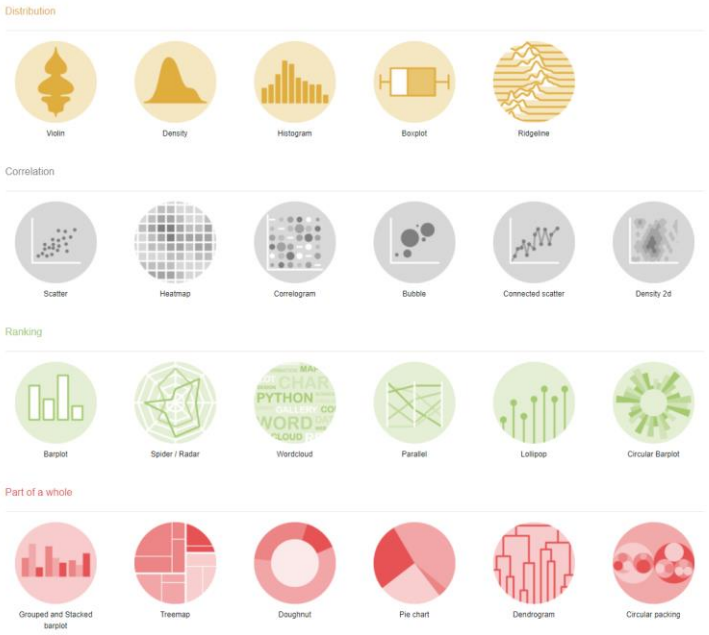


R Graph: ggplot2 Part 2

Pilsung Kang
School of Industrial Management Engineering
Korea University

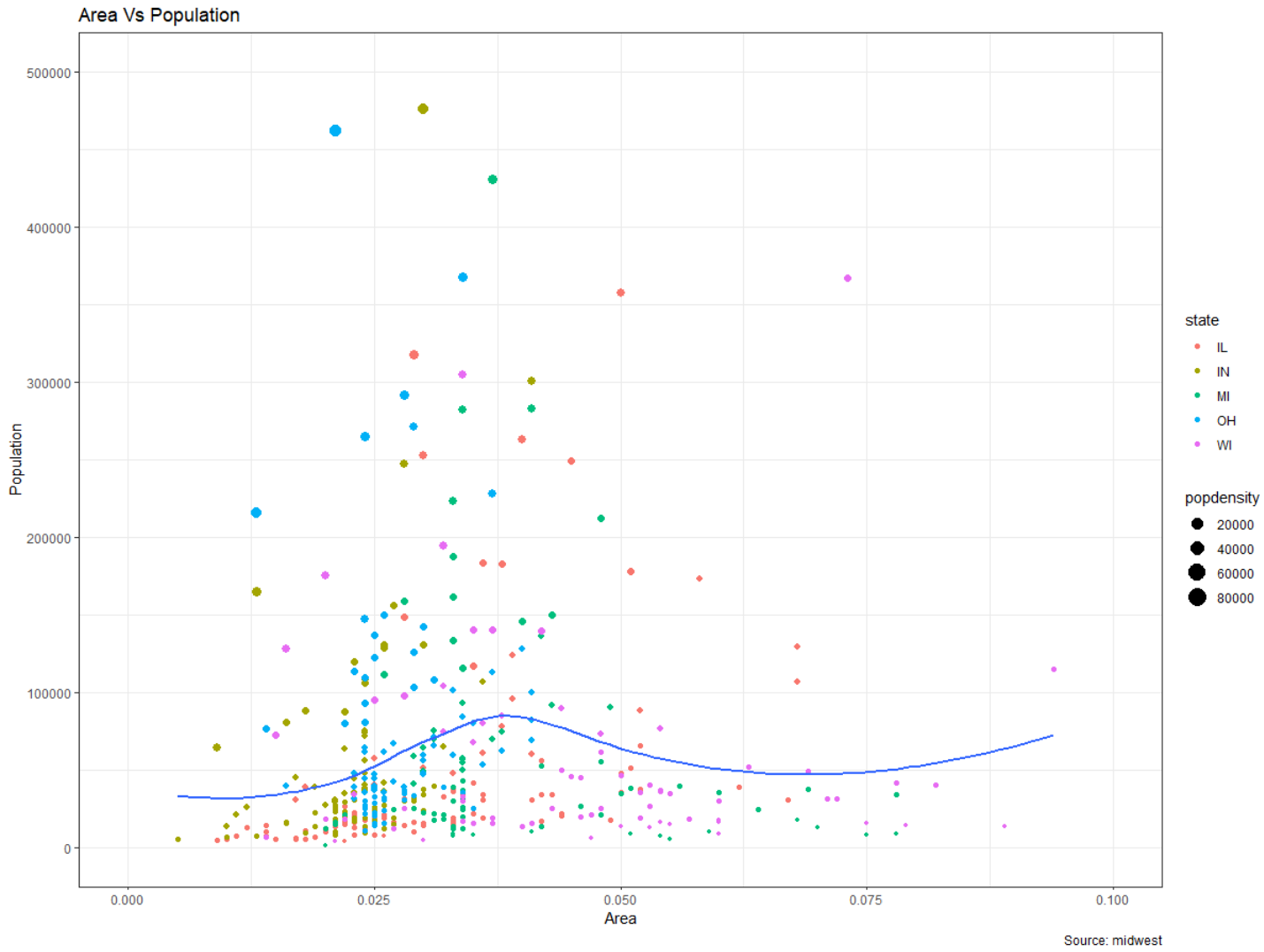
Part 2: Customize the look and feel

- Setup

```
# Setup
options(scipen=999)
data("midwest", package = "ggplot2")
theme_set(theme_bw())

# Add plot components ---------------------------------
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
        geom_point(aes(col=state, size=popdensity)) +
        geom_smooth(method="loess", se=F) +
        xlim(c(0, 0.1)) + ylim(c(0, 500000)) +
        labs(title="Area Vs Population",
            y="Population", x="Area", caption="Source: midwest")

# Call plot ------------------------------------------
plot(gg)
```

Part 2: Customize the look and feel

- Setup



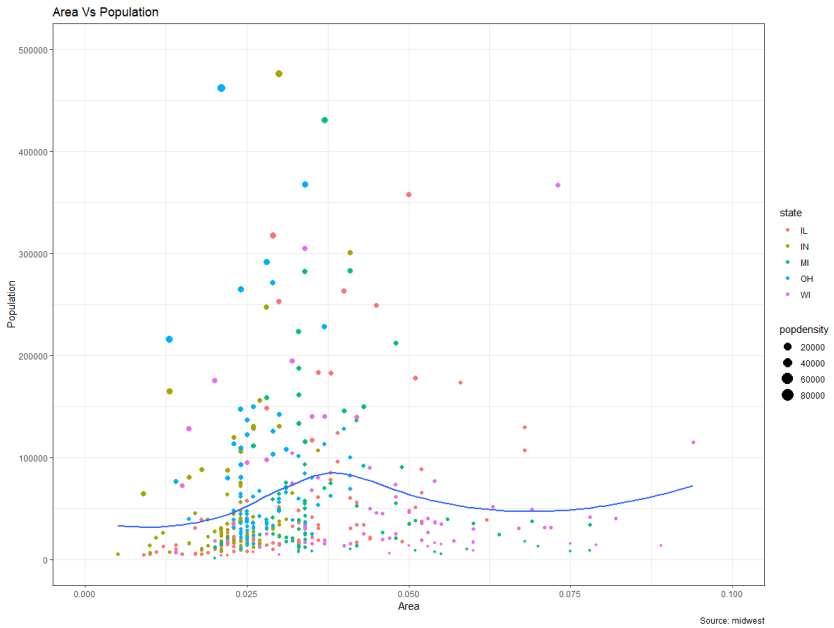
Part 2: Customize the look and feel

- The arguments passed to `theme()` components require to be set using special `element_type( )` functions. They are of 4 major types.
  - ✓ `element_text()`: Since the title, subtitle and captions are textual items, `element_text()` function is used to set it.
  - ✓ `element_line()`: Likewise, `element_line()` is used to modify line-based components such as the axis lines, major and minor grid lines, etc.
  - ✓ `element_rect()`: Modifies rectangle components such as plot and panel background.
  - ✓ `element_blank()`: Turns off displaying the theme item.

1. Adding plot and axis titles

- `theme()` for modifying plot and axis titles

```
# Base Plot
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
        geom_point(aes(col=state, size=popdensity)) +
        geom_smooth(method="loess", se=F) +
        xlim(c(0, 0.1)) + ylim(c(0, 500000)) +
        labs(title="Area Vs Population", y="Population", x="Area",
            caption="Source: midwest")
plot(gg)
```



1. Adding plot and axis titles

- `theme()` for modifying plot and axis titles

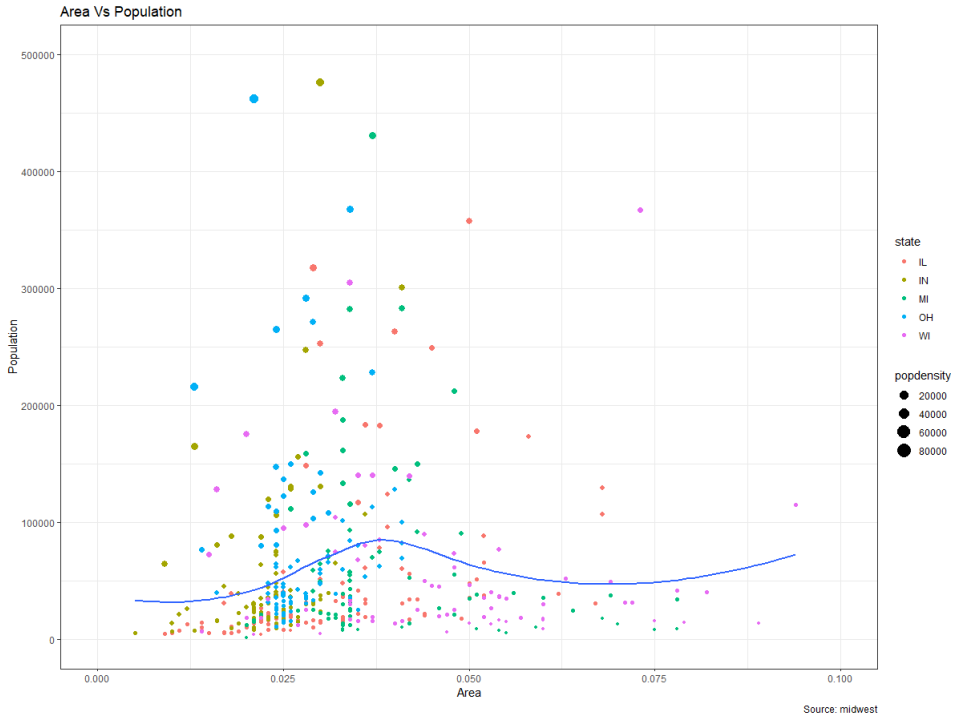
```
# Modify theme components -----------------------------------------
gg + theme(plot.title=element_text(size=20, face="bold", color="tomato",
                                   hjust=0.5, lineheight=1.2), # title
           plot.subtitle=element_text(size=15, face="bold",
                                   hjust=0.5), # subtitle
           plot.caption=element_text(size=15), # caption
           axis.title.x=element_text(vjust=10, size=15), # X axis title
           axis.title.y=element_text(size=15), # Y axis title
           axis.text.x=element_text(size=10, angle = 30,
           vjust=.5), # X axis text
           axis.text.y=element_text(size=10)) # Y axis text
```

- ✓ `vjust`, controls the vertical spacing between title (or label) and plot.
- ✓ `hjust`, controls the horizontal spacing. Setting it to 0.5 centers the title.
- ✓ `family`, is used to set a new font
- ✓ `face`, sets the font face ("plain", "italic", "bold", "bold.italic")

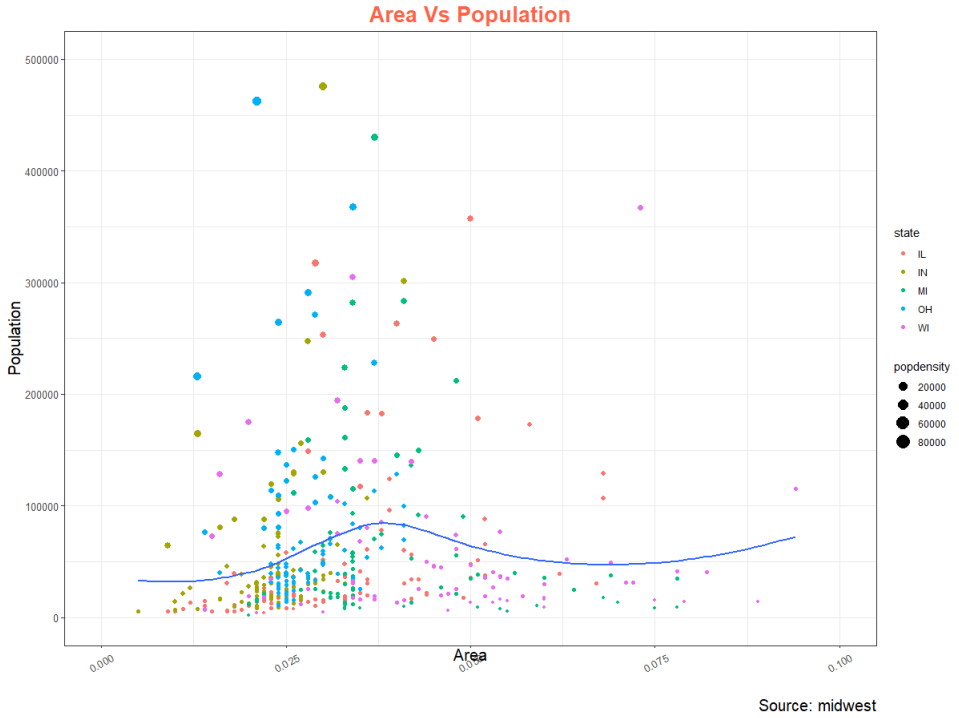
1. Adding plot and axis titles

- `theme()` for modifying plot and axis titles

Original



Modified



2. Modifying Legend

- Whenever your plot's geom (like points, lines, bars, etc) is set to change the aesthetics (`fill`, `size`, `col`, `shape`, or `stroke`) based on another column, as in `geom_point(aes(col=state, size=popdensity))`, a legend is automatically drawn.

2. Modifying Legend

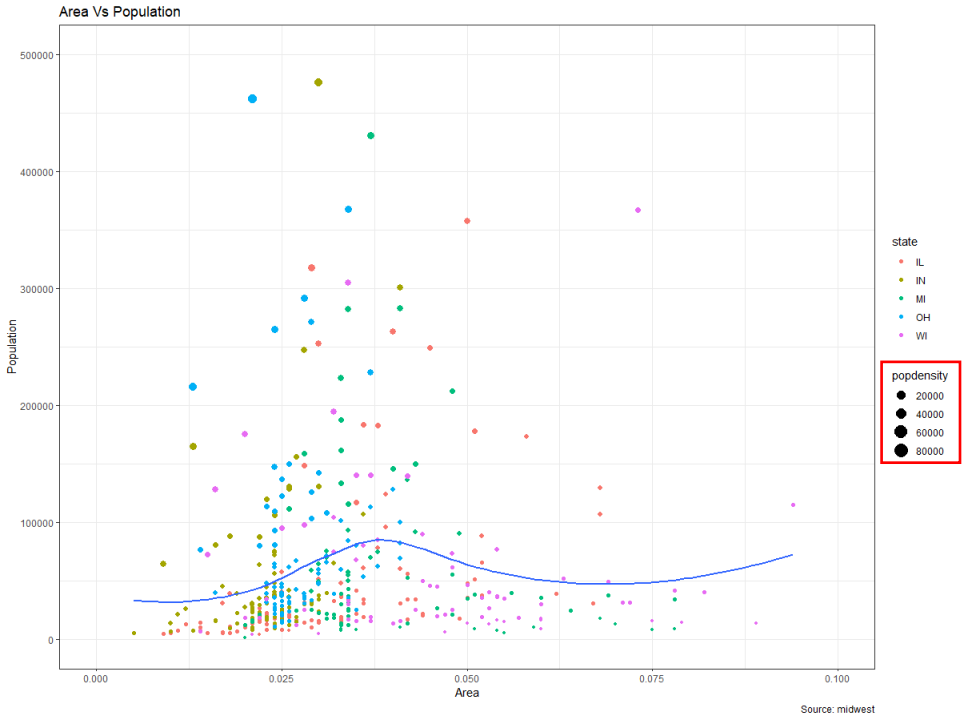
- Change the legend title
  - ✓ Method 1: Using `labs()`

```
# Method 1: using labs()
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
        geom_point(aes(col=state, size=popdensity)) +
        geom_smooth(method="loess", se=F) +
        xlim(c(0, 0.1)) + ylim(c(0, 500000)) +
        labs(title="Area Vs Population", y="Population", x="Area",
                caption="Source: midwest")
plot(gg)

gg + labs(color="State", size="Density") # modify legend title
```

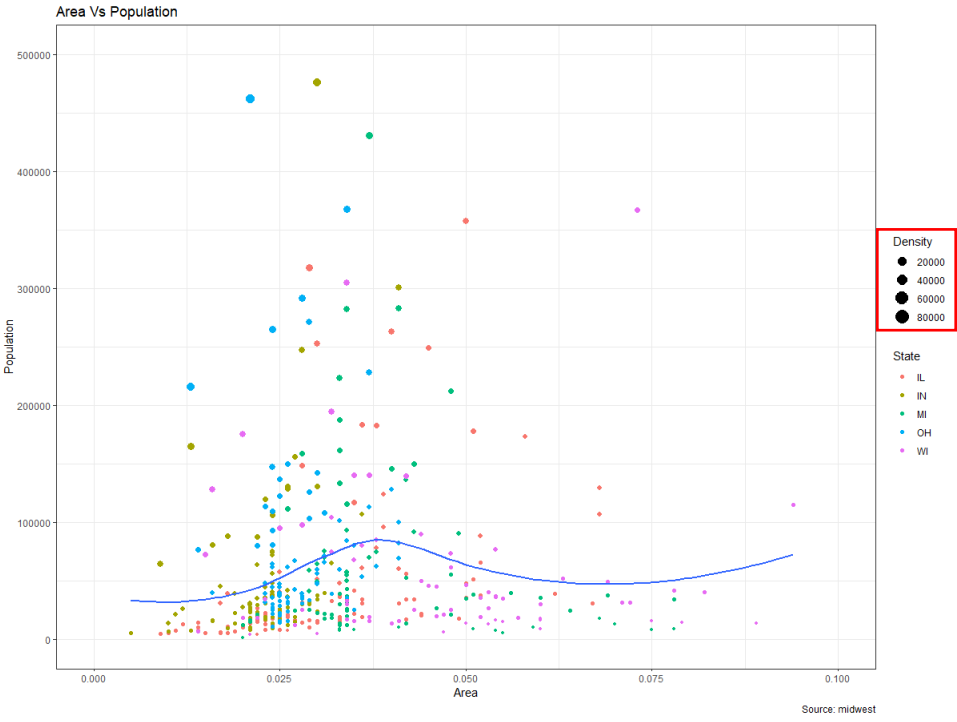
2. Modifying Legend

- Change the legend title
  - ✓ Method 1: Using `labs()`

Original



Modified



2. Modifying Legend

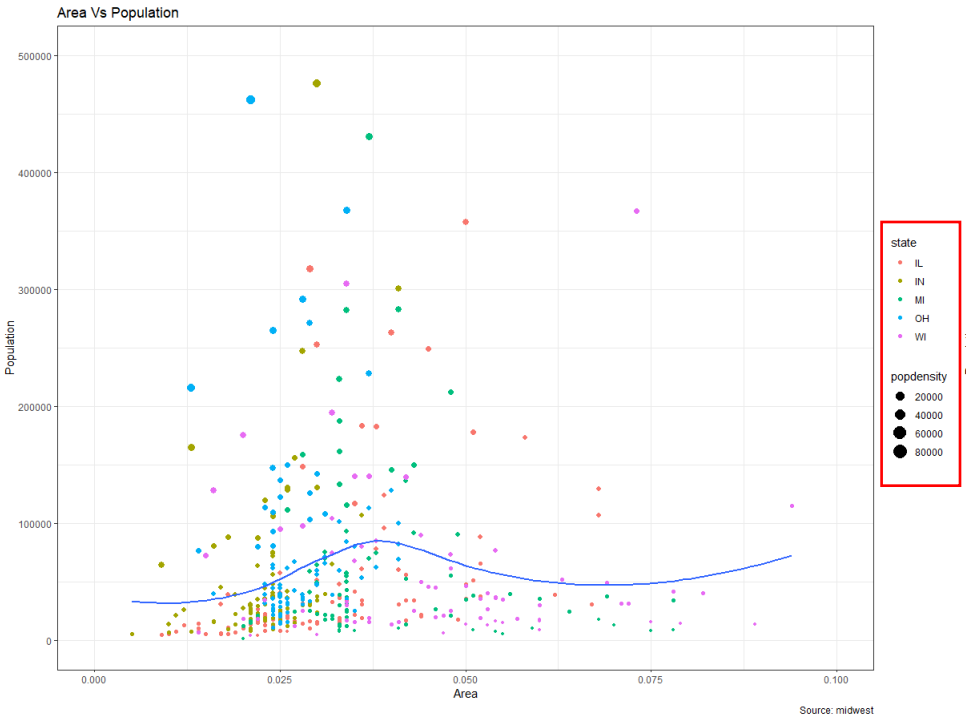
- Change the legend title
  - ✓ Method 2: Using `guides()`

```
# Method 2: using guides()
# Base Plot
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
        geom_point(aes(col=state, size=popdensity)) +
        geom_smooth(method="loess", se=F) +
        xlim(c(0, 0.1)) + ylim(c(0, 500000)) +
        labs(title="Area Vs Population", y="Population", x="Area",
            caption="Source: midwest")

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

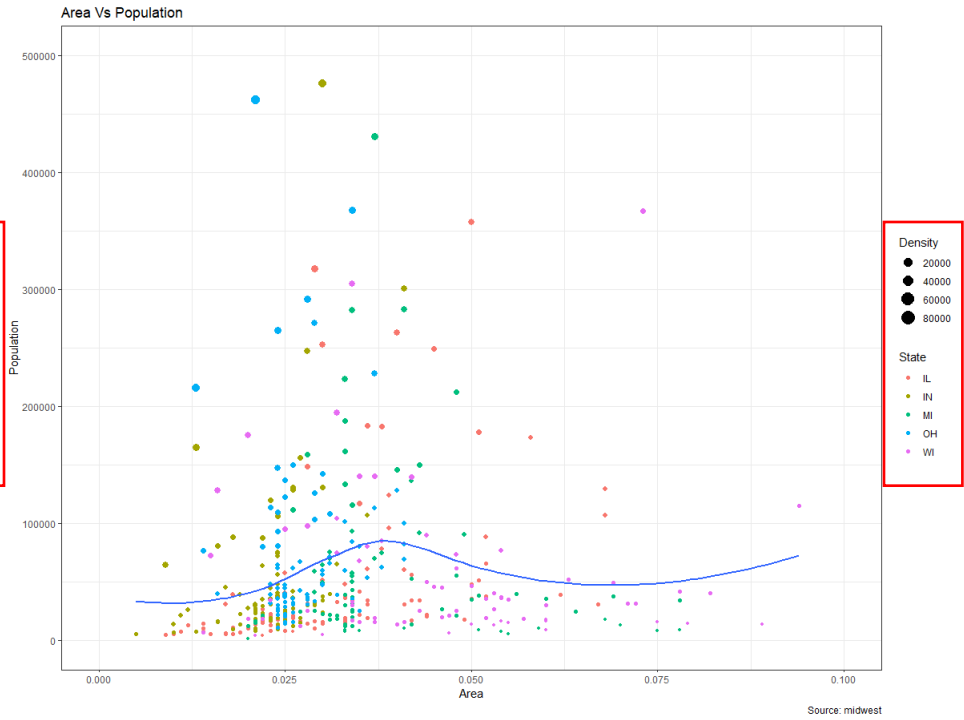
2. Modifying Legend

- Change the legend title
  - ✓ Method 2: Using `guides()`

Original



Modified



2. Modifying Legend

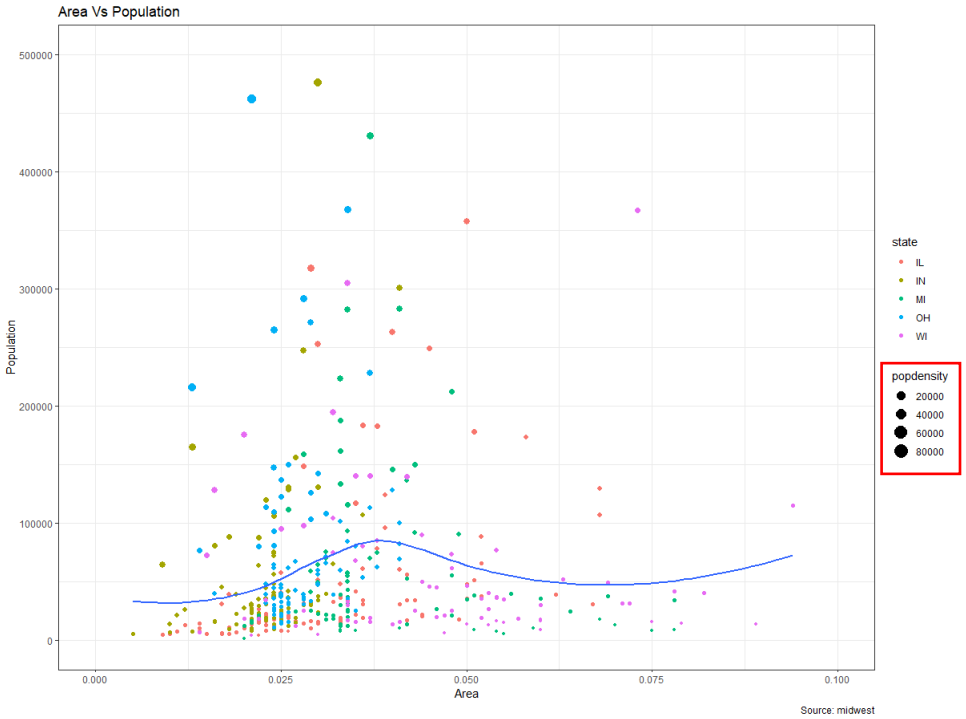
- Change the legend title
  - ✓ Method 3: Using `scale_aesthetic_vartype()`

```
# Method 3: Using scale_aesthetic_vartype() format
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
        geom_point(aes(col=state, size=popdensity)) +
        geom_smooth(method="loess", se=F) +
        xlim(c(0, 0.1)) + ylim(c(0, 500000)) +
        labs(title="Area Vs Population", y="Population", x="Area",
            caption="Source: midwest")
plot(gg)

# Modify Legend
gg + scale_color_discrete(name="State") +
        scale_size_continuous(name = "Density", guide = FALSE)
# turn off legend for size
```

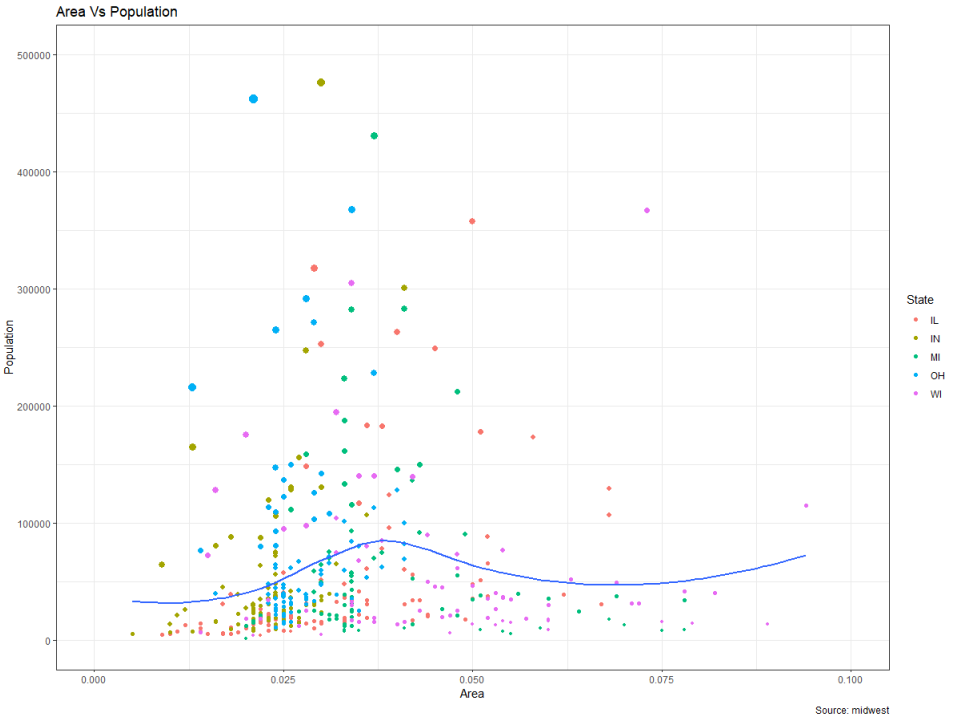
2. Modifying Legend

- Change the legend title
  - ✓ Method 3: Using `scale_aesthetic_vartype()`

Original



Modified



2. Modifying Legend

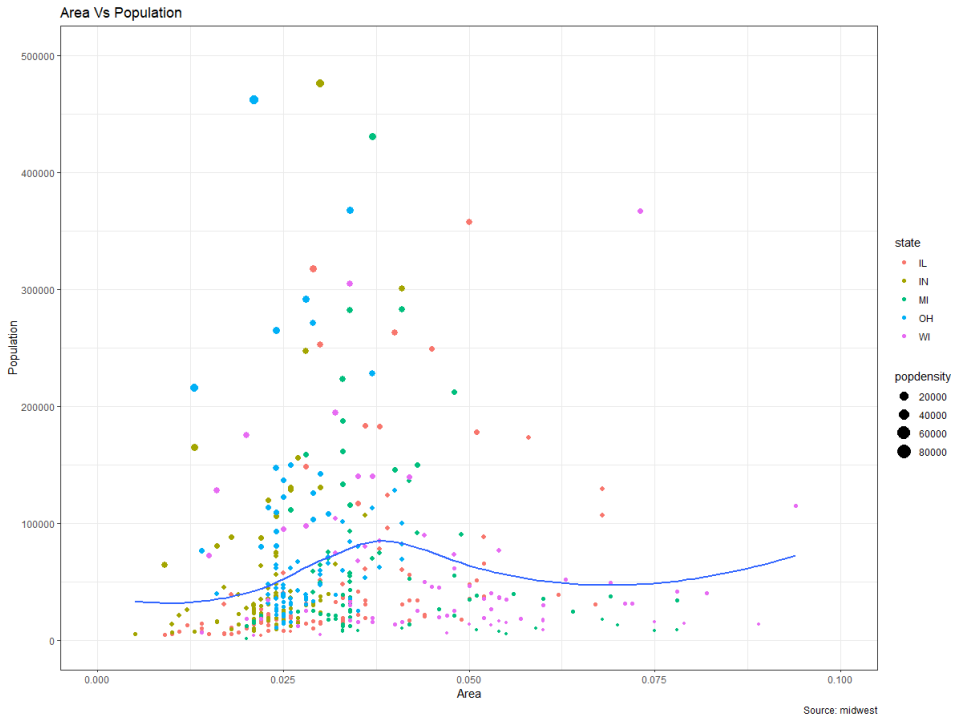
- Change Legend Labels and Point Colors for Categories
  - ✓ Can be done using the respective `scale_aesthetic_manual()` function
  - ✓ The new legend labels are supplied as a character vector to the `labels` argument
  - ✓ If you want to change the color of the categories, it can be assigned to the `values` argument as shown in below example.

```
gg + scale_color_manual(name="State",
                        labels = c("Illinois", "Indiana", "Michigan",
                                   "Ohio", "Wisconsin"),
                        values = c("IL"="blue", "IN"="red", "MI"="green",
                                   "OH"="brown", "WI"="orange"))
```

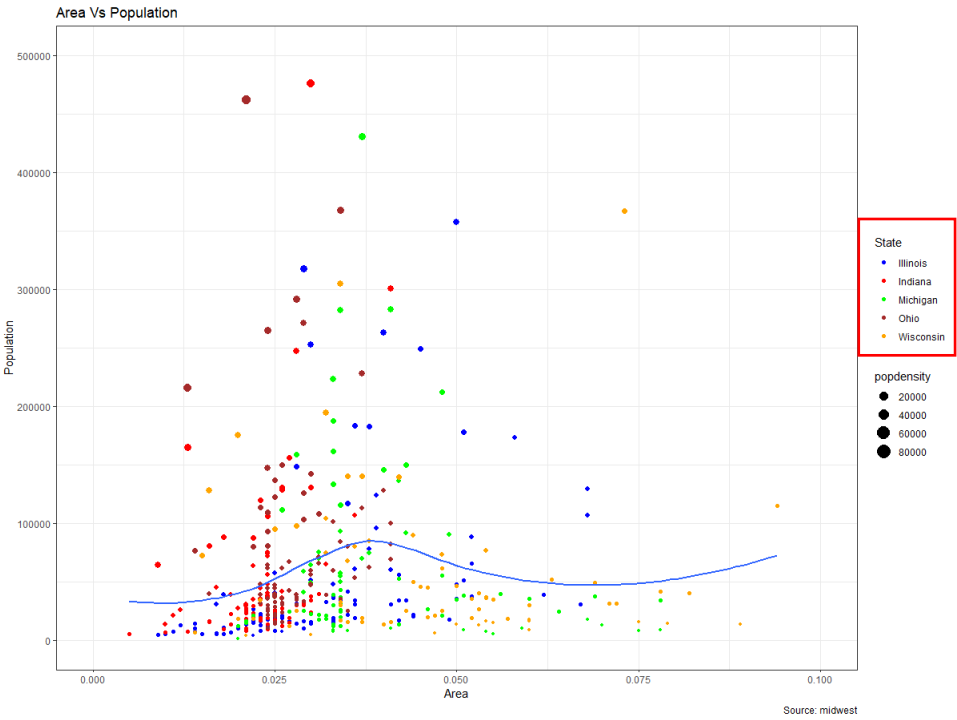
2. Modifying Legend

- Change Legend Labels and Point Colors for Categories

Original



Modified



2. Modifying Legend

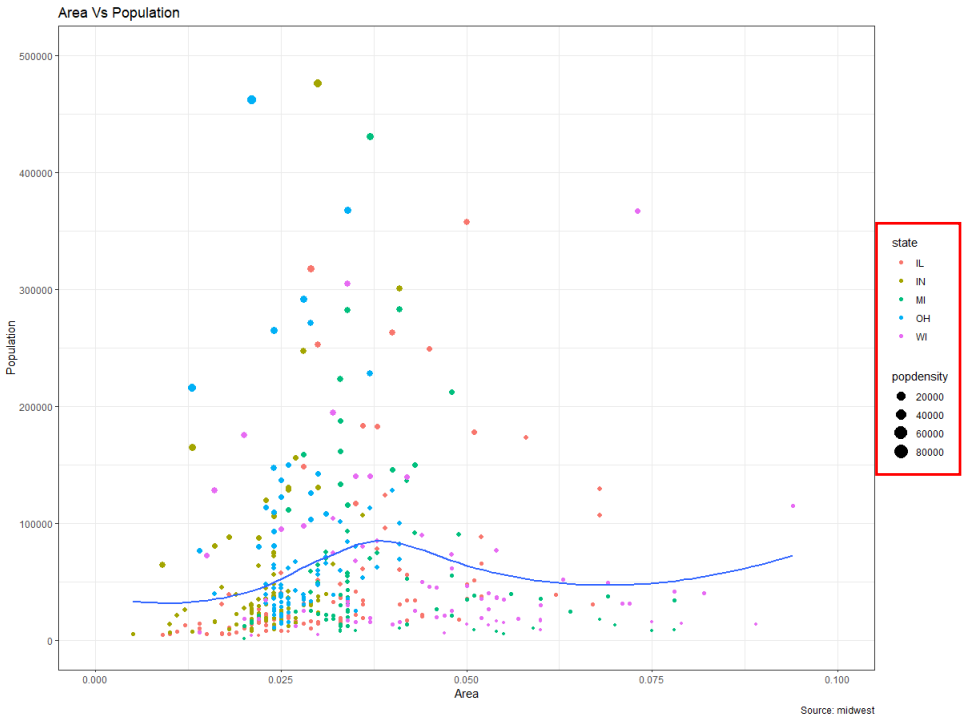
- Change the order of legend
  - ✓ In case you want to show the legend for color (State) before size (Density), it can be done with the `guides()` function: the order of the legend has to be set as desired.

```
# Change the order of legend
# Base Plot
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
        geom_point(aes(col=state, size=popdensity)) +
        geom_smooth(method="loess", se=F) +
        xlim(c(0, 0.1)) + ylim(c(0, 500000)) +
        labs(title="Area Vs Population", y="Population", x="Area",
        caption="Source: midwest")
plot(gg)

gg + guides(colour = guide_legend(order = 2), size = guide_legend(order = 1))
```

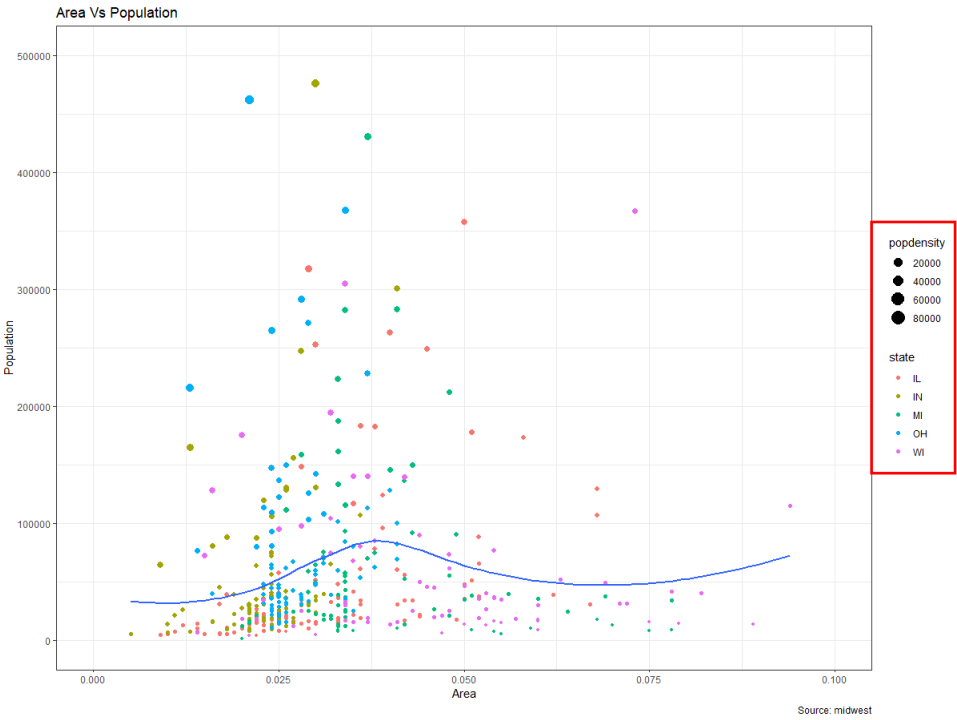
2. Modifying Legend

- Change the order of legend

Original



Modified



2. Modifying Legend

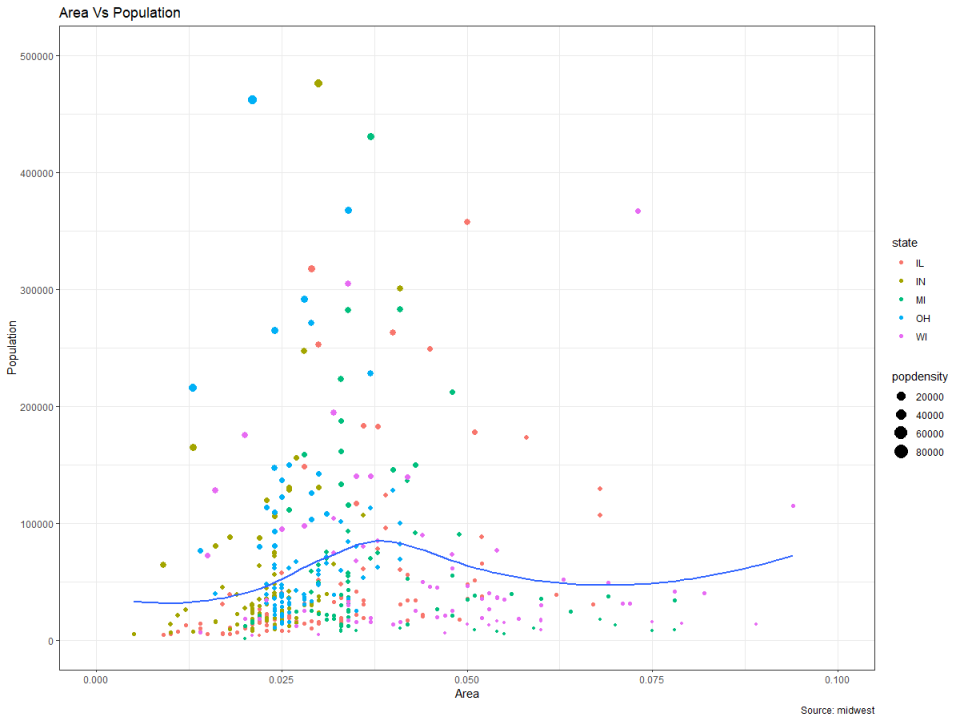
- Style the legend title, text, and key
  - ✓ The styling of legend title, text, key, and the guide can also be adjusted
  - ✓ The legend's key is a figure like element, so it has to be set using `element_rect()`

```
# Style the legend title, text, and key
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
        geom_point(aes(col=state, size=popdensity)) +
        geom_smooth(method="loess", se=F) +
        xlim(c(0, 0.1)) + ylim(c(0, 500000)) +
        labs(title="Area Vs Population", y="Population", x="Area",
            caption="Source: midwest")
plot(gg)

gg + theme(legend.title = element_text(size=12, color = "firebrick"),
            legend.text = element_text(size=10),
            legend.key=element_rect(fill='springgreen')) +
    guides(colour = guide_legend(override.aes = list(size=2, stroke=1.5)))
```

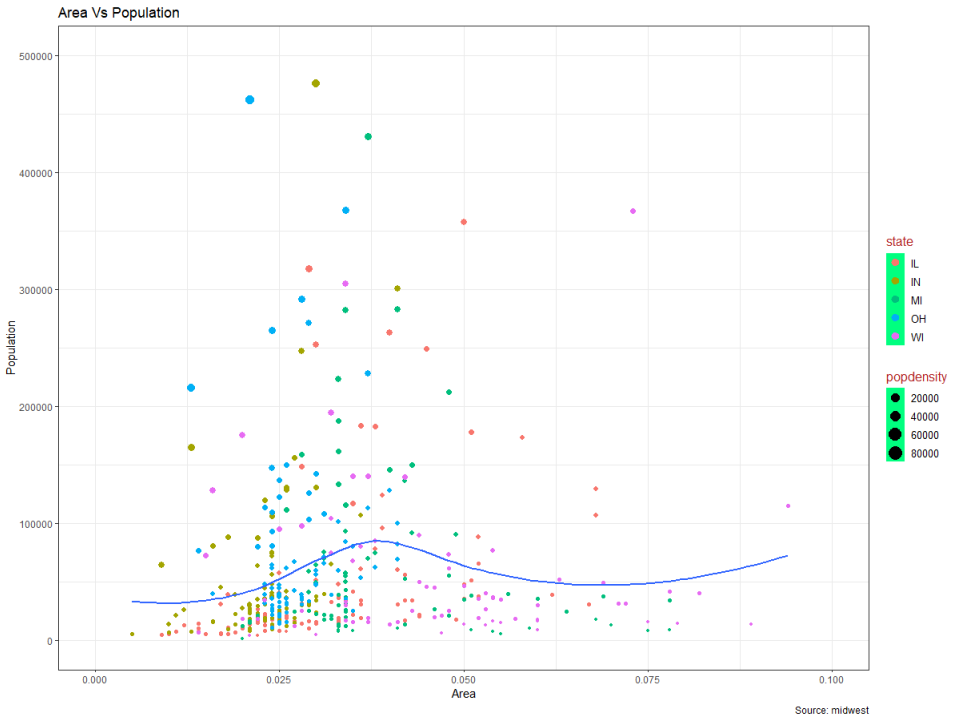
2. Modifying Legend

- Style the legend title, text, and key

Original



Modified



2. Modifying Legend

- Remove the Legend and Change Legend Positions
  - ✓ The legend's position inside the plot is an aspect of the theme. So it can be modified using the `theme()` function. If you want to place the legend inside the plot, you can additionally control the hinge point of the legend using `legend.justification`.
  - ✓ The legend.position is the x and y axis position in chart area, where `(0,0)` is bottom left of the chart and `(1,1)` is top right. Likewise, `legend.justification` refers to the hinge point inside the legend.

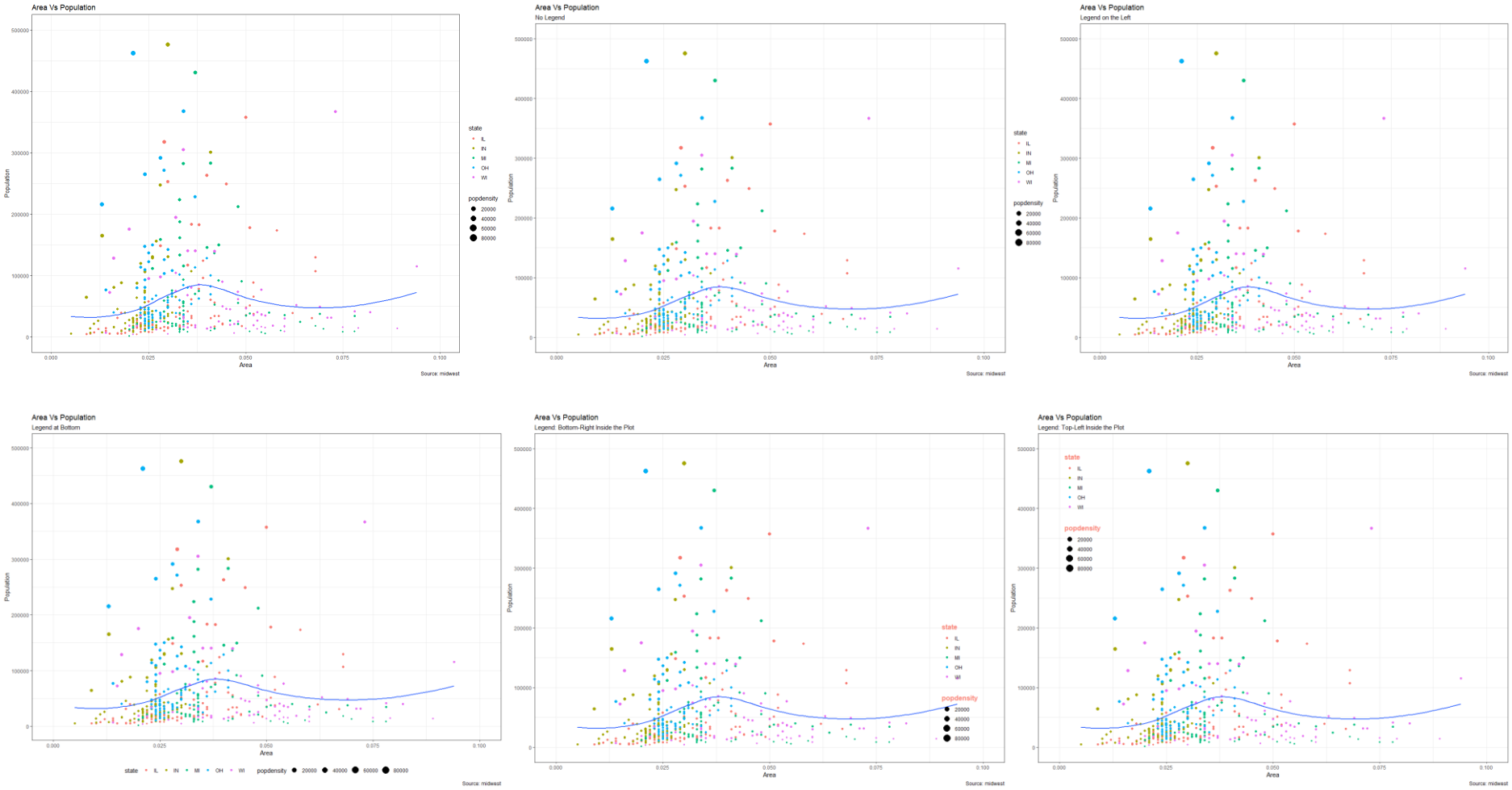
2. Modifying Legend

- Remove the Legend and Change Legend Positions

```
# No legend ------------------------------------------------
gg + theme(legend.position="None") + labs(subtitle="No Legend")

# Legend to the left ---------------------------------------
gg + theme(legend.position="left") + labs(subtitle="Legend on the Left")

# legend at the bottom and horizontal ----------------------
gg + theme(legend.position="bottom", legend.box = "horizontal") +
          labs(subtitle="Legend at Bottom")

# legend at bottom-right, inside the plot ------------------
gg + theme(legend.title = element_text(size=12, color = "salmon", face="bold"),
               legend.justification=c(1,0),
               legend.position=c(0.95, 0.05),
               legend.background = element_blank(),
               legend.key = element_blank()) +
          labs(subtitle="Legend: Bottom-Right Inside the Plot")

# legend at top-left, inside the plot ----------------------
gg + theme(legend.title = element_text(size=12, color = "salmon", face="bold"),
               legend.justification=c(0,1),
               legend.position=c(0.05, 0.95),
               legend.background = element_blank(),
               legend.key = element_blank()) +
     labs(subtitle="Legend: Top-Left Inside the Plot")
```

2. Modifying Legend

- Remove the Legend and Change Legend Positions



3. Adding text, label, and annotation

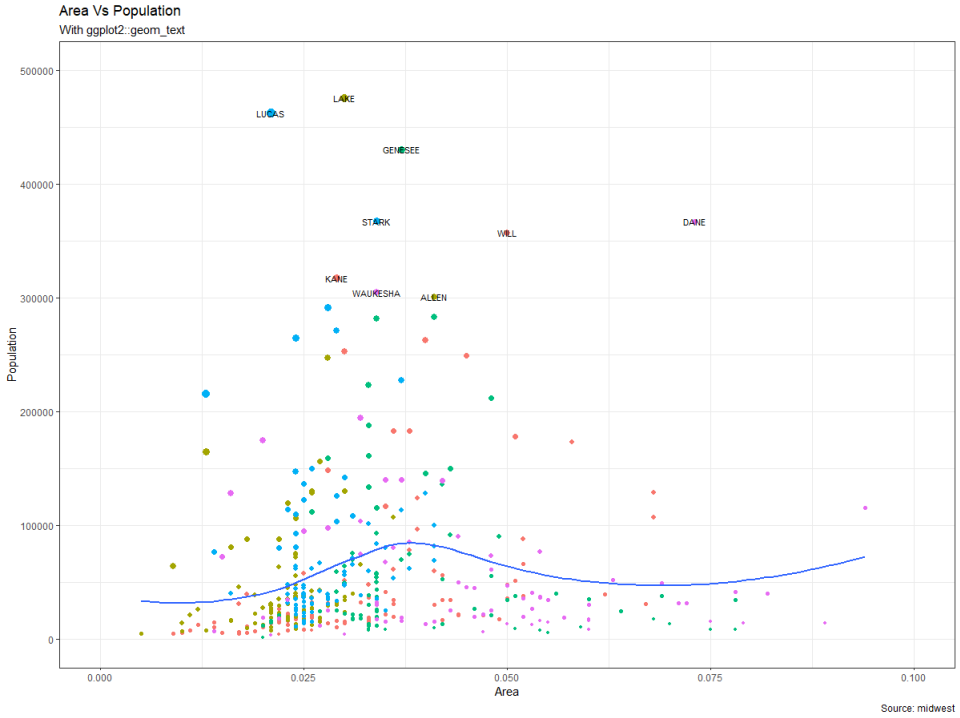
- Add text and label around the points
  - ✓ Use geom_text and geom_label

```
# Add text and label around the points
# Filter required rows.
midwest_sub <- midwest[midwest$poptotal > 300000, ]
midwest_sub$large_county <- ifelse(midwest_sub$poptotal > 300000,
                                   midwest_sub$county, "")
# Base Plot
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
          geom_point(aes(col=state, size=popdensity)) +
          geom_smooth(method="loess", se=F) +
          xlim(c(0, 0.1)) + ylim(c(0, 500000)) +
          labs(title="Area Vs Population", y="Population", x="Area",
          caption="Source: midwest")
plot(gg)

# Plot text and label ------------------------------------------------------
gg + geom_text(aes(label=large_county), size=3, data=midwest_sub) +
          labs(subtitle="With ggplot2::geom_text") +
          theme(legend.position = "None")

gg + geom_label(aes(label=large_county), size=3, data=midwest_sub, alpha=0.25) +
          labs(subtitle="With ggplot2::geom_label") +
          theme(legend.position = "None")
```

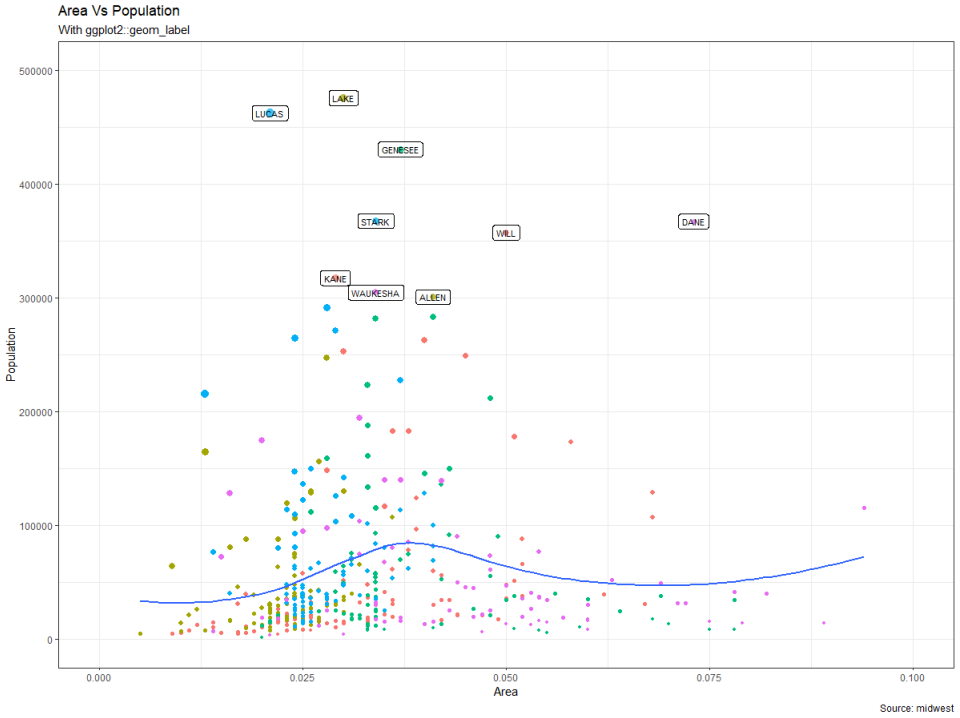
3. Adding text, label, and annotation

- Add text and label around the points
  - ✓ Use `geom_text` and `geom_label`

`geom_text`



`geom_label`



3. Adding text, label, and annotation

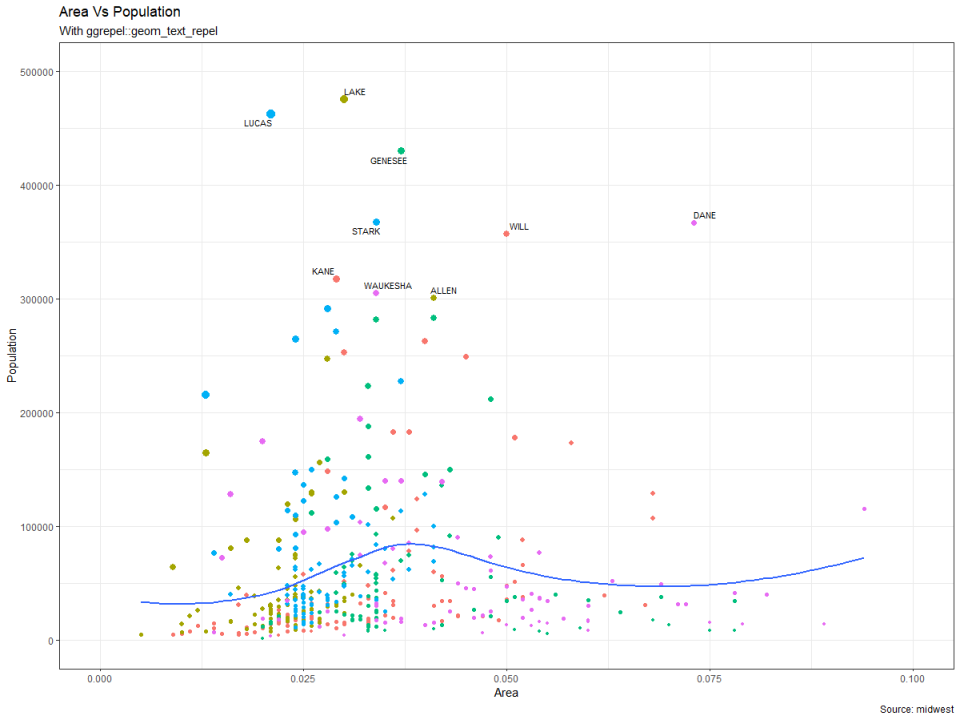
- Add text and label around the points
  - ✓ With "ggrepel" package

```
# Plot text and label that REPELS eachother (using ggrepel pkg) ------------
install.packages("ggrepel")
library(ggrepel)

gg + geom_text_repel(aes(label=large_county), size=3, data=midwest_sub) +
        labs(subtitle="With ggrepel::geom_text_repel") +
        theme(legend.position = "None")

gg + geom_label_repel(aes(label=large_county), size=3, data=midwest_sub) +
        labs(subtitle="With ggrepel::geom_label_repel") +
        theme(legend.position = "None")
```

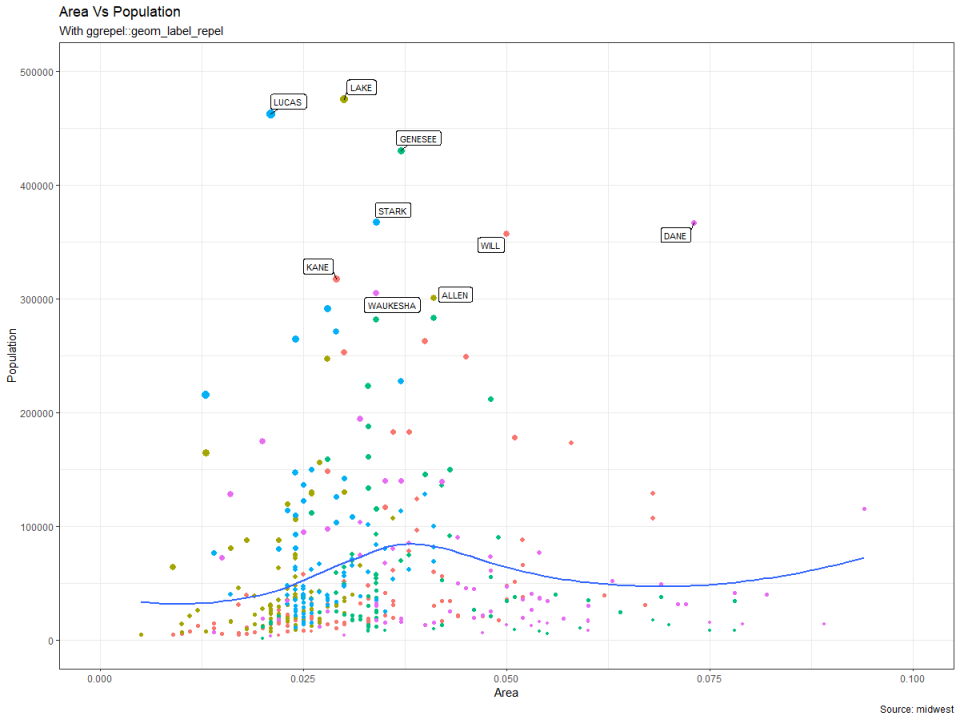
3. Adding text, label, and annotation

- Add text and label around the points
  - ✓ With "ggrepel" package

`geom_text_repel`



`geom_label_repel`



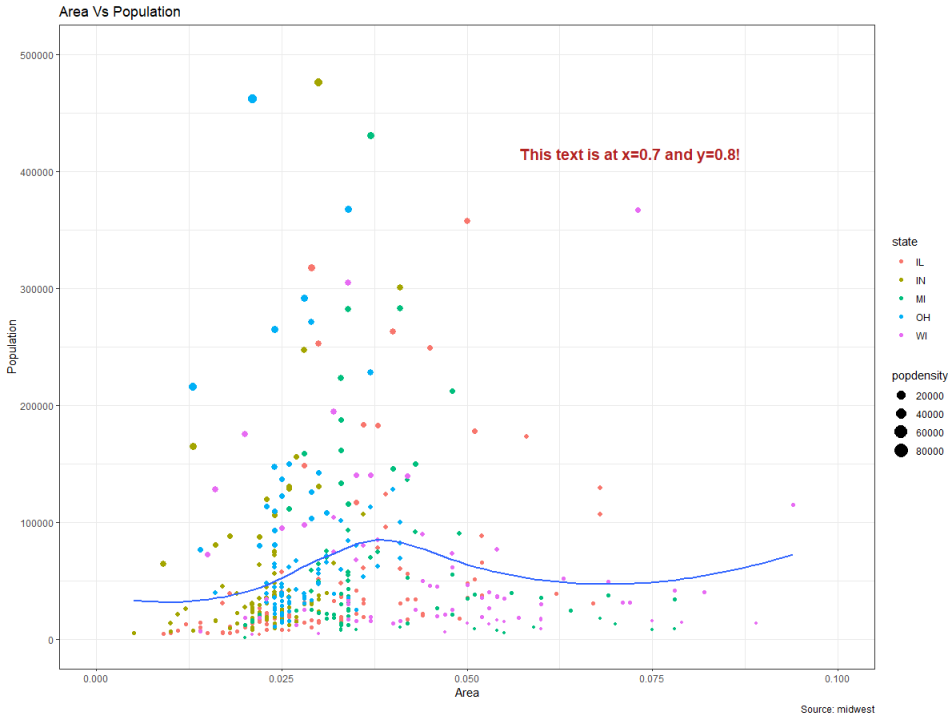
3. Adding text, label, and annotation

- Add annotations anywhere inside plot
  - ✓ Use `annotation_custom()` with `grob` as the argument

```
# Define and add annotation ------------------------------------
library(grid)
my_text <- "This text is at x=0.7 and y=0.8!"
my_grob = grid.text(my_text, x=0.7, y=0.8,
                    gp=gpar(col="firebrick", fontsize=14, fontface="bold"))

gg + annotation_custom(my_grob)
```

3. Adding text, label, and annotation

- Add annotations anywhere inside plot
  - ✓ Use `annotation_custom()` with `grob` as the argument



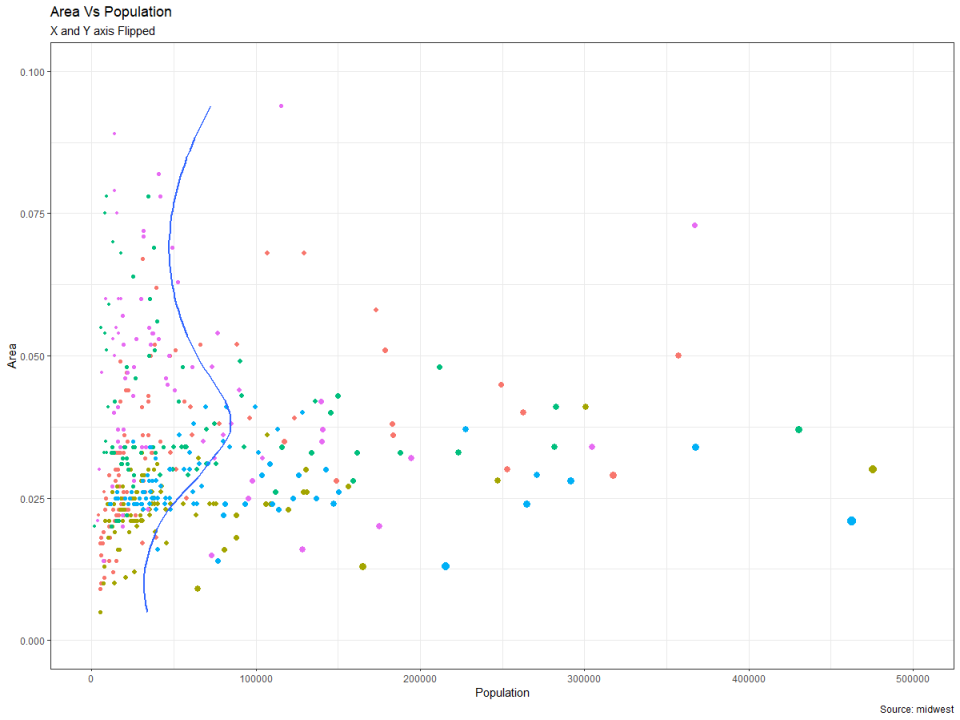
4. Flipping and Reversing X and Y Axis

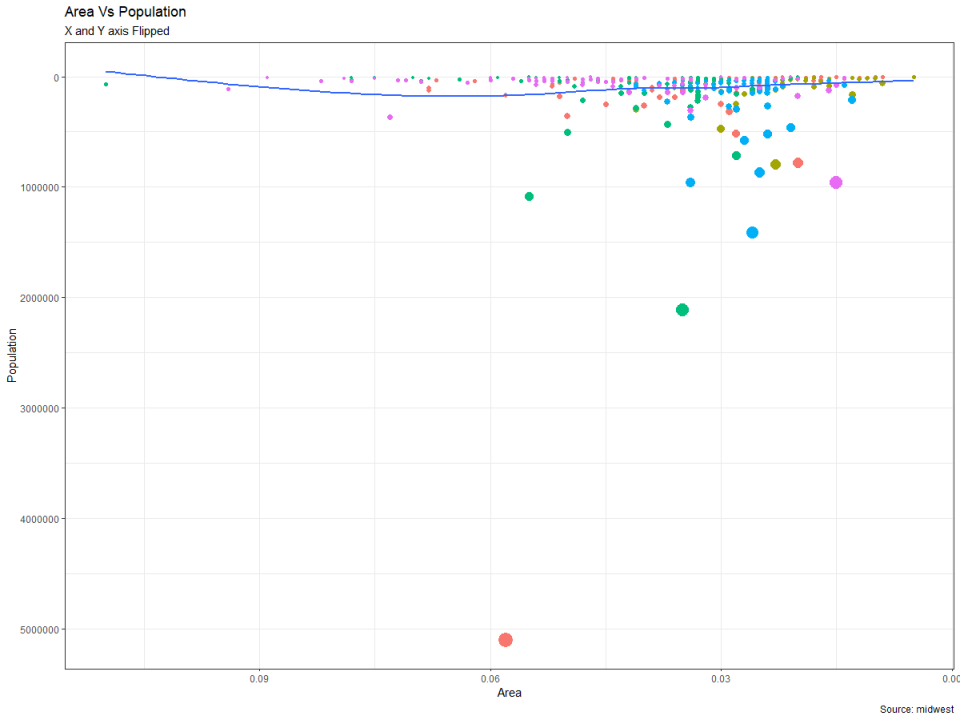
- How to flip the X and Y axis?
  - ✓ Just add `coord_flip()`
- How to reverse the scale of an axis?
  - ✓ Use `scale_x_reverse()` for X axis and `scale_reverse()` for Y axis

```
# Flip the X and Y axis -----------------------------
gg + coord_flip()

# Reverse the X and Y Axis --------------------------
gg + scale_x_reverse() + scale_y_reverse()
```

4. Flipping and Reversing X and Y Axis

- How to flip the X and Y axis?
  - ✓ Just add `coord_flip()`
- How to reverse the scale of an axis?
  - ✓ Use `scale_x_reverse()` for X axis and `scale_reverse()` for Y axis

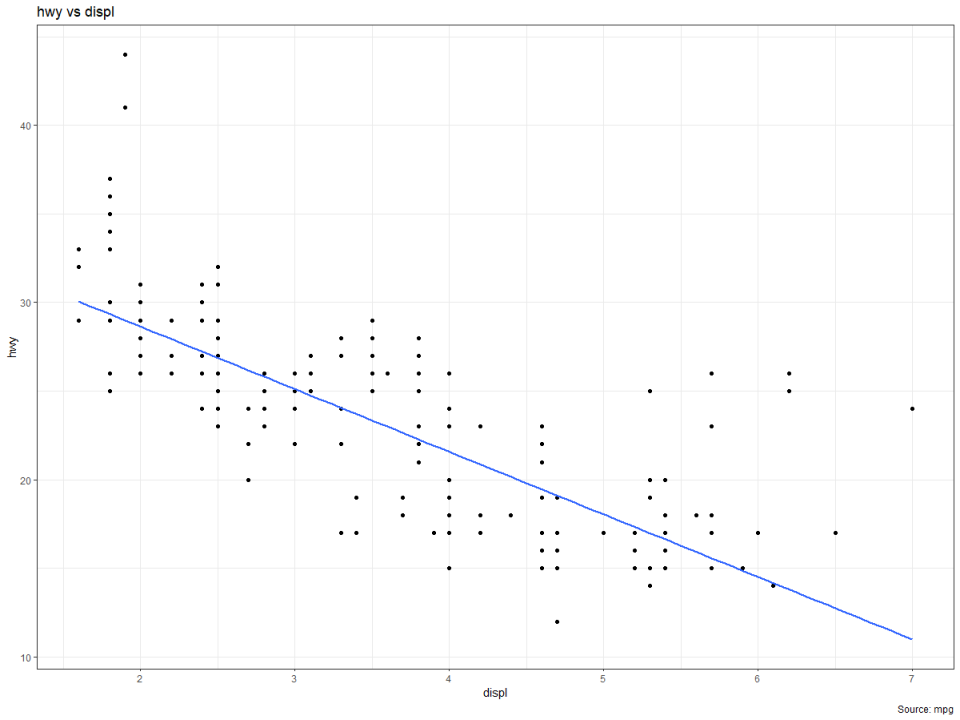




5. Faceting: Draw Multiple Plots within One Figure

- Dataset: mpg

```
data(mpg, package="ggplot2")
# load data # mpg <- read.csv("http://goo.gl/uEeRGu") # alt data source

g <- ggplot(mpg, aes(x=displ, y=hwy)) + geom_point() +
        labs(title="hwy vs displ", caption = "Source: mpg") +
        geom_smooth(method="lm", se=FALSE) + theme_bw() # apply bw theme

plot(g)
```



5. Faceting: Draw Multiple Plots within One Figure

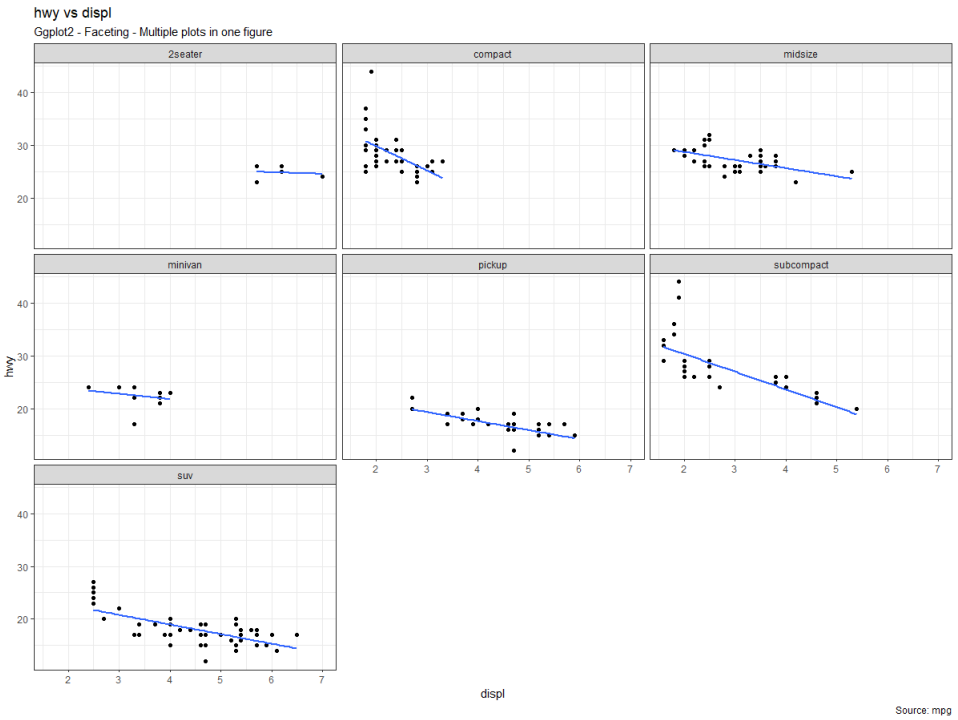
- Facet_wrap()
  - ✓ Break down a large plot into multiple small plots for individual categories

```
# Facet wrap with common scales
g + facet_wrap( ~ class, nrow=3) +
    labs(title="hwy vs displ", caption = "Source: mpg",
         subtitle="Ggplot2 - Faceting - Multiple plots in one figure")
# Shared scales

# Facet wrap with free scales
g + facet_wrap( ~ class, scales = "free") +
         labs(title="hwy vs displ", caption = "Source: mpg",
         subtitle="Ggplot2 - Faceting - Multiple plots in one figure with free
                        scales") # Scales free
```

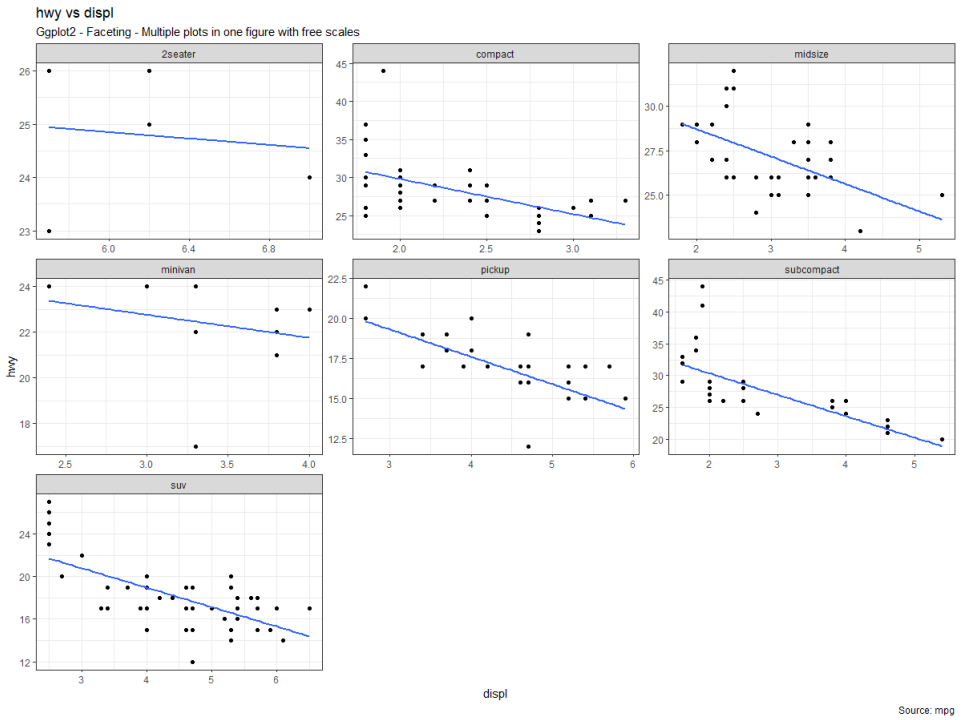
5. Faceting: Draw Multiple Plots within One Figure

- `Facet_wrap()`
  - ✓ Break down a large plot into multiple small plots for individual categories

Shared scale



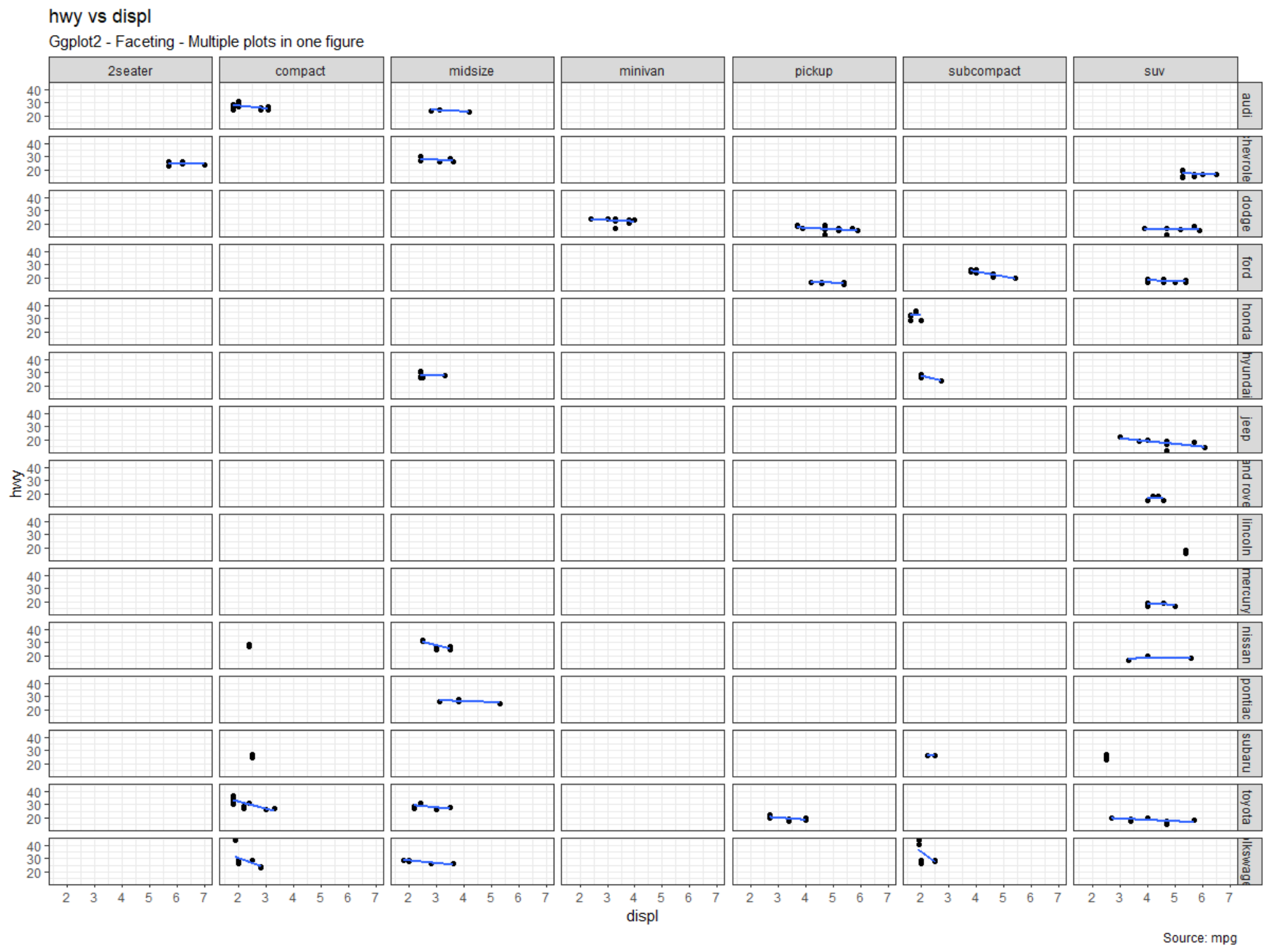
Different scale



5. Faceting: Draw Multiple Plots within One Figure

- `Facet_grid()`
  - ✓ The headings of the middle and bottom rows take up significant space.
  - ✓ The `facet_grid()` would get rid of it and give more area to the charts.
  - ✓ The main difference with `facet_grid` is that it is not possible to choose the number of rows and columns in the grid.

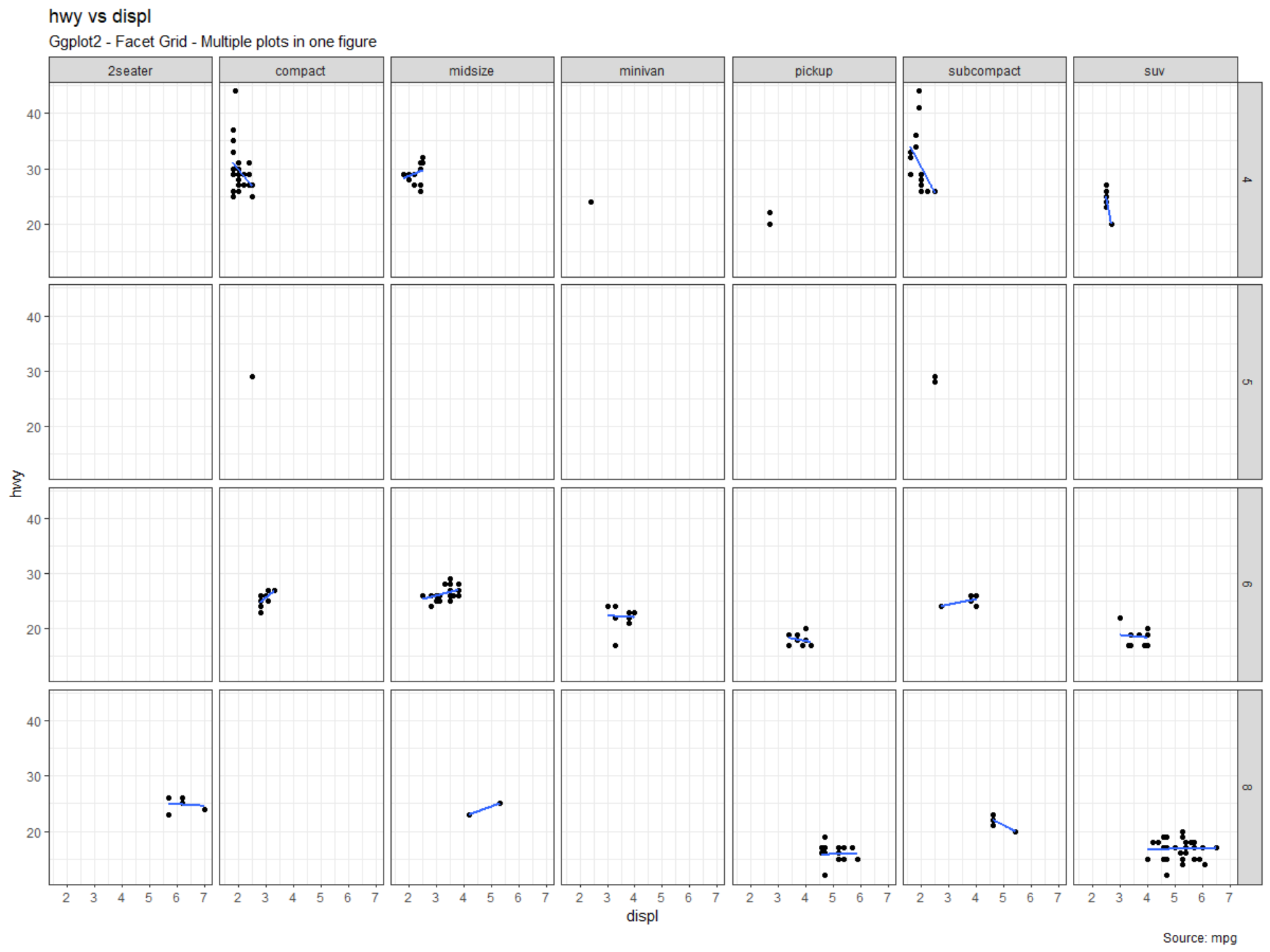
```
# Facet_grid()
# Base Plot
g <- ggplot(mpg, aes(x=displ, y=hwy)) + geom_point() +
        labs(title="hwy vs displ",
        caption = "Source: mpg",
        subtitle="Ggplot2 - Faceting - Multiple plots in one figure") +
        geom_smooth(method="lm", se=FALSE) + theme_bw() # apply bw theme
plot(g)

# Add Facet Grid
g1 <- g + facet_grid(manufacturer ~ class)
# manufacturer in rows and class in columns
plot(g1)
```

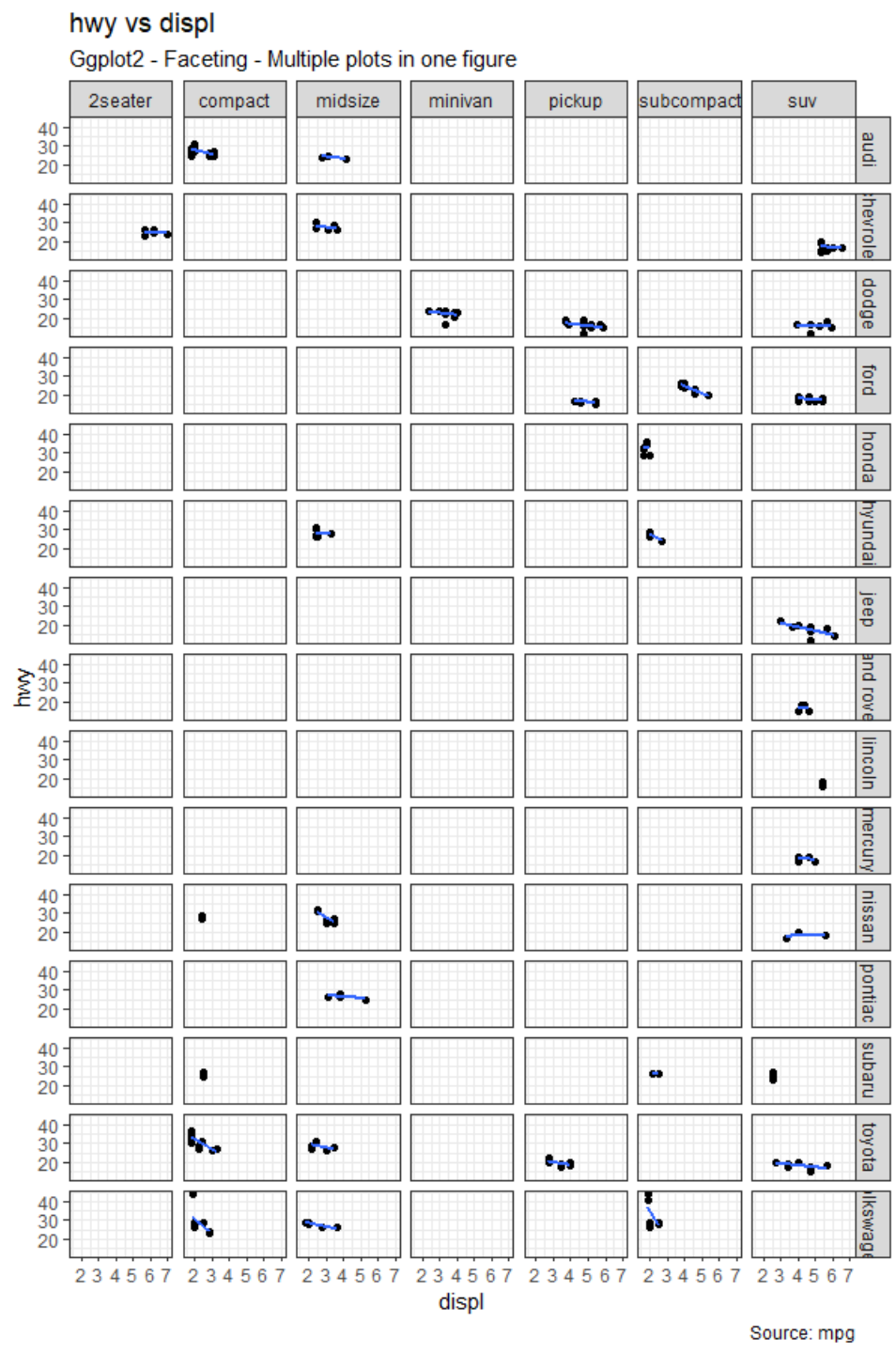

hwy vs displ
Ggplot2 - Faceting - Multiple plots in one figure
2seater
compact
midsize
minivan
pickup
subcompact
suv
audi
chevrole
dodge
ford
honda
hyundai
jeep
and rove
lincoln
mercury
nissan
pontiac
subaru
toyota
lkswage
hwy
displ
Source: mpg

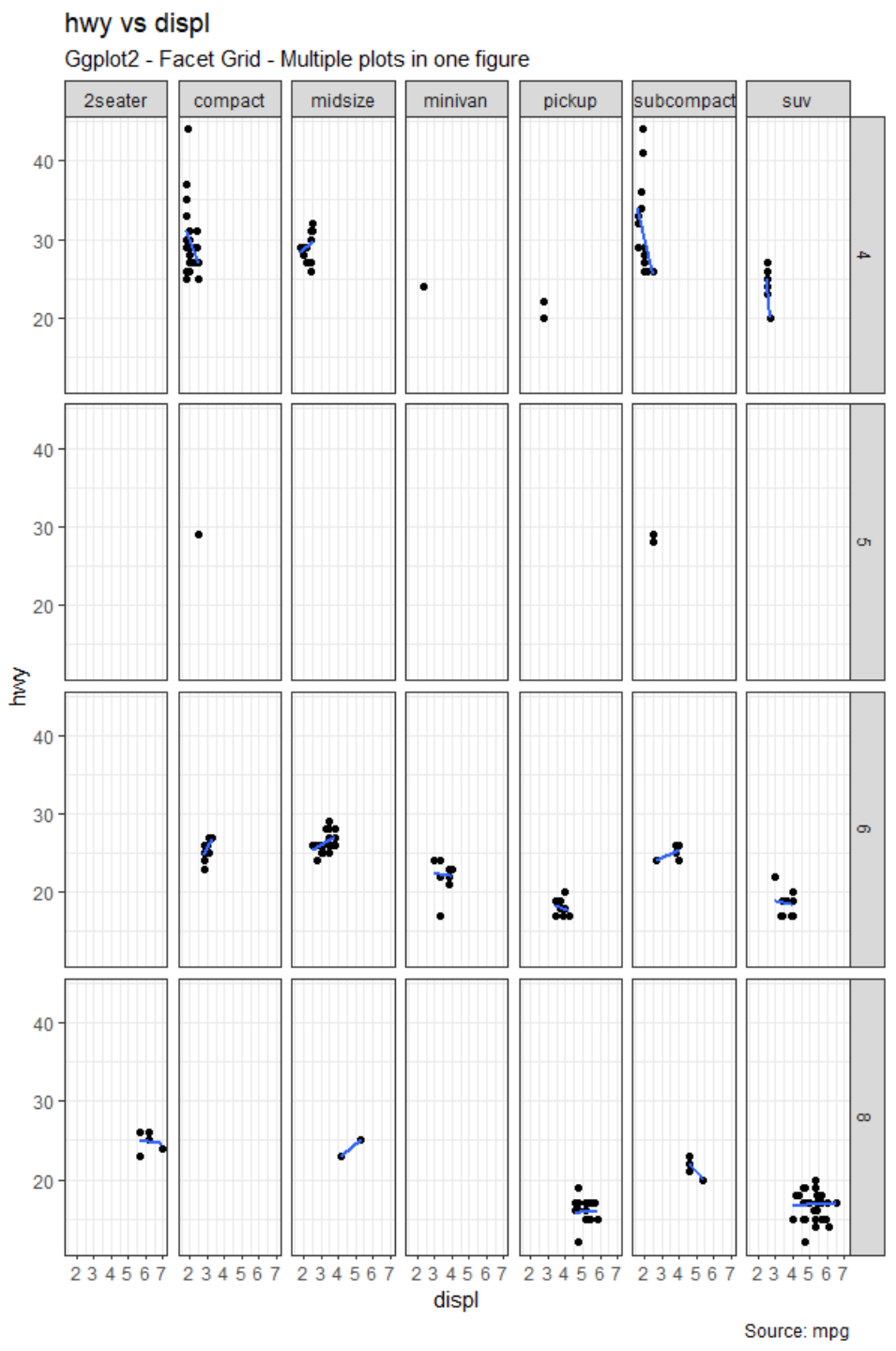
5. Faceting: Draw Multiple Plots within One Figure

- Facet_grid()
  - ✓ One more figure by varying cylinder

```
# Base Plot
g <- ggplot(mpg, aes(x=displ, y=hwy)) +
     geom_point() +
     geom_smooth(method="lm", se=FALSE) +
     labs(title="hwy vs displ", caption = "Source: mpg",
              subtitle="Ggplot2 - Facet Grid - Multiple plots in one figure") +
theme_bw() # apply bw theme

# Add Facet Grid
g2 <- g + facet_grid(cyl ~ class) # cyl in rows and class in columns.
plot(g2)

# Draw Multiple plots in same figure.
install.packages("gridExtra")
library(gridExtra)
gridExtra::grid.arrange(g1, g2, ncol=2)
```


hwy vs displ
Ggplot2 - Facet Grid - Multiple plots in one figure
2seater
compact
midsize
minivan
pickup
subcompact
suv
4
5
6
8
hwy
displ
Source: mpg

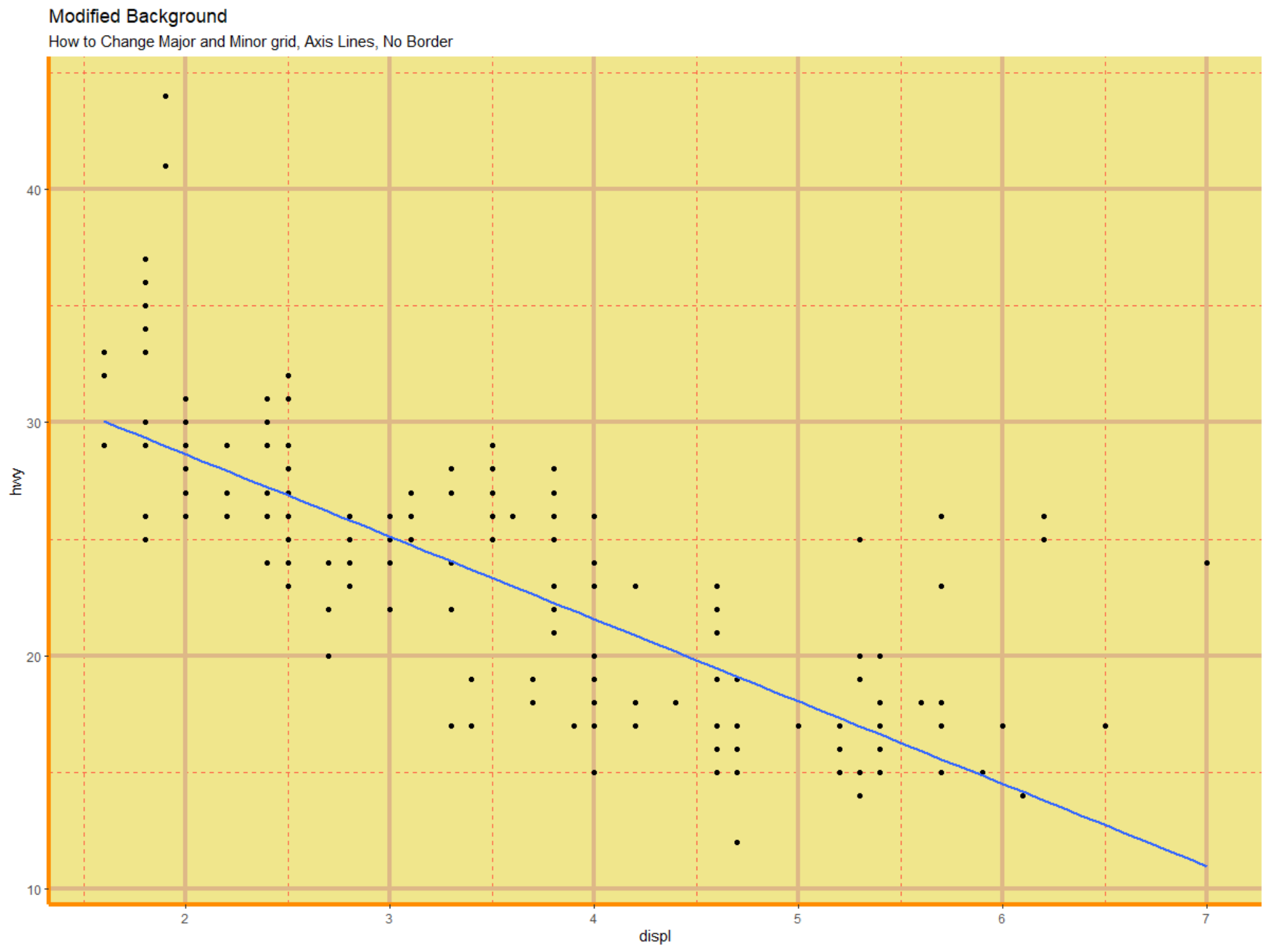

hwy vs displ
Ggplot2 - Faceting - Multiple plots in one figure
2seater
compact
midsize
minivan
pickup
subcompact
suv
audi
chevrolet
dodge
ford
honda
hyundai
jeep
land rover
lincoln
mercury
nissan
pontiac
subaru
toyota
volkswagen
hwy
displ
Source: mpg


hwy vs displ
Ggplot2 - Facet Grid - Multiple plots in one figure
2seater
compact
midsize
minivan
pickup
subcompact
suv
4
5
6
8
hwy
displ
Source: mpg

6. Modifying Plot Background, Major and Minor Axis

- Change plot background

```
# Change Plot Background elements ----------------------------------
g + theme(panel.background = element_rect(fill = 'khaki'),
    panel.grid.major = element_line(colour = "burlywood", size=1.5),
    panel.grid.minor = element_line(colour = "tomato", size=.25,
                          linetype = "dashed"),
    panel.border = element_blank(),
    axis.line.x = element_line(colour = "darkorange", size=1.5,
                          lineend = "butt"),
    axis.line.y = element_line(colour = "darkorange", size=1.5)) +
    labs(title="Modified Background",
         subtitle="How to Change Major and Minor grid, Axis Lines, No Border")

# Change Plot Margins ----------------------------------------------
g + theme(plot.background=element_rect(fill="salmon"),
          plot.margin = unit(c(2, 2, 1, 1), "cm")) + # top, right, bottom, left
          labs(title="Modified Background", subtitle="How to Change Plot Margin")
```


Modified Background
How to Change Major and Minor grid, Axis Lines, No Border
hwy
40
30
20
10
2
3
4
5
6
7
displ


Modified Background
How to Change Plot Margin
40
30
20
10
hwy
2
3
4
5
6
7
displ

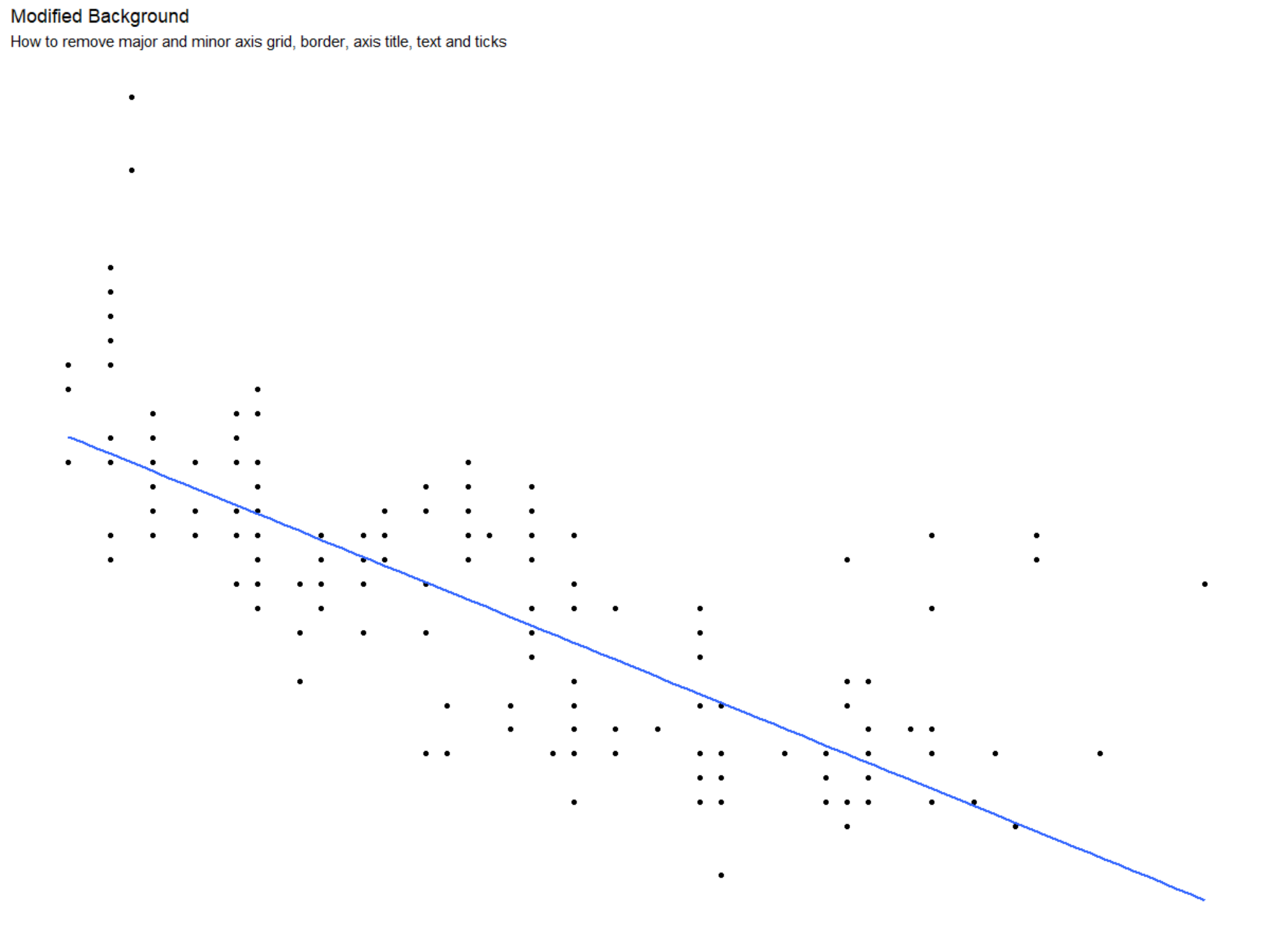
6. Modifying Plot Background, Major and Minor Axis

- Remove major and minor grid, change border, axis title, text, and ticks

```
# How to Remove Major and Minor Grid, Change Border, Axis Title, Text and Ticks
g + theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank()) +
          labs(title="Modified Background",
          subtitle="How to remove major and minor axis grid, border,
                    axis title, text and ticks")
```

Modified Background

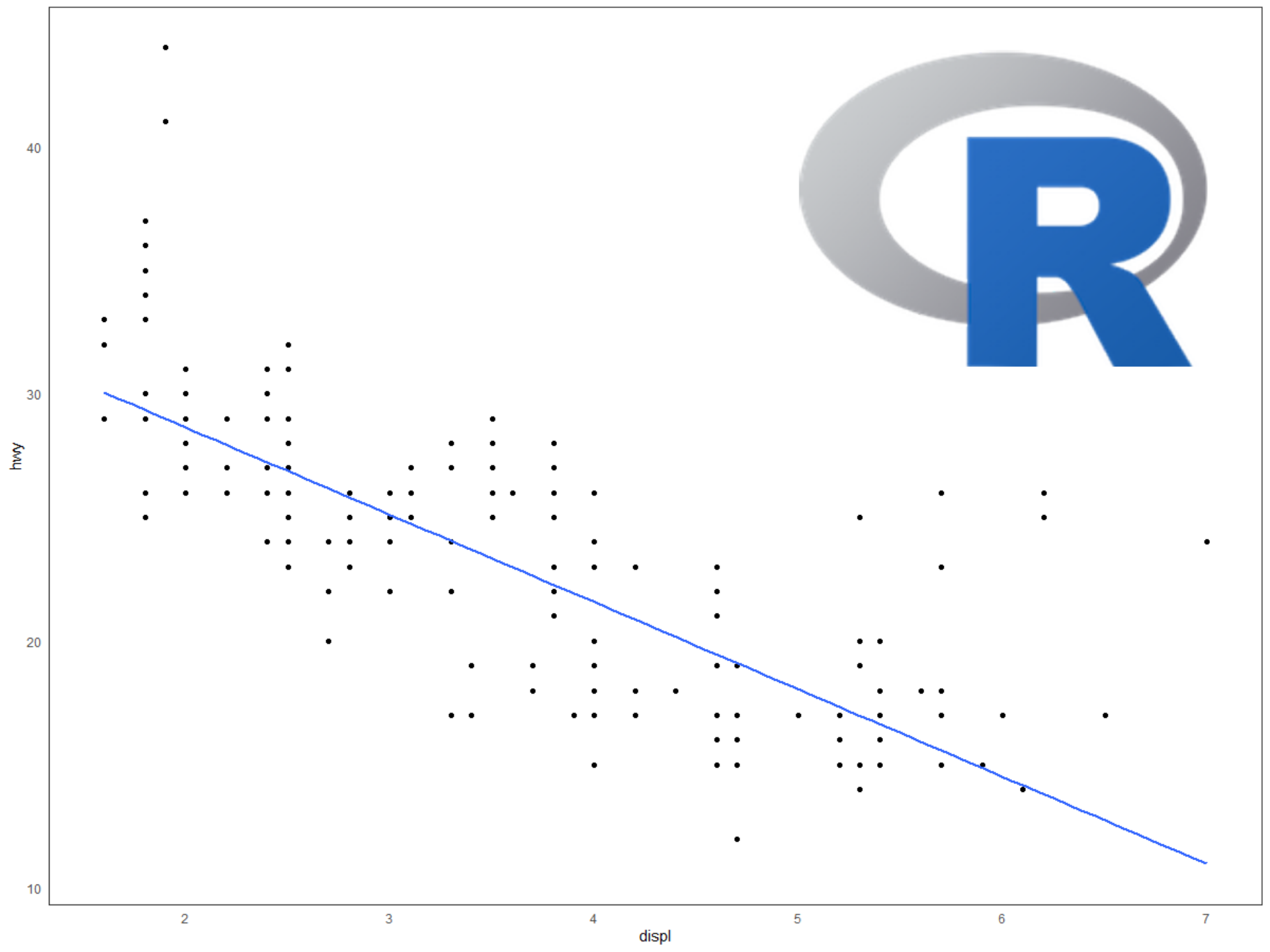
How to remove major and minor axis grid, border, axis title, text and ticks



6. Modifying Plot Background, Major and Minor Axis

- Add an image in background

```
# Add an image in background
install.packages("grid")
library(grid)
install.packages("magick")
library(magick)

image_url <- "https://www.r-project.org/Rlogo.png"
pic <- image_read(image_url)
print(pic)
g_pic <- rasterGrob(pic, interpolate=TRUE)

# Base Plot
…

g + theme(panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          plot.title = element_text(size = rel(1.5), face = "bold"),
          axis.ticks = element_blank()) +
          annotation_custom(g_pic, xmin=5, xmax=7, ymin=30, ymax=45)
```


40
30
20
10
hwy
2
3
4
5
6
7
displ


ANY
questions?