STA234: HW4

Derin Gezgin

2025-03-06

95/100

Importing the Required Libraries

```
library(ggplot2)
library(ggmap)
library(dplyr)
```

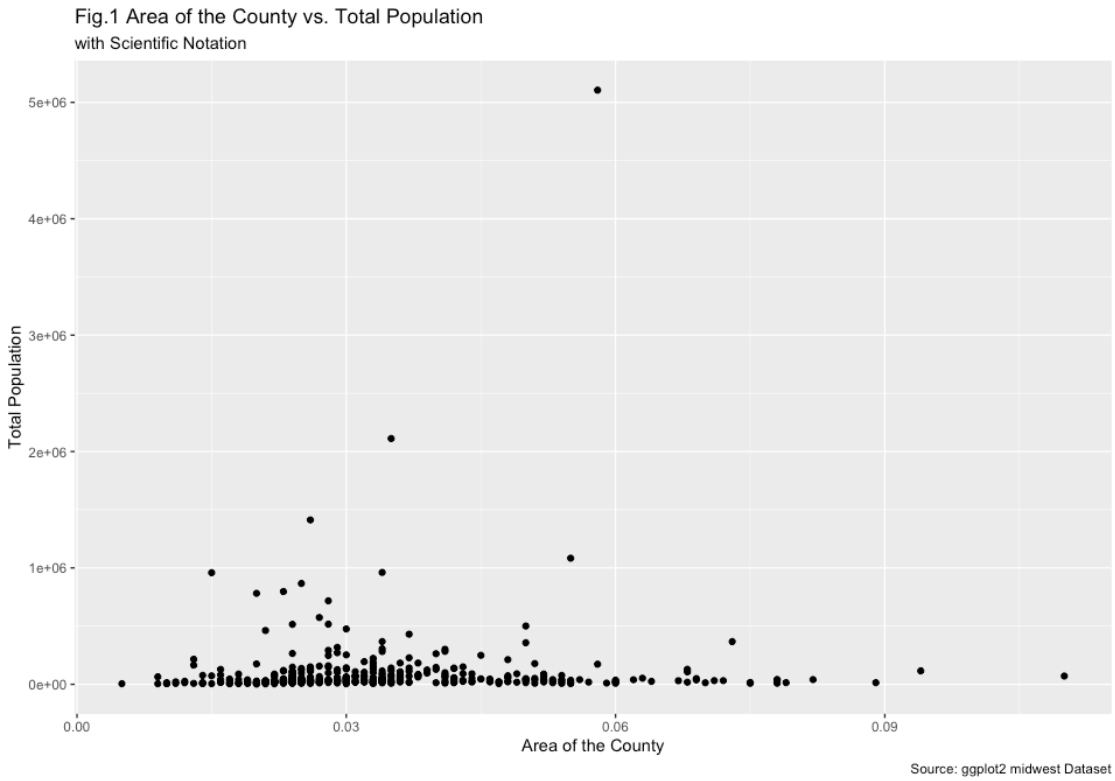
Problem 1

In this problem we will use the *midwest* data in **ggplot2** package. First read about the data to understand the variables. In steps stated below, we will make a scatterplot between two numerical variables, then we will add more variables to the plot using aesthetics like color and size, and enhance plot in various ways. Do the following parts:

Part A [5 Points]

Make a scatterplot of area (x-axis) versus total population (y-axis). Label axes and add title. What do you notice?

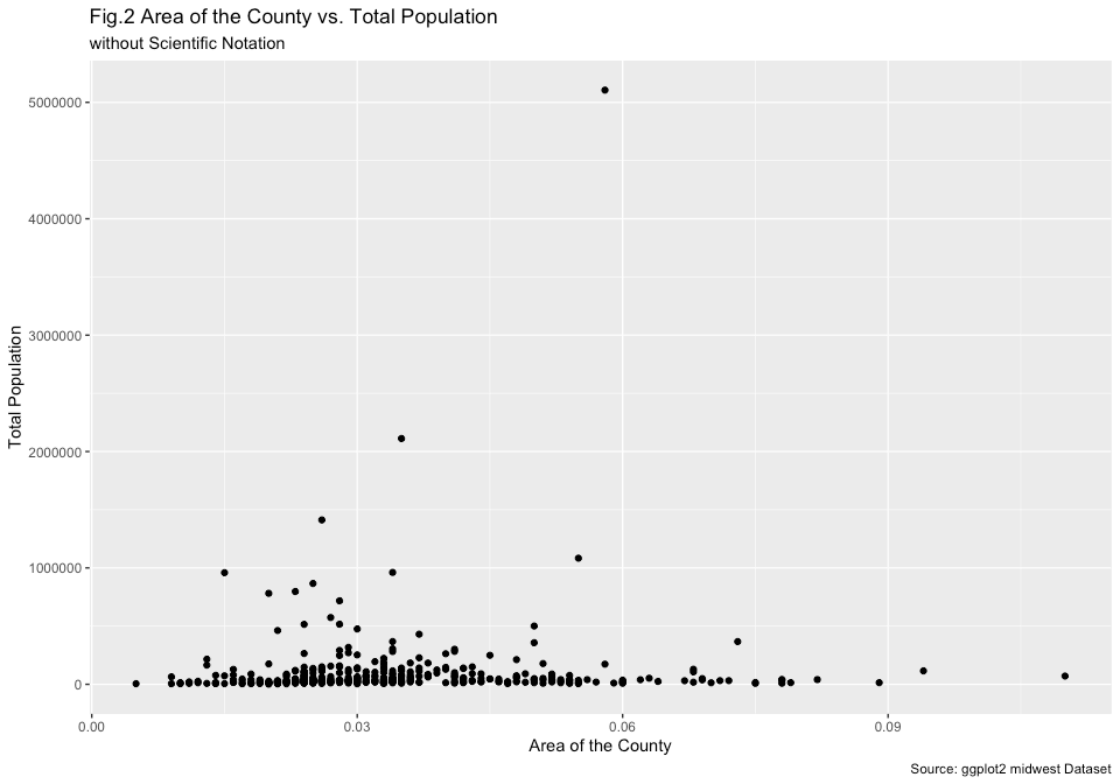
```
area.population.plot = ggplot(data = midwest,
                              aes(x = area,
                                  y = poptotal)) +
    geom_point() +
    xlab("Area of the County") +
    ylab("Total Population") +
    labs(title = "Fig.1 Area of the County vs. Total Population",
         subtitle = "with Scientific Notation",
         caption = "Source: ggplot2 midwest Dataset")

area.population.plot
```



In this initial graph the y-axis values are in scientific notation. R automatically simplifies them into scientific notation as the full numbers take significantly more space in the plot. I can also see there are some obvious outliers and most of the data-points have clustered in the bottom-side of the graph.

Use options(scipen=999) to turn off scientific notation like 1e+06 and redo the plot. Add a chart number (Fig.1 etc.) with title and subtitle to your plot.

```
options(scipen = 999)

area.population.plot = area.population.plot +
    labs(title = "Fig.2 Area of the County vs. Total Population",
         subtitle = "without Scientific Notation",
         caption = "Source: ggplot2 midwest Dataset")

area.population.plot
```



When I used `options(scipen = 999)`, the y-axis values changed to normal base-10 numbers.

Part B [5 Points]

Identify all places (state and county) with total population above 1000000 and report them.

We can achieve this by using the base-R and also the dplyr package. I am going to be showing both versions.

```
filter(midwest[, c("county", "state", "poptotal")], midwest$poptotal > 1000000)
```

```
## # A tibble: 4 × 3
##   county   state poptotal
##   <chr>    <chr>    <int>
## 1 COOK     IL     5105067
## 2 OAKLAND  MI     1083592
## 3 WAYNE    MI     2111687
## 4 CUYAHOGA OH     1412140
```

```
midwest[midwest$poptotal > 1000000, c("county", "state", "poptotal")]
```

```
## # A tibble: 4 × 3
##   county   state poptotal
```

```
##   <chr>    <chr>    <int>
## 1 COOK     IL     5105067
## 2 OAKLAND  MI     1083592
## 3 WAYNE    MI     2111687
## 4 CUYAHOGA OH     1412140
```

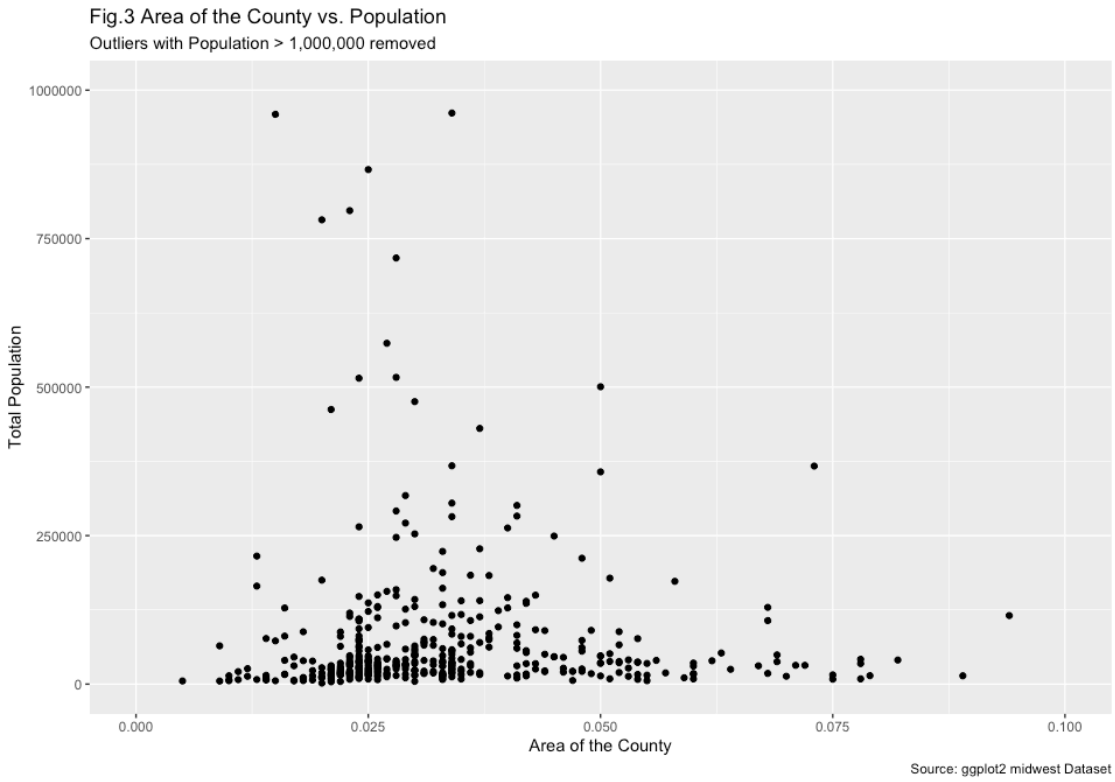
in both cases, we can see that Illinois Cook, Minnesota Oakland & Wayne, and Ohio Cuyahoga has a total population above 1,000,000.

Part C [7.5 Points]

Delete the above outlier and redo the scatterplot. You can use any way to remove this, here are some options:

- Change x and y axes limits for the plot using two additional settings:
    - xlim(c(0, 0.1))
    - ylim(c(0, 1000000))
- use function coord_cartesian() with same limits as above.
- create a subset of the Midwest data with population total less than or equal to 1000000 and then redo the scatter plot.

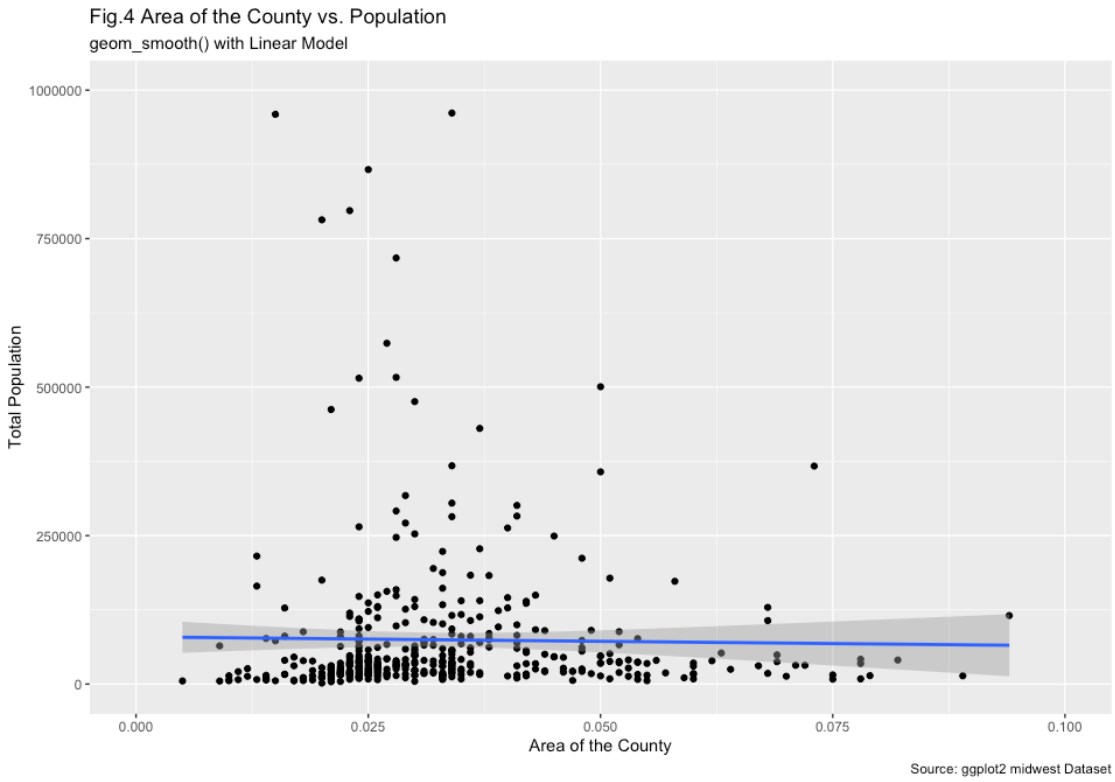
```
area.population.plot =  area.population.plot +
    xlim(c(0, 0.1)) +
    ylim(c(0, 1000000)) +
    labs(title = "Fig.3 Area of the County vs. Population",
         subtitle = "Outliers with Population > 1,000,000 removed",
         caption = "Source: ggplot2 midwest Dataset")

area.population.plot
```



Part D [5 Points]

Using geom_smooth() add linear regression model. What can you say about the relation between population and area?

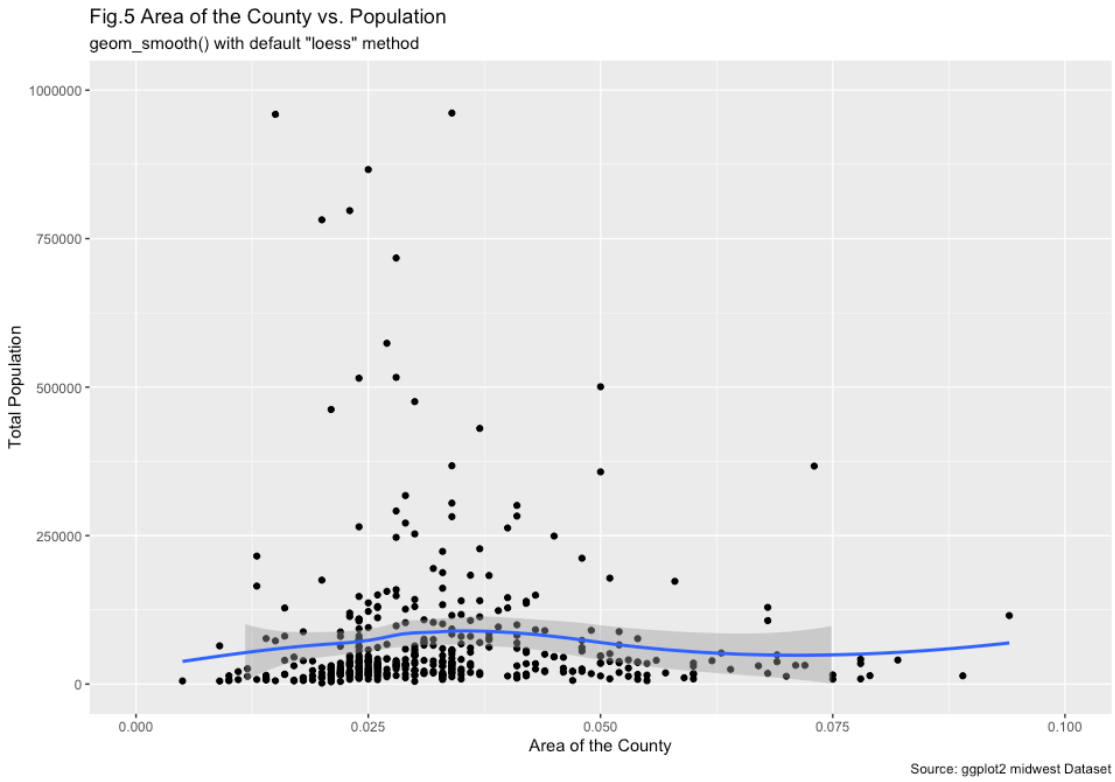
```
area.population.plot.lm = area.population.plot +
    geom_smooth(method = "lm") +
    labs(title = "Fig.4 Area of the County vs. Population",
         subtitle = "geom_smooth() with Linear Model",
         caption = "Source: ggplot2 midwest Dataset")

area.population.plot.lm
```



From the linear regression line, we can see that there is weak or nearly no correlation at all between the area of the county and the total population as the best=fit line is nearly flat. At the same time, there is high variability in the relatively small countries as they have both high and low populations. Even though we removed the countries with population larger than 1,000,000, there are still some obvious outliers in the data. Lastly, we can also see that, as the county gets larger, it is tend to have a smaller population, we can see a slight triangle covering the data points which shows us this.

Using geom_smooth() with R's default method "loess" add a fitted non-linear curve instead of lm.

```
area.population.plot.loess = area.population.plot +
    geom_smooth(method = "loess") +
    labs(title = "Fig.5 Area of the County vs. Population",
         subtitle = "geom_smooth() with default \"loess\" method",
         caption = "Source: ggplot2 midwest Dataset")

area.population.plot.loess
```



Read help file for the geom_smooth() to explain this ”loess” and tell what do you observe is different in lm and R’s default.

When I checked the help file and recalling my previous experiences, I can say that `geom_smooth` adds a fitted line to the plots to show the relationship between my variables of interest. By default, if nothing is specified in the parameters, `geom_smooth` uses `stats::loess()` as the smoothing method for samples less than 1,000 observations, or `mgcv:gam()` otherwise.

When I checked the documentation for `stats::loess()`, I saw that it creates the best-fit-line by performing **local** polynomial regression, determining the fit line using the nearby data points.

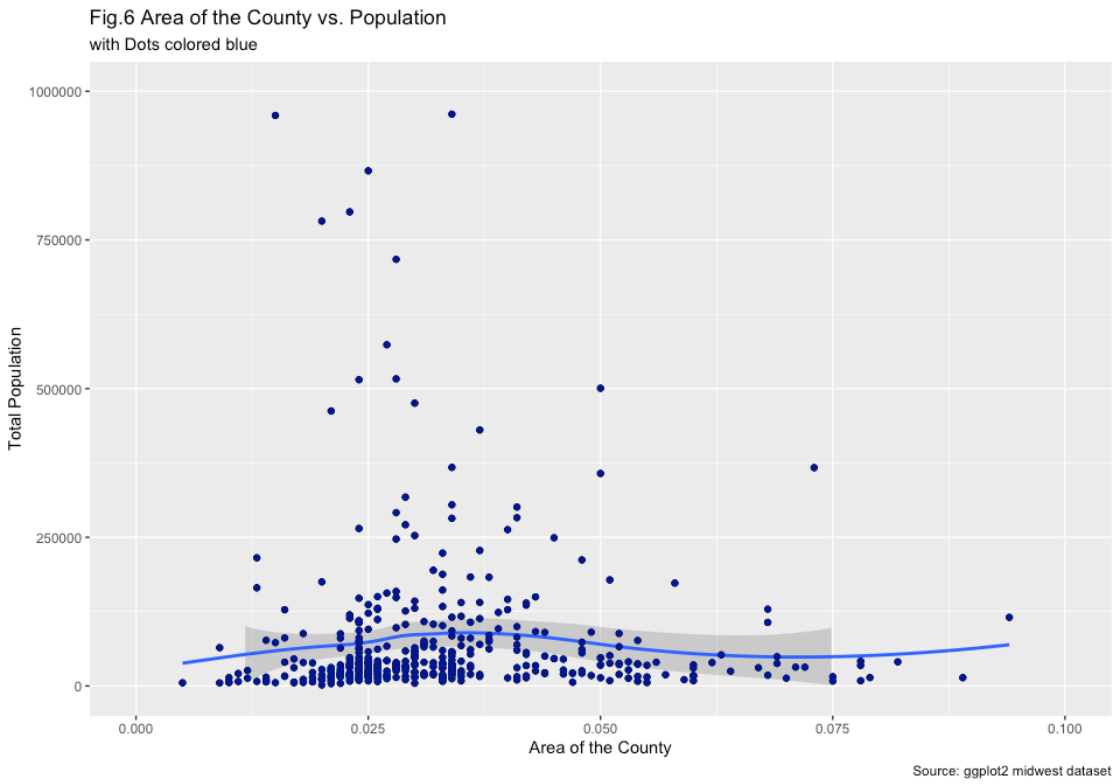
We can say that, in general, `stats::loess()` or the default method of `geom_smooth()` fits a smooth curve that can show the non-linear patterns in the data. On the other hand, `method = "lm"` fits a linear model in the data which is a straight line, representing the best **linear** fit.

In a case where our data has a linear relationship, both methods may show a similar result. On the other hand, in a non-linear relationship, the `stats::loess()` can fit a smoother curve which can better show the patterns in the data. Even in our example -which is mostly linear- the curved version has a slight bump in the relatively small counties, taking the outliers into account, compared to the nearly straight line.

Part E [2.5 Points]

Change color of the dots in the above plot.

```
area.population.colored = area.population.plot.loess +
    geom_point(color = "darkblue") +
    labs(title = "Fig.6 Area of the County vs. Population",
         subtitle = "with Dots colored blue",
         caption = "Source: ggplot2 midwest dataset")

area.population.colored
```

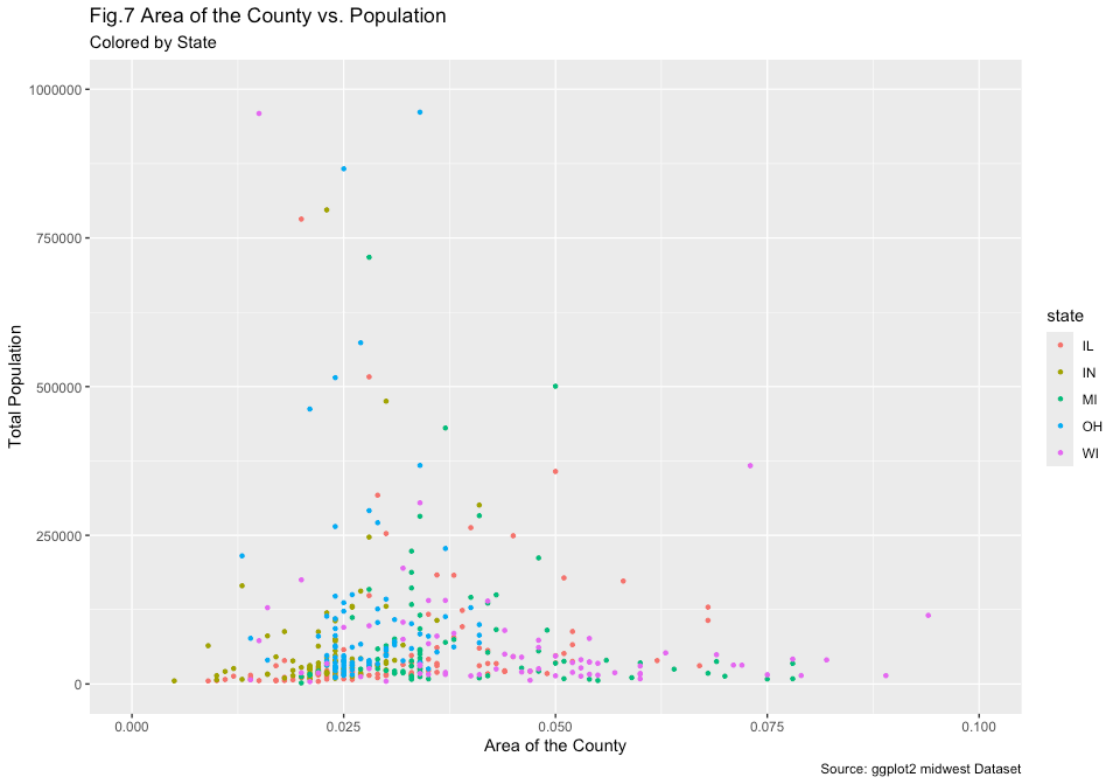


Part F [7.5 Points]

Change color of the dots based on the state (categorical) variable.

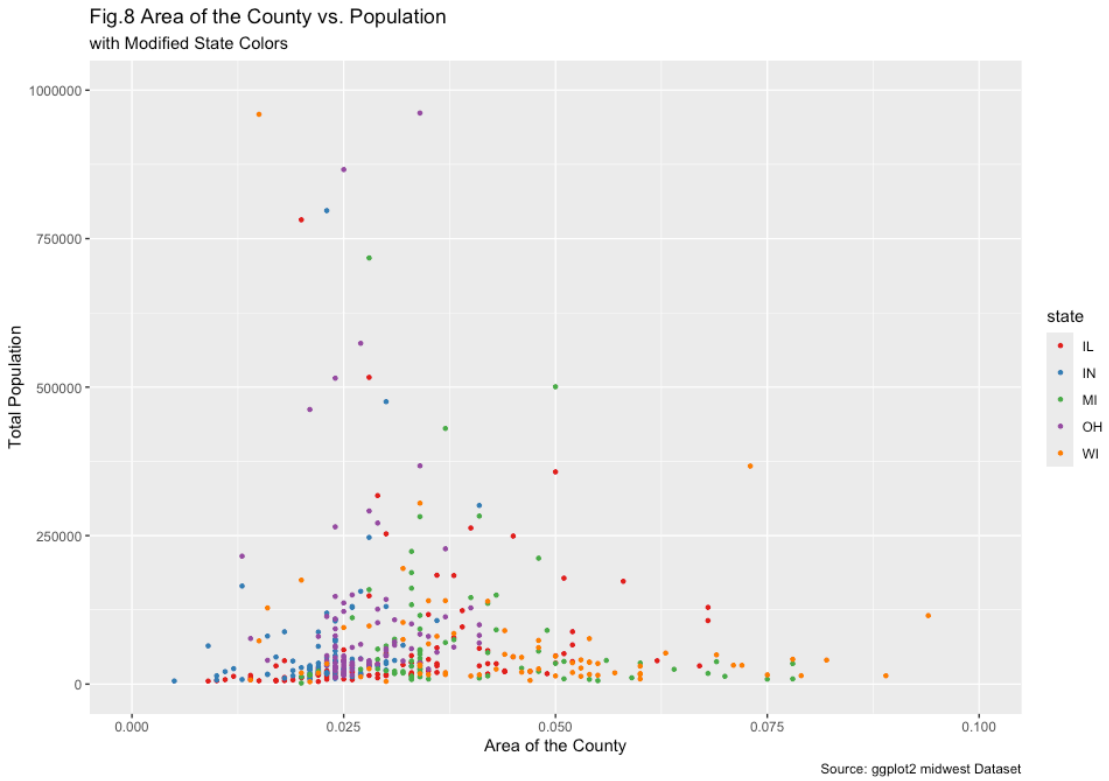
```
area.population.state = ggplot(data = midwest,
                               aes(x = area,
                                   y = poptotal,
                                   color = state)) +
    geom_point(size = 0.9) +
    xlim(c(0, 0.1)) +
    ylim(c(0, 1000000)) +
    xlab("Area of the County") +
```

```
    ylab("Total Population") +
    labs(title = "Fig.7 Area of the County vs. Population",
         subtitle = "Colored by State",
         caption = "Source: ggplot2 midwest Dataset")

area.population.state
```



Show how to change the color of the state dots from the R's default choice in above plot.

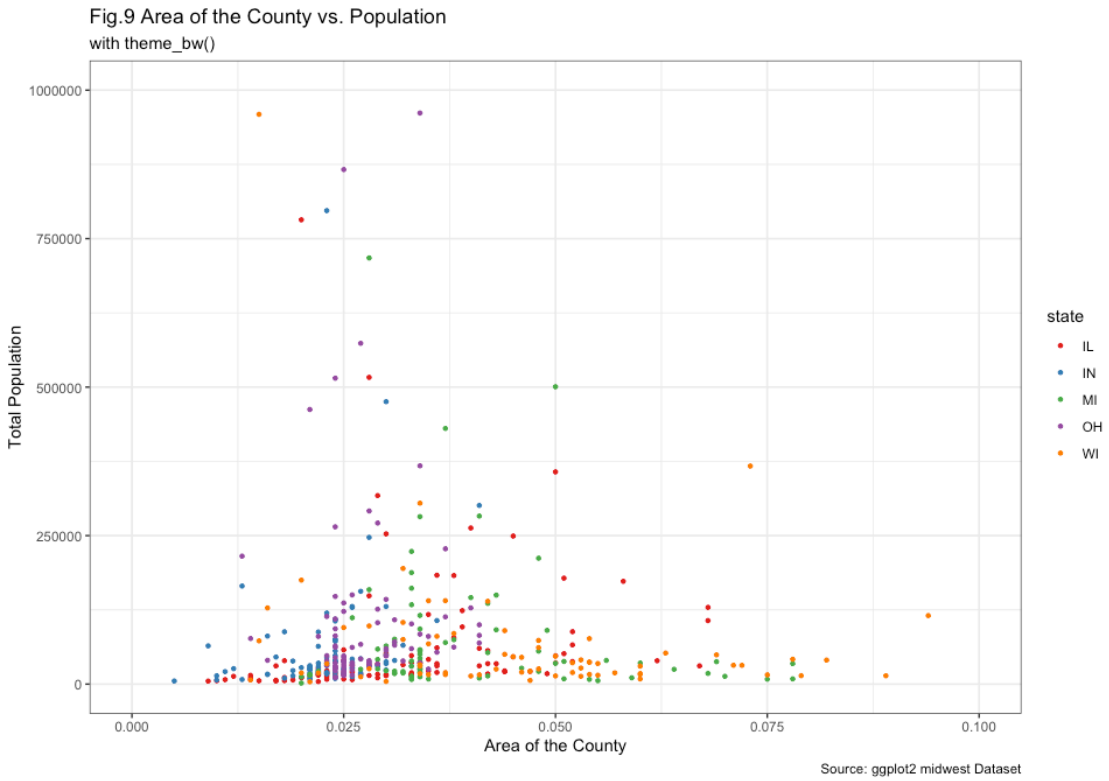
```
area.population.state.specific = area.population.state +
    scale_color_manual(values = c("#E41A1C", "#377EB8", "#4DAF4A", "#984EA3",
"#FF7F00")) +
    labs(title = "Fig.8 Area of the County vs. Population",
         subtitle = "with Modified State Colors",
         caption = "Source: ggplot2 midwest Dataset")

area.population.state.specific
```



Part G [5 Points]

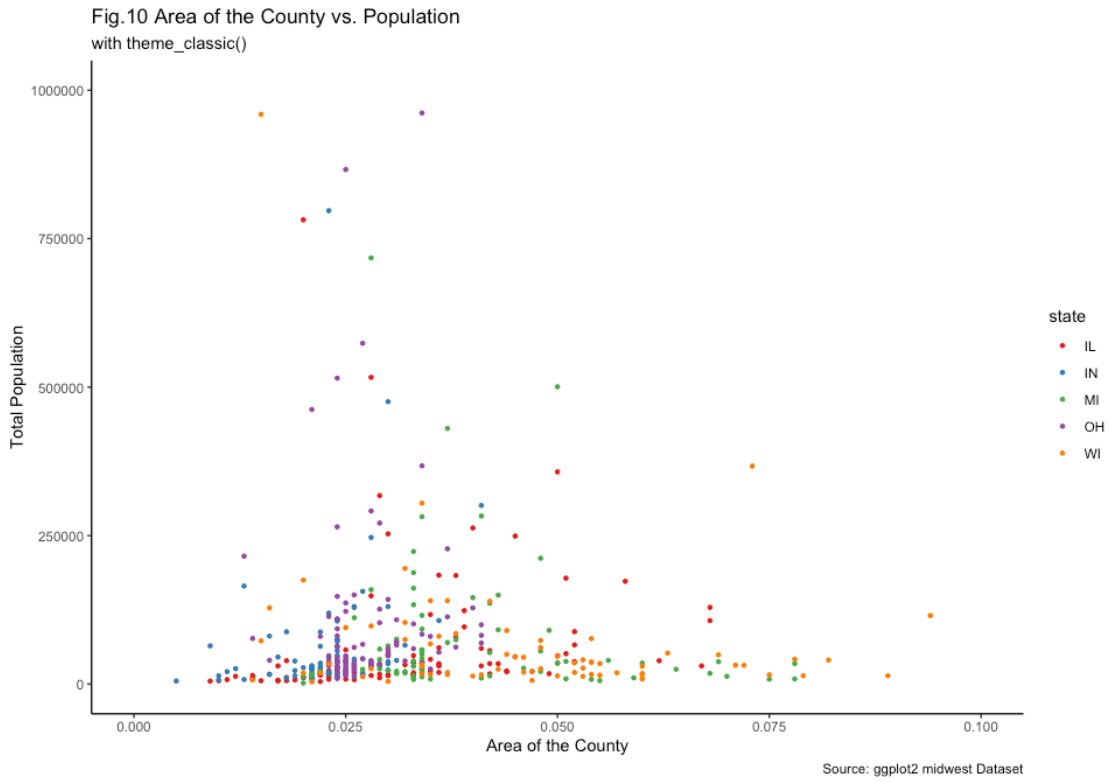
Use theme_bw() to change the theme of the above (Fig.8) plot.

```
apss_bw = area.population.state.specific +
    theme_bw() +
    labs(title = "Fig.9 Area of the County vs. Population",
         subtitle = "with theme_bw()",
         caption = "Source: ggplot2 midwest Dataset")

apss_bw
```



Use theme_classic() to change the theme of the above (Fig.8) plot.

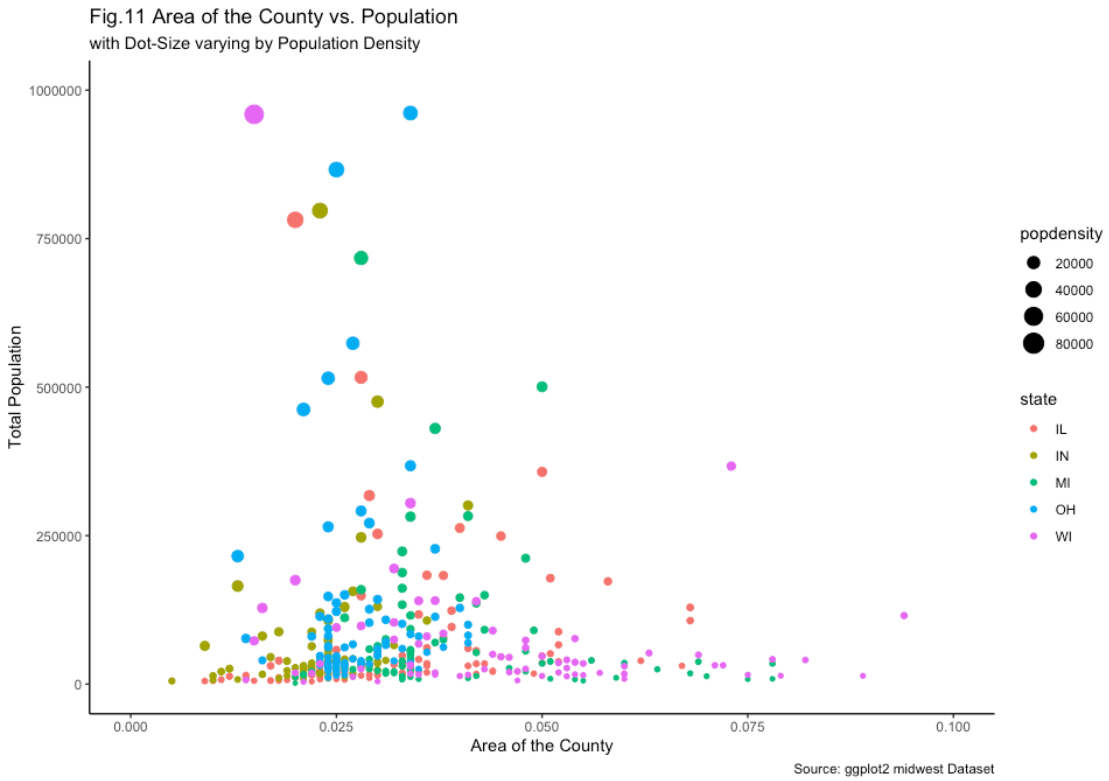
```
apss_classic = area.population.state.specific +
    theme_classic() +
    labs(title = "Fig.10 Area of the County vs. Population",
         subtitle = "with theme_classic()",
         caption = "Source: ggplot2 midwest Dataset")

apss_classic
```



Part H [7.5 Points]

Have the dot size vary by popdensity (continuous) variable in above plot. What do you observe?

```
apss.density = ggplot(data = midwest,
                      aes(x = area,
                          y = poptotal,
                          color = state,
                          size = popdensity)) +
    geom_point() +
    xlim(c(0, 0.1)) +
    ylim(c(0, 1000000)) +
    theme_classic() +
    xlab("Area of the County") +
    ylab("Total Population") +
    labs(title = "Fig.11 Area of the County vs. Population",
         subtitle = "with Dot-Size varying by Population Density",
         caption = "Source: ggplot2 midwest Dataset")

apss.density
```



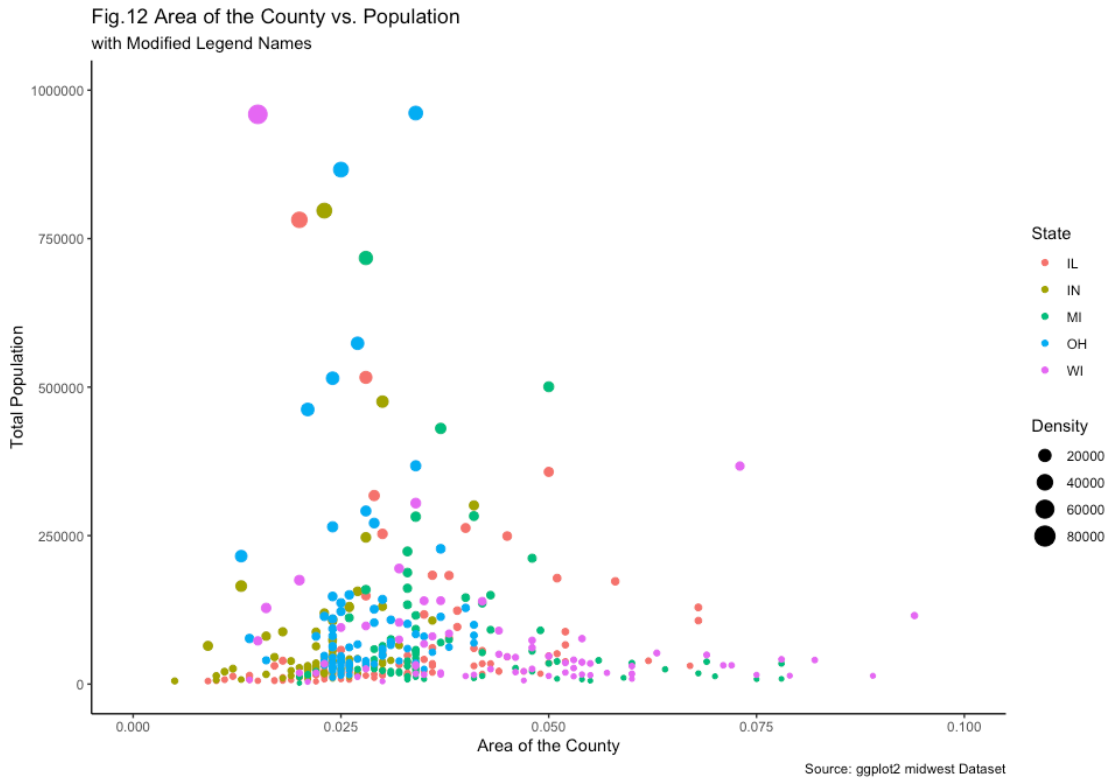
From the graph, we can see that there is a higher population density in larger counties as the large dots are mostly clustered on the relatively small counties. As expected, we can see that, counties with small areas and large populations have higher population densities.

At the same time, we can clearly see that some states have relatively small counties as they do not appear on the right-side of the graph. For example Indiana, Ohio, and -slightly- Illinois. We can see that Wisconsin has a high variability in both county size and population as it spans across both Axis and have varying dot sizes.

Part I [5 Points]

Modify legend for state and popdensity to States and Density respectively.

```
apss.density = apss.density +
    labs(title = "Fig.12 Area of the County vs. Population",
         subtitle = "with Modified Legend Names",
         caption = "Source: ggplot2 midwest Dataset",
         size = "Density",
         color = "State")

apss.density
```



Note: I do not know why but when I set the legend names to State and Density, it flips the order of the legends. I could not figure out the reason for this.

Part J [7.5 Points]

Change state names to actual names of the state, replace IL with Illinois, IN with Indiana and so on.

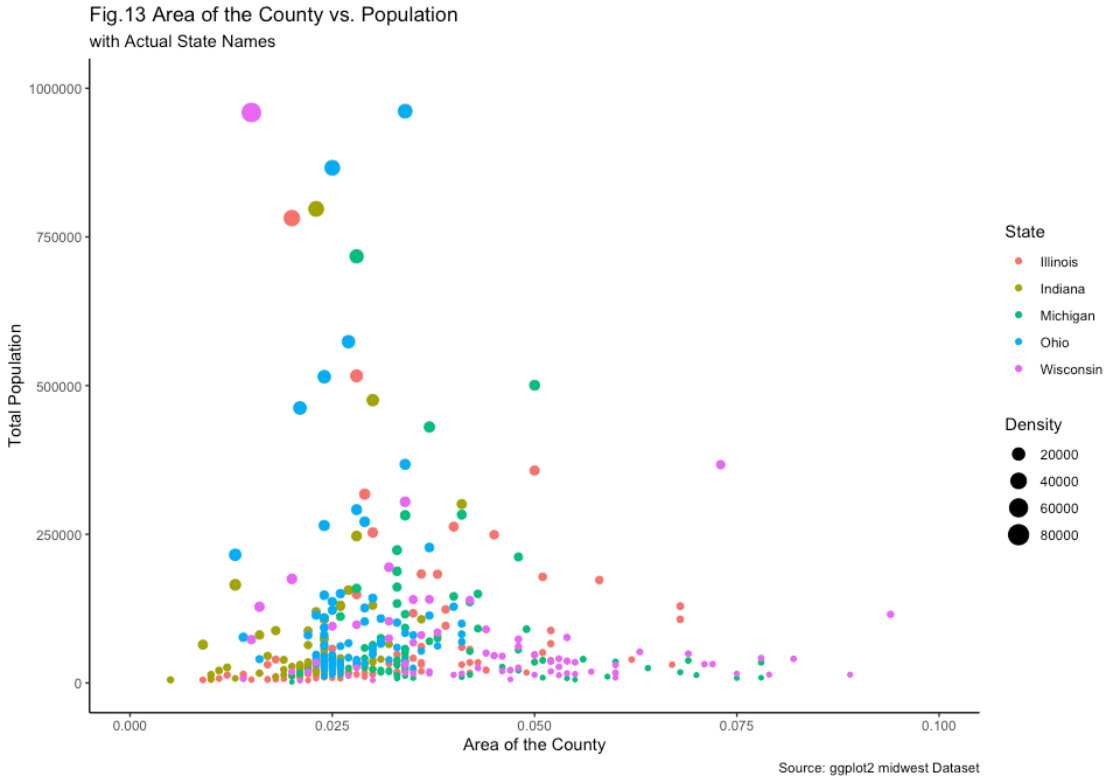
To do this, we can first convert the state variable into a factor

```
midwest$state = factor(midwest$state,
                       levels = c("IL", "IN", "MI", "OH", "WI"),
                       labels = c("Illinois", "Indiana", "Michigan", "Ohio",
"Wisconsin"))
```

Following this, we can plot the data as usual.

```
apss.density.named = ggplot(data = midwest,
                       aes(x = area,
                           y = poptotal,
                           color = state,
                           size = popdensity)) +
    geom_point() +
    xlim(c(0, 0.1)) +
    ylim(c(0, 1000000)) +
```

```
    theme_classic() +
    xlab("Area of the County") +
    ylab("Total Population") +
    labs(title = "Fig.13 Area of the County vs. Population",
         subtitle = "with Actual State Names",
         caption = "Source: ggplot2 midwest Dataset",
         size = "Density",
         color = "State")

apss.density.named
```

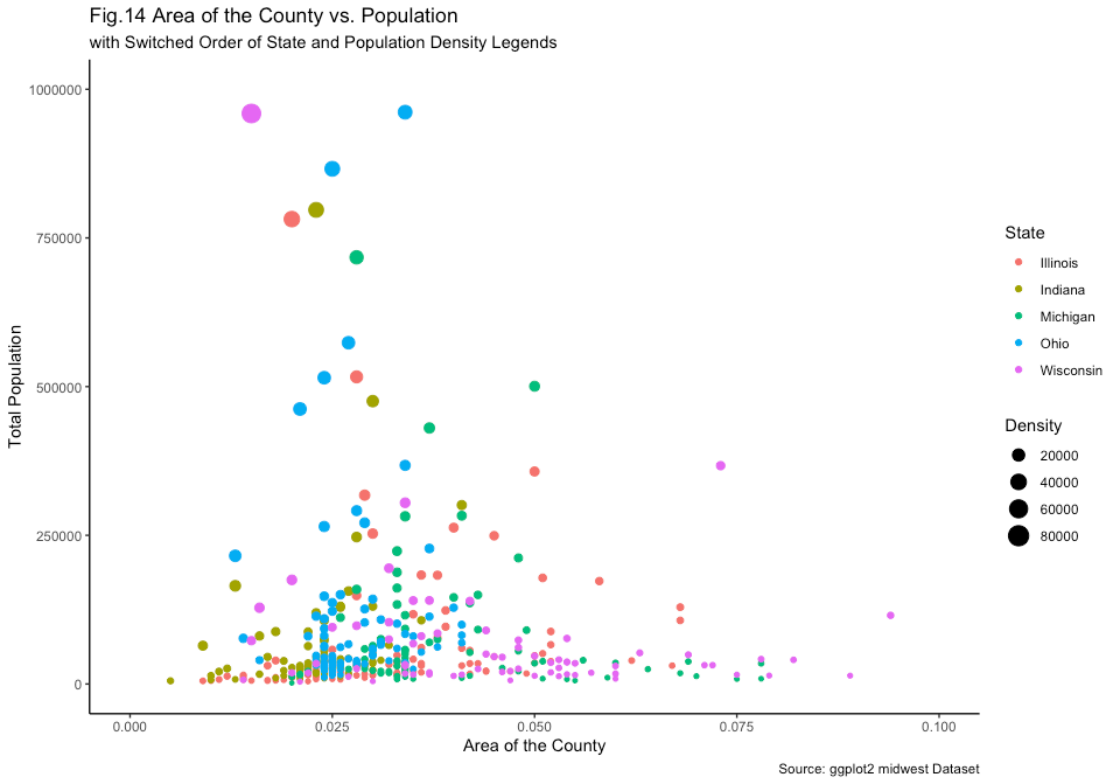


Part K [7.5 Points]

Switch the order of legend for state and popdensity using guides() function.

```
apss.density.ordered = apss.density.named +
    guides(color = guide_legend(order = 1),
           size = guide_legend(order = 2)) +
    labs(title = "Fig.14 Area of the County vs. Population",
         subtitle = "with Switched Order of State and Population Density
Legends",
         caption = "Source: ggplot2 midwest Dataset",
         color = "State",
         size = "Density")
```

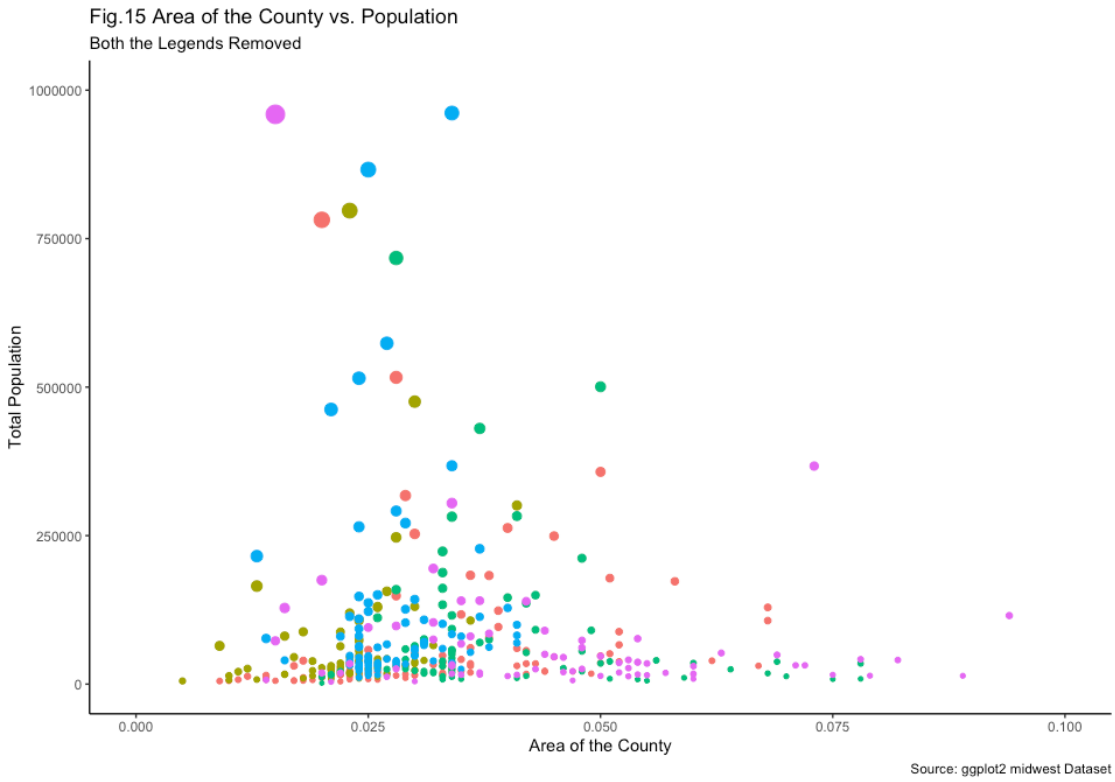
```
apss.density.ordered
```



Part L [5 Points]

Remove legend from the plot.

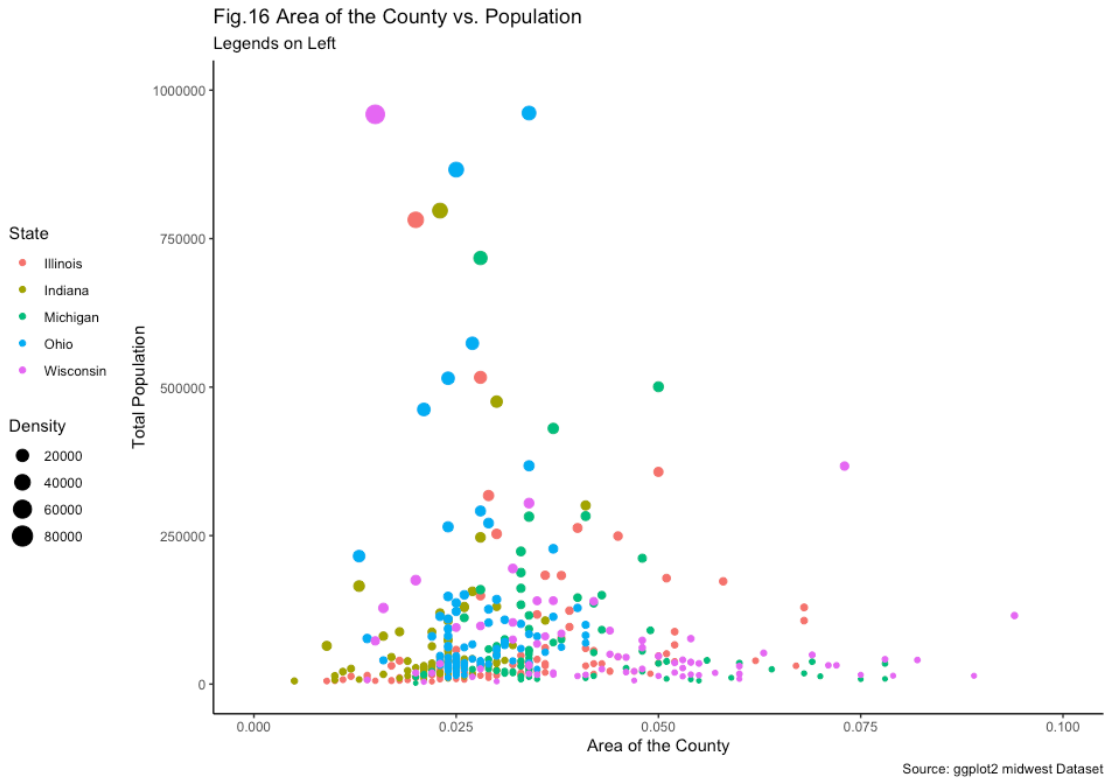
```
apss.density.removed = apss.density.named +
    labs(title = "Fig.15 Area of the County vs. Population",
         subtitle = "Both the Legends Removed",
         caption = "Source: ggplot2 midwest Dataset",
         color = "State",
         size = "Density") +
    theme(legend.position = "none")

apss.density.removed
```



Part M [5 Points]

Make legend move to the left side.

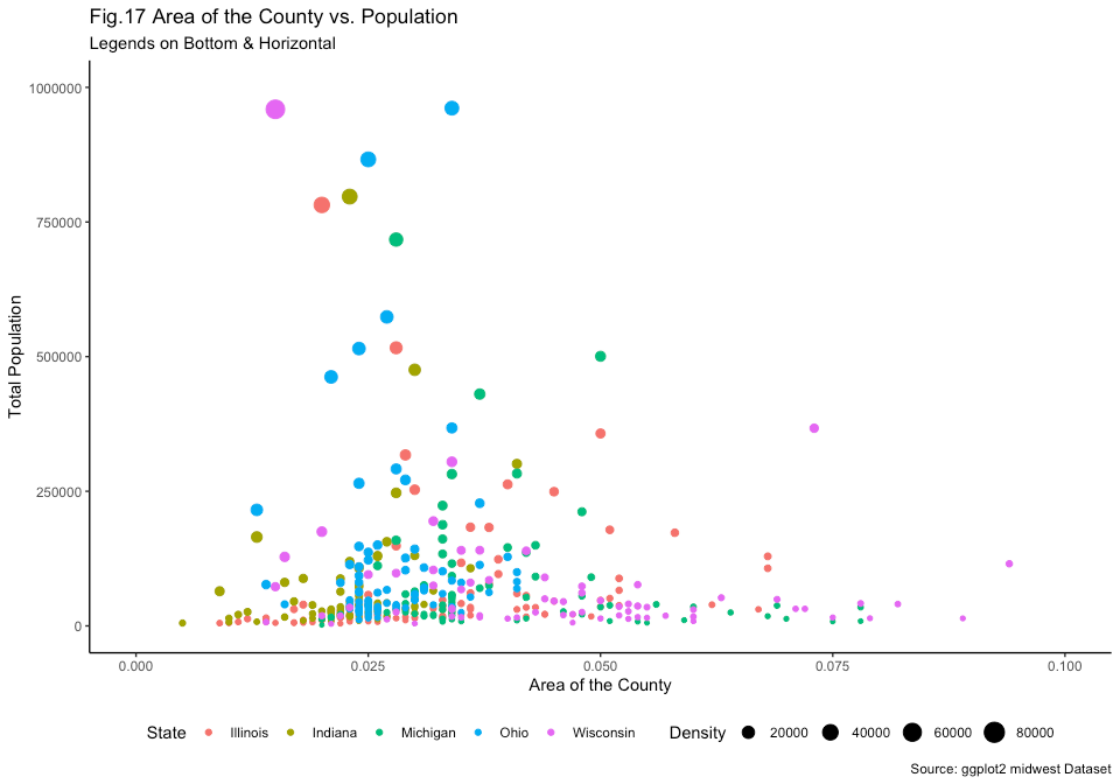
```
apss.density.left = apss.density.named +
    labs(title = "Fig.16 Area of the County vs. Population",
         subtitle = "Legends on Left",
         caption = "Source: ggplot2 midwest Dataset",
         color = "State",
         size = "Density") +
    theme(legend.position = "left")

apss.density.left
```



Part N [5 Points]

Make legend move to the bottom and horizontal.

```
apss.density.bottom.h = apss.density.named +
    theme(legend.position = "bottom",
          legend.direction = "horizontal") +
    labs(title = "Fig.17 Area of the County vs. Population",
         subtitle = "Legends on Bottom & Horizontal",
         caption = "Source: ggplot2 midwest Dataset",
         color = "State",
         size = "Density")

apss.density.bottom.h
```



In all the plots, keep the fitted line!!! -3

Part O [7.5 Points]

Filter subset of data with poptotal values > 300,000 and call this dataframe as midwest_sub. Report how many counties satisfy this criterion.

```
midwest_sub = filter(midwest, poptotal > 300000)
nrow(midwest_sub)

## [1] 23
```

There are 23 counties that have a population larger than 300,000.

Part P [5 Points]

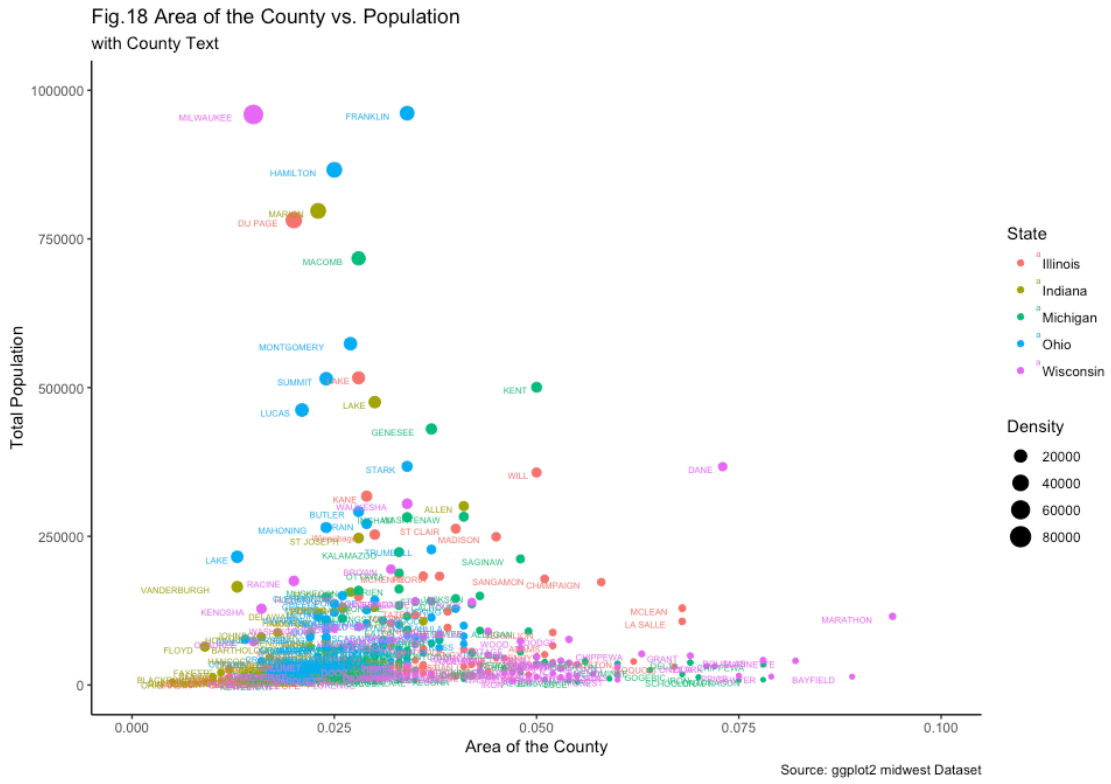
Create a variable in midwest_sub data for large county which satisfies poptotal>300,000 as follows:

```
midwest_sub$large_county <- ifelse(midwest_sub$poptotal > 300000,
midwest_sub$county, "")
```

Part Q [7.5 Points]

In the scatter plot of area versus population, with dots colored for states and dots varying in size for popdensity, add text to highlight the identifies counties using function geom_text().

```
apss.text = apss.density.named +
    geom_text(aes(label = county),
              hjust = 1.4,
              vjust = 1,
              size = 2) +
    labs(title = "Fig.18 Area of the County vs. Population",
         subtitle = "with County Text",
         caption = "Source: ggplot2 midwest Dataset",
         color = "State",
         size = "Density")

apss.text
```

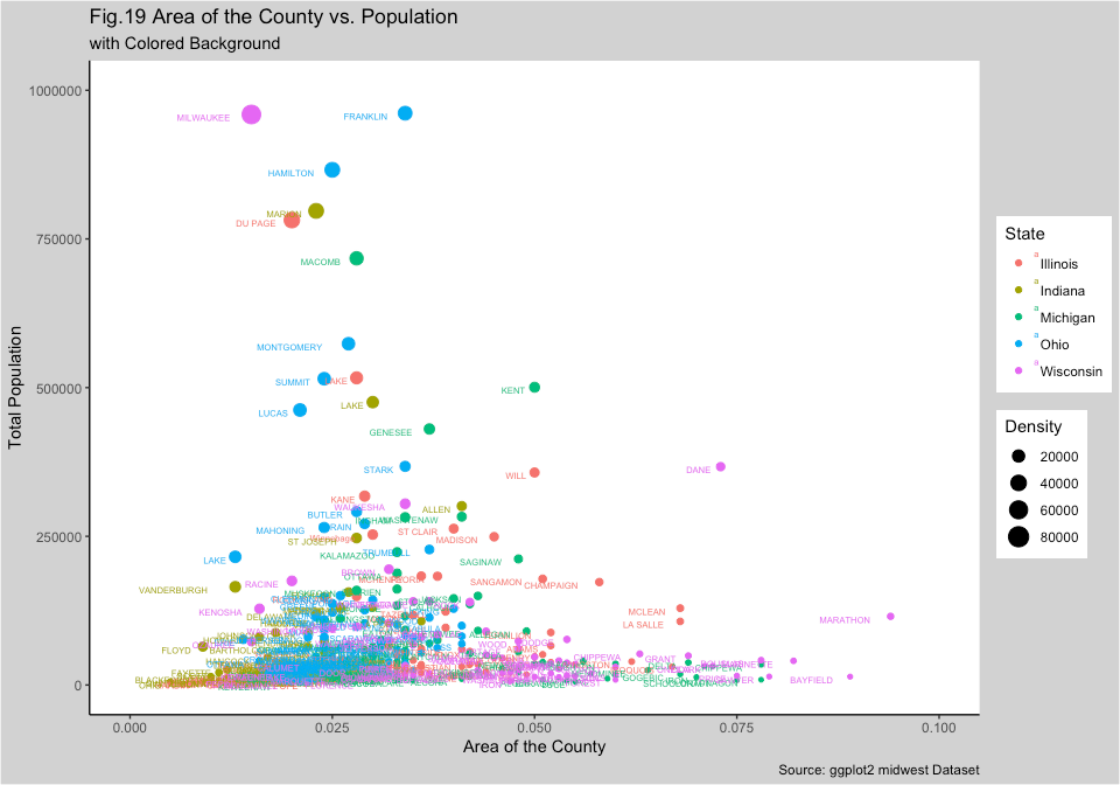


Only add text to large counties (pop > 300000) -2

Part R [5 Points]

Change the background color of your plot.

```
apss.colored = apss.text +
    theme(plot.background = element_rect(fill = "lightgrey")) +
    labs(title = "Fig.19 Area of the County vs. Population",
         subtitle = "with Colored Background",
         caption = "Source: ggplot2 midwest Dataset",
         color = "State",
         size = "Density")

apss.colored
```



Problem 2 [50 points]

Write a summary of your project, including information about:

- **Introduction:** Tell us what problem of interest. Why is this problem interesting and important? Introduce recent research done in area related to your problem. You can pack all this together to motivate us. Do keep it short, to the point, and interesting.
- **Data:** Tell us about the data resource and explain dimensions of the data, variables in the data, and how does this data relate to your research questions.