

Categorical Variable How-to

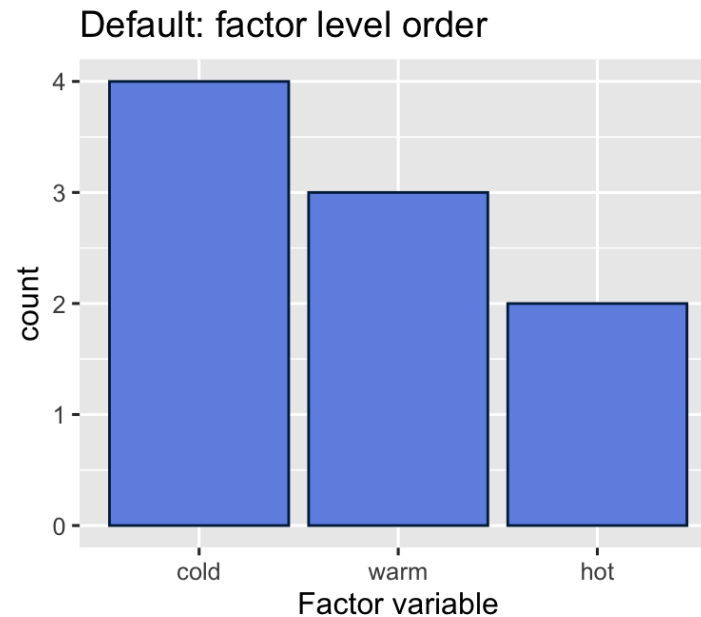
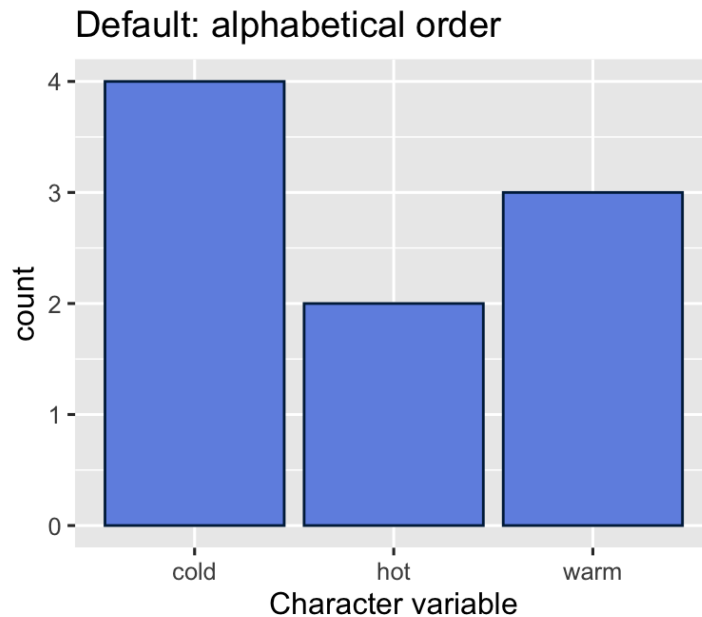
Prof. Joyce Robbins

Character vs factor data

character data: plotted alphabetically

factor data: plotted in order of factor levels

```
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")))
```



Recoding factor levels: don't assign levels with `levels()`

Not the best approach

```
x <- factor(c("G234", "G452", "G136"))  
levels(x)
```

```
## [1] "G136" "G234" "G452"
```

```
levels(x) <- c("Physics", "Math", "Chemistry")  
x
```

```
## [1] Math      Chemistry Physics  
## Levels: Physics Math Chemistry
```

Recoding factor levels

Not the best approach

```
x <- factor(c("G234", "G452", "G136"))  
levels(x)
```

```
## [1] "G136" "G234" "G452"
```

```
levels(x) <- c("Physics", "Math", "Chemistry")  
x
```

```
## [1] Math      Chemistry Physics  
## Levels: Physics Math Chemistry
```

levels(x) ✓

~~levels(x) <-~~

Recoding factor levels: `fct_recode()`

A better approach: Keep a trail of breadcrumbs

```
x <- factor(c("G234", "G452", "G136"))  
y <- fct_recode(x, Physics = "G234", Math = "G452", Chemistry = "G136")  
y
```

```
## [1] Physics    Math      Chemistry  
## Levels: Chemistry Physics Math
```

Binned data

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

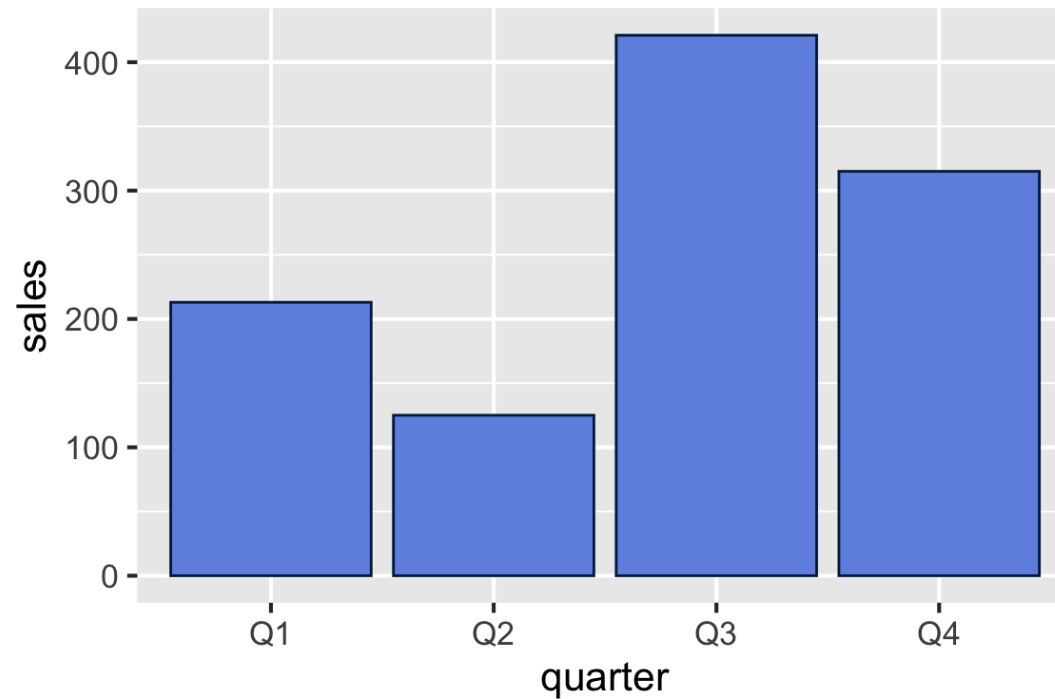
```
##      quarter sales  
## 1         Q1    213  
## 2         Q2    125  
## 3         Q3    421  
## 4         Q4    315
```

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

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

Binned, ordinal data, correct level order

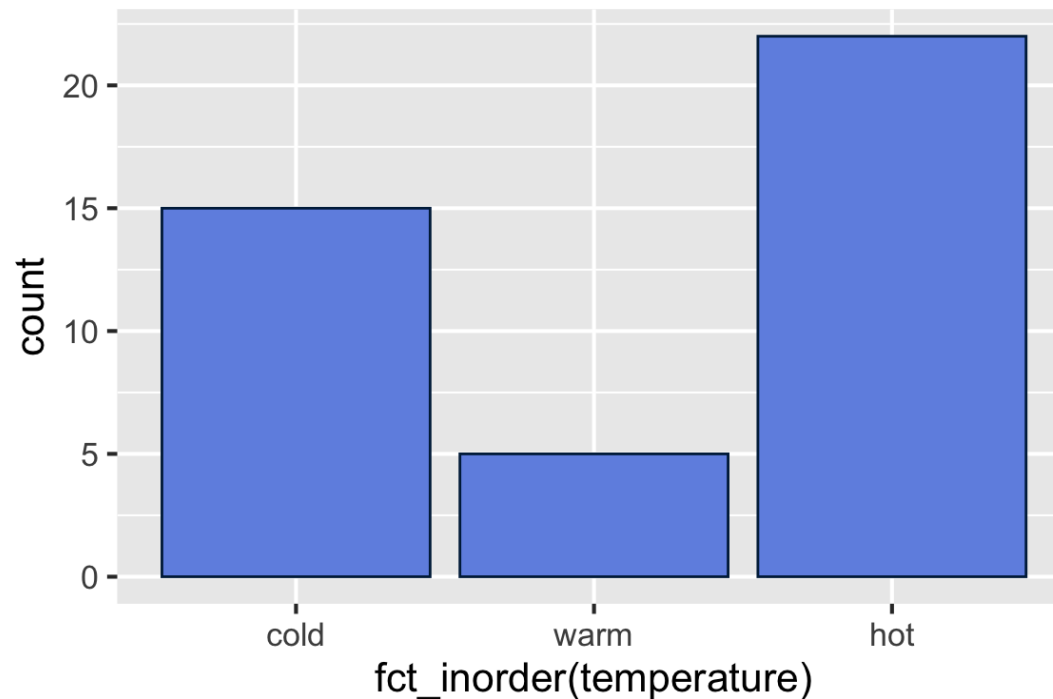
```
# reordering is not necessary  
ggplot(df, aes(x = quarter, y = sales)) +  
  geom_col(color = mycolor, fill = myfill) +  
  theme_grey(16)
```



Binned, ordinal data, levels out of order

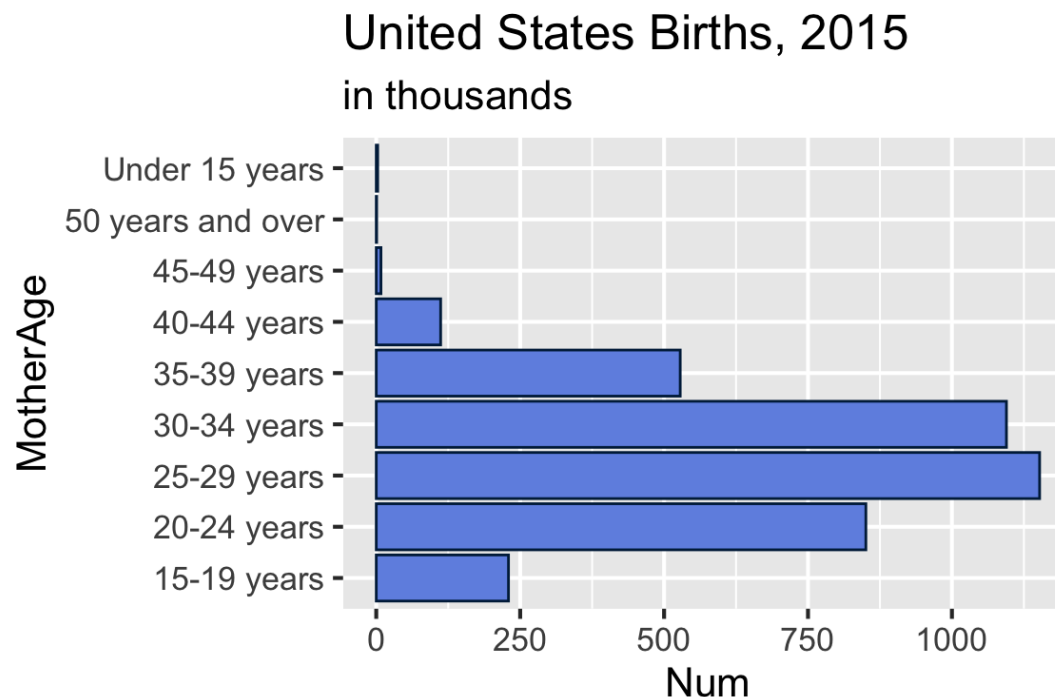
If the row order is correct, use `fct_inorder()`

```
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)
```



Binned, ordinal data, levels out of order

```
Births2015 <- tibble(MotherAge = c("15-19 years", "20-24 years", "25-29 years", "30-34 years", "35-39 years",  
                                "40-44 years", "45-49 years", "50 years and over", "Under 15 years"),  
                    Num = c(229.715, 850.509, 1152.311, 1094.693, 527.996, 111.848, 8.171, .754, 2.5))  
  
ggplot(Births2015, aes(MotherAge, Num)) +  
  geom_col(color = mycolor, fill = myfill) +  
  ggtitle("United States Births, 2015", subtitle = "in thousands") +  
  scale_y_continuous(breaks = seq(0, 1250, 250)) +  
  coord_flip() +  
  theme_grey(16)
```

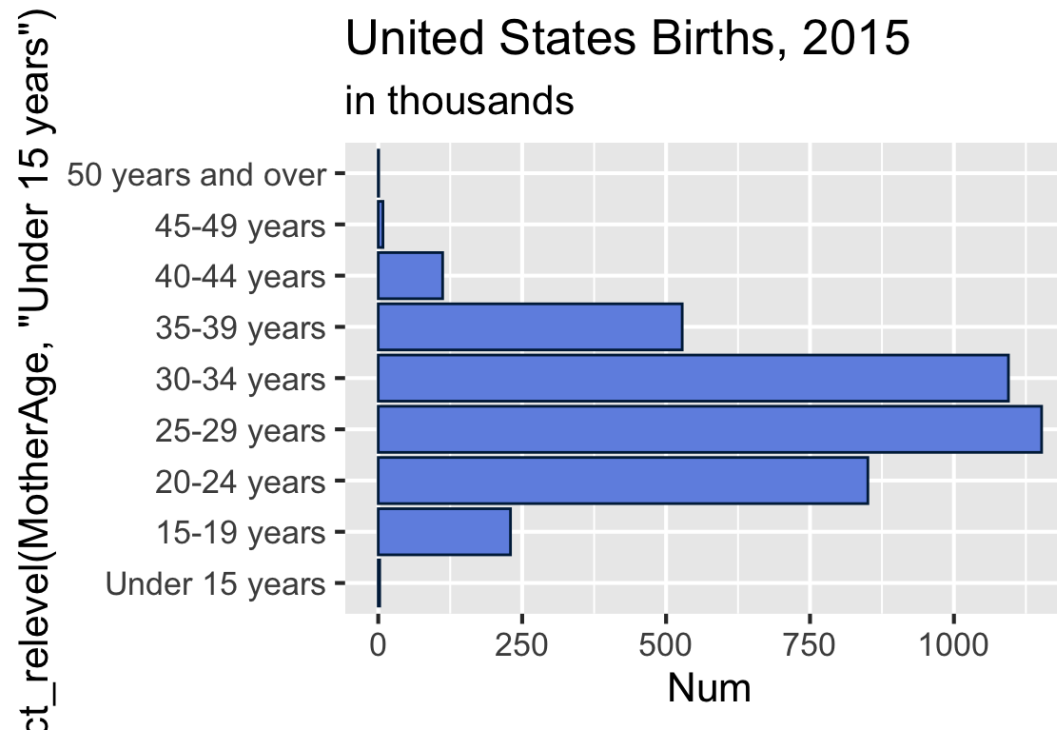


What's the problem?

Binned, ordinal data, levels out of order

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

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



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

```
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)

```
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

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

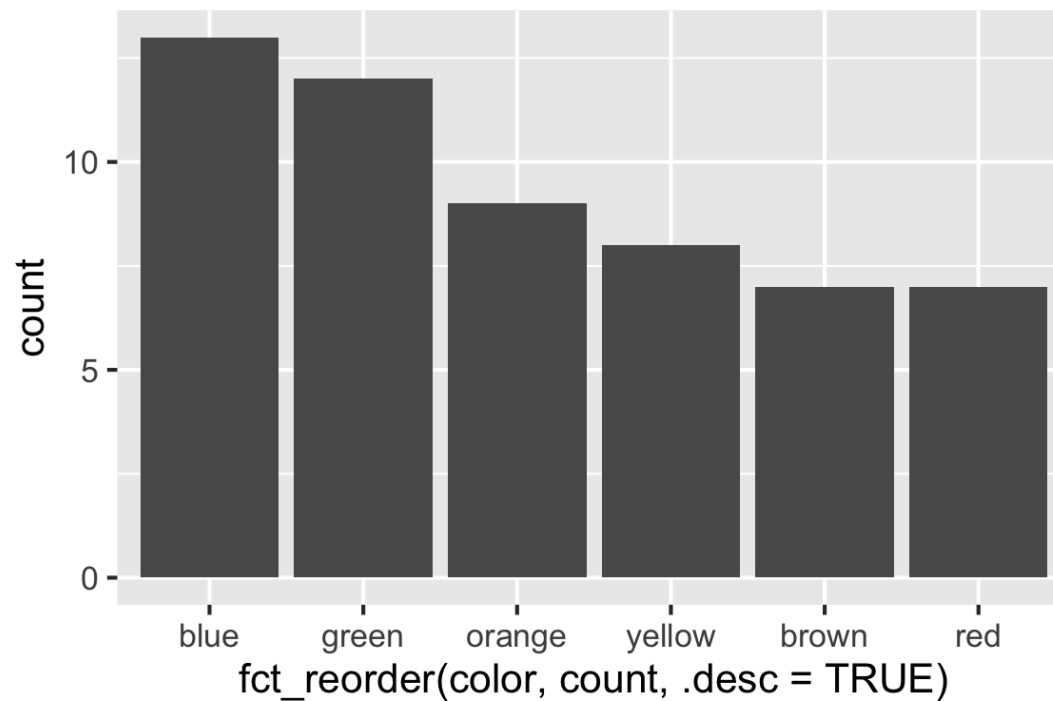
```
fct_relevel(x, "move1", "move2", after = Inf)
```

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

Binned, nominal

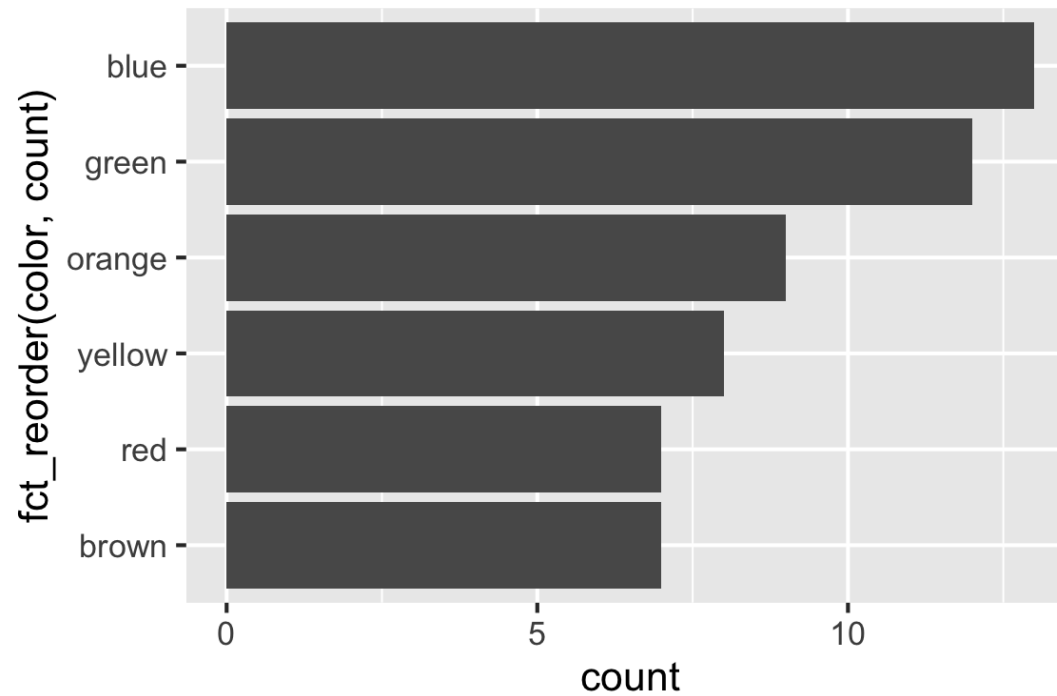
Order bars by frequency count using `fct_reorder()` (or `reorder()`)

```
pack1 <- data.frame(  
  color = c("blue", "brown", "green", "orange", "red", "yellow"),  
  count = c(13, 7, 12, 9, 7, 8)  
)  
  
ggplot(pack1, aes(fct_reorder(color, count, .desc = TRUE), count)) +  
  geom_col() +  
  theme_grey(16)
```



Binned, nominal (horizontal bars)

```
ggplot(pack1, aes(fct_reorder(color, count), count)) +  
  geom_col() +  
  coord_flip() +  
  theme_grey(16)
```



Unbinned, ordinal, correct level order

```
# data available here: https://github.com/jtrl3/data  
student <- read.csv("student_data.csv")  
str(student)
```

```
## 'data.frame':    44 obs. of  3 variables:  
## $ School      : chr  "CC " "CC " "CC " "CC " ...  
## $ Level       : chr  "U01" "U01" "U01" "U01" ...  
## $ Affiliation: chr  "CCUNDC" "CCUNDC" "CCUNDC" "CCUNDC" ...
```

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

```
## NULL
```

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

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

Unbinned, ordinal, correct level order

```
# 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"...
```

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

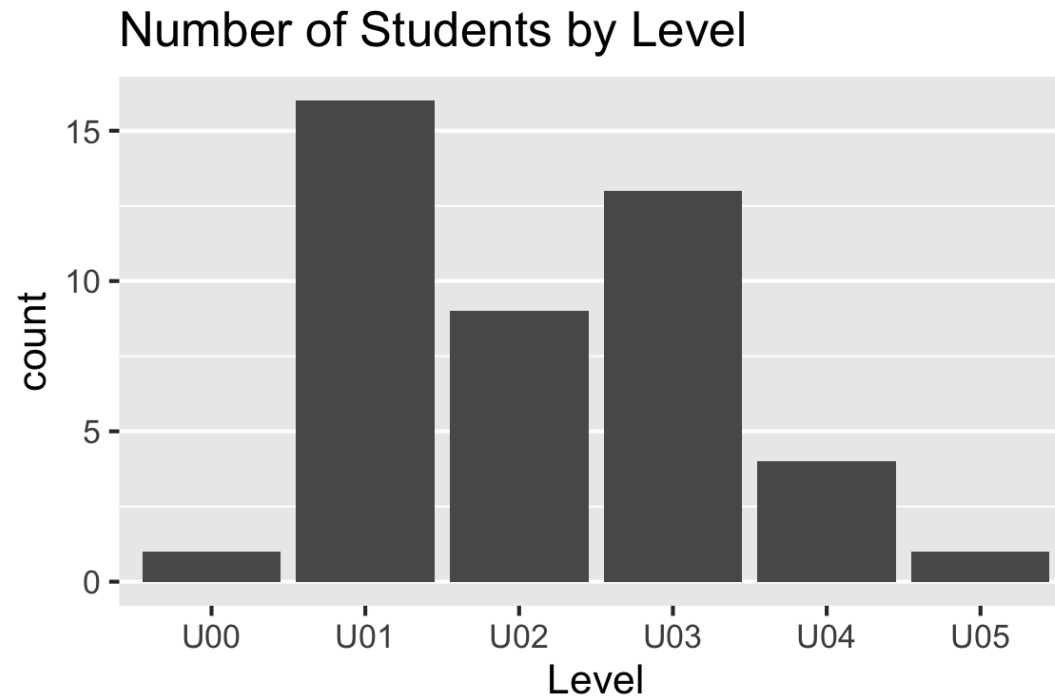
```
## NULL
```

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

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

Unbinned, ordinal, correct level order

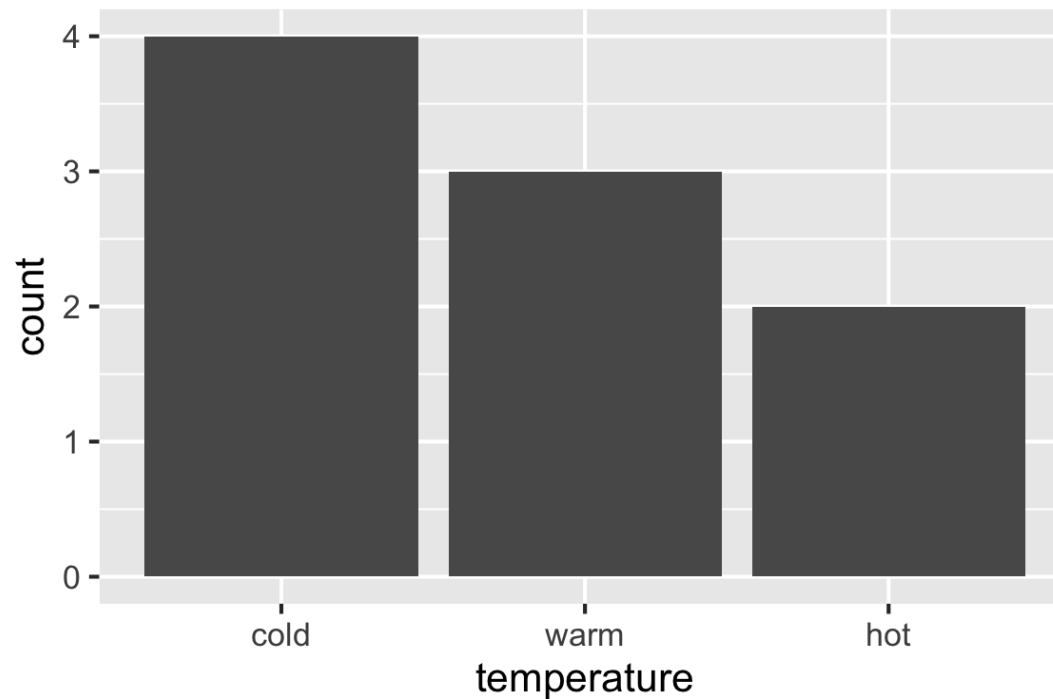
```
ggplot(student, aes(Level)) +  
  geom_bar() +  
  ggtitle("Number of Students by Level") +  
  theme_grey(16) +  
  theme(panel.grid.major.x = element_blank())
```



Unbinned, ordinal, levels out of order

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

```
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)
```



Unbinned, nominal data

```
dim(df)
```

```
## [1] 100  1
```

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

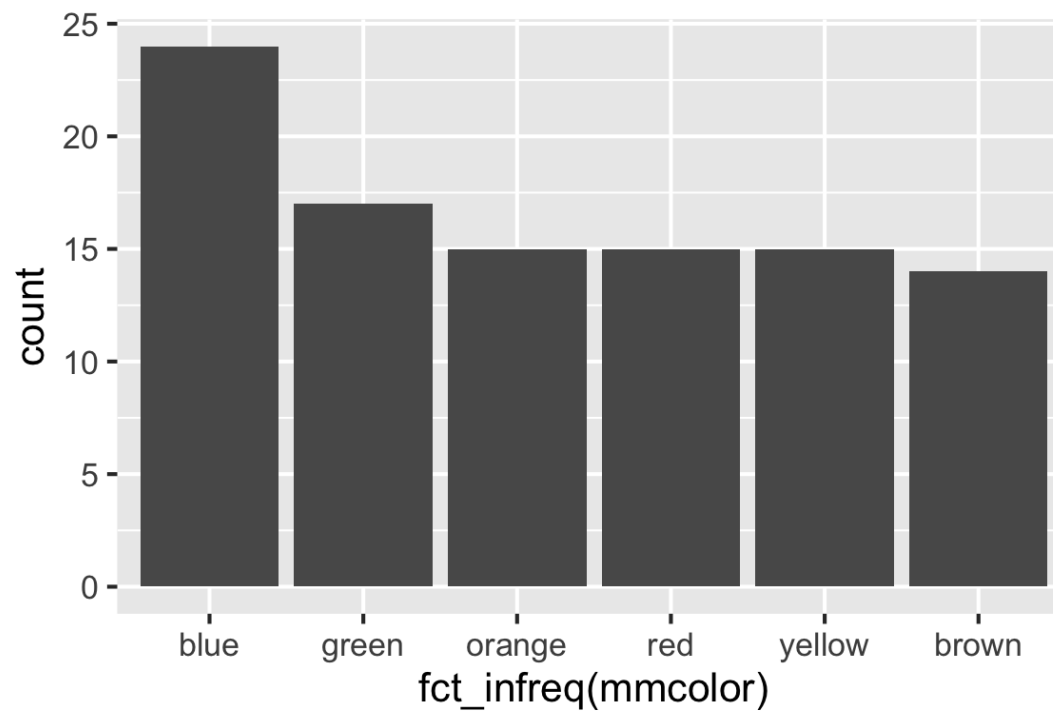
```
##      mmcolor  
## 1    green  
## 2     blue  
## 3   yellow  
## 4   orange  
## 5   orange  
## 6     blue  
## 7      red  
## 8   orange  
## 9   brown  
## 10  yellow
```

Unbinned, nominal data

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

Vertical bars:

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

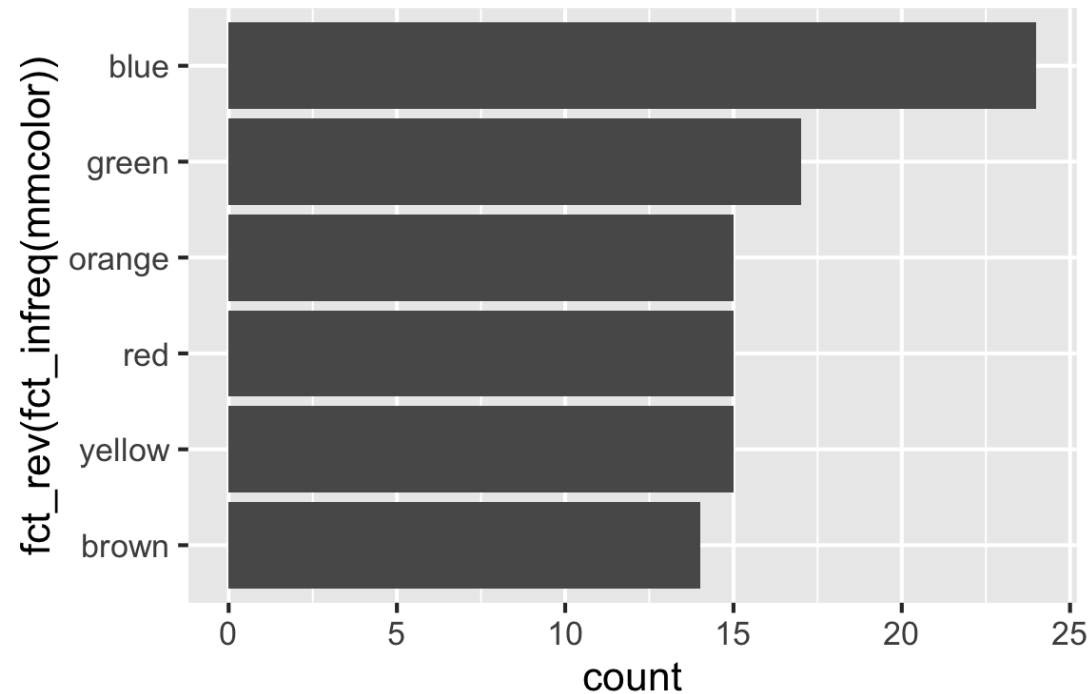


Unbinned, nominal data

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

Horizontal bars:

```
ggplot(df, aes(fct_rev(fct_infreq(mmcolor)))) +  
  geom_bar() +  
  coord_flip() +  
  theme_grey(16)
```



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

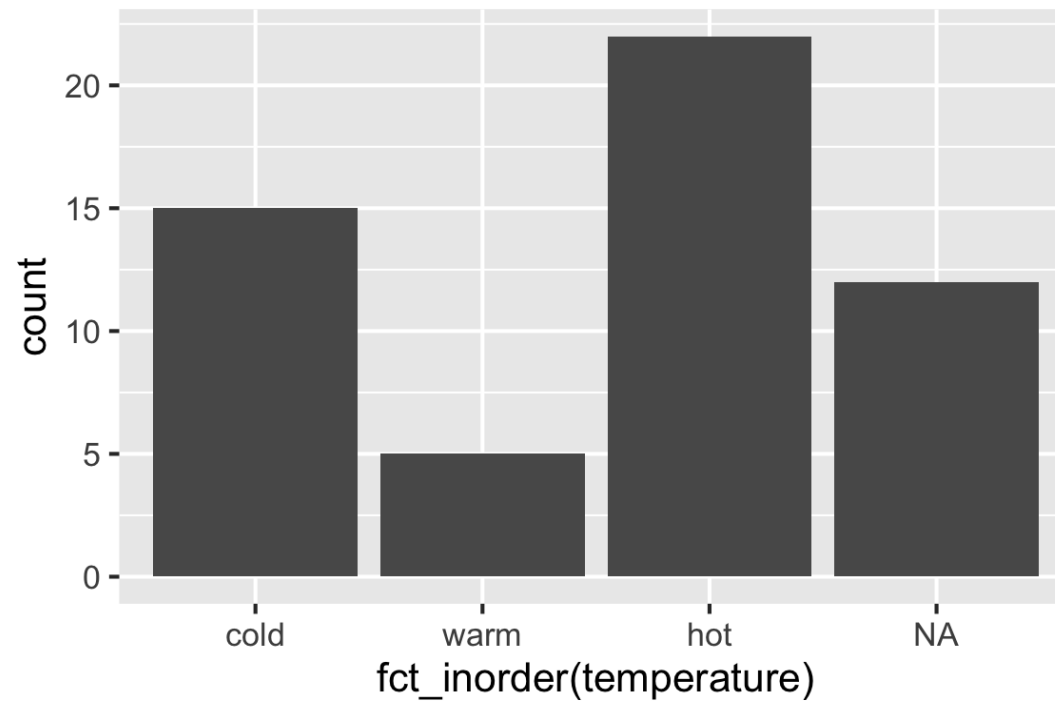
`fct_rev(x)` – reverse the order of factor levels of `x`

coming up:

`fct_explicit_na(x)` – turn NAs into a real factor level

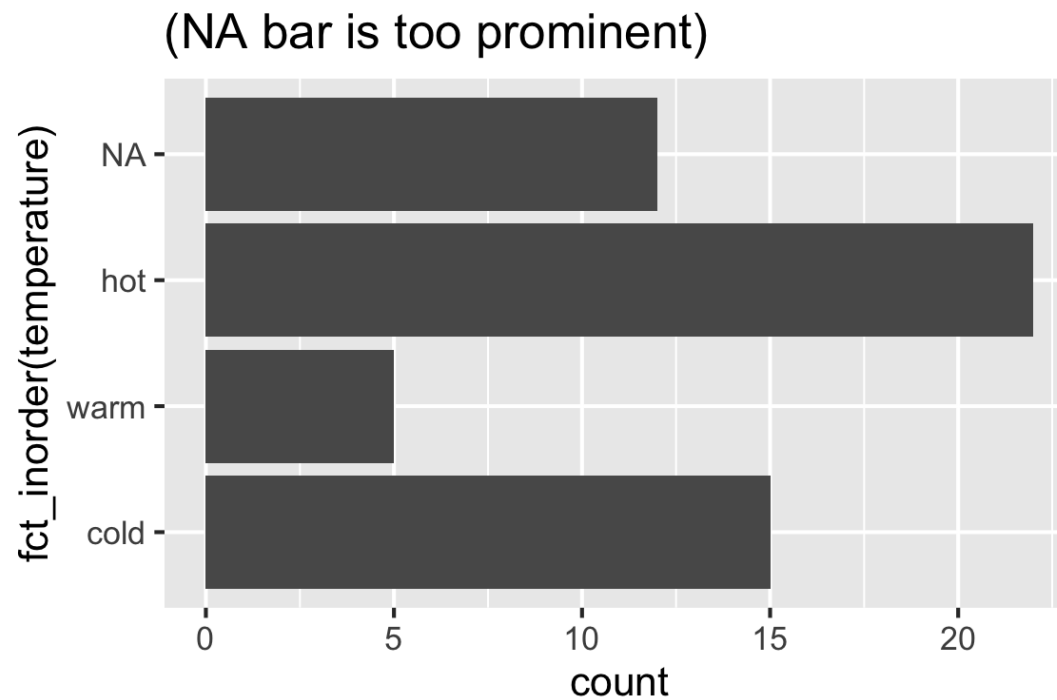
Dealing with NAs

```
df <- data.frame(temperature = factor(c("cold", "warm", "hot", NA)),  
  count = c(15, 5, 22, 12))  
  
ggplot(df, aes(x = fct_inorder(temperature), y = count)) +  
  geom_col() +  
  theme_grey(16)
```



Dealing with NAs

```
df <- data.frame(temperature = factor(c("cold", "warm", "hot", NA)),  
  count = c(15, 5, 22, 12))  
  
ggplot(df, aes(x = fct_inorder(temperature), y = count)) +  
  geom_col() +  
  coord_flip() +  
  ggtitle("(NA bar is too prominent)") +  
  theme_grey(16)
```



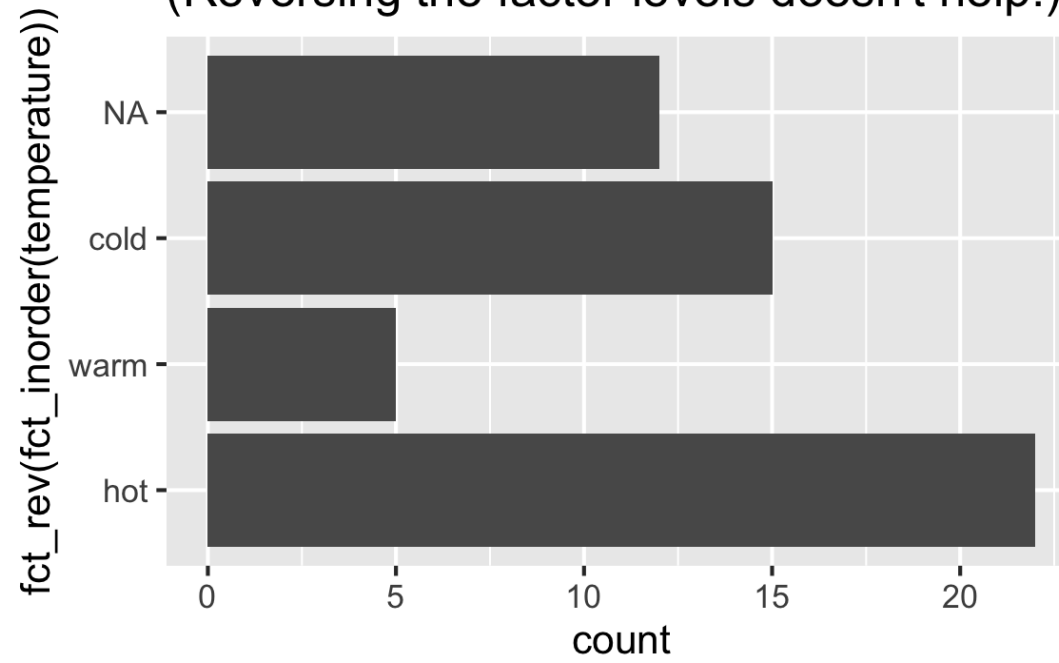
Dealing with NAs

```
df <- data.frame(temperature = factor(c("cold", "warm", "hot", NA)),  
                 count = c(15, 5, 22, 12))  
df
```

```
##   temperature count  
## 1         cold    15  
## 2         warm     5  
## 3          hot    22  
## 4         <NA>    12
```

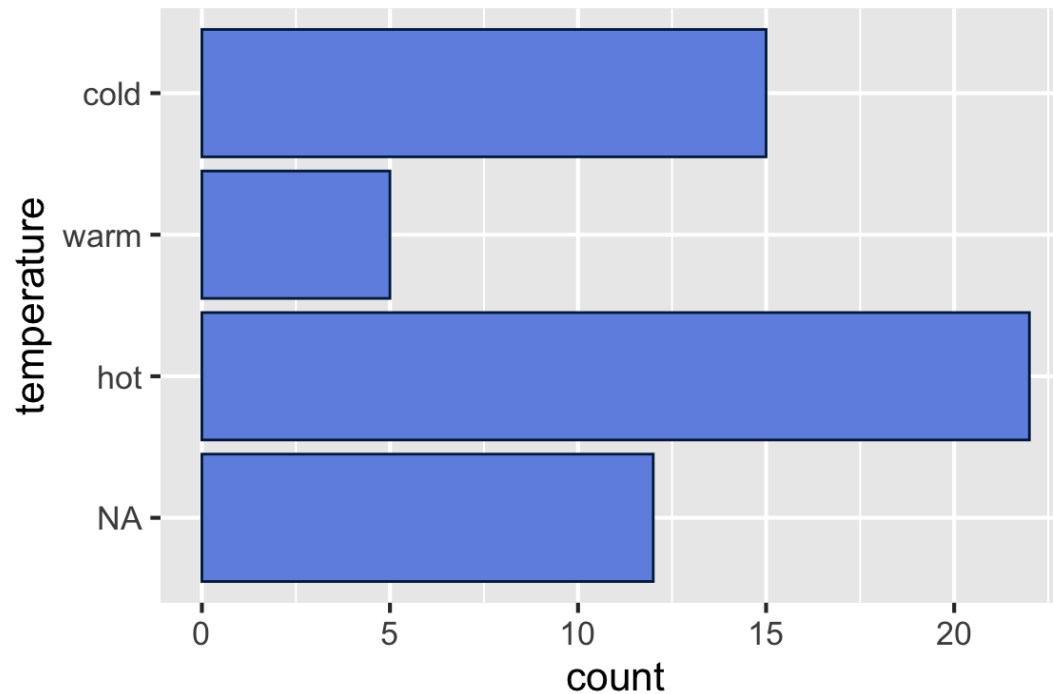
```
ggplot(df, aes(x = fct_rev(fct_inorder(temperature)), y = count)) +  
  geom_col() +  
  coord_flip() +  
  ggtitle("(Reversing the factor levels doesn't help.)") +  
  theme_grey(16)
```

(Reversing the factor levels doesn't help.)



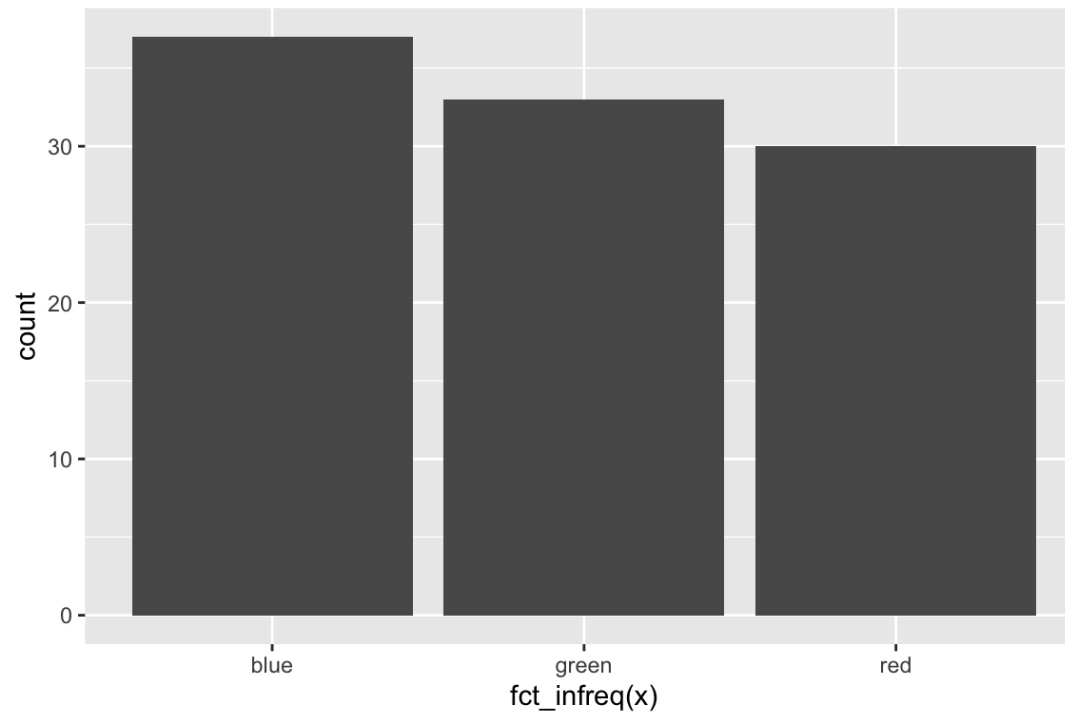
Dealing with NAs

```
df <- data.frame(temperature = factor(c("cold", "warm", "hot", NA)),  
                 count = c(15, 5, 22, 12))  
  
df %>%  
  mutate(temperature = fct_explicit_na(temperature, "NA") %>%  
    fct_relevel("NA", "hot", "warm", "cold")) %>%  
  ggplot(aes(x = temperature, y = count)) +  
  geom_col(color = mycolor, fill = myfill) +  
  coord_flip() +  
  theme_grey(16)
```



Binning

```
df <- data.frame(x = sample(c("red", "green", "blue"), 100, replace = TRUE))  
  
ggplot(df, aes(fct_infreq(x))) + geom_bar()
```



```
df %>% group_by(x) %>% summarize(n = n())
```

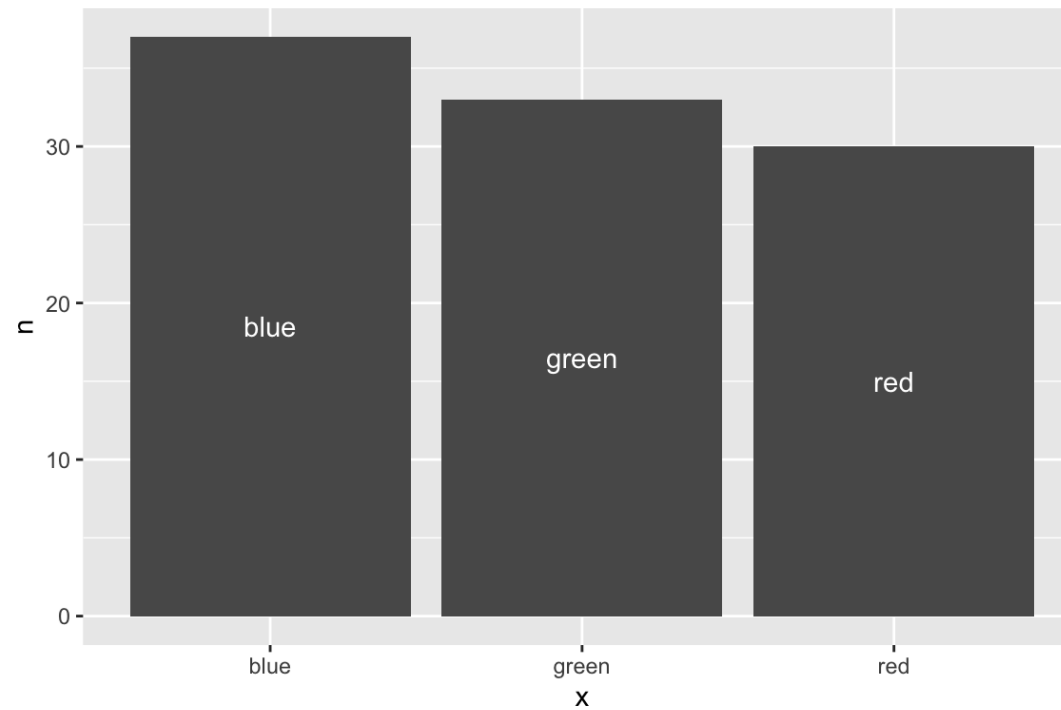
```
## # A tibble: 3 × 2  
##   x         n  
##   <chr> <int>  
## 1 blue    37  
## 2 green   33  
## 3 red     30
```

```
binned_df <- df %>% count(x)
```

```
binned_df
```

```
##      x  n  
## 1 blue 37  
## 2 green 33  
## 3 red 30
```

```
ggplot(binned_df, aes(x = x, y = n, label = x)) +  
  geom_col() +  
  geom_text(aes(y = n/2), col = "white")
```

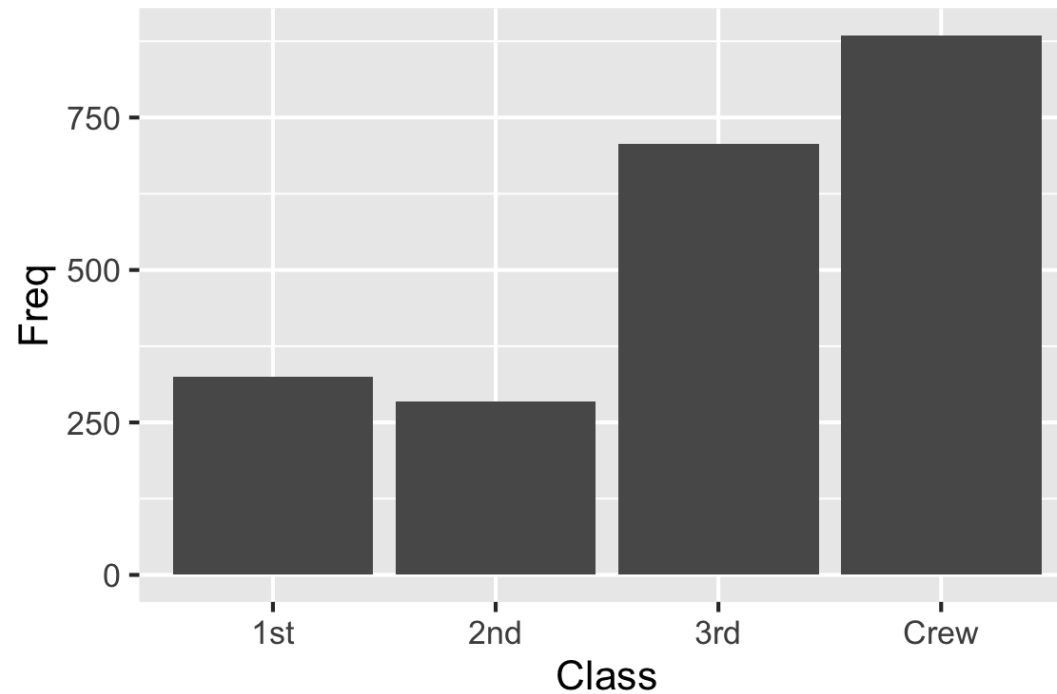


Rebinning

```
df <- as.data.frame(Titanic)
head(df)
```

##	Class	Sex	Age	Survived	Freq
## 1	1st	Male	Child	No	0
## 2	2nd	Male	Child	No	0
## 3	3rd	Male	Child	No	35
## 4	Crew	Male	Child	No	0
## 5	1st	Female	Child	No	0
## 6	2nd	Female	Child	No	0

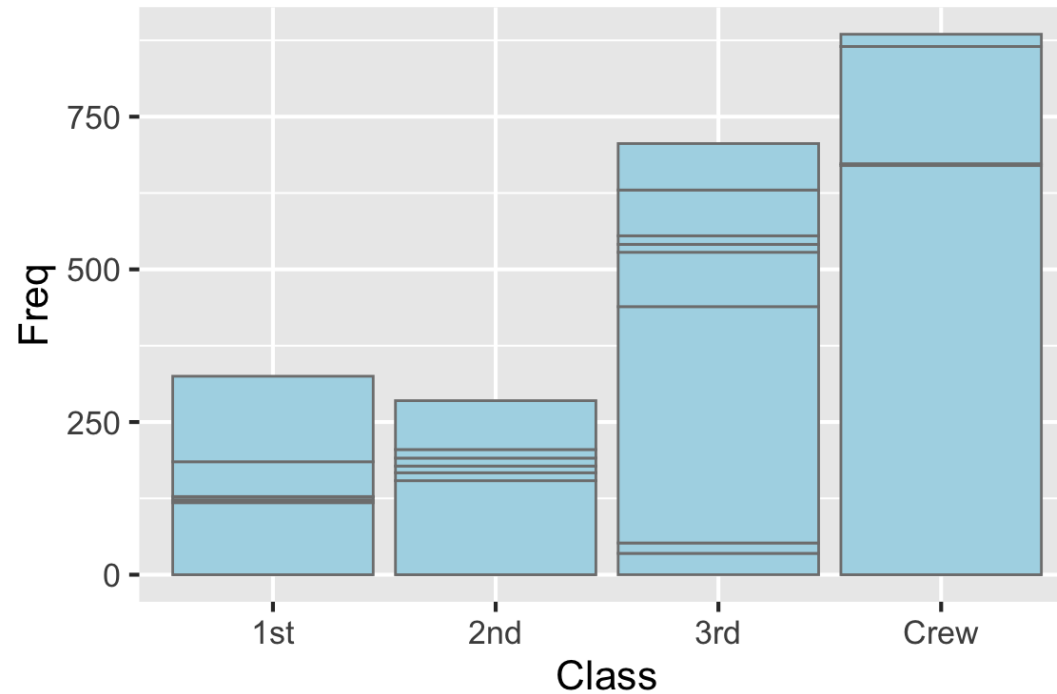
```
ggplot(df, aes(Class, Freq)) +
  geom_col() +
  theme_grey(16)
```



Rebinning

The problem:

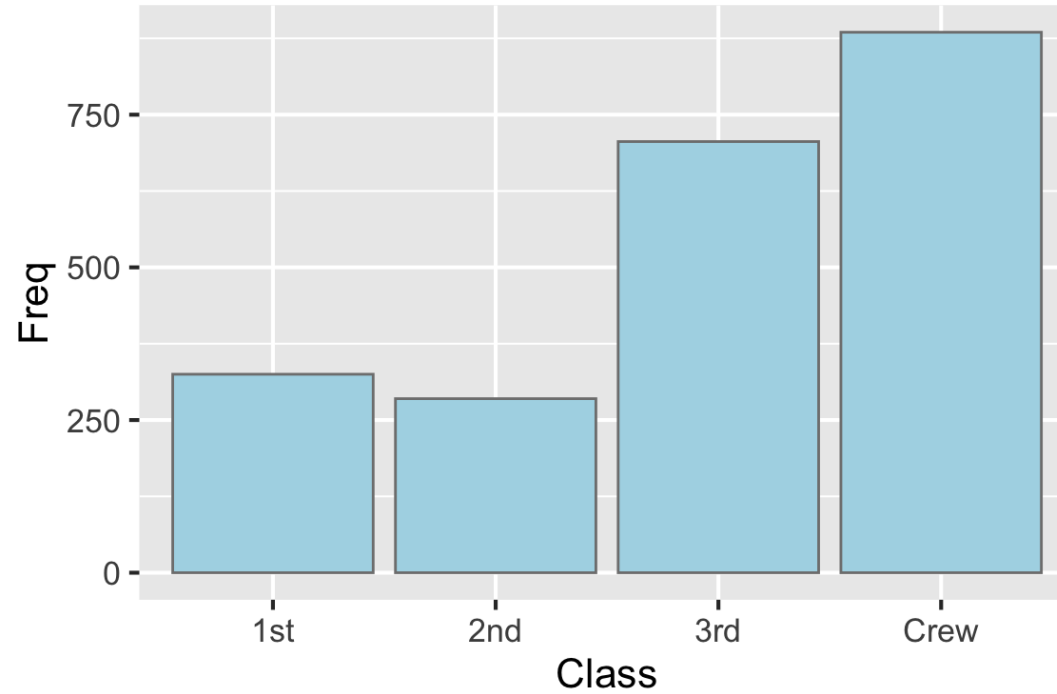
```
ggplot(df, aes(Class, Freq)) +  
  geom_col(color = "grey50", fill = "lightblue") +  
  theme_grey(16)
```



Rebinning

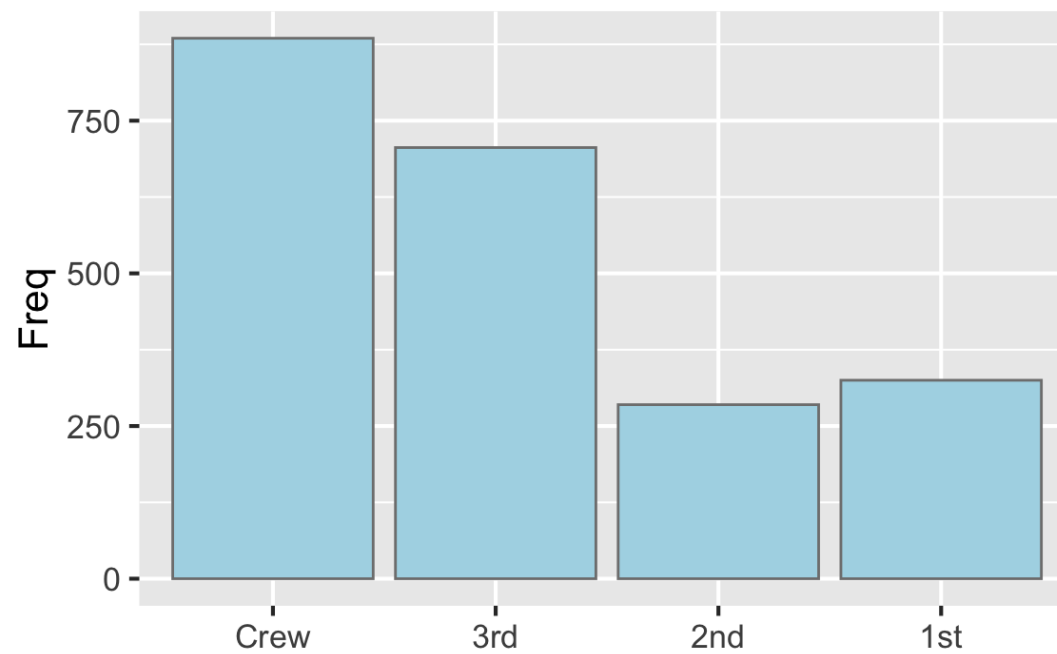
Rebin:

```
df %>%  
  group_by(Class) %>%  
  summarize(Freq = sum(Freq)) %>%  
  ggplot(aes(Class, Freq)) +  
  geom_col(color = "grey50", fill = "lightblue") +  
  theme_grey(16)
```



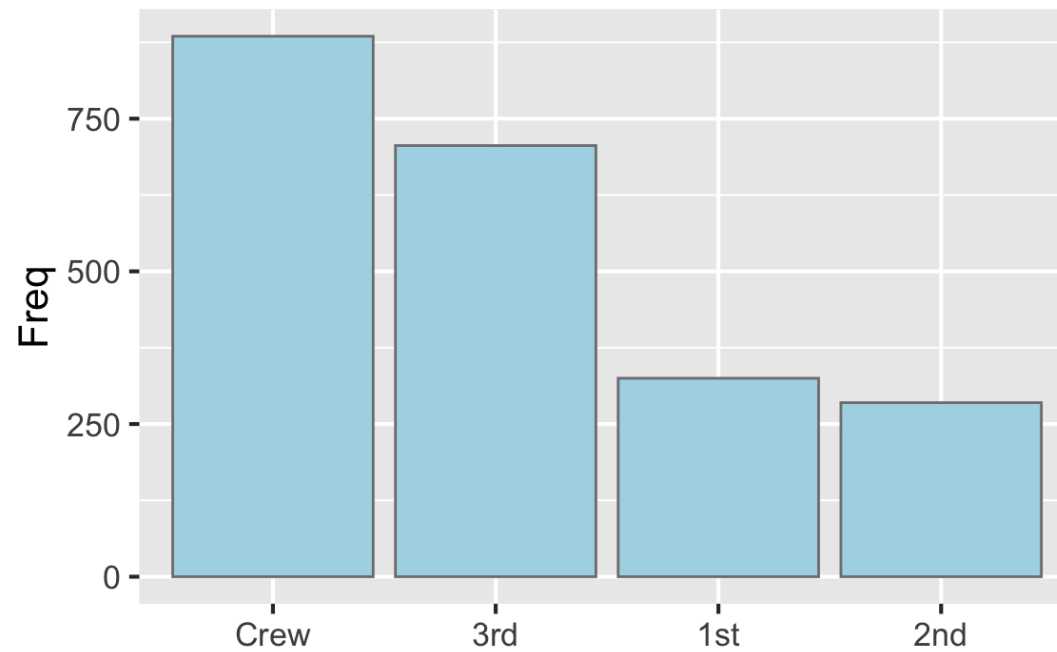
Natural order bias?

```
df %>%  
  group_by(Class) %>%  
  summarize(Freq = sum(Freq)) %>%  
  ggplot(aes(fct_rev(Class), Freq)) +  
  geom_col(color = "grey50", fill = "lightblue") +  
  xlab("") +  
  theme_grey(16)
```



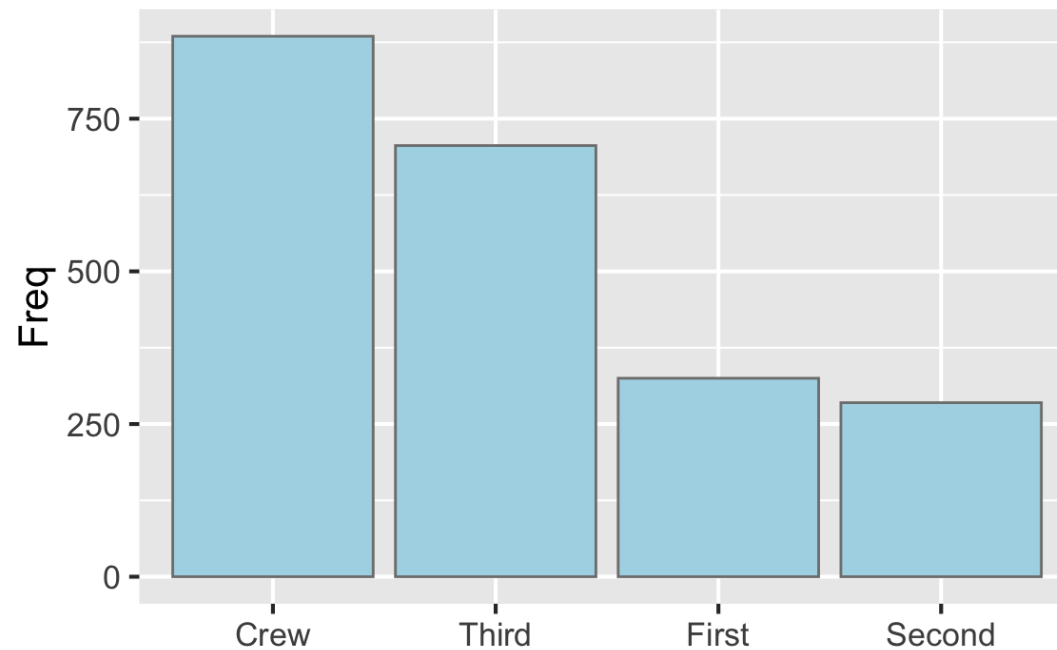
Is Class ordinal or nominal?

```
df %>%  
  group_by(Class) %>%  
  summarize(Freq = sum(Freq)) %>%  
  ggplot(aes(fct_reorder(Class, Freq, .desc = TRUE), Freq)) +  
  geom_col(color = "grey50", fill = "lightblue") +  
  xlab("") +  
  theme_grey(16)
```



Is Class ordinal or nominal?

```
df %>%  
  mutate(Class = fct_recode(Class, Third = "3rd", First = "1st", Second = "2nd")) %>%  
  group_by(Class) %>%  
  summarize(Freq = sum(Freq)) %>%  
  ggplot(aes(fct_reorder(Class, Freq, .desc = TRUE), Freq)) +  
  geom_col(color = "grey50", fill = "lightblue") + xlab("") +  
  theme_grey(16)
```



Percentages

```
df %>%  
  group_by(Class) %>%  
  summarize(Freq = sum(Freq)) %>%  
  mutate(prop = Freq/sum(Freq))
```

```
## # A tibble: 4 × 3  
##   Class  Freq  prop  
##   <fct> <dbl> <dbl>  
## 1 1st      325 0.148  
## 2 2nd      285 0.129  
## 3 3rd      706 0.321  
## 4 Crew     885 0.402
```

Percentages, more than one group

Rebin first:

```
df2 <- df %>%  
  group_by(Class, Survived) %>%  
  summarize(Freq = sum(Freq)) %>%  
  ungroup() # very important  
df2
```

```
## # A tibble: 8 × 3  
##   Class Survived Freq  
##   <fct> <fct>   <dbl>  
## 1 1st    No      122  
## 2 1st    Yes     203  
## 3 2nd    No      167  
## 4 2nd    Yes     118  
## 5 3rd    No     528  
## 6 3rd    Yes     178  
## 7 Crew   No     673  
## 8 Crew   Yes     212
```


Percentages, more than one group

Overall percentages:

```
df2 %>%
```

```
mutate(prop = Freq/sum(Freq))
```

```
## # A tibble: 8 × 4
##   Class Survived  Freq  prop
##   <fct> <fct>    <dbl> <dbl>
## 1 1st    No       122 0.0554
## 2 1st    Yes      203 0.0922
## 3 2nd    No       167 0.0759
## 4 2nd    Yes      118 0.0536
## 5 3rd    No      528 0.240
## 6 3rd    Yes      178 0.0809
## 7 Crew   No      673 0.306
## 8 Crew   Yes      212 0.0963
```

Percentages, more than one group

Proportions for each class sum to 1:

```
df2 %>%  
  group_by(Class) %>%  
  mutate(prop = Freq/sum(Freq)) %>%  
  ungroup()
```

```
## # A tibble: 8 × 4  
##   Class Survived Freq prop  
##   <fct> <fct>     <dbl> <dbl>  
## 1 1st    No        122 0.375  
## 2 1st    Yes        203 0.625  
## 3 2nd    No        167 0.586  
## 4 2nd    Yes        118 0.414  
## 5 3rd    No        528 0.748  
## 6 3rd    Yes        178 0.252  
## 7 Crew   No        673 0.760  
## 8 Crew   Yes        212 0.240
```

Proportions for each level of Survived sum to 1:

```
df2 %>%  
  # (arrange reorders the rows for viewing)  
  arrange(Survived) %>%  
  group_by(Survived) %>%  
  mutate(prop = Freq/sum(Freq)) %>%  
  ungroup()
```

```
## # A tibble: 8 × 4  
##   Class Survived Freq prop  
##   <fct> <fct>     <dbl> <dbl>  
## 1 1st    No        122 0.0819  
## 2 2nd    No        167 0.112  
## 3 3rd    No        528 0.354  
## 4 Crew   No        673 0.452  
## 5 1st    Yes        203 0.286  
## 6 2nd    Yes        118 0.166  
## 7 3rd    Yes        178 0.250  
## 8 Crew   Yes        212 0.298
```

Percentages, more than one group

shortcut method (be careful!)

```
df %>%  
  group_by(Class, Survived) %>%      # grouped by Class, Survived  
  summarize(Freq = sum(Freq)) %>%    # grouped by Class only  
  mutate(prop = Freq/sum(Freq)) %>%  
  ungroup()
```

```
## # A tibble: 8 × 4  
##   Class Survived   Freq prop  
##   <fct> <fct>     <dbl> <dbl>  
## 1 1st     No         122 0.375  
## 2 1st     Yes         203 0.625  
## 3 2nd     No         167 0.586  
## 4 2nd     Yes         118 0.414  
## 5 3rd     No         528 0.748  
## 6 3rd     Yes         178 0.252  
## 7 Crew    No         673 0.760  
## 8 Crew    Yes         212 0.240
```

summarize() removes the last group

```
groups(df)
```

```
## list()
```

```
df %>% group_by(Class, Survived) %>% groups()
```

```
## [[1]]  
## Class  
##  
## [[2]]  
## Survived
```

```
df %>% group_by(Class, Survived) %>% summarize(Freq = sum(Freq)) %>% groups()
```

```
## [[1]]  
## Class
```

Percentages, more than one group

shortcut method (be careful!)

```
df %>%  
  group_by(Survived, Class) %>%      # grouped by Survived, Class ORDER MATTERS  
  summarize(Freq = sum(Freq)) %>%   # grouped by Survived only  
  mutate(prop = Freq/sum(Freq)) %>%  
  ungroup()
```

```
## # A tibble: 8 × 4  
##   Survived Class   Freq   prop  
##   <fct>    <fct> <dbl> <dbl>  
## 1 No      1st     122 0.0819  
## 2 No      2nd     167 0.112  
## 3 No      3rd     528 0.354  
## 4 No      Crew    673 0.452  
## 5 Yes     1st     203 0.286  
## 6 Yes     2nd     118 0.166  
## 7 Yes     3rd     178 0.250  
## 8 Yes     Crew    212 0.298
```