Exploring Data Frames

At this point, you’ve seen it all: in the last lesson, we toured all the basic data types and data structures in R. Everything you do will be a manipulation of those tools. But most of the time, the star of the show is the data frame—the table that we created by loading information from a csv file. In this lesson, we’ll learn a few more things about working with data frames.

library(readxl)  
wetland <- read\_excel("data/WetPlots.xlsx")

We can find the size and various other attributes of our data with a few handy functions:

nrow(wetland)

## [1] 32

ncol(wetland)

## [1] 10

dim(wetland)

## [1] 32 10

names(wetland)

## [1] "fid" "newid" "wetland\_id" "wpt"   
## [5] "date" "surveyors" "slope\_position" "aspect"   
## [9] "slope\_pc" "comments"

str(wetland)

## Classes 'tbl\_df', 'tbl' and 'data.frame': 32 obs. of 10 variables:  
## $ fid : chr "111563" "8798" "8475" "163604" ...  
## $ newid : num 40076 36819 36353 46206 7386 ...  
## $ wetland\_id : chr "(Headwaters to) Gitanyow Lake" "Morin Lake ID: 13599" "Chapman Lake South (19349)" "Kaldo Lake" ...  
## $ wpt : num 157 122 121 NA 145 NA NA 123 NA 142 ...  
## $ date : chr "2019-08-30" NA NA "2019-08-24" ...  
## $ surveyors : chr "Don, Dustin, Neil" NA NA "Carlos, Jasmine" ...  
## $ slope\_position: chr NA NA NA "UP" ...  
## $ aspect : chr "S" NA NA "NE" ...  
## $ slope\_pc : chr "2" NA NA "3" ...  
## $ comments : chr "157" "122" "121" NA ...

A way to get a nice overall view of a data frame is to use the summary function. This function can be used on various objects in R. For data frames, summary yields a numeric, tabular, or descriptive summary of each column. Factor columns are summarized by the number of items in each level, numeric or integer columns by the descriptive statistics (quartiles and mean), and character columns by its length, class, and mode.

summary(wetland)

## fid newid wetland\_id wpt   
## Length:32 Min. : 6598 Length:32 Min. :121.0   
## Class :character 1st Qu.:32620 Class :character 1st Qu.:122.5   
## Mode :character Median :38448 Mode :character Median :142.0   
## Mean :33802 Mean :136.6   
## 3rd Qu.:42902 3rd Qu.:145.5   
## Max. :48222 Max. :157.0   
## NA's :25   
## date surveyors slope\_position aspect   
## Length:32 Length:32 Length:32 Length:32   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## slope\_pc comments   
## Length:32 Length:32   
## Class :character Class :character   
## Mode :character Mode :character   
##   
##   
##   
##

A really nice feature of RStudio is that it allows us to view our data in a very readable and familiar way. Go ahead and click on the name of the wetland data frame in the environment pane.

Note that this is a read-only view, it’s not editable (which is a good thing!)

## Lets look into the rest of the data

So far, you have seen the basics of manipulating data frames with our wetland data; however we are only reading in the first sheets. Let’s use those skills to read in and investigate the rest of the wetland data

## Challenge 1

Read in sheet 2 “WetVegPlots\_DF” dataset using read\_excel and assign it to a data frame called veg. Hint use ?read\_excel to see how to read in the second sheet. The first thing we should always do is check out what the data looks like with str or summary:

At this stage, it’s important to ask ourselves if the structure R is reporting matches our intuition or expectations; do the basic data types reported for each column make sense? If not, we need to sort any problems out now before they turn into bad surprises down the road, using what we’ve learned about how R interprets data, and the importance of *strict consistency* in how we record our data.

Once we’re happy that the data types and structures seem reasonable, it’s time to start digging into our data proper. Check out the first few lines:

library(readxl)  
veg <- read\_excel("data/WetPlots.xlsx", 3)  
  
head(veg)

## # A tibble: 6 x 8  
## fid plot.id species.name latin pc\_cover gt\_3m eq\_1to3 lt\_1m  
## <chr> <dbl> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 111563 1 Sitka sedge/ Water~ carex aquatil~ 10 NA NA NA  
## 2 111563 1 Calamograsis canad~ calamograsis ~ 15 NA NA NA  
## 3 111563 1 Mannagrass sp. glyceria sp 5 NA NA NA  
## 4 111563 1 Water arum calla palustr~ 5 NA NA NA  
## 5 111563 1 Pink Spirea sprea salicif~ 5 NA NA 100  
## 6 111563 1 Mountain alder alnus incana 10 90 NA NA

## Challenge 2

It’s good practice to also check the last few lines of your data and some in the middle. How would you do this?

Searching for ones specifically in the middle isn’t too hard but we could simply ask for a few lines at random. How would you code this?

Another very helpful function for looking at your data is the unique function, to see the unique values in a particular column:

unique(veg$species.name)  
sort(unique(veg$species.name))

To make sure our analysis is reproducible, we should put the code into a script file so we can come back to it later.

## Challenge 3

Go to file -> new file -> R script, and write an R script to load in the veg plot dataset. Put it in the scripts/ directory.

Run the script using the source function, using the file path as its argument (or by pressing the “source” button in RStudio).

## Basic subsetting

Data frames are two dimensions - rows and columns. We can access these using a specific notation with square brackets.

my\_data\_frame[rows, columns]

### Selecting rows

We can select rows using the square brackets like so:

veg[1, ]

## # A tibble: 1 x 8  
## fid plot.id species.name latin pc\_cover gt\_3m eq\_1to3 lt\_1m  
## <chr> <dbl> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 111563 1 Sitka sedge/ Water s~ carex aquat~ 10 NA NA NA

veg[100:110, ]

## # A tibble: 11 x 8  
## fid plot.id species.name latin pc\_cover gt\_3m eq\_1to3 lt\_1m  
## <chr> <dbl> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 236997 1 Tealeaf Willow salix pulchra 30 NA 90 10  
## 2 236997 1 Scrub Birch betula nana 15 NA 100 NA  
## 3 236997 1 Black Spruce picea mariana 30 80 20 NA  
## 4 236997 1 Labrador Tea <NA> 15 NA NA NA  
## 5 236997 1 Current - high bus~ viburnum tri~ T NA NA NA  
## 6 236997 1 Horse tail sp. equisetum sp 10 ? NA NA NA  
## 7 236997 1 Bluejoint Grass calamagrosti~ 5 NA NA NA  
## 8 236997 1 Sphagnum sp. spagnum sp 100 NA NA NA  
## 9 236997 1 Sedge sp. carex sp dispers~ NA NA NA  
## 10 191640 1 Black Spruce picea mariana 70 NA NA 100  
## 11 191640 1 Sitka Willow salix sitche~ 15 33 33 33

veg[c(1, 3, 5, 7, 9), ]

## # A tibble: 5 x 8  
## fid plot.id species.name latin pc\_cover gt\_3m eq\_1to3 lt\_1m  
## <chr> <dbl> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 111563 1 Sitka sedge/ Water ~ carex aquati~ 10 NA NA NA  
## 2 111563 1 Mannagrass sp. glyceria sp 5 NA NA NA  
## 3 111563 1 Pink Spirea sprea salici~ 5 NA NA 100  
## 4 111563 1 Salix Sp. (Pacific?) <NA> T NA 80 20  
## 5 111563 1 Open water <NA> 5 NA NA NA

It’s unlikely that you’ll often do this directly though, you’ll usually want to use some criteria. Say we just wanted the rows from Asia:

#willow <- veg[veg$species.name == "Scrub Birch", ]  
p1 <- veg[veg$plot.id == 1, ]

This works because subsetting can take either numerical positions like we showed above, OR a vector of TRUE and FALSE, returning the rows that are TRUE. veg$plot.id == 1 generates a vector of TRUE and FALSE values:

veg$species.name == "Scrub Birch"

## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [49] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
## [61] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [73] NA FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [85] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE  
## [97] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [109] FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE  
## [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [157] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [181] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [193] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [205] FALSE FALSE FALSE FALSE FALSE NA FALSE FALSE FALSE FALSE FALSE FALSE  
## [217] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE NA FALSE  
## [229] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [241] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE  
## [253] FALSE TRUE FALSE FALSE FALSE FALSE FALSE

and inserting that vector into the first position of the square brackets gives us just the rows where the species.name column is "Scrub Birch".

Another very useful operator in this context is the %in% operator - it can tell you if a value is *one of* a set of values:

Say we wanted to select just Canada and Mexico:

veg$species.name %in% c("Pacific Willow", "Sitka Willow")

## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [73] FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [109] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [145] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [157] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [181] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [193] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE  
## [217] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [229] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [241] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [253] FALSE FALSE FALSE FALSE FALSE FALSE FALSE

Now we can insert this into the first position of the square brackets:

veg\_willow <- veg$species.name %in% c("Pacific Willow", "Sitka Willow")  
veg[veg\_willow, ]

## # A tibble: 5 x 8  
## fid plot.id species.name latin pc\_cover gt\_3m eq\_1to3 lt\_1m  
## <chr> <dbl> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 44835 1 Pacific Willow salix lucida 80 80 20 0  
## 2 44835 2 Sitka Willow salix sitchensis 80 NA NA 100  
## 3 191640 1 Sitka Willow salix sitchensis 15 33 33 33  
## 4 111036 1 Pacific Willow salix lucida <NA> NA NA NA  
## 5 111036 1 Sitka Willow salix sitchensis <NA> NA NA NA

## Challenge 4

Select all the rows in vegfrom plot 4:

We saw previously how we can select a single column using the dollar sign $:

veg$plot.id

## [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 1 1 1 1  
## [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [75] 1 1 1 1 2 2 2 2 3 1 1 1 1 1 1 1 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [112] 1 1 1 1 2 2 2 2 2 2 1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 3  
## [149] 3 3 3 4 4 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## [223] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 1 1 1 1 1 2 2 2 2 2 2 2

If we wanted to select a subset of columns, we can specify that in the second position using a character vector of column names:

veg[ , c("plot.id", "pc\_cover")]

## # A tibble: 259 x 2  
## plot.id pc\_cover  
## <dbl> <chr>   
## 1 1 10   
## 2 1 15   
## 3 1 5   
## 4 1 5   
## 5 1 5   
## 6 1 10   
## 7 1 T   
## 8 1 15   
## 9 1 5   
## 10 1 T   
## # ... with 249 more rows

## Challenge 5

Select all the rows in veg with data from P3, and just the columns fid, species.name and pc\_cover

### Solution to Challenge 1

Read in the third sheet from the excel “WetPlots.xlsx” in the data folder

veg <- read\_excel("data/WetPlots.xlsx", 3)  
 veg

Then tell me how many rows and columns veg has:

dim(veg)  
 nrow(veg)  
 ncol(veg)

And tell me what kind of data is in the fid and utm\_zone columns

str(veg$fid)  
 str(veg$utm\_zone)  
 str(veg)

### Solution to Challenge 2

To check the last few lines it’s relatively simple as R already has a function for this:

tail(veg)

## # A tibble: 6 x 8  
## fid plot.id species.name latin pc\_cover gt\_3m eq\_1to3 lt\_1m  
## <chr> <dbl> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 19670 2 Scrub Birch betula nana 20 NA 20 80  
## 2 19670 2 Sphagnum sp. spagnum sp 90 NA NA NA  
## 3 19670 2 Beaked Sedge ccarex rostrata 5 NA NA NA  
## 4 19670 2 Calamagrostis calamagrostis sp 4 NA NA NA  
## 5 19670 2 Lodgepole Pine pinus contorta 15 30 70 NA  
## 6 19670 2 Alder sp. alnus sp T NA NA NA

tail(veg, n = 15)

## # A tibble: 15 x 8  
## fid plot.id species.name latin pc\_cover gt\_3m eq\_1to3 lt\_1m  
## <chr> <dbl> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 14539 3 Bluejoint grass calamagrostis can~ 30 NA 50 50  
## 2 14539 3 Willow sp. salix sp 15 NA NA NA  
## 3 14539 3 Twinberry lonicera involucr~ 5 NA NA 100  
## 4 19670 1 Beaked Sedge carex rostrata 20 NA NA NA  
## 5 19670 1 Ci. Interior? <NA> 60 NA NA NA  
## 6 19670 1 Marsh Cinquefo~ comarum palustre T NA NA NA  
## 7 19670 1 Scrub Birch betula nana 5 NA NA 100  
## 8 19670 1 Bog Willow salix pedicellaris 5 NA NA 200  
## 9 19670 2 Labrador Tea <NA> 40 NA NA 100  
## 10 19670 2 Scrub Birch betula nana 20 NA 20 80  
## 11 19670 2 Sphagnum sp. spagnum sp 90 NA NA NA  
## 12 19670 2 Beaked Sedge ccarex rostrata 5 NA NA NA  
## 13 19670 2 Calamagrostis calamagrostis sp 4 NA NA NA  
## 14 19670 2 Lodgepole Pine pinus contorta 15 30 70 NA  
## 15 19670 2 Alder sp. alnus sp T NA NA NA

What about a few arbitrary rows just for sanity (or insanity depending on your view)?

## Tip: There are several ways to achieve this.

The solution here presents one form of using nested functions, i.e. a function passed as an argument to another function. This might sound like a new concept, but you are already using it! Remember my\_dataframe[rows, cols] will print to screen your data frame with the number of rows and columns you asked for (although you might have asked for a range or named columns for example). How would you get the last row if you don’t know how many rows your data frame has? R has a function for this. What about getting a (pseudorandom) sample? R also has a function for this.

veg[sample(nrow(veg), 5), ]

## # A tibble: 5 x 8  
## fid plot.id species.name latin pc\_cover gt\_3m eq\_1to3 lt\_1m  
## <chr> <dbl> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 88182 1 Water Sedge carex aquatilis 25 NA NA 100  
## 2 191640 2 Bog Cranberry vaccinium oxyco~ T NA NA NA  
## 3 109507b 1 Marsh Cinquefoil comarum palustre 10 NA NA NA  
## 4 109507b 1 Aspen sp. populus sp <NA> NA NA NA  
## 5 163604 1 Cabbage sp. <NA> 80 NA NA NA

### Solution to Challenge 4

veg[veg$plot.id == 4, ]

## # A tibble: 4 x 8  
## fid plot.id species.name latin pc\_cover gt\_3m eq\_1to3 lt\_1m  
## <chr> <dbl> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 109507 4 Spirea sp. sprea sp 80 NA NA NA  
## 2 109507 4 Marsh Cinquefoil comarum palus~ 5 NA NA NA  
## 3 109507 4 Skunk Cabbage symplocarpus ~ 15 NA NA NA  
## 4 109507 4 Water sedge (in ch~ carex aquatil~ 15 NA NA NA

### Solution to Challenge 5

veg[(veg$plot.id == 3), (c("fid", "plot.id", "pc\_cover"))]

## # A tibble: 21 x 3  
## fid plot.id pc\_cover  
## <chr> <dbl> <chr>   
## 1 111563 3 10   
## 2 111563 3 15   
## 3 111563 3 50   
## 4 111563 3 50   
## 5 111563 3 T   
## 6 111563 3 T   
## 7 111563 3 T   
## 8 111563 3 10   
## 9 111563 3 T   
## 10 111563 3 T   
## # ... with 11 more rows