Introduction to R for microbial ecologists

Daniel Vaulot

14 jan 2018

Contents

Aim

This document introduces basic R functions that can be used by microbial ecologists.

Prerequisites to be installed

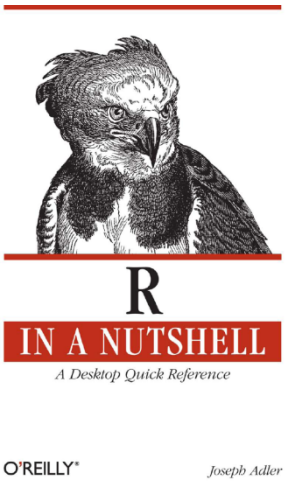
- R : https://pbil.univ-lyon1.fr/CRAN/
- R studio : https://www.rstudio.com/products/rstudio/download/#download
- Download and install the following libraries by running under R studio the following lines

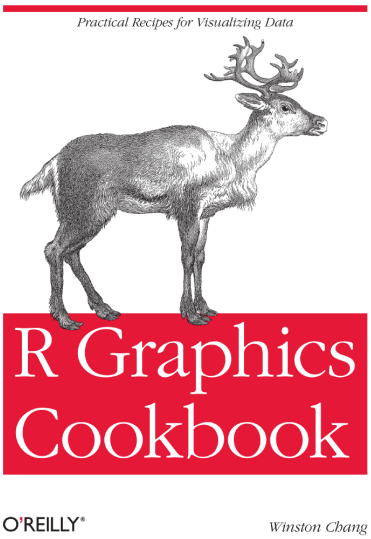
```
install.packages("dplyr")     # To manipulate dataframes
install.packages("tidyr")     # To manipulate dataframes
install.packages("readxl")    # To read Excel files into R

install.packages("ggplot2")   # for high quality graphics
```

```r
install.packages("maps")       # to make maps

install.packages("treemap")    # for treemaps

install.packages("FactoMineR") # multivariate analysis


source("https://bioconductor.org/biocLite.R")
biocLite('phyloseq')           # metabarcode data analysis
biocLite("Biostrings")         # manipulate sequences
```

Ressources

Books

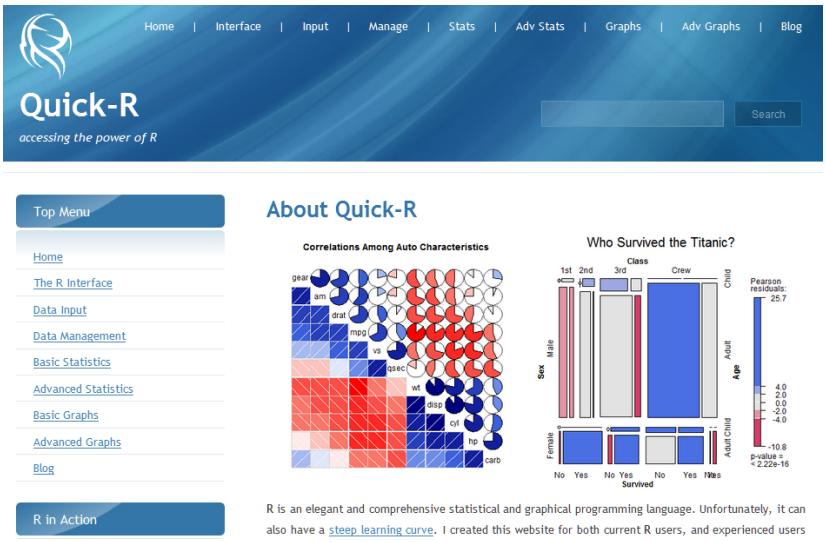




- R-intro.pdf : Very good introduction to R, short and clear
- R_in_a_nutshell.pdf : Many many receipes to solve all your questions
- R graphics cook book : very good for ggplot2

Web

- Quick-R, very simple : http://www.statmethods.net/
- Maps : http://www.molecularecologist.com/2012/09/making-maps-with-r/

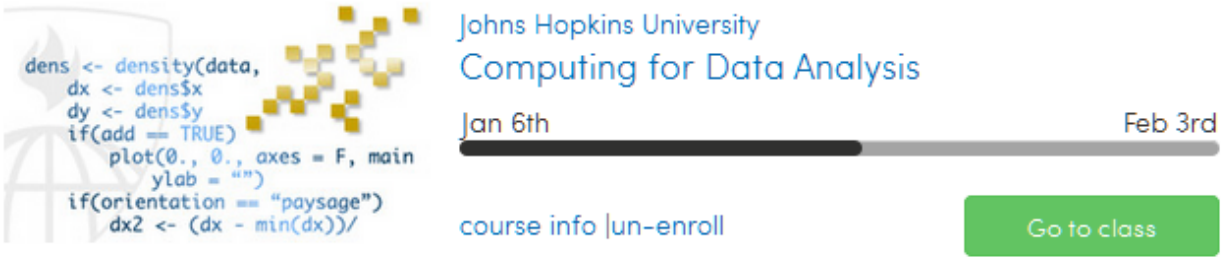


Cheat sheets

- R basics : http://github.com/rstudio/cheatsheets/raw/master/base-r.pdf
- ggplot2 : https://github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.pdf
- dplyr : https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf

On line course

- Coursera : https://www.coursera.org/



Step by step tutorial

Some important points before starting

- R is an interpreted language
- R is **case sensitive**
- R works with vectors
- Types of variables : character, real, logical, factor
- Special values : TRUE, FALSE, NA
- Types of structures : vector, matrix, list, data frame
- Directory names use the linux convention: use / and not

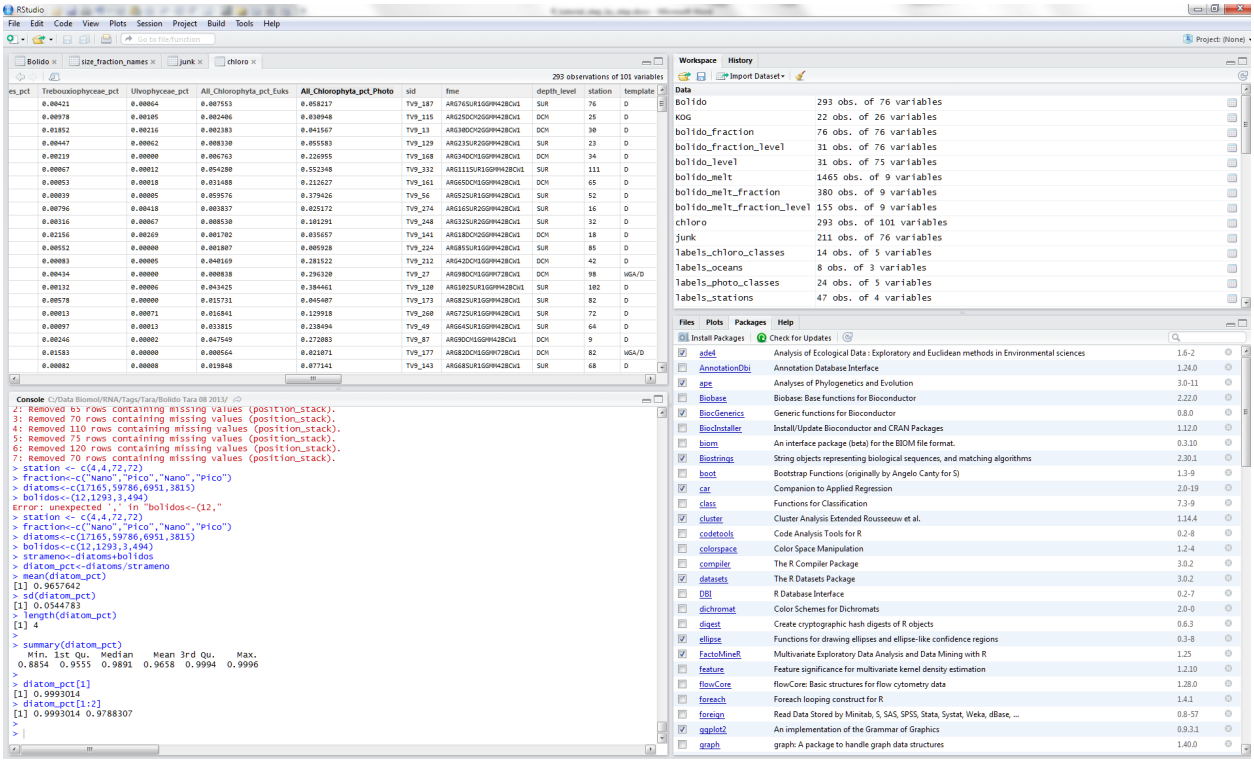
Start R Studio

Go to the directory where you put the tutorial.

Launch R Studio

Four windows

- top-left : script files / data tables
- bottom -left: code
- top - left : objects
- bottom - right : help / libraries / files / graphics



Load necessary libraries

```
library("dplyr")          # Needed to filter tables
library("tidyr")          # Needed to reshape tables from wide to long format
library("readxl")         # To read data easily
```

1 - Create simple vectors and data frame

Enter the data

Our aim here to create a small table and then to compute some simple statistics

station	fraction	diatoms	bolidos
4	Nano	17165	2
4	Pico	59786	1293
72	Nano	6951	3
72	Pico	3815	494

```
# We enter each column as a vector
station <- c("4","4","72","72")
fraction<-c("Nano","Pico","Nano","Pico")
diatoms<-c(17165,59786,6951,3815)
bolidos<-c(2,1293,3,494)
```

Compute new quantities

```
# Add 2 columns
strameno<-diatoms+bolidos
strameno
```

```
## [1] 17167 61079  6954  4309
```

```
# Divide one column by the other
diatoms_pct<-diatoms/strameno
diatoms_pct
```

```
## [1] 0.9998835 0.9788307 0.9995686 0.8853562
```

Compute statistics

```
# mean
mean(diatoms_pct)
```

```
## [1] 0.9659098
```

```
# standard deviation
sd(diatoms_pct)
```

```
## [1] 0.05459839
```

```
# number of observations
length(diatoms_pct)
```

```
## [1] 4
```

```
# quick summary
summary(diatoms_pct)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.8854  0.9555  0.9892  0.9659  0.9996  0.9999
```

Accessing subsets

```
diatoms_pct[1]
```

```
## [1] 0.9998835
```

```
diatoms_pct[1:2]
```

```
## [1] 0.9998835 0.9788307
```

Data frames

```
tara<-data.frame(station, fraction, diatoms, bolidos, diatoms_pct)
tara
```

```
##   station fraction diatoms bolidos diatoms_pct
## 1       4     Nano   17165       2   0.9998835
## 2       4     Pico   59786    1293   0.9788307
## 3      72     Nano    6951       3   0.9995686
## 4      72     Pico    3815     494   0.8853562
```

Access individual columns

```
tara$diatoms
```

```
## [1] 17165 59786  6951  3815
```

Access specific lines

```
tara$diatoms[tara$station==4]
```

```
## [1] 17165 59786
```

Compute statistics of a specific group

```
mean(tara$diatoms[tara$station==4])
```

```
## [1] 38475.5
```

Computing statistics according to a factor

This can be done at least two different ways, but you will see later that it is much easier to do with the dplyr package

```
# Using the tapply function
tapply(tara$diatoms, tara$station, mean)
```

```
##       4      72
## 38475.5  5383.0
```

```
# Using the aggregate functions
aggregate(data=tara, diatoms~station, FUN="mean")
```

```
##   station diatoms
## 1       4 38475.5
## 2      72  5383.0
```

2 - Importing data

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Sample	Bacillariophyta	Bolidophyceae	Chrysophyceae	Dictyochophyceae	Pelagophyceae	Phaeophyceae	Pinguiophyceae	Raphidophyceae	Strameno_all	Photo_all	depth_lev	station	template	fraction
2	TV9_237	17165	12	26	155	233	0	11	0	17602	22708	DCM	4	WGA/D	5-20
3	TV9_234	6159	42	223	487	138	12	2	0	7063	8817	SUR	4	D	5-20
4	TV9_254	59786	1293	8758	21967	73474	1835	19	0	167132	427846	DCM	4	D	0.8-5
5	TV9_235	4689	1036	7494	21293	4774	526	40	0	39852	93006	SUR	4	D	0.8-5
6	TV9_236	6280	2	21	14	13	0	6	0	6336	8976	DCM	4	WGA/D	180-2000
7	TV9_233	1000	188	670	1026	722	11	5	0	3622	5392	SUR	4	D	180-2000
8	TV9_20	12517	24	296	265	40	12	50	18	13222	14299	DCM	7	WGA/D	5-20
9	TV9_16	64721	163	593	1658	31	25	229	21	67441	70406	SUR	7	WGA/D	5-20
10	TV9_21	8126	2991	10069	19440	1687	382	20	48	42763	81891	DCM	7	D	0.8-5
11	TV9_17	13584	2261	25834	48876	871	2738	32	23	94219	144725	SUR	7	D	0.8-5
12	TV9_19	661	0	14	13	41	1	4	0	734	892	DCM	7	D	180-2000
13	TV9_15	227	0	5	2	4	0	5	0	243	342	SUR	7	D	180-2000
14	TV9_22	10354	58	226	510	54	5	40	1	11248	12400	DCM	7	D	20-180
15	TV9_18	10192	1	51	33	19	3	117	0	10416	11464	SUR	7	D	20-180
16	TV9_265	46	0	4	11	7	0	3	0	71	85	DCM	9	WGA/D	5-20
17	TV9_266	53108	866	4821	5586	2591	142	2429	0	69543	104023	SUR	9	WGA/D	5-20
18	TV9_87	17753	265	3870	15548	37127	1478	1	16	76058	180809	DCM	9	D	0.8-5
19	TV9_85	7466	2242	18754	39977	970	4516	90	56	74071	159650	SUR	9	D	0.8-5
20	TV9_86	32	0	2	4	16	0	0	0	54	383	DCM	9	D	180-2000
21	TV9_84	2262	65	816	2460	3914	276	4	1	9798	23658	SUR	9	D	180-2000
22	TV9_268	617858	0	1147	0	1	0	521	0	619527	625534	SUR	11	WGA/D	5-20
23	TV9_267	23786	490	5509	8066	1785	898	151	0	40685	67697	SUR	11	D	0.8-5
24	TV9_270	655	11	404	920	865	118	1	1	2975	6012	SUR	11	D	180-2000
25	TV9_269	560	13	106	154	37	16	5	0	891	1477	SUR	11	D	20-180

A few important points :

- Your data must be formatted in a clean table form
  - No blank line
  - Each column must contain data of the same type (e.g. dates)
  - Missing data can be represented by empty cells
  - Each line must contain data in ALL columns
- Column titles (the first line)
  - No space (use _)
  - Always begin by letter (not a number)
- Only import primary data, all derived data can (and must) be computed with R which makes data changes much more easy

The hard way - exporting from Excel to a tab-delimited file

- Open Excel file in /data directory : R_Tara.xlsx
- Copy and Paste into text file using Notepad++
- Save as R_Tara.txt

Note : you can also export from Excel but then it must be TAB-delimited (tsv file)

```
tara <- read.delim("data/R_Tara.txt")
```

Get the name and type of all the columns - Note that strings are of type “factor” Note that empty cells are labelled as **NA** (not available) which is a R constant

```
str(tara)
```

```
## 'data.frame':    293 obs. of  28 variables:
##  $ Sample          : Factor w/ 293 levels "TV9_1","TV9_10",..: 124 121 141 122 123 120 93 58 101 68
##  $ Bacillariophyta : int  17165 6159 59786 4689 6280 1000 12517 64721 8126 13584 ...
##  $ Bolidophyceae   : int  12 42 1293 1036 2 188 24 163 2991 2261 ...
##  $ Chrysophyceae   : int  26 223 8758 7494 21 670 296 593 10069 25834 ...
##  $ Dictyochophyceae: int  155 487 21967 21293 14 1026 265 1658 19440 48876 ...
##  $ Pelagophyceae   : int  233 138 73474 4774 13 722 40 31 1687 871 ...
##  $ Phaeophyceae    : int  0 12 1835 526 0 11 12 25 382 2738 ...
##  $ Pinguiophyceae  : int  11 2 19 40 6 5 50 229 20 32 ...
##  $ Raphidophyceae  : int  0 0 0 0 0 0 18 21 48 23 ...
##  $ Strameno_all    : int  17602 7063 167132 39852 6336 3622 13222 67441 42763 94219 ...
```

```
##  $ Photo_all       : int  22708 8817 427846 93006 8976 5392 14299 70406 81891 144725 ...
##  $ depth_level     : Factor w/ 2 levels "DCM","SUR": 1 2 1 2 1 2 1 2 1 2 ...
##  $ station         : int  4 4 4 4 4 4 7 7 7 7 ...
##  $ template        : Factor w/ 2 levels "D","WGA/D": 2 1 1 1 2 1 2 2 1 1 ...
##  $ fraction        : Factor w/ 4 levels "0.8-5","180-2000",..: 4 4 1 1 2 2 4 4 1 1 ...
##  $ ntags           : int  1796545 2128487 2122955 976685 1857697 3150580 2549282 1606212 1625284 133
##  $ Month           : Factor w/ 10 levels "apr","aug","dec",..: 10 10 10 10 10 10 10 10 10 10 ...
##  $ Latitude        : num  36.6 36.6 36.6 36.6 36.6 ...
##  $ Longitude       : num  -6.57 -6.57 -6.57 -6.57 -6.57 ...
##  $ sampling_depth  : num  40 3 40 3 40 3 42 3 42 3 ...
##  $ date            : Factor w/ 77 levels "01-Aug-2011 20:13:34",..: 44 44 44 44 44 44 65 64 65 64 ..
##  $ chloro_hplc     : num  NA 0.0984 NA 0.0984 NA ...
##  $ tara_NO2        : num  NA NA NA NA NA NA 0.005 0.076 0.005 0.076 ...
##  $ tara_PO4        : num  NA NA NA NA NA NA 0.026 0.041 0.026 0.041 ...
##  $ NO2NO3          : num  NA NA NA NA NA NA 0.4 0.23 0.4 0.23 ...
##  $ tara_SI         : num  NA NA NA NA NA NA 0.652 0.998 0.652 0.998 ...
##  $ tara_temp       : num  NA NA NA NA NA ...
##  $ tara_salinity   : num  NA NA NA NA NA ...
```

The easy way - Read directly Excel (readxl library)

```
tara <- read_excel("data/R_Tara.xlsx", sheet = "R Tara")
```

Get the name and type of all the columns - Note that strings are now of type "char", which is better

```
str(tara)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    293 obs. of  28 variables:
##  $ Sample          : chr  "TV9_237" "TV9_234" "TV9_254" "TV9_235" ...
##  $ Bacillariophyta : num  17165 6159 59786 4689 6280 ...
##  $ Bolidophyceae   : num  12 42 1293 1036 2 ...
##  $ Chrysophyceae   : num  26 223 8758 7494 21 ...
##  $ Dictyochophyceae: num  155 487 21967 21293 14 ...
##  $ Pelagophyceae   : num  233 138 73474 4774 13 ...
##  $ Phaeophyceae    : num  0 12 1835 526 0 ...
##  $ Pinguiophyceae  : num  11 2 19 40 6 5 50 229 20 32 ...
##  $ Raphidophyceae  : num  0 0 0 0 0 0 18 21 48 23 ...
##  $ Strameno_all    : num  17602 7063 167132 39852 6336 ...
##  $ Photo_all       : num  22708 8817 427846 93006 8976 ...
##  $ depth_level     : chr  "DCM" "SUR" "DCM" "SUR" ...
##  $ station         : num  4 4 4 4 4 4 7 7 7 7 ...
##  $ template        : chr  "WGA/D" "D" "D" "D" ...
##  $ fraction        : chr  "5-20" "5-20" "0.8-5" "0.8-5" ...
##  $ ntags           : num  1796545 2128487 2122955 976685 1857697 ...
##  $ Month           : chr  "sep" "sep" "sep" "sep" ...
##  $ Latitude        : num  36.6 36.6 36.6 36.6 36.6 ...
##  $ Longitude       : num  -6.57 -6.57 -6.57 -6.57 -6.57 ...
##  $ sampling_depth  : num  40 3 40 3 40 3 42 3 42 3 ...
##  $ date            : chr  "15-Sep-2009 16:45:02" "15-Sep-2009 16:45:02" "15-Sep-2009 16:45:02" "15-S
##  $ chloro_hplc     : num  NA 0.0984 NA 0.0984 NA ...
##  $ tara_NO2        : num  NA NA NA NA NA ...
##  $ tara_PO4        : num  NA NA NA NA NA ...
##  $ NO2NO3          : num  NA NA NA NA NA ...
##  $ tara_SI         : num  NA NA NA NA NA ...
##  $ tara_temp       : num  NA NA NA NA NA ...
```

```
##  $ tara_salinity   : num  NA NA NA NA NA ...
```

3 - Compute derived quantities and Statistics (using dplyr library)

Compute % of Bacilliarophyta and Pelagophyceae vs Total photosynthetic

```
tara <- tara %>% mutate(Baci_pct = Bacillariophyta/Photo_all*100,
                        Pela_pct = Pelagophyceae/Photo_all*100)
```

Mean and SD as a function of size fraction and depth_level

```
tara_stat <- tara %>% group_by(fraction, depth_level) %>%
                      summarise(Baci_pct_mean = mean(Baci_pct),
                                Baci_pct_SD = sd(Baci_pct),
                                n=n())
tara_stat
```

```
## # A tibble: 8 x 5
## # Groups: fraction [?]
##   fraction depth_level Baci_pct_mean Baci_pct_SD     n
##   <chr>    <chr>               <dbl>       <dbl> <int>
## 1 0.8-5    DCM                  14.5        15.6    33
## 2 0.8-5    SUR                  12.5        15.0    40
## 3 180-2000 DCM                  59.0        30.2    31
## 4 180-2000 SUR                  53.7        29.9    45
## 5 20-180   DCM                  84.7        20.6    28
## 6 20-180   SUR                  81.7        19.6    42
## 7 5-20     DCM                  73.8        26.5    33
## 8 5-20     SUR                  74.6        27.7    41
```

4 - Do simple X-Y plots (using ggplot2 library)

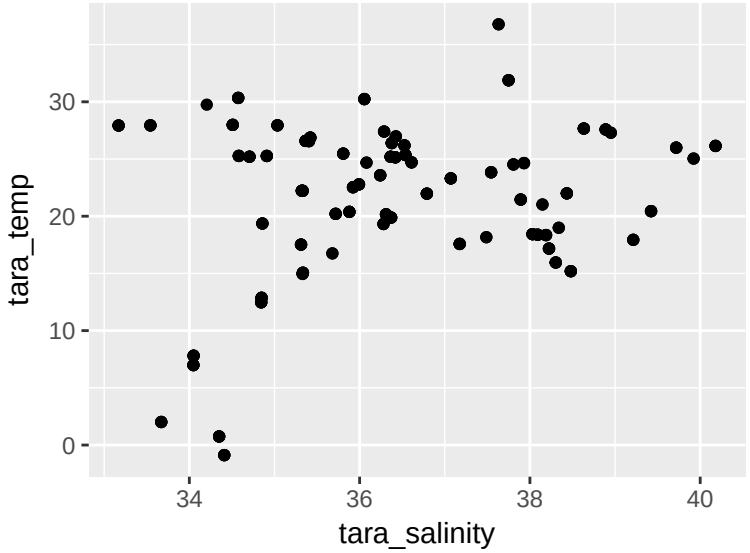
Load the ggplot2 library

```
library("ggplot2")        # To do graphics
```

X vs Y

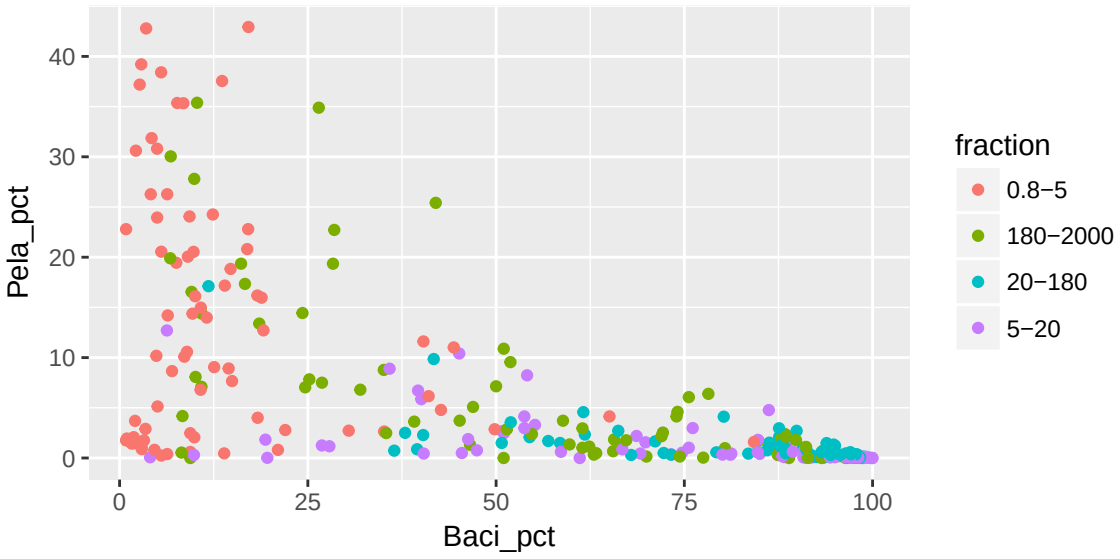
```
qplot(tara_salinity,tara_temp, data=tara)
```

```
## Warning: Removed 32 rows containing missing values (geom_point).
```



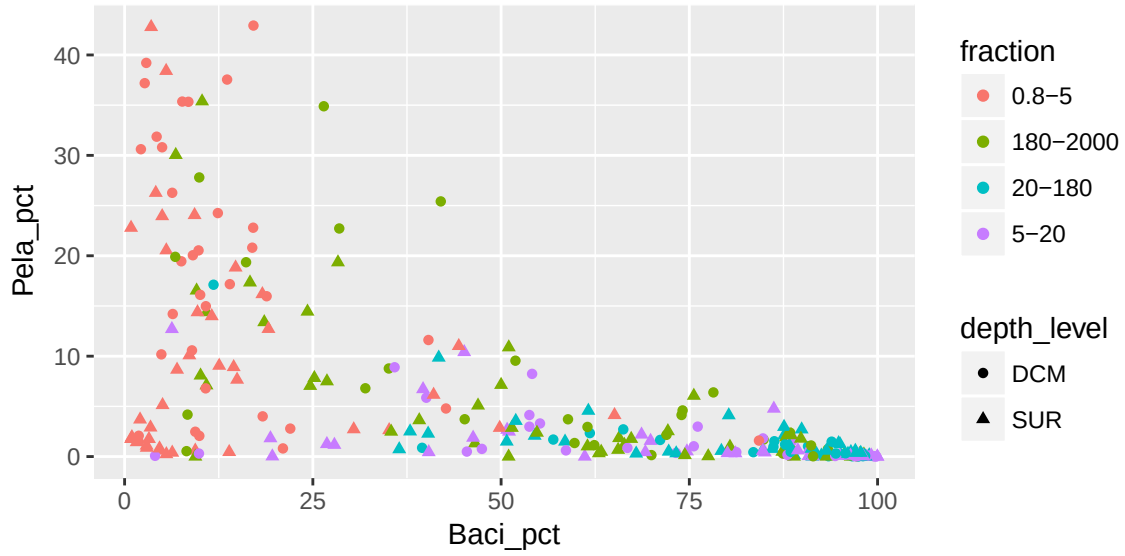
X vs Y with variation in color of points with size fraction

```
qplot(Baci_pct,Pela_pct, data=tara,color=fraction)
```



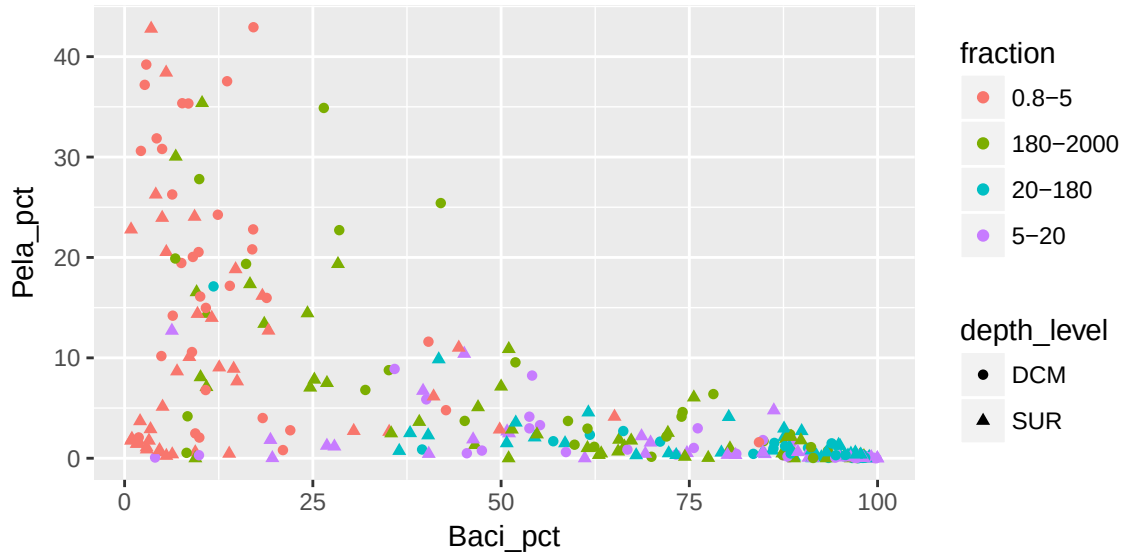
X vs Y with variation in color of points with size fraction and shape with depth level

```
qplot(Baci_pct,Pela_pct, data=tara, color=fraction, shape=depth_level)
```



X vs Y with variation in color of points with size fraction and shape with depth level

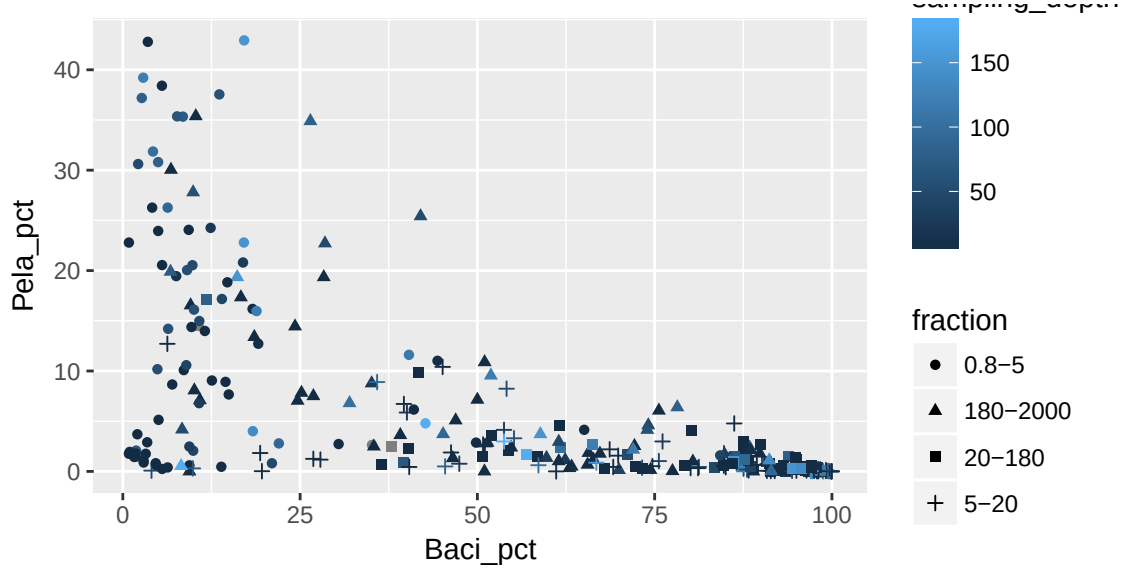
```
qplot(Baci_pct,Pela_pct, data=tara,color=fraction, shape=depth_level)
```



X vs Y with variation sampling_depth for color of points and shape with with size fraction.

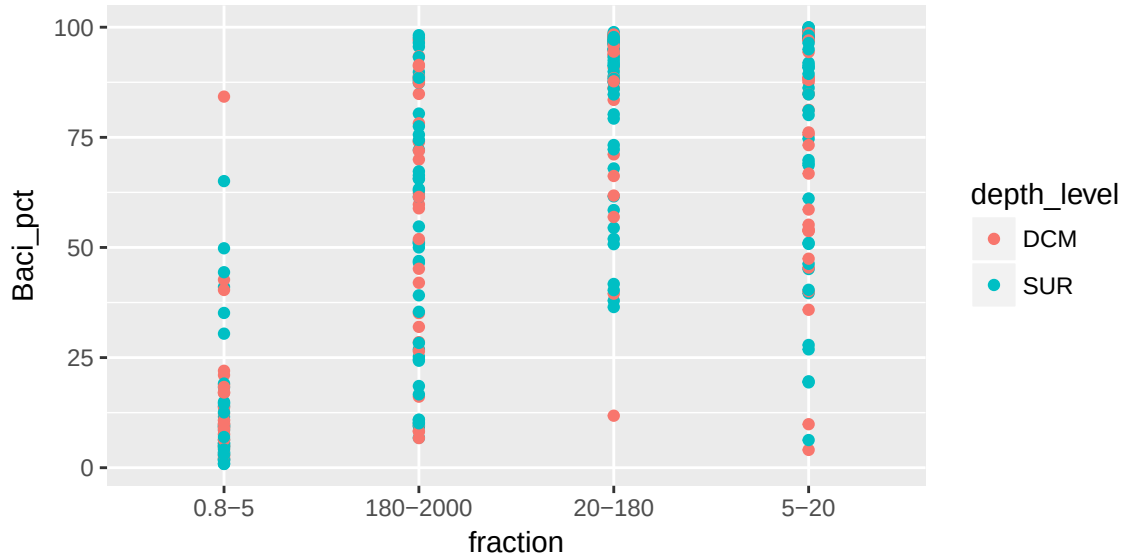
Note that sampling_depth is a **continuous variable**

```
qplot(Baci_pct,Pela_pct, data=tara, color = sampling_depth, shape = fraction)
```



Categorical data vs y with variation in color of points with depth level

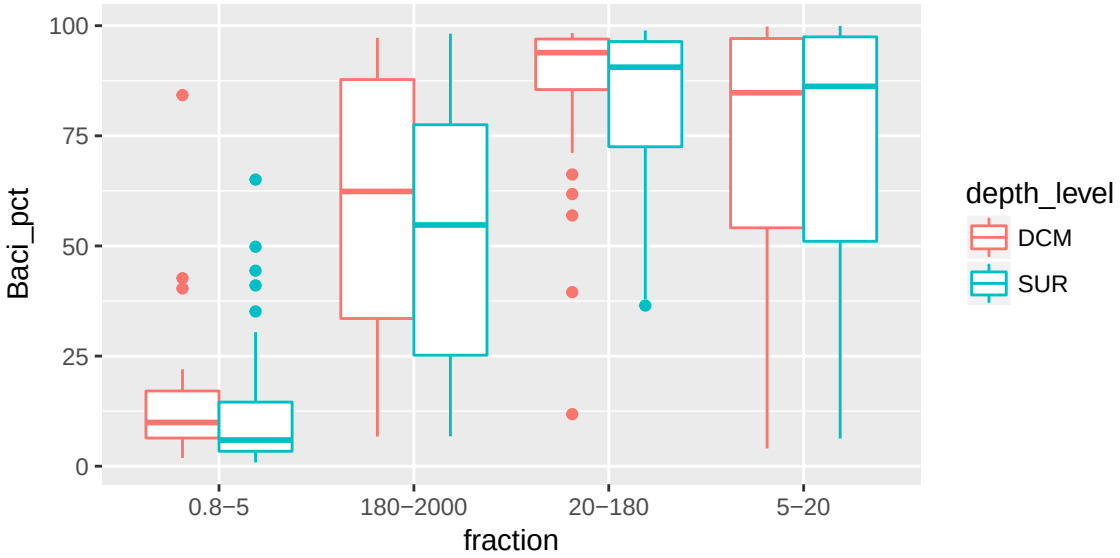
```
qplot(fraction,Baci_pct, data=tara, color=depth_level)
```



5 - Other types of plots

Boxplot for the same data

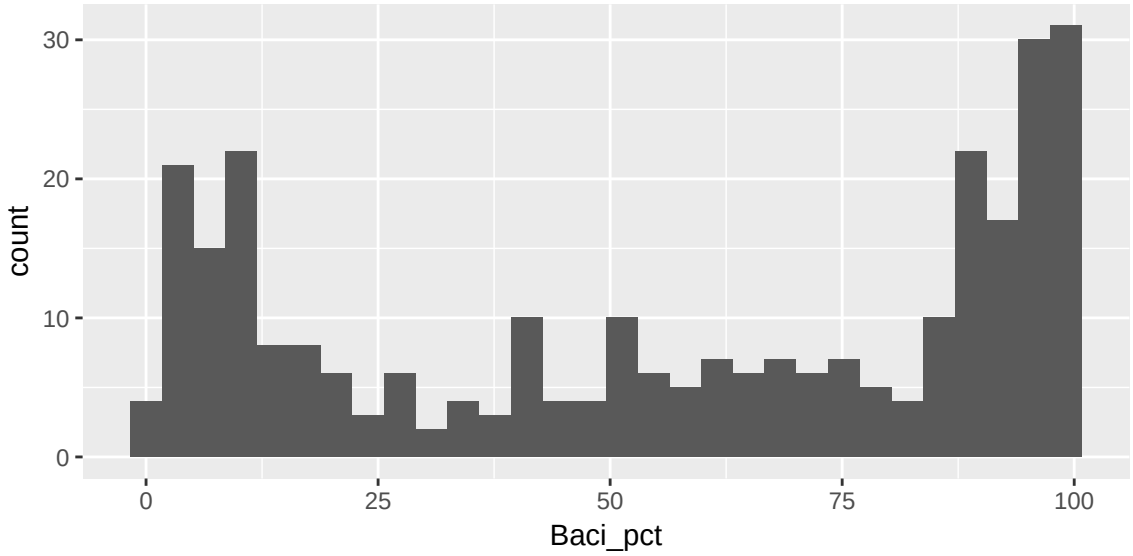
```
qplot(fraction,Baci_pct, data=tara, color=depth_level, geom="boxplot")
```



Histogram for all the data

```
qplot(Baci_pct, data=tara,  geom="histogram")
```

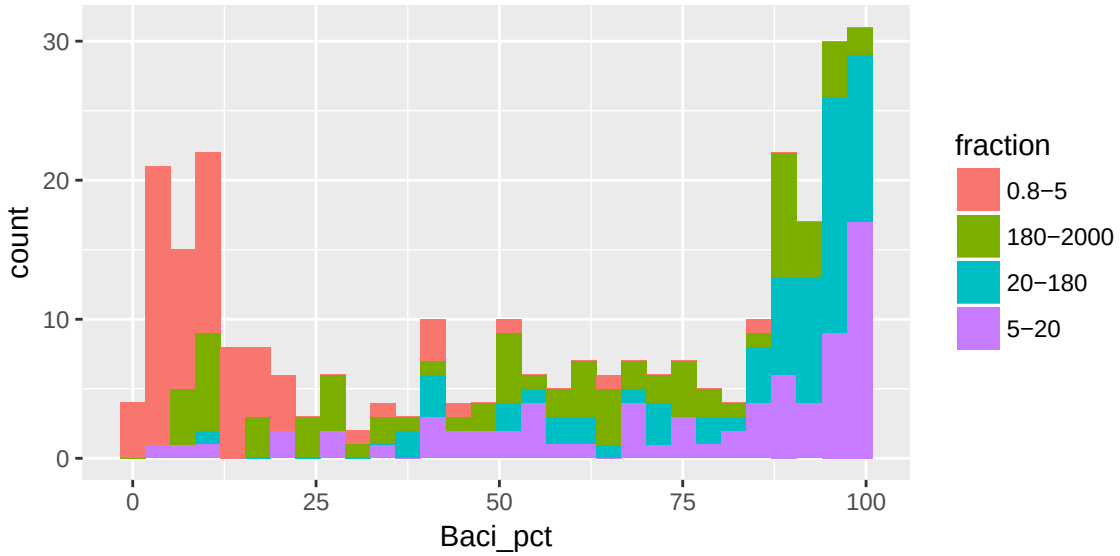
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Histogram with different color for each size fraction

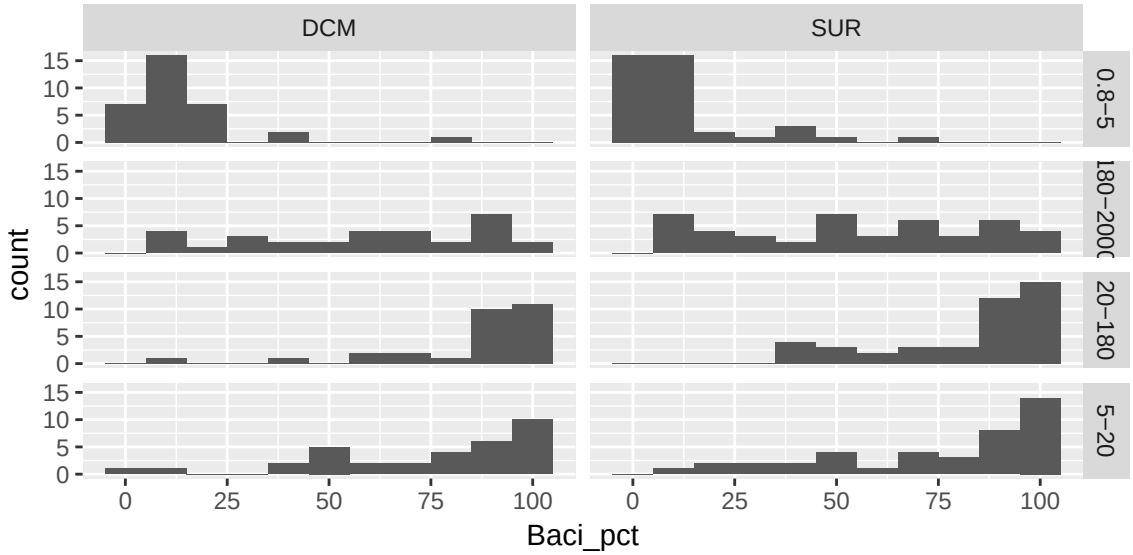
```
qplot(Baci_pct, data=tara,  fill=fraction, geom="histogram")
```

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



Histogram with different graphs (facets) for each size fraction and depth and change bin width

```
qplot(Baci_pct, data=tara,  facets=fraction~depth_level, geom="histogram",binwidth=10)
```



6 - Tree maps (much better than Pie charts...)

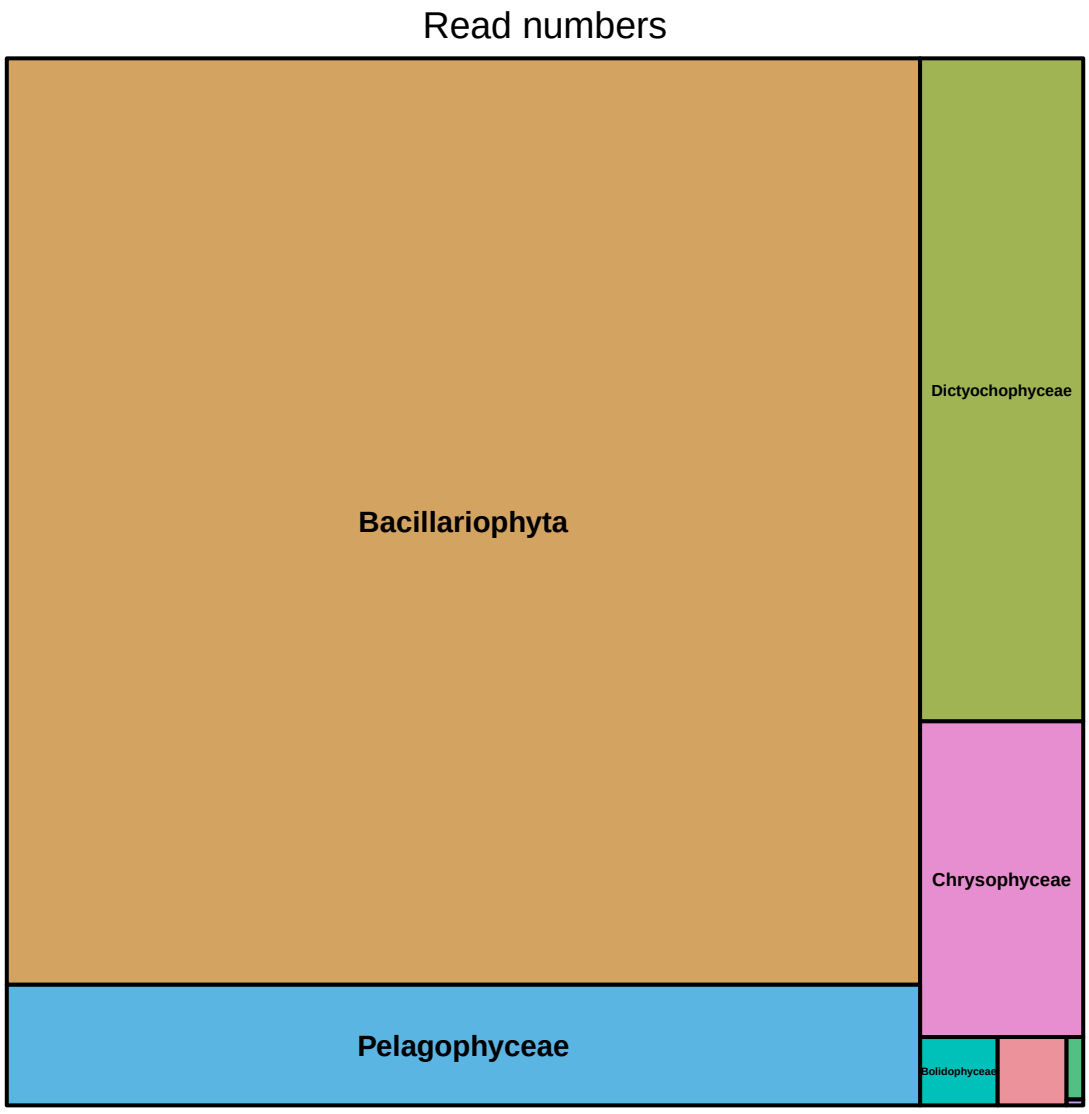
```
library("treemap")         # To do treemaps
```

Reshape the data in order to go from the wide format to go too the long format

```
tara_tree <- tara %>% select(Sample, depth_level:fraction, Strameno_all,
                             Bacillariophyta:Raphidophyceae)  %>%
                      gather(key = Class, value = n_seq,
                             Bacillariophyta:Raphidophyceae)
```

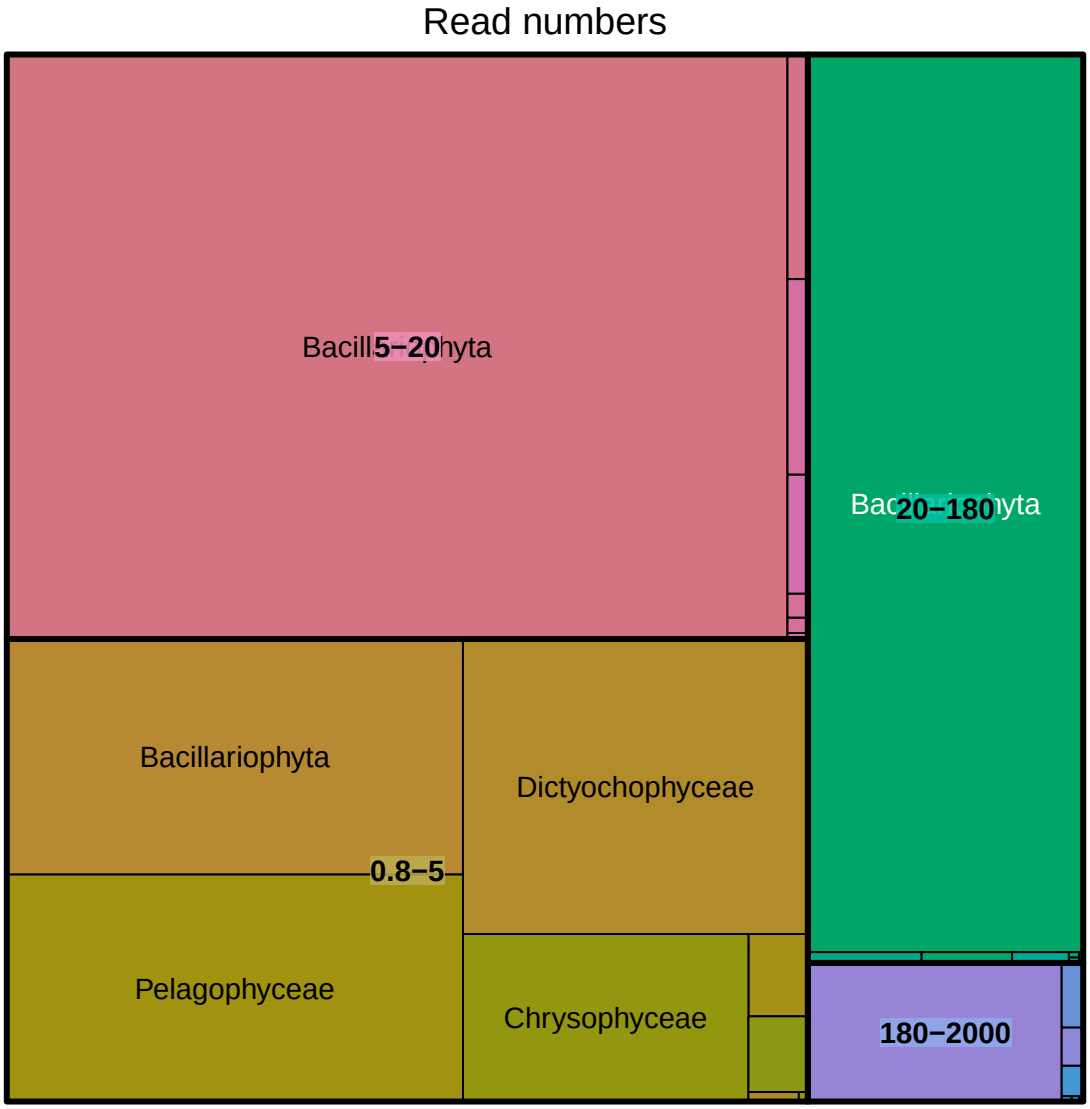
Do a global tree map

```
treemap(tara_tree, index = "Class", vSize= "n_seq", title = "Read numbers")
```



Split the tree map according to size fraction

```
treemap(tara_tree, index = c("fraction", "Class"), vSize= "n_seq", title = "Read numbers")
```



7 - Bar graphs

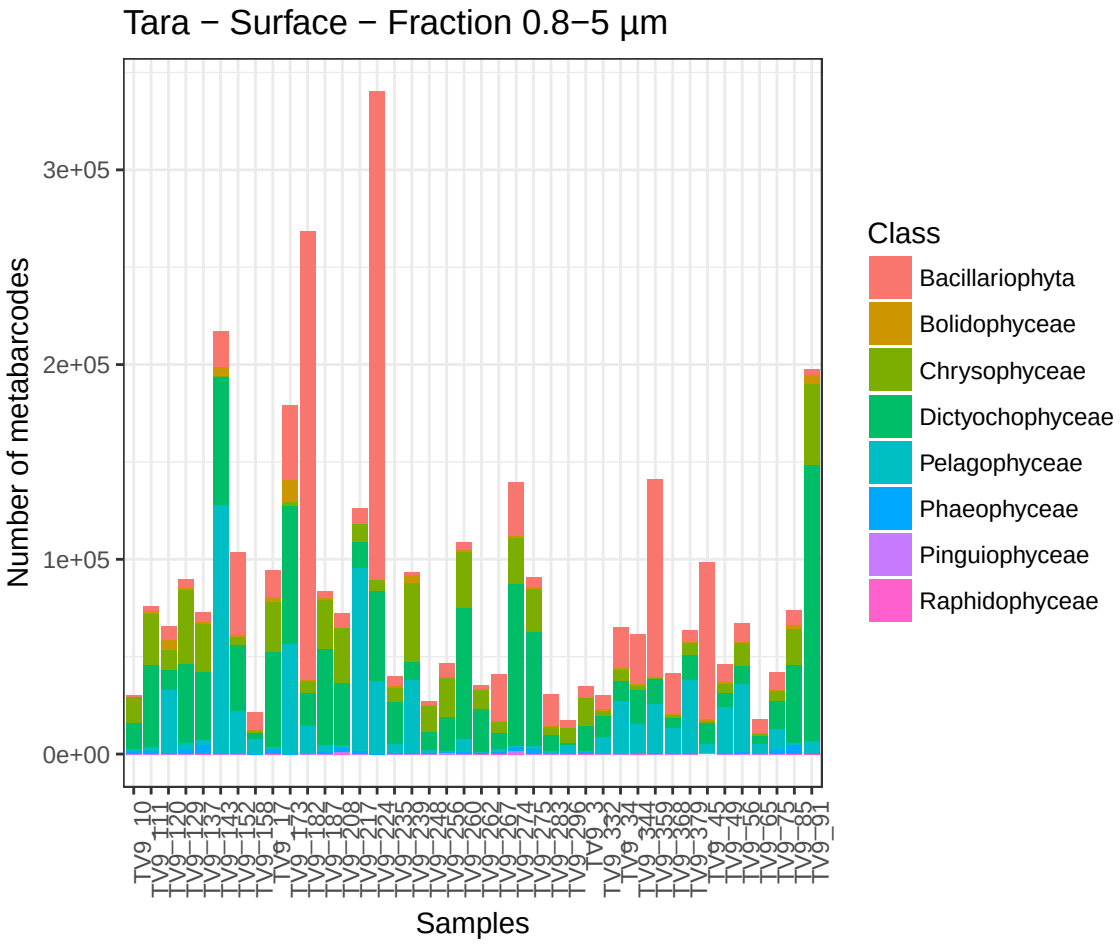
Absolute abundance

Only keep surface samples

```
tara_bar <- tara_tree %>% filter((depth_level=="SUR")&(fraction=="0.8-5"))
```

Do the bar plot for absolute read numbers
* Note : rotation of labels : `theme(axis.text.x = element_text(angle = 90, hjust = 1))`

```
ggplot(tara_bar, aes(x = Sample , y = n_seq, fill=Class) ) +
        geom_bar(stat = "identity") +
        theme_bw()  + ggtitle("Tara - Surface - Fraction 0.8-5 µm") +
        xlab("Samples")+ylab("Number of metabarcodes") +
        theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



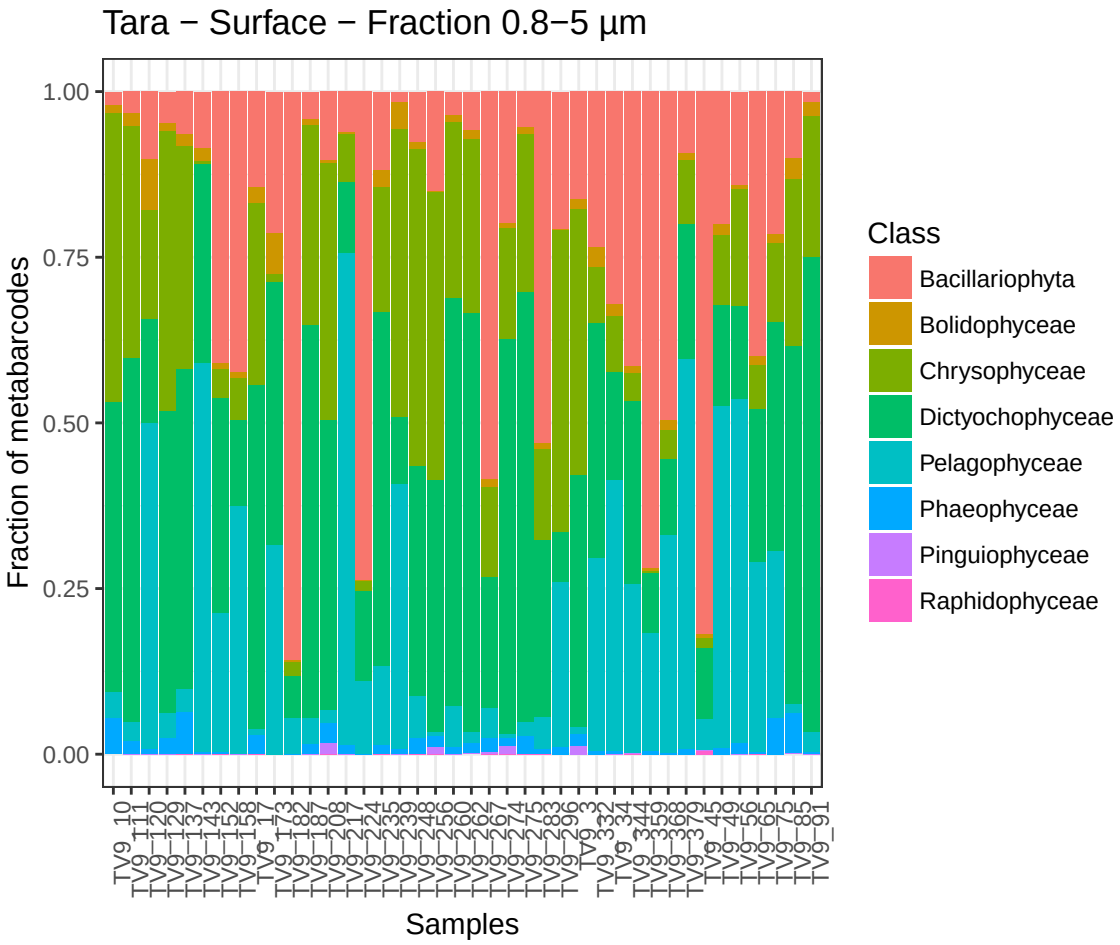
Relative abundance

Compute the relative abundance of each sequence by dividing by the total number of barcodes

```
tara_bar <- tara_bar %>% mutate(n_seq_rel = n_seq / Strameno_all)
```

Do the bar plot for relative read numbers

```
ggplot(tara_bar, aes(x = Sample , y = n_seq_rel, fill=Class) ) +
        geom_bar(stat = "identity") +
        theme_bw() + ggtitle("Tara - Surface - Fraction 0.8-5 µm") +
        xlab("Samples")+ ylab("Fraction of metabarcodes") +
        theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



8 - Heat maps

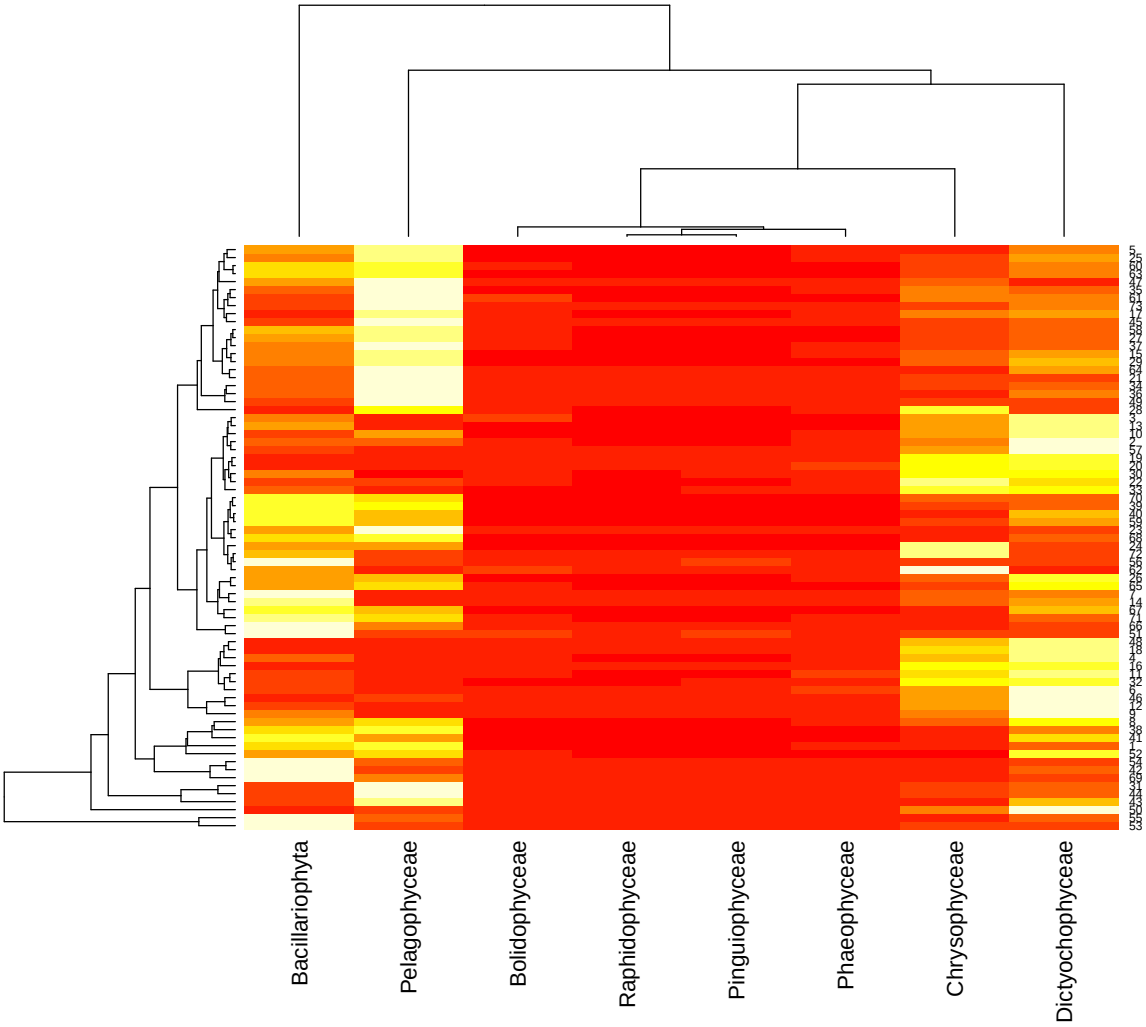
Note : for metabarcoding data use phyloseq package.

Select the fraction and columns (from Bacillariophyta to Raphidophyceae) to be plotted and transform to a matrix

```
tara_heat <- tara %>% filter(fraction=="0.8-5") %>%
                      select(Bacillariophyta:Raphidophyceae)
tara_heat.matrix <-data.matrix(tara_heat)

# It is necessary to give names to the row for heatmap labels
row.names(tara_heat.matrix) <- tara$station[fraction=="0.8-5"]
```

Draw heatmap

```
heatmap(tara_heat.matrix, margins = c(20,6) )
```


Bacillariophyta
Pelagophyceae
Bolidophyceae
Raphidophyceae
Pinguiophyceae
Phaeophyceae
Chrysophyceae
Dictyochophyceae

9 - Multivariate analysis (FactoMiner package)

```
library("FactoMineR")     # For PCA
```

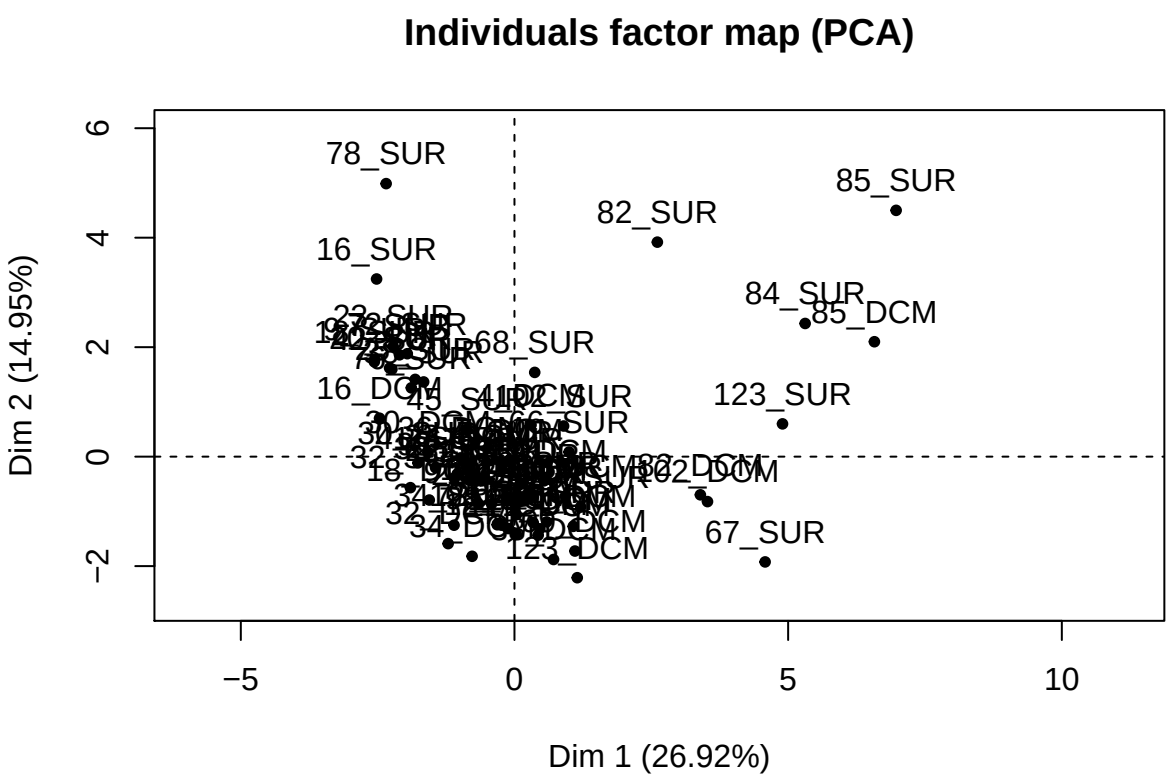
Principal component analysis (PCA)

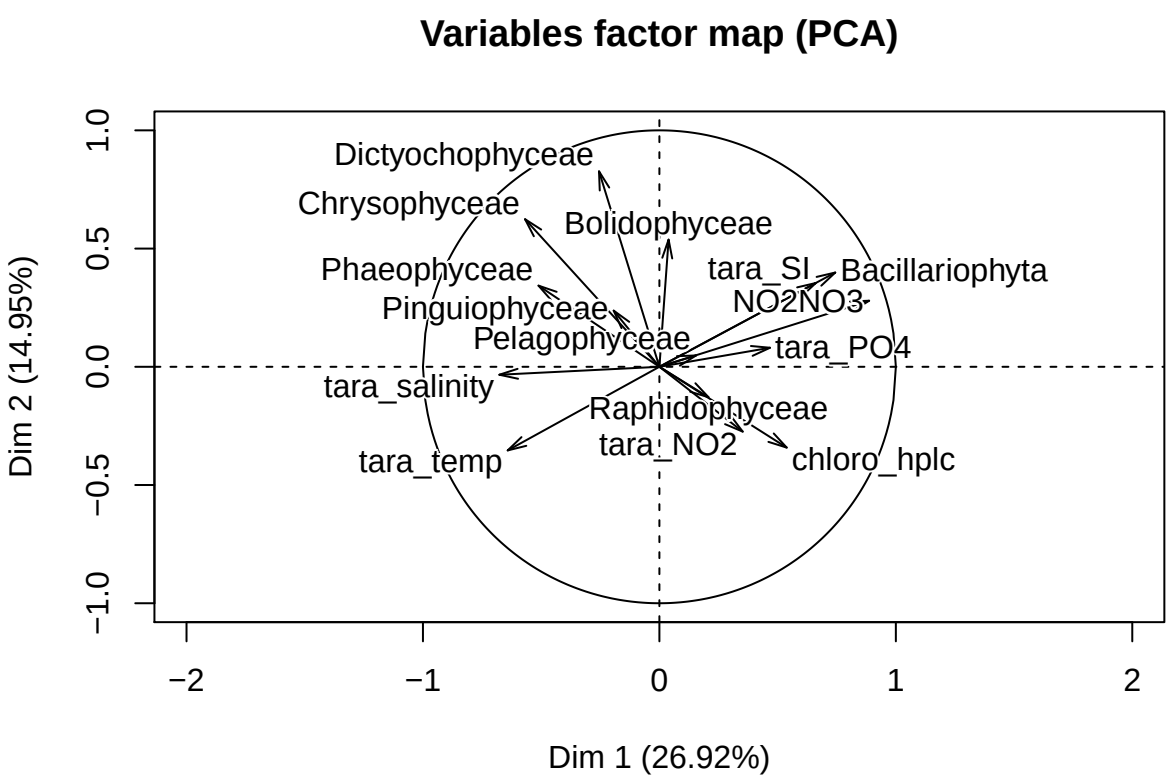
```
# Select only the 0.8-5 µm fraction and only the colums with phytplankon data and metadata
tara_multi<- tara %>% filter(fraction=="0.8-5")

# Define row names as "Station_Depth level" (points with be labelled by row names)
row.names(tara_multi)<-paste(tara_multi$station,tara_multi$depth_level,sep="_")
```

```
## Warning: Setting row names on a tibble is deprecated.
```

```
# Select only with phytoplankon data and metadata
tara_multi<- tara_multi %>% select(Bacillariophyta:Raphidophyceae, chloro_hplc:tara_salinity)

# Scale the matrix
tara_multi<- scale(tara_multi)

# Do the PCA
tara_pca<-PCA(tara_multi)
```

```
## Warning in PCA(tara_multi): Missing values are imputed by the mean of the
## variable: you should use the imputePCA function of the missMDA package
```




Variables factor map (PCA)
Dictyochophyceae
Chrysophyceae
Bolidophyceae
Phaeophyceae
tara_SI
Bacillariophyta
NO2NO3
Pinguiophyceae
Pelagophyceae
tara_PO4
tara_salinity
Raphidophyceae
tara_NO2
tara_temp
chloro_hplc
Dim 2 (14.95%)
Dim 1 (26.92%)
1.0
0.5
0.0
−0.5
−1.0
−2
−1
0
1
2

10 - Maps

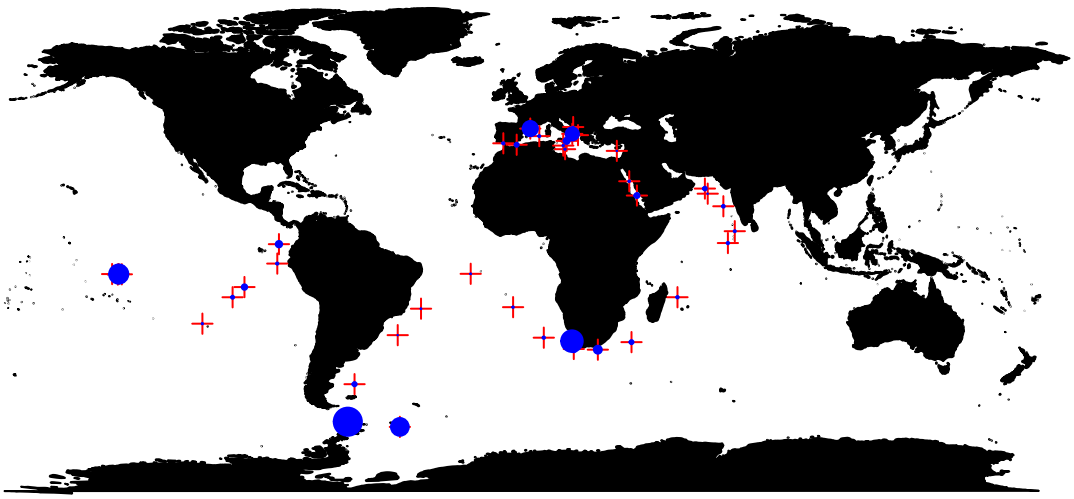
```
library("maps")            # Maps
```

Select only surface and small fraction

```
tara_map <-tara %>% filter((fraction=="0.8-5")&(depth_level=="SUR"))
```

Draw the world map and add the stations

```
# Draw the world map
map(database = "world",fill=TRUE)

# Add stations
points(tara_map$Longitude, tara_map$Latitude, pch=3, col="red", cex=1)

# Add data - circle size is proprotional to proportion of
points(tara_map$Longitude, tara_map$Latitude,
       pch=19, col="blue", cex= tara_map$Baci_pct *3/100)

# Add title
title("Bacilliorophyta as % of Photosynthetic - 0.8-5 µm - surface",cex.main=1.0)
```



11 - Manipulate sequences

In BioConductor there are many packages that can process sequences either GenBank or short reads

```
library("Biostrings")      # To manipulate sequences
```

Read sequences from metagenome (454)

```
seq<-readDNAStringSet("data/BIOSOPE_T142_reads_random.fasta", format="fasta")
```

Compute length of sequence (discard N), compute statistics and plot histogram

```
Length_seq<-letterFrequency(seq, letters="ATCG")

range(Length_seq)
```

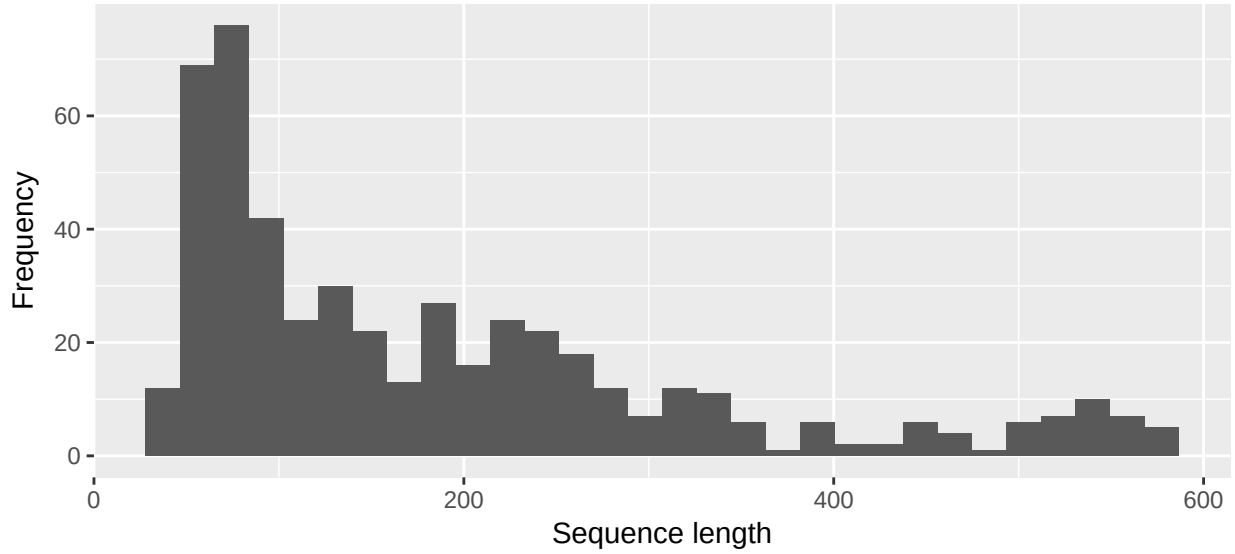
```
## [1]  41 581
```

```
mean(Length_seq)
```

```
## [1] 185.89
```

```
qplot(Length_seq, geom="histogram", xlab="Sequence length", ylab="Frequency")
```

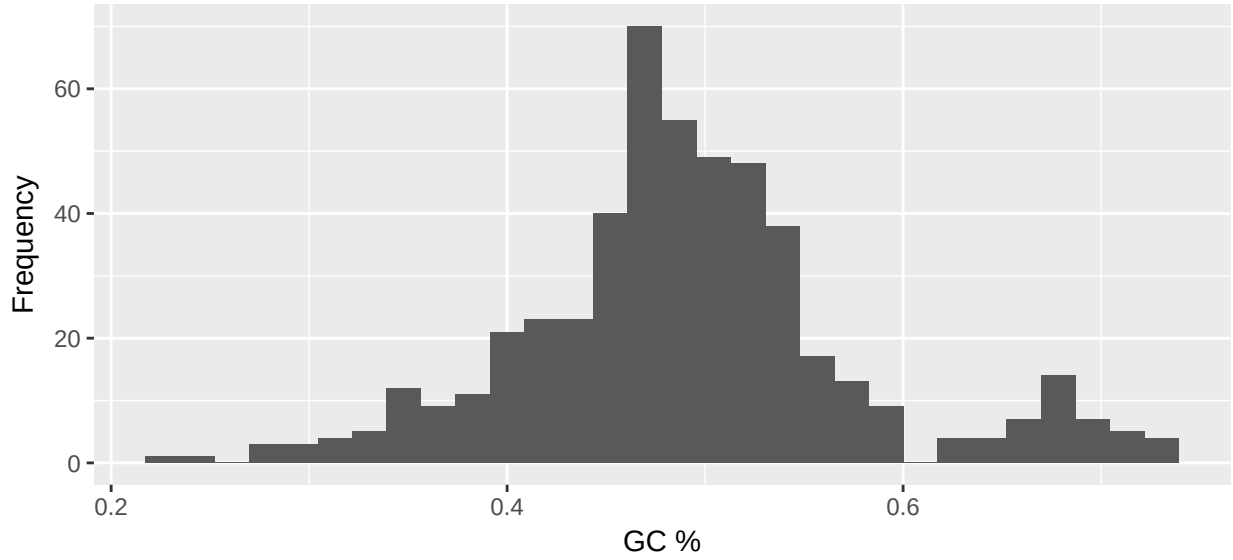
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



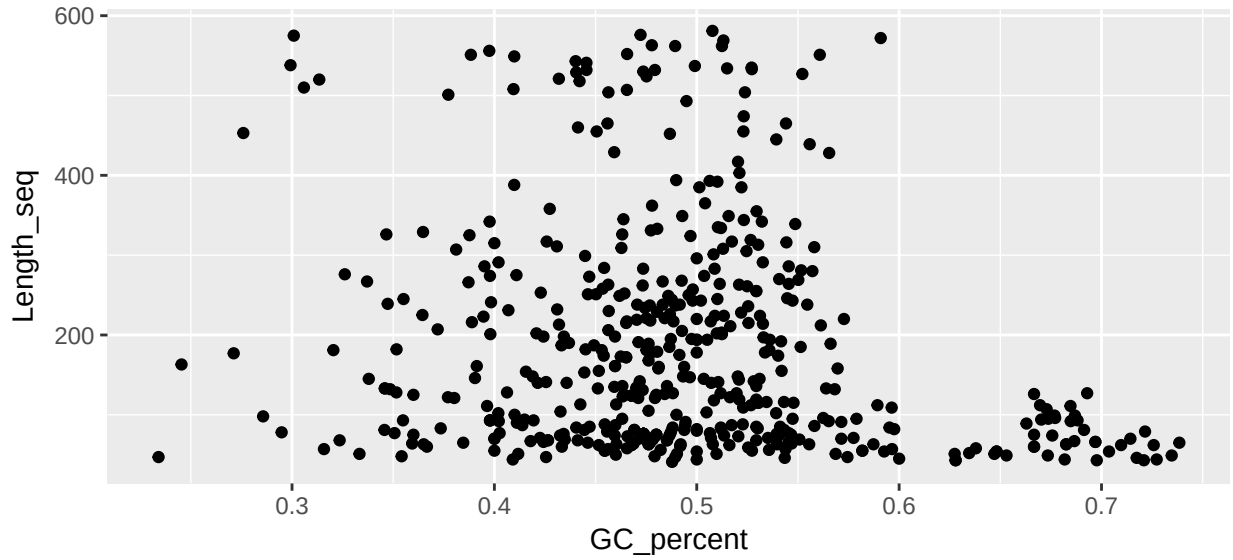
Compute GC% and do simple plots

```
# Compute number of "GC"
GC_seq <- letterFrequency(seq, letters="CG")

# Compute GC % in sequence
GC_percent <- GC_seq/Length_seq

# Do histogram
qplot(GC_percent, geom="histogram", xlab="GC %", ylab="Frequency")
```

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

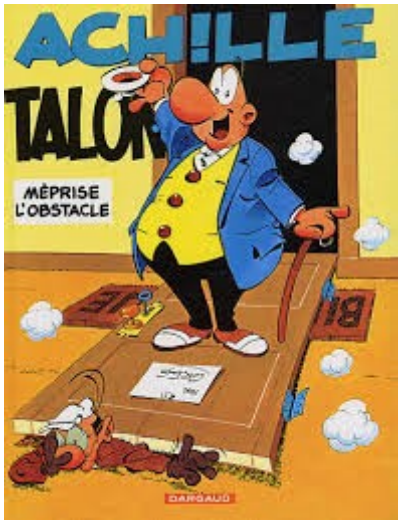



```
# Plot GC % vs Length of sequence
qplot(GC_percent,Length_seq)
```



Exercice : Load sequence from Bathycoccus and compare GC% to that of the whole metagenome

```
seq <- readDNAStringSet("data/BIOSOPE_T142_reads_Bathy.fasta", format="fasta")
```



Your turn now. These are just a few of the things you can do, possibilities are endless...