

HMMexploration

2023-09-27

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

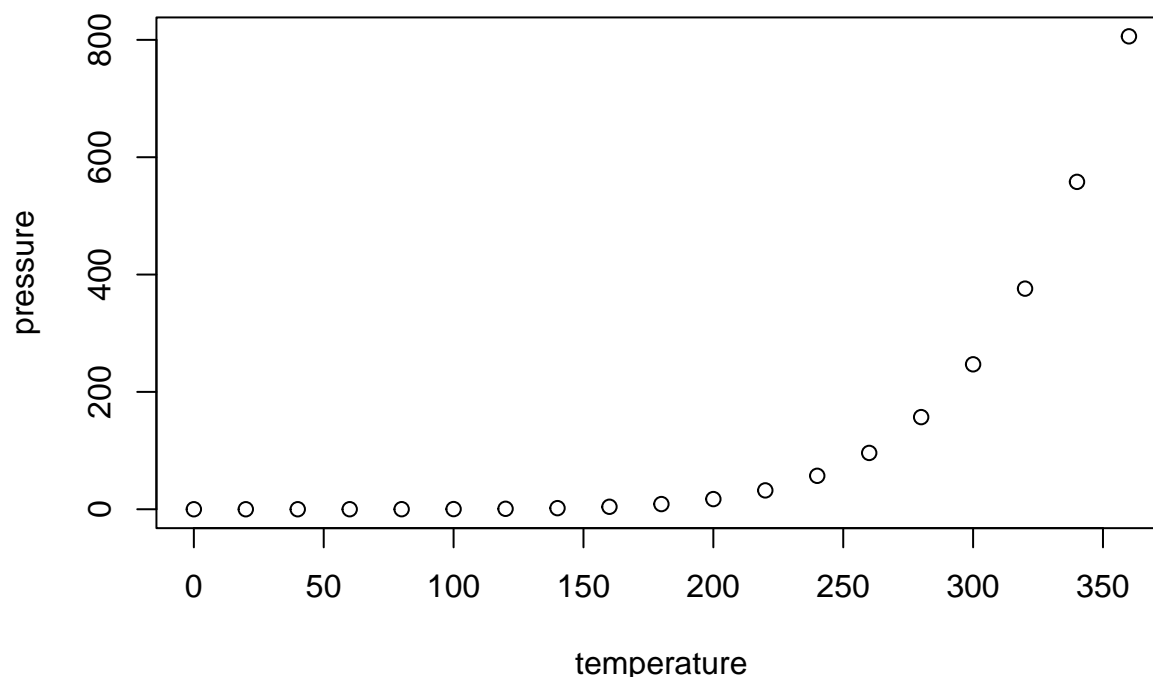
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   :  2.00
## 1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##  Mean   :15.4    Mean   : 42.98
## 3rd Qu.:19.0    3rd Qu.: 56.00
##  Max.   :25.0    Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Hidden Markov Model exploration for calves

loading and visualizing data

load the libraries and the data from Matlab (it will already be chunked and averaged over a specified window)

```
library(fitdistrplus)
```

```
## Warning: package 'fitdistrplus' was built under R version 4.3.1
```

```
## Loading required package: MASS
```

```
## Loading required package: survival
```

```
library(momentuHMM)
```

```
## Warning: package 'momentuHMM' was built under R version 4.3.1
```

```
## The legacy packages mapproj, rgdal, and rgeos, underpinning the sp package,
## which was just loaded, will retire in October 2023.
```

```
## Please refer to R-spatial evolution reports for details, especially
## https://r-spatial.org/r/2023/05/15/evolution4.html.
```

```
## It may be desirable to make the sf package available;
## package maintainers should consider adding sf to Suggests:.
```

```
## The sp package is now running under evolution status 2
```

```
## (status 2 uses the sf package in place of rgdal)
```

```
## momentuHMM 1.5.5 (2022-10-18)
```

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.3.1
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.1
```

```
all_other = read.csv('all_other.csv')
```

```
all_rest = read.csv('all_rest.csv')
```

```
all_travel = read.csv('all_travel.csv')
```

```
# rename the columns
```

```
col_names = c('date_no', 'avg_speed', 'mean_resultant_length',  
              'max_depth', 'msa')
```

```
names(all_other) = col_names
```

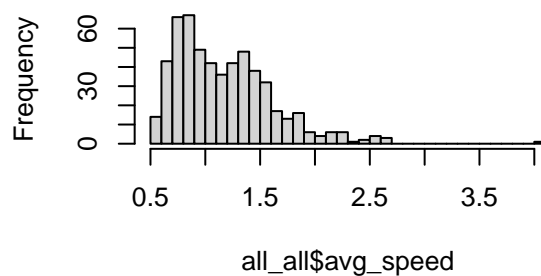
```
names(all_rest) = col_names
```

```
names(all_travel) = col_names
```

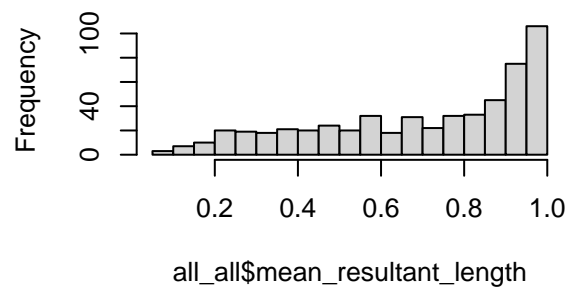
```
all_all = rbind(all_rest, all_travel, all_other)
```

Look at the distributions of all 4 data streams

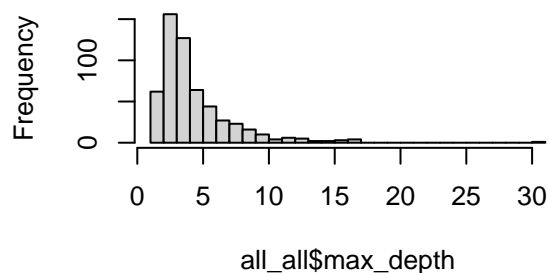
Histogram of all_all\$avg_speed



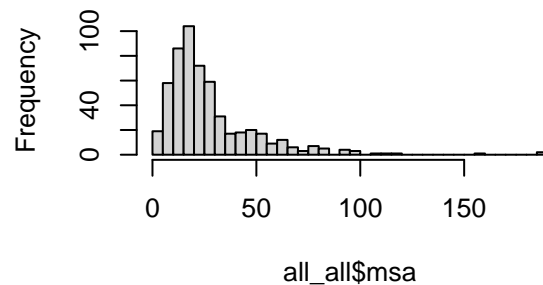
Histogram of all_all\$mean_resultant_len



Histogram of all_all\$max_depth



Histogram of all_all\$msa



parameter estimation for each variable

SPEED

```

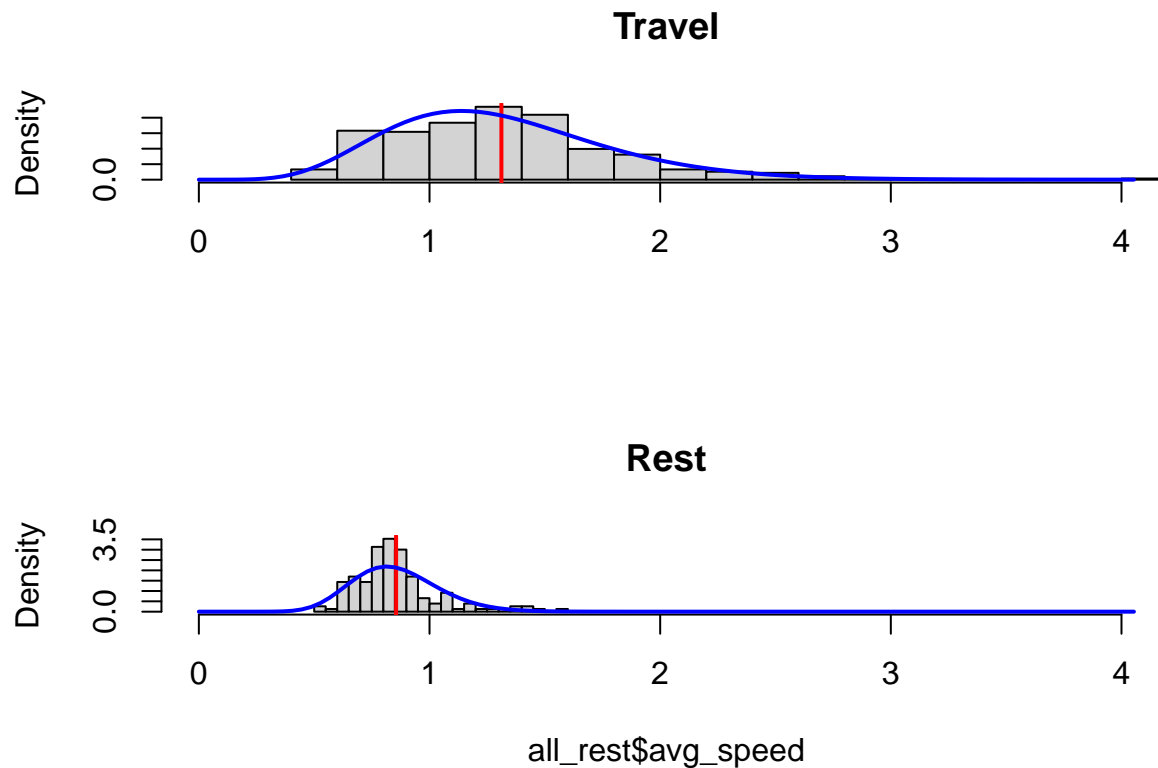
moment_match_gamma <- function(mu, sigma){
  # sigma is the standard deviation
  alpha = (mu ^ 2) / (sigma ^ 2)
  beta = mu / (sigma ^ 2)
  parameters = c(alpha, beta)
  names(parameters) = c('alpha', 'beta')
  parameters
}

# calculate the parameters for the travel state
travel_params = moment_match_gamma(
  mean(all_travel$avg_speed),
  sd(all_travel$avg_speed)
)

#and for the rest state
rest_params = moment_match_gamma(
  mean(all_rest$avg_speed),
  sd(all_rest$avg_speed)
)

```

plot the traveling and resting data and distributions



MEAN RESULTANT LENGTH

```

moment_match_beta <- function(mu, var){
  # var is the variance
  alpha = ((1 - mu) / var - 1 / mu) * mu ^ 2
  beta = alpha * (1 / mu - 1)
  parameters = c(alpha, beta)
  names(parameters) = c('alpha', 'beta')
  parameters
}

travel_mrl_params = moment_match_beta(
  mean(all_travel$mean_resultant_length),
  var(all_travel$mean_resultant_length)
)

rest_mrl_params = moment_match_beta(
  mean(all_rest$mean_resultant_length),
  var(all_rest$mean_resultant_length)
)

```

plot the traveling and resting data and distributions

```

par(mfrow = c(2, 1))
hist(all_travel$mean_resultant_length, probability = TRUE, breaks=20, main = 'Travel',
      xlim=c(0, max(all_travel$mean_resultant_length)), xlab='')

# vertical line for mean
abline(v=mean(all_travel$mean_resultant_length), col='red', lwd=2)

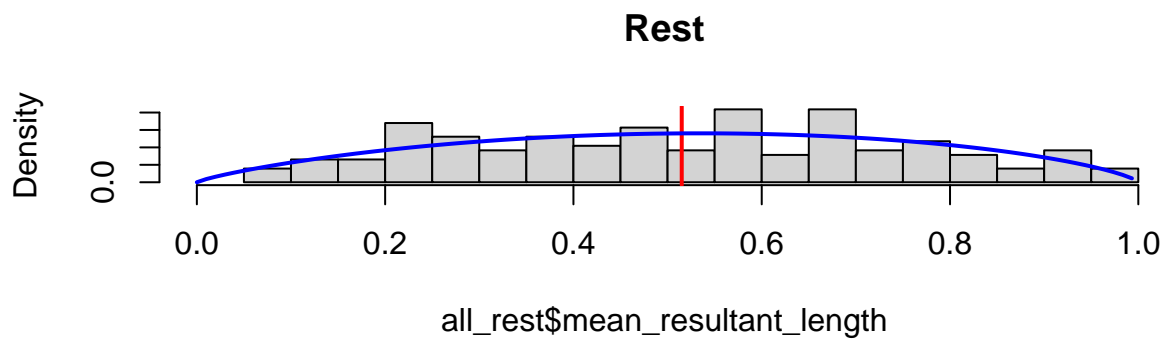
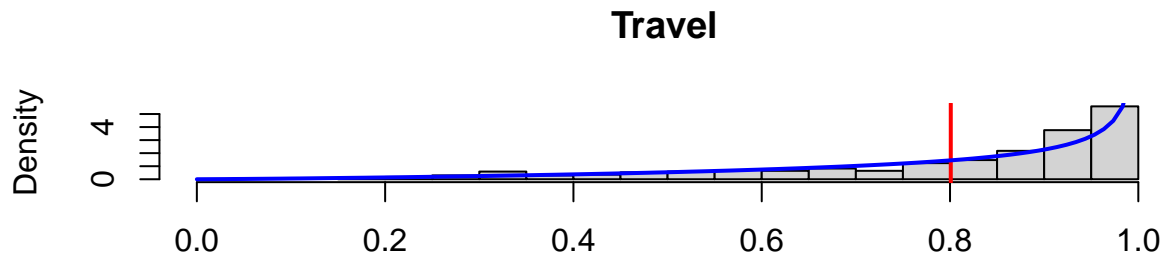
# curve for estimated beta
curve(dbeta(x, travel_mrl_params['alpha'], travel_mrl_params['beta']),
      add = T, col = 'blue', lwd = 2)

# ditto as above but for rest
hist(all_rest$mean_resultant_length, probability = TRUE, breaks=20, main = 'Rest',
      xlim=c(0, max(all_travel$mean_resultant_length)))

abline(v=mean(all_rest$mean_resultant_length), col='red', lwd=2)

curve(dbeta(x, rest_mrl_params['alpha'], rest_mrl_params['beta']),
      add = T, col = 'blue', lwd = 2)

```



DEPTH

```
travel_depth_params = moment_match_gamma(
  mean(all_travel$max_depth),
  sd(all_travel$max_depth)
)

rest_depth_params = moment_match_gamma(
  mean(all_rest$max_depth),
  sd(all_rest$max_depth)
)
```

plot the traveling and resting data and distributions

```
par(mfrow = c(2, 1))
hist(all_travel$max_depth, probability = TRUE, breaks=20, main = 'Travel',
     xlim=c(0, max(all_travel$max_depth)), xlab='')

# vertical line for mean
abline(v=mean(all_travel$max_depth), col='red', lwd=2)

# curve for estimated gamma
curve(dgamma(x, travel_depth_params['alpha'], travel_depth_params['beta']),
      add = T, col = 'blue', lwd = 2)

# ditto as above but for rest
hist(all_rest$max_depth, probability = TRUE, breaks=20, main = 'Rest',
```

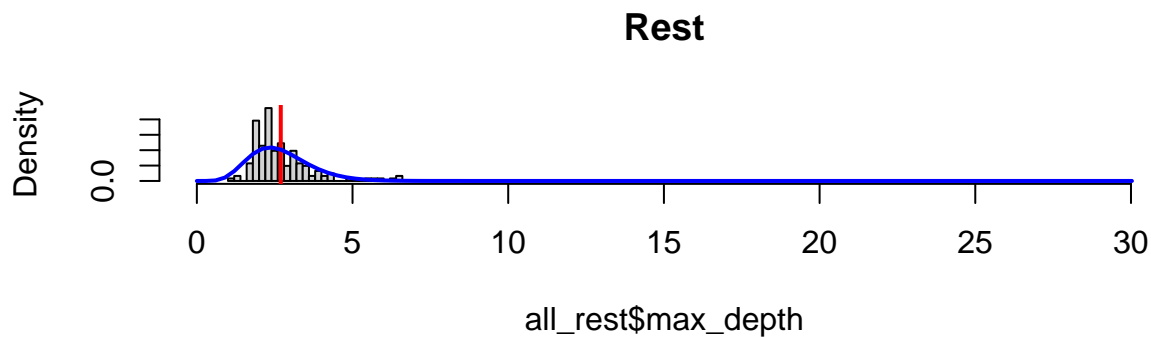
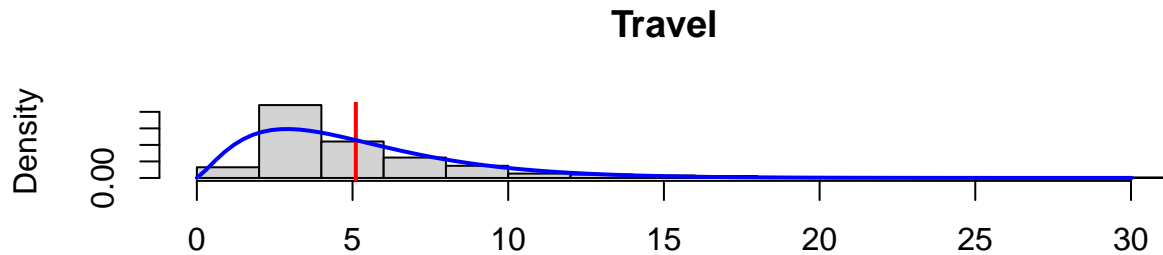
```

xlim=c(0, max(all_travel$max_depth)))

abline(v=mean(all_rest$max_depth), col='red', lwd=2)

curve(dgamma(x, rest_depth_params['alpha'], rest_depth_params['beta']),
      add = T, col = 'blue', lwd = 2)

```



AVERAGE MINIMUM SPECIFIC ACCELERATION

```

travel_msa_params = moment_match_gamma(
  mean(all_travel$msa),
  sd(all_travel$msa)
)

rest_msa_params = moment_match_gamma(
  mean(all_rest$msa),
  sd(all_rest$msa)
)

```

plot the traveling and resting data and distributions

```

par(mfrow = c(2, 1))
hist(all_travel$msa, probability = TRUE, breaks=20, main = 'Travel',
      xlim=c(0, max(all_travel$msa)), xlab='')

# vertical line for mean
abline(v=mean(all_travel$msa), col='red', lwd=2)

```

```

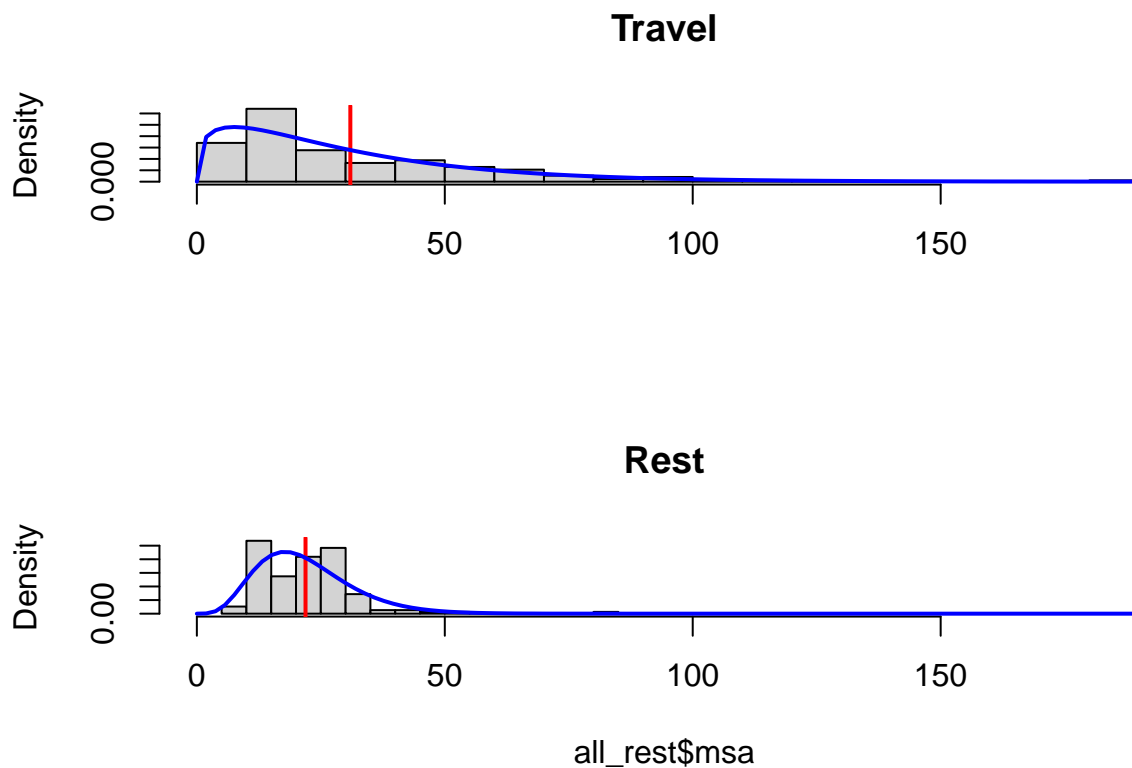
# curve for estimated gamma
curve(dgamma(x, travel_msa_params['alpha'], travel_msa_params['beta']),
      add = T, col = 'blue', lwd = 2)

# ditto as above but for rest
hist(all_rest$msa, probability = TRUE, breaks=20, main = 'Rest',
      xlim=c(0, max(all_travel$msa)))

abline(v=mean(all_rest$msa), col='red', lwd=2)

curve(dgamma(x, rest_msa_params['alpha'], rest_msa_params['beta']),
      add = T, col = 'blue', lwd = 2)

```



simulate data and run simple HMMs

Simulate speed

```

# simulate rest data from parameters
n_rest = nrow(all_rest)
sim_rest = rgamma(n_rest, rest_params['alpha'], rest_params['beta'])

# simulate travel data from parameters
n_travel = nrow(all_travel)
sim_travel = rgamma(n_travel, travel_params['alpha'], travel_params['beta'])

xlim = c(0, max(all_travel$avg_speed))

```

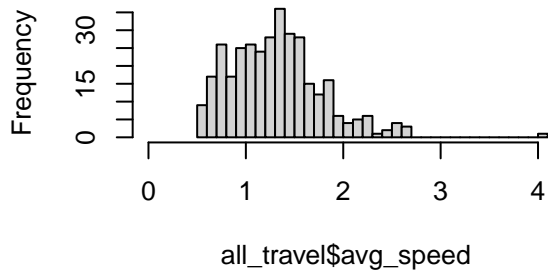


```

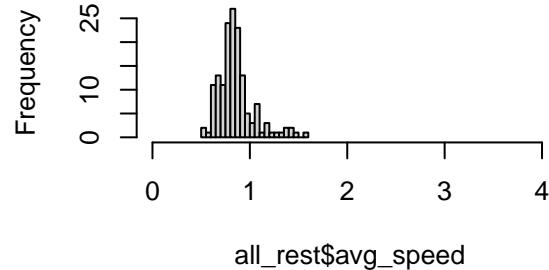
par(mfrow=c(2, 2))
hist(all_travel$avg_speed, breaks=30, xlim=xlim)
hist(all_rest$avg_speed, breaks=30, xlim=xlim)
hist(sim_travel, breaks=30, xlim=xlim)
hist(sim_rest, breaks=30, xlim=xlim)

```

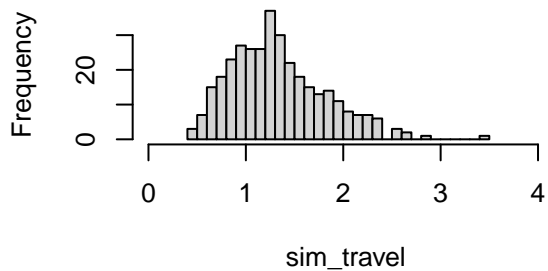
Histogram of all_travel\$avg_speed



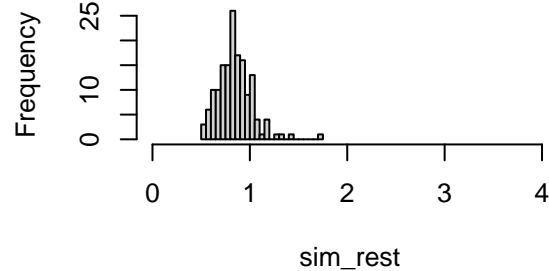
Histogram of all_rest\$avg_speed



Histogram of sim_travel



Histogram of sim_rest



compare simulated speed HMM output to real speed data HMM output

```

#try the HMM on the simulated speed data
sim_speed<- c(sim_travel, sim_rest)
combined_sim_speeds <- data.frame(avg_speed = sim_speed)

sim_speed_data<- prepData(combined_sim_speeds, type='LL', coordNames=NULL)
dist<-list(avg_speed="gamma")
Par0 <- list(avg_speed= c(rest_params[['alpha']],travel_params[['alpha']], rest_params[['beta']], travel_params[['beta']]))
stateNames <- c("resting","traveling")

sim_test1_speed <- fitHMM(sim_speed_data, nbStates=2, dist=dist, Par0=Par0, stateNames=stateNames)

## =====
## Fitting HMM with 2 states and 1 data stream
## -----
## avg_speed ~ gamma(mean=~1, sd=~1)
##

```

```

## Transition probability matrix formula: ~1
##
## Initial distribution formula: ~1
## =====
## DONE
print(sim_test1_speed)

## Value of the maximum log-likelihood: -278.7787
##
##
## avg_speed parameters:
## -----
##           resting traveling
## mean 2.779147e-06 1.181837
## sd   5.082706e+00 0.447573
##
## Regression coeffs for the transition probabilities:
## -----
##           1 -> 2    2 -> 1
## (Intercept) 16.43828 -172.753
##
## Transition probability matrix:
## -----
##           resting traveling
## resting    7.260180e-08 0.9999999
## traveling  9.426126e-76 1.0000000
##
## Initial distribution:
## -----
##   resting traveling
## 0.1455826 0.8544174
#plot(sim_test1_speed, plotCI = TRUE, breaks=30)

```

The model did not work on the simulated data...

real speed data HMM

```

all_speed<- c(all_travel$avg_speed, all_rest$avg_speed)
combined_avg_speeds <- data.frame(avg_speed = all_speed)

speed_data<- prepData(combined_avg_speeds, type='LL', coordNames=NULL)
dist<-list(avg_speed="gamma")
Par0 <- list(avg_speed= c(rest_params['alpha'],travel_params['alpha'] , rest_params['beta'], travel_params['beta']))
stateNames <- c("traveling","resting")

test1_speed <- fitHMM(speed_data, nbStates=2, dist=dist, Par0=Par0, stateNames=stateNames)

## =====
## Fitting HMM with 2 states and 1 data stream
## -----
##   avg_speed ~ gamma(mean=~1, sd=~1)

```

```

##
## Transition probability matrix formula: ~1
##
## Initial distribution formula: ~1
## =====
## DONE
print(test1_speed)

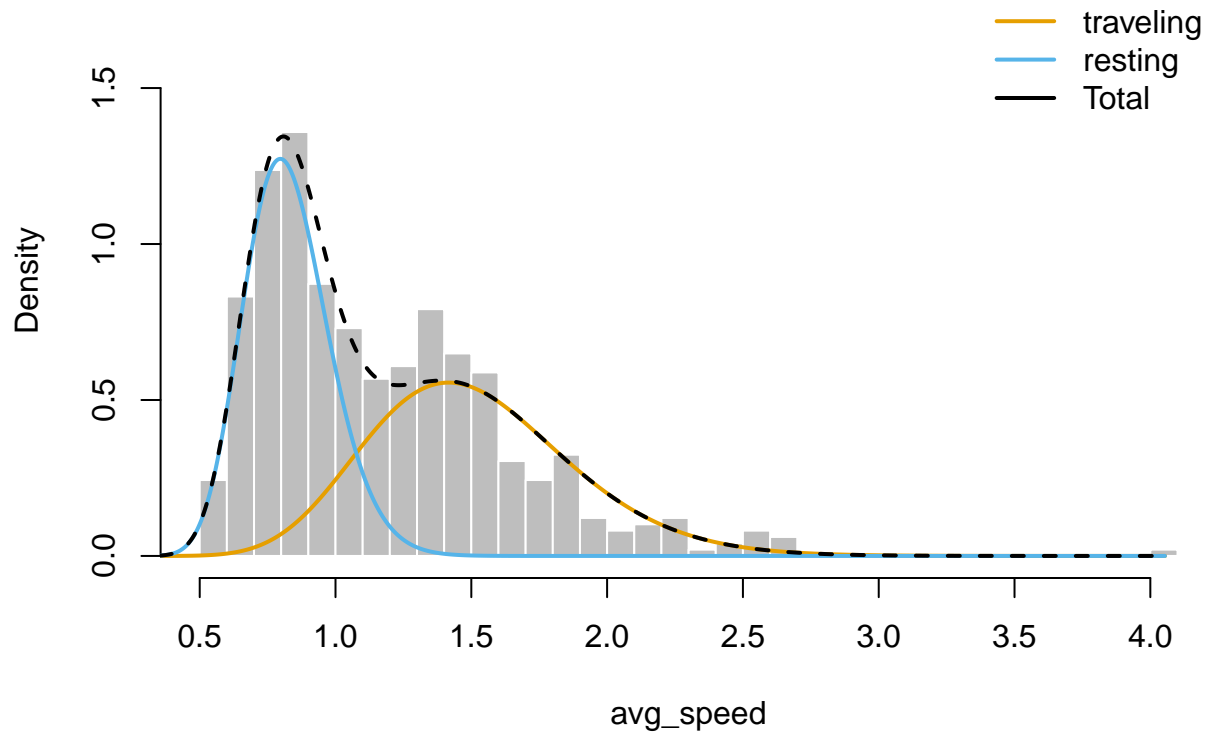
## Value of the maximum log-likelihood: -74.09784
##
##
## avg_speed parameters:
## -----
##      traveling    resting
## mean 1.5096063 0.8263913
## sd   0.3752685 0.1561869
##
## Regression coeffs for the transition probabilities:
## -----
##              1 -> 2    2 -> 1
## (Intercept) -2.790865 -2.777805
##
## Transition probability matrix:
## -----
##              traveling    resting
## traveling 0.94218017 0.05781983
## resting   0.05853542 0.94146458
##
## Initial distribution:
## -----
##      traveling    resting
## 2.568529e-09 1.000000e+00
states <- viterbi(test1_speed)
table(states)/nrow(speed_data) #well that didn't really work

## states
##      1      2
## 0.5091278 0.4908722
plot(test1_speed, plotCI = TRUE, breaks=30)

## Decoding state sequence... DONE

```

ID Animal1



Simulate MRL

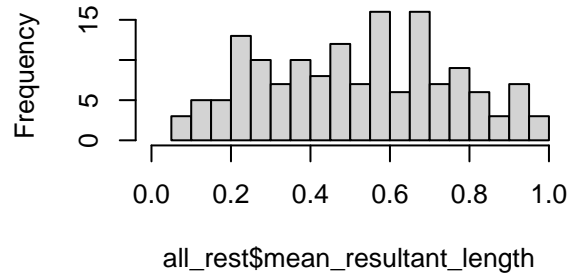
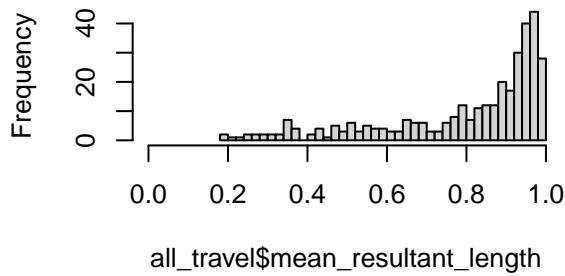
```
# simulate rest data from parameters
n_rest = nrow(all_rest)
sim_mrl_rest = rbeta(n_rest, rest_mrl_params['alpha'], rest_mrl_params['beta'])

# simulate travel data from parameters
n_travel = nrow(all_travel)
sim_mrl_travel = rbeta(n_travel, travel_mrl_params['alpha'], travel_mrl_params['beta'])

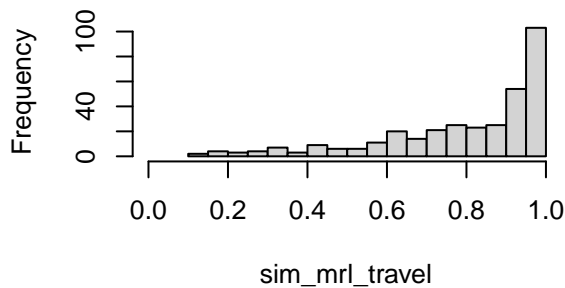
xlim = c(0, max(all_travel$mean_resultant_length))

par(mfrow=c(2, 2))
hist(all_travel$mean_resultant_length, breaks=30, xlim=xlim)
hist(all_rest$mean_resultant_length, breaks=30, xlim=xlim)
hist(sim_mrl_travel, breaks=30, xlim=xlim)
hist(sim_mrl_rest, breaks=30, xlim=xlim)
```

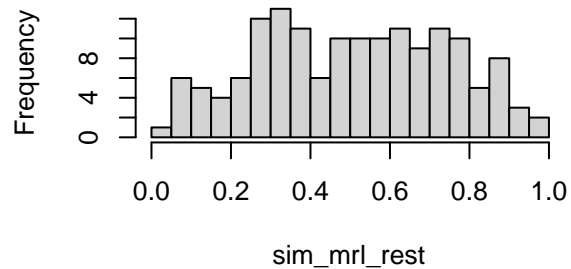
histogram of all_travel\$mean_resultant_leHistogram of all_rest\$mean_resultant_lei



Histogram of sim_mrl_travel



Histogram of sim_mrl_rest



Compare simulated MRL results to real MRL data results

```
sim_mrl<- c(sim_mrl_travel, sim_mrl_rest)
combined_sim_mrl <- data.frame(mrl = sim_mrl)

sim_mrl_data<- prepData(combined_sim_mrl, type='LL', coordNames=NULL)
dist<-list(mrl="beta")
Par0 <- list(mrl= c(rest_mrl_params['alpha'],travel_mrl_params['alpha'] , rest_mrl_params['beta'], travel_mrl_params['beta']),
stateNames <- c("resting","traveling")

sim_test2_mrl <- fitHMM(sim_mrl_data, nbStates=2, dist=dist, Par0=Par0, stateNames=stateNames)

## =====
## Fitting HMM with 2 states and 1 data stream
## -----
## mrl ~ beta(shape1=~1, shape2=~1)
##
## Transition probability matrix formula: ~1
##
## Initial distribution formula: ~1
## =====
## DONE
```

```

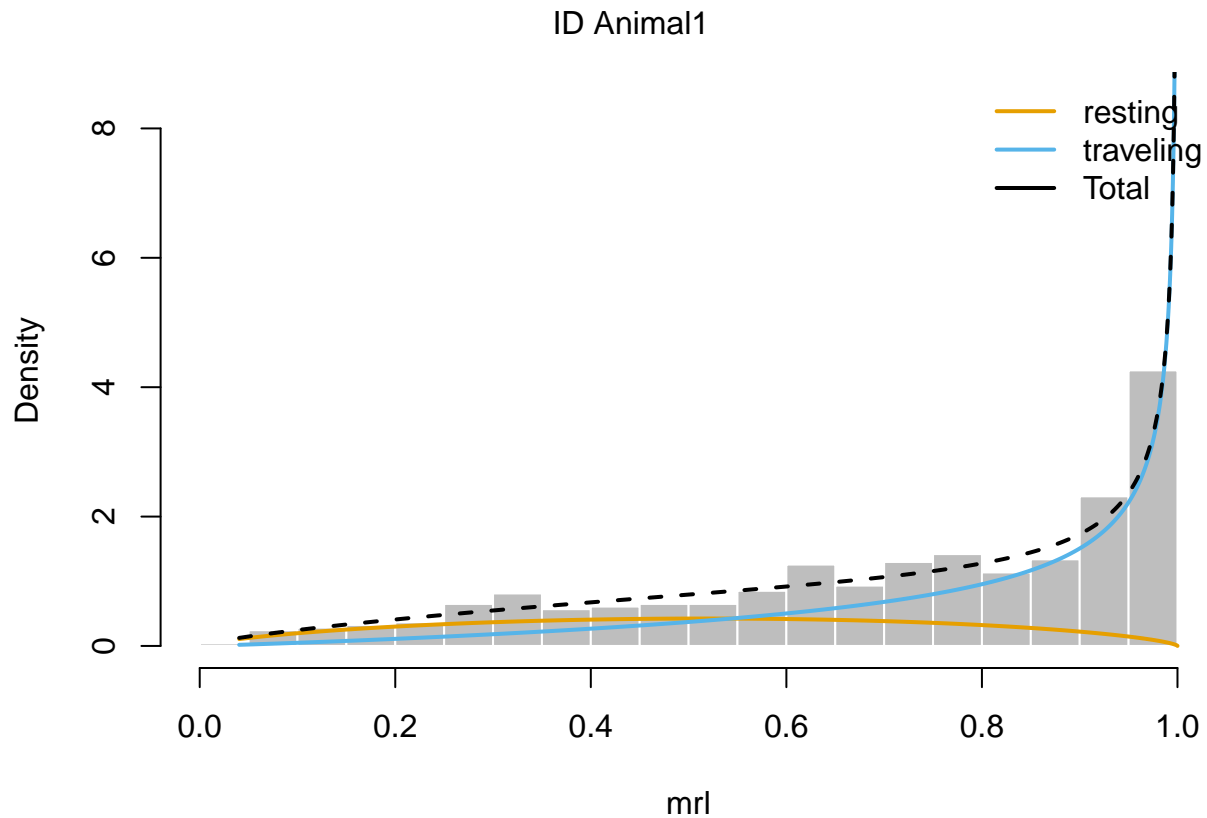
print(sim_test2_mrl)

## Value of the maximum log-likelihood: 255.7218
##
##
## mrl parameters:
## -----
##           resting traveling
## shape1 1.719708 2.0859862
## shape2 1.665464 0.5254493
##
## Regression coeffs for the transition probabilities:
## -----
##           1 -> 2    2 -> 1
## (Intercept) -109.5512 -5.830175
##
## Transition probability matrix:
## -----
##           resting    traveling
## resting    1.000000000 2.645509e-48
## traveling  0.002928958 9.970710e-01
##
## Initial distribution:
## -----
##           resting    traveling
## 1.679740e-07 9.999998e-01

plot(sim_test2_mrl, plotCI = TRUE, breaks=30)

## Decoding state sequence... DONE

```



real mrl data HMM

```
all_mrl<- c(all_travel$mean_resultant_length, all_rest$mean_resultant_length)
combined_mrl <- data.frame(mrl = all_mrl)

mrl_data<- prepData(combined_mrl, type='LL', coordNames=NULL)
dist<-list(mrl="beta")
Par0 <- list(mrl= c(rest_mrl_params['alpha'],travel_mrl_params['alpha'] , rest_mrl_params['beta'], travel_mrl_params['beta']),
stateNames <- c("resting","traveling")

test2_mrl <- fithMM(mrl_data, nbStates=2, dist=dist, Par0=Par0, stateNames=stateNames)

## =====
## Fitting HMM with 2 states and 1 data stream
## -----
## mrl ~ beta(shape1=~1, shape2=~1)
##
## Transition probability matrix formula: ~1
##
## Initial distribution formula: ~1
## =====
## DONE
```

```

print(test2_mrl)

## Value of the maximum log-likelihood: 266.8004
##
##
## mrl parameters:
## -----
##           resting traveling
## shape1 2.260958 24.079289
## shape2 1.690591 1.706516
##
## Regression coeffs for the transition probabilities:
## -----
##           1 -> 2    2 -> 1
## (Intercept) -2.350092 -1.838156
##
## Transition probability matrix:
## -----
##           resting traveling
## resting    0.9129415 0.08705849
## traveling 0.1372695 0.86273049
##
## Initial distribution:
## -----
##           resting traveling
## 9.999999e-01 6.708947e-08

states <- viterbi(test2_mrl)
table(states)/nrow(mrl_data) #well that didn't really work

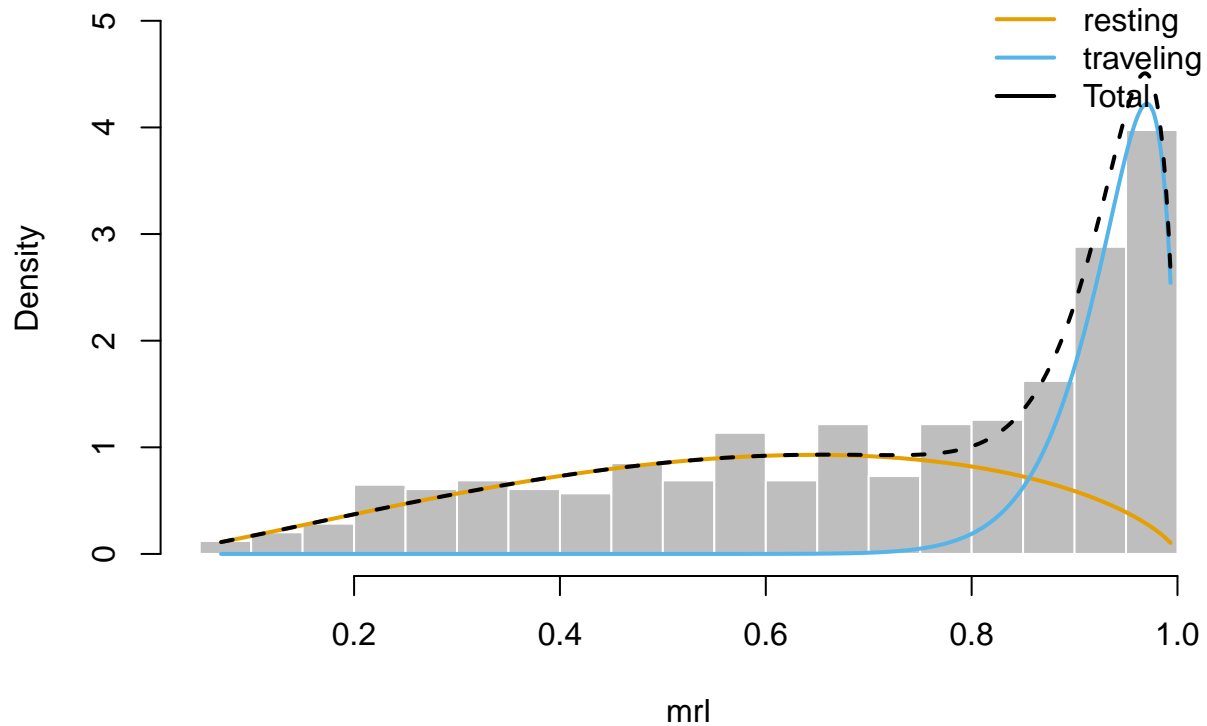
## states
##           1           2
## 0.6044625 0.3955375

plot(test2_mrl, plotCI = TRUE, breaks=30)

## Decoding state sequence... DONE

```


ID Animal1



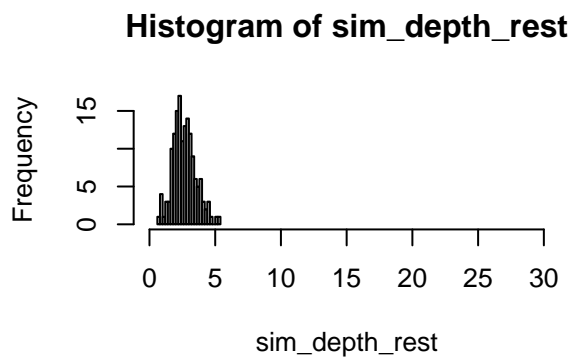
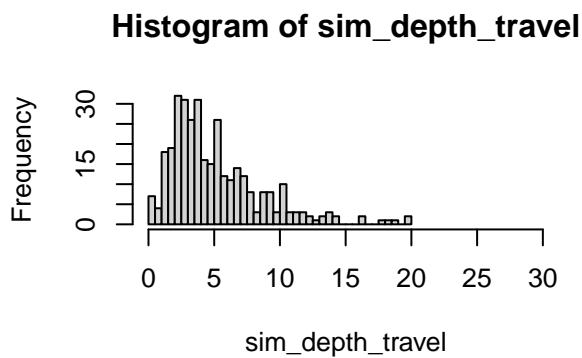
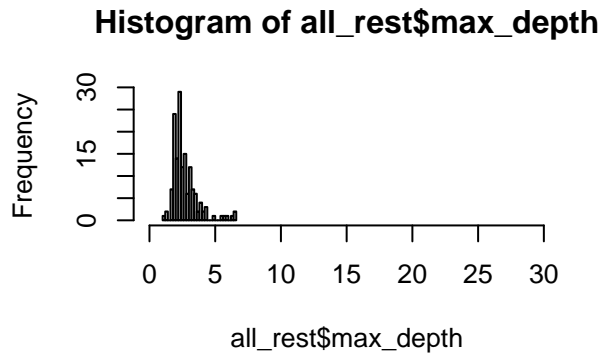
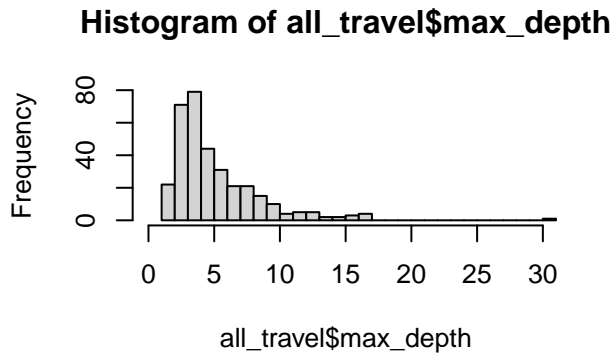
simulate max depth

```
# simulate rest data from parameters
n_rest = nrow(all_rest)
sim_depth_rest = rgamma(n_rest, rest_depth_params['alpha'], rest_depth_params['beta'])

# simulate travel data from parameters
n_travel = nrow(all_travel)
sim_depth_travel = rgamma(n_travel, travel_depth_params['alpha'], travel_depth_params['beta'])

xlim = c(0, max(all_travel$max_depth))

par(mfrow=c(2, 2))
hist(all_travel$max_depth, breaks=30, xlim=xlim)
hist(all_rest$max_depth, breaks=30, xlim=xlim)
hist(sim_depth_travel, breaks=30, xlim=xlim)
hist(sim_depth_rest, breaks=30, xlim=xlim)
```



Compare simulated max depth HMM to real max depth HMM

```
sim_depth<- c(sim_depth_travel, sim_depth_rest)
combined_sim_depth<- data.frame(max_depth = sim_depth)
sim_depth_data<- prepData(combined_sim_depth, type='LL', coordNames=NULL)
dist<-list(max_depth="gamma")
Par0 <- list(max_depth= c(rest_depth_params['alpha'],travel_depth_params['alpha'] , rest_depth_params['alpha']),
stateNames <- c("traveling","resting")

sim_test2_depth <- fitHMM(sim_depth_data, nbStates=2, dist=dist, Par0=Par0, stateNames=stateNames)

## =====
## Fitting HMM with 2 states and 1 data stream
## -----
## max_depth ~ gamma(mean=~1, sd=~1)
##
## Transition probability matrix formula: ~1
##
## Initial distribution formula: ~1
## =====
## DONE
print(sim_test2_depth)
```

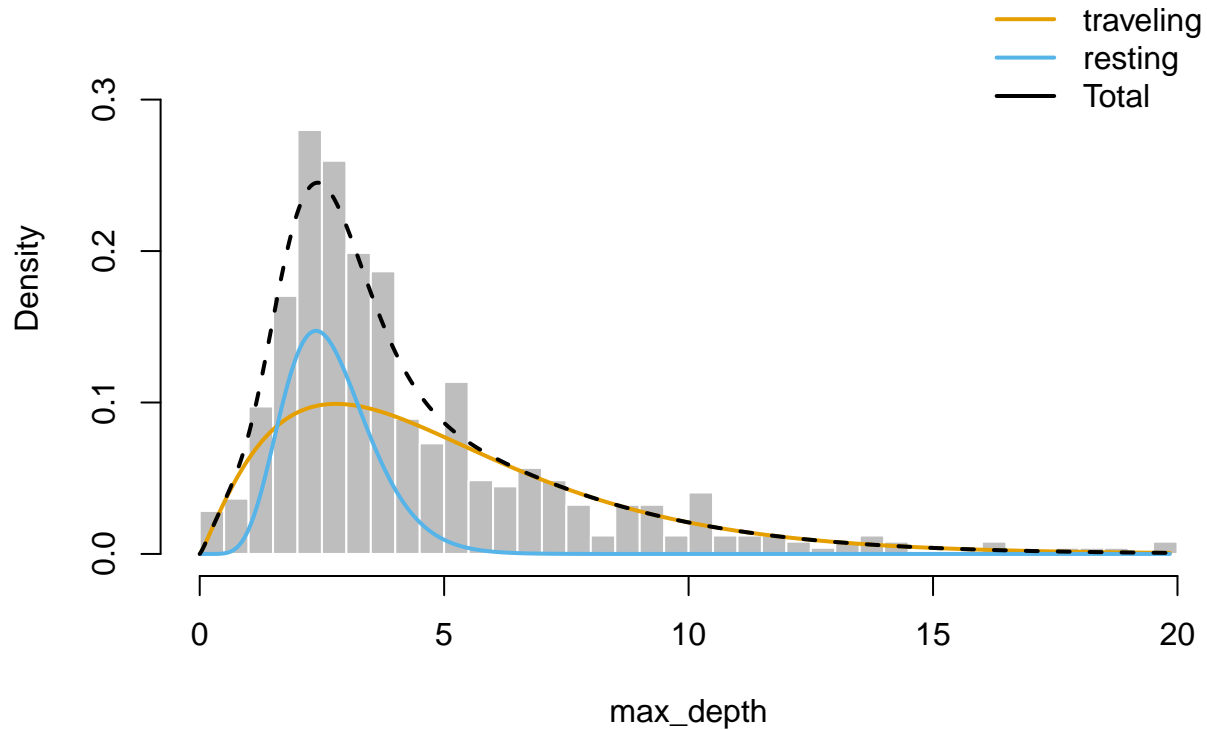
```

## Value of the maximum log-likelihood: -1043.227
##
##
## max_depth parameters:
## -----
##      traveling    resting
## mean  5.128630 2.6891625
## sd    3.465335 0.9127352
##
## Regression coeffs for the transition probabilities:
## -----
##              1 -> 2    2 -> 1
## (Intercept) -5.814667 -52.64033
##
## Transition probability matrix:
## -----
##              traveling    resting
## traveling 9.970254e-01 0.0029746
## resting   1.375932e-23 1.0000000
##
## Initial distribution:
## -----
##      traveling    resting
## 9.999987e-01 1.339326e-06
states <- viterbi(sim_test2_depth)
plot(sim_test2_depth, plotCI = TRUE, breaks=30)

## Decoding state sequence... DONE

```

ID Animal1



real max depth data HMM

```
all_depth<- c(all_travel$max_depth, all_rest$max_depth)
combined_depth <- data.frame(max_depth = all_depth)
depth_data<- prepData(combined_depth, type='LL', coordNames=NULL)
dist<-list(max_depth="gamma")
Par0 <- list(max_depth= c(rest_depth_params['alpha'],travel_depth_params['alpha'] , rest_depth_params['alpha'])
stateNames<- c("traveling","resting")

test2_depth <- fitHMM(depth_data, nbStates=2, dist=dist, Par0=Par0, stateNames=stateNames)
```

```
## =====
## Fitting HMM with 2 states and 1 data stream
## -----
## max_depth ~ gamma(mean=~1, sd=~1)
##
## Transition probability matrix formula: ~1
##
## Initial distribution formula: ~1
## =====
## DONE
print(test2_depth)
```

```
## Value of the maximum log-likelihood: -900.5824
```

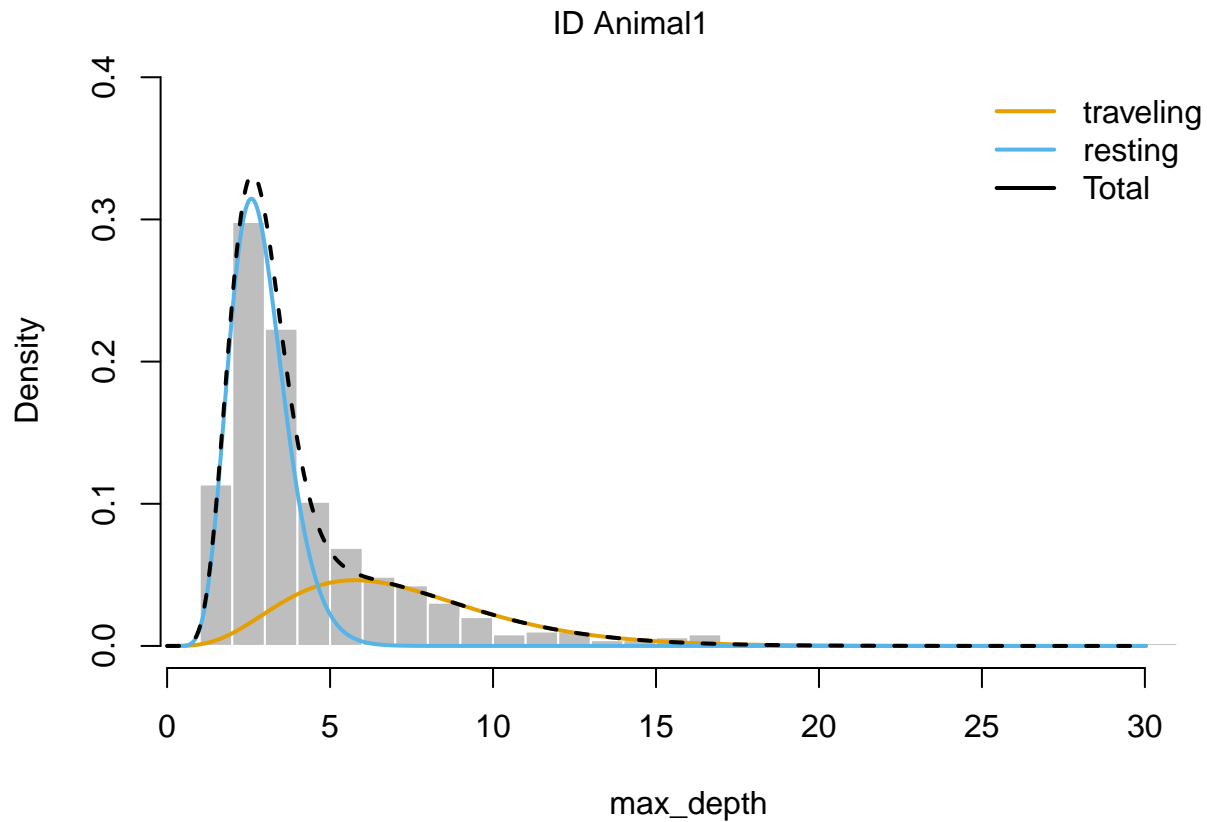
```

##
##
## max_depth parameters:
## -----
##      traveling   resting
## mean  7.183501 2.8538984
## sd    3.196963 0.8734001
##
## Regression coeffs for the transition probabilities:
## -----
##           1 -> 2    2 -> 1
## (Intercept) -2.523868 -3.266419
##
## Transition probability matrix:
## -----
##           traveling   resting
## traveling 0.92579818 0.07420182
## resting   0.03674137 0.96325863
##
## Initial distribution:
## -----
##           traveling   resting
## 9.999999e-01 7.088787e-08
states <- viterbi(test2_depth)
table(states)/nrow(depth_data) #well that didn't really work

## states
##           1           2
## 0.3387424 0.6612576
plot(test2_depth, plotCI = TRUE, breaks=30)

## Decoding state sequence... DONE

```



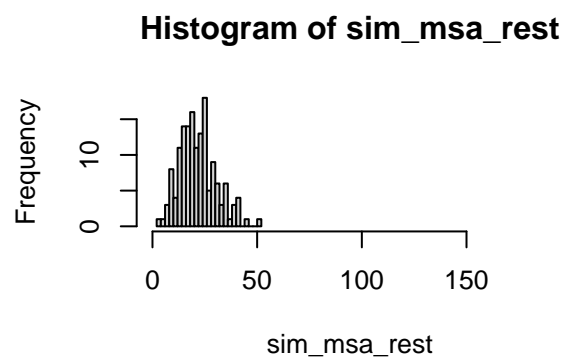
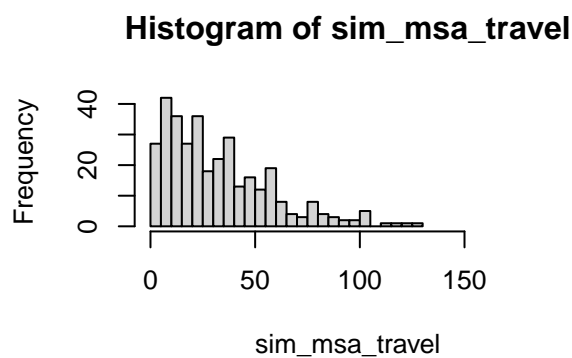
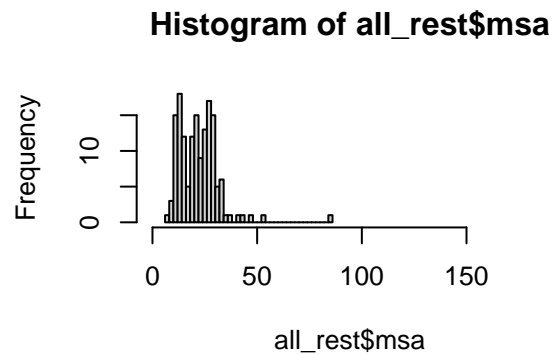
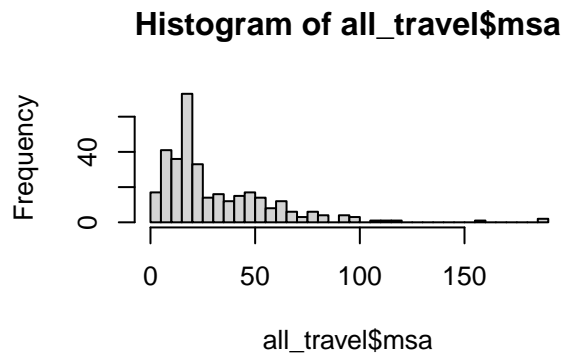
simulate average MSA

```
# simulate rest data from parameters
n_rest = nrow(all_rest)
sim_msa_rest = rgamma(n_rest, rest_msa_params['alpha'], rest_msa_params['beta'])

# simulate travel data from parameters
n_travel = nrow(all_travel)
sim_msa_travel = rgamma(n_travel, travel_msa_params['alpha'], travel_msa_params['beta'])

xlim = c(0, max(all_travel$msa))

par(mfrow=c(2, 2))
hist(all_travel$msa, breaks=30, xlim=xlim)
hist(all_rest$msa, breaks=30, xlim=xlim)
hist(sim_msa_travel, breaks=30, xlim=xlim)
hist(sim_msa_rest, breaks=30, xlim=xlim)
```



compare simulated HMMs to real HMMs

```
sim_msa<- c(sim_msa_travel, sim_msa_rest)
combined_sim_msa<- data.frame(msa = sim_msa)
sim_msa_data<- prepData(combined_sim_msa, type='LL', coordNames=NULL)
dist<-list(msa="gamma")
Par0 <- list(msa= c(rest_msa_params['alpha'],travel_msa_params['alpha'] , rest_msa_params['beta'], travel_msa_params['beta']),
stateNames <- c("traveling","resting")

sim_test2_msa<- fitHMM(sim_msa_data, nbStates=2, dist=dist, Par0=Par0, stateNames=stateNames)

## =====
## Fitting HMM with 2 states and 1 data stream
## -----
## msa ~ gamma(mean=~1, sd=~1)
##
## Transition probability matrix formula: ~1
##
## Initial distribution formula: ~1
## =====
## DONE
print(sim_test2_msa)
```

```

## Value of the maximum log-likelihood: -1.797693e+308
##
##
## msa parameters:
## -----
##      traveling      resting
## mean 5.2629057 1.31835179
## sd   0.2401494 0.04258705
##
## Regression coeffs for the transition probabilities:
## -----
##           1 -> 2 2 -> 1
## (Intercept)  -1.5   -1.5
##
## Transition probability matrix:
## -----
##           traveling      resting
## traveling 0.8175745 0.1824255
## resting   0.1824255 0.8175745
##
## Initial distribution:
## -----
## traveling      resting
##           0.5         0.5

```

```

all_msa<- c(all_travel$msa, all_rest$msa)
combined_msa <- data.frame(msa = all_msa)

msa_data<- prepData(combined_msa, type='LL', coordNames=NULL)
dist9<-list(msa="gamma")
Par09 <- list(msa= c(rest_msa_params['alpha'],travel_msa_params['alpha'] , rest_msa_params['beta'], tra
stateNames7 <- c("traveling","resting")

test2_msa <- fitHMM(msa_data, nbStates=2, dist=dist9, Par0=Par09, stateNames=stateNames7)

```

```

## =====
## Fitting HMM with 2 states and 1 data stream
## -----
## msa ~ gamma(mean=~1, sd=~1)
##
## Transition probability matrix formula: ~1
##
## Initial distribution formula: ~1
## =====
## DONE

```

```
print(test2_msa)
```

```

## Value of the maximum log-likelihood: -1.797693e+308
##
##
## msa parameters:

```



```
## -----
##      traveling      resting
## mean 5.2629057 1.31835179
## sd   0.2401494 0.04258705
##
## Regression coeffs for the transition probabilities:
## -----
##           1 -> 2 2 -> 1
## (Intercept)  -1.5   -1.5
##
## Transition probability matrix:
## -----
##           traveling      resting
## traveling 0.8175745 0.1824255
## resting   0.1824255 0.8175745
##
## Initial distribution:
## -----
## traveling      resting
##           0.5         0.5
```

The HMM on the average MSA data didn't work, basically the max log likelihood was infinity which is not good... maybe means that MSA is not a good summary statistic.

Multivariate HMMs

The best candidate variables seem to be the average speed, the mean resultant length, and the maximum depth for each 5 minute segment.

A model with those 3 variables and 2 states:

```
var3_test<-all_all[,c(2,3,4)]

var3_test_data<-prepData(var3_test, type='LL',coordNames = NULL)

dist<-list(avg_speed="gamma",mean_resultant_length="beta",max_depth="gamma")

Par0 <- list( avg_speed=c(rest_params['alpha'],travel_params['alpha'] , rest_params['beta'], travel_params['beta']),
              mean_resultant_length= c(rest_mrl_params['alpha'],travel_mrl_params['alpha'] , rest_mrl_params['beta'], travel_mrl_params['beta']),
              max_depth=c(rest_depth_params['alpha'], travel_depth_params['alpha'], rest_depth_params['beta'], travel_depth_params['beta']))

stateNames <- c("traveling","resting")

test_3vars <- fitHMM(var3_test_data, nbStates=2, dist=dist, Par0=Par0, stateNames=stateNames)

## =====
## Fitting HMM with 2 states and 3 data streams
## -----
## avg_speed ~ gamma(mean=~1, sd=~1)
## mean_resultant_length ~ beta(shape1=~1, shape2=~1)
## max_depth ~ gamma(mean=~1, sd=~1)
##
## Transition probability matrix formula: ~1
```

```

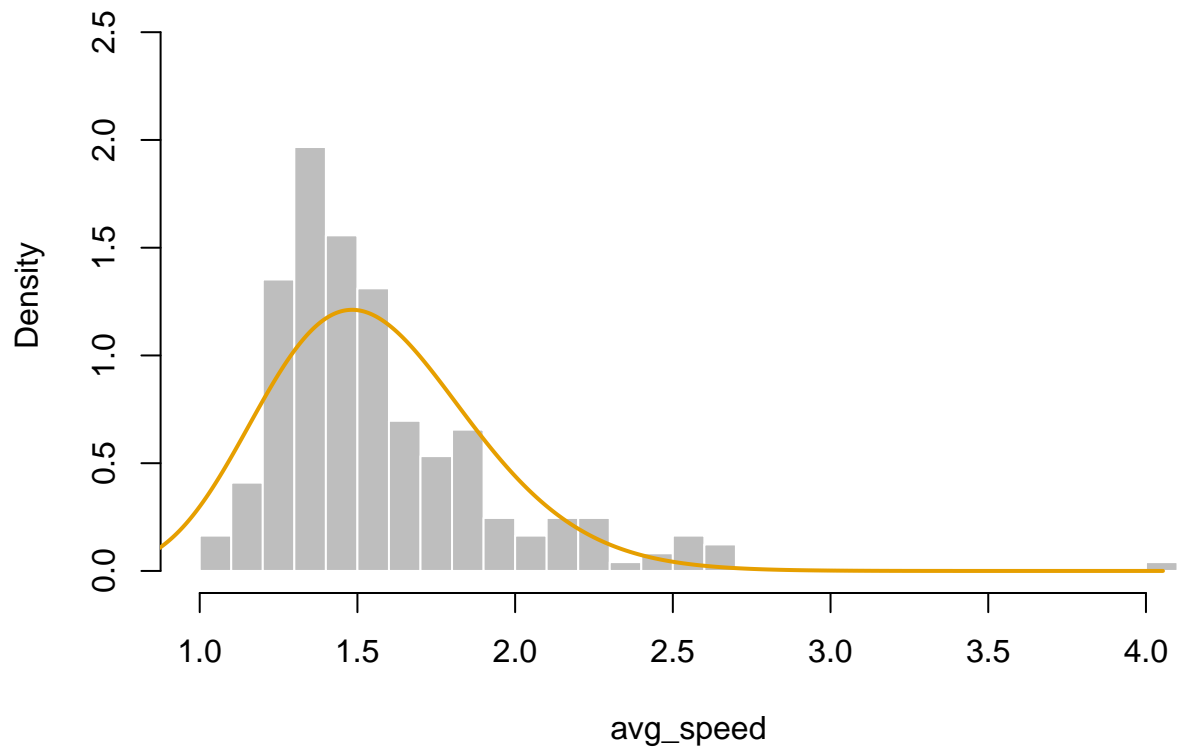
##
## Initial distribution formula: ~1
## =====
## DONE
print(test_3vars)

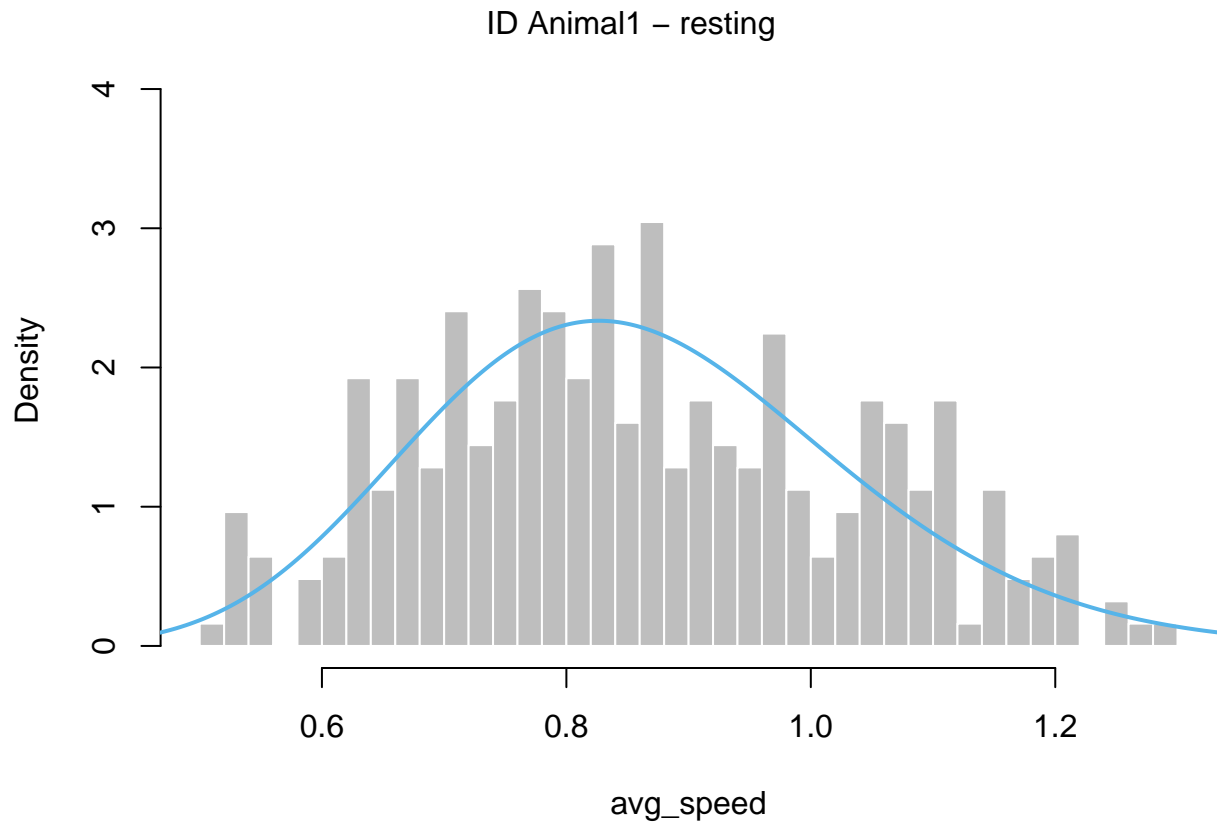
## Value of the maximum log-likelihood: -862.7959
##
##
## avg_speed parameters:
## -----
##      traveling   resting
## mean 1.5556149 0.8616075
## sd   0.3358055 0.1737782
##
## mean_resultant_length parameters:
## -----
##      traveling   resting
## shape1 2.8424114 1.659252
## shape2 0.7134265 1.042105
##
## max_depth parameters:
## -----
##      traveling   resting
## mean  6.382661 2.6911358
## sd    2.850409 0.7475391
##
## Regression coeffs for the transition probabilities:
## -----
##              1 -> 2    2 -> 1
## (Intercept) -2.134131 -2.369454
##
## Transition probability matrix:
## -----
##              traveling   resting
## traveling 0.89417657 0.1058234
## resting   0.08553182 0.9144682
##
## Initial distribution:
## -----
##      traveling      resting
## 9.923066e-07 9.999990e-01
plot(test_3vars, plotCI = TRUE, breaks=30, sepStates=TRUE, ask=TRUE)

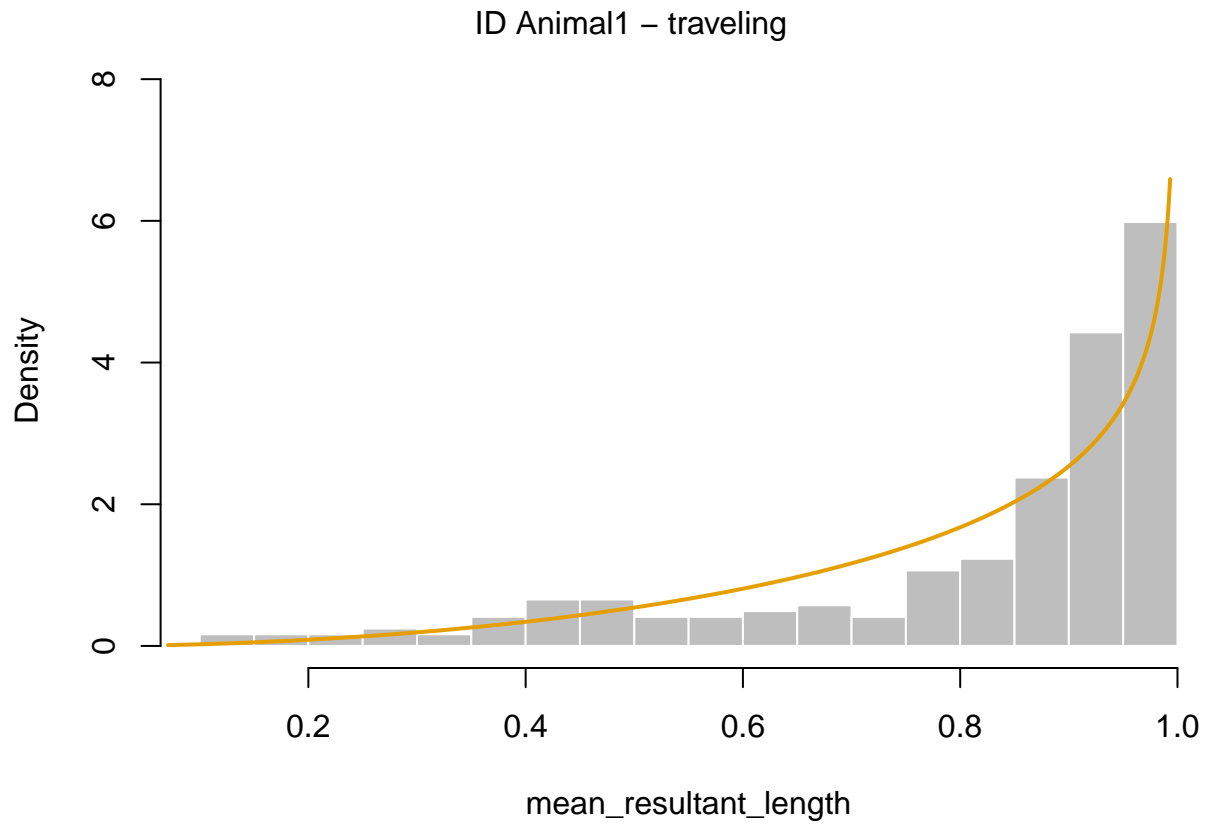
## Decoding state sequence... DONE

```

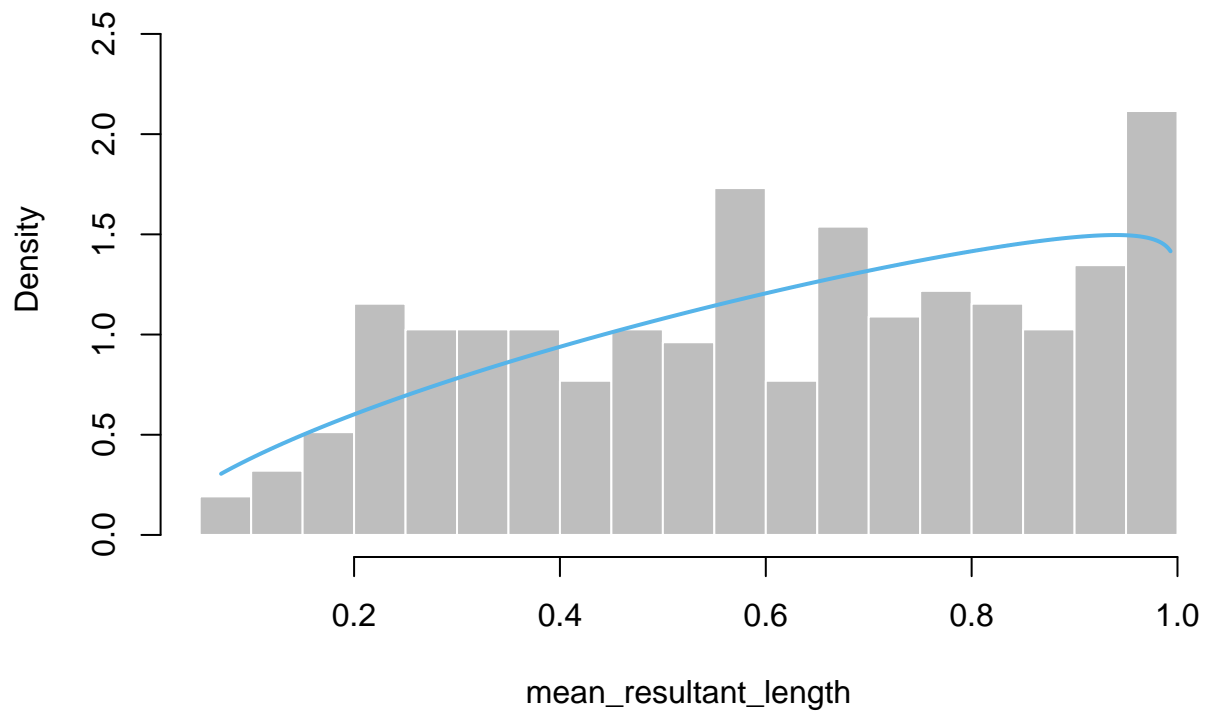
ID Animal1 – traveling



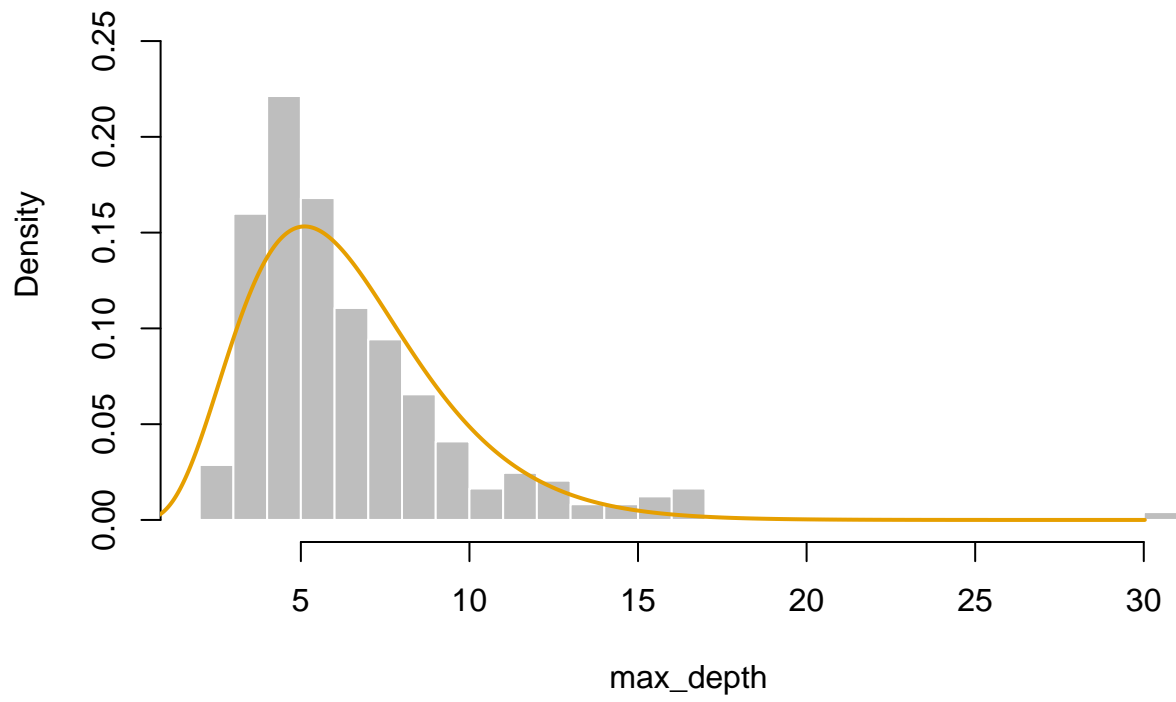


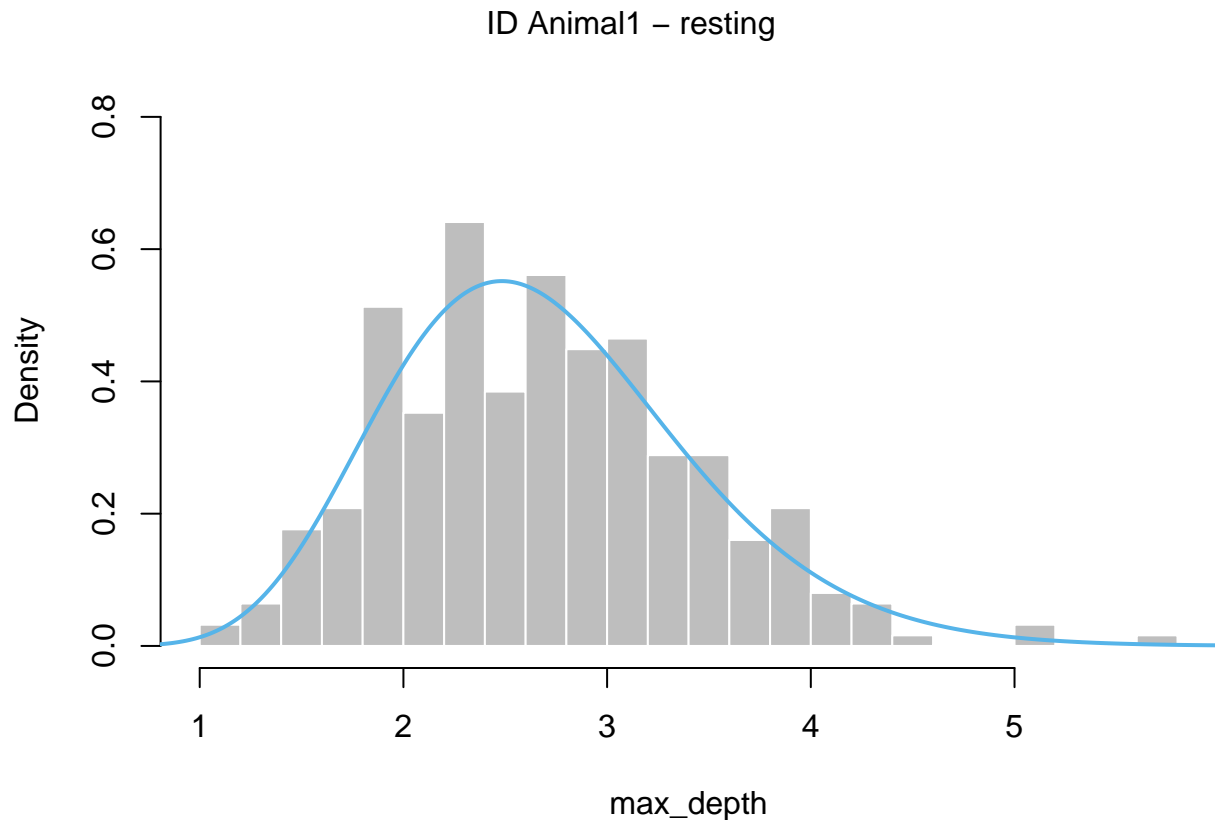


ID Animal1 – resting



ID Animal1 – traveling





Now try a model with the same 3 variables and 3 states: resting, traveling, and other. The other state is when the video watchers were unsure if the animal was traveling or resting, so it is unclear yet biologically what this state is.

```
other_params = moment_match_gamma(
  mean(all_other$avg_speed),
  sd(all_other$avg_speed)
)

other_mrl_params = moment_match_beta(
  mean(all_other$mean_resultant_length),
  var(all_other$mean_resultant_length)
)

other_depth_params = moment_match_gamma(
  mean(all_other$max_depth),
  sd(all_other$max_depth)
)

dist<-list(avg_speed="gamma",mean_resultant_length="beta",max_depth="gamma") #same as before
Par0 <- list( avg_speed=c(rest_params['alpha'],travel_params['alpha'], other_params['alpha'], rest_params['alpha']),
  mean_resultant_length= c(rest_mrl_params['alpha'],travel_mrl_params['alpha'], other_mrl_params['alpha'], rest_mrl_params['alpha']),
  max_depth=c(rest_depth_params['alpha'], travel_depth_params['alpha'],other_depth_params['alpha'], rest_depth_params['alpha'])

stateNames <- c("traveling", "other", "resting")

test_3vars_3st <- fitHMM(var3_test_data, nbStates=3, dist=dist, Par0=Par0, stateNames=stateNames)
```



```

## =====
## Fitting HMM with 3 states and 3 data streams
## -----
## avg_speed ~ gamma(mean=~1, sd=~1)
## mean_resultant_length ~ beta(shape1=~1, shape2=~1)
## max_depth ~ gamma(mean=~1, sd=~1)
##
## Transition probability matrix formula: ~1
##
## Initial distribution formula: ~1
## =====
## DONE
print(test_3vars_3st)

## Value of the maximum log-likelihood: -659.3679
##
##
## avg_speed parameters:
## -----
##      traveling      other      resting
## mean 1.8991738 0.8481820 1.3430907
## sd   0.3497262 0.1623112 0.1261911
##
## mean_resultant_length parameters:
## -----
##      traveling      other      resting
## shape1 5.5936954 1.647284 2.1840239
## shape2 0.9512027 1.040729 0.6739152
##
## max_depth parameters:
## -----
##      traveling      other      resting
## mean  9.358082 2.6383512 4.569592
## sd    3.164639 0.7141967 1.063880
##
## Regression coeffs for the transition probabilities:
## -----
##              1 -> 2    1 -> 3    2 -> 1    2 -> 3    3 -> 1    3 -> 2
## (Intercept) -3.684055 -1.237185 -3.811611 -2.277468 -1.96011 -1.326603
##
## Transition probability matrix:
## -----
##      traveling      other      resting
## traveling 0.76027071 0.01909868 0.22063061
## other     0.01966158 0.88916073 0.09117768
## resting   0.10015710 0.18871673 0.71112617
##
## Initial distribution:
## -----

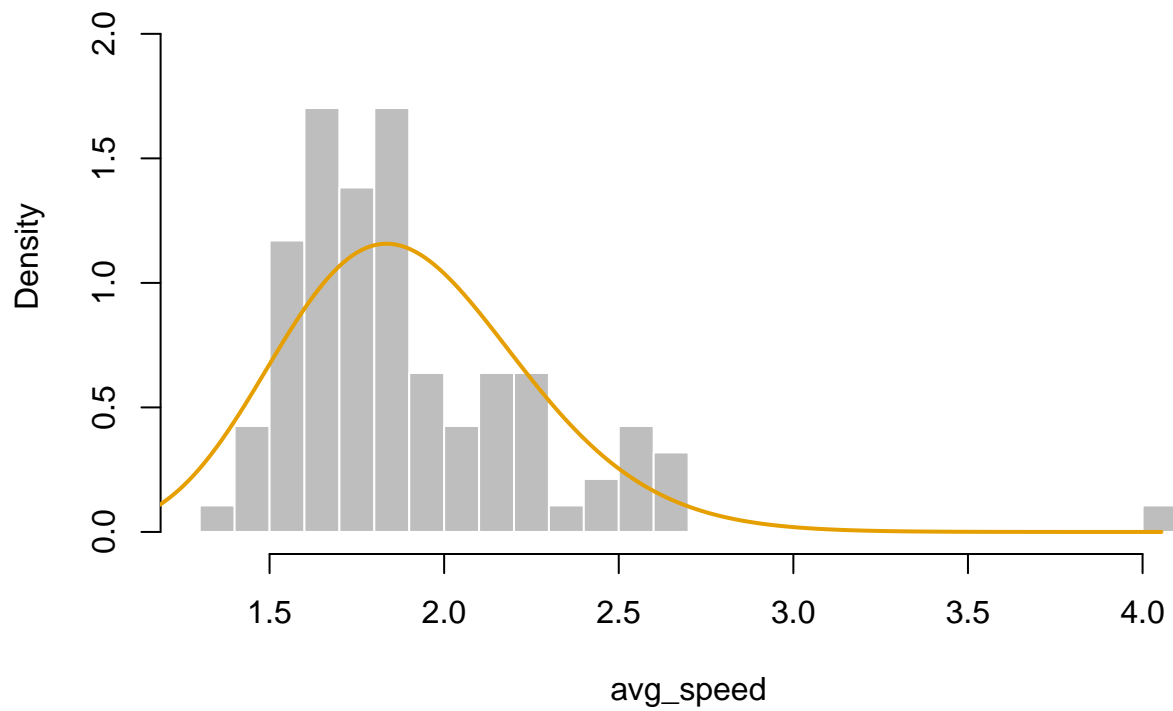
```

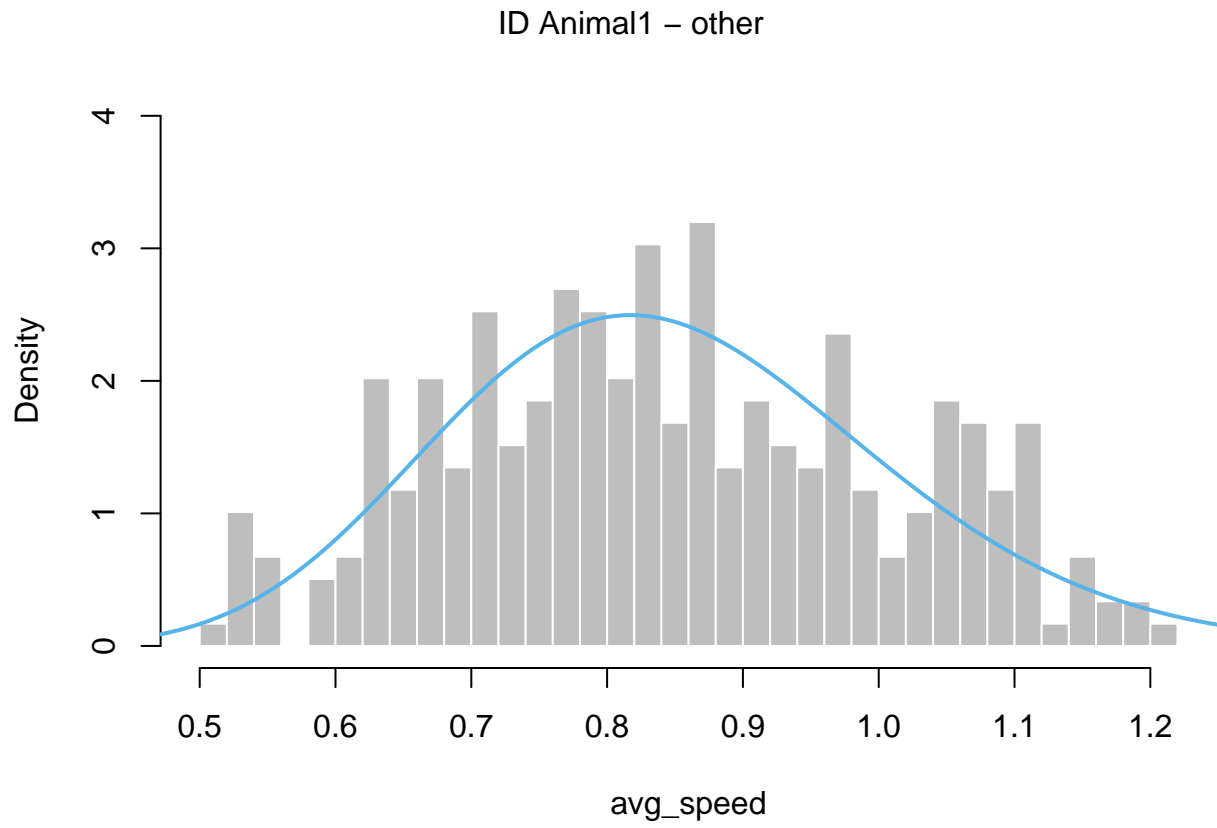
```
##      traveling      other      resting  
## 4.903545e-08 1.000000e+00 1.561386e-10
```

```
plot(test_3vars_3st, plotCI = TRUE, breaks=30, sepStates=TRUE, ask=TRUE)
```

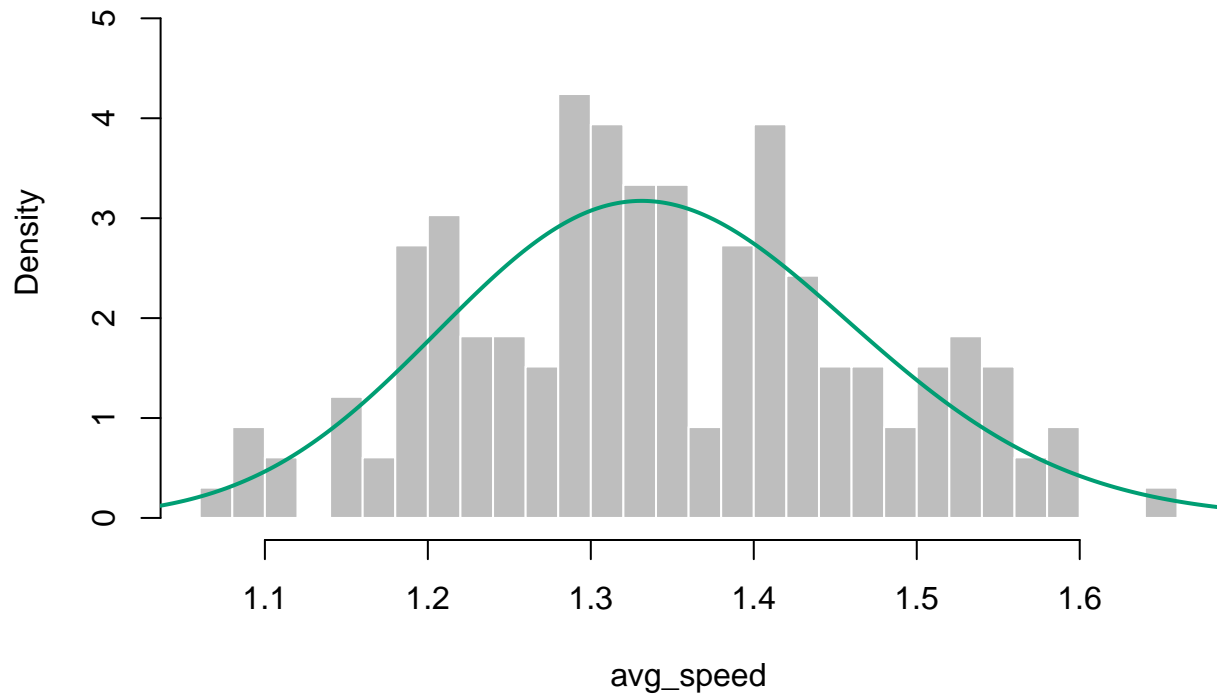
```
## Decoding state sequence... DONE
```

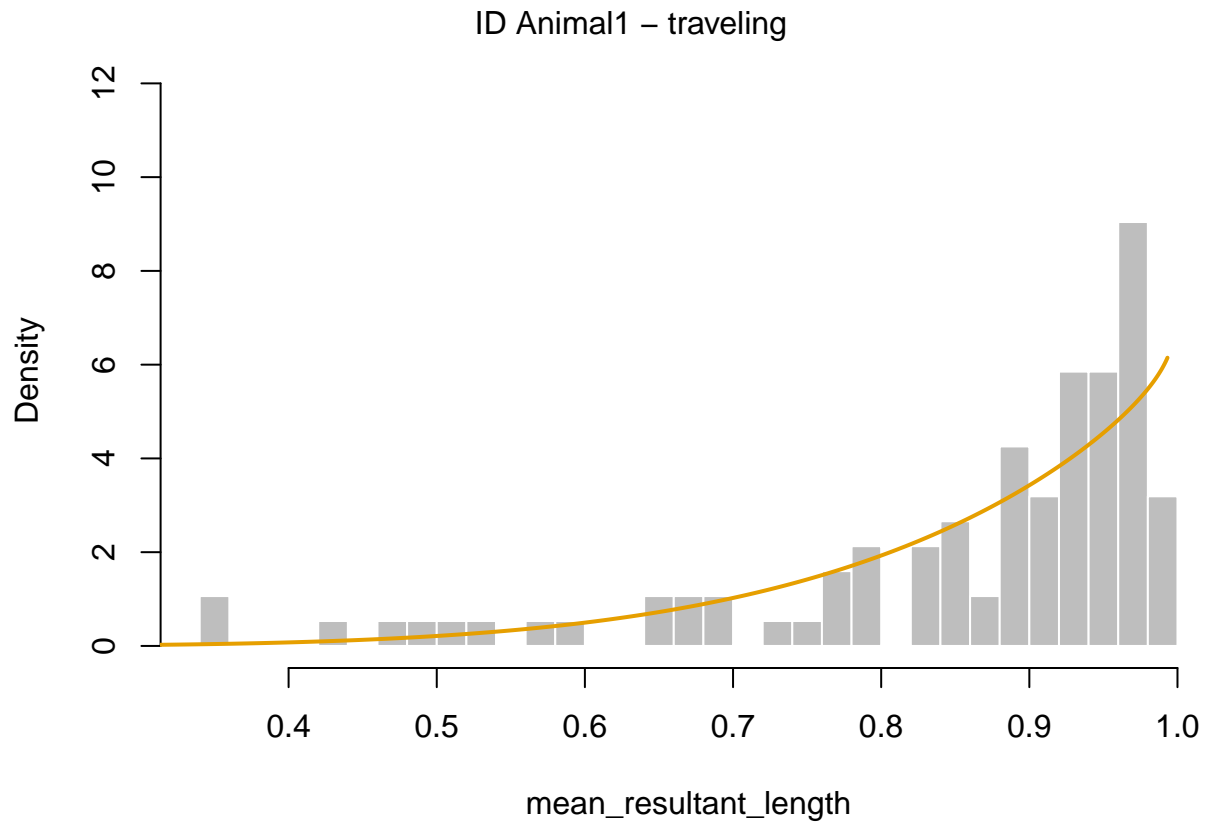
ID Animal1 – traveling



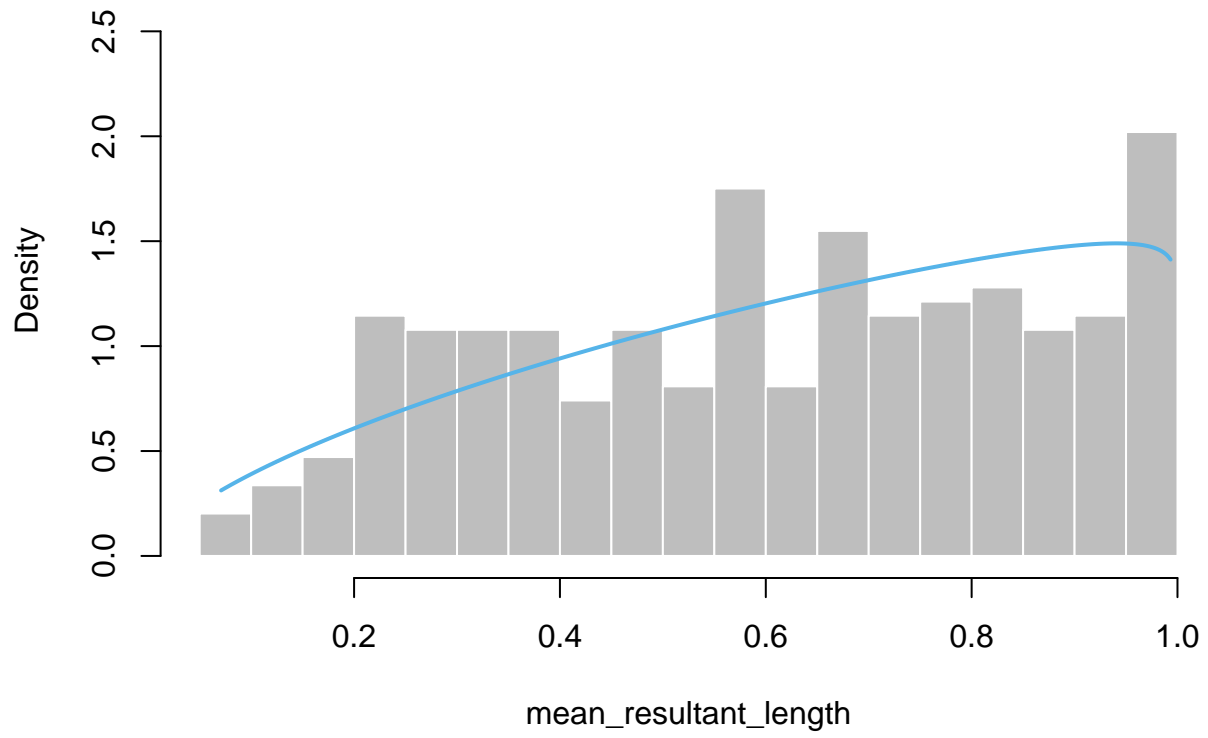


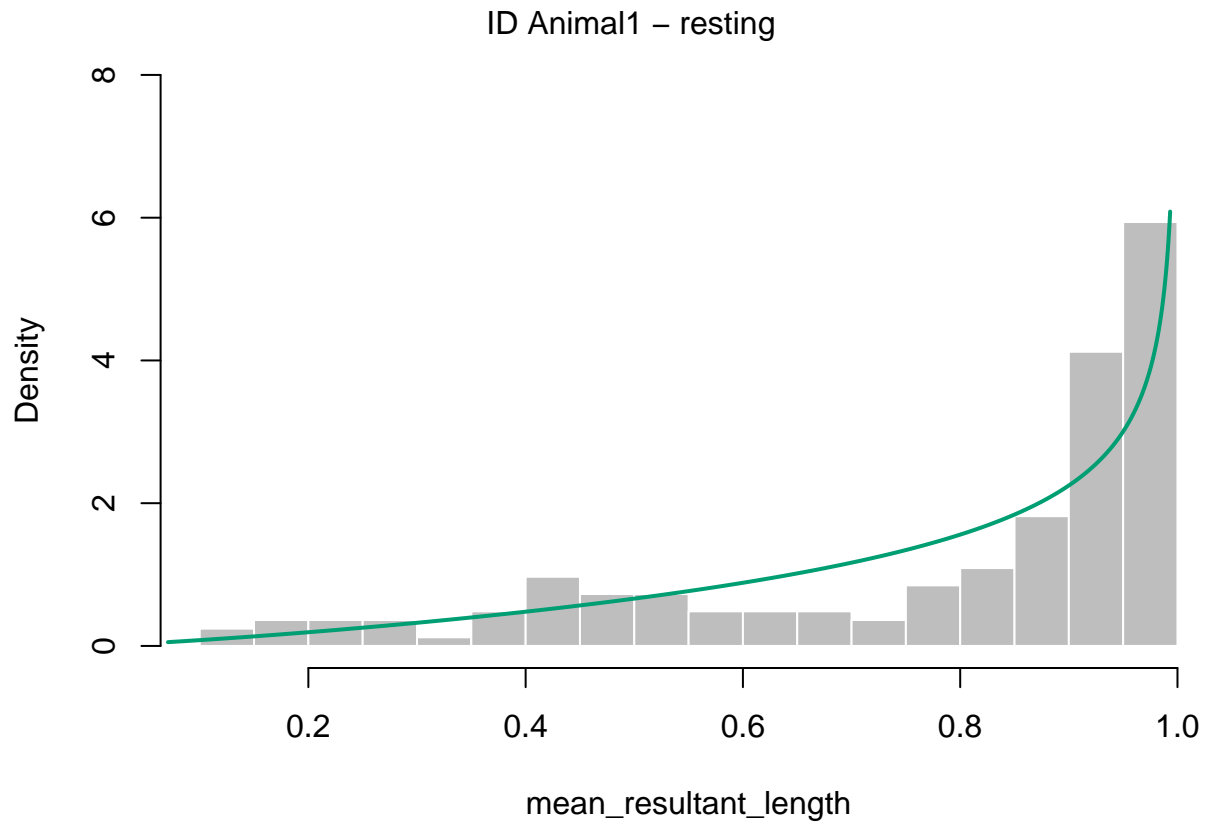
ID Animal1 – resting

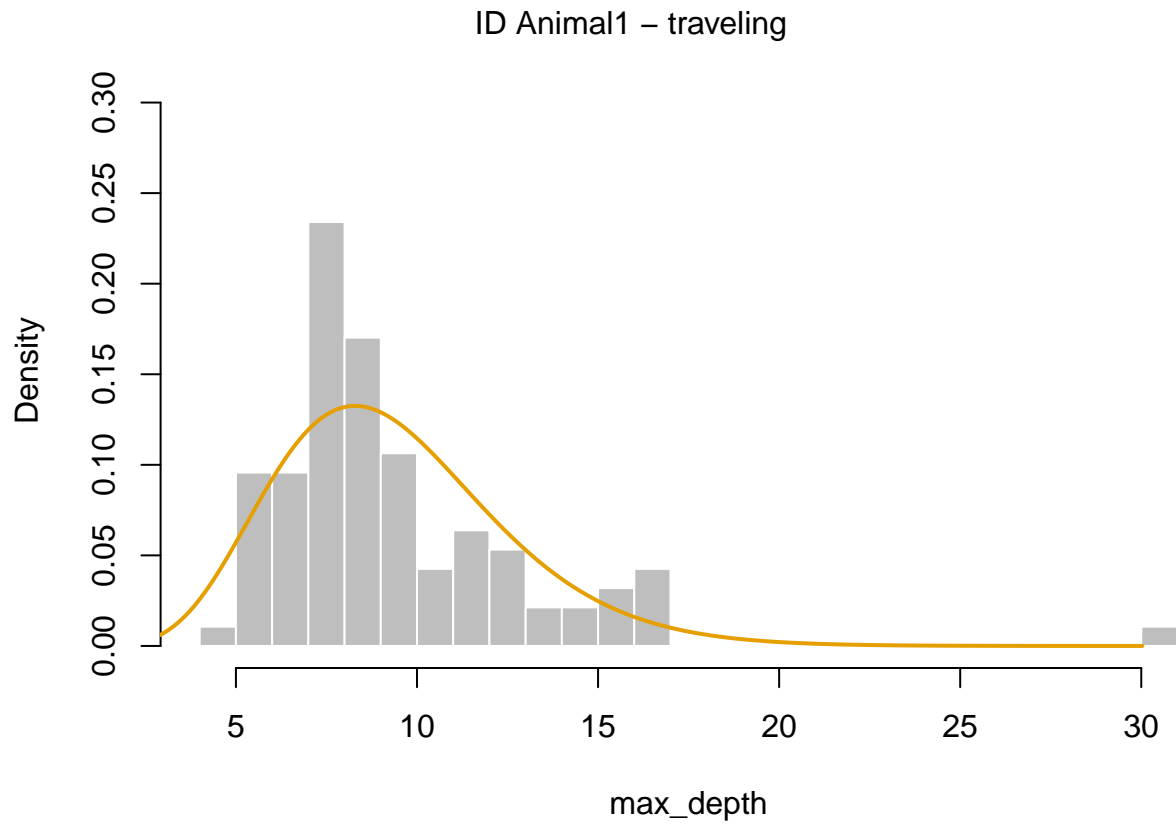




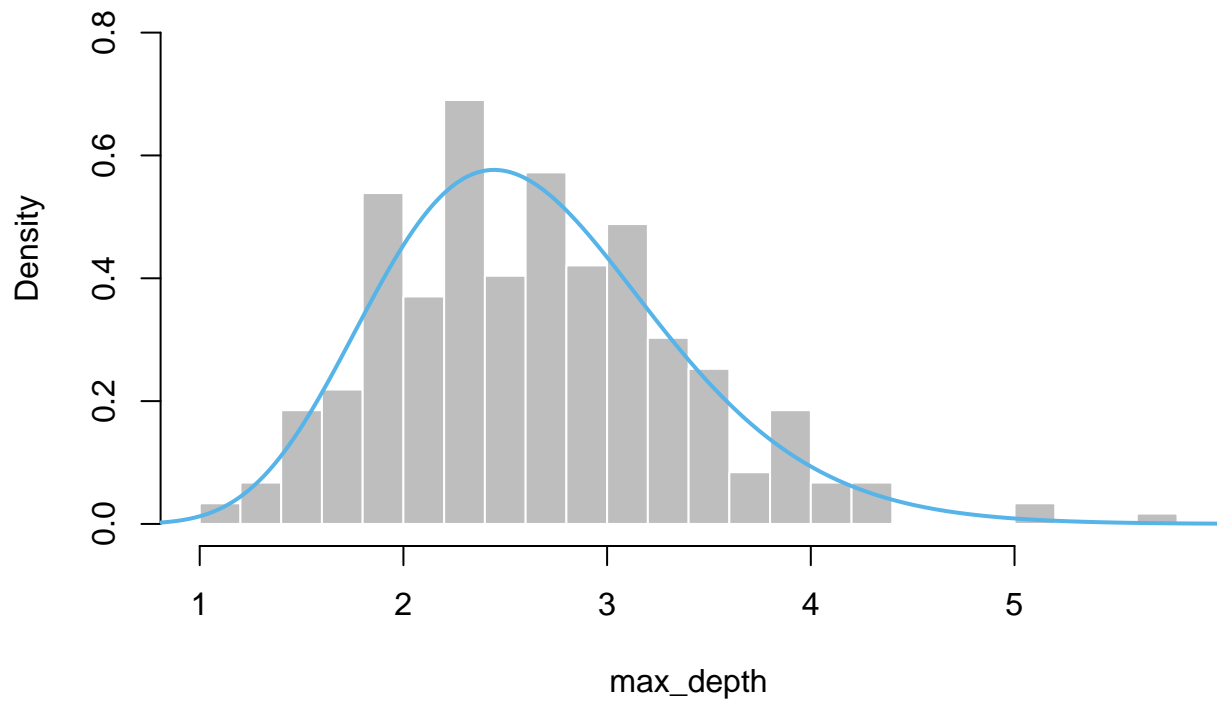
ID Animal1 – other

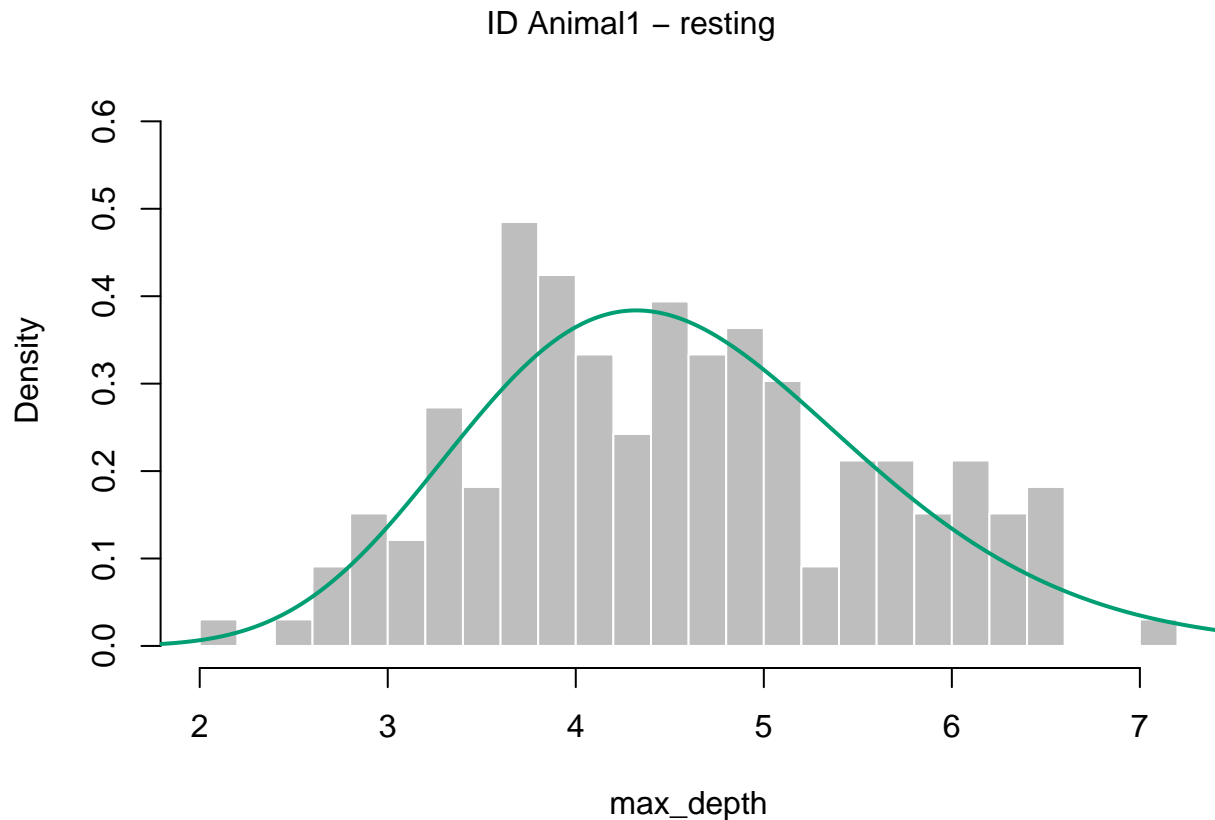






ID Animal1 – other





Now compare the AIC values of the models with different numbers of states

```
AIC(test_3vars, test_3vars_3st, k = 2, n = NULL)
```

```
##           Model      AIC
## 1 test_3vars_3st 1370.736
## 2      test_3vars 1755.592
```

```
AICweights(test_3vars, test_3vars_3st, k = 2, n = NULL)
```

```
##           Model      weight
## 1 test_3vars_3st 1.000000e+00
## 2      test_3vars 2.688841e-84
```

so the model with three states better fits the data, but this third state is maybe not that important

Incorporating Covariates into the HMMS

Important covariates: Individual identity (some whales might naturally behave differently from others) - this would affect the transition probabilities and the state distributions I think Hour of day (there could be cycles in the whales behavior) - this would just affect the transition probabilities? Age of the calf (as they age, calves might behave differently) - this would affect the transition probabilities and the state distributions I think

```
calf_demo = read.csv('calf_demo_data.csv')
```

```
demo_hr<-calf_demo[,c(1,3,4,5,7)]
```

```
#try the model on this data set not broken up into traveling or resting
```

```

data_hr<-prepData(demo_hr, type='LL',coordNames = NULL)
dist<-list(avg_speed="gamma",mrl="beta",max_depth="gamma")

Par0 <- list( avg_speed=c(rest_params['alpha'],travel_params['alpha'] , rest_params['beta'], travel_params['beta']),
              mrl= c(rest_mrl_params['alpha'],travel_mrl_params['alpha'] , rest_mrl_params['beta'], travel_mrl_params['beta']),
              max_depth=c(rest_depth_params['alpha'], travel_depth_params['alpha'], rest_depth_params['beta'], travel_depth_params['beta']))

stateNames <- c("resting","traveling")

test_3vars_all <- fitHMM(data_hr, nbStates=2, dist=dist, Par0=Par0, stateNames=stateNames)

## =====
## Fitting HMM with 2 states and 3 data streams
## -----
## avg_speed ~ gamma(mean=~1, sd=~1)
## mrl ~ beta(shape1=~1, shape2=~1)
## max_depth ~ gamma(mean=~1, sd=~1)
##
## Transition probability matrix formula: ~1
##
## Initial distribution formula: ~1
## =====
## DONE
print(test_3vars_all)

## Value of the maximum log-likelihood: -3425.762
##
##
## avg_speed parameters:
## -----
##           resting traveling
## mean 0.9606769 1.5234772
## sd   0.2203391 0.3949817
##
## mrl parameters:
## -----
##           resting traveling
## shape1 1.994863 8.351758
## shape2 1.538289 1.015229
##
## max_depth parameters:
## -----
##           resting traveling
## mean 20.301384 35.69286
## sd   8.851783 29.98266
##
## Regression coeffs for the transition probabilities:
## -----
##           1 -> 2    2 -> 1

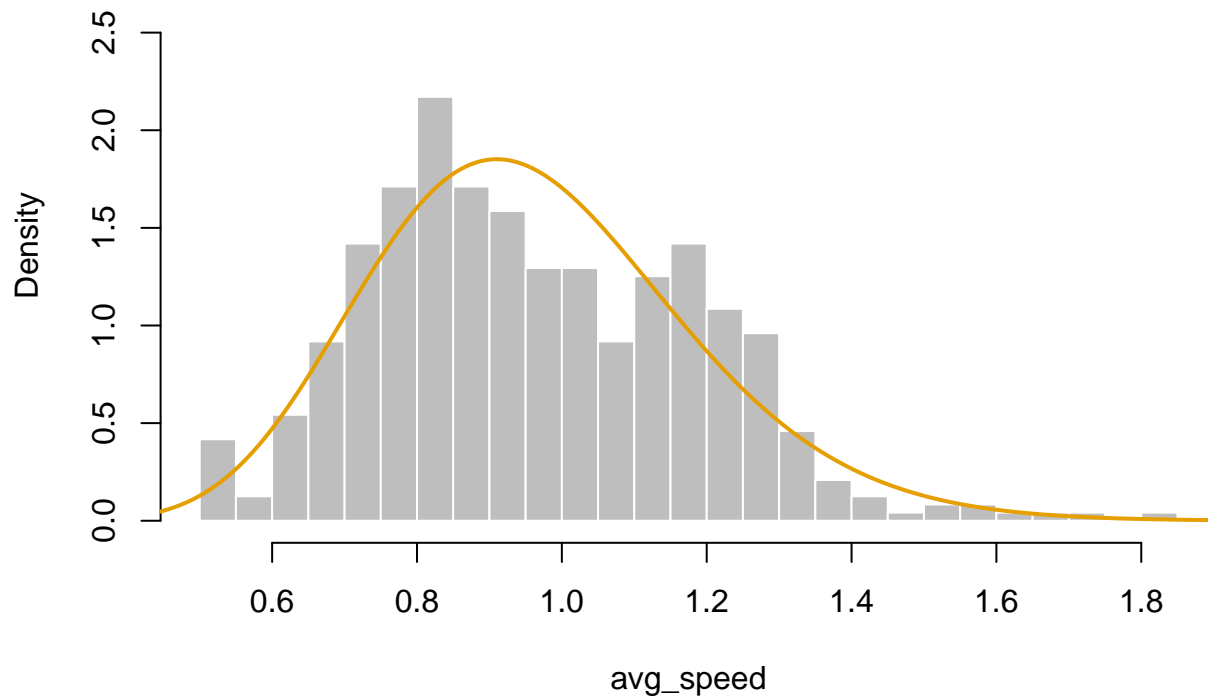
```

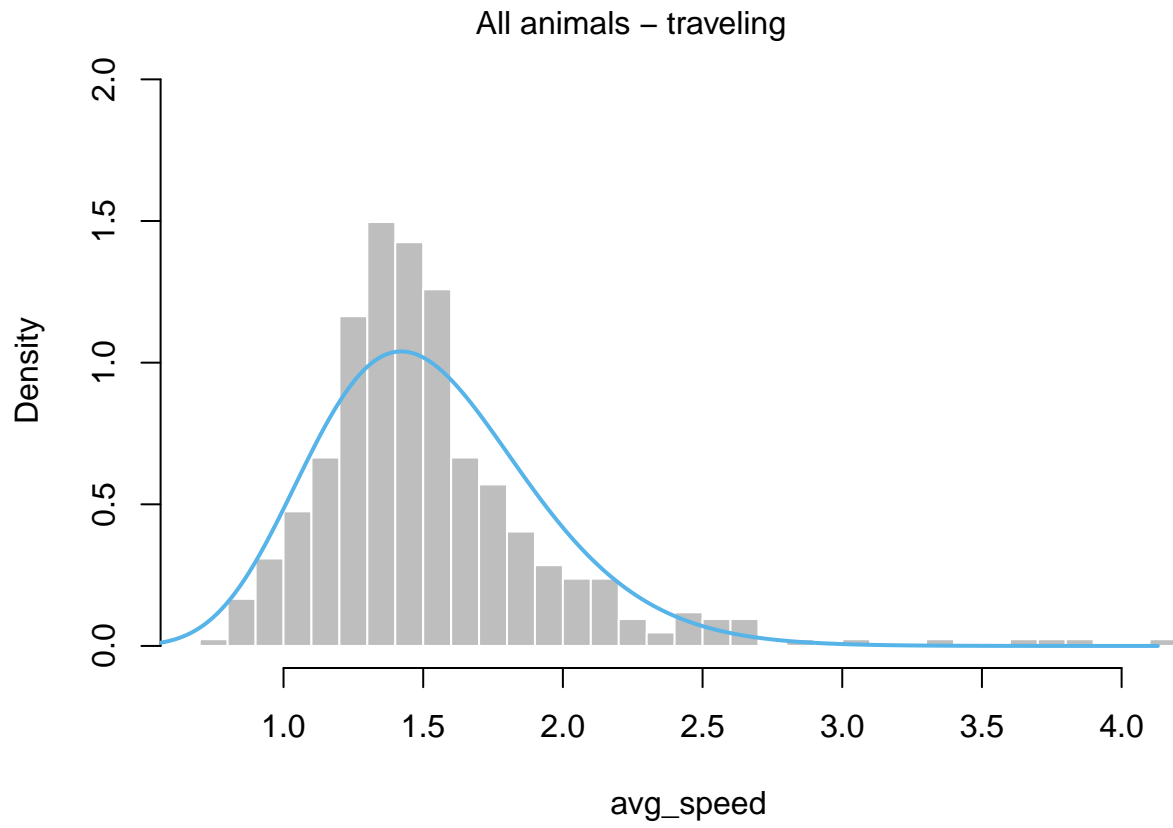
```
## (Intercept) -2.19158 -2.014586
##
## Transition probability matrix:
## -----
##           resting traveling
## resting  0.8994908 0.1005092
## traveling 0.1176800 0.8823200
##
## Initial distribution:
## -----
##           resting traveling
## 0.1814039 0.8185961
```

```
plot(test_3vars_all, plotCI = TRUE, breaks=30, sepStates=TRUE, ask=TRUE)
```

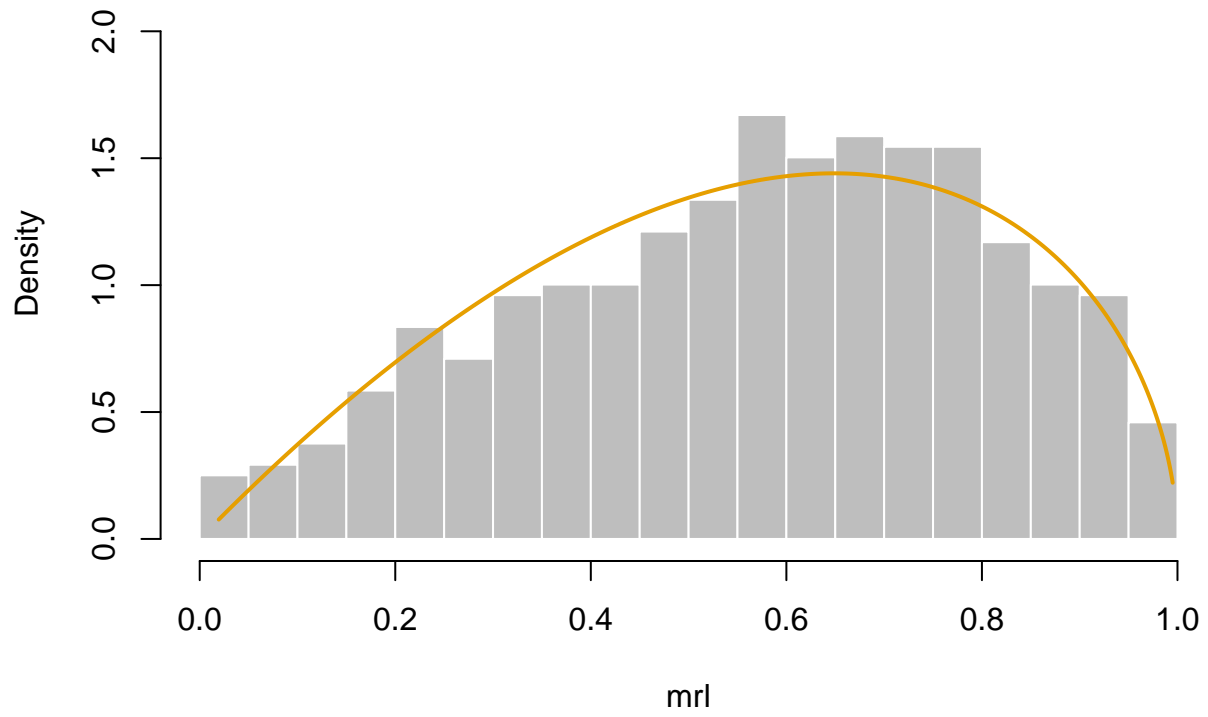
```
## Decoding state sequence... DONE
```

All animals – resting

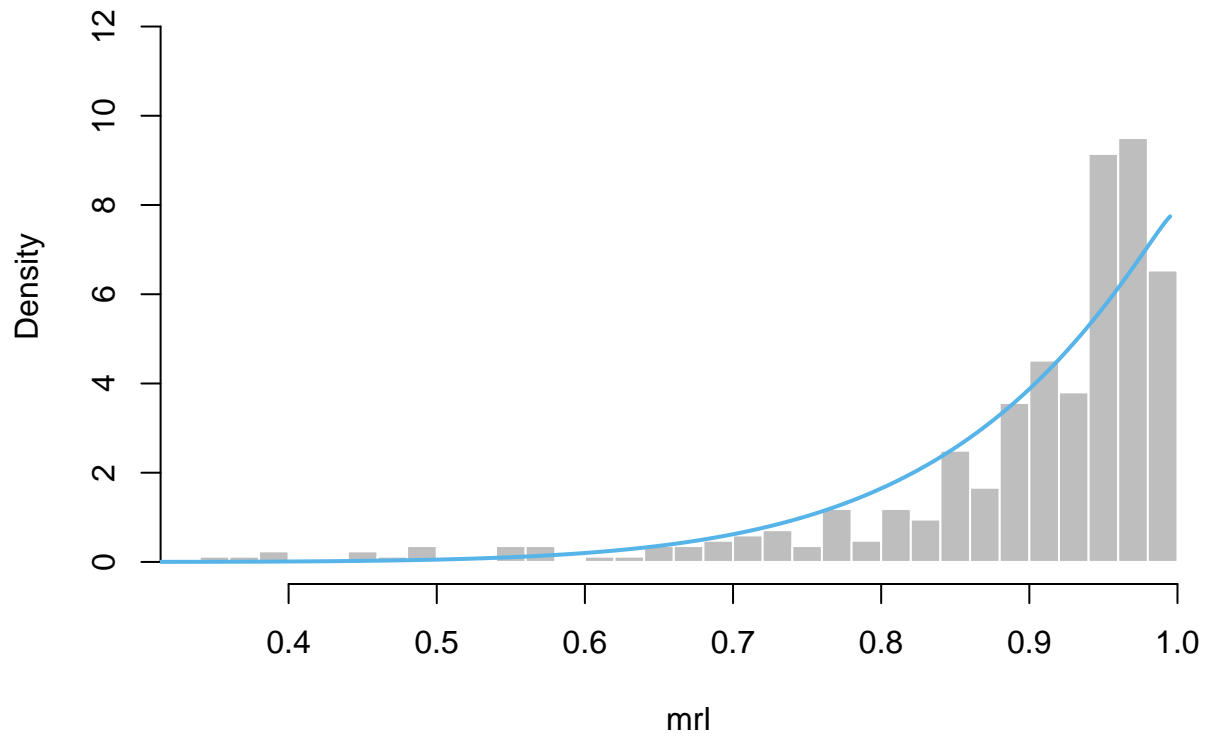


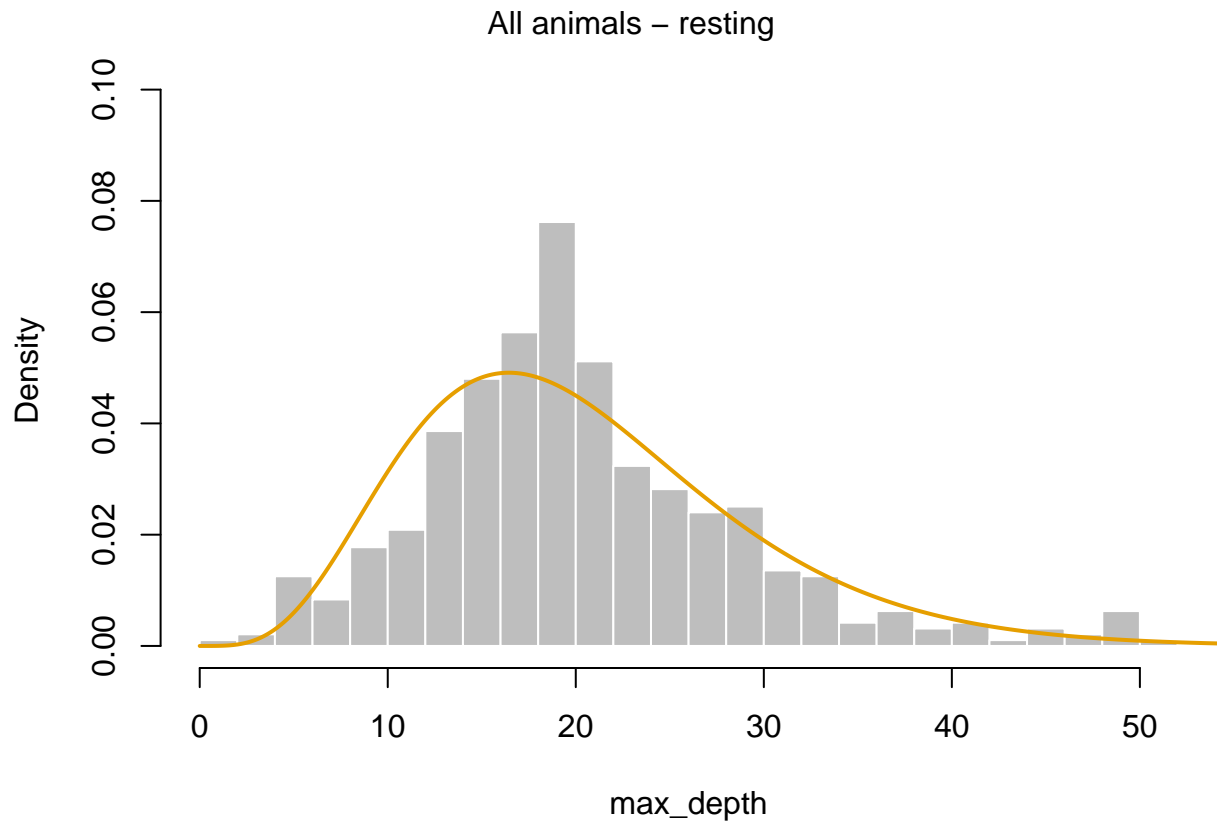


All animals – resting

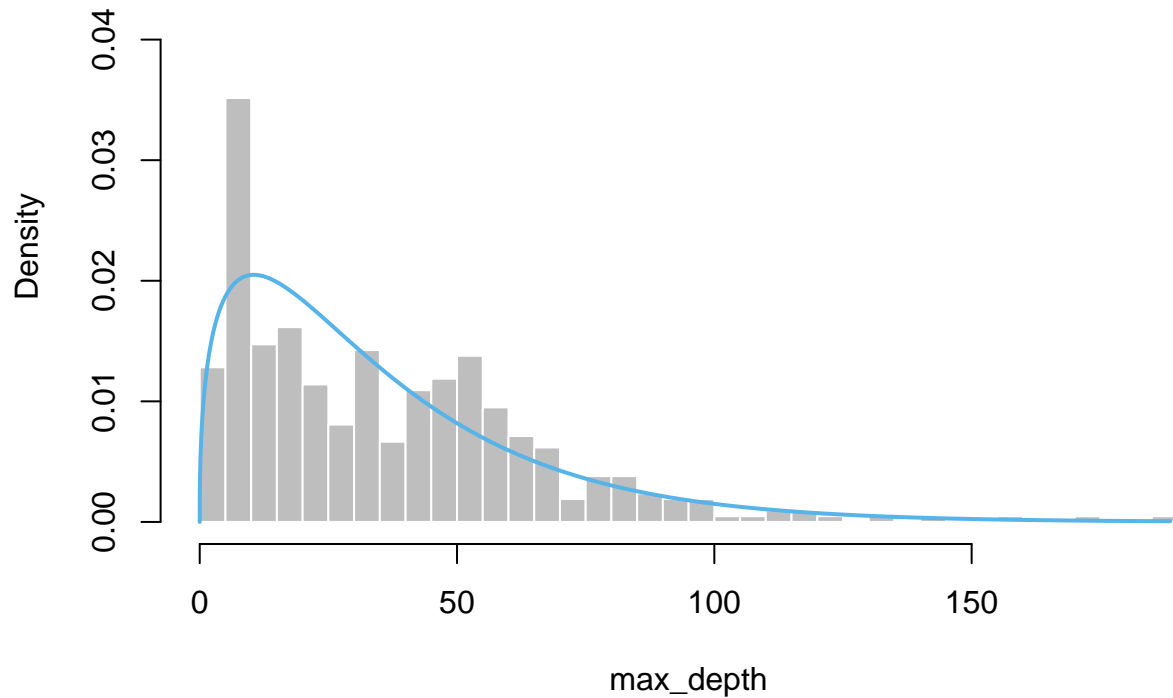


All animals – traveling





All animals – traveling



```
formula <- ~ cosinor(hour, period = 24)

test_3vars_hr <- fitHMM(data_hr, nbStates=2, dist=dist, Par0=Par0, stateNames=stateNames, formula=formula)

## =====
## Fitting HMM with 2 states and 3 data streams
## -----
## avg_speed ~ gamma(mean=~1, sd=~1)
## mrl ~ beta(shape1=~1, shape2=~1)
## max_depth ~ gamma(mean=~1, sd=~1)
##
## Transition probability matrix formula: ~cosinor(hour, period = 24)
##
## Initial distribution formula: ~1
## =====
## DONE
print(test_3vars_hr)

## Value of the maximum log-likelihood: -3417.372
##
##
## avg_speed parameters:
```

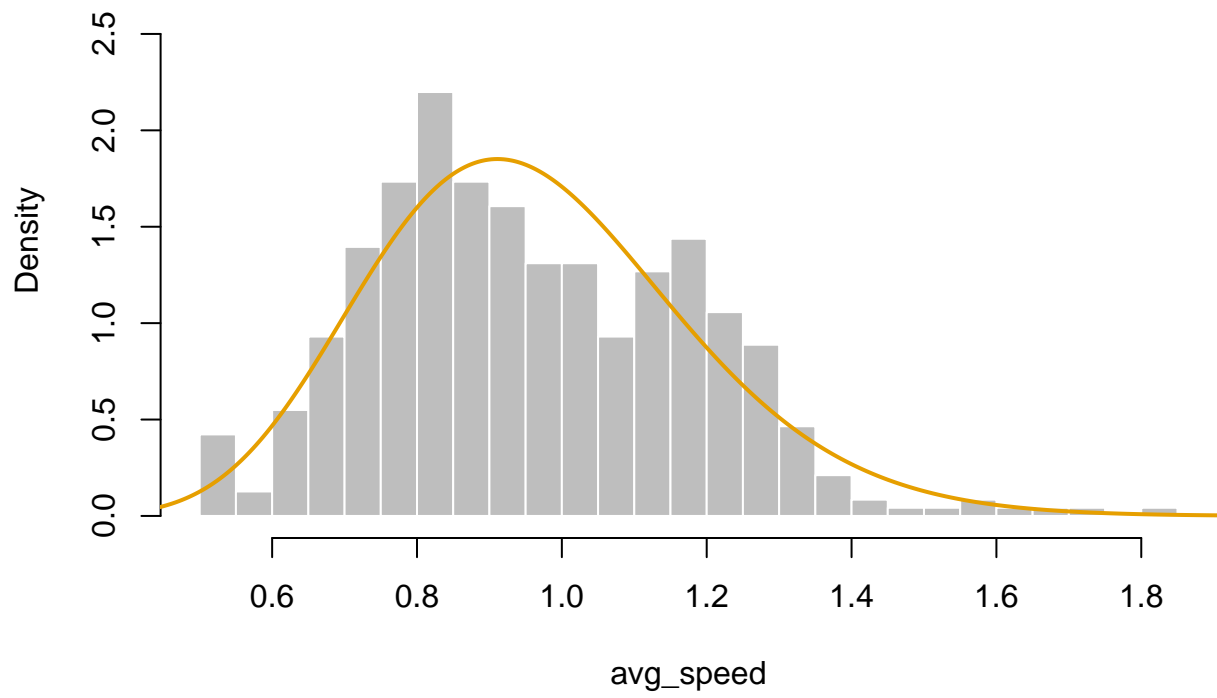
```

## -----
##           resting traveling
## mean 0.9615883 1.5278255
## sd   0.2203893 0.3939728
##
## mrl parameters:
## -----
##           resting traveling
## shape1 1.968265  8.144833
## shape2 1.493568  1.002109
##
## max_depth parameters:
## -----
##           resting traveling
## mean 20.241954  35.90724
## sd   8.857007  30.13756
##
## Regression coeffs for the transition probabilities:
## -----
##                                     1 -> 2      2 -> 1
## (Intercept)                      -2.4101336 -0.91116911
## cosinorCos(hour, period = 24)  0.2296027  1.49237105
## cosinorSin(hour, period = 24) -1.1697199  0.03183823
##
## Transition probability matrix (based on mean covariate values):
## -----
##           resting traveling
## resting   0.91025214 0.08974786
## traveling 0.08650095 0.91349905
##
## Initial distribution:
## -----
##           resting traveling
## 0.178297  0.821703
plot(test_3vars_hr, plotCI = TRUE, breaks=30, sepStates=TRUE, ask=TRUE)

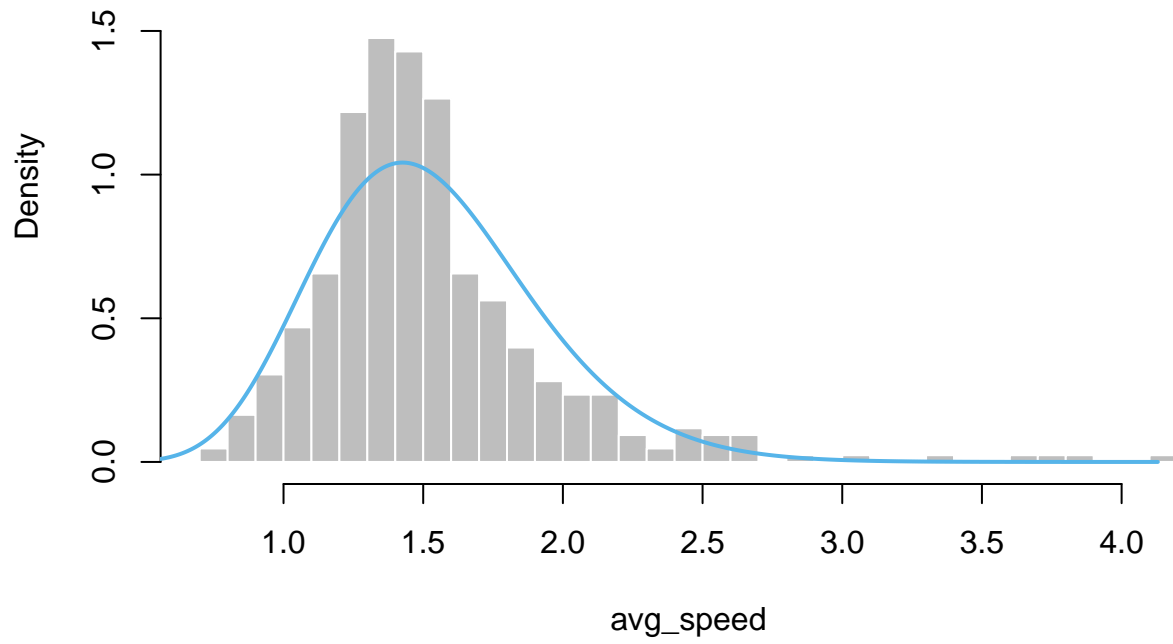
## Decoding state sequence... DONE

```

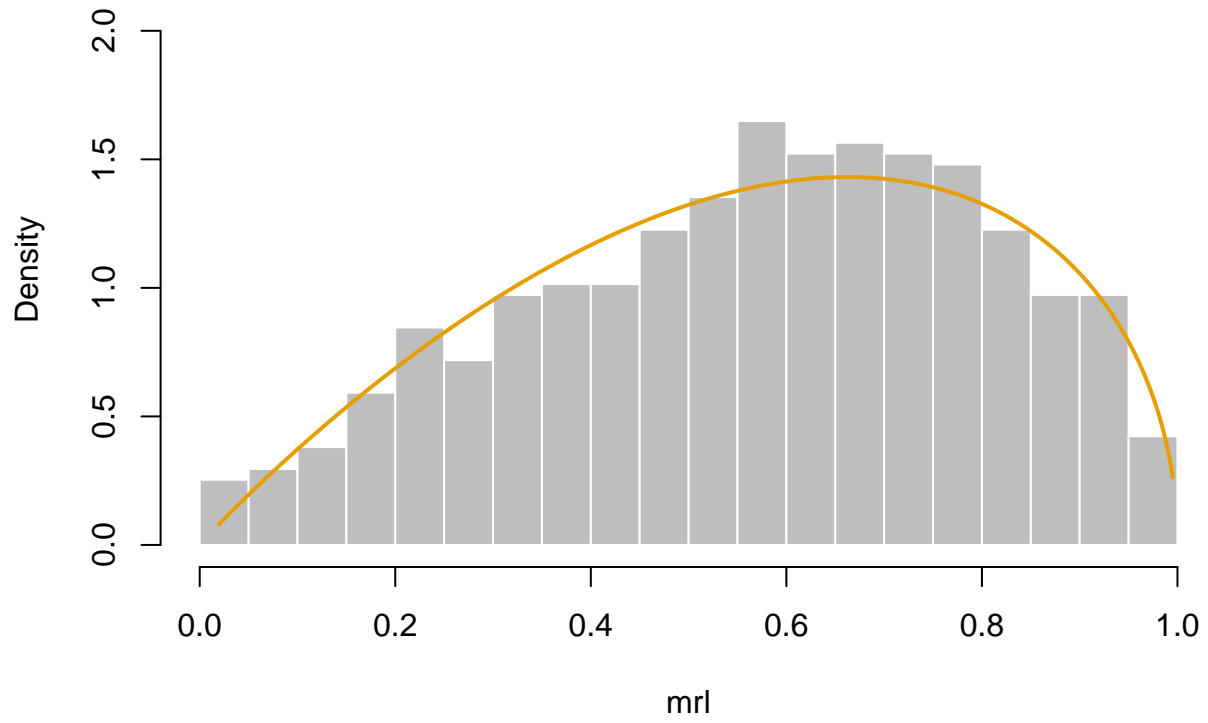
All animals – resting

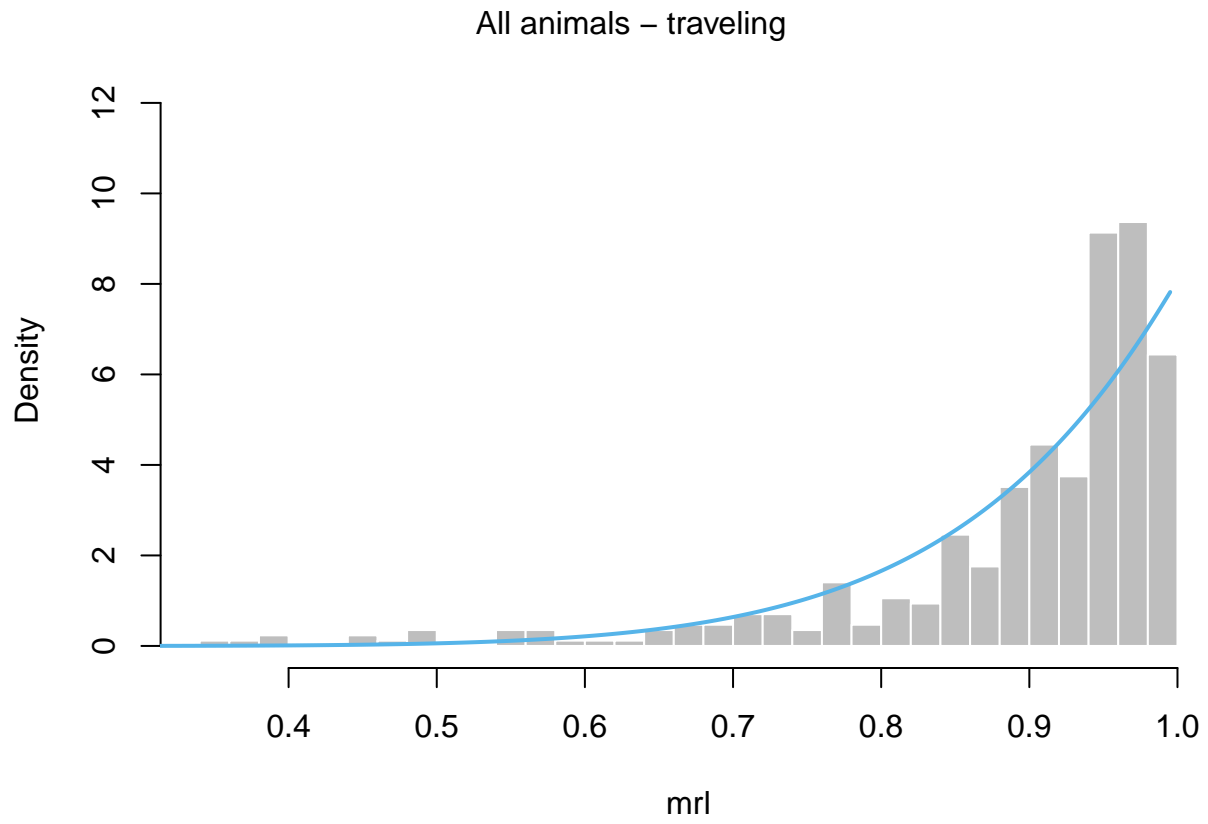


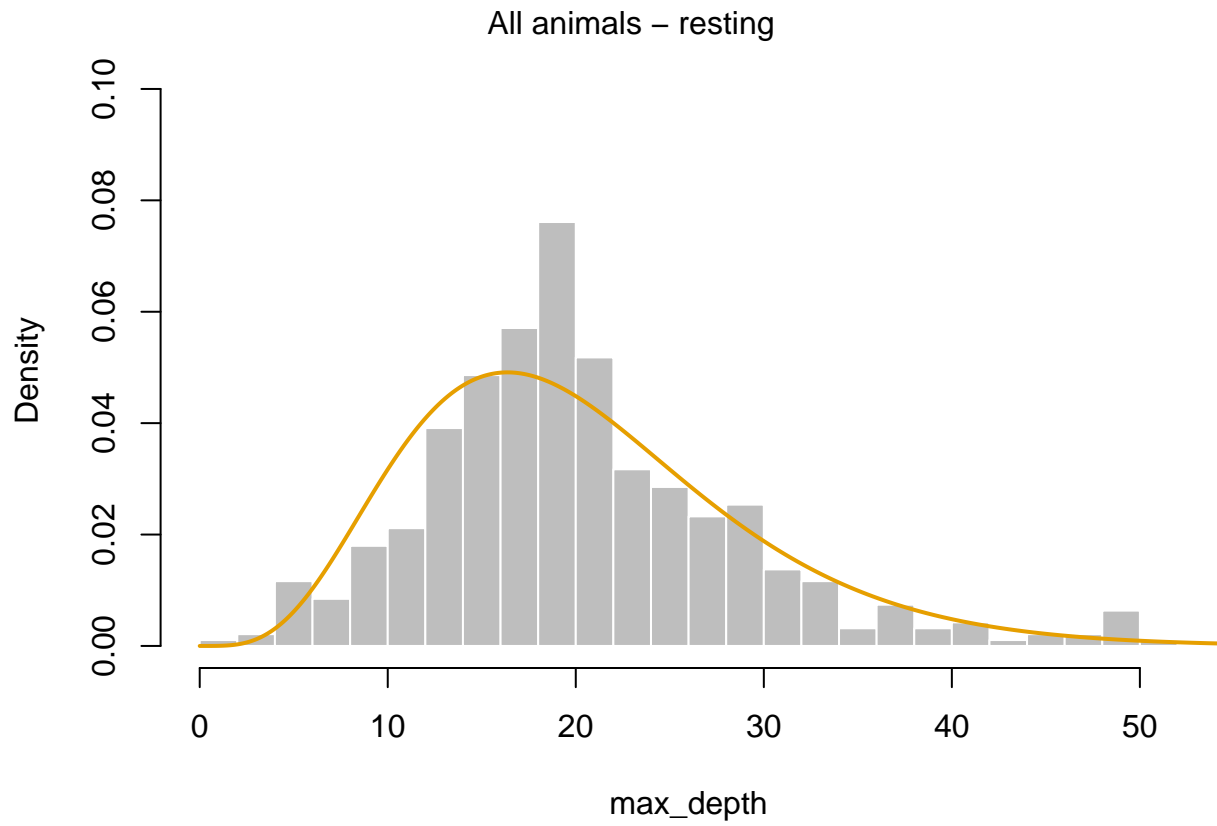
All animals – traveling



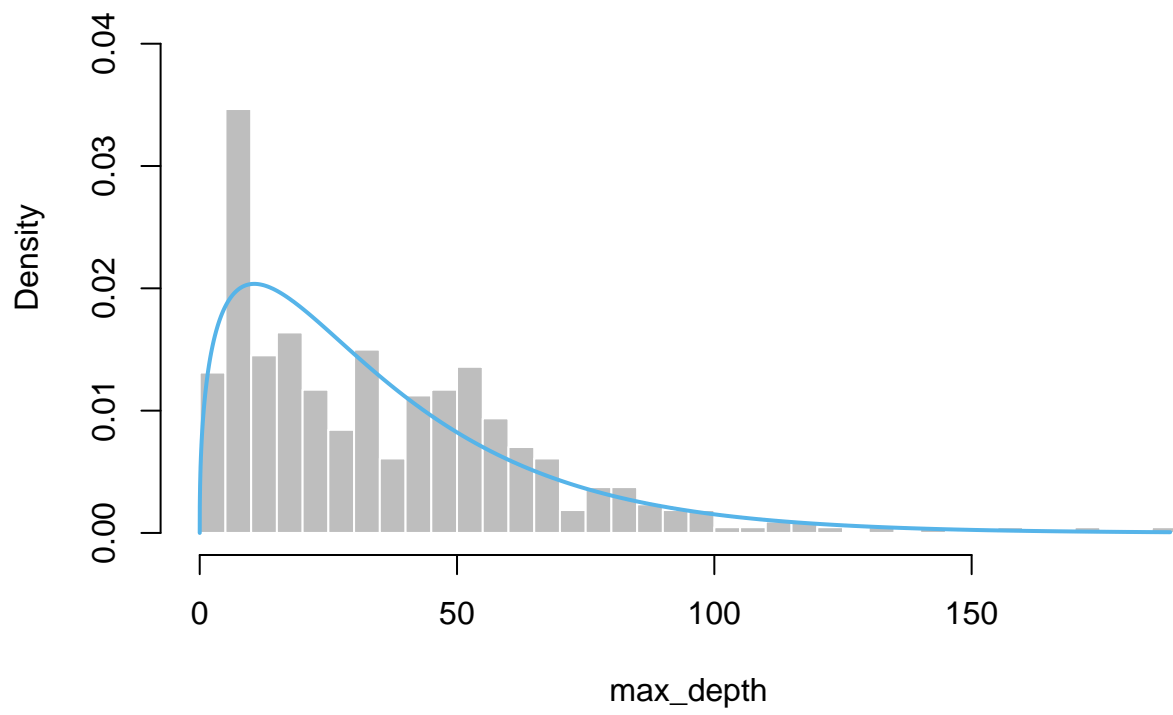
All animals – resting



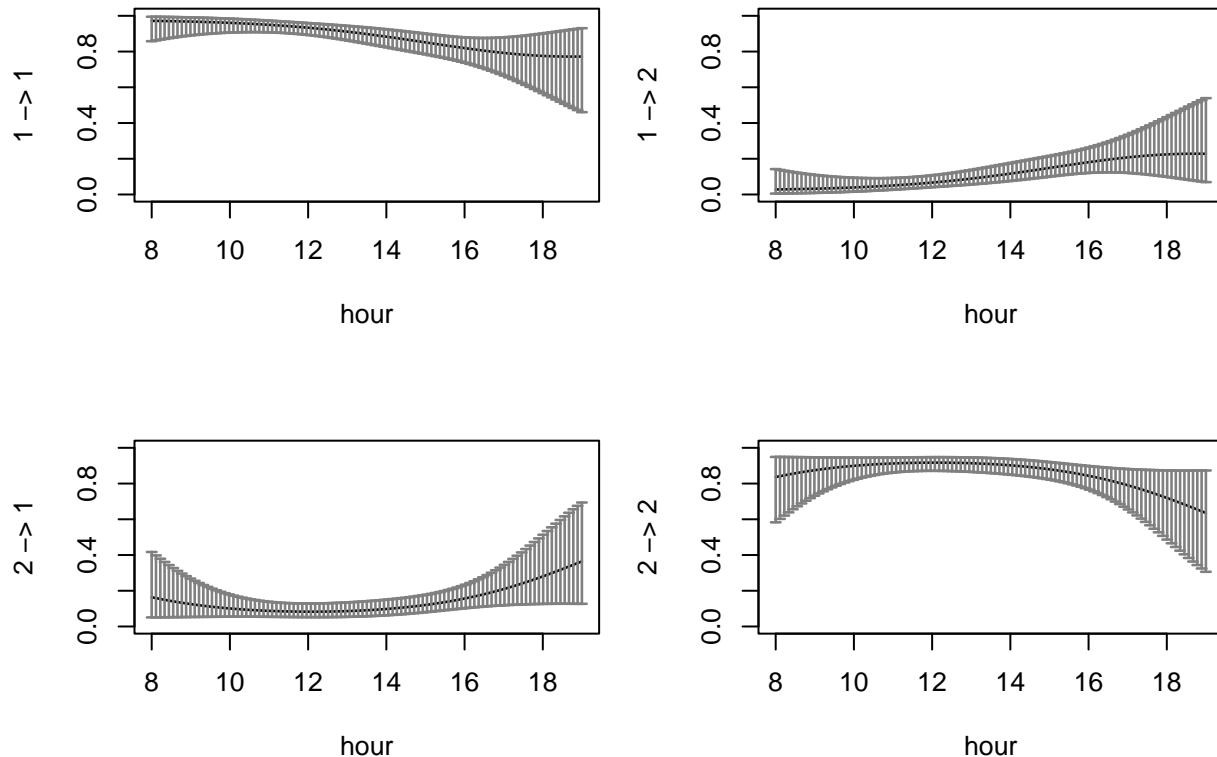




All animals – traveling



Transition probabilities



```
AIC(test_3vars_all, test_3vars_hr, k = 2, n = NULL)
```

```
##           Model      AIC
## 1 test_3vars_hr 6872.744
## 2 test_3vars_all 6881.525
```

```
AICweights(test_3vars_all, test_3vars_hr, k = 2, n = NULL)
```

```
##           Model      weight
## 1 test_3vars_hr 0.98775719
## 2 test_3vars_all 0.01224281
```

According to the AIC values, the model with the cosinor 24 hour cycle as a covariate for the transition probabilities is better.