Four Page dplyr

Andy Grogan-Kaylor

2018-10-18

Contents

1	Background	1
2	Sample Data	1
3	Piping	2
4	Select A Subset of Variables: select()	2
5	Filter A Subset of Rows: filter()	2
6	Create New Variables: mutate()	2
7	Recode Variables: mutate()	3
8	Rename Variables: rename()	3
9	Drop Missing Values: filter()	3
10	Connecting To Other Packages Like ggplot	4

1 Background

dplyr is a very powerful R library for managing and processing data.

While dplyr is very powerful, learning to use dplyr can be very confusing. This guide aims to present some of the most common dplyr functions and commands in the form of a brief cheatsheet.

library(dplyr)

2 Sample Data

year	Х	у	Z
2007	NA	Group B	87.25
2012	44.94	Group A	89.56
2017	46.34	Group A	78.24

year	Х	у	Z
2001	31.18	Group A	100.3
2010	36.65	Group A	81.47

3 Piping

Pipes %>% connect pieces of a command e.g. data to data wrangling to a graph command.

4 Select A Subset of Variables: select()

mynewdata <- mydata %>% select(x,y) # select only x and y

X	у
NA	Group B
44.94	Group A
46.34	Group A
31.18	Group A
36.65	Group A

5 Filter A Subset of Rows: filter()

mynewdata <- mydata %>%
filter(year > 2010) # filter on year

year	Х	у	Z
2012	44.94	Group A	89.56
2017	46.34	Group A	78.24

6 Create New Variables: mutate()

mynewdata <- mydata %>%

mutate(myscale = x + z) # create a new variable e.g. a scale

year	Х	у	Z	myscale
2007	NA	Group B	87.25	NA
2012	44.94	Group A	89.56	134.5

year	Х	у	Z	myscale
2017	46.34	Group A	78.24	124.6
2001	31.18	Group A	100.3	131.5
2010	36.65	Group A	81.47	118.1

7 Recode Variables: mutate()

year	Х	у	Z	zcategorical
2007	NA	Group B	87.25	low
2012	44.94	Group A	89.56	low
2017	46.34	Group A	78.24	low
2001	31.18	Group A	100.3	high
2010	36.65	Group A	81.47	low

8 Rename Variables: rename()

```
newdata <- mydata %>%
rename(age = x, # rename
    mental health = z)
```

year	age	у	mental_health
2007	NA	Group B	87.25
2012	44.94	Group A	89.56
2017	46.34	Group A	78.24
2001	31.18	Group A	100.3
2010	36.65	Group A	81.47

9 Drop Missing Values: filter()

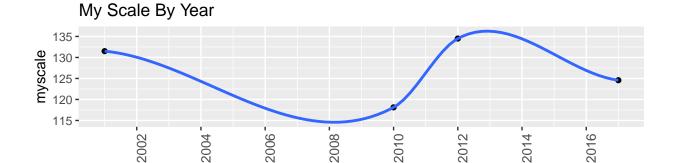
```
newdata <- mydata %>%
filter(!is.na(x)) # filter by x is not missing
```

year	Х	у	Z
2012	44.94	Group A	89.56
2017	46.34	Group A	78.24
2001	31.18	Group A	100.3
2010	36.65	Group A	81.47

10 Connecting To Other Packages Like ggplot

Notice how, in the code below, I never actually create the new data set mynewdata. I simply pipe mydata into a dplyr command, and pipe the result directly to ggplot2.

```
library(ggplot2)
```



year