Dplyr Subset Observations

Victoire Migashane

9/4/2022

```
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. Filter Function

The `filter()` function is used to extract rows of a data frame that satisfy logical conditions.

Arguments

```
.data: A data frame to extract from
...: Expressions that return a logical value, adn defined in terms of the variables in the data frame.
.preserve: if FALSE (deafult) the grouping structure is recalculated based on the resulting data.
```

Useful functions

```
== : equal to, != : not equal to,
> : greater than, < : less than,
>= : greater than or equal to,
<= : less than or eual to,
& : and, | : or, ! : not,
%in% = in, xor(): or,
is.na() : is none, !is.na() isn't none,
between(): is between, near(): is near
```

```
df %>%
    filter(species == "Adelie" & sex == "male" & between(bill_length_mm, 38,39))
```

Usage

```
## # A tibble: 7 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           38.6          21.2         191    3800 male   2007
## 2 Adelie  Biscoe              38.2          18.1         185    3950 male   2007
## 3 Adelie  Biscoe              38.8          17.2         180    3800 male   2007
## 4 Adelie  Dream               38.8          20           190    3950 male   2007
## 5 Adelie  Dream               38.3          19.2         189    3950 male   2008
## 6 Adelie  Biscoe              38.2          20           190    3900 male   2009
## 7 Adelie  Dream               39            18.7         185    3650 male   2009
## # ... with abbreviated variable names 1: flipper_length_mm, 2: body_mass_g
```

```
df %>%
    filter(between(flipper_length_mm, 180,190) | is.na(bill_depth_mm),sort = TRUE) %>%
    count(species)
```

```
## # A tibble: 3 x 2
##   species       n
##   <fct>     <int>
## 1 Adelie       78
## 2 Chinstrap    14
## 3 Gentoo        1
```

2. Slice Function

The `slice()` function lets you select rows by their index positions.

Arguments

```
.data: A data frame to select from.
...: the index positions (integer row values)
.preserve(): False: false the grouping structure is recalculated based on the resulting data, other wis
order_by: the variable(column) to order by.
with_ties: if TRUE return more rows than you request, other wise return one row.
weight_by: must evaluate a vector of positive numbers an that all sum to 100%.
repace: if TRUE, sampling should be performed with replacement.
```

Related slice methodes

```
slice_head(): Returns the first few rows.
slice_tail(): Returns the last few rows.
slice_min(): Returns the minimum value in the specified column
slice_max(): Returns the maximum value in teh specified column
slice_sample(): Return the sample of specified lengh
```

```
# return row 5 to row 10
df %>% slice(5:10)
```

Usage

```
## # 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           36.7          19.3         193    3450 fema~  2007
## 2 Adelie  Torgersen           39.3          20.6         190    3650 male   2007
## 3 Adelie  Torgersen           38.9          17.8         181    3625 fema~  2007
## 4 Adelie  Torgersen           39.2          19.6         195    4675 male   2007
## 5 Adelie  Torgersen           34.1          18.1         193    3475 <NA>   2007
## 6 Adelie  Torgersen           42            20.2         190    4250 <NA>   2007
## # ... with abbreviated variable names 1: flipper_length_mm, 2: body_mass_g
```

```
# return the first 3 rows
df %>% slice_head(n = 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
```

```
# return the random 8 rows
df %>% slice_sample(n = 8)
```

```
## # A tibble: 8 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  Biscoe              43.2          19           197    4775 male   2009
## 2 Adelie  Dream               40.8          18.9         208    4300 male   2008
## 3 Adelie  Biscoe              45.6          20.3         191    4600 male   2009
## 4 Gentoo  Biscoe              47.3          15.3         222    5250 male   2007
## 5 Adelie  Biscoe              35.9          19.2         189    3800 fema~  2007
## 6 Adelie  Torgersen           38.9          17.8         181    3625 fema~  2007
## 7 Adelie  Dream               38.3          19.2         189    3950 male   2008
## 8 Gentoo  Biscoe              44            13.6         208    4350 fema~  2008
## # ... with abbreviated variable names 1: flipper_length_mm, 2: body_mass_g
```

```
# return the min value in a column
df %>% slice_min(body_mass_g )
```

```
## # A tibble: 1 x 8
##   species   island bill_length_mm bill_depth_mm flipper_le~1 body_~2 sex    year
##   <fct>     <fct>           <dbl>         <dbl>        <int>   <int> <fct> <int>
## 1 Chinstrap Dream            46.9          16.6          192    2700 fema~  2008
## # ... with abbreviated variable names 1: flipper_length_mm, 2: body_mass_g
```

3. Distict Function

The `distinct()` function selects only unique rows from from a data frame, removing all rows with duplicated values. This is similar to unique() but faster.

Arguments

```
.data: A data frame to select from.
...: Optional variable to use when determining uniqueness.
.keep_all: if TRUE, keep all variables in the data frame if ... is not distinct
```

```
df %>%
    distinct(island)
```

Usage

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

```
# similar to distinct is count()
df %>%
    count(island)
```

```
## # A tibble: 3 x 2
##   island        n
##   <fct>     <int>
## 1 Biscoe      168
## 2 Dream       124
## 3 Torgersen    52
```

4. Sample Function

The `sample()` function retunrn a random sample of specified zise from the elements of argument x. `sample()` takes the place for `sample_frac()` and `sample_n()`.

Arguments

```
x: A vector of one or more elements or a positive integer value.
n: The number of items to choose form.
size: The number of items to return.
replace: if TRUE, sampling should be performed with replacement.
prob: A vector of probability weights to get the elements of the sample vector.
useHash: indicats if the hash-version of the algorithm should be used.
```

Note: Install `sampling` package from CRAN for other methods of weighted sampling.

```
sample(c(0,1),10,  size = 4)
```

Usage

```
## [1] 0 0 1 0
```