DHRUV GARG 16 BCE 1190

DATA VISUALIZATION

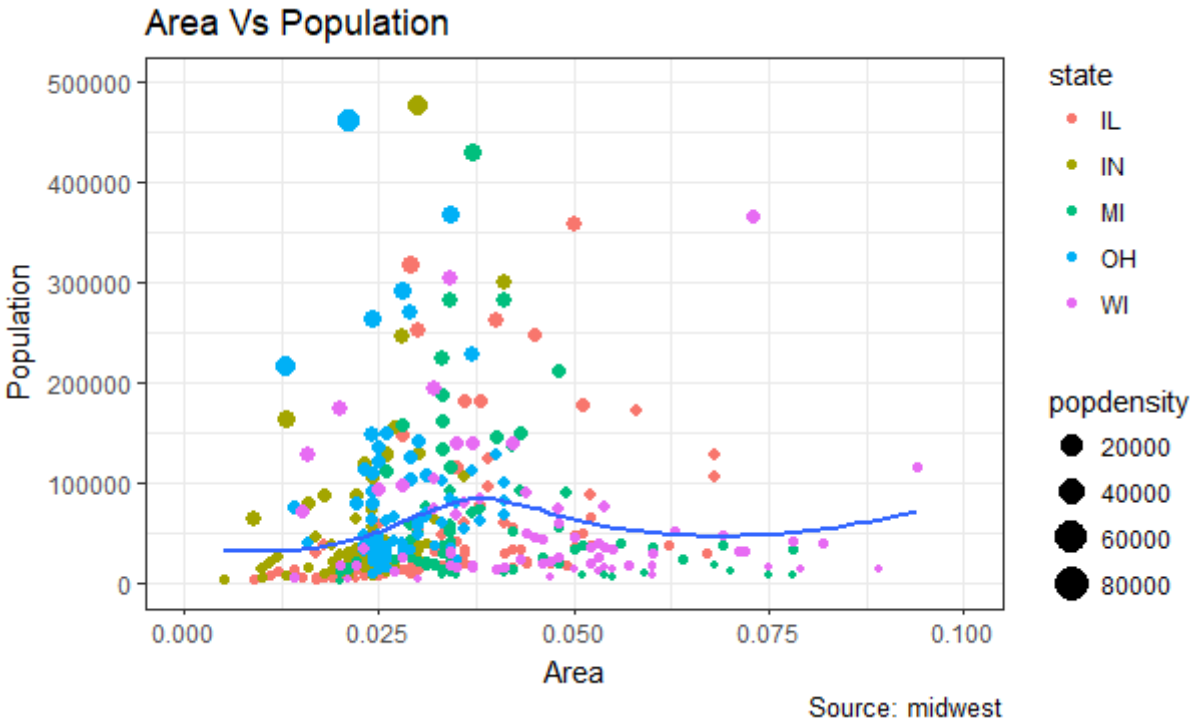
WEEK 2

GGPLOT2 – Part 2

1. Theme

```
options(scipen=999)

library(ggplot2)

data("midwest", package = "ggplot2")

theme_set(theme_bw())

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)
```

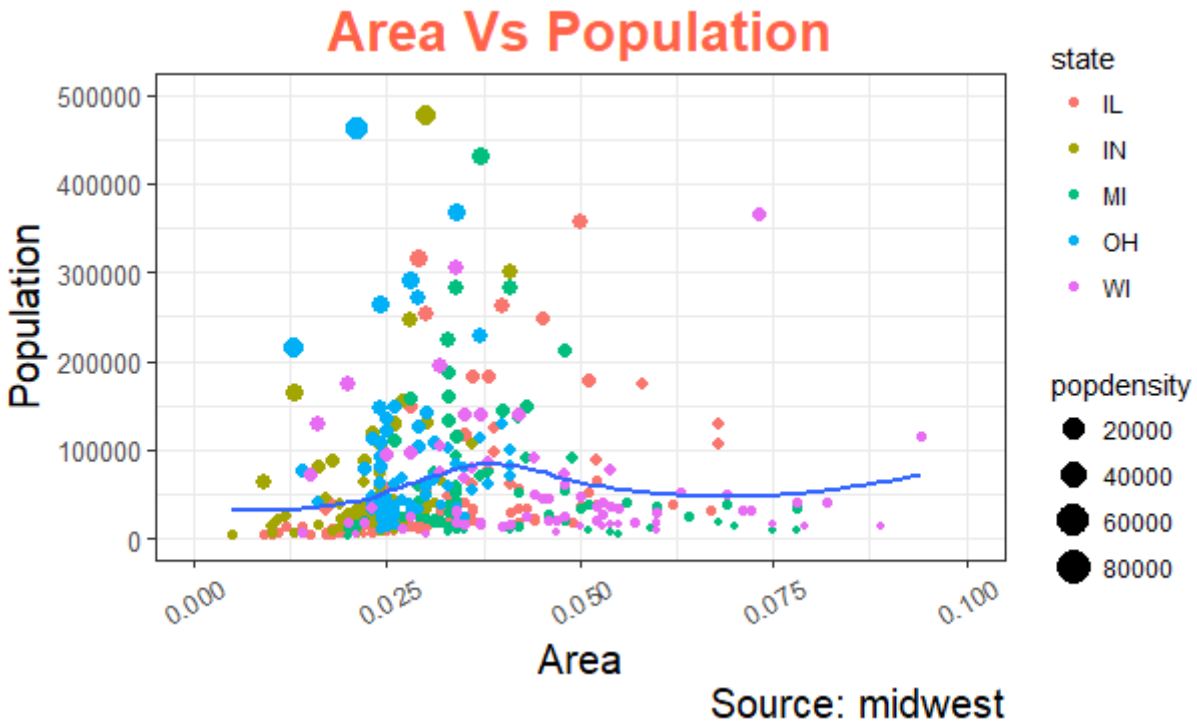
// Use theme_set() to completely override the current theme. Here we have the old theme so we can later restore it.



2. Adding plot and axis titles

```
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 + theme(plot.title=element_text(size=20, face="bold", family = "American Typewriter", color="tomato",
hjust=0.5, lineheight=1.2), plot.subtitle=element_text ( size=15, family = "American Typewriter",
face="bold", hjust=0.5), plot.caption = element_text(size=15) , axis.title.x = element_text(vjust=10, size=15),
axis.title.y = element_text(size=15), axis.text.x = element_text(size=10, angle = 30, vjust = .5),
axis.text.y = element_text(size=10))
```

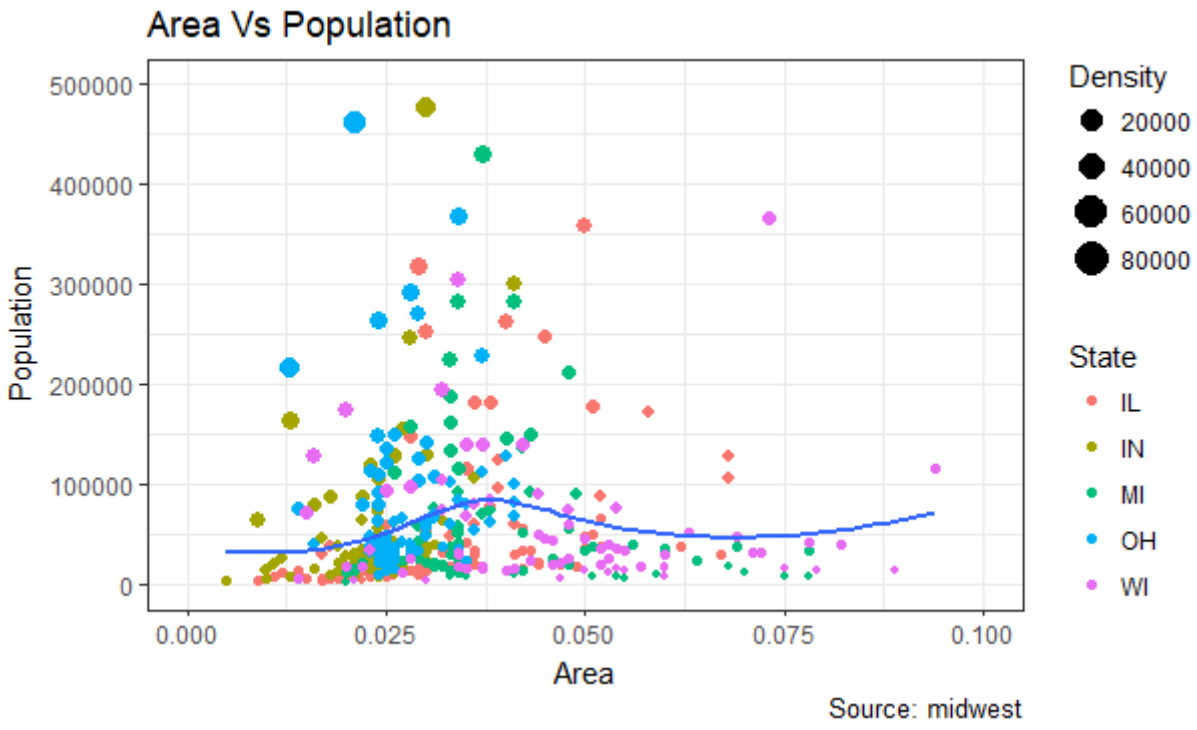
//Using attributes of plot: subtitle, caption and axis: title, text



3. Modifying legend title (using labs)

```
library(ggplot2)

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

//labs()function is used to specify the labels

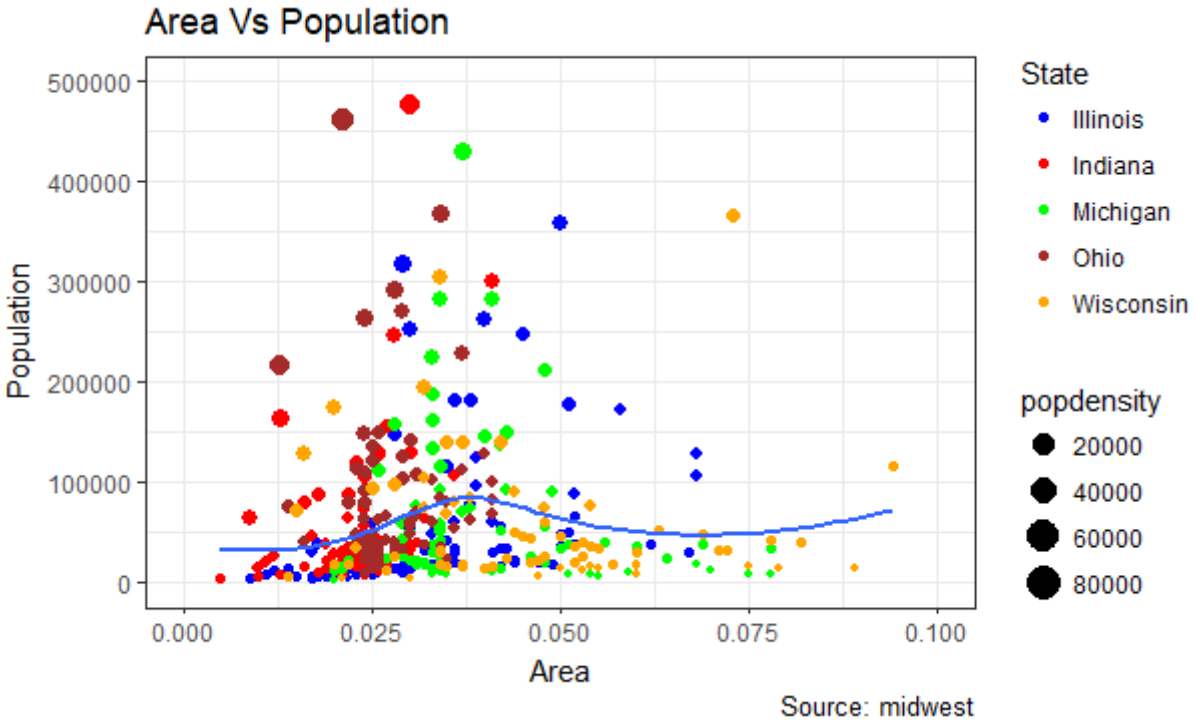


4. Change legend labels and print colours for categories

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

// scale_color_manual() allows you to specify you own set of mappings from levels in the data to aesthetic values.



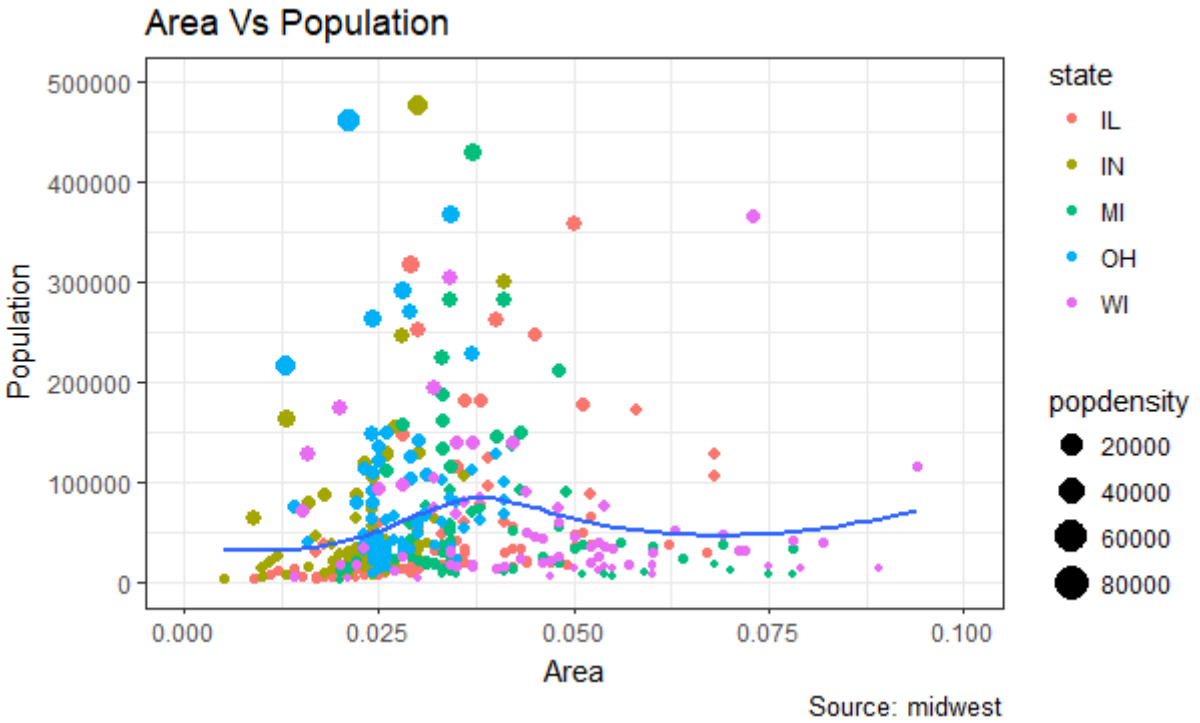
5. Changing the order of the legend

```
library(ggplot2)
```

```
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(colour = guide_legend(order = 1), size = guide_legend(order = 2))
```

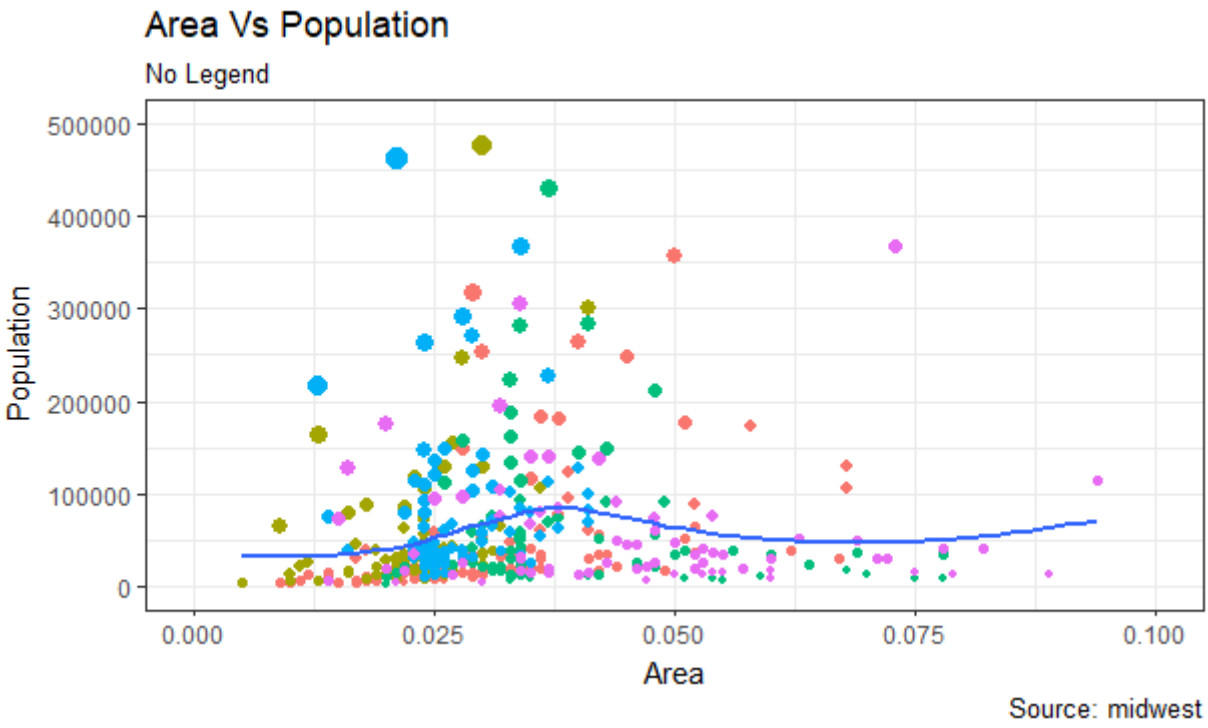
// Legend type guide shows key (i.e., geoms) mapped onto values. Legend guides for various scales are integrated if possible.



6. Remove the legend and change legend positions

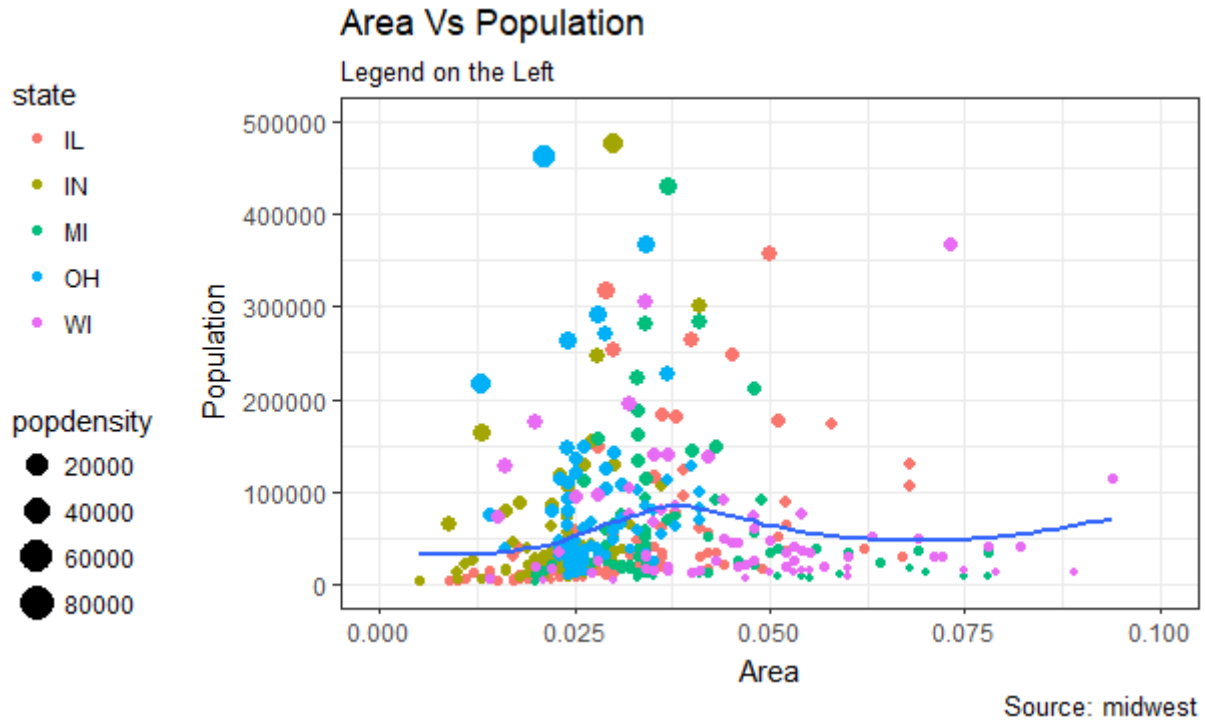
NO LEGEND

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



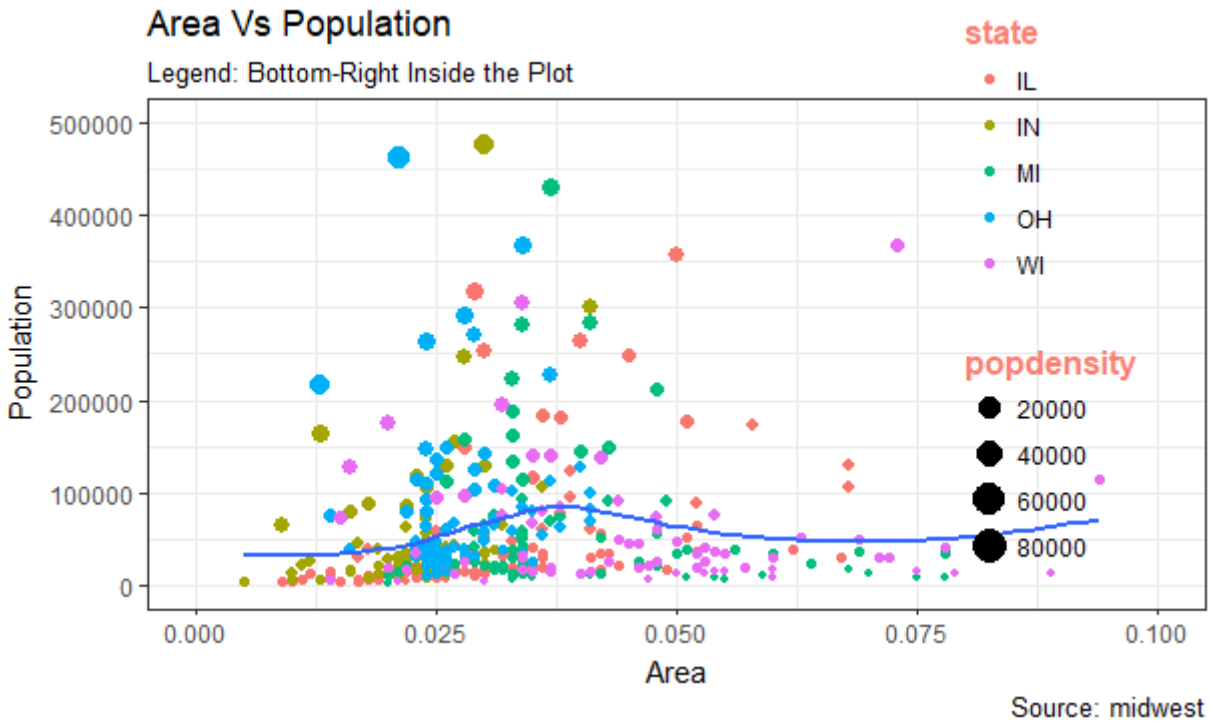
LEFT LEGEND

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



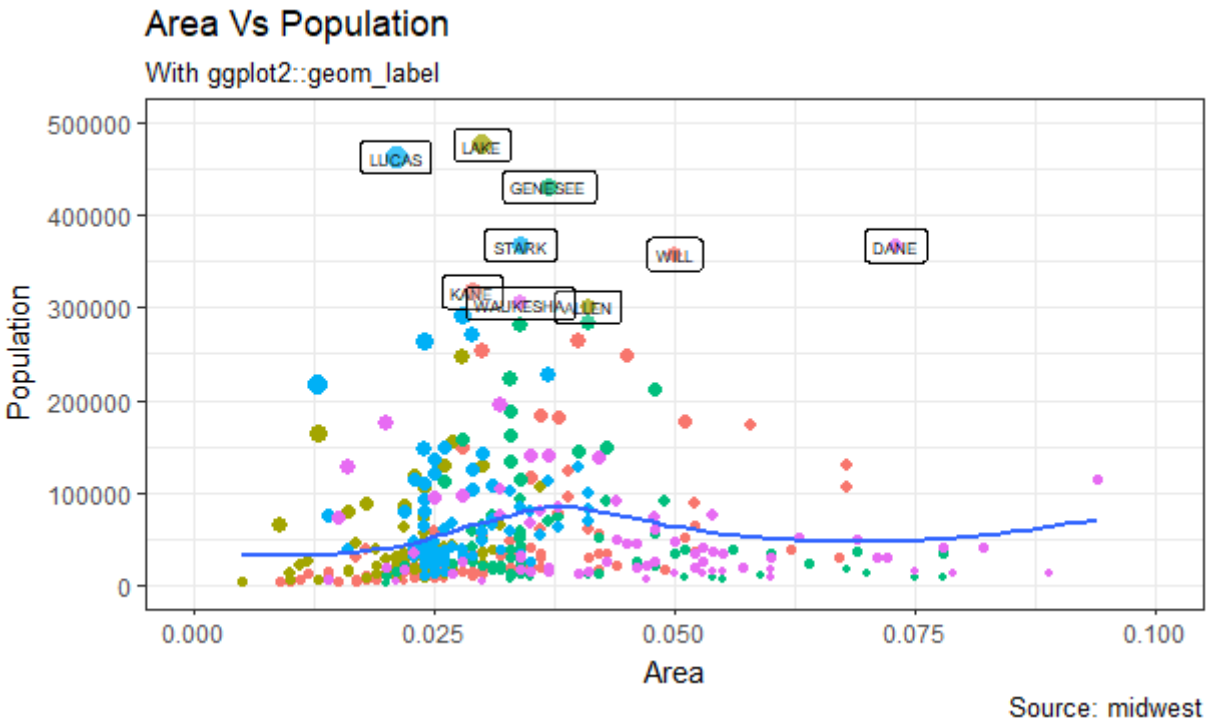
BOTTOM-RIGHT LEGEND (INSIDE THE PLOT)

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")



7. Adding text and label around the points

```
midwest_sub <- midwest[midwest$poptotal > 300000]
midwest_sub$large_county <- ifelse(midwest_sub$poptotal > 300000, midwest_sub$county, "")

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 + geom_text(aes(label=large_county), size=2, data=midwest_sub) + labs(subtitle="With
ggplot2::geom_text") + theme(legend.position = "None")
gg + geom_label(aes(label=large_county), size=2, data=midwest_sub, alpha=0.25) + labs(subtitle="With
ggplot2::geom_label") + theme(legend.position = "None")
```

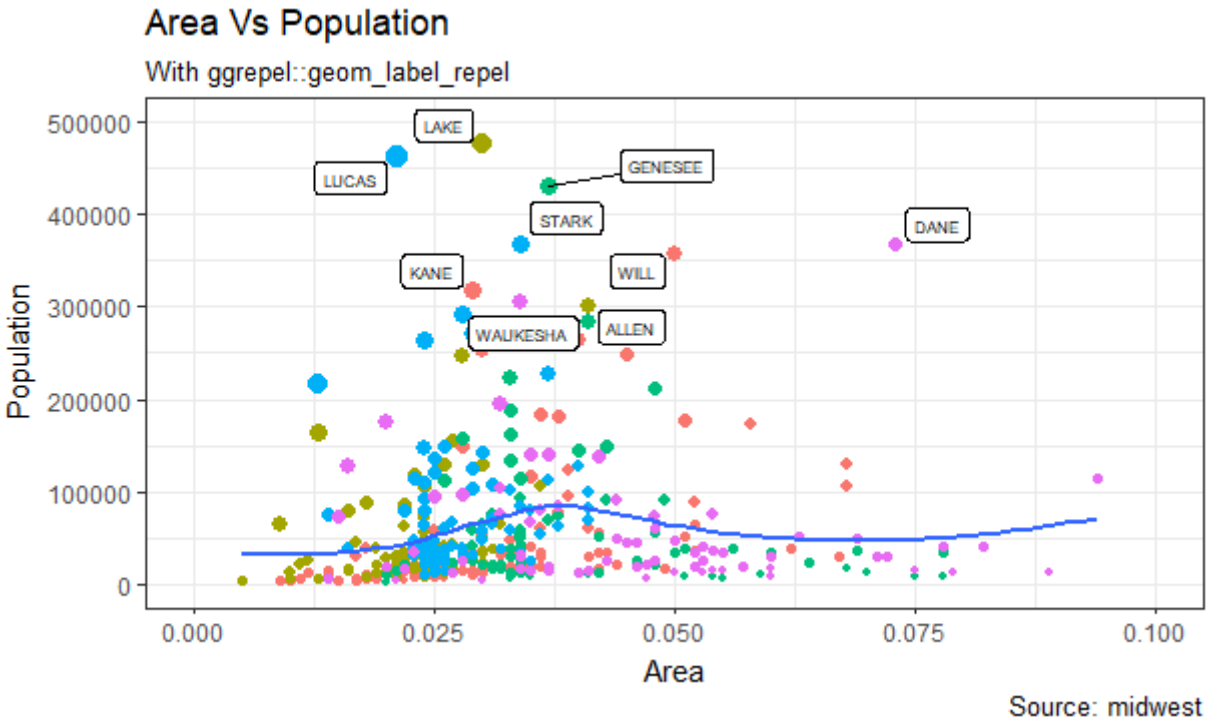


<u>PLOT AND TEXT LABEL REPELS EACH OTHER</u>

```
library(ggrepel)

gg + geom_text_repel(aes(label=large_county), size=2, data=midwest_sub) + labs(subtitle="With
ggrepel::geom_text_repel") + theme(legend.position = "None")

gg + geom_label_repel(aes(label=large_county), size=2, data=midwest_sub) + labs(subtitle="With
ggrepel::geom_label_repel") + theme(legend.position = "None")
```

//We can repel the text labels away from each other by loading ggrepel and using geom_text_repel

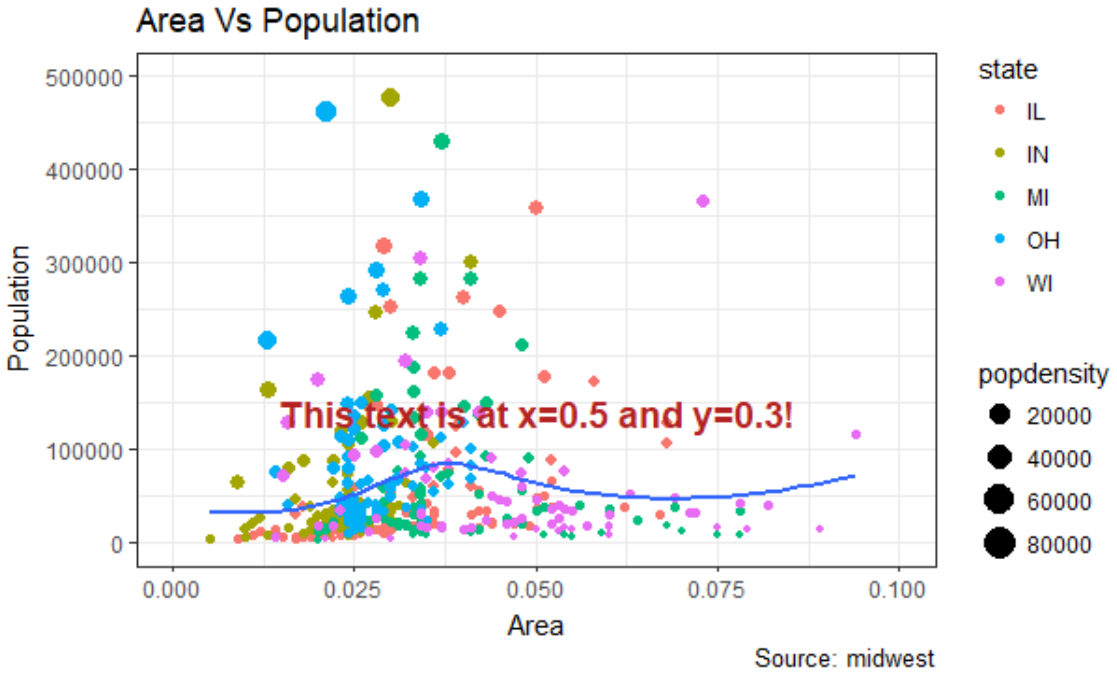
// geom_label_repel draws a rectangle underneath the text, making it easier to read.



8. Adding annotations inside the 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")
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)
```

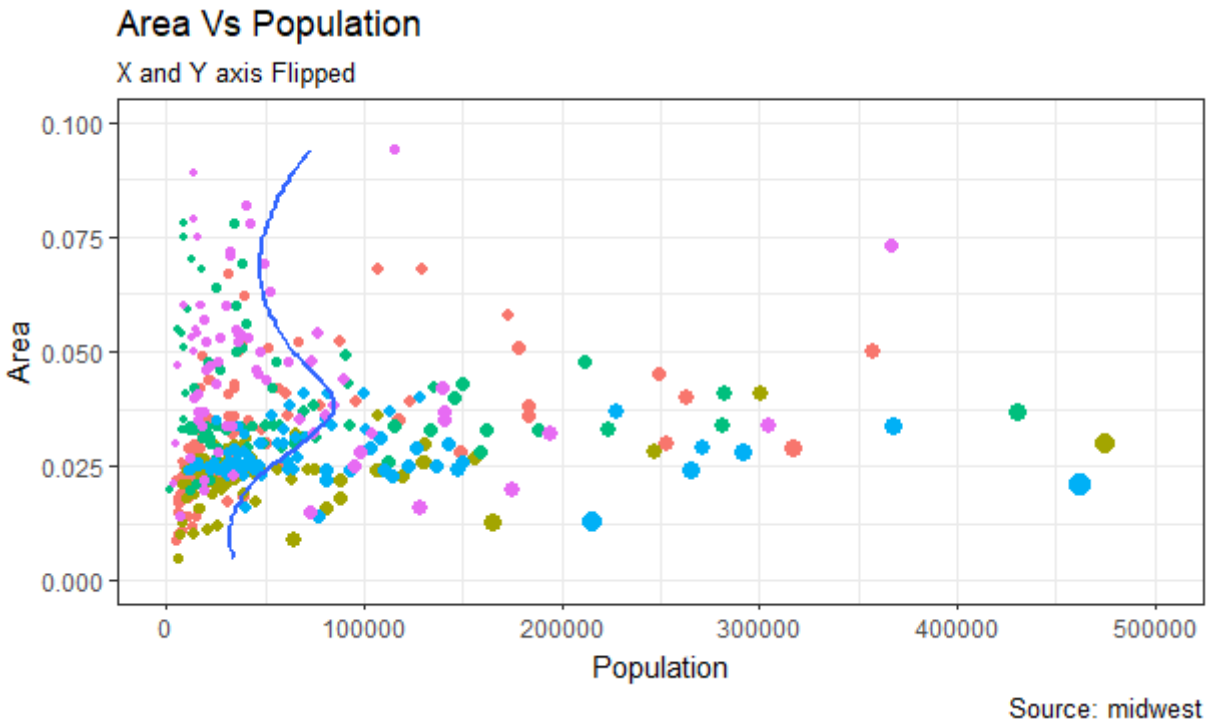
//annotation_custom() is a special geom intended for use as static annotations that are the same in every panel. These annotations will not affect scales.



9. Flipping the X and Y axis

```
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", subtitle="X and Y axis Flipped") +
theme(legend.position = "None")

gg + coord_flip()
```

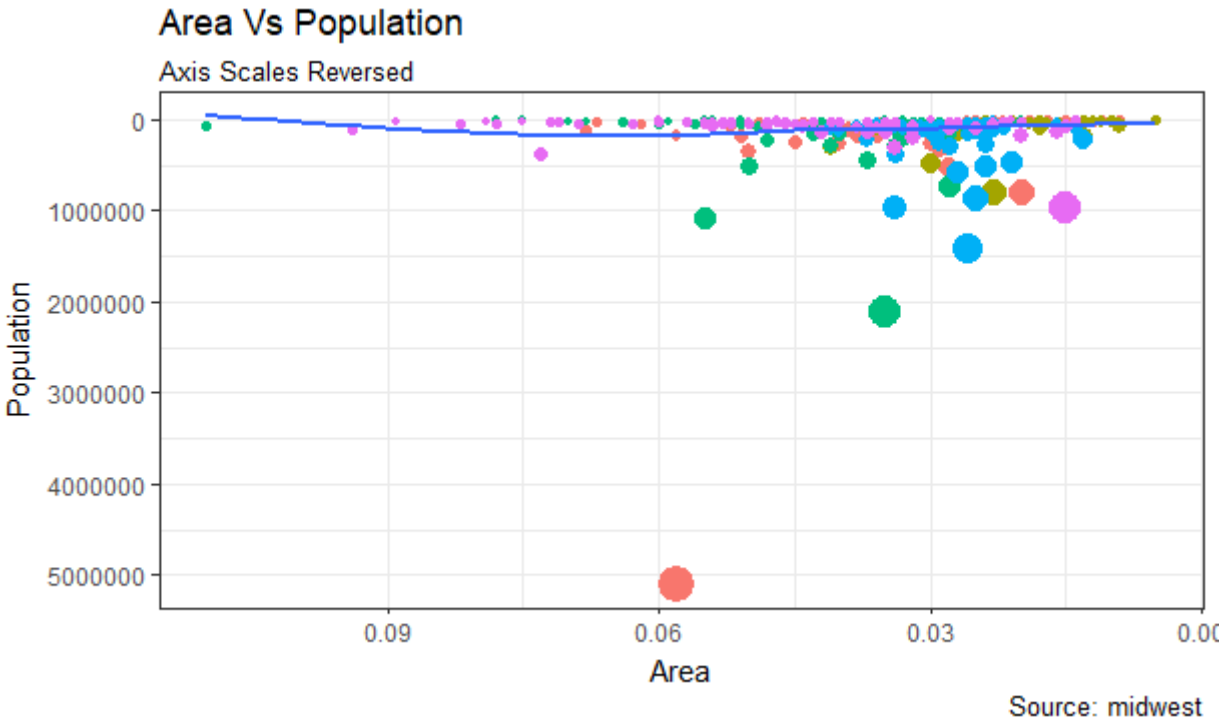
//coord_flip() flips cartesian coordinates so that horizontal becomes vertical, and vertical, horizontal.



10. Reversing the scale of an axis

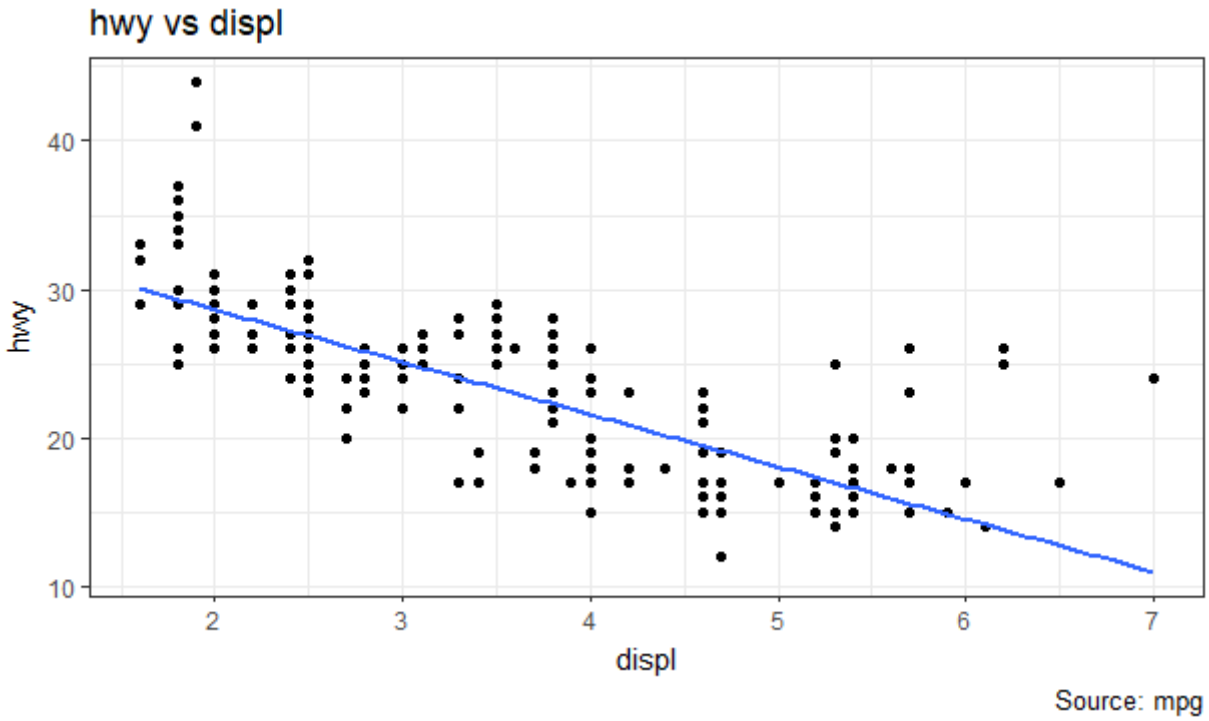
```
gg <- ggplot(midwest, aes(x=area, y=poptotal)) +
  geom_point(aes(col=state, size=popdensity)) +  geom_smooth(method="loess", se=F) + xlim(c(0, 0.1)) +
ylim(c(0, 500000)) +  labs(title="Area Vs Population", y="Population", x="Area", caption="Source: midwest",
subtitle="Axis Scales Reversed") + theme(legend.position = "None")

gg + scale_x_reverse() + scale_y_reverse()
```

// scale_x_reverse() reverse the x axis scale. Similarly, scale_y_reverse() for the y axis.



11. Faceting – drawing multiple plots in one figure

```
data(mpg, package="ggplot2")

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()

plot(g)
```



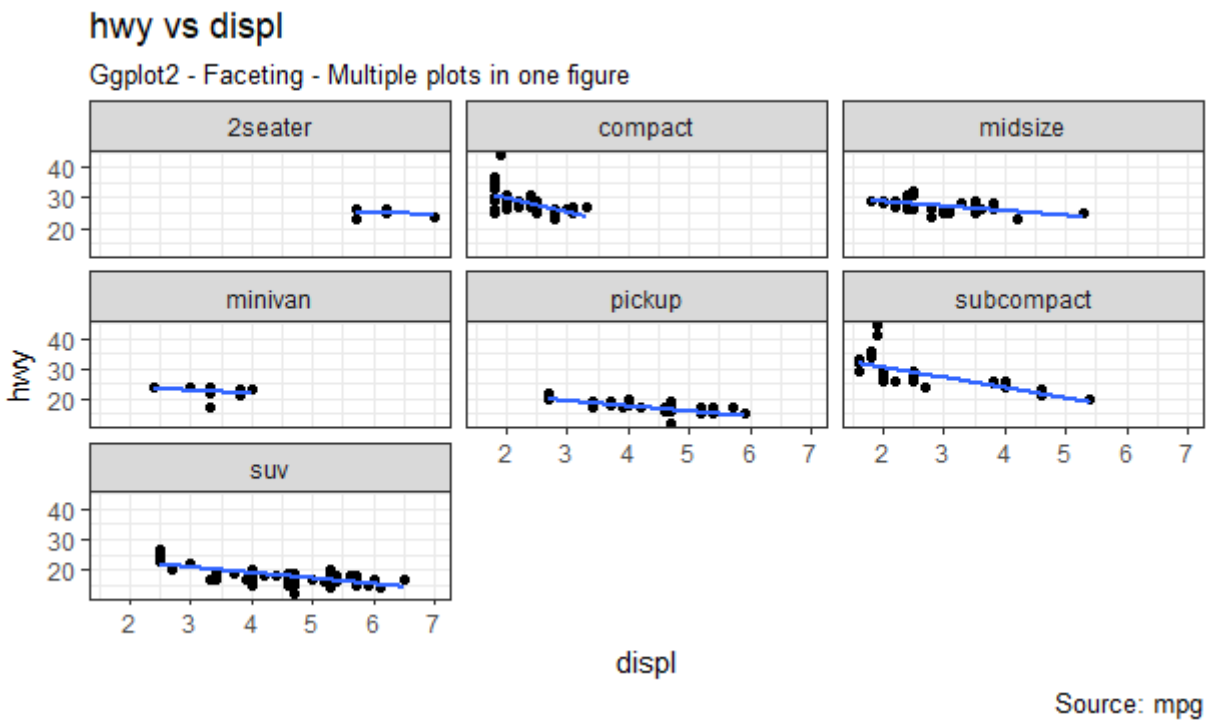
FACET WRAP

g <- ggplot(mpg, aes(x=displ, y=hwy)) + geom_point() + geom_smooth(method = "lm", se=FALSE) + theme_bw()

Common scales:

g + facet_wrap(~ class, nrow=3) + labs(title="hwy vs displ", caption = "Source: mpg", subtitle="Ggplot2 - Faceting - Multiple plots in one figure")

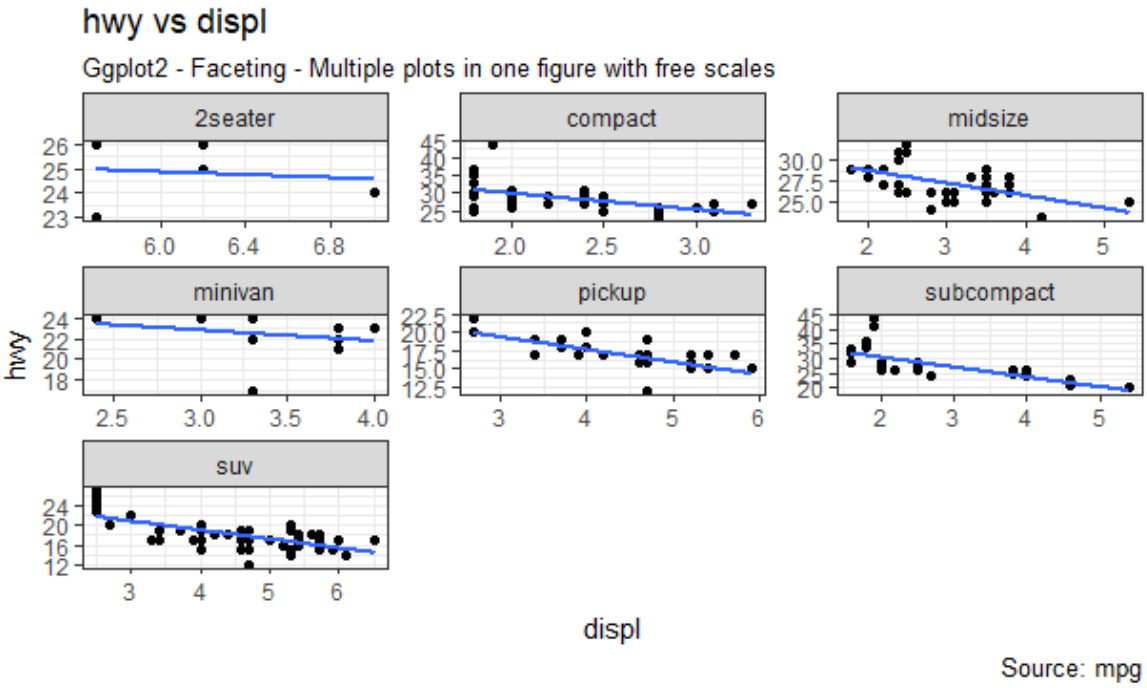
// facet_wrap wraps a 1d sequence of panels into 2d



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")

// facet_wrap wraps a 1d sequence of panels into 2d

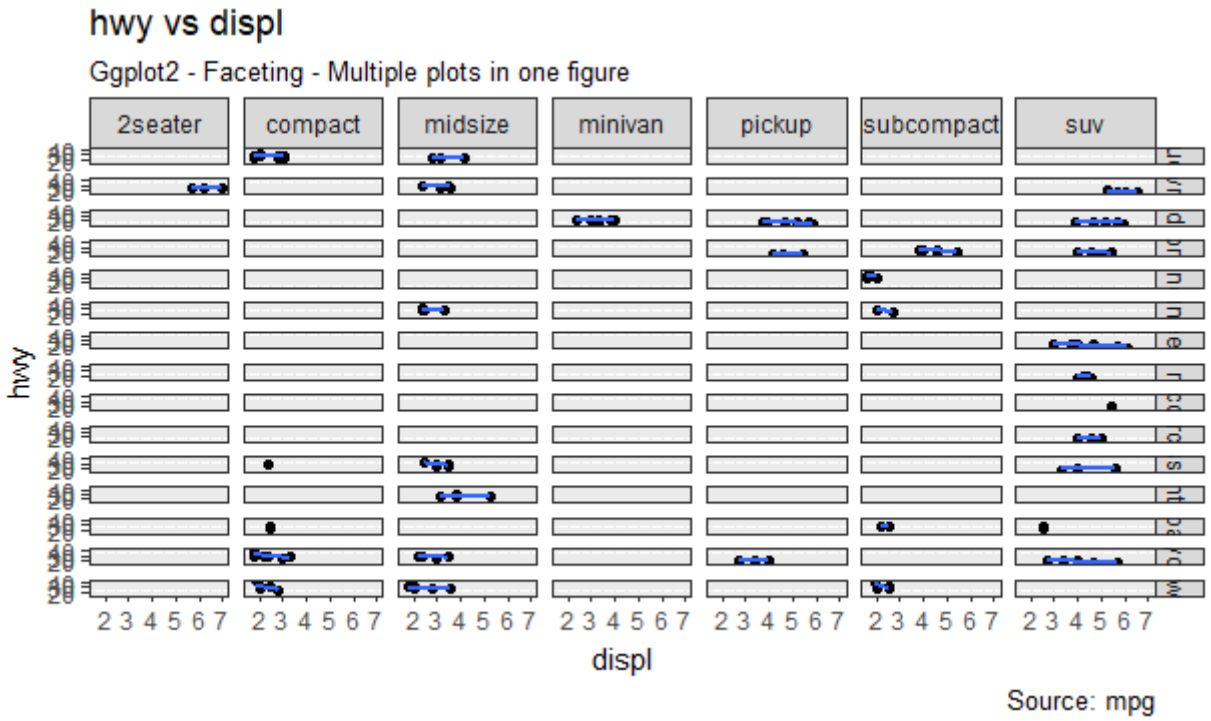


12. Facet grid

Variation with manufacturer

```
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()

g1 <- g + facet_grid(manufacturer ~ class)

plot(g1)
```

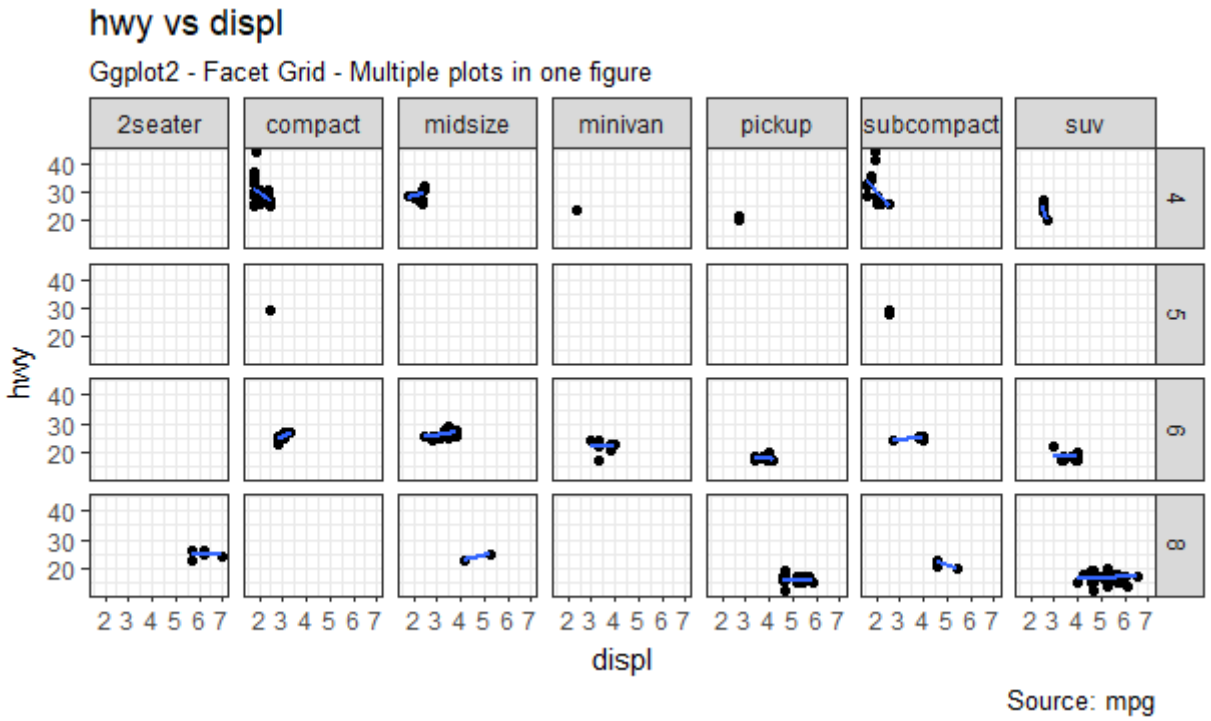
// facet_grid forms a matrix of panels defined by row and column facetting variables. It is most useful when you have two discrete variables



Variation with cylinder

```
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()

g2 <- g + facet_grid(cyl ~ class)

plot(g2)
```

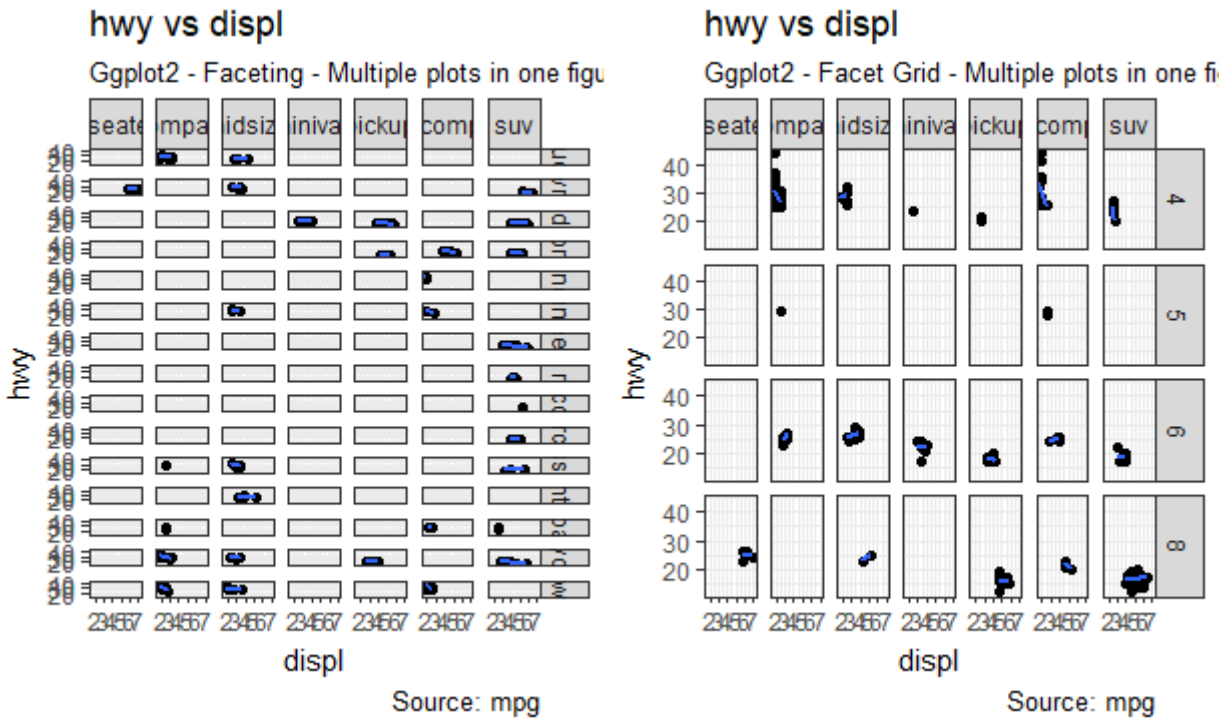
// facet_grid forms a matrix of panels defined by row and column facetting variables. It is most useful when you have two discrete variables



Drawing multiple plots in the same figure
library(gridExtra)

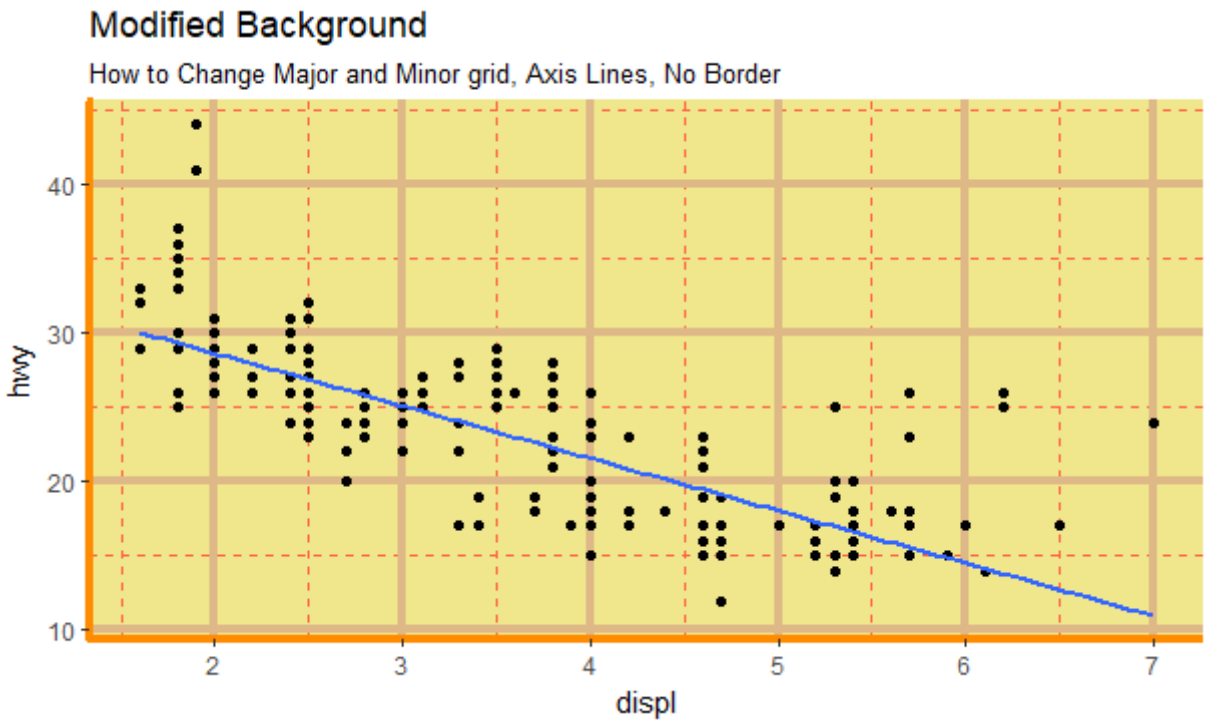
gridExtra::grid.arrange(g1, g2, ncol=2)

//The grid package provides low-level functions to create graphical objects (grobs), and position them on a page in specific viewports .



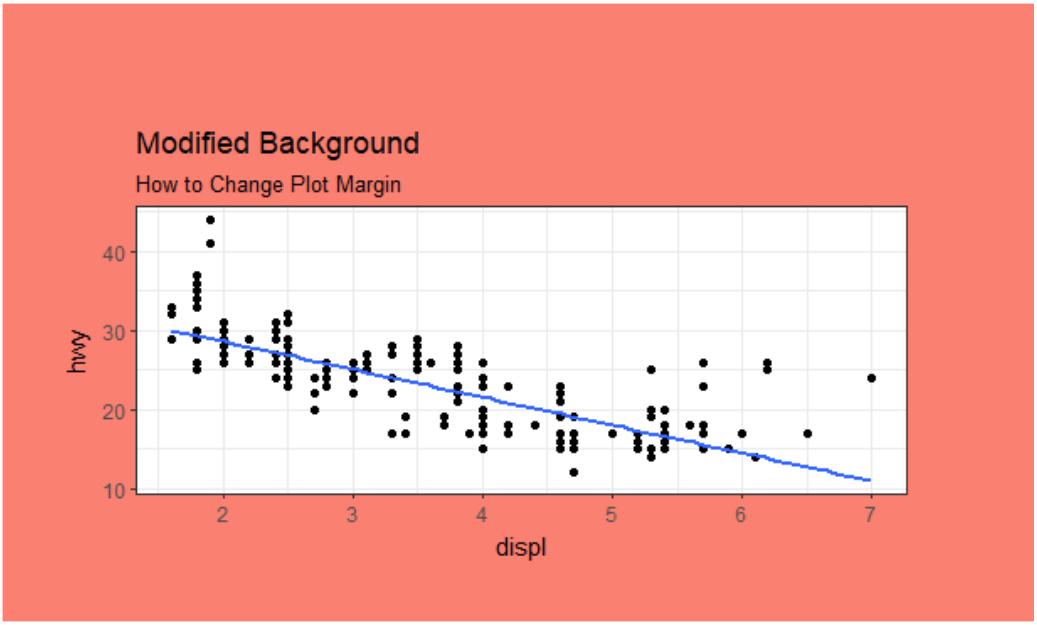
13. Modifying plot background, major and minor axes

```
g <- ggplot(mpg, aes(x=displ, y=hwy)) + geom_point() + geom_smooth(method="lm", se=FALSE) + theme_bw()

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")
```



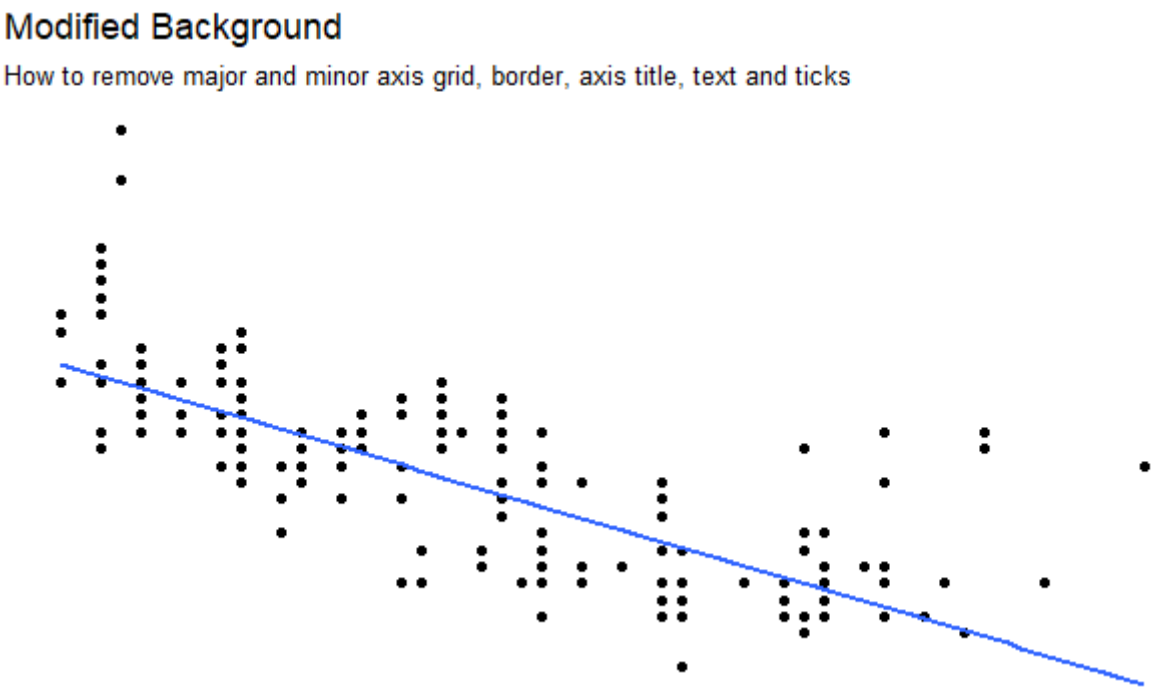
Changed plot margin

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



14. Removing major and minor grid, change border, axis title, text and ticks

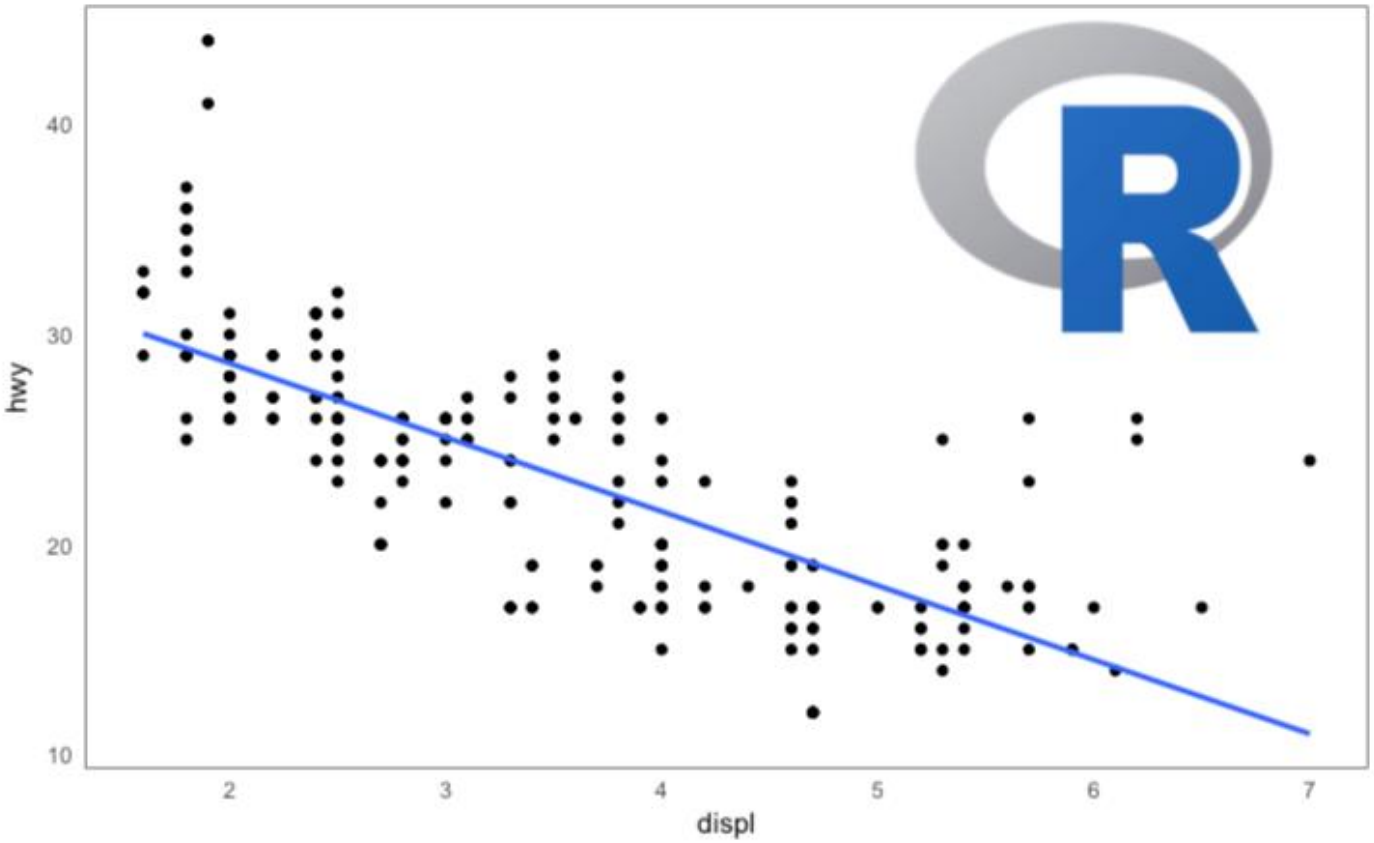
```
g <- ggplot(mpg, aes(x=displ, y=hwy)) + geom_point() + geom_smooth(method="lm", se = FALSE) + theme_bw()

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")
```



15. Adding an image in the background

```
library(grid)
library(png)

img <- png::readPNG("Desktop/rlogo.png")
g_pic <- rasterGrob(img, interpolate=TRUE)

g <- ggplot(mpg, aes(x=displ, y=hwy)) + geom_point() + geom_smooth(method="lm", se=FALSE) + theme_bw()

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)
```

//Takes the argument as source - either name of the file to read from or a raw vector representing the PNG file content.


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