In this post, We’re simply going to see 5 tricks that could help improve your tooling using {tidyverse}.

**Create a difference variable between the current value and the next value**

This is also known as lead and lag – especially in a time series dataset this varaible becomes very important in feature engineering. In Excel, This is simply done by creating a new formula field and subtracting the next cell with the current cell or the current cell with the previous cell and dragging the cell formula to the last cell.

Let’s take our [{fakir} fake\_visits dataset](https://www.programmingwithr.com/how-to-generate-meaningful-fake-data-for-learning-experimentation-and-teaching/) and if we are trying to find a day when there was a huge peak/drop of home visits, we can identify only by first creating a column which must be the difference between the next value and the current value.

We can do with the {dplyr} function lag() and lead() for respective purposes.

library(tidyverse)

df <- fakir::fake\_visits()

df %>% # filter(year %in% '2017') %>%

mutate(day\_lag = lag(home, default = home[1]) - home) %>% head()

## # A tibble: 6 x 9

## timestamp year month day home about blog contact day\_lag

##

## 1 2017-01-01 2017 1 1 352 176 521 NA 0

## 2 2017-01-02 2017 1 2 203 115 492 89 149

## 3 2017-01-03 2017 1 3 103 59 549 NA 100

## 4 2017-01-04 2017 1 4 484 113 633 331 -381

## 5 2017-01-05 2017 1 5 438 138 423 227 46

## 6 2017-01-06 2017 1 6 NA 75 478 289 NA

**Combining Multiple Columns into one Column**

One of the things that we often do in Excel is combining multiple columns into one column by concatenating the cell values. Like in the above example, we can see three columns year, month, date but all of them combined together can give us a date-format date (assuming the timestamp varible isn’t present) and that’s where the function unite() comes handy.

df %>% select(-timestamp) %>% head() %>% unite("date-format", c("year", "month",

"day"), sep = "-")

## # A tibble: 6 x 5

## `date-format` home about blog contact

##

## 1 2017-1-1 352 176 521 NA

## 2 2017-1-2 203 115 492 89

## 3 2017-1-3 103 59 549 NA

## 4 2017-1-4 484 113 633 331

## 5 2017-1-5 438 138 423 227

## 6 2017-1-6 NA 75 478 289

**Splitting One Column into Multiple Columns**

This is the inverse of what we did above and another very frequently used excel feature *Text to Columns*.

In the fakir\_visits(), let’s assume we don’t have year,month and day separately and now we’ve got to create those three columns from timestamp. This is quite simple with separate() function.

df %>% select(-c("year", "month", "day")) %>% head() %>% separate(col = timestamp,

into = c("year", "month", "day"), sep = "-")

## # A tibble: 6 x 7

## year month day home about blog contact

##

## 1 2017 01 01 352 176 521 NA

## 2 2017 01 02 203 115 492 89

## 3 2017 01 03 103 59 549 NA

## 4 2017 01 04 484 113 633 331

## 5 2017 01 05 438 138 423 227

## 6 2017 01 06 NA 75 478 289

**Summary**

The idea of this post was to introduce those four functions:

* lead()
* lag()
* unite()
* separate()

and show case how super-useful they are for many commonly used Excel features in Data Analysis.