



Arpan Sapkota

2023-05-06

a

```
library(tidyverse)

## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.2    ✓ readr      2.1.4
## ✓ forcats    1.0.0    ✓ stringr    1.5.0
## ✓ ggplot2     3.4.2    ✓ tibble     3.2.1
## ✓ lubridate  1.9.2    ✓ tidyr      1.3.0
## ✓ purrr      1.0.1
## — Conflicts — tidyverse_conflicts() —
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

set.seed(07)
df <- tibble(country = sample(c("China", "Nepal", "india"), 10, replace = TRUE),
              year = sample(2000:2020, 10, replace = TRUE),
              cases = sample(100:500, 10, replace = TRUE),
              population = sample(100000:1000000, 10, replace = TRUE))

df

## # A tibble: 10 × 4
##   country year cases population
##   <chr>   <int> <int>      <int>
## 1 Nepal   2014    249   457623
## 2 india   2007    286   780391
## 3 india   2011    120   849520
## 4 india   2018    487   776301
## 5 Nepal   2019    494   350104
## 6 india   2007    457   294412
## 7 Nepal   2017    381   409335
## 8 Nepal   2014    314   665428
## 9 india   2005    485   453031
## 10 Nepal  2015    199   898702
```

b

```
df_long <- df %>% pivot_longer(cols = c("cases", "population"), names_to = "variable", values_to = "value")
df_long

## # A tibble: 20 × 4
##   country year variable value
##   <chr>   <int> <chr>      <int>
## 1 Nepal   2014 cases        249
## 2 Nepal   2014 population 457623
## 3 india   2007 cases        286
## 4 india   2007 population 780391
## 5 india   2011 cases        120
## 6 india   2011 population 849520
## 7 india   2018 cases        487
## 8 india   2018 population 776301
## 9 Nepal   2019 cases        494
## 10 Nepal  2019 population 350104
## 11 india   2007 cases        457
## 12 india   2007 population 294412
## 13 Nepal   2017 cases        381
## 14 Nepal   2017 population 409335
## 15 Nepal   2014 cases        314
## 16 Nepal   2014 population 665428
## 17 india   2005 cases        485
## 18 india   2005 population 453031
## 19 Nepal   2015 cases        199
## 20 Nepal   2015 population 898702

#The long format has two columns - "variable" and "value". The "variable" column contains the variable names "cases" and "population", and the "value" column contains the corresponding values for each observation. It allows us to easily filter, group, and plot the data based on the variable of interest.
```

c

```
df_log <- df %>% mutate(log_cases = log(cases), log_population = log(population))
df_log

## # A tibble: 10 × 6
##   country year cases population log_cases log_population
##   <chr>   <int> <int>      <int>      <dbl>      <dbl>
## 1 Nepal   2014    249   457623     5.52      13.0
## 2 india   2007    286   780391     5.66      13.6
## 3 india   2011    120   849520     4.79      13.7
## 4 india   2018    487   776301     6.19      13.6
## 5 Nepal   2019    494   350104     6.20      12.8
## 6 india   2007    457   294412     6.12      12.6
## 7 Nepal   2017    381   409335     5.94      12.9
## 8 Nepal   2014    314   665428     5.75      13.4
## 9 india   2005    485   453031     6.18      13.0
## 10 Nepal  2015    199   898702     5.29      13.7

#The above code will create two new columns in the tibble that contain the log of cases and the log of population, respectively. We can interpret this transformation to be useful when the original variable has a wide range of values.
```

d

