Functions Learned So Far

Ronnie Steinitz

2025-09-25

Week 1: Is My Data Clean? Exploring, Diagnosing, and Visualizing Problems

0. Load Required Packages

- `library()` # Loads an installed package into your R session so its functions can be used.
  Example: `library(tidyverse)` → loads the tidyverse collection of packages.
- `janitor::clean_names()` # Cleans column names (lowercase, underscores instead of spaces/symbols).
  Example: `data <- janitor::clean_names(data_raw)` → turns "Flipper Length (mm)" into "flipper_length_mm".

1. Load and Preview Dataset

- `getwd()` # Shows the current working directory (the folder R is looking in by default).
  Example: `getwd()` → might return "/Users/rsteinitz/Documents/github/R Data Analysis Course".
- `setwd()` # Sets the working directory (where R should look for or save files).
  Example: `setwd("/Users/rsteinitz/Documents/github/R Data Analysis Course")`.
- `read_csv()` # Reads a .csv file into R as a data frame (from the readr package).
  Example: `data_raw <- read_csv("Week 1/Palmer Penguins Raw.csv")`.
- `glimpse()` # Provides a compact overview of a dataset (rows, columns, and types).
  Example: `glimpse(data)` → shows columns, data types, and sample values.
- `str()` # Displays the structure of an object.
  Example: `str(data)` → tells you number of rows, columns, and types.
- `head()` # Prints the first 6 rows of a dataset.
  Example: `head(data)` → shows the top rows of the penguins dataset.
- `names()` # Lists the column names in a dataset.
  Example: `names(data)` → returns column headers like "species", "island", "sex".
- `View()` # Opens the dataset in a spreadsheet-like viewer (interactive).
  Example: `View(data)` → opens a new tab in RStudio with your dataset.

2. Diagnosing Data Types and Structure

- `class()` # Shows the data type (numeric, character, factor, etc.) of an object.
  Example: `class(data$sex)` → returns "character".
- `table()` # Summarizes counts of unique values in a variable.
  Example: `table(data$species)` → counts how many penguins belong to each species.
- `unique()` # Lists unique values in a variable.
  Example: `unique(data$island)` → shows "Biscoe", "Dream", "Torgersen".
- `length()` # Tells how many elements are in a vector.
  Example: `length(unique(data$flipper_length_mm))` → number of distinct flipper lengths.
- `count()` # Counts rows by categories of a variable (from dplyr).
  Example: `count(data, island)` → counts penguins per island.

3. Missing Data: Detection and Summary

- `is.na()` # Tests whether values are missing (returns TRUE/FALSE).
  Example: `is.na(data$sex)` → shows TRUE for rows missing sex info.
- `colSums()` # Adds up values across each column. Often used with is.na().
  Example: `colSums(is.na(data))` → number of NAs in each column.
- `sum()` # Adds up all numeric values, or counts TRUE values in logical vectors.
  Example 1: `sum(is.na(data$flipper_length_mm))` → number of missing flipper lengths.
  Example 2: `sum(data$flipper_length_mm > 200, na.rm = TRUE)` → number of penguins with long flippers.
- `summary()` # Gives descriptive statistics (mean, median, min, max).
  Example: `summary(data$bill_depth_mm)` → outputs min, max, mean, etc.
- `range(..., na.rm = TRUE)` # Shows the minimum and maximum values.
  Example: `range(data$bill_length_mm, na.rm = TRUE)` → min and max bill length.

4. Basic Visualizations

- `hist()` # Creates a histogram of a numeric variable (base R).
  Example: `hist(data$flipper_length_mm)`.
- `ggplot()` # Starts a ggplot graph.
  Example: `ggplot(data, aes(x = flipper_length_mm)) + geom_histogram()`.

All `ggplot()` plots must have three basic components: `data`, `aes`, and a `geom`

- `aes()` # Maps variables to visual properties.
  Example: `aes(x = species, fill = sex)`.
- `geom_histogram()` # Adds a histogram layer in ggplot.
  Example: `geom_histogram(binwidth = 2, fill = "steelblue")`.
- `facet_wrap()` # Splits one plot into multiple panels by a grouping variable.
  Example: `facet_wrap(~ species)` → separate histograms per species.
- `geom_bar()` # Creates a bar chart for categorical variables.
  Example: `geom_bar()` → counts penguins per species.

By the end of Week 1, you should be comfortable with:

- Importing and previewing data (`read_csv()`, `glimpse()`, `head()`, `names()`).
- Checking and diagnosing data types (`class()`, `unique()`, `table()`).
- Detecting and summarizing missing data (`is.na()`, `colSums()`, `summary()`).
- Converting variables to correct types (`mutate()`, `as.factor()`).
- Making basic plots (`hist()`, `ggplot()`, `geom_bar()`, `geom_histogram()`).

Week 2: Wrangling Basics – Select, Filter, Mutate

1. Pipe Operator

- `%>%` (pipe operator) # Sends the output of one function as the input to the next.
  Example: `data_raw %>% clean_names() %>% glimpse()`.

2. Select Columns

- `select()` # Keeps or drops specific columns.
  Example 1: `data %>% select(species, island)` → keep these columns.
  Example 2: `data %>% select(-comments)` → drop the comments column.

3. Filter Rows

- `filter()` # Keeps rows meeting conditions.
  Example 1: `filter(data, species == "Adelie")`.
  Example 2: `filter(data, flipper_length_mm > 200)` → penguins with long flippers.
  Example 3: `ggplot(data %>% filter(flipper_length_mm > 200)) + geom_bar()`.
- Logical operators: `==` equal, `!=` not equal, `>`, `<`, `&` (and), `|` (or).
  Example: `filter(data, species == "Adelie" & island == "Dream")`.

4. Mutate / Create New Variables

- `mutate()` # Adds or transforms columns in a dataset.
  Example: `data <- mutate(data, body_mass_kg = body_mass_g / 1000)`.
- `as.factor()` # Converts a variable into a factor (categorical variable).
  Example: `data$sex <- as.factor(data$sex)`.
- `as.numeric()`, `as.integer()`, `as.logical()`, `as.character()`, `as.Date()` # Convert variables between data types.
  Example: `as.Date(data$date_egg, format = "%m/%d/%y")`.
- `case_when()` # Recode values or create categories.
  Example: `data %>% mutate(size_class = case_when(flipper_length_mm < 185 ~ "Small", flipper_length_mm >= 200 ~ "Large"))`.
- `ifelse()` # Conditional operation: if [condition], then do [action], otherwise do [different action].
  Example: `data %>% mutate(size_class = ifelse(flipper_length_mm > 200, "Large", "Small"))`.
- `word()` # Extracts words from a text string.
  Example: `data %>% mutate(species_simple = word(species, 1))` → "Adelie", "Gentoo", "Chinstrap".

5. Visualization

- `labs()` # Adds or edits labels for titles, axes, and legends.
  Example: `labs(title = "Penguin Counts", x = "Species", y = "Number of Penguins")`.
- `scale_fill_manual()` # Manually sets fill colors.
  Example: `scale_fill_manual(values = c("male" = "blue", "female" = "red"))`.

By the end of Week 2, you should be comfortable with:

- Using `%>%` pipes to link commands together and write clean, readable code.
- Selecting specific columns with `select()`.
- Filtering rows with conditions using `filter()`.
- Creating new variables with `mutate()` and `case_when()`.
- Visualizing subsets of data using `ggplot()` with filters.