Fitting Normal HMMs

Simone

24/05/2021

Generate Data from a 3-State Normal HMM

$$\boldsymbol{\mu} = (12,\ 18,\ 22), \quad \boldsymbol{\sigma} = (3,\ 1.5,\ 2), \quad \boldsymbol{\Gamma} = \begin{pmatrix} 0.9 & 0.03 & 0.07n \\ 0.050 & 0.899 & 0.051 \\ 0.05 & 0.15 & 0.80 \end{pmatrix}, \quad \boldsymbol{\delta} = (0.3333333,\ 0.4382470,\ 0.2284197)$$

```
T=10000
norm3s <- list(m = 3,
               mu = c(12, 18, 22),
               sigma = c(3, 1.5, 2),
               gamma = matrix(c(0.9, 0.03, 0.07,
                                0.050, 0.899, 0.051,
                                0.05, 0.15, 0.80), nrow=3, ncol=3, byrow = TRUE),
               delta = c(0.3333333, 0.4382470, 0.2284197))
sample <- norm.HMM.generate_sample(T, norm3s)
normdata = data.frame(Time = c(1:T),
                      Observation = sample$observ,
                      State = factor(sample$state))
x <- normdata$Observation
```

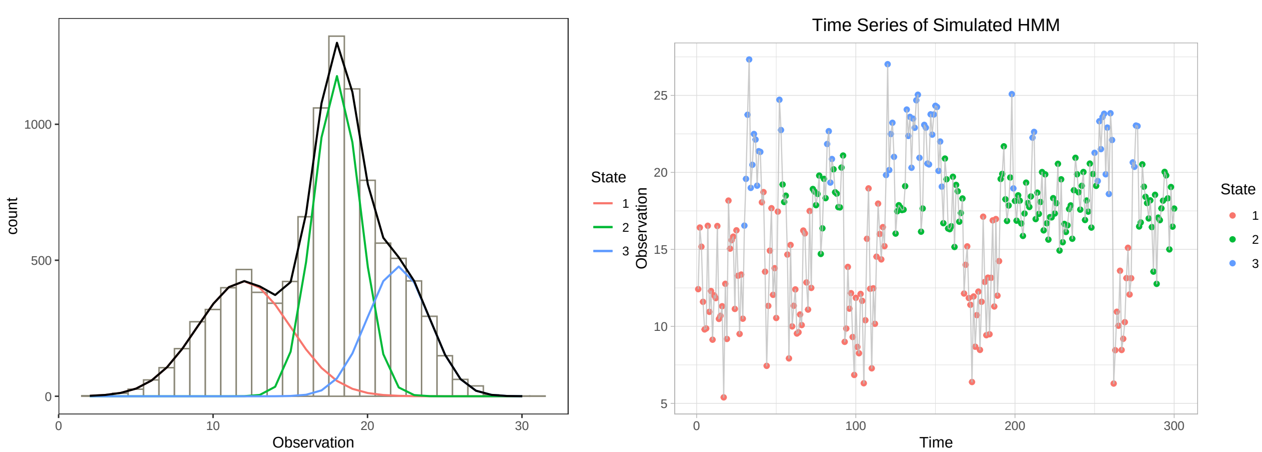


Figure 1: LEFT: Histogram of observations with state dependent normal distributions overlayed. Black line is the marginal distribution. RIGHT: Timeseries of first 300 observations with colors corresponding hidden states

Fit the Generated Data with a 3-State Normal HMM

To start, I summarize the data so that I can pick reasonable means and standard deviations.

```
summary = summary(x)
Q1 = as.numeric(summary[2])
med = as.numeric(summary[3])
Q3 = as.numeric(summary[5])
min = as.numeric(summary[1])
max = as.numeric(summary[6])
sd = max(c((Q1-min)/3, (max-Q3)/3))

m       <-3
mu0 <- c(Q1, med, Q3)
sigma0 <-c(sd, sd, sd)
gamma0  <-matrix(c(0.9 ,0.05 ,0.05,
                   0.05 ,0.9 ,0.05,
                   0.05 ,0.05 ,0.9),m,m,byrow=TRUE)
delta0 <- c(1/3,1/3,1/3)
summary
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.022  14.294  17.728  17.053  19.880  30.521
```

I choose $\boldsymbol{\mu}_0$ according to the 1st quartile, median, and 3rd quartile. I set the standard deviation to the max difference between the 3rd quartile and the maximum and the 1st quartile and the minimum divided by 3. That is, all data should be within 3 sds of the chosen means. I choose $\boldsymbol{\Gamma}_0$ so that the diagonals 0.9 and the off-diagonals are 0.5 (common practice in book).

```
modnorm3s <-norm.HMM.mle(x, m, mu0, sigma0, gamma0, stationary=TRUE)
modnorm3s
```

```
## $m
## [1] 3
##
## $mu
## [1] 12.00068 18.01874 21.94713
##
## $sigma
## [1] 2.990452 1.476932 2.097478
##
## $gamma
##             [,1]      [,2]       [,3]
## [1,] 0.90386004 0.0244951 0.07164486
## [2,] 0.04696274 0.8991299 0.05390731
## [3,] 0.04151814 0.1503289 0.80815300
##
## $delta
## [1] 0.3189535 0.4386713 0.2423753
##
## $code
## [1] 1
##
```

```
## $mllk
## [1] 24560.53
##
## $AIC
## [1] 49145.07
##
## $BIC
## [1] 49231.59
```

Compute the standard error for each fitted parameter using the bootstrap method and then get the confidence interval using the Monte Carlo approach.

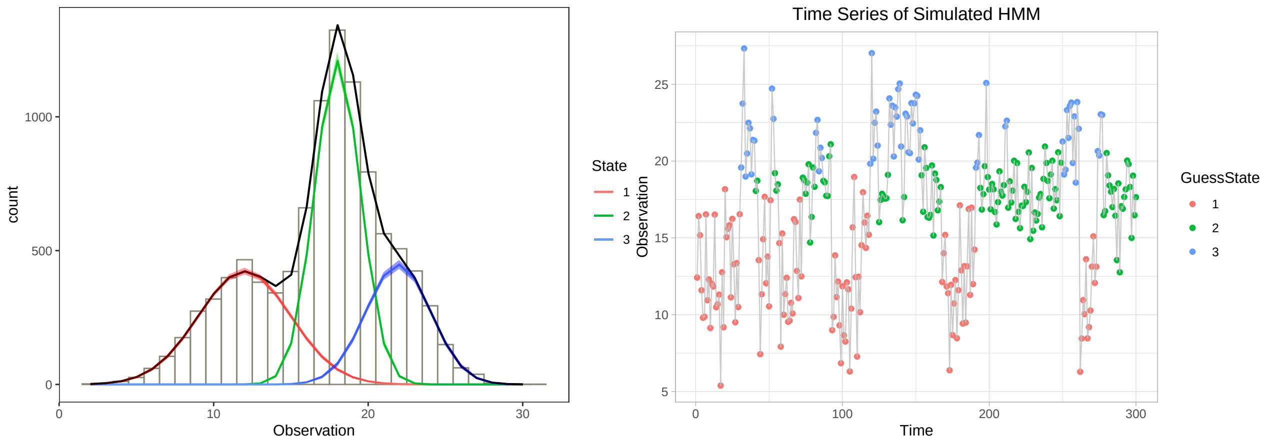
```r
SEnorm3 <- norm.HMM.params_SE(x, n=20, modnorm3s, stationary=TRUE)
SEnorm3
```

```
## $mu
## [1] 12.00068 18.01874 21.94713
##
## $mu.SE
## [1] 0.05637882 0.02425436 0.05051489
##
## $sigma
## [1] 2.990452 1.476932 2.097478
##
## $sigma.SE
## [1] 0.04365304 0.01867804 0.04023817
##
## $gamma
##            [,1]      [,2]       [,3]
## [1,] 0.90386004 0.0244951 0.07164486
## [2,] 0.04696274 0.8991299 0.05390731
## [3,] 0.04151814 0.1503289 0.80815300
##
## $gamma.SE
##             [,1]        [,2]        [,3]
## [1,] 0.004132691 0.003476699 0.004301956
## [2,] 0.003636148 0.005775998 0.004256561
## [3,] 0.004138679 0.008363985 0.008776755
##
## $delta
## [1] 0.3189535 0.4386713 0.2423753
##
## $delta.SE
## [1] 0.016257404 0.016030894 0.008735306
```

```r
minx = round(min(x))
maxx = round(max(x))
range = minx:maxx
ci.plt <- norm.HMM.CI_MonteCarlo(range, m=3, n=500, SEnorm3)
```

Now I find the underlying state sequence using global decoding by the Viterbi algorithm

```
state_seq <- norm.HMM.viterbi(x, modnorm3s)
normdata3s <- normdata
normdata3s$GuessState <- as.factor(state_seq)
```



\begin{figure}
\caption{LEFT: Histogram of observations with fitted state dependent normal distributions and their 95% confidence intervals overlayed. Black line is the marginal distribution. RIGHT: Timeseries of first 300 observations with colors corresponding fitted hidden states decoded by Viterbi algorithm.} \end{figure}

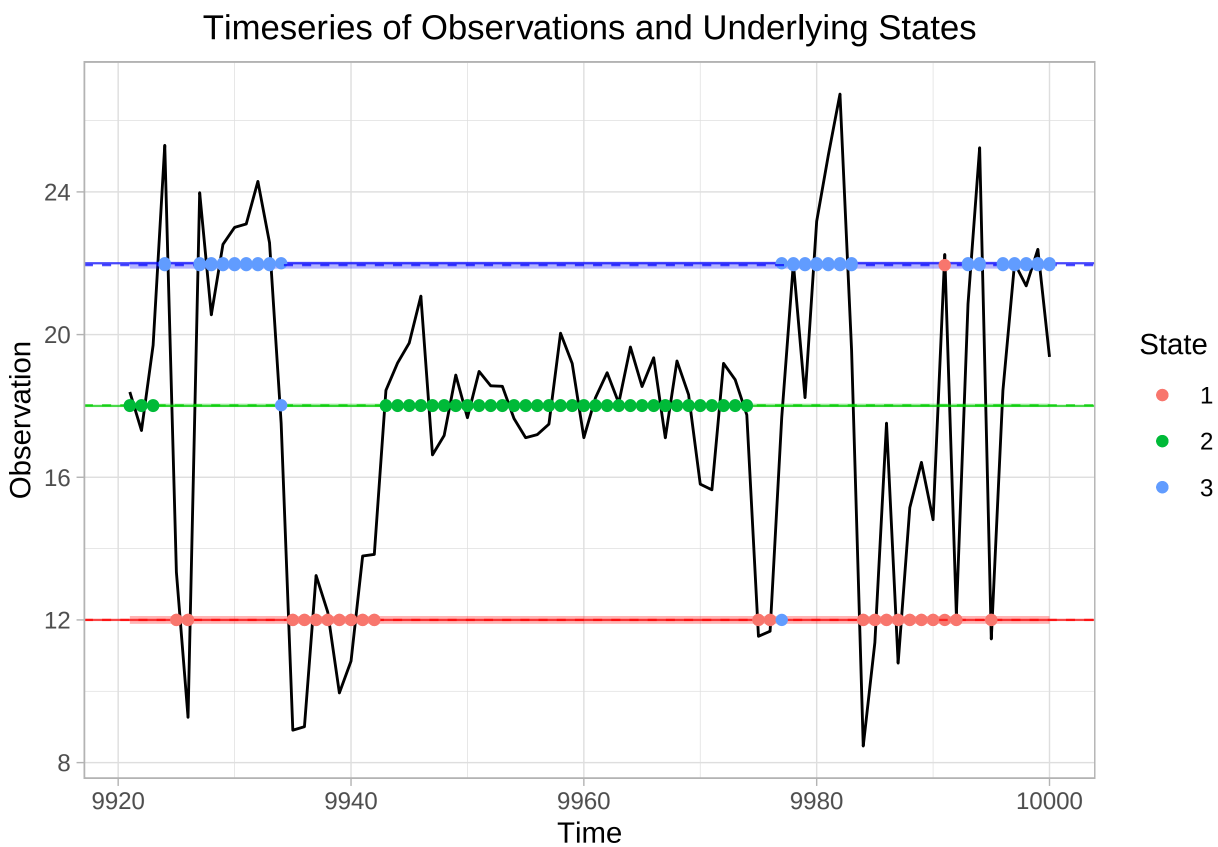


Figure 2: Timeseries of first 80 observations. The solid horizontal lines are the true means of each state dependent normal distribution and the dashed lines are the fitted means. For each time interval there are two points, one falling on the true state mean line and one on the fitted state mean line. The colors of the points correspond to the true states of the observation at that time.

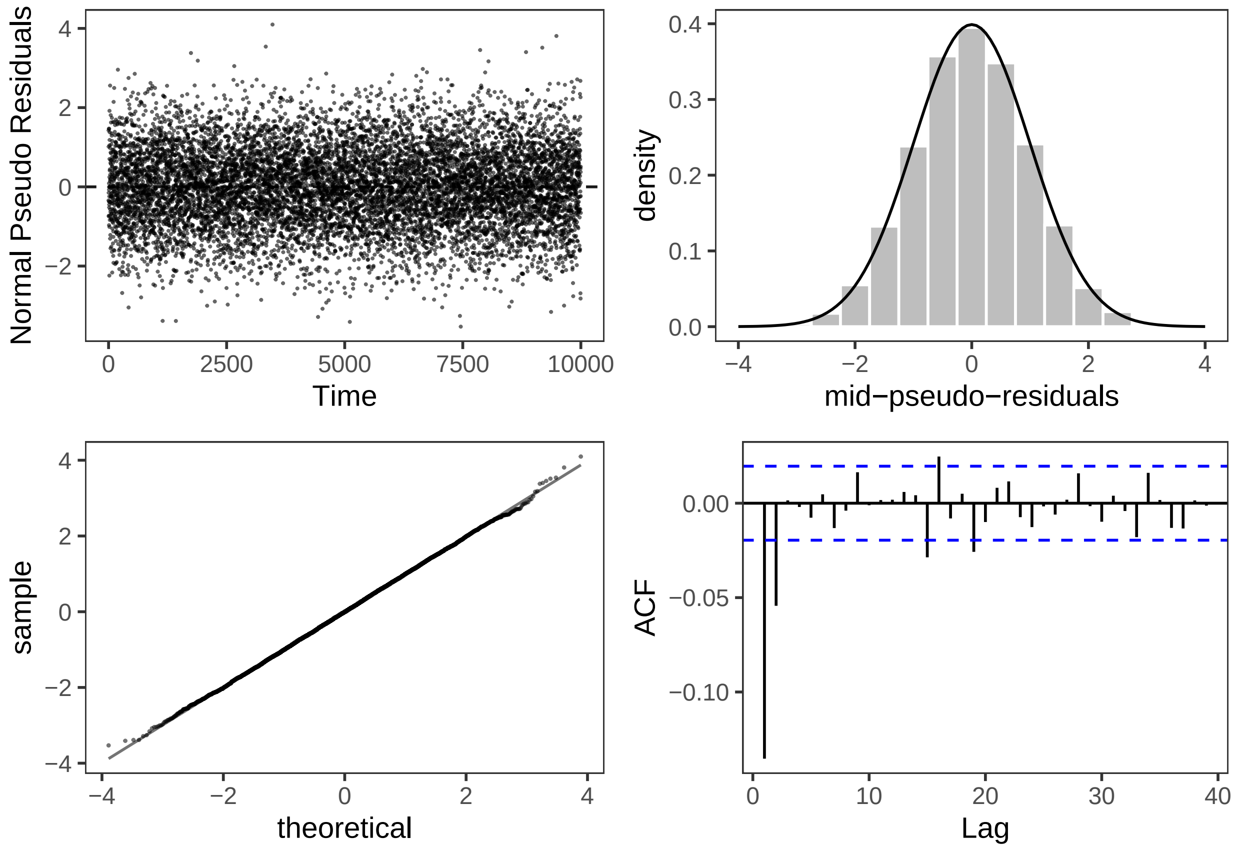


Figure 3: Plots for the normal pseudo-residuals of the fitted 3-state normal HMM. TOP LEFT: Index plot of the normal pseudo residuals. TOP RIGHT: Histogram of the normal pseudo-residuals with N(0,1) overlayed. BOTTOM LEFT: Quantile–quantile plot of the normal pseudo-residuals. BOTTOM RIGHT: Autocorrelation function of the normal pseudo-residuals.

Fit the Generated Data with a 2-State Normal HMM

Summarize the data so that I can pick reasonable means and sds.

```
sd = max(c((Q1-min)/3, (max-Q3)/3, (med-Q3)/3, (med-Q1)/3))

m       <-2
mu0 <-c(Q1,Q3)
sigma0 <- c(sd, sd)
gamma0  <-matrix(c(0.9 ,0.1,
                   0.1 ,0.9),m,m,byrow=TRUE)
delta0 <- c(1/2,1/2)
summary
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.022  14.294  17.728  17.053  19.880  30.521
```

I choose $\boldsymbol{\mu}_0$ according to the 1st quartile, and 3rd quartile. I set the standard deviation so that all data is within 3 sds of the chosen means. I choose $\boldsymbol{\Gamma}_0$ so that the diagonals 0.9 and the off-diagonals are 0.5 (common practice in book).

```r
modnorm2s <-norm.HMM.mle(x, m, mu0, sigma0, gamma0, stationary=TRUE)
modnorm2s
```

```
## $m
## [1] 2
##
## $mu
## [1] 11.92006 19.36001
##
## $sigma
## [1] 2.960955 2.600192
##
## $gamma
##            [,1]       [,2]
## [1,] 0.90501339 0.09498661
## [2,] 0.04274425 0.95725575
##
## $delta
## [1] 0.3103462 0.6896538
##
## $code
## [1] 1
##
## $mllk
## [1] 25955.6
##
## $AIC
## [1] 51923.2
##
## $BIC
## [1] 51966.46
```

Compute the standard error for each fitted parameter using the bootstrap method and then get the confidence interval using the Monte Carlo approach.

```r
SEnorm2 <- norm.HMM.params_SE(x, 20, modnorm2s, stationary=TRUE)
SEnorm2
```

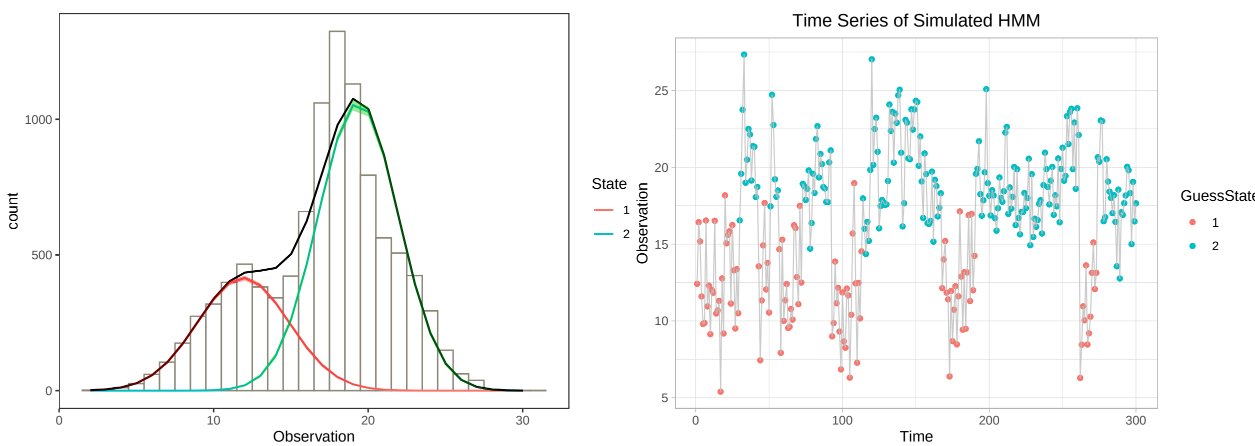
```
## $mu
## [1] 11.92006 19.36001
##
## $mu.SE
## [1] 0.04668288 0.03059284
##
## $sigma
## [1] 2.960955 2.600192
##
## $sigma.SE
## [1] 0.03341973 0.02330276
##
## $gamma
##            [,1]       [,2]
## [1,] 0.90501339 0.09498661
```

```
## [2,] 0.04274425 0.95725575
##
## $gamma.SE
##              [,1]        [,2]
## [1,] 0.005129494 0.005129494
## [2,] 0.002293147 0.002293147
##
## $delta
## [1] 0.3103462 0.6896538
##
## $delta.SE
## [1] 0.01501729 0.01501729
```

```
ci.plt2 <- norm.HMM.CI_MonteCarlo(range, m=2, n=500, SEnorm2)
```

Now I find the underlying state sequence using global decoding by the Viterbi algorithm

```
state_seq <- norm.HMM.viterbi(x, modnorm2s)
normdata2s <- normdata
normdata2s$GuessState <- as.factor(state_seq)
```



\begin{figure}
\caption{LEFT: Histogram of observations with fitted state dependent normal distributions and their 95% confidence intervals overlayed. Black line is the marginal distribution. RIGHT: Timeseries of first 300 observations with colors corresponding fitted hidden states decoded by Viterbi algorithm.} \end{figure}

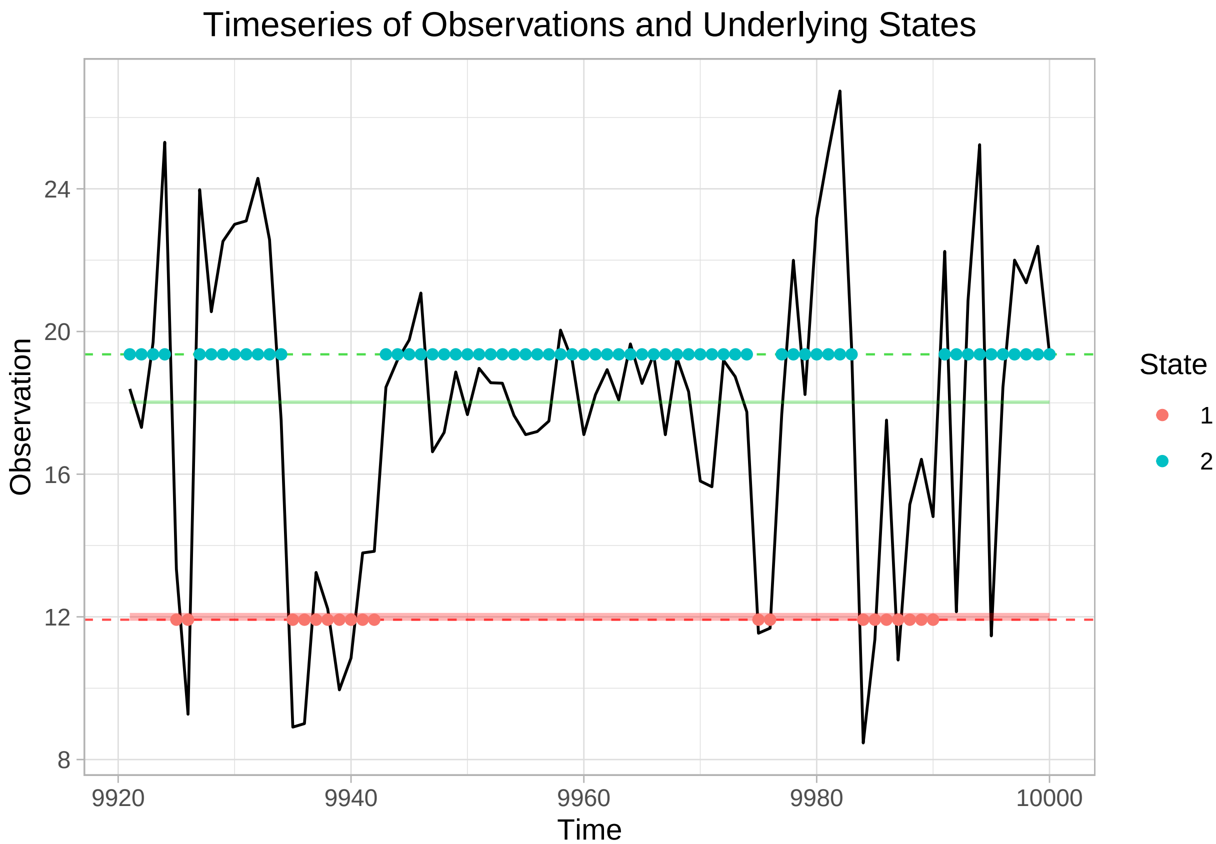


Figure 4: Timeseries of first 80 observations. The dashed lines are the fitted means for each state dependent normal distribution. The colors of the points correspond to the fitted state of the observation at that time.

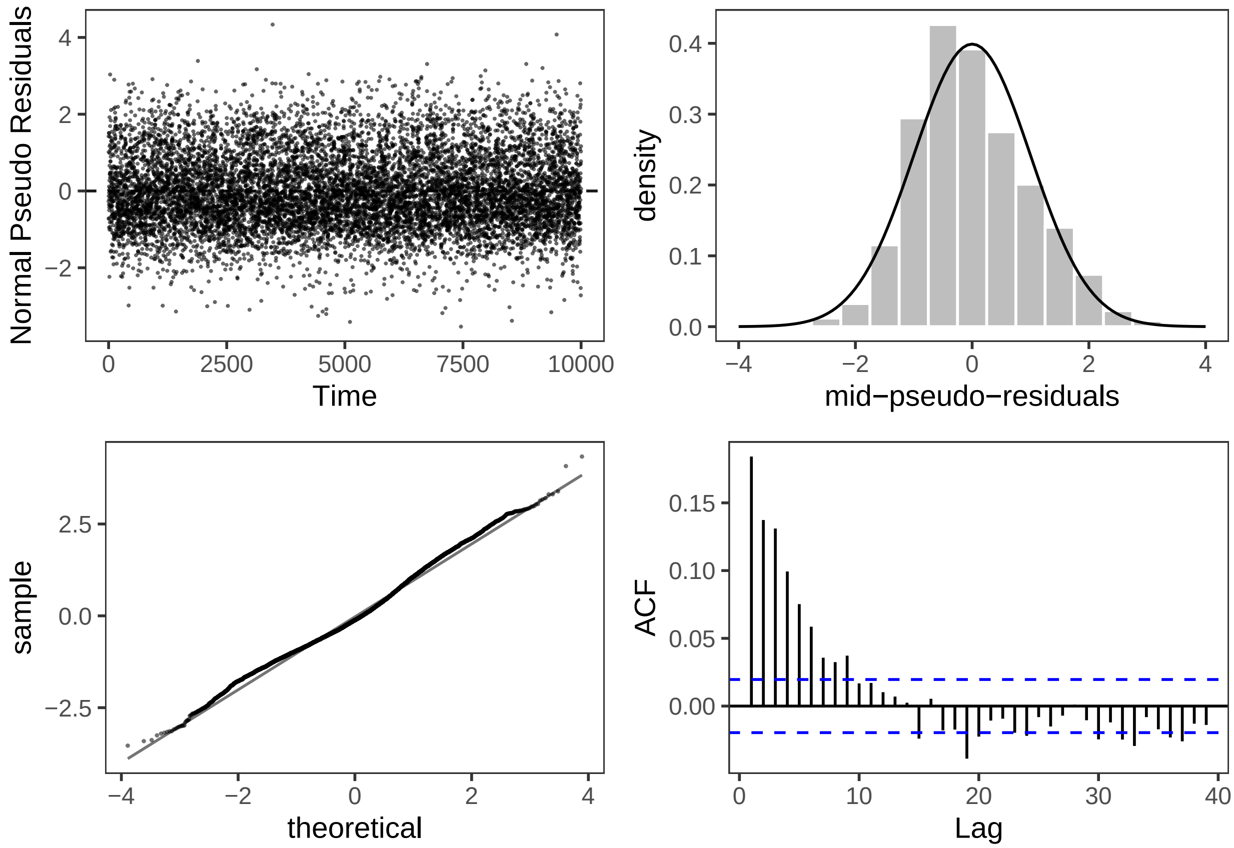


Figure 5: Plots for the normal pseudo-residuals of the fitted 2-state gamma HMM. TOP LEFT: Index plot of the normal pseudo residuals. TOP RIGHT: Histogram of the normal pseudo-residuals with N(0,1) overlayed. BOTTOM LEFT: Quantile–quantile plot of the normal pseudo-residuals. BOTTOM RIGHT: Autocorrelation function of the normal pseudo-residuals.

Fit the Generated Data with a 4-State Normal HMM

Summarize the data so that I can pick reasonable means and sds.

```
Q1 = as.numeric(summary[2])-2
med = as.numeric(summary[3])-2
Q3 = as.numeric(summary[5])-2
w = as.numeric(summary[5])+4

m       <-4
mu0 <-c(10,15,20,25)
sigma0 <- c(3, 3, 3, 3)
gamma0  <-matrix(c(0.91 ,0.03 ,0.03, 0.03,
                   0.03 ,0.91 ,0.03, 0.03,
                   0.03 ,0.03, 0.03 ,0.91),m,m,byrow=TRUE)
delta0 <- c(1/4,1/4,1/4,1/4)
summary
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.022  14.294  17.728  17.053  19.880  30.521
```

I choose $\boldsymbol{\lambda}_0$ based on the 1st quartile, median, and 3rd quartile with some variation. I choose $\boldsymbol{\Gamma}_0$ so that the diagonals 0.91 and the off-diagonals are 0.03.

```
modnorm4s <-norm.HMM.mle(x, m, mu0, sigma0, gamma0, stationary=TRUE)
modnorm4s
```

```
## $m
## [1] 4
##
## $mu
## [1] 11.99876 18.02835 21.67096 22.18208
##
## $sigma
## [1] 2.991127 1.482208 2.318048 1.886435
##
## $gamma
##             [,1]       [,2]       [,3]         [,4]
## [1,] 0.903910066 0.02473141 0.03282204 0.0385364775
## [2,] 0.048406552 0.90063664 0.04999149 0.0009653144
## [3,] 0.002161766 0.08400593 0.01772685 0.8961054605
## [4,] 0.071679666 0.20536102 0.63290472 0.0900545972
##
## $delta
## [1] 0.3186991 0.4392437 0.1149210 0.1271361
##
## $code
## [1] 4
##
## $mllk
## [1] 24555.56
##
## $AIC
## [1] 49151.12
##
## $BIC
## [1] 49295.33
```

Compute the standard error for each fitted parameter using the bootstrap method and then get the confidence interval using the Monte Carlo approach.

```
SEnorm4 <- norm.HMM.params_SE(x, 20, modnorm4s, stationary=TRUE)
SEnorm4
```

```
## $mu
## [1] 11.99876 18.02835 21.67096 22.18208
##
## $mu.SE
## [1] 0.04959569 0.01770131 0.26303714 0.12349370
##
## $sigma
## [1] 2.991127 1.482208 2.318048 1.886435
##
## $sigma.SE
```

```
## [1] 0.04487632 0.02153761 0.07789560 0.07137426
##
## $gamma
##              [,1]       [,2]       [,3]         [,4]
## [1,] 0.903910066 0.02473141 0.03282204 0.0385364775
## [2,] 0.048406552 0.90063664 0.04999149 0.0009653144
## [3,] 0.002161766 0.08400593 0.01772685 0.8961054605
## [4,] 0.071679666 0.20536102 0.63290472 0.0900545972
##
## $gamma.SE
##              [,1]        [,2]        [,3]        [,4]
## [1,] 0.005390176 0.003411182 0.013558337 0.013497005
## [2,] 0.003253452 0.006072202 0.005162097 0.002427662
## [3,] 0.004442179 0.051028879 0.020334440 0.061197343
## [4,] 0.010263637 0.033364516 0.172444331 0.179484502
##
## $delta
## [1] 0.3186991 0.4392437 0.1149210 0.1271361
##
## $delta.SE
## [1] 0.01090504 0.01463869 0.01990915 0.01650267
```

```r
ci.plt4 <- norm.HMM.CI_MonteCarlo(range, m, n=500, SEnorm4)
```

Now I find the underlying state sequence using global decoding by the Viterbi algorithm

```r
state_seq <- norm.HMM.viterbi(x, modnorm4s)
normdata4s <- normdata
normdata4s$GuessState <- as.factor(state_seq)
```

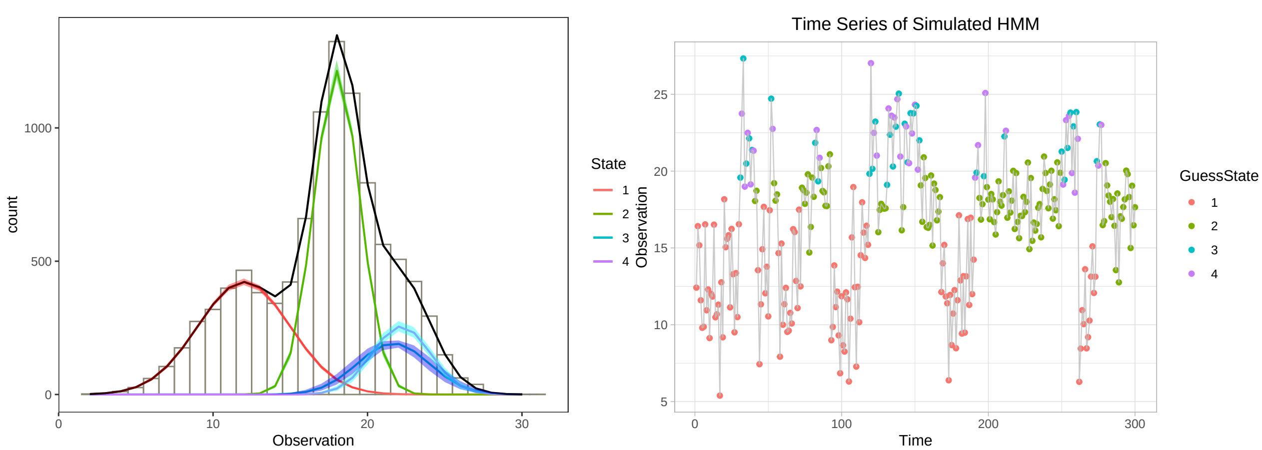


Figure 6: LEFT: Histogram of observations with fitted state dependent normal distributions overlayed. Black line is the marginal distribution. RIGHT: Timeseries of first 300 observations with colors corresponding fitted hidden states decoded by Viterbi algorithm.

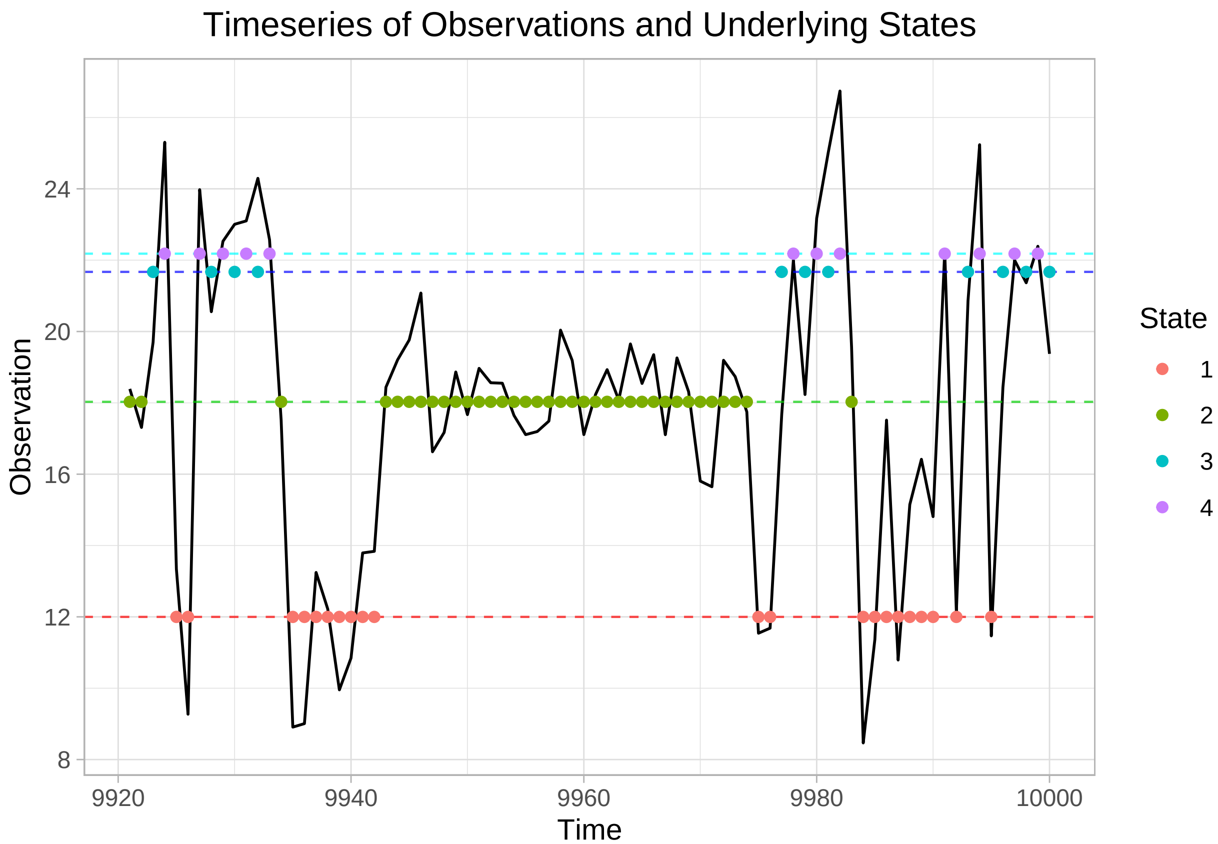


Figure 7: Timeseries of first 80 observations. The dashed lines are the fitted means for each state dependent normal distribution. The colors of the points correspond to the fitted state of the observation at that time.

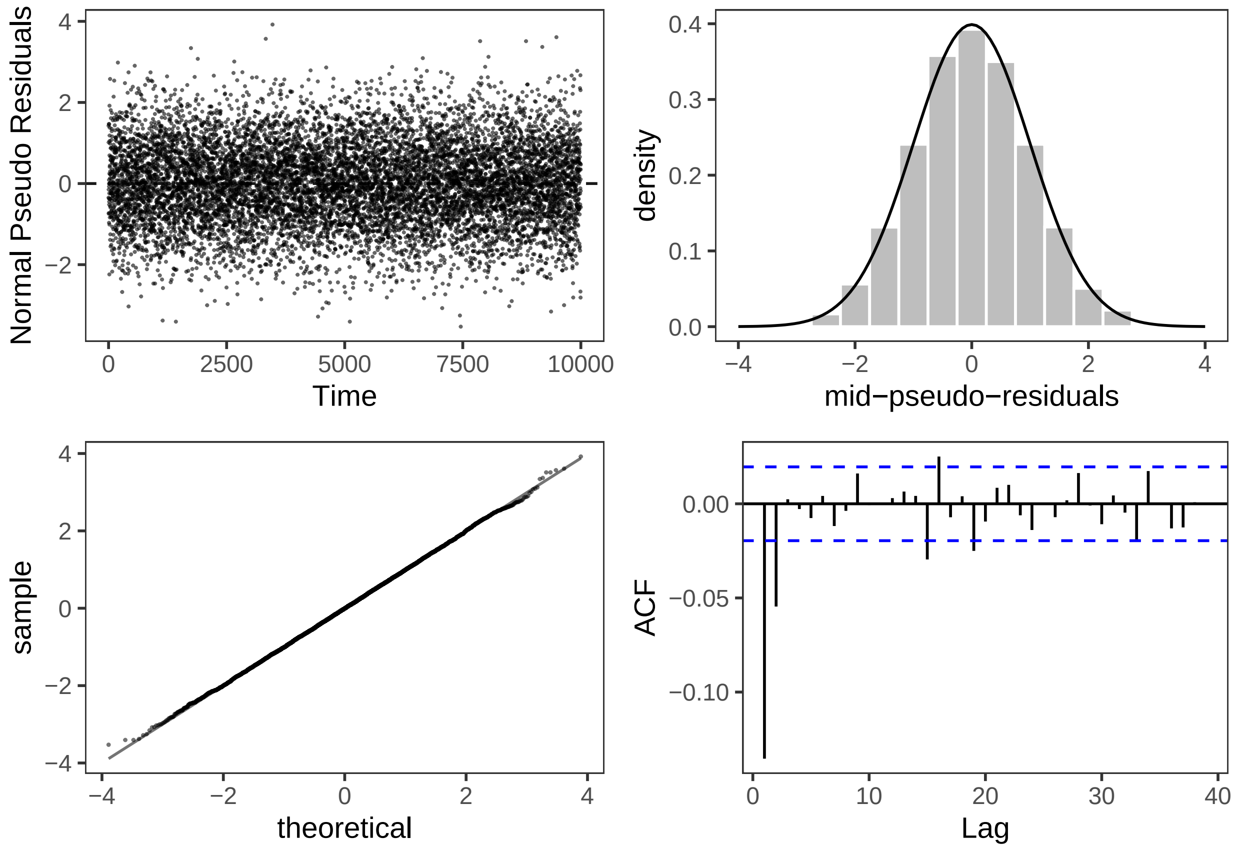


Figure 8: Plots for the normal pseudo-residuals of the fitted 4-state normal HMM. TOP LEFT: Index plot of the normal pseudo residuals. TOP RIGHT: Histogram of the normal pseudo-residuals with N(0,1) overlayed. BOTTOM LEFT: Quantile–quantile plot of the normal pseudo-residuals. BOTTOM RIGHT: Autocorrelation function of the normal pseudo-residuals.

Fit the Generated Data with a 3-State Gamma HMM

To start, I summarize the data so that I can pick reasonable shapes and scales.

```
summary = summary(x)
mean = mean(x)

m        <-3
alpha0 <- c(Q1, med, Q3)
theta0 <-c(1,1,1)
gamma0  <-matrix(c(0.9 ,0.05 ,0.05,
                   0.05 ,0.9 ,0.05,
                   0.05 ,0.05 ,0.9),m,m,byrow=TRUE)
delta0 <- c(1/3,1/3,1/3)
summary
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.022  14.294  17.728  17.053  19.880  30.521
```

I choose $\boldsymbol{\alpha}_0$ to be equidistant from the mean of the observations. I set $\boldsymbol{\theta}_0$ to 1 for all states. I choose $\boldsymbol{\Gamma}_0$ so that the diagonals are 0.9 and the off-diagonals are 0.5 (common practice in book).

```r
modgam3s <-gam.HMM.mle(x, m, alpha0, theta0, gamma0, stationary=TRUE)
modgam3s
```

```
## $m
## [1] 3
##
## $alpha
## [1]  14.12361 115.13277 145.83458
##
## $theta
## [1] 0.8565430 0.1912171 0.1236547
##
## $gamma
##            [,1]       [,2]       [,3]
## [1,] 0.90765494 0.06794325 0.02440181
## [2,] 0.03815787 0.80742832 0.15441381
## [3,] 0.04726205 0.05286576 0.89987219
##
## $delta
## [1] 0.3231949 0.2352482 0.4415569
##
## $code
## [1] 4
##
## $mllk
## [1] 24624.52
##
## $AIC
## [1] 49273.05
##
## $BIC
## [1] 49359.57
```

Compute the standard error for each fitted parameter using the bootstrap method and then get the confidence interval using the Monte Carlo approach.

```r
SEgam3 = gam.HMM.params_SE(x, 20, modgam3s, stationary=TRUE)
SEgam3
```

```
## $alpha
## [1]  14.12361 115.13277 145.83458
##
## $alpha.SE
## [1] 0.4410194 4.3475434 4.7659523
##
## $theta
## [1] 0.8565430 0.1912171 0.1236547
##
## $theta.SE
## [1] 0.026965416 0.006574585 0.004027771
##
## $gamma
```

```
##              [,1]       [,2]       [,3]
## [1,] 0.90765494 0.06794325 0.02440181
## [2,] 0.03815787 0.80742832 0.15441381
## [3,] 0.04726205 0.05286576 0.89987219
##
## $gamma.SE
##               [,1]        [,2]        [,3]
## [1,] 0.006101848 0.005886118 0.002957466
## [2,] 0.004352959 0.006997827 0.006165785
## [3,] 0.003129997 0.004446160 0.004116675
##
## $delta
## [1] 0.3231949 0.2352482 0.4415569
##
## $delta.SE
## [1] 0.01737709 0.01228091 0.01131267
```

```
ci.plt.gam3 <- gam.HMM.CI_MonteCarlo(range, m, n=500, SEgam3)
```

Now I find the underlying state sequence using global decoding by the Viterbi algorithm

```
state_seq <- gam.HMM.viterbi(x, modgam3s)
gamdata3s <- normdata
gamdata3s$GuessState <- as.factor(state_seq)
```

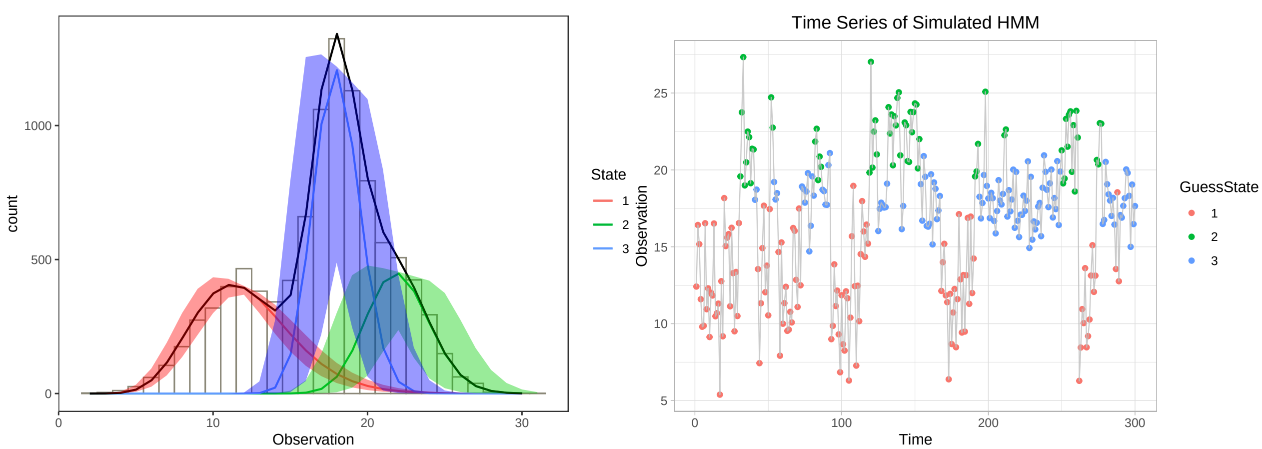


Figure 9: LEFT: Histogram of observations with fitted state dependent gamma distributions overlayed. Black line is the marginal distribution. RIGHT: Timeseries of first 300 observations with colors corresponding fitted hidden states decoded by Viterbi algorithm.

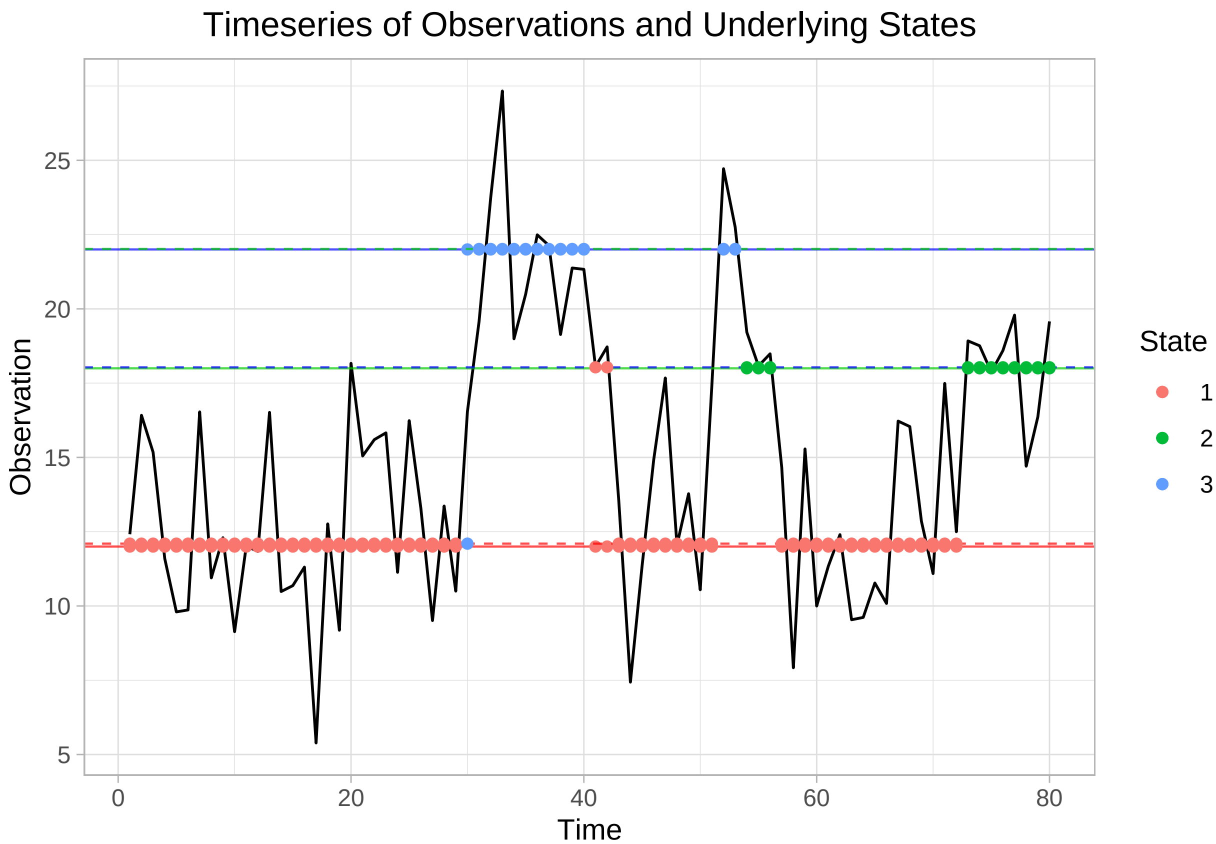


Figure 10: Timeseries of first 80 observations. The solid horizontal lines are the true means (alpha*theta) of each state dependent gamma distribution and the dashed lines are the fitted means. For each time interval there are two points, one falling on the true state mean line and one on the fitted state mean line. The colors of the points correspond to the true states of the observation at that time.

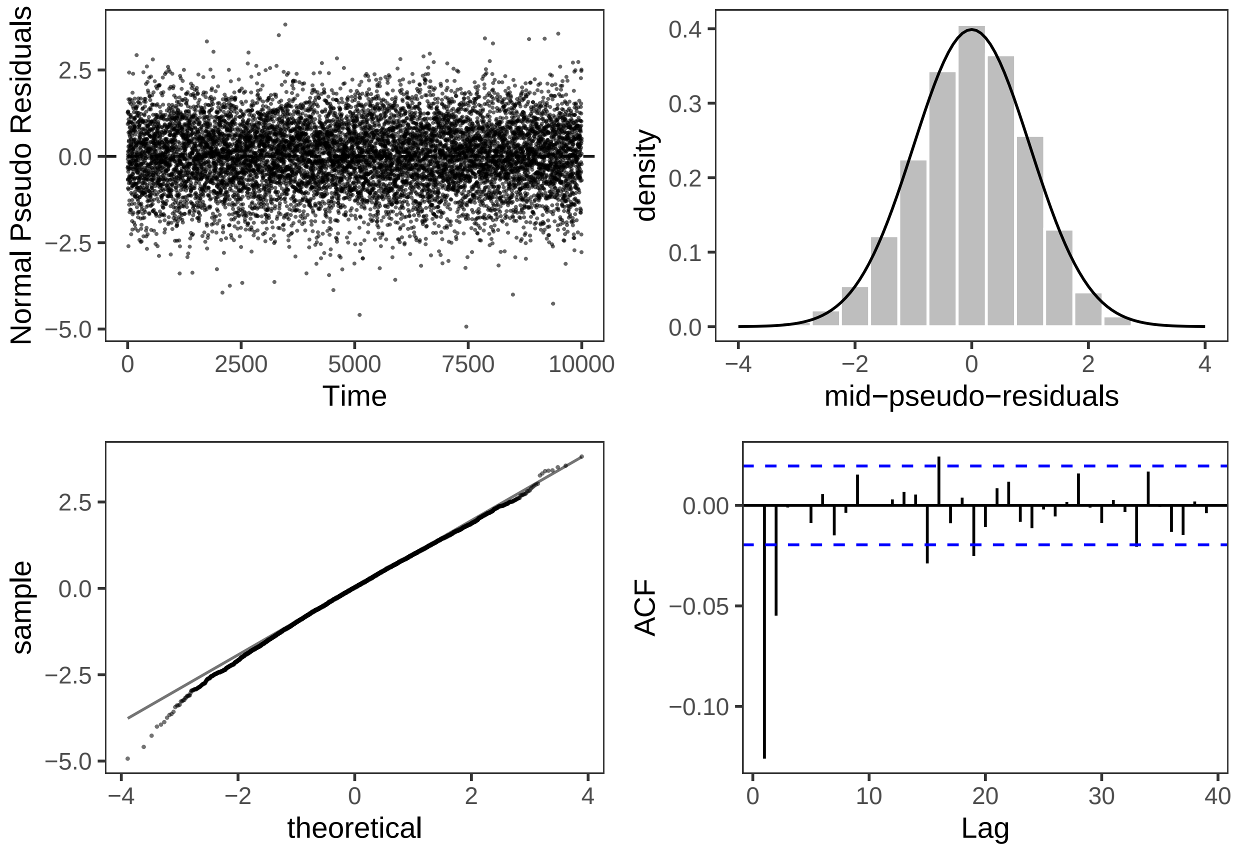


Figure 11: Plots for the normal pseudo-residuals of the fitted 3-state gamma HMM. TOP LEFT: Index plot of the normal pseudo residuals. TOP RIGHT: Histogram of the normal pseudo-residuals with N(0,1) overlayed. BOTTOM LEFT: Quantile–quantile plot of the normal pseudo-residuals. BOTTOM RIGHT: Autocorrelation function of the normal pseudo-residuals.

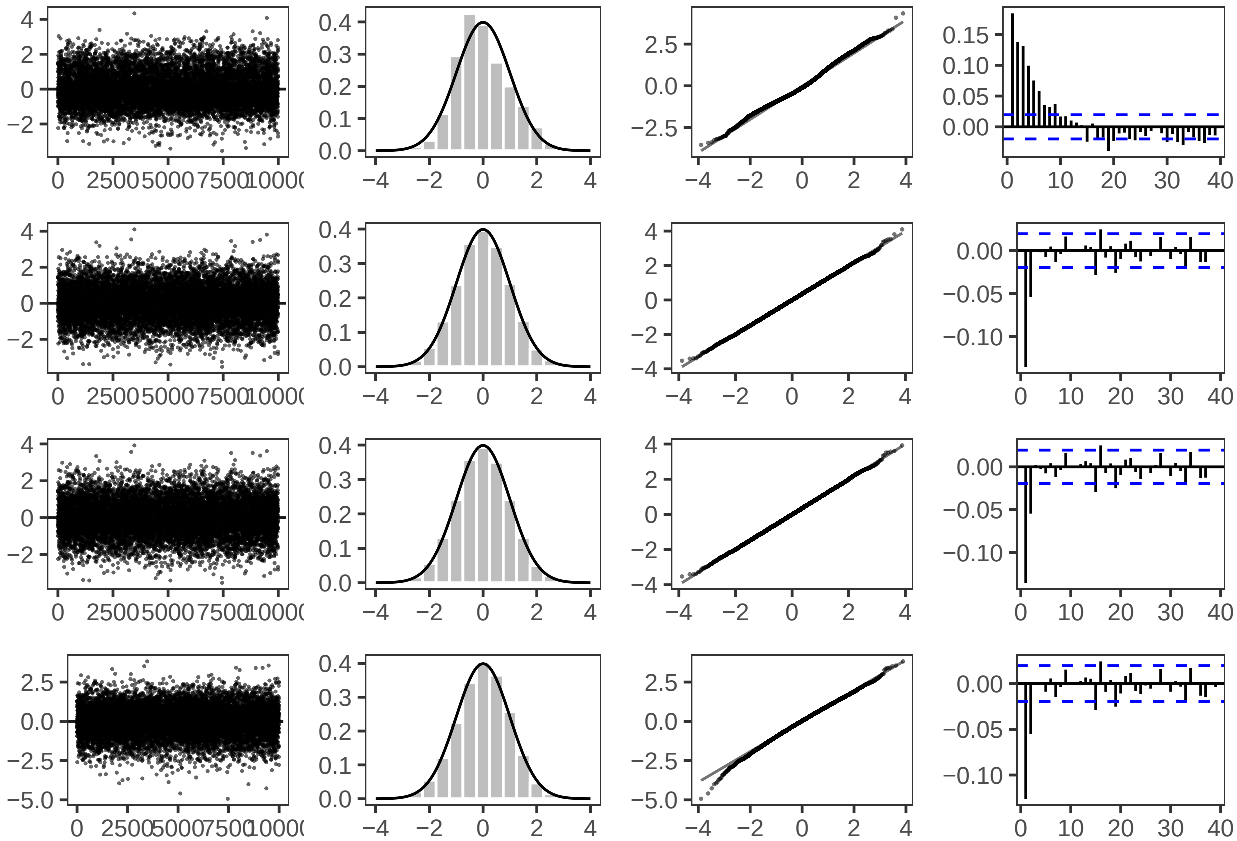


Figure 12: Each row shows plots of the pseudo residuals for 2, 3, and 4 state HMMs with normal state dependent distributions respectively. The first column shows index plots of the normal pseudo-residuals. The second column shows the histograms of the normal pseudo-residuals with N(0,1) overlayed. The third column shows the quantile–quantile plots of the normal pseudo-residuals. The fourth column shows the autocorrelation functions of the normal pseudo-residuals