Capstone Project: Data Wrangling Report

Sergey Mouzykin

November 2, 2017

### Overview

This raw data will be cleaned and wrangled into a form which then can be analyzed. The following will be performed:

* Column names will be renamed to be short, simple and descriptive
* All columns with characters will be changed to lower case. This includes brewery name, beer name, beer style, and profile name
* Any missing values found will be replaced accordingly
* The amount of beer styles will be reduced from 104 to a more manageable size

### Lower Case

Change all columns with characters to lower case. This includes four columns, brewery name, beer name, profile name, and beer style.

# ---- Lower Case ----  
brewery\_name = tolower(beer\_reviews$brewery\_name)  
beer\_name = tolower(beer\_reviews$beer\_name)  
profile\_name = tolower(beer\_reviews$review\_profilename)  
beer\_style = tolower(beer\_reviews$beer\_style)

### Rename Columns

Some columns need to be renamed in order to be more concise. The changes are summarized in the table below.

|  |  |
| --- | --- |
| Old Name | New Name |
| review\_overall | overall |
| review\_aroma | aroma |
| review\_appearance | appearance |
| review\_palate | palate |
| review\_taste | taste |
| review\_profilename | profile\_name |
| brewery\_name | *no change* |
| brewery\_id | *no change* |
| beer\_style | *no change* |
| beer\_name | *no change* |
| beer\_abv | *no change* |
| beer\_id | *no change* |
| review\_time | *no change* |

# ---- Rename Columns ----  
taste = beer\_reviews$review\_taste  
palate = beer\_reviews$review\_palate  
appearance = beer\_reviews$review\_appearance  
aroma = beer\_reviews$review\_aroma  
overall = beer\_reviews$review\_overall  
brewery\_id = beer\_reviews$brewery\_id  
beer\_id = beer\_reviews$beer\_beerid  
review\_time = beer\_reviews$review\_time  
beer\_abv = beer\_reviews$beer\_abv  
  
# put together the data frame  
beer\_reviews = data\_frame(brewery\_name, beer\_name, profile\_name, beer\_style, taste, palate,   
 appearance, aroma, overall, brewery\_id, beer\_id, review\_time, beer\_abv)

### Finding Missing Values

Approach:

1. Create a function for finding missing values
2. Iterate this function over every column
3. Create a matrix of missing values for easier visualization

find\_NA = function(column\_name){  
 beer\_reviews %>%   
 filter(is.na(column\_name)) %>%  
 summarise(missing\_values = n())  
}  
  
columns\_name = list(  
 beer\_reviews$beer\_id,  
 beer\_reviews$brewery\_name,  
 beer\_reviews$brewery\_id,  
 beer\_reviews$beer\_abv,  
 beer\_reviews$profile\_name,  
 beer\_reviews$taste,  
 beer\_reviews$palate,  
 beer\_reviews$beer\_style,  
 beer\_reviews$appearance,  
 beer\_reviews$aroma,  
 beer\_reviews$overall,  
 beer\_reviews$review\_time,  
 beer\_reviews$beer\_name  
)  
missing\_vals = sapply(columns\_name, find\_NA)  
# matrix of missing values  
columns = c('beer id', 'brewery name', 'brewery id', 'beer ABV', 'profile name', 'taste', 'palate', 'beer style',  
 'appearance', 'aroma', 'overall', 'review time', 'beer name')  
rows = c('missing values')  
missing\_matrix = matrix(missing\_vals, byrow = TRUE, nrow = 1)  
colnames(missing\_matrix) = columns  
rownames(missing\_matrix) = rows  
missing\_matrix

## beer id brewery name brewery id beer ABV profile name taste  
## missing values 0 15 0 67785 348 0   
## palate beer style appearance aroma overall review time  
## missing values 0 0 0 0 0 0   
## beer name  
## missing values 0

Missing values are found in the following columns:

|  |  |
| --- | --- |
| Column | Amount of missing values |
| brewery name | 15 |
| beer ABV | 67,785 |
| profile name | 348 |

The alcohol content is not always written on the container and relatively low ABV is not required to be printed on containers. This may explain the large amount of missing values in the beer\_abv column.

### Missing Values: beer ABV (alcohol by volume)

To deal with these missing values, the mean of the beer\_abv will be computed and used to replace the missing values. After replacing the missing values, the column will be checked again for any missing values in order to confirm the result.

mean\_abv = mean(beer\_reviews$beer\_abv, na.rm = TRUE) # = 7.04  
median\_abv = median(beer\_reviews$beer\_abv, na.rm = TRUE) # 6.6  
  
# replace the missing values in ABV with the mean  
beer\_reviews =   
 beer\_reviews %>%  
 replace\_na(list(beer\_abv = mean\_abv))  
# check for missing values again  
find\_NA(beer\_reviews$beer\_abv)

## # A tibble: 1 x 1  
## missing\_values  
## <int>  
## 1 0

### Missing Values: Brewery Name

At the moment, we can't make any accurate guesses as to what the names are of the breweries. Therefore their missing values will be replaced with the string '*unknown*'. After replacing the missing values, the column will be checked again for any missing values in order to confirm the result.

# view the matrix with missing values  
missing\_matrix

## beer id brewery name brewery id beer ABV profile name taste  
## missing values 0 15 0 67785 348 0   
## palate beer style appearance aroma overall review time  
## missing values 0 0 0 0 0 0   
## beer name  
## missing values 0

# replace the missing names with 'unknown'  
beer\_reviews =   
 beer\_reviews %>%  
 replace\_na(list(brewery\_name = 'unknown'))  
# check for missing values again  
find\_NA(beer\_reviews$brewery\_name)

## # A tibble: 1 x 1  
## missing\_values  
## <int>  
## 1 0

### Missing Values: Profile Name

Profile names will be replaced with the string '*unknown*' since we can't make any accurate guesses of somebody's name in this instance.

# view the matrix with missing values  
missing\_matrix

## beer id brewery name brewery id beer ABV profile name taste  
## missing values 0 15 0 67785 348 0   
## palate beer style appearance aroma overall review time  
## missing values 0 0 0 0 0 0   
## beer name  
## missing values 0

# glance over the missing values  
beer\_reviews %>%  
 select(profile\_name) %>%  
 filter(is.na(profile\_name))

## # A tibble: 348 x 1  
## profile\_name  
## <chr>  
## 1 <NA>  
## 2 <NA>  
## 3 <NA>  
## 4 <NA>  
## 5 <NA>  
## 6 <NA>  
## 7 <NA>  
## 8 <NA>  
## 9 <NA>  
## 10 <NA>  
## # ... with 338 more rows

# replace the missing profile names with 'Unknown'  
beer\_reviews =   
 beer\_reviews %>%  
 replace\_na(list(profile\_name = 'unknown'))  
  
# check for missing values again  
find\_NA(beer\_reviews$profile\_name)

## # A tibble: 1 x 1  
## missing\_values  
## <int>  
## 1 0

### Beer Styles

Currently, there are 104 unique beer styles included in this data set. However, some of these styles are highly specific due to their brewing process, ingredient ratios, yeast type, or other factors; but they can be grouped together as they are, in some cases, a variation of an ale or a lager. The goal is to classify them into more general terms, but without over simplifying, in order to produce plots that are easy to read and translate. A new column will be created to represent these styles. The approach will involve doing some research on the current beer styles included and deciding how to categorize them to yield a smaller list. The *gsub* function will be used to iterate over beer styles and categorize them accordingly.

New Beer-Style column will consist of the following styles:

|  |  |
| --- | --- |
| Beer Styles to be Added |  |
| ale | lager | stout | lambic | spiced |
| pilsner | porter | smoked | barleywine | ipa |
| wheat | bock | bitter | rye | trappist |

# create a list of beer styles to use within 'gsub'  
style\_list = c(beer\_reviews %>% select(beer\_style))  
  
style\_list = unlist(style\_list)  
ale = gsub(pattern = '.\*ale.\*|^alt.\*|.\*winter.\*|.\*garde$|^k.\*lsch.\*', replacement = 'ale', x = style\_list)  
lager = gsub(pattern = '.\*lager.\*|^schwarz.\*|^m.\*rzen.\*|.\*steam.\*|.\*zwickel.\*', replacement = 'lager', x = ale)  
stout = gsub(pattern = '.\*stout.\*', replacement = 'stout', x = lager)  
pils = gsub(pattern = '.\*pils.\*', replacement = 'pilsner', x = stout)  
porter = gsub(pattern = '.\*porter.\*', replacement = 'porter', x = pils)  
wheat = gsub(pattern = '.\*weizen.\*|.\*wit.\*|.\*weiss.\*|.\*gose.\*', replacement = 'wheat', x = porter)  
bock = gsub(pattern = '.\*bock.\*', replacement = 'bock', x = wheat)  
lambic = gsub(pattern = '^lambic.\*|.\*faro.\*|.\*gueuze.\*', replacement = 'lambic', x = bock)  
smoked = gsub(pattern = '^smoked.\*|^rauch.\*', replacement = 'smoked', x = lambic)  
barleywine = gsub(pattern = '.\*barleywine.\*', replacement = 'barleywine', x = smoked)  
bitter = gsub(pattern = '.\*bitter.\*', replacement = 'bitter', x = barleywine)  
rye = gsub(pattern = '.\*rye.\*|.\*roggen.\*|kvass', replacement = 'rye', x = bitter)  
spiced = gsub(pattern = '.\*herbed.\*|.\*braggot.\*|^chile.\*', replacement = 'spiced/herbed', x = rye)  
trappist = gsub(pattern = '^quad.\*|^dub.\*|^tri.\*', replacement = 'trappist', x = spiced)  
ipa = gsub(pattern = '.\*ipa.\*', replacement = 'ipa', x = trappist)  
  
style\_list\_mod = ipa  
unique(style\_list\_mod) # print out the list of distinct beer styles

## [1] "wheat" "ale"   
## [3] "stout" "pilsner"   
## [5] "ipa" "spiced/herbed"   
## [7] "lager" "smoked"   
## [9] "porter" "fruit / vegetable beer"   
## [11] "bitter" "barleywine"   
## [13] "bock" "low alcohol beer"   
## [15] "rye" "trappist"   
## [17] "flanders oud bruin" "black & tan"   
## [19] "wheatwine" "american malt liquor"   
## [21] "lambic" "happoshu"   
## [23] "sahti" "bière de champagne / bière brut"

length(unique(style\_list\_mod)) # number of distinct beer styles: 24

## [1] 24

# create new column with the new styles using mutate()  
beer\_reviews =   
 beer\_reviews %>%  
 mutate(general\_beer\_style = style\_list\_mod)  
  
glimpse(beer\_reviews)

## Observations: 1,586,614  
## Variables: 14  
## $ brewery\_name <chr> "vecchio birraio", "vecchio birraio", "vecc...  
## $ beer\_name <chr> "sausa weizen", "red moon", "black horse bl...  
## $ profile\_name <chr> "stcules", "stcules", "stcules", "stcules",...  
## $ beer\_style <chr> "hefeweizen", "english strong ale", "foreig...  
## $ taste <dbl> 1.5, 3.0, 3.0, 3.0, 4.5, 3.5, 4.0, 3.5, 4.0...  
## $ palate <dbl> 1.5, 3.0, 3.0, 2.5, 4.0, 3.0, 4.0, 2.0, 3.5...  
## $ appearance <dbl> 2.5, 3.0, 3.0, 3.5, 4.0, 3.5, 3.5, 3.5, 3.5...  
## $ aroma <dbl> 2.0, 2.5, 2.5, 3.0, 4.5, 3.5, 3.5, 2.5, 3.0...  
## $ overall <dbl> 1.5, 3.0, 3.0, 3.0, 4.0, 3.0, 3.5, 3.0, 4.0...  
## $ brewery\_id <int> 10325, 10325, 10325, 10325, 1075, 1075, 107...  
## $ beer\_id <int> 47986, 48213, 48215, 47969, 64883, 52159, 5...  
## $ review\_time <int> 1234817823, 1235915097, 1235916604, 1234725...  
## $ beer\_abv <dbl> 5.0, 6.2, 6.5, 5.0, 7.7, 4.7, 4.7, 4.7, 4.7...  
## $ general\_beer\_style <chr> "wheat", "ale", "stout", "pilsner", "ipa", ...

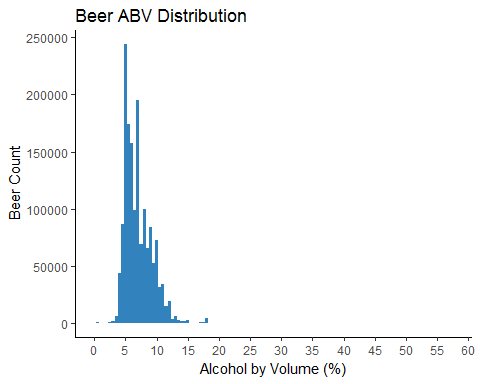
# summarize the new beer styles by calling distinct()  
beer\_reviews %>%  
 group\_by(general\_beer\_style) %>%  
 summarise(style\_count = n()) %>%  
 arrange(desc(style\_count))

## # A tibble: 24 x 2  
## general\_beer\_style style\_count  
## <chr> <int>  
## 1 ale 577361  
## 2 ipa 216034  
## 3 stout 182268  
## 4 lager 132481  
## 5 wheat 80947  
## 6 porter 73249  
## 7 trappist 68397  
## 8 bock 46501  
## 9 barleywine 40459  
## 10 pilsner 40330  
## # ... with 14 more rows

### Plot: Beer ABV

Plotting the beer's ABV as a histogram will reveal it's distribution and help us spot any outliers that may be present before diving into deeper analysis.

#glimpse(beer\_reviews)  
blues = brewer.pal(6, 'Blues')  
ggplot(beer\_reviews, aes(x = beer\_abv)) +   
 geom\_histogram(binwidth = 0.5, position = 'identity', fill = '#3182BD') +   
 scale\_x\_continuous('Alcohol by Volume (%)', breaks = seq(0, 60, by = 5)) +  
 scale\_y\_continuous('Beer Count') +  
 ggtitle('Beer ABV Distribution') +  
 theme\_classic()



This plot shows us that the ABV is actually a right-skewed distribution due to some beers having a a very high alcohol by volume content. The minimum and maximum of ABV content are 0.01% and 57.7%, respectively. The mean lies at 7.04, the median at 6.6, and the standard deviation is 2.27. We can also infer that the distribution is right-skewed due to the median being lower than the mean. Although the values range from 0.01 to 57.7, in theory, about 95% of these values should lie within two standard deviations from the mean.

# find max and min for beer ABV  
# replace the missing values before finding MAX and MIN  
max(beer\_reviews$beer\_abv) # 57.7 %

## [1] 57.7

min(beer\_reviews$beer\_abv) # 0.01 %

## [1] 0.01

### Interval: Beer ABV

In order to better visualize the alcohol level content, the ABV can be distributed into five factored levels using the calculated mean and standard deviation. These five levels will be labeled as, 'low', 'below normal', 'normal', 'above normal', and 'high'. The computed standard deviation will be used to create the breaks for the labels.

abv\_vector = beer\_reviews$beer\_abv  
mean\_abv = mean(abv\_vector) # 7.04  
median\_abv = median(abv\_vector) # 6.6  
sd(abv\_vector) # 2.27

## [1] 2.272372

var(abv\_vector) # 5.16

## [1] 5.163673

length(abv\_vector)

## [1] 1586614

# calculate the deviation breaks  
one\_sd\_below = mean(abv\_vector) - sd(abv\_vector) # 4.77 -> 1 SD below mean  
one\_sd\_above = mean(abv\_vector) + sd(abv\_vector) # 9.31 -> 1 SD above mean  
two\_sd\_below = mean(abv\_vector) - 2\*sd(abv\_vector) # 2.50 -> 2 SD below mean  
two\_sd\_above = mean(abv\_vector) + 2\*sd(abv\_vector) # 11.59 -> 2 SD above mean  
  
# Interval of ABV  
# use cut() and a vector for the breaks  
# normal ABV level will be considered being within 1 SD of the mean  
breaks\_vect = c(0, two\_sd\_below, one\_sd\_below, one\_sd\_above, two\_sd\_above, 60)  
interval\_abv = cut(abv\_vector,   
 breaks = breaks\_vect,   
 labels = c('low', 'below normal', 'normal', 'above normal', 'high'))  
table(interval\_abv)

## interval\_abv  
## low below normal normal above normal high   
## 1630 140735 1193157 200598 50494

length(interval\_abv)

## [1] 1586614

# sum the amounts within two standard deviations and divide by total amount of rows  
# this will contain about 95% of all the data points in the beer's abv column  
(140735 + 1193157 + 200598) / length(abv\_vector) \* 100

## [1] 96.71476

# NEW COLUMN: beer\_abv\_factor  
beer\_reviews =   
 beer\_reviews %>%  
 mutate(beer\_abv\_factor = interval\_abv)  
  
glimpse(beer\_reviews)

## Observations: 1,586,614  
## Variables: 15  
## $ brewery\_name <chr> "vecchio birraio", "vecchio birraio", "vecc...  
## $ beer\_name <chr> "sausa weizen", "red moon", "black horse bl...  
## $ profile\_name <chr> "stcules", "stcules", "stcules", "stcules",...  
## $ beer\_style <chr> "hefeweizen", "english strong ale", "foreig...  
## $ taste <dbl> 1.5, 3.0, 3.0, 3.0, 4.5, 3.5, 4.0, 3.5, 4.0...  
## $ palate <dbl> 1.5, 3.0, 3.0, 2.5, 4.0, 3.0, 4.0, 2.0, 3.5...  
## $ appearance <dbl> 2.5, 3.0, 3.0, 3.5, 4.0, 3.5, 3.5, 3.5, 3.5...  
## $ aroma <dbl> 2.0, 2.5, 2.5, 3.0, 4.5, 3.5, 3.5, 2.5, 3.0...  
## $ overall <dbl> 1.5, 3.0, 3.0, 3.0, 4.0, 3.0, 3.5, 3.0, 4.0...  
## $ brewery\_id <int> 10325, 10325, 10325, 10325, 1075, 1075, 107...  
## $ beer\_id <int> 47986, 48213, 48215, 47969, 64883, 52159, 5...  
## $ review\_time <int> 1234817823, 1235915097, 1235916604, 1234725...  
## $ beer\_abv <dbl> 5.0, 6.2, 6.5, 5.0, 7.7, 4.7, 4.7, 4.7, 4.7...  
## $ general\_beer\_style <chr> "wheat", "ale", "stout", "pilsner", "ipa", ...  
## $ beer\_abv\_factor <fctr> normal, normal, normal, normal, normal, be...

### Write the cleaned data to a new file

The clean file is now ready to be written.

beer\_reviews\_clean = beer\_reviews  
glimpse(beer\_reviews\_clean)

## Observations: 1,586,614  
## Variables: 15  
## $ brewery\_name <chr> "vecchio birraio", "vecchio birraio", "vecc...  
## $ beer\_name <chr> "sausa weizen", "red moon", "black horse bl...  
## $ profile\_name <chr> "stcules", "stcules", "stcules", "stcules",...  
## $ beer\_style <chr> "hefeweizen", "english strong ale", "foreig...  
## $ taste <dbl> 1.5, 3.0, 3.0, 3.0, 4.5, 3.5, 4.0, 3.5, 4.0...  
## $ palate <dbl> 1.5, 3.0, 3.0, 2.5, 4.0, 3.0, 4.0, 2.0, 3.5...  
## $ appearance <dbl> 2.5, 3.0, 3.0, 3.5, 4.0, 3.5, 3.5, 3.5, 3.5...  
## $ aroma <dbl> 2.0, 2.5, 2.5, 3.0, 4.5, 3.5, 3.5, 2.5, 3.0...  
## $ overall <dbl> 1.5, 3.0, 3.0, 3.0, 4.0, 3.0, 3.5, 3.0, 4.0...  
## $ brewery\_id <int> 10325, 10325, 10325, 10325, 1075, 1075, 107...  
## $ beer\_id <int> 47986, 48213, 48215, 47969, 64883, 52159, 5...  
## $ review\_time <int> 1234817823, 1235915097, 1235916604, 1234725...  
## $ beer\_abv <dbl> 5.0, 6.2, 6.5, 5.0, 7.7, 4.7, 4.7, 4.7, 4.7...  
## $ general\_beer\_style <chr> "wheat", "ale", "stout", "pilsner", "ipa", ...  
## $ beer\_abv\_factor <fctr> normal, normal, normal, normal, normal, be...

# write the clean file   
write\_csv(beer\_reviews\_clean, 'beer\_reviews\_clean.csv')