Lab-1.1: Assignment

Instructions

- Put this folder into a R-project that has access to the `renv` environment
  - Alternatively, you can copy `renv` and `renv.lock` files into this folder
  - If you are on Mac or Linux you can use a soft link `ln -s path_to_renv_folder/ren* ./`
- Work through the code in `Part-1` and make sure you understand it
- `Part-2: Assignment` (re-create the provided plot)

Submission:

- You need to upload ONE document to Canvas when you are done
  - An HTML (or PDF) version of the completed form of this notebook
- The final uploaded version should NOT have any code-errors present
- The final code should be well documented with comments
- All outputs must be visible in the uploaded version, including code-cell outputs, images, graphs, etc to

Part-1: ggplot fundamentals review

In this part of the assignment, we will build up a figure piece by piece, to demonstrate the grammar of graphics.

Set environment

```
renv::restore()
```

```
## * The library is already synchronized with the lockfile.
```

Load library and data

```
# LOAD LIBRARY
library(ggplot2)

# LOAD DATA
data("midwest", package = "ggplot2")  # load the data

#EXPLORE DATA: DATA COMES IN AS A TIBBLE
head(midwest)
```

```
## # A tibble: 6 x 28
##     PID county    state  area poptotal popdens~1 popwh~2 popbl~3 popam~4 popas~5
##   <int> <chr>     <chr> <dbl>    <int>     <dbl>   <int>   <int>   <int>   <int>
## 1   561 ADAMS     IL    0.052    66090     1271.   63917    1702      98     249
## 2   562 ALEXANDER IL    0.014    10626      759     7054    3496      19      48
## 3   563 BOND      IL    0.022    14991      681.   14477     429      35      16
## 4   564 BOONE     IL    0.017    30806     1812.   29344     127      46     150
## 5   565 BROWN     IL    0.018     5836      324.    5264     547      14       5
## 6   566 BUREAU    IL    0.05     35688      714.   35157      50      65     195
## # ... with 18 more variables: popother <int>, percwhite <dbl>, percblack <dbl>,
## #   percamerindan <dbl>, percasian <dbl>, percother <dbl>, popadults <int>,
## #   perchsd <dbl>, percollege <dbl>, percprof <dbl>, poppovertyknown <int>,
## #   percpovertyknown <dbl>, percbelowpoverty <dbl>, percchildbelowpovert <dbl>,
## #   percadultpoverty <dbl>, percelderlypoverty <dbl>, inmetro <int>,
## #   category <chr>, and abbreviated variable names 1: popdensity, 2: popwhite,
## #   3: popblack, 4: popamerindian, 5: popasian
```

Initialize a blank canvas

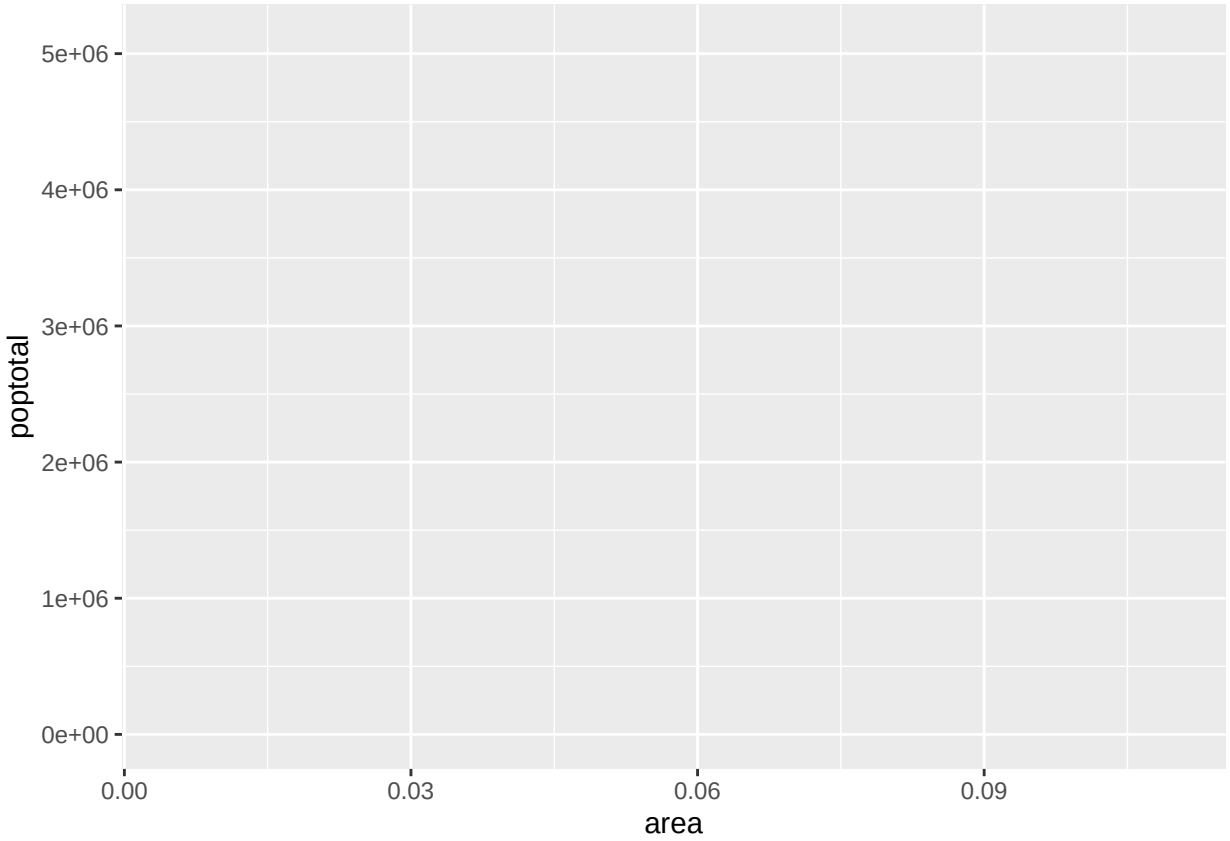
```
g <- ggplot(midwest)
plot(g)
```

Customize canvas

```
g <- ggplot(midwest)+theme(panel.background = element_rect(fill = 'lightblue', color = 'purple'))
plot(g)
```

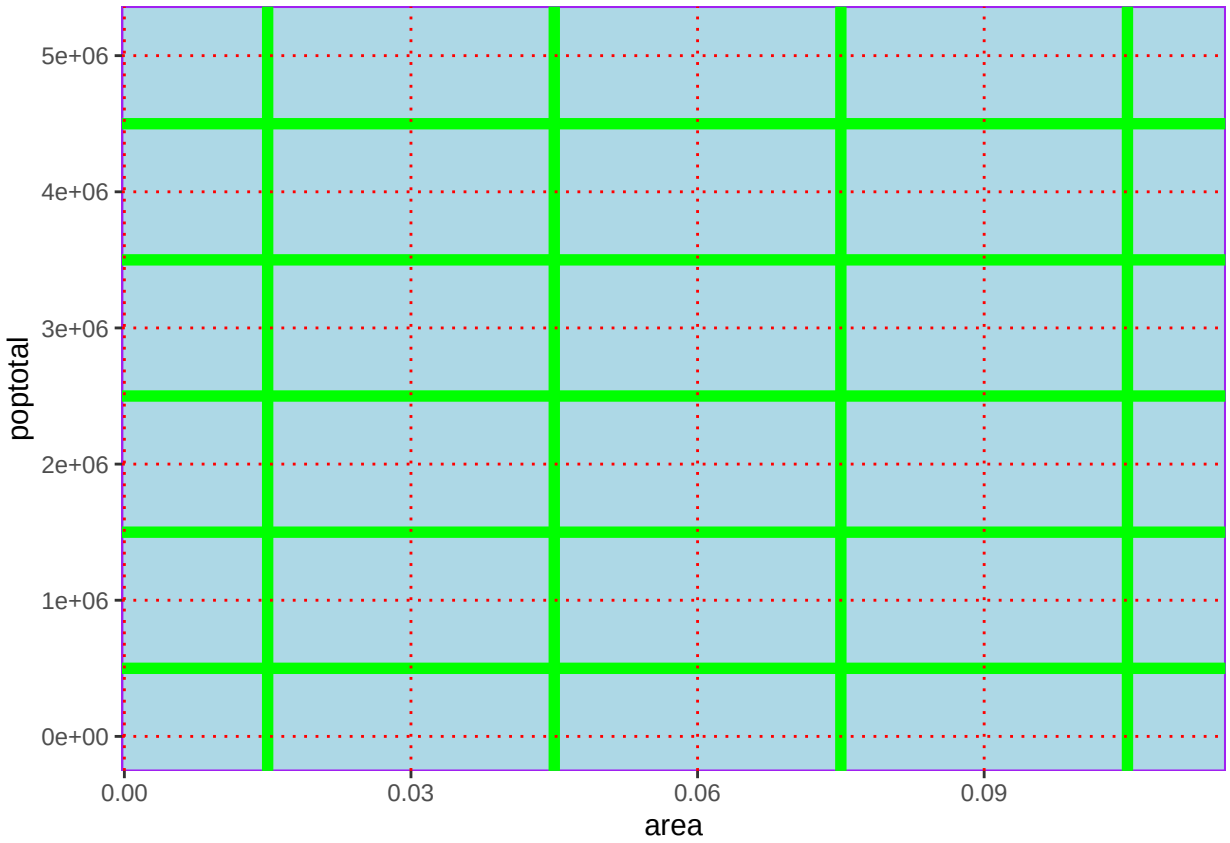
Add coordinate system

```
# aes=Aesthetic mappings describe how variables in the data
# are mapped to visual properties (aesthetics) of geoms.
g <- ggplot(midwest, aes(x=area, y=poptotal))
plot(g)
```



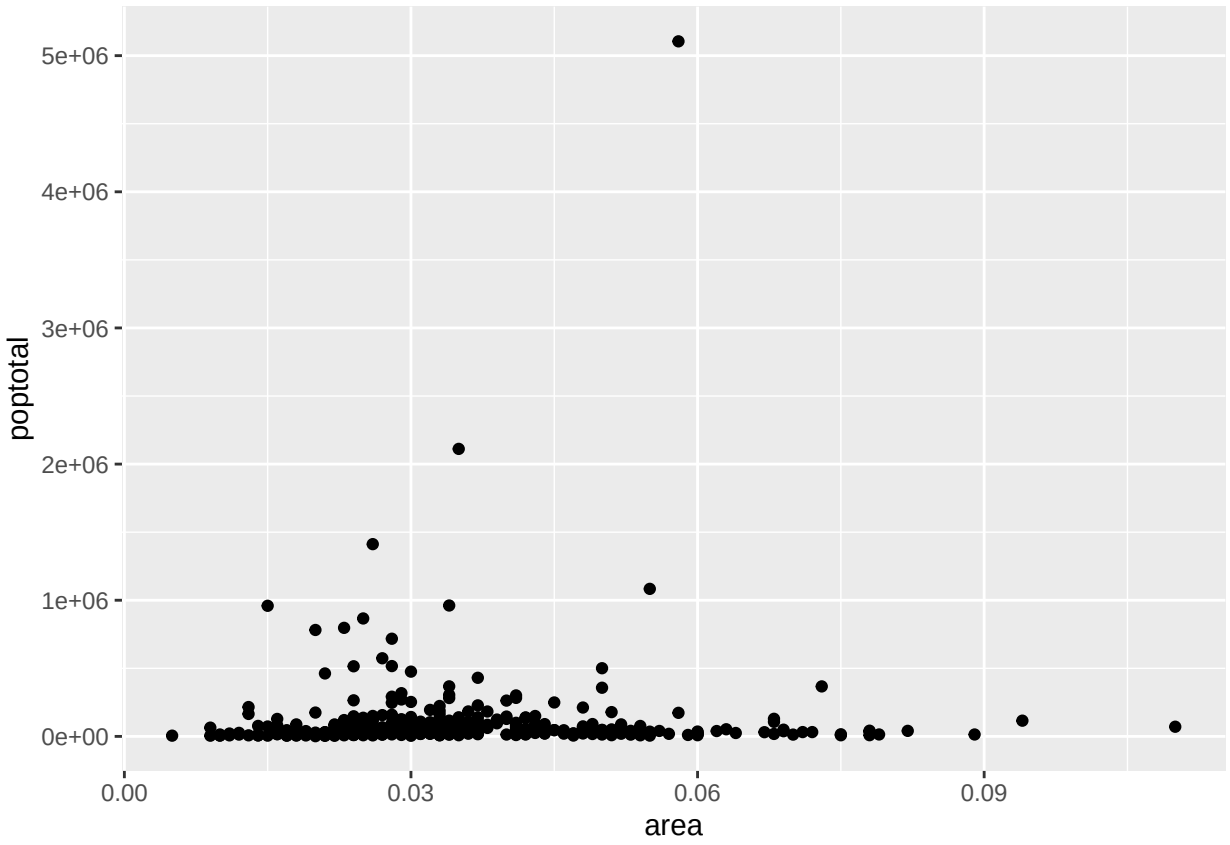
Customize the coordinate system

```
# aes=Aesthetic mappings describe how variables in the data
# are mapped to visual properties (aesthetics) of geoms.
g <- ggplot(midwest, aes(x=area, y=poptotal))
g <- g + theme(panel.background = element_rect(fill = 'lightblue', color = 'purple'),
          panel.grid.major = element_line(color = 'red', linetype = 'dotted'),
          panel.grid.minor = element_line(color = 'green', linewidth = 2))
plot(g)
```



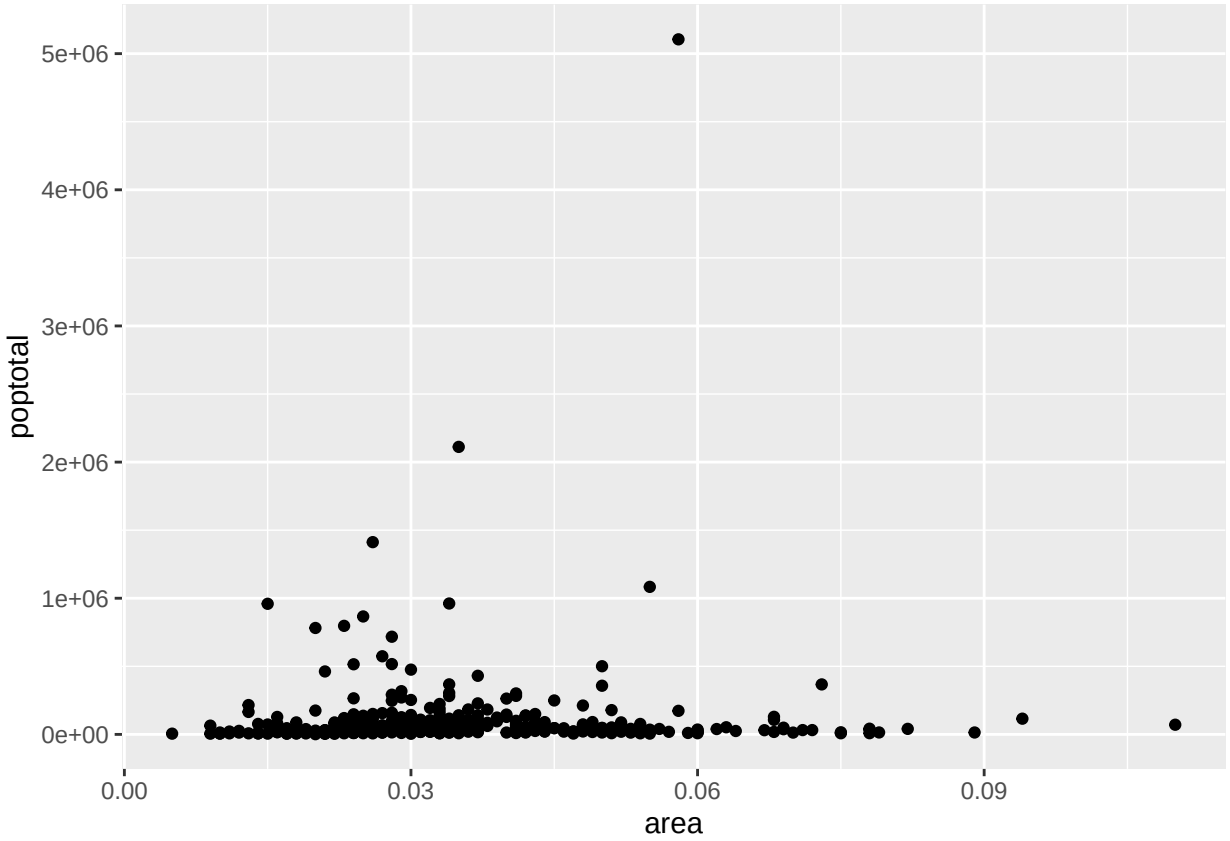
Add some points

```
#SCATTER PLOT-1
# aes=Aesthetic mappings describe how variables in the data
# are mapped to visual properties (aesthetics) of geoms.
g <- ggplot(midwest, aes(x=area, y=poptotal)) + #TELL IT WHAT TO PLOT
geom_point()                                    #DO SCATTER PLOT
plot(g)                                         #GENERATE PLOT
```



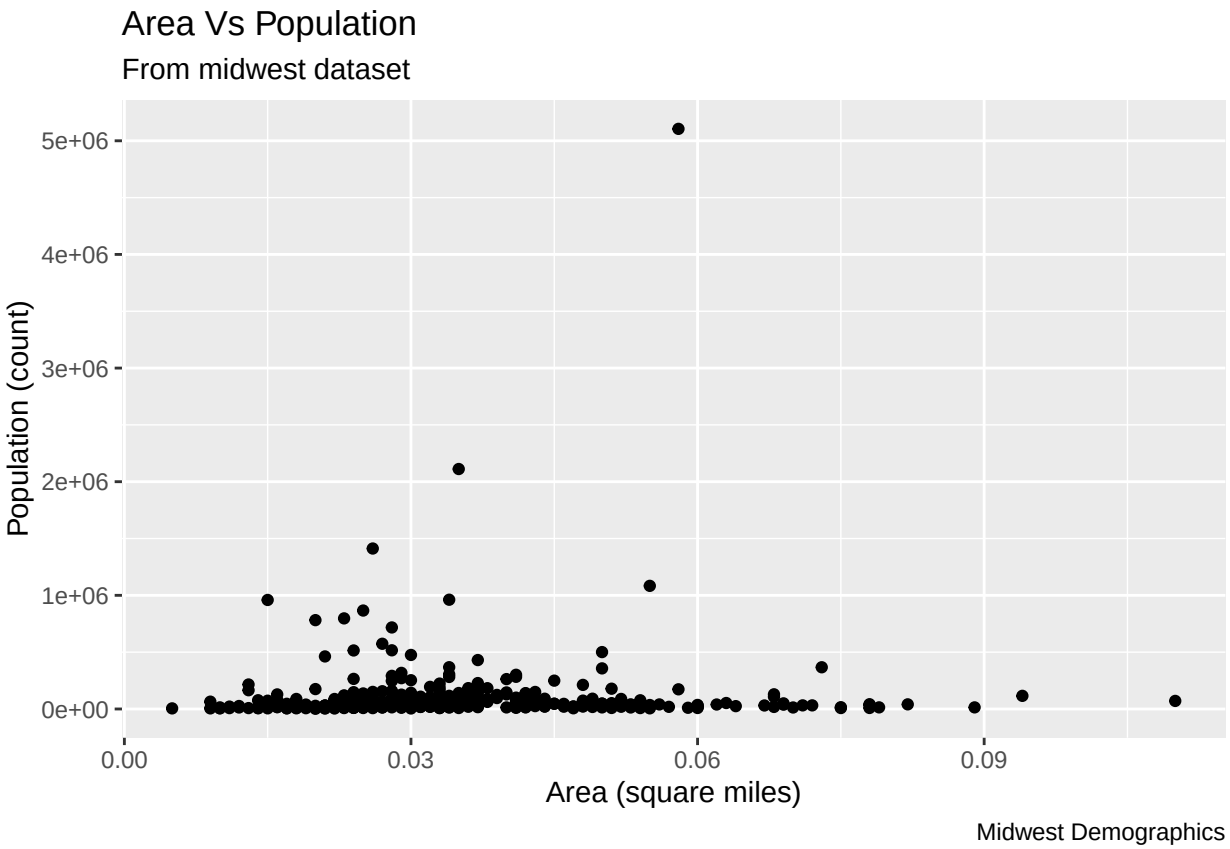
Alternative method

```
#SCATTER PLOT-1 (ALT METHOD)
g <- ggplot(midwest, aes(x=area, y=poptotal))   #TELL IT WHAT TO PLOT
g <- g+geom_point()                             #TELL IT TO DO SCATTER PLOT
plot(g)                                         #GENERATE PLOT
```



Add context with labels

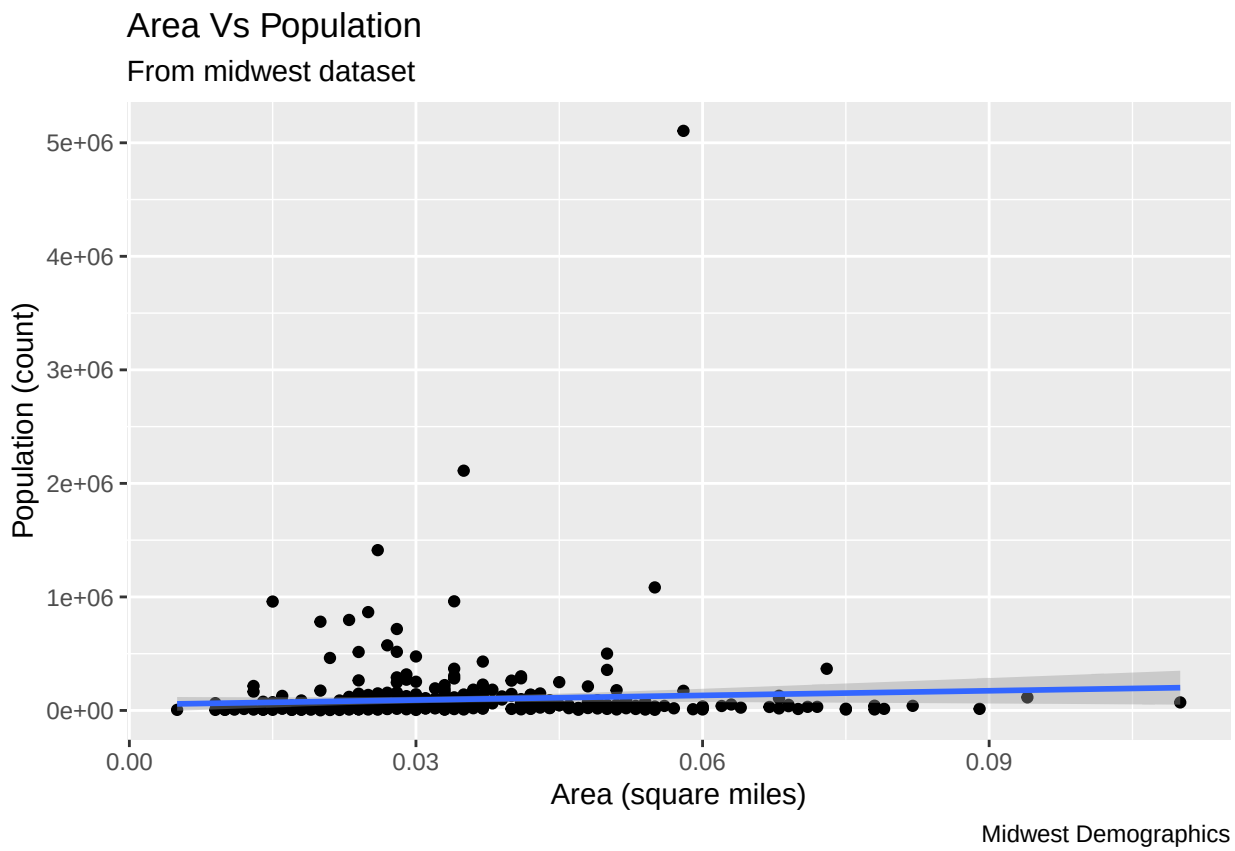
```
#SCATTER PLOT-1 (ALT METHOD)
g <- ggplot(midwest, aes(x=area, y=poptotal))   #TELL IT WHAT TO PLOT
g <- g+geom_point()+                             #TELL IT TO DO SCATTER PLOT
labs(title="Area Vs Population",                   #SET LABELS
    subtitle="From midwest dataset",
    y="Population (count)",
    x="Area (square miles)",
    caption="Midwest Demographics")
plot(g)                                          #GENERATE PLOT
```



Visualize the trend

```
g <- ggplot(midwest, aes(x=area, y=poptotal)) + #TELL IT WHAT TO PLOT
geom_point() +                                  #TELL IT TO DO SCATTER PLOT
labs(title="Area Vs Population",                        #SET LABELS
    subtitle="From midwest dataset",
    y="Population (count)",
    x="Area (square miles)",
    caption="Midwest Demographics") +                                    #DO SCATTER PLOT
# set se=FALSE to turnoff confidence bands
geom_smooth(method="lm")                        #ADD LINEAR BEST FIT LINE
plot(g)                                         #GENERATE PLOT
```

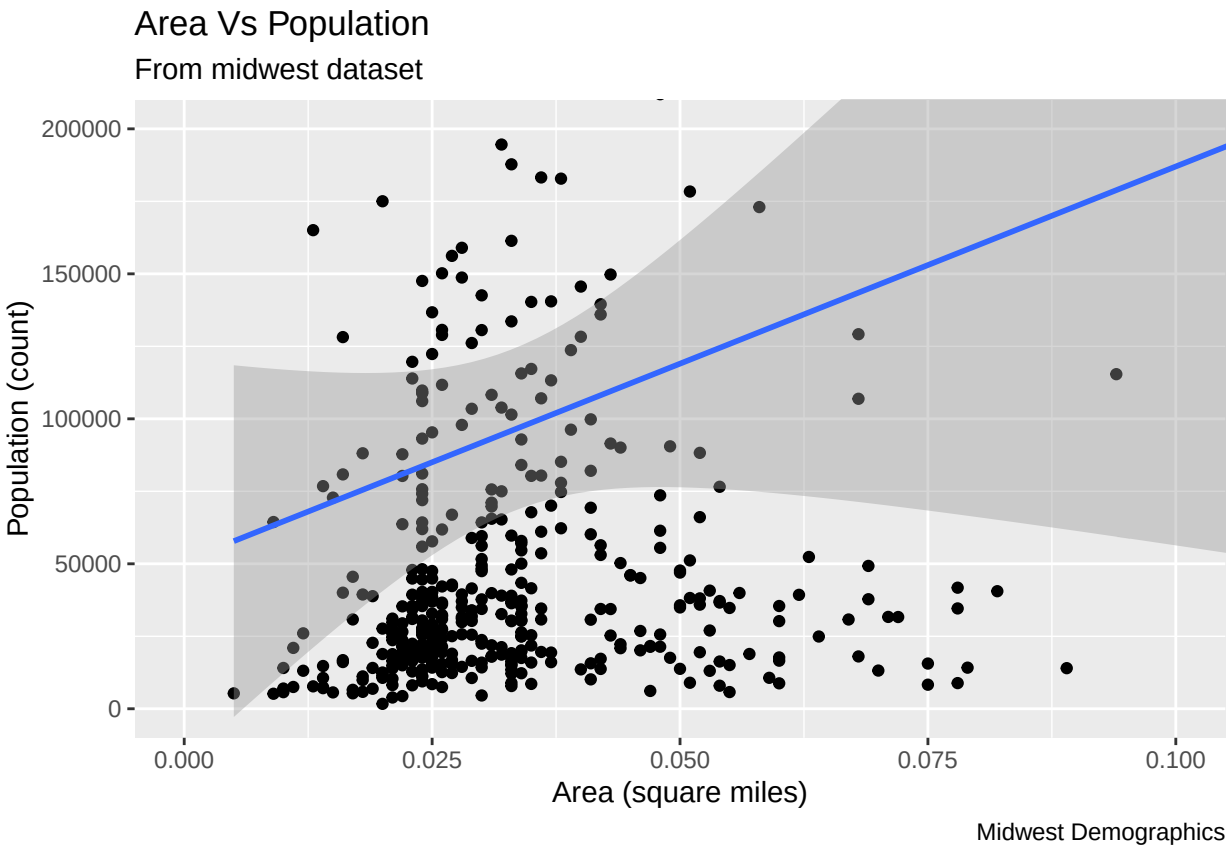
```
## `geom_smooth()` using formula = 'y ~ x'
```



Change axis limits

```
#ADJUST AXIS LIMITS: (ZOOM IN)
#DOESN"T EFFECT POINTS OUTSIDE LIMITS OR THE TREND-LINE
g <- ggplot(midwest, aes(x=area, y=poptotal))+     #TELL IT WHAT TO PLOT
geom_point() +                                     #DO SCATTER PLOT
geom_smooth(method="lm")+                          #ADD LINEAR BEST FIT LINE
coord_cartesian(xlim=c(0,0.1), ylim=c(0, 200000))+ #SET LIMITS
labs(title="Area Vs Population",                   #SET LABELS
subtitle="From midwest dataset",
y="Population (count)",
x="Area (square miles)",
caption="Midwest Demographics")
plot(g)                                            #GENERATE PLOT
```

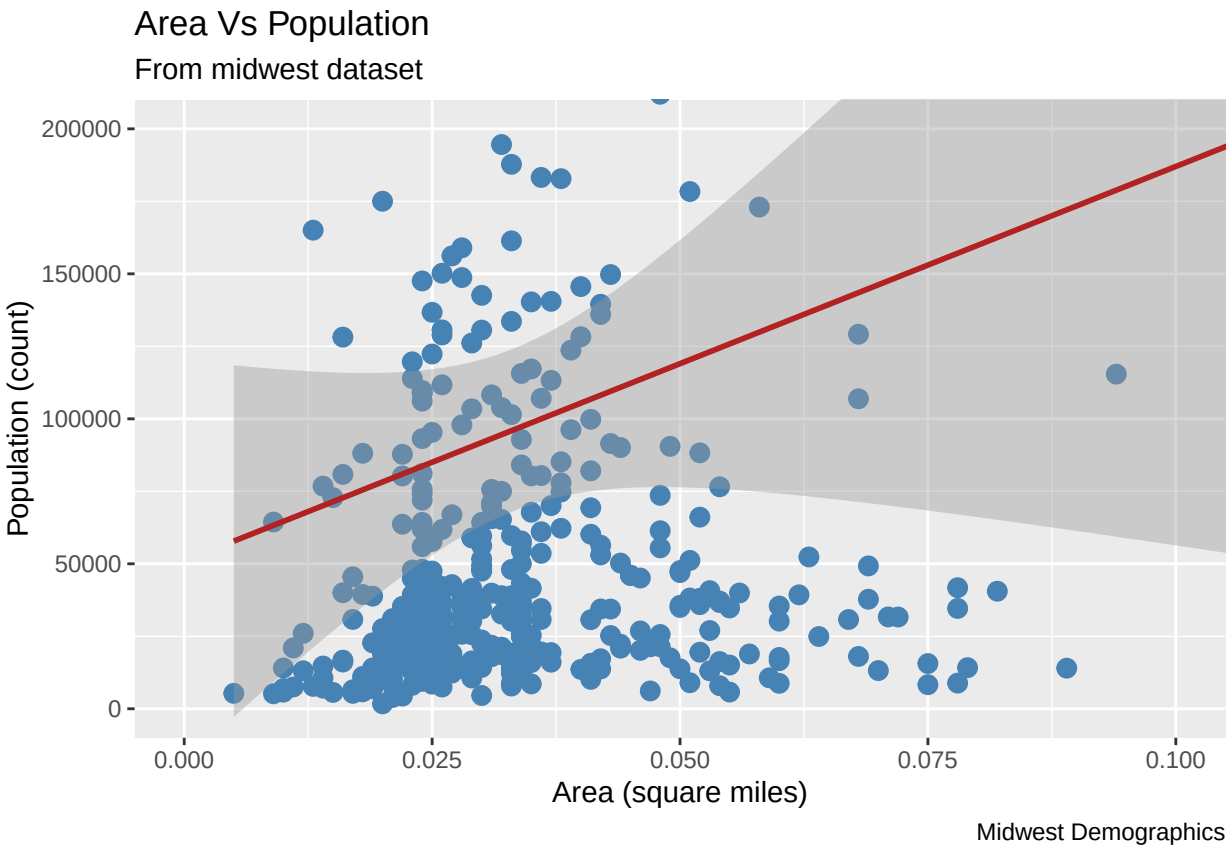
```
## `geom_smooth()` using formula = 'y ~ x'
```



Lets explore some color

```
#POINT SIZE AND COLOR
g <- ggplot(midwest, aes(x=area, y=poptotal))+     #TELL IT WHAT TO PLOT
geom_point(col="steelblue", size=3) +              #DO SCATTER PLOT
geom_smooth(method="lm", col="firebrick")+         #ADD LINEAR BEST FIT LINE
coord_cartesian(xlim=c(0,0.1), ylim=c(0, 200000))+ #SET LIMITS
labs(title="Area Vs Population",                   #SET LABELS
subtitle="From midwest dataset",
y="Population (count)",
x="Area (square miles)",
caption="Midwest Demographics")
plot(g)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



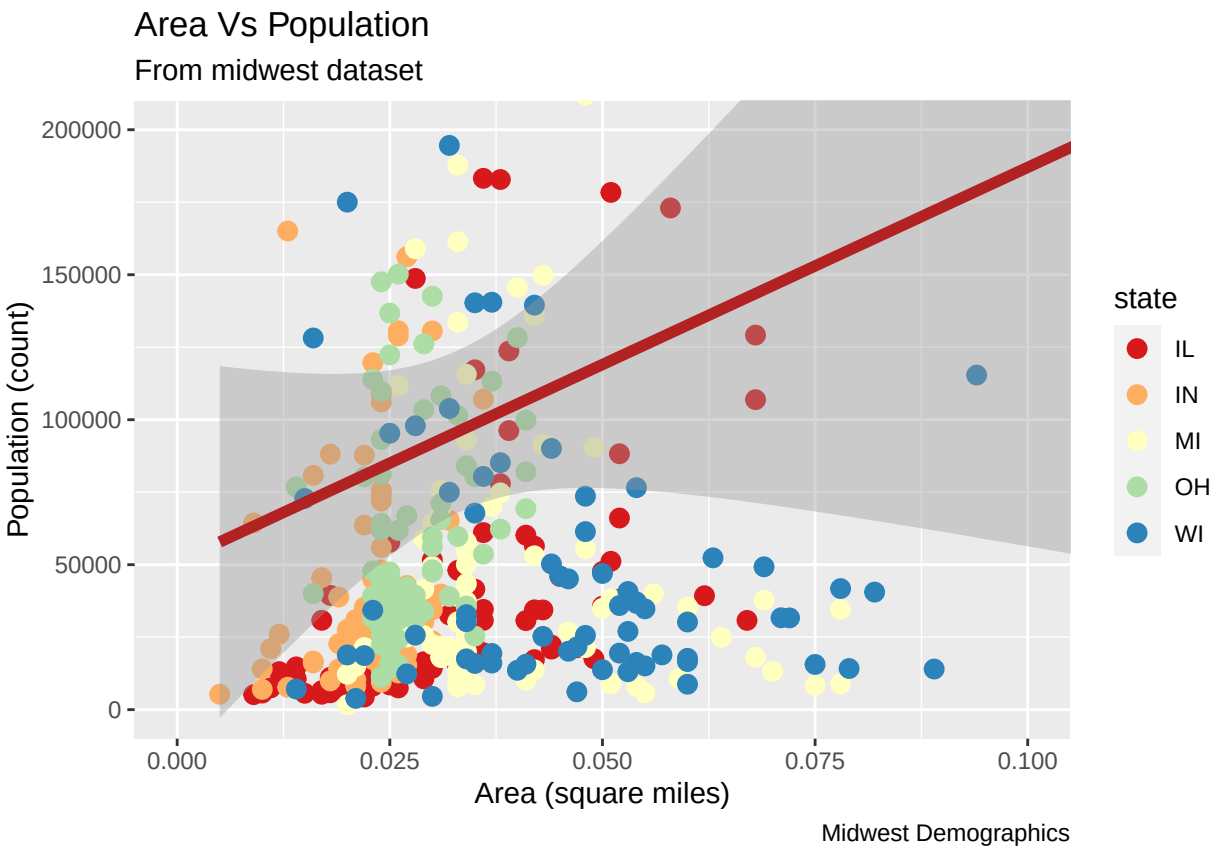
Variable color encoding

```
#COLOR BASED ON ANOTHER COLUMN
# Not just color, but size, shape, stroke (thickness of boundary)
# and fill (fill color) can be used to discriminate groupings.
g <- ggplot(midwest, aes(x=area, y=poptotal))+     #TELL IT WHAT TO PLOT
geom_point(aes(col=state), size=3) +               #DO SCATTER PLOT
#LEGEND ADDED AUTOMATICALLY
# theme(legend.position="None")+                   #REMOVE LEGEND
#se=FALSE  --> TURN OFF CONFIDENCE BANDS
geom_smooth(method="lm", col="firebrick", size=2)+ #ADD LINEAR BEST FIT LINE
coord_cartesian(xlim=c(0,0.1), ylim=c(0, 200000))+ #SET LIMITS
labs(title="Area Vs Population",                   #SET LABELS
subtitle="From midwest dataset",
y="Population (count)",
x="Area (square miles)",
caption="Midwest Demographics")+
scale_colour_brewer(palette = "Spectral")          #CHANGE COLOR PALETTE
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
```

```
plot(g)                                                    #GENERATE PLOT
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Part-3: Assignment (plot re-creation)

- Insert code below to re-create the following plots EXACTLY
  - use `palette = "Pastel1"`
- **Do NOT use any built in themes**
- Make sure to comment your code carefully

```
# INSERT YOUR SOLUTION HERE
g <- ggplot(midwest, aes(x=area, y=poptotal)) + #info
geom_point(aes(col=state), size=3) + #scatterplot geom
geom_smooth(method="lm", col="deepskyblue3", size=2) + #trendline geom
theme(panel.background = element_blank(), #panel
      panel.border = element_rect(color="black", fill=NA),
      axis.text.x = element_text(angle=90, vjust=0.5, hjust=1, size=15),
      axis.text.y = element_text(size=15),
      axis.title = element_text(size=15)) +
coord_cartesian(xlim=c(0, 0.1), ylim=c(0, 600000)) + #limits
scale_x_continuous(breaks=seq(0,0.1,0.02)) + #custom breaks
```

```
scale_y_continuous(breaks=seq(0,600000,200000),
                   labels=c("0K","200K","400K","600K")) +
labs(title="Area Vs Population", #labels
     subtitle="From midwest dataset",
     y="Population (count)",
     x="Area (square miles)",
     caption="Midwest Demographics") +
scale_colour_brewer(palette = "Pastel1")          #color palette

plot(g)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```