

# Blue Whiting

2024-07-21

```
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
```

```
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
```

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

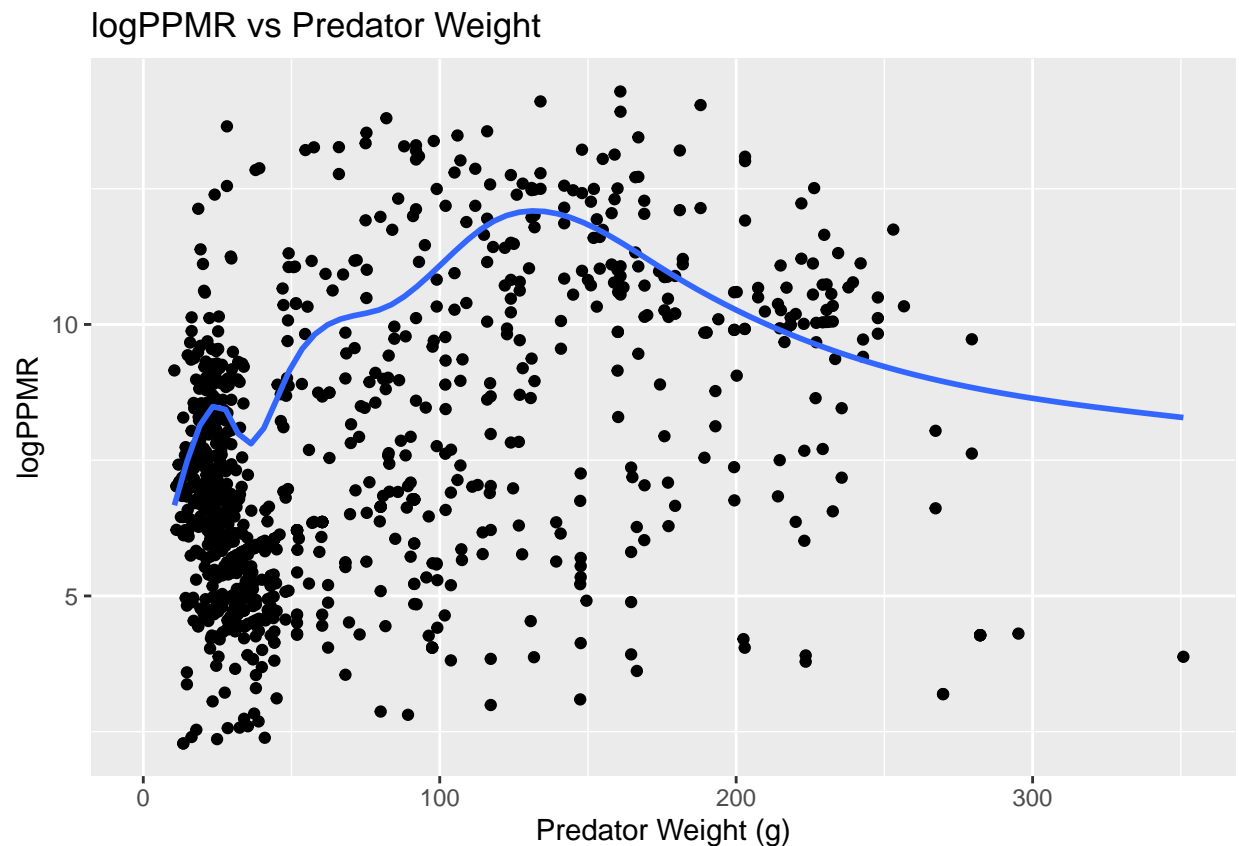
```
sprat <- stom_df %>% filter(pred_taxa=="Micromesistius poutassou")
```

```
ggplot(sprat, aes(x=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="Predator Weight (g)", y="logPPMR")
```

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

## Warning: Removed 43 rows containing non-finite outside the scale range
## ('stat_smooth()').

## Warning: Removed 43 rows containing missing values or values outside the scale range
## ('geom_point()').
```



```
dig <- 1

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

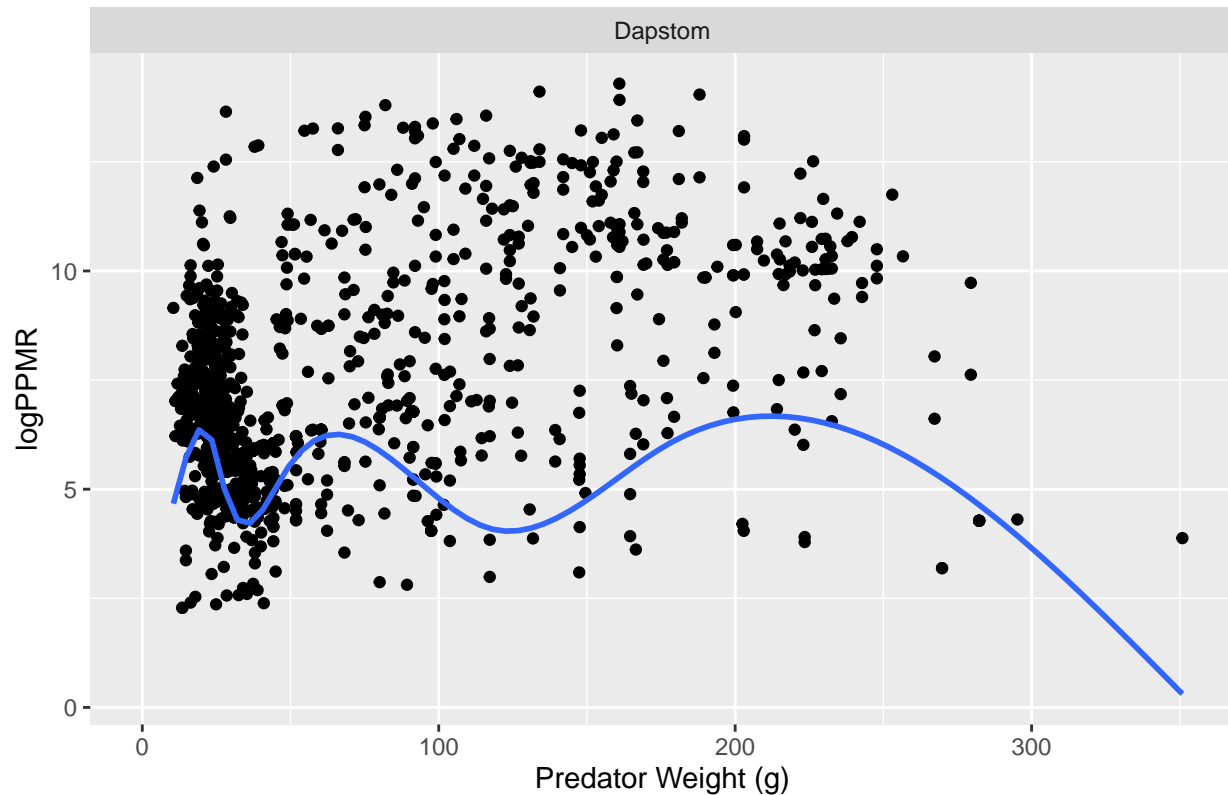
```
## Warning: Use of 'sprat$prey_ind_weight_g' is discouraged.
## i Use 'prey_ind_weight_g' instead.

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

## Warning: Removed 43 rows containing non-finite outside the scale range
## ('stat_smooth()').
```

```
## Warning: Removed 43 rows containing missing values or values outside the scale range
## ('geom_point()').
```

## logPPMR vs Predator Weight



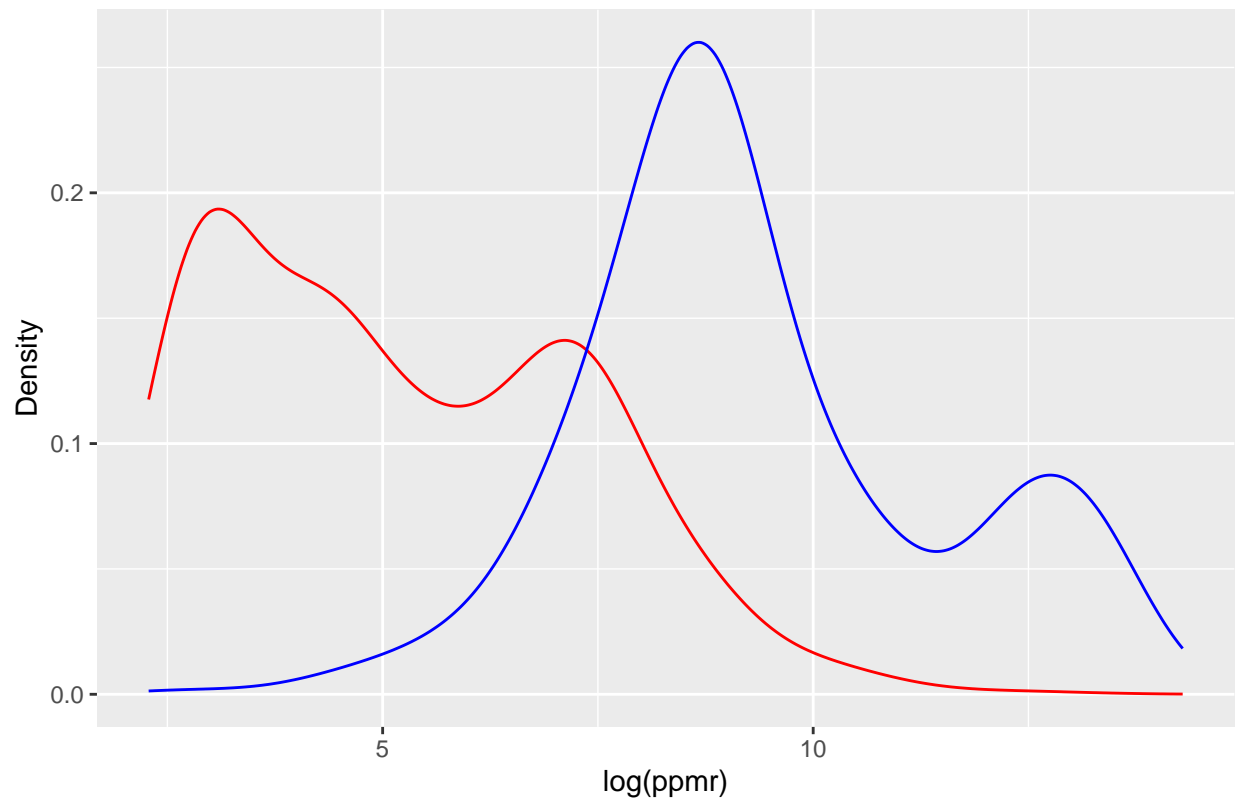
The PPMR fluctuates around 5, which I think is good enough to state that PPMR is constant across predator sizes.

```
sprat$weight_numbers <- sprat$nprey_perpred
sprat$weight_biomass <- sprat$nprey_perpred*sprat$prey_ind_weight_g^dig

ggplot() +
  geom_density(data = sprat, aes(x = log(ppmr), weight = weight_biomass), color = "red") +
  geom_density(data = sprat, 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")
```

```
## Warning: Removed 43 rows containing non-finite outside the scale range
## ('stat_density()').
## Removed 43 rows containing non-finite outside the scale range
## ('stat_density()').
```

Density Plot of  $\log(\text{ppmr})$  Weighted by Biomass and Numbers

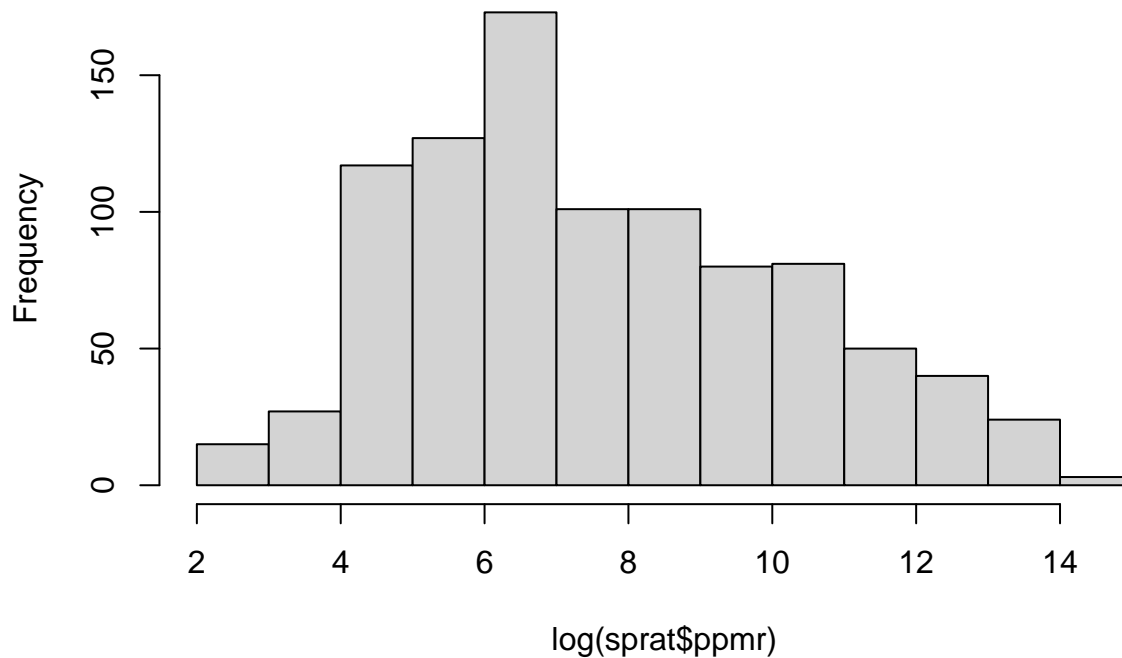


This looks like a double gaussian?

It really doesn't look right that it the biggest peak for the density when weigted by biomass is at nearly 0 PPMR.

```
hist(log(sprat$ppmr))
```

## Histogram of log(sprat\$ppmr)



```
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)  
}
```

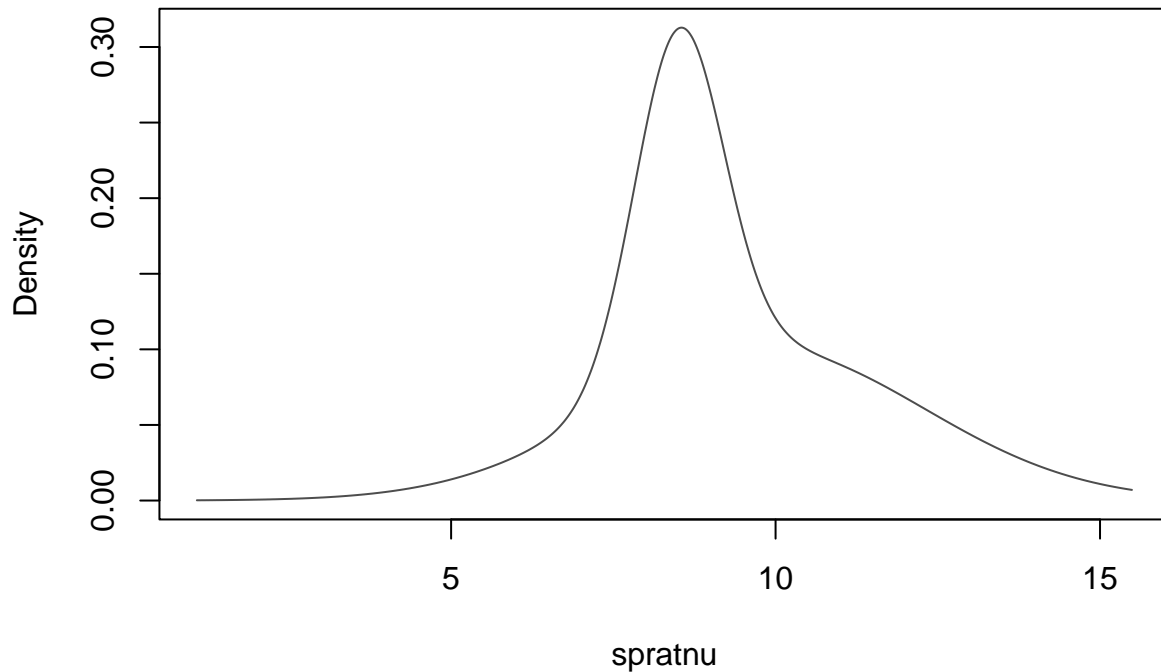
```

}

spratnu <- repeat_elements(log(sprat$ppmr), sprat$npred_perpred)

gmm <- densityMclust(spratnu, G=2)

```



```

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

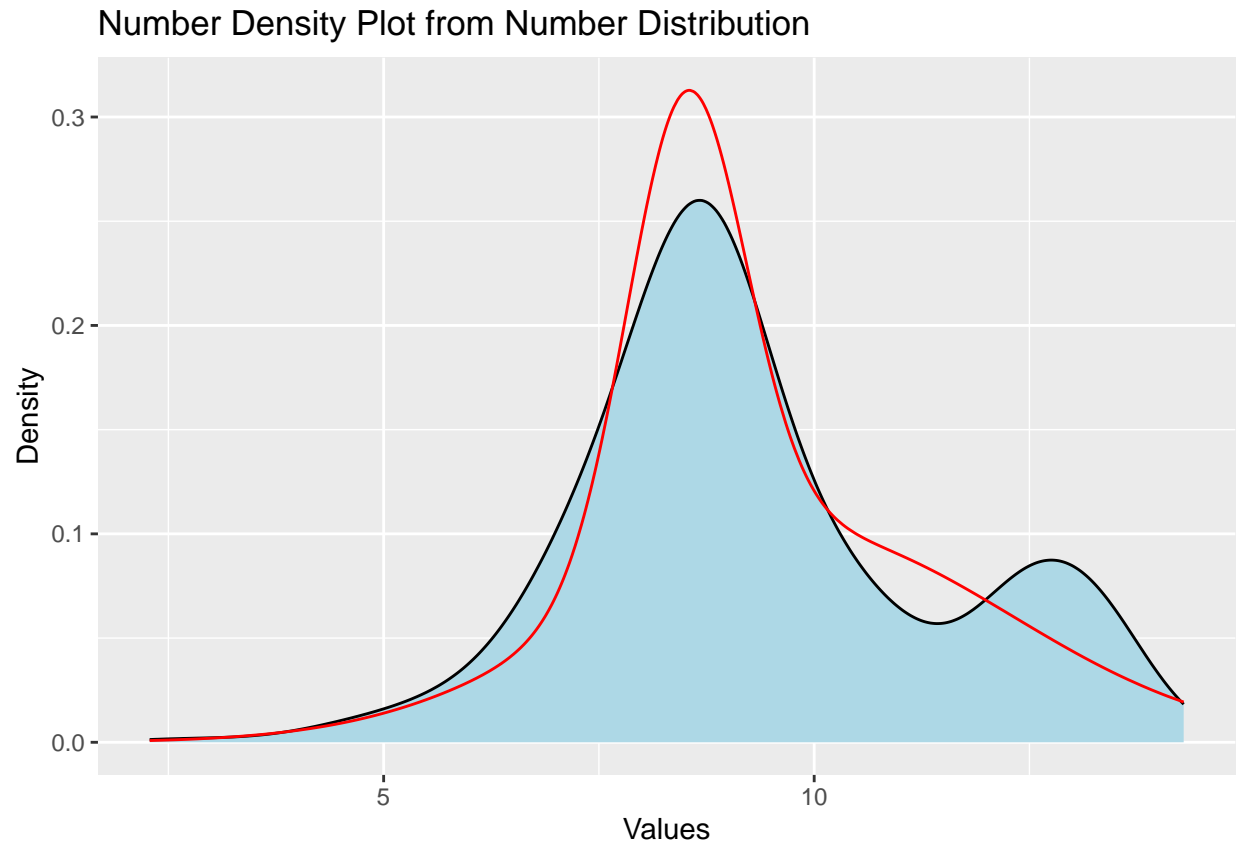
(numbfitt <- ggplot() +
  geom_density(data=sprat, aes(log(ppmr), weight=weight_numbers), fill="lightblue")+
  geom_line(data=dplot, aes(x = x, y = density), color="red") +
  labs(x = "Values", y = "Density") +
  ggtitle("Number Density Plot from Number Distribution"))

```

```

## Warning: Removed 43 rows containing non-finite outside the scale range
## ('stat_density()').

```



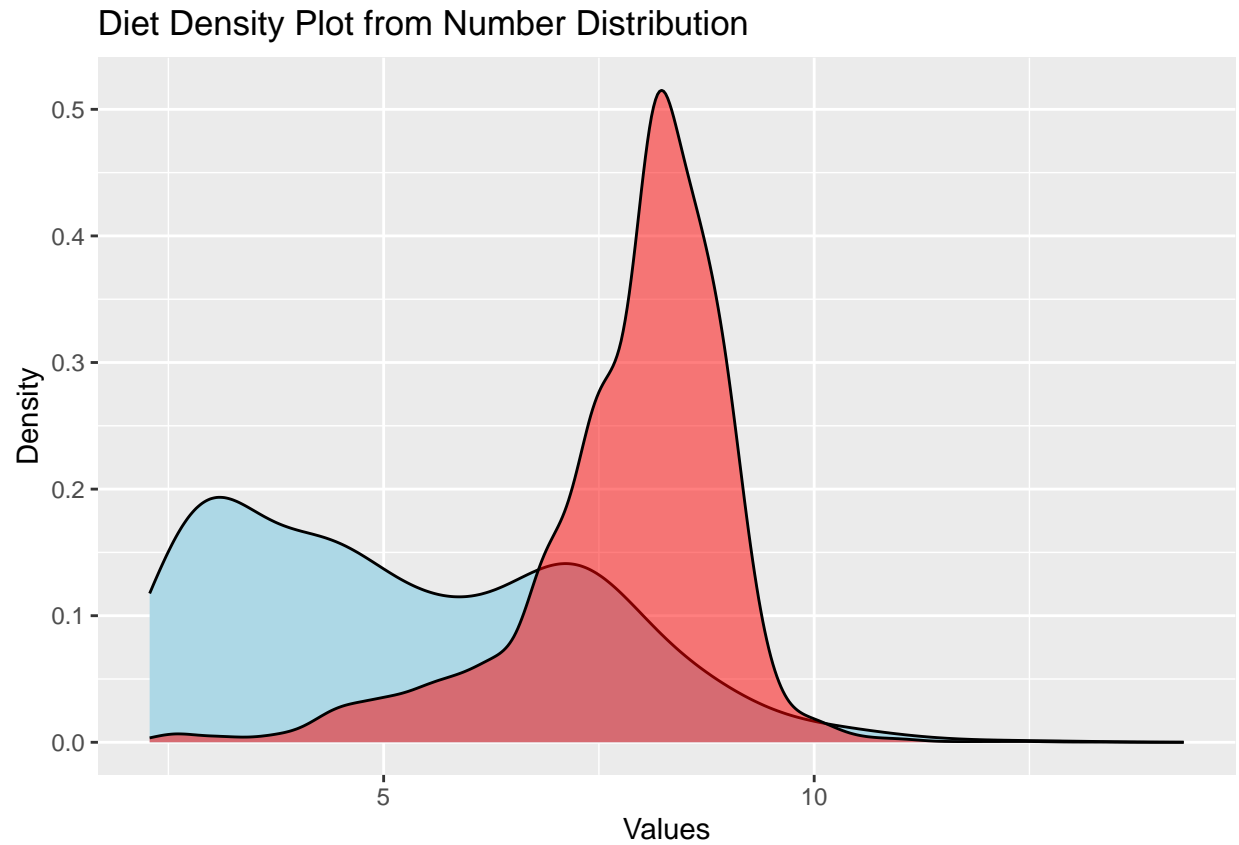
It is pretty good.

```
newspratdens <- gmm[["density"]]*exp(-dig*gmm[["data"]])
shifted_pdf_normalized <- newspratdens / sum(newspratdens)

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

(numbfitebio <- ggplot() +
  geom_density(data=sprat, 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"))
```

```
## Warning: Removed 43 rows containing non-finite outside the scale range
## ('stat_density()').
```

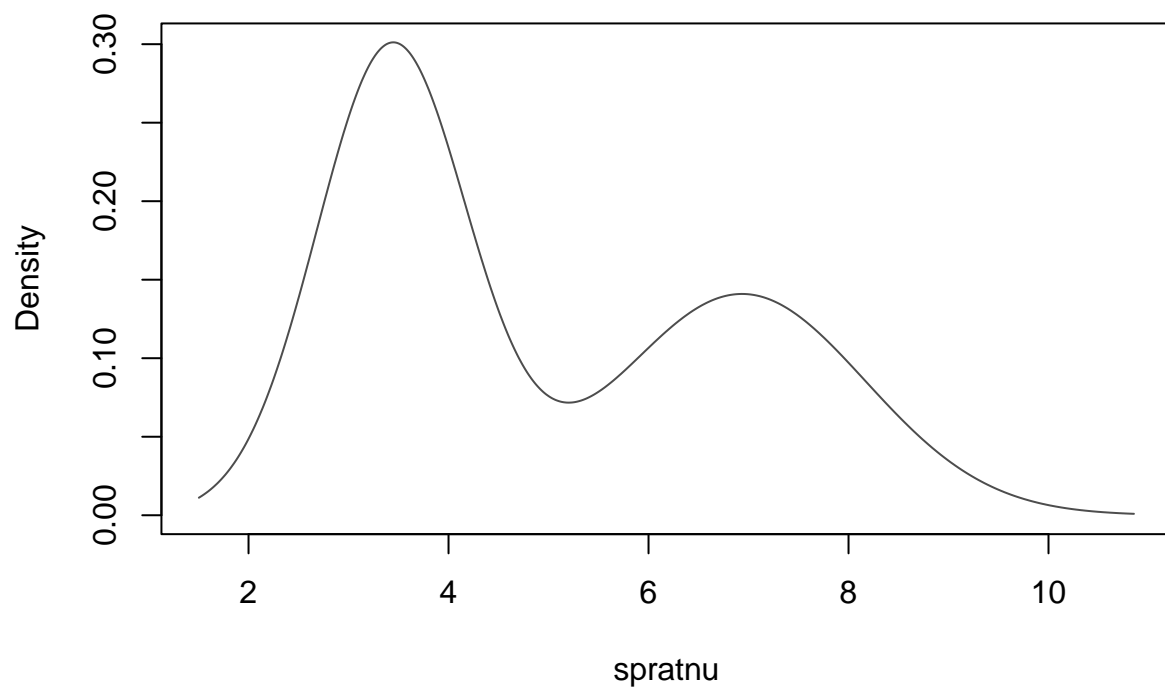


This Isn't a good shift.

Will try the other way around.

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



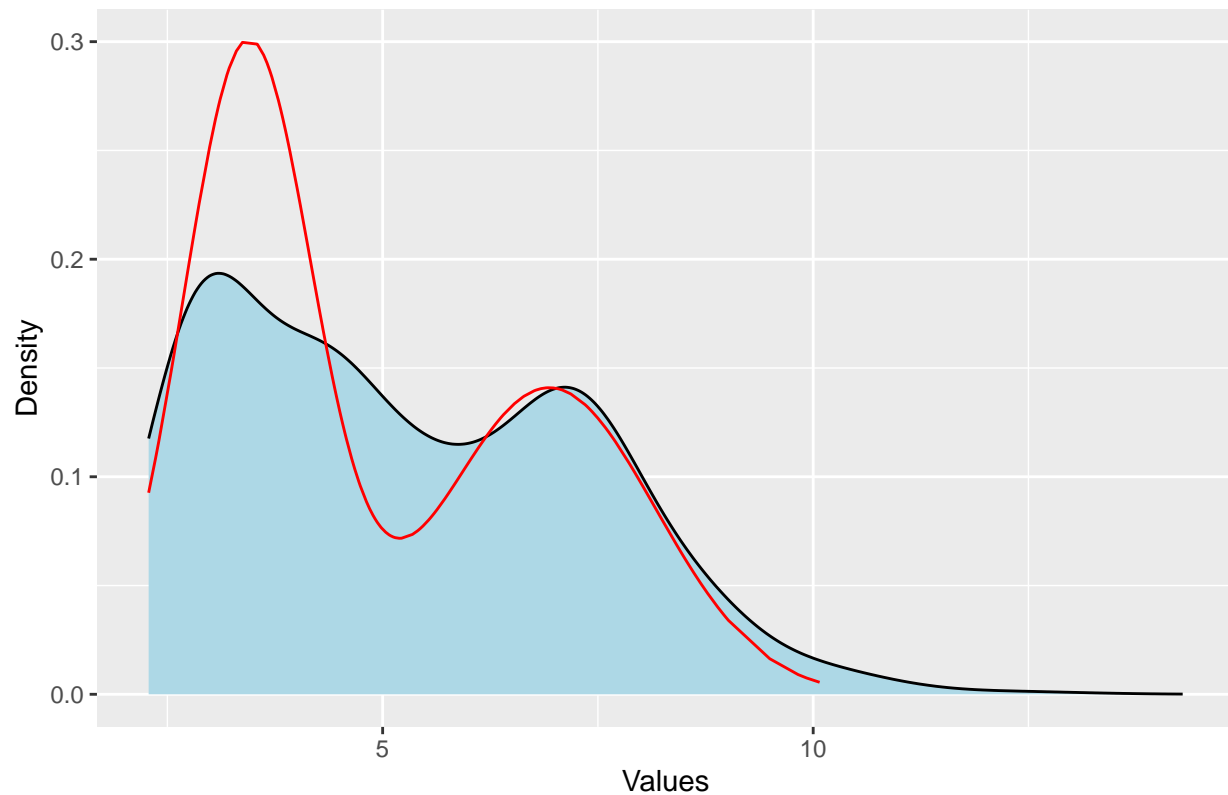


```
dplot <- data.frame(x=gmm[["data"]], density=gmm[["density"]])

(biofit <- ggplot() +
  geom_density(data=sprat, 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"))
```

```
## Warning: Removed 43 rows containing non-finite outside the scale range
## ('stat_density()').
```

Diet Density Plot from Diet Distribution

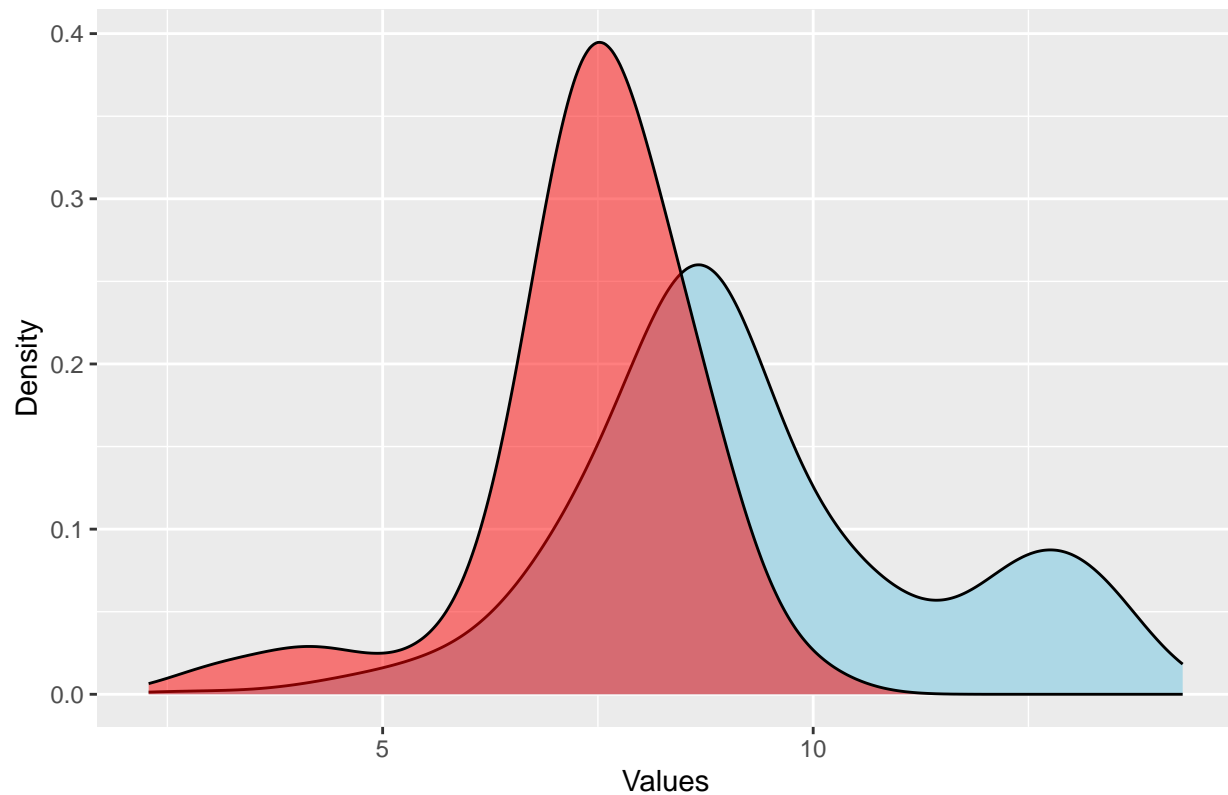


```
newspratdens <- gmm[["density"]]*exp(dig*gmm[["data"]])
shifted_pdf_normalized <- newspratdens / sum(newspratdens)
dplot <- data.frame(x=gmm[["data"]], density=shifted_pdf_normalized)

(numbfittbio <- ggplot() +
  geom_density(data=sprat, 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 Plot from Number Distribution"))

## Warning: Removed 43 rows containing non-finite outside the scale range
## ('stat_density()').
```

Diet Density Plot from Number Distribution



```
sprat$l <- log(sprat$ppmr)
sprat <- sprat[!is.na(sprat$l),]
x_vals <- seq(min(sprat$l), max(sprat$l), length.out = 1000)

#I dont think I have done it right here, so I will do it in another way

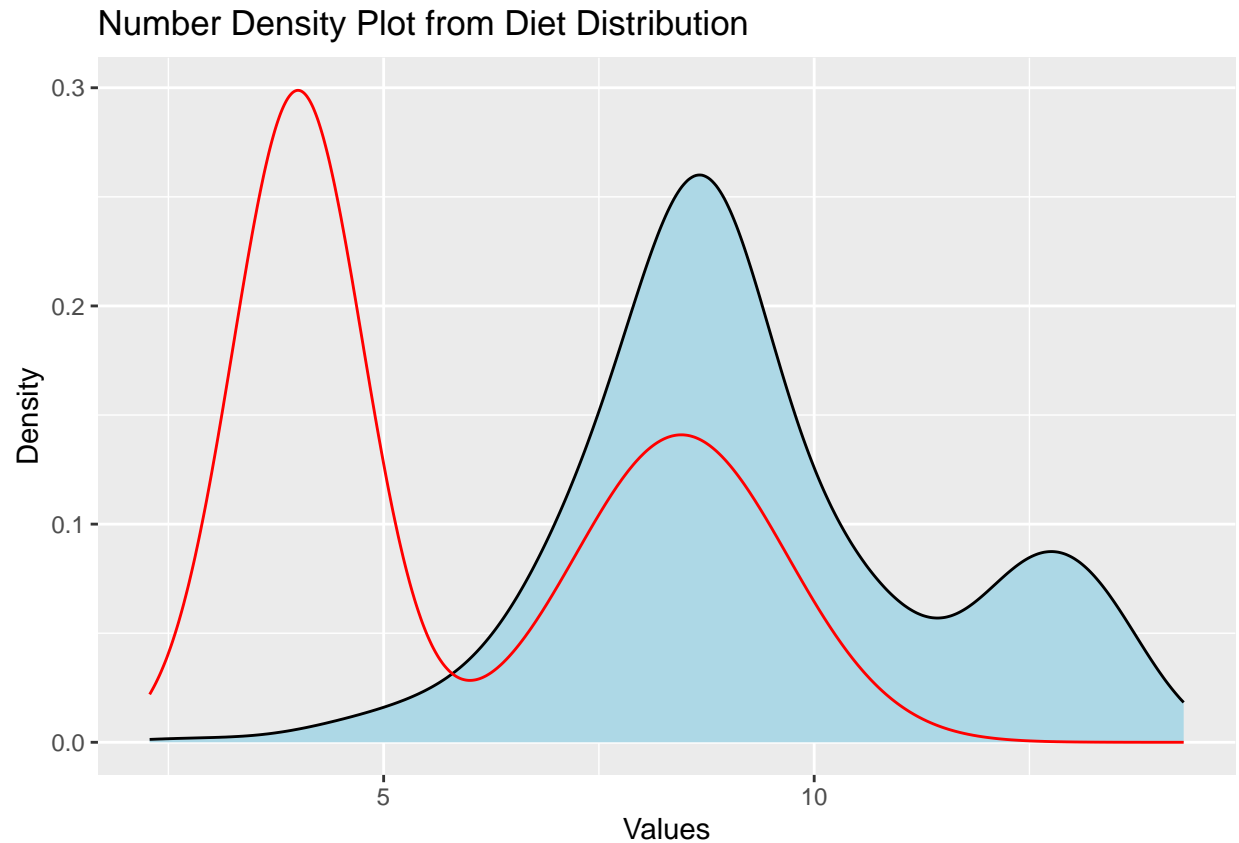
shifted_fit <- gmm
shifted_fit[["parameters"]][["mean"]] <- shifted_fit[["parameters"]][["mean"]] +
  (1)*shifted_fit[["parameters"]][["variance"]][["sigmasq"]]

#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$pro * shifted_fit$parameters$variance$sigmasq)))
})

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

(biofitnum <- ggplot() +
  geom_density(data=sprat, aes(log(sprat$ppmr), weight=weight_numbers), fill="lightblue")+
  geom_line(data=plot_data, aes(x = x_vals, y = shifted_pdf), color="red") +
  labs(x = "Values", y = "Density") +
  ggtitle("Number Density Plot from Diet Distribution"))
```

```
## Warning: Use of 'sprat$ppmr' is discouraged.
## i Use 'ppmr' instead.
```



It is better, but still not great.

Will try the exponential

```
stomach <- sprat
stomach$l <- log(stomach$ppmr)
stomach <- stomach[!is.na(stomach$l),]
stomach <- stomach %>% mutate(weight_numbers = nprey_perpred)

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

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

  return(fl_values)
}

## Define the truncated exponential PDF with debugging
dtxp <- function(l, alpha, ll, ul, lr, ur) {
  fl_values <- fl(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 <- dtemp(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)
}

est <- mle_texp(stomach)

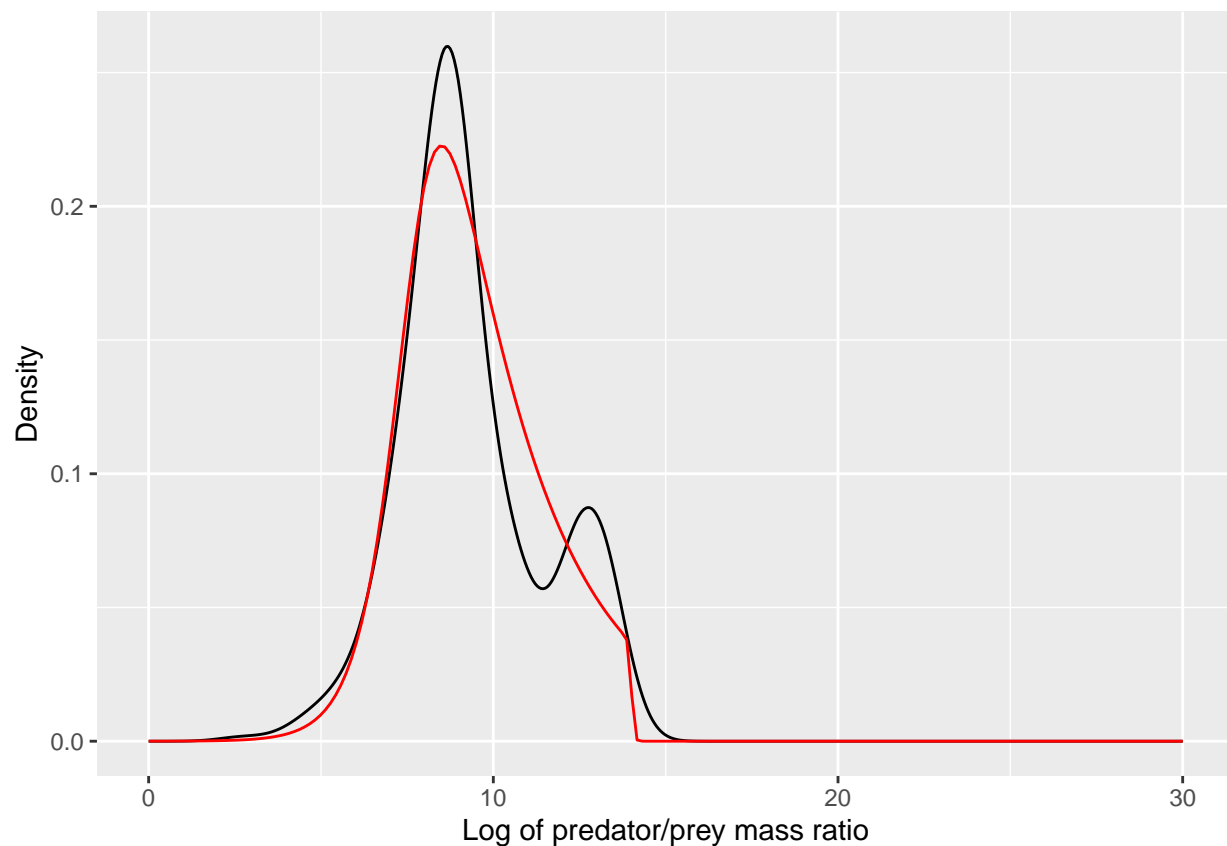
biomassco <- est@coef

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

```



Lets try to plot both.

```

stomach <- stomach %>% mutate(weight_numbers = nprey_perpred * prey_ind_weight_g)
est <- mle_texp(stomach)
biomassestco <- est@coef

```

```

stomach <- stomach %>% mutate(weight_numbers = nprey_perpred)
est <- mle_texp(stomach)
numberestco <- est@coef

grid = seq(0, 30, length.out = 200)
dist <- dtexp(grid, alpha = (biomassestco[1]), ll = biomassestco[2], ul = biomassestco[3], lr = biomassestco[4])
biomassdist <- data.frame(l=grid, Density=dist)
shiftdist <- dtexp(grid, alpha = (biomassestco[1]+1), ll = biomassestco[2], ul = biomassestco[3], lr = biomassestco[4])
shiftbiomassdist <- data.frame(l=grid, Density=shiftdist)

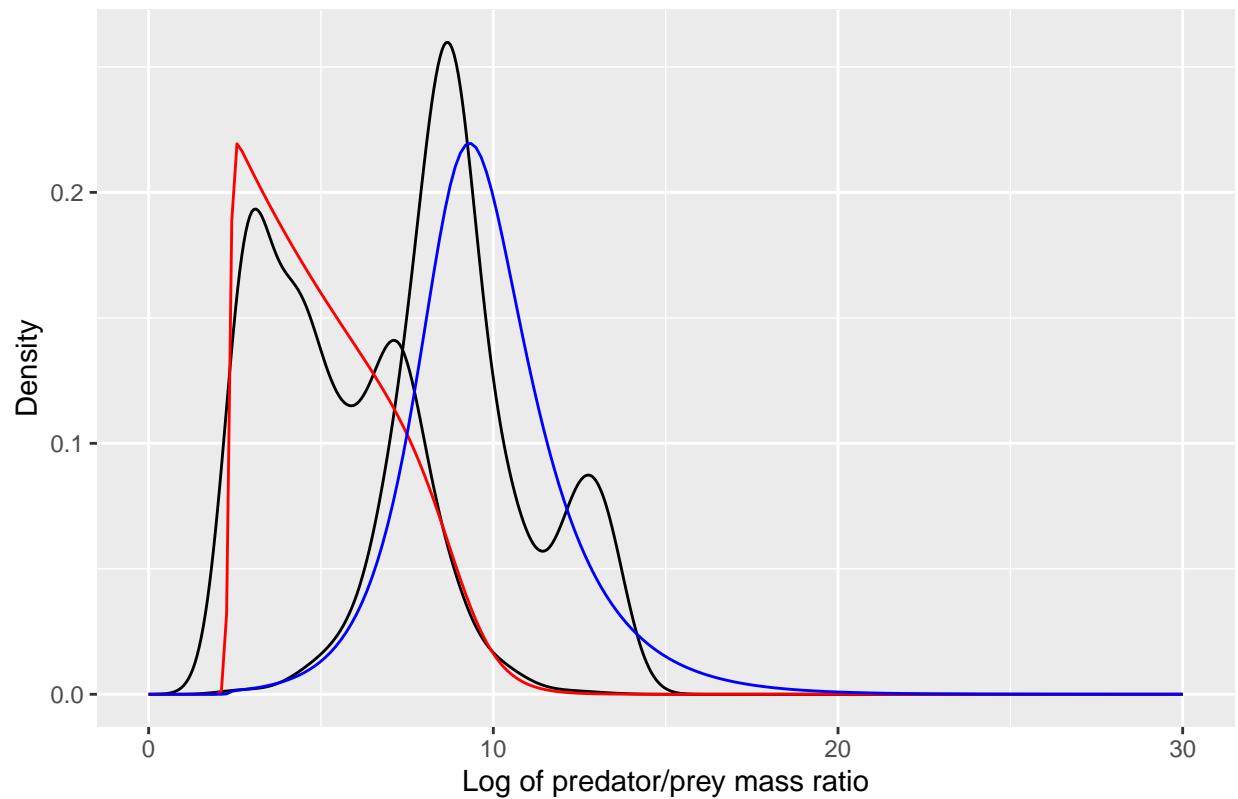
dist <- dtexp(grid, alpha = (numberestco[1]), ll = numberestco[2], ul = numberestco[3], lr = numberestco[4])
numberdist <- data.frame(l=grid, Density=dist)
shiftdist <- dtexp(grid, alpha = (numberestco[1]-1), ll = numberestco[2], ul = numberestco[3], lr = numberestco[4])
shiftnumberdist <- data.frame(l=grid,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

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

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

```

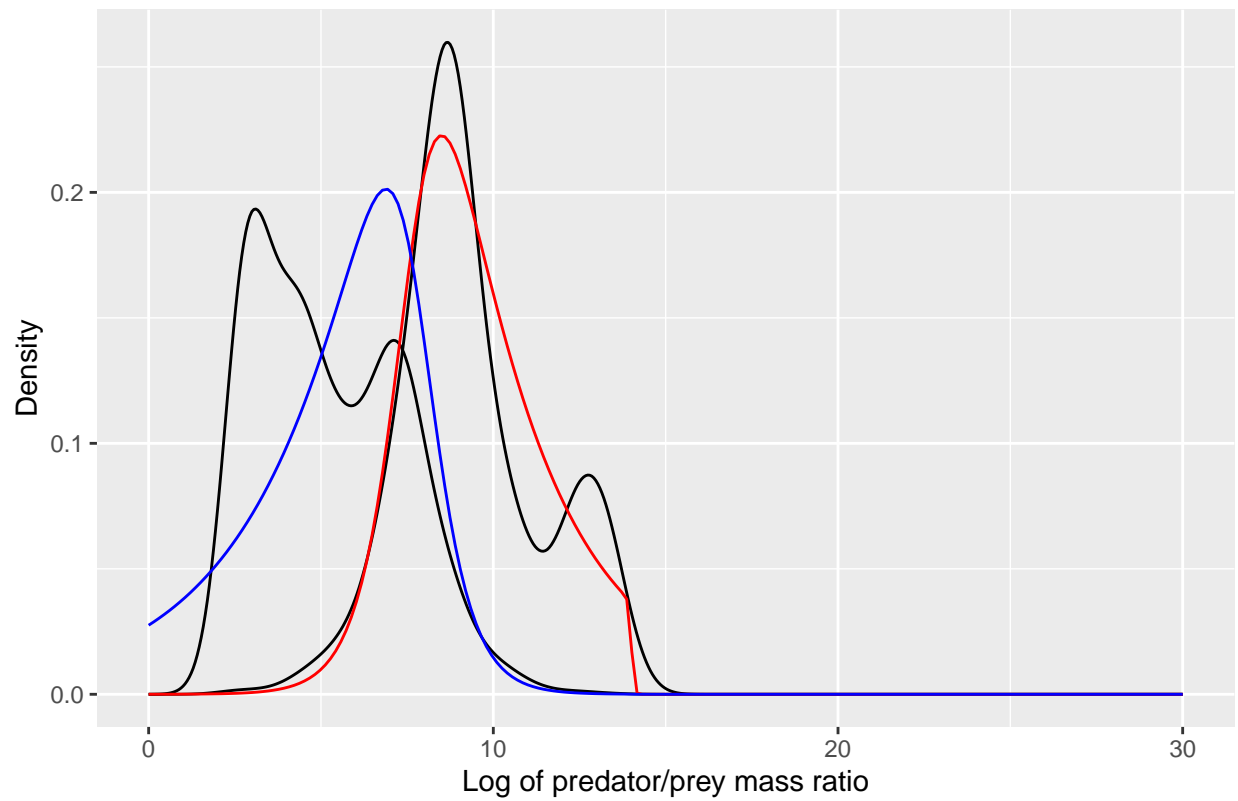
Fitted to Biomass, shift to number



```
ggplot(stomach) +  
  geom_density(aes(l, weight=weight_numbers))+  
  geom_density(aes(l, weight=biomass))+  
  xlab("Log of predator/prey mass ratio") +  
  geom_line(aes(l, Density), data = numberdist, color = "red")+  
  geom_line(aes(l, Density), data = shiftnumberdist, color = "blue")+  
  ggtitle("Fitted to NUMBER, shift to Biomass")
```



Fitted to NUMBER, shift to Biomass



Fitting to biomass works well for hte exponential

```
print(biomassestco)
```

```
##      alpha      ll      ul      lr      ur
## -0.1310297  2.3405346 22.9956791  9.0047846  1.4277256
```