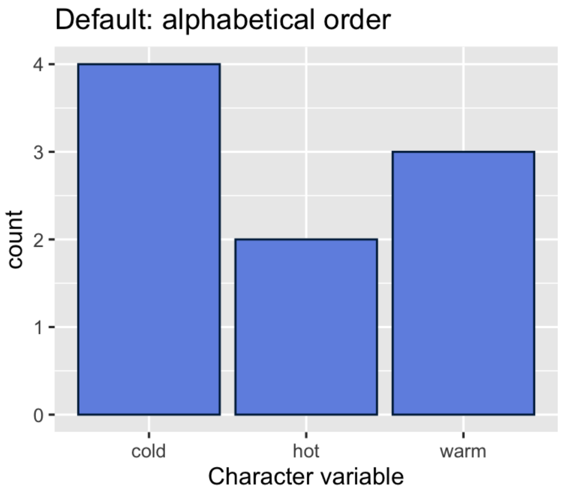
Useful forcats

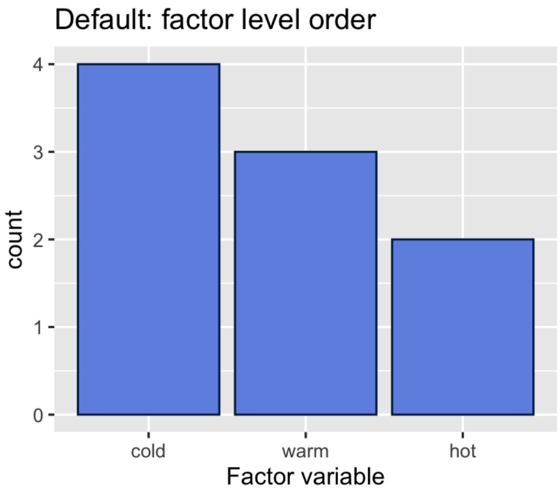
Character vs factor data

character data: plotted alphabetically

factor data: plotted in order of factor levels

```r
library(tidyverse)
df <- tibble(chardata = c("cold", "warm", "hot", "hot", "warm", "warm", "cold",
                          "cold", "cold"),
    factordata = factor(c("cold", "warm", "hot", "hot", "warm", "warm", "cold",
                          "cold", "cold"), levels = c("cold", "warm", "hot")))
```


Default: alphabetical order


Default: factor level order

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:

```r
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:

```r
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

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

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

```r
fct_inorder(x)
```

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

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

```r
x <- c("A", "B", "C", "move1", "D", "E", "move2", "F")

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)

```r
x <- c("A", "B", "C", "move1", "D", "E", "move2", "F")

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

```r
x <- c("A", "B", "C", "move1", "D", "E", "move2", "F")

fct_relevel(x, "move1", "move2", after = Inf)
```

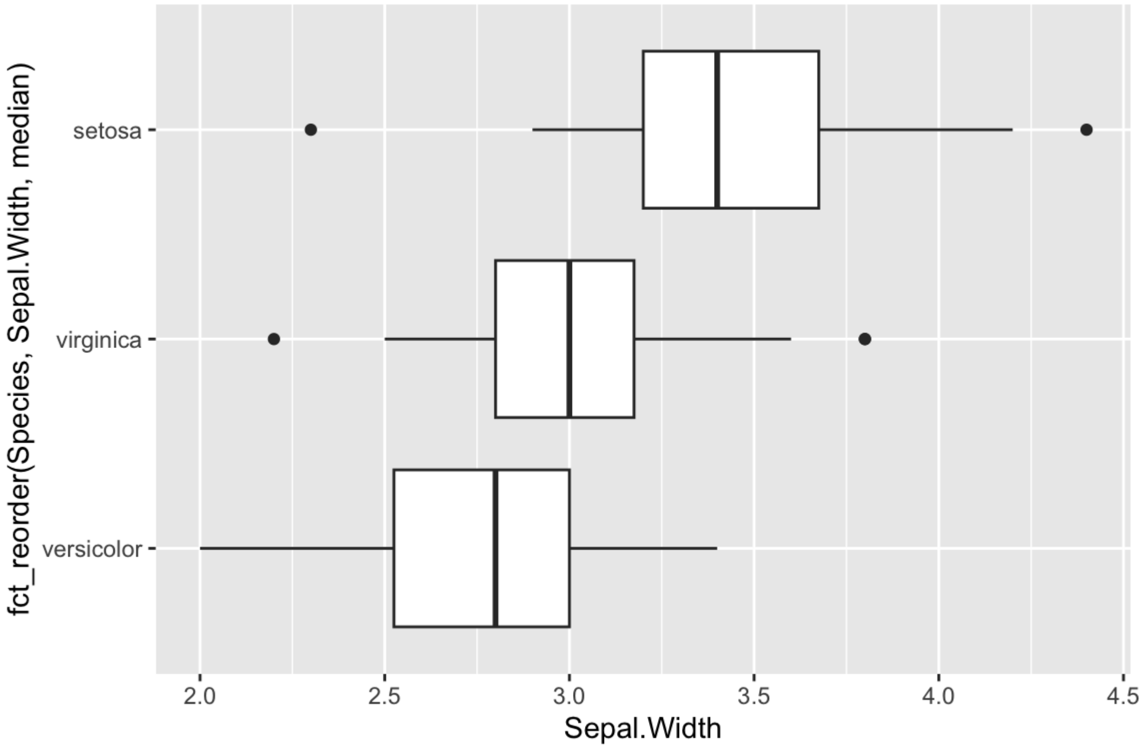
```
[1] A     B     C     move1 D     E     move2 F    
Levels: A B C D E F move1 move2
```

Set factor level to the order of another variable

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

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

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



Order factor levels by reverse frequency count

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

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

```r
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
```

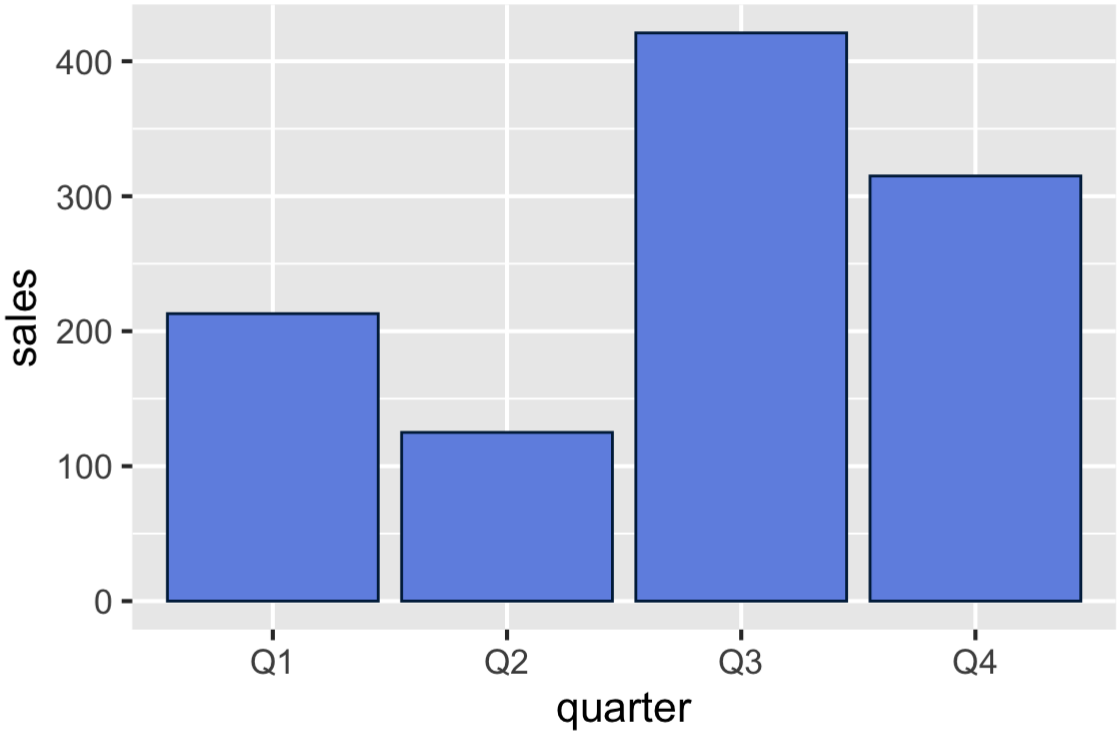
```
  quarter sales
1      Q1   213
2      Q2   125
3      Q3   421
4      Q4   315
```

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

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

Binned, ordinal data, correct level order

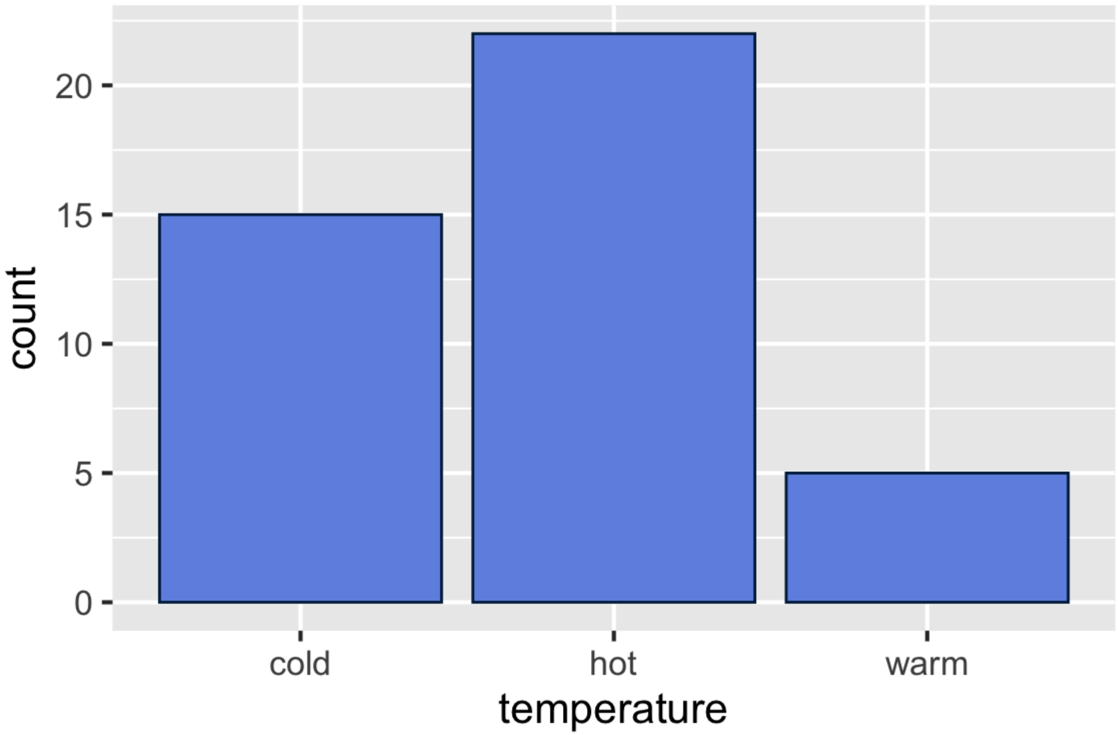
```r
mycolor <- "#002448"; myfill = "#7192E3"
ggplot(df, aes(x = quarter, y = sales)) +
  geom_col(color = mycolor, fill= myfill) +
  theme_grey(16)
```



Binned, ordinal data, levels out of order

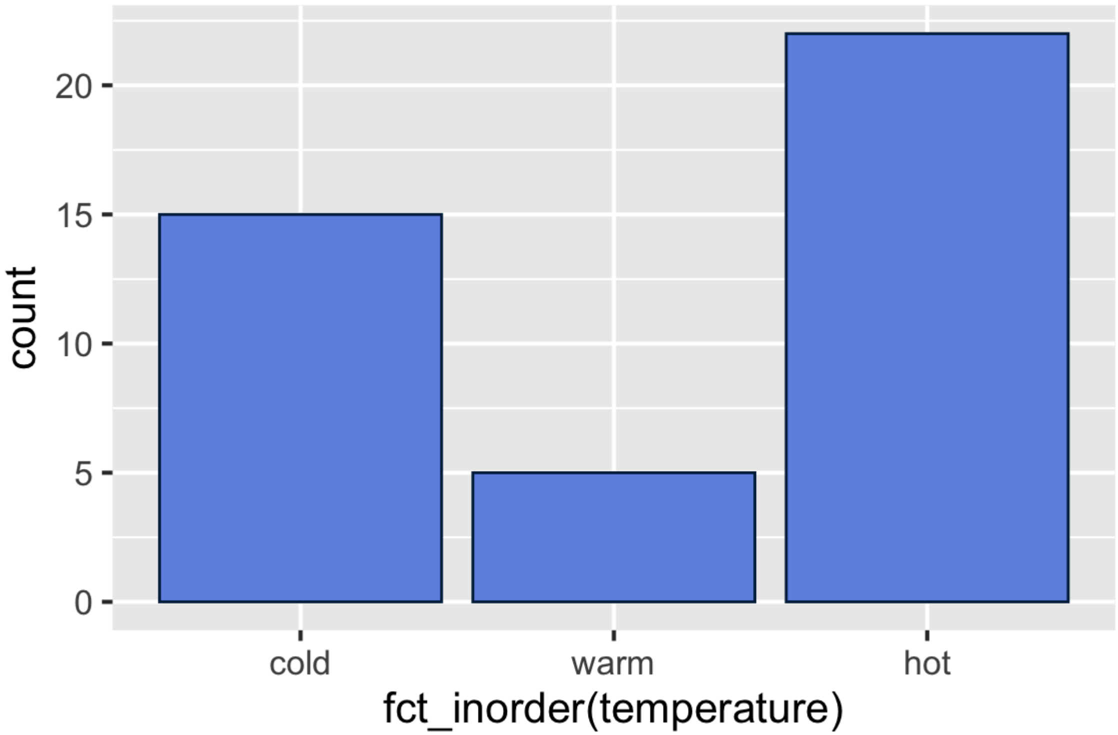
INCORRECT

```r
df <- data.frame(temperature = factor(c("cold", "warm", "hot")),
                 count = c(15, 5, 22))

ggplot(df, aes(x = temperature, y = count)) +
  geom_col(color = mycolor, fill = myfill) +
  theme_grey(16)
```


20
15
10
5
0
count
cold
hot
warm
temperature

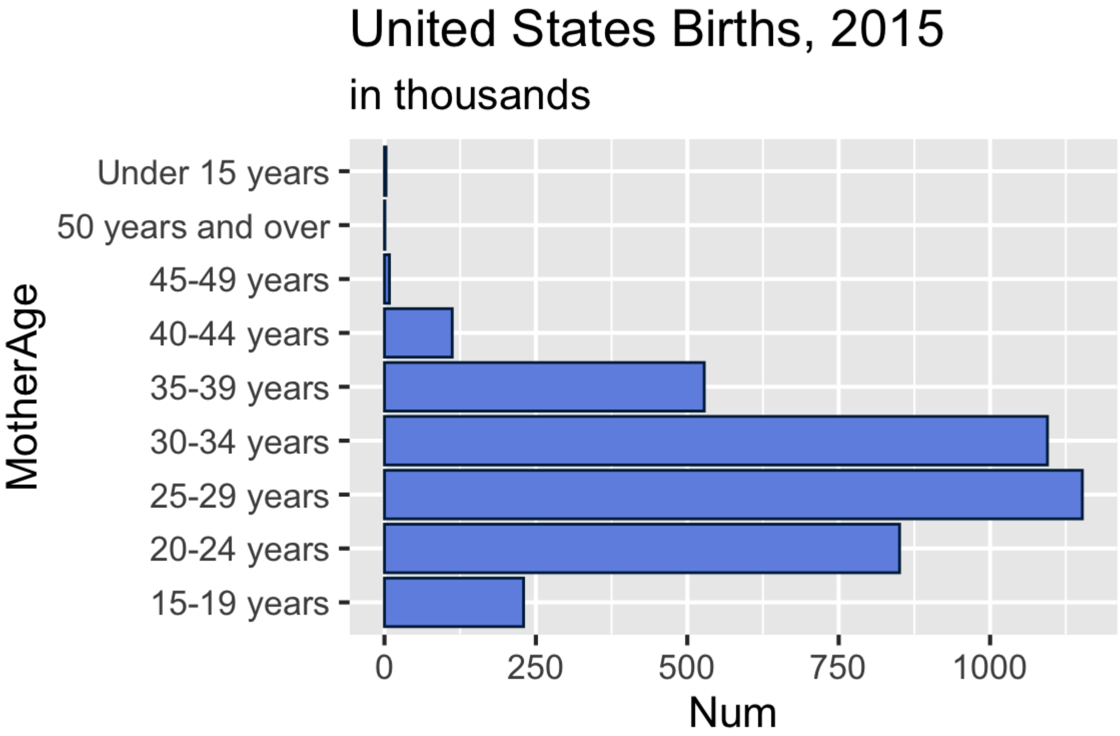
Binned, ordinal data, levels out of order

```r
# SOLUTION
df <- data.frame(temperature = factor(c("cold", "warm", "hot")),
                 count = c(15, 5, 22))

# row order is correct (think: factor in ROW order)
ggplot(df, aes(x = fct_inorder(temperature), y = count)) +
  geom_col(color = mycolor, fill = myfill) +
  theme_grey(16)
```


20
15
10
5
0
count
cold
warm
hot
fct_inorder(temperature)

Binned, ordinal data, levels out of order

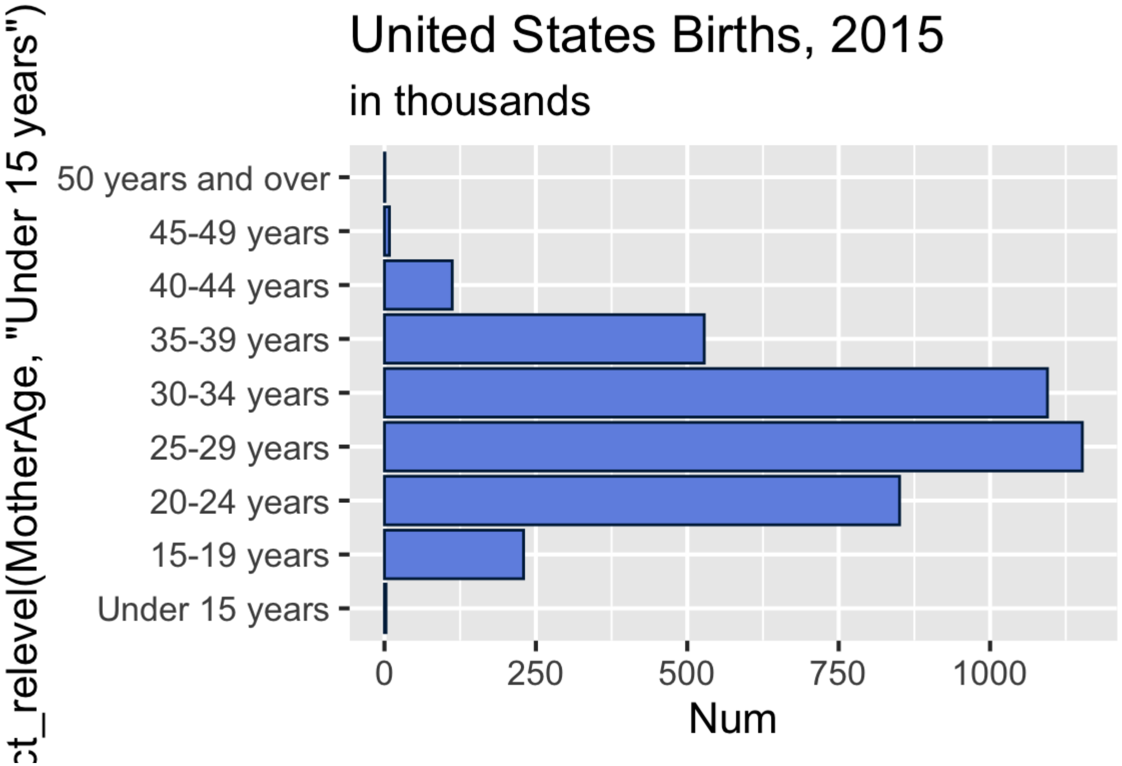
INCORRECT



Binned, ordinal data, levels out of order

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

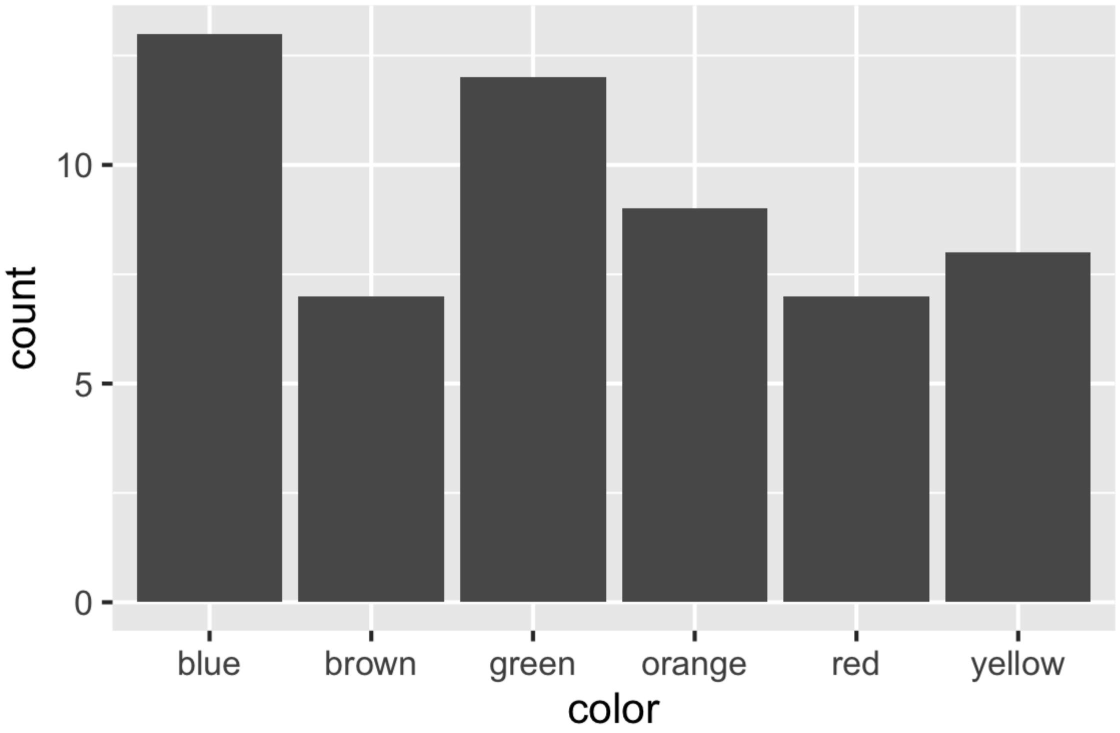
```r
# SOLUTION
ggplot(Births2015, aes(x = Num, y = fct_relevel(MotherAge, "Under 15 years"))) +
  ggtitle("United States Births, 2015", subtitle = "in thousands") +
  scale_x_continuous(breaks = seq(0, 1250, 250)) +
  geom_col(color = mycolor, fill = myfill) +
  theme_grey(16)
```


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

Binned, nominal, vertical bars

INCORRECT

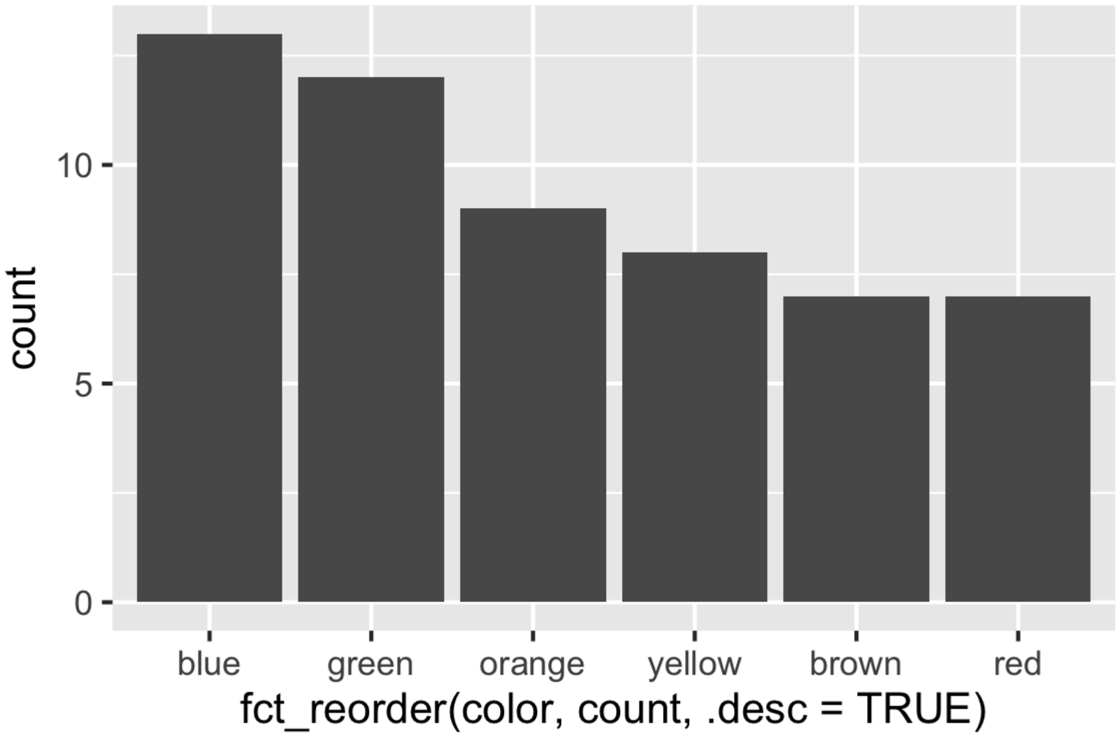
```r
pack1 <- data.frame(color = c("blue", "brown", "green", "orange", "red", "yellow"),
  count = c(13, 7, 12, 9, 7, 8))

ggplot(pack1, aes(x = color, y = count)) +
  geom_col() +
  theme_grey(16)
```


count
10
5
0
blue
brown
green
orange
red
yellow
color

Binned, nominal, vertical bars

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

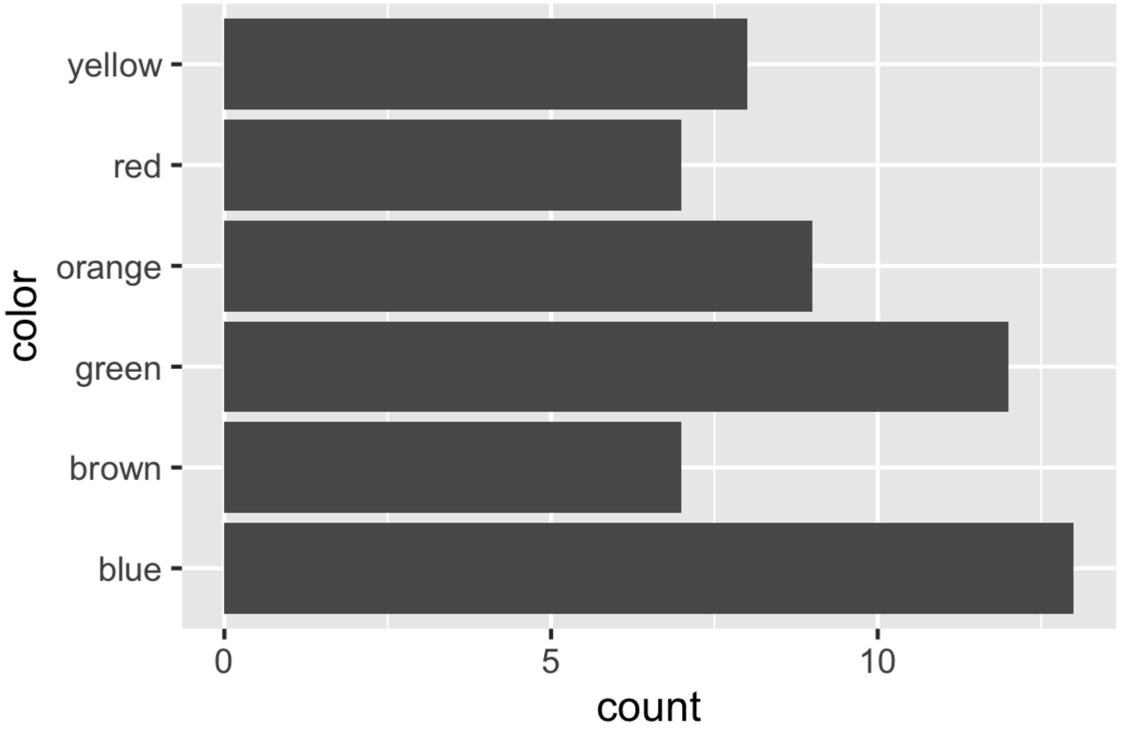
```r
# SOLUTION
ggplot(pack1, aes(x = fct_reorder(color, count, .desc = TRUE), y = count)) +
  geom_col() +
  theme_grey(16)
```


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

Binned, nominal (horizontal bars)

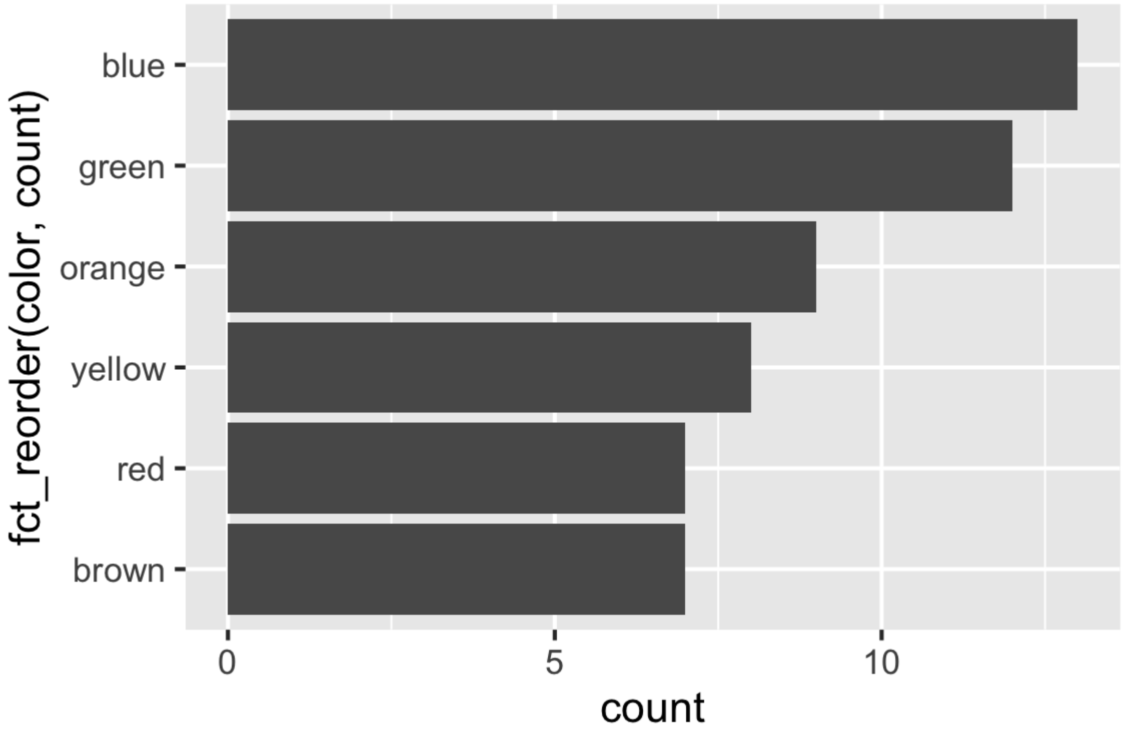
INCORRECT

```r
ggplot(pack1, aes(x = count, y = color)) +
  geom_col() +
  theme_grey(16)
```



Binned, nominal (horizontal bars)

```r
# SOLUTION
ggplot(pack1, aes(x = count, y = fct_reorder(color, count))) +
  geom_col() +
  theme_grey(16)
```



Unbinned, ordinal, correct level order

```r
# data available here: https://github.com/jtr13/data
student <- read.csv("student_data.csv") # or use readr::read_csv()
glimpse(student)
```

```
Rows: 44
Columns: 3
$ School      <chr> "CC  ", "CC  ", "CC  ", "CC  ", "CC  ", "GS  ", "CC  ", "C…
$ Level       <chr> "U01", "U01", "U01", "U01", "U01", "U03", "U01", "U01", "U…
$ Affiliation <chr> "CCUNDC", "CCUNDC", "CCUNDC", "CCUNDC", "CCUNDC", "GSUNDC"…
```

```r
levels(student$Level)
```

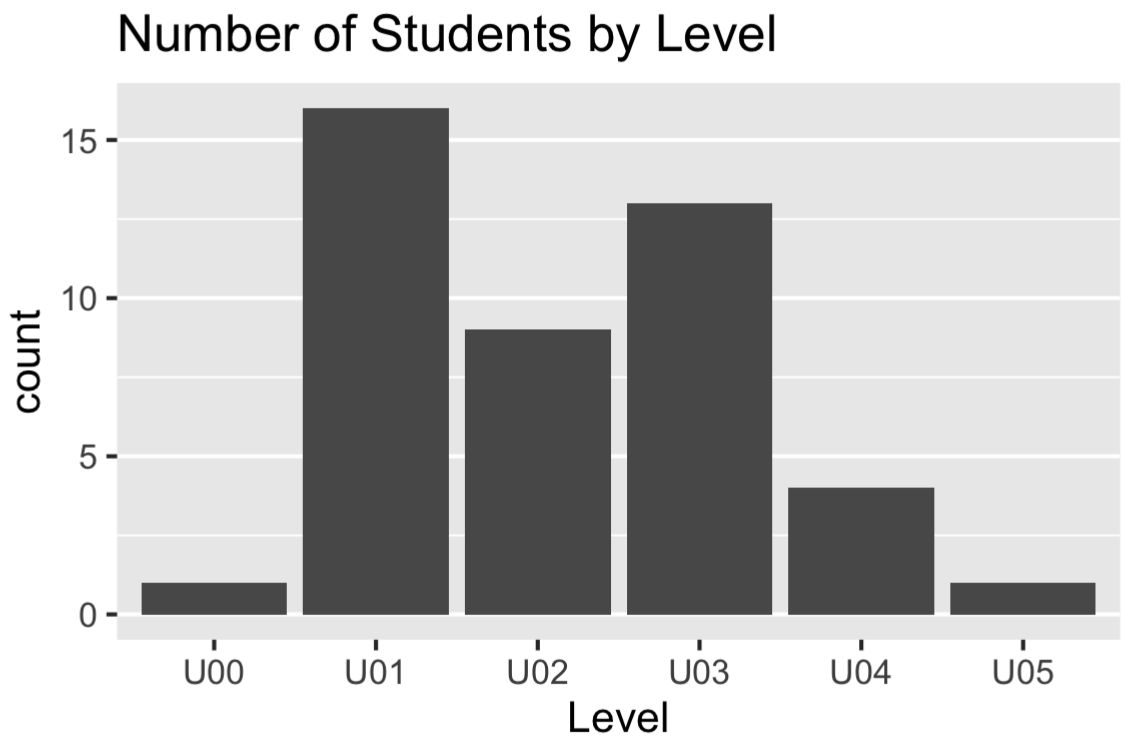
```
NULL
```

```r
levels(factor(student$Level))
```

```
[1] "U00" "U01" "U02" "U03" "U04" "U05"
```

Unbinned, ordinal, correct level order

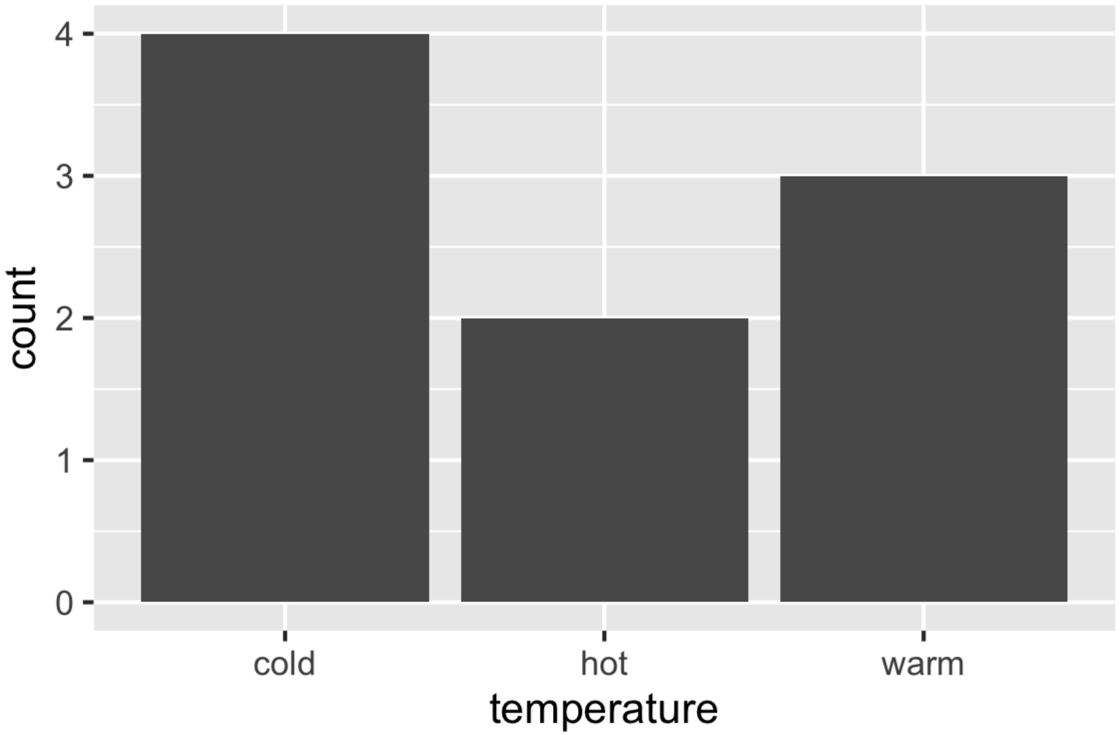
```r
ggplot(student, aes(Level)) +
  geom_bar()  +
  ggtitle("Number of Students by Level") +
  theme_grey(16) +
  theme(panel.grid.major.x = element_blank())
```



Unbinned, ordinal, levels out of order

INCORRECT

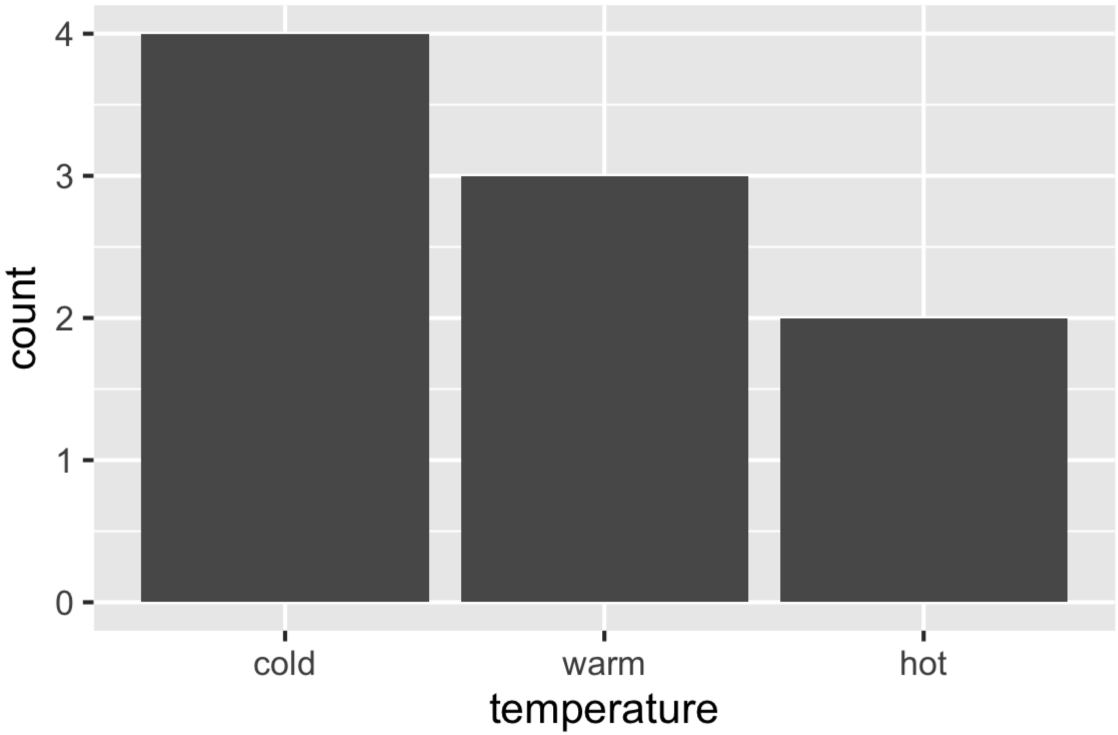
```r
df <- tibble(temperature = factor(c("cold", "warm", "hot", "hot", "warm",
                                    "warm", "cold", "cold", "cold")))
ggplot(df, aes(x = temperature)) +
  geom_bar() +
  theme_grey(16)
```


4
3
2
1
0
count
cold
hot
warm
temperature

Unbinned, ordinal, levels out of order

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

```r
# SOLUTION
df <- tibble(temperature = factor(c("cold", "warm", "hot", "hot", "warm",
                                    "warm", "cold", "cold", "cold")))
df |>
  mutate(temperature = fct_relevel(temperature, "warm", after = 1)) %>%
  ggplot(aes(temperature)) +
  geom_bar() +
  theme_grey(16)
```


4
3
2
1
0
count
cold
warm
hot
temperature

Unbinned, nominal data

```r
dim(df)
```

```
[1] 100   1
```

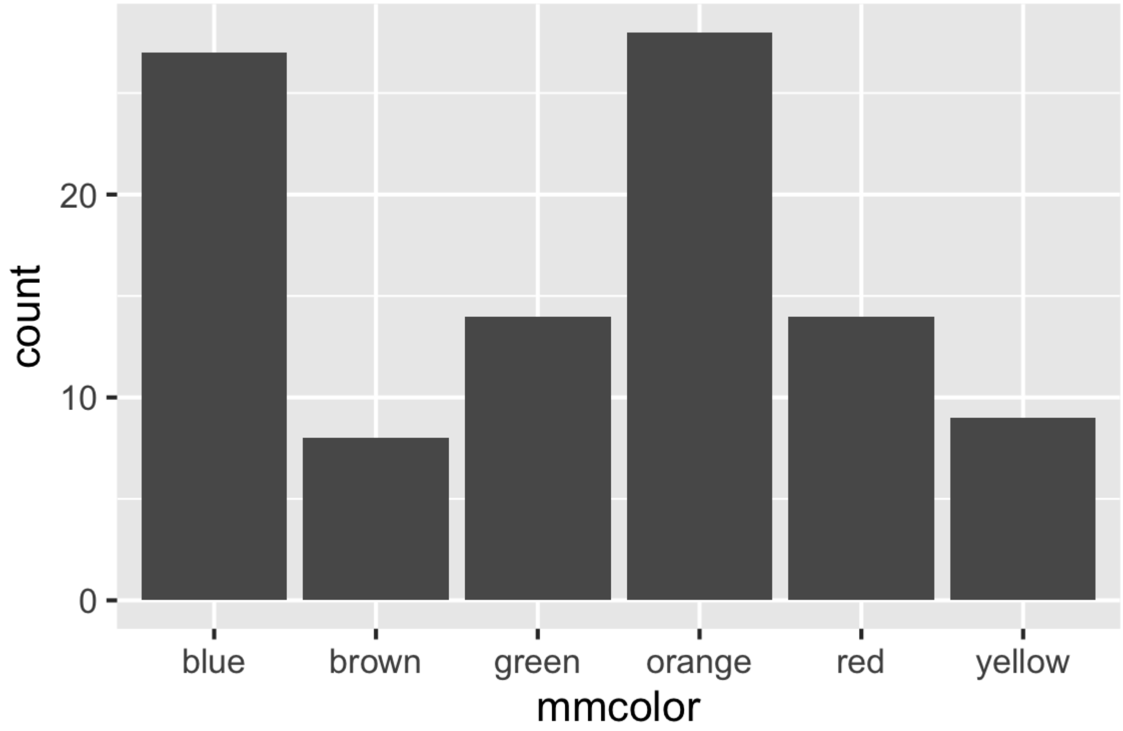
```r
head(df, 10)
```

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

Unbinned, nominal data

INCORRECT

```r
ggplot(df, aes(mmcolor)) +
  geom_bar() +
  theme_grey(16)
```

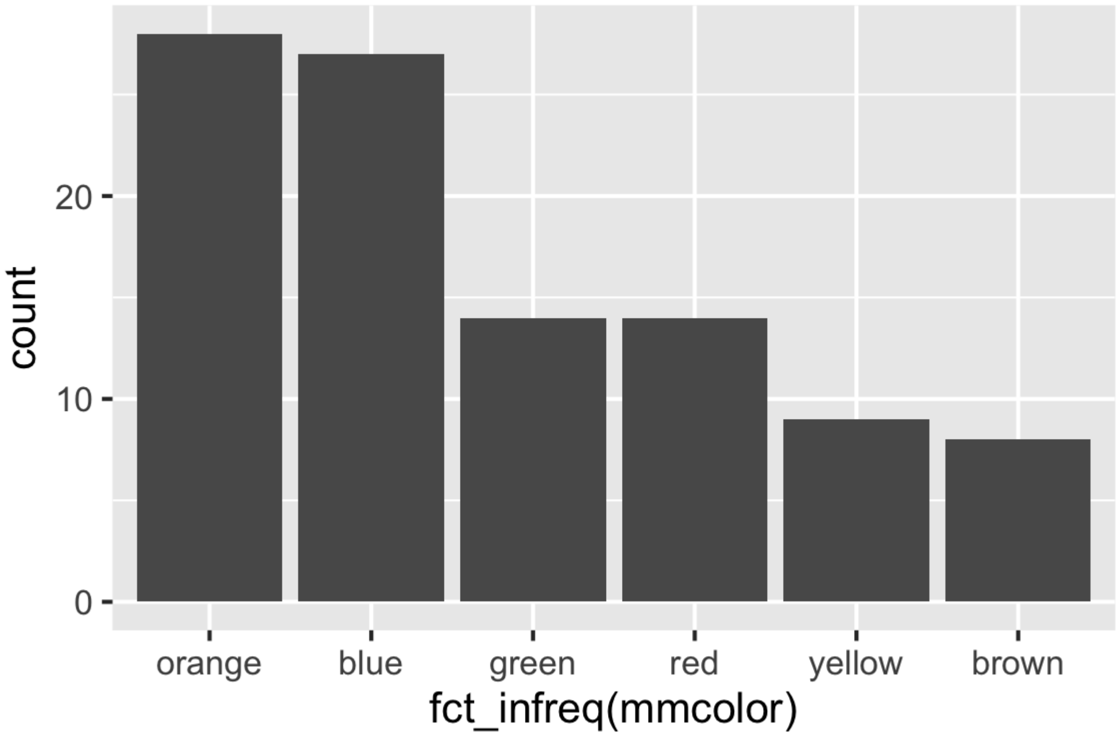


Unbinned, nominal data

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

Vertical bars:

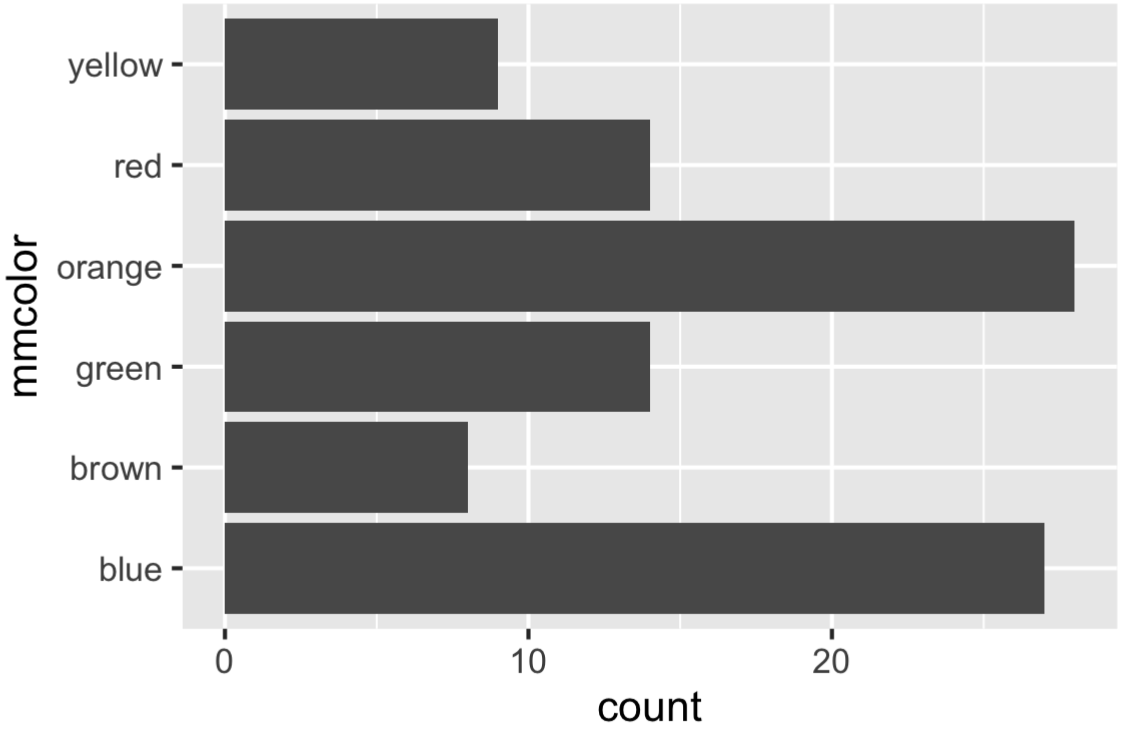
```r
# SOLUTION
ggplot(df, aes(fct_infreq(mmcolor))) +
  geom_bar() +
  theme_grey(16)
```


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

Unbinned, nominal data, horizontal bars

INCORRECT

```r
ggplot(df, aes(y = mmcolor)) +
  geom_bar() +
  theme_grey(16)
```

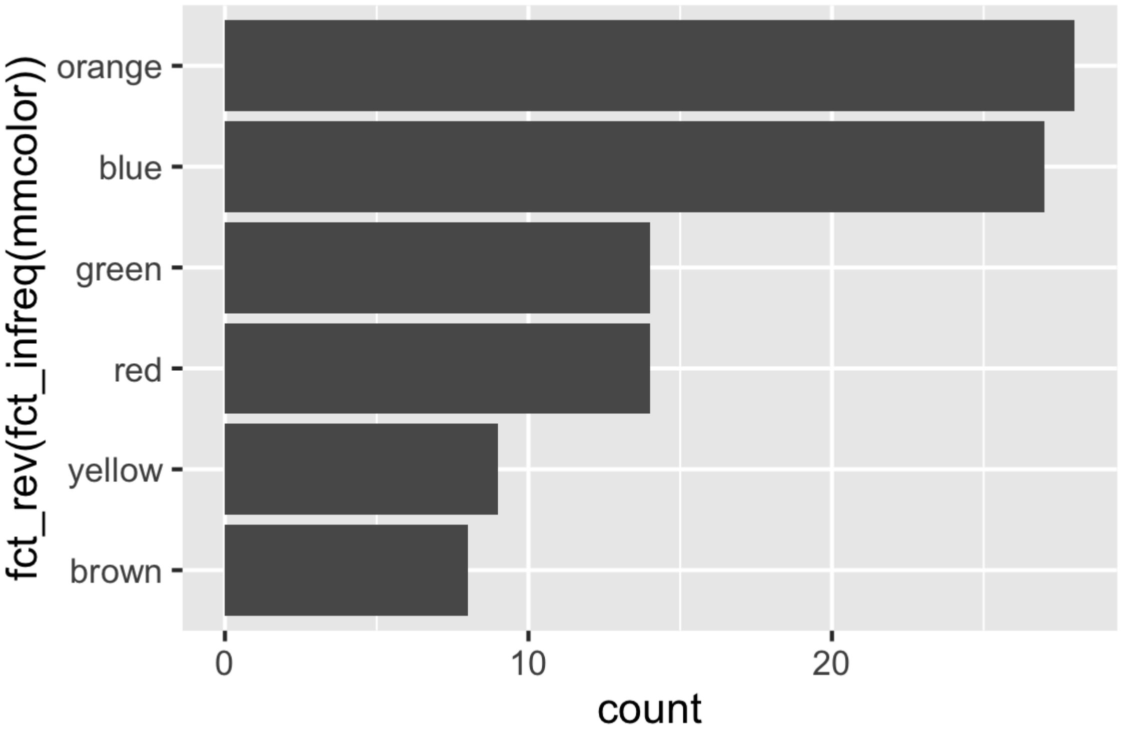


Unbinned, nominal data

`fct_rev(fct_infreq())`

Horizontal bars:

```r
# SOLUTION
ggplot(df, aes(y = fct_rev(fct_infreq(mmcolor)))) +
  geom_bar() +
  theme_grey(16)
```


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