible class of statistical models...

- for many different types of sequential (time series) data...
- ...when observations are proxies for underlying system state of interest

Various types of state-space models (also MMPPs!) are closely related, rendering the intuitive class of HMMs a good starting point in this area.

Some further reading

- Bartolucci et al. (2014), *Latent Markov Models for Longitudinal Data*, Chapman and Hall/CRC.
- Langrock et al. (2018), Spline-based nonparametric inference in general state-switching models, *Statistica Neerlandica*.
- McClintock et al. (2020), Uncovering ecological state dynamics with hidden Markov models, *Ecology Letters*.
- Maruotti (2014), Mixed hidden Markov models for longitudinal data: An overview, *International Statistical Review*.
- Mews et al. (2024), How to build your latent Markov model — the role of time and space, *arXiv*.
- Zucchini et al. (2016), *Hidden Markov Models for Time Series: An Introduction Using R*, Chapman and Hall/CRC.