- To check current working directory
    - getwd()
- To change current working directory
    - setwd()
- Assigning new objects to the environment
    - Object_name <-
    - After you assign, state the object name again underneath for it to show up
- round(variable, number of digits to be round to)
- c() is combine and create a vector
    - mean() for avg
- class()
    - To figure out what type of data
- is()
    - is.integer(my_numeric)
    - Asking if the “my_numeric is an integer
    - my_interger <- as.integer(my_numeric)
        - Converting numeric to integer
    - is.na()
        - Is this value missing
    - any.na()
        - Any value missing
- na.rm=TRUE
    - Removes NA values
- Create vector
    - My_vector <- c(chr or numeric)
    - [] pulls out an element from a vector
- Data frames
    - tibble() creates new data frame
    - names() gives variable names in column
    - For dimensions of data frame
        - dim()
        - str()
        - [#,#] -> row, column
    - $
        - Access a column (variable) in a data frame
        - mean(data$column) calculates mean
    - Writing data to file
        - write.csv(data, “data.csv”, row.names = FALSE)
- Read.csv reads data into new name/variable
- as.factor(data$columns)

    - Changes type for the column in the data
    - levels() -> levels for column
- Summary
    - glimpse()
    - summary()
    - Nrow, ncol, dim, names
    - head() gives first n rows of data frame
    - tail() gives last n rows
- select()
    - Used to extract columns
    - Range of columns -> start_col:end_col
    - !
        - Select everything except specified variable
        - !c(), choose multiple columns to exclude
- rename(data, new_name = old_name)
- filter()
    - Extract rows
    - filter()` allows all of the expected operators; i.e. >, >=, <, <=, != (not equal), and == (equal).
        - filter(penguins, **species!=**"Chinstrap")
    - %in%
        - Determines whether or not a value is part of a vector
    - between()
        - Range of specific values
    - How would you use `%in%` to get the same result?
        - filter(penguins, body_mass_g %in% c(5250, 5300, 5350, 5400, 5450, 5500))
    - Filter the fish data to include all fish with a scalelength within 0.25 of 8.
        - filter(fishes, near(scalelength, 8, tol=0.25))
    - | is or either condition occurs.
    - + `filter(condition1, condition2)` will return rows where both conditions are met. By default the , means &.
    - + `filter(condition1, !condition2)` will return all rows where condition one is true but condition 2 is not.
    - + `filter(condition1 | condition2)` will return rows where condition 1 or condition 2 is met.
    - . We are interested in the fish from the lakes "AL" and "AR" with a radii length between 2 and 4. Extract this information from the `fish` data. Please sort the data in descending order by radii length.
        - fishes %>%
            - select(lakeid, radii_length_mm) %>%

        - filter(lakeid=="AL"| lakeid=="AR") %>%
        - filter(between(radii_length_mm, 2, 4)) %>%
        - arrange(desc(radii_length_mm))
- homerange$mean.hra.m2 <- as.numeric(homerange$mean.hra.m2)
    - filter(homerange, class == "mammalia", trophic.guild == "herbivore", mean.hra.m2>1000000) %>%
    - select(common.name, order, mean.hra.m2, mean.mass.g) %>%
    - arrange(desc(mean.hra.m2)) %>%
    - slice_max(mean.hra.m2, n=10)
- distinct()
    - Acts on rows
- mutate()
    - Acts on columns
        - Create new columns from existing ones
    - penguins %>%
        - mutate(body_mass_kg=body_mass_g/1000) %>%
        - select(species, body_mass_g, body_mass_kg) %>%
        - arrange(body_mass_kg)
    - Use across() to apply to multiple columns
        - So, `./10` means "take the current column and divide it by 10". This operation is applied to all columns that end with `mm`.
    - mammals %>%
        - mutate(across(where(is.character), tolower)) #only variables that are class character to lower case

    ~ifelse(.==-999, NA,.)

        Replaces -999 with NA