

Center for Conservation Biology | UC Riverside
Research Unit Resource Guide

Lynn Sweet | Principal Investigator, Assistant Research Ecologist
Julia Parish | Data Science Fellow

2022-09-09

Contents

Welcome to the CCB Research Unit Resource Guide	5
1 Introduction	7
1.1 Computer requirements	7
1.2 Software Installation	7
2 R and RStudio Installation	9
2.1 Install R	9
2.2 Install or Update R Studio	13
2.3 Install Quarto	15
2.4 Learn How to Use R & RStudio	15
3 Bookdown Guide	17
3.1 Primary Reference Resources	17
3.2 Install the Bookdown package	17
3.3 Render, Build & Publish the Resource Guide	18
3.4 Formatting	21
3.5 Citations	22
3.6 Alt Text for Accessibility	22
4 Git Installation & GitHub Account	25
4.1 GitHub	25
4.2 Git	25
4.3 Personal Access Token	27

5 Project Management Tools	33
5.1 UCR VPN - GlobalProtect	33
5.2 Google Apps	33
5.3 Microsoft 360 Suite	34
5.4 Trello	34
5.5 Zotero	34
5.6 Zoom	38
5.7 Slack	43
6 ESRI GIS Platform	47
6.1 ESRI ArcGIS	47
7 File Directory Organization	53
7.1 File Naming	53
7.2 Folder Organization	54
7.3 Data Management	54
8 Equipment	55
9 Code of Conduct	57
9.1 We are committed to working as a team.	57
9.2 We are committed to prioritizing and championing the people and communities that host us.	57
9.3 We are committed to the working hours, professional expectations and responsibilities defined by the overall project directors.	58
9.4 We are representatives and extensions of the University of California, Riverside and its staff, and of the professional bodies to which we and our project leaders are subscribed.	58
9.5 We recognize that fieldwork can be intense, emotional and tiring.	59
9.6 We have the right to a safe, secure and non-threatening working and living environment.	59
9.7 Important University of California, Riverside Links	60

Welcome to the CCB Research Unit Resource Guide

The Center for Conservation Biology (the Center, or CCB) is a University of California Riverside (UCR) organized research unit. Our mission is to assist in the conservation and restoration of species and ecosystems by facilitating the collection, evaluation, and dissemination of scientific information.



East of Cima Dome in the Mojave National Preserve, California. **Image Credit:** Center for Conservation Biology



Chapter 1

Introduction

This Resource Guide provides an overview of the the UC Riverside CCB research team work processes and expectations. It hosts documentation on how to install software and project management tools utilized by the CCB team to facilitate collaborative research.

If you have suggestions for additions or changes, please make a pull request or contact Lynn Sweet (lynn.sweet AT ucr.edu).

1.1 Computer requirements

CCB work computers (laptop or desktop) operating systems should either be Windows or macOS. Please note that Windows based machine it required to run ESRI ArcDesktop and ArcPro software. Both Windows and macOS may utilize ESRI ArcOnline tools.

For **Mac** users, update macOS to the newest supported version. Navigate to System Preferences → Software Update.

For **PC** users, ensure you have Windows 10 or 11 installed. If not, request a Windows key from UCR IT at UC Riverside ServiceLink

1.2 Software Installation

Software installation covered in this resource guide includes:

- ESRI Platform
 - ArcGIS Desktop

- ArcGIS Pro
 - ArcGIS Online
 - git
 - GitHub
 - Google Apps
 - R
 - RStudio
 - Quarto
 - Bookdown
 - Slack
 - Trello
 - Zotero
-

Thank you to UC Santa Barbara’s Bren School of Environmental Science & Management and National Center for Ecological Analysis and Synthesis (NCEAS) staff for providing many of the resources listed in this reference guide. Information was made available on the UCSB-MEDS GitHub page.

Chapter 2

R and RStudio Installation

2.1 Install R

R is a programming language and environment used for statistical computing and graphics. For more information, please visit What is R.

To install R, visit cloud.r-project.org to download the most recent version for your operating system.

Would you like a visual guide for install R and R Studio? Check out this YouTube R for Ecologist

2.1.1 Updating R

If you already have R installed, you may want to update the version of R you are using. Usually this occurs with a major version upgrades versus minor updates.

Please be aware that when you update R, it means re-installing all of the packages you have previously installed. It is worth updating R when major updates are released, especially if new packages you are attempting to install are not compatible with the version of R currently installed.

In order to update R, you have to find your installed version of R and run it on its own, outside of RStudio. This is easy if you have an R desktop shortcut, but not too hard if you hunt around a bit in your Applications folder.

Be Mindful



Do not just upgrade R to a new minor (or major) version PRIOR to updating your project. This will take a bit of work and time, maybe even days, depending on how many packages you have installed, as this will require re-installing them all into the library folder.

Please see details below in Section 2.1.2 to save all R packages currently installed PRIOR to updating R, so you won't have to reinstall them individually after the update

Double click the R icon to start up R. It will open the R Console and a menu. Click on the R menu at top left, and select Check for R Updates. If you are up to date, the R Console will report “Your version of R is up to date”. If not, this process will provide windows and buttons to click to upgrade to the latest version of R. When done, quit R and start RStudio to make sure the update has carried over. You should see the new version number when RStudio starts.

2.1.2 Bulk Save R Packages Prior to Updates

In order to follow the bulk save R package instructions, you will need to have R Studio installed, and be familiar with the terminal and console panels.**



Before you jump into upgrading R versions and packages, any important projects are protected. Make sure that your ongoing projects use the `{renv}` package to preserve their environments and that you have taken a recent snapshot with `renv::snapshot()`.

Before updating R, check out where your current package library is saved by entering `.libPaths()` in the console of RStudio. This should look something like:

```
[1] "/Library/Frameworks/R.framework/Versions/4.2/Resources/library"
```

Note that this package library is specific to the most recent major and minor version of R. If you upgrade from 4.1.x to 4.2.x, you will have a new library path to a new 4.2 folder, and will need to re-install packages in this folder.

You can do this from scratch, and install packages individually as needed (which is not a bad way to clean out packages you are not using), but you may prefer to re-install all packages in an organized way, rather than when you need to use one

You can see a dataframe of all currently installed packages by running the base R function `installed.packages()` in the R Studio console. This will help make a list of packages prior to updating R.

There is also a package for **Windows**, `{installr}`, which will pull the list of your current packages and install these in the correct folder for the new version of R. If you are on a Windows OS, and have installed the `{installr}` package, the `installr::updateR()` command in the console performs the following:

- finding the latest R version
- downloading it
- running the installer
- deleting the installation file
- copy old packages to the new R installation
- updating old packages as needed for the new R installation

But for MacOS and UnixOS folks, this must be done manually.

2.1.2.1 Saving a List of R Packages Manually

The short script below will save a vector of your current packages to a file in your root (home) directory, named `installed.packages.rda`. Copy this code chunk below into your Console pane in RStudio and run it. Then reopen RStudio, go to the Files tab, click on Home , and check your root (home) directory for the `installed.packages.rda` file. You can click on this file to load it into your Environment tab (answer yes to load). In your Environment tab, it should look like a character vector with the names of all of your packages in quotes.

```
# save list to tmp as a matrix object
tmp = installed.packages()

# filter for all rows where the 'Priority' column is NA,
# and select just (column 1) the package names in quotes,
# and then assign this vector to the installedpackages
# object
installedpackages = as.vector(tmp[is.na(tmp[, "Priority"])), 1])

# save this vector as an *.rda file in your root (home)
# directory
save(installedpackages, file = "~/installed_packages.rda")

# remove the tmp matrix and the installedpackages vector
# from your Environment
rm(tmp)
rm(installedpackages)
```

Now that you have your saved list of packages stored in a file, you are ready to upgrade R. Quit RStudio, and go to r-project.org.

2.1.2.2 Reinstalling list of R Packages

First, let's check your new R version with `R.Version()` - run this in the Console pane. This should be the new version.

Now let's check your library path with `.libPaths()` - run this in the Console Pane - this should match your R version.

Now run `installed.packages()`. This should show that your new library is mostly empty, except for the base R packages. We will now fix that.

Now you need to take advantage of your nicely-stored list of packages to reinstall all of your packages in your new (mostly empty) library.

Use the short script below to load the file `installed.packages.rda` to open a vector of your previous packages in your Environment pane, and then reinstall all of these packages. Copy the code chunk below into your Console pane

in RStudio and run it. This will take a while, especially if you have a lot of packages installed.

While you are waiting, check out the code chunk below. The first step is to load the `installedpackages.rda` file from your root (home) directory into your Environment pane. Then it starts taking action on this vector of package names. See what it does in the next step, the for loop, and think about how it works.
- It measures the length of the vector, `installedpackages`. - Then it counts from 1 to the length of `installedpackages`. - For each count, it installs the package at position [count] in the vector - Then it goes to the next [count] value, and installs the next package in the vector - Until it reaches the full length of the vector (installs the last package), and then stops.

```
# loads vector into your Environment pane
load("~/installed_packages.rda")

# For each package in this vector of length [count],
# install them one by one until the last one is installed.

for (count in 1:length(installedpackages)) {
  install.packages(installedpackages[count])
}
```

When all of the installation is complete, you can check your work by running `installed.packages()` again in the R Studio console. You should see a full list of your packages, and you are updated and ready to go!

2.2 Install or Update R Studio

RStudio is a software (considered an Integrated Development Environment, or IDE) that provides R programmers with an easy-to-use interface for coding in R.

Note: RStudio will not work without R installed, and you won't particularly enjoy using R without having RStudio installed. Be sure to install both!

- **New install:** To install RStudio, visit rstudio.com/products/rstudio/. Download the free (“Open Source Edition”) Desktop version for your operating system.
- **Update:** If you already have RStudio and need to update: Open RStudio, and under ‘Help’ in the top menu, choose ‘Check for updates.’ If you have the most recent release, it will return ‘No update available. You are running the most recent version of RStudio.’ Otherwise, you should follow the instructions to install an updated version.

Open RStudio (logo you'll click on shown below). **If you are prompted to install Command Line Tools, do it.**



Mac Users

There may be a need to install command line tools and XQuartz:

- To install command line tools (if you're not automatically prompted), run in the Terminal tab in RStudio: `xcode-select --install`
- Visit xquartz.org to download & install XQuartz.

2.2.1 Update R Packages in R Studio

You will want to periodically update R packages to ensure you have the most up to date version. You can do this within RStudio by navigating via the Task Bar to Tools/Check for Package Updates.

- Select All, and let the updates begin.

2.3 Install Quarto

This is an optional tool within R Studio that is extremely powerful, but it is not required.

Quarto is a scientific publishing tool built on Pandoc that allows R, Python, Julia, and ObservableJS users to create dynamic documents, websites, books and more.

As of July 2022, Quarto comes pre-installed in R Studio (v2022.07). If you haven't updated your R Studio IDE (and concerned about doing so), you can install Quarto separately.

- Download Quarto here and install
- To use Quarto through the RStudio IDE, be sure to have at least version v2022.02 installed (see directions in step 2, above)

Learn more about Quarto [here](#).

2.4 Learn How to Use R & RStudio

There are many resources out there to help you learn how to use R and the RStudio IDE (YouTube, Googling, StackOverflow, etc...). This is a short list of primary resources for you to reference. We encourage you to join R community Slack workspaces, R User community groups (such as R-Ladies), and UC Riverside Data Science clubs!

- ACM at UCR - UC Riverside community dedicated to technical, professional, and personal development in the context of computer science.
- R for Data Science
- R Markdown Cookbook
- Cookbook for R
- Advanced R
- R Studio (posit) Book Catalog

- R for Data Science Online Learning Community
- R OpenSci
- RStudio Cloud - interactive tutorials to learn data science basics.
- R Studio Cheatsheets - invaluable tool to learn how to use various R packages.
- Swirl - R package that is a built in tutorial.
- Teacups, Giraffes, & Statistics - an interactive tutorial to learn statistics and R coding, plus it is a beautiful!
- Data Science in a Box
- Adventures in R
- Data Analysis & Visualization in R for Ecologist

Chapter 3

Bookdown Guide

This Resource Guide was created using the **bookdown** package. **bookdown** provides a platform to publish references using various output formats (PDF, HTML, Word) and allows for interactivity, insertion of code chunks, and supports a variety of languages (R, Python, Julia, SQL, etc.).

3.1 Primary Reference Resources

Here is a list of resources to learn how to use and edit in bookdown

- Bookdown Package Documentation
- Authoring Books with R Markdown
- R Markdown Cookbook
- R Markdown: The Definitive Guide

3.2 Install the Bookdown package

The first step to edit and add to this Resource Guide is to install the **bookdown** package from CRAN or Github. In the RStudio console, run the following:

```
install.packages("bookdown")
# or the development version
# devtools::install_github('rstudio/bookdown')
```

3.3 Render, Build & Publish the Resource Guide

When updates are made to the Resource Guide, it needs to be render, built, and then the updates published.

In your Console, type either of these commands depending on which type of render you prefer:

```
bookdown::render_book("index.Rmd", "bookdown::gitbook") bookdown::render_book("index.Rmd", "bookdown::pdf_book")
```

- *Note:* to compile to PDF, you need XeLaTeX. It is recommended to install TinyTeX (which includes XeLaTeX): <https://yihui.name/tinytex/>.

Once you have rendered updates using the console commands above:

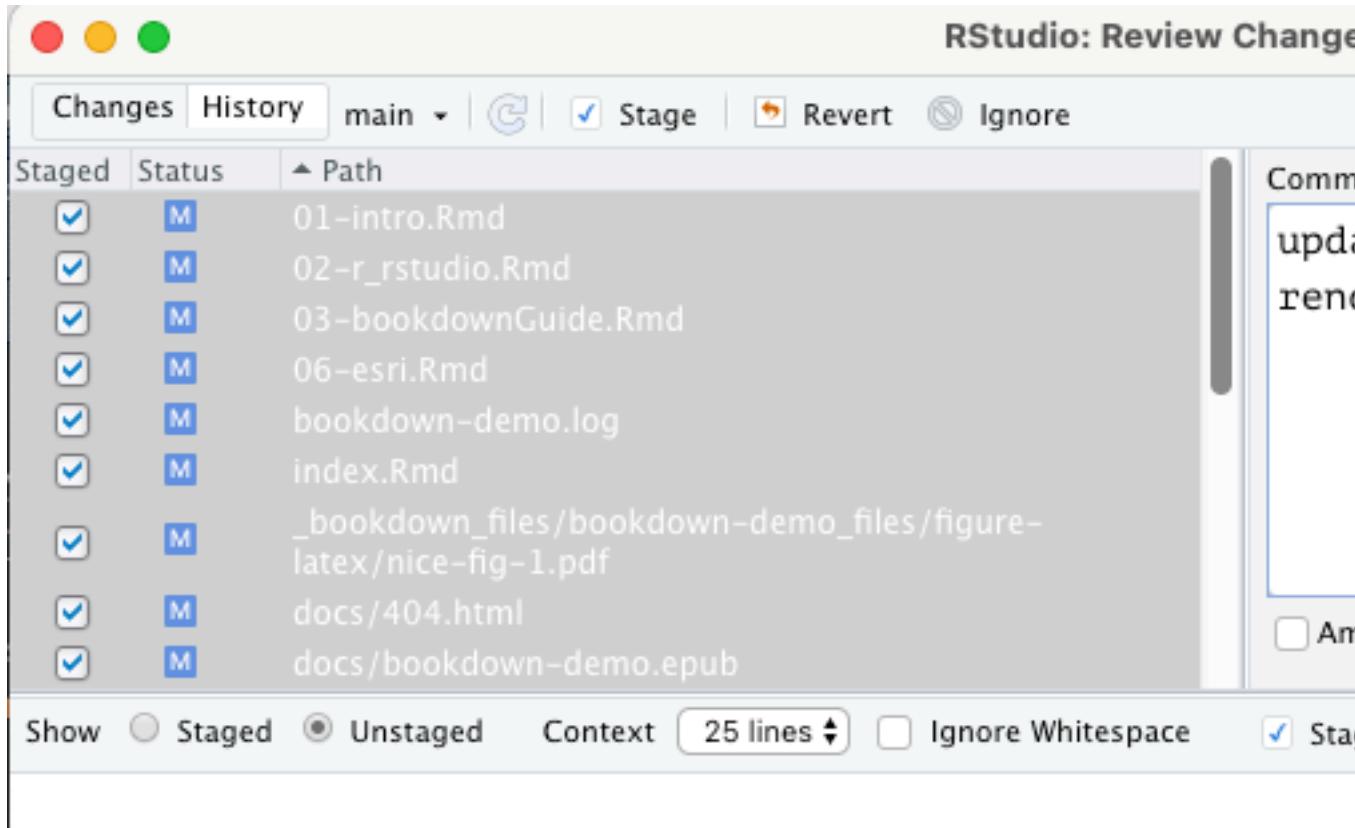
- navigate to the **Environment**, **Connections**, **Build**, **Git** panel,
- select the **Build** tab,
- select **Build Bookdown**
- once the build is complete (the final line should read `Output created: docs/index/html`), commit, pull and push updates to GitHub via the **Git** tab.

```
Environment History Connections Build Git Tutorial
Build Book More
brary/Frameworks/R.framework/Versions/4.1-arm64/Resources
ua/pagebreak.lua --lua-filter /Library/Frameworks/R.frame
ces/library/rmarkdown/rmarkdown/lua/latex-div.lua --meta
ftx0sx5dxbhjvxvvz840000gn/T//Rtmp7gbD31/file6c51f76b5e6 -
section-divs --template /Library/Frameworks/R.framework/V
brary/bookdown/templates-bs4_book.html --highlight-style
athjax --include-in-header /var/folders/8c/syp2qftx0sx5dx
1/rmarkdown-str6c52c15fb5d.html --citeproc
Tweaking docs/index.html
Tweaking docs/intro.html
Tweaking docs/r.html
Tweaking docs/bookdown.html
Tweaking docs/git.html
Tweaking docs/projmgmt.html
Tweaking docs/esri.html
Tweaking docs/organization.html
Tweaking docs/references.html
Tweaking docs/404.html

Output created: docs/index.html
Warning message:
The label(s) methods not found
```

The screenshot shows the RStudio interface with the Git tab selected. The main area displays a list of files staged for commit. The columns are Staged, Status, and Path. The Status column uses icons to indicate file status: blue 'M' for modified, green '+' for added, and red '-' for deleted.

Staged	Status	Path
✓	M	01-intro.Rmd
✓	M	02-r rstudio.Rmd
✓	M	03-bookdownGuide.Rmd
✓	M	06-esri.Rmd
✓	M	bookdown-demo.log
✓	M	index.Rmd
✓	M	_bookdown_files/bookdown-demo_files/figure-latex/nice
✓	M	docs/404.html
✓	M	docs/bookdown-demo.epub
✓	M	docs/bookdown-demo.pdf
✓	M	docs/bookdown-demo.tex
✓	M	docs/bookdown.html
✓	M	docs/esri.html
✓	M	docs/index.html
✓	M	docs/intro.html
✓	M	docs/r.html
✓	M	docs/reference-keys.txt
✓	M	docs/search.json
✓	M	docs/search_index.json
✓	A	docs/images/build.png
✓	A	images/build.png



The following information is directly taken from the *bookdown* package (Xie, 2022).

3.4 Formatting

You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

Remember each Rmd file contains one and only one chapter, and a **chapter** is defined by the first-level heading #.

You can label chapter and section titles using {#label} after them, e.g., we can reference Chapter 1. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter ??.

Figures and tables with captions will be placed in **figure** and **table** environments, respectively.

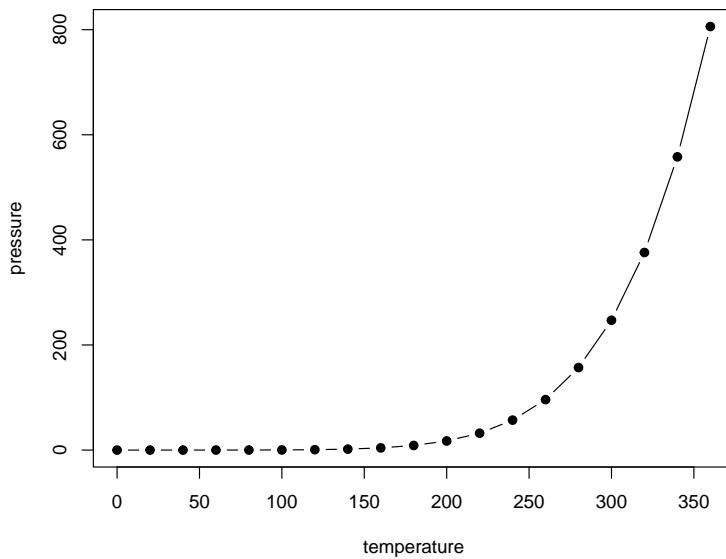


Figure 3.1: Here is a nice figure!

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 3.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 3.1.

3.5 Citations

You can easily write citations using .bib files within this repository formatted using BibTEX. For example, the `bookdown` package (Xie, 2022) in this reference book, which was built on top of R Markdown and `knitr` (Xie, 2015).

3.6 Alt Text for Accessibility

Use the `knitr` package to add alt text to graphics in R Markdown files

Table 3.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

Chapter 4

Git Installation & GitHub Account

4.1 GitHub

GitHub is a internet based code hosting platform for collaboration and version control. GitHub lets you (and others) work together on projects.

Navigate to github.com, and create an account! Please use either your work or personal email account. You can add several emails to your account, and assign a particular email as the primary email for the account.

Review this article on choosing a GitHub username: happygitwithr.com/github-acct.html.

Once you have created a GitHub account, join the CCB's GitHub Organization. Please email the CCB PI to request they invite you to the organization.

4.2 Git

Git is a tool utilized for source code management. It tracks changes and updates to code and allows for multiple users to work together within the same repository. Recommended reading for background information on Git and its utility (Yildirim, 2020).

Check to see if your computer already has git:

1. Open RStudio
2. In the terminal, run the following command:

- `which git` (Mac, Linux)
- `where git` (Windows)

If `git` is already installed, the return to the above command should return a filepath similar to:

- MacOS `/usr/local/bin/git`
- Windows `C:/Program Files/Git/bin/git.exe`

3. If there is no response, download and install `git` from here: git-scm.com/downloads. Select the default settings within the prompts **except** the default to `master` branch. This branch is being phased out, so select the option that let's you select alternative branches (ex: `main`).

Once Git is installed and your GitHub account has been set up, Git needs to be configured on the computer. Configuring Git is required to push & pull commits to GitHub.

Configuring Git:

1. In RStudio, open the terminal.
2. Run the following commands separately, pressing `Enter` after each line. Replace username and email with your GitHub account username and email. Make sure to keep the quotes around the username in the code below:

- `git config --global user.name "Jane Doe"`
- `git config --global user.email janedoe@example.com`

3. Once you have entered the above command line code, check that the configuration was set correctly. In the RStudio Terminal, type the following command and hit enter:

- `git config --list --global`

In the terminal, it should show code similar to this in the Terminal:

```
user.name=janedoe user.email=janedoe@ucr.edu core.excludesfile=/Users/janedoe/.gitignore
```

If when installing or updating Git, the default branch was not set to `main` (it is defaulting to the old `master` branch), you can change this setting globally. In the Terminal again, enter the line below:

```
git config --global init.defaultBranch main
```

For more information on configuring Git: check out this Git reference

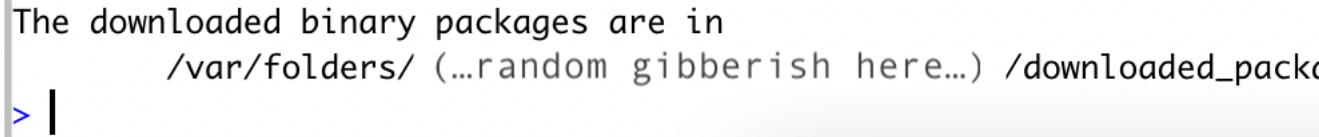
4.3 Personal Access Token

Once Git has been configured to commit to your GitHub account, a **Personal Access Token (PAT)** must be created for **each computer you intend to use**. A PAT is an alternative password authentication method for Git to access GitHub accounts.

- In the RStudio Console, install the `usethis` package in R by running the following code:

```
install.packages("usethis")
```

If the `usethis` package is installed correctly, at the end of the stream of text there should be a message similar to the image below:



```
The downloaded binary packages are in  
/var/folders/ (...random gibberish here...) /downloaded_packages
```

- Once the `usethis` package is installed, run the following in the RStudio Console:

```
usethis::create_github_token()
```

- Enter your GitHub password when prompted.

This should take you to the **Settings/Developer settings** section of your GitHub account:

[Settings / Developer settings](#)[GitHub Apps](#)[OAuth Apps](#)[Personal access tokens](#)

New personal access token

Personal access tokens function like OAuth tokens, but provide full access over HTTPS, or can be used to authenticate to GitHub API endpoints.

Note

CCB_PC_WorkComputer

What's this token for?

Expiration *

Custom...

01/31/2023

Select scopes

Scopes define the access for personal access tokens.

 [repo](#)

Fu

 [repo:status](#)

Acc

 [repo_deployment](#)

Acc

 [public_repo](#)

Acc

 [repo:invite](#)

Acc

 [security_events](#)

Rea

 [workflow](#)

Up

 [write:packages](#)

Up

 [read:packages](#)

Dow

- **Note Field**

Change the PAT name to a meaningful reference (see image for an example). You may end up creating multiple PATs, so you want to ensure that you know which PAT is designated for each computer | server.

- **Expiration Field**

This is to select a set expiration timeframe for your PAT. Setting an expiration is highly recommended, and GitHub will send you *SEVERAL* emails prior to it expiring to remind you to renew it. Use the drop down to select a set time frame (7 days to 90 days) or create a custom expiration time frame (exactly a year or particular date).

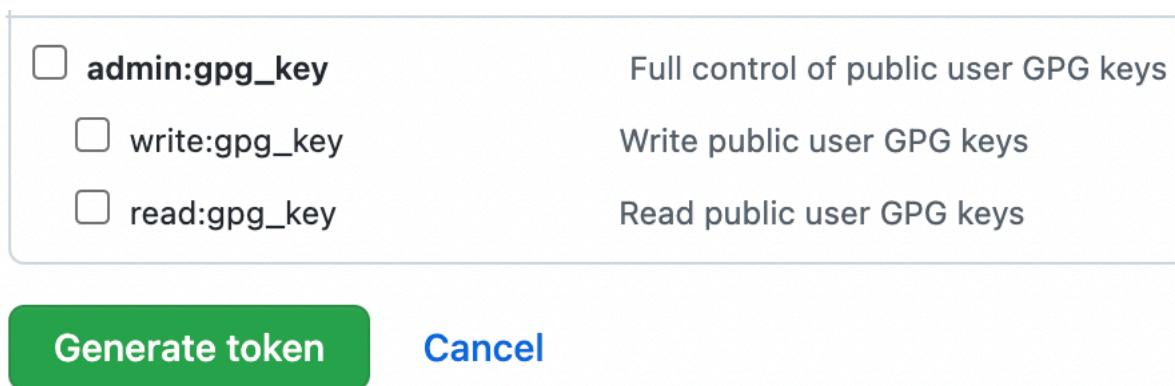
- **Select Scopes Field**

Define access for the Personal Access Token being generated.

It is recommended to select at least the following scopes: - repo - workflow - write:packages - notifications - delete repo - write:discussions - project

To learn more about Scopes, visit the GitHub Scopes for OAuth Apps page.

- Once Scopes are selected, click on the green **Generate token** button:



- Copy the generated PAT to your clipboard.
- Paste and Save this PAT in a text file in a secure folder that will NEVER be accessed by other users or the internet. You can create a **private** folder on your personal computer to store these files.

- Return to RStudio Console and run the following command:

```
gitcreds::gitcreds_set()
```

You will be prompted to paste the PAT into the console:

```
> gitcreds::gitcreds_set()
```

```
? Enter password or token: [REDACTED] REDACTED
-> Adding new credentials...
-> Removing credentials from cache...
-> Done.
```

Paste the PAT at the end of the line `Enter password or token:` and press enter.

- In the console, run:

```
usethis::git_sitrep()
```

This command should return your GitHub account information (see example below).

```
> usethis::git_sitrep()
Git config (global)
• Name: 'juliaparish'
• Email: 'jparish@bren.ucsb.edu'
• Global (user-level) gitignore file: '/Users/julia/.gitignore'
• Vaccinated: FALSE
  i See `?git_vaccinate` to learn more
  i Defaulting to 'https' Git protocol
• Default Git protocol: 'https'
• Default initial branch name: <unset>
GitHub
• Default GitHub host: 'https://github.com'
• Personal access token for 'https://github.comhttps' GitHub protocol
  i GitHub user: 'juliaparish'
```

For more information on PATs, check out GitHub's PAT information page.

Chapter 5

Project Management Tools

5.1 UCR VPN - GlobalProtect

UC Riverside provides free access to a Virtual Private Network (VPN), GlobalProtect, that allows you to access the CCB NAS Server or campus resources (ex: library services) when working remotely.

To install and use the GlobalProtect VPN, navigate to the UCR ServiceLink GlobalProtect Connection Installation Instructions.

5.2 Google Apps

The CCB team utilizes a number of Google applications for project management, data & file sharing, and communicating with colleagues.

Google Calendar: There is a lab Google calendar, *UCR CCB Palm Desert*. This calendar maintains lab mates work and leave schedules as well as any team meetings. Please make sure this calendar is shared with you once you have obtained your UCR NetId.

Google Drive: CCB utilizes *Shared Drive* folders as a file sharing application. You must request access to the shared drive folder from the lab P.I.

Google Sites: There is a CCB Google Site that is in draft form at the moment. This site may serve as an intranet or provide public facing information in the future.

5.3 Microsoft 360 Suite

UCR provides free access to Microsoft Office 365 Pro Plus for faculty, staff, and students! Please note that the available applications differ between PC and Mac OS. To learn more about downloading Microsoft applications, please visit the UC Riverside Microsoft 360 information page.

Microsoft 360 training resources can be found [here](#).

5.4 Trello



Trello is a project management app tool that provides teams the opportunity to create task lists, reference & resource lists, and communicate via tagging! To connect with CCB's Trello Workspace, send a request to CCB's PI. The main board is **CCB Team Projects**.

5.5 Zotero

The Center for Conservation Biology shares research references through Zotero software. Zotero is a citation management tool and allows groups to collect and share references easily. The Zotero browser extension makes it easy to copy citations from webpages.

To install Zotero and the extension, please follow UC Riverside's installation instructions [here](#). *Note:* These instructions are for a **Windows** machine, but it is a similar process for Macs.

Install Zotero

If you have never installed Zotero, or if you are transitioning from the Firefox-embedded version, follow these steps:

1. Go to [Zotero.com](https://www.zotero.com) and click on

[Download](#)

3. Download **Zotero for Windows**. Select the blue download button, and save the file (if necessary).



4. Run the install executable wherever it downloaded to.

Install Connectors

For each browser you use, you would want to install a connector, so that you can e

For each browser (all at once or when you are next using it):

1. Go to Zotero.com



8. When prompted click **add extension, or allow or whatever prompting your browser**

The next step is to create a Zotero account and enable syncing to enable extension use and to be able to join the CCB Zotero Reference Group. Make sure to use your UCR email address when setting up the Zotero account.

Syncing Your Account

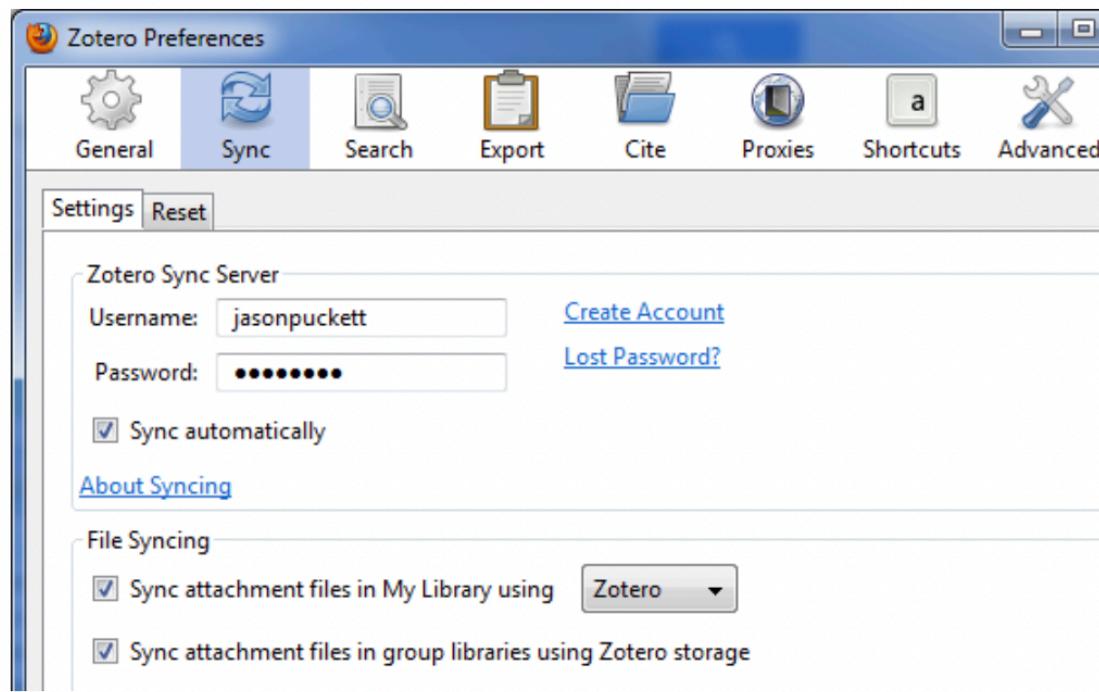
If you're regularly using more than one computer in your research, Zotero's sync feature can keep your library up to date on all copy of your library on the Zotero.org server and check it for updates whenever you open your library on a different computer. A running the same version of Zotero.

First, set up a (free, of course) [Zotero.org user account](#). Then:

- Open Zotero preferences (via the gear menu) and select the Sync tab.
- Enter your Zotero user name and password.
- Check the "sync automatically" box.
- Check both boxes under File Syncing and choose Zotero storage for My Library. This will sync your PDF attachments and
- Click the green circular arrow button at the top right corner of the Zotero window.
- Zotero will upload your library to the server.

Repeat this configuration on each of your computers. Any updates you make on one of your computers will be reflected on the synchronize your library among Windows, Mac and Linux computers.

For more details and help troubleshooting sync problems, check the [Zotero site](#).



To connect to the CCB Zotero Reference Group, navigate to this link. If you do not have access to the Group page, please send a request that you be added to the CCB PI.

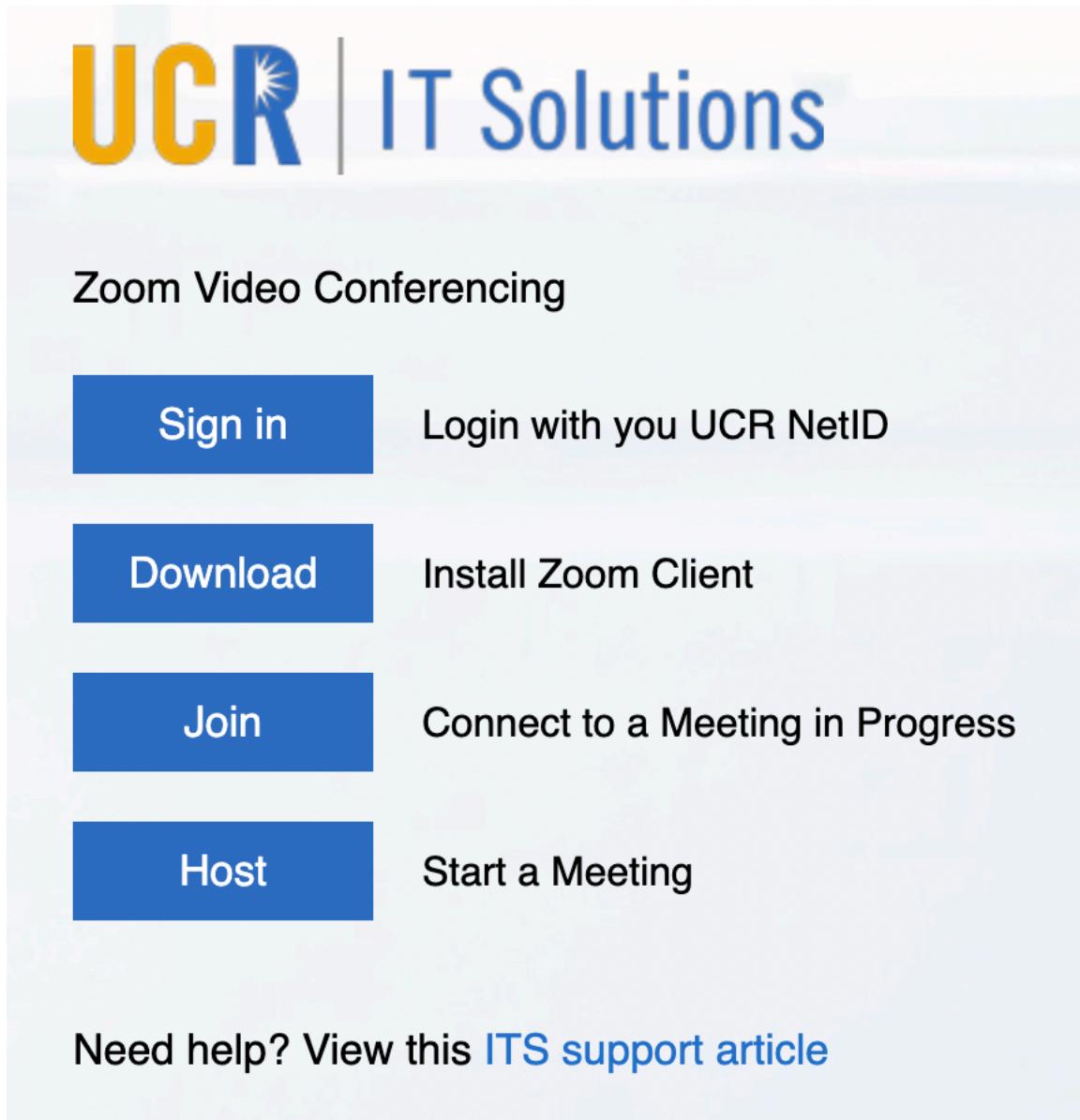
For more information on how to use Zotero, check out these links:

- UC Riverside Zotero Tips & Tricks
- Zotero Quick Start Guide
- Zotero Groups
- RStudio Citations - add BibTex citation format in Zotero then quickly use .bib files in R to create references in markdown.

5.6 Zoom

Zoom is a great tool to schedule virtual meetings, and the CCB hosts Zoom-based team meeting regularly. UC Riverside provides all faculty, staff, and students a free premium account. In order to utilize this service, download the application and sign in via Single Sign-ON (SSO). For the most current installation and log in information, visit UC Riverside's Information Technology Systems Zoom account page.

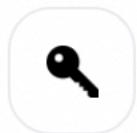
- **Step 1:** Download the Zoom application via UC Riverside's Zoom service page - just select the **Download** button!



The image shows the UCR IT Solutions Zoom Video Conferencing landing page. At the top is the UCR logo and the text "IT Solutions". Below this, the heading "Zoom Video Conferencing" is displayed. There are four main buttons arranged vertically: "Sign in" (with the subtext "Login with your UCR NetID"), "Download" (with the subtext "Install Zoom Client"), "Join" (with the subtext "Connect to a Meeting in Progress"), and "Host" (with the subtext "Start a Meeting"). At the bottom, there is a call-to-action: "Need help? View this [ITS support article](#)".

- **Step 2:** Once the application is downloading, Zoom will automatically open. You will be prompted to either *Join a Meeting* or *Sign In*. Select *Sign In* and set up your UCR Zoom account using your NetId information.
- **Step 3:** Click the *Sign In with SSO* button, then enter `ucr` in the *company domain* section.

Or sign in with



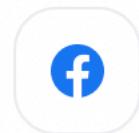
SSO



Apple



Google



Facebook

Zoom is protected by reCAPTCHA and their [Privacy Policy](#) and [Terms of Service](#) apply.



Sign In with SSO

Company Domain

ucr

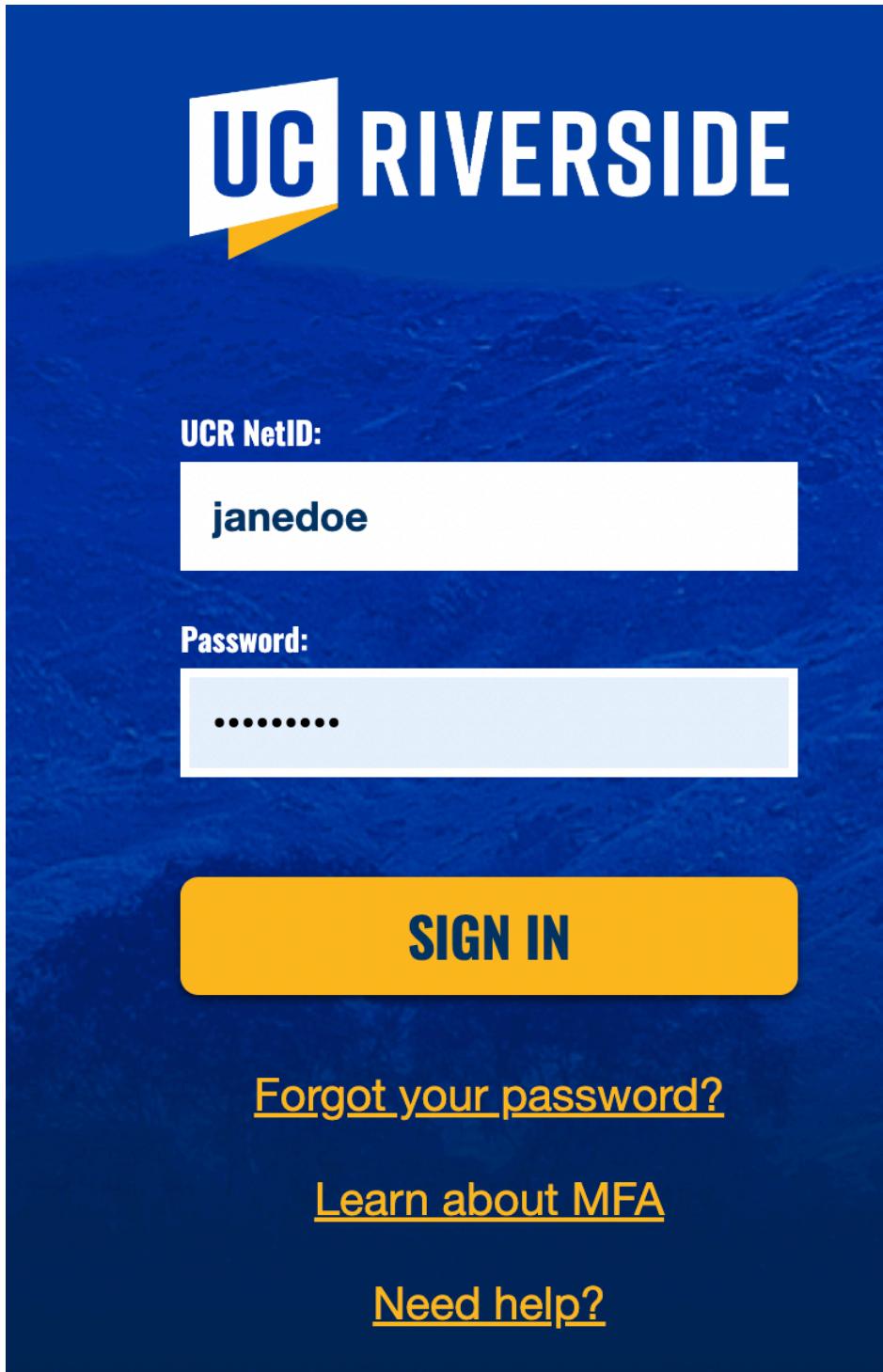
.zo

Continue

[I don't know the company domain](#)

< Back

- **Step 4:** You will then be prompted to sign in, this time using your UCR NetId. The program may open up an authentication window requesting your UCR NetId as shown below:



- **Step 5:** Personalize your Zoom account! You can add your name, pronouns, and an image to your profile, create a *personal Zoom room* to utilize as a default meeting space, and set settings like *mute all participants when they join a meeting* or turn off settings such as *start meetings with participant video on*. *Participants can change this during the meeting.*

Happy Zooming!

Zoom Resources:

- Zoom Help Desk
- Zoom meetings trainings
- UC Riverside Zoom Services

5.7 Slack



Slack is a messaging app that connects teams via channels and direct messages (DMs). To make the most of this team communication tool, you will need to download the application to your desktop, have the application open throughout the workday, and adjust the notification settings to your preferences to receive timely alerts.

5.7.1 Download the Slack app

- For Windows: download Slack here.
- For Macs: download Slack here.
- Consider adding the app to your Android or iOS device(s) for mobile communication with the CCB team!

5.7.2 Join the CCB_UCR Slack Workspace

Follow this link to join the CCB Slack workspace. Once you have joined, you will be automatically be added to several workspace channels:

- **#general:** a channel for general announcements and team-wide conversations

- **#how_to_slack:** a channel for tips for how to best use Slack and its apps, as well as a space for the team to ask questions re: how to best utilize this communication tool. Check out the **pinned** messages and **bookmarked** websites.
- **#meetings:** a channel for announcing meetings and sharing resources for team meetings.
- **#random:** a channel to share random news, updates, ideas, jokes, etc.

5.7.3 Slack Resources

The Slack help center has useful *how to* guides and tutorials to facilitate working in Slack.

Slack Trainings

Check out Slack's **Top 5 tips for getting started in Slack** (5 min read) to learn the app's basic functions and communication workflows.

The Slack help center's "how to" guide has several sections:

- Quick Reference Guide (20 min read)
- Using Slack: recommended Sections to review
 - Format & style messages (5 min read)
 - Message features & tools (15 min read)
 - Audio & video (10 min read)
- Your Profile
 - Adjust your notifications: configure your notifications (5 min read)
 - Change your settings & preferences:
 - * set your Slack status & availability (3 min read)
 - * adjust your sidebar preferences (3 min read)
- Connect Tools
 - Connect tools from the Slack app directory: add apps to your Slack workspace (3 min read)
- Tutorials - a long list of tutorials to review as needed and as time allows

Slack Apps

Up to 10 Slack apps are available for free Slack accounts, and unlimited apps are available for Pro Slack accounts.

- A **Trello app** is incorporated in the CCB_UCR Slack workspace. You can create cards in Slack for the CCB Team Projects Trello workspace. To learn how, follow this Slack connect tool.

- The **Zoom app** can be added to Slack that allows you to start or join a Zoom Meeting or Zoom Phone call by using the “/zoom” slash command in any Slack channel or message.
- Add the **Google Drive app** to Slack to share resources from the CCB Shared Drive.
- There is an app for **Microsoft Teams** which allows a Teams video call from Slack.
 - Start a Teams video call right from Slack Launch a call in Slack with the /teams-calls slash command. Before joining, get a quick glance at who's already on the call and when the call kicked off.
 - Jump from Slack straight into a meeting — join Teams meetings directly from calendar reminders in Slack, using the Outlook or Google Calendar apps.
 - Customize call settings for your team. Set Microsoft Teams Calls as your default calling provider, so anyone can start a video call in Teams with a quick click of the phone button in Slack.

Other cool Slack workspaces to join:

- EcoDataScience - an environmental data science study group that started at UC Santa Barbara, but now has an international following!
- R4DS Online Learning Community
- Society for Open, Reliable, and Transparent Ecology and Evolutionary Biology (SORTEE)

Chapter 6

ESRI GIS Platform

6.1 ESRI ArcGIS

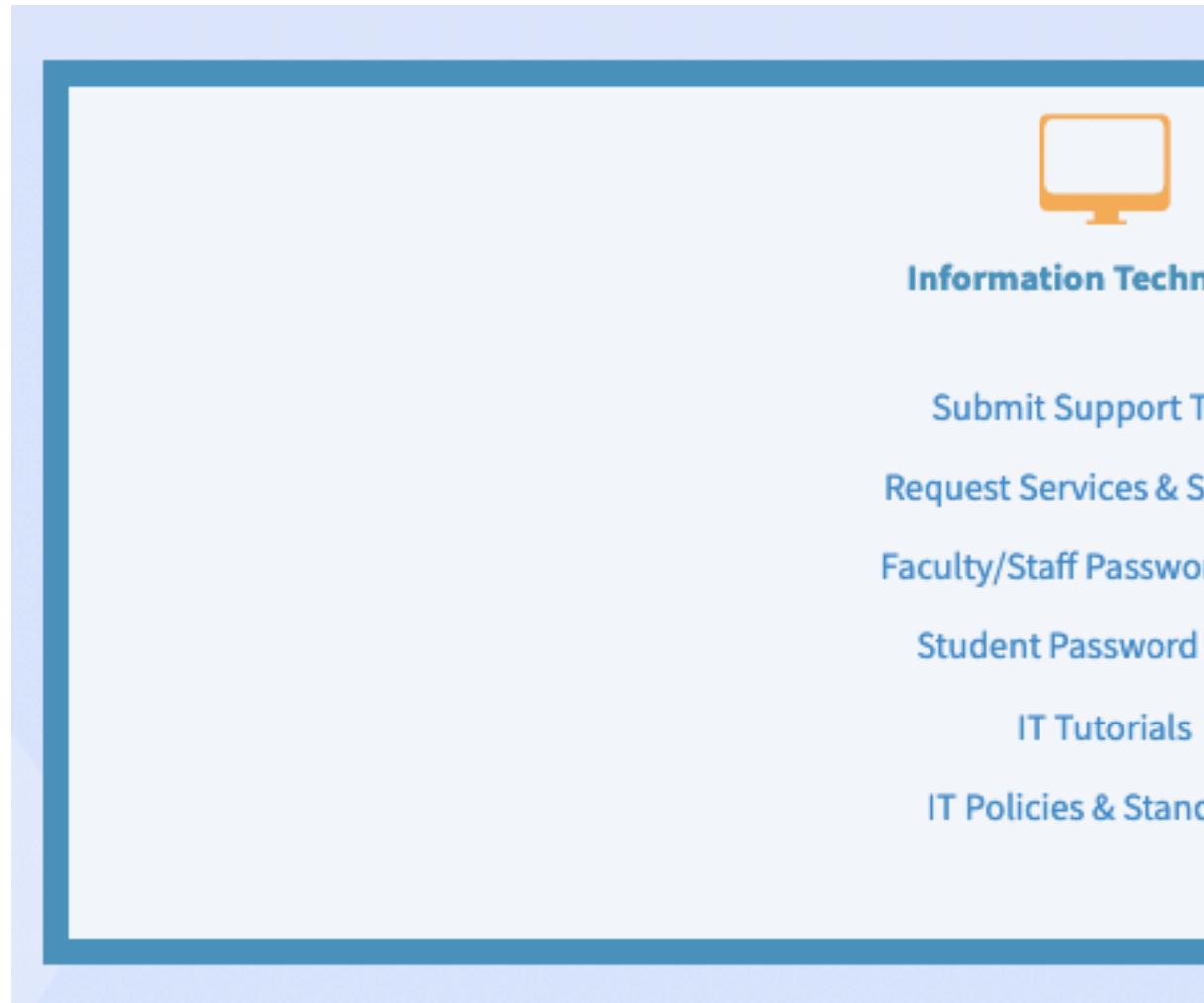
UC Riverside provides free access to ESRI ArcGIS software and platforms for faculty, staff, and students. Learn more about these data analysis and visualization tools here.

ESRI Platform Software

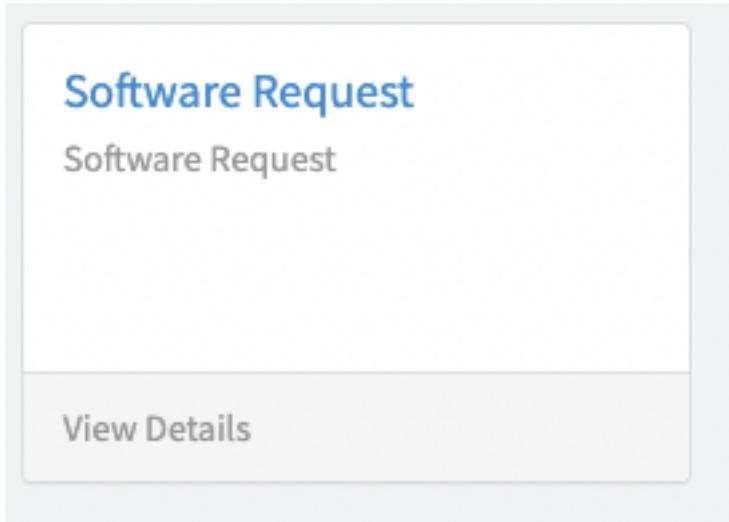
- ESRI ArcGIS Desktop: a proprietary desktop GIS software suite that allows you to create maps, perform spatial analysis and manage data. Required software for installation: Microsoft .NET (aka a PC machine), Python 2.7+.
- ESRI ArcGIS Pro: ArcGIS Desktop is transitioning to ArcGIS Pro, an interactive spatial data visualization tool that allows users to create 2D maps, 3D scenes, and share projects on ArcGIS Online. Training resources for ArcGIS Pro are here.
- ESRI ArcGIS Online: This is cloud-based software that runs on any device with an internet connection. It allows users to easily publish maps on-line, access other UCR spatial products, and create spatial data dashboards and hubs. Check out UCR ArcGIS Online.

6.1.1 Install ESRI ArcGIS software

- **Step 1:** Navigate to the UCR Portal ServiceLink Software Catalog.
- **Step 2:** Under the Information Technology section, select **Request Services & Software** link.



- **Step 3:** Scroll down to the **Software Request** card and select link.



- **Step 4:** Complete the Software Request form using your UCR NetId and UCR email, then select **Submit**.

Software Request

Software Request

* Requestor (If requesting on behalf of another person, please enter name of individual)

NetID

Email

* Which Software

ArcGIS

Additional Information

ArcGIS Desktop and ArcGIS Pro

- **Step 5:** The UC IT team will send email notifications to provide further instructions.

6.1.2 ESRI Resources

Here are few resources that provide tutorials and workshops on how to use ESRI software:

- **ESRI Academy:** Search the ESRI Acadmey Course Catalog for MOOCs, Tutorials, and Web Courses.
- **UC ANR IGIS** - the UC Agriculture and Natural Resources Informatics & GIS Program host ESRI ArcGIS training webinars and virtual workshops throughout the year.
- **LinkedIn Learning** - Through UCR LinkedIn Learning, users may take ESRI ArcGIS courses and learning paths.

Chapter 7

File Directory Organization

7.1 File Naming

When creating file names, best practices dictate that you should:

- Create meaningful but brief names;
- Use file names to classify types of files (ex: DTLA_RA_20230131.csv);
- Avoid using spaces, dots and special characters (& or ? or !);
- Use capital letters to delimit words, not spaces or underscores for data files, *as much as possible* (ex: CVStormwaterVAP20170217.csv);
- Use hyphens (-) or underscores (_) to separate elements in a file name for documents;
- Always use the ISO 8601 date format in file names: **YYYYMMDD**. For example, in the file name *DTLA_RA_20230131.csv* January 31, 2023 is represented as **20230131**;
- Preserve the 3-letter file extension for application-specific codes of file format (e.g. .doc, .xls, .mov, .tif);
- With sequential numbering (e.g., 1, 2, 3, etc.), use leading zeros to accommodate multi-digit versions. For example, use 01-10 for 1-10, 001-100 for 1-100, and so on;
- Include versioning of file names where appropriate (ex: filename_rev20230131.doc).
- Order the elements in a file name in the most appropriate way to retrieve the record. For example,

- the project name (or acronym) or project location should be first,
- then the type of data or file (either RA or Report),
- then version or author,
- end file name with the date,
- Examples: *CVStormwaterVAP20170217.csv*, *CVMC_CRCA01_Proposal_LCS_20221231.docx*

7.2 Folder Organization

7.3 Data Management

Chapter 8

Equipment

Inventory of software and equipment used and references

- ARC MAP DESKTOP: ArcMap ArcGIS Desktop, version 10.9 (Esri, Redlands, California)
- LASER RANGEFINDER: Hilti PD-42 laser rangefinder (Hilti Corporation, Schaan, Liechtenstein)
- TABLETS: Samsung Galaxy Tab Active 2 tablet (Samsung Group, Seoul, South Korea)
- GPS: Trimble Juno 3

Chapter 9

Code of Conduct

The Center of Conservation Biology's Code of Conduct adapted from (Perry, 2018)

9.1 We are committed to working as a team.

All aspects of our professional contributions to the project are discussed and agreed upon together, and all tasks - although they might be led by individual team members - are developed through collaborative practice. Devotion to supporting the team, working as a team player, providing constructive critique to your team members, and respecting the interests of the team as a successful working group (without compromising their safety or security, as described below), are paramount.

9.2 We are committed to prioritizing and championing the people and communities that host us.

Our work is driven by local needs, and decision-making is grounded in evidence and robust data gathered in local contexts. We are critically aware of the existing evidence. We attend events and participate in activities that are organised by our host communities. We respect, care for and create long-lasting friendships with our hosts. We aim to abide by local expectations around dress and custom, and if working in communities where the primary language is not our own, we are committed to learning the language. We maintain links with our hosts after the project ends and we support their future professional endeavors.

9.3 We are committed to the working hours, professional expectations and responsibilities defined by the overall project directors.

We typically work as part of a larger project team guided by wider goals than ours alone. We are aware of their responsibilities, we have read the necessary guidance documents, we have understood and signed the necessary insurance and risk assessment documentation, and in all cases, we respect and abide by the instructions given by the directors. This includes zero tolerance in relation to behavior that compromises the well being, equality, security or dignity of other human beings, as described below.

9.4 We are representatives and extensions of the University of California, Riverside and its staff, and of the professional bodies to which we and our project leaders are subscribed.

We recognize our duty of care to, and our responsibility for professionalism in, not only the communities where we work and reside, but the university and surrounding organisations to which we and our project leaders are accountable. Our behaviors reflect on these institutions and we acknowledge that our direct supervisor is (and therefore we too are) bound by the ethical and professional codes of both UC Riverside and other institutional affiliations.

Considering these obligations, you agree with the following:

- I will come to my direct supervisor the moment that I experience problems, challenges or trouble of any kind. I will keep her informed of any issues that I feel may manifest themselves in relation to myself, my teammates or affiliates while in the field. If I feel I need support beyond my direct supervisor, I will turn to the 2nd identified lead for their advice. I have already disclosed to my direct supervisor any potential matters of concern (which may include matters relating to health, psychological and physical wellbeing, security, equality, confidence, interpersonal relations, previous travel or fieldwork experiences, etc.) so that they are aware of them and can mitigate them prior to departing for - and during - fieldwork. If I have not yet disclosed such matters, I agree to do so as soon as possible. I have shared this information in confidence, with an expectation of complete privacy unless urgent medical, safety, security, or other legal intervention is required.

9.5 We recognize that fieldwork can be intense, emotional and tiring.

We understand that things can go wrong, that we may need to compromise, and that in exceptional circumstances, we made need to shorten or modify your work on site to help manage these circumstances. In such cases, we will have a series of conversations about how to deal with difficulties, led by your direct supervisor and/or other lead. If the difficulties are not resolved within 7 days of identification, we will consult with the university for their guidance. If it is agreed with the University that the difficulties are unresolvable in the field, we will help you to organize your safe return home.

9.6 We have the right to a safe, secure and non-threatening working and living environment.

We do not tolerate any form of discriminatory, abusive, aggressive, harassing, threatening, sexually- or physically-intimidating, or related problematic behaviors that compromise the wellbeing, equality, security or dignity of other human beings (whether those humans are our peers, colleagues, supervisors, collaborators, local community members or any persons at all). Our supervisors are trained in supporting those who have experienced or are experiencing harassment. They are obliged to investigate and respond to observed, implied or directly reported harassment. Considering this zero-tolerance policy, you agree to the following:

- I will not engage in behavior that compromises the wellbeing, equality, security or dignity of other human beings. I recognize that if I am implicated in such behavior I will be required to leave the project at my own expense and may be subject to criminal investigation.

If I witness others being subjected to such behavior, I will report it immediately to my direct supervisor. If I feel I cannot speak to my direct supervisor, I will report it to the 2nd identified lead. If I feel I cannot report it to either my direct supervisor or the 2nd lead, then I will contact the University of California, Riverside Compliance office.

- If I myself feel unsafe or uncomfortable, I will report it immediately to my direct supervisor. My supervisor will support me and will implement actions to keep me safe while working to stop the behavior. If I feel I cannot speak to my direct supervisor, I will report it to the 2nd identified lead. If I feel I cannot report it to either my direct supervisor or the 2nd

lead, then I will contact the University of California, Riverside Compliance office.

- My commitment to creating and maintaining safety and security for all extends to my online (web and social media) and mobile phone interactions, and I recognize that the process for reporting and acting on threatening online/mobile phone behaviors is the same as above.

9.7 Important University of California, Riverside Links

- UCR Values
- Ethics & Compliance Program
- Equal Opportunity and Affirmative Action
- Discrimination, Harassment, and Retaliation Complaint and Resolution
- Standards of Conduct
- Environmental Health & Safety
- Alcohol and Drug Policy
- Personal Relationship Policy

Bibliography

Perry, S. (2018). Six fieldwork expectations: Code of conduct for teams on field projects.

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2022). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.28.

Yıldırım, S. (2020). What is git and why is it so important?