R Code Examples

Derek Chiu

2019-05-23

Example Data

A toy dataset is constructed below. We simulate 10 variables across 100 observations, creating various data types such as: integer, double, factor, ordered factor, logical, date, and character.

```r
# Set number of cases and random seed
library(magrittr)
n_cases <- 100
set.seed(1)

# Toy data with various data types
toy_data <- tibble::tibble(
  patient_id = seq_len(n_cases),
  age = rnorm(n = n_cases, mean = 55, sd = 5),
  bmi = rnorm(n = n_cases, mean = 23, sd = 2),
  date_dx = sample(seq(as.Date("2000-01-01"), as.Date("2010-12-31"), by = "day"), n_cases),
  stage = factor(sample(c("I", "II", "III", "IV"), size = n_cases, replace = TRUE)),
  grade = factor(sample(1:3, size = n_cases, replace = TRUE)),
  nodes = rbinom(n = n_cases, size = 5, prob = 0.2),
  feel = sample(forcats::fct_inorder(
    c("Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree"),
    ordered = TRUE), size = n_cases, replace = TRUE),
  alive = sample(c(TRUE, FALSE), size = n_cases, replace = TRUE),
  comment = sample(stringr::words, size = n_cases)
)

toy_data
```

```
#> # A tibble: 100 x 10
#>    patient_id   age   bmi date_dx    stage grade nodes feel   alive comment
#>         <int> <dbl> <dbl> <date>     <fct> <fct> <int> <ord>  <lgl> <chr>
#>  1          1  51.9  21.8 2007-04-01 III   3         2 Stron~ FALSE million
#>  2          2  55.9  23.1 2002-01-13 III   3         2 Stron~ FALSE quite
#>  3          3  50.8  21.2 2010-06-29 III   1         2 Stron~ TRUE  half
#>  4          4  63.0  23.3 2009-11-13 III   3         1 Agree  TRUE  we
#>  5          5  56.6  21.7 2010-05-15 II    3         1 Disag~ TRUE  budget
#>  6          6  50.9  26.5 2007-12-14 IV    3         0 Agree  TRUE  another
#>  7          7  57.4  24.4 2004-01-25 II    2         1 Disag~ TRUE  lot
#>  8          8  58.7  24.8 2008-07-29 IV    2         1 Stron~ TRUE  now
#>  9          9  57.9  23.8 2000-02-14 IV    3         3 Disag~ TRUE  struct~
#> 10         10  53.5  26.4 2010-04-27 III   1         1 Stron~ TRUE  power
#> # ... with 90 more rows
```

Filters

Use `==` to filter for equalities.

```r
toy_data %>%
  dplyr::filter(grade == 1)
```

```
#> # A tibble: 27 x 10
#>    patient_id   age   bmi date_dx    stage grade nodes feel   alive comment
#>         <int> <dbl> <dbl> <date>     <fct> <fct> <int> <ord>  <lgl> <chr>
#>  1          3  50.8  21.2 2010-06-29 III   1         2 Stron~ TRUE  half
#>  2         10  53.5  26.4 2010-04-27 III   1         1 Stron~ TRUE  power
#>  3         17  54.9  22.4 2005-06-02 III   1         0 Stron~ TRUE  televi~
#>  4         20  58.0  22.6 2009-01-03 III   1         2 Agree  FALSE eat
#>  5         22  58.9  25.7 2005-08-19 III   1         1 Neutr~ FALSE sort
#>  6         24  45.1  22.6 2001-07-25 I     1         1 Stron~ TRUE  stick
#>  7         25  58.1  22.8 2003-10-07 II    1         0 Stron~ FALSE close
#>  8         27  54.2  22.9 2000-12-06 IV    1         1 Disag~ FALSE worry
#>  9         33  56.9  24.1 2004-02-18 II    1         2 Stron~ TRUE  fine
#> 10         36  52.9  19.9 2008-09-28 I     1         0 Neutr~ FALSE contin~
#> # ... with 17 more rows
```

Use `&` or `,` to separate additional conditions.

```r
toy_data %>%
  dplyr::filter(grade == 1, stage == "II")
```

```
#> # A tibble: 9 x 10
#>   patient_id   age   bmi date_dx    stage grade nodes feel    alive comment
#>        <int> <dbl> <dbl> <date>     <fct> <fct> <int> <ord>   <lgl> <chr>
#> 1         25  58.1  22.8 2003-10-07 II    1         0 Strong~ FALSE close
#> 2         33  56.9  24.1 2004-02-18 II    1         2 Strong~ TRUE  fine
#> 3         37  53.0  22.4 2008-01-24 II    1         0 Strong~ FALSE four
#> 4         38  54.7  21.9 2000-07-26 II    1         2 Strong~ TRUE  load
#> 5         44  57.8  22.1 2006-08-20 II    1         0 Neutral TRUE  level
#> 6         56  64.9  20.8 2007-02-19 II    1         0 Strong~ TRUE  Christ~
#> 7         63  58.4  25.1 2000-01-08 II    1         2 Agree   TRUE  wind
#> 8         78  55.0  27.2 2004-09-14 II    1         2 Neutral FALSE agree
#> 9         98  52.1  21.0 2005-07-20 II    1         1 Strong~ TRUE  very
```

```r
toy_data %>%
  dplyr::filter(grade == 1 & stage == "II")
```

```
#> # A tibble: 9 x 10
#>   patient_id   age   bmi date_dx    stage grade nodes feel    alive comment
#>        <int> <dbl> <dbl> <date>     <fct> <fct> <int> <ord>   <lgl> <chr>
#> 1         25  58.1  22.8 2003-10-07 II    1         0 Strong~ FALSE close
#> 2         33  56.9  24.1 2004-02-18 II    1         2 Strong~ TRUE  fine
#> 3         37  53.0  22.4 2008-01-24 II    1         0 Strong~ FALSE four
#> 4         38  54.7  21.9 2000-07-26 II    1         2 Strong~ TRUE  load
#> 5         44  57.8  22.1 2006-08-20 II    1         0 Neutral TRUE  level
#> 6         56  64.9  20.8 2007-02-19 II    1         0 Strong~ TRUE  Christ~
#> 7         63  58.4  25.1 2000-01-08 II    1         2 Agree   TRUE  wind
#> 8         78  55.0  27.2 2004-09-14 II    1         2 Neutral FALSE agree
#> 9         98  52.1  21.0 2005-07-20 II    1         1 Strong~ TRUE  very
```

Pipe to `nrow()` to get number of cases.

```
toy_data %>%
  dplyr::filter(grade == 1, stage == "II") %>%
  nrow()
```

```
#> [1] 9
```

We can use inequalities for numeric variables (type `dbl`).

```
toy_data %>%
  dplyr::filter(age < 50, bmi >= 20)
```

```
#> # A tibble: 11 x 10
#>    patient_id   age   bmi date_dx    stage grade nodes feel   alive comment
#>         <int> <dbl> <dbl> <date>     <fct> <fct> <int> <ord>  <lgl> <chr>
#>  1         14  43.9  21.7 2008-09-10 III   2         2 Stron~ TRUE  without
#>  2         24  45.1  22.6 2001-07-25 I     1         1 Stron~ TRUE  stick
#>  3         28  47.6  22.9 2010-03-10 IV    2         1 Disag~ TRUE  egg
#>  4         35  48.1  23.6 2001-02-22 I     2         1 Neutr~ FALSE per
#>  5         54  49.4  21.1 2001-11-06 IV    2         0 Stron~ FALSE tax
#>  6         58  49.8  21.8 2010-09-14 I     2         0 Agree  TRUE  guess
#>  7         67  46.0  22.5 2007-09-29 I     3         1 Neutr~ TRUE  fact
#>  8         75  48.7  22.3 2001-04-09 IV    2         1 Neutr~ TRUE  want
#>  9         84  47.4  20.1 2010-03-14 I     2         1 Stron~ TRUE  along
#> 10         97  48.6  25.9 2004-08-29 II    3         0 Neutr~ FALSE address
#> 11         99  48.9  23.8 2009-01-30 II    3         2 Stron~ TRUE  once
```

Counts

We can tabulate counts for every level of a factor.

```
toy_data %>%
  dplyr::count(feel)
```

```
#> # A tibble: 5 x 2
#>   feel                  n
#>   <ord>             <int>
#> 1 Strongly Disagree    21
#> 2 Disagree             25
#> 3 Neutral              15
#> 4 Agree                17
#> 5 Strongly Agree       22
```

Bivariate counts also work.

```
toy_data %>%
  dplyr::count(feel, stage)
```

```
#> # A tibble: 20 x 3
#>    feel              stage     n
#>    <ord>             <fct> <int>
#>  1 Strongly Disagree I         2
#>  2 Strongly Disagree II        5
#>  3 Strongly Disagree III       8
#>  4 Strongly Disagree IV        6
#>  5 Disagree          I         4
#>  6 Disagree          II        5
#>  7 Disagree          III       6
#>  8 Disagree          IV       10
#>  9 Neutral           I         7
#> 10 Neutral           II        3
#> 11 Neutral           III       3
#> 12 Neutral           IV        2
#> 13 Agree             I         1
#> 14 Agree             II        6
#> 15 Agree             III       7
#> 16 Agree             IV        3
#> 17 Strongly Agree    I         5
#> 18 Strongly Agree    II        6
#> 19 Strongly Agree    III       7
#> 20 Strongly Agree    IV        4
```