R for DS - 3/1/17

Tidy Data

Exercise

Why is gather and spread not symetrical?

```
library(tibble)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)

stocks <- tibble(
  year   = c(2015, 2015, 2016, 2016),
  half  = c(   1,    2,     1,    2),
  return = c(1.88, 0.59, 0.92, 0.17)
)
stocks %>%
  spread(year, return)
```

```
## # A tibble: 2 × 3
##    half `2015` `2016`
## * <dbl>  <dbl>  <dbl>
## 1     1   1.88   0.92
## 2     2   0.59   0.17
```

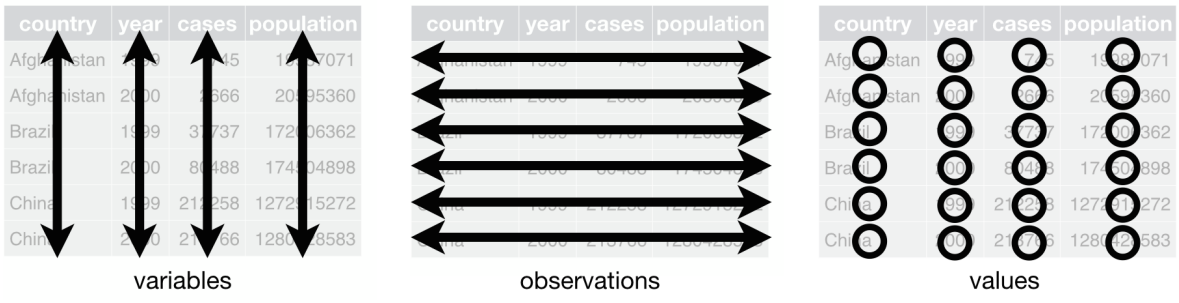


Figure 1: tidy data

```
stocks %>%
  spread(year, return) %>%
  gather("year", "return", `2015`:`2016`)
```

```
## # A tibble: 4 × 3
##    half  year return
##   <dbl> <chr>  <dbl>
## 1     1  2015   1.88
## 2     2  2015   0.59
## 3     1  2016   0.92
## 4     2  2016   0.17
```

```
stocks
```

```
## # A tibble: 4 × 3
##    year  half return
##   <dbl> <dbl>  <dbl>
## 1  2015     1   1.88
## 2  2015     2   0.59
## 3  2016     1   0.92
## 4  2016     2   0.17
```

```
stocks %>%
  spread(year, return) %>%
  gather("year", "return", `2015`:`2016`) %>%
  select(year, half, return)
```

```
## # A tibble: 4 × 3
##    year  half return
##   <chr> <dbl>  <dbl>
## 1  2015     1   1.88
## 2  2015     2   0.59
## 3  2016     1   0.92
## 4  2016     2   0.17
```

Tidy the data below

```
library(tibble)
preg <- tribble(
  ~pregnant, ~male, ~female,
  "yes",     NA,    10,
  "no",      20,    12
)
preg
```

```
## # A tibble: 2 × 3
##   pregnant  male female
##      <chr> <dbl>  <dbl>
## 1      yes    NA     10
## 2       no    20     12
```

```
preg %>% gather(male, female, key = "sex", value = "count")
```

```
## # A tibble: 4 × 3
##   pregnant    sex count
##      <chr>  <chr> <dbl>
## 1      yes   male    NA
```

```
## 2       no   male    20
## 3      yes female    10
## 4       no female    12
```

What do the extra and fill arguments do in separate()? Experiment with the various options for the following two toy datasets.

```
tibble(x = c("a,b,c", "d,e,f,g", "h,i,j")) %>%
  separate(x, c("one", "two", "three"), extra = "merge")
```

```
## # A tibble: 3 × 3
##     one   two three
## * <chr> <chr> <chr>
## 1     a     b     c
## 2     d     e   f,g
## 3     h     i     j
```

```
tibble(x = c("a,b,c", "d,e", "f,g,i")) %>%
  separate(x, c("one", "two", "three"), fill = "right")
```

```
## # A tibble: 3 × 3
##     one   two three
## * <chr> <chr> <chr>
## 1     a     b     c
## 2     d     e  <NA>
## 3     f     g     i
```

Who Example

```
who_dt <- tidyr::who
who_dt
```

```
## # A tibble: 7,240 × 60
##        country  iso2  iso3  year new_sp_m014 new_sp_m1524 new_sp_m2534
##          <chr> <chr> <chr> <int>       <int>        <int>        <int>
## 1  Afghanistan    AF   AFG  1980          NA           NA           NA
## 2  Afghanistan    AF   AFG  1981          NA           NA           NA
## 3  Afghanistan    AF   AFG  1982          NA           NA           NA
## 4  Afghanistan    AF   AFG  1983          NA           NA           NA
## 5  Afghanistan    AF   AFG  1984          NA           NA           NA
## 6  Afghanistan    AF   AFG  1985          NA           NA           NA
## 7  Afghanistan    AF   AFG  1986          NA           NA           NA
## 8  Afghanistan    AF   AFG  1987          NA           NA           NA
## 9  Afghanistan    AF   AFG  1988          NA           NA           NA
## 10 Afghanistan    AF   AFG  1989          NA           NA           NA
## # ... with 7,230 more rows, and 53 more variables: new_sp_m3544 <int>,
## #   new_sp_m4554 <int>, new_sp_m5564 <int>, new_sp_m65 <int>,
## #   new_sp_f014 <int>, new_sp_f1524 <int>, new_sp_f2534 <int>,
## #   new_sp_f3544 <int>, new_sp_f4554 <int>, new_sp_f5564 <int>,
## #   new_sp_f65 <int>, new_sn_m014 <int>, new_sn_m1524 <int>,
## #   new_sn_m2534 <int>, new_sn_m3544 <int>, new_sn_m4554 <int>,
## #   new_sn_m5564 <int>, new_sn_m65 <int>, new_sn_f014 <int>,
## #   new_sn_f1524 <int>, new_sn_f2534 <int>, new_sn_f3544 <int>,
## #   new_sn_f4554 <int>, new_sn_f5564 <int>, new_sn_f65 <int>,
```

```
## #   new_ep_m014 <int>, new_ep_m1524 <int>, new_ep_m2534 <int>,
## #   new_ep_m3544 <int>, new_ep_m4554 <int>, new_ep_m5564 <int>,
## #   new_ep_m65 <int>, new_ep_f014 <int>, new_ep_f1524 <int>,
## #   new_ep_f2534 <int>, new_ep_f3544 <int>, new_ep_f4554 <int>,
## #   new_ep_f5564 <int>, new_ep_f65 <int>, newrel_m014 <int>,
## #   newrel_m1524 <int>, newrel_m2534 <int>, newrel_m3544 <int>,
## #   newrel_m4554 <int>, newrel_m5564 <int>, newrel_m65 <int>,
## #   newrel_f014 <int>, newrel_f1524 <int>, newrel_f2534 <int>,
## #   newrel_f3544 <int>, newrel_f4554 <int>, newrel_f5564 <int>,
## #   newrel_f65 <int>
```

```
# remove "iso2" and "iso3" from data
who_dt <- who_dt %>% select(-iso2, -iso3)

# look at data
tibble::glimpse(who_dt)
```

```
## Observations: 7,240
## Variables: 58
## $ country      <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afg...
## $ year         <int> 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1...
## $ new_sp_m014  <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sp_m1524 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sp_m2534 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sp_m3544 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sp_m4554 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sp_m5564 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sp_m65   <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sp_f014  <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sp_f1524 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sp_f2534 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sp_f3544 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sp_f4554 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sp_f5564 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sp_f65   <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sn_m014  <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sn_m1524 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sn_m2534 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sn_m3544 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sn_m4554 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sn_m5564 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sn_m65   <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sn_f014  <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sn_f1524 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sn_f2534 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sn_f3544 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sn_f4554 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sn_f5564 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_sn_f65   <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_ep_m014  <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_ep_m1524 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_ep_m2534 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_ep_m3544 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_ep_m4554 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_ep_m5564 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
```

```
## $ new_ep_m65   <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_ep_f014  <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_ep_f1524 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_ep_f2534 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_ep_f3544 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_ep_f4554 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_ep_f5564 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ new_ep_f65   <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ newrel_m014  <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ newrel_m1524 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ newrel_m2534 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ newrel_m3544 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ newrel_m4554 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ newrel_m5564 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ newrel_m65   <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ newrel_f014  <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ newrel_f1524 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ newrel_f2534 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ newrel_f3544 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ newrel_f4554 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ newrel_f5564 <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
## $ newrel_f65   <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N...
```

```
# gather all columns and make a 'tall' dataset
who_dt <- who_dt %>%
  gather(new_sp_m014:newrel_f65, key = "key", value = "cases", na.rm = TRUE)
who_dt
```

```
## # A tibble: 76,046 × 4
##        country  year         key cases
## *        <chr> <int>       <chr> <int>
## 1  Afghanistan  1997 new_sp_m014     0
## 2  Afghanistan  1998 new_sp_m014    30
## 3  Afghanistan  1999 new_sp_m014     8
## 4  Afghanistan  2000 new_sp_m014    52
## 5  Afghanistan  2001 new_sp_m014   129
## 6  Afghanistan  2002 new_sp_m014    90
## 7  Afghanistan  2003 new_sp_m014   127
## 8  Afghanistan  2004 new_sp_m014   139
## 9  Afghanistan  2005 new_sp_m014   151
## 10 Afghanistan  2006 new_sp_m014   193
## # ... with 76,036 more rows
```

```
library(stringr)
# key_parts <- str_split(who_dt$key, "_")
# parts_length <- key_parts %>% lapply(length) %>% unlist()
# who_dt[parts_length != 3,]

# fix bad entry
who_dt$key <- str_replace(who_dt$key, "newrel", "new_rel")
# split the 'key' column into 'new', 'type', and 'sexage'
who_dt <- who_dt %>% separate(key, c("new", "type", "sexage"), sep = "_")
who_dt
```

```
## # A tibble: 76,046 × 6
```

```
##        country  year   new  type sexage cases
## *        <chr> <int> <chr> <chr>  <chr> <int>
## 1  Afghanistan  1997   new    sp   m014     0
## 2  Afghanistan  1998   new    sp   m014    30
## 3  Afghanistan  1999   new    sp   m014     8
## 4  Afghanistan  2000   new    sp   m014    52
## 5  Afghanistan  2001   new    sp   m014   129
## 6  Afghanistan  2002   new    sp   m014    90
## 7  Afghanistan  2003   new    sp   m014   127
## 8  Afghanistan  2004   new    sp   m014   139
## 9  Afghanistan  2005   new    sp   m014   151
## 10 Afghanistan  2006   new    sp   m014   193
## # ... with 76,036 more rows
```

```
# who_dt$sex <- who_dt$sex_age %>% str_sub(1, 1)
# who_dt$age_range <- who_dt$sex_age %>% str_sub(2)

# make the 'sexage' column into 'sex' and 'age' by spliting after the first character
who_dt <- who_dt %>% separate(sexage, c("sex", "age"), sep = 1)
who_dt
```

```
## # A tibble: 76,046 × 7
##        country  year   new  type   sex   age cases
## *        <chr> <int> <chr> <chr> <chr> <chr> <int>
## 1  Afghanistan  1997   new    sp     m   014     0
## 2  Afghanistan  1998   new    sp     m   014    30
## 3  Afghanistan  1999   new    sp     m   014     8
## 4  Afghanistan  2000   new    sp     m   014    52
## 5  Afghanistan  2001   new    sp     m   014   129
## 6  Afghanistan  2002   new    sp     m   014    90
## 7  Afghanistan  2003   new    sp     m   014   127
## 8  Afghanistan  2004   new    sp     m   014   139
## 9  Afghanistan  2005   new    sp     m   014   151
## 10 Afghanistan  2006   new    sp     m   014   193
## # ... with 76,036 more rows
```

```
# could remove 'new' as it is the same value
who_dt %>% select(-new)
```

```
## # A tibble: 76,046 × 6
##        country  year  type   sex   age cases
## *        <chr> <int> <chr> <chr> <chr> <int>
## 1  Afghanistan  1997    sp     m   014     0
## 2  Afghanistan  1998    sp     m   014    30
## 3  Afghanistan  1999    sp     m   014     8
## 4  Afghanistan  2000    sp     m   014    52
## 5  Afghanistan  2001    sp     m   014   129
## 6  Afghanistan  2002    sp     m   014    90
## 7  Afghanistan  2003    sp     m   014   127
## 8  Afghanistan  2004    sp     m   014   139
## 9  Afghanistan  2005    sp     m   014   151
## 10 Afghanistan  2006    sp     m   014   193
## # ... with 76,036 more rows
```

```
# select also works to select certain columns only
who_dt %>% select(country, type)
```

```
## # A tibble: 76,046 × 2
##        country  type
## *        <chr> <chr>
## 1  Afghanistan    sp
## 2  Afghanistan    sp
## 3  Afghanistan    sp
## 4  Afghanistan    sp
## 5  Afghanistan    sp
## 6  Afghanistan    sp
## 7  Afghanistan    sp
## 8  Afghanistan    sp
## 9  Afghanistan    sp
## 10 Afghanistan    sp
## # ... with 76,036 more rows
```

```
# can use mutate and select to combine (then drop previous) columns
who_dt %>%
  mutate(
    new_type = paste(new, type, sep = "_")
  ) %>%
  select(-new, -type)
```

```
## # A tibble: 76,046 × 6
##        country  year   sex   age cases new_type
##          <chr> <int> <chr> <chr> <int>    <chr>
## 1  Afghanistan  1997     m   014     0   new_sp
## 2  Afghanistan  1998     m   014    30   new_sp
## 3  Afghanistan  1999     m   014     8   new_sp
## 4  Afghanistan  2000     m   014    52   new_sp
## 5  Afghanistan  2001     m   014   129   new_sp
## 6  Afghanistan  2002     m   014    90   new_sp
## 7  Afghanistan  2003     m   014   127   new_sp
## 8  Afghanistan  2004     m   014   139   new_sp
## 9  Afghanistan  2005     m   014   151   new_sp
## 10 Afghanistan  2006     m   014   193   new_sp
## # ... with 76,036 more rows
```

```
# get summary metrics about the cases per country
who_dt %>%
  group_by(country) %>%
  summarise(
    total_cases = sum(cases),
    min_cases = min(cases),
    max_cases = max(cases)
  )
```

```
## # A tibble: 219 × 4
##                country total_cases min_cases max_cases
##                  <chr>       <int>     <int>     <int>
## 1          Afghanistan      140225         0      2449
## 2              Albania        5335         0        67
## 3              Algeria      128119        25      1982
## 4       American Samoa          41         0         2
## 5              Andorra         103         0         6
## 6               Angola      308365        14      3792
## 7             Anguilla           2         0         1
```

```
## 8  Antigua and Barbuda          55         0         3
## 9             Argentina      117156        17      1124
## 10              Armenia       15991         0       254
## # ... with 209 more rows
```

```
# count the occurances of each case per country
who_dt %>%
  group_by(country) %>%
  count(cases)
```

```
## Source: local data frame [29,932 x 3]
## Groups: country [?]
##
##        country cases     n
##          <chr> <int> <int>
## 1  Afghanistan     0    17
## 2  Afghanistan     1     1
## 3  Afghanistan     2     1
## 4  Afghanistan     3     1
## 5  Afghanistan     5     2
## 6  Afghanistan     6     1
## 7  Afghanistan     8     4
## 8  Afghanistan    10     1
## 9  Afghanistan    14     1
## 10 Afghanistan    20     1
## # ... with 29,922 more rows
```

```
# sum the cases per country
who_dt %>%
  group_by(country) %>%
  tally(cases)
```

```
## # A tibble: 219 × 2
##                 country      n
##                   <chr>  <int>
## 1           Afghanistan 140225
## 2               Albania   5335
## 3               Algeria 128119
## 4        American Samoa     41
## 5               Andorra    103
## 6                Angola 308365
## 7              Anguilla      2
## 8  Antigua and Barbuda     55
## 9             Argentina 117156
## 10              Armenia  15991
## # ... with 209 more rows
```

```
# group by country
who_country <- who_dt %>%
  group_by(country)

# get total count (ungroup first)
who_country %>%
  ungroup() %>%
  tally(cases)
```

```
## # A tibble: 1 × 1
##          n
##      <int>
## 1 43397518
```

Example

Clean data

```
library(readr)
library(stringr)
library(dplyr)
library(tidyr)

billboard <- read_csv("https://github.com/hadley/tidy-data/raw/master/data/billboard.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   artist.inverted = col_character(),
##   track = col_character(),
##   time = col_time(format = ""),
##   genre = col_character(),
##   date.entered = col_date(format = ""),
##   date.peaked = col_date(format = ""),
##   x66th.week = col_character(),
##   x67th.week = col_character(),
##   x68th.week = col_character(),
##   x69th.week = col_character(),
##   x70th.week = col_character(),
##   x71st.week = col_character(),
##   x72nd.week = col_character(),
##   x73rd.week = col_character(),
##   x74th.week = col_character(),
##   x75th.week = col_character(),
##   x76th.week = col_character()
## )
```

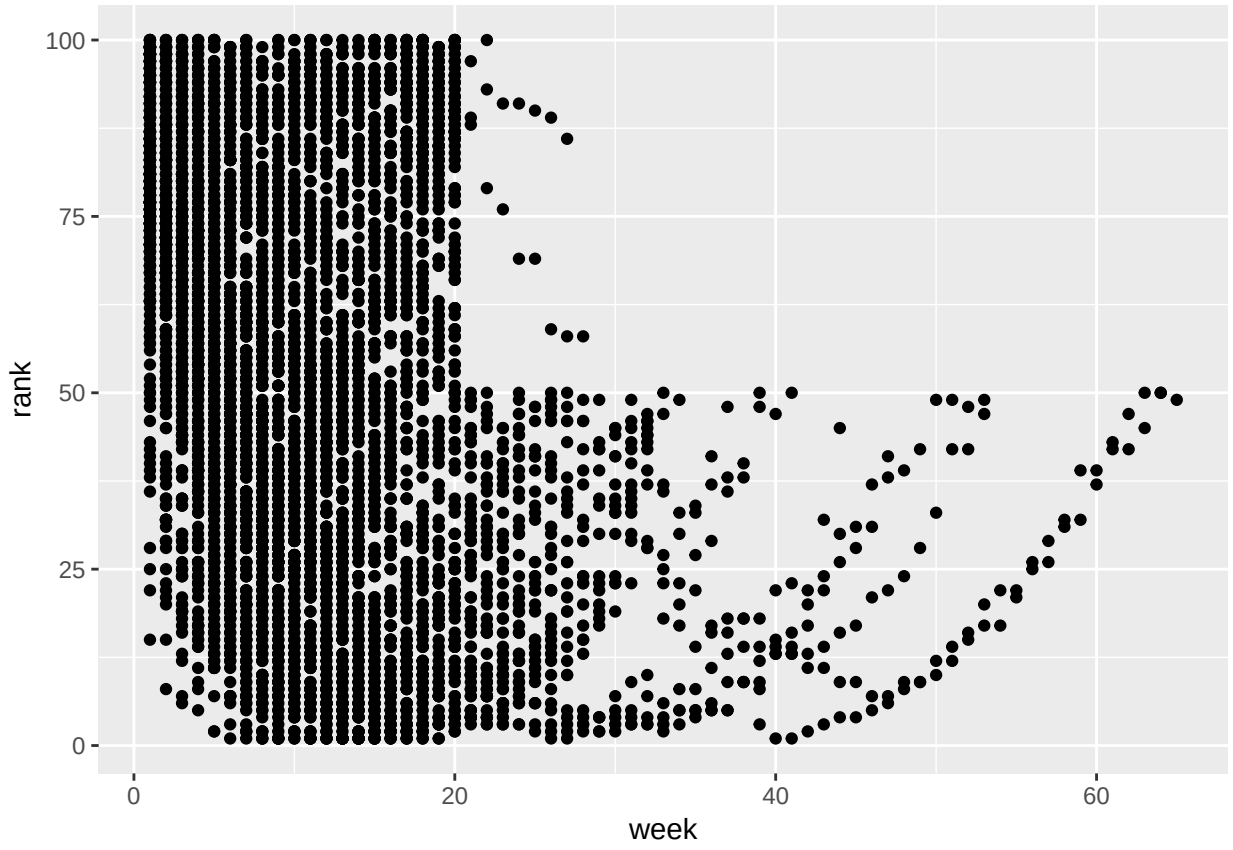
```
## See spec(...) for full column specifications.
```

```
billboard <- billboard %>% select(-date.peaked)
colnames(billboard)[2] <- "artist"

week_cols <- str_c("wk", 1:76)
colnames(billboard)[-(1:6)] <- week_cols

billboard_tall <- billboard %>%
  mutate(
    artist = iconv(artist, "MAC", "ASCII//translit"),
    track = str_replace(track, " \\(.*?\\)", "")
  ) %>%
  gather("week", "rank", wk1:wk76) %>%
  mutate(
```

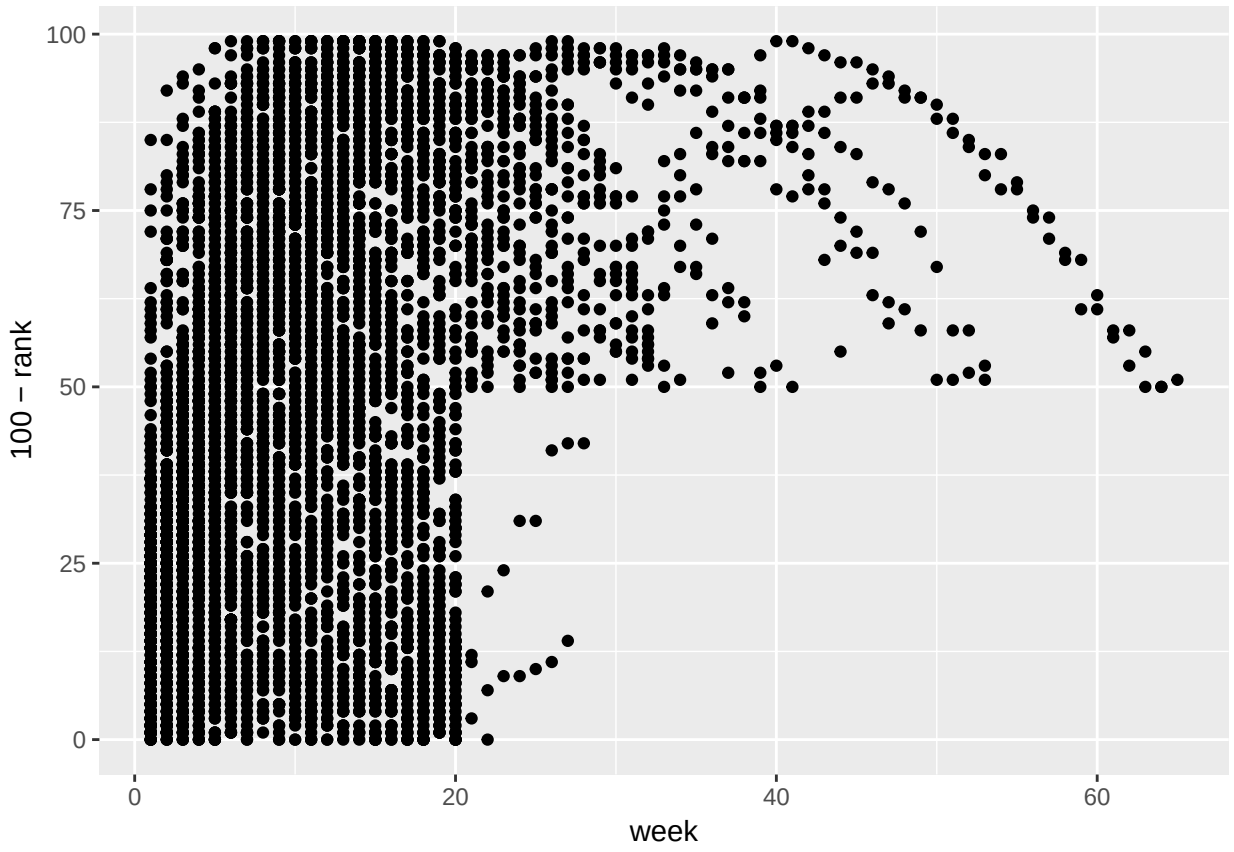
```
    week = as.numeric(str_sub(week, 3)),
    rank = as.numeric(rank)
  ) %>%
  filter(!is.na(rank))
```

Explore the data

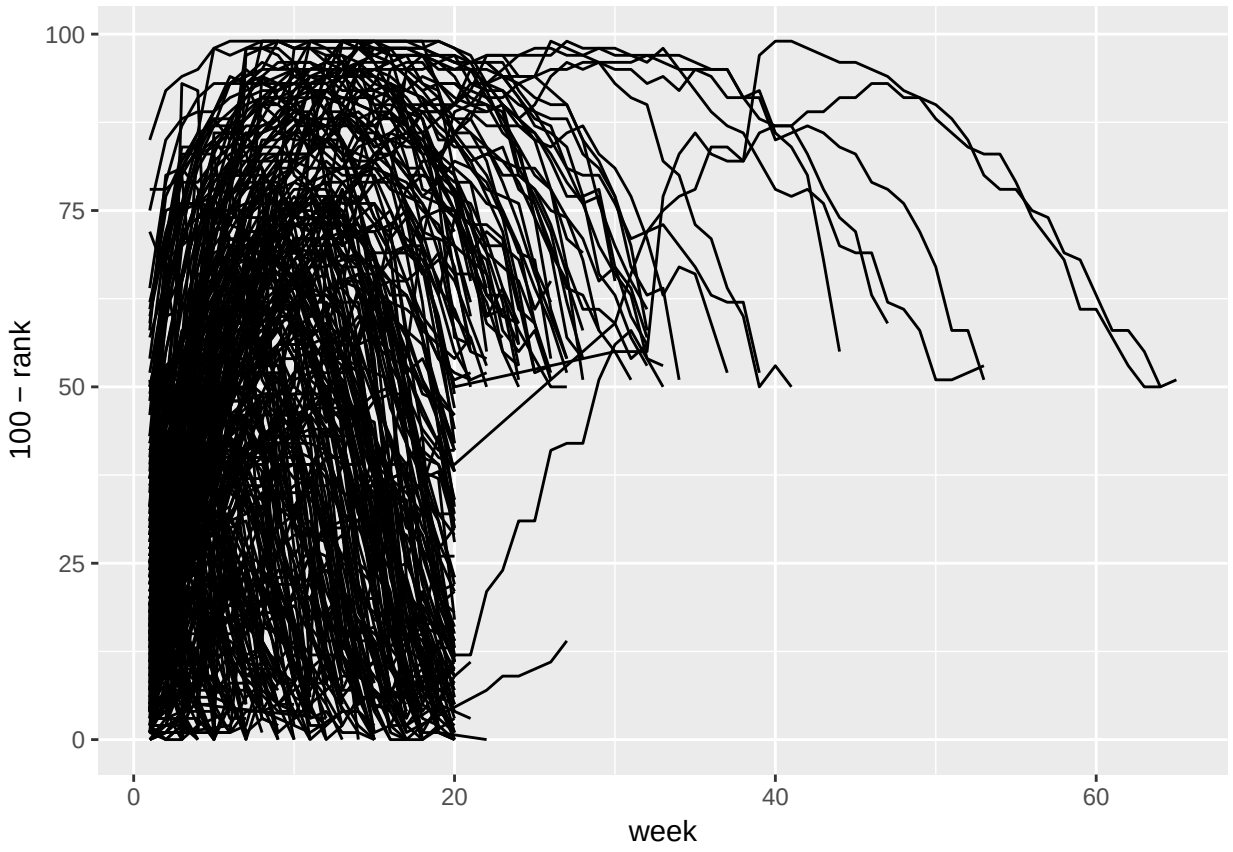
```
library(ggplot2)
qplot(week, rank, data = billboard_tall)
```



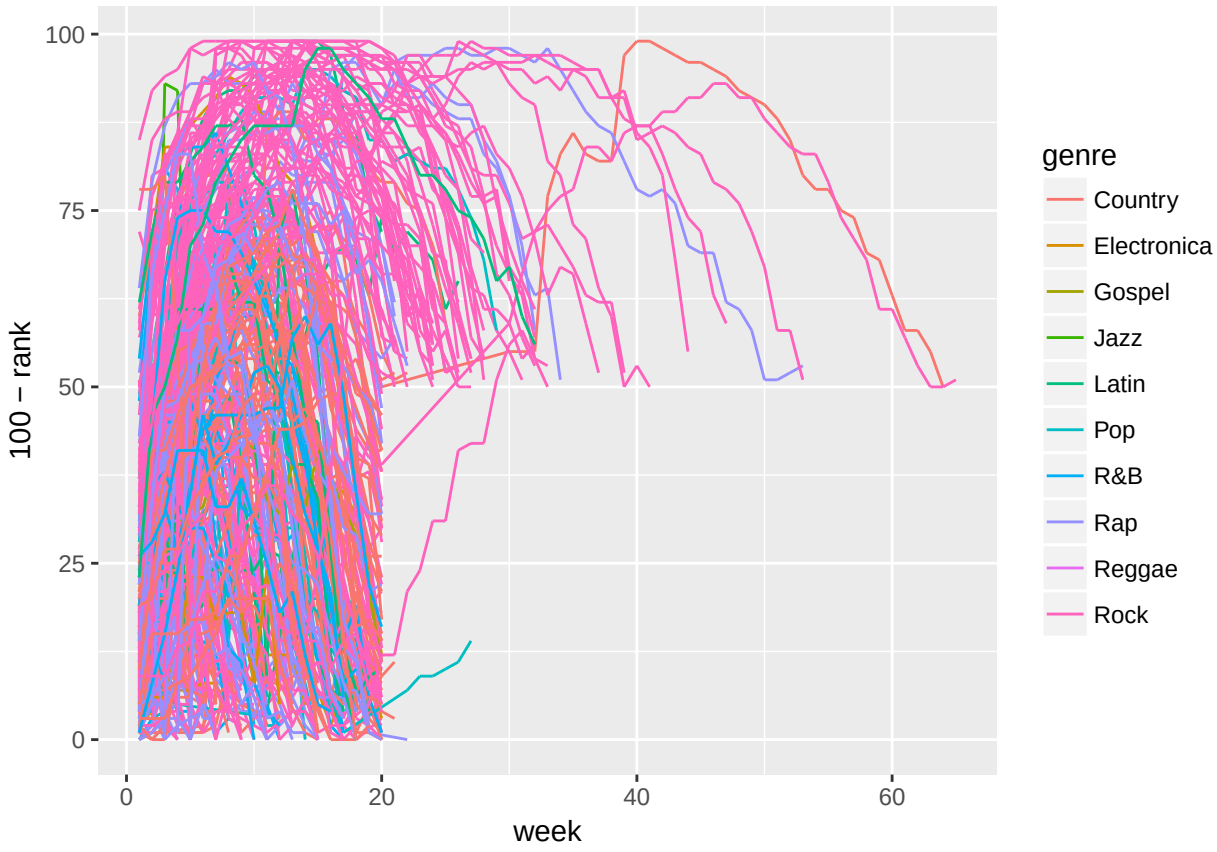

```
# higher is better
qplot(week, 100 - rank, data = billboard_tall)
```



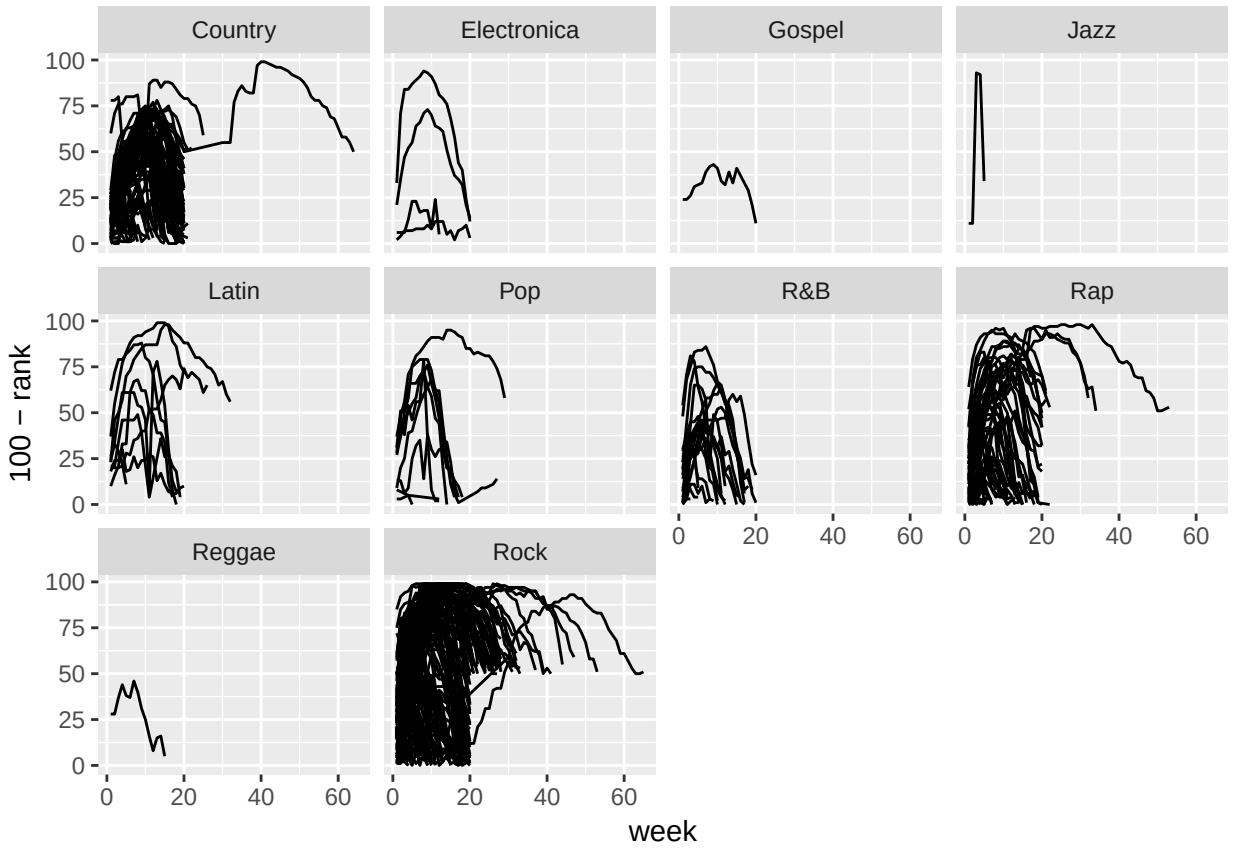

```
# show the path of a each song
qplot(week, 100 - rank, data = billboard_tall, geom = "line", group = track)
```




```
# color by genre
qplot(week, 100 - rank, data = billboard_tall, geom = "line", group = track, color = genre)
```




```
# split by genre
qplot(week, 100 - rank, data = billboard_tall, geom = "line", group = track) + facet_wrap(~ genre)
```



There are two odd occurances that I can see.

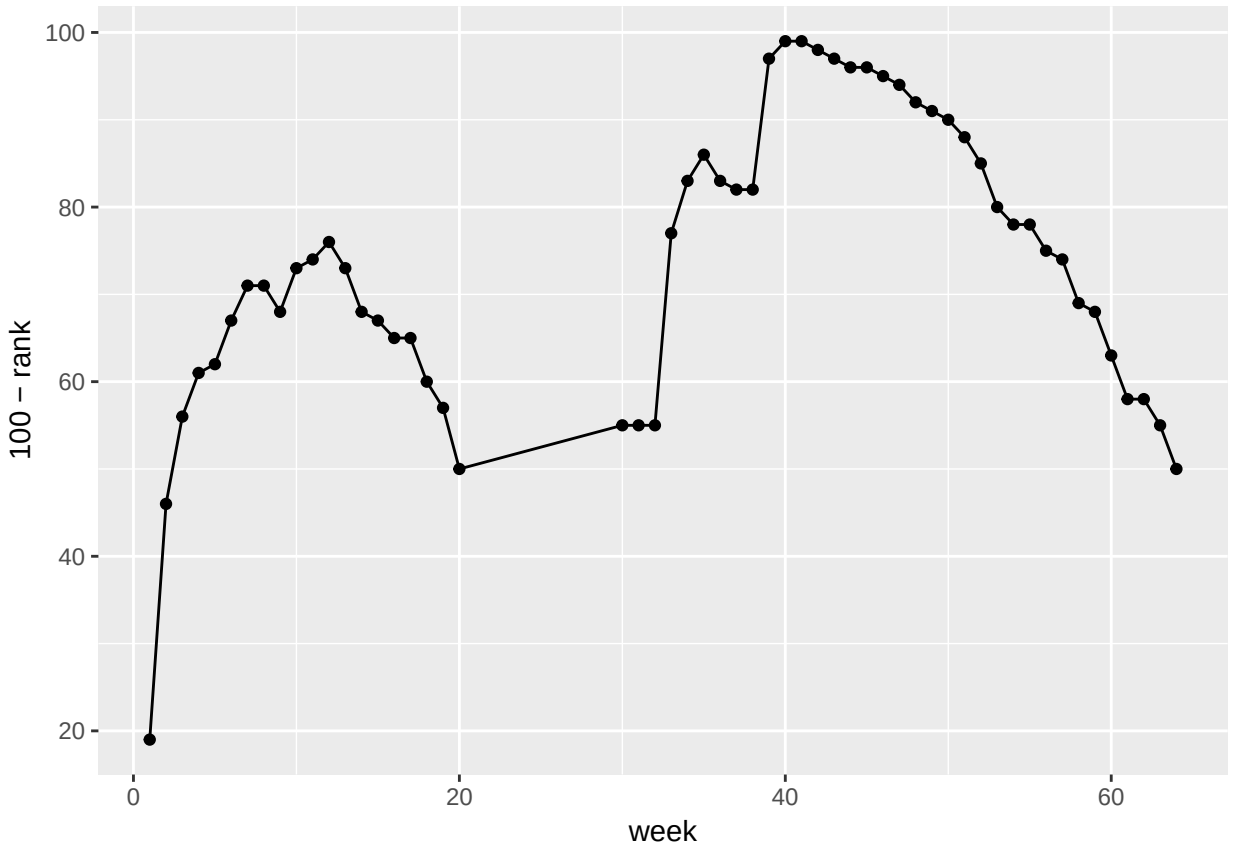
1. 20 weeks seems to be a hard cut-off
2. There is a country song that left the charts and came back.

Country song

```
billboard_tall %>%
  filter(genre == "Country", week > 40) %>%
  select(artist, track) %>%
  unique()
```

```
## # A tibble: 1 × 2
##     artist  track
##      <chr>  <chr>
## 1 Lonestar Amazed
```

```
lonestar <- billboard_tall %>%
  filter(artist == "Lonestar", track == "Amazed")

ggplot(data = lonestar, mapping = aes(week, 100 - rank)) +
  geom_point() +
  geom_line()
```



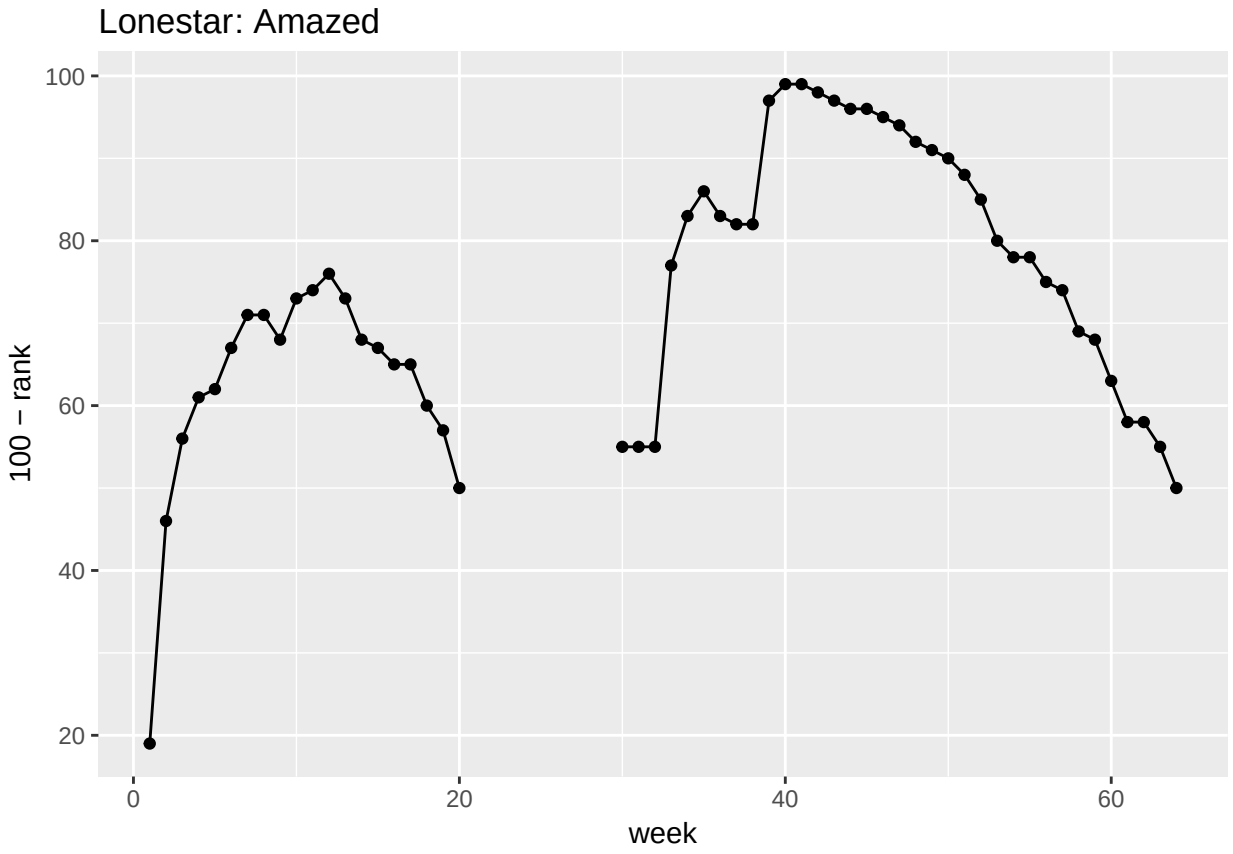
Missing weeks around the 20's

```
all_weeks <- 1:max(lonestar$week)
missing_weeks <- all_weeks[! all_weeks %in% lonestar$week]
missing_weeks
```

```
## [1] 21 22 23 24 25 26 27 28 29
```

```
lonestarNA <- lonestar[seq_along(missing_weeks),] %>%
  mutate(
    week = missing_weeks,
    rank = NA
  )
lonestar_with_missing <- rbind(lonestar, lonestarNA)

ggplot(data = lonestar_with_missing, mapping = aes(week, 100 - rank)) +
  geom_point() +
  geom_line() +
  labs(title = "Lonestar: Amazed")
```

```
## Warning: Removed 9 rows containing missing values (geom_point).
```



Hard Cutoff

```
billboard_tall %>% group_by(artist, track) %>% summarise(max_week = max(week, na.rm = TRUE))
```

```
## Source: local data frame [317 x 3]
## Groups: artist [?]
##
##            artist                            track max_week
##             <chr>                            <chr>    <dbl>
## 1           2 Pac                   Baby Don't Cry        7
## 2         2Ge+her The Hardest Part Of Breaking Up        3
## 3    3 Doors Down                       Kryptonite       53
## 4    3 Doors Down                            Loser       20
## 5        504 Boyz                     Wobble Wobble       18
## 6            98^0          Give Me Just One Night       20
## 7         A*Teens                     Dancing Queen        5
## 8         Aaliyah                     I Don't Wanna       20
## 9         Aaliyah                         Try Again       32
## 10 Adams, Yolanda                     Open My Heart       20
## # ... with 307 more rows
```

```
billboard_tall %>% group_by(artist, track, genre) %>% summarise(max_week = max(week, na.rm = TRUE)) %>%
```

```
## Source: local data frame [317 x 4]
## Groups: artist, track [317]
##
```

```
##              artist                      track   genre max_week
##               <chr>                      <chr>   <chr>    <dbl>
## 1             Creed                     Higher    Rock       65
## 2          Lonestar                    Amazed Country       64
## 3      3 Doors Down                 Kryptonite    Rock       53
## 4       Hill, Faith                    Breathe     Rap       53
## 5             Creed        With Arms Wide Open    Rock       47
## 6               Joe               I Wanna Know    Rock       44
## 7  Vertical Horizon        Everything You Want    Rock       41
## 8   matchbox twenty                       Bent    Rock       39
## 9     Braxton, Toni       He Wasn't Man Enough    Rock       37
## 10            Nelly (Hot S**t) Country Grammar     Rap       34
## # ... with 307 more rows
```

```
max_week_dt <- billboard_tall %>% group_by(artist, track, genre) %>% summarise(max_week = max(week, na.

nrow(max_week_dt)
```

```
## [1] 317
```

```
max_week_dt %>%
  filter(max_week == 20) %>%
  nrow()
```

```
## [1] 80
```

```
80/317
```

```
## [1] 0.2523659
```

```
done_at_20 <- max_week_dt %>%
  filter(max_week == 20) %>%
  group_by(genre) %>%
  count() %>%
  mutate(twenty_count = n) %>%
  select(-n)
done_at_20
```

```
## # A tibble: 7 × 2
##         genre twenty_count
##         <chr>        <int>
## 1     Country           31
## 2 Electronica            3
## 3      Gospel            1
## 4       Latin            2
## 5         R&B            2
## 6         Rap            9
## 7        Rock           32
```

```
genre_count <- max_week_dt %>%
  group_by(genre) %>%
  count() %>%
  mutate(total_count = n) %>%
  select(-n)
genre_count
```

```
## # A tibble: 10 × 2
##          genre total_count
```

```
##          <chr>       <int>
## 1      Country          74
## 2  Electronica           4
## 3       Gospel           1
## 4         Jazz           1
## 5        Latin           9
## 6          Pop           9
## 7          R&B          23
## 8          Rap          58
## 9       Reggae           1
## 10        Rock         137
```

```
left_join(done_at_20, genre_count) %>%
  mutate(
    perc = twenty_count / total_count
  )
```

```
## Joining, by = "genre"
```

```
## # A tibble: 7 × 4
##         genre twenty_count total_count       perc
##         <chr>        <int>       <int>      <dbl>
## 1     Country           31          74 0.41891892
## 2 Electronica            3           4 0.75000000
## 3      Gospel            1           1 1.00000000
## 4       Latin            2           9 0.22222222
## 5         R&B            2          23 0.08695652
## 6         Rap            9          58 0.15517241
## 7        Rock           32         137 0.23357664
```

A lot of country songs stop at exactly 20 weeks.