Reproducing Graphs

Georgina Shaw

30/11/2021

Cod

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 == "Gadus morhua",
         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 Gadus morhua    0.00001     8097. 0.0200  21.2
```

```
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 Gadus morhua 8097.      9401.   100 0.149
```

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: 23 x 5
## # Groups:   Species [1]
##    Species         cut Numbers Biomass      l
##    <chr>         <int>   <dbl>   <dbl>  <dbl>
##  1 Gadus morhua      1 0.00295 0.143   0.0200
##  2 Gadus morhua      2 0.0138  0.248   0.751
##  3 Gadus morhua      3 0.0489  0.451   1.48
##  4 Gadus morhua      4 0.0807  0.277   2.21
##  5 Gadus morhua      5 0.115   0.0907  2.94
##  6 Gadus morhua      6 0.196   0.0674  3.68
##  7 Gadus morhua      7 0.222   0.0478  4.41
##  8 Gadus morhua      8 0.205   0.0248  5.14
##  9 Gadus morhua      9 0.168   0.0111  5.87
## 10 Gadus morhua     10 0.130   0.00424 6.60
## # ... with 13 more rows
```

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

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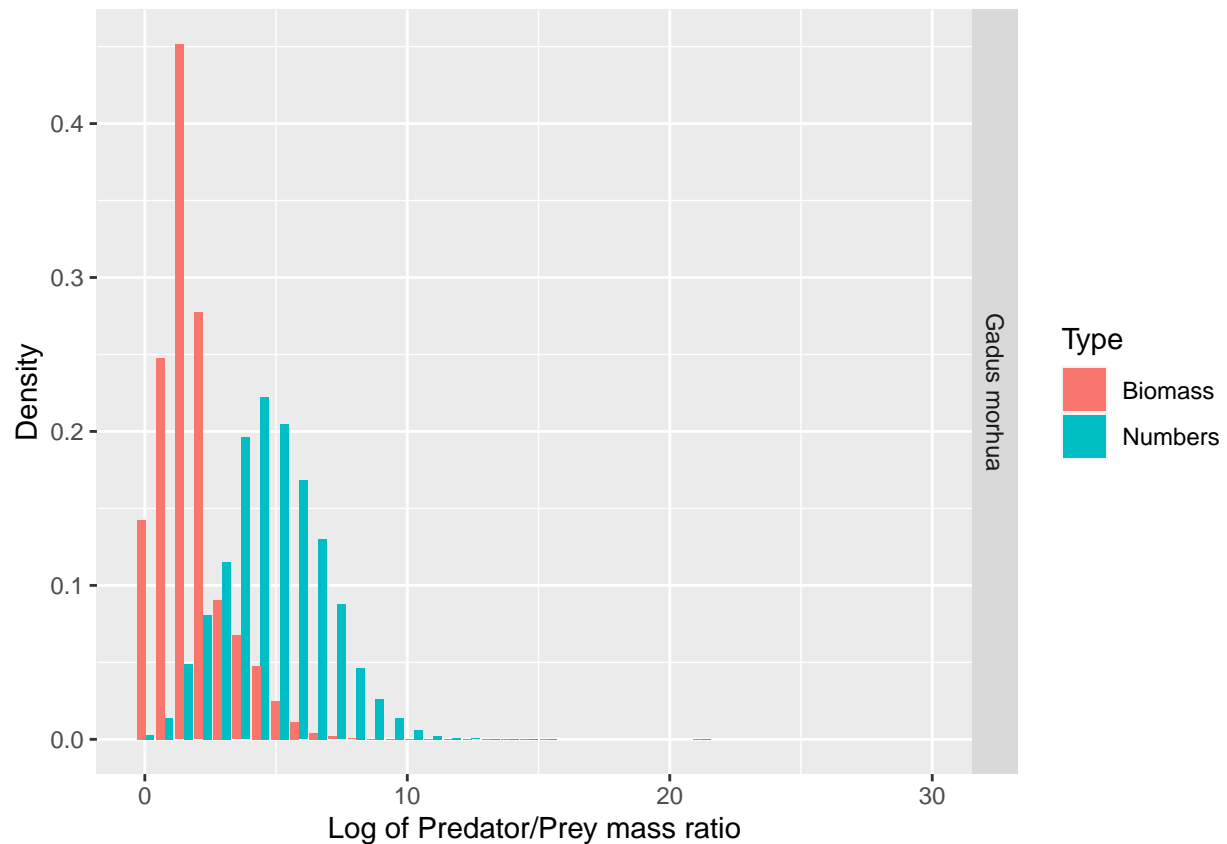
Histograms

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))
```


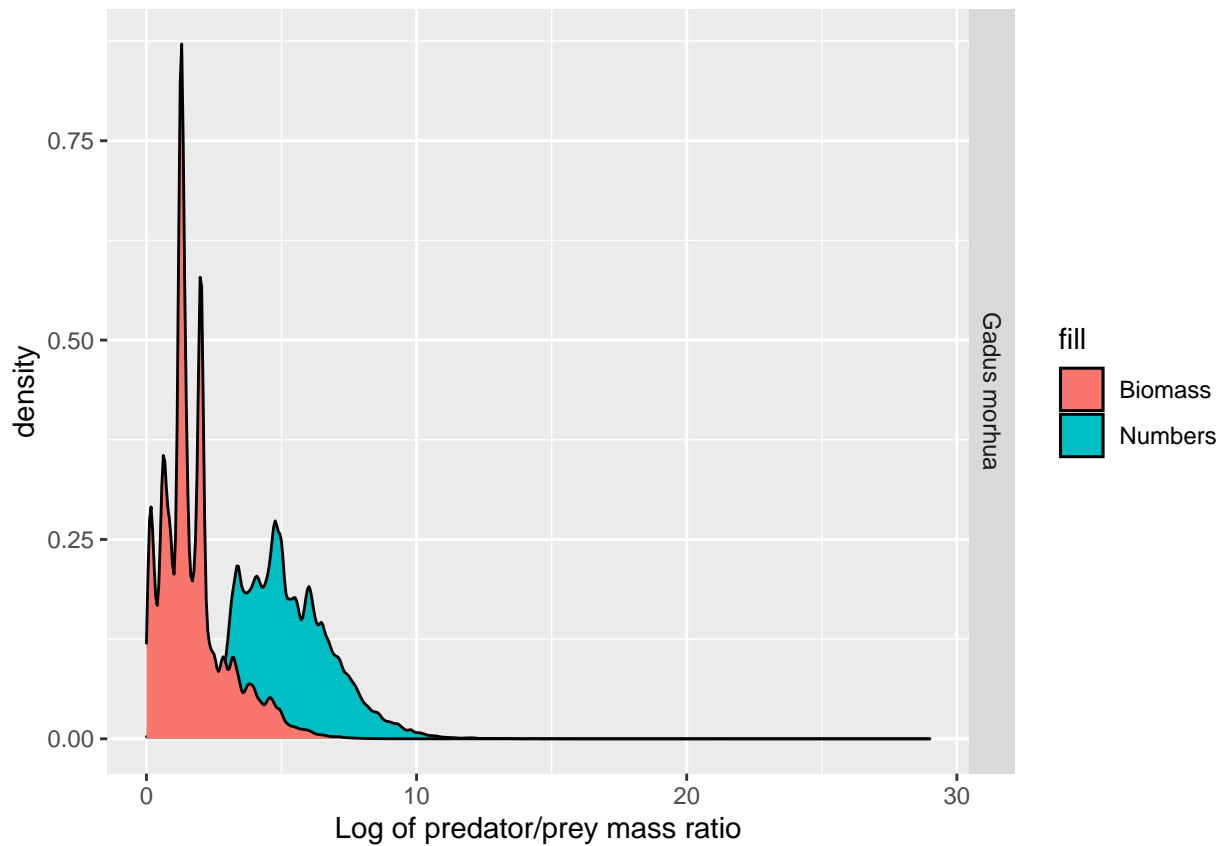## Kernal Density Estimation

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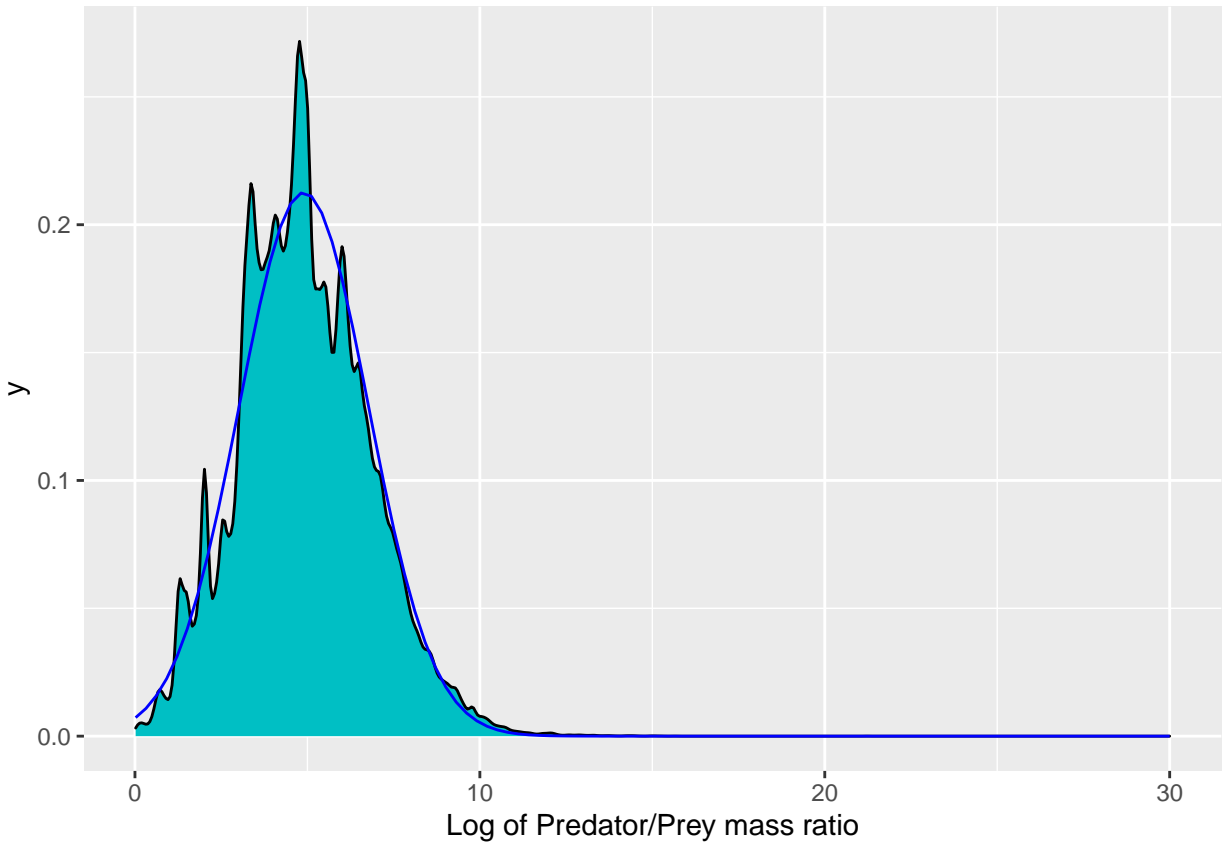
```
xlab("Log of predator/prey mass ratio") +
expand_limits(x = c(0, 29))
```


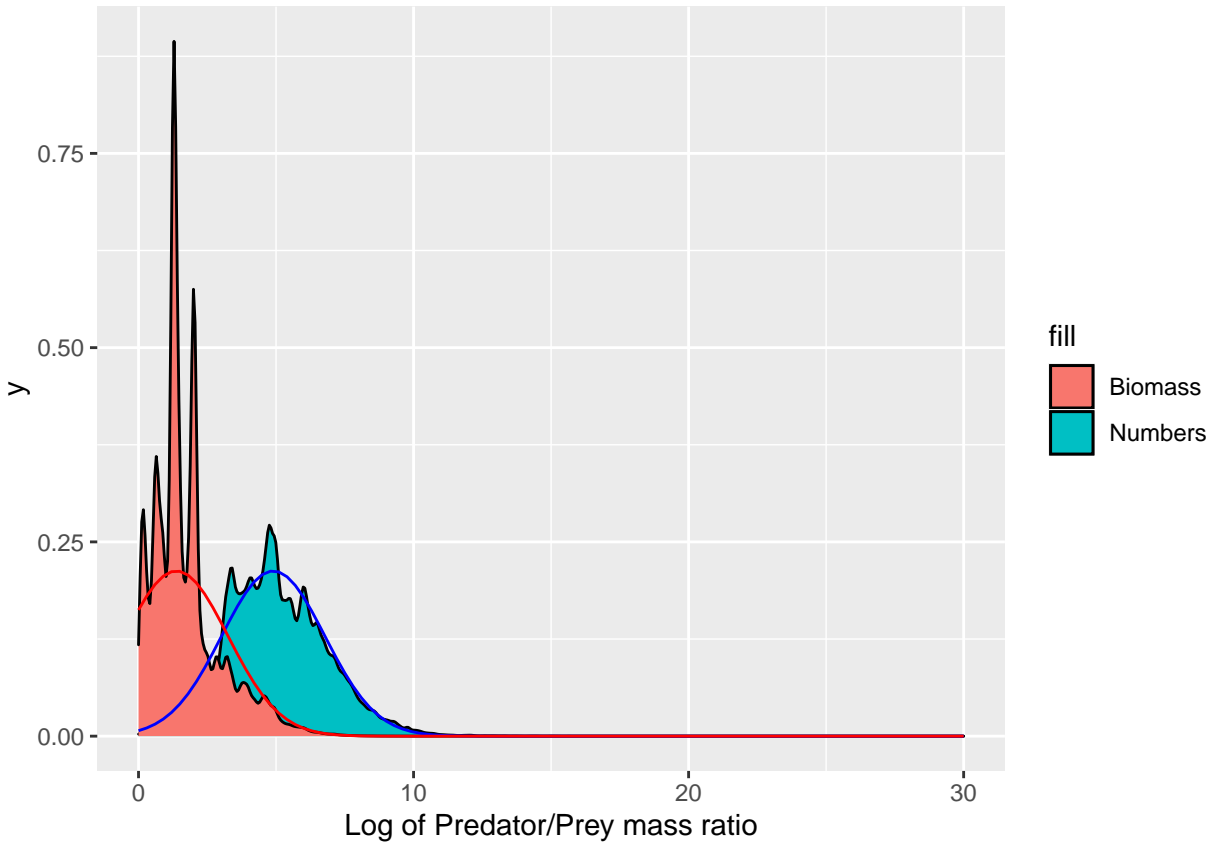## Gaussian Distribution fit

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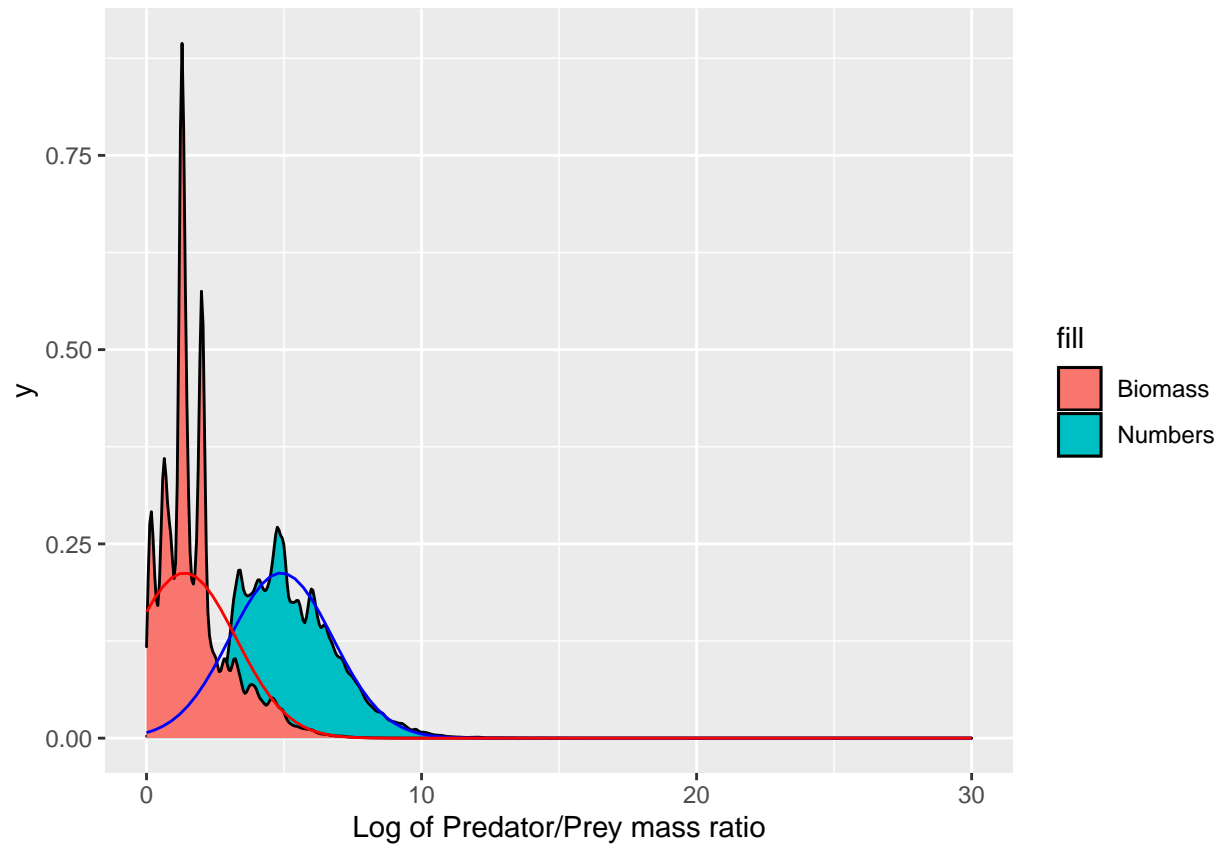
```r
weighted.sd <- function(x, w) {
  sqrt(sum(w * (x - weighted.mean(x, w))^2))
}
fit <- stomach %>%
  summarise(mean = weighted.mean(l, weight_numbers),
            sd = weighted.sd(l, weight_numbers))
stomach %>%
  ggplot() +
  geom_density(aes(l, weight = weight_numbers,
                   fill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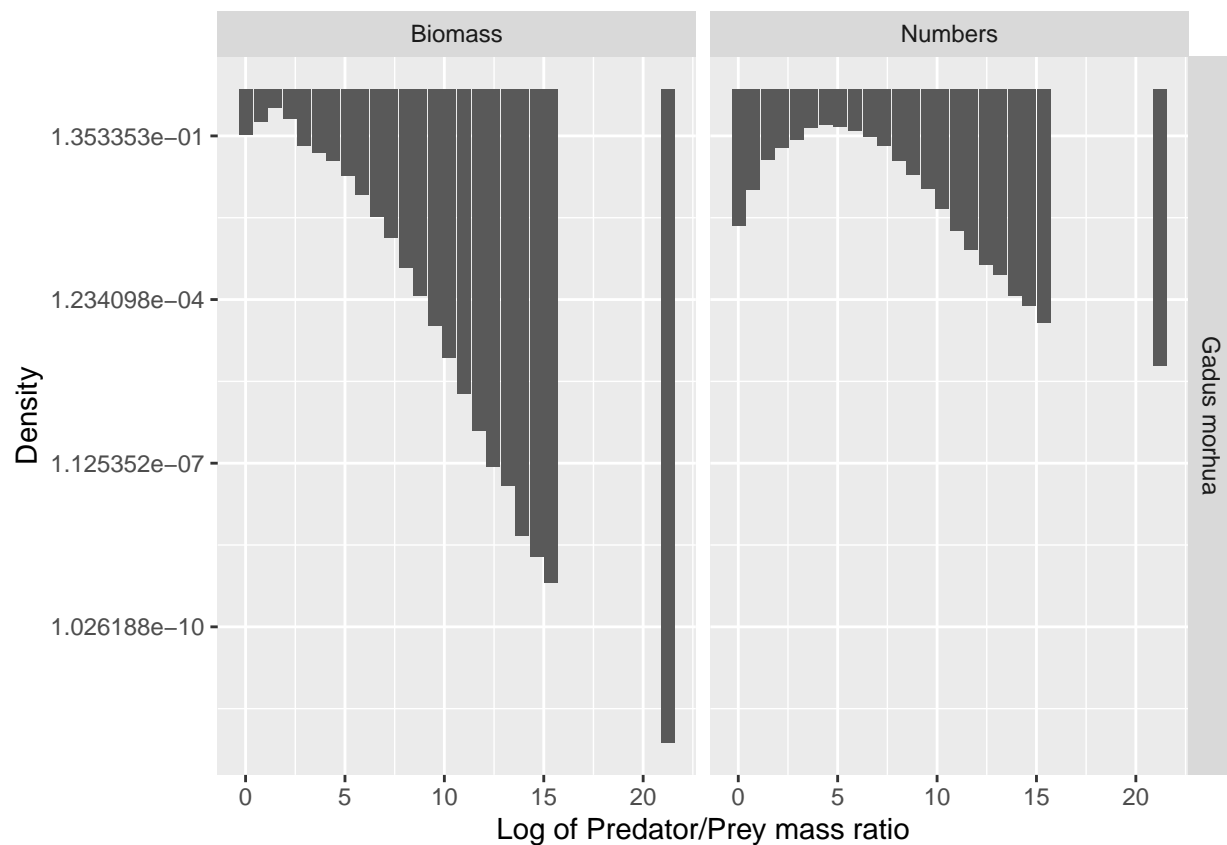
Fitting the Exponential DIstribution

Replot the histograms but with the logarithmic y-axis

```
binned_stomach %>%
  ggplot(aes(l, Density)) +
  geom_col() +
  facet_grid(Species ~ Type, scales = "free_y") +
  xlab("Log of Predator/Prey mass ratio") +
  scale_y_continuous(trans = "log")
```

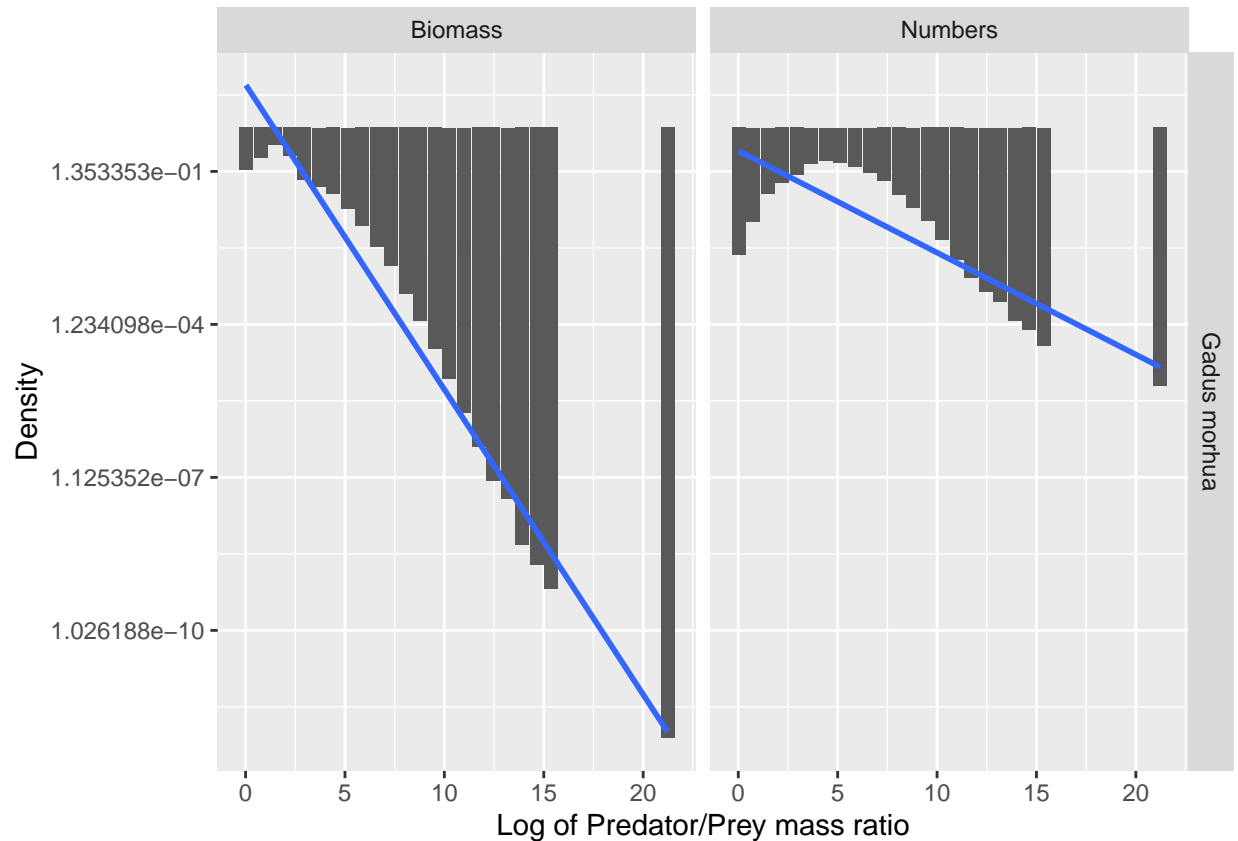Fitting the truncated exponential distribution

```
binned_stomach %>%
  ggplot(aes(l, Density)) +
  geom_col() +
  geom_smooth(method = "lm", se = FALSE) +
  facet_grid(Species ~ Type, scales = "free_y") +
  xlab("Log of Predator/Prey mass ratio") +
  scale_y_continuous(trans = "log")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
binned_stomach %>%
  group_by(Species, Type) %>%
  group_modify(~ broom::tidy(lm(log(Density) ~ l, data = .x))) %>%
  filter(term == "l")
```

```
## # A tibble: 2 x 7
## # Groups:   Species, Type [2]
##   Species      Type     term  estimate std.error statistic  p.value
##   <chr>        <chr>    <chr>    <dbl>     <dbl>     <dbl>     <dbl>
## 1 Gadus morhua Biomass  l       -1.40     0.0610    -22.9  2.53e-16
## 2 Gadus morhua Numbers  l       -0.466    0.0709     -6.58 1.63e- 6
```

Maximum Likelihood fit

```
est <- stomach %>%
  group_by(Species) %>%
  summarise(lbar = weighted.mean(l, weight_numbers),
            lmax = max(l),
            lmin = min(l),
            alpha = 1/(lmax - lbar))
est
```

```
## # A tibble: 1 x 5
```

```
##   Species         lbar   lmax   lmin   alpha
##   <chr>          <dbl>  <dbl>  <dbl>   <dbl>
## 1 Gadus morhua    4.90   21.2 0.0200  0.0612
```

```r
dnumbers <- function(l, alpha, lmax) {
  d <- as.numeric(l <= lmax)
  d[d > 0] <- dexp(lmax - l[d > 0], alpha)
  return(d)
}
dbiomass <- function(l, alpha, lmin) {
  d <- as.numeric(l >= lmin)
  d[d > 0] <- dexp(l[d > 0] - lmin, 1 - alpha)
  return(d)
}
selected_species <- "Gadus morhua"
binned_stomach %>%
  filter(Species == selected_species) %>%
  ggplot() +
    geom_col(aes(l, Density, fill = Type)) +
    stat_function(fun = dnumbers,
                  args = list(alpha = est$alpha[est$Species == selected_species],
                              lmax = est$lmax[est$Species == selected_species]),
                  colour = "blue") +
    stat_function(fun = dbiomass,
                  args = list(alpha = est$alpha[est$Species == selected_species],
                              lmin = est$lmin[est$Species == selected_species]),
                  colour = "red") +
    xlab("Log of Predator/Prey mass ratio")  +
    expand_limits(x = c(0, 30))
```
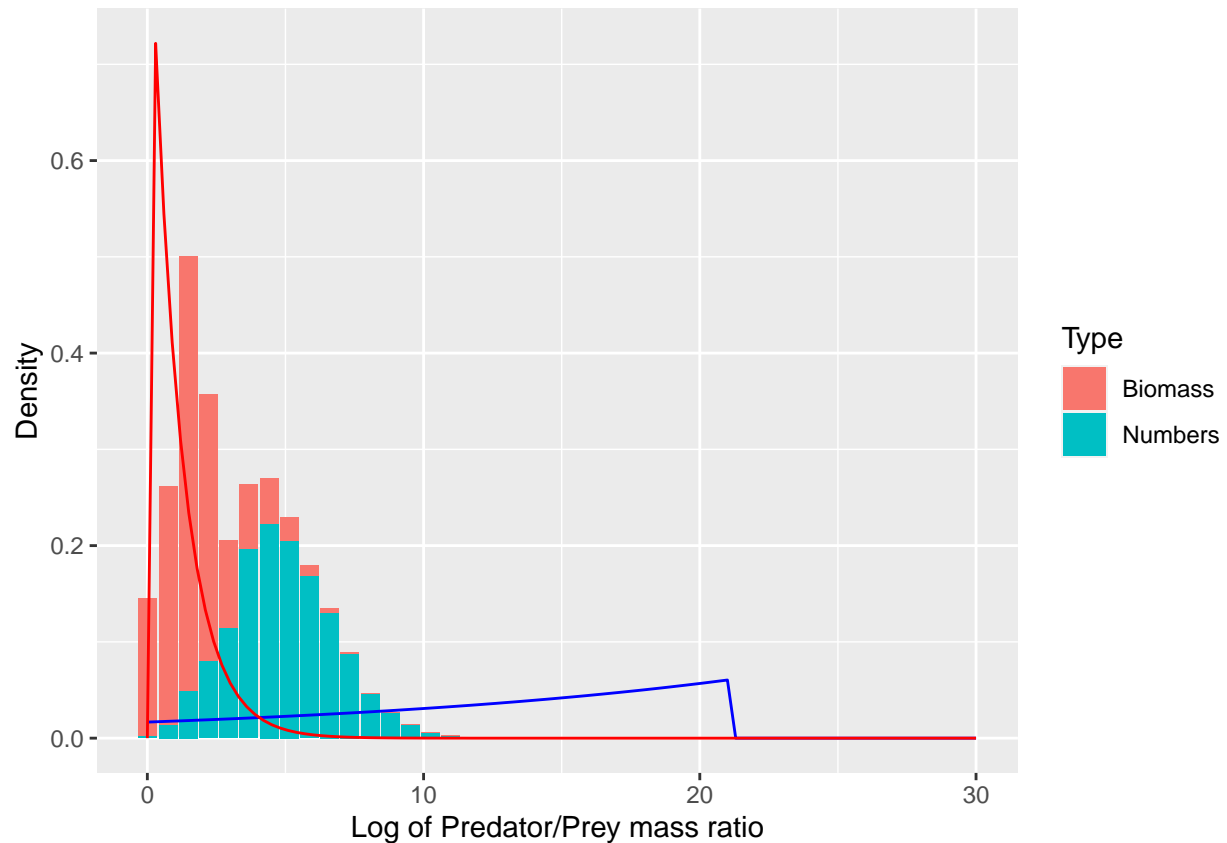
Herrin

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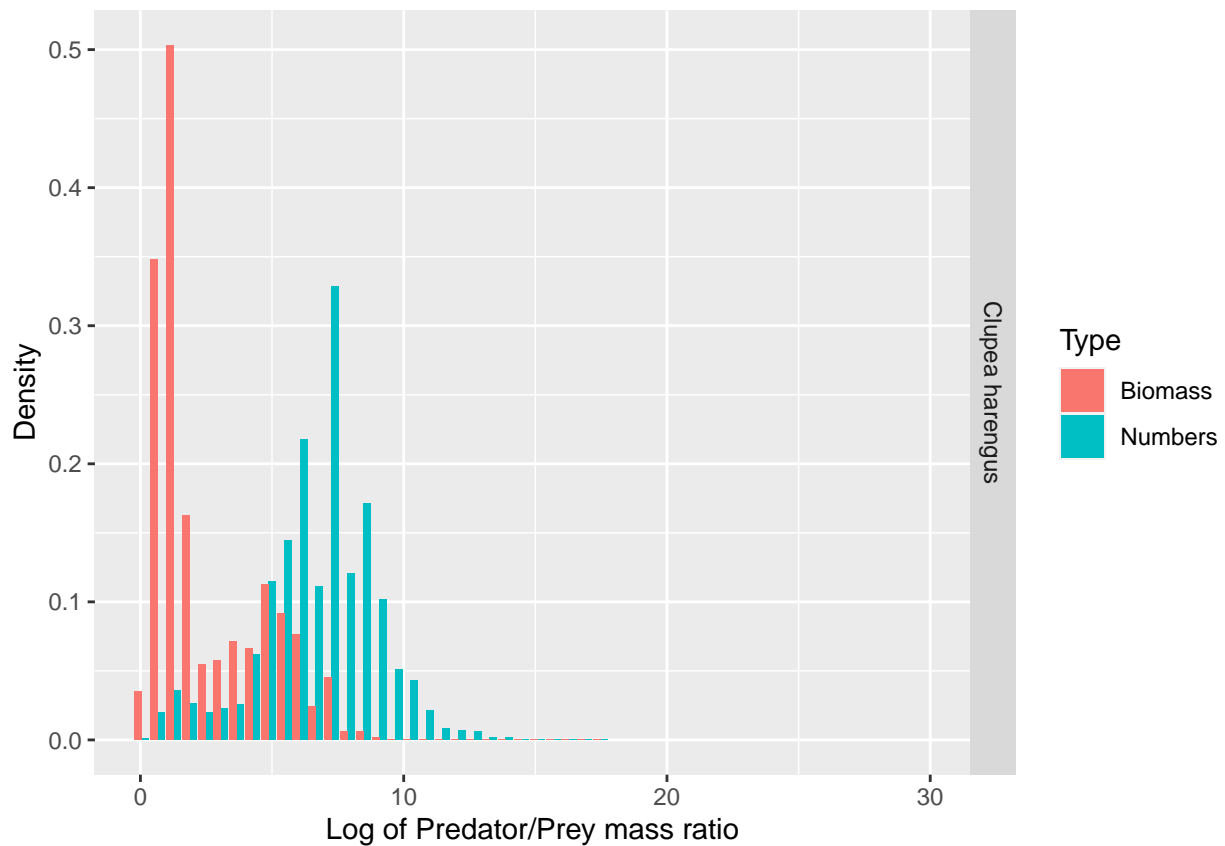
We convert this into the long table format preferred by ggplot2.

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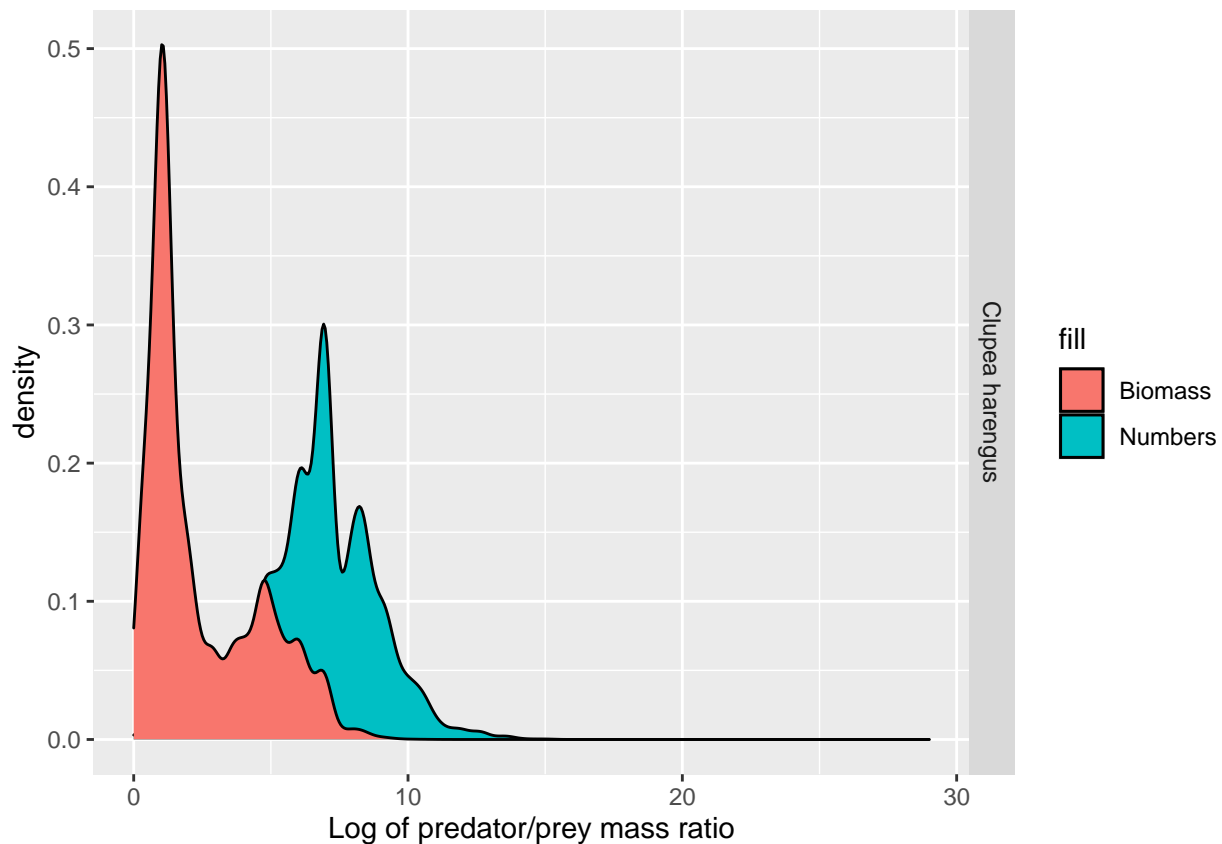
Histograms

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))
```


13

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))
```
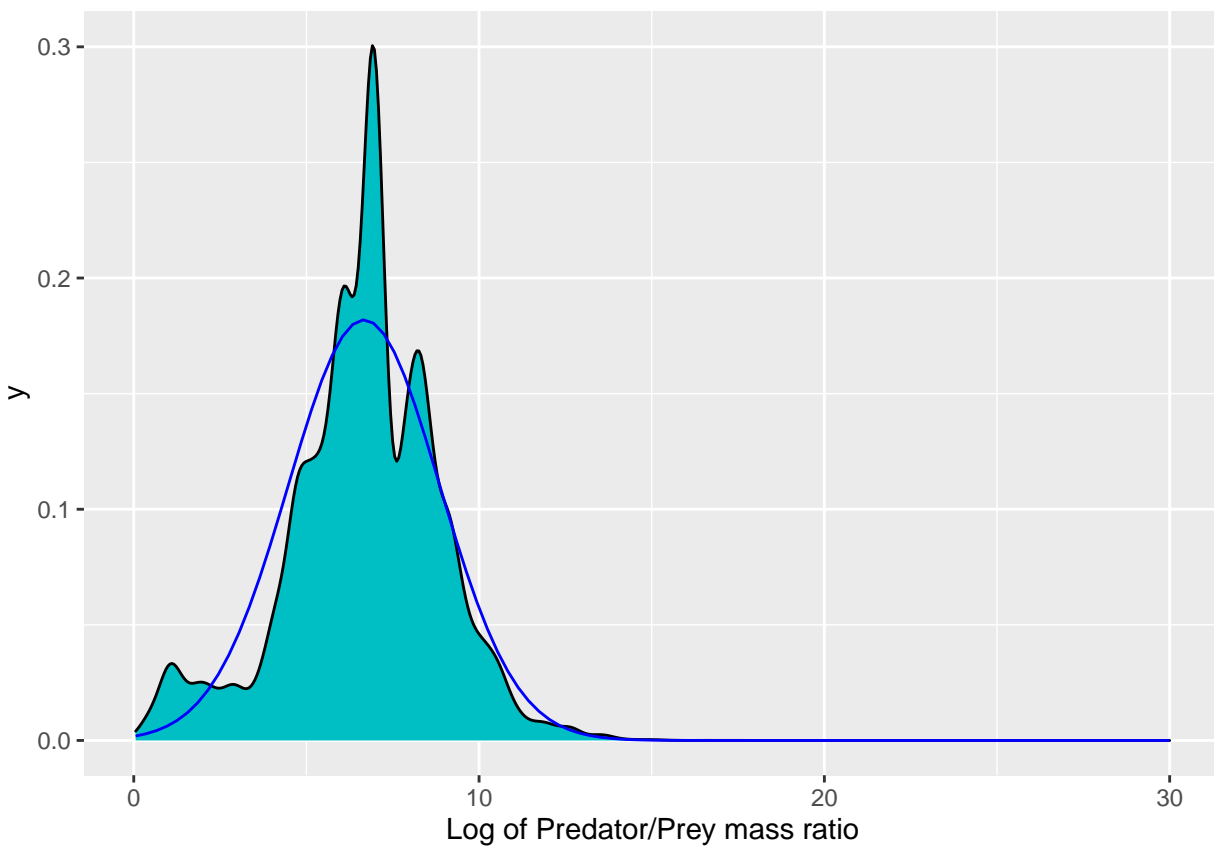

Gaussian Distribution fit

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))
```


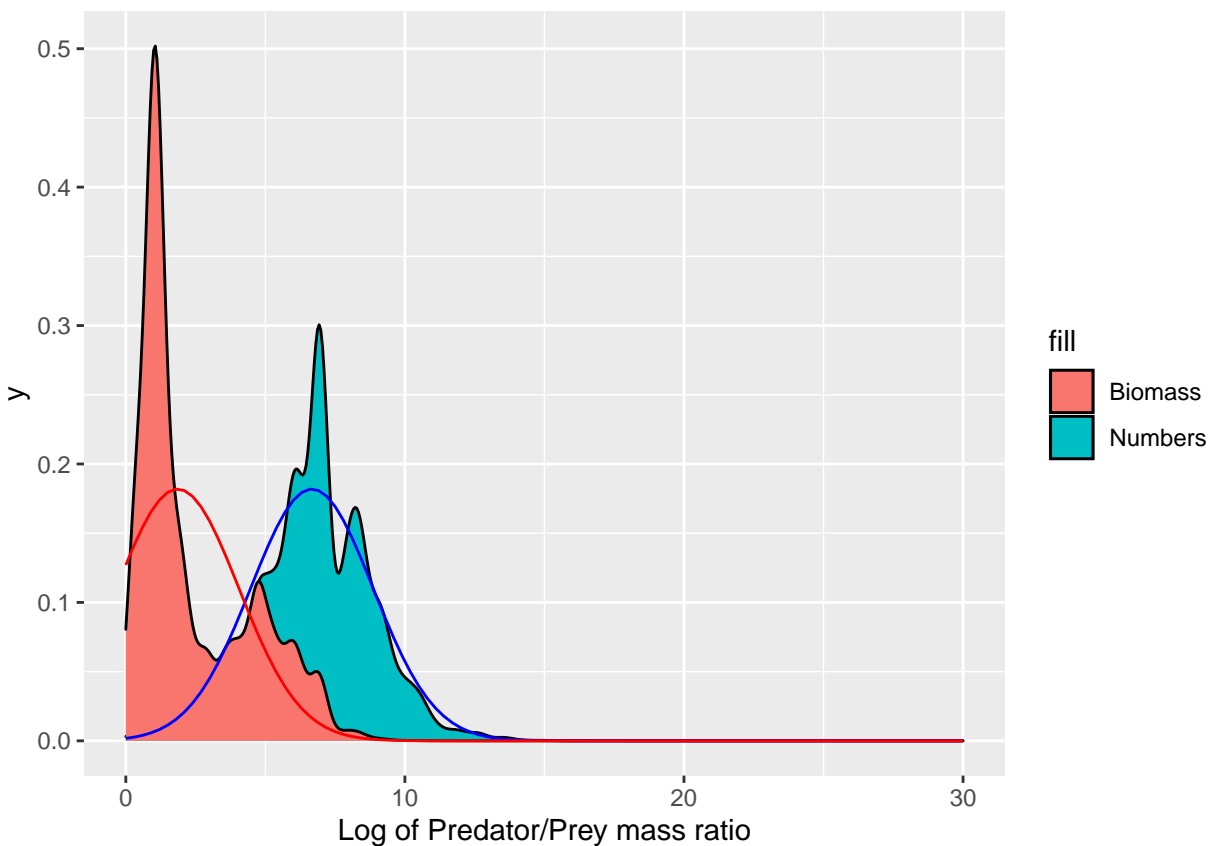Now adding biomass

```
stomach %>%
  ggplot() +
  geom_density(aes(l, weight = weight_numbers,
                   fill = "Numbers"),
               adjust = adjust) +
  geom_density(aes(l, weight = weight_biomass,
                   fill = "Biomass"),
```

15

```
                   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"),
```
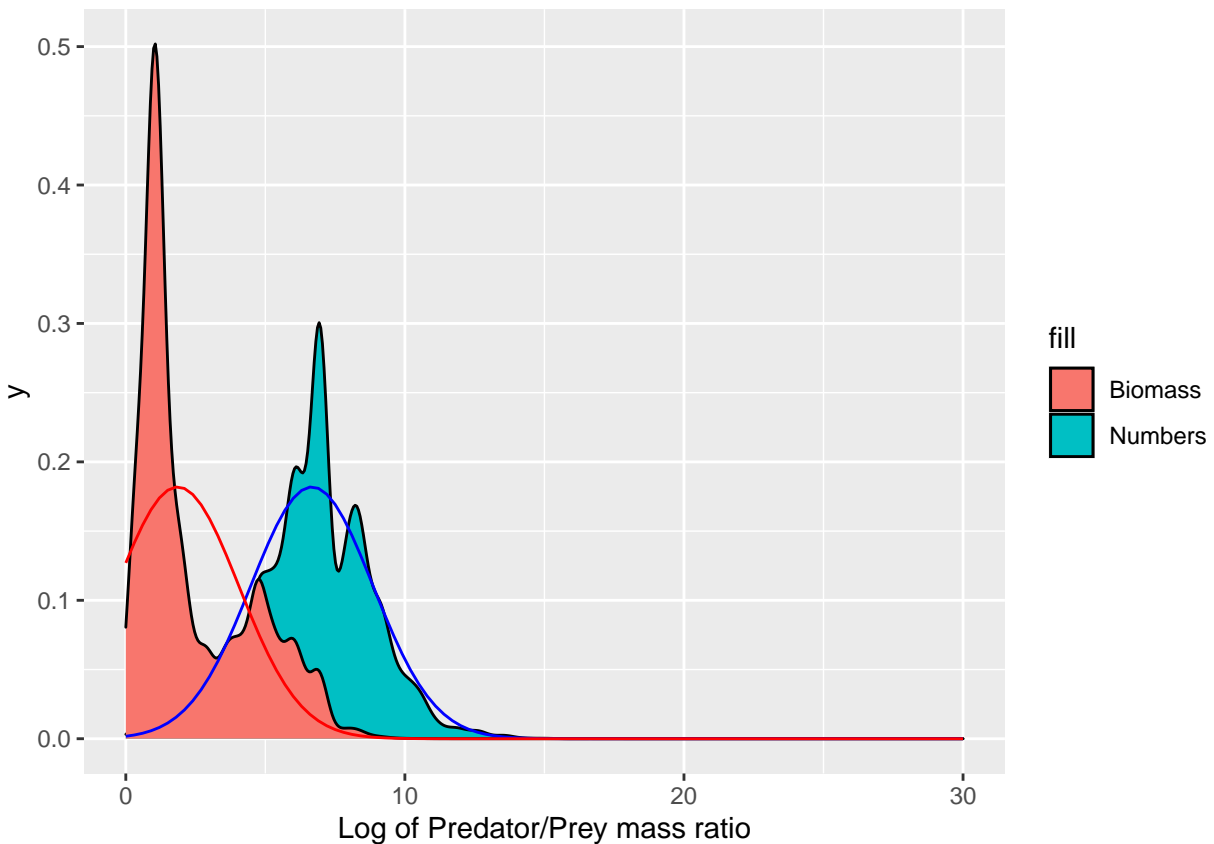
```
                adjust = adjust) +
    xlab("Log of Predator/Prey mass ratio") +
    stat_function(fun = dnorm,
                  args = list(mean = fit$mean,
                              sd = fit$sd),
                  colour = "blue") +
    stat_function(fun = dnorm,
                  args = list(mean = fit$mean - fit$sd^2,
                              sd = fit$sd),
                  colour = "red") +
    expand_limits(x = c(0, 30))
```
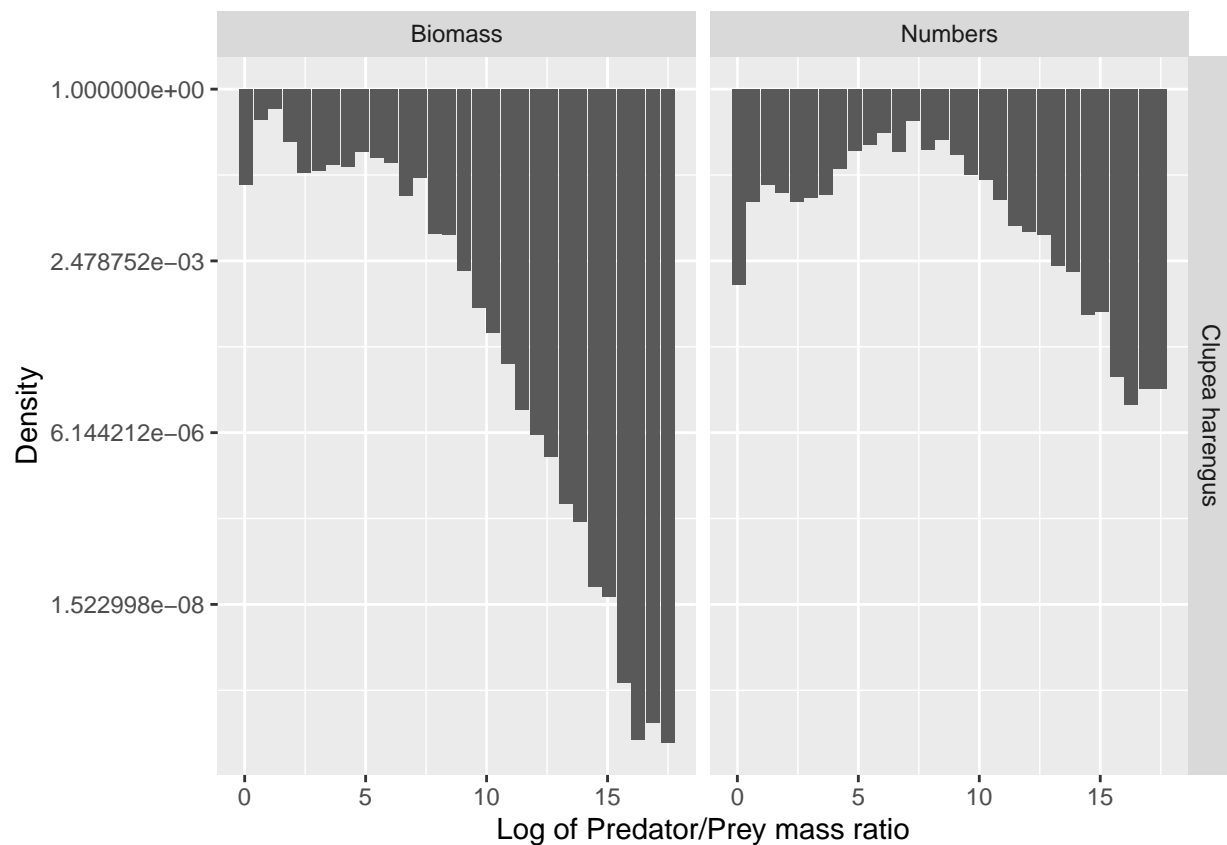

Fitting the Exponential DIstribution

Replot the histograms but with the logarithmic y-axis

```
binned_stomach %>%
  ggplot(aes(l, Density)) +
  geom_col() +
  facet_grid(Species ~ Type, scales = "free_y") +
  xlab("Log of Predator/Prey mass ratio") +
  scale_y_continuous(trans = "log")
```
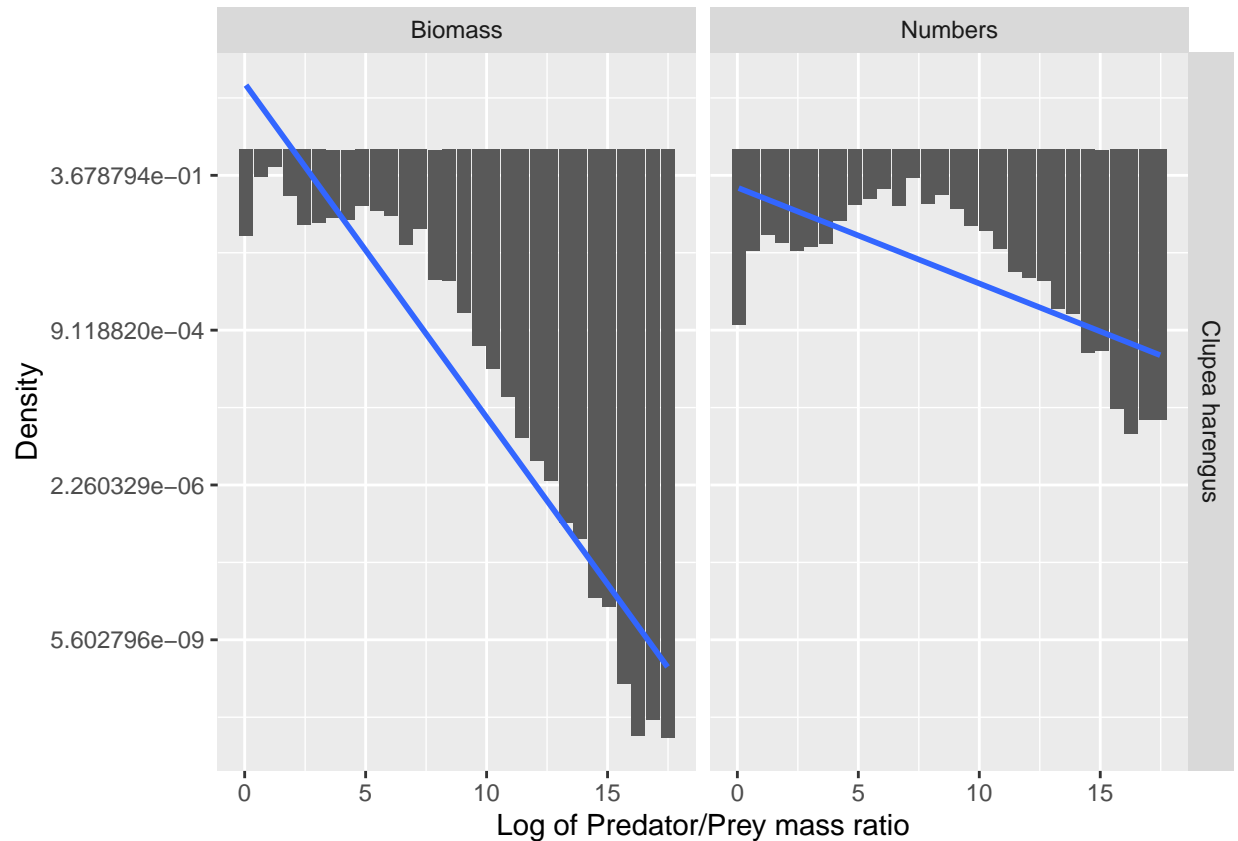
17

Fitting the truncated exponential distribution

```
binned_stomach %>%
  ggplot(aes(l, Density)) +
  geom_col() +
  geom_smooth(method = "lm", se = FALSE) +
  facet_grid(Species ~ Type, scales = "free_y") +
  xlab("Log of Predator/Prey mass ratio") +
  scale_y_continuous(trans = "log")
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
binned_stomach %>%
  group_by(Species, Type) %>%
  group_modify(~ broom::tidy(lm(log(Density) ~ l, data = .x))) %>%
  filter(term == "l")
```

```
## # A tibble: 2 x 7
## # Groups:   Species, Type [2]
##   Species          Type     term   estimate std.error statistic  p.value
##   <chr>            <chr>    <chr>      <dbl>     <dbl>     <dbl>    <dbl>
## 1 Clupea harengus Biomass l          -1.29    0.0883     -14.7 1.17e-14
## 2 Clupea harengus Numbers l         -0.372    0.0764      -4.87 3.98e- 5
```

Maximum Likelihood fit

```
est <- stomach %>%
  group_by(Species) %>%
  summarise(lbar = weighted.mean(l, weight_numbers),
            lmax = max(l),
            lmin = min(l),
            alpha = 1/(lmax - lbar))
est
```

```
## # A tibble: 1 x 5
```

```
##   Species          lbar  lmax   lmin  alpha
##   <chr>           <dbl> <dbl>  <dbl>  <dbl>
## 1 Clupea harengus  6.67  17.5 0.0544 0.0925
```

```r
dnumbers <- function(l, alpha, lmax) {
  d <- as.numeric(l <= lmax)
  d[d > 0] <- dexp(lmax - l[d > 0], alpha)
  return(d)
}
dbiomass <- function(l, alpha, lmin) {
  d <- as.numeric(l >= lmin)
  d[d > 0] <- dexp(l[d > 0] - lmin, 1 - alpha)
  return(d)
}
selected_species <- "Clupea harengus"
binned_stomach %>%
  filter(Species == selected_species) %>%
  ggplot() +
    geom_col(aes(l, Density, fill = Type)) +
    stat_function(fun = dnumbers,
                  args = list(alpha = est$alpha[est$Species == selected_species],
                              lmax = est$lmax[est$Species == selected_species]),
                  colour = "blue") +
    stat_function(fun = dbiomass,
                  args = list(alpha = est$alpha[est$Species == selected_species],
                              lmin = est$lmin[est$Species == selected_species]),
                  colour = "red") +
    xlab("Log of Predator/Prey mass ratio")  +
    expand_limits(x = c(0, 30))
```