Buzz vs. Bite

An Analysis of Alcohol Content and Bitterness in American Craft Beers

June 26, 2018

Introduction

It goes without saying that entering the craft beer market, in pretty much any state, is a monstrous task. The explosion of micro-breweries has spread expeditiously in nearly every rapidly-growing urban environment. Luckily for you, the explosion is not in isolation. Demand has never been higher for unique and complex alcoholic libations. Where once the thought process was "the simpler the better," newer generations are constantly on the hunt for a drinking experience that fits their lifestyle and temperament. Although the market may seem saturated in many areas, it is, in reality, a rich field filled with almost never-ending demand ready to be tapped by the right combination of ingenuity, experimentation and a knowledge of what sells.

Purpose of this study

Purpose of this study * To organize and analyze a list of 2410 craft beers from the United States and a list of 558 breweries. * To help you identify trends within this data to help narrow your focus for production. Manufacturing a beer that will outsell your competitors is more than just a quality product, it's knowing what quality is proven to sell. * To provide you with a functional list of each beer's alcohol content, bitterness level, style and other information to help you decide which direction to take your production facilities and supply chain.

Loading required libraries

The following code loads useful libraries that aren't included in base R. The of these libraries come from the "tidyverse" including dplyr for manipulating dataframes, tidyr for making data tidy, knitr for creating reproducible documents ggplot2 for plots, maps for help with geographic plots, RColorBrewer for improved map graphics, summarytools for summarizing data, magrittr for better code, and gridExtra to assist with plots

```
## Loading required package: 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
```

```
## Loading required package: tidyr
```

```
## Loading required package: knitr
```

```
## Loading required package: ggplot2
```

```
## Loading required package: maps
```

```
## Loading required package: RColorBrewer
## Loading required package: summarytools
```

```
## Loading required package: magrittr
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
## Loading required package: gridExtra
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

Breweries Data

Import Breweries

In this section we load and begin cleaning the data in order to aid our exploratory analysis. Column names are set to lowercase for ease of reading and we begin to summarize the data.

```r
#import breweries data
breweries_data <- read.csv("../data/Breweries.csv", header=TRUE)


colnames(breweries_data) %<>% tolower #lower case colnames

breweries_data %<>% rename(brewery_id = brew_id) #rename
```

Inspect Raw Breweries Dataset

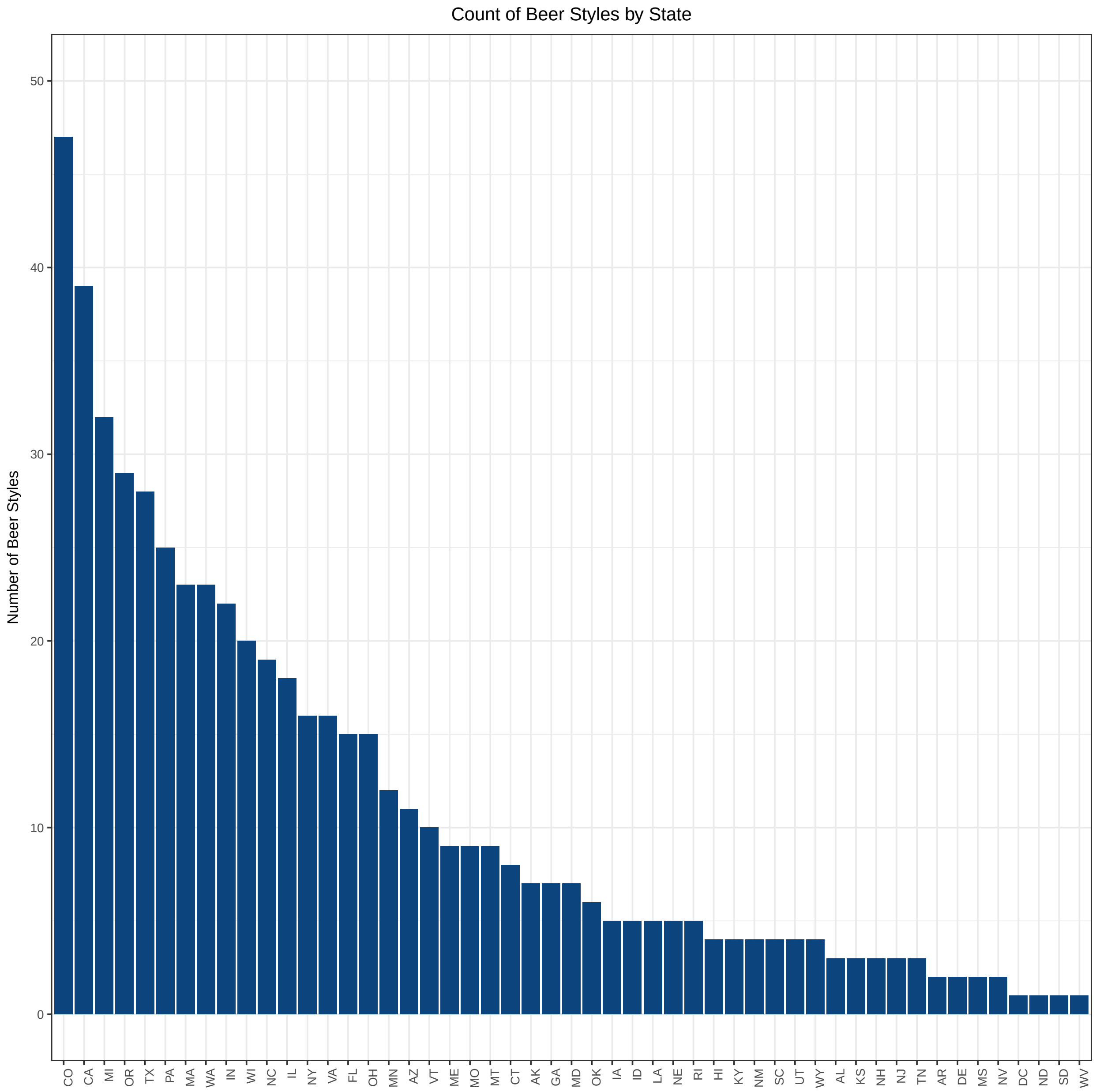
```r
# count breweries by state
brewery_summary_raw <- select(breweries_data, state, brewery_id) %>% #select columns
                   dplyr::group_by(state) %>% #group by
                   dplyr::summarize_all(funs(n_distinct(.))) %>%
                   arrange(desc(brewery_id))




# print top 5 states with the most breweries
kable(head(brewery_summary_raw, 5), digits = 0)
```

state	brewery_id
CO	47
CA	39
MI	32
OR	29
TX	28

```
# print bottom 5 states with the most breweries
kable(tail(brewery_summary_raw, 5), digits = 0)
```

state	brewery_id
NV	2
DC	1
ND	1
SD	1
WV	1

```
ggplot(brewery_summary_raw) +
      geom_bar(aes(x = reorder(state, -brewery_id, FUN=max),
                   y = brewery_id),
               stat ="identity",
               fill = misc_cool) +
      guides(fill=guide_legend(title= NULL)) +
      xlab(NULL) +
      ylab("Number of Beer Styles") +
      scale_y_continuous(breaks = c(0, 10, 20, 30, 40, 50),
                         limits = c(0, 50),
                         minor_breaks = c(5, 15, 25, 35 ,45)) +
      ggtitle("Count of Beer Styles by State") +
      theme(plot.title = element_text(hjust = 0.5)) + # center plot title
      theme(text = element_text(size=10),
            axis.text.x = element_text(angle=90, hjust=1)) # rotate x-axis labels
```



Clean Breweries Data

Before we can confidently proceed with our analysis it's important to ensure we have scrubbed the data, removed duplicates, and decide how we will deal with errors and missing values.

We start this process by removing punctuation and whitespace from columns. Humans are fallible and typos are easy to make. Without knowing the origin of the data in the files provided, its prudent to assume that mistakes have been made and take measures to correct them.

Remvoing punctuation allows us to mitigate the possibility of commas being erroneously typed as periods. "Detroit, MI", for example, would be identified as a different city than "Detroit. MI" Removing punctuation resolves this issue. Both city/state combinations simply become "Detroit MI."

Likewise, it's helpful to remove whitespace. Although whitespace can appear "invisible" to the human eye, computers can “see” this space as if it were a number or a letter.

We use the apply function to make these changes to every row in the dataframe.

Removing duplicates is more of a challenge. Before we can remove duplicates we need to confirm whether or not two rows are the same. We identify duplicates by creating a unique key for each brewery that’s a combination of the brewery ID, city, and state.

De-duplicating in this case is a multi-step process. We start by identifying brewery ids that show up more than once which indicate possible duplicates. Further investigation determines whether or not they are actually duplicates.

In addition to removing identifying and removing duplicates programatically, we also need to correct a few entries manually. There are some entries that are clearly mis-spelled and need to be addressed.

Once potential duplicates are identified and assigned temporary keys, they are evaluated apart from the main dataset and returned to the main dataset once duplicates have been removed.

1) Remove punctuation and trim whitespace

```
# remove punctionation from all columns and trim whitespace
breweries_data <- as.data.frame(
                      apply(breweries_data #data set
                            , 2 #apply function column-wise
                            , function(x) trimws(gsub('[[:punct:] ]+',' ',x))) #anonymous function to r
                            , stringsAsFactors = FALSE)  #do not implicitly convert strings to factors


breweries_clean <- distinct(breweries_data, brewery_id, .keep_all = TRUE) %>% rename(brewery_name = name
```

2) Configure column types

```
breweries_data$name <- as.factor(breweries_data$name) # convert Name column to factor
breweries_data$brewery_id <- as.integer(breweries_data$brewery_id) # convert Brew_ID to integer
```

3) Identify and capture potential duplicate records

```
# confirm Brew_ID + City + State is a unique key
breweries_summary <-
  select(breweries_data, brewery_id, city, state, name) %>%
  group_by(name) %>%
  summarize_all(funs(
    count = n_distinct(brewery_id, city, state))) %>%
  select(name, brewery_id_count) %>% # select only Name and Brew_ID_count columns
  arrange(desc(brewery_id_count)) # sort by Brew_ID_count desc


# capture potential duplicates
breweries_dups <- filter(breweries_summary, brewery_id_count > 1) # if Brew_ID_count > 1 then there is

# rejoin potential dups to original dataset
breweries_dups <- select(breweries_dups %>% inner_join(breweries_data, by="name"), -ends_with("_count"))

breweries_dups
```

```
## # A tibble: 14 x 4
##    name                      brewery_id city         state
##    <fctr>                         <int> <chr>        <chr>
##  1 Blackrocks Brewery               13 Marquette    MI
##   2 Blackrocks Brewery              96 Marquette    MA
##   3 Blue Mountain Brewery          383 Afton        VA
##   4 Blue Mountain Brewery          415 Arrington    VA
##   5 Lucette Brewing Company        378 Menominee    WI
##   6 Lucette Brewing Company        457 Menominie    WI
##   7 Oskar Blues Brewery            167 Longmont     CO
##   8 Oskar Blues Brewery            504 Lyons        CO
##   9 Otter Creek Brewing            262 Waterbury    VT
##  10 Otter Creek Brewing            276 Middlebury   VT
##  11 Sly Fox Brewing Company        164 Phoenixville PA
##  12 Sly Fox Brewing Company        372 Pottstown    PA
##  13 Summit Brewing Company          59 St Paul      MN
##  14 Summit Brewing Company         139 St Paul      MN
```

4) Correct duplicates

    - City name “Menominie” misspelled as “Menominee”

```
# Fix Brew_ID=378, change City(Menominee -> Menominie)
breweries_dups <- breweries_dups %>%
     mutate(city=replace(city, brewery_id==378, "Menominie")) %>%
     as.data.frame()
```

```
* Marquette is not a city name in Massachusetts. Changed to Michigan based on other records existing fo
```

```
# Fix Brew_ID=96, change State(MA -> MI)
breweries_dups <- breweries_dups %>%
     mutate(state=replace(state, brewery_id==96, "MI")) %>%
     as.data.frame()
```

```
* Merge duplicates into single records on name + city + state
```

```
# group corrected duplicates to
breweries_dups <- breweries_dups %>%
                  group_by(name, city, state) %>%
                  filter(n()>1)
```

```
* Create new brewery_id for corrected duplicates
```

```
# create surrogate keys for duplicates
breweries_sk <- breweries_dups %>%
                    group_by(name, city, state) %>%
                    summarize_all(funs(
                        brew_sk = (sum(brewery_id)*sum(brewery_id)),
                        count = n()
                        )) %>%
                    ungroup() %>%
                    right_join(breweries_dups, by = c("name", "city", "state")) %>% # rejoin to dupes b
                    rename(old_brewery_id=brewery_id, new_brewery_id=brew_sk)


breweries_sk
```

```
## # A tibble: 6 x 6
##   name                     city      state new_brewery_id count old_brewer~
##   <fctr>                   <chr>     <chr>          <int> <int>       <int>
## 1 Blackrocks Brewery       Marquette MI             11881     2          13
## 2 Blackrocks Brewery       Marquette MI             11881     2          96
## 3 Lucette Brewing Company Menominie WI            697225     2         378
## 4 Lucette Brewing Company Menominie WI            697225     2         457
## 5 Summit Brewing Company  St Paul   MN             39204     2          59
## 6 Summit Brewing Company  St Paul   MN             39204     2         139
```

* Update original breweries dataset with corrections and de-duped records

```
# create cleaned dataset

breweries_clean <- dplyr::bind_rows( # append rows of dataframes together
                          breweries_data %>%
                            filter(!brewery_id %in% breweries_sk$old_brewery_id), # remove duplicated r
                          breweries_sk %>%
                            rename(brewery_id=new_brewery_id) %>% # rename new_brewery_id to brewery_id
                            select(-count, -old_brewery_id)) %>% # remove extra columns from brewery_sk
                          rename(brewery_name = name) %>% #change column name "name" to "brewery_name"
                          mutate(state = as.factor(state))

summarytools::dfSummary(breweries_clean)
```

```
## Data Frame Summary
## breweries_clean
## N: 558
## ----------------------------------------------------------------------------------------------------
## No   Variable         Stats / Values                     Freqs (% of Valid)   Text Graph          Vali
## ---- ---------------- ---------------------------------- -------------------- ------------------ ---
## 1    brewery_id       mean (sd) : 2959.57 (41747.55)     555 distinct val.    :                  558
##      [integer]        min < med < max :                                       :                  (10
##                       1 < 283.5 < 697225                                      :
##                       IQR (CV) : 279.5 (14.11)                                :
##                                                                               :
##                                                                               :
##
## 2    brewery_name     1. 10 Barrel Brewing Company       1 ( 0.2%)            IIIIIIIIIIIIIIII   558
##      [factor]         2. 18th Street Brewery             1 ( 0.2%)                               (10
##                       3. 2 Towns Ciderhouse              1 ( 0.2%)
##                       4. 21st Amendment Brewery          1 ( 0.2%)
##                       5. 3 Daughters Brewing             1 ( 0.2%)
##                       6. 4 Hands Brewing Company         1 ( 0.2%)
##                       7. 450 North Brewing Company       1 ( 0.2%)
##                       8. 7 Seas Brewing Company          1 ( 0.2%)
##                       9. 7venth Sun                      1 ( 0.2%)
##                       10. Abita Brewing Company          1 ( 0.2%)
##                       [ 541 others ]                     548 (98.6%)
##
## 3    city             1. Portland                        17 ( 3.0%)           IIIIIIIIIIIIIIII   558
##      [character]      2. Boulder                         9 ( 1.6%)                               (10
##                       3. Chicago                         9 ( 1.6%)
##                       4. Seattle                         9 ( 1.6%)
##                       5. Austin                          8 ( 1.4%)
##                       6. Denver                          8 ( 1.4%)
##                       7. San Diego                       8 ( 1.4%)
##                       8. Bend                            6 ( 1.1%)
##                       9. San Francisco                   5 ( 0.9%)
##                       10. Anchorage                      4 ( 0.7%)
##                       [ 372 others ]                     475 (85.5%)
##
## 4    state            1. AK                              7 ( 1.2%)            I                  558
##      [factor]         2. AL                              3 ( 0.5%)            I                  (10
##                       3. AR                              2 ( 0.4%)            IIIIIIIIIIIIIIII
##                       4. AZ                              11 ( 2.0%)
##                       5. CA                              39 ( 7.0%)
##                       6. CO                              47 ( 8.4%)
##                       7. CT                              8 ( 1.4%)
##                       8. DC                              1 ( 0.2%)
##                       9. DE                              2 ( 0.4%)
##                       10. FL                             15 ( 2.7%)
##                       [ 41 others ]                      423 (75.8%)
## ----------------------------------------------------------------------------------------------------
```

Clean Beer Data

A similar process is used to remove duplicates from the Beers dataset.

```
beer_data <- read.csv("../data/Beers.csv", header=TRUE)

head(beer_data)
```

```
##                   Name Beer_ID   ABV IBU Brewery_id
## 1            Pub Beer    1436 0.050  NA        409
## 2         Devil's Cup    2265 0.066  NA        178
## 3 Rise of the Phoenix    2264 0.071  NA        178
## 4            Sinister    2263 0.090  NA        178
## 5       Sex and Candy    2262 0.075  NA        178
## 6        Black Exodus    2261 0.077  NA        178
##                             Style Ounces
## 1            American Pale Lager     12
## 2        American Pale Ale (APA)     12
## 3                   American IPA     12
## 4 American Double / Imperial IPA     12
## 5                   American IPA     12
## 6                  Oatmeal Stout     12
```

```
nrow(beer_data)
```

```
## [1] 2410
```

```
colnames(beer_data) %<>% tolower #lower case colnames





beer_clean <- merge(beer_data, # first data frame
             (breweries_sk %>% select(old_brewery_id, new_brewery_id)), # second data frame
             by.x = "brewery_id", # left table join key
             by.y = "old_brewery_id", # right table join key
             all = T) %>% # keep all columns
       mutate(brewery_id = ifelse(!is.na(new_brewery_id), new_brewery_id, brewery_id)) %>%  # update br
       select(-new_brewery_id) %>% # drop new_brewery_id column
       rename(beer_name = name)


summarytools::dfSummary(beer_clean)
```

```
## Data Frame Summary
## beer_clean
## N: 2410
## ----------------------------------------------------------------------------------------------------
## No   Variable       Stats / Values                     Freqs (% of Valid)      Text Graph
## ---- -------------- ---------------------------------- ----------------------- ---------------------
## 1    brewery_id     mean (sd) : 1488.62 (28425.37)     555 distinct val.       :
##      [integer]      min < med < max :                                          :
##                     1 < 207 < 697225                                           :
##                     IQR (CV) : 273.5 (19.1)                                    :
##                                                                                :
##                                                                                :
##
## 2    beer_name      1. #001 Golden Amber Lager         1 ( 0.0%)               IIIIIIIIIIIIIIII
##      [factor]       2. #002 American I.P.A.            1 ( 0.0%)
##                     3. #003 Brown & Robust Porter      1 ( 0.0%)
##                     4. #004 Session I.P.A.             1 ( 0.0%)
##                     5. #9                              2 ( 0.1%)
##                     6. 077XX                           1 ( 0.0%)
##                     7. 10 Degrees of Separation        1 ( 0.0%)
##                     8. 10 Ton                          1 ( 0.0%)
##                     9. 113 IPA                         1 ( 0.0%)
##                     10. 11th Hour IPA                  1 ( 0.0%)
##                     [ 2295 others ]                    2399 (96.0%)
##
## 3    beer_id        mean (sd) : 1431.11 (752.46)       2410 distinct val.              : : . : : . :
##      [integer]      min < med < max :                                          .   : : : : : : : : :
##                     1 < 1453.5 < 2692                                          : . : : : : : : : : :
##                     IQR (CV) : 1267.5 (0.53)                                   : : : : : : : : : : :
##                                                                                : : : : : : : : : : :
##                                                                                : : : : : : : : : : :
##
## 4    abv            mean (sd) : 0.06 (0.01)            74 distinct val.                : .
##      [numeric]      min < med < max :                                                  : :
##                     0 < 0.06 < 0.13                                                    : : .
##                     IQR (CV) : 0.02 (0.23)                                             : : :
##                                                                                        : : :
##                                                                                      : : : : : . .
##
## 5    ibu            mean (sd) : 42.71 (25.95)          107 distinct val.         :
##      [integer]      min < med < max :                                            :
##                     4 < 35 < 138                                                 : :     .
##                     IQR (CV) : 43 (0.61)                                       . : : .   :
##                                                                                : : : : . : .
##                                                                                : : : : : : : : :
##
## 6    style          1.                                 5 ( 0.2%)               I
##      [factor]       2. Abbey Single Ale                2 ( 0.1%)               IIIIIIIIIIIIIIII
##                     3. Altbier                         13 ( 0.5%)
##                     4. American Adjunct Lager          18 ( 0.8%)
##                     5. American Amber / Red Ale        133 ( 5.5%)
##                     6. American Amber / Red Lager      29 ( 1.2%)
##                     7. American Barleywine             3 ( 0.1%)
##                     8. American Black Ale              36 ( 1.5%)
##                     9. American Blonde Ale             108 ( 4.5%)
##                     10. American Brown Ale             70 ( 2.9%)
##                     [ 90 others ]                      1993 (82.7%)
##
## 7    ounces         mean (sd) : 13.59 (2.35)           8.4 :    1 ( 0.0%)      IIIIIIIIIIIIIIII
##      [numeric]      min < med < max :                  12 : 1525 (63.3%)       IIIIIIII
##                     8.4 < 12 < 32                      16 :  841 (34.9%)
##                     IQR (CV) : 4 (0.17)                16.9 :    1 ( 0.0%)
##                                                        19.2 :   15 ( 0.6%)
##                                                        24 :   22 ( 0.9%)
##                                                        32 :    5 ( 0.2%)
## ----------------------------------------------------------------------------------------------------
```

Question 1

To determine the number breweries in each state we simply count the number of times each state appears in the table.

The prominent brewing states with twenty or more breweries include: Colorado 47, California 39, Michigan 32, Oregon 29, Texas 28, Pennsylvania 25, Massachusetts 23, Washington 23, Indiana 22, Wisconsin 20.

These prominent brewing states are important to the beer market, because of their distinct beer types and styles that are produced in state and consumed nationally.

What kinds of beers and their characteristics will be of great interest for the analysis.

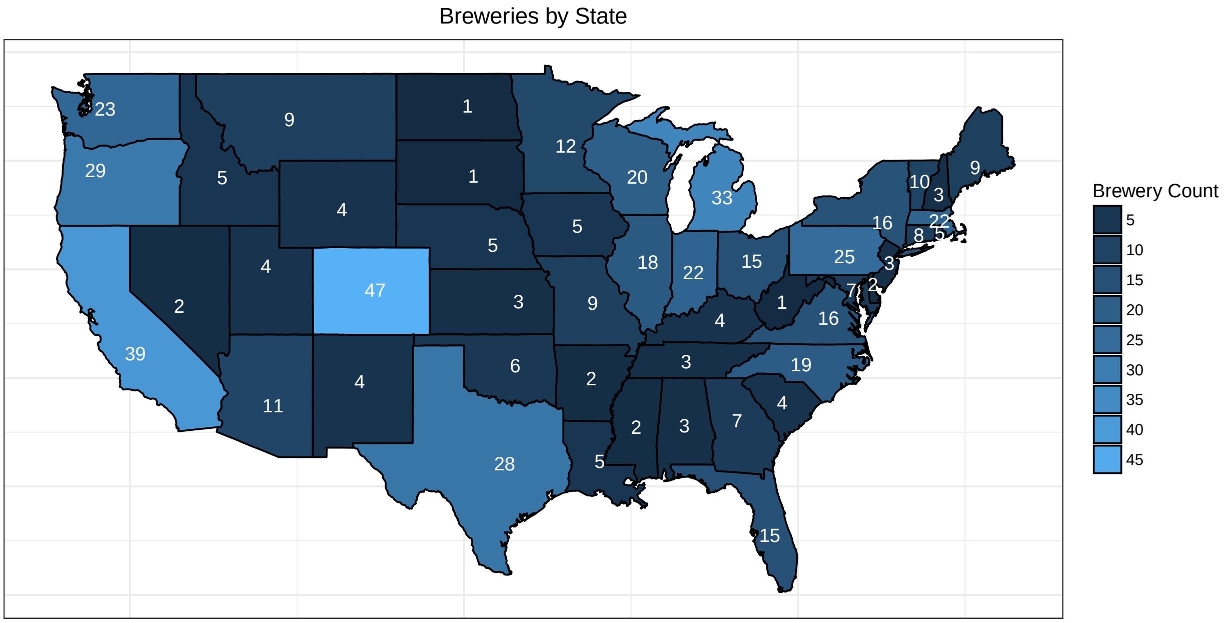
```r
#TODO: break up chunk

state_ll <- read.csv("../data/state_coords.csv") %>%
                    mutate(State = toupper(State)) %>%
                    rename(name = State, lat_center = Latitude, lon_center = Longitude)


states <- map_data("state") %>%
          mutate(region = toupper(region)) %>%
          rename(name=region) %>%
          select(long, lat, name, group)

# states %>% group_by(name) %>%
#              summarise_all(funs(n=n()))
#
#
states <- states %>%
          left_join(
            states %>%
            group_by(name) %>%
            summarise_all(funs(n=n())) %>%
            select(name, group_n) %>%
            distinct(name, .keep_all = TRUE)
          )
```

```r
breweries_by_state <- select(breweries_clean, brewery_id, state) %>%
  group_by(state) %>%
  summarise_all(funs(brewery_count = n()))  %>%
  left_join(state_ll, by=c("state" = "Abbr"))


# state_ll %>%
#   inner_join(states)


summarytools::dfSummary(breweries_by_state, transpose = TRUE)
```

```
## Data Frame Summary
## breweries_by_state
## N: 51
## ----------------------------------------------------------------------------------------------------
## No   Variable          Stats / Values                 Freqs (% of Valid)   Text Graph
## ---- ----------------- ------------------------------ -------------------- -------------------------
## 1    state             1. AK                          1 ( 2.0%)            IIIIIIIIIIIIIII
##      [character]       2. AL                          1 ( 2.0%)
##                        3. AR                          1 ( 2.0%)
##                        4. AZ                          1 ( 2.0%)
##                        5. CA                          1 ( 2.0%)
##                        6. CO                          1 ( 2.0%)
##                        7. CT                          1 ( 2.0%)
##                        8. DC                          1 ( 2.0%)
##                        9. DE                          1 ( 2.0%)
##                        10. FL                         1 ( 2.0%)
##                        [ 41 others ]                  41 (80.4%)
##
## 2    brewery_count     mean (sd) : 10.94 (10.63)      25 distinct val.     :
##      [integer]         min < med < max :                                   :
##                        1 < 7 < 47                                          :
##                        IQR (CV) : 12.5 (0.97)                              :
##                                                                            : :
##                                                                            : : : : : .
##
## 3    name              1. ALABAMA                     1 ( 2.0%)            IIIIIIIIIIIIIII
##      [character]       2. ALASKA                      1 ( 2.0%)
##                        3. ARIZONA                     1 ( 2.0%)
##                        4. ARKANSAS                    1 ( 2.0%)
##                        5. CALIFORNIA                  1 ( 2.0%)
##                        6. COLORADO                    1 ( 2.0%)
##                        7. CONNECTICUT                 1 ( 2.0%)
##                        8. DELAWARE                    1 ( 2.0%)
##                        9. FLORIDA                     1 ( 2.0%)
##                        10. GEORGIA                    1 ( 2.0%)
##                        [ 40 others ]                  40 (80.0%)
```

```
##
## 4    lat_center        mean (sd) : 39.48 (6.13)       50 distinct val.             :
##      [numeric]         min < med < max :                                           :
##                        21.09 < 40 < 61.37                                        : :
##                        IQR (CV) : 7.52 (0.16)                                  . : :
##                                                                                : : :
##                                                                            . . : : : :     .
##
## 5    lon_center        mean (sd) : -93.67 (19.34)     50 distinct val.                   : .
##      [numeric]         min < med < max :                                               : : :
##                        -157.5 < -89.65 < -69.38                                        : : :
##                        IQR (CV) : 23.84 (-0.21)                                    :   : : :
##                                                                                    : . : : :
##                                                                            :     : : : : : : .
## ----------------------------------------------------------------------------------------------------
```

```
#map of breweries by state

#one to many join of breweries by state
breweries_geo <- breweries_by_state %>%
                  inner_join(states, by = c("name" = "name"))

# map chart of brweeries_by_state
ggplot((breweries_geo %>% arrange(desc(brewery_count))),
       aes(group = state, stat="identity")) +
  geom_polygon(aes(x = long,
                   y = lat,
                   group=group,
                   fill=brewery_count),
               color = "black") +
  geom_text(data = (breweries_by_state %>%
                    filter(!(state %in% c("AK", "DC", "HI")))), #filter to continental 50 states
            aes(x = lon_center,
                y = lat_center,
                label = as.character(brewery_count)),
            color = 'white'
            ) +
  guides(fill=guide_legend(title= "Brewery Count")) +
  scale_fill_continuous(breaks = seq(0,50, by = 5)) +
  coord_fixed(1.3) + # fix lat/long display ratio

  ggtitle("Breweries by State") + # set plot title
  theme(plot.title = element_text(hjust = 0.5)) + # center plot title
  theme(legend.position = "right",
        axis.title.x=element_blank(), # hide x axis title
        axis.text.x=element_blank(),  # hide x axis text
        axis.ticks.x=element_blank(), # hide x axis ticks
        axis.title.y=element_blank(), # hide y axis title
        axis.text.y=element_blank(),  # hide y axis text
        axis.ticks.y=element_blank()) # hide y axis ticks
```



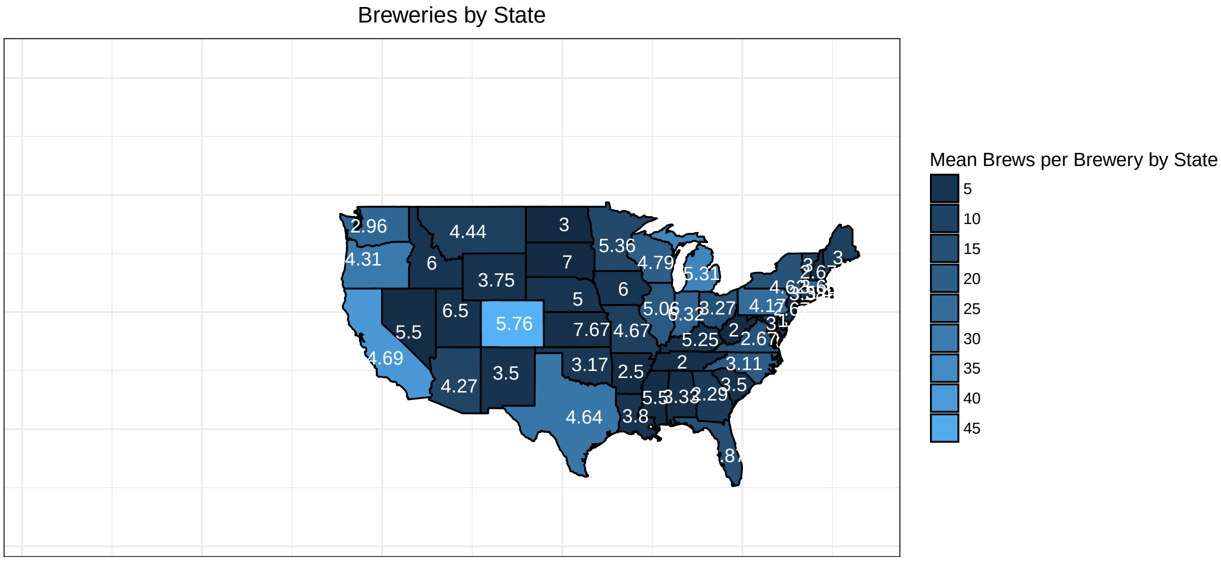
Question 2

In data science, like in life, sometimes less is more. Instead of maintaining separate tables for breweries and beers, it's helpful to merge the two datasets into a single combined dataset.

We do this by joining the two tables by the Brew_ID variable to create a new object variable named merged_data, which allows us to view the desired characteristics of beers produced by the prominent breweries.

Those characteristics will be of high importance for the analysis of most deisired beers, and we can begin to get clues of the prominent beers when we programmatically arrange these data by the most frequent style name variable sorted by every brewery.

```
# merge beer and breweries
merged_data <- breweries_clean %>%
               full_join(beer_clean, by="brewery_id")



x <- select(merged_data, brewery_name, state) %>%
    group_by(state) %>%
    summarise_all(funs(brews=n(), breweries = n_distinct(brewery_name))) %>%
                  inner_join(state_ll, by = c("state" = "Abbr"))


#TODO: Plot -> brews by brewery

ggplot((breweries_geo %>% arrange(desc(brewery_count))),
       aes(group = state, stat="identity")) +
  geom_polygon(aes(x = long,
                   y = lat,
                   group=group,
                   fill=brewery_count),
               color = "black") +
  geom_text(data = x,
            aes(x = lon_center,
                y = lat_center,
                label = as.character(round((brews/breweries),2))),
            color = 'white'
            ) +
  guides(fill=guide_legend(title= "Mean Brews per Brewery by State")) +
  scale_fill_continuous(breaks = seq(0,50, by = 5)) +
  coord_fixed(1.3) + # fix lat/long display ratio
  ggtitle("Breweries by State") + # set plot title
  theme(plot.title = element_text(hjust = 0.5)) + # center plot title
  theme(legend.position = "right",
        axis.title.x=element_blank(), # hide x axis title
        axis.text.x=element_blank(),  # hide x axis text
        axis.ticks.x=element_blank(), # hide x axis ticks
        axis.title.y=element_blank(), # hide y axis title
        axis.text.y=element_blank(),  # hide y axis text
        axis.ticks.y=element_blank()) # hide y axis ticks
```



```
#TODO: FILTER to contiguous 50 states
```

Question 3

Sometimes data are not available. This analysis is no exception. To better understand how our analysis could be impacted by missing values we first have to identify and county them.

These missing values would interfere with our analysis of center for the numeric variables and frequency of our factor and character variables. Once the missing values are removed, we can use the clean data to conduct the descriptive and quantitative analysis.

Below is a count missing values by variable.

```
# Number of nulls in each column
merged_data %>%
  select_if(function(x) any(is.na(x))) %>%
  summarise_all(funs(sum(is.na(.))))
```

```
##   abv  ibu
## 1  62 1012
```

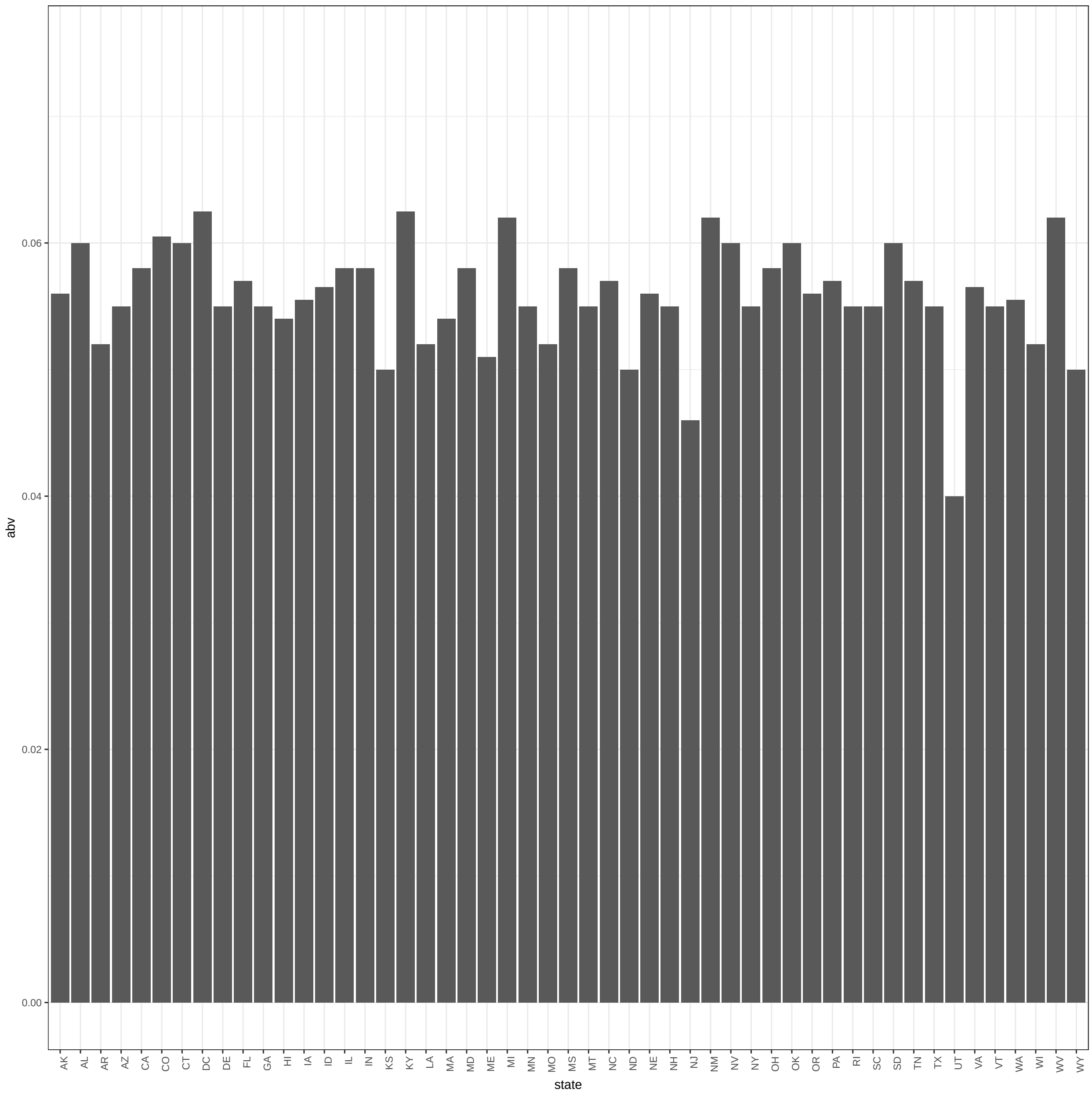
Question 4

Computing the median is straightforward. We simply merge all of the cleaned data by state, and calculate the median ABV and IBU for each state, which is summarized into a table and plotted as bar charts recording the median values across each state side by side.

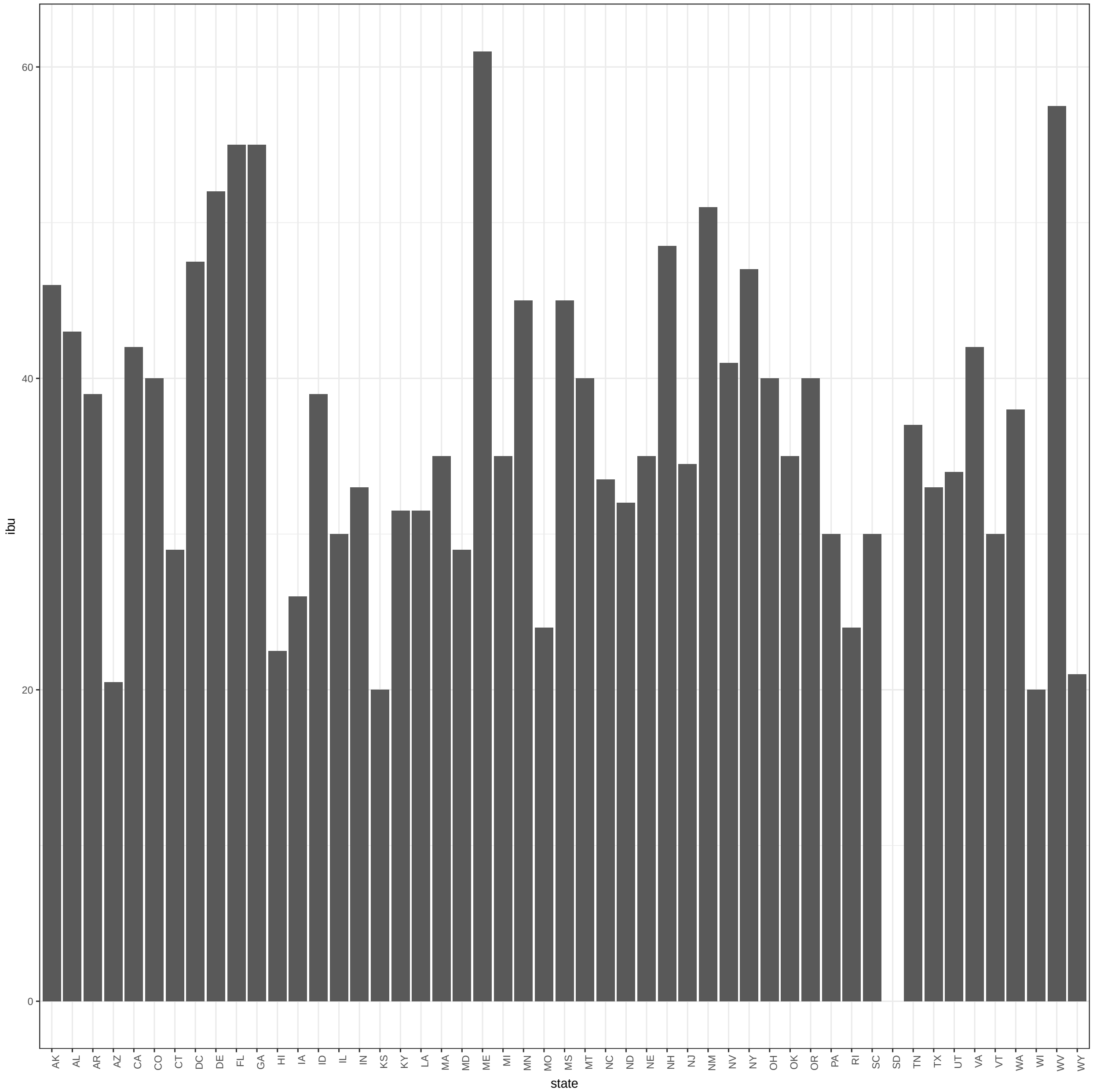
This plot is benificial to the analysis, because it gives insight into the beer characteristics of the prominent brewing states and the other states with less than twenty breweries. The prominent brewing states all share high ABV and IBU values, which brings more evidence to investigate for the analysis.

Are high ABV and IBU values always a characteristic of highly demanded beers in the respective market? We will have to produce a plausable claim and conduct a hypothesis test of that claim after further investigation.

```
#TODO: Make bar plot pretty

merged_by_state <- select(merged_data, state, abv, ibu) %>%
                   group_by(state) %>%
                   summarise_all(median, na.rm = TRUE)#funs(median(!is.na(.)))) #TODO: Double check thi

merged_by_state$state <- as.factor(merged_by_state$state)

#kable(as.data.frame(summarytools::descr(beer_clean)),digits = 2)


#TODO: facet by state

ggplot(merged_by_state, aes(x=state, y=abv)) +
  geom_bar(stat = "identity", position = "dodge") +
  ylim(0, .075) +
  #facet_grid(state ~ .) +
  theme(text = element_text(size=10),
        axis.text.x = element_text(angle=90, hjust=1))
```



```r
ggplot(merged_by_state, aes(x=state, y=ibu)) +
  geom_bar(stat = "identity", position = "dodge") +
  #ylim(0, .075) +
  #facet_grid(state ~ .) +
  theme(text = element_text(size=10),
        axis.text.x = element_text(angle=90, hjust=1))
```



Question 5

Before you can "push the limits" you have to know what the limits are. We want to determine which state has the most alcoholic beer and which state has the most bitter beer.

This is relatively simple. We can determine this visually using boxplots and confirm programmatically by sorting the tables in descending order based on the values of interest.

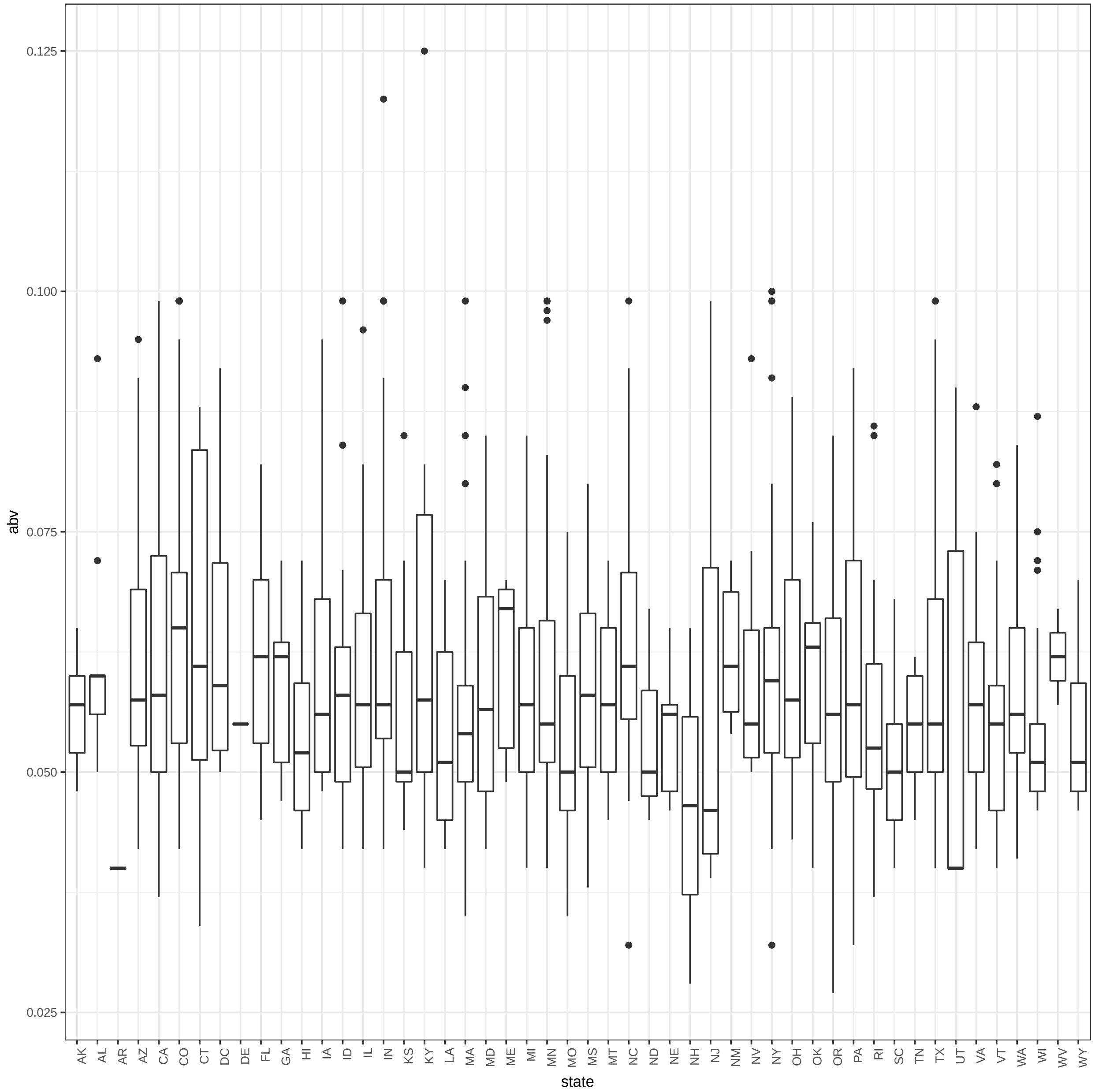
The state with the highest ABV value is Colorado with a 0.128 ABV value for the Lee Hill Series Vol. 5 - Belgian Style Quadrupel Ale beer that is a Quadrupel (Quad) style of beer brewed by the Upslope Brewing Company in Boulder, CO.

The state with the highest IBU value is Oregon with a 138 IBU value for the Bitter Bitch Imperial IPA beer that is a American Double / Imperial IPA style of beer brewed by the Astoria Brewing Company in Astoria, OR.

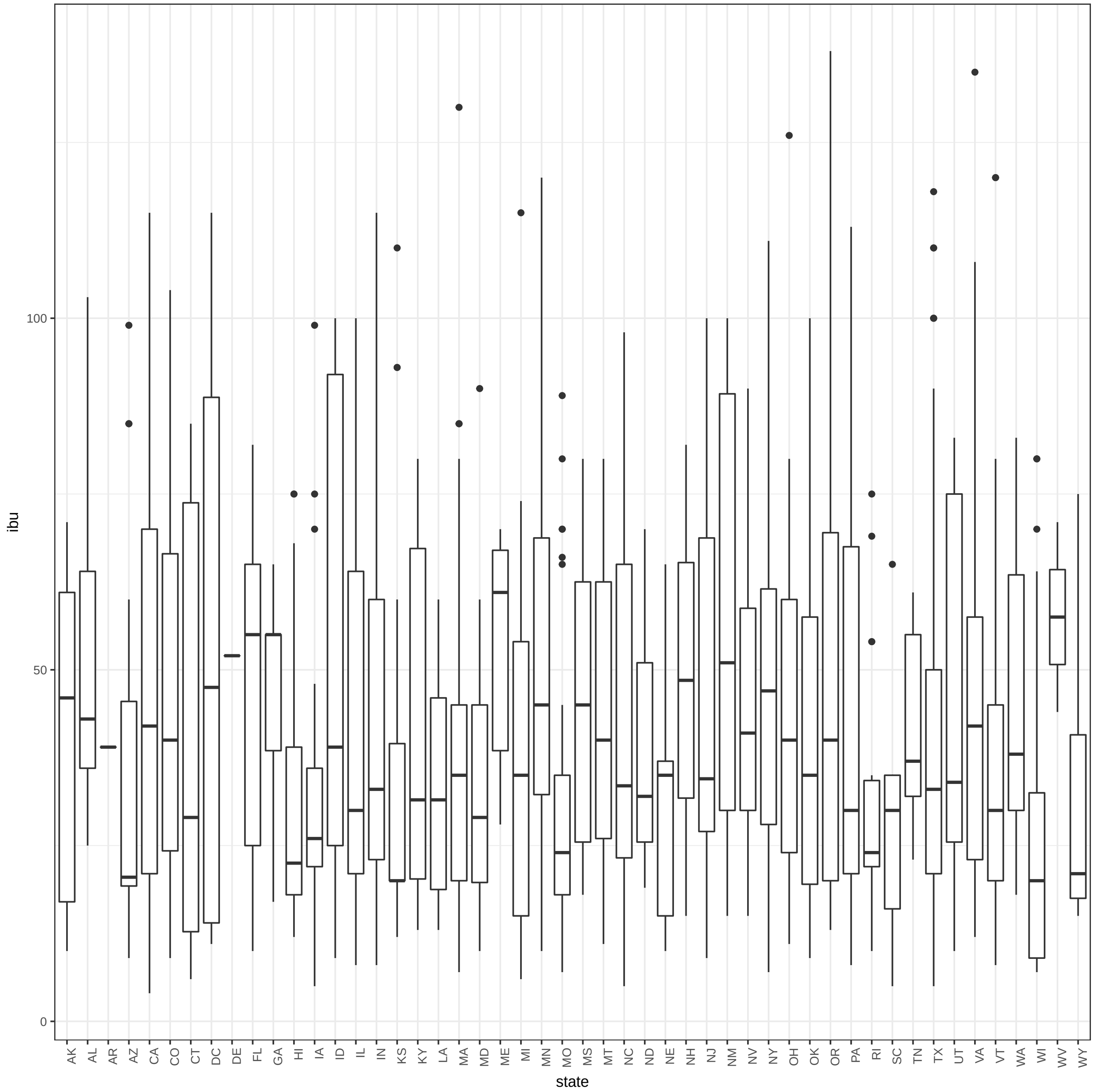
The states with the highest ABV and IBU values are found to be comprised of a majority of the prominent brewing states including the folling values State(maxABV, maxIBU):

Colorado(.128, 104), California(.099, 115), Michigan(.099, 115), Oregon(.082, 138), Texas(.099, 118), Pennsylvania(.099, 113), Massachusetts(.099, 130), Washington(.084, 83), Indiana(.120, 115), Wisconsin(.099, 80).

```
ggplot((merged_data %>% na.omit(abv)),
       aes(x=state , y=abv)) +  #TODO: Move to Appendix
  geom_boxplot() +
  #ylim(0, .075) +
  theme(text = element_text(size=10),
        axis.text.x = element_text(angle=90, hjust=1))
```



```
ggplot((merged_data %>% na.omit(ibu)),
       aes(x=state , y=ibu)) +  #TODO: Move to Appendix
  geom_boxplot() +
  #ylim(0, .075) +
  theme(text = element_text(size=10),
        axis.text.x = element_text(angle=90, hjust=1))
```



```
max_abv <-  (select(merged_data, state, abv) %>%
             group_by(state) %>%
             #filter(ABV == max(ABV)) %>%
             arrange(desc(abv))  %>% #sort by ABV
             filter(row_number() == 1))[1,] #get first row

max_abv
```

```
## # A tibble: 1 x 2
## # Groups: state [1]
##   state    abv
##   <fctr> <dbl>
## 1 CO     0.128
```

```r
max_ibu <-  (select(merged_data, state, ibu) %>%
                  group_by(state) %>%
                  #filter(ABV == max(ABV)) %>%
                  arrange(desc(ibu))  %>% #sort by ABV
                  filter(row_number() == 1))[1,] #get first row

max_ibu
```

```
## # A tibble: 1 x 2
## # Groups: state [1]
##    state    ibu
##   <fctr> <int>
## 1 OR       138
```

Question 6

The amount of alcohol by volume ABV is a good representation of the beer market, where consumer demand is infered from the geographical spread and number of breweries produce a certain style of beer. The certain style of a beer is controlled in-part by the ABV content. From the ABV five number summary we can better describe the use of the ABV variable as a controlling factor in the consumer market of beer.

The minimum ABV value of 0.001 is represented only one style of beer -Low Alcohol Beer produced by 1 brewery in CA:1. From these data we can infer the lack of consumer demand by the geographical spread of the style and by the lack of 0.001 ABV variability.

The first quartile(Q1) is represented by beers with a 0.050 ABV value represented by the -American -IPA and -Ale styles of beer that ranges from 5-100 in IBU. There are 38 different styles of beers with this ABV, that are produced by 141 different breweries in 46 different states AK:3, AL:1, AR:1, AZ:3, CA:10, CO:12, CT:3, DC:1, FL:5, GA:1, HI:1, IA:3, ID:2, IL:4, IN:3, KS:2, KY:1, LA:3, MA:4, MD:2, ME:2, MI:8, MN:3, MO:3, MS:1, MT:3, NC:4, ND:1, NE:1, NH:1, NJ:1, NV:1, NY:1, OH:6, Ok:1, OR:8, PA:5, RI:1, SC:1, TN:1, TX:5, UT:2, VA:2, WA:2, WI:7, WY:2. The use of a 0.050 ABV for a beer will allow for a large amount of variation with respect to the IBU of a beer style. The use of an ABV of 0.050 with any IBU between 5-100 will likely be a higly demanded beer by the consumer market.

The median ABV value of 0.056 represents the -American Ale and -Pale Ale styles of beer that ranges from 4-70 in IBU. There are 21 different styles of beers with this ABV, that are produced by 49 different breweries in 25 different states AK:1, AL:1, CA:5, CO:6, FL:1, IA:1, ID:1, IL:2, IN:2, KS:1, MA:3, MI:4, MN:2, MO:1, MT:1, NC:1, NE:1, NH:1, NY:1, OR:1, PA:4, TX:3, VA:2, WA:1, WI:2.

The third quartile(Q3) is represented by beers with a 0.067 ABV value represented by the -IPA and -Ale styles of beer that ranges from 33-85 in IBU. There are 10 different styles of beers with this ABV, that are produced by 22 different breweries in 15 different states AZ:1, CA:2, CO:3, MA:1, ME:1, MI:4, MN:2, NC:1, ND:1, NY:1, OH:1, OR:4, PA:1, WA:1, WV:1.

The maximum ABV value of 0.128 is represented only one style of beer -Quadrupel (Quad) that is produce by 3 breweries from 3 different states CO:1, IN:1, MI:1, and we can infer the demand of the consumer as a moderate demand by the breweries producing these beer being spread out geographically across the nation even though the style variability is very-low for the 0.128 ABV.

Summarizing the statistics for ABV can be accomplished in a signle command.

```r
#summaryize ABV

# tidy_summary <- tidy(summary(merged_data$ABV)) #For some reason this line wont knit

```

```
abv_stats <- as.data.frame(t(summary(merged_data$abv))) %>% #summarize and transpose
             rename("abv"=Freq, Statistic=Var2) %>%
             select(Statistic, abv)


abv_stats$abv <- round(abv_stats$abv, digits = 3)


abv_stats #TODO: Add IQR, stdev    #TODO: Compare to quinton's summary
```

```
##   Statistic    abv
## 1      Min.  0.001
## 2   1st Qu.  0.050
## 3    Median  0.056
## 4      Mean  0.060
## 5   3rd Qu.  0.067
## 6      Max.  0.128
## 7      NA's 62.000
```

Question 7

To determine the relationship between ABV and IBU it's helpful to see all values for both variables at the same time. This is most easily accomplished using a scatterplot.

Linear regression was used to model the relationship between ABV and IBU from a sample of cleaned data that was created in the previous questions above.

The equation:

```
y-intercept = IBU - slope * ABV
```

was used to plot thw linear model for the ABV and IBU data in this study.

With ABV on the x-axis and IBU on the y-axis, we start to see that there is a positive linear correlation between the ABV and IBU values, with R-squared = 0.44593.

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, and the definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by the linear model.

Since the creation, consumption, and distribution of beer by methods of breweries is a human behavior, it is very important to note that it is common for studies to measure R-squared values less than 0.50. The reason is , that human behavior is harder to predict with linear models.

The outcome of our study measured an R-squared value of 0.44593 which is an awesome fit, and much better than we expected for this study to produce, since the study is based on the human behavior of beer consumption with respect to the variation of ABV and IBU accross the United States of America.

Adding a trendline allows us to determine a formula that specifies this correlation. The regression is plotted and the results of the Spearman's Rank Correlation Test are in the following figures.

Spearman's Rank Correlation Results for rho

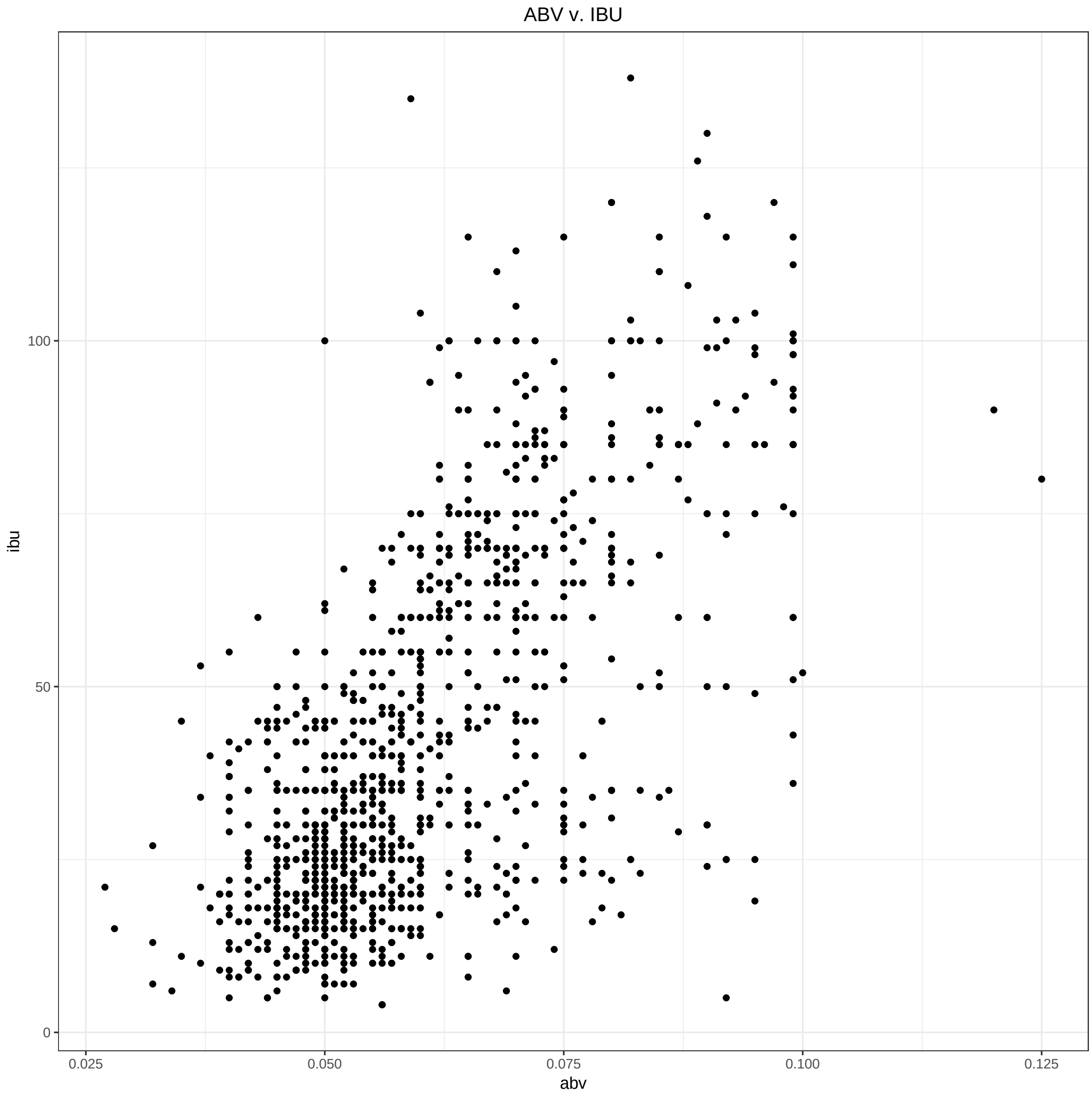
```
data:  styles$ibu and styles$abv
S = 153570000, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
```

```
sample estimates:
rho
0.6677798

[1] 0.4459299
```

```
#A distinct list of beer styles, as classified by a unique style Name, IBU, ABV, and Ounces values.

styles <- beer_clean %>%
              distinct(beer_id, style, ibu, abv, ounces) %>%
              arrange(style) %>%
              na.omit(ibu, abv)
```

- Plot ABV v. IBU

```
ggplot(styles, aes(x=abv, y=ibu)) +
  geom_point() +
  scale_colour_brewer() +
  ggtitle("ABV v. IBU") +
  theme(plot.title = element_text(hjust = 0.5))
```



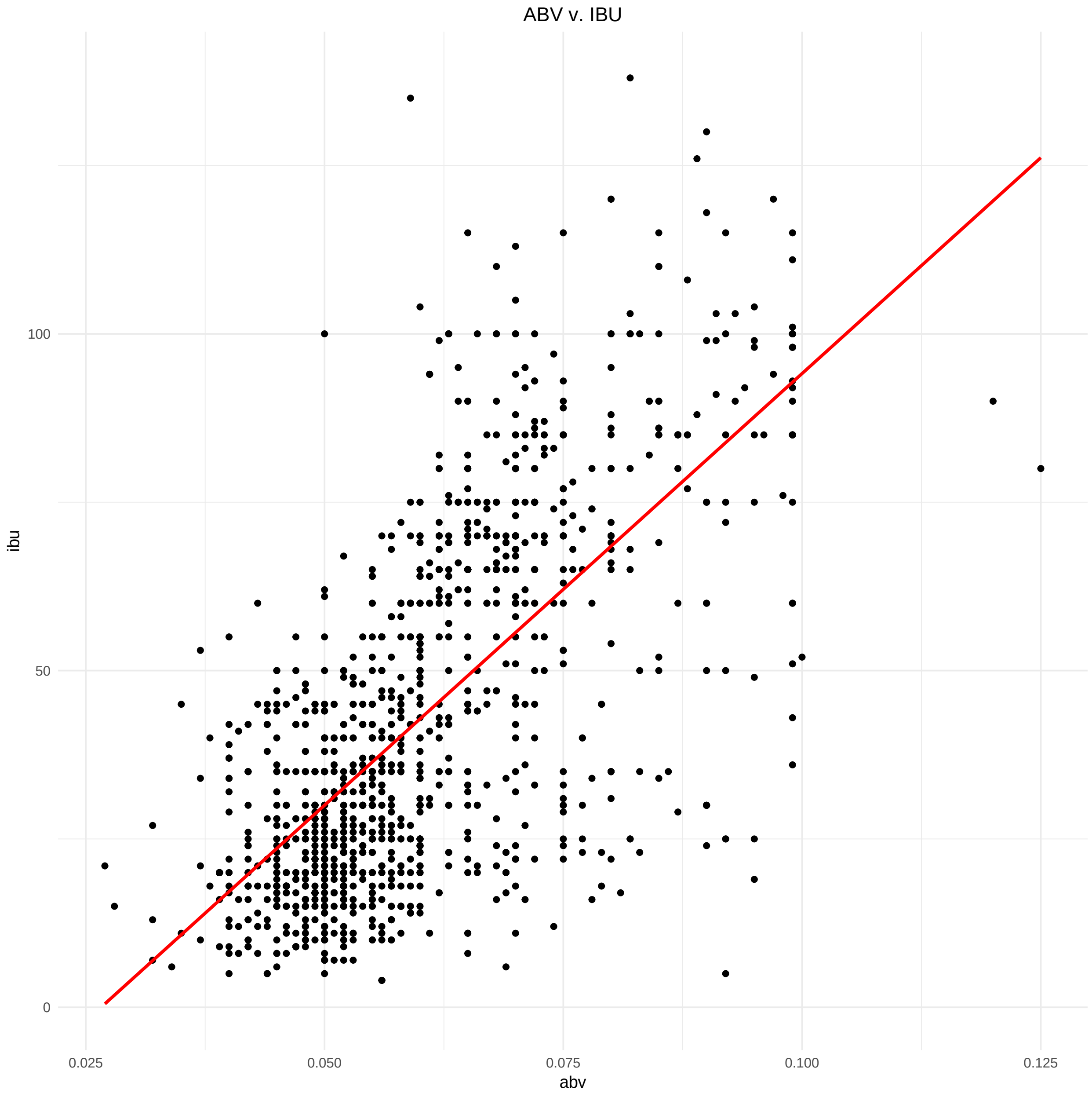
Analysis of ABV and IBU

```
+ More info on Spearman test: https://statistics.laerd.com/statistical-guides/spearmans-rank-order-corre
```

- Problem: We wish to test if there is a monotonic association between the alochol by volume (ABV) and international bitterness unit (IBU) rating of beers selected from domestic craft breweries.
- Hypotheses:
  - H_o: $\rho = 0$
  - H_A: $\rho \neq 0$
- Assumptions:
  - Continuity of data:/cmark

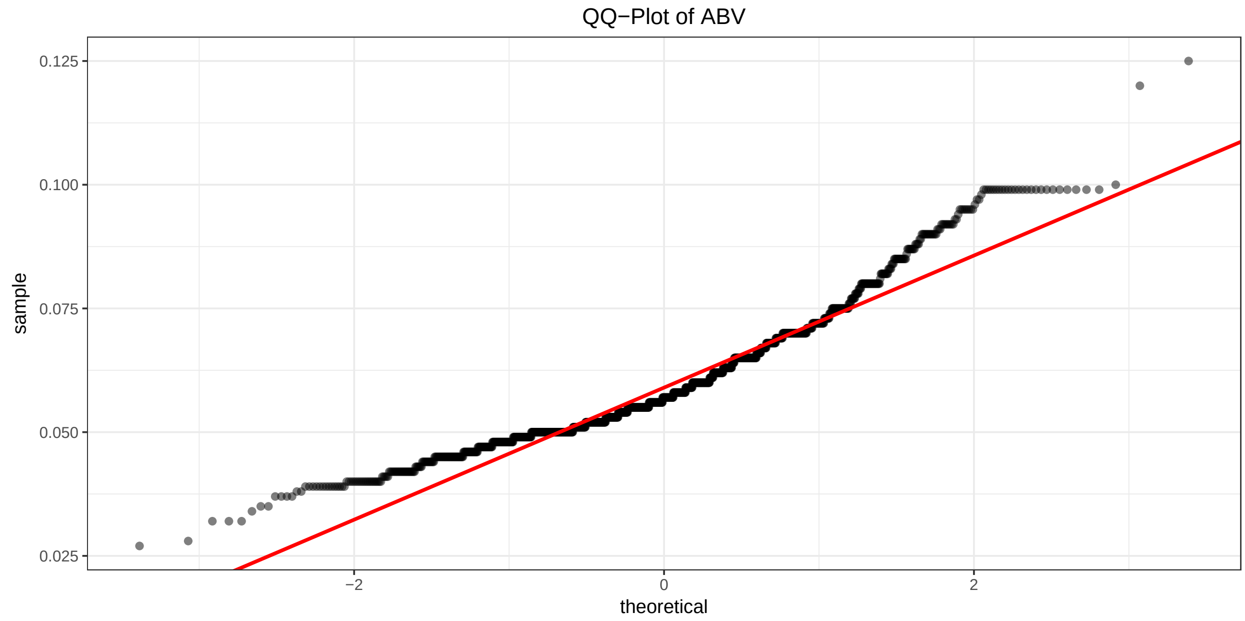
- Paired observations: ✔
- Data has linear relationship: ✔
- No significant outliers: ✗
- Normality: ✗

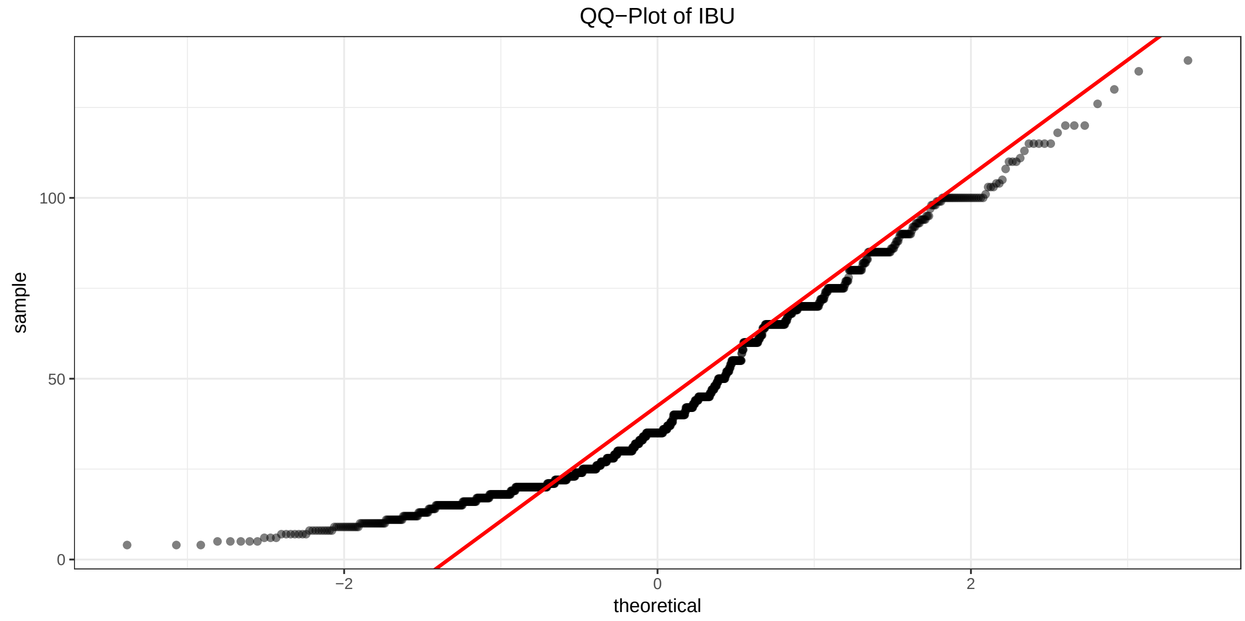
```r
#Scatter plot of ABV v IBU
ggplot(styles, aes(x=abv, y=ibu)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color="red") +
  theme(legend.position="none") +
  ggtitle("ABV v. IBU") +
  theme_minimal()  +
  theme(plot.title = element_text(hjust = 0.5))
```



QQ-Plot - Check for Normality

```
# QQ Plots of IBU and ABV


#calulate line fit
y <- quantile((styles$ibu %>% na.omit()), c(0.25, 0.75))
x <- qnorm(c(0.25, 0.75))
slope <- diff(y)/diff(x)
y_int <- y[1] - slope * x[1]

qq_ibu <- ggplot(styles, aes(sample = styles$ibu)) +
             geom_qq(shape = 16, size = 2, alpha = 0.5) +
              geom_abline(slope = slope, intercept = y_int, colour ='red', size = 1) +
              ggtitle("QQ-Plot of IBU") +
              theme_bw()  +
              theme(plot.title = element_text(hjust = 0.5))


#calulate line fit
y <- quantile((styles$abv %>% na.omit()), c(0.25, 0.75))
x <- qnorm(c(0.25, 0.75))
slope <- diff(y)/diff(x)
y_int <- y[1] - slope * x[1]

qq_abv <- ggplot(styles, aes(sample = styles$abv)) +
            geom_qq(shape = 16, size = 2, alpha = 0.5) +
            geom_abline(slope = slope, intercept = y_int, colour ='red', size = 1) +
            ggtitle("QQ-Plot of ABV") +
            theme_bw() +
            theme(plot.title = element_text(hjust = 0.5))


grid.arrange(qq_abv, qq_ibu)
```

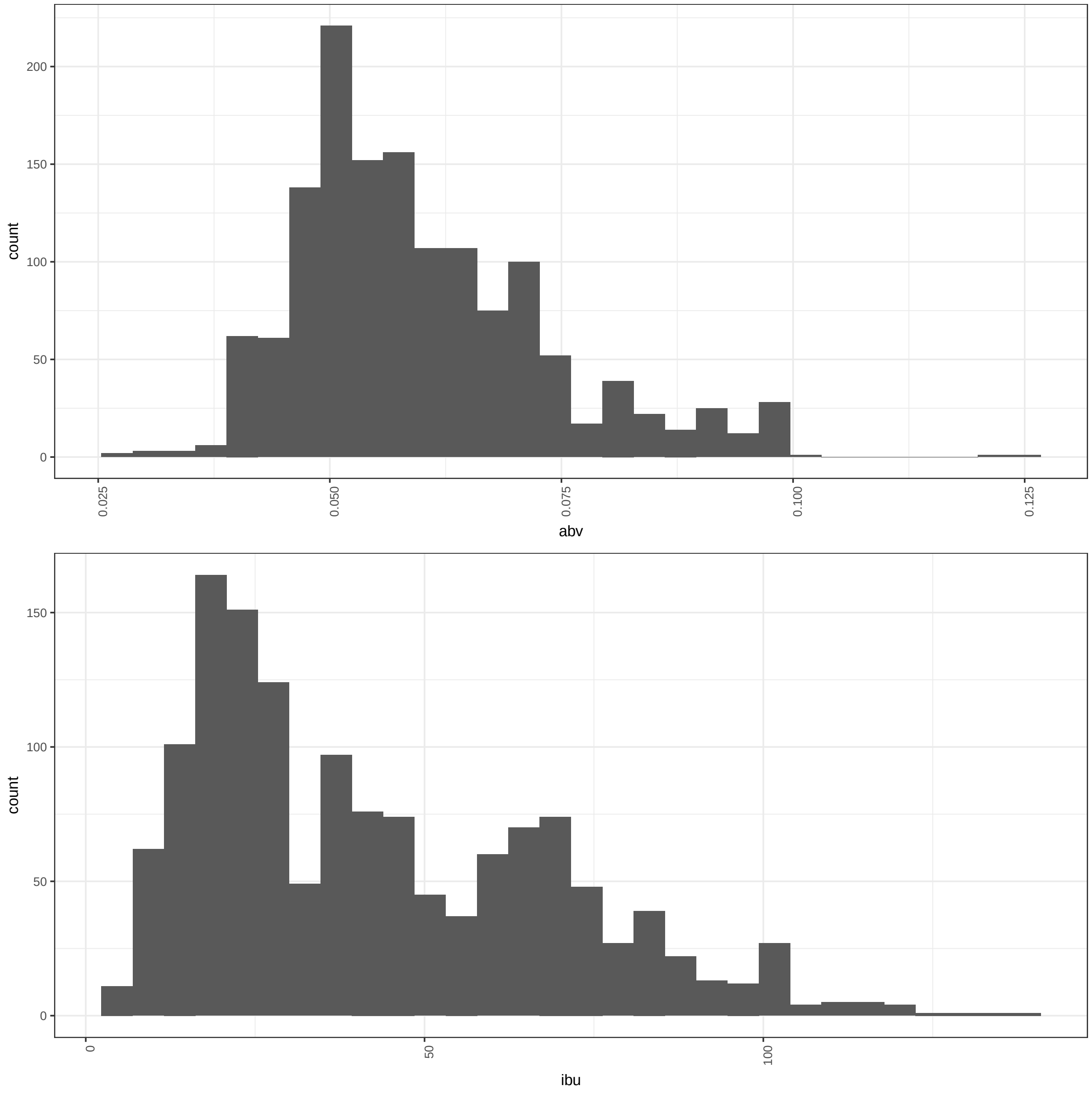




Histogram - Check for Normality

```
# Histograms of IBU and ABV

hist_ibu <- ggplot(styles %>% na.omit(ibu)) +
              geom_histogram(aes(x=ibu)) +
              theme(text = element_text(size=10),
                  axis.text.x = element_text(angle=90, hjust=1))

hist_abv <- ggplot(styles %>% na.omit(abv)) +
              geom_histogram(aes(x=abv)) +
              theme(text = element_text(size=10),
                  axis.text.x = element_text(angle=90, hjust=1))
```

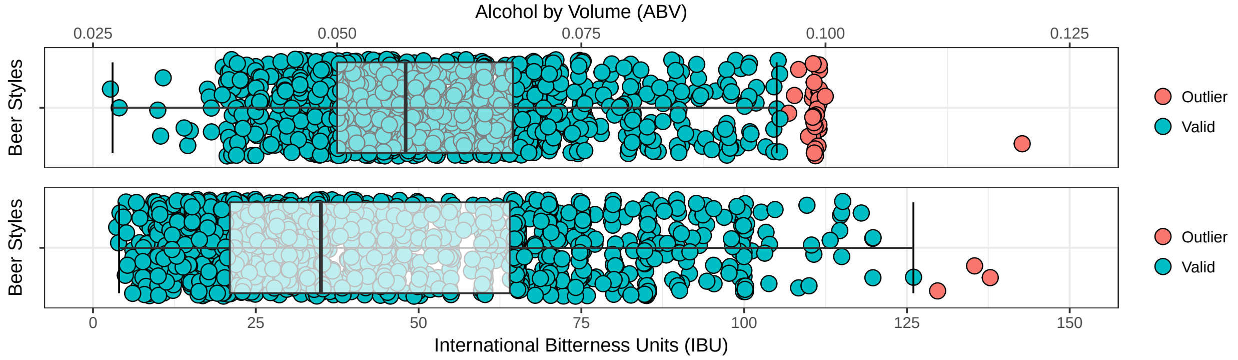
```
grid.arrange(hist_abv, hist_ibu)
```



Boxplot - Check for Outliers

```
# Boxplots of IBU and ABV


ibu_outliers <- boxplot(styles$ibu, plot = FALSE)[["out"]]

abv_outliers <- boxplot(styles$abv, plot = FALSE)[["out"]]
```

```r
x<-boxplot(styles$ibu, plot = FALSE)

bp_abv <- ggplot((styles %>% drop_na(abv)), aes(x="", y=abv)) +
      geom_point(aes(fill = ifelse((abv %in% abv_outliers),"Outlier","Valid")),
                 size = 4,
                 shape = 21,
                 position = position_jitter())+
      stat_boxplot(geom ='errorbar') +
      geom_boxplot(alpha=.5,
                   outlier.shape = NA) +
      guides(fill=guide_legend(title= NULL)) +
      xlab("Beer Styles") +
      ylab("Alcohol by Volume (ABV)") +
      scale_y_continuous(position = "right",
                         breaks = c(.025, .05, .075, .1, .125),
                         limits = c(0.025, .125)) +
      coord_flip()


bp_ibu <- ggplot((styles %>% drop_na(ibu)), aes(x="", y=ibu)) +
      geom_point(aes(fill = ifelse((ibu %in% ibu_outliers),"Outlier","Valid")),
                 size = 4,
                 shape = 21,
                 position = position_jitter())+
      stat_boxplot(geom ='errorbar') +
      geom_boxplot(alpha = .75,
                   outlier.shape = NA) +
      guides(fill=guide_legend(title= NULL)) +
      xlab("Beer Styles") +
      ylab("International Bitterness Units (IBU)") +
      scale_y_continuous(breaks = c(0, 25, 50, 75, 100, 125, 150),
                         limits = c(0, 150)) +
      coord_flip()

grid.arrange(bp_abv, bp_ibu)
```



- Due to the lack of normality of the IBU variable and the presence of outliers in both variables, we will use the Spearman Rank-Correlation test as an alternative to the preferred Pearson Correlation.

Significance Testing: Spearman Rank-Order Correlation

```
+ More info on Spearman test: https://statistics.laerd.com/statistical-guides/spearmans-rank-order-corre
```

- Hypotheses:
  - H$_o$: $\rho = 0$
  - H$_A$: $\rho \neq 0$

```
# Significance test
spear_test_result <- cor.test(styles$ibu, styles$abv, method = "spearman", conf.level = .05, exact=FALS

spear_test_result
```

```
##
##  Spearman's rank correlation rho
##
## data:  styles$ibu and styles$abv
## S = 153570000, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.6677798
```

```
r_sq <- spear_test_result[["estimate"]][["rho"]]^2 # capture r-squared

r_sq
```

```
## [1] 0.4459299
```

Conclusion

There is strong evidence that the ABV and IBU are positively associated (p-value < 0.001 from a Spearman Rank-Order Correlation). At a 95% confidence level, the IBU rating accounts for 44.59% of the variation in the ABV. While IBU and ABV certainly have a correlation, the correlation is weak ($r^2 = 0.45$). Thus, we reject the null hypothesis that IBU rating and ABV are un-corrolated across the beer styles in our sample. Beer styles were not randomly assigned to any treatment and we do not know if the beer data were randomly selected, so we must limit our results to indicating an association between IBU rating an ABV. No causality or inferences to larger populations can be drawn.

Appendex

Session Info

```
sessionInfo()
```

```
## R version 3.4.3 (2017-11-30)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 16299)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] bindrcpp_0.2         gridExtra_2.3        magrittr_1.5
##  [4] summarytools_0.8.0   RColorBrewer_1.1-2   maps_3.2.0
##  [7] ggplot2_2.2.1        knitr_1.18           tidyr_0.7.2
## [10] dplyr_0.7.4          RevoUtilsMath_10.0.1 RevoUtils_10.0.7
## [13] RevoMods_11.0.0      MicrosoftML_9.3.0    mrsdeploy_1.1.3
## [16] RevoScaleR_9.3.0     lattice_0.20-35      rpart_4.1-11
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_0.2.3       purrr_0.2.4            pander_0.6.1
##  [4] colorspace_1.3-2       htmltools_0.3.6        yaml_2.1.16
##  [7] CompatibilityAPI_1.1.0 utf8_1.1.2             rlang_0.1.6
## [10] pillar_1.0.1           glue_1.2.0             pryr_0.1.3
## [13] matrixStats_0.52.2     foreach_1.4.5          bindr_0.1
## [16] plyr_1.8.4             stringr_1.2.0          munsell_0.4.3
## [19] gtable_0.2.0           codetools_0.2-15       evaluate_0.10.1
## [22] labeling_0.3           curl_3.1               highr_0.6
## [25] Rcpp_0.12.14           scales_0.5.0           backports_1.1.2
## [28] jsonlite_1.5           rapportools_1.0        digest_0.6.13
## [31] stringi_1.1.6          grid_3.4.3             rprojroot_1.3-1
## [34] cli_1.0.0              tools_3.4.3            bitops_1.0-6
## [37] lazyeval_0.2.1         RCurl_1.95-4.9         tibble_1.4.1
## [40] crayon_1.3.4           pkgconfig_2.0.1        assertthat_0.2.0
## [43] rmarkdown_1.8          iterators_1.0.9        R6_2.2.2
## [46] compiler_3.4.3
```