DATA VISUALIZATION

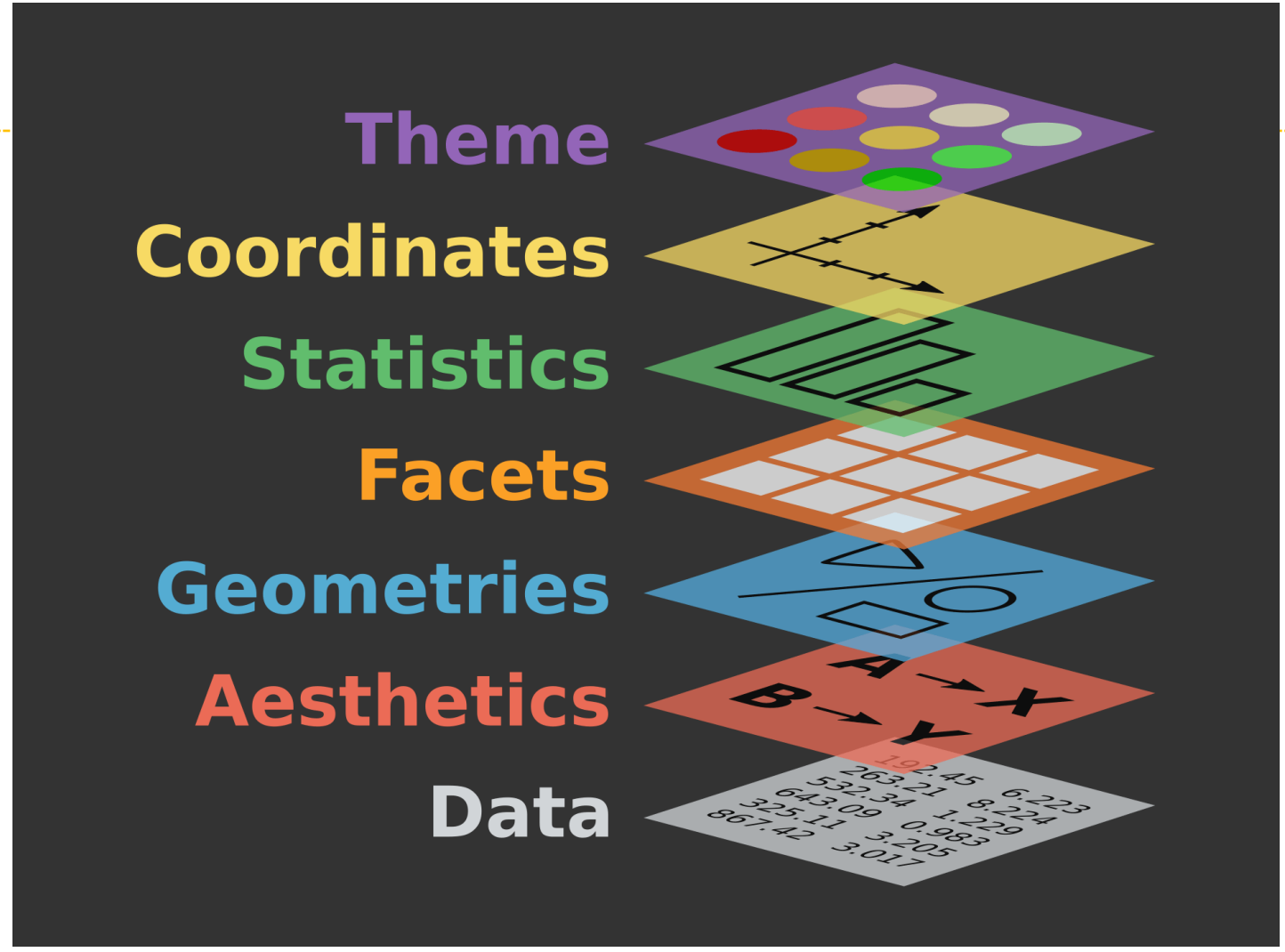
Part 1: Grammar of ggplot2

Khuong Quynh Long
Ha Noi, 03/2019
https://gitlab.com/LongKhuong/data-visualization

Vai trò của data visualization

ggplot2

- ggplot2 là package đồ họa của R
- Viết bởi Hadley Wickham (bắt đầu năm 2005)
- Dựa vào nguyên lý "*Grammar of Graphics*" của Leland Wilkinson
- Hệ thống graphics thứ 3 (cùng với **base** và **lattice**)
- Nằm trong hệ thống "Tidyverse" (có thể cài đặt trực tiếp từ **install.packages("ggplot2")**, hoặc **install.packages("tidyverse")**)
- Website: https://ggplot2.tidyverse.org


Theme
Coordinates
Statistics
Facets
Geometries
Aesthetics
Data
A → X
B → Y
192.45 6.223
263.21 8.224
532.34 1.229
643.09 0.983
325.11 3.205
867.42 3.017

Grammar of ggplot2

- Cấu trúc ggplot2 bao gồm tất cả các thành phần của biểu đồ (tương tự “subject”, “verb”, “noun”, “adj”... Cho đồ họa).
- Hoạt động theo lớp (layers) (tương tự cơ chế của photoshop)

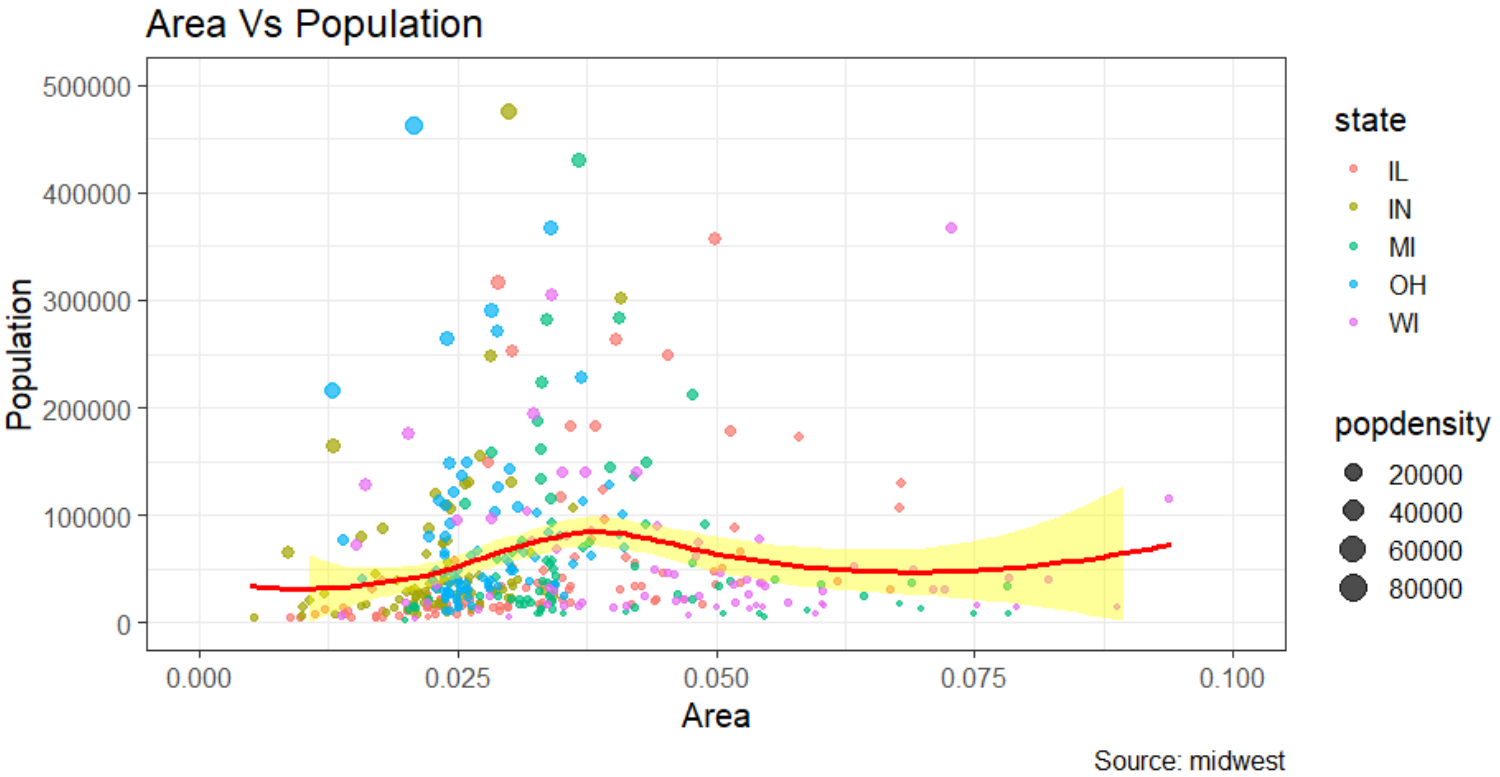
➔ linh hoạt, có thể kết hợp nhiều loại biểu đồ (trên cùng một hay nhiều bộ data) cùng lúc

Grammar of ggplot2

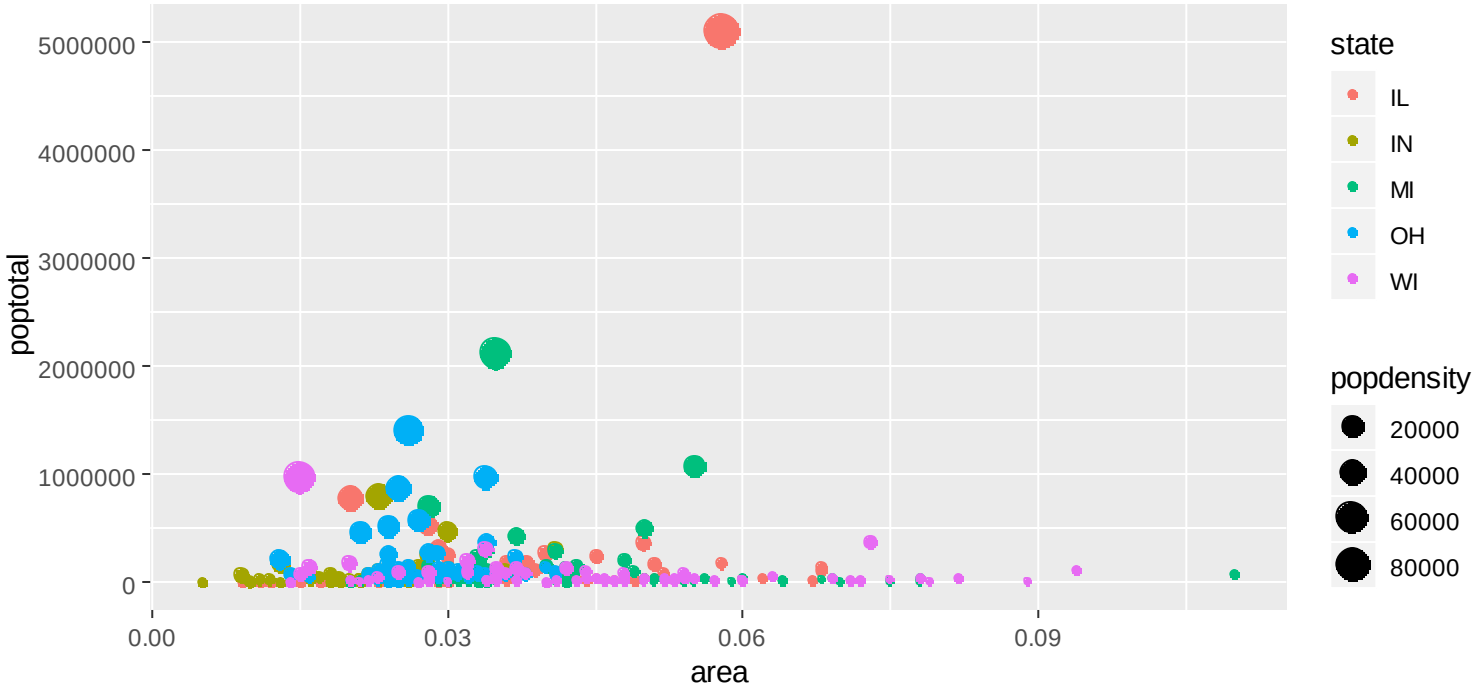
- “In brief, the grammar tells us that a statistical graphic is a mapping from data to aesthetic attributes (colour, shape, size) of geometric objects (points, lines, bars). The plot may also contain statistical transformations of the data and is drawn on a specific coordinate system”

-- Ggplot2 - Elegant Graphics for Data Analysis --

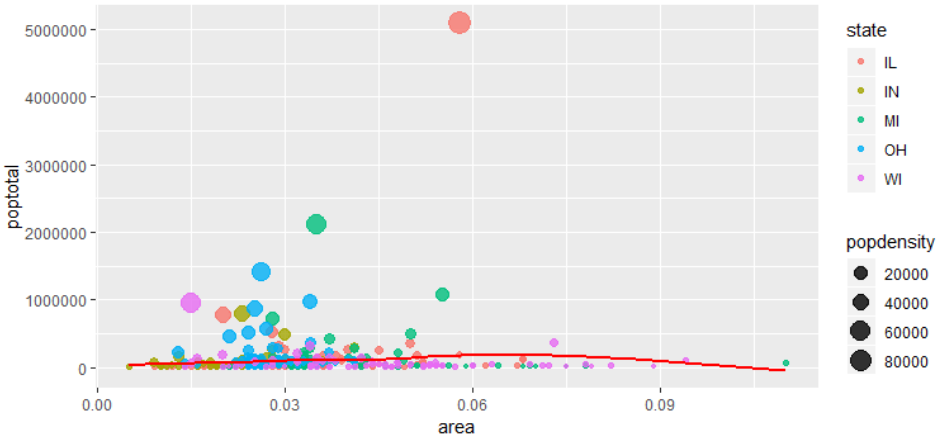
```
ggplot(midwest, aes(x = area, y = poptotal)) +
geom_point(aes(col = state, size = popdensity), alpha = 0.7) +
geom_smooth(method = "loess", size = 1.2, fill = "yellow", col = "red") +
xlim(c(0, 0.1)) + ylim(c(0, 500000)) +
labs(title = "Area Vs Population", y = "Population", x = "Area", caption = "Source: midwest") +
theme_bw(15)
```



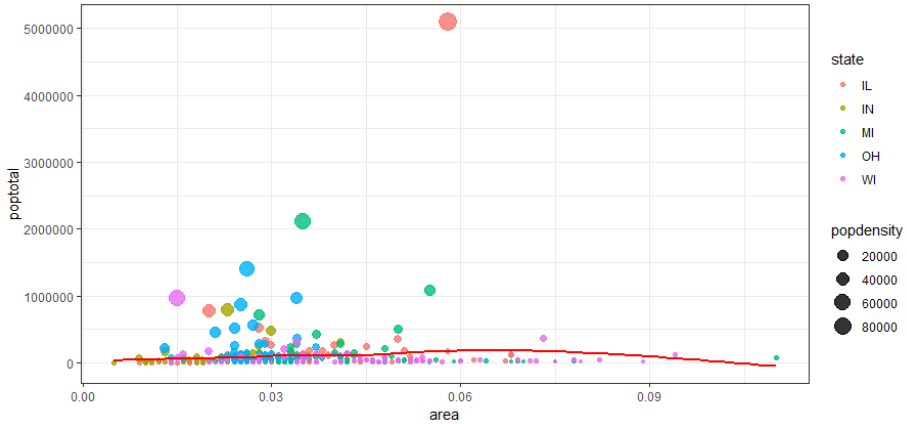
```
gg <- ggplot(data = midwest, aes(x = area, y = poptotal)) +
geom_point(aes(col=state, size=popdensity))
gg
```



```
gg + geom_smooth(method="loess", se=F)
```




```
gg + geom_smooth(method="loess", se=F) +
theme_bw()
```



Components

- Data: bao gồm dataset và aesthetic mappings (bao gồm các trục x, y, color, shape, size…).
- Geometric objects: loại biểu đồ muốn vẽ (point, bar, lines…)
- Statistical transformations, stats (hàm thống kê)
- Scale
- Coordinate system
- Faceting
- Annotation

Ví dụ với scatter plot

Mục tiêu ví dụ

1. Hiểu câu lệnh làm việc với ggplot2
2. Tạo biểu đồ scatter plot đơn giản
3. Thay đổi giới hạn trục x, y
4. Thay đổi title và axis labels

Data – “midwest”

- Data khảo sát dân số miền trung tây nước Mỹ
- Bao gồm 28 vars và 437 obs

```
# library
library(tidyverse)
#---------------------------------
# load data
data("midwest")
options(scipen=999)
```

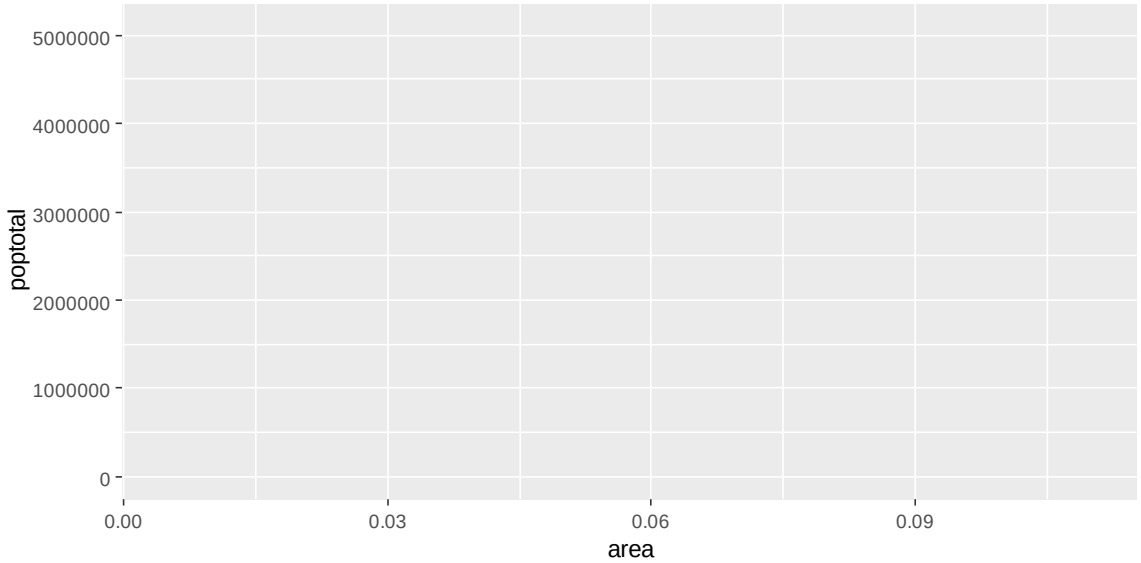
Data – “midwest”

```
> glimpse(midwest)
Observations: 437
Variables: 28
$ PID                  <int> 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, ...
$ county               <chr> "ADAMS", "ALEXANDER", "BOND", "BOONE", "BROWN", "B...
$ state                <chr> "IL", "IL", "IL", "IL", "IL", "IL", "IL", "IL", "I...
$ area                 <dbl> 0.052, 0.014, 0.022, 0.017, 0.018, 0.050, 0.017, 0...
$ poptotal             <int> 66090, 10626, 14991, 30806, 5836, 35688, 5322, 168...
$ popdensity           <dbl> 1270.9615, 759.0000, 681.4091, 1812.1176, 324.2222...
$ popwhite             <int> 63917, 7054, 14477, 29344, 5264, 35157, 5298, 1651...
$ popblack             <int> 1702, 3496, 429, 127, 547, 50, 1, 111, 16, 16559, ...
$ popamerindian        <int> 98, 19, 35, 46, 14, 65, 8, 30, 8, 331, 51, 26, 17,...
$ popasian             <int> 249, 48, 16, 150, 5, 195, 15, 61, 23, 8033, 89, 36...
$ popother             <int> 124, 9, 34, 1139, 6, 221, 0, 84, 6, 1596, 20, 7, 7...
$ percwhite            <dbl> 96.71206, 66.38434, 96.57128, 95.25417, 90.19877, ...
$ percblack            <dbl> 2.57527614, 32.90043290, 2.86171703, 0.41225735, 9...
$ percamerindan        <dbl> 0.14828264, 0.17880670, 0.23347342, 0.14932156, 0....
$ percasian            <dbl> 0.37675897, 0.45172219, 0.10673071, 0.48691813, 0....
$ percother            <dbl> 0.18762294, 0.08469791, 0.22680275, 3.69733169, 0....
$ popadults            <int> 43298, 6724, 9669, 19272, 3979, 23444, 3583, 11323...
$ perchsd              <dbl> 75.10740, 59.72635, 69.33499, 75.47219, 68.86152, ...
$ percollege           <dbl> 19.63139, 11.24331, 17.03382, 17.27895, 14.47600, ...
$ percprof             <dbl> 4.355859, 2.870315, 4.488572, 4.197800, 3.367680, ...
$ poppovertyknown      <int> 63628, 10529, 14235, 30337, 4815, 35107, 5241, 164...
$ percpovertyknown     <dbl> 96.27478, 99.08714, 94.95697, 98.47757, 82.50514, ...
$ percbelowpoverty     <dbl> 13.151443, 32.244278, 12.068844, 7.209019, 13.5202...
$ percchildbelowpovert <dbl> 18.011717, 45.826514, 14.036061, 11.179536, 13.022...
$ percadultpoverty     <dbl> 11.009776, 27.385647, 10.852090, 5.536013, 11.1432...
$ percelderlypoverty   <dbl> 12.443812, 25.228976, 12.697410, 6.217047, 19.2000...
$ inmetro              <int> 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0,...
$ category             <chr> "AAR", "LHR", "AAR", "ALU", "AAR", "AAR", "LAR", "...
> 
```

1. Data & aesthetic mappings

- ggplot2 chỉ nhận cấu trúc dữ liệu “data frame”, không nhận vector hay matrix
- Trong thực hành, có thể cần nhiều bước chuẩn bị để từ data gốc ➔ data input cho ggplot2. Một số package hữu ích trong biên tập số liệu như “dplyr”, “tidyverse”, toán tử pipe %>%
- aesthetic mappings bao gồm trục x, y, z muốn vẽ (tùy loại biểu đồ 1D, 2D, 3D…) shape, size, color
- Data và aesthetic mappings có thể khai báo chung cho tất cả các layer hoặc từng layer

```
ggplot(data = midwest, aes(x = area, y = poptotal))
```



- Biểu đồ trống vì chỉ mới khai báo Data (midwest) và aesthetic mappings (trục x là area, trục y là total population) ➔ chưa biết vẽ loại biểu đồ gì
- Cần thêm thành phần “Geometric”

2. Geometric objects

- Loại biểu đồ muốn vẽ
- Sử dụng thông tin trong aesthetic mappings (ví dụ scatter phải cần 2 trục (x,y), bar chỉ cần trục x …)
- Câu lệnh:

✓ geom_point: điểm

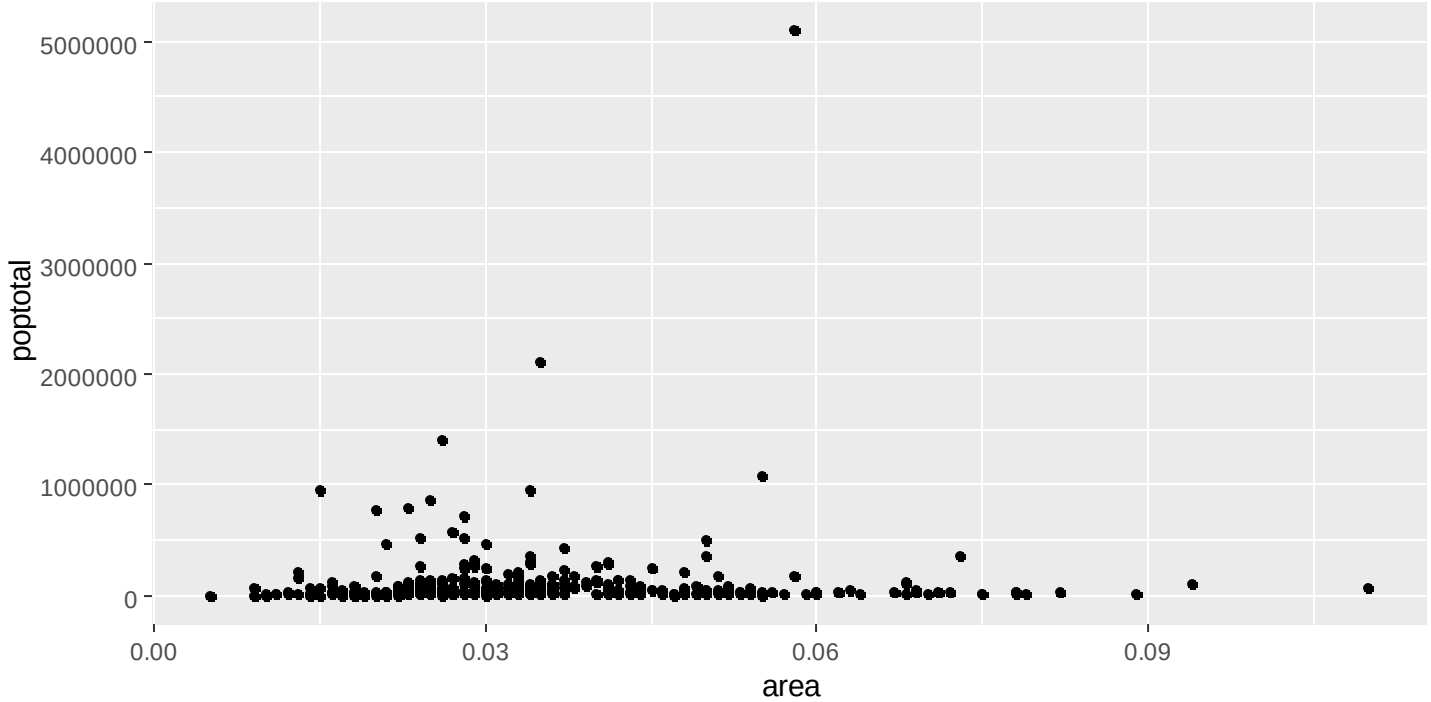
✓ geom_line: đường

✓ geom_histogram: histogram

✓ .... : tham khảo cheat sheet

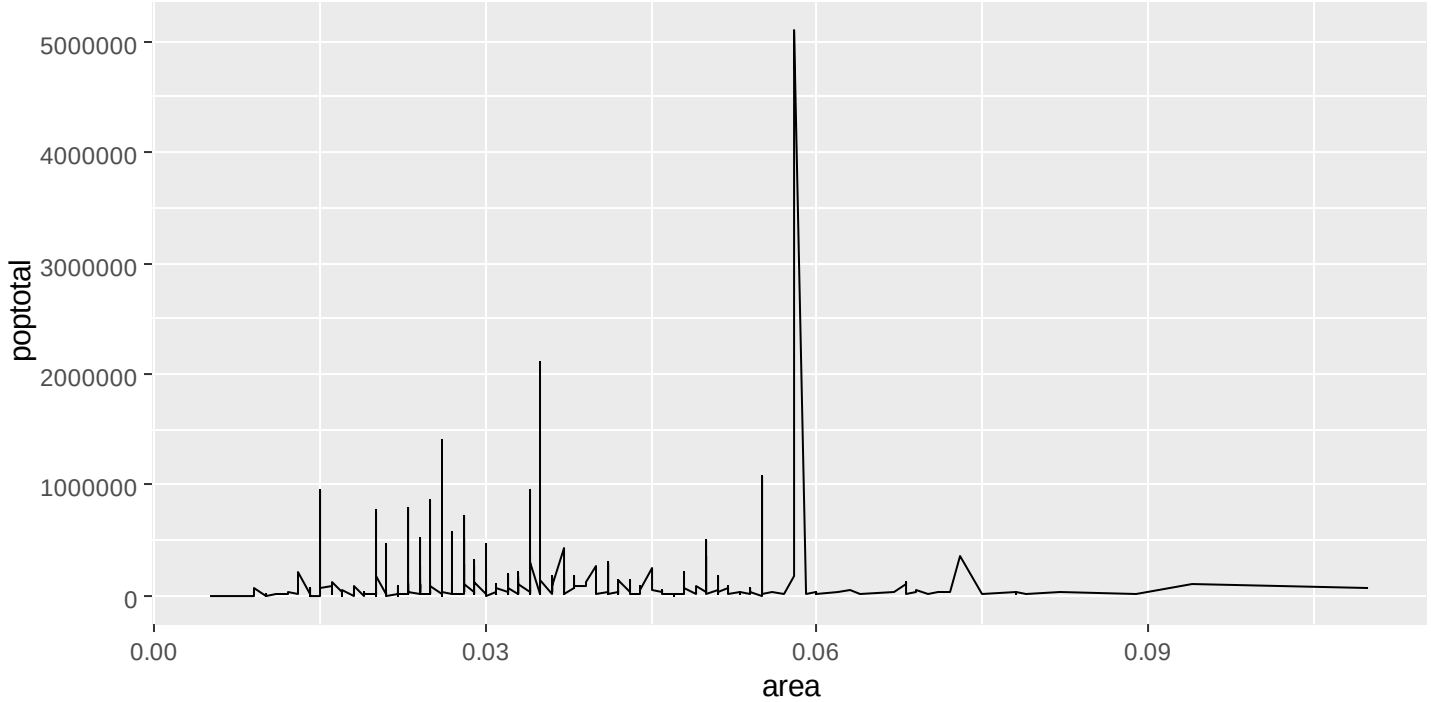
- Data + aesthetic + geometric object là 3 thành phần tối thiểu của một biểu đồ

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



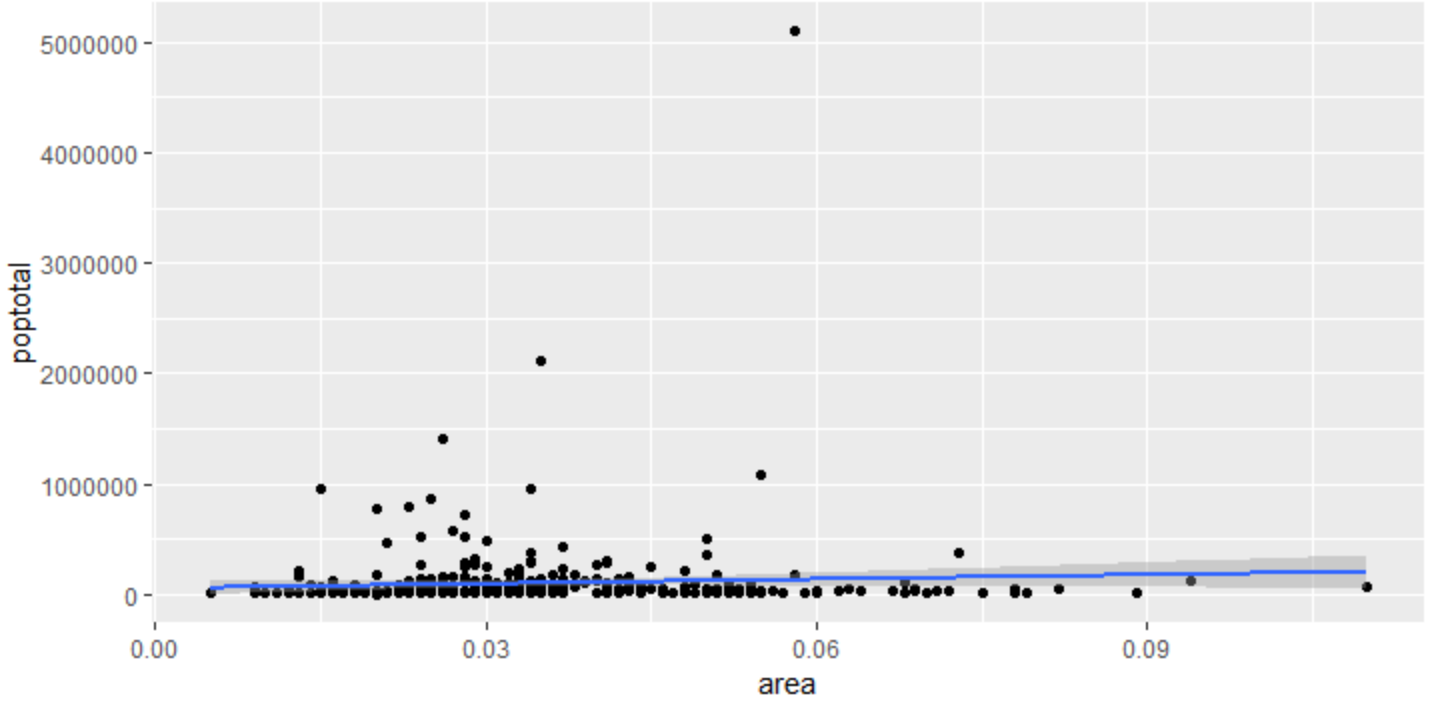
- Thêm geom_point() để vẽ dạng biểu đồ điểm
- geom_point sử dụng thông tin trục x, y từ aes()

```
ggplot(data = midwest, aes(x = area, y = poptotal)) +
geom_line()
```



- Hoặc geom_line() để vẽ dạng biểu đồ đường

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



- Từ biểu đồ điểm, thêm 1 layer smooth (linear model)

Chú ý

- Data và aes() được khai báo ở "**ggplot()**" ➔ toàn bộ các layer sau (geom_point(), geom_smooth()) đều sử dụng thông tin khai báo này
- Nếu data và aes() được khai báo riêng lẻ ở từng layer ➔ thông tin này chỉ được áp dụng cho riêng từng layer đó (trường hợp này được sử dụng khi muốn vẽ nhiều layer từ nhiều data khác nhau)
- Ví dụ

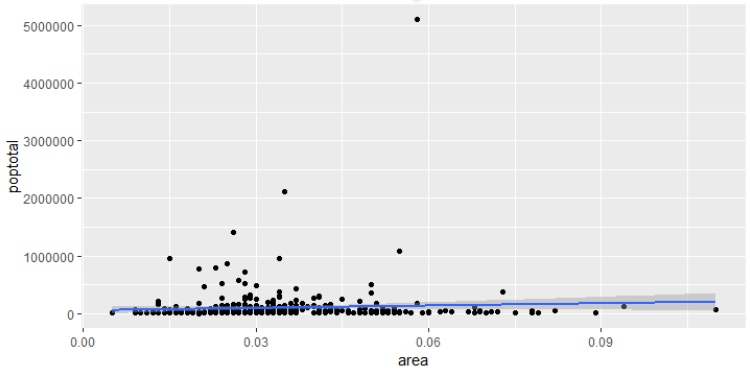
```
ggplot(data = midwest, aes(x = area, y = poptotal)) +
geom_point()
```

Data và aes() được dùng chung cho các layers

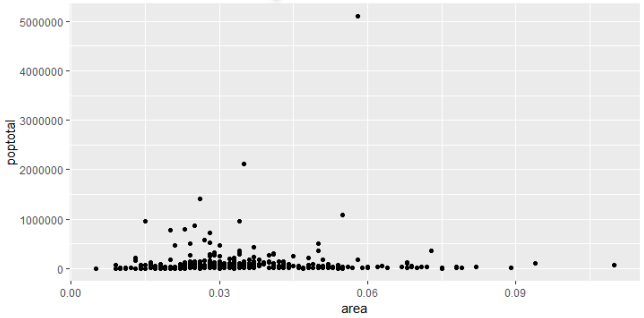
```
ggplot() +
geom_point(data = midwest, aes(x = area, y = poptotal))
```

Data và aes() chỉ áp dụng cho layer geom_point()

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




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



layer geom_smooth() chưa có data & aes()

3. Thay đổi giới hạn trục x, y

- Có 2 cách giới hạn trục x và y

Cách 1:

✓ Giới hạn trục x và y đồng thời loại bỏ các giá trị nằm ngoài khoảng giới hạn

✓ Sử dụng câu lệnh **`xlim()`** & **`ylim()`**

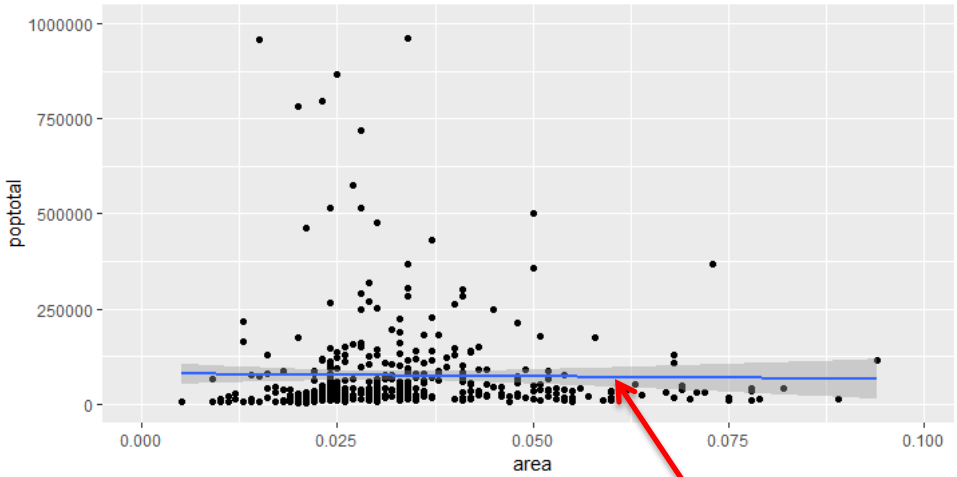
Cách 2:

✓ "zoom in" vào khoảng giới hạn ➔ không loại bỏ các giá trị nằm ngoài giới hạn

✓ Sử dụng câu lệnh **`coord_cartesian()`**

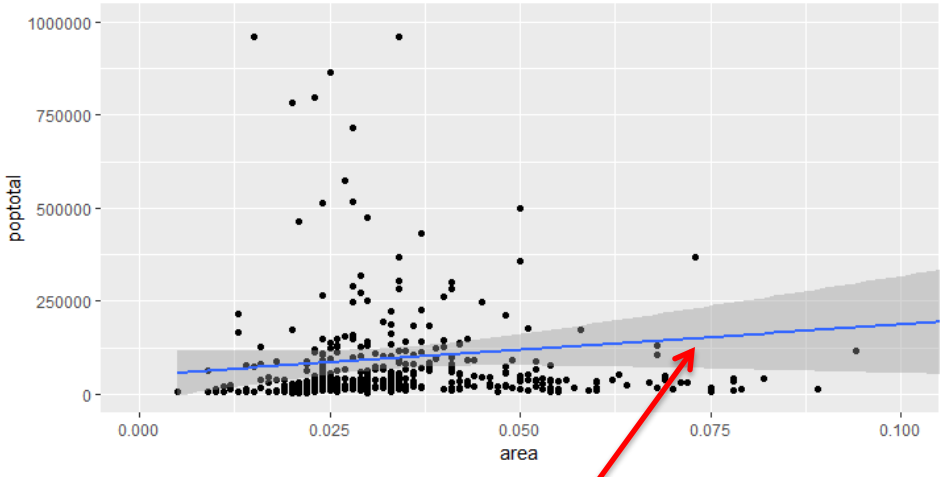
```r
p <- ggplot(data = midwest, aes(x = area, y = poptotal)) + geom_point() +
geom_smooth(method = "lm")
```

```r
p1 <- p + xlim(c(0, 0.1)) + ylim(c(0,
1000000))
p1
```



Thay đổi xu hướng do loại bỏ các giá trị ngoài phạm vi 0 - 1000000

```r
p2 <- p + coord_cartesian(xlim=c(0,0.1),
ylim=c(0, 1000000))
p2
```



Giữ nguyên xu hướng do chỉ "zoom in" vào phạm vi 0 - 1000000

4. Thay đổi title và axis labels

- Có nhiều cách

✓ Cách 1: dùng tất cả trong 1 câu lệnh

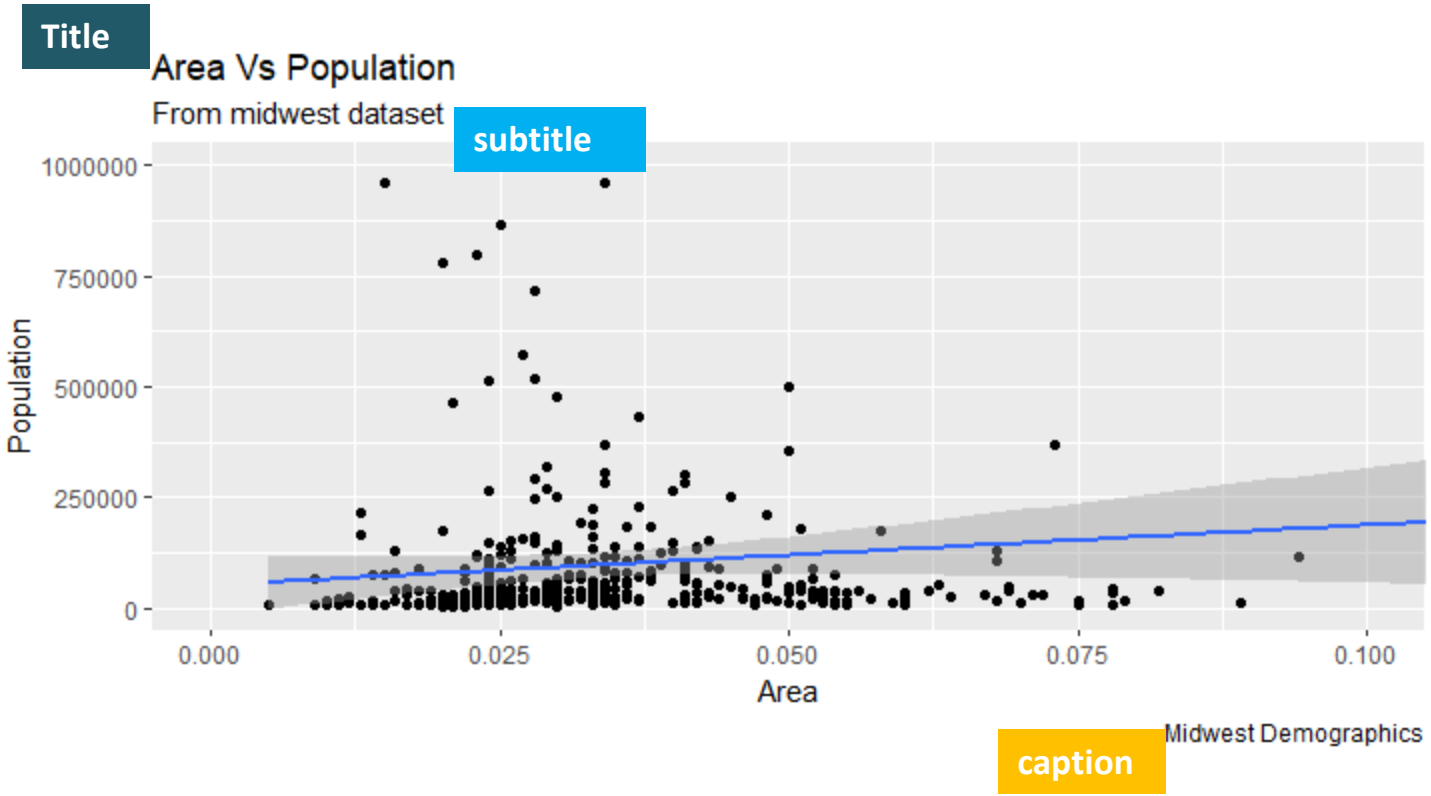
labs(title = “”, subtitle= “...”, y= “…”, x= “…”, caption= “…”)

✓ Cách 2: từng thành phần riêng lẻ

ggtitle(“…”, subtitle= “...”) + **xlab**(“…”) + **ylab**(“…”)

✓ Cách 3 : kết hợp một số “scale” (ít dùng hơn)

```
p2 + labs(title="Area Vs Population",
          subtitle="From midwest dataset",
          y="Population", x="Area",
          caption="Midwest Demographics")
```



full syntax

- Syntax cơ bản cho scatter plot

```
ggplot(data = 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="Midwest Demographics")
```

Tùy chỉnh color, shape, size, theme

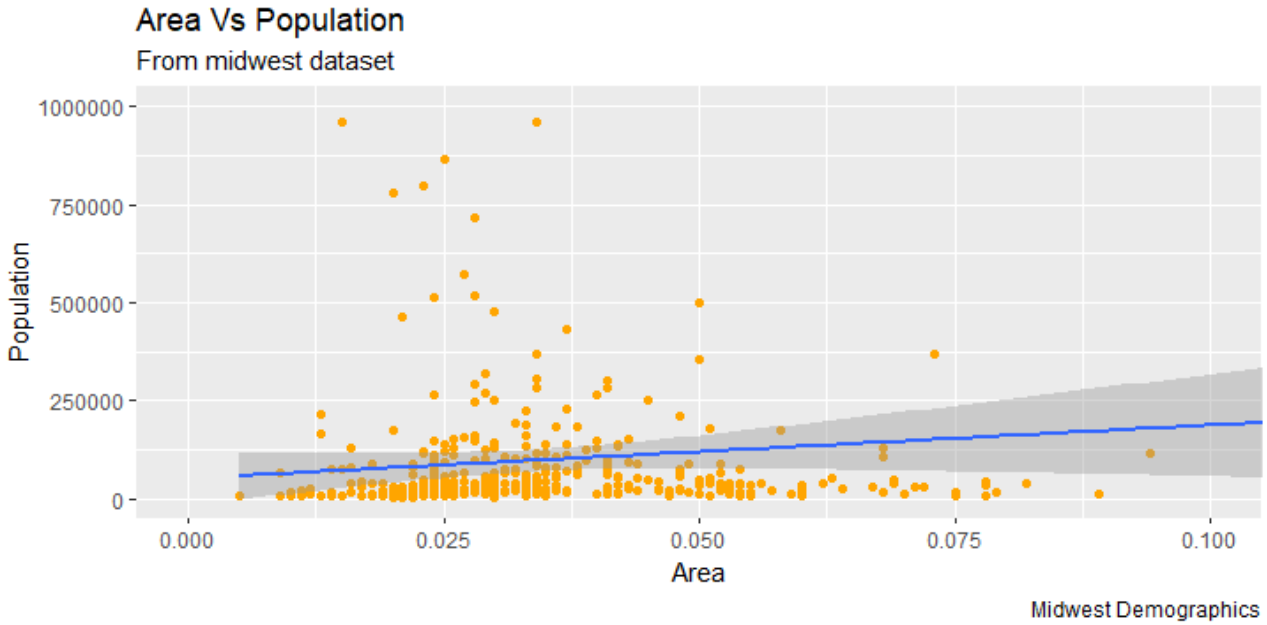
Nội dung

- Tùy chỉnh color, shape, size
- Theme

Color

Chỉ áp dụng cho layer này, tất cả các point thành màu "orange"

```
ggplot(data = midwest, aes(x = area, y = poptotal)) +
    geom_point(col = "orange") +
    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="Midwest Demographics")
```

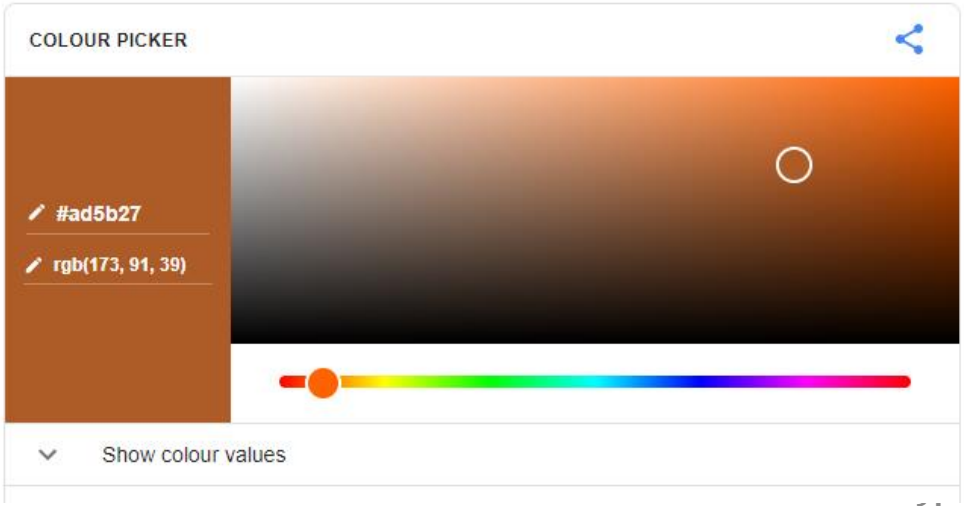


Color

- Có thể gọi tên trực tiếp: “red”, “blue”…

www.stat.columbia.edu/~tzheng/files/Rcolor.pdf

- Hệ thống màu Hexadecimal code (search từ khóa “hex color” bằng google)
- Hệ thống RGB

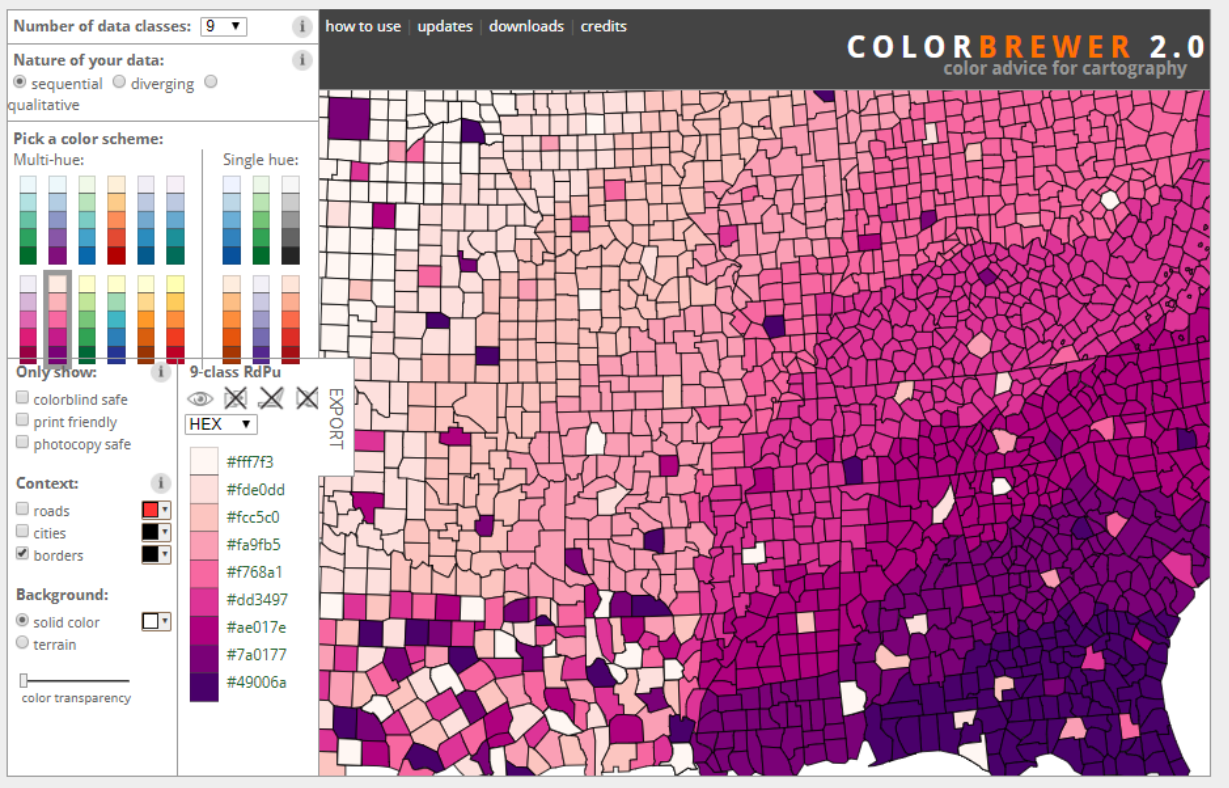


Color

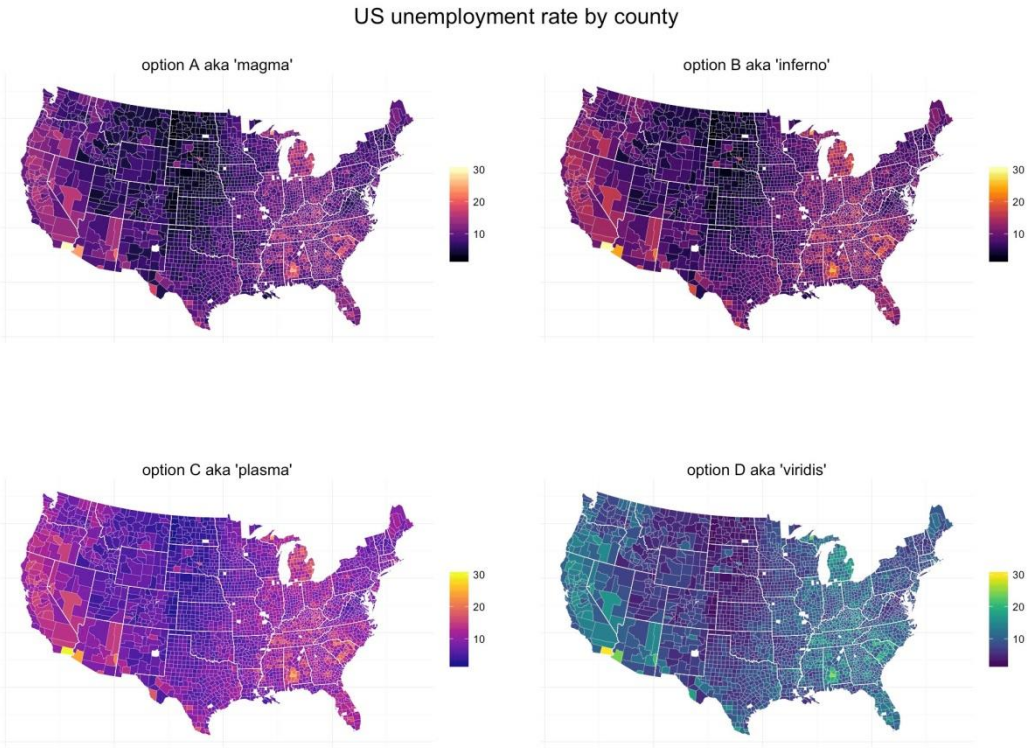
- Một số package cung cấp phổ màu riêng
- “viridis” cung cấp 4 phổ màu liên tục
- “RColorBrewer” cung cấp các phổ màu liên tục và rời rạc

Color

- http://colorbrewer2.org/#type=sequential&scheme=RdPu&n=9

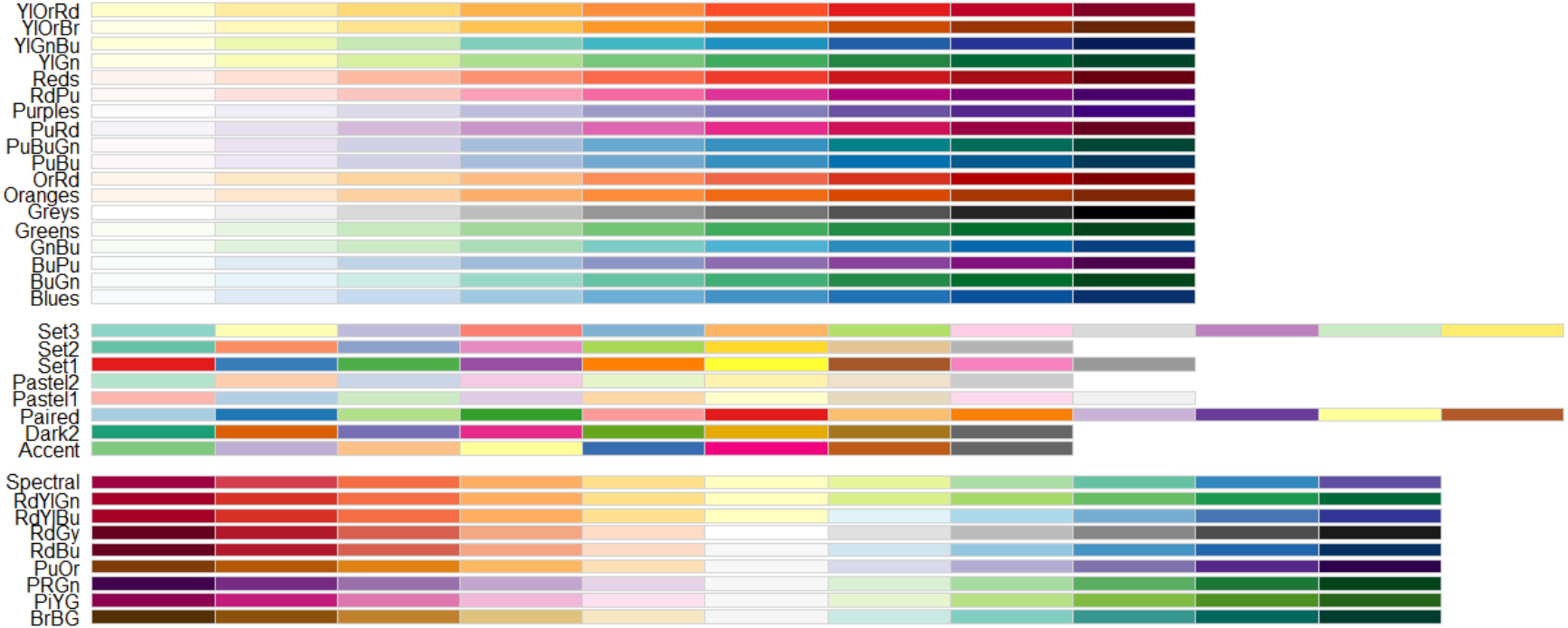


Package viridis



Package RColorBrewer

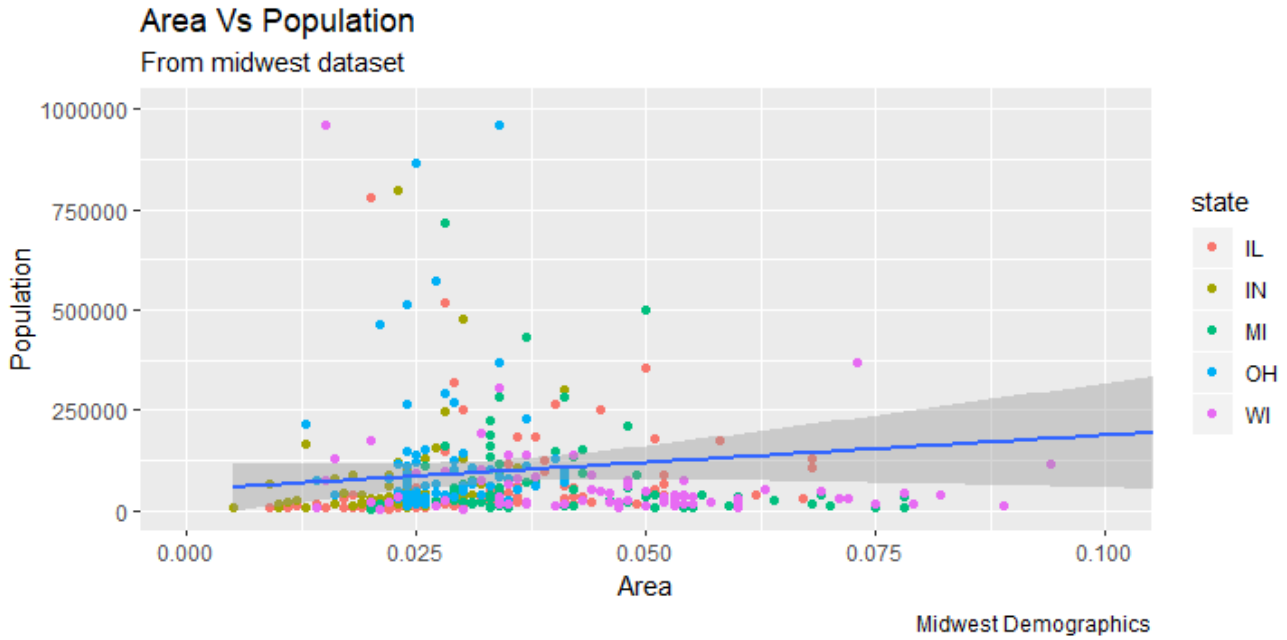
```
library(RColorBrewer)
display.brewer.all()
```



Color

Màu khác nhau cho từng "state"

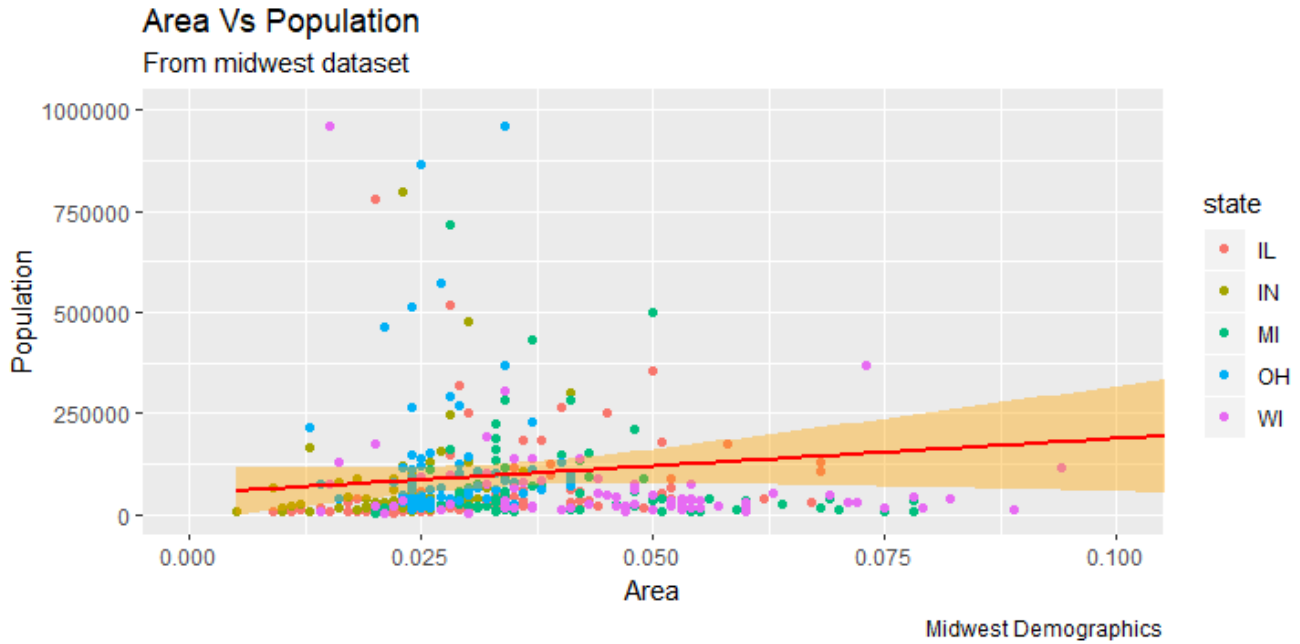
```
ggplot(data = midwest, aes(x = area, y = poptotal)) +
    geom_point(aes(col = state)) +
    geom_smooth(method = "lm") +
    coord_cartesian(xlim=c(0,0.1), ylim=c(0, 1000000)) +
    labs(title="Area Vs Population",
         subtitle="From midwest dataset",
         y="Population", x="Area",
   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



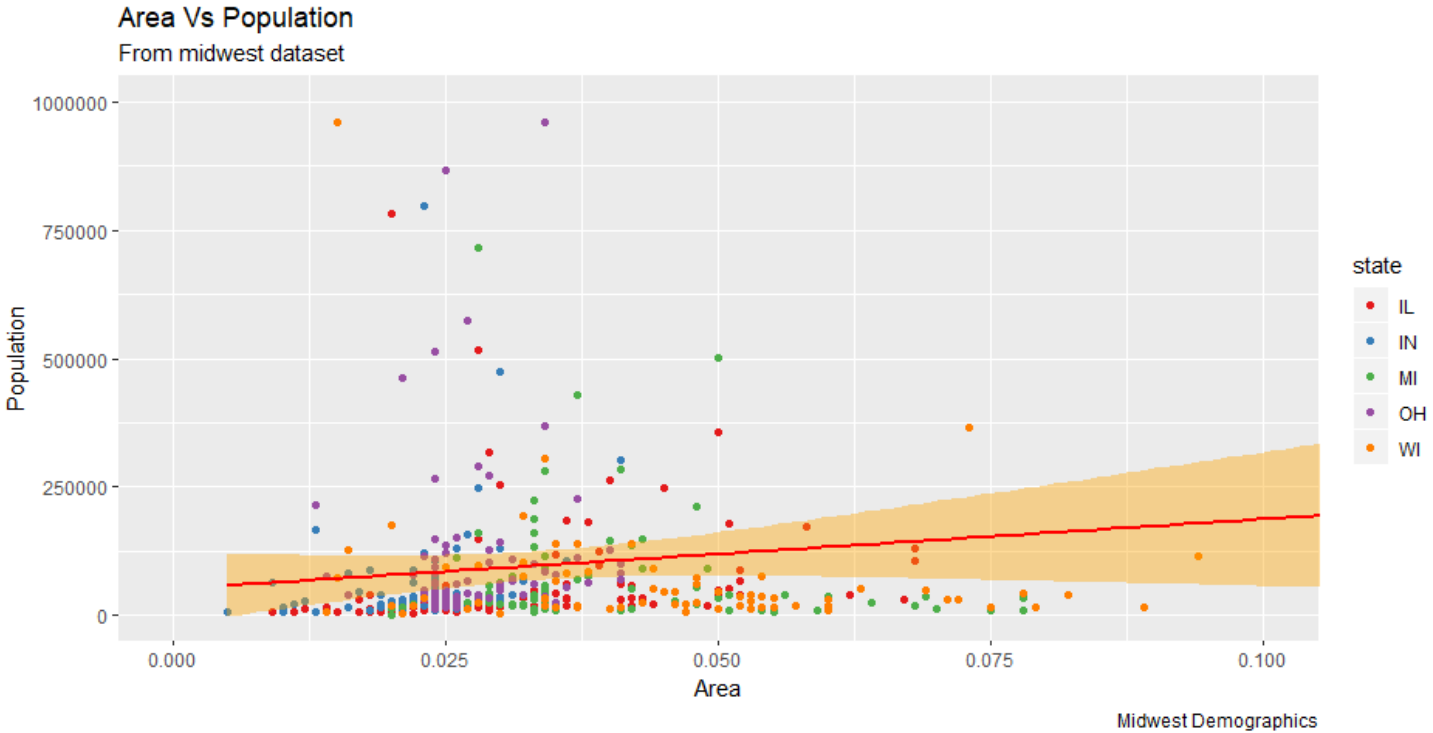
Color

Thêm màu cho đường smooth

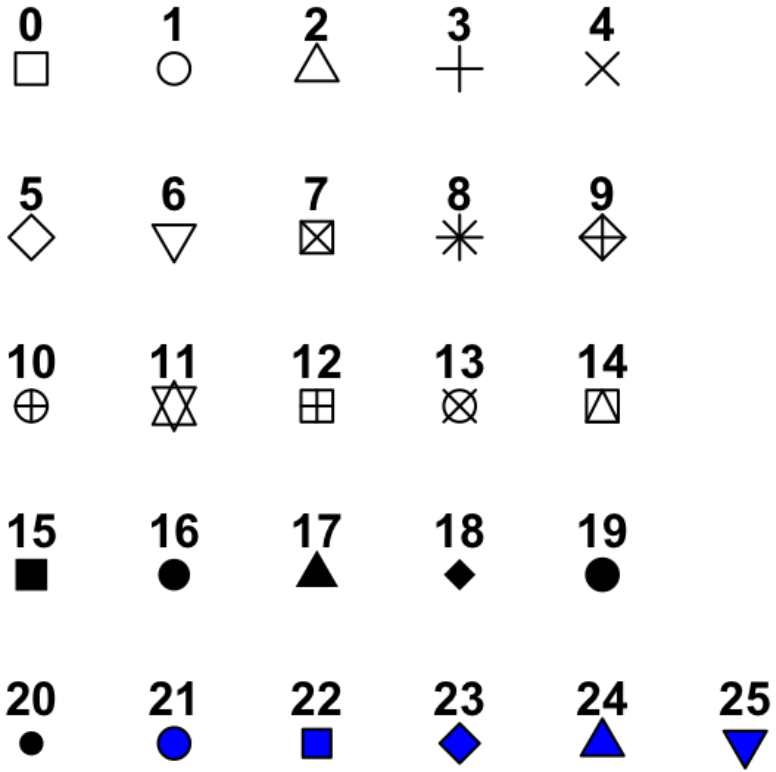
```
ggplot(data = midwest, aes(x = area, y = poptotal)) +
    geom_point(aes(col = state)) +
    geom_smooth(method = "lm", color = "red", fill = "orange") +
    coord_cartesian(xlim=c(0,0.1), ylim=c(0, 1000000)) +
    labs(title="Area Vs Population",
         subtitle="From midwest dataset",
         y="Population", x="Area",
         caption="Midwest Demographics")
```



```
ggplot(data = midwest, aes(x = area, y = poptotal)) +
    geom_point(aes(col = state)) +
    geom_smooth(method = "lm", color = "red", fill = "orange") +
    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="Midwest Demographics") +
     scale_colour_brewer(palette = "Set1")
```

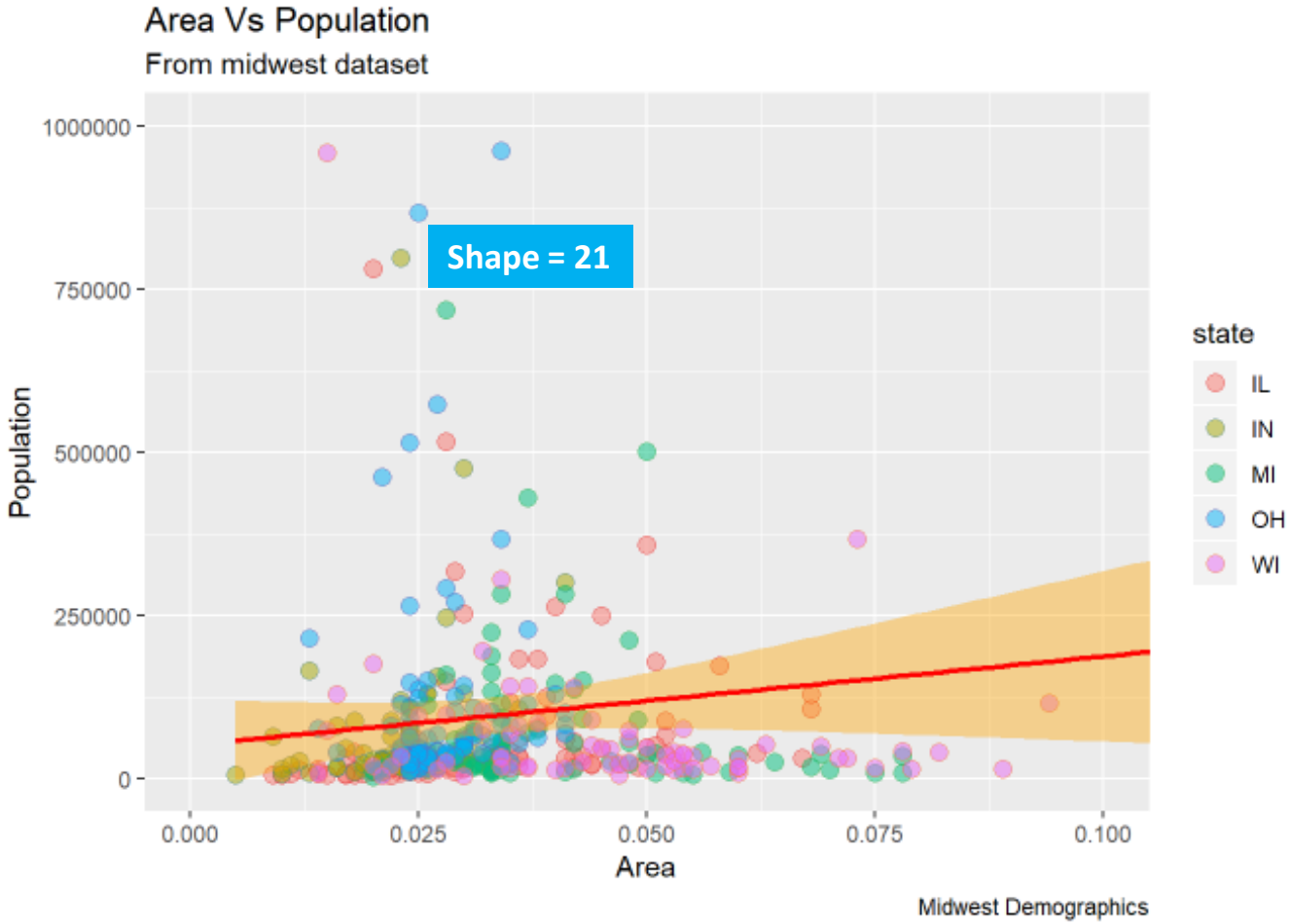


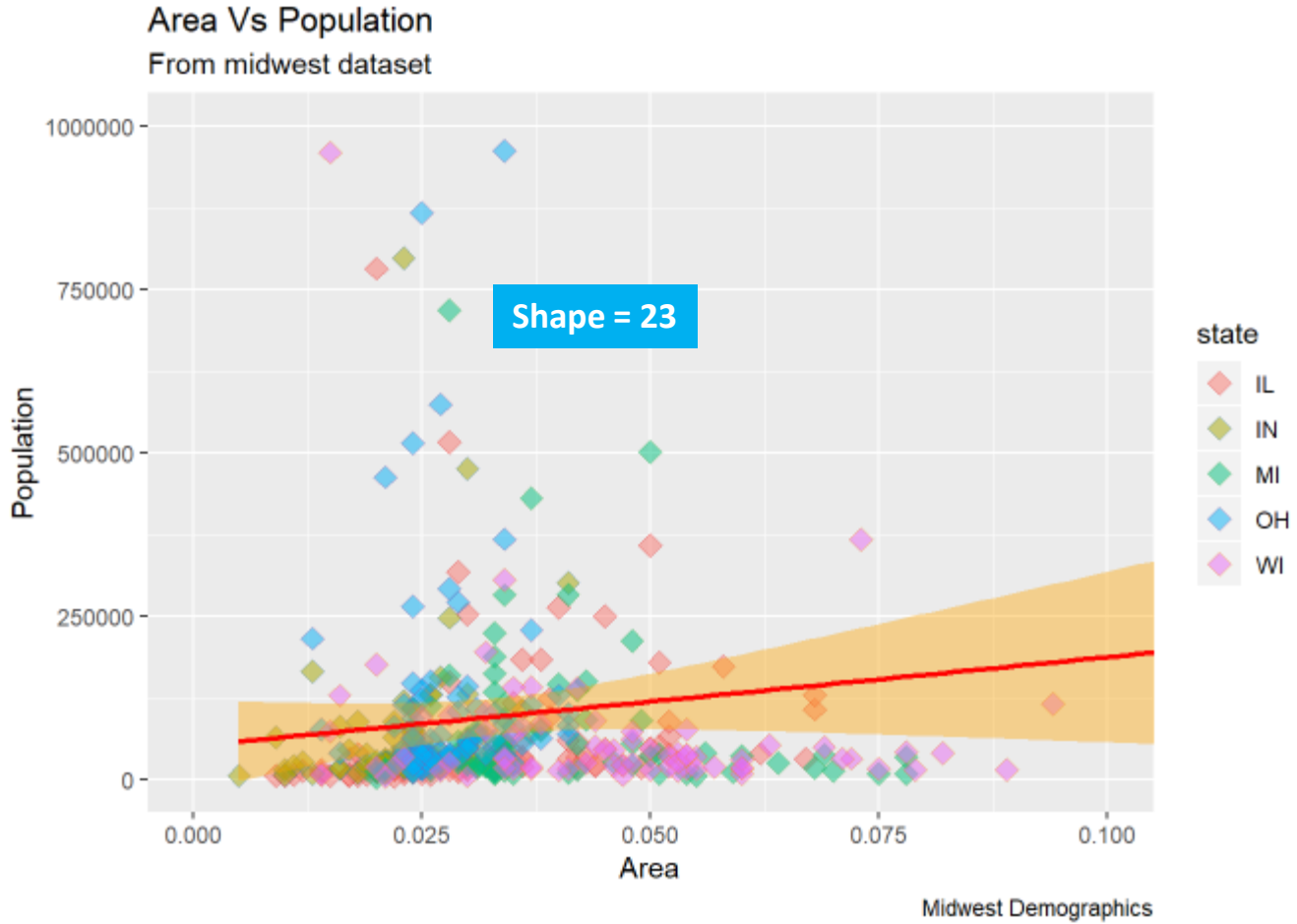
Size & shape



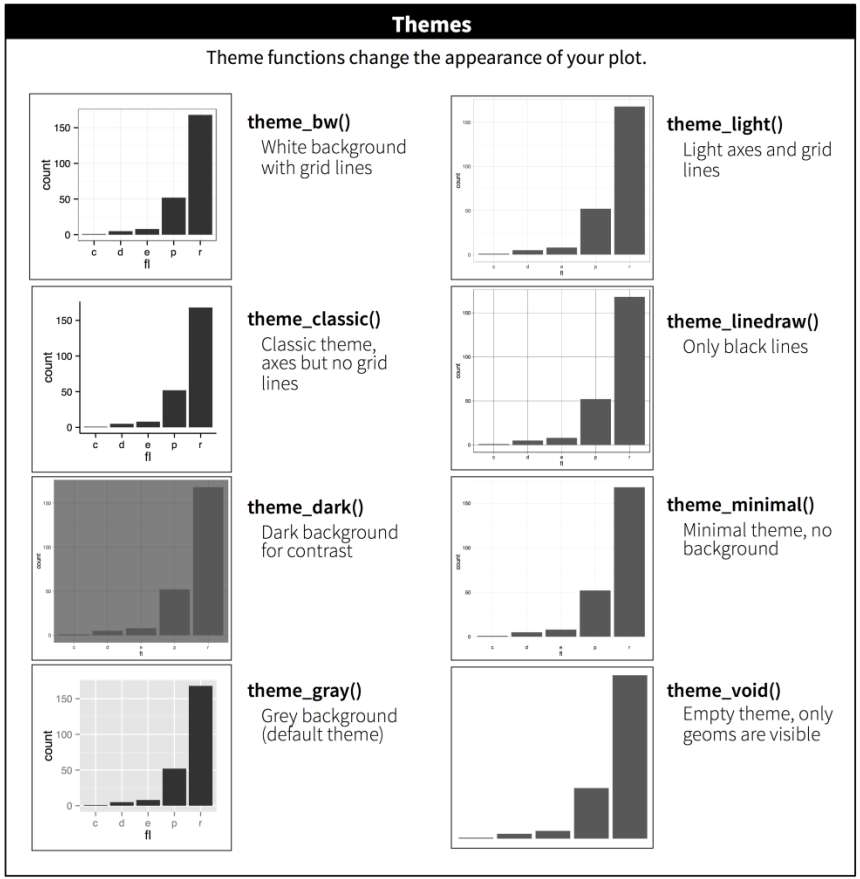
```
ggplot(data = midwest, aes(x = area, y = poptotal)) +
    geom_point(aes(col = state, fill = state), size = 3, shape = 21, alpha = 0.5) +
    geom_smooth(method = "lm", color = "red", fill = "orange") +
    coord_cartesian(xlim=c(0,0.1), ylim=c(0, 1000000)) +
    labs(title="Area Vs Population",
         subtitle="From midwest dataset",
         y="Population", x="Area",
         caption="Midwest Demographics") +
     scale_colour_brewer(palette = "Set1")
```

```
ggplot(data = midwest, aes(x = area, y = poptotal)) +
    geom_point(aes(col = state, fill = state), size = 3, shape = 23, alpha = 0.5) +
    geom_smooth(method = "lm", color = "red", fill = "orange") +
    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="Midwest Demographics") +
     scale_colour_brewer(palette = "Set1")
```


Area Vs Population
From midwest dataset
Shape = 21
Population
1000000
750000
500000
250000
0
0.000
0.025
0.050
0.075
0.100
Area
state
IL
IN
MI
OH
WI
Midwest Demographics


Area Vs Population
From midwest dataset
Shape = 23
Population
1000000
750000
500000
250000
0
0.000
0.025
0.050
0.075
0.100
Area
state
IL
IN
MI
OH
WI
Midwest Demographics

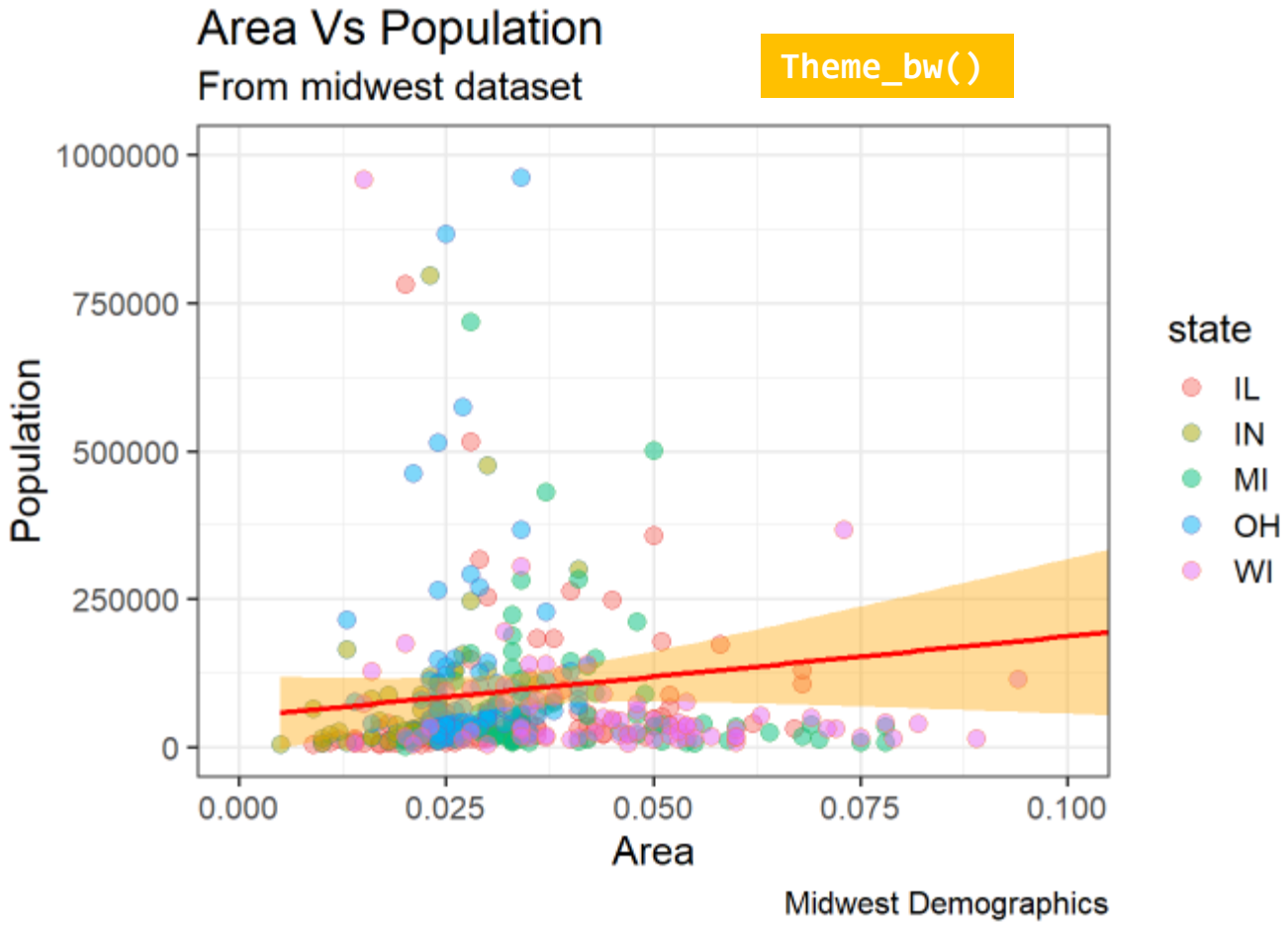
Theme


Themes
Theme functions change the appearance of your plot.
theme_bw()
White background with grid lines
theme_light()
Light axes and grid lines
theme_classic()
Classic theme, axes but no grid lines
theme_linedraw()
Only black lines
theme_dark()
Dark background for contrast
theme_minimal()
Minimal theme, no background
theme_gray()
Grey background (default theme)
theme_void()
Empty theme, only geoms are visible
count
fl
c d e p r

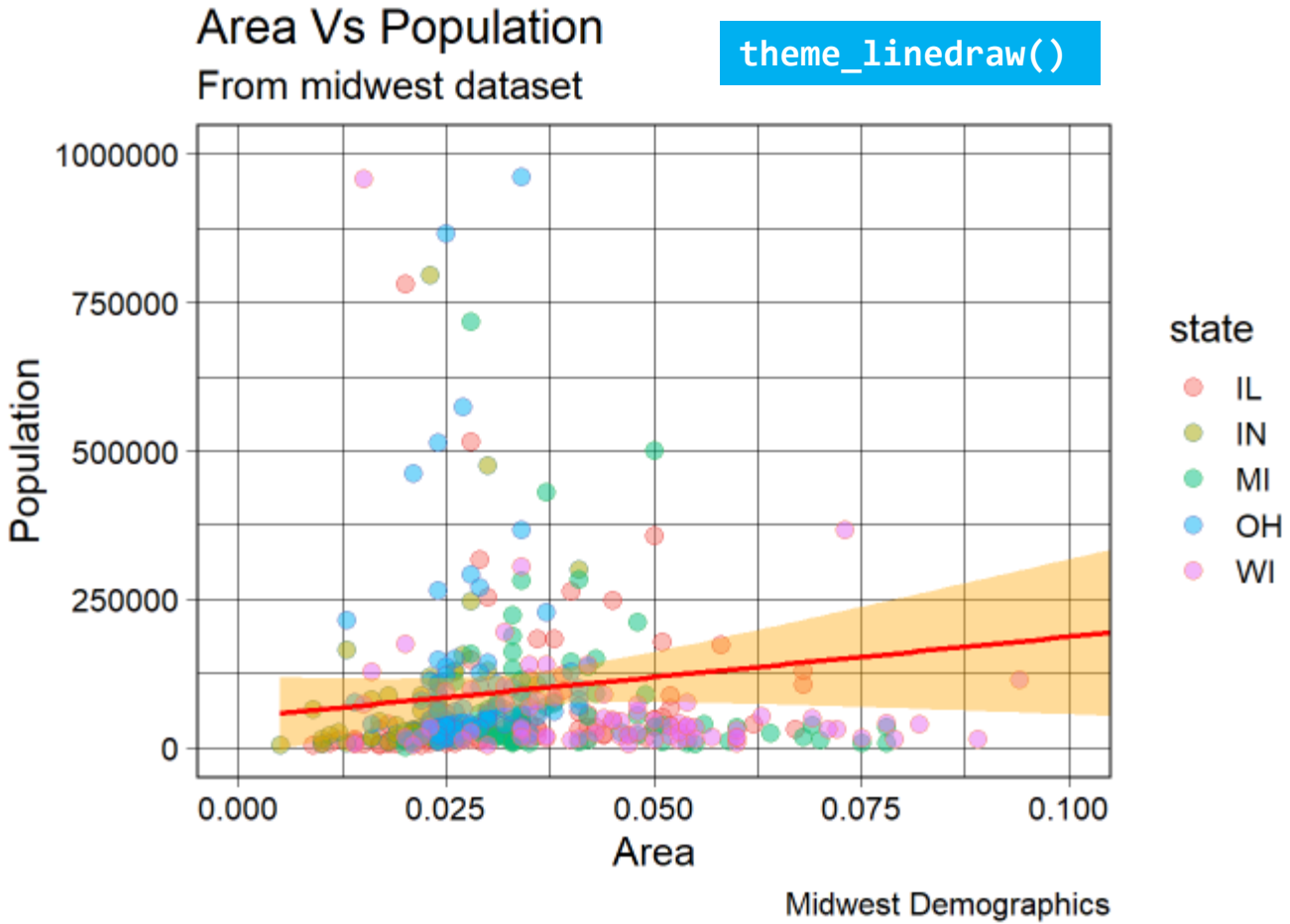
```
ggplot(data = midwest, aes(x = area, y = poptotal)) +
    geom_point(aes(col = state, fill = state), size = 3, shape = 21, alpha = 0.5) +
    geom_smooth(method = "lm", color = "red", fill = "orange") +
    coord_cartesian(xlim=c(0,0.1), ylim=c(0, 1000000)) +
    labs(title="Area Vs Population",
          subtitle="From midwest dataset",
          y="Population", x="Area",
          caption="Midwest Demographics") +
    scale_colour_brewer(palette = "Set1") +
    theme_bw(15)
```

```
ggplot(data = midwest, aes(x = area, y = poptotal)) +
    geom_point(aes(col = state, fill = state), size = 3, shape = 21, alpha = 0.5) +
    geom_smooth(method = "lm", color = "red", fill = "orange") +
    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="Midwest Demographics") +
    scale_colour_brewer(palette = "Set1") +
    theme_linedraw(15)
```

Theme_bw()


Area Vs Population
From midwest dataset
Population
1000000
750000
500000
250000
0
0.000
0.025
0.050
0.075
0.100
Area
state
IL
IN
MI
OH
WI
Midwest Demographics

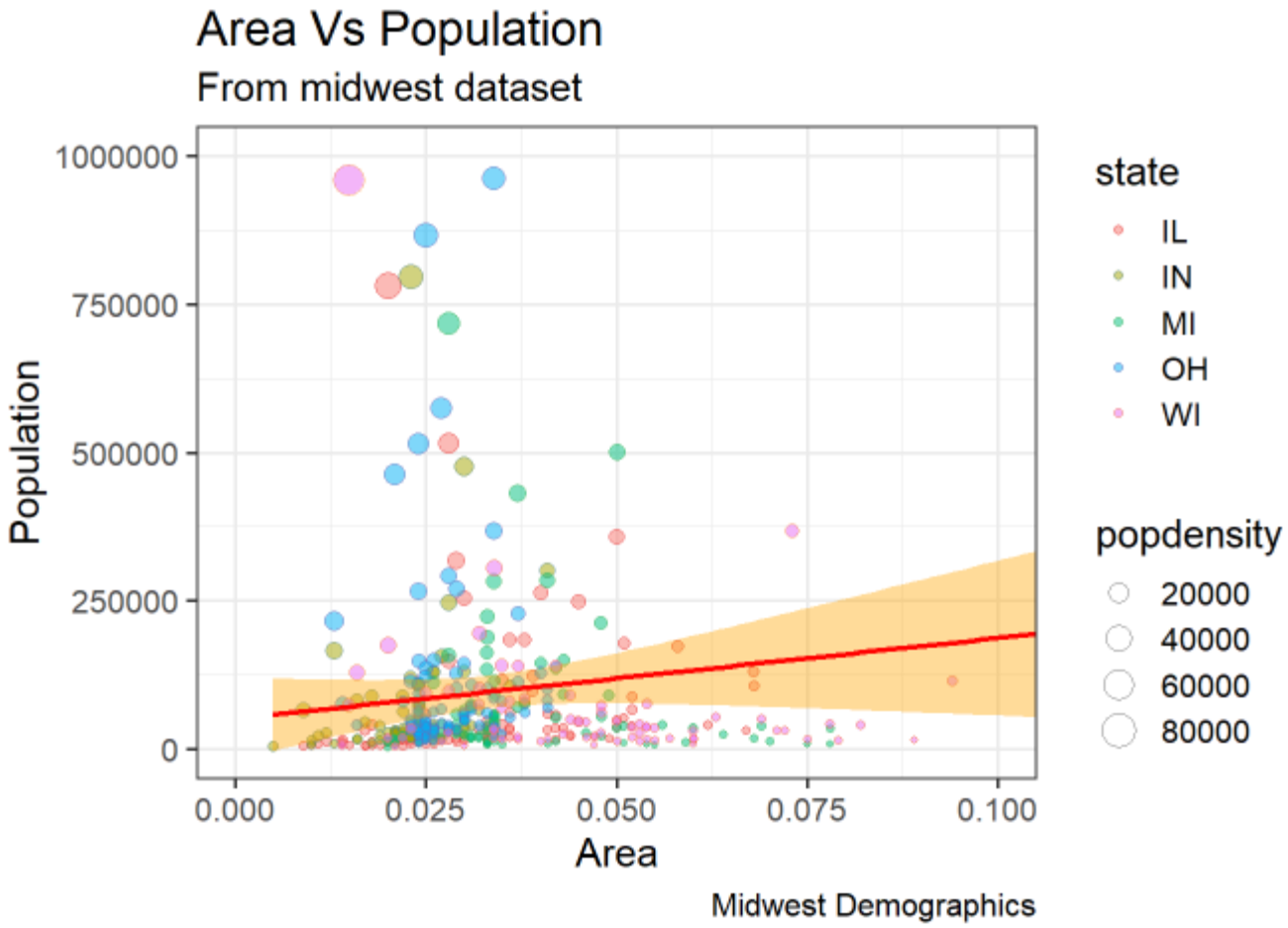
`theme_linedraw()`

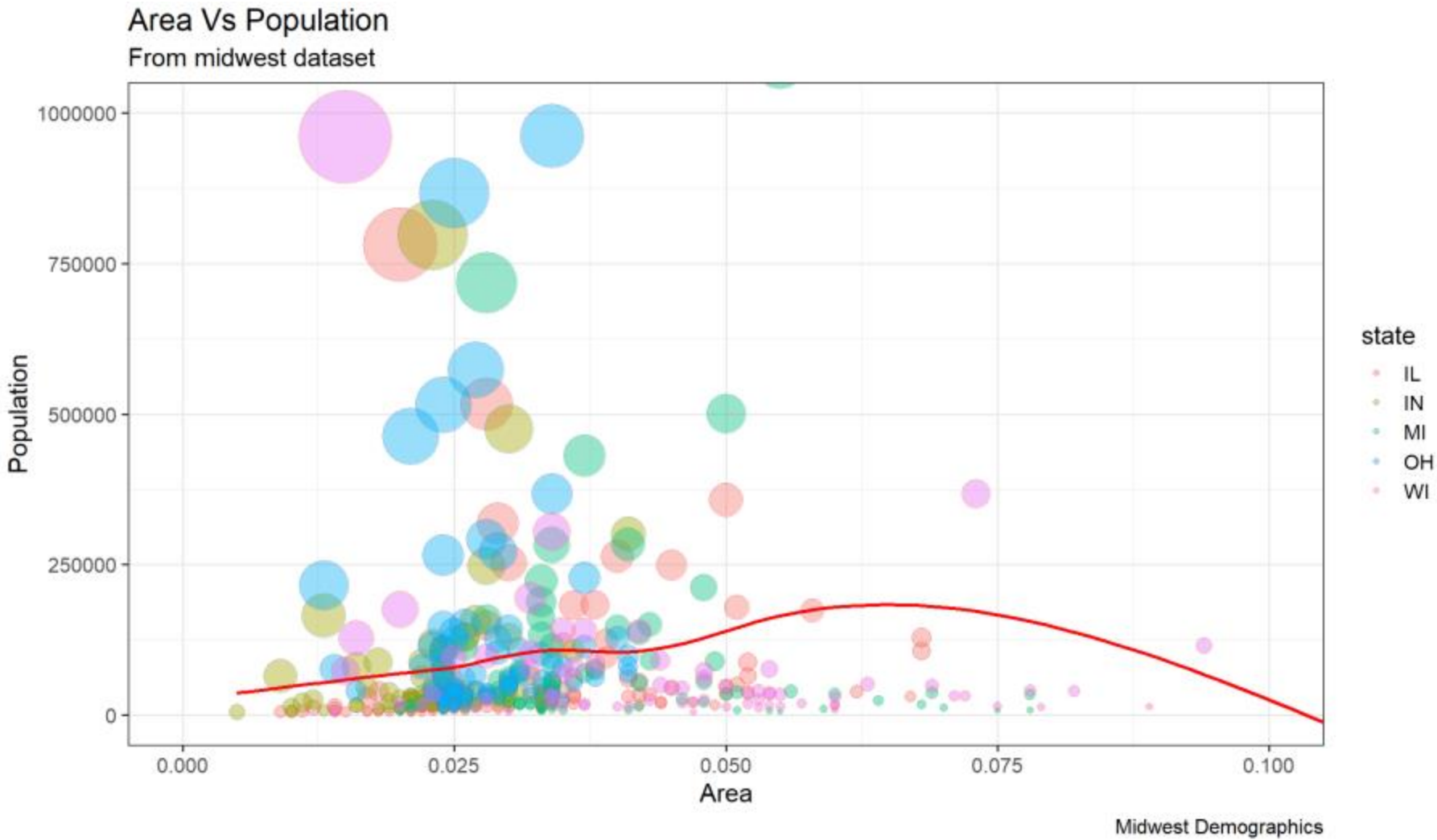


Tóm tắt

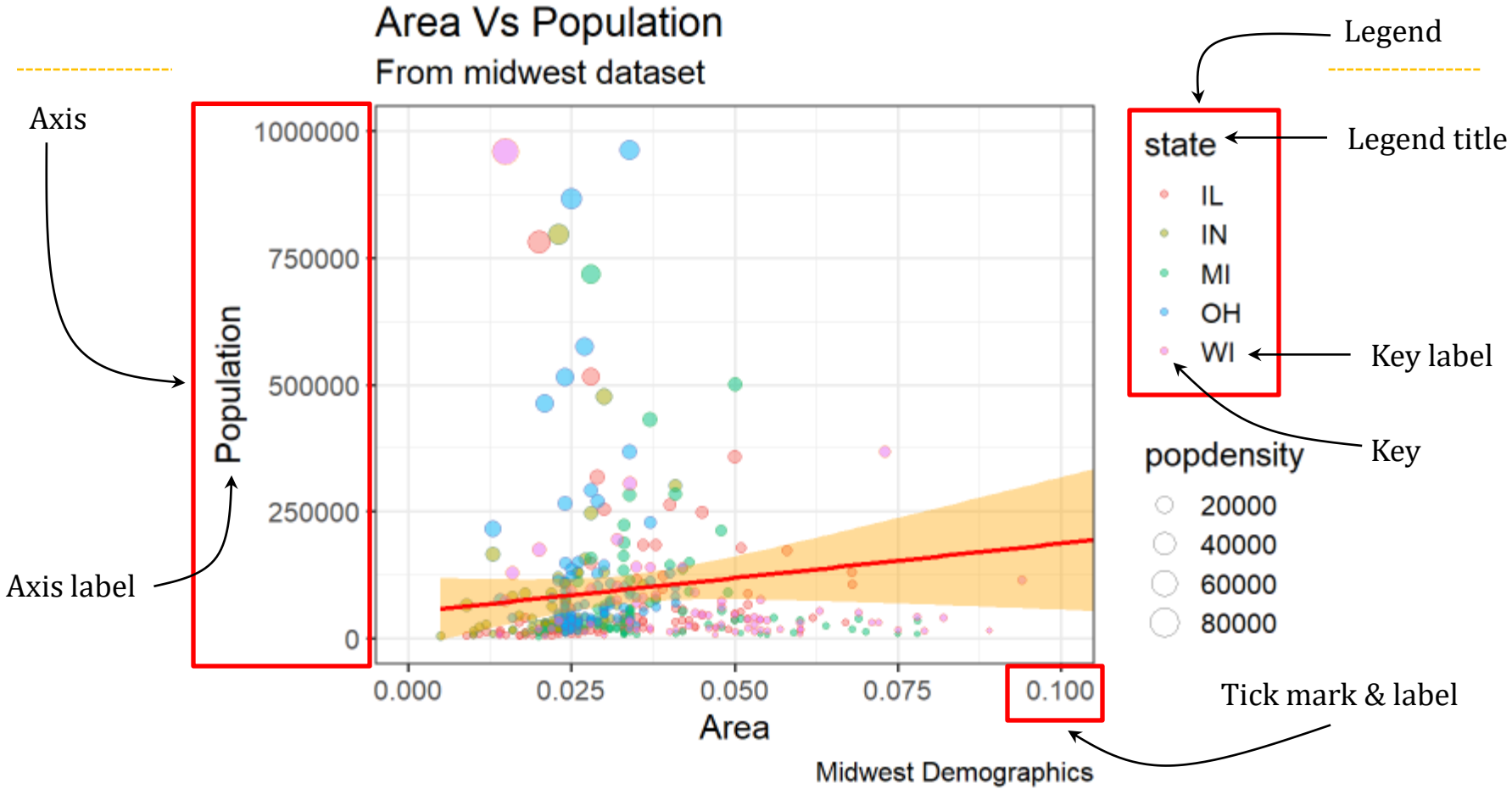
```
ggplot(data = midwest, aes(x = area, y = poptotal)) +
    geom_point(aes(size = popdensity, col = state, fill = state), shape = 21, alpha = 0.5) +
    geom_smooth(method = "lm", color = "red", fill = "orange") +
    coord_cartesian(xlim=c(0,0.1), ylim=c(0, 1000000)) +
    labs(title="Area Vs Population",
          subtitle="From midwest dataset",
          y="Population", x="Area",
          caption="Midwest Demographics") +
    scale_colour_brewer(palette = "Set1") +
    theme_bw(15)
```

```
ggplot(data = midwest, aes(x = area, y = poptotal, size = popdensity, col = state, fill =
state)) +
    geom_point(shape = 21, alpha = 0.4) +
    scale_size(range = c(1,30))+
    geom_smooth(aes(group = 1), se = F, color = "red", fill = "orange") +
    guides(size = F) +
    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="Midwest Demographics") +
    scale_colour_brewer(palette = "Set1") +
    theme_bw(15)
```


Area Vs Population
From midwest dataset
Population
1000000
750000
500000
250000
0
0.000
0.025
0.050
0.075
0.100
Area
Midwest Demographics
state
IL
IN
MI
OH
WI
popdensity
20000
40000
60000
80000


Area Vs Population
From midwest dataset
1000000
750000
500000
250000
0
Population
0.000
0.025
0.050
0.075
0.100
Area
state
IL
IN
MI
OH
WI
Midwest Demographics

Legend, text, labels & Annotation


Area Vs Population
From midwest dataset
Axis
Axis label
Legend
Legend title
Key label
Key
Tick mark & label
state
IL
IN
MI
OH
WI
popdensity
20000
40000
60000
80000
Population
1000000
750000
500000
250000
0
0.000
0.025
0.050
0.075
0.100
Area
Midwest Demographics

Thay đổi Legend

- Thay đổi title cho legend
- Thay đổi labels & color cho legend ứng với categories
- Remove hay thay đổi legend positions

Thay đổi title cho legend

- Có 3 cách

✓ Sử dụng `labs()`

```
labs(color ="State", fill ="State", size = "Density")
```

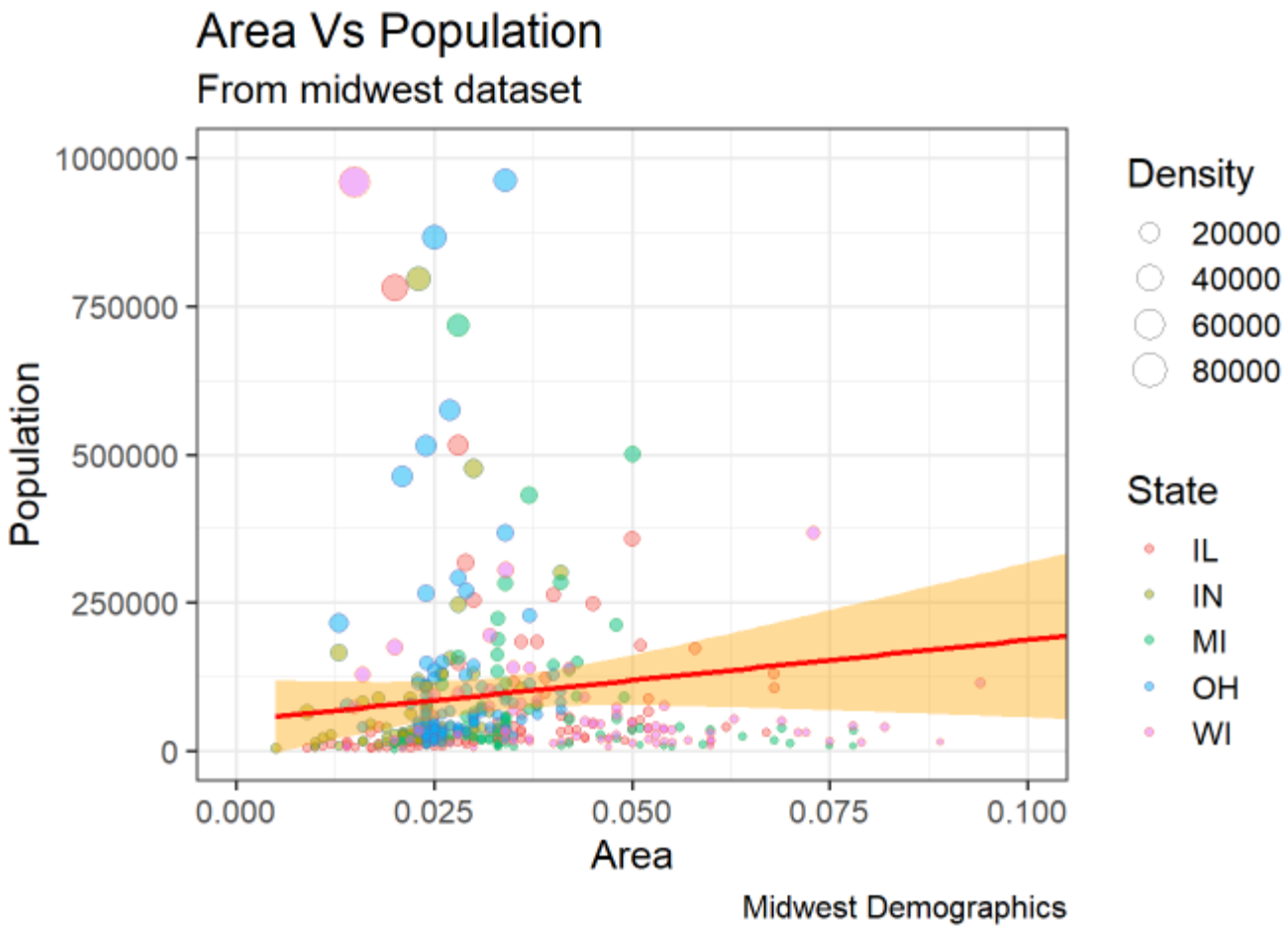
✓ Sử dụng guides()

```
guides(color = guide_legend("State"), fill =
guide_legend("State"), size = guide_legend("Density"))
```

✓ Sử dụng scale_[…]

Tùy vào việc sử dụng loại “scale_”, cho phép thay đổi legend cho màu “scale” tương ứng (ví dụ trong trường hợp này)

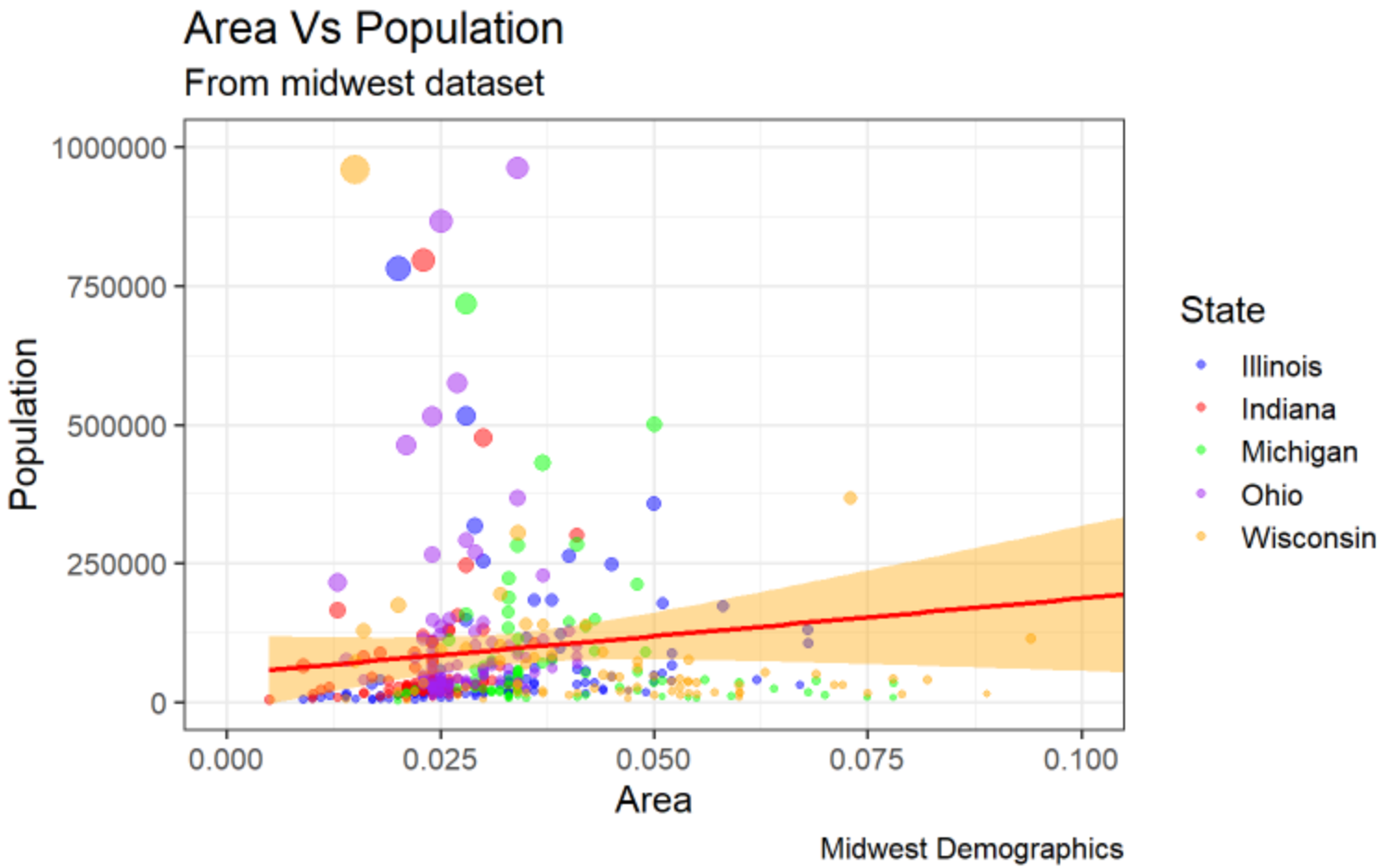
```
scale_color_discrete(name="State") +
scale_fill_discrete(name="State") +
scale_size_continuous(name = "Density", guide = FALSE)
```


Area Vs Population
From midwest dataset
Population
1000000
750000
500000
250000
0
0.000
0.025
0.050
0.075
0.100
Area
Midwest Demographics
Density
20000
40000
60000
80000
State
IL
IN
MI
OH
WI

Thay đổi labels & color cho legend

- Có thể sử dụng chức năng scale_[color/fill]_manual()

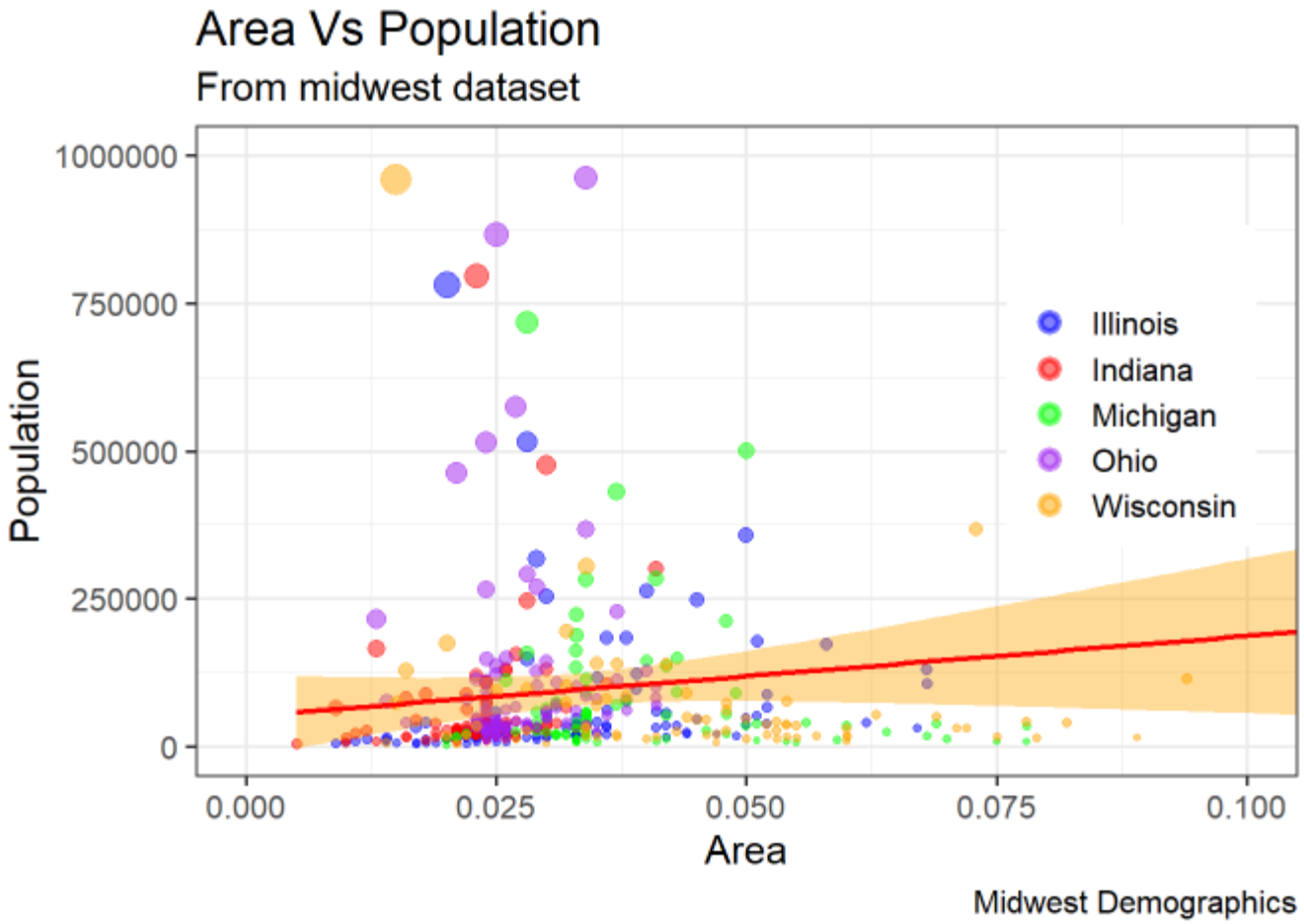
```
ggplot(data = midwest, aes(x = area, y = poptotal)) +
    geom_point(aes(size = popdensity, col = state), alpha = 0.5) +
    geom_smooth(method = "lm", color = "red", fill = "orange") +
    coord_cartesian(xlim=c(0,0.1), ylim=c(0, 1000000)) +
    labs(title="Area Vs Population",
          subtitle="From midwest dataset",
          y="Population", x="Area",
          caption="Midwest Demographics") +
    scale_color_manual(name = "State",
                       labels = c("Illinois",
                                  "Indiana",
                                  "Michigan",
                                  "Ohio",
                                  "Wisconsin"),
                       values = c("IL"="blue",
                                  "IN"="red",
                                  "MI"="green",
                                  "OH"="purple",
                                  "WI"="orange")) +
    scale_size_continuous(name = "Density", guide = F)+
    theme_bw(15)
```


Area Vs Population
From midwest dataset
1000000
750000
500000
250000
0
Population
0.000
0.025
0.050
0.075
0.100
Area
Midwest Demographics
State
Illinois
Indiana
Michigan
Ohio
Wisconsin

Remove hay thay đổi legend positions

- Sử dụng `theme(legend.position = "...")`
- `"left"/“right"/"top"/"bottom"`
- `theme(legend.position = c(0.85,0.5))`

```r
ggplot(data = midwest, aes(x = area, y = poptotal)) +
    geom_point(aes(size = popdensity, col = state), alpha = 0.5) +
    geom_smooth(method = "lm", color = "red", fill = "orange") +
    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="Midwest Demographics") +
    scale_color_manual(name = "State",
                       labels = c("Illinois",
                                  "Indiana",
                                  "Michigan",
                                  "Ohio",
                                  "Wisconsin"),
                       values = c("IL"="blue",
                                  "IN"="red",
                                  "MI"="green",
                                  "OH"="purple",
                                  "WI"="orange")) +
    scale_size_continuous(name = "Density", guide = F)+
    guides(color = guide_legend(override.aes = list(size=2, stroke = 2))) +
    theme(legend.position = c(0.85, 0.6))+
    theme_bw(15)
```


Area Vs Population
From midwest dataset
1000000
750000
500000
250000
0
Population
0.000
0.025
0.050
0.075
0.100
Area
Illinois
Indiana
Michigan
Ohio
Wisconsin
Midwest Demographics

Text & labels

- Mục tiêu: hiện tên “country” của các nước có dân số >500,000

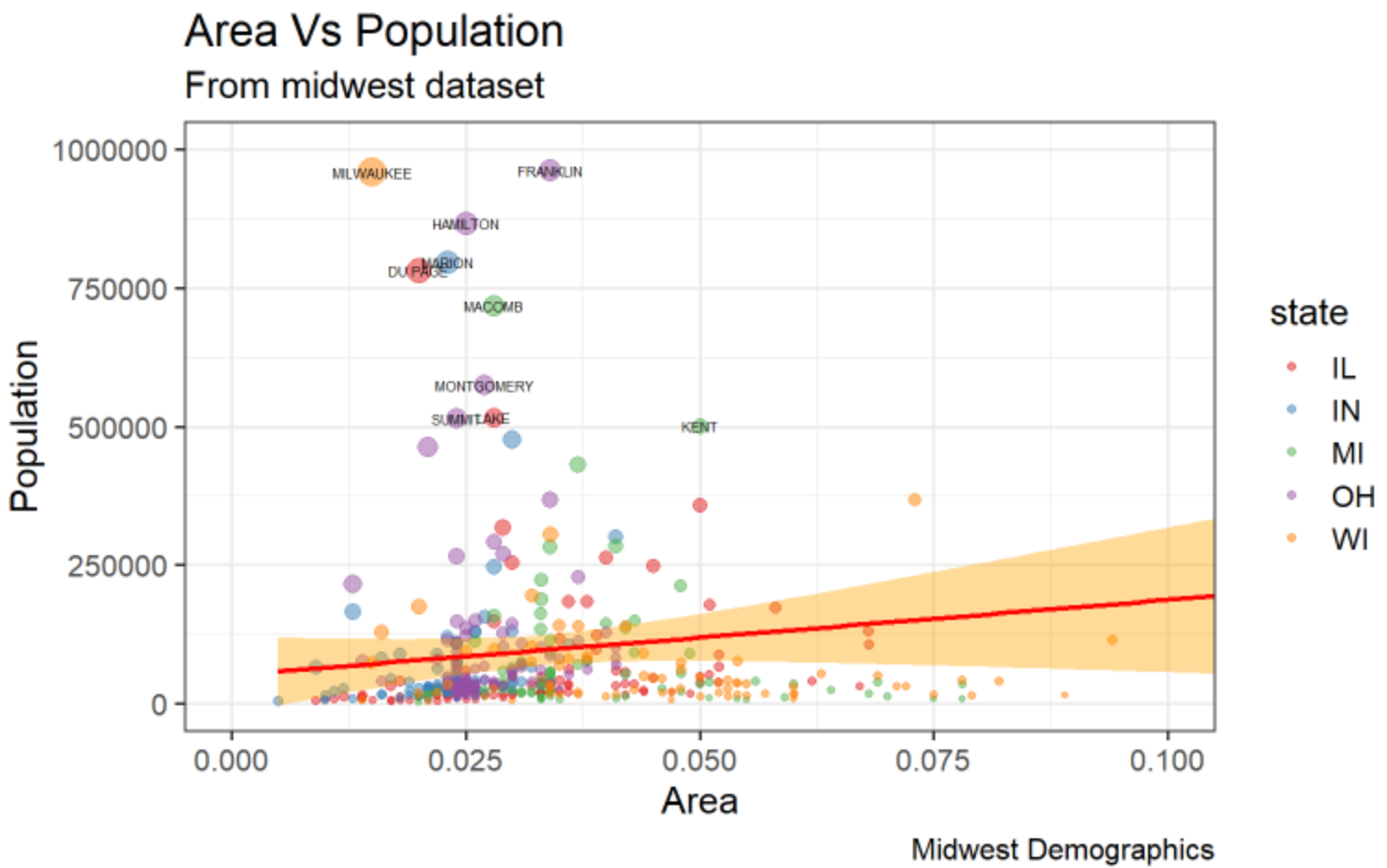
B1: Lọc những nước có poptotal > 500,000 (đặt tên là large_country)

```
large_country = midwest %>% filter(poptotal > 500000)
```

B2: sử dụng `geom_text()`

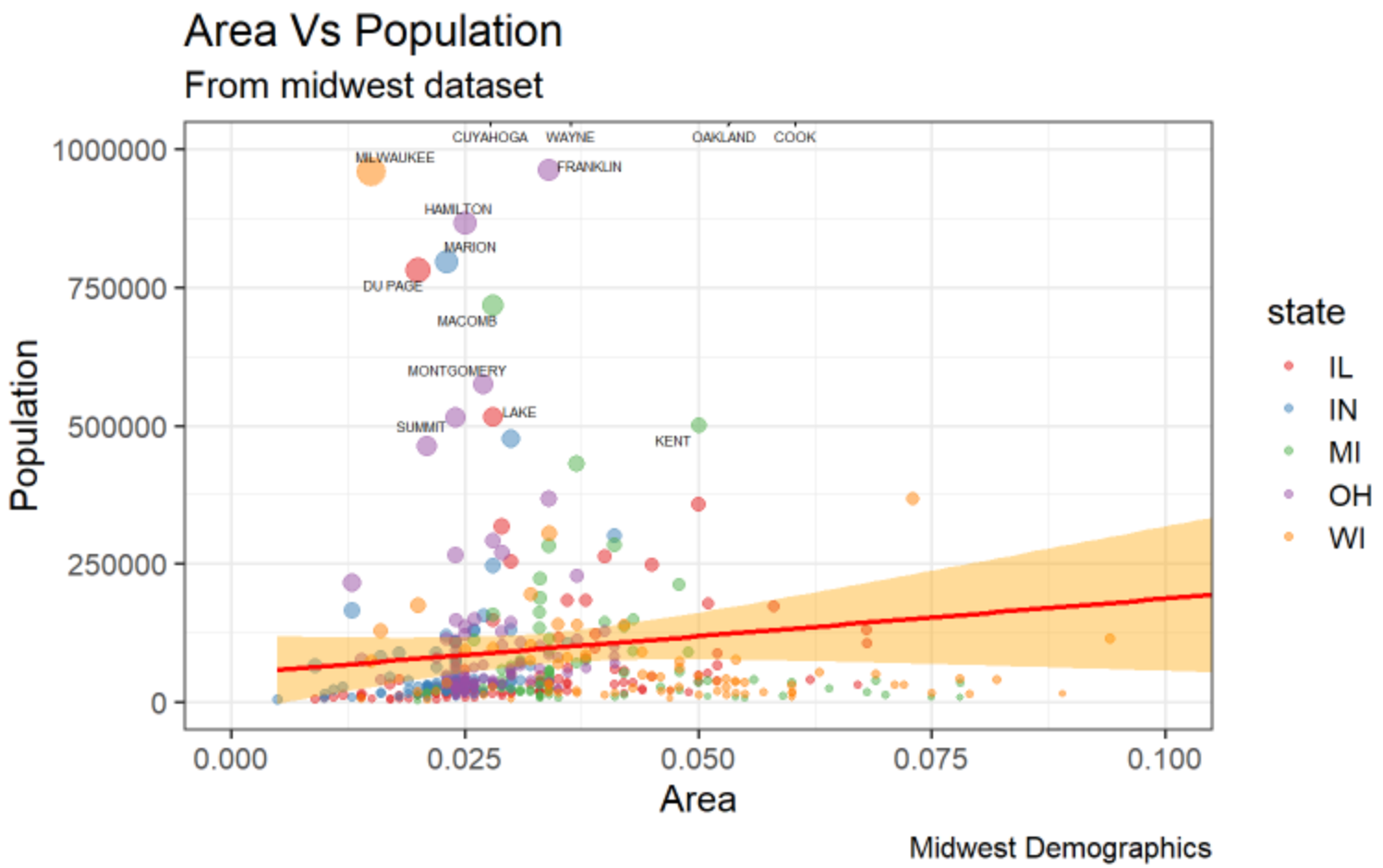
```r
large_country = midwest %>% filter(poptotal > 500000)

ggplot(data = midwest, aes(x = area, y = poptotal)) +
    geom_point(aes(size = popdensity, col = state), alpha = 0.5) +
    geom_smooth(method = "lm", color = "red", fill = "orange") +
    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="Midwest Demographics") +
    scale_colour_brewer(palette = "Set1") +
    scale_size_continuous(name = "Density", guide = F)+
    theme_bw(15)+
    geom_text(data = large_country, aes(label = county), size = 2)
```



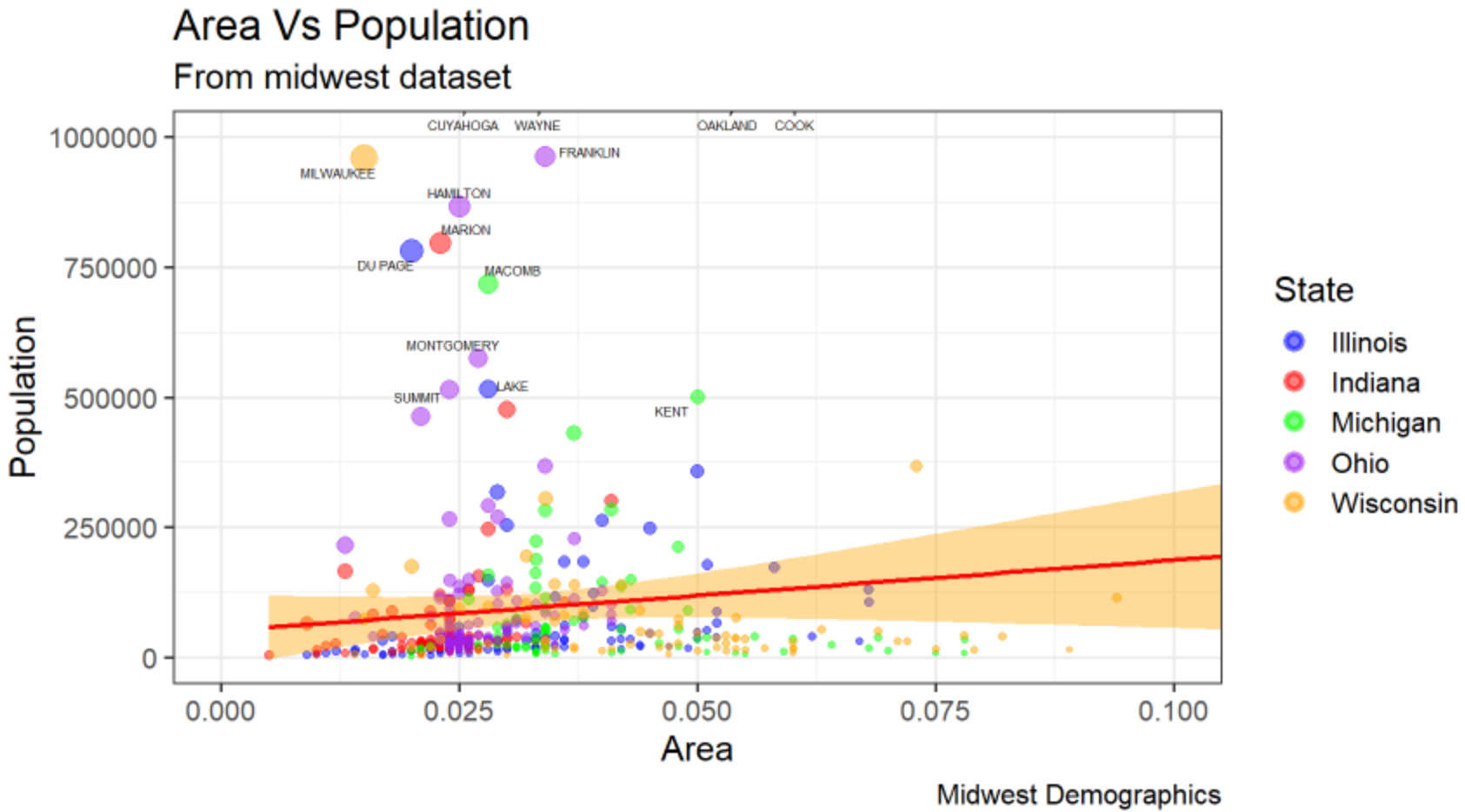
Text trùng với point ➔ chỉnh vị trí text bằng package “ggrepel” ➔ thay geom_text() = geom_text_repel()

```
large_country = midwest %>% filter(poptotal > 500000)
library(ggrepel)
ggplot(data = midwest, aes(x = area, y = poptotal)) +
    geom_point(aes(size = popdensity, col = state), alpha = 0.5) +
    geom_smooth(method = "lm", color = "red", fill = "orange") +
    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="Midwest Demographics") +
    scale_colour_brewer(palette = "Set1") +
    scale_size_continuous(name = "Density", guide = F)+
    theme_bw(15)+
    geom_text_repel(data = large_country, aes(label = county), size = 2)
```


Area Vs Population
From midwest dataset
CUYAHOGA
WAYNE
OAKLAND
COOK
MILWAUKEE
FRANKLIN
HAMILTON
MARION
DU PAGE
MACOMB
MONTGOMERY
LAKE
SUMMIT
KENT
1000000
750000
500000
250000
0
Population
0.000
0.025
0.050
0.075
0.100
Area
state
IL
IN
MI
OH
WI
Midwest Demographics

```r
large_country = midwest %>% filter(poptotal > 500000)
library(ggrepel)

p <- ggplot(data = midwest, aes(x = area, y = poptotal)) +
    geom_point(aes(size = popdensity, col = state), alpha = 0.5) +
    geom_smooth(method = "lm", color = "red", fill = "orange") +
    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="Midwest Demographics") +
    scale_color_manual(name = "State",
                       labels = c("Illinois",
                                  "Indiana",
                                  "Michigan",
                                  "Ohio",
                                  "Wisconsin"),
                       values = c("IL"="blue",
                                  "IN"="red",
                                  "MI"="green",
                                  "OH"="purple",
                                  "WI"="orange")) +
    guides(color = guide_legend(override.aes = list(size=2, stroke = 2))) +
    scale_size_continuous(name = "Density", guide = F)+
    theme_bw(15) +
    geom_text_repel(data = large_country, aes(label = county), size = 2)
p
```


Area Vs Population
From midwest dataset
CUYAHOGA
WAYNE
OAKLAND
COOK
FRANKLIN
MILWAUKEE
HAMILTON
MARION
DU PAGE
MACOMB
MONTGOMERY
LAKE
SUMMIT
KENT
1000000
750000
500000
250000
0
Population
0.000
0.025
0.050
0.075
0.100
Area
State
Illinois
Indiana
Michigan
Ohio
Wisconsin
Midwest Demographics

Advance theme & font

Theme

- `?theme()`
- element_blank(): draws nothing, and assigns no space.
- element_rect(): borders and backgrounds.
- element_line(): lines.
- element_text(): text.
- Cài đặt thêm font

```
install.packages("extrafont")
library(extrafont)
font_import()
loadfonts(device = "win")
```

```r
p <- p + theme(
        # Chọn font chữ
        text = element_text(family = "Georgia"),
        # Tùy chỉnh text cho title (cỡ chữ 18, bold)
        plot.title = element_text(size = 18,color = "grey10",  face = "bold"),
        # Tùy chỉnh cho subtitle
        plot.subtitle = element_text(color = "gray40", size = 12),
        # Tùy chỉnh cation
        plot.caption = element_text(face = "italic", size = 12, color = "red"),
        # Tùy chỉnh title cho trục x
        axis.title.x = element_text(face = "bold", size = 11, color = "grey20"),
        # Tùy chỉnh title cho trục y
        axis.title.y = element_text(face = "bold", size = 11, color = "grey20"),
        # Tùy chỉnh background, grid
        panel.grid.major = element_line(color = "#4d5566"),
        panel.grid.minor.y = element_blank(),
        panel.grid.minor.x = element_blank(),
        panel.background = element_rect(fill = "#444B5A"),
        # Tùy chỉnh hiện thị đơn vị trục x, y
        axis.text.x = element_text(size = 13, color = "grey10"),
        axis.text.y = element_text(size = 13, color = "grey10"),
        # Tùy chỉnh tick cho 2 trục
        axis.ticks = element_line(size = 13)
    )
p
```

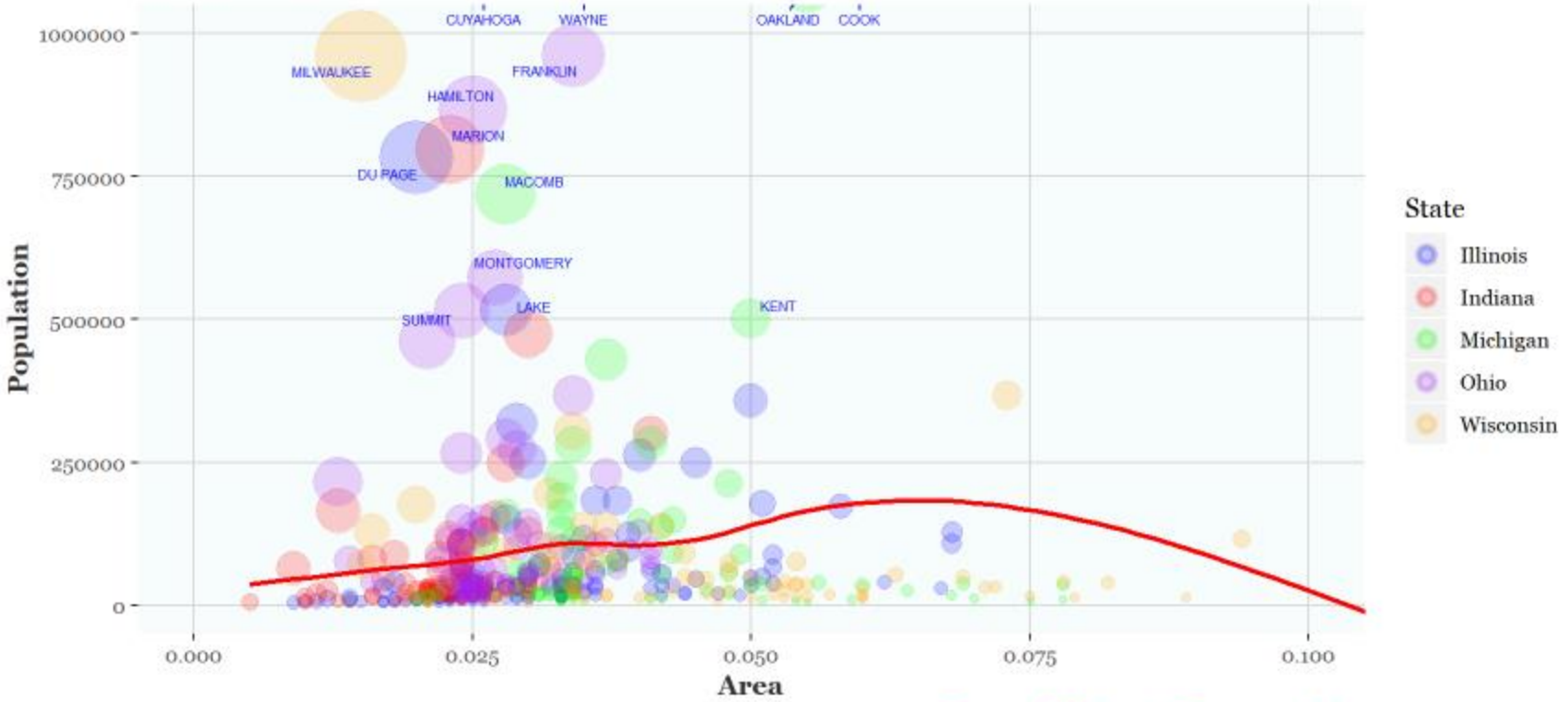
Area Vs Population

From midwest dataset

Source: Midwest Demographics

Area Vs Population

From midwest dataset
Bubble plot style



Source: Midwest Demographics

Facet

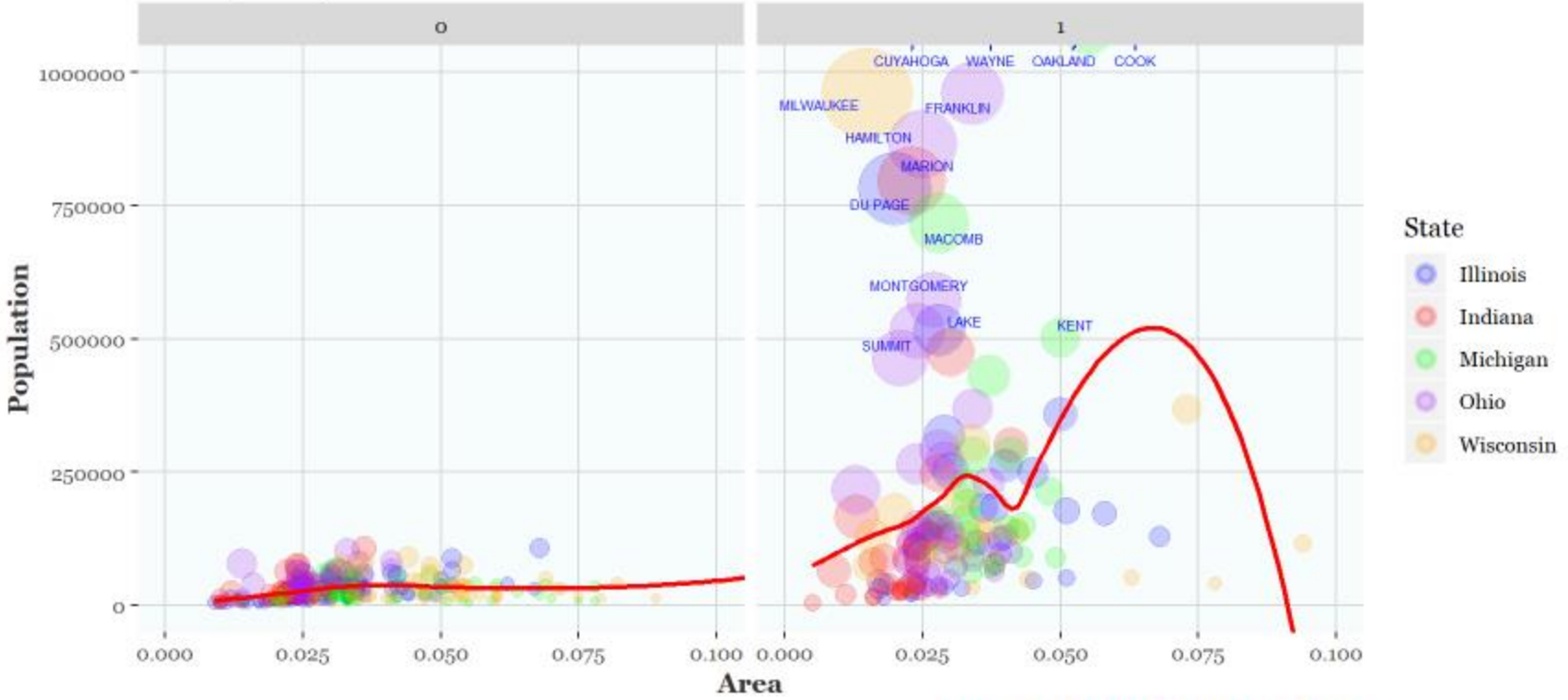
Facet

- Phân chia theo 2 nhóm khu vực “inmetro”: Yes/No

p + facet_grap(~inmetro, ncol = 2)

Area Vs Population

From midwest dataset
Bubble plot style



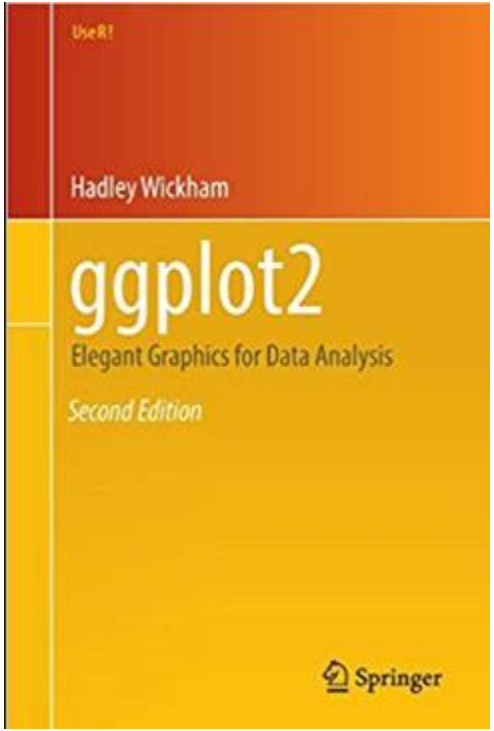
Source: Midwest Demographics

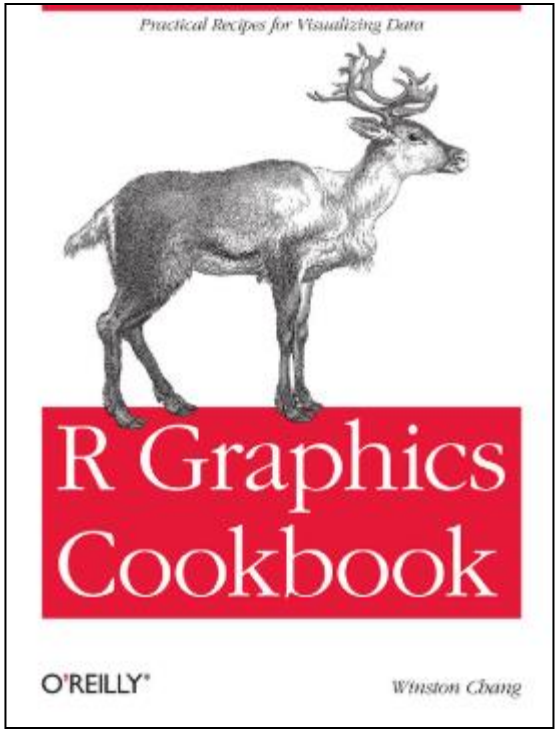
Tìm hiểu thêm

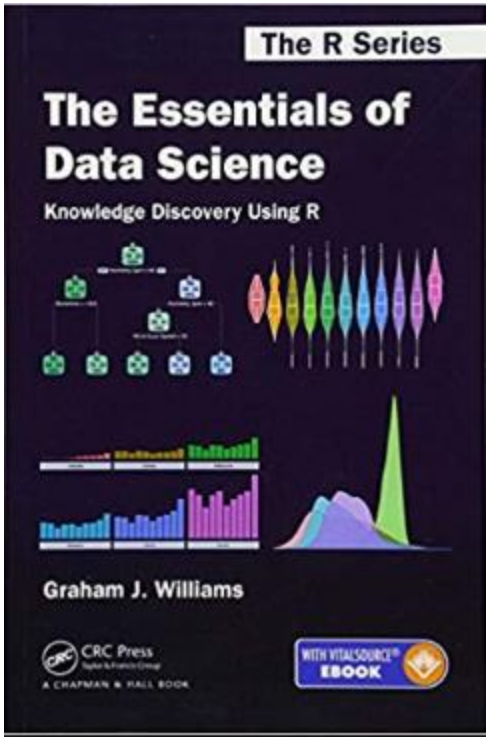
- coordinate
- Ghép các ảnh riêng lẻ vào 1 qua package “gridExtra”

Thực hành

Tài liệu tham khảo







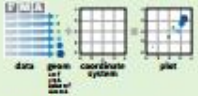
Data Visualization with ggplot2 Cheat Sheet

RStudio

Basics

ggplot2 is based on the grammar of graphics, the idea that you can build every graph from the same few components: a data set, a set of geoms—visual marks that represent data points, and a coordinate system.

To display data values, map variables in the data set to aesthetic properties of the geom like size, color, and x and y locations.



Build a graph with ggplot() or qplot()

```
ggplot(data = mpg, aes(x = cty, y = hwy))
```

Begins a plot that you finish by adding layers to. No defaults, but provides more control than qplot().

```
ggplot(mpg, aes(hwy, cty)) +
  geom_point(aes(color = cyl)) +
  geom_smooth(method ="lm") +
  coord_cartesian() +
  scale_color_gradient() +
  theme_bw()
```

data — add layers, elements with + — layer = geom + default stat + layer specific mappings — additional elements

Add a new layer to a plot with a geom_*() or stat_*() function. Each provides a geom, a set of aesthetic mappings, and a default stat and position adjustment.

aesthetic mappings — data — geom

```
qplot(x = cty, y = hwy, color = cyl, data = mpg, geom = "point")
```

Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

```
last_plot()
```

Returns the last plot

```
ggsave("plot.png", width = 5, height = 5)
```

Saves last plot as 5" x 5" file named "plot.png" in working directory. Matches file type to file extension.

Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

Graphical Primitives

```
a <- ggplot(seals, aes(x = long, y = lat))
b <- ggplot(economics, aes(date, unemploy))
```

- a + geom_blank() (Useful for expanding limits)
- a + geom_curve(aes(yend = lat + delta_lat, xend = long + delta_long, curvature = z)) x, xend, y, yend, alpha, angle, color, curvature, linetype, size
- b + geom_path(lineend="butt", linejoin="round", linemitre=1) x, y, alpha, color, group, linetype, size
- b + geom_polygon(aes(group = group)) x, y, alpha, color, fill, group, linetype, size
- a + geom_rect(aes(xmin = long, ymin = lat, xmax= long + delta_long, ymax = lat + delta_lat)) xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size
- b + geom_ribbon(aes(ymin=unemploy - 900, ymax=unemploy + 900)) x, ymax, ymin, alpha, color, fill, group, linetype, size
- a + geom_segment(aes(yend=lat + delta_lat, xend = long + delta_long)) x, xend, y, yend, alpha, color, linetype, size
- a + geom_spoke(aes(yend = lat + delta_lat, xend = long + delta_long)) x, y, angle, radius, alpha, color, linetype, size

One Variable

Continuous

```
c <- ggplot(mpg, aes(hwy))
```

- c + geom_area(stat = "bin") x, y, alpha, color, fill, linetype, size
  a + geom_area(aes(y = ..density..), stat = "bin")
- c + geom_density(kernel = "gaussian") x, y, alpha, color, fill, group, linetype, size, weight
- c + geom_dotplot() x, y, alpha, color, fill
- c + geom_freqpoly() x, y, alpha, color, group, linetype, size
  a + geom_freqpoly(aes(y = ..density..))
- c + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight
  a + geom_histogram(aes(y = ..density..))

Discrete

```
d <- ggplot(mpg, aes(fl))
```

- d + geom_bar() x, alpha, color, fill, linetype, size, weight

Two Variables

Continuous X, Continuous Y

```
e <- ggplot(mpg, aes(cty, hwy))
```

- e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE) x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust
- e + geom_jitter(height = 2, width = 2) x, y, alpha, color, fill, shape, size
- e + geom_point() x, y, alpha, color, fill, shape, size, stroke
- e + geom_quantile() x, y, alpha, color, group, linetype, size, weight
- e + geom_rug(sides = "bl") x, y, alpha, color, linetype, size
- e + geom_smooth(method = lm) x, y, alpha, color, fill, group, linetype, size, weight
- e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE) x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

Discrete X, Continuous Y

```
f <- ggplot(mpg, aes(class, hwy))
```

- f + geom_bar(stat = "identity") x, y, alpha, color, fill, linetype, size, weight
- f + geom_boxplot() x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight
- f + geom_dotplot(binaxis = "y", stackdir = "center") x, y, alpha, color, fill, group
- f + geom_violin(scale = "area") x, y, alpha, color, fill, group, linetype, size, weight

Discrete X, Discrete Y

```
g <- ggplot(diamonds, aes(cut, color))
```

- g + geom_count() x, y, alpha, color, fill, shape, size, stroke

Continuous Bivariate Distribution

```
h <- ggplot(diamonds, aes(carat, price))
```

- h + geom_bin2d(binwidth = c(0.25, 500)) x, y, alpha, color, fill, linetype, size, weight
- h + geom_density2d() x, y, alpha, colour, group, linetype, size
- h + geom_hex() x, y, alpha, colour, fill, size

Continuous Function

```
i <- ggplot(economics, aes(date, unemploy))
```

- i + geom_area() x, y, alpha, color, fill, linetype, size
- i + geom_line() x, y, alpha, color, group, linetype, size
- i + geom_step(direction = "hv") x, y, alpha, color, group, linetype, size

Visualizing error

```
df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))
```

- j + geom_crossbar(fatten = 2) x, y, ymax, ymin, alpha, color, fill, group, linetype, size
- j + geom_errorbar() x, ymax, ymin, alpha, color, group, linetype, size, width (also geom_errorbarh())
- j + geom_linerange() x, ymin, ymax, alpha, color, group, linetype, size
- j + geom_pointrange() x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

Maps

```
data <- data.frame(murder = USArrests$Murder,
state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))
```

- k + geom_map(aes(map_id = state), map = map) + expand_limits(x = map$long, y = map$lat) map_id, alpha, color, fill, linetype, size

Three Variables

```
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))
l <- ggplot(seals, aes(long, lat))
```

- l + geom_contour(aes(z = z)) x, y, z, alpha, colour, group, linetype, size, weight
- l + geom_raster(aes(fill = z), hjust=0.5, vjust=0.5, interpolate=FALSE) x, y, alpha, fill
- l + geom_tile(aes(fill = z)) x, y, alpha, color, fill, linetype, size, width

Learn more at docs.ggplot2.org · ggplot2 2.0.0 · Updated: 12/15