Introducing R and the RStudio IDE

Alana Schick, IMC Bioinformatics, University of Calgary

a.schick@ucalgary.ca, @alana_schick

UNIVERSITY OF CALGARY
CUMMING SCHOOL OF MEDICINE

UNIVERSITY OF CALGARY

International **Microbiome Centre**

What is R?

- R (since 1995) is a programming language developed to teach statistics
- R is open source (ie. free), widely used, flexible, and powerful

Packages: the power of R

A way for the R **community** to share functions and data sets

Importing & Exporting data
From
text files, excel, stata, SPSS , and databases

Data Modeling
Statistical tests
Linear & non-linear models
Machine learning
Survival analysis

Data Sharing
Plotting, interactive plots, reporting with markdown and shinny apps

https://support.rstudio.com/hc/en-us/articles/201057987-Quick-list-of-useful-R-packages

What is RStudio?

RStudio is an Integrated Development Environment (IDE) that allows users to run R in a more user-friendly way

R: Engine + RStudio: Dashboard

http://moderndive.com/

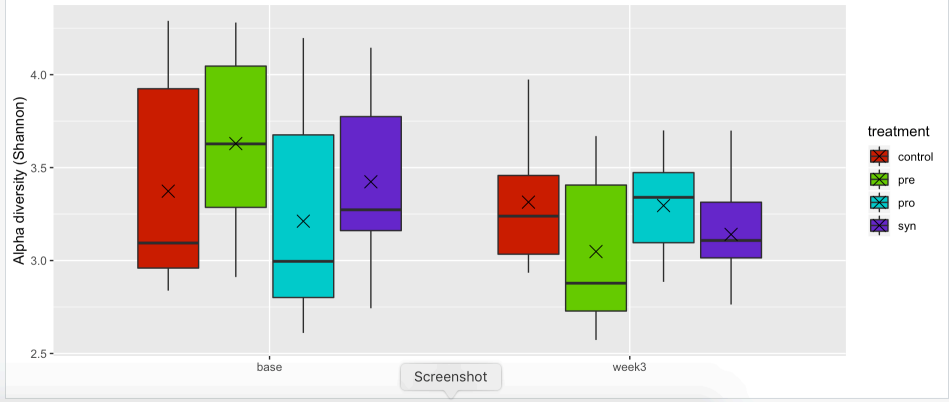
Let’s open Rstudio and get to know it !!

RStudio looks like this

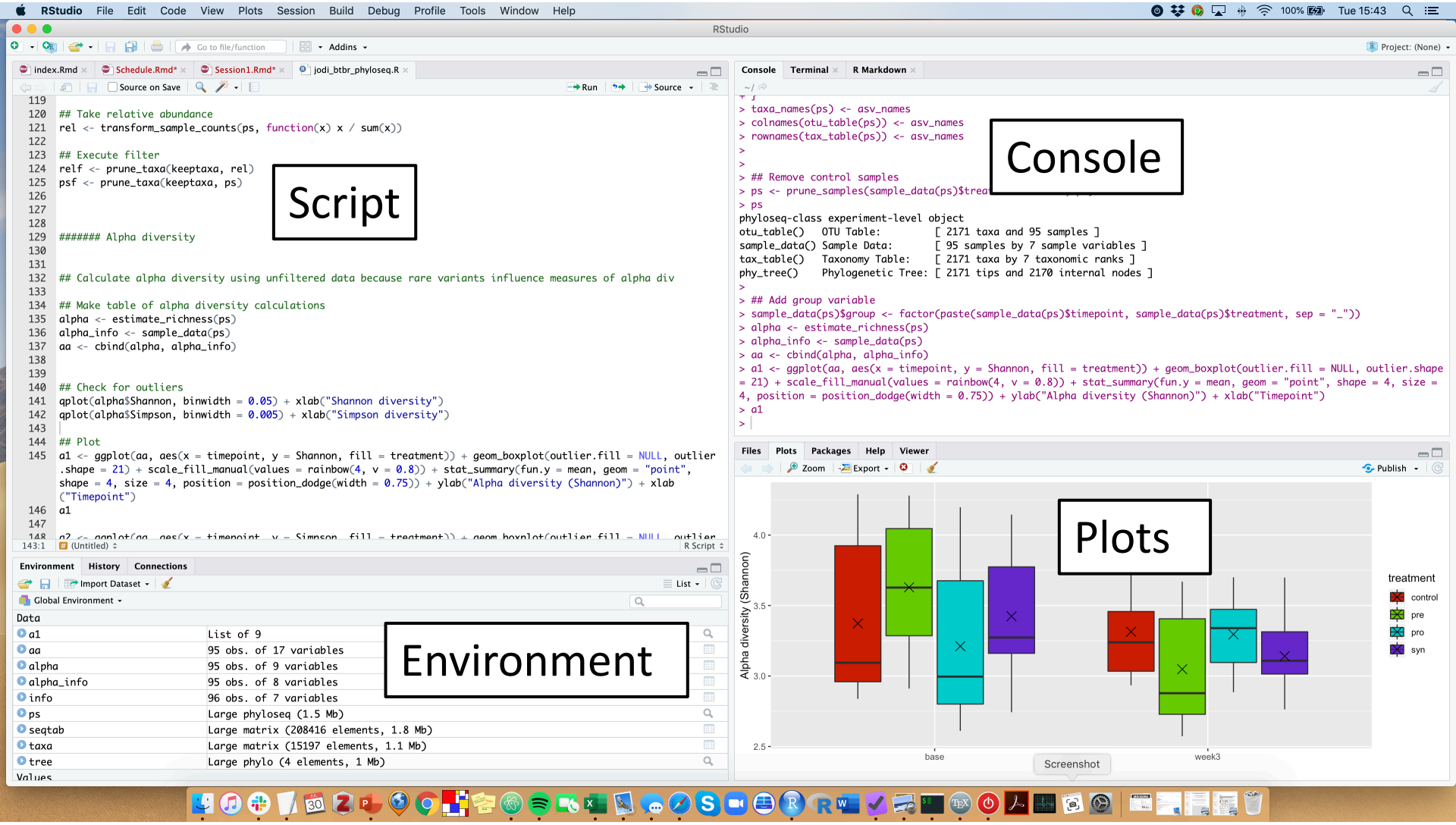
```
## Take relative abundance
rel <- transform_sample_counts(ps, function(x) x / sum(x))

## Execute filter
relf <- prune_taxa(keeptaxa, rel)
psf <- prune_taxa(keeptaxa, ps)



####### Alpha diversity


## Calculate alpha diversity using unfiltered data because rare variants influence measures of alpha div

## Make table of alpha diversity calculations
alpha <- estimate_richness(ps)
alpha_info <- sample_data(ps)
aa <- cbind(alpha, alpha_info)


## Check for outliers
qplot(alpha$Shannon, binwidth = 0.05) + xlab("Shannon diversity")
qplot(alpha$Simpson, binwidth = 0.005) + xlab("Simpson diversity")

## Plot
a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Shannon)") + xlab("Timepoint")
a1
```

```
> taxa_names(ps) <- asv_names
> colnames(otu_table(ps)) <- asv_names
> rownames(tax_table(ps)) <- asv_names
>
>
> ## Remove control samples
> ps <- prune_samples(sample_data(ps)$treatment != "NA", ps)
> ps
phyloseq-class experiment-level object
otu_table()   OTU Table:         [ 2171 taxa and 95 samples ]
sample_data() Sample Data:       [ 95 samples by 7 sample variables ]
tax_table()   Taxonomy Table:    [ 2171 taxa by 7 taxonomic ranks ]
phy_tree()    Phylogenetic Tree: [ 2171 tips and 2170 internal nodes ]
>
> ## Add group variable
> sample_data(ps)$group <- factor(paste(sample_data(ps)$timepoint, sample_data(ps)$treatment, sep = "_"))
> alpha <- estimate_richness(ps)
> alpha_info <- sample_data(ps)
> aa <- cbind(alpha, alpha_info)
> a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Shannon)") + xlab("Timepoint")
> a1
>
```

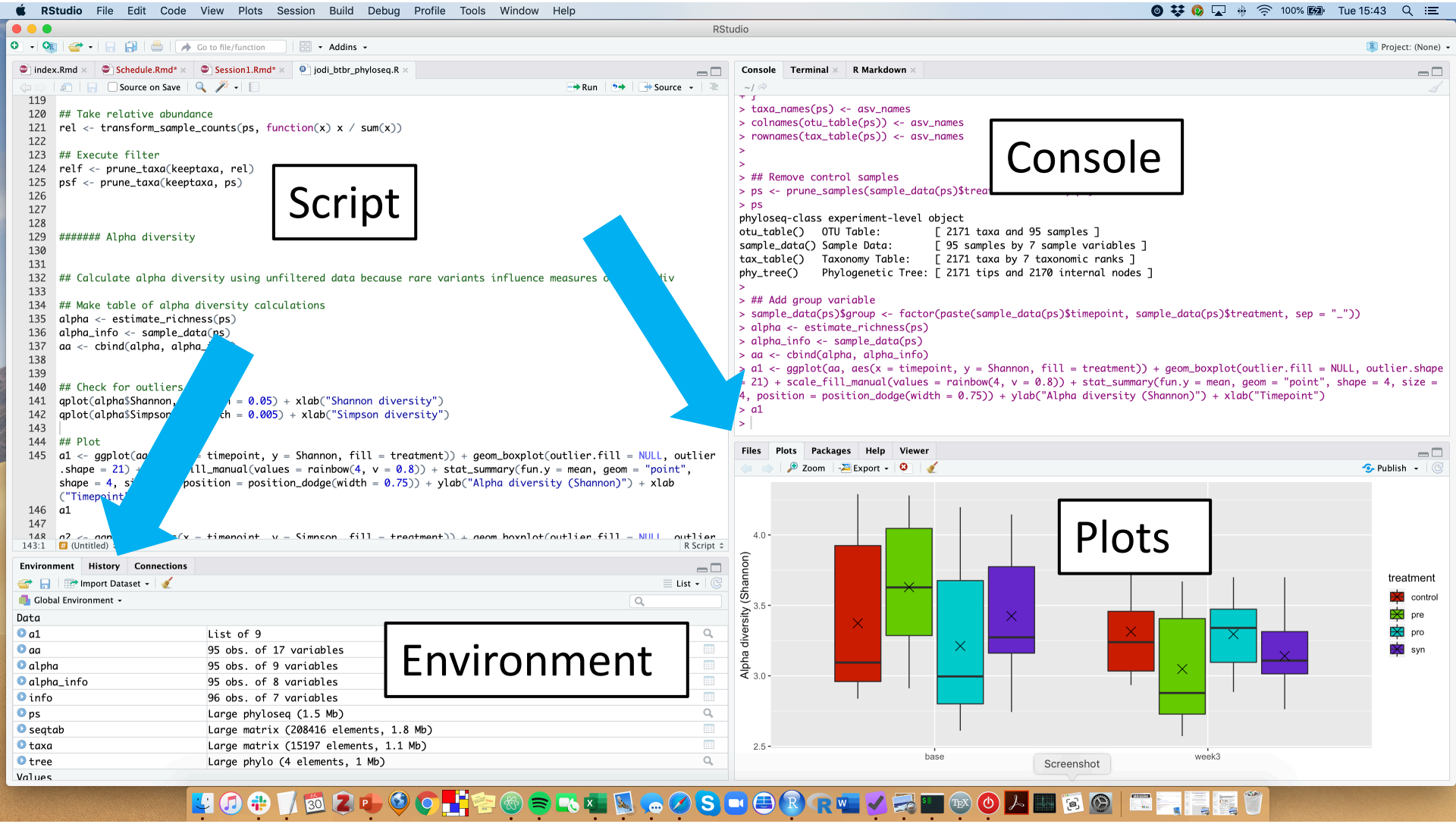
Data	
a1	List of 9
aa	95 obs. of 17 variables
alpha	95 obs. of 9 variables
alpha_info	95 obs. of 8 variables
info	96 obs. of 7 variables
ps	Large phyloseq (1.5 Mb)
seqtab	Large matrix (208416 elements, 1.8 Mb)
taxa	Large matrix (15197 elements, 1.1 Mb)
tree	Large phylo (4 elements, 1 Mb)



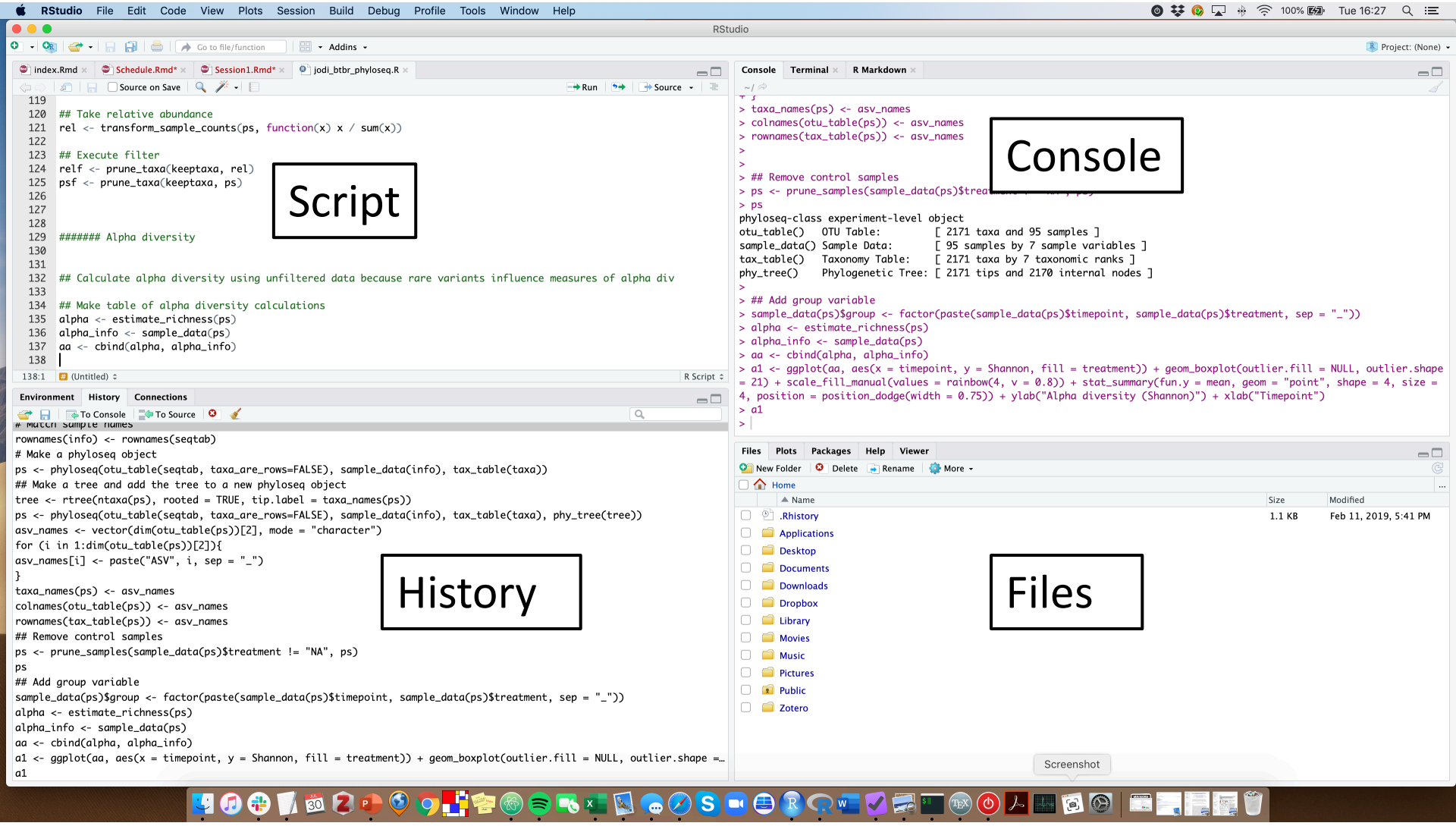
RStudio screen



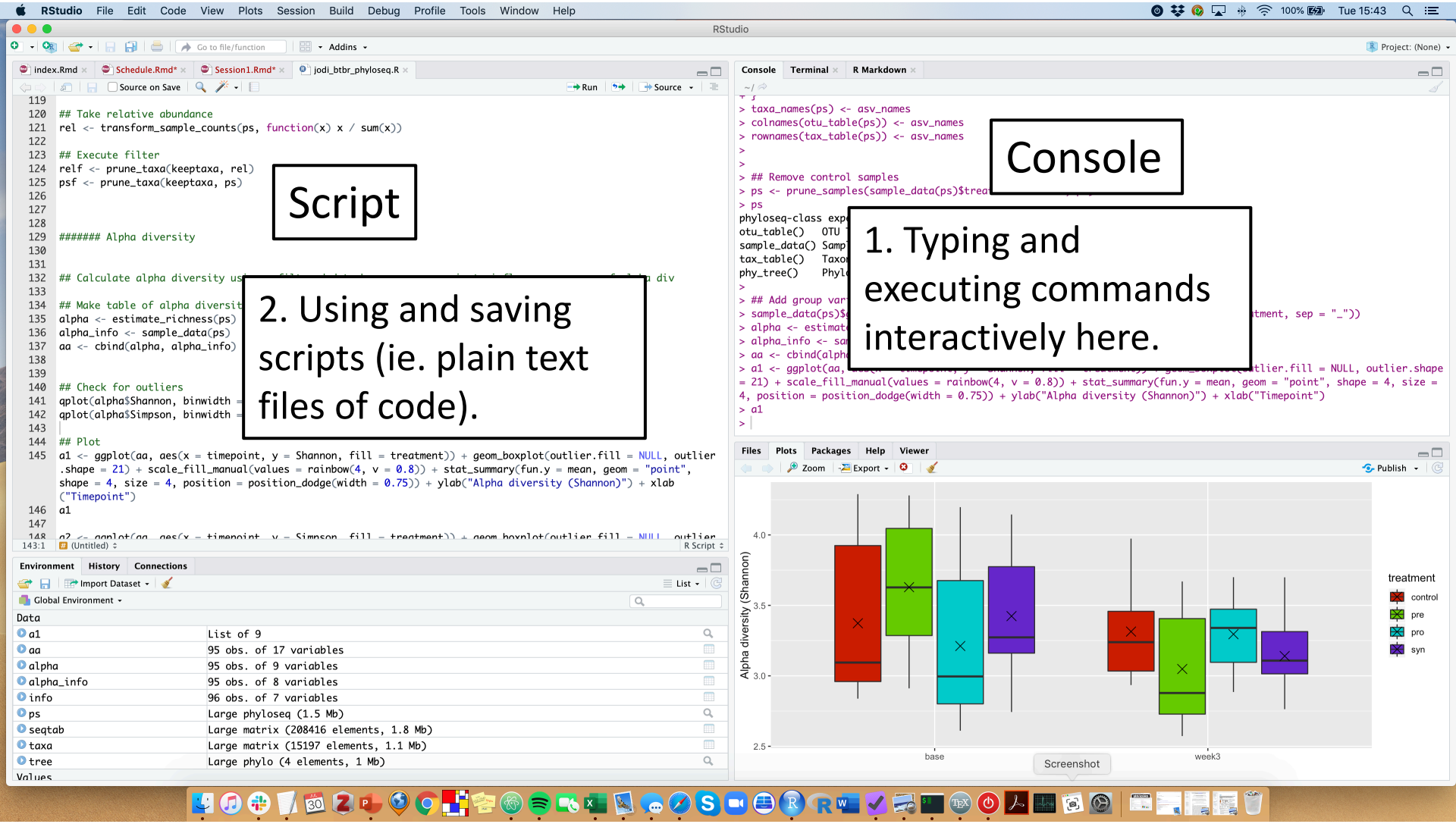
RStudio screen



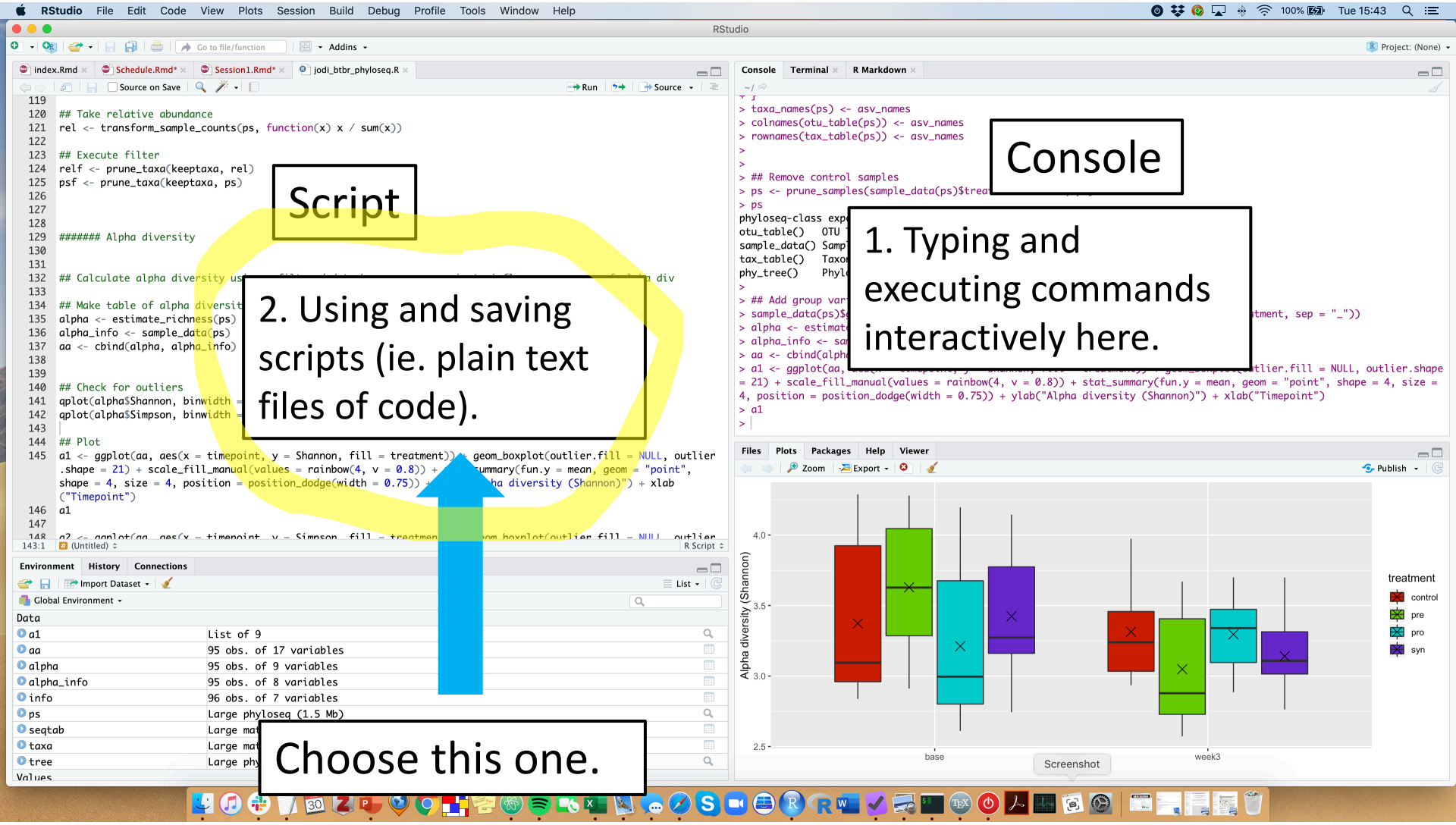
RStudio screen



How to R – 2 ways



How to R – 2 ways



Scripts

```
# jodi_btbr project, Alana Schick, April 2019
# This is a script to analyze the output tables of the DADA2 workflow in phyloseq
# Have two output files from dada2 - a sequence table and a taxonomy table, read them into R using the readRDS()
function
# The formatted sample metadata is in a table called "jodi_btbr_metadata.txt"

library(phyloseq)
#packageVersion("phyloseq")
library(ggplot2)
#packageVersion("ggplot2")
library(ape)
library(viridis)
library(grid)
library(gridExtra)
library(reshape2)
library(DESeq2)
library(fields)
library(vegan)
library(ggpubr)
library(plyr)
library(RColorBrewer)

path_to_project <- "/Users/alanaschick/Drop

# Read in files
seqtab <- readRDS(file.path(path_to_projec
taxa <- readRDS(file.path(path_to_project,
info <- read.table(file.path(path_to_projec

# Match sample names
rownames(info) <- rownames(seqtab)

# Make a phyloseq object
```

38:1 (Untitled) R Script

Everything in the console will be forgotten when you close the session.

Scripts are saved, keeping a complete record of the commands you ran so you can run them again (ie. completely reproducible).

Can execute parts of this or the entire script.

Scripts - commenting

```r
# jodi_btbr project, Alana Schick, April 2019
# This is a script to analyze the output tables of the DADA2 workflow in phyloseq
# Have two output files from dada2 - a sequence table and a taxonomy table, read them into R using the readRDS()
function
# The formatted sample metadata is in a table called "jodi_btbr_metadata.txt"

library(phyloseq)
#packageVersion("phyloseq")
library(ggplot2)
#packageVersion("ggplot2")
library(ape)
library(viridis)
library(grid)
library(gridExtra)
library(reshape2)
library(DESeq2)
library(fields)
library(vegan)
library(ggpubr)
library(plyr)
library(RColorBrewer)

path_to_project <- "/Users/alanaschick/Dropbox/lmc/projects/jodi_btbr/"

# Read in files
seqtab <- readRDS(file.path(path_to_project, "results/seqtab_final.rds"))
taxa <- readRDS(file.path(path_to_project, "results/taxa_final.rds"))
info <- read.table(file.path(path_to_project, "jodi_btbr_metadata2.txt"), header = TRUE)

# Match sample names
rownames(info) <- rownames(seqtab)

# Make a phyloseq object
```

Comment out lines of your scripts by using the # symbol. R will not run these.

Be descriptive. You will not remember what you did a year later.

Packages

```
# jodi_btbr project, Alana Schick, April 2019
# This is a script to analyze the output tables of the DADA2 workflow in phyloseq
# Have two output files from dada2 - a sequence table and a taxonomy table, read them into R using the readRDS() function
# The formatted sample metadata is in a table called "jodi_btbr_metadata.txt"

library(phyloseq)
#packageVersion("phyloseq")
library(ggplot2)
#packageVersion("ggplot2")
library(ape)
library(viridis)
library(grid)
library(gridExtra)
library(reshape2)
library(DESeq2)
library(fields)
library(vegan)
library(ggpubr)
library(plyr)
library(RColorBrewer)

path_to_project <- "/Users/alanaschick/

# Read in files
seqtab <- readRDS(file.path(path_to_project, "results/seqtab_final.rds"))
taxa <- readRDS(file.path(path_to_project, "results/taxa_final.rds"))
info <- read.table(file.path(path_to_project, "jodi_btbr_metadata2.txt"), header = TRUE)

# Match sample names
rownames(info) <- rownames(seqtab)

# Make a phyloseq object
```

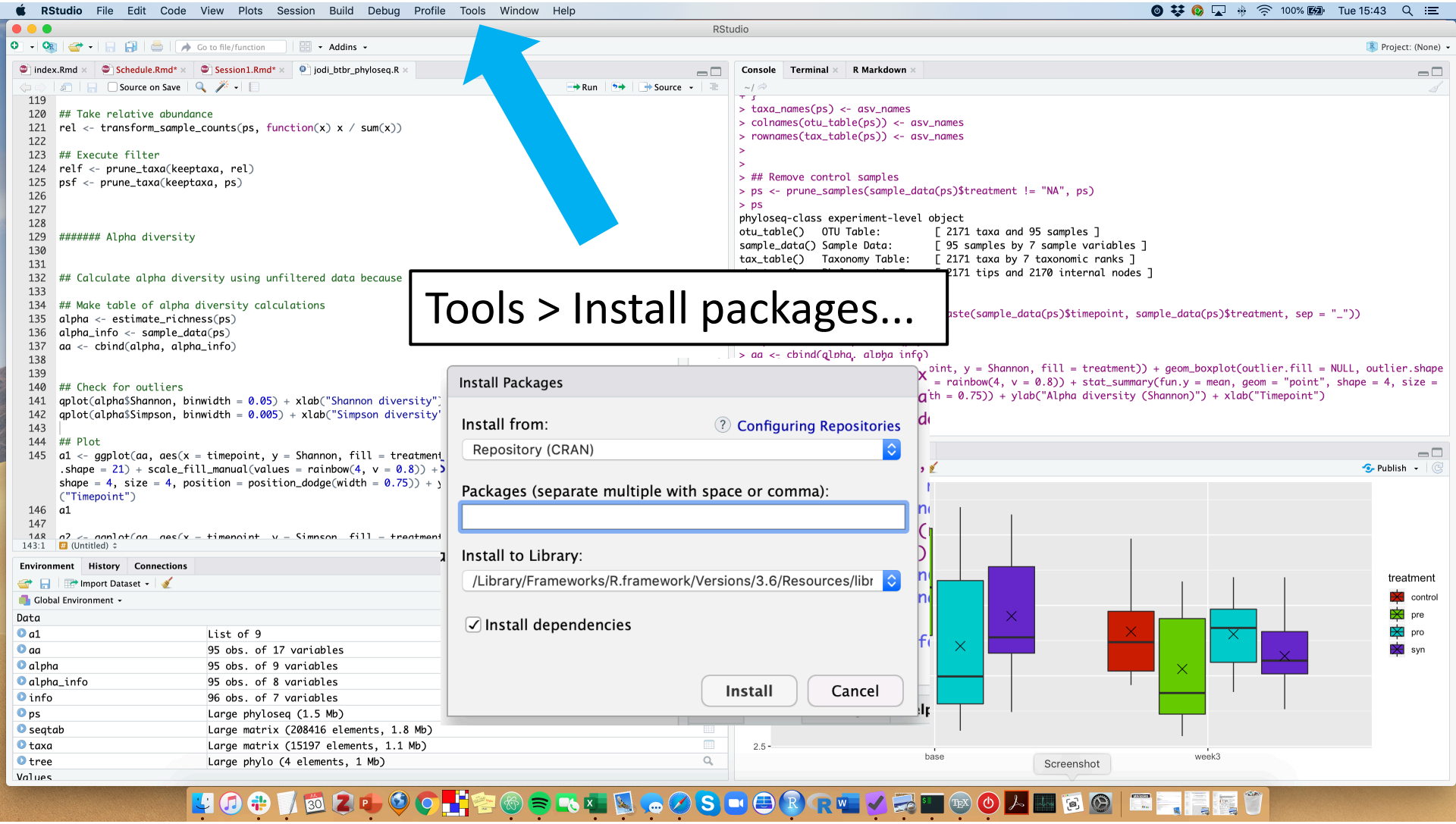
38:1 # (Untitled) R Script

Packages are collections of R functions developed for a specific task.

Packages need to first be installed on your computer.

After installed, library() is the command used to load a package.

Packages


Tools > Install packages...
Install Packages
Install from:
Configuring Repositories
Repository (CRAN)
Packages (separate multiple with space or comma):
Install to Library:
/Library/Frameworks/R.framework/Versions/3.6/Resources/libr
Install dependencies
Install
Cancel
Screenshot

Pay close attention to the next few slides

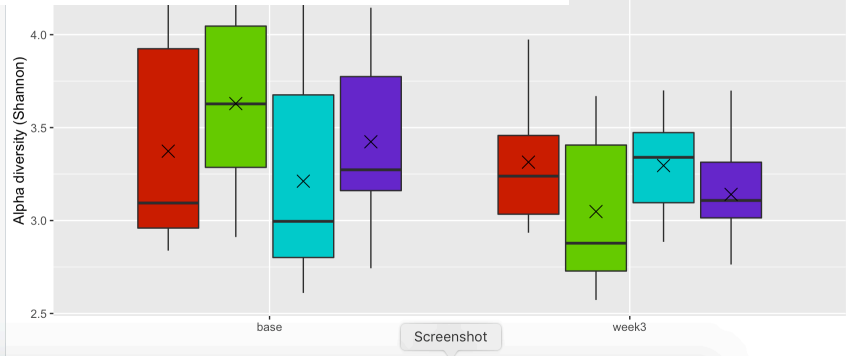
About half of the students in every workshop have problems in understanding the concept of working directory!!

Working directory

Every time you open RStudio, it goes to a default directory, usually your home directory.

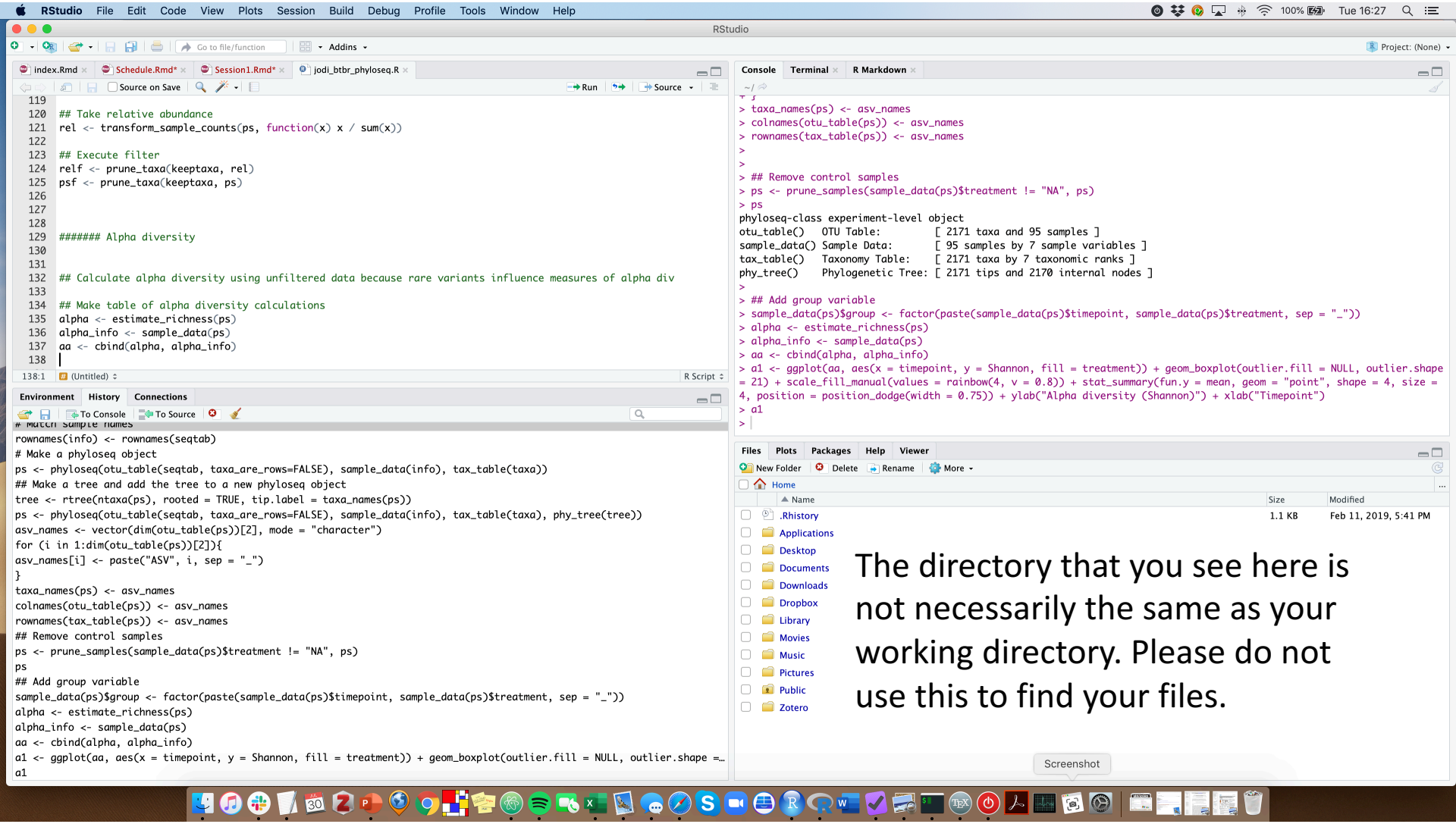
You can use the command **setwd()** to change the working directory.

```
setwd("home/aschick/projects/workshop")
```

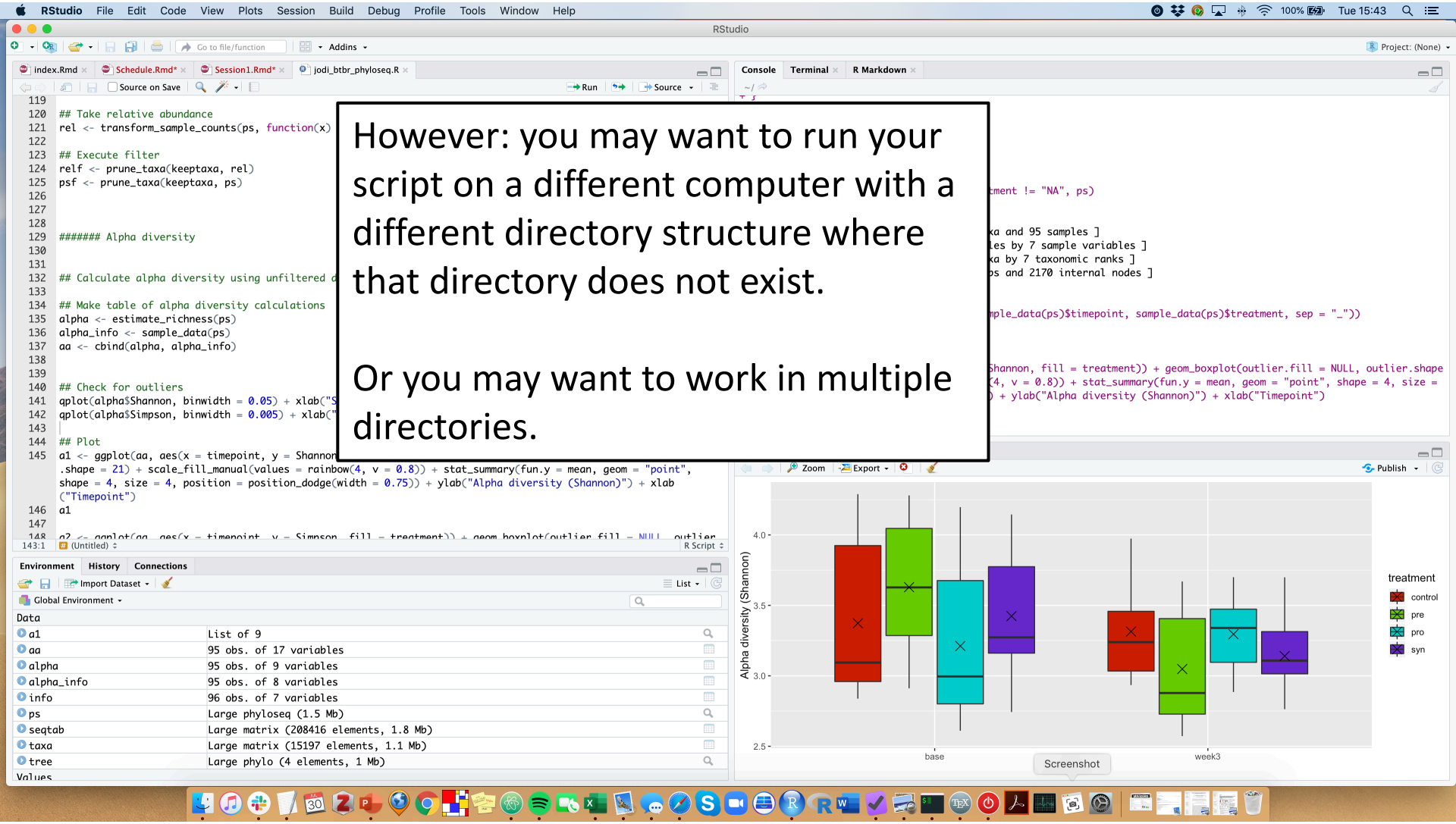


Screenshot

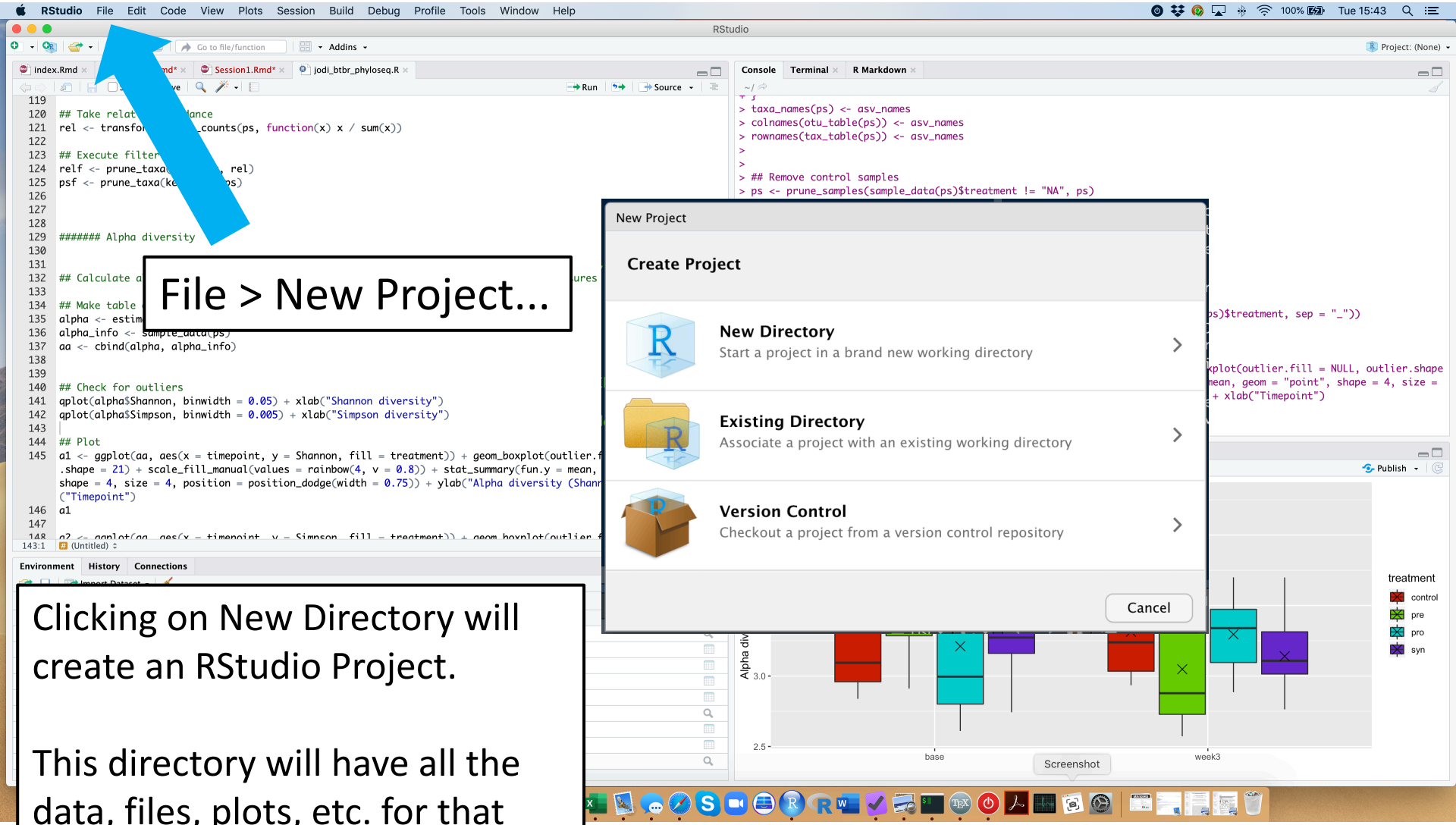
Relative paths


RStudio
index.Rmd
Schedule.Rmd*
Session1.Rmd*
jodi_btbr_phyloseq.R
Source on Save
Run
Source
Take relative abundance
rel <- transform_sample_counts(ps, function(x) x / sum(x))
Execute filter
relf <- prune_taxa(keeptaxa, rel)
psf <- prune_taxa(keeptaxa, ps)
####### Alpha diversity
Calculate alpha diversity using unfiltered data because rare variants influence measures of alpha div
Make table of alpha diversity calculations
alpha <- estimate_richness(ps)
alpha_info <- sample_data(ps)
aa <- cbind(alpha, alpha_info)
Console
Terminal
R Markdown
> taxa_names(ps) <- asv_names
> colnames(otu_table(ps)) <- asv_names
> rownames(tax_table(ps)) <- asv_names
> ## Remove control samples
> ps <- prune_samples(sample_data(ps)$treatment != "NA", ps)
> ps
phyloseq-class experiment-level object
otu_table() OTU Table: [2171 taxa and 95 samples]
sample_data() Sample Data: [95 samples by 7 sample variables]
tax_table() Taxonomy Table: [2171 taxa by 7 taxonomic ranks]
phy_tree() Phylogenetic Tree: [2171 tips and 2170 internal nodes]
> ## Add group variable
> sample_data(ps)$group <- factor(paste(sample_data(ps)$timepoint, sample_data(ps)$treatment, sep = "_"))
> alpha <- estimate_richness(ps)
> alpha_info <- sample_data(ps)
> aa <- cbind(alpha, alpha_info)
> a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape = 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size = 4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Shannon)") + xlab("Timepoint")
> a1
Environment
History
Connections
To Console
To Source
rownames(info) <- rownames(seqtab)
# Make a phyloseq object
ps <- phyloseq(otu_table(seqtab, taxa_are_rows=FALSE), sample_data(info), tax_table(taxa))
Make a tree and add the tree to a new phyloseq object
tree <- rtree(ntaxa(ps), rooted = TRUE, tip.label = taxa_names(ps))
ps <- phyloseq(otu_table(seqtab, taxa_are_rows=FALSE), sample_data(info), tax_table(taxa), phy_tree(tree))
asv_names <- vector(dim(otu_table(ps))[2], mode = "character")
for (i in 1:dim(otu_table(ps))[2]){
asv_names[i] <- paste("ASV", i, sep = "_")
}
taxa_names(ps) <- asv_names
colnames(otu_table(ps)) <- asv_names
rownames(tax_table(ps)) <- asv_names
Remove control samples
ps <- prune_samples(sample_data(ps)$treatment != "NA", ps)
ps
Add group variable
sample_data(ps)$group <- factor(paste(sample_data(ps)$timepoint, sample_data(ps)$treatment, sep = "_"))
alpha <- estimate_richness(ps)
alpha_info <- sample_data(ps)
aa <- cbind(alpha, alpha_info)
a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape =…
a1
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
Name
Size
Modified
.Rhistory
1.1 KB
Feb 11, 2019, 5:41 PM
Applications
Desktop
Documents
Downloads
Dropbox
Library
Movies
Music
Pictures
Public
Zotero
The directory that you see here is not necessarily the same as your working directory. Please do not use this to find your files.
Screenshot

Working directory



RStudio Project



Error messages

Console | Terminal | R Markdown

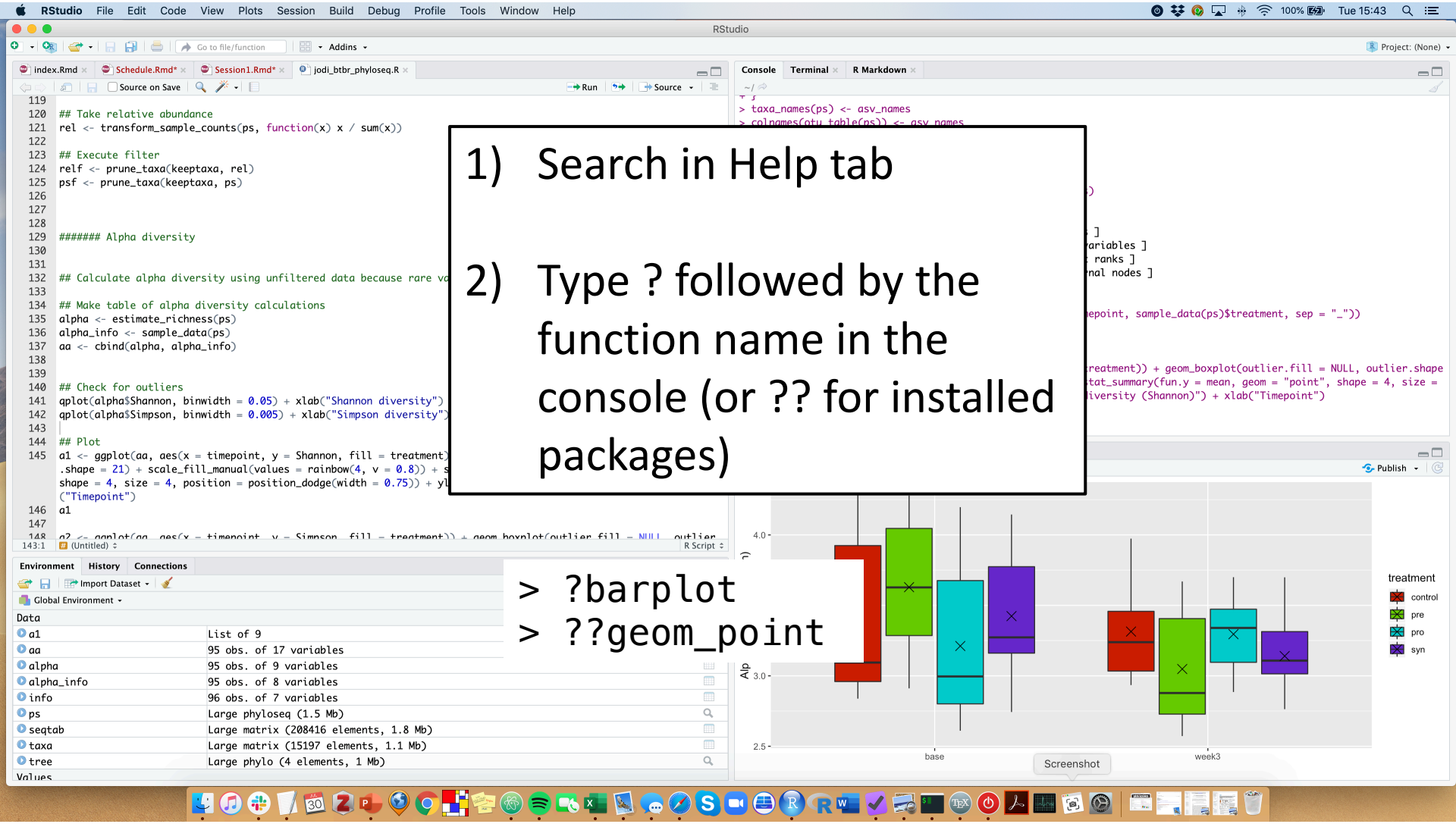
```
>
> ## Remove control samples
> ps <- prune_samples(sample_data(ps)$treatment != "NA", ps)
> ps
phyloseq-class experiment-level object
otu_table()   OTU Table:         [ 2171 taxa and 95 samples ]
sample_data() Sample Data:       [ 95 samples by 7 sample variables ]
tax_table()   Taxonomy Table:    [ 2171 taxa by 7 taxonomic ranks ]
phy_tree()    Phylogenetic Tree: [ 2171 tips and 2170 internal nodes ]
>
> ## Add group variable
> sample_data(ps)$group <- factor(paste(sample_data(ps)$timepoint, sample_data(ps)$treatment, sep = "_"))
> alpha <- estimate_richness(ps)
> alpha_info <- sample_data(ps)
> aa <- cbind(alpha, alpha_info)
> a1 <- ggplot(aa, aes(x = timepoint, y = Shannon, fill = treatment)) + geom_boxplot(outlier.fill = NULL, outlier.shape
= 21) + scale_fill_manual(values = rainbow(4, v = 0.8)) + stat_summary(fun.y = mean, geom = "point", shape = 4, size =
4, position = position_dodge(width = 0.75)) + ylab("Alpha diversity (Shannon)") + xlab("Timepoint")
> a1
> ord1 <- ordinate(relf, method = "NMDS", distance = "bray")
Error in ordinate(relf, method = "NMDS", distance = "bray") :
  object 'relf' not found
> b1 <- plot_ordination(relf, ord1, color = "timepoint", shape = "treatment", title = "NMDS - Bray") + scale_colour_man
ual(values = viridis(3))
Error in plot_ordination(relf, ord1, color = "timepoint", shape = "treatment",  :
  object 'relf' not found
> b1
Error: object 'b1' not found
>
```

Error messages

Console | Terminal | R Markdown

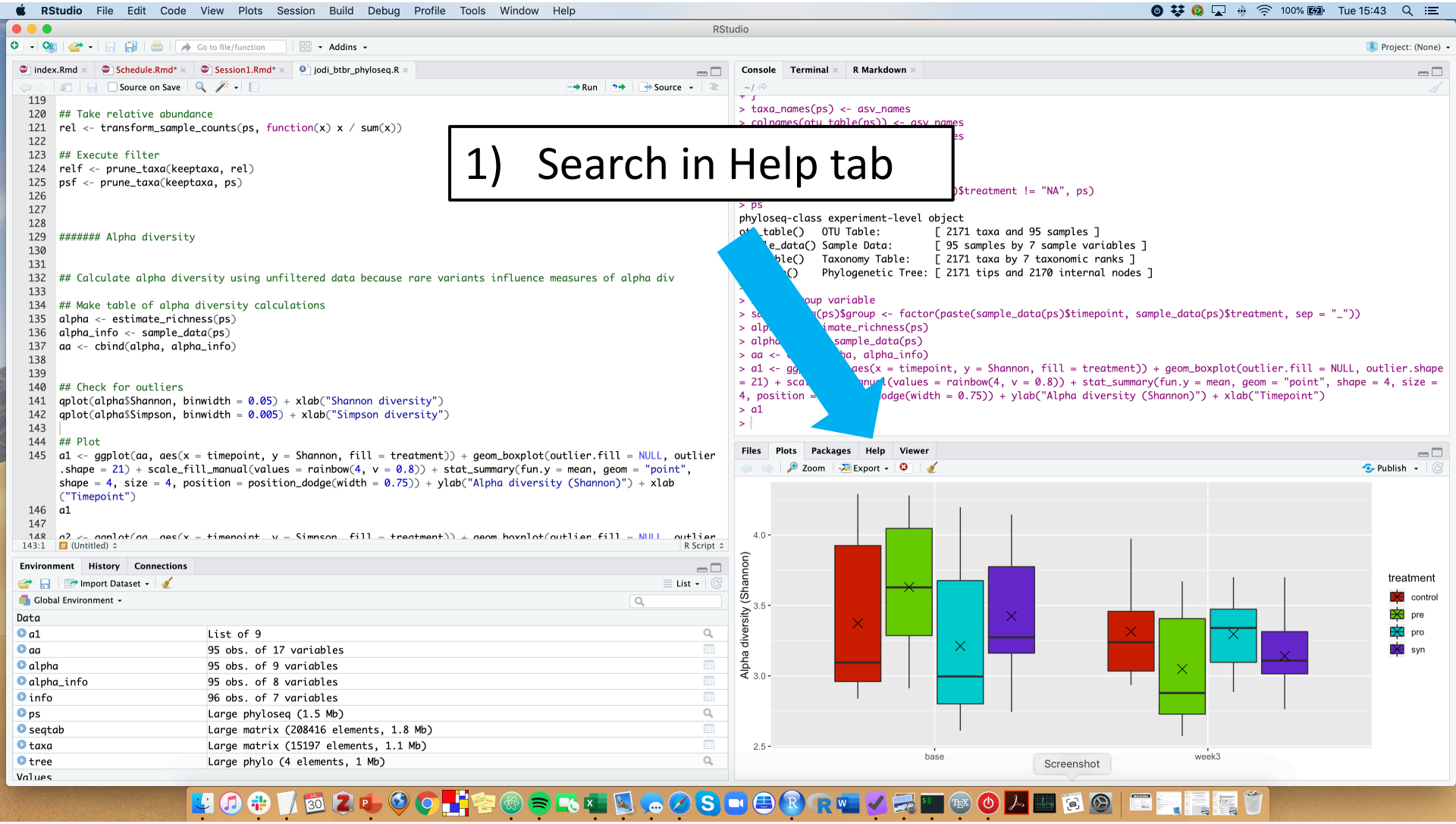
```
>
> ## Remove control samples
> ps <- prune_samples(sample_data(ps)$tr
> ps
phyloseq-class experiment-level object
otu_table()   OTU Table:         [ 2171
sample_data() Sample Data:       [ 95 sa
tax_table()   Taxonomy Table:    [ 2171
phy_tree()    Phylogenetic Tree: [ 2171
>
> ## Add group variable
> sample_data(ps)$group <- factor(paste(                    tment, sep = "_"))
> alpha <- estimate_richness(ps)
> alpha_info <- sample_data(ps)
> aa <- cbind(alpha, alpha_info)
> a1 <- ggplot(aa, aes(x = timepoint, y                     tlier.fill = NULL, outlier.shape
= 21) + scale_fill_manual(values = rainb                    om = "point", shape = 4, size =
4, position = position_dodge(width = 0.7                    "Timepoint")
> a1
> ord1 <- ordinate(relf, method = "NMDS"
Error in ordinate(relf, method = "NMDS",
  object 'relf' not found
> b1 <- plot_ordination(relf, ord1, colo                    NMDS - Bray") + scale_colour_man
ual(values = viridis(3))
Error in plot_ordination(relf, ord1, col
  object 'relf' not found
> b1
Error: object 'b1' not found
>
```

Getting help



Getting help

1) Search in Help tab



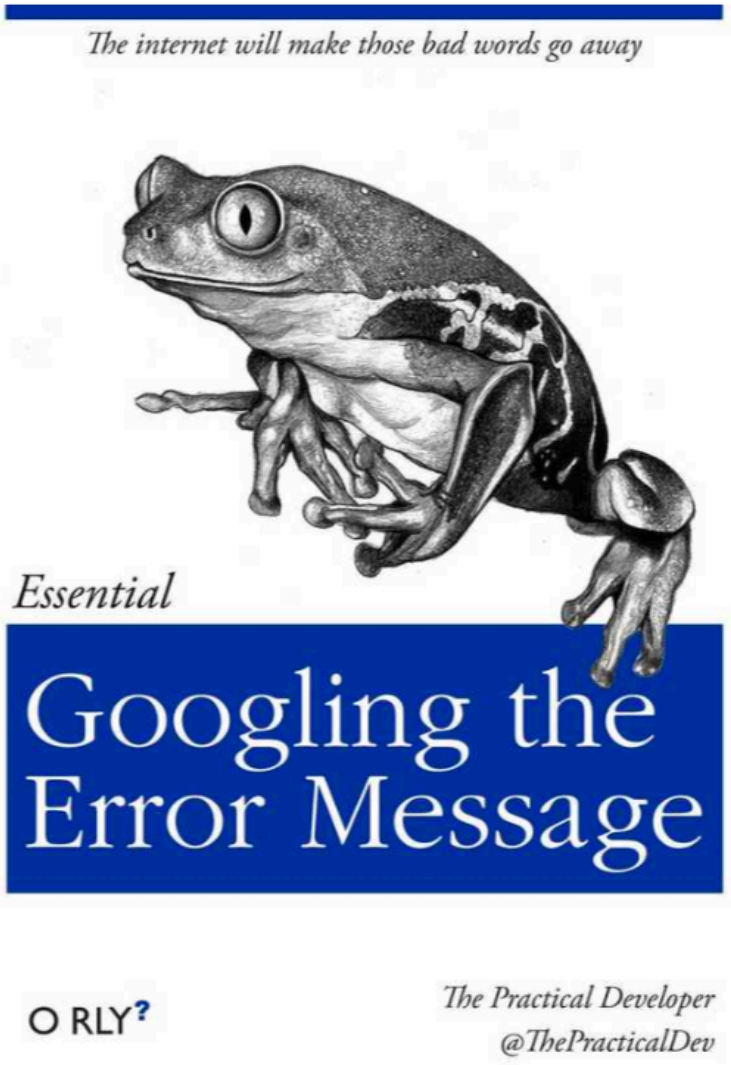
Getting help

1) Search in Help tab

2) Type ? followed by the function name in the console (or ?? for installed packages)

3) Google the error message

See website for tips and resources!

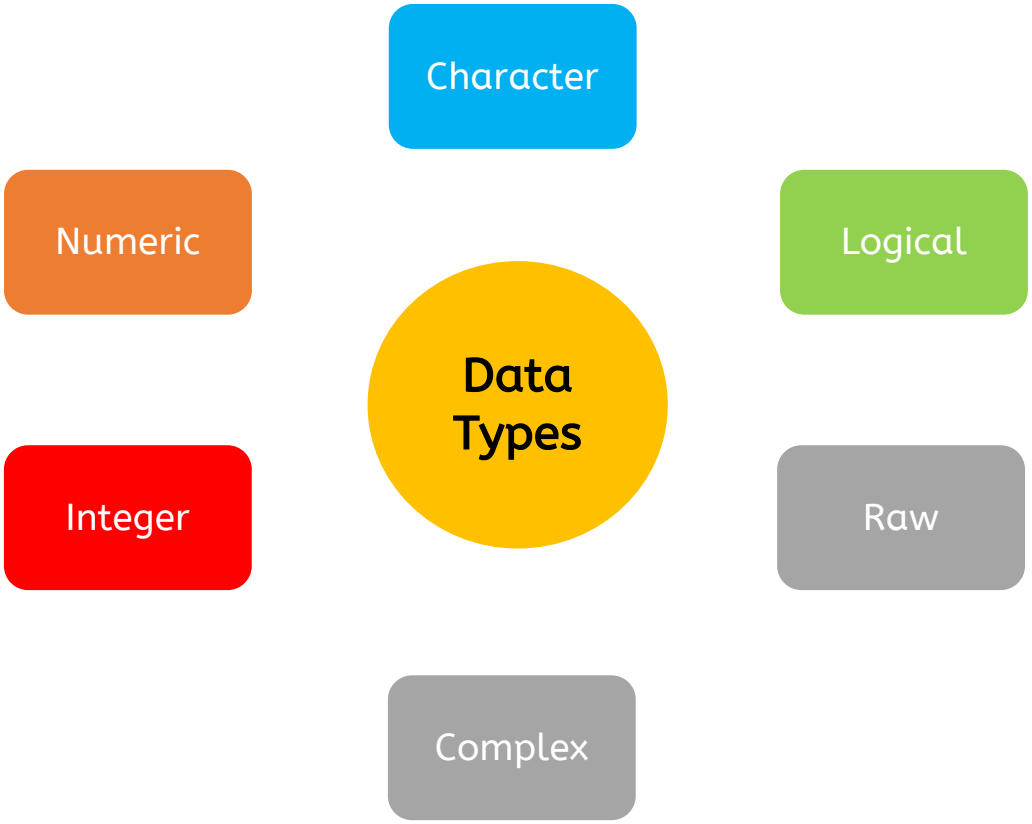


Summary and best practices

- Always save your code in R scripts
- Load packages using library() at the top of your script
- Write clear, readable code with comments*
- Be mindful of your working directory or location of files
- Use RStudio projects to organize scripts, data, and output

*See http://adv-r.had.co.nz/Style.html for tips.

Data Types



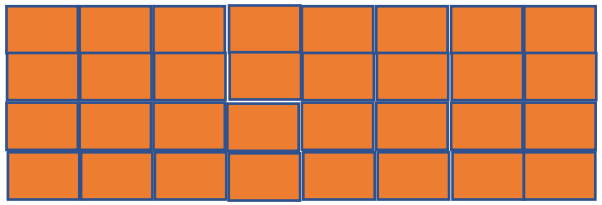
Data Structures

Vector

1d, homogeneous

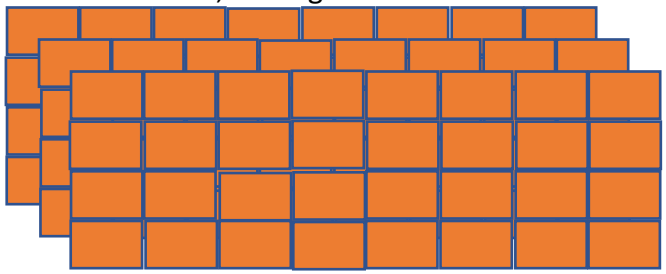
Matrix

2d, homogeneous



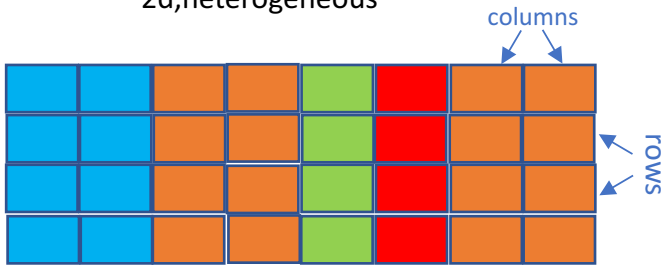
Array

3d, homogeneous



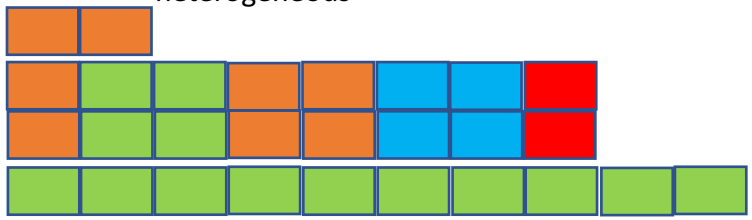
Dataframe & tibble

2d,heterogeneous



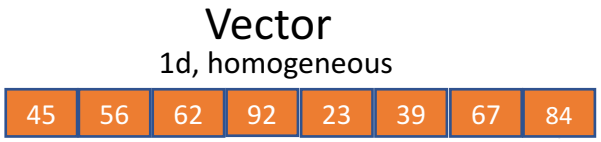
Lists

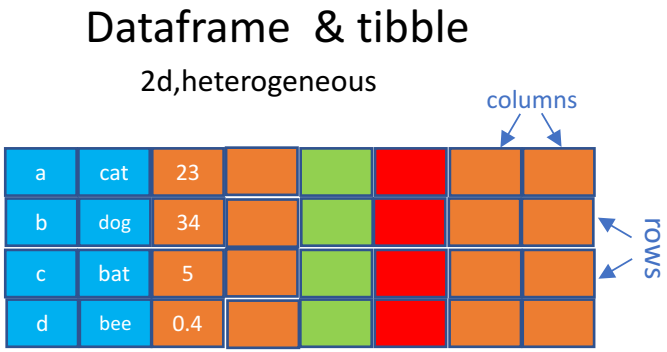
heterogeneous



Homogenous means that it can hold only one data type at a time.
Heterogeneous means it can hold multiple datatypes at a time.

Data Structures





Data Structures