Data Pre-processing and Tidy Data
CPSC 260

For this assignment you will use the *crime_and_incarceration_altered.csv* data set that is provided on Bb. This data comes from Kaggle (and was altered for this assignment). You will perform the tasks and answer the questions below to practice pre-processing data. This assignment will require you to use *dplyr*, *tidyr*, and *ggplot2* which are all inside of our *tidyverse* package. So make sure to load this library first!

When finished you will submit this Word document containing your answers, R code, and copies of graphs. You will also submit your working R script file.

Information About the Kaggle Data Set:
https://www.kaggle.com/christophercorrea/prisoners-and-crime-in-united-states

The website says:

> *In 1975, the United States set a new record with 240,593 prisoners incarcerated by state or federal agencies. The United States achieved new record totals during each of the next 34 years. Today, there are over 1,500,000 prisoners in the United States. Over one quarter of the world's entire population of prisoners is located in the United States.*
>
> *The U.S. Education department reports state and local government expenditures on prisons (not reflected in this dataset) have increased about three times as fast as spending on elementary and secondary education during this time period.* ***Does this significant investment into imprisonment improve public safety?***
>
> *The Bureau of Justice Statistics administers the National Prisoners Statistics Program (NPS), an annual data collection effort that began in response to a 1926 congressional mandate. The Uniform Crime Report (UCR) has served as the FBI's primary national data collection tool since a 1930 congressional mandate directed the Attorney General to "acquire, collect, classify, and preserve identification, criminal identification, crime, and other records."*

1. Bring the data set into R. Then create a subset of the data containing only the variables listed below for the states in the Midwest: Iowa, Illinois, North Dakota, South Dakota, Minnesota, Wisconsin, Nebraska, Kansas, Missouri, Indiana, Michigan, and Ohio. Is the data, by definition, "tidy"? Explain why or why not.

Variables of interest:

- jurisdiction
- year
- crimes_estimated
- state_population
- prisoner_count
- murder_manslaughter
- violent_crime_total
- robbery
- agg_assault
- vehicle_theft
- property_crime_total

```
crime_prison2 = filter(crime_prison, jurisdiction %in% c('IOWA', 'ILLINOIS', "NORTH DAKOTA", "SOUTH DAKOTA", "MINNESOTA", "WISCONSIN", "NEBRASKA", "KANSAS", "MISSOURI", "INDIANA", "MICHIGAN", "OHIO"))

crime_prison3 = crime_prison2 %>%
  select(1, 3, 4, 6, 7, 8, 9, 12, 13, 14, 17) #only selecting certain columns
```

Data Pre-processing and Tidy Data
CPSC 260

crime_prison3

```
jurisdiction year prisoner_count crimes_estimated state_population
1      ILLINOIS 2001          44348             TRUE          12520227
2       INDIANA 2001          19646            FALSE           6126743
violent_crime_total murder_manslaughter robbery agg_assault
1               79270                 982   24931       49347
2               22734                 413    7171       13434
property_crime_total vehicle_theft
1               434648         48733
2               211548         21499
```

Only copy and pasted first 2 rows

The data is tidy because

1. Each variable has its own column.
2. Each observation has its own row.
3. Each value has its own cell.

2. Look at the prisoner_count over the years for each Midwestern state by creating a new data set containing this information (jurisdiction, year, prisoner_count).

a) Change the data format so that you have "wide" data, i.e. data in a spreadsheet style. Do this by using the spread() command and then View() the data. [Hint: There should be a column for each year]

```
crime_problem2 = crime_prison2 %>%
  select(jurisdiction, year, prisoner_count)

crime_problem2

#making wide data
crime_problem2_spread = spread(crime_problem2, key = 'year', value = 'prisoner_count')
view(crime_problem2_spread)
```

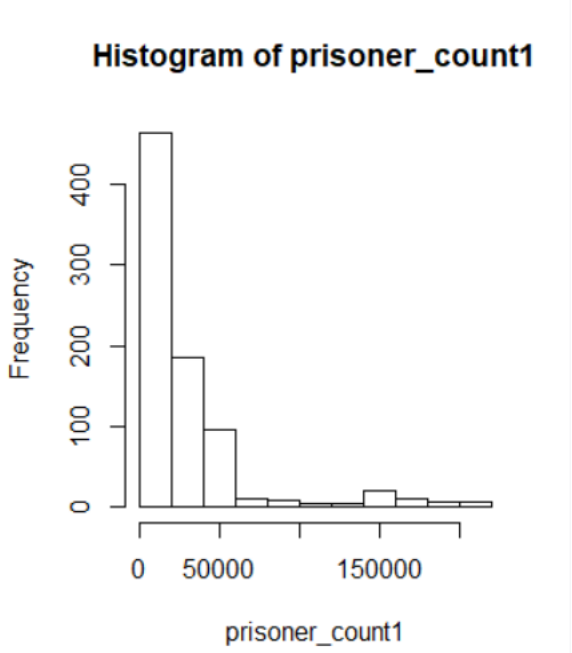
b) Put your data back into the "long" format (tidy) using gather() and then View() the data.

```
#making long data -
crime_problem2_long = gather(crime_problem2, year, prisoner_count)
view(crime_problem2_long)
```

3. Start looking for outliers and/or other "problems" in the data set. Do this by creating a histogram for each variable in the data set where this would be appropriate, i.e. for all quantitative variables except year. What do you see? Describe any possible errors that you find. You can make histograms very quickly using *hist()*. You are also welcome to use the ggplot2 package.

Data Pre-processing and Tidy Data
CPSC 260

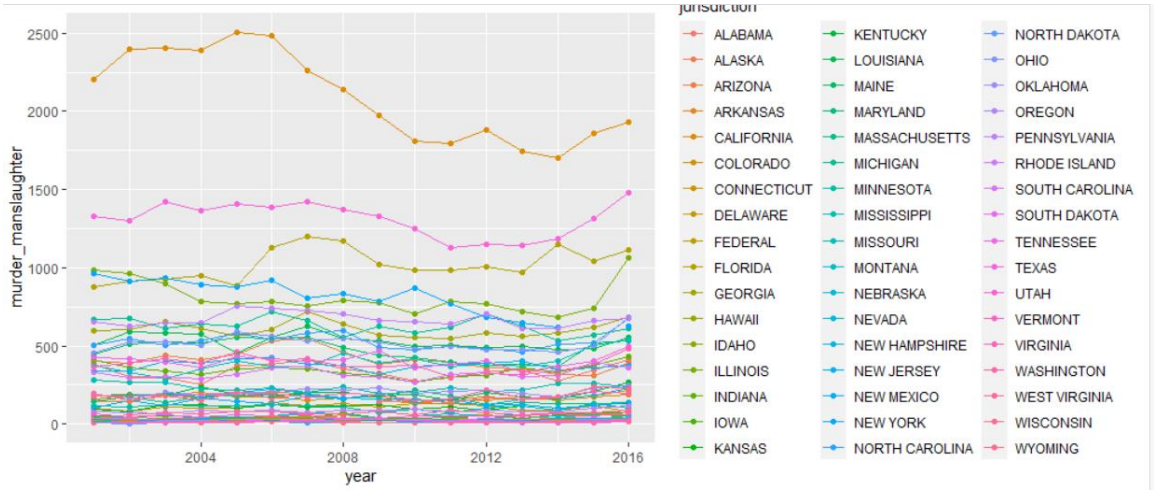
```
prisoner_count1 = crime_prison$prisoner_count
hist(prisoner_count1)

state_population1 = crime_prison$state_population
hist(state_population1)

violent_crime_total1 = crime_prison$violent_crime_total
hist(violent_crime_total1)

murder_manslaughter1 = crime_prison$murder_manslaughter
hist(murder_manslaughter1)

rape_legacy1 = crime_prison$rape_legacy
hist(rape_legacy1)

robbery1 = crime_prison$robbery
hist(robbery1)

agg_assault1 = crime_prison$agg_assault
hist(agg_assault1)

property_crime_total1 = crime_prison$property_crime_total
hist(property_crime_total1)

burglary1 = crime_prison$burglary
hist(burglary1)

larceny1 = crime_prison$larceny
hist(larceny1)

vehicle_theft1 = crime_prison$vehicle_theft
hist(vehicle_theft1)
```



The bar shouldn't be on 0 since there's not zero prisoners in a state

4. Due to the type of data we have, the histograms were not super helpful. Since the data was recorded over time, we should try looking at some time series plots. Use the code below to make the time series plot that shows how the number of *murder_manslaughter* crimes have changed over time for each state. What do you see? Describe any possible errors that you find.

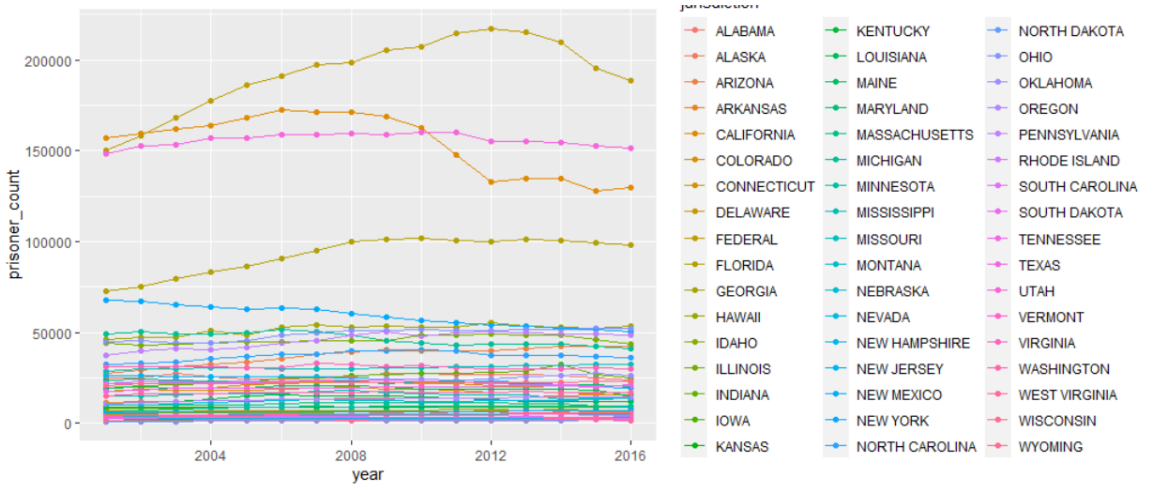
ggplot(data set, mapping=aes(x=year, y= murder_manslaughter, color=jurisdiction))+
geom_point()+
geom_line()



They mostly went down between 2008-2012 and then spiked back up after 2012. The graph is a huge mess and you can't really tell which states are which. Looks like some zero values

5. Create similar time series plots for the remaining quantitative variables. What do you see? Describe any possible errors that you find.

```
ggplot(crime_prison, mapping=aes(x=year, y= prisoner_count, color=jurisdiction))+
  geom_point()+
  geom_line()

ggplot(crime_prison, mapping=aes(x=year, y= state_population, color=jurisdiction))+
  geom_point()+
  geom_line()

ggplot(crime_prison, mapping=aes(x=year, y= violent_crime_total, color=jurisdiction))+
  geom_point()+
  geom_line()

ggplot(crime_prison, mapping=aes(x=year, y= rape_legacy, color=jurisdiction))+
  geom_point()+
  geom_line()

ggplot(crime_prison, mapping=aes(x=year, y= robbery, color=jurisdiction))+
  geom_point()+
  geom_line()

ggplot(crime_prison, mapping=aes(x=year, y= agg_assault, color=jurisdiction))+
  geom_point()+
  geom_line()

ggplot(crime_prison, mapping=aes(x=year, y= property_crime_total, color=jurisdiction))+
  geom_point()+
  geom_line()

ggplot(crime_prison, mapping=aes(x=year, y= burglary, color=jurisdiction))+
  geom_point()+
  geom_line()

ggplot(crime_prison, mapping=aes(x=year, y= larceny, color=jurisdiction))+
  geom_point()+
  geom_line()

ggplot(crime_prison, mapping=aes(x=year, y= vehicle_theft, color=jurisdiction))+
  geom_point()+
  geom_line()
```



Plus all the other graphs Seems to be missing values in a lot of them

6. You should have found several "problems" in the data. You will now "fix" the more obvious ones by doing the following:

a) Fill in the missing value that you saw in the murder_manslaughter data using either the mean or median of the data for that state. Explain how you chose which one to use. Show your work.

The mean because the median could be a tiny or super high number one year and it wouldn't be accurate.

```
crime_problem6a = crime_prison %>%
  select(jurisdiction, year, murder_manslaughter)

crime_problem6a_spread = spread(crime_problem6a, key = 'year', value =
    'murder_manslaughter')
view(crime_problem6a_spread)

crime_problem6a_spread[50,9] = 165
view(crime_problem6a_spread)
```

b) The same state had a missing value for vehicle_theft. The trend for this variable seems to closely follow that of another state. So it seems reasonable to fill in the missing value with the same value from the other state. Do this.

```
crime_problem6b = crime_prison %>%
  select(jurisdiction, year, vehicle_theft)

crime_problem6b_spread = spread(crime_problem6b, key = 'year', value = 'vehicle_theft')
view(crime_problem6b_spread)

crime_problem6b_spread[50,9] = 11023
view(crime_problem6b_spread)
```

c) There was most likely a typo in the robbery data. What do you think happened, i.e. what was the typo? Take your best guess and then make the change.

New York 2015, NA value. Going to take the average and insert that for the NA value.

```
crime_problem6c = crime_prison %>%
  select(jurisdiction, year, robbery)

crime_problem6c_spread = spread(crime_problem6c, key = 'year', value = 'robbery')
view(crime_problem6c_spread)

crime_problem6c_spread[33,16] = 30828
view(crime_problem6c_spread)
```

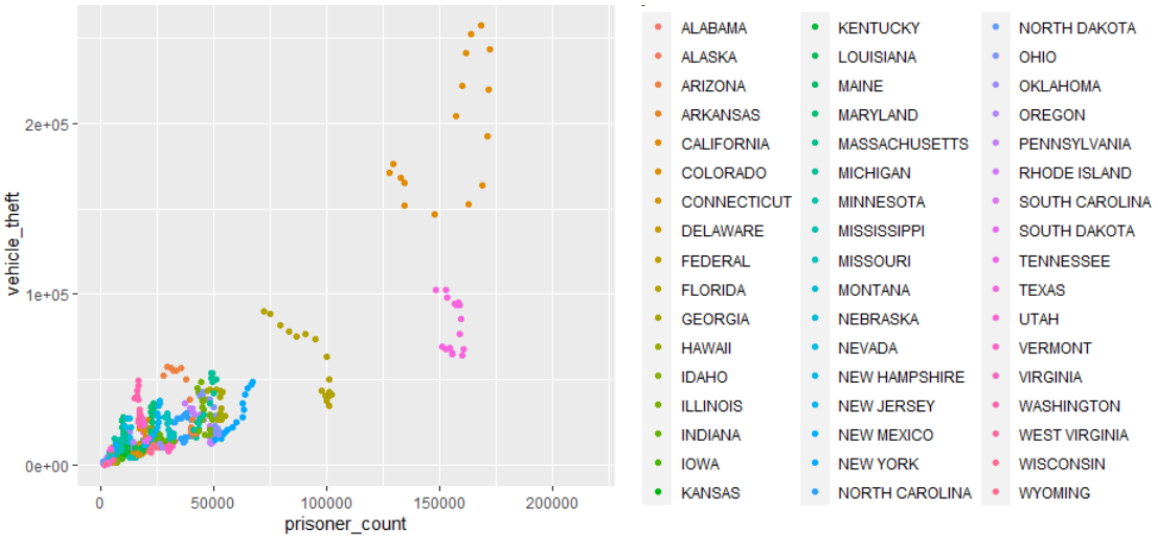
d) Another fairly obvious typo can be seen in the state_population data. Since state populations are easy to look up, find and replace the typo with the correct value.

```
crime_problem6d = crime_prison %>%
  select(jurisdiction, year, state_population)

crime_problem6d_spread = spread(crime_problem6d, key = 'year', value =
    'state_population')
view(crime_problem6d_spread)

crime_problem6d_spread[33,16] = 19795792
view(crime_problem6d_spread)
```

e) There also seem to be missing values in the property_crime_total data. A few values were coded as a zero. It's possible that there were no crimes of this nature, but it's more likely that the 0's represent missing values. Replace this data with the first non-missing value for the state (2004). This may not be a good replacement, but the downward trend in the data for this state is fairly slow.

```
crime_problem6e = crime_prison %>%
  select(jurisdiction, year, property_crime_total)

crime_problem6e_spread = spread(crime_problem6e, key = 'year', value = 'property_crime_total')
view(crime_problem6e_spread)

crime_problem6e_spread[28,2] = 61512
crime_problem6e_spread[28,3] = 61512
crime_problem6e_spread[28,4] = 61512

view(crime_problem6e_spread)
```

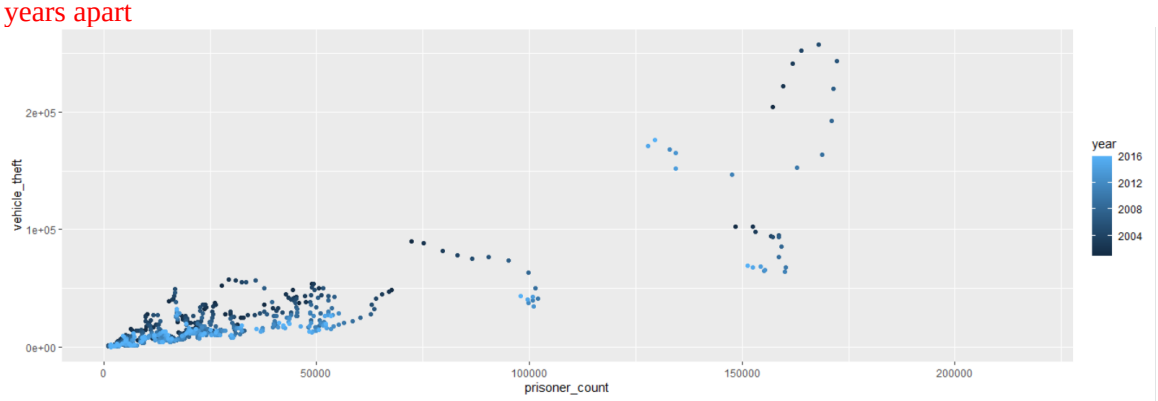
7. Another graphical option is to use scatterplots that display the relationship between two variables (rather than a time series plot). For example, plot prisoner_count and vehicle_theft together. Do you think this type of graph is helpful? You can do this using the *plot()* function or use ggplot2 functions.



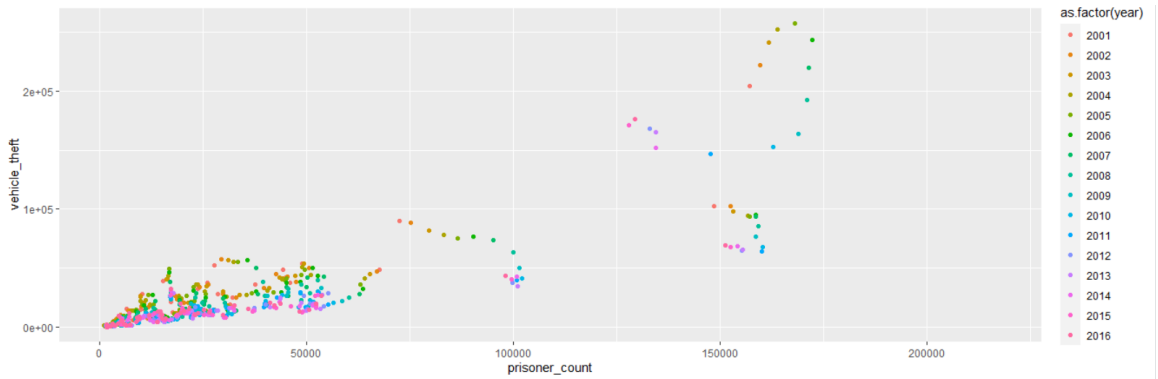
I think it is better because it is less of a mess than lines scattered everywhere. Although, differentiating some of the colors is rather difficult and it is a huge cluster in the bottom left.

8. What if we add a third variable to the scatter plot? Try mapping the year variable to color. Use as.factor(year) to make the color-coding easier to read. Then try mapping jurisdiction to color. Do either of these present the data in a way that's easier to identify possible outliers, typos, etc? Explain.

If you had points to label the dots then I think the first 2 would be much better. Easy to tell the years apart




```
ggplot(crime_prison, mapping=aes(x=prisoner_count, y= vehicle_theft, color=year))+
  geom_point()
```




```
ggplot(crime_prison, mapping=aes(x=prisoner_count, y= vehicle_theft, color=as.factor(year)))+
  geom_point()

ggplot(crime_prison, mapping=aes(x=prisoner_count, y= vehicle_theft, color = jurisdiction))+
  geom_point()

#same graph as number 7
```

9. We still need to check the crimes_estimated variable. Create a table for this categorical variable using *table()*. What do you see? Describe any possible errors that you find.

```
10.  table(crime_prison$crimes_estimated)
11.
12.             0     1 FALSE  TRUE
13.    18    96     2   683    17
```

0 and 1 for column names have no meaning

I pressed enter and inserted so many numbers that it changed the question numbers on accident for down below

14. You should have found two problems with the crimes_estimated data. You will now "fix" them by doing the following:

a) The years 2015 and 2016 were coded differently. Rather than True/False, we have 1/0 (where 1 = true). Change this so that all of the data is recorded in the same manner.

```
crimes_est_fix = crime_prison$crimes_estimated
crimes_est_fix[crimes_est_fix == 0] = FALSE
crimes_est_fix[crimes_est_fix == 1] = TRUE
crimes_est_fix
table(crimes_est_fix)
view(crimes_est_fix)
crimes_est_fix
          0     1 FALSE  TRUE
   18     0     0   779    19
```

b) Replace the missing value in this data with the mode for this variable.

```
crime_problem10_wide[9,2] = FALSE
```

```
crime_problem10_wide[9,3] = FALSE
crime_problem10_wide[9,4] = FALSE
crime_problem10_wide[9,5] = FALSE
crime_problem10_wide[9,6] = FALSE
crime_problem10_wide[9,7] = FALSE
crime_problem10_wide[9,8] = FALSE
crime_problem10_wide[9,9] = FALSE
crime_problem10_wide[9,10] = FALSE
crime_problem10_wide[9,11] = FALSE
crime_problem10_wide[9,12] = FALSE
crime_problem10_wide[9,13] = FALSE
crime_problem10_wide[9,14] = FALSE
crime_problem10_wide[9,15] = FALSE
crime_problem10_wide[9,16] = FALSE
crime_problem10_wide[9,17] = FALSE

crime_problem10_wide[36,17] = FALSE
crime_problem10_wide[33,16] = FALSE
view(crime_problem10_wide)
```

c) Check and make sure that the above replacement makes sense. This is easiest to do by looking at the crimes_estimated data in wide/spreadsheet format. What do you see?

```
crime_problem10 = crime_prison %>%
  select(jurisdiction, year, crimes_estimated)

crime_problem10

crime_problem10_wide = spread(crime_problem10, key = 'year', value = 'crimes_estimated')
view(crime_problem10_wide)
```

The rows with NA are now FALSE

15. Normally this type of data would be scaled based on population. This enables us to compare states and makes more sense since population changes over time. Use *mutate()* to create and add two new variables to the data set: total violent crimes per 100,000 population and total property crime per 100,000 population. [Hint: Divide by state_population then multiply by 100,000.]

```
Violent_crime_per_100k = (crime_prison$violent_crime_total /
crime_prison$state_population) * 100000
Property_crime_per_100k = (crime_prison$property_crime_total /
crime_prison$state_population) * 100000

crime_prison %>%
  mutate(Violent_crime_per_100k)

crime_prison %>%
  mutate(Property_crime_per_100k)
```

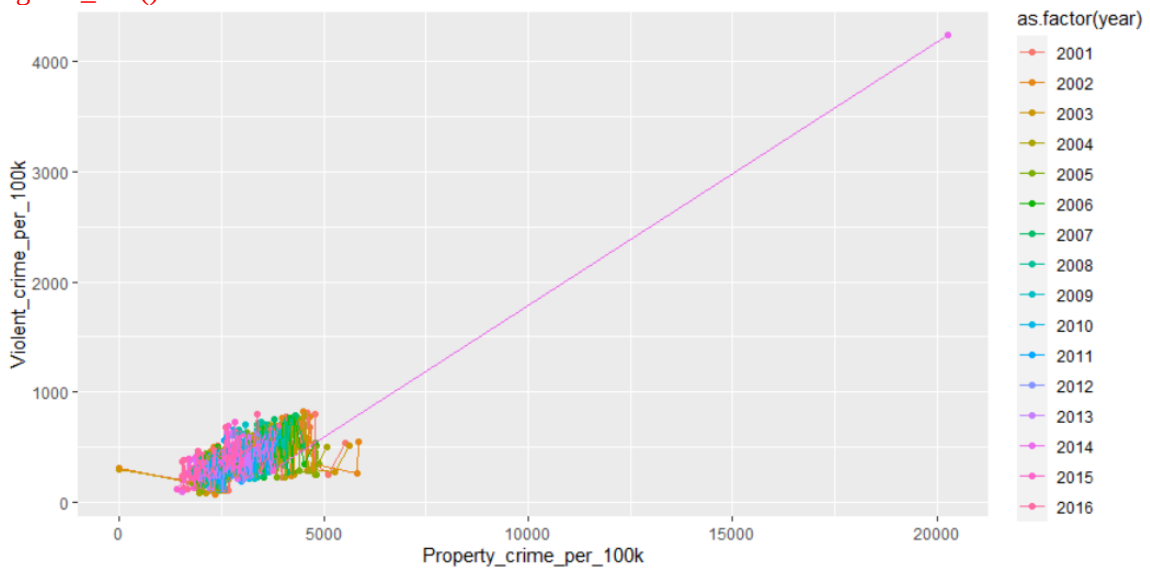
Violent_crime_per_100k
NA
438.18272
589.46073
540.32756
452.36980
615.21431
349.62849
334.59481
611.09793

Property_crime_per_100k
NA
3876.850
3655.130
5537.514
3677.815
3277.959
3856.633
2774.674
3439.497

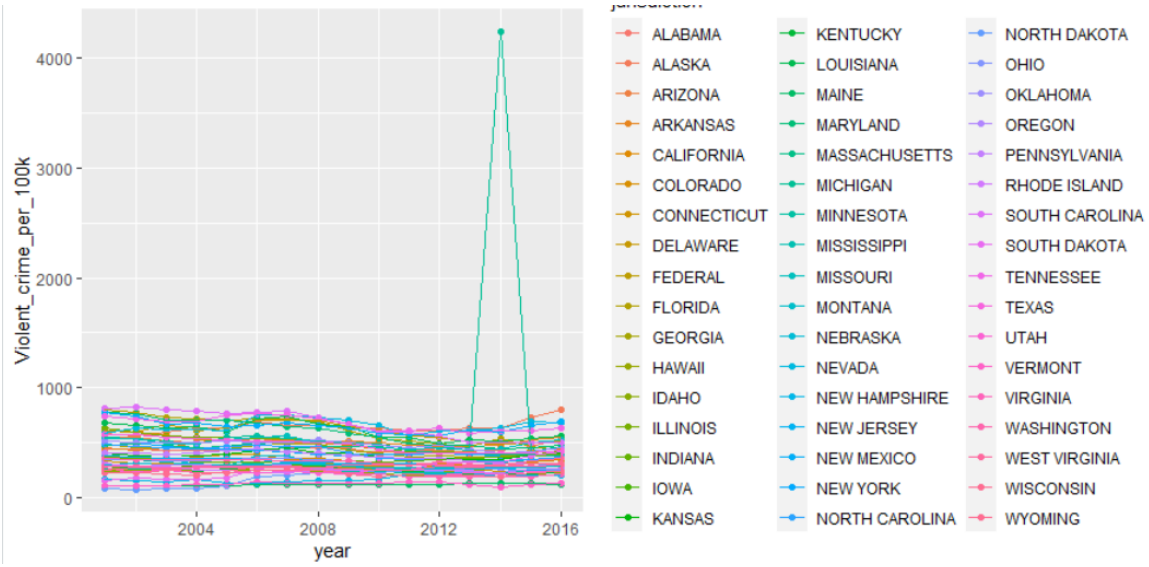
16. Compare the time series graph for violent crimes for each state to the time series graph for violent crimes per 100,000 population over time for each state. What do you see? How does the story change?

Violent crimes & property crimes skyrocketed in 2014. As property crime went up, so did violent crime

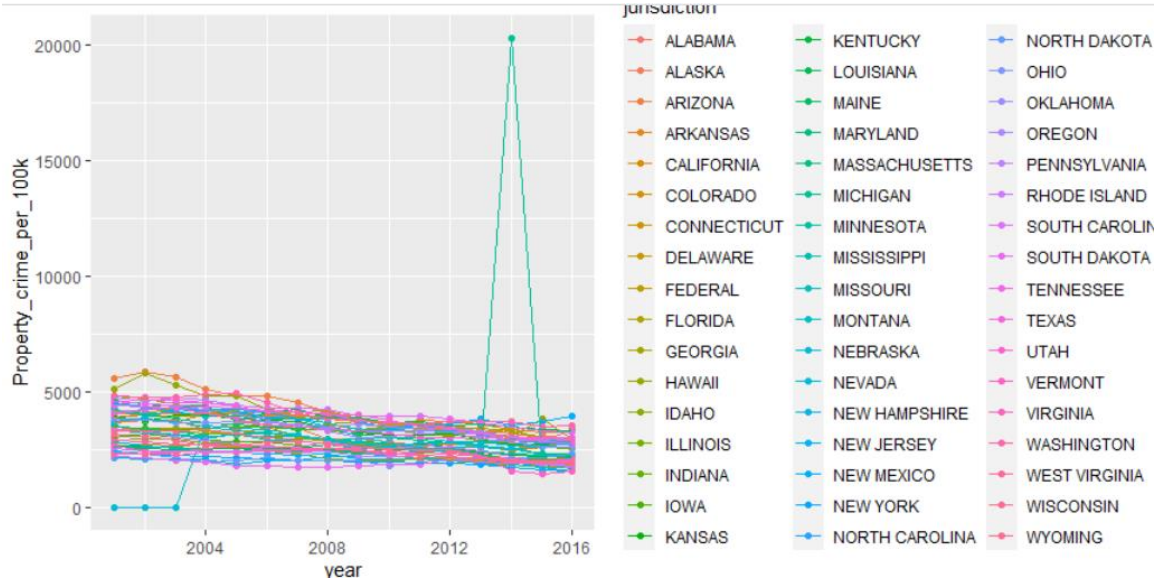
```
ggplot(crime_prison, mapping=aes(x=Property_crime_per_100k, y= Violent_crime_per_100k, color=as.factor(year)))+
  geom_point()+
  geom_line()
```




```
ggplot(crime_prison, mapping=aes(x=year, y= Violent_crime_per_100k, color=jurisdiction))+
  geom_point()+
  geom_line()
```




```
ggplot(crime_prison, mapping=aes(x=year, y= Property_crime_per_100k, color=jurisdiction))+
  geom_point()+
  geom_line()
```



17. Practice doing another data transformation by creating and adding a new variable to the data set: the min-max normalization for prisoner_count. Then create a histogram to verify that all values fall between 0 and 1. [Hint: Subtract the minimum prisoner_count and then divide by its range.]

```
prisoner_count_max = max(crime_prison$prisoner_count)
prisoner_count_min = min(crime_prison$prisoner_count)
prisoner_count_range = prisoner_count_max - prisoner_count_min

pris_count_min_max_norm = (crime_prison$prisoner_count -
prisoner_count_min) / prisoner_count_range
```

```
crime_prison %>%
  mutate(pris_count_min_max_norm)

hist(pris_count_min_max_norm)
```