

dyplr library vignette

Oscar A. Trevizo

Contents

Purpose	1
Libraries	1
Load possum dataset from DAAG	2
Pipes	3
SELECT	3
FILTER	5
ARRANGE	6
SUMMARISE	6
GROUP BY	7
Example: arrange() on that table created by group_by()	7
Create a new table (mutate)	7
References	8

Purpose

This vignette aims to introduce you dplyr library. It is here for learning purposes.

- {dplyr} (pronounced d - plier dataset plier... pliers to trim data)

Libraries

- Load dplyr

Tip: Shortcut select a word and click quotations to automate.

It has several functions or methods

- Pipes %>% to chain commands

Load possum dataset from DAAG

```
# Add DAAG to pull some data
library(DAAG)

# For example DAAG has 'possum'

# data()

# From the console, you can do:
# > ?datasetname
```

```
data('possum')
str(possum)
```

```
## 'data.frame': 104 obs. of 14 variables:
## $ case : num 1 2 3 4 5 6 7 8 9 10 ...
## $ site : num 1 1 1 1 1 1 1 1 1 1 ...
## $ Pop : Factor w/ 2 levels "Vic","other": 1 1 1 1 1 1 1 1 1 1 ...
## $ sex : Factor w/ 2 levels "f","m": 2 1 1 1 1 1 2 1 1 1 ...
## $ age : num 8 6 6 6 2 1 2 6 9 6 ...
## $ hdlngth : num 94.1 92.5 94 93.2 91.5 93.1 95.3 94.8 93.4 91.8 ...
## $ skullw : num 60.4 57.6 60 57.1 56.3 54.8 58.2 57.6 56.3 58 ...
## $ totlngth: num 89 91.5 95.5 92 85.5 90.5 89.5 91 91.5 89.5 ...
## $ taill : num 36 36.5 39 38 36 35.5 36 37 37 37.5 ...
## $ footlngth: num 74.5 72.5 75.4 76.1 71 73.2 71.5 72.7 72.4 70.9 ...
## $ earconch: num 54.5 51.2 51.9 52.2 53.2 53.6 52 53.9 52.9 53.4 ...
## $ eye : num 15.2 16 15.5 15.2 15.1 14.2 14.2 14.5 15.5 14.4 ...
## $ chest : num 28 28.5 30 28 28.5 30 30 29 28 27.5 ...
## $ belly : num 36 33 34 34 33 32 34.5 34 33 32 ...
```

```
summary(possum)
```

```
##      case      site      Pop      sex      age
## Min.   : 1.00   Min.   :1.000   Vic :46   f:43   Min.   :1.000
## 1st Qu.: 26.75   1st Qu.:1.000   other:58  m:61   1st Qu.:2.250
## Median : 52.50   Median :3.000
## Mean   : 52.50   Mean   :3.625
## 3rd Qu.: 78.25   3rd Qu.:6.000
## Max.   :104.00   Max.   :7.000
##                               NA's   :2
##      hdlngth      skullw      totlngth      taill
## Min.   : 82.50   Min.   :50.00   Min.   :75.00   Min.   :32.00
## 1st Qu.: 90.67   1st Qu.:54.98   1st Qu.:84.00   1st Qu.:35.88
## Median : 92.80   Median :56.35   Median :88.00   Median :37.00
## Mean   : 92.60   Mean   :56.88   Mean   :87.09   Mean   :37.01
## 3rd Qu.: 94.72   3rd Qu.:58.10   3rd Qu.:90.00   3rd Qu.:38.00
## Max.   :103.10   Max.   :68.60   Max.   :96.50   Max.   :43.00
##
##      footlngth      earconch      eye      chest      belly
## Min.   :60.30   Min.   :40.30   Min.   :12.80   Min.   :22.0   Min.   :25.00
```

```
## 1st Qu.:64.60 1st Qu.:44.80 1st Qu.:14.40 1st Qu.:25.5 1st Qu.:31.00
## Median :68.00 Median :46.80 Median :14.90 Median :27.0 Median :32.50
## Mean :68.46 Mean :48.13 Mean :15.05 Mean :27.0 Mean :32.59
## 3rd Qu.:72.50 3rd Qu.:52.00 3rd Qu.:15.72 3rd Qu.:28.0 3rd Qu.:34.12
## Max. :77.90 Max. :56.20 Max. :17.80 Max. :32.0 Max. :40.00
## NA's :1
```

```
# Try from the console:
# > ?possum
```

Pipes

- Mac shortcut shift-command-m for %>% (that is from {dplyr})
- To be demonstrated throughout this vignette

SELECT

- To specific column, by column number or column 'name'

```
# Select specific columns.
#
possum %>% select(2:3, 4:7)
```

```
##      site  Pop sex age hdlngth skullw
## C3      1  Vic  m   8    94.1    60.4
## C5      1  Vic  f   6    92.5    57.6
## C10     1  Vic  f   6    94.0    60.0
## C15     1  Vic  f   6    93.2    57.1
## C23     1  Vic  f   2    91.5    56.3
## C24     1  Vic  f   1    93.1    54.8
## C26     1  Vic  m   2    95.3    58.2
## C27     1  Vic  f   6    94.8    57.6
## C28     1  Vic  f   9    93.4    56.3
## C31     1  Vic  f   6    91.8    58.0
## C32     1  Vic  f   9    93.3    57.2
## C34     1  Vic  f   5    94.9    55.6
## C36     1  Vic  m   5    95.1    59.9
## C37     1  Vic  m   3    95.4    57.6
## C39     1  Vic  m   5    92.9    57.6
## C40     1  Vic  m   4    91.6    56.0
## C45     1  Vic  f   1    94.7    67.7
## C47     1  Vic  m   2    93.5    55.7
## C48     1  Vic  f   5    94.4    55.4
## C50     1  Vic  f   4    94.8    56.3
## C54     1  Vic  f   3    95.9    58.1
## C55     1  Vic  m   3    96.3    58.5
## C58     1  Vic  f   4    92.5    56.1
## C59     1  Vic  m   2    94.4    54.9
## C60     1  Vic  m   3    95.8    58.5
## C61     1  Vic  m   7    96.0    59.0
## C63     1  Vic  f   2    90.5    54.5
```

## C64	1	Vic	m	4	93.8	56.8
## A1	1	Vic	f	3	92.8	56.0
## A2	1	Vic	f	2	92.1	54.4
## A3	1	Vic	m	3	92.8	54.1
## A4	1	Vic	f	4	94.3	56.7
## AD1	1	Vic	m	3	91.4	54.6
## BB4	2	Vic	m	2	90.6	55.7
## BB13	2	Vic	m	4	94.4	57.9
## BB15	2	Vic	m	7	93.3	59.3
## BB17	2	Vic	f	2	89.3	54.8
## BB25	2	Vic	m	7	92.4	56.0
## BB31	2	Vic	f	1	84.7	51.5
## BB33	2	Vic	f	3	91.0	55.0
## BB36	2	Vic	f	5	88.4	57.0
## BB38	2	Vic	m	3	85.3	54.1
## BB40	2	Vic	f	2	90.0	55.5
## BB41	2	Vic	m	NA	85.1	51.5
## BB44	2	Vic	m	3	90.7	55.9
## BB45	2	Vic	m	NA	91.4	54.4
## WW1	3	other	m	2	90.1	54.8
## WW2	3	other	m	5	98.6	63.2
## WW3	3	other	m	4	95.4	59.2
## WW4	3	other	f	5	91.6	56.4
## WW5	3	other	f	5	95.6	59.6
## WW6	3	other	m	6	97.6	61.0
## WW7	3	other	f	3	93.1	58.1
## BR1	4	other	m	7	96.9	63.0
## BR2	4	other	m	2	103.1	63.2
## BR3	4	other	m	3	99.9	61.5
## BR4	4	other	f	4	95.1	59.4
## BR5	4	other	m	3	94.5	64.2
## BR6	4	other	m	2	102.5	62.8
## BR7	4	other	f	2	91.3	57.7
## CD1	5	other	m	7	95.7	59.0
## CD2	5	other	f	3	91.3	58.0
## CD3	5	other	f	6	92.0	56.4
## CD4	5	other	f	3	96.9	56.5
## CD5	5	other	f	5	93.5	57.4
## CD6	5	other	f	3	90.4	55.8
## CD7	5	other	m	4	93.3	57.6
## CD8	5	other	m	5	94.1	56.0
## CD9	5	other	m	5	98.0	55.6
## CD10	5	other	f	7	91.9	56.4
## CD11	5	other	m	6	92.8	57.6
## CD12	5	other	m	1	85.9	52.4
## CD13	5	other	m	1	82.5	52.3
## BSF1	6	other	f	4	88.7	52.0
## BSF2	6	other	m	6	93.8	58.1
## BSF3	6	other	m	5	92.4	56.8
## BSF4	6	other	m	6	93.6	56.2
## BSF5	6	other	m	1	86.5	51.0
## BSF6	6	other	m	1	85.8	50.0
## BSF7	6	other	m	1	86.7	52.6
## BSF8	6	other	m	3	90.6	56.0

```
## BSF9      6 other f 4 86.0 54.0
## BSF10     6 other f 3 90.0 53.8
## BSF11     6 other m 3 88.4 54.6
## BSF12     6 other m 3 89.5 56.2
## BSF13     6 other f 3 88.2 53.2
## BTP1      7 other m 2 98.5 60.7
## BTP3      7 other f 2 89.6 58.0
## BTP4      7 other m 6 97.7 58.4
## BTP5      7 other m 3 92.6 54.6
## BTP6      7 other m 3 97.8 59.6
## BTP7      7 other m 2 90.7 56.3
## BTP8      7 other m 3 89.2 54.0
## BTP9      7 other m 7 91.8 57.6
## BTP10     7 other m 4 91.6 56.6
## BTP12     7 other m 4 94.8 55.7
## BTP13     7 other m 3 91.0 53.1
## BTP14     7 other m 5 93.2 68.6
## BTP15     7 other f 3 93.3 56.2
## BTP16     7 other m 1 89.5 56.0
## BTP17     7 other m 1 88.6 54.7
## BTP19     7 other f 6 92.4 55.0
## BTP20     7 other m 4 91.5 55.2
## BTP21     7 other f 3 93.6 59.9
```

FILTER

```
# This example shows a filter with multiple conditions.
#
possum %>% filter(sex == 'f', Pop == 'Vic', age < 4)
```

```
##      case site Pop sex age hdlngth skullw totlngth taill footlngth earconch eye
## C23      5   1 Vic f 2  91.5  56.3   85.5  36.0   71.0   53.2 15.1
## C24      6   1 Vic f 1  93.1  54.8   90.5  35.5   73.2   53.6 14.2
## C45     17   1 Vic f 1  94.7  67.7   89.5  36.5   73.2   53.2 14.7
## C54     21   1 Vic f 3  95.9  58.1   96.5  39.5   77.9   52.9 14.2
## C63     27   1 Vic f 2  90.5  54.5   85.0  35.0   70.3   50.8 14.2
## A1      29   1 Vic f 3  92.8  56.0   88.0  35.0   74.9   51.8 14.0
## A2      30   1 Vic f 2  92.1  54.4   84.0  33.5   70.6   50.8 14.5
## BB17    37   2 Vic f 2  89.3  54.8   82.5  35.0   71.2   52.0 13.6
## BB31    39   2 Vic f 1  84.7  51.5   75.0  34.0   68.7   53.4 13.0
## BB33    40   2 Vic f 3  91.0  55.0   84.5  36.0   72.8   51.4 13.6
## BB40    43   2 Vic f 2  90.0  55.5   81.0  32.0   72.0   49.4 13.4
##      chest belly
## C23    28.5  33.0
## C24    30.0  32.0
## C45    29.0  31.0
## C54    30.0  40.0
## C63    23.0  28.0
## A1     24.0  32.0
## A2     24.5  33.0
## BB17   28.0  31.5
## BB31   25.0  25.0
```

```
## BB33 27.0 30.0
## BB40 29.0 31.0
```

ARRANGE

- A type of sort

```
# Arrange, or sort
# Here we start to pipe using multiple lines.
#
possum %>% filter(sex == 'f', Pop == 'Vic', age < 4) %>%
  arrange(desc(belly))
```

```
##      case site Pop sex age hdlngth skullw totlngth taill footlngth earconch eye
## C54      21   1 Vic  f  3   95.9   58.1    96.5  39.5    77.9    52.9 14.2
## C23       5   1 Vic  f  2   91.5   56.3    85.5  36.0    71.0    53.2 15.1
## A2       30   1 Vic  f  2   92.1   54.4    84.0  33.5    70.6    50.8 14.5
## C24       6   1 Vic  f  1   93.1   54.8    90.5  35.5    73.2    53.6 14.2
## A1       29   1 Vic  f  3   92.8   56.0    88.0  35.0    74.9    51.8 14.0
## BB17     37   2 Vic  f  2   89.3   54.8    82.5  35.0    71.2    52.0 13.6
## C45      17   1 Vic  f  1   94.7   67.7    89.5  36.5    73.2    53.2 14.7
## BB40     43   2 Vic  f  2   90.0   55.5    81.0  32.0    72.0    49.4 13.4
## BB33     40   2 Vic  f  3   91.0   55.0    84.5  36.0    72.8    51.4 13.6
## C63      27   1 Vic  f  2   90.5   54.5    85.0  35.0    70.3    50.8 14.2
## BB31     39   2 Vic  f  1   84.7   51.5    75.0  34.0    68.7    53.4 13.0
##      chest belly
## C54    30.0 40.0
## C23    28.5 33.0
## A2     24.5 33.0
## C24    30.0 32.0
## A1     24.0 32.0
## BB17   28.0 31.5
## C45    29.0 31.0
## BB40   29.0 31.0
## BB33   27.0 30.0
## C63    23.0 28.0
## BB31   25.0 25.0
```

SUMMARISE

- You can introduce functions or equations and summarise.

```
# summarise() with multiple functions... Avg, SD...
#
possum %>% filter(sex == 'f', Pop == 'Vic', age < 4) %>%
  arrange(desc(belly)) %>%
  summarise(Avg = mean(belly),
            SD = sd(belly),
            count = n())
```

```
##      Avg      SD count
## 1 31.5 3.667424    11
```

GROUP BY

- It creates a table.

```
# group_by() before summarising.
#
possum %>% filter(sex == 'm') %>%
  group_by(site) %>%
  summarise(Avg = mean(belly),
            SD = sd(belly),
            count = n())
```

```
## # A tibble: 7 x 4
##   site   Avg    SD count
##   <dbl> <dbl> <dbl> <int>
## 1     1  33.2  2.49    14
## 2     2  32.1  3.37     8
## 3     3  34    1.47     4
## 4     4  34.6  2.22     5
## 5     5  30.9  2.28     7
## 6     6  31.5  2.78     9
## 7     7  31.8  2.25    14
```

Example: `arrange()` on that table created by `group_by()`

```
# Add another function, descending order
possum %>% filter(sex == 'm') %>%
  group_by(site) %>%
  summarise(Avg = mean(belly),
            SD = sd(belly),
            count = n()) %>%
  arrange(desc(Avg))
```

```
## # A tibble: 7 x 4
##   site   Avg    SD count
##   <dbl> <dbl> <dbl> <int>
## 1     4  34.6  2.22     5
## 2     3  34    1.47     4
## 3     1  33.2  2.49    14
## 4     2  32.1  3.37     8
## 5     7  31.8  2.25    14
## 6     6  31.5  2.78     9
## 7     5  30.9  2.28     7
```

Create a new table (`mutate`)

```
# New variable TR
mytable <- possum %>%
  group_by(site) %>%
```

```

summarise(TR = sum(taill) / sum(totlngth),
          count = n()) %>%
arrange(desc(TR))

print(mytable)

```

```

## # A tibble: 7 x 3
##   site    TR count
##   <dbl> <dbl> <int>
## 1     6 0.445    13
## 2     7 0.440    18
## 3     5 0.433    13
## 4     4 0.431     7
## 5     2 0.426    13
## 6     3 0.423     7
## 7     1 0.406    33

```

References

Dr. Bharatendra <https://www.youtube.com/watch?v=rsfV57N7Uns&list=PL34t5iLfZddtUUABMikey6NtL05hPAp42&index=10>