DDS_Case_Study_1

```
require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
## 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
```

```
require(tidyr)
```

```
## Loading required package: tidyr
```

```
require(knitr)
```

```
## Loading required package: knitr
```

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
require(maps)
```

```
## Loading required package: maps
```

```
require(RColorBrewer)
```

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

Clean Breweries Data

```
breweries_data <- read.csv("../data/Breweries.csv", header=TRUE)

str(breweries_data)
```

```
## 'data.frame':    558 obs. of  4 variables:
##  $ Brew_ID: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Name   : Factor w/ 551 levels "10 Barrel Brewing Company",..: 355 12 266 319 201 136 227 477 59 49
##  $ City   : Factor w/ 384 levels "Abingdon","Abita Springs",..: 228 200 122 299 300 62 91 48 152 136
##  $ State  : Factor w/ 51 levels " AK"," AL"," AR",..: 24 18 20 5 5 41 6 23 23 23 ...
```

```
# confirm brew_id is a unique key
# summary(breweries_data)
# summarize breweries
# breweries_summary <-
#   select(breweries_data, Brew_ID, City, State, Name) %>%
#   group_by(Name) %>%
#   summarize_all(funs(
#     count = n_distinct(Brew_ID, City, State)
```

```
#   )) %>%
#   arrange(desc(Brew_ID_count))

# remove punctionation from all columns and trim whitespace
breweries_data <- as.data.frame(apply(breweries_data, 2, function(x) trimws(gsub('[[:punct:] ]+',' ',x))
breweries_data$Name <- as.factor(breweries_data$Name)
breweries_data$Brew_ID <- as.integer(breweries_data$Brew_ID)

# confirm Brew_ID + City + State is a unique key
breweries_summary <-
  select(breweries_data, Brew_ID, City, State, Name) %>%
  group_by(Name) %>%
  summarize_all(funs(
    count = n_distinct(Brew_ID, City, State)
  )) %>%
  arrange(desc(Brew_ID_count))


# capture potential duplicates
breweries_dups <- filter(breweries_summary, Brew_ID_count > 1)

# rejoin potential dups to original dataset
breweries_dups <- select(breweries_dups %>% inner_join(breweries_data), -ends_with("_count"))
```

```
## Joining, by = "Name"
```

```
# Fix Errors #

# Fix Brew_ID=378, change City(Menominee -> Menominie)
breweries_dups <- breweries_dups %>%
     mutate(City=replace(City, Brew_ID==378, "Menominie")) %>%
     as.data.frame()

# Fix Brew_ID=96, change State(MA -> MI)
breweries_dups <- breweries_dups %>%
     mutate(State=replace(State, Brew_ID==96, "MI")) %>%
     as.data.frame()

#capture known duplicates
breweries_dups <- breweries_dups %>%
                  group_by(Name, City, State) %>%
                  filter(n()>1)


#create surrogate key for duplicates
breweries_sk <- breweries_dups %>%
                    group_by(Name, City, State) %>%
                    summarize_all(funs(
                      Brew_SK = (sum(Brew_ID)*sum(Brew_ID)),
                      count = n()
                      )) %>% #end summarize_all
                    ungroup() %>%
                    right_join(breweries_dups) %>% #rejoin to dupes by name, city, state
                    select(Brew_ID, Brew_SK)
```

```
## Joining, by = c("Name", "City", "State")
```

```
breweries_data$Brew_ID[(breweries_data$Brew_ID %in% breweries_sk$Brew_ID)] <- breweries_sk$Brew_SK
```

```
breweries_clean <- distinct(breweries_data, Brew_ID, .keep_all = TRUE) %>% rename(Brewery_Name = Name)
```

Clean Beer Data

```
beer_data <- read.csv("../data/Beers.csv", header=TRUE)
```

```
head(beer_data)
```

```
##                  Name Beer_ID   ABV IBU Brewery_id
## 1            Pub Beer    1436 0.050  NA        409
## 2         Devil's Cup    2265 0.066  NA        178
## 3 Rise of the Phoenix    2264 0.071  NA        178
## 4            Sinister    2263 0.090  NA        178
## 5       Sex and Candy    2262 0.075  NA        178
## 6        Black Exodus    2261 0.077  NA        178
##                             Style Ounces
## 1             American Pale Lager     12
## 2        American Pale Ale (APA)     12
## 3                  American IPA     12
## 4 American Double / Imperial IPA     12
## 5                  American IPA     12
## 6                 Oatmeal Stout     12
```

```
beer_data$Brewery_id[(beer_data$Brewery_id %in% breweries_sk$Brew_ID)]  <- breweries_sk$Brew_SK #update
```

```
## Warning in beer_data$Brewery_id[(beer_data$Brewery_id %in% breweries_sk
## $Brew_ID)] <- breweries_sk$Brew_SK: number of items to replace is not a
## multiple of replacement length
```

```
beer_clean <- distinct(beer_data) %>% rename(Brew_ID = Brewery_id, Beer_Name = Name)
```

Question 1

```
state_ll <- read.csv("../data/state_coords.csv") %>% mutate(State = toupper(State)) %>% rename(state = S
states <- map_data("state") %>%
          mutate(region = toupper(region)) %>%
          rename(state=region) %>%
          select(long, lat, state, group)

states <- states %>%
          left_join(
            states %>%
            group_by(state) %>%
            summarise_all(funs(n=n())) %>%
            select(state, group_n) %>%
```

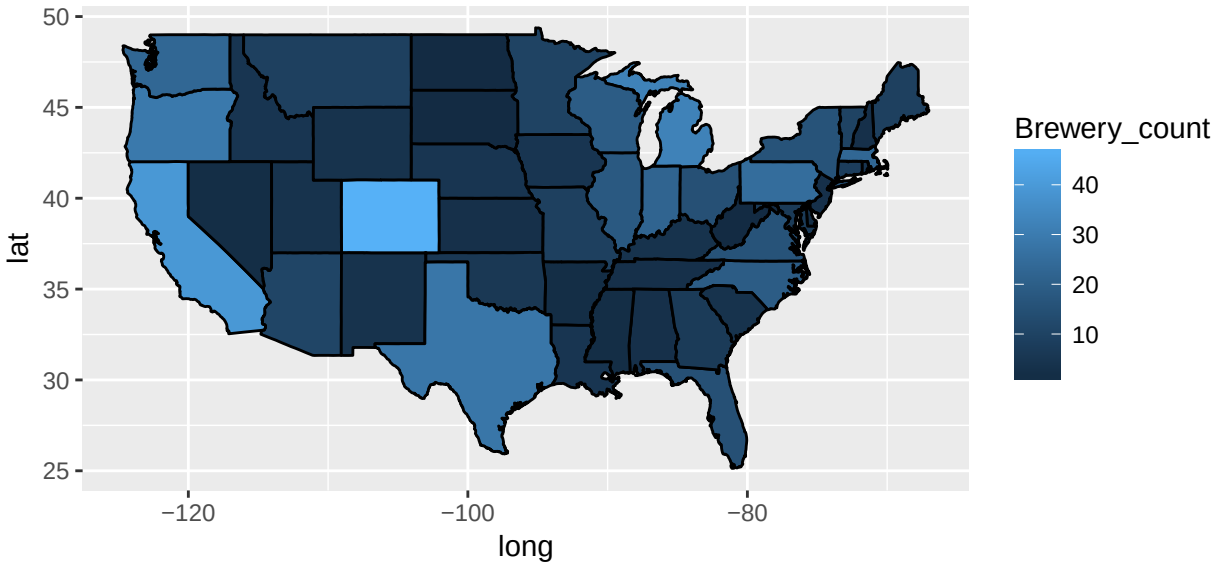
```
            distinct(state, .keep_all = TRUE)
          )
```

```
## Joining, by = "state"
```

```
breweries_by_state <- select(breweries_clean, Brew_ID, State) %>%
  group_by(State) %>%
  summarise_all(funs(Brewery_count = n()))

# state_ll %>%
#   inner_join(states)

ggplot(data = breweries_by_state %>%
  inner_join(state_ll, by=c("State" = "Abbr")) %>%
  inner_join(states)) +
  geom_polygon(aes(x = long, y = lat, group=group, fill=Brewery_count), color = "black") +
  #geom_text(aes(x = long, y = lat, label = as.character(Brewery_count), color = "black")) +
  coord_fixed(1.3) +
  guides(alpha=FALSE)
```

```
## Warning: Column `State`/`Abbr` joining character vector and factor,
## coercing into character vector
```

```
## Joining, by = "state"
```




```
  # scale_fill_gradientn(colours = "black",
  #                      breaks = c(2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50))
```

```
summary(breweries_by_state)
```

```
##     State           Brewery_count
##  Length:51          Min.   : 1.00
##  Class :character   1st Qu.: 3.50
##  Mode  :character   Median : 7.00
##                     Mean   :10.88
##                     3rd Qu.:16.00
##                     Max.   :47.00
```

Question 2

```
# merge beer and breweries
merged_data <- breweries_clean %>%
               full_join(beer_clean, by="Brew_ID")


#TODO: Plot -> brews by brewery
```

Question 3

```
# Number of nulls in each column
merged_data %>%
  select_if(function(x) any(is.na(x))) %>%
  summarise_all(funs(sum(is.na(.))))
```

```
##   ABV  IBU
## 1  62 1005
```

```
#TODO: add plot?
```

Question 4

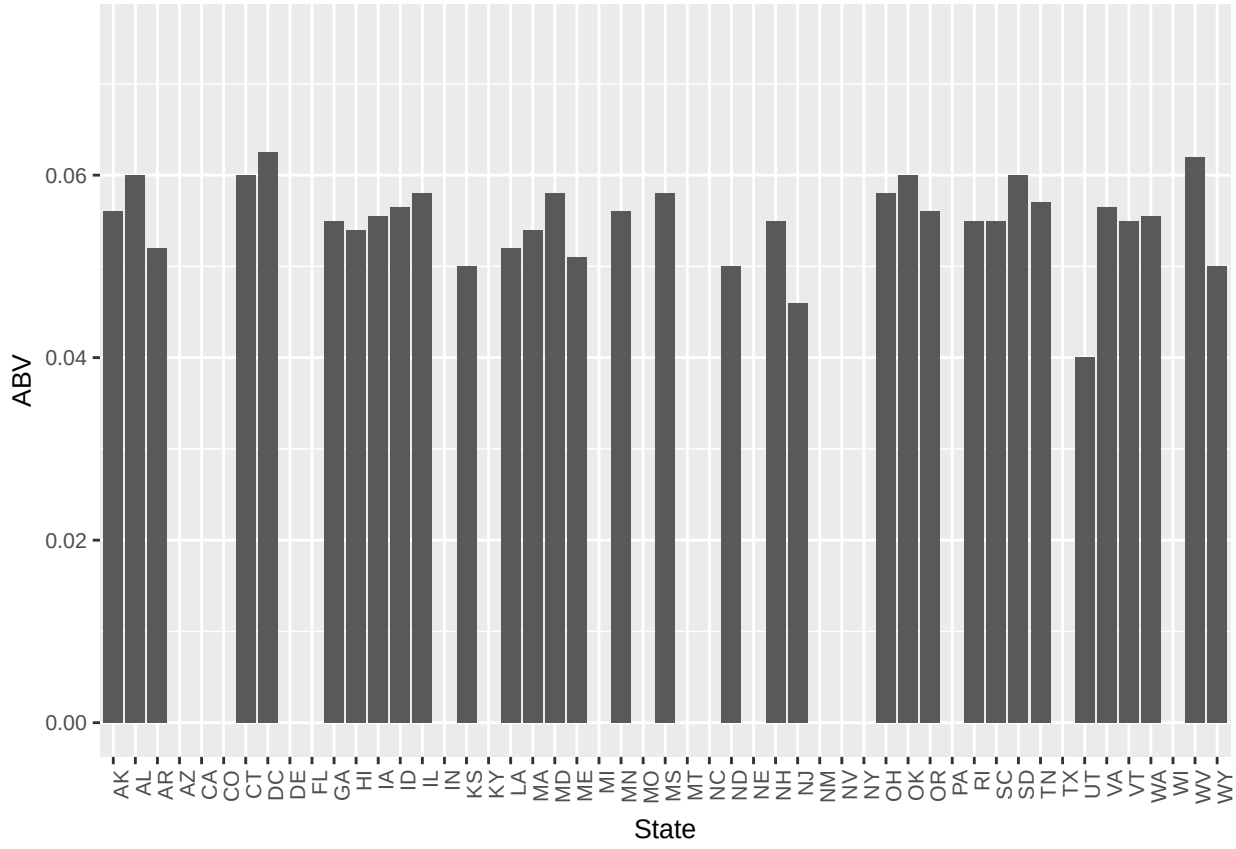
```
#fig.width=11

# compute median ABV and IBU by state


merged_by_state <- select(merged_data, State, ABV, IBU) %>%
                   group_by(State) %>%
                   summarise_all(median)#funs(median(!is.na(.)))) #TODO: Double check this is calculati

merged_by_state$State <- as.factor(merged_by_state$State)


# vars <- rbind(merged_by_state %>% mutate(var="ABV") %>%rename(value=ABV) %>% select(State, var, value
#       merged_by_state %>% mutate(var="IBU") %>%rename(value=IBU) %>% select(State, var, value))
#
# vars$value
```

```
summary(merged_by_state)
```

```
##      State          ABV               IBU        
##  AK     : 1   Min.   :0.04000   Min.   :32.00  
##  AL     : 1   1st Qu.:0.05400   1st Qu.:33.88  
##  AR     : 1   Median :0.05550   Median :39.75  
##  AZ     : 1   Mean   :0.05514   Mean   :42.25  
##  CA     : 1   3rd Qu.:0.05800   3rd Qu.:48.12  
##  CO     : 1   Max.   :0.06250   Max.   :57.50  
##  (Other):45   NA's   :18        NA's   :47
```

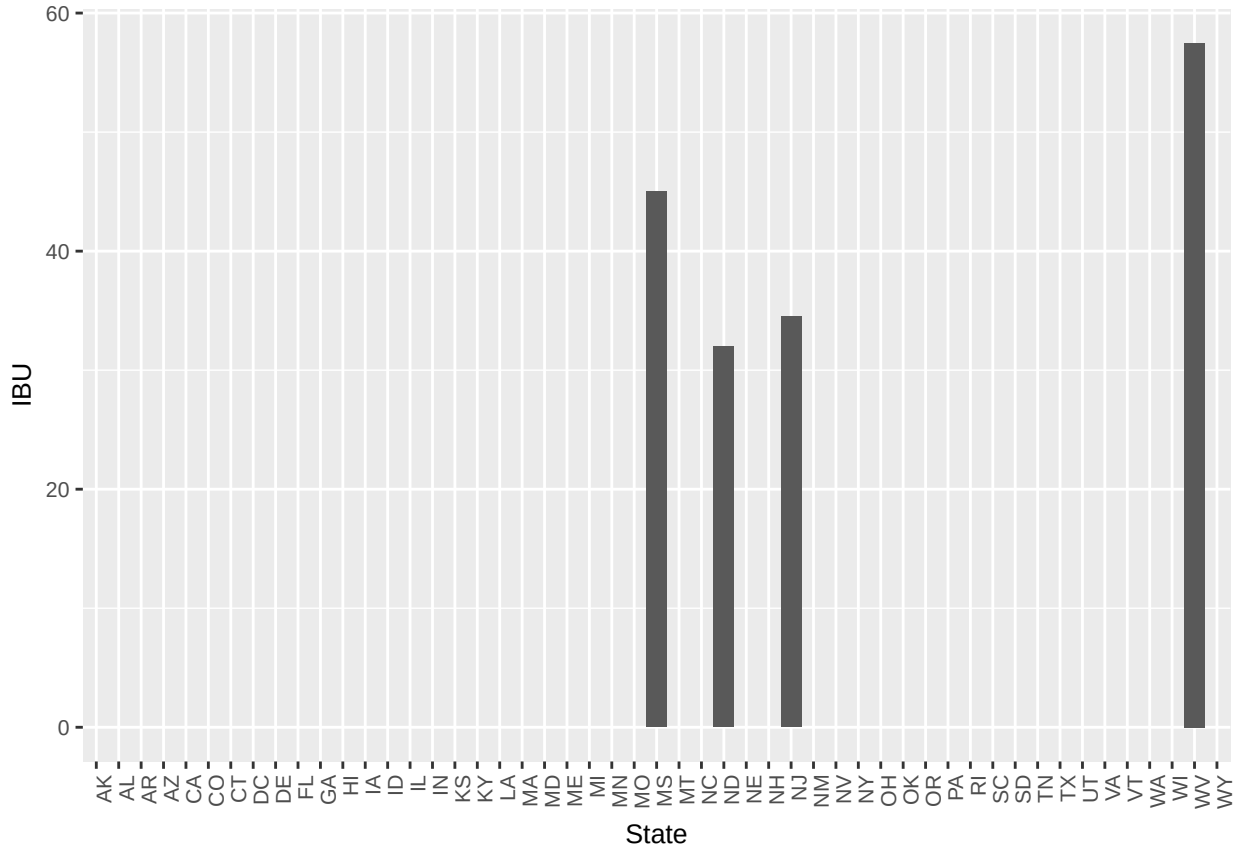
```
ggplot(merged_by_state, aes(x=State, y=ABV)) +
  geom_bar(stat = "identity", position = "dodge") +
  ylim(0, .075) +
  theme(text = element_text(size=10),
        axis.text.x = element_text(angle=90, hjust=1))
```

```
## Warning: Removed 18 rows containing missing values (geom_bar).
```




```
ggplot(merged_by_state, aes(x=State, y=IBU)) + #TODO: something is fishy with IBU
  geom_bar(stat = "identity", position = "dodge") +
  #ylim(0, .075) +
  theme(text = element_text(size=10),
        axis.text.x = element_text(angle=90, hjust=1))
```

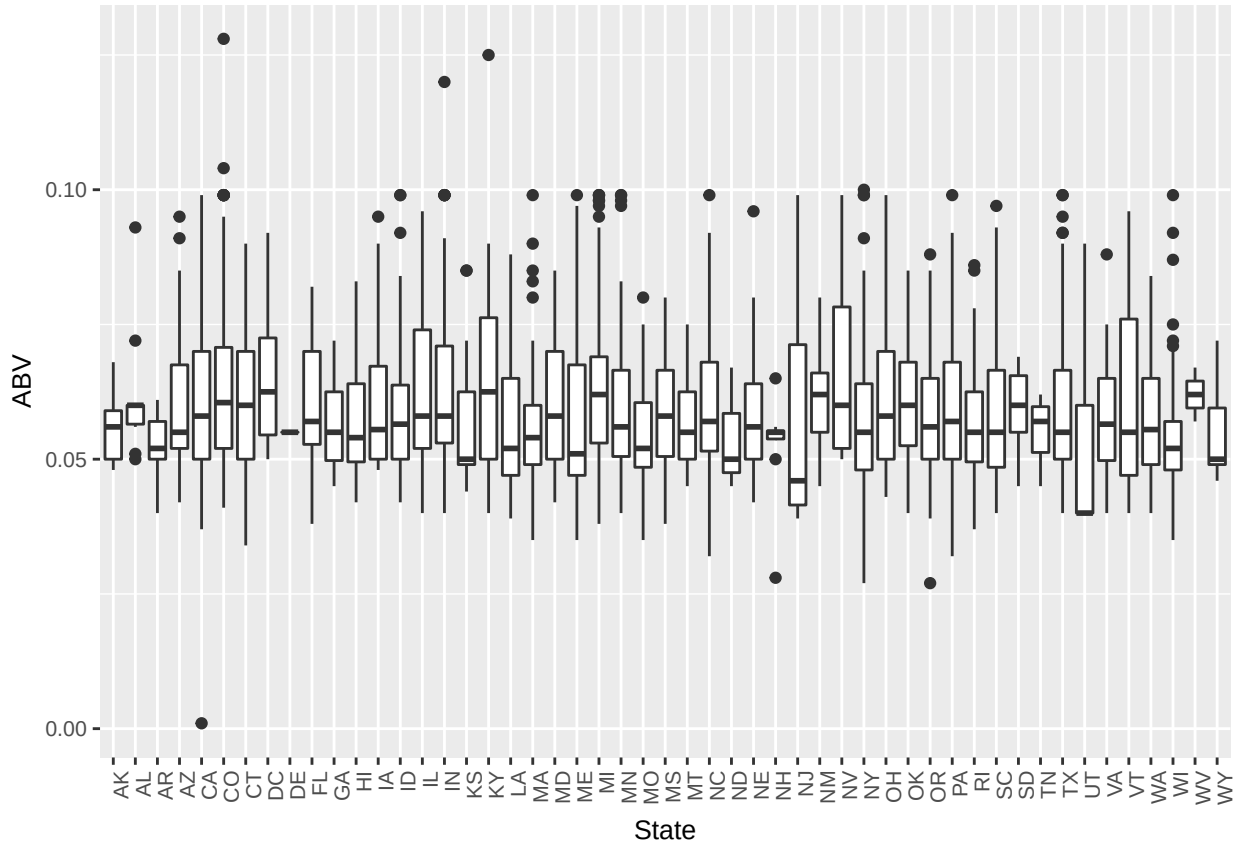
```
## Warning: Removed 47 rows containing missing values (geom_bar).
```



Question 5

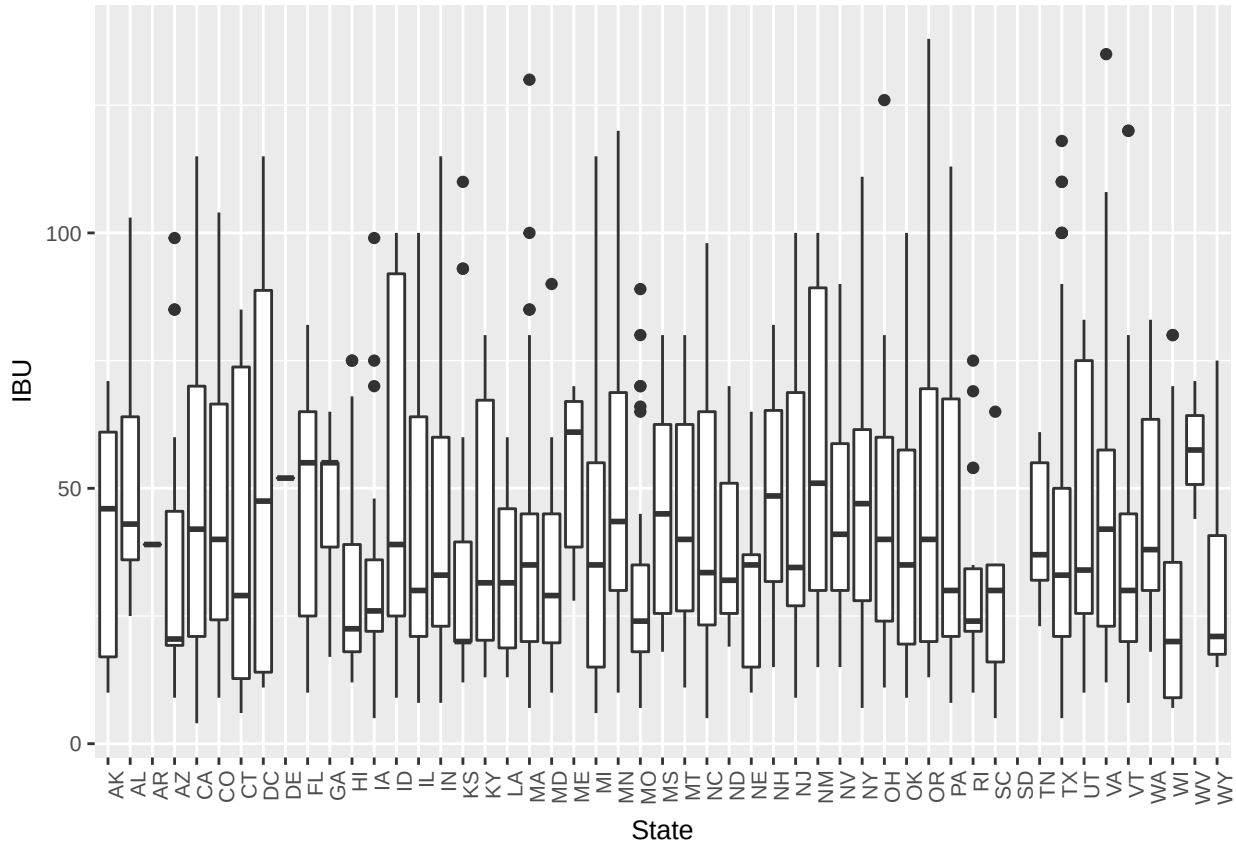
```
# max_abv <- max(merged_data$ABV, na.rm = TRUE)


ggplot(merged_data, aes(x=State , y=ABV)) +  #TODO: Make Pretty
  geom_boxplot() +
  #ylim(0, .075) +
  theme(text = element_text(size=10),
        axis.text.x = element_text(angle=90, hjust=1))
```

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




```
ggplot(merged_data, aes(x=State , y=IBU)) +  #TODO: Make Pretty
  geom_boxplot() +
  #ylim(0, .075) +
  theme(text = element_text(size=10),
        axis.text.x = element_text(angle=90, hjust=1))
```

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




```
max_abv <-  (select(merged_data, State, ABV) %>%
                   group_by(State) %>%
                   #filter(ABV == max(ABV)) %>%
                   arrange(desc(ABV))  %>% #sort by ABV
                   filter(row_number() == 1))[1,] #get first row

max_abv
```

```
## # A tibble: 1 x 2
## # Groups: State [1]
##   State   ABV
##   <chr> <dbl>
## 1 CO    0.128
```

```
max_ibu <-  (select(merged_data, State, IBU) %>%
                   group_by(State) %>%
                   #filter(ABV == max(ABV)) %>%
                   arrange(desc(IBU))  %>% #sort by ABV
                   filter(row_number() == 1))[1,] #get first row

max_ibu
```

```
## # A tibble: 1 x 2
## # Groups: State [1]
##   State   IBU
##   <chr> <int>
## 1 OR      138
```

Question 6

```
#summaryize ABV

# tidy_summary <- tidy(summary(merged_data$ABV)) #For some reason this line wont knit


abv_stats <- as.data.frame(t(summary(merged_data$ABV))) %>% #summarize and transpose
             rename("ABV"=Freq, Statistic=Var2) %>%
             select(Statistic, ABV)

abv_stats$ABV <- round(abv_stats$ABV, digits = 3)


abv_stats #TODO: Add IQR, stdev
```

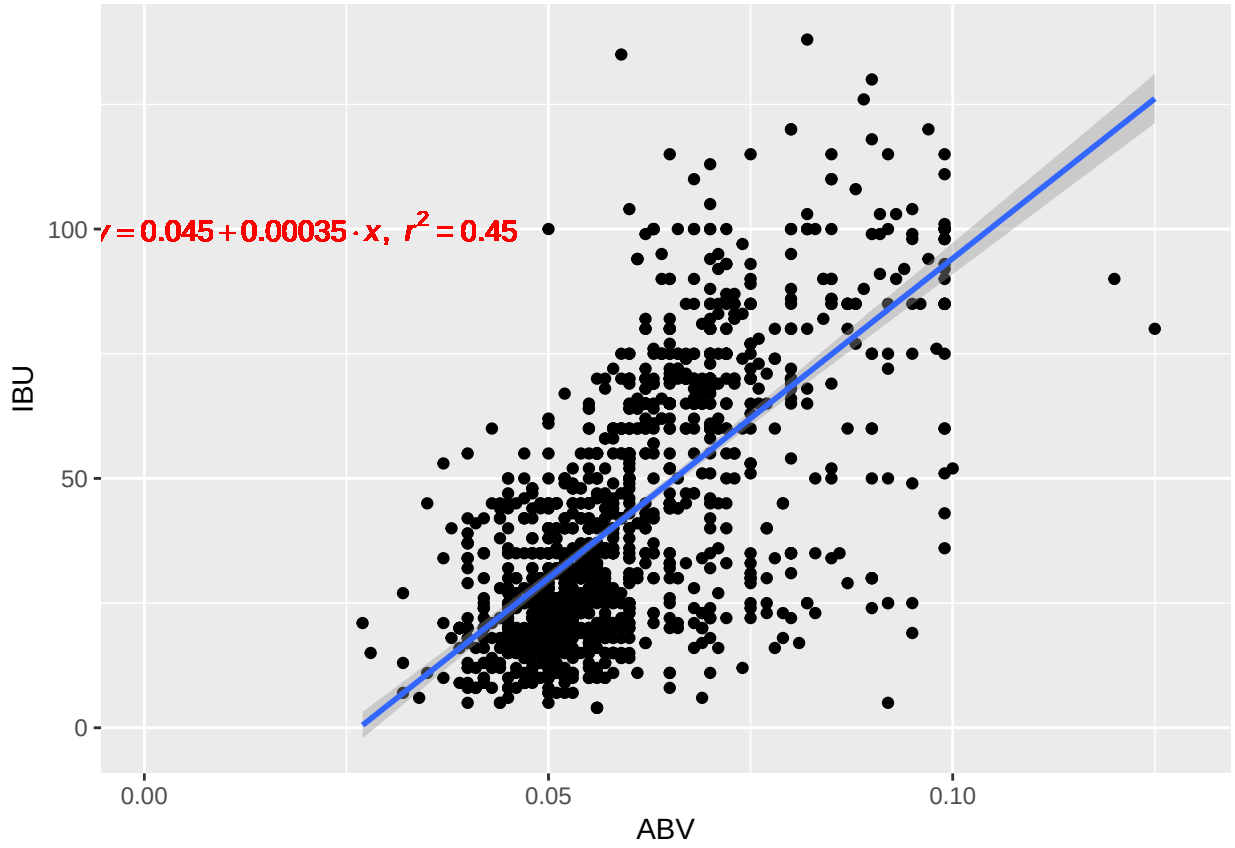
```
##   Statistic    ABV
## 1      Min.  0.001
## 2   1st Qu.  0.050
## 3    Median  0.056
## 4      Mean  0.060
## 5   3rd Qu.  0.067
## 6      Max.  0.128
## 7      NA's 62.000
```

Question 7

```
# fig.height=48
#plot relationshiop of ABV and IBU

#retreive linear model equation -- source(https://stackoverflow.com/questions/7549694/adding-regression
lm_eqn = function(m) {

  l <- list(a = format(coef(m)[1], digits = 2),
      b = format(abs(coef(m)[2]), digits = 2),
      r2 = format(summary(m)$r.squared, digits = 3));

  if (coef(m)[2] >= 0)  {
    eq <- substitute(italic(y) == a + b %.% italic(x)*","~~italic(r)^2~"="~r2,l)
  } else {
    eq <- substitute(italic(y) == a - b %.% italic(x)*","~~italic(r)^2~"="~r2,l)
  }

  as.character(as.expression(eq));
}
ggplot(beer_clean, aes(x=ABV, y=IBU)) +
  geom_point() +
  geom_smooth(method = "lm") +
  geom_text(aes(x = .02, y = 100, label = lm_eqn(lm(ABV ~ IBU ,beer_clean))), parse = TRUE, color = "re
```

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

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




```
# Yes, there is a positive relationship between ABV and IBU. #TODO:Add explanation
```