18.2 Lesson Summary - Tibbles R Terrific

Tibble objects in R offer extensive functionality to import, process, and analyze tabular data in a manner similar to DataFrames in Python's Pandas library.

Concept: To **read a csv** into a **tibble** you can use the ***read\_csv*** function from the tidyverse R package. For example:

*library(tidyverse)*

*my\_tibble <- read\_csv("my.csv")*

To get the **first few rows** of a **tibble** you can use the head function. To use the head function with pipe syntax you could use the following code:

*my\_tibble %>% head()*

To access a **tibble's column** put the column name after the tibble variable followed by '$'. For example:

*my\_tibble$my\_column*

To get the number of unique values in a tibble's column you can use the following code:

*length(my\_tibble$my\_column %>% unique())*

* Activity: 01-Stu\_UFO

Concept: **Tibble's** can be **created** by associating vectors with column names. For example:

*my\_tibble <-*

*tibble(*

*Names = c(*

*"Sam",*

*"Franklin",*

*"Susan"*

*),*

*Numbers = c(*

*"Sam",*

*"Franklin",*

*"Susan"*

*)*

*)*

* Activity: 02-Ins\_Tibbles, 03-Stu\_Tibbles\_and\_Tibbles

Concept: When you want to apply a function to each element of a vector you can use the ***sapply*** function. For example:

*my\_new\_vector <- sapply(my\_vector, myFunction)*

If you want a list after applying your function to a vector you could use the ***lapply*** For example:

*my\_list <- lapply(my\_vector, myFunction)*

* Activity: 04-Ins\_Apply

Concept: To **join** two tibbles together you can use the **inner\_join** or other join functions from the tidyverse. For example:

*my\_new\_tibble = inner\_join(my\_first\_tibble, my\_second\_tibble, by=c("join\_on\_column"))*

* Activity: 05-School\_Tibble
* Suppl link: <https://dplyr.tidyverse.org/reference/join.html>