

# forcats: working with categorical data

Emily Malcolm-White

The R package **forcats** is designed to make working with categorical variables easier and more efficient. It provides a set of functions that allow you to manipulate and analyze categorical data with ease. In this lesson, we'll cover the basics of the **forcats** package and some of its most useful functions.



## Categorical Variables

Let's review what categorical data is. Categorical data is a type of data that consists of categories or labels.

Examples of categorical data include:

- Colors (red, blue, green, etc.)
- Types of vehicles (sedan, SUV, truck)
- Educational degrees (high school, college, graduate school)

Categorical data can be further divided into two types: *nominal* and *ordinal*. Nominal data consists of categories that have no inherent order, while ordinal data consists of categories that have a natural order. For example, educational degrees are ordinal data because they can be ordered from least to most advanced.

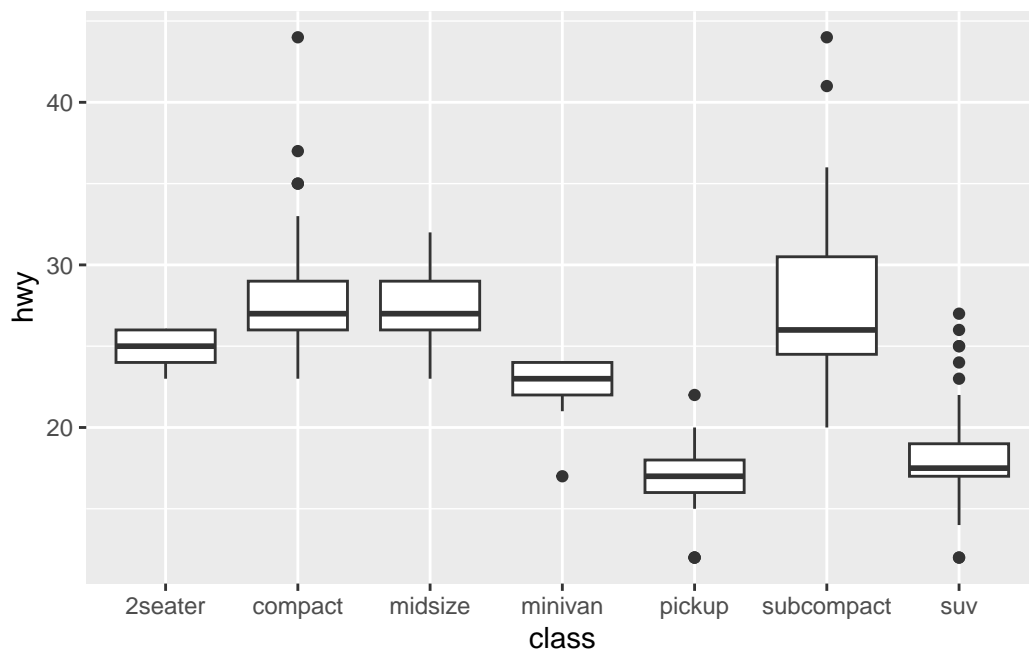
## mpg Data

We will play with different functions in the `forcats` packages using the `mpg` dataset from earlier in the semester.

```
library(forcats)
library(tidyverse)
data("mpg")
```

Recall our side-by-side boxplot:

```
mpg %>%
  ggplot(aes(x=class, y=hwy)) +
  geom_boxplot()
```



## Reordering Factor Levels

One of the most useful functions is `fct_relevel()`, which allows you to reorder the levels of a factor. This can be useful when you want to change the default ordering of the levels or when you want to group certain levels together.

Is `class` a factor?

```
mpg$class %>% is.factor()
```

```
[1] FALSE
```

Let's make it a factor!

```
mpg <- mpg %>%  
  mutate(class = class %>% as.factor())
```

Let's check the levels and their current ordering!

```
mpg$class %>%  
  levels()
```

```
[1] "2seater"    "compact"    "midsize"    "minivan"    "pickup"  
[6] "subcompact" "suv"
```

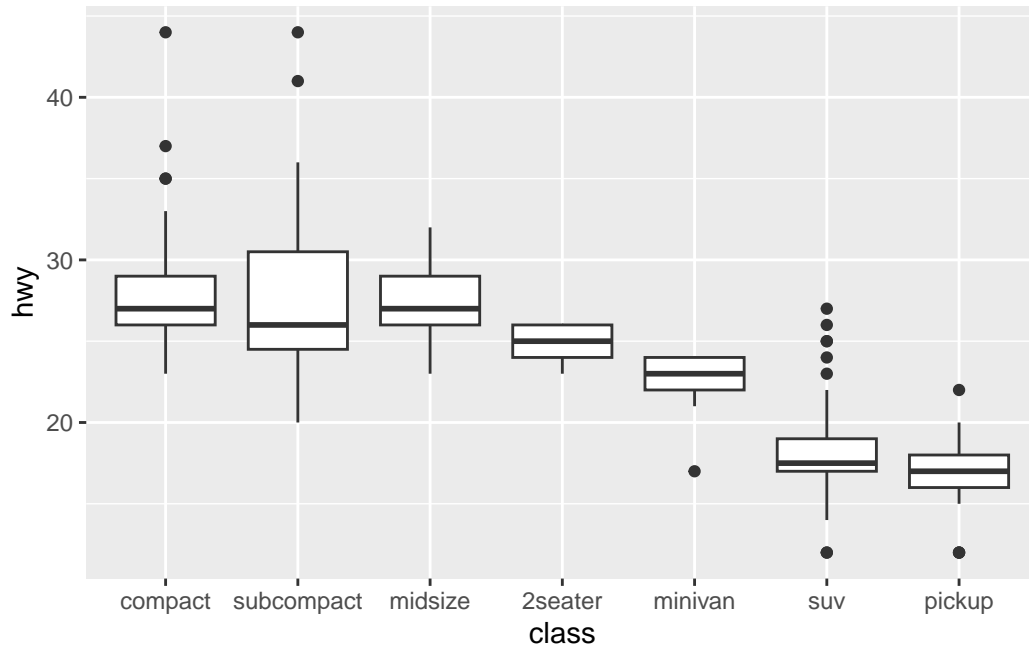
To reorder the levels with `fct_relevel()`

```
mpg <- mpg %>%  
  mutate(class = class %>% fct_relevel( "compact","subcompact","midsize","2seater","minivan",  
                                         "suv","pickup"))  
  
mpg$class %>%  
  levels()
```

```
[1] "compact"    "subcompact" "midsize"    "2seater"    "minivan"  
[6] "suv"        "pickup"
```

Let's recreate our side-by-side boxplot now:

```
mpg %>%  
  ggplot(aes(x=class, y=hwy)) +  
  geom_boxplot()
```



Rather than reordering them manually by typing the order, you could also re-level by some numeric criteria using `fct_reorder()`. For example:

```
mpg <- mpg %>%
  mutate(class = class %>% fct_reorder(hwy, median))

mpg$class %>%
  levels()
```

```
[1] "pickup"      "suv"         "minivan"     "2seater"     "subcompact"
[6] "compact"     "midsize"
```

## Renaming Factor levels with `fct_recode`

Sometimes you might not like the way the levels are named.

```
mpg <- mpg %>%
  mutate(class = class %>% fct_recode("two-seater" = "2seater"))

## NEW NAME = OLD NAME

mpg$class %>%
```

```
levels()
```

```
[1] "pickup"      "suv"          "minivan"      "two-seater"   "subcompact"  
[6] "compact"     "midsize"
```

## Factor Collapsing with `fct_collapse()`

Let's say we wanted to create only two categories – cars and larger vehicles.

```
mpg <- mpg %>%  
  mutate(class_two = class %>% fct_collapse( cars = c("compact", "subcompact", "midsize",  
  
mpg$class_two %>%  
  levels()
```

```
[1] "big"  "cars"
```

## Lumping into an other category

- `fct_lump_min()`: lumps levels that appear fewer than min times.
- `fct_lump_prop()`: lumps levels that appear in fewer than (or equal to)  $\text{prop} * n$  times.
- `fct_lump_n()` lumps all levels except for the n most frequent (or least frequent if  $n < 0$ )

```
mpg %>%  
  count(manufacturer)
```

```
# A tibble: 15 x 2  
  manufacturer      n  
    <chr>          <int>  
1 audi             18  
2 chevrolet        19  
3 dodge            37  
4 ford             25  
5 honda             9  
6 hyundai          14  
7 jeep             8  
8 land rover       4
```

9	lincoln	3
10	mercury	4
11	nissan	13
12	pontiac	5
13	subaru	14
14	toyota	34
15	volkswagen	27

Let's say we wanted only the manufacturers with at least 15 cars produced. Everything else we want to just be other:

```
mpg <- mpg %>%
  mutate(class_lumped = class %>% fct_lump_min(15))

mpg$manufacturer %>%
  levels()
```

NULL

Create a table using kableExtra:

```
library(kableExtra)

mpg %>%
  count(manufacturer) %>%
  kbl() %>%
  kable_styling()
```

manufacturer	n
audi	18
chevrolet	19
dodge	37
ford	25
honda	9
hyundai	14
jeep	8
land rover	4
lincoln	3
mercury	4
nissan	13
pontiac	5
subaru	14
toyota	34
volkswagen	27