Case Study\_01\_BeerDataAnalysis

February 19, 2018

Load Libraries

rm(list = ls())  
library(ggplot2)  
library(readr)  
library(repmis)  
library(RCurl)

## Loading required package: bitops

library(bitops)  
library(tidyverse)

## -- Attaching packages --------------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v tibble 1.4.2 v dplyr 0.7.4  
## v tidyr 0.7.2 v stringr 1.2.0  
## v purrr 0.2.4 v forcats 0.2.0

## -- Conflicts ------------------------------------------------------------------------------------------ tidyverse\_conflicts() --  
## x tidyr::complete() masks RCurl::complete()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(plyr)

## -------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following object is masked from 'package:purrr':  
##   
## compact

## Environment Information

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] plyr\_1.8.4 forcats\_0.2.0 stringr\_1.2.0 dplyr\_0.7.4   
## [5] purrr\_0.2.4 tidyr\_0.7.2 tibble\_1.4.2 tidyverse\_1.2.1  
## [9] RCurl\_1.95-4.8 bitops\_1.0-6 repmis\_0.5 readr\_1.1.1   
## [13] ggplot2\_2.2.1   
##   
## loaded via a namespace (and not attached):  
## [1] reshape2\_1.4.3 haven\_1.1.1 lattice\_0.20-35   
## [4] colorspace\_1.3-2 htmltools\_0.3.6 yaml\_2.1.16   
## [7] rlang\_0.1.6 R.oo\_1.21.0 pillar\_1.1.0   
## [10] foreign\_0.8-69 glue\_1.2.0 R.utils\_2.6.0   
## [13] modelr\_0.1.1 readxl\_1.0.0 bindrcpp\_0.2   
## [16] R.cache\_0.12.0 bindr\_0.1 munsell\_0.4.3   
## [19] gtable\_0.2.0 cellranger\_1.1.0 rvest\_0.3.2   
## [22] R.methodsS3\_1.7.1 psych\_1.7.8 evaluate\_0.10.1   
## [25] knitr\_1.17 parallel\_3.4.3 broom\_0.4.3   
## [28] Rcpp\_0.12.14 scales\_0.5.0 backports\_1.1.2   
## [31] formatR\_1.5 jsonlite\_1.5 mnormt\_1.5-5   
## [34] hms\_0.4.0 digest\_0.6.13 stringi\_1.1.6   
## [37] grid\_3.4.3 rprojroot\_1.3-1 cli\_1.0.0   
## [40] tools\_3.4.3 magrittr\_1.5 lazyeval\_0.2.1   
## [43] crayon\_1.3.4 pkgconfig\_2.0.1 xml2\_1.2.0   
## [46] data.table\_1.10.4-3 lubridate\_1.7.2 rstudioapi\_0.7   
## [49] assertthat\_0.2.0 rmarkdown\_1.8 httr\_1.3.1   
## [52] R6\_2.2.2 nlme\_3.1-131.1 compiler\_3.4.3

## Brewery Data Analysis

[Link to the Github Repository Associated with this Study](https://github.com/davxdan/MSDS_6306_DoingDataScience_Case-Study_01)

The purpose of this is to present findings from blah blah …. The questions asked are listed below with data analysis methods and answers.

### The Data Provided

Load Beers.csv

RawBeerData <- read\_csv("Input/RawDataFiles/Beers.csv")

## Parsed with column specification:  
## cols(  
## Name = col\_character(),  
## Beer\_ID = col\_integer(),  
## ABV = col\_double(),  
## IBU = col\_integer(),  
## Brewery\_id = col\_integer(),  
## Style = col\_character(),  
## Ounces = col\_double()  
## )

Load Breweries.csv

RawBreweryData <- read\_csv("Input/RawDataFiles/Breweries.csv")

## Parsed with column specification:  
## cols(  
## Brew\_ID = col\_integer(),  
## Name = col\_character(),  
## City = col\_character(),  
## State = col\_character()  
## )

### 1. How many breweries are present in each state?

The record (110,“Woodstock Inn, Station & Brewery”,North Woodstock, NH) was causing errors but readr fixed this. Identify the records:

RawBreweryData[c(110, 111, 112), ] #Identified erroneous records

## # A tibble: 3 x 4  
## Brew\_ID Name City State  
## <int> <chr> <chr> <chr>  
## 1 110 Woodstock Inn, Station & Brewery North Woodstock NH   
## 2 111 Renegade Brewing Company Denver CO   
## 3 112 Mother Earth Brew Company Vista CA

Stage1BreweryData <- RawBreweryData  
Stage1BreweryData <- transform(Stage1BreweryData, State = as.character(State))

CountBreweriesByState <- data.frame(Stage1BreweryData$State)  
summary(CountBreweriesByState, maxsum = 100)

## Stage1BreweryData.State  
## AK: 7   
## AL: 3   
## AR: 2   
## AZ:11   
## CA:39   
## CO:47   
## CT: 8   
## DC: 1   
## DE: 2   
## FL:15   
## GA: 7   
## HI: 4   
## IA: 5   
## ID: 5   
## IL:18   
## IN:22   
## KS: 3   
## KY: 4   
## LA: 5   
## MA:23   
## MD: 7   
## ME: 9   
## MI:32   
## MN:12   
## MO: 9   
## MS: 2   
## MT: 9   
## NC:19   
## ND: 1   
## NE: 5   
## NH: 3   
## NJ: 3   
## NM: 4   
## NV: 2   
## NY:16   
## OH:15   
## OK: 6   
## OR:29   
## PA:25   
## RI: 5   
## SC: 4   
## SD: 1   
## TN: 3   
## TX:28   
## UT: 4   
## VA:16   
## VT:10   
## WA:23   
## WI:20   
## WV: 1   
## WY: 4

### 2. Merge beer data with the breweries data. Print the ﬁrst 6 observations and the last six observations to check the merged ﬁle.

colnames(Stage1BreweryData) <- c("Brewery\_id", "BreweryName", "City", "State")  
Stage2 <- merge(x = RawBeerData, y = Stage1BreweryData, by = c("Brewery\_id"),   
 all = FALSE)  
head(Stage2)

## Brewery\_id Name Beer\_ID ABV IBU  
## 1 1 Get Together 2692 0.045 50  
## 2 1 Maggie's Leap 2691 0.049 26  
## 3 1 Wall's End 2690 0.048 19  
## 4 1 Pumpion 2689 0.060 38  
## 5 1 Stronghold 2688 0.060 25  
## 6 1 Parapet ESB 2687 0.056 47  
## Style Ounces BreweryName City  
## 1 American IPA 16 NorthGate Brewing Minneapolis  
## 2 Milk / Sweet Stout 16 NorthGate Brewing Minneapolis  
## 3 English Brown Ale 16 NorthGate Brewing Minneapolis  
## 4 Pumpkin Ale 16 NorthGate Brewing Minneapolis  
## 5 American Porter 16 NorthGate Brewing Minneapolis  
## 6 Extra Special / Strong Bitter (ESB) 16 NorthGate Brewing Minneapolis  
## State  
## 1 MN  
## 2 MN  
## 3 MN  
## 4 MN  
## 5 MN  
## 6 MN

tail(Stage2)

## Brewery\_id Name Beer\_ID ABV IBU  
## 2405 556 Pilsner Ukiah 98 0.055 NA  
## 2406 557 Heinnieweisse Weissebier 52 0.049 NA  
## 2407 557 Snapperhead IPA 51 0.068 NA  
## 2408 557 Moo Thunder Stout 50 0.049 NA  
## 2409 557 Porkslap Pale Ale 49 0.043 NA  
## 2410 558 Urban Wilderness Pale Ale 30 0.049 NA  
## Style Ounces BreweryName  
## 2405 German Pilsener 12 Ukiah Brewing Company  
## 2406 Hefeweizen 12 Butternuts Beer and Ale  
## 2407 American IPA 12 Butternuts Beer and Ale  
## 2408 Milk / Sweet Stout 12 Butternuts Beer and Ale  
## 2409 American Pale Ale (APA) 12 Butternuts Beer and Ale  
## 2410 English Pale Ale 12 Sleeping Lady Brewing Company  
## City State  
## 2405 Ukiah CA  
## 2406 Garrattsville NY  
## 2407 Garrattsville NY  
## 2408 Garrattsville NY  
## 2409 Garrattsville NY  
## 2410 Anchorage AK

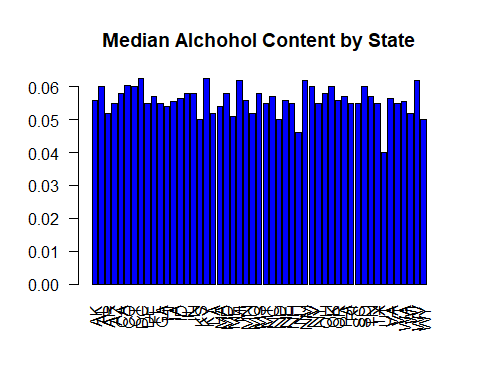
### 3. Report the number of NA’s in each column.

Brewery\_id <- sum(is.na(Stage2$Brewery\_id))  
Name <- sum(is.na(Stage2$Name))  
Beer\_ID <- sum(is.na(Stage2$Beer\_ID))  
ABV <- sum(is.na(Stage2$ABV))  
IBU <- sum(is.na(Stage2$IBU))  
Style <- sum(is.na(Stage2$Style))  
Ounces <- sum(is.na(Stage2$Ounces))  
BreweryName <- sum(is.na(Stage2$BreweryName))  
City <- sum(is.na(Stage2$City))  
State <- sum(is.na(Stage2$State))  
NASummary <- as.matrix(c(Brewery\_id, Name, Beer\_ID, ABV, IBU, Style, Ounces,   
 BreweryName, City, State))  
colnames(NASummary) <- c("Count of NA's")  
rownames(NASummary) <- c("Brewery\_id", "Name", "Beer\_ID", "ABV", "IBU", "Style",   
 "Ounces", "BreweryName", "City", "State")  
NASummary

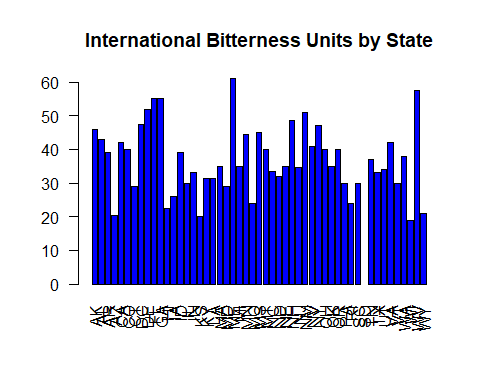
## Count of NA's  
## Brewery\_id 0  
## Name 0  
## Beer\_ID 0  
## ABV 62  
## IBU 1005  
## Style 5  
## Ounces 0  
## BreweryName 0  
## City 0  
## State 0

### 4. Compute the median alcohol content and international bitterness unit for each state. Plot a bar chart to compare.

getMedians <- function(x) {  
 c(median = median(x, na.rm = TRUE))  
}  
ABVMedians <- as.data.frame(tapply(Stage2$ABV, Stage2$State, getMedians))  
  
par(las = 2)  
barplot(ABVMedians[, 1], main = "Median Alchohol Content by State", horiz = FALSE,   
 col = 4)



IBUMedians <- as.data.frame(tapply(Stage2$IBU, Stage2$State, getMedians))  
  
par(las = 2)  
barplot(IBUMedians[, 1], main = "International Bitterness Units by State", horiz = FALSE,   
 col = 4)



### 5. Which state has the maximum alcoholic (ABV) beer? Which state has the most bitter (IBU) beer?

MaxABVState <- ddply(Stage2, .(State), summarise, MaxABVState = max(ABV, na.rm = TRUE))  
MaxABVState <- MaxABVState[order(MaxABVState$MaxABVState), ]  
head(MaxABVState, 1)

## State MaxABVState  
## 9 DE 0.055

summary(Stage2$ABV)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00100 0.05000 0.05600 0.05977 0.06700 0.12800 62

MaxIBUState <- ddply(Stage2, .(State), summarise, MaxIBUState = max(as.double(Stage2$IBU),   
 na.rm = TRUE))  
MaxIBUState <- MaxIBUState[order(MaxIBUState$MaxIBUState), ]  
head(MaxIBUState, 1)

## State MaxIBUState  
## 1 AK 138

### 6. Summary statistics for the ABV variable.

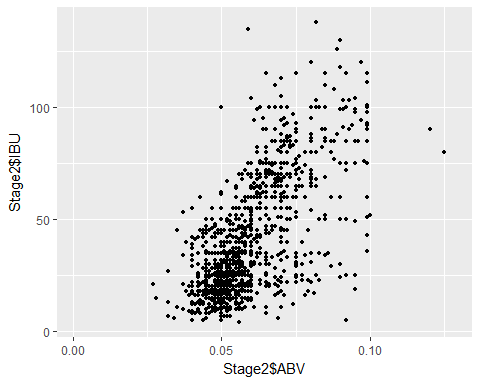
summary(Stage2$ABV)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00100 0.05000 0.05600 0.05977 0.06700 0.12800 62

### 7. Is there an apparent relationship between the bitterness of the beer and its alcoholic content? Draw a scatter plot. You are welcome to use the ggplot2 library for graphs. Please ignore missing values in your analysis. Make your best judgment of a relationship and EXPLAIN your answer.

p <- ggplot(Stage2, aes(Stage2$ABV, Stage2$IBU))  
p + geom\_point(size = 1)

## Warning: Removed 1005 rows containing missing values (geom\_point).



Formatting Samples

# Header 1

## Header 2

### THis is a header

#### Header 4

##### Header 5

###### Header 6

–

…



block quote

* unordered list
* item 2

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.