R Installation and Setup

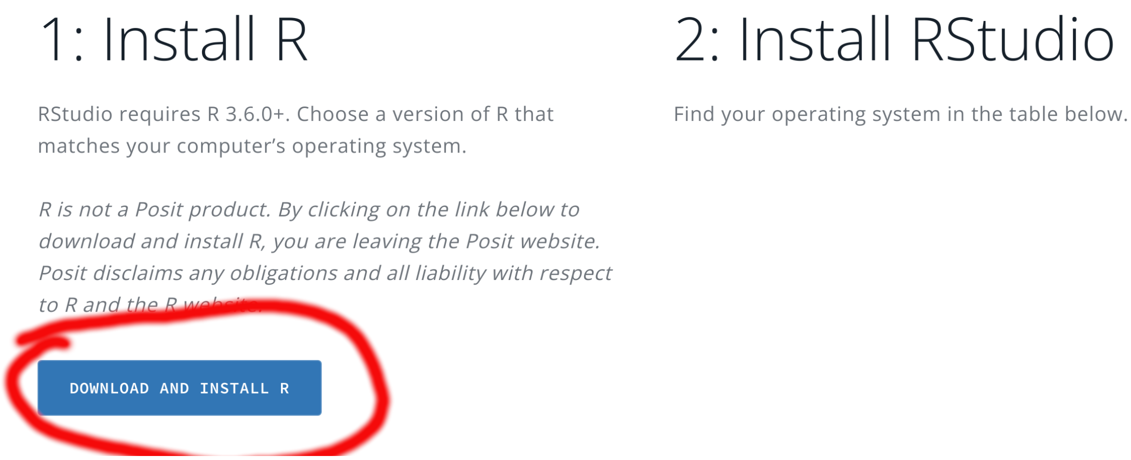
Sam Gifford

2024-10-08

Downloading R

https://posit.co/download/rstudio-desktop/

First download R. You'll want to follow the links based on your specific system. The following screenshots show how this works for Windows, though you'll follow similar instructions for a mac. If using a chromebook you'll need to see the specific instructions below. Once you have the installer downloaded, you should be able to click "next", "install", or "finish" with the default options without any issue.



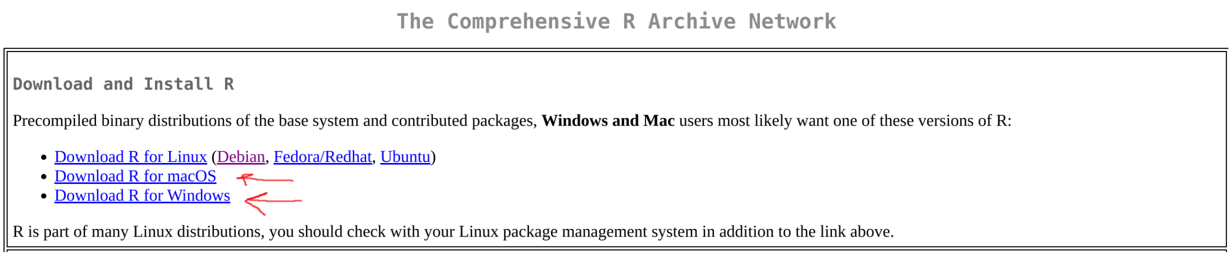
1: Install R

RStudio requires R 3.6.0+. Choose a version of R that matches your computer's operating system.

R is not a Posit product. By clicking on the link below to download and install R, you are leaving the Posit website. Posit disclaims any obligations and all liability with respect to R and the R website.

2: Install RStudio

Find your operating system in the table below.



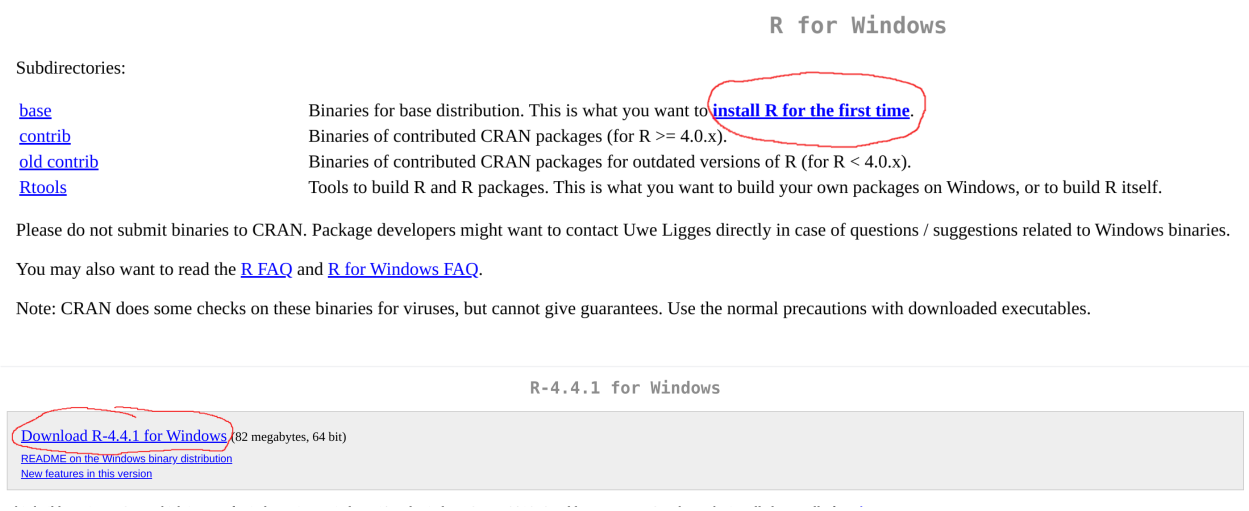
The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux (Debian, Fedora/Redhat, Ubuntu)
- Download R for macOS
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.



R for Windows

Subdirectories:

base	Binaries for base distribution. This is what you want to install R for the first time.
contrib	Binaries of contributed CRAN packages (for R >= 4.0.x).
old contrib	Binaries of contributed CRAN packages for outdated versions of R (for R < 4.0.x).
Rtools	Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the R FAQ and R for Windows FAQ.

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

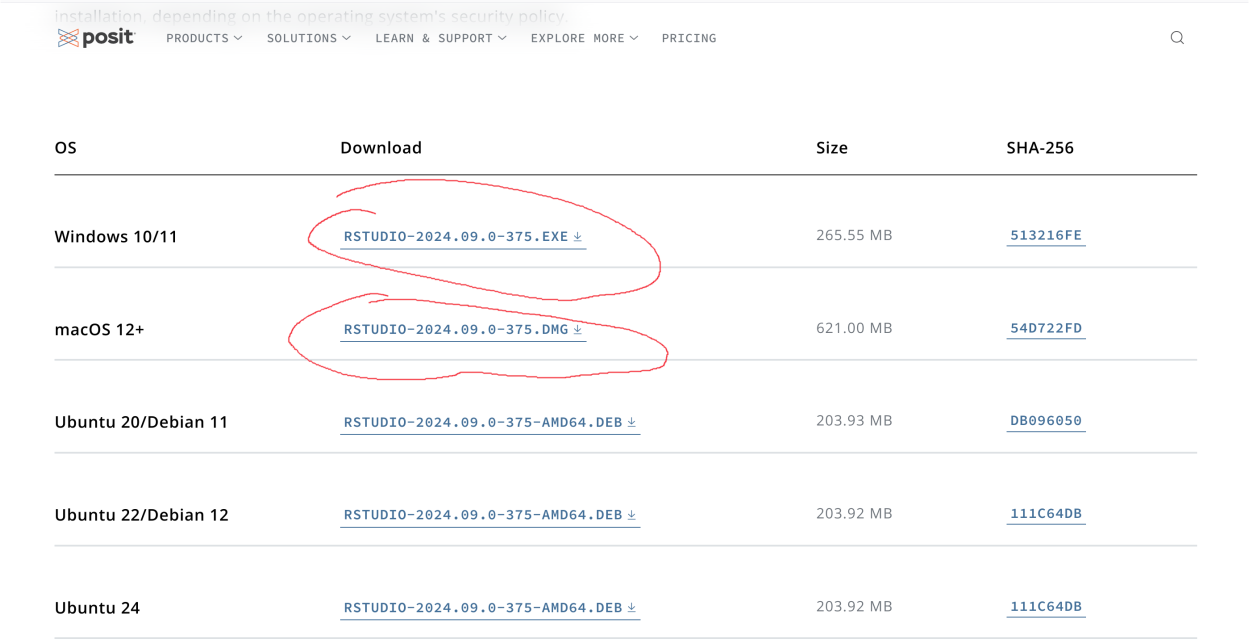
R-4.4.1 for Windows

Download R-4.4.1 for Windows (82 megabytes, 64 bit)
README on the Windows binary distribution
New features in this version

then download and run Rstudio desktop. R can be used as a standalone with RGui (or via command line) but most people use Rstudio.

Installing Rstudio

For rstudio, use the same link as above and scroll down to directly obtain the executable or dpkg file for windows and mac OS, respectively. Similar to the base R installation, the default options will all be good. It will ask you to associate RStudio with an R installation at some point, so make sure you have installed R first. The default option should always be good to use, so you can just click "next" using all default options as before.



installation, depending on the operating system's security policy.

OS	Download	Size	SHA-256
Windows 10/11	RSTUDIO-2024.09.0-375.EXE	265.55 MB	513216FE
macOS 12+	RSTUDIO-2024.09.0-375.DMG	621.00 MB	54D722FD
Ubuntu 20/Debian 11	RSTUDIO-2024.09.0-375-AMD64.DEB	203.93 MB	DB096050
Ubuntu 22/Debian 12	RSTUDIO-2024.09.0-375-AMD64.DEB	203.92 MB	111C64DB
Ubuntu 24	RSTUDIO-2024.09.0-375-AMD64.DEB	203.92 MB	111C64DB

Confirming the R installation

Just to clarify, R is the programming language, which comes with a command line interface (that is amazing for power users but otherwise no one uses) as well as an "RGui" application that allows you to run R code in a graphical interface. Virtually everyone uses Rstudio, which is an alternative to the base RGui with much more functionality. Rstudio is an example of an IDE, or integrated development environment. The base icons for the two programs looks similar, so make sure you're launching Rstudio:

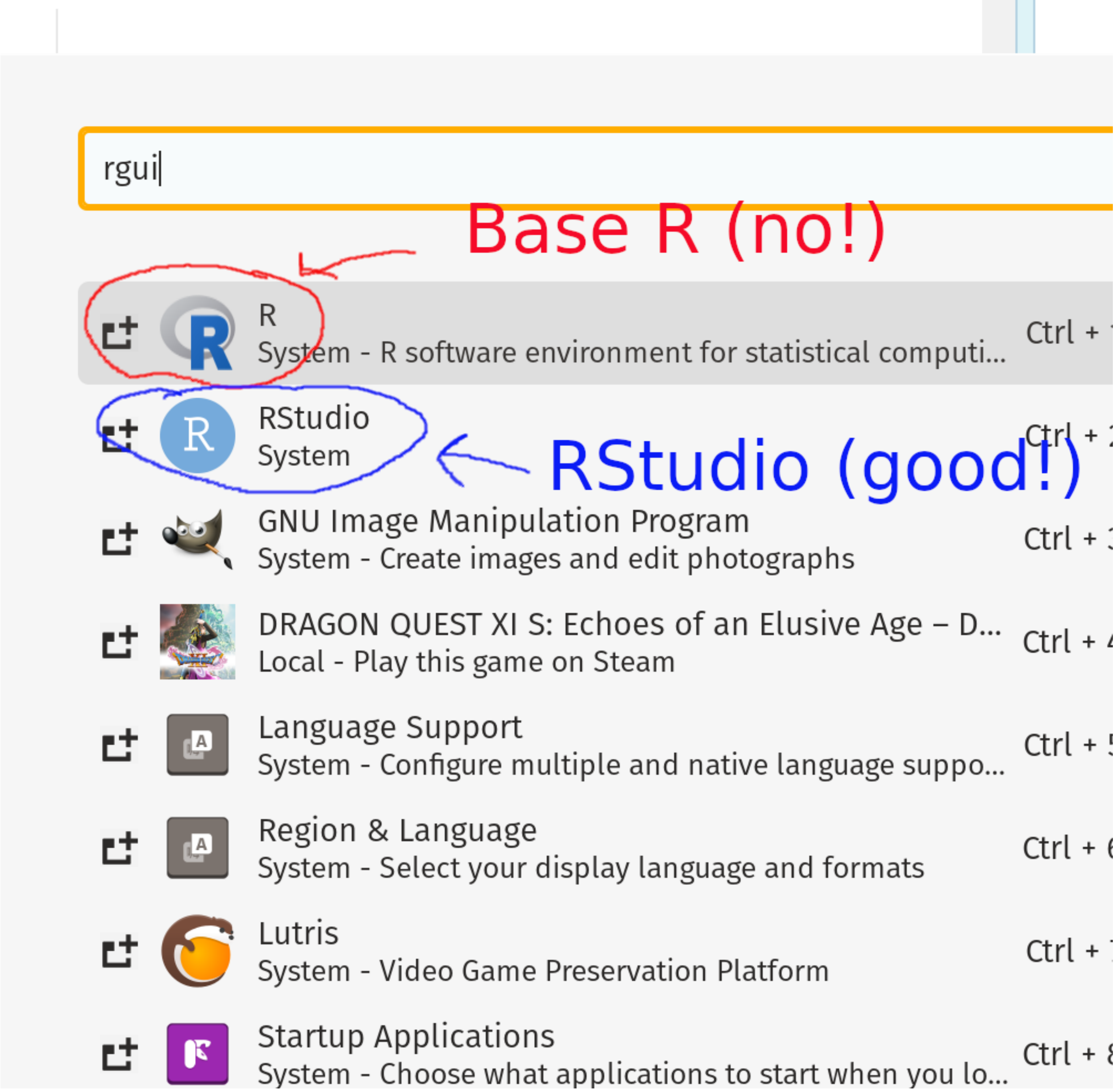
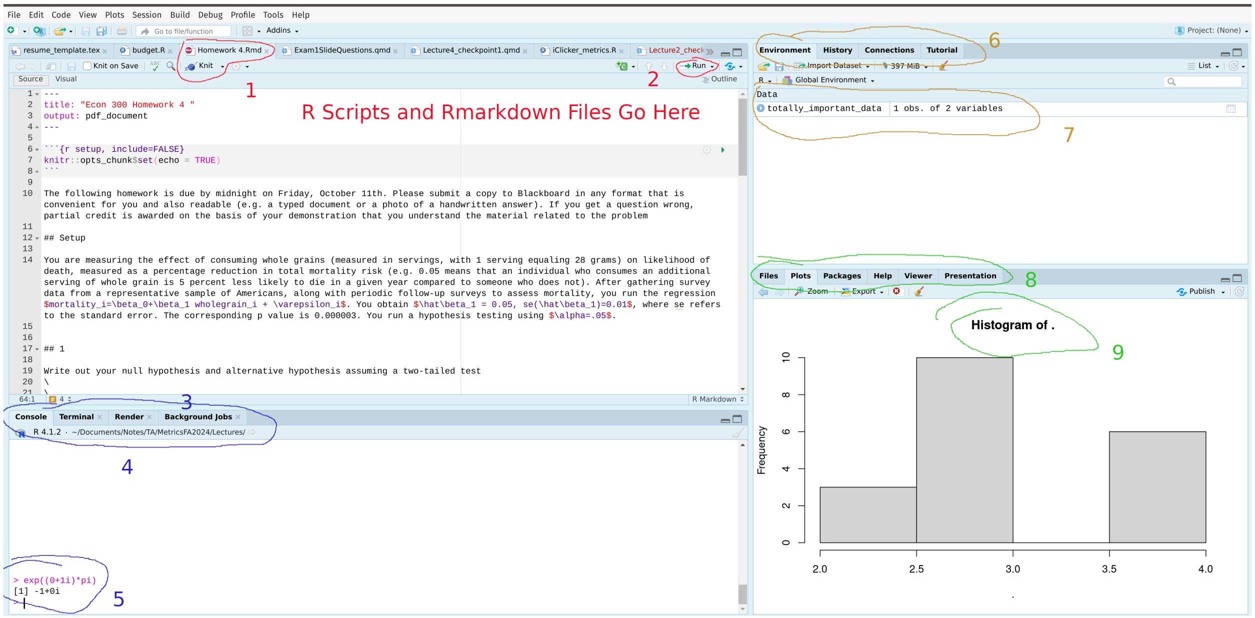

rgui
Base R (no!)
R
System - R software environment for statistical computi...
Ctrl +
RStudio
System
Ctrl +
RStudio (good!)
GNU Image Manipulation Program
System - Create images and edit photographs
Ctrl +
DRAGON QUEST XI S: Echoes of an Elusive Age – D...
Local - Play this game on Steam
Ctrl +
Language Support
System - Configure multiple and native language suppo...
Ctrl +
Region & Language
System - Select your display language and formats
Ctrl +
Lutris
System - Video Game Preservation Platform
Ctrl +
Startup Applications
System - Choose what applications to start when you lo...
Ctrl +

Figure 1: This will look a bit different from your system since I'm running Debian. And since some of my games library is popping up in the search.

Using Rstudio

Rstudio is broken up into 4 quadrants by default (you may be missing the upper left quadrant if you haven't opened an R file yet). A basic overview of this is given below



~~~
---
title: "Econ 300 Homework 4 "
output: pdf_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

The following homework is due by midnight on Friday, October 11th. Please submit a copy to Blackboard in any format that is
convenient for you and also readable (e.g. a typed document or a photo of a handwritten answer). If you get a question wrong,
partial credit is awarded on the basis of your demonstration that you understand the material related to the problem

## Setup

You are measuring the effect of consuming whole grains (measured in servings, with 1 serving equaling 28 grams) on likelihood of
death, measured as a percentage reduction in total mortality risk (e.g. 0.05 means that an individual who consumes an additional
serving of whole grain is 5 percent less likely to die in a given year compared to someone who does not). After gathering survey
data from a representative sample of Americans, along with periodic follow-up surveys to assess mortality, you run the regression
$mortality_i=\beta_0+\beta_1 wholegrain_i + \varepsilon_i$. You obtain $\hat\beta_1 = 0.05, se(\hat\beta_1)=0.01$, where se refers
to the standard error. The corresponding p value is 0.000003. You run a hypothesis testing using $\alpha=.05$.


## 1

Write out your null hypothesis and alternative hypothesis assuming a two-tailed test
\
\
~~~

```
R 4.1.2 · ~/Documents/Notes/TA/MetricsFA2024/Lectures/
```

```
> exp((0+1i)*pi)
[1] -1+0i
>
```

```
Data
totally_important_data   1 obs. of 2 variables
```

1. Files go in the upper left panel by default. If the file has unsaved changes it's name will be displayed in red. In general Rstudio's autosaving features are great, but you should still save frequently to avoid losing progress. If you have an rstudio file, a 'knit' option (circled) will be available, which will allow you to convert a markdown file into PDF or other portable, human-readable formats
2. You can run the entire contents of a script by pressing this button in the upper-right, or by using the "source" function. More commonly, you can run an individual line of code in a file by pressing control+enter (command+enter on a mac)
3. Your console will be in the lower left by default, where you can run scratch code that you don't need to save. Note that there are several tabs here - you also have access to a terminal (for advanced users), a render tab for rmarkdown, nad a background jobs tab (also for advanced users). When you knit a document your focus will be automatically moved from console to 'render', so if you can't run code in console please make sure you are on the correct tab
4. Directly below the console tab is your version of R and your working directory. When attempting to read in files, your base path will be this. refers to your home directory, and is generally something like "C:/Users/username" but it can vary by system. You can change this with the "setwd" function
5. This is the part of the console where you write code. Do not run important code here as it's easy to lose work this way. I normally use this to test out syntax if I'm unsure.
6. This tab contains your environment by default, which lists out every variable you have stored. You can also get this by using the "ls" function. The history tab is also available here, which includes every line of code you've run in console since starting the session (in case you need to retrieve something you previously wrote)
7. Clicking on data frames will bring up an excel-style visualization of the data set (it calls the "view" function). Once you understand how to interact with data in R you will never use this feature, as it is far too slow.
8. Most commonly your plots will appear in this panel. The file explorer is also useful if you are unsure of where your working directory is. You can also bring up help documentation for any function by typing ?functionName in console (e.g. ?read.csv). Quality of documentation will vary by package, but is generally a good place to start if you don't know how to call a function
9. Your plots will go here - in the upper-left is an option to export the plot to an image file if you're unsure of how to do that via code (functions like 'png' and 'pdf' will do this).

Also note that there are many ways to customize your R experience. Check out Tools > Global Options for a list. Some highlights include an ability to change to a dark theme, change autocompletion settings

(RStudio has great autocomplete options), or changing keybindings (I use vim keybindings). You also have a hidden file named .Rprofile where you can customize launch settings for advanced users.

Basic Packages

One of the benefits of R is its robust package system, which allows users to upload their own packages of functions for other users to download. While base R has many features, many R packages are so commonly used that you should generally install them immediately upon downloading R. In general, to install a packages you use the "install.packages" function. Packages only ever need to be installed once:

```
install.packages("dplyr") #common data manipulation
install.packages("ggplot2") #Make good looking graphs
install.packages("rmarkdown") #knit rmarkdown files
install.packages("magrittr") #pipe operator for old people before it was put into base R
install.packages("tinytex") #Probably already installed - used to easily install latex
```

Even if a package is installed, R will not know of these functions by default (it's namespace isn't attached). There are two ways to call a function in a package, the first is to use a scoping operator explicitly, e.g. ggplot2::ggplot(). More commonly, you run the library command, which only needs to be run once per session (but needs to be rerun every time R is opened)

```
library(dplyr)
```

By default, messages in R are displayed in red text. Sometimes loading packages, such as dplyr, will display information in red text, but this does not necessarily indicate an error has occurred.

Basic LaTeX Installation

Rmarkdown files can be used to 'knit' a file containing R code along with text into a nicely formatted document. I use this to create homework files and research papers, among other things. In order to render this output to a PDF, you will need a LaTeX installation. This is generally a painful process if you're unfamiliar with package management systems, but thankfully there is an easy way to install latex via R that will work on most systems:

```
tinytex::install_tinytex()
```

You may need to type "Y" or sign in as admin for this to proceed, but this is the easiest way to get a minimal LaTeX installation on most systems.

Setting up Rstudio Server for Chromebooks

If using a Chromebook you'll want to use the Linux (Debian) install using the built in terminal. The link has more information, but the following should be the barebones of what is needed. You'll run these commands within your linux terminal (penguin). The keys and file links change periodically as builds are updated, but these are up to date as of September 4th, 2024

```
sudo apt-get update
sudo apt-get install r-base r-base-dev
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-key '95C0FAF38DB3CCAD0C080A7BDC78B2DDEABC47B7'
```

This will install R as a command line program. You then need to install rstudio server (https://posit.co/download/rstudio-server/). You'll need to run

```
sudo apt-get install gdebi-core
wget https://download2.rstudio.org/server/focal/amd64/rstudio-server-2024.04.2-764-amd64.deb
sudo gdebi rstudio-server-2024.04.2-764-amd64.deb
```

Once you have these installed, you'll need to launch Rstudio server using the following command

```
/usr/sbin/rstudio-server start
```

then open 127.0.0.1:8787 in url bar in chrome (rstudio server uses port 8787 by default, but this can be changed using different arguments)

Setting Up Rmarkdown for chrome

A tex package needs to be installed to render PDFs for rmarkdown - the following will work in a *nix environment with the firm run in terminal and the second run within R

```
sudo apt-get install texlive
```

```
install.packages("tinytex")
tinytex::install_tinytex()
```

Virtual Lab

If you're unable to install R, UIC has a virtual lab with a default R/RStudio installation that you can use.

https://it.uic.edu/services/student/software-resources/remote-access-to-specialized-software-with-virtual-computer-labs/