BioDIGS: Exploring Soil Data

February 24, 2025

Table of Contents

# About this Book

This is a companion training guide for BioDIGS, a GDSCN project that brings a research experience into the classroom. This activity guides students through exploration of the BioDIGS soil data using the tidyverse in R. Students will learn basic data summarization, visualization, and mapping skills. Visit the BioDIGS (BioDiversity and Informatics for Genomics Scholars) website [here](https://biodigs.org/) for more information about this collaborative, distributed research project, including how you can get involved!

The GDSCN (Genomics Data Science Community Network) is a consortium of educators who aim to create a world where researchers, educators, and students from diverse backgrounds are able to fully participate in genomic data science research. You can find more information about its mission and initiatives [here](https://www.gdscn.org/home).



## 0.1 Target Audience

The activities in this guide are written for undergraduate students and beginning graduate students. Some sections require basic understanding of the R programming language, which is indicated at the beginning of the chapter.

## 0.2 Platform

The activities in this guide are demonstrated on NHGRI’s [AnVIL](https://anvilproject.org/) cloud computing platform. AnVIL is the preferred computing platform for the GDSCN. However, all of these activities can be done using your personal installation of R or using the online [Galaxy](usegalaxy.org) portal.



## 0.3 Data

The data generated by the BioDIGS project is available through the [BioDIGS website](biodigs.org), as well as through an [AnVIL workspace](https://anvilproject.org).

Data about the soil itself as well as soil metal content was generated by the [Delaware Soil Testing Program](https://www.udel.edu/canr/cooperative-extension/environmental-stewardship/soil-testing/) at the University of Delaware. Sequences were generated by the [Johns Hopkins University Genetic Resources Core Facility](https://grcf.jhmi.edu/) and by [PacBio](https://www.pacb.com/).

In order to run your analyses, you will use the [AnVIL cloud computing platform](https://anvilproject.org/). The AnVIL (Analysis Visualization and Informatics Lab-space) platform is specially designed for analyzing biological data, and is used by scientists doing all sorts of biological research.

**AnVIL in a nutshell**

* Behind the scenes, AnVIL relies on Google Cloud Platform to provide computing infrastructure. Basically, AnVIL lets you “rent” computers through the internet. The analysis is run on the rented computer. AnVIL lets you see the results in your browser.
* AnVIL uses [Terra](https://anvil.terra.bio/) to provide many computational tools useful for biological data analysis, such as [RStudio](https://www.rstudio.com/products/rstudio/), [Galaxy](https://usegalaxy.org/), and [Jupyter Notebooks](https://jupyter.org/). Terra takes care of installing these tools so you can use them right away.

### 0.3.1 Create Google Account

First, you will need to set up a (free) Google account.

A Google account usually looks like “myname@gmail.com”. Alternatively, you can enable Google for an existing non-Gmail email address using [these instructions](https://support.terra.bio/hc/en-us/articles/360029186611).

If you do not already have a Google account that you would like to use for accessing AnVIL, [create one now](https://accounts.google.com/SignUp).

### 0.3.2 Log In to Terra

Next, make sure you can log in to Terra – you will use Terra to perform computations on AnVIL.

You can access Terra by going to [anvil.terra.bio](https://anvil.terra.bio/). Open Terra, and you should be prompted to sign in with your Google account.

### 0.3.3 Share Username

Make sure your instructor has your Google account username (e.g. myname@gmail.com), so they can give you access to everything you need.

* Make sure there are no typos!
* If you have multiple Google accounts, make sure you give them the username that you will be using to log in to anvil.terra.bio.

It is *very important* that you share the Google account you will be using to access AnVIL with with your instructor! Otherwise, the instructor cannot add you to Billing Projects or Workspaces, and you will be unable to proceed with your assignments.

## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output

This **will not work** until your instructor has given you permission to spend money to “rent” the computers that will power your analyses (by adding you to a “Billing Project”).

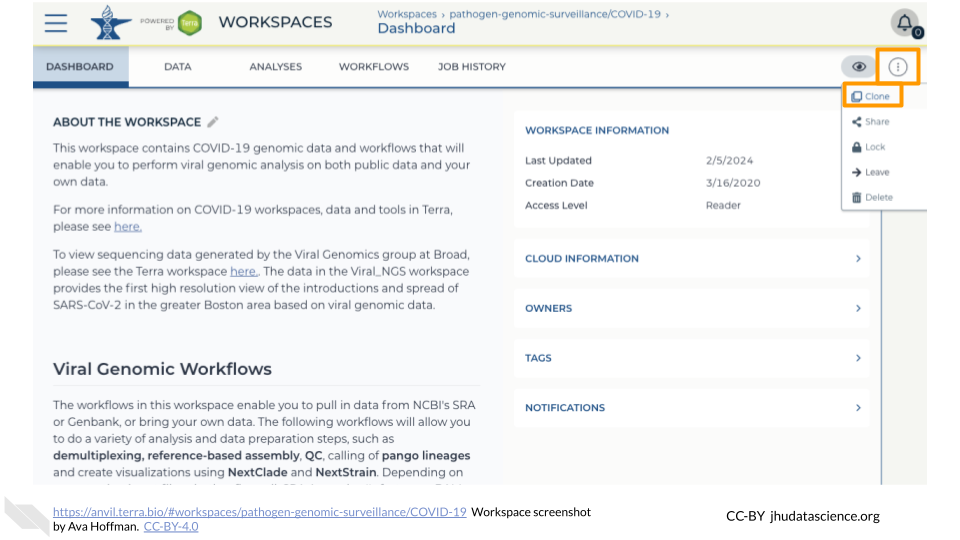
On AnVIL, you access files and computers through **Workspaces**. Each Workspace functions almost like a mini code laboratory - it is a place where data can be examined, stored, and analyzed. The first thing we want to do is to copy or “clone” a Workspace to create a space for you to experiment. This will give you access to

* the files you will need (data, code)
* the computing environment you will use

**Tip** At this point, it might make things easier to open up a new window in your browser and split your screen. That way, you can follow along with this guide on one side and execute the steps on the other.

To clone an AnVIL Workspace:

1. Go to the Workspace by clicking this link: ask your instructor.
2. Clone the workspace by clicking the teardrop button (). Select “Clone”. See the screenshot below on a different Workspace:

* 

1. You will see a popup box appear, asking you to configure your Workspace
   1. Give your Workspace clone a name by adding an underscore (“\_“) and your name. For example, "ExampleWorkspace\_Firstname\_Lastname".
   2. Select the Billing Project provided by your instructor.
   3. Leave the bottom two boxes as-is.
   4. Click “CLONE WORKSPACE”.

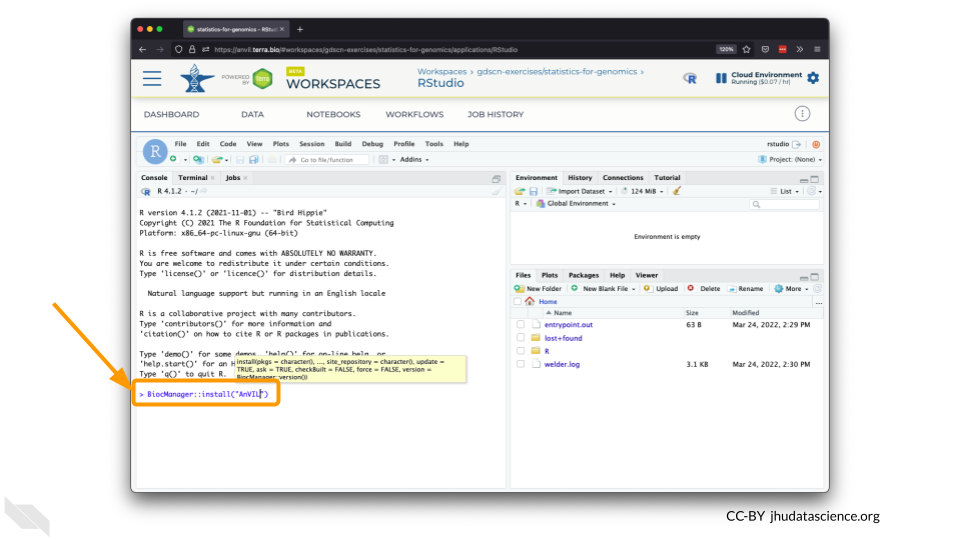
* See the screenshot below on a different Workspace:
* 

1. The new Workspace should now show up under <https://anvil.terra.bio/#workspaces>. This is your own copy of the Workspace to work in.

## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output

Next, we will be using RStudio and the package Glimma to create interactive plots. See [this vignette](https://bioconductor.org/packages/release/bioc/vignettes/Glimma/inst/doc/limma_edger.html) for more information.

1. The Bioconductor team has created a very useful package to programmatically interact with Terra and Google Cloud. Install the AnVIL package. It will make some steps easier as we go along.

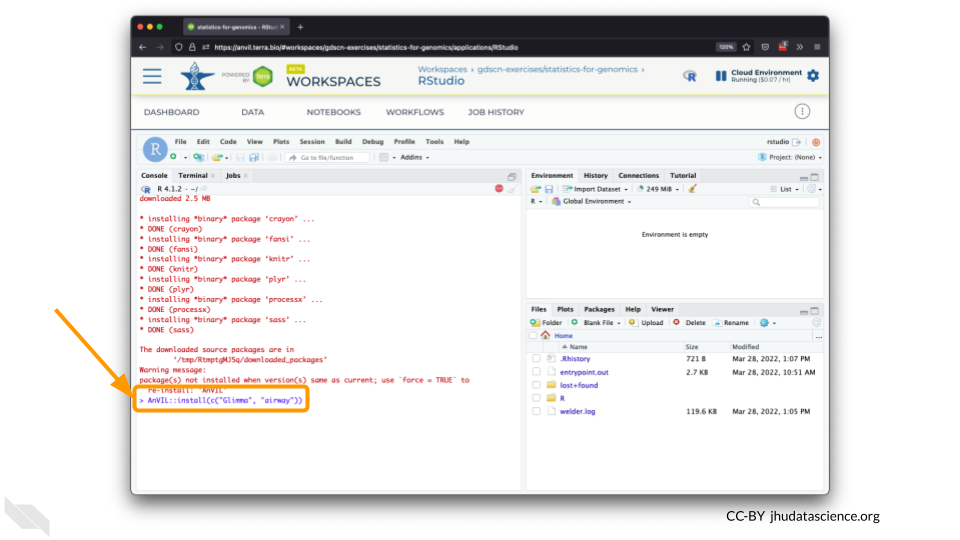
* 

1. You can now quickly install precompiled binaries using the AnVIL package’s install() function. We will use it to install the Glimma package and the airway package. The airway package contains a SummarizedExperiment data class. This data describes an RNA-Seq experiment on four human airway smooth muscle cell lines treated with dexamethasone.

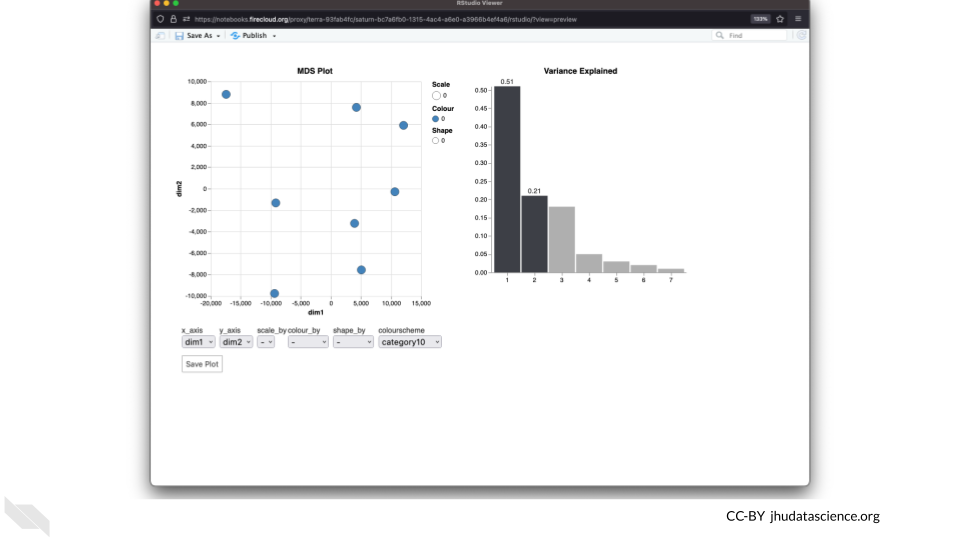
{Note: for some of the packages, you will have to install packaged from the CRAN repository, using the install.packages() function. The examples will show you which install method to use.}

![](09-student\_guide\_files/figure-docx//1BLTCaogA04bbeSD1tR1Wt-mVceQA6FHXa8FmFzIARrg\_g11f12bc99af\_0\_56.png)

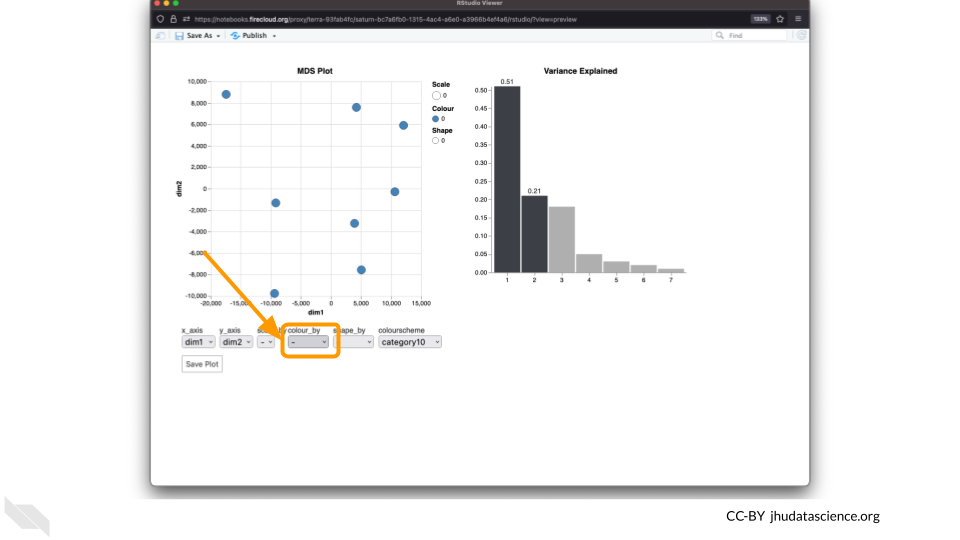
1. Load the example data.

* 

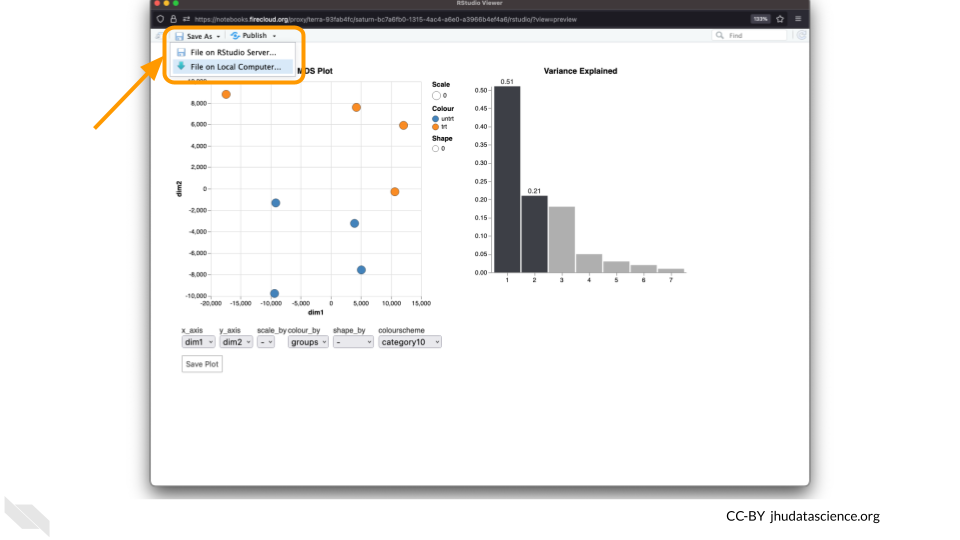
1. The multidimensional scaling (MDS) plot is frequently used to explore differences in samples. When this data is MDS transformed, the first two dimensions explain the greatest variance between samples, and the amount of variance decreases monotonically with increasing dimension. The following code will launch a new window where you can interact with the MDS plot.

* 

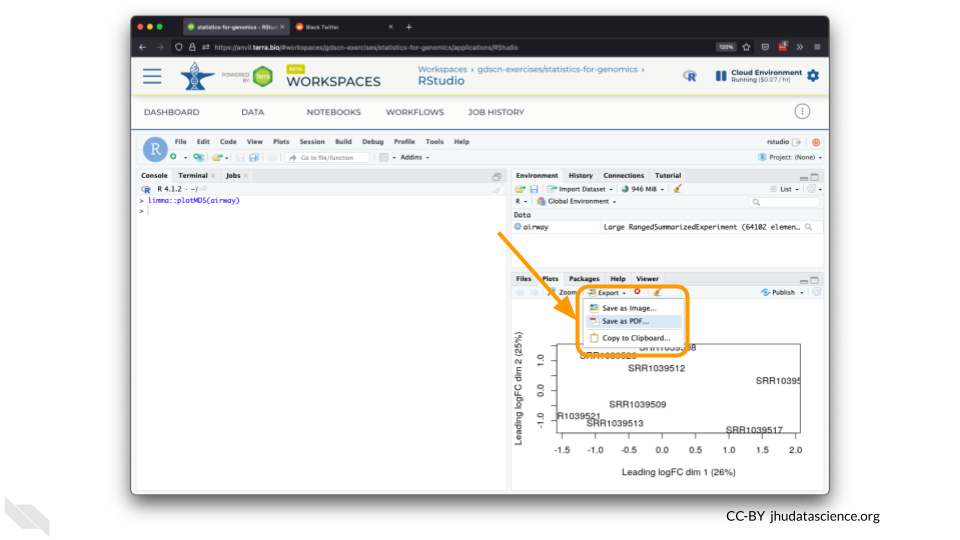
1. Change the colour\_by setting to “groups” so you can easily distinguish between groups. In this data, the “group” is the treatment.

* 

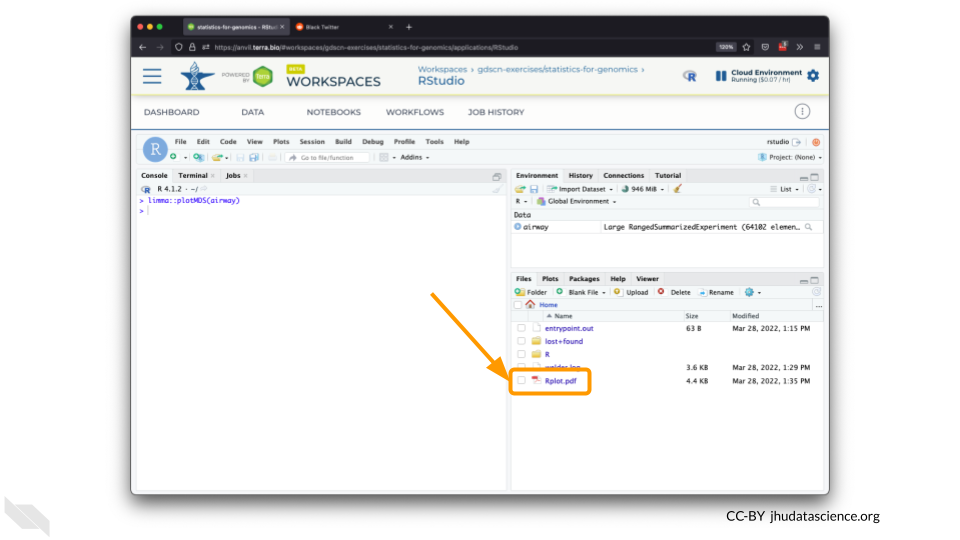
1. You can download the interactive html file by clicking on “Save As”.

* 

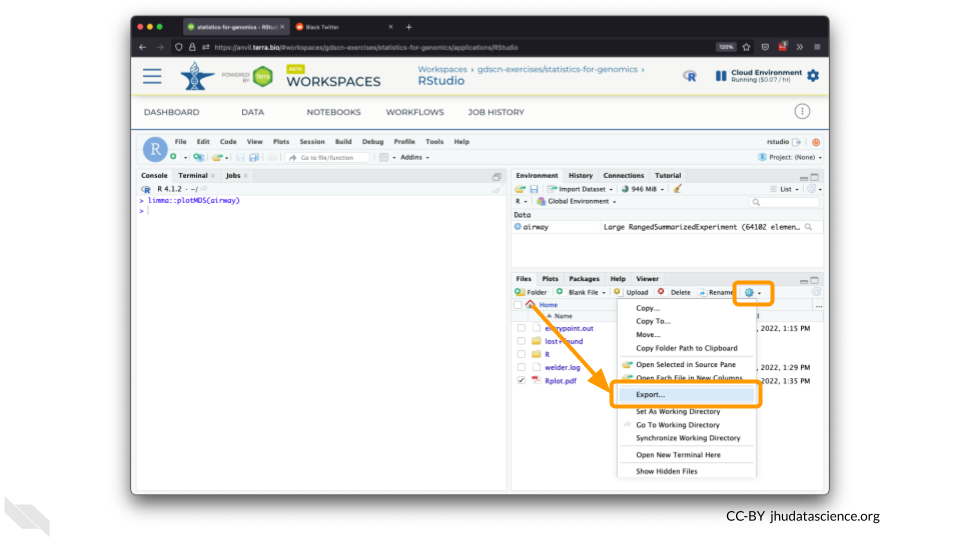
1. You can also download plots and other files created directly in RStudio. To download the following plot, click on “Export” and save in your preferred format to the default directory. This saves the file in your cloud environment.

* 

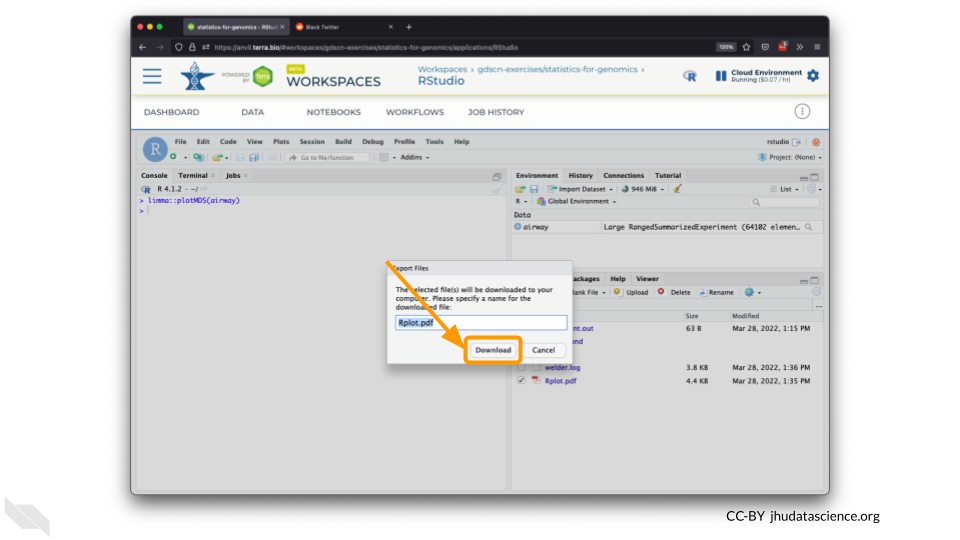
1. You should see the plot in the “Files” pane.

* 

1. Select this file and click “More” > “Export”

* 

1. Select “Download” to save the file to your local machine.

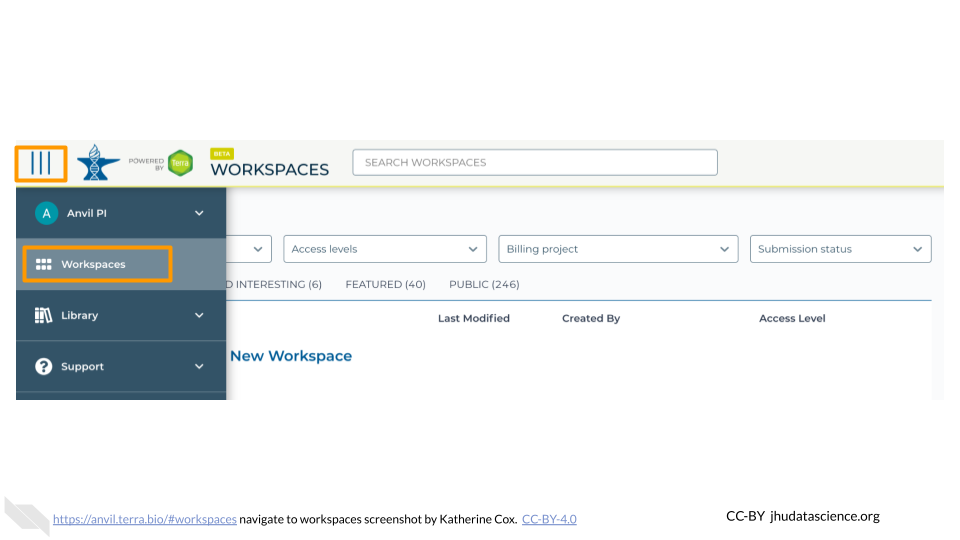
* 

## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output

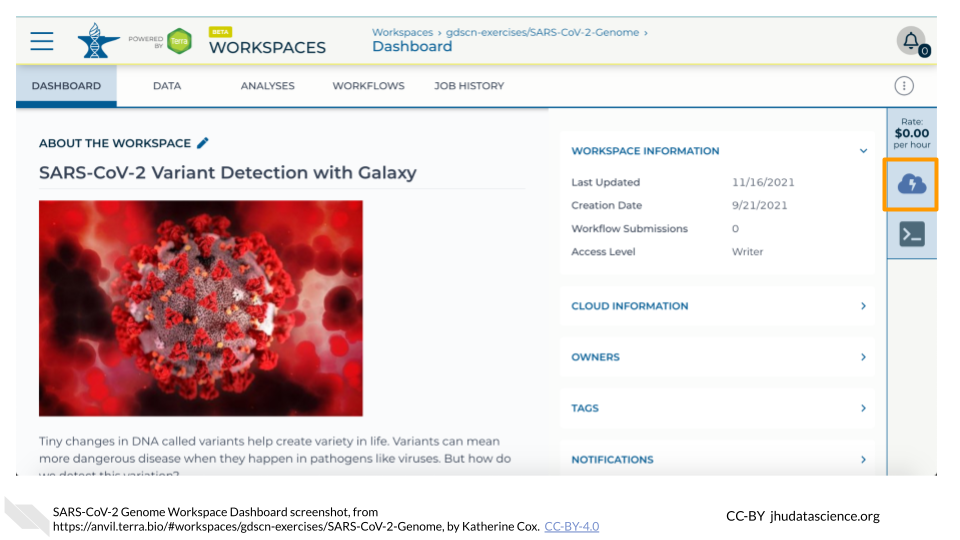
AnVIL is very versatile and can scale up to use very powerful cloud computers. It’s very important that you select a cloud computing environment appropriate to your needs to avoid runaway costs. If you are uncertain, start with the default settings; it is fairly easy to increase your compute resources later, if needed, but harder to scale down.

Note that, in order to use RStudio, you must have access to a Terra Workspace with permission to compute (i.e. you must be a “Writer” or “Owner” of the Workspace).

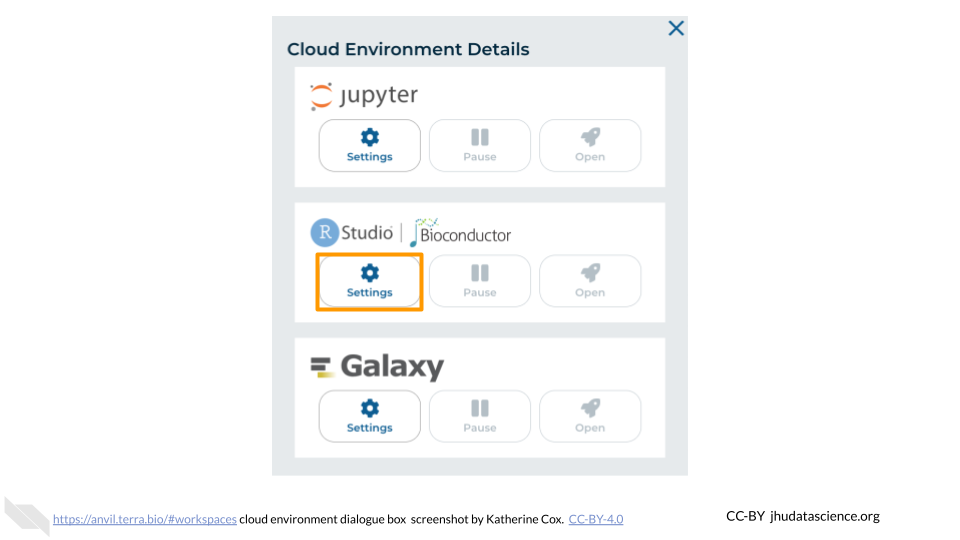
1. Open Terra - use a web browser to go to [anvil.terra.bio](https://anvil.terra.bio/)
2. In the drop-down menu on the left, navigate to “Workspaces”. Click the triple bar in the top left corner to access the menu. Click “Workspaces”.

* 

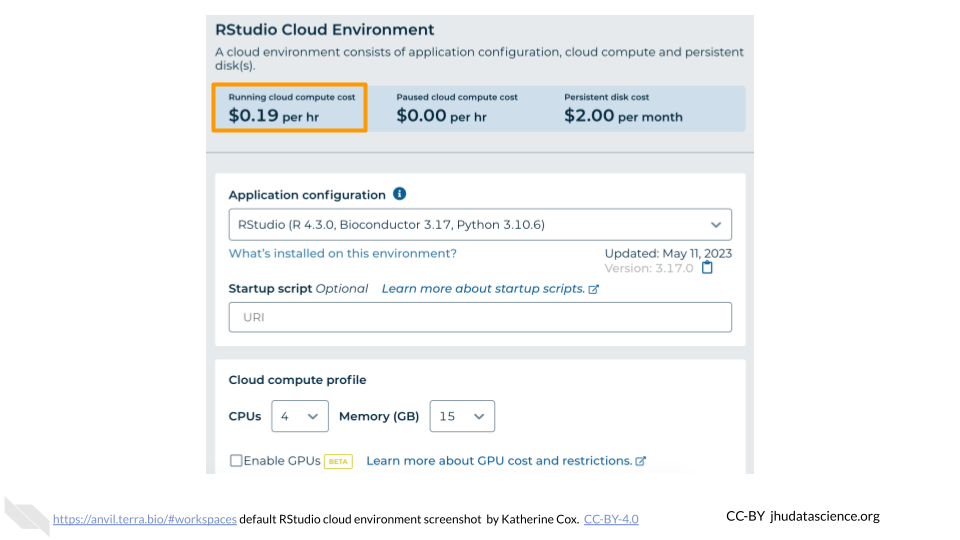
1. Click on the name of your Workspace. You should be routed to a link that looks like: https://anvil.terra.bio/#workspaces/<billing-project>/<workspace-name>.
2. Click on the cloud icon on the far right to access your Cloud Environment options. If you don’t see this icon, you may need to scroll to the right.

* 

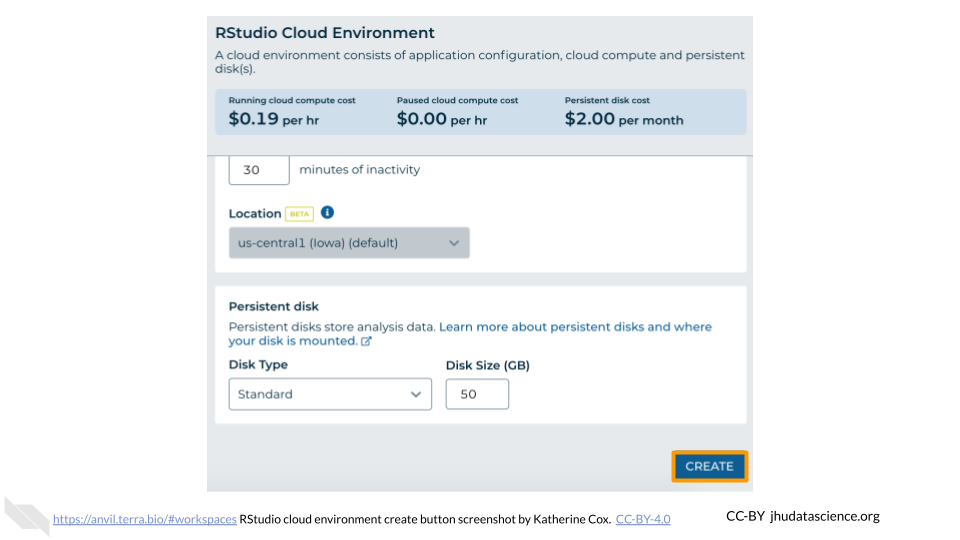
1. In the dialogue box, click the “Settings” button under RStudio.

* 

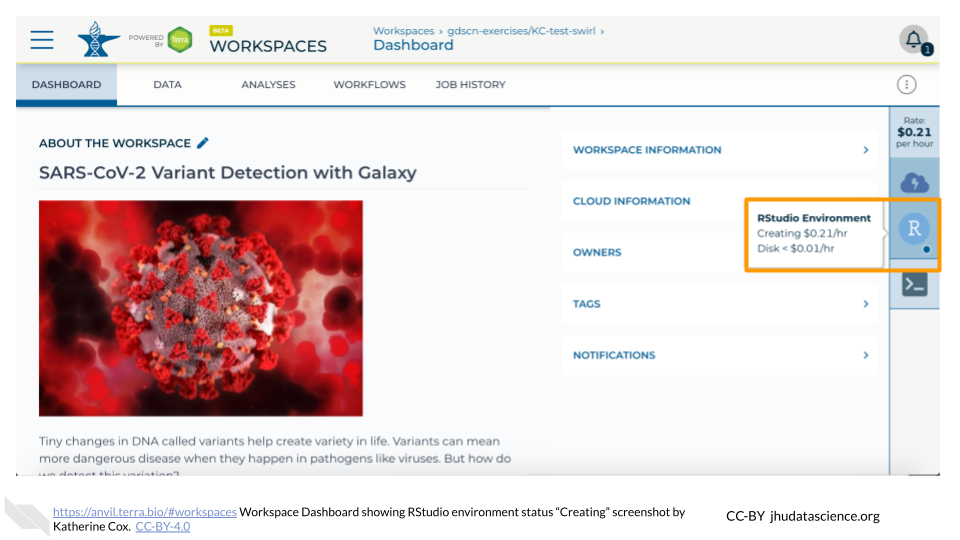
1. You will see some configuration options for the RStudio cloud environment, and a list of costs because it costs a small amount of money to use cloud computing.

* 

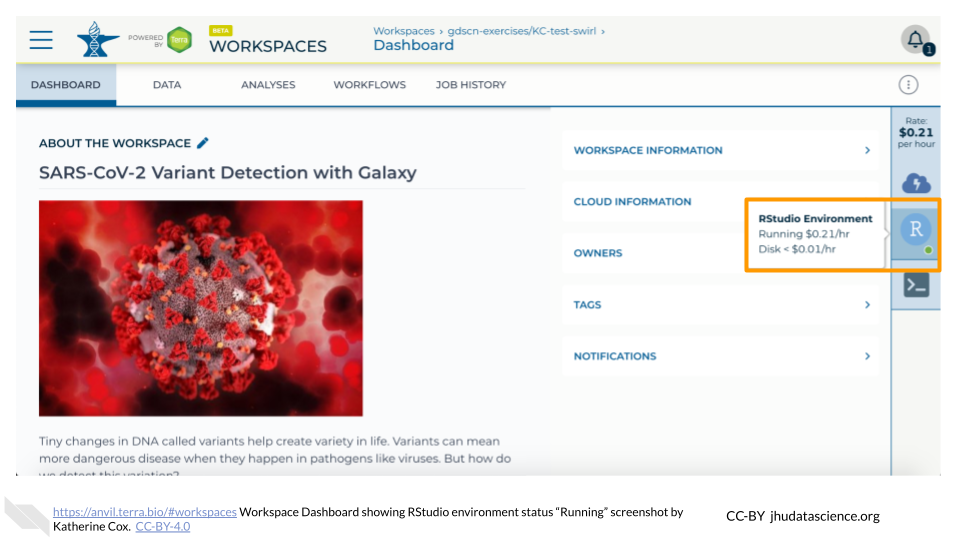
1. Configure any settings you need for your cloud environment. If you are uncertain about what you need, the default configuration is a reasonable, cost-conservative choice. It is fairly easy to increase your compute resources later, if needed, but harder to scale down. Scroll down and click the “CREATE” button when you are satisfied with your setup.

* 

1. The dialogue box will close and you will be returned to your Workspace. You can see the status of your cloud environment by hovering over the RStudio icon. It will take a few minutes for Terra to request computers and install software.

* 

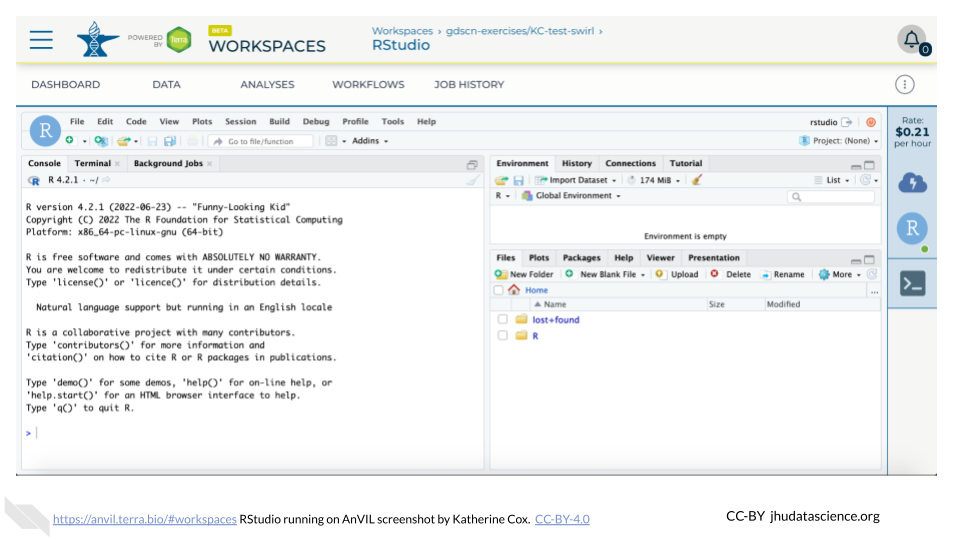
1. When your environment is ready, its status will change to “Running”. Click on the RStudio logo to open a new dialogue box that will let you launch RStudio.

* 

1. Click the launch icon to open RStudio. This is also where you can pause, modify, or delete your environment when needed.

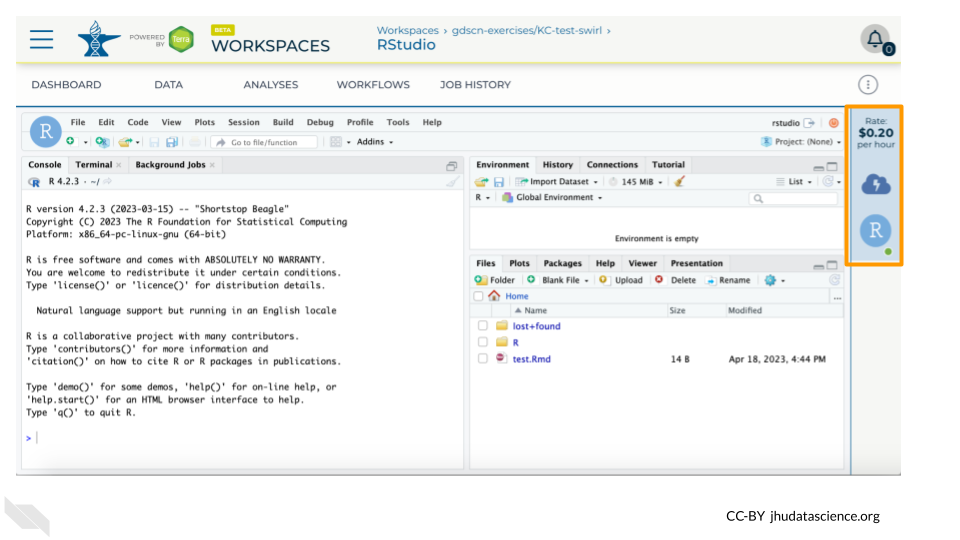
* 

1. You should now see the RStudio interface with information about the version printed to the console.

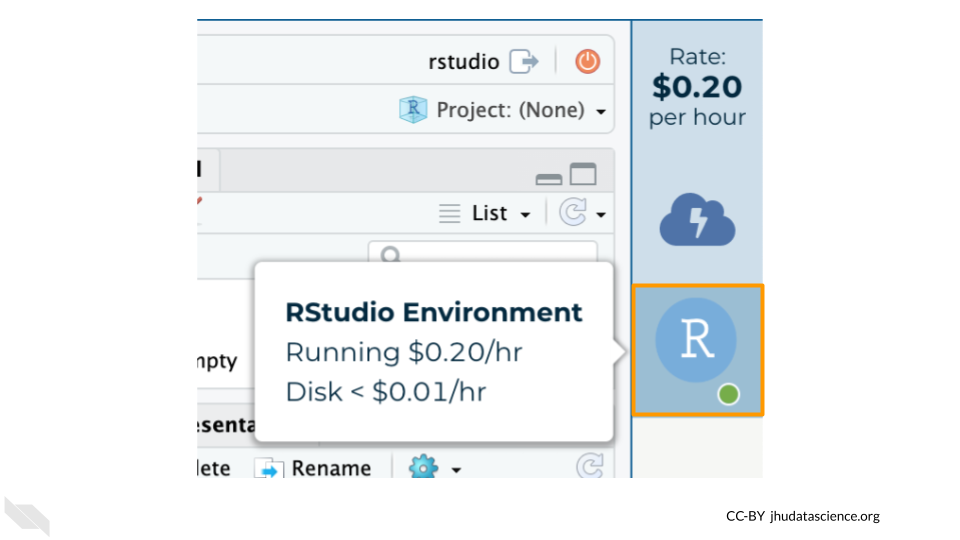
* 

## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output

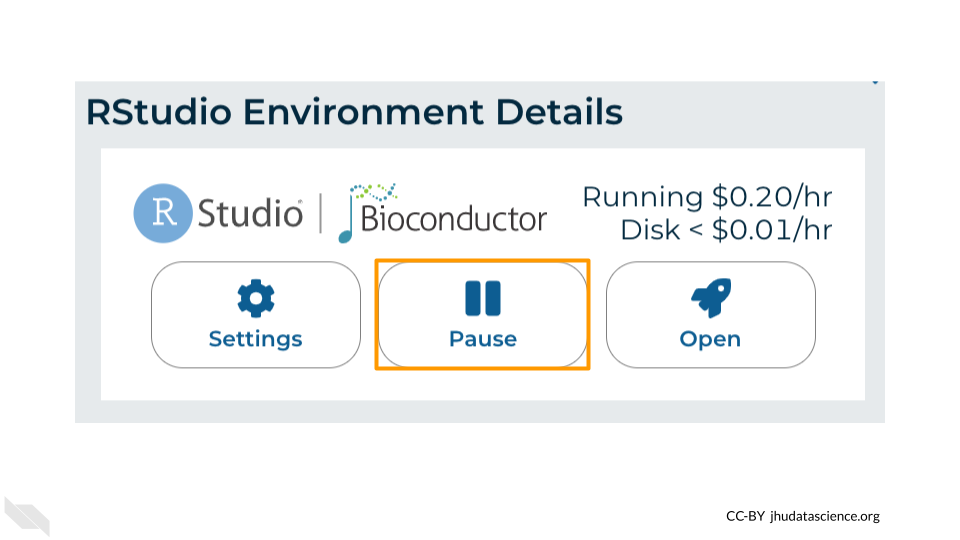
1. You can view costs and make changes to your cloud environments from the panel on the far right of the page. If you don’t see this panel, you may need to scroll to the right. Running environments will have a green dot, and paused environments will have an orange dot.

* 

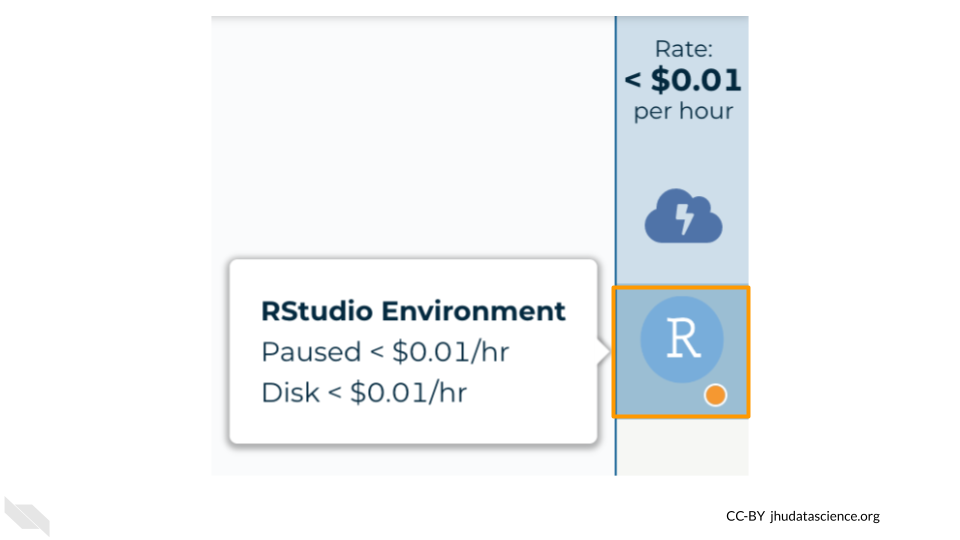
1. Hovering over the RStudio icon will show you the costs associated with your RStudio environment. Click on the RStudio icon to open the cloud environment settings.

* 

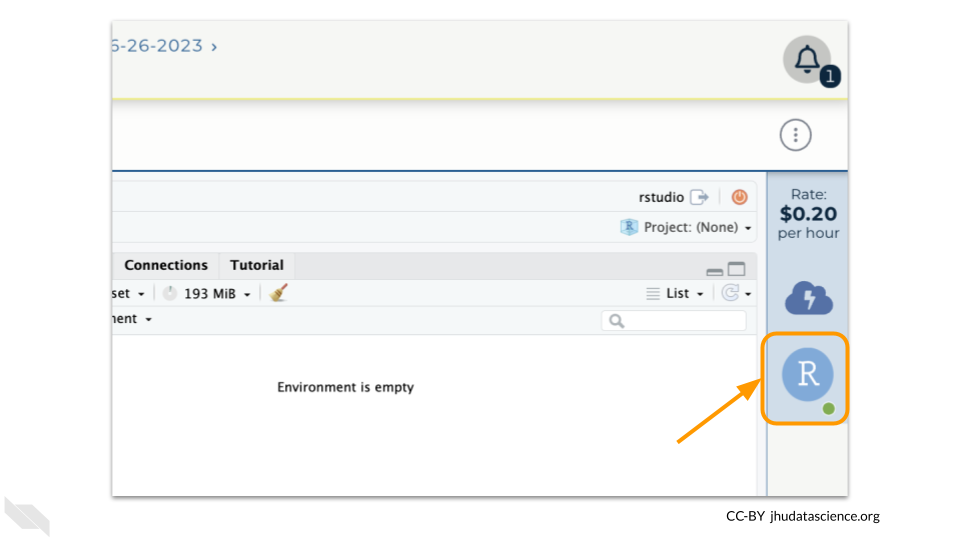
1. Click the Pause button to pause RStudio. This will take a few minutes.

* 

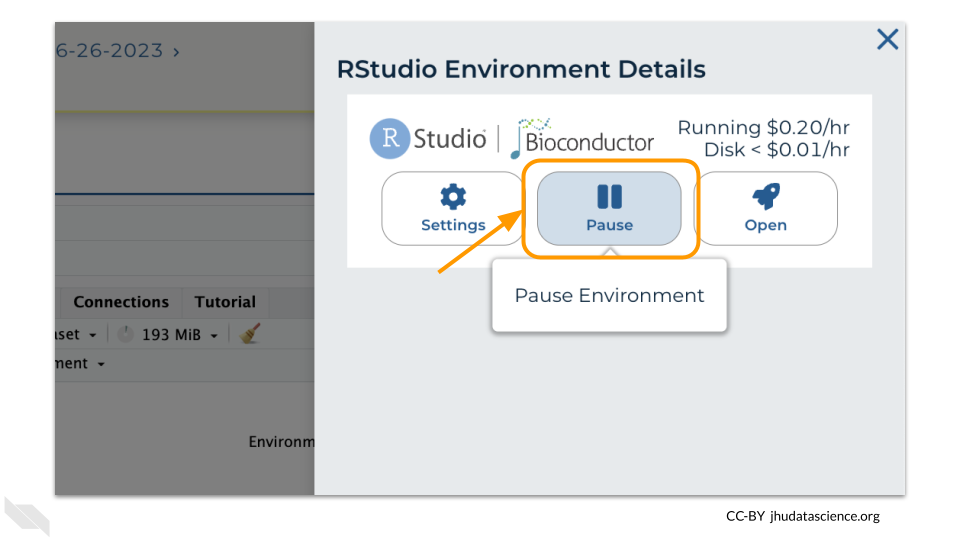
1. When the environment is paused, an orange dot will be displayed next to the RStudio icon. If you hover over the icon, you will see that it is paused, and has a small ongoing cost as long as it is paused. When you’re ready to resume working, you can do so by clicking the RStudio icon and clicking Resume.

* 

1. The right-hand side icon reminds you that you are accruing cloud computing costs. If you don’t see this icon, you may need to scroll to the right.

* 

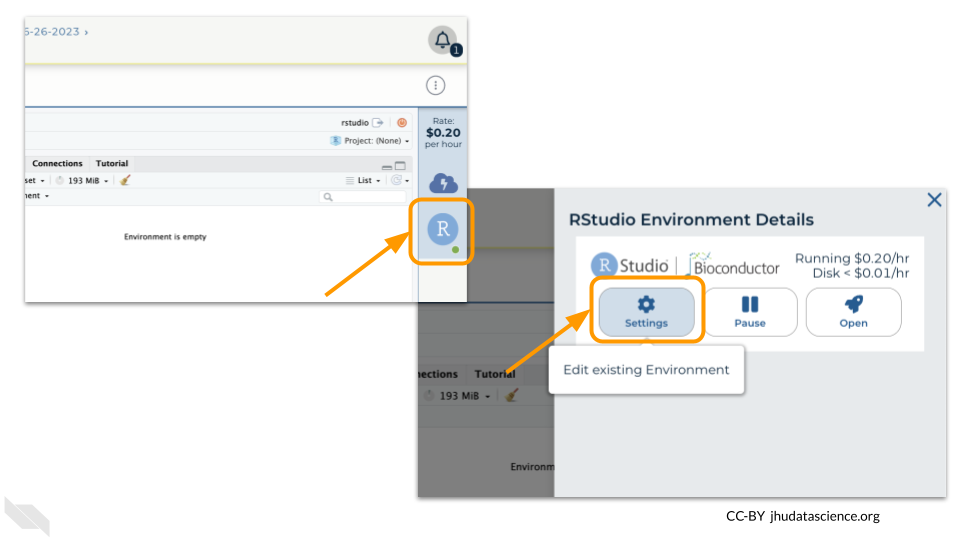
1. You should minimize charges when you are not performing an analysis. You can do this by clicking on the RStudio icon and selecting “Pause”. This will release the CPU and memory resources for other people to use. Note that your work will be saved in the environment and continue to accrue a very small cost. This work will be lost if the cloud environment gets deleted. If there is anything you would like to save permanently, it’s a good idea to copy it from your compute environment to another location, such as the Workspace bucket, GitHub, or your local machine, depending on your needs.

* 

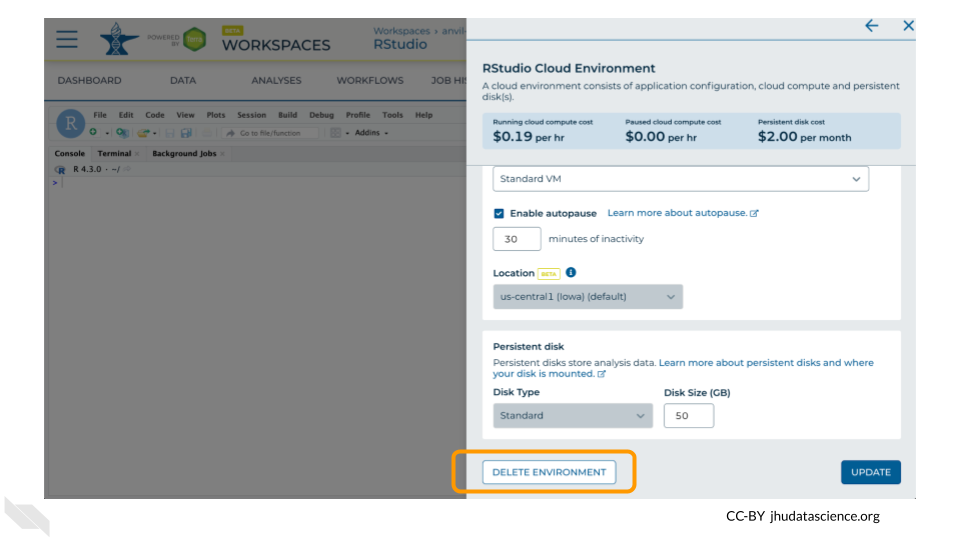
You can also pause your cloud environment(s) at <https://anvil.terra.bio/#clusters>.

## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output  
## Warning: Chunk option fig.alt is not supported for docx output

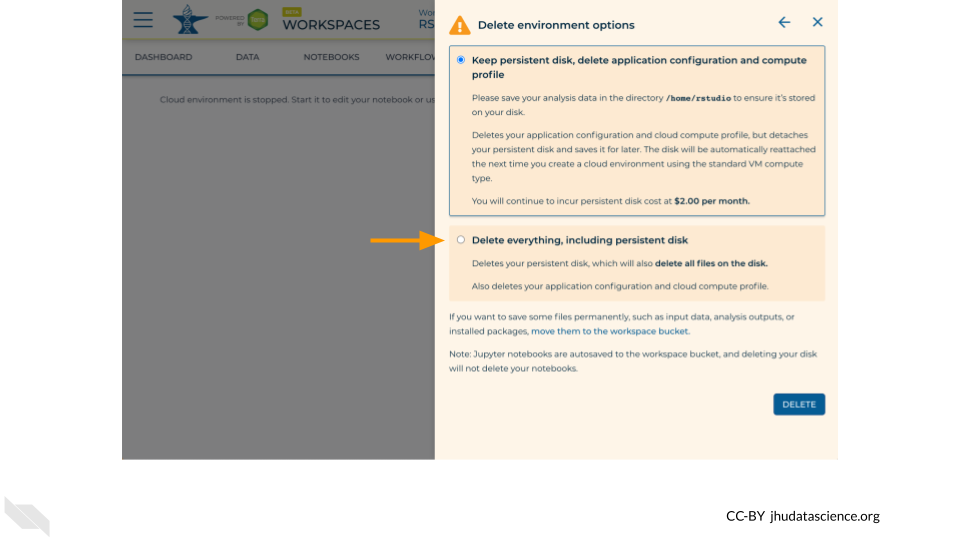
1. Pausing your cloud environment only temporarily stops your work. When you are ready to delete the cloud environment, click on the RStudio icon on the right-hand side and select “Settings”. If you don’t see this icon, you may need to scroll to the right.

* 

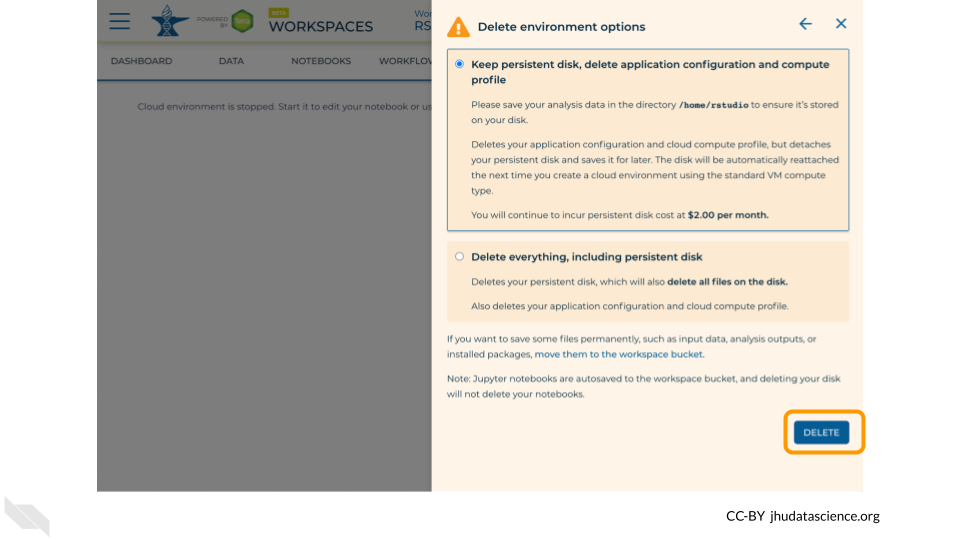
1. Click on “Delete Environment”.

* 

1. If you are certain that you do not need the data and configuration on your disk, you should select “Delete everything, including persistent disk”. If there is anything you would like to save, open the compute environment and copy the file(s) from your compute environment to another location, such as the Workspace bucket, GitHub, or your local machine, depending on your needs.

* 

1. Select “DELETE”.

* 

You can also delete your cloud environment(s) and disk storage at <https://anvil.terra.bio/#clusters>.

## Warning: Chunk option fig.align is not supported for docx output

## Warning: Chunk option fig.alt is not supported for docx output

## Warning: Chunk option fig.align is not supported for docx output

## Warning: Chunk option fig.alt is not supported for docx output

## Warning: Chunk option fig.align is not supported for docx output

## Warning: Chunk option fig.alt is not supported for docx output

## Warning: Chunk option fig.align is not supported for docx output

## Warning: Chunk option fig.alt is not supported for docx output

## Warning: Chunk option fig.align is not supported for docx output

## Warning: Chunk option fig.alt is not supported for docx output

# (PART\*) Student Activity

# 1 Download

Coming soon!

# 2 Introduction

In this activity, you’ll have a chance to become familiar with the BioDIGS soil testing data. This dataset includes information on the inorganic components of each soil sample, particularly metal concentrations. Human activity can increase the concentration of inorganic compounds in the soil. When cars drive on roads, compounds from the exhaust, oil, and other fluids might settle onto the roads and be washed into the soil. When we put salt on roads, parking lots, and sidewalks, the salts themselves will eventually be washed away and enter the ecosystem through both water and soil. Chemicals from factories and other businesses also leech into our environment. All of this means the concentration of heavy metals and other chemicals will vary among the soil samples collected for the BioDIGS project.

## 2.1 Before You Start

This activity requires RStudio. Make sure you have access to a working version of this software.

## 2.2 Objectives

This activity will teach you how to use the AnVIL platform to:

1. Open data from an R package
2. Examine objects in R
3. Calculate summary statistics for variables in the soil testing data
4. Create and interpret histograms and boxplots for variables in the soil testing data

# 3 Part 1. Examining the Data

We will use the BioDIGSData package to retrieve the data. We first need to install the package from where it is stored on GitHub.

**Packages and libraries**

You might see or hear the term “package” or “library” when working with the R programming language.

Packages are collections of R functions, data, and documentation that extend the base functionality of R. They are the fundamental units of shareable code in R. Packages are developed by the R community and made available through repositories like CRAN (Comprehensive R Archive Network), Bioconductor, and GitHub. When you install a package, you gain access to all the functions, data, and documentation provided by that package.

Libraries are the directories where packages are stored.

We also use the library command to load and attach packages to the R environment. When you load a package using library(package\_name), you make the functions and objects from a package available for use in your current R session.

You may first need to install the remotes package to your RStudio environment. remotes is a package (or chunk of pre-written code) that allows you to download code that has been stored on GitHub into RStudio.

If you’re having trouble with the code above, you can also try:

Once you’ve installed the package, we can load the package (which just means we have access to all the data stored in the BioDIGSData package). Then we assign the soil testing data to an *object*. This command follows the code structure:

dataset\_object\_name <- stored\_BioDIGS\_dataset

The “dataset\_object\_name” is what RStudio will call the dataset after you open it. The “stored\_BioDIGS\_dataset” is what the dataset is called within the BioDIGSData package. Finally, the arrow (“<-”) tells R to open the “stored\_BioDIGS\_dataset” and save it in your environment as “dataset\_object\_name”.

The order of these commands might be odd to our eyes, but it makes perfect sense to RStudio!

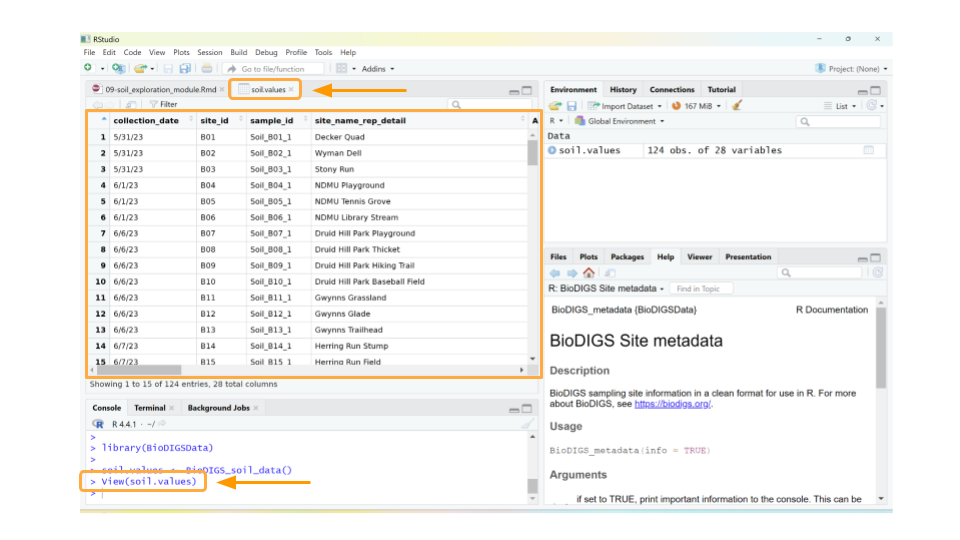
The soil testing data is called BioDIGS\_soil\_data in the BioDIGSData package. When we save this dataset into our environment, we’re calling is soil.values.

It *seems* like the dataset loaded, but it’s always a good idea to verify. There are many ways to check, but the easiest approach (if you’re using RStudio) is to look at the Environment tab on the upper right-hand side of the screen. You should now have an object called soil.values that includes some number of observations for 28 variables. The *observations* refer to the number of rows in the dataset, while the *variables* tell you the number of columns. As long as neither the observations or variables are 0, you can be confident that your dataset loaded.



Let’s take a quick look at the dataset. We can do this by clicking on soil.values object in the Environment tab. (Note: this is equivalent to typing View(soil.values) in the R console.)

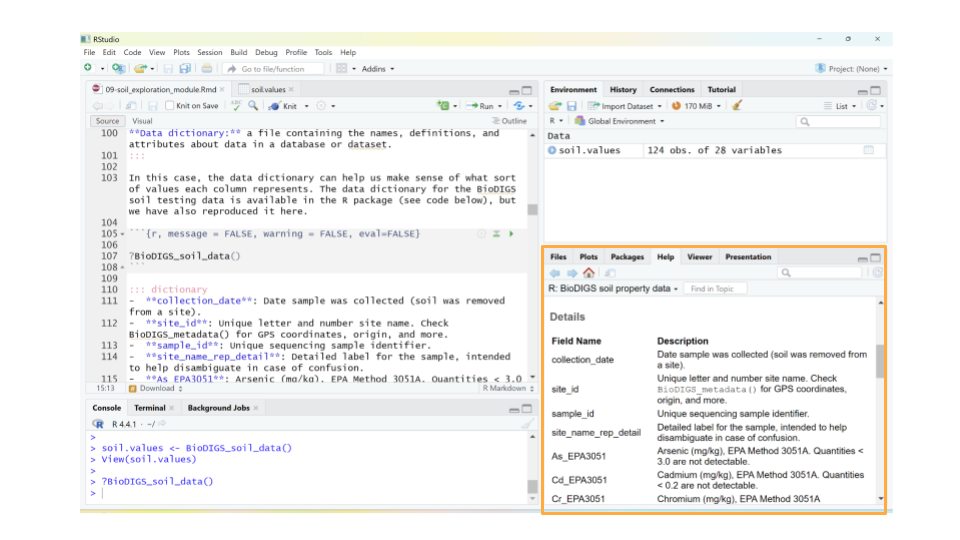
This will open a new window for us to scroll through the dataset.



Well, the data definitely loaded, but those column names aren’t immediately understandable. What could **As\_EPA3051** possibly mean? In addition to the dataset, we need to load the *data dictionary* as well.

**Data dictionary:** a file containing the names, definitions, and attributes about data in a database or dataset.

In this case, the data dictionary can help us make sense of what sort of values each column represents. The data dictionary for the BioDIGS soil testing data is available in the R package (see code below), but we have also reproduced it here.



* **collection\_date**: Date sample was collected (soil was removed from a site).
* **site\_id**: Unique letter and number site name. Check BioDIGS\_metadata() for GPS coordinates, origin, and more.
* **sample\_id**: Unique sequencing sample identifier.
* **site\_name\_rep\_detail**: Detailed label for the sample, intended to help disambiguate in case of confusion.
* **As\_EPA3051**: Arsenic (mg/kg), EPA Method 3051A. Quantities < 3.0 are not detectable.
* **Cd\_EPA3051**: Cadmium (mg/kg), EPA Method 3051A. Quantities < 0.2 are not detectable.
* **Cr\_EPA3051**: Chromium (mg/kg), EPA Method 3051A
* **Cu\_EPA3051**: Copper (mg/kg), EPA Method 3051A
* **Ni\_EPA3051**: Nickel (mg/kg), EPA Method 3051A
* **Pb\_EPA3051**: Lead (mg/kg), EPA Method 3051A
* **Zn\_EPA3051**: Zinc (mg/kg), EPA Method 3051A
* **water\_pH**: Water pH
* **OM\_by\_LOI\_pct**: Organic Matter by Loss on Ignition
* **P\_Mehlich3**: Phosphorus (mg/kg), using the Mehlich 3 soil test extractant
* **K\_Mehlich3**: Potassium (mg/kg), using the Mehlich 3 soil test extractant
* **Ca\_Mehlich3**: Calcium (mg/kg), using the Mehlich 3 soil test extractant
* **Mg\_Mehlich3**: Magnesium (mg/kg), using the Mehlich 3 soil test extractant
* **Mn\_Mehlich3**: Manganese (mg/kg), using the Mehlich 3 soil test extractant
* **Zn\_Mehlich3**: Zinc (mg/kg), using the Mehlich 3 soil test extractant
* **Cu\_Mehlich3**: Copper (mg/kg), using the Mehlich 3 soil test extractant
* **Fe\_Mehlich3**: Iron (mg/kg), using the Mehlich 3 soil test extractant
* **B\_Mehlich3**: Boron (mg/kg), using the Mehlich 3 soil test extractant
* **S\_Mehlich3**: Sulfur (mg/kg), using the Mehlich 3 soil test extractant
* **Na\_Mehlich3**: Sodium (mg/kg), using the Mehlich 3 soil test extractant
* **Al\_Mehlich3**: Aluminum (mg/kg), using the Mehlich 3 soil test extractant
* **Est\_CEC**: Cation Exchange Capacity (meq/100g) at pH 7.0 (CEC)
* **Base\_Sat\_pct**: Base saturation (BS). This represents the percentage of CEC occupied by bases (Ca2+, Mg2+, K+, and Na+). The %BS increases with increasing soil pH (Figure 5). The availability of Ca2+, Mg2+, and K+ increases with increasing %BS.
* **P\_Sat\_ratio**: Phosphorus saturation ratio. This is the ratio between the amount of phosphorus present in the soil and the total capacity of that soil to retain phosphorus. The ability of phosphorus to be bound in the soil is primary a function of iron (Fe) and aluminum (Al) content in that soil.

Using the data dictionary, we find that the values in column As\_EPA3051 give us the arsenic concentration in mg/kg of each soil sample, as determined by [EPA Method 3051A](https://www.epa.gov/sites/default/files/2015-12/documents/3051a.pdf). This method uses a combination of heat and acid to extract specific elements (like arsenic, cadmium, chromium, copper, nickel, lead, and zinc) from soil samples.

While arsenic can occur naturally in soils, higher levels suggest the soil may have been contaminated by mining, hazardous waste, or pesticide application. Arsenic is toxic to humans.

QUESTIONS:

1. What data is found in the column labeled “Fe\_Mehlich3”? Why would we be interested how much of this is in the soil? (You may have to search the internet for this answer.)
2. What data is found in the column labeled “Base\_Sat\_pct”? What does this variable tell us about the soil?

You may notice that some cells in the soil.values table contain *NA*. This just means that the soil testing data for that sample isn’t available yet. We’ll take care of those values in the next part.

QUESTIONS:

1. How many observations are in the soil testing values dataset that you loaded? What do each of these observations refer to?
2. How many different soil characteristics are in the dataset? How can you tell?

# 4 Part 2. Summarizing with Statistics

Now that we have the dataset loaded, let’s explore the data in more depth.

First, we should remove those samples that don’t have soil testing data yet. We *could* keep them in the dataset, but removing them at this stage will make the analysis a little cleaner. In this case, as we know the reason the data are missing (and that reason will not skew our analysis), we can safely remove these samples. This will *not* be the case for every data analysis.

We can remove the unanalyzed samples using the drop\_na() function from the tidyverse package. The tidyverse package is a collection of code that makes organizing data tables (also known as data wrangling) easy in R. The drop\_na function removes any rows from a table that contains *NA* for a particular column. This command follows the code structure:

dataset\_new\_name <- dataset\_object\_name %>% drop\_na(column\_name)

The %>% is called a pipe and it tells R that the commands after it should all be applied to the object in front of it. (In this case, we can filter out all samples missing a value for “As\_EPA3051” as a proxy for samples without soil testing data.)

The following code block opens the tidyverse package, then tells R to filter out all samples missing a value for “As\_EPA3051” (as a proxy for samples without soil testing data) from the soil.values object, and finally to save the new, filtered dataset as an object called soil.values.clean.

Great! Now let’s calculate some basic statistics. For example, we might want to know what the mean (average) arsenic concentration is for all the soil samples. We can use a combination of two functions: pull() and mean(). pull() lets you extract a column from your table for statistical analysis, while mean() calculates the average value for the extracted column.

This command follows the code structure:

object\_name %>% pull(column\_name) %>% mean()

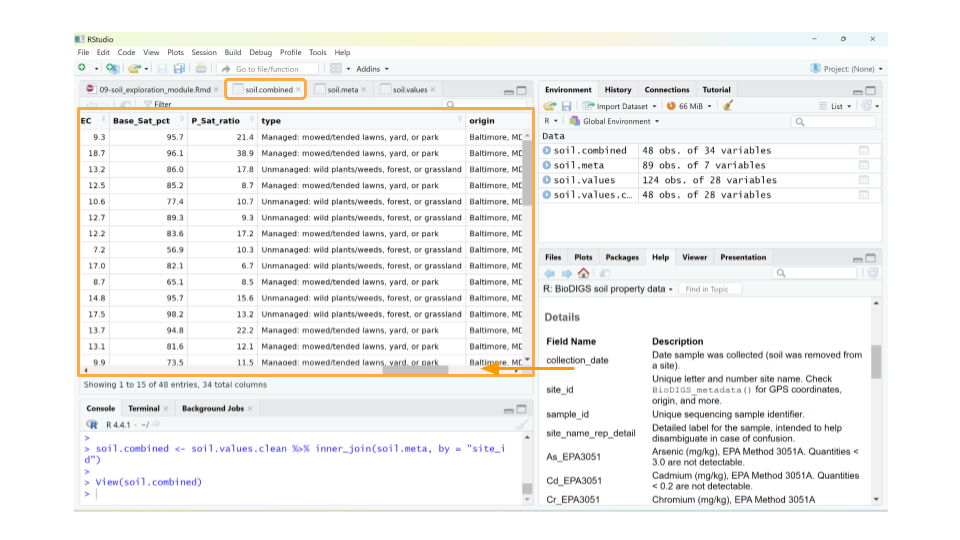
For this chunk of code, R will calculate the mean for the AS\_EPA3051 values in the soil.values.clean dataset. Because we have not used the <- to save the mean in a new object, R will simply display the mean.

[1] 5.10875

We can run similar commands to calculate the standard deviation (sd), minimum (min), and maximum (max) for the soil arsenic values.

[1] 5.606926 [1] 0 [1] 27.3 The soil testing dataset contains samples from multiple geographic regions, so maybe it’s more meaningful to find out what the average arsenic values are for each region. However, first we need to grab a new dataset from the BioDIGSData package that tells us the geographic regions of each sample.

Let’s open BioDIGS\_metadata, save it as soil.meta, and look at it.

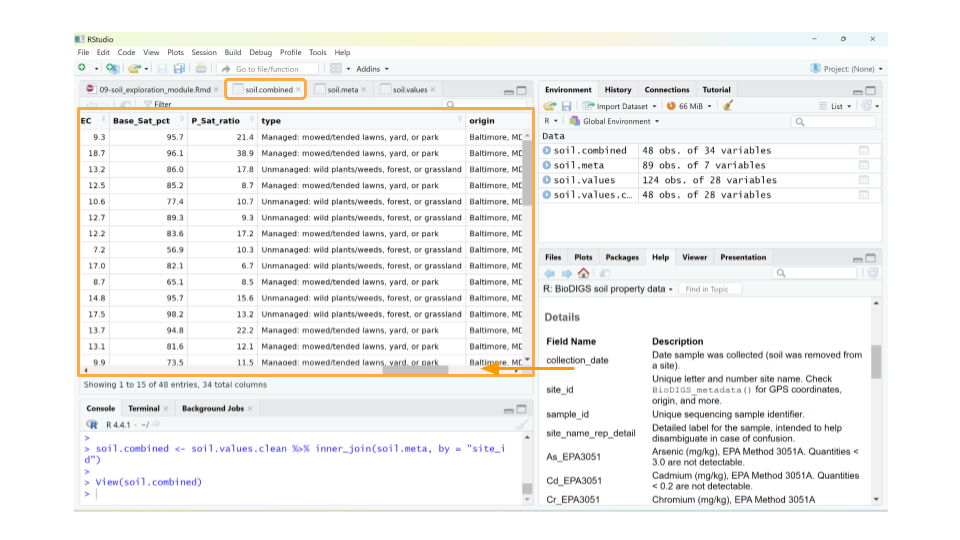


The metadata (or, data about the samples) contains information stored as 7 different variables. We can see that this dataset contains a variable called site\_id that matches a column in the soil.values and soil.values.clean datasets. This is important! Using this variable, we can combine the soil.values.clean and soil.meta into a single dataset.

The command for combining the datasets follows the code structure:

combined\_dataset\_name <- first\_dataset %>% inner\_join(second\_dataset, by = column\_name\_in\_common)

The inner\_join command tells R to combine the first\_dataset and the second\_dataset based on matching information in the column\_name\_in\_column. It also only keeps those rows that are found in both datasets. Finally, the <- tells R to save this combined dataset as a new object.



When you scroll through the soil.combined dataset, you now see the metadata columns after all the soil characteristics. In particular, there’s a column called origin which gives the town or city location for each sample. Many of the samples from the pilot study came from 5 places in Maryland: Baltimore, Derwood, Boyds, Germantown, and Bethesda.

Let’s look at the average arsenic content of the soil for each of these locations! We have to do a little bit of clever coding trickery for this using the group\_by and summarize functions.

First, we tell R to split our dataset up by a particular column (in this case, origin) using the group\_by function, then we tell R to summarize the mean arsenic concentration for each group.

When using the summarize function, we tell R to make a new table (technically, a tibble in R) that contains two columns: the column used to group the data and the statistical measure we calculated for each group.

This command follows the code structure:

dataset %>% group\_by(column\_name) %>% summarize(mean(column\_name))

# 5 A tibble: 5 × 2

origin mean(As\_EPA3051) 1 Baltimore, MD 5.56 2 Bethesda, MD 3.46 3 Boyds, MD 6.94 4 Derwood, MD 4.26 5 Germantown, MD 4.30

Now we know that the mean arsenic concentration might be different for each region of origin.

QUESTIONS:

1. All the samples in the initial pilot study were collected from public park land. Some parks were located in suburban and rural areas, while others were collected from urban parks. Why might soil arsenic concentration be different for rural parks than for urban parks?
2. What is the mean iron concentration for samples in this dataset? What about the standard deviation, minimum value, and maximum value?
3. Calculate the mean iron concentration by region of origin. Which region has the highest mean iron concentration? What about the lowest?

Let’s say we’re interested in looking at mean concentration for any element that was determined using EPA Method 3051. Given that there are 8 of these measures in the soil.combined dataset, it would be time consuming to run our code from above that calculates overall mean for each individual measure.

We can add two arguments to our summarize statement to calculate statistical measures for multiple columns at once: the across argument, which tells R to apply the summarize command to multiple columns; and the ends\_with parameter, which tells R which columns should be included in the statistical calculation.

We are using ends\_with because for this question, all the columns that we’re interested in end with the string ‘EPA3051’.

This command follows the code structure:

dataset %>% group\_by(column\_name) %>% summarize(across(ends\_with(common\_column\_name\_ending), mean))

# 6 A tibble: 5 × 8

origin As\_EPA3051 Cd\_EPA3051 Cr\_EPA3051 Cu\_EPA3051 Ni\_EPA3051 Pb\_EPA3051 1 Baltimore, … 5.56 0.359 34.5 35.0 17.4 67.2 2 Bethesda, MD 3.46 0.375 67.5 17.0 26.6 41.8 3 Boyds, MD 6.94 0.0525 16.8 16.8 12.9 31.6 4 Derwood, MD 4.26 0.335 39.5 31.3 35.1 42.1 5 Germantown,… 4.30 0.602 19.9 23.3 17.7 38.2 # ℹ 1 more variable: Zn\_EPA3051

This is a *much* more efficient way to calculate statistics.

QUESTIONS:

1. Calculate the maximum values for concentrations that were determined using EPA Method 3051. (HINT: change the function you call in the summarize statement.) Which of these metals has the maximum concentration you see, and in which region is it found?
2. Calculate both the mean and maximum values for concentrations that were determined using the Mehlich3 test. (HINT: change the terms in the the part of the code that uses ends\_with, as well as the function you call in the summarize statement.) Which of these metals has the highest average and maximum concentrations, and in which region are they found?

# 7 Part 3. Visualizing the Data

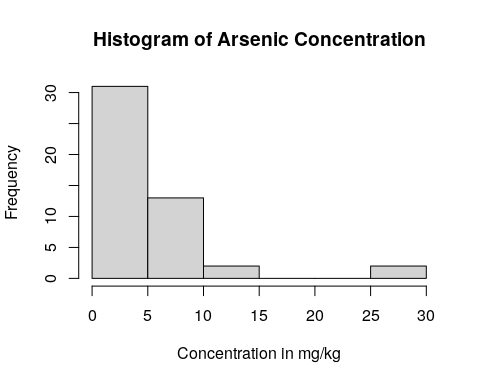
Often, it can be easier to immediately interpret data displayed as a plot than as a list of values. For example, we can more easily understand how the arsenic concentration of the soil samples are distributed if we create histograms compared to looking at point values like mean, standard deviation, minimum, and maximum.

One way to make histograms in R is with the hist() function. This function only requires that we tell R which column of the dataset that we want to plot. (However, we also have the option to tell R a histogram name and a x-axis label.)

We can again use the pull() command and pipes (%>%) to choose the column we want from the soil.values.clean dataset and make a histogram of them.

This combination of commands follows the code structure:

dataset %>% pull(column\_name) %>% hist(main = chart\_title, xlab = x\_axis\_title)



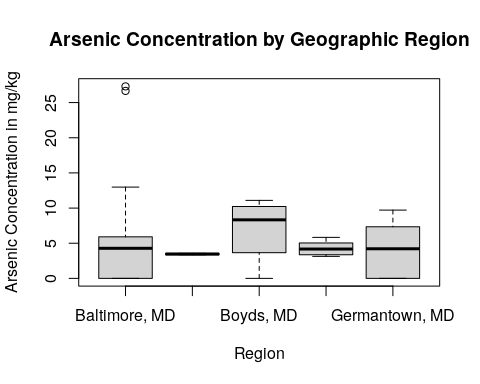
We can see that almost all the soil samples had very low concentrations of arsenic (which is good news for the soil health!). In fact, many of them had arsenic concentrations close to 0, and only a few sampling locations appear to have high levels of arsenic.

We might also want to graphically compare arsenic concentrations among the geographic regions in our dataset. We can do this by creating boxplots. Boxplots are particularly useful when comparing the mean, variation, and distributions among multiple groups.

In R, one way to create a boxplot is using the boxplot() function. We don’t need to use pipes for this command, but instead will specify what columns we want to use from the dataset inside the boxplot() function itself.

This command follows the code structure:

boxplot(column\_we’re\_plotting ~ grouping\_variable, data = dataset, main = “Title of Graph”, xlab = “x\_axis\_title”, ylab = “y\_axis\_title”)



By using a boxplot, we can quickly see that, while two sampling sites within Baltimore, MD have a very high concentration of arsenic in the soil (indicated by the two small circles on the plot), in general there isn’t a difference in arsenic content between any of our locations.

QUESTIONS:

1. Create a histogram for *iron* concentration, as well as a boxplot comparing iron concentration by region. Is the iron concentration similar among regions? Are there any outlier sites with unusually high or low iron concentrations?
2. Create a histogram for *lead* concentration, as well as a boxplot comparing lead concentration by region. Is the lead concentration similar among regions? Are there any outlier sites with unusually high or low lead concentrations?

# 8 Activity Questions

## 8.1 Part 1. Examining the Data

1. What data is found in the column labeled “Fe\_Mehlich3”? Why would we be interested how much of this is in the soil? (You may have to search the internet for this answer.)
2. What data is found in the column labeled “Base\_Sat\_pct”? What does this variable tell us about the soil?
3. How many observations are in the soil testing values dataset that you loaded? What do each of these observations refer to?
4. How many different soil characteristics are in the dataset? How can you tell?

## 8.2 Part 2. Summarizing the Data with Statistics

1. All the samples in the initial pilot study were collected from public park land. Some parks were located in suburban and rural areas, while others were collected from urban parks. Why might soil arsenic concentration be different for rural parks than for urban parks?
2. What is the mean iron concentration for samples in this dataset? What about the standard deviation, minimum value, and maximum value?
3. Calculate the mean iron concentration by region. Which region has the highest mean iron concentration? What about the lowest?
4. Calculate the maximum values for concentrations that were determined using EPA Method 3051. (HINT: change the function you call in the summarize statement.) Which of these metals has the maximum concentration you see, and in which region is it found?
5. Calculate both the mean and maximum values for concentrations that were determined using the Mehlich3 test. (HINT: change the terms in the columns\_to\_include vector, as well as the function you call in the summarize statement.) Which of these metals has the highest average and maximum concentrations, and in which region are they found?

## 8.3 Part 3. Visualizing the Data

1. Create a histogram for *iron* concentration, as well as a boxplot comparing iron concentration by region. Is the iron concentration similar among regions? Are there any outlier sites with unusually high or low iron concentrations?
2. Create a histogram for *lead* concentration, as well as a boxplot comparing lead concentration by region. Is the lead concentration similar among regions? Are there any outlier sites with unusually high or low lead concentrations?