Cleaning data

Stewart Li

9/15/2020

Import

```
billboard_raw <- read_csv("https://raw.githubusercontent.com/hadley/tidy-data/master/dat
a/billboard.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   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_logical(),
##   x67th.week = col_logical(),
##   x68th.week = col_logical(),
##   x69th.week = col_logical(),
##   x70th.week = col_logical(),
##   x71st.week = col_logical(),
##   x72nd.week = col_logical(),
##   x73rd.week = col_logical(),
##   x74th.week = col_logical(),
##   x75th.week = col_logical(),
##   x76th.week = col_logical()
## )
```

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

Clean

```
df <- billboard_raw %>%
  pivot_longer(starts_with('x'),
               names_to = 'week',
               names_pattern = '^x(\\d\\d?)[a-z]{2}\\.week', # parse_number()
               # names_transform = list(week = as.integer), # version https://github.co
m/tidyverse/tidyr/issues/980
               values_to = 'rank',
               values_drop_na = TRUE) %>%
  mutate(week = readr::parse_number(week)) %>%
  separate(time,
           into = c('minutes', 'seconds', 'other'),
           sep = ':',
           convert = TRUE,
           remove = TRUE) %>% # use two cols
  mutate(len = minutes + seconds / 60,
         date = date.entered + (week - 1)*7) %>%
  select(-minutes, -seconds, -other) %>%
  rename(artist = artist.inverted)

write_csv(df, here::here('data', "billboard_clean.csv"))
```

Mung

```
df <- read_csv(here::here('data/billboard_clean.csv'))
```

```
## Parsed with column specification:
## cols(
##   year = col_double(),
##   artist = col_character(),
##   track = col_character(),
##   genre = col_character(),
##   date.entered = col_date(format = ""),
##   date.peaked = col_date(format = ""),
##   week = col_double(),
##   rank = col_double(),
##   len = col_double(),
##   date = col_date(format = "")
## )
```

```
head(df)
```

```
## # A tibble: 6 x 10
##    year artist track genre date.entered date.peaked  week  rank   len date
##   <dbl> <chr>  <chr> <chr> <date>       <date>      <dbl> <dbl> <dbl> <date>
## 1  2000 Desti~ Inde~ Rock  2000-09-23   2000-11-18      1    78  3.63 2000-09-23
## 2  2000 Desti~ Inde~ Rock  2000-09-23   2000-11-18      2    63  3.63 2000-09-30
## 3  2000 Desti~ Inde~ Rock  2000-09-23   2000-11-18      3    49  3.63 2000-10-07
## 4  2000 Desti~ Inde~ Rock  2000-09-23   2000-11-18      4    33  3.63 2000-10-14
## 5  2000 Desti~ Inde~ Rock  2000-09-23   2000-11-18      5    23  3.63 2000-10-21
## 6  2000 Desti~ Inde~ Rock  2000-09-23   2000-11-18      6    15  3.63 2000-10-28
```

```
best <- df %>%
  group_by(artist, track) %>%
  summarise(best_score = min(rank),
            best_last = sum(rank == 1)) %>%
  mutate(peak = ifelse(best_score == 1, 'best hit', 'no hit')) %>%
  ungroup()
```

```
## `summarise()` regrouping output by 'artist' (override with `.groups` argument)
```

Table

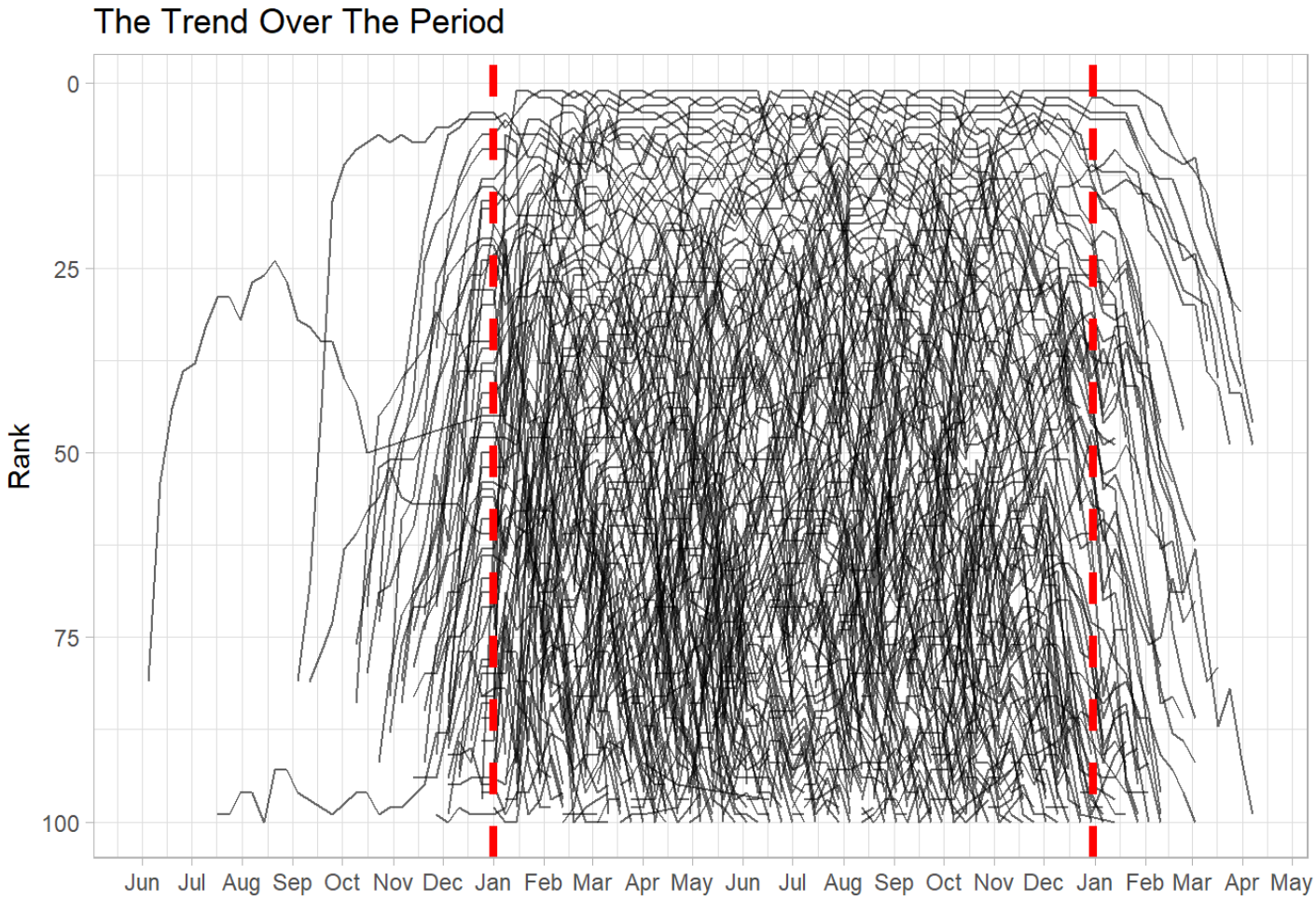
```
best %>%
  filter(fct_lump(artist, 5, w = best_last) != 'Other') %>%
  kable() %>%
  kable_styling(full_width = F) %>%
  column_spec(1, bold = TRUE) %>%
  collapse_rows(columns = c(1), valign = "top")
```

artist	track	best_score	best_last	peak
Aguilera, Christina	Come On Over Baby (All I Want Is You)	1	4	best hit
	I Turn To You	3	0	no hit
	What A Girl Wants	1	2	best hit
Destiny's Child	Independent Women Part I	1	11	best hit
	Jumpin' Jumpin'	3	0	no hit
	Say My Name	1	3	best hit
Madonna	American Pie	29	0	no hit
	Music	1	4	best hit
Santana	Maria, Maria	1	10	best hit

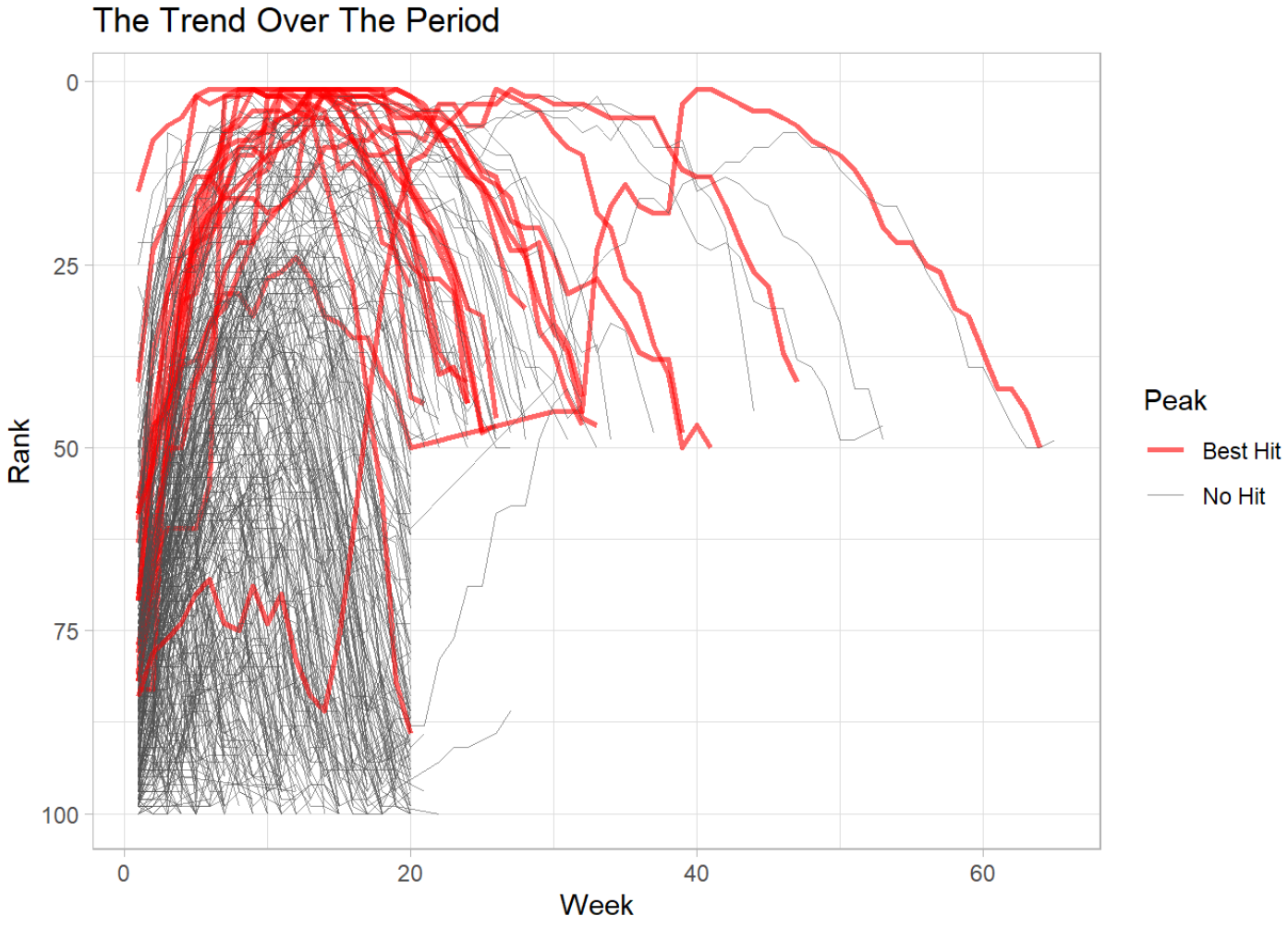
artist	track	best_score	best_last	peak
Savage Garden	Crash And Burn	24	0	no hit
	I Knew I Loved You	1	4	best hit

Plot

```
df %>%
  ggplot(aes(date, rank)) +
  geom_line(aes(group = track), alpha = 0.6) +
  geom_vline(xintercept = as.numeric(as.Date("2000-01-01", "%Y-%m-%d")), color = 'red',
 size = 1.5, lty = 2) +
  geom_vline(xintercept = as.numeric(as.Date("2000-12-31", "%Y-%m-%d")), color = 'red',
 size = 1.5, lty = 2) +
  scale_x_date(date_breaks = "1 month", date_labels = "%b") +
  scale_y_reverse() +
  labs(title = "The Trend Over The Period",
       x = "",
       y = "Rank") +
  theme_light()
```



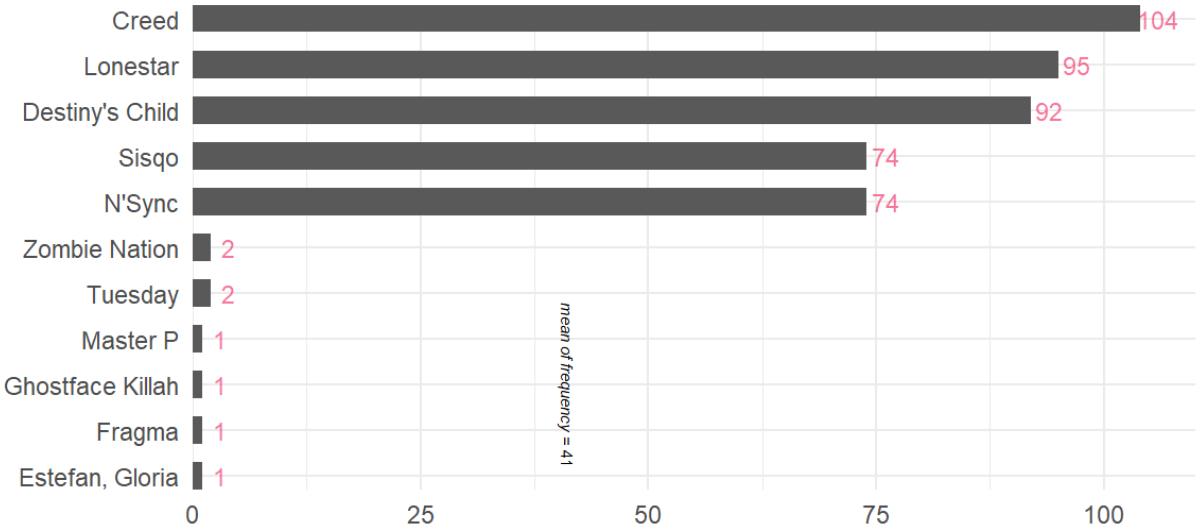
```
df %>%
  left_join(best, by = c('artist', 'track')) %>%
  ggplot(aes(week, rank, group = track, color = peak)) +
  geom_line(aes(size = peak), alpha = 0.6) +
  scale_y_reverse() +
  scale_color_manual(name = "Peak",
                     label = c('Best Hit', 'No Hit'),
                     values = c('red', 'grey30')) +
  scale_size_manual(name = "Peak",
                    label = c('Best Hit', 'No Hit'),
                    values = c(1, .25)) +
  labs(title = "The Trend Over The Period",
       x = "Week",
       y = "Rank") +
  theme_light()
```



```
df %>%
  count(artist, sort = TRUE) %>%
  slice(1:5, (n()-5):n()) %>%
  mutate(artist = fct_reorder(artist, n)) %>%
  ggplot(aes(n, artist)) +
  geom_col(width = 0.6) +
  geom_text(aes(label = n), nudge_x = 2, color = '#F11B59', alpha = .6, size = 3) +
  scale_x_continuous(limits = c(0, 110), expand = c(0, 0)) +
  scale_y_discrete(expand = c(0, 0)) +
  labs(title = paste0("<b><span style = 'font-size:20pt'>Artists Frequently",
                      "<span style = 'color:#F11B59;'> ",
                      "**Appear On Billboard**",
                      "</span></span></b>",
                      "<br><b><span style = 'font-size:14pt'>",
                      "*Top 5 and Bottom 5 Artists*",
                      "</span></b><br>"),
       caption = "RAuidt Solution LLP | Stewart Li",
       x = "",
       y = "") +
  ggtext::geom_richtext(data = . %>%
                          summarise(the_mean = round(mean(n)), 0),
                        aes(x = the_mean,
                            y = "Ghostface Killah",
                            label = glue::glue("*mean of frequency* = {the_mean}")),
                        fill = NA,
                        label.color = NA,
                        size = 2,
                        angle = -90) +
  theme_minimal() +
  theme(
    plot.margin = margin(35, 35, 10, 35),
    plot.title = ggtext::element_textbox_simple(
      size = 13,
      face = NULL,
      lineheight = 1.75,
      padding = margin(5, 5, 0, 5),
      margin = margin(0, 0, 0, 0),
      fill = "white"),
    plot.title.position = "plot",
    plot.caption = ggtext::element_textbox_simple(
      size = 10,
      lineheight = 1,
      padding = margin(10, 10, 10, 10),
      margin = margin(10, 0, 10, 0),
      fill = "#F5F5F5",
      halign = 0.5,
      valign = 0.5)
  )
```

Artists Frequently Appear On Billboard

Top 5 and Bottom 5 Artists



RAuidt Solution LLP | Stewart Li