Categorical data

slides/05_categorical.pdf

Numeric data

```
 1 library(ade4)
 2 data("clementines")
 3 str(clementines)
```

```
'data.frame':   15 obs. of  20 variables:
 $ a1 : num  18.6 37.6 71.6 94.2 100.2 ...
 $ a2 : num  17 38.2 67.8 106.8 64.2 ...
 $ a3 : num  19 36.2 90.4 110.9 83.4 ...
 $ a4 : num  6 48.6 77 115.5 94.1 ...
 $ a5 : num  15.8 43.6 81.6 133 87.6 ...
 $ a6 : num  0 22.8 36.6 111.2 54.8 ...
 $ a7 : num  6.2 31 62 101.5 66.8 ...
 $ a8 : num  5 30.2 31.1 89.7 53.5 ...
 $ a9 : num  7.2 27 65 124.1 104.9 ...
 $ a10: num  0 25.8 60.8 69.5 81.9 ...
 $ a11: num  8 19.4 60.2 102.7 56.5 ...
 $ a12: num  15 38 71.4 106.9 67.4
```

Categorical data

```r
library(fivethirtyeight)
str(food_world_cup[,1:12])
```

```
tibble [1,373 × 12] (S3: tbl_df/tbl/data.frame)
 $ respondent_id   : num [1:1373] 3308895255 3308891308 3308891135 3308879091
3308871671 ...
 $ knowledge       : Ord.factor w/ 4 levels "Novice"<"Intermediate"<..: 2 1 2 1 1 3 1
3 1 1 ...
 $ interest        : Ord.factor w/ 4 levels "Not at all"<"Not much"<..: 3 3 4 2 2 4 3
4 2 3 ...
 $ gender          : chr [1:1373] "Male" "Male" "Male" "Male" ...
 $ age             : Factor w/ 4 levels "18-29","30-44",..: 1 1 2 3 2 2 3 3 2 NA ...
 $ household_income: Factor w/ 5 levels "$0 - $24,999",..: 4 4 3 1 2 3 NA 1 3 NA ...
 $ education       : Ord.factor w/ 5 levels "Less than high school degree"<..: 1 3 5
1 2 5 2 3 3 NA ...
 $ location        : chr [1:1373] "West South Central" "West South Central" "Pacific"
```

Two geoms for bar charts

- Binned data (has a count column) `geom_col()`
- Unbinned data (no count column) `geom_bar()`

`geom_col()`

- Requires an **x** and **y**
- Intended to be used with one **continuous** and one **discrete** variables but other combinations may also work

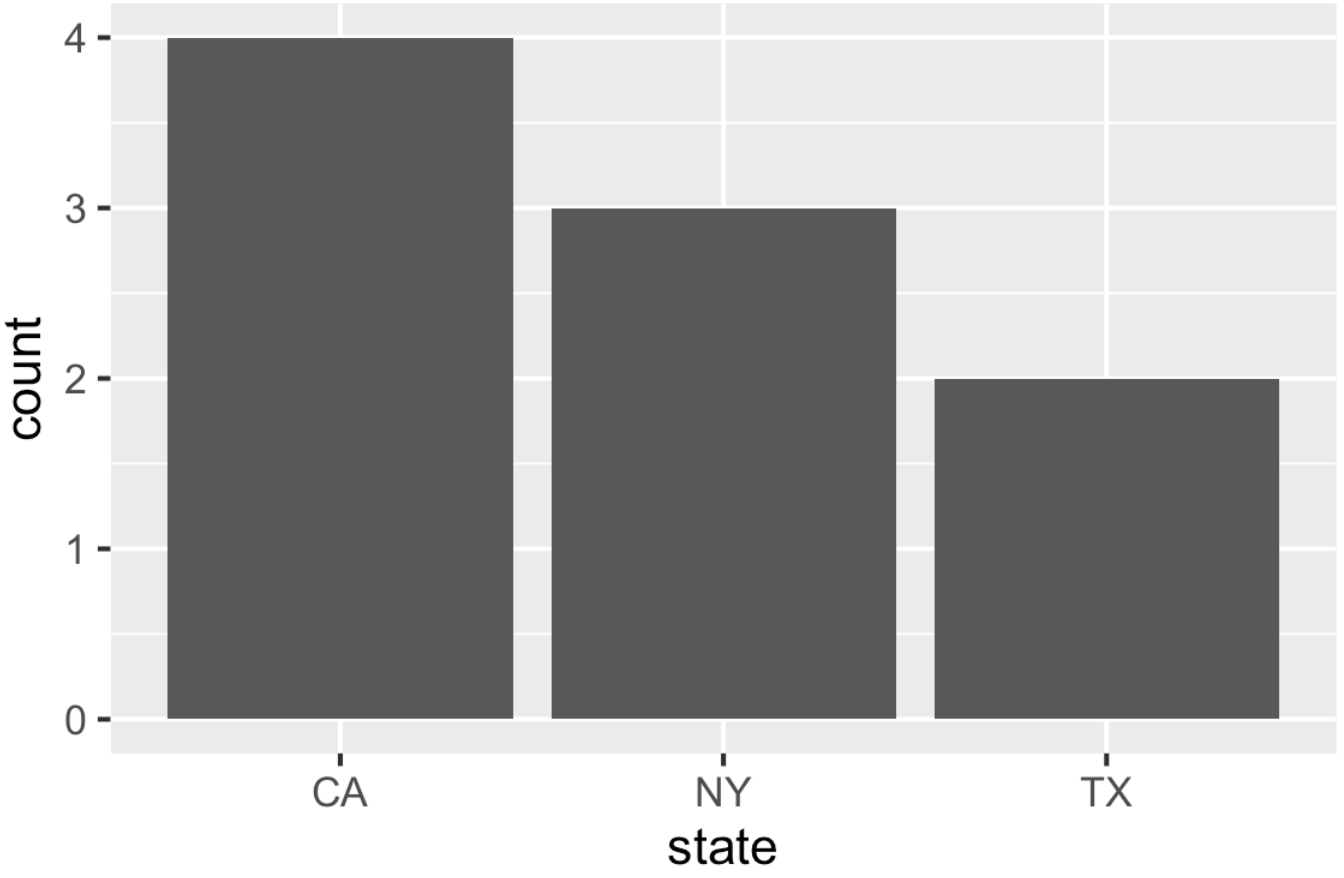
Look at the data

```
1 df_binned
```

```
  state count
1    CA     4
2    NY     3
3    TX     2
```

Bar chart with binned data

```
1 ggplot(df_binned, aes(x = state, y = count)) +
2   geom_col()
```

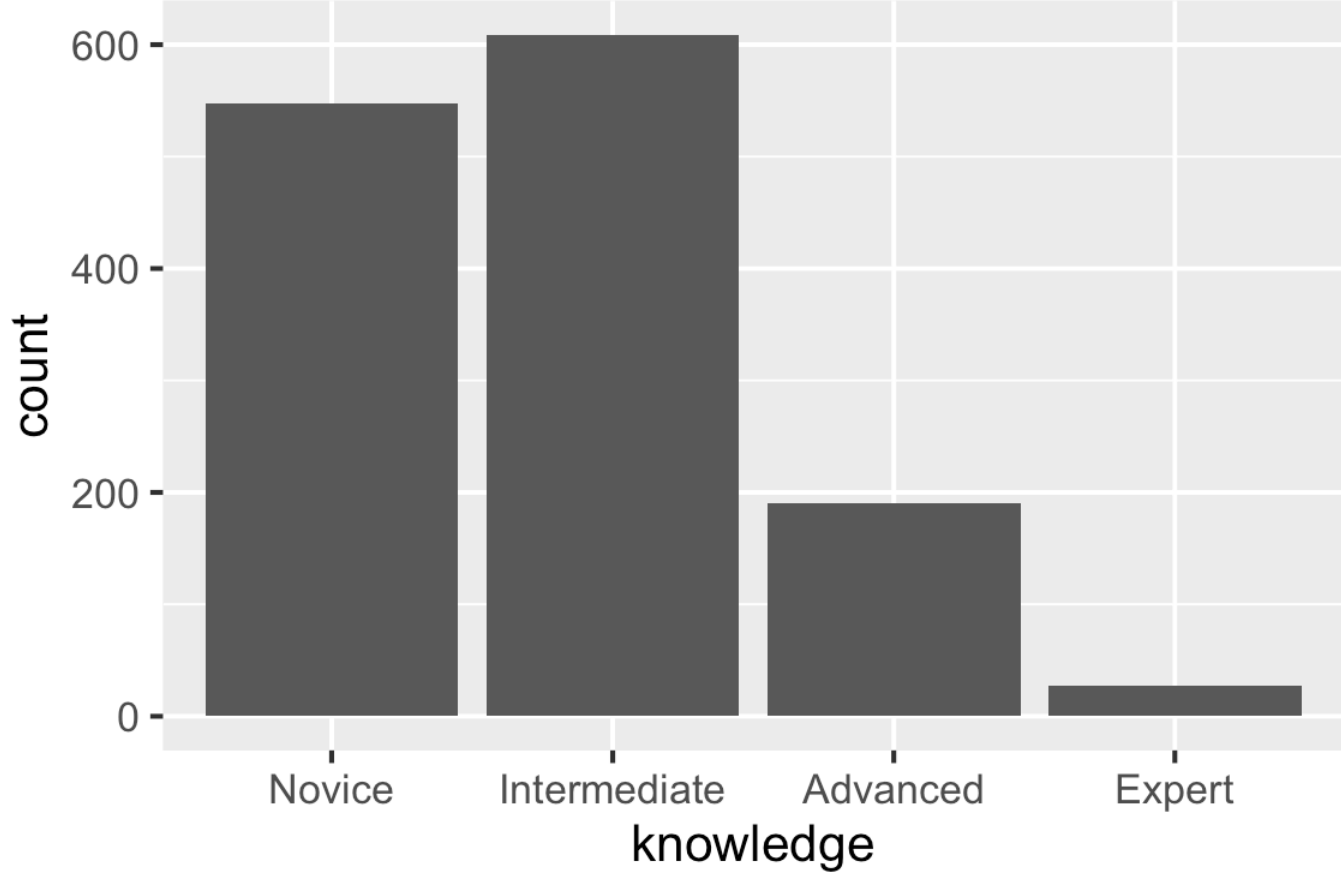

4
3
2
1
0
count
CA
NY
TX
state

`geom_bar()`

- Requires an **x** or **y**
- Intended to be used with one **discrete** variable (unbinned data)

Bar chart with unbinned data

```
1 ggplot(food_world_cup, aes(x = knowledge)) +
2   geom_bar()
```



Bar order

Claus Wilke
@ClausWilke

The answer to all ggplot2 questions on stackoverflow: "You need to turn the variable into a factor and then order the levels in the order you want the bars to be drawn."

10:19 PM · Feb 5, 2018

2 10 80

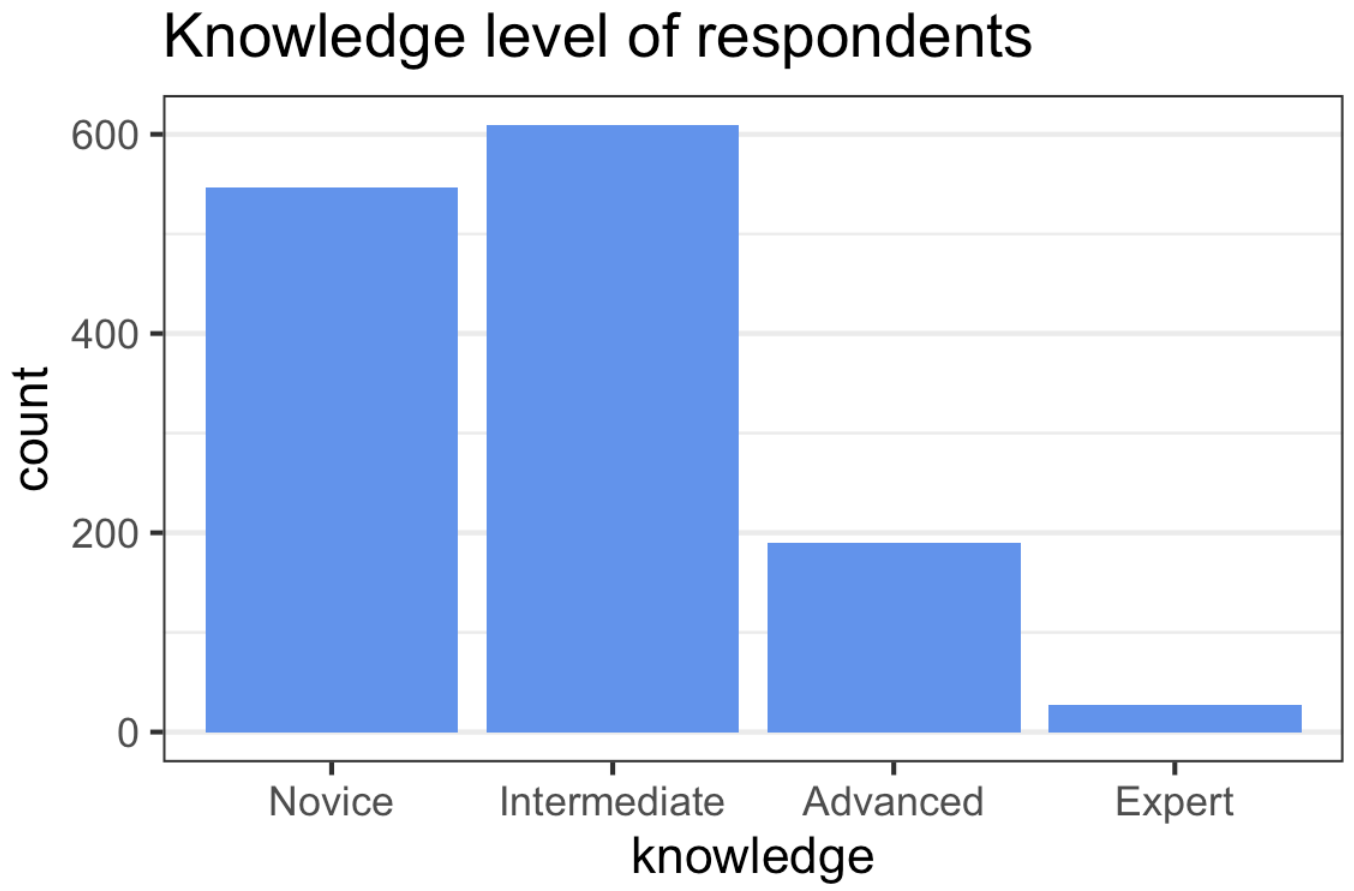
Types of data

- nominal does not have a fixed category order
- ordinal does have a fixed category order
- (“real”) discrete, small number of possibilities
- Not always clearcut: nominal vs. ordinal, ordinal vs. discrete, etc.
- Sometimes numbers = nominal, not discrete

Ordinal data

Sort in logical order of the categories (left to right)

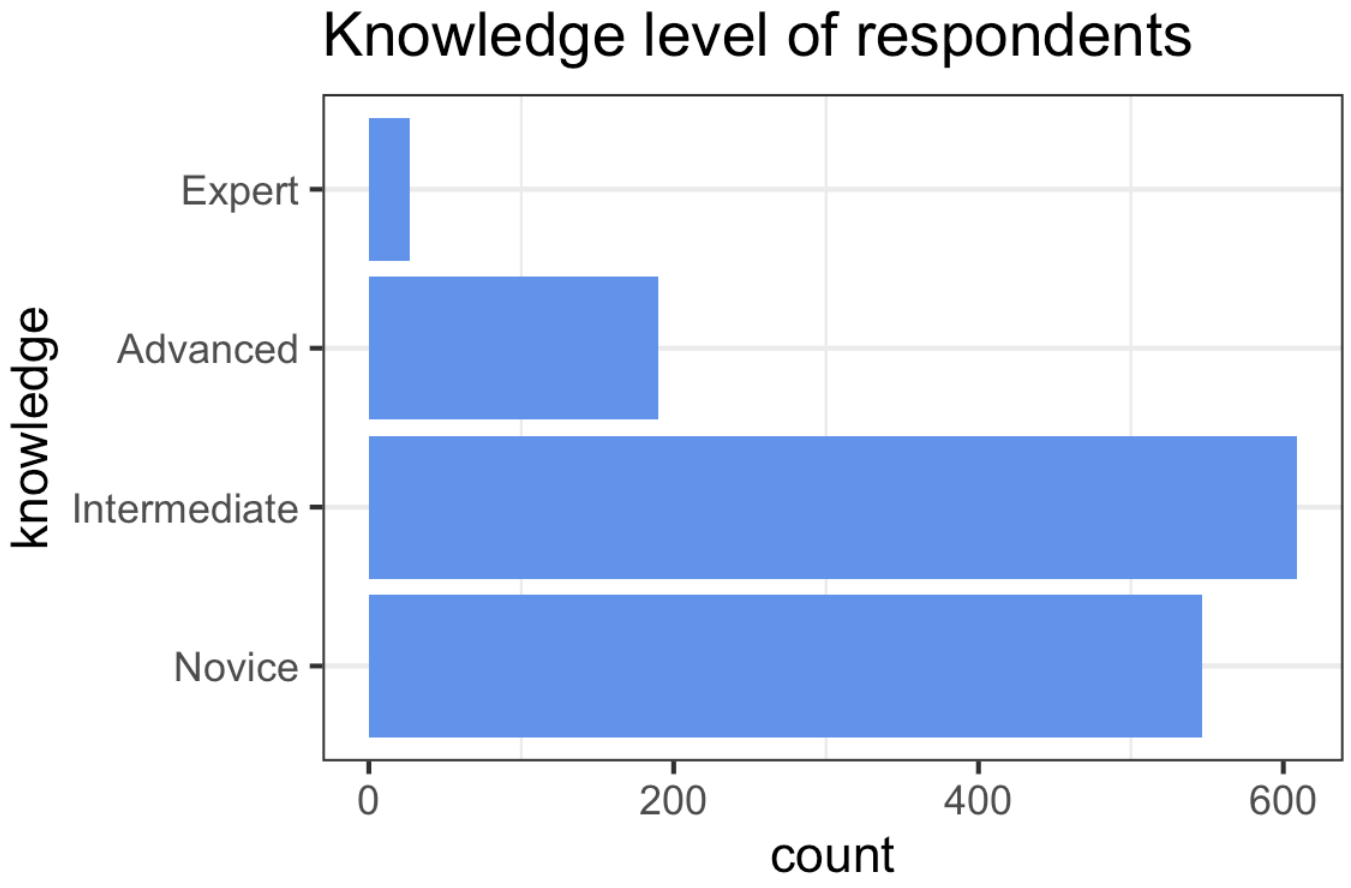
```
ggplot(food_world_cup, aes(knowledge)) +
    geom_bar(fill = "cornflowerblue") +
    labs(title ="Knowledge level of respondents") +
    theme_bw(16) +
    theme(panel.grid.major.x = element_blank())
```


Knowledge level of respondents
600
400
200
0
count
Novice
Intermediate
Advanced
Expert
knowledge

Ordinal data, horizontal bars

Sort in logical order of the categories (starting at bottom OR top)

```
ggplot(food_world_cup, aes(y = knowledge)) +
    geom_bar(fill = "cornflowerblue") +
    labs(title ="Knowledge level of respondents") +
    theme_bw(16) +
    theme(panel.grid.major.x = element_blank())
```

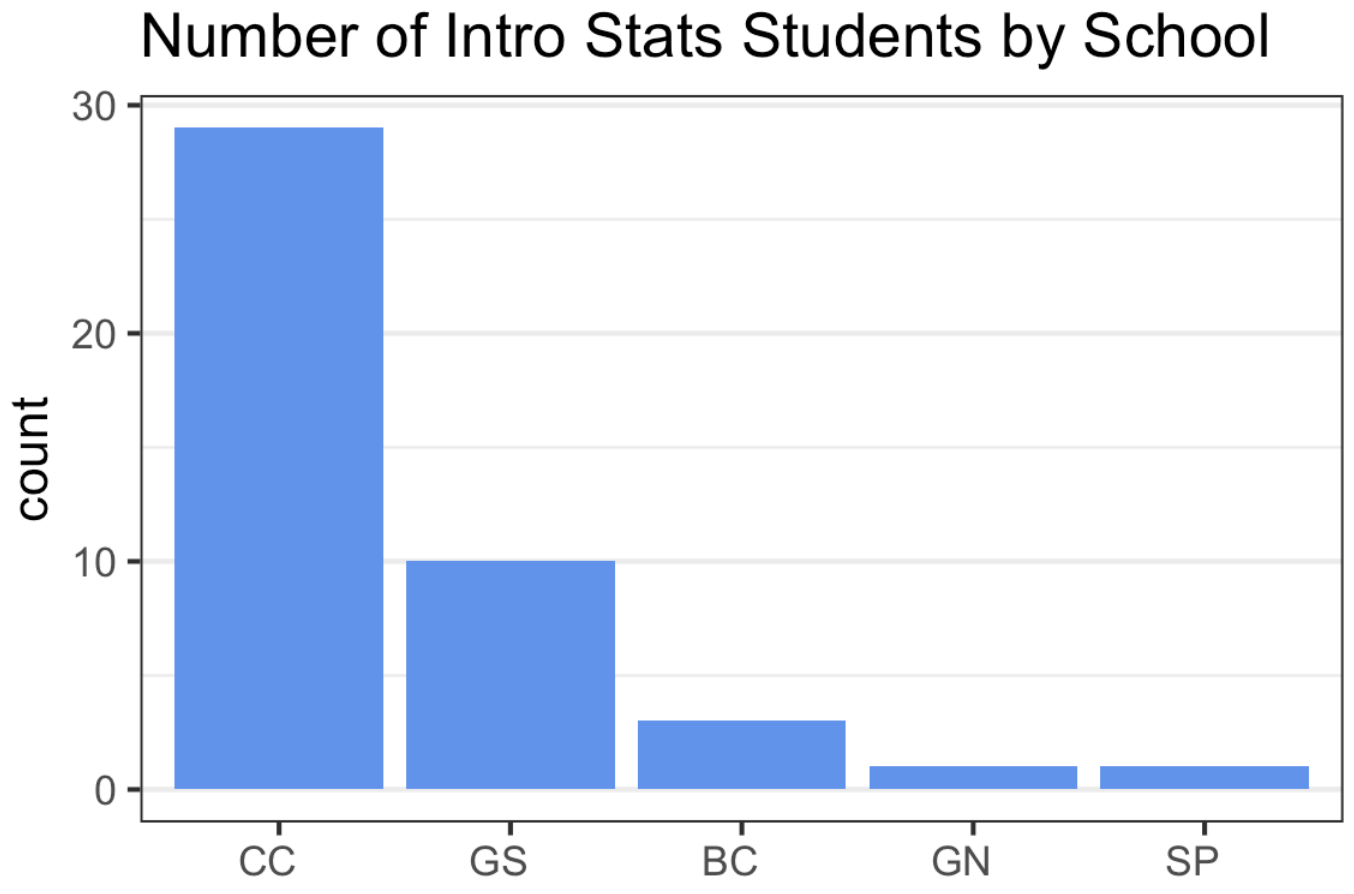

Knowledge level of respondents
knowledge
Expert
Advanced
Intermediate
Novice
0
200
400
600
count

Nominal data, vertical bars

Sort from highest to lowest count (left to right, or top to bottom)

```
student <- read.csv("student_data.csv")
## See "School Codes and Descriptions" in SSOL help menu

ggplot(student, aes(x = fct_infreq(School))) +
    geom_bar(fill = "cornflowerblue") +
    labs(title = "Number of Intro Stats Students by School", x = NULL) +
    theme_bw(16) +
    theme(panel.grid.major.x = element_blank())
```

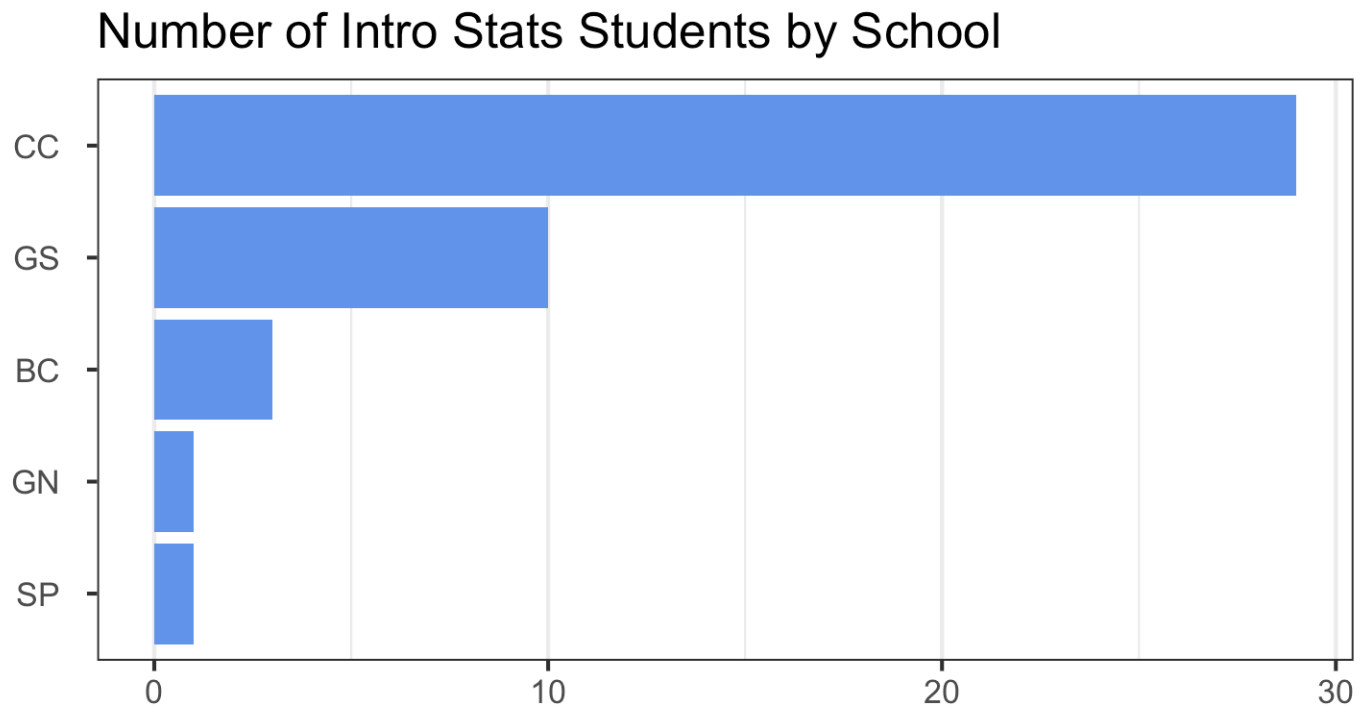

Number of Intro Stats Students by School
count
30
20
10
0
CC
GS
BC
GN
SP

Nominal data, horizontal bars

… or top to bottom

```
student$School <- fct_recode(student$School,
                             `Barnard College`= "BC",
                             `Columbia College` = "CC",
                             `General Studies Post Bac` = "GN",
                             `General Studies` = "GS",
                             `School of Professional Studies` = "SP")


ggplot(student, aes(y = fct_rev(fct_infreq(School)))) +
  geom_bar(fill = "cornflowerblue") +
  labs(title = "Number of Intro Stats Students by School", x = NULL, y = NULL) +
  theme_bw(16) +
  theme(panel.grid.major.y = element_blank())
```


Number of Intro Stats Students by School
CC
GS
BC
GN
SP
0
10
20
30

Useful forcats functions for rdering bars

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

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

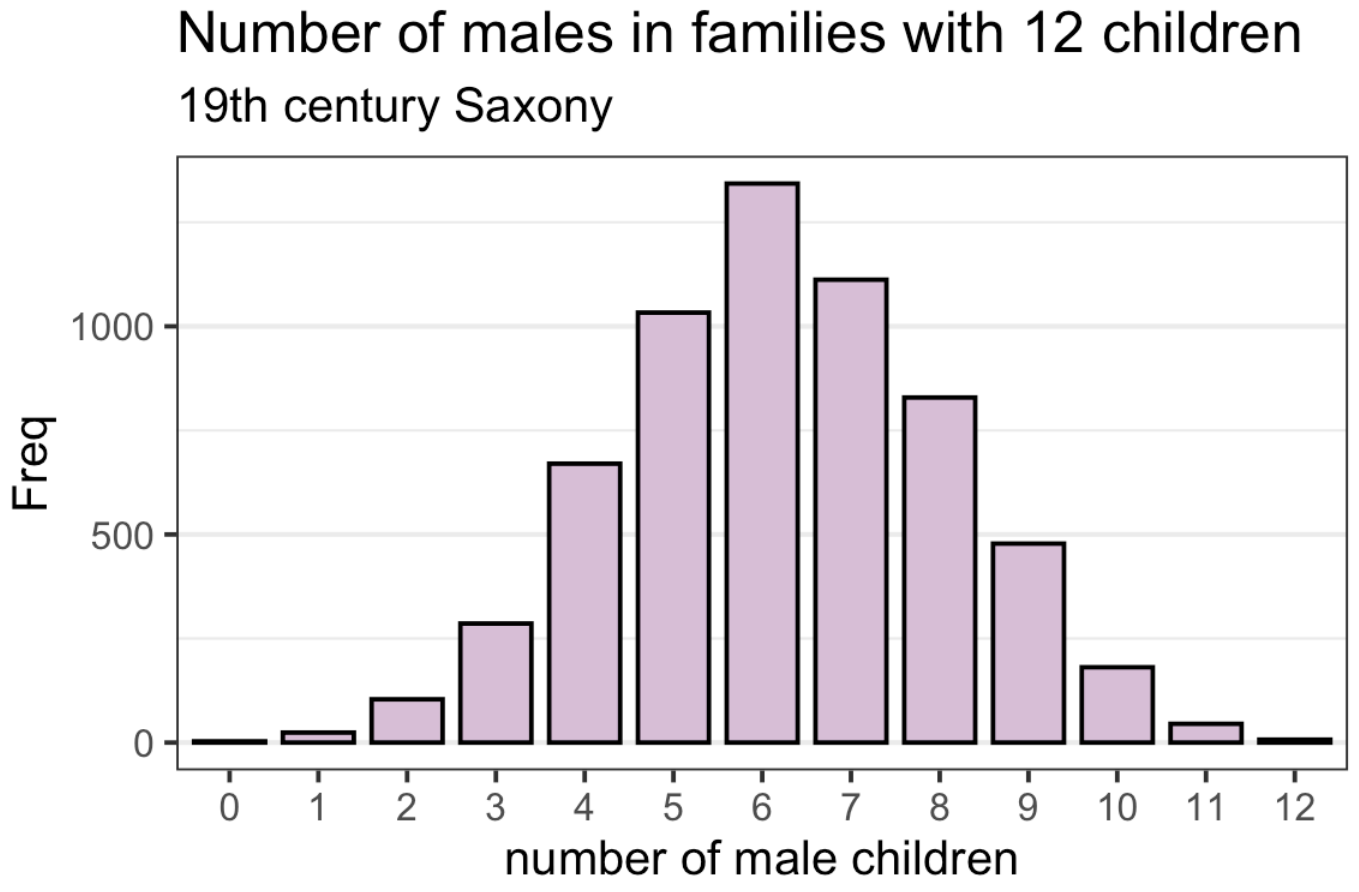
`fct_reorder(x, y)` – reorder `x` by `y`

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

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

Discrete data

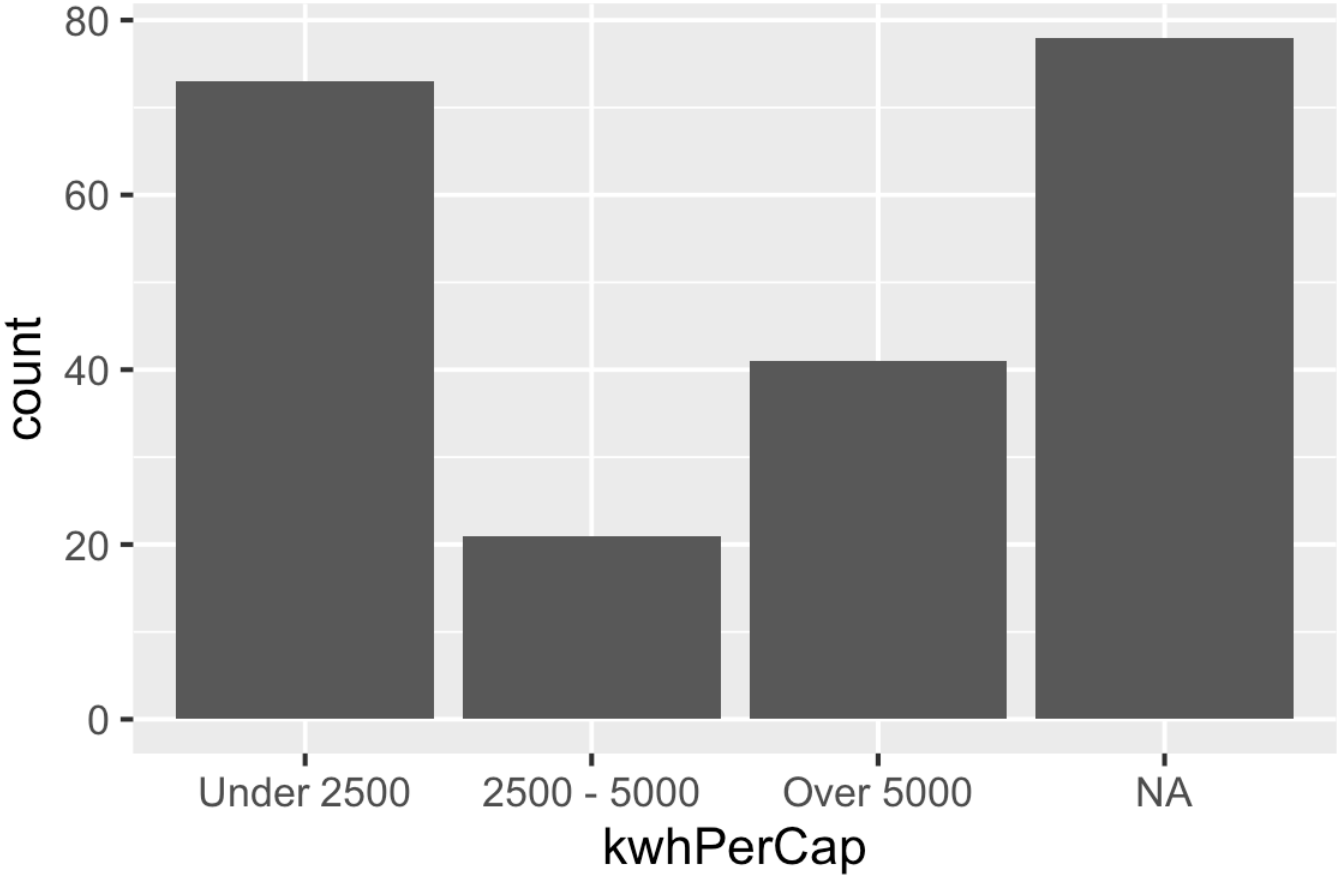
```
library(vcd)
df <- data.frame(Saxony)
ggplot(df, aes(x = nMales, y = Freq)) +
  geom_col(color = "black", fill = "thistle", width = .8) +
  labs(title = "Number of males in families with 12 children", subtitle = "19th ce
       x = "number of male children") +
  theme_bw(15) +
  theme(panel.grid.major.x = element_blank())
```


Number of males in families with 12 children
19th century Saxony
Freq
1000
500
0
0
1
2
3
4
5
6
7
8
9
10
11
12
number of male children

Missing values

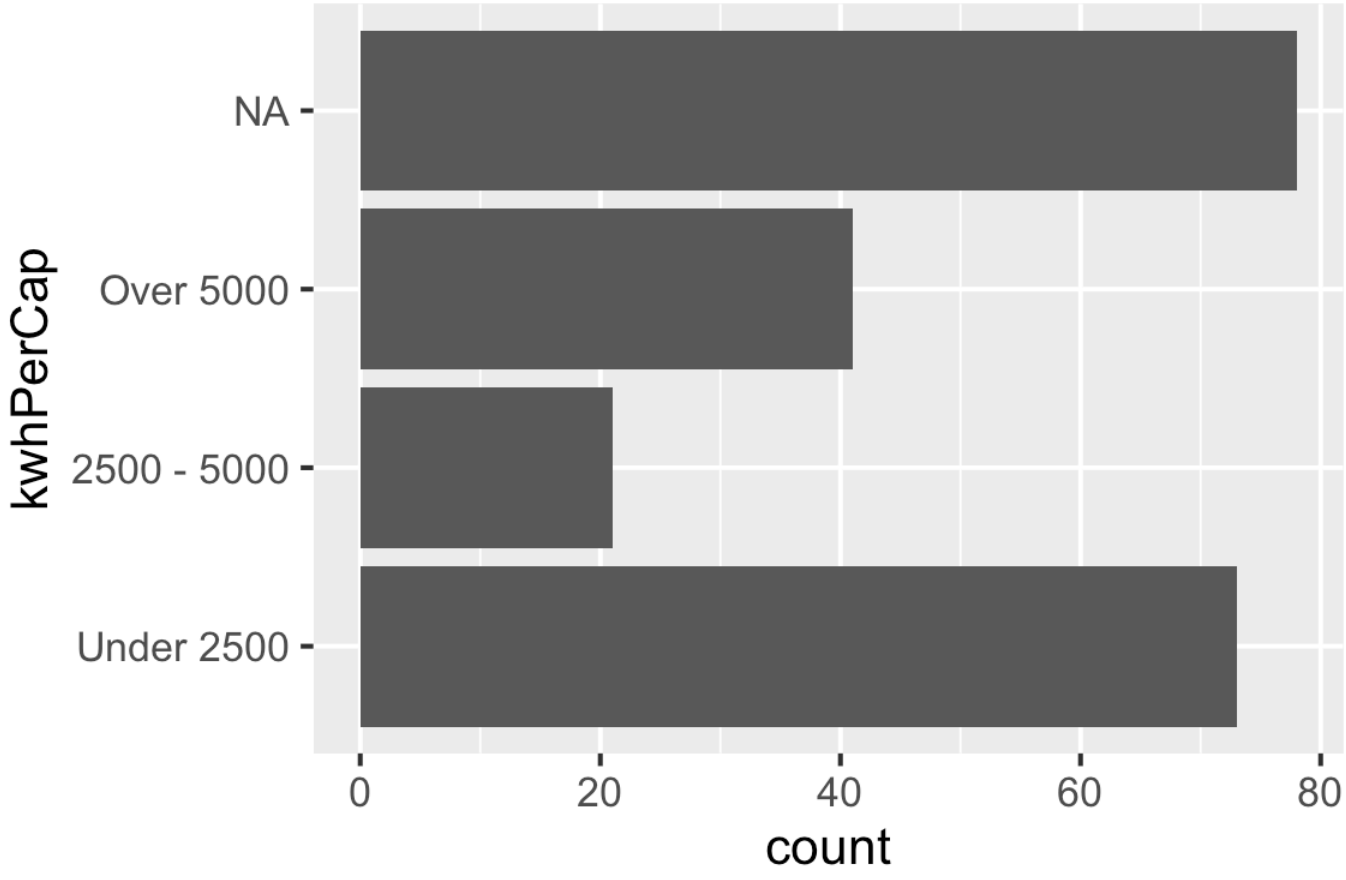
Discrete (categorical) default in ggplot2: keep NAs

```
1 library(Lock5withR)
2 ggplot(AllCountries, aes(x = kwhPerCap)) +
3   geom_bar()
```


80
60
40
20
0
count
Under 2500
2500 - 5000
Over 5000
NA
kwhPerCap

Discrete (categorical) default in ggplot2: keep NAs

```
1 ggplot(AllCountries, aes(y = kwhPerCap)) +
2   geom_bar()
```

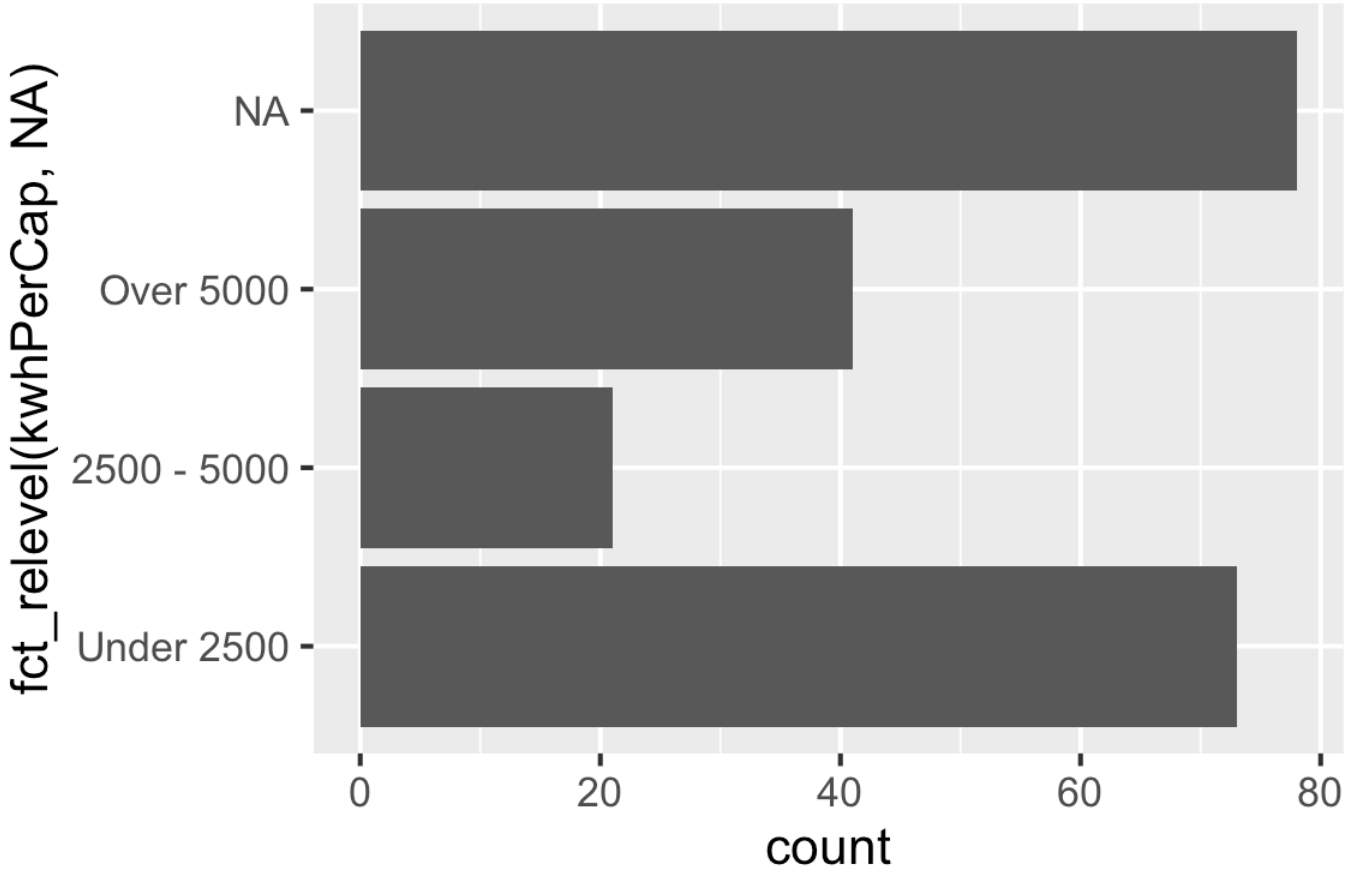


The NA bar is too prominent… let’s move it to the bottom.

Rearranging factor levels with NAs

```
1 ggplot(AllCountries, aes(y = fct_relevel(kwhPerCap, NA))) +
2   geom_bar()
```

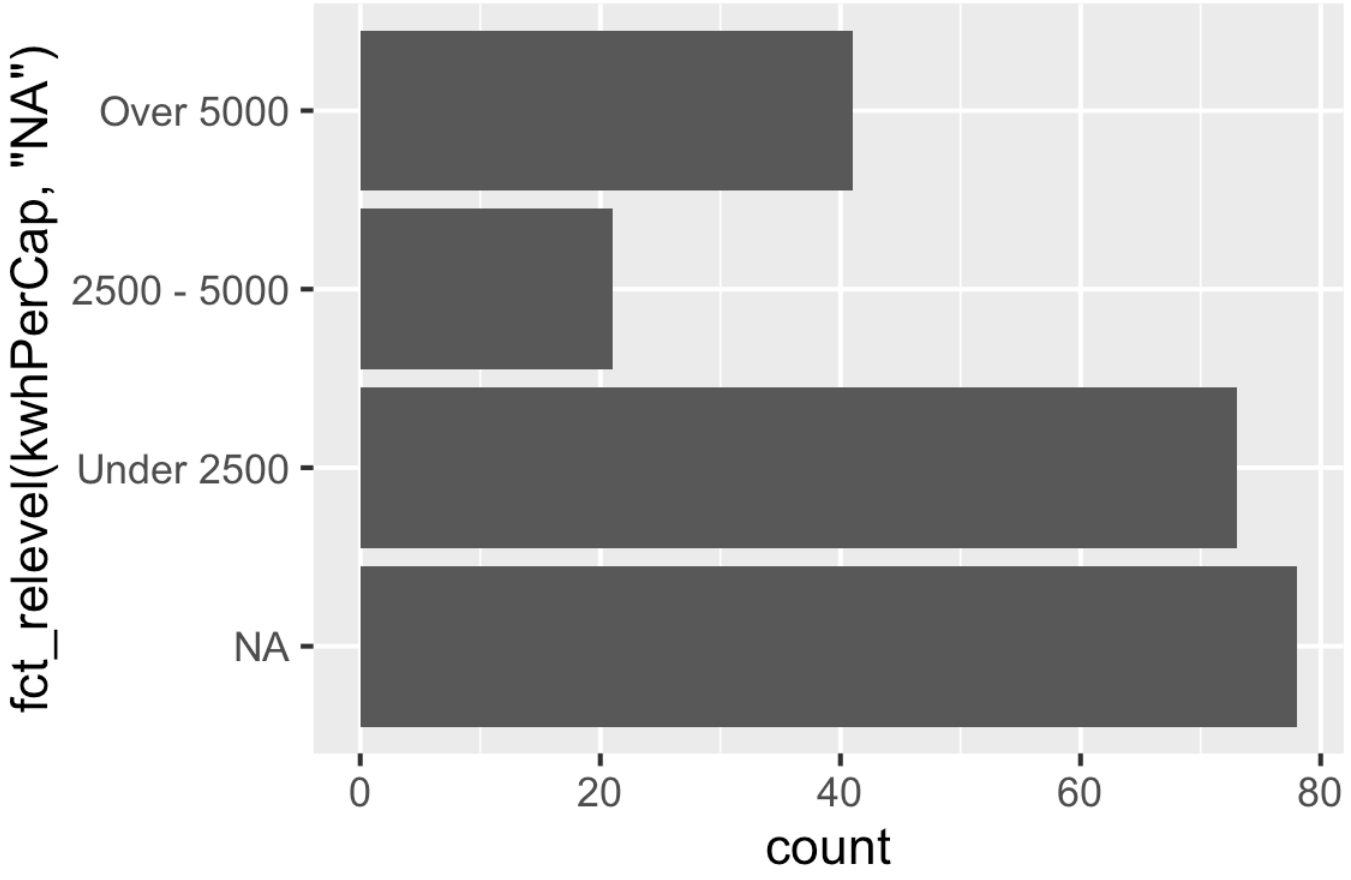
```
Warning: 1 unknown level in `f`: NA
```



Oops, that didn’t work.

Making NA explicit (a.k.a. NA value -> NA level)

```
1 AllCountries |>
2   mutate(kwhPerCap = fct_explicit_na(kwhPerCap, "NA")) |>
3   ggplot(aes(y = fct_relevel(kwhPerCap, "NA"))) +
4   geom_bar()
```


fct_relevel(kwhPerCap, "NA")
Over 5000
2500 - 5000
Under 2500
NA
0
20
40
60
80
count

Why not always make NAs explicit?

(That is, make them factor levels)

```
animal <- factor(c(NA, "ant", "ant", NA, "bee", "cat", NA))
animal
```

```
[1] <NA> ant  ant  <NA> bee  cat  <NA>
Levels: ant bee cat
```

```
is.na(animal)
```

```
[1]  TRUE FALSE FALSE  TRUE FALSE FALSE  TRUE
```

```
animal2 <- fct_explicit_na(animal, "NA")
animal2
```

```
[1] NA  ant ant NA  bee cat NA
Levels: ant bee cat NA
```

```
is.na(animal2)
```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

Option 2: remove NA's

```
1 df <- data.frame(col1 = 1:4, col2 = c(NA, 5:7), col3 = c(8:10, NA))
2 df
```

```
  col1 col2 col3
1    1   NA    8
2    2    5    9
3    3    6   10
4    4    7   NA
```

Remove all rows with *any* NAs (keep complete cases):

```
1 df |> na.omit() # base R
```

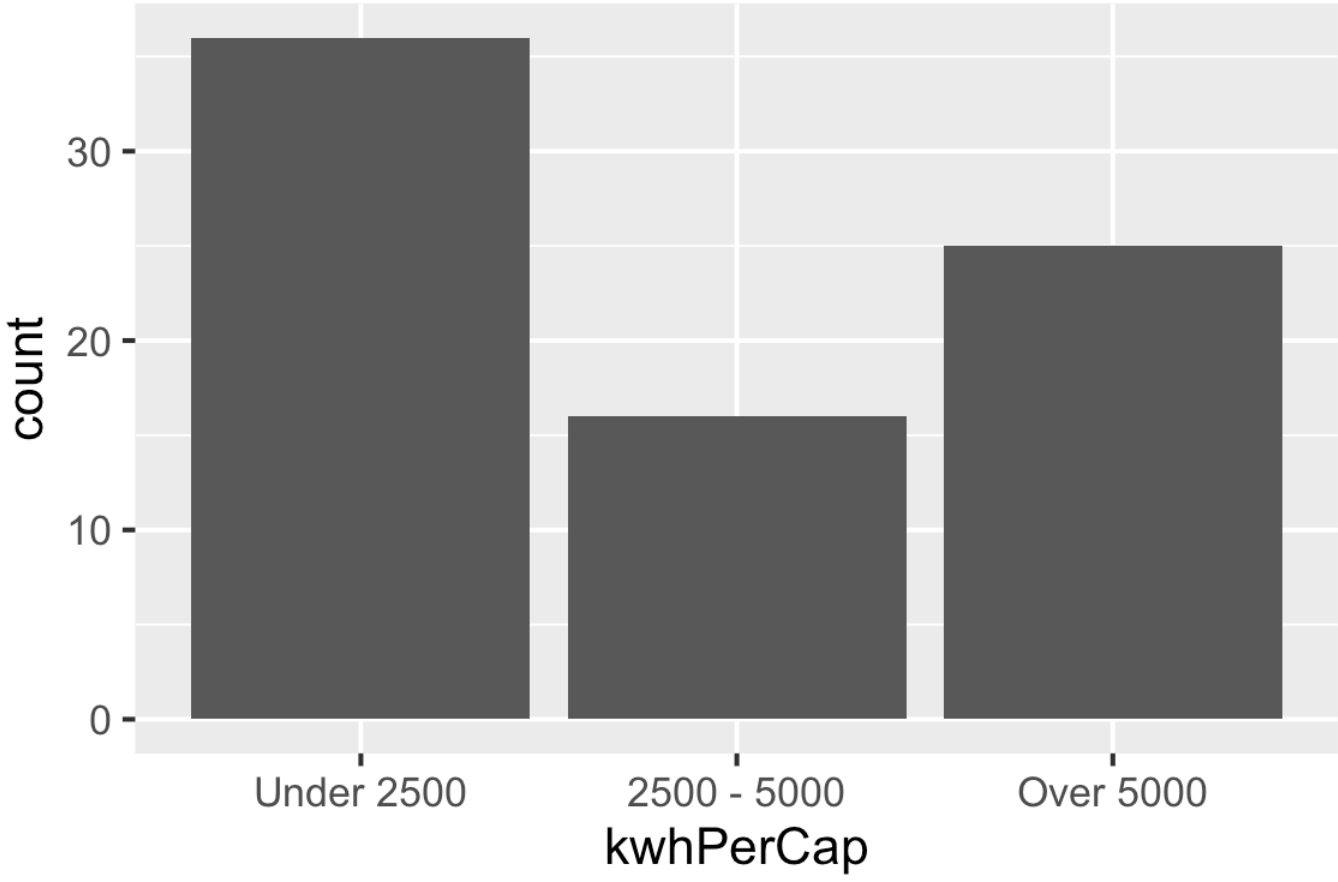
```
  col1 col2 col3
2    2    5    9
3    3    6   10
```

```
1 df |> drop_na() # tidyverse
```

```
  col1 col2 col3
1    2    5    9
2    3    6   10
```

Removing NAs

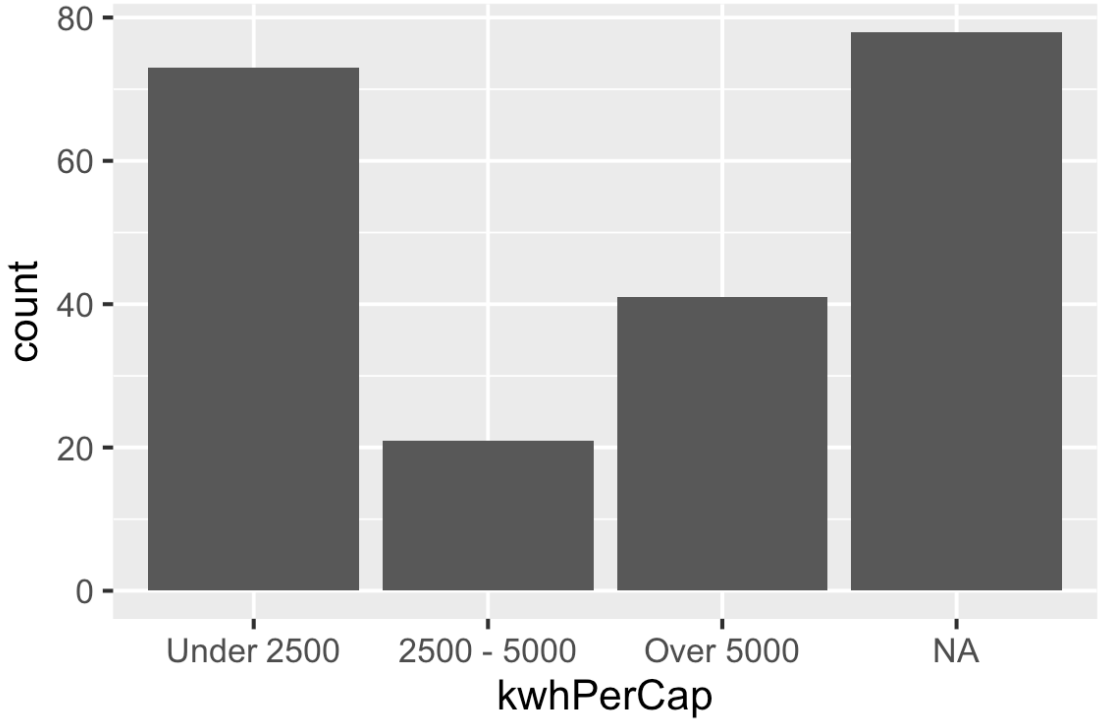
```
1 AllCountries |>
2   drop_na() |>
3   ggplot(aes(x = kwhPerCap)) +
4   geom_bar()
```



What’s wrong??

Compare

```
ggplot(AllCountries, aes(x = kwhPerCap)) +
  geom_bar()
AllCountries |>
  drop_na() |>
  ggplot(aes(x = kwhPerCap)) +
  geom_bar()
```


80
60
40
20
0
count
Under 2500
2500 - 5000
Over 5000
NA
kwhPerCap


30
20
10
0
count
Under 2500
2500 - 5000
Over 5000
kwhPerCap

Only remove NAs from some columns

Remove rows with NAs in a particular column or columns:

```
1 df
```

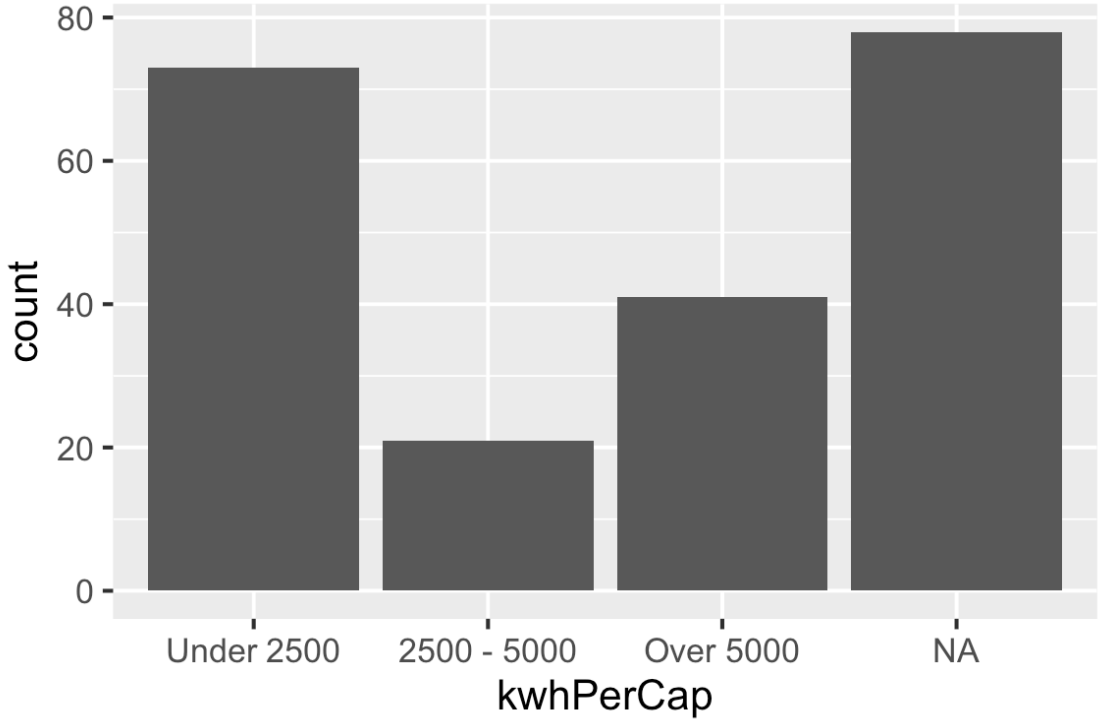
```
  col1 col2 col3
1    1   NA    8
2    2    5    9
3    3    6   10
4    4    7   NA
```

```
1 df |> drop_na(col2)
```

```
  col1 col2 col3
1    2    5    9
2    3    6   10
3    4    7   NA
```

Only remove NAs from some columns

```
ggplot(AllCountries, aes(x = kwhPerCap)) +
  geom_bar()
AllCountries |>
  drop_na(kwhPerCap) |>
  ggplot(aes(x = kwhPerCap)) +
  geom_bar()
```

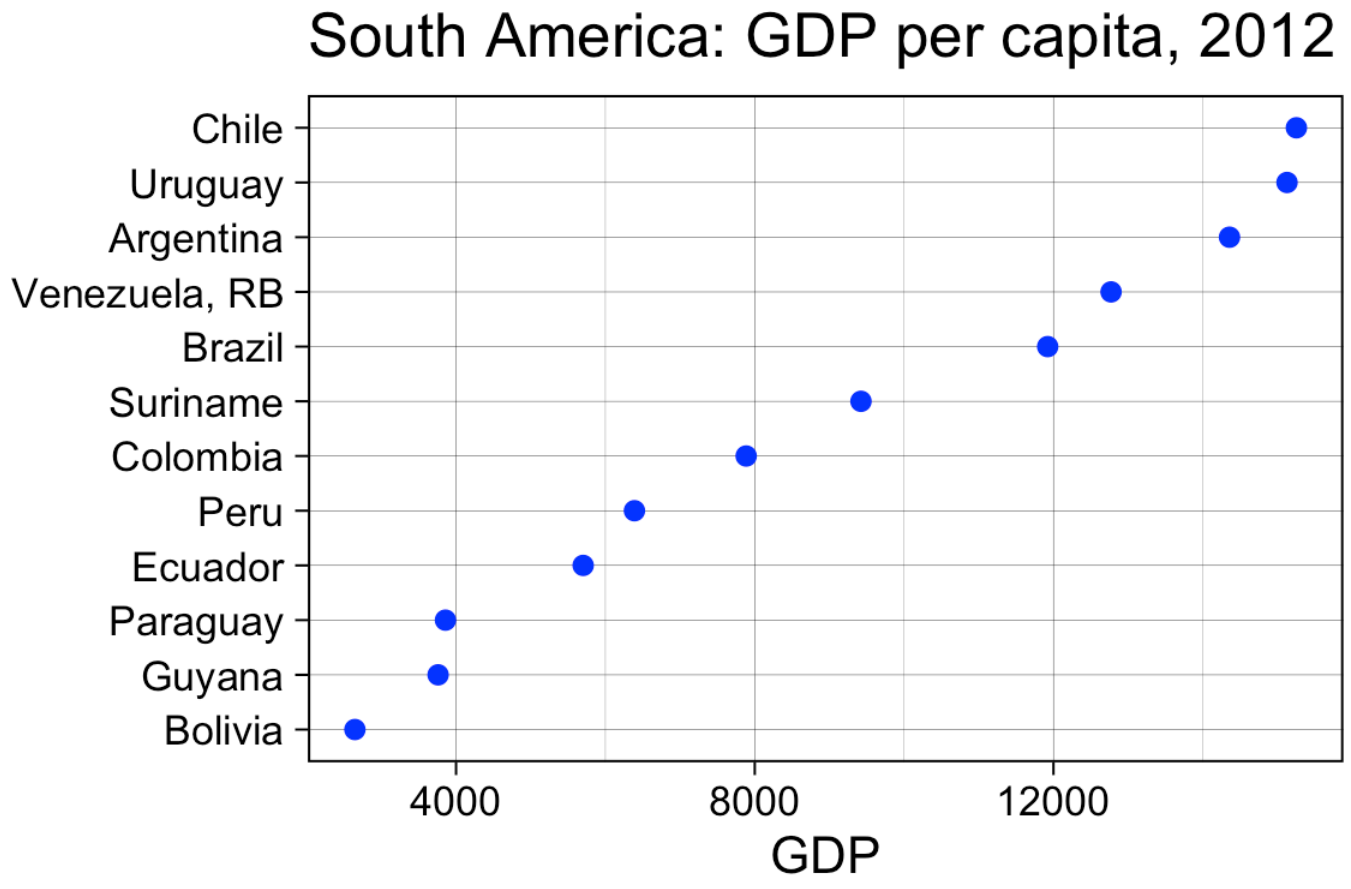

80
60
40
20
0
count
Under 2500
2500 - 5000
Over 5000
NA
kwhPerCap


60
40
20
0
count
Under 2500
2500 - 5000
Over 5000
kwhPerCap

Cleveland dot plot

Cleveland dot plot

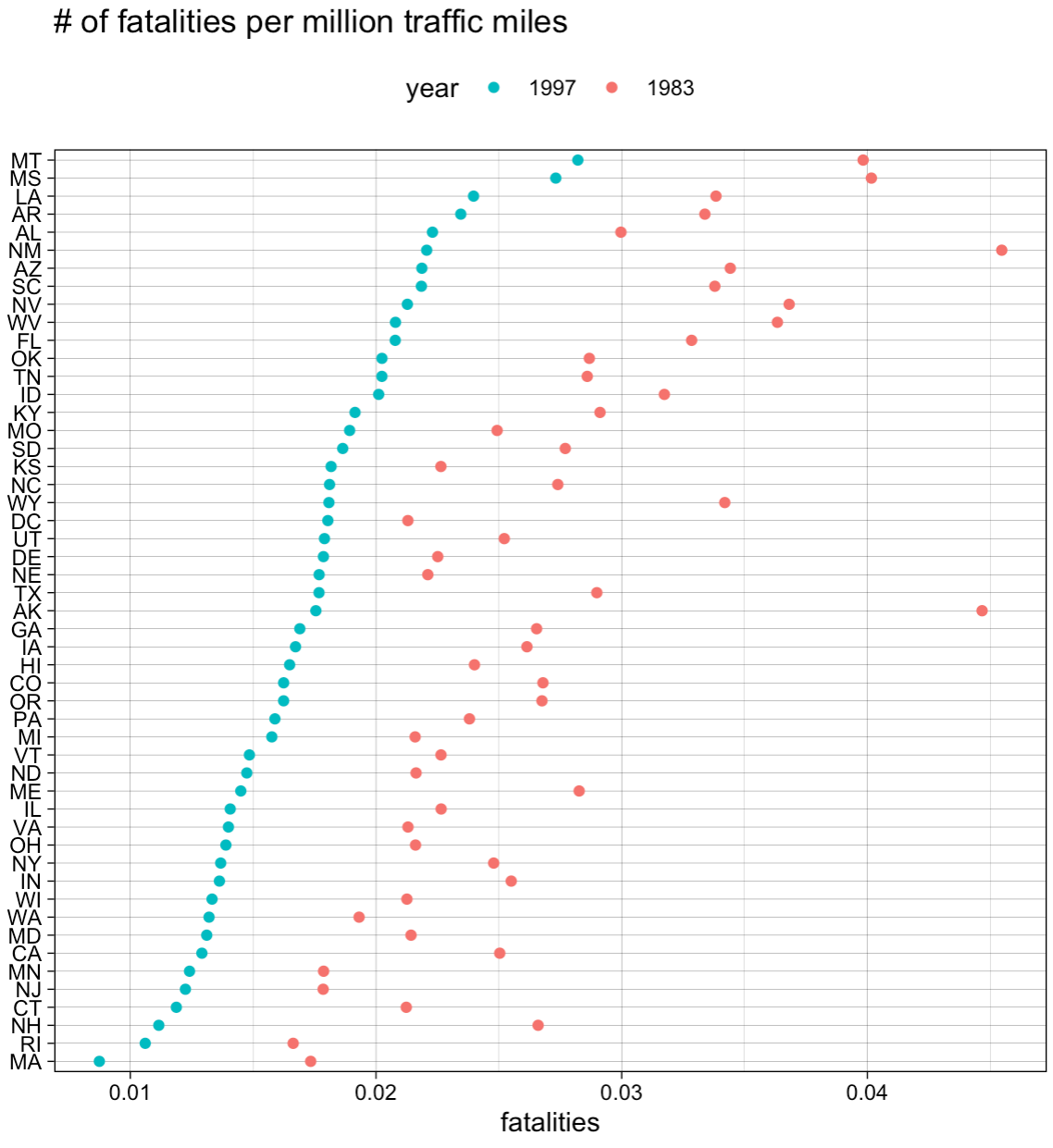
```
world <- read_csv("countries2012.csv")
sa <- world |>
  filter(CONTINENT == "South America")
ggplot(sa, aes(x = GDP, y = fct_reorder(COUNTRY, GDP))) +
  geom_point(color = "blue") +
  labs(title = "South America: GDP per capita, 2012", y = NULL) +
  theme_linedraw(16) ## works well for dotplots
```


South America: GDP per capita, 2012
Chile
Uruguay
Argentina
Venezuela, RB
Brazil
Suriname
Colombia
Peru
Ecuador
Paraguay
Guyana
Bolivia
4000
8000
12000
GDP

Cleveland dot plot with multiple dots

```r
library(AER)
data("USSeatBelts")
belts <- USSeatBelts |>
  filter(year %in% c(1983, 1997)) |>
  select(state, year, fatalities)

## `fct_reorder2` --double sort: year, then fatalities
ggplot(belts, aes(x = fatalities,
                  y = fct_reorder2(state, year == 1997, fatalities, .desc = FALSE)
                  color = year)) +
  geom_point() +
  labs(title = "# of fatalities per million traffic miles", y = NULL) +
  guides(color = guide_legend(reverse=TRUE)) +
  theme_linedraw() +
  theme(legend.position = "top")
```


# of fatalities per million traffic miles
year
1997
1983
MT
MS
LA
AR
AL
NM
AZ
SC
NV
WV
FL
OK
TN
ID
KY
MO
SD
KS
NC
WY
DC
UT
DE
NE
TX
AK
GA
IA
HI
CO
OR
PA
MI
VT
ND
ME
IL
VA
OH
NY
IN
WI
WA
MD
CA
MN
NJ
CT
NH
RI
MA
0.01
0.02
0.03
0.04
fatalities