

Whiting

2024-07-13

```
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")
```

```
whiting <- stom_df %>% filter(pred_taxa=="Merlangius merlangus")
```

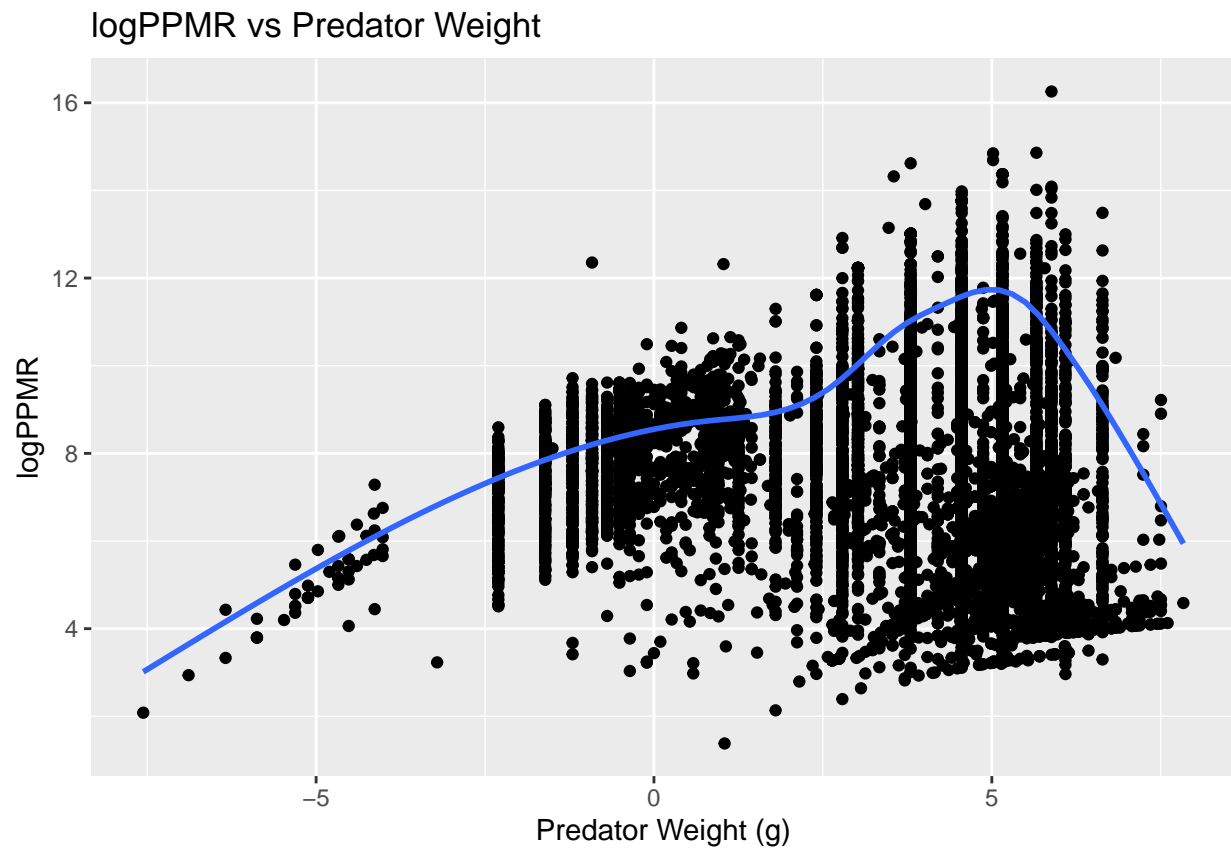
```
whiting <- whiting %>% filter(pred_weight_g < 10000, nprey_perpred > 1)
```

```
ggplot(whiting, 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="Predator Weight (g)", y="logPPMR")
```

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

## Warning: Removed 10 rows containing non-finite outside the scale range
## ('stat_smooth()').

## Warning: Removed 10 rows containing missing values or values outside the scale range
## ('geom_point()').
```



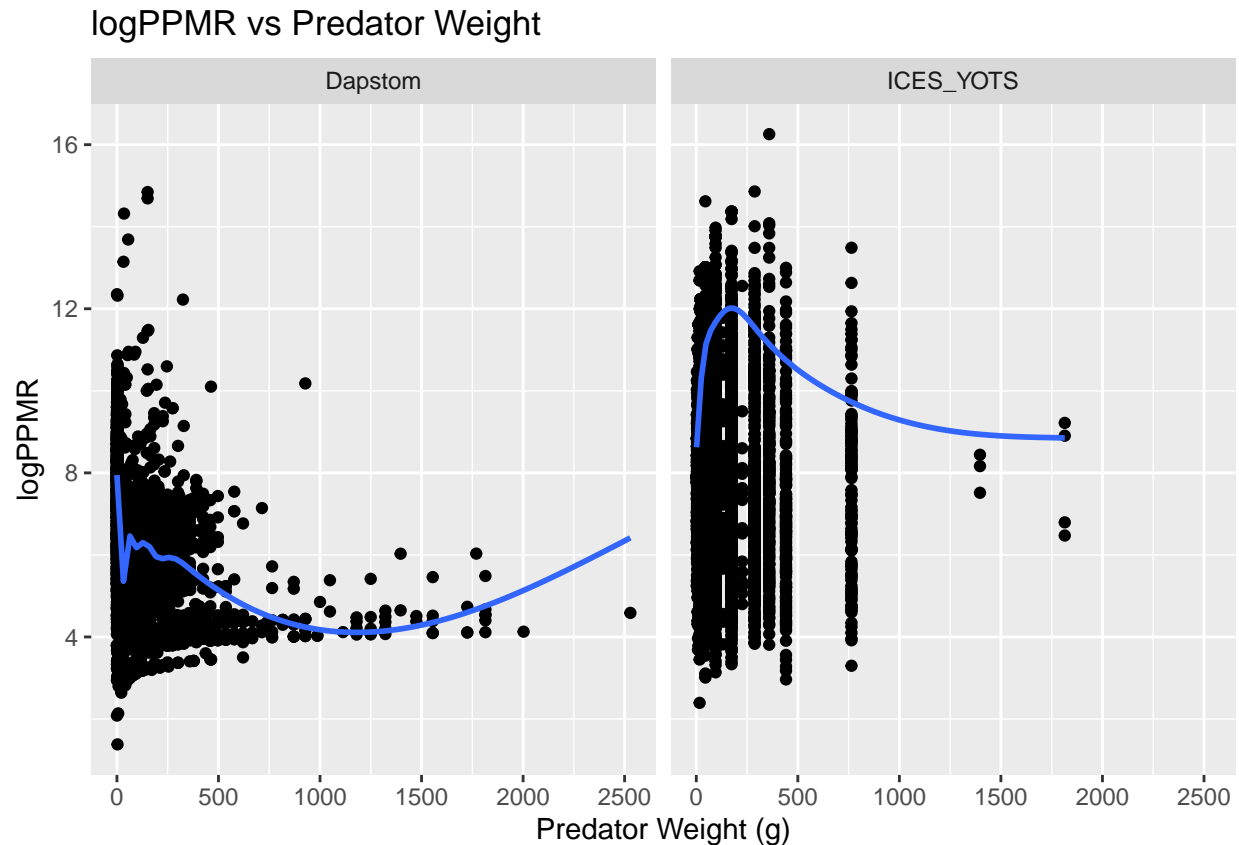
There appears to be two separate data formats in this, I will plot separately.

```
ggplot(whiting, aes(x=pred_weight_g, y=log(ppmr)))+
  geom_point()+
  facet_wrap(~data)+
  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 10 rows containing non-finite outside the scale range
## ('stat_smooth()').

## Warning: Removed 10 rows containing missing values or values outside the scale range
## ('geom_point()').
```



There is still variation in the PPMR, but this might dissapear once I control for biomass contribution (diet)

```
dig <- 1

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

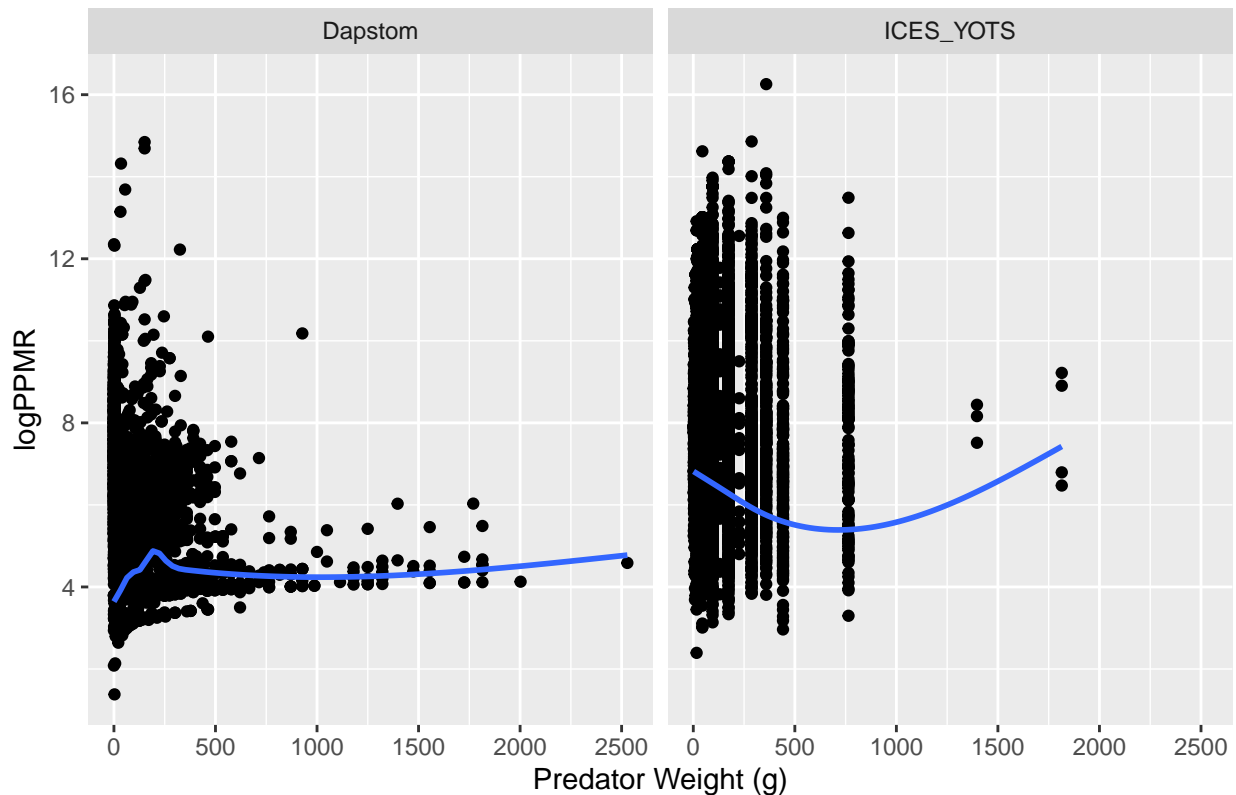
## Warning: Use of 'whiting$prey_ind_weight_g' is discouraged.
## i Use 'prey_ind_weight_g' instead.

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

## Warning: Removed 10 rows containing non-finite outside the scale range
## ('stat_smooth()').

## Warning: Removed 10 rows containing missing values or values outside the scale range
## ('geom_point()').
```

logPPMR vs Predator Weight



This looks good, the PPMR is approximately the same across the predator size range. Other than for the juveniles, this may be due to sampling effort for smaller sized prey only being used with smaller sized predators (So a differential sampling strategy for juveniles or different sizes of whiting) However, it also may be due to the ontogeny of whiting - when they are 1 years of age they migrate to the open water, so the feeding ecology will change.

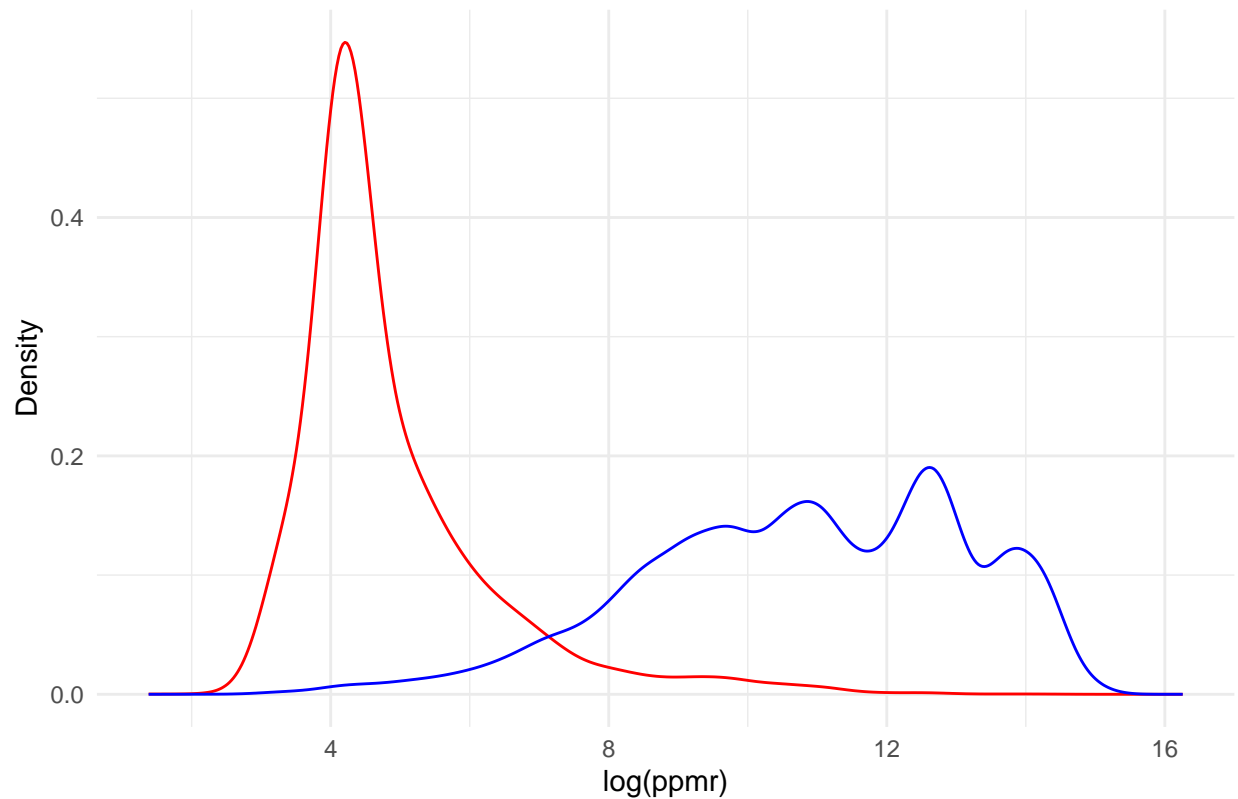
Small bit on feeding ecology of whiting - Analysing the stomach contents of North Sea whiting, found that as whiting increased in size they switched from a crustacean, to a fish dominated diet, and the average size of prey eaten increases - The whiting appeared to be exploiting one prey species in an area. The majority of fish stomachs sampled at a station contained the same prey type.

```
whiting$weight_numbers <- whiting$npred_perpred
whiting$weight_biomass <- whiting$npred_perpred*whiting$prey_ind_weight_g^dig

ggplot() +
  geom_density(data = whiting, aes(x = log(ppmr), weight = weight_biomass), color = "red") +
  geom_density(data = whiting, 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 10 rows containing non-finite outside the scale range
## ('stat_density()').
## Removed 10 rows containing non-finite outside the scale range
## ('stat_density()').
```

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



So, from a preliminary look at the data, I think that the truncated exponential would fit the best. Or a mixture gaussian model.

```
whiting <- whiting%>%mutate(logppmr = log(ppmr))
names(whiting)[names(whiting) == "logppmr"] <- "l"

stomach <- whiting

stomach <- stomach[!is.na(stomach$l),]

names(stomach)[names(stomach) == "nprey_perpred"] <- "Nprey"

stomach <- stomach %>% mutate(weight_numbers = Nprey / sum(Nprey))
#stomach <- stomach%>%mutate(weight_numbers=Nprey*wprey^(1))

#defining the exponential and the MLE
fl <- function(l, alpha, ll, ul, lr, ur) {
  dl <- ll - l
  dr <- l - lr
  fl <- exp(alpha * l) /
    (1 + exp(ul * dl)) /
    (1 + exp(ur * dr))
  # fl[fl <= 0] <- 0
}

dtexp <- function(l, alpha, ll, ul, lr, ur) {
  d <- fl(l, alpha, ll, ul, lr, ur) /
```

```

    integrate(f1, 0, 30, alpha = alpha,
              ll = ll, ul = ul, lr = lr, ur = ur)$value
  return(d)
}
mle_texp <- function(df) {
  loglik <- function(alpha, ll, ul, lr, ur) {
    L <- dtexp(stomach$l, alpha, ll, ul, lr, ur)
    - sum(log(L) * stomach$weight_numbers)
  }
  mle2(loglik, start = list(
    alpha = 0.5,
    ll = min(stomach$l),
    lr = max(stomach$l),
    ul = 5,
    ur = 5))
}

est <- mle_texp(stomach)

```

```

## Warning in mle2(loglik, start = list(alpha = 0.5, ll = min(stomach$l), lr =
## max(stomach$l), : convergence failure: code=1 (iteration limit 'maxit' reached)

```

```

estco <- est@fullcoef
grid = seq(0, 30, length.out = 200)
dist <- dtexp(grid, alpha = estco[1], ll = estco[2], ul = estco[3], lr = estco[4], ur = estco[5])
dist <- data.frame(l=grid, Density=dist)

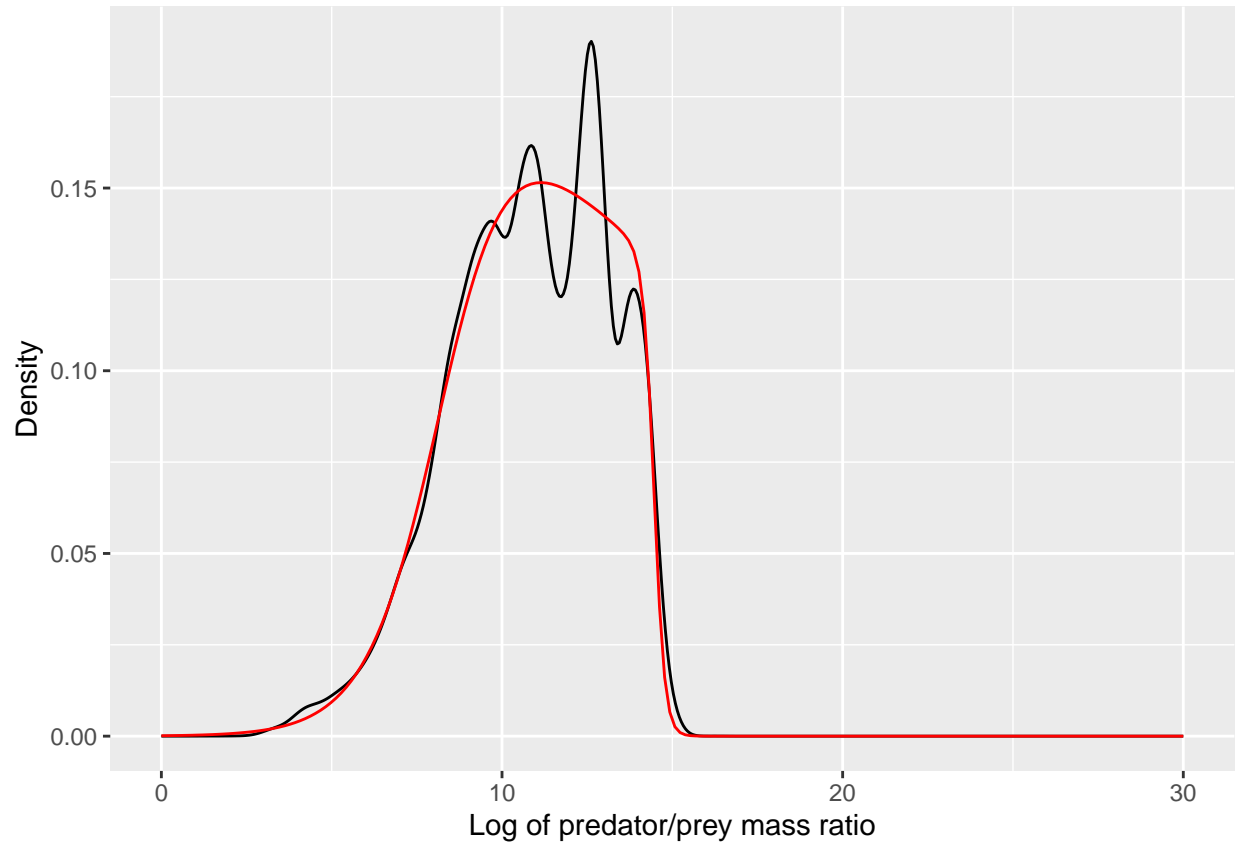
ggplot(whiting) +
  geom_density(aes(l, weight=weight_numbers))+
  xlab("Log of predator/prey mass ratio") +
  geom_line(aes(l, Density), data = dist, color = "red")

```

```

## Warning: Removed 10 rows containing non-finite outside the scale range
## ('stat_density()').

```



It wont fit to the data correctly. Maybe I have to clean it

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

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
dtepx <- function(l, alpha, ll, ul, lr, ur) {
  fl_values <- fl(l, alpha, ll, ul, lr, ur)

  integral_result <- tryCatch(
    integrate(fl, 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 <- fl_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)
}

```



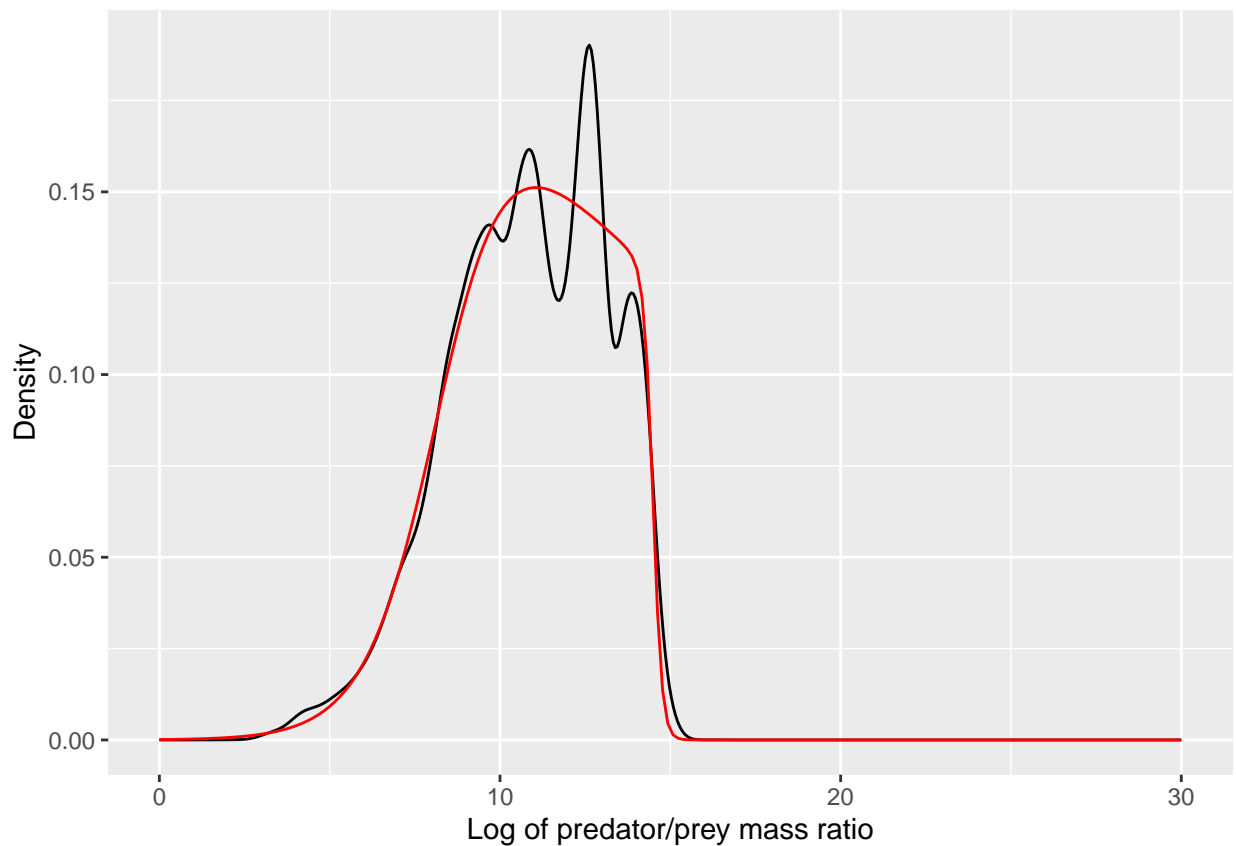
```

est <- mle_texp(stomach)

estco <- est@fullcoef
grid = seq(0, 30, length.out = 200)
dist <- dtexp(grid, alpha = estco[1], ll = estco[2], ul = estco[3], lr = estco[4], ur = estco[5])
dist <- data.frame(l=grid, Density=dist)

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

```



This is a very good fit!. alpha ll ul lr ur -0.06855893 8.38566550 0.96082440 14.49587913 7.62656791

Hopefully it moves across.

```

dist <- dtexp(grid, alpha = estco[1]-1, ll = estco[2], ul = estco[3], lr = estco[4], ur = estco[5])
dist <- data.frame(l=grid, Density=dist)

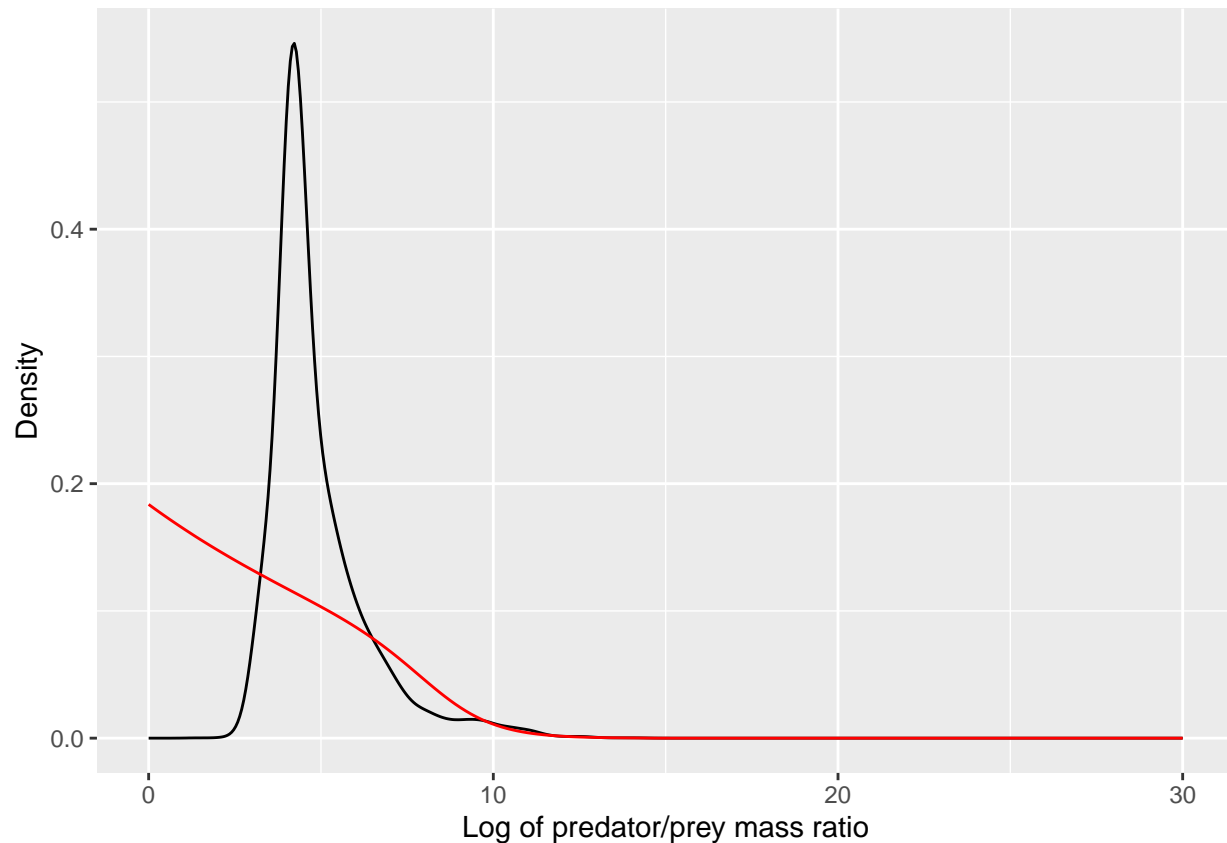
ggplot(stomach) +
  geom_density(aes(l, weight=Nprey*stomach$prey_ind_weight_g^dig))+
  xlab("Log of predator/prey mass ratio") +
  geom_line(aes(l, Density), data = dist, color = "red")

```

```

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

```



Ok this has ruined it. It is not a good fit for the biomass density.
I will try the biomass density first.

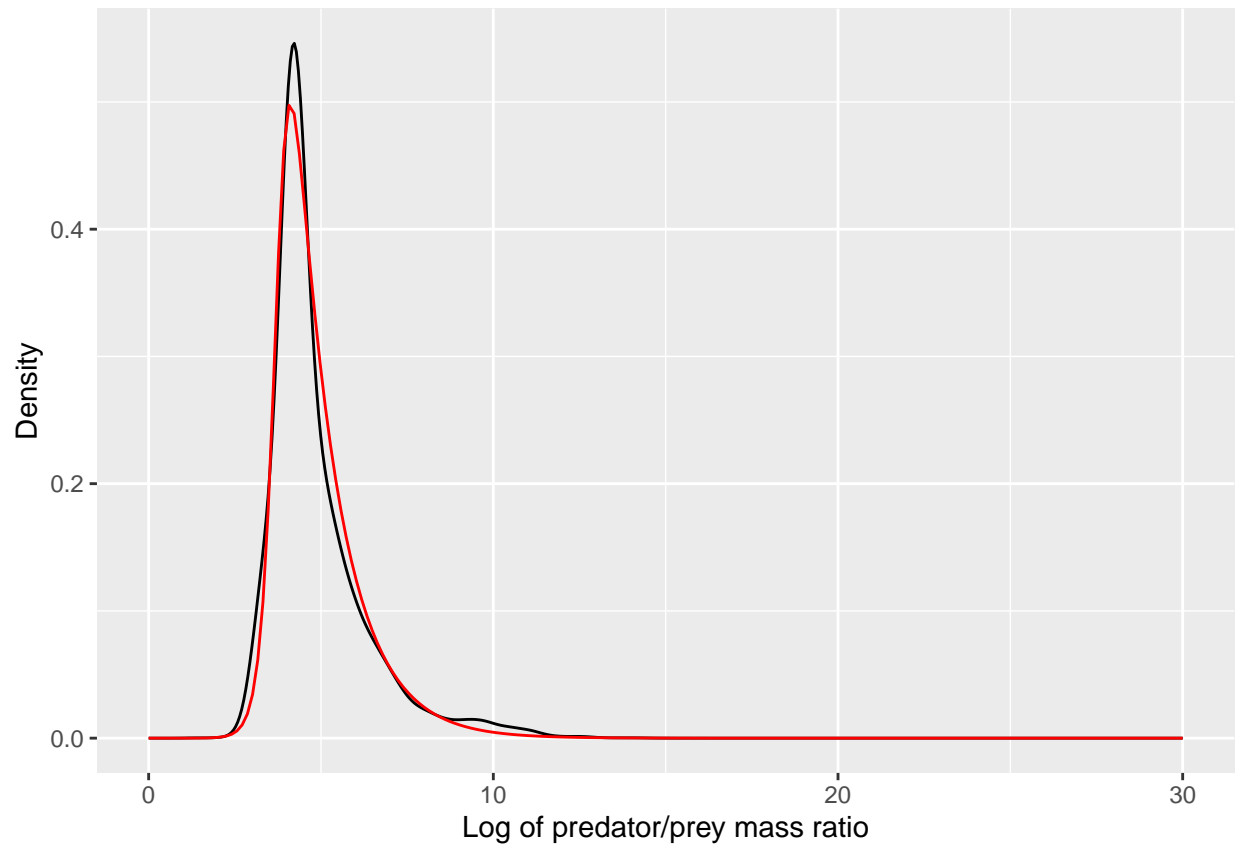
```
stomach <- stomach%>%mutate(weight_numbers=Nprey*stomach$prey_ind_weight_g^dig)

est <- mle_texp(stomach)

estco <- est@fullcoef
grid = seq(0, 30, length.out = 200)
dist <- dtexp(grid, alpha = estco[1], ll = estco[2], ul = estco[3], lr = estco[4], ur = estco[5])
dist <- data.frame(l=grid, Density=dist)

ggplot(stomach) +
  geom_density(aes(l, weight=Nprey*stomach$prey_ind_weight_g^dig))+
  xlab("Log of predator/prey mass ratio") +
  geom_line(aes(l, Density), data = dist, color = "red")
```

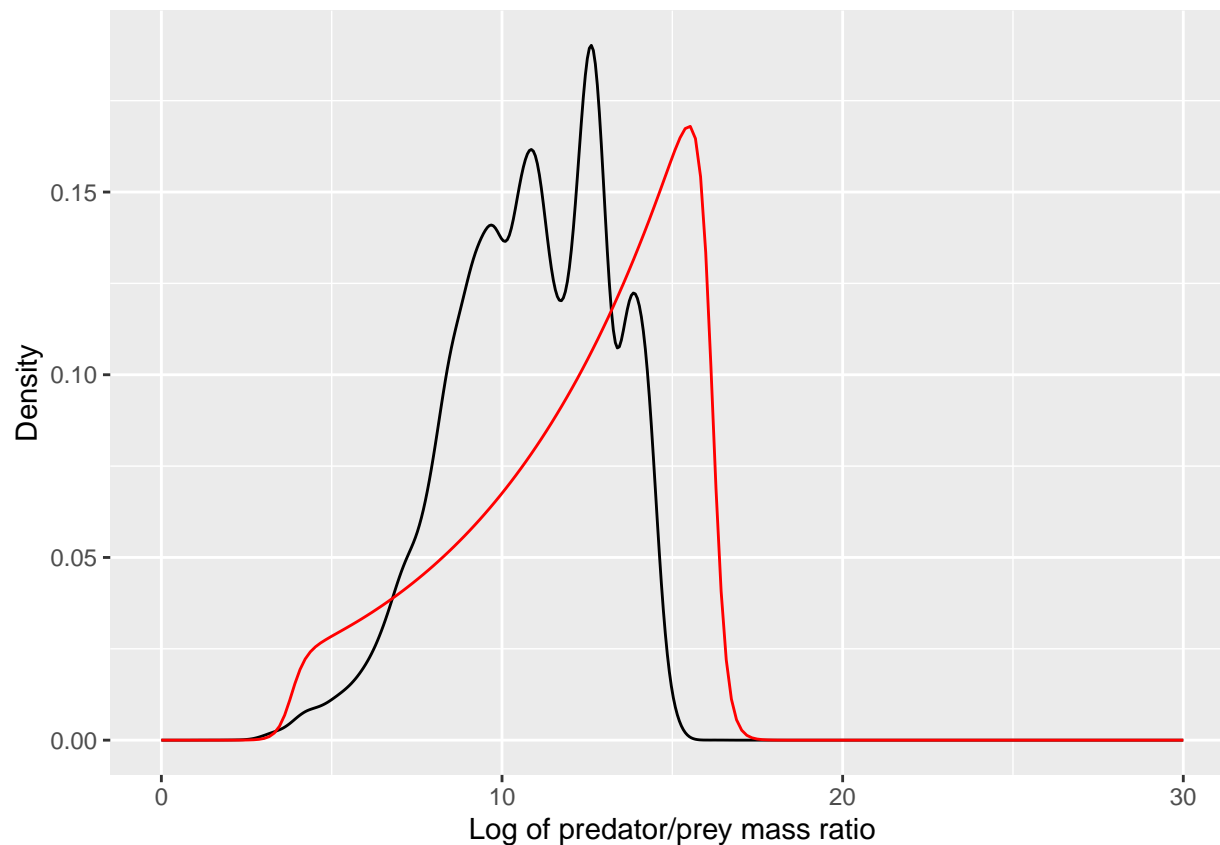
```
## Warning: Use of 'stomach$prey_ind_weight_g' is discouraged.
## i Use 'prey_ind_weight_g' instead.
```



Ok the fit is good. Now will it shift??

```
dist <- dtexp(grid, alpha = estco[1]+1, ll = estco[2], ul = estco[3], lr = estco[4], ur = estco[5])
dist <- data.frame(l=grid, Density=dist)

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



This looks good. I think we have it. (note, if we factor in the diet, so 1, then it looks very good - look below.)

```
alpha      ll      ul      lr      ur
-0.3145214 3.6304283 4.5452752 13.3696362 1.6451461

dist <- dtexp(grid, alpha = estco[1]+1, ll = estco[2], ul = estco[3], lr = estco[4], ur = estco[5])
dist <- data.frame(l=grid, Density=dist)

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

