Data Engineering
COMP2031/8031

- Topic Coordinator: Dr Mehwish Nasim
- Office: 3.13 Tonsley

Flinders
UNIVERSITY

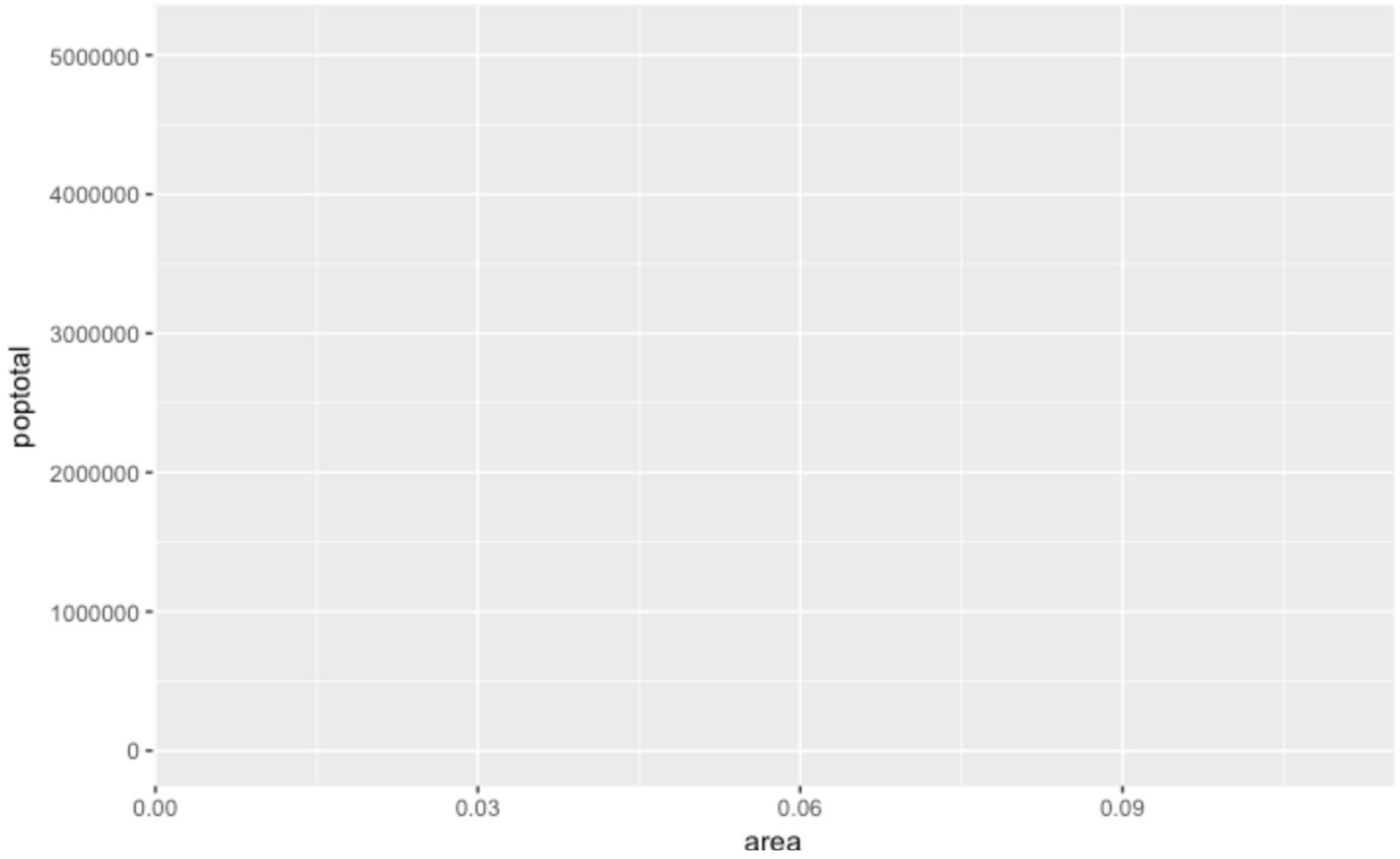
ggplot2

- 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.

Basic ggplot

- aes() function is used to specify the X and Y axes. That's because, any information that is part of the source dataframe has to be specified inside the aes() function.

```
# Setup
options(scipen=999)  # turn off scientific notation like 1e+06
library(ggplot2)
data("midwest", package = "ggplot2")  # load the data
# midwest <- read.csv("http://goo.gl/G1K41K") # alt source

# Init Ggplot
ggplot(midwest, aes(x=area, y=poptotal))  # area and poptotal are columns in 'midwest'
```

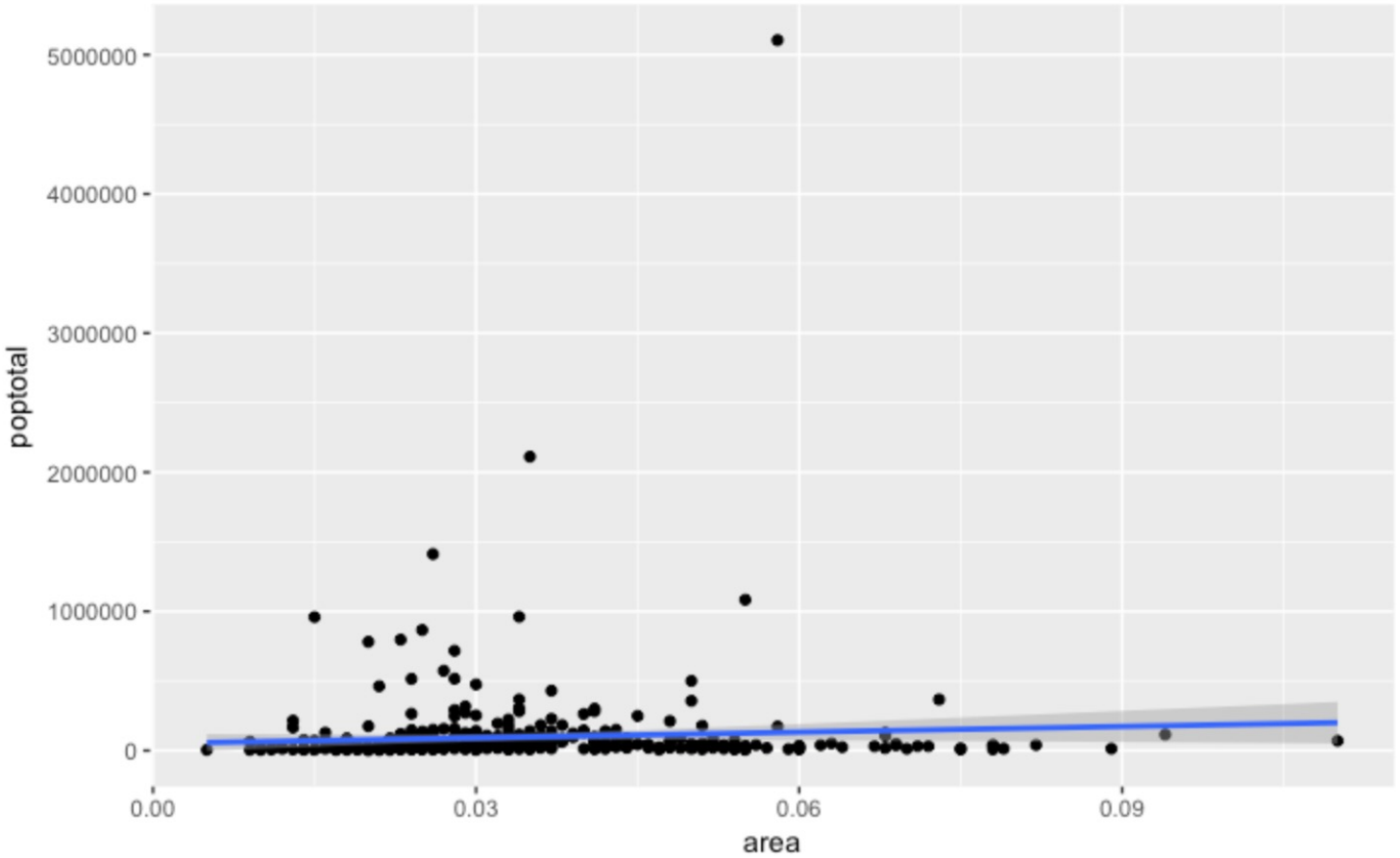


Flinders UNIVERSITY

Scatter plot

using a geom layer called geom_point.

```
library(ggplot2)
ggplot(midwest, aes(x=area, y=poptotal)) + geom_point()
```

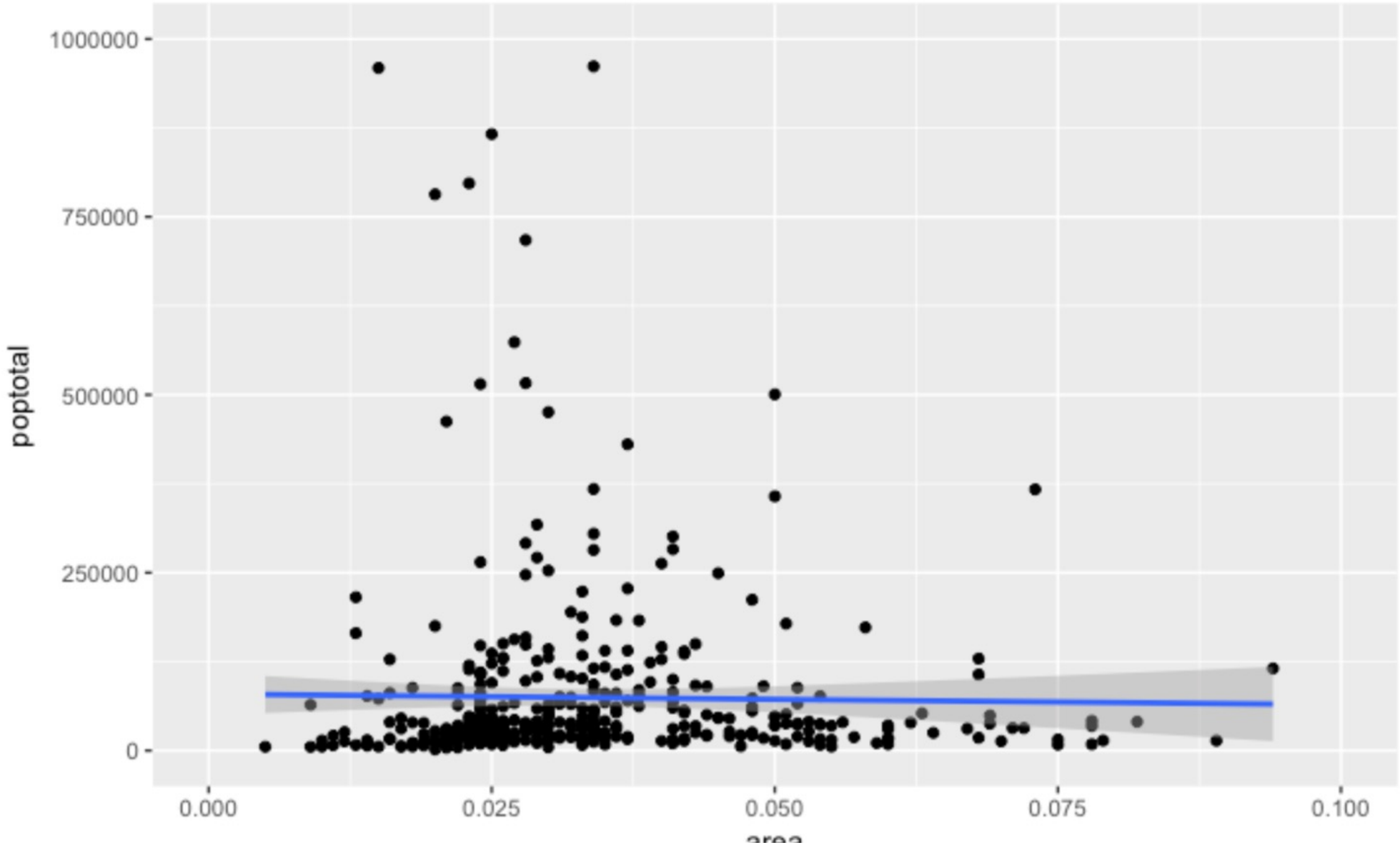



```
library(ggplot2)
g <- ggplot(midwest, aes(x=area, y=poptotal)) + geom_point() + geom_smooth(method="lm")  # set se=FALSE to turnoff confidence bands
plot(g)
```

Flinders UNIVERSITY

Adjusting limits xlim() and ylim()

```
library(ggplot2)
g <- ggplot(midwest, aes(x=area, y=poptotal)) + geom_point() + geom_smooth(method="lm")  #
set se=FALSE to turnoff confidence bands

# Delete the points outside the limits
g + xlim(c(0, 0.1)) + ylim(c(0, 1000000))   # deletes points
# g + xlim(0, 0.1) + ylim(0, 1000000)   # deletes points
```



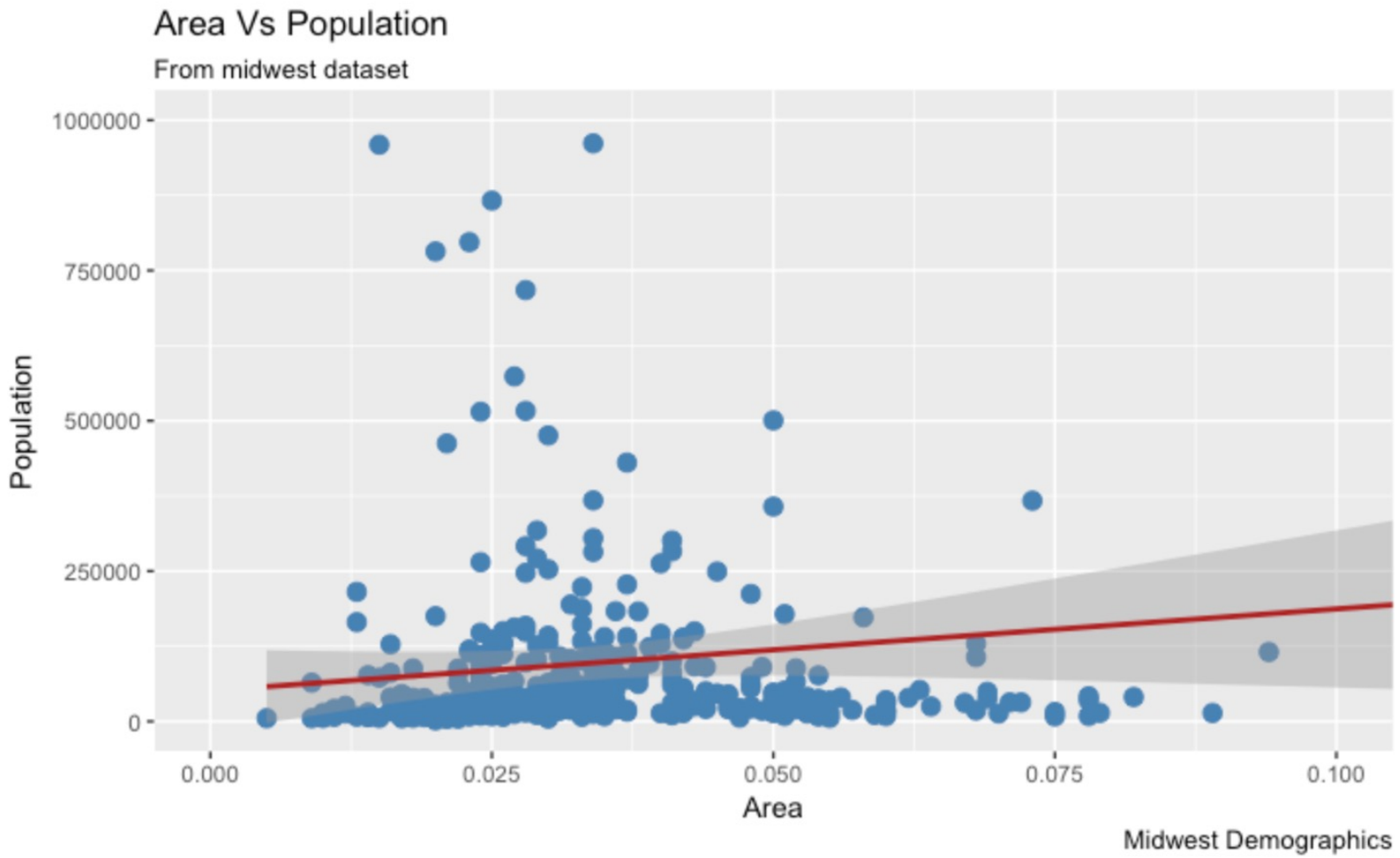
Flinders UNIVERSITY

Title and axis labels

```
library(ggplot2)
g <- ggplot(midwest, aes(x=area, y=poptotal)) + geom_point() + geom_smooth(method="lm")  #
set se=FALSE to turnoff confidence bands

g1 <- g + coord_cartesian(xlim=c(0,0.1), ylim=c(0, 1000000))  # zooms in

# Add Title and Labels
g1 + labs(title="Area Vs Population", subtitle="From midwest dataset", y="Population", x="A
rea", caption="Midwest Demographics")

# or

g1 + ggtitle("Area Vs Population", subtitle="From midwest dataset") + xlab("Area") + ylab
("Population")
```

Color and size of points

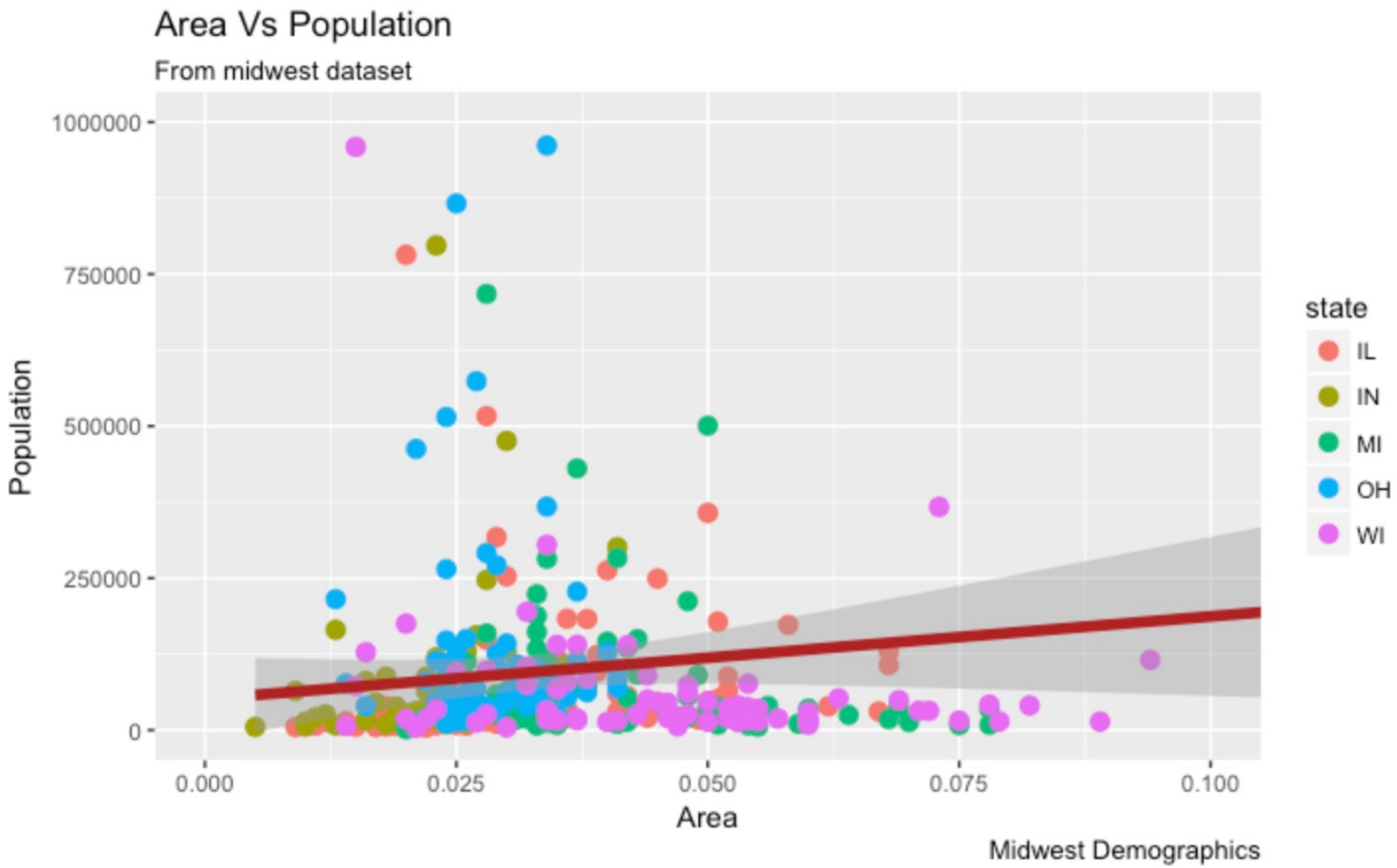
```
library(ggplot2)
ggplot(midwest, aes(x=area, y=poptotal)) +
  geom_point(col="steelblue", size=3) +   # Set static color and size for points
  geom_smooth(method="lm", col="firebrick") +  # change the color of line
  coord_cartesian(xlim=c(0, 0.1), ylim=c(0, 1000000)) +
  labs(title="Area Vs Population", subtitle="From midwest dataset", y="Population", x="Are
a", caption="Midwest Demographics")
```



Flinders UNIVERSITY

Change the Color To Reflect Categories in Another Column

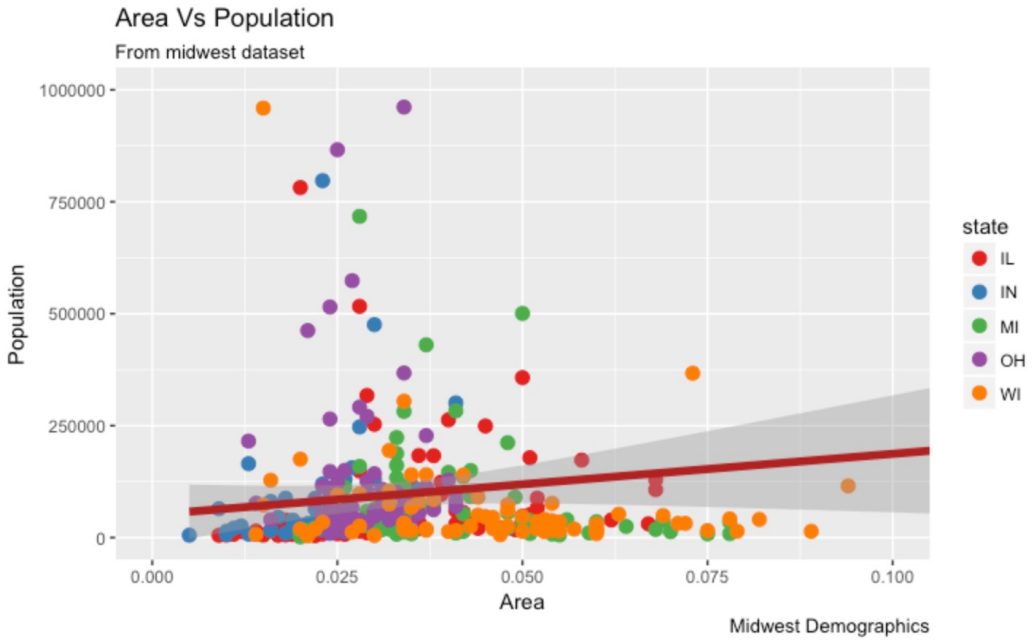
```
library(ggplot2)
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
  geom_point(aes(col=state), size=3) +  # Set color to vary based on state categories.
  geom_smooth(method="lm", col="firebrick", size=2) +
  coord_cartesian(xlim=c(0, 0.1), ylim=c(0, 1000000)) +
  labs(title="Area Vs Population", subtitle="From midwest dataset", y="Population", x="Are
a", caption="Midwest Demographics")
plot(gg)
```



Flinders UNIVERSITY

Remove legend

- gg + **theme**(legend.position="None") *# remove legend*
- *You can also change the colour palette*
  - gg + **scale_colour_brewer**(palette = "Set1") *# change colour palette*



Colour Palettes

```
library(RColorBrewer)
head(brewer.pal.info, 10)  # show 10 palettes
#>          maxcolors category colorblind
#> BrBG            11      div       TRUE
#> PiYG            11      div       TRUE
#> PRGn            11      div       TRUE
#> PuOr            11      div       TRUE
#> RdBu            11      div       TRUE
#> RdGy            11      div      FALSE
#> RdYlBu          11      div       TRUE
#> RdYlGn          11      div      FALSE
#> Spectral        11      div      FALSE
#> Accent           8     qual      FALSE
```

Flinders UNIVERSITY

Colour Palettes

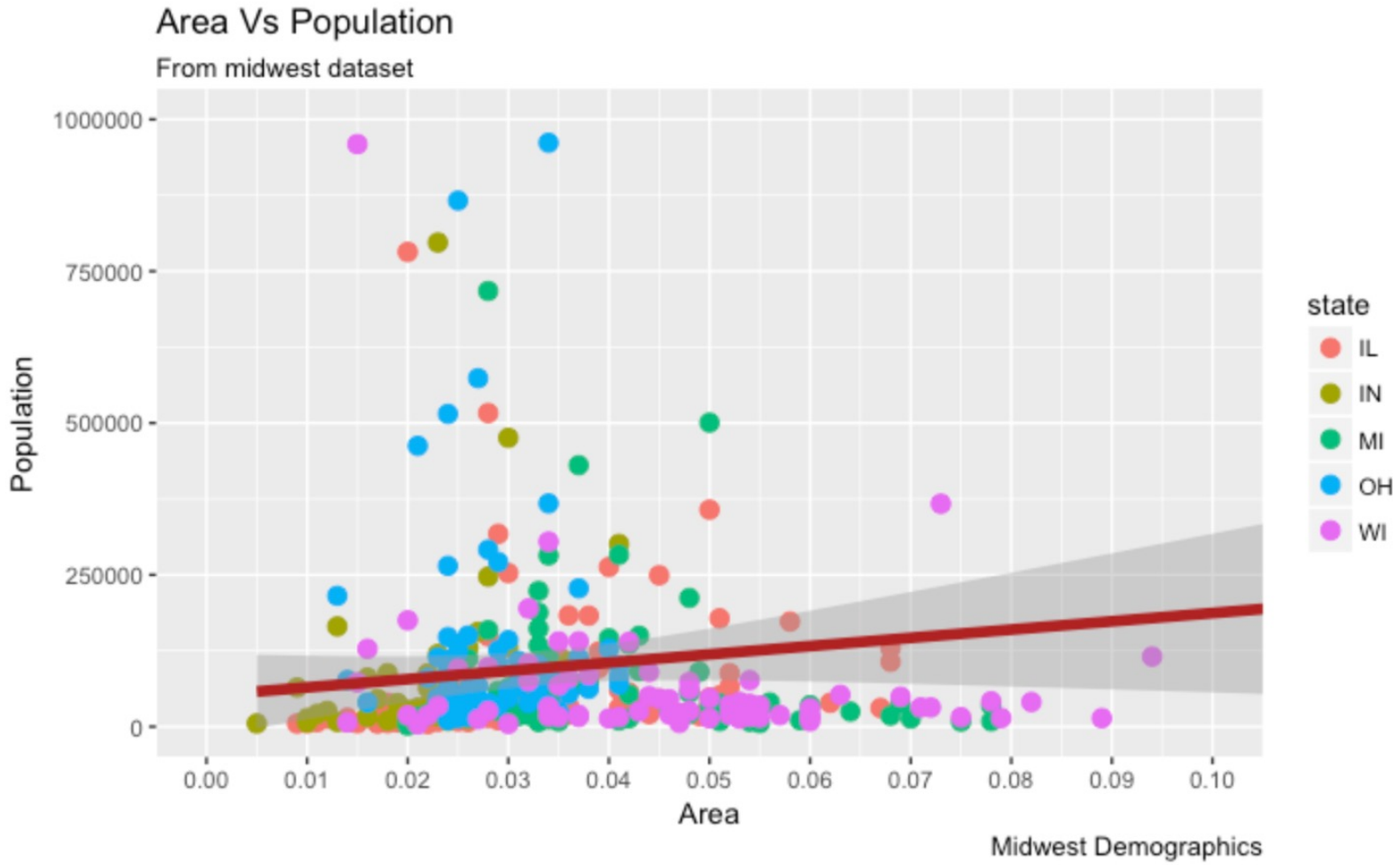
Flinders UNIVERSITY

How to Change the X and Y Axis Text and its Location?

- **Step 1: Set the breaks**
  The breaks should be of the same scale as the X axis variable.
- Using scale_x_continuous because, the X axis variable is a continuous variable. Had it been a date variable, scale_x_date could be used. Like scale_x_continuous() an equivalent scale_y_continuous() is available for Y axis.

- **Step 2: Change the labels** You can optionally change the labels at the axis ticks. labels take a vector of the same length as breaks.

```r
library(ggplot2)

# Base plot
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
  geom_point(aes(col=state), size=3) +  # Set color to vary based on state categories.
  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="Are
a", caption="Midwest Demographics")

# Change breaks
gg + scale_x_continuous(breaks=seq(0, 0.1, 0.01))
```



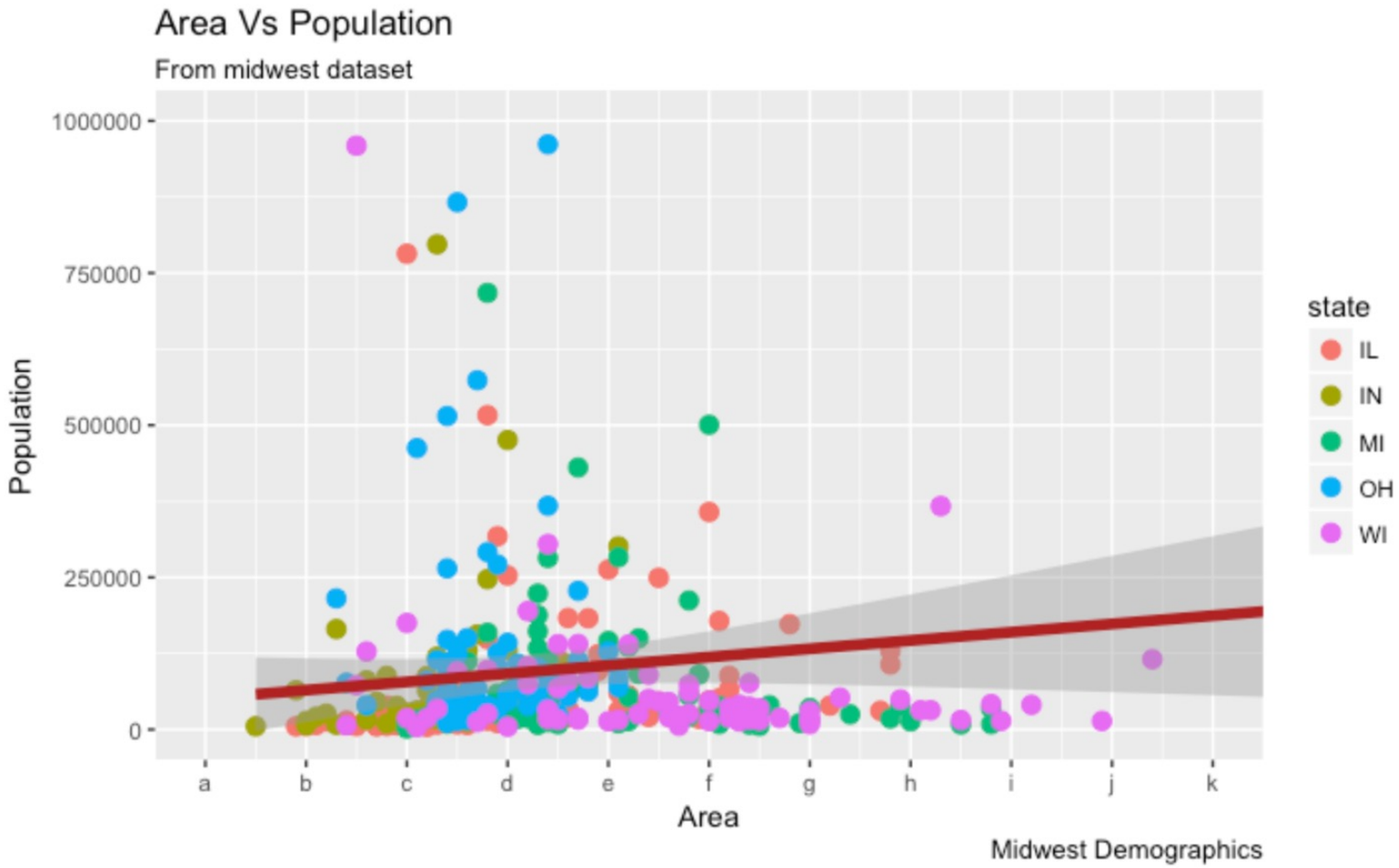
Flinders UNIVERSITY

How to Change the X and Y Axis Text and its Location?

- **Step 1: Set the breaks**
  The breaks should be of the same scale as the X axis variable.
- Using scale_x_continuous because, the X axis variable is a continuous variable. Had it been a date variable, scale_x_date could be used. Like scale_x_continuous() an equivalent scale_y_continuous() is available for Y axis.

- **Step 2: Change the labels** You can optionally change the labels at the axis ticks. labels take a vector of the same length as breaks.

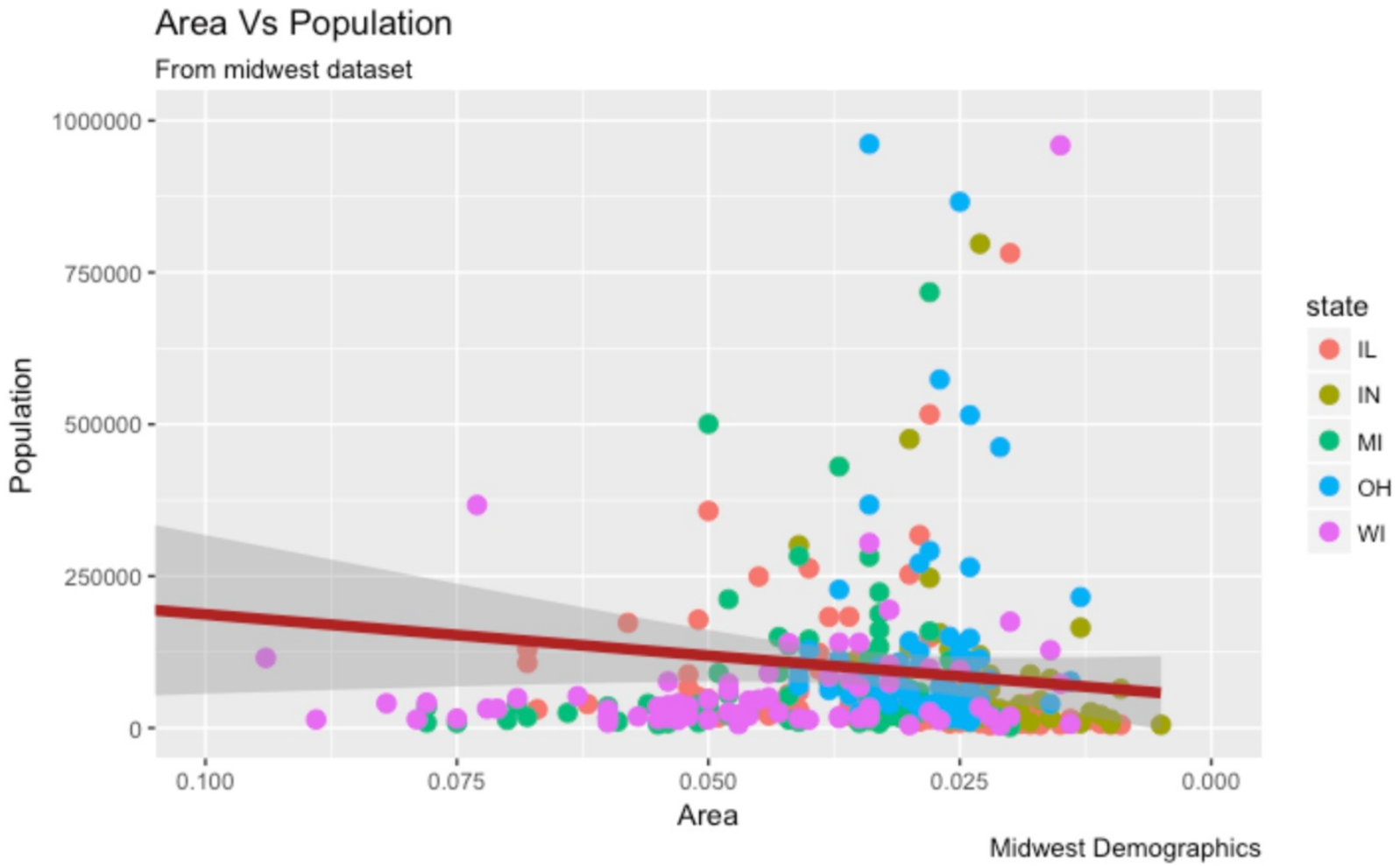
```r
library(ggplots)

# Base Plot
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
  geom_point(aes(col=state), size=3) +  # Set color to vary based on state categories.
  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="Are
a", caption="Midwest Demographics")

# Change breaks + label
gg + scale_x_continuous(breaks=seq(0, 0.1, 0.01), labels = letters[1:11])
```



Flinders UNIVERSITY

Reverse scale

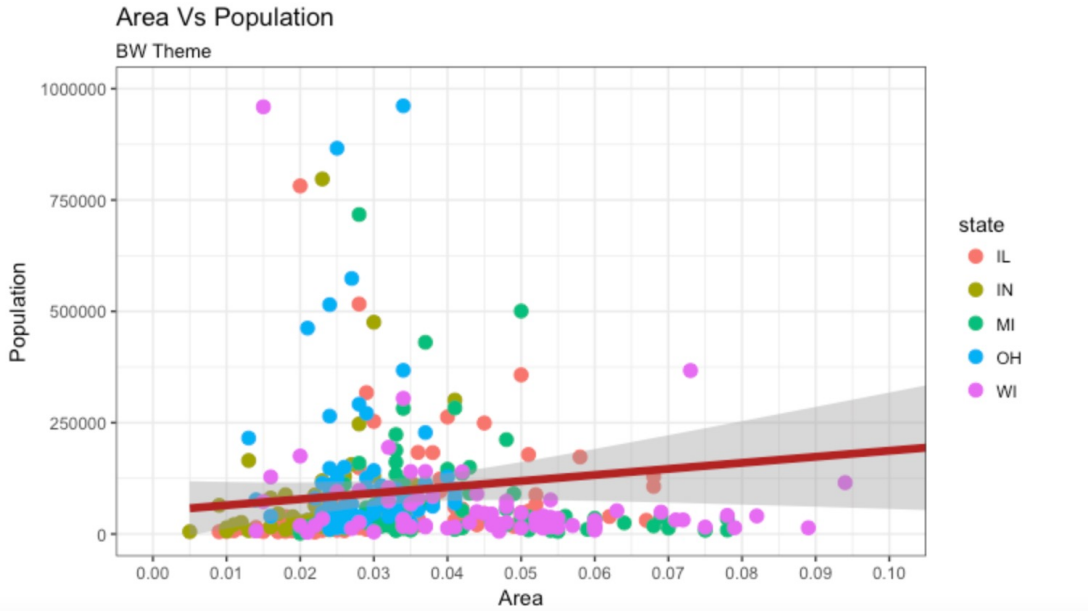
- use scale_x_reverse().

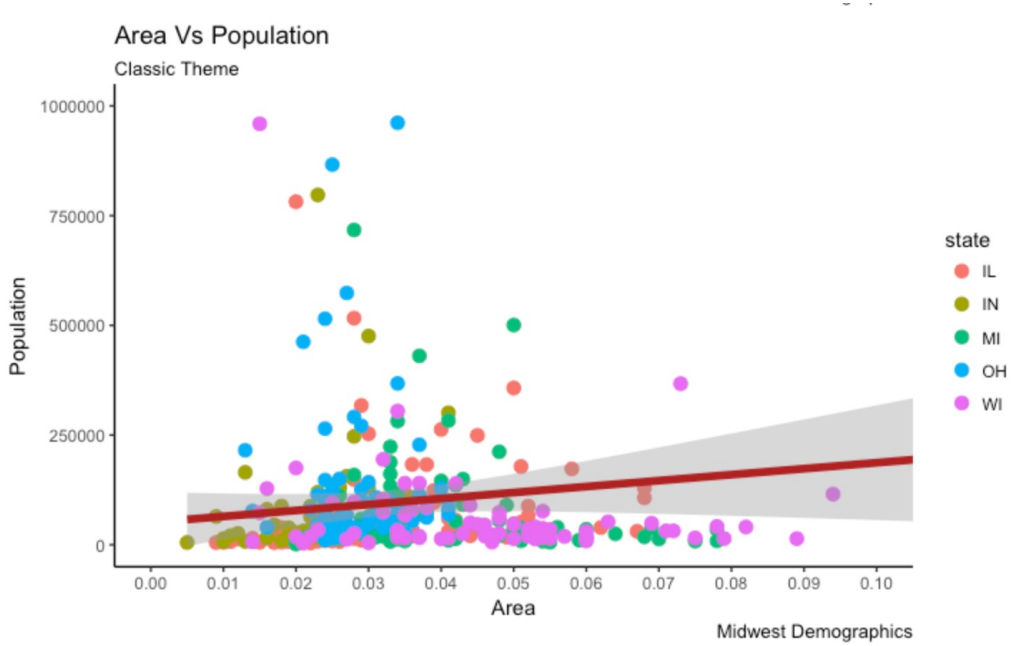


Flinders UNIVERSITY

Customize the Entire Theme

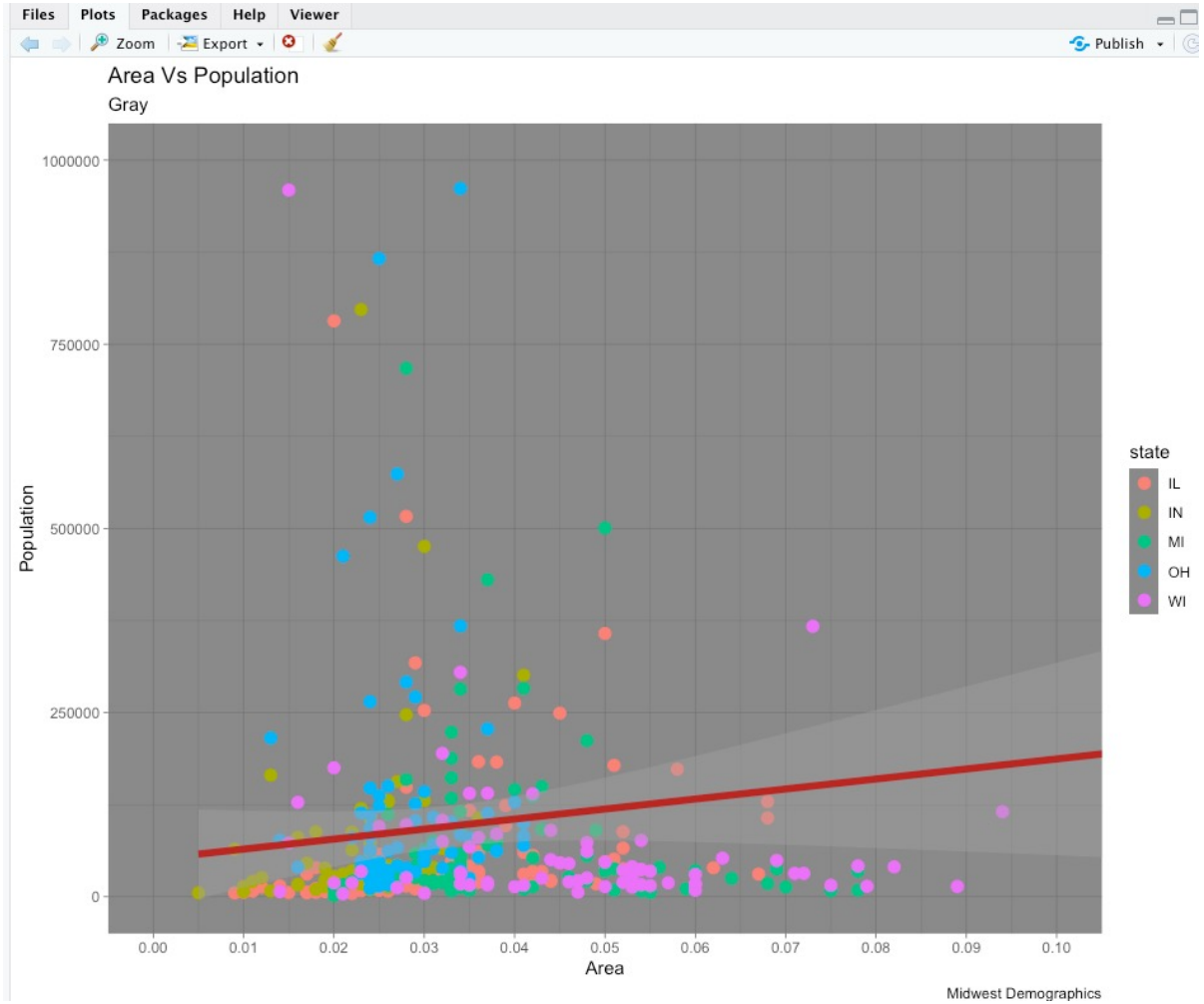
- Use the theme_set() to set the theme before drawing the ggplot. Note that this setting will affect all future plots.
- Draw the ggplot and then add the overall theme setting (eg. theme_bw())





Flinders UNIVERSITY

Dark theme



Flinders UNIVERSITY

Further reading

- http://r-statistics.co/Complete-Ggplot2-Tutorial-Part2-Customizing-Theme-With-R-Code.html

- http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html