

Lesson 2: Base R vs Tidy R - Homework Answers

Instructor: Emily Markowitz (Emily.Markowitz@noaa.gov)

January 21, 2021

Answers to Questions:

1. Let's explore Tidyverse!

a. Think of Tidyverse as a family of packages. Which packages are loaded with {tidyverse}? What do each of these packages do? Check out: <https://tidyverse.tidyverse.org/>

```
# You can find out by looking at the packages that are loaded when you use  
library(tidyverse)  
  
# Or, here is a screenshot from the https://tidyverse.tidyverse.org/#usage website:  
webshot::webshot("https://tidyverse.tidyverse.org/", "tidyverse_usage.png",  
  cliprect = c(500, 5, 600, 505))
```

Usage

`library(tidyverse)` will load the core tidyverse packages:

- `ggplot2`, for data visualisation.
- `dplyr`, for data manipulation.
- `tidyr`, for data tidying.
- `readr`, for data import.
- `purrr`, for functional programming.
- `tibble`, for tibbles, a modern re-imagining of data frames.
- `stringr`, for strings.
- `forcats`, for factors.

You also get a condensed summary of conflicts with other packages you have loaded

```
library(tidyverse)
#> — Attaching packages —————
#> ✓ ggplot2 3.2.1      ✓ purrr   0.3.3
#> ✓ tibble  2.1.3      ✓ dplyr   0.8.3
```

b. Which package is `pivot_wider` from?

```
?pivot_wider
# {tidyr}
```

c. Which package is `rename` from?

```
?rename
# {dplyr}
```

d. Can you use `{tidyverse}` without `{base}` R?:

```
# Nope! {Tidyverse} is built on {Base} R.
```

2. Let's play with some data!

You can view the dataset `CO2` in more detail using `View(CO2)` and learn about it using `?CO2`. `CO2` comes from the `{datasets}` package which should already be automatically loaded in your R.

Some info about the `CO2` dataset: “The `CO2` data frame has 84 rows and 5 columns of data from an experiment on the cold tolerance of the grass species *Echinochloa crus-galli*.”

```
C02<-data.frame(datasets::C02)
kable(head(C02))
```

Plant	Type	Treatment	conc	uptake
Qn1	Quebec	nonchilled	95	16.0
Qn1	Quebec	nonchilled	175	30.4
Qn1	Quebec	nonchilled	250	34.8
Qn1	Quebec	nonchilled	350	37.2
Qn1	Quebec	nonchilled	500	35.3
Qn1	Quebec	nonchilled	675	39.2

a. `rename()` the “conc” column to “Concentration mL/L” and “Treatment” column to “condition”. The new name for the conc column is not a great name (dare I say ‘tidy’ name?) so we’ll fix that in the next question. Assign your object here as a new object (name up to you!).

```
C02_a<-rename(C02,
               "Concentration mL/L" = "conc")
kable(head(C02_a))
```

Plant	Type	Treatment	Concentration mL/L	uptake
Qn1	Quebec	nonchilled	95	16.0
Qn1	Quebec	nonchilled	175	30.4
Qn1	Quebec	nonchilled	250	34.8
Qn1	Quebec	nonchilled	350	37.2
Qn1	Quebec	nonchilled	500	35.3
Qn1	Quebec	nonchilled	675	39.2

b. Use the `{janitor}` function `clean_names` on the new CO2 data you just created in 2a. What does it do? How did `{janitor}` change our “Concentration mL/L” column?

Again, assign your object here as a new object (name up to you!).

```
C02_b<-janitor::clean_names(C02_a)
kable(head(C02_b))
```

plant	type	treatment	concentration_m_l_l	uptake
Qn1	Quebec	nonchilled	95	16.0
Qn1	Quebec	nonchilled	175	30.4
Qn1	Quebec	nonchilled	250	34.8
Qn1	Quebec	nonchilled	350	37.2
Qn1	Quebec	nonchilled	500	35.3
Qn1	Quebec	nonchilled	675	39.2

```
# clean_names changed the "conc" column from "Concentration mL/L" to "concentration_m_l_l"
```

c. Use `pivot_wider` make columns of uptake (`values_from`) for each plant (`names_from`) in your new data set from question 2b.

This is not a 'tidy' way of looking at data, but is good practice! Assign your object here as a new object (name up to you!).

```
C02_c<-pivot_wider(data = C02_b, names_from = plant, values_from = uptake)
kable(head(C02_c))
```

type	treatment	concentration_m_l_l	Qn1	Qn2	Qn3	Qc1	Qc2	Qc3	Mn1	Mn2	Mn3	Mc1	Mc2
Quebec	nonchilled	95	16.0	13.6	16.2	NA	NA	NA	NA	NA	NA	NA	NA
Quebec	nonchilled	175	30.4	27.3	32.4	NA	NA	NA	NA	NA	NA	NA	NA
Quebec	nonchilled	250	34.8	37.1	40.3	NA	NA	NA	NA	NA	NA	NA	NA
Quebec	nonchilled	350	37.2	41.8	42.1	NA	NA	NA	NA	NA	NA	NA	NA
Quebec	nonchilled	500	35.3	40.6	42.9	NA	NA	NA	NA	NA	NA	NA	NA
Quebec	nonchilled	675	39.2	41.4	43.9	NA	NA	NA	NA	NA	NA	NA	NA

d. Use `pivot_longer` to undo what you did in 2c using the data that you created in 2c.

To see how to get the old names back, check out the `names_to` and `values_to` variable in `?pivot_longer`. This will likely incur some new rows with NAs, so you'll need to remove that here with `values_drop_na`. You can check if you actually got it back to original form by seeing if the dimensions of the data.frame are the same as the original dataset. As stated earlier, `dim(datasets::C02)` was 84 rows and 5 columns.

```
C02_d<-pivot_longer(data = C02_c,
  cols = c(Qn1:Mc3),
  names_to = "plant",
  values_to = "uptake",
  values_drop_na = TRUE)
kable(head(C02_d))
```

type	treatment	concentration_m_l_l	plant	uptake
Quebec	nonchilled	95	Qn1	16.0
Quebec	nonchilled	95	Qn2	13.6
Quebec	nonchilled	95	Qn3	16.2
Quebec	nonchilled	175	Qn1	30.4
Quebec	nonchilled	175	Qn2	27.3
Quebec	nonchilled	175	Qn3	32.4

```
dim(C02_d)
```

```
## [1] 84 5
```

```
#alternatively
C02_d<-pivot_longer(data = C02_c,
  cols = c(4:15),
  names_to = "plant",
  values_to = "uptake",
  values_drop_na = TRUE)
```