Dplyr Manipulate Variables

Victoire Migashane

9/5/2022

```
head(df)
```

Dataset

```
## # A tibble: 6 x 8
##   species island    bill_length_mm bill_depth_mm flipper_l~1 body_~2 sex    year
##   <fct>   <fct>              <dbl>         <dbl>       <int>   <int> <fct> <int>
## 1 Adelie  Torgersen           39.1          18.7         181    3750 male   2007
## 2 Adelie  Torgersen           39.5          17.4         186    3800 fema~  2007
## 3 Adelie  Torgersen           40.3          18           195    3250 fema~  2007
## 4 Adelie  Torgersen           NA            NA            NA      NA <NA>   2007
## 5 Adelie  Torgersen           36.7          19.3         193    3450 fema~  2007
## 6 Adelie  Torgersen           39.3          20.6         190    3650 male   2007
## # ... with abbreviated variable names 1: flipper_length_mm, 2: body_mass_g
```

1. Pull Function

The `pull()` function lates you extract column values as a vector by name or index. `pull()` is very similar to `$`

Arguments .data: A data frame to full from.

var: a variable (column name, positive/negative integer)

name: Specifies the column to be used as names for a named vector.

. . . : for use by methods.

```
df %>%
  pull(species) %>% # not a data frame but the column cell values
  unique()
```

Usage

```
## [1] Adelie    Gentoo    Chinstrap
## Levels: Adelie Chinstrap Gentoo
```

```r
#similarly the $
df$species %>%
  unique()
```

```
## [1] Adelie    Gentoo    Chinstrap
## Levels: Adelie Chinstrap Gentoo
```

Relocate Function

The `relocate()` function is use to move a specified column to a different position in the data frame.

Arguments .data: A data frame to select from. ...: columns to move. .before: destination of the columns. .after: selected using the columns to move

```r
df %>%
  relocate(year, sex) %>%
  head(3)
```

Usage

```
## # A tibble: 3 x 8
##    year sex    species island    bill_length_mm bill_depth_mm flipper_~1 body_~2
##   <int> <fct>  <fct>   <fct>              <dbl>         <dbl>      <int>   <int>
## 1  2007 male   Adelie  Torgersen           39.1          18.7        181    3750
## 2  2007 female Adelie  Torgersen           39.5          17.4        186    3800
## 3  2007 female Adelie  Torgersen           40.3          18          195    3250
## # ... with abbreviated variable names 1: flipper_length_mm, 2: body_mass_g
```

3. Select Function

The `select()` function makes it easier to refer (select) variables by name or heper function.

Ex: a:f == Select all columns from `a` on the left to `f` on the right.

Arguments:

```
.data: A data frame to select from
...: One or more unquoted expressions separated by commas.
```

Helper function:

```
‘:‘: selecting a range of consecutive variables (select between)
‘-‘ : Select columns except
‘!‘: take the complement of a set of variables
‘c()‘: for combining selections.
‘& and |‘: for selecting the intersextion of twe sets of variables
‘everything()‘ : Select every column.
```

‘last_col()‘: Select the last column.
‘starts_with()‘: Select columns whose names starts with (characture string).
‘ends_with()‘: Select columns whose name ends with (characture string).
‘contains()‘: Select columns whose name contains (characture string).
‘matches()‘: Select columns whose name matches a regular expression.
‘num_range()‘: Select columns that mache a numerical range like x1,x2,x3
‘all_of(), any_of()‘: Select columns whose names are all present in a character vector.
‘one_of()‘: select columns whose names are in a group of names.
‘where()‘: select a variable with a function

```
# Return every column between bill_length_mm and body_mass_g
df %>%
    select(bill_length_mm:body_mass_g) %>%
    head(3)
```

Usage:

```
## # A tibble: 3 x 4
##   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##            <dbl>         <dbl>             <int>       <int>
## 1           39.1          18.7               181        3750
## 2           39.5          17.4               186        3800
## 3           40.3          18                 195        3250
```

```
# return every column that is not in the list
df %>%
    select(-c(island, species, sex, year)) %>%
    head(3)
```

```
## # A tibble: 3 x 4
##   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##            <dbl>         <dbl>             <int>       <int>
## 1           39.1          18.7               181        3750
## 2           39.5          17.4               186        3800
## 3           40.3          18                 195        3250
```

```
# return every column
df %>%
    select(everything()) %>%
    head(3)
```

```
## # A tibble: 3 x 8
##   species island    bill_length_mm bill_depth_mm flipper_l~1 body_~2 sex    year
##   <fct>   <fct>              <dbl>         <dbl>       <int>   <int> <fct> <int>
## 1 Adelie  Torgersen           39.1          18.7         181    3750 male   2007
## 2 Adelie  Torgersen           39.5          17.4         186    3800 fema~  2007
## 3 Adelie  Torgersen           40.3          18           195    3250 fema~  2007
## # ... with abbreviated variable names 1: flipper_length_mm, 2: body_mass_g
```

```r
# return the last column
df %>%
    select(last_col()) %>%
    head(3)
```

```
## # A tibble: 3 x 1
##    year
##   <int>
## 1  2007
## 2  2007
## 3  2007
```

```r
# return every column that starts with b and ends with m
df %>%
    select(starts_with("b") & ends_with("m")) %>%
    head(3)
```

```
## # A tibble: 3 x 2
##   bill_length_mm bill_depth_mm
##            <dbl>         <dbl>
## 1           39.1          18.7
## 2           39.5          17.4
## 3           40.3          18
```

```r
# return column names that contain the string land
df %>%
    select(contains("land")) %>%
    head(3)
```

```
## # A tibble: 3 x 1
##   island
##   <fct>
## 1 Torgersen
## 2 Torgersen
## 3 Torgersen
```

```r
# return columns whose names are all present in a character vector
# also works for any_of()
df %>%
    select(all_of(c("bill_length_mm", "flipper_length_mm", "body_mass_g"))) %>%
    head(3)
```

```
## # A tibble: 3 x 3
##   bill_length_mm flipper_length_mm body_mass_g
##            <dbl>             <int>       <int>
## 1           39.1               181        3750
## 2           39.5               186        3800
## 3           40.3               195        3250
```

```r
# return columns whoes data type are integers
df %>%
    select(where(is.numeric)) %>%
    head(3)
```

```
## # A tibble: 3 x 5
##   bill_length_mm bill_depth_mm flipper_length_mm body_mass_g  year
##            <dbl>         <dbl>             <int>       <int> <int>
## 1           39.1          18.7               181        3750  2007
## 2           39.5          17.4               186        3800  2007
## 3           40.3          18                 195        3250  2007
```