

Scenario 2

Install Packages (if necessary)

```
# install.packages("tidyverse") # will take some times since multiple packages in one
# install.packages("readxl")
# install.packages("janitor")
```

Calling libraries

```
library(tidyverse)
```

```
-- Attaching packages ----- tidyverse 1.3.2 --
v ggplot2 3.3.6      v purrr   0.3.4
v tibble  3.1.8      v dplyr  1.0.9
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()
```

```
library(readxl)
library(janitor)
```

Attaching package: 'janitor'

The following objects are masked from 'package:stats':

chisq.test, fisher.test

```
library(lubridate)
```

Attaching package: 'lubridate'

The following objects are masked from 'package:base':

```
date, intersect, setdiff, union
```

Bringing in the data (do not need to re-run if you are following after scenario 1)

```
## Getting all the sheet names from the excel workbook
sheets <- excel_sheets("Data/sample_animal_shelter_data.xlsx")

## Iterating the process above through map_df
## Getting the path with the sheet name and creating the new column all at once
after_times_complete_data <- purrr::map_df(sheets, ~dplyr::mutate(readxl::read_excel("Data
```

The Before Times

In this case you are writing out the text multiple times and if you need to change it you would have to make sure you change ever value. In this mini example that might be fairly simple to do, but when you have more things going on in your code, you might want a simpler way to go about changes.

```
version_a_data_edits <- after_times_complete_data |>
  janitor::clean_names() |> # adding this just so that column names are in snake case
  mutate(status = case_when(date_of_birth < ymd(20220715) ~ "Too soon",
                             date_of_birth > ymd(20220715) ~ "Too late",
                             date_of_birth == ymd(20220715) ~ "Winner"),
         extract_date = ymd(20220715))
```

The After Times

In this case we are adding an additional line and step, but that step saves you time in the long run. Here you are creating the value by prompting yourself to write it in.

```
# This pauses your code to input a value. The prompt is whatever text you want to pop up i
date_for_processing <- readline(prompt="Enter the winning date in ymd format with no dashes
```

Enter the winning date in ymd format with no dashes (e.g., 20220715):

```
# The same code as before, but not with the value you just created instead of the actual t
version_b_data_edits <- after_times_complete_data |>
  janitor::clean_names() |> # adding this just so that column names are in snake case
  mutate(status = case_when(date_of_birth < ymd(date_for_processing) ~ "Too soon",
                             date_of_birth > ymd(date_for_processing) ~ "Too late",
                             date_of_birth == ymd(date_for_processing) ~ "Winner"),
         extract_date = ymd(date_for_processing))
```