

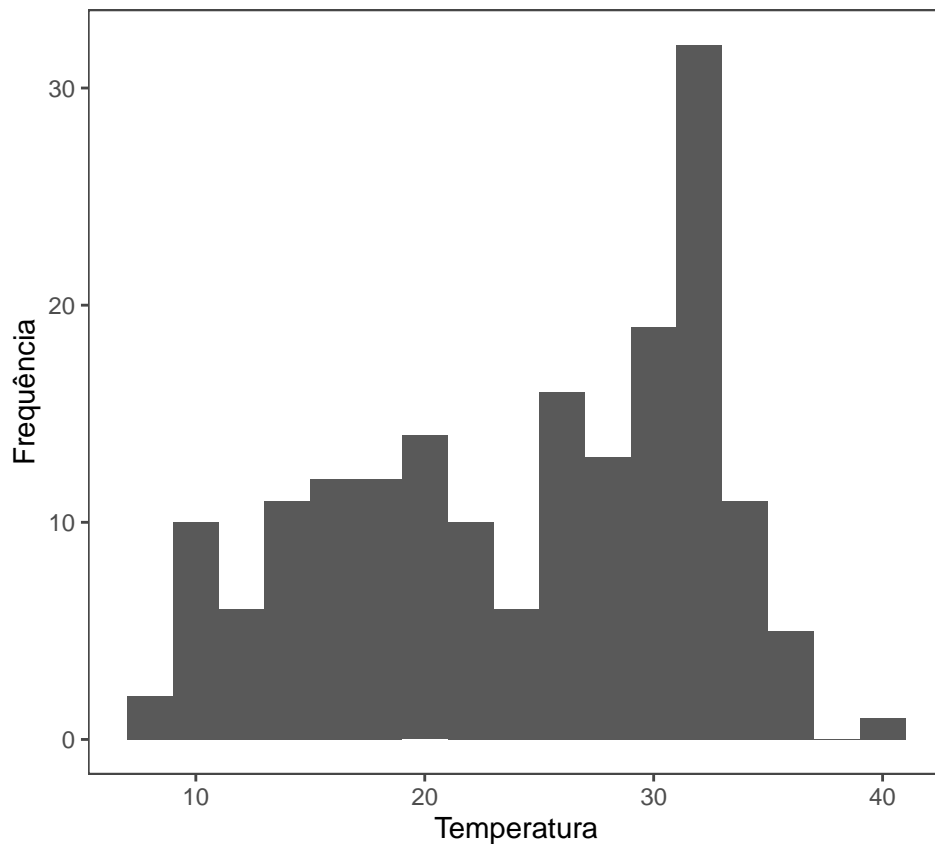
Entendendo Hidden Markov Models

Cadeia de Markov e Misturas de distribuições

Antes de definir o que são os Modelos de Markov Ocultos (HMM) vou definir os seus componentes: as Cadeias de Markov e os Modelos de Misturas de Distribuições, seguindo o livro Zucchini, Iain MacDonald, and Roland Langrock (2016) .

Misturas de distribuições

Os Modelos de Misturas são modelos estatísticos que representam distribuições multimodais. Normalmente essas distribuições multimodais são na verdade uma mistura de múltiplas distribuições. Isso pode acontecer por diversas razões mas geralmente estamos lidando com grupos diferentes ou muito heterogêneos. Por exemplo, se coletarmos a temperatura de São Paulo no inverno e de Anillaco no Verão mas nos esquecermos de identificar de onde é cada amostra a mistura dessas amostras provavelmente será bimodal, como no gráfico do exemplo abaixo.



Vou descrever um outro exemplo que foi tirado desse site. Vamos imaginar que coletamos as alturas de estudantes de uma universidade. Vamos assumir que a altura média dos homens é de 175cm e das mulheres

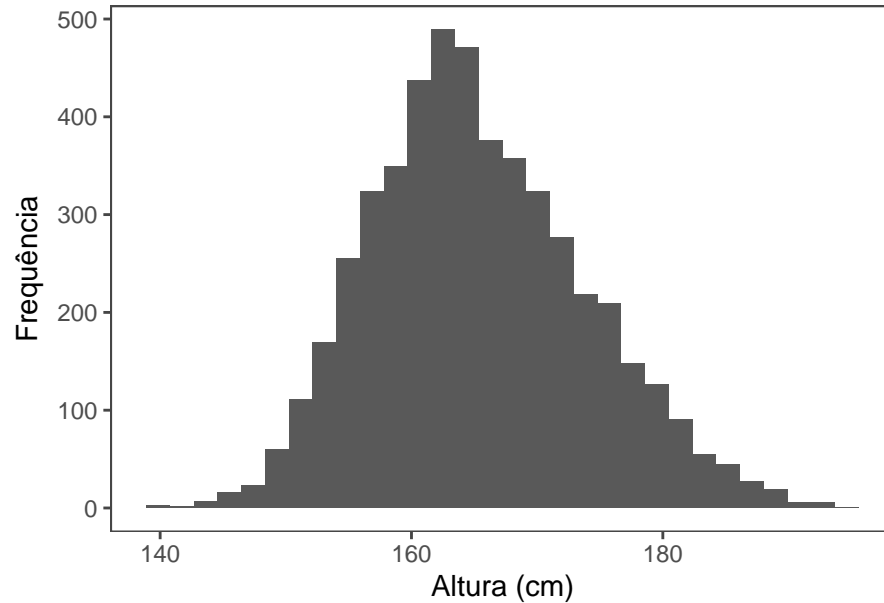
162cm, ambas distribuições com desvio padrão de 6.5cm. Porém, nessa universidade a proporção de homens na população é de 25% e de mulheres 75%.

Podemos simular a coleta desses dados. Para isso, primeiro sortearmos se a pessoa amostrada é homem ou mulher. Para cada pessoa a chance de ser homem é de 75% e mulher 25%. Depois disso, sortearmos o valor da altura dessa pessoa. Se for homem sortearmos um número dentro da distribuição normal $N(175cm, 6.5cm)$ caso contrário a amostra será sorteada da distribuição normal $N(162cm, 6.5cm)$.

Uma amostra de 5000 pessoas nos dá o gráfico A abaixo. A mistura dessas distribuições ainda é unimodal porque existe uma sobreposição grande entre a altura de homens e mulheres. No entanto, ainda é possível notar que a distribuição da amostra não é simétrica e portanto não é normal. No gráfico B podemos ver a densidade probabilística dessa distribuição sobreposta às densidades das distribuições das alturas de homens e mulheres, o que deixa um pouco mais claro como acontece a mistura de distribuições.

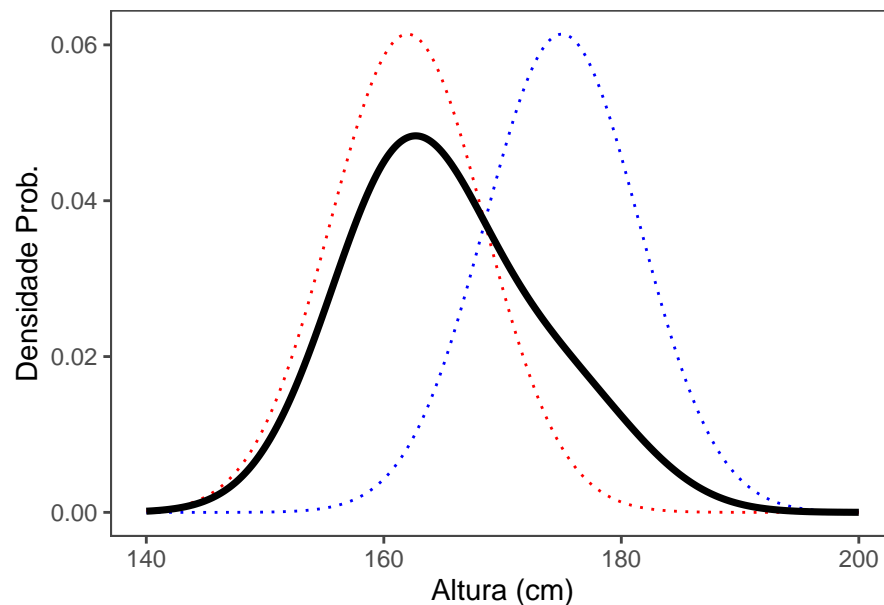
A

Amostra de 5000 pessoas



B

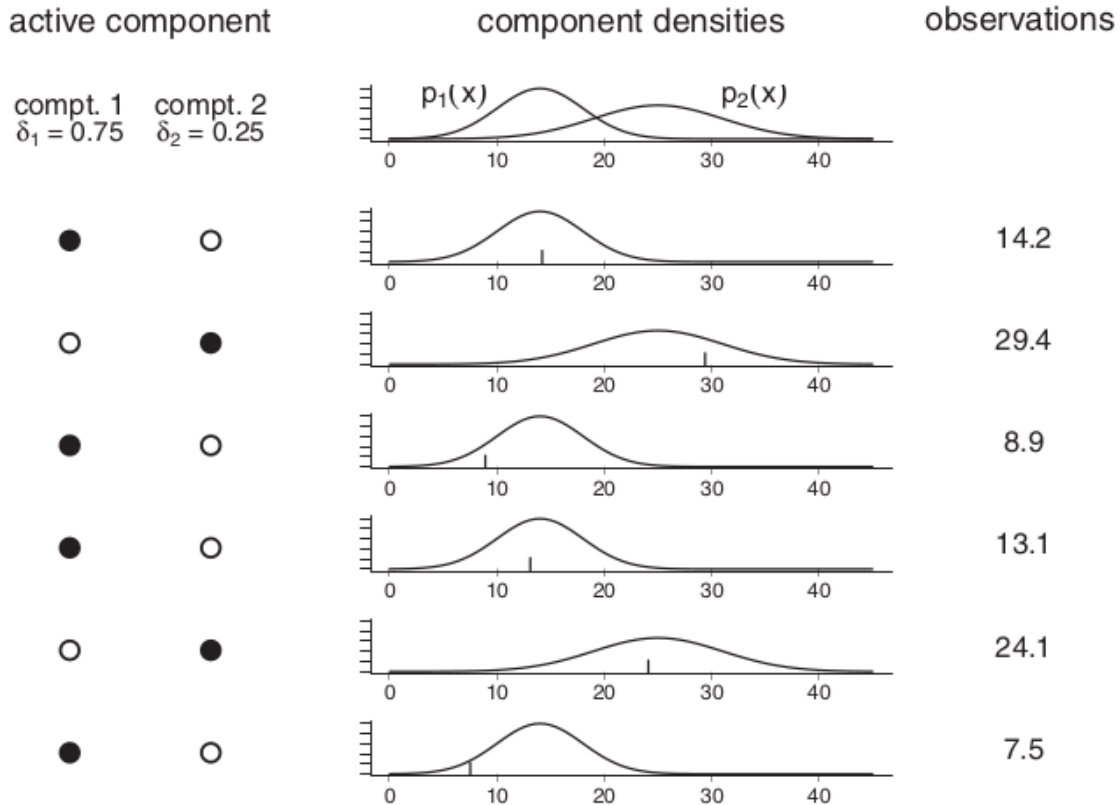
Gráfico de Densidade



Esse último exemplo dá uma noção do que são os modelos de mistura de distribuições. Basicamente esse modelo são compostos por um conjunto de distribuições e uma chance de sortearmos uma dessas distribuições. No exemplo dos estudantes temos uma mistura composta de duas distribuições normais, com chances de 75% e 25% de cada uma delas ser amostrada.

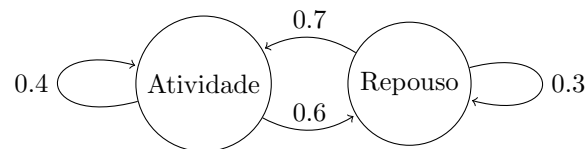
A ilustração abaixo, retirado de Zucchini, Iain MacDonald, and Roland Langrock (2016), mostra como é o processo de geração de dados nesse modelo. No lado esquerdo temos quais distribuições, ou estados, estão ativos naquele momento. No meio estão as curvas de densidade das distribuições correspondentes a cada estado. Por fim, à direita estão os valores observados em cada momento, que são provenientes da distribuição ativa naquele momento. Podemos pensar nesse processo como uma lançamento de uma moeda combinado

à um sorteio de um número de um saco. Nesse caso temos dois sacos, com números diferentes dentro de cada um. O saco do qual vamos sortear um número vai depender do resultado do lançamento da moeda. Se tirarmos Cara sortearmos um número do primeiro saco mas se tirarmos Coroa sortearmos um número do segundo saco.



Cadeias de Markov

O segundo componente de um HMM é uma Cadeia de Markov. Uma cadeia de markov é um modelo estocástico (ou aleatório) que descreve como uma sequência de estados ao longo do tempo. Nesse modelo as mudanças de um estado para outro possuem probabilidades definidas.



Como cada ponto da sequência está ligado no tempo eles formam uma ‘corrente’. Uma característica importante desse modelo é que cada transição de estado é dependente *apenas* do estado que o sistema estava antes da transição. Isso significa que a o caminho do sistema até um estado não influencia no seu próximo estado. Essa propriedade é conhecida como Propriedade de Markov e é um relaxamento do princípio da independência.

Simulando Cadeias de Markov no R

Para simularmos uma cadeia de Markov precisamos definir nossa matriz de transição. Essa matriz mostra as probabilidades de transições entre os estados. As linhas da matriz representam o estado atual e as colunas o próximo estado na cadeia. Vamos simular as transições entre repouso e atividade de um animal, assumindo que esse animal é arritmico. Esse animal quando em repouso tende a permanecer em repouso 90% das vezes. Agora, se esse animal está em atividade a chance dele entrar em repouso ou permanecer em atividade é a mesma: 50%. Com esses dados, nossa matriz de transição fica assim:

```
states = c("repouso", "atividade")
(transition_matrix = matrix(c(0.9, 0.1, 0.5, 0.5), byrow = T, nrow = 2, dimnames = list(states,states)))
```

```
##           repouso atividade
## repouso      0.9      0.1
## atividade    0.5      0.5
```

O próximo passo é simular as mudanças de estado a partir de um estado inicial. Para isso vamos definir também uma matriz 1x2 com as probabilidades do sistema começar em cada um dos estado. Nesse exemplo a probabilidade do animal iniciar em atividade ou repouso é a mesma, 50% para cada.

```
(initial_prob = matrix(c(0.5, 0.5), nrow = 1, dimnames = list(c(),states)))
```

```
##           repouso atividade
## [1,]      0.5      0.5
```

Com essas probabilidades iniciais e a matriz de transição podemos simular algumas cadeias de Markov. Vamos simular 9 cadeias de Markov com 500 transições cada. Cada transição é correspondente a uma unidade de tempo em que estamos coletando os dados, o que poderia corresponder a minutos ou horas dependendo do tipo de coleta.

```
set.seed(53)

# Quantas simulações e probabilidades iniciais
nchains = 9
niter = 500

# Function that simulates a Markov Chain given a
# vector containing the starting state in position 1
markov_chain_sim = function(transition_matrix, initial_prob, niter = 50){
  nstates = nrow(transition_matrix)
  chain = numeric(niter)

  #Sorteia os estados iniciais de acordo com a matriz de prob.
  chain[1] = sample(1:nstates, 1, prob = t(initial_prob))

  for(t in 2:niter){
    chain[t] = sample(1:nstates, 1, prob = transition_matrix[chain[t-1],])
  }

  return(chain)
}
```

```

# Criando uma matriz que conterá todas simulações
# Cada coluna é uma simulação diferente.
# O número de linhas corresponde à quantidade de transições que queremos simular
sim = matrix(NA, niter, nchains)
colnames(sim) = paste0("sim", 1:nchains)

# Repetimos a função markov_chain_sim para cada coluna
for(i in 1:nchains){
  sim[,i] = markov_chain_sim(transition_matrix, initial_prob, niter)
}

tail(sim)

```

```

##      sim1 sim2 sim3 sim4 sim5 sim6 sim7 sim8 sim9
## [495,]  1   2   2   1   1   1   1   1   1
## [496,]  1   1   1   1   1   1   1   1   1
## [497,]  1   2   1   1   1   1   1   1   1
## [498,]  1   2   1   1   1   1   1   1   1
## [499,]  1   1   2   1   1   2   1   1   1
## [500,]  1   2   2   1   1   1   1   1   1

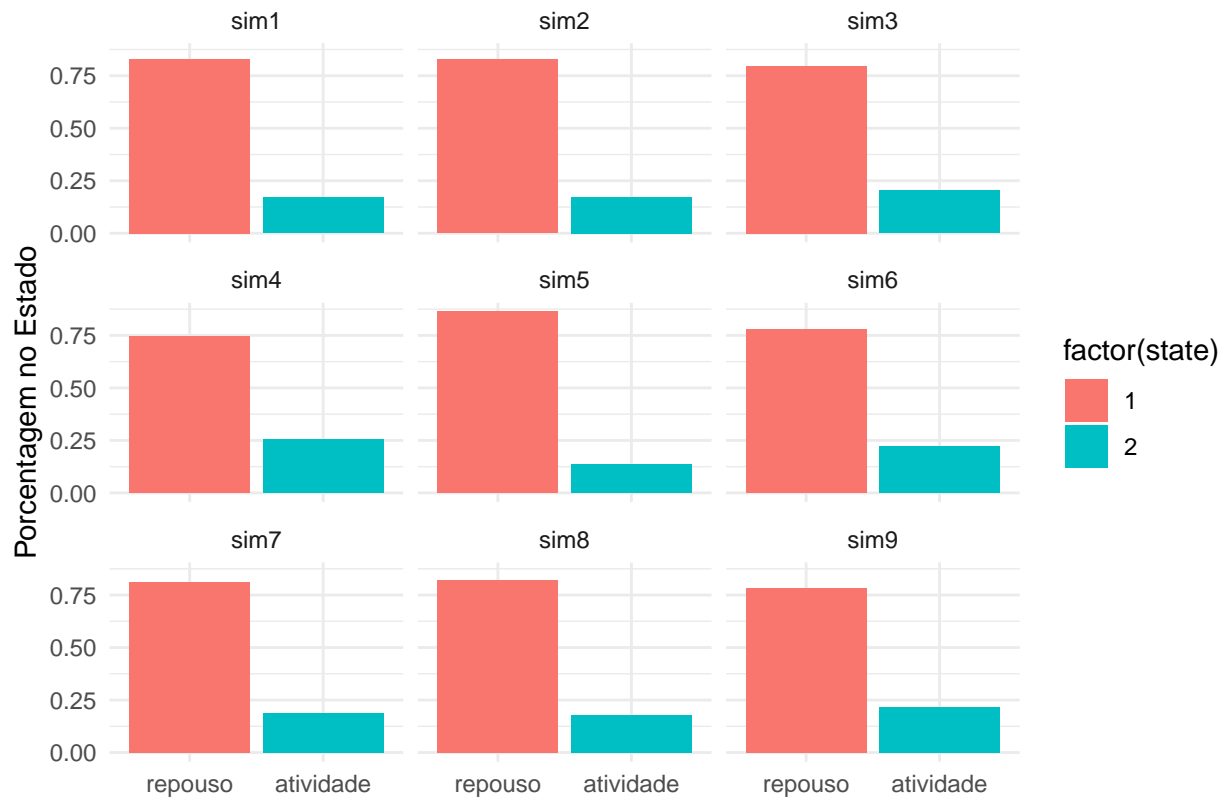
```

Visualizando os resultados

Vamos plotar esses os resultados de algumas simulações para entendermos como o sistema está mudando no tempo. Podemos ver no gráfico abaixo que pela Cadeia de Markov ser um processo estocástico nenhuma das simulações é igual a outra.



Ou seja, não é possível prever com certeza a sequência de estados. Porém, o que é interessante é que se calcularmos a proporção de repouso e atividade sobre o total de observações, que no nosso caso são 500, vamos obter um resultado muito próximo entre as simulações.



O fato das simulações se estabilizarem em certas proporções não é uma coincidência. Essa era justamente a propriedade que Markov queria demonstrar. Dado algumas condições para a cadeia e um número de observações suficientes as proporções vão se estabilizar ao redor de algum valor mesmo que as observações não sejam totalmente independentes entre si.

Usando Matrizes

Ao invés de simularmos uma sequência de transições e só depois calcularmos as proporções dos estados podemos fazer isso com matrizes. Para isso multiplicamos a matriz de transição pela matriz de probabilidades iniciais. O resultado é então multiplicado pela matriz de transição. Esse processo é repetido até o número de iterações, ou tempo, que queremos que a cadeia tenha.

```
niter = 500

step = initial_prob %*% transition_matrix
for (i in 1:niter){
  step = step %*% transition_matrix
}

step
```

```
##      repouso atividade
## [1,] 0.8333333 0.1666667
```

Aplicações

Usando Modelos de Markov podemos investigar perguntas como:

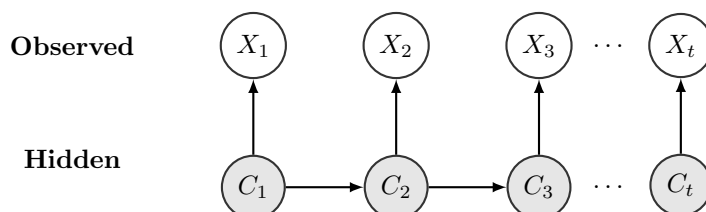
- Como o sistema muda no tempo: Qual o estado o sistema vai estar em x tempo?
- Traçar as possibilidades de caminhos entre o estado atual do sistema e um estado inicial
- Caso tenhamos os dados coletados: Quais as probabilidade de mudança de estados (REF).

Hidden Markov Models

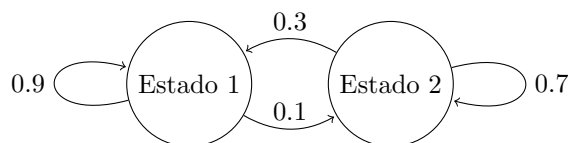
Assim como as Cadeias de Markov os *Hidden Markov Models* também são modelos estocásticos. Os HMM descrevem a mudança de eventos que dependem de algum fator interno, que não pode ser observado, ao longo do tempo. Esse modelo é dividido em dois componentes: (i) uma Mistura de Distribuições e (ii) uma Cadeia de Markov. A cadeia de Markov é a parte oculta do modelo, que representa o seu estado interno. Cada um dos estados dessa cadeia está conectado a uma distribuição com diferentes parâmetros. Assim, as observações são exatamente as realizações dessas diferentes distribuições, o que forma uma mistura de distribuições, como no exemplo das alturas.

De uma forma mais formal o HMM consiste de dois processos distintos. Um ‘parameter process’ que satisfaz as propriedades de Markov e um ‘state-dependent process’ que depende apenas do estado que está ativo num dado momento. Nossas observações fazem parte do *state-dependent process* e podem se denominadas por X_1, \dots, X_t , o que representa uma sequência da observação da variável X no tempo 1 até a observação de X no tempo t . A cadeia de Markov é o *parameter processes*, denominada C_t . Essa cadeia pode ter m estados, e assim vamos chamar X_t de um HMM de m estados.

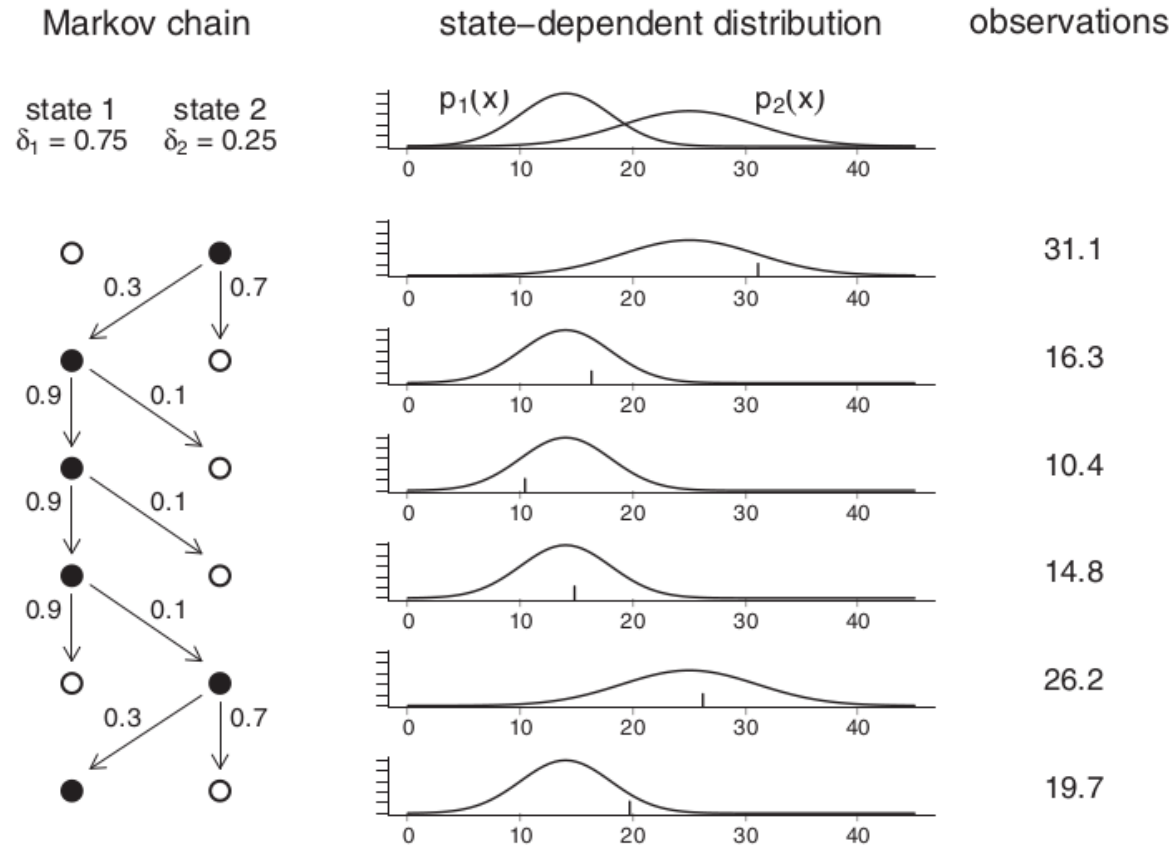
A figura abaixo mostra uma sequência de estados C_t e o processo dependente do estados X_t que é emitido pelo estado que está ativo em cada momento.



Podemos voltar ao exemplo de Zucchini, Iain MacDonald, and Roland Langrock (2016) mas agora esquematizá-lo como um HMM. Vamos tomar as probabilidades iniciais da Cadeia de Markov como $(0.75, 0.25)$ e a matriz de transição: $\begin{bmatrix} 0.9 & 0.1 \\ 0.3 & 0.7 \end{bmatrix}$.



Com essas probabilidades de transição podemos esquematizar como é o processo de geração dos dados sob um HMM com dois estados. Nesse exemplo, a escolha da distribuição ativa é coordenada por uma Cadeia de Markov e, diferentemente do exemplo de Misturas de Distribuições, as probabilidades de mudança de estado dependem do estado anterior. Esse é o princípio dos HMMs, de uma maneira simples (que não sei se é correta) estamos tentando identificar a distribuições que compõe essa mistura, onde cada distribuição é representada por um estado levando em conta a dependência temporal que existe entre as observações. As perguntas que podem ser respondidas por esse modelos são diversas e dependem do tipo de dados que temos.



No nosso caso temos as observações mas não sabemos nada sobre os estados. Assim, queremos usar as observações para tentar agrupar os dados em um número de estados escolhidos à priori. Assim, estamos usando o HMM de uma maneira não supervisionada. Um caso diferente, ou supervisionado, seria se tivéssemos tanto os dados de atividade quanto os estados correspondentes a esses dados em alguns momentos. Isso aconteceria se, por exemplo, conseguíssemos filmar o comportamento dos Tucos em alguns momentos do dia. Dessa maneira podemos classificar os dados que não foram filmados e compará-los com os dados que foram filmados.

Referências

- https://willhipson.netlify.app/post/markov-sim/markov_chain/
- <https://a-little-book-of-r-for-bioinformatics.readthedocs.io/en/latest/src/chapter10.html>
- <http://recologia.com.br/2013/03/13/um-exemplo-de-markov-chain/>
- <https://towardsdatascience.com/markov-models-and-markov-chains-explained-in-real-life-probabilistic-workout-routine-65e47b5c9a73>
- <https://www.stat.auckland.ac.nz/~fewster/325/notes/ch8.pdf>

Zucchini, Walter, Iain MacDonald, and Roland Langrock. 2016. *Hidden Markov Models for Time Series - an Introduction Using R*. Vol. 43.