Reshaping Data (tidyr) and Grouping Data (dplyr)

Victoire Migashane

9/6/2022

```
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. Pivot longer function

The `pivot_longer()` function is the replacement of the `gather()` function. The `pivot_longer()` function is used to gather columns into rows, by increasing the numnber of rows and decreasing the nunmber of coulmns.

Arguments

```
data: A data frame to pivot.
cols: columns to pivot to a longer format.
names_to: specify the new column names.
names_prefix: A regular expression used to remove matching text from the start of each variable name.
names_sep, names_pattern: control how column names are broken up if there are multple columns.
names_ptypes, values_ptypes: Used to verify wheather the created columns are in expected data types.
names_transform, values_tranform: Used to change data type of a specific column.
names_repair: Controls what happens if the ouput has invalid column names.
values_to: A string specifying the name of the column to create from the data stored in cell values.
values_drop_na: drop rows that contain NAs if TRUE.
...: Additional arguments passed on the methods.
```

```
# # Convert columns to rows
long_df <- df %>%
    pivot_longer(cols= c(bill_length_mm, bill_depth_mm, flipper_length_mm), names_to = "body part", valu
    head(24)
long_df
```

Usage:

```
## # A tibble: 24 x 7
##    species island    body_mass_g sex     year ‘body part‘       length_mm
##    <fct>   <fct>           <int> <fct>  <int> <chr>                 <dbl>
##  1 Adelie  Torgersen        3750 male    2007 bill_length_mm         39.1
##  2 Adelie  Torgersen        3750 male    2007 bill_depth_mm          18.7
##  3 Adelie  Torgersen        3750 male    2007 flipper_length_mm     181
##  4 Adelie  Torgersen        3800 female  2007 bill_length_mm         39.5
##  5 Adelie  Torgersen        3800 female  2007 bill_depth_mm          17.4
##  6 Adelie  Torgersen        3800 female  2007 flipper_length_mm     186
##  7 Adelie  Torgersen        3250 female  2007 bill_length_mm         40.3
##  8 Adelie  Torgersen        3250 female  2007 bill_depth_mm          18
##  9 Adelie  Torgersen        3250 female  2007 flipper_length_mm     195
## 10 Adelie  Torgersen        3450 female  2007 bill_length_mm         36.7
## # ... with 14 more rows
```

2. Pivot wider function

The reverse function for `pivot_longer()` is `pivot_wider()` which will increase the number of columns and decrease the number of rows. `pivot_wider()` is the replacement of the `spread()` function.

Arguments

```
data: A data frame to pivot.
id_cols: A set of columns that uniquely identifies each obsevation.
id_expand: Should the values in the ‘id_cols‘ columns be expanded by ‘expand()‘ before pivoting.
names_from, values_from: Which column to get the name of the output column.
names_prefix: String added to the start of theevery variable name.
names_sep: used to join values of ‘names_from‘ or ‘values_from‘ into a single string to use as a column
names_glue: Used to supply a glue specifivation that uses the ‘names_from‘ columns to create custom colu
names_sort: Should the new columns be sorted?
names_vary: When ‘names_from‘ identifies a column (or columns) with multiple unique values, and multiple
names_expand: Should the values in the ‘names_from‘ column be expanded by ‘expand()‘ before pivoting?
names_pair: What happens if the output has invalid columns names?
values_fill: A value tha specifies what each value should be filled in with when missing.
value_fn: function applied to the value in each cell in the output.
unused_fn: Function applied to summaruze the valies from the unused colomns.
...: Additional argumets passed on the methods.
```

```
# Convert rows to columns
long_df %>%
    pivot_wider(names_from = `body part`
, values_from = `length_mm`, names_prefix= "new_")
```

Usage

```
## # A tibble: 8 x 8
##   species island    body_mass_g sex     year new_bill_length_mm new_bi~1 new_f~2
##   <fct>   <fct>           <int> <fct>  <int>              <dbl>    <dbl>   <dbl>
## 1 Adelie  Torgersen        3750 male    2007               39.1     18.7     181
## 2 Adelie  Torgersen        3800 female  2007               39.5     17.4     186
## 3 Adelie  Torgersen        3250 female  2007               40.3     18       195
## 4 Adelie  Torgersen        3450 female  2007               36.7     19.3     193
## 5 Adelie  Torgersen        3650 male    2007               39.3     20.6     190
## 6 Adelie  Torgersen        3625 female  2007               38.9     17.8     181
## 7 Adelie  Torgersen        4675 male    2007               39.2     19.6     195
## 8 Adelie  Torgersen        3475 <NA>    2007               34.1     18.1     193
## # ... with abbreviated variable names 1: new_bill_depth_mm,
## #   2: new_flipper_length_mm
```

3. Unite function

The `unite()` function paste(unite) multiple columns into on.

Arguments:

```
data: A data frame to unite
col: The name of the new column
...: Columns to unite
sep: Seperator to use between values
remove: Remove input columns from output data frame if TRUE
na.rm: Missing values will be removed prior to uniting each value if TRUE
```

```


united_col <- df %>%
    unite("body_size", c(bill_length_mm, bill_depth_mm,flipper_length_mm), sep= "; ", remove =TRUE) %>%
    head()
united_col
```

Usage

```
## # A tibble: 6 x 6
##   species island    body_size       body_mass_g sex     year
##   <fct>   <fct>     <chr>                 <int> <fct>  <int>
## 1 Adelie  Torgersen 39.1; 18.7; 181        3750 male    2007
## 2 Adelie  Torgersen 39.5; 17.4; 186        3800 female  2007
## 3 Adelie  Torgersen 40.3; 18; 195          3250 female  2007
## 4 Adelie  Torgersen NA; NA; NA               NA <NA>    2007
## 5 Adelie  Torgersen 36.7; 19.3; 193        3450 female  2007
## 6 Adelie  Torgersen 39.3; 20.6; 190        3650 male    2007
```

4. Separate function

The `separate()` function turns a character column to multiple columns with either a regular expression or a vector of charcter positions.

Arguments

```
data: A data from separate
col: coulmn name or position
into: Names of new variables to create as character vector
sep: Separator between columns
remove: Remove input columns from out put data frame if TRUE
convert: If TRUE, will run ‘type.covert()‘ with ‘as.is = TRUE‘ on new columns.
extra: Controls what happens when there are to may pieces if ‘sep‘ if a charcter vector
fill: Controls what happens when there are not enough pieces if ‘sep‘ is a charcter vector
...: Additional arguments passed to the methods.
```

```
united_col %>%
    separate(col= `body_size`, into= c("new_bill_length", "new_bill_depth", "new_flipper_length"), sep =
```

Usage

```
## # A tibble: 6 x 8
##   species island    new_bill_length new_bill_depth new_fli~1 body_~2 sex    year
##   <fct>   <fct>     <chr>           <chr>          <chr>       <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: new_flipper_length, 2: body_mass_g
```

5. Group by function

The `group_by()` function is used to create a grouped copy of a table by columns. `group_by()` takes an existing `tbl` and converts it into a grouped `tbl` where operations are performed by group.

Arguments

```
.data: A data frame to group.
...: Variables to group by.
.add: Overide existing groups when FALSE (default)
.drop: Drop groups formed by vector levels that don’t appear in the data.
x: A table ‘tbl()‘
```

```
df %>%
    group_by(island, sex) %>%
    count(species, name = "count")
```

Usage

```
## # A tibble: 13 x 4
## # Groups:   island, sex [9]
##    island    sex    species   count
##    <fct>     <fct>  <fct>     <int>
##  1 Biscoe    female Adelie       22
##  2 Biscoe    female Gentoo       58
##  3 Biscoe    male   Adelie       22
##  4 Biscoe    male   Gentoo       61
##  5 Biscoe    <NA>   Gentoo        5
##  6 Dream     female Adelie       27
##  7 Dream     female Chinstrap    34
##  8 Dream     male   Adelie       28
##  9 Dream     male   Chinstrap    34
## 10 Dream     <NA>   Adelie        1
## 11 Torgersen female Adelie       24
## 12 Torgersen male   Adelie       23
## 13 Torgersen <NA>   Adelie        5
```