615_Assignment3_Honey

Dave Anderson, Sky Liu, Tingrui Huang, Xiang Xu

October 3, 2018

```
#Loading data files
Honey <- read_csv("Honey.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   Year = col_integer(),
##   `State ANSI` = col_integer(),
##   watershed_code = col_integer(),
##   Value = col_number()
## )
```

```
## See spec(...) for full column specifications.
```

```
Deadout <- read_csv("Deadout.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   Year = col_integer(),
##   `State ANSI` = col_integer(),
##   watershed_code = col_integer(),
##   Value = col_number()
## )
## See spec(...) for full column specifications.
```

```
Price_per_lb <- read_csv("Price per lb.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   Year = col_integer(),
##   `State ANSI` = col_integer(),
##   watershed_code = col_integer(),
##   Value = col_double()
## )
## See spec(...) for full column specifications.
```

```
Production_per_Colony <- read_csv("Production per Colony.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   Year = col_integer(),
##   `State ANSI` = col_integer(),
##   watershed_code = col_integer(),
##   Value = col_double()
## )
## See spec(...) for full column specifications.
```

```
Honey_value_annual <- read_csv('Honey_value.csv')
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   Year = col_integer(),
##   Value = col_double()
## )
## See spec(...) for full column specifications.
```

```
CPI <- read_csv('1987_2017CPI.csv')
```

```
## Parsed with column specification:
## cols(
##   Year = col_integer(),
##   CPI = col_double()
## )
```

```
honey_loss_dt <- read.csv('Honey_Loss_6_States.csv')
```

```
#sort each data set to variables we want.
Honey <- dplyr::select(Honey,Year,State,Value)
Deadout <- dplyr::select(Deadout,Year,Period,State,Value)
Price_per_lb <- dplyr::select(Price_per_lb,Year,State,Value)
Production_per_Colony <- dplyr::select(Production_per_Colony,Year,State,Value)
Honey_value_annual <- dplyr::select(Honey_value_annual,Year,Value)



#Filter out totals, group each variable by state, average values from each year
Production <- filter(Honey, State != "US TOTAL") %>% group_by(State) %>% summarise(Average_production =
Loss <- filter(Deadout, State != "US TOTAL") %>% group_by(State) %>% summarise(Average_loss = mean(Valu
Price <- filter(Price_per_lb, State != "US TOTAL") %>% group_by(State) %>% summarise(Average_price = mea
Colony_production <- filter(Production_per_Colony, State != "US TOTAL") %>% group_by(State) %>% summaris

#Combine into one set. Add new variables to show number of colonies and loss/colony
Honey_by_State <- full_join(Production, Loss, by = "State")
Honey_by_State <- full_join(Honey_by_State, Price, by = "State")
Honey_by_State <- full_join(Honey_by_State, Colony_production, by = "State") %>%
  mutate(Colonies = Average_production*2000/Average_per_colony) %>%
  mutate(Loss_per_colony = Average_loss/Colonies)

# pick 6 top states with highest production and complete data
Honey_State <- Honey %>%  group_by(State)
unique(Honey$State)
```

```
##  [1] "US TOTAL"       "ALABAMA"        "ARIZONA"        "ARKANSAS"
##  [5] "CALIFORNIA"     "COLORADO"       "FLORIDA"        "GEORGIA"
##  [9] "HAWAII"         "IDAHO"          "ILLINOIS"       "INDIANA"
## [13] "IOWA"           "KANSAS"         "KENTUCKY"       "LOUISIANA"
## [17] "MAINE"          "MICHIGAN"       "MINNESOTA"      "MISSISSIPPI"
## [21] "MISSOURI"       "MONTANA"        "NEBRASKA"       "NEW JERSEY"
## [25] "NEW YORK"       "NORTH CAROLINA" "NORTH DAKOTA"   "OHIO"
## [29] "OREGON"         "OTHER STATES"   "PENNSYLVANIA"   "SOUTH CAROLINA"
## [33] "SOUTH DAKOTA"   "TENNESSEE"      "TEXAS"          "UTAH"
```

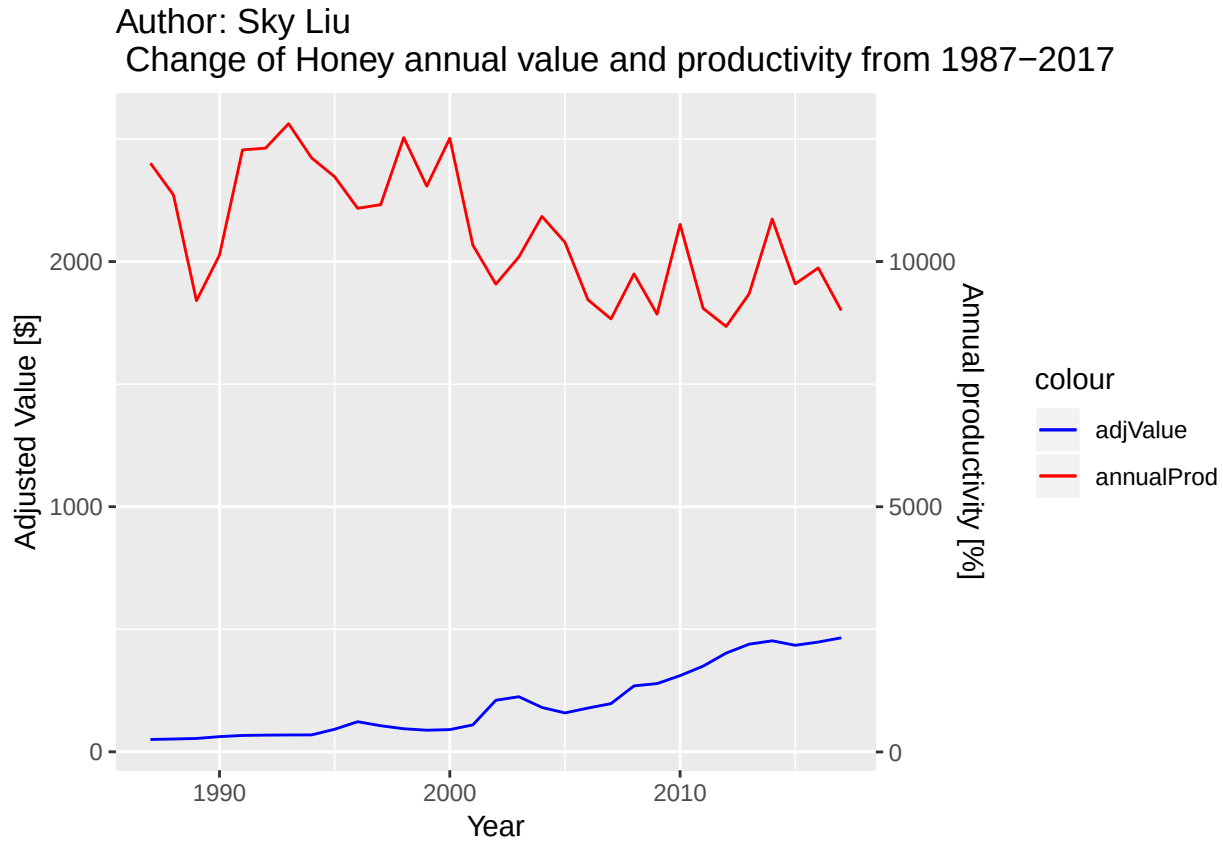
```
## [37] "VERMONT"        "VIRGINIA"       "WASHINGTON"     "WEST VIRGINIA"
## [41] "WISCONSIN"      "WYOMING"        "NEW MEXICO"     "NEVADA"
## [45] "MARYLAND"       "OKLAHOMA"       "CONNECTICUT"    "DELAWARE"
## [49] "MASSACHUSETTS"  "NEW HAMPSHIRE"  "RHODE ISLAND"
```

```
Honey_sixstate <- Honey_State %>%
  filter(State %in% c("CALIFORNIA", "FLORIDA" ,"SOUTH DAKOTA", "NORTH DAKOTA","MONTANA", "MINNESOTA")) %
  arrange(State, Year)

#Honey lost in 6 states
#Sum by year (Since we only have the data in 1st and 2nd quarter in 2018, we will exclude the data in 2
honey_loss_dt$Value <- as.numeric(gsub(",","",honey_loss_dt$Value))
honey_2017 <- honey_loss_dt %>% select(Year,State,Value) %>% filter(Year==2017) %>% group_by(Year,State)
honey_2016 <- honey_loss_dt %>% select(Year,State,Value) %>% filter(Year==2016) %>% group_by(Year,State)
honey_2015 <- honey_loss_dt %>% select(Year,State,Value) %>% filter(Year==2015) %>% group_by(Year,State)
# Total loss from 2015-2017
honey_total <- rbind(honey_2017,honey_2016,honey_2015)


#Adjust the annual honey value (price received) by 1987 inflation rate.
baseCPI <- rep(113.6, 21)
adjusted_Price <- as.data.frame(Honey_value_annual$Value * (CPI$CPI / baseCPI))
```

```
## Warning in CPI$CPI/baseCPI: longer object length is not a multiple of
## shorter object length
```

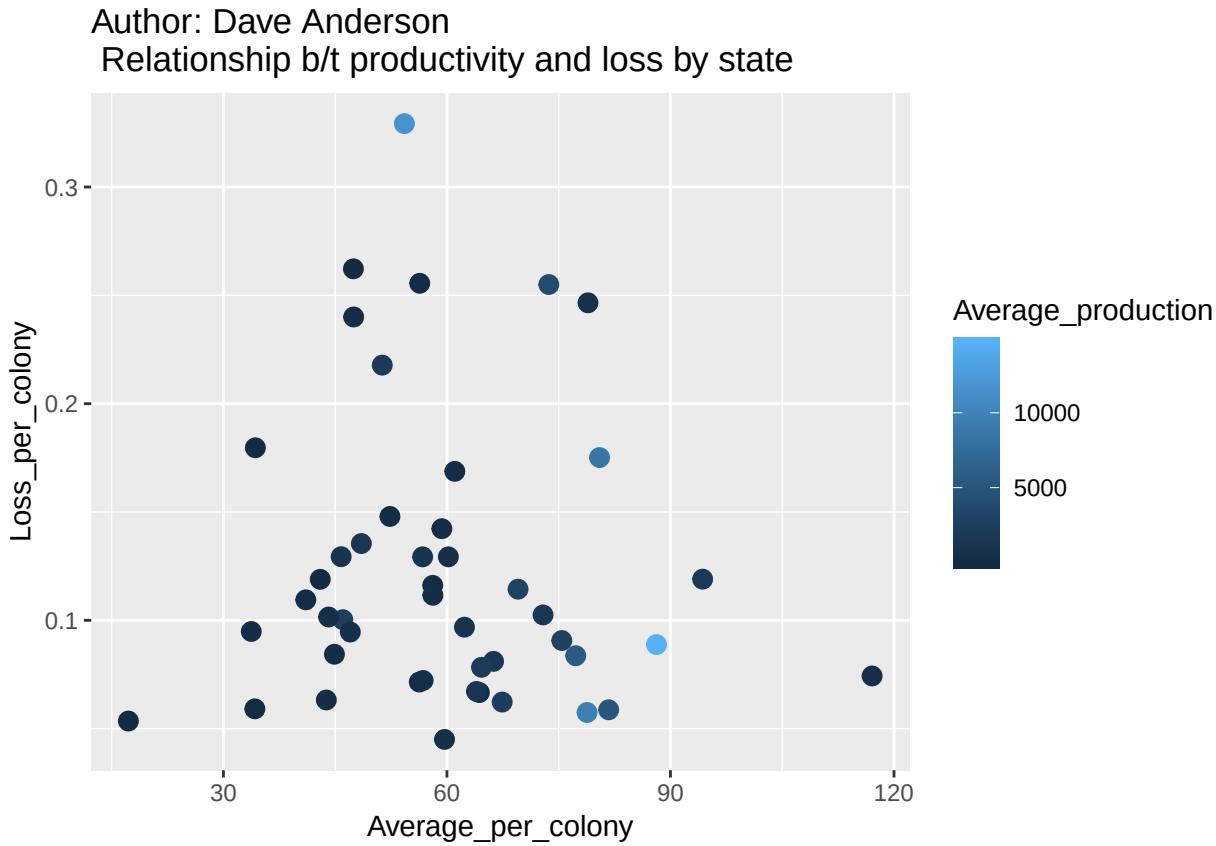
```
Honey_value_annual <- cbind(Honey_value_annual,adjusted_Price)
names(Honey_value_annual) <- c('Year','Value','adjValue')
#Add annual productivity
Annual_production <- filter(Honey, State != "US TOTAL") %>% group_by(Year)  %>% summarise(Average_produ
Annual_production <- arrange(Annual_production, desc(Year))
Honey_value_annual <- cbind(Honey_value_annual,Annual_production$Average_production)
names(Honey_value_annual) <- c('Year','Value','adjValue','annualProd')
```

```
#Change of Honey annual value and productivity from 1987-2017
ggplot(data = Honey_value_annual, aes(x = Year)) +
  geom_line(aes(y = adjValue, colour = "adjValue"))+
  geom_line(aes(y = annualProd, colour = "annualProd"))  +
  ggtitle("Author: Sky Liu \n Change of Honey annual value and productivity from 1987-2017")+
  scale_y_continuous(sec.axis = sec_axis(~.*5, name = "Annual productivity [%]"))+
  scale_colour_manual(values = c("blue", "red"))+
  labs(y = "Adjusted Value [$]",
                x = "Year")
```

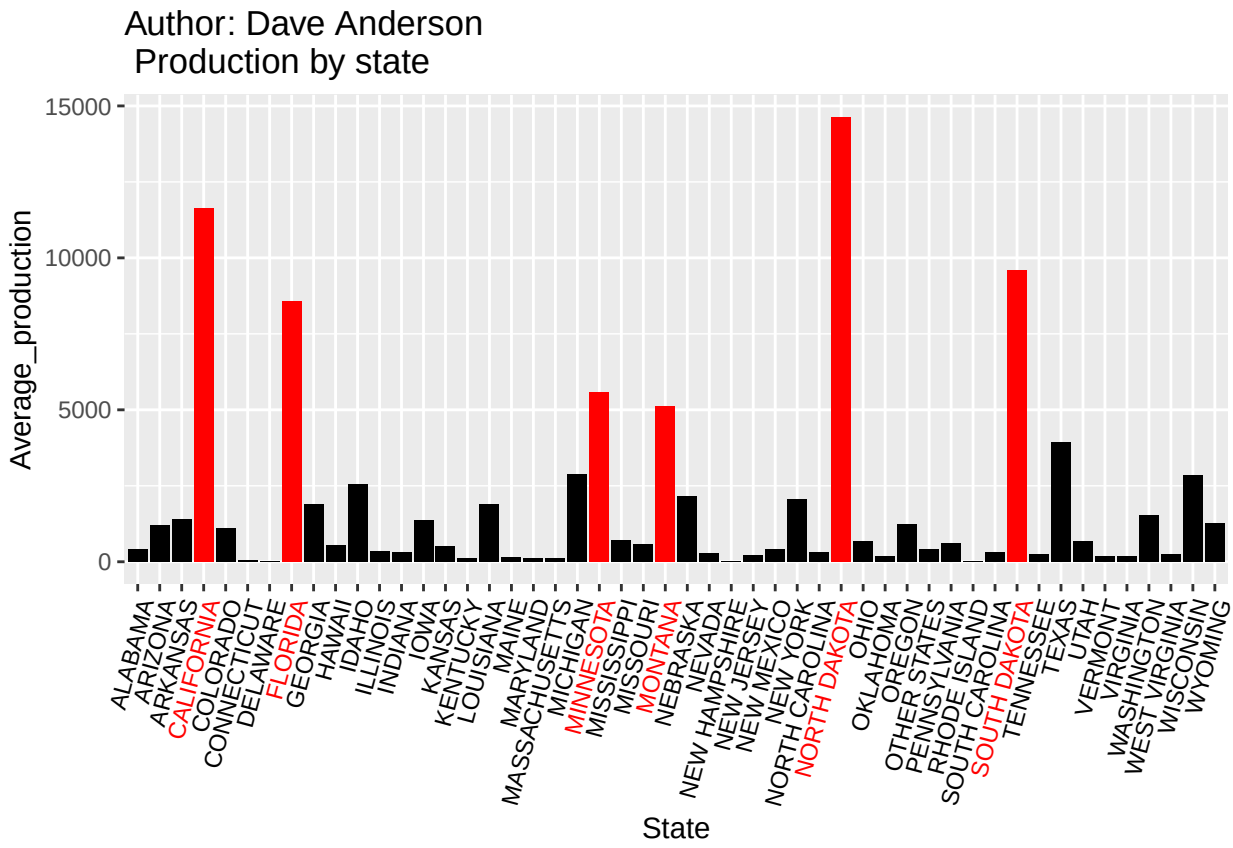



```
#Scatterplot to show relationship between productivity and loss by state.
#Colored to show overall high-producing states.
ggplot(data = Honey_by_State, mapping = aes(Average_per_colony,Loss_per_colony))+
  geom_point(aes(color = Average_production),size = 3)+
  ggtitle("Author: Dave Anderson \n Relationship b/t productivity and loss by state")
```

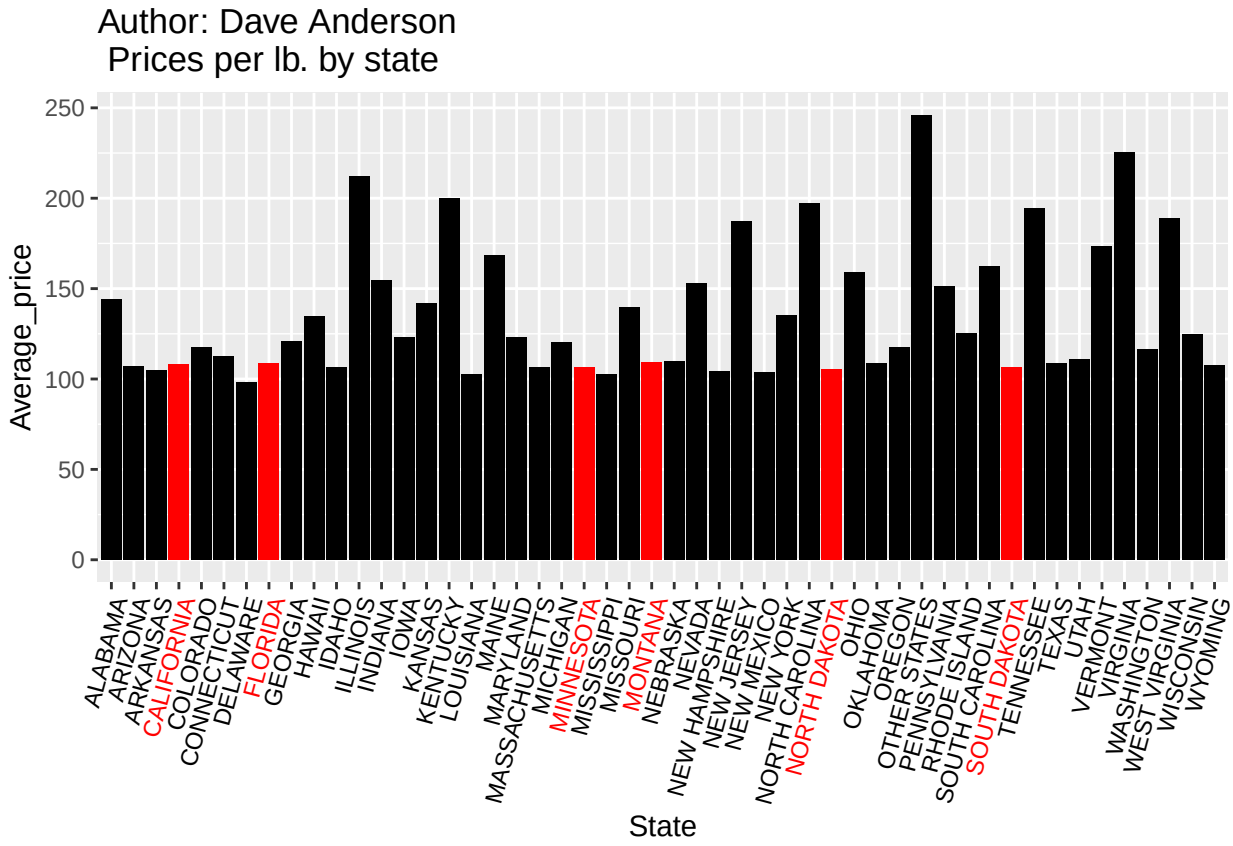
```
## Warning: Removed 4 rows containing missing values (geom_point).
```



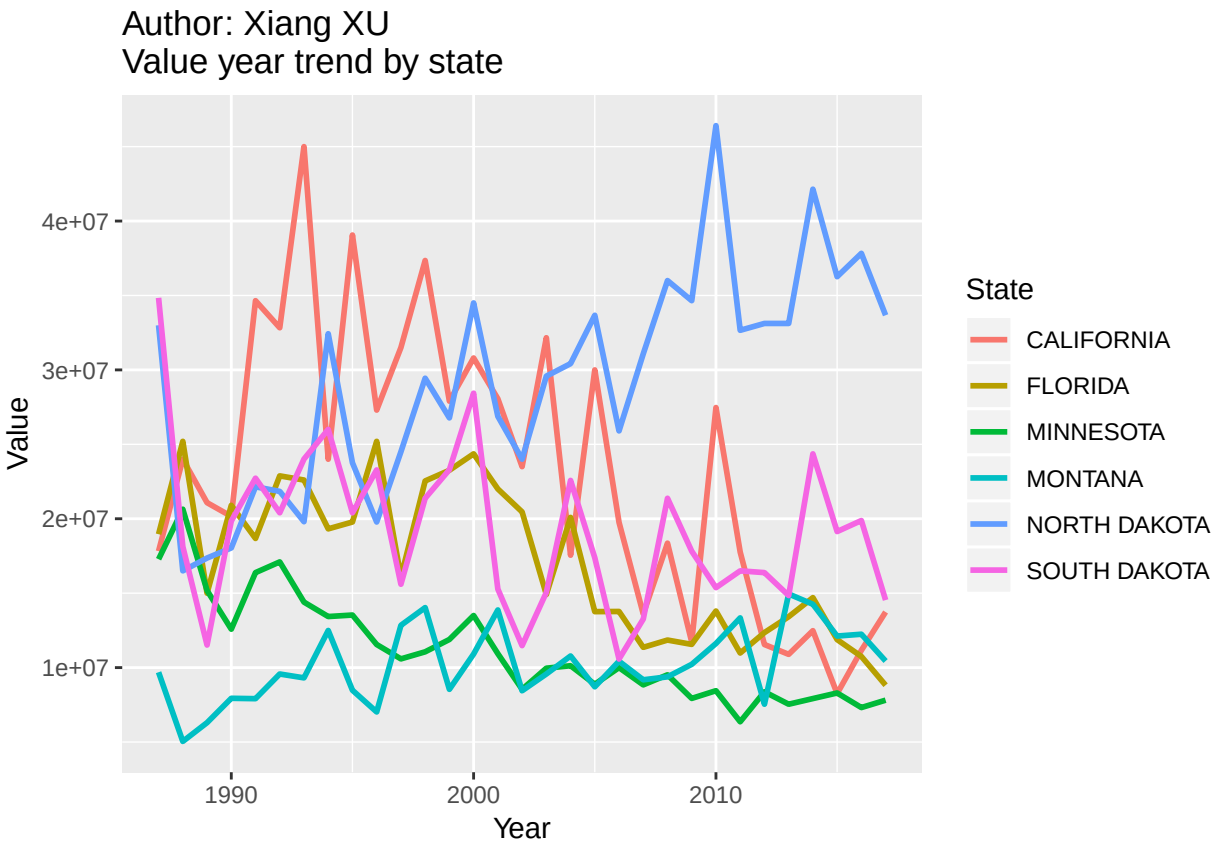

```
#Total Production by state, largest six states indicated in red.
ggplot(data = Honey_by_State, mapping = aes(State, Average_production, fill=ifelse(Average_production >
  geom_col()+
  scale_fill_manual(guide=FALSE, values=c("red", "black"))+
  theme(axis.text.x = element_text(color = ifelse(Honey_by_State$Average_production > 5000, "red", "bla
  ggtitle("Author: Dave Anderson \n Production by state")
```



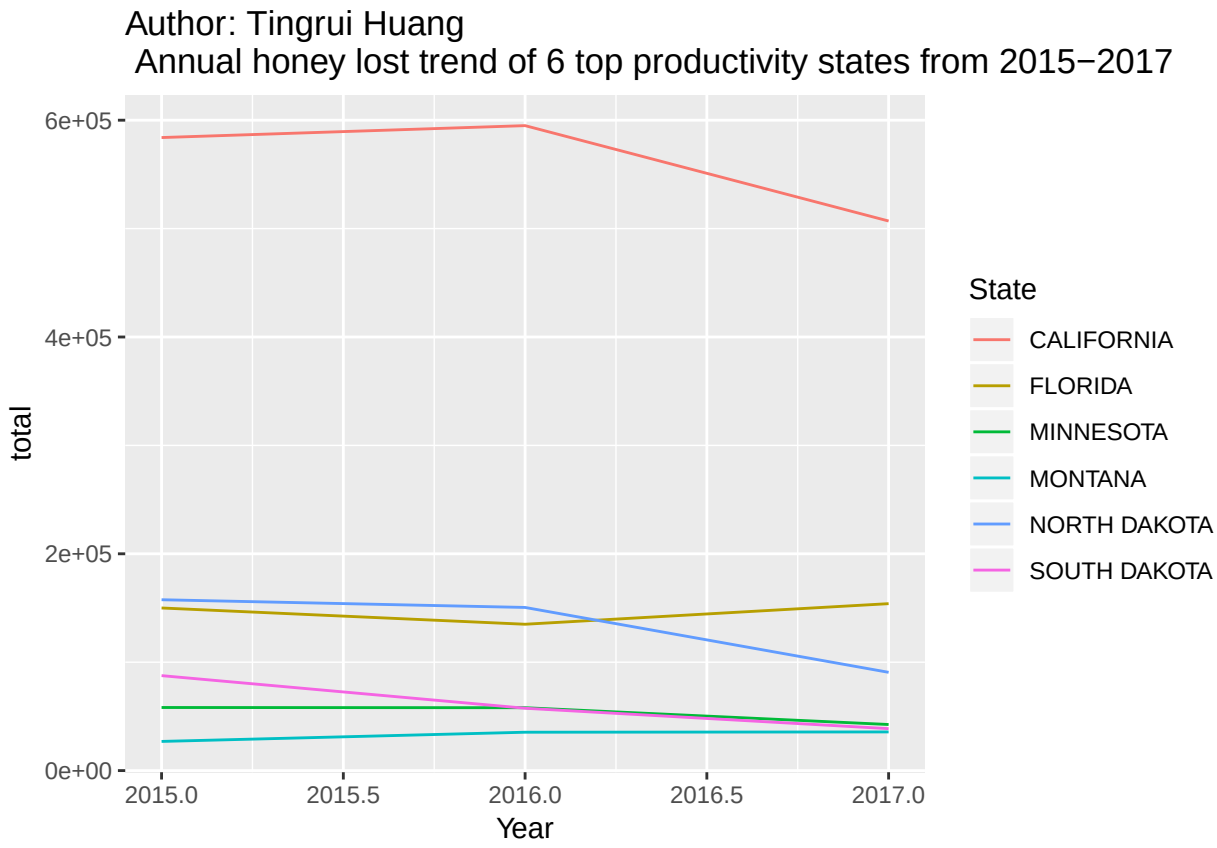

```
#Display of prices per lb. by state. Top six producing states still in red to show their low prices.
ggplot(data = Honey_by_State, mapping = aes(State ,Average_price, fill = ifelse(Average_production > 500
  geom_col()+
  scale_fill_manual(guide=FALSE, values=c("red", "black"))+
  theme(axis.text.x = element_text(color = ifelse(Honey_by_State$Average_production > 5000, "red", "bla
  ggtitle("Author: Dave Anderson \n Prices per lb. by state")
```




```
#trend of annual value of 6 top productivity states
ggplot(Honey_sixstate, aes(x=Year, y=Value ,color = State)) +
  geom_line(size = 1) +
  ggtitle("Author: Xiang XU \nValue year trend by state")
```



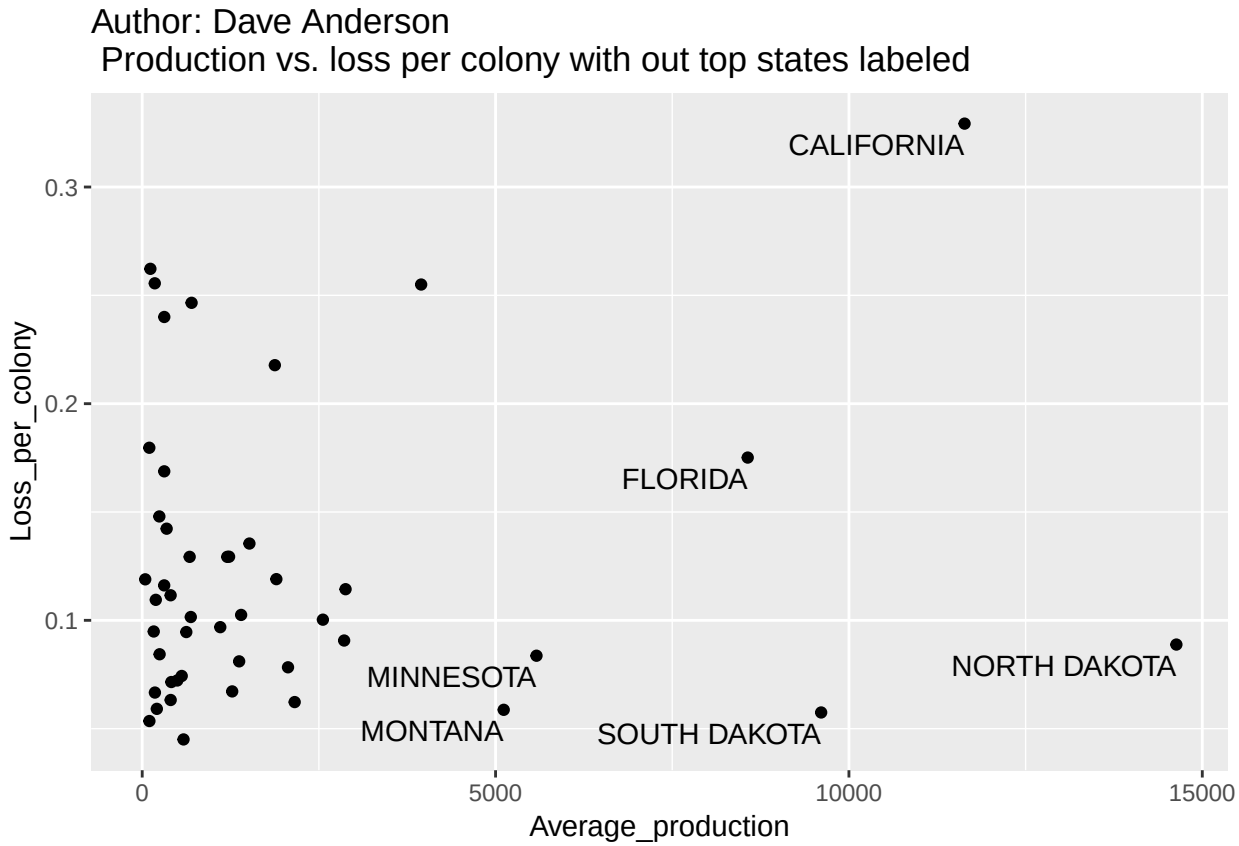

```
# annual honey lost trend of 6 top productivity states from 2015-2017
ggplot(honey_total, aes(x=Year, y=total, color=State))+geom_line()+
  ggtitle("Author: Tingrui Huang \n Annual honey lost trend of 6 top productivity states from 2015-2017
```



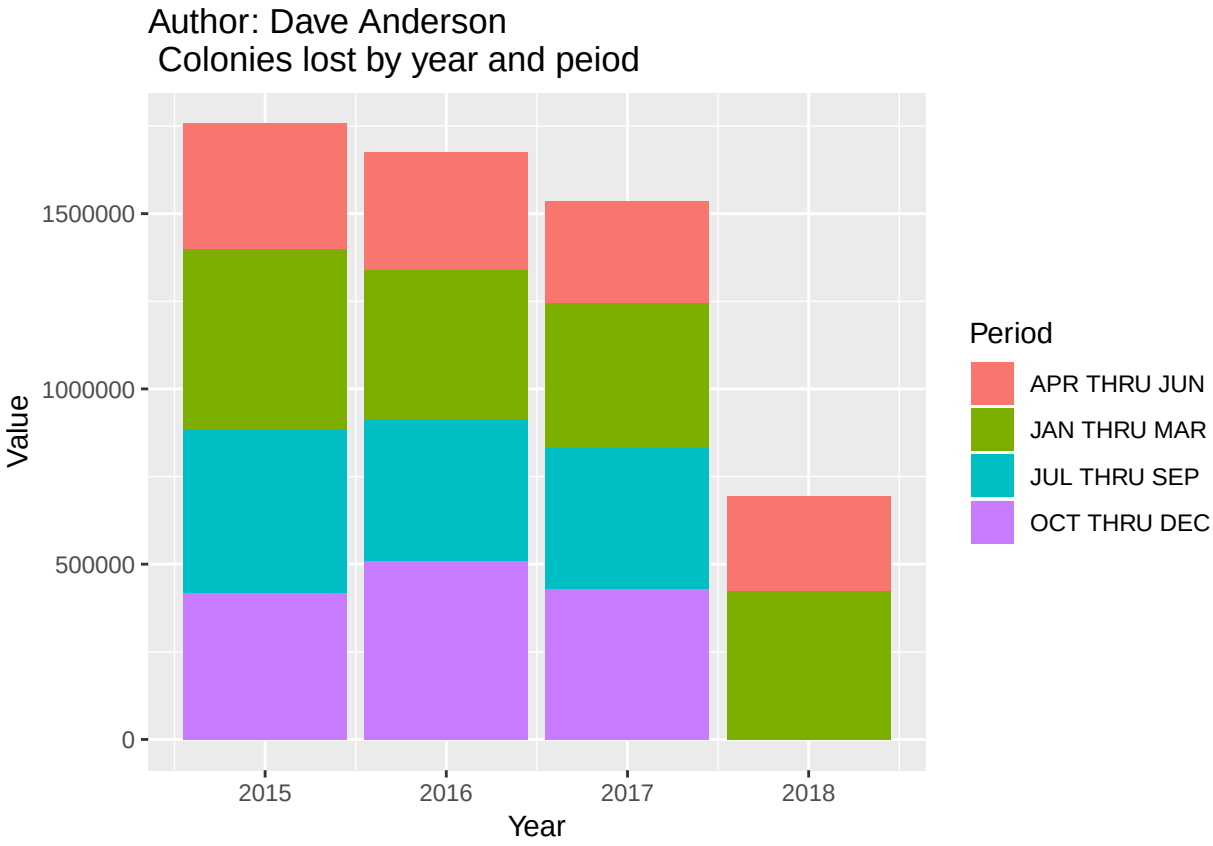

```
#Display of production vs. loss per colony with out top states labeled.
ggplot(data = Honey_by_State, mapping = aes(Average_production,Loss_per_colony, label = State))+
  geom_point() + geom_text(aes(label = ifelse(Average_production > 5000,as.character(State),'')),vjust =
   ggtitle("Author: Dave Anderson \n Production vs. loss per colony with out top states labeled")
```

```
## Warning: Removed 4 rows containing missing values (geom_point).
```

```
## Warning: Removed 4 rows containing missing values (geom_text).
```




```
#Display of colonies lost by year and peiod.
loss_by_year <- Deadout %>% filter(State == "US TOTAL") %>% group_by(Year)
ggplot(data = loss_by_year)+
  geom_col(mapping = aes(x = Year, y = Value, fill = Period))+
  ggtitle("Author: Dave Anderson \n Colonies lost by year and peiod")
```



It is interesting to see two of our top producers from, big, southern,

costal states while the other 4 are from the midwest.

Looking at prices, we see that the big producers are also among cheapest states.

The two large states have high rates of deadout colonies. The 4 big producers from the midwest have low loss rates.

We pick six states, which have 31 years data, CALIFORNIA, FLORIDA , SOUTH DAKOTA, NORTH DAKOTA, MONTANA and MINNESOTA. Accordng to the graph above, we can see the trend about annual value of each state for 31 years.

Clearly, NORTH DAKOTA is the only state among these six states, whose yearly value has a upward trend in last 31 years. The other five states decline volatility in year-values.

As we can see in the graph, California has the highest loss all the time, but there was a decrease

from 2016 to 2017. Florida and North Dakota has very close loss in 2015 and the begining of 2016,

however, North Dakota has decreased its loss by 35% from 2016 to 2017 and it now has lower loss

than Florida. We won't be suprised if there is an decrease in the Price of honey and increase in the

Production of honey in North Dakota in the year of 2016-2017. The other three states have steady

low loss comparing with California, Florida and North Dakota.