R Data Types and Functions Demo

Scott Schwartz

Week 2 Demo

The purpose of this demo is to introduce R data types and functions:

- ☐ ***Variables*** like `my_vec_var` and ***assignments*** with `<-` (and `->` and `=`)
- ☐ ***Vectors*** and the `c()` ***concatenate/combine*** function: `c(1,"2")`
- ☐ ***Data types*** (`1`,`1.1`, `TRUE`, `"1"`) and `class()` and `typeof()` functions
- ☐ ***Coercion*** and `as.numeric()`, `as.logical()`, `character()` functions
- ☐ ***Ordinal*** with `as.integer()` versus ***nominal*** with `as.factor()`
- ☐ ***Tibbles*** `read_csv()` versus ***Data Frames*** `read.csv()`
- ☐ **Columns** are **variables** (but not "R variables"!): specific things to be observed and recorded
- ☐ **Rows** are **observational units** (on which variables are observed and recorded)
- ☐ A **Row** is the **observation**
  - `The value of a specific variable for a specific row is one piece of the collection of information on an entire row which comprise an **observation**`
- ☐ Beyond `glimpse()` and `head()` with `names()`, `[]`, `[[]]`, and `$` "functions"
- ☐ The `mean`, `median`, `mode`, `range`, `IQR`, `var`, and `sd` functions (and `%>%`)

1. [] ***Boolean Vector Mask*** selection

This demo is paired with a quercus practice quiz and a homework assignment

Q1q7 types Q1q8 coercion Q1q9 head/glimpse for number of observations Q1q10 speaking "%>%"

Q2 lines up very well with the lecture/demo content

[ ] create pollev lect 2 companion [ ] finalize week 2 material usage/harmonization [ ] finalize github links for week 1+2

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

```
coffee_ratings <- read_csv("coffee_ratings.csv")
```

```
## Rows: 1338 Columns: 36
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (18): species, owner, country_of_origin, farm_name, mill, company, altit...
## dbl (18): total_cup_points, aroma, flavor, aftertaste, acidity, body, balanc...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Extra Material

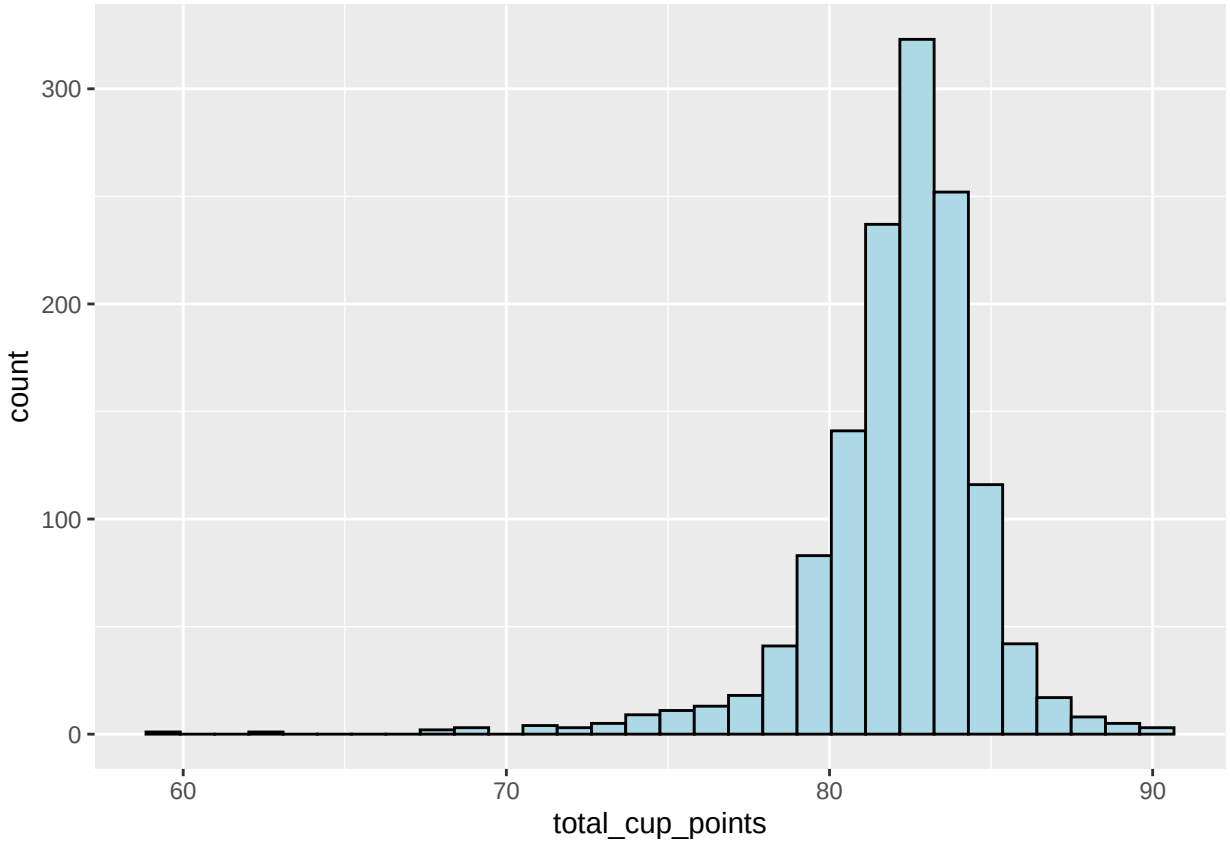
Topic 1: *ggplot2 syntax review*

(a) What are the first two lines of this code doing?

```
library(tidyverse)
coffee_ratings <- read_csv("coffee_ratings.csv")
```

```
## Rows: 1338 Columns: 36
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (18): species, owner, country_of_origin, farm_name, mill, company, altit...
## dbl (18): total_cup_points, aroma, flavor, aftertaste, acidity, body, balanc...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
coffee_ratings %>%
  ggplot(aes(x=total_cup_points)) +
  geom_histogram(bins=30, color="black", fill="light blue")
```



(b) Can you read the finel three lines of the code above? What do you roughly think it's doing?

(c) What parts of the code are parameters and what are parts are arguments?

(d) What does the `%>%` seem to be doing in the code above?

(e) What does the `+` seem to be doing in the code above?

Topic 2: a little more about *tibbles* and *types*

(a) Without variable assignments, use the `%>%` "pipe" command to find the `class()` of `tibble::tibble(col=c(1,2,3))`

(b) Without variable assignments, use the `%>%` "pipe" command to find out the `class()` of `read_csv()` and `read.csv()`.

Topic 3: a little more about *factors*

(a) Confirm the observed column data types using the following stackoverflow conversation.

- https://stackoverflow.com/questions/21125222/determine-the-data-types-of-a-data-frames-columns

(b) Change column data types in the `coffee_ratings` data and confirm the change using the code provided below based on the following stackoverflow conversation.

- https://stackoverflow.com/questions/27668266/dplyr-change-many-data-types

Topic 4: A little more about *factors* and *boolean vector masks*

(a) Create the variable `coffee_ratings_nlevels` which lists the number of factor levels of each column using the provided code based on the following stackoverflow conversation and `nlevels` documentation.

- https://stackoverflow.com/questions/27676404/list-all-factor-levels-of-a-data-frame
- https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/nlevels

(b) Create a boolean mask vector from the `coffee_ratings_nlevels` variable above which is `TRUE` for each column with two or more but less than ten factor levels and `FALSE` otherwise using the provided code below.

(c) List the factor levels of each column with two or more factor levels using a vector boolean mask code provided below.