Final_Project

Henry Lewis

22 December, 2021

The dataset chosen for this assignment is called Craft Beers dataset. Description: This dataset contains a list of 2,410 US craft beers and 510 US breweries and its available at https://www.kaggle.com/nickhould/craft-cans (https://www.kaggle.com/nickhould/craft-cans). It contains information on craft beers and is divided across two datasets, beer and breweries and are linked together by the brewery id. The beers dataset details the attributes of the beers in 2017, such as the bothe size(in ounces), the bitterness (ibu),alcohol content(abv), beer name and styles of beers in 2017, as well as information on their bottle size.

```
# Import Libraries
library("dplyr")
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

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

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

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Part 1: Analysis

Beer Dataset

```
# Read the beers csv to a variable beer with first row containing column names and removing w
hite space
beer <- read.csv("beers.csv", header = TRUE, strip.white=TRUE)
# set the column name to beer_id
colnames(beer)[4] <- c("beer_id")
# set the column name to beer_name
colnames(beer)[5] <- c("beer_name")
# Remove the index column
beer_df <- beer[,-1]
```

Breweries Dataset

```
# Read the breweries csv to a variable breweries with first row containing column names and r
emoving white space
breweries <- read.csv("breweries.csv", header = TRUE, strip.white=TRUE)
# set the column name to brewery_name
colnames(breweries)[2] <- c("brewery_name")
# set the column id to brewery_id
colnames(breweries)[1] <- c("brewery_id")
```

Merged dataset

```
# merge the beer and breweries and remove the first column
df <- merge(beer_df, breweries, by.x = "brewery_id")
df <- df[,-1]
head(df)
```

```
##     abv ibu beer_id     beer_name                                style ounces
## 1 0.045  50    2692  Get Together                          American IPA     16
## 2 0.049  26    2691 Maggie's Leap                    Milk / Sweet Stout     16
## 3 0.048  19    2690    Wall's End                     English Brown Ale     16
## 4 0.060  38    2689       Pumpion                           Pumpkin Ale     16
## 5 0.060  25    2688    Stronghold                        American Porter     16
## 6 0.056  47    2687   Parapet ESB Extra Special / Strong Bitter (ESB)     16
##        brewery_name        city state
## 1 NorthGate Brewing Minneapolis    MN
## 2 NorthGate Brewing Minneapolis    MN
## 3 NorthGate Brewing Minneapolis    MN
## 4 NorthGate Brewing Minneapolis    MN
## 5 NorthGate Brewing Minneapolis    MN
## 6 NorthGate Brewing Minneapolis    MN
```

Breweries Analysis

```
# remove the first column of the dataframe
breweries_df <- breweries[,-1]
# first 6 rows display
head(breweries_df)
```

```
##                brewery_name          city state
## 1         NorthGate Brewing   Minneapolis    MN
## 2 Against the Grain Brewery    Louisville    KY
## 3  Jack's Abby Craft Lagers    Framingham    MA
## 4 Mike Hess Brewing Company     San Diego    CA
## 5   Fort Point Beer Company San Francisco    CA
## 6     COAST Brewing Company    Charleston    SC
```

The breweries dataframe contains 558 observations and 3 columns that include the brewery name, city location, and state within the United States where the brewery is located.

```
# Structure of teh Breweries dataset
str(breweries_df)
```

```
## 'data.frame':    558 obs. of  3 variables:
##  $ brewery_name: chr  "NorthGate Brewing" "Against the Grain Brewery" "Jack's Abby Craft L
agers" "Mike Hess Brewing Company" ...
##  $ city        : chr  "Minneapolis" "Louisville" "Framingham" "San Diego" ...
##  $ state       : chr  "MN" "KY" "MA" "CA" ...
```

Analyse the number of Breweries

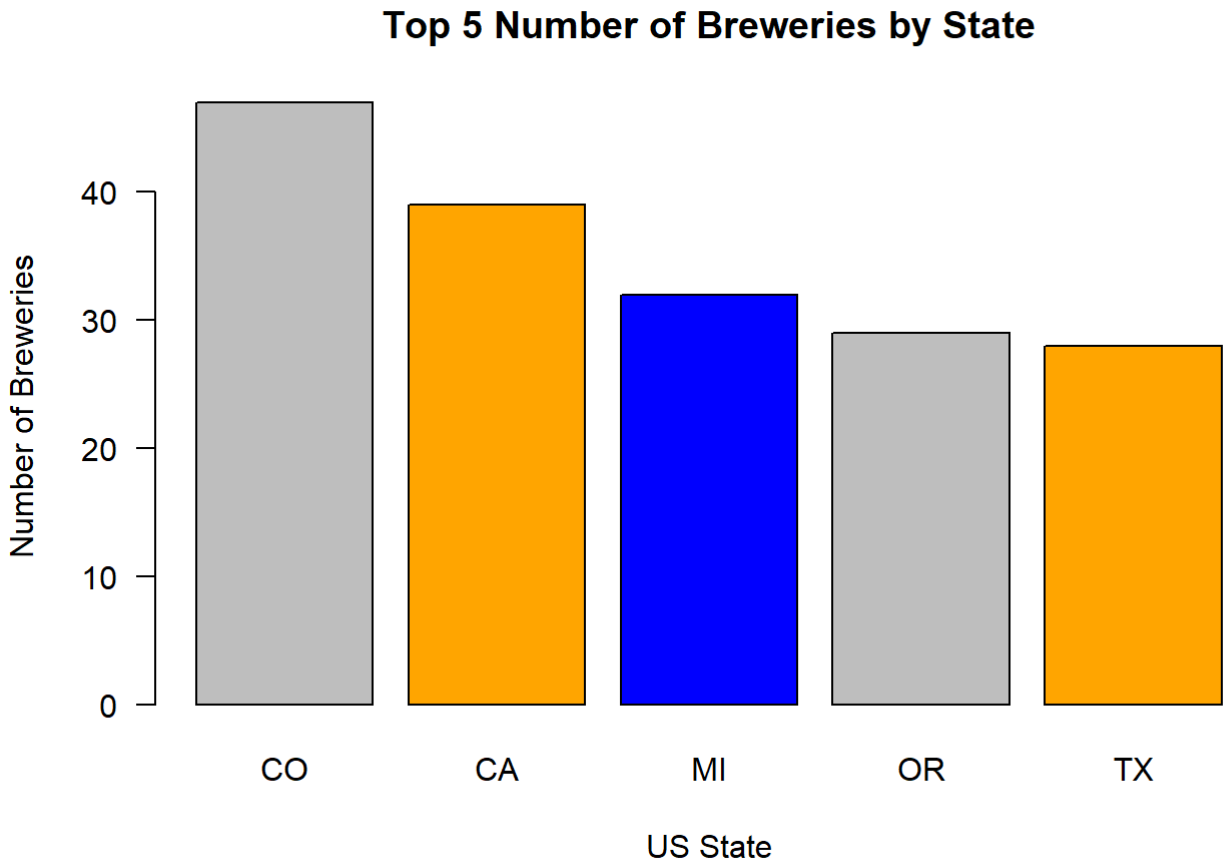
- The number of breweries is displayed per state

```
#Number of breweries per state
state_breweries <- table(breweries_df$state)
state_breweries
```

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

Displaying the top 5 number of breweries per state

```
max5_state_breweries <-head(sort(state_breweries,decreasing = TRUE),5)
colors = c("gray", "orange", "blue")
barplot(max5_state_breweries, main = "Top 5 Number of Breweries by State", xlab ="US State",
 ylab = "Number of Breweries",
col= colors, las = 1)
```

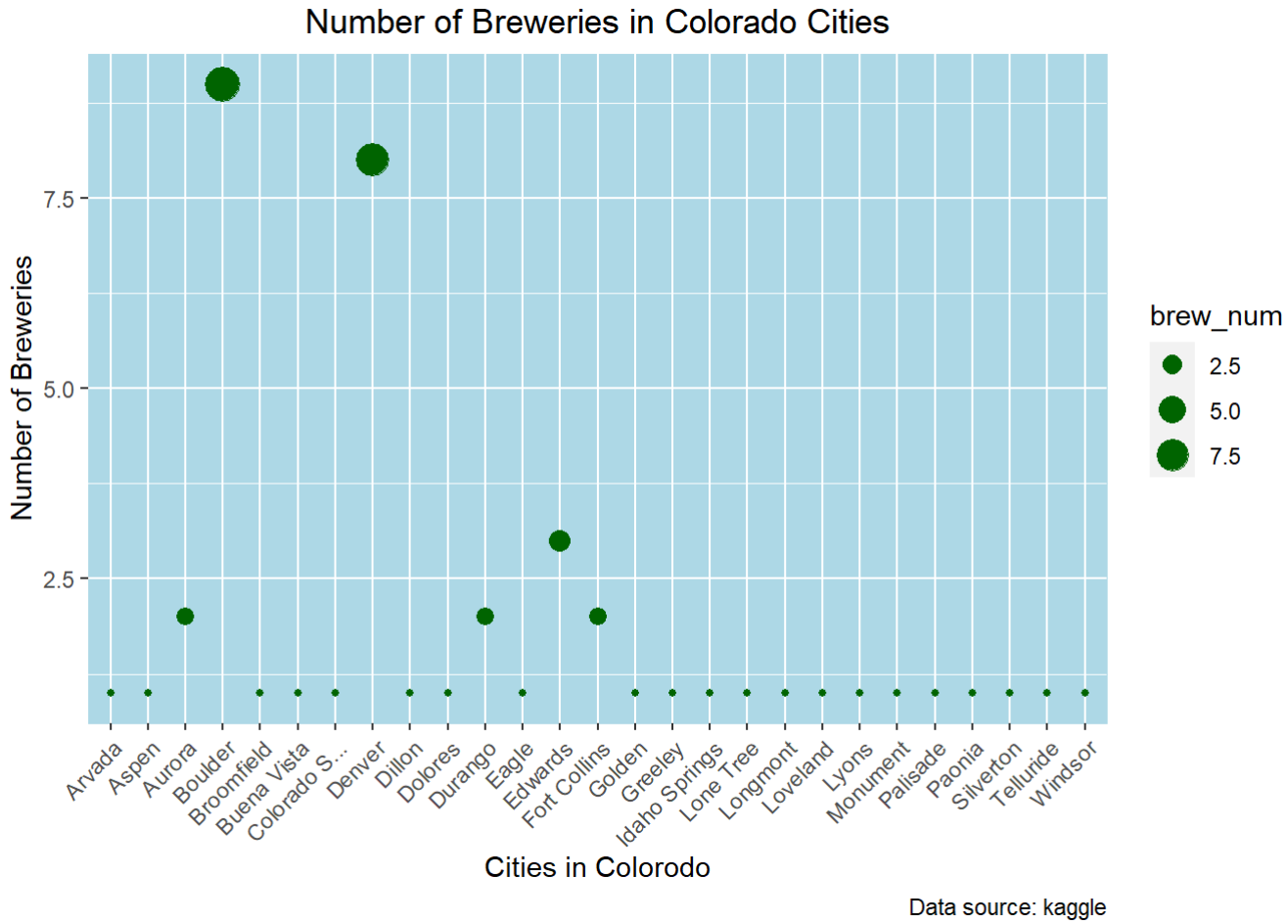


As can be seen Colorado CO has the largest quantity of breweries with 47. Then comes California with 39, Michigan with 32, Oregon with 29 and Texas with 28.

```
# get all the breweries in Colorado state
colorado_brew <- breweries_df[which(breweries_df$state == "CO"),]

#colorado_breweries <- colorado_brew[1]
#nrow(colorado_breweries)

# take only the brewery name and the city its located in
colorado_brew_cities <- colorado_brew[1:2]
# group the dataframe by the cities and get the number of breweries per city
brewery_cities <- colorado_brew_cities %>%
  group_by(city) %>% summarize(n())
# convert it to a dataframe from a tibble
brewery_cities <- as.data.frame(brewery_cities)
# change the names to be more readable
colnames(brewery_cities)[1] <- c("city")
colnames(brewery_cities)[2] <- c("brew_num")

# Plot the number of breweries per city
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
ggplot(brewery_cities, aes(x=city, y=brew_num, size = brew_num)) +
    geom_point(color = "darkgreen")+
  # angles the labels
  theme(axis.text.x = element_text(angle = 45, hjust=1))+
  # truncates labels
  scale_x_discrete(label = function(x) stringr::str_trunc(x, 13)) +
  ggtitle("Number of Breweries in Colorado Cities") +
  # Adds a theme and adds a centered title
  theme(plot.title = element_text(hjust = 0.5))+
  # Adds detailed caption information alongside a theme
  labs(caption = "Data source: kaggle",
              x = "Cities in Colorodo", y = "Number of Breweries") + theme(
              panel.background = element_rect(fill = "lightblue",
                    colour = "lightblue",
                    size = 0.5, linetype = "solid"),
                    panel.grid.major = element_line(size = 0.5, linetype = 'solid',
                    colour = "white"),
              panel.grid.minor = element_line(size = 0.25, linetype = 'solid',
                    colour = "white")
    )
```



Analyse the Beers

```
# first 5 lines of the beers dataset
head(beer_df)
```

```
##     abv ibu beer_id           beer_name                          style
## 1 0.050  NA    1436            Pub Beer             American Pale Lager
## 2 0.066  NA    2265         Devil's Cup        American Pale Ale (APA)
## 3 0.071  NA    2264 Rise of the Phoenix                   American IPA
## 4 0.090  NA    2263            Sinister American Double / Imperial IPA
## 5 0.075  NA    2262       Sex and Candy                   American IPA
## 6 0.077  NA    2261        Black Exodus                  Oatmeal Stout
##   brewery_id ounces
## 1        408     12
## 2        177     12
## 3        177     12
## 4        177     12
## 5        177     12
## 6        177     12
```

The total number of beers contained within the dataset is 2410.

```
# function for calculating the number of N/A values
beer_missing <- sapply(beer_df, function(x)sum(is.na(x)))
```

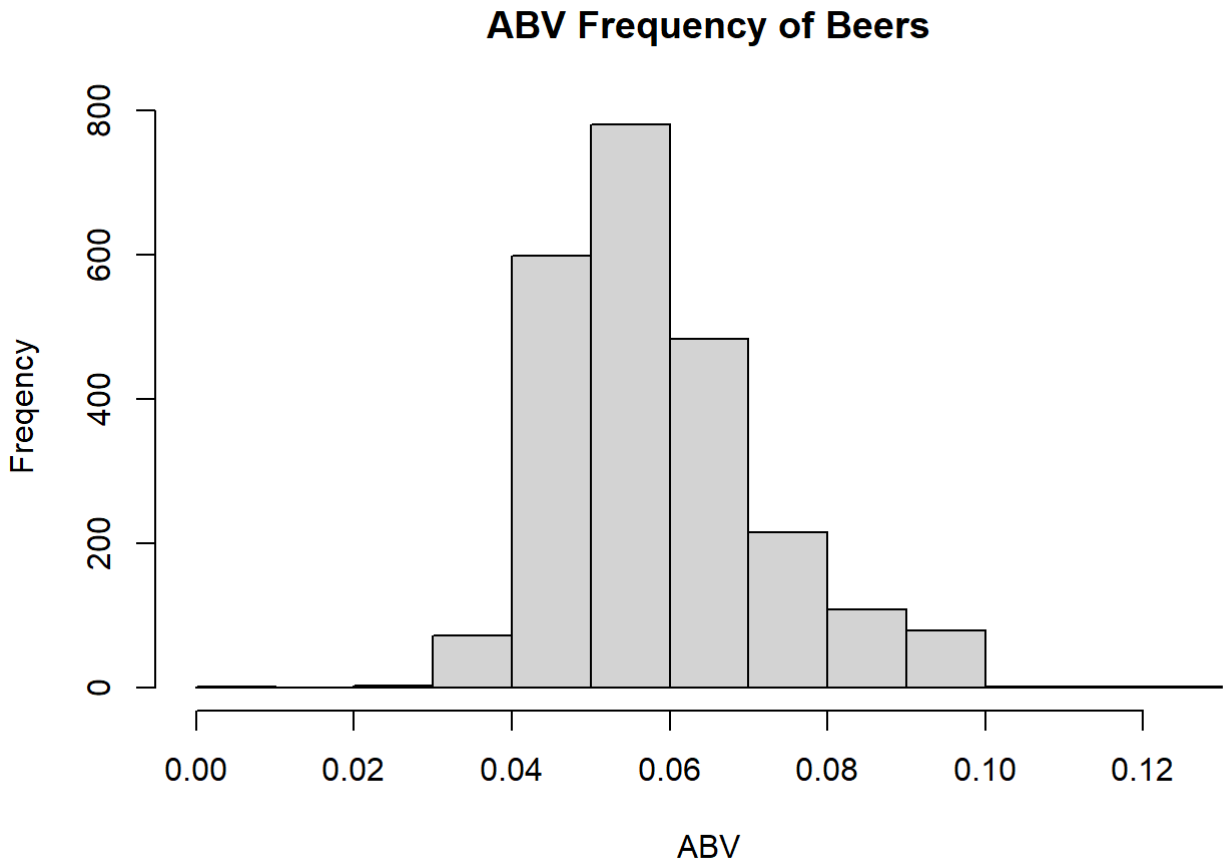
The total number of missing values contained within the ABV (Alcohol By Volume) column is 62 representing 2.5726141% of the dataset, whilst the number of missing values contained within IBU (International Bitterness Units) is 1005 representing 41.7012448 % of the dataset. The remaining columns have no missing values.

```
beer_averages <- sapply(beer_df[1:2], function(x) mean(x, na.rm = TRUE))
beer_averages
```

```
##         abv         ibu
##  0.05977342 42.71316726
```

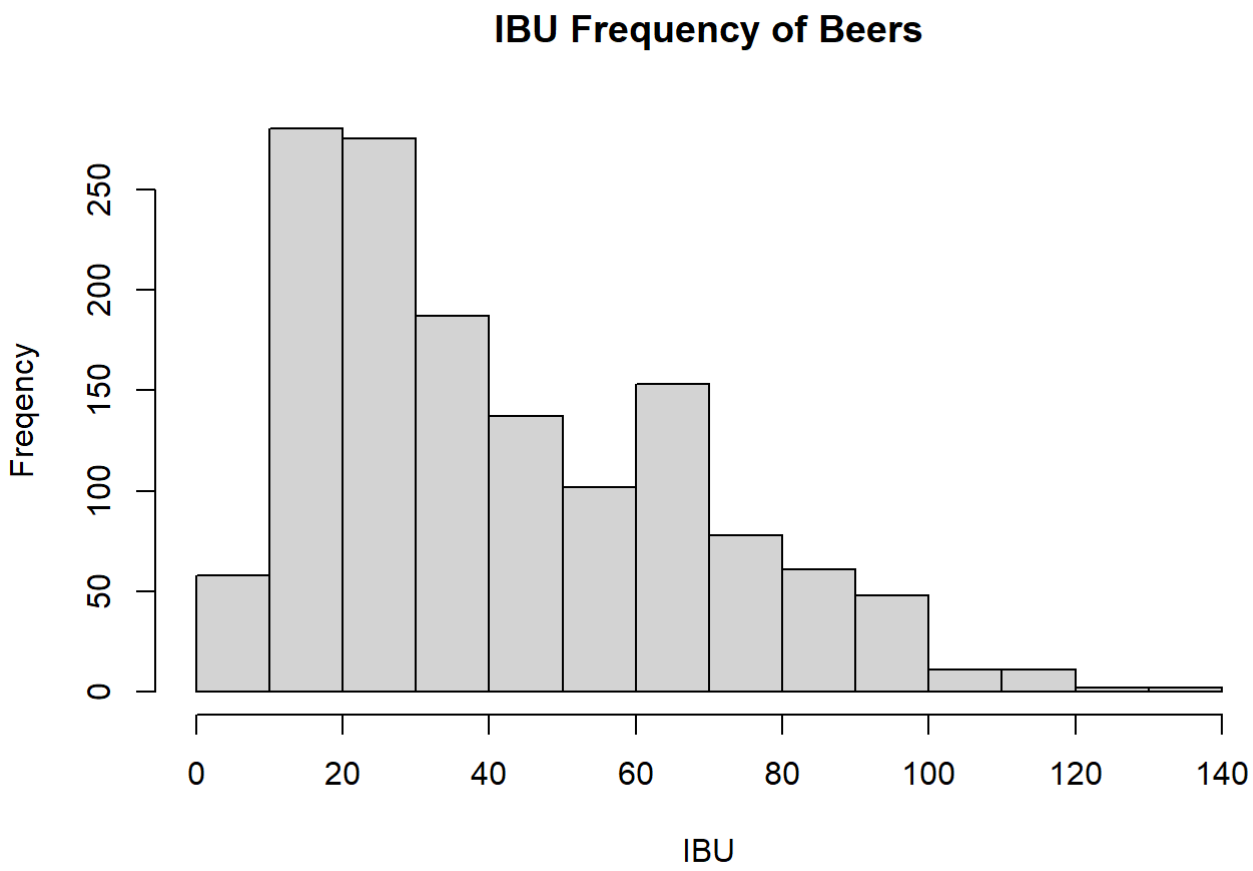
The average ABV of the beers within the dataset is 5.9773424% and the average IBU 42.7131673

```
# plot the most frequent ABV
hist(beer_df$abv,
     main="ABV Frequency of Beers",
      ylab="Freqency",
      xlab="ABV")
```



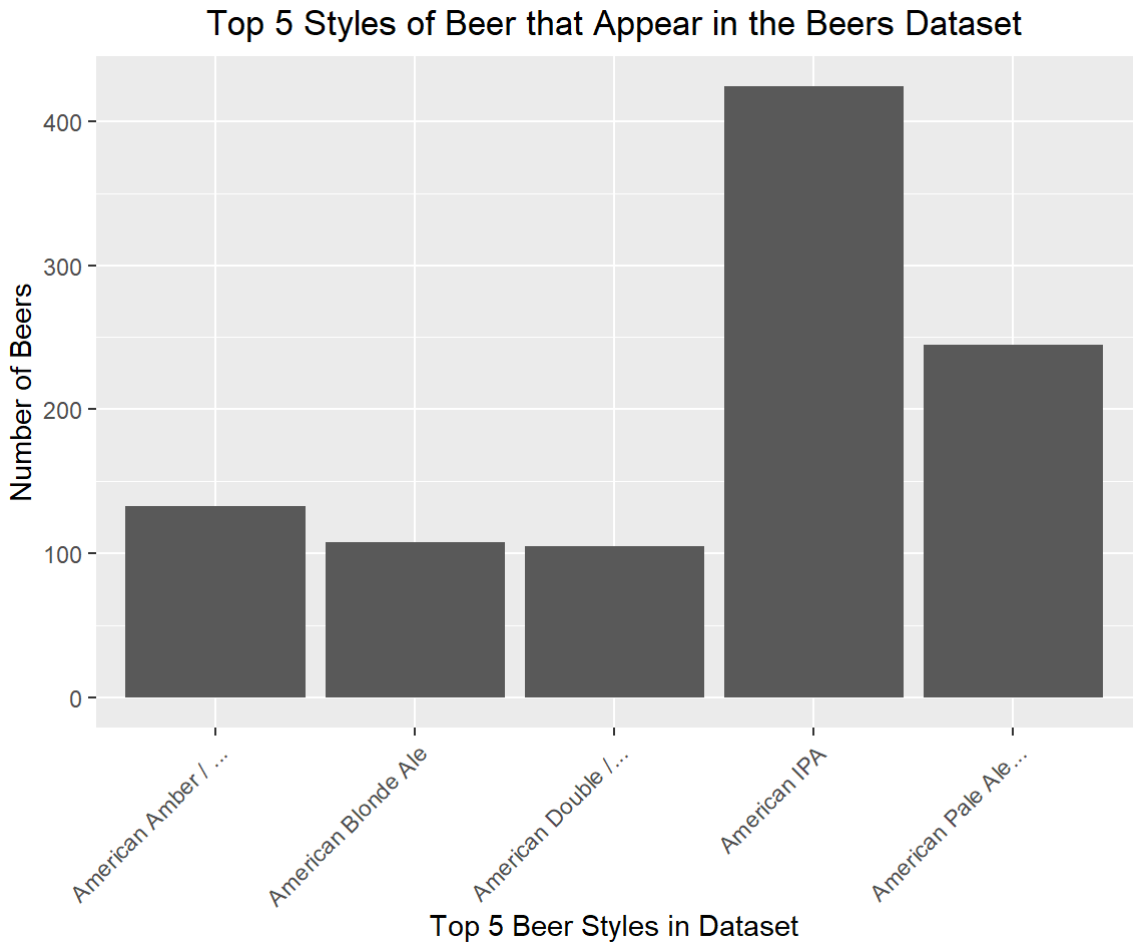
The histogram displays the frequency of ABV for the beers. From the histogram, we can see the majority of the beers have an ABV of approximately 5% with smaller numbers of beers having greater than 6% ABV.

```
# plot the most frequent IBU
hist(beer_df$ibu,
     main="IBU Frequency of Beers",
      ylab="Freqency",
      xlab="IBU")
```



The histogram displays the frequency of IBU for the beers. From the histogram, we can see the majority of the beers are not very bitter, observed by the sqewed nature of the histogram towards the mildly bitter end. Most of the beers from the dataset have a bitterness of between 10 and 20 IBU.

```
# count the number of each beer type
beer_type <- beer_df %>%
  count(style)

df_bt <-beer_type[order(beer_type$n,decreasing = TRUE),]
# get the top 5 beers
top_5_beer_types <- head(df_bt,5)

# Plot the 5 top beer types in the dataset

ggplot(data = top_5_beer_types, aes(x=style, y=n)) +
    geom_bar(stat="identity")+
  theme(axis.text.x = element_text(angle = 45, hjust=1))+
  # truncates labels
  scale_x_discrete(label = function(x) stringr::str_trunc(x, 20)) + ggtitle("Top 5 Styles of
 Beer that Appear in the Beers Dataset") +
  # Adds a theme and adds a centered title
  theme(plot.title = element_text(hjust = 0.5))+
  # Adds detailed caption information alongside a theme
  labs(x = "Top 5 Beer Styles in Dataset", y = "Number of Beers")
```



- The top beer present in the dataset is American IPA followed by America Pale Ale. The American IPA is almost twice as popular as the America pale ale. The other beers are represented in fewer quantities within the dataset than the top two beers.

```
# Beer with the highest abv
abv <- beer_df[order(beer_df$abv,decreasing = TRUE),]
ibu <- beer_df[order(beer_df$ibu,decreasing = TRUE),]
# max and min abv
max_abv <- head(abv,1)
min_abv <- tail(abv,1)
# max and min ibu
max_ibu <- head(ibu,1)
min_ibu <- tail(ibu,1)
```

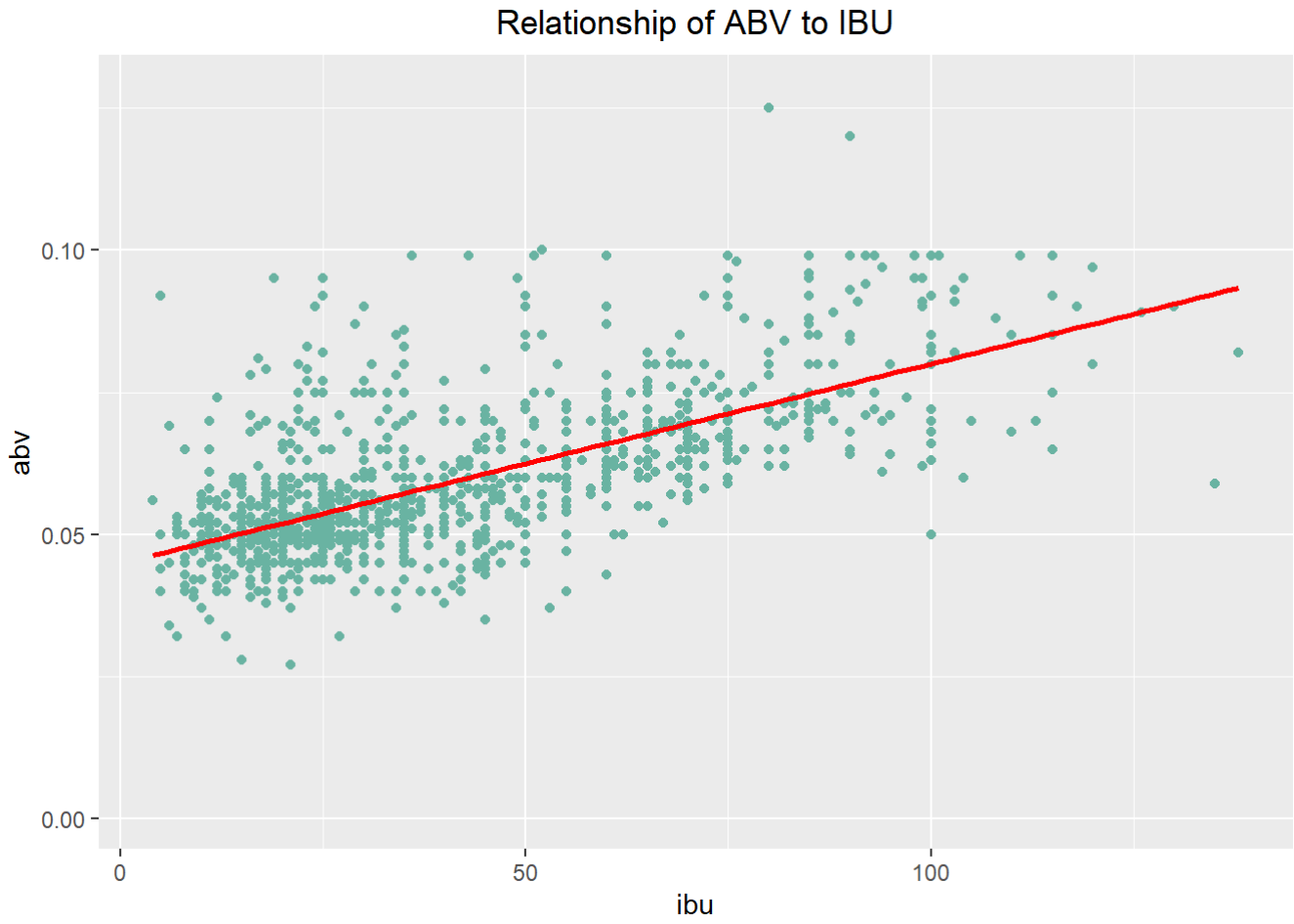
- The beer with the highest ABV is Lee Hill Series Vol. 5 - Belgian Style Quadrupel Ale with an ABV of 12.8 %, whilst the beer with the lowest ABV is Oâ€™Malleyâ€™s Irish Style Cream Ale with an ABV of 0.1 %.
- The beer with the highest IBU is Bitter Bitch Imperial IPA with an IBU of 138, whilst the beer with the lowest IBU is Rail Yard Ale (2009) with an IBU of 4.

```
# graph of ABV vs IBU
x <- beer_df$abv
y <- beer_df$ibu
ggplot(data = beer_df, aes(x=ibu, y=abv)) +
   ggtitle("Relationship of ABV to IBU") +
  theme(plot.title = element_text(hjust = 0.5))+
    geom_point( color="#69b3a2")+
  geom_smooth(method=lm , color="red", se=FALSE)
```

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

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

```
## Warning: Removed 1005 rows containing missing values (geom_point).
```



The plot above details the relationship between ABV and IBU from the 2017 beers dataset and a regression line is plotted. The plot shows the correlation between the bitterness and alcohol content. The higher the alcohol content the more bitter the beer

```
# getting the top 5 beers highest average abv
highest_avg_abv <- beer_df %>%
                  group_by(style) %>%
                  summarize(Mean = mean(abv*100, na.rm=TRUE))
# order the dataset
highest_average_abv <- highest_avg_abv[order(highest_avg_abv$Mean,decreasing = TRUE),]
# take top 5 abv
high_5_abv <- head(highest_average_abv,5)
high_5_abv
```

```
## # A tibble: 5 x 2
##   style                   Mean
##   <chr>                  <dbl>
## 1 English Barleywine     10.8
## 2 Quadrupel (Quad)       10.4
## 3 American Barleywine     9.9
## 4 American Malt Liquor    9.9
## 5 Russian Imperial Stout  9.76
```

The table above shows the top 5 beers highest average ABV. The beer with the highest average ABV is English Barleywine with an average ABV of 10.8 %, whilst the next highest is Quadrupel (Quad) with an average ABV of 10.4 %.

```
# getting the top 5 beers with the highest average ibu
highest_avg_ibu <- beer_df %>%
                  group_by(style) %>%
                  summarize(Mean = mean(ibu, na.rm=TRUE))
# order the dataset
highest_average_ibu <- highest_avg_ibu[order(highest_avg_ibu$Mean,decreasing = TRUE),]
# take top 5 ibu
head(highest_average_ibu,5)
```

```
## # A tibble: 5 x 2
##   style                              Mean
##   <chr>                             <dbl>
## 1 American Barleywine                96
## 2 American Double / Imperial IPA     93.3
## 3 Russian Imperial Stout             86.5
## 4 American Double / Imperial Pilsner 85
## 5 Belgian Strong Dark Ale            72
```

The table above shows the top 5 beers highest average IBU. The beer with the highest average IBU is American Barleywine with an average ABV of 96, whilst the next highest is American Double / Imperial IPA with an average ABV of 93.3.

```
# getting the top 5 beers with the lowest 5 average ibu
min_ibu <- beer_df %>%
                  group_by(style) %>%
                  summarize(Mean = mean(ibu, na.rm=TRUE))

mimimum_ibu <- min_ibu[order(min_ibu$Mean,decreasing = FALSE),]
head(mimimum_ibu,5)
```

```
## # A tibble: 5 x 2
##   style                   Mean
##   <chr>                  <dbl>
## 1 Berliner Weissbier      7.8
## 2 Gose                    9.43
## 3 American Adjunct Lager 11
## 4 Light Lager            11.7
## 5 Fruit / Vegetable Beer 14.2
```

The table above shows the top 5 beers lowest average IBU. The beer with the lowest average IBU is Berliner Weissbier with an average ABV of 7.8, whilst the next highest is Gose with an average ABV of 9.4.

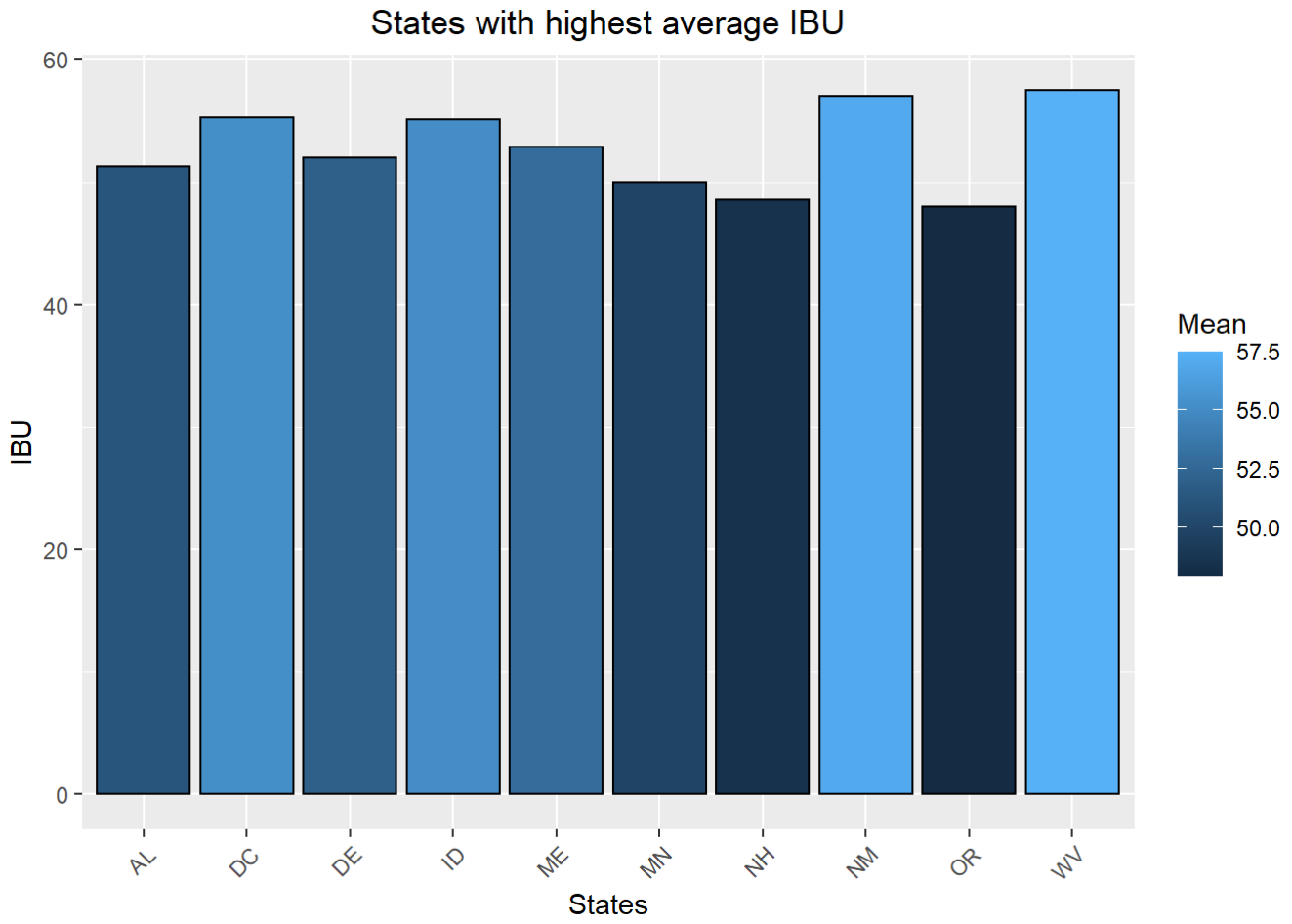
Merged Dataset Analysis

States with the Highest IBU

```
# getting the state with the highest 10 average ibu
avg_state_ibu <- df %>%
                group_by(state)%>%
                summarize(Mean = mean(ibu, na.rm=TRUE))
# order the results
average_state_ibu <- avg_state_ibu[order(avg_state_ibu$Mean,decreasing = TRUE),]
# take the top 10
average_state_ibu <- head(average_state_ibu,10)
average_state_ibu
```

```
## # A tibble: 10 x 2
##    state  Mean
##    <chr> <dbl>
##  1 WV     57.5
##  2 NM     57
##  3 DC     55.2
##  4 ID     55.1
##  5 ME     52.9
##  6 DE     52
##  7 AL     51.2
##  8 MN     50.0
##  9 NH     48.5
## 10 OR     47.9
```

```
# Plot the states with the highest 10 average ibu in the dataset

ggplot(data = average_state_ibu, aes(x=state, y=Mean, fill=Mean)) +
    geom_bar(stat="identity", color="black", position=position_dodge())+
  theme(axis.text.x = element_text(angle = 45, hjust=1))+
  # truncates labels
  scale_x_discrete(label = function(x) stringr::str_trunc(x, 20)) +
  ggtitle("States with highest average IBU") +
  # Adds a theme and adds a centered title
  theme(plot.title = element_text(hjust = 0.5))+
  # Adds detailed caption information alongside a theme
  labs(x = "States", y = "IBU")
```



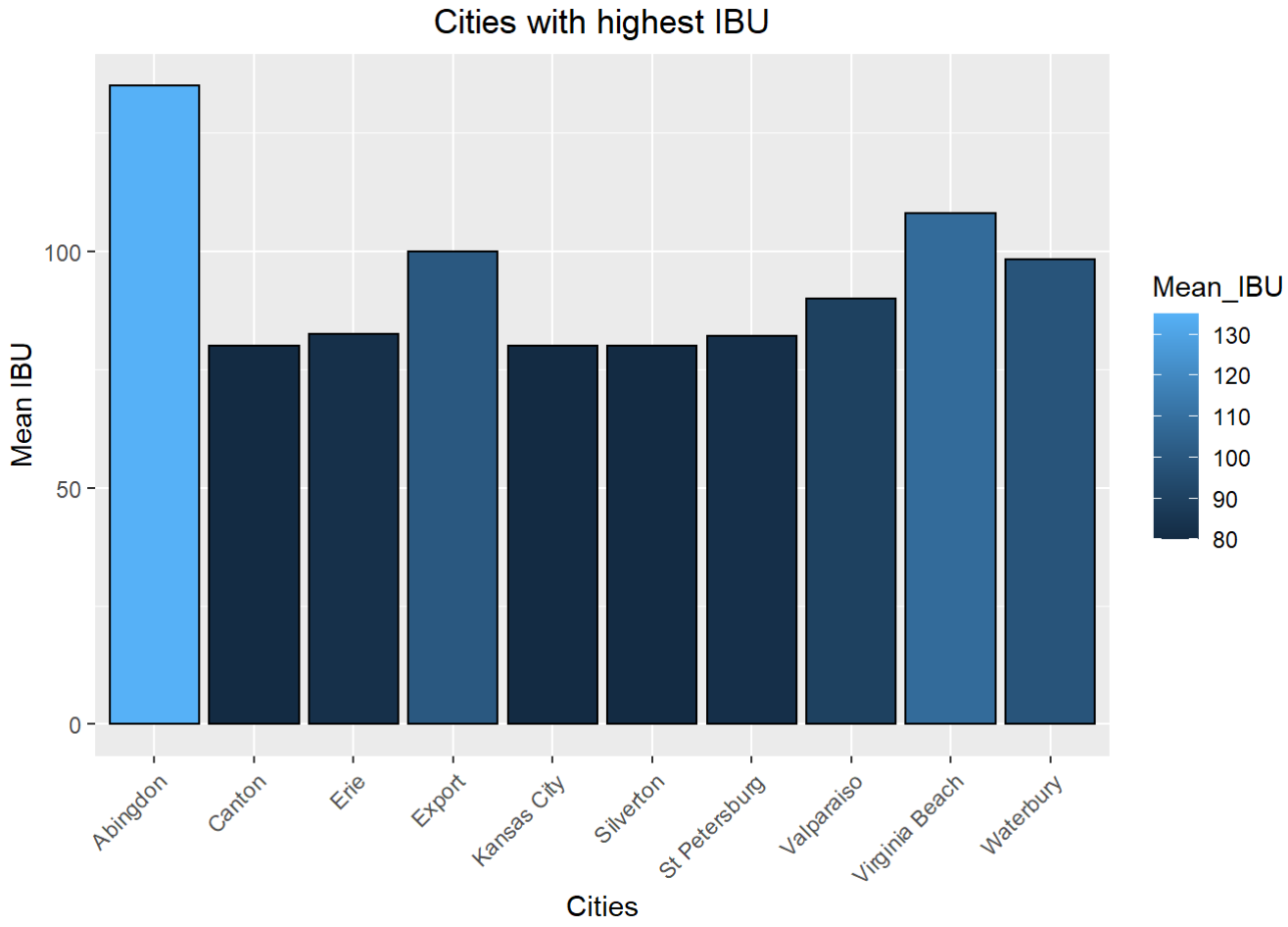
The chart shows the top 10 states with the highest average IBU. The state with the highest average IBU is West Virginia (WV) with an average IBU of 57.5, whilst the next highest is New Mexico (NM) with an average IBU of 57.

Cities with the Highest IBU

```
# Getting the cities with the highest 10 average ibu
avg_city_ibu <- df %>%
                group_by(city)%>%
                summarize(Mean_IBU= mean(ibu, na.rm=TRUE))
average_city_ibu <- avg_city_ibu[order(avg_city_ibu$Mean_IBU,decreasing = TRUE),]
average_cities_ibu <- head(average_city_ibu,10)
average_cities_ibu
```

```
## # A tibble: 10 x 2
##    city           Mean_IBU
##    <chr>             <dbl>
##  1 Abingdon          135
##  2 Virginia Beach    108
##  3 Export            100
##  4 Waterbury          98.3
##  5 Valparaiso         90
##  6 Erie               82.5
##  7 St Petersburg      82
##  8 Canton             80
##  9 Kansas City        80
## 10 Silverton          80
```

```
# Plotting the cities with the highest 10 average ibu
ggplot(data = average_cities_ibu, aes(x=city, y=Mean_IBU, fill=Mean_IBU)) +
    geom_bar(stat="identity", color="black", position=position_dodge())+
  theme(axis.text.x = element_text(angle = 45, hjust=1))+
  # truncates labels
  scale_x_discrete(label = function(x) stringr::str_trunc(x, 20)) + ggtitle("Cities with high
est IBU") +
  # Adds a theme and adds a centered title
  theme(plot.title = element_text(hjust = 0.5))+
  # Adds detailed caption information alongside a theme
  labs(x = "Cities", y = "Mean IBU")
```



The chart shows the top 10 cities with the highest average IBU. The state with the highest average IBU is Abingdon with an average IBU of 135, whilst the next highest is Virginia Beach with an average IBU of 108.

Conclusion

To summarize, we find that the state of Colorado has the highest number off breweries in the United States in 2017. This is followed closely by California with 39, Michigan with 32, Oregon with 29 and Texas with 28. Further, analysis of he cities within Colorado state found that the greatest concentration of breweries occurred in the main cities of Boulder and Denver followed by Edwards. As part of this process, an analysis of the beers also took place. The dataset also contains 2410 beers in a separate csv file from a diverse range of breweries through each state in the USA. Both the beers and breweries datasets were merged to provide an overall view of the craft beer dataset.

The key conclusions from analysing the 2017 craft beer data in the USA, are the following:

-The average alcohol strength (ABV) of all the beers surveyed is 5.97% whilst the average bitterness (IBU) of all the beers surveyed is 42.7. The most frequent ABV is 5% whilst most beers are not very bitter with IBU of between 10 and 20.

-The most common beer from the 2017 dataset is the American IPA and the second most common is American Pale Ale. The beer with the highest ABV is the Lee Hill Series Vol. 5 - Belgian Style Quadrupel Ale with an ABV of 12.8 %, whilst the beer with the lowest ABV is O'Malleys Irish Style Cream Ale with an ABV of 0.1 %.

-The most bitter beer (highest IBU) is Bitter Bitch Imperial IPA with an IBU of 138, whilst the beer with the lowest IBU is Rail Yard Ale (2009) with an ABI of 4. As might be expected there is a linear relationship between the strength of the beer and the bitterness. The higher the alcohol content the greater the bitterness of the beer.

-The Beers with the highest average ABV is English Barleywine, whilst American Barleywine has the highest average IBU. This makes sense that the beers with the highest alcohol content will tend to be the most bitter. Barleywines are a class of beer that tend to have very high alcohol content resembling that of wines, hence Barleywines. The beer with the lowest average IBU is Berliner Weissbier with an average ABV of 7.8.

-The merged dataset indicates that West Virginia is the state where the most bitter beers are to be found, followed by New Mexico. Analysis of the IBU by city indicates that Abingdon is where the most bitter beer is to be found, with an average IBU of 135, whilst the next highest is Virginia Beach with an average IBU of 108.

Part 2: R Package

As part of this section, I have chosen to explore the plotly package on the diamonds dataset. This is an R package that facilitates data manipulation and visualisation in a variety of different plot types from line graphs to more sophisticated charting options for exploratory data analysis. This allows the operators to easily visualize data and in a user friendly way. As part of this analysis I am using the diamonds dataset. It contains information on prices of diamonds, their attributes,price of the diamonds in 2008, as well as information on their carat, cut, color, and clarity. The dataset also includes some physical measurements.

```
# Import Libraries
library("plotly")
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
# Read the diamonds csv to a variable beer with first row containing column names and removin
g white space
diamonds <- read.csv("diamonds.csv", header = TRUE, strip.white=TRUE)
# Examine the structure of the diamonds dataset
str(diamonds)
```

```
## 'data.frame':    53940 obs. of  11 variables:
##  $ X      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
##  $ cut    : chr  "Ideal" "Premium" "Good" "Premium" ...
##  $ color  : chr  "E" "E" "E" "I" ...
##  $ clarity: chr  "SI2" "SI1" "VS1" "VS2" ...
##  $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

```
# Examine the top 6 rows
head(diamonds)
```

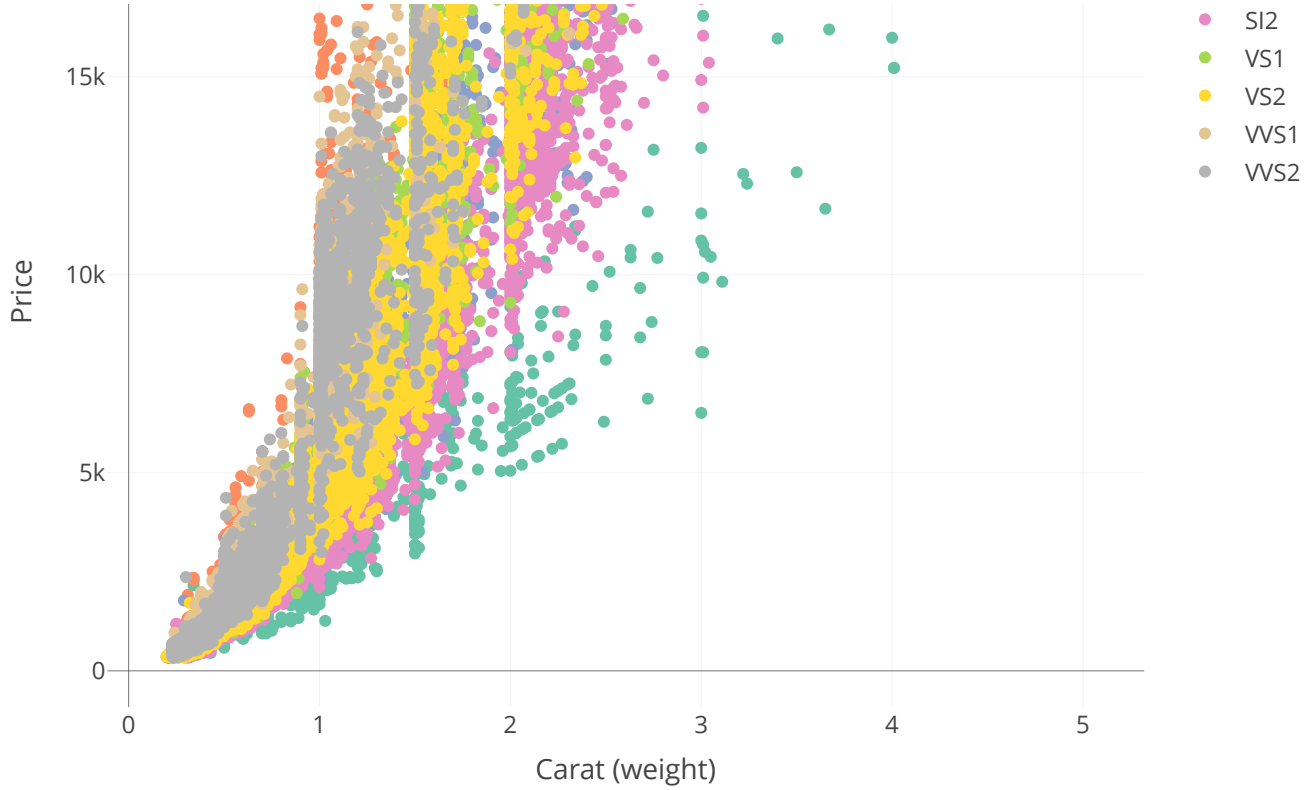
```
##   X carat       cut color clarity depth table price    x    y    z
## 1 1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
## 2 2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
## 3 3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
## 4 4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
## 5 5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
## 6 6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48
```

Scatter Plot

```
# scatter plot for weight and price by clarity
plot_ly(diamonds, x=~carat, y= ~price, color = ~clarity)%>%
  layout(title = "Plot of Weight and Price",
         xaxis = list(title = "Carat (weight)"),
         yaxis = list(title = "Price"))
```

```
## No trace type specified:
##   Based on info supplied, a 'scatter' trace seems appropriate.
##   Read more about this trace type -> https://plotly.com/r/reference/#scatter
```

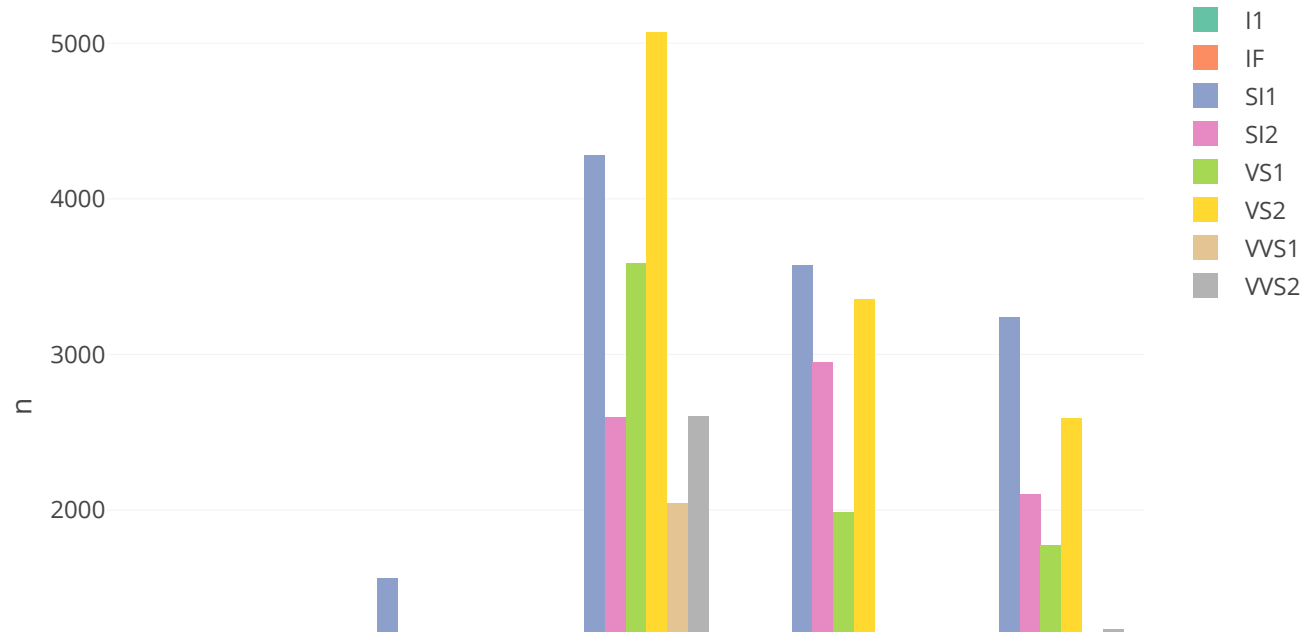
```
## No scatter mode specifed:
##   Setting the mode to markers
##   Read more about this attribute -> https://plotly.com/r/reference/#scatter-mode
```

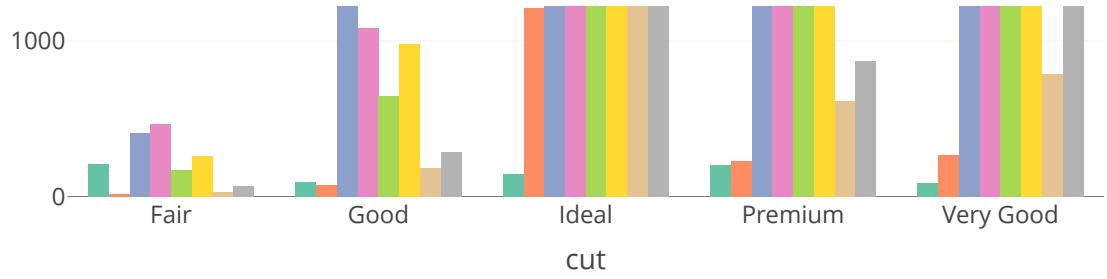


The plotly package has a excellent scatter plot that enables us to plot the quality of the clarity of the diamond affects the price. As we can see from the plot, there is a strong correlation between price and diamond weight (carat). The dots are also coloured by the clarity of the diamonds. As can be seen lighter diamonds that are Internally Flawless (IF) are more expensive than their heavier counterparts. Lower weight diamonds are less expensive than their higher weight counterparts. The interactive visualisation also allows the user to hover over a point and the detail of specific price, clarity and weight for the diamond is displayed. Other functions such as zoom, autoscale, image download and a variety of selection tools.

Bar Chart

```
# number of the diamonds for each cut/clarity combination
diamonds_cut_clarity <- diamonds %>% count(cut, clarity)
plot_ly(diamonds_cut_clarity, x=~cut, y= ~n, type="bar",color = ~clarity)
```

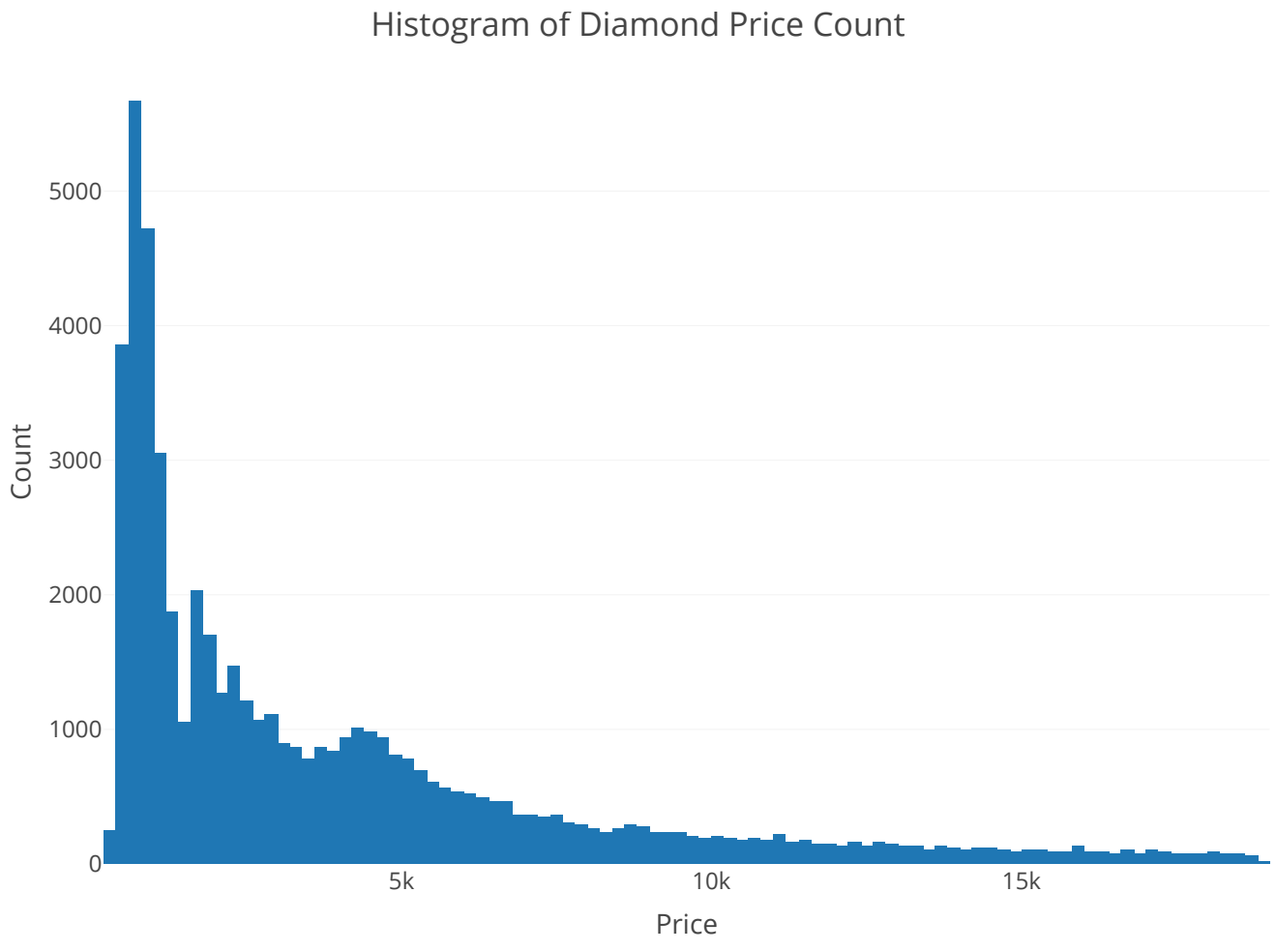




The plotly package also has the facility to create bar charts. The bar chart plots the number of diamonds for each cut. As we can see, most of the diamonds are cut as ideal, premium or very good. Lower quality diamonds, fair and good are less numerous. The bars are coloured based on the clarity of the diamonds. by the clarity of the diamonds. From the plot it is clear that SI1 and VS2 ideal diamonds are the most cut diamonds across the dataset. Again the interactive visualisation also allows the user to pan and zoom in and out to provide greater clarity.

Histogram

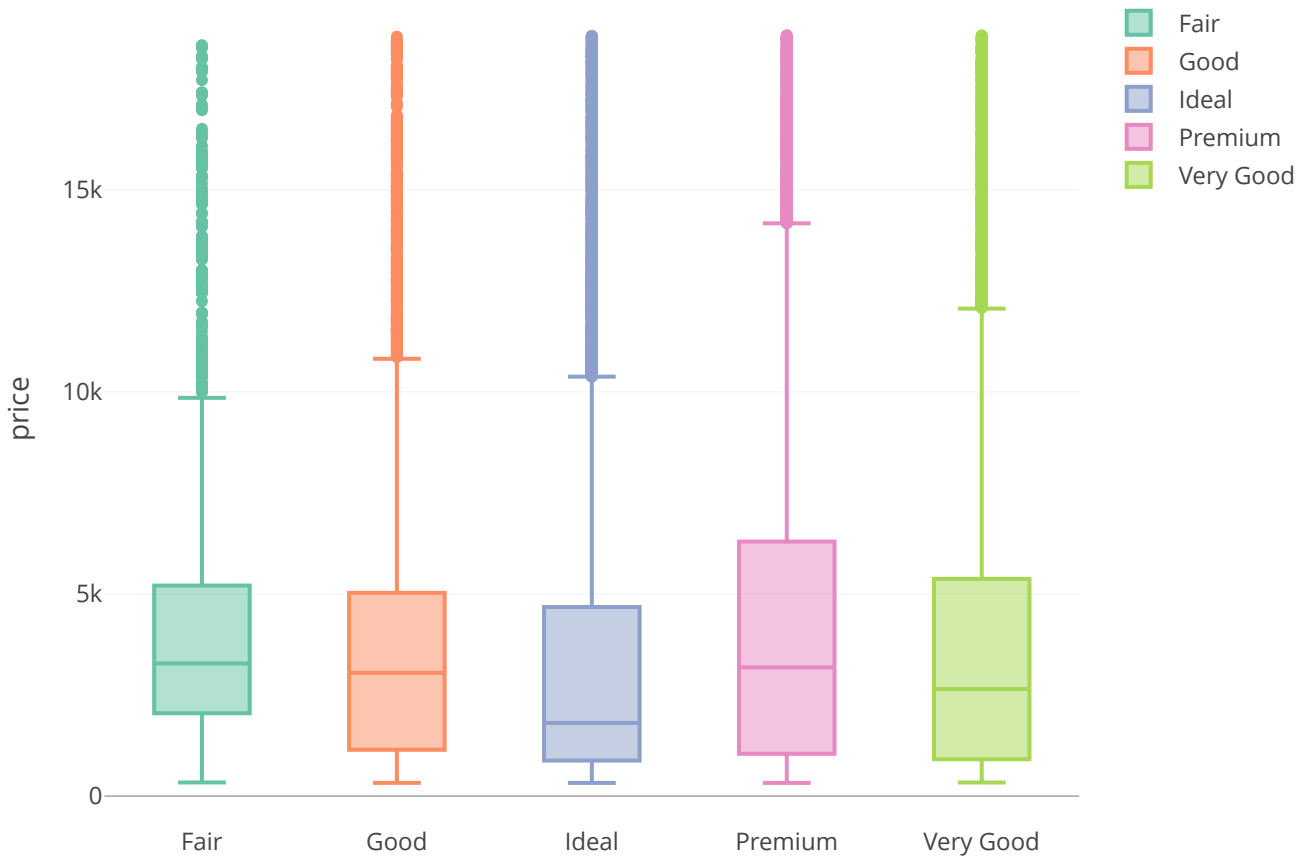
```
#histogram of the frequency of price of the diamonds
plot_ly(diamonds, x = ~price, type = "histogram") %>%
  layout(title = "Histogram of Diamond Price Count",
         xaxis = list(title = "Price",
                      zeroline = FALSE),
         yaxis = list(title = "Count",
                      zeroline = FALSE))
```



The plotly package also enables the user to develop histograms. We can see from the distribution of values that the histogram is skewed towards the lower price range with relatively few diamonds having prices beyond 5,000 dollars. Hovering over each bar gives frequency information.

Box Plots

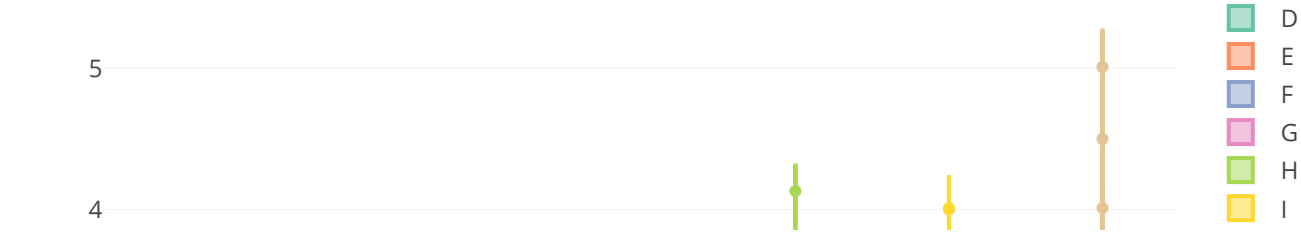
```
# box plot of price of the diamonds by cut category
plot_ly(diamonds, y = ~price, color = ~cut, type = "box")
```

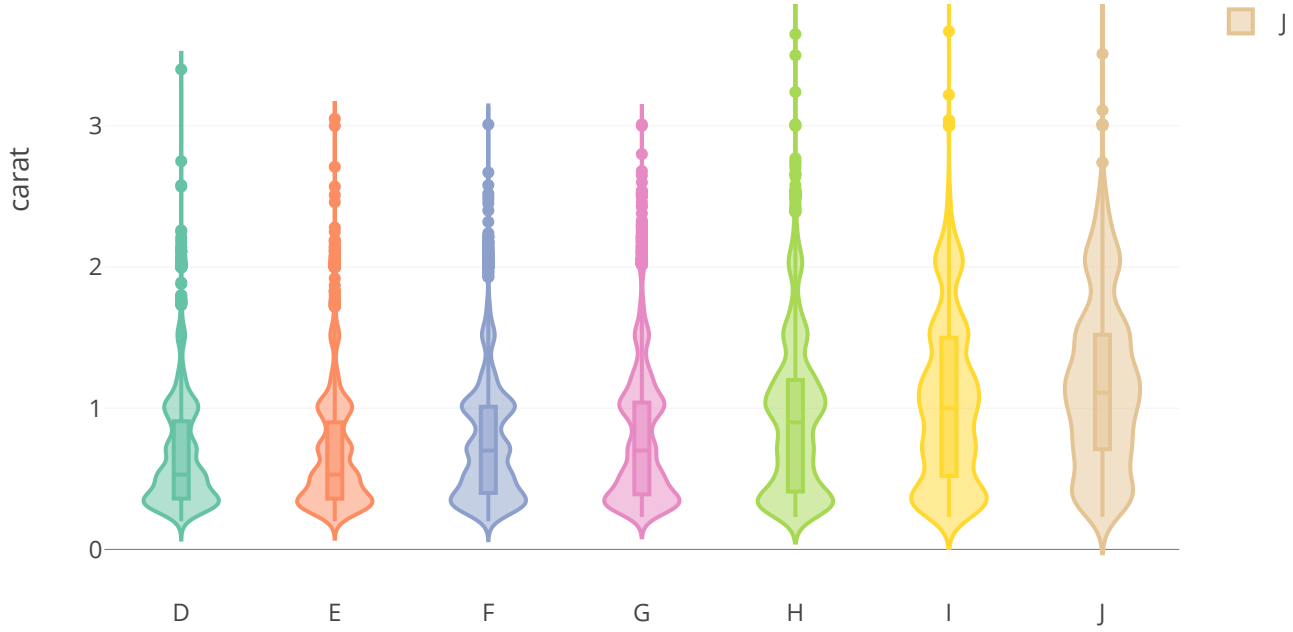


It can also produce box plots that give alot of information about the price of diamonds per cut category. In addition, hovering over the plot displays the statistics for the min, median and maximum prices for the specific cut of diamond.

Violin Plots

```
# violin plot of weight(carat) of the diamonds by colour category
plot_ly(diamonds, y = ~carat, color = ~color, type = "violin", box = list(visible =TRUE, widt
h = 0.2))
```





The violin plot shows the density of observations for the colour by carat. The line in the centre represents the median line, whilst the width represents the frequency of that weight. From the violin plot we can see that diamonds that have better colour have more density around lower weights. Diamonds that dont have a good colour tend to be larger.

Conclusion

The plotly package is an excellent plotting package for providing high quality data analysis and visualisation. This enables rapid and easy analysis of features for the analysis and evaluation of data modelling. The interactive feature also allow the user to get a better and more in depth understanding of the data without cluttering the visualisation with endless labels which are intrusive. In addition, much of the functionality is built in to the package, meaning the user can focus on the analysis which speeds up the CRISP-DM process in the early data modelling phases.

Part 3: Functions/Programming

The third part of the assignment requires that a working function be developed. The function will focus on providing the state where the beers are brewed and will assess the mean, median, max and minimum IBU and ABV statistics for the 2017 craft beers dataset. The summary method is as follows:

```
stateAnalysis <- function(state){

  # Creating a list to extract the ibu and abv
  filtered_df <- df[df$state == state, ]
  ibu_column <- filtered_df$ibu
  abv_column <- filtered_df$abv

  StateParameters <- list(name=state,
              beer_mean_ibu_state = mean(ibu_column, na.rm=TRUE ),
              beer_mean_abv_state = mean(abv_column, na.rm=TRUE),
              beer_max_ibu_state = max(ibu_column,na.rm=TRUE ),
              beer_max_abv_state = max(abv_column, na.rm=TRUE),
              beer_min_ibu_state = min(ibu_column,na.rm=TRUE),
              beer_min_abv_state = min(abv_column, na.rm=TRUE),
              beer_median_ibu_state = median(ibu_column,na.rm=TRUE),
              beer_median_abv_state = median(abv_column, na.rm=TRUE)

              )
  class(StateParameters) <- "state"
  return(StateParameters)
}
```

Summary S3 Class for providing Summary Statistics

The summary S3 class below will summarise the statistics in terms of max, min, mean and median IBU and ABV for a specific selected state and print out a table of the results.

```
# The summary S3 class

summary.state <- function(obj){
  cat("============================================================\n")
  cat("The summary Statistics for the State of", obj$name, "are as follows: ", "\n")
  cat("------------------------------------------------------------\n")
  cat("The average IBU is: ", round(obj$beer_mean_ibu_state,1), "\n")
  cat("The average ABV is: ", round(obj$beer_mean_abv_state*100,1),"%", "\n")
  cat("The max IBU is: ", round(obj$beer_max_ibu_state,1), "\n")
  cat("The max ABV is: ", round(obj$beer_max_abv_state*100,1),"%", "\n")
  cat("The min IBU is: ", round(obj$beer_min_ibu_state,1), "\n")
  cat("The min ABV is: ", round(obj$beer_min_abv_state*100,1),"%", "\n")
  cat("The median IBU is: ", round(obj$beer_median_ibu_state,1), "\n")
  cat("The median ABV is: ", round(obj$beer_median_abv_state*100,1),"%", "\n")
  cat("============================================================\n")
}
```

Print S3 Class for the State

The print S3 class below will print the top 5 records of the ABV, IBU, Beer Name, Style of the beers within the dataset. This approach is considered appropriate given that these are the most relevant columns that a somebody looking to choose a beer would be interested in viewing. The limit of 5 rows is also considered appropriate given the size of the dataset and ease of readability.

```
# The print S3 class below will print the name of the top 5 American IPA
# with the highest bitterness in ibu.

print.state <- function(obj){
  data <- subset(df, df$state == obj$name)
  cat("================================================================================\n")
  cat("     The ABV, IBU, Beer Name and Style Information for the State of", obj$name, "\n")
  cat("--------------------------------------------------------------------------------\n")
  data_set <- data %>% select(abv, ibu, beer_name, style)
  print(head(data_set,5))
  cat("--------------------------------------------------------------------------------\n")
  cat("The dataset has: ", nrow(data_set),"rows and ",ncol(data_set), "columns", "\n")
  cat("================================================================================\n")
  }
```

Plot S3 Class for Plotting top 5 strongest American IPA Beers

The print S3 class below will Plot the name of the top 5 American IPA Beers with the highest alcohol content in ABV in the specified state that produces the ale.

```
# Plot S3 Class for top 5 strongest in terms of ABV American IPA Beers per selected state
plot.state <- function(obj){
  state <- obj$name
  #set the chart header to be dynamic
  header <- chart_title <- substitute(paste("American IPA Beers in ",state," with highest AB
V",))

  #getting the American IPA Beers with highest abv
  highest_abv_state <- subset(df, df$state == obj$name)
  sorted_abv_State <- highest_abv_state[order(highest_abv_state$abv,decreasing = TRUE),]
  filtered_abv <- sorted_abv_State %>% filter(style == "American IPA")

  top_5_abv <- head(filtered_abv,5)

  # Plot the top 5 highest strength American IPA
  ggplot(data = top_5_abv, aes(x=beer_name, y=abv*100)) +
    geom_bar(stat="identity", color="black", position=position_dodge())+
  theme(axis.text.x = element_text(angle = 45, hjust=1))+
  # truncates labels
  scale_x_discrete(label = function(x) stringr::str_trunc(x, 20)) + ggtitle(header) +
  # Adds a theme and adds a centered title
  theme(plot.title = element_text(hjust = 0.5))+
  # Adds detailed caption information alongside a theme
  labs(x = "American IPA Beers", y = "ABV (%)")
}
```

To test the classes, they are run on the following american states:

-state of Colorado (CO)

-State of Washington(WA)

Results for the State of Colorado

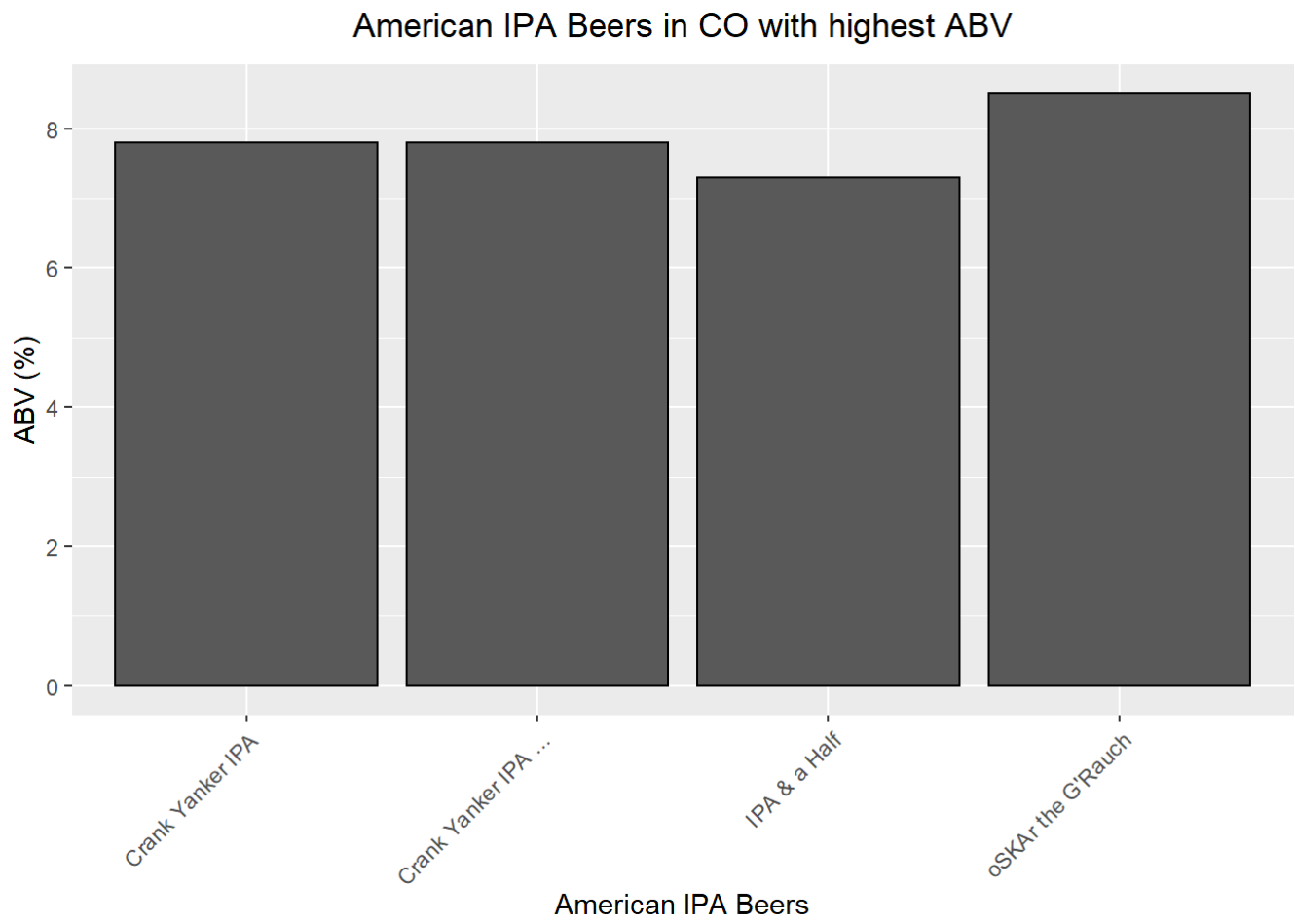
```
# The first step is to use the stateAnalysis function to create a state object.
# Once this has occurred this is then assigned to the variable US_State_Colorado, which
# details the specific information for the state of Colorado.

US_State_Colorado <- stateAnalysis("CO")

# The summary method for the state of Colorado will give the statistics for the state.
summary(US_State_Colorado)
```

```
## ============================================================
## The summary Statistics for the State of CO are as follows:
## ------------------------------------------------------------
## The average IBU is:  47.4
## The average ABV is:  6.3 %
## The max IBU is:  104
## The max ABV is:  12.8 %
## The min IBU is:  9
## The min ABV is:  4.1 %
## The median IBU is:  40
## The median ABV is:  6 %
## ============================================================
```

```
# The print method will print ABV,IBU, beer name and Beer Style for the state.
print(US_State_Colorado)
```

```
## ================================================================================
##      The ABV, IBU, Beer Name and Style Information for the State of CO
## --------------------------------------------------------------------------------
##      abv ibu                              beer_name                   style
## 41 0.050  NA Denver Pale Ale (Artist Series No. 1) American Pale Ale (APA)
## 42 0.087  NA                       Hibernation Ale                 Old Ale
## 43 0.061  NA                            Whitewater American Pale Wheat Ale
## 44 0.071  NA                                Rumble            American IPA
## 45 0.083  NA                              Orabelle                  Tripel
## --------------------------------------------------------------------------------
## The dataset has:  265 rows and  4 columns
## ================================================================================
```

```
# The plot method for the state of Colorado will give a
# bar plot of the selected state of the top 5 strongest
# American IPA in terms of ABV for the state of Colorado
plot(US_State_Colorado)
```



Results for the State of Washington

```
# The first step is to use the stateAnalysis function to create a state object.
# Once this has occurred this is then assigned to the variable US_State_Washington, which
# details the specific information for the state of Washington.

US_State_Washington <- stateAnalysis("WA")

# The summary method for the state of Washington will give the statistics for the state.
summary(US_State_Washington)
```

```
## ============================================================
## The summary Statistics for the State of WA are as follows:
## ------------------------------------------------------------
## The average IBU is:  45
## The average ABV is:  5.8 %
## The max IBU is:  83
## The max ABV is:  8.4 %
## The min IBU is:  18
## The min ABV is:  4 %
## The median IBU is:  38
## The median ABV is:  5.5 %
## ============================================================
```

```
# The print method will print ABV,IBU, beer name and Beer Style for the state.
print(US_State_Washington)
```

```
## ================================================================================
##      The ABV, IBU, Beer Name and Style Information for the State of WA
## --------------------------------------------------------------------------------
##        abv ibu                                beer_name                    style
## 1030 0.043  60 Little Sister India Style Session Ale             American IPA
## 1031 0.062  80                       Country Boy IPA             American IPA
## 1129 0.041  41                     Day Hike Session              American IPA
## 1130 0.048  48                        Trailhead ISA              American IPA
## 1131 0.052  27                      Immersion Amber American Amber / Red Ale
## --------------------------------------------------------------------------------
## The dataset has:  68 rows and  4 columns
## ================================================================================
```

```
# The plot method for the state of Washington will give a
# bar plot of the selected state of the top 5 strongest
# American IPA in terms of ABV for the state of Washington
plot(US_State_Washington)
```