Fitting Gamma HMMs

Simone

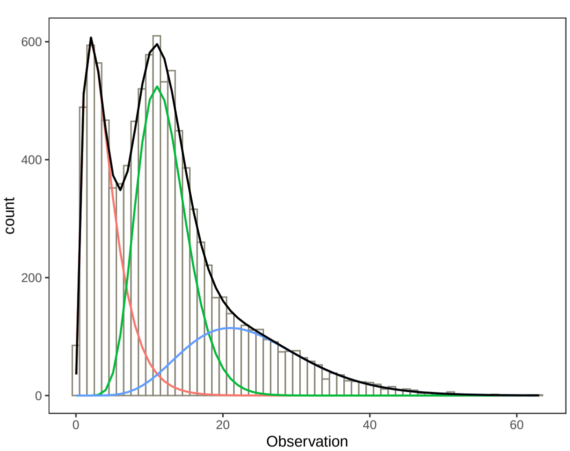
24/05/2021

Generate Data from a 3-State Gamma HMM

We have shapes $\boldsymbol{\alpha}$ and scales $\boldsymbol{\theta}$ for the state dependent gamma distributions. We have transition probability $\boldsymbol{\Gamma}$ and stationary distribution $\boldsymbol{\delta}$.

$$\boldsymbol{\alpha} = (2,\ 12,\ 8), \quad \boldsymbol{\theta} = (2,\ 1,\ 3), \quad \boldsymbol{\Gamma} = \begin{pmatrix} 0.9 & 0.03 & 0.07 \\ 0.050 & 0.899 & 0.051 \\ 0.05 & 0.15 & 0.80 \end{pmatrix}, \quad \boldsymbol{\delta} = (0.3333333,\ 0.4382470,\ 0.2284197)$$

```r
T=10000
gam3s <- list(m = 3,
              alpha = c(2, 12, 8),
              theta = c(2, 1, 3),
              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 <- gam.HMM.generate_sample(T, gam3s)
gamdata = data.frame(Time = c(1:T),
                     Observation = sample$observ,
                     State = factor(sample$state))
x <- gamdata$Observation
```



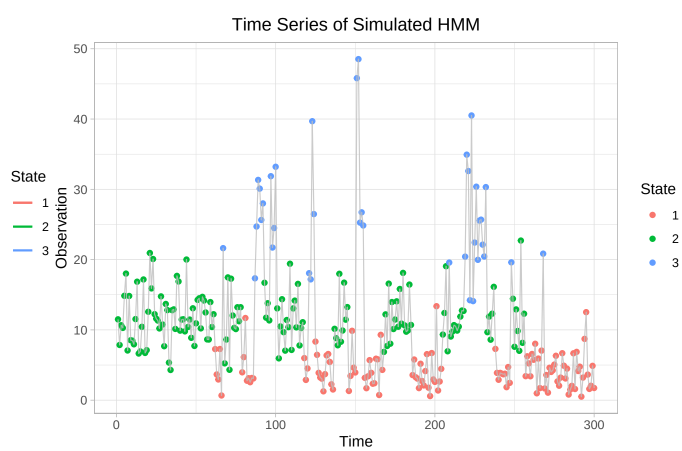


Figure 1: LEFT: Histogram of observations with state dependent gamma distributions overlayed. Black line is the marginal distribution. RIGHT: Timeseries of first 300 observations with colors corresponding hidden states

Fit the Generated Data with a 3-State Gamma HMM

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

```
summary = summary(x)
mean = mean(x)
Q1 = as.numeric(summary[2])
Q2 = as.numeric(summary[5])
m       <-3
alpha0 <- c(Q1, mean, Q2)
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)
summary
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.04427  5.30798 10.70416 12.08399 15.83421 63.20622
```

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 0.9 and the off-diagonals are 0.5 (common practice in book).

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

```
## $m
## [1] 3
##
## $alpha
## [1]  2.025627 11.782847  7.674815
##
## $theta
## [1] 1.976838 1.016791 3.114131
##
## $gamma
##            [,1]       [,2]       [,3]
## [1,] 0.89573064 0.02816912 0.07610024
## [2,] 0.05116238 0.89819551 0.05064212
## [3,] 0.05184991 0.15436614 0.79378396
##
## $delta
## [1] 0.3301799 0.4399350 0.2298851
##
## $code
## [1] 1
##
## $mllk
## [1] 30576.34
##
## $AIC
## [1] 61176.68
##
## $BIC
## [1] 61263.2
```

Find the standard error for each fitted paramter using the parametric bootstrap then compute the confidence intervals using the Monte Carlo method.

```
SEgam3 = gam.HMM.params_SE(x, 10, modgam3s, stationary = TRUE)
minx = round(min(x))
maxx = round(max(x))
range = minx:maxx
ci.plt <- gam.HMM.CI_MonteCarlo(range, m=3, 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 <- gamdata
gamdata3s$GuessState <- as.factor(state_seq)
```

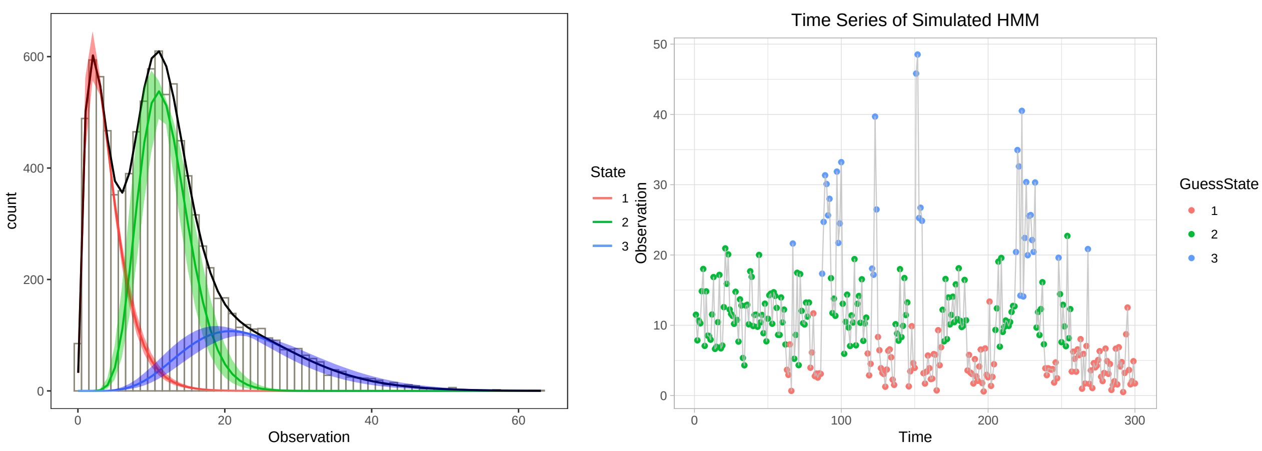


Figure 2: LEFT: Histogram of observations with fitted state dependent gamma distributions overlayed. Black line is the marginal distribution. RIGHT: Timeseries of first 300 observations with colors corresponding fitted hidden states decoded by Viterbi algorithm.

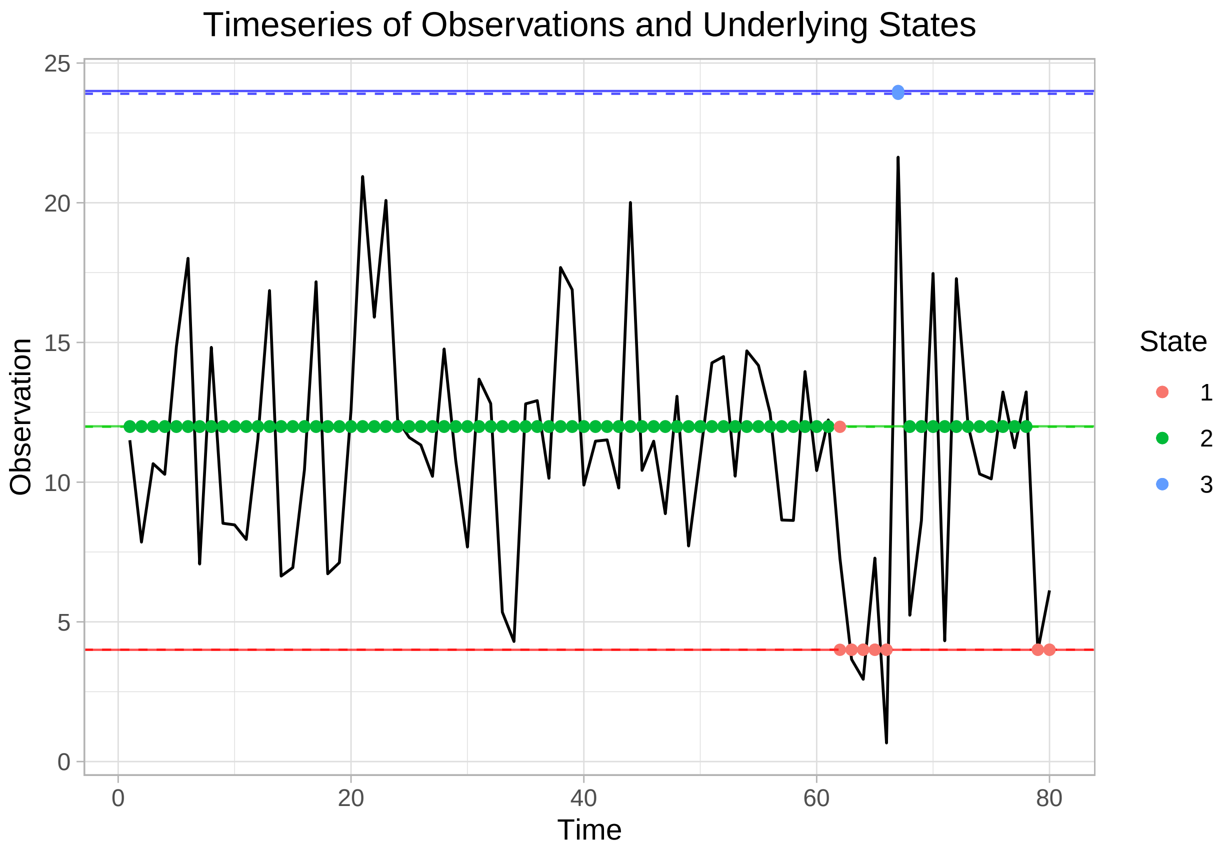


Figure 3: Timeseries of first 80 observations. The solid horizontal lines are the true means (alpha*theta) of each state dependent gamma distribution and the dashed lines are the fitted means. For each time interval there are two points, one falling on the true state mean line and one on the fitted state mean line. The colors of the points correspond to the true states of the observation at that time.

Now we can compute the normal pseudo-residuals

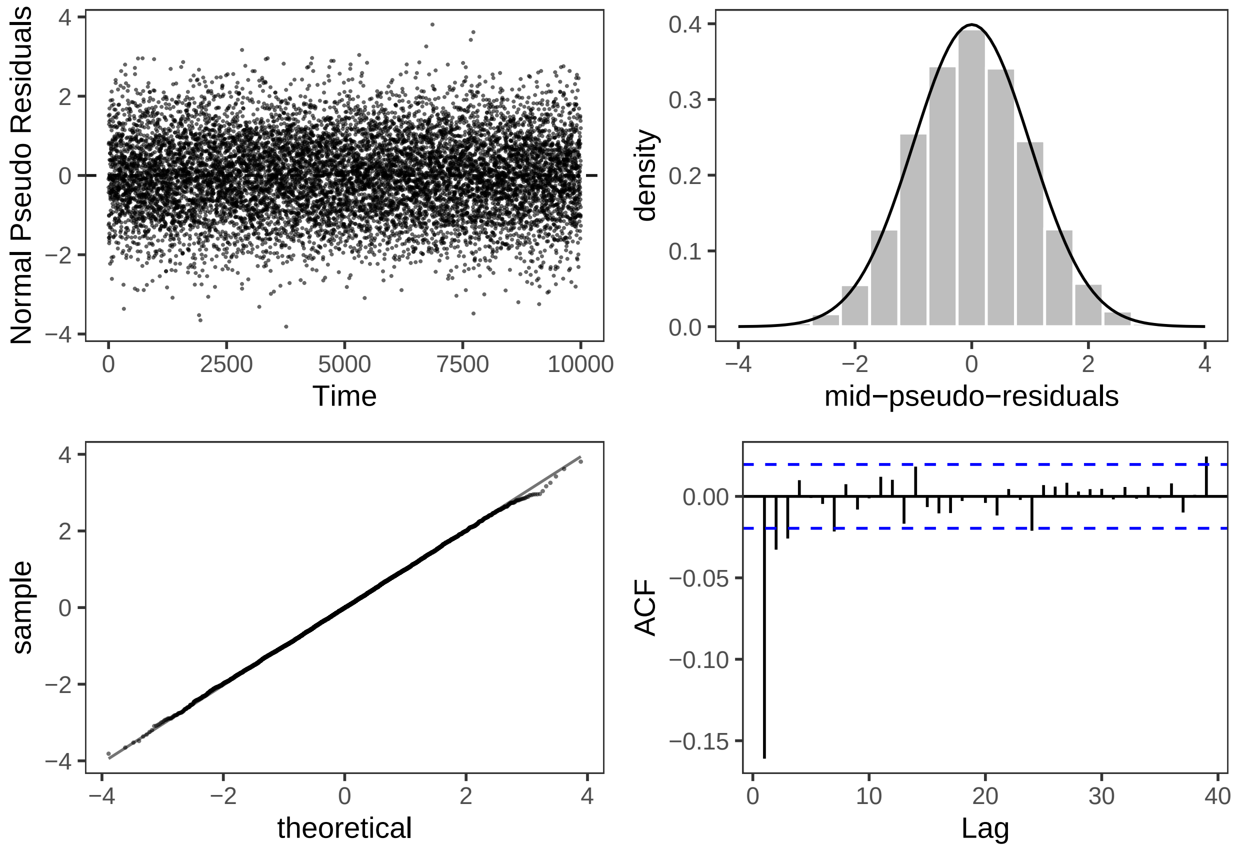


Figure 4: Plots for the normal pseudo-residuals of the fitted 3-state gamma HMM. TOP LEFT: Index plot of the normal pseudo residuals. TOP RIGHT: Histogram of the normal pseudo-residuals with N(0,1) overlayed. BOTTOM LEFT: Quantile–quantile plot of the normal pseudo-residuals. BOTTOM RIGHT: Autocorrelation function of the normal pseudo-residuals.

Fit the Generated Data with a 2-State Gamma HMM

Summarize the data so that I can pick reasonable shapes and scales.

```
m       <- 2
alpha0 <- c(Q1, Q2)
theta0 <-c(1,1)
gamma0  <-matrix(c(0.9 ,0.1,
                   0.1 ,0.9),m,m,byrow=TRUE)
summary
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.04427  5.30798 10.70416 12.08399 15.83421 63.20622
```

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 0.9 and the off-diagonals are 0.1.

```
modgam2s <-gam.HMM.mle(x, m, alpha0, theta0, gamma0, stationary=TRUE)
modgam2s
```

```
## $m
## [1] 2
##
## $alpha
## [1] 2.020764 4.518582
##
## $theta
## [1] 1.944952 3.518771
##
## $gamma
##           [,1]      [,2]
## [1,] 0.8994778 0.1005222
## [2,] 0.0469682 0.9530318
##
## $delta
## [1] 0.3184493 0.6815507
##
## $code
## [1] 1
##
## $mllk
## [1] 31845.03
##
## $AIC
## [1] 63702.06
##
## $BIC
## [1] 63745.32
```

Find the standard error for each fitted paramter using the parametric bootstrap then compute the confidence intervals using the Monte Carlo method.

```
SEgam2 = gam.HMM.params_SE(x, 10, modgam2s, stationary = TRUE)
ci.plt <- gam.HMM.CI_MonteCarlo(range, m, n=500, SEgam2)
```

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

```
state_seq <- gam.HMM.viterbi(x, modgam2s)
gamdata2s <- gamdata
gamdata2s$GuessState <- as.factor(state_seq)
```

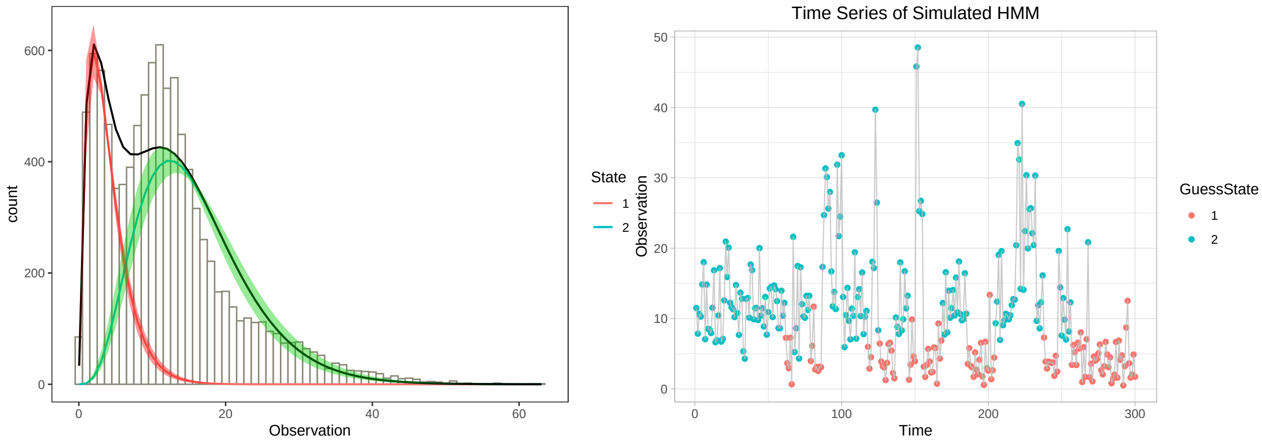


Figure 5: LEFT: Histogram of observations with fitted state dependent gamma distributions overlayed. Black line is the marginal distribution. RIGHT: Timeseries of first 300 observations with colors corresponding fitted hidden states decoded by Viterbi algorithm.

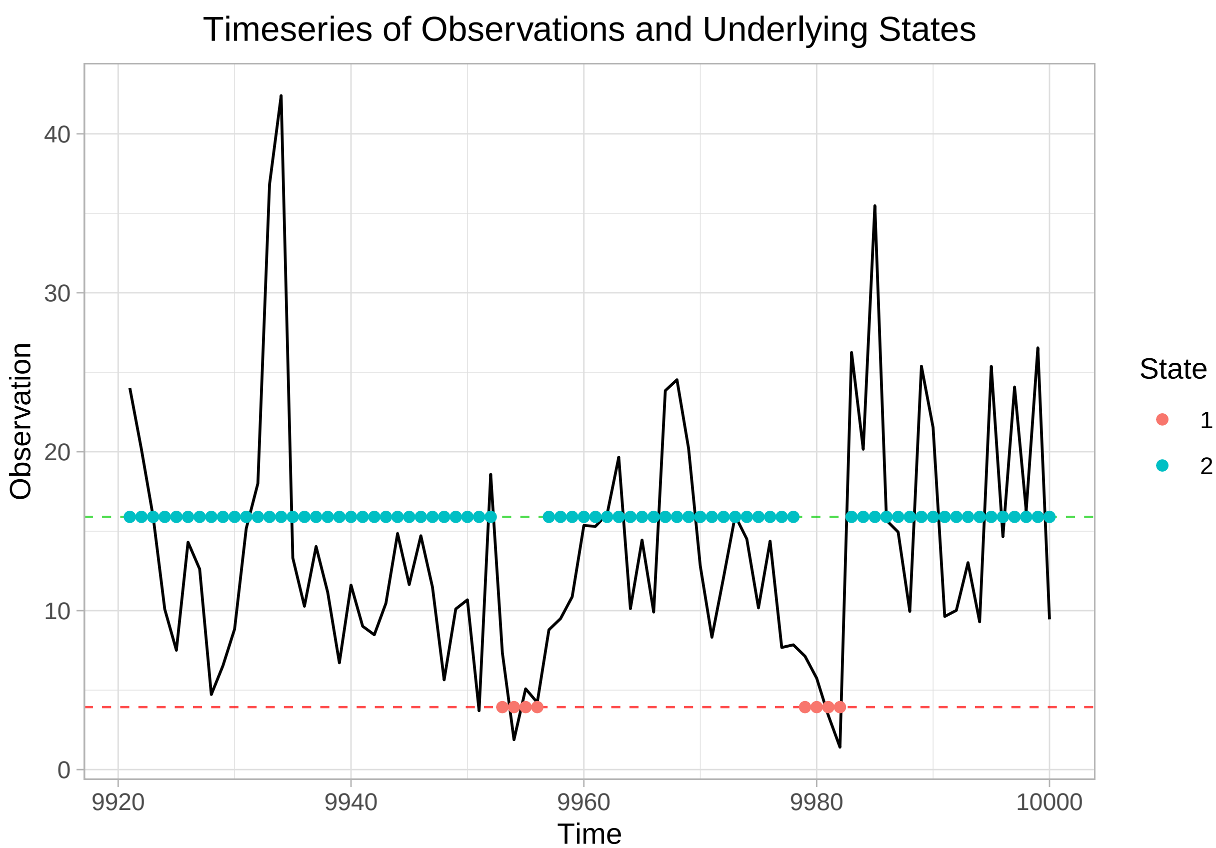


Figure 6: Timeseries of first 80 observations. The dashed lines are the fitted means (alpha*theta) for each state dependent gamma distribution. The colors of the points correspond to the fitted state of the observation at that time.

```
gamdata2s$PR <- gam.HMM.pseudo_residuals(x, modgam2s)
gamdata2.pr <- filter(gamdata2s, PR != -Inf)
```

```
gamdata2.pr <- filter(gamdata2.pr, PR != Inf)
index2 <- pr.plot.cont(gamdata2.pr)
hist2 <- pr.hist(gamdata2.pr)
qq2 <- pr.qq(gamdata2.pr)
acf2 <- pr.acf(gamdata2.pr$PR)
grid.arrange(index2,hist2,qq2,acf2, ncol=2)
```

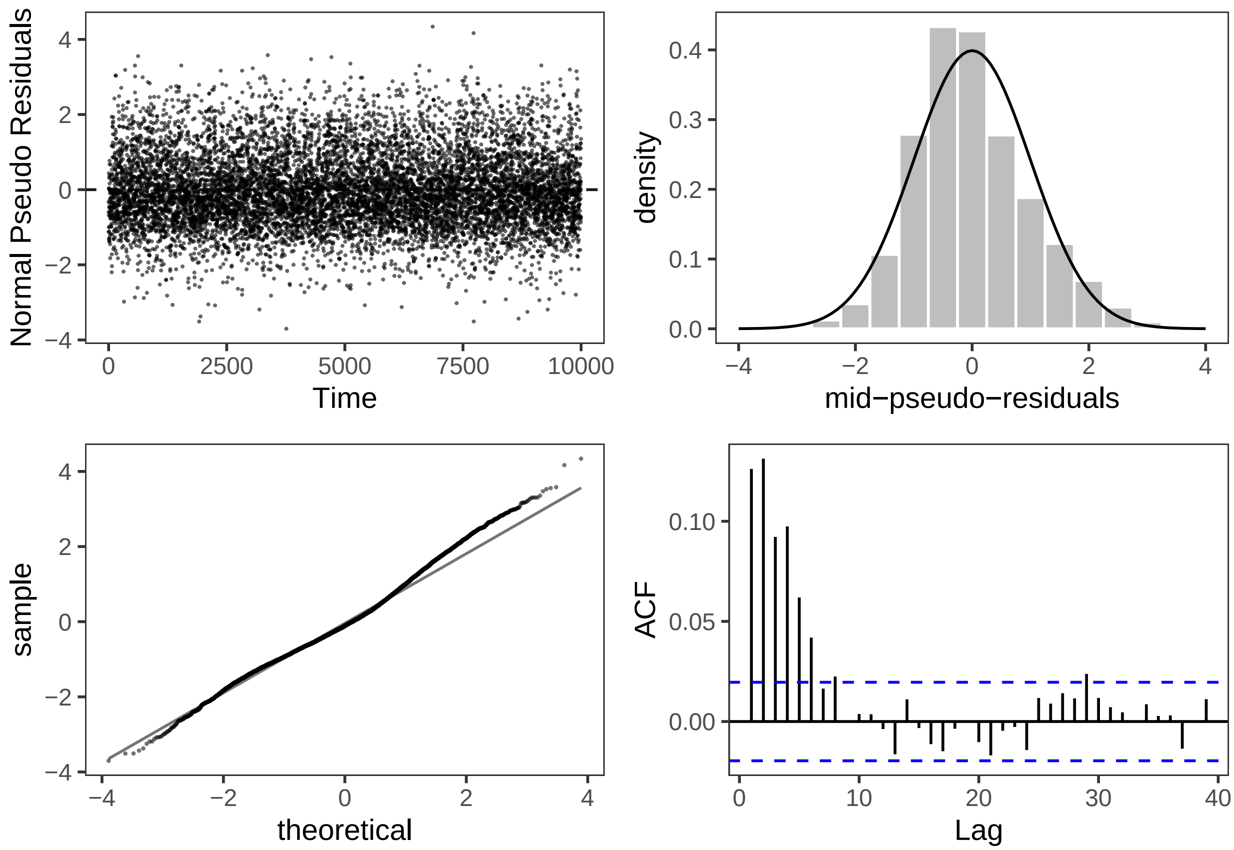


Figure 7: Plots for the normal pseudo-residuals of the fitted 2-state gamma HMM. TOP LEFT: Index plot of the normal pseudo residuals. TOP RIGHT: Histogram of the normal pseudo-residuals with N(0,1) overlayed. BOTTOM LEFT: Quantile–quantile plot of the normal pseudo-residuals. BOTTOM RIGHT: Autocorrelation 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.

```
min = as.numeric(summary[1])
max = as.numeric(summary[6])
diff = max-min
d = diff/5

m       <-4
alpha0 <- c(3, 12, 20, 30)
theta0 <- c(1,1,1,1)
```

```r
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)
summary
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.04427  5.30798 10.70416 12.08399 15.83421 63.20622
```

I choose $\boldsymbol{\alpha}_0$ by looking at the histogram and inferring the means for four possible states. I set $\boldsymbol{\theta}_0$ to 1 for all states. I choose $\boldsymbol{\Gamma}_0$ so that the diagonals 0.91 and the off-diagonals are 0.03.

```r
modgam4s <-gam.HMM.mle(x, m, alpha0, theta0, gamma0, stationary=TRUE)
modgam4s
```

```
## $m
## [1] 4
##
## $alpha
## [1]  2.024550 11.743501  7.896274  8.170763
##
## $theta
## [1] 1.978789 1.021106 2.844901 3.100434
##
## $gamma
##            [,1]       [,2]       [,3]       [,4]
## [1,] 0.89589667 0.02879521 0.02822886 0.04707925
## [2,] 0.05132666 0.89861192 0.03126578 0.01879565
## [3,] 0.03653581 0.07320810 0.04912714 0.84112895
## [4,] 0.06442285 0.22520614 0.62180334 0.08856767
##
## $delta
## [1] 0.3306102 0.4420481 0.1046086 0.1227330
##
## $code
## [1] 4
##
## $mllk
## [1] 30574.26
##
## $AIC
## [1] 61188.52
##
## $BIC
## [1] 61332.72
```

Find the standard error for each fitted paramter using the parametric bootstrap then compute the confidence intervals using the Monte Carlo method.

```r
SEgam4 = gam.HMM.params_SE(x, 10, modgam4s, stationary = TRUE)
ci.plt <- gam.HMM.CI_MonteCarlo(range, m, n=500, SEgam4)
```

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

```r
state_seq <- gam.HMM.viterbi(x, modgam4s)
gamdata4s <- gamdata
gamdata4s$GuessState <- as.factor(state_seq)
```

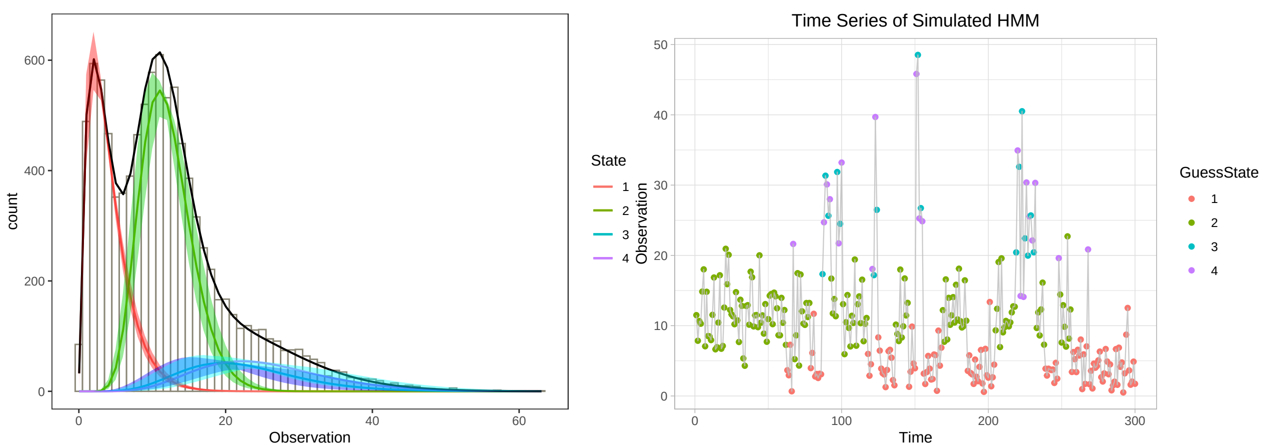


Figure 8: LEFT: Histogram of observations with fitted state dependent gamma distributions overlayed. Black line is the marginal distribution. RIGHT: Timeseries of first 300 observations with colors corresponding fitted hidden states decoded by Viterbi algorithm.

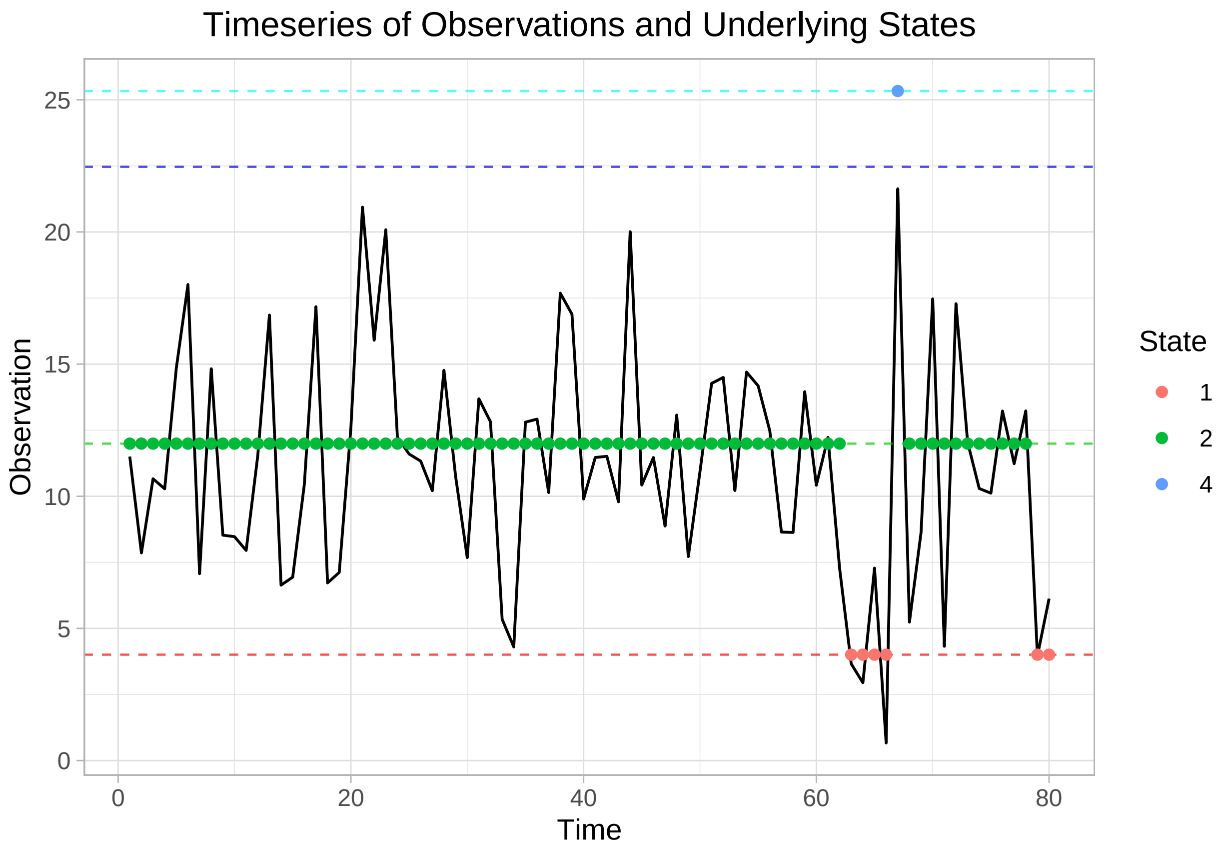


Figure 9: Timeseries of first 80 observations. The dashed lines are the fitted means (alpha*theta) for each state dependent gamma distribution. The colors of the points correspond to the fitted state of the observation at that time.

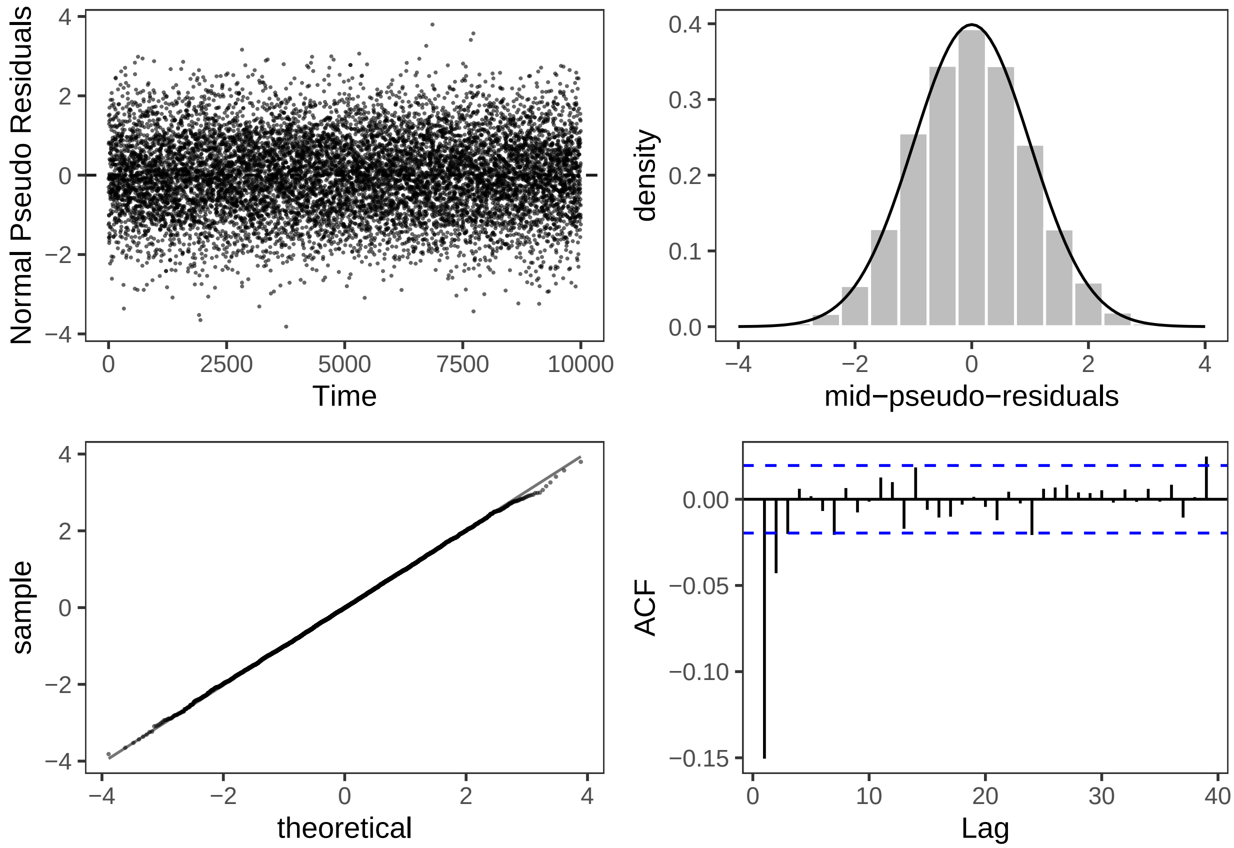


Figure 10: Plots for the normal pseudo-residuals of the fitted 4-state gamma HMM. TOP LEFT: Index plot of the normal pseudo residuals. TOP RIGHT: Histogram of the normal pseudo-residuals with N(0,1) overlayed. BOTTOM LEFT: Quantile–quantile plot of the normal pseudo-residuals. BOTTOM RIGHT: Autocorrelation function of the normal pseudo-residuals.

Fit the Generated Data with a 3-State Normal HMM

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

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.04427  5.30798 10.70416 12.08399 15.83421 63.20622
```

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 2. 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]  3.634154 11.806472 23.509934
```

```
##
## $sigma
## [1] 2.285003 3.383684 8.934558
##
## $gamma
##            [,1]      [,2]       [,3]
## [1,] 0.87607288 0.0521821 0.07174502
## [2,] 0.06099396 0.8763842 0.06262183
## [3,] 0.04287749 0.1693811 0.78774144
##
## $delta
## [1] 0.3065224 0.4554894 0.2379882
##
## $code
## [1] 1
##
## $mllk
## [1] 31170.27
##
## $AIC
## [1] 62364.55
##
## $BIC
## [1] 62451.07
```

Confidence intervals

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

```
## $mu
## [1]  3.634154 11.806472 23.509934
##
## $mu.SE
## [1] 0.04149520 0.05745413 0.20302509
##
## $sigma
## [1] 2.285003 3.383684 8.934558
##
## $sigma.SE
## [1] 0.03166268 0.04919728 0.14511870
##
## $gamma
##            [,1]      [,2]       [,3]
## [1,] 0.87607288 0.0521821 0.07174502
## [2,] 0.06099396 0.8763842 0.06262183
## [3,] 0.04287749 0.1693811 0.78774144
##
## $gamma.SE
##             [,1]        [,2]        [,3]
## [1,] 0.005170810 0.006852390 0.005211138
## [2,] 0.003824455 0.005160703 0.004640016
## [3,] 0.004234688 0.015717061 0.014668106
##
```

```
## $delta
## [1] 0.3065224 0.4554894 0.2379882
##
## $delta.SE
## [1] 0.01261316 0.01251845 0.01330563
```

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

Gloabl decoding of states

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

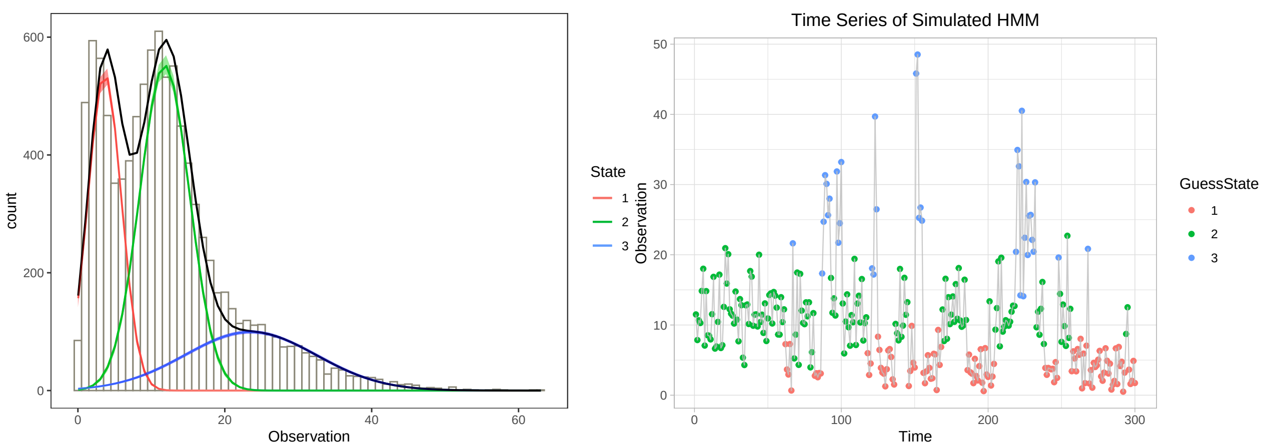


Figure 11: LEFT: Histogram of observations with fitted state dependent gamma distributions overlayed. Black line is the marginal distribution. RIGHT: Timeseries of first 300 observations with colors corresponding fitted hidden states decoded by Viterbi algorithm.

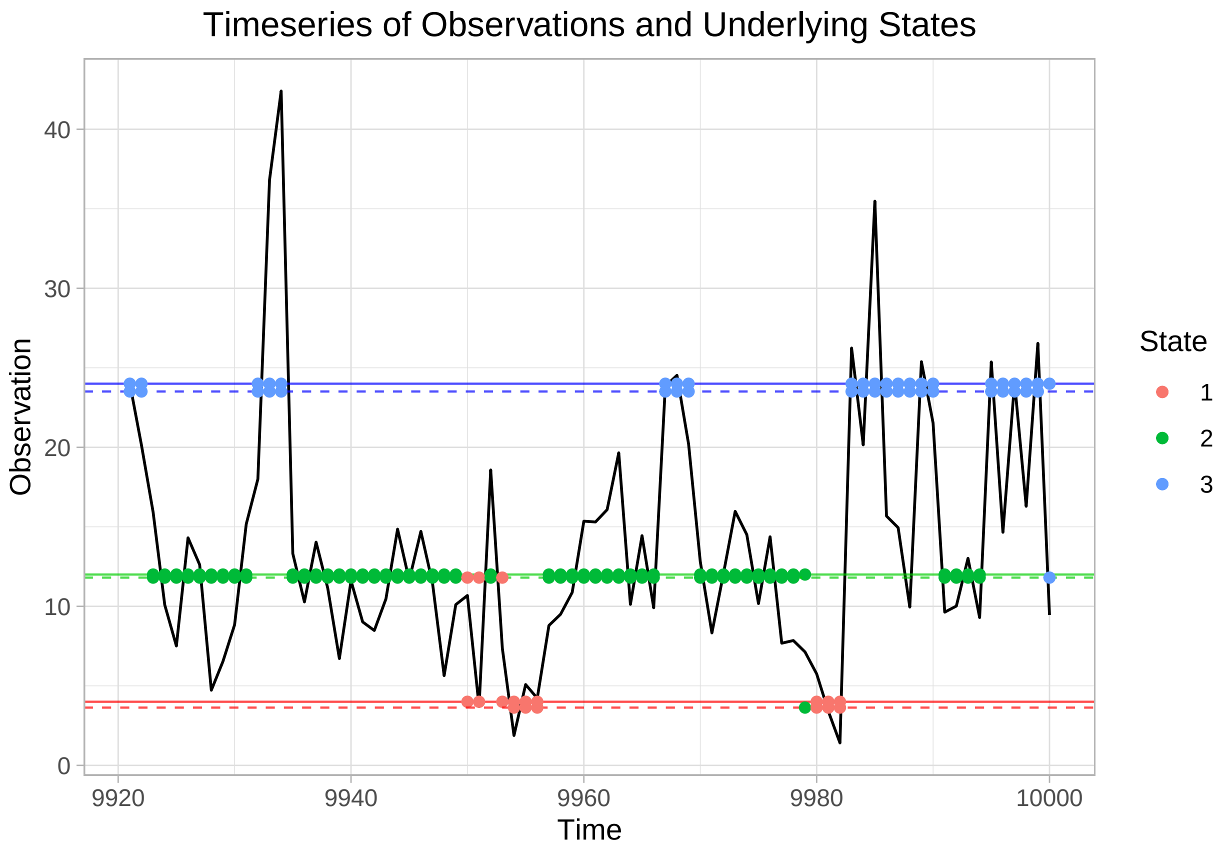


Figure 12: Timeseries of first 80 observations. The solid horizontal lines are the true means (alpha*theta) of each state dependent gamma distribution and the dashed lines are the fitted means for the normal distributions (mu). For each time interval there are two points, one falling on the true state 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.

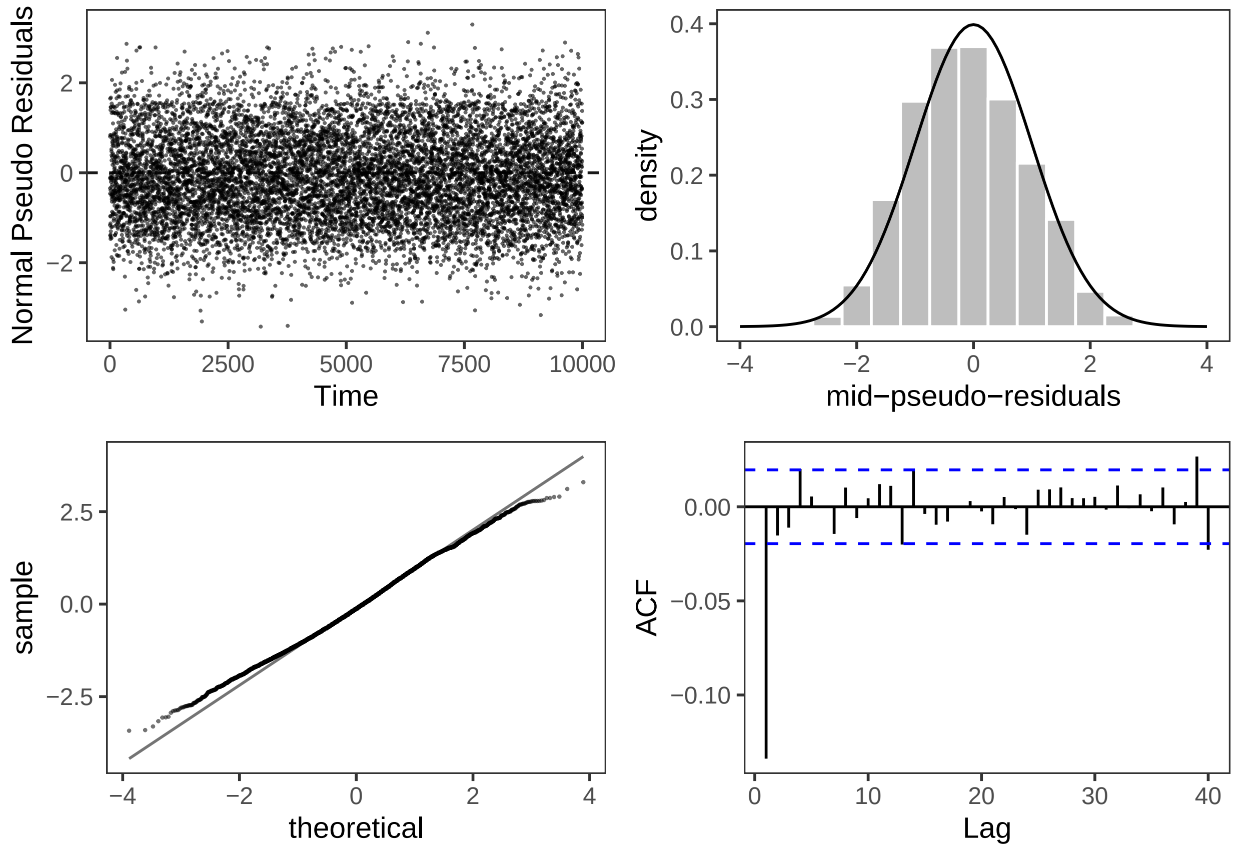


Figure 13: Plots for the normal pseudo-residuals of the fitted 3-state normal HMM. TOP LEFT: Index plot of the normal pseudo residuals. TOP RIGHT: Histogram of the normal pseudo-residuals with N(0,1) overlayed. BOTTOM LEFT: Quantile–quantile plot of the normal pseudo-residuals. BOTTOM RIGHT: Autocorrelation function of the normal pseudo-residuals.

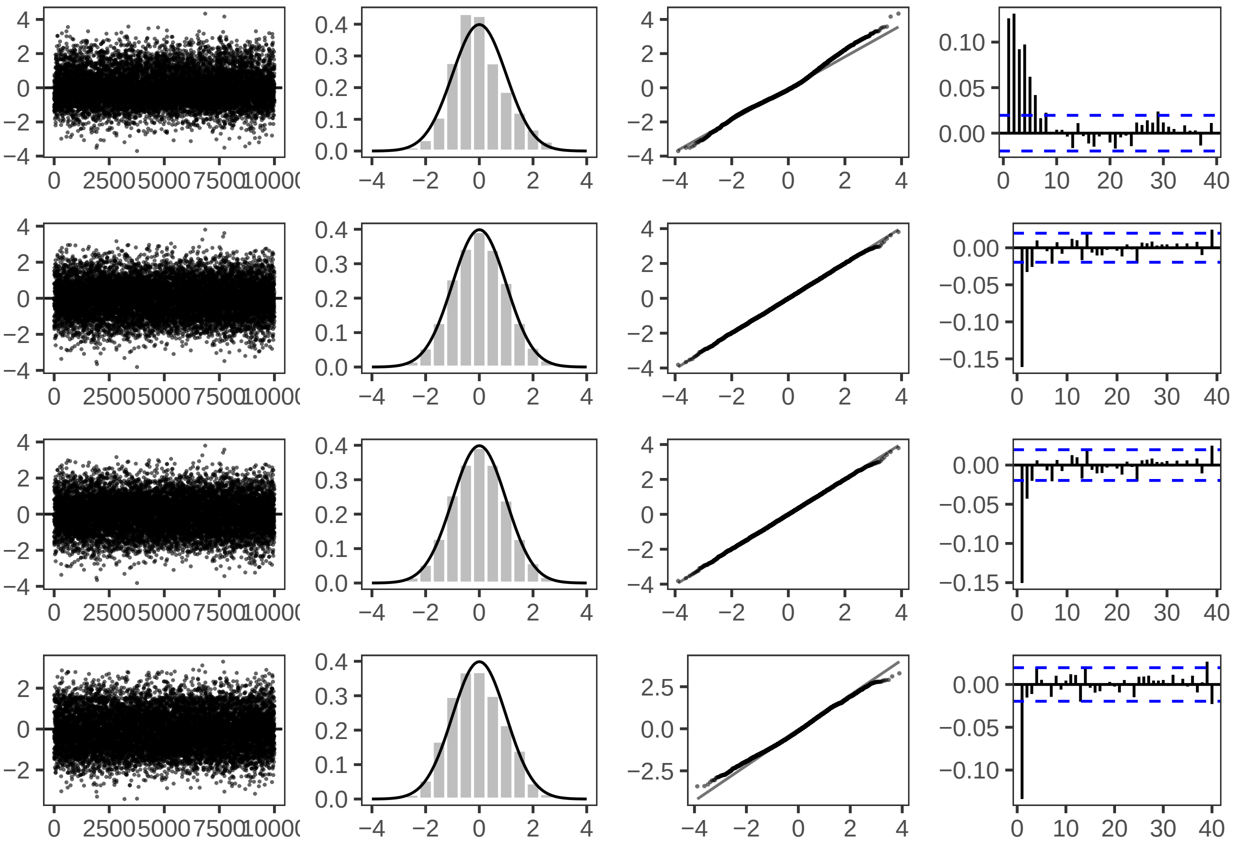


Figure 14: Each row shows plots of the pseudo residuals for different fitted HMMs. The first row is for a 2-state HMM with gamma state dependent distributions. The second is a 3-State gamma distributed HMM and the third is a 4-state gamma distributed HMM. The last row has a 3-state HMM with normal state dependent distributions. 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