

Organizing and Visualizing Data in R Workshop

By Gabriel Kamener
FCE Information Manager, Florida International University

Today's workshop

- Workshop will focus on:
 - RStudio projects
 - Organizing and visualizing data in R¹
 - Organizing and documenting code²
- After this workshop, you will be able to:
 - Organize data projects
 - Manipulate and visualize data in R
 - Organize, document, and review code

1. With content adapted from Seok et al. 2023

2. With content adapted from Ivimey-Cook et al. 2023

Today's workshop

- Workshop files:
https://github.com/gkamener/2024_spring_workshop_organizing_and_visualizing_data_in_r

Schedule

Time	Topic	Content
4:00 – 4:10	Introduction and setup	Why organize data? Why organize and document code?
4:10 – 4:15	Best practices in using RStudio projects	What are RStudio projects, and how can we utilize them?
4:15 – 4:35	Manipulating imported data	Manipulate, analyze, and export data with tidyverse.
4:35 – 4:50	Visualizing data	Plot and customize visualizations with ggplot.
4:50 – 5:20	Code organization and commenting	The 4Rs of code review, organizing code, reviewing code
5:20 – 6:00	Personal project and code development	Work on improving project and code

Introduction

- Why organize data projects?
- Why organize and document code?

My initial R experience

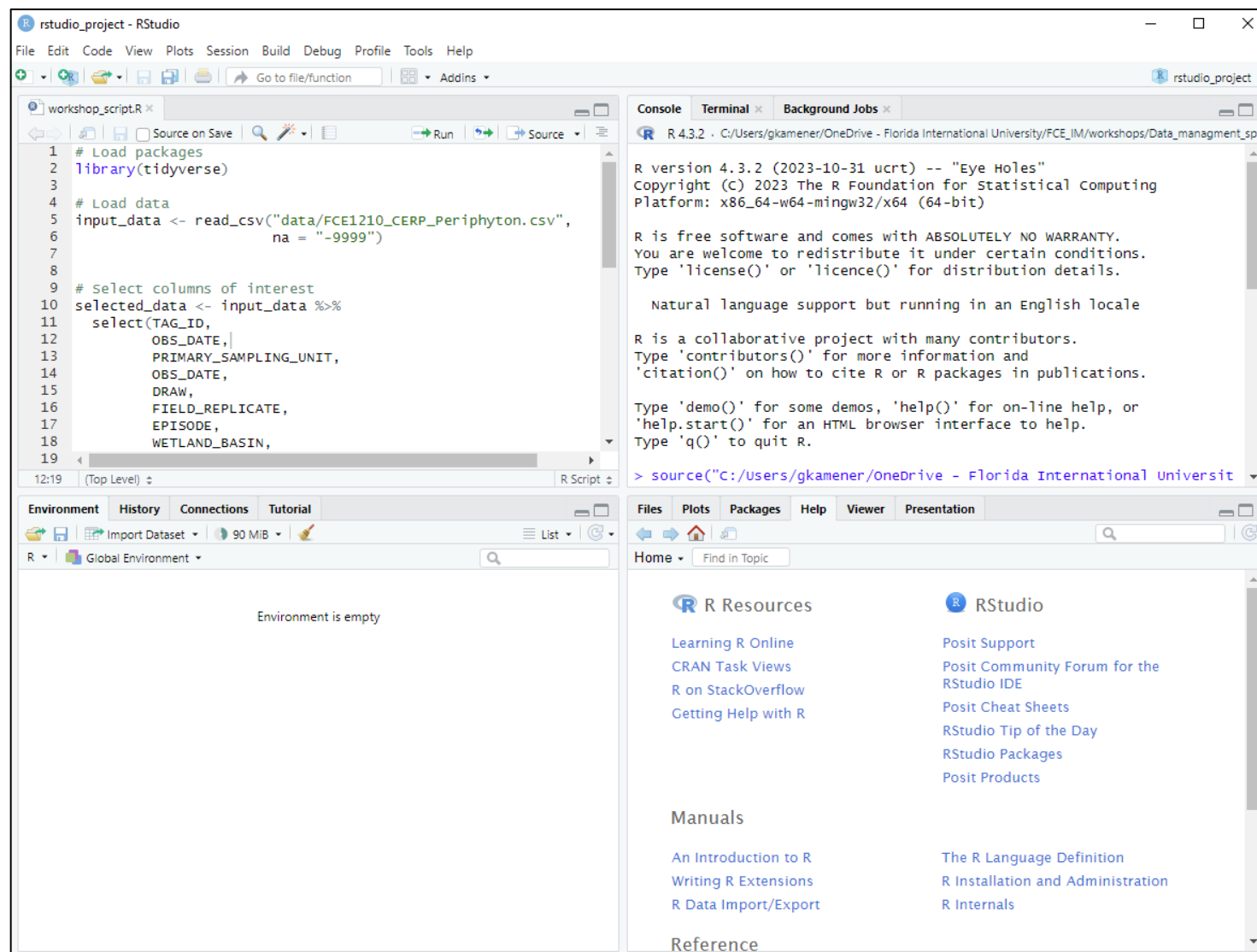


Artwork by @allison_horst (CC BY 4.0)

After some organization



RStudio



RStudio Projects

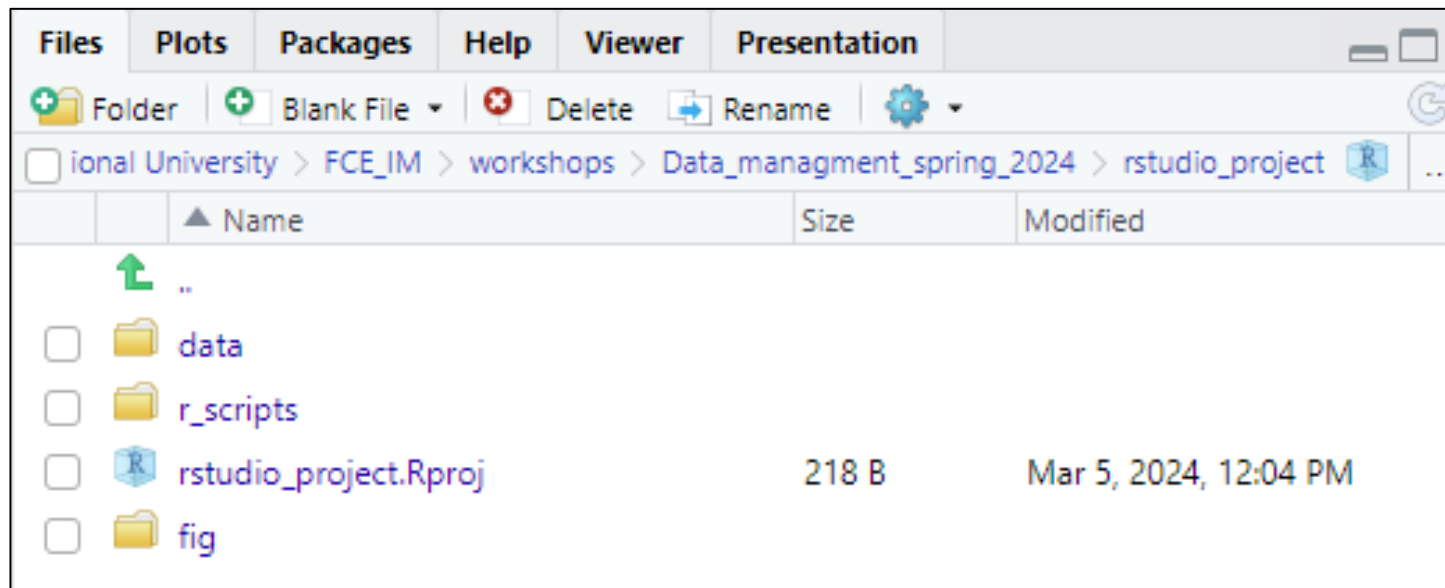
Advantages of RStudio Projects:

- Automatically set working directory
- Portable
- Share friendly
- Can integrate with Git/GitHub
- Can aid reproducibility with renv package

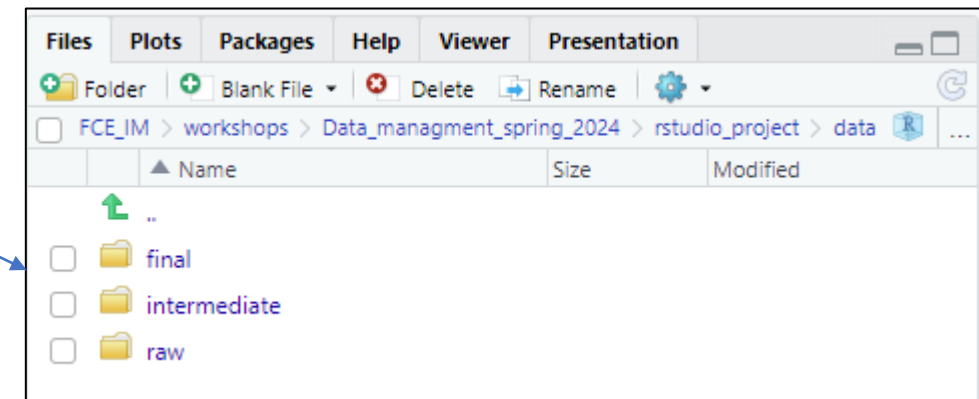
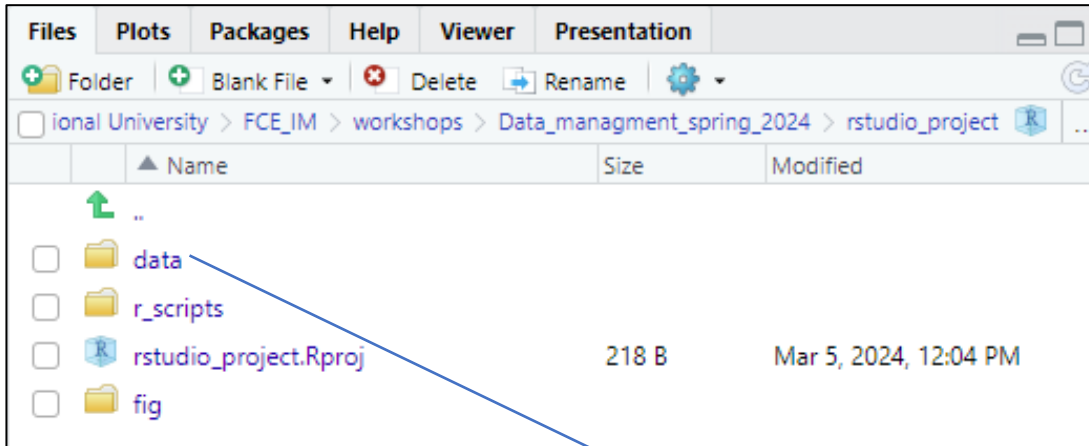
Setup an RStudio Project

1. Start RStudio.
2. Under the File menu, click on New Project. Choose New Directory, then New Project.
3. Enter a name for this new folder (or “directory”), and choose a convenient location for it. This will be your working directory for the rest of the day (e.g., ~/fce_data_workshop).
4. Click on Create Project.
5. Create folders for data (with “raw”, “intermediate”, and “final” subfolders), r scripts, and figures in your working directory.
6. Download code handout + data files and place into folders.
7. (Optional) Set Preferences to ‘Never’ save workspace in RStudio.

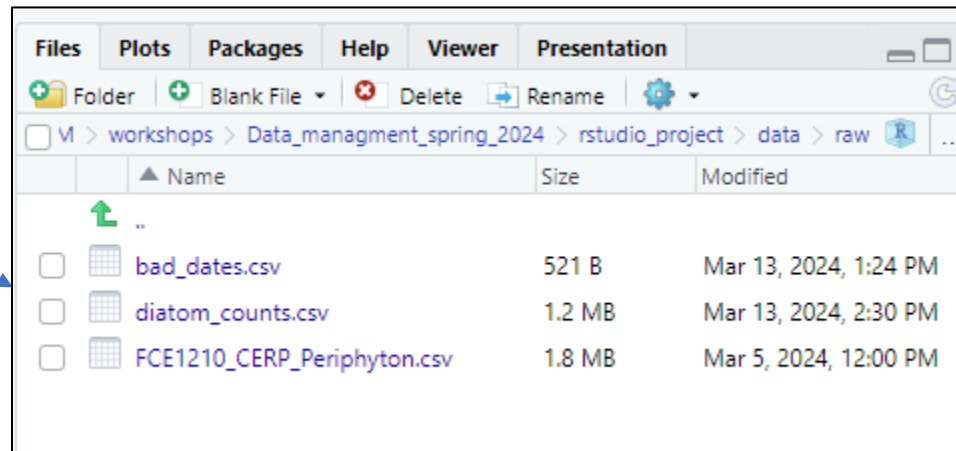
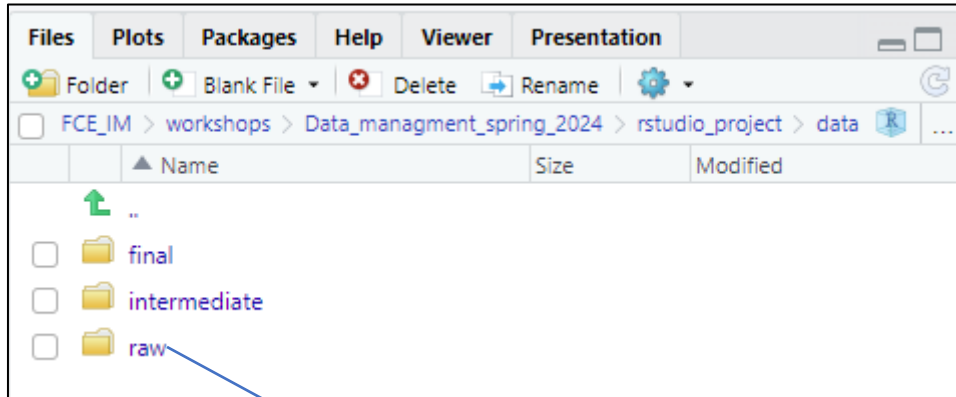
Organizing your working directory



Organizing your working directory



Organizing your working directory



The data we are using today

[Home](#) / [Data](#) / [Core](#) / [Metadata](#)

Metadata

Periphyton and Associated Environmental Data Relative from Samples Collected from the Greater Everglades, Florida, USA from September 2005 to November 2014

At a Glance

[Download data](#)

Authors: Evelyn Gaiser

Time period: 2005-09-14 to 2014-11-14

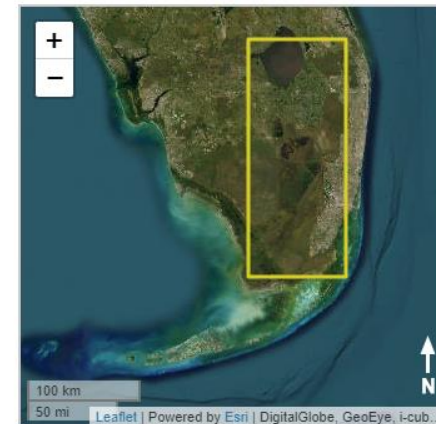
Package id: knb-lter-fce.1210.5

Dataset id: FCE1210_CERP_PeriphytonEnv

How to cite:

Gaiser, E.. 2022. Periphyton and Associated Environmental Data Relative from Samples Collected from the Greater Everglades, Florida, USA from September 2005 to November 2014. Environmental Data Initiative. <https://doi.org/10.6073/pasta/eadd93a36c2d935c069f3b0a4c98775b>. Dataset accessed 2024-03-13.

Geographic Coverage



View detailed metadata as: [HTML](#) [Text](#) [XML](#)

<https://fce-lter.fiu.edu/data/core/metadata/?packageid=knb-lter-fce.1210.5>

Starting with data

- Open code handout file
- `library(tidyverse)`

```
peri_df <- read_csv("data/raw/FCE1210_CERP_Periphyton.csv",  
                    na = "-9999")
```

```
diatoms_df <- read_csv("data/raw/diatom_counts.csv",  
                       na = "NA")
```

```
bad_dates_df <- read_csv("data/raw/bad_dates.csv",  
                         na = "-9999")
```

Indexing and subsetting data frames

- Indexed by row, column
- Can subset as `data_frame[row_index, column_index]`
- Subset data frame
 - `peri_df[1, 1]`
 - `peri_df[1,]`
 - `peri_df[, 1]`

Indexing and subsetting data frames

- Subset multiple rows/columns
 - `peri_df[c(1, 2, 3), c(5, 6)]`
 - `peri_df[1:3, 5:6]`
- Return vector instead with “[[]]”
 - `peri_df[[1, 1]]`

Indexing and subsetting data frames

- Subset by column name
 - `peri_df[1, "WATER_DEPTH_CM"]`
- Heads and tails
 - `first_ten_rows <- head(peri_df, 10)`
 - `first_ten_rows`
 - `last_ten_rows <- tail(peri_df, 10)`
 - `view(last_ten_rows)`

Factors

- Useful for categorical variables
 - `peri_df$FLOATING_SP1 <- factor(peri_df$FLOATING_SP1)`
 - `nlevels(peri_df$FLOATING_SP1)`
 - `levels(peri_df$FLOATING_SP1)`

Formatting dates

- `my_date <- ymd("2015-01-01")`
- `str(my_date)`

- `my_date <- ymd(paste("2015", "1", "1", sep = "-"))`
- `str(my_date)`

Formatting dates

- `paste(bad_dates_df$YEAR, bad_dates_df$MONTH, bad_dates_df$DAY, sep = "-")`
- `ymd(paste(bad_dates_df$YEAR, bad_dates_df$MONTH, bad_dates_df$DAY, sep = "-"))`

Formatting dates

- `bad_dates_df$DATE <- ymd(paste(bad_dates_df$YEAR, bad_dates_df$MONTH, bad_dates_df$DAY, sep = "-"))`
- `str(bad_dates_df)`
- `summary(bad_dates_df$DATE)`
- `missing_dates <- bad_dates_df[is.na(bad_dates_df$DATE), c("YEAR", "MONTH", "DAY")]`
- `head(missing_dates)`

Manipulating data

- Select
- Filter
- Mutate
- Pivot wider and longer
- Group and summarize
- Count
- Exporting data

Select

```
selected_peri <- peri_df %>%  
  dplyr::select(TAG_ID,  
                OBS_DATE,  
                PRIMARY_SAMPLING_UNIT,  
                FIELD_REPLICATE,  
                EPISODE,  
                WETLAND_BASIN,  
                WATER_DEPTH_CM,  
                FLOATING_SP1,  
                PERI_AFDM_G_PER_M2,  
                PERI_TP_UG_PER_G_DRY_MASS,  
                PERI_PROP_ORGANIC  
  )
```


Filter

```
filtered_peri <- selected_peri %>%  
  filter(month(OBS_DATE) > 8)
```

```
filter_missing_peri_tp <- selected_peri %>%  
  filter(!is.na(PERI_TP_UG_PER_G_DRY_MASS))
```

Mutate

```
mutated_peri <- filtered_peri %>% mutate(peri_percent_organic  
= PERI_PROP_ORGANIC*100)
```

```
mutated_peri %>%  
  select(PERI_PROP_ORGANIC,  
         peri_percent_organic)
```

Pivot wider and longer

```
diatoms_counts_long <- diatoms_df %>%  
  pivot_longer(-c(TAG_ID:LSU_NAME),  
               values_to = "SPECIMENS_COUNTED",  
               names_to = "TAXON_CODE")  
  
diatom_counts_wide <- diatoms_counts_long %>%  
  pivot_wider(names_from = "TAXON_CODE",  
              values_from = "SPECIMENS_COUNTED",  
              names_sort = TRUE,  
              values_fill = 0)
```

Group and summarize

```
summarized_water_depths <- filtered_peri %>%  
  group_by(WETLAND_BASIN,  
            YEAR = year(OBS_DATE)) %>%  
  summarize(mean_water_depth_cm = mean(WATER_DEPTH_CM))
```

```
summarized_water_depths
```

```
summarized_water_depths_wide <- summarized_water_depths %>%  
  pivot_wider(names_from = WETLAND_BASIN,  
              values_from = mean_water_depth_cm)
```

```
summarized_water_depths_wide
```

Group and summarize

```
diatoms_counts_filtered <- diatoms_counts_long %>%  
  filter(!is.na(SPECIMENS_COUNTED)  
    & SPECIMENS_COUNTED != 0)
```

```
total_diatoms <- diatoms_counts_long %>%  
  group_by(TAG_ID) %>%  
  summarize(TOTAL_COUNT = sum(SPECIMENS_COUNTED))
```

```
total_diatoms
```

Group and summarize

```
rel_abund <- diatoms_counts_filtered %>%  
  # Using dplyr's left_join function to join with the total_diatoms data frame  
  left_join(., total_diatoms, by = "TAG_ID") %>%  
  mutate(RELATIVE_PCT_ABUND = SPECIMENS_COUNTED/TOTAL_COUNT*100) %>%  
  select(TAG_ID,  
         OBS_DATE,  
         PRIMARY_SAMPLING_UNIT,  
         WETLAND_BASIN,  
         TAXON_CODE,  
         RELATIVE_PCT_ABUND  
  )  
  
rel_abund
```

Count

filtered_peri %>%

count(FLOATING_SP1, sort = TRUE)

diatoms_counts_filtered %>%

count(TAXON_CODE, sort = TRUE)

Exporting data

```
write_csv(filtered_peri, "data/intermediate/filtered_peri.csv")
```

```
write_csv(summarized_water_depths_wide,  
"data/final/summarized_water_depths_wide.csv")
```


Activity: make your own variable

- Organize your data
- Mutate a new variable
- Share that variable
(including how you made it)
with your neighbor

Visualizing data with ggplot

```
#ggplot(data = <DATA>, mapping = aes(<MAPPINGS>)) + <GEOM_FUNCTION>()
```

```
ggplot(data = filtered_peri)
```

```
ggplot(data = filtered_peri, mapping = aes(x = WATER_DEPTH_CM, y =  
PERI_AFDM_G_PER_M2))
```

Scatter plots

```
ggplot(data = filtered_peri, mapping = aes(x = WATER_DEPTH_CM, y =  
PERI_AFDM_G_PER_M2)) +  
  geom_point()
```

Boxplots

```
ggplot(data = filtered_peri, mapping = aes(x = WETLAND_BASIN, y =  
WATER_DEPTH_CM)) +  
  geom_boxplot()
```

Trend lines

```
ggplot(data = filtered_peri, mapping = aes(x = year(OBS_DATE), y =  
WATER_DEPTH_CM)) +  
  geom_line()
```

```
ggplot(data = summarized_water_depths, mapping = aes(x = YEAR, y =  
mean_water_depth_cm, color = WETLAND_BASIN)) +  
  geom_line() +  
  scale_x_continuous(breaks = c(2005:2014))
```

Stacked bar

```
rel_abund_2014_W3B <- rel_abund %>%  
  filter(year(OBS_DATE) == 2014  
    & WETLAND_BASIN == "W3B")  
  
ggplot(rel_abund_2014_W3B,  
  aes(fill = TAXON_CODE,  
    y = RELATIVE_PCT_ABUND,  
    x = PRIMARY_SAMPLING_UNIT)) +  
  geom_bar(position="stack", stat="identity")
```

Customizing plots

```
ggplot(rel_abund_2014_W3B,  
  aes(fill = TAXON_CODE,  
    y = RELATIVE_PCT_ABUND,  
    x = PRIMARY_SAMPLING_UNIT)) +  
  geom_bar(position="stack", stat="identity") +  
  labs(title = "Relative % abundance at W3B sites in 2014",  
    x = "Primary Sampling Unit",  
    y = "Relative Percent Abundance",  
    fill = "Diatom taxon code") +  
  theme(axis.title.x = element_text(size = 11),  
    axis.title.y = element_text(size = 15))
```

Customizing plots

```
rel_abund_2014_w3b_cut <- rel_abund %>%  
  filter(year(OBS_DATE) == 2014  
    & WETLAND_BASIN == "W3B") %>%  
  mutate(TAXON_CODE = if_else(RELATIVE_PCT_ABUND < 2,  
                                "OTHER",  
                                TAXON_CODE)) %>%  
  group_by(PRIMARY_SAMPLING_UNIT,  
            TAXON_CODE) %>%  
  summarize(RELATIVE_PCT_ABUND = sum(RELATIVE_PCT_ABUND))
```


Customizing plots

```
plot_2014_w3b_cut <- ggplot(rel_abund_2014_w3b_cut,  
                             aes(fill = TAXON_CODE,  
                                y = RELATIVE_PCT_ABUND,  
                                x = PRIMARY_SAMPLING_UNIT)) +  
  geom_bar(position="stack", stat="identity") +  
  labs(title = "Relative % abundance at W3B sites in 2014",  
        x = "Primary Sampling Unit",  
        y = "Relative Percent Abundance",  
        fill = "Diatom taxon code") +  
  theme(plot.title = element_text(size = 15),  
        axis.title.x = element_text(size = 12),  
        axis.title.y = element_text(size = 12))
```

Customizing plots

plot_2014_w3b_cut

Saving plots


```
ggsave("fig/W3B_2014_relative_abundance.png",  
       plot = plot_2014_w3b_cut)
```




```
ggsave("fig/W3B_2014_relative_abundance_custom_size.png",  
       plot = plot_2014_w3b_cut,  
       width = 46,  
       height = 26,  
       units = "cm",  
       limitsize = FALSE)
```

Make your own plot


- Make your own plot
- Share it with your neighbor (including how you made it)

Code organization and review

JOURNAL OF Evolutionary Biology 

FORUM |  **Open Access** |  

Implementing code review in the scientific workflow: Insights from ecology and evolutionary biology

Edward R. Ivimey-Cook , Joel L. Pick, Kevin R. Bairos-Novak, Antica Culina, Elliot Gould, Matthew Grainger, Benjamin M. Marshall, David Moreau, Matthieu Paquet ... [See all authors](#) ▾

First published: 09 October 2023 | <https://doi.org/10.1111/jeb.14230> | Citations: 1

<https://doi.org/10.1111/jeb.14230>

Potential code issues

- Reading in data files with paths outside project
- Order of workflow is unclear
- Comments say “go right” but code goes left
- Code fails to run on another computer
- Code fails or outputs change after updating R or packages
- Results not reproducible for manuscript reviewer

The four 'Rs' of code review

The 4Rs

- 1 Is the code as **R**eported?
Methods and code must match
- 2 Does the code **R**un?
Code must be executable
- 3 Is the code **R**eliable?
Code runs and completes as intended
- 4 Are the results **R**e producible?
Results must be able to be reproduced

Is the code as Reported?

- Are analyses as described in manuscript?
- Are relevant packages (with version numbers) documented in manuscript?

Does the code Run?

- `library(tidyverse)`
Error in `library(tidyverse)`: there is no package called 'tidyverse'.
- `install.packages("tRophicPosition")`
Warning in `install.packages` :
package 'tRophicPosition' is not available for this version of R
- Misspelled code

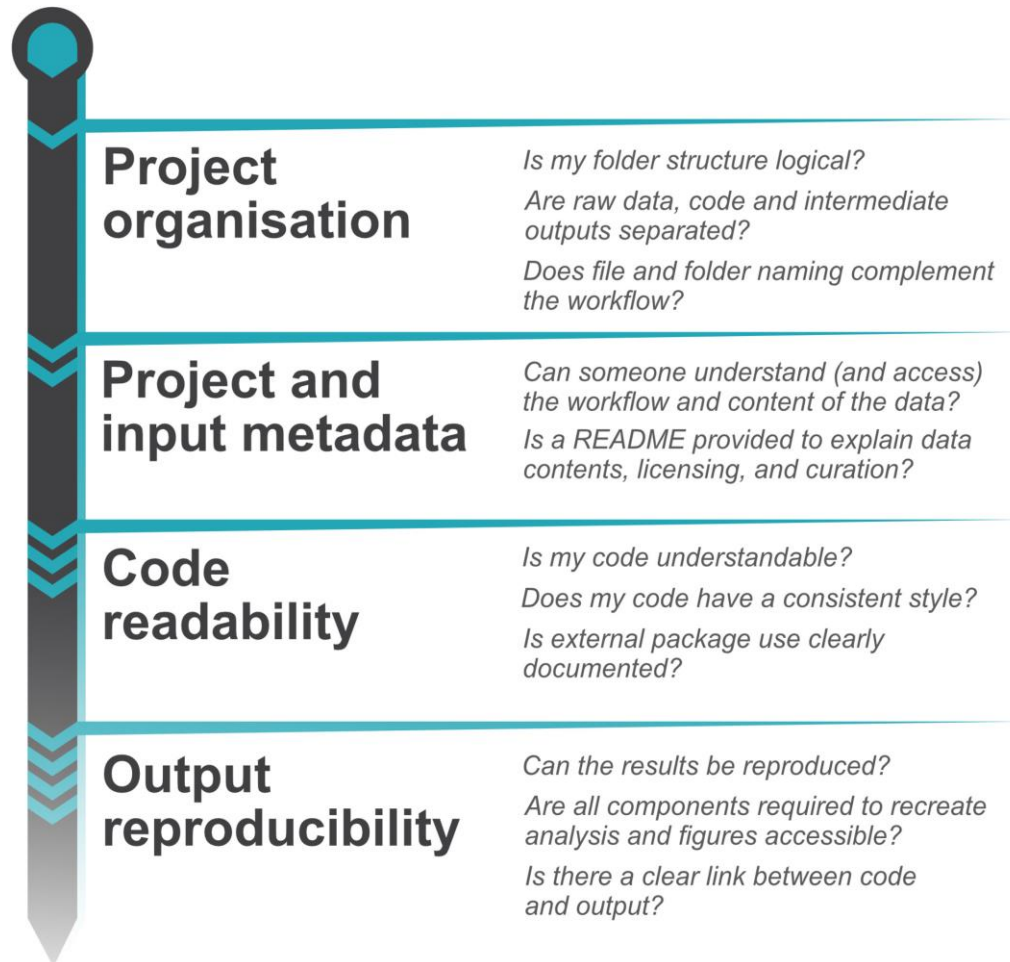
Is the code Reliable?

- #remove MC28 duplicate empty stomach
stomachs.raw <- stomachs.raw[-c(4),]
 - May run but removes different row than described

Are the results Reproducible?

- Do reproduced results match those reported?
- If they differ, by how much?
- Is an observed difference reasonable to expect?

Where code fits into the workflow



Project organization: data

FCE_IM > workshops > Data_managment_spring_2024 > rstudio_project >		
Name	Size	Type
data		File folder
fig		File folder
r_scripts		File folder
renv		File folder
.Rprofile	1 KB	RPROFILE File
readme.txt	0 KB	TXT File
renv.lock	34 KB	LOCK File
rstudio_project.Rproj	1 KB	R Project

<< FCE_IM > workshops > Data_managment_spring_2024 > rstudio_project > data		
Name	Size	Type
final		File folder
intermediate		File folder
raw		File folder

Project and input metadata: README

FCE_IM > workshops > Data_managment_spring_2024 > rstudio_project >		
Name	Size	Type
data		File folder
fig		File folder
r_scripts		File folder
renv		File folder
.Rprofile	1 KB	RPROFILE File
readme.txt	0 KB	TXT File
renv.lock	34 KB	LOCK File
rstudio_project.Rproj	1 KB	R Project

Date: 2024-03-14

Author: Gabriel Kamener

This is an RStudio project for the FCE LTER "Organizing and Visualizing Data in R" spring 2024 workshop.


Steps to run code in this project:

- 1: Open included .ppt or .pdf presentation file
- 2: Open .Rproj file to start an RStudio session
- 3: Load the "code_handout" R script from the /r_scripts folder
- 4: Follow the presentation, copy the code chunks into the R script, and run them

Resources for the code review portion of the workshop can be found in the presentation file and in the /code_review folder.

Project and input metadata: README

- Can scale in complexity

 README

JRN_grass_climate_correlation

Code and data for project investigating cycles of high and low grass cover on the Jornada for the past 100 years.

Folders

data - Contains raw data files and R scripts for handling data

Figures - Contains final figures for paper

other data sets - Contains data and scripts analyzing data from other locations

supplement - Analyses for supplement of paper

Workflow

Prepare data

Run scripts in 'data' folder to prepare quadrat grass data and climate variable data for analyses.

- `data/process_raw_plant_data.R`
- `data/get_climate_data.R`
- `data/determineenso_categories.R`

Run GAMs to smooth timeseries

A GAM is used to model the long-term trends in the grass timeseries. GAMs are also used to smooth each of the climate variables. Creates *smoothed_grass_gam.csv* and *smoothed_climate_variables.csv*.

- `Run GAMs_grass_pdo.R`

Main model scripts

The main models correlating grass with climate variables are run in `grass_pdo_model_selection.Rmd`. This script depends on intermediate csv files created in the first two steps. Creates *results_aic_table.csv*.

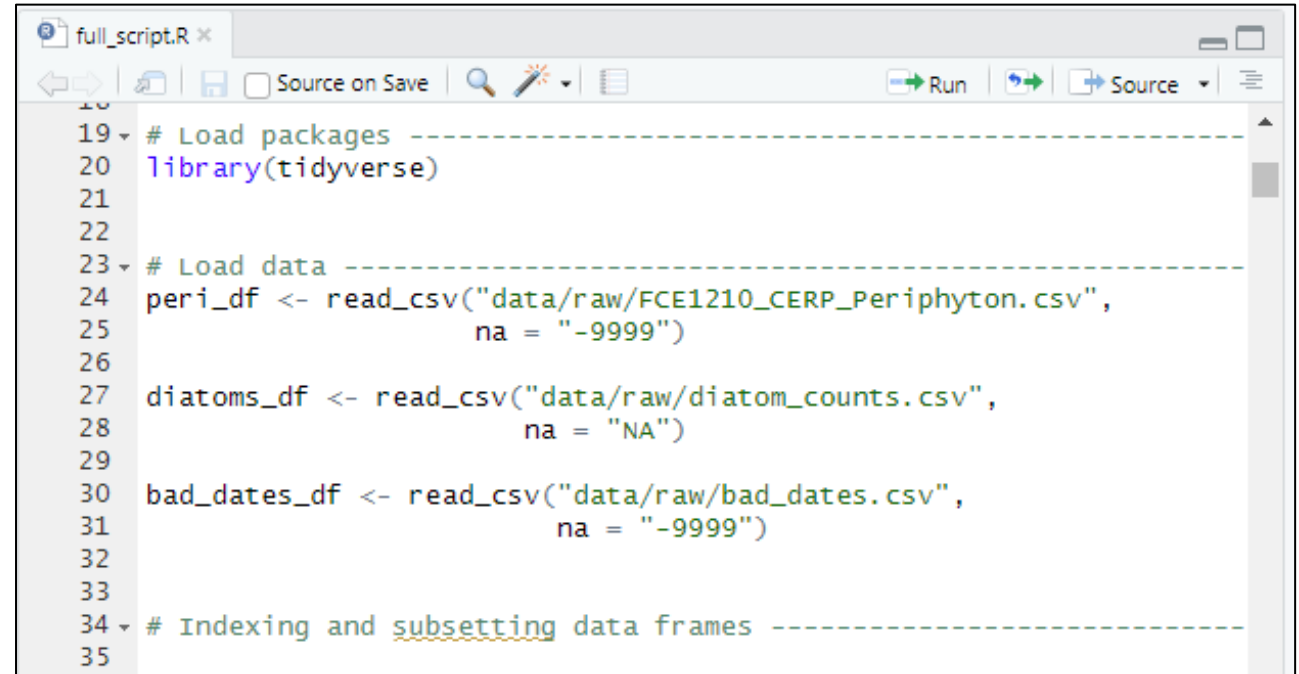
The figure of grass functional groups is created by `plot_grass_func_groups_timeseries.R`.

Code readability

1. Explicitly calling package namespace (e.g. `dplyr::select()`)
2. Using relative file paths
3. Removing redundant packages
4. Writing code with
 - a) Clear subheadings
 - b) Intuitive comments
 - c) Easy-to-understand object names

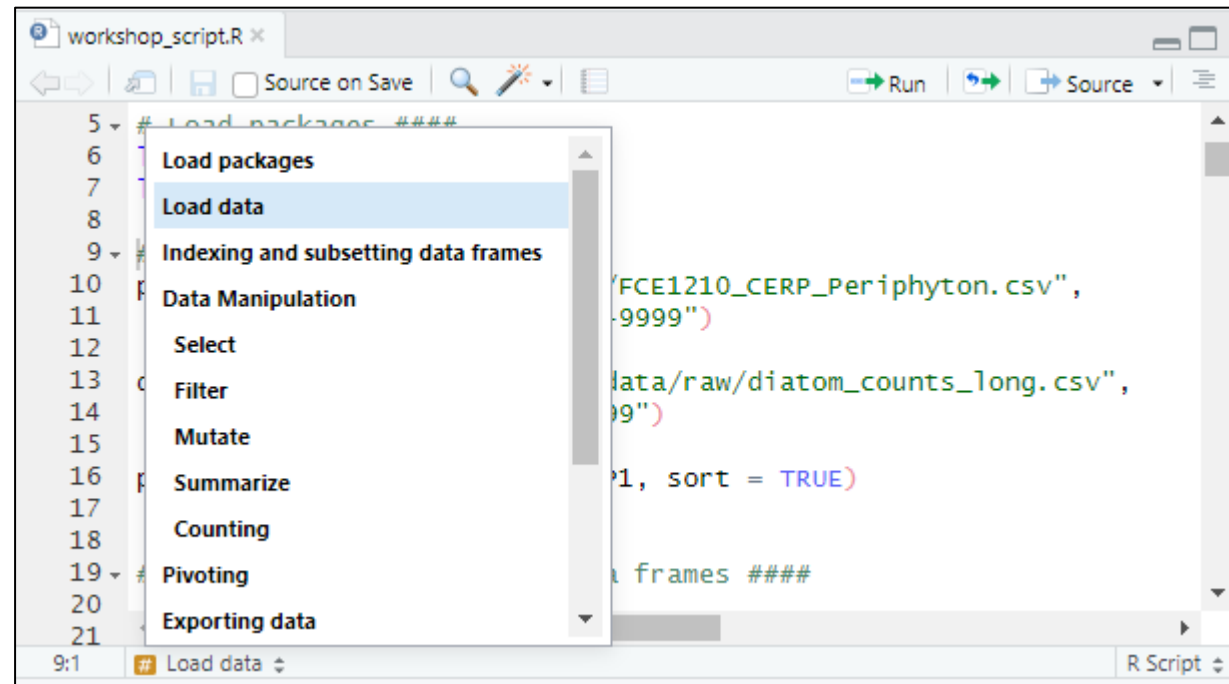
Code readability: code sections

- Break code down by region
- Comment line with at least 4 trailing “-”, “=”, or “#”
- Code -> Insert Section
- Crtl+Shift+R

A screenshot of an R script editor window titled 'full_script.R'. The editor shows a script with several sections separated by dashed lines. The first section is '# Load packages' with the code 'library(tidyverse)'. The second section is '# Load data' with three lines of code: 'peri_df <- read_csv("data/raw/FCE1210_CERP_Periphyton.csv", na = "-9999")', 'diatoms_df <- read_csv("data/raw/diatom_counts.csv", na = "NA")', and 'bad_dates_df <- read_csv("data/raw/bad_dates.csv", na = "-9999")'. The third section is '# Indexing and subsetting data frames'. The code is color-coded: comments are green, function names like 'library' and 'read_csv' are blue, and file paths are green. The editor has a toolbar at the top with icons for undo, redo, save, and other standard editing functions. The line numbers 18 through 35 are visible on the left side of the script.

```
18
19 ▾ # Load packages -----
20   library(tidyverse)
21
22
23 ▾ # Load data -----
24   peri_df <- read_csv("data/raw/FCE1210_CERP_Periphyton.csv",
25                       na = "-9999")
26
27   diatoms_df <- read_csv("data/raw/diatom_counts.csv",
28                           na = "NA")
29
30   bad_dates_df <- read_csv("data/raw/bad_dates.csv",
31                             na = "-9999")
32
33
34 ▾ # Indexing and subsetting data frames -----
35
```

Code readability: navigating sections



Output reproducibility

- Link between code sections and published outputs should be clear
- Use `set.seed()` before running simulations
- Increase reproducibility of environment with `renv` package

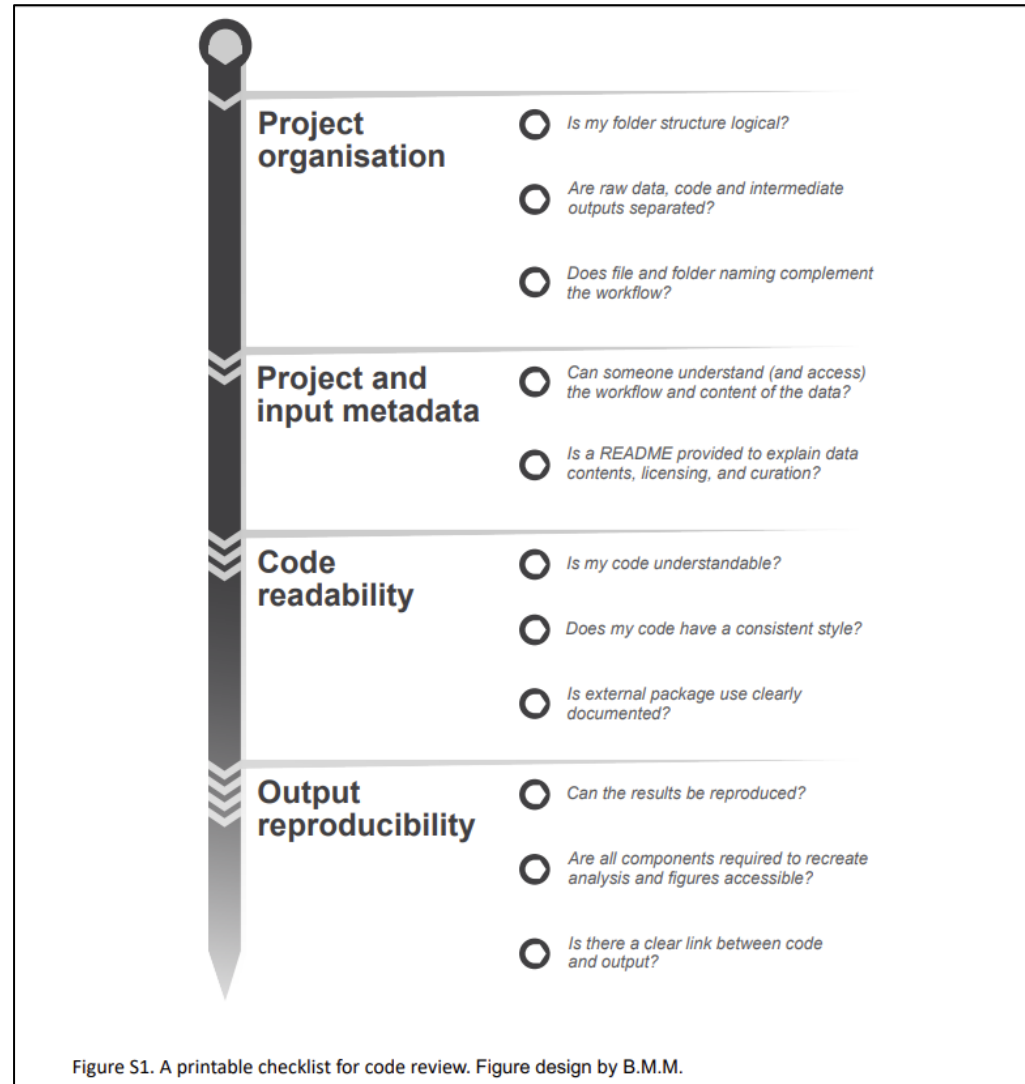
Generating citations

- R version citation: `citation()`
- Package version: `citation("dplyr")`
- `RStudio.Version()`

Other suggestions

- Keep primary R scripts for yourself
 - Include all analyses and comments
- Distill down code for manuscript submission
 - Only relevant analyses and comments
- Use version control (e.g. Git/GitHub) to track changes
- Code sections to cover each dependent variable
- Form code review group in your lab or with other students

Project code checklist



Project organisation	<input type="radio"/> <i>Is my folder structure logical?</i>
	<input type="radio"/> <i>Are raw data, code and intermediate outputs separated?</i>
	<input type="radio"/> <i>Does file and folder naming complement the workflow?</i>
Project and input metadata	<input type="radio"/> <i>Can someone understand (and access) the workflow and content of the data?</i>
	<input type="radio"/> <i>Is a README provided to explain data contents, licensing, and curation?</i>
Code readability	<input type="radio"/> <i>Is my code understandable?</i>
	<input type="radio"/> <i>Does my code have a consistent style?</i>
	<input type="radio"/> <i>Is external package use clearly documented?</i>
Output reproducibility	<input type="radio"/> <i>Can the results be reproduced?</i>
	<input type="radio"/> <i>Are all components required to recreate analysis and figures accessible?</i>
	<input type="radio"/> <i>Is there a clear link between code and output?</i>

Figure S1. A printable checklist for code review. Figure design by B.M.M.

Activity: Project and Code Development

- Use the checklist to review your own project and code
- Work with your neighbor to review each other's code or the provided example code

Constructive feedback

- How much of this did you know?
- How much did you not know?
- What would you like us to include or exclude in the future?

References and resources

- Christensen, E. 2023. “Emchristensen/JRN_Grass_Climate_Correlation: ARIMA Models.” Zenodo. <https://doi.org/10.5281/zenodo.7787243>.
- Ivimey-Cook, E. R., Pick, J. L., Bairos-Novak, K. R., Culina, A., Gould, E., Grainger, M., ... & Windecker, S. M. (2023). Implementing code review in the scientific workflow: Insights from ecology and evolutionary biology. *Journal of evolutionary biology*, 36(10), 1347-1356. <https://doi.org/10.1111/jeb.14230>
- Brian Seok, François Michonneau, Tobias Busch, Katrin Leinweber, Maneesha Sane, njlyon0, Ed Bennett, Hugo Tavares, Mike Mahoney, Paula Nieto, Susan Washko, Terry Loecke, Wasila Dahdul, xli677, Abhijna Parigi, Aleksander Jankowski, Allison Shay Theobold, Analytics Enlightened LLC, Anna K. Moeller, ... vmzhang. (2023). datacarpentry/R-ecology-lesson: Data Carpentry: Data Analysis and Visualization in R for Ecologists 2023-05 (2023.05). Zenodo. <https://doi.org/10.5281/zenodo.7892261>

References and resources

- Data Analysis and Visualization in R for Ecologists workshop website
 - <https://datacarpentry.org/R-ecology-lesson/>
- R cheatsheets (also findable as PDFs from Rstudio “help” tab)
 - <https://rstudio.github.io/cheatsheets/>
- Code Folding and Sections in the RStudio IDE
 - <https://support.posit.co/hc/en-us/articles/200484568-Code-Folding-and-Sections-in-the-RStudio-IDE>
- How to Cite R and R Packages
 - <https://ropensci.org/blog/2021/11/16/how-to-cite-r-and-r-packages/>
- Introduction to renv to help create reproducible environments for R projects
 - <https://rstudio.github.io/renv/articles/renv.html>