Midterm Porject

CJ (With Vivian)

10/24/2016

Introduction

County-level data from oil and/or natural gas producing States—for onshore production in the lower 48 States only—are compiled on a State-by-State basis. Most States have production statistics available by county, field, or well, and these data were compiled at the county level to create a database of county-level production, annually for 2000 through 2011. Raw data for natural gas is for gross withdrawals, and oil data almost always include natural gas liquids. Note that State-provided natural gas withdrawals were not available for Illinois or Indiana; those estimates were produced using geocoded wells and State total production reported by the U.S. Department of Energy's Energy Information Agency. The raw dataset is not tidy with many columns that can be combined and reshaped into rows. Some columns are less useful than others because of redundancy issues. My prior interest lies in making the dataset tidy with appropriate use of data wrangling and subsetting tools, and provide clients with data visualization by using ggplot2 package.

Data Cleaning and Preparation

Our dataset is messy in terms of organization. Many variables that belong to the same categories can be combined into one column with different observations as rows. Two classic examples are Oil withdrawals and Gas withdrawals. In the following code, I will reorganize the dataset using tidyr and dplyr packages, delete redundant columns, and eventually obtain the tidydata.csv.(Of course, save it for later use.)

```
library(tidyr)
library(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
```

```
library(ggplot2)
library(ggthemes)
library(plotrix)

# Data Cleaning  ------------------------------------------------------------

#load data
data.raw <- read.csv("oilgascounty.csv")

#convert data type to data frame
```

```
data0 <- data.frame(data.raw)

#gather oil data from 2000 to 2011. Note I create two new columns named year and oidwithdraw
data.oil0 <- gather(data0, year, oilwithdraw, oil2000:oil2011)
data.oil <- mutate(data.oil0, year=(gsub("oil","",year)))

#gather gas data from 2000 to 2011. Note I create two new columns named year2 and gaswithdraw
data.gas0 <- gather(data.oil, year2, gaswithdraw, gas2000:gas2011)
data.gas <- mutate(data.gas0, year2=(gsub("gas","",year2)))

#delete verbose rows where year is not equal to year2
data1 <- filter(data.gas, year == year2)

#delete verbose column year2 and geoid
data <- select(data1, -year2, -geoid)

#Swap columns to make data more self-describing
Tidydata <- data[,c(1:7,11:13,8:10)]

#Save Tidydata.csv file for futher data Exploration
write.csv(Tidydata, 'Tidydata.csv')
```

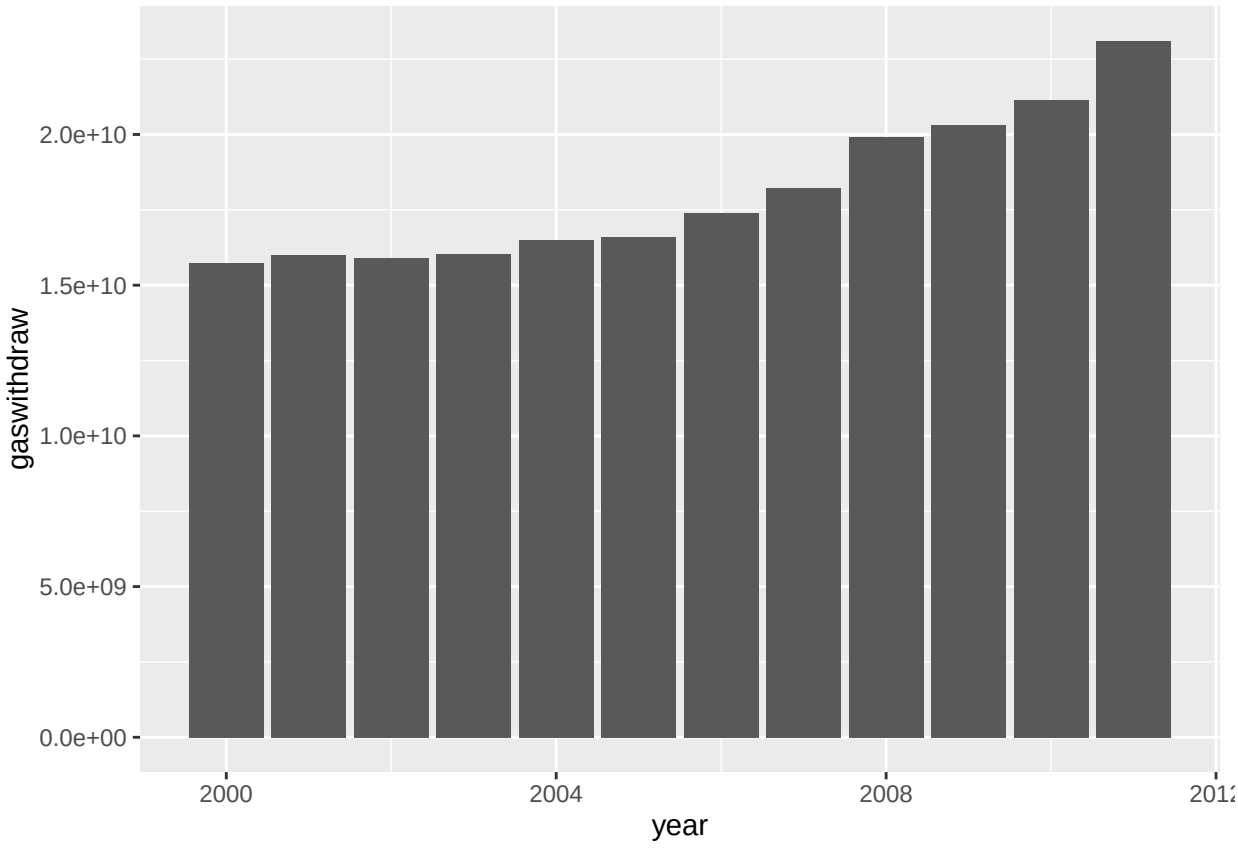
Data Visulization and Exploration

Now our data is tidy. We need to load it back up and explore it a little bit more. A brand new column has been introduced when I loaded back tidy data, and thus I deleted that X column so that my tidy data is consistent. Now, let us do some graphics starting with bar diagrams.

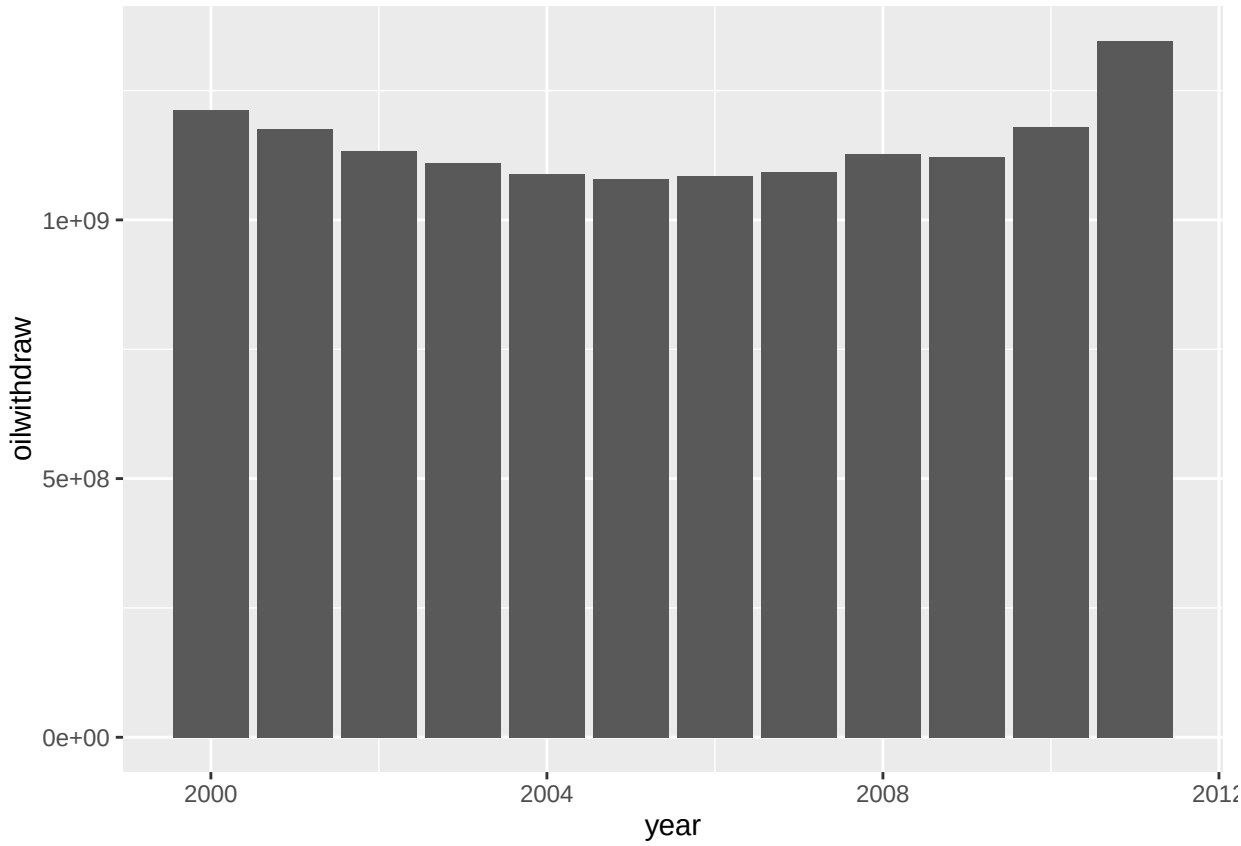
Sample 1

Sample 1 shows Oil withdrawals and Gas withdrawals in each year, illustrated by ggplot() bar diagrams.

```
df <- read.csv('Tidydata.csv')

# Delete the unnecessary new column
dfnew <- select(df,-X)

# Data Visualization ----------------------------------------------------------

# Oidwithdraw for each year in bar diagram
a <- ggplot(dfnew, aes(x = year, y = gaswithdraw))
a + geom_bar(stat = "identity")
```



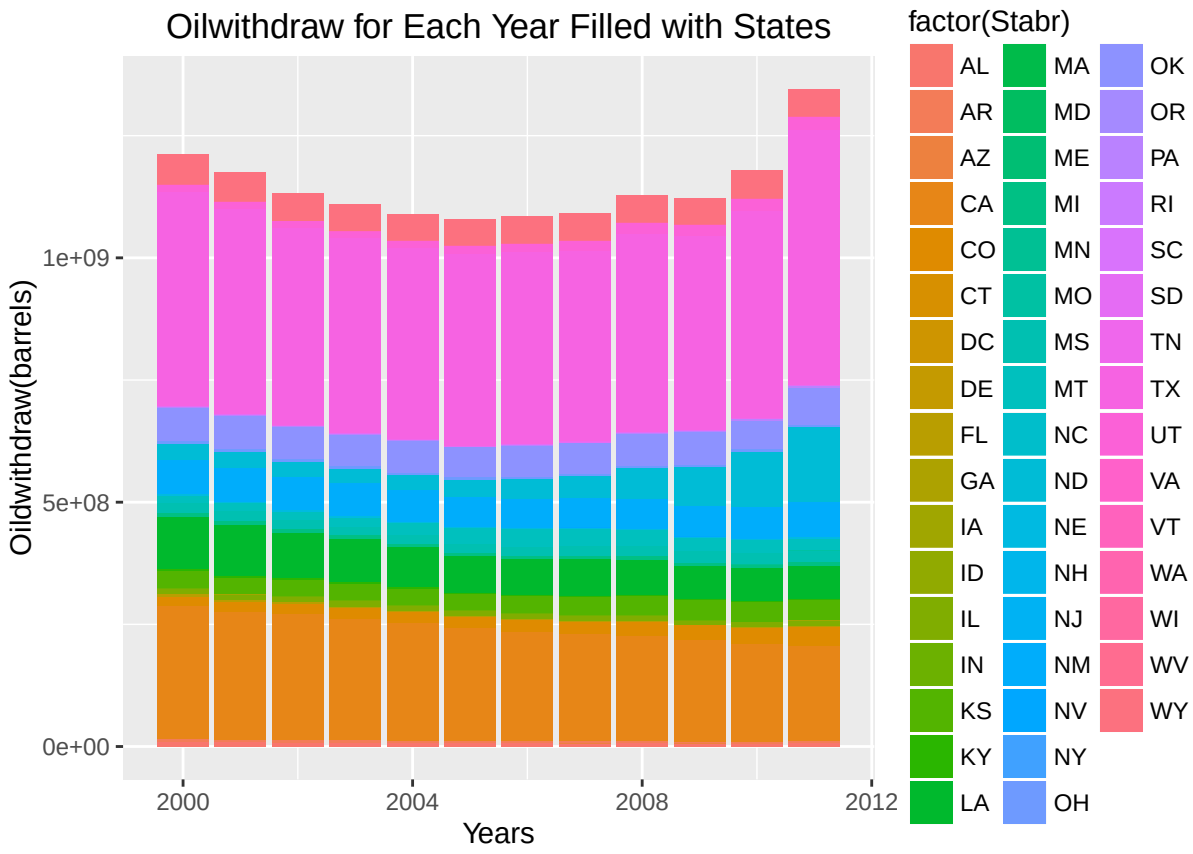

```
# Gaswithdraw for each year in bar diagram
b <- ggplot(dfnew, aes(x = year, y = oilwithdraw))
b + geom_bar(stat = "identity")
```



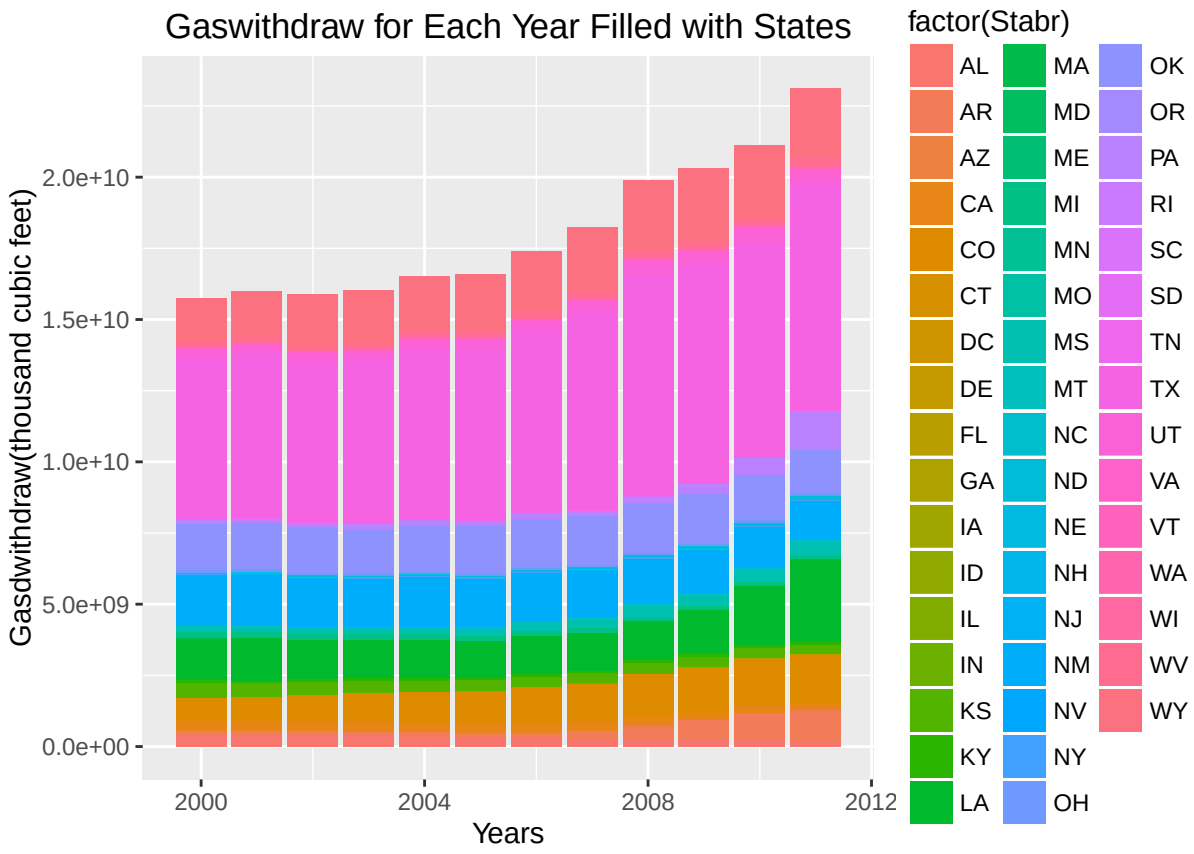
Sample 2

Sample 2 is a step further than 1. It uses Year as x-axis, Gas withdrawals or Oil withdrawals as Y-axis, filled different states as factors, with titles and themes on the picture.

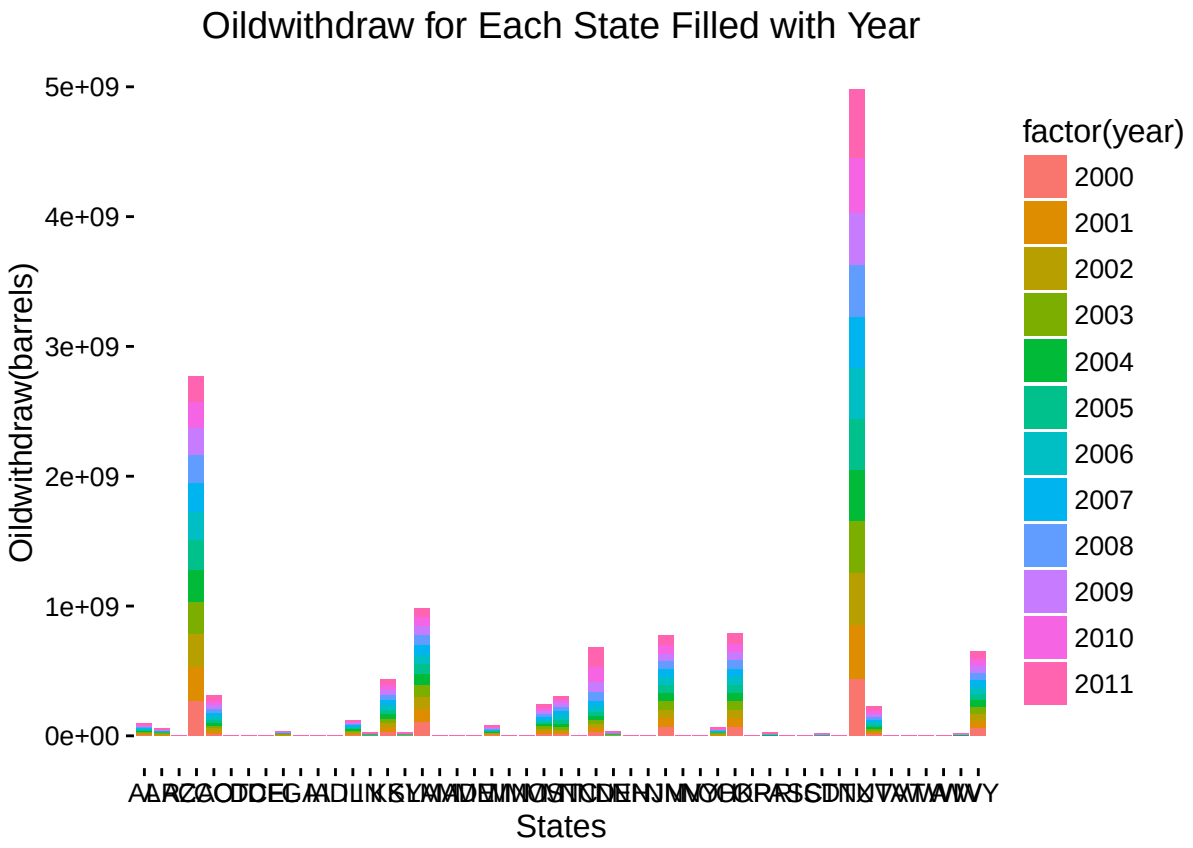
```
# Each year's oilwithdraw filled with States
ab <- ggplot(dfnew, aes(x = year, y = oilwithdraw, fill = factor(Stabr))) + geom_bar(stat = "identity")
ab + labs(title = "Oilwithdraw for Each Year Filled with States", x = "Years", y = "Oildwithdraw(barrels
```



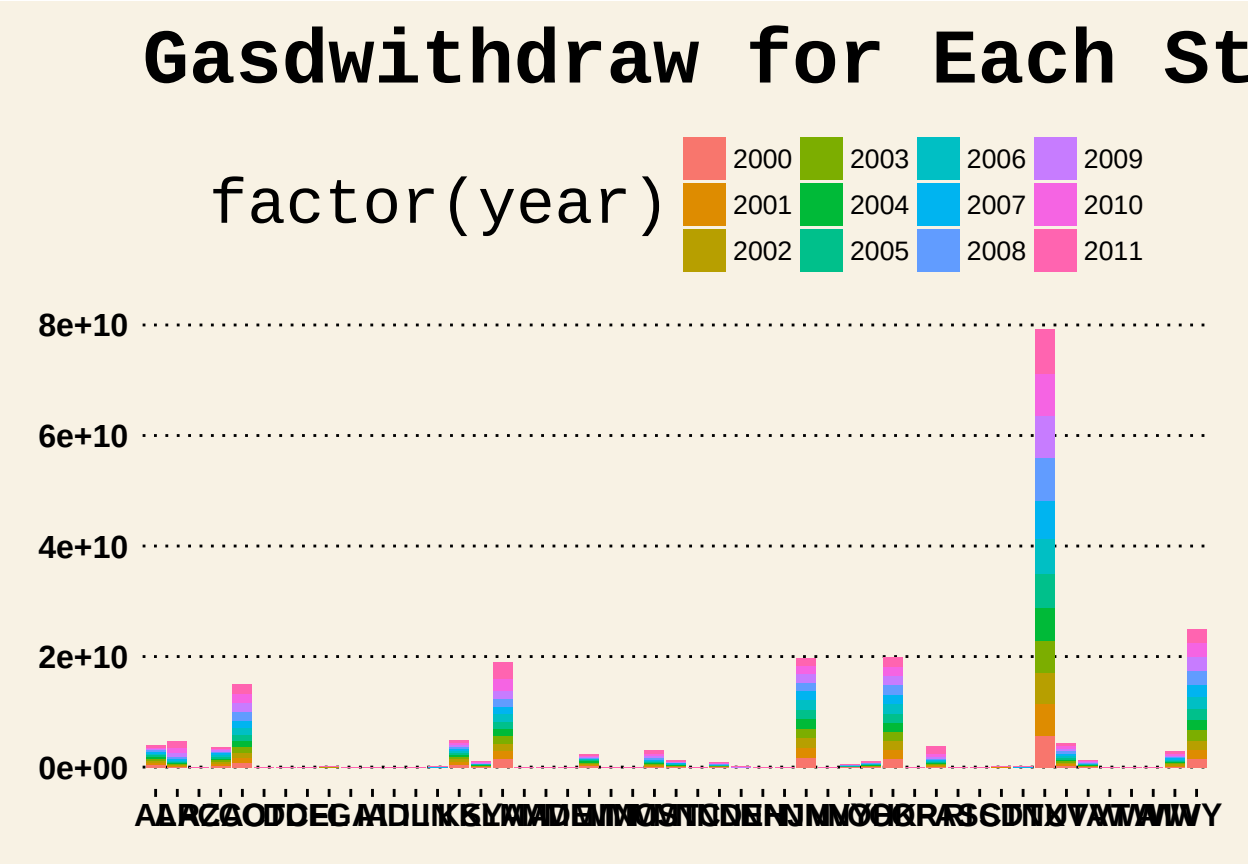

```
# Each year's gaswithdraw filled with States
ab <- ggplot(dfnew, aes(x = year, y = gaswithdraw, fill = factor(Stabr))) + geom_bar(stat = "identity")
ab + labs(title = "Gaswithdraw for Each Year Filled with States", x = "Years", y = "Gasdwithdraw(thousa
```




```
# Each state's Oilwithdraw filled with year
c <- ggplot(dfnew, aes(x = Stabr, y = oilwithdraw, fill = factor(year))) + geom_bar(stat = "identity")
c + labs(title = "Oildwithdraw for Each State Filled with Year", x = "States", y = "Oildwithdraw(barrels
  theme_classic()
```




```
# Each state's gasdwithdraw filled with year
c <- ggplot(dfnew, aes(x = Stabr, y = gaswithdraw, fill = factor(year))) + geom_bar(stat = "identity")
c + labs(title = "Gasdwithdraw for Each State Filled with Year", x = "States", y = "Gaswithdraw(thousand
  theme_wsj()
```

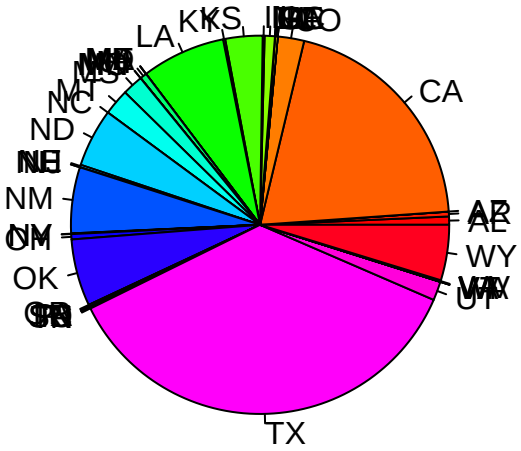


Sample 3

Sample 3 is quite fun becuase I get a chance to use pipe, which is widely-used in the circle of R. Sorting, summarizing, manipulating Data frames are also discussed. Two pie charts are used to generate graphics. One demonstrates top 10 oil production states, and the other with gas. I could have include more states in pie chart, but problems arise when a few states barely produce any oil or gas. Pie charts can easily get clumstered if we decide not to select fewer states, which defeats the purpose of showing user-friendly graphs.

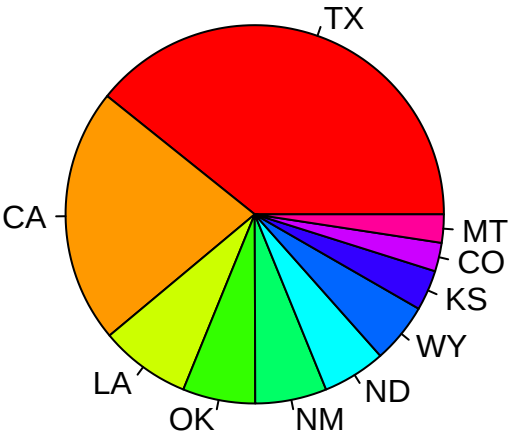
```
# Present top 10 biggest oil production states in piechart
newstatesoil <- dfnew %>% group_by(Stabr) %>% summarize(sum_oil = sum(as.numeric(oilwithdraw)))
pie(newstatesoil$sum_oil, labels = newstatesoil$Stabr, col = rainbow(length(newstatesoil$Stabr)),main =
```

Pie chart of country

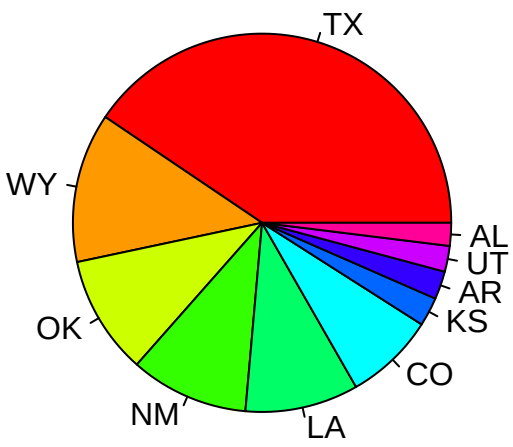



```
# We find that the pie chart is too dense. Let's list top 10 states, and its relative pie chart
newstatedata <- newstatesoil[order(-newstatesoil$sum_oil),]
# Now, here comes the top 10 states in oilwithdraw
Cleanstatedata <- newstatedata[1:10,]
pie(Cleanstatedata$sum_oil, labels = Cleanstatedata$Stabr, col = rainbow(length(Cleanstatedata$Stabr)),
```

Pie chart of country




```
# Present top 10 biggest gas production states in piechart
newstatesgas <- dfnew %>% group_by(Stabr) %>% summarize(sum_gas = sum(as.numeric(gaswithdraw)))
newstatedata1 <- newstatesgas[order(-newstatesgas$sum_gas),]
# Now, here comes the top 10 states in gaswithdraw
Cleanstatedata1 <- newstatedata1[1:10,]
pie(Cleanstatedata1$sum_gas, labels = Cleanstatedata1$Stabr, col = rainbow(length(Cleanstatedata1$Stabr
```



Sample 4

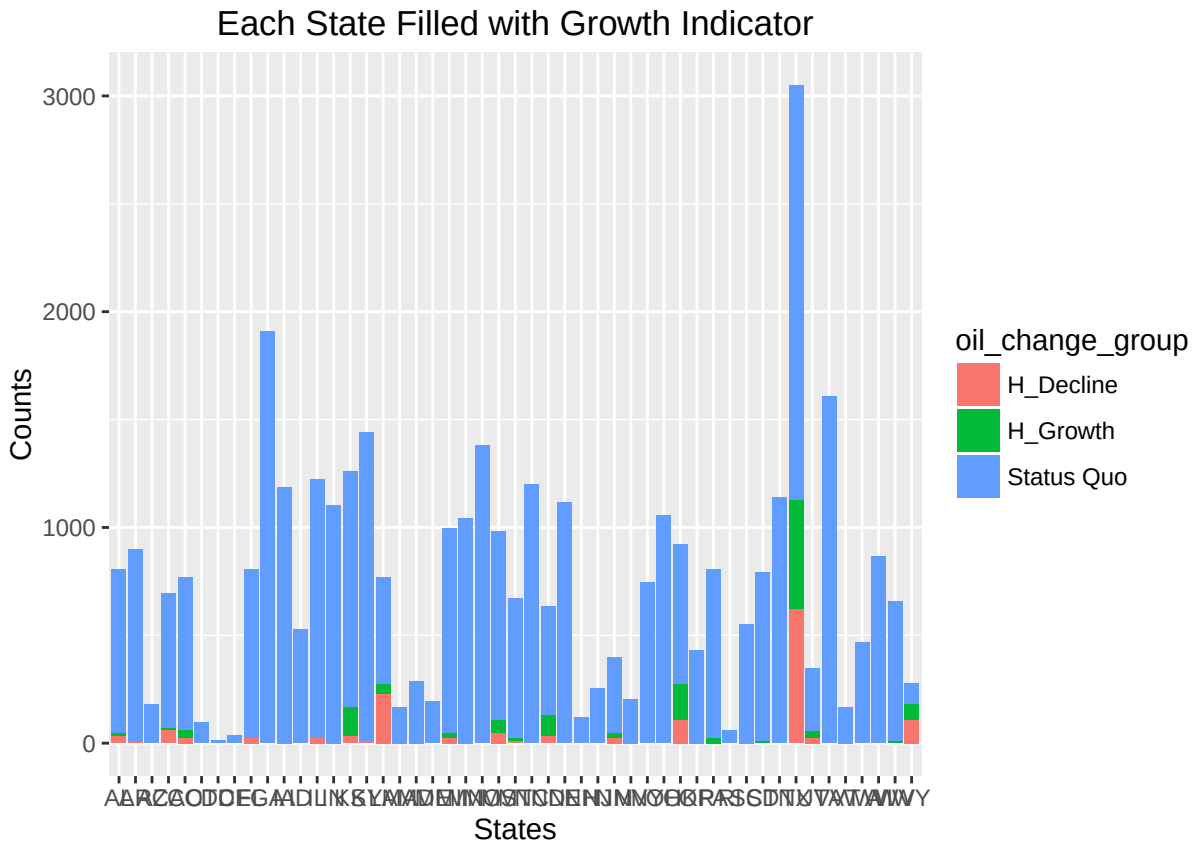
Sample 4 is short but quite significant becuase it illustrates how to merge relative datasets if both sets happen to have the same columns. Merge() is properly used. Now we obtain a new dataset waited for further data exploration and graphics after generating a new column that operated on other columns. Total is a new column in the dataset that sums up oil productions and gas productions. I decided not to explore on this dataset because gas and oil have distinct productions unit, which implies that they are not addable. However, I just want to showcase merge() and mutate() methods in package "dplyr"

```
# Merge data frames to get total gas and oil production
newstate_oil_gas = merge(newstatedata,newstatedata1)
oil_gas_sum <- mutate(newstate_oil_gas, total = sum_oil + sum_gas)
```

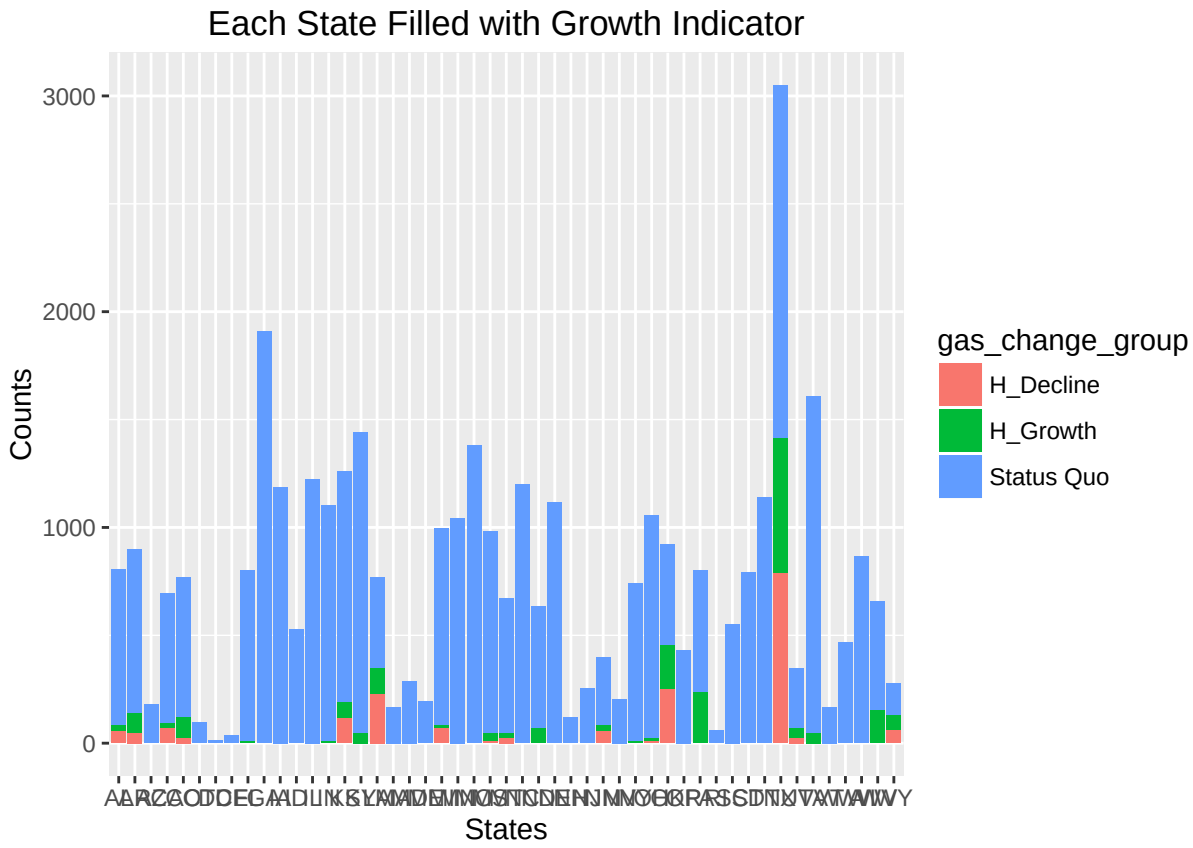
Sample 5

Oil_change_group is a categorical variable based upon change in the dollar value of oil production, 2000-11, Values are H Growth (>=$20 million),H Decline (<=-$20 million), Status Quo (change between +/- $20 million). Same principles appply to gas_change_group and oil_gas_change_group. I used different states as x-axis, filled count values with H Growth, H Decline, and Status Quo.

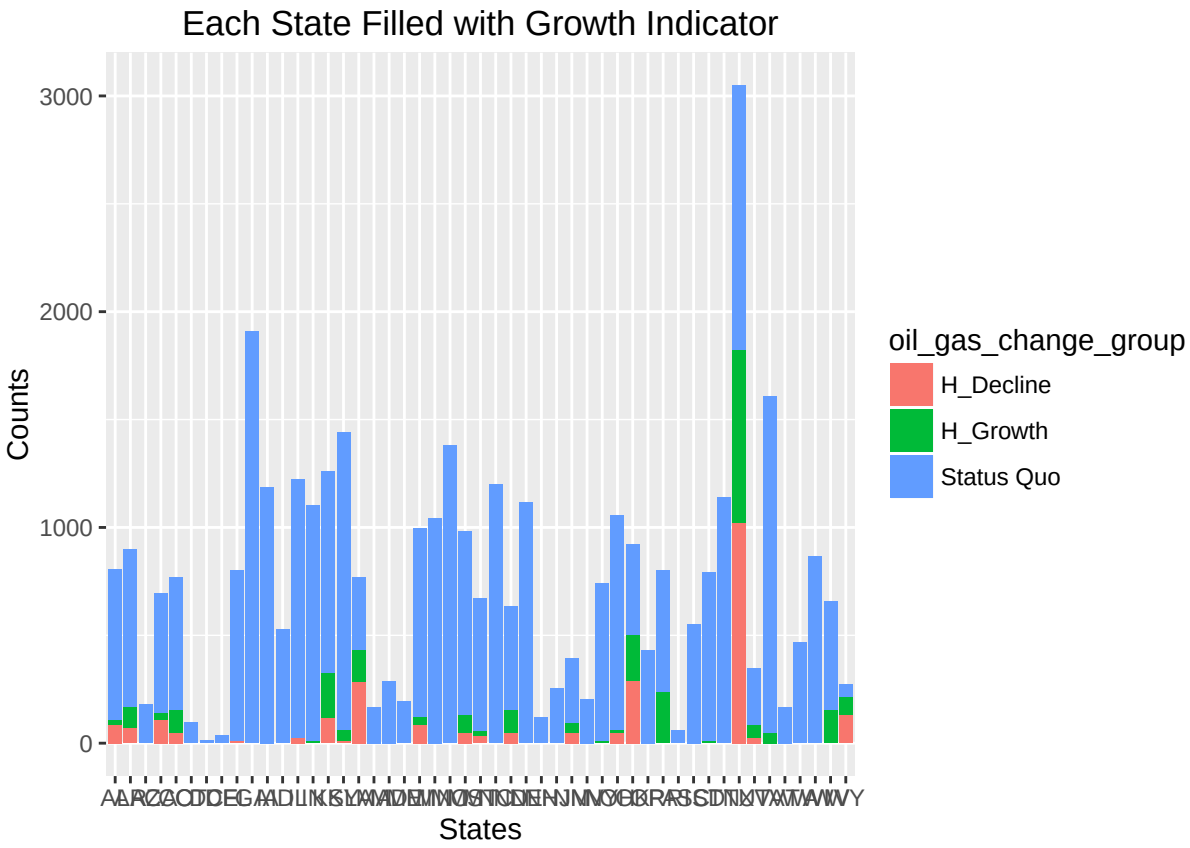
```
# Each State filled with categorical indicators
state_oil_growth<- ggplot(dfnew, aes(x=Stabr, fill=oil_change_group))+
  geom_bar() + labs(title = "Each State Filled with Growth Indicator", x = "States", y = "Counts")
state_oil_growth
```




```
state_gas_growth<- ggplot(dfnew, aes(x=Stabr, fill=gas_change_group))+
  geom_bar() + labs(title = "Each State Filled with Growth Indicator", x = "States", y = "Counts")
state_gas_growth
```



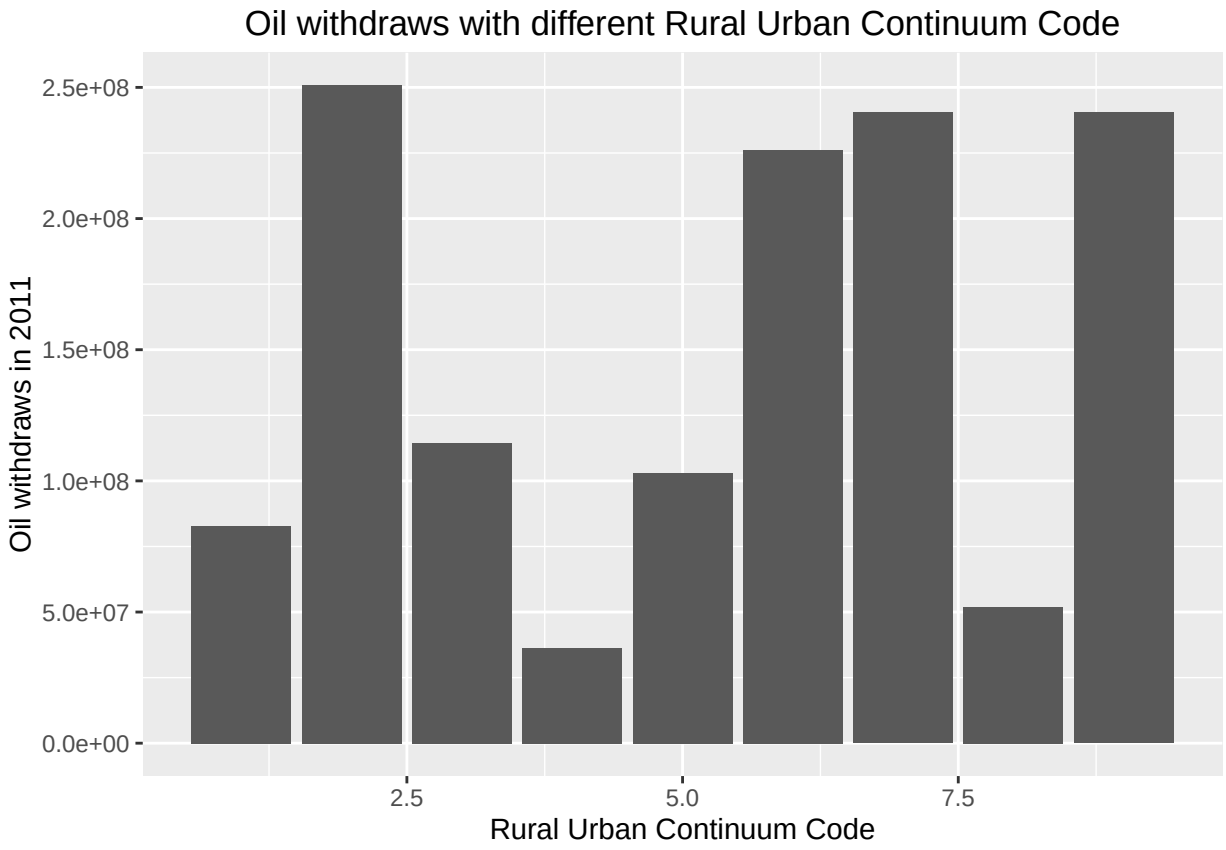

```
state_oil_gas_growth<- ggplot(dfnew, aes(x=Stabr, fill=oil_gas_change_group))+
  geom_bar() + labs(title = "Each State Filled with Growth Indicator", x = "States", y = "Counts")
state_oil_gas_growth
```



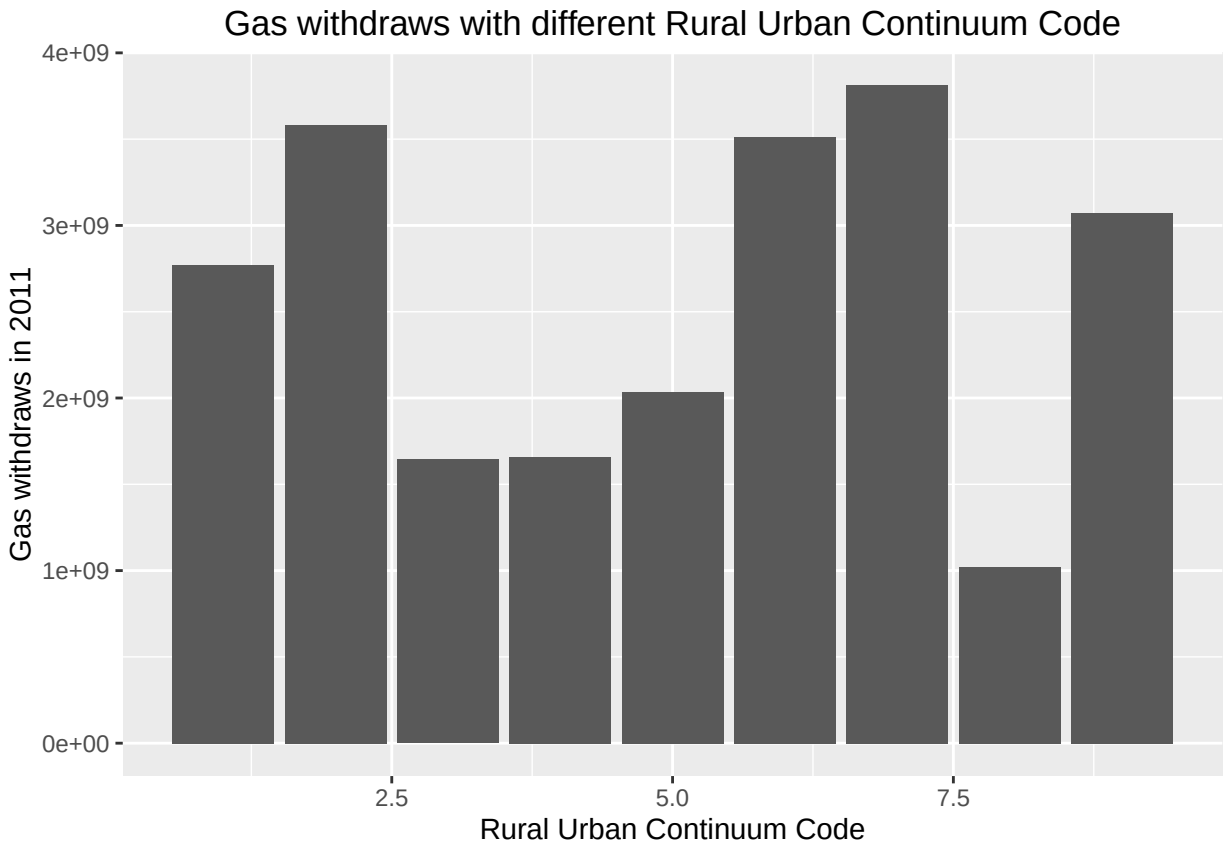
Sample 6

Sample 6 aims to demonstrate the connection between Rural Urban Continumm code and oil, gas withdrawals at year of 2011. Select() and Filter() methods are used to reduce sample size. Continum code 1-9 represent metro and non-metro areas with differet range of population. For more information, you can visit this link: http://www.ers.usda.gov/data-products/rural-urban-continuum-codes/documentation/

```
# Bar graphics indicating oil and gas withdraws corresponding to different Rural Urban Continuum Code a
dfnew.RUC <- select(dfnew, County_Name, Rural_Urban_Continuum_Code_2013, year:gaswithdraw)
dfnew.RUC2011 <- filter(dfnew.RUC, year == 2011)

gg <- ggplot(data = dfnew.RUC2011, aes(x = Rural_Urban_Continuum_Code_2013, y = oilwithdraw))
gg + geom_bar(stat = "identity") +
  labs(title = "Oil withdraws with different Rural Urban Continuum Code", x = "Rural Urban Continuum Co
```




```
gg <- ggplot(data = dfnew.RUC2011, aes(x = Rural_Urban_Continuum_Code_2013, y = gaswithdraw))
gg + geom_bar(stat = "identity") +
  labs(title = "Gas withdraws with different Rural Urban Continuum Code", x = "Rural Urban Continuum Co
```



References

Please note that my teammate is Vevian, and we have worked consistently on this project. I really appreciate her unique insights and contribution. The entire project would have not progressed this far without her dedication.