Introduction to R for microbial ecologists

Daniel Vaulot

25 11 2018

Contents

1 Aim

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

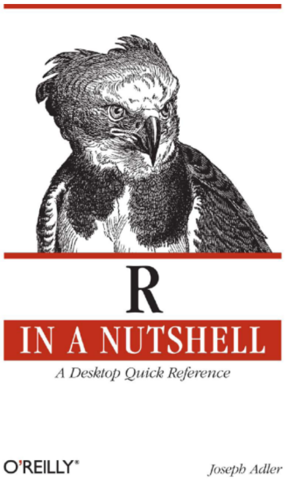
2 Prerequisites

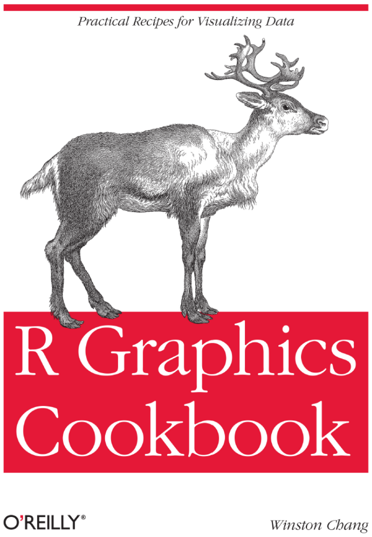
- Download from GitHub the whole set of tutorial
- Unzip the files to a folder on your computer
- Install R
- Install R studio
- Once R and R installed start R Studio and download and install the following libraries by running under R studio the following lines

```
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
install.packages("maps")       # to make maps
install.packages("treemap")    # for treemaps
install.packages("FactoMineR") # multivariate analysis

source("https://bioconductor.org/biocLite.R")
biocLite("Biostrings")         # manipulate sequences
```

3 Ressources

3.1 Books





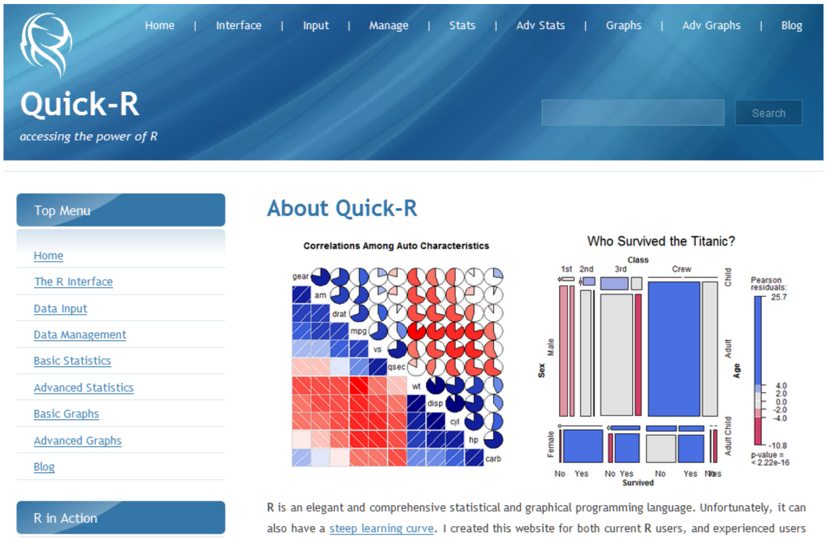
- R-intro.pdf : Very good introduction to R, short and clear
- R_in_a_nutshell.pdf : Many many receipes to solve all your questions
- R graphics cook book : very good for ggplot2

3.2 Web

3.2.1 Introductory

- Quick-R, very simple

- Maps
- Minimal R



Quick-R

accessing the power of R

About Quick-R

R is an elegant and comprehensive statistical and graphical programming language. Unfortunately, it can also have a steep learning curve. I created this website for both current R users, and experienced users

3.2.2 To go one step beyond

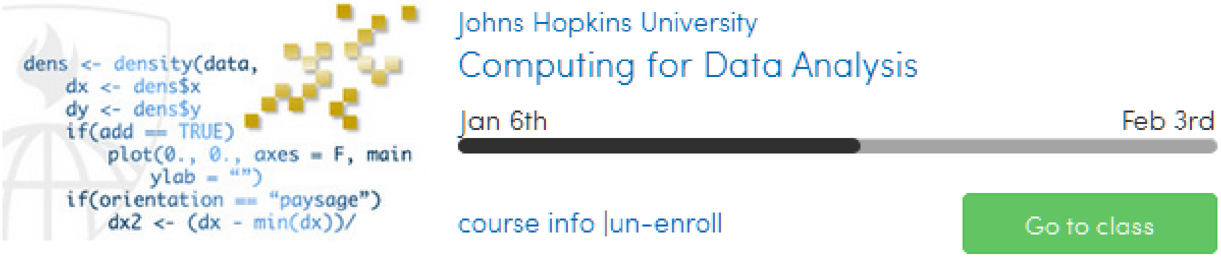
- Modern Dive : An Introduction to Statistical and Data Sciences via R
- R for Data Science
- Fundamentals of data visualization
- From data to visualization

3.3 Cheat sheets

- R basics
- ggplot2
- dplyr

3.4 On line course

- Coursera



```
dens <- density(data,
    dx <- dens$x
    dy <- dens$y
    if(add == TRUE)
        plot(0., 0., axes = F, main
            ylab = "")
    if(orientation == "paysage")
        dx2 <- (dx - min(dx))/
```

Johns Hopkins University

Computing for Data Analysis

Jan 6th Feb 3rd

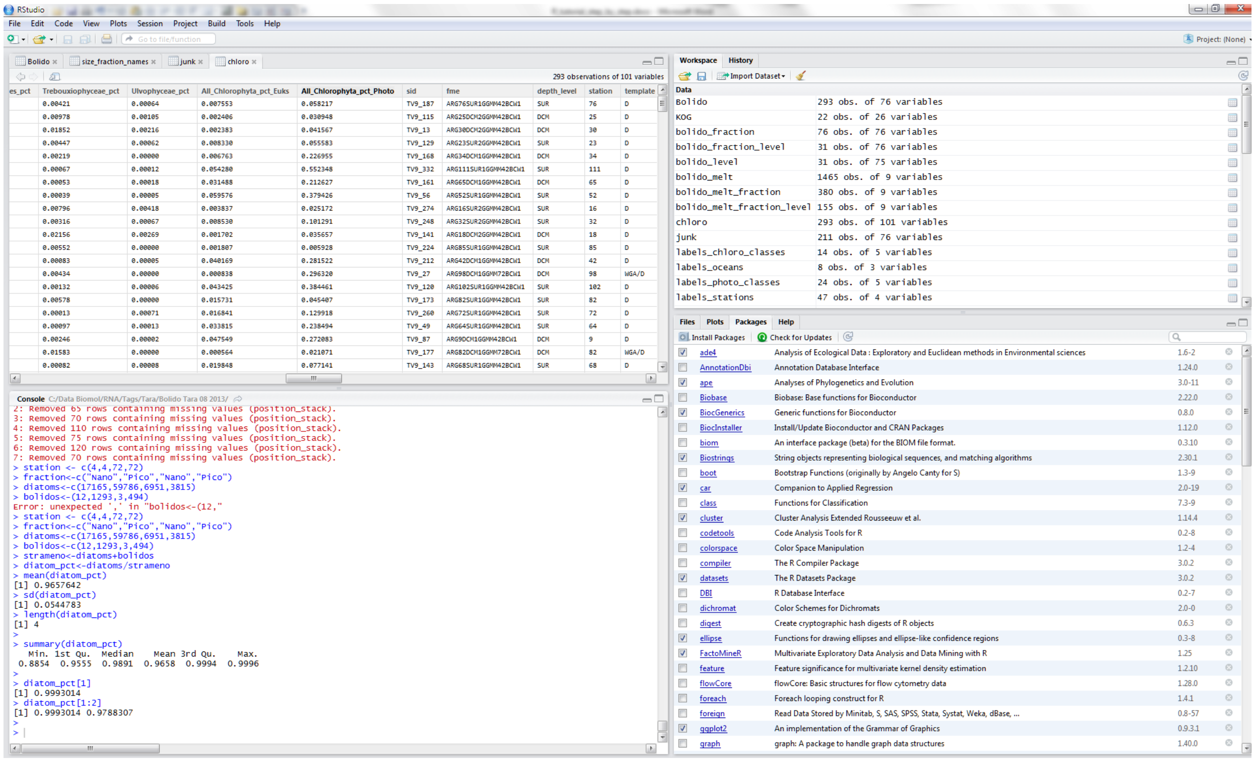
4 Step by step tutorial

4.1 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

4.2 Start R Studio

- Go to the tutorial folder
- Switch to the subdirectory `\introduction`
- Launch R Studio
- Four windows
  - top-left : script files / data tables
  - bottom -left: code
  - top - left : objects
  - bottom - right : help / libraries / files / graphics



293 observations of 101 variables

es_pct	Trebouxiophyceae_pct	Ulvophyceae_pct	All_Chlorophyta_pct_Euks	All_Chlorophyta_pct_Photo	sid	fme	depth_level	station	template
	0.00421	0.00064	0.007553	0.058217	TV9_187	ARG76SUR1GGMM42BCW1	SUR	76	D
	0.00978	0.00105	0.002406	0.030948	TV9_115	ARG25DCM2GGMM42BCW1	DCM	25	D
	0.01852	0.00216	0.002383	0.041567	TV9_13	ARG30DCM2GGMM42BCW1	DCM	30	D
	0.00447	0.00062	0.008330	0.055583	TV9_129	ARG23SUR2GGMM42BCW1	SUR	23	D
	0.00219	0.00000	0.006763	0.226955	TV9_168	ARG34DCM1GGMM42BCW1	DCM	34	D
	0.00067	0.00012	0.054280	0.552348	TV9_332	ARG111SUR1GGMM42BCW1	SUR	111	D
	0.00053	0.00018	0.031488	0.212627	TV9_161	ARG65DCM1GGMM42BCW1	DCM	65	D
	0.00039	0.00005	0.059576	0.379426	TV9_56	ARG52SUR1GGMM42BCW1	SUR	52	D
	0.00796	0.00418	0.003837	0.025172	TV9_274	ARG16SUR2GGMM42BCW1	SUR	16	D
	0.00316	0.00067	0.008530	0.101291	TV9_248	ARG32SUR2GGMM42BCW1	SUR	32	D
	0.02156	0.00269	0.001702	0.035657	TV9_141	ARG18DCM2GGMM42BCW1	DCM	18	D
	0.00552	0.00000	0.001807	0.005928	TV9_224	ARG85SUR1GGMM42BCW1	SUR	85	D
	0.00083	0.00005	0.040169	0.281522	TV9_212	ARG42DCM1GGMM42BCW1	DCM	42	D
	0.00434	0.00000	0.000838	0.296320	TV9_27	ARG98DCM1GGMM72BCW1	DCM	98	WGA/D
	0.00132	0.00006	0.043425	0.384461	TV9_120	ARG102SUR1GGMM42BCW1	SUR	102	D
	0.00578	0.00000	0.015731	0.045407	TV9_173	ARG82SUR1GGMM42BCW1	SUR	82	D
	0.00013	0.00071	0.016841	0.129918	TV9_260	ARG72SUR1GGMM42BCW1	SUR	72	D
	0.00097	0.00013	0.033815	0.238494	TV9_49	ARG64SUR1GGMM42BCW1	SUR	64	D
	0.00246	0.00002	0.047549	0.272083	TV9_87	ARG9DCM1GGMM42BCW1	DCM	9	D
	0.01583	0.00000	0.000564	0.021071	TV9_177	ARG82DCM1GGMM72BCW1	DCM	82	WGA/D
	0.00082	0.00008	0.019848	0.077141	TV9_143	ARG68SUR1GGMM42BCW1	SUR	68	D

Console C:/Data Biomol/RNA/Tags/Tara/Bolido Tara 08 2013/

```
2: Removed 65 rows containing missing values (position_stack).
3: Removed 70 rows containing missing values (position_stack).
4: Removed 110 rows containing missing values (position_stack).
5: Removed 75 rows containing missing values (position_stack).
6: Removed 120 rows containing missing values (position_stack).
7: Removed 70 rows containing missing values (position_stack).
> station <- c(4,4,72,72)
> fraction<-c("Nano","Pico","Nano","Pico")
> diatoms<-c(17165,59786,6951,3815)
> bolidos<-(12,1293,3,494)
Error: unexpected ',' in "bolidos<-(12,"
> station <- c(4,4,72,72)
> fraction<-c("Nano","Pico","Nano","Pico")
> diatoms<-c(17165,59786,6951,3815)
> bolidos<-c(12,1293,3,494)
> strameno<-diatoms+bolidos
> diatom_pct<-diatoms/strameno
> mean(diatom_pct)
[1] 0.9657642
> sd(diatom_pct)
[1] 0.0544783
> length(diatom_pct)
[1] 4
>
> summary(diatom_pct)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.8854  0.9555  0.9891  0.9658  0.9994  0.9996
>
> diatom_pct[1]
[1] 0.9993014
> diatom_pct[1:2]
[1] 0.9993014 0.9788307
>
> |
```

Data	
Bolido	293 obs. of 76 variables
KOG	22 obs. of 26 variables
bolido_fraction	76 obs. of 76 variables
bolido_fraction_level	31 obs. of 76 variables
bolido_level	31 obs. of 75 variables
bolido_melt	1465 obs. of 9 variables
bolido_melt_fraction	380 obs. of 9 variables
bolido_melt_fraction_level	155 obs. of 9 variables
chloro	293 obs. of 101 variables
junk	211 obs. of 76 variables
labels_chloro_classes	14 obs. of 5 variables
labels_oceans	8 obs. of 3 variables
labels_photo_classes	24 obs. of 5 variables
labels_stations	47 obs. of 4 variables

☑	ade4	Analysis of Ecological Data : Exploratory and Euclidean methods in Environmental sciences	1.6-2
☐	AnnotationDbi	Annotation Database Interface	1.24.0
☑	ape	Analyses of Phylogenetics and Evolution	3.0-11
☐	Biobase	Biobase: Base functions for Bioconductor	2.22.0
☑	BiocGenerics	Generic functions for Bioconductor	0.8.0
☐	BiocInstaller	Install/Update Bioconductor and CRAN Packages	1.12.0
☐	biom	An interface package (beta) for the BIOM file format.	0.3.10
☑	Biostrings	String objects representing biological sequences, and matching algorithms	2.30.1
☐	boot	Bootstrap Functions (originally by Angelo Canty for S)	1.3-9
☑	car	Companion to Applied Regression	2.0-19
☐	class	Functions for Classification	7.3-9
☑	cluster	Cluster Analysis Extended Rousseeuw et al.	1.14.4
☐	codetools	Code Analysis Tools for R	0.2-8
☐	colorspace	Color Space Manipulation	1.2-4
☐	compiler	The R Compiler Package	3.0.2
☑	datasets	The R Datasets Package	3.0.2
☐	DBI	R Database Interface	0.2-7
☐	dichromat	Color Schemes for Dichromats	2.0-0
☐	digest	Create cryptographic hash digests of R objects	0.6.3
☑	ellipse	Functions for drawing ellipses and ellipse-like confidence regions	0.3-8
☑	FactoMineR	Multivariate Exploratory Data Analysis and Data Mining with R	1.25
☐	feature	Feature significance for multivariate kernel density estimation	1.2.10
☐	flowCore	flowCore: Basic structures for flow cytometry data	1.28.0
☐	foreach	Foreach looping construct for R	1.4.1
☐	foreign	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, Weka, dBase, ...	0.8-57
☑	ggplot2	An implementation of the Grammar of Graphics	0.9.3.1
☐	graph	graph: A package to handle graph data structures	1.40.0

4.3 Load necessary libraries

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

4.4 Create simple vectors and data frame

4.4.1 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

```r
# 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)
```

4.4.2 Compute new quantities

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

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

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

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

4.4.3 Compute statistics

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

```
[1] 0.9659098
```

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

```
[1] 0.05459839
```

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

```
[1] 4
```

```r
# 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 
```

4.4.4 Accessing subsets

```r
diatoms_pct[1]
```

```
[1] 0.9998835
```

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

```
[1] 0.9998835 0.9788307
```

4.4.5 Data frames

```r
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
```

4.4.6 Access individual columns

```r
tara$diatoms
```

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

4.4.7 Access specific lines

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

```
[1] 17165 59786
```

4.4.8 Compute statistics of a specific group

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

```
[1] 38475.5
```

4.4.9 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

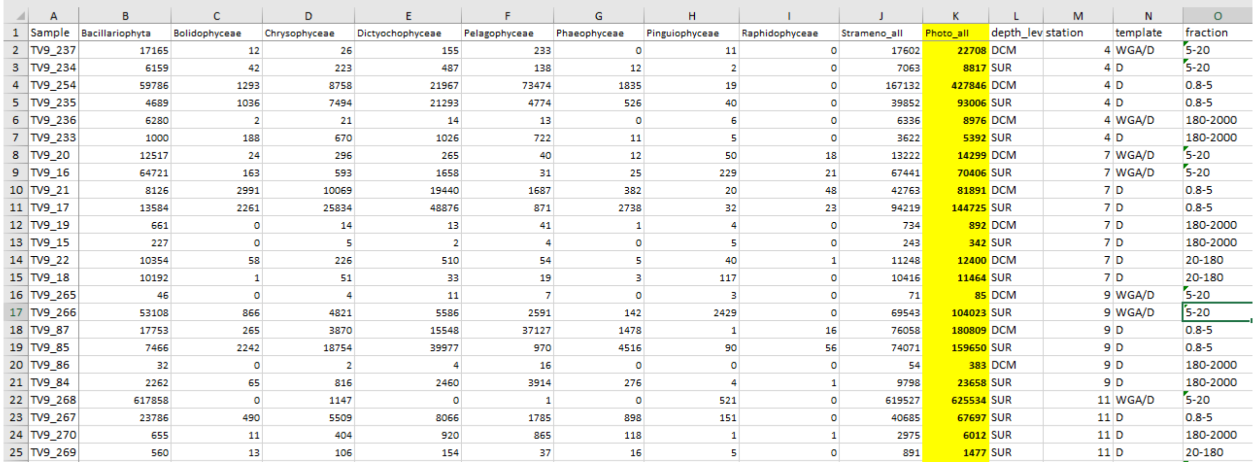
```
# Using the tapply function
tapply(tara$diatoms, tara$station, mean)
```

```
      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
```

4.5 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

4.5.1 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 1334743
 $ 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 ...
```

4.5.2 The easy way - Read directly Excel (readxl library)

```r
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

```r
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-Sep-2
 $ 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 ...
```

4.6 Compute derived quantities and Statistics (using dplyr library)

Compute % of Bacilliarophyta and Pelagophyceae vs Total photosynthetic

```r
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

```r
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
```

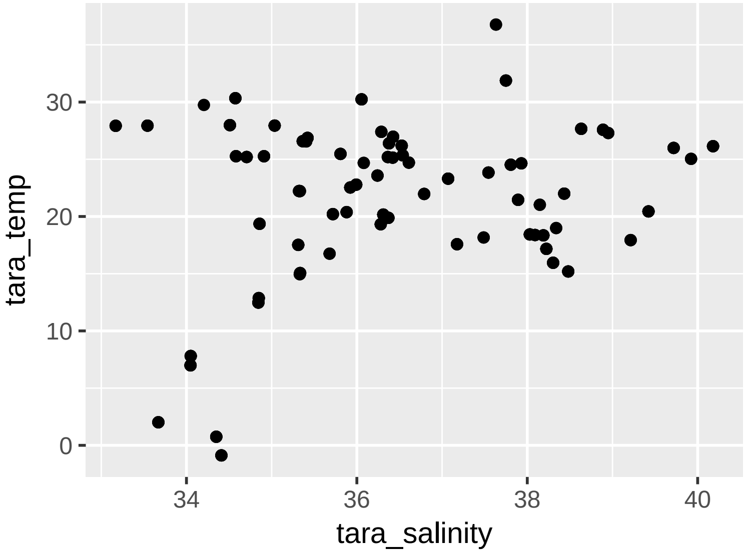
4.7 Do simple X-Y plots (using ggplot2 library)

Load the ggplot2 library

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

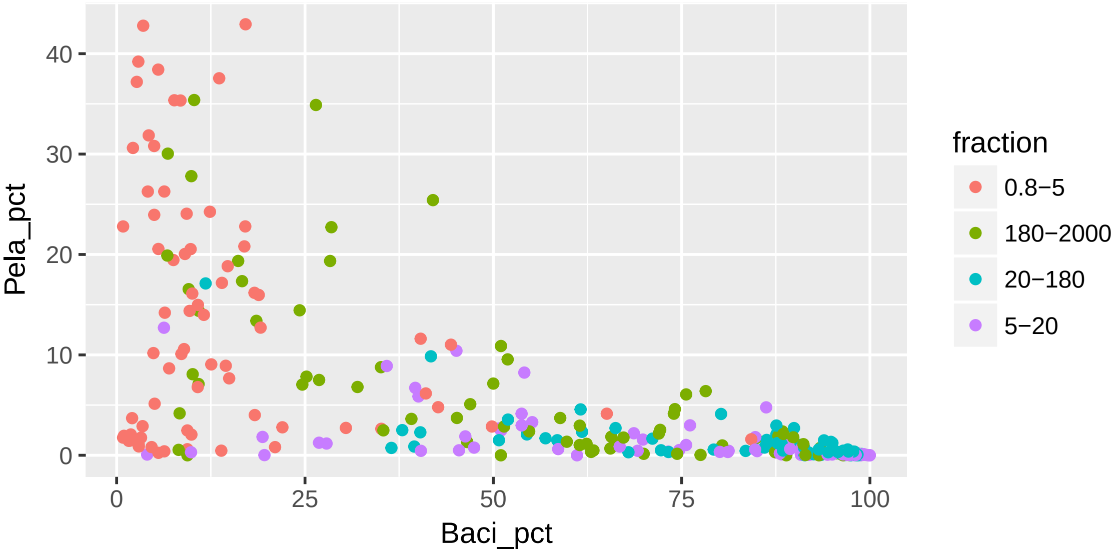
X vs Y

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



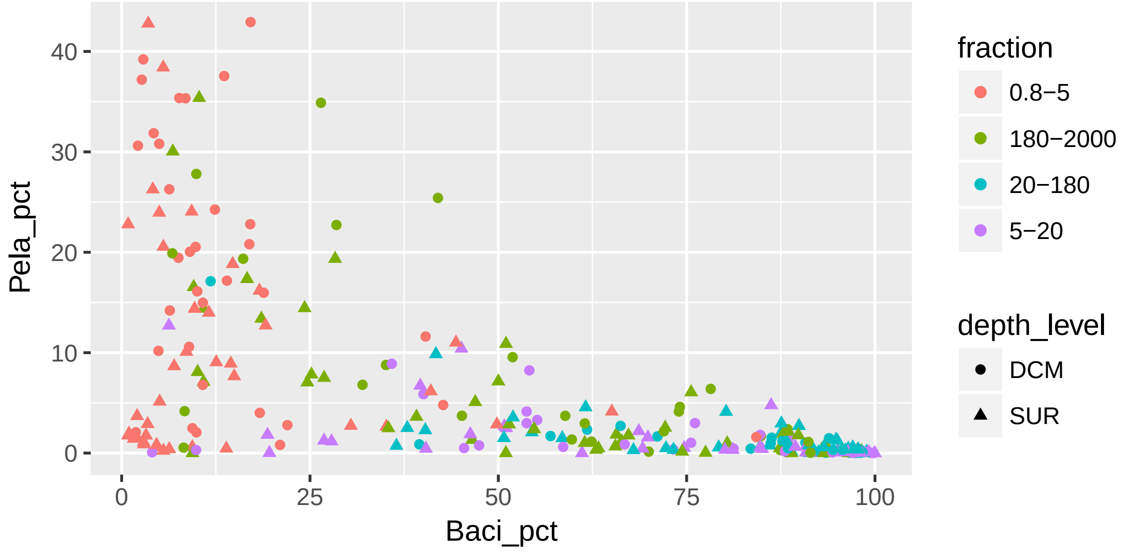
X vs Y with variation in color of points with size fraction

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



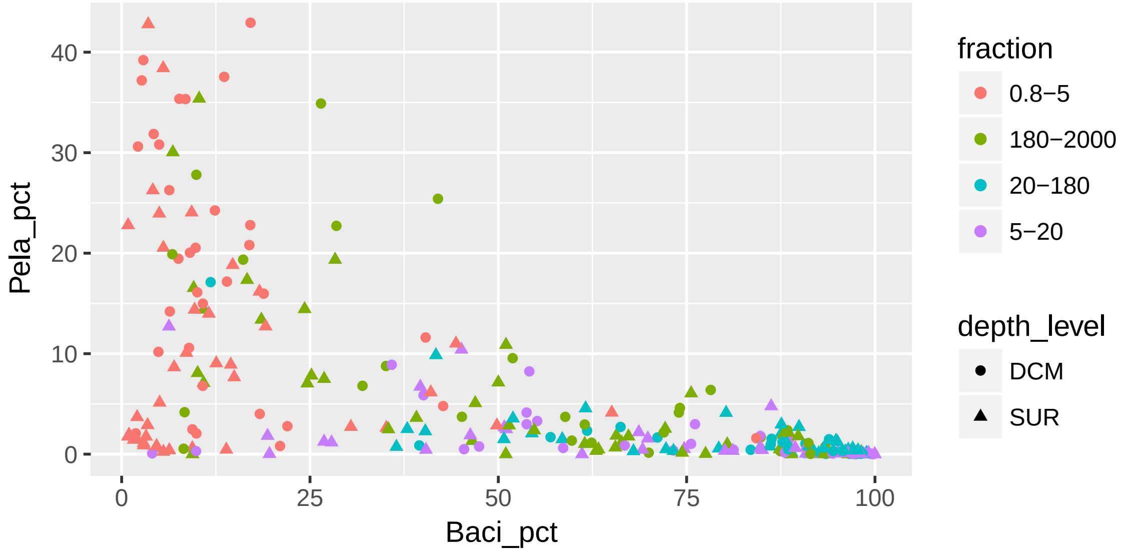
X vs Y with variation in color of points with size fraction and shape with depth level

```r
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

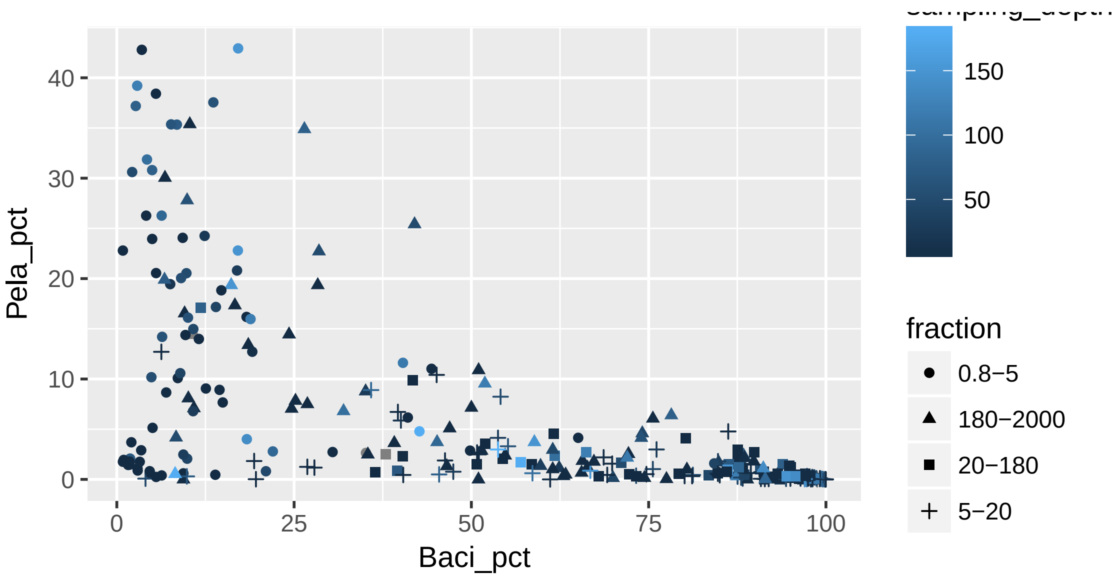
```r
qplot(Baci_pct, Pela_pct, data = tara, color = fraction, shape = depth_level)
```



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

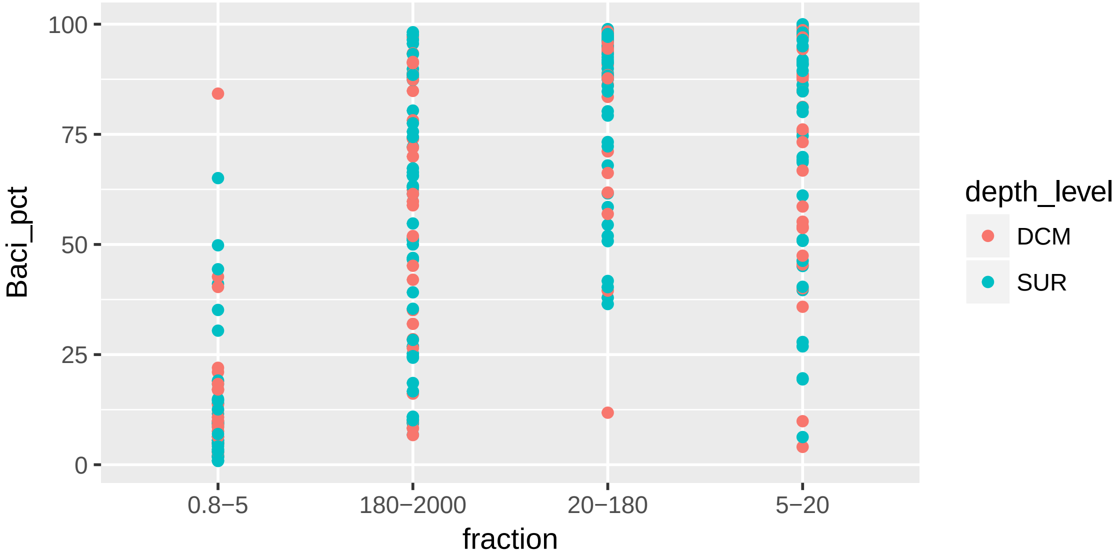
Note that sampling_depth is a **continuous variable**

```
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

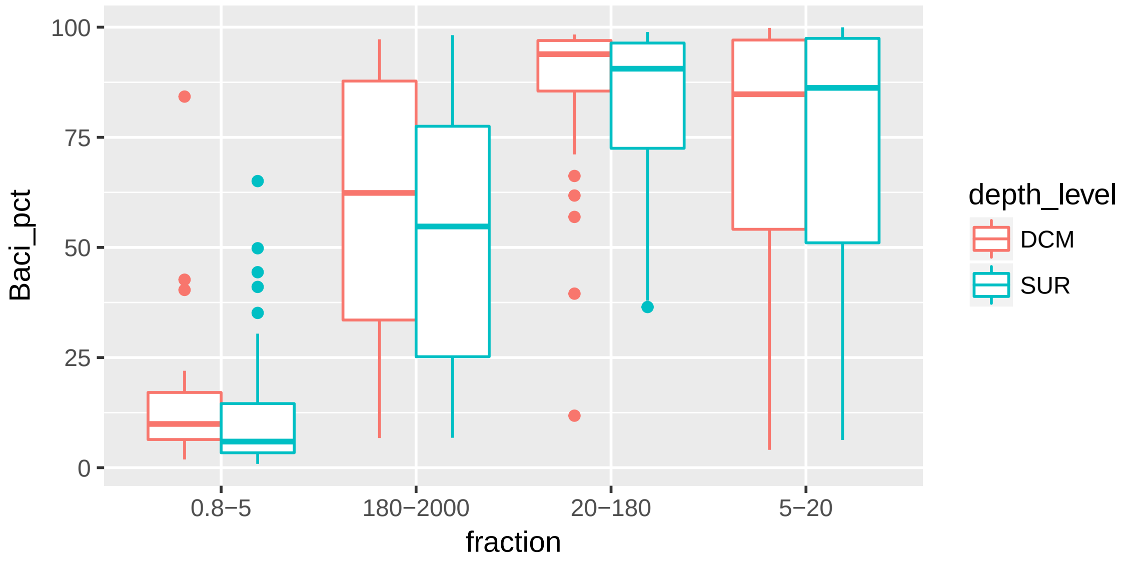
```
qplot(fraction, Baci_pct, data = tara, color = depth_level)
```



4.8 Other types of plots

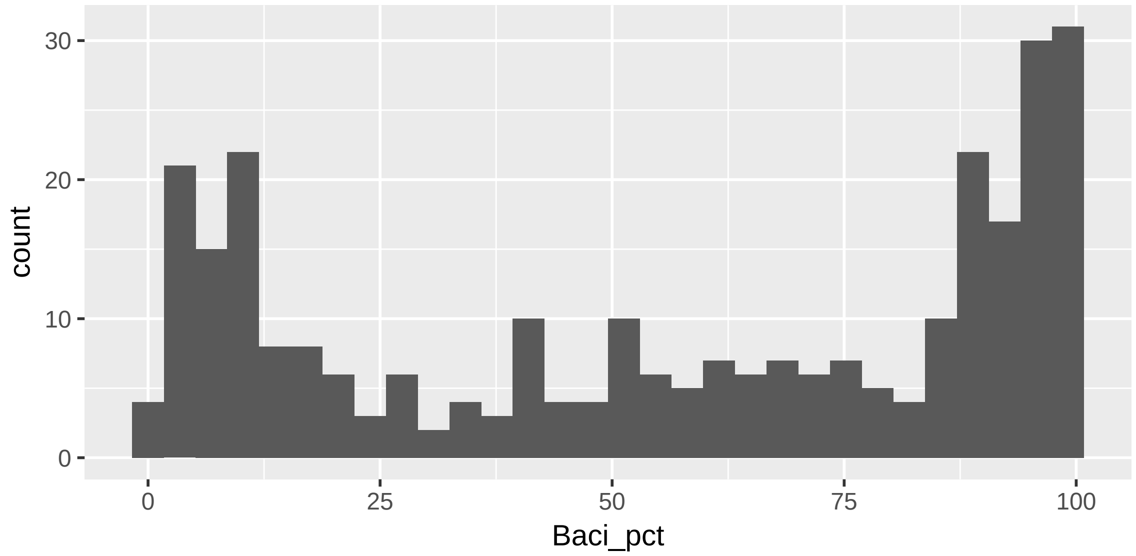
Boxplot for the same data

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



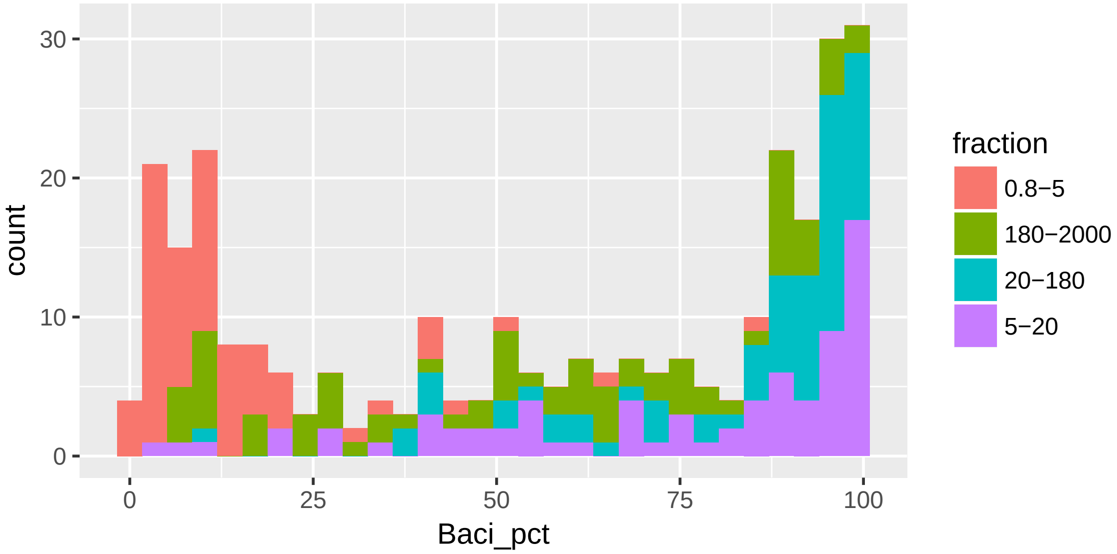
Histogram for all the data

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



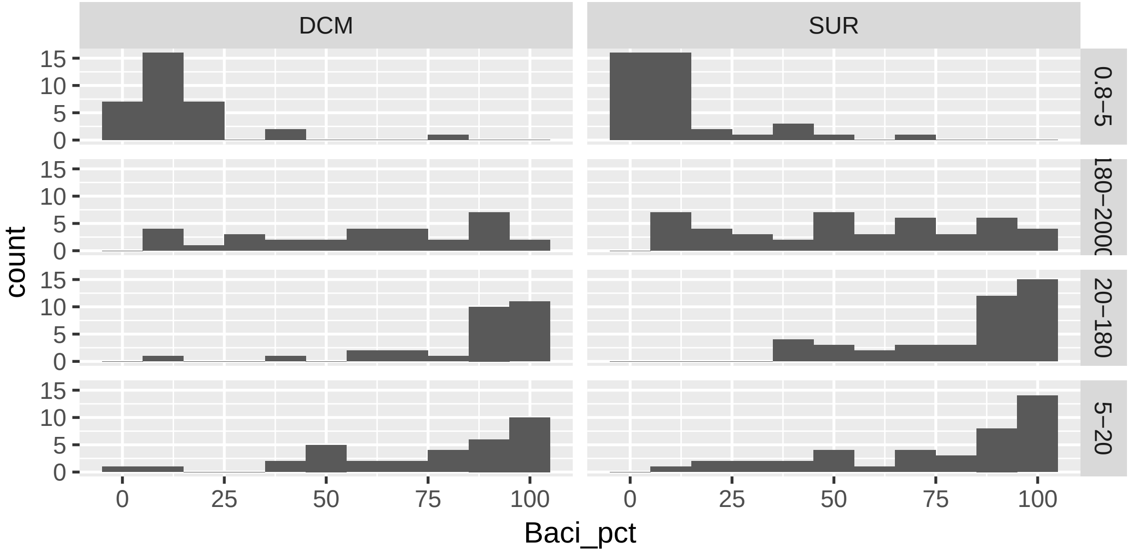
Histogram with different color for each size fraction

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



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)
```



4.9 Tree maps (much better than Pie charts...)

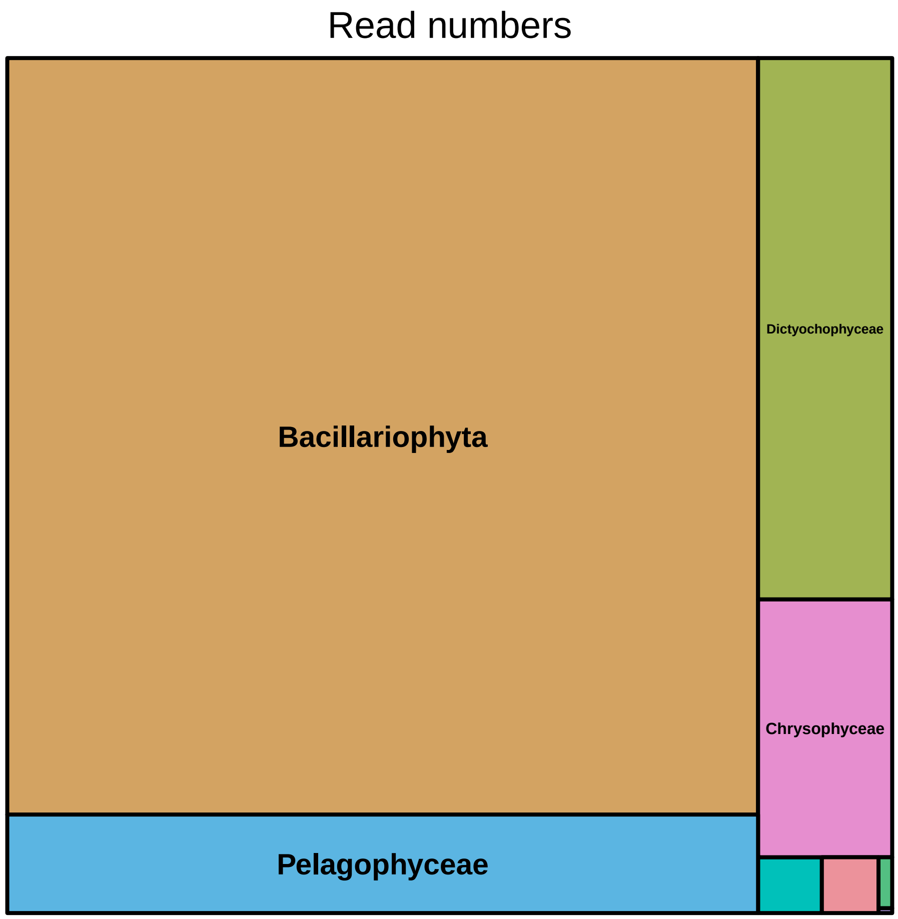
```r
library("treemap")  # To do treemaps
```

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

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

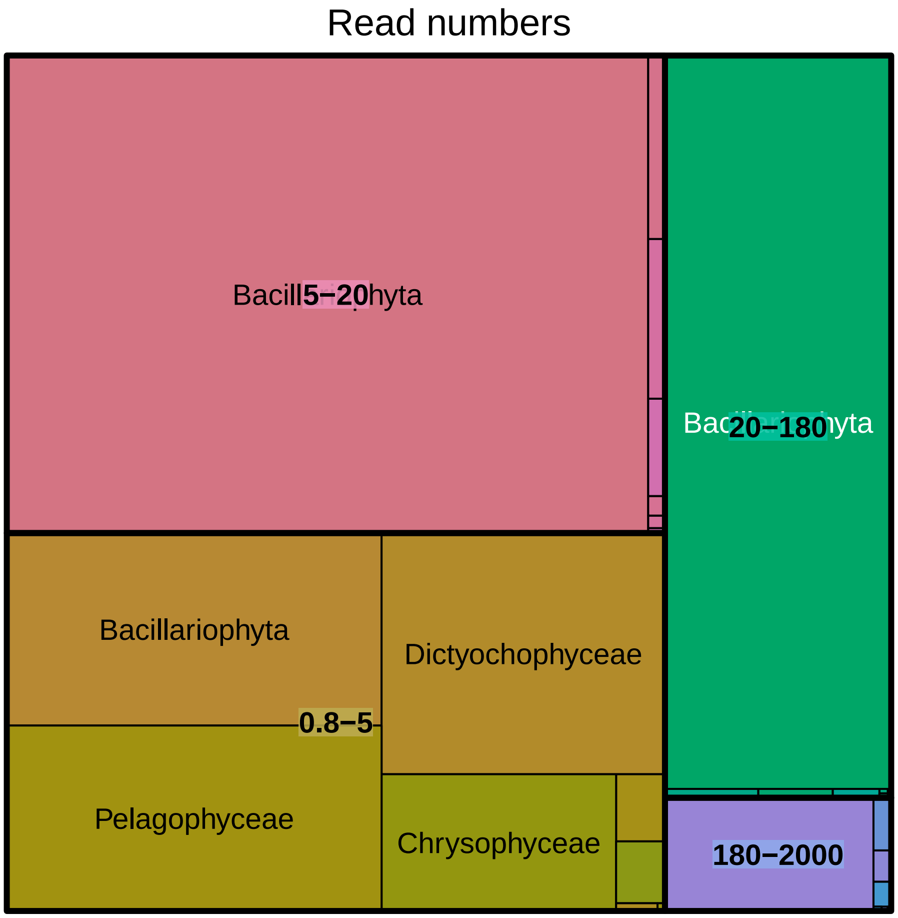
Do a global tree map

```r
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")
```



4.10 Bar graphs

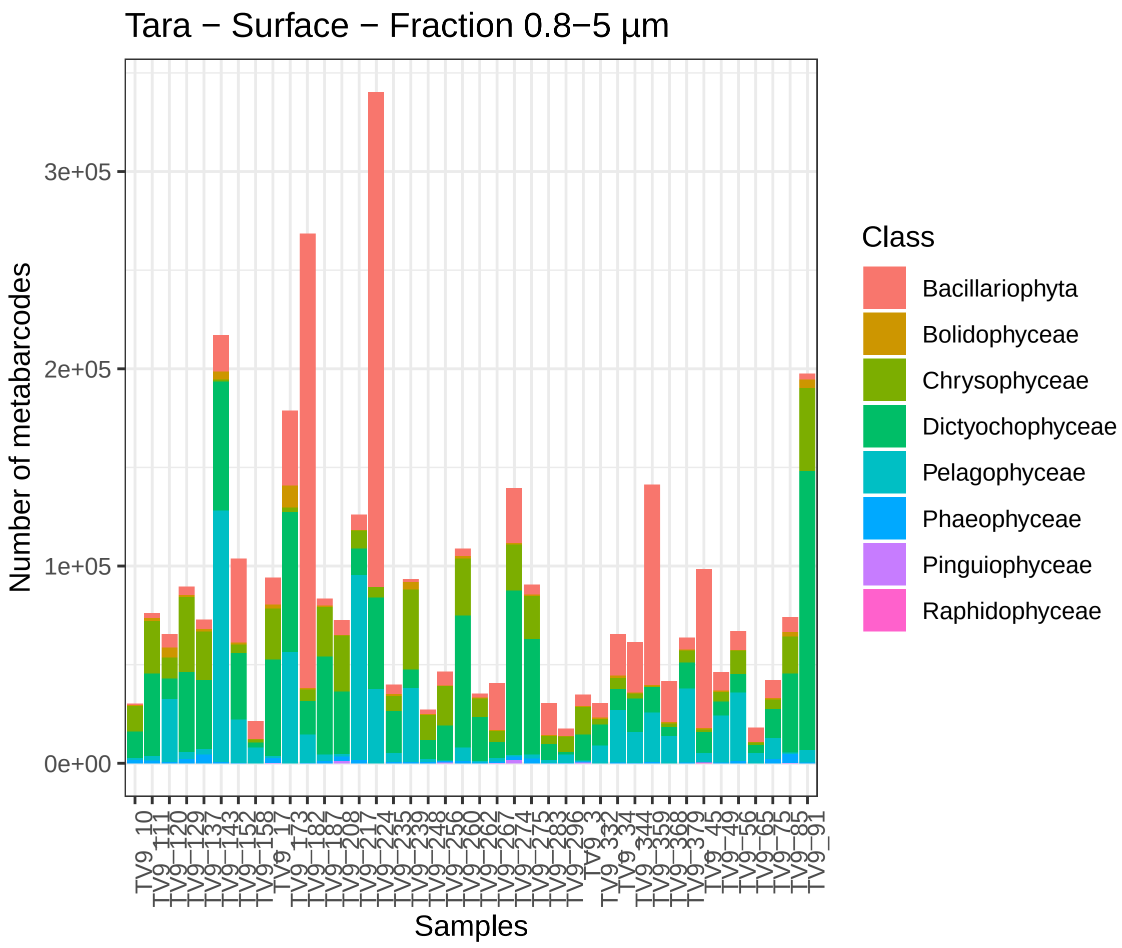
4.10.1 Absolute abundance

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))
```



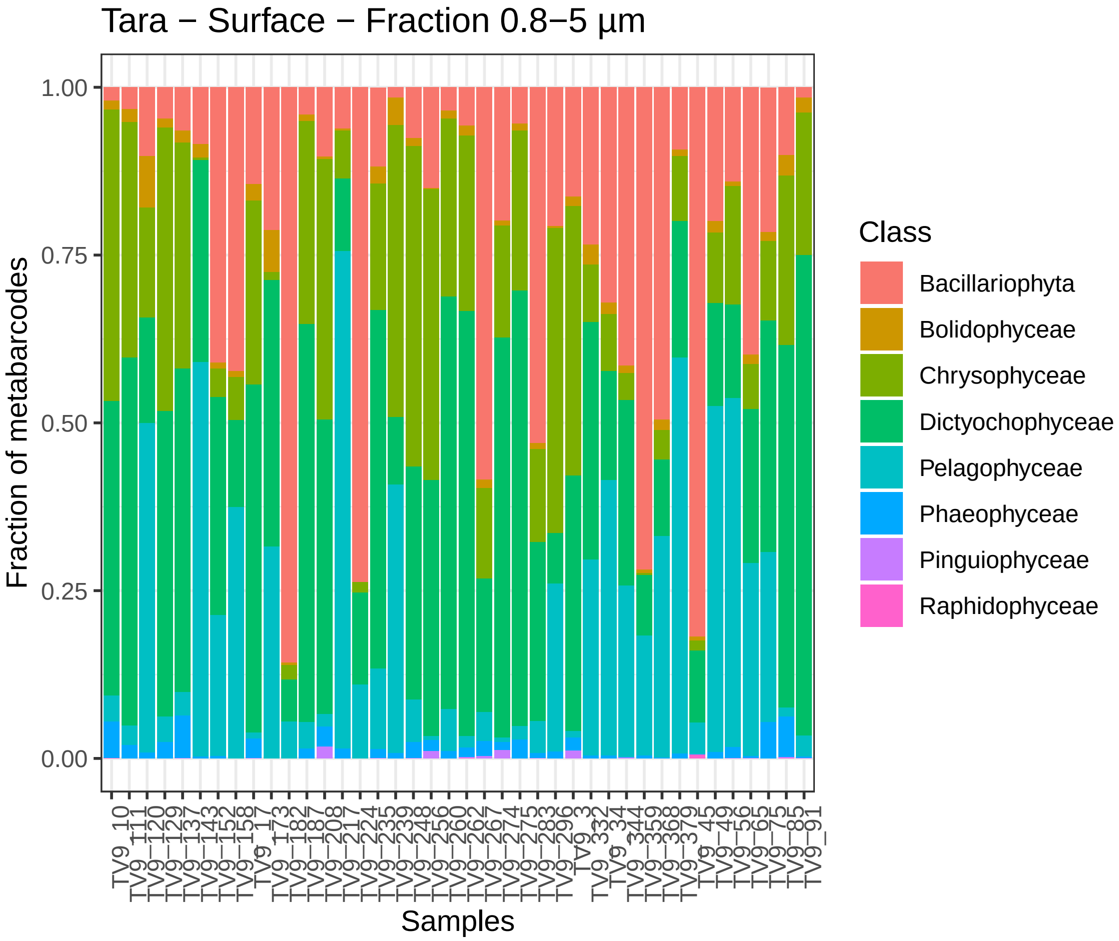
4.10.2 Relative abundance

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))
```



4.11 Heat maps

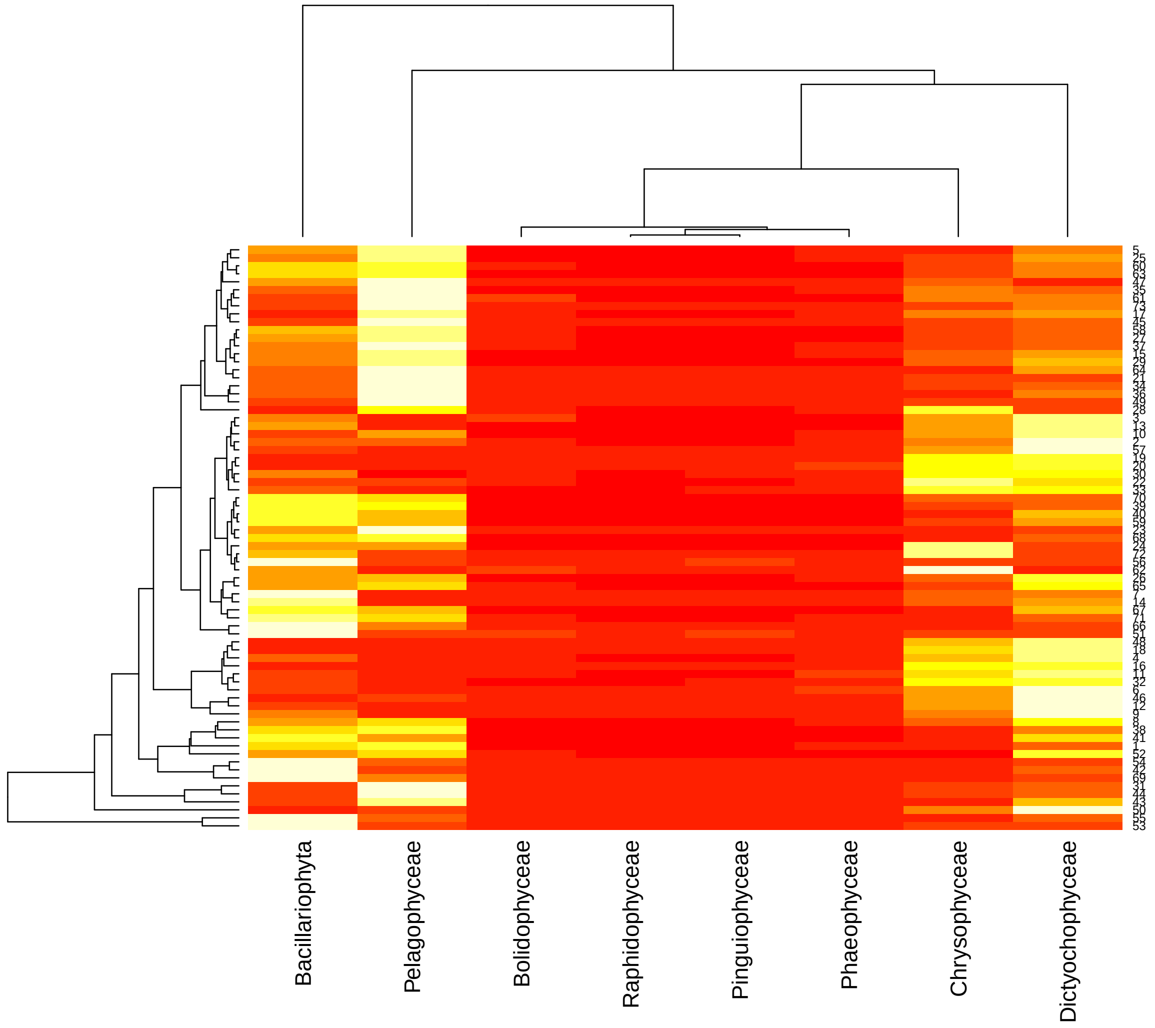
Note : for metabarcoding data use phyloseq package.

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))
```

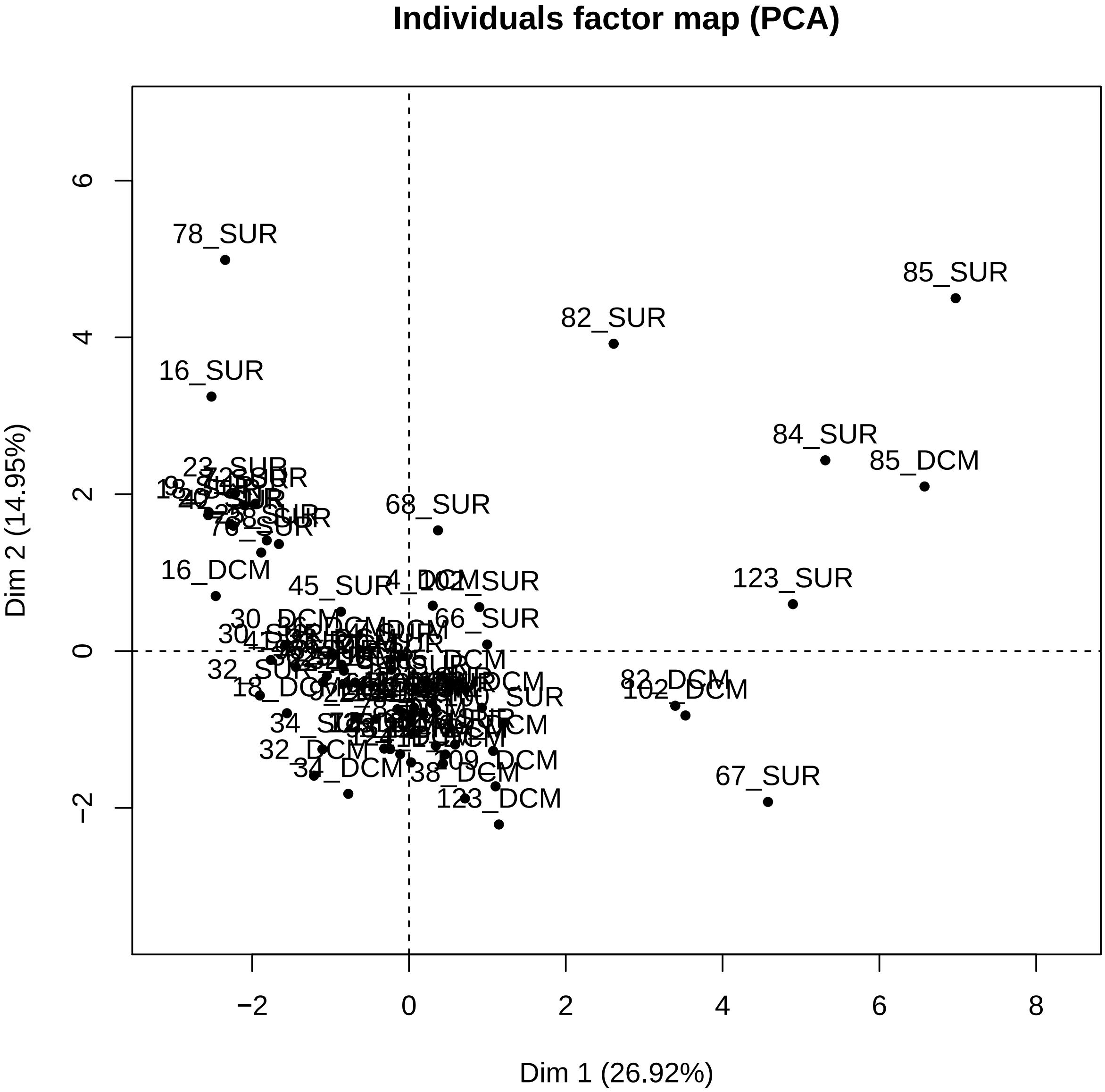


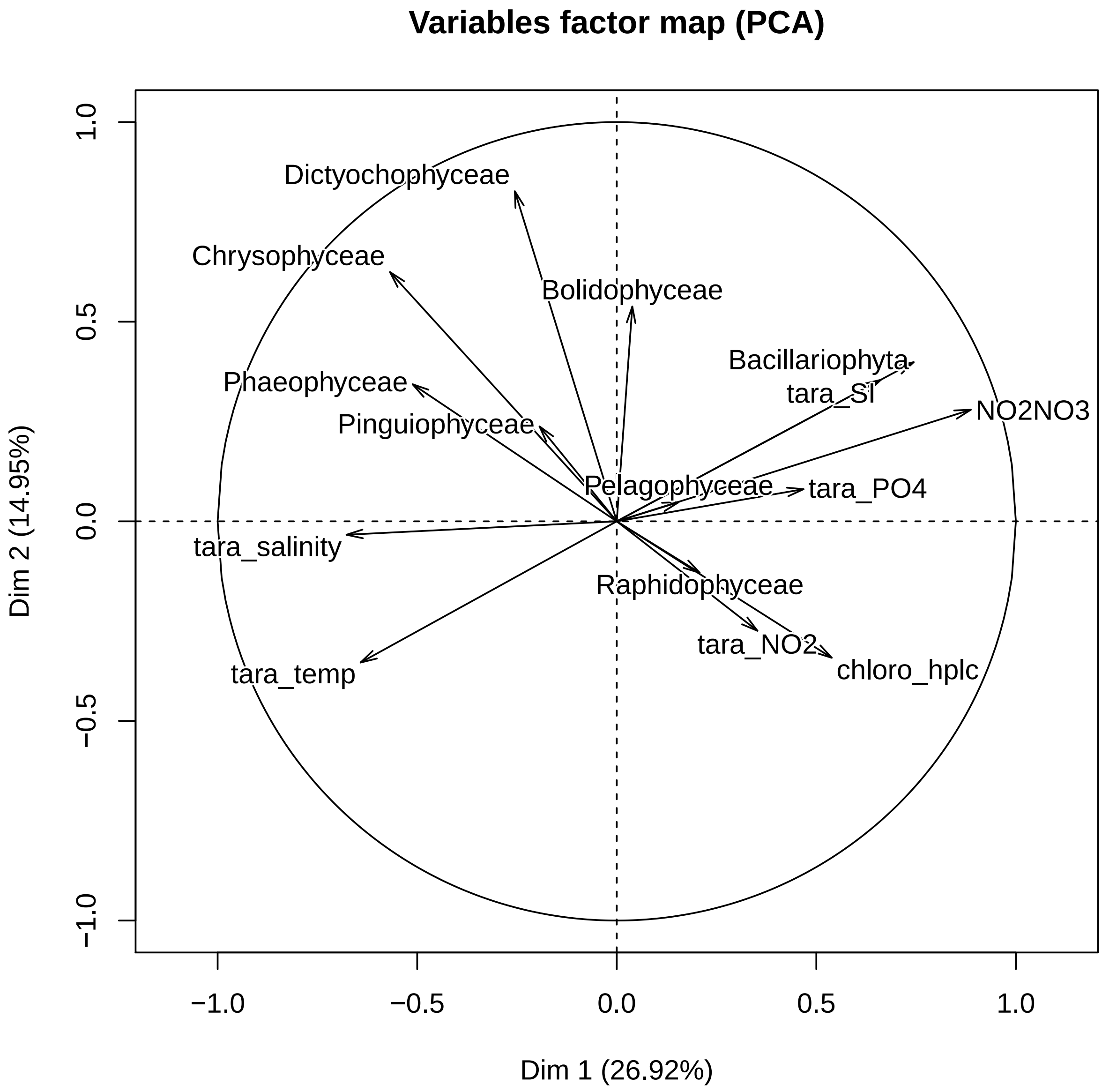
4.12 Multivariate analysis (FactoMiner package)

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

Principal component analysis (PCA)

```r
# 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 = "_")

# 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)
```


Individuals factor map (PCA)
78_SUR
85_SUR
82_SUR
16_SUR
84_SUR
85_DCM
23_SUR
68_SUR
76_SUR
16_DCM
45_SUR
4_DCM
102_SUR
66_SUR
123_SUR
30_DCM
32_SUR
18_DCM
82_DCM
102_DCM
34_SUR
32_DCM
34_DCM
38_DCM
109_DCM
123_DCM
67_SUR
Dim 2 (14.95%)
Dim 1 (26.92%)
6
4
2
0
−2
−2
0
2
4
6
8



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

4.13 Maps

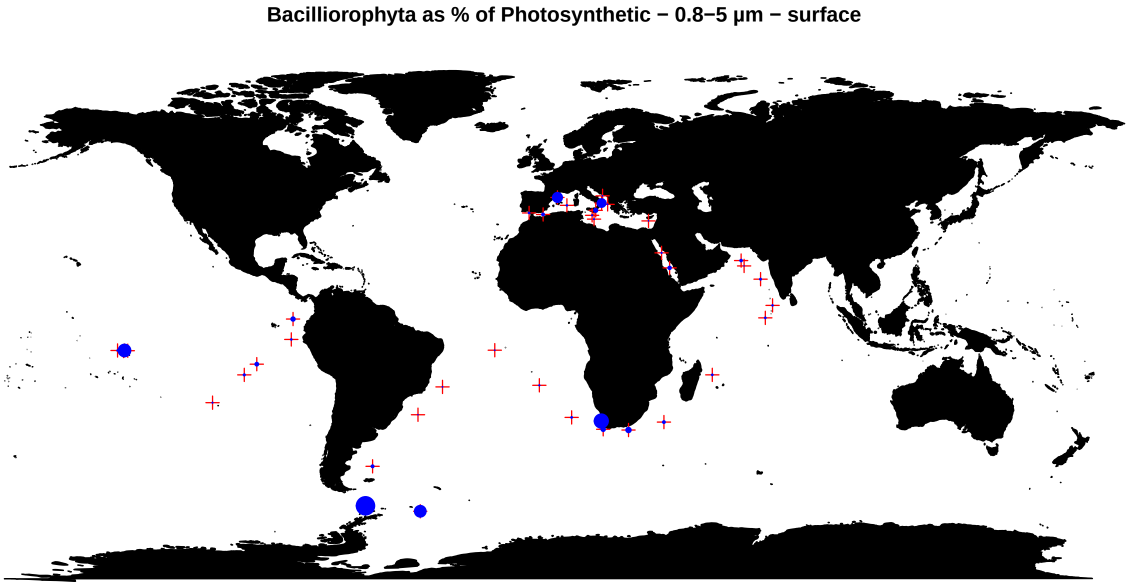
```
library("maps")  # Maps
```

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)
```



4.14 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

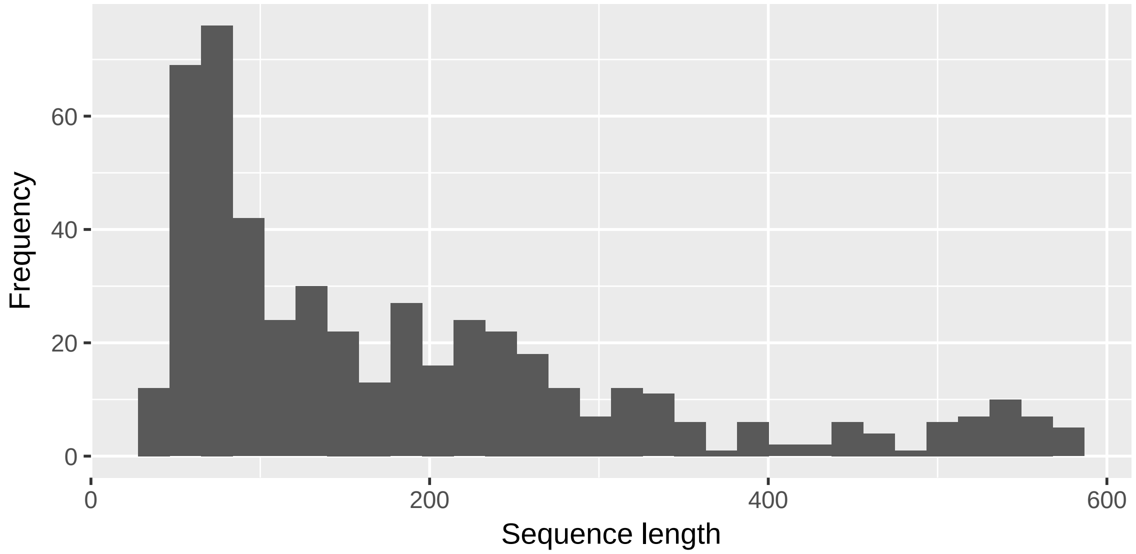
```
Length_seq <- letterFrequency(seq, letters = "ATCG")

range(Length_seq)
```

```
[1]  41 581
```

```
mean(Length_seq)
```

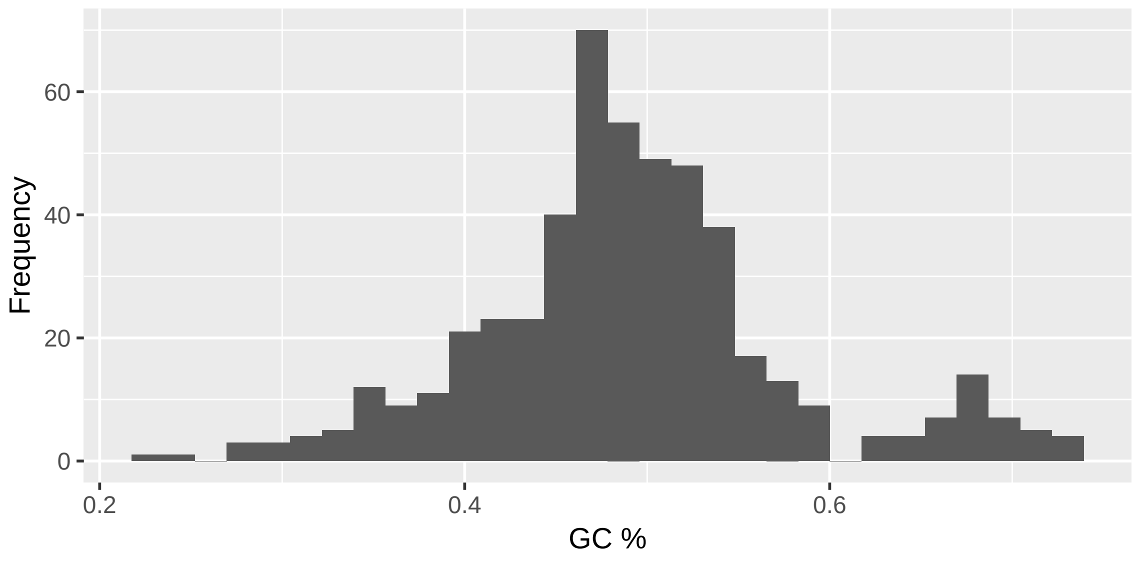
```
[1] 185.89
```

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

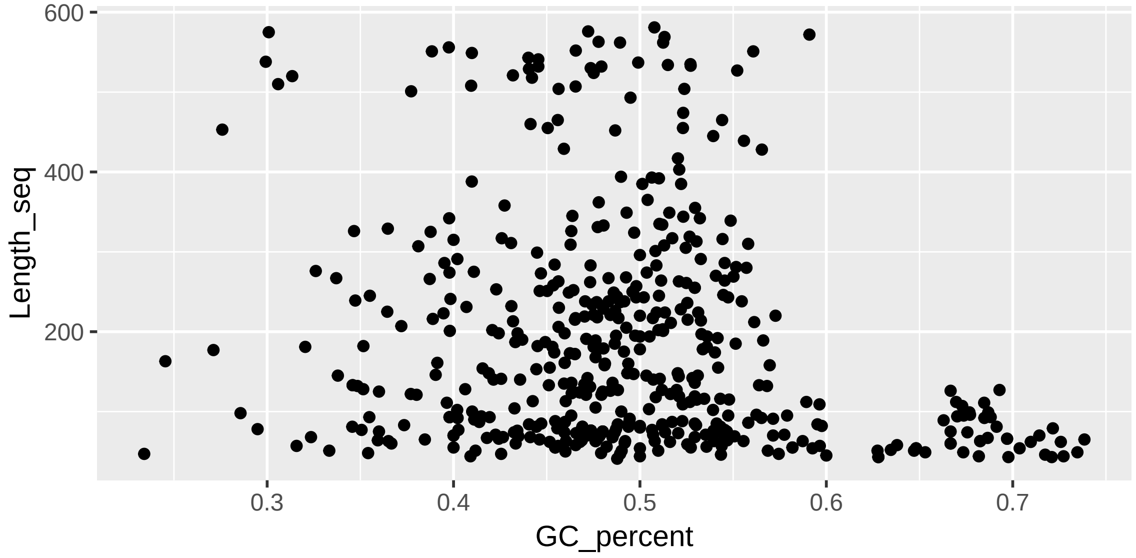


Compute GC% and do simple plots

```
# 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")
```

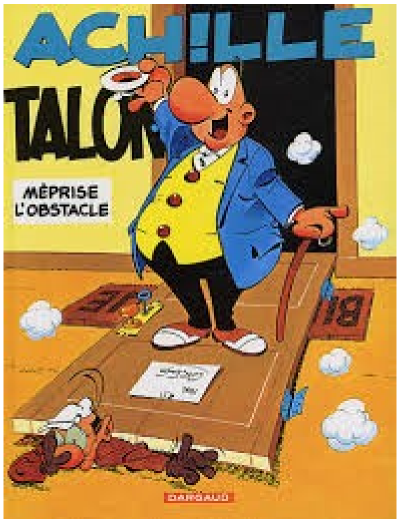


```
# Plot GC % vs Length of sequence
qplot(GC_percent, Length_seq)
```



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...