Beer and Breweries Case Study

David Grijalva and Apurv Mittal

10/5/2020

Load Libraries

```
library(tm)
```

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

```
library(tidyr)
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
##     intersect, setdiff, setequal, union
```

```r
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.3.2     ✓ purrr   0.3.4
## ✓ tibble  3.0.3     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.5.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x ggplot2::annotate() masks NLP::annotate()
## x dplyr::arrange()    masks plyr::arrange()
## x purrr::compact()    masks plyr::compact()
## x dplyr::count()      masks plyr::count()
## x dplyr::failwith()   masks plyr::failwith()
## x dplyr::filter()     masks stats::filter()
## x dplyr::id()         masks plyr::id()
## x dplyr::lag()        masks stats::lag()
## x dplyr::mutate()     masks plyr::mutate()
## x dplyr::rename()     masks plyr::rename()
## x dplyr::summarise()  masks plyr::summarise()
## x dplyr::summarize()  masks plyr::summarize()
```

```r
library(caret)
```

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

```
##
## Attaching package: 'caret'
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

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

```r
library(class)
library(e1071)
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

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

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```r
library(gganimate)
```

```
## No renderer backend detected. gganimate will default to writing frames to separate files
## Consider installing:
## - the `gifski` package for gif output
## - the `av` package for video output
## and restarting the R session
```

```r
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
##    +.gg    ggplot2
```

```r
require(ggthemes)
```

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

Introduction

We have evaluated the data of the Breweries across USA and various different popular Beers with their Alcohol content (ABV) and Bitterness level (IBU). We did a thorough analysis of the data provided and came up with some interesting facts. We also have some recommendations following our data analysis provided towards the end of the presentation.

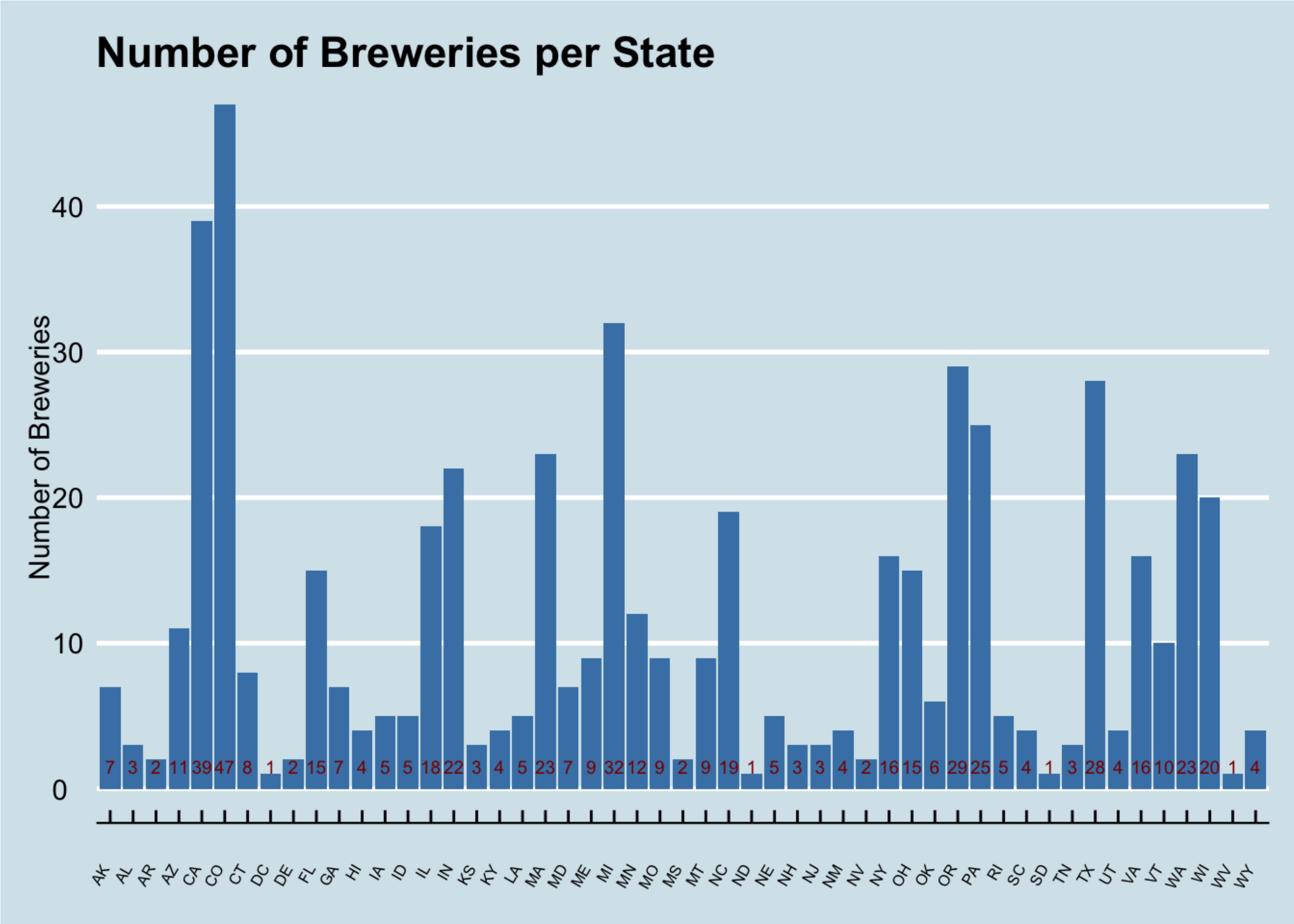
```r
# Load the CSVs
breweries = read.csv("./Data/Breweries.csv")
beers = read.csv("./Data/Beers.csv")
```

We analyzed the number of breweries across US. The distribution of breweries varies significantly across the country. Colorado and California are the states with most Breweries. While Delaware and West Virginia are among the states with lowest number of breweries.

Below Bar chart and Heat-map gives a good pictorial representation of the data.

```r
# Question 1
# Filter and plot the number of breweries in each state.
breweries_per_state = breweries %>% count(State)
breweries_per_state %>% ggplot() + geom_bar(aes(State, n), fill="steelblue" ,stat = 'identity')+ geom_text(stat =
 "count", aes(State, label = n, vjust=-0.1),size = 2.5, color = "darkred")+ labs(x='', y="Number of Breweries", t
itle='Number of Breweries per State') +theme_economist()+ theme(axis.text.x = element_text(angle = 60, hjust = 1,
size = 6))
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



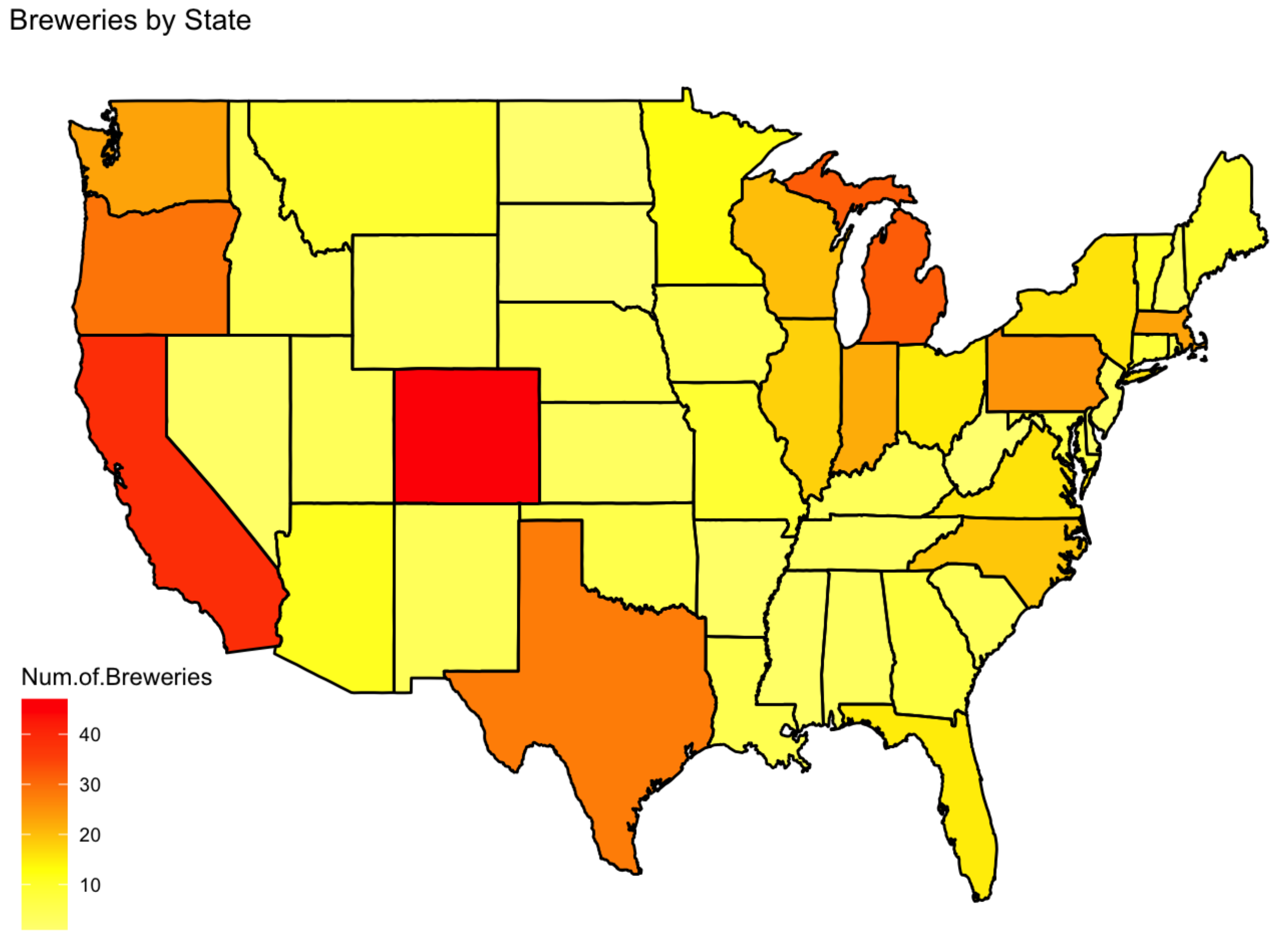
Below is the Heat-map with same data with different visualization.

```
# Building Heatmap of US for number of breweries in each state
breweries_per_state_1 = breweries_per_state

# Remove any whitespaces present in the dataset
breweries_per_state_1$State = trimws(breweries_per_state_1$State,which=c("both"),whitespace = "[ ]")

# Setup dataframe for State name lookup
lookup = data.frame(abbr = state.abb, State = state.name)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```r
# Merge the dataset
breweries_per_state_1 = merge(breweries_per_state_1, lookup, by.x="State", by.y="abbr")

# Change the state names to lowercase
breweries_per_state_1$StateLower = tolower(breweries_per_state_1$State.y)

states = map_data("state")

map.df = merge(states,breweries_per_state_1, by.x = "region", by.y ="StateLower",  all.x=T)
map.df = map.df[order(map.df$order),]

# Rename the column Name
colnames(map.df)[8] = "Num.of.Breweries"

# Plot the breweries data on US political map
ggplot(map.df, aes(x=long,y=lat,group=group))+
  geom_polygon(aes(fill=Num.of.Breweries))+
  geom_path()+
  scale_fill_gradientn(colours=rev(heat.colors(5)), na.value="grey90") + ggtitle("Breweries by State")+theme_map
()
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



Heat-map helps to easily identify which states are the ones with the most breweries. The two states with the most breweries are California and Colorado, with 39 and 47 breweries respectively.

Now, we will merge the two data sets. Snippet of merged data is provided below. This merge results in a data-frame of 2410 rows.

```
# Question 2 - Merge beer data with the breweries data
data = merge(x=breweries, y=beers, by.y ="Brewery_id" , by.x = "Brew_ID")
# Rename the variables
data = data%>%rename(Beer.Name=Name.y, Brewery = Name.x)
nrow(data)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

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

```
head(data,6)
```

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

```
tail(data,6)
```

```
##      Brew_ID                       Brewery          City State
## 2405     556          Ukiah Brewing Company         Ukiah    CA
## 2406     557        Butternuts Beer and Ale Garrattsville    NY
## 2407     557        Butternuts Beer and Ale Garrattsville    NY
## 2408     557        Butternuts Beer and Ale Garrattsville    NY
## 2409     557        Butternuts Beer and Ale Garrattsville    NY
## 2410     558 Sleeping Lady Brewing Company     Anchorage    AK
##                Beer.Name Beer_ID   ABV IBU                   Style Ounces
## 2405        Pilsner Ukiah      98 0.055  NA         German Pilsener     12
## 2406     Porkslap Pale Ale      49 0.043  NA American Pale Ale (APA)     12
## 2407       Snapperhead IPA      51 0.068  NA            American IPA     12
## 2408     Moo Thunder Stout      50 0.049  NA      Milk / Sweet Stout     12
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
## 2409  Heinnieweisse Weissebier      52 0.049  NA              Hefeweizen     12
## 2410 Urban Wilderness Pale Ale      30 0.049  NA        English Pale Ale     12
```

With the merged data, we first needs to check-out if there are any missing values. After data evaluation, we identified that there are only two variables with missing data. ABV is missing 62 values while IBU is missing 1005 rows.Since its a large number of missing values we need to identify a way to impute the missing values.

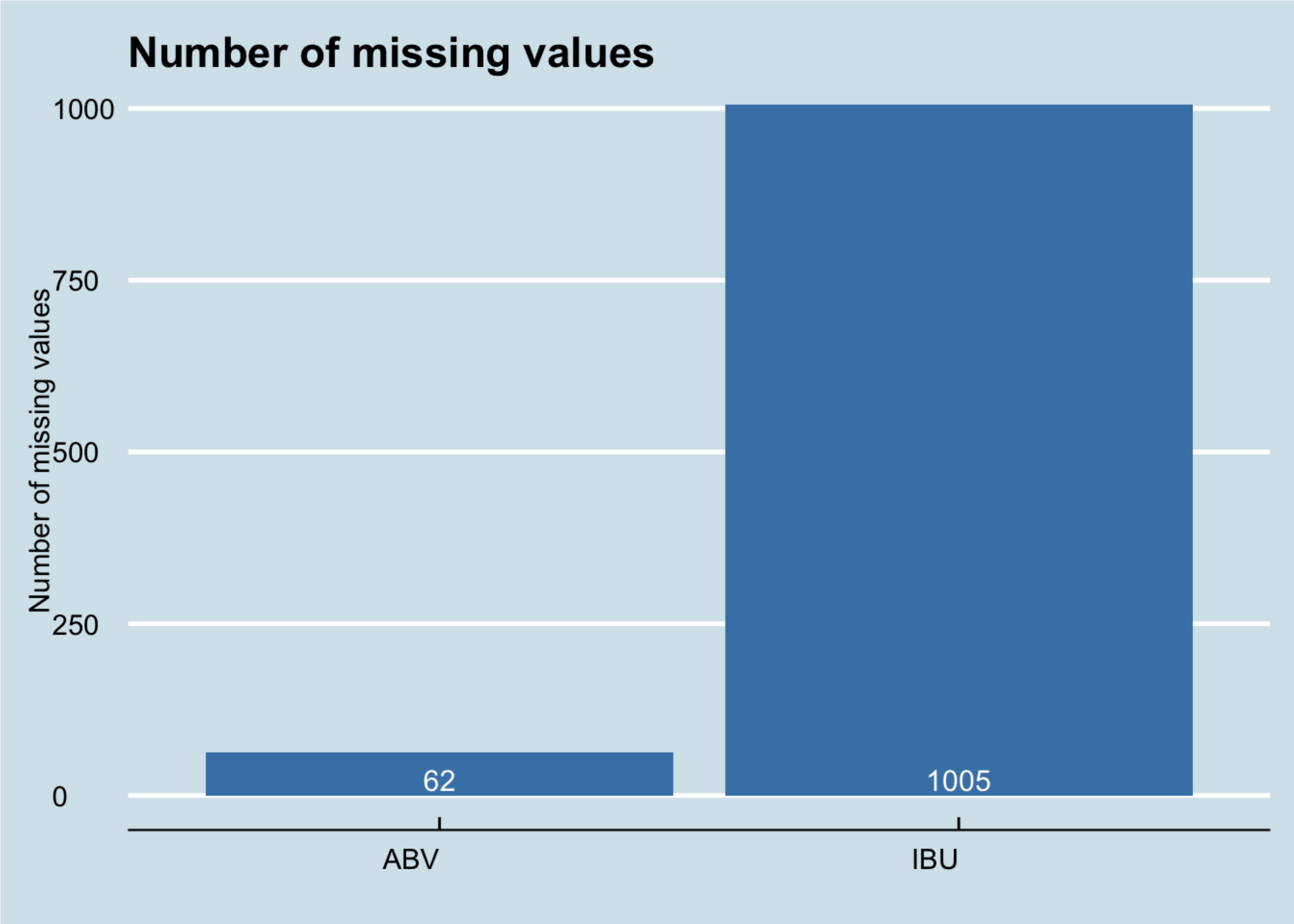
```
# Questions 3 - Handle missing values
missing.values <- data %>%
    gather(key = "key", value = "val") %>%
    mutate(is.missing = is.na(val)) %>%
    group_by(key, is.missing) %>%
    summarise(num.missing = n()) %>%
    filter(is.missing==T) %>%
    select(-is.missing) %>%
    arrange(desc(num.missing))
```

```
## Warning: attributes are not identical across measure variables;
## they will be dropped
```

```
## `summarise()` regrouping output by 'key' (override with `.groups` argument)
```

```
# Plot the missing values to identify the variables with missing data.
missing.values %>% ggplot() + geom_bar(aes(x=key, y=num.missing), fill="steelblue",stat = 'identity') + geom_text
(stat = "count", aes(key, label = num.missing, vjust=-0.2),size = 4, color = "white")+
   labs(x='', y="Number of missing values", title='Number of missing values') +theme_economist()+
   theme(axis.text.x = element_text(angle = 0, hjust = 1))
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



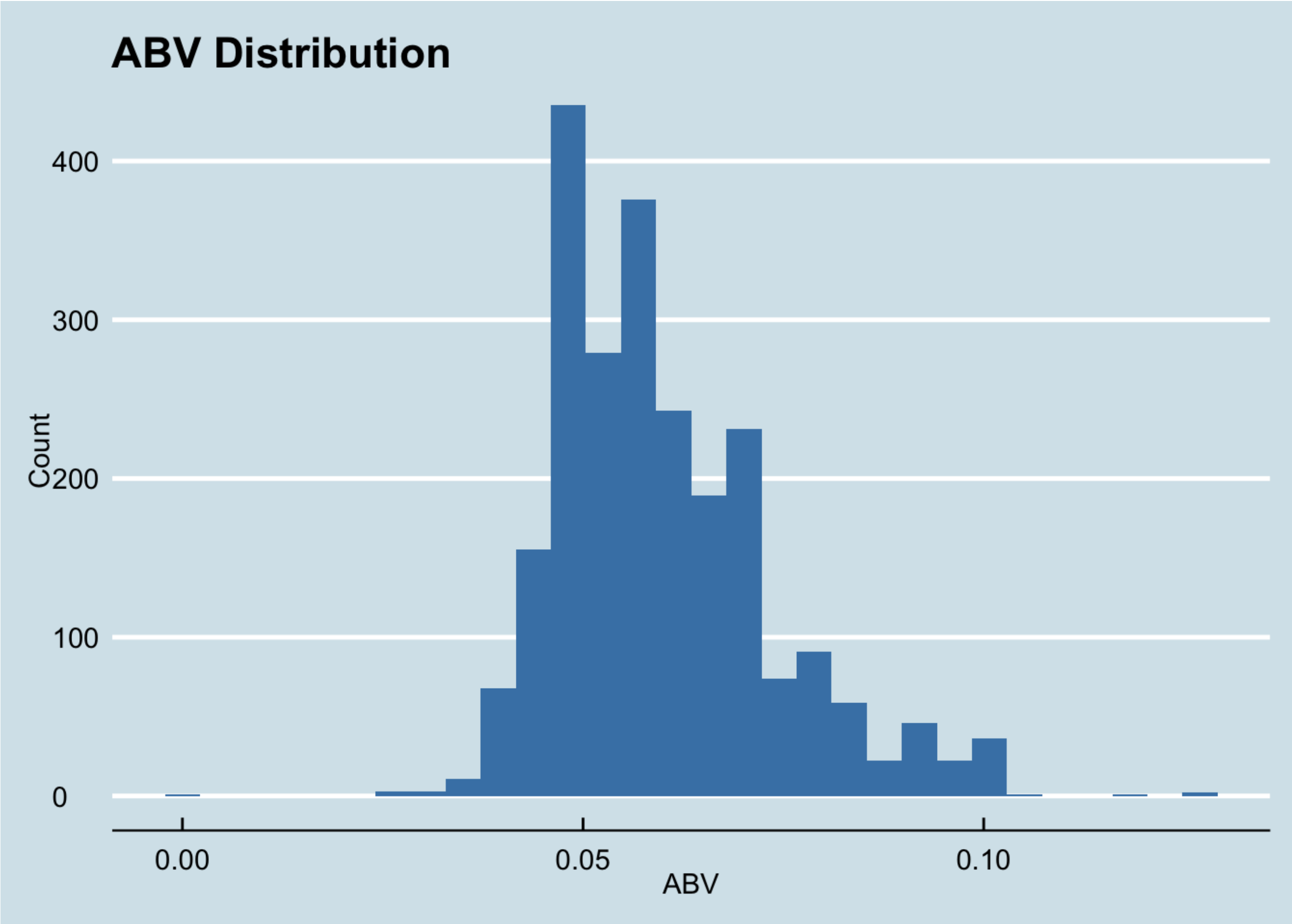
Next we plot IBU and ABV distribution.

```
# Questions 3 - Handle missing values

# Plot ABV distribution
data %>% ggplot() + geom_histogram(aes(x=ABV), fill="steelblue") +theme_economist()+
   labs(x="ABV", y="Count", title="ABV Distribution")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
## Warning: Removed 62 rows containing non-finite values (stat_bin).
```

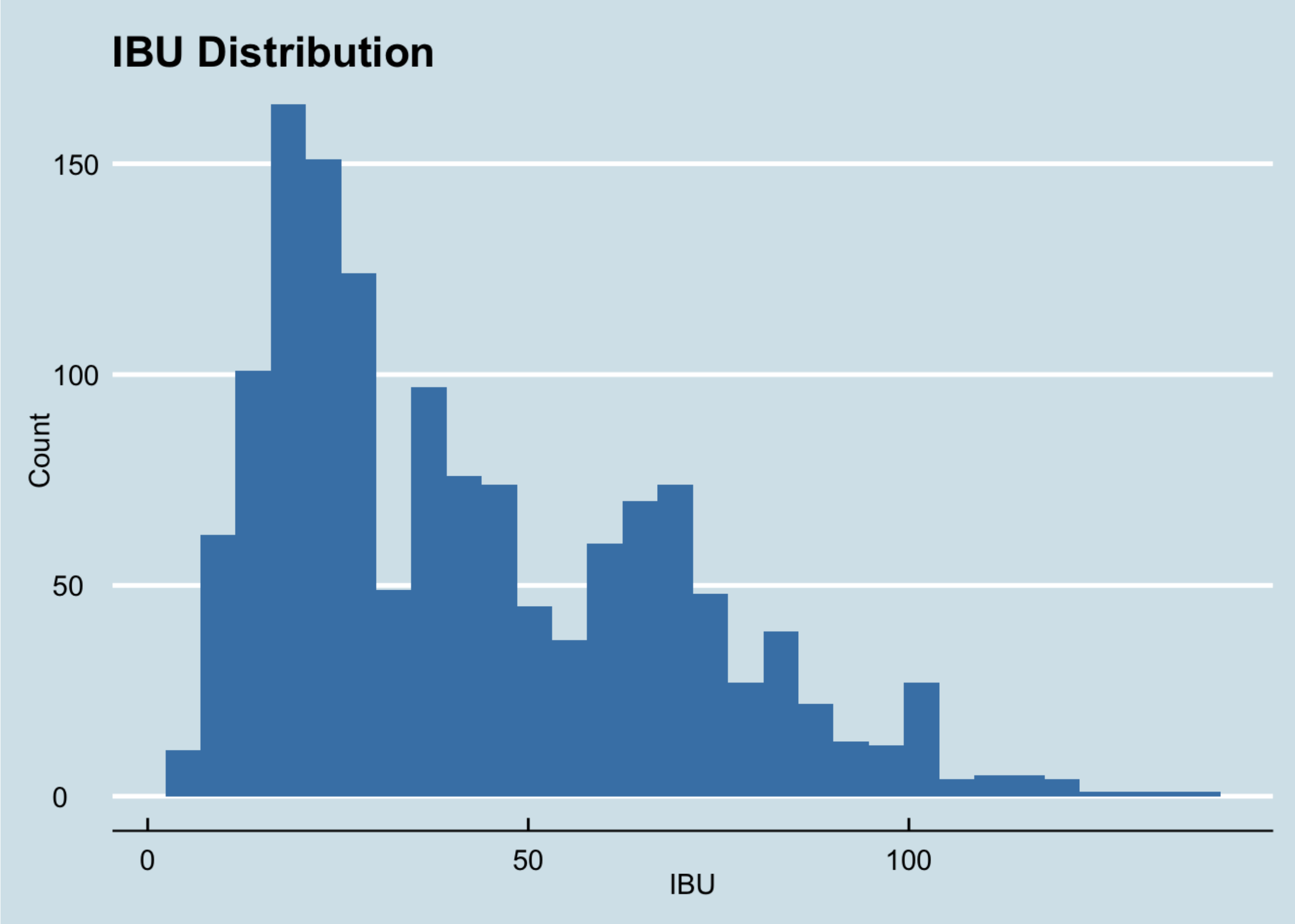



```
# Plot IBU distribution
data %>% ggplot() + geom_histogram(aes(x=IBU), fill="steelblue") +theme_economist()+
   labs(x="IBU", y="Count", title="IBU Distribution")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
## Warning: Removed 1005 rows containing non-finite values (stat_bin).
```



Quick visual inspection into the distribution of each variable, we notice that IBU is highly right skewed while ABV is slightly skewed.

Each type of beer style has its unique bitterness which might vary a bit from brand to brand but will still be in same ballpark for the beer style. There are 100 beer styles and more than 1,000 missing values we felt that the best approach was not to impute the missing IBU with the median IBU from all the known data for IBU, instead we decided to impute the missing values with medians for ABV, while for IBU we calculated the median for each beer style and imputed missing data with the median IBU value for that style.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
# Get rid of special characters in the beer styles
data$Style = gsub("[^0-9A-Za-z' ]"," " , data$Style ,ignore.case = TRUE)

#Deal with NA in IBU
# Finds the median value per beer style
meanIBU = matrix(nrow = 100)
styles = list()
for (i in 1:length(unique(data$Style)) )
{
  beer_style = unique(data$Style)[i]
  ibu_mean = mean(data[grep(beer_style, data$Style, ignore.case = T),]$IBU,na.rm = T )
  meanIBU[i] = ibu_mean
  styles[[i]] = beer_style
}

# Create a new styles dataframe with the IBU medians per beer style
styles_impute = data.frame(IBU=meanIBU, Style = matrix(unlist(styles), nrow=length(styles), byrow=T))

# merge the beer styles median IBU dataframe with the working dataframe on style name
impute_data = merge(data, styles_impute, by.x="Style", by.y="Style")

# If NA in original IBU value, then use median IBU per style, else use original value
impute_data = impute_data %>% mutate(imputed_IBU = ifelse(is.na(IBU.x) == TRUE,IBU.y,IBU.x))
# Impute any impute_data value with the median for the ABV and imputed_IBU columns
impute_data= impute_data %>% mutate_at(vars(ABV,imputed_IBU),~ifelse(is.na(.x), median(.x, na.rm = TRUE), .x))

# Get rid of the 5 rows without a beer style.
impute_data = impute_data%>% filter(!Style=="")

# Drop redundant columns
drops =  c("IBU.x","IBU.y")
impute_data = impute_data[ , !(names(impute_data) %in% drops)]
```

There were also 5 beers with a missing style. We decided to drop those records.

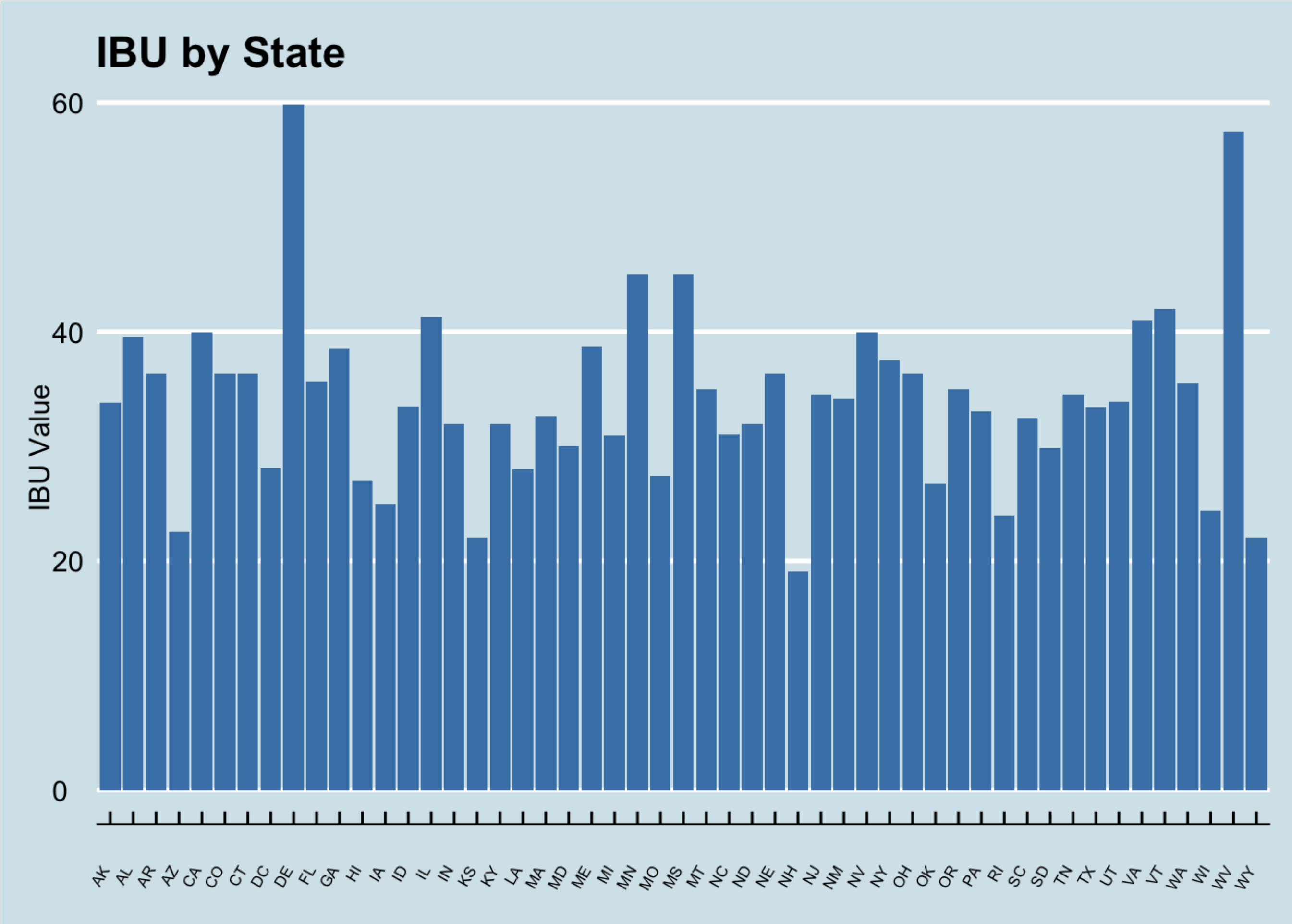
Next, we computed the median Alcohol content (ABV) and median Bitterness (IBU) fir each state.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

Median IBU by state:

```
# Question 4 - Compute the median alcohol content and international bitterness unit for each state. Plot a bar ch
art to compare.

# Calculate Median IBU

median_ibu_state = aggregate(impute_data[, 10], list(impute_data$State), median)

# Plot median IBU by state
median_ibu_state %>% ggplot() + geom_bar(aes(Group.1, x),fill="steelblue", stat = 'identity') +
    labs(x='', y="IBU Value", title='IBU by State') + theme_economist()+
  theme(axis.text.x = element_text(angle = 60, hjust = 1, size=6))
```

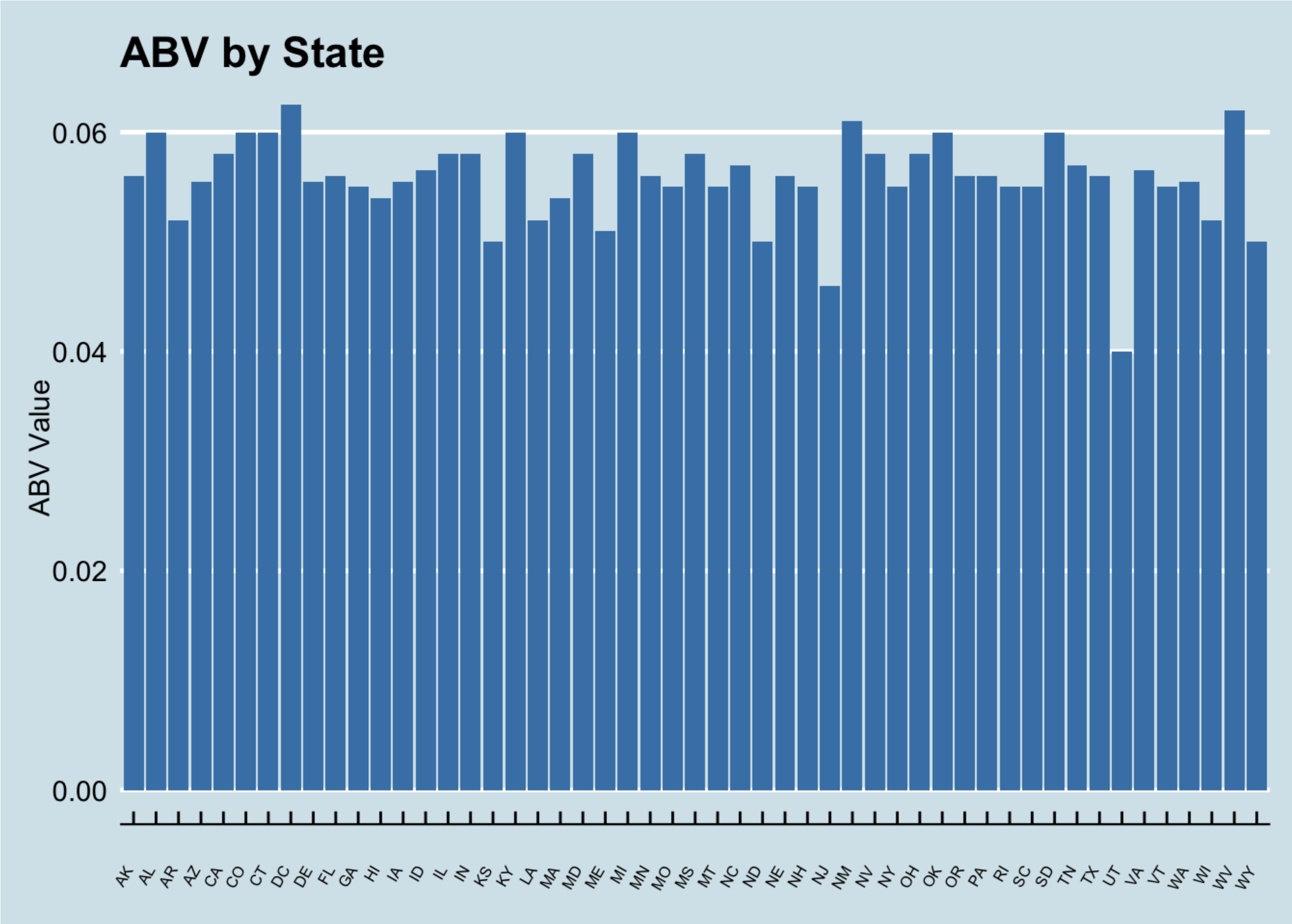
Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



Median ABV by State:

```
# Calculate Median ABV
median_abv_state = aggregate(impute_data[, 8], list(impute_data$State), median)

# Plot Median ABV by State
median_abv_state %>% ggplot() + geom_bar(aes(Group.1, x), fill="steelblue", stat = 'identity') +
    labs(x='', y="ABV Value", title='ABV by State') + theme_economist()+
  theme(axis.text.x = element_text(angle = 60, hjust = 1, size = 6 ))
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



Delaware and West Virginia are by far leading on the IBU and New Hampshire among the lowest.

In terms of alcohol content West Virginia is leading again which gives an impression there could be a relationship between the IBU and ABV.

Next we identified the states with a Beer with highest Alcohol content (ABV) and Beer with most bitterness.

```
# Question 5 - Which state has the maximum alcoholic (ABV) beer?

impute_data %>% filter(ABV==max(ABV))
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
##              Style Brew_ID                 Brewery    City State
## 1 Quadrupel  Quad       52 Upslope Brewing Company Boulder    CO
##                                              Beer.Name Beer_ID   ABV Ounces
## 1 Lee Hill Series Vol. 5 - Belgian Style Quadrupel Ale    2565 0.128   19.2
##   imputed_IBU
## 1          24
```

The state with the maximum ABV in a beer is CO. The beer is the Lee Hill Series Vol. 5 - Belgian Style Quadruple Ale with an ABV of 0.128.

```
# Question 5 -Which state has the most bitter (IBU) beer?

impute_data %>% filter(imputed_IBU==max(imputed_IBU))
```

```
##                            Style Brew_ID                 Brewery    City State
## 1 American Double   Imperial IPA     375 Astoria Brewing Company Astoria    OR
##                     Beer.Name Beer_ID   ABV Ounces imputed_IBU
## 1 Bitter Bitch Imperial IPA     980 0.082     12         138
```

The state with the maximum IBU in a beer is OR. The beer is the Bitter Bitch Imperial IPA with an IBU of 138.

Next, we calculated the mean, max and median of ABV across the states. Also, checked the distribution,

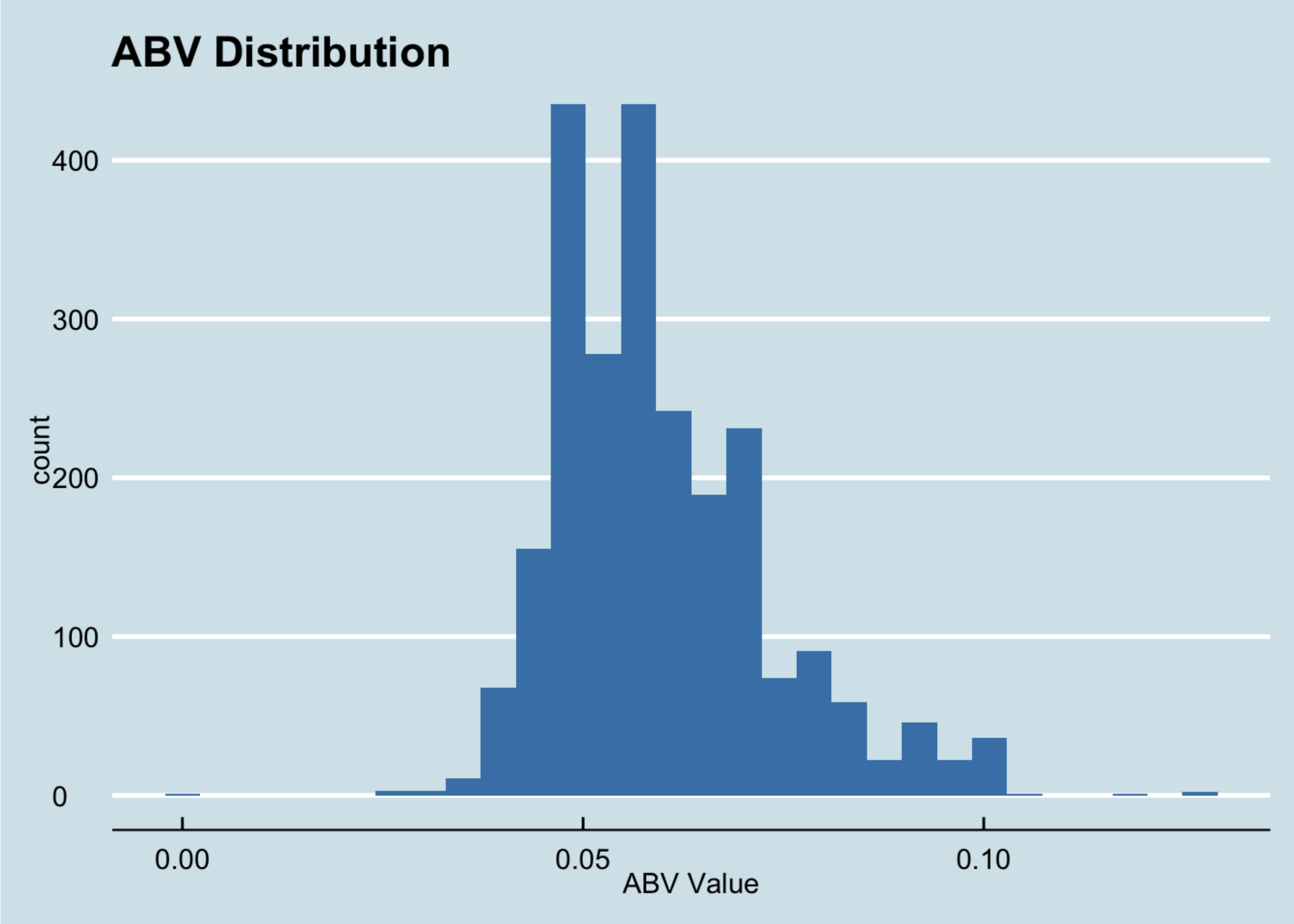
```
# Question 6 - Comment on the summary statistics and distribution of the ABV variable.
# Calculate Summary
summary(impute_data$ABV)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00100 0.05000 0.05600 0.05968 0.06700 0.12800
```

```
# Histogram for Distribution
impute_data %>% ggplot() + geom_histogram(aes(ABV), fill="steelblue") +
  labs(x='ABV Value', y="count", title='ABV Distribution') + theme_economist()
```

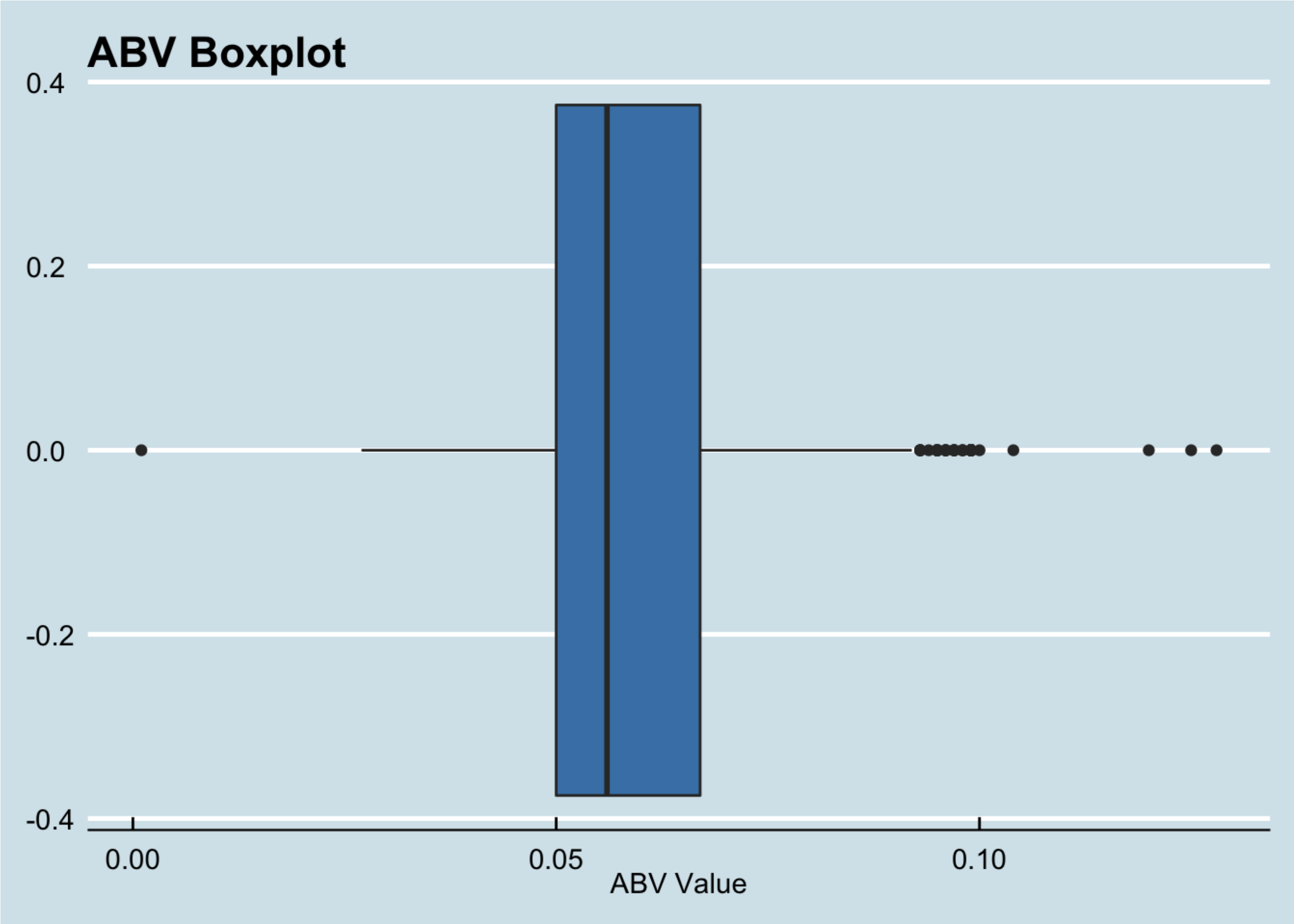
Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Boxplot for distribution
impute_data %>% ggplot() + geom_boxplot(aes(ABV), fill="steelblue") + theme_economist()+
  labs(x='ABV Value', title='ABV Boxplot')
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

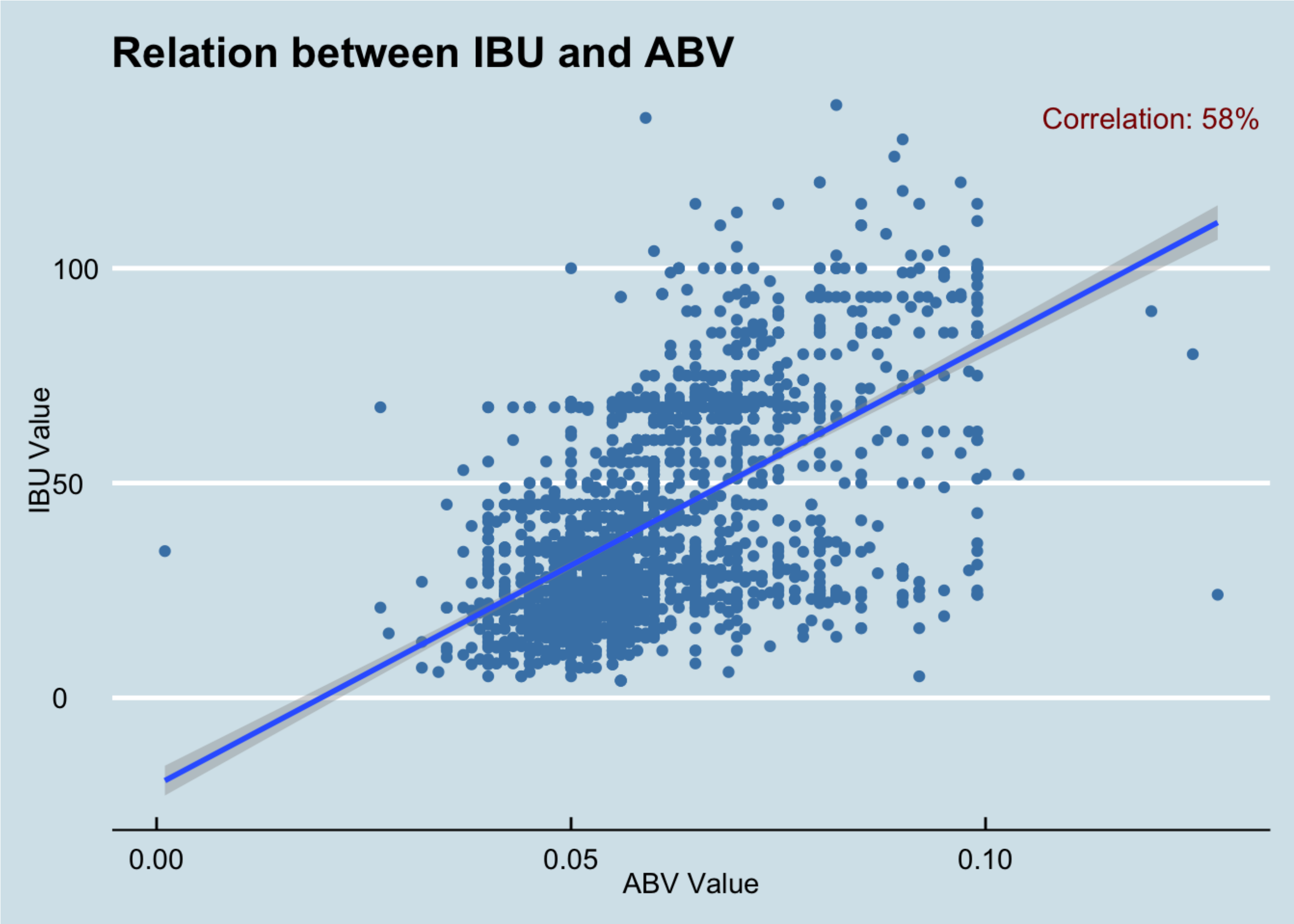


It appears that the distribution of the ABV variable is slightly right skewed. There appears to be outliers particularly on the left side as ABV is almost zero.
Min: 0.10% , Median: 5.60% , Mean: 5.96% , Max: 12.80%

```
# 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.

impute_data %>% ggplot() + geom_point(aes(x=ABV, y=imputed_IBU), color="steelblue", size = 1.5) + geom_smooth(aes(x=ABV, y=imputed_IBU),method = "lm")+ labs(x='ABV Value', y="IBU Value", title="Relation between IBU and ABV") + theme_economist() + annotate("text",x=0.12, y=135, label="Correlation: 58%", color="darkred", size=4)
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
## `geom_smooth()` using formula 'y ~ x'
```




```
cor(impute_data$ABV, impute_data$imputed_IBU, method = "pearson")
```

```
## [1] 0.5802225
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
lm1<-lm(ABV~imputed_IBU, data = impute_data)
summary(lm1)
```

```
##
## Call:
## lm(formula = ABV ~ imputed_IBU, data = impute_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.056510 -0.006384 -0.002144  0.004060  0.073830
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.627e-02  4.438e-04  104.25   <2e-16 ***
## imputed_IBU 3.291e-04  9.423e-06   34.92   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01091 on 2403 degrees of freedom
## Multiple R-squared:  0.3367, Adjusted R-squared:  0.3364
## F-statistic:  1220 on 1 and 2403 DF,  p-value: < 2.2e-16
```

From looking at the scatter-plot above it seems that there is a positive linear relation between ABV and IBU variables. As the ABV the increases the IBU is expected to increase as well. Correlation coefficient of 58.02% explains the variability in IBU based on the changes in ABV. This suggest some evidence that the more alcohol content in the beer the bitter it will be which can be associated to the fact that more bitterness requires breweries to add sweetness to the beer to balance the taste and additional sugar leads to higher alcohol. Which makes it apparent that increase in IBU leads to increase in ABV and vice versa.

```
# 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 one way or the other. You can of course assume your audience is comfortable with percentages … KNN is very easy to understand conceptually.

# Create new ipa_ale column based on regex from the beer style column
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

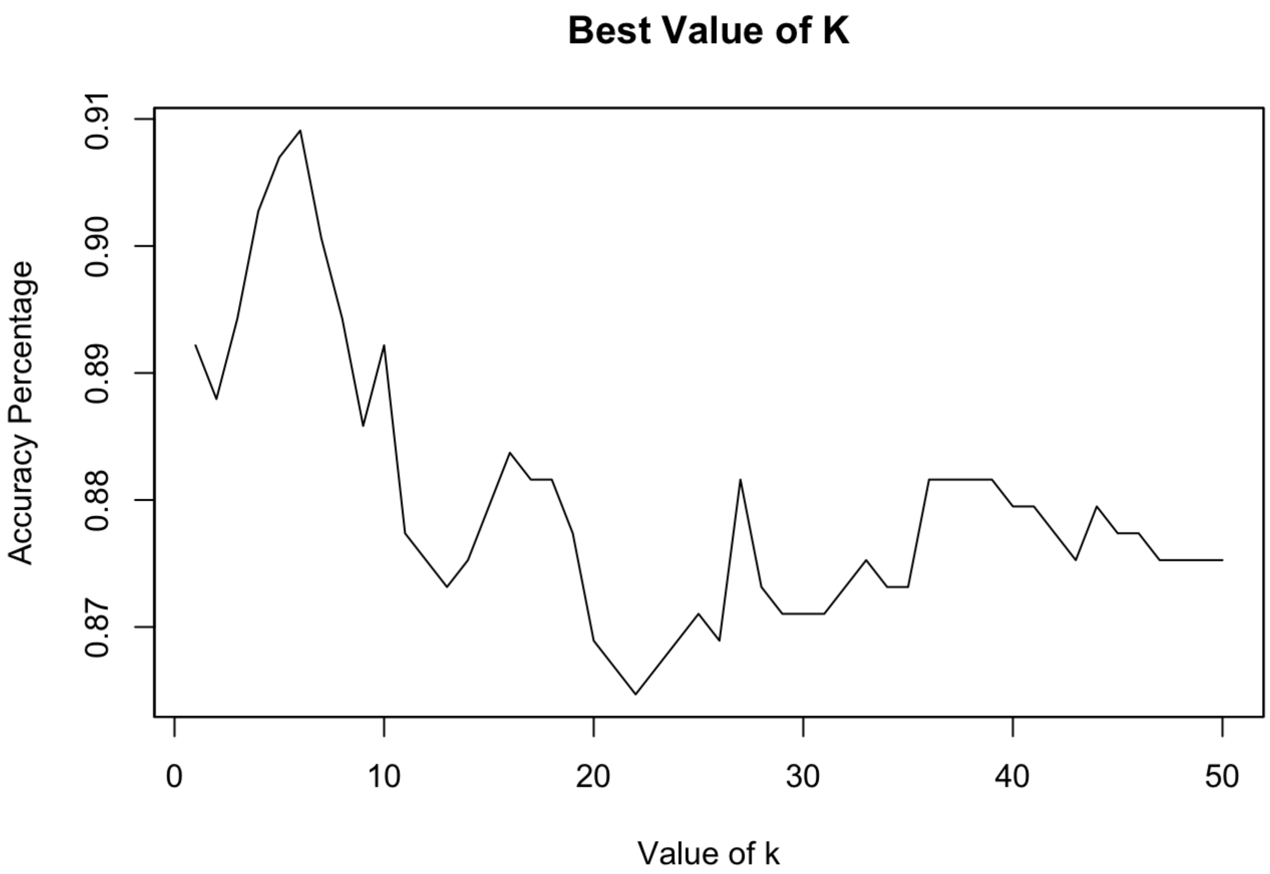
```
impute_data$ipa_ale = ifelse(grepl("ipa", impute_data$Style, ignore.case = T), "ipa",
         ifelse(grepl("ale", impute_data$Style, ignore.case = T), "ale", "Other"))
# Filter out other type
ale_ipa = impute_data %>% filter(!ipa_ale=="Other")
```

To assess the relation between IBU and ABV between IPA and Ales we will first need to create a variable with classifies the beers between “Ale”, “IPA”, and “other”. We will then filter out the “other” variable from the data set. This will result in a data set containing only “ALE” and “IPA” labels.

```
#Choose the best K
set.seed(12)
splitPerc = .70

# Split the dataset into train and test
trainIndices = sample(1:dim(ale_ipa)[1],round(splitPerc * dim(ale_ipa)[1]))
train = ale_ipa[trainIndices,]
test = ale_ipa[-trainIndices,]

# Run iterations to find the best K
iterations = 50
accs = data.frame(accuracy = numeric(iterations), k = numeric(iterations))

for(i in 1:iterations)
{

  classification = knn(train[,c(8,10)], test[,c(8,10)],train$ipa_ale,k=i)
  cm = confusionMatrix(table(test$ipa_ale, classification ), positive="ale")


  accs$accuracy[i] = cm$overall[1]
  accs$k[i] = i
}

# Plot the K values with accuracy variation

plot(accs$k,accs$accuracy, type = "l", xlab = "Value of k", ylab = "Accuracy Percentage", main = "Best Value of
 K")
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
axis(side = 2, at = c(0:50, 5))
box()
```



We ran 50 iterations of the K-NN (K nearest neighbors) classier to choose the K with the highest accuracy classifying between “ALE” and “IPA”. It appears that the best value K with highest accuracy is 5. We will use K = 5 to run different train/test splits.

The Accuracy, specificity and sensitivity measures are quite high for K=5. Accuracy is at 90%.Specificity and Sensitivity at 87.0% and 92.0% respectively.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
# Run 1000 iterations  on different train/test sets. We will compute the average accuracy, specificity and Sensit
ivity.
iterations = 1000
masterAcc = matrix(nrow = iterations)
masterSensitivity = matrix(nrow = iterations)
masterSpecificity = matrix(nrow = iterations)
splitPerc = .7 #Training / Test split Percentage
for(j in 1:iterations)
{
  splitPerc = .70
  set.seed(j*49+15)
  trainIndices = sample(1:dim(ale_ipa)[1],round(splitPerc * dim(ale_ipa)[1]))
  train = ale_ipa[trainIndices,]
  test = ale_ipa[-trainIndices,]

  classification = knn(train[,c(8,10)], test[,c(8,10)],train$ipa_ale,k=5)
  cm = confusionMatrix(table(test$ipa_ale, classification ), positive="ale")


  masterAcc[j] = cm$overall[1]
  masterSpecificity[j] = cm$byClass[2]
  masterSensitivity[j] = cm$byClass[1]

}


MeanAcc = colMeans(masterAcc)
MeanSpecificity = colMeans(masterSpecificity)
MeanSensitivity = colMeans(masterSensitivity)

MeanAcc
```

```
## [1] 0.9048203
```

```
MeanSpecificity
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

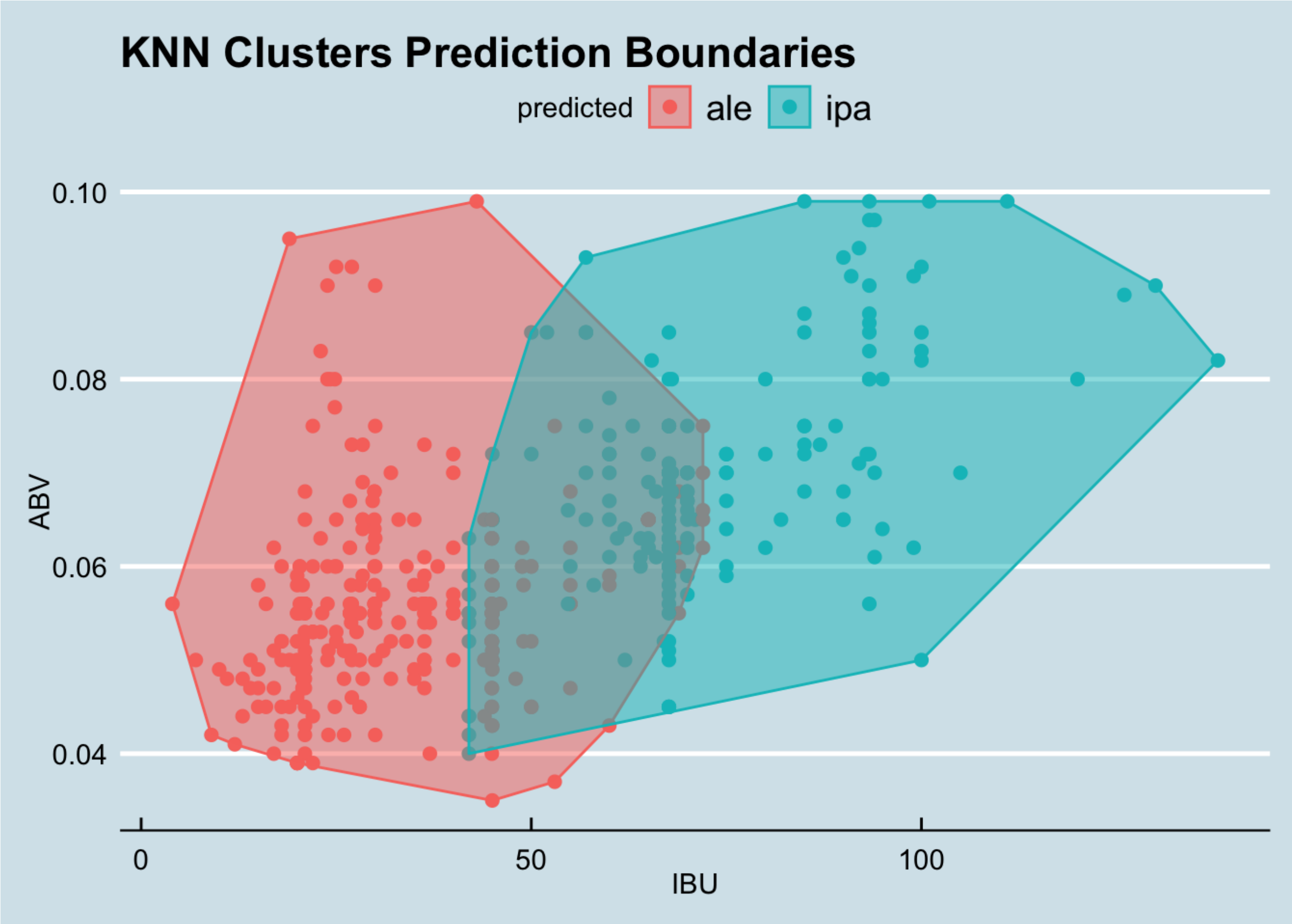
```
## [1] 0.8699283
```

```
MeanSensitivity
```

```
## [1] 0.9247805
```

```
splitPerc = .70
set.seed(j*49+15)
trainIndices = sample(1:dim(ale_ipa)[1],round(splitPerc * dim(ale_ipa)[1]))
train = ale_ipa[trainIndices,]
test = ale_ipa[-trainIndices,]




# Do knn
fit = knn(train[,c(8,10)], test[,c(8,10)],train$ipa_ale,k=5)

# Create a dataframe to simplify charting
plot.df = data.frame(test, predicted = fit)
plot.df$ipa_ale = as.factor(plot.df$ipa_ale)


# First use Convex hull to determine boundary points of each cluster
plot.df1 = data.frame(x = plot.df$imputed_IBU,
                      y = plot.df$ABV,
                      predicted = plot.df$predicted)

find_hull = function(df) df[chull(df$x, df$y), ]
boundary = ddply(plot.df1, .variables = "predicted", .fun = find_hull)

ggplot(plot.df, aes(imputed_IBU, ABV, color = predicted, fill = predicted)) +
  geom_point(size = 2) +   geom_polygon(data = boundary, aes(x,y), alpha = 0.5) + ggtitle("KNN Clusters Predictio
n Boundaries") + theme_economist() + xlab("IBU")
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD




```
# plot source: https://stackoverflow.com/questions/35402850/how-to-plot-knn-clusters-boundaries-in-r
```

From the above plots its evident that ABV and IBU are correlated and varies significantly for IPA and ALEs. We can see a clear trend that the higher value of IBU is associated to IPAs while smaller values of IBU associated to ALEs. There is middle ground where IPAs and ALEs both overlap for the same level of IBUs and ABV but that area is comparatively small. There is a clear distinction between ALE and IPAs based on the IBU and ABV values.

Since, we know that IPAs and ALEs are clearly different and have different properties. We took our analysis to the next step. We checked the most popular words among the Beer Styles and among Beer Names.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
# Question 9 - Knock their socks off!  Find one other useful inference from the data that you feel Budweiser may
 be able to find value in.  You must convince them why it is important and back up your conviction with appropria
te statistical evidence.

library(wordcloud)
```

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

```
#install.packages("RColorBrewer")
library(RColorBrewer)
#install.packages("wordcloud2")
library(wordcloud2)
#install.packages("tm")
library(tm)

# Set Beer Style as Vector
text = as.vector(impute_data['Style'])
docs = Corpus(VectorSource(text))

# Remove punctuations, whitespaces and numbers
docs = docs %>%
  tm_map(removeNumbers) %>%
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace)
```

```
## Warning in tm_map.SimpleCorpus(., removeNumbers): transformation drops documents
```

```
## Warning in tm_map.SimpleCorpus(., removePunctuation): transformation drops
## documents
```

```
## Warning in tm_map.SimpleCorpus(., stripWhitespace): transformation drops
## documents
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```r
# Move to lower case
docs = tm_map(docs, content_transformer(tolower))
```

```
## Warning in tm_map.SimpleCorpus(docs, content_transformer(tolower)):
## transformation drops documents
```

```r
# Ignore Stop Words
docs = tm_map(docs, removeWords, stopwords("english"))
```

```
## Warning in tm_map.SimpleCorpus(docs, removeWords, stopwords("english")):
## transformation drops documents
```

```r
tdm = TermDocumentMatrix(docs)
matrix = as.matrix(tdm)
words = sort(rowSums(matrix),decreasing=TRUE)

# Make a dataframe
df_style = data.frame(word = names(words),freq=words)

# Set Beer Name as Vector
text = as.vector(impute_data$Beer.Name)
docs = Corpus(VectorSource(text))

# Remove punctuations, whitespaces and numbers
docs = docs %>%
  tm_map(removeNumbers) %>%
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace)
```

```
## Warning in tm_map.SimpleCorpus(., removeNumbers): transformation drops documents
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
## Warning in tm_map.SimpleCorpus(., removePunctuation): transformation drops
## documents
```

```
## Warning in tm_map.SimpleCorpus(., stripWhitespace): transformation drops
## documents
```

```
docs = tm_map(docs, content_transformer(tolower))
```

```
## Warning in tm_map.SimpleCorpus(docs, content_transformer(tolower)):
## transformation drops documents
```

```
# Ignore Stop Words
docs = tm_map(docs, removeWords, stopwords("english"))
```

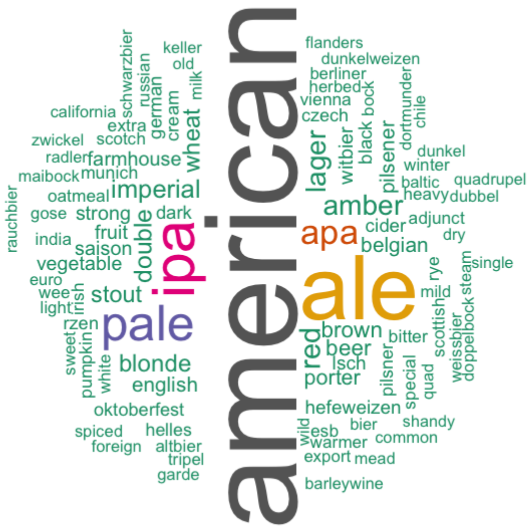
```
## Warning in tm_map.SimpleCorpus(docs, removeWords, stopwords("english")):
## transformation drops documents
```

```
tdm = TermDocumentMatrix(docs)

matrix = as.matrix(tdm)
words = sort(rowSums(matrix),decreasing=TRUE)

# Make a dataframe
df_name = data.frame(word = names(words),freq=words)
```

Word Cloud - Beer Styles

```
# Create the Word cloud of Beer Style
wordcloud(words = df_style$word, freq = df_style$freq, min.freq = 1,max.words=100, random.order=FALSE, rot.per=0.
35,colors=brewer.pal(8, "Dark2"))
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



Word Cloud - Beer Names

```
# Create the Word cloud of Beer Style
set.seed(1234) # for reproducibility
wordcloud(words = df_name$word, freq = df_name$freq, min.freq = 1,max.words=100, random.order=FALSE, rot.per=0.35
,colors=brewer.pal(6, "Dark2"))
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

We notice the most popular words are American, IPA and ALE.

Now we will run another test to check if the IPA and ALE have different Mean for IBUs and ABV.

```
ale_ipa$ipa_ale = as.factor(ale_ipa$ipa_ale)
t.test(ale_ipa$imputed_IBU ~ale_ipa$ipa_ale)
```

```
##
##  Welch Two Sample t-test
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
##
## data:  ale_ipa$imputed_IBU by ale_ipa$ipa_ale
## t = -42.582, df = 1095.1, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -39.37772 -35.90862
## sample estimates:
## mean in group ale mean in group ipa
##          33.72304          71.36620
```

```
t.test(ale_ipa$ABV ~ale_ipa$ipa_ale)
```

```
##
##  Welch Two Sample t-test
##
## data:  ale_ipa$ABV by ale_ipa$ipa_ale
## t = -19.143, df = 1070.2, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01317336 -0.01072383
## sample estimates:
## mean in group ale mean in group ipa
##        0.05659782        0.06854641
```

Two sample t-test confirms that the mean of IBU and ABV is different for ALE from IPA. This confirms our earlier inference from the KNN test.

Powered with this information, we tried to focus on the top 5 states in the US in terms of consumption of beer.

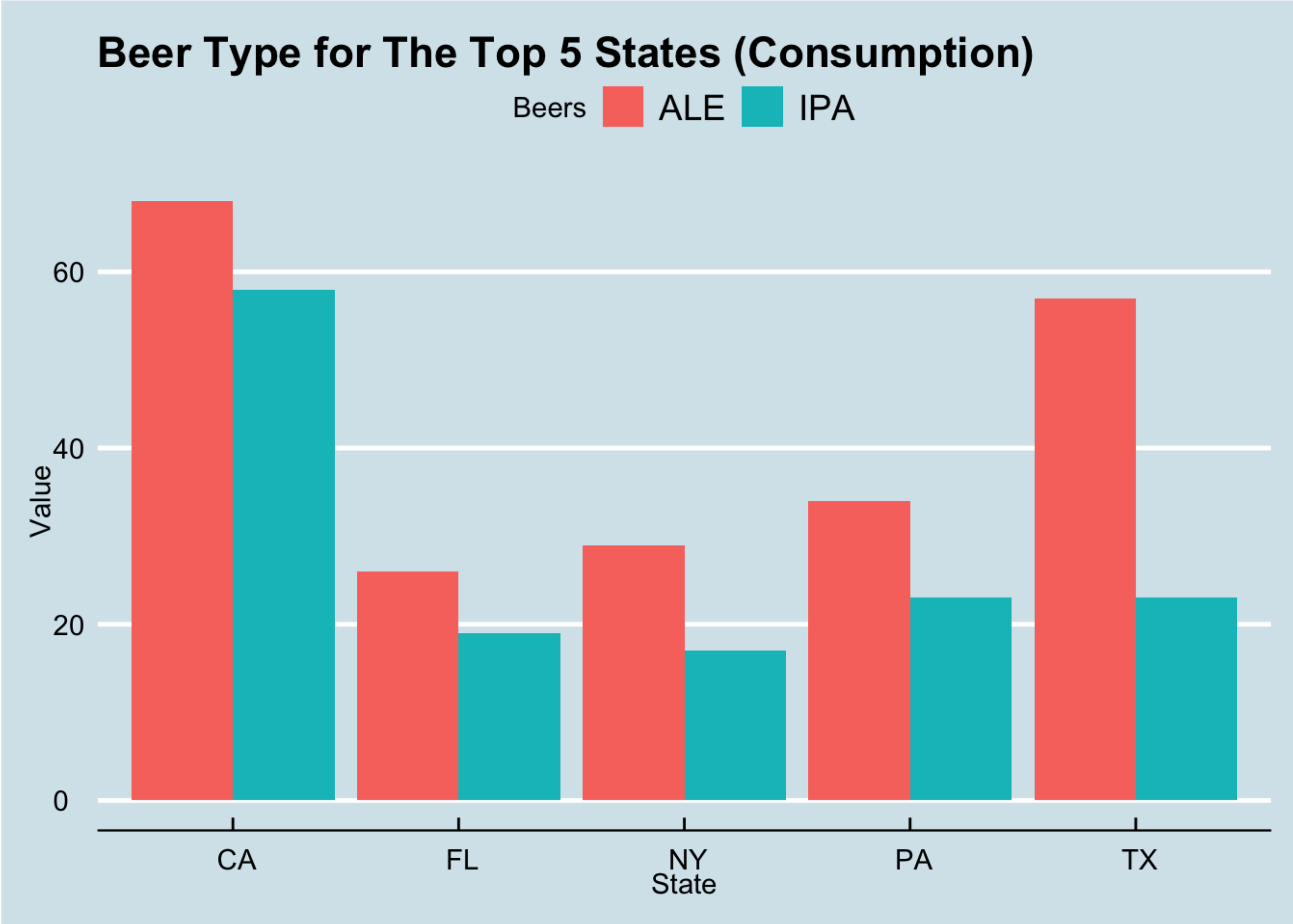
The top 5 states in terms of beer consumption are California, Texas, Florida, New York and Pennsylvania. referring to the report published at https://vinepair.com/articles/map-states-drink-beer-america-2020/

```
# Seperate ALE
ale = impute_data %>% filter(ipa_ale=="ale")

# Seperate IPA
ipa = impute_data %>% filter(ipa_ale=="ipa")
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD

```
# Step 1 -  group by state and count the number of beer in each state IPA and ALE
ipa_state =ipa %>% count(State)
ale_state =ale %>% count(State)
ale_state  = ale_state %>% rename(ALE = n)
ipa_state  = ipa_state %>% rename(IPA = n)

# Merge the Data Frame
beers_state = merge(ale_state, ipa_state, by="State")


# Step 2-  Filter for the top 5 states for Beer Consumption
beers_state_top_5= beers_state %>% filter(State == " CA" | State== " TX" | State== " FL" | State== " NY" | State=
= " PA")

# Step 3-  Plot # of beers that are ale or ipa per state
beers_state_tall = beers_state_top_5 %>% gather(key= Beers, value=Value, ALE:IPA)
beers_state_tall %>% ggplot(aes(State, Value, fill=Beers), ) + geom_col(position="dodge") + theme_economist() + g
gtitle("Beer Type for The Top 5 States (Consumption)")
```

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD



Based on the US census report. Texas is adding more population every year than any other state in the USA.
https://www.census.gov/newsroom/press-releases/2019/popest-nation.html

In terms of Beer consumption Texas is at number 2 (as mentioned above). Considering the growth in population and the beer consumption in Texas. We recommend to launch new beer(s) in the state of Texas.

Considering there is a huge demand for IPA and Texas has lot less IPAs compared to ALEs as shown the plot above. Since American, IPAs are most popular beer styles, we recommend American Pale Ale (APA) or Indian Pale Ale(IPA) for Texas market.

Create PDF in your applications with the Pdfcrowd HTML to PDF API PDFCROWD