

Information about the tidyverse

Maksim Nikiforov

Contents

1	Code to Create this Document	1
2	R packages for data science	1
3	Some Core Packages	1
3.1	dplyr	1
3.2	ggplot2	2
3.3	readr	3
3.4	tidyr	3

1 Code to Create this Document

```
#This is where your code with the render function should appear.  
#Remember that you don't want this code chunk to evaluate but  
#you do want it to show up in your document.
```

2 R packages for data science

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```

3 Some Core Packages

The four **core** packages that we'll use the most are given below along with their purpose and a quick example of some functionality.

3.1 dplyr



[dplyr is a grammar of data manipulation](#), providing a consistent set of verbs that help you solve the most common data manipulation challenges:

- `mutate()` adds new variables that are functions of existing variables
- `select()` picks variables based on their names.
- `filter()` picks cases based on their values.
- `summarise()` reduces multiple values down to a single summary.
- `arrange()` changes the ordering of the rows.

These all combine naturally with `group_by()` which allows you to perform any operation “by group”. You can learn more about them in `vignette(“dplyr”)`. As well as these single-table verbs, dplyr also provides a variety of two-table verbs, which you can learn about in `vignette(“two-table”)`.

If you are new to dplyr, the best place to start is the data transformation chapter in R for data science.

```
library(dplyr)
```

```
starwars %>%  
  filter(species == "Droid")
```

```
## # A tibble: 6 x 14  
##   name    height  mass hair_color skin_color eye_color  
##   <chr>   <int> <dbl> <chr>      <chr>      <chr>  
## 1 C-3P0    167    75 <NA>      gold        yellow  
## 2 R2-D2     96    32 <NA>      white, blue red  
## 3 R5-D4     97    32 <NA>      white, red  red  
## 4 IG-88    200   140 none      metal       red  
## 5 R4-P17    96     NA none      silver, red red, blue  
## 6 BB8      NA     NA none      none        black  
## # ... with 8 more variables: birth_year <dbl>, sex <chr>,  
## #   gender <chr>, homeworld <chr>, species <chr>,  
## #   films <list>, vehicles <list>, starships <list>
```

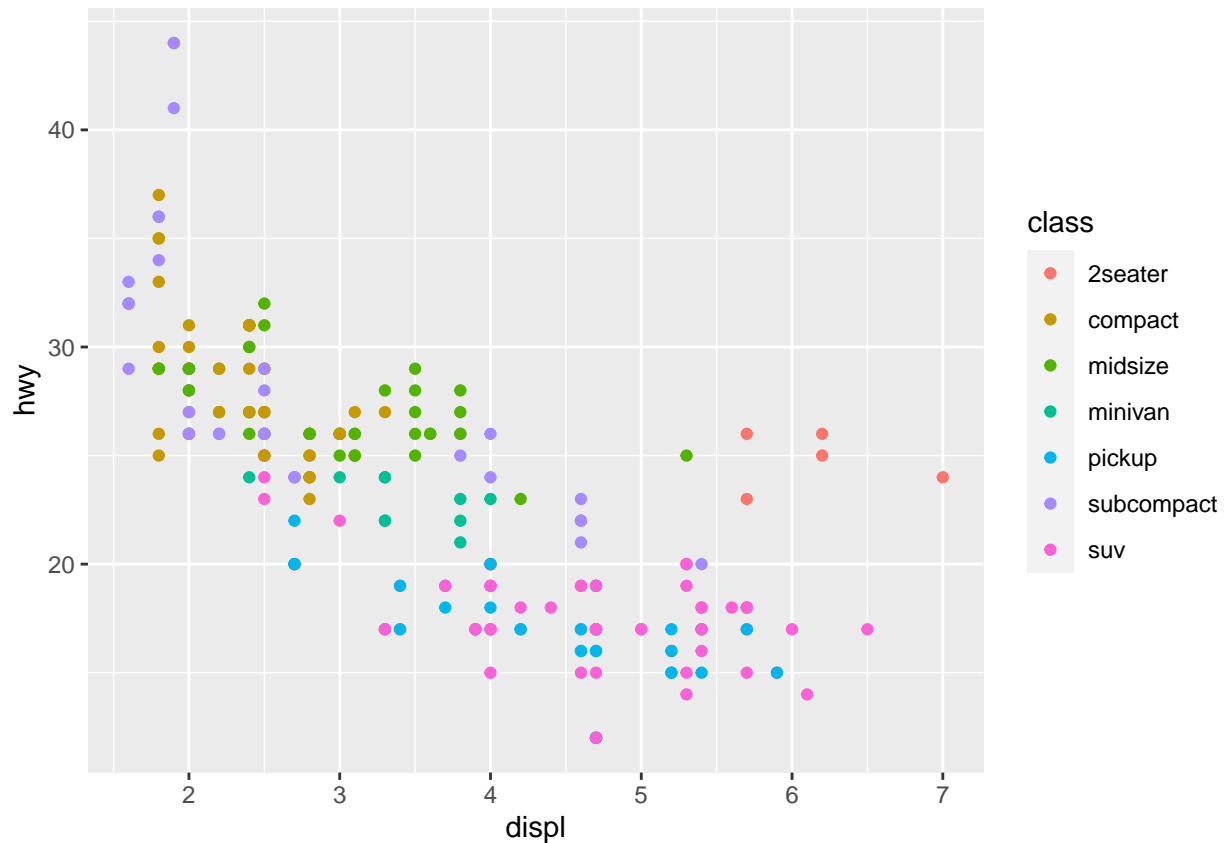
3.2 ggplot2



[ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics](#). You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

```
library(ggplot2)
```

```
ggplot(mpg, aes(displ, hwy, colour = class)) +  
  geom_point()
```

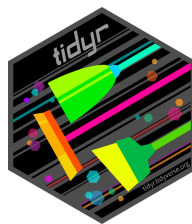


3.3 readr



The goal of [readr](#) is to provide a fast and friendly way to read rectangular data (like csv, tsv, and fwf). It is designed to flexibly parse many types of data found in the wild, while still cleanly failing when data unexpectedly changes. If you are new to readr, the best place to start is the data import chapter in R for data science.

3.4 tidyr



The goal of [tidyr](#) is to help you create tidy data. Tidy data is data where:

1. Every column is variable.

2. Every row is an observation.
3. Every cell is a single value.

Tidy data describes a standard way of storing data that is used wherever possible throughout the tidyverse. If you ensure that your data is tidy, you'll spend less time fighting with the tools and more time working on your analysis. Learn more about tidy data in `vignette("tidy-data")`.

```
library(tidyr)
```

```
relig_income
```

```
## # A tibble: 18 x 11
##   religion    `<$10k` ` $10-20k` ` $20-30k` ` $30-40k` ` $40-50k`
##   <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1 Agnostic      27      34      60      81      76
## 2 Atheist       12      27      37      52      35
## 3 Buddhist      27      21      30      34      33
## 4 Catholic     418     617     732     670     638
## 5 Don't know~   15      14      15      11      10
## 6 Evangelica~  575     869    1064     982     881
## 7 Hindu         1       9       7       9      11
## 8 Historical~  228     244     236     238     197
## 9 Jehovah's ~   20      27      24      24      21
## 10 Jewish       19      19      25      25      30
## 11 Mainline P~  289     495     619     655     651
## 12 Mormon       29      40      48      51      56
## 13 Muslim        6       7       9      10       9
## 14 Orthodox     13      17      23      32      32
## 15 Other Chri~   9       7      11      13      13
## 16 Other Fait~  20      33      40      46      49
## 17 Other Worl~   5       2       3       4       2
## 18 Unaffiliat~ 217     299     374     365     341
## # ... with 5 more variables: $50-75k <dbl>, $75-100k <dbl>,
## #   $100-150k <dbl>, >150k <dbl>, Don't know/refused <dbl>
```

```
relig_income %>%
```

```
  pivot_longer(-religion, names_to = "income", values_to = "frequency")
```

```
## # A tibble: 180 x 3
##   religion income      frequency
##   <chr>    <chr>      <dbl>
## 1 Agnostic <$10k          27
## 2 Agnostic $10-20k         34
## 3 Agnostic $20-30k         60
## 4 Agnostic $30-40k         81
## 5 Agnostic $40-50k         76
## 6 Agnostic $50-75k        137
## 7 Agnostic $75-100k       122
## 8 Agnostic $100-150k      109
## 9 Agnostic >150k          84
## 10 Agnostic Don't know/refused 96
## # ... with 170 more rows
```