hw 8

Cory Costello

November 17, 2017

Load Data

```
library(rio)
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr
```

```
## Conflicts with tidy packages ----------------------------------------------
```

```
## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
library(stringr)

breweries <- import("../../../Data/breweries.csv",
                    setclass = "tbl_df")

beers <- import("../../../Data/beers.csv",
                    setclass = "tbl_df")
```

1. Strong vs. Standard Ales

First, create a variable that codes if a beer is a strong style (Imperial or Double) or standard ale.

It looks like double and imperial are usually in style, but occassionally show up in the name and not the style (e.g., Imperial Pumpkin Stout is coded as a pumpkin beer, rather than an imperial). Going to search in style and name for imperial, because that seems to catch those cases that are imperial ales but not noted as such in style. However, I'll only search in style for double, because many beers have the word double in their name but are not truly double ales (e.g., Double Play Pilsner). This seemed like the best way to get those strong ales that aren't called imperials or doubles in style without getting too many false hits.

```
beers <- beers %>%
  mutate(strong_v_standard = ifelse(str_detect(style, "Double") | str_detect(style, "Imperial") |
                                    str_detect(name, "Imperial"),
                                    "Strong", "Standard")) %>%
  arrange(desc(strong_v_standard))
```

1a: How many beers in each category:

```
n_strong_v_standard <- beers %>%
  count(strong_v_standard)

n_strong_v_standard
```

```
## # A tibble: 2 x 2
##   strong_v_standard     n
##               <chr> <int>
## 1          Standard  2277
## 2            Strong   133
```

It looks like there are 2277 *standard* beers and 133 *strong* beers. So there are overwhelmingly more standard beers.

1b: Which brewery has the highest number of strong ales?

First, I need to merge the datasets. The brewery_id variable in the beers dataset is a foreign key for the breweries dataset. However, the brewery_id is called V1 in the breweries dataset (was able to figure this out because I'm familiar with the breweries; not sure if there was another way to figure that out). Also, both datasets contain a variable name, which corresponds to the brewery name and the beer name in the breweries and beers datasets respectively. So, I'll rename V1 brewery_id in the breweries dataset, rename name to brewery_name in the breweries dataset, then right_join it to the beers data (beers on the right; made sense in this pipeline), rename name to beer_name in the beers dataset, and finally change brewery_name to be lower case (in case some are entered in different cases).

```
beers <- breweries %>%
  rename (brewery_id = V1,
          brewery_name = name) %>%
  right_join(beers, by = "brewery_id") %>%
  rename(beer_name = name) %>%
  mutate(brewery_name = str_to_lower(brewery_name))
```

Second, I want to see if there are any breweries that have two names entered (this could happen if, for instance, a name was entered as capital in one place and lower case in another).

Third, I'll filter for beers that were categorized as strong, and then get a count of brewery names

```
brewery_most_strong <- beers %>%
  filter(strong_v_standard == "Strong") %>%
  count(brewery_name) %>%
  arrange(desc(n)) %>%
  slice(1)
brewery_most_strong
```

```
## # A tibble: 1 x 2
##          brewery_name     n
##                 <chr> <int>
## 1 oskar blues brewery    12
```

It looks like oskar blues brewery has the most strong (double/imperial) beers at 12 beers.

1c: Which state has the highest number of strong (imperial/double) ales?

```
state_highest_prop_strong <- beers %>%
  add_count(state) %>%
  add_count(state, strong_v_standard) %>%
  group_by(state, strong_v_standard) %>%
  summarize(n_strong_v_standard = sum(nn),
            n_beers_tot = sum(n)) %>%
  mutate (prop_strong_v_standard = n_strong_v_standard / n_beers_tot) %>%
  filter (strong_v_standard == "Strong") %>%
  ungroup() %>%
  arrange(desc(prop_strong_v_standard)) %>%
  slice(1)

state_highest_n_strong <- beers %>%
  add_count(state) %>%
  add_count(state, strong_v_standard) %>%
  group_by(state, strong_v_standard) %>%
  summarize(n_strong_v_standard = sum(nn),
            n_beers_tot = sum(n)) %>%
  mutate (prop_strong_v_standard = n_strong_v_standard / n_beers_tot) %>%
  filter (strong_v_standard == "Strong") %>%
  ungroup() %>%
  arrange(desc(n_strong_v_standard)) %>%
  slice(1)
```

It looks like the state with the greatest proportion of Strong (double/imperial) beers is VT; approximately 25.93% of VT's beers are in the strong category. However, the State with the greatest total number of strong beers is CO with a grand total of 441 strong beers. However, that only accounts for 7.92% of CO's beer.

Part 2: All about IPAs

First, subset IPAs using `str_detect()`. I'll filter for anything that has the string "IPA" in the style.

```
IPAs <- beers %>%
  filter(str_detect(style, "IPA"))
```

2a How many IPAs have IPA in their name?

```
ipa_in_name <- IPAs %>%
  mutate(
    beer_name = str_to_lower(IPAs$beer_name),
    ipa_in_name = ifelse(str_detect(beer_name, "ipa") == TRUE, 1, 0),
    ipa = 1) %>%
  summarize(ipa_in_name_tot = sum(ipa_in_name),
            ipa_tot = sum(ipa),
            ipa_in_name_prop = ipa_in_name_tot / ipa_tot)

ipa_in_name
```

```
## # A tibble: 1 x 3
```

```
##   ipa_in_name_tot ipa_tot ipa_in_name_prop
##             <dbl>   <dbl>            <dbl>
## 1             324     571        0.5674256
```

It looks like 324 IPAs have IPA somewhere in the name, which amounts to 56.74% of all IPAs in this sample. So, the majority of IPAs have the word IPA in the name, making it easy to pick them out.

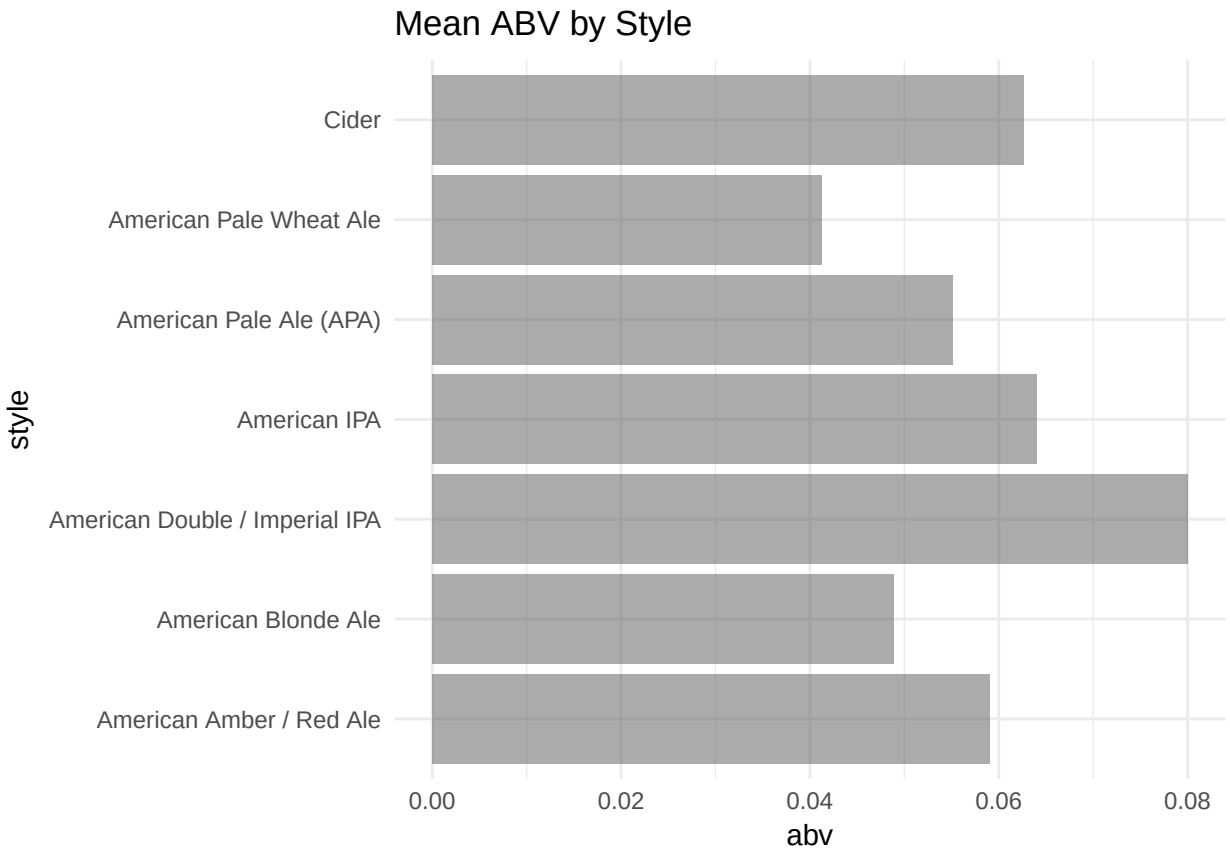
```
state_most_ipas <- IPAs %>%
  count(state) %>%
  arrange(desc(n)) %>%
  slice(1)

state_most_ipas
```

```
## # A tibble: 1 x 2
##   state     n
##   <chr> <int>
## 1    CA    58
```

It looks like CA has the most IPAs at 58.

```
beers %>%
  filter(state == "OR") %>%
  filter(style == "Cider" |
           str_detect(style, "IPA") &
          !str_detect(style, "White")|
           str_detect(style, "American") &
           str_detect(style, "Ale") &
          !str_detect(style, "Black") &
          !str_detect(style, "Strong")) %>%
  group_by(style) %>%
  summarize(abv = mean(abv, na.rm = TRUE)) %>%
  ggplot(aes(x = style, y = abv)) +
  geom_bar(stat = "identity", alpha = .5) +
  coord_flip() +
  ggtitle("Mean ABV by Style") +
  theme_minimal()
```


Mean ABV by Style
Cider
American Pale Wheat Ale
American Pale Ale (APA)
American IPA
American Double / Imperial IPA
American Blonde Ale
American Amber / Red Ale
style
0.00
0.02
0.04
0.06
0.08
abv