Merluccius merluccius

title: “European Hake” output: html_document date: “2024-07-17” —

```
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
```

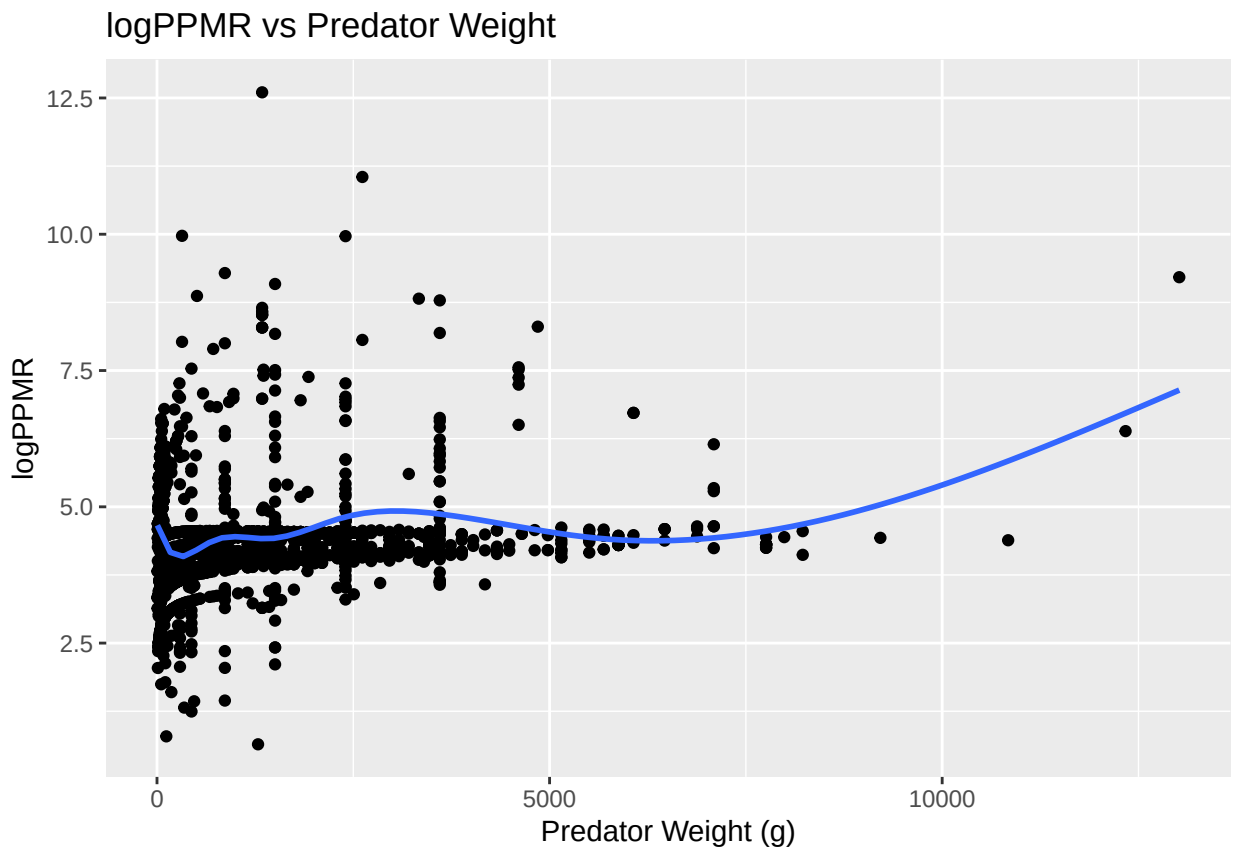
```
##
## Attaching package: 'bbmle'
```

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

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

sprat <- stom_df%>%filter(pred_taxa=="Merluccius merluccius")
```

```
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")
```

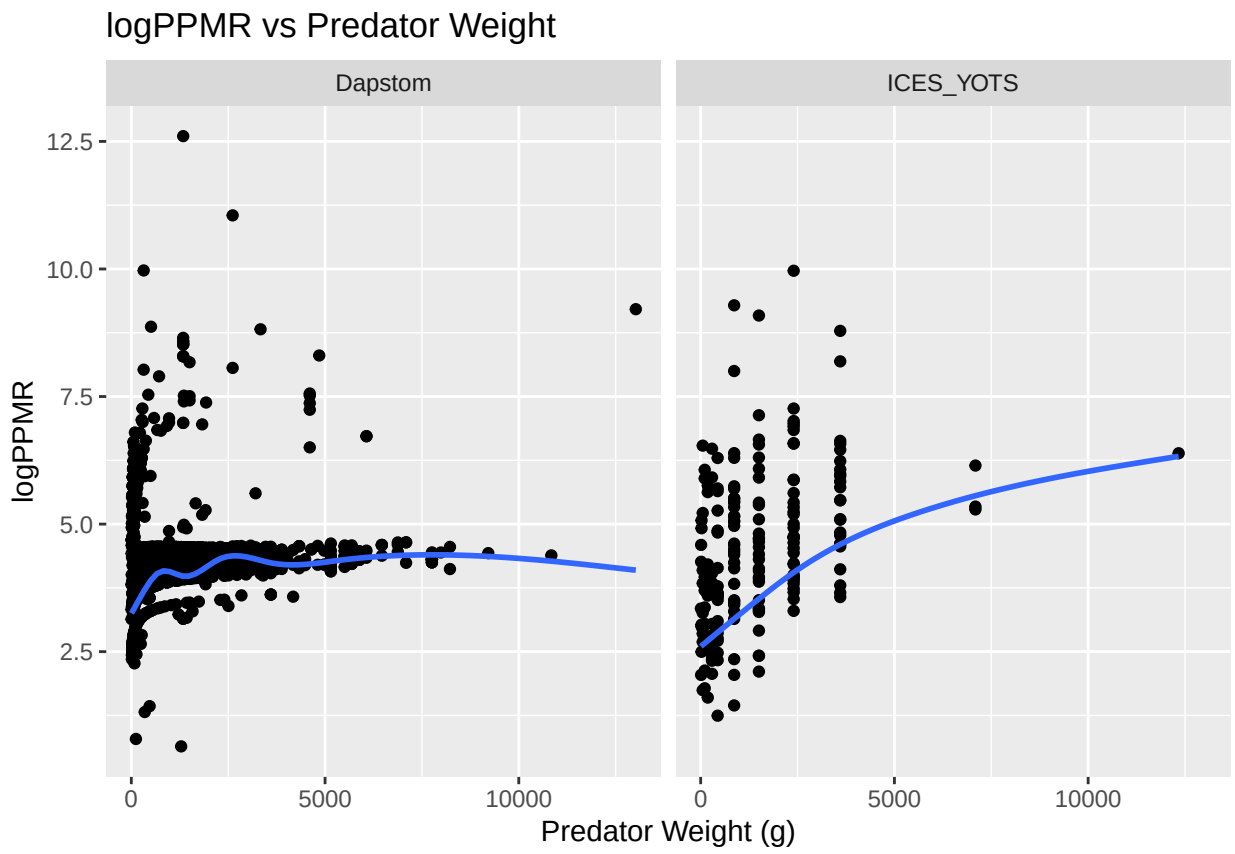
```
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
```



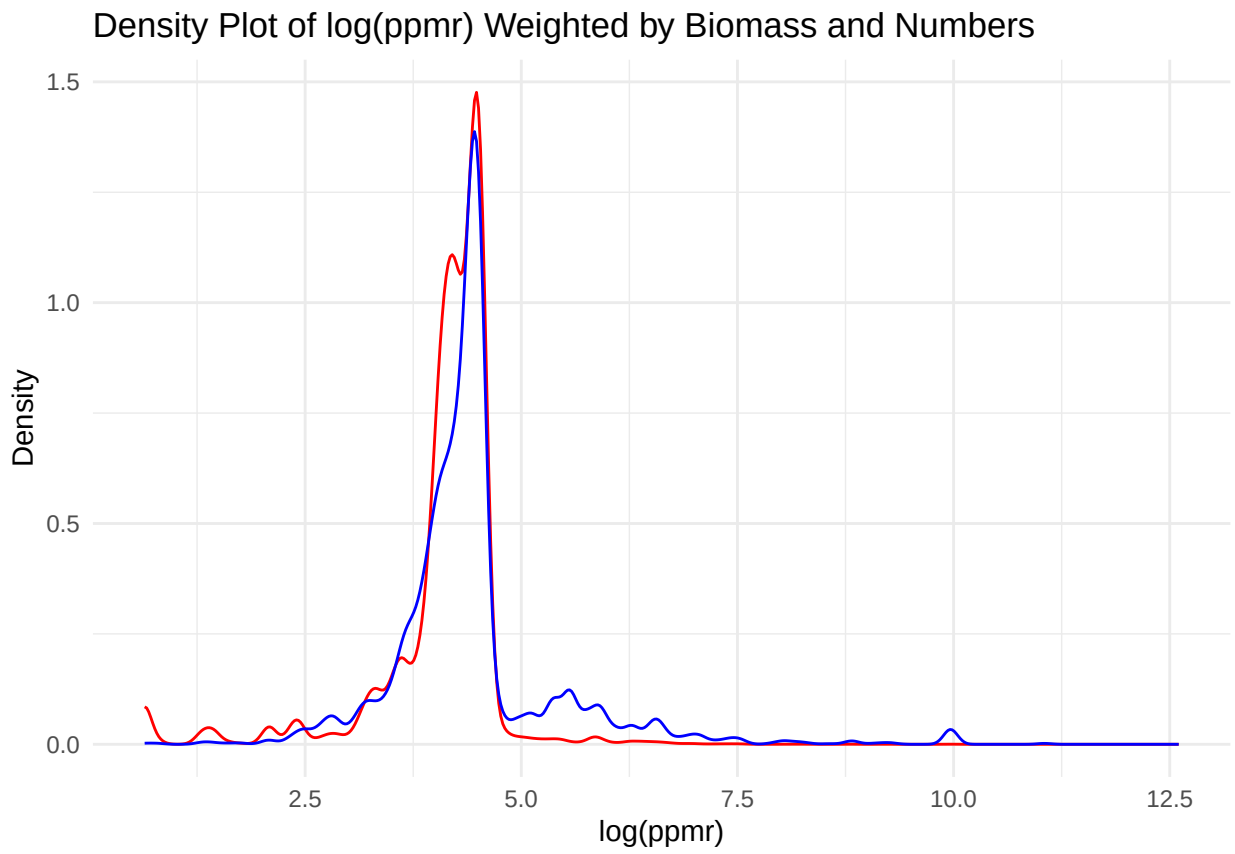

```
dig <- 1

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")’
```




```
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()
```



This looks like a single gaussian?

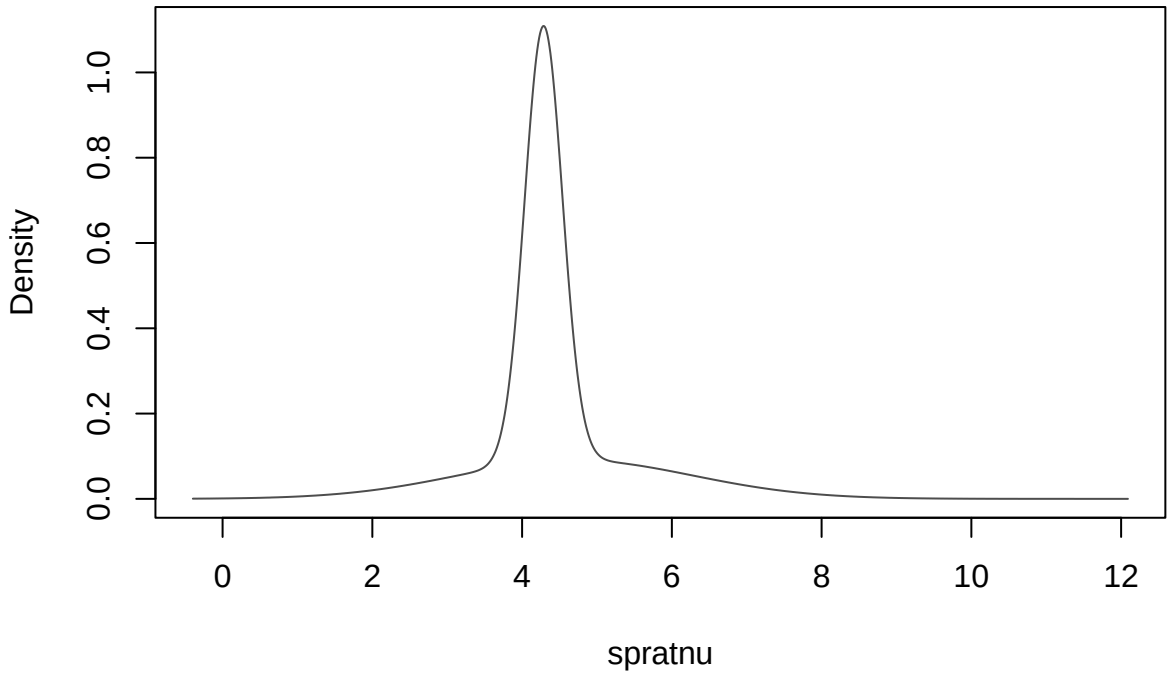
```
library(mclust)
```

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

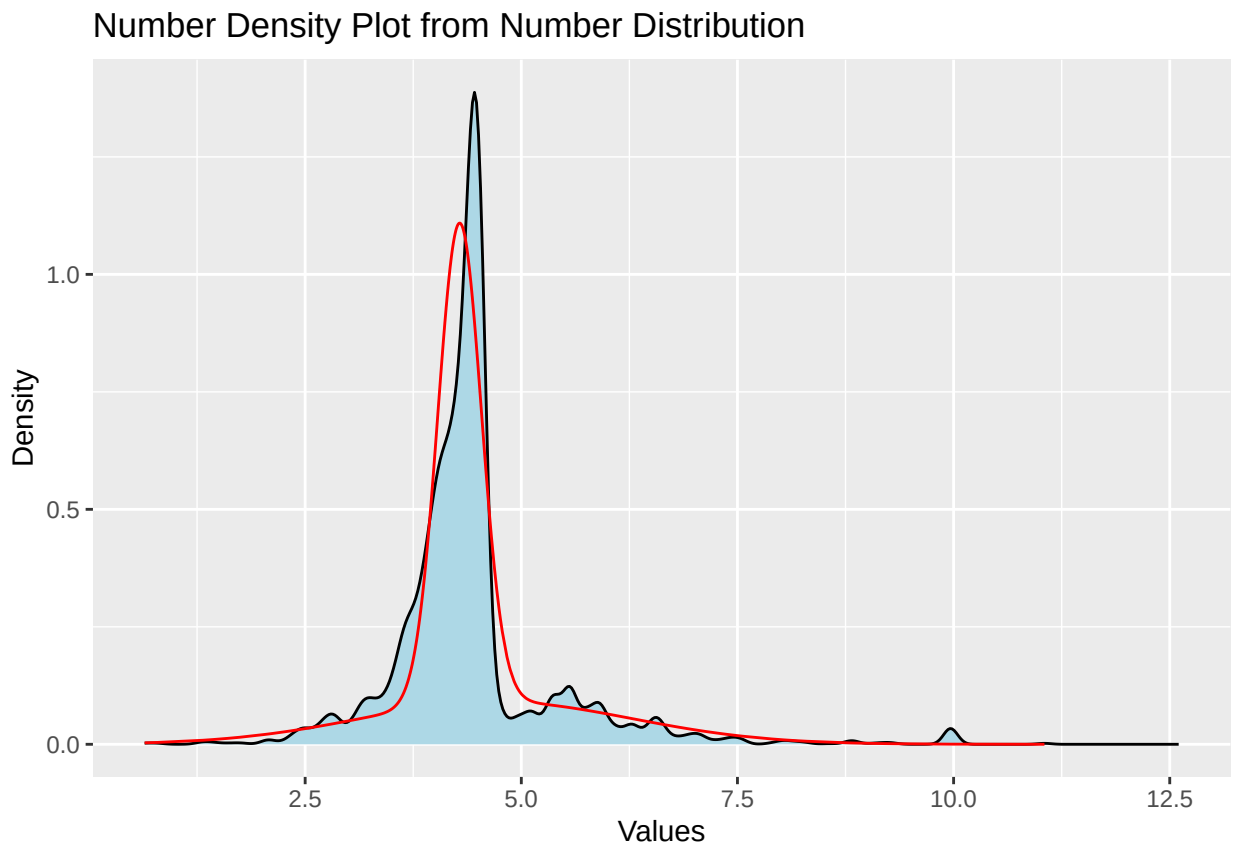
```
## Package 'mclust' version 6.1.1
## Type 'citation("mclust")' for citing this R package in publications.
```

```
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)
```

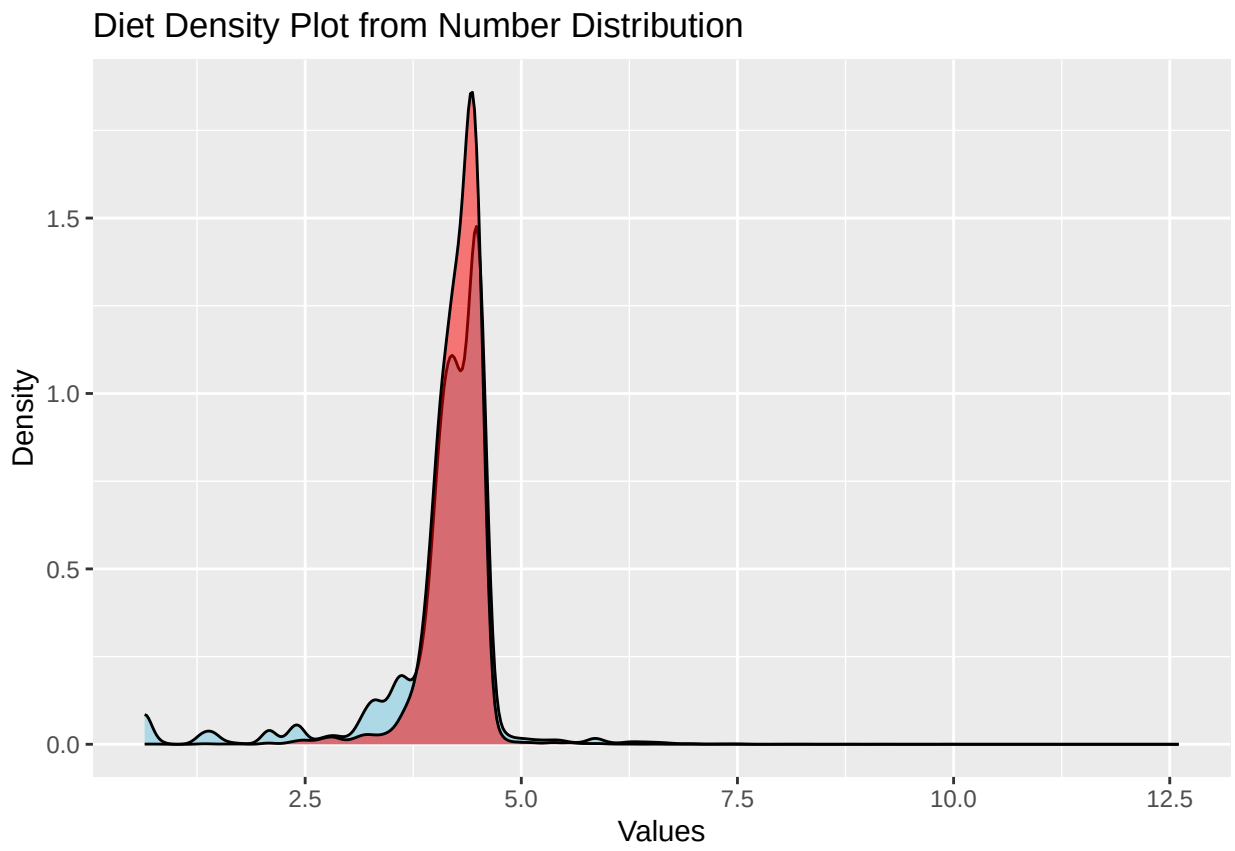



```
dplot <- data.frame(x=gmm[["data"]], density=gmm[["density"]])


(numbfit <- 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"))
```



Ok 2 gaussians fits nicely. It does under estimate the peak however.

```
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)

(numbfitbio <- 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"))
```



This is a very nice shift I think. I think this works.

```
print(gmm$parameters)
```

```
## $pro
## [1] 0.6450399 0.3549601
##
## $mean
##        1        2
## 4.286325 4.710199
##
## $variance
## $variance$modelName
## [1] "V"
##
## $variance$d
## [1] 1
##
## $variance$G
## [1] 2
##
## $variance$sigmasq
## [1] 0.06341901 2.44856948
##
## $variance$scale
## [1] 0.06341901 2.44856948
```