MB590: Microbiome Analysis Week 1 Introduction

Prof: Christine Hawkes

TA: Rachel Hammer

Today's Overview

- Introductions
- Semester schedule
- Assignments
- Grading
- See syllabus for all other class policies

- Some R basics
- GitHub
- R markdown

Semester schedule

Week	Date	Topic
1	12-Jan	Introduction – R, GitHub, Rmarkdown
2	19-Jan	Sequence prep, 16S ASV pipeline
3	26-Jan	Identification, normalization
4	2-Feb	Practicum – ITS ASV pipeline
5	9-Feb	Exploratory analysis 1 – alpha diversity
6	16-Feb	Exploratory analysis 2 – beta diversity
7	23-Feb	dada2 on the HPC
8	2-Mar	Exploratory analysis 3 – core microbiomes
9	9-Mar	Practicum – full exploratory analysis
10	16-Mar	Spring break – no class
11	23-Mar	Hypothesis testing 1 – regression
12	30-Mar	Hypothesis testing 2 – permutation tests
13	6-Apr	Hypothesis testing 3 – TBD
14	13-Apr	Practicum – full hypothesis testing
15	20-Apr	Final project presentations

Assignments

- Weekly coding exercises
- Final project

- Readings (optional)

Assignments: weekly coding

- Each week, you'll have coding exercises that begin in class and can be finished as homework
- These should be created as .Rmd files and knitted to html for submission
- Knitted files must be pushed to your MicrobiomeAnalysis2022 GitHub repo (info to follow) with the name:
  - Wk#_YourLastName
  - Replace # with week number
- Coding exercises this week will not be handed in – the exercises are for your benefit

Assignments: final project

- See MB590_FinalProjectGuide.pdf in GitHub
- You will re-analyze data in a published paper
  - Must have publicly available data
  - Cannot have published R code
    - Exception to this is if you plan to analyze data entirely differently (e.g., OTU vs. ASV approach)
  - Either repeat or revise the published analysis and compare
  - Feedback is available on which option to take – just ask!

Assignments: final project

- See MB590_FinalProjectGuide.pdf in GitHub
- Submit your proposal by Feb 9 on GitHub
- Include
  - Original paper pdf
  - Proposal that includes
    - Why paper is a good choice
    - Plan for re-analysis and how that differs from original
    - Confirmation that metadata and sequences are available to support the proposed analysis
    - Confirmation that R code are not available (or that your analysis is sufficiently different that it does not matter)

Assignments: final project

- Final projects include the following:
  - 15-min presentation to the class
  - A well-annotated R markdown document that shows
    - accession numbers and links for any public data you used
    - your re-analysis plan and reasoning
    - your re-analysis
    - interpretation of the new results in the context of the published results, including discussion of whether/how and why your results did/did not differ from the published results
  - The original PDF and any supplemental materials

Assignments: final project deadlines

Item	Due Date	Points
Final project proposal	Feb 9	25
Data successfully downloaded from SRA	Feb 23	5
Optional use for in-class practicums	Mar 9, Apr 6	--
Final presentation	Apr 20	40
Final Rmd file	Apr 27	50

Assignments: final project resources

- SRA
  - https://www.ncbi.nlm.nih.gov/sra

- NCSU HPC
  - https://projects.ncsu.edu/hpc/Guide/
  - https://projects.ncsu.edu/hpc/Documents/UserTraining.php#course

- NCSU Bioinformatics Users Group (BUG)
  - https://ncsu-debug.readthedocs.io/en/latest/

Assignments: weekly readings are optional

List and papers are available on GitHub/MicrobiomeAnalysis2022/ClassDocs

Week	Date	Topic	Readings
1	12-Jan	Introduction – R, GitHub, Rmarkdown	
2	19-Jan	Sequence prep, 16S ASV pipeline	Callahan et al. 2016 Nat Meth
3	26-Jan	Identification, normalization	McMurdie & Holmes 2014 PLoS CB Gloor et al. 2017 Fr Micro
4	2-Feb	Practicum – ITS ASV pipeline	Froslev etal. 2017
5	9-Feb	Exploratory analysis 1 – alpha diversity	Pauvert et al. 2019 Fun Eco
6	16-Feb	Exploratory analysis 2 – beta diversity	Anderson et al. 2011 Eco Lett
7	23-Feb	Dada2 on the HPC (Lisa Lowe)	
8	2-Mar	Exploratory analysis 3 – core microbiomes	Risley 2020 J Anim Ecol
9	9-Mar	Practicum – full exploratory analysis	
10	16-Mar	Spring break – no class	
11	23-Mar	Hypothesis testing 1 – regression	TBD
12	30-Mar	Hypothesis testing 2 – permutation tests	Collyer & Adams 2018 MEE
13	6-Apr	Hypothesis testing 3 – TBD	TBD
14	13-Apr	Practicum – full hypothesis testing	
15	20-Apr	Final project presentations	

Grading

Grade components

	Points	Contribution (%)
Participation	80	20
GitHub contributions	50	12.5
HPC training	30	7.5
Weekly coding assignments	120	30
Final project	120	30

This Course uses Standard NCSU Letter Grading, unless you've selected the S/U option
https://studentservices.ncsu.edu/your-resources/covid-19/spring2020-sat-grading/

97	≤	A+	≤	100
93	≤	A	<	97
90	≤	A-	<	93
87	≤	B+	<	90
83	≤	B	<	87
80	≤	B-	<	83
77	≤	C+	<	80
73	≤	C	<	77
70	≤	C-	<	73
67	≤	D+	<	70
63	≤	D	<	67
60	≤	D-	<	63
0	≤	F	<	60

More on the S/U option is provided here and in the syllabus:
https://studentservices.ncsu.edu/your-resources/covid-19/spring2020-sat-grading/

R Studio IDE

Top left - scripts

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

Wk1_Intro.Rmd

Knit | Run

```
### Other useful package commands

```{r eval=FALSE, echo=TRUE}
# Missing dependencies?
# CRAN
# install.packages("additional_missing_package_2")
# Bioconductor
# BiocManager::install("additional_missing_package_1", version = "3.12")
# example:
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("ShortRead")

# See what packages are installed - can also use Rstudio Packages tab
library()

# load a package and determine the software version
library(vegan); packageVersion("vegan")

# find the citation for package
citation("vegan")

# get help with packages
packageDescription("vegan")
help(package="vegan")
ls("package:vegan") # lists package functions
help(anosim, package="vegan") # help with specific function
browseVignettes(package="vegan") # locally or online

# delete packages
remove.packages("packagename")
```
```

Bottom left – console, terminal, R markdown

Console | Terminal | R Markdown | Jobs

R 4.1.2 · S:/PMB/HawkesLab/People/HawkesChristine/Teaching/MB590_MicrobiomeAnalysis/Spring 2022/Scripts & Presentations/

```
> # Install packages from GitHub repository
> # Lulu
> library(devtools)
Loading required package: usethis
> # See what packages are installed - can also use Rstudio Packages tab
> library()
> # load a package and determine the software version
> library(vegan); packageVersion("vegan")
Loading required package: permute

Attaching package: 'permute'

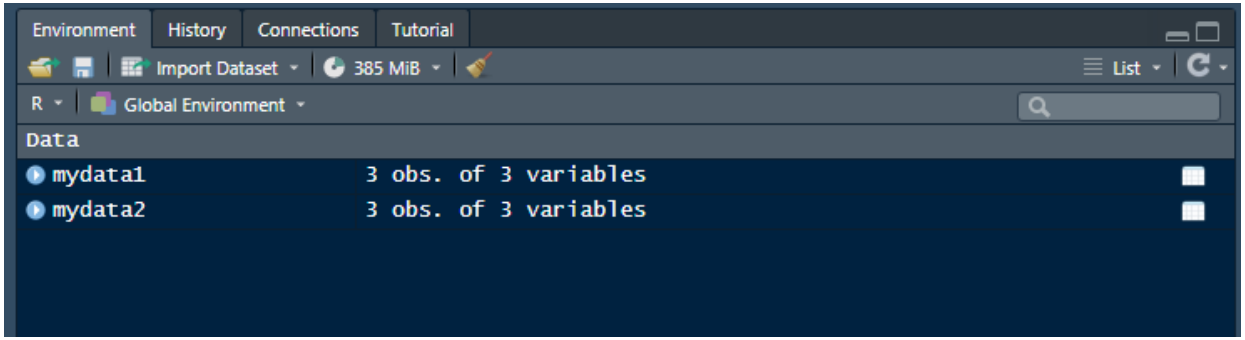
The following object is masked from 'package:devtools':

    check

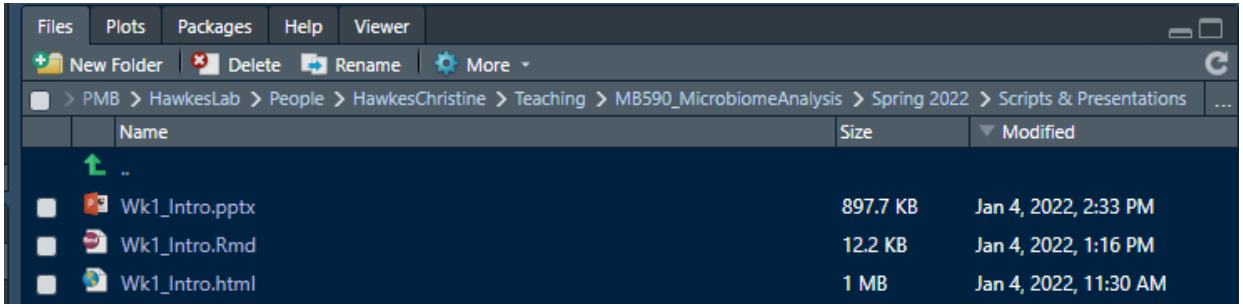
Loading required package: lattice
This is vegan 2.5-7
```

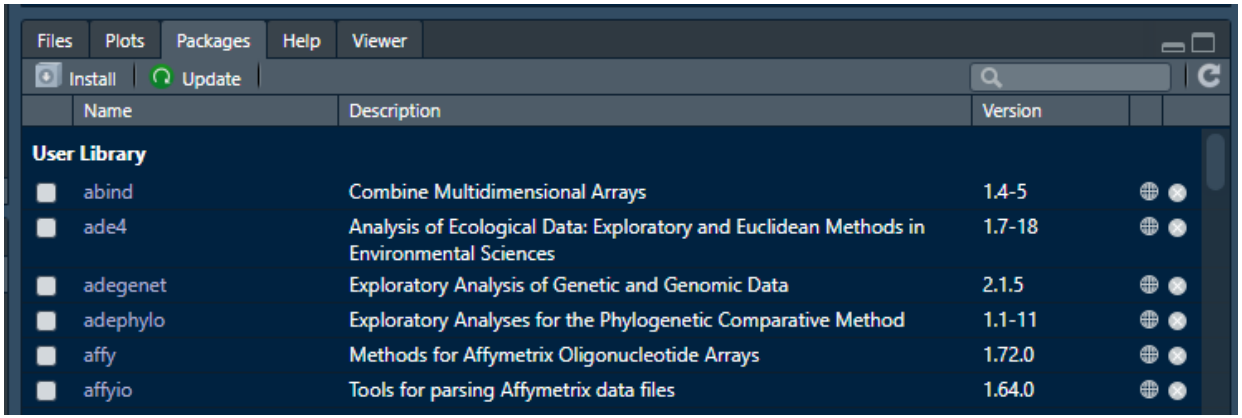
R Studio IDE

Top right – environment



Bottom right – files, plots, packages, help

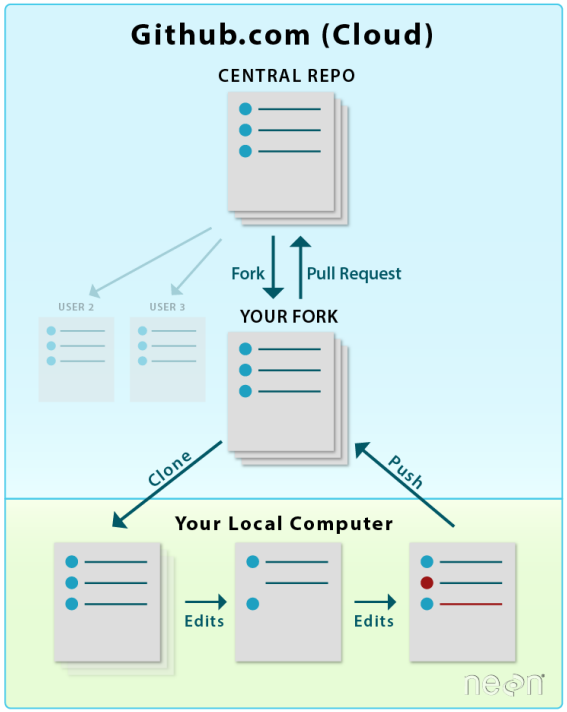




R basics – install packages

- Switch to html file

GitHub – what and why

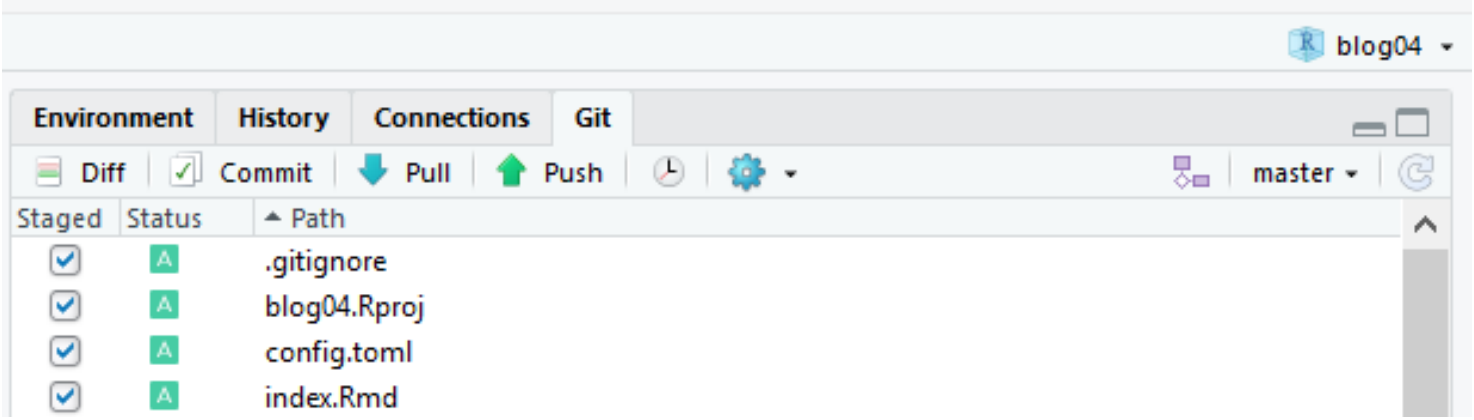


- GitHub is a web-based hosting service
  - **uses version control**
  - **allows for collaboration**
  - **supports reproducible research**

- NCSU version is private – limited to NCSU only
- Can connect to R studio
- Individual student repos
- ClassDocs and ClassData repos
  - Docs = syllabus, assignments, readings, presentations
  - Data = scripts (.html, .pdf), and data (.csv)

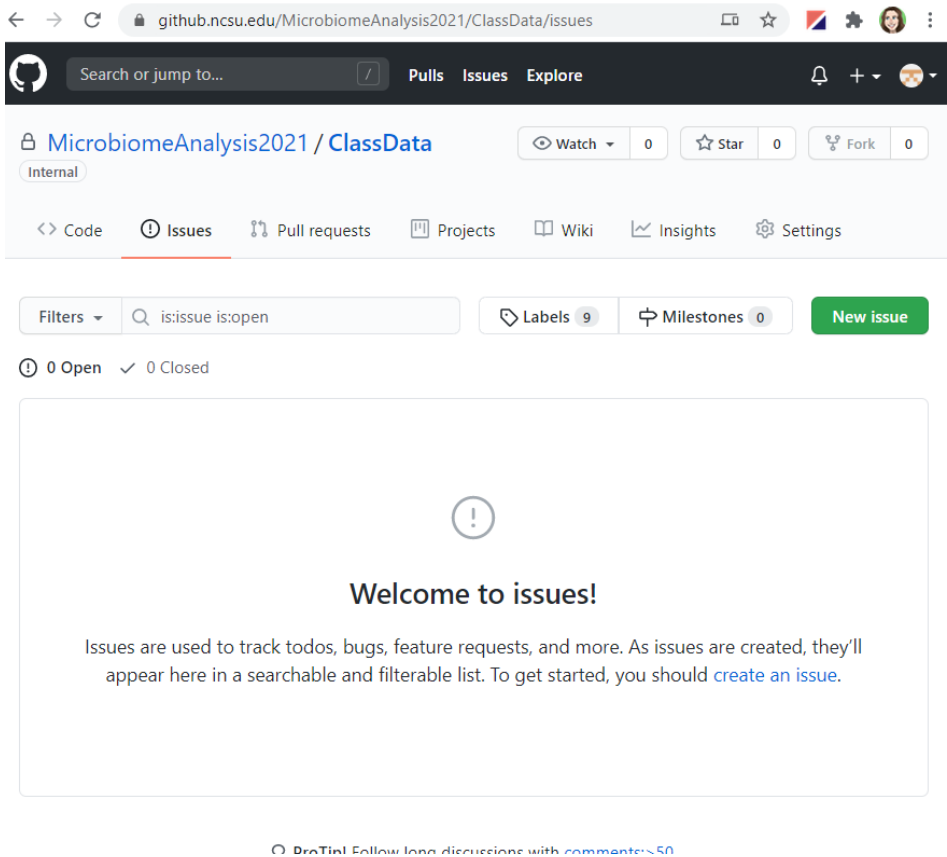
GitHub commits in RStudio

- When linked, Git tab appears in Environment window
- Check files to commit
- **Push** to upload modifications to GitHub repo
  - Provide detailed comments on what modifications were made in each push
- **Pull** to load modifications from the GitHub repo
  - Useful in collaborations when multiple people are working in the same repo



GitHub – Issues

- GitHub Issues allows us to ask and answer questions
- True for class and for many R packages
- Check existing issues before posting your own (do not duplicate)
- Office hours will check issues each week
- Help others with their issues



Practicing this will help you to use GitHub, Stack Overflow, and other

Reproducible Example (reprex)

- Whenever you post an issue, include a min reprex
- Use an informative issue title (at least Rpkg & command)
- Include single R script with
  - Packages loaded at top
  - Shortest amount of code that reproduces the problem
  - If data are needed, include the R code to recreate it (plus set.seed() if data are generated randomly)
  - Ensure code is easy to understand with informative variable names, comments to indicate the problem, etc.
  - Summarize your R environment by including output of sessionInfo() as a comment
- Confirm you have a reprex by starting a fresh R session and pasting your script in a new window

https://gist.github.com/hadley/270442

Reproducible Example (reprex)

Not easily reproducible

dplyr select using logical

Asked 6 years, 2 months ago Active 2 years, 9 months ago Viewed 8k times

Can `select` in dplyr be used with a logical vector?

Easily reproducible

dplyr select using logical

Asked 6 years, 2 months ago Active 2 years, 9 months ago Viewed 8k times

Can `select` in dplyr be used with a logical vector?

22

3

```
dat <- tbl_df(mtcars)
isNum <- sapply(dat, is.numeric)
select(dat, isNum)
```

data

code

```
select(dat, isNum)
```

Error in names(sel)[unnamed] <- sel[unnamed] : NAs are not allowed in subscripted assignments

*would be even better if they had included library(dplyr)

https://stackoverflow.com/questions/26875199/dplyr-select-using-logical?

Reproducible **Ex**ample (reprex)

Typical sessionInfo()

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18363)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] forcats_0.5.0     stringr_1.4.0     dplyr_1.0.3       purrr_0.3.4
##  [5] tidyr_1.1.2       tibble_3.0.5      ggplot2_3.3.3     tidyverse_1.3.0
##  [9] data.table_1.13.6 readr_1.4.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.0  xfun_0.20         haven_2.3.1       colorspace_2.0-0
##  [5] vctrs_0.3.6       generics_0.1.0    htmltools_0.5.1   yaml_2.2.1
##  [9] rlang_0.4.10      pillar_1.4.7      glue_1.4.2        withr_2.4.0
## [13] DBI_1.1.1         dbplyr_2.0.0      modelr_0.1.8      readxl_1.3.1
## [17] lifecycle_0.2.0   munsell_0.5.0     gtable_0.3.0      cellranger_1.1.0
## [21] rvest_0.3.6       evaluate_0.14     knitr_1.30        curl_4.3
## [25] fansi_0.4.2       broom_0.7.3       Rcpp_1.0.6        backports_1.2.1
## [29] scales_1.1.1      jsonlite_1.7.2    fs_1.5.0          hms_1.0.0
## [33] digest_0.6.27     stringi_1.5.3     grid_4.0.3        cli_2.2.0
## [37] tools_4.0.3       magrittr_2.0.1    crayon_1.3.4      pkgconfig_2.0.3
## [41] ellipsis_0.3.1    xml2_1.3.2        reprex_0.3.0      lubridate_1.7.9.2
## [45] assertthat_0.2.1  rmarkdown_2.6     httr_1.4.2        rstudioapi_0.13
## [49] R6_2.5.0          compiler_4.0.3
```

Reprex R package: https://reprex.tidyverse.org

Usage

Let's say you copy this code onto your clipboard (or, on RStudio Server or Cloud, select it):

```
(y <- 1:4)
mean(y)
```

Then call `reprex()`, where the default target venue is GitHub:

```
reprex()
```

A nicely rendered HTML preview will display in RStudio's Viewer (if you're in RStudio) or your default browser otherwise.

The relevant bit of GitHub-flavored Markdown is ready to be pasted from your clipboard (on RStudio Server or Cloud, you will need to copy this yourself):

```
```r
(y <- 1:4)
#> [1] 1 2 3 4
mean(y)
#> [1] 2.5
```
```

GitHub basics – link to RStudio

- Switch to html file

R Markdown

- Coursework is generally submitted as .Rmd rendered as HTML or PDF
- Allows us to create reproducible documents that integrate narrative text, code, and results (data analysis notebook)
- I provide HTML for class to make it easier to copy and paste code to your own script
- PDF may be easier for class exercises and for providing analysis documents to your advisor in the future

```
## Limit data to top 100 taxa based on abundance
```{r}
top100 <-names(sort(taxa_sums(ps_gp_bact), decreasing=TRUE)) [1:100]
ps_gp_top100 <- prune_taxa(top100, ps_gp_bact)
ntaxa(ps_gp_top100)
sample_sums(ps_gp_top100) #check that there are no zero samples

plot_bar(ps_gp_top100, "SampleType", fill = "Genus")
```
```

Limit data to top 100 taxa based on abundance

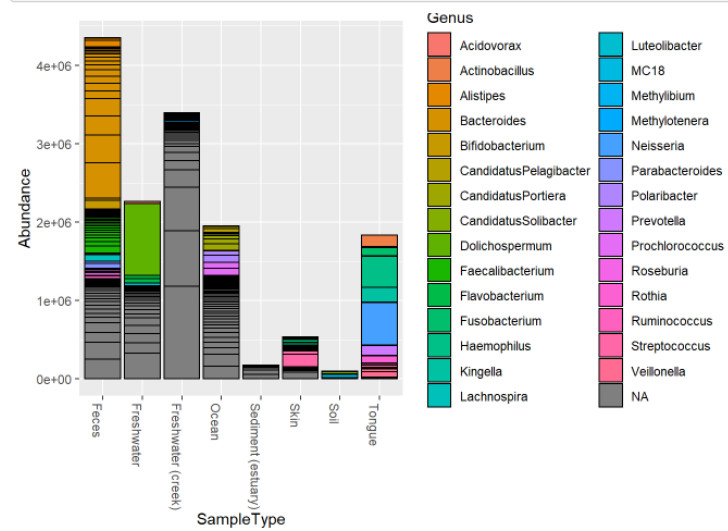
```
top100 <-names(sort(taxa_sums(ps_gp_bact), decreasing=TRUE)) [1:100]
ps_gp_top100 <- prune_taxa(top100, ps_gp_bact)
ntaxa(ps_gp_top100)
```

```
## [1] 100
```

```
sample_sums(ps_gp_top100) #check that there are no zero samples
```

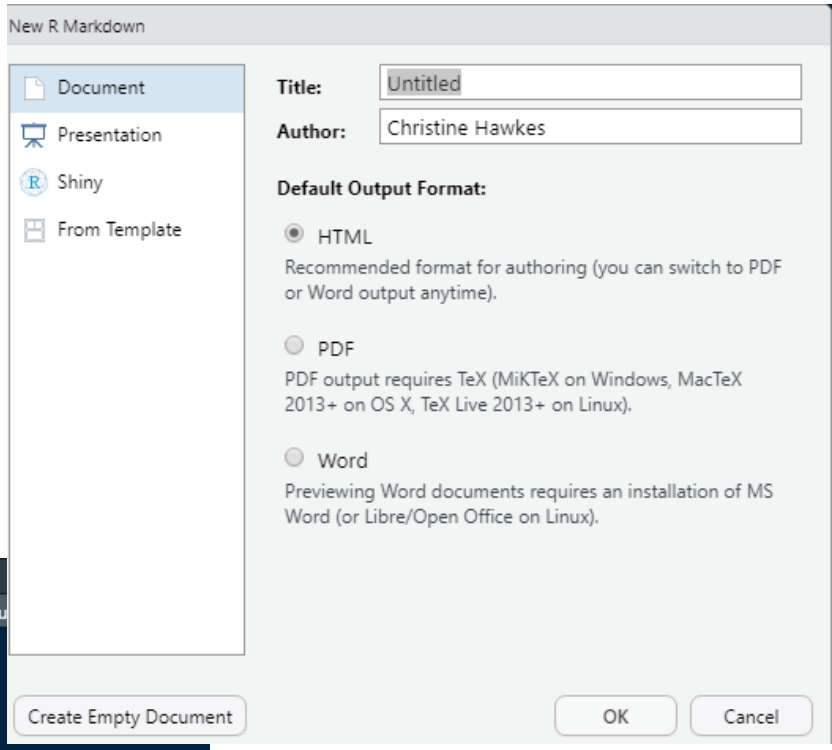
```
##      CL3      CC1      SV1  M31Fcsw  M11Fcsw  M31Plmr  M11Plmr  F21Plmr
##    44764    48214     3975  1061411  1747789   347461   123750    62271
##  M31Tong  M11Tong LMEpi24M SLEpi20M   AQC1cm   AQC4cm   AQC7cm      NP2
##  1757942    75387  1563626   697776   797234  1610632   984999   280223
##      NP3      NP5  TRRsed1  TRRsed2  TRRsed3     TS28     TS29
##   783531   885273     8482   127231    40525   596747   941428
```

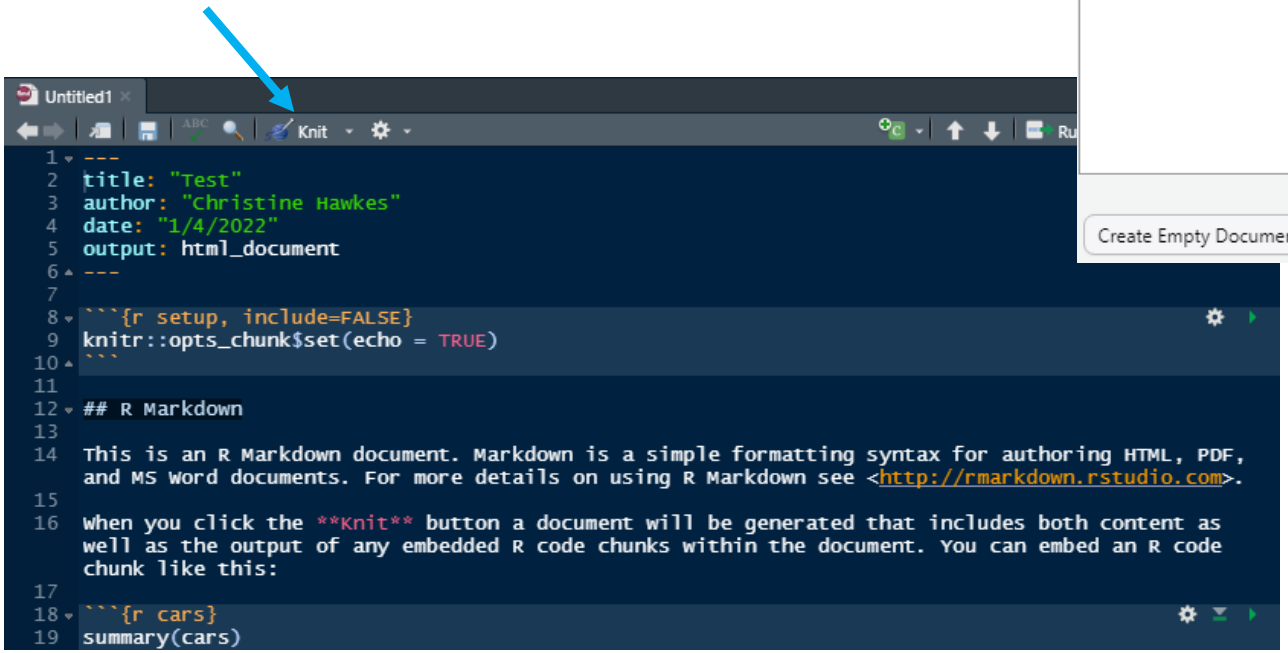
```
plot_bar(ps_gp_top100, "SampleType", fill = "Genus")
```



R Markdown

- From RStudio: *File > New File > R Markdown*
  - Provide informative title
  - Default is HTML
  - Save in your top-level project folder
- Use the .Rmd for your script
- Knit to render the file when ready





R Markdown elements

- YAML header (YAML = Yet Another Markup Language)
  - Author, date
  - Output format
  - Document appearance

```yaml
---
title: "Wk1 Microbiome Analysis Intro"
author: "Christine V. Hawkes"
date: "1/12/2022"
output:
  html_document:
    toc: true
    toc_float: true
---
```

- Embedded code chunks
  - PC: Ctrl + Alt + i
  - Mac: Command + Option + i

```{r}
# This is an empty code chunk

```

R Markdown Style

- Use style syntax to organize your headings and narrative text throughout the document

The syntax on the left renders as the output on the right.

Plain text. End a line with two spaces to start a new paragraph. Also end with a backslash\ to make a new line.	Plain text. End a line with two spaces to start a new paragraph. Also end with a backslash to make a new line.
italics and **bold** superscript^2^/subscript~2~ ~~strikethrough~~ escaped: * _ \\ endash: --, emdash: ---	*italics* and **bold** superscript^2/subscript$_2$ ~~strikethrough~~ escaped: * _ \ endash: –, emdash: —
# Header 1 ## Header 2 ... ###### Header 6	**Header 1** **Header 2** ... **Header 6**

- unordered list
- item 2
  - item 2a (indent 1 tab)
  - item 2b
 	
1. ordered list
2. item 2
  - item 2a (indent 1 tab)
  - item 2b
	• unordered list
	• item 2
	  • item 2a (indent 1 tab)
	  • item 2b
	
	1. ordered list
	2. item 2
	  • item 2a (indent 1 tab)
	  • item 2b

<link url> [This is a link.](link url) [This is another link][id]. At the end of the document: [id]: link url ![Caption](image.png) or ![Caption][id2] At the end of the document: [id2]: image.png	http://www.rstudio.com/ This is a link. This is another link.  Caption.

`verbatim code`
``
multiple lines
of verbatim code
``
	`verbatim code`
	multiple lines
	of verbatim code

R Markdown Style

Title of the first code chunk Info on this code chunk that is relevant for interpretation of its content and results Any relevant links or references

```
# This is an empty code chunk
```

Title of the first code chunk

Info on this code chunk that is relevant for interpretation of its content and results

- Any relevant *links* or **references** to `code`

```
# This is an empty code chunk
```

Template available on GitHub MicrobiomeAnalysis/ClassDocs

Troubleshooting errors in R

- Most common issue is getting an error message that you can't understand or don't know how to fix
- This happens to everyone at all stages of expertise!

- Check where the x shows up next to your code lines
- Common code mistakes: capitalization, misspelling, punctuation, libraries not loaded

```
Source on Save
25  Beetle <- filter(edidiv, taxonGroup == "Beetle")
26  Bird <- filter(edidiv, taxonGroup == "Bird")
27  Butterfly <- filter(edidiv, taxonGroup == "Butterfly")
28  Dragonfly <- filter(edidiv, taxonGroup == "Dragonfly")
29  Flowering.Plants <- filter(edidiv, taxonGroup == "Flowering
30  Fungus <- filter(edidiv, taxonGroup == "Fungus")
31  Hymenopteran <- filter(edidiv, taxonGroup == "Hymenopteran"
32  Lichen <- filter(edidiv, taxonGroup == "Lichen")
33  Liverwort <- filter(edidiv taxonGroup == "Liverwort")
34  Mammal <- filter(edidiv, taxonGroup == "Mammal")
35  Mollusc <- filter(edidiv, taxonGroup == "Mollusc")
36
37  # To find out the number of different species in each taxa,
38
39  a <- lengthunique(Beetle$taxonName))
40  b <- length(unique(Bird$taxonName))
41  c <- length(unique(Butterfly$taxonName))
42  d <- length(unique(Dragonfly$taxonName))
43  e <- length(unique(Flowering.Plants$taxonName))
```

```
>
>
>
>
>
>
>
>
>
> Liverwort <- filter(edidiv taxonGroup == "Liverwort")
Error: unexpected symbol in "Liverwort <- filter(edidiv taxonGroup"
> a <- lengthunique(Beetle$taxonName))
Error: unexpected ')' in "a <- lengthunique(Beetle$taxonName))"
> e <- length(unique(FloweringPlants$taxonName))
Error in unique(FloweringPlants$taxonName) :
  object 'FloweringPlants' not found
> Beetle <- filter(edidiv, taxonGroup == "Beetle")
Error in filter(edidiv, taxonGroup == "Beetle") :
  object 'taxonGroup' not found
>
```

Troubleshooting errors in R

- Read the details of the error message
- Check resources associated with the package either on GitHub Issues, in the documentation (?function_name), or in vignettes
- Check resources for common errors (see next slide)
- Google the error message
  - Error message + function or package name
  - Error message + r
  - Error message
- Restart R
- Still stuck? Post a reprex on the class or the package GitHub Issues page, on Stack Overflow, etc.

Troubleshooting resources

- https://ucsb-meds.github.io/teach-me-how-to-google/#1

- Common R errors
  - https://warin.ca/posts/rcourse-howto-interpretcommonerrors/
  - http://varianceexplained.org/courses/errors/
  - https://www.r-bloggers.com/2016/06/common-r-programming-errors-faced-by-beginners/
  - https://rpubs.com/Altruimetavasi/Troubleshooting-in-R

- General questions
  - https://stackoverflow.com/