

b— title: “Haddock” output: html\_document date: “2024-07-15” —

```
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.3.3

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##     filter, lag

## The following objects are masked from 'package:base':
##     intersect, setdiff, setequal, union

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.3

load("C:/Users/lucab/Downloads/stomach_dataset.Rdata")

haddock <- stom_df%>%filter(pred_taxa=="Melanogrammus aeglefinus")

#whiting <- whiting%>%filter(pred_weight_g<10000, nprey_perpred > 1)

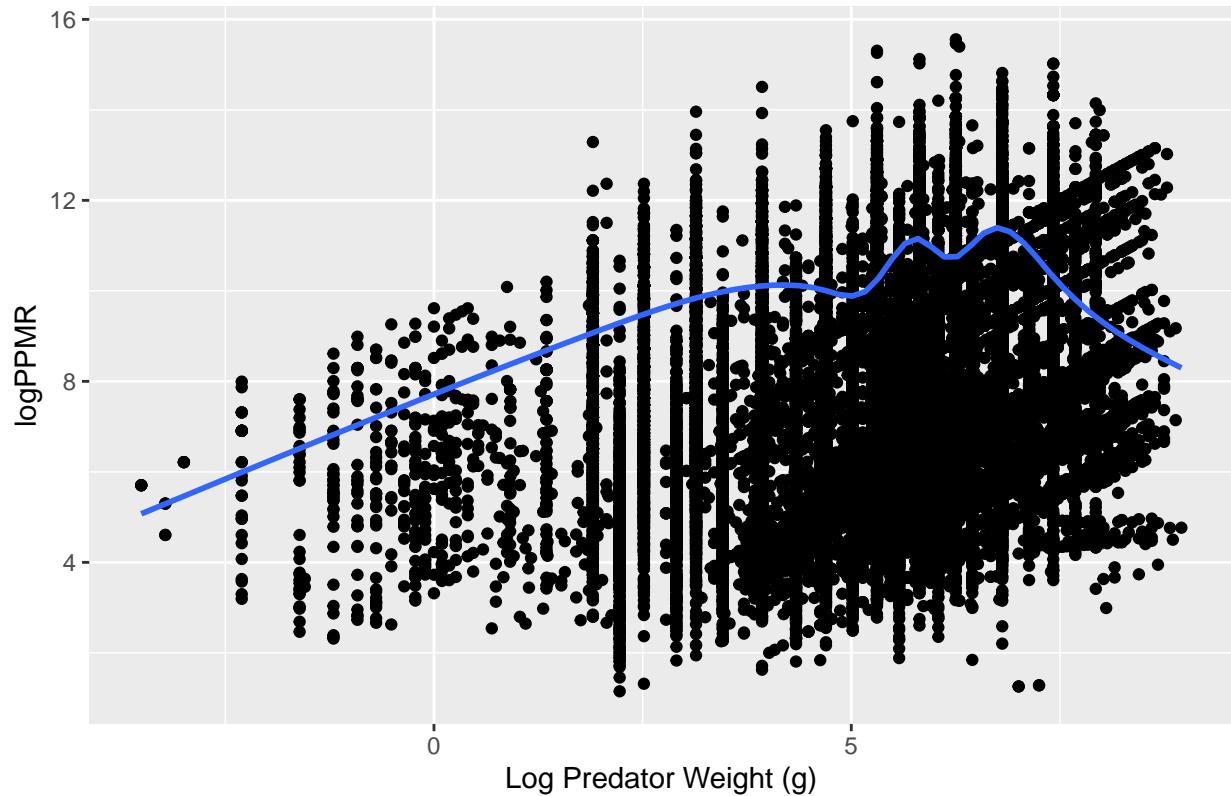
ggplot(haddock, aes(x=log(pred_weight_g), y=log(ppmr)))+
  geom_point()+
  geom_smooth(method="gam", se=FALSE, aes(weight = nprey_perpred))+
  labs(title="logPPMR vs Predator Weight", x="Log Predator Weight (g)", y="logPPMR")

## ‘geom_smooth()’ using formula = 'y ~ s(x, bs = "cs")'

## Warning: Removed 26 rows containing non-finite outside the scale range
## ('stat_smooth()').

## Warning: Removed 26 rows containing missing values or values outside the scale range
## ('geom_point()').
```

## logPPMR vs Predator Weight



```

dig <- 1

ggplot(haddock, aes(x=(log(pred_weight_g)), y=log(ppmr)))+
  geom_point()+
  #facet_wrap(~data)+
  geom_smooth(method="gam", se=FALSE, aes(weight = nprey_perpred*haddock$prey_ind_weight_g^dig))+ 
  labs(title="logPPMR vs Predator Weight", x="Log Predator Weight (g)", y="logPPMR")

## Warning: Use of 'haddock$prey_ind_weight_g' is discouraged.
## i Use 'prey_ind_weight_g' instead.

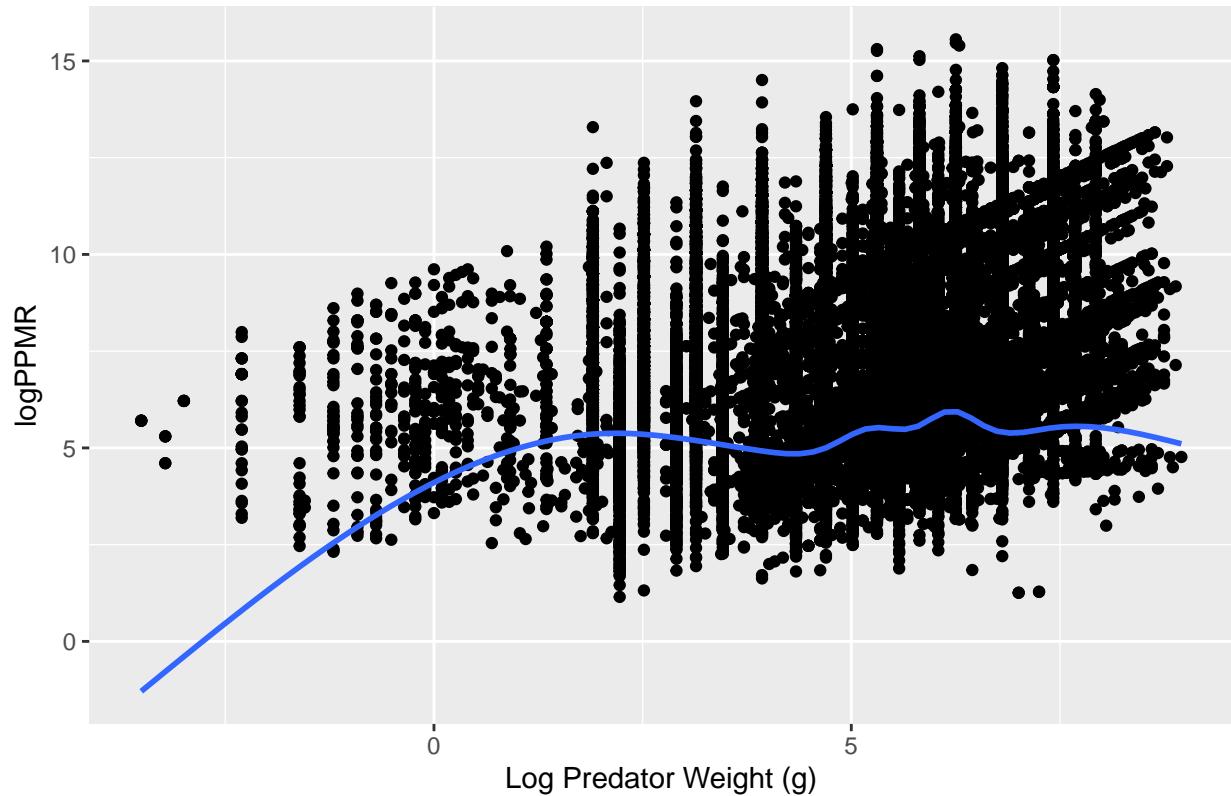
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'

## Warning: Removed 26 rows containing non-finite outside the scale range
## ('stat_smooth()').

## Warning: Removed 26 rows containing missing values or values outside the scale range
## ('geom_point()').

```

## logPPMR vs Predator Weight



Once it is weighted by biomass, it works, but the PPMR still decreases at larger sizes. I think it should still be okay.

```

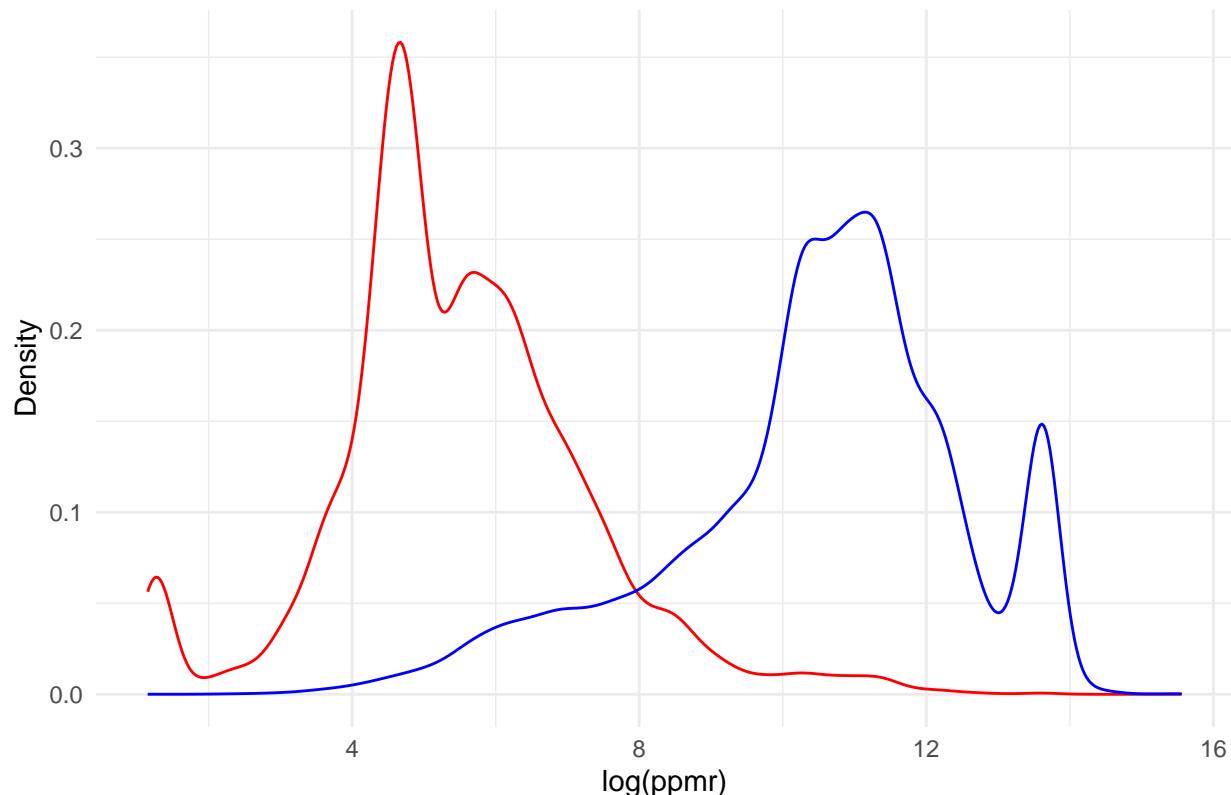
stomach <- haddock
stomach$weight_numbers <- stomach$nprey_perpred
stomach$weight_biomass <- stomach$nprey_perpred*stomach$prey_ind_weight_g^dig

ggplot() +
  geom_density(data = stomach, aes(x = log(ppmr), weight = weight_biomass), color = "red") +
  geom_density(data = stomach, aes(x = log(ppmr), weight = weight_numbers), color = "blue") +
  labs(title = "Density Plot of log(ppmr) Weighted by Biomass and Numbers",
       x = "log(ppmr)",
       y = "Density") +
  theme_minimal()

## Warning: Removed 26 rows containing non-finite outside the scale range
## ('stat_density()').
## Removed 26 rows containing non-finite outside the scale range
## ('stat_density()').

```

## Density Plot of log(ppmr) Weighted by Biomass and Numbers



I think that a mixture gaussian model would fit this best.

```
library(mclust)

## Warning: package 'mclust' was built under R version 4.3.3

## Package 'mclust' version 6.1.1
## Type 'citation("mclust")' for citing this R package in publications.

repeat_elements <- function(data, weights) {

  valid_indices <- !is.na(data) & !is.na(weights)
  data <- data[valid_indices]
  weights <- weights[valid_indices]

  final_vector <- c()

  for (i in seq_along(data)) {

    rounded_weight <- round(weights[i])

    repeated_values <- rep(data[i], times = rounded_weight)

    final_vector <- c(final_vector, repeated_values)
  }
}
```

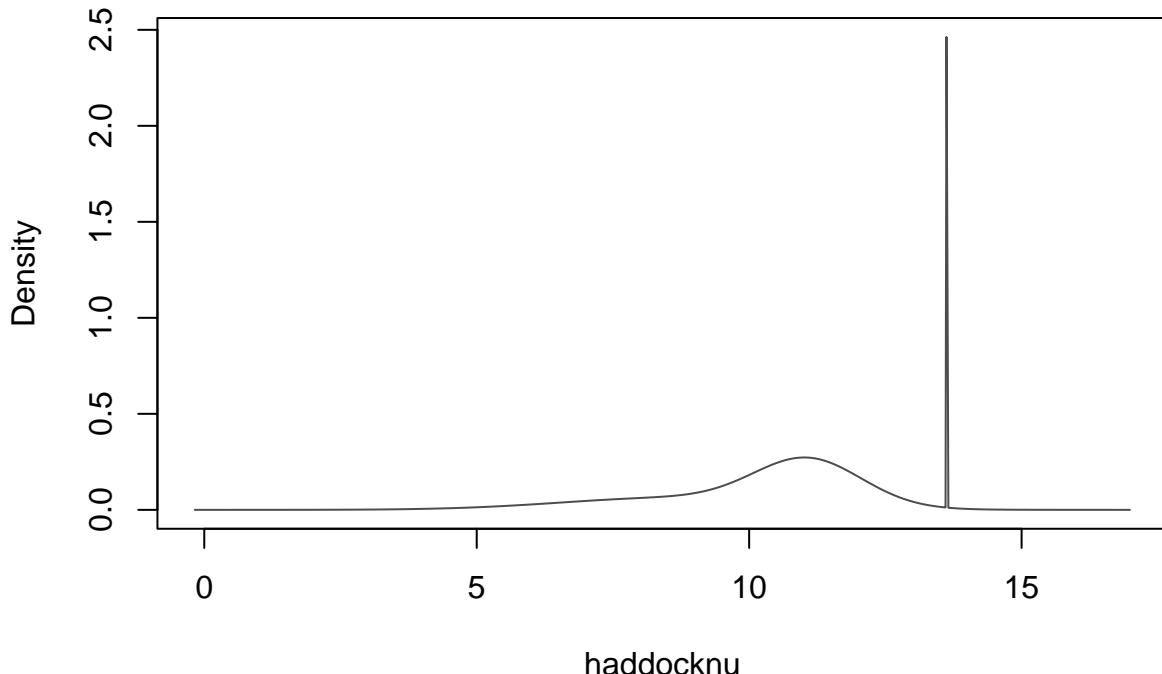
```

    return(final_vector)
}

haddocknu <- repeat_elements(log(stomach$ppmr), stomach$nprey_perpred)

gmm <- densityMclust(haddocknu, G=3)

```



This isn't the best fit, it misses out a peak at higher PPMR. When it is run with 3 gaussians, the 3rd peak has an extremely high density, which makes me think that there is something wrong with the data here. If it didn't have such a high density it would fit it very well.

I want to see if I filter out this data it would work better, or if this data is from the same year/region.

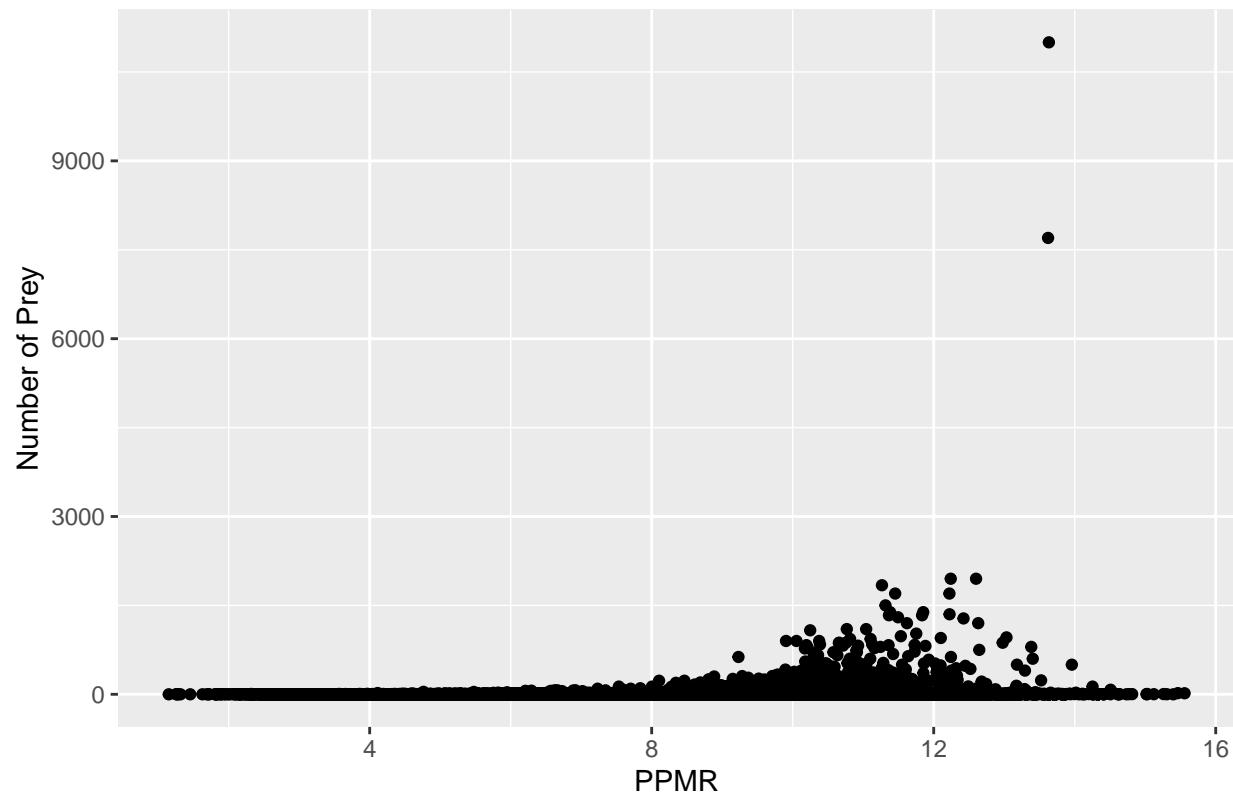
```

ggplot(stomach, aes(x=log(ppmr), y=nprey_perpred))+
  geom_point()+
  labs(title="PPMR vs Number of Prey", x="PPMR", y="Number of Prey")

## Warning: Removed 26 rows containing missing values or values outside the scale range
## ('geom_point()').

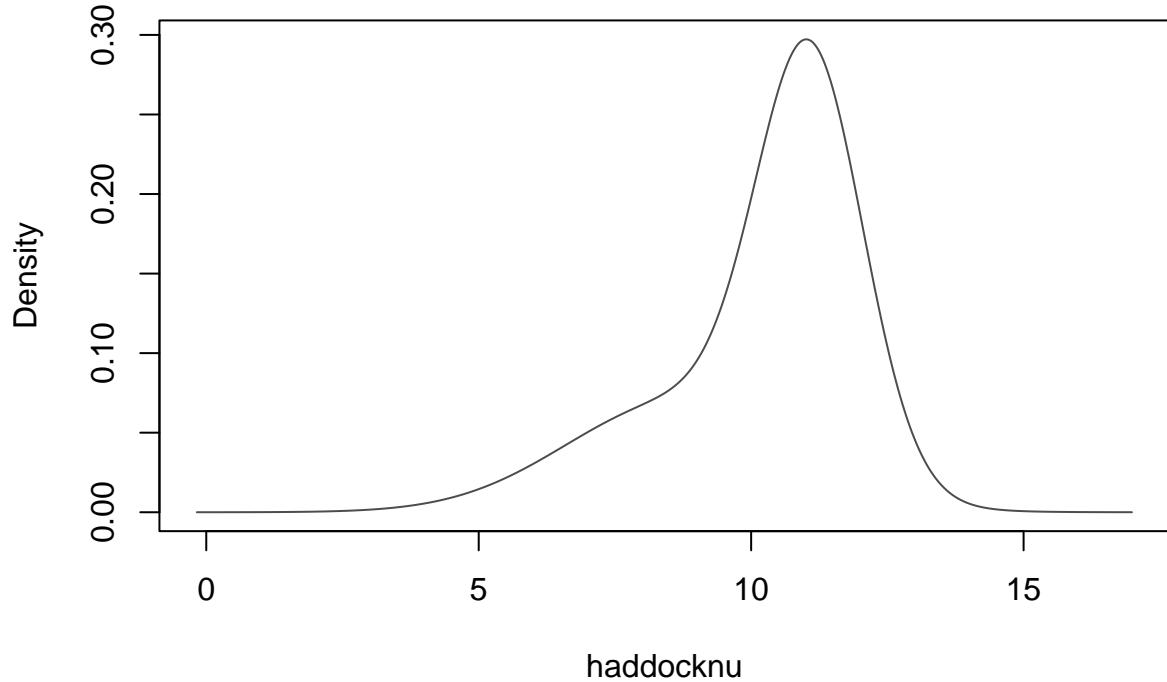
```

## PPMR vs Number of Prey



Ok, so there are two datapoints with a very high nprey per pred, so this means that it skews the distribution, I want to remove it and rerun the fit.

```
stomach <- stomach%>%filter(nprey_perpred<2000)
haddocknu <- repeat_elements(log(stomach$ppmr), stomach$nprey_perpred)
gmm <- densityMclust(haddocknu, G=2)
```



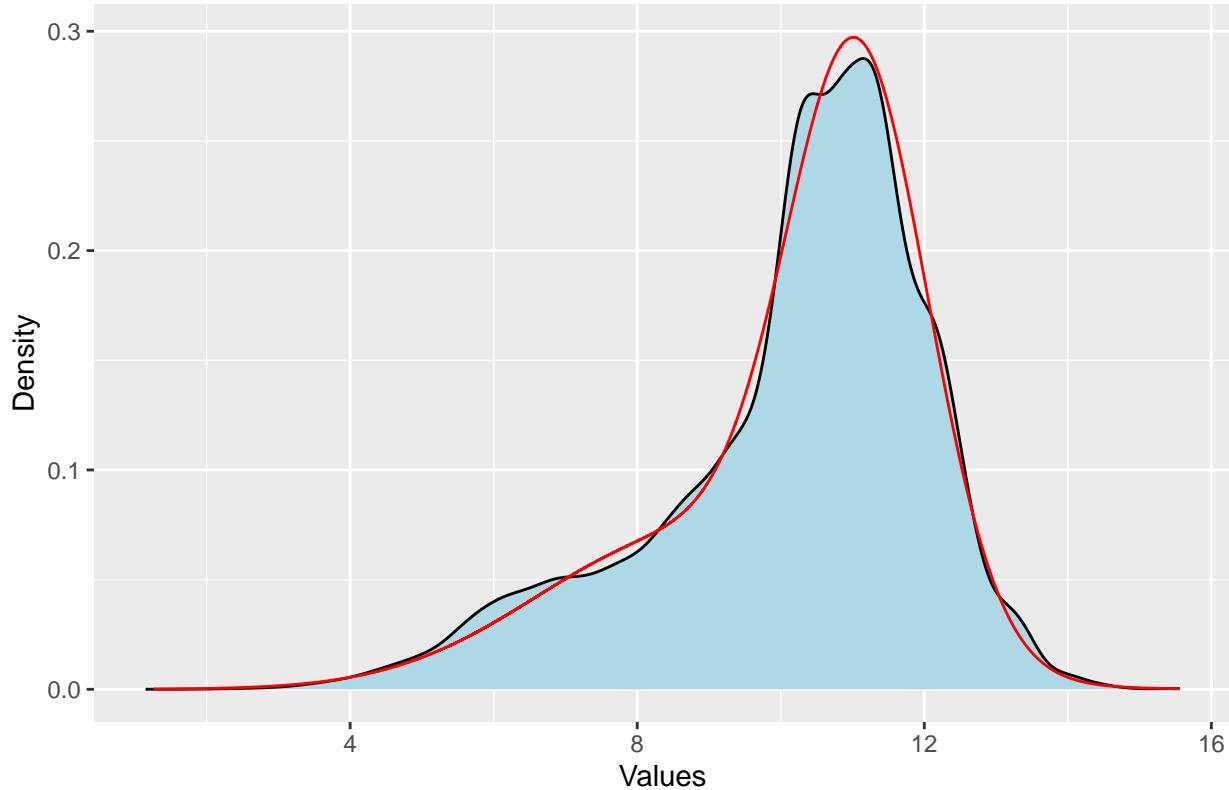
This looks to be a good fit. Lets see if it shifts!

```
dplot <- data.frame(x=gmm[["data"]], density=gmm[["density"]])

(biofit <- ggplot() +
  geom_density(data=stomach, aes(log(ppmr), weight=weight_numbers), fill="lightblue")+
  geom_line(data=dplot, aes(x = x, y = density), color="red") +
  labs(x = "Values", y = "Density") +
  ggtitle("Diet Density Plot from Diet Distribution"))

## Warning: Removed 26 rows containing non-finite outside the scale range
## ('stat_density()').
```

## Diet Density Plot from Diet Distribution



```

shifted_fit <- gmm
shifted_fit[["parameters"]][["mean"]] <- shifted_fit[["parameters"]][["mean"]]-
  (2/3)*shifted_fit[["parameters"]][["variance"]][["sigmasq"]]

stomach <- stomach%>%filter(!is.na(ppmr))

x_vals <- seq(min(log(stomach$ppmr)), max(log(stomach$ppmr)), length.out = 1000)

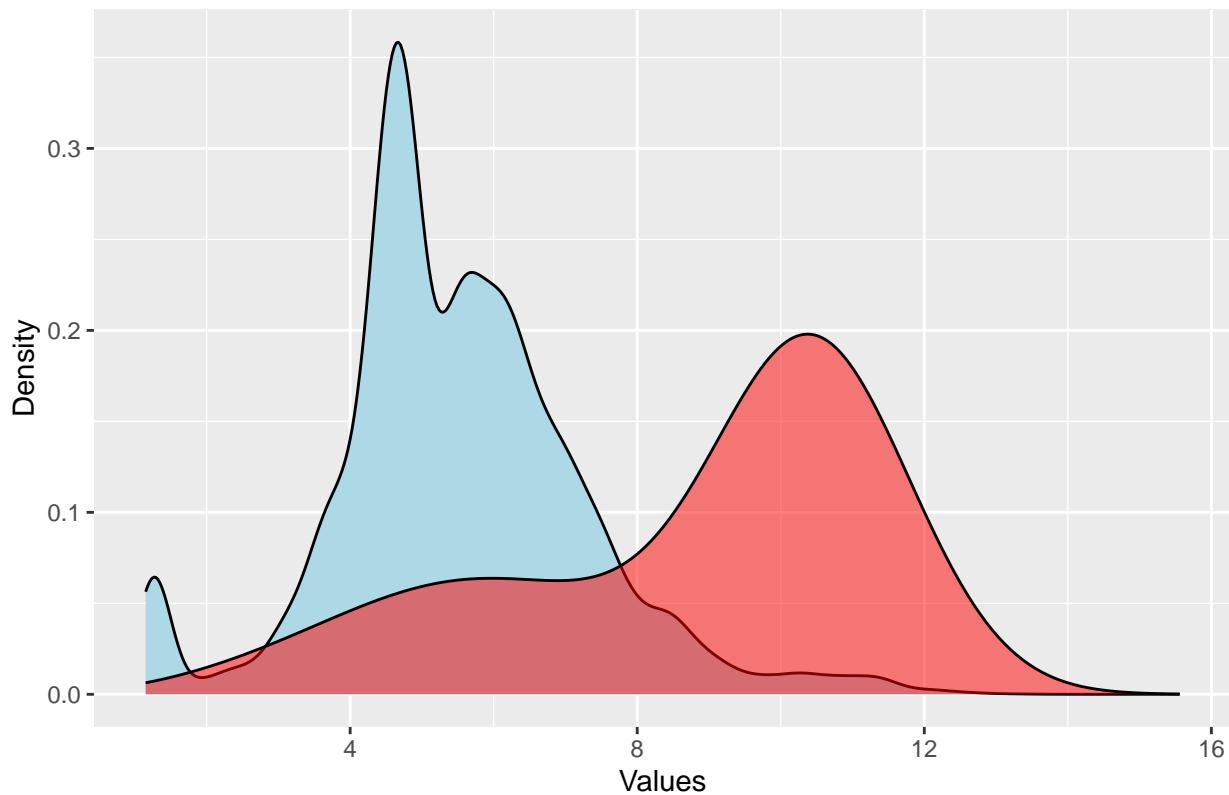
#generating the density values
shifted_pdf <- sapply(x_vals, function(x) {
  sum(shifted_fit$parameters$pro * dnorm(x, mean = shifted_fit$parameters$mean, sd = sqrt(shifted_fit$parameters$variance)))
})

plot_data <- data.frame(x = x_vals, shifted_pdf = shifted_pdf)

(numbfitbio <- ggplot() +
  geom_density(data=stomach, aes(log(ppmr), weight=weight_biomass), fill="lightblue")+
  geom_density(data=plot_data, aes(x, weight=shifted_pdf), fill="red", alpha=0.5)+
  #geom_line(data=dplot, aes(x = x, y = density), color="red") +
  labs(x = "Values", y = "Density") +
  ggtitle("Diet Density Plot from Number Distribution"))

```

## Diet Density Plot from Number Distribution



```
#doing it in a different way

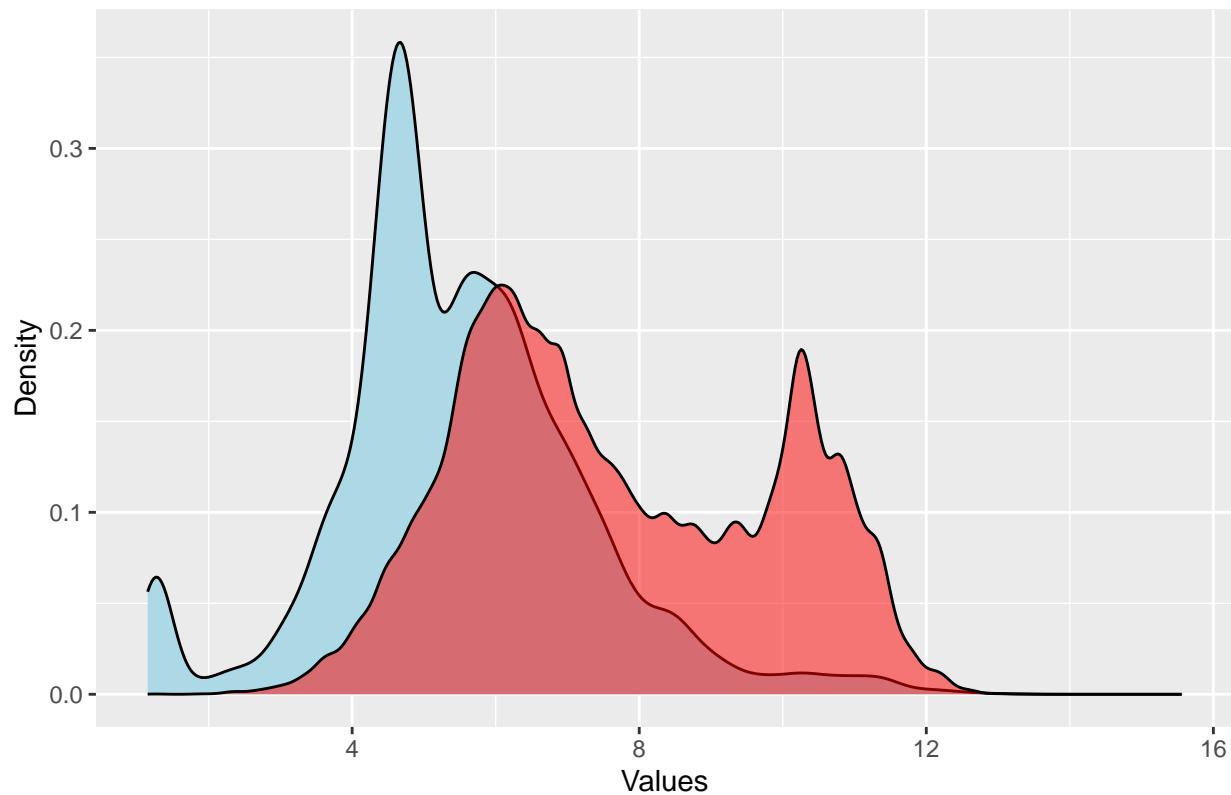
newspratdens <- gmm[["density"]]*exp(-dig*gmm[["data"]])

shifted_pdf_normalized <- newspratdens / sum(newspratdens)

dplot <- data.frame(x=gmm[["data"]], density=shifted_pdf_normalized)

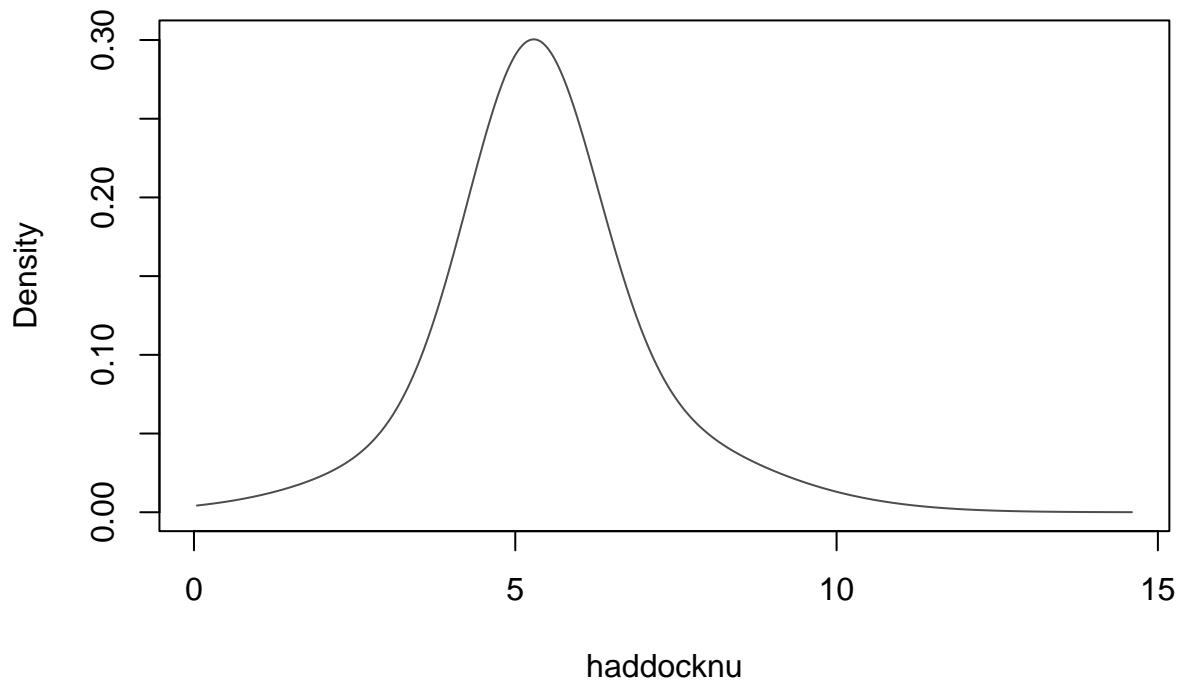
(numbbfitbio <- ggplot() +
  geom_density(data=stomach, aes(log(ppmr), weight=weight_biomass), fill="lightblue")+
  geom_density(data=dplot, aes(x, weight=shifted_pdf_normalized), fill="red", alpha=0.5) +
  #geom_line(data=dplot, aes(x = x, y = density), color="red") +
  labs(x = "Values", y = "Density") +
  ggtitle("Diet Density Plot from Number Distribution"))
```

## Diet Density Plot from Number Distribution



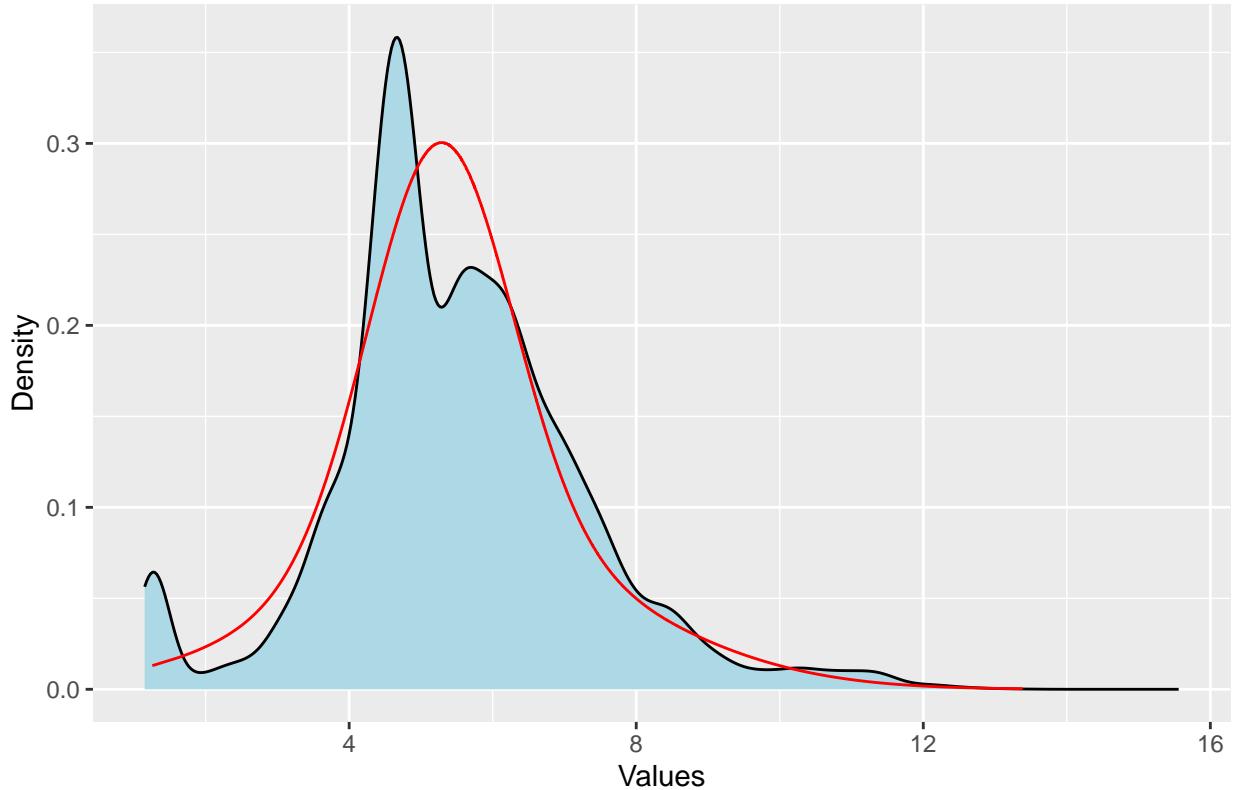
Ok, so it doesn't shift, will have to try biomass to number

```
haddocknu <- repeat_elements(log(stomach$ppmr), stomach$nprey_perpred*
                                stomach$prey_ind_weight_g^dig)
gmm <- densityMclust(haddocknu, G=2)
```



```
dplot <- data.frame(x=gmm[["data"]], density=gmm[["density"]])  
  
(biofit <- ggplot() +  
  geom_density(data=stomach, aes(log(ppmr), weight=weight_biomass), fill="lightblue") +  
  geom_line(data=dplot, aes(x = x, y = density), color="red") +  
  labs(x = "Values", y = "Density") +  
  ggtitle("Diet Density Plot from Diet Distribution"))
```

## Diet Density Plot from Diet Distribution



```

shifted_fit <- gmm
shifted_fit[["parameters"]][["mean"]] <- shifted_fit[["parameters"]][["mean"]]+
  (2/3)*shifted_fit[["parameters"]][["variance"]][["sigmasq"]]

stomach <- stomach%>%filter(!is.na(ppmr))

x_vals <- seq(min(log(stomach$ppmr)), max(log(stomach$ppmr)), length.out = 1000)

#generating the density values
shifted_pdf <- sapply(x_vals, function(x) {
  sum(shifted_fit$parameters$pro * dnorm(x, mean = shifted_fit$parameters$mean, sd = sqrt(shifted_fit$p
})}

plot_data <- data.frame(x = x_vals, shifted_pdf = shifted_pdf)

#different way

newspratdens <- gmm[["density"]]*exp(dig*gmm[["data"]])

shifted_pdf_normalized <- newspratdens / sum(newspratdens)

dplot <- data.frame(x=gmm[["data"]], density=shifted_pdf_normalized)

(numbfitbio <- ggplot() +
  geom_density(data=stomach, aes(log(ppmr), weight=weight_numbers), fill="lightblue")+

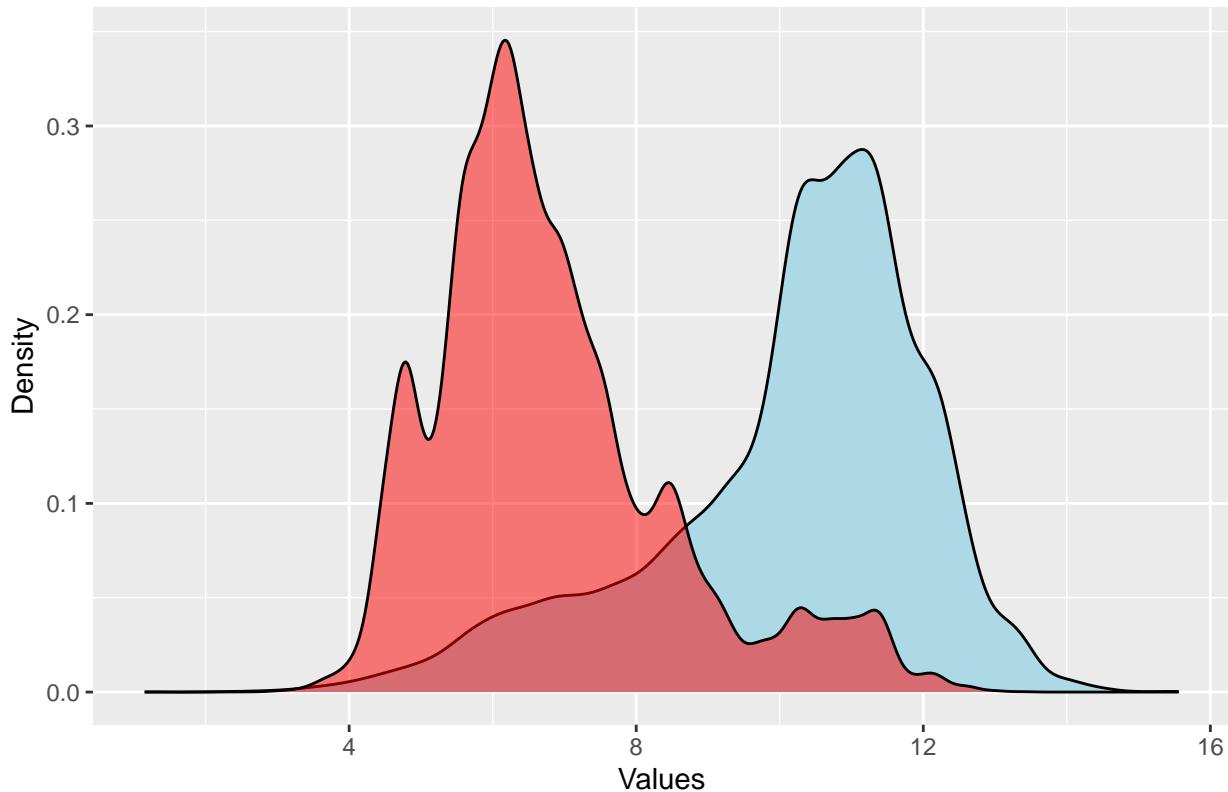
```

```

geom_density(data=dplot, aes(x, weight=shifted_pdf_normalized), fill="red", alpha=0.5) +
#geom_line(data=dplot, aes(x = x, y = density), color="red") +
labs(x = "Values", y = "Density") +
ggtitle("Diet density shifted to number density"))

```

Diet density shifted to number density



This isn't good. Will try the truncated exponential

```

f1 <- function(l, alpha, ll, ul, lr, ur) {
  dl <- ll - l
  dr <- l - lr
  f1_values <- exp(alpha * l) / (1 + exp(ul * dl)) / (1 + exp(ur * dr))

  # Debugging output
  if (any(!is.finite(f1_values))) {
    print("Non-finite f1 values found")
    print(f1_values)
  }

  return(f1_values)
}

## Define the truncated exponential PDF with debugging
dtxp <- function(l, alpha, ll, ul, lr, ur) {
  f1_values <- f1(l, alpha, ll, ul, lr, ur)

  integral_result <- tryCatch(

```

```

integrate(f1, 0, 30, alpha = alpha, ll = ll, ul = ul, lr = lr, ur = ur),
error = function(e) {
  print("Integration failed")
  print(e)
  return(NULL)
}
)

if (is.null(integral_result)) {
  return(rep(NA, length(l)))
}

d <- f1_values / integral_result$value

# Debugging output
if (any(!is.finite(d))) {
  print("Non-finite d values found")
  print(d)
}

return(d)
}

#Define the MLE function with debugging
mle_texp <- function(df) {
  loglik <- function(alpha, ll, ul, lr, ur) {
    L <- dtexp(df$l, alpha, ll, ul, lr, ur)

    # Debugging output
    if (any(!is.finite(L) | L <= 0)) {
      print("Non-finite or non-positive L values found")
      print(which(!is.finite(L) | L <= 0))
      return(Inf)
    }

    -sum(log(L) * df$weight_numbers)
  }

  result <- tryCatch(
    mle2(loglik, start = list(
      alpha = 0.5,
      ll = min(df$l),
      lr = max(df$l),
      ul = 5,
      ur = 5
    ), method = "L-BFGS-B", control = list(maxit = 10000)),
    error = function(e) {
      print("MLE fitting failed")
      print(e)
      return(NULL)
    }
  )
}

```

```

    return(result)
}

library(bbmle)

## Warning: package 'bbmle' was built under R version 4.3.3

## Loading required package: stats4

##
## Attaching package: 'bbmle'

## The following object is masked from 'package:dplyr':
##      slice

stomach <- stomach%>%mutate(l=log(ppmr))

est <- mle_texp(stomach)

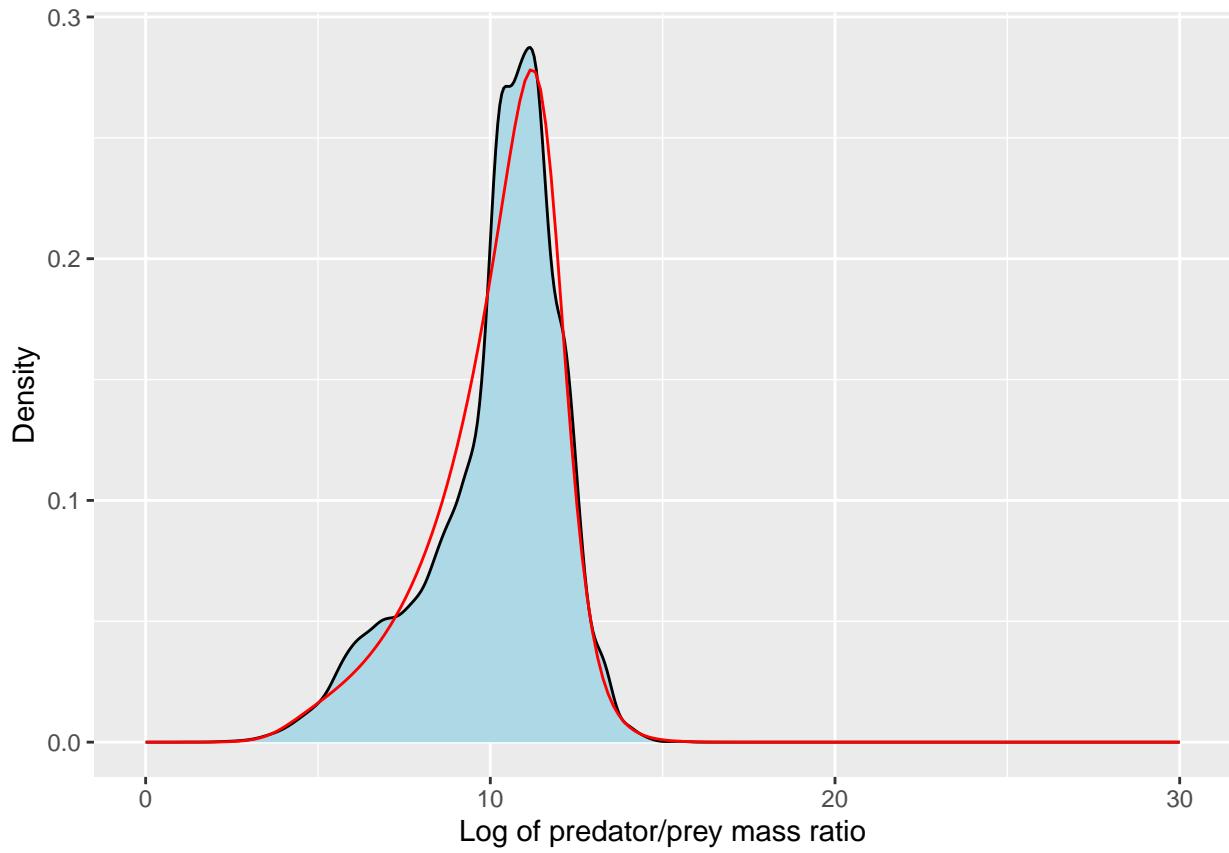
biomassco <- est@coef

grid = seq(0, 30, length.out = 200)
#here, the alpha is meant to be -1, but I have to subtract 0.7 to make it work, so I am going to run the
#for the biomass, and see the difference
dist <- dtexp(grid, alpha = (biomassco[1]), ll = biomassco[2], ul = biomassco[3],
              lr = biomassco[4], ur = biomassco[5])

dist <- data.frame(l=grid, Density=dist)

ggplot(stomach) +
  geom_density(aes(l, weight=weight_numbers), fill="lightblue")+
  xlab("Log of predator/prey mass ratio") +
  geom_line(aes(l, Density), data = dist, color = "red")

```



Ok the exponential fits well, now lets shift!

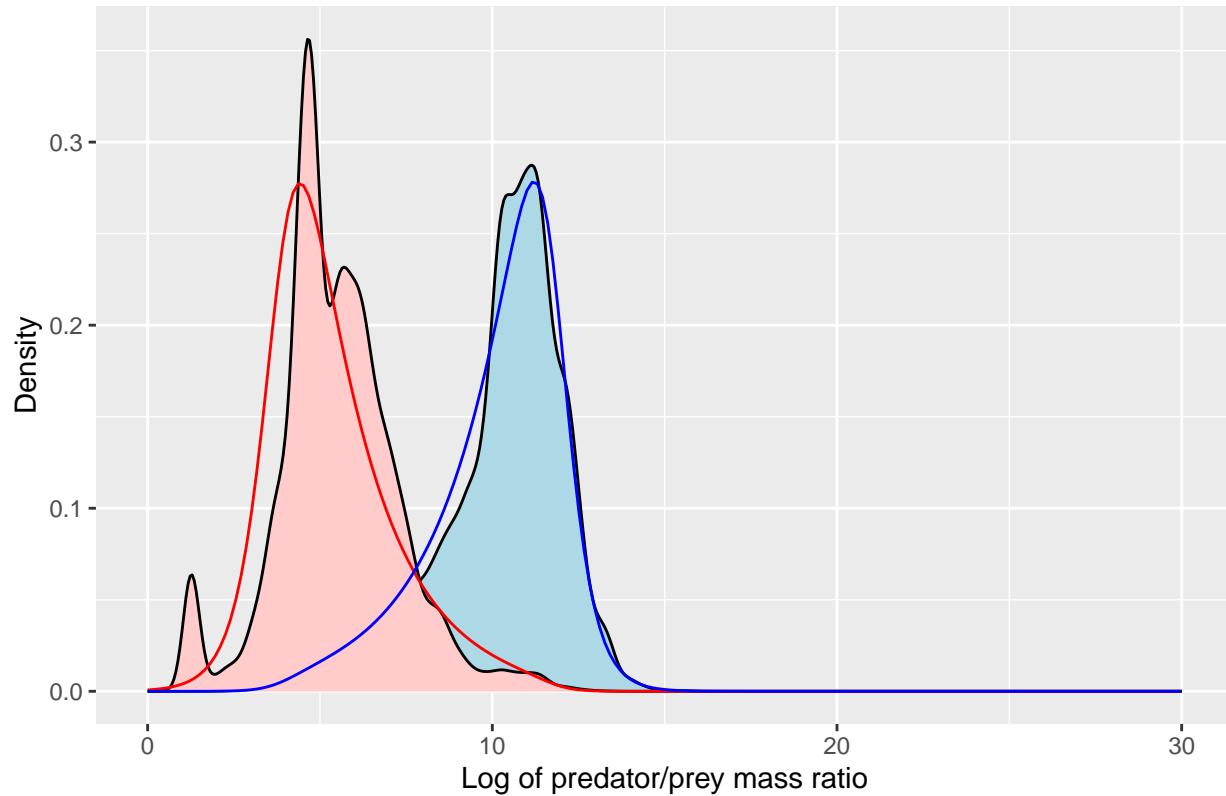
```
numberestco <- biomassco

dist <- dexp(grid, alpha = (numberestco[1]), ll = numberestco[2], ul = numberestco[3], lr = numberestco[4])
numberdist <- data.frame(l=grid, Density=dist)
dist <- dexp(grid, alpha = (numberestco[1]-1), ll = numberestco[2], ul = numberestco[3], lr = numberestco[4])
biomassdist <- data.frame(l=grid, Density=dist)
#now plot these two together

stomach <- stomach %>% mutate(biomass = nprey_perpred * prey_ind_weight_g)

ggplot(stomach) +
  geom_density(aes(l, weight=weight_numbers), fill="lightblue")+
  geom_density(aes(l, weight=biomass), fill="#ffcccb")+
  xlab("Log of predator/prey mass ratio") +
  geom_line(aes(l, Density), data = biomassdist, color = "red")+
  geom_line(aes(l, Density), data = numberdist, color = "blue")+
  ggtitle("Fitted to number, shifted to biomass")
```

## Fitted to number, shifted to biomass



This fits well.

```
alpha      ll      ul      lr      ur
```

```
0.4805395 3.8865861 2.1827656 11.7890770 2.3939047
```

Does fitting to biomass and shifting mean a better fit?

```
stomach <- stomach%>%mutate(l=log(ppmr))

stomach <- stomach%>%mutate(weight_numbers = stomach$nprey_perpred*
                           stomach$prey_ind_weight_g^dig)

est <- mle_texp(stomach)

biomassco <- est@coef

grid = seq(0, 30, length.out = 200)
#here, the alpha is meant to be -1, but I have to subtract 0.7 to make it work, so I am going to run the
#for the biomass, and see the difference
dist <- dexp(grid, alpha = (biomassco[1]), ll = biomassco[2], ul = biomassco[3],
             lr = biomassco[4], ur = biomassco[5])

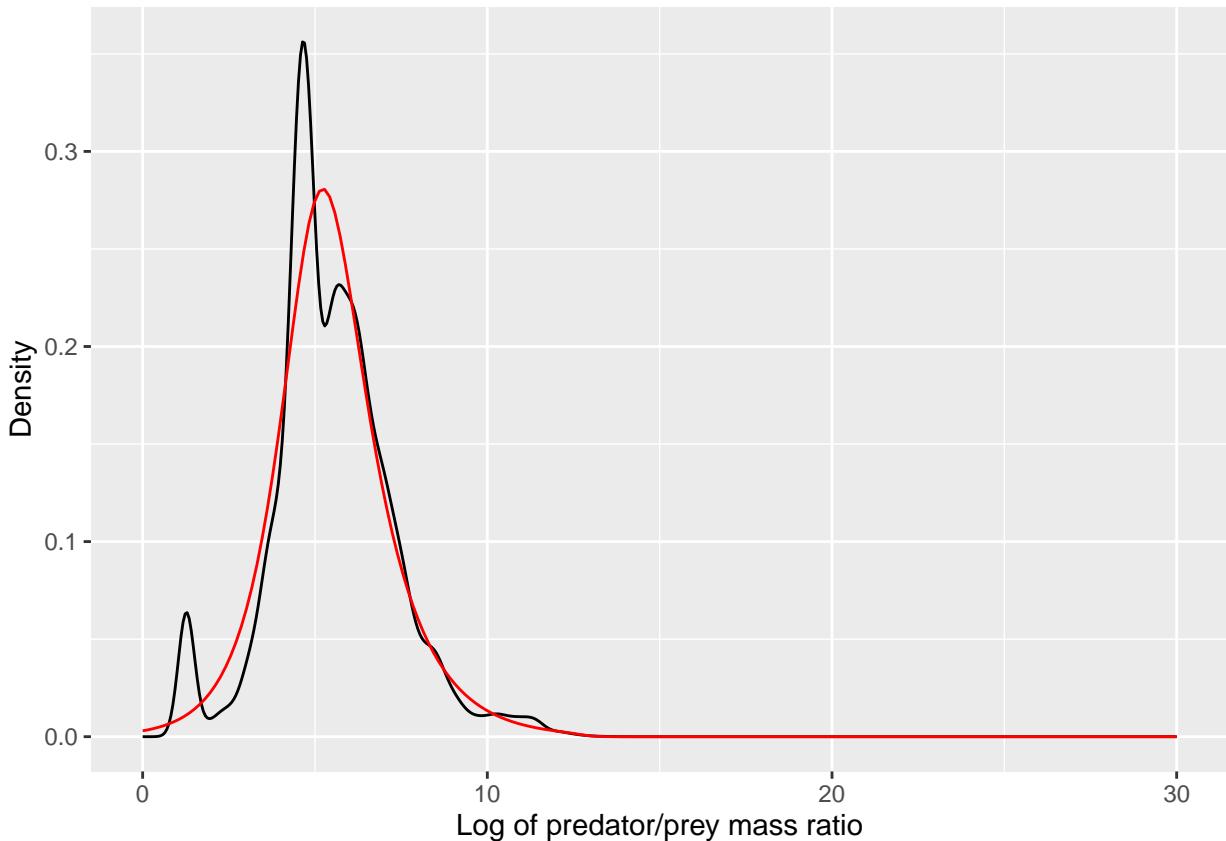
dist <- data.frame(l=grid, Density=dist)

ggplot(stomach) +
```

```

geom_density(aes(l, weight=weight_numbers))+
xlab("Log of predator/prey mass ratio") +
geom_line(aes(l, Density), data = dist, color = "red")

```



```

numberestco <- biomassco

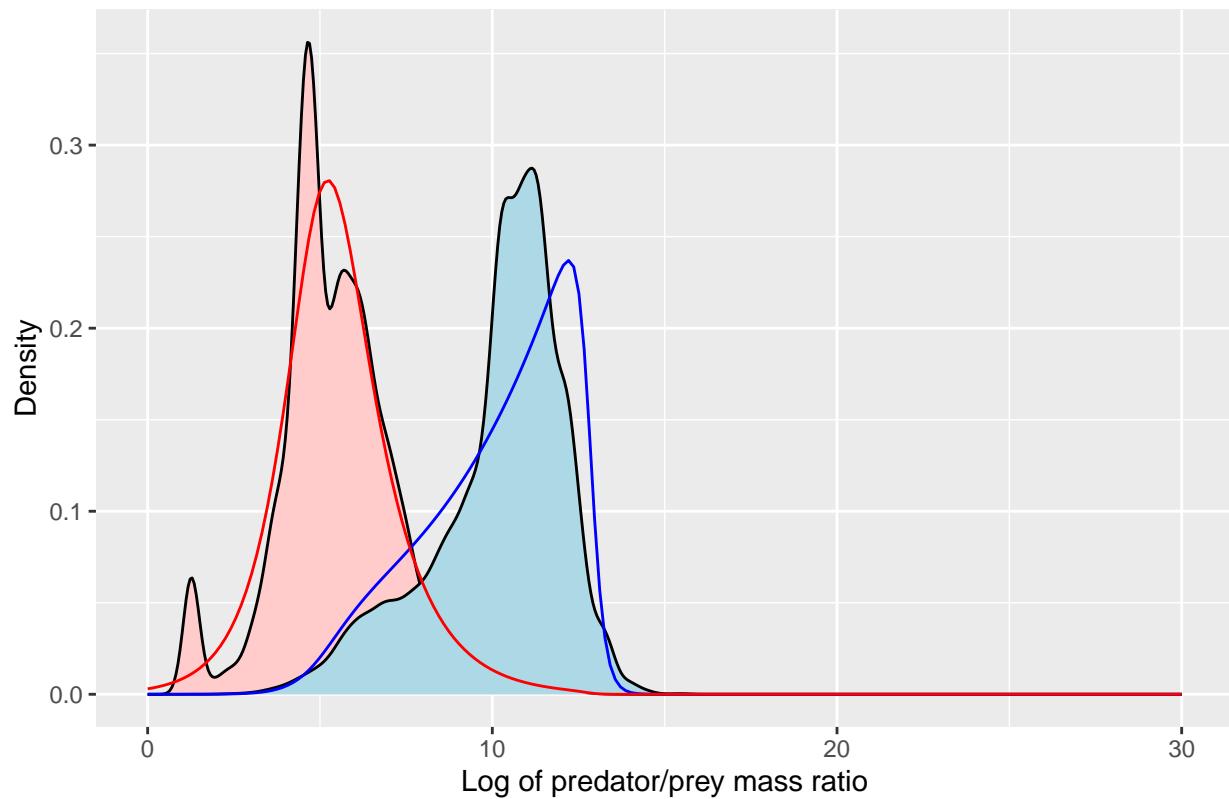
dist <- dexp(grid, alpha = (numberestco[1]), ll = numberestco[2], ul = numberestco[3], lr = numberestco[4])
numberdist <- data.frame(l=grid, Density=dist)
dist <- dexp(grid, alpha = (numberestco[1]+1), ll = numberestco[2], ul = numberestco[3], lr = numberestco[4])
biomassdist <- data.frame(l=grid, Density=dist)
#now plot these two together

stomach <- stomach %>% mutate(biomass = nprey_perpred)

ggplot(stomach) +
  geom_density(aes(l, weight=weight_numbers), fill="#ffcccb")+
  geom_density(aes(l, weight=biomass), fill="lightblue")+
  xlab("Log of predator/prey mass ratio") +
  geom_line(aes(l, Density), data = biomassdist, color = "blue")+
  geom_line(aes(l, Density), data = numberdist, color = "red")+
  ggtitle("Fitted to Biomass, shifted to number")

```

Fitted to Biomass, shifted to number



They are both the same - both underestimate either number or biomass. Fitting to the number density then shifting to the biomass fits the best however, so the previous density function is best.