DS6306: Cast Study 01

Ben Goodwin/ Justin Ehly

10/6/2020

```r
# Introduction #########
#
#
# The following is an analysis of one-hundred styles of beer brewed in the United States for the execut
# CEO and CFO at Budweiser. Budweiser is interested in exploring the how many breweries are in the Unit
# how each beer is reported in terms of its International Bitterness Unit and Alcohol By Content and ba
# statistics and conclusions we are able to uncover with the beer data provided. Statistics will includ
# missing data and explaining why it was possibly not included in the initial dataset, as well as uncov
# and maximum (IBU and ABV) ratings by state. Conclusions will include basic summary statics on the ABV
# any relationship between the IBU and ABV variables (such as dependencies, e.g. does a higher IBU resu
# higher ABV) and finally we will look to see if we can determine general beer styles (Ales and IPAs) b
# ABV and IBU values. Additionally, we will report on any findings that are discovered during the analy
#
######################
#                    #
#     Libraries      #
#                    #
######################
######################
library(usmap)
library(ggplot2)
library(magrittr)
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
library(readr)
library(tibble)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v tidyr   1.1.2     v stringr 1.4.0
## v purrr   0.3.4     v forcats 0.5.0
## v dplyr   1.0.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::extract()   masks magrittr::extract()
## x dplyr::filter()    masks stats::filter()
## x dplyr::lag()       masks stats::lag()
## x purrr::set_names() masks magrittr::set_names()
```

```r
library(robustbase)
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
```

```r
library(class)
library(caret)
```

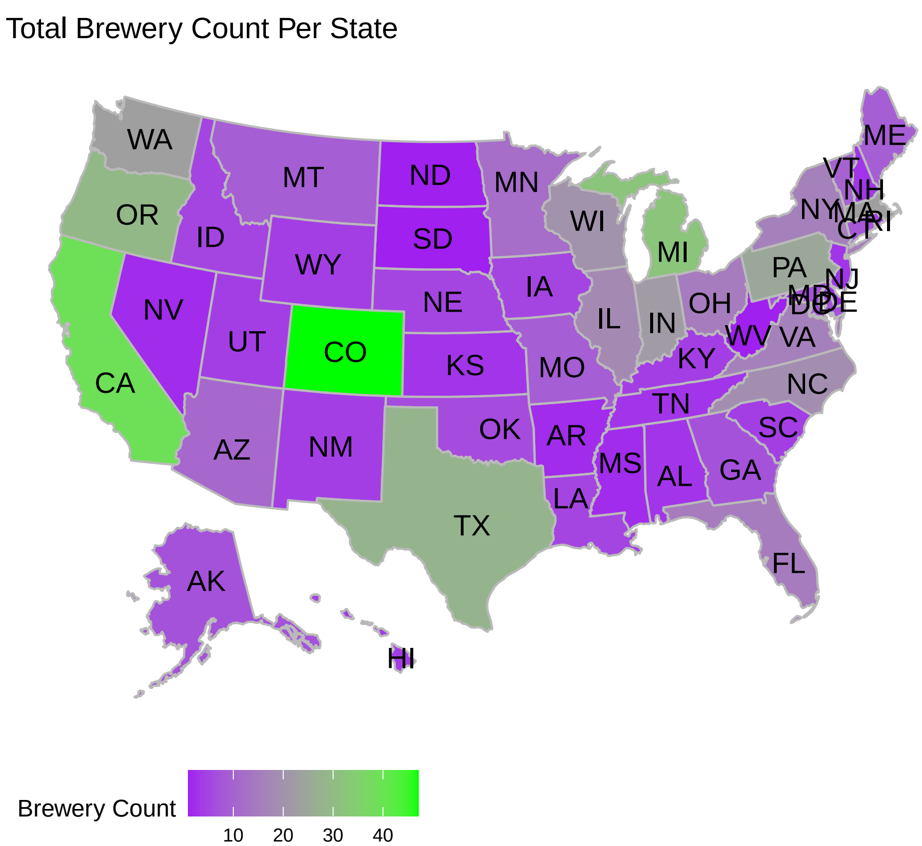
```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(e1071)
library(dplyr)
library(RColorBrewer)
######################
######################
#                    #
#        Data        #
#                    #
######################
#####################################################
#read in brewery data
setwd("C:/Users/justi/Documents/GitHub/MSDS6306/CaseStudy1/project_files/")
breweryDat <- read.csv("breweries.csv")
breweryDat$State <- trimws(breweryDat$State)

#datafile to organize states into census regions
regionData <- read.csv("state-geocodes-v2017.csv")
regionData <- regionData[,c(-1,-2)]
regionData <- dplyr::rename(regionData, "FIPS"="State..FIPS.", "Region" = "Region.1", "Division" = "Div
regionData$State <- trimws(regionData$State)
#Ensure structure of data is compliant
#head(breweryDat)
#read in beer data
```

```r
beerDat <- read.csv("beers.csv")
#Loop to fix leading decimal places on ABV
i <- 1
count <- length(beerDat$Name)
for (i in 1:count) {
 if(is.na(beerDat[i,3])){
  beerDat[i,3]=0
   }
   if(beerDat[i,3]<1){
    beerDat[i,3] <- beerDat[i,3]*100
  }
}


#Ensure structure of data is compliant
#head(beerDat)
#########################################################
```

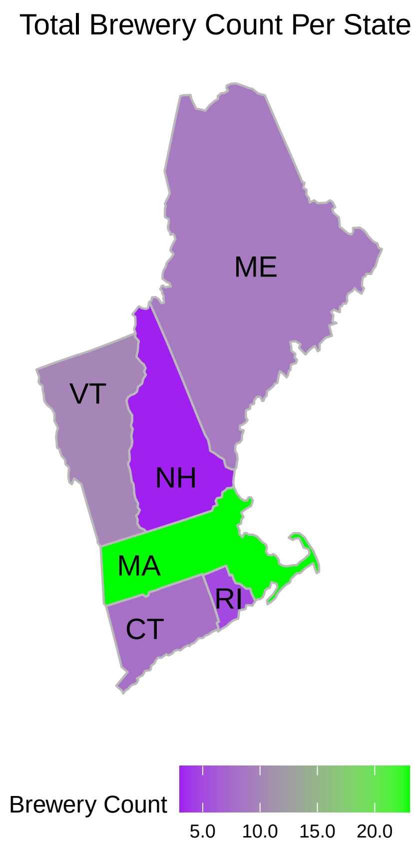
```r
# Question 1 - How many breweries are in each state?
#
# During this analysis, we explored how many breweries are in each state and grouped the states
# by US Census Divisions. The data is visually displayed using maps of each USC Division below
# and summarized in a simple chart at the end.
#
######################
#                    #
#     Question 1     #
#                    #
######################
#
##########################################################
#Use Dplyr to group breweries by state
brewByState <- breweryDat %>%
  group_by(State) %>%
  dplyr::count()
##########################################################
##########################################################
#Add breweries by state to state information dataframe
statepop$brewByState <- brewByState$n
##########################################################
##########################################################
#Fix mismatched state brewery count to state info df
statepop[1,5] <- 3
statepop[2,5] <- 7
statepop[3,5] <- 11
statepop[4,5] <- 2
statepop[8,5] <- 2
statepop[9,5] <- 1
statepop[14,5] <- 18
statepop[15,5] <- 22
statepop[16,5] <- 5
statepop[20,5] <- 9
statepop[22,5] <- 23
statepop[25,5] <- 2
```

```
statepop[26,5] <- 9
statepop[28,5] <- 5
statepop[29,5] <- 2
statepop[30,5] <- 3
statepop[32,5] <- 4
statepop[34,5] <- 19
statepop[33,5] <- 16
statepop[35,5] <- 1
statepop[45,5] <- 4
statepop[46,5] <- 10
statepop[47,5] <- 16
statepop[49,5] <- 1
statepop[50,5] <- 20
#Check data
#View(statepop)
#View(brewByState)
############################################################
############################################################
#Call plot functions to plot state brewery count on USmap
nationBrewPlot <- plot_usmap(data = statepop, values = "brewByState",labels=TRUE, color = "grey73") + s
#display plot
nationBrewPlot
```

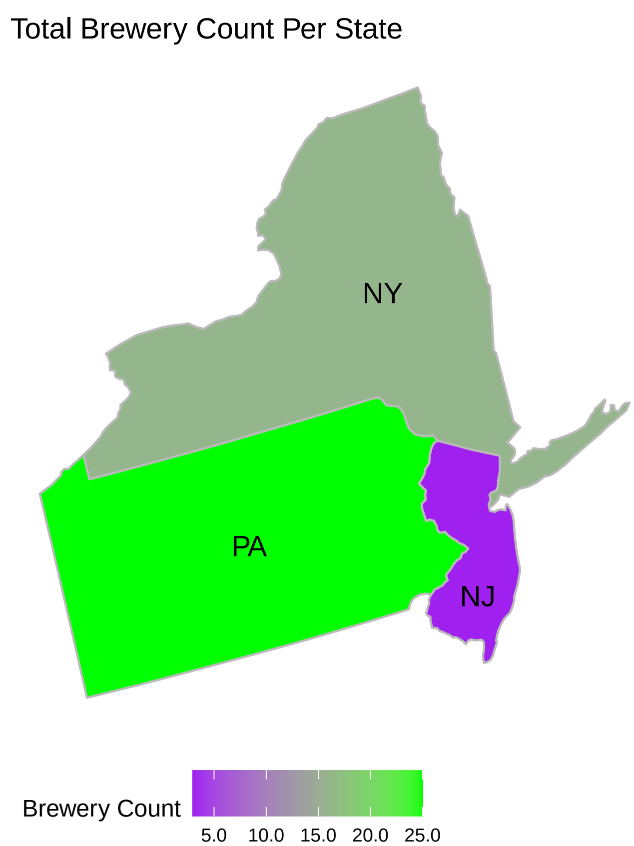



```
############################################################
############################################################
#Break down by region, NE first
```

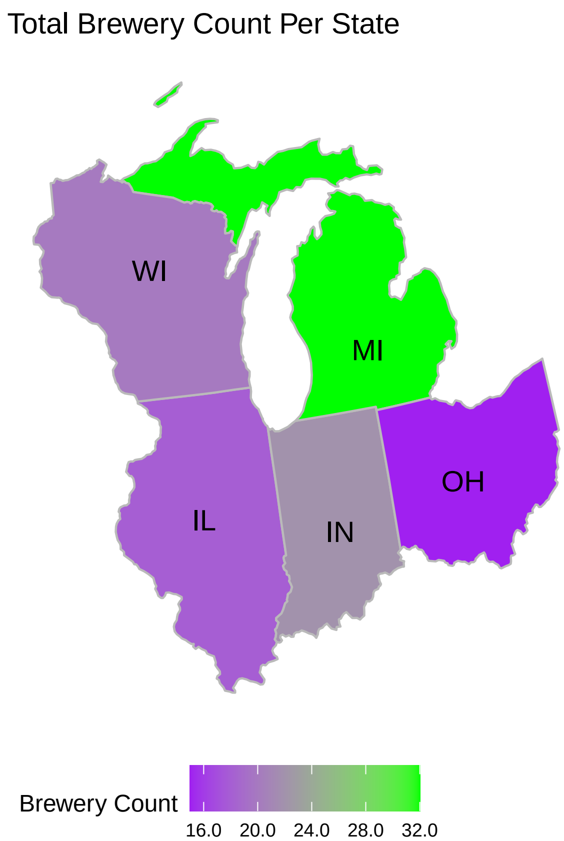
```
NEplot <- plot_usmap(data=statepop, values = "brewByState",labels = TRUE,include = .new_england,color =
NEplot
```



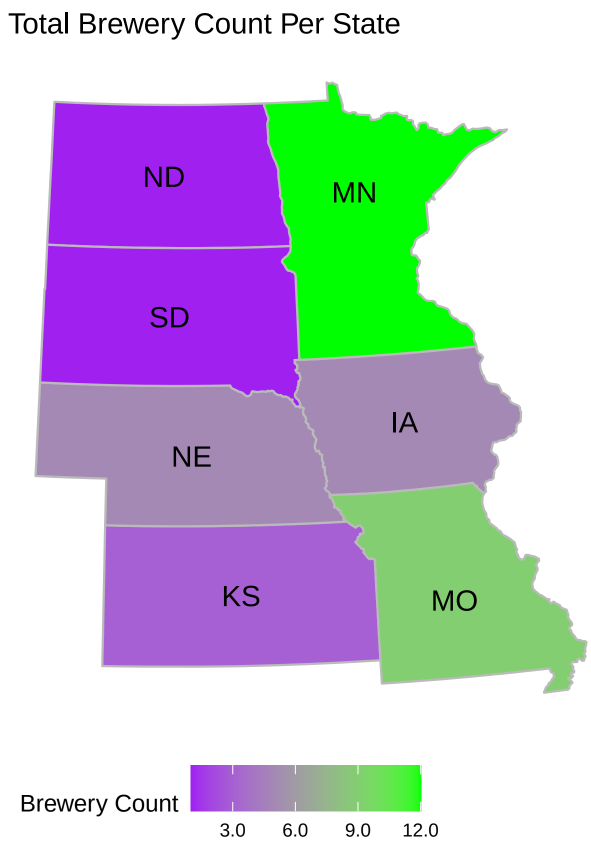

```
##########################################################
##########################################################
#Break down by region, Mid Atlantic second
MAplot <- plot_usmap(data=statepop, values = "brewByState",labels = TRUE,include = .mid_atlantic,color =
MAplot
```



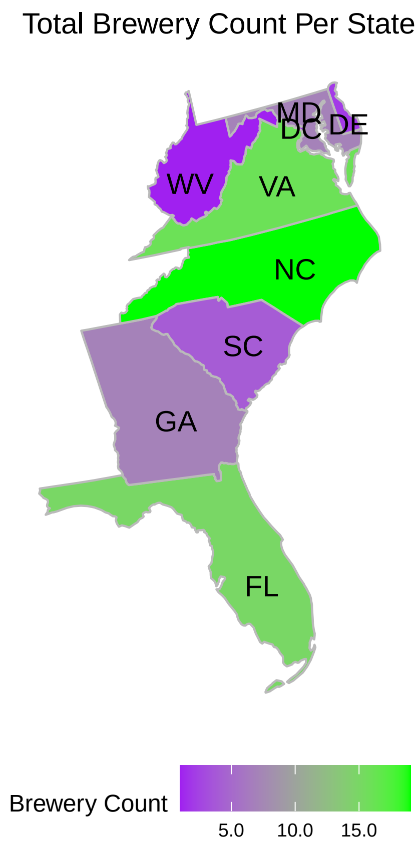

```
########################################################
########################################################
#Break down by region, East North Central third
ENCplot <- plot_usmap(data=statepop, values = "brewByState",labels = TRUE,include = .east_north_central
ENCplot
```



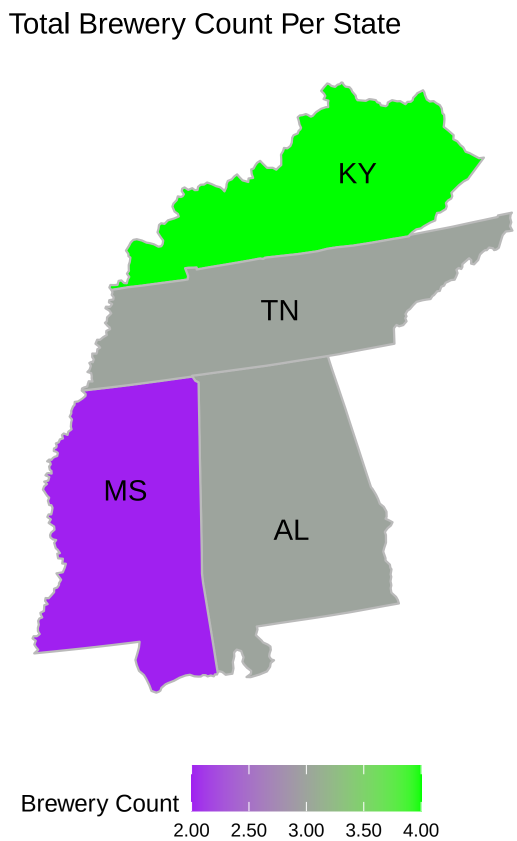

```
##########################################################
##########################################################
#Break down by region, West North Central fourth
WNCplot <- plot_usmap(data=statepop, values = "brewByState",labels = TRUE,include = .west_north_central
WNCplot
```



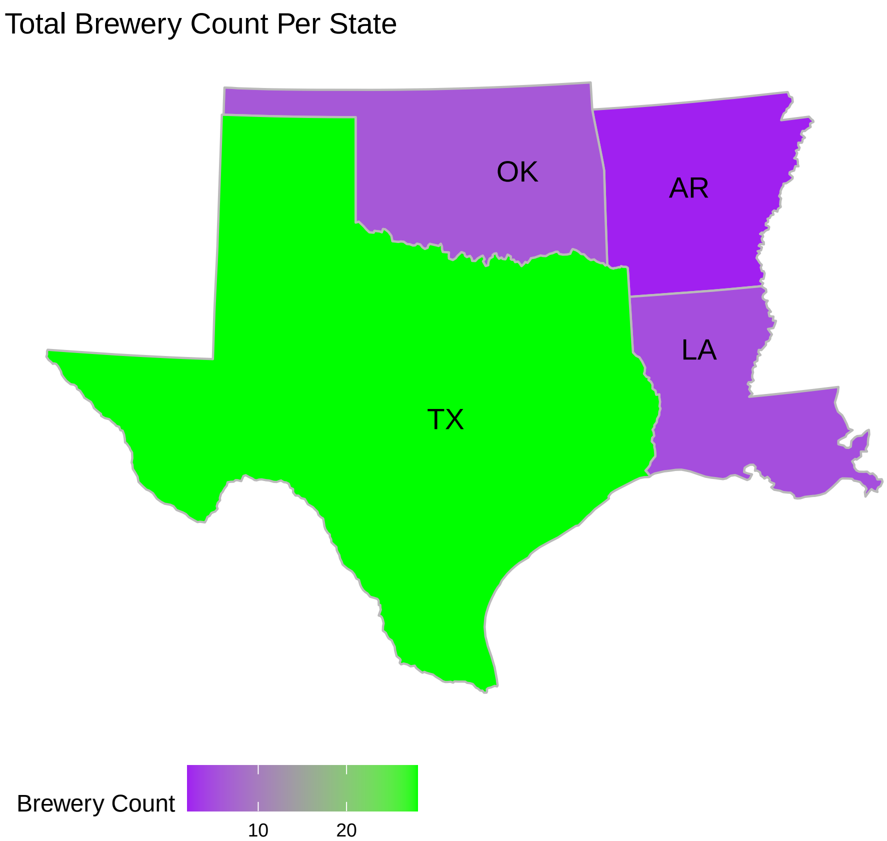

```
##########################################################
##########################################################
#Break down by region, South Atlantic fifth
SAplot <- plot_usmap(data=statepop, values = "brewByState",labels = TRUE,include = .south_atlantic,colo
SAplot
```



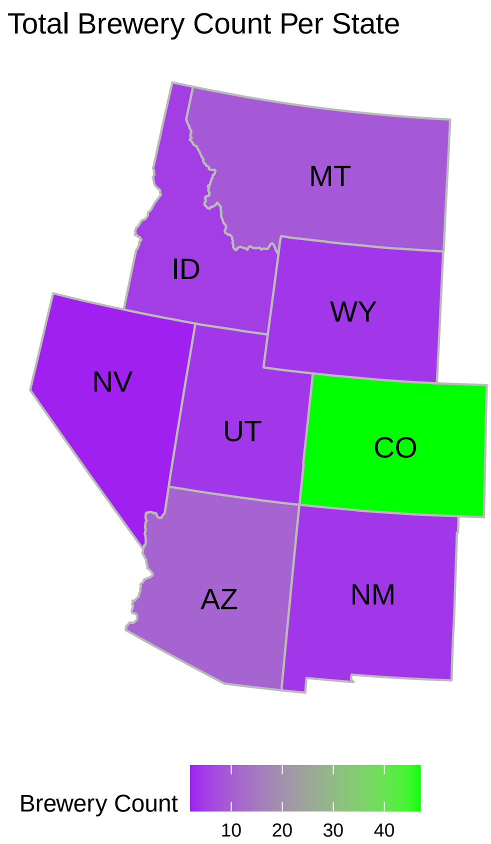

```
##########################################################
##########################################################
#Break down by region, East South Central sixth
ESCplot <- plot_usmap(data=statepop, values = "brewByState",labels = TRUE,include = .east_south_central
ESCplot
```



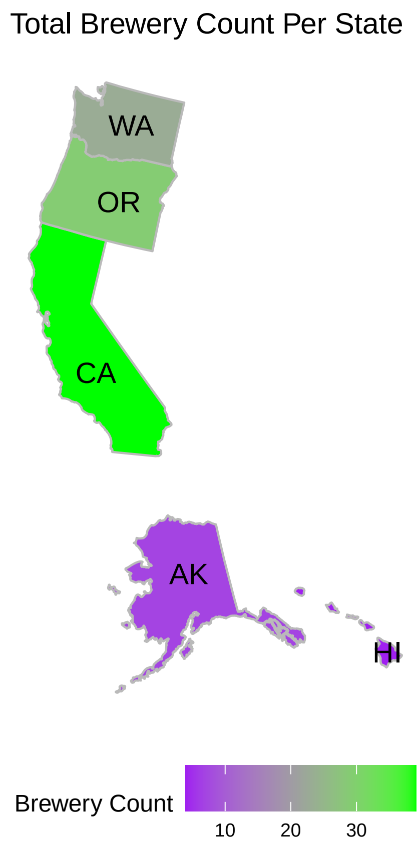

```
##########################################################
##########################################################
#Break down by region, West South Central seventh
WSCplot <- plot_usmap(data=statepop, values = "brewByState",labels = TRUE,include = .west_south_central
WSCplot
```



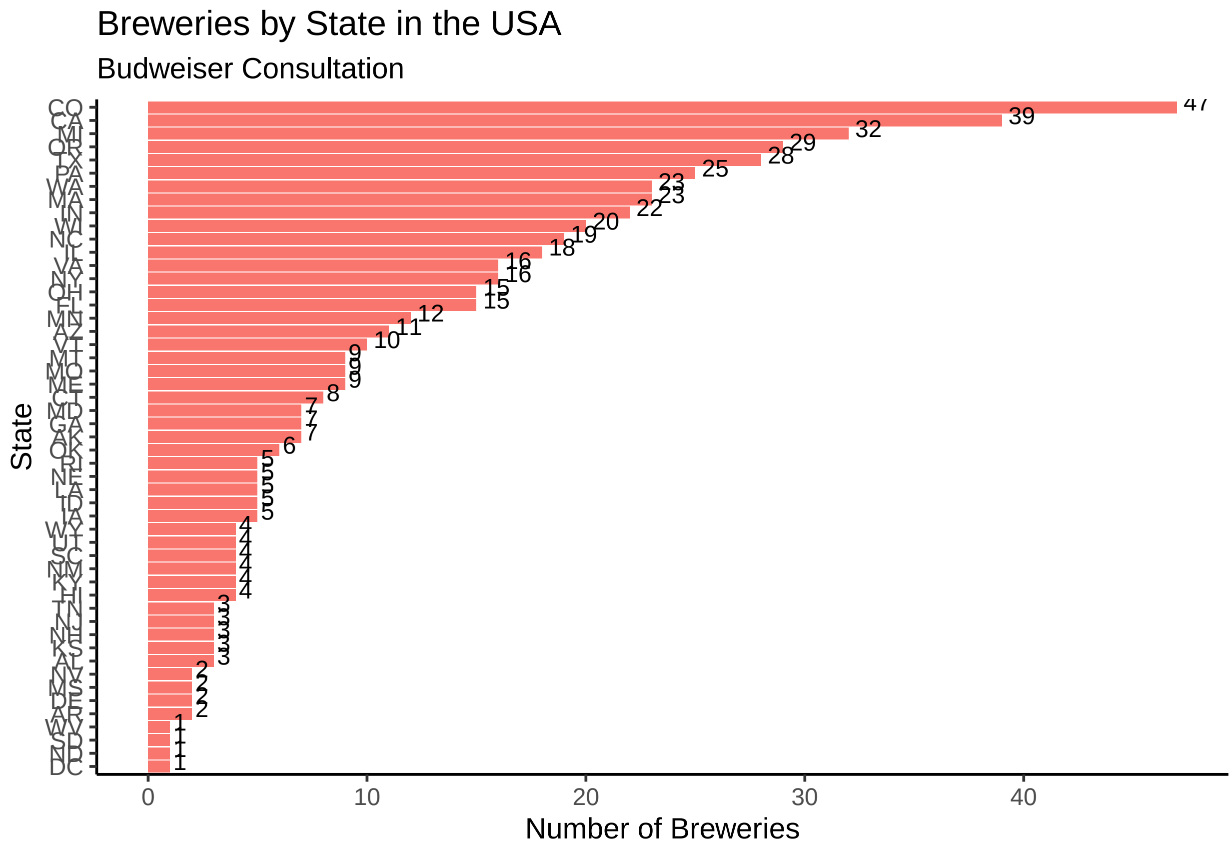

```
##########################################################
##########################################################
#Break down by region, Mountain eighth
Mplot <- plot_usmap(data=statepop, values = "brewByState",labels = TRUE,include = .mountain,color = "gr
Mplot
```




```
##########################################################
##########################################################
#Break down by region, Pacific ninth
Pplot <- plot_usmap(data=statepop, values = "brewByState",labels = TRUE,include = .pacific,color = "grey
Pplot
```




```
############################################################
############################################################

#### Bar Plot ####

#Plot overall breweries by state in bar chart

brewByState %>% ggplot(aes(y=reorder(State, n), x= n, fill = "#C8102E")) +
  geom_bar(stat = "identity", show.legend = FALSE, position = 'dodge') +
  geom_text(aes(label = brewByState$n), position=position_dodge(width=0.9), hjust = -0.25, vjust= .2, s
  theme_classic() +
  labs(title = "Breweries by State in the USA",
       subtitle = "Budweiser Consultation",
       x = "Number of Breweries", y = "State")
```




```
######################
#                    #
#     Question 2     #
#                    #
######################
########################################################################################################
# Question 2 - Merge the individual data sets
#
# We merged the breweries.csv dataset with the beers.csv dataset, additionally when we imported
# the individual datasets, we also imported a dataset that allows us to associate each beer with
# its brewery's US Census Division.
#
########################################################################################################
#Use Dplyr package to merge the two tables together
buzzbrews <- merge(breweryDat, beerDat, by.x = "Brew_ID", by.y = "Brewery_id", all = TRUE )
#Use Dplyr package to rename "Name.x" to "Brewery" and "Name.y" to "Beer"
buzzbrews <- dplyr::rename(buzzbrews, "Brewery" = "Name.x","Beer"="Name.y")
bzbwTestDf <- buzzbrews
#Check the results
#View(buzzbrews)
```

```
######################
#                    #
#     Question 3     #
#                    #
######################
###########################################
```

```
# Question 3 - Address the missing values in each column.
#
# During the initial exploratory process we discovered NA's in both the IBU and ABV columns.
# Upon further investigation we determined that some styles of beer, mixed or barrel aged beers
# do not have an ABV available at the time the brewery submits packaging labels to TTB, or Alcohol
# and Tobacco Tax and Trade Bureau. The TTB is the federal agency that determines what can and cannot
# be put on a beer label including the art, type size, verbiage, where elements are placed and etc.
# So beers without an AVB available either do not inlcude it, or add it to the bottom of the cans
# or packaging at a later date.
#
# In terms of the missing IBU values, we determined that even though the IBU alludes to the bitterness
# of a beer's taste, it is somewhat misleading because it is derived from a test that measures differen
# chemical compounds that are known to cause bitter flavoring. For instance, a beer may have a high IBU
# value, but due to other ingredients, such as added lactose or sucrose may actually have a sweeter tas
# than would be expected from a high IBU. The other comfounding variable is if the brewery can afford t
# equipment used to generate an IBU value, smaller breweries simply cannot afford it while the larger
# breweries typically just use IBU as a quality control measure.
#
# Finally, we concluded that imputing data or filling in the missing gaps was a good idea for this
# analysis and that was done by taking an average of from similiar styles of beer and assigning that to
# beers in the same sytle classification that did not have values. Upon random testing of different imp
# values, by googling beers that had missing values in the dataset and comparing that to the created av
# it was determined that the imputed values were very close to the actual values in the marketplace.
#
###########################################
#Loop to fix numbering for Column 1 "brew ID"
iterations <- length(buzzbrews$Brew_ID)
for (i in 1:iterations) {
  buzzbrews[i,1]=i
}
#Fix no style beers to none
levels(buzzbrews$Style) <- c(levels(buzzbrews$Style), "none")
for (i in 1:iterations) {
  if(is.na(buzzbrews[i,9])){
  }
}
for (i in 1:iterations) {
  if((buzzbrews[i,9])==''){
    #print(buzzbrews[i,9])
    buzzbrews[i,9]="none"
  }
}
#Prep new df to contain style and averages
buzzbrews$Style <- as.factor(buzzbrews$Style)
#Create a data frame with each style and a variable for average IBU
styleCount <- as.data.frame(levels(buzzbrews$Style))
styleCount$`levels(buzzbrews$Style)` <- as.character(styleCount$`levels(buzzbrews$Style)`)
#View(styleCount)
#Initialize mean ibu to zero (to avoid problems with N/As)
styleCount$meanIbu <- 0
#Make beer count to keep track of total in each style
styleCount$beerCount <- 0
#Make column for total ibus
```

```r
styleCount$totalIBU <- 0
styleCount$meanABV <- 0
styleCount$ABVbeerCount <- 0
styleCount$totalABV <- 0
#Checking
#View(styleCount)
#styleCount <- styleCount[-c(1), ]
#View(styleCount)
#Calculate mean IBU for each category and store it in IBU df
#Calculate average IBU for each style and add it to df
#outer loop for all the beers
ibuSum <- 0
beerCount <- 0
i <- 1
for (i in 1:iterations) {
  if(is.na(buzzbrews[i,8])) {
    buzzbrews[i,8]=0
  }

  #inner for each style
  for (j in 1:100) {

    if(buzzbrews[i,9]==styleCount[j,1]){
     #Compute IBU sum
     styleCount[j,4] <- styleCount[j,4]+buzzbrews[i,8]



     #Total of each beer count
     styleCount[j,3] <- styleCount[j,3]+1

     if(buzzbrews[i,8]==0){
       styleCount[j,3] <- styleCount[j,3]-1
     }




    }
    #Mean IBU for each style
    styleCount[j,2] <- styleCount[j,4]/styleCount[j,3]
    }}
#Add average column from style count to buzzbrews df
for (i in 1:iterations) {
  if(buzzbrews[i,8]==0){
    for(j in 1:100){
      if(buzzbrews[i,9]==styleCount[j,1]){
        buzzbrews[i,8]=styleCount[j,2]
      }
    }
  }
}
# View(styleCount)
# View(buzzbrews)
```

```
# Now do it all again for ABV
# Calculate average ABV for each style and add it to df
# outer loop for all the beers
AlcSum <- 0
AlcVeerCount <- 0
i <- 1
for (i in 1:iterations) {
  if(is.na(buzzbrews[i,7])) {
   buzzbrews[i,7]=0
  }


  #inner for each style
  for (j in 1:100) {

    if(buzzbrews[i,9]==styleCount[j,1]){

     #Compute ALC sum
     styleCount[j,7] <- styleCount[j,7]+buzzbrews[i,7]*100



     #Total of each beer count
     styleCount[j,6] <- styleCount[j,6]+1

     if(buzzbrews[i,7]==0){
       styleCount[j,6] <- styleCount[j,6]-1
     }



    }
    #Mean ABV for each style
    styleCount[j,5] <- (styleCount[j,7]/styleCount[j,6])/100
    }
}
#Add average column from style count to buzzbrews df
for (i in 1:iterations) {
  if(buzzbrews[i,7]==0){
    for(j in 1:100){
      if(buzzbrews[i,9]==styleCount[j,1]){
        buzzbrews[i,7]=styleCount[j,5]

      }
      }
  }
}
#kill NaN's for other alcohol types with no hops
i <- 1
for(i in 1:iterations){
  if(is.na(buzzbrews[i,8])){
    buzzbrews[i,8] <- 0
  }
```

```r
}
#Check out end results
buzzbrews <- merge(buzzbrews, regionData, by = "State")

View(buzzbrews)
```

```r
######################
#                    #
#     Question 4     #
#                    #
######################
####################################################################################################
# Question 4 - Compute the median alcohol content and international bitterness unit for
# each state. Plot a bar chart to compare.
#
# We computed the MedStateABV and IBU for each state and created a visualisation that allowed
# us to further explore what those medians tell us. We found there appears to be a relationship
# between IBU and ABV where we can use IBU to estimate ABV of a given beer.
#
# We explored this further by developing a model to make predictions based on historical IBU
# and ABV data and were able to predict that a beer with 32 IBU could have an ABV of 5.72% and
# we were 97.5% confident that beer would at least fall between 3.24% and 8.21%.
#
####################################################################################################
buzzbrews$State <- trimws(buzzbrews$State)

# Group by state and compute
combineddf <- buzzbrews %>%
  group_by(State) %>%
  dplyr::summarise(MedStateIBU = median(IBU), MedStateABV = median(ABV))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
combineddf <- as.data.frame(combineddf)
combineddf$MedStateIBU <- as.numeric(combineddf$MedStateIBU)
combineddf$MedStateABV <- as.numeric(combineddf$MedStateABV)

# Divisional measurements
divisiondf <- buzzbrews %>%
  group_by(Division) %>%
  dplyr::summarise(MedDivIBU = median(IBU), MedDivABV = median(ABV))
```

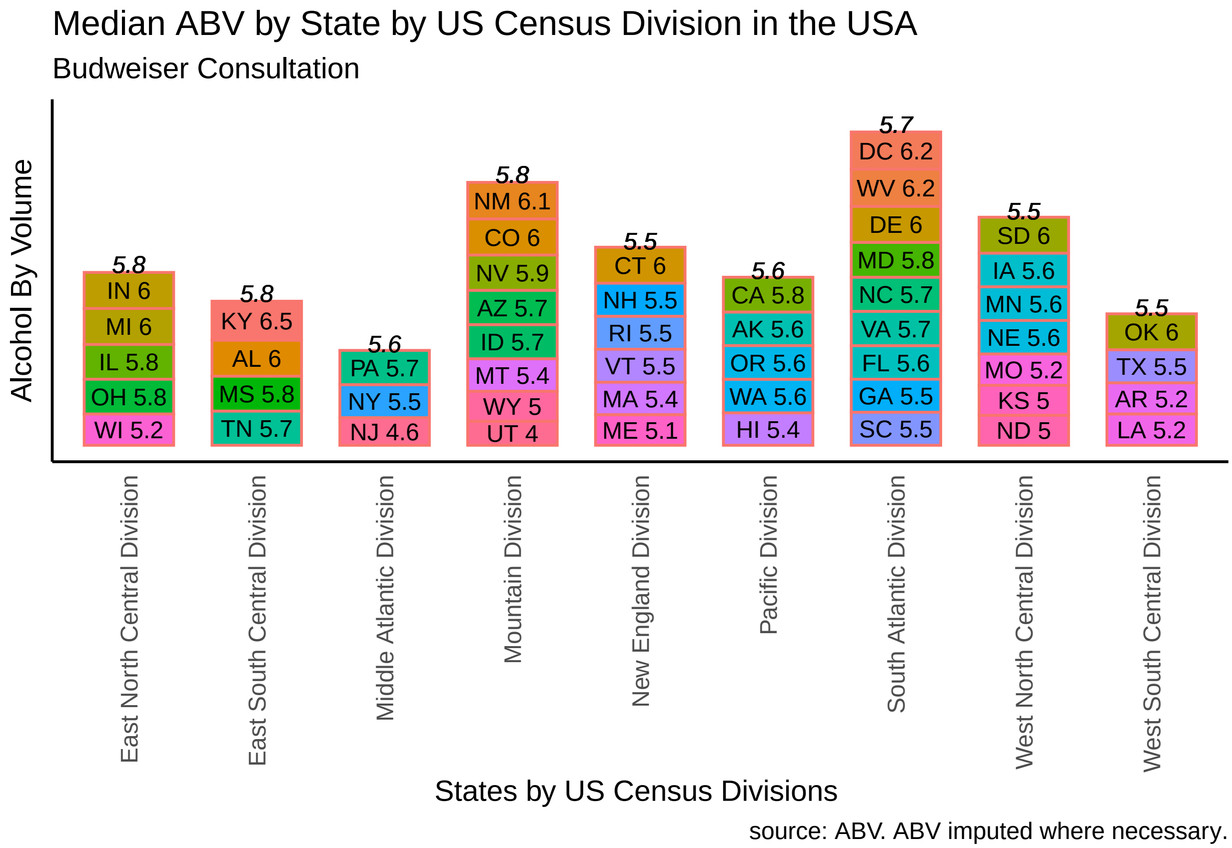
```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
# round values to xx.x ###
divisiondf$MedDivIBU <- round(divisiondf$MedDivIBU, digits = 1)
divisiondf$MedDivABV <- round(divisiondf$MedDivABV, digits = 1)
combineddf$MedStateIBU <- round(combineddf$MedStateIBU, digits = 1)
combineddf$MedStateABV <- round(combineddf$MedStateABV, digits = 1)

# Add regions to combinddf
combineddf <- merge(combineddf,regionData,by="State")

# Add in divisional values
combineddf <- merge(combineddf, divisiondf, by = "Division")
```

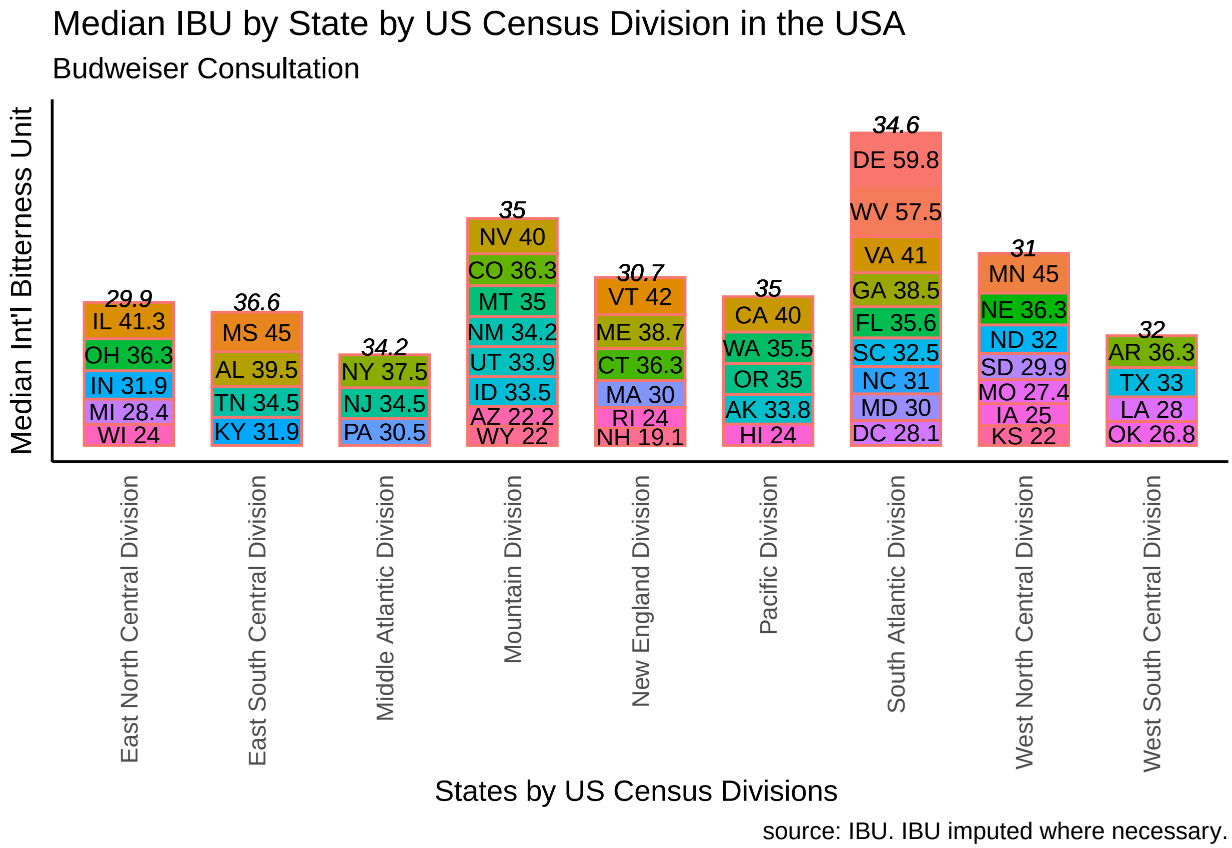
```r
####### Create chart labels for stacked charts #####
combineddf$ABVlabel <- paste(combineddf$State, combineddf$MedStateABV)
combineddf$IBUlabel <- paste(combineddf$State, combineddf$MedStateIBU)

view(combineddf)

# Create sums of medians for labeling charts #
StateSums <- combineddf %>%
  group_by(Division) %>%
  dplyr::summarise(SumStateABV = sum(MedStateABV), SumStateIBU = sum(MedStateIBU))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
combineddf <- merge(combineddf, StateSums, by = "Division")

#
############################################################
#########                                  ################
######### Draw Bar Chart = Median State ABV ################
#########                                  ################
############################################################
#
combineddf %>%
  ggplot(aes(x=Division, y=MedStateABV,fill= reorder(State,-MedStateABV))) +
  # Create stacked by chart organized by Division with States stacked in each bar
  geom_bar(aes(color = "#c8102e"),stat="identity", width= 0.7, position = position_stack(), show.legend
  # Add state and ABV value to each state's chart position
  geom_text(aes(label = ABVlabel), size = 3, position = position_stack(vjust = 0.5)) +
  # Add Division ABV Values to top of each chart stack
  geom_text(aes(Division, MedDivABV + SumStateABV -3, label = MedDivABV), size = 3, vjust = 1, fontface
  # Label the chart objects
  labs(title="Median ABV by State by US Census Division in the USA",
       subtitle="Budweiser Consultation",
       caption="source: ABV. ABV imputed where necessary.",
       y = "Alcohol By Volume",
       x = "States by US Census Divisions ") +
  theme_classic() +
  # Adjust the X-axis labels, remove y-labels since this is a stacked chart
  theme(axis.text.x = element_text(angle=90, vjust = 0.5,hjust = 1),
        axis.text.y = element_blank(), axis.ticks = element_blank())
```

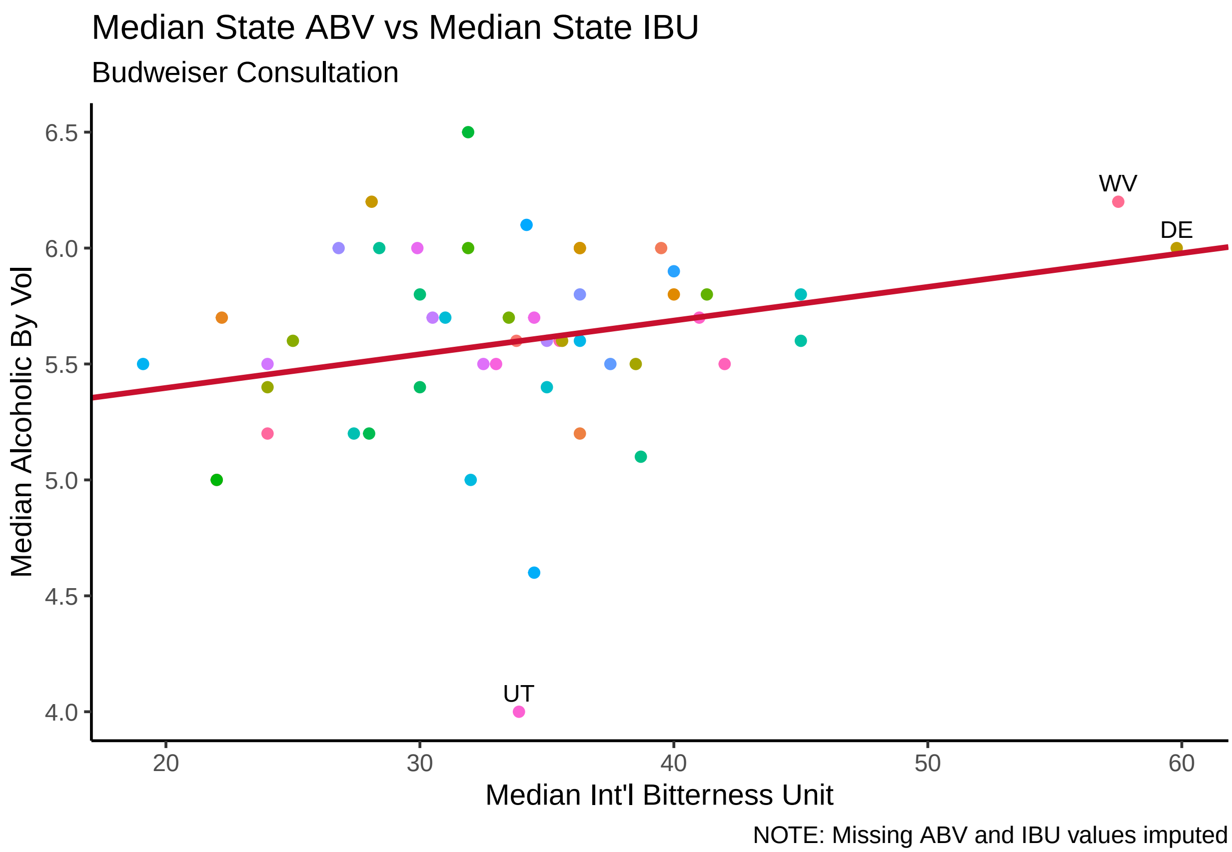



```
#
##### Create bar plot for IBU #####
#
combineddf %>%
  ggplot(aes(x=Division, y=MedStateIBU,fill= reorder(State,-MedStateIBU))) +
  # Create stacked by chart organized by Division with States stacked in each bar
  geom_bar(aes(color = "#c8102e"),stat="identity", width= 0.7, position = position_stack(), show.legend
  # Add state and IBU value to each state's chart position
  geom_text(aes(label = IBUlabel), size = 3, position = position_stack(vjust = 0.5)) +
  # Add Division IBU Values to top of each chart stack
  geom_text(aes(Division, MedDivIBU + SumStateIBU - 15, label = MedDivIBU), size = 3, vjust = 1, fontfa
  # Label the chart objects
  labs(title="Median IBU by State by US Census Division in the USA",
       subtitle="Budweiser Consultation",
       caption="source: IBU. IBU imputed where necessary.",
       y = "Median Int'l Bitterness Unit",
       x = "States by US Census Divisions ") +
  theme_classic() +
  # Adjust the X-axis labels, remove y-labels since this is a stacked chart
  theme(axis.text.x = element_text(angle=90, vjust = 0.5,hjust = 1),
        axis.text.y = element_blank(), axis.ticks = element_blank())
```




```
#
#
##################################################
##                                              ##
## Scatterplot MedStateIBU vs MedStateABV by State  ##
##                                              ##
##################################################

## Calculate slope and intercept of line of best fit ##
abline_values <- coef(lm(MedStateABV ~ MedStateIBU, combineddf))
#  (Intercept) MedStateABV
#   14.5377926     0.4013998

ggplot(combineddf,aes(x = MedStateIBU, y = MedStateABV, color = State)) +
  geom_point(show.legend = FALSE) +
  # Add ABLine to the chart to see if there is a linear relationship
  geom_abline(intercept = abline_values[1] , slope = abline_values[2] , color = "#c8102E", size = 1) +
  # Add state labels, but only for outliers
  geom_text(data = subset(combineddf,MedStateIBU > 45 | MedStateABV < 4.5,
                          select = c(State, MedStateIBU, MedStateABV)),
            aes(label = State), vjust= -0.6, size = 3, na.rm = TRUE,
            show.legend = FALSE, color = "#000000") +
  theme_classic() +
  labs(title = "Median State ABV vs Median State IBU",
       subtitle = "Budweiser Consultation",
       y = "Median Alcoholic By Vol",
```

```
       x = "Median Int'l Bitterness Unit",
       caption = "NOTE: Missing ABV and IBU values imputed")
```




```
######################
#                    #
#     Question 5     #
#                    #
######################
##########################################################################################################
# Question 5 - Which state has the maximum alcoholic (ABV) beer? Which state has the most bitter (IBU)
#
# We determined that the maximum observed IBU was 138 in Oregon for Bitter Bitch Imperial IPA that
# is an American Double/ Imperial IPA from the Astoria Brewing Company in Austoria, OR.
#
# We also determined that maximum observed ABV was 12.8% in Colorado for Lee Hill Series Vol. 5 -
# Belgian Style Quadrupel Ale from Upslope Brewing Company in Boulder, CO.
#
##########################################################################################################
#Figure out which has highest ABV
MaxStateABV <- arrange(buzzbrews, desc(ABV))
print(MaxStateABV[1,4])
```

```
## [1] "Boulder"
```

```
#Figure out which has highest IBU
maxIBU <- arrange(buzzbrews,desc(IBU))
print(maxIBU[1,4])
```

```
## [1] "Astoria"
```

```r
##### Question 5 Answer #####
## Colorado has the highest ABV = 12.8, Oregon has the highest IBU = 138.
###################################################
###### Create DF for just the max ABV & IBU values ######
# State measurements
maxStateValues <- buzzbrews %>%
  group_by(State) %>%
  dplyr::count(MaxStateABV = max(ABV), MaxStateIBU = max(IBU))
maxStateValues <- maxStateValues[,-4]
maxStateValues <- as.data.frame(maxStateValues)
maxStateValues$State <- trimws(maxStateValues$State)
str(maxStateValues)
```

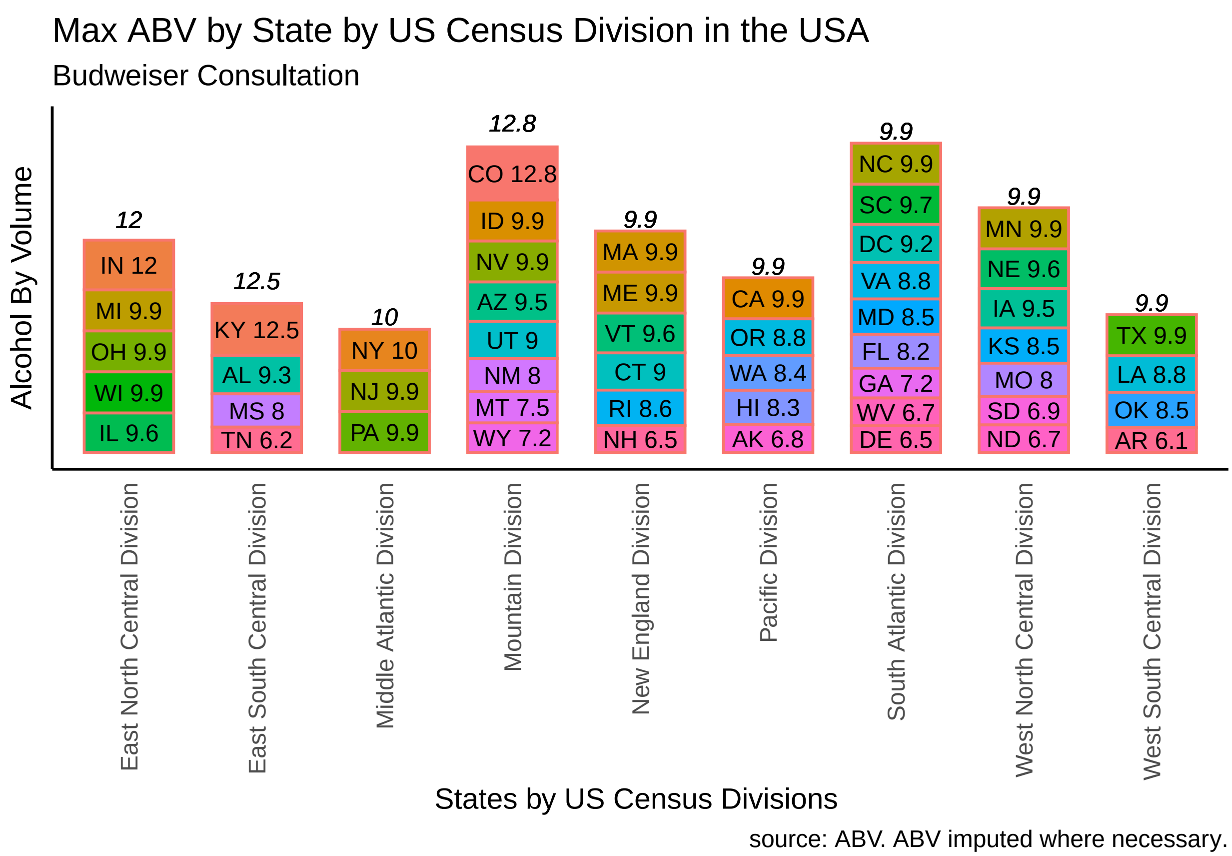
```
## 'data.frame':    51 obs. of  3 variables:
##  $ State      : chr  "AK" "AL" "AR" "AZ" ...
##  $ MaxStateABV: num  6.8 9.3 6.1 9.5 9.9 ...
##  $ MaxStateIBU: num  71 103 45.7 99 115 ...
```

```r
view(maxStateValues)

# Divisional measurements
divMaxValdf <- buzzbrews %>%
  group_by(Division) %>%
  dplyr::count(MaxDivABV = max(ABV), MaxDivIBU = max(IBU))
divMaxValdf <- divMaxValdf[,-4]
divMaxValdf <- as.data.frame(divMaxValdf)


# round values to xx.x ###
maxStateValues$MaxStateABV <- round(maxStateValues$MaxStateABV, digits = 1)
maxStateValues$MaxStateIBU <- round(maxStateValues$MaxStateIBU, digits = 1)

# Add regions to maxStateValues
maxStateValues <- merge(maxStateValues,regionData,by="State")

# Add in divisional values
maxStateValues <- merge(maxStateValues, divMaxValdf, by = "Division")

####### Create chart labels for stacked charts #####
maxStateValues$ABVmaxLabel <- paste(maxStateValues$State, maxStateValues$MaxStateABV)
maxStateValues$IBUmaxLabel <- paste(maxStateValues$State, maxStateValues$MaxStateIBU)

view(maxStateValues)

# Create sums of max values for labeling charts #
StateMaxSums <- maxStateValues %>%
  group_by(Division) %>%
  dplyr::summarise(SumStateABV = sum(MaxStateABV), SumStateIBU = sum(MaxStateIBU))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

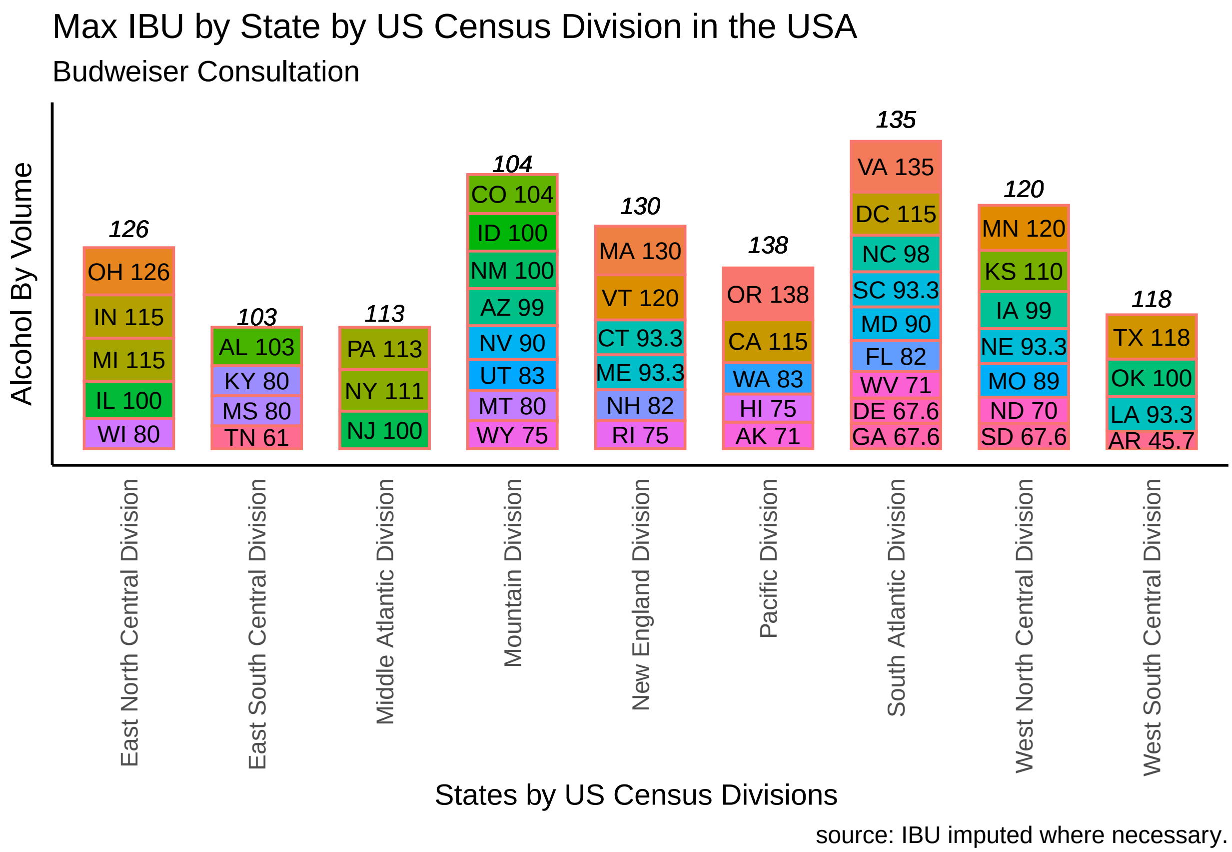
```r
maxStateValues <- merge(maxStateValues, StateMaxSums, by = "Division")
###################################################
###### Plot for Max ABV ##########################
```

```
maxStateValues %>%
  ggplot(aes(x=Division, y=MaxStateABV,fill= reorder(State,-MaxStateABV))) +
  # Create stacked by chart organized by Division with States stacked in each bar
  geom_bar(aes(color = "#c8102e"),stat="identity", width= 0.7, position = position_stack(), show.legend
  # Add state and ABV value to each state's chart position
  geom_text(aes(label = ABVmaxLabel), size = 3, position = position_stack(vjust = 0.5)) +
  # Add Division ABV Values to top of each chart stack
  geom_text(aes(Division, MaxDivABV + SumStateABV, label = MaxDivABV), size = 3, nudge_y = -7, fontface
  # Label the chart objects
  labs(title="Max ABV by State by US Census Division in the USA",
       subtitle="Budweiser Consultation",
       caption="source: ABV. ABV imputed where necessary.",
       y = "Alcohol By Volume",
       x = "States by US Census Divisions ") +
  theme_classic() +
  # Adjust the X-axis labels, remove y-labels since this is a stacked chart
  theme(axis.text.x = element_text(angle=90, vjust = 0.5,hjust = 1),
        axis.text.y = element_blank(), axis.ticks = element_blank())
```



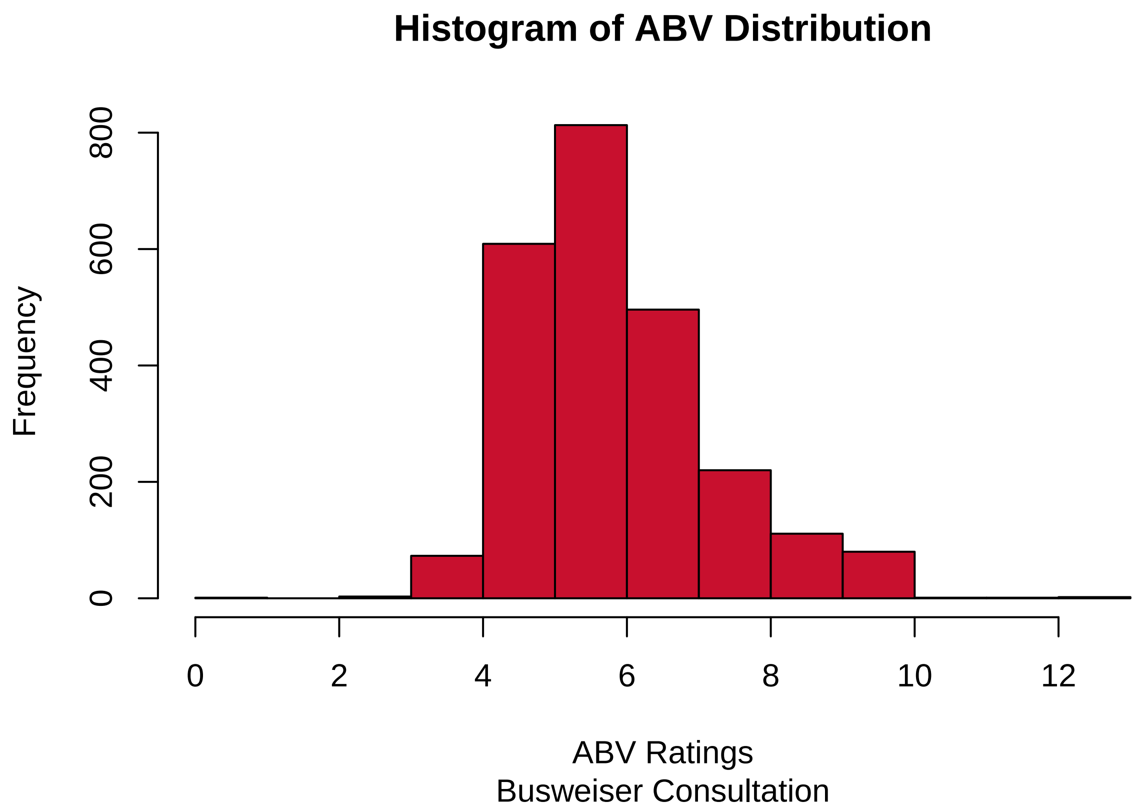

```
################################################
############### Chart Max IBU #####################

maxStateValues %>%
  ggplot(aes(x=Division, y=MaxStateIBU,fill= reorder(State,-MaxStateIBU))) +
  # Create stacked by chart organized by Division with States stacked in each bar
  geom_bar(aes(color = "#c8102e"),stat="identity", width= 0.7, position = position_stack(), show.legend
```

```r
  # Add state and ABV value to each state's chart position
  geom_text(aes(label = IBUmaxLabel), size = 3, position = position_stack(vjust = 0.5)) +
  # Add Division ABV Values to top of each chart stack
  geom_text(aes(Division, MaxDivIBU + SumStateIBU, label = MaxDivIBU), size = 3, nudge_y = -75, fontface
  # Label the chart objects
  labs(title="Max IBU by State by US Census Division in the USA",
       subtitle="Budweiser Consultation",
       caption="source: IBU imputed where necessary.",
       y = "Alcohol By Volume",
       x = "States by US Census Divisions ") +
  theme_classic() +
  # Adjust the X-axis labels, remove y-labels since this is a stacked chart
  theme(axis.text.x = element_text(angle=90, vjust = 0.5,hjust = 1),
        axis.text.y = element_blank(), axis.ticks = element_blank())
```

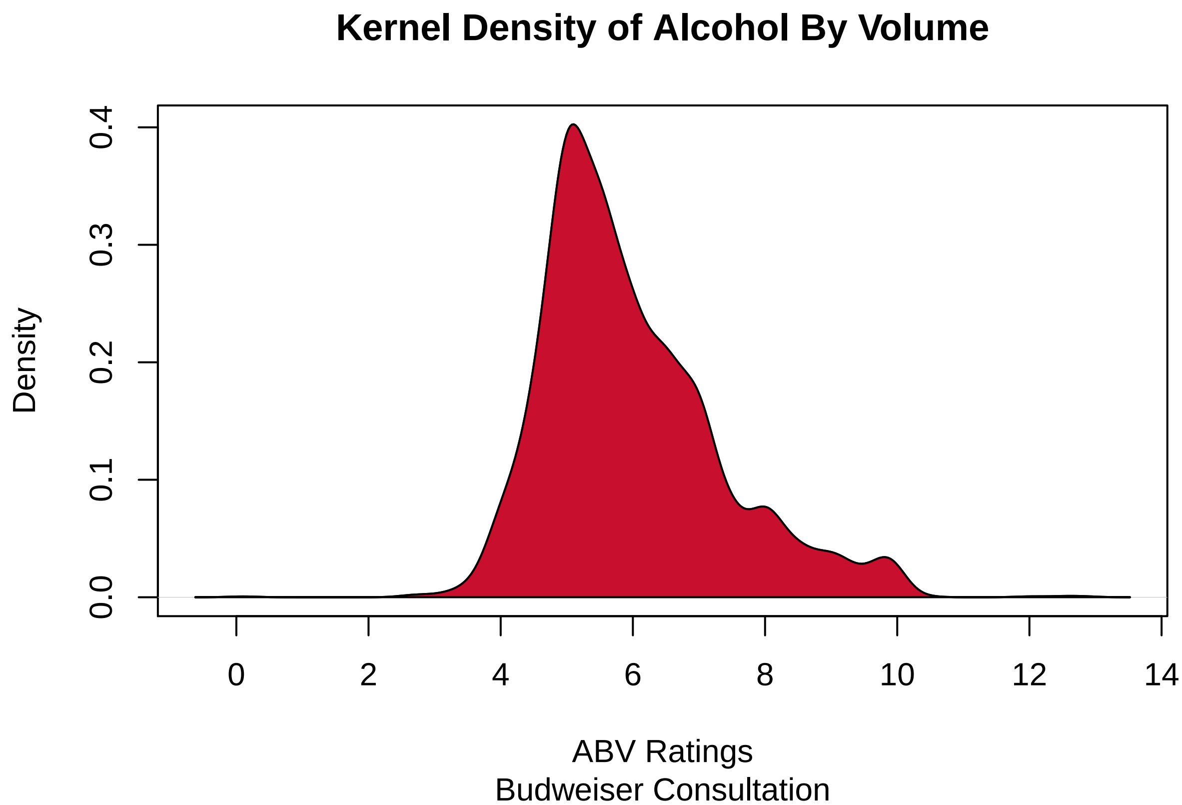



```r
######################
#                    #
#     Question 6     #
#                    #
######################
########################################################################################################
# Question 6 - Comment on the summary statistics and distribution of the ABV variable.
#
# We observed summary statistics from the ABV data showing that once we filled in the missing
# values as best as we could, there was a range of 0.10% to 12.80% with a median of 5.65% (median
# is simply the middle value if we were to arrange all the ABVs in either decending or ascending order)
```

```r
#
# We also found a very common range within the overall range that went from 5.0% ABV to 6.70% ABV and
# upon further review noticed this is where many commonly mass produced beers fall, for example: Bud
# Ice (5.5%), Bud Light Platinum (6%), Natural Ice (5.9%), Bud Ice (5.5%), Budweiser (5%), Blue Moon
# (5%), Stella Artois (5%), Heinekin (5%), Pabst Blue Ribbon (4.74%) and Miller Genuinine Draft (4.6%).
#
##########################################################################################################
# Check on the distribution of ABV
hist(buzzbrews$ABV, col = "#c8102e",
     main = "Histogram of ABV Distribution",
     sub = "Busweiser Consultation",
     xlab = "ABV Ratings")
```




```r
densityABV <- density(buzzbrews$ABV)
plot(densityABV,
     main = "Kernel Density of Alcohol By Volume",
     sub = "Budweiser Consultation",
     xlab = "ABV Ratings")
polygon(densityABV, col = "#c8102e")
```




```
ABVsummary <- summary(buzzbrews$ABV)
ABVsummary
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.100   5.000   5.650   5.975   6.700  12.800
```

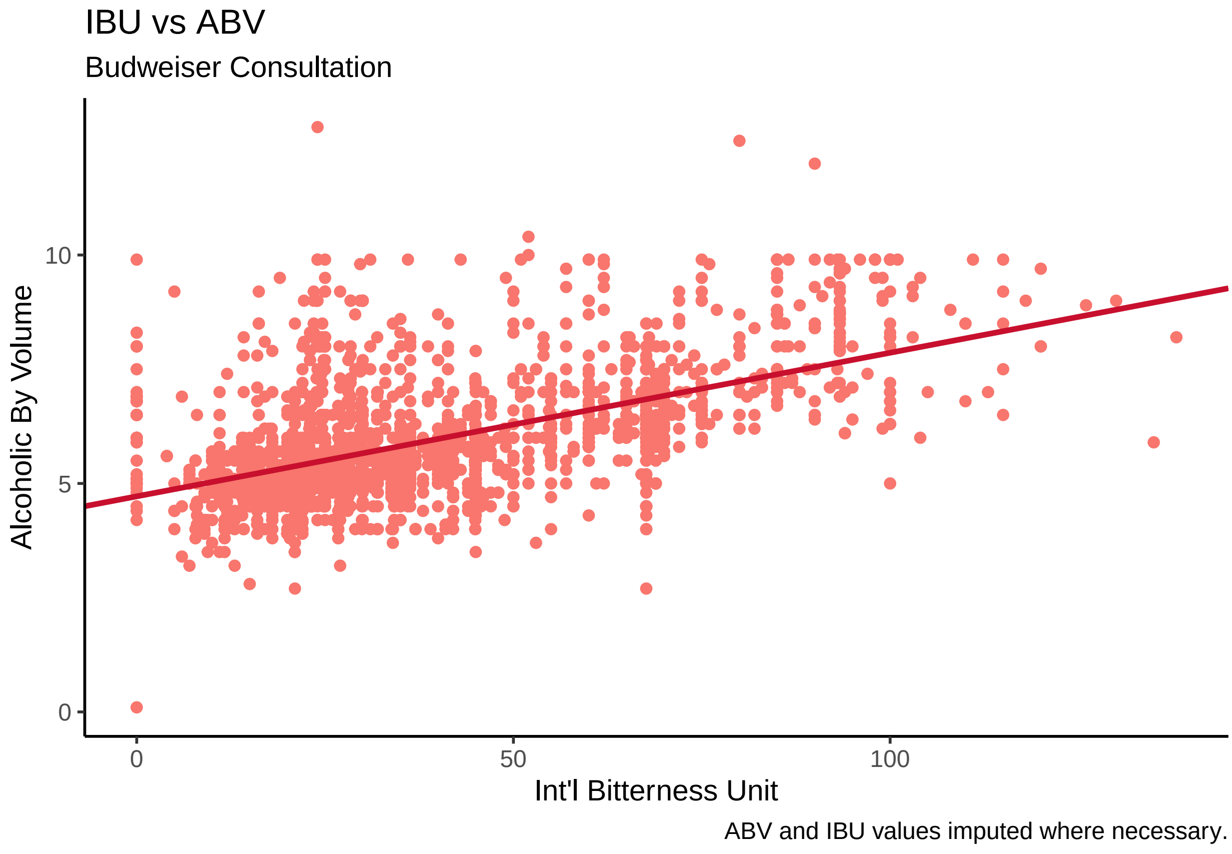
```
############### Not sure what we want to do down here #################
#ABVSumNames <- names(ABVsummary)
#ABVSumNames <- as.factor(ABVSumNames)
#ABVsummary <- as.list(unname(ABVsummary))
#ABVsum <- data.frame(ABVsummary)
#ABVsum <- (ABVSumNames)
#as.data.frame(ABVsum)
#ABVsum <- rename(ABVsum, "Summary"="", "Value"="ABVsummary")
######################################################################
```

```
######################
#                    #
#     Question 7     #
#                    #
######################
##########################################################################################################
# Question 7 - Is there an apparent relationship between the bitterness of the beer and its
# alcoholic content? Draw a scatter plot.  Make your best judgment of a relationship and
# EXPLAIN your answer.
#
# We used a scatter plot to viusally explore if there is any sort of relationship between
```

```r
# IBU and ABV, in other words can IBU determine ABV or can ABV be used to determine IBU.
# There was evidence of a positive relationship, but and we will discuss this further shortly,
# it appears one can potentially predict the other.
#
################################################################################################
## Calculate slope and intercept of line of best fit ##
comparisonCoef <- coef(lm(ABV ~ IBU, buzzbrews))
comparisonCoef
```

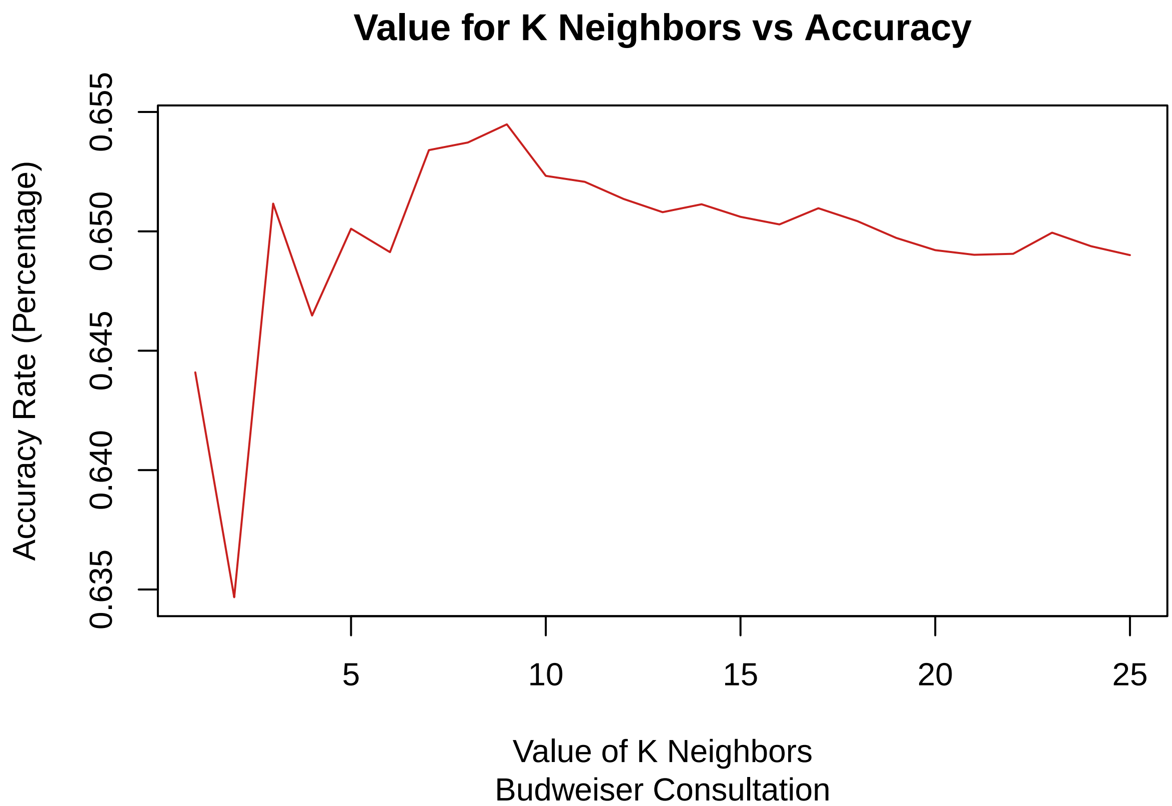
```
## (Intercept)         IBU
##  4.71799073  0.03142639
```

```r
#  (Intercept)      MaxIBU
#   4.71799073  0.03142639

buzzbrews %>%
  ggplot(aes(x = IBU, y = ABV, color = "#c8102e")) +
  geom_point(show.legend = FALSE, na.rm = TRUE) +
  geom_abline(intercept =  comparisonCoef[1] , slope = comparisonCoef[2], color = "#c8102E", size = 1) +
  theme_classic() +
  labs(title = "IBU vs ABV",
       subtitle = "Budweiser Consultation",
       y = "Alcoholic By Volume",
       x = "Int'l Bitterness Unit",
       caption="ABV and IBU values imputed where necessary.")
```



```
######################
#                    #
#     Question 8     #
#                    #
######################
##############################################################################################
# Question 8 - .  Budweiser would also like to investigate the difference with respect to IBU and ABV
# between IPAs (India Pale Ales) and other types of Ale (any beer with "Ale" in its name other than IPA
# You decide to use KNN classification to investigate this relationship.  Provide statistical evidence
# way or the other. You can of course assume your audience is comfortable with percentages ... KNN is v
# to understand conceptually.
# In addition, while you have decided to use KNN to investigate this relationship (KNN is required) you
# also feel free to supplement your response to this question with any other methods or techniques you
# learned.  Creativity and alternative solutions are always encouraged.
#
# Response:
# We built a kNN (nearest neighbor) classifier to see if there is a difference between IPA and Ale, and
# we were at it, we also added in a third class called, "neither." In building the kNN, we wanted to ex
# what the appropriate number of "neighbors" was to compare to since there is so many observations so c
# together (think New York city and all the noise generated). We found that generally 8 neighbors were
# best estimation (we randomly parsed the data 100 times to find the best neighbors value).
#
# Our classifier was accurate in determining if a beer was an Ale, IPA or neither about 64.5% of the ti
# when we used 8 nearest neighbors.
# Next we created some random pairings of IBU and ABV to see how the classifier handled the data and di
# it again was about 64.5% accurate. It is far more accurate identifyly neither style of beer 78% of th
# then IPAs 67.5% of the time and Ale's 26% of the time.
# We also look a look at the ranges for IBU and ABV for each of the 3 broad types of beers IPA, Ale or
# "neither" and found the following results, showing that it should be more difficult to predict betwee
# the 3 different types of beers.
#
# IPAAle  ABV.min ABV.med ABV.max IBU.min IBU.med IBU.max
# Ale         3.5     5.4    12.8       7    31       120
# IPA         4       6.7     9.9      19    67.6     138
# neither     0.1     5.5    12.5       0    28       130
#
# Additionally we re-visualized the plot chart with a regression line from the previous question, this
# time showing the plots colored based on the classification of Ale, IPA or neither.
#
##############################################################################################


#Label Ales, IPAs and neither
buzzbrews$IPAAle = case_when(grepl("\\bIPA\\b", buzzbrews$Beer, ignore.case = TRUE) ~ "IPA",
                             grepl("\\bindia pale ale\\b", buzzbrews$Beer, ignore.case = TRUE) ~ "IPA",
                             grepl("\\bale\\b", buzzbrews$Beer, ignore.case = TRUE ) ~ "Ale",
                             TRUE ~ "neither")
view(buzzbrews)

##### Find the best value of K and train the model ##############
iterations = 100
numks = 25
splitPerc = .70
```

```r
set.seed(33)

masterAcc = matrix(nrow = iterations, ncol = numks)

for(j in 1:iterations)
{
accs = data.frame(accuracy = numeric(30), k = numeric(30))
trainIndices = sample(1:dim(buzzbrews)[1],round(splitPerc * dim(buzzbrews)[1]))
train = buzzbrews[trainIndices,]
test = buzzbrews[-trainIndices,]
for(i in 1:numks)
  {
  classifications = knn(train[,c(7,8)],test[,c(7,8)],train$IPAAle, prob = TRUE, k = i)
  table(classifications,test$IPAAle)
  CM = confusionMatrix(table(classifications,test$IPAAle))
  masterAcc[j,i] = CM$overall[1]
  }
}

MeanAcc = colMeans(masterAcc)
# Visually find the best value of k by using it's location in the dataframe based on the highest Mean v
plot(seq(1,numks,1),MeanAcc, type = "l",
     col = "#c8201e",
     main = "Value for K Neighbors vs Accuracy",
     sub = "Budweiser Consultation",
     xlab = "Value of K Neighbors",
     ylab = "Accuracy Rate (Percentage)")
```




```r
# Locate the value of k based on the best MeanAcc in the dataframe
kvalue = match(max(MeanAcc), MeanAcc)
max(MeanAcc)
```

```
## [1] 0.6544813
```

```r
kvalue
```

```
## [1] 9
```

```r
####### Best value of k = 8 between 59% - 67% Accuracy ####################
####### Train the model using k = 8 ####################
classifications = knn(train[,c(7,8)],test[,c(7,8)],train$IPAAle, prob = TRUE, k = kvalue, use.all = TRUE
  table(classifications,test$IPAAle)
```

```
##
## classifications Ale IPA neither
##         Ale      67  12      49
##         IPA       9  64      55
##         neither 112  30     325
```

```r
  CM = confusionMatrix(table(classifications,test$IPAAle))
CM
```

```
## Confusion Matrix and Statistics
##
##
## classifications Ale IPA neither
##         Ale      67  12      49
```

```
##         IPA       9  64      55
##         neither 112  30     325
##
## Overall Statistics
##
##                Accuracy : 0.6307
##                  95% CI : (0.5944, 0.666)
##     No Information Rate : 0.5934
##     P-Value [Acc > NIR] : 0.02199
##
##                   Kappa : 0.3221
##
##  Mcnemar's Test P-Value : 4.24e-07
##
## Statistics by Class:
##
##                      Class: Ale Class: IPA Class: neither
## Sensitivity             0.35638    0.60377         0.7576
## Specificity             0.88598    0.89627         0.5170
## Pos Pred Value          0.52344    0.50000         0.6959
## Neg Pred Value          0.79664    0.92941         0.5938
## Prevalence              0.26003    0.14661         0.5934
## Detection Rate          0.09267    0.08852         0.4495
## Detection Prevalence    0.17704    0.17704         0.6459
## Balanced Accuracy       0.62118    0.75002         0.6373
```

```
####### Test the Classifier with some random data ###
classifyMyBeers <- data.frame(ABV = c(6,6,5,4,5, 12, 7),
       IBU = c(78, 65, 55, 38, 100, 148, 98))
classifications = knn(train[,c(7,8)],classifyMyBeers,train$IPAAle, prob = TRUE, k = kvalue)

classifications
```

```
## [1] neither Ale     Ale     neither IPA     neither IPA
## attr(,"prob")
## [1] 0.6666667 0.6666667 0.5000000 0.4444444 0.7777778 0.6666667 0.8181818
## Levels: Ale IPA neither
```

```
############ Test Results ######################
#Class: neither   Ale       Ale       neither   IPA       neither   IPA
#Prob:  0.6250000 0.6250000 0.6250000 0.7500000 0.7500000 0.5000000 0.7777778
##################################################


############# Summary data by classification #############
IPAAleSummary <- buzzbrews %>%
  group_by(IPAAle) %>%
  dplyr::summarise(ABV.min = min(ABV),
                   ABV.med = median(ABV),
                   ABV.max = max(ABV),
                   IBU.min = min(IBU),
                   IBU.med = median(IBU),
                   IBU.max = max(IBU))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

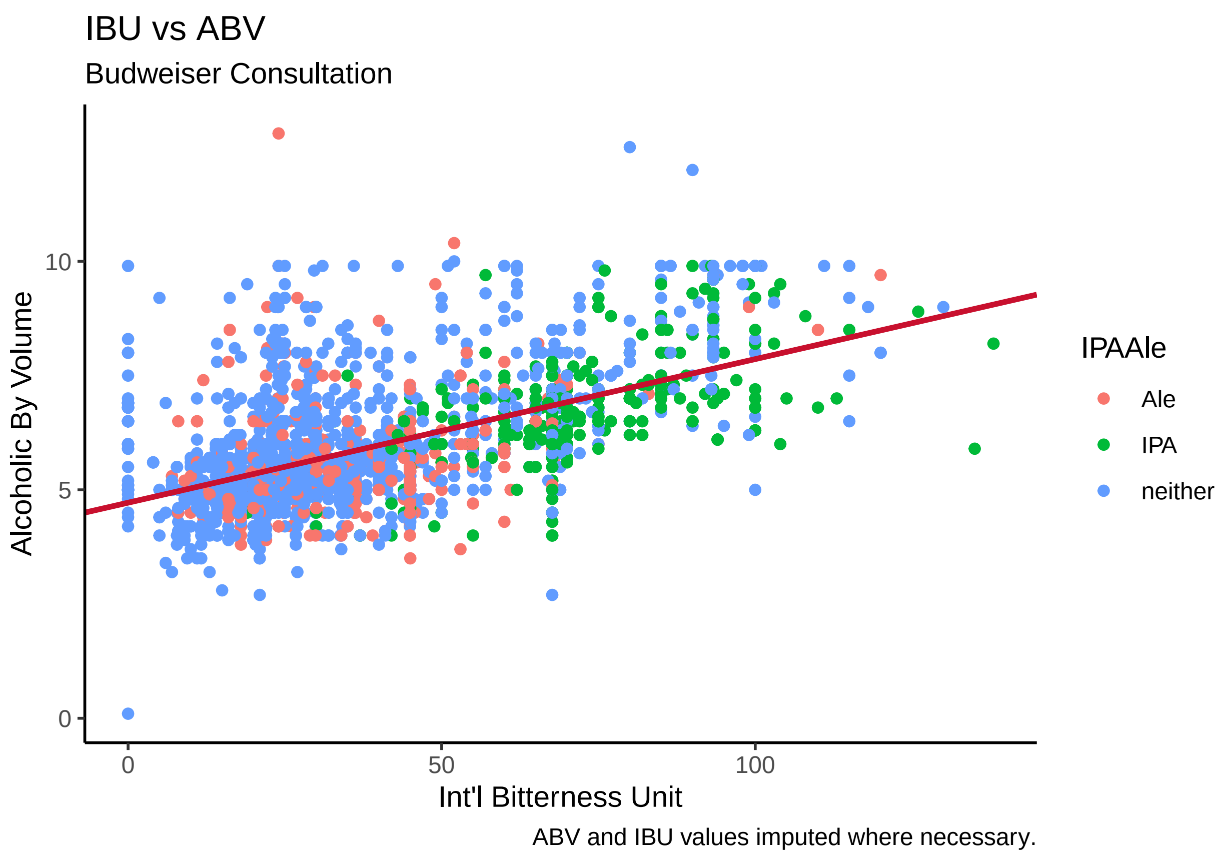
```r
IPAAleSummary
```

```
## # A tibble: 3 x 7
##   IPAAle  ABV.min ABV.med ABV.max IBU.min IBU.med IBU.max
##   <chr>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Ale         3.5     5.4    12.8       7    31       120
## 2 IPA         4       6.7     9.9      19    67.6     138
## 3 neither     0.1     5.5    12.5       0    28       130
```

```r
########################################################
##### Replot and color by beer style ##################
comparisonCoef <- coef(lm(ABV ~ IBU, buzzbrews))
comparisonCoef
```

```
## (Intercept)         IBU
##  4.71799073  0.03142639
```

```r
#  (Intercept)    MaxIBU
#   4.71799073  0.03142639

buzzbrews %>%
  ggplot(aes(x = IBU, y = ABV, color = IPAAle)) +
  geom_point(show.legend = TRUE, na.rm = TRUE) +
  geom_abline(intercept =  comparisonCoef[1] , slope = comparisonCoef[2], color = "#c8102E", size = 1) +
  theme_classic() +
  labs(title = "IBU vs ABV",
       subtitle = "Budweiser Consultation",
       y = "Alcoholic By Volume",
       x = "Int'l Bitterness Unit",
       caption="ABV and IBU values imputed where necessary.")
```




```
#####################
#                   #
#     Question 9    #
#                   #
#####################
##########################################################################################
#Knock their socks off!  Find one other useful inference from the data that you feel Budweiser may be a
##########################################################################################
```