

Kernel - Norway Pout

2024-07-12

```
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=="Trisopterus esmarkii")
```

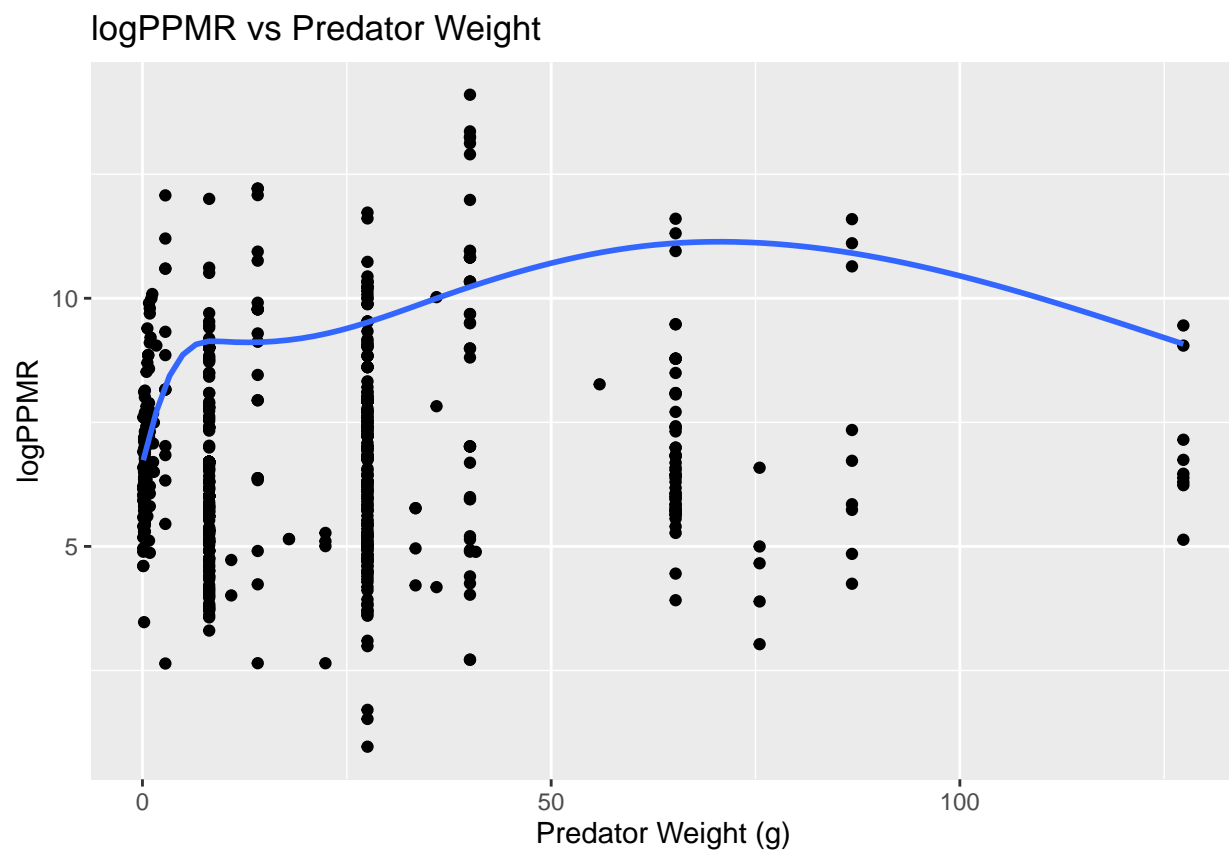
First lets check the PPMR predator size distribution.

```
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 9 rows containing non-finite outside the scale range
## ('stat_smooth()').

## Warning: Removed 9 rows containing missing values or values outside the scale range
## ('geom_point()').
```



Not good. Will now plot the same but with the diet weighting

```
dig <- 1

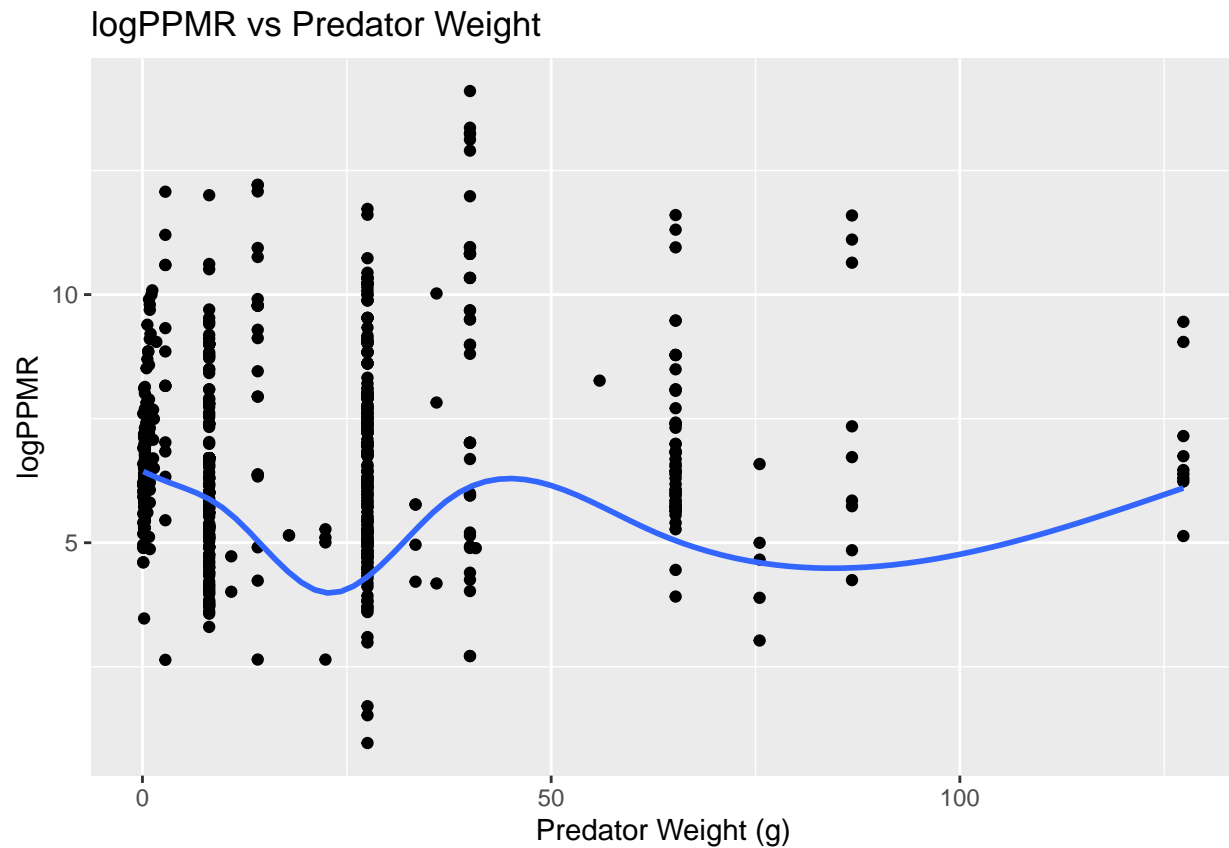
ggplot(sprat, aes(x=pred_weight_g, y=log(ppmr)))+
  geom_point()+
  geom_smooth(method="gam", se=FALSE, aes(weight = nprey_perpred*sprat$prey_ind_weight_g^dig))+
  labs(title="logPPMR vs Predator Weight", x="Predator Weight (g)", y="logPPMR")

## 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 9 rows containing non-finite outside the scale range
## ('stat_smooth()').

## Warning: Removed 9 rows containing missing values or values outside the scale range
## ('geom_point()').
```



When looking at the contribution to the diet, there is variation around a PPMR value, but this might be to do with the groupings of the pred weights, so I think it is safe to say that PPMR is the same across all pred sizes.

So first, let's look at the density to see what might best fit.

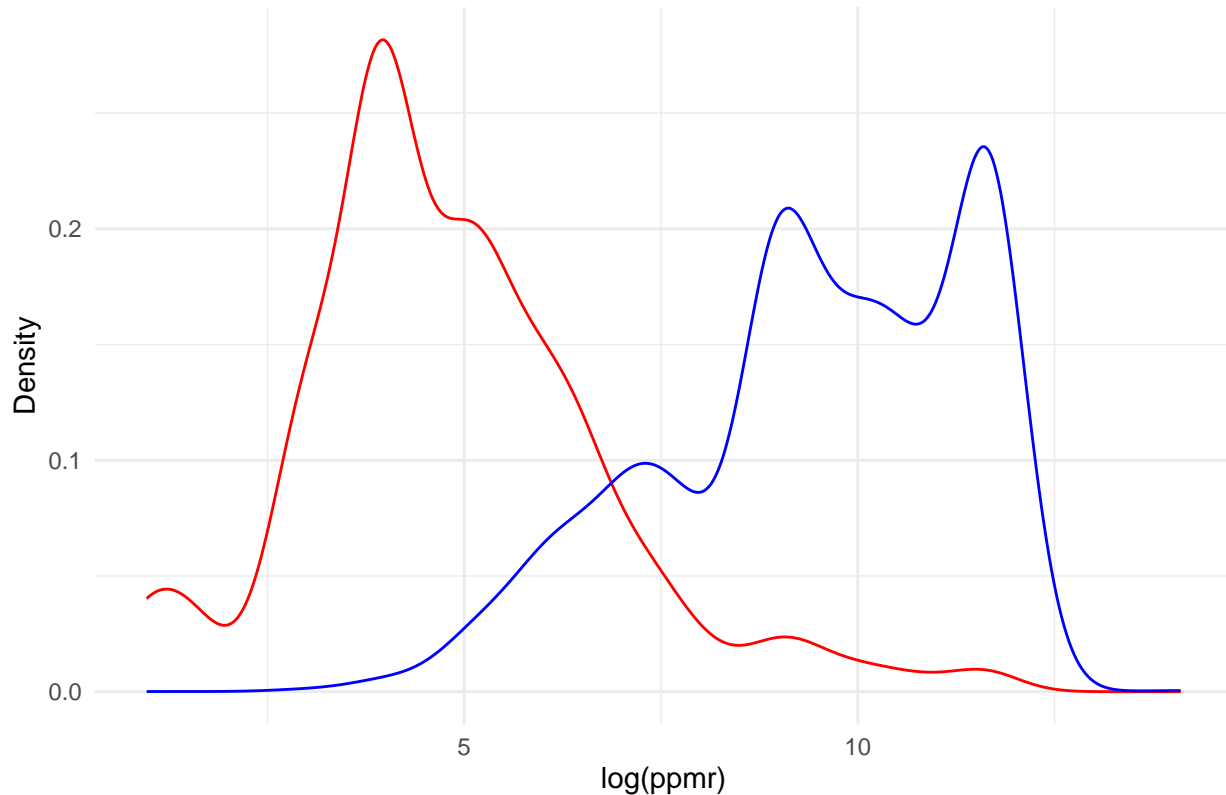
```
sprat$weight_numbers <- sprat$nprey_perpred
sprat$weight_biomass <- sprat$nprey_perpred*sprat$prey_ind_weight_g^dig

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") +
  theme_minimal()
```

```
## Warning: Removed 9 rows containing non-finite outside the scale range
```

```
## ('stat_density()').
## Removed 9 rows containing non-finite outside the scale range
## ('stat_density()').
```

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



Not sure which fit would be best here. a mixture gaussian model with 2 gaussians will fit well.

The ecology of norway pout is a mixture of plantivore and piscovore. Therefore it might make sense there are 2 feeding modes, one for filtering, the other for active predation.

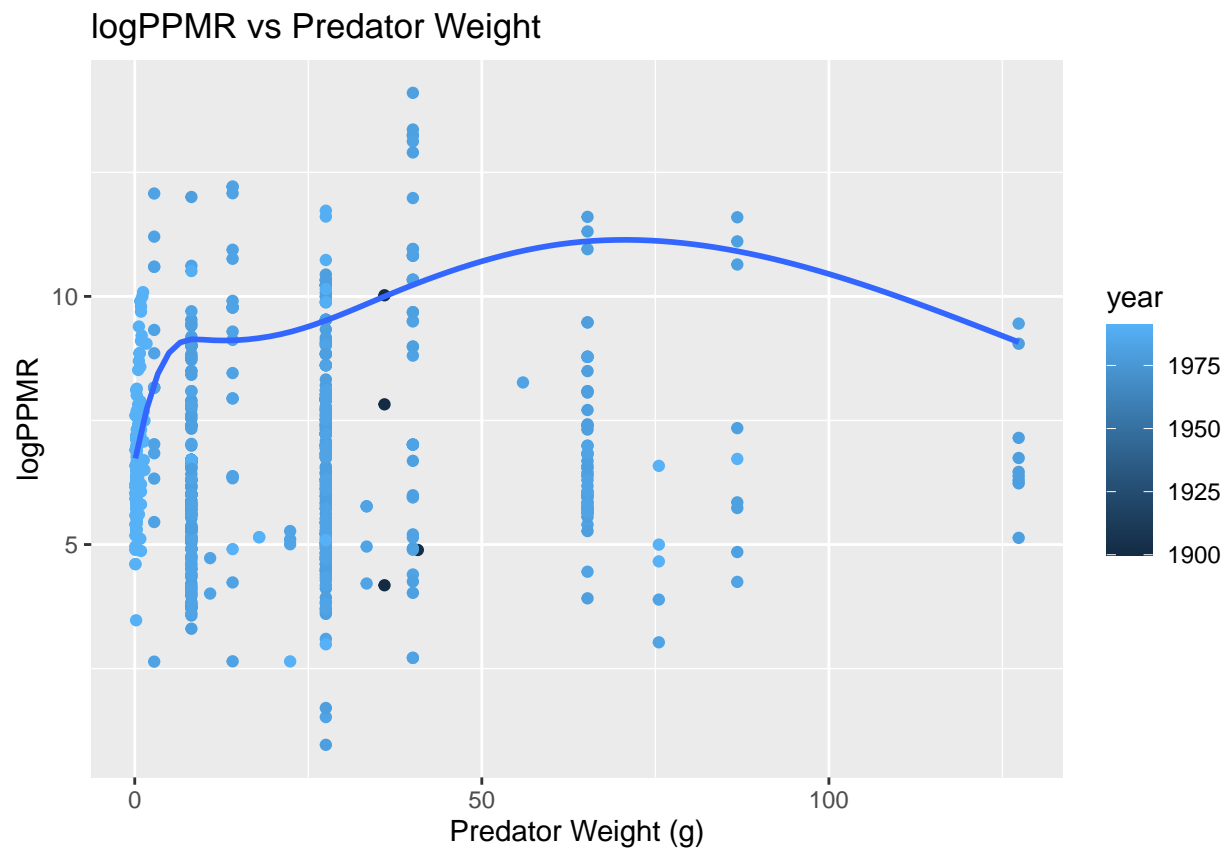
```
ggplot(sprat, aes(x=pred_weight_g, y=log(ppmr), color=year))+
  geom_point()+
  geom_smooth(method="gam", se=FALSE, aes(weight = nprey_perpred))+
  labs(title="logPPMR vs Predator Weight", x="Predator Weight (g)", y="logPPMR")
```

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

```
## Warning: Removed 9 rows containing non-finite outside the scale range
## ('stat_smooth()').
```

```
## Warning: The following aesthetics were dropped during statistical transformation:
## colour.
## i This can happen when ggplot fails to infer the correct grouping structure in
## the data.
## i Did you forget to specify a 'group' aesthetic or to convert a numerical
## variable into a factor?
```

```
## Warning: Removed 9 rows containing missing values or values outside the scale range
## ('geom_point()').
```

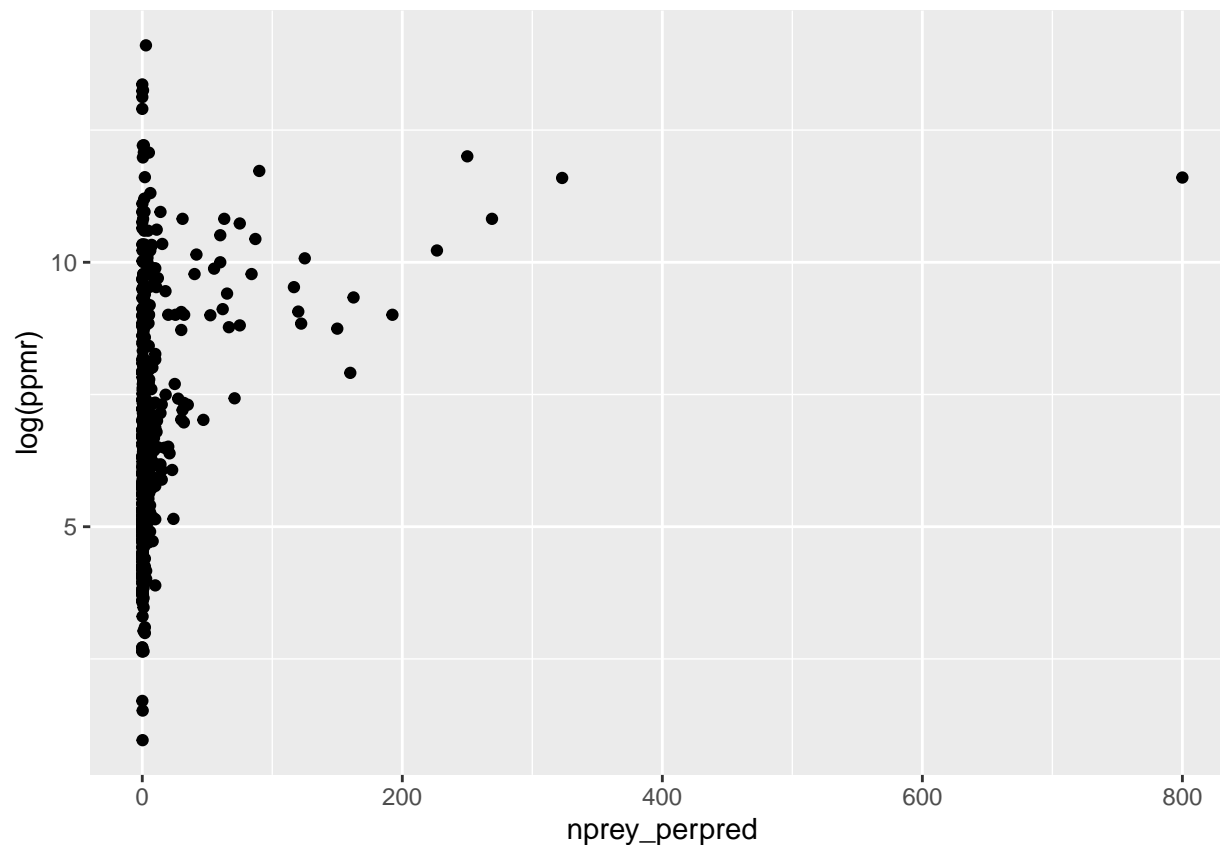


There doesn't seem to be a pattern in the years and PPMRs taken, so I assume that we can use all the data.

I am going to check now that there aren't any problem datapoints

```
ggplot()+
  geom_point(data=sprat, aes(x=npred_perpred, y=log(ppmr)))
```

```
## Warning: Removed 9 rows containing missing values or values outside the scale range
## ('geom_point()').
```



*#There is one point at around 12, which is a bit of an outlier. I will remove it.
#it changes the gaussians fitted too much.*

```
sprat <- sprat%>%filter(nprey_perpred<600)
```

```
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)

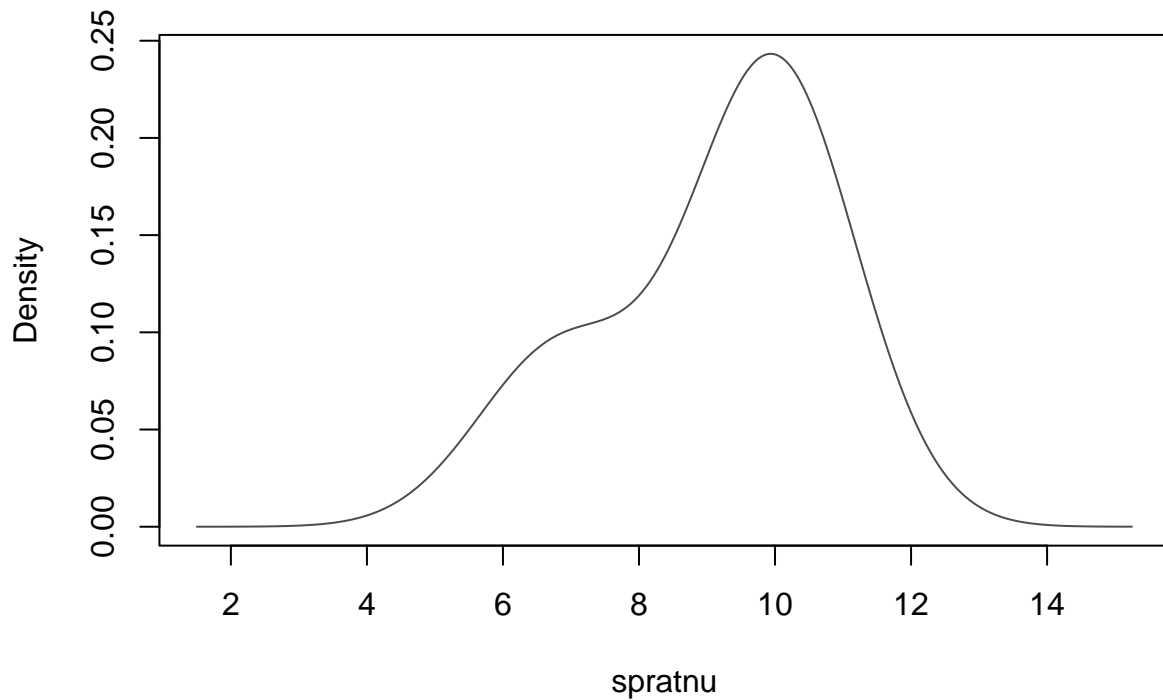
    final_vector <- c(final_vector, repeated_values)
  }

  return(final_vector)
}

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

gmm <- densityMclust(spratnu, G=2)

```



2 clusters fits nicely.

```

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

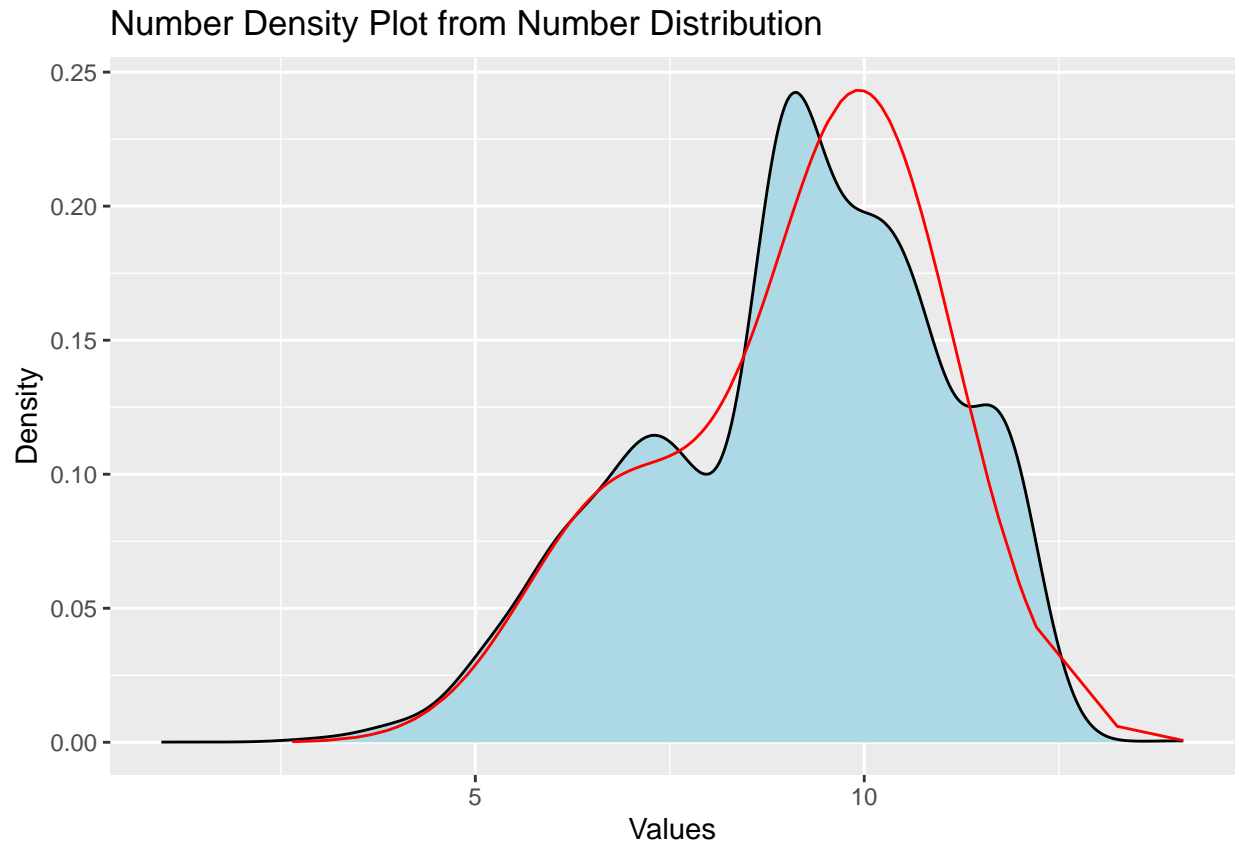
(numbfitt <- ggplot() +
  geom_density(data=sprat, aes(log(ppmr), weight=weight_numbers), fill="lightblue")+
  geom_line(data=dplot, aes(x = x, y = density), color="red") +
  labs(x = "Values", y = "Density") +
  ggtitle("Number Density Plot from Number Distribution"))

```

```

## Warning: Removed 9 rows containing non-finite outside the scale range
## ('stat_density()').

```



This is not bad, this works well. I will shift it over now.

```
#sorting out the column names and cleaning the data
sprat$l <- log(sprat$ppmr)
sprat <- sprat[!is.na(sprat$l),]
x_vals <- seq(min(sprat$l), max(sprat$l), length.out = 1000)

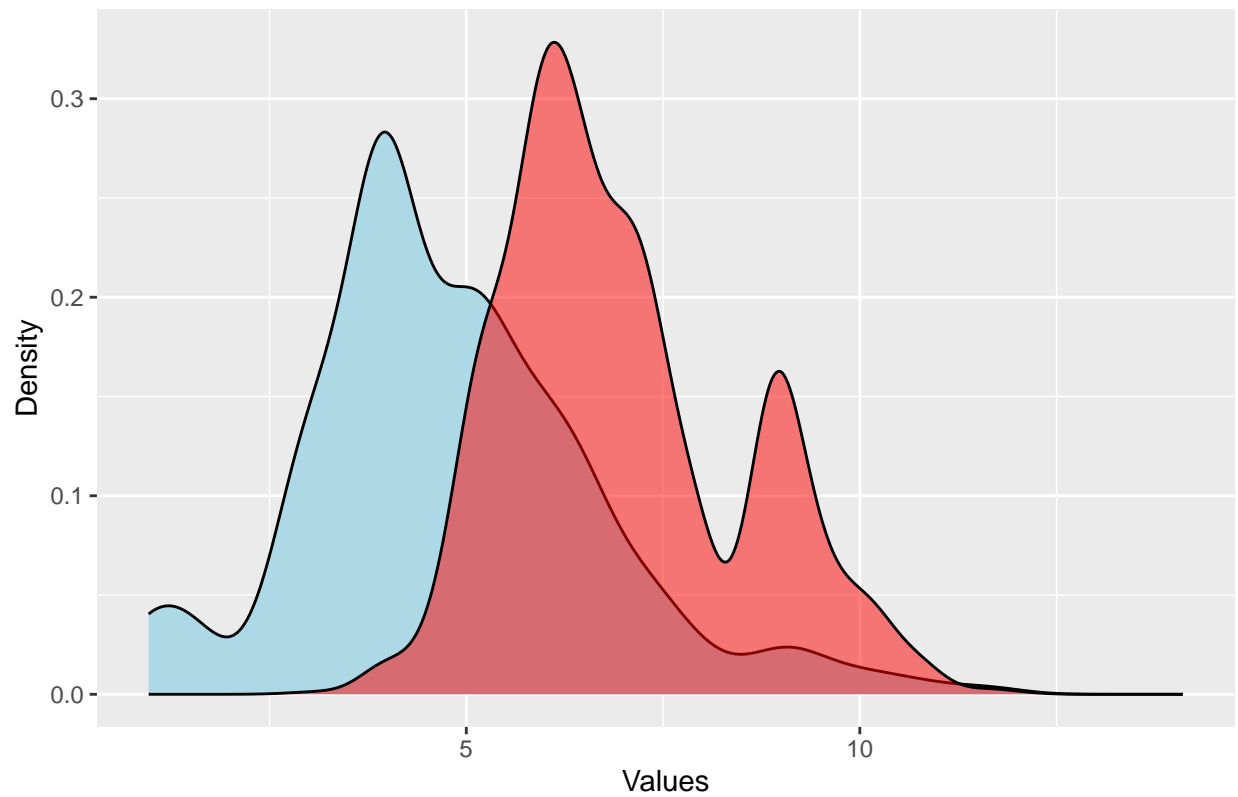
#now creating the new density

#shifting and normalising the new density
newspratsdens <- gmm[["density"]]*exp(-dig*gmm[["data"]])
shifted_pdf_normalized <- newspratsdens / sum(newspratsdens)

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

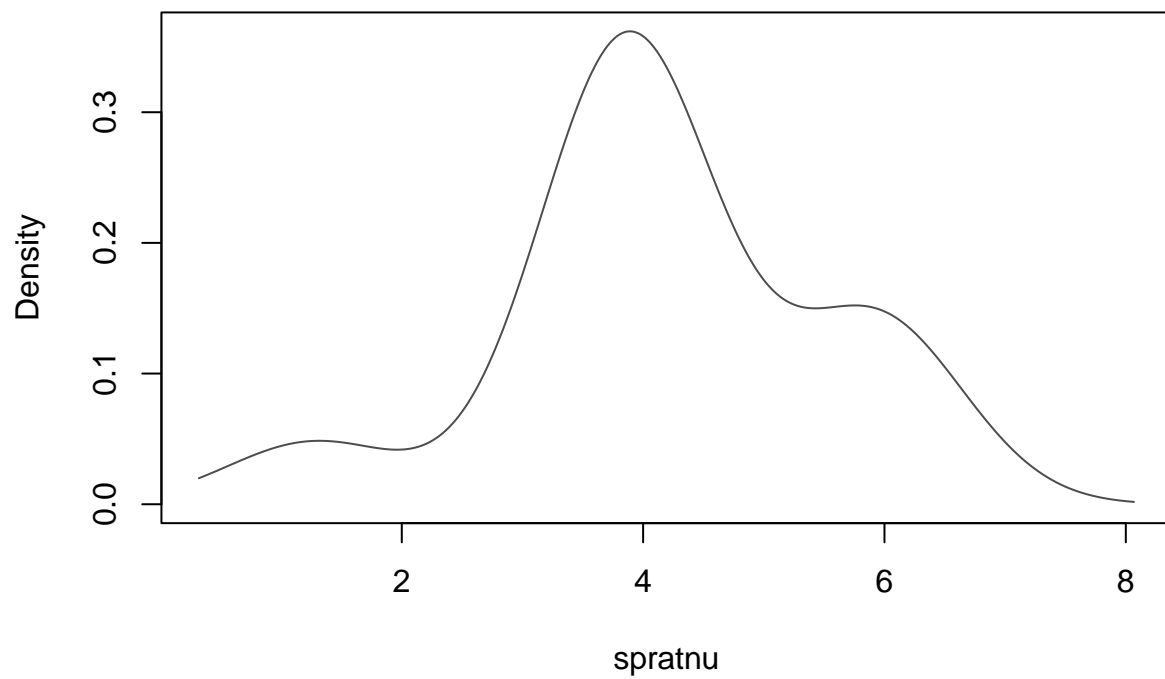
(numbfittbio <- ggplot() +
  geom_density(data=sprat, aes(log(ppmr), weight=weight_biomass), fill="lightblue")+
  geom_density(data=dplot, aes(x, weight=shifted_pdf_normalized), fill="red", alpha=0.5)+
  #geom_line(data=dplot, aes(x = x, y = density), color="red") +
  labs(x = "Values", y = "Density")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


Diet Density Plot from Number Distribution



It doesn't shift over very well. Will fit to the biomass distribution, then see if it works for the number distribution.

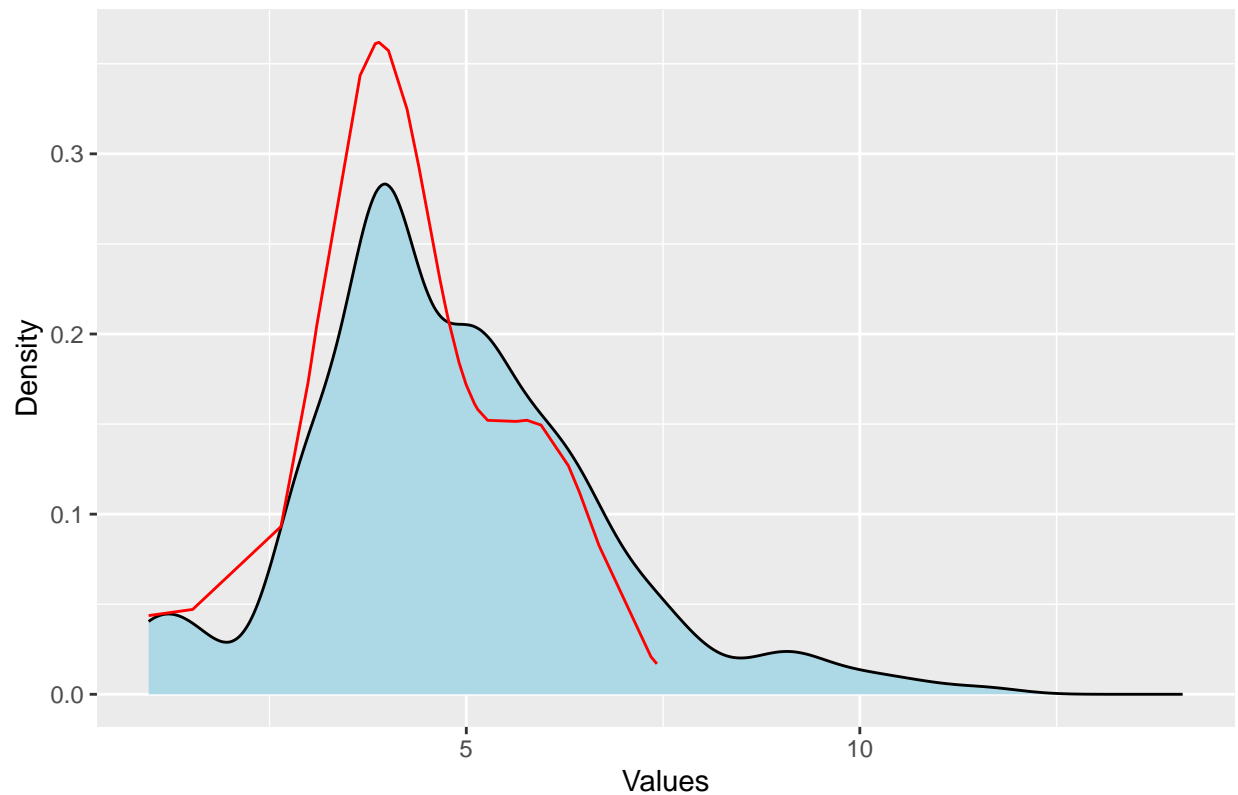
```
spratnu <- repeat_elements(log(sprat$ppmr), sprat$npred_perpred*sprat$prey_ind_weight_g^dig)
gmm <- densityMclust(spratnu, G=3)
```



```
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"))
```

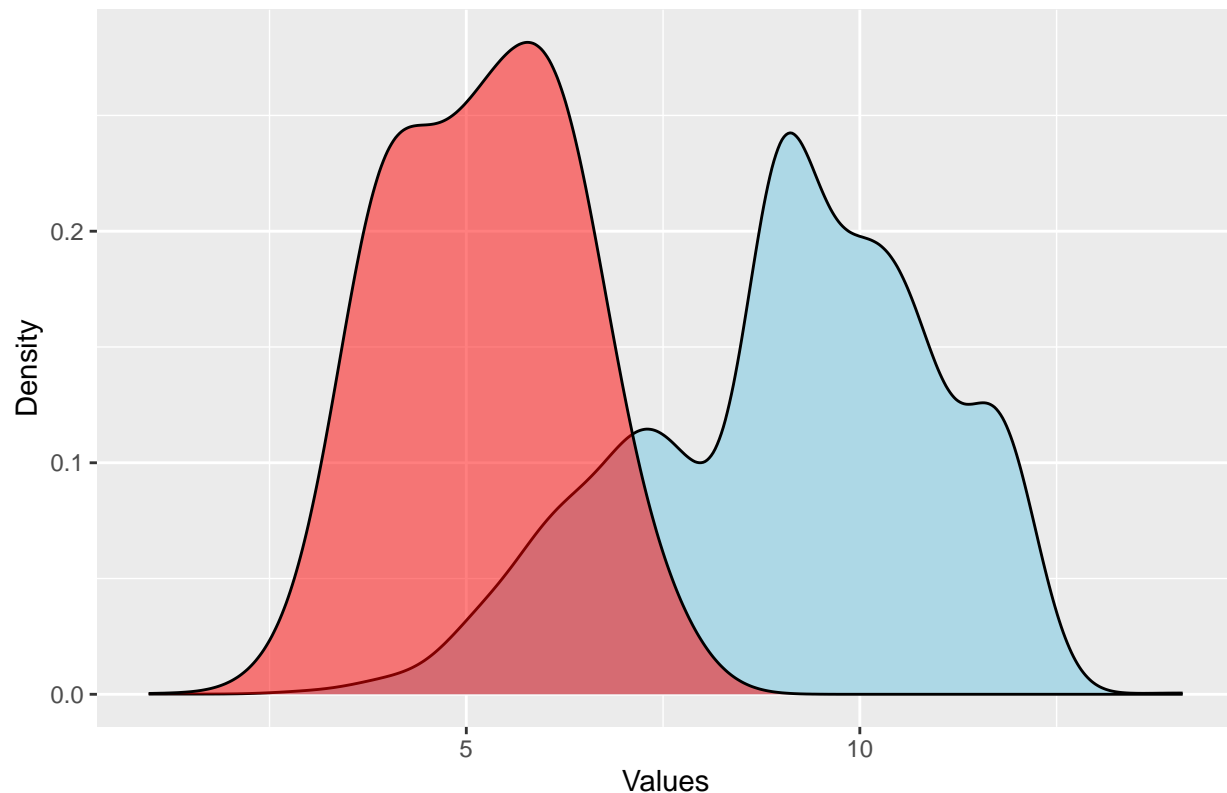
Diet Density Plot from Diet Distribution



```
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"))
```

Diet Density Plot from Number Distribution



#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"]] +
  (2/3)*shifted_fit[["parameters"]][["variance"]][["sigmasq"]]

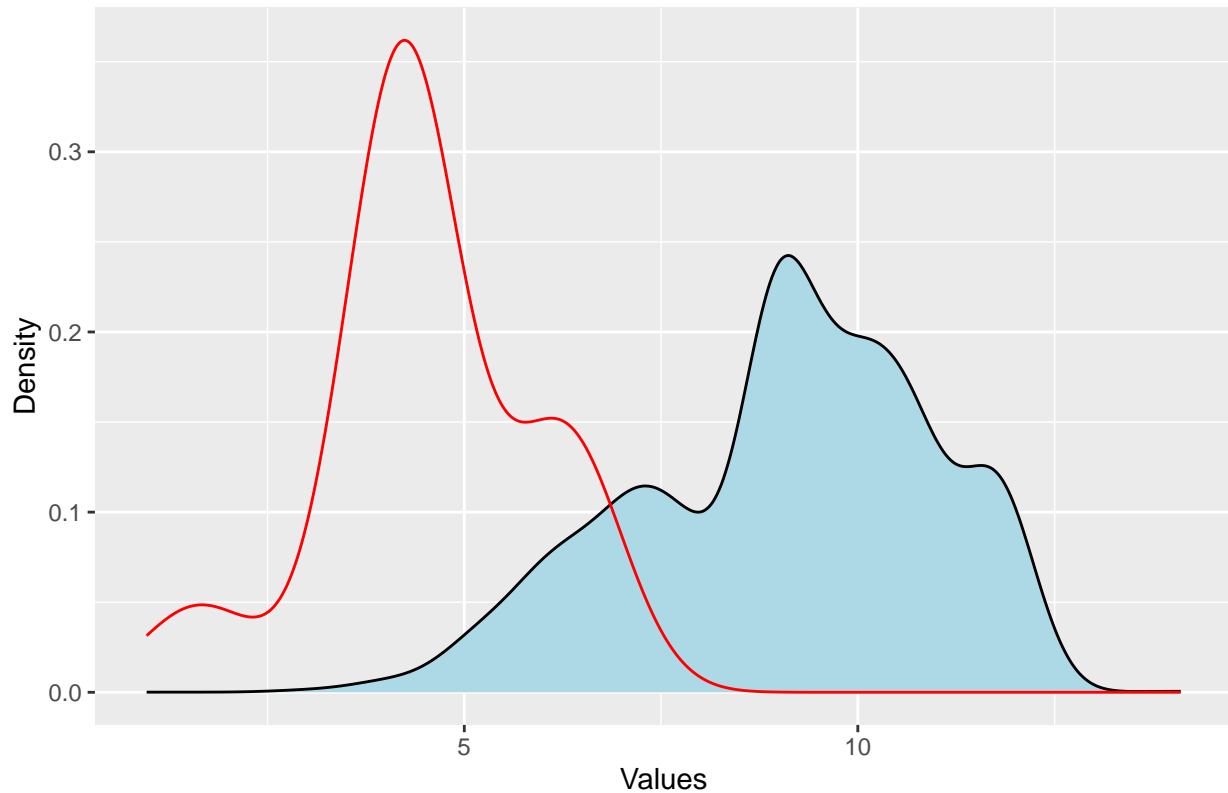
#generating the density values
shifted_pdf <- sapply(x_vals, function(x) {
  sum(shifted_fit$parameters$pro * dnorm(x, mean = shifted_fit$parameters$mean, sd = sqrt(shifted_fit$parameters$variance))
})

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.
```

Number Density Plot from Diet Distribution



This also doesnt shify very well. Should I 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
dteexp <- 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 <- 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)
  }

  est <- mle_texp(stomach)

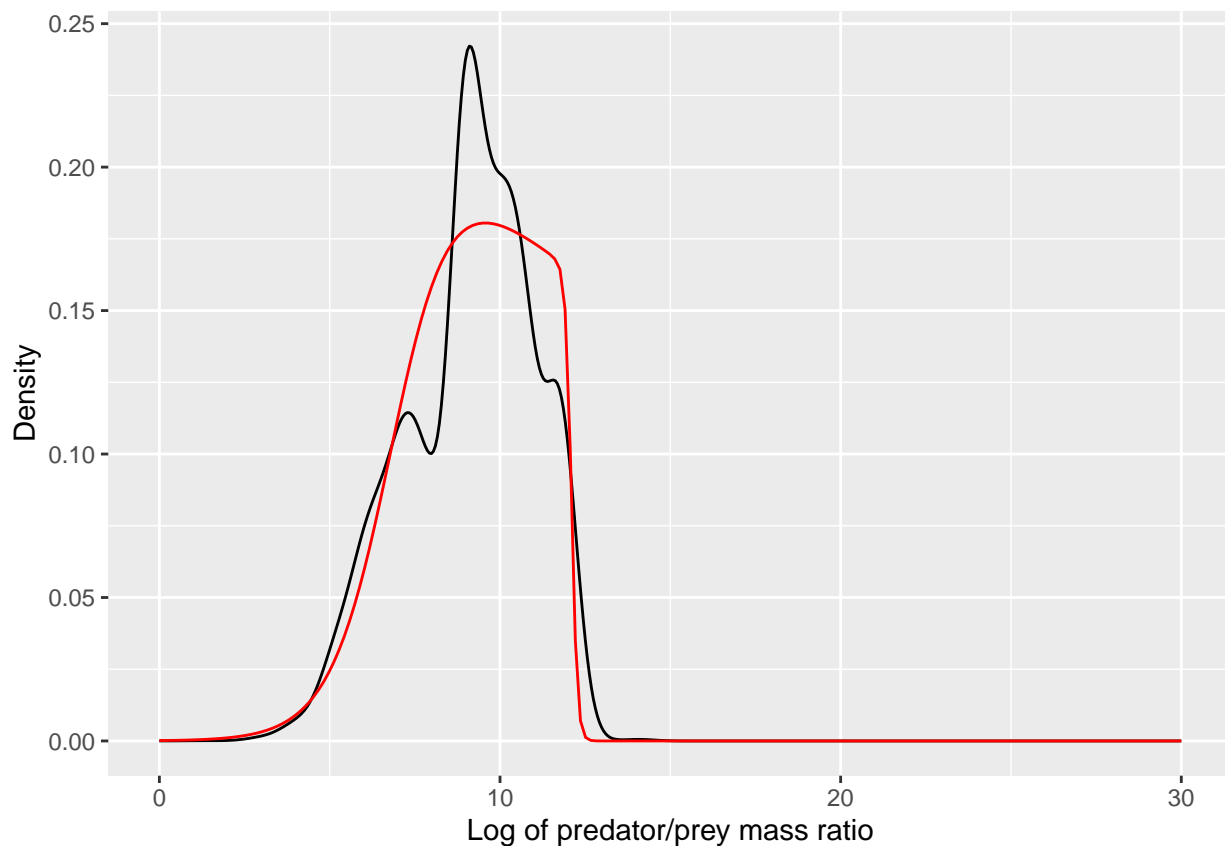
  biomassco <- est@coef

  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)

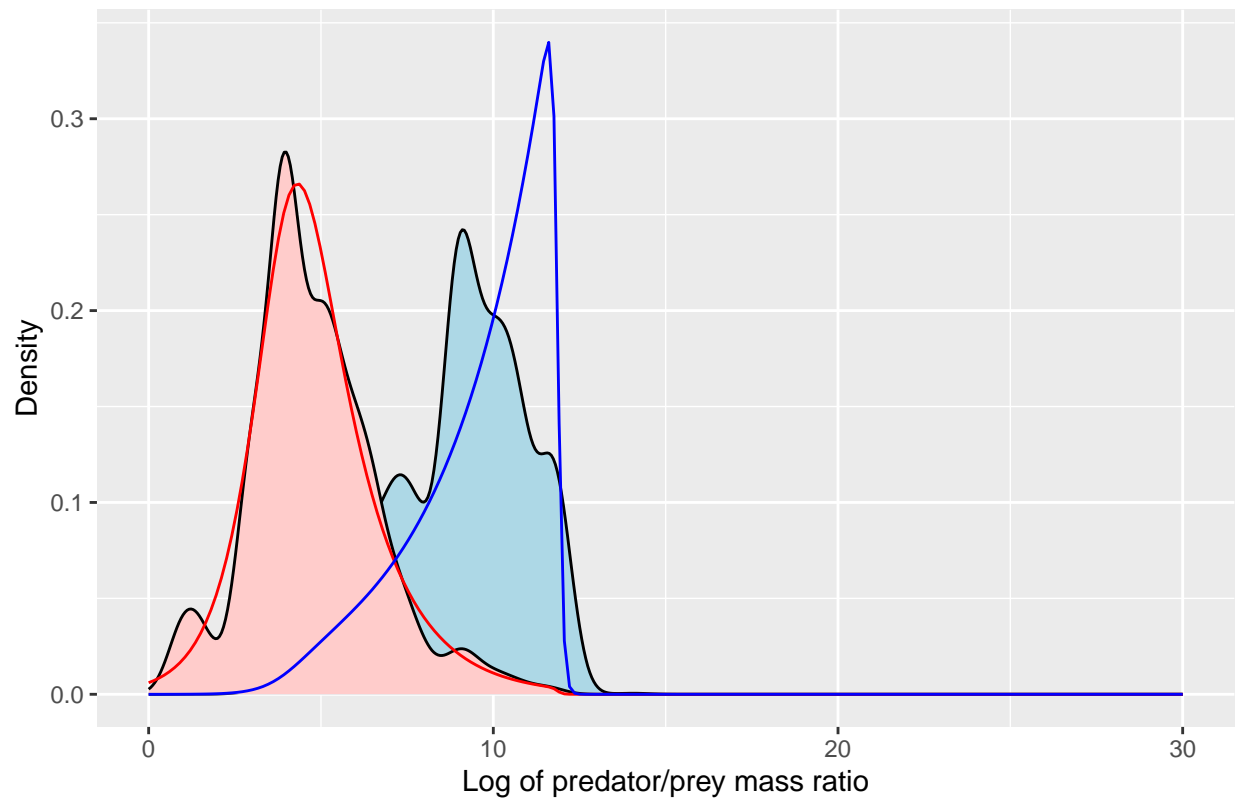
dist <- dtexp(grid, alpha = (numberestco[1]), ll = numberestco[2], ul = numberestco[3], lr = numberestco[4])
numberdist <- data.frame(l=grid, Density=dist)
shiftdist <- dtexp(grid, alpha = (numberestco[1]-1), ll = numberestco[2], ul = numberestco[3], lr = numberestco[4])
shiftnumberdist <- data.frame(l=grid, 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), 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 = shiftbiomassdist, color = "blue")+
  ggtitle("Fitted to Biomass, shift to number")

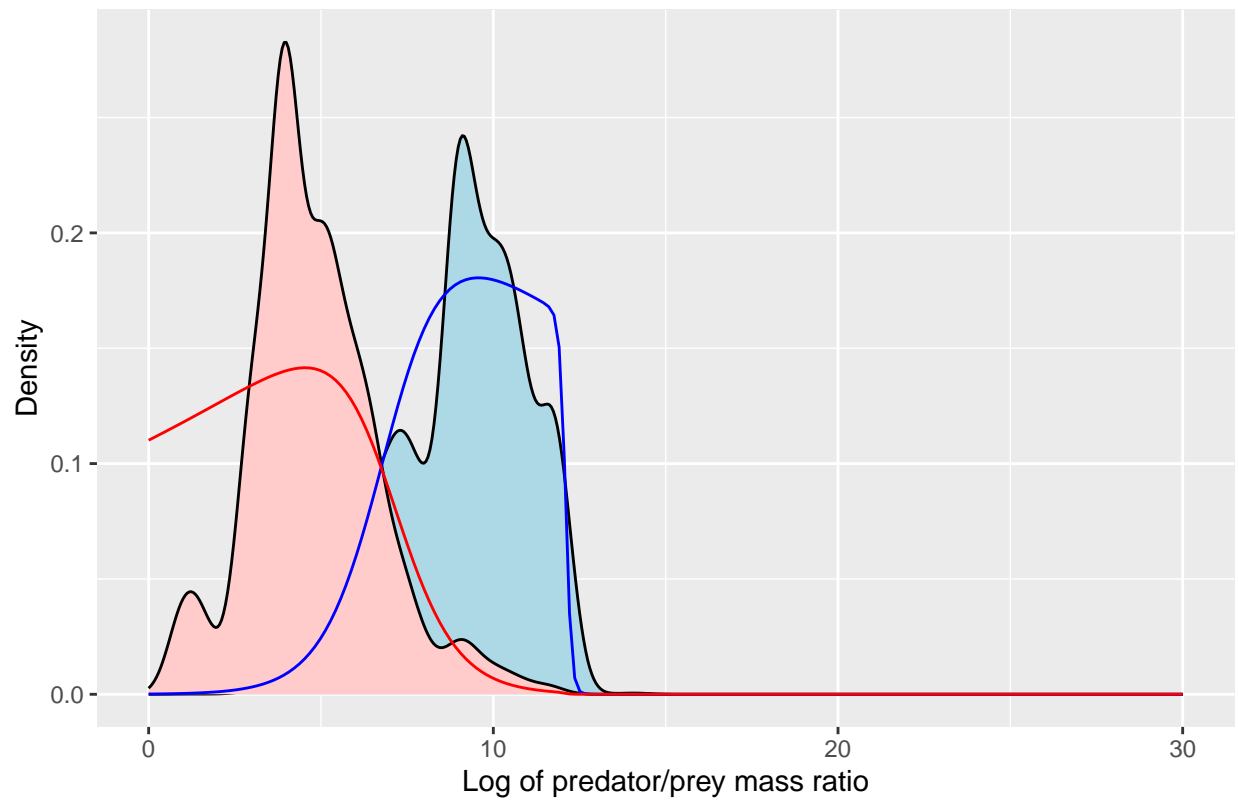
```


Fitted to Biomass, shift to number



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

Fitted to NUMBER, shift to Biomass



```
print(biomassestco)
```

```
##      alpha      ll      ul      lr      ur
## -0.6399368  4.0059442  1.7483370 11.8681193 13.6461300
```

When fitted to biomass, and shifted to number, it is good enough.