Teaching Introductory Statistics with R/RStudio

Holyoke Community College

May 22 – 26, 2023

A VERY WARM WELCOME!

Thank you for Joining Us!

Introductions! ☺

Name? College? Reasons for taking this workshop? Any fun summer plans?

NSF GRANT: DSC-WAV

Multifaceted project impacting local organizations and providing valuable hands-on data science experience for undergraduate students. Two objectives:

1) Data Science WAV teams of specially-trained undergraduate students deployed to community-based organizations to Wrangle, Analyze, and Visualize their data.

2) Offer summer faculty development workshops designed to help new instructors, especially those at community colleges, teach data science at their institutions.

https://www.nsf.gov/awardsearch/showAward?AWD_ID=1924017&HistoricalAwards=false
https://dsc-wav.github.io/www/index.html

GAISE REPORT

Guidelines for Assessment & Instruction in Statistics Education

American Statistical Association; Endorsed by AMATYC

“Technology has changed the practice of statistics and hence should change what and how we teach.”

“By adopting the best available tools (subject to institutional constraints), we allow students to do analysis more easily and therefore open up time to focus on interpretation of results … rather than on computational mechanics.”

“Technology should aid students in learning to think statistically and to discover concepts. It should also facilitate access to real (and often large) datasets and foster active learning.”

RStudio has emerged as a technology of choice for addressing these guidelines and engaging the modern student.

https://www.amstat.org/asa/files/pdfs/GAISE/GaiseCollege_Full.pdf
https://www.amstat.org/docs/default-source/amstat-documents/2016-amatycgaise.pdf

The R Programing Language

- First Release 1993 (Ihaka and Gentleman)
- Successor of S Language (1976)
- Wide use in Academia, Research & Industry
- Statistics, Data Analytics, Data Science, Data Mining, Bioinformatics, AI, Machine Learning
- TIOBE Index Ranking #8 - #16 since 2020, monthly language popularity

https://www.tiobe.com/tiobe-index/r/

RStudio

- Integrated Development Environment (IDE) for the R language
- First Release 2011
- Desktop/**Server (Workbench)**/Cloud
- R Markdown – generate professional documents with text/code/graphics
- Multilingual – supports Python, SQL, JavaScript, CSS and others

R PACKAGES

- Extensions to the R Language that has allowed the language itself and the user experience to dynamically grow and evolve.
- Similar to smartphone apps. A new phone can do many things, but the real power, utility and fun comes from our favorite apps which expands its capabilities.

blogdown
bookdown
broom
covr
dbplot
devtools
dplyr
feather
forcats
fs
ggplot2
glue
googledrive
googlesheets
haven
hms
knitr
lobstr
lubridate
modeldb
modelr
parsnip
%>%
Ceci n'est pas une pipe
pkgdown
plumber
purrr
R2D3
readr
readxl
recipes
reprex
rlang
rmarkdown
roxygen2
rsample
RStudio
rvest
scales
Shiny
sparklyr
stringr
test that
TIBBLE
tidymodels
tidyposterior
tidypredict
tidyr
tidyverse
usethis
vctrs
withr
xaringan
yardstick

Three Distinct Coding Styles

- **BASE R - The original R Language**
  - A Distinctive Code Feature: "Dollar Sign" **$**
  - Little to no use in our MTH142
  - We tend to avoid this unless it's unavoidable.
- **FORMULA BASED ("Mosaic" coding)**
  - A Distinctive Code Feature: "Tilda" ~
  - Prof. Nick Horton of Amherst College & colleagues developed mosaic package.
  - Excellent for beginners. We use this extensively in MTH142.
- **Tidyverse**
  - A Distinctive Code Feature "Pipe" `%>%` or `|>`
  - Ecosystem of related packages. The de facto language of Data Science in R.
  - Our new MTH190 course is centered around the Tidyverse.
  - We use it minimally/secretly in MTH142 but aspire to use it more.

R PACKAGES

- Currently ≈ 19,516 available packages! (CRAN repository).
- In our courses, we only use a few (less than 10?)
- Our “Big Three”:
    - **Openintro** → Contains many **data frames** related to Openintro textbook.
    - **Tibble** → Contains **view** function for viewing complete data frames.
    - **Mosaic** → Small toolbox of useful **functions**. Simple, uniform template.

“*Less Volume, More Creativity*”

Mosaic Package

Many **different tools**, but with a very **similar style**.

Basic structure of MOSAIC coding:

- One variable x: `function(~ x, data = . . .)`
- Two numerical variables: `function(y ~ x, data = . . .)`
- Two or more categorical variables: `function(~ x + y, data = . . .)`
- Think of all of these as: `Do Something(~ To some Variables, From some Dataset)`

→ "`function`" will be replaced by the name of some **action** we want to do.

→ `x` and `y` are replaced with names of any variables (columns) in the data set.

→ "`data = . . .`" state data set being used. Let's R know where the variables are located.

RStudio (Posit) Workbench

HCC hosts an RStudio Workbench server, installed/maintained by us.

Access from any web browser with a simple login.

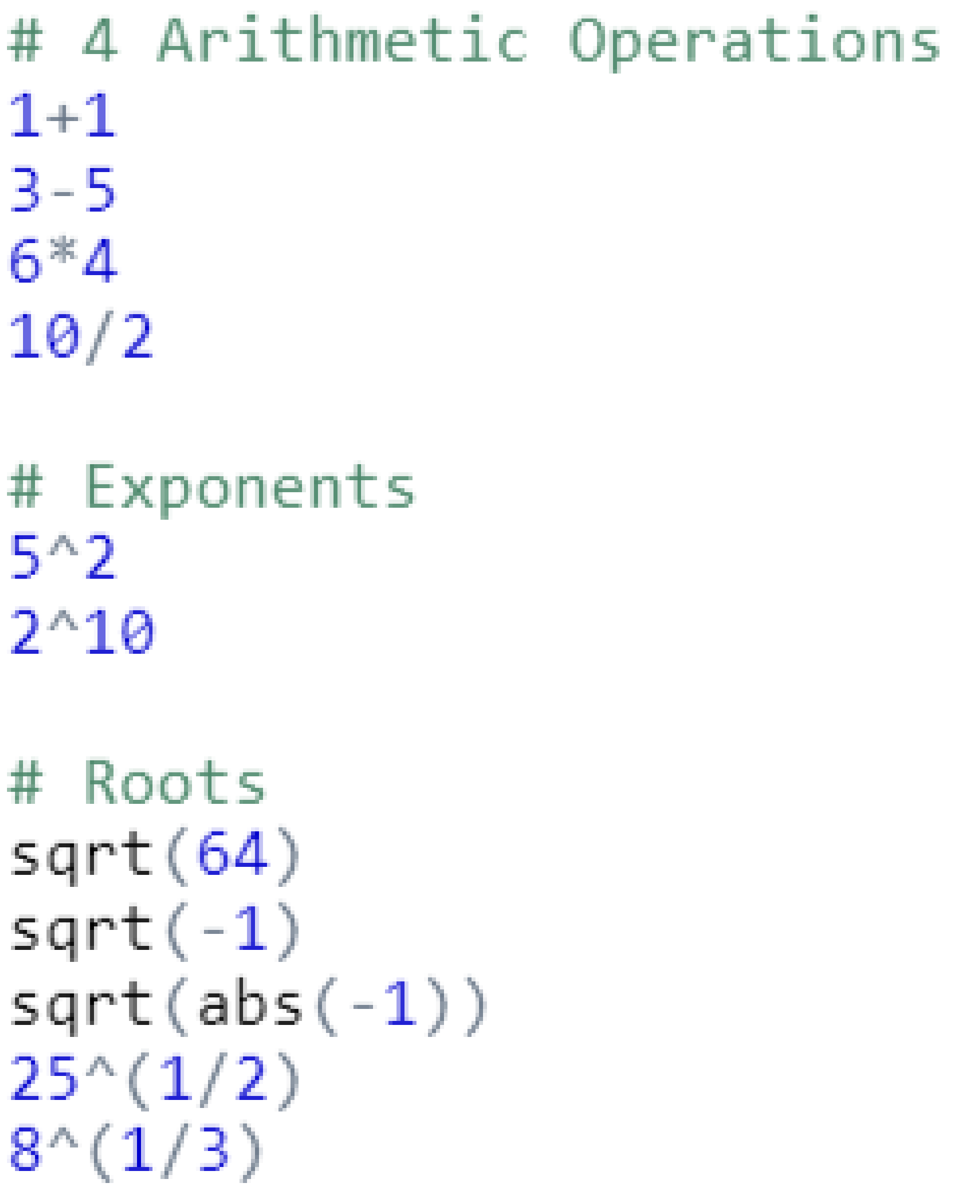
Critical for Equity: all students can access RStudio from any connected device.

Cannot install R/RStudio on Chromebook or tablet, which many students rely on.

Courses have day-one access, circumventing cumbersome installation process (and the myriad tech problems) of having students install on their personal computers Base R, RStudio and all necessary packages.

Workbench: Free academic license. Cloud: available for a fee (hosted by Posit.)

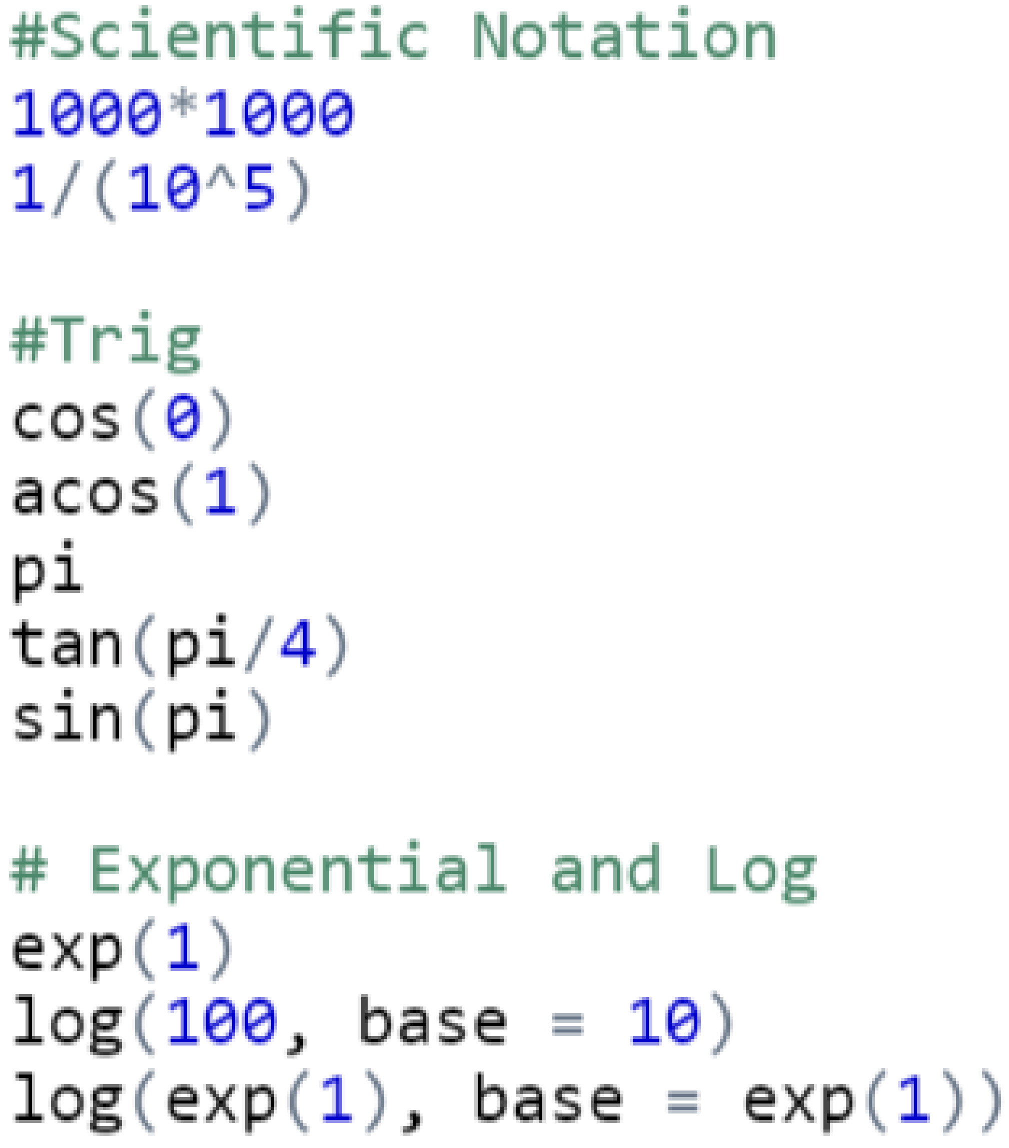
Start Simple: R as a Giant Calculator!



```
# 4 Arithmetic Operations
1+1
3-5
6*4
10/2

# Exponents
5^2
2^10

# Roots
sqrt(64)
sqrt(-1)
sqrt(abs(-1))
25^(1/2)
8^(1/3)
```

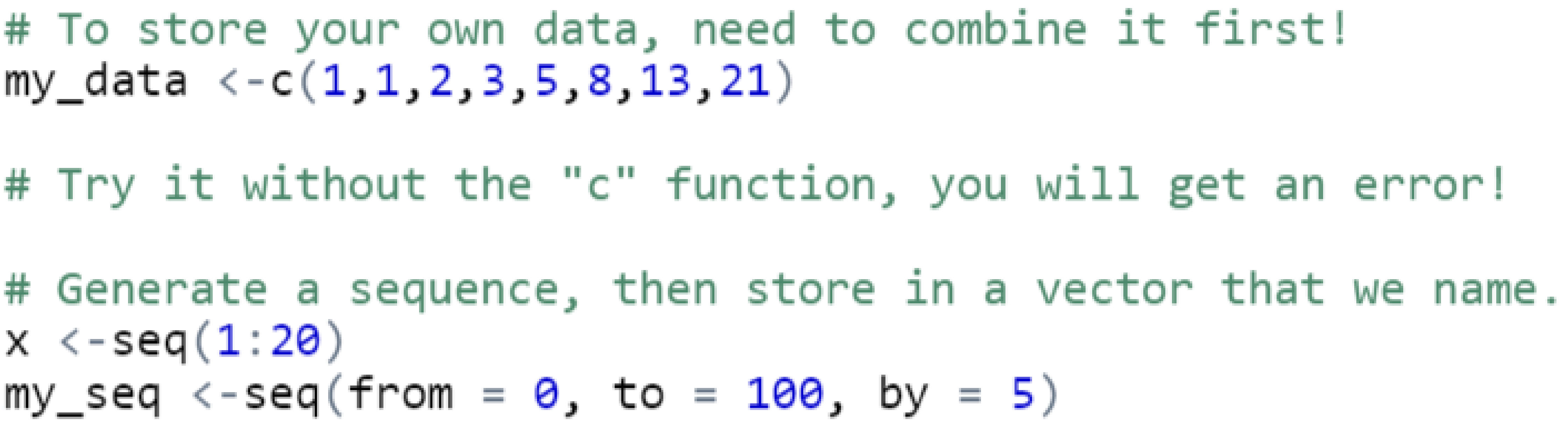


```
#Scientific Notation
1000*1000
1/(10^5)

#Trig
cos(0)
acos(1)
pi
tan(pi/4)
sin(pi)

# Exponential and Log
exp(1)
log(100, base = 10)
log(exp(1), base = exp(1))
```

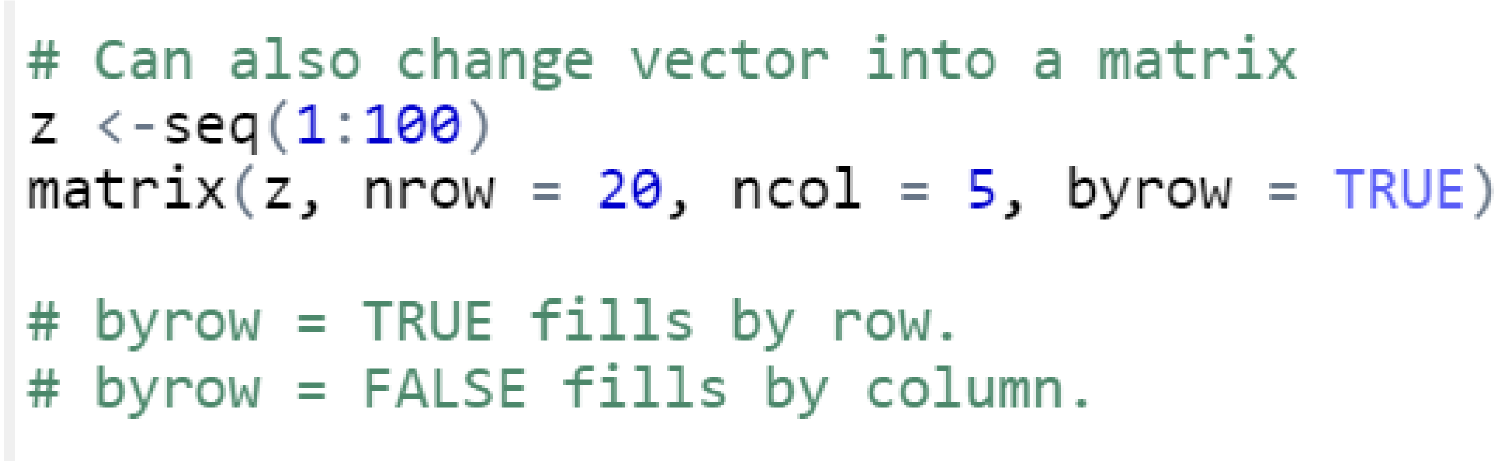
Vectors – Creating a Data Variable



```
# To store your own data, need to combine it first!
my_data <-c(1,1,2,3,5,8,13,21)

# Try it without the "c" function, you will get an error!

# Generate a sequence, then store in a vector that we name.
x <-seq(1:20)
my_seq <-seq(from = 0, to = 100, by = 5)
```



```
# Can also change vector into a matrix
z <-seq(1:100)
matrix(z, nrow = 20, ncol = 5, byrow = TRUE)

# byrow = TRUE fills by row.
# byrow = FALSE fills by column.
```