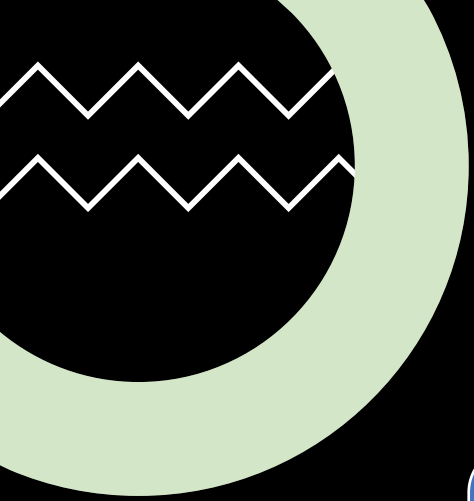




Using R in a Docker Container

An introduction to Rocker for FISH 549





Hello!

Dara Farrell

- Feel free to say hi if you see me around!

Currently a postdoc in the Applied Ecology lab/Scheuerell lab

Here to share some of what I've been learning about Docker





Outline



What is Docker?

Using R in Docker

Rocker and definition of some terms

Prerequisites for tutorial

Overview

Explanation



Examples:

Applying knowledge to a past FISH 549 assignment

What is Docker?



Docker helps avoid the “it works on my machine” problem.

It makes sharing code and coding environments easier, enhances security, and minimizes the headache of package conflicts.

- You could have two different versions of Python running on your machine, at the same time.
- You can run RStudio without even installing R or RStudio on your hard drive.

What I mean when I say



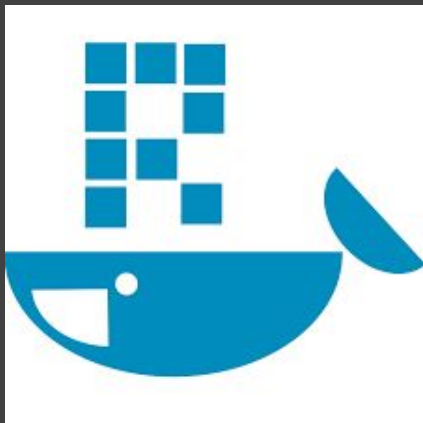
Docker image: contains **application code, libraries, tools, dependencies and other files needed to make an application run.**

Docker container: When a user runs an image, it can become one or many instances of a container. **A container is a running image.**

Dockerfile: a **text document** that contains all the **commands** a user could call on the command line to assemble an image



R + Docker =
Rocker

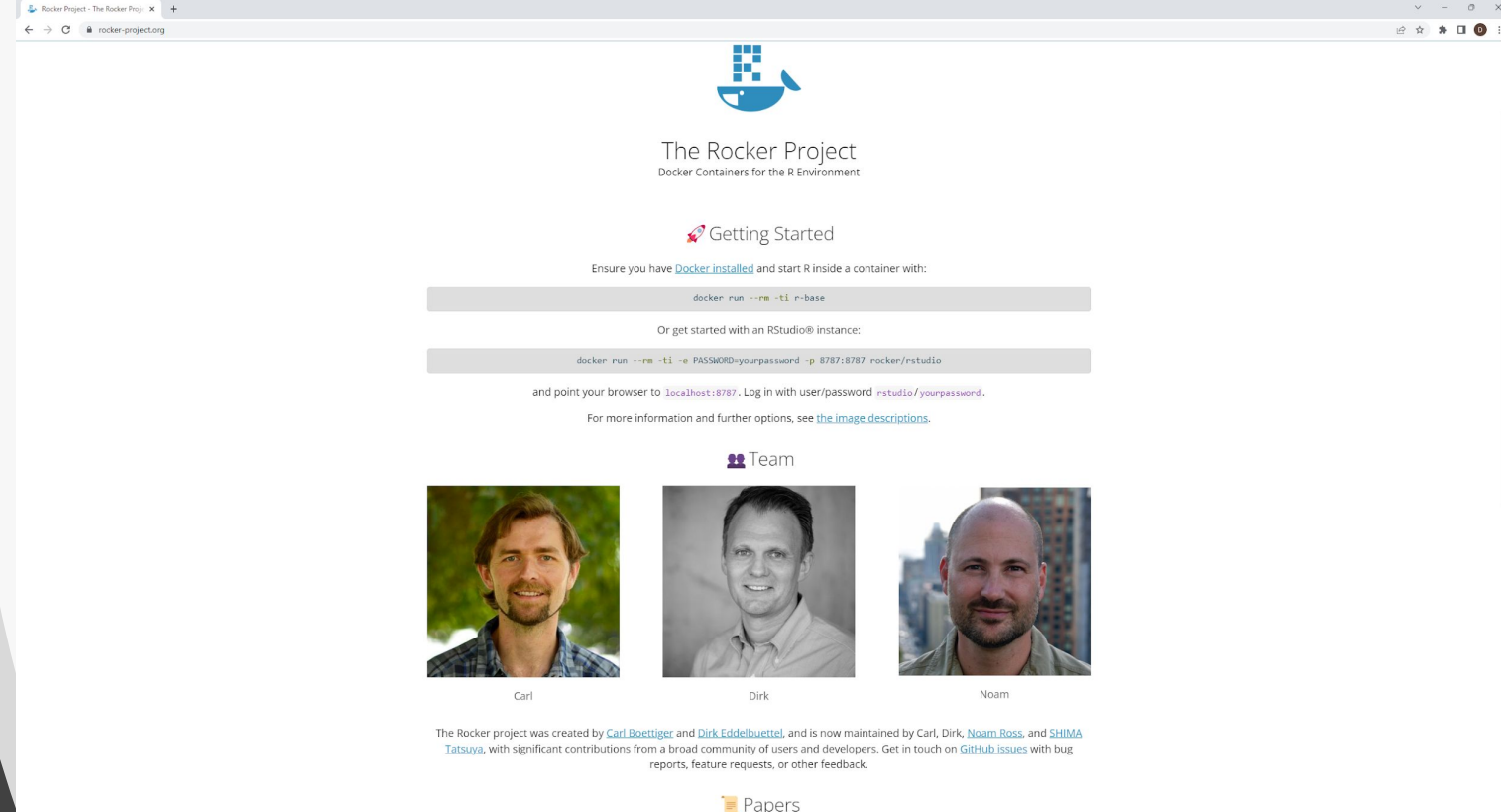


Rocker is a set of tools to simplify running Docker containers as non-root users.

The maintainers of the project update the images as new security vulnerabilities or new versions of R are released.

Rocker rocker-project.org

- Relevant papers:
 - [An Introduction to Rocker: Docker Containers for R](#)
 - [The Rockerverse: Packages and Applications for Containerisation with R](#)
- Instructions for the command line are available



The screenshot shows the homepage of the Rocker Project. At the top, there is a logo consisting of a blue square with a white grid pattern and a blue circle below it. The text "The Rocker Project" is centered, with the tagline "Docker Containers for the R Environment" underneath. Below this, a "Getting Started" section features a rocket icon and instructions on how to run Docker containers. Two code blocks are provided: one for a basic R environment and another for an RStudio instance. A link to "image descriptions" is also present. The "Team" section follows, showing three headshots of the project's creators: Carl, Dirk, and Noam. A paragraph of text describes the project's history and community involvement. At the bottom, there is a "Papers" section with a book icon.

The Rocker Project
Docker Containers for the R Environment

Getting Started

Ensure you have [Docker installed](#) and start R inside a container with:

```
docker run --rm -ti r-base
```

Or get started with an RStudio® instance:

```
docker run --rm -ti -e PASSWORD=yourpassword -p 8787:8787 rocker/rstudio
```

and point your browser to [localhost:8787](#). Log in with user/password `rstudio/yourpassword`.

For more information and further options, see [the image descriptions](#).

Team

Carl Dirk Noam

The Rocker project was created by [Carl Boettiger](#) and [Dirk Eddelbuettel](#), and is now maintained by Carl, Dirk, [Noam Ross](#), and [SHIMA Tatsuya](#), with significant contributions from a broad community of users and developers. Get in touch on [GitHub issues](#) with bug reports, feature requests, or other feedback.

Papers

Pre-requisites for this tutorial



Docker Desktop (for Windows or macOS)

- You do not need to have R or RStudio installed on your computer (the machine that I'm using for this tutorial doesn't have R or RStudio installed)

Develop faster. Run anywhere.

The most-loved Tool in Stack Overflow's 2022 Developer Survey.

Download Docker Desktop

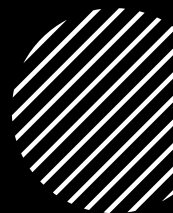
Windows



WHAT'S NEW



Demo



Overview: Quickstart



Step by step

Files for this tutorial are
on the FISH 549
website



The Rocker Project

Docker Containers for the R Environment

Getting Started

Ensure you have [Docker installed](#) and start R inside a container with:

```
docker run --rm -ti r-base
```

Or get started with an RStudio® instance:

```
docker run --rm -ti -e PASSWORD=yourpassword -p 8787:8787 rocker/rstudio
```

and point your browser to `localhost:8787`. Log in with user/password `rstudio/yourpassword`.

For more information and further options, see [the image descriptions](#).

Command	Description
<code>`docker pull IMAGE`</code>	Download a Docker image from a registry
<code>`docker build [OPTIONS] PATH`</code>	Build a Docker image from a Dockerfile
<code>`docker push IMAGE`</code>	Push a Docker image to a registry
<code>`docker stop CONTAINER`</code>	Stop a running Docker container
<code>`docker rm CONTAINER`</code>	Remove a stopped Docker container
<code>`docker rmi IMAGE`</code>	Remove a Docker image
<code>`docker-compose up`</code>	Start all services defined in a Docker Compose file
<code>`docker-compose down`</code>	Stop all services defined in a Docker Compose file
<code>`docker exec CONTAINER COMMAND`</code>	Run a command inside a running Docker container

Command	Description	Components
<code>`docker run --rm -ti r-base`</code>	Run an interactive R programming environment container	<code>`docker run`</code> : command to run a Docker container, <code>`--rm`</code> : automatically remove container when it exits, <code>`-ti`</code> : run container interactively with a pseudo-TTY terminal, <code>`r-base`</code> : name of the Docker image to use
<code>`docker run --rm -ti -e PASSWORD=yourpassword -p 8787:8787 rocker/rstudio`</code>	Run an interactive RStudio Server container	<code>`docker run`</code> : command to run a Docker container, <code>`--rm`</code> : automatically remove container when it exits, <code>`-ti`</code> : run container interactively with a pseudo-TTY terminal, <code>`-e PASSWORD=yourpassword`</code> : set environment variable <code>`PASSWORD`</code> to <code>`yourpassword`</code> , <code>`-p 8787:8787`</code> : map port 8787 on host to port 8787 in container, <code>`rocker/rstudio`</code> : name of the Docker image to use
<code>`docker run [OPTIONS] -v /path/on/host:/path/in/container IMAGE [COMMAND]`</code>	Run a container with a mounted volume	<code>`docker run`</code> : command to run a Docker container, <code>`[OPTIONS]`</code> : additional options for the container (e.g., <code>`-d`</code> to run in detached mode), <code>`-v /path/on/host:/path/in/container`</code> : mount a volume from the host to the container, <code>`IMAGE`</code> : name of the Docker image to use, <code>`[COMMAND]`</code> : optional command to run inside the container
<code>`docker ps`</code>	List all running containers	<code>`docker ps`</code> : command to list running containers
<code>`docker images`</code>	List all Docker images	<code>`docker images`</code> : command to list Docker images
<code>`docker build -t fish549_container fish549_image`</code>	Build a Docker container from an image	<code>`docker build`</code> : command to build a Docker container, <code>`-t fish549_container`</code> : tag the container with the name <code>`fish549_container`</code> , <code>`fish549_image`</code> : name of the Docker image to use as the base image

```
Select Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows


Loading personal and system profiles took 1081ms.
(base) PS C:\Users\Dara-SAFS>
```

Docker Desktop Upgrade plan Search for local and remote images, containers, and more... **Ctrl+K** Sign in

- Containers [Give feedback](#)
- Images
- Volumes
- Dev Environments **BETA**
- Extensions **+**
- Add Extensions

Containers

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)





Run a Sample Container

Try running a container: Copy and paste this command into your terminal and then come back

```
docker run -d -p 80:80 docker/getting-started
```

[Explore more in the Docker Docs](#)

Guides

-  **Redis** [📄](#) [🔔](#)
An open-source in-memory key-value store that functions as a data structure server.
-  **NGINX** [📄](#) [🔔](#)
An open-source web server, reverse proxy, load balancer and HTTP cache.

RAM 11.21 GB CPU 0.04% Not connected to Hub v4.17.0

- Containers
- Images**
- Volumes
- Dev Environments BETA
- Extensions
- Add Extensions

Images [Give feedback](#)

An image is a read-only template with instructions for creating a Docker container. [Learn more](#)

Local Hub

1.82 GB / 2.45 GB in use 2 images

Last refresh: about 19 hours ago

Search

<input type="checkbox"/>	Name	Tag	Status	Created	Size	Actions
<input type="checkbox"/>	rocker/rstudio e66f6ea0cc2a	latest	in use	3 days ago	1.82 GB	
<input type="checkbox"/>	r-base 3de1ef2039fb	latest	Unused	4 months ago	838.21 MB	

Sign in to RStudio

Username:

Password:

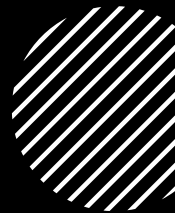
Stay signed in when browser closes

You will automatically be signed out after 60 minutes of inactivity.

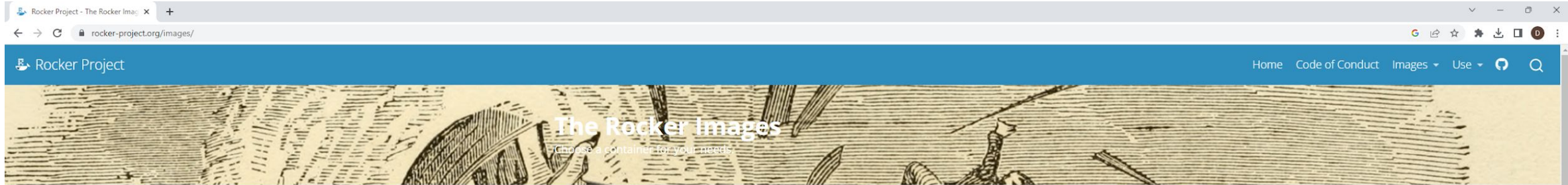
Sign in



Steps



- Docker commands
- Dockerfile
- Building the image
- Building the container
- Renaming the image and container
- Checking our work then running the container
- Entering a username and password
- Using RStudio!



- The Rocker Images versioned stack
 - r-ver
 - rstudio, tidyverse, verse, geospatial
 - binder
 - shiny, shiny-verse
 - cuda, ml, ml-verse
- base stack
 - r-base
 - rocker/r-base
 - r-devel, drd, drp, r-devel-san, r-devel-ubsan-clang
- Additional images
 - r-ubuntu
 - r-bspm
- Dev Container
- Features
- Images
- Templates
- Other projects
- External images and tools for R

1 Overview

The Rocker Project provides a collection of (Linux) containers suited for different needs. Find a base image to extend or images with popular software and optimized libraries pre-installed.

Get the latest version or a reproducibly fixed environment.

2 Images

2.1 The versioned stack

image	base image	description	pulls
rocker/r-ver	ubuntu	Install R from source and set RSPM as default CRAN mirror	docker pulls: 3,731
rocker/rstudio	rocker/r-ver	Adds RStudio Server	docker pulls: 6,881
rocker/tidyverse	rocker/rstudio	Adds tidyverse packages & devtools	docker pulls: 5,724
rocker/verse	rocker/tidyverse	Adds tex & publishing-related package	docker pulls: 1,724
rocker/geospatial	rocker/verse	Adds geospatial packages	docker pulls: 6724
rocker/binder	rocker/geospatial	Adds requirements to run repositories on mybinder.org	docker pulls: 894
rocker/shiny	rocker/r-ver	Adds shiny server	docker pulls: 2,114
rocker/shiny-verse	rocker/shiny	Adds tidyverse packages	docker pulls: 6,344
rocker/cuda	rocker/r-ver	Adds CUDA support to rocker/r-ver	docker pulls: 174
rocker/ml	rocker/cuda	Adds CUDA support to rocker/tidyverse	docker pulls: 574
rocker/ml-verse	rocker/ml	Adds CUDA support to rocker/geospatial	docker pulls: 254

This stack builds on stable Debian releases (for R versions $\leq 3.6.3$) or Ubuntu LTS (for R versions $\geq 4.0.0$). Images in this stack accept a version tag specifying which version of R is desired, e.g. `rocker/rstudio:4.0.0` for R 4.0.0. The `latest` tag always follows the latest release version of R.

On this page

- 1 Overview
- 2 Images

View source
Edit this page



rocker/verse ☆

↓ Pulls 1M+

By [rocker](#) • Updated 2 hours ago

Adds tex & related publishing packages to version-locked tidyverse image

Image

Overview

Tags

Sort by

Newest ▾

Filter Tags



TAG

[latest](#)

Last pushed a month ago by [cboettig](#)

docker pull rocker/verse:lat...

DIGEST

[0b81a9ea0199](#)

OS/ARCH

linux/amd64

SCANNED

COMPRESSED SIZE ⓘ

1.25 GB

TAG

[devel](#)

Last pushed 2 hours ago by [cboettig](#)

docker pull rocker/verse:dev...

DIGEST

[b23dca57cc25](#)

OS/ARCH

linux/amd64

SCANNED

COMPRESSED SIZE ⓘ

1.25 GB

```
File Edit Selection View Go Run Terminal Help
tutorial_fish549

EXPLORER
TUTORIAL_FISH549
  presentation_images_info
  .R.profile
  Dockerfile
  play.R

Dockerfile
1 # Use the rocker/tidyverse:4.2.2 image as the base image
2 FROM rocker/tidyverse:4.2.2
3
4 #Metadata
5 #Prevent dangling images by naming your image and tagging it if you want to track different versions
6
7 # Labels
8 LABEL maintainer="Dara Farrell"
9
10 ## Set the working directory to /the path to the working directory on your machine
11 #WORKDIR C:\Users\Dara-SAFS\Desktop\tutorial_fish549
12 WORKDIR /rstudio/tutorial_fish549
13
14 # Dependencies/packages
15 # Change the password for the default user "rstudio" here and uncomment if you want a visible password in the Dockerfile
16 #ENV PASSWORD=rockon
17
18
19 # Install RSTAN package
20 #https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started
21 RUN R -e "install.packages('rstan', repos=c('https://mc-stan.org/r-packages/', getOption('repos')))"
22 #After the container launch in RStudio run the RStan example/test model to verify the installation in the container
23 #example(stan_model, package = "rstan", run.dontrun = TRUE)
24
25 #Expose ports
26 EXPOSE 8787
27
28 #Runtime commmands (i.e. what you enter at the command line)
29 #To build the image in the current container with the name fish549_image
30 #docker build -t fish549_image .
31
32 ## This next command mounts the directory. It also sets the name of the container to fish549_container. Mounting the volume will allow you to save the results from working in the container on your host machine.
33 # For this option You need to have a.Rprofile file with the following Sys.setenv(PASSWORD = Sys.getenv("PASSWORD"))
34
35 #docker run --rm -it -e PASSWORD=rockon -v <Your path to tutorial_fish549 folder>:/home/rstudio/tutorial_fish549 -p 8787:8787 --name fish549_container fish549_image
36 #e.g. docker run --rm -it -e PASSWORD=rockon -v C:\Users\Dara-SAFS\Desktop\tutorial_fish549:/home/rstudio/tutorial_fish549 -p 8787:8787 --name fish549_container fish549_image
37
38 #Runtime command to mount directory etc. if you have included a visible password in the Dockerfile
39 #docker run --rm -it -v <Your path to tutorial_fish549 folder>:/home/rstudio/tutorial_fish549 -p 8787:8787 --name fish549_container fish549_image
40
41
```

EXPLORER

- TUTORIAL_FISH549
 - presentation_images_info
 - .R.profile
 - Dockerfile
 - play.R

```
Dockerfile > ...
1 # Use the rocker/tidyverse:4.2.2 image as the base image
2 FROM rocker/tidyverse:4.2.2
3
4 #Metadata
5 #Prevent dangling images by naming your image and tagging it if you want to track different versions
6
7 # Labels
8 LABEL maintainer="Dara Farrell"
9
10 ## Set the working directory to /the path to the working directory on your machine
11 #WORKDIR C:\Users\Dara-SAFS\Desktop\tutorial_fish549
12 WORKDIR /rstudio/tutorial_fish549
13
14 # Dependencies/packages
15 # Change the password for the default user "rstudio" here and uncomment if you want a visible password in the Dockerfile
16 #ENV PASSWORD=rockon
17
18
19 # Install RSTAN package
20 #https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started
21 RUN R -e "install.packages('rstan', repos=c('https://mc-stan.org/r-packages/', getOption('repos')))"
22 #After the container launch in RStudio run the RStan example/test model to verify the installation in the container
23 #example(stan_model, package = "rstan", run.dontrun = TRUE)
24
25 #Expose ports
26 EXPOSE 8787
27
28 #Runtime commmands (i.e. what you enter at the command line)
29 #To build the image in the current container with the name fish549_image
30 #docker build -t fish549_image .
31
32 ## This next command mounts the directory. It also sets the name of the container to fish549_container. Mounting the volume will allow you to save the results from working in the container on your host machine.
33 # For this option You need to have a.Rprofile file with the following Sys.setenv(PASSWORD = Sys.getenv("PASSWORD"))
34
35 #docker run --rm -it -e PASSWORD=rockon -v <Your path to tutorial_fish549 folder>:/home/rstudio/tutorial_fish549 -p 8787:8787 --name fish549_container fish549_image
36 #e.g. docker run --rm -it -e PASSWORD=rockon -v C:\Users\Dara-SAFS\Desktop\tutorial_fish549:/home/rstudio/tutorial_fish549 -p 8787:8787 --name fish549_container fish549_image
37
38 #Runtime command to mount directory etc. if you have included a visible password in the Dockerfile
39 #docker run --rm -it -v <Your path to tutorial_fish549 folder>:/home/rstudio/tutorial_fish549 -p 8787:8787 --name fish549_container fish549_image
40
41
```

EXPLORER

TUTORIAL_FISH549

- presentation_images_info
- .Rprofile
- Dockerfile
- play.R

OUTLINE

TIMELINE

Dockerfile play.R .Rprofile

Dockerfile > ...

```
1 # Use the rocker/tidyverse:4.2.2 image as the base image
2 FROM rocker/tidyverse:4.2.2
3
4 #Metadata
5 #Prevent dangling images by naming your image and tagging it if you want to track different versions
6
7 # Labels
8 LABEL maintainer="Dara Farrell"
9
10 ## Set the working directory to /the path to the working directory on your machine
11 #WORKDIR C:\Users\Dara-SAFS\Desktop\tutorial_fish549
12 WORKDIR /rstudio/tutorial_fish549
13
14 # Dependencies/packages
15 # Change the password for the default user "rstudio" here and uncomment if you want a visible password in the Dockerfile
16 #ENV PASSWORD=rockon
17
18
19 # Install RSTAN package
20 #https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started
21 RUN R -e "install.packages('rstan', repos=c('https://mc-stan.org/r-packages/', getOption('repos')))"
22 #After the container launch in RStudio run the RStan example/test model to verify the installation in the container
23 #example(stan_model, package = "rstan", run.dontrun = TRUE)
24
25 #Expose ports
26 EXPOSE 8787
27
28 #Runtime commands (i.e. what you enter at the command line)
29 #To build the image in the current container with the name fish549_image
30 #docker build -t fish549_image .
31
32 ## This next command mounts the directory. It also sets the name of the container to fish549_container. Mounting the volume will allow you to save the results from working in the container on your host machine.
33 # For this option You need to have a .Rprofile file with the following Sys.setenv(PASSWORD = Sys.getenv("PASSWORD"))
34
35 #docker run --rm -it -e PASSWORD=rockon -v <Your path to tutorial_fish549 folder>:/home/rstudio/tutorial_fish549 -p 8787:8787 --name fish549_container fish549_image
36 #e.g. docker run --rm -it -e PASSWORD=rockon -v C:\Users\Dara-SAFS\Desktop\tutorial_fish549:/home/rstudio/tutorial_fish549 -p 8787:8787 --name fish549_container fish549_image
37
38 #Runtime command to mount directory etc. if you have included a visible password in the Dockerfile
39 #docker run --rm -it -v <Your path to tutorial_fish549 folder>:/home/rstudio/tutorial_fish549 -p 8787:8787 --name fish549_container fish549_image
40
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

PS C:\Users\Dara-SAFS\Desktop\tutorial_fish549> |

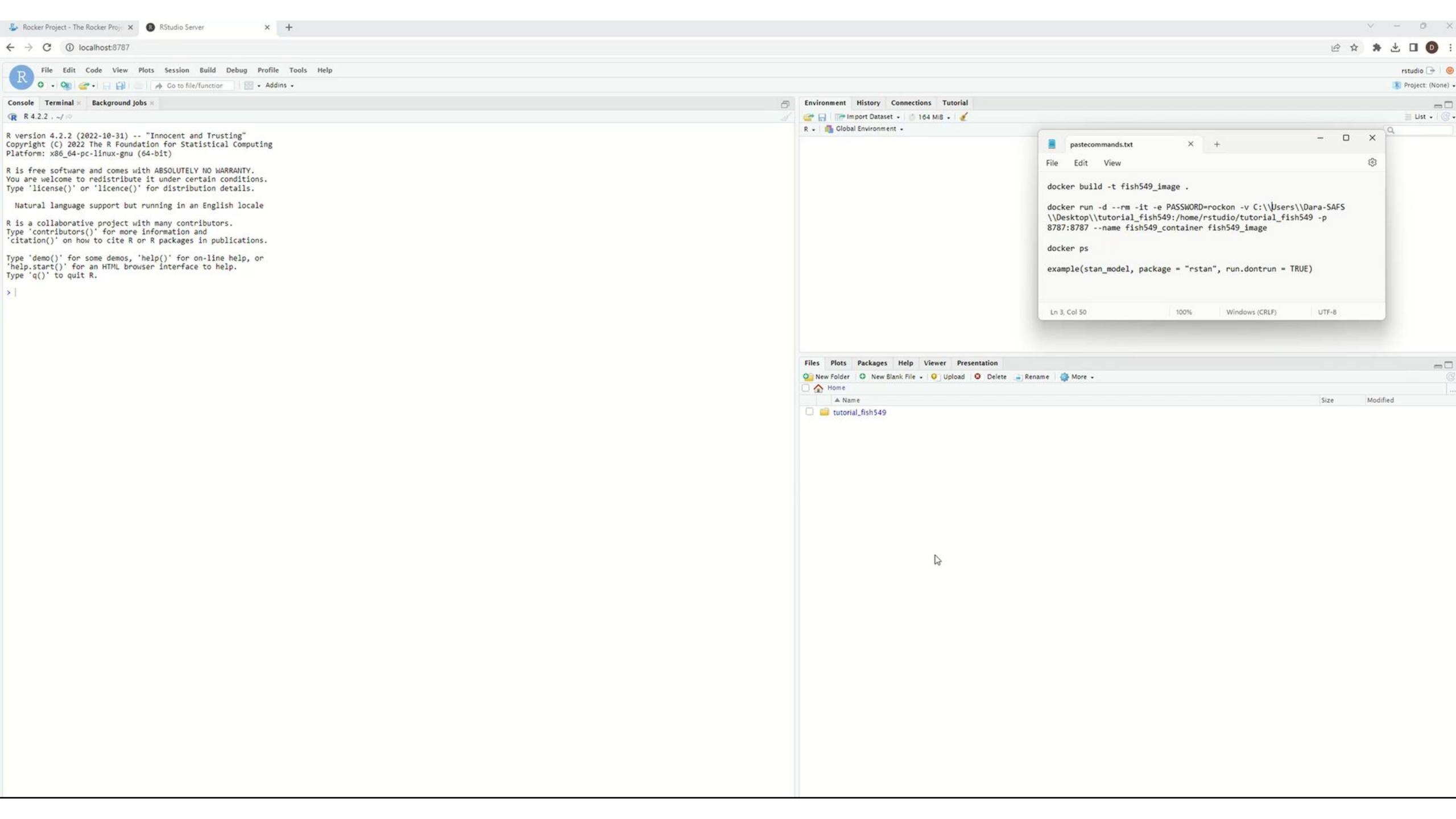
pwsh + - - - ^ x



Examples

Working with the package installed

Applying knowledge to a past FISH 549
assignment



R version 4.2.2 (2022-10-31) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

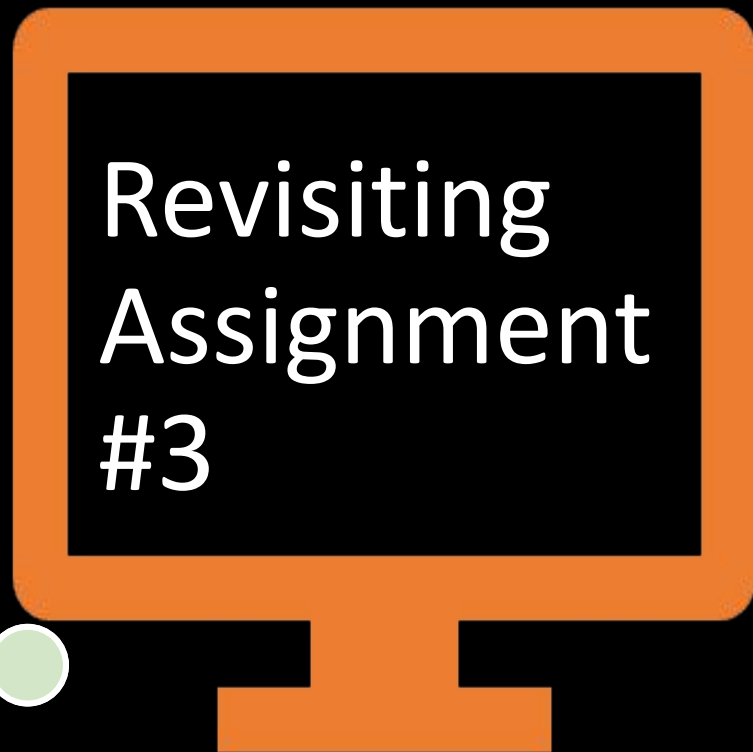
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |

```
docker build -t fish549_image .  
  
docker run -d --rm -it -e PASSWORD=rockon -v C:\\Users\\Dana-SAFS\\Desktop\\tutorial_fish549\\home\\rstudio\\tutorial_fish549 -p 8787:8787 --name fish549_container fish549_image  
  
docker ps  
  
example(stan_model, package = "rstan", run.dontrun = TRUE)
```

New Folder New Blank File Upload Delete Rename More
Home
tutorial_fish549



Task 4

Assume that you do not understand the source of the warning and resulting value for `mean_count` (and it's OK if you don't). Create a [new Gist](#) that asks for help in resolving the error. Make sure to

- Give your Gist a meaningful description
- In the box that says "Filename including extension...", enter `problem_with_mean.md`, which will allow you to use Markdown to mix text and code in the body of the message below just as you would with a repo's `README.md` file.
- In the body of the Gist, include *all of the information that someone would need to assist you* without the need to load additional files, navigate elsewhere, etc.

Hint: You can denote R code in Markdown documents in 1 of 2 ways:

1. For a single line of code, enclose the statement with a single back-tick (``) like ``a <- 1``
2. For a block of code, enclose the statements with three back-ticks (```) like

```
```\na <- 1\nb <- 2\na / b\n```
```



# Suggestion

- You could include a Readme file with information pertinent to the error and push the directory to a GitHub. You'd probably have gitignore file so that you're only including the necessary files in the repository
- Your repository could be pulled by your helper and they should be able to replicate your error, make changes to the file and then push the result back to the repository





## Workflow

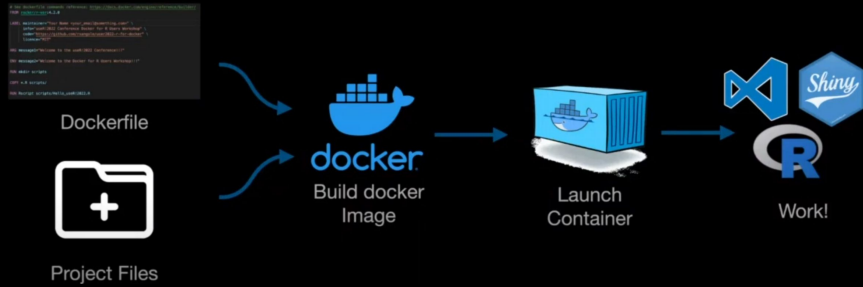
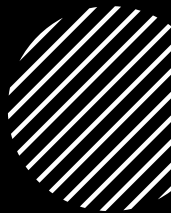


Image from [Docker for R users Tutorial](#)

# More?



**There are lots of tutorials on YouTube** [Here's one I have looked at that is part of a series](#)

**[Docker documentation](#)**

# Takeaways

## Docker:

- Allows packaging your development environment
- Allows sharing your development environment with others
  - can make debugging easier
- The Rocker project makes things easier by taking some of the headache out of installing R and RStudio in a Docker environment because they provide and maintain images that you can use.



# Credits

The Muppet Show snippet is from The Muppet Show's opening theme, written by Sam Pottle and [Jim Henson](#).