

R Training 2

Rachel Kenny

4/01/2020

R code notes

- Capitalization matters in r code. Lowercase and uppercase are recognized as different characters
- To refer to a column in a dataframe, you can use a `[$]` symbol between the name of the dataframe and the name of the column

Loading packages & data

```
# Load packages
library(readxl)

## Warning: package 'readxl' was built under R version 3.4.4
library(tidyverse)

## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'zone/tz/2019c.1.0/
## zoneinfo/America/New_York'

## -- Attaching packages ----- tidyverse 1.3.0 --

## v ggplot2 3.3.0      v purrr  0.3.3
## v tibble  2.1.3      v dplyr  0.8.5
## v tidyr   1.0.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.5.0

## Warning: package 'readr' was built under R version 3.4.4
## Warning: package 'stringr' was built under R version 3.4.4

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
library(janitor)

## Warning: package 'janitor' was built under R version 3.4.4

# Agua chinon veg data
ac_data_raw <- read_excel("OCWR_AC_2019_Data.xlsx")
ac_data <- clean_names(ac_data_raw)

# Weir oak restoration data
oak_data_raw <- read_csv("Weir_Oak_Restoration_Data_winter19.csv")

## Parsed with column specification:
## cols(
```

```
## `Short ID` = col_character(),
## Survival = col_logical(),
## Quantity = col_double(),
## `Height (cm)` = col_double(),
## `Open Closed` = col_character(),
## `Location UML` = col_character(),
## `Water Yes No` = col_character(),
## `Sampling Group` = col_character()
## )
```

```
oak_data <- oak_data_raw %>%
  clean_names()
```

Data wrangling 1

```
# To refer to a column in a dataframe, you can use a [$] symbol
oak_data$height_cm
```

```
##      [1] 16.5  8.0 24.5  0.0  0.0  0.0  0.0  0.0 13.0 13.2  0.0  0.0
##     [13]  0.0  9.5  0.0 15.0  0.0  7.0  0.0  9.0 21.6 20.5 18.5  0.0
##     [25] 15.0 11.5 16.5  0.0  0.0  0.0 18.5 24.2  0.0  0.0  0.0  0.0
##     [37]  0.0 20.1  0.0 14.0 14.0  0.0  0.0 26.5  0.0 22.1  0.0  0.0
##     [49]  6.5  8.1  0.0  6.5 11.0 13.0 12.0 17.0  0.0  1.0  7.5  0.0
##     [61] 13.5  9.8 16.7  0.0 19.0  0.0 14.0 16.0  0.0 11.0 15.0  0.0
##     [73] 22.4 17.5 16.5 19.3 17.0 17.2  0.0 18.0 13.0 18.0 18.0  0.0
##     [85] 25.5  0.0  0.0 11.0 11.0  0.0 15.0  0.0  0.0 17.0 16.0 23.0
##     [97]  0.0  0.0 16.0  0.0  0.0 21.0  0.0 21.0  9.0 22.5  0.0 12.5
##    [109] 25.0 13.0  0.0 20.0 14.0 17.0 18.0  6.0 18.0 18.0 13.0 23.0
##    [121] 24.0  0.0  9.0  9.0 11.0  0.0 10.0 17.0 14.0  0.0  0.0  0.0
##    [133]  0.0 14.0 17.0  0.0  0.0  0.0 15.0  0.0  0.0  0.0 12.0 14.0
##    [145]  0.0  0.0 16.0  0.0 13.0  0.0 11.0 11.0 21.0  0.0 24.0 10.0
##    [157]  9.0 13.0  0.0 11.5  0.0  0.0  0.0  0.0  0.0  1.7 14.0  0.0
##    [169] 16.0 14.0  0.0 15.0 10.0 13.0  0.0 15.0  0.0  0.0  0.0  0.0
##    [181]  0.0  0.0  0.0  0.0  0.0  0.0  0.0 13.0  0.0  0.0  0.0  0.0
##    [193]  8.0 11.0  0.0 12.0 15.0 11.0 11.0 10.0  0.0 23.0 22.0  0.0
##    [205] 11.0  0.0  0.0  0.0  9.5  0.0 24.0 24.0 11.0  0.0 17.0  0.0
##    [217] 17.0 22.0  0.0  0.0  9.5 21.0 18.0 16.0 12.0 19.0  0.0  0.0
##    [229] 23.0  0.0  6.0 19.0  0.0 23.2 16.1 18.5  3.1  0.0  0.0  0.0
##    [241] 10.5 25.5 29.0 17.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
##    [253] 16.2  0.0  0.0  0.0  0.0  0.0 17.1 14.0 19.0 12.3 12.0 16.0
##    [265]  0.0  0.0  0.0  0.0 12.5  0.0  0.0  0.0  0.0  0.0  0.0  0.0
##    [277]  0.0 12.0  0.0 19.6  0.0  0.0  0.0  0.0 18.0  0.0  0.0  0.0
##    [289]  0.0 22.0 21.5  0.0  0.0 15.0  0.0  0.0 14.0  0.0 11.5  0.0
##    [301]  0.0  0.0  0.0  0.0  0.0  0.0 21.5  0.0  0.0  0.0  0.0  0.0
##    [313]  0.0 14.5  0.0 12.0 21.5 18.5  0.0  0.0  0.0  0.0  0.0 22.0
##    [325]  0.0  0.0  0.0  0.0  0.0  0.0 27.0 20.5 19.0 26.0  0.0 25.0
##    [337]  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
##    [349]  0.0  0.0 14.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  6.0  0.0
##    [361]  0.0  0.0  0.0  0.0 10.0  0.0  0.0  0.0  0.0  0.0  0.0 13.0
##    [373]  9.0  7.5 12.0  0.0 13.0  6.5  1.5 15.0 16.0 11.0  9.9  0.0
##    [385] 13.2 12.0 10.0 12.3  0.0  0.0  0.0  0.0  0.0  0.0  0.0  5.0
##    [397] 15.0 16.0 14.0  8.0  0.0  0.0  0.0  0.0  0.0  0.0 19.0 24.0
##   [409] 15.0 11.0  0.0  5.0  0.0 14.0  7.0 16.0  0.0 15.0 13.0  0.0
```

##	[421]	10.0	16.0	16.0	11.0	0.0	8.0	15.0	18.0	12.0	0.0	12.0	16.0
##	[433]	12.0	0.0	14.0	11.0	13.0	0.0	0.0	13.0	10.0	0.0	15.0	0.0
##	[445]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	13.0	16.0	27.0
##	[457]	16.0	7.0	0.0	11.0	8.0	11.0	20.0	16.0	0.0	0.0	0.0	20.0
##	[469]	0.0	12.0	15.0	0.0	18.0	18.0	10.0	12.0	18.0	16.0	16.0	0.0
##	[481]	10.0	0.0	0.0	14.0	0.0	13.0	11.0	0.0	12.0	21.0	0.0	0.0
##	[493]	0.0	10.0	0.0	8.0	15.0	17.0	15.0	0.0	12.0	21.0	0.0	16.0
##	[505]	15.0	16.0	15.0	22.0	13.0	0.0	12.0	17.0	0.0	0.0	0.0	24.0
##	[517]	0.0	15.0	0.0	15.0	14.0	9.0	19.0	0.0	15.0	11.0	6.0	0.0
##	[529]	6.0	0.0	0.0	0.0	13.0	16.0	15.0	7.0	10.0	0.0	20.0	20.0
##	[541]	13.0	18.5	27.0	11.0	23.0	0.0	0.0	13.0	0.0	17.0	0.0	24.0
##	[553]	0.0	5.5	0.0	25.0	9.0	21.0	22.0	10.0	17.0	10.5	15.0	13.5
##	[565]	13.0	15.5	0.0	17.0	0.0	10.5	0.0	15.0	27.0	15.0	14.0	0.0
##	[577]	8.0	10.0	6.0	8.0	23.0	0.0	0.0	17.0	0.0	13.0	16.5	10.0
##	[589]	0.0	0.0	15.0	10.0	0.0	0.0	21.0	16.0	8.0	0.0	0.0	0.0
##	[601]	0.0	0.0	0.0	0.0	15.0	12.0	17.0	24.5	22.0	0.0	17.0	22.0
##	[613]	18.0	0.0	23.5	0.0	0.0	12.0	0.0	0.0	0.0	0.0	25.5	27.0
##	[625]	0.0	0.0	0.0	13.0	11.0	11.5	0.0	14.5	21.0	24.5	20.5	46.5
##	[637]	19.4	23.8	29.0	12.3	18.0	17.5	19.5	0.0	0.0	14.3	25.0	0.0
##	[649]	31.7	21.5	22.3	25.6	16.3	19.0	24.5	0.0	0.0	0.0	0.0	0.0
##	[661]	0.0	0.0	11.7	0.0	14.1	0.0	13.0	0.0	8.0	0.0	20.9	10.6
##	[673]	18.9	13.3	17.7	14.9	24.9	10.7	0.0	15.0	14.0	12.5	0.0	7.5
##	[685]	10.5	9.0	23.0	12.0	0.0	14.5	0.0	15.0	7.0	0.0	18.0	0.0
##	[697]	20.0	17.0	11.0	6.0	28.0	0.0	19.5	12.0	16.5	13.5	14.5	0.0
##	[709]	6.5	22.0	0.0	4.5	0.0	20.5	0.0	9.0	0.0	22.5	0.0	22.0
##	[721]	0.0	9.0	46.0	35.0	22.0	16.0	16.0	18.0	26.0	11.0	24.0	0.0
##	[733]	0.0	40.0	0.0	0.0	22.0	15.0	17.0	0.0	0.0	12.0	9.0	0.0
##	[745]	10.0	37.0	15.0	0.0	0.0	40.0	13.0	0.0	8.0	0.0	11.0	0.0
##	[757]	17.0	22.0	17.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	14.0	19.0
##	[769]	10.0	21.0	0.0	0.0	0.0	24.5	26.0	8.0	12.0	18.6	0.0	22.0
##	[781]	21.0	13.0	19.0	10.0	0.0	0.0	0.0	0.0	0.0	14.5	9.8	0.0
##	[793]	0.0	9.0	19.0	17.5	21.0	0.0	17.8	18.0	10.0	18.0	0.0	10.0
##	[805]	21.0	9.0	17.0	11.0	0.0	0.0	18.0	12.0	0.0	0.0	0.0	0.0
##	[817]	19.0	0.0	NA	17.5	28.0	17.0	19.5	15.0	12.0	15.0	0.0	15.0
##	[829]	10.0	11.0	0.0	14.0	0.0	7.0	0.0	0.0	0.0	12.0	13.0	18.0
##	[841]	10.0	15.0	0.0	0.0	17.0	21.0	0.0	0.0	11.0	17.0	0.0	14.0
##	[853]	17.0	15.0	18.0	15.0	11.0	0.0	10.5	11.0	18.5	18.0	19.0	26.0
##	[865]	27.0	14.0	13.0	16.0	0.0	5.0	20.0	0.0	0.0	17.5	0.0	12.0
##	[877]	22.5	20.0	17.0	16.3	0.0	0.0	8.7	6.0	17.5	16.8	9.5	17.5
##	[889]	19.0	0.0	18.4	31.6	14.0	0.0	15.3	14.5	14.2	13.9	10.9	0.0
##	[901]	14.5	12.7	14.5	22.0	19.5	0.0	12.5	12.0	9.9	22.3	8.1	10.5
##	[913]	19.0	14.1	21.5	30.5	0.0	0.0	20.0	20.3	11.6	0.0	11.5	6.0
##	[925]	8.0	0.0	0.0	0.0	0.0	0.0	14.5	10.0	834.0	11.4	0.0	0.0
##	[937]	0.0	0.0	12.0	16.3	11.2	0.0	11.5	16.0	10.5	10.0	11.0	6.0
##	[949]	18.0	0.0	0.0	0.0	0.0	16.5	10.0	14.0	8.5	13.0	0.0	19.0
##	[961]	9.5	9.0	0.0	0.0	9.0	22.0	13.0	0.0	0.0	0.0	6.5	9.0
##	[973]	0.0	10.0	0.0	0.0	14.0	12.0	0.0	18.0	15.0	15.5	0.0	21.0
##	[985]	18.0	14.0	0.0	13.0	17.0	24.0	13.0	14.0	12.0	23.0	15.0	17.0
##	[997]	13.0	17.0	20.0	9.0	0.0	0.0	16.0	24.0	13.0	17.0	21.0	12.0
##	[1009]	23.0	14.0	15.0	10.0	0.0	10.0	0.0	3.0	0.0	0.0	0.0	5.0
##	[1021]	16.0	5.0	11.0	16.0	0.0	0.0	10.0	0.0	15.0	0.0	15.0	0.0
##	[1033]	0.0	9.0	0.0	0.0	0.0	22.0	0.0	0.0	0.0	6.0	7.0	0.0
##	[1045]	6.0	0.0	0.0	7.0	9.0	0.0	0.0	13.0	8.0	12.0	15.0	4.0
##	[1057]	24.0	15.0	14.0	14.0	0.0	11.0	0.0	0.0	15.0	0.0	0.0	15.0

```
## [1069]  0.0  15.0   0.0  22.0   0.0   0.0  14.0  12.0   0.0   0.0  10.0   0.0
## [1081]  0.0   0.0   0.0   0.0   4.0   0.0   0.0   0.0   0.0  13.0   0.0  11.0
## [1093]  0.0  11.0  12.0   0.0  16.0  13.0  14.0  14.0  10.0  10.0   0.0   0.0
## [1105]  0.0   0.0   0.0   0.0   8.0   0.0   0.0  18.0   0.0   0.0  13.0   0.0
## [1117]  0.0  21.0  18.0   0.0   0.0   0.0  19.0   0.0  22.0   0.0  16.0   7.0
## [1129]  0.0   0.0  14.0   9.0   0.0  16.0   0.0  14.0   0.0  19.0   0.0   0.0
## [1141]  11.0  12.0  10.0   5.0   0.0   0.0   0.0   0.0   0.0   0.0   6.0   0.0
## [1153]  0.0  16.0  14.0   0.0   0.0  11.0  18.0   0.0  10.0  10.0  10.0   0.0
## [1165]  0.0  12.0  11.0  17.0   0.0  26.0  20.0  16.0  21.0   0.0   8.0  10.0
## [1177]  0.0   0.0   0.0   0.0  15.0   0.0   0.0  17.0   0.0  14.0   0.0   0.0
## [1189]  0.0  18.0  63.0  16.0   9.0  19.0  20.0   0.0   6.0  25.0   6.0  15.0
## [1201]  9.0   0.0   0.0   0.0
```

```
View(oak_data$height_cm)
```

```
# Another way to refer to a column is by referring to the number of the column using square brackets []
oak_data[4]
```

```
## # A tibble: 1,204 x 1
##   height_cm
##   <dbl>
## 1    16.5
## 2     8
## 3    24.5
## 4     0
## 5     0
## 6     0
## 7     0
## 8     0
## 9     13
## 10    13.2
## # ... with 1,194 more rows
```

```
oak_data[4:6]
```

```
## # A tibble: 1,204 x 3
##   height_cm open_closed location_uml
##   <dbl> <chr>         <chr>
## 1    16.5 Closed      Lower
## 2     8 Closed      Lower
## 3    24.5 Closed      Lower
## 4     0 Closed      Lower
## 5     0 Closed      Lower
## 6     0 Closed      Lower
## 7     0 Closed      Lower
## 8     0 Closed      Lower
## 9     13 Closed      Lower
## 10    13.2 Closed      Lower
## # ... with 1,194 more rows
```

```
oak_data[-4]
```

```
## # A tibble: 1,204 x 7
##   short_id survival quantity open_closed location_uml water_yes_no
##   <chr>    <lgl>      <dbl> <chr>      <chr>      <chr>
## 1 L_C_01_1 TRUE          2 Closed      Lower      No
```

```
## 2 L_C_01_2 TRUE          1 Closed      Lower      Yes
## 3 L_C_01_3 TRUE          1 Closed      Lower      No
## 4 L_C_01_4 FALSE         0 Closed      Lower      No
## 5 L_C_02_1 FALSE         0 Closed      Lower      Yes
## 6 L_C_02_2 FALSE         0 Closed      Lower      Yes
## 7 L_C_02_3 FALSE         0 Closed      Lower      Yes
## 8 L_C_02_4 FALSE         0 Closed      Lower      No
## 9 L_C_03_1 TRUE          1 Closed      Lower      Yes
## 10 L_C_03_2 TRUE         1 Closed      Lower      Yes
## # ... with 1,194 more rows, and 1 more variable: sampling_group <chr>
```

Use the [which] code to refer to specific observations in the code. In this example, we are replacing
`ac_data$polygon_id[which(ac_data$polygon_id=="5M Buffer")] <- "5M BUFFER"`

There are a few ways to rename a column

Method 1: use function [names]

```
ac_data2 <- ac_data
```

```
names(ac_data2)[1] <- paste("habitat") # name column 1 "habitat"
```

Method 2: use function [colnames]

```
ac_data2 <- ac_data
```

```
colnames(ac_data2)[4:6] <- c("pin", "type_data", "code_species") # rename columns 4 through 6
```

Data wrangling 2

There are a few ways to create a new column

Method 1: Use the function [mutate]. Within the new column you can make transformations on your data.

```
oak_data2 <- oak_data %>%
```

```
  mutate(new_column=height_cm*2)
```

Method 2: Create a new column using the [\$] symbol

```
oak_data3 <- oak_data
```

```
oak_data3$height_rounded <- round(oak_data3$height_cm)
```

```
View(oak_data2)
```

```
View(oak_data3)
```

Creating tables

To count observations by group, use the [group_by] and [count] functions. Here are two examples below

agua chinon data example - number of observations per species code in each polygon and transect

```
richness1 <- ac_data %>%
```

```
  select(transect, polygon_id, species_code) %>%
```

```
  group_by(polygon_id, transect) %>%
```

```
  count(species_code)
```

```
View(richness1)
```

oak data example - # of plantings watered or not watered by sampling group

```
oak_water_sampling_grp <- oak_data %>%
```

```
select(sampling_group, water_yes_no) %>%  
group_by(sampling_group) %>%  
count(water_yes_no)  
  
View(oak_water_sampling_grp)
```