Intro to R: base-R

Basic Building Blocks

Time: 7 minutes

- `x <-`: variable storage
- `c()`: creating vectors
- `?`: accessing R's built-in help files
- `+, -, *, /, ^`: basic arithmetic functions
- `sqrt()`: square root

Workspace and Files

Time: 10 minutes

- `getwd()`: get working directory
- `ls()`: list objects in workspace
- `list.files(), dir()`: list files in working directory
- `args()`: see arguments a function takes in
- `dir.create()`: create directory within working directory
- `setwd()`: set working directory
- `file.create()`: creates R file
- `file.exists()`: checks if file exists in working directory
- `file.info()`: access information about file
- `file.rename()`: rename file
- `file.copy()`: generates a copy of file
- `file.path()`: provides relative path for file

Sequences of Numbers

Time: 5 minutes

- `:`: creates a sequence of numbers in increments of 1
- `seq()`: creates a sequence of numbers with more control than ":"
- `length()`: returns the length of a vector
- `seq_along()`: creates a sequence of numbers from 1 to N
- `rep()`: creates a sequence of specified repetitive numbers

Vectors

Time: 8 minutes

- `<, <=, >, >=, ==, !=`: logical operators

- `!`: negates
- `|`: or
- `&`: and
- `paste()`: creates character strings and concatenates

Missing Values

Time: 6 minutes

- `rnorm()`: creates a vector of N draws from a standard normal distribution
- `sample()`: randomly selects N numbers from a set
- `is.na()`: returns T/F if an element is "NA"
- `sum()`: counts total

Subsetting Vectors

Time: 8 minutes

- `names()`: returns or creates name attribute to object
- `identical()`: tests two objects to identify if they are exactly equal

Matrices and Data Frames

Time: 9 minutes

- `dim()`: gets or sets the dimension attribute for an object
- `attributes()`: accesses an object's attributes
- `class()`: returns type of obejct
- `matrix()`: creates a matrix (can only contain one type of class)
- `cbind()`: combines columns
- `data.frame()`: creates a dataframe (can contain multiple types of classes)
- `colnames()`: sets column names to a matrix-like object

Looking at Data

Time: 11 minutes

- `nrow()`: returns number of rows
- `ncol()`: returns number of columns
- `object.size()`: returns how much space the object is occupying in memory
- `head()`: previews top entries of a dataset
- `tail()`: previews bottom entries of a dataset

- `summary()`: prints various result summaries (ex. mean, quartiles, min, max, etc. depending on object type)
- `str()`: concise function which returns summary features for objects

Base Graphics

Time: 9 minutes

- `plot()`: R tries to create a helpful graph given input data; defaults to scatterplot
- `boxplot()`: produces box-and-whisker plots
- `hist()`: produces histograms

Working with the Tidyverse

Manipulating Data with dplyr

Time: 18 minutes

- `packageVersion()`: returns package version
- `tbl\_df()`: creates type object tbl_df
- `rm()`: removes object
- `dplr::select()`: keeps only the variables mentioned
- `dplr::filter()`: finds rows where conditions met are true
- `dplr::arrange()`: orders rows in a dataset, defaults to ascending
- `desc()`: descending order
- `dplr::mutate()`: adds new variables
- `dplr::summarize()`: reduces multiple values to a single value; typically used on group_by()

Grouping and Chaining with dplyr

Time: 23 minutes

- `dplr::group_by()`: groups tbls; methods applied to this output will be by group
- `%>%`: controller for chaining; line breaks
- `dplr::n()`: number of observations in the current group
- `dplr::n_distinct()`: number of unique observations in the current group
- `quantile()`: produces quantiles from an inputted dataset corresponding to the given probabilities
- `View()`: invokes spreadsheet-like dataviewer; allows to see all data points

Tidying Data with tidyr

Time: 25 minutes

- `tidyr::gather()`: gathers columns into key-value pairs
- `tidyr::separate()`: separates one column into multiple columns
- `tidyr::spread()`: spreads key-value pairs across multiple columns
- `readr::parse_number()`: drops non-numeric characters from a string
- `dplyr::bind_rows()`: binds multiple dataframes together, adds rows
- `dplyr::contains()`: select variable if it contains a literal string