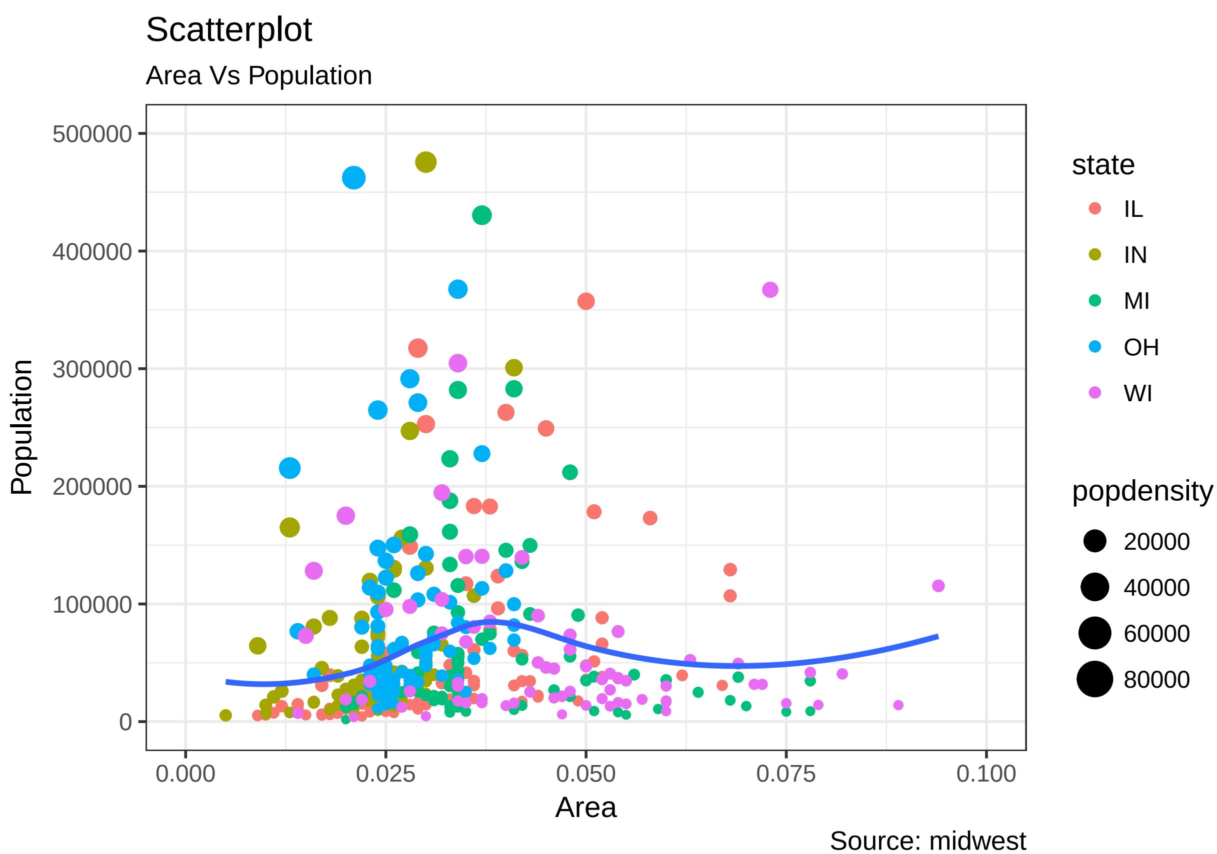
Data Visualisation in R (link for html version)

1. Correlation

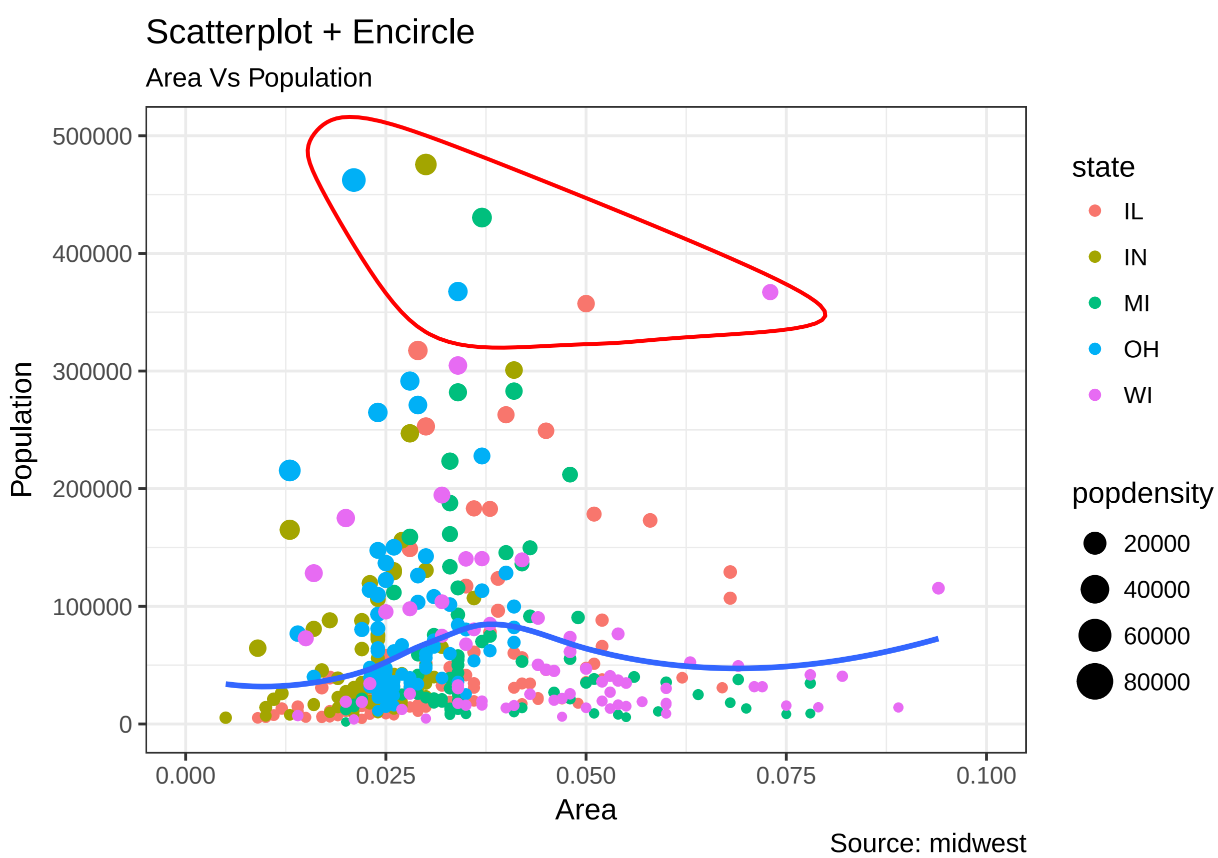
1.1 Scatterplot

```
options(scipen=999)  # turn-off scientific notation like 1e+48
library(ggplot2)
theme_set(theme_bw())  # pre-set the bw theme
data("midwest", package = "ggplot2")
# midwest <- read.csv("http://goo.gl/G1K41K")  # bkup data source

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

plot(gg)
```



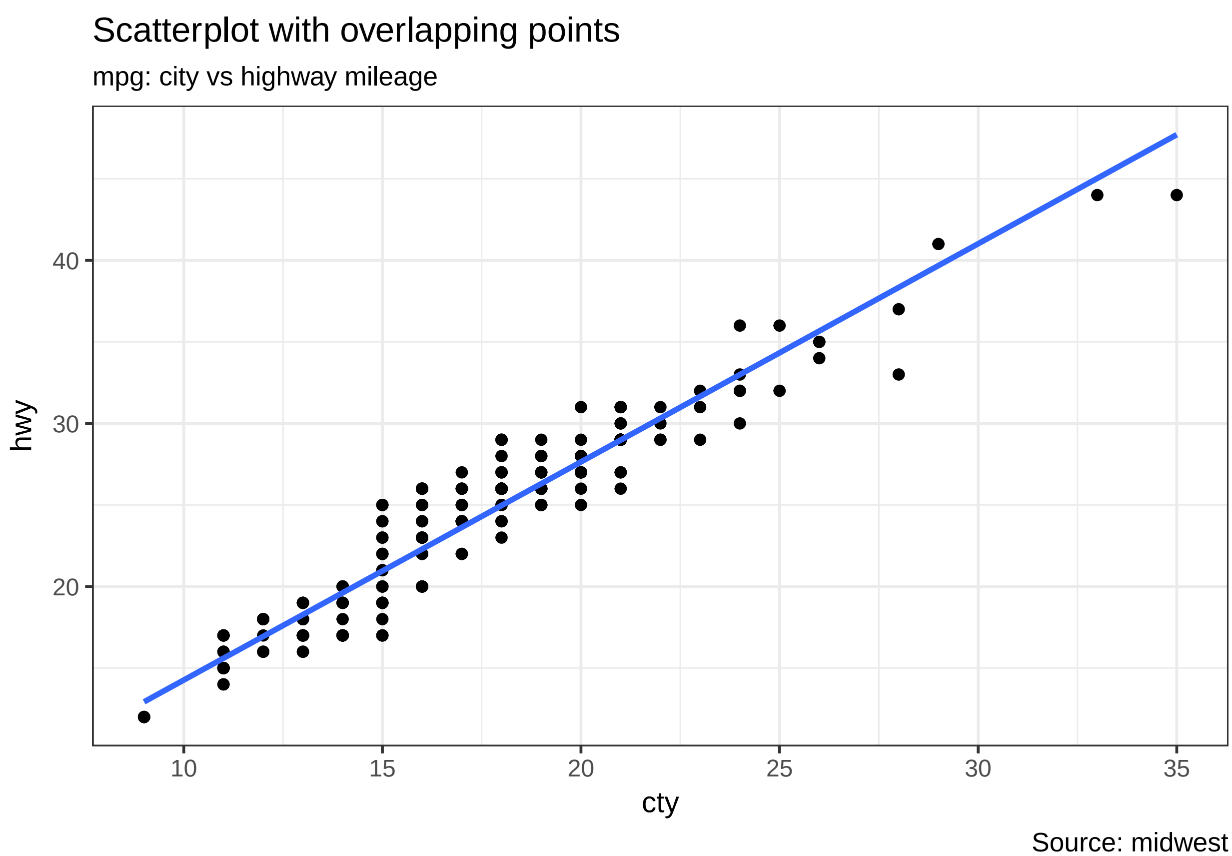
1.2 Scatterplot with Encircling

```
# install 'ggalt' pkg
# devtools::install_github("hrbrmstr/ggalt")
options(scipen = 999)
library(ggplot2)
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")
```



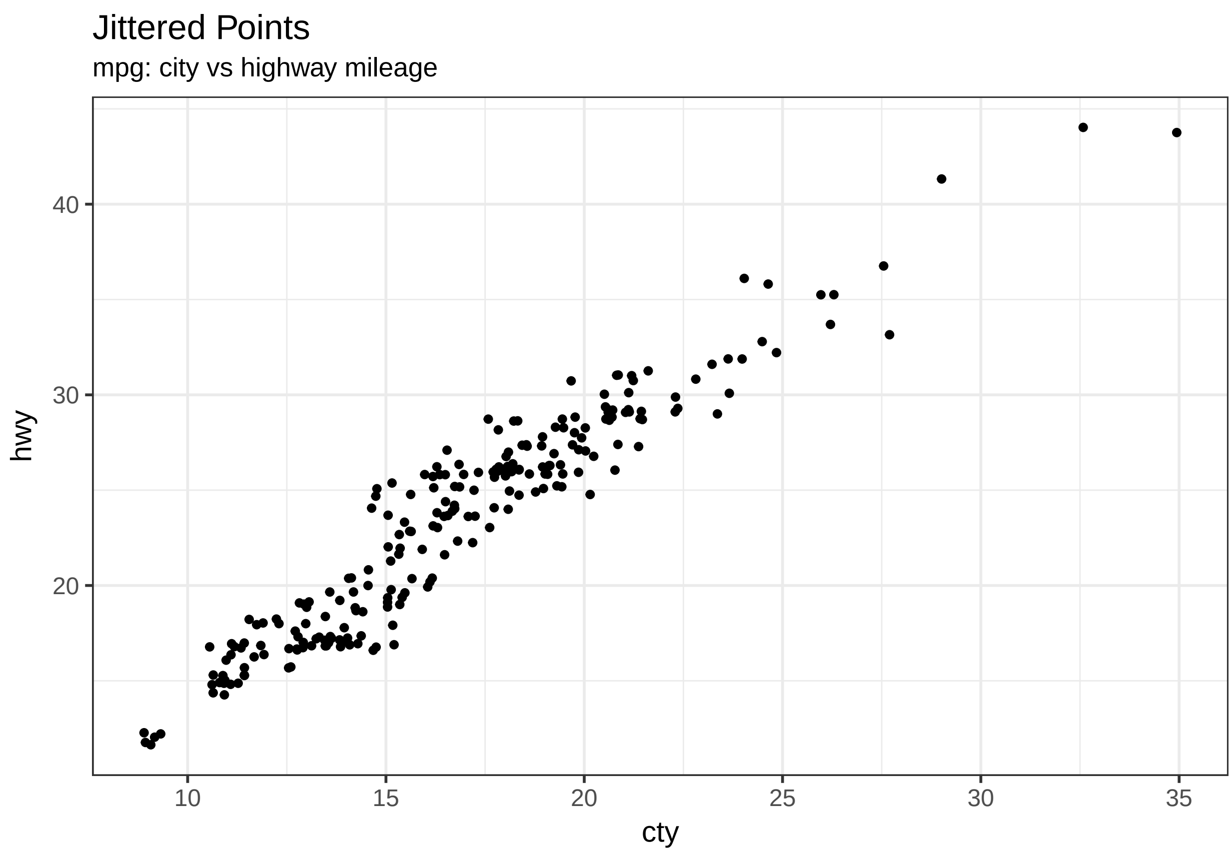
1.3 Jitterplot

```r
library(ggplot2)
data(mpg, package="ggplot2") # alternate source: "http://goo.gl/uEeRGu")
theme_set(theme_bw())  # pre-set the bw theme.

g <- ggplot(mpg, aes(cty, hwy))

# Scatterplot
g + 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")
```



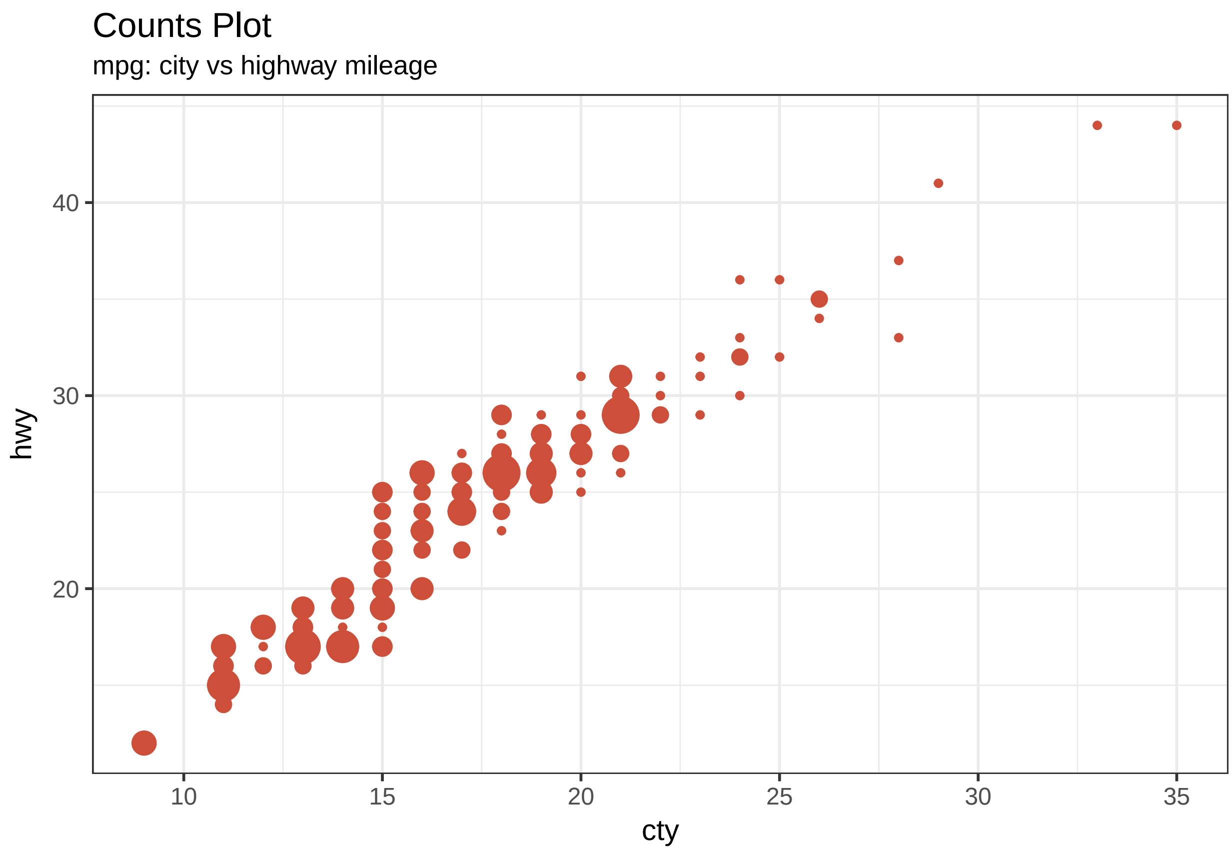
1.3 Jitterplot (2nd version)

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

# Scatterplot
theme_set(theme_bw())  # pre-set the bw theme.
g <- ggplot(mpg, aes(cty, hwy))
g + geom_jitter(width = .5, size=1) +
  labs(subtitle="mpg: city vs highway mileage",
       y="hwy",
       x="cty",
       title="Jittered Points")
```



1.4 Counts Chart

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

# Scatterplot
theme_set(theme_bw())  # pre-set the bw theme.
g <- ggplot(mpg, aes(cty, hwy))
g + geom_count(col="tomato3", show.legend=F) +
  labs(subtitle="mpg: city vs highway mileage",
       y="hwy",
       x="cty",
       title="Counts Plot")
```



1.5 Bubble plot

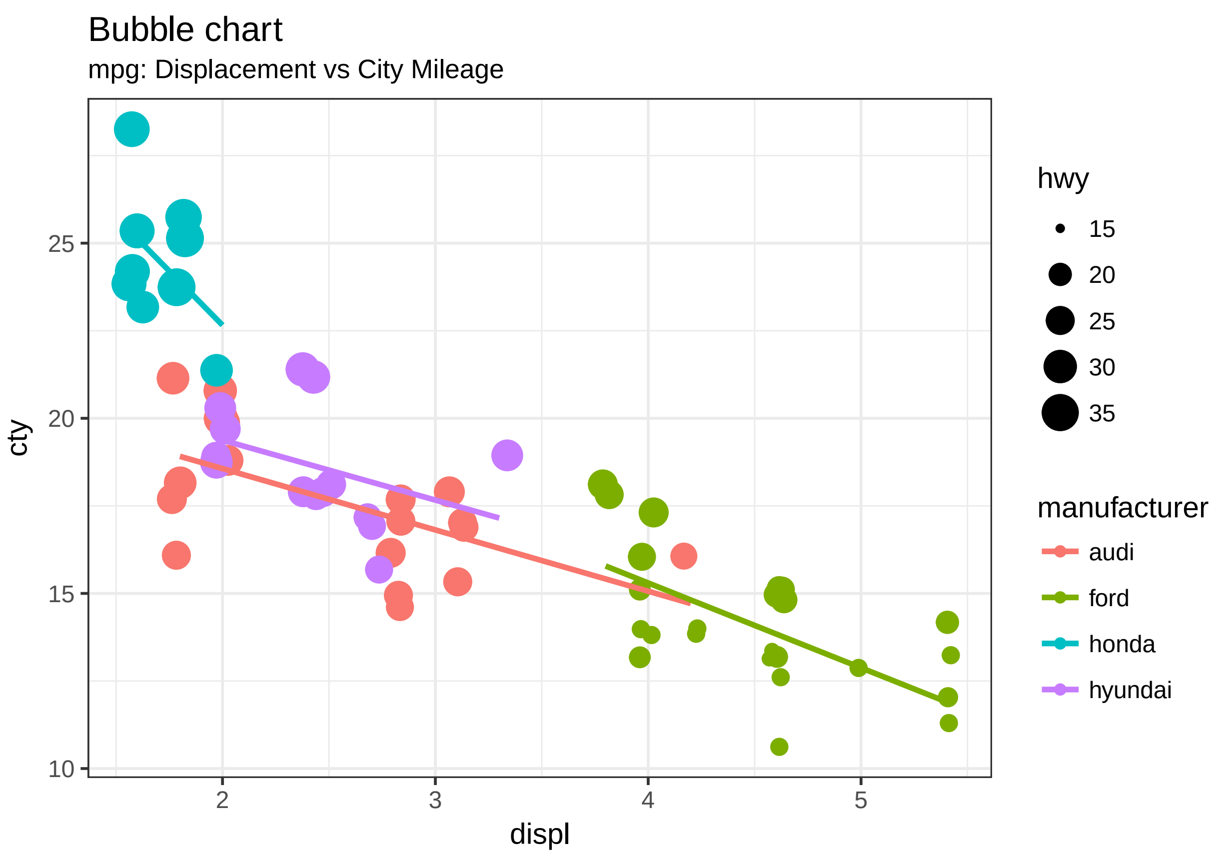
While scatterplot lets you compare the relationship between 2 continuous variables, a bubble chart serves well if you want to understand relationship within the underlying groups based on:

- A Categorical variable (by changing the color) and
- Another continuous variable (by changing the size of points).

Bubble charts are more suitable if you have 4-Dimensional data where two of them are numeric (X and Y) and one other categorical (color) and another numeric variable (size).

1.5 Bubble plot

```
library(ggplot2)
data(mpg, package="ggplot2")
# mpg <- read.csv("http://goo.gl/uEeRGu")
mpg_select <- mpg[mpg$manufacturer %in% c("audi", "ford", "honda", "hyundai"), ]
# Scatterplot
theme_set(theme_bw())  # pre-set the bw theme
g <- ggplot(mpg_select, aes(displ, cty)) +
  labs(subtitle="mpg: Displacement vs City Mileage",title="Bubble chart")
g + geom_jitter(aes(col=manufacturer, size=hwy)) +
  geom_smooth(aes(col=manufacturer), method="lm", se=F)
```

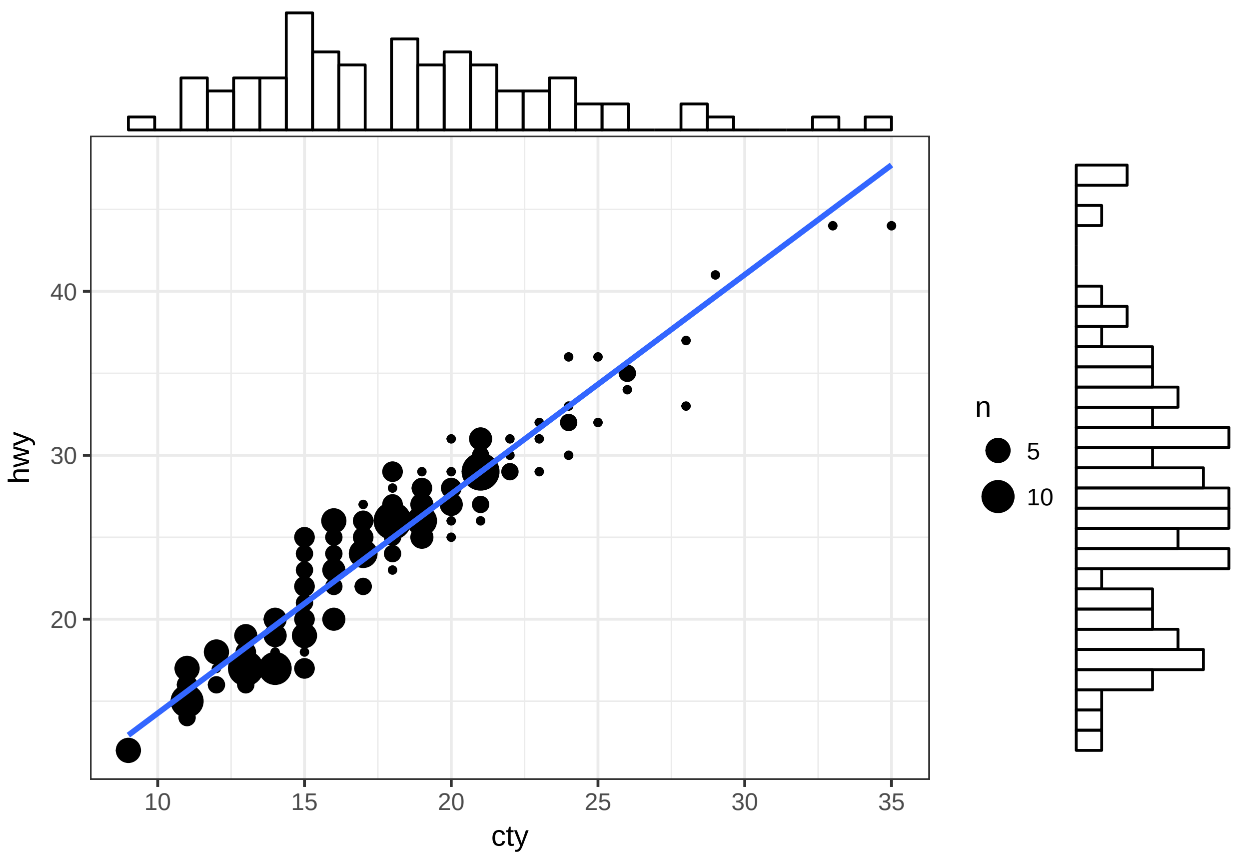


1.6 Marginal Histogram/Boxplot

If you want to show the relationship as well as the distribution in the same chart, use the marginal histogram. It has a histogram of the X and Y variables at the margins of the scatterplot.

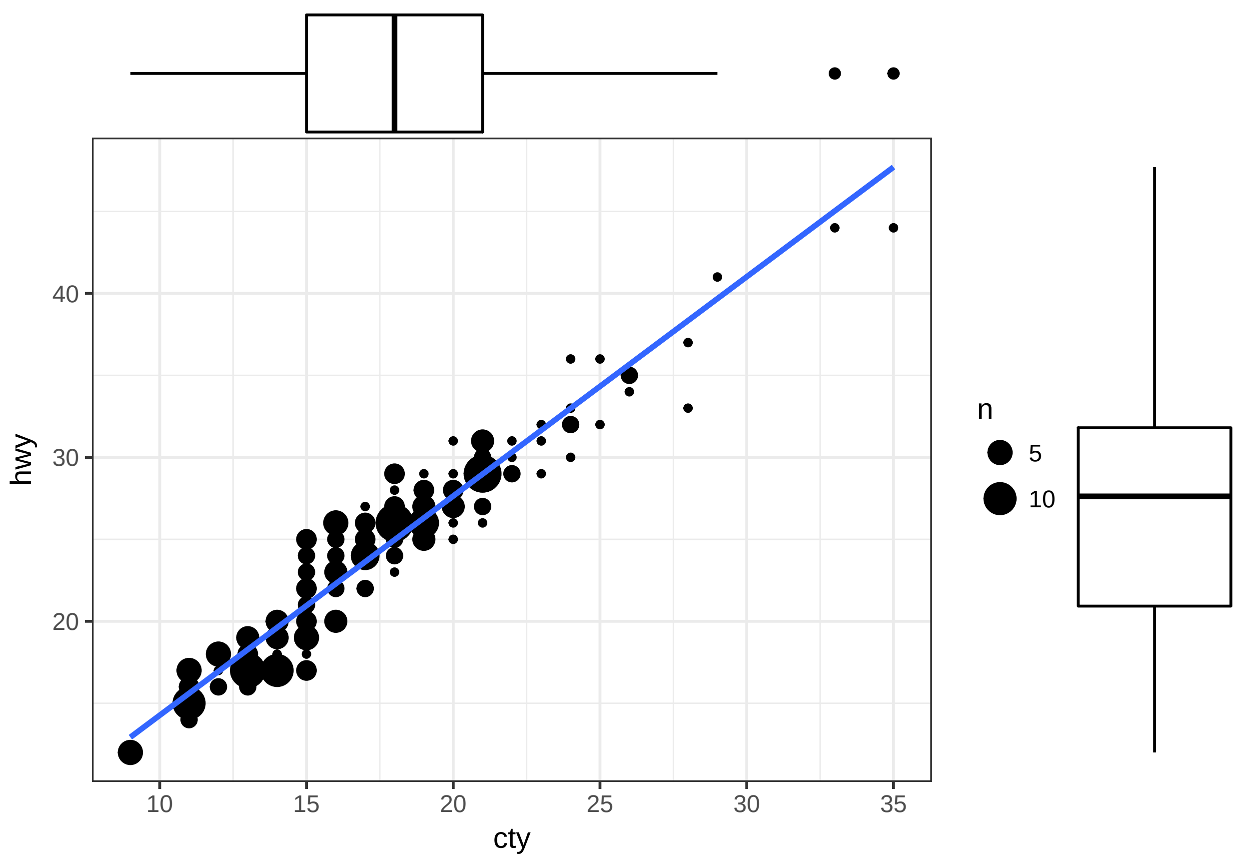
1.6 Marginal Histogram

```
# load package and data
library(ggplot2)
library(ggExtra)
data(mpg, package="ggplot2")
# mpg <- read.csv("http://goo.gl/uEeRGu")
# Scatterplot
theme_set(theme_bw())  # pre-set the bw theme.
mpg_select <- mpg[mpg$hwy >= 35 & mpg$cty > 27, ]
g <- ggplot(mpg, aes(cty, hwy)) +
  geom_count() +
  geom_smooth(method="lm", se=F)

#ggMarginal(g, type = "density", fill="transparent")
ggMarginal(g, type = "histogram", fill="transparent")
```



1.6 Marginal Boxplot

```
# load package and data
library(ggplot2)
library(ggExtra)
data(mpg, package="ggplot2")
# mpg <- read.csv("http://goo.gl/uEeRGu")
# Scatterplot
theme_set(theme_bw())  # pre-set the bw theme.
mpg_select <- mpg[mpg$hwy >= 35 & mpg$cty > 27, ]
g <- ggplot(mpg, aes(cty, hwy)) +
  geom_count() +
  geom_smooth(method="lm", se=F)

#ggMarginal(g, type = "density", fill="transparent")
ggMarginal(g, type = "boxplot", fill="transparent")
```

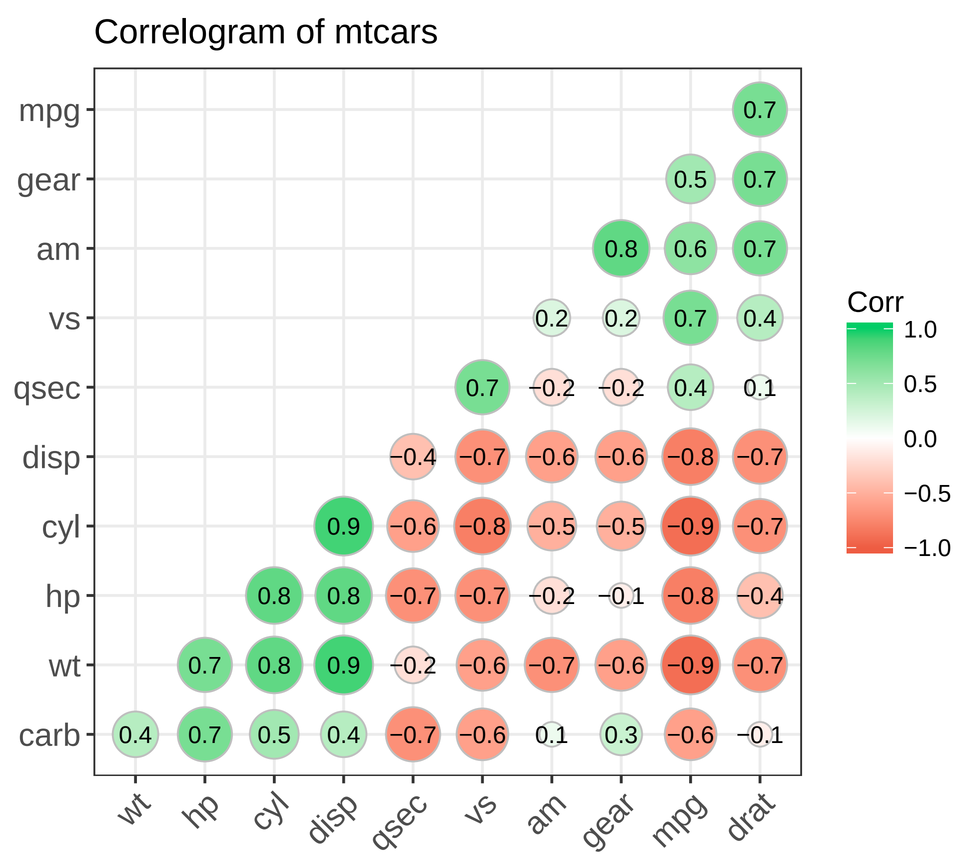


1.7 Correlogram

Correlogram let's you examine the corellation of multiple continuous variables present in the same dataframe.

1.7 Correlogram

```r
library(ggplot2)
library(ggcorrplot)

# Correlation matrix
data(mtcars)
corr <- round(cor(mtcars), 1)

# Plot
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)
```

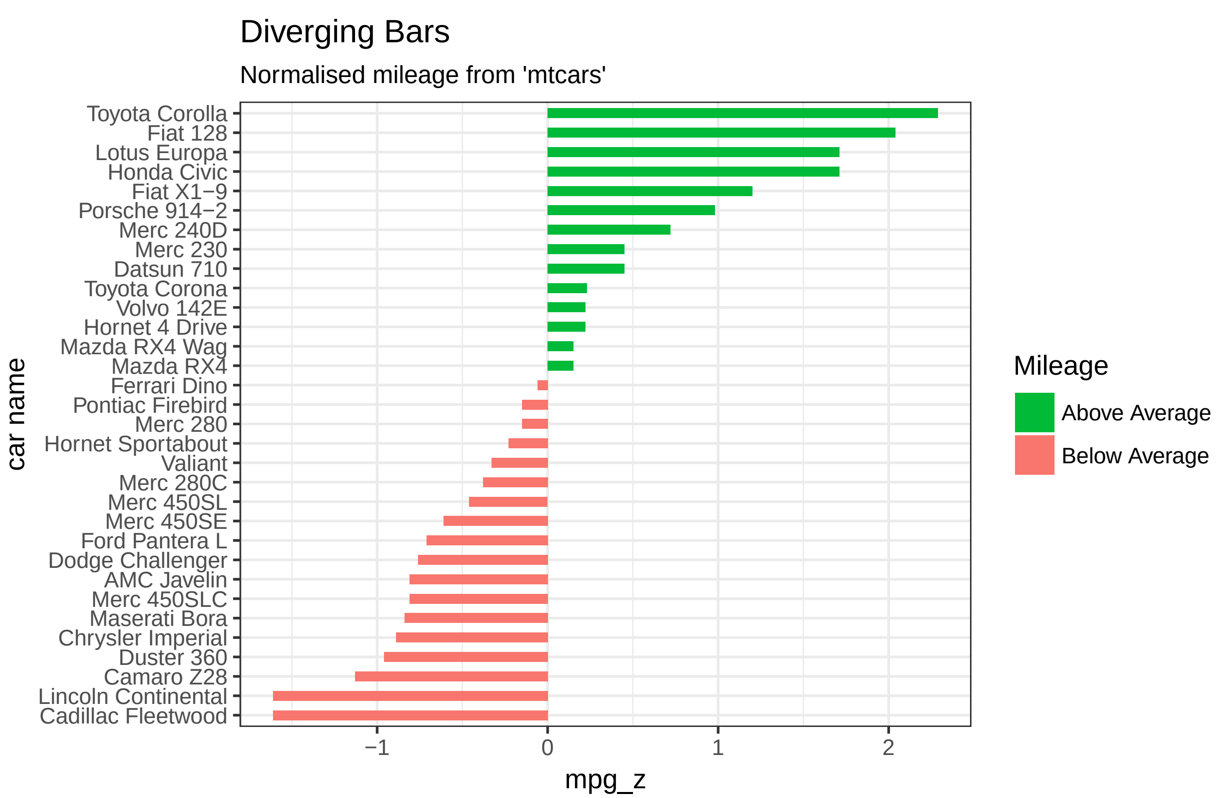


2. Deviation

Compare variation in values between small number of items (or categories) with respect to a fixed reference.

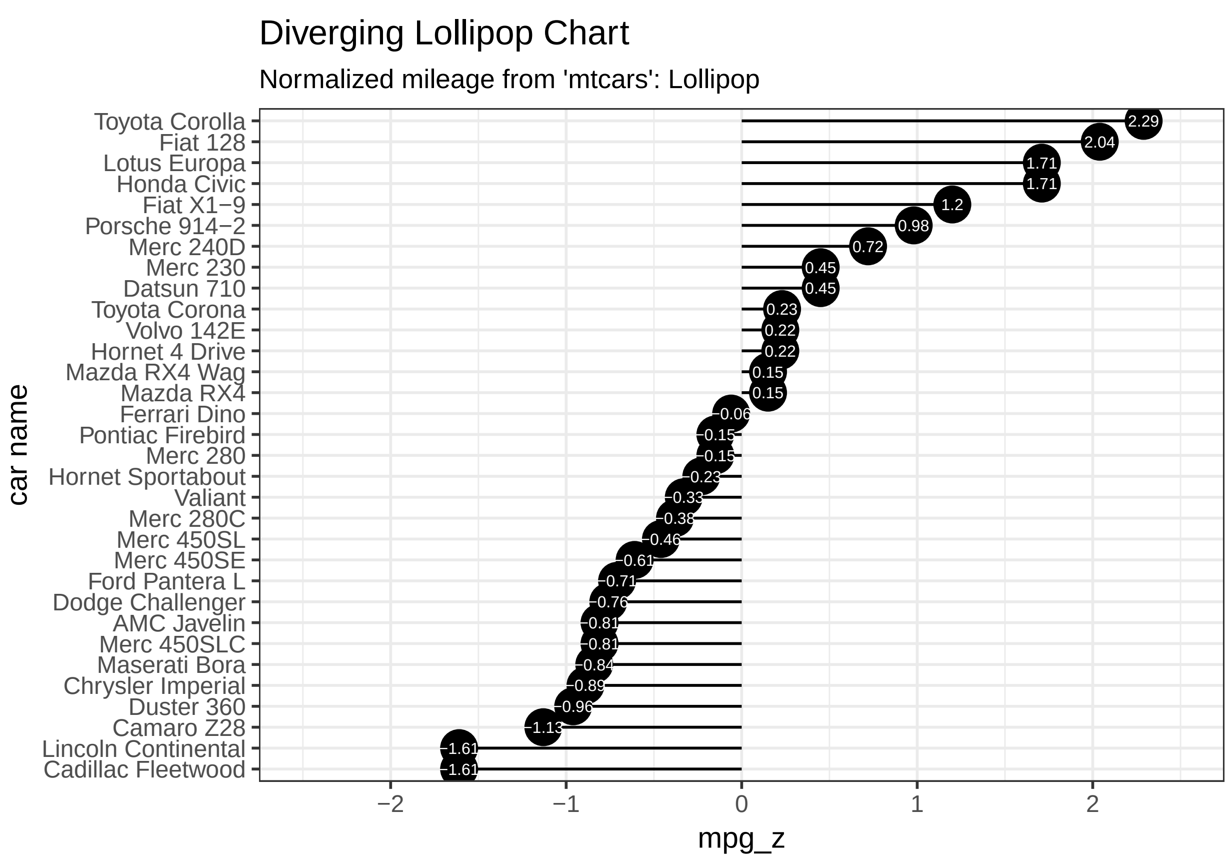
2.1 Diverging bars

```
library(ggplot2)
theme_set(theme_bw())
# Data Prep
data("mtcars")  # load data
mtcars$`car name` <- rownames(mtcars)  # create new column for car names
# Compute normalized mpg
mtcars$mpg_z <- round((mtcars$mpg - mean(mtcars$mpg))/sd(mtcars$mpg), 2)
mtcars$mpg_type <- ifelse(mtcars$mpg_z < 0, "below", "above")  # above/below avg flag
mtcars <- mtcars[order(mtcars$mpg_z), ]  # sort
# Convert to factor to retain sorted order in plot
mtcars$`car name` <- factor(mtcars$`car name`, levels = mtcars$`car name`)
# 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()
```



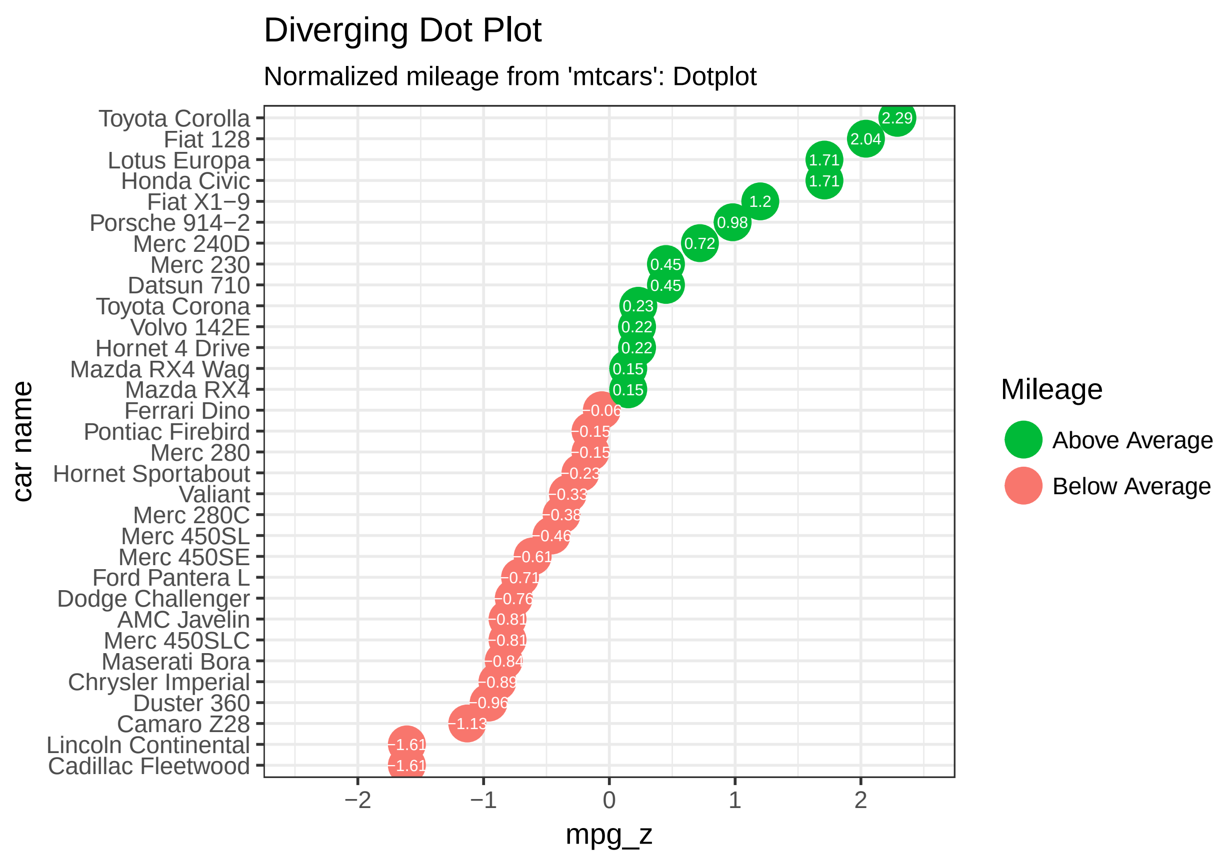
2.2 Diverging Lollipop chart

```
library(ggplot2)
theme_set(theme_bw())

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()
```



2.3 Diverging Dot Plot

```
library(ggplot2)
theme_set(theme_bw())

# Plot
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()
```

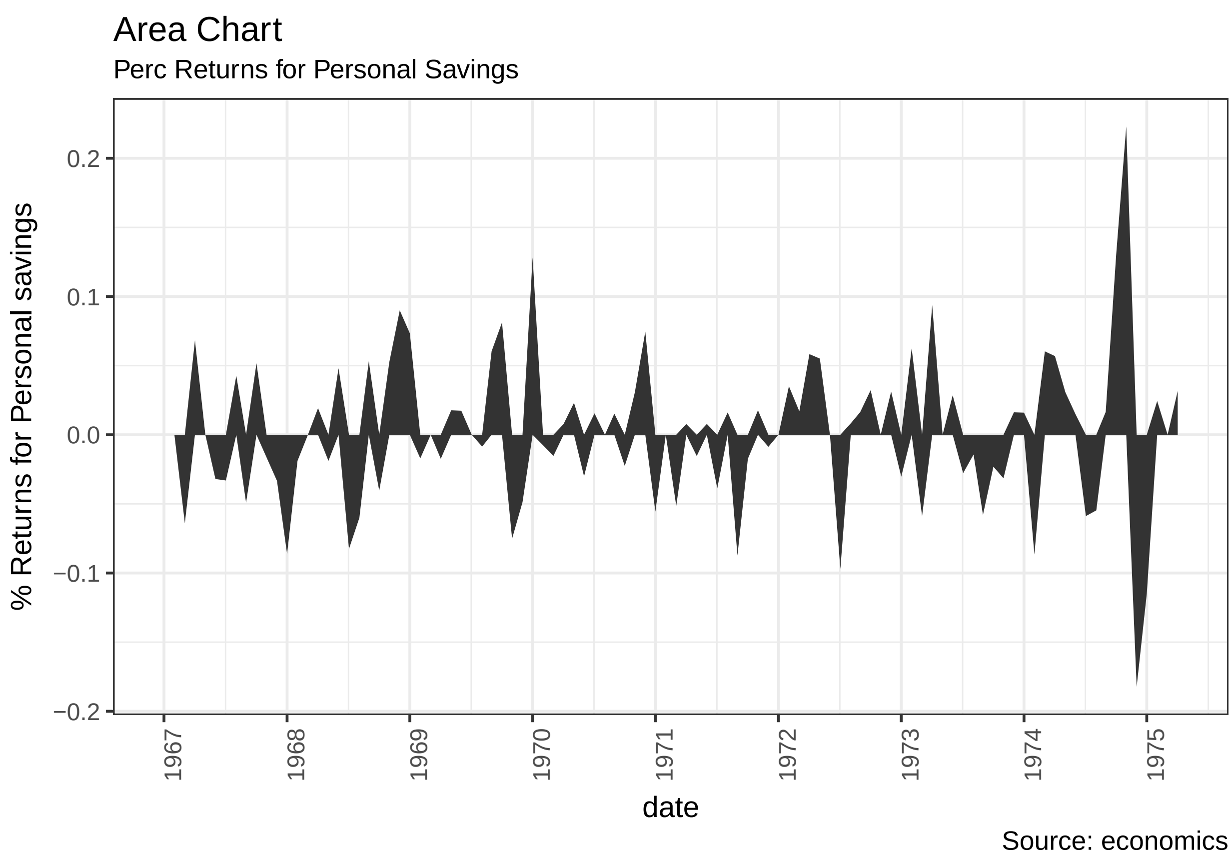


2.4 Area Chart

Area charts are typically used to visualize how a particular metric (such as % returns from a stock) performed compared to a certain baseline.

Area Chart

```
library(ggplot2)
library(quantmod)
data("economics", package = "ggplot2")

# Compute % Returns
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")
```

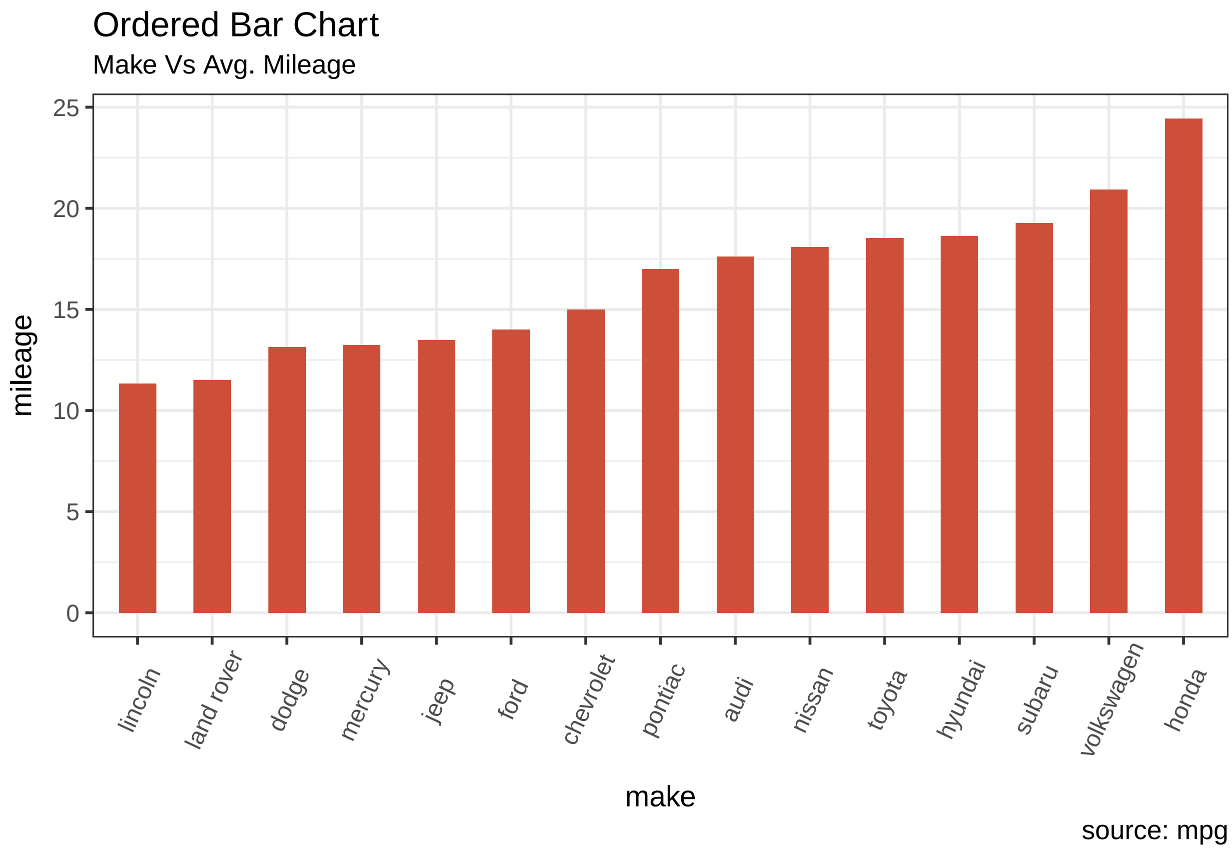


3. Ranking

Used to compare the position or performance of multiple items with respect to each other. Actual values matters somewhat less than the ranking.

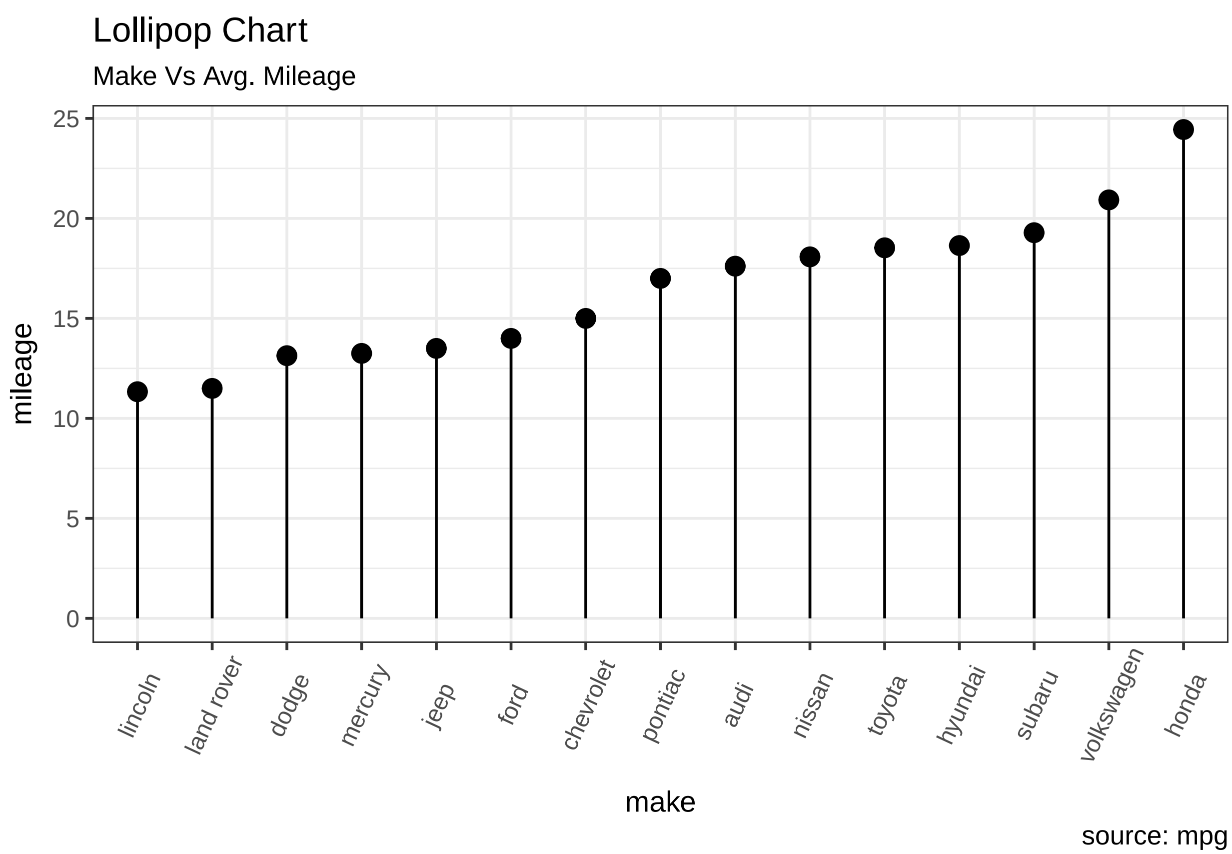
3.1 Ordered Bar Chart

```
# 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
library(ggplot2)
theme_set(theme_bw())

# Draw 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))
```



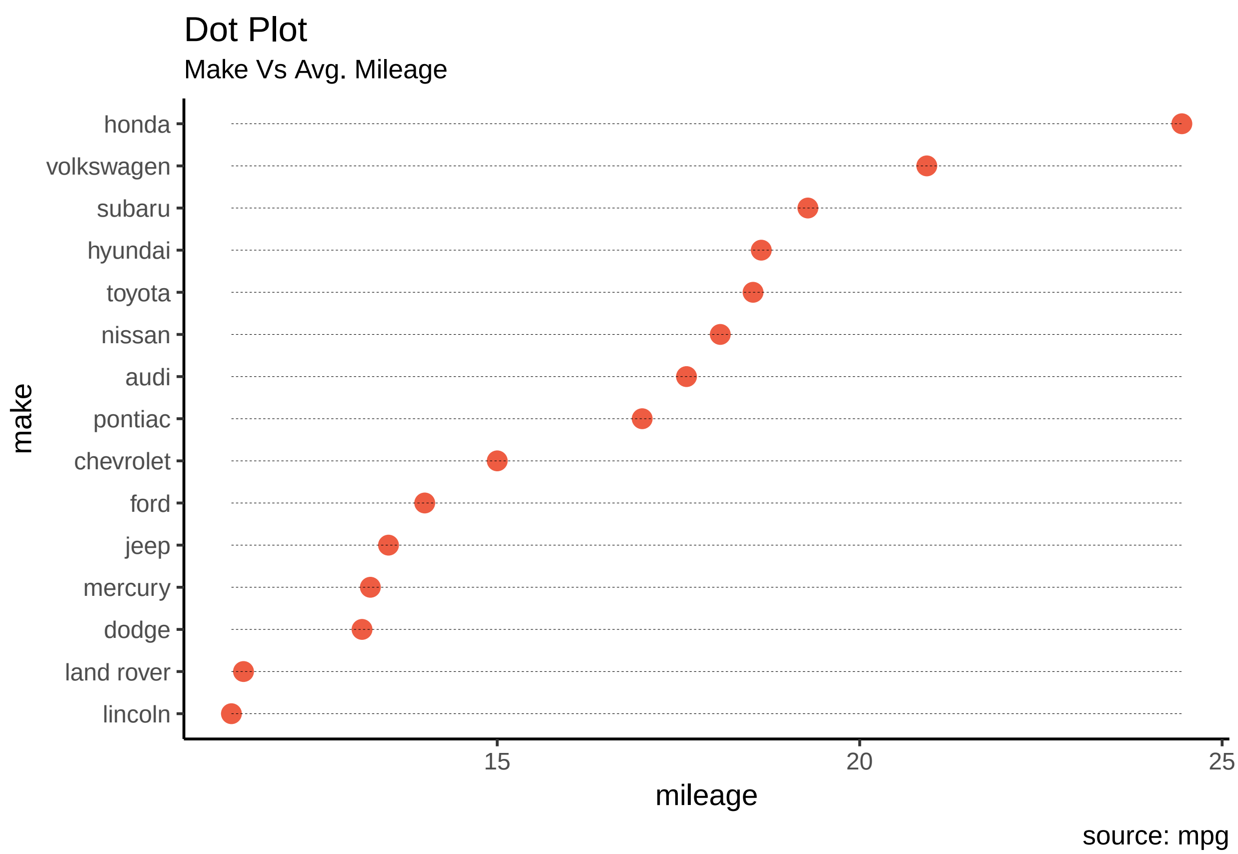
3.2 Lollipop Chart

```
library(ggplot2)
theme_set(theme_bw())

# Plot
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))
```



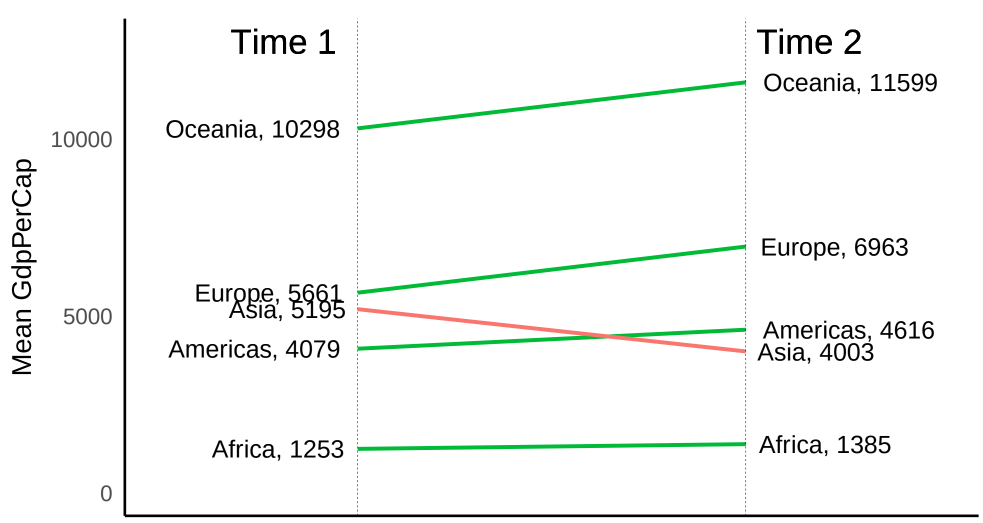
3.3 Dot plot

```
library(ggplot2)
library(scales)
theme_set(theme_classic())

# 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()
```



3.4 Slope Chart

```
library(ggplot2)
library(scales)
theme_set(theme_classic())

# 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$`1952`) < 0, "red", "green")
# Plot
p <- ggplot(df) + geom_segment(aes(x=1, xend=2, y=`1952`, yend=`1957`, col=class),
                               size=.75, show.legend=F) +
                  geom_vline(xintercept=1, linetype="dashed", size=.1) +
                  geom_vline(xintercept=2, linetype="dashed", size=.1) +
                  scale_color_manual(labels = c("Up", "Down"),
                                     values = c("green"="#00ba38", "red"="#f8766d")) +
                  labs(x="", y="Mean GdpPerCap") +  # color of lines and axes labels
                  xlim(.5, 2.5) + ylim(0,(1.1*(max(df$`1952`, df$`1957`)))) # axes limits
# Add texts
p <- p + geom_text(label=left_label, y=df$`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$`1957`)), hjust=1.2, size=5)
p <- p + geom_text(label="Time 2", x=2, y=1.1*(max(df$`1952`, df$`1957`)), hjust=-0.1, size=5)
# Minify theme
p + theme(panel.background = element_blank(),panel.grid = element_blank(),
          axis.ticks = element_blank(),axis.text.x = element_blank(),
          panel.border = element_blank(),plot.margin = unit(c(1,2,1,2), "cm"))
```



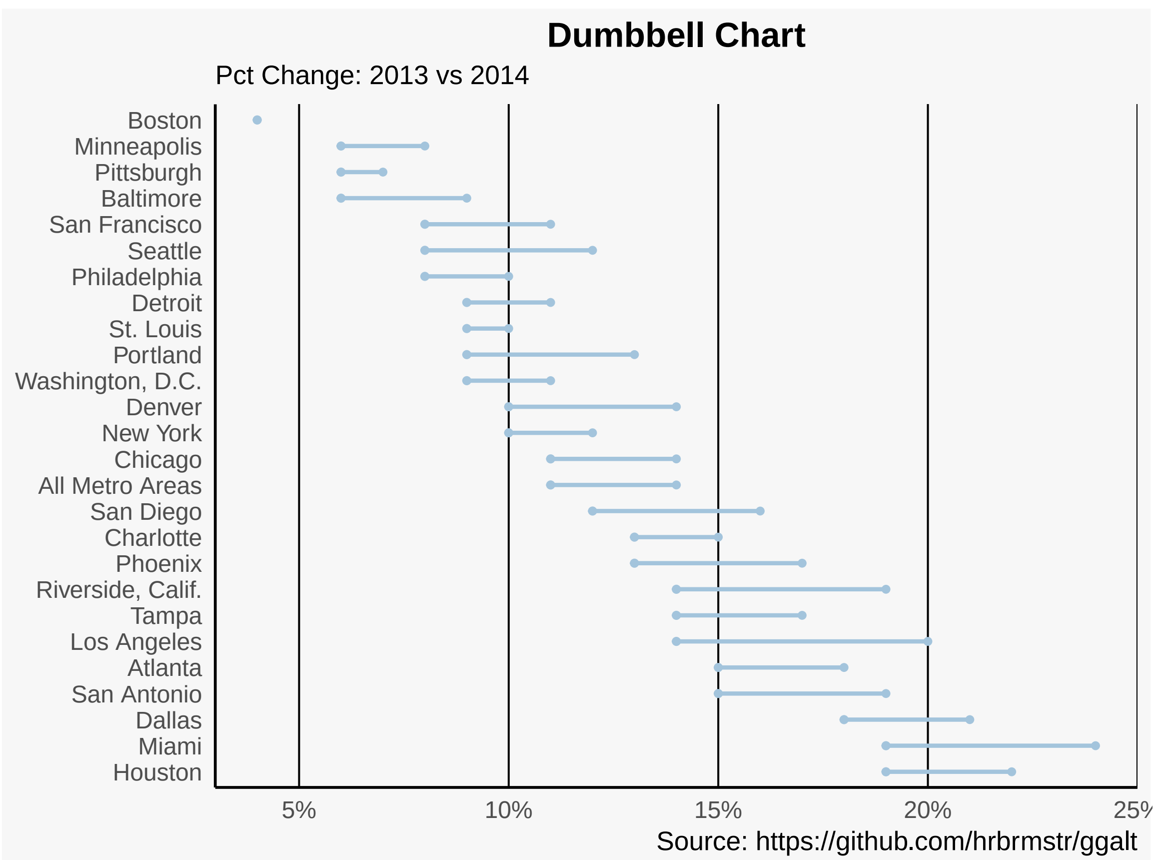
3.5 Dumbbell Plot

Dumbbell charts are a great tool if you wish to:

- Visualise relative positions (like growth and decline) between two points in time.
- Compare distance between two categories.

Dumbbell Plot

```
library(ggplot2)
library(ggalt)
theme_set(theme_classic())

health <- read.csv("https://raw.githubusercontent.com/selva86/datasets/master/health.csv")
health$Area <- factor(health$Area, levels=as.character(health$Area)) # right ordering
# health$Area <- factor(health$Area)
gg <- ggplot(health, aes(x=pct_2013, xend=pct_2014, y=Area, group=Area)) +
        geom_dumbbell(color="#a3c4dc",
                      size=0.75,
                      point.colour.l="#0e668b") +
        scale_x_continuous(label=percent) +
        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"),panel.grid.minor=element_blank(),
              panel.grid.major.y=element_blank(),panel.grid.major.x=element_line(),
              axis.ticks=element_blank(),legend.position="top",panel.border=element_blank())
plot(gg)
```

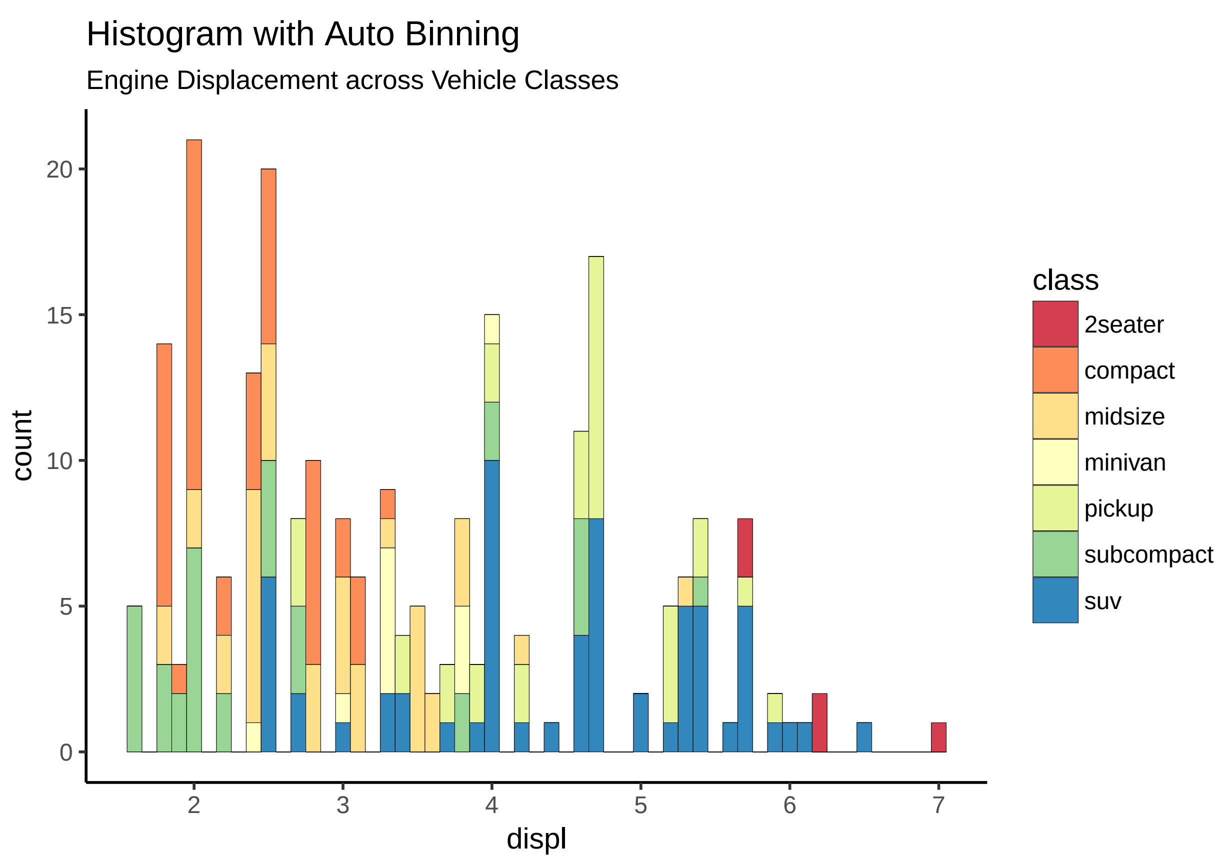


4. Distribution

- When you have a lot of data points and want to study where and how the data points are distributed.

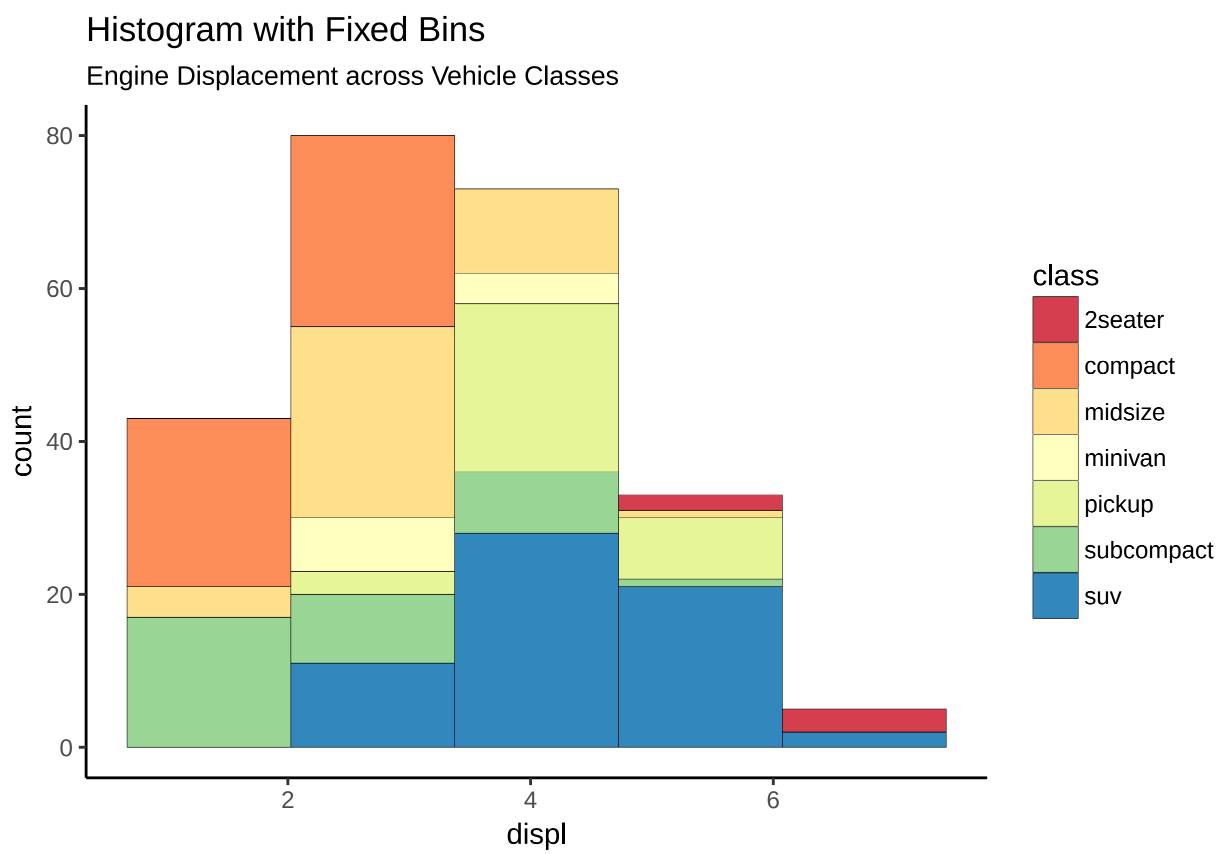
4.1 Histogram (with automatic binning)

```
library(ggplot2)
theme_set(theme_classic())

# 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")
```



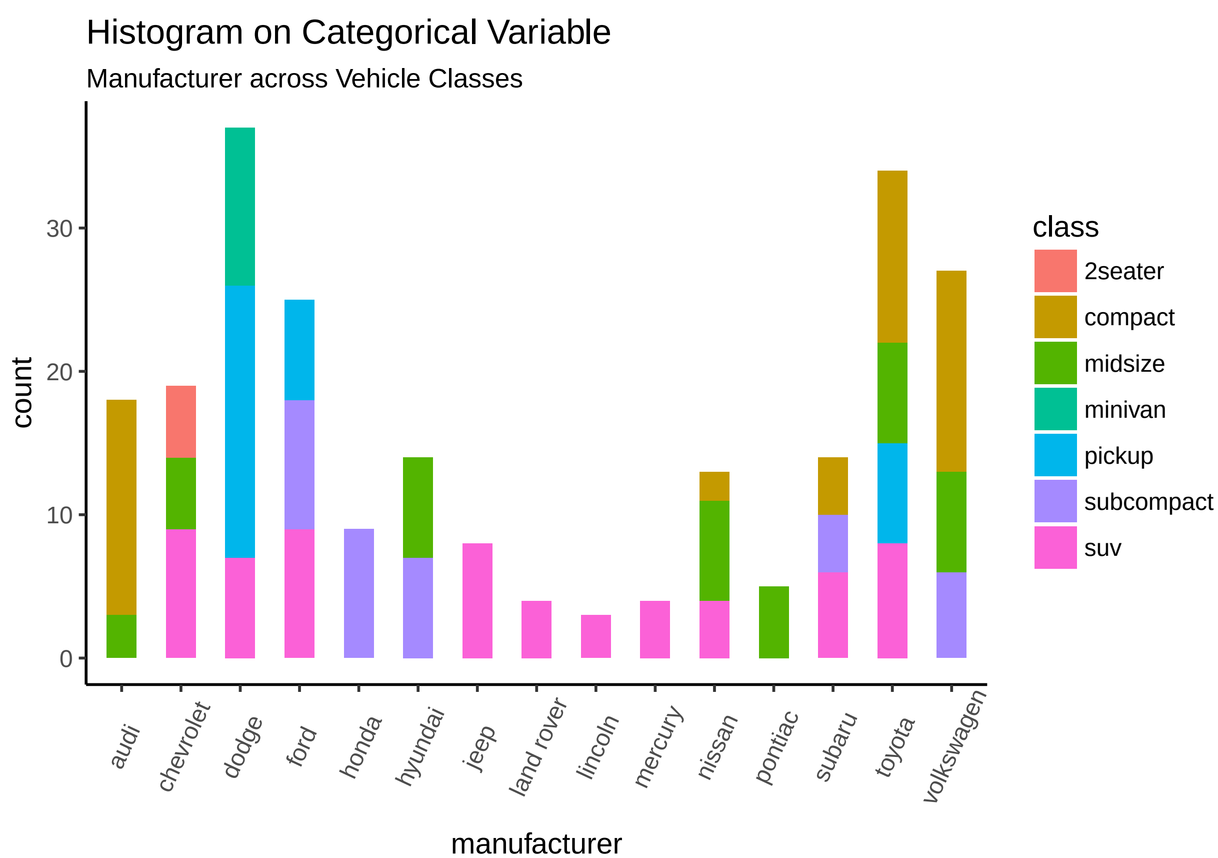
4.1 Histogram (with fixed binning)

```
library(ggplot2)
g + geom_histogram(aes(fill=class),
                   bins=5,
                   col="black",
                   size=.1) +   # change number of bins
  labs(title="Histogram with Fixed Bins",
       subtitle="Engine Displacement across Vehicle Classes")
```



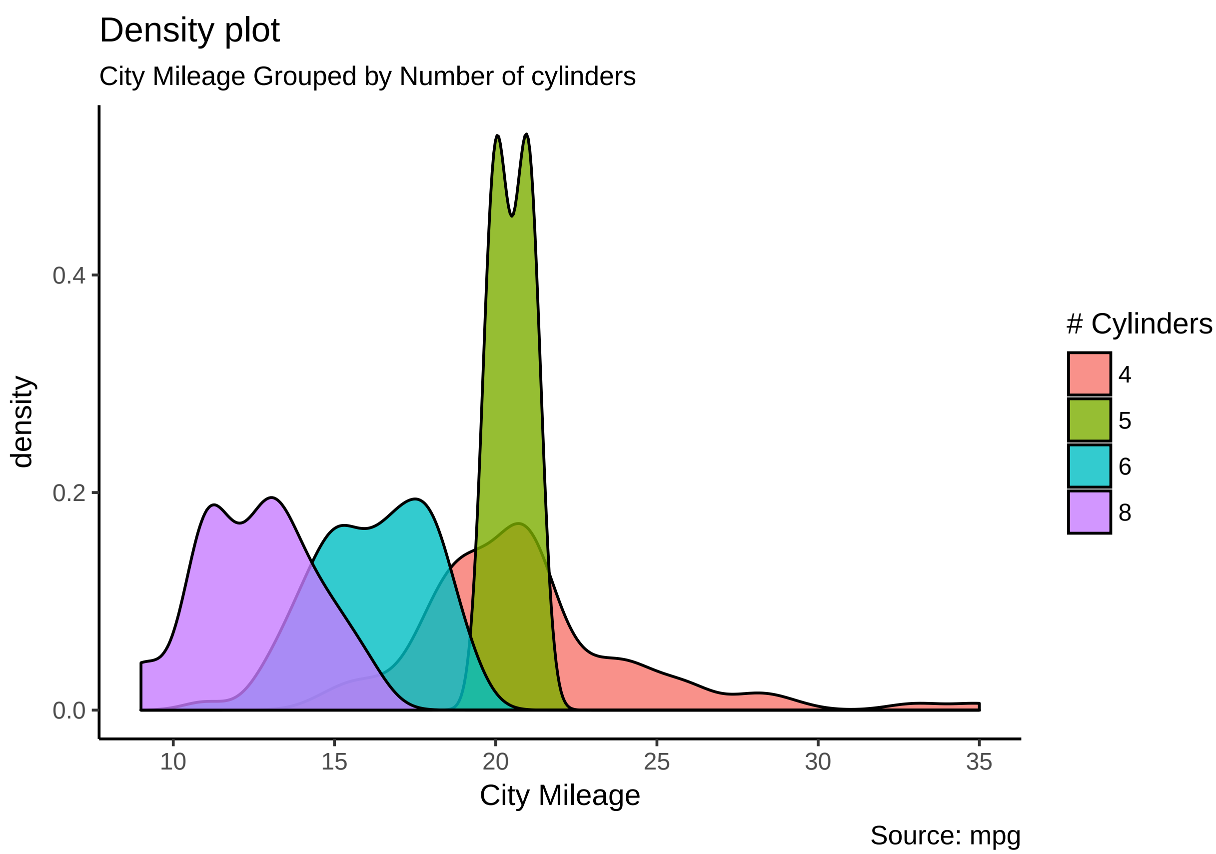
4.1 Histogram (on a categorical variable)

```
library(ggplot2)
theme_set(theme_classic())

# Histogram on a Categorical variable
g <- ggplot(mpg, aes(manufacturer))
g + geom_bar(aes(fill=class), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Histogram on Categorical Variable",
       subtitle="Manufacturer across Vehicle Classes")
```



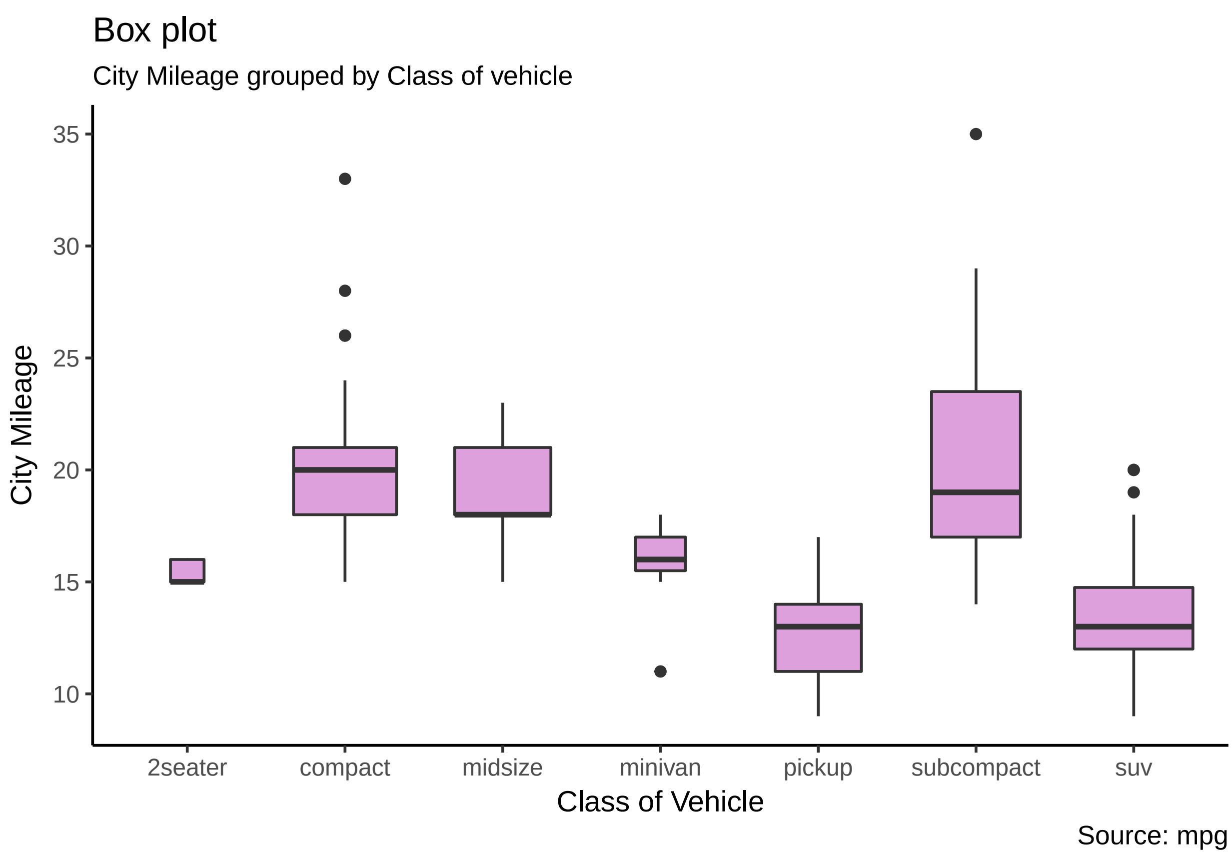
4.2 Density plot

```
library(ggplot2)
theme_set(theme_classic())

# Plot
g <- ggplot(mpg, aes(cty))
g + geom_density(aes(fill=factor(cyl)), alpha=0.8) +
    labs(title="Density plot",
         subtitle="City Mileage Grouped by Number of cylinders",
         caption="Source: mpg",
         x="City Mileage",
         fill="# Cylinders")
```



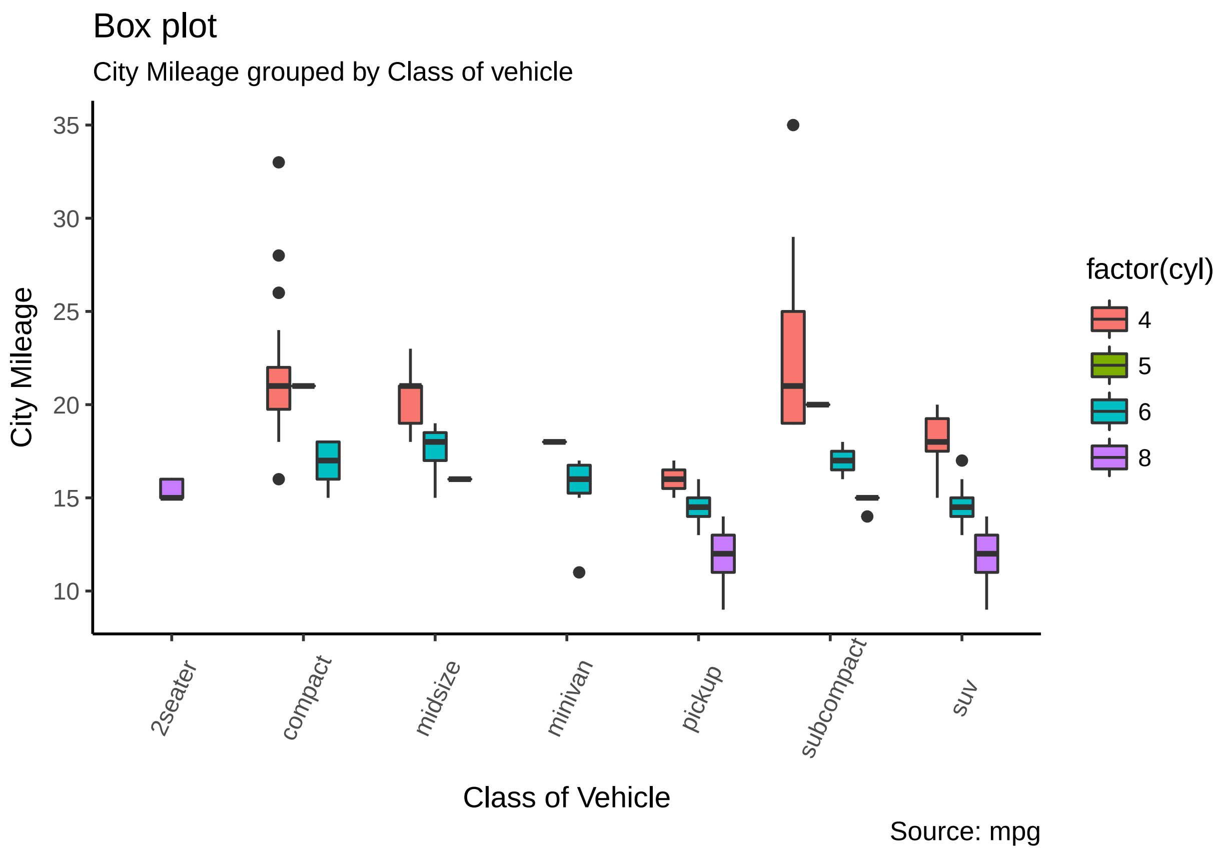
4.3 Boxplot

```
library(ggplot2)
theme_set(theme_classic())

# Plot
g <- ggplot(mpg, aes(class, cty))
g + geom_boxplot(varwidth=T, fill="plum") +
    labs(title="Box plot",
         subtitle="City Mileage grouped by Class of vehicle",
         caption="Source: mpg",
         x="Class of Vehicle",
         y="City Mileage")
```



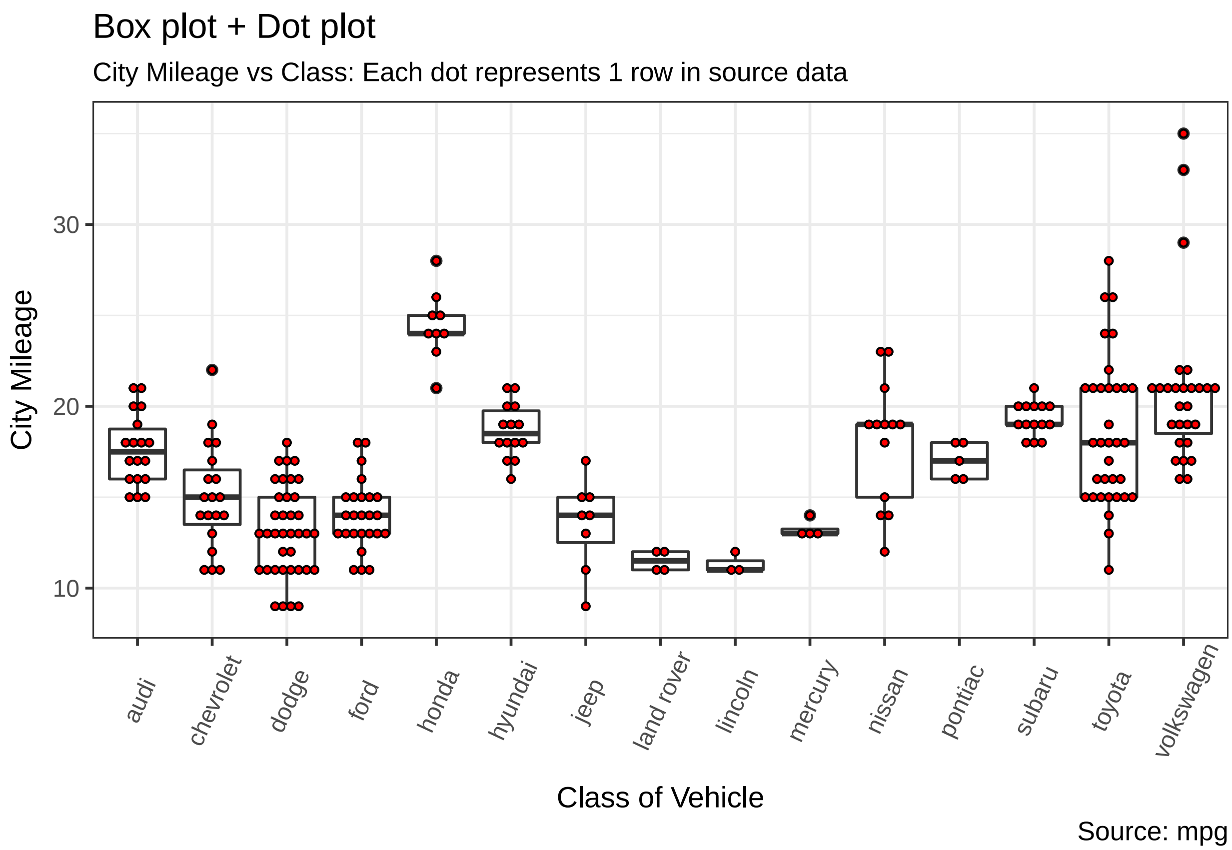
4.3 Boxplot (2nd version)

```r
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")
```



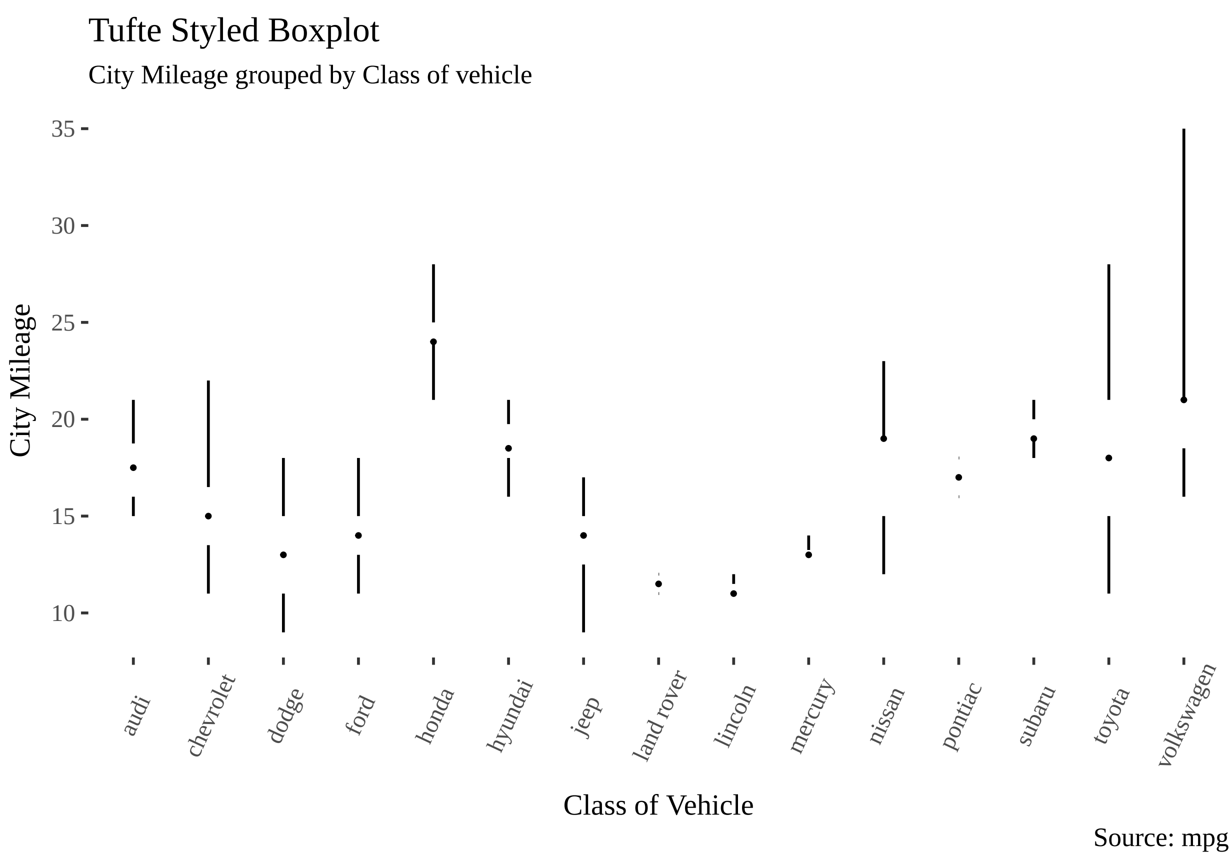
4.4 Dot + Boxplot

```
library(ggplot2)
theme_set(theme_bw())

# plot
g <- ggplot(mpg, aes(manufacturer, cty))
g + geom_boxplot() +
  geom_dotplot(binaxis='y',
               stackdir='center',
               dotsize = .5,
               fill="red") +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Box plot + Dot plot",
       subtitle="City Mileage vs Class: Each dot represents 1 row in source data",
       caption="Source: mpg",
       x="Class of Vehicle",
       y="City Mileage")
```



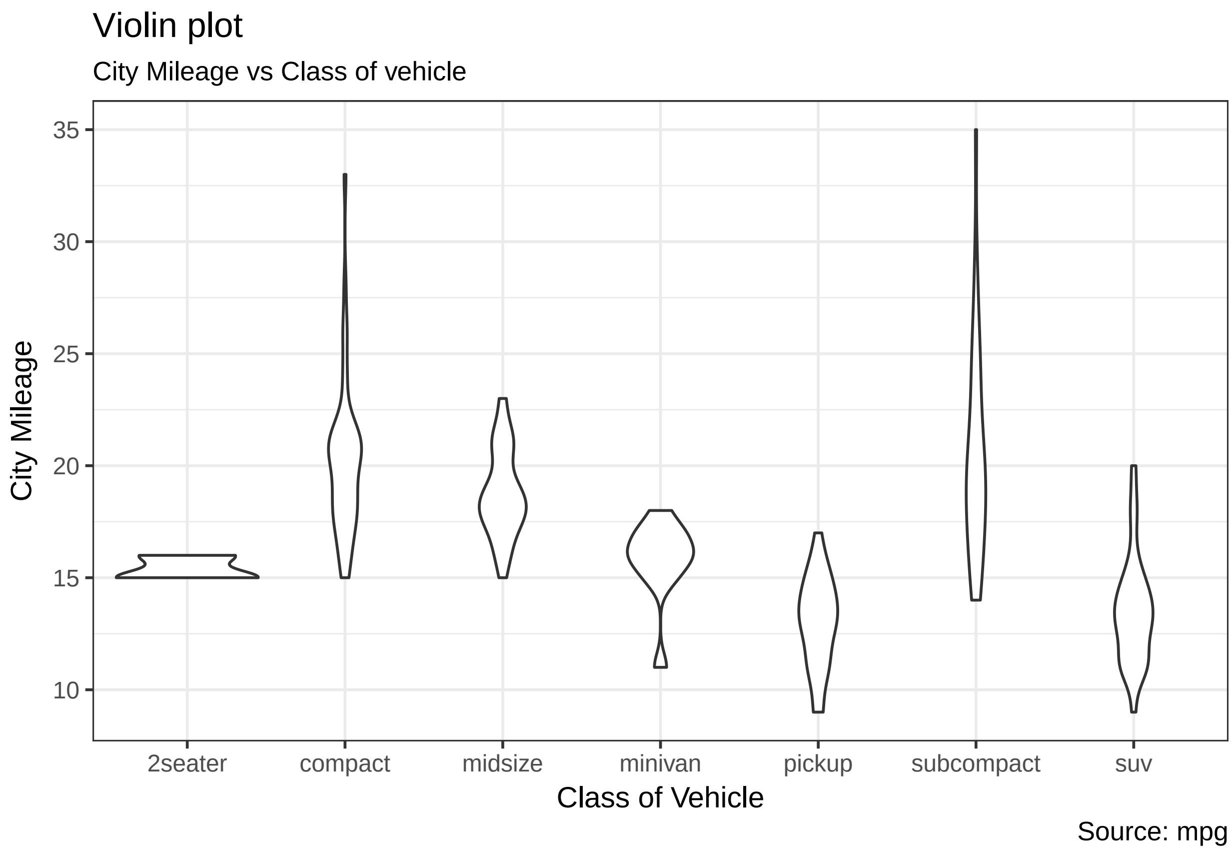
4.5 Tufte's Boxplot

```
library(ggthemes)
library(ggplot2)
theme_set(theme_tufte())  # from ggthemes

# plot
g <- ggplot(mpg, aes(manufacturer, cty))
g + geom_tufteboxplot() +
      theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
      labs(title="Tufte Styled Boxplot",
           subtitle="City Mileage grouped by Class of vehicle",
           caption="Source: mpg",
           x="Class of Vehicle",
           y="City Mileage")
```



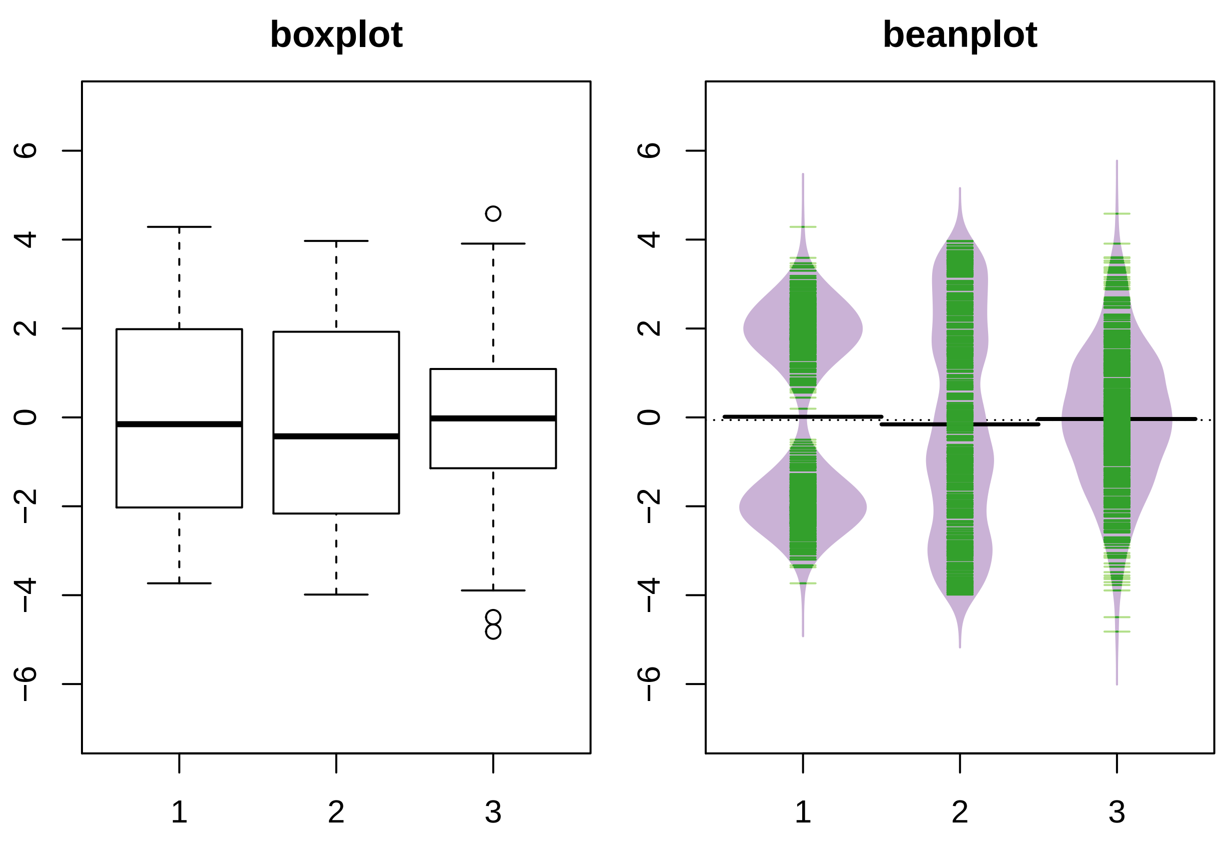
4.6 Violin Plot

```r
library(ggplot2)
theme_set(theme_bw())

# plot
g <- ggplot(mpg, aes(class, cty))
g + geom_violin() +
  labs(title="Violin plot",
       subtitle="City Mileage vs Class of vehicle",
       caption="Source: mpg",
       x="Class of Vehicle",
       y="City Mileage")
```



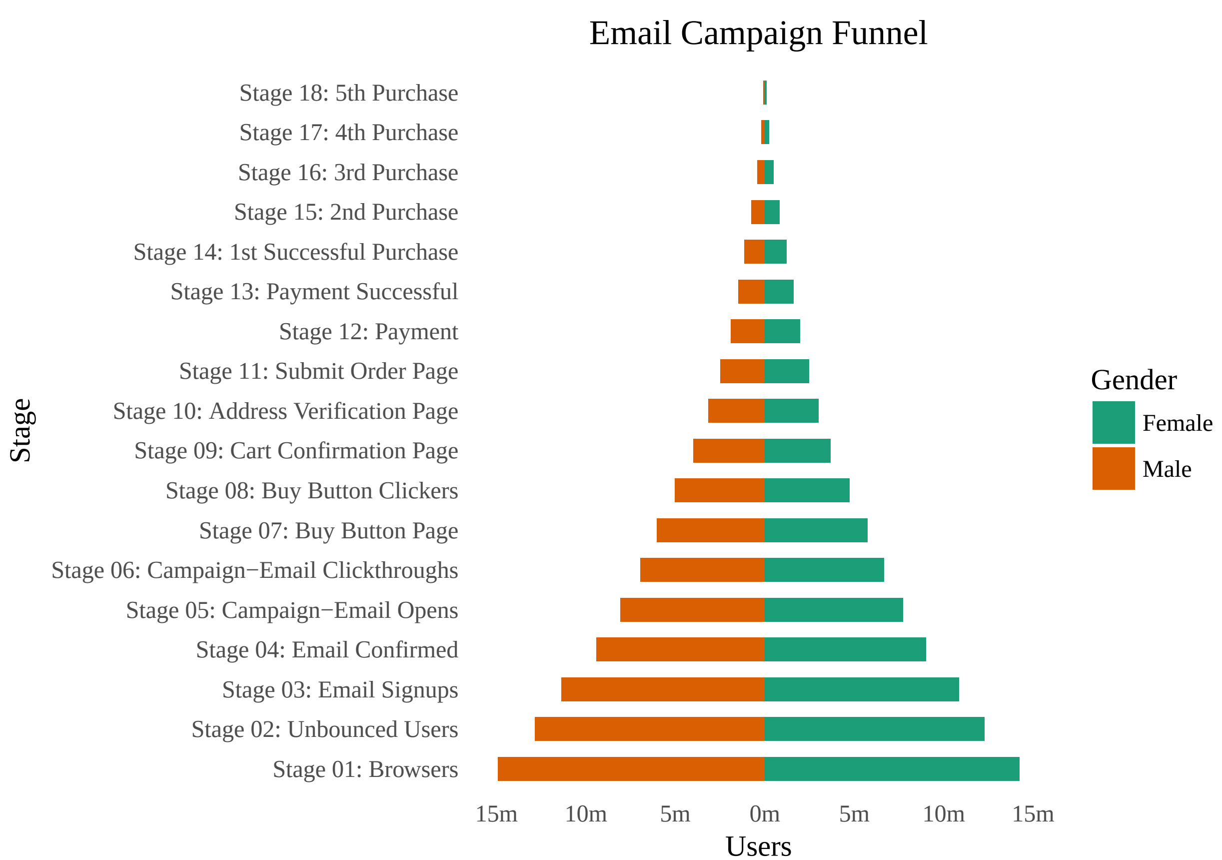
4.7 Bean Plot

```
library(beanplot)
set.seed(1)
 par(mfrow = c(1, 2), mai = c(0.5, 0.5, 0.5, 0.1))
 mu <- 2
 si <- 0.6
 c <- 500
 bimodal <- c(rnorm(c/2, -mu, si), rnorm(c/2, mu, si))
 uniform <- runif(c, -4, 4)
 normal <- rnorm(c, 0, 1.5)
 ylim <- c(-7, 7)
 boxplot(bimodal, uniform, normal, ylim = ylim, main = "boxplot", names = 1:3)
 beanplot(bimodal, uniform, normal, ylim = ylim, main = "beanplot",
          col = c("#CAB2D6", "#33A02C",
                  "#B2DF8A"), border = "#CAB2D6")
```



4.8 Population Pyramid

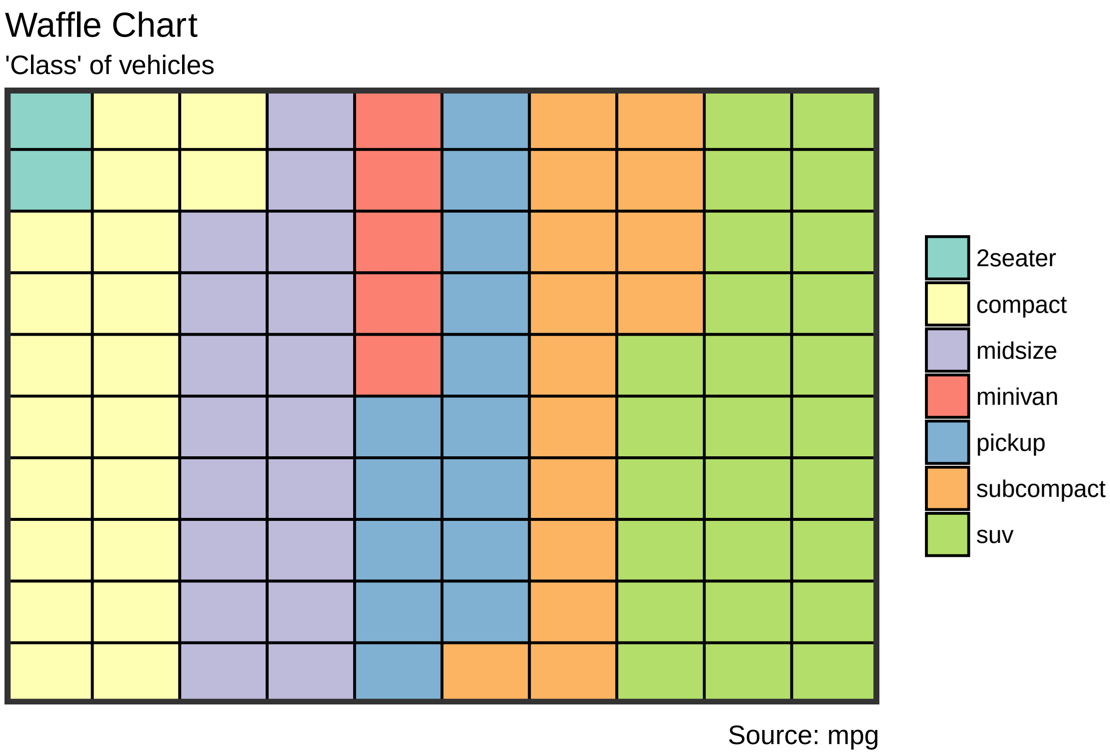
```
library(ggplot2)
library(ggthemes)
options(scipen = 999)  # turns of scientific notations like 1e+40

# Read data
email_campaign_funnel <-
  read.csv("https://raw.githubusercontent.com/selva86/datasets/master/email_campaign_funnel.csv")

# 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")  # colour palette
```



5. Composition

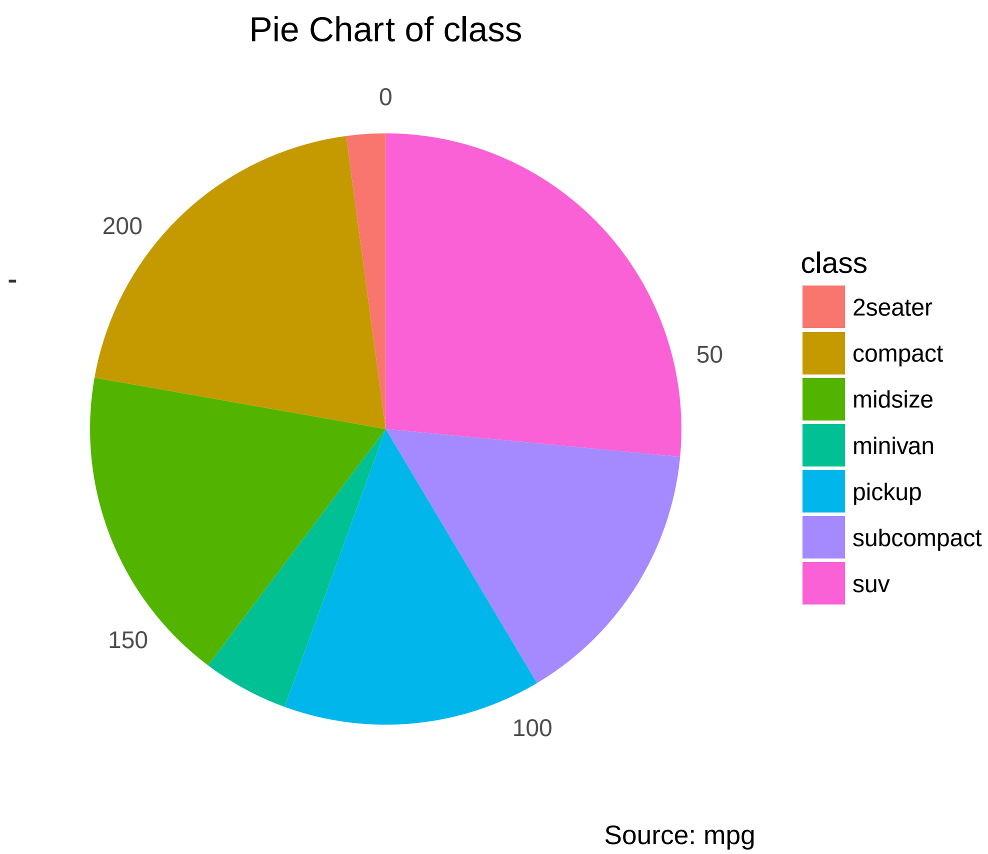
5.1 Waffle Chart

```r
var <- mpg$class  # categorical data

# Prep data (nothing to change here)
nrows <- 10
df <- expand.grid(y = 1:nrows, x = 1:nrows)
categ_table <- round(table(var) * ((nrows*nrows)/(length(var))))
df$category <- factor(rep(names(categ_table), categ_table))
# NOTE: if sum(categ_table) is not 100 (i.e. nrows^2),
# it will need adjustment to make the sum to 100
# Plot
ggplot(df, aes(x = x, y = y, fill = category)) +
        geom_tile(color = "black", size = 0.5) +
        scale_x_continuous(expand = c(0, 0)) +
        scale_y_continuous(expand = c(0, 0), trans = 'reverse') +
        scale_fill_brewer(palette = "Set3") +
        labs(title="Waffle Chart", subtitle="'Class' of vehicles",
             caption="Source: mpg") +
        theme(panel.border = element_rect(size = 2),
              plot.title = element_text(size = rel(1.2)),
              axis.text = element_blank(),axis.title = element_blank(),
              axis.ticks = element_blank(), legend.title = element_blank(),
              legend.position = "right")
```



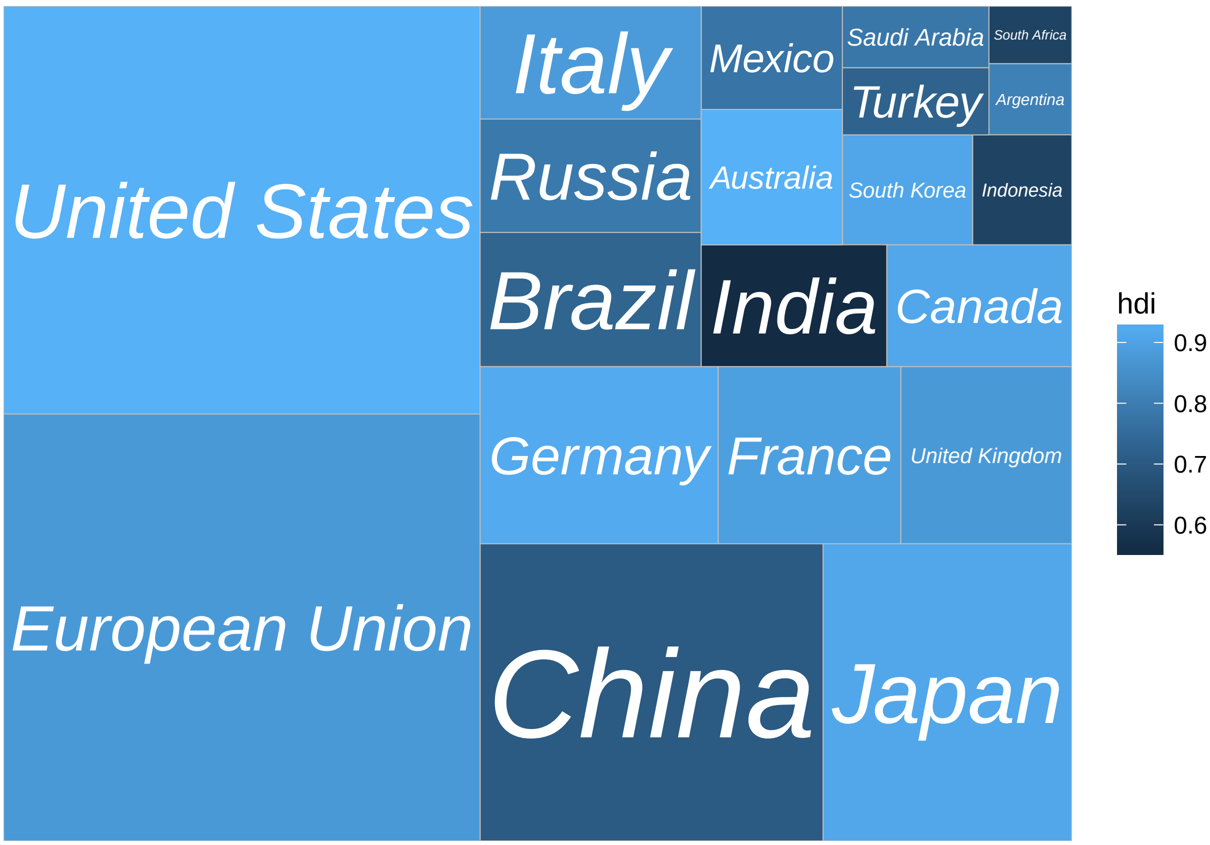
5.2 Pie Chart

```
library(ggplot2)
theme_set(theme_classic())

# Source: Frequency table
df <- as.data.frame(table(mpg$class))
colnames(df) <- c("class", "freq")
pie <- ggplot(df, aes(x = "", y=freq, fill = factor(class))) +
  geom_bar(width = 1, stat = "identity") +
  theme(axis.line = element_blank(),
        plot.title = element_text(hjust=0.5)) +
  labs(fill="class",
       x=NULL,
       y=NULL,
       title="Pie Chart of class",
       caption="Source: mpg")

pie + coord_polar(theta = "y", start=0)
```



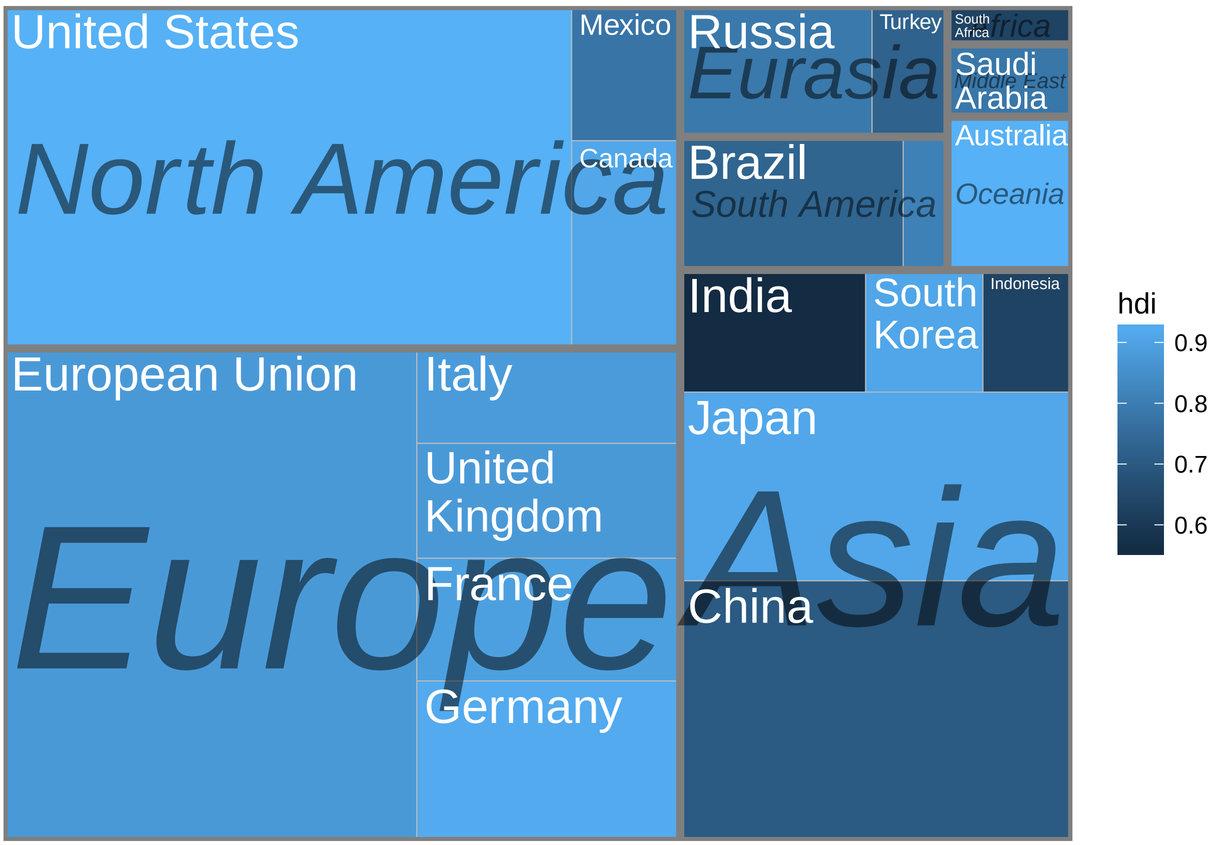
5.3 Treemap

```
library(ggplot2)
library(treemapify)
ggplot(G20, aes(area = gdp_mil_usd, fill = hdi, label = country)) +
  geom_treemap() +
  geom_treemap_text(fontface = "italic", colour = "white", place = "centre",
                    grow = TRUE)
```



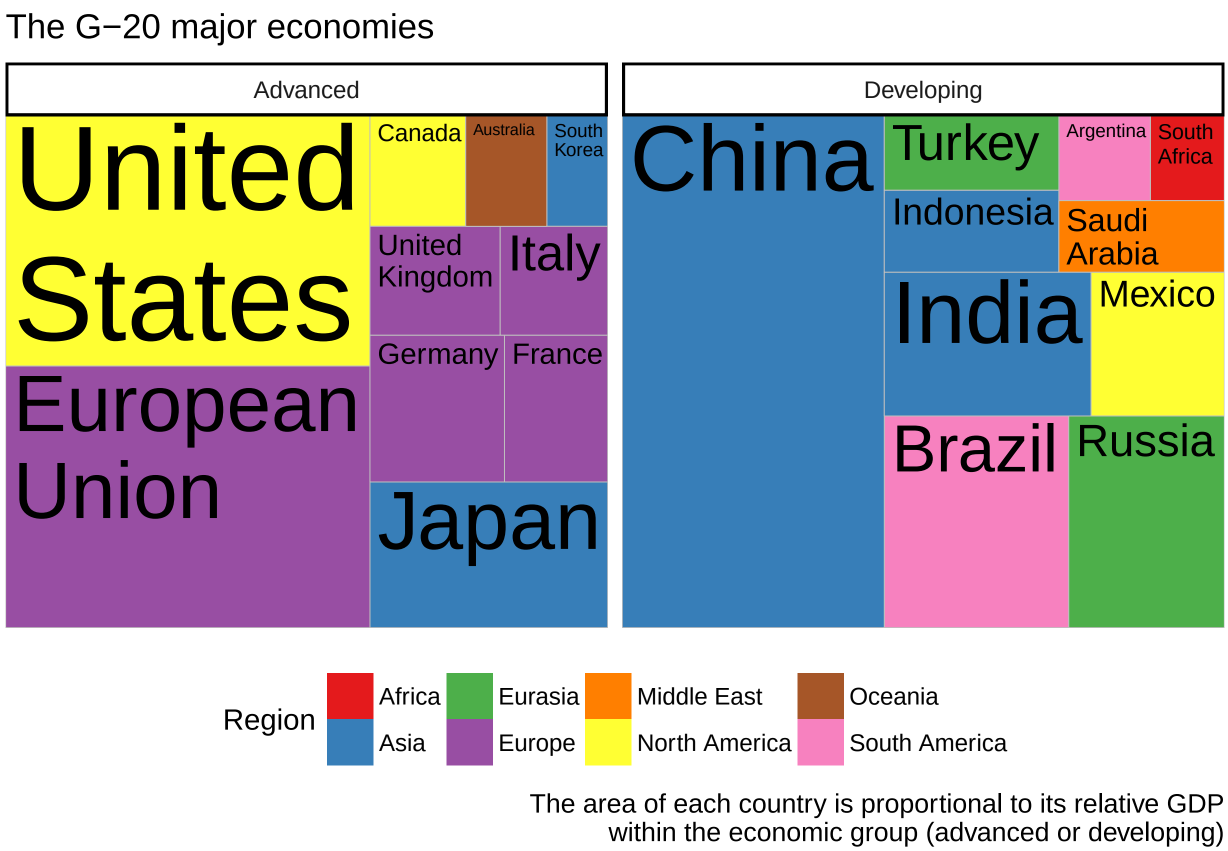
5.3 Treemap (second version)

```
ggplot(G20, aes(area = gdp_mil_usd, fill = hdi, label = country,
                subgroup = region)) +
  geom_treemap() +
  geom_treemap_subgroup_border() +
  geom_treemap_subgroup_text(place = "centre", grow = T, alpha = 0.5, colour =
                             "black", fontface = "italic", min.size = 0) +
  geom_treemap_text(colour = "white", place = "topleft", reflow = T)
```



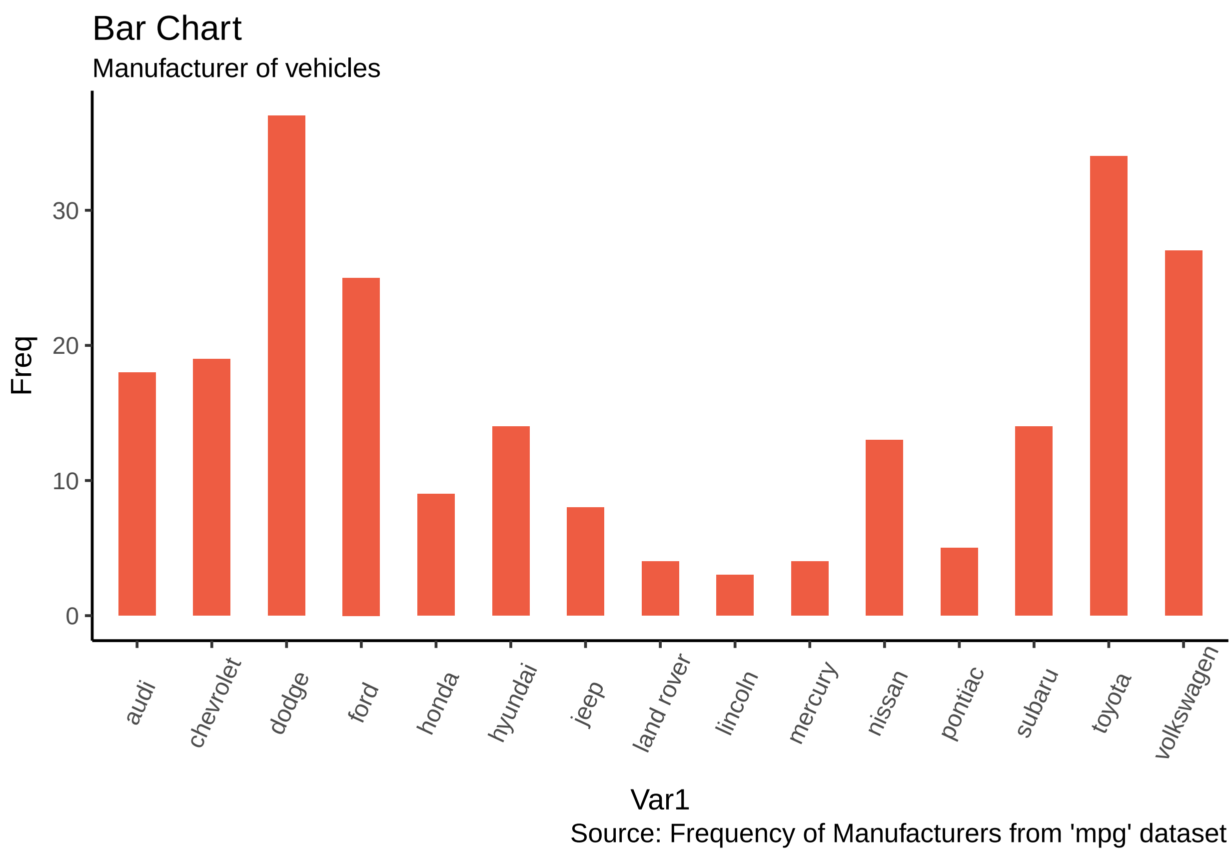
5.3 Treemap (third version)

```r
ggplot(G20, aes(area = gdp_mil_usd, fill = region, label = country)) +
  geom_treemap() +
  geom_treemap_text(grow = T, reflow = T, colour = "black") +
  facet_wrap( ~ econ_classification) +
  scale_fill_brewer(palette = "Set1") +
  theme(legend.position = "bottom") +
  labs(
    title = "The G-20 major economies",
    caption = "The area of each country is proportional to its relative GDP
               within the economic group (advanced or developing)",
    fill = "Region"
  )
```



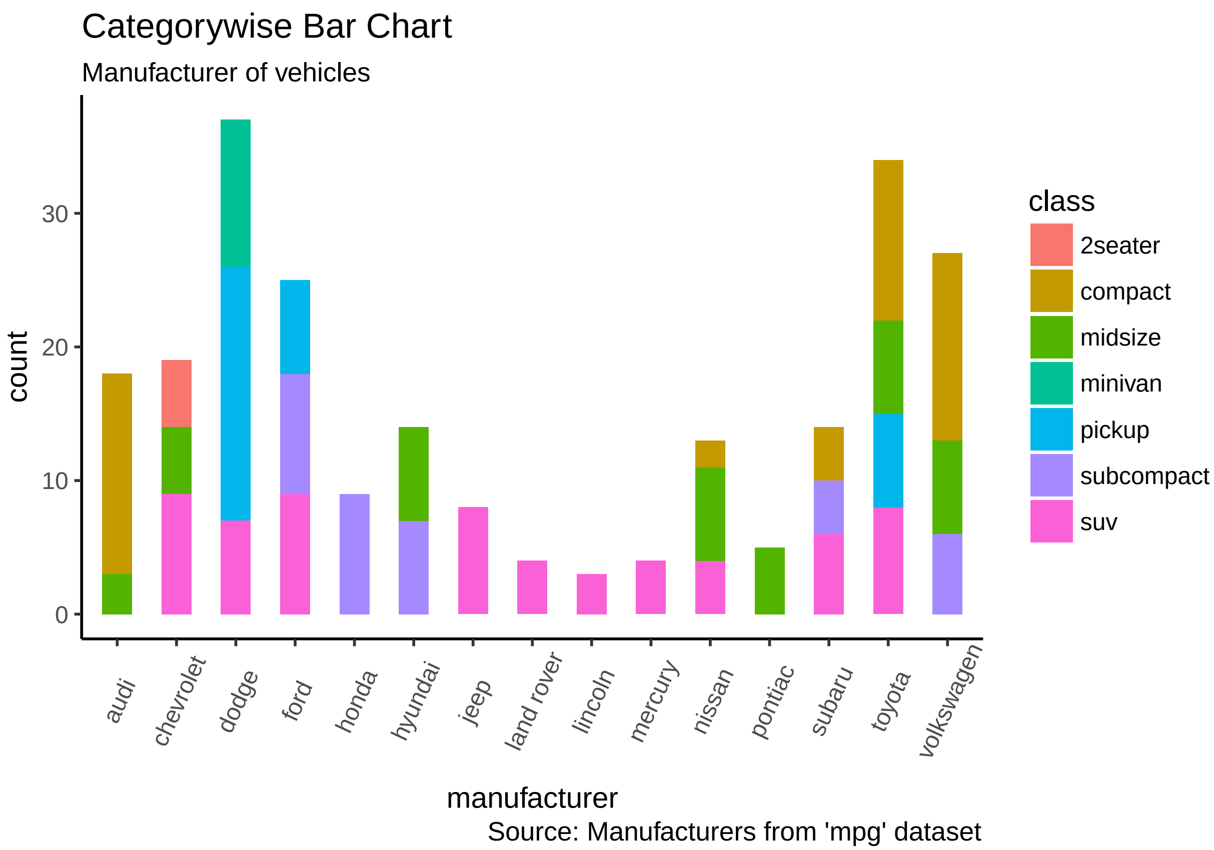
5.4 Bar Chart

```
freqtable <- table(mpg$manufacturer)
df <- as.data.frame.table(freqtable)
library(ggplot2)
theme_set(theme_classic())

# Plot
g <- ggplot(df, aes(Var1, Freq))
g + geom_bar(stat="identity", width = 0.5, fill="tomato2") +
      labs(title="Bar Chart",
           subtitle="Manufacturer of vehicles",
           caption="Source: Frequency of Manufacturers from 'mpg' dataset") +
      theme(axis.text.x = element_text(angle=65, vjust=0.6))
```



5.4 Bar Chart (2nd version)

```
g <- ggplot(mpg, aes(manufacturer))
g + geom_bar(aes(fill=class), width = 0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.6)) +
  labs(title="Categorywise Bar Chart",
       subtitle="Manufacturer of vehicles",
       caption="Source: Manufacturers from 'mpg' dataset")
```

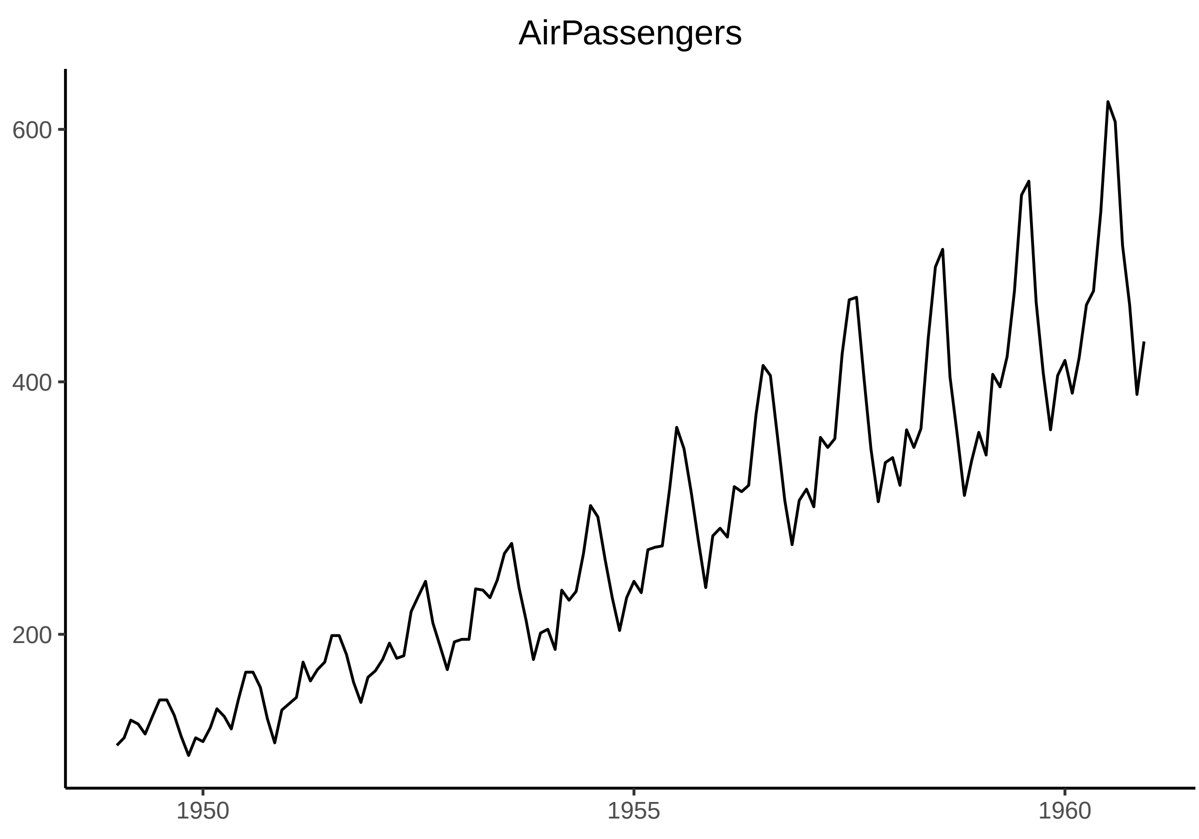


6. Change

Used to compare the position or performance of multiple items with respect to each other. Actual values matters somewhat less than the ranking.

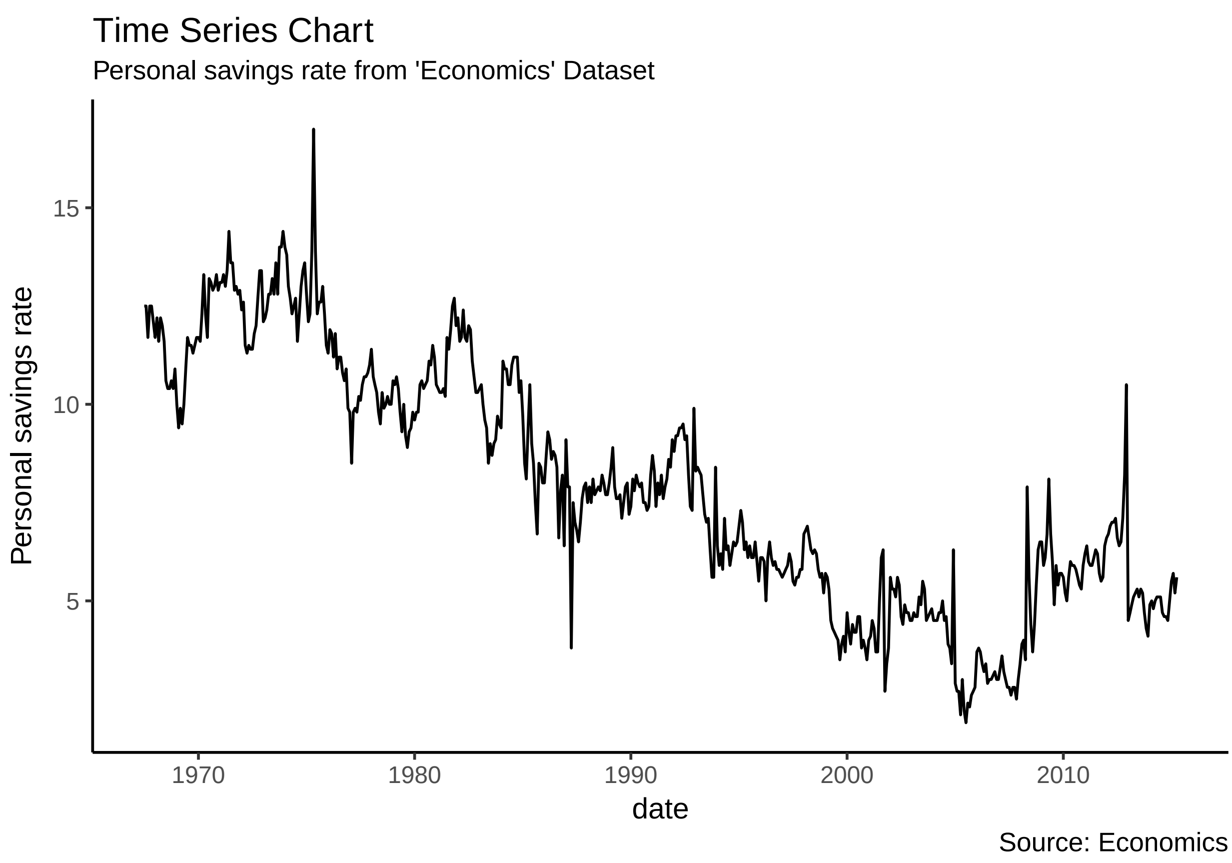
6.1 Time Series Plot (from a time series object)

```
# Prepare data: group mean city mileage by manufacturer.
## From Timeseries object (ts)
library(ggplot2)
library(ggfortify)
theme_set(theme_classic())

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



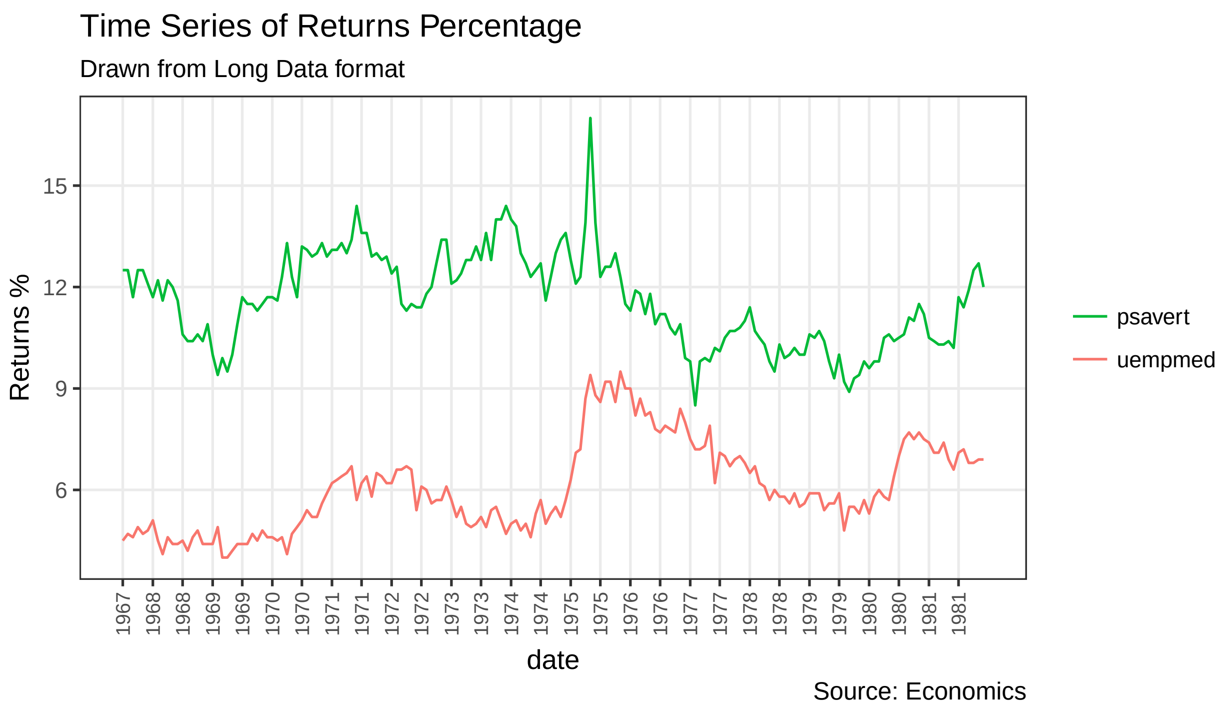
6.1 Time Series Plot (from a data frame)

```
library(ggplot2)
library(ggfortify)

theme_set(theme_classic())

# Allow Default X Axis Labels
ggplot(economics, aes(x=date)) +
  geom_line(aes(y=psavert)) +
  labs(title="Time Series Chart",
       subtitle="Personal savings rate from 'Economics' Dataset",
       caption="Source: Economics",
       y="Personal savings rate")
```



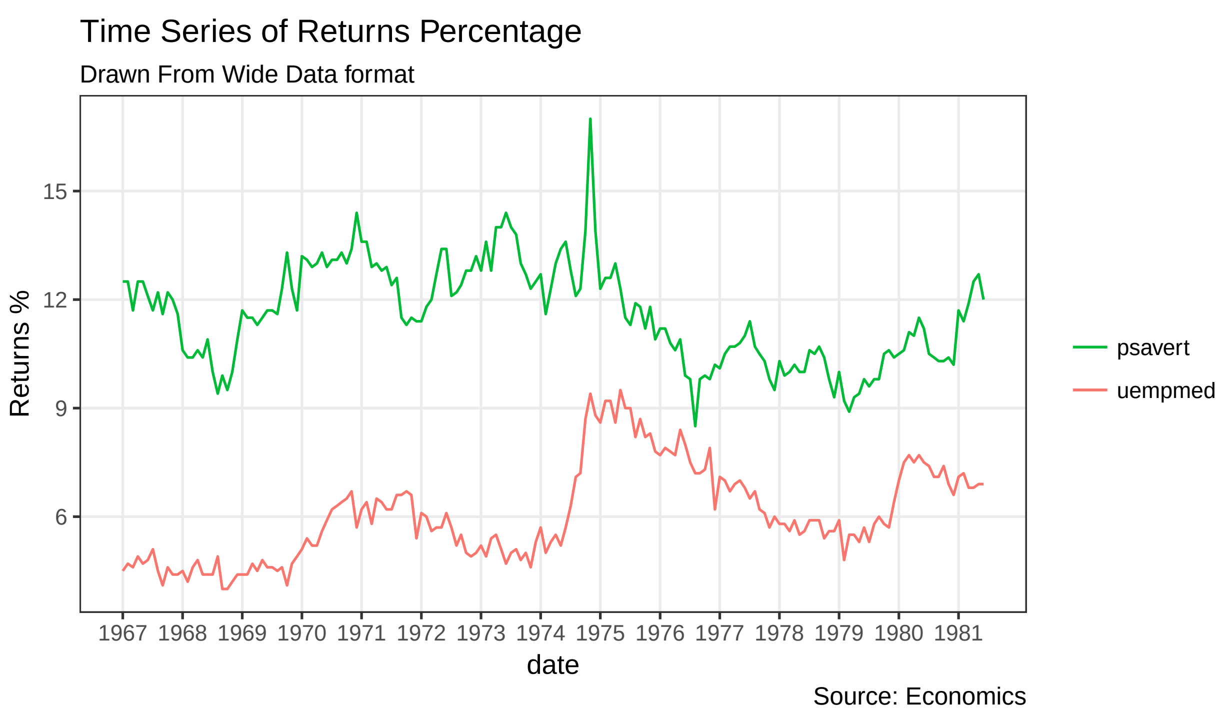
6.1 Time Series Plot (from *long* data format)

```
data(economics_long, package = "ggplot2")
library(ggplot2)
library(lubridate)
theme_set(theme_bw())

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 colour
  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
```



6.1 Time Series Plot (from *wide* data format)

```
library(ggplot2)
library(lubridate)
theme_set(theme_bw())

df <- economics[, c("date", "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=psavert, col="psavert")) +
  geom_line(aes(y=uempmed, col="uempmed")) +
  labs(title="Time Series of Returns Percentage",
       subtitle="Drawn From Wide Data format",
       caption="Source: Economics", y="Returns %") + # title and caption
  scale_x_date(labels = lbls, breaks = brks) + # change to monthly ticks and labels
  scale_color_manual(name="",
                     values = c("psavert"="#00ba38", "uempmed"="#f8766d")) + # line colour
  theme(panel.grid.minor = element_blank()) # turn off minor grid
```



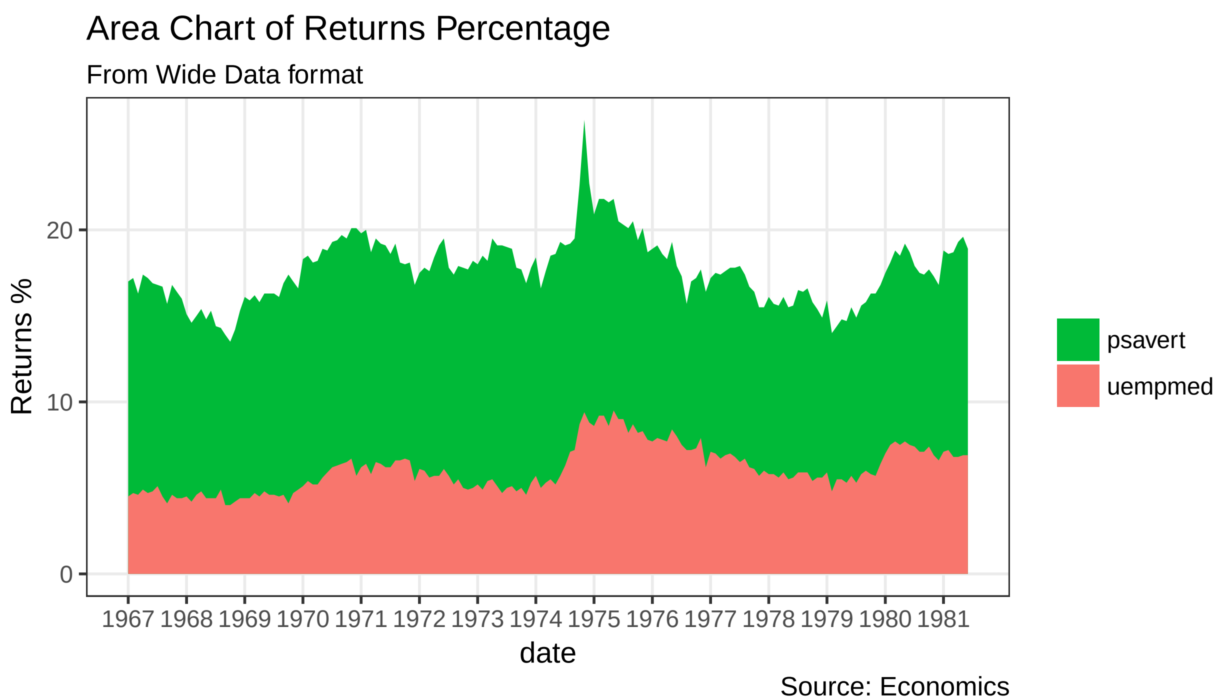
6.2 Stacked Area Chart

Stacked area chart is just like a line chart, except that the region below the plot is all colored. This is typically used when:

- You want to describe how a quantity or volume (rather than something like price) changed over time.
- You have many data points. For very few data points, consider plotting a bar chart.
- You want to show the contribution from individual components.

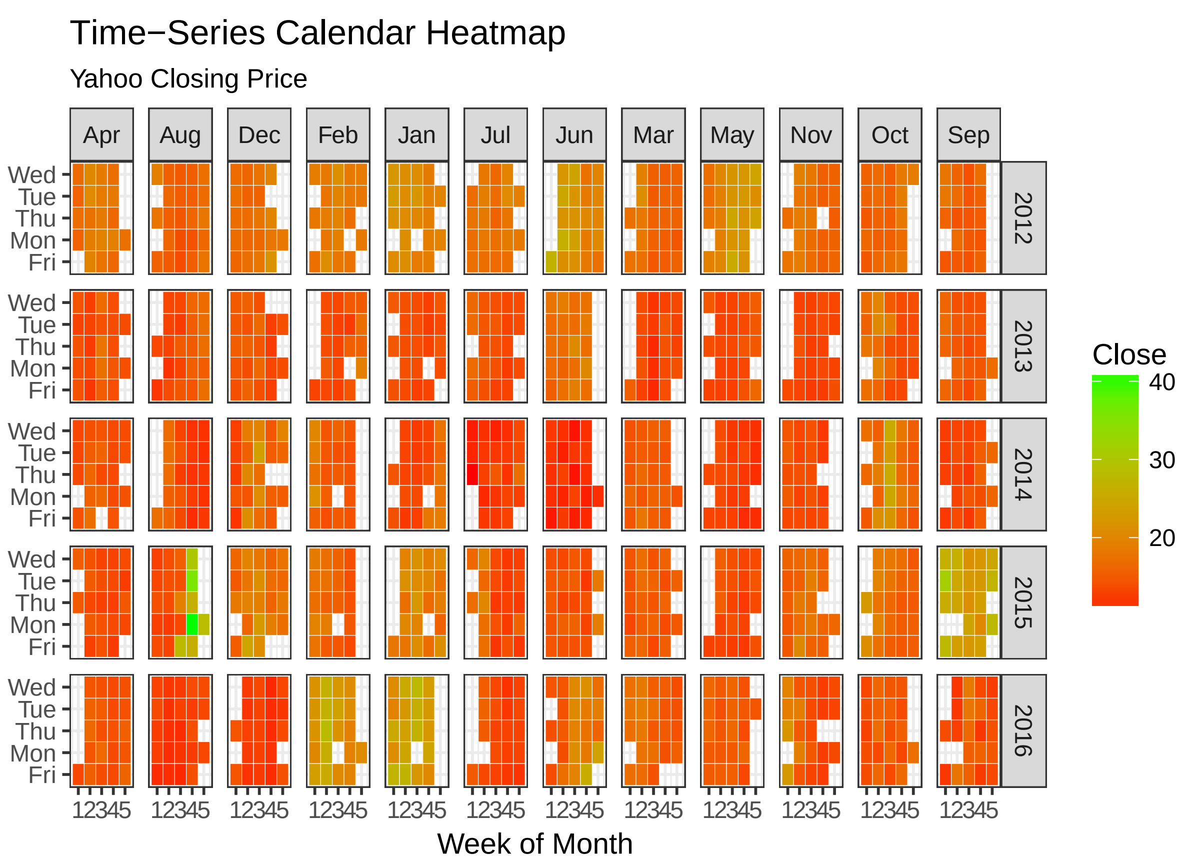
Stacked Area Chart

```
library(ggplot2)
library(lubridate)
theme_set(theme_bw())
df <- economics[, c("date", "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_area(aes(y=psavert+uempmed, fill="psavert")) + geom_area(aes(y=uempmed, fill="uempmed")) +
  labs(title="Area Chart of Returns Percentage",subtitle="From Wide Data format",
       caption="Source: Economics",y="Returns %") +  # title and caption
  scale_x_date(labels = lbls, breaks = brks) + # change to monthly ticks and labels
  scale_fill_manual(name="",
                    values = c("psavert"="#00ba38", "uempmed"="#f8766d")) +  # line colour
  theme(panel.grid.minor = element_blank()) # turn off minor grid
```



6.3 Calendar Heat 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
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",
       subtitle="Yahoo Closing Price", fill="Close")
```

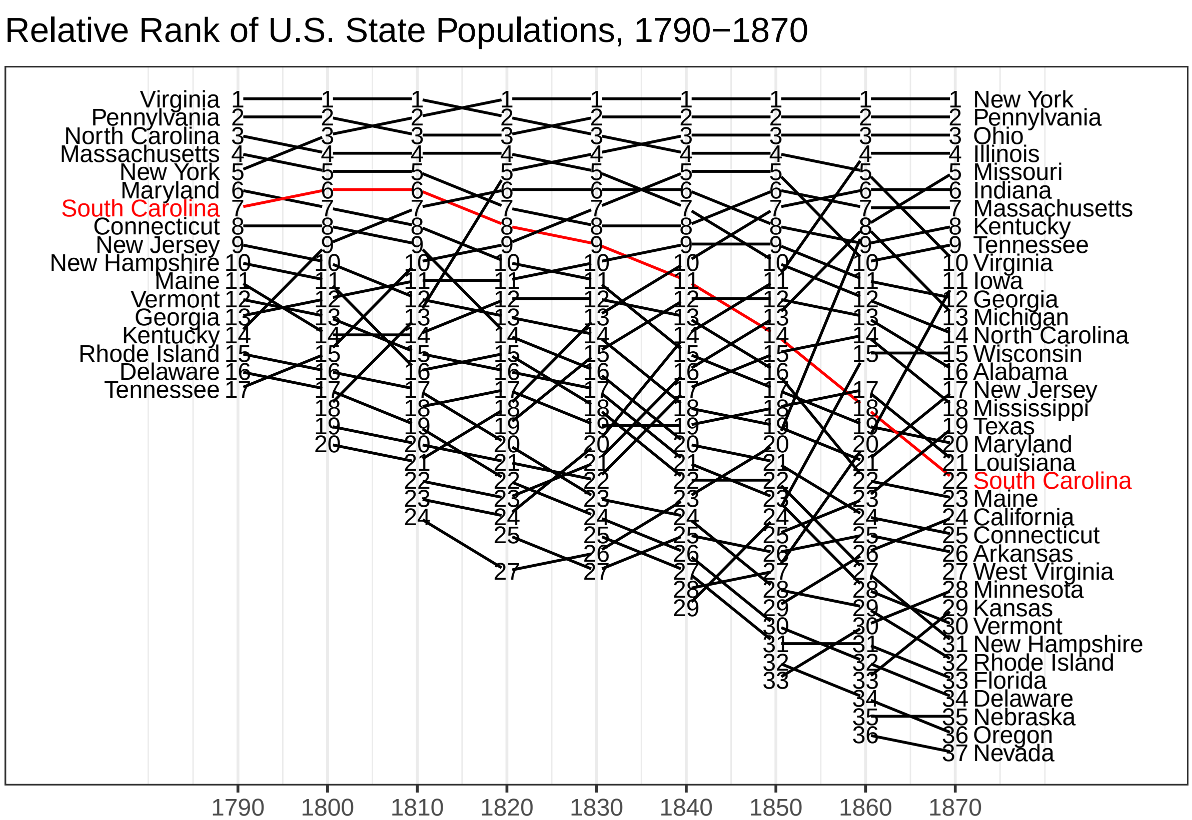


6.4 Slope Chart

Slope charts are a great tool if you want to visualise change in value and ranking between categories. This is more suitable over a time series when there are very few time points.

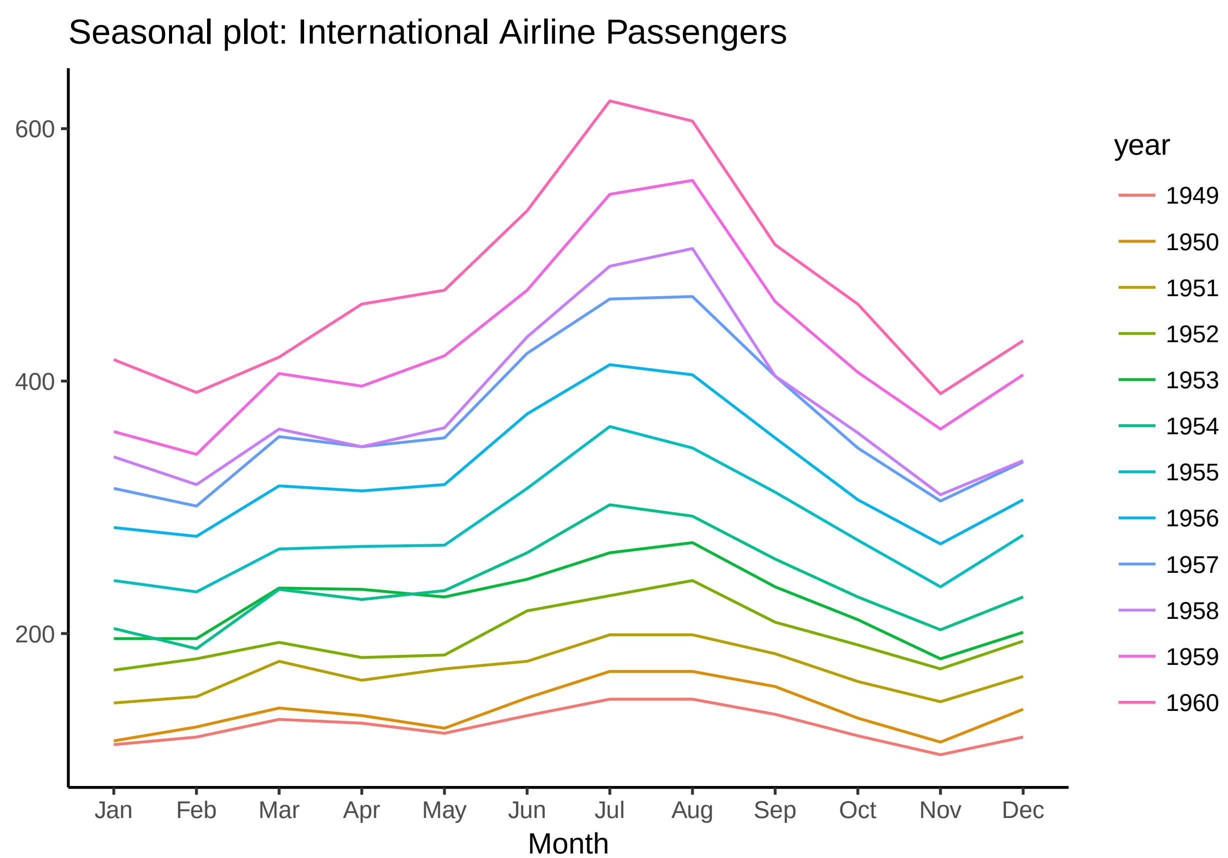
Slope Chart

```r
library(slopegraph)
library(ggplot2)
data(states)
cols <- `[<-`(rep("black", 37), 7, "red")
ggslopegraph(states, offset.x = 0.06, yrev = TRUE,
  col.lines = cols, col.lab = cols,
  main = 'Relative Rank of U.S. State Populations, 1790-1870') +
 theme_bw()
```



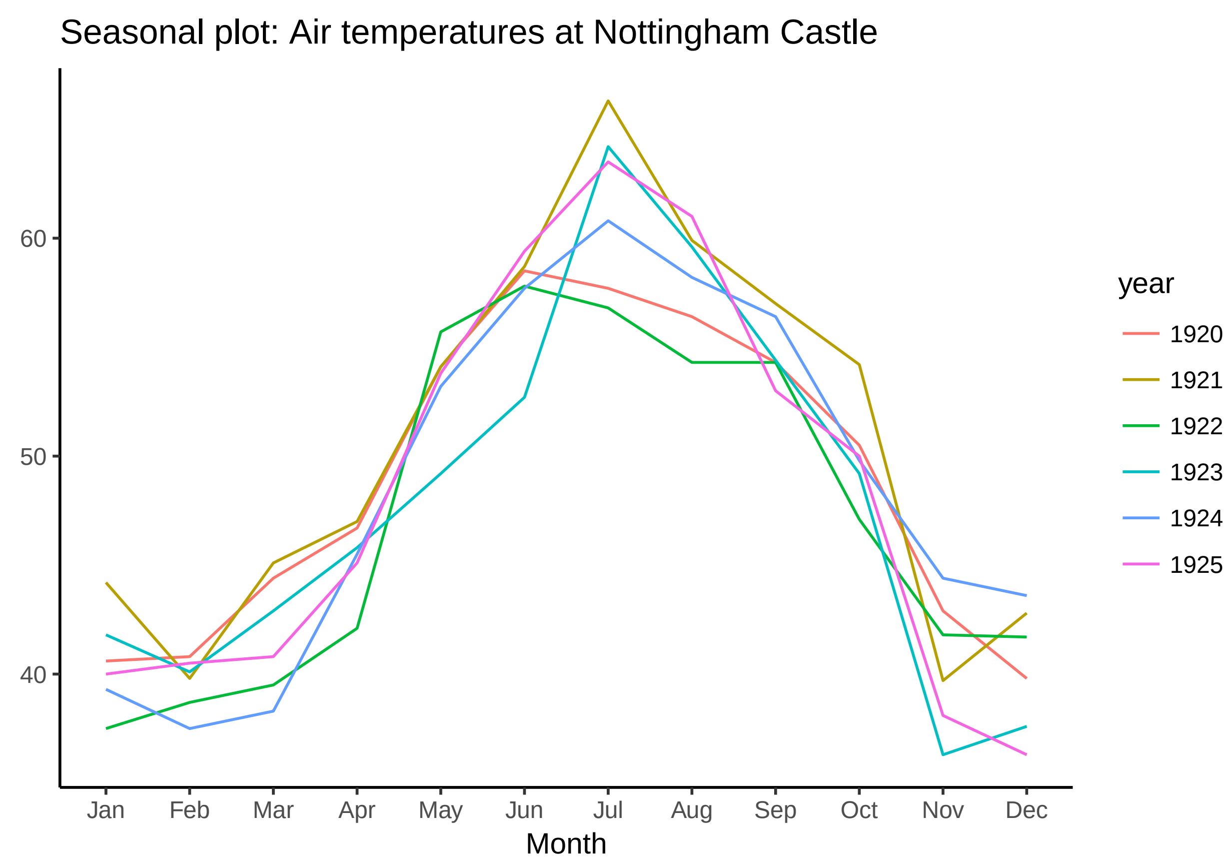
6.5 Seasonal Plot

```
library(ggplot2)
library(forecast)
theme_set(theme_classic())

# Subset data
nottem_small <- window(nottem, start=c(1920, 1), end=c(1925, 12))

# Plot
ggseasonplot(AirPassengers) +
  labs(title="Seasonal plot: International Airline Passengers")
```



6.5 Seasonal Plot

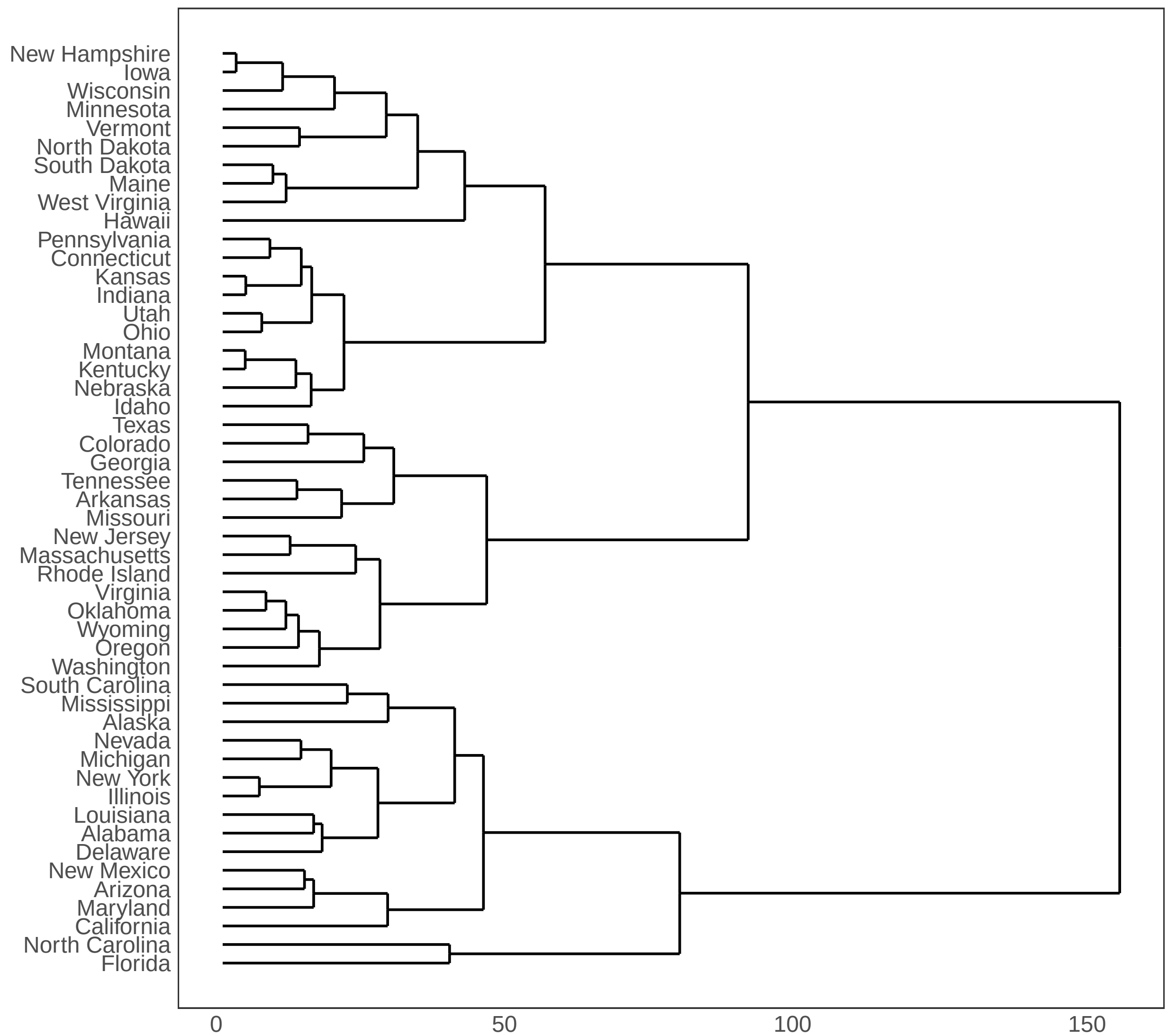
```
library(ggplot2)
library(forecast)
theme_set(theme_classic())

# Subset data
nottem_small <- window(nottem, start=c(1920, 1), end=c(1925, 12))

# Plot
ggseasonplot(nottem_small) +
  labs(title="Seasonal plot: Air temperatures at Nottingham Castle")
```



7. Groups

7.1 Hierarchical Dendrogram

```
library(ggplot2)
library(ggdendro)
theme_set(theme_bw())

hc <- hclust(dist(USArrests), "ave") # hierarchical clustering

# plot
ggdendrogram(hc, rotate = TRUE, size = 2)
```



7.2 Clusters

```
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 axes 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)) +
  geom_encircle(data = df_pc_ver, aes(x=PC1, y=PC2))
```