All Code, Files, etc. are available at:

https://julesjulesjules.quarto.pub/r-workshops-with-micom/

Please download the data files.

Quick Introductions

Name, Department, Role

MICOM

MI-Support
Prepared For Anything

A data analysis, capacity building program

What is MI-Support?

MI-Support is a program offered through the Michigan Public Health Integrated Center for Outbreak Analytics and Modeling (MICOM) that aims to provide support and capacity building in response to urgent and/or under supported modeling and analytics needs for local and tribal health departments across the state of Michigan.

The MI-Support team will be deployed for data analysis projects, contributing to increased preparedness for public health emergencies and enhanced capacity for analytics and modeling.

MICOM

How does MI-Support work?

➡️ Health department needs assistance with a data analysis project that they may not have the staffing/resources to complete

➡️ Member of the health department reaches out to MI-Support via Google Form or email, providing details about the project they are looking to complete

➡️ Request enters queue for review by MI-Support team

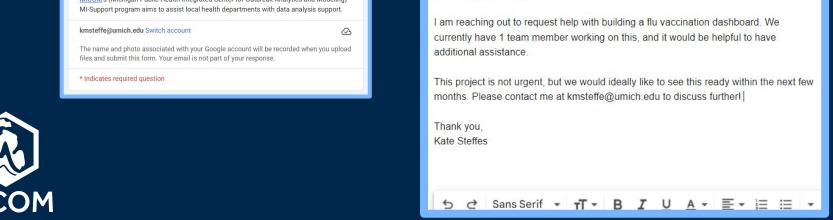➡️ MI-Support deploys team to assist with the requested project

MICOM

Submitting a MI-Support Request

→ Submit a request through our Google Form

→ Not sure about your request or want more information? Email us at micom-misupport@umich.edu

MI-Support Request Form

MICOM's (Michigan Public Health Integrated Center for Outbreak Analytics and Modeling) MI-Support program aims to assist local health departments with data analysis support.

kmsteffe@umich.edu Switch account

The name and photo associated with your Google account will be recorded when you upload files and submit this form. Your email is not part of your response.

* Indicates required question

MI-Support Project Request - Washtenaw County Health Department

Hi MI-Support Team,

I am reaching out to request help with building a flu vaccination dashboard. We currently have 1 team member working on this, and it would be helpful to have additional assistance.

This project is not urgent, but we would ideally like to see this ready within the next few months. Please contact me at kmsteffe@umich.edu to discuss further! |

Thank you,
Kate Steffes

MICOM

Introduction to R Coding

2025-02-07

Agenda:

1. Small lecture: A bit about R and some basic concepts

2. Orientation to RStudio

3. Data set-up

4. Hands-on coding!

Suggestion:

Ask questions! :)

This is meant to be a learning experience.

What is R?

https://www.r-project.org/about.html

It's a coding language, but it's a lot more too.

What can you do with R?

Create reproducible reports.

Automate tasks.

Explore data.

Generate graphs and tables.

Create models and forecasts.

Check data for errors.

Set up alerts to monitor data.

Create dashboards or websites.

… and lots more!

Some basic ideas and tenets in R:

The Code Only Does What You Tell It To Do.

It looks where you tell it to look.

It changes what you tell it to change.

Always talk through what you're doing, check what you think should be happening against what is actually happening.

This is a "new language"! It's hard sometimes, but it gets easier with practice.

R (sometimes) knows when things are wrong.

Error messages will happen! Sometimes they're easy to understand, sometimes they're not. During this workshop, if you get an error, raise your hand! We'll talk through it together.

R is case sensitive.

For R, "happy_birthday" is DIFFERENT from "Happy_Birthday".

So sometimes, code "doesn't work" because of a mis-spelling, or an errant capital or lower case letter.

Variables & Data Types

Character types: "today"

Numeric types: 1

Boolean types: TRUE, FALSE

All of these can be assigned to a variable, and referenced later in the code.

For Example

If you had an analysis to run, and you knew you'd need to reference the population of a city often, you could store that in a variable:

```
1
2  city_pop <- 110000
3
```

And use it later for all calculations. That way, your code would be reusable by another city, or for next year, when the city population might increase or decrease.

Assigning values to a variable

Traditional in R to use **<-** to assign the value on the RHS to the variable on the LHS, like we just did

```
1
2    city_pop <- 110000
3
```

Can also use **=**

Both will basically do the same thing!

Manipulating a Variable

You can change the values or contents of variables, too.

For numerics, you might do some math:

```
> city_pop <- 110000
>
> city_pop <- city_pop - 50000
>
> city_pop
[1] 60000
>
```

Manipulating a Variable

Or for character strings, you might add on new items.

```
> todays_message <- "Hello"
>
> todays_message <- paste0(todays_message, " and goodbye.")
>
> todays_message
[1] "Hello and goodbye."
>
```

Data Structures

Vectors

Dataframes

Factors

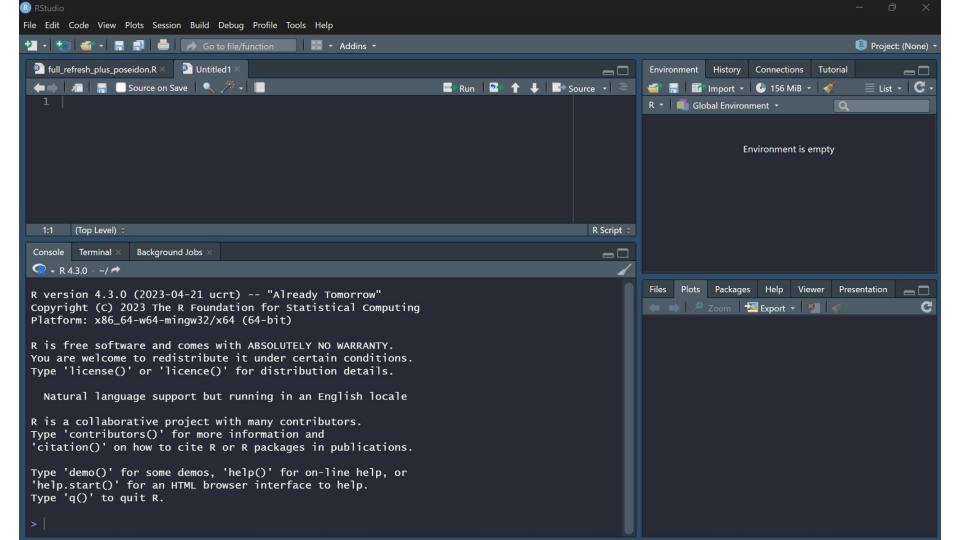There are more, but these are the only ones we'll need for today.

Data Structures

Vectors are ordered lists. We can add or remove items from these lists, change values in the list, and reference specific items within the list.

```
> groceries <- c("apples", "bread", "ice cream")
>
> groceries[2]
[1] "bread"
>
> groceries[3] <- "spinach"
>
> groceries
[1] "apples"  "bread"    "spinach"
>
> groceries <- c(groceries, "beans")
>
> groceries
[1] "apples"  "bread"    "spinach" "beans"
>
```

Data Structures

Data frames are collections of vectors, but are essentially data tables that can be referenced by their rows and columns.

```
doggy_day_care <- data.frame(dog_breed = c("German Shepard", "Labrador", "Beagle"),
                             dog_name = c("Captain", "Moose", "Richard"))
```

	dog_breed	dog_name
1	German Shepard	Captain
2	Labrador	Moose
3	Beagle	Richard

Let's write some code!
Follow along here, but we'll write code from scratch in RStudio.

R_workshop_demo_code_20250207.html