Introduction to Data Visualization in R

Libraries

- You need ggplot2!!!

```r
# Use the commented command below if you do not have ggplot2 installed
# install.packages("ggplot2")
library(ggplot2)
```

Overall Goal

- Create a scatter plot with plot
- Create plots with ggplot2 and compare
- Look at real world applications

Great Resources

- Tutorial based on r-statistics.co by Selva Prabhakaran
- Note: tutorial had library(ggplot2) at each step. Only needed once per session
- How to make any plot in ggplot2
- http://r-statistics.co/ggplot2-Tutorial-With-R.html
- The Complete ggplot2 Tutorial Part1: Introduction To ggplot2
- http://r-statistics.co/Complete-Ggplot2-Tutorial-Part1-With-R-Code.html
- The Complete ggplot2 Tutorial Part 2: How To Customize ggplot2
- http://r-statistics.co/Complete-Ggplot2-Tutorial-Part2-Customizing-Theme-With-R-Code.html
- Top 50 ggplot2 Visualizations: The Master List
- http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html
- ggplot2 Quickref
- http://r-statistics.co/ggplot2-cheatsheet.html

ggplot2

- Very aesthetically pleasing graphics framework in R
- Making plots in ggplot2 is different from base graphics
- Can be more problematic but can do more

ggplot vs plot

- The syntax for constructing ggplots can be more intensive than plot
- Main difference: unlike base graphics, ggplot works with dataframes and not individual vectors
- All the data needed to make the plot is typically be contained within the dataframe supplied to the ggplot() itself or can be supplied to respective geoms
- Also, you can keep enhancing the plot by adding more layers (and themes) to an existing plot created using the ggplot() function

GGplot2 dataset: midwest

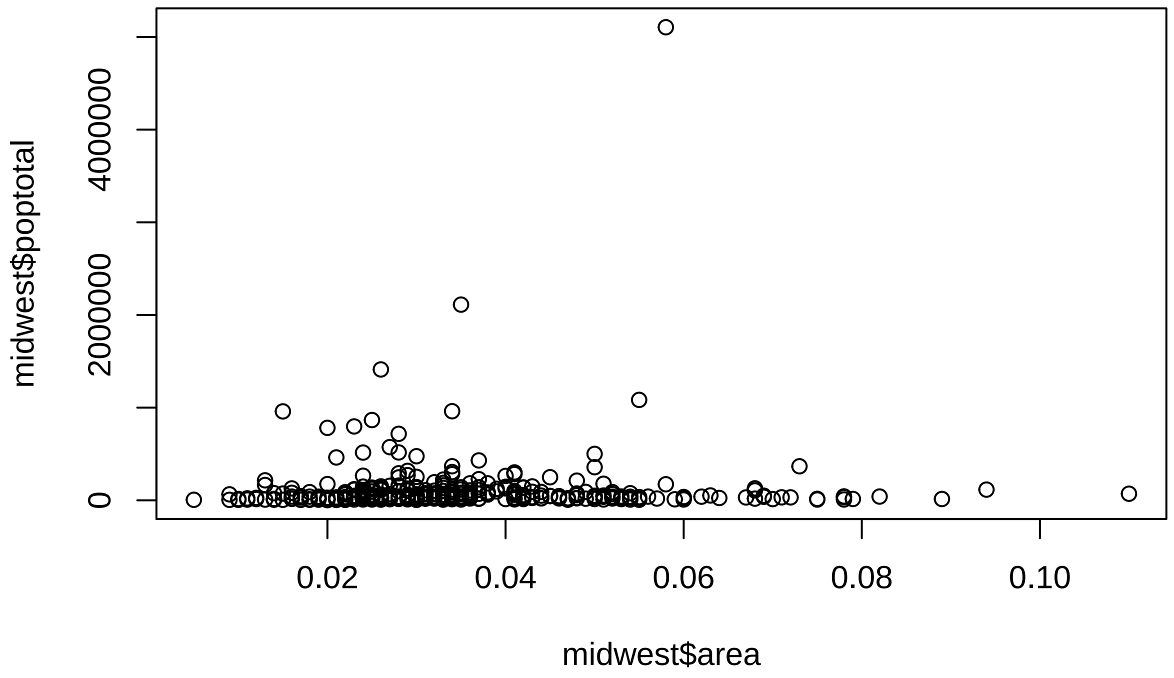
- Demographic information of midwest counties from 2000 US census
- https://ggplot2.tidyverse.org/reference/midwest.html

```
library(ggplot2)

# turn off scientific notation like 1e+06 library(ggplot2)
options(scipen=999)

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

names(midwest)
```

```
##  [1] "PID"                  "county"               "state"
##  [4] "area"                 "poptotal"             "popdensity"
##  [7] "popwhite"             "popblack"             "popamerindian"
## [10] "popasian"             "popother"             "percwhite"
## [13] "percblack"            "percamerindan"        "percasian"
## [16] "percother"            "popadults"            "perchsd"
## [19] "percollege"           "percprof"             "poppovertyknown"
## [22] "percpovertyknown"     "percbelowpoverty"     "percchildbelowpovert"
## [25] "percadultpoverty"     "percelderlypoverty"   "inmetro"
## [28] "category"
```

- Scatter plot with plot (area=area of county and poptotal=total population)

```
plot(midwest$area, midwest$poptotal)
```



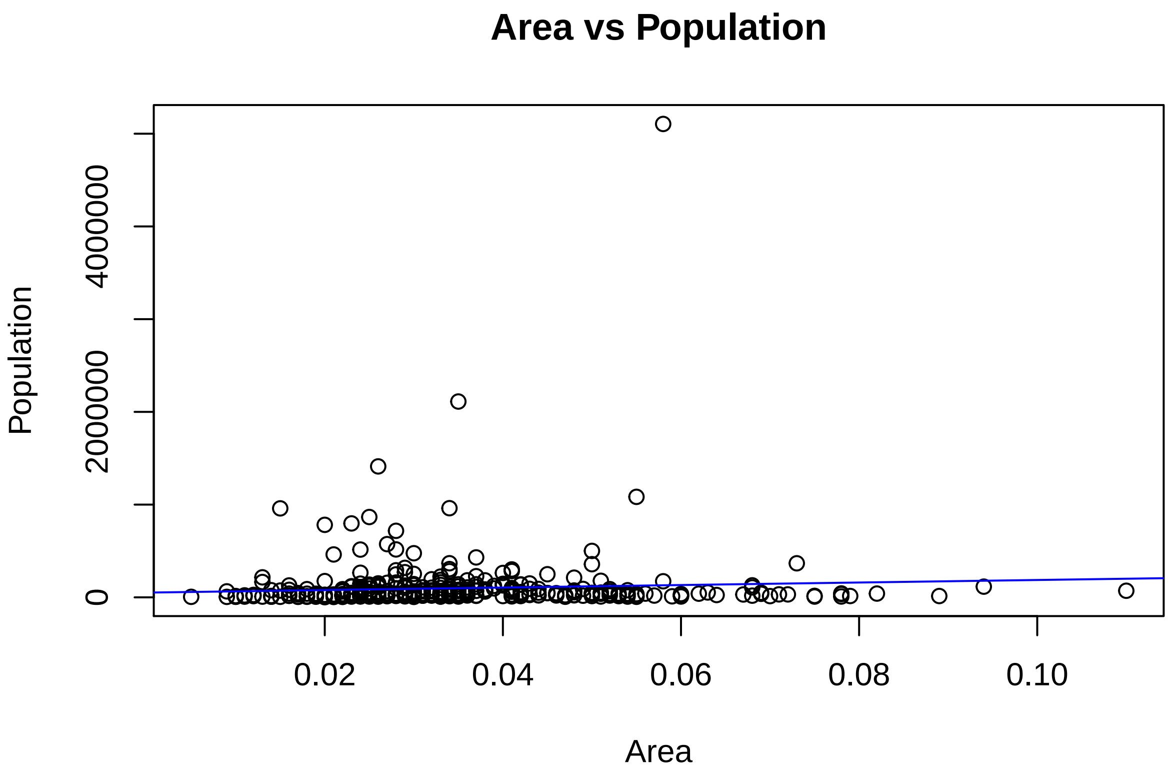
Scatter plot with line

- To plot area vs poptotal, want to fit a line to the data

```
# Consider linear regression for poptotal
modelLM<-summary(lm(poptotal~area, data=midwest))
modelLM
```

```
##
## Call:
## lm(formula = poptotal ~ area, data = midwest)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -158055  -71403  -56193  -20778 4975166
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    51018      35241   1.448    0.148
## area         1360059     971750   1.400    0.162
##
## Residual standard error: 297800 on 435 degrees of freedom
## Multiple R-squared:  0.004483,   Adjusted R-squared:  0.002194
## F-statistic: 1.959 on 1 and 435 DF,  p-value: 0.1623
```

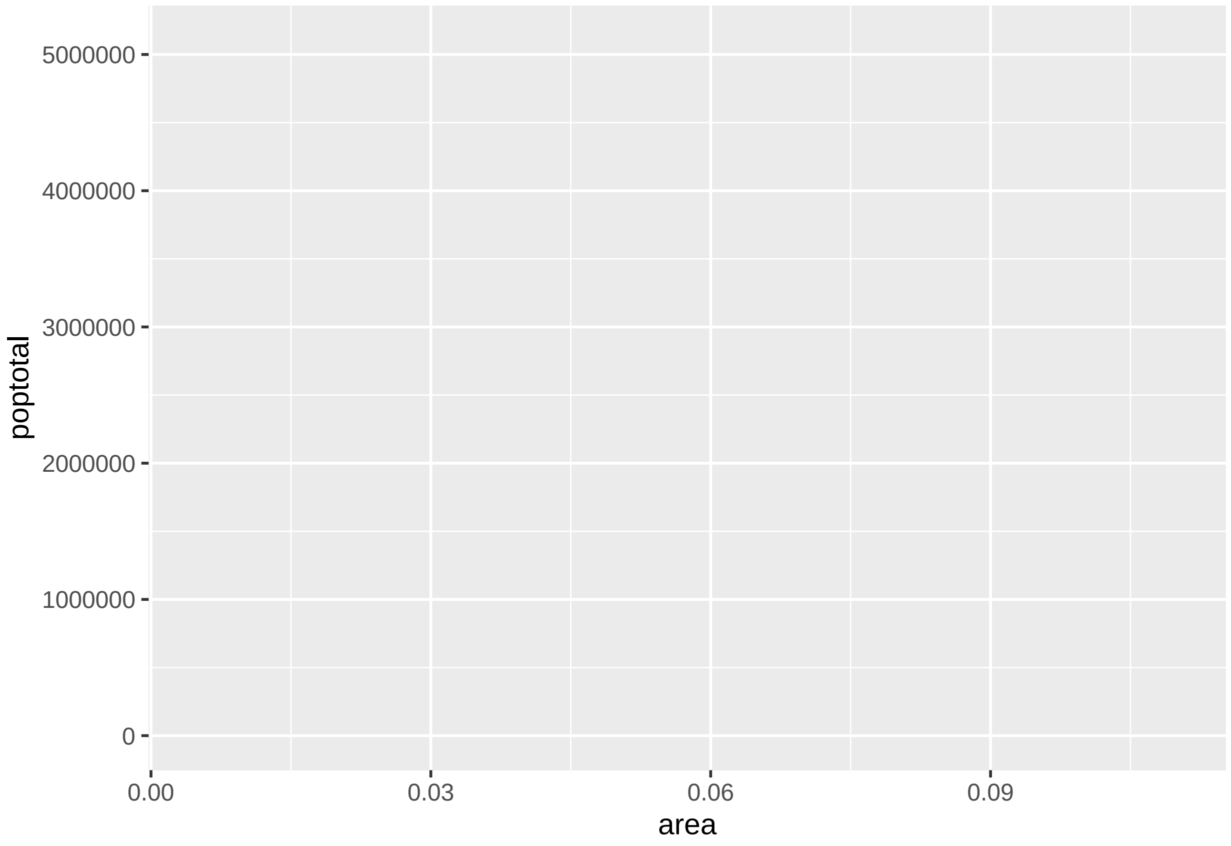
```
plot(midwest$area, midwest$poptotal, main="Area vs Population", xlab="Area", ylab="Population")

# add a line to the data, intercept and slope
abline(coef=c(modelLM$coef[1,1],modelLM$coef[2,1]),col="blue")
```



Setup with ggplot2

- Blank plot in ggplot2

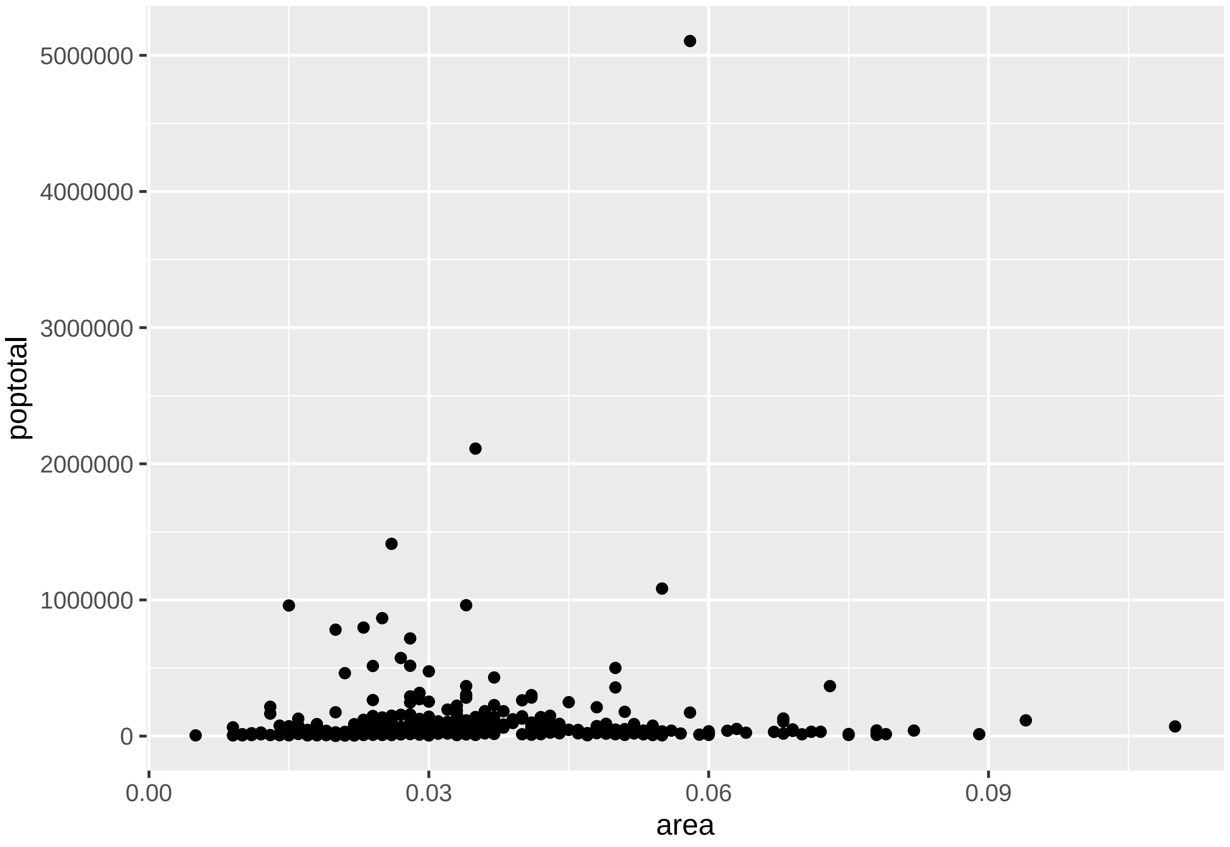
```r
# Intialize the plot
# area and poptotal are columns in 'midwest'
#  aes() function is used to specify the X axis as area and Y axis as poptotal
ggplot(midwest, aes(x=area, y=poptotal))
```



Scatter plot

- Create a scatter plot where each point represents a county
- Use geom_point() to add data points to graph
- There are many such geom layers that can be used

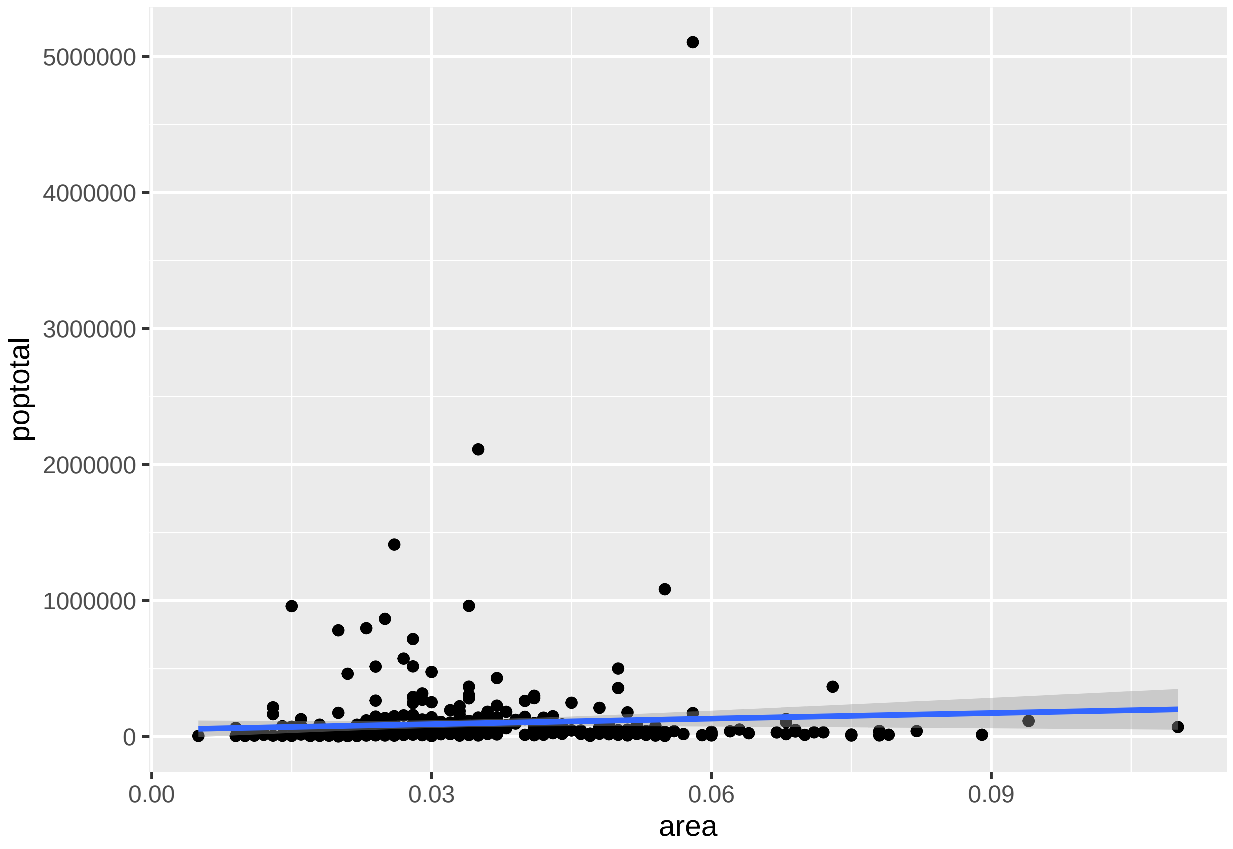
```
# Create a scatterplot on top of the blank ggplot by adding points
# Using a geom layer called geom_point.
ggplot(midwest, aes(x=area, y=poptotal)) + geom_point()
```



Scatter plot with a line

- Add a line
- geom_smooth(method='lm')
- lm stands for linear model, adding a line

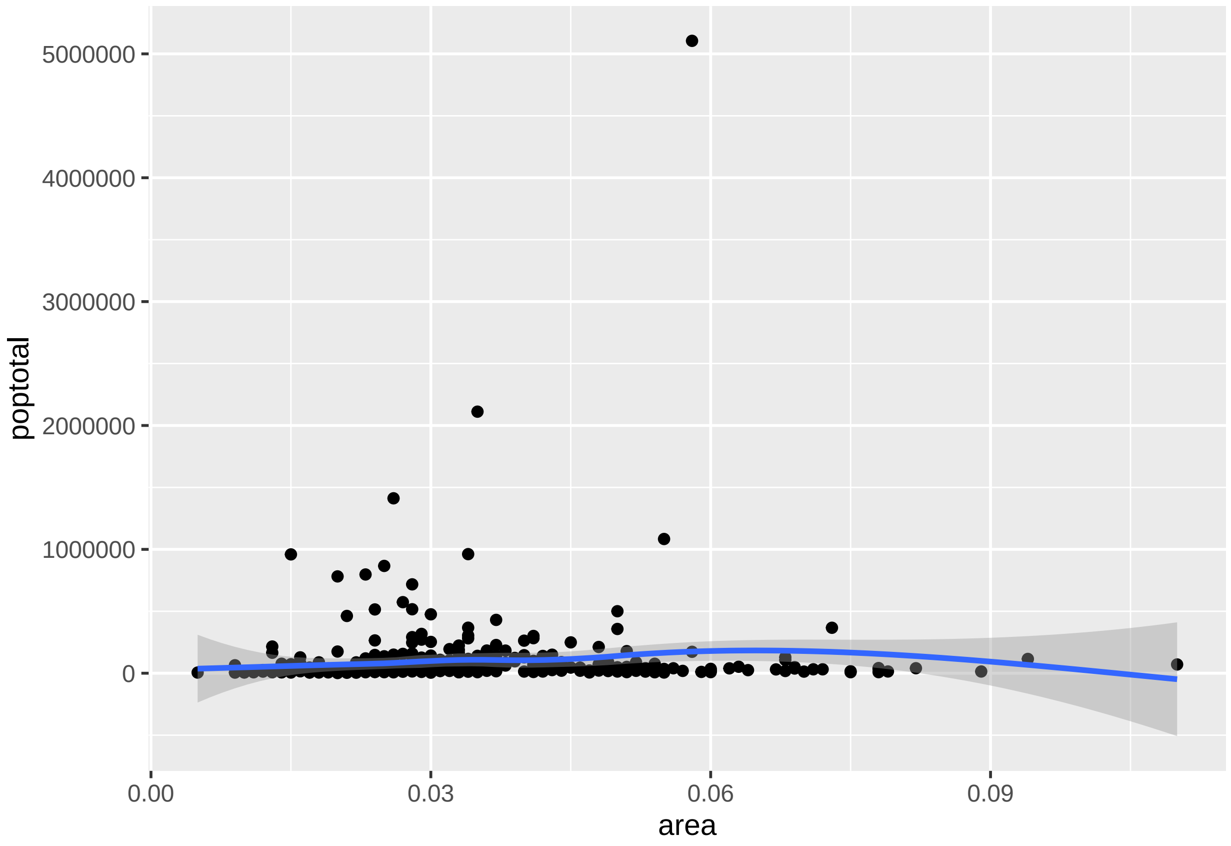
```r
g <- ggplot(midwest, aes(x=area, y=poptotal)) + geom_point() + geom_smooth(method="lm")

# set se=FALSE to turnoff confidence bands
plot(g)
```



Scatter plot with a smoothed line

- ?geom_smooth to see other methods
- loess for locally weighted scatterplot smoothing
- Good way to see trends
- https://www.statisticshowto.com/lowess-smoothing/

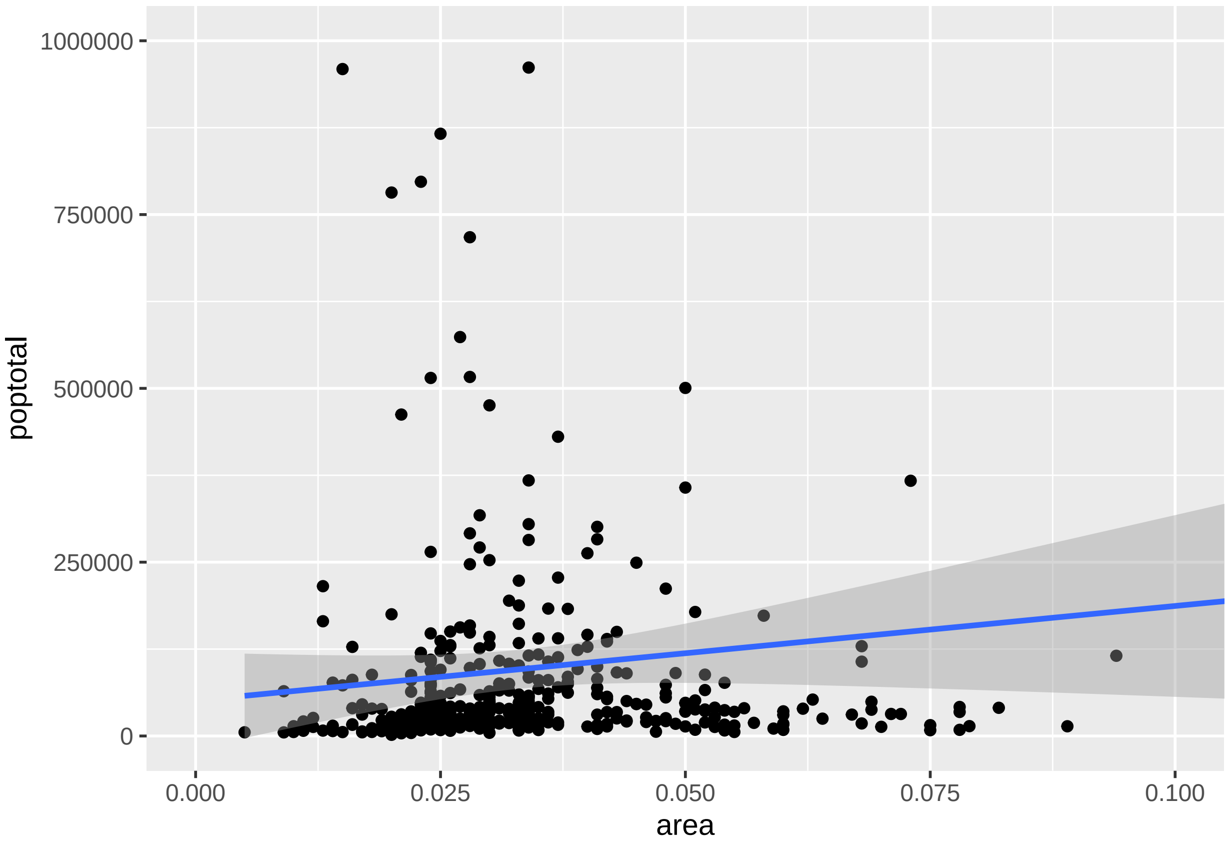
```
g <- ggplot(midwest, aes(x=area, y=poptotal)) + geom_point() + geom_smooth(method="loess")

# set se=FALSE to turnoff confidence bands
plot(g)
```



Changing the range of x and y axis

- Majority of points lie in the bottom of the plot which doesn't really look nice
- Change the Y-axis limits to focus on the lower half
- Or change the X and Y axis limits by zooming in to the region of interest without deleting the points by using coord_cartesian()
- Before ylim went to 5000000, change to 1000000, keep xlim the same
- Since all points are considered, the line of best fit does not change

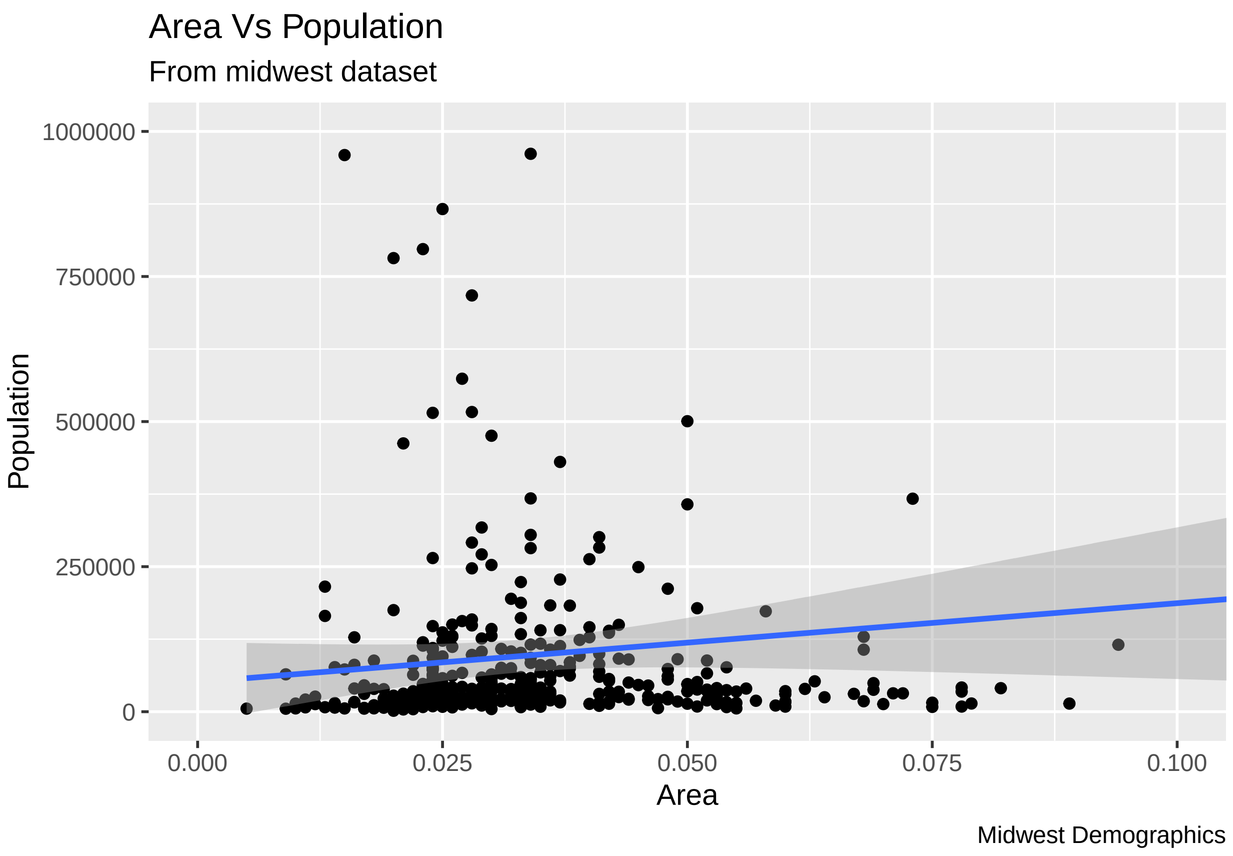
```
g <- ggplot(midwest, aes(x=area, y=poptotal)) + geom_point() + geom_smooth(method="lm")

# Zoom in without deleting the points outside the limits
# As a result, the line of best fit is the same as the original plot.
 # zooms in plot(g1)
g1 <- g + coord_cartesian(xlim=c(0,0.1), ylim=c(0, 1000000))
plot(g1)
```



How to change the title and axis labels

- Add the plot title and labels for X and Y axis
- Done using the labs() function with title, x and y arguments
- Another option is to use the ggtitle(), xlab() and ylab()

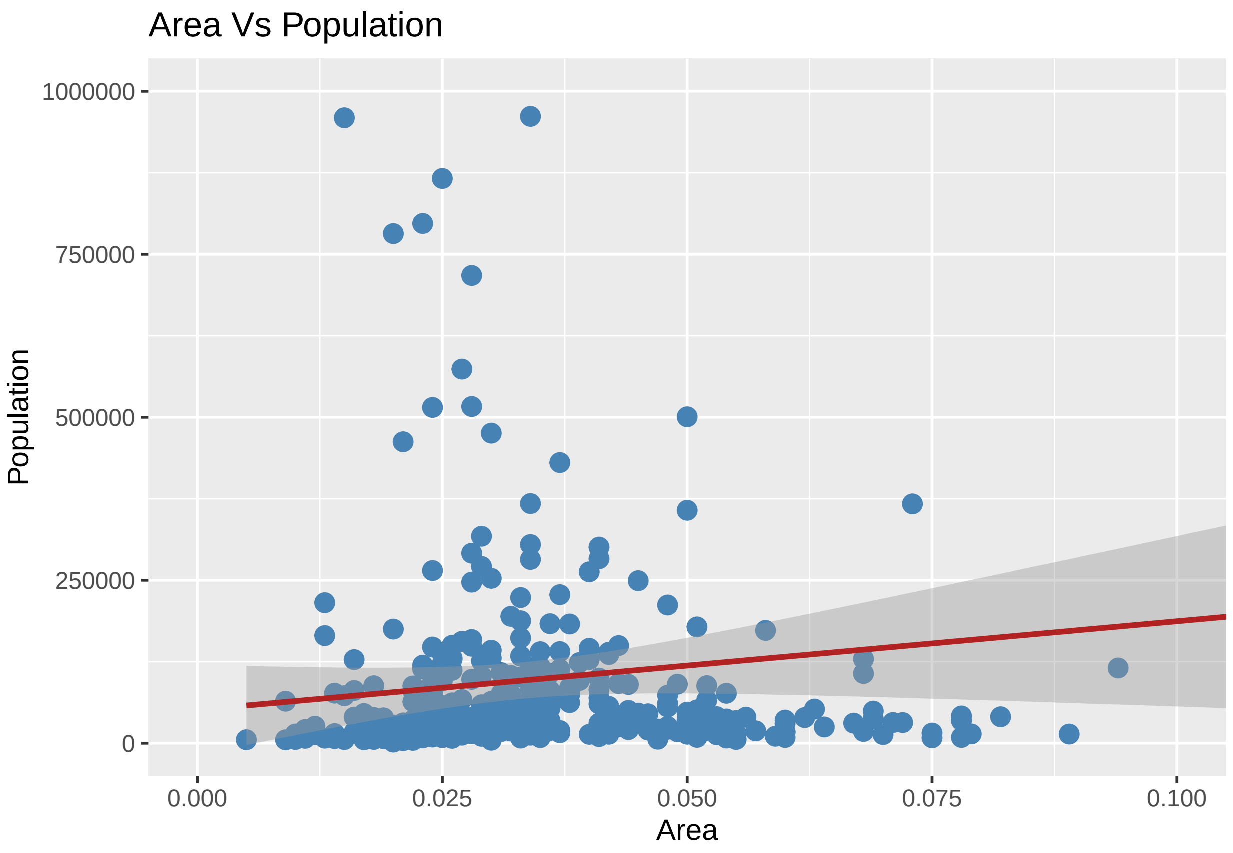
```
ggplot(midwest, aes(x=area, y=poptotal)) +
geom_point() +
geom_smooth(method="lm") +
coord_cartesian(xlim=c(0,0.1), ylim=c(0, 1000000)) +
labs(title="Area Vs Population", subtitle="From midwest dataset", y="Population", x="Area", caption="Mi
```



Changing the color and size of points

- Set the color to steel blue and change the size to 3
- Change the line color to fire brick red

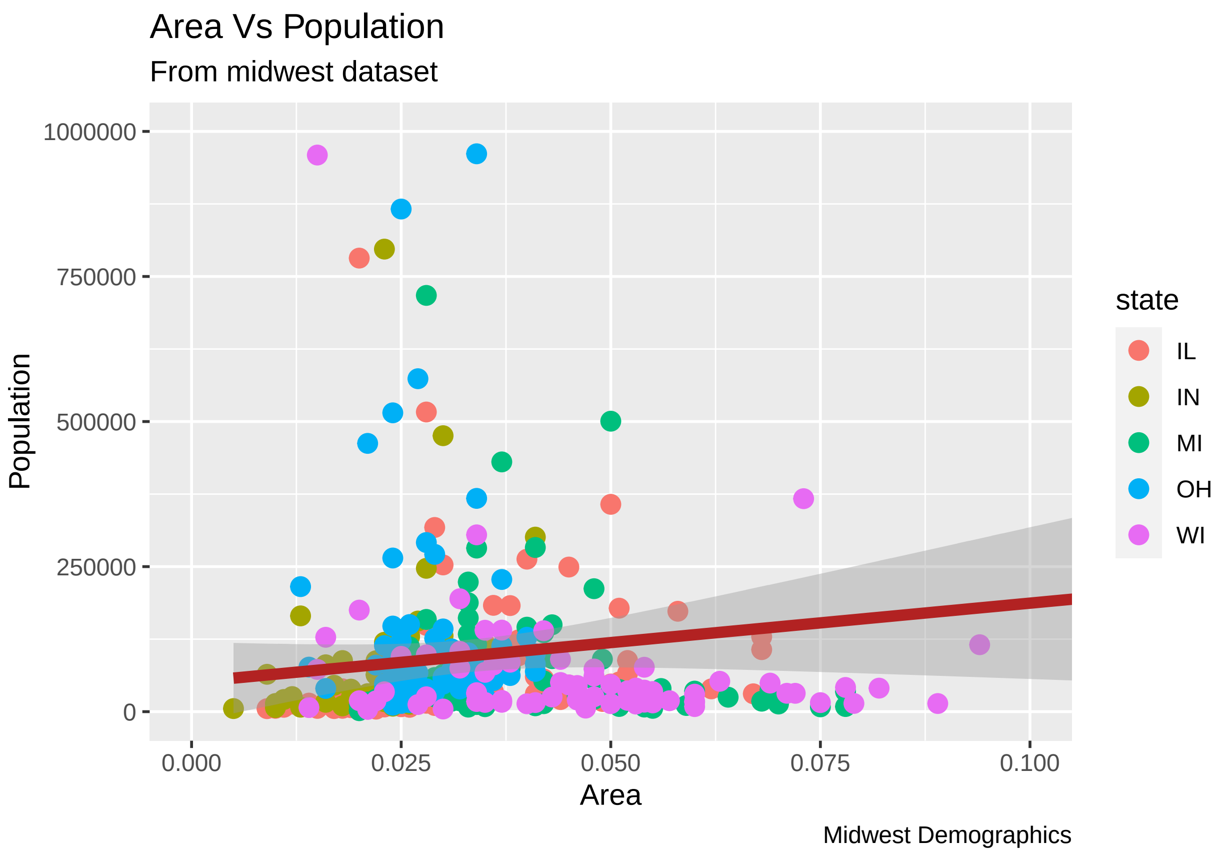
```
ggplot(midwest, aes(x=area, y=poptotal)) +
geom_point(col="steelblue", size=3) +
geom_smooth(method="lm", col="firebrick") +
coord_cartesian(xlim=c(0, 0.1), ylim=c(0, 1000000)) +
labs(title="Area Vs Population", y="Population", x="Area")
```



Changing the color by third variable

- Have the color vary by state (col=state)
- Legend is added automatically. Remove with gg + theme(legend.position="None")

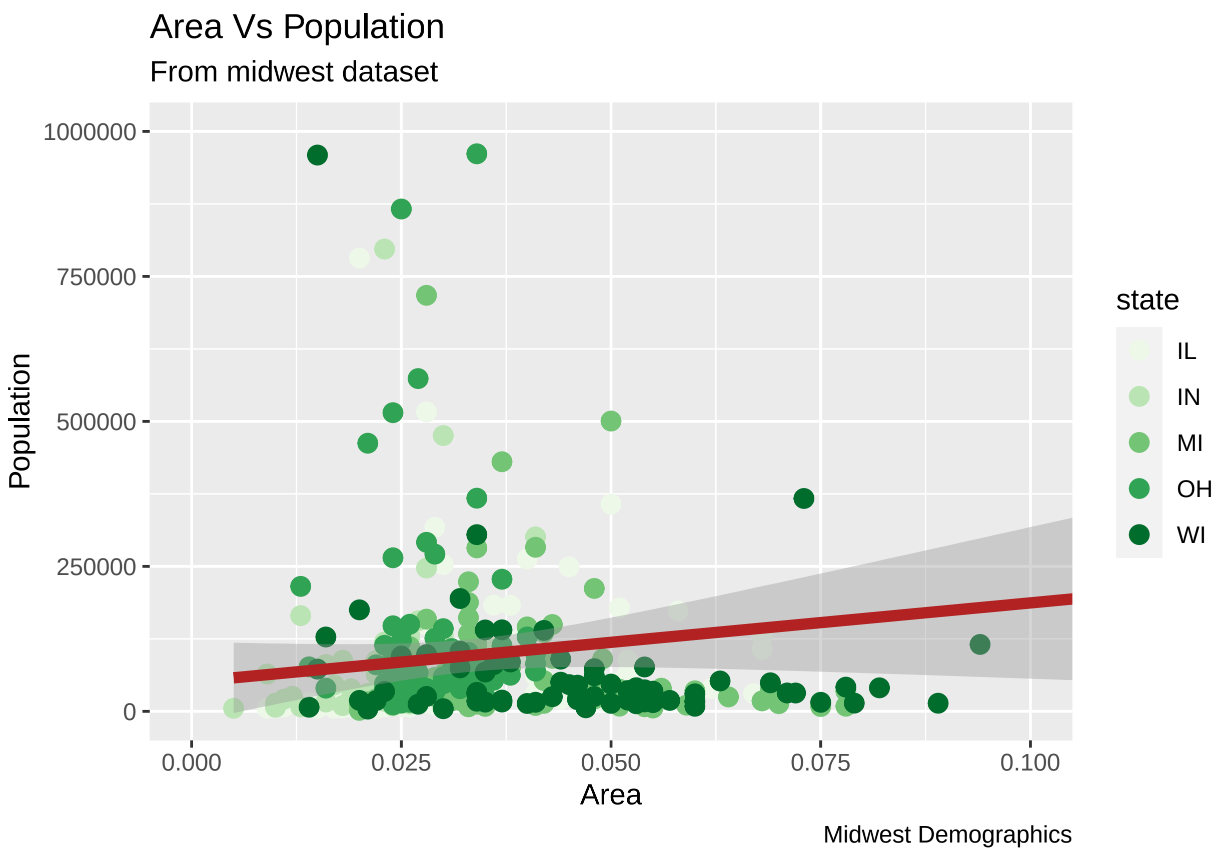
```
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
geom_point(aes(col=state), size=3) +
geom_smooth(method="lm", col="firebrick", size=2) +
coord_cartesian(xlim=c(0, 0.1), ylim=c(0, 1000000)) +
labs(title="Area Vs Population", subtitle="From midwest dataset", y="Population", x="Area", caption="Mi
plot(gg)
```



Color palette

- Change the color palettes similar to the RColorBrewer package
- https://ggplot2.tidyverse.org/reference/scale_brewer.html

```
gg + scale_colour_brewer(palette = "Set4")
```



So much more

Tutorial 1

- http://r-statistics.co/Complete-Ggplot2-Tutorial-Part1-With-R-Code.html
- Change the X axis texts and ticks location
- Customize the entire theme using pre-built themes

Tutorial 2

- http://r-statistics.co/Complete-Ggplot2-Tutorial-Part2-Customizing-Theme-With-R-Code.html
- Change plot and axis title styles
- Modify the legend: change the legend order or position
- Add text labels and annotation
- Add an image in the background

Tutorial 3

- http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html
- So many more plots in options

R resources

- Harvard Catalyst or other online Harvard courses https://online-learning.harvard.edu/subject/r

- There are online courses through coursera https://www.coursera.org/learn/r-programming

- Software carpentry offers really fun 2 day workshops. You can check when there is one in Boston and make sure to sign up right away because they fill up quickly https://software-carpentry.org/workshops/