Dataanalysis

2023-02-17

Reading in data

Read in the data, and set the data types to the correct types Drop index number from the pandas dataframe and NCBI.tax.ID

```
D_tmp <- read_csv("../data/dataset_joined.csv",show_col_types = FALSE)
```

```
## New names:
## * `` -> `...1`
```

```
D_tmp <- mutate(D_tmp, across(species:PH.range, factor))
D_tmp <- mutate(D_tmp, across(GC.content:total_genes, as.double))
D_tmp <- select(D_tmp, !c(NCBI.tax.ID,...1))
```

Split up the dataset in two, one with the specific antibiotic resistance information and another with the rest of the data

```
# With ar
D_ar <- select(D_tmp, !antibiotics & species:spiramycin.II)
# Without ar
D <- select(D_tmp, !lincomycin:spiramycin.II)
```

Summary Statistics

```
sum <- summary(D)
sum
```

```
##                              species     antibiotics                family
##  Abiotrophia defectiva           :   1   NR:3710     Pseudomonadaceae : 136
##  Abyssalbus ytuae                :   1   R : 294     Bacillaceae      : 130
##  Abyssicoccus albus              :   1               Streptomycetaceae: 130
##  Acetilactobacillus jinshanensis:   1               Flavobacteriaceae: 119
##  Acetivibrio clariflavus         :   1               Lactobacillaceae :  98
##  Acetivibrio saccincola          :   1               Burkholderiaceae :  97
##  (Other)                         :3998               (Other)          :3294
##                order                       class                phylum
##  Caryophanales   : 305   Gammaproteobacteria:895   Pseudomonadota:1803
##  Lactobacillales : 235   Actinomycetes      :668   Bacillota     : 776
##  Enterobacterales: 208   Bacilli            :540   Actinomycetota: 703
##  Burkholderiales : 206   Alphaproteobacteria:432   Bacteroidota  : 352
```

```
##  Mycobacteriales : 190   Betaproteobacteria :311   Mycoplasmatota: 100
##  Pseudomonadales : 188   Clostridia         :196   Spirochaetota :  45
##  (Other)         :2672   (Other)            :962   (Other)       : 225
##       domain     motility       gram.stain       growth
##  Bacteria:4004   no  : 550   negative: 807   37.0   : 533
##                  yes : 513   positive: 422   30.0   : 342
##                  NA's:2941   variable:  12   28.0   : 167
##                              NA's    :2763   29.0   : 152
##                                              25.0   :  99
##                                              (Other):1124
##                                              NA's   :1587
##              genus                   oxygen.tolerance        PH.range
##  Pseudomonas    : 127   aerobe              :1097    acidophile :  39
##  Streptomyces   : 126   anaerobe            : 532    alkaliphile: 503
##  Corynebacterium:  85   microaerophile      : 355    NA's       :3462
##  Vibrio         :  79   facultative anaerobe: 175
##  Streptococcus  :  55   obligate aerobe     :  45
##  Mycobacterium  :  50   (Other)             :  33
##  (Other)        :3482   NA's                :1767
##    GC.content     Total.samples     soil.counts      aquatic.counts
##  Min.   :   1.0   Min.   :   1.0   Min.   :   1.00   Min.   :   1.0
##  1st Qu.:   1.0   1st Qu.:   1.0   1st Qu.:   1.00   1st Qu.:   1.0
##  Median : 146.4   Median : 290.0   Median :  98.98   Median : 129.6
##  Mean   : 417.2   Mean   : 823.5   Mean   : 348.40   Mean   : 445.4
##  3rd Qu.: 740.0   3rd Qu.:1426.1   3rd Qu.: 590.33   3rd Qu.: 802.0
##  Max.   :1855.0   Max.   :3522.0   Max.   :1618.00   Max.   :1925.0
##
##   plant.counts        optimum             n16              div
##  Min.   :   1.00   Min.   :  1.000   Min.   : 1.000   Min.   :  0.0000
##  1st Qu.:   1.00   1st Qu.:  1.000   1st Qu.: 2.000   1st Qu.:  0.0000
##  Median :  41.97   Median :  1.000   Median : 4.000   Median :  0.6931
##  Mean   : 206.71   Mean   :  3.247   Mean   : 4.629   Mean   :  3.2324
##  3rd Qu.: 330.05   3rd Qu.:  1.000   3rd Qu.: 6.000   3rd Qu.:  3.0237
##  Max.   :1033.00   Max.   :134.000   Max.   :37.000   Max.   :341.6293
##
##    gc_percent    genome_components  chromosomes     total_seq_length
##  Min.   :23.00   Min.   : 1.000    Min.   : 1.000   Min.   :  579782
##  1st Qu.:40.00   1st Qu.: 1.000    1st Qu.: 1.000   1st Qu.: 2779606
##  Median :52.50   Median : 1.000    Median : 1.000   Median : 4041968
##  Mean   :51.83   Mean   : 1.803    Mean   : 1.789   Mean   : 4275199
##  3rd Qu.:64.00   3rd Qu.: 2.000    3rd Qu.: 2.000   3rd Qu.: 5294559
##  Max.   :76.00   Max.   :29.000    Max.   :29.000   Max.   :13646761
##
##     genes_nc       genes_coding      pseudogenes      total_genes
##  Min.   : 33.00   Min.   :  493.8   Min.   :   1.0   Min.   :  562
##  1st Qu.: 58.00   1st Qu.: 2489.0   1st Qu.:  28.0   1st Qu.: 2615
##  Median : 71.90   Median : 3593.0   Median :  54.0   Median : 3751
##  Mean   : 79.53   Mean   : 3764.8   Mean   :  85.8   Mean   : 3930
##  3rd Qu.: 92.38   3rd Qu.: 4651.0   3rd Qu.: 100.8   3rd Qu.: 4876
##  Max.   :333.00   Max.   :11023.0   Max.   :1257.0   Max.   :11242
##
```

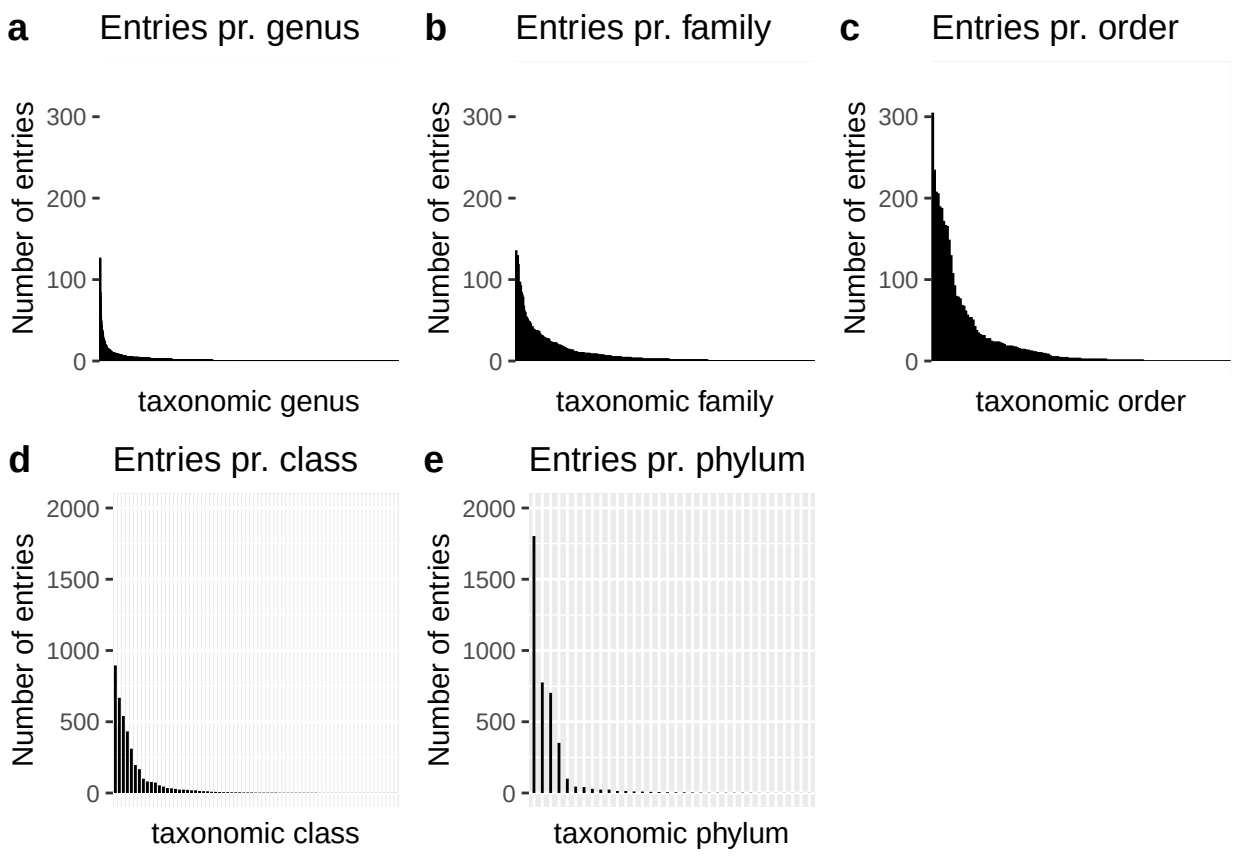
We have NA for the following columns: motility,gram.stain,growth,oxygen.tolerance,PH.range

Looking at data distribution

For different taxonomic scales

TODO: also make plot of just n species

```
# Function to plot how often each tax is seen
taxplot <- function(D, tax, name,ylimit){
  plot <- D %>%
    group_by({{tax}}) %>%
    summarise(n=n()) %>%
    mutate(tax_ordered = fct_reorder({{tax}},desc(n))) %>%
    ggplot() +
    geom_segment(aes(x=tax_ordered, xend = tax_ordered,y=0, yend=n)) +
    xlab(glue("taxonomic {name}")) + ylab("Number of entries") +
    theme(axis.text.x=element_blank(),axis.ticks.x=element_blank()) +
    ggtitle(glue("Entries pr. {name}")) +
    ylim(0,ylimit)
  return(plot)
}

# Plloting the different taxs
p1 <- taxplot(D, family, "family",350)
p2 <- taxplot(D, order, "order",350)
p3 <- taxplot(D, class, "class",2000)
p4 <- taxplot(D, phylum, "phylum",2000)
p5 <- taxplot(D, genus, "genus",350)
plot_grid(p5,p1, p2, p3, p4,labels ="auto")
```



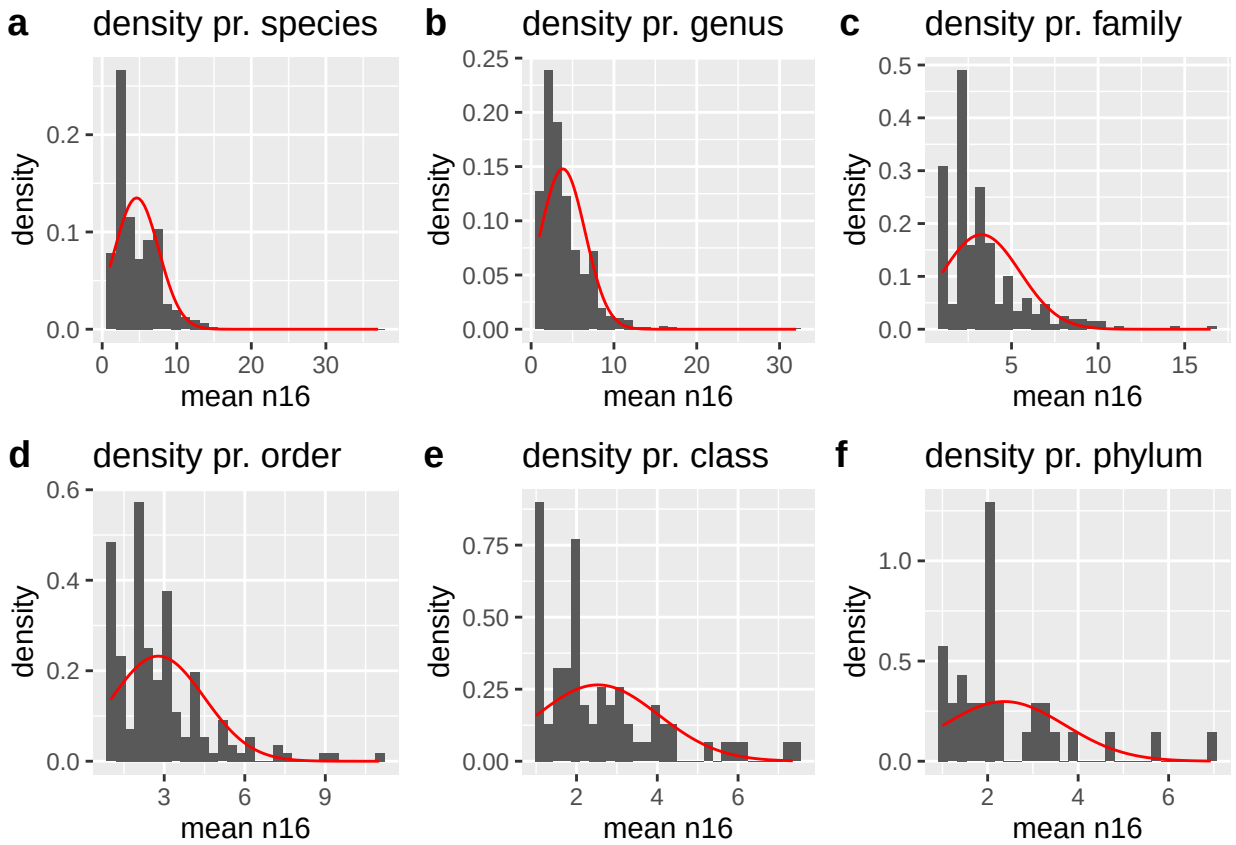

```
#gridExtra::grid.arrange(p5,p1,p2,p3,p4)
```

The goal of this part is to figure out how to take into consideration that some strains are sequences a lot more than others. I have already calculated the mean/mode of each value at the species level, but it seems that we still have an unequal distribution. Furthermore it seems that nomatter how high the taxonomic scale we go, There is still an unequal distribution. Even though wee took into consideration that some species are sequences more we would still except an unequeal distribution. Based on this and the fact that information is lost for each "step" on the taxonomic scale I have choosen to go with the speceis level It is relevant as when we for example look at the distribution of numbers of 16s genes, more sequences genes will have a higher weight.

```
taxboxplot <- function(D, tax, n16_var, name){
  D_new <- D %>%
    group_by({{tax}}) %>%
    summarise(mean_value=mean({{n16_var}}))
  plot <- ggplot(D_new) +
    geom_histogram(aes(x=mean_value, y=after_stat(density))) +
    ggtitle(glue("density pr. {name}"))+
    xlab("mean n16") + ylab("density") +
    stat_function(fun = dnorm, col = "red", args = list(mean = mean(D_new$mean_value), sd = sd(D_new$mea
  return(plot)
}


# Plloting the different taxs
```

```
to_plot <- "n16"
p0 <- taxboxplot(D, species, .data[[to_plot]],"species")
p1 <- taxboxplot(D, genus, .data[[to_plot]],"genus")
p2 <- taxboxplot(D, family,.data[[to_plot]], "family")
p3 <- taxboxplot(D, order, .data[[to_plot]],"order")
p4 <- taxboxplot(D, class,.data[[to_plot]], "class")
p5 <- taxboxplot(D, phylum,.data[[to_plot]], "phylum")

plot_grid(p0,p1, p2, p3, p4,p5,labels ="auto")
```

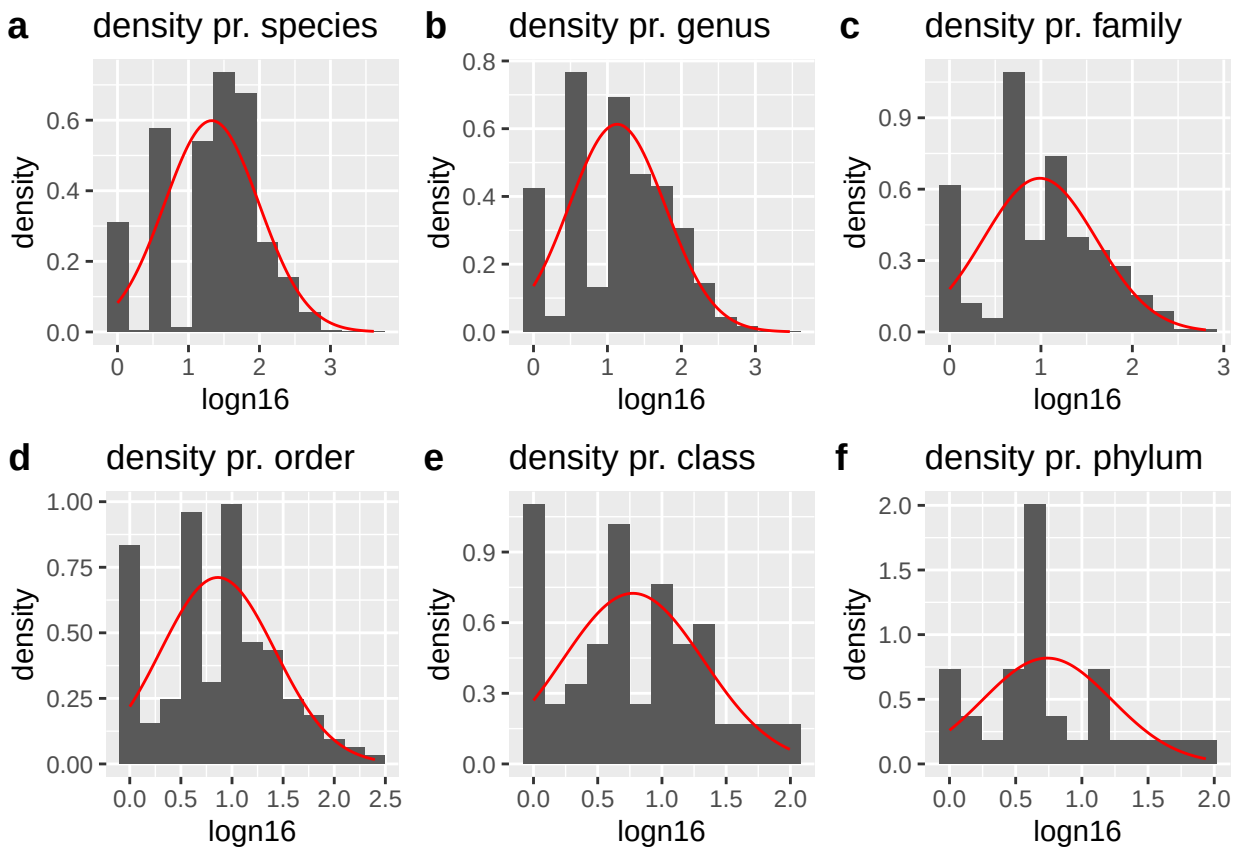
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



But the most important thing is the distribution. Here we can see that for species it looks a bit wierd, it looks better for genus Lets have a look at the log transformed

```
# Not working:: Error in if (!is.finite(log_num)) { : the condition has length > 1
L2log <- function(num){
  log_num = log2(num)
  print(log_num)
  if (!is.finite(log_num)){return(0)}
```

```
  else{
  return (log_num)}
}

functaxboxplot <- function(D, tax, n16_var, name,func){
  D_new <- D %>%
    group_by({{tax}}) %>%
    summarise(mean_value=mean({{n16_var}}))
  plot <- ggplot(D_new) +
    geom_histogram(aes(x=func(mean_value), y=after_stat(density)),bins = 13) +
    ggtitle(glue("density pr. {name}"))+
    xlab("logn16") + ylab("density") +
    stat_function(fun = dnorm, col = "red", args = list(mean = mean(func(D_new$mean_value)), sd = sd(fu
  return(plot)
}


# Plloting the different taxs
to_plot <- "n16"
p0 <- functaxboxplot(D, (species), .data[[to_plot]],"species",log)
p1 <- functaxboxplot(D, (genus), .data[[to_plot]],"genus",log)
p2 <- functaxboxplot(D, (family),.data[[to_plot]], "family",log)
p3 <- functaxboxplot(D, (order), .data[[to_plot]],"order",log)
p4 <- functaxboxplot(D, (class),.data[[to_plot]], "class",log)
p5 <- functaxboxplot(D, (phylum),.data[[to_plot]], "phylum",log)

plot_grid(p0,p1, p2, p3, p4,p5,labels ="auto")
```

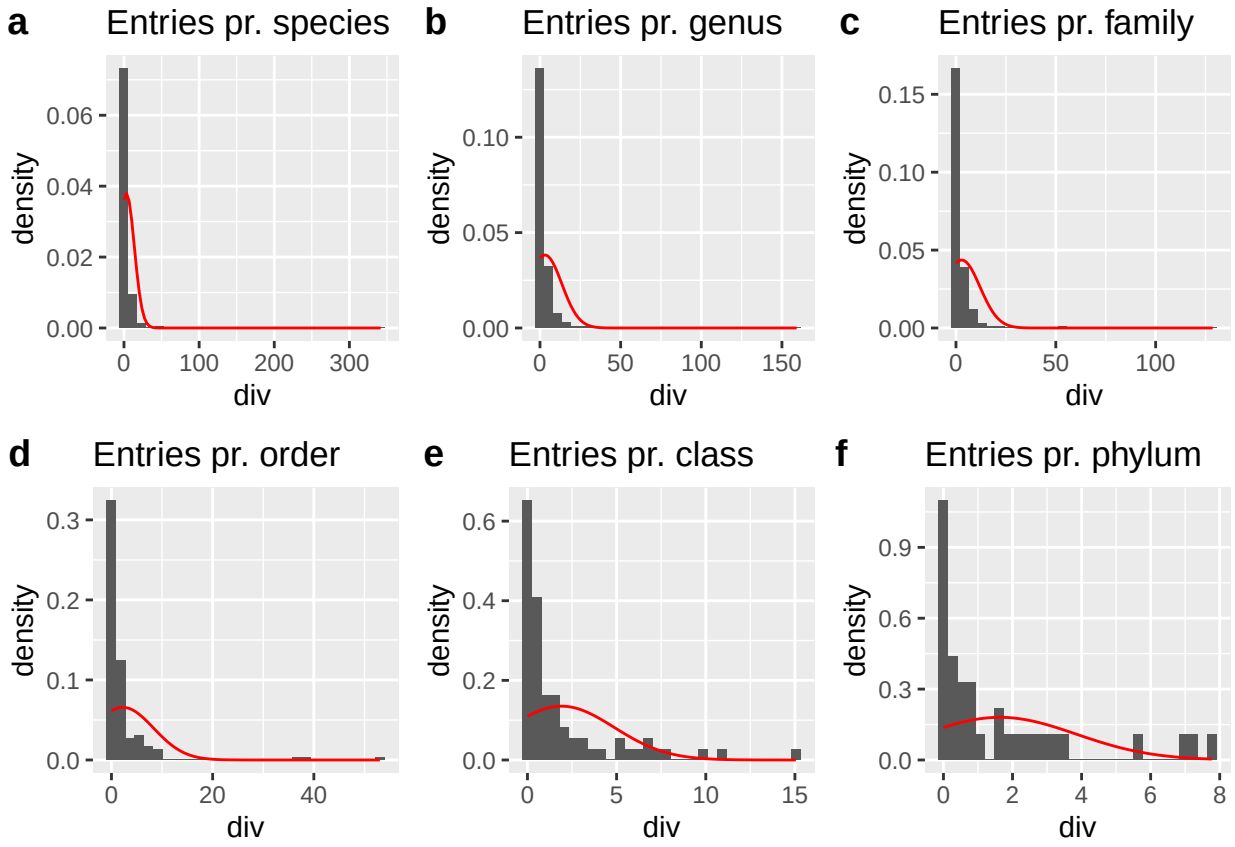


Here the distributions look ok for species. And again to not lose information im going with species.s

D Lets have a look at the distributions for D

```
taxboxplot <- function(D, tax, n16_var, name){
  D_new <- D %>%
    group_by({{tax}}) %>%
    summarise(mean_value=mean({{n16_var}}))
  plot <- ggplot(D_new) +
    geom_histogram(aes(x=mean_value, y=after_stat(density))) +
    ggtitle(glue("Entries pr. {name}"))+
    xlab("div") + ylab("density") +
    stat_function(fun = dnorm, col = "red", args = list(mean = mean(D_new$mean_value), sd = sd(D_new$mea
  return(plot)
}

to_plot <- "div"
p0 <- taxboxplot(D, species, .data[[to_plot]],"species")
p1 <- taxboxplot(D, genus, .data[[to_plot]],"genus")
p2 <- taxboxplot(D, family,.data[[to_plot]], "family")
p3 <- taxboxplot(D, order, .data[[to_plot]],"order")
p4 <- taxboxplot(D, class,.data[[to_plot]], "class")
p5 <- taxboxplot(D, phylum,.data[[to_plot]], "phylum")

plot_grid(p0,p1, p2, p3, p4,p5,labels ="auto")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

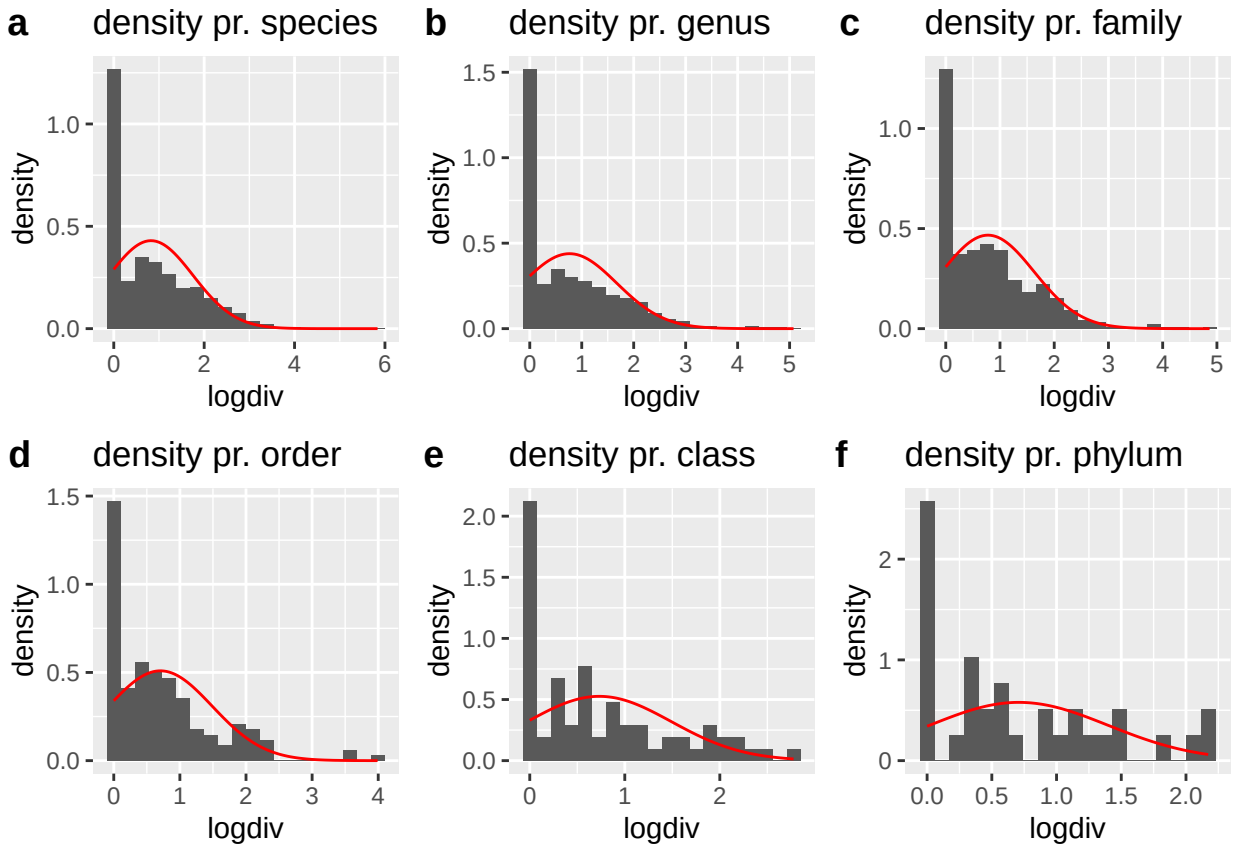


They do not look normal distributed.

Looking at n16 and var

```
functaxboxplot <- function(D, tax, n16_var, name,func){
  D_new <- D %>%
    group_by({{tax}}) %>%
    summarise(mean_value=mean({{n16_var}}))
  plot <- ggplot(D_new) +
    geom_histogram(aes(x=func(mean_value), y=after_stat(density)),bins = 20) +
    ggtitle(glue("density pr. {name}"))+
    xlab("logdiv") + ylab("density") +
    stat_function(fun = dnorm, col = "red", args = list(mean = mean(func(D_new$mean_value)), sd = sd(fu
  return(plot)
}

# Plloting the different taxs
```

```
to_plot <- "div"
p0 <- functaxboxplot(D, (species), .data[[to_plot]],"species",log1p)
p1 <- functaxboxplot(D, (genus), .data[[to_plot]],"genus",log1p)
p2 <- functaxboxplot(D, (family),.data[[to_plot]], "family",log1p)
p3 <- functaxboxplot(D, (order), .data[[to_plot]],"order",log1p)
p4 <- functaxboxplot(D, (class),.data[[to_plot]], "class",log1p)
p5 <- functaxboxplot(D, (phylum),.data[[to_plot]], "phylum",log1p)

plot_grid(p0,p1, p2, p3, p4,p5,labels ="auto")
```

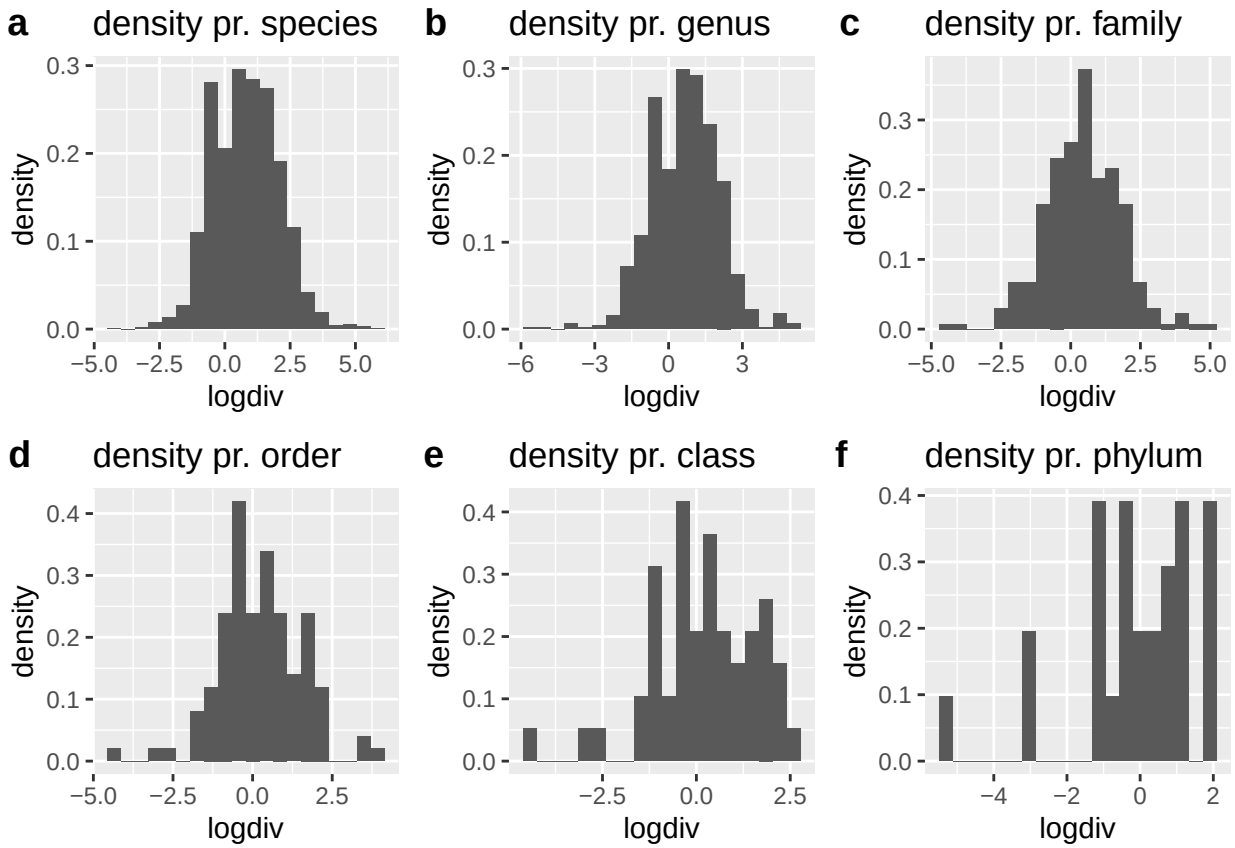


Here we can see we have alot of divs at $= 0$, Looking at not having them we get this

```
# Plloting the different taxs
to_plot <- "div"
p0 <- functaxboxplot(D, (species), .data[[to_plot]],"species",log)
p1 <- functaxboxplot(D, (genus), .data[[to_plot]],"genus",log)
p2 <- functaxboxplot(D, (family),.data[[to_plot]], "family",log)
p3 <- functaxboxplot(D, (order), .data[[to_plot]],"order",log)
p4 <- functaxboxplot(D, (class),.data[[to_plot]], "class",log)
p5 <- functaxboxplot(D, (phylum),.data[[to_plot]], "phylum",log)

plot_grid(p0,p1, p2, p3, p4,p5,labels ="auto")
```

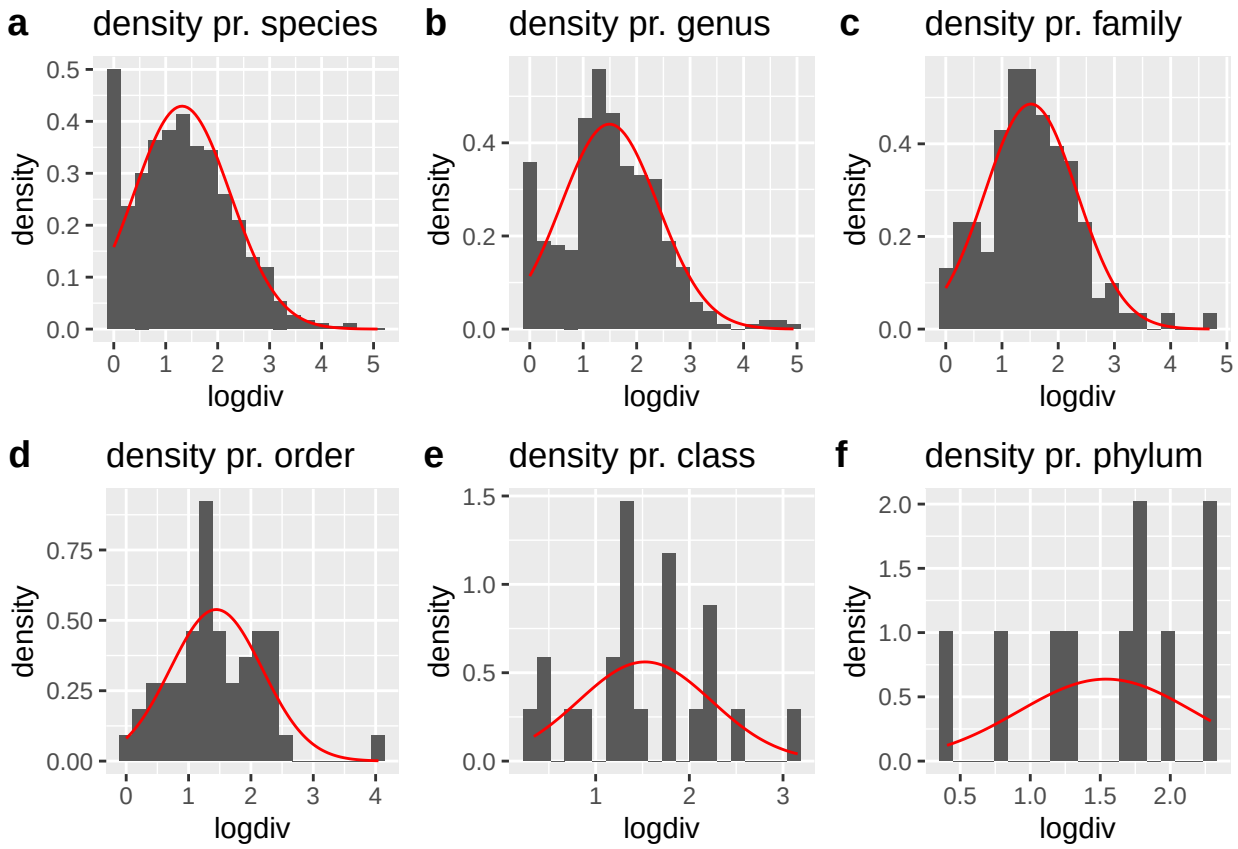
```
## Warning: Removed 1524 rows containing non-finite values (‘stat_bin()‘).

## Warning: Removed 101 rows containing missing values (‘geom_function()‘).
```

```
## Warning: Removed 513 rows containing non-finite values (‘stat_bin()‘).

## Warning: Removed 101 rows containing missing values (‘geom_function()‘).

## Warning: Removed 119 rows containing non-finite values (‘stat_bin()‘).

## Warning: Removed 101 rows containing missing values (‘geom_function()‘).

## Warning: Removed 47 rows containing non-finite values (‘stat_bin()‘).

## Warning: Removed 101 rows containing missing values (‘geom_function()‘).

## Warning: Removed 19 rows containing non-finite values (‘stat_bin()‘).

## Warning: Removed 101 rows containing missing values (‘geom_function()‘).

## Warning: Removed 7 rows containing non-finite values (‘stat_bin()‘).

## Warning: Removed 101 rows containing missing values (‘geom_function()‘).
```

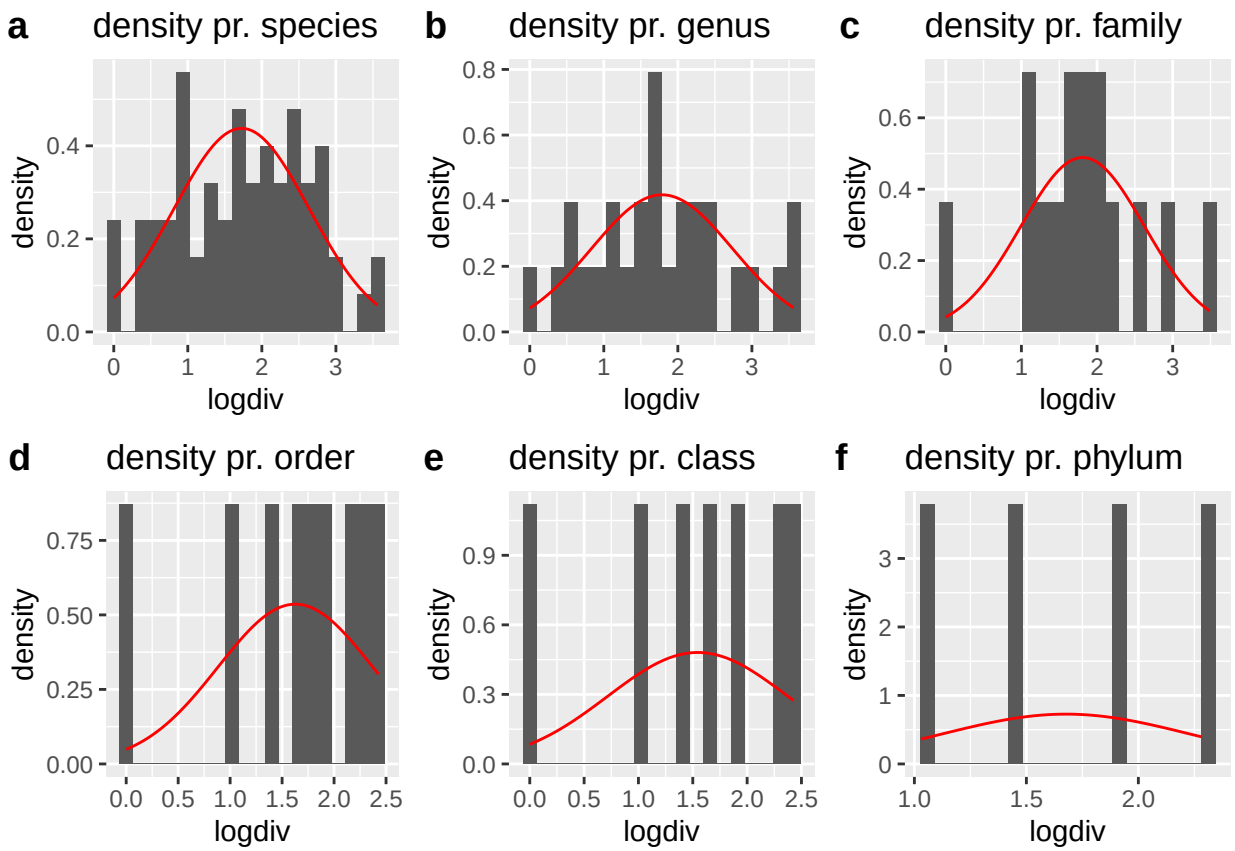


There is an argument for not working with the var of 0 or we can try and remove all entries with n16=1 (since they always has zero) It turns out n16=1 gives a ton with no var, n16 $>$ 4 is more acurate. Alternativly you could have it for one speciefic n16 or take n16 into the model when modelling div

```
# For n16 > 4
D_sub <- filter(D, n16 > 4)
to_plot <- "div"
p0 <- functaxboxplot(D_sub, (species), .data[[to_plot]],"species",log1p)
p1 <- functaxboxplot(D_sub, (genus), .data[[to_plot]],"genus",log1p)
p2 <- functaxboxplot(D_sub, (family),.data[[to_plot]], "family",log1p)
p3 <- functaxboxplot(D_sub, (order), .data[[to_plot]],"order",log1p)
p4 <- functaxboxplot(D_sub, (class),.data[[to_plot]], "class",log1p)
p5 <- functaxboxplot(D_sub, (phylum),.data[[to_plot]], "phylum",log1p)
plot_grid(p0,p1, p2, p3, p4,p5,labels ="auto")
```

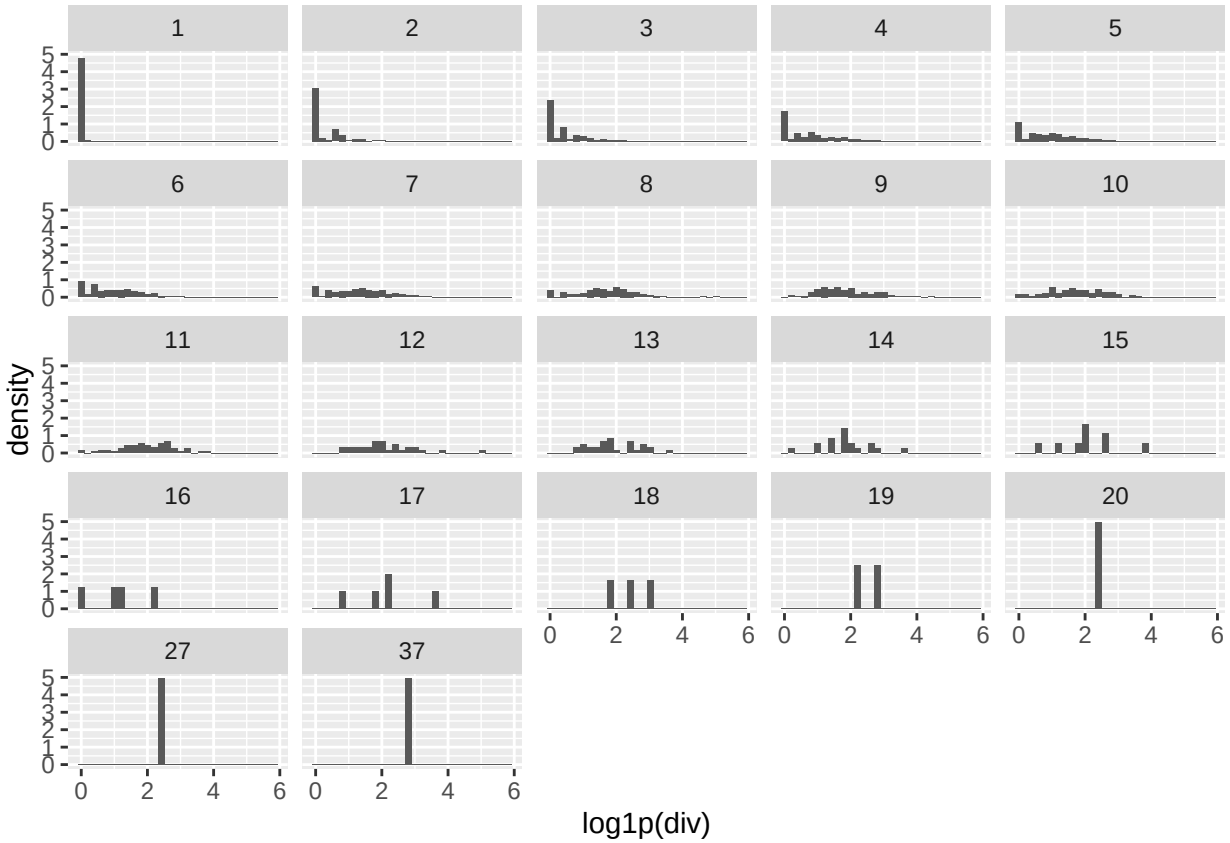



```
# For n16 = 4
D_sub <- filter(D, n16 == 10)
to_plot <- "div"
p0 <- functaxboxplot(D_sub, (species), .data[[to_plot]],"species",log1p)
p1 <- functaxboxplot(D_sub, (genus), .data[[to_plot]],"genus",log1p)
p2 <- functaxboxplot(D_sub, (family),.data[[to_plot]], "family",log1p)
p3 <- functaxboxplot(D_sub, (order), .data[[to_plot]],"order",log1p)
p4 <- functaxboxplot(D_sub, (class),.data[[to_plot]], "class",log1p)
p5 <- functaxboxplot(D_sub, (phylum),.data[[to_plot]], "phylum",log1p)

plot_grid(p0,p1, p2, p3, p4,p5,labels ="auto")
```




```
# Lets try for all different n16

D_sub <- mutate(D, n16 = factor(as.integer(n16)))
ggplot(D_sub) +
    geom_histogram(aes(x=log1p(div), y=after_stat(density))) +
  facet_wrap(~n16)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



1 n16 always has zero, ofc. It seems the more n16 we get the less with zero div. Which again makes sense

Furthermore i believe im going to continue working from a speceis level as the distributions look ok at that level and it is not worth sacreficing resolution to get better model assumptions.. for now

Div v n16

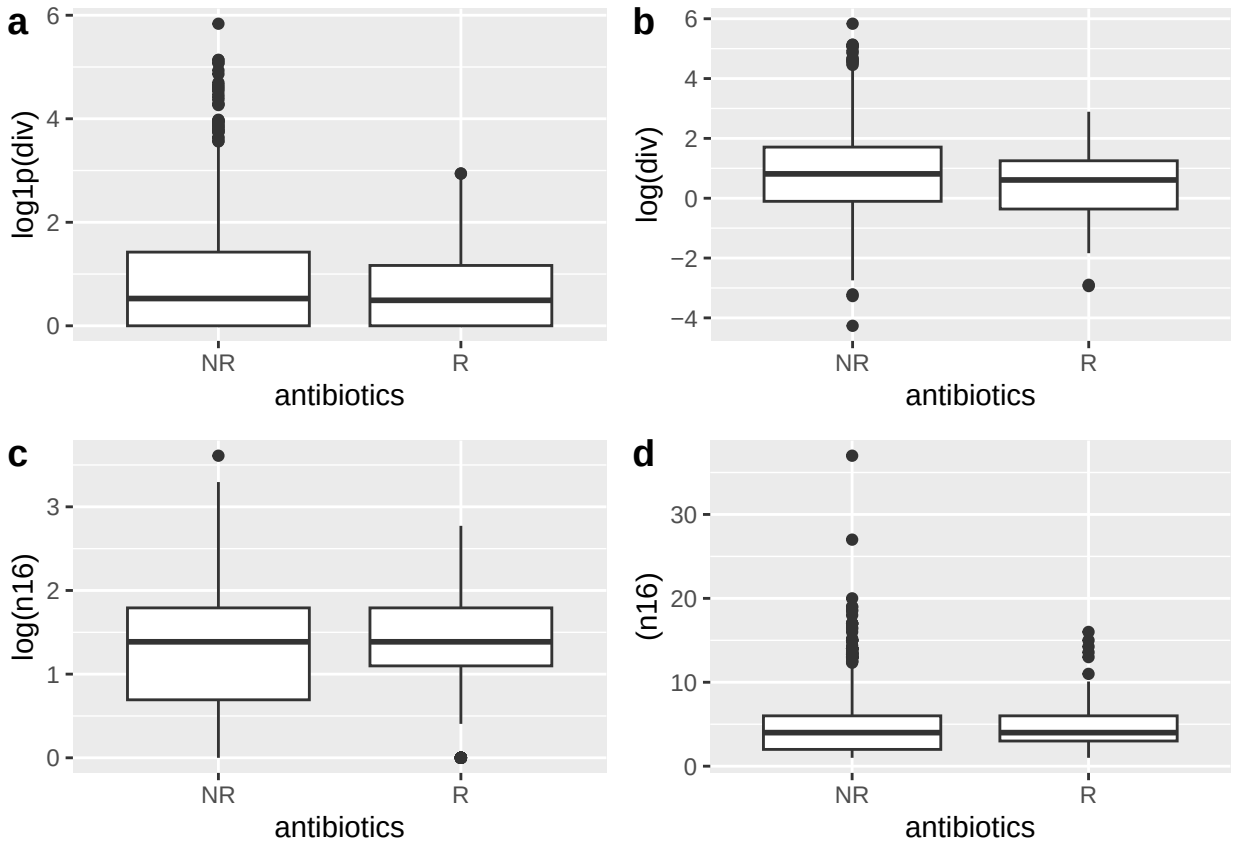
We expect there to be a relationshio.. lets see

Antibiotics

Lets start by checking in general The hypthesis was that they might have more div.

```
p1 <- ggplot(D)+
  geom_boxplot(aes(x=antibiotics, y=log1p(div)))
p2 <- ggplot(D)+
  geom_boxplot(aes(x=antibiotics, y=log(div)))
p3 <- ggplot(D)+
  geom_boxplot(aes(x=antibiotics, y=log(n16)))
p4 <- ggplot(D)+
  geom_boxplot(aes(x=antibiotics, y=(n16)))
plot_grid(p1,p2,p3,p4,labels="auto")
```

```
## Warning: Removed 1524 rows containing non-finite values (`stat_boxplot()`).
```



Formatting daTA Lets check for more specifc types of AR First getting the subset of the data with AR resistence info about the Antibiotics which target the 16s rRNA

```
# Getting the ones which are actually targeting 16S
# Reading them from ARtarget16s.csv
target16S <- read_csv2("../data/ARtarget16s.csv",show_col_types = FALSE,col_names = FALSE)
```

```
## i Using "','" as decimal and "'.'" as grouping mark. Use `read_delim()` for more control.
```

```
targetvector <- as.array(target16S$X1)
found_16S <- as.array(colnames(select(D_tmp,lincomycin:spiramycin.II)))
intersect <- intersect(targetvector,found_16S)
D_ar <- select(D_tmp, intersect, n16, div)
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use `all_of()` or `any_of()` instead.
##   # Was:
##   data %>% select(intersect)
##
##   # Now:
##   data %>% select(all_of(intersect))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
```

Different types

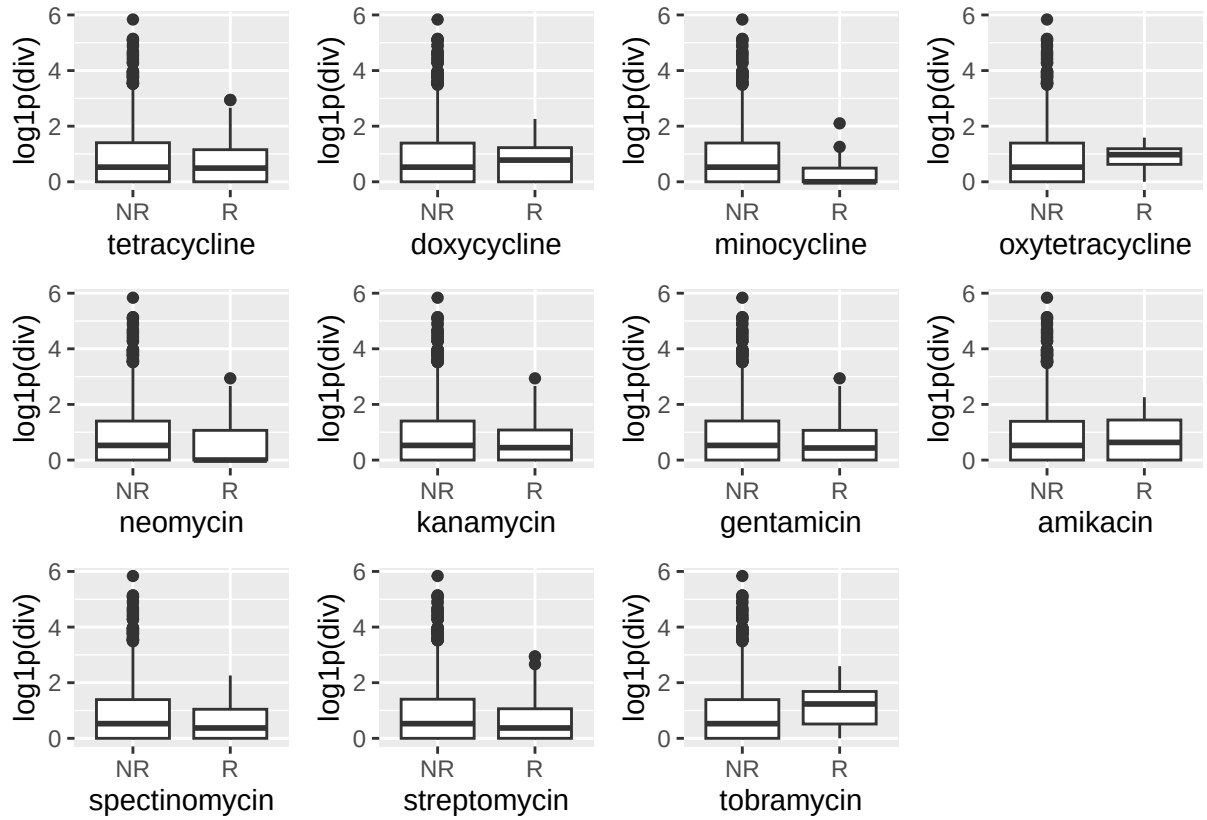
Div Now lets look at some plots firstly for div

```
library(gridExtra)
```

```
##
## Vedhæfter pakke: 'gridExtra'
```

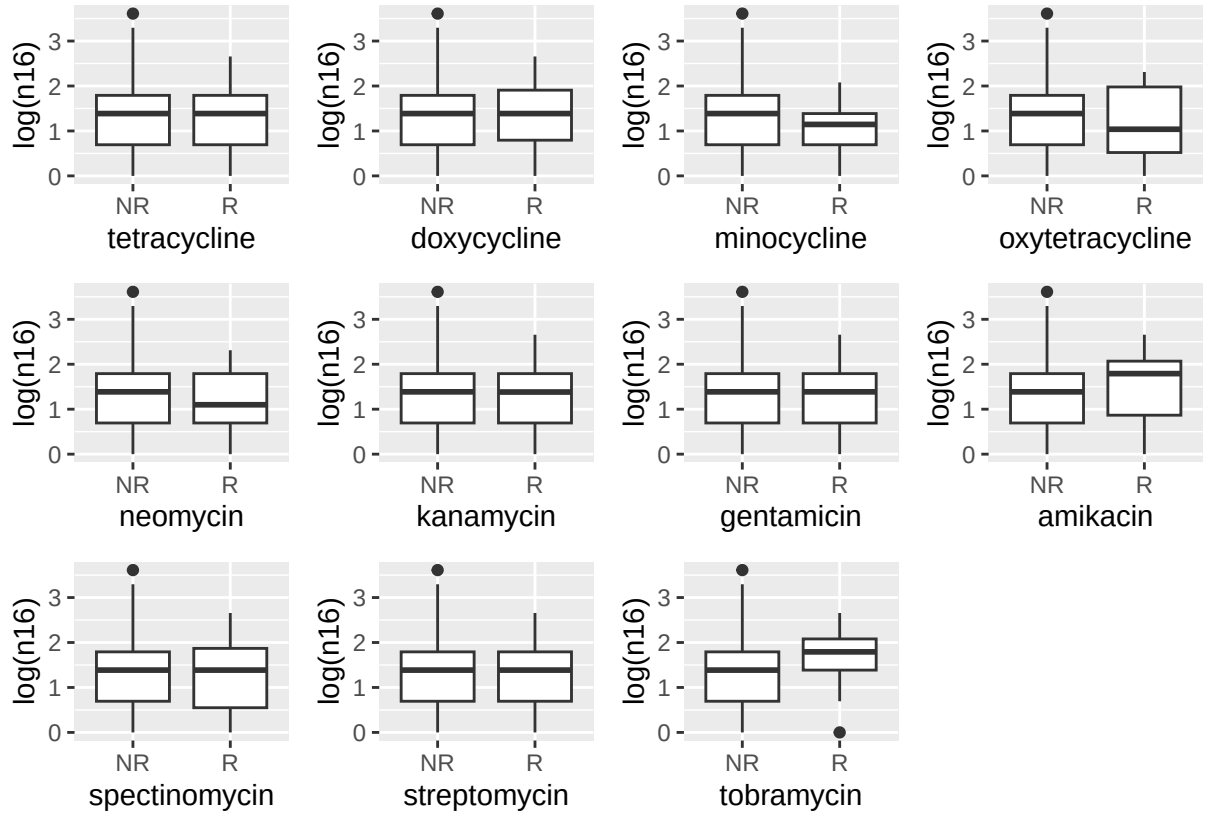
```
## Det følgende objekt er maskeret fra 'package:dplyr':
##
##     combine
```

```
plotlist = list()
for(i in seq_along(intersect)){
  antibiotic = intersect[i]
  p <- ggplot(D_ar)+
    geom_boxplot(aes(x=.data[[antibiotic]], y=log1p(div)))
  plotlist = c(plotlist, list(p))
}

wrap_plots(plotlist)
```



n16 and now for n16

```
plotlist = list()
for(i in seq_along(intersect)){
  antibiotic = intersect[i]
  p <- ggplot(D_ar)+
    geom_boxplot(aes(x=.data[[antibiotic]], y=log(n16)))
  plotlist = c(plotlist, list(p))
}

wrap_plots(plotlist)
```



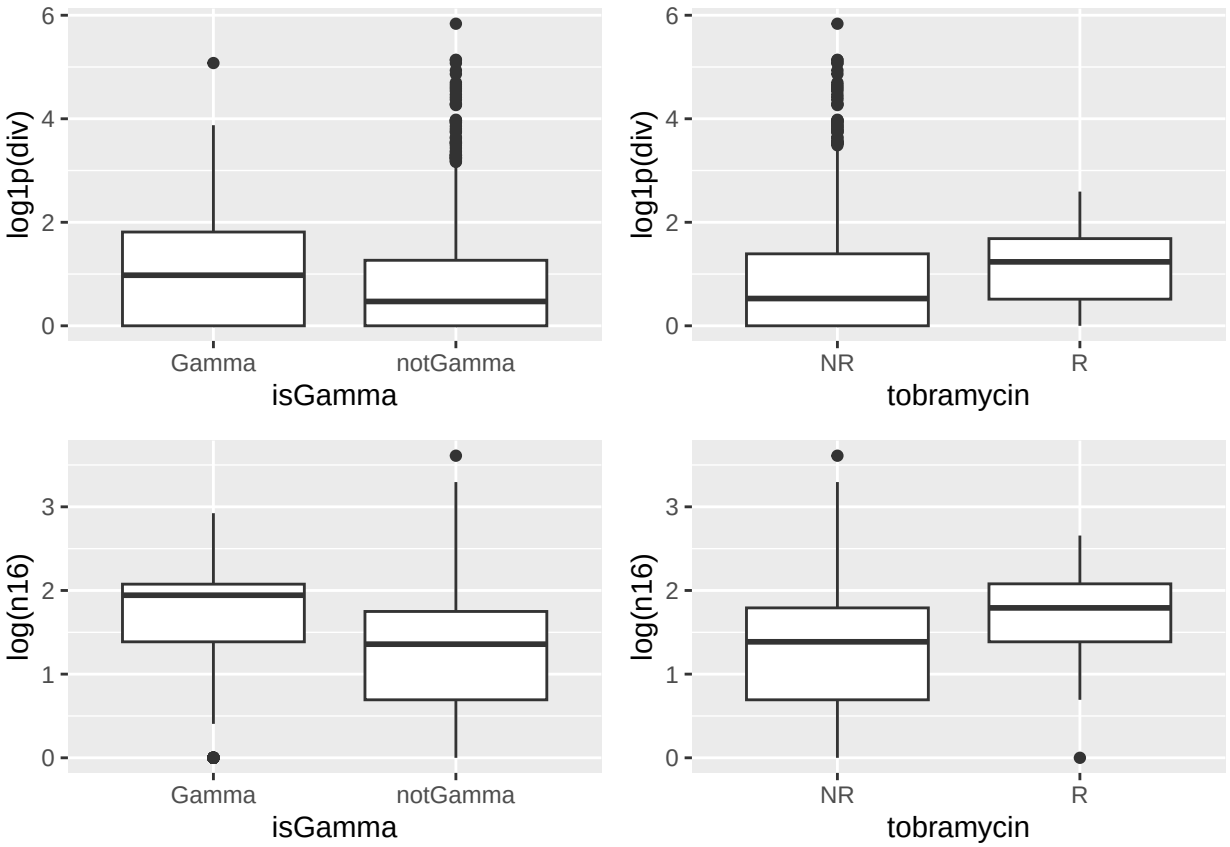
Deep dive into topramycin Some seem to have a change but it is my suspesion that its because of the randomness with a small subset randomly having more or just because of other factors. Eg the small subset which are Antibiotic resistenet to a specific AB might just share taxonmic. Lets check for tobramycin

```
# Get the taxonomy info for all the different bacteria resistent to tobramycin
D_ar_tax <- select(D_tmp, intersect, n16, div,genus,family,order, class,phylum)
filter(D_ar_tax, tobramycin == "R") %>%
  select(genus,family,order,class,phylum)
```

```
## # A tibble: 17 x 5
##    genus          family              order            class           phylum
##    <fct>          <fct>               <fct>            <fct>           <fct>
##  1 Aeromonas      Aeromonadaceae      Aeromonadales    Gammaproteobac~ Pseud~
##  2 Aeromonas      Aeromonadaceae      Aeromonadales    Gammaproteobac~ Pseud~
##  3 Agarilytica    Cellvibrionaceae    Cellvibrionales  Gammaproteobac~ Pseud~
##  4 Alteromonas    Alteromonadaceae    Alteromonadales  Gammaproteobac~ Pseud~
```

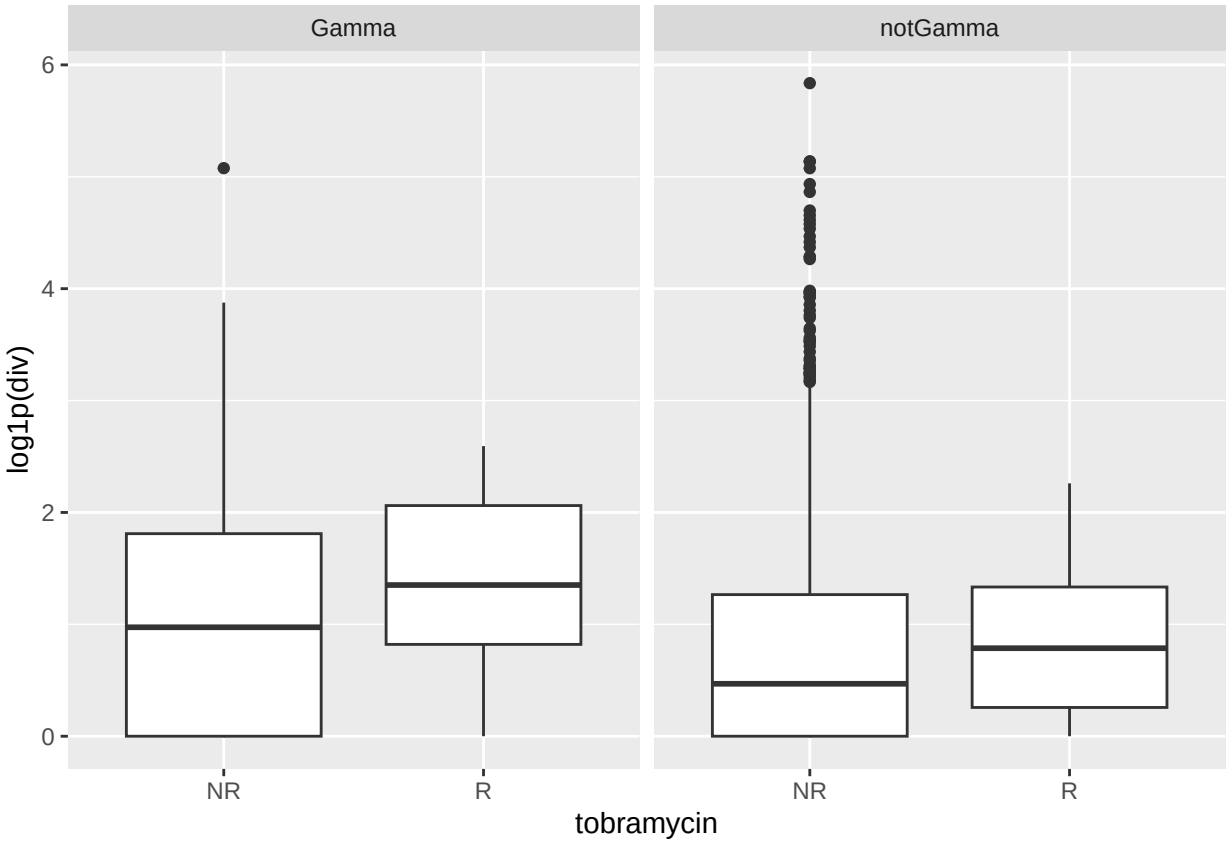
```
##  5 Brucella       Brucellaceae           Hyphomicrobiales Alphaproteobac~ Pseud~
##  6 Catenovulum    Alteromonadaceae       Alteromonadales  Gammaproteobac~ Pseud~
##  7 Dyella         Rhodanobacteraceae     Lysobacterales   Gammaproteobac~ Pseud~
##  8 Enterobacter   Enterobacteriaceae     Enterobacterales Gammaproteobac~ Pseud~
##  9 Enterobacter   Enterobacteriaceae     Enterobacterales Gammaproteobac~ Pseud~
## 10 Flavobacterium Flavobacteriaceae      Flavobacteriales Flavobacteriia  Bacte~
## 11 Indioceanicola Rhodospirillaceae      Rhodospirillales Alphaproteobac~ Pseud~
## 12 Marinobacter   Alteromonadaceae       Alteromonadales  Gammaproteobac~ Pseud~
## 13 Mycobacterium  Mycobacteriaceae       Mycobacteriales  Actinomycetes   Actin~
## 14 Priestia       Bacillaceae            Caryophanales    Bacilli         Bacil~
## 15 Pseudomonas    Pseudomonadaceae       Pseudomonadales  Gammaproteobac~ Pseud~
## 16 Staphylospora  Thermoactinomycetaceae Caryophanales    Bacilli         Bacil~
## 17 Streptomyces   Streptomycetaceae      Streptomycetales Actinomycetes   Actin~
```

```
# It seems most of them share the Gammaproteobacteria class / are from the..


GorNot <- mutate(D_ar_tax, isGamma = ifelse(class == "Gammaproteobacteria","Gamma","notGamma")) %>%
  mutate(isGamma=factor(isGamma))

p1 <- ggplot(GorNot) +
geom_boxplot(aes(x=isGamma, y=log1p(div)))

p2 <- ggplot(D_ar)+
    geom_boxplot(aes(x=tobramycin, y=log1p(div)))


p3 <- ggplot(GorNot) +
geom_boxplot(aes(x=isGamma, y=log(n16)))

p4 <- ggplot(D_ar)+
    geom_boxplot(aes(x=tobramycin, y=log(n16)))

plot_grid(p1,p2,p3,p4)
```

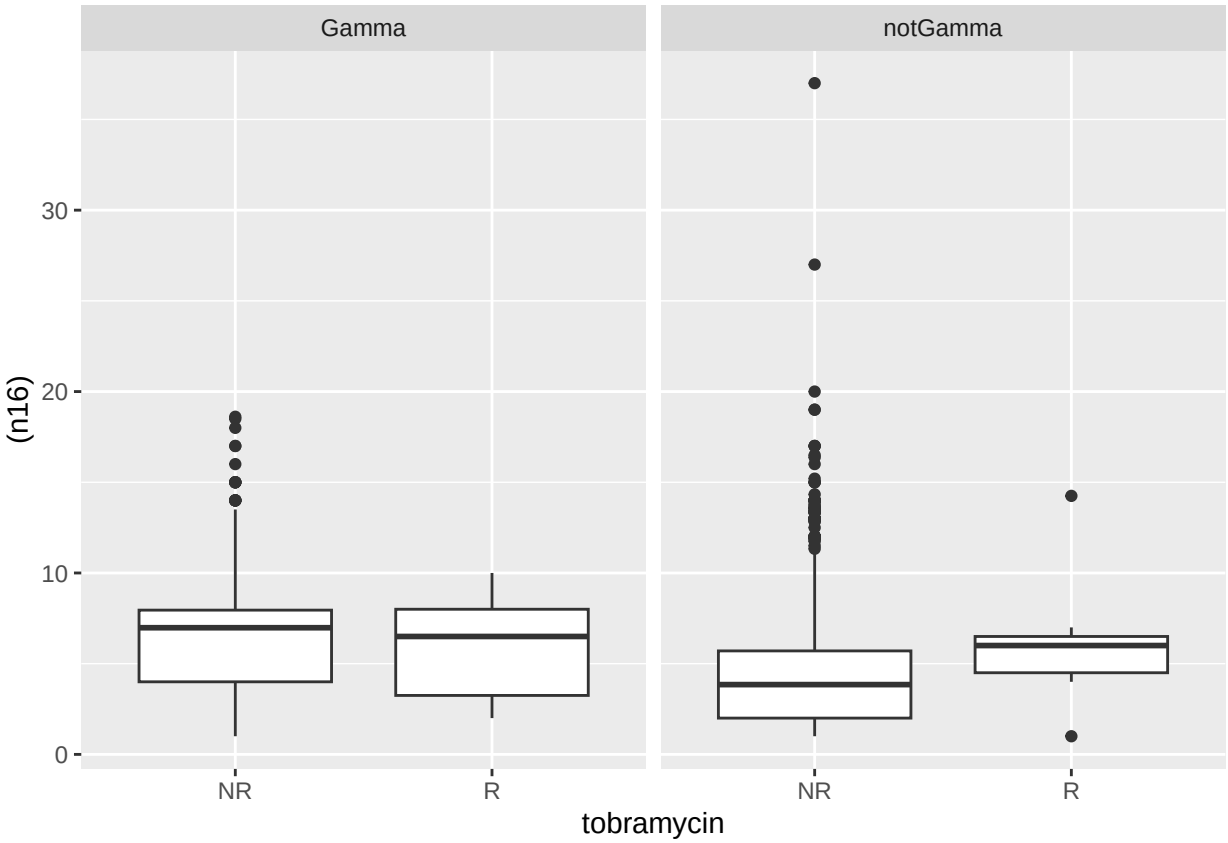


So we can see that it's proberly just tax- information messing it up Statistically we should test types of AR agains tax. Here we have 895 different in Gammaproteo group but only 8 of them with the AR

```
#Lastly lets try and plot the ones with AR against the ones which do not for gammaproteo
# Its mostly saying for the ones which are Gammaproteo

GorNot %>%
  ggplot()+
    geom_boxplot(aes(x=tobramycin, y=log1p(div))) +
    facet_wrap(~isGamma)
```



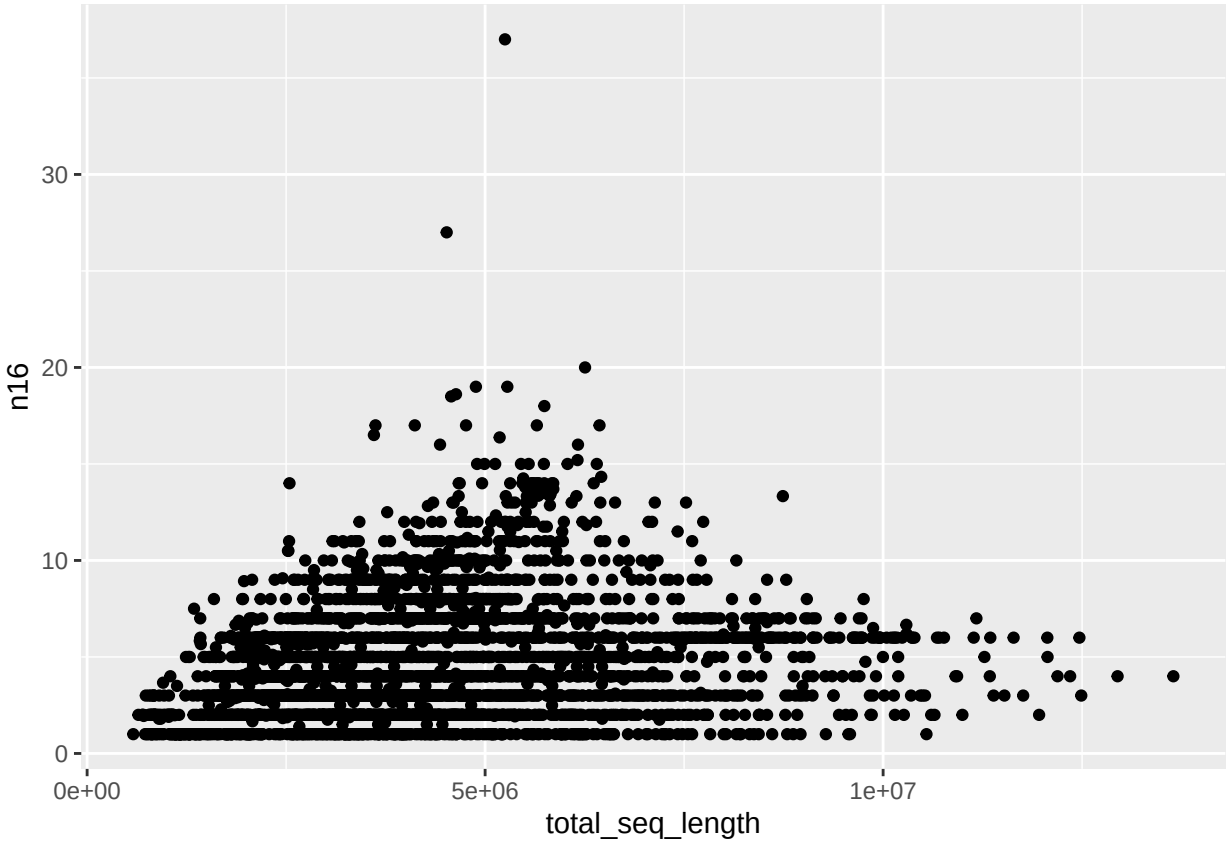

```
# There actually seem to be more
# Lest see for n16
# We expect n16 to go up for R again since the are correlated
GorNot %>%
  ggplot()+
    geom_boxplot(aes(x=tobramycin, y=(n16))) +
    facet_wrap(~isGamma)
```




```
# It seems to actually go down !
# so there is something going on :I
```

Genome size v total seq length

```
ggplot(D, aes(x=total_seq_length, y=n16, label=species)) +
  geom_point()
```


n16
30
20
10
0
0e+00
5e+06
1e+07
total_seq_length