

# Complex recoding with `case_when`

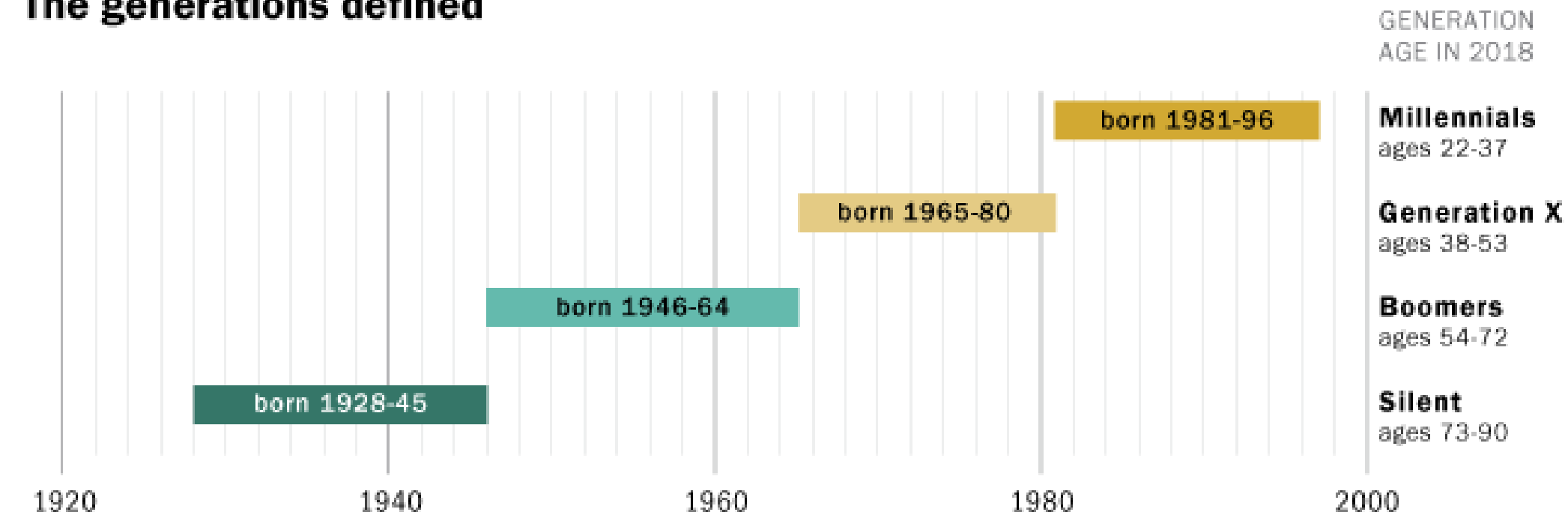
WORKING WITH DATA IN THE TIDYVERSE



**Alison Hill**  
Professor & Data Scientist

# Generations & age

## The generations defined



PEW RESEARCH CENTER

<sup>1</sup> <http://www.pewresearch.org/topics/generations> <sup>2</sup> and <sup>3</sup> age/

?case\_when

## Usage

```
case_when( ... )
```

### Arguments

...

A sequence of two-sided formulas. The left hand side (LHS) determines which values match this case. The right hand side (RHS) provides the replacement value.

The LHS must evaluate to a logical vector. Each logical vector can either have length 1 or a common length. All RHSs must evaluate to the same type of vector.

These dots are evaluated with [explicit splicing](#).

# Bakers

bakers

```
# A tibble: 10 x 2
  baker    birth_year
  <chr>      <dbl>
1  Liam      1998.
2  Martha    1997.
3  Jason     1992.
4  Stuart    1986.
5  Manisha   1985.
6  Simon     1980.
7  Natasha   1976.
8  Richard   1976.
9  Robert    1959.
10 Diana    1945.
```

# Simple `if\_else`

```
bakers %>%  
  mutate(gen = if_else(between(birth_year, 1981, 1996),  
                        "millennial", "not millennial"))
```

```
# A tibble: 10 x 3  
  baker    birth_year gen  
  <chr>      <dbl> <chr>  
1 Liam      1998. not millennial  
2 Martha    1997. not millennial  
3 Jason     1992. millennial  
4 Stuart    1986. millennial  
5 Manisha   1985. millennial  
6 Simon     1980. not millennial  
7 Natasha   1976. not millennial  
8 Richard   1976. not millennial  
9 Robert    1959. not millennial  
10 Diana    1945. not millennial
```

# Multiple `if\_else` pairs

```
bakers %>%  
  mutate(gen = case_when(  
    between(birth_year, 1965, 1980) ~ "gen_x",  
    between(birth_year, 1981, 1996) ~ "millennial"))
```

```
# A tibble: 10 x 3  
  baker  birth_year gen  
  <chr>    <dbl> <chr>  
1 Liam    1998. NA  
2 Martha  1997. NA  
3 Jason   1992. millennial  
4 Stuart  1986. millennial  
5 Manisha 1985. millennial  
6 Simon   1980. gen_x  
7 Natasha 1976. gen_x  
8 Richard 1976. gen_x  
9 Robert  1959. NA  
10 Diana  1945. NA
```

# Make multiple bins

```
bakers %>% mutate(gen = case_when(  
  between(birth_year, 1928, 1945) ~ "silent",  
  between(birth_year, 1946, 1964) ~ "boomer",  
  between(birth_year, 1965, 1980) ~ "gen_x",  
  between(birth_year, 1981, 1996) ~ "millenial",  
  TRUE ~ "gen_z"))
```

```
# A tibble: 10 x 3  
  baker    birth_year gen  
  <chr>      <dbl> <chr>  
1 Liam      1998. gen_z  
2 Martha    1997. gen_z  
3 Jason     1992. millenial  
4 Stuart    1986. millenial  
5 Manisha   1985. millenial  
6 Simon     1980. gen_x  
7 Natasha   1976. gen_x  
8 Richard   1976. gen_x  
9 Robert    1959. boomer  
10 Diana    1945. silent
```

# List of "if-then" pairs

*if TRUE* ..... `bakers %>%  
 mutate(gen = case_when(  
 between(birth_year, 1928, 1945) ~ "silent",  
 between(birth_year, 1946, 1964) ~ "boomer",  
 between(birth_year, 1965, 1980) ~ "gen_x",  
 between(birth_year, 1981, 1996) ~ "millenial",  
 TRUE ~ "gen_z"  
 ))`

*then replace with*

↓



# The last "if-then" pair

```
bakers %>%  
  mutate(gen = case_when(  
    between(birth_year, 1928, 1945) ~ "silent",  
    between(birth_year, 1946, 1964) ~ "boomer",  
    between(birth_year, 1965, 1980) ~ "gen_x",  
    between(birth_year, 1981, 1996) ~ "millenial",  
    ELSE ..... TRUE ~ "gen_z"  
  ))
```

*replace with*

# Know your new variable!

bakers

```
# A tibble: 95 x 3
  baker      birth_year gen
  <chr>      <dbl> <chr>
1 Liam      1998. gen_z
2 Martha    1997. gen_z
3 Flora     1996. millenial
4 Michael   1996. millenial
5 Julia     1996. millenial
6 Ruby      1993. millenial
7 Benjamina 1993. millenial
8 Jason     1992. millenial
9 James     1991. millenial
10 Andrew   1991. millenial
# ... with 85 more rows
```

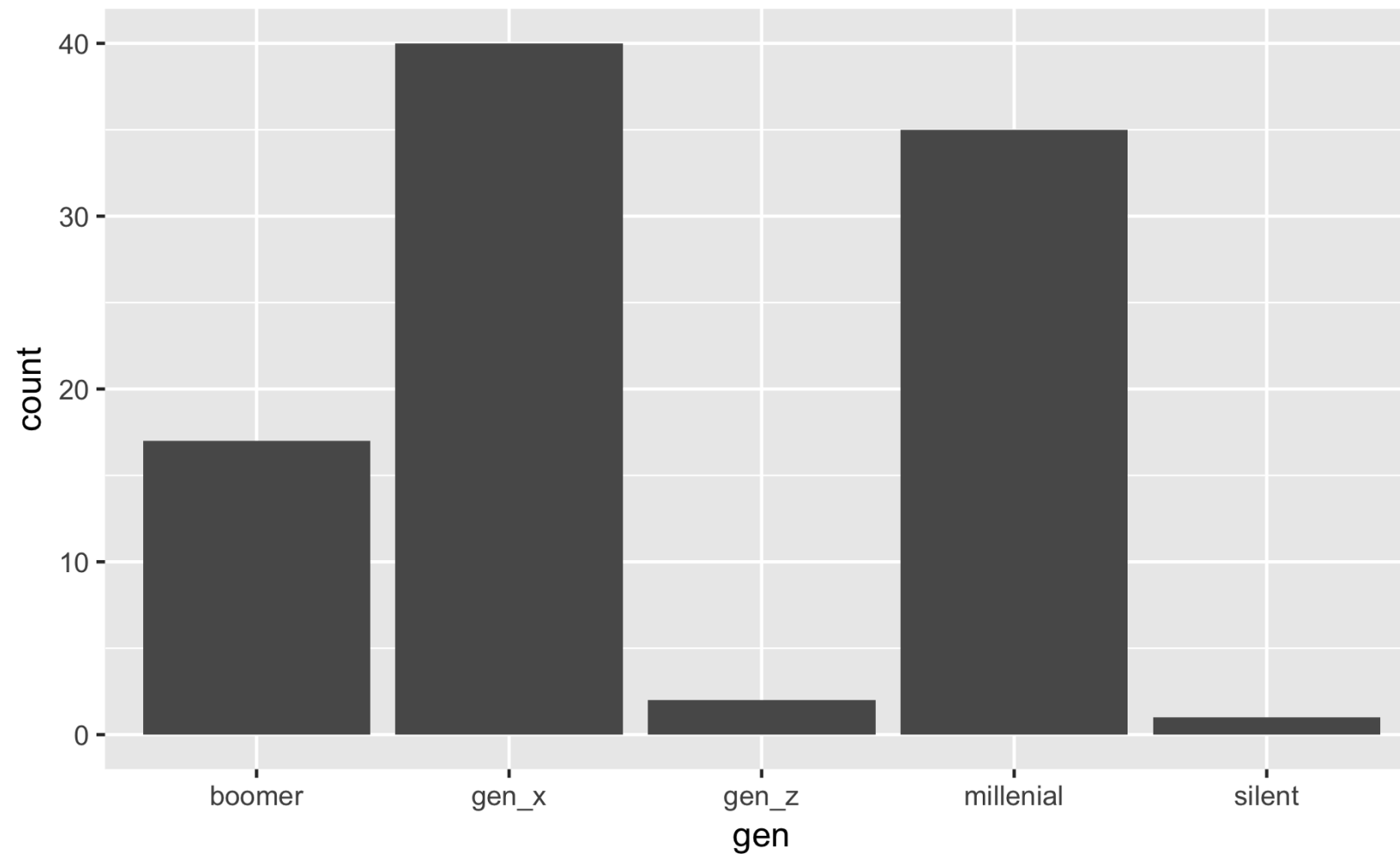
# Count bakers by generation

```
bakers %>%  
  count(gen, sort = TRUE) %>%  
  mutate(prop = n / sum(n))
```

```
# A tibble: 5 x 3  
  gen      n  prop  
  <chr> <int> <dbl>  
1 gen_x    40 0.421  
2 millenial 35 0.368  
3 boomer   17 0.179  
4 gen_z     2 0.0211  
5 silent    1 0.0105
```

# Plot bakers by generation

```
ggplot(bakers, aes(x = gen)) + geom_bar()
```



# Let's practice!

WORKING WITH DATA IN THE TIDYVERSE

# Factors

WORKING WITH DATA IN THE TIDYVERSE



**Alison Hill**

Professor & Data Scientist

# The `forcats` package

```
library(forcats) # once per work session
```



<sup>1</sup> <http://forcats.tidyverse.org>

# What is a factor?

"In R, factors are used to work with categorical variables, variables that have a fixed and known set of possible values."

<sup>1</sup> Garrett Grolmund & Hadley Wickham, <http://r4ds.had.co.nz/factors.html>



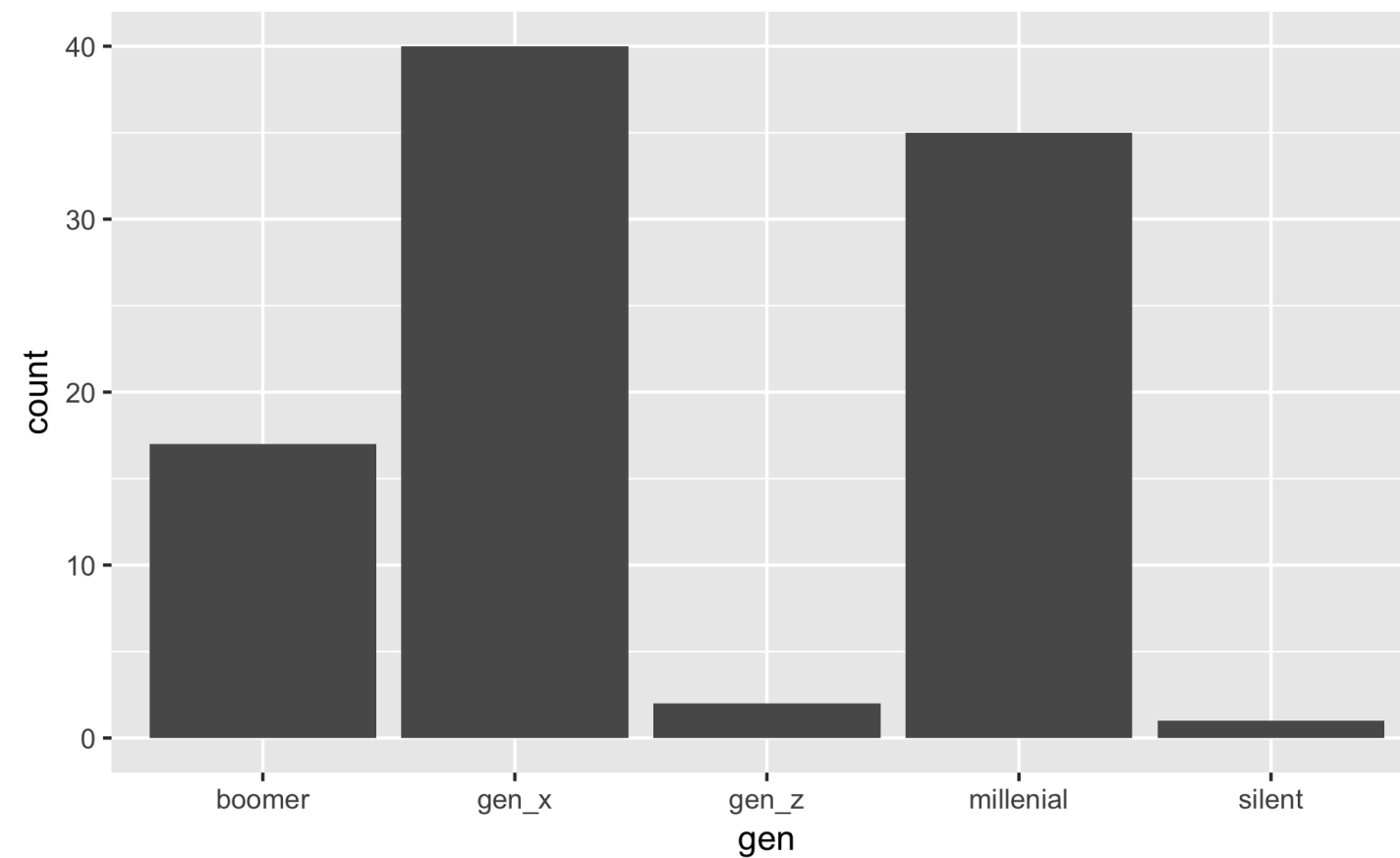
# Count bakers by generation

```
bakers %>%  
  count(gen, sort = TRUE) %>%  
  mutate(prop = n / sum(n))
```

```
# A tibble: 5 x 3  
  gen      n  prop  
  <chr> <int> <dbl>  
1 gen_x    40 0.421  
2 millenial 35 0.368  
3 boomer   17 0.179  
4 gen_z     2 0.0211  
5 silent    1 0.0105
```

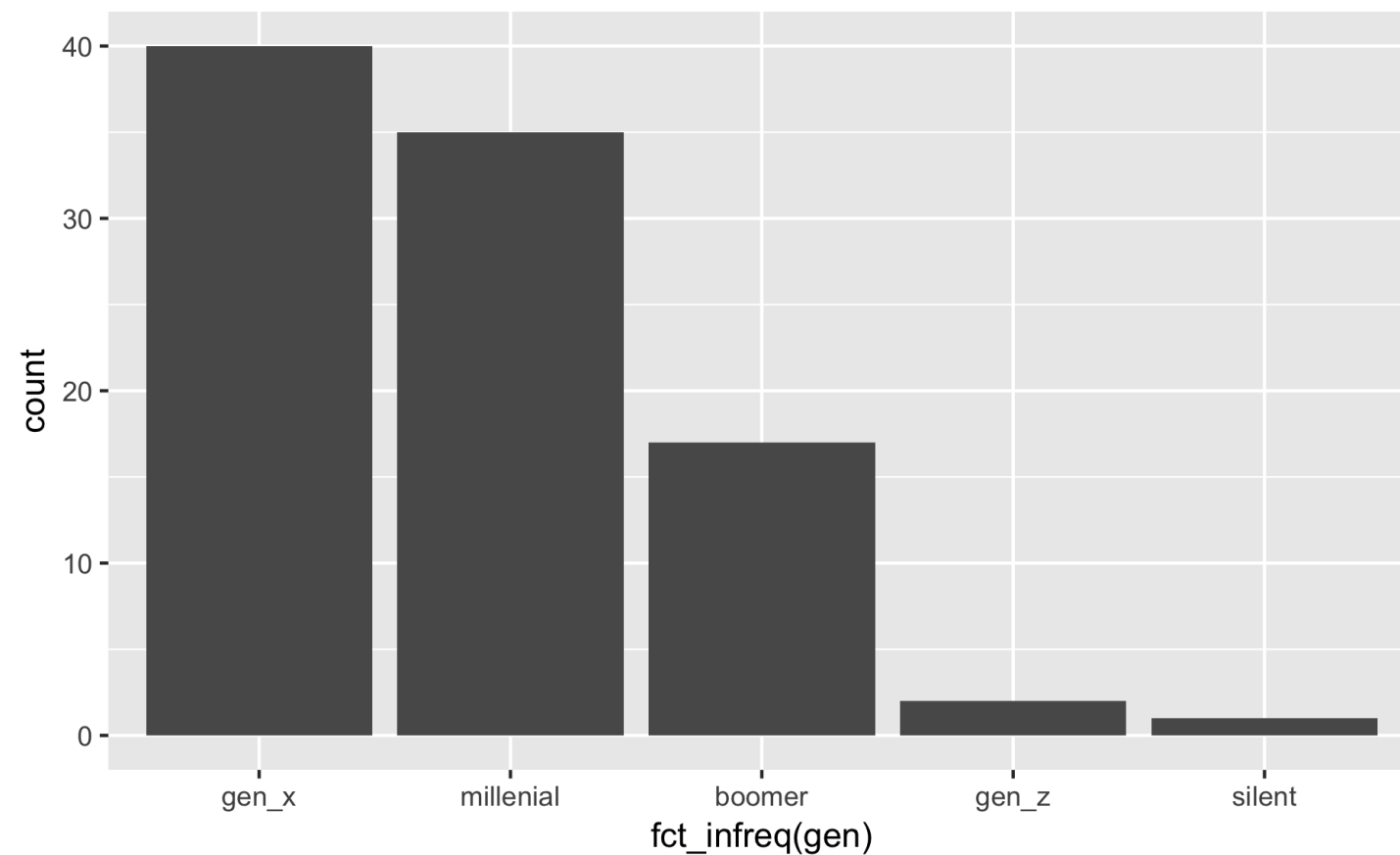
# Plot bakers by generation

```
ggplot(bakers, aes(x = gen)) +  
  geom_bar()
```



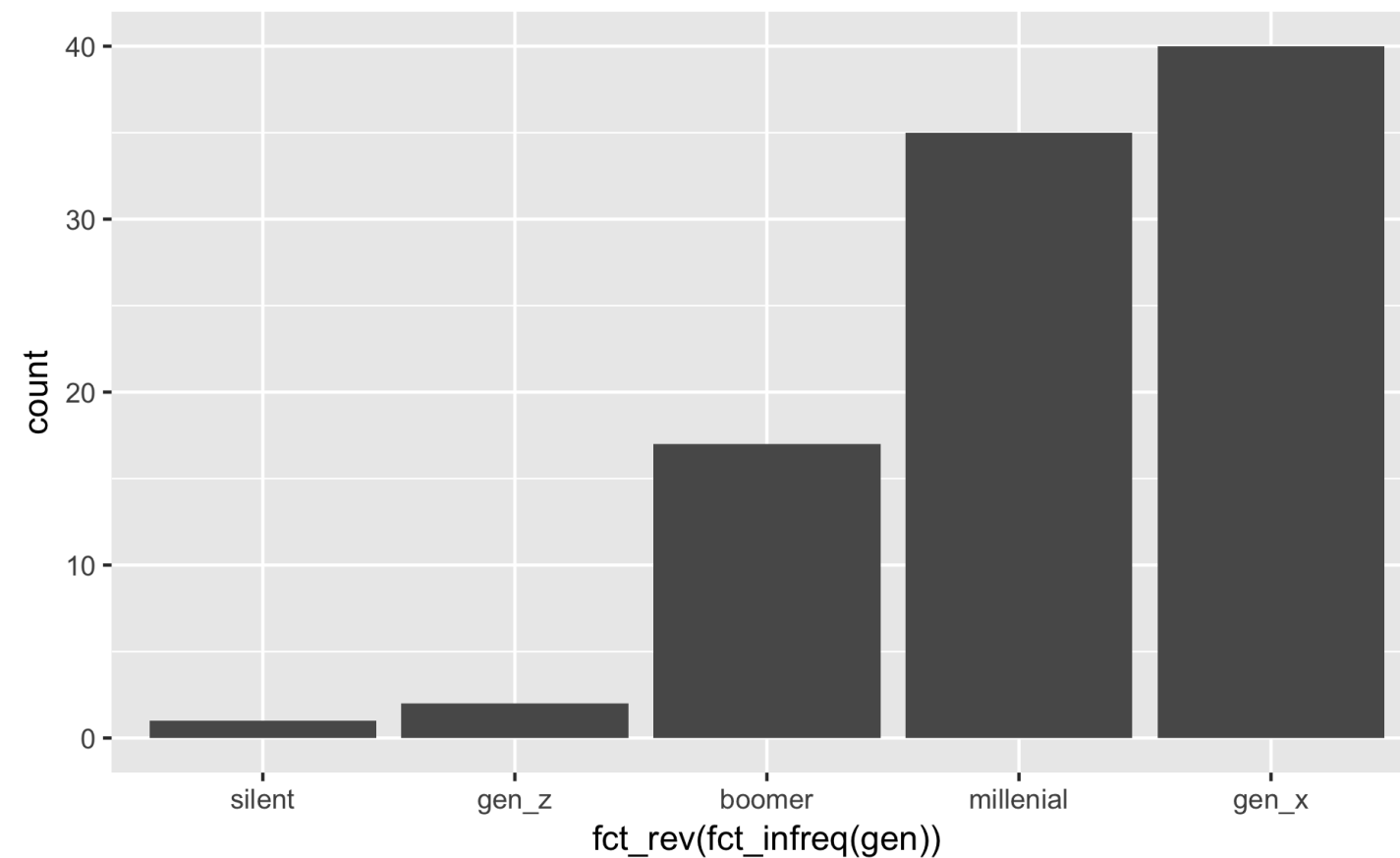
# Reorder from most to least bakers

```
ggplot(bakers, aes(x = fct_infreq(gen))) +  
  geom_bar()
```



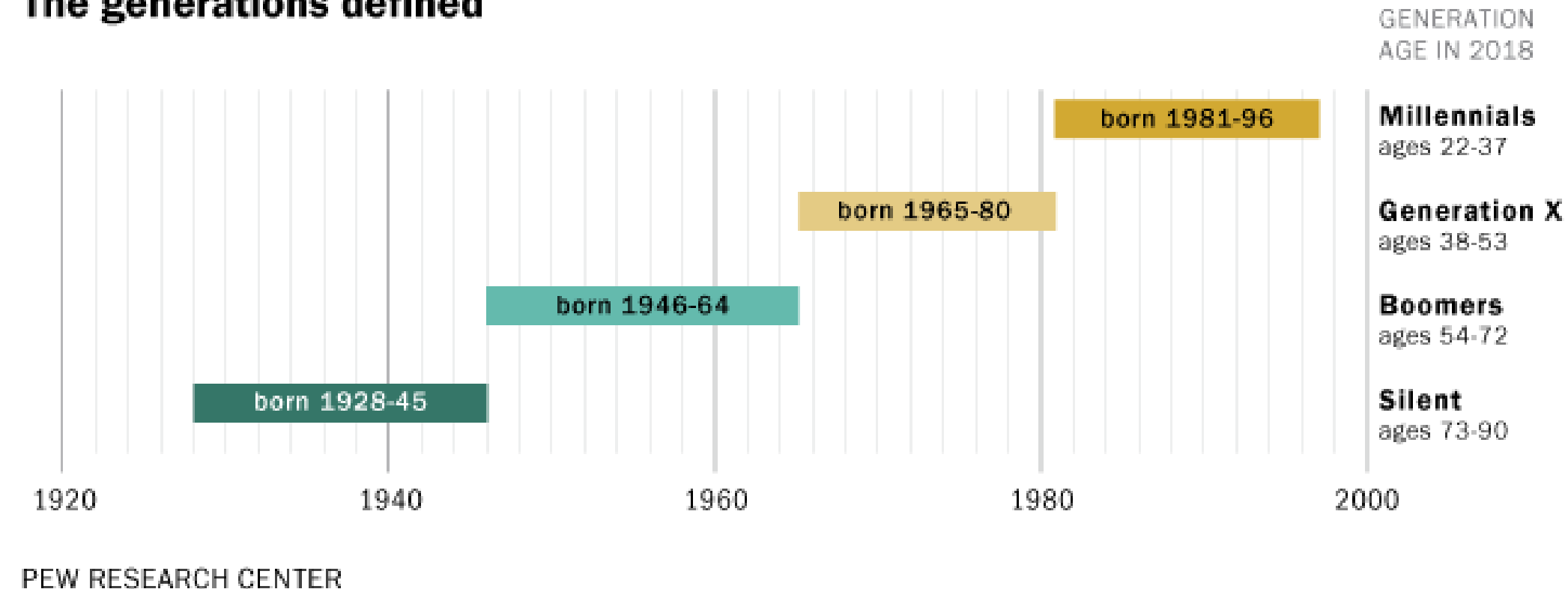
# Reorder from least to most bakers

```
ggplot(bakers, aes(x = fct_rev(fct_infreq(gen)))) +  
  geom_bar()
```



# Relevel using natural order

## The generations defined



<sup>1</sup> <http://www.pewresearch.org/topics/generations> <sup>2</sup> and <sup>3</sup> age/

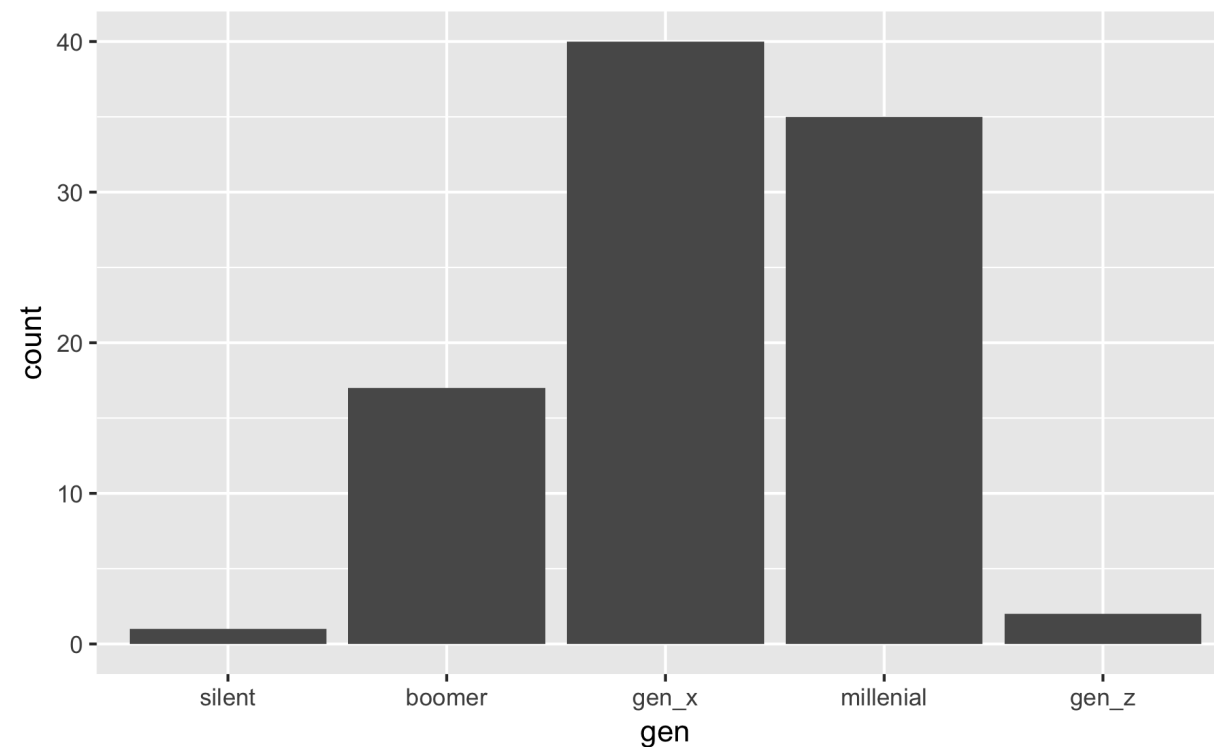
# Reorder by hand

```
bakers <- bakers %>%  
  mutate(gen = fct_relevel(gen, "silent", "boomer",  
                           "gen_x", "millenial", "gen_z"))  
  
bakers %>%  
  dplyr::pull(gen) %>%  
  levels()
```

```
"silent"    "boomer"    "gen_x"    "millenial" "gen_z"
```

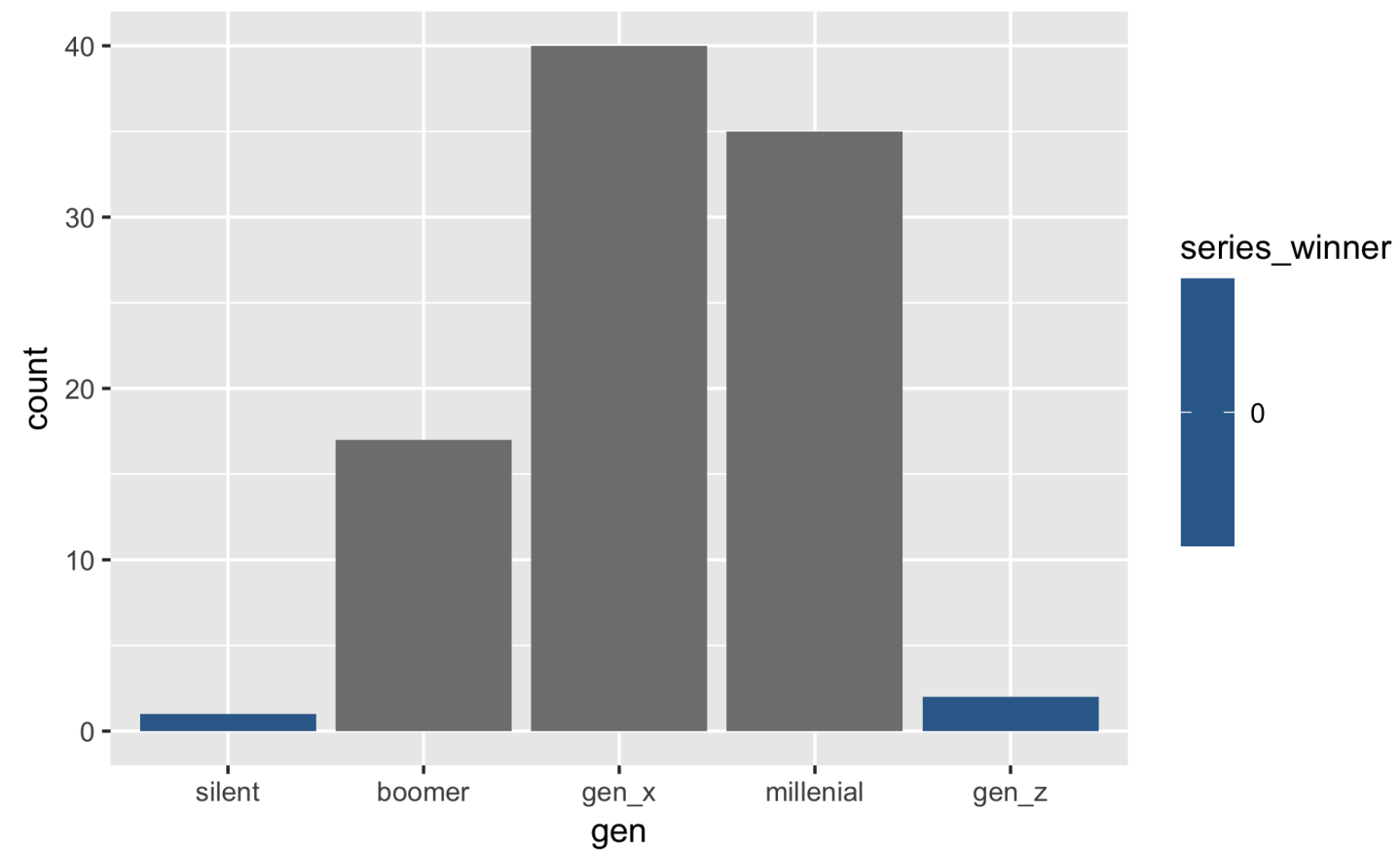
# Reorder generations chronologically

```
bakers <- bakers %>%  
  mutate(gen = fct_relevel(gen, "silent", "boomer",  
                           "gen_x", "millenial", "gen_z"))  
  
ggplot(bakers, aes(x = gen)) + geom_bar()
```



# Fill fail

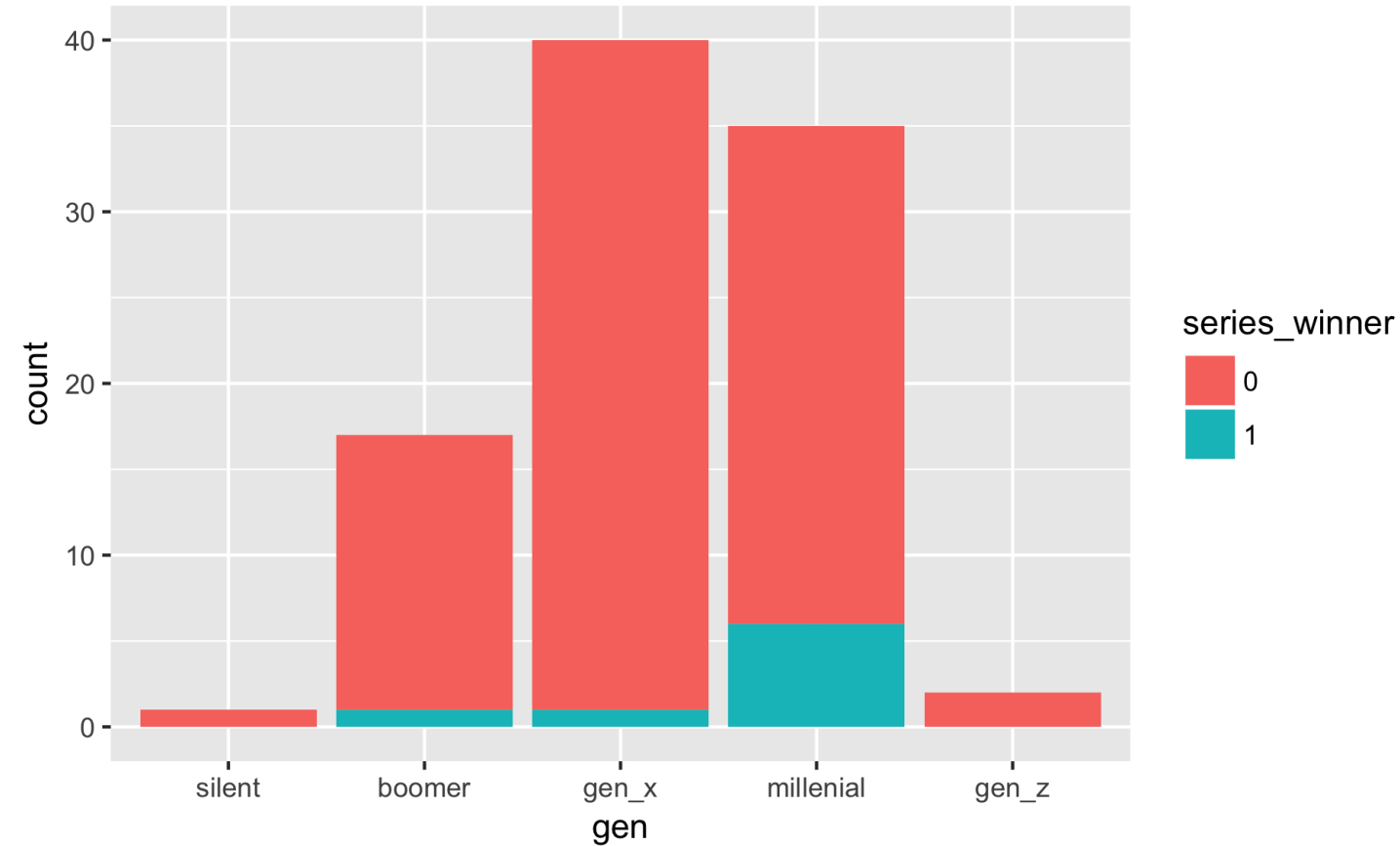
```
ggplot(bakers, aes(x = gen, fill = series_winner)) +  
  geom_bar()
```





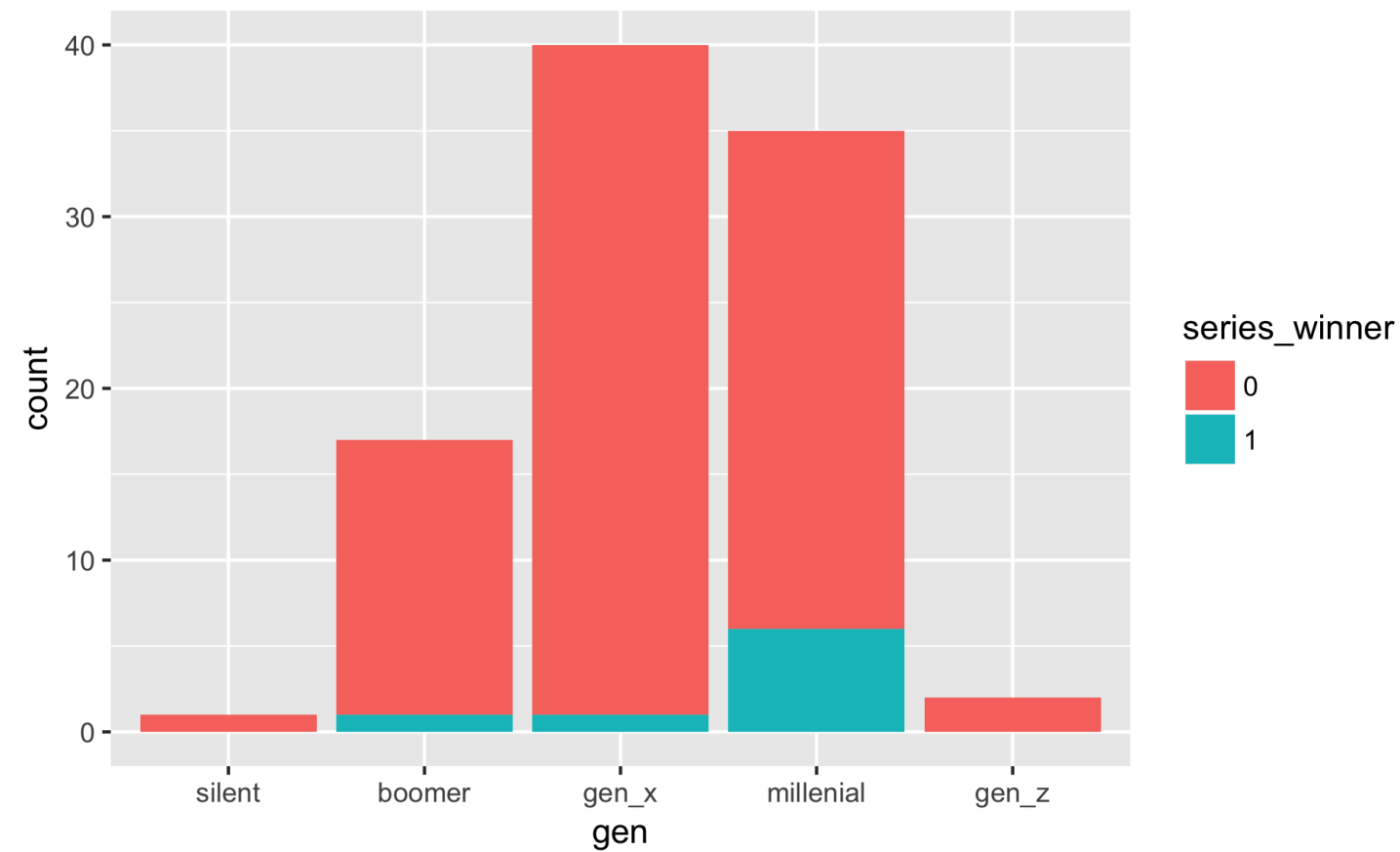
# Fill win!

```
bakers <- bakers %>%  
  mutate(series_winner = as.factor(series_winner))  
ggplot(bakers, aes(x = gen, fill = series_winner)) + geom_bar()
```



# Fill win!

```
ggplot(bakers, aes(x = gen, fill = as.factor(series_winner))) +  
  geom_bar()
```



# Let's practice!

WORKING WITH DATA IN THE TIDYVERSE

# Dates

WORKING WITH DATA IN THE TIDYVERSE



**Alison Hill**

Professor & Data Scientist

# The lubridate package

```
library(lubridate) # once per work session
```



<sup>1</sup> <http://lubridate.tidyverse.org>

# Cast character as a date

?ymd

## Usage

```
ymd(..., quiet = FALSE, tz = NULL, locale = Sys.getlocale("LC_TIME"), truncated = 0)

ydm(..., quiet = FALSE, tz = NULL, locale = Sys.getlocale("LC_TIME"), truncated = 0)

mdy(..., quiet = FALSE, tz = NULL, locale = Sys.getlocale("LC_TIME"), truncated = 0)

myd(..., quiet = FALSE, tz = NULL, locale = Sys.getlocale("LC_TIME"), truncated = 0)

dmy(..., quiet = FALSE, tz = NULL, locale = Sys.getlocale("LC_TIME"), truncated = 0)

dym(..., quiet = FALSE, tz = NULL, locale = Sys.getlocale("LC_TIME"), truncated = 0)
```

# ymd: Arguments

?ymd

## Arguments

... a character or numeric vector of suspected dates

## Examples

```
ymd("2010-08-17")  
mdy(c("08/17/2010", "January 01, 2018"))  
dmy("17 08 2010")
```

# Parse Dates

```
dmy("17 August 2010") # does this work?
```

```
"2010-08-17"
```

```
mdy("17 August 2010") # what about this?
```

```
NA  
Warning message: All formats failed to parse. No formats found.
```

```
ymd("17 August 2010") # what about this?
```

```
Warning message: All formats failed to parse. No formats found.
```



# Dates in a data frame

```
hosts <- tibble::tribble(~host, ~bday, ~premiere,  
  "Mary", "24 March 1935", "August 17th, 2010",  
  "Paul", "1 March 1966", "August 17th, 2010")
```

```
hosts
```

```
# A tibble: 2 x 3  
  host    bday      premiere  
  <chr> <chr>      <chr>  
1 Mary  24 March 1935 August 17th, 2010  
2 Paul  1 March 1966 August 17th, 2010
```

# Cast as dates

```
hosts
```

```
# A tibble: 2 x 3
  host  bday      premiere
  <chr> <chr>      <chr>
1 Mary  24 March 1935 August 17th, 2010
2 Paul  1 March 1966  August 17th, 2010
```

```
hosts <- hosts %>% mutate(bday = dmy(bday), premiere = mdy(premiere))
```

```
# A tibble: 2 x 3
  host  bday      premiere
  <chr> <date>      <date>
1 Mary  1935-03-24 2010-08-17
2 Paul  1966-03-01 2010-08-17
```

# Types of timespans

- `interval` : time spans bound by two real date-times.
- `duration` : the exact number of seconds in an interval.
- `period` : the change in the clock time in an interval.

<sup>1</sup> Lubridate Reference Manual  
(<http://lubridate.tidyverse.org/reference/timespan.html>)

# Calculating an interval

```
hosts <- hosts %>%  
  mutate(age_int = interval(bday, premiere))
```

```
hosts
```

```
# A tibble: 2 x 4  
  host  bday      premiere age_int  
  <chr> <date>    <date>    <S4: Interval>  
1 Mary  1935-03-24 2010-08-17 1935-03-24 UTC--2010-08-17 UTC  
2 Paul  1966-03-01 2010-08-17 1966-03-01 UTC--2010-08-17 UTC
```

# Converting units of timespans

```
years(1)
```

```
"1y 0m 0d 0H 0M 0S"
```

```
hosts %>%  
  mutate(years_decimal = age_int / years(1),  
         years_whole = age_int %/% years(1))
```

```
# A tibble: 2 x 4  
  host age_int years_decimal years_whole  
  <chr> <S4: Interval> <dbl> <dbl>  
1 Mary 1935-03-24 UTC--2010-08-17 UTC 75.4 75.  
2 Paul 1966-03-01 UTC--2010-08-17 UTC 44.5 44.
```

# Converting units of timespans

```
hosts %>%  
  mutate(age_y = age_int %/% years(1),  
         age_m = age_int %/% months(12))
```

```
# A tibble: 2 x 6  
  host  bday      premiere age_int          age_y age_m  
  <chr> <date>      <date>    <S4: Interval>    <dbl> <dbl>  
1 Mary  1935-03-24 2010-08-17 1935-03-24 UTC--2010-08-17 UTC    75.    75.  
2 Paul  1966-03-01 2010-08-17 1966-03-01 UTC--2010-08-17 UTC    44.    44.
```

# Let's practice!

WORKING WITH DATA IN THE TIDYVERSE

# Strings

WORKING WITH DATA IN THE TIDYVERSE



**Alison Hill**

Professor & Data Scientist



# String wrangling

```
series5
```

```
# A tibble: 7 x 3
  baker      about      showstopper
  <chr>      <chr>      <chr>
1 Chetna    35 years, Fashion designer Fusion Tiered Pies
2 Luis      42 years, Graphic designer Four Fruity Seasons Tower
3 Martha    17 years, Student Three Little Pigs Pie
4 Nancy     60 years, Retired manager Trio of Apple Pies
5 Richard   38 years, Builder Three Course Autumn Pie Fe
6 Norman    66 years, Retired naval officer Pieful Tower
7 Kate      41 years, Furniture restorer Rhubarb, Prune & Apple Por
```

# tidyr::separate

```
series5 <- series5 %>%  
  separate(about, into = c("age", "occupation"), sep = ", ")
```

```
series5  
# A tibble: 7 x 4  
  baker    age    occupation    showstopper  
  <chr>  <chr>    <chr>        <chr>  
1 Chetna 35 years Fashion designer Fusion Tiered Pies  
2 Luis  42 years Graphic designer Four Fruity Seasons Tower  
3 Martha 17 years Student Three Little Pigs Pie  
4 Nancy 60 years Retired manager Trio of Apple Pies  
5 Richard 38 years Builder Three Course Autumn Pie Feast  
6 Norman 66 years Retired naval officer Pieful Tower  
7 Kate  41 years Furniture restorer Rhubarb, Prune & Apple Pork Pies
```

# readr::parse\_number

```
series5 <- series5 %>%  
  separate(about, into = c("age", "occupation"), sep = ", ") %>%  
  mutate(age = parse_number(age))
```

series5

```
# A tibble: 7 x 4  
  baker      age occupation      showstopper  
  <chr>   <dbl> <chr>         <chr>  
1 Chetna   35. Fashion designer Fusion Tiered Pies  
2 Luis    42. Graphic designer Four Fruity Seasons Tower  
3 Martha  17. Student       Three Little Pigs Pie  
4 Nancy   60. Retired manager Trio of Apple Pies  
5 Richard 38. Builder       Three Course Autumn Pie Feast  
6 Norman  66. Retired naval officer Pieful Tower  
7 Kate    41. Furniture restorer Rhubarb, Prune & Apple Pork Pies
```

# The `stringr` package

```
library(stringr) # once per work session
```



<sup>1</sup> <http://stringr.tidyverse.org>

# String Basics

```
series5 <- series5 %>%  
  mutate(baker = str_to_upper(baker),  
         showstopper = str_to_lower(showstopper))
```

```
series5
```

```
# A tibble: 7 x 4  
  baker      age occupation      showstopper  
  <chr>   <dbl> <chr>      <chr>  
1 CHETNA   35. Fashion designer fusion tiered pies  
2 LUIS     42. Graphic designer four fruity seasons tower  
3 MARTHA  17. Student      three little pigs pie  
4 NANCY    60. Retired manager trio of apple pies  
5 RICHARD  38. Builder      three course autumn pie feast  
6 NORMAN   66. Retired naval officer pieful tower  
7 KATE     41. Furniture restorer rhubarb, prune & apple pork pies
```

# Detect String Patterns

```
series5 %>%  
  mutate(pie = str_detect(showstopper, "pie"))
```

```
# A tibble: 7 x 5  
  baker    age occupation      showstopper      pie  
  <chr>  <dbl> <chr>      <chr>      <lgl>  
1 CHETNA   35. Fashion designer fusion tiered pies TRUE  
2 LUIS     42. Graphic designer four fruity seasons tower FALSE  
3 MARTHA  17. Student three little pigs pie TRUE  
4 NANCY    60. Retired manager trio of apple pies TRUE  
5 RICHARD  38. Builder three course autumn pie feast TRUE  
6 NORMAN  66. Retired naval officer pieful tower TRUE  
7 KATE    41. Furniture restorer rhubarb, prune & apple pork pies TRUE
```

# Replace String Patterns

```
series5 %>%  
  mutate(showstopper = str_replace(showstopper, "pie", "tart"))
```

```
# A tibble: 7 x 4  
  baker      age occupation      showstopper  
  <chr>   <dbl> <chr>      <chr>  
1 CHETNA    35. Fashion designer fusion tiered tarts  
2 LUIS      42. Graphic designer four fruity seasons tower  
3 MARTHA    17. Student      three little pigs tart  
4 NANCY     60. Retired manager trio of apple tarts  
5 RICHARD   38. Builder      three course autumn tart feast  
6 NORMAN    66. Retired naval officer tartful tower  
7 KATE      41. Furniture restorer rhubarb, prune & apple pork tarts
```

# Remove String Patterns

```
series5 %>%  
  mutate(showstopper = str_remove(showstopper, "pie"))
```

```
# A tibble: 7 x 4  
  baker      age occupation      showstopper  
  <chr>    <dbl> <chr>      <chr>  
1 CHETNA    35. Fashion designer fusion tiered s  
2 LUIS      42. Graphic designer four fruity seasons tower  
3 MARTHA   17. Student      "three little pigs "  
4 NANCY     60. Retired manager trio of apple s  
5 RICHARD   38. Builder      three course autumn feast  
6 NORMAN   66. Retired naval officer ful tower  
7 KATE     41. Furniture restorer rhubarb, prune & apple pork s
```



# Trim white space

```
series5 %>%  
  mutate(showstopper = str_remove(showstopper, "pie"),  
         showstopper = str_trim(showstopper))
```

```
# A tibble: 7 x 4  
  baker      age occupation      showstopper  
  <chr>   <dbl> <chr>      <chr>  
1 CHETNA   35. Fashion designer fusion tiered s  
2 LUIS     42. Graphic designer four fruity seasons tower  
3 MARTHA   17. Student      three little pigs  
4 NANCY    60. Retired manager trio of apple s  
5 RICHARD  38. Builder      three course autumn feast  
6 NORMAN   66. Retired naval officer ful tower  
7 KATE     41. Furniture restorer rhubarb, prune & apple pork s
```

# Let's practice!

WORKING WITH DATA IN THE TIDYVERSE

# Final thoughts

WORKING WITH DATA IN THE TIDYVERSE



**Alison Hill**

Professor & Data Scientist

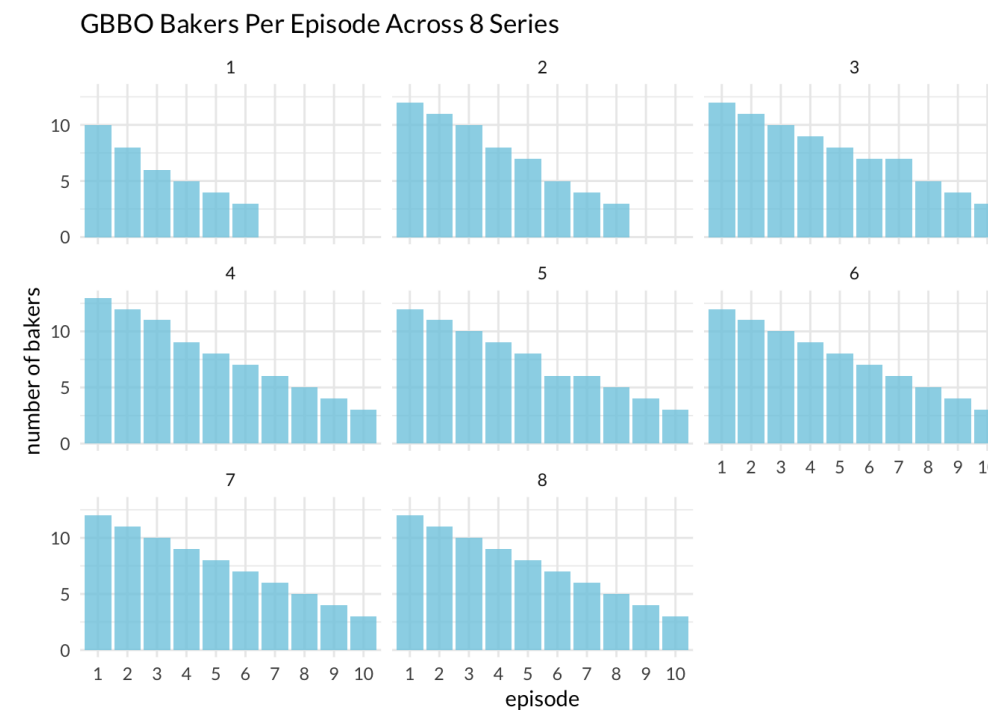
# Explore your data

```
bakeoff <- read_csv("bakeoff.csv")
glimpse(bakeoff)
skim(bakeoff)

bakeoff %>%
  count(series, baker) %>%
  count(series)

ggplot(bakeoff, aes(episode)) +
  geom_bar() +
  facet_wrap(~series)

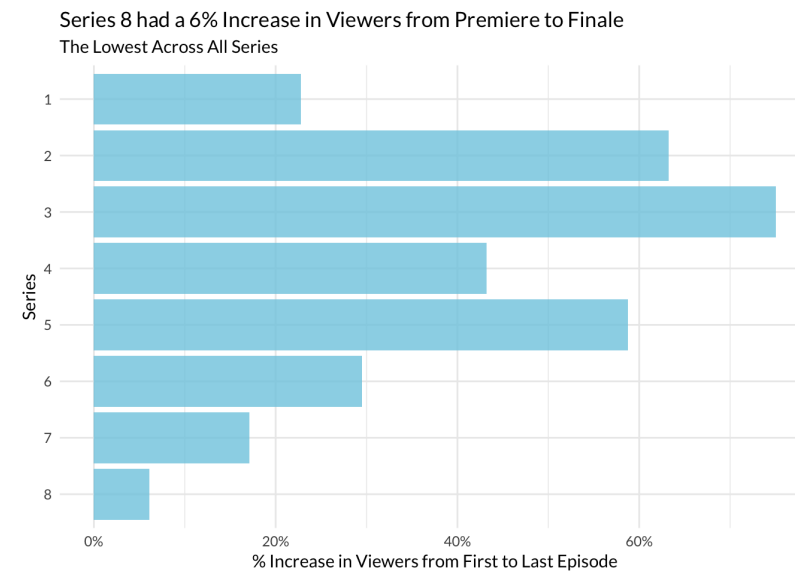
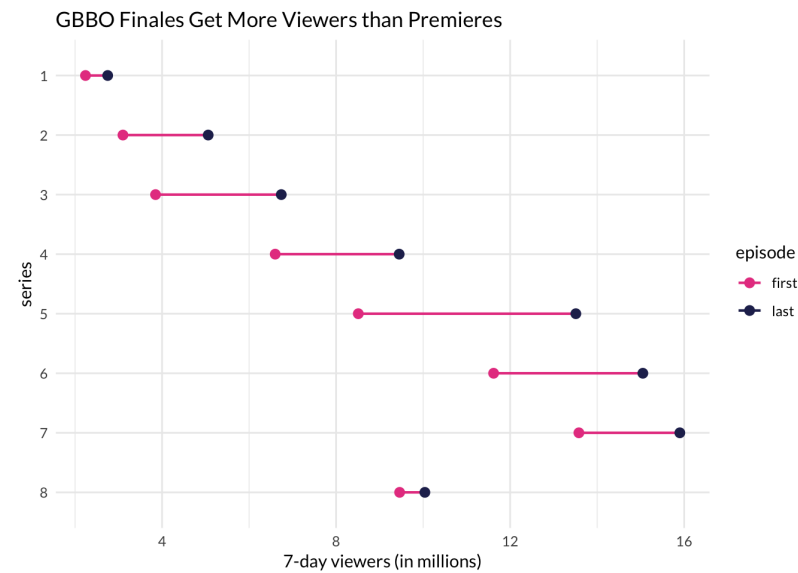
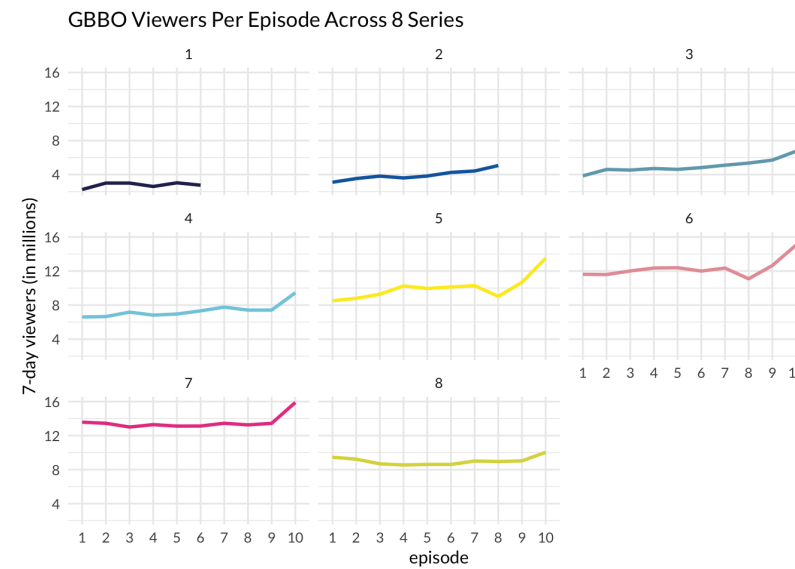
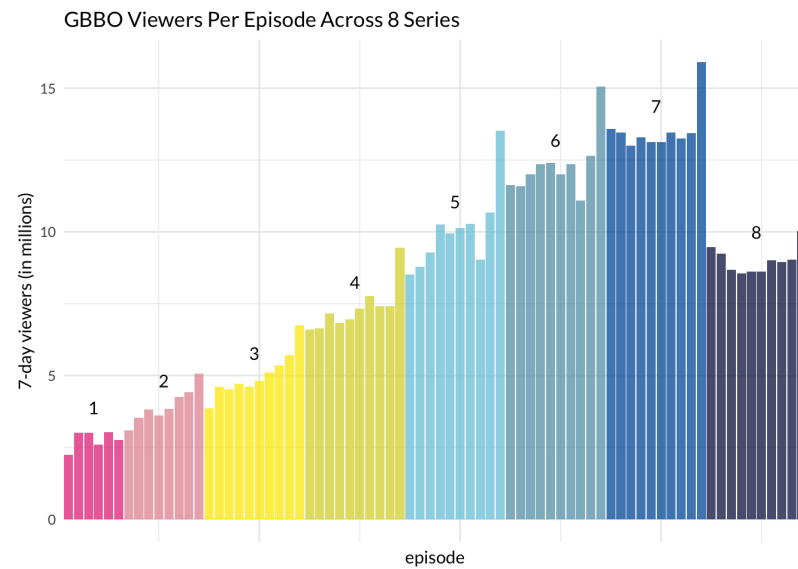
?read_csv
```



# Tame your data

```
ratings <- read_csv("ratings.csv",  
                    col_types = cols(  
                      series = col_factor(levels = NULL))) %>%  
  clean_names()  
viewers_7day <- ratings %>%  
  mutate(bbc = recode(channel, "Channel 4" = 0,  
                        .default = 1)) %>%  
  select(series, bbc, viewers_7day_ = ends_with("7day"))
```

# Tidy your data



# Transform your data

```
bakers <- bakers %>%
  mutate(gen = case_when(
    between(birth_year, 1928, 1945) ~ "silent",
    between(birth_year, 1946, 1964) ~ "boomer",
    between(birth_year, 1965, 1980) ~ "gen_x",
    between(birth_year, 1981, 1996) ~ "millenial",
    TRUE ~ "gen_z"
  ))

bakers <- bakers %>%
  mutate(gen = fct_relevel(gen, "silent", "boomer",
                           "gen_x", "millenial", "gen_z"))

ggplot(bakers, aes(x = gen)) + geom_bar()

bakers <- bakers %>%
  mutate(last_date_appeared_us = dmy(last_date_appeared_us),
         occupation = str_to_lower(occupation),
         student = str_detect(occupation, "student"))
```

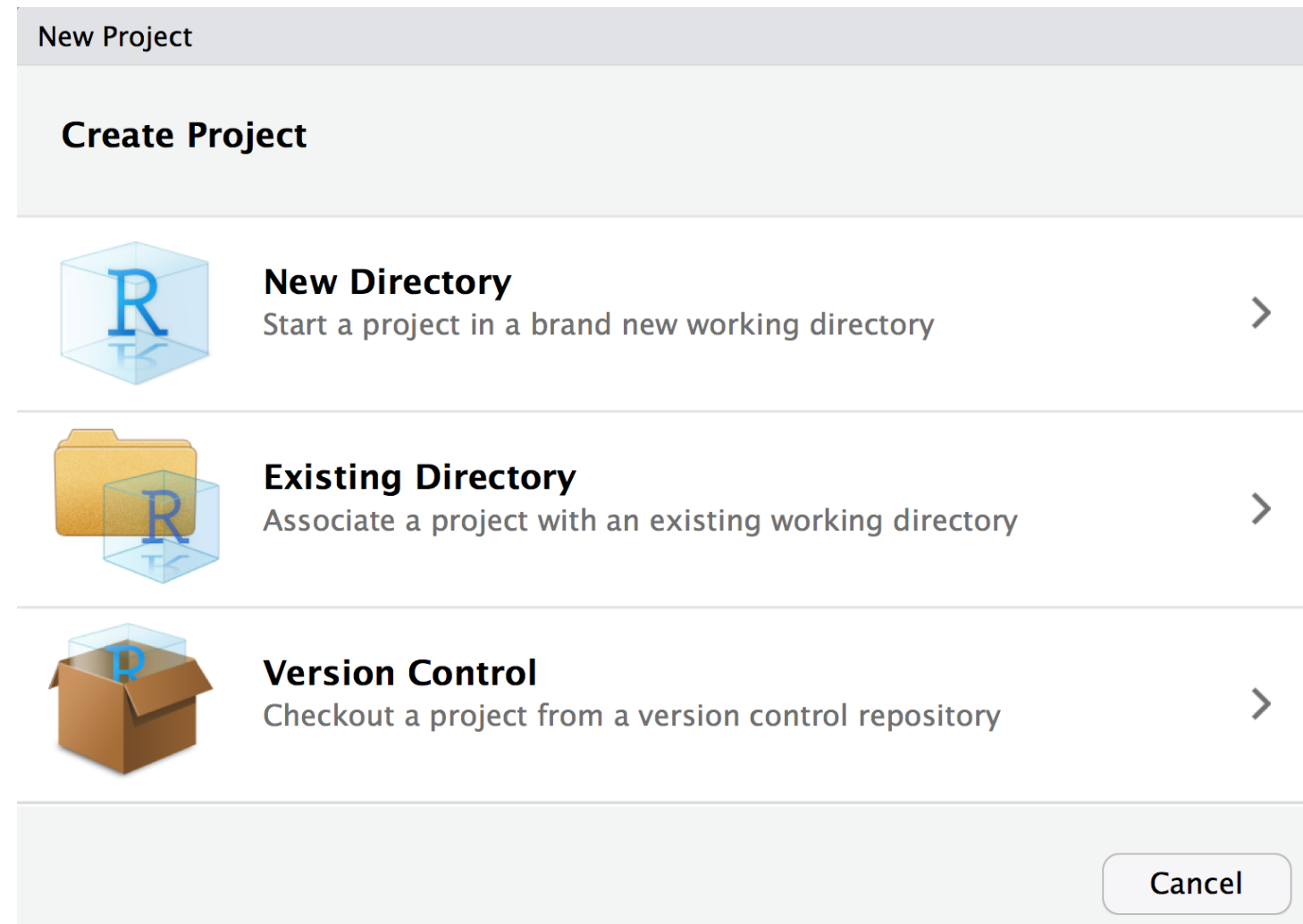
On your own



Studio<sup>®</sup>



# R Projects in RStudio



# Project-oriented workflows

```
bakeoff  
??? bakeoff.Rproj  
??? data  
|    ??? bakers.csv <-- this is my file!  
??? figures
```

```
# install.packages("here")  
library(here)  
bakers <- read_csv(here("data", "bakers.csv"))
```

The `here` package: <https://here.r-lib.org/>

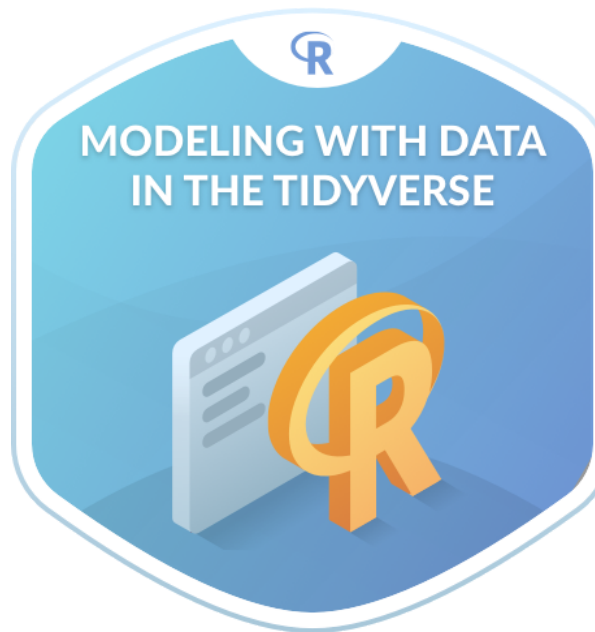
# What's next?



# What's next?



# What's next?



# Congratulations!

WORKING WITH DATA IN THE TIDYVERSE