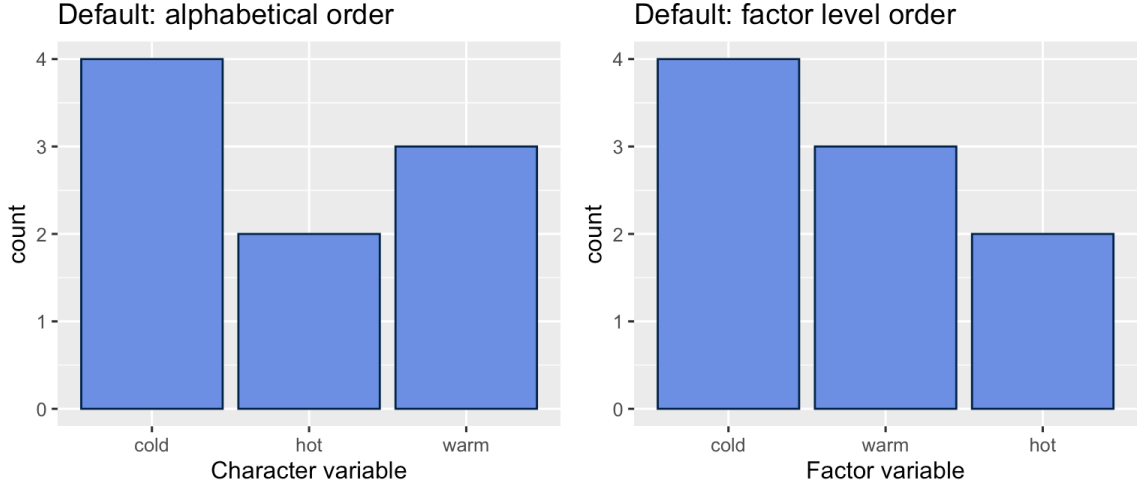
Useful forcats

Character vs factor data

character data: plotted alphabetically

factor data: plotted in order of factor levels

```
1  library(tidyverse)
2  df <- tibble(chardata = c("cold", "warm", "hot", "hot", "warm", "warm", "cold",
3                            "cold", "cold"),
4       factordata = factor(c("cold", "warm", "hot", "hot", "warm", "warm", "cold",
5                            "cold", "cold"), levels = c("cold", "warm", "hot")))
```



Summary of useful forcats functions

`fct_recode(x, ...)` – change names of levels

`fct_inorder(x)` – set level order of `x` to row order

`fct_relevel(x, ...)` – manually set the order of levels of `x`

`fct_reorder(x, y)` – reorder `x` by `y`

`fct_infreq(x)` – order the levels of `x` by decreasing frequency

`fct_rev(x)` – reverse the order of factor levels of `x`

Recoding factor levels

Not a good idea for recoding since the order of the factor levels must be matched. This is INCORRECT:

```
df <- data.frame(name = factor(c("STAT-UN-1201", "STAT-GR-5702", "STAT-GR-5293")),
                 enrollment = c(86, 172, 12))
df2 <- df
levels(df2$name) = c("Intro", "EDAV", "IMLV")
df2
```

```
   name enrollment
1  IMLV         86
2  EDAV        172
3 Intro         12
```

(Only use `levels()` to see the current levels.)

Recoding factor levels: `fct_recode()`

A better approach:

```
df$name <- fct_recode(df$name, Intro = "STAT-UN-1201", EDAV = "STAT-GR-5702",
                      IMLV = "STAT-GR-5293")
df
```

```
  name enrollment
1 Intro         86
2  EDAV        172
3  IMLV         12
```

Set factor level order to row order

```
1 x <- factor(c("Jack", "Queen", "King", "Ace"))
2 levels(x)
```

```
[1] "Ace"   "Jack"  "King"  "Queen"
```

```
1 fct_inorder(x)
```

```
[1] Jack  Queen King  Ace
Levels: Jack Queen King Ace
```

Using `fct_relevel()` to move levels to the beginning

```
1 x <- c("A", "B", "C", "move1", "D", "E", "move2", "F")
2
3 fct_relevel(x, "move1", "move2")
```

```
[1] A     B     C     move1 D     E     move2 F
Levels: move1 move2 A B C D E F
```

Using `fct_relevel()` to move levels after an item (by position)

```
1 x <- c("A", "B", "C", "move1", "D", "E", "move2", "F")
2
3 fct_relevel(x, "move1", "move2", after = 4) # move after the fourth item
```

```
[1] A     B     C     move1 D     E     move2 F
Levels: A B C D move1 move2 E F
```

Using `fct_relevel()` to move levels to the end

```
1 x <- c("A", "B", "C", "move1", "D", "E", "move2", "F")
2
3 fct_relevel(x, "move1", "move2", after = Inf)
```

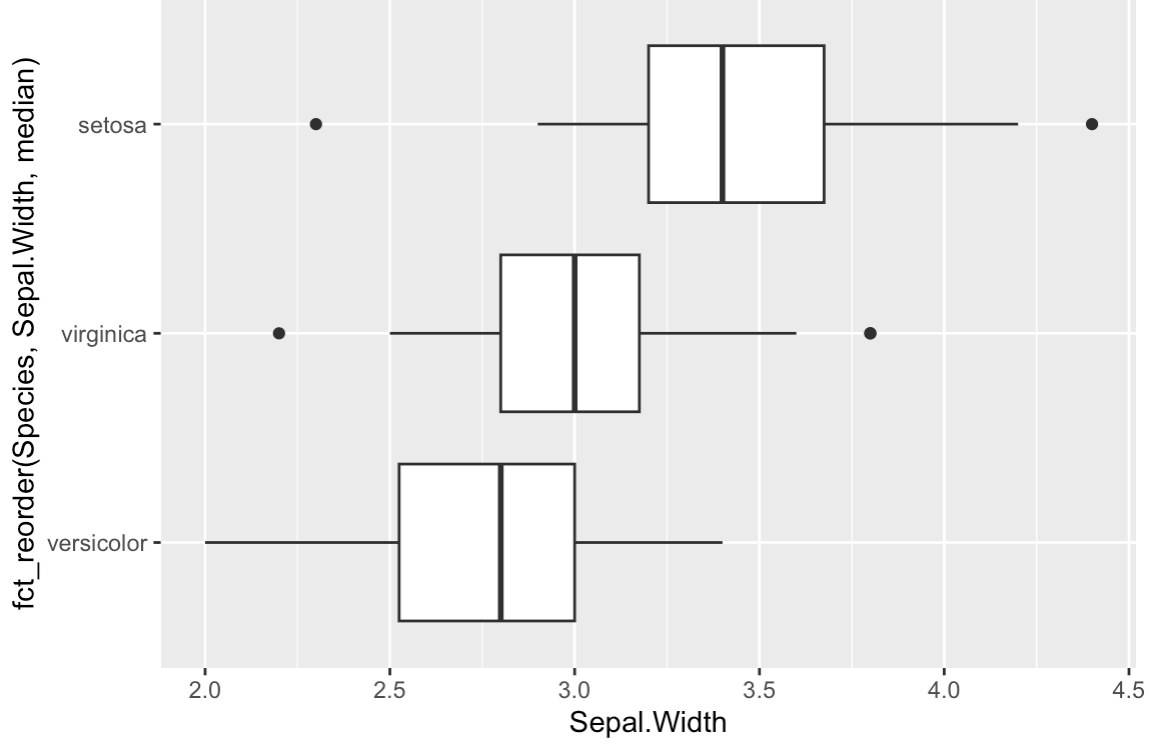
```
[1] A     B     C     move1 D     E     move2 F
Levels: A B C D E F move1 move2
```

Set factor level to the order of another variable

```
levels(iris$Species)
```

```
[1] "setosa"     "versicolor" "virginica"
```

```
ggplot(iris, aes(x = Sepal.Width, y = fct_reorder(Species, Sepal.Width, median))) +
geom_boxplot()
```



Order factor levels by reverse frequency count

```
1 x <- factor(c("Jack", "Queen", "King", "Ace", "Queen", "King", "King"))
2 levels(x)
```

```
[1] "Ace"   "Jack"  "King"  "Queen"
```

```
1 fct_infreq(x)
```

```
[1] Jack  Queen King  Ace   Queen King  King
Levels: King Queen Ace Jack
```

Binned data

```r
df <- data.frame(quarter = factor(c("Q1", "Q2", "Q3", "Q4")),
                 sales = c(213, 125, 421, 315))
df
```

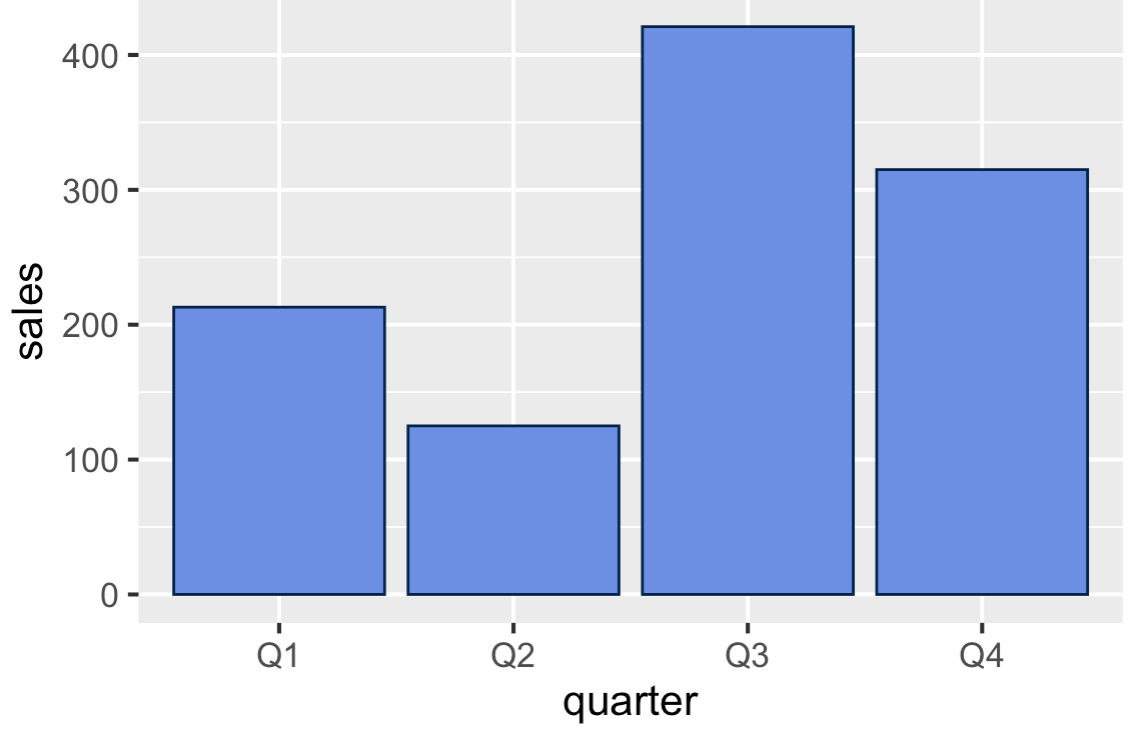
```
  quarter sales
1      Q1   213
2      Q2   125
3      Q3   421
4      Q4   315
```

```r
levels(df$quarter)
```

```
[1] "Q1" "Q2" "Q3" "Q4"
```

Binned, ordinal data, correct level order

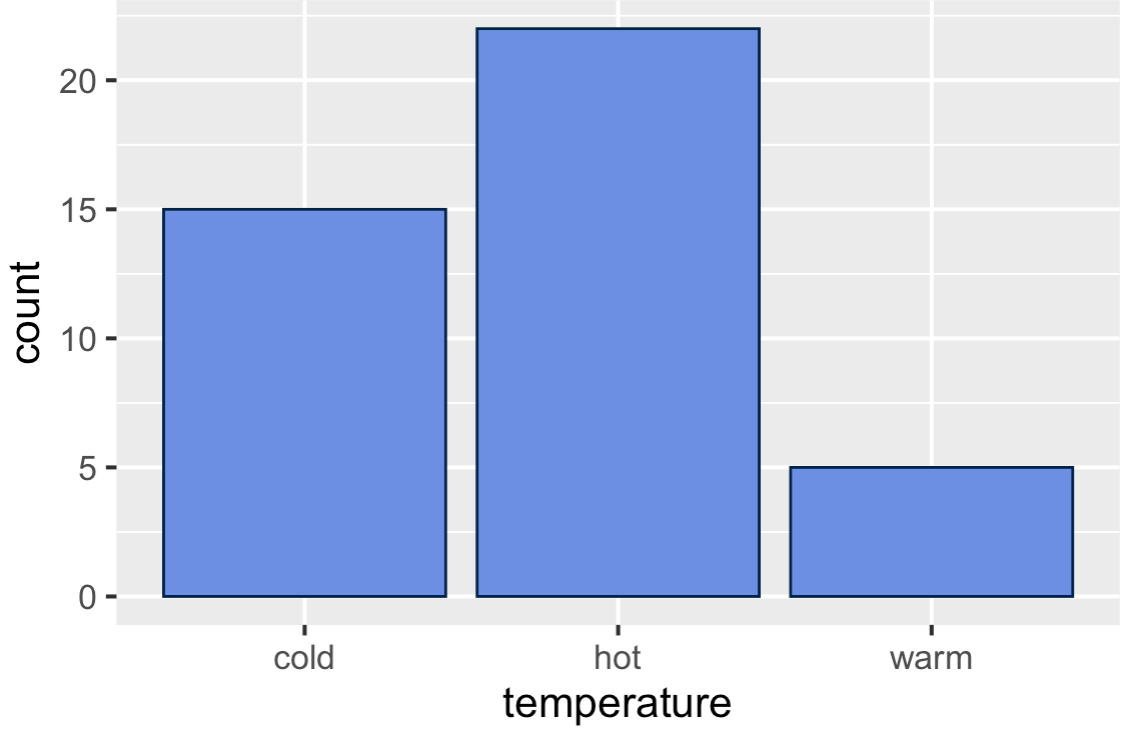
```
1  mycolor <- "#002448"; myfill = "#7192E3"
2  ggplot(df, aes(x = quarter, y = sales)) +
3    geom_col(color = mycolor, fill= myfill) +
4    theme_grey(16)
```



Ex. 1

INCORRECT

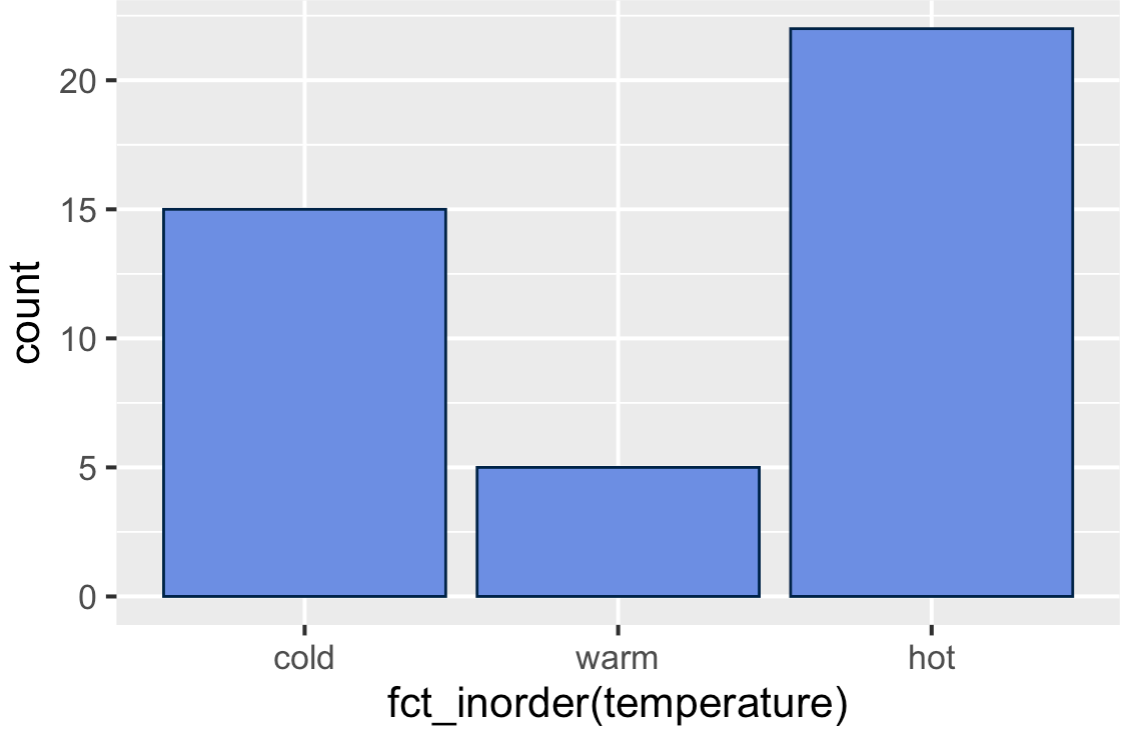
</>



Ex. 1 solution

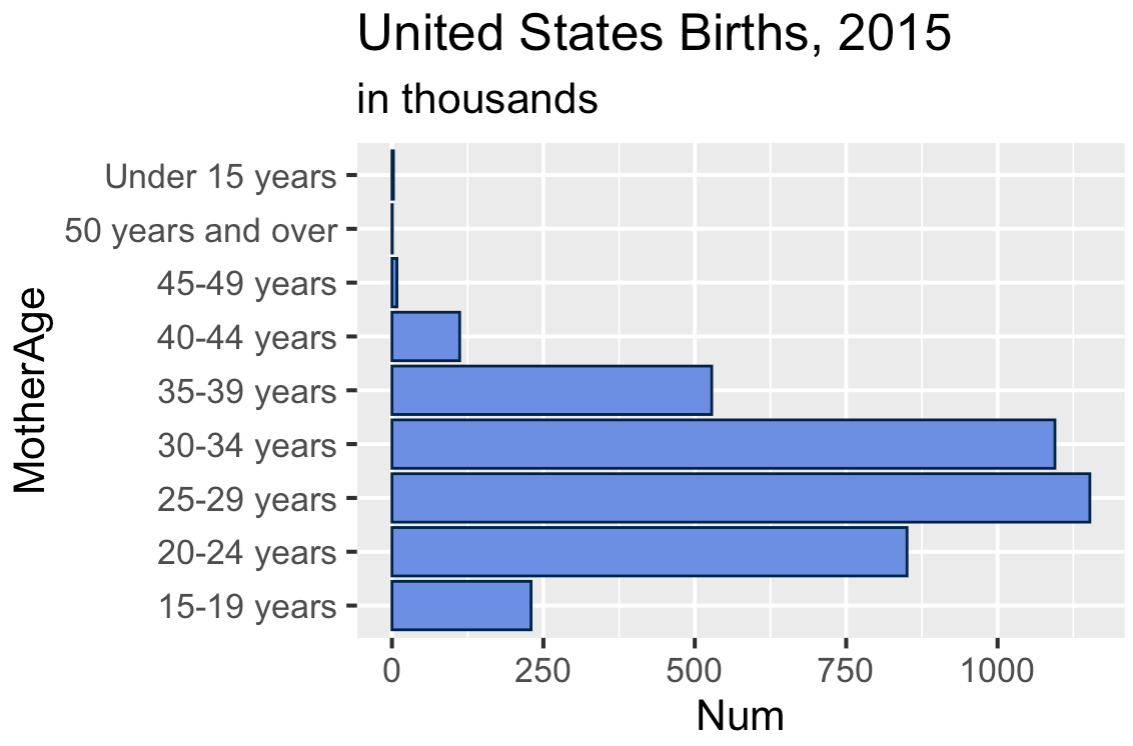
Binned, ordinal data, levels out of order

</>



Ex. 2

INCORRECT

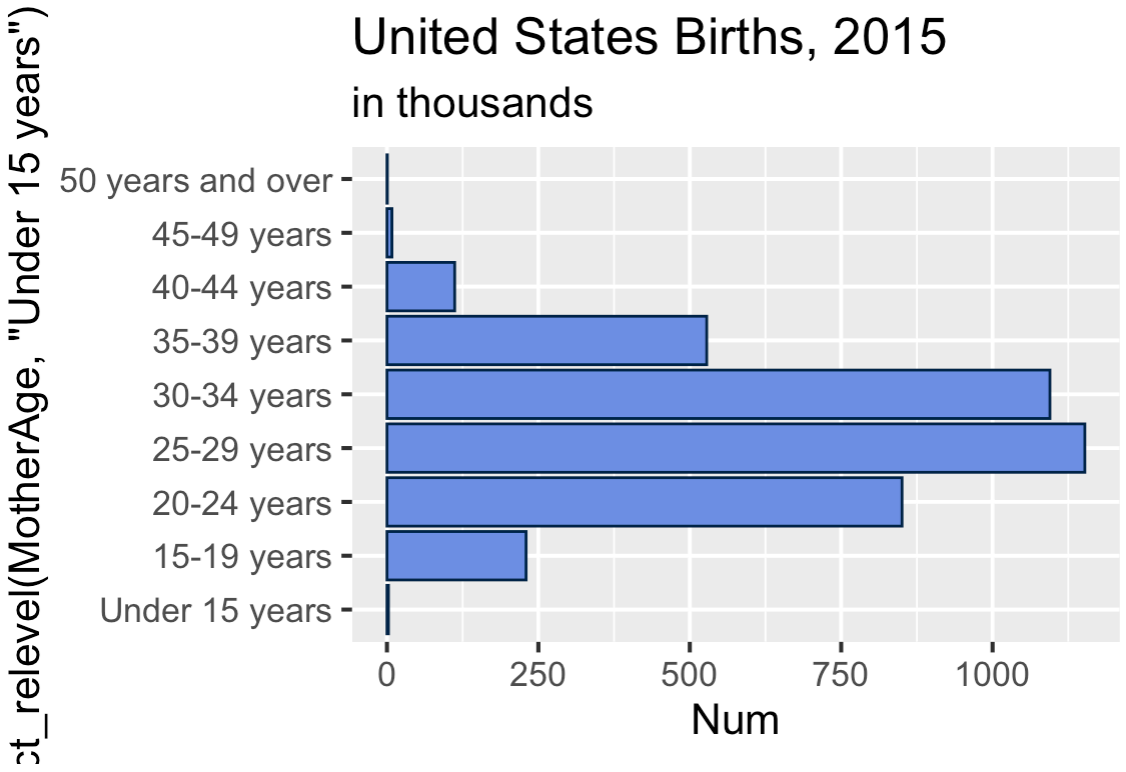


Ex. 2 solution

Binned, ordinal data, levels out of order

`fct_relevel()` can be used to set the correct order

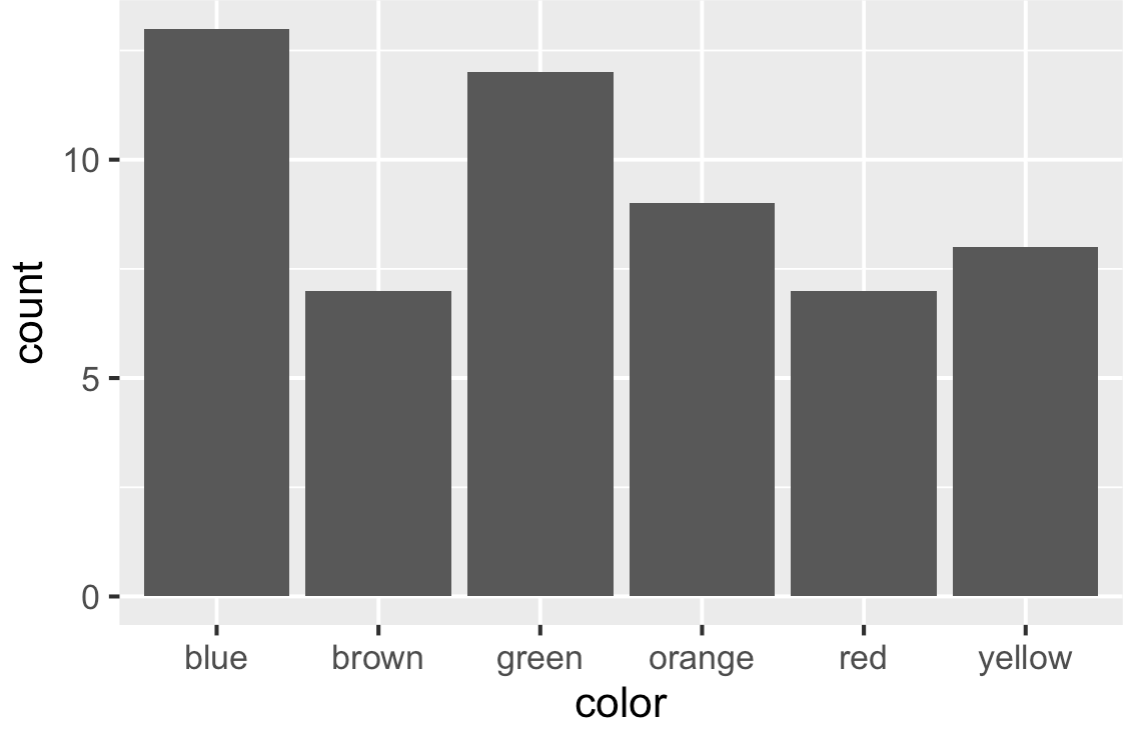
</>


United States Births, 2015
in thousands
ct_relevel(MotherAge, "Under 15 years")
50 years and over
45-49 years
40-44 years
35-39 years
30-34 years
25-29 years
20-24 years
15-19 years
Under 15 years
0
250
500
750
1000
Num

Ex. 3

INCORRECT

</>

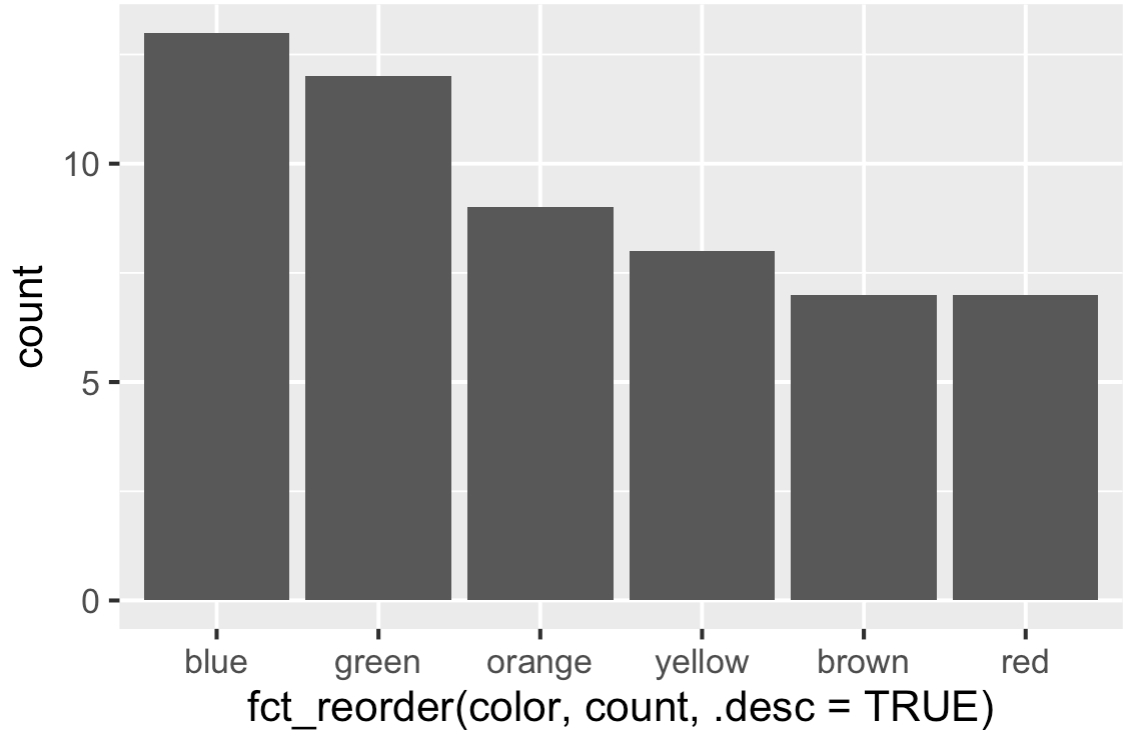


Ex. 3 solution

Binned, nominal, vertical bars

Order bars by frequency count using `fct_reorder()` (or `reorder()`)

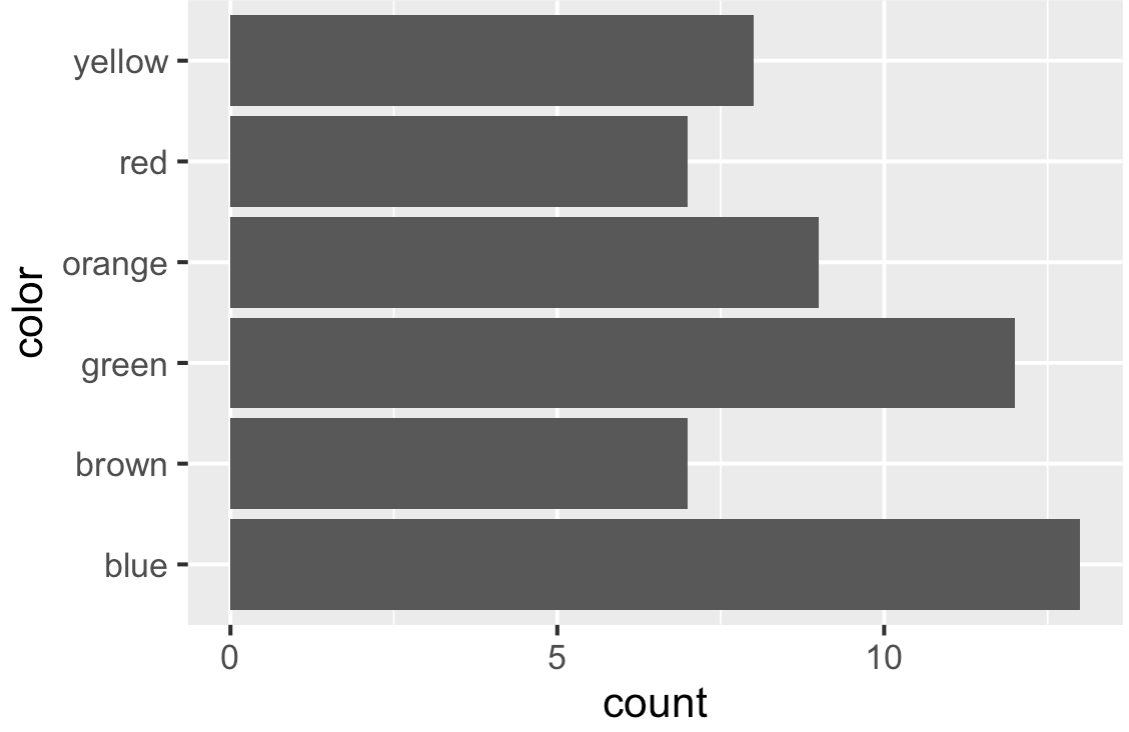
</>


count
10
5
0
blue
green
orange
yellow
brown
red
fct_reorder(color, count, .desc = TRUE)

Ex. 4

INCORRECT

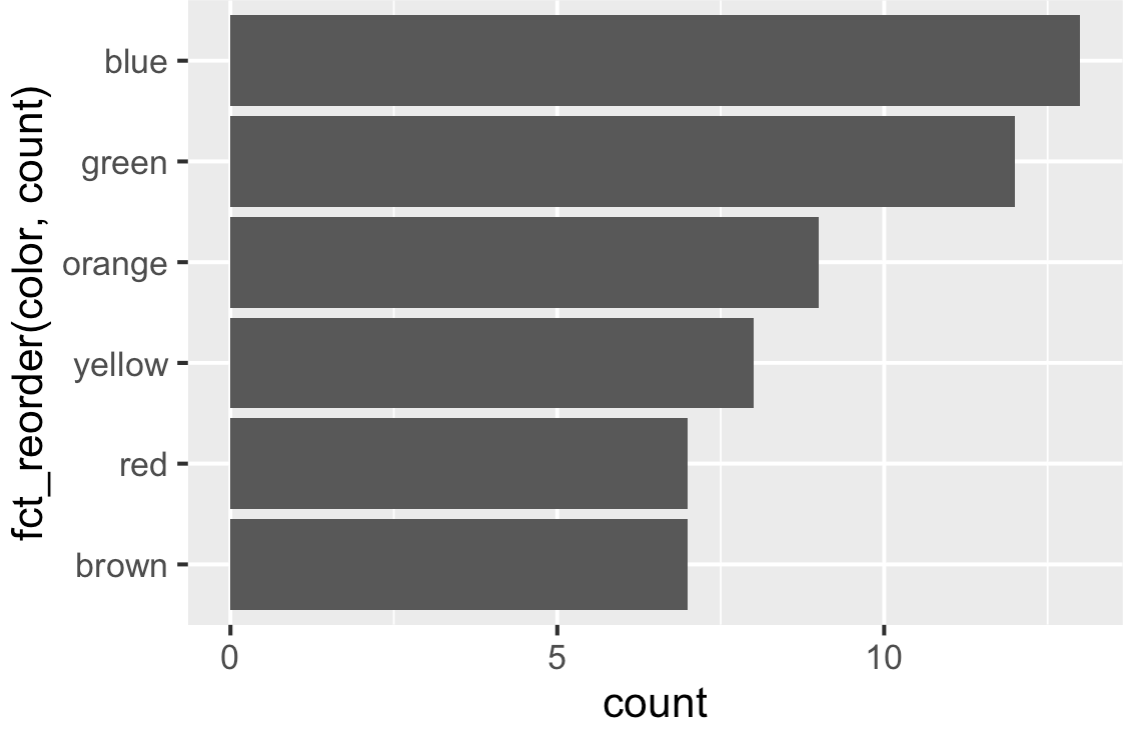
</>



Ex. 4 solution

Binned, nominal (horizontal bars)

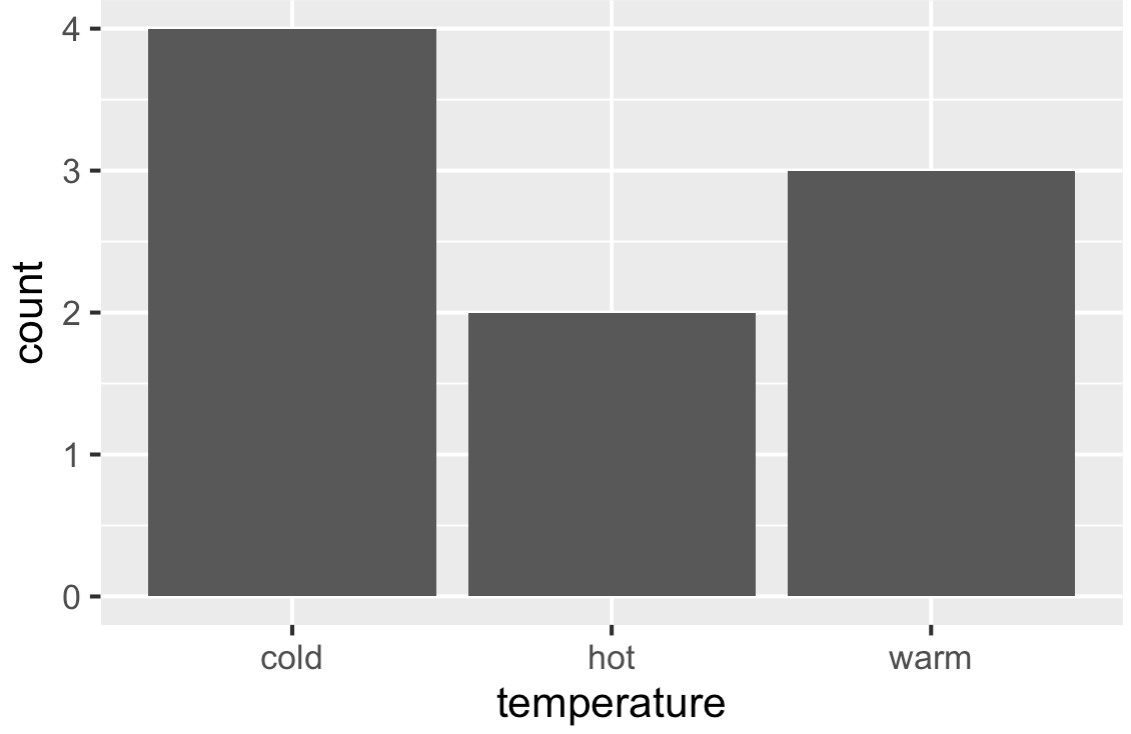
</>



Ex. 5

INCORRECT

</>

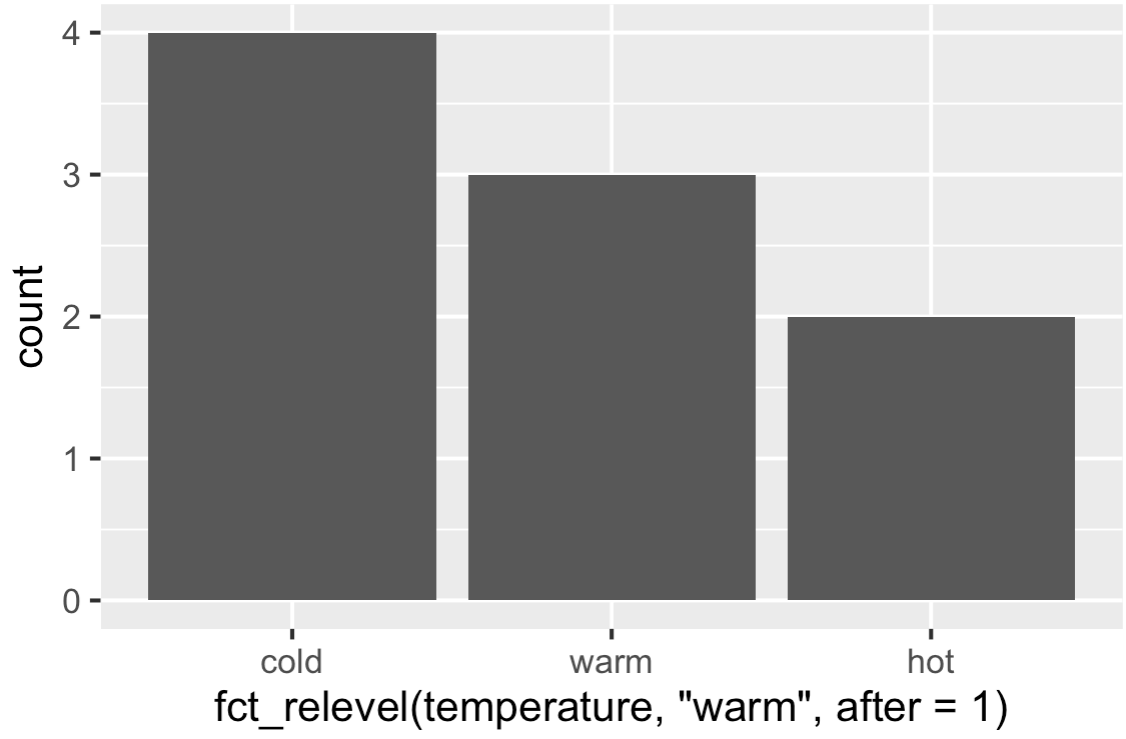


Ex. 5 solution

Unbinned, ordinal, levels out of order

Use `fct_relevel()` (as with binned, ordinal data)

</>


4
3
2
1
0
count
cold
warm
hot
fct_relevel(temperature, "warm", after = 1)

M & M data

</>

```
[1] 100   1
```

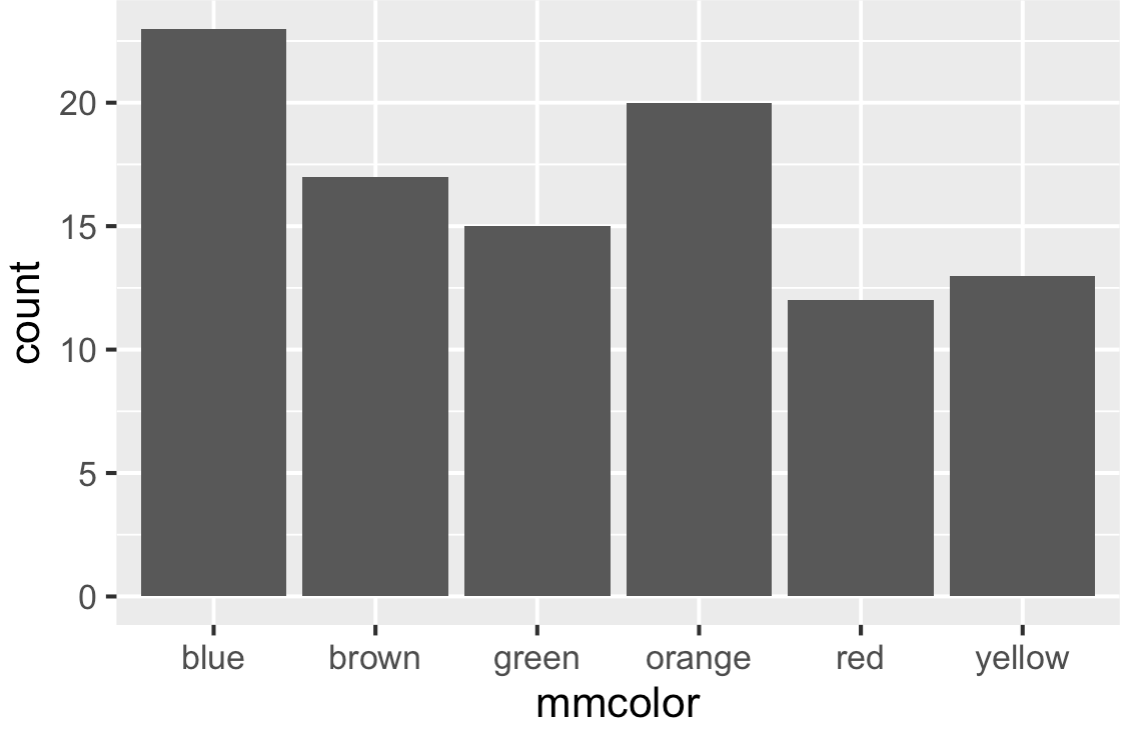
</>

```
    mmcolor
1     brown
2    yellow
3     brown
4    orange
5       red
6    yellow
7     brown
8       red
9     green
10   orange
```

Ex. 6

INCORRECT

</>



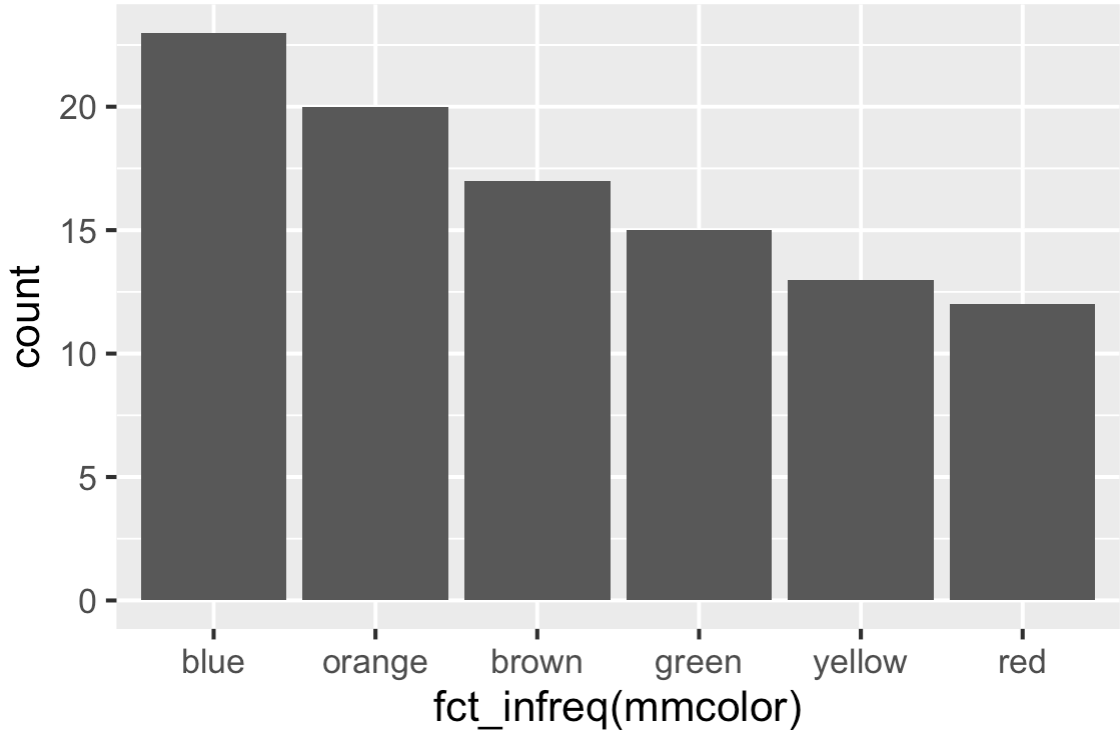
Ex. 6 solution

Unbinned, nominal data

`fct_infreq()` (default is decreasing order of frequency)

Vertical bars:

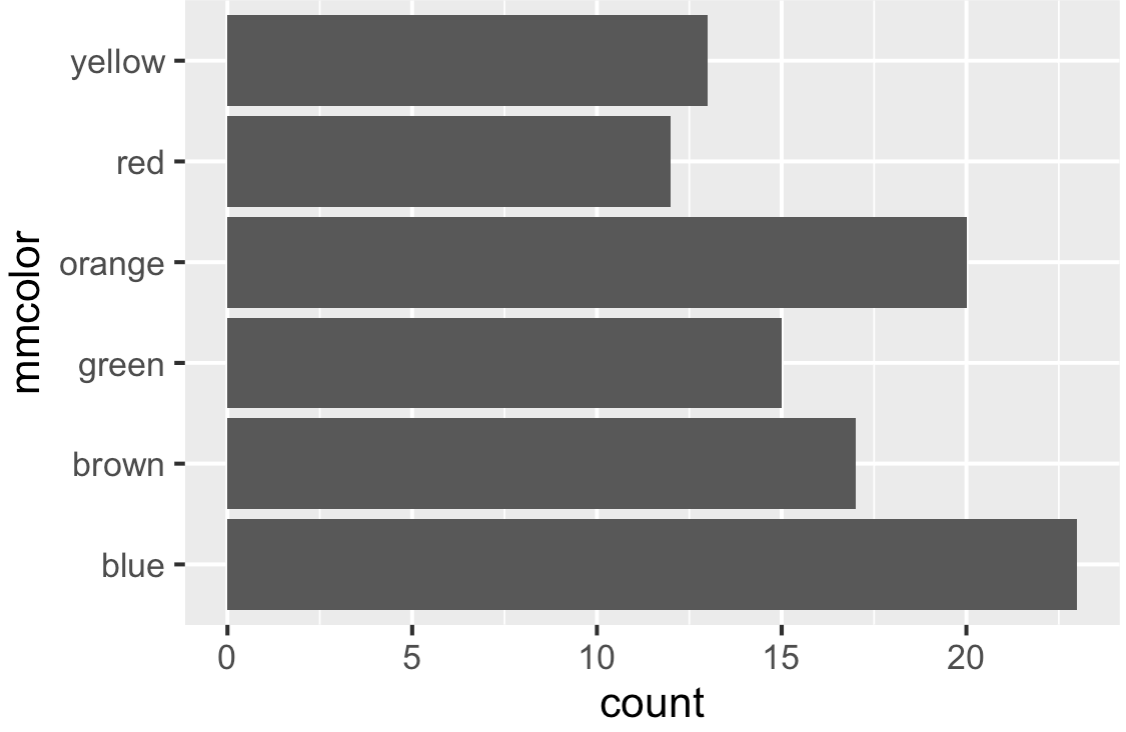
</>


count
20
15
10
5
0
blue
orange
brown
green
yellow
red
fct_infreq(mmcolor)

Ex. 7

INCORRECT

</>



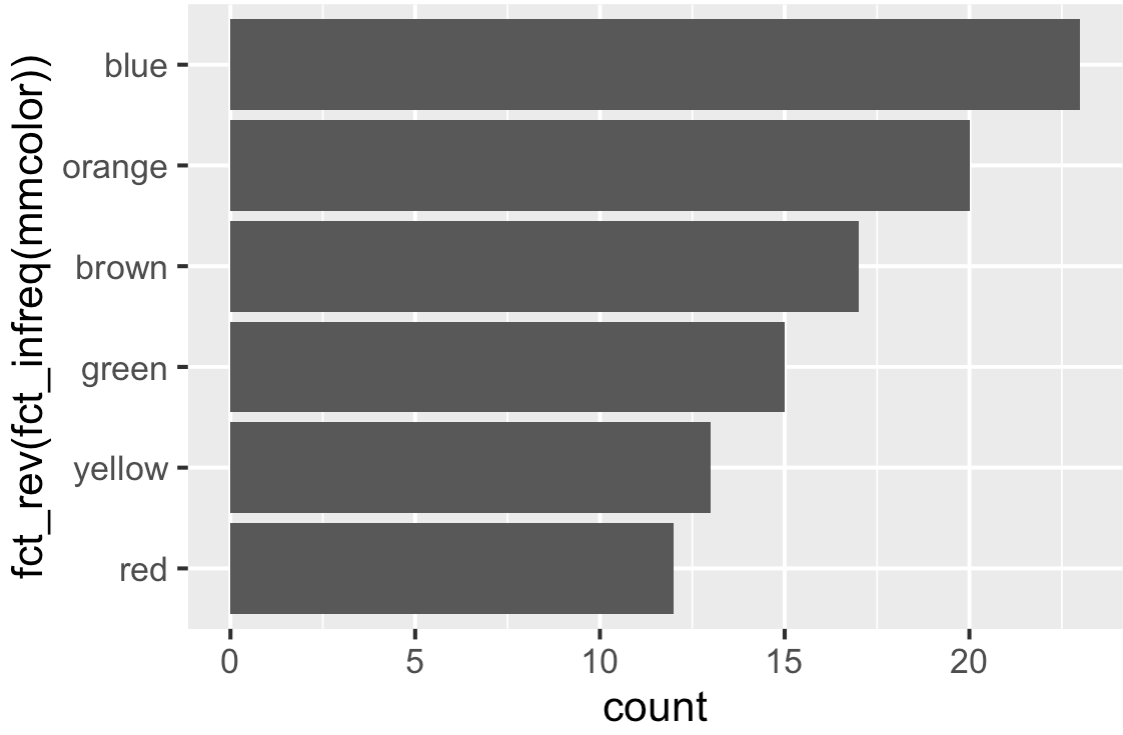
Ex. 7 solution

Unbinned, nominal data

```
fct_rev(fct_infreq())
```

Horizontal bars:

</>


fct_rev(fct_infreq(mmcolor))
blue
orange
brown
green
yellow
red
0
5
10
15
20
count