IE6600 Computation and Visualization for Analytics

Application of Data Visualization

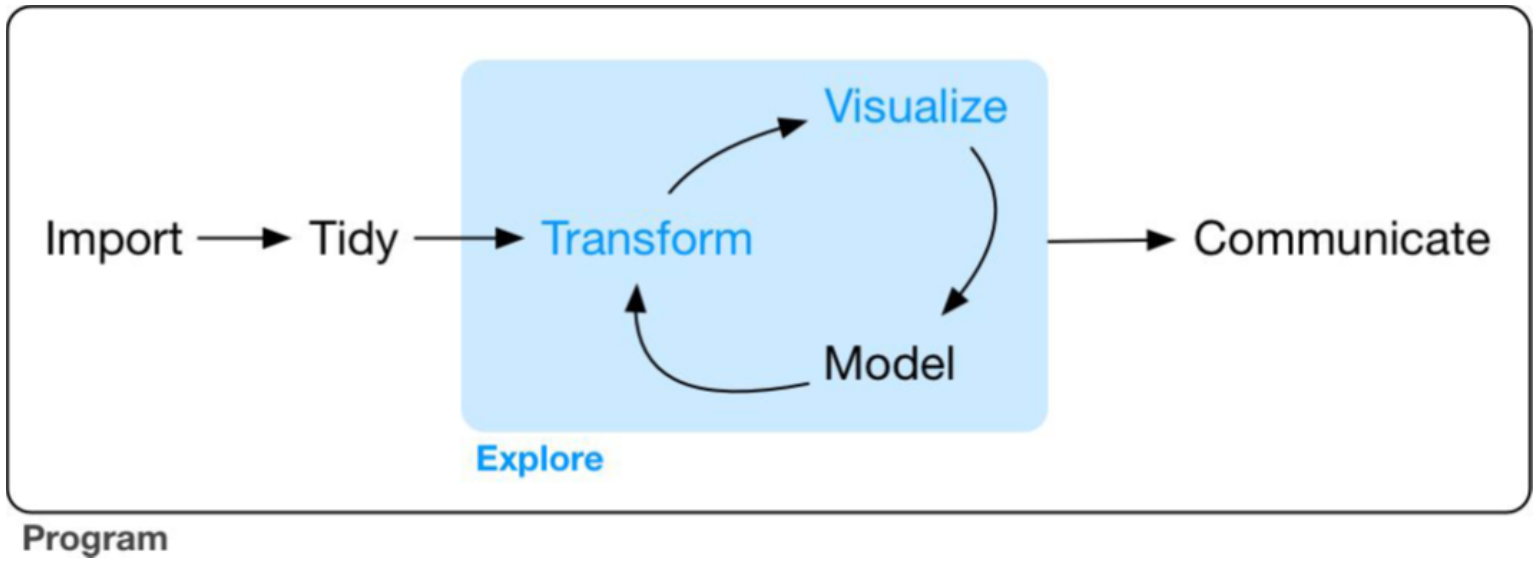
Zhenyuan Lu

updated: 2022-07-11

Applicatin of Data Visualization

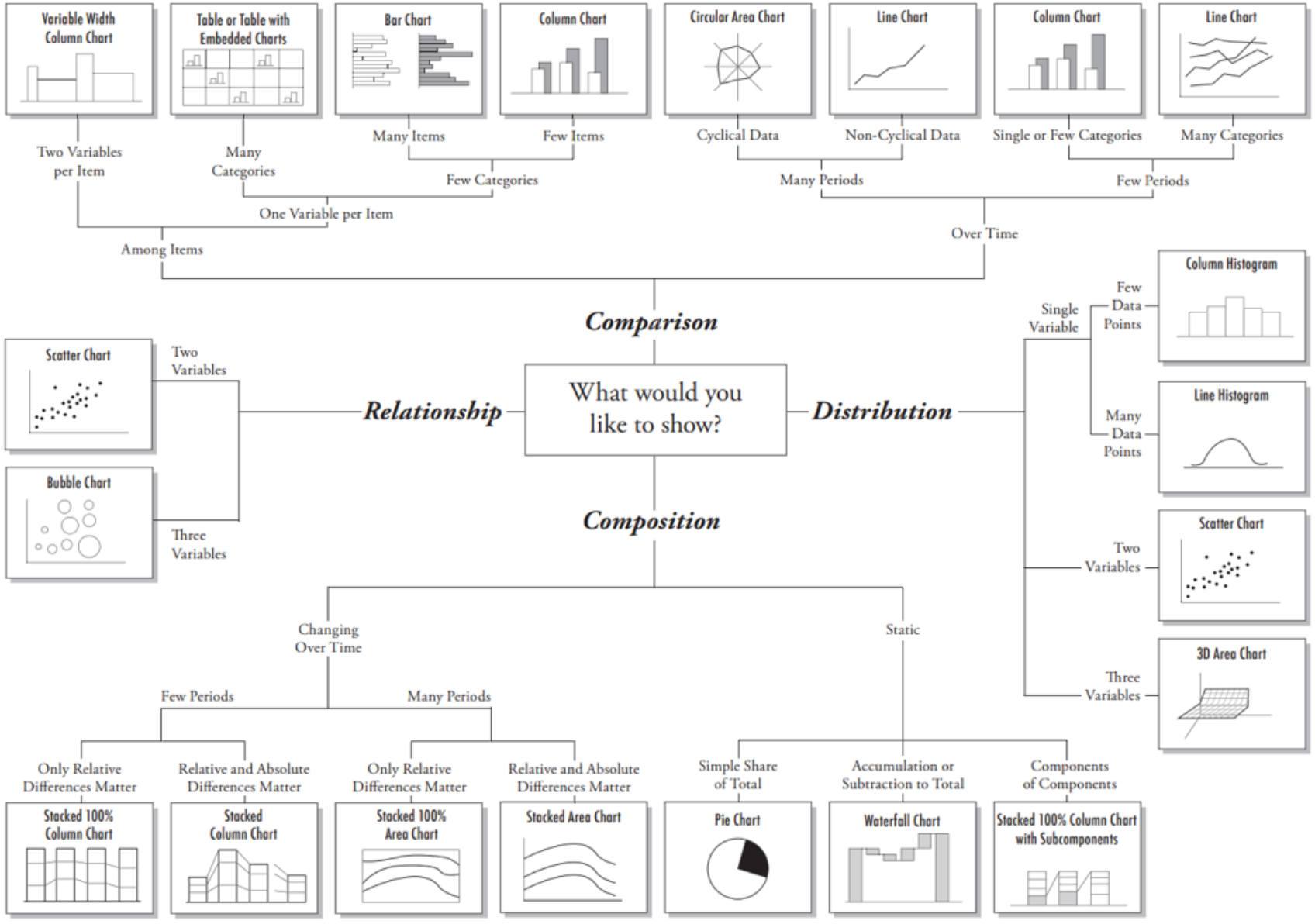
© 2022 Zhenyuan LU

Goal



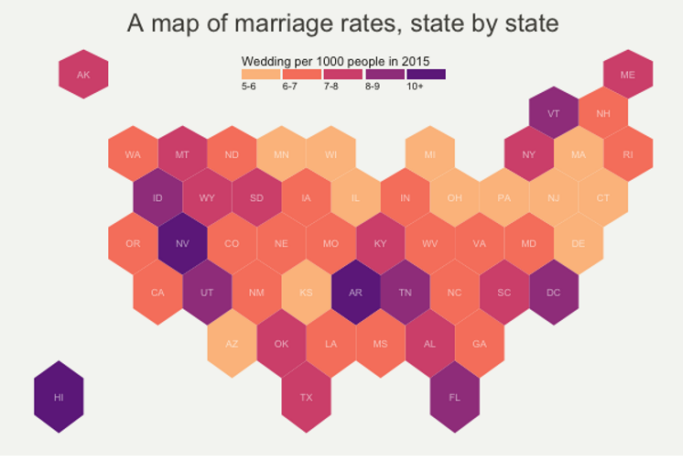
Wickham, Hadley, and Garrett Grolemund. R For Data Science. OReilly, 2017.

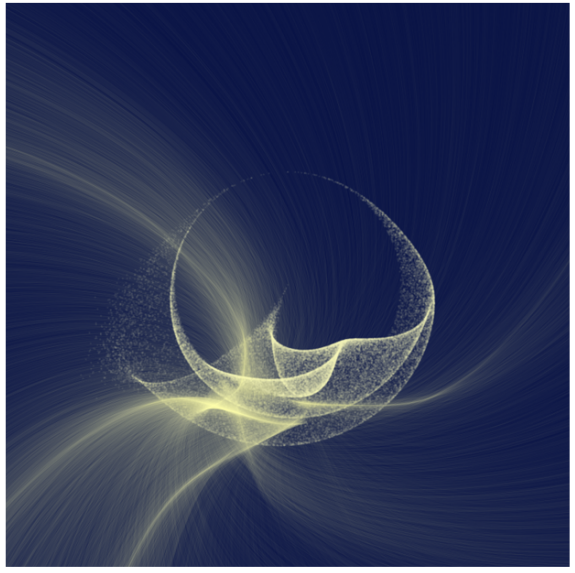
© 2022 Zhenyuan LU

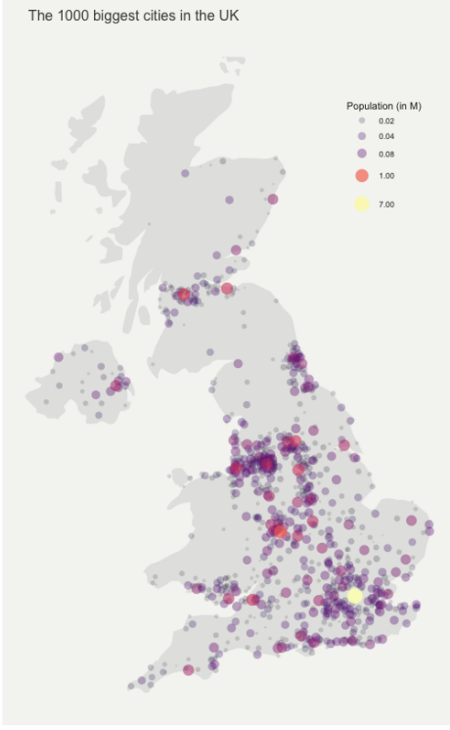


© 2022 Zhenyuan LU

All the figures above are generated by ggplot2! Yes, included the middle one!



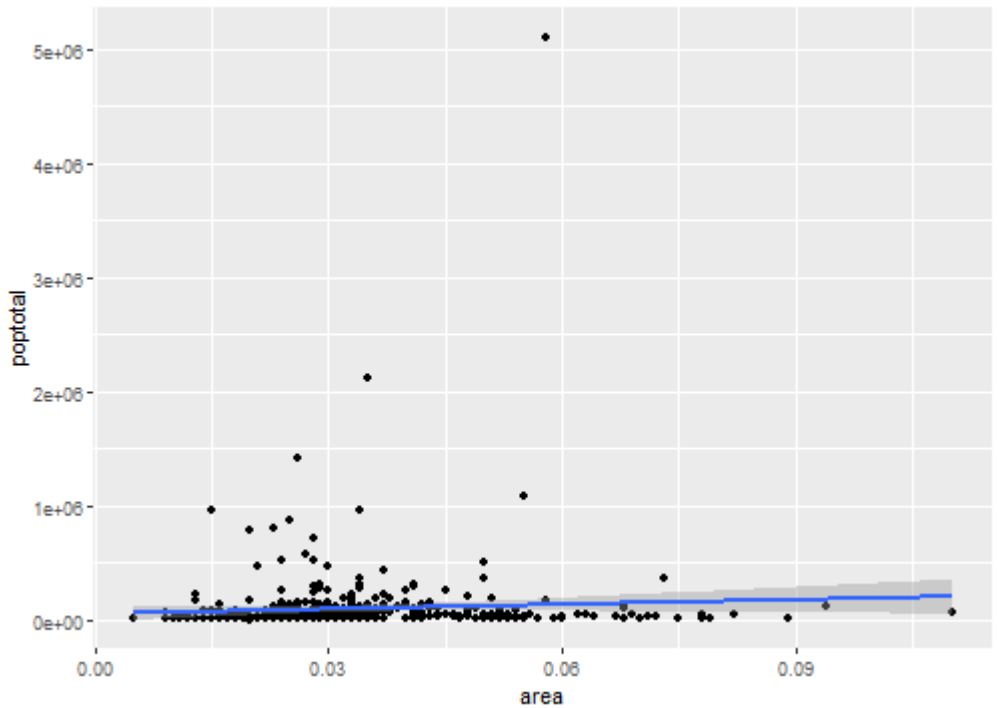




© 2022 Zhenyuan LU

ggplot2 - simple plot

```
library(tidyverse)
# area and poptotal are columns in 'midwest'
midwest %>%
  ggplot(aes(x=area, y=poptotal)) +
  geom_point() + geom_smooth(method="lm")
```

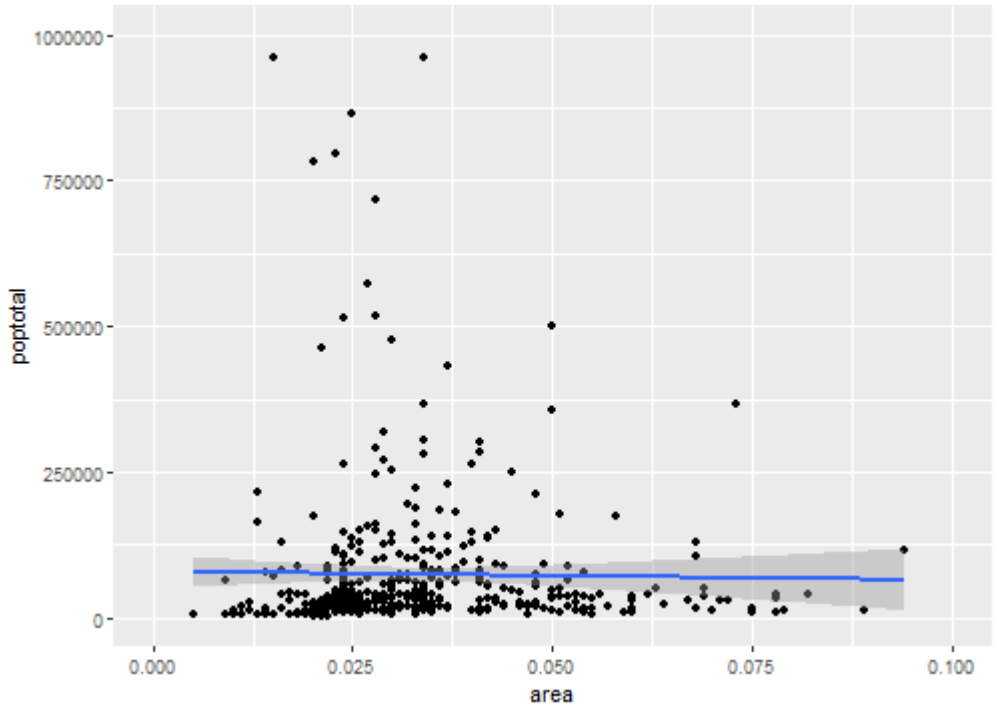


© 2022 Zhenyuan LU

Adjustment of X And Y Axis Limits

`Method 1`: deleting the points outside the range

```
ggplot(midwest, aes(x = area, y = poptotal)) +
  geom_point() +
  geom_smooth(method = "lm") +
  xlim(c(0, 0.1)) + ylim(c(0, 1000000))
```

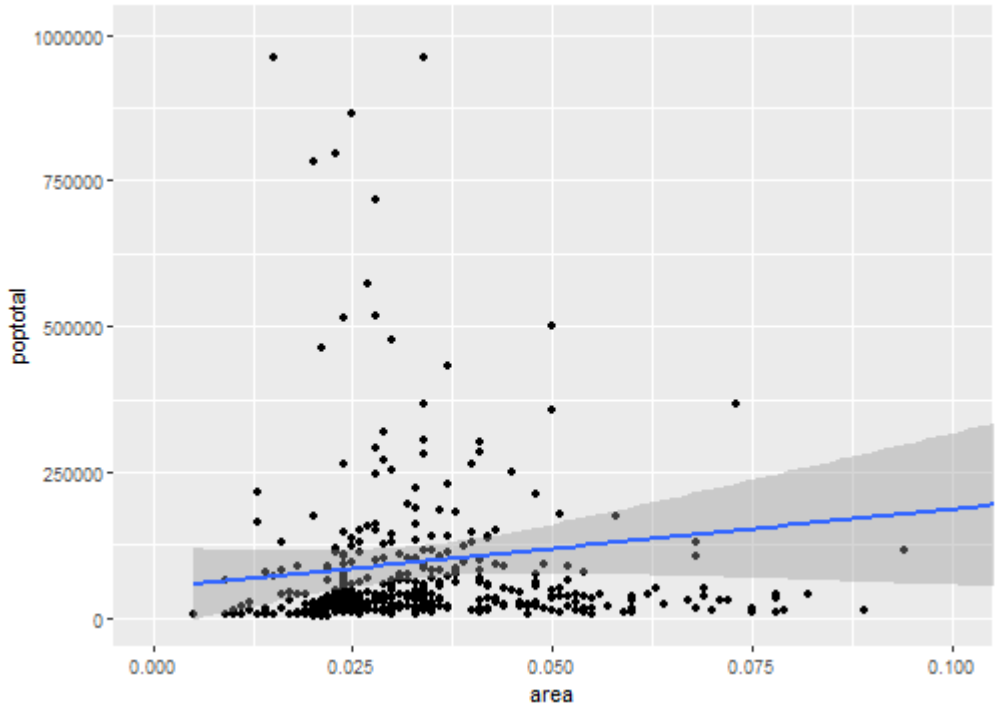


© 2022 Zhenyuan LU

Adjustment of X And Y Axis Limits

Method 2: zooming in

```
ggplot(midwest, aes(x = area, y = poptotal)) +
  geom_point() +
  geom_smooth(method = "lm") +
  coord_cartesian(xlim = c(0, 0.1), ylim = c(0, 1000000))
```



© 2022 Zhenyuan LU

Titles and Axis Labels

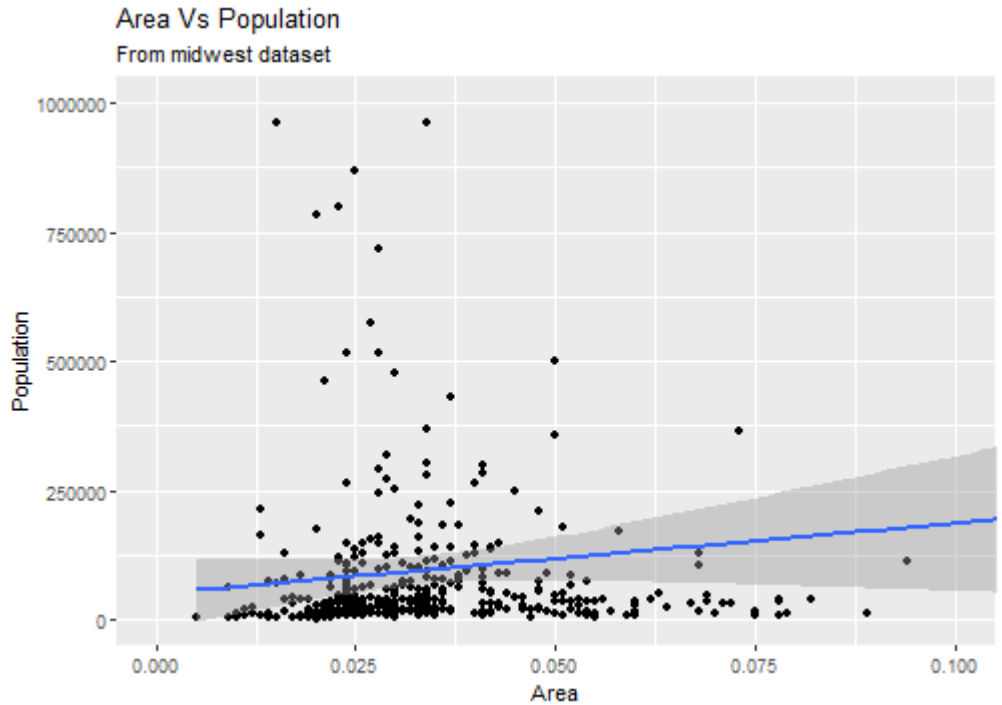
Two different ways to change the titles and labels `Method 1`

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

© 2022 Zhenyuan LU

Method 2

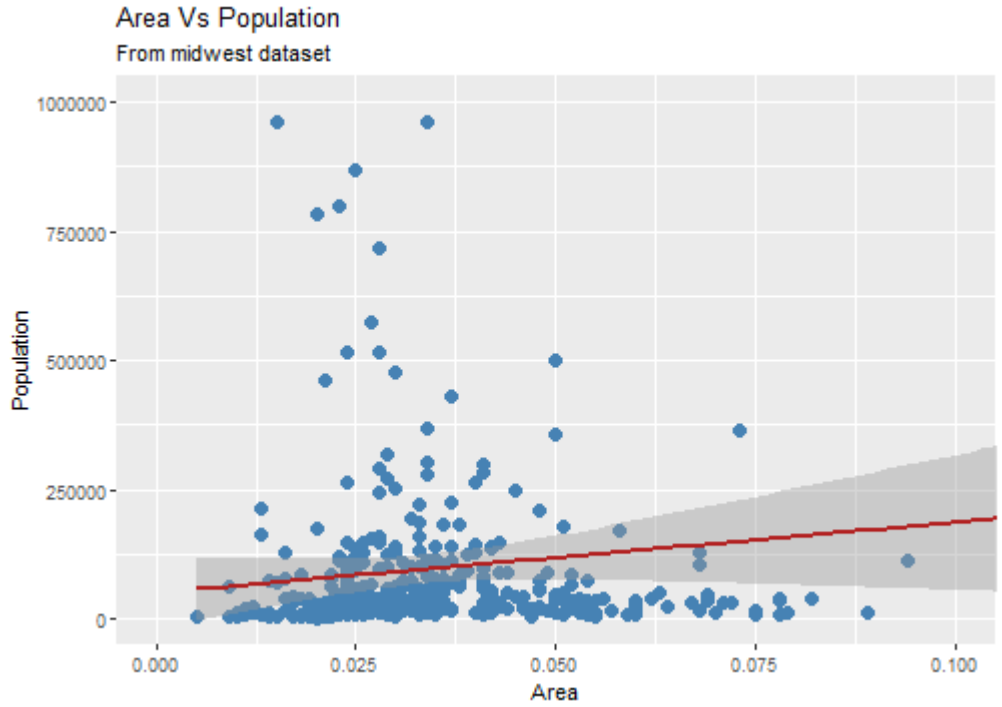
```
ggplot(midwest, aes(x = area, y = poptotal)) +
  geom_point() +
  geom_smooth(method = "lm") +
  coord_cartesian(xlim = c(0, 0.1), ylim = c(0, 1000000))+
  ggtitle("Area Vs Population", subtitle = "From midwest dataset") +
  xlab("Area") + ylab("Population")
```



© 2022 Zhenyuan LU

Color Changes

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



© 2022 Zhenyuan LU

Legend Removing

```
ggplot(midwest, aes(x = area, y = poptotal)) +
  geom_point(aes(color=state), size=3) +
  geom_smooth(method = "lm", col="firebrick") +
  coord_cartesian(xlim = c(0, 0.1), ylim = c(0, 1000000))+
  ggtitle("Area Vs Population", subtitle = "From midwest dataset") +
  xlab("Area") + ylab("Population") +
  theme(legend.position="None")
```

© 2022 Zhenyuan LU

Color Palette Changing

```
ggplot(midwest, aes(x = area, y = poptotal)) +
  geom_point(aes(color=state), size=3) +
  geom_smooth(method = "lm", col="firebrick") +
  coord_cartesian(xlim = c(0, 0.1), ylim = c(0, 1000000))+
  ggtitle("Area Vs Population", subtitle = "From midwest dataset") +
  xlab("Area") + ylab("Population") +
  theme(legend.position="None") +
  scale_colour_brewer(palette = "Set1")
```

© 2022 Zhenyuan LU

Show more color palette

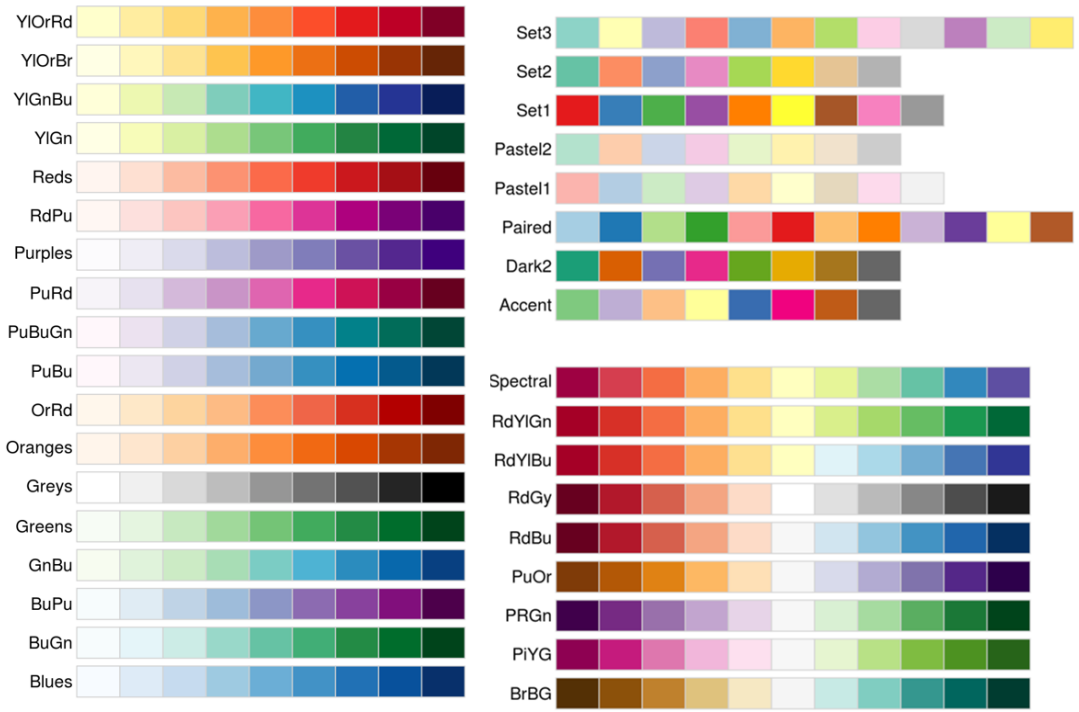
```
library(RColorBrewer)
head(brewer.pal.info, 10)
```

```
##          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
```

```
#Exp. scale_colour_brewer(palette = “BrBG")
```

© 2022 Zhenyuan LU

More color palette



© 2022 Zhenyuan LU

Change the Default Themes

```
ggplot(midwest, aes(x = area, y = poptotal)) +
  geom_point(aes(col=state), size=3) +
  geom_smooth(method = "lm", color="firebrick") +
  coord_cartesian(xlim = c(0, 0.1), ylim = c(0, 1000000))+
  ggtitle("Area Vs Population", subtitle = "From midwest dataset") +
  xlab("Area") + ylab("Population") +
  theme(legend.position="None") +
  scale_colour_brewer(palette = "Set1") +
  theme_bw() + labs(subtitle="BW Theme")
```

© 2022 Zhenyuan LU

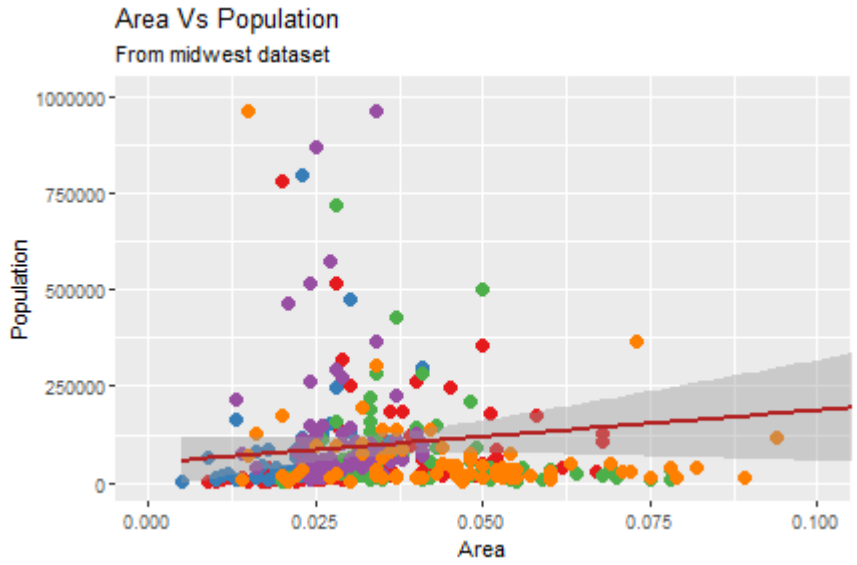
Change the Default Themes (cont'd)

```
ggplot(midwest, aes(x = area, y = poptotal)) +
  geom_point(aes(col=state), size=3) +
  geom_smooth(method = "lm", color="firebrick") +
  coord_cartesian(xlim = c(0, 0.1), ylim = c(0, 1000000))+
  ggtitle("Area Vs Population", subtitle = "From midwest dataset") +
  xlab("Area") + ylab("Population") +
  theme(legend.position="None") +
  scale_colour_brewer(palette = "Set1") +
  theme_classic() + labs(subtitle="Classic Theme")
```

© 2022 Zhenyuan LU

Default plot assigned

```
gg <- ggplot(midwest, aes(x = area, y = poptotal)) +
  geom_point(aes(col = state), size = 3) +
  geom_smooth(method = "lm", color = "firebrick") +
  coord_cartesian(xlim = c(0, 0.1), ylim = c(0, 1000000)) +
  ggtitle("Area Vs Population", subtitle = "From midwest dataset") +
  xlab("Area") + ylab("Population") +
  theme(legend.position = "None") +
  scale_colour_brewer(palette = "Set1")
plot(gg)
```



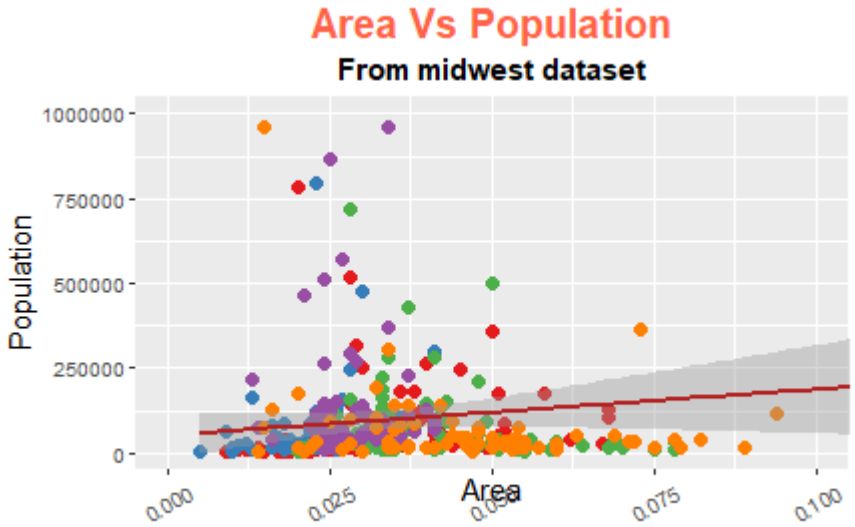
© 2022 Zhenyuan LU

Customizing Titles

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

© 2022 Zhenyuan LU

Customizing Titles (cont'd)



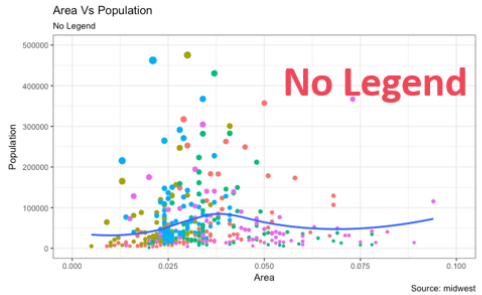
© 2022 Zhenyuan LU

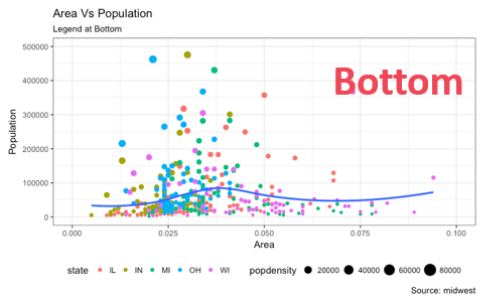
Legend Position

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

# Legend to the left
#labs(subtitle="Legend on the Left")

# legend at the bottom and horizontal
#labs(subtitle="Legend at Bottom")

# legend at bottom-right, inside the plot
#labs(subtitle="Legend: Bottom-Right Inside the Plot")

# legend at top-left, inside the plot
#labs(subtitle="Legend: Top-Left Inside the Plot")
```

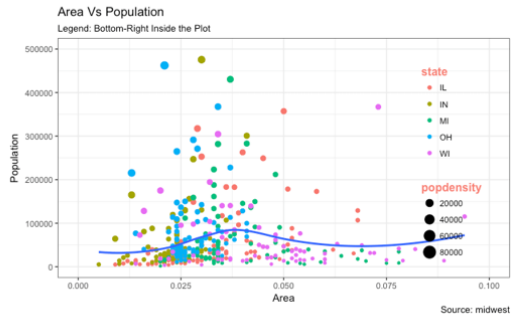
© 2022 Zhenyuan LU

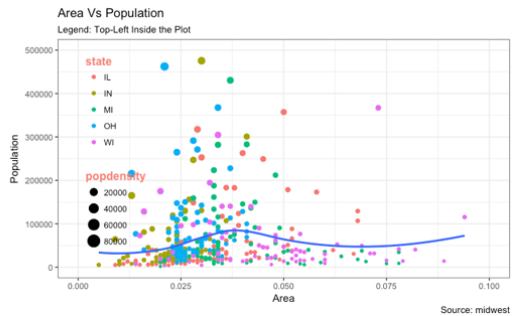
Legend Position (cont'd)





Inside, bottom right





Inside, top left

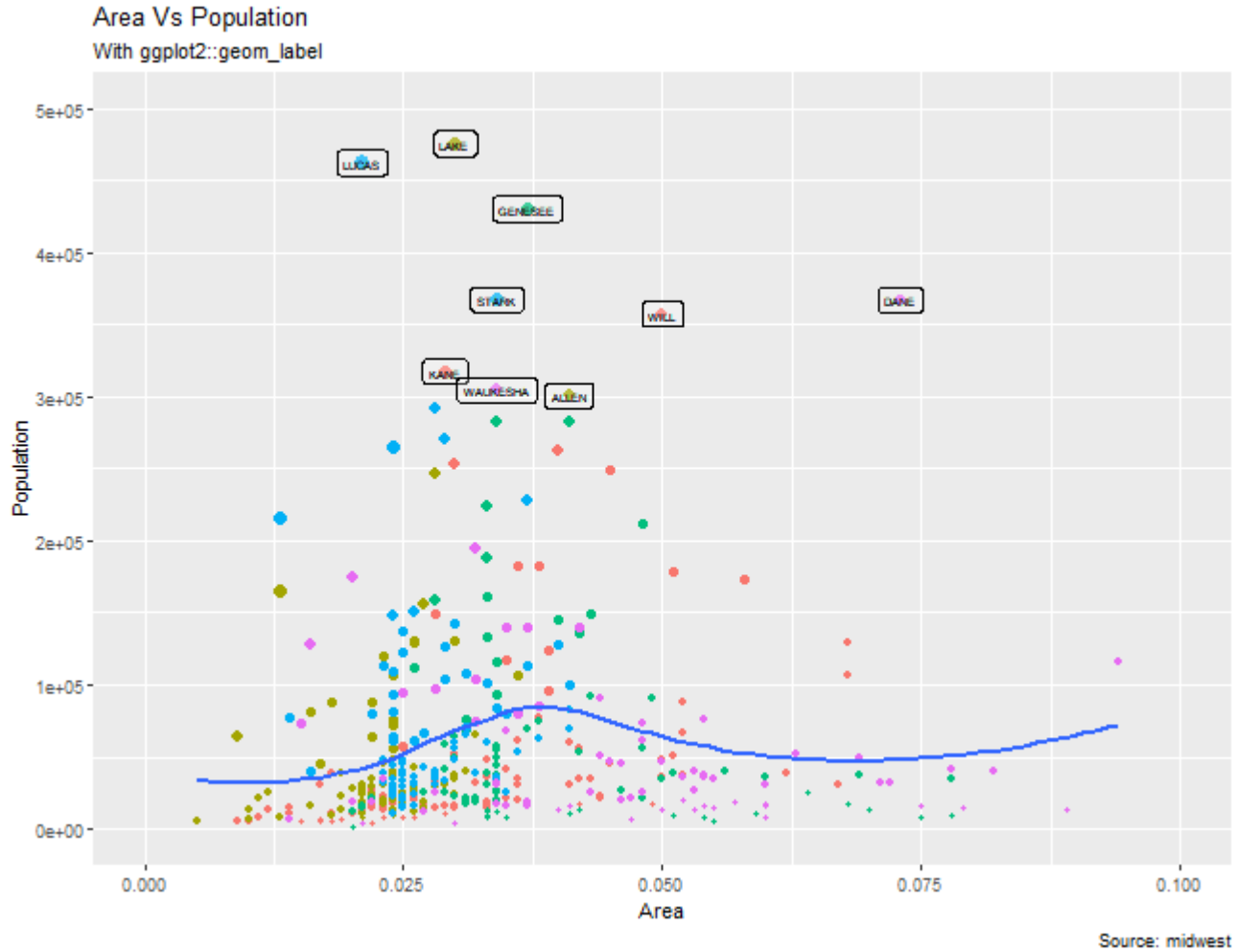
© 2022 Zhenyuan LU

Label, Text, and Annotation

```
# Filter required rows.
midwest_sub <- midwest %>% filter(poptotal > 300000) %>%
  mutate(large_county = ifelse(poptotal > 300000, county, ""))
# Base Plot
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 text
  geom_text(data = midwest_sub, aes(label = large_county), size = 2) +
  labs(subtitle = "With ggplot2::geom_text") + theme(legend.position = "None") +
  # label
  geom_label(
    data = midwest_sub,
    aes(label = large_county),
    size = 2, alpha = 0.25
  ) + labs(subtitle = "With ggplot2::geom_label") + theme(legend.position = "None")
```

© 2022 Zhenyuan LU

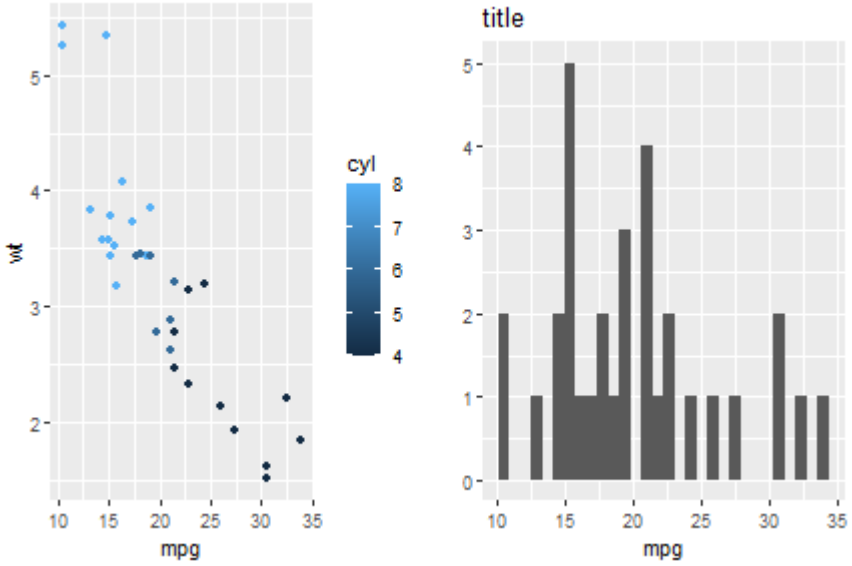
Label, Text, and Annotation



© 2022 Zhenyuan LU

Multiple plots

```
library(gridExtra)
p1 <- qplot(mpg, wt, data = mtcars, colour = cyl)
p2 <- qplot(mpg, data = mtcars) + ggtitle("title")
grid.arrange(p1, p2, nrow = 1)
```



© 2022 Zhenyuan LU

Charts

1.Correlation

1. Scatterplot
2. Scatterplot With Encircling
3. Jitter Plot
4. Counts Chart
5. Bubble Plot
6. Correlogram

2.Deviation

1. Diverging Bars
2. Diverging Lollipop Chart
3. Diverging Dot Plot
4. Area Chart

3.Ranking

1. Ordered Bar Chart
2. Lollipop Chart
3. Dot Plot
4. Slope Chart
5. Dumbbell Plot

4.Distribution

1. Histogram
2. Density Plot
3. Box Plot
4. Dot + Box Plot
5. Tufte Boxplot
6. Violin Plot
7. Population Pyramid

5.Change

1. Time Series Plots
2. Stacked Area Chart
3. Calendar Heat Map
4. Seasonal Plot
5. Heat Map

6.Groups

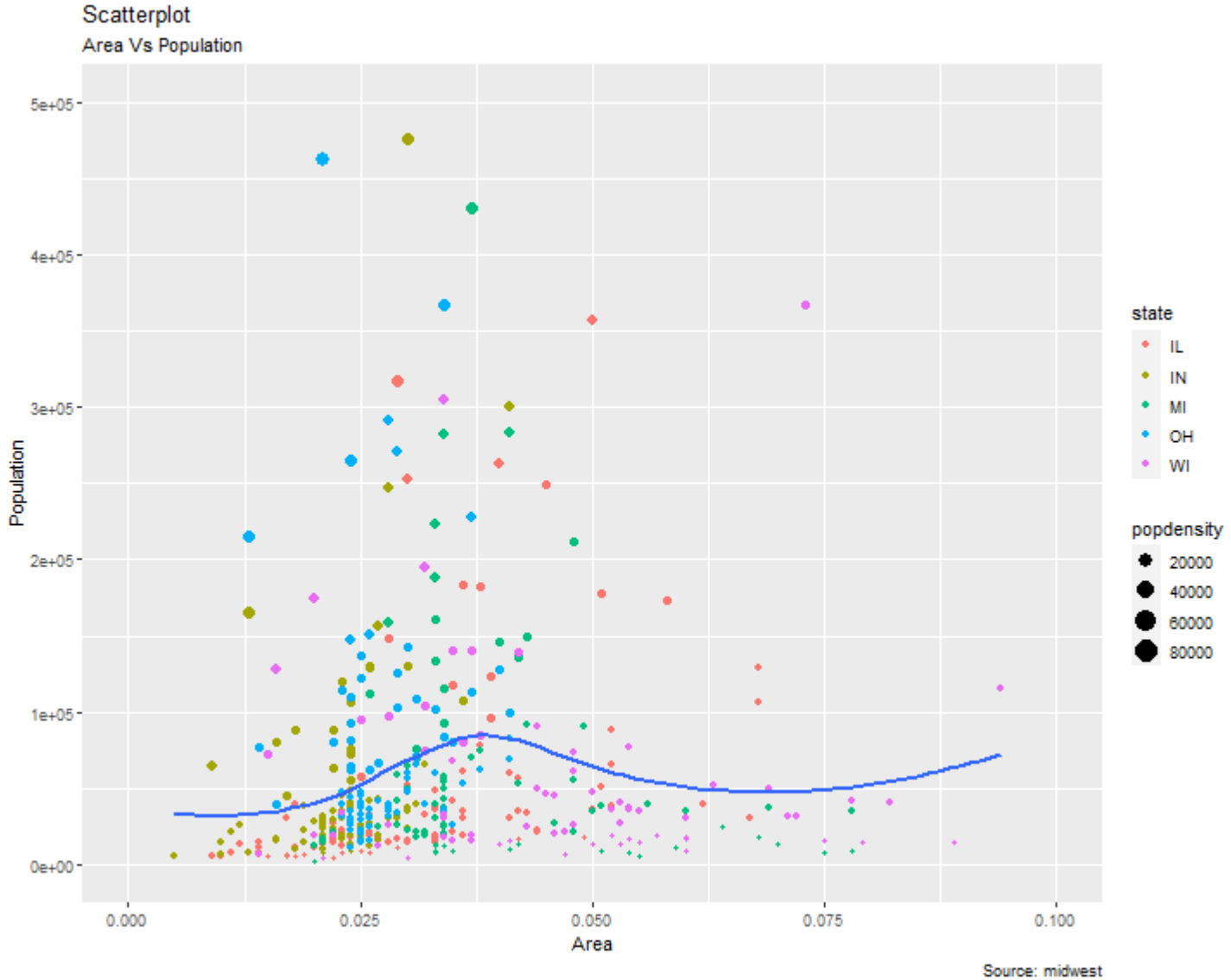
1. Dendrogram

© 2022 Zhenyuan LU

Correlation, scatterplot

```
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(subtitle="Area Vs Population",
       y="Population",
       x="Area",
       title="Scatterplot",
       caption = "Source: midwest")
```

© 2022 Zhenyuan LU

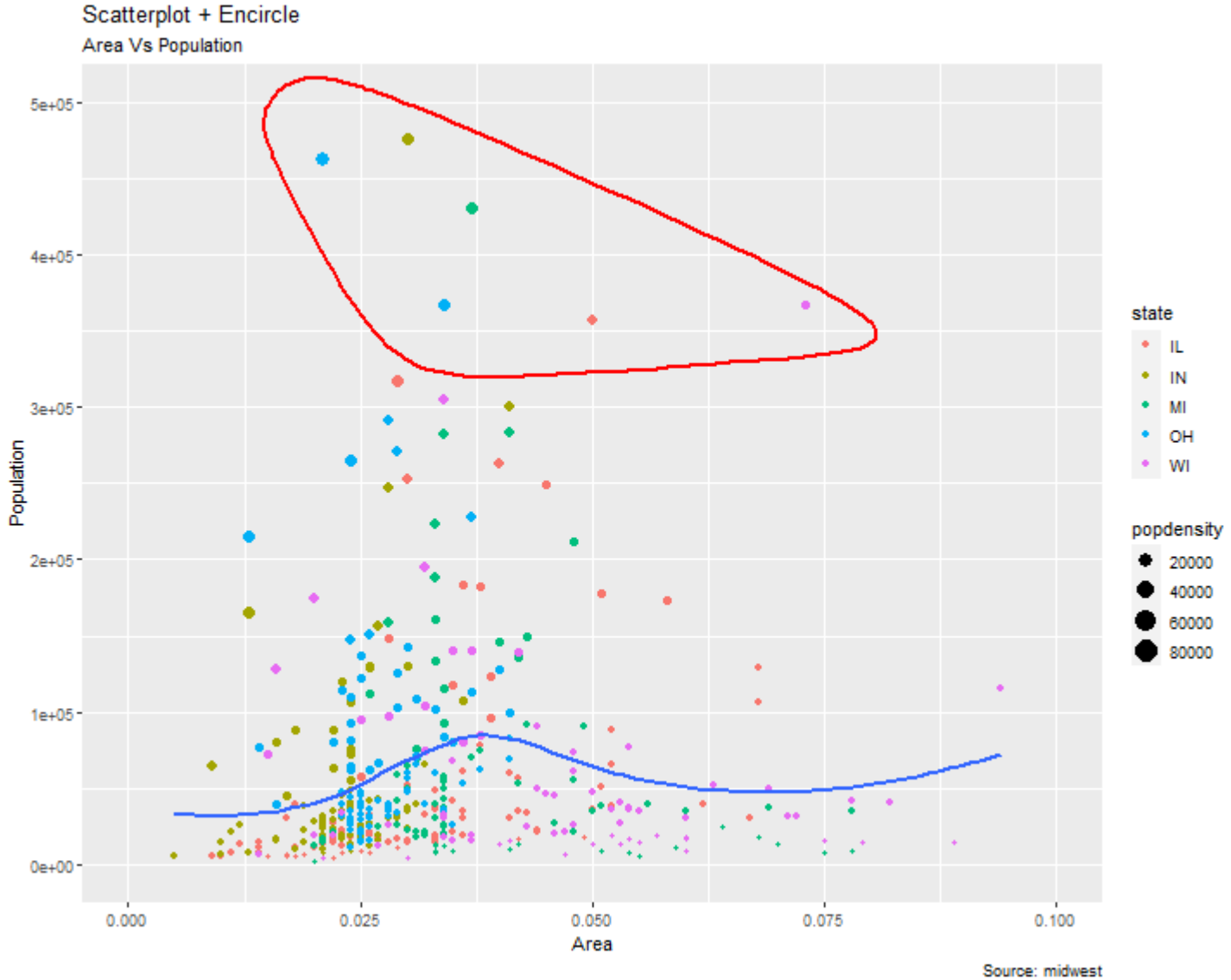


© 2022 Zhenyuan LU

Correlation, scatterplot with encircling

```
library(ggalt)
midwest_select <- midwest[midwest$poptotal > 350000 &
                          midwest$poptotal <= 500000 &
                          midwest$area > 0.01 &
                          midwest$area < 0.1,]
# Plot
ggplot(midwest, aes(x = area, y = poptotal)) +
  geom_point(aes(col = state, size = popdensity)) +   # draw points
  geom_smooth(method = "loess", se = F) +
  xlim(c(0, 0.1)) +
  ylim(c(0, 500000)) +   # draw smoothing line
  geom_encircle(
    aes(x = area, y = poptotal),
    data = midwest_select,
    color = "red",
    size = 2,
    expand = 0.08
  ) +   # encircle
  labs( subtitle = "Area Vs Population",
    y = "Population",
    x = "Area",
    title = "Scatterplot + Encircle",
    caption = "Source: midwest"
  )
```

© 2022 Zhenyuan LU

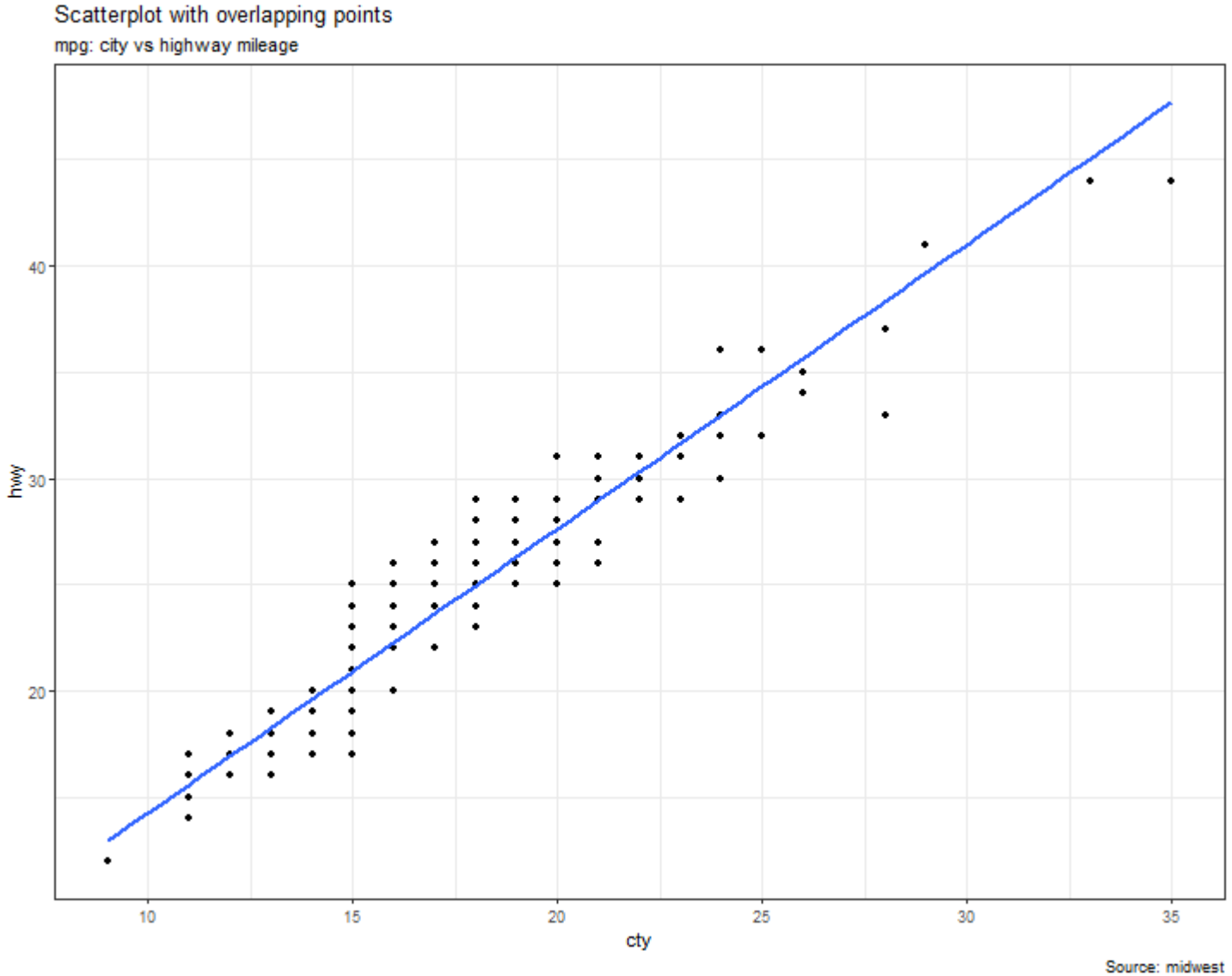


© 2022 Zhenyuan LU

Correlation, jitter plot

```
ggplot(mpg, aes(cty, hwy)) + geom_point() +
  geom_smooth(method = "lm", se = F) +
  labs(
    subtitle = "mpg: city vs highway mileage",
    y = "hwy",
    x = "cty",
    title = "Scatterplot with overlapping points",
    caption = "Source: midwest"
  ) +
  theme_bw()
```

© 2022 Zhenyuan LU

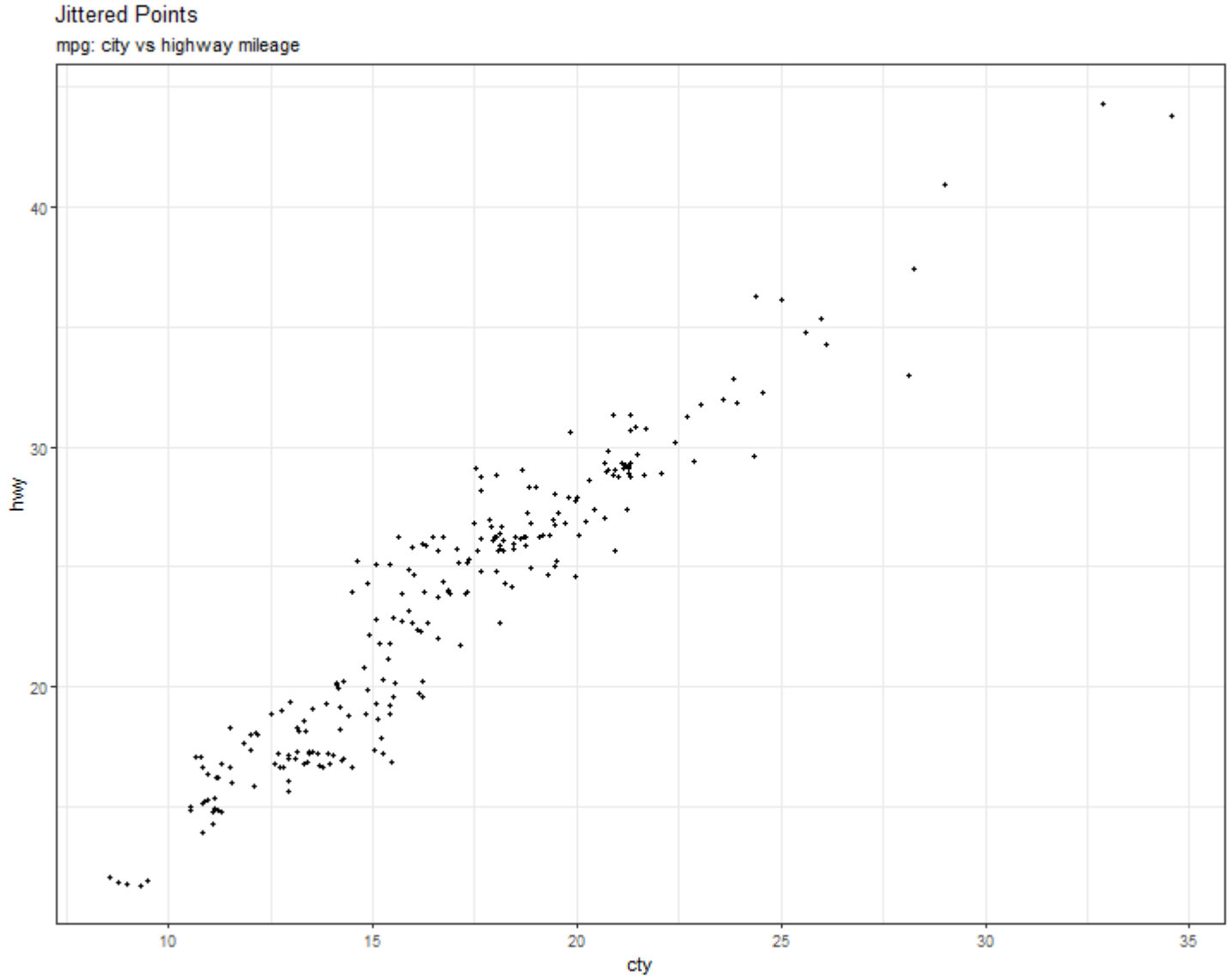


© 2022 Zhenyuan LU

Correlation, jitter plot

```
ggplot(mpg, aes(cty, hwy)) +
  geom_jitter(width = .5, size = 1) +
  labs(
    subtitle = "mpg: city vs highway mileage",
    y = "hwy",
    x = "cty",
    title = "Jittered Points"
  )+
  theme_bw()
```

© 2022 Zhenyuan LU

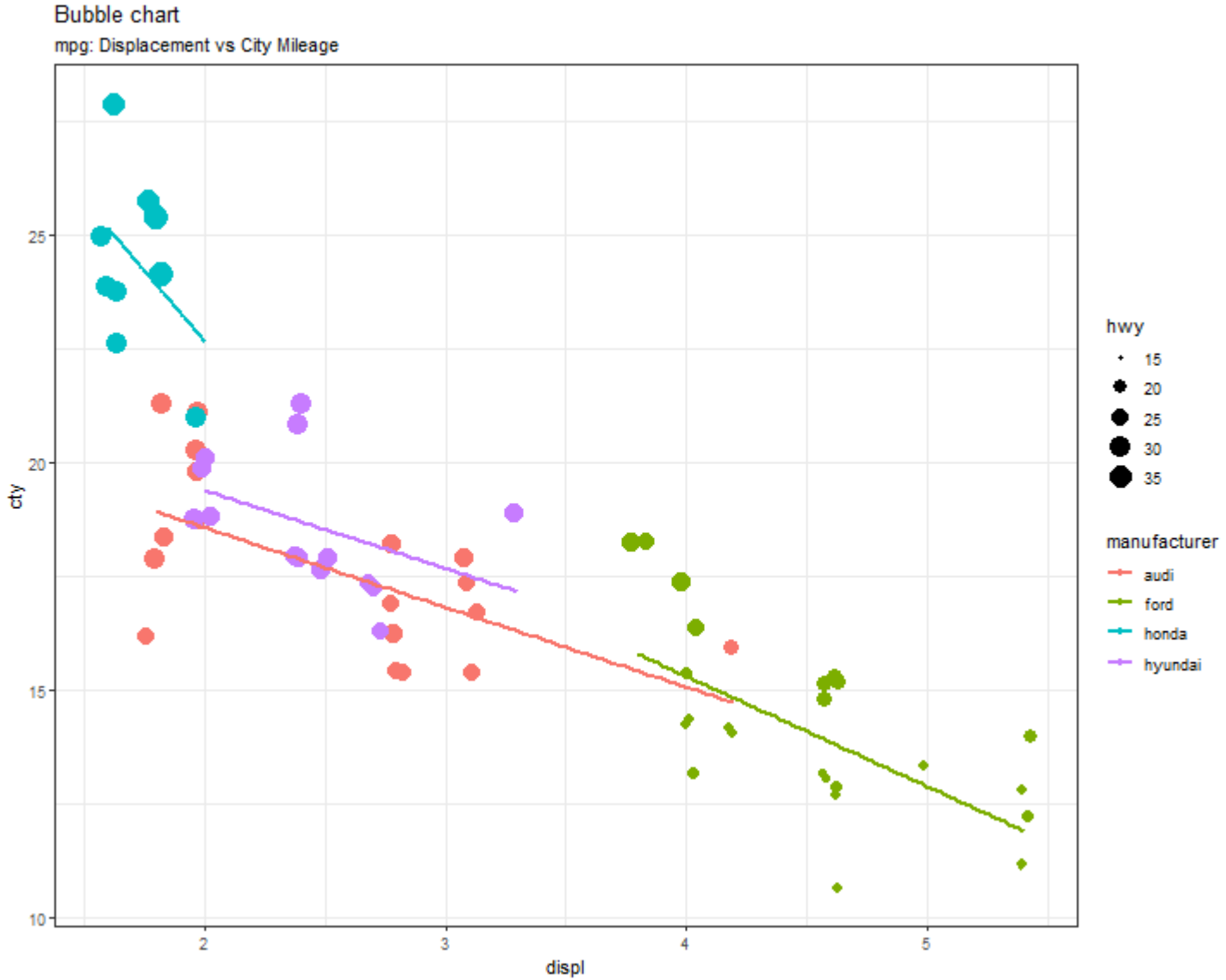


© 2022 Zhenyuan LU

Correlation, bubble plot

```
mpg %>%
  filter(manufacturer %in% c("audi", "ford", "honda", "hyundai")) %>%
  ggplot(aes(displ, cty)) +
  labs(subtitle = "mpg: Displacement vs City Mileage",
  title = "Bubble chart") +
  geom_jitter(aes(col = manufacturer, size = hwy)) +
  geom_smooth(aes(col = manufacturer), method = "lm", se = F) +
  theme_bw()
```

© 2022 Zhenyuan LU



© 2022 Zhenyuan LU

Correlation, correlogram

```
library(ggcorrplot)
corr <- round(cor(mtcars), 1)
ggcorrplot(corr, hc.order = TRUE,
           type = "lower",
           lab = TRUE,
           lab_size = 3,
           method="circle",
           colors = c("tomato2", "white", "springgreen3"),
           title="Correlogram of mtcars",
           ggtheme=theme_bw)
```

© 2022 Zhenyuan LU

```
## Error in library(ggcorrplot): there is no package called 'ggcorrplot'
```

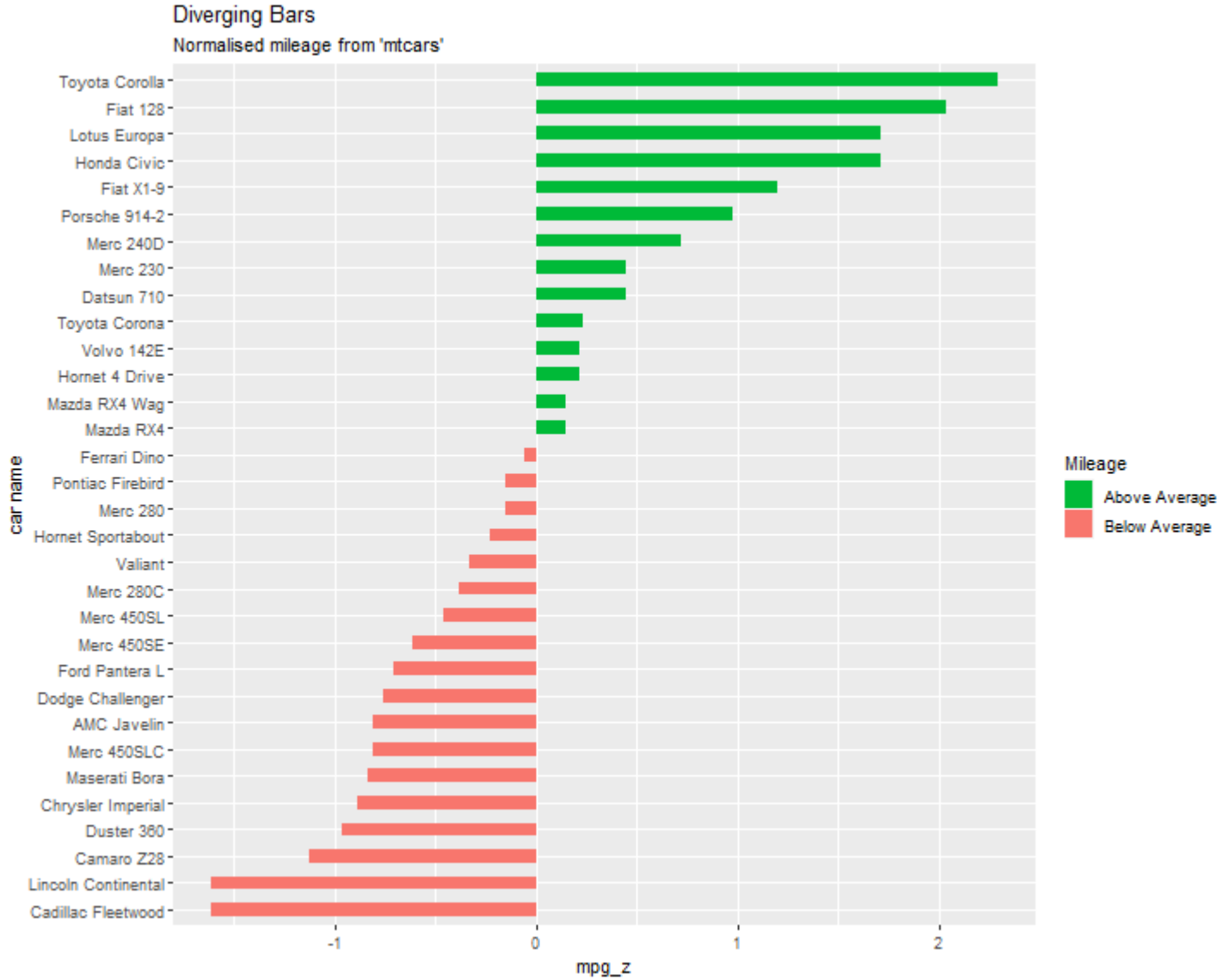
```
## Error in ggcorrplot(corr, hc.order = TRUE, type = "lower", lab = TRUE, : could not find function "ggcorrplot"
```

© 2022 Zhenyuan LU

Deviation, diverging bar

```
# Data Prep
data("mtcars")  # load data
mtcars$`car name` <- rownames(mtcars)  # create new column for car names
mtcars$mpg_z <- round((mtcars$mpg - mean(mtcars$mpg))/sd(mtcars$mpg), 2)  # compute normalized mpg
mtcars$mpg_type <- ifelse(mtcars$mpg_z < 0, "below", "above")  # above / below avg flag
mtcars <- mtcars[order(mtcars$mpg_z), ]  # sort
mtcars$`car name` <- factor(mtcars$`car name`, levels = mtcars$`car name`)  # convert to factor to retain sor

# Diverging Barcharts
ggplot(mtcars, aes(x=`car name`, y=mpg_z, label=mpg_z)) +
  geom_bar(stat='identity', aes(fill=mpg_type), width=.5)  +
  scale_fill_manual(name="Mileage",
                    labels = c("Above Average", "Below Average"),
                    values = c("above"="#00ba38", "below"="#f8766d")) +
  labs(subtitle="Normalised mileage from 'mtcars'",
       title= "Diverging Bars") +
  coord_flip()
```

© 2022 Zhenyuan LU

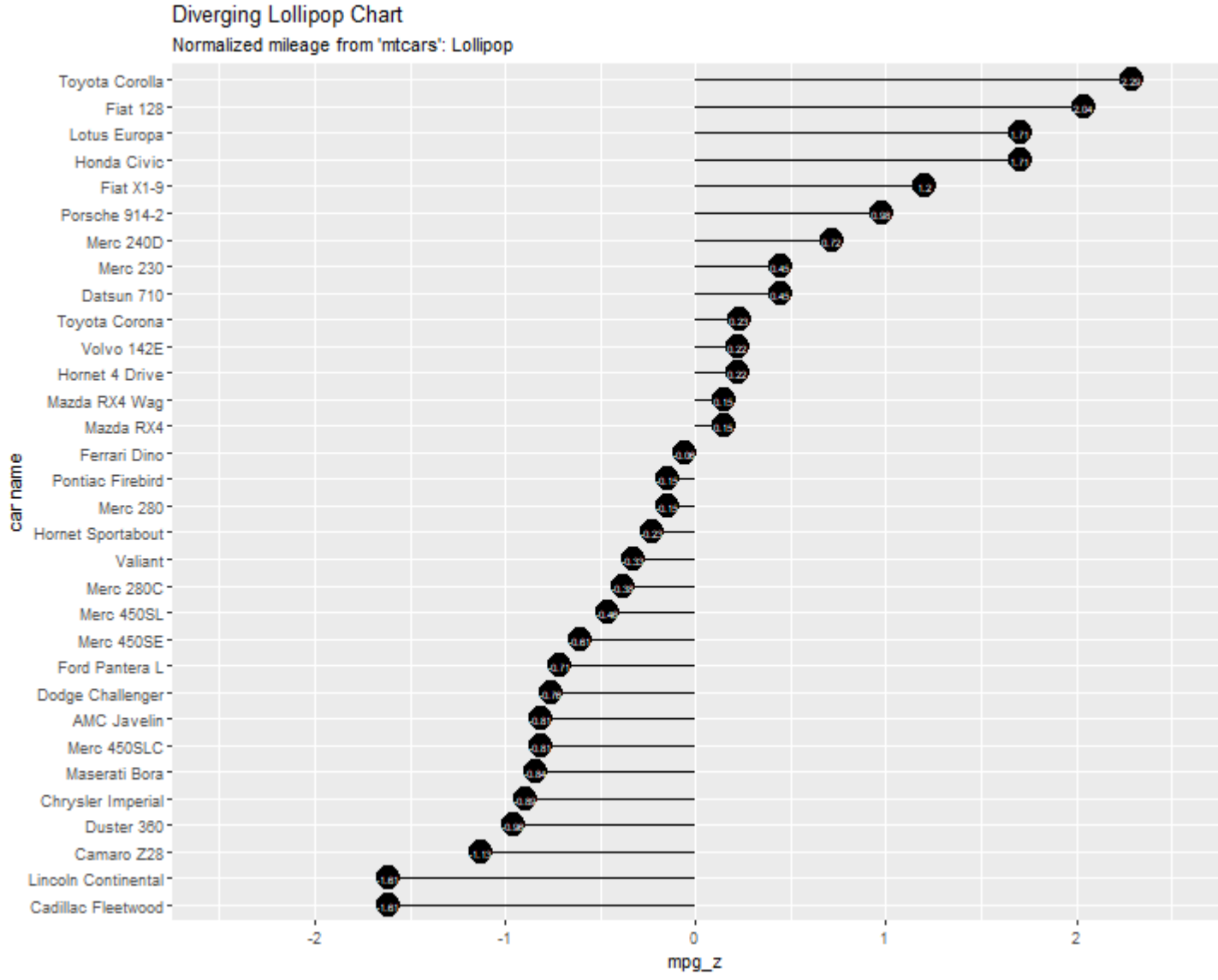


© 2022 Zhenyuan LU

Deviation, diverging lollipop bar

```
ggplot(mtcars, aes(x=`car name`, y=mpg_z, label=mpg_z)) +
  geom_point(stat='identity', fill="black", size=6)  +
  geom_segment(aes(y = 0,
                   x = `car name`,
                   yend = mpg_z,
                   xend = `car name`),
               color = "black") +
  geom_text(color="white", size=2) +
  labs(title="Diverging Lollipop Chart",
       subtitle="Normalized mileage from 'mtcars': Lollipop") +
  ylim(-2.5, 2.5) +
  coord_flip()
```

© 2022 Zhenyuan LU

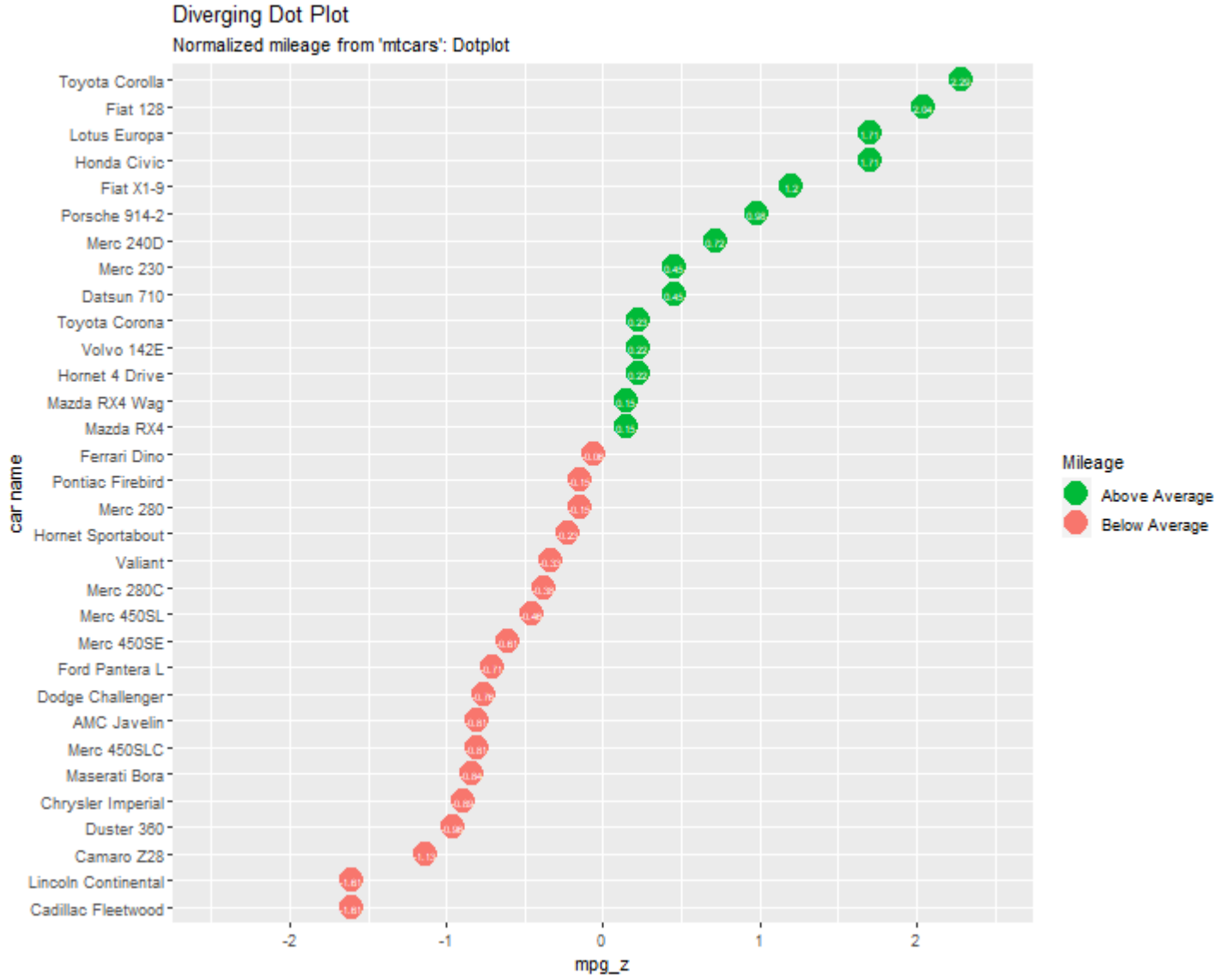


© 2022 Zhenyuan LU

Deviation, diverging lollipop bar

```
ggplot(mtcars, aes(x=`car name`, y=mpg_z, label=mpg_z)) +
  geom_point(stat='identity', aes(col=mpg_type), size=6)  +
  scale_color_manual(name="Mileage",
                     labels = c("Above Average", "Below Average"),
                     values = c("above"="#00ba38", "below"="#f8766d")) +
  geom_text(color="white", size=2) +
  labs(title="Diverging Dot Plot",
       subtitle="Normalized mileage from 'mtcars': Dotplot") +
  ylim(-2.5, 2.5) +
  coord_flip()
```

© 2022 Zhenyuan LU

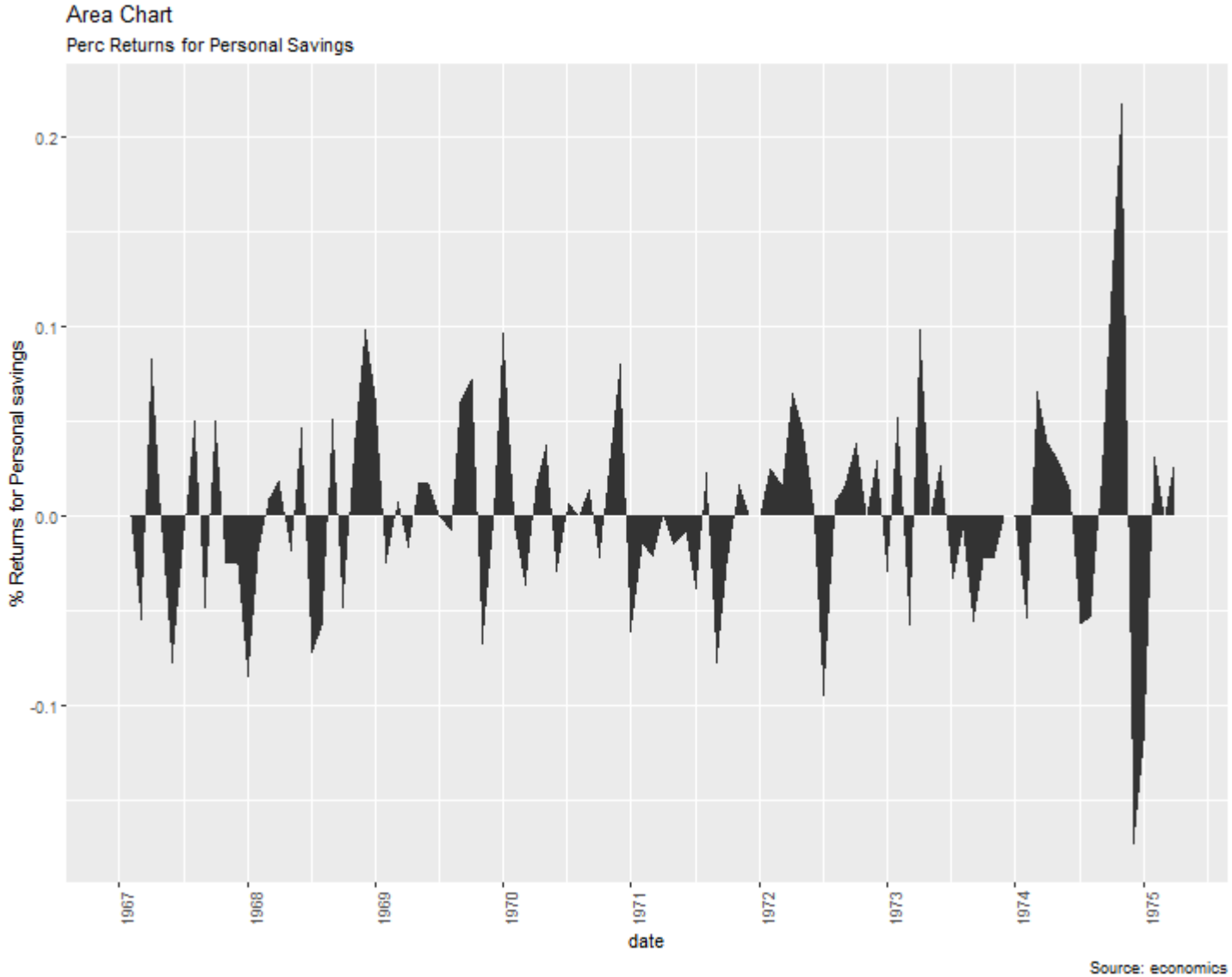


© 2022 Zhenyuan LU

Deviation, area chart

```
economics$returns_perc <- c(0, diff(economics$psavert)/economics$psavert[-length(economics$psavert)])

# Create break points and labels for axis ticks
brks <- economics$date[seq(1, length(economics$date), 12)]
lbls <- lubridate::year(economics$date[seq(1, length(economics$date), 12)])

# Plot
ggplot(economics[1:100, ], aes(date, returns_perc)) +
  geom_area() +
  scale_x_date(breaks=brks, labels=lbls) +
  theme(axis.text.x = element_text(angle=90)) +
  labs(title="Area Chart",
       subtitle = "Perc Returns for Personal Savings",
       y="% Returns for Personal savings",
       caption="Source: economics")
```

© 2022 Zhenyuan LU

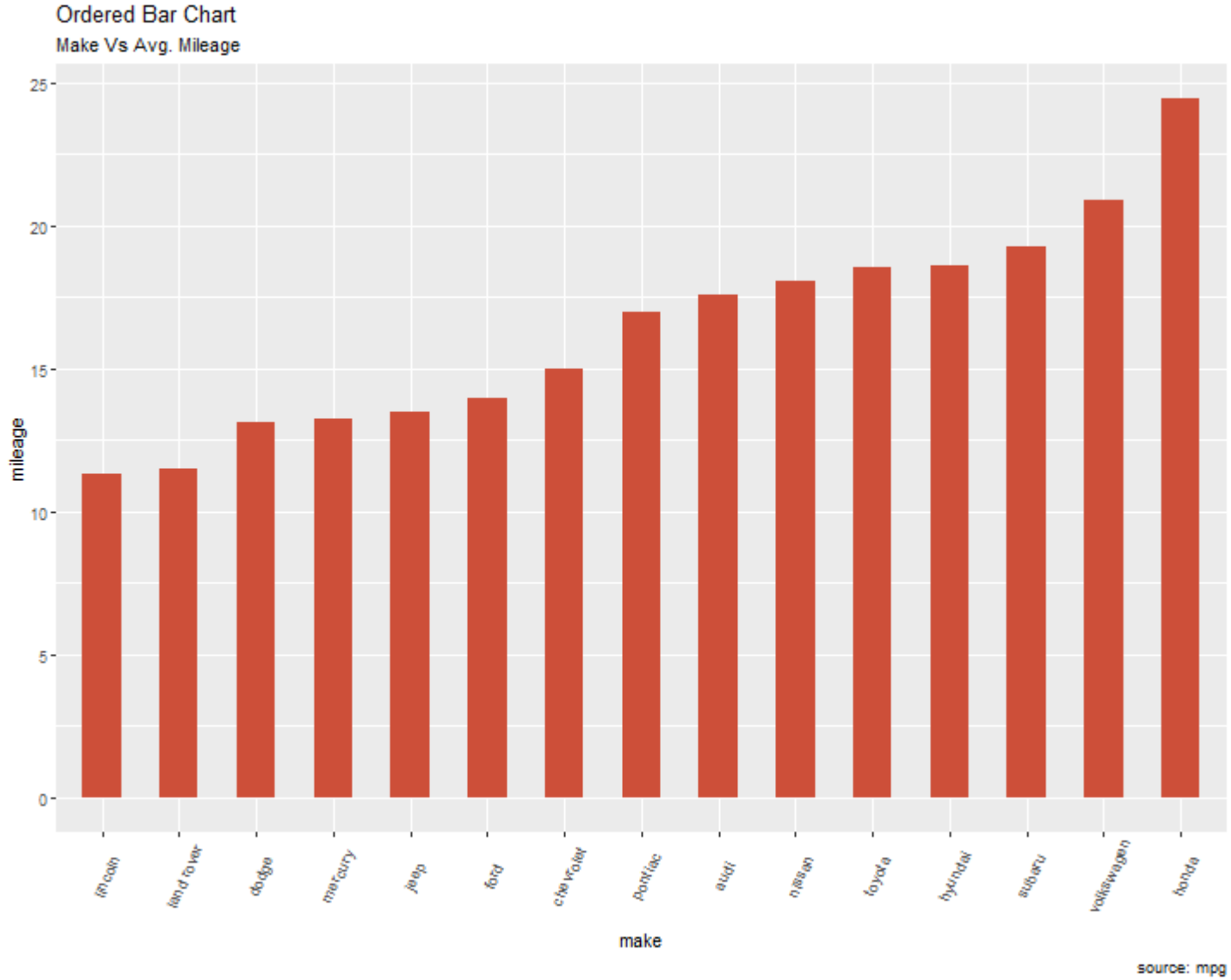


© 2022 Zhenyuan LU

Ranking, ordered bar chart

```
# Prepare data: group mean city mileage by manufacturer.
cty_mpg <- aggregate(mpg$cty, by=list(mpg$manufacturer), FUN=mean)  # aggregate
colnames(cty_mpg) <- c("make", "mileage")  # change column names
cty_mpg <- cty_mpg[order(cty_mpg$mileage), ]  # sort
cty_mpg$make <- factor(cty_mpg$make, levels = cty_mpg$make)  # to retain the order in plot.
ggplot(cty_mpg, aes(x=make, y=mileage)) +
  geom_bar(stat="identity", width=.5, fill="tomato3") +
  labs(title="Ordered Bar Chart",
       subtitle="Make Vs Avg. Mileage",
       caption="source: mpg") +
  theme(axis.text.x = element_text(angle=65, vjust=0.6))
```

© 2022 Zhenyuan LU

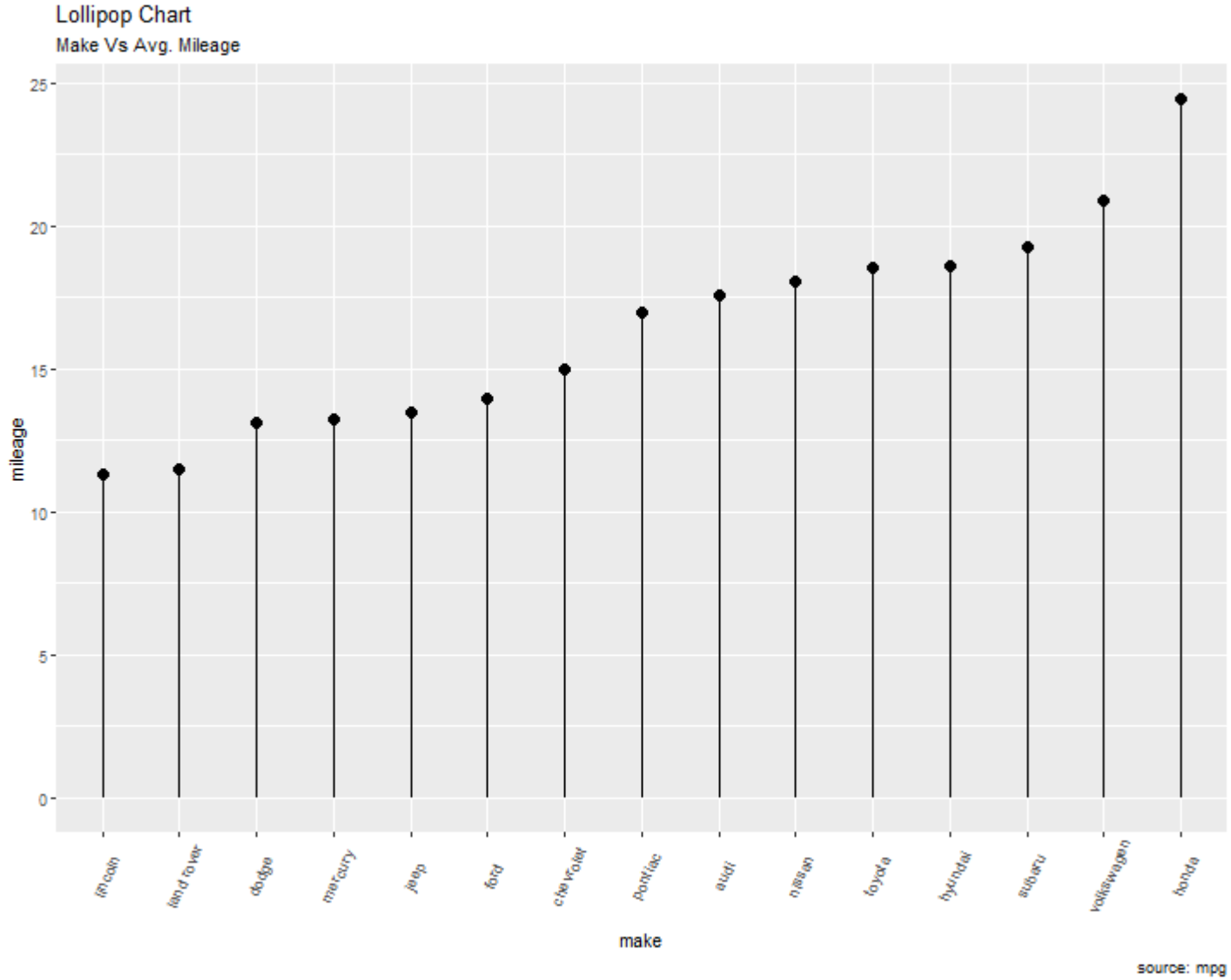


© 2022 Zhenyuan LU

Ranking, lollipop chart

```
ggplot(cty_mpg, aes(x=make, y=mileage)) +
  geom_point(size=3) +
  geom_segment(aes(x=make,
                   xend=make,
                   y=0,
                   yend=mileage)) +
  labs(title="Lollipop Chart",
       subtitle="Make Vs Avg. Mileage",
       caption="source: mpg") +
  theme(axis.text.x = element_text(angle=65, vjust=0.6))
```

© 2022 Zhenyuan LU

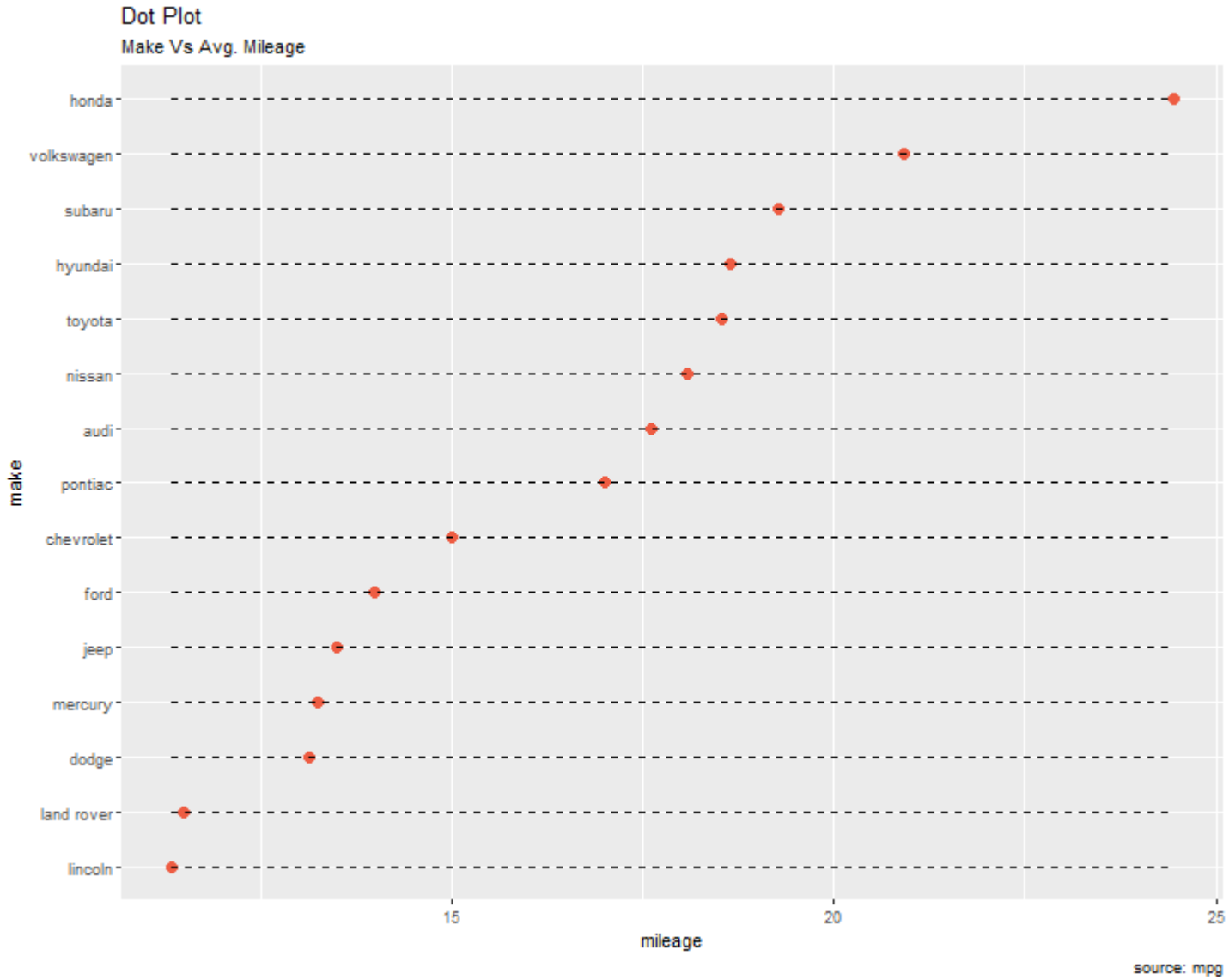


© 2022 Zhenyuan LU

Ranking, dot plot

```
# Plot
ggplot(cty_mpg, aes(x=make, y=mileage)) +
  geom_point(col="tomato2", size=3) +   # Draw points
  geom_segment(aes(x=make,
                   xend=make,
                   y=min(mileage),
                   yend=max(mileage)),
               linetype="dashed",
               size=0.1) +   # Draw dashed lines
  labs(title="Dot Plot",
       subtitle="Make Vs Avg. Mileage",
       caption="source: mpg") +
  coord_flip()
```

© 2022 Zhenyuan LU

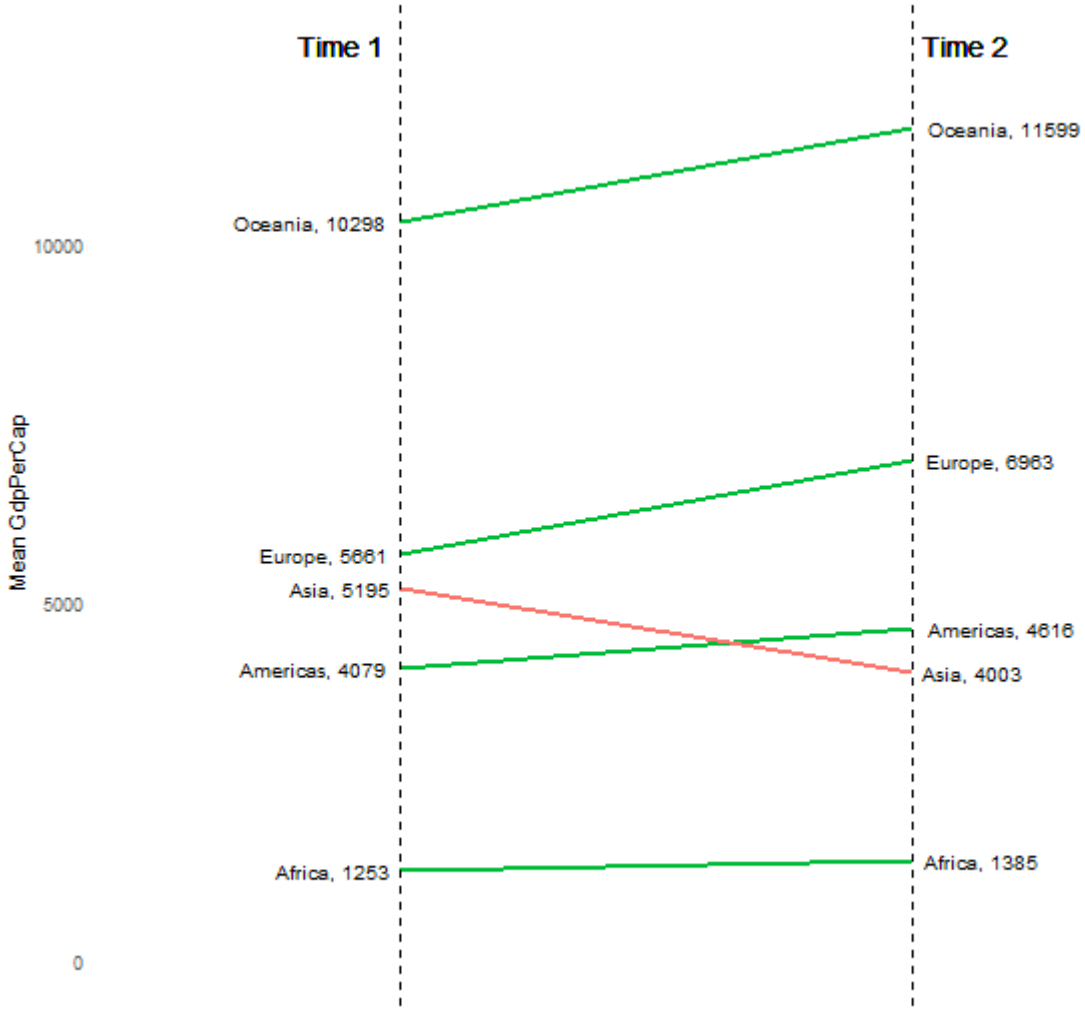


© 2022 Zhenyuan LU

Ranking, slop chart

```
# prep data
df <- read.csv("https://raw.githubusercontent.com/selva86/datasets/master/gdppercap.csv")
colnames(df) <- c("continent", "1952", "1957")
left_label <- paste(df$continent, round(df$`1952`),sep=", ")
right_label <- paste(df$continent, round(df$`1957`),sep=", ")
df$class <- ifelse((df$`1957` - df$`1952`) < 0, "red", "green")

# Plot
p <- ggplot(df) + geom_segment(aes(x=1, xend=2, y=`1952`, yend=`1957`, col=class), size=.75, show.legend=F) +
                  geom_vline(xintercept=1, linetype="dashed", size=.1) +
                  geom_vline(xintercept=2, linetype="dashed", size=.1) +
                  scale_color_manual(labels = c("Up", "Down"),
                                     values = c("green"="#00ba38", "red"="#f8766d")) +  # color of lines
                  labs(x="", y="Mean GdpPerCap") +  # Axis labels
                  xlim(.5, 2.5) + ylim(0,(1.1*(max(df$`1952`, df$`1957`))))  # X and Y axis limits

# Add texts
p <- p + geom_text(label=left_label, y=df$`1952`, x=rep(1, NROW(df)), hjust=1.1, size=3.5)
p <- p + geom_text(label=right_label, y=df$`1957`, x=rep(2, NROW(df)), hjust=-0.1, size=3.5)
p <- p + geom_text(label="Time 1", x=1, y=1.1*(max(df$`1952`, df$`1957`)), hjust=1.2, size=5)  # title
p <- p + geom_text(label="Time 2", x=2, y=1.1*(max(df$`1952`, df$`1957`)), hjust=-0.1, size=5)  # title

# Minify theme
p + theme(panel.background = element_blank(), panel.grid = element_blank(),
```

© 2022 Zhenyuan LU

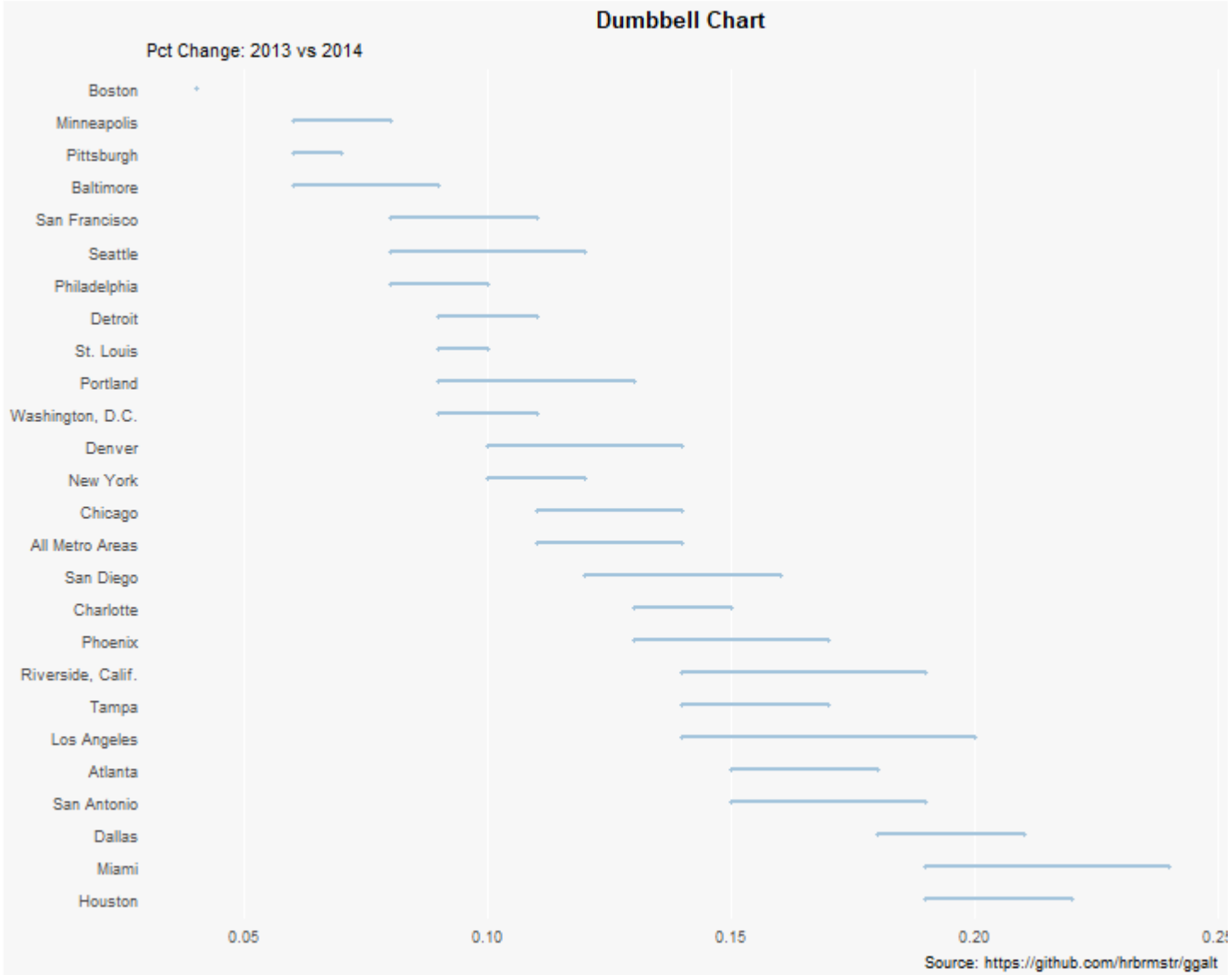


© 2022 Zhenyuan LU

Ranking, dumbbell

```
library(ggalt)
health <-
  read.csv("https://raw.githubusercontent.com/selva86/datasets/master/health.csv")
health$Area <-
  factor(health$Area, levels = as.character(health$Area))  # for right ordering of the dumbells
ggplot(health, aes(
  x = pct_2013,
  xend = pct_2014,
  y = Area,
  group = Area
)) +
  geom_dumbbell(color = "#a3c4dc",
                size = 0.75,
                point.colour.l = "#0e668b") +
  labs(
    x = NULL, y = NULL,
    title = "Dumbbell Chart",
    subtitle = "Pct Change: 2013 vs 2014",
    caption = "Source: https://github.com/hrbrmstr/ggalt"
  ) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    plot.background = element_rect(fill = "#f7f7f7"),
    panel.background = element_rect(fill = "#f7f7f7"),
```

© 2022 Zhenyuan Lu

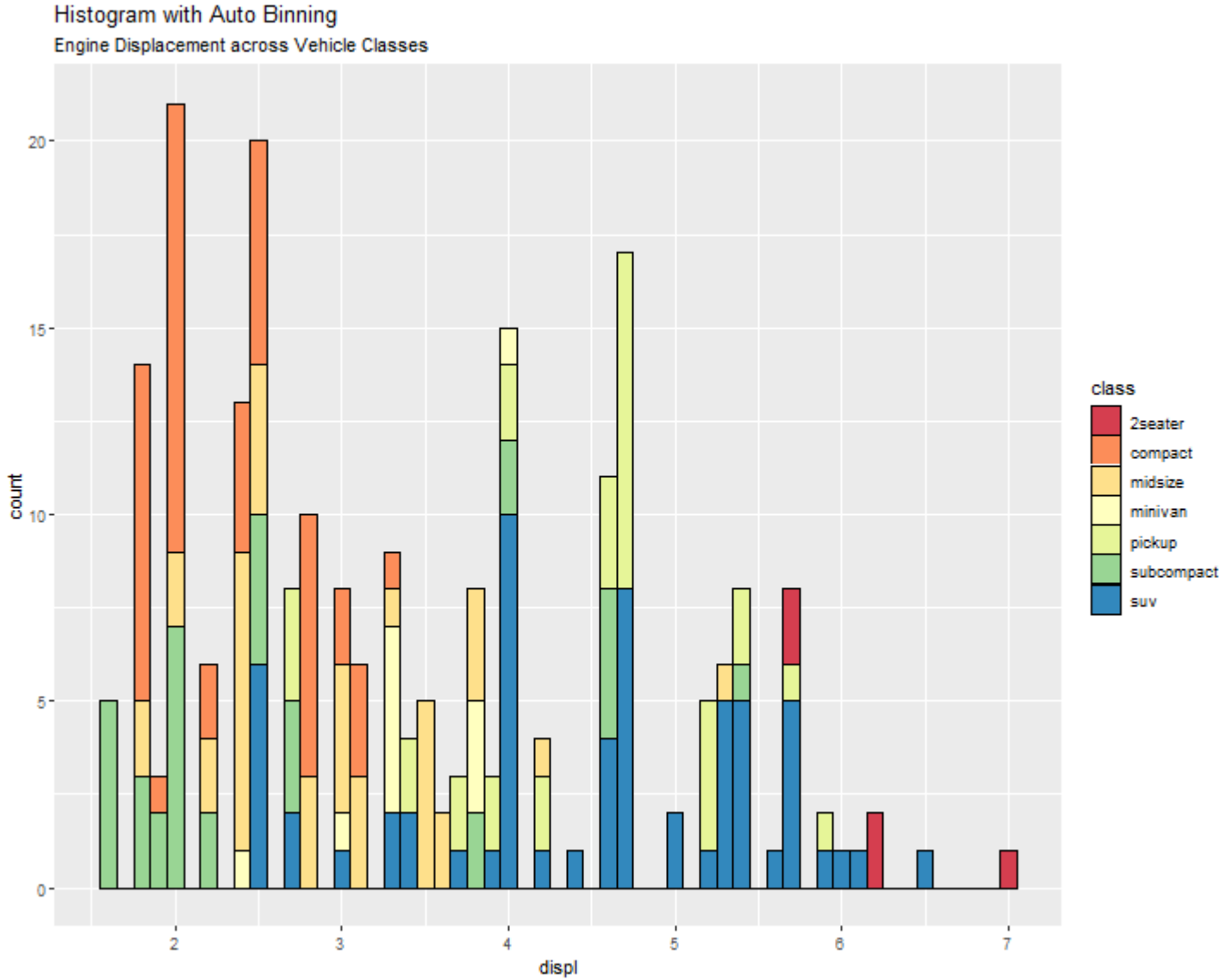


© 2022 Zhenyuan LU

Distribution, histogram

```
# Histogram on a Continuous (Numeric) Variable
g <- ggplot(mpg, aes(displ)) + scale_fill_brewer(palette = "Spectral")

g + geom_histogram(aes(fill=class),
                   binwidth = .1,
                   col="black",
                   size=.1) +  # change binwidth
  labs(title="Histogram with Auto Binning",
       subtitle="Engine Displacement across Vehicle Classes")
```

© 2022 Zhenyuan LU

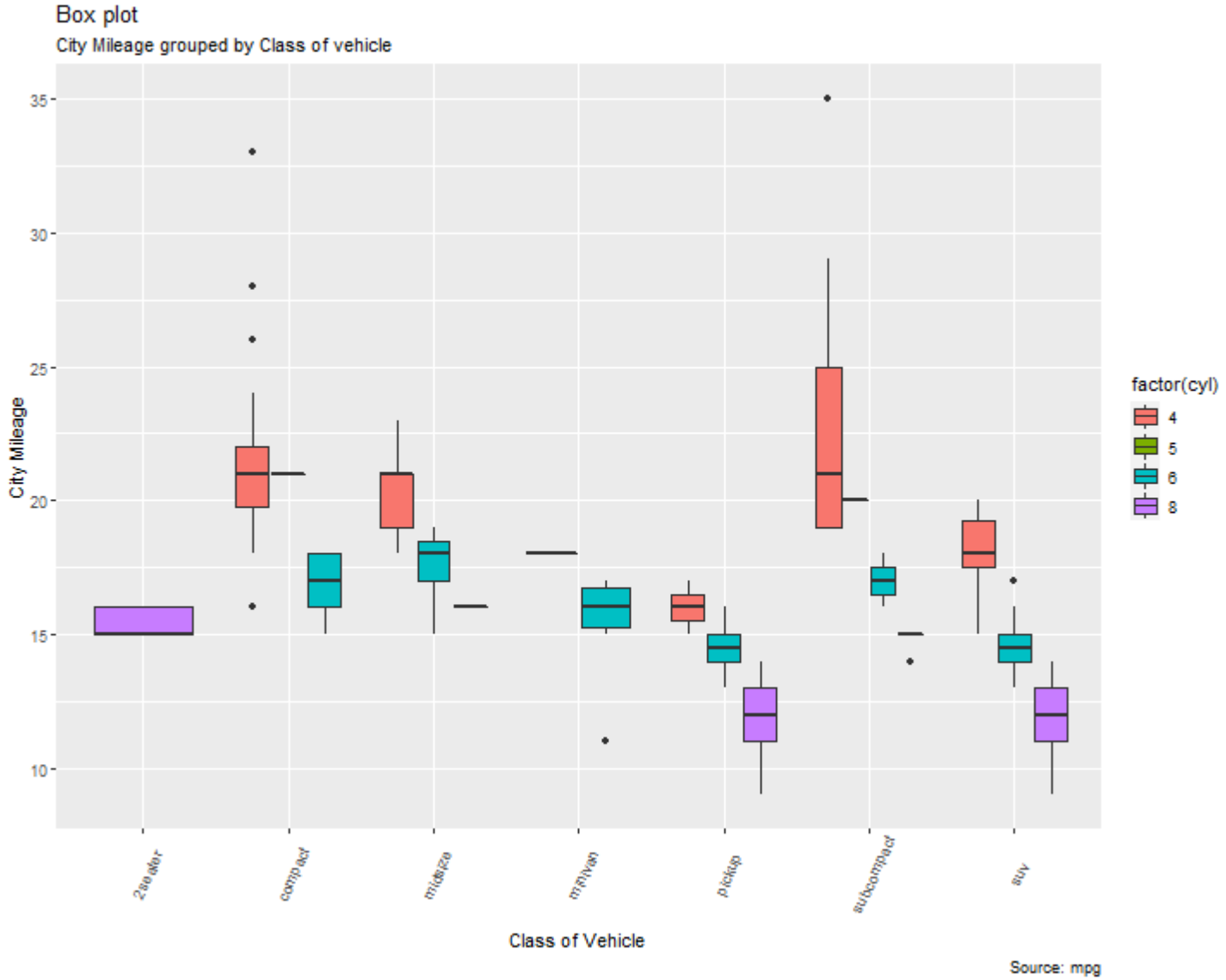


© 2022 Zhenyuan LU

Distribution, boxplot

```
library(ggthemes)
g <- ggplot(mpg, aes(class, cty))
g + geom_boxplot(aes(fill=factor(cyl))) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Box plot",
       subtitle="City Mileage grouped by Class of vehicle",
       caption="Source: mpg",
       x="Class of Vehicle",
       y="City Mileage")
```

© 2022 Zhenyuan LU

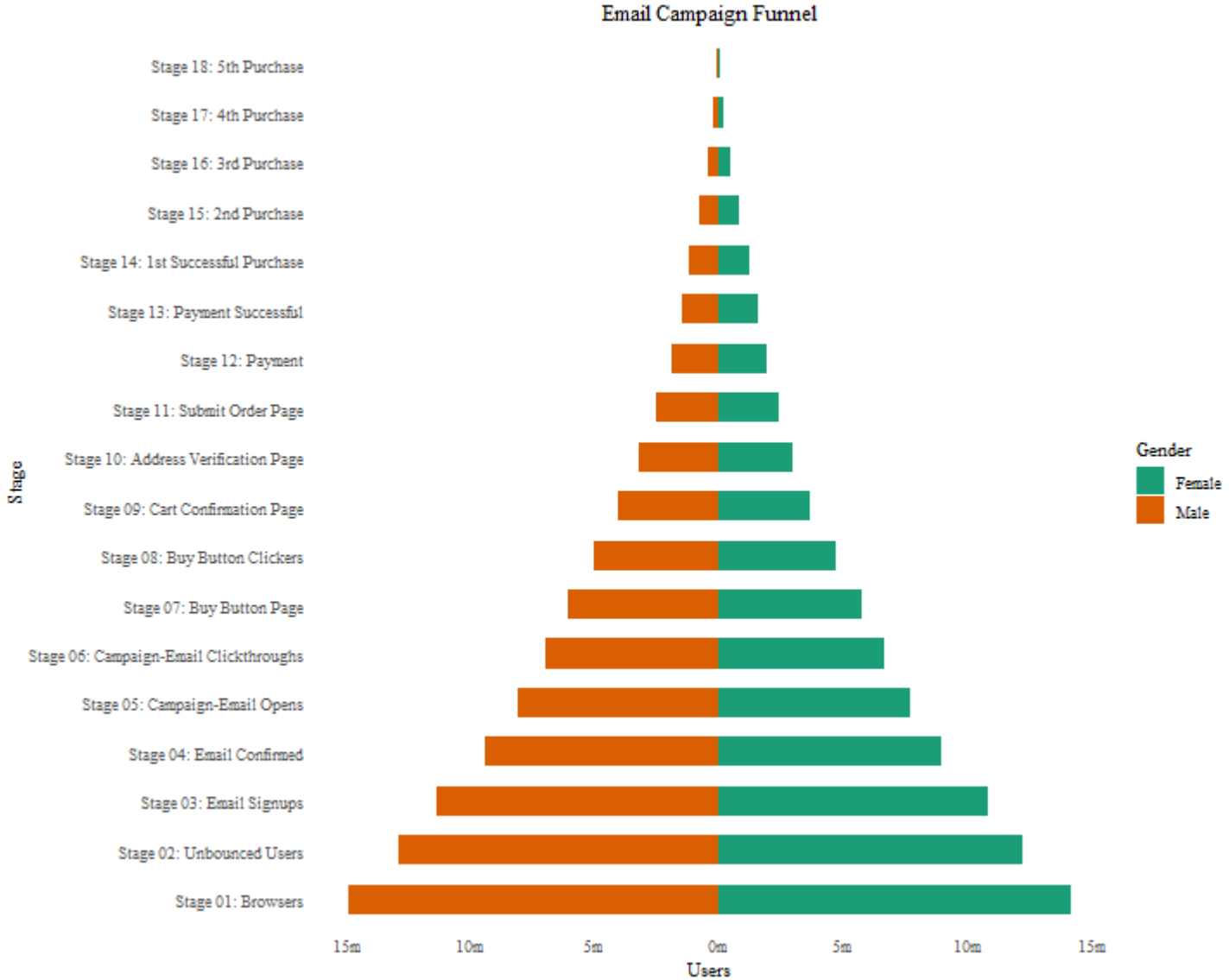


© 2022 Zhenyuan LU

Distribution, population pyramid

```r
library(ggplot2)
library(ggthemes)
# Read data
email_campaign_funnel <- read.csv("https://raw.githubusercontent.com/selva86/datasets/master/email_campaign_f

# X Axis Breaks and Labels
brks <- seq(-15000000, 15000000, 5000000)
lbls = paste0(as.character(c(seq(15, 0, -5), seq(5, 15, 5))), "m")

# Plot
ggplot(email_campaign_funnel, aes(x = Stage, y = Users, fill = Gender)) +   # Fill column
                              geom_bar(stat = "identity", width = .6) +   # draw the bars
                              scale_y_continuous(breaks = brks,   # Breaks
                                                 labels = lbls) + # Labels
                              coord_flip() +  # Flip axes
                              labs(title="Email Campaign Funnel") +
                              theme_tufte() +  # Tufte theme from ggfortify
                              theme(plot.title = element_text(hjust = .5),
                                    axis.ticks = element_blank()) +   # Centre plot title
                              scale_fill_brewer(palette = "Dark2")  # Color palette
```

© 2022 Zhenyuan LU



© 2022 Zhenyuan LU

Change, time series

```
library(ggfortify)
# Plot
autoplot(AirPassengers) +
  labs(title="AirPassengers") +
  theme(plot.title = element_text(hjust=0.5))
```

© 2022 Zhenyuan LU

```
## Error in library(ggfortify): there is no package called 'ggfortify'
```

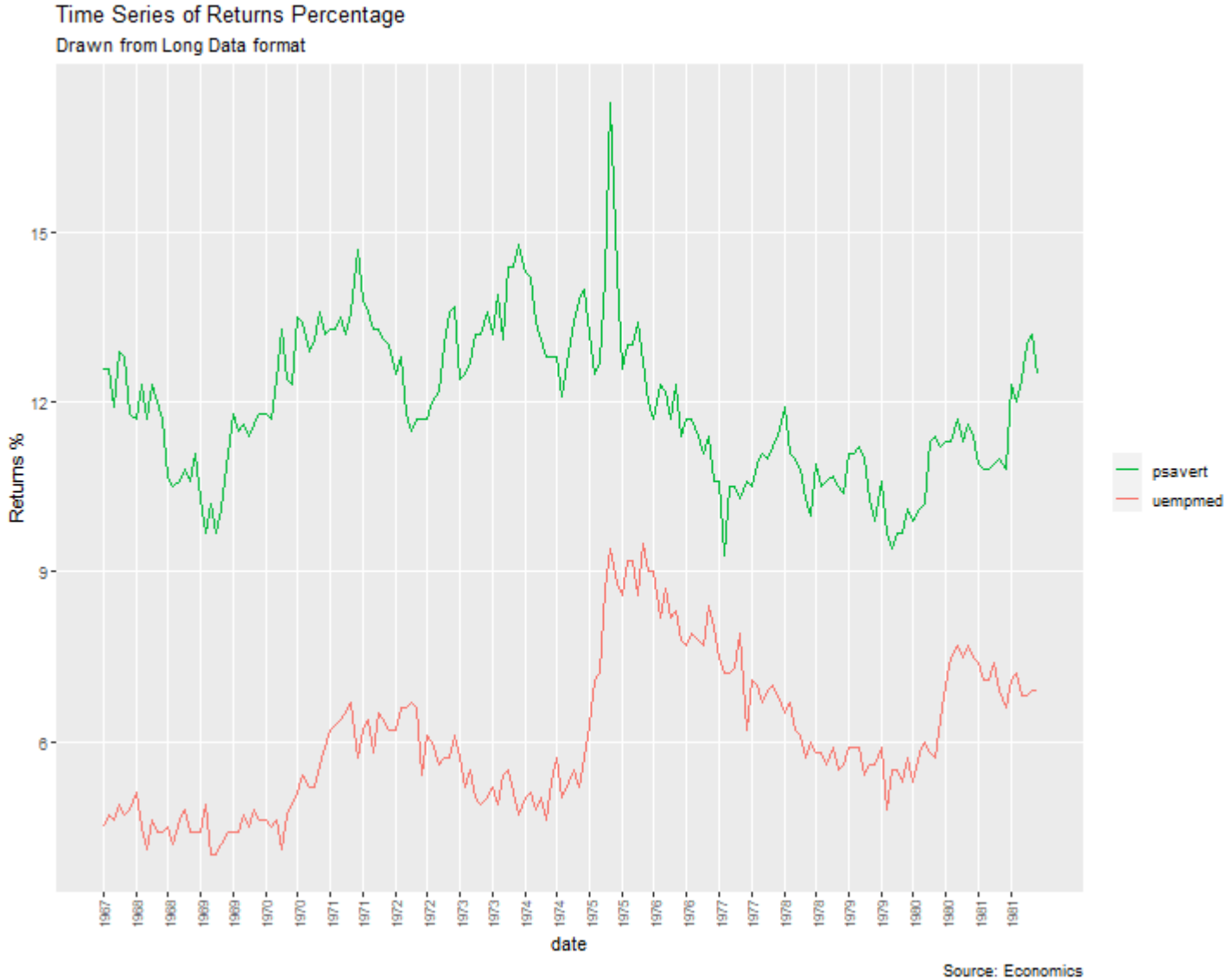
```
## Error in `autoplot()`:
## ! Objects of type ts not supported by autoplot.
```

© 2022 Zhenyuan LU

Change, time series

```
library(ggplot2)
library(lubridate)


df <- economics_long[economics_long$variable %in% c("psavert", "uempmed"), ]
df <- df[lubridate::year(df$date) %in% c(1967:1981), ]

# labels and breaks for X axis text
brks <- df$date[seq(1, length(df$date), 12)]
lbls <- lubridate::year(brks)

# plot
ggplot(df, aes(x=date)) +
  geom_line(aes(y=value, col=variable)) +
  labs(title="Time Series of Returns Percentage",
       subtitle="Drawn from Long Data format",
       caption="Source: Economics",
       y="Returns %",
       color=NULL) +  # title and caption
  scale_x_date(labels = lbls, breaks = brks) +  # change to monthly ticks and labels
  scale_color_manual(labels = c("psavert", "uempmed"),
                     values = c("psavert"="#00ba38", "uempmed"="#f8766d")) +  # line color
  theme(axis.text.x = element_text(angle = 90, vjust=0.5, size = 8),  # rotate x axis text
        panel.grid.minor = element_blank())  # turn off minor grid
```

© 2022 Zhenyuan LU

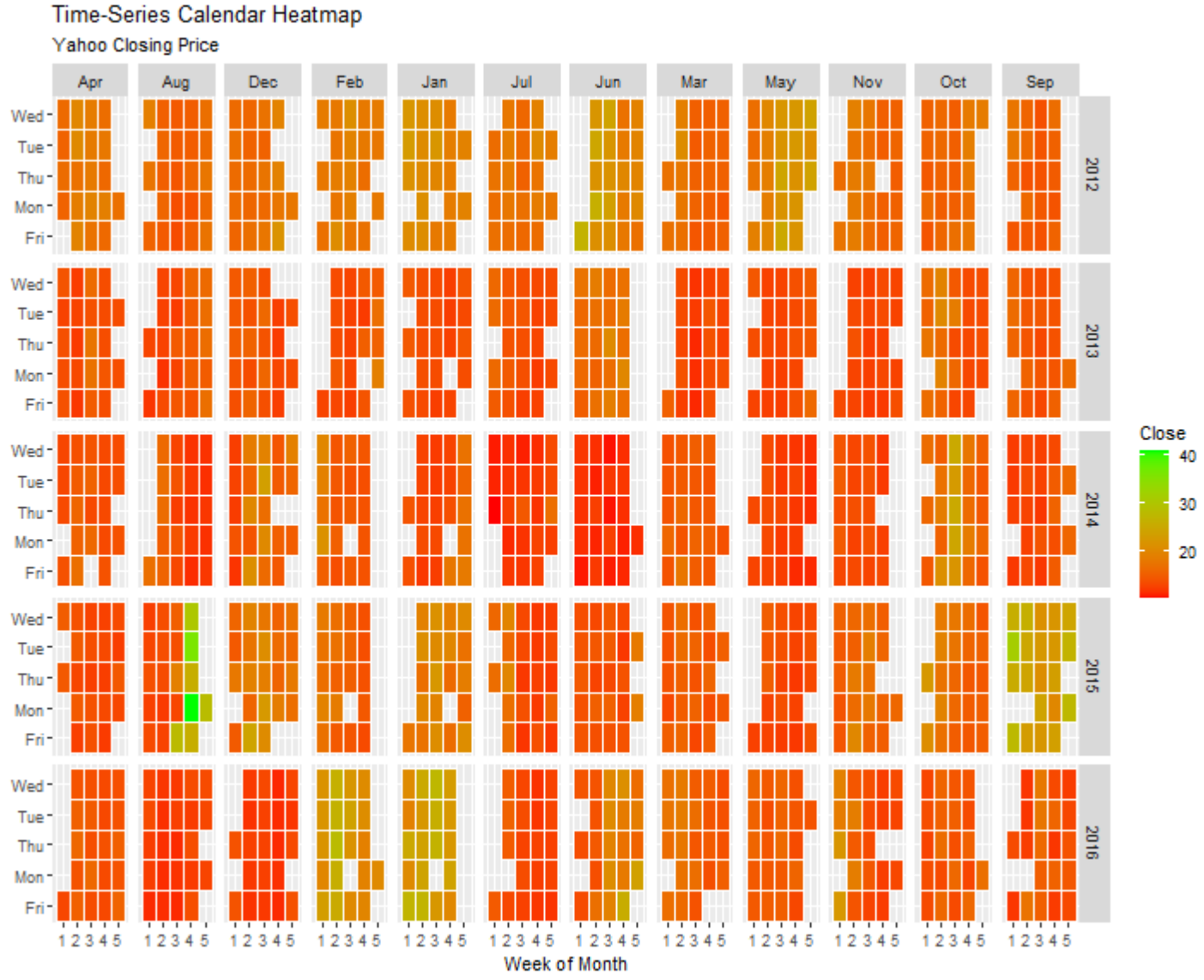


© 2022 Zhenyuan LU

Change, calendar map

```
# http://margintale.blogspot.in/2012/04/ggplot2-time-series-heatmaps.html
library(ggplot2)
library(plyr)
library(scales)
library(zoo)

df <- read.csv("https://raw.githubusercontent.com/selva86/datasets/master/yahoo.csv")
df$date <- as.Date(df$date)  # format date
df <- df[df$year >= 2012, ]  # filter reqd years

# Create Month Week
df$yearmonth <- as.yearmon(df$date)
df$yearmonthf <- factor(df$yearmonth)
df <- ddply(df,.(yearmonthf), transform, monthweek=1+week-min(week))  # compute week number of month
df <- df[, c("year", "yearmonthf", "monthf", "week", "monthweek", "weekdayf", "VIX.Close")]

# Plot
ggplot(df, aes(monthweek, weekdayf, fill = VIX.Close)) +
  geom_tile(colour = "white") +
  facet_grid(year~monthf) +
  scale_fill_gradient(low="red", high="green") +
  labs(x="Week of Month",
       y="",
       title = "Time-Series Calendar Heatmap",
```

© 2022 Zhenyuan LU

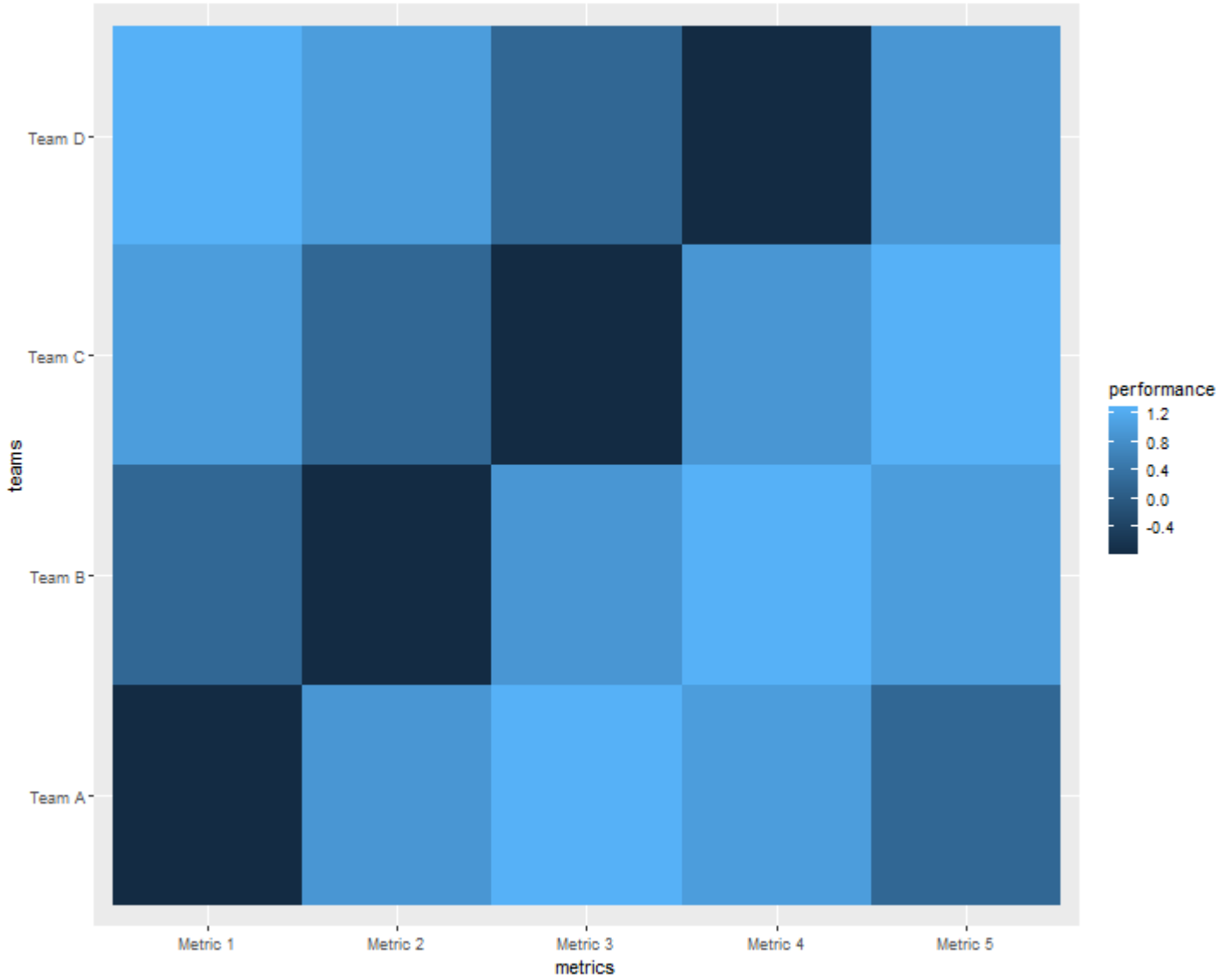


© 2022 Zhenyuan LU

Change, heat map

```
#Heat map
set.seed(41)
expand.grid(
  teams = c("Team A", "Team B", "Team C", "Team D")
  ,
  metrics = c("Metric 1", "Metric 2", "Metric 3", "Metric 4", "Metric 5")
) %>%
  mutate(performance = rnorm(5)) %>% # add variable: performance
  ggplot(aes(x = metrics, y = teams)) + geom_tile(aes(fill = performance))
```

© 2022 Zhenyuan LU



© 2022 Zhenyuan LU

Change, seasonal plot

```
library(ggplot2)
library(forecast)


# Subset data
window(nottem, start=c(1920, 1), end=c(1925, 12)) %>%   # subset a smaller timewindow
ggseasonplot() + labs(title="Seasonal plot: Air temperatures at Nottingham Castle")
```

© 2022 Zhenyuan LU

```
## Error in library(forecast): there is no package called 'forecast'
```

```
## Error in ggseasonplot(.): could not find function "ggseasonplot"
```

© 2022 Zhenyuan LU

Advanced Visualization

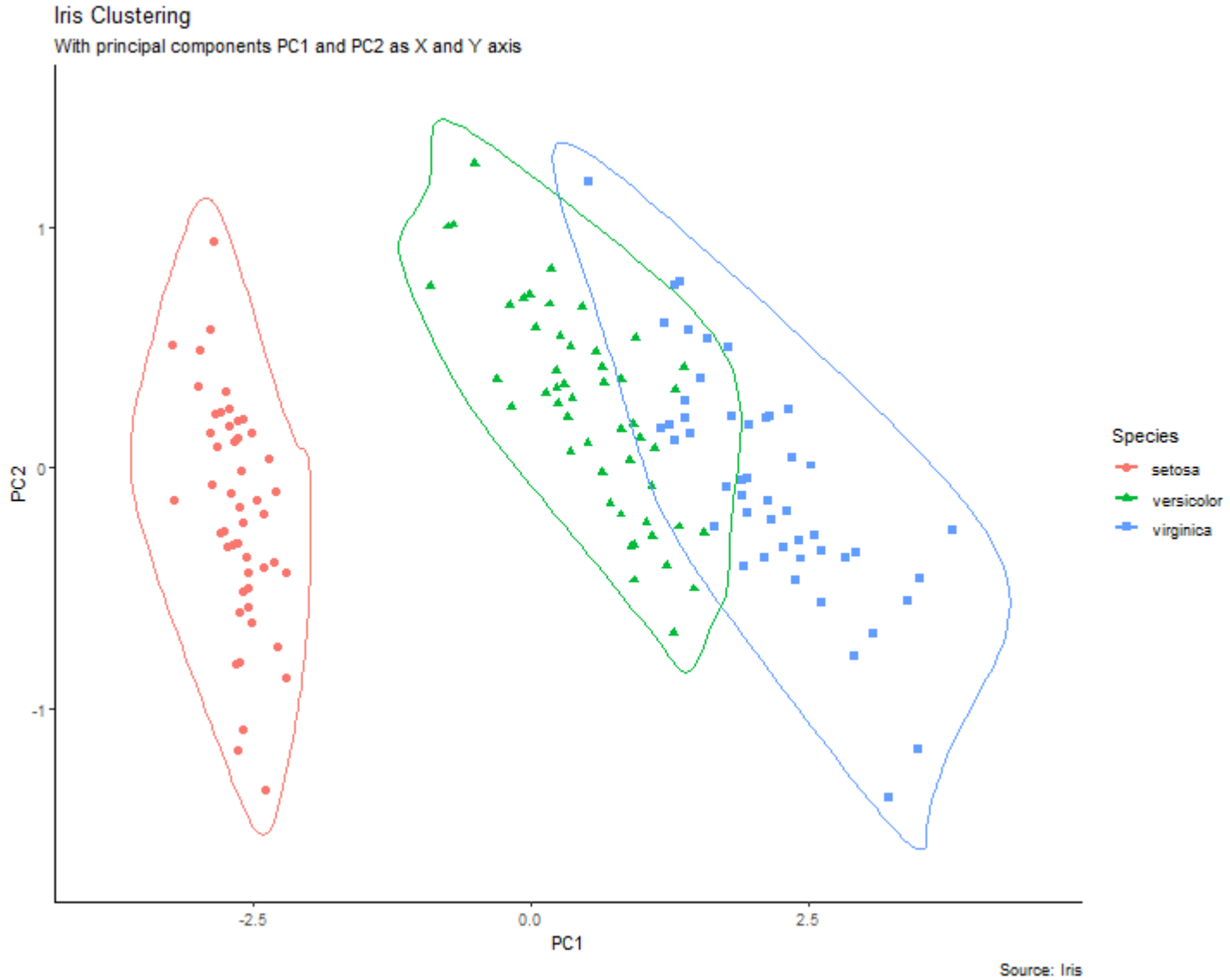
© 2022 Zhenyuan LU

Static, cluster

```
library(ggplot2)
library(ggalt)
library(ggfortify)
theme_set(theme_classic())
# Compute data with principal components
df <- iris[c(1, 2, 3, 4)]
pca_mod <- prcomp(df)  # compute principal components

# Data frame of principal components
df_pc <- data.frame(pca_mod$x, Species=iris$Species)  # dataframe of principal components
df_pc_vir <- df_pc[df_pc$Species == "virginica", ]  # df for 'virginica'
df_pc_set <- df_pc[df_pc$Species == "setosa", ]  # df for 'setosa'
df_pc_ver <- df_pc[df_pc$Species == "versicolor", ]  # df for 'versicolor'

# Plot
ggplot(df_pc, aes(PC1, PC2, col=Species)) +
  geom_point(aes(shape=Species), size=2) +   # draw points
  labs(title="Iris Clustering",
       subtitle="With principal components PC1 and PC2 as X and Y axis",
       caption="Source: Iris") +
  coord_cartesian(xlim = 1.2 * c(min(df_pc$PC1), max(df_pc$PC1)),
                  ylim = 1.2 * c(min(df_pc$PC2), max(df_pc$PC2))) +   # change axis limits
  geom_encircle(data = df_pc_vir, aes(x=PC1, y=PC2)) +   # draw circles
  geom_encircle(data = df_pc_set, aes(x=PC1, y=PC2)) +
```

© 2022 Zhenyuan LU

```
## Error in library(ggfortify): there is no package called 'ggfortify'
```



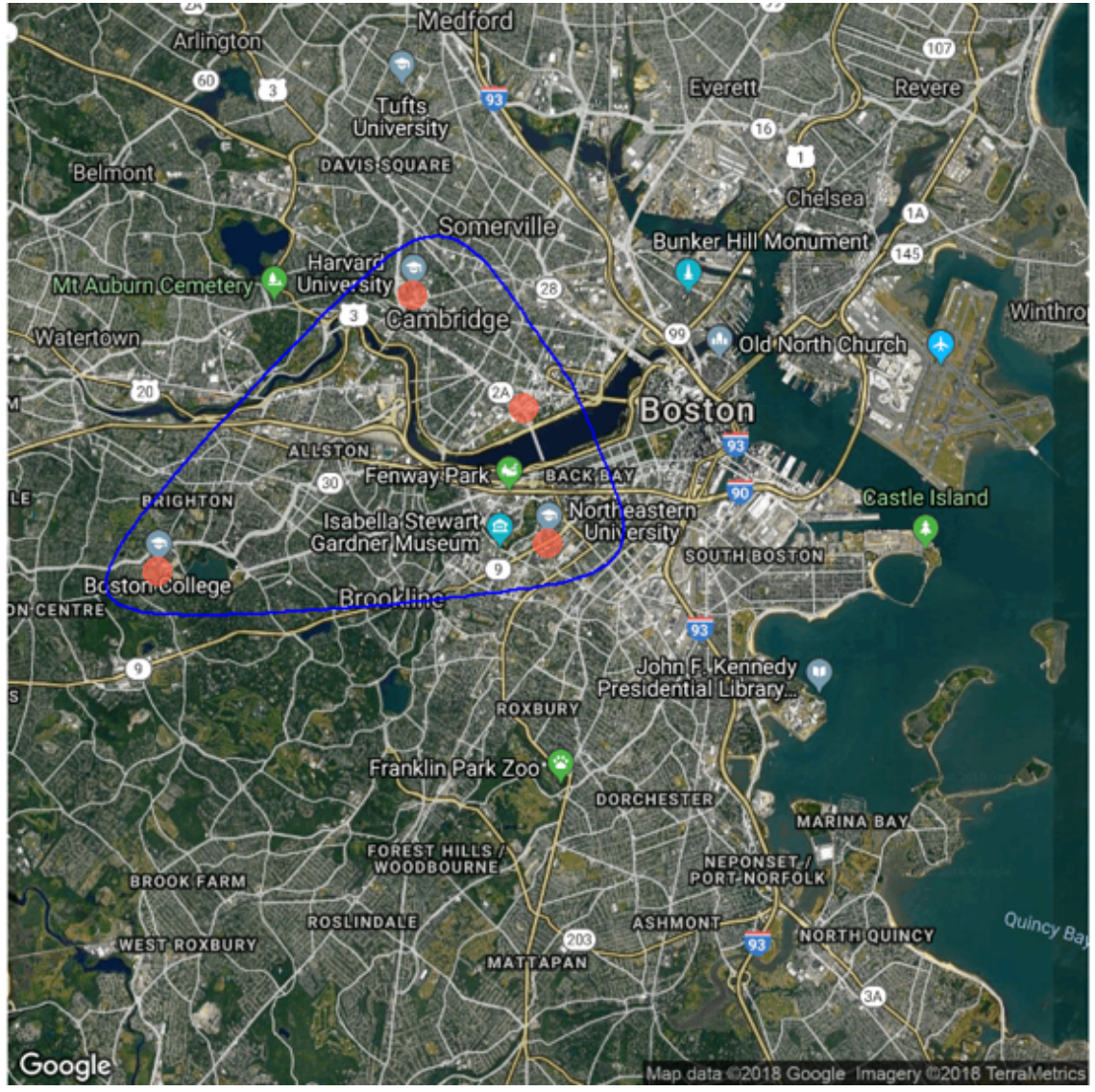
© 2022 Zhenyuan LU

Static, spatial

google needs api key for retrieving the map now Use ?reigister_google for more information

```
library(ggplot2)
library(ggmap)
library(ggalt)

# Google Hybrid Map
neu_ggl_hybrid_map <-
  qmap("neu", zoom = 12,
       source = "google",
       maptype = "hybrid")
neu_places <- c("Northeastern University, MA",
                "MIT",
                "Harvard University",
                "Boston College, MA")
places_loc <- geocode(neu_places)
# Google Hybrid Map
neu_ggl_hybrid_map + geom_point(
  aes(x = lon, y = lat),
  data = places_loc,
  alpha = 0.7,
  size = 7,
  color = "tomato"
) +
```

© 2022 Zhenyuan LU



© 2022 Zhenyuan LU

Static, network

```
#Network Map
library(maps)
library(geosphere)
airports <- read.csv("E:/IE6600/IE6600_SEA_Spring2021/R/R/Network Data Sets/Dataset3-Airlines-NODES.csv",
                     header=TRUE)
flights <- read.csv("E:/IE6600/IE6600_SEA_Spring2021/R/R/Network Data Sets/Dataset3-Airlines-EDGES.csv",
                    header=TRUE, as.is=TRUE)
# Select only large airports: ones with more than 10 connections in the data.
tab <- table(flights$Source)
big.id <- names(tab)[tab>10]
airports <- airports[airports$ID %in% big.id,]
flights  <- flights[flights$Source %in% big.id &
                     flights$Target %in% big.id, ]

# Plot a map of the united states:
map("state", col="grey20", fill=TRUE, bg="black", lwd=0.1)
```

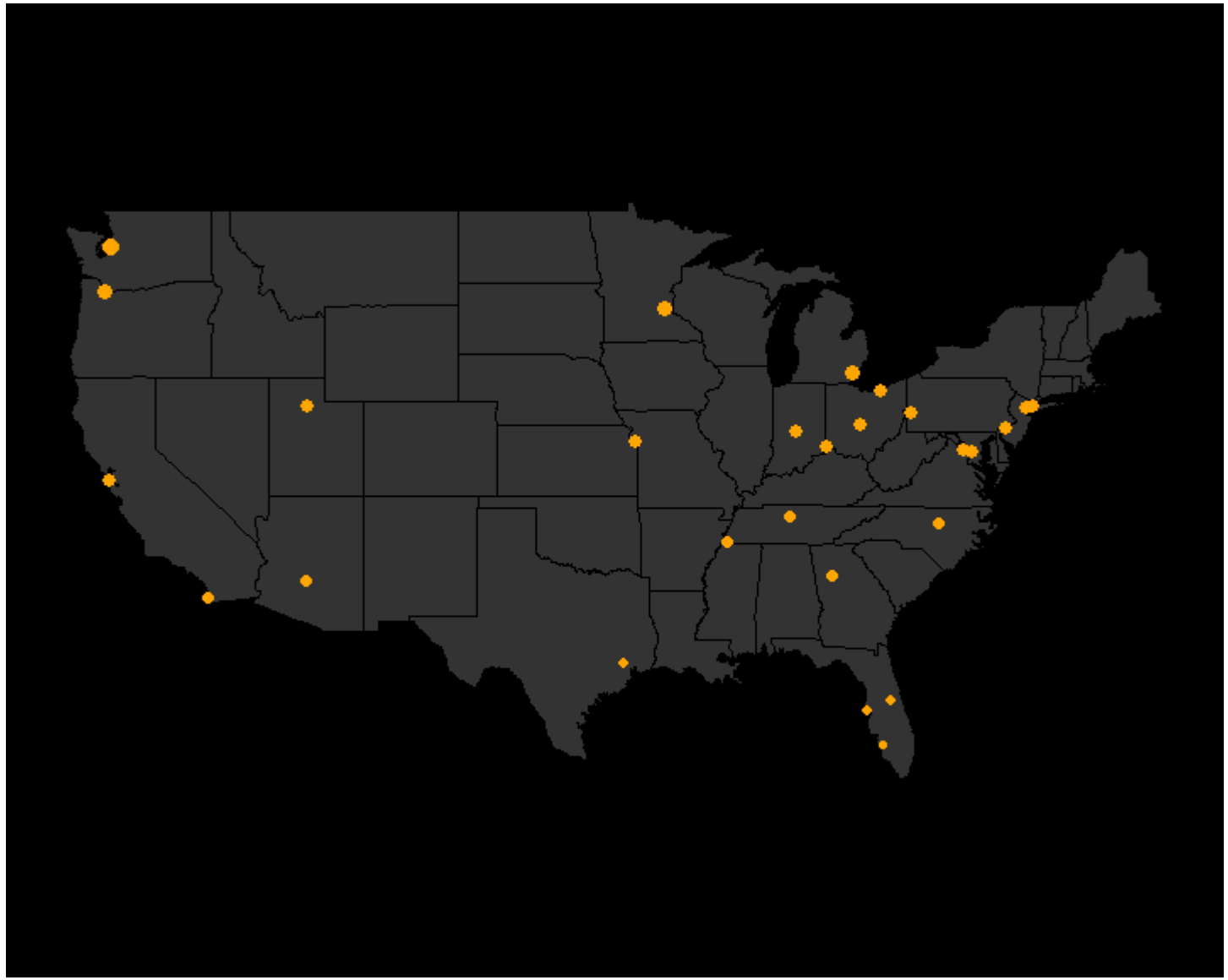
© 2022 Zhenyuan LU

Static, network (cont'd)

```
# Add a point on the map for each airport:
points(x=airports$longitude, y=airports$latitude, pch=19,
       cex=airports$Visits/80, col="orange")
col.1 <- adjustcolor("orange red", alpha=0.4)
col.2 <- adjustcolor("orange", alpha=0.4)
edge.pal <- colorRampPalette(c(col.1, col.2), alpha = TRUE)
edge.col <- edge.pal(100)
for(i in 1:nrow(flights))  {
  node1 <- airports[airports$ID == flights[i,]$Source,]
  node2 <- airports[airports$ID == flights[i,]$Target,]

  arc <- gcIntermediate( c(node1[1,]$longitude, node1[1,]$latitude),
                         c(node2[1,]$longitude, node2[1,]$latitude),
                         n=1000, addStartEnd=TRUE )
  edge.ind <- round(100*flights[i,]$Freq / max(flights$Freq))

  lines(arc, col=edge.col[edge.ind], lwd=edge.ind/30)
}
```

© 2022 Zhenyuan LU

```
## Error in library(geosphere): there is no package called 'geosphere'
```



```
## Error in gcIntermediate(c(node1[1, ]$longitude, node1[1, ]$latitude), : could not find function "gcIntermediate"
```

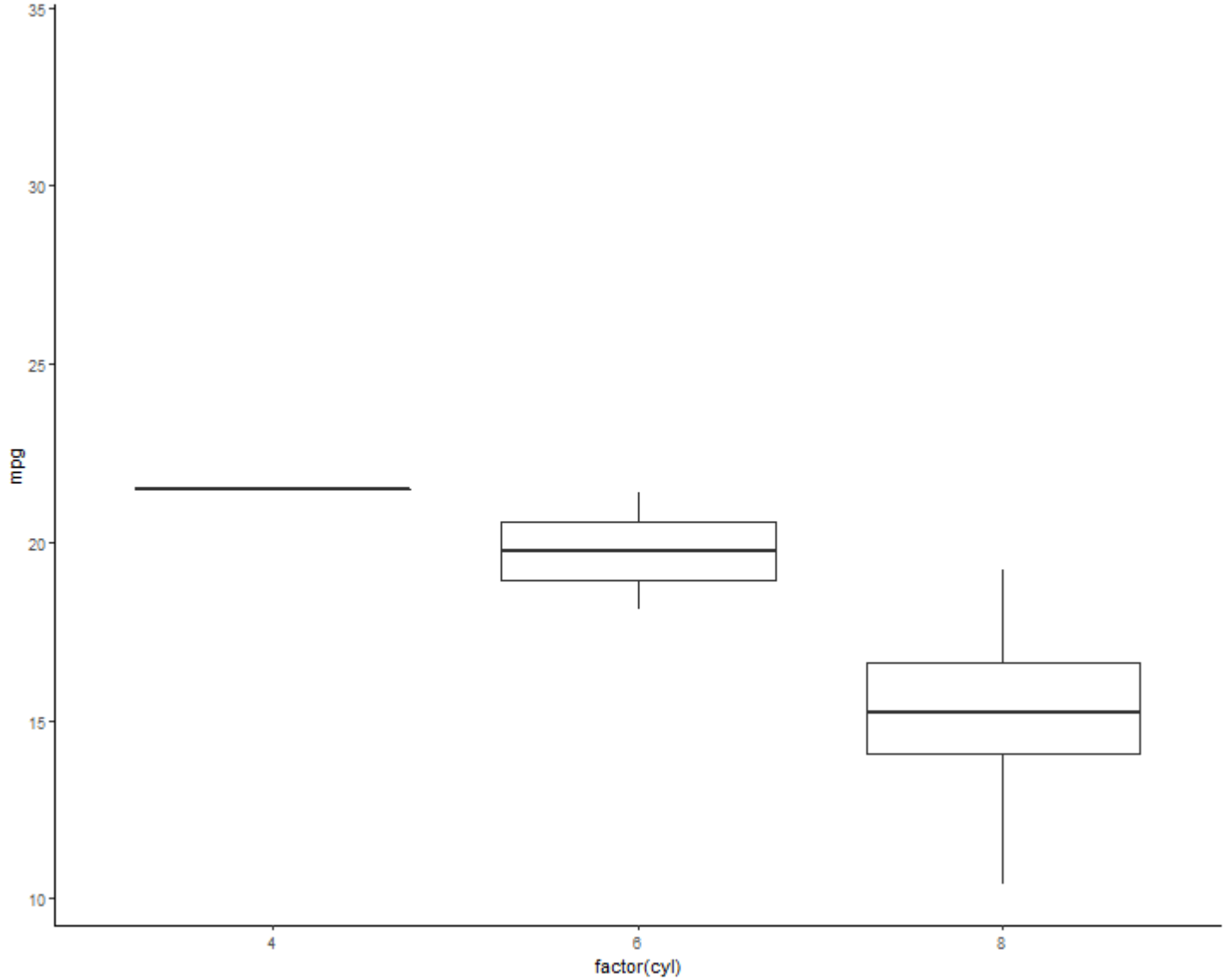
© 2022 Zhaoyuan Liu

Animated, Box plot

```
library(ggplot2)
library(gganimate)
library(gifski)
ggplot(mtcars, aes(factor(cyl), mpg)) +
  geom_boxplot() +
  # Here comes the gganimate code
  transition_states(
    gear,
    transition_length = 2,
    state_length = 1
  ) +
  enter_fade() +
  exit_shrink() +
  ease_aes('sine-in-out')
```

© 2022 Zhenyuan LU

```
## Error in library(gifski): there is no package called 'gifski'
```

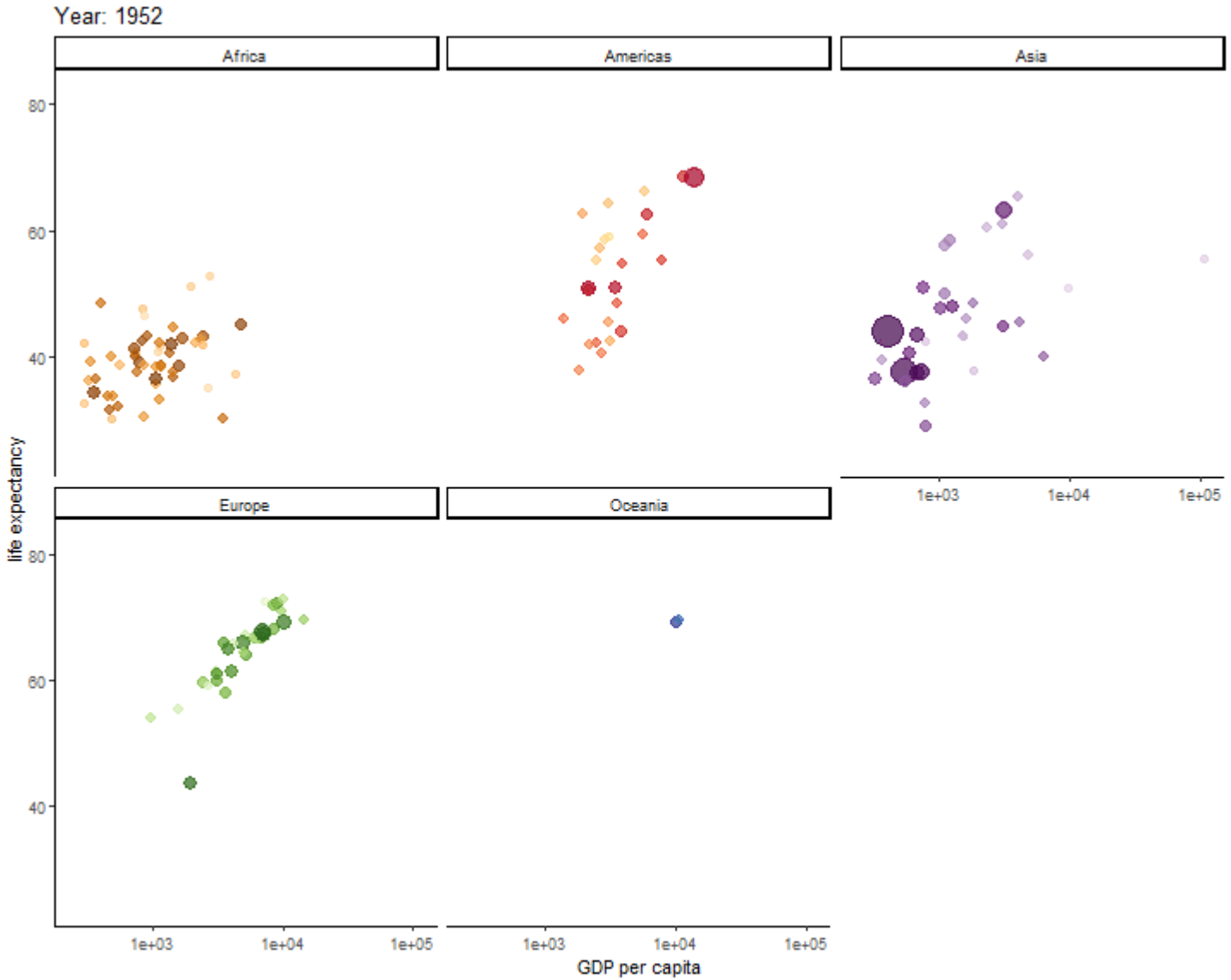


© 2022 Zhenyuan LU

Animated, Dot Plot

```
library(gapminder)

ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, colour = country)) +
  geom_point(alpha = 0.7, show.legend = FALSE) +
  scale_colour_manual(values = country_colors) +
  scale_size(range = c(2, 12)) +
  scale_x_log10() +
  facet_wrap(~continent) +
  # Here comes the gganimate specific bits
  labs(title = 'Year: {frame_time}', x = 'GDP per capita', y = 'life expectancy') +
  transition_time(year) +
  ease_aes('linear')
```

© 2022 Zhenyuan LU

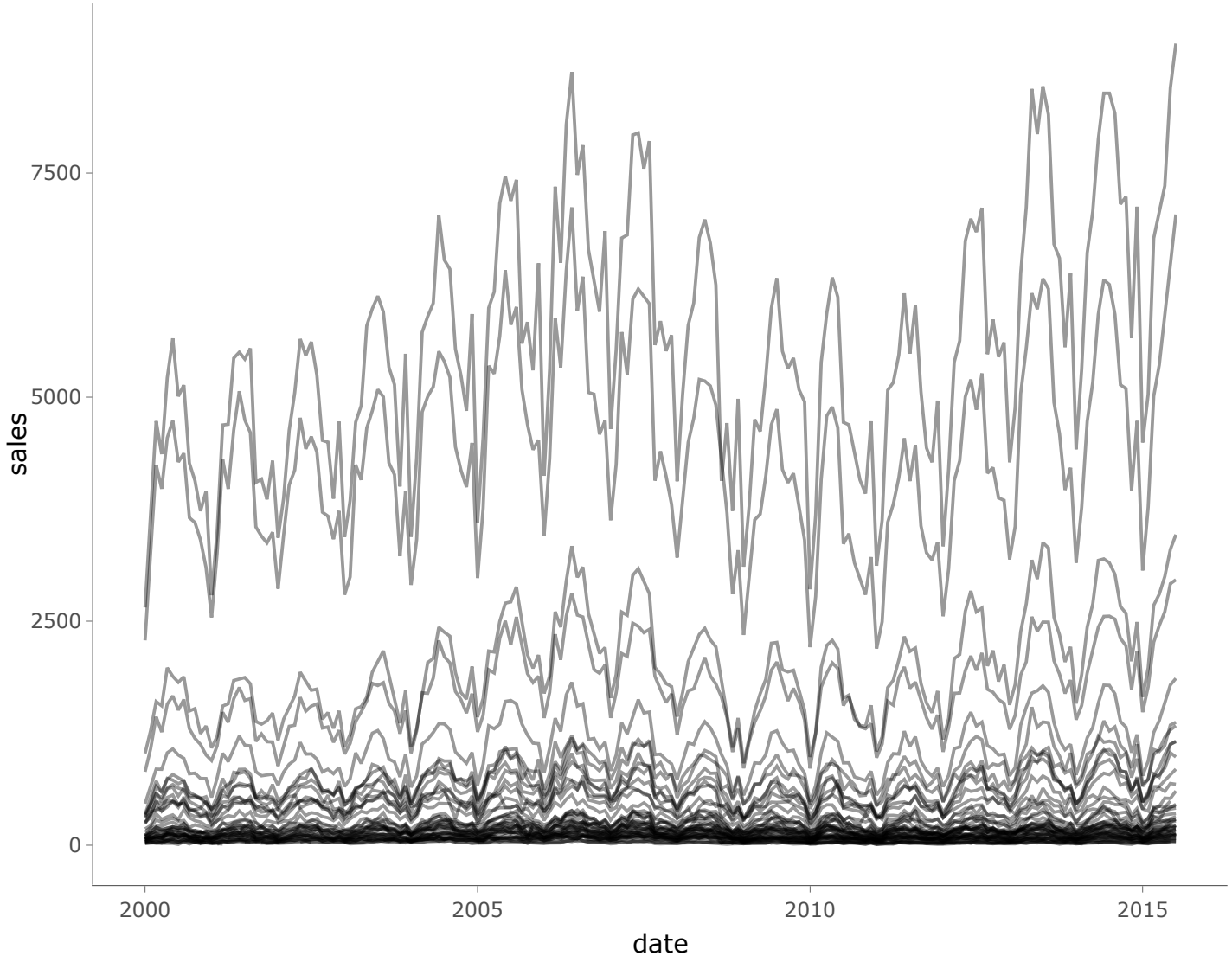


© 2022 Zhenyuan LU

Interactive plot with ggplotlly()

```
library(plotly)
g <- ggplot(txhousing, aes(x = date, y = sales, group = city)) +
  geom_line(alpha = 0.4)
ggplotly(g, tooltip = c("city"))
```

© 2022 Zhenyuan LU

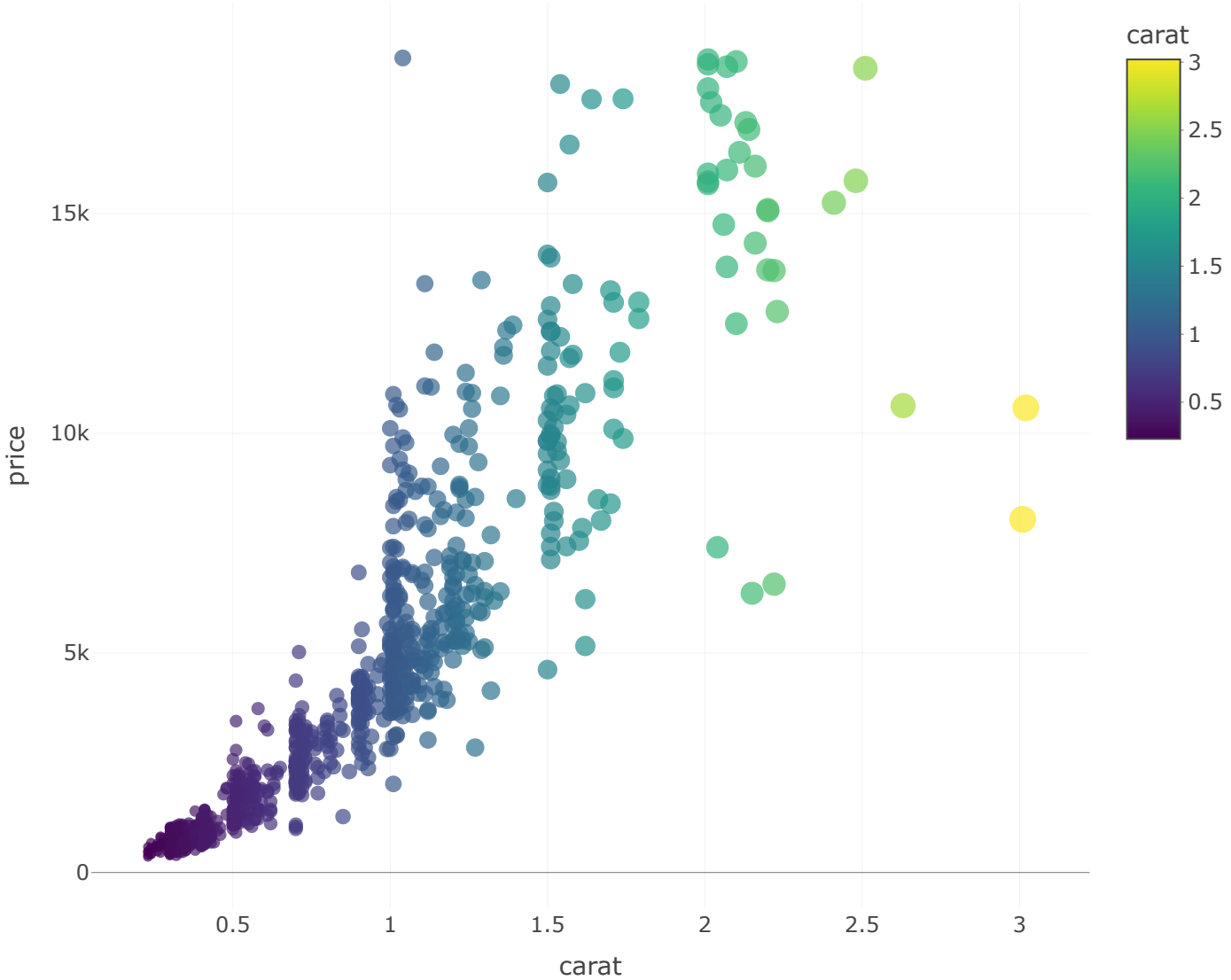


© 2022 Zhenyuan LU

Interactive plot, plotly

```
library(plotly)
diamonds[sample(nrow(diamonds), 1000), ] %>%
  plot_ly(
  x = ~carat, y = ~price,
  color = ~carat, size = ~carat
)
```

© 2022 Zhenyuan LU



© 2022 Zhenyuan LU

Interactive plot, D3

```
library(networkD3)
forceNetwork(Links = MisLinks, Nodes = MisNodes, Source = "source",
             Target = "target", Value = "value", NodeID = "name",
             Group = "group", opacity = 0.9, Nodesize = 3,
             linkDistance = 100, fontSize = 20)
```

© 2022 Zhenyuan LU

```
## Error in library(networkD3): there is no package called 'networkD3'
```

```
## Error in forceNetwork(Links = MisLinks, Nodes = MisNodes, Source = "source", : could not find function "forceNetwo
```

© 2022 Zhenyuan LU

Applicatin in real life

© 2022 Zhenyuan LU

GB

```
# Libraries
library(tidyverse)

# Get the world polygon and extract USA
library(maps)
USA <- map_data("world") %>% filter(region=="USA")
# Get a data frame with longitude, latitude, and size of bubbles (a bubble = a city)
data=world.cities %>% filter(country.etc=="USA")

# Left chart
ggplot() +
  geom_polygon(data = USA, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data=data, aes(x=long, y=lat)) +
  theme_void() + ylim(20,56) + xlim(-125,-65)+coord_map()
```

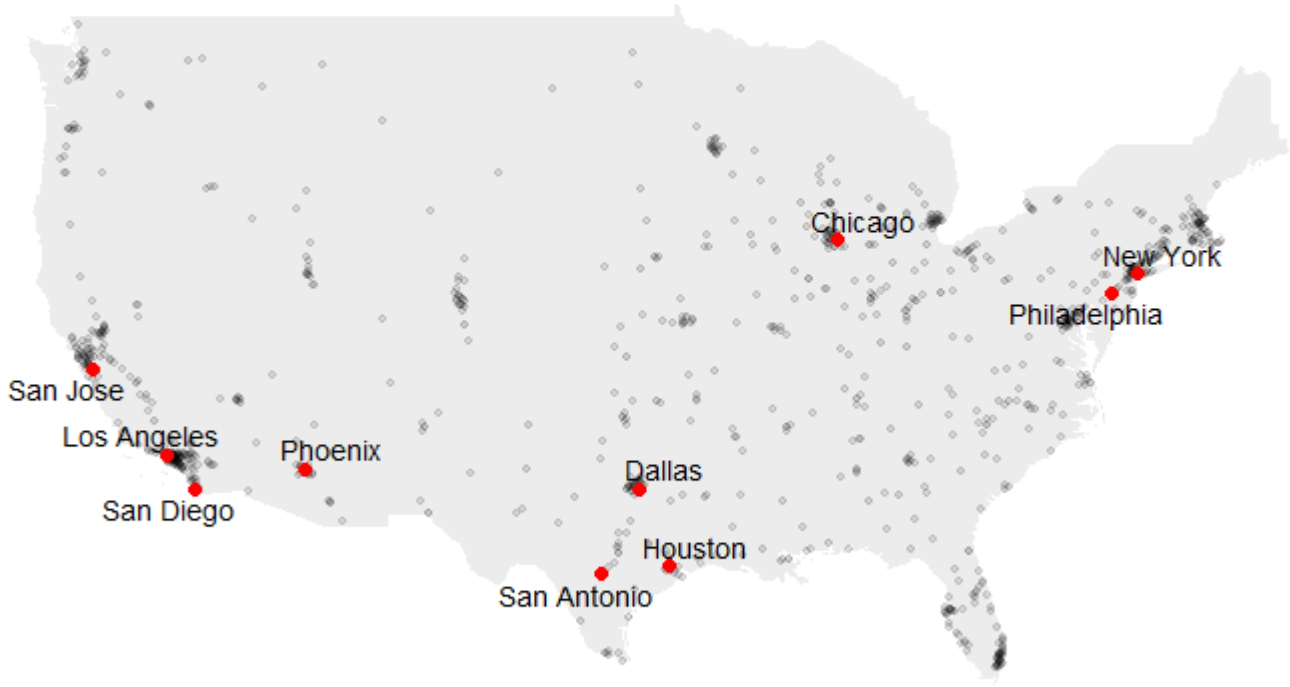
© 2022 Zhenyuan LU

© 2022 Zhenyuan LU

GB

```
# Second graphic with names of the 10 biggest cities
library(ggrepel)
ggplot() +
  geom_polygon(data = USA, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data=data, aes(x=long, y=lat, alpha=pop)) +
  geom_text_repel(data=data %>% arrange(pop) %>% tail(10), aes(x=long, y=lat, label=name), size=5) +
  geom_point( data=data %>% arrange(pop) %>% tail(10), aes(x=long, y=lat), color="red", size=3) +
  theme_void() + ylim(20,56) + xlim(-125,-65)+coord_map()  +
  theme(legend.position="none")
```

© 2022 Zhenyuan LU

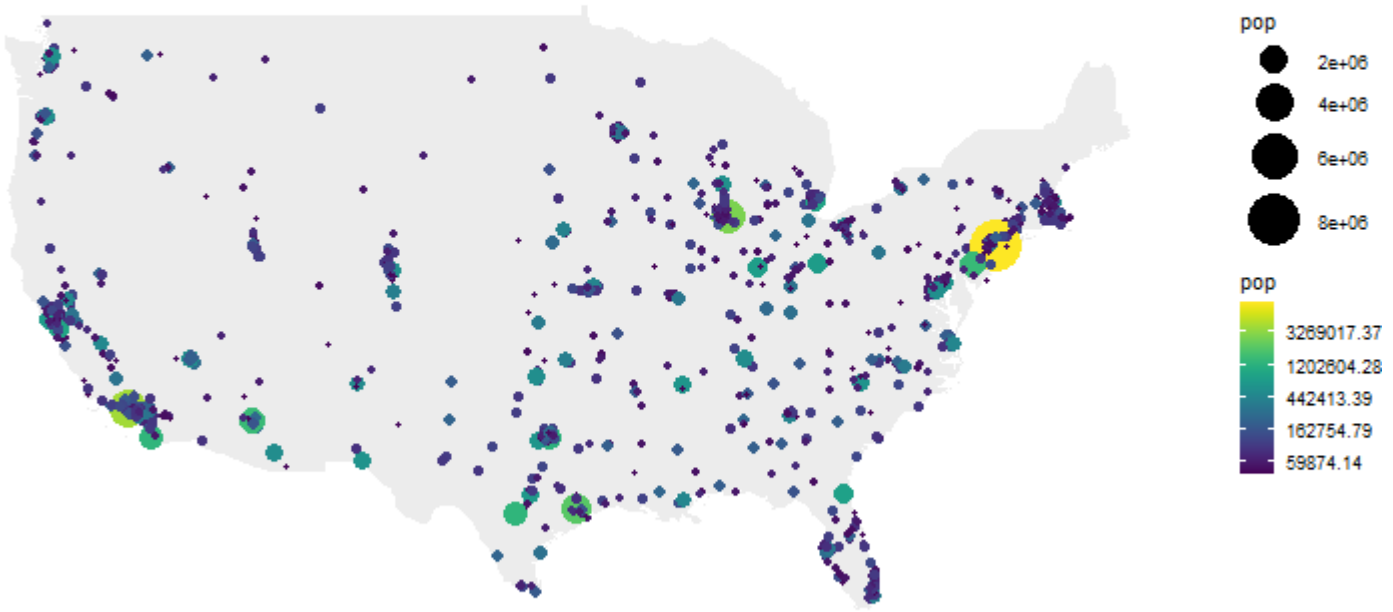


© 2022 Zhenyuan LU

GB

```
library(viridis)
# first: use size and color
ggplot() +
  geom_polygon(data = USA, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( data=data, aes(x=long, y=lat, size=pop, color=pop)) +
  scale_size_continuous(range=c(1,12)) +
  scale_color_viridis(trans="log") +
  theme_void() + ylim(20,56) + xlim(-125,-65)+coord_map()
```

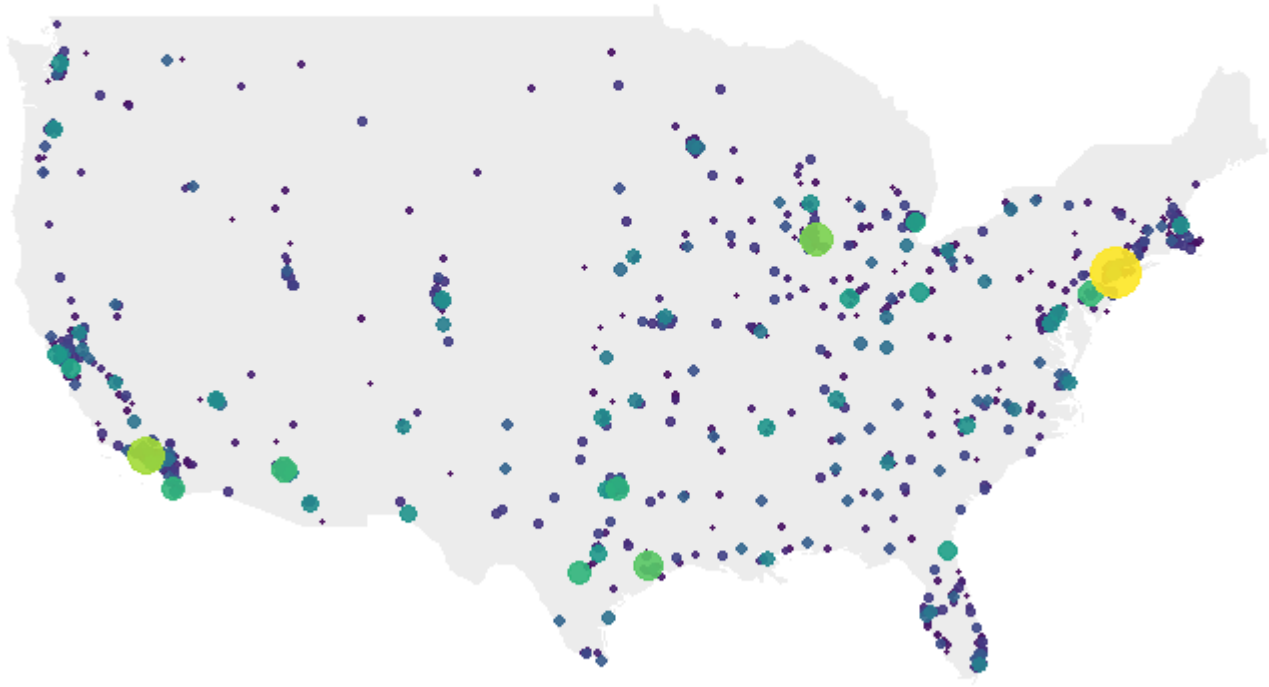
© 2022 Zhenyuan LU



© 2022 Zhenyuan LU

```r
# second: reorder your dataset first! Big cities appear later = on top
data %>%
  arrange(pop) %>% # This reorder your data frame
  mutate( name=factor(name, unique(name))) %>% #this reorder the order of the levels of your factor --> this
  ggplot() +
  geom_polygon(data = USA, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point( aes(x=long, y=lat, size=pop, color=pop), alpha=0.9) +
  scale_size_continuous(range=c(1,12)) +
  scale_color_viridis(trans="log") +
  theme_void() + ylim(20,56) + xlim(-125,-65)+coord_map()  + theme(legend.position="none")
```

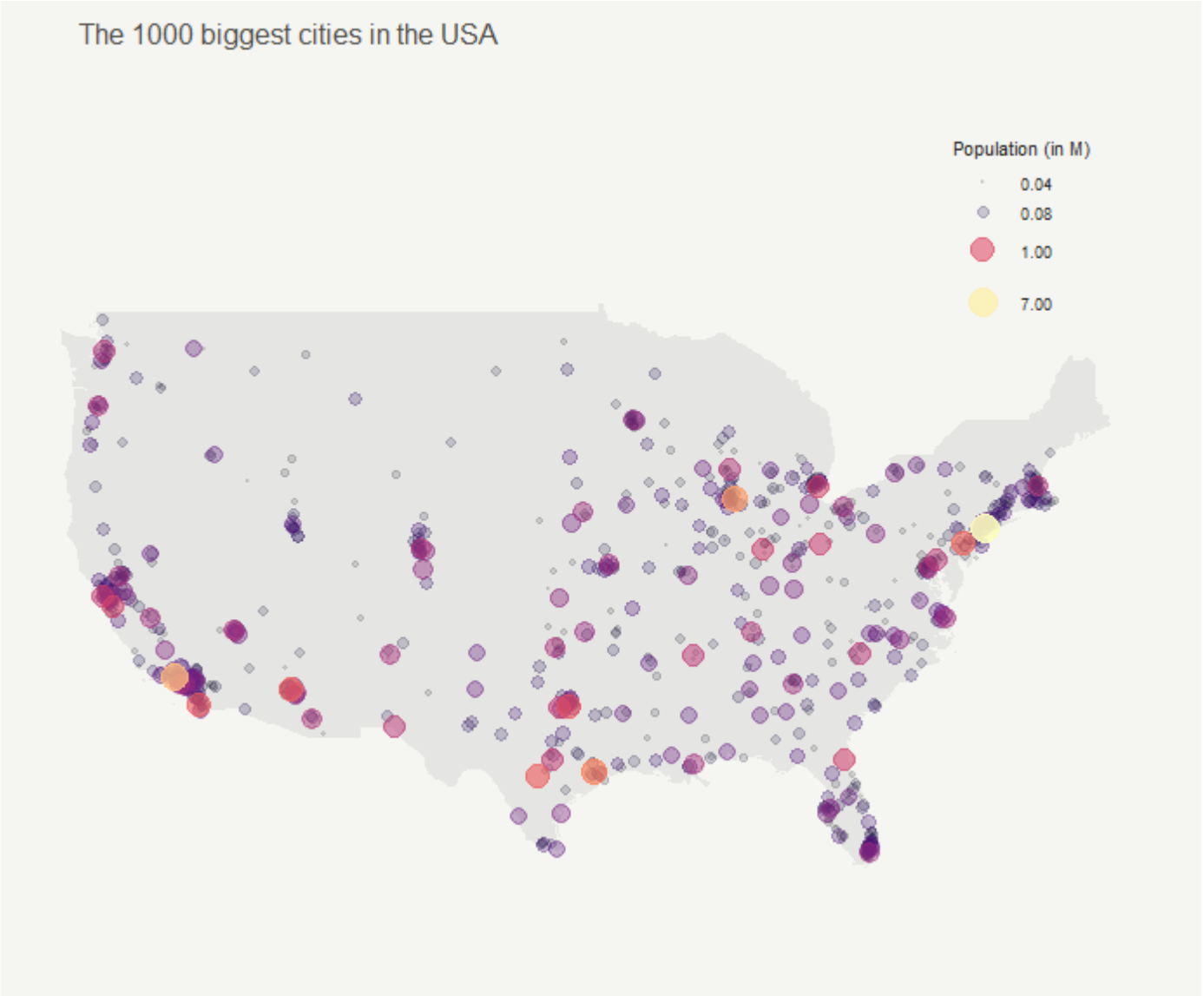
© 2022 Zhenyuan LU



© 2022 Zhenyuan LU

```
library(viridis)
mybreaks=c(0.02, 0.04, 0.08, 1, 7)
data %>%
  arrange(pop) %>%
  mutate( name=factor(name, unique(name))) %>%
  mutate(pop=pop/1000000) %>%
  ggplot() +
  geom_polygon(data =USA, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point(  aes(x=long, y=lat, size=pop, color=pop, alpha=pop), shape=20, stroke=FALSE) +
  scale_size_continuous(name="Population (in M)", trans="log", range=c(1,12), breaks=mybreaks) +
  scale_alpha_continuous(name="Population (in M)", trans="log", range=c(0.1, .9), breaks=mybreaks) +
  scale_color_viridis(option="magma", trans="log", breaks=mybreaks, name="Population (in M)" ) +
   theme_void() + ylim(20,56) + xlim(-125,-65)+coord_map() +
  guides( colour = guide_legend()) +
  ggtitle("The 1000 biggest cities in the USA") +
  theme(
    legend.position = c(0.85, 0.8),
    text = element_text(color = "#22211d"),
    plot.background = element_rect(fill = "#f5f5f2", color = NA),
    panel.background = element_rect(fill = "#f5f5f2", color = NA),
    legend.background = element_rect(fill = "#f5f5f2", color = NA),
    plot.title = element_text(size= 16, hjust=0.1, color = "#4e4d47", margin = margin(b = -0.1, t = 0.4, l = 
```

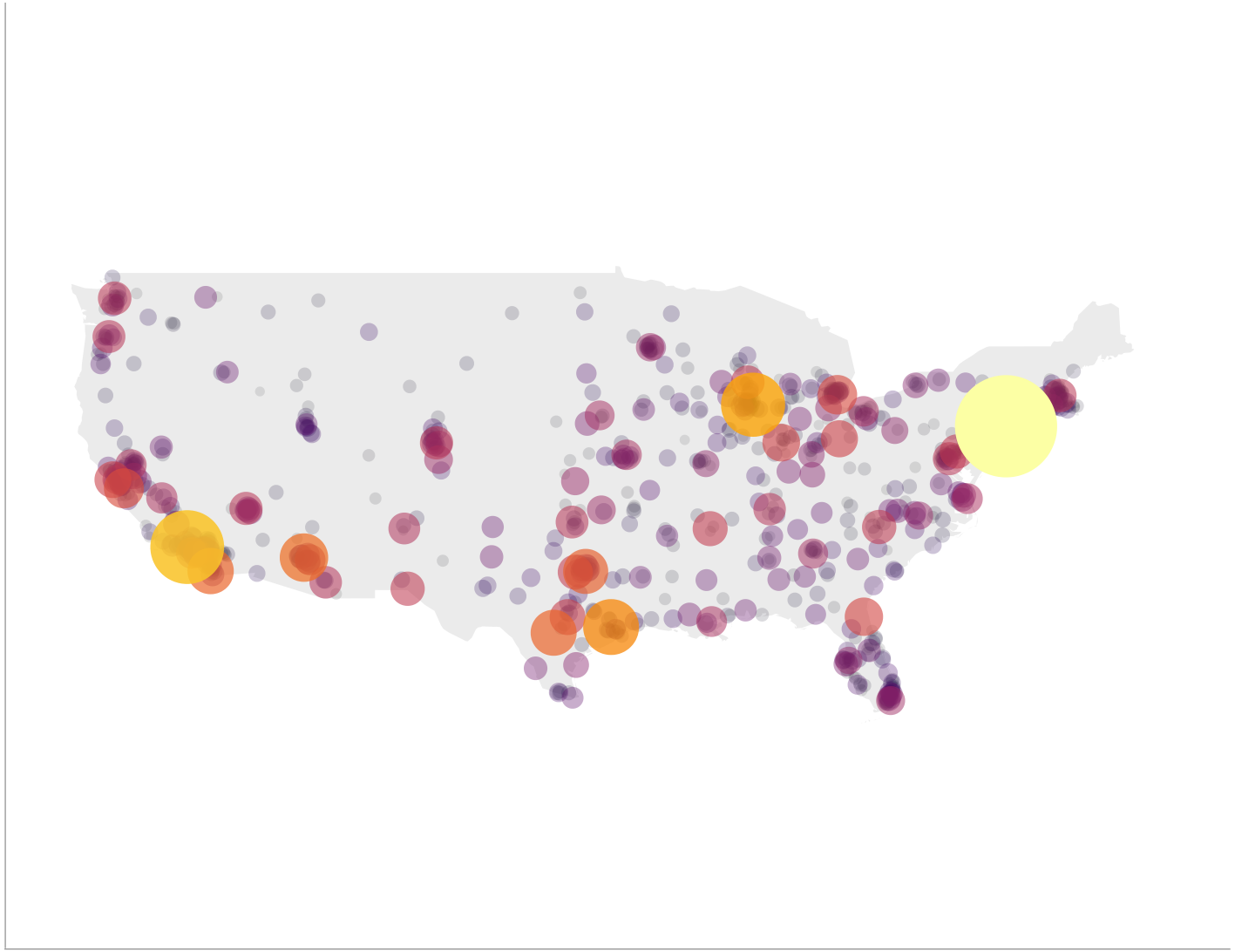
© 2022 Zhenyuan LU



© 2022 Zhenyuan LU

```r
# Easy to make it interactive!
library(plotly)

# plot
p=data %>%

  arrange(pop) %>%
  mutate( name=factor(name, unique(name))) %>%
  mutate( mytext=paste("City: ", name, "\n", "Population: ", pop, sep="")) %>% # This prepare the text displa

  # Makte the static plot calling this text:
  ggplot() +
  geom_polygon(data = USA, aes(x=long, y = lat, group = group), fill="grey", alpha=0.3) +
  geom_point(aes(x=long, y=lat, size=pop, color=pop, text=mytext, alpha=pop) ) +

  scale_size_continuous(range=c(1,15)) +
  scale_color_viridis(option="inferno", trans="log" ) +
  scale_alpha_continuous(trans="log") +
  theme_void() +
  ylim(0,100) + xlim(-125,-65) +
  coord_map() +
  theme(legend.position = "none")

ggplotly(p, tooltip="text")
```

© 2022 Zhenyuan LU



© 2022 Zhenyuan LU

References

Wickham, Hadley, and Garrett Grolemund. R For Data Science. OReilly, 2017.

Lander, Jared P.R For Everyone - Advanced Analytics and Graphics. Pearson Education (Us), 2017

"Companies Using R."R-Bloggers, R-Bloggers, 28 Dec. 2016, www.r-bloggers.com/companies-using-r/.

Kabacoff, Robert. "Access to Database Management Systems (DBMS)."

www.statmethods.net/input/dbinterface.html.

www.r-bloggers.com

http://r-statistics.co

www.r-graph-gallery.com

Kabacoff, Robert.R In Action: Data Analysis and Graphics with R. Manning, 2015.

© 2022 Zhenyuan LU