Applying Gaussian Mixture Model

Georgina Shaw

07/03/2022

Herrin

Only keeping the information we need

```
df <- stom_df %>%
  transmute(Species = pred_species,
            wprey = prey_weight_g,
            wpredator = pred_weight_g,
            Nprey = prey_count / n_stomachs,
            l = log(wpredator / wprey))
```

Choosing the type of fish to use, this time looking at cod

```
stomach <- df %>%
  filter(Species == "Clupea harengus",
         wprey > 0)

stomach %>%
  group_by(Species) %>%
  summarise(wprey_min = min(wprey),
            wprey_max = max(wprey),
            lmin = min(l),
            lmax = max(l))
```

```
## # A tibble: 1 x 5
##   Species         wprey_min wprey_max   lmin  lmax
##   <chr>               <dbl>     <dbl>  <dbl> <dbl>
## 1 Clupea harengus   0.00001      104. 0.0544  17.5
```

```
stomach %>%
  group_by(Species) %>%
  filter(wprey == max(wprey))
```

```
## # A tibble: 1 x 5
## # Groups:   Species [1]
##   Species         wprey wpredator Nprey     l
##   <chr>           <dbl>     <dbl> <dbl> <dbl>
## 1 Clupea harengus  104.      116.  35.2 0.111
```

Creating bins for the data

```
no_bins <- 30  # Number of bins
binsize <- (max(stomach$l) - min(stomach$l)) / (no_bins - 1)
breaks <- seq(min(stomach$l) - binsize/2,
              by = binsize, length.out = no_bins + 1)
```

Splitting the data into the bins that have been made

```
binned_stomach <- stomach %>%
  # bin data
  mutate(cut = cut(l, breaks = breaks, right = FALSE,
                   labels = FALSE)) %>%
  group_by(Species, cut) %>%
  summarise(Numbers = sum(Nprey),
            Biomass = sum(Nprey * wprey)) %>%
  # normalise
  mutate(Numbers = Numbers / sum(Numbers) / binsize,
         Biomass = Biomass / sum(Biomass) / binsize)  %>%
  # column for predator/prey size ratio
  mutate(l = map_dbl(cut, function(idx) breaks[idx] + binsize/2))
```

```
## `summarise()` has grouped output by 'Species'. You can override using the `.groups` argument.
```

```
binned_stomach
```

```
## # A tibble: 30 x 5
## # Groups:   Species [1]
##    Species           cut Numbers Biomass      l
##    <chr>           <int>   <dbl>   <dbl>  <dbl>
##  1 Clupea harengus     1 0.00110  0.0351 0.0544
##  2 Clupea harengus     2 0.0197   0.348  0.655
##  3 Clupea harengus     3 0.0362   0.503  1.26
##  4 Clupea harengus     4 0.0264   0.162  1.86
##  5 Clupea harengus     5 0.0197   0.0546 2.46
##  6 Clupea harengus     6 0.0230   0.0575 3.06
##  7 Clupea harengus     7 0.0254   0.0717 3.66
##  8 Clupea harengus     8 0.0618   0.0663 4.26
##  9 Clupea harengus     9 0.115    0.113  4.86
## 10 Clupea harengus    10 0.145    0.0913 5.46
## # ... with 20 more rows
```

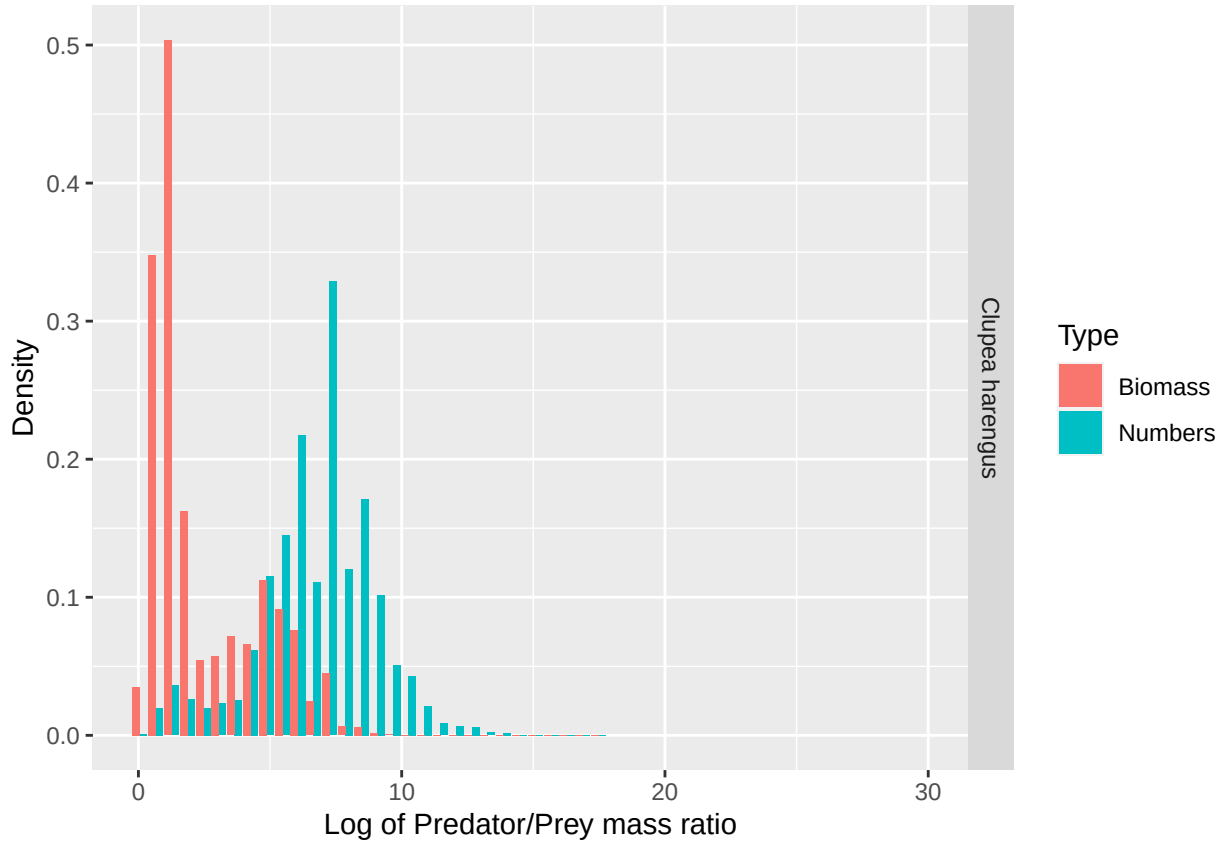
We convert this into the long table format preferred by ggplot2.

```
binned_stomach <- binned_stomach %>%
  gather(key = "Type", value = "Density", Numbers, Biomass)
```

Histograms

Plot the histogram that represents estimates of the normalised number density and the normalised biomass density

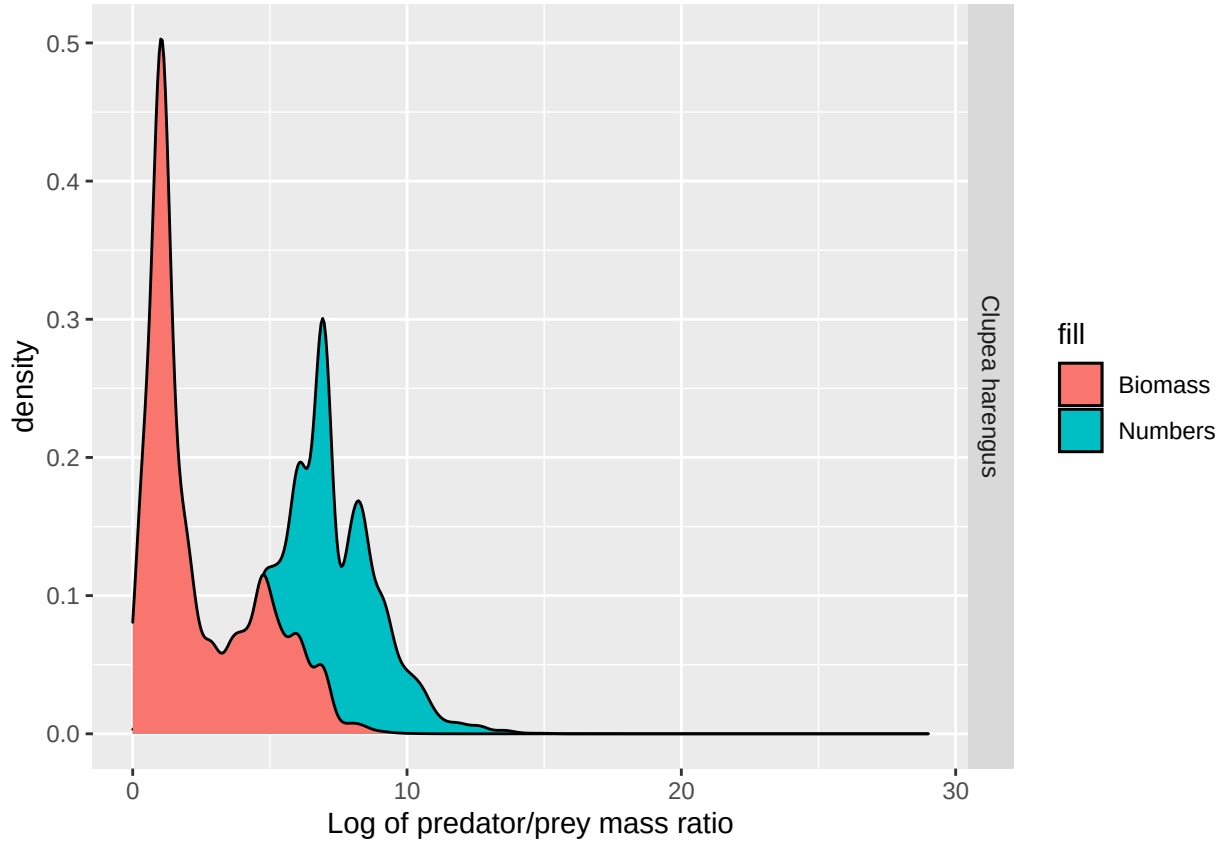
```
binned_stomach %>%
  ggplot(aes(l, Density, fill = Type)) +
  geom_col(position = "dodge") +
  facet_grid(rows = vars(Species), scales = "free_y") +
  xlab("Log of Predator/Prey mass ratio") +
  expand_limits(x = c(0, 30))
```



Kernal Density Estimation

```
adjust <- 1/2  # decrease bandwidth for kernel estimate
stomach <- stomach %>%
  group_by(Species) %>%
  mutate(weight_numbers = Nprey / sum(Nprey),
         weight_biomass = Nprey * wprey / sum(Nprey * wprey))
ggplot(stomach) +
  geom_density(aes(l, weight = weight_numbers,
                   fill = "Numbers"),
               adjust = adjust) +
  geom_density(aes(l, weight = weight_biomass,
                   fill = "Biomass"),
               adjust = adjust) +
  facet_grid(rows = vars(Species), scales = "free_y") +
```

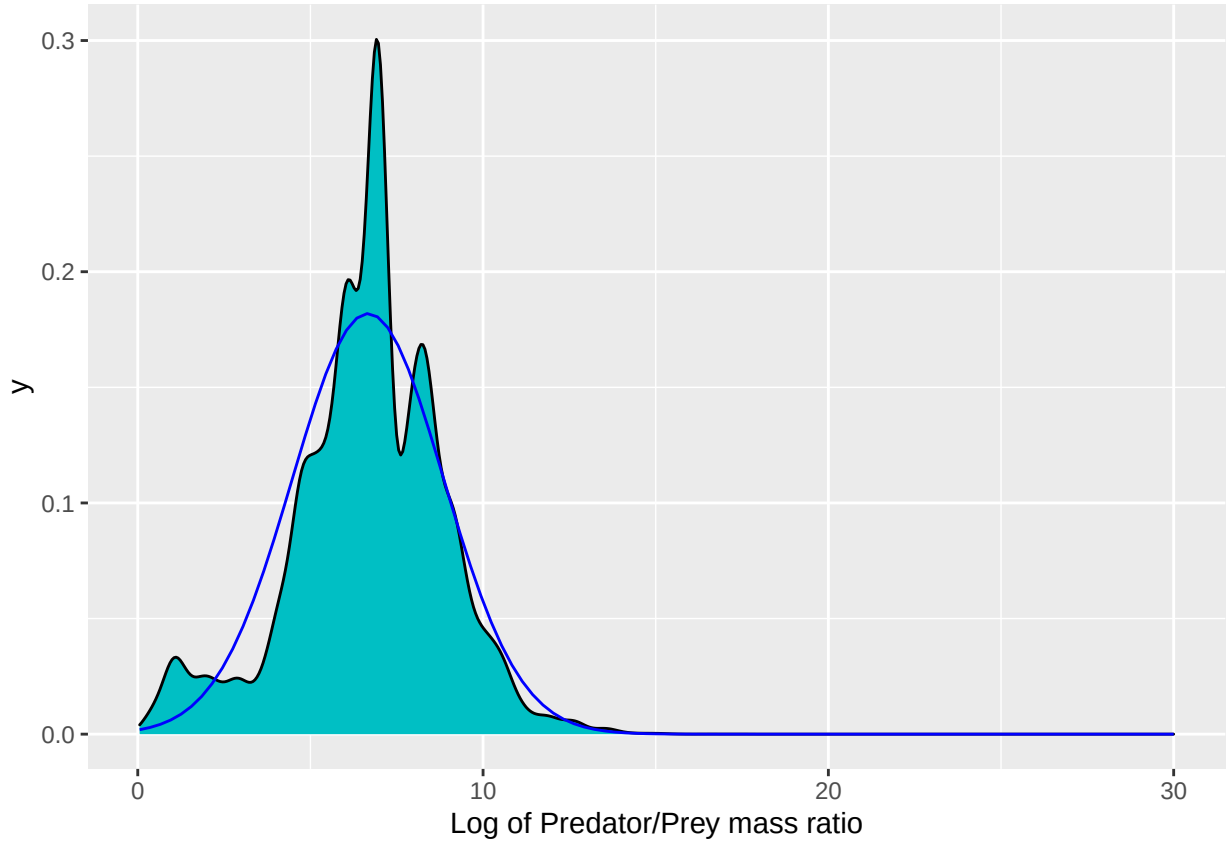
```
  xlab("Log of predator/prey mass ratio") +
  expand_limits(x = c(0, 29))
```



Gaussian Distribution fit

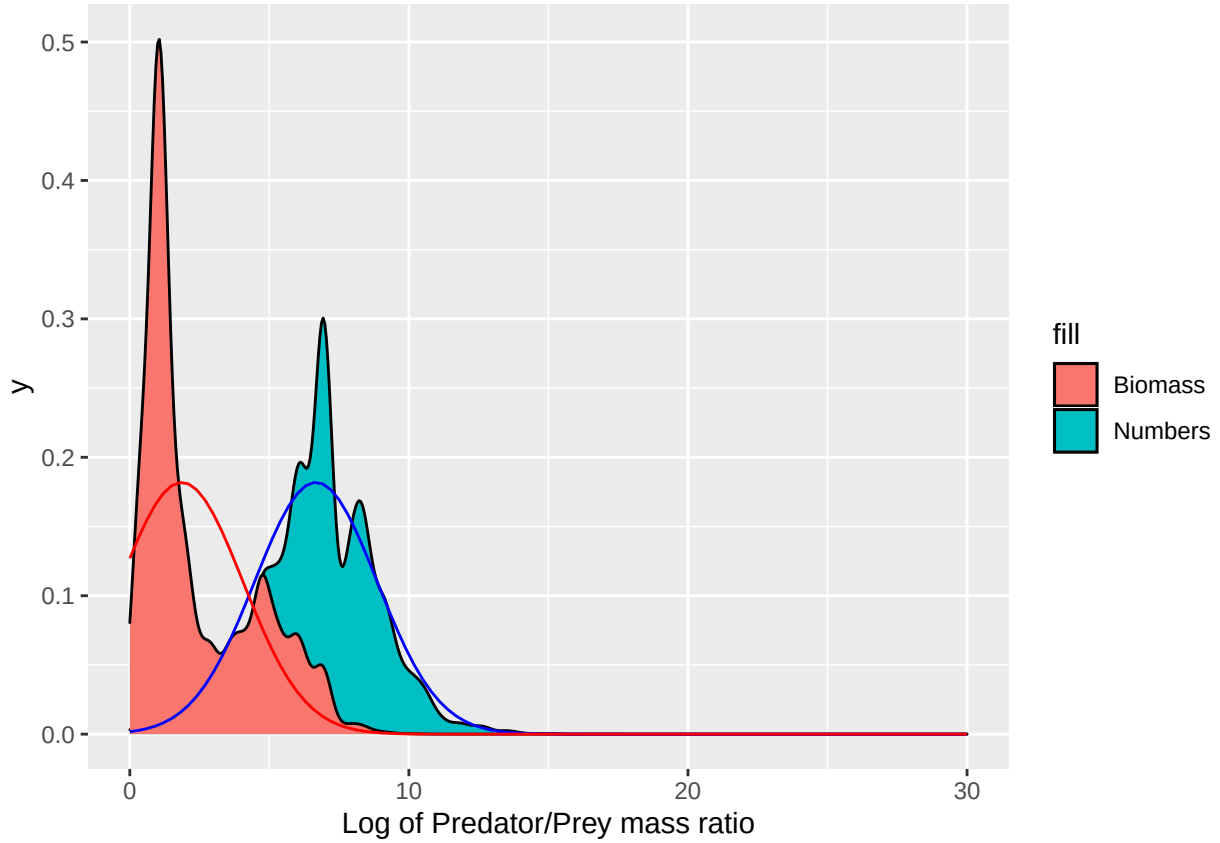
Plotting the normal distribution for numbers

```
weighted.sd <- function(x, w) {
  sqrt(sum(w * (x - weighted.mean(x, w))^2))
}
fit <- stomach %>%
  summarise(mean = weighted.mean(l, weight_numbers),
            sd = weighted.sd(l, weight_numbers))
stomach %>%
  ggplot() +
  geom_density(aes(l, weight = weight_numbers),
               fill = "#00BFC4", adjust = adjust) +
  xlab("Log of Predator/Prey mass ratio") +
  stat_function(fun = dnorm,
                args = list(mean = fit$mean,
                            sd = fit$sd),
                colour = "blue") +
  expand_limits(x = c(6, 30))
```

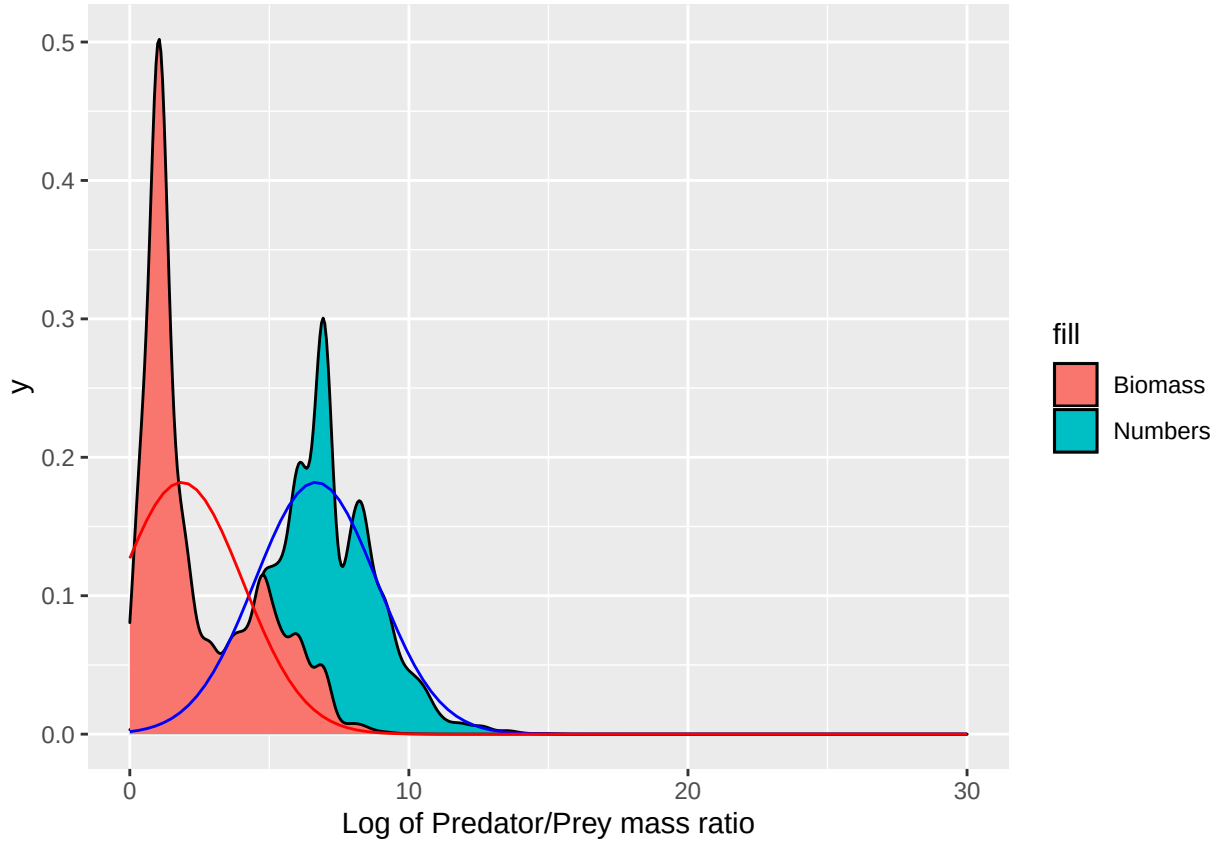


Now adding biomass

```
stomach %>%
  ggplot() +
  geom_density(aes(l, weight = weight_numbers,
                   fill = "Numbers"),
               adjust = adjust) +
  geom_density(aes(l, weight = weight_biomass,
                   fill = "Biomass"),
               adjust = adjust) +
  xlab("Log of Predator/Prey mass ratio") +
  stat_function(fun = dnorm,
                args = list(mean = fit$mean,
                            sd = fit$sd),
                colour = "blue") +
  stat_function(fun = dnorm,
                args = list(mean = fit$mean - fit$sd^2,
                            sd = fit$sd),
                colour = "red") +
  expand_limits(x = c(0, 30))
```




```
weighted.sd <- function(x, w) {
  sqrt(sum(w * (x - weighted.mean(x, w))^2))
}
fit <- stomach %>%
  summarise(mean = weighted.mean(l, weight_numbers),
            sd = weighted.sd(l, weight_numbers))
stomach %>%
  ggplot() +
  geom_density(aes(l, weight = weight_numbers,
                   fill = "Numbers"),
               adjust = adjust) +
  geom_density(aes(l, weight = weight_biomass,
                   fill = "Biomass"),
               adjust = adjust) +
  xlab("Log of Predator/Prey mass ratio") +
  stat_function(fun = dnorm,
                args = list(mean = fit$mean,
                            sd = fit$sd),
                colour = "blue") +
  stat_function(fun = dnorm,
                args = list(mean = fit$mean - fit$sd^2,
                            sd = fit$sd),
                colour = "red") +
  expand_limits(x = c(0, 30))
```



Gaussaian Mixture Model

The Gaussian Mixture model is the plotting of multiple Gaussian distributions on one plot. The estimates of the parameters are found using expectation maximization (EM), which consists of two steps, the expectation step (E step) and the maximization step (M step).

For this analysis only the the univariant case needs to be considered, with parameters μ_k and σ_k for each k-th component, with $k = 2$. The mixture component weightings are defined as ϕ_k with $\sum_{i=1}^{K} \phi_i = 1$, so the probabilities add to one.

The E step involves calculating the expectation of of the the assignment of each class for each data point given the model parameters ϕ_k, μ_k and σ_k. The M step involves maximization the expectations calculated in the E step, which updates the values ϕ_k, μ_k and σ_k. Eventually this should converge, giving a maximum likelihood estimate of the parameters.

If the Gaussian Mixture Model fitted the numbers density then is would be distributed by two Gaussians models such as;

$$n_1(p) \propto \exp\left(-\frac{(p-\mu_1)^2}{2\sigma_1^2}\right),$$

and

$$n_2(p) \propto \exp\left(-\frac{(p-\mu_2)^2}{2\sigma_2^2}\right).$$

Then the biomass density would be given by a two normal distributions but with the means shifted by σ^2, which would give the following;

$$b_1(p) \propto \exp\left(-\frac{(p-(\mu_1-\sigma_1^2)^2}{2\sigma_1^2}\right)$$

$$b_2(p) \propto \exp\left(-\frac{(p-(\mu_2-\sigma_2^2)^2}{2\sigma_2^2}\right)$$

Firstly, find the numbers and biomass PPMR for each observation using the definition of each;

$$r_i^{num} = \frac{1}{n}\sum_{j=1}^{n}\frac{M_i}{m_j}$$

$$r_i^{bio} = \frac{M_i}{\frac{1}{n}\sum_{j=1}^{n} m_j}$$

Applying Gaussian Mixture Models in different ways First way by calculating each individual PPMR by numbers and biomass and then applying the EM algorithm to find the estimated parameters and then plotting

```
ppmr <- stomach %>%
  mutate(numbers = log(wpredator/(wprey * sum(Nprey))),
         biomass = log(wpredator*sum(Nprey)/wprey))
library(mixtools)
```

```
## mixtools package, version 1.2.0, Released 2020-02-05
## This package is based upon work supported by the National Science Foundation under Grant No. SES-0518
```

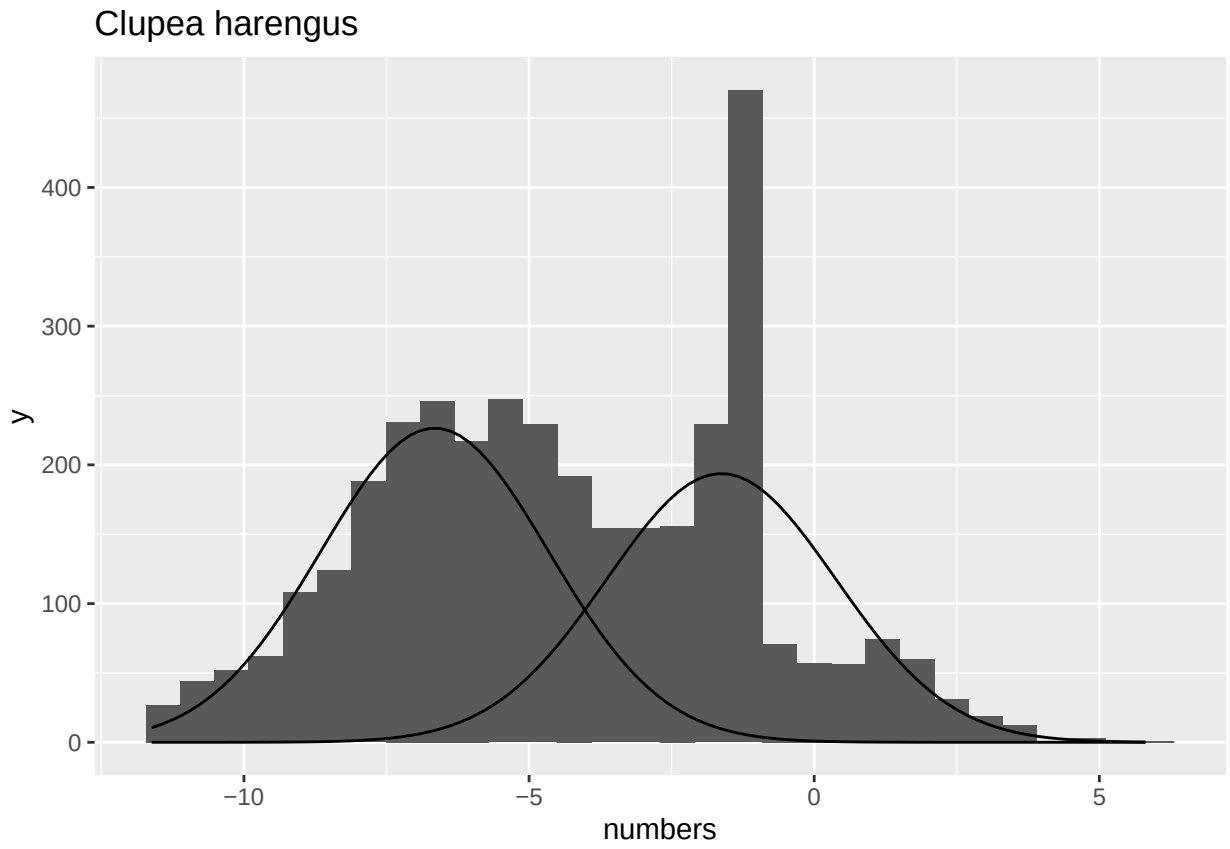
```
my_mix1 <- normalmixEM(ppmr$numbers, k = 2) #applied the EM algorithm finding the estimates of the para
```

```
## number of iterations= 196
```

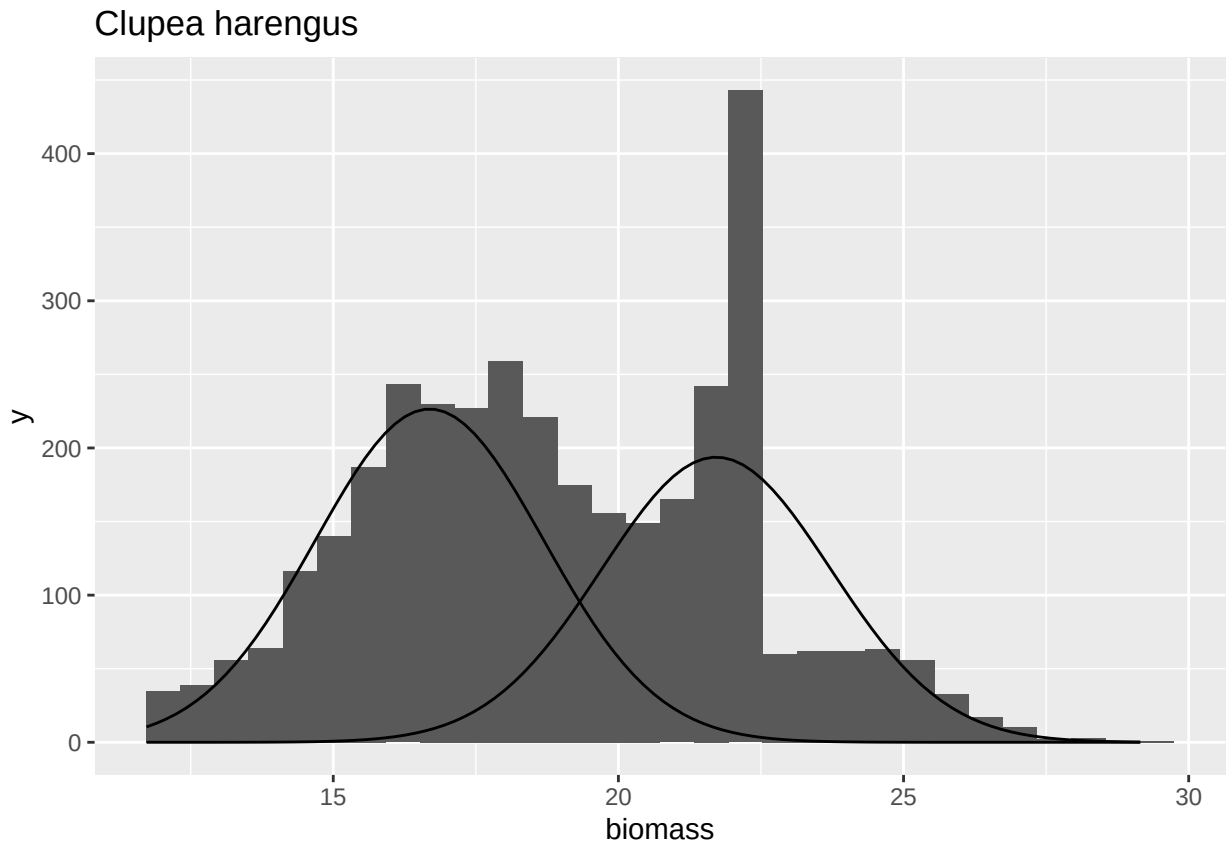
```
my_mix2 <- normalmixEM(ppmr$biomass, k = 2)
```

```
## number of iterations= 172
```

```
#plot of numbers
ggplot(ppmr, aes(x = numbers)) +
  geom_histogram(binwidth = binsize) +
  ggtitle("Clupea harengus") +
  mapply(
    function(mean, sd, lambda, n, binwidth) {
      stat_function(
        fun = function(x) {
          (dnorm(x, mean = mean, sd = sd)) * n * binwidth * lambda
        }
      )
    },
    mean = my_mix1[["mu"]], #mean
    sd = my_mix1[["sigma"]], #standard deviation
    lambda = my_mix1[["lambda"]], #amplitude
    n = length(ppmr$numbers), #sample size
    binwidth = binsize #binwidth used for histogram
  )
```




```
#plot of biomass
ggplot(ppmr, aes(x = biomass), main ="Clupea harengus") +
  geom_histogram(binwidth = binsize) +
  ggtitle("Clupea harengus") +
  mapply(
    function(mean, sd, lambda, n, binwidth) {
      stat_function(
        fun = function(x) {
          (dnorm(x, mean = mean, sd = sd)) * n * binwidth * lambda
        }
      )
    },
    mean = my_mix2[["mu"]], #mean
    sd = my_mix2[["sigma"]], #standard deviation
    lambda = my_mix2[["lambda"]], #amplitude
    n = length(ppmr$biomass), #sample size
    binwidth = binsize #binwidth used for histogram
  )
```



Looking at Gaussian distribution for different species

```
stomach_all <- stom_df %>%
  select(Species = pred_species,
            wprey = prey_weight_g,
            wpredator = pred_weight_g) %>%
  group_by(Species) %>%
  filter(n() > 1000, wprey > 0) %>%
  mutate(Nprey = 1,
         l = log(wpredator / wprey),
         weight_numbers = Nprey / sum(Nprey),
         weight_biomass = Nprey * wprey / sum(Nprey * wprey))
unique(stomach_all$Species)
```

```
##  [1] "Clupea harengus"            "Scomber scombrus"
##  [3] "Limanda limanda"            "Pleuronectes platessa"
##  [5] "Gadus morhua"               "Merluccius merluccius"
##  [7] "Merlangius merlangus"       "Melanogrammus aeglefinus"
##  [9] "Eutrigla gurnardus"         "Trachurus trachurus"
## [11] "Raja clavata"               "Scophthalmus maximus"
## [13] "Amblyraja radiata"          "Pollachius virens"
## [15] "Lepidorhombus whiffiagonis"
```

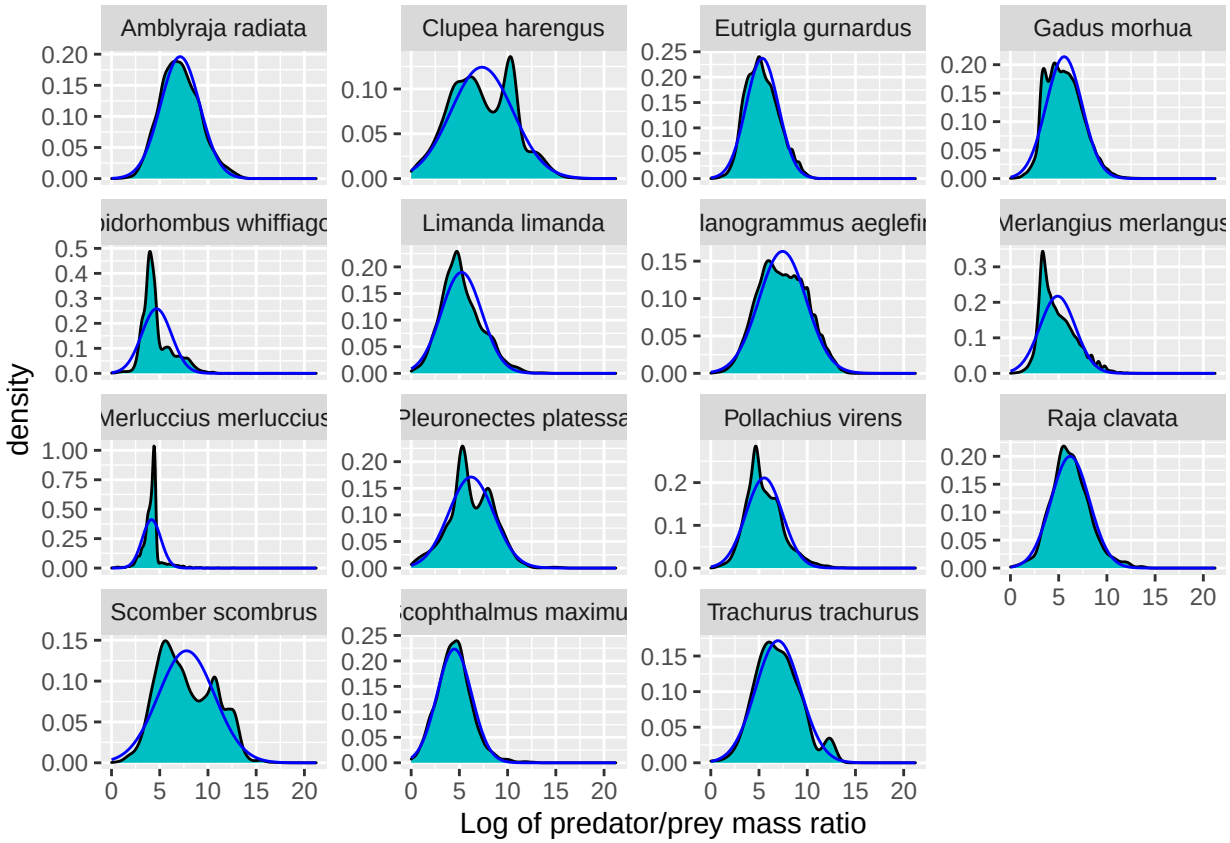
```
no_bins <- 30  # Number of bins
binsize <- (max(stomach_all$l) - min(stomach_all$l)) / (no_bins - 1)
breaks <- seq(min(stomach_all$l) - binsize/2,
```

```
              by = binsize, length.out = no_bins + 1)

binned_stomach <- stomach_all %>%
  # bin data
  mutate(cut = cut(l, breaks = breaks, right = FALSE,
                   labels = FALSE)) %>%
  group_by(Species, cut) %>%
  summarise(Numbers = sum(Nprey),
            Biomass = sum(Nprey * wprey)) %>%
  # normalise
  mutate(Numbers = Numbers / sum(Numbers) / binsize,
         Biomass = Biomass / sum(Biomass) / binsize)  %>%
  # column for predator/prey size ratio
  mutate(l = map_dbl(cut, function(idx) breaks[idx] + binsize/2)) %>%
  gather(key = "Type", value = "Density", Numbers, Biomass)
```

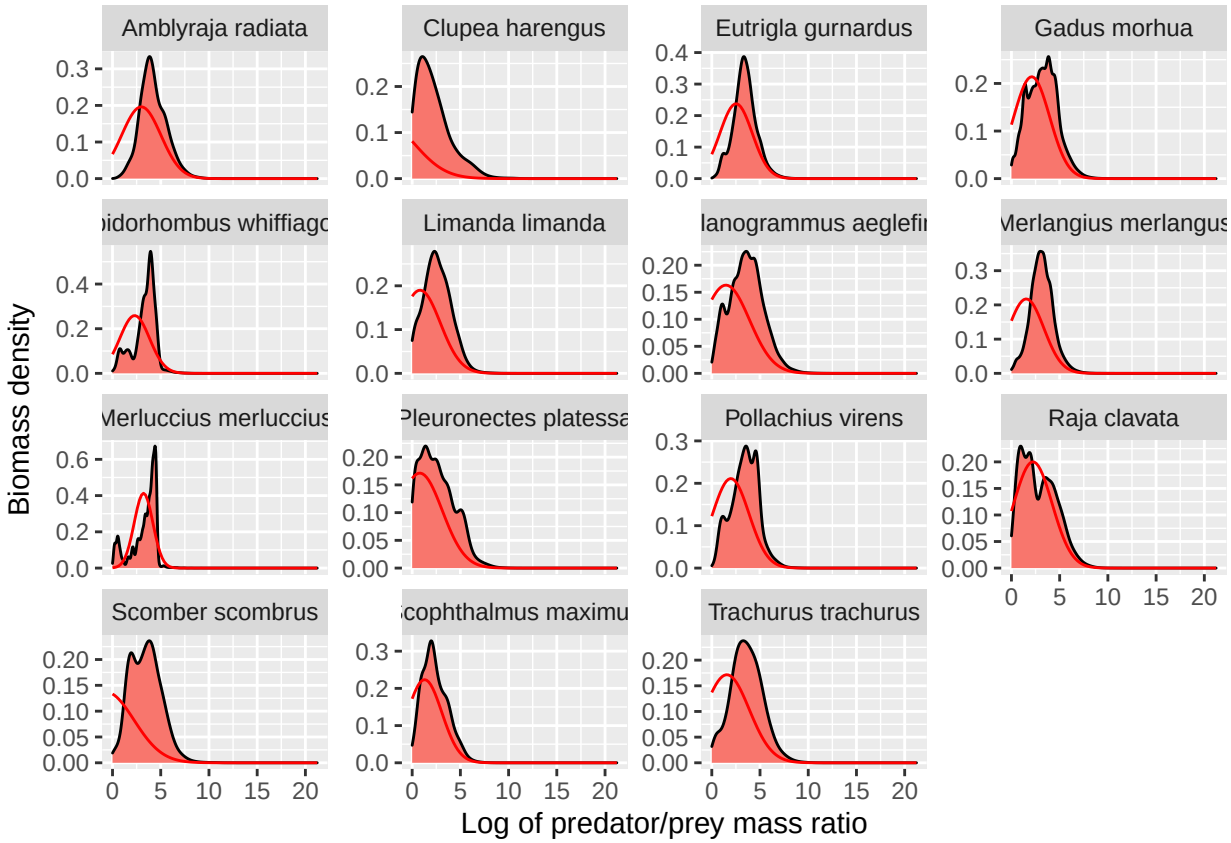
```
## `summarise()` has grouped output by 'Species'. You can override using the `.groups` argument.
```

Plotting the graphs

```
grid <- seq(0, max(stomach_all$l), length = 100)
normaldens <- plyr::ddply(stomach_all, "Species", function(df) {
  data.frame(
    l = grid,
    density = dnorm(grid, mean(df$l), sd(df$l))
  )
})

ggplot(stomach_all) +
  geom_density(aes(l, weight = weight_numbers), fill = "#00BFC4") +
  facet_wrap(~Species, scales = "free_y", ncol = 4) +
  xlab("Log of predator/prey mass ratio")  +
  geom_line(aes(l, density), data = normaldens,
            colour = "blue")
```



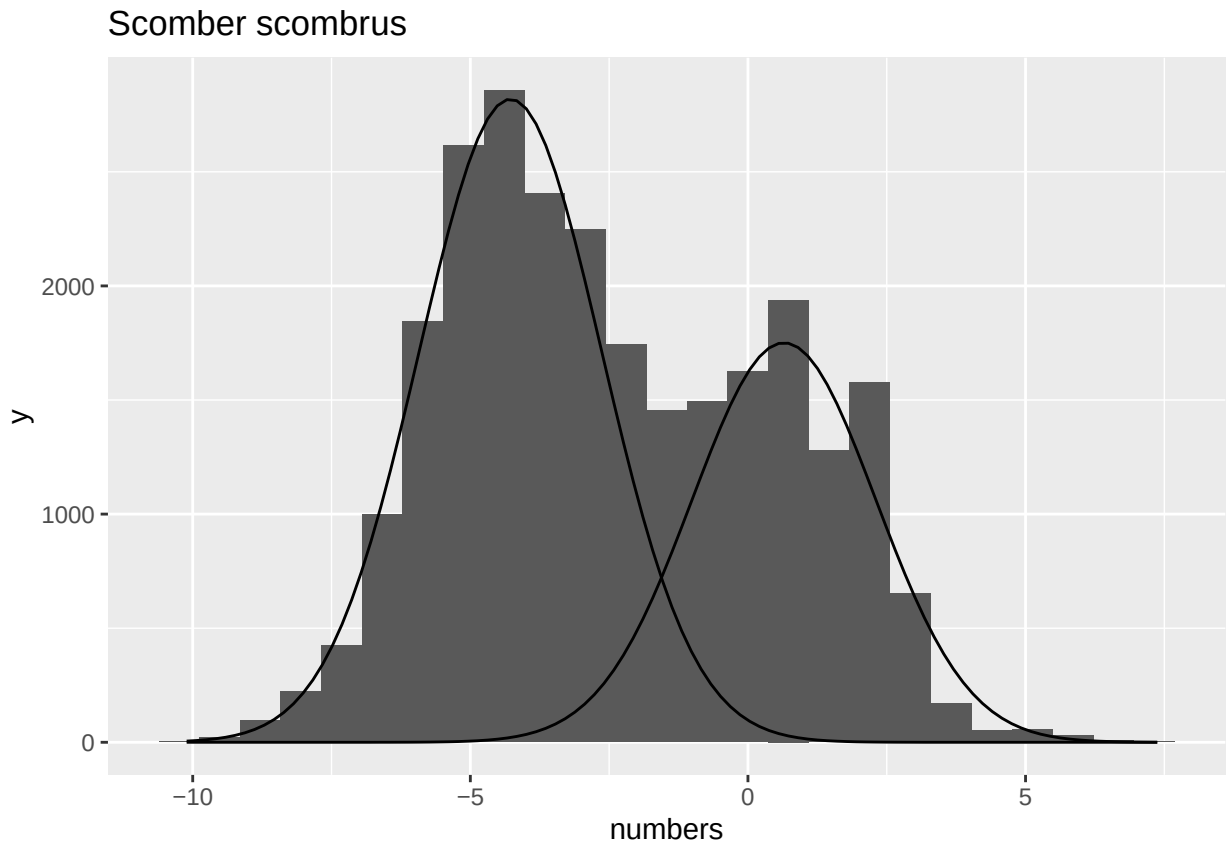

```
grid <- seq(0, max(stomach_all$l), length = 100)
shifted_normaldens <- plyr::ddply(stomach_all, "Species", function(df) {
  data.frame(
    l = grid,
    density = dnorm(grid, mean(df$l) - sd(df$l)^2, sd(df$l))
  )
})

ggplot(stomach_all) +
  geom_density(aes(l, weight = weight_biomass), fill = "#F8766D") +
  facet_wrap(~Species, scales = "free_y", ncol = 4) +
  xlab("Log of predator/prey mass ratio") +
  ylab("Biomass density") +
  geom_line(aes(l, density), data = shifted_normaldens,
            colour = "red")
```

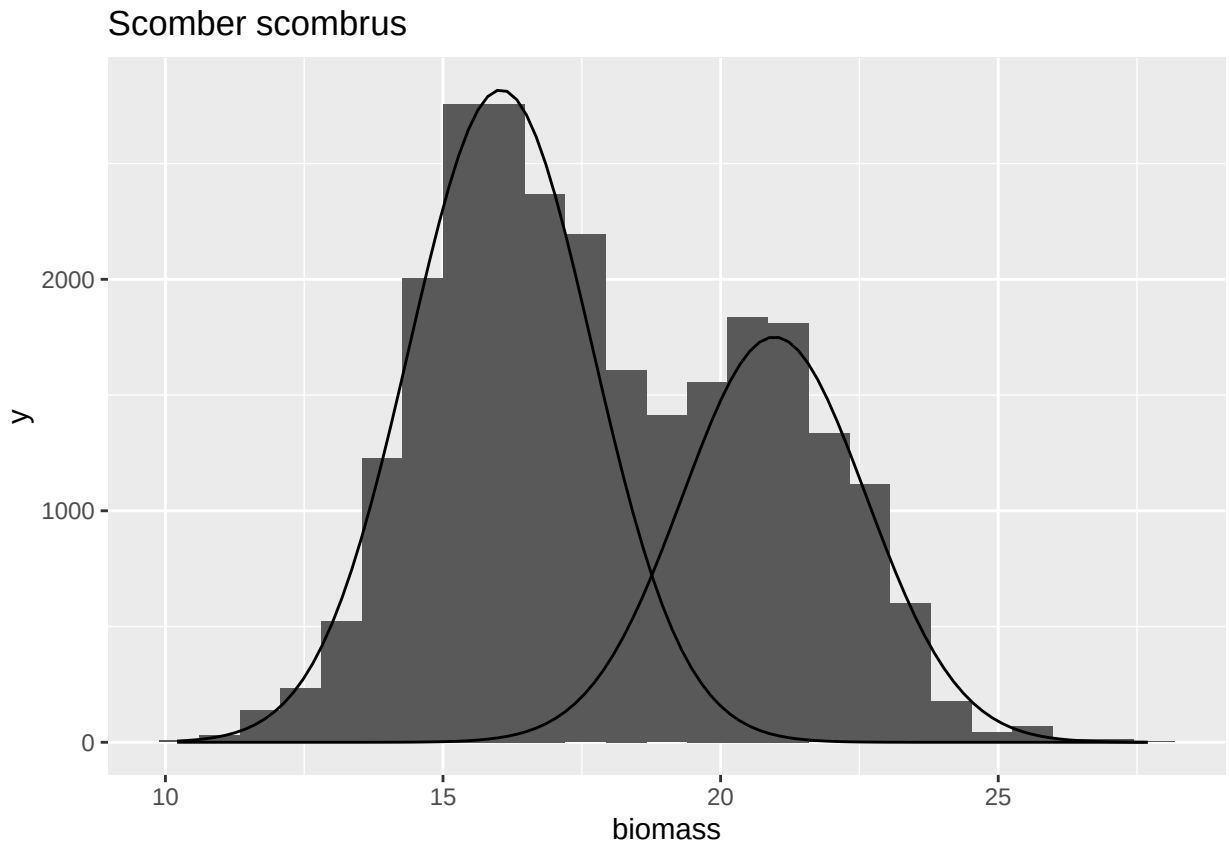


Scomber scombrus

```
## number of iterations= 135
```

```
## number of iterations= 131
```

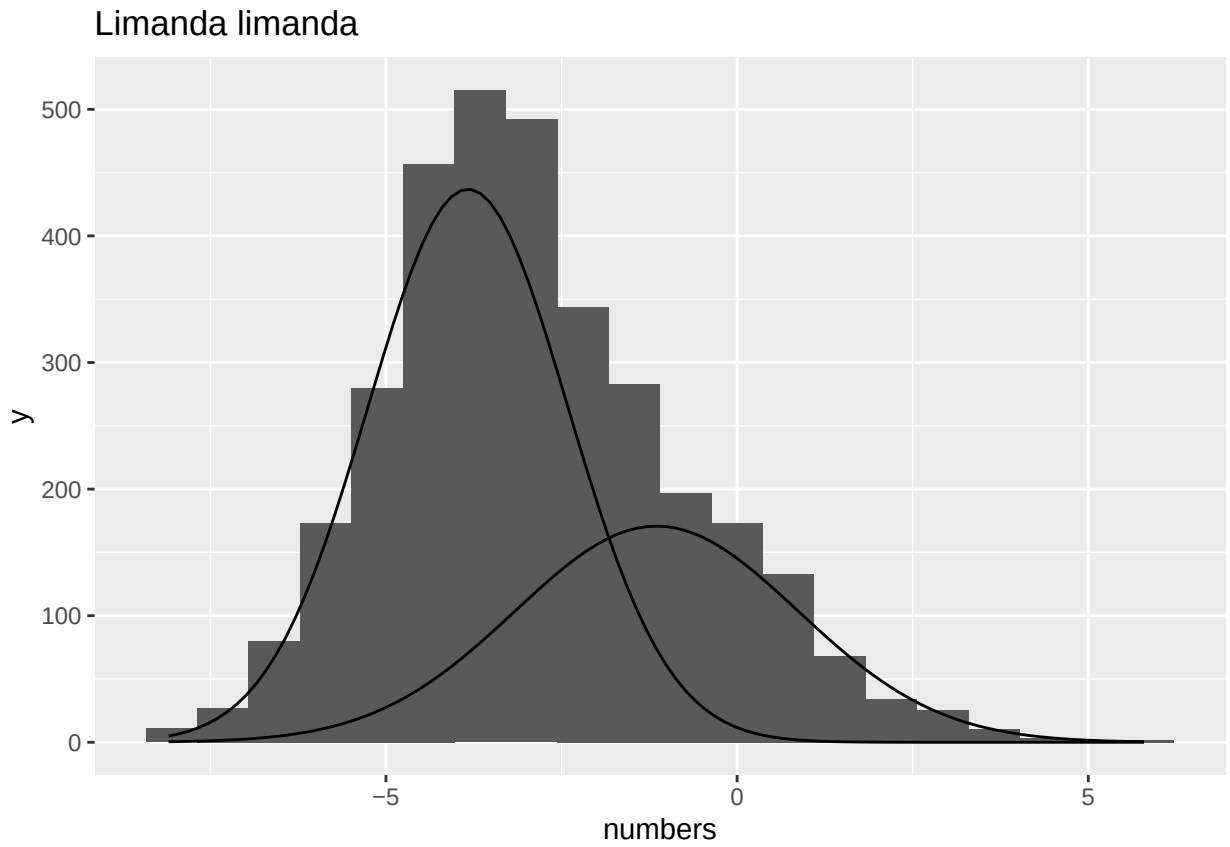

Scomber scombrus
2000
1000
0
y
−10
−5
0
5
numbers

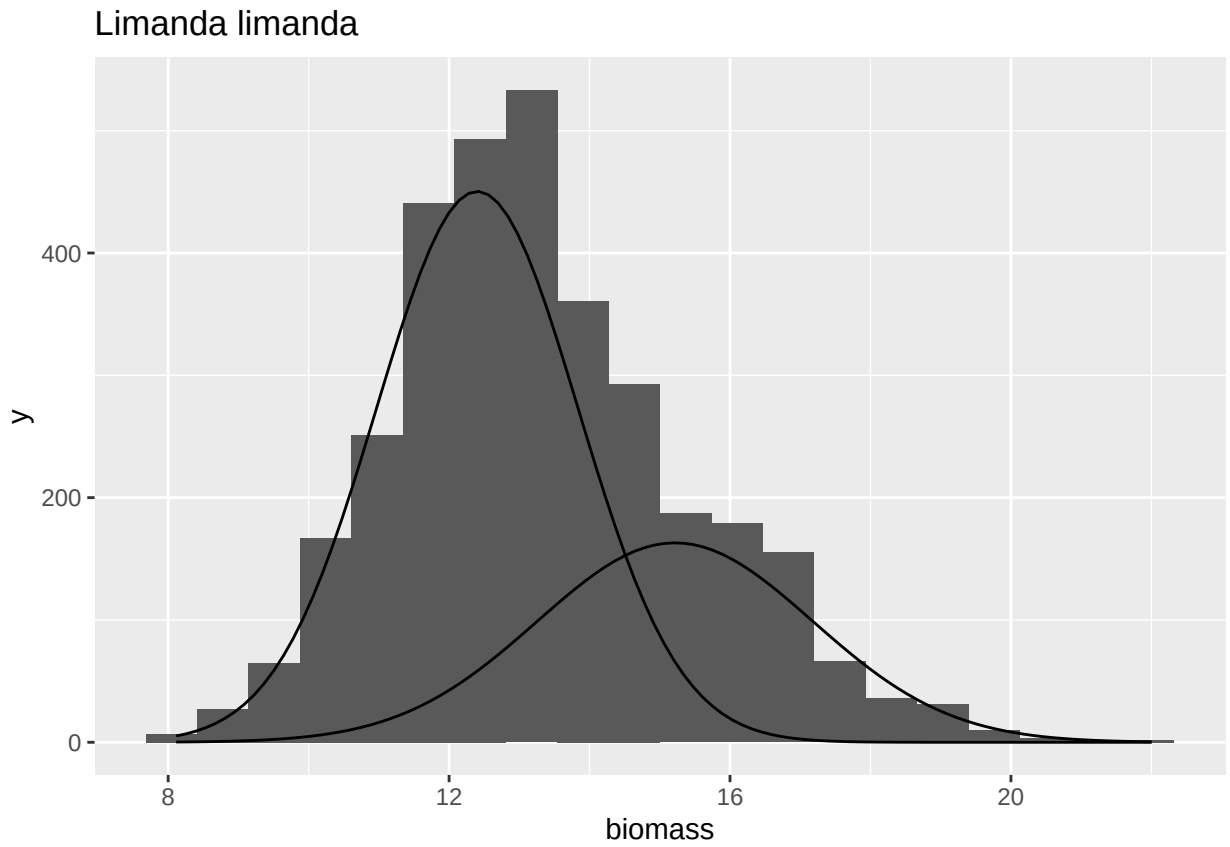


Limanda limanda

```
## WARNING! NOT CONVERGENT!
## number of iterations= 1000
```

```
## WARNING! NOT CONVERGENT!
## number of iterations= 1000
```

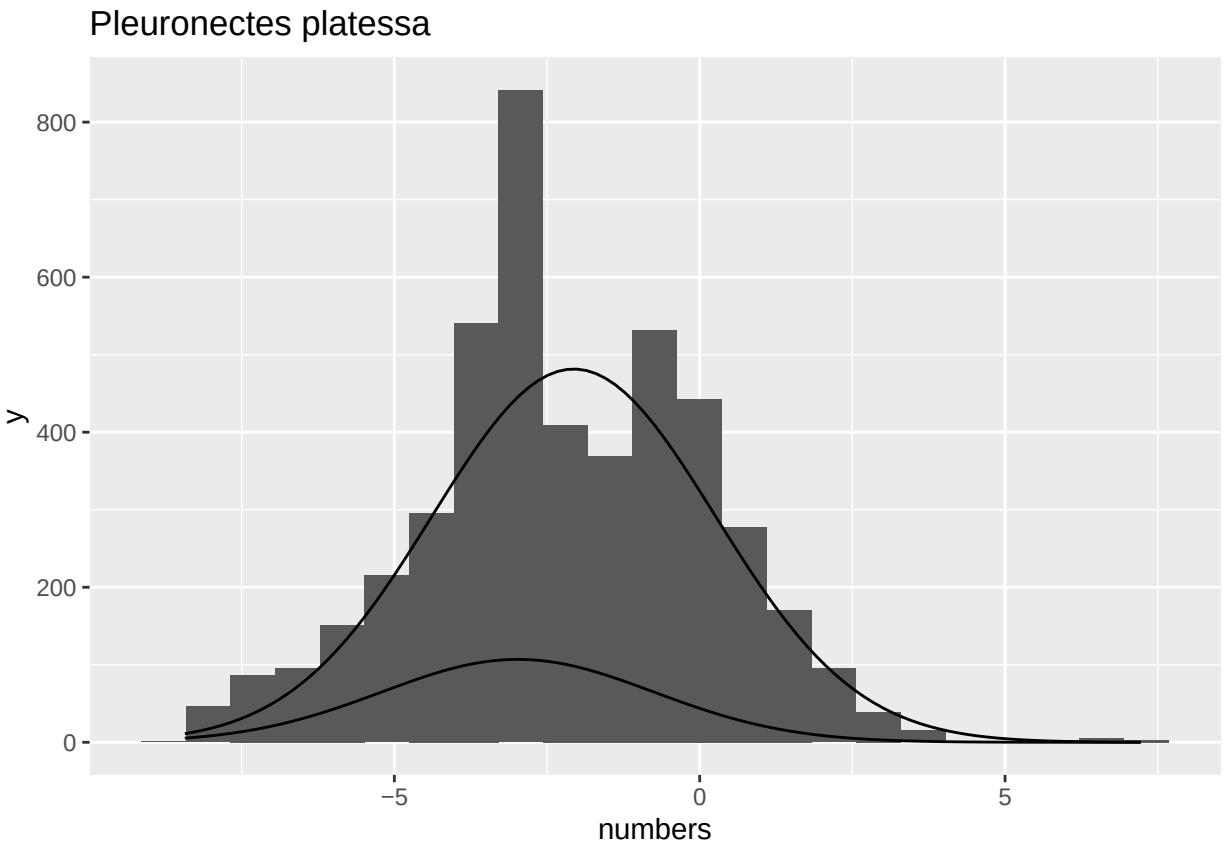

Limanda limanda
500
400
300
200
100
0
y
−5
0
5
numbers

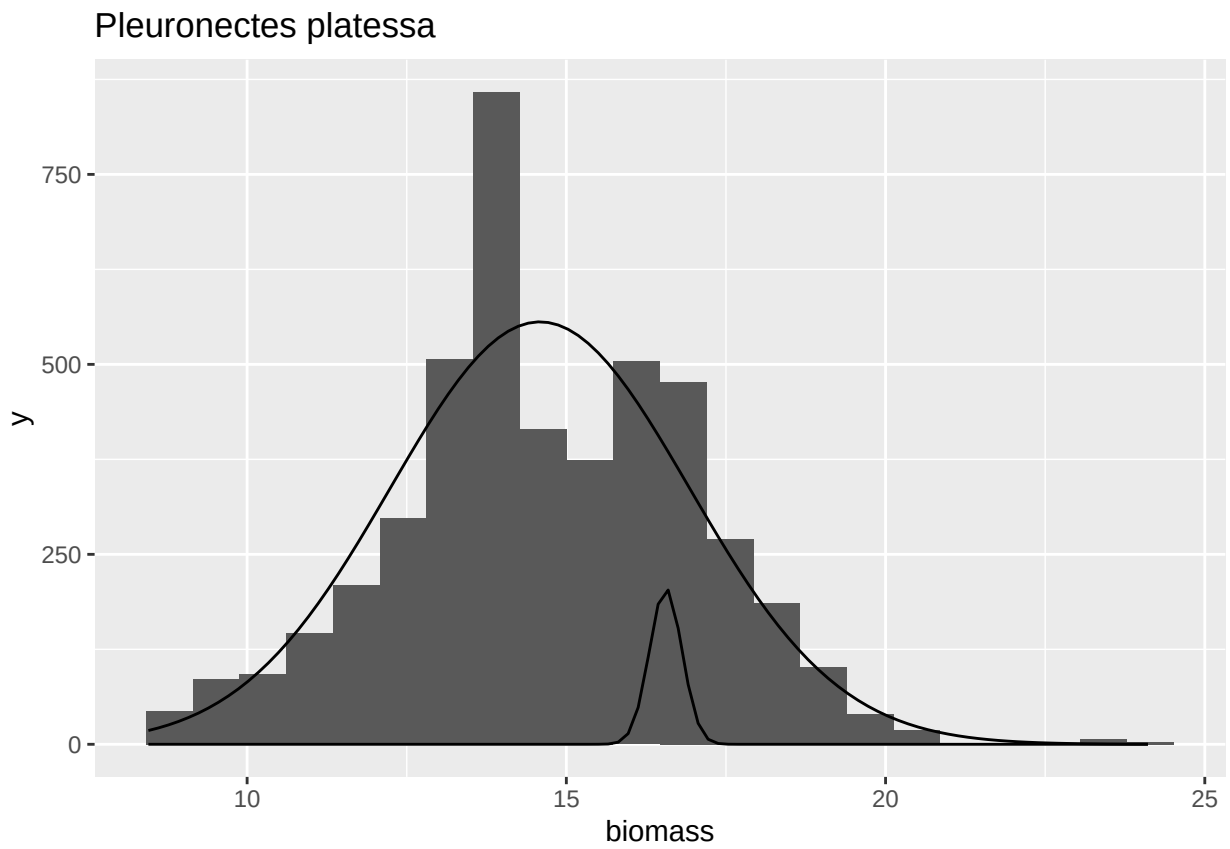


Pleuronectes platessa

```
## WARNING! NOT CONVERGENT!
## number of iterations= 1000
```

```
## number of iterations= 227
```

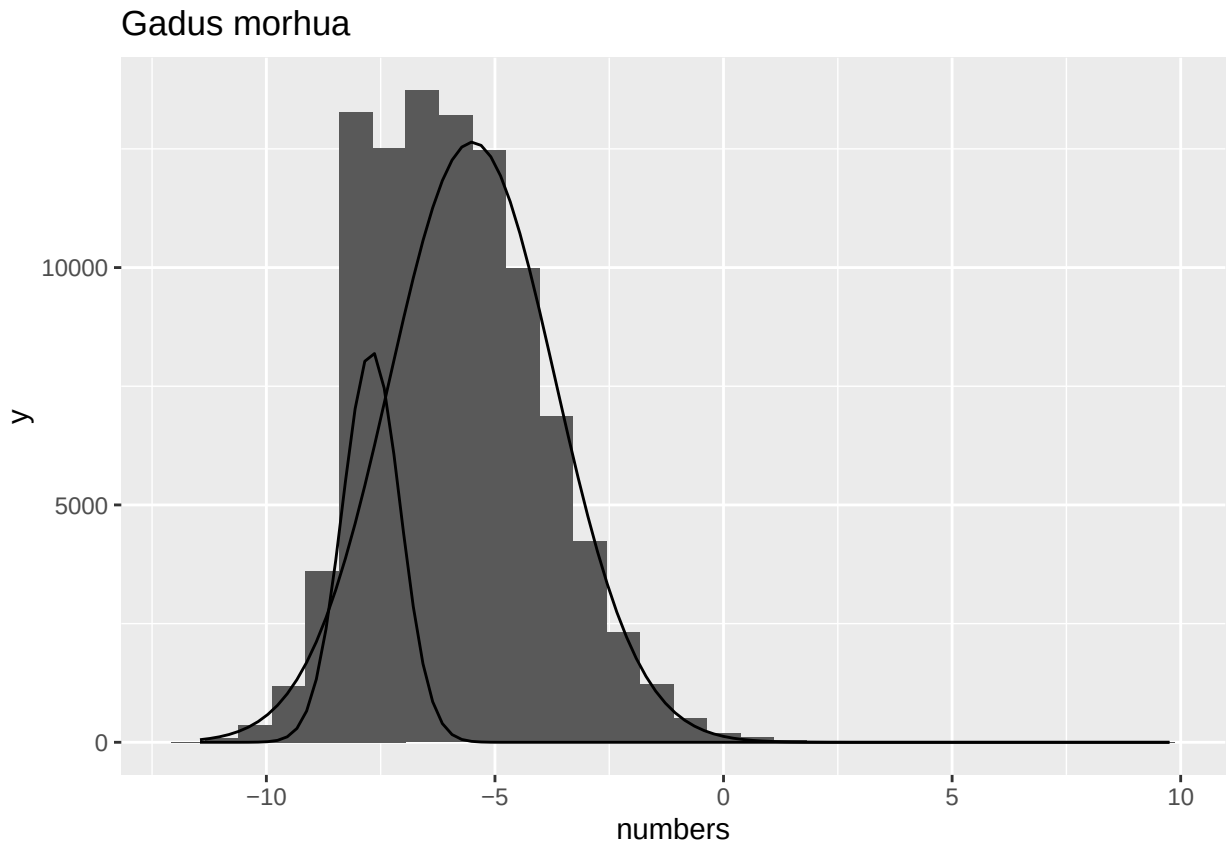

Pleuronectes platessa
800
600
400
200
0
y
−5
0
5
numbers

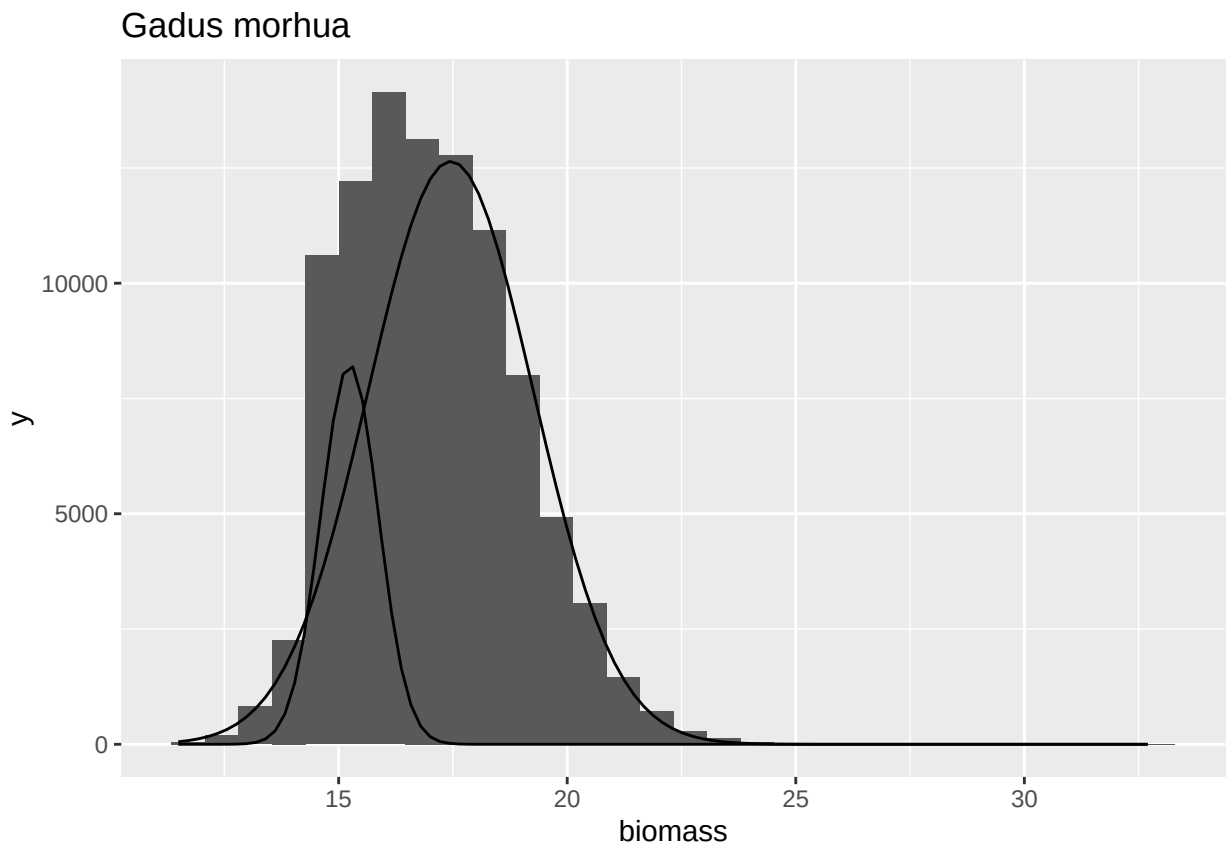


Gadus morhua

```
## number of iterations= 446
```

```
## number of iterations= 352
```

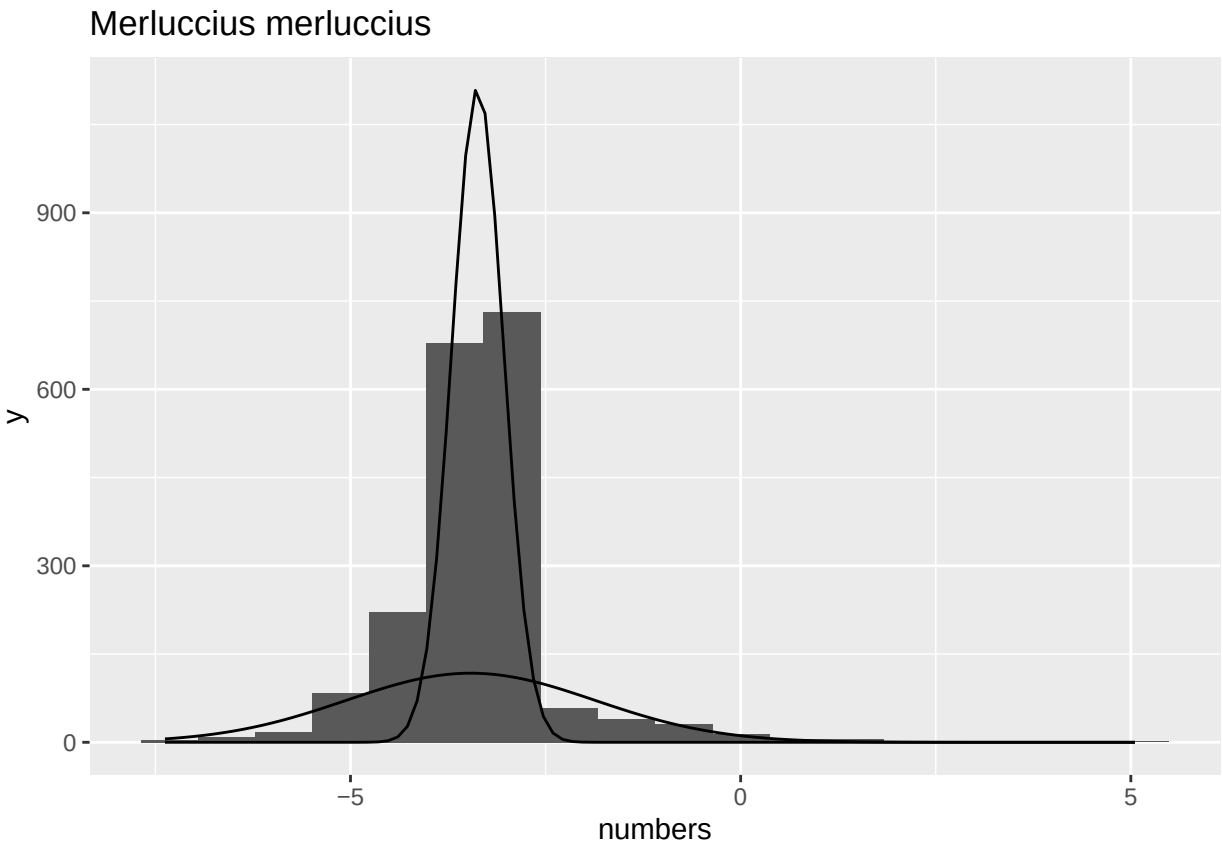

Gadus morhua
10000
5000
0
y
−10
−5
0
5
10
numbers

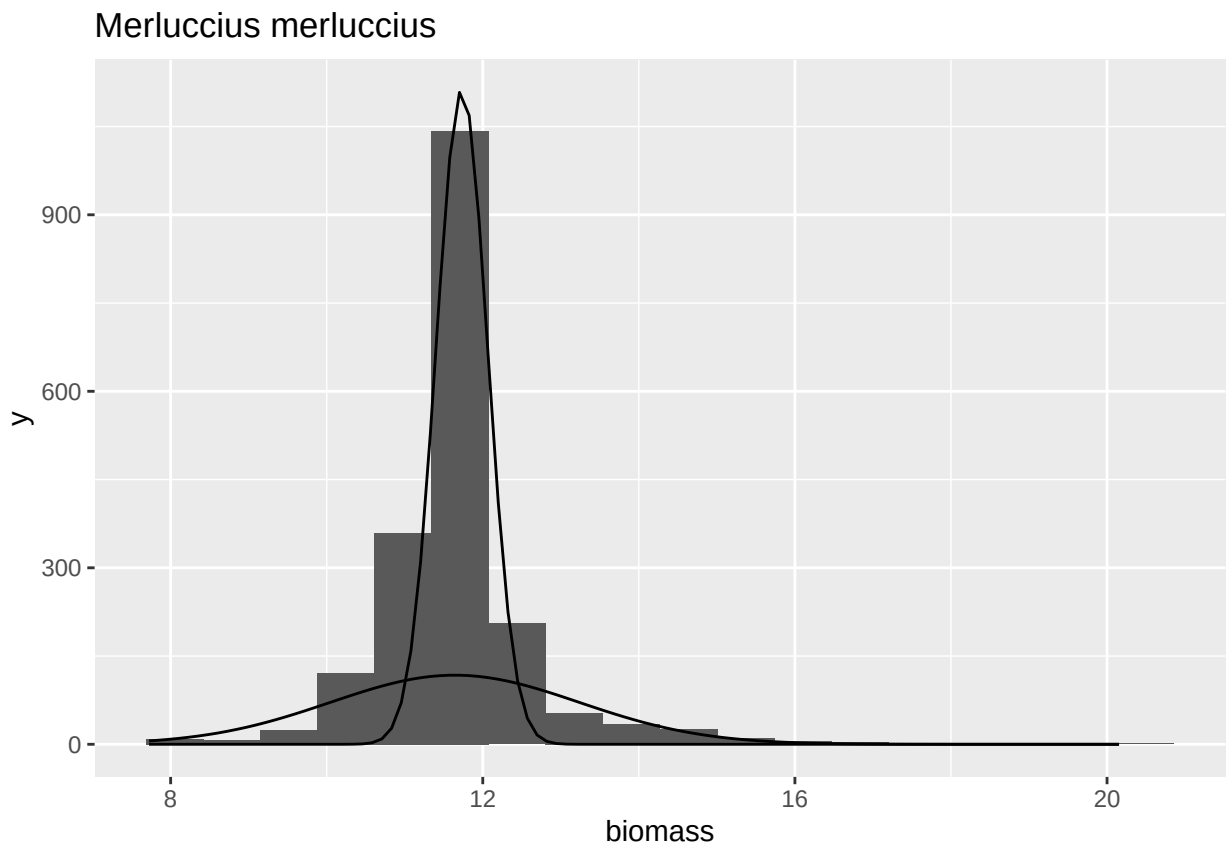


Merluccius merluccius

```
## number of iterations= 74
```

```
## number of iterations= 75
```

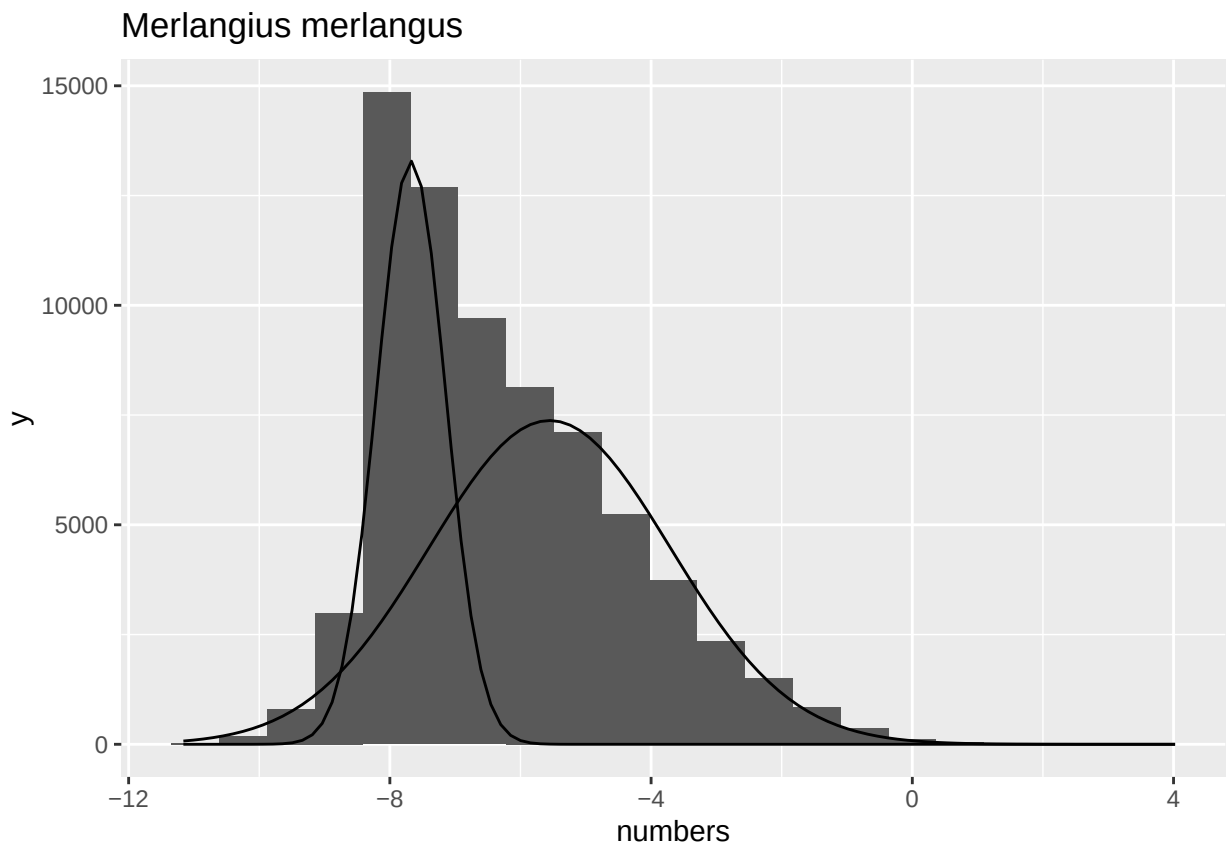

Merluccius merluccius
900
600
300
0
y
−5
0
5
numbers

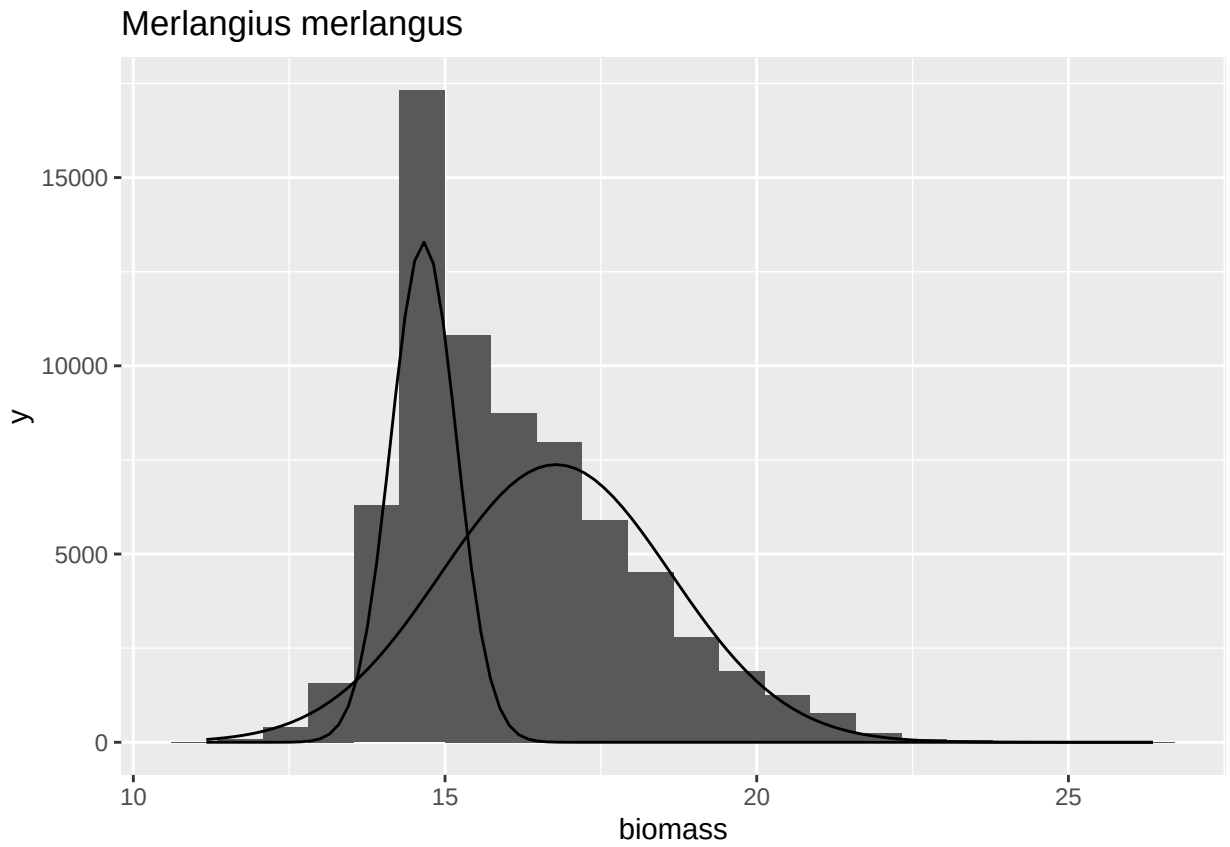


Merlangius merlangus

```
## number of iterations= 135
```

```
## number of iterations= 166
```

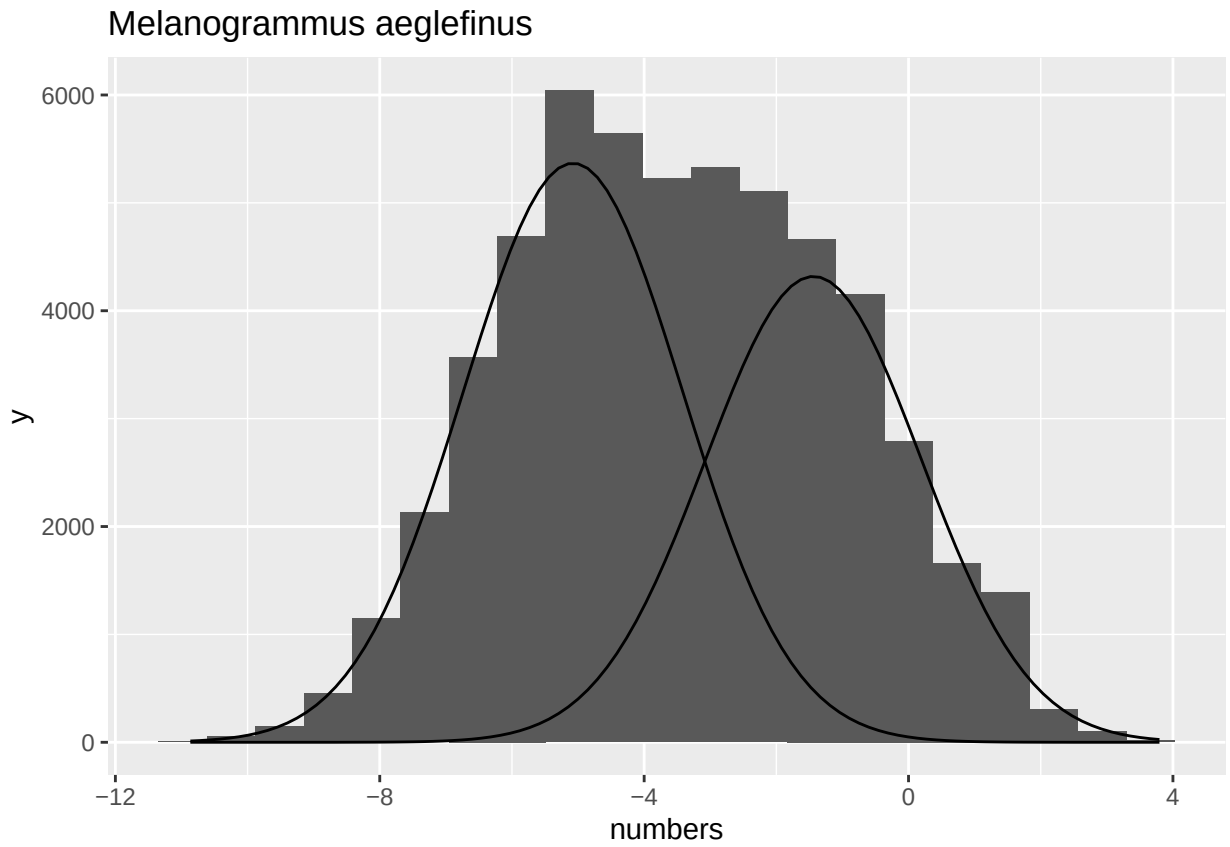

Merlangius merlangus
15000
10000
5000
0
y
−12
−8
−4
0
4
numbers

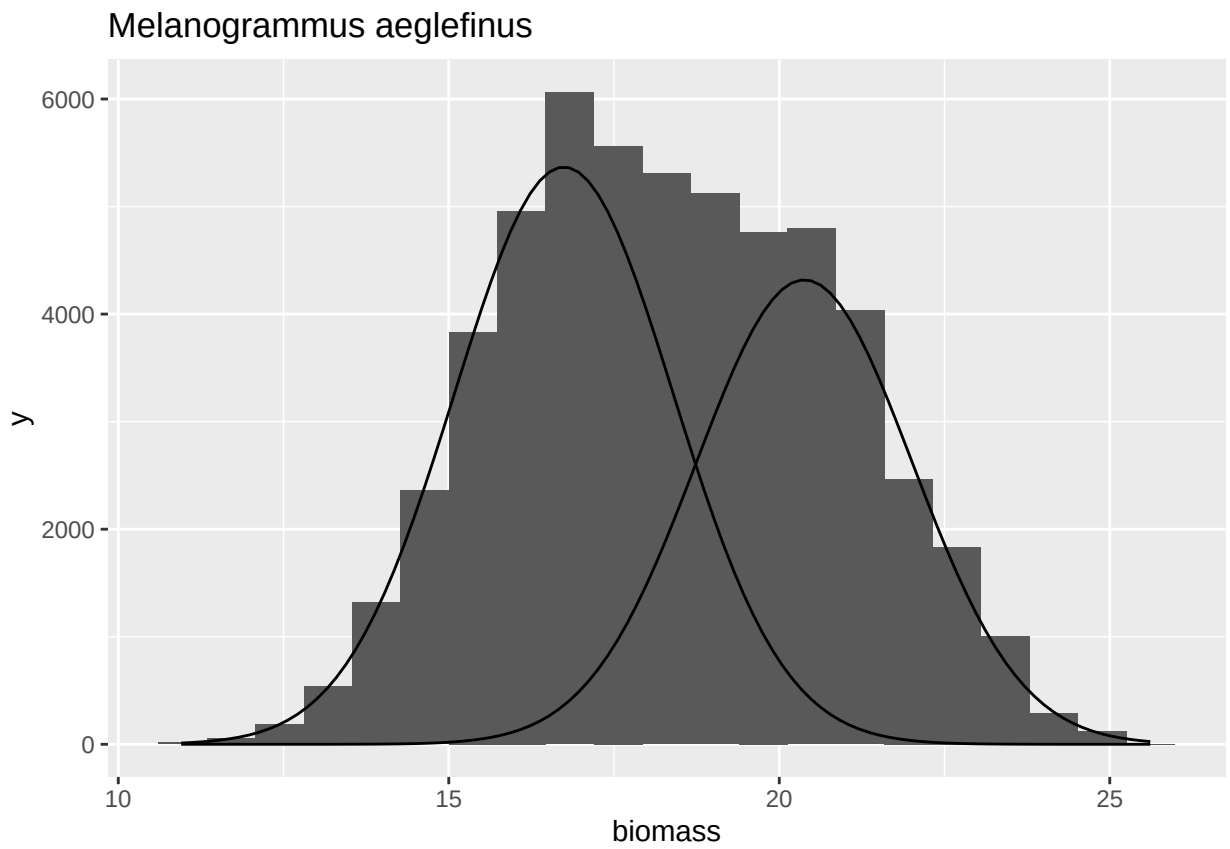


Melanogrammus aeglefinus

```
## number of iterations= 705
```

```
## number of iterations= 746
```

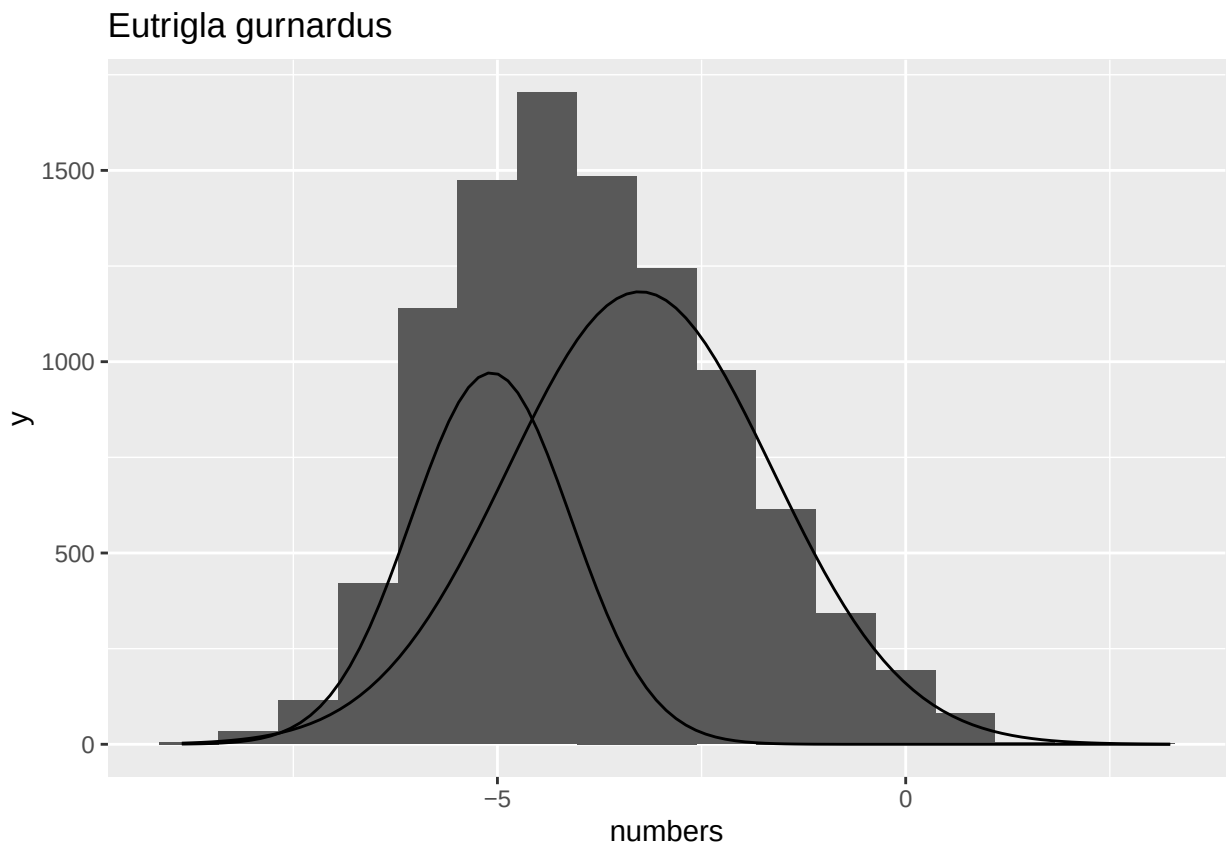

Melanogrammus aeglefinus
6000
4000
2000
0
y
−12
−8
−4
0
4
numbers

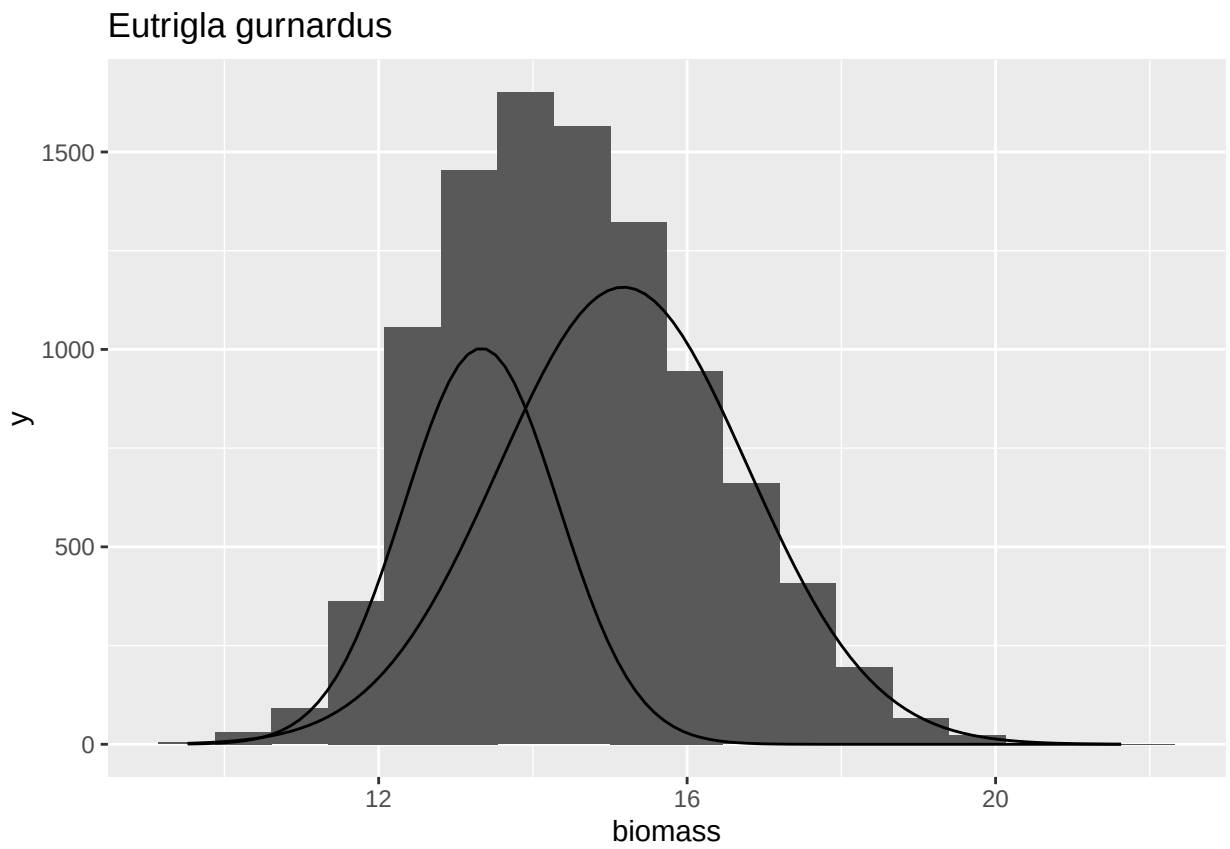


Eutrigla gurnardus

```
## number of iterations= 744
```

```
## WARNING! NOT CONVERGENT!
## number of iterations= 1000
```

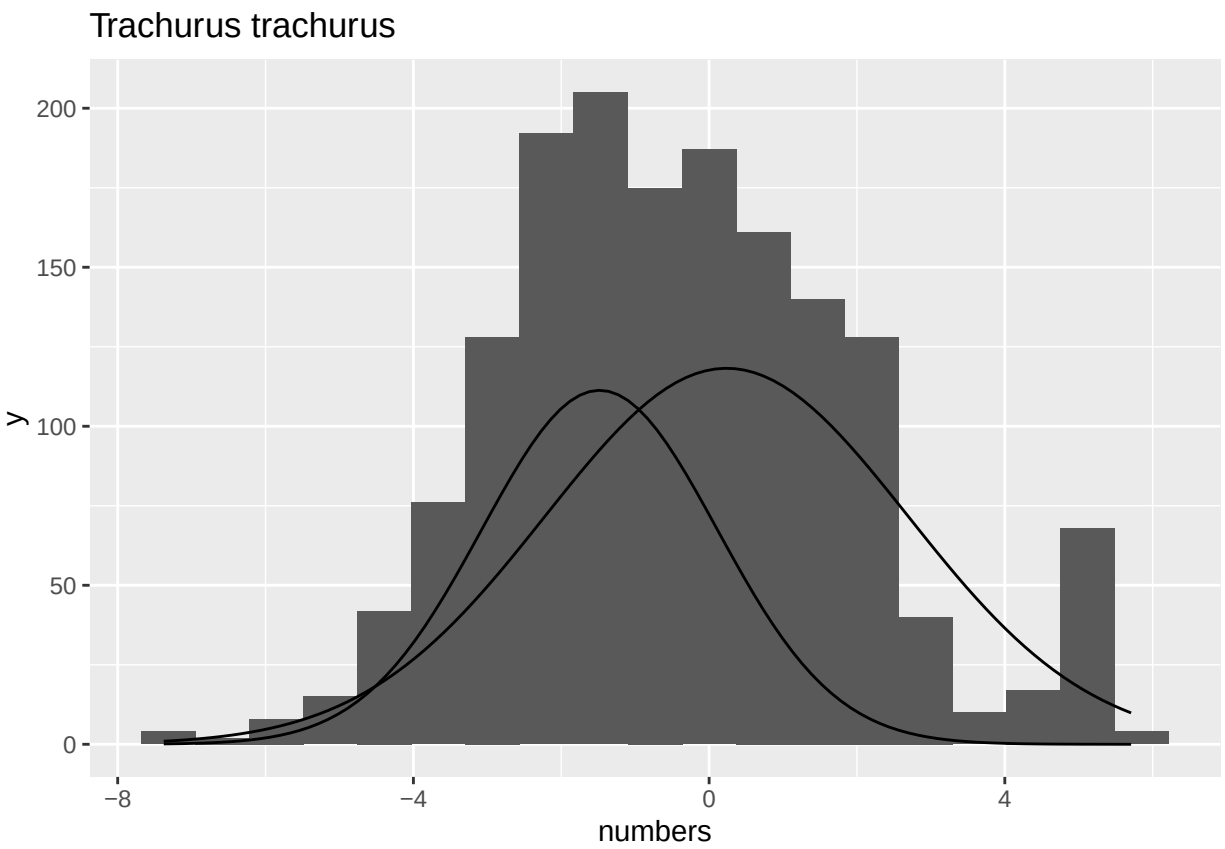

Eutrigla gurnardus
1500
1000
500
0
y
−5
0
numbers

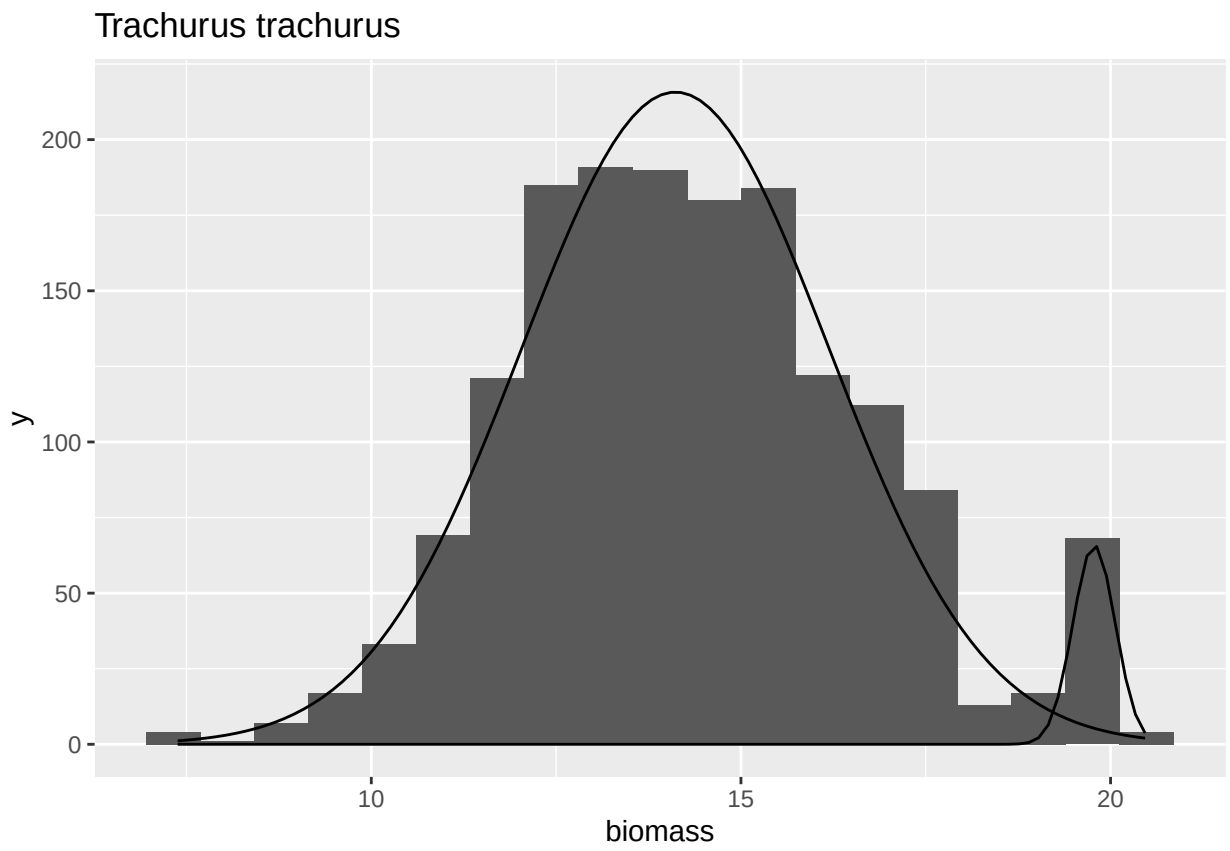


Trachurus trachurus

```
## number of iterations= 892
```

```
## number of iterations= 259
```

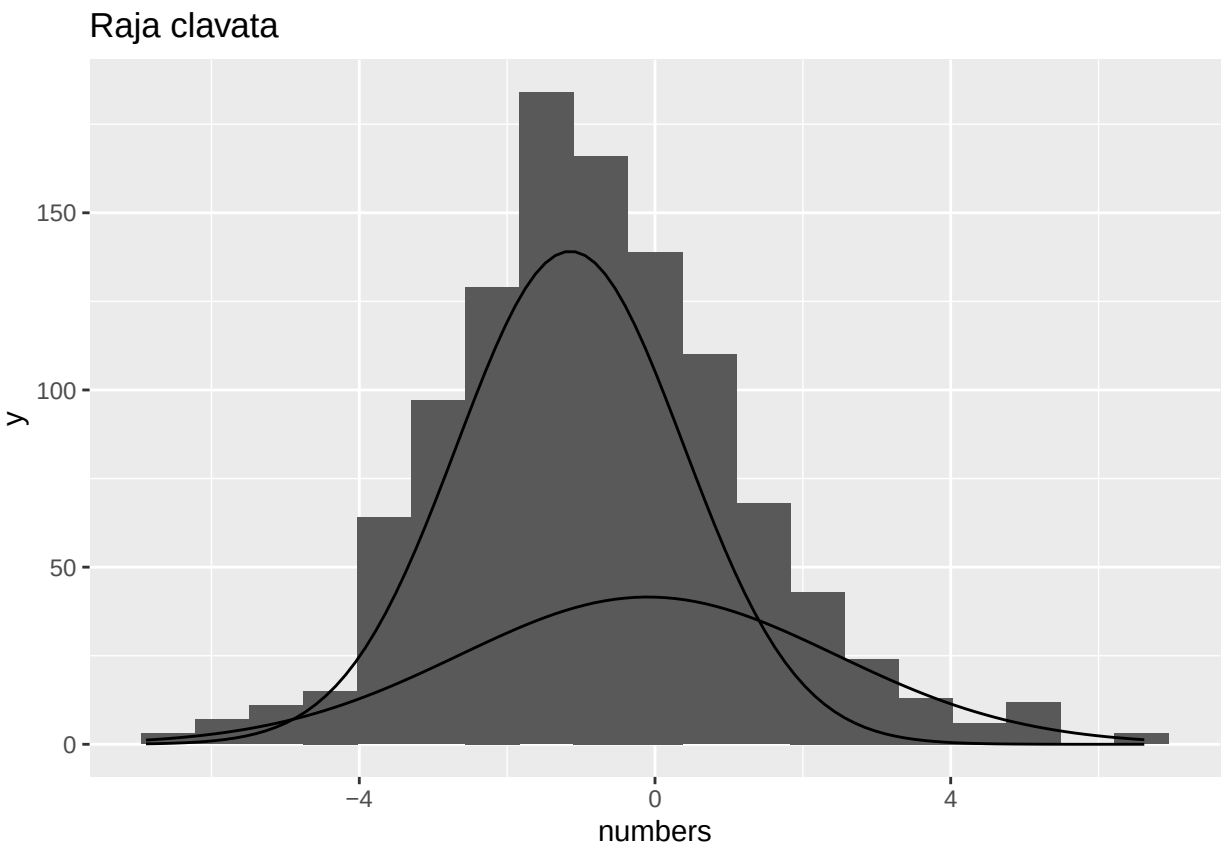

Trachurus trachurus
200
150
100
50
0
y
−8
−4
0
4
numbers

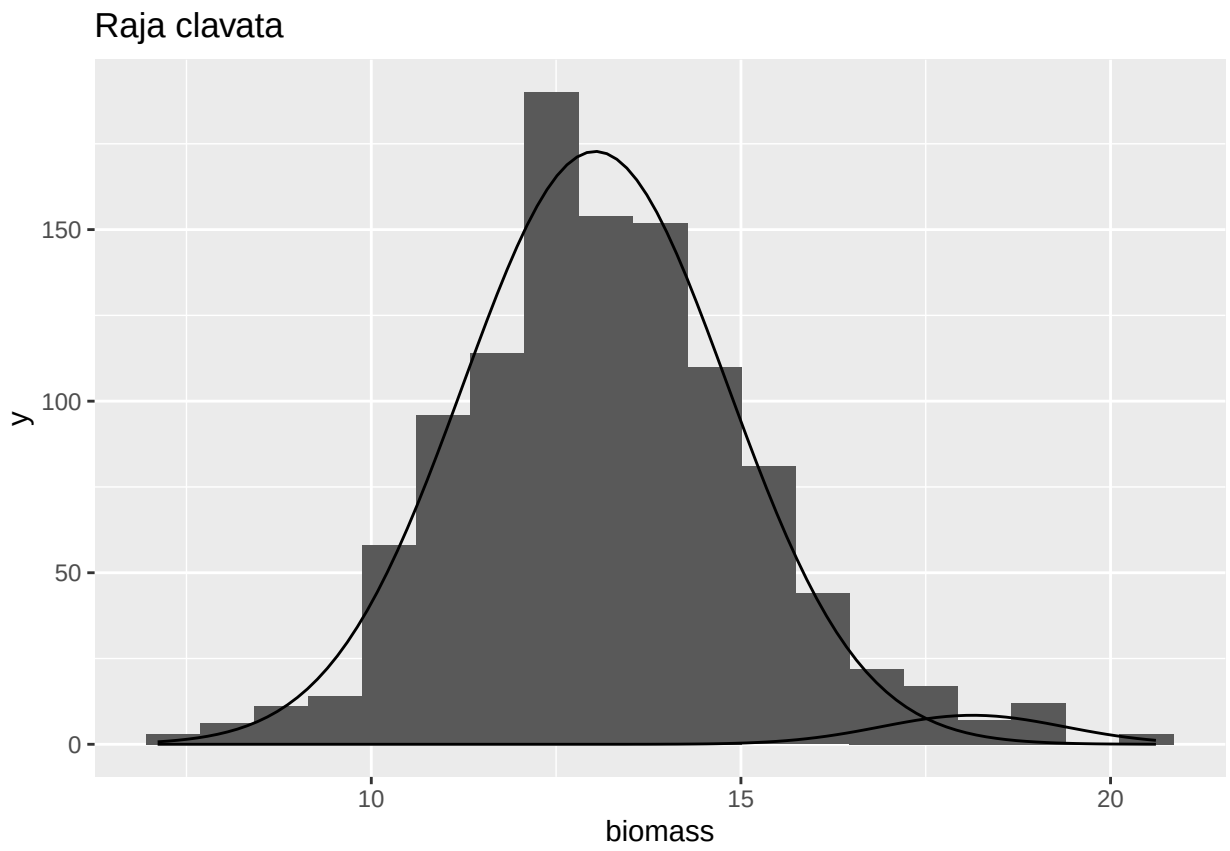


Raja clavata

```
## number of iterations= 399
```

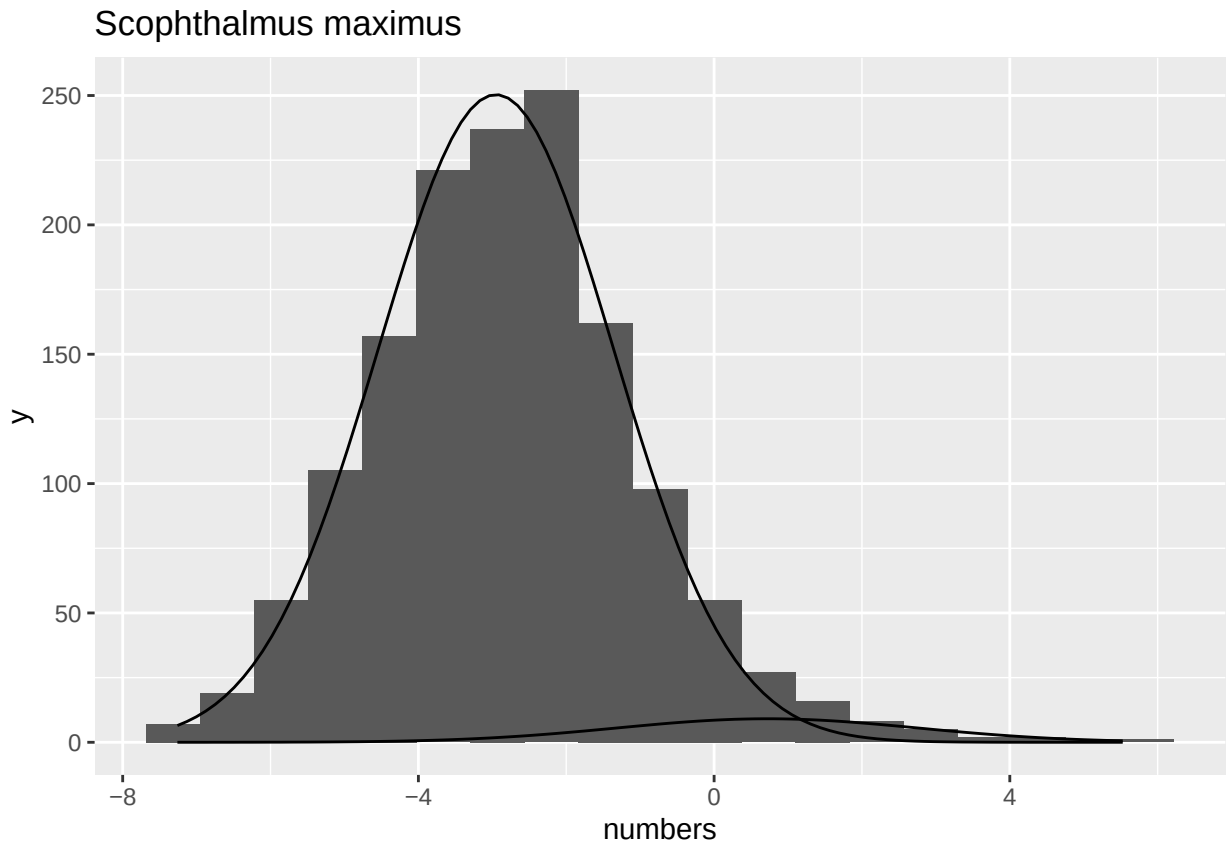
```
## number of iterations= 270
```

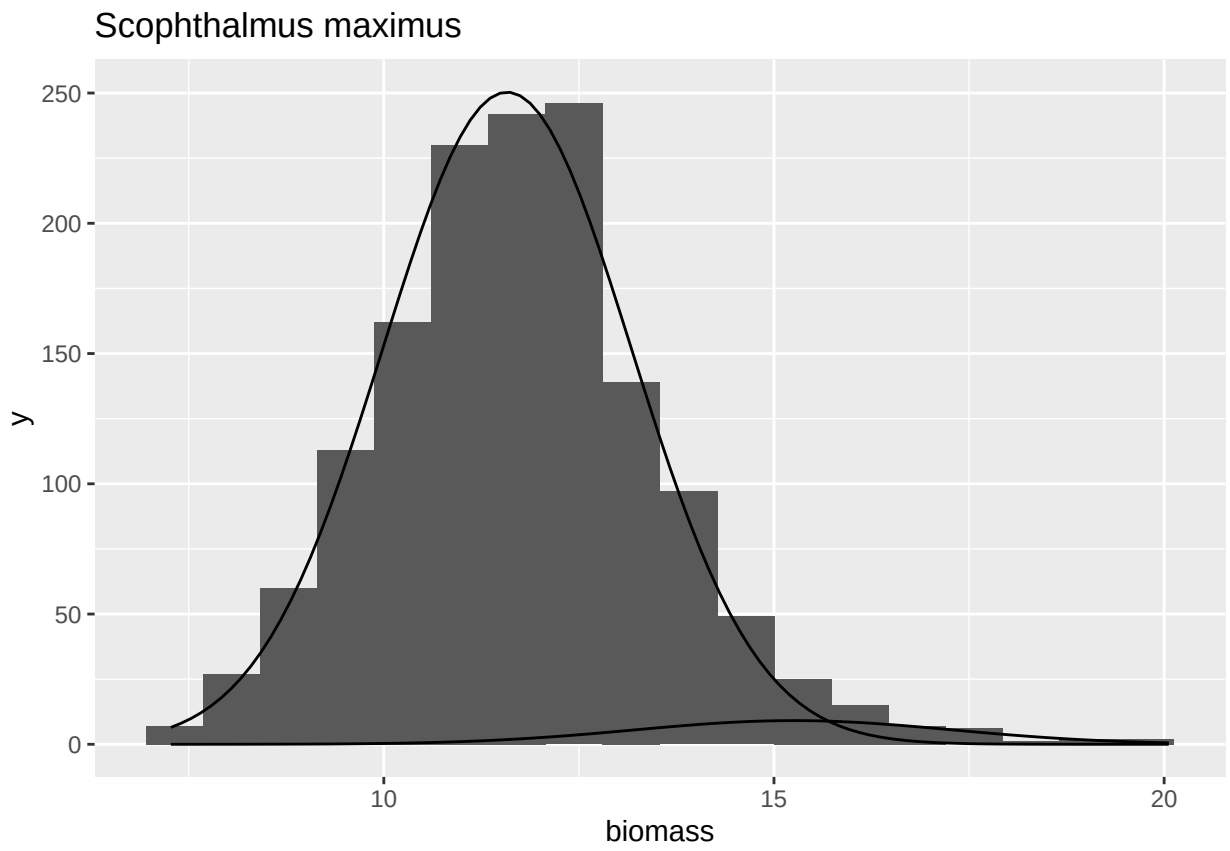

Raja clavata
150
100
50
0
y
−4
0
4
numbers



Scophthalmus maximus

```
## WARNING! NOT CONVERGENT!
## number of iterations= 1000

## number of iterations= 996
```

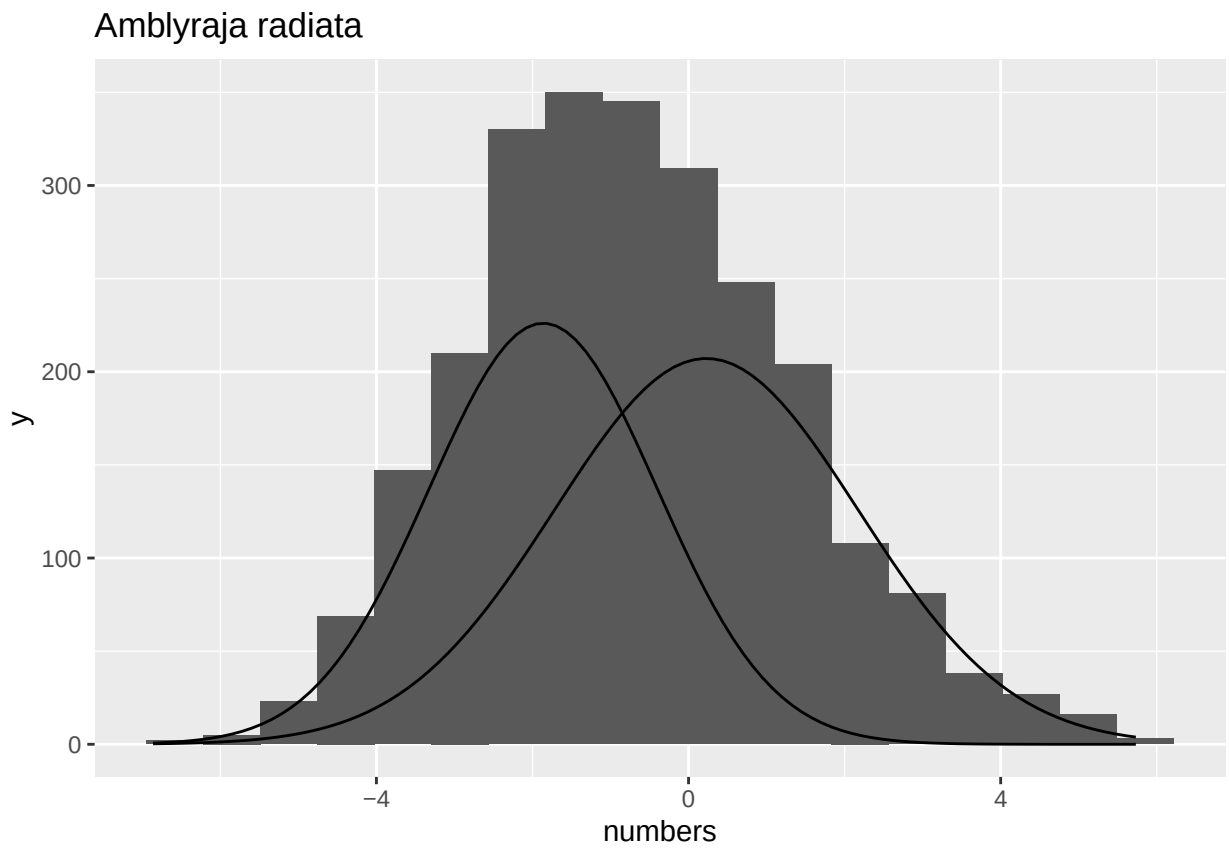

Scophthalmus maximus
250
200
150
100
50
0
y
−8
−4
0
4
numbers

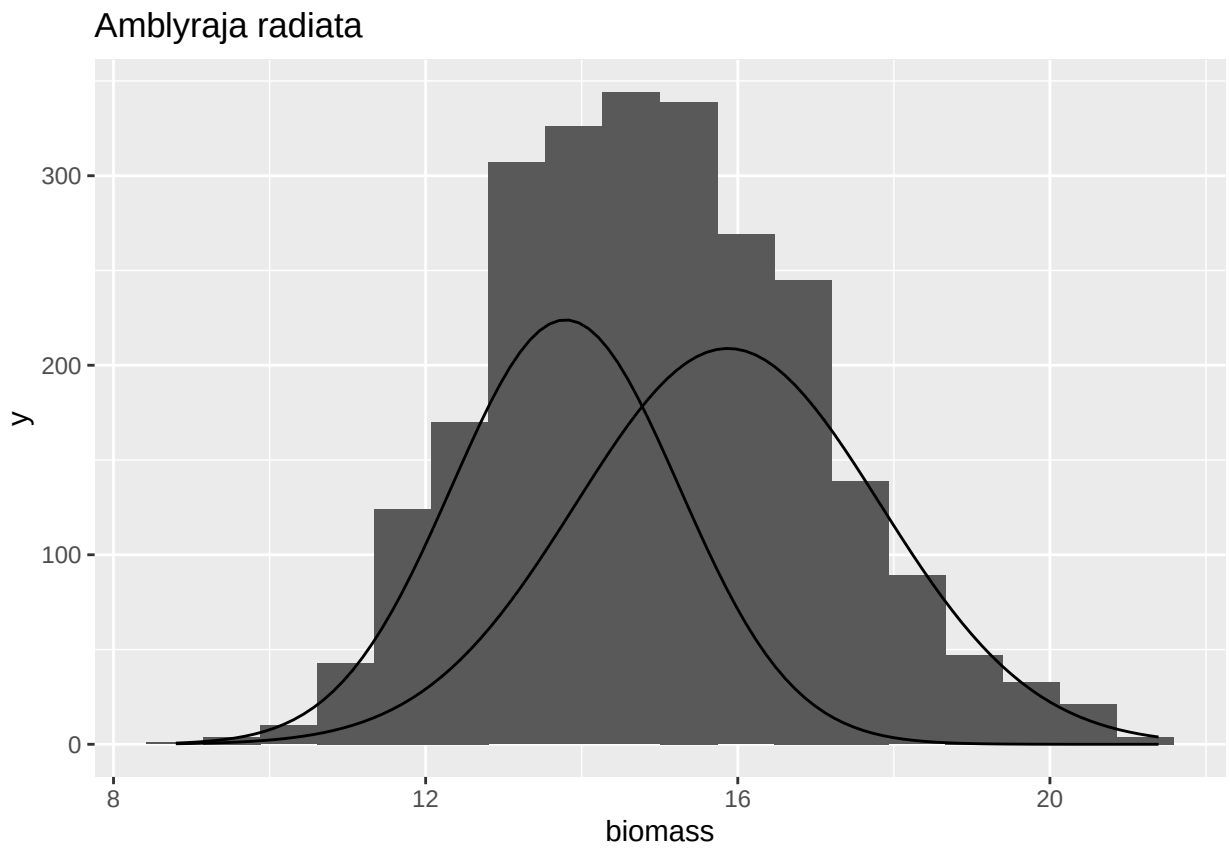


Amblyraja radiata

```
## WARNING! NOT CONVERGENT!
## number of iterations= 1000
```

```
## WARNING! NOT CONVERGENT!
## number of iterations= 1000
```

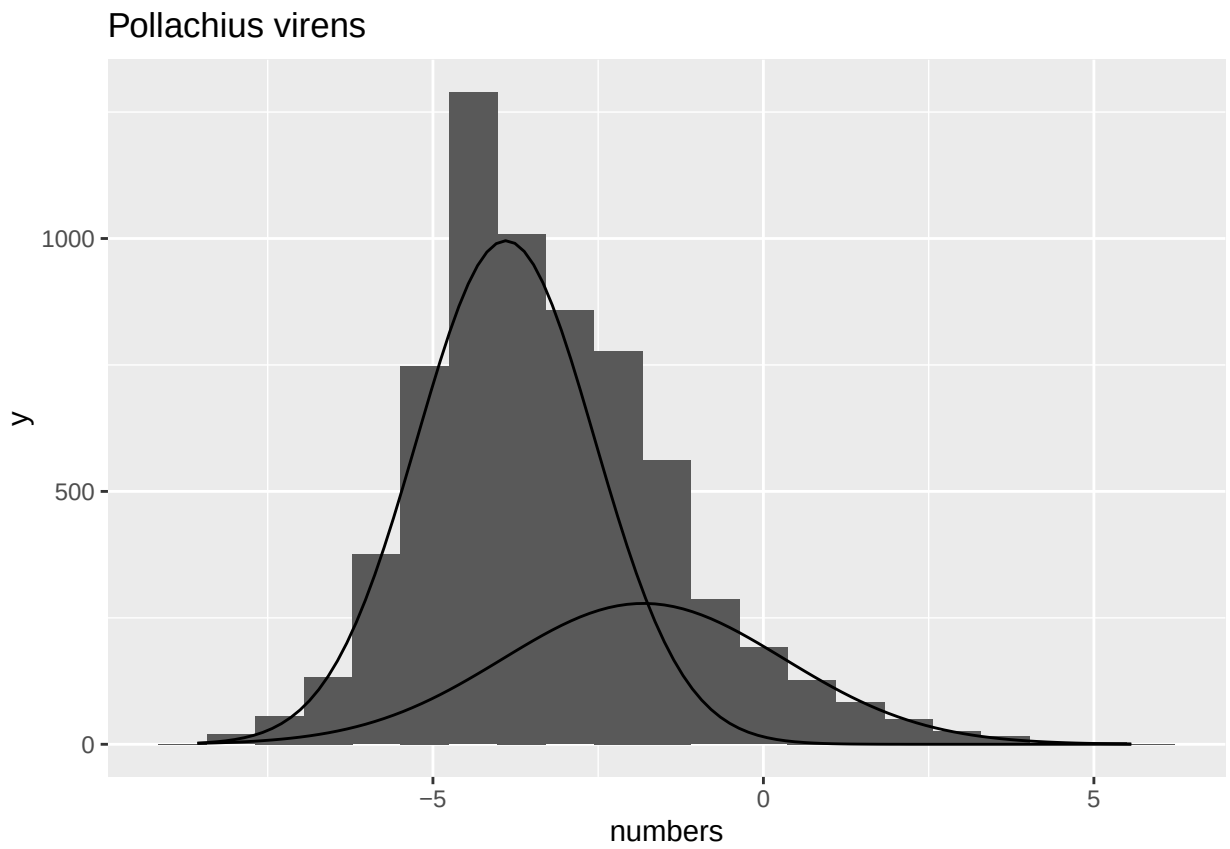

Amblyraja radiata
300
200
100
0
y
−4
0
4
numbers

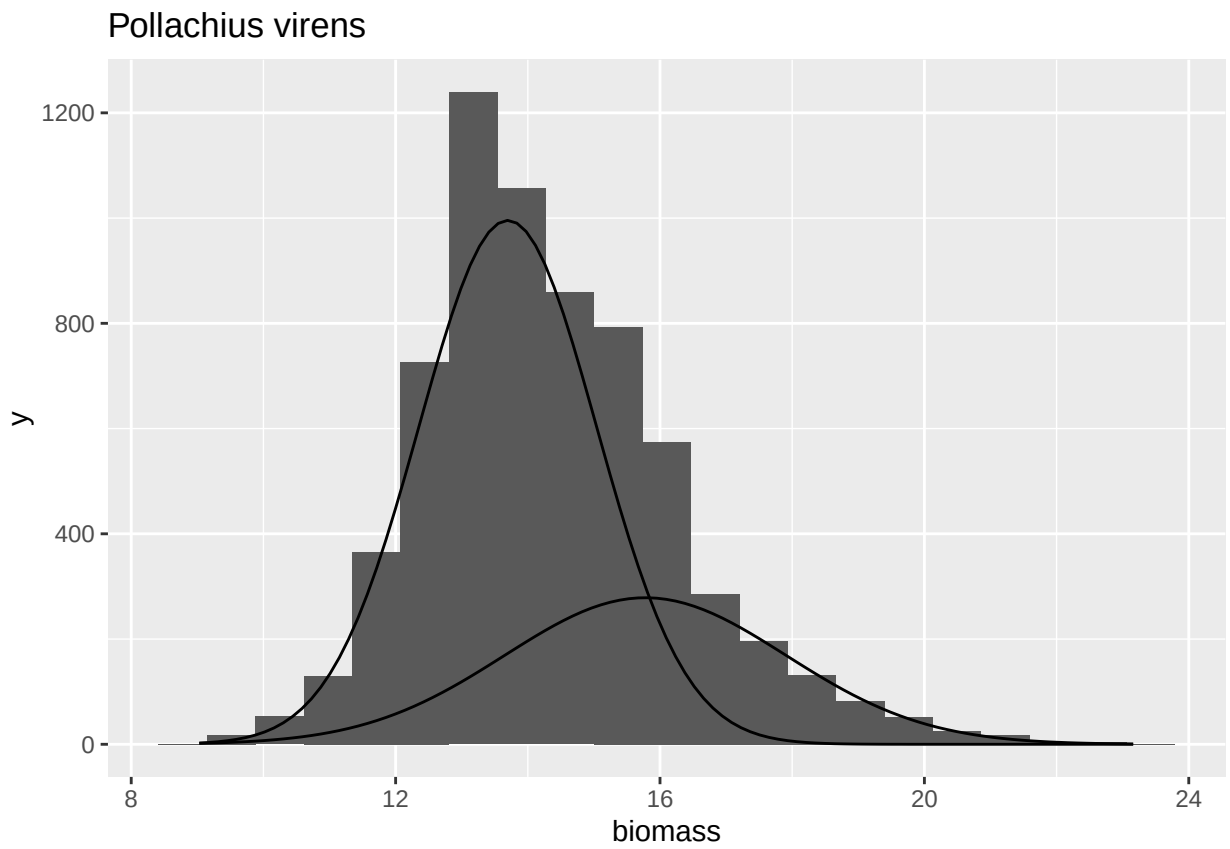


Pollachius virens

```
## number of iterations= 668
```

```
## number of iterations= 657
```

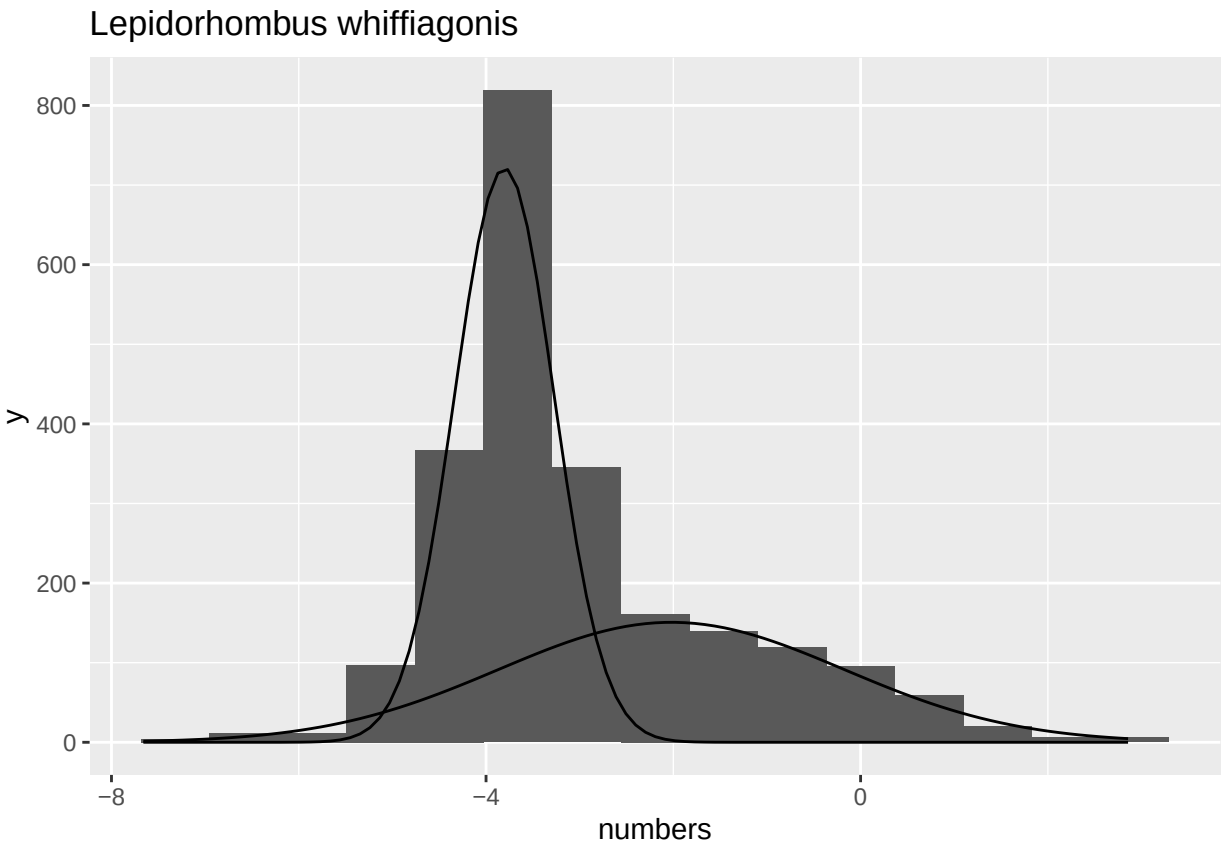

Pollachius virens
1000
500
0
y
−5
0
5
numbers

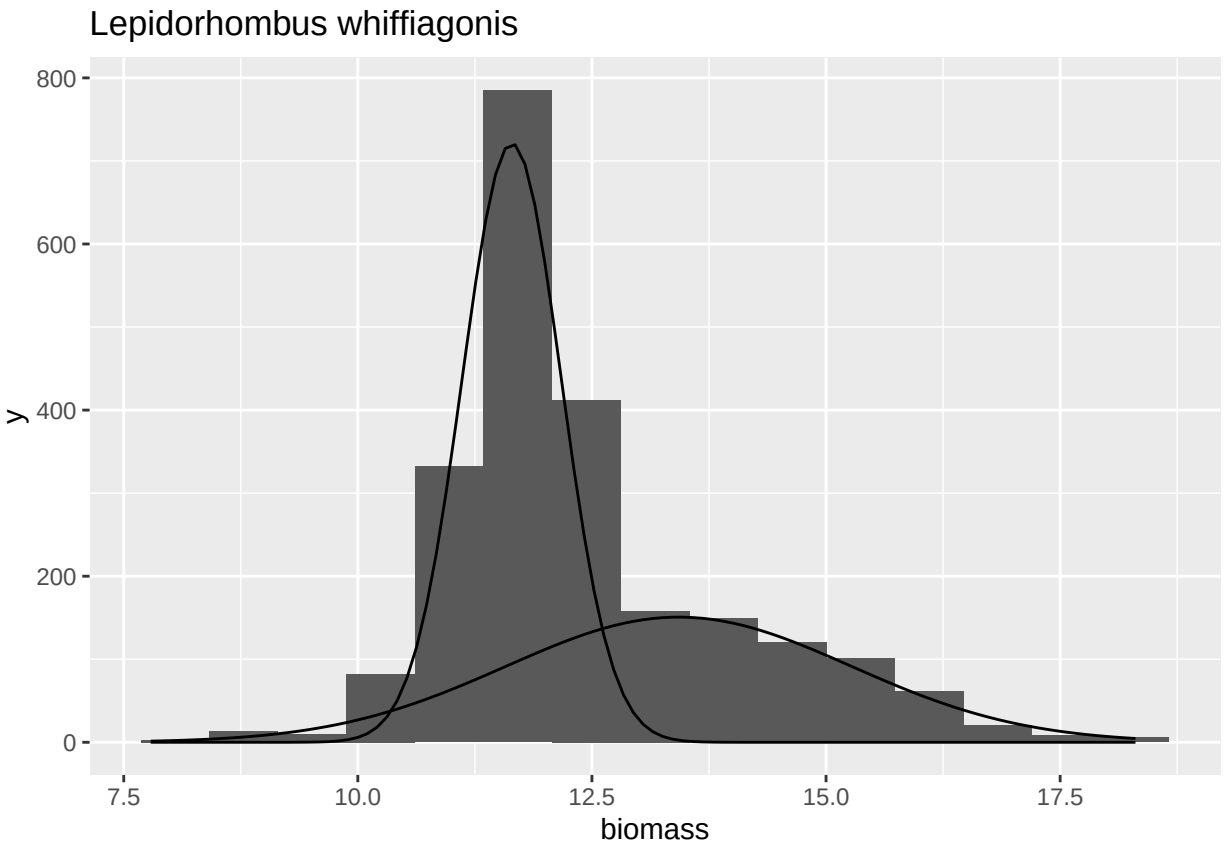


Lepidorhombus whiffiagonis

```
## number of iterations= 35
```

```
## number of iterations= 37
```


Lepidorhombus whiffiagonis
800
600
400
200
0
y
−8
−4
0
numbers


Lepidorhombus whiffiagonis
800
600
400
200
0
y
7.5
10.0
12.5
15.0
17.5
biomass