Beauty of ggplot2

Jihui Lee

Contents

1 Goal: No more basic plots!

1.1 plot vs ggplot

- plot(x = , y = , type = , col, xlab = , ylab = , main =)
- ggplot(data = , aes(x = , y = , col =)) + "type"
  - geom_point()
  - geom_boxplot()
  - geom_line()

1.2 Install & load the package "ggplot2"

```
#install.packages("ggplot2")
library(ggplot2)
```

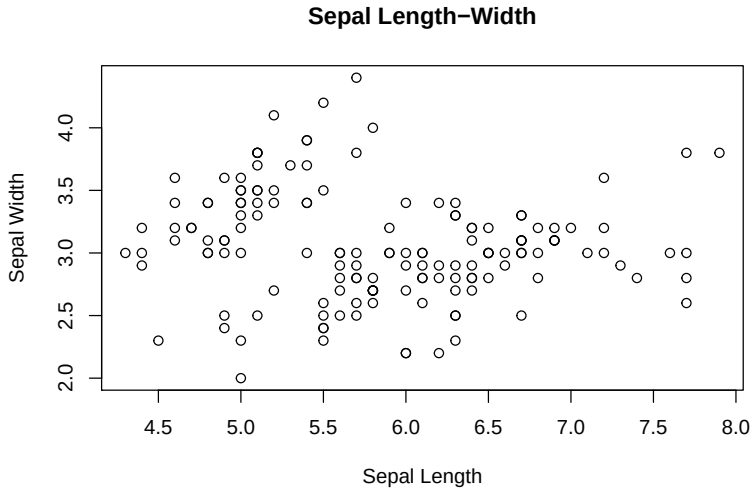
2 Frequently Used Plots

2.1 Scatter plot

```
head(iris)
```

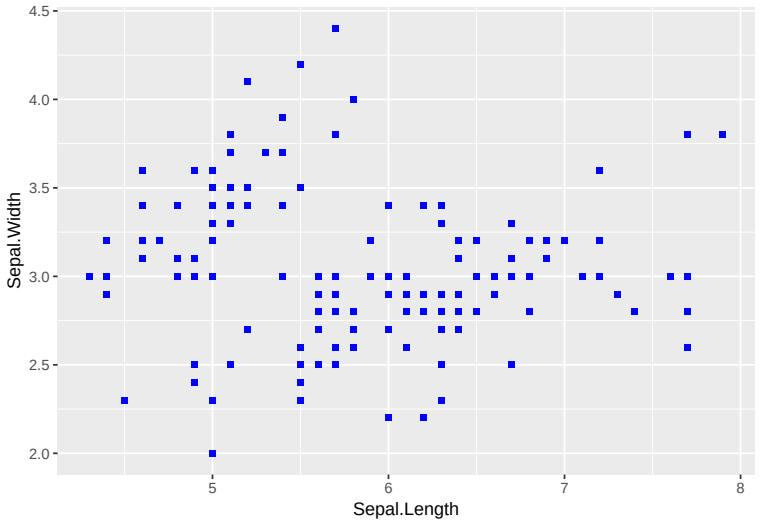
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
plot(x = iris$Sepal.Length, y = iris$Sepal.Width,
     xlab = "Sepal Length", ylab = "Sepal Width", main = "Sepal Length-Width")
```

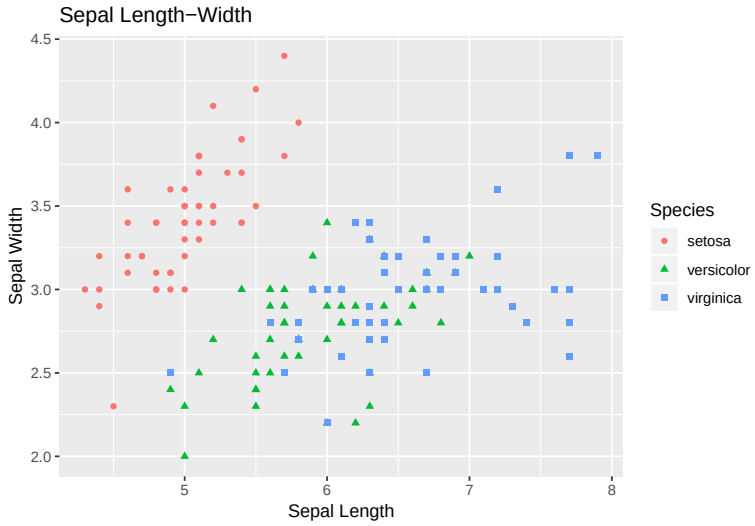



```
#qplot(x = Sepal.Length, y = Sepal.Width, data = iris,
#      xlab="Sepal Length", ylab="Sepal Width",
```

```
#      main="Sepal Length-Width", color=Species, shape=Species)
scatter = ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width))
# One color/shape
scatter + geom_point(color = "blue", shape = 15)
```




```
# Different color/shape for Species
scatter + geom_point(aes(color = Species, shape = Species)) +
  xlab("Sepal Length") +  ylab("Sepal Width") + ggtitle("Sepal Length-Width")
```

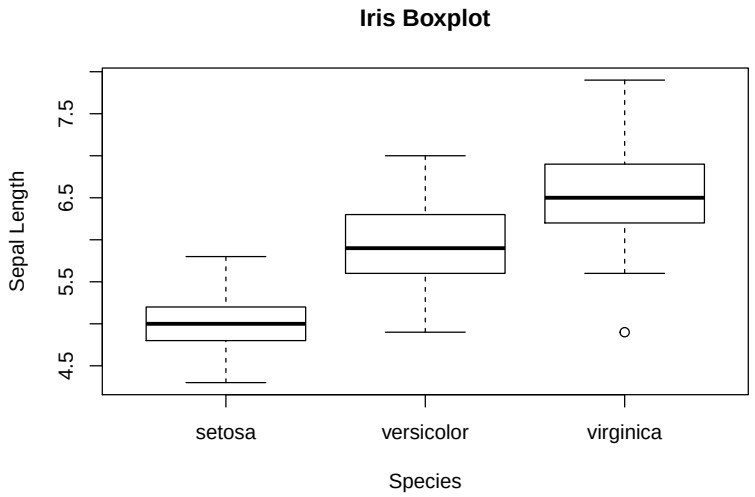



```
scatter + geom_point(aes(color = Petal.Width, shape = Species), size = 2, alpha = I(1/2)) +
          geom_vline(aes(xintercept = mean(Sepal.Length)), color = "red", linetype = "dashed") +
          geom_hline(aes(yintercept = mean(Sepal.Width)), color = "red", linetype = "dashed") +
          scale_color_gradient(low = "yellow", high = "red") +
```

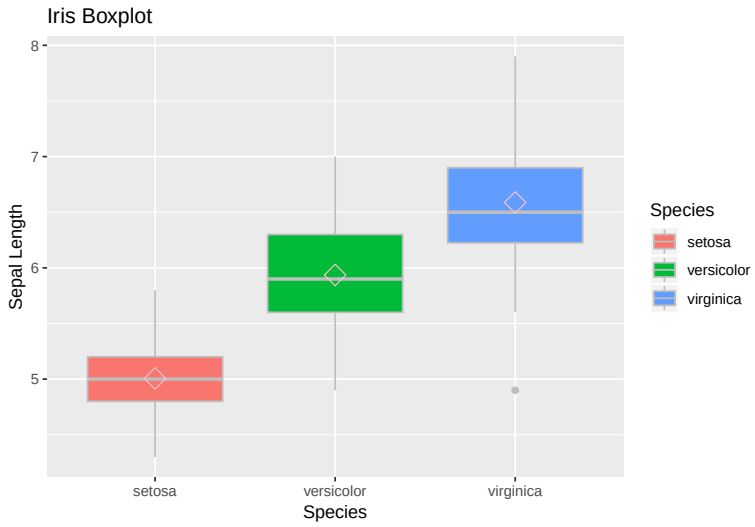
```
        xlab("Sepal Length") +  ylab("Sepal Width") + ggtitle("Sepal Length-Width")
```

2.2 Box plot

```
boxplot(Sepal.Length ~ Species, data = iris,
        xlab = "Species", ylab = "Sepal Length", main = "Iris Boxplot")
```




```
box = ggplot(data = iris, aes(x = Species, y = Sepal.Length))
box + geom_boxplot(aes(fill = Species), col = "grey") +
  ylab("Sepal Length") + ggtitle("Iris Boxplot") +
  stat_summary(fun.y = mean, geom = "point", shape = 5, size = 4, color = "pink")
```

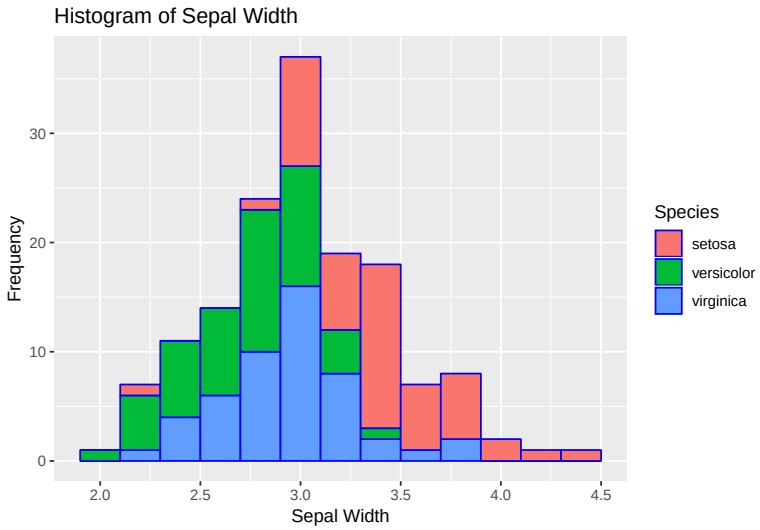



```
# Remove the legend : guides(fill=FALSE)

# Flipped axes : coord_flip()
```

2.3 Histogram

```
hist(iris$Sepal.Width, freq = NULL, density = NULL, breaks = 12,
     xlab = "Sepal Width", ylab = "Frequency", main = "Histogram of Sepal Width")
```


```
histogram = ggplot(data = iris, aes(x = Sepal.Width))

histogram + geom_histogram(binwidth = 0.2, color = "blue", aes(fill = Species)) +

  xlab("Sepal Width") +  ylab("Frequency") + ggtitle("Histogram of Sepal Width")
```

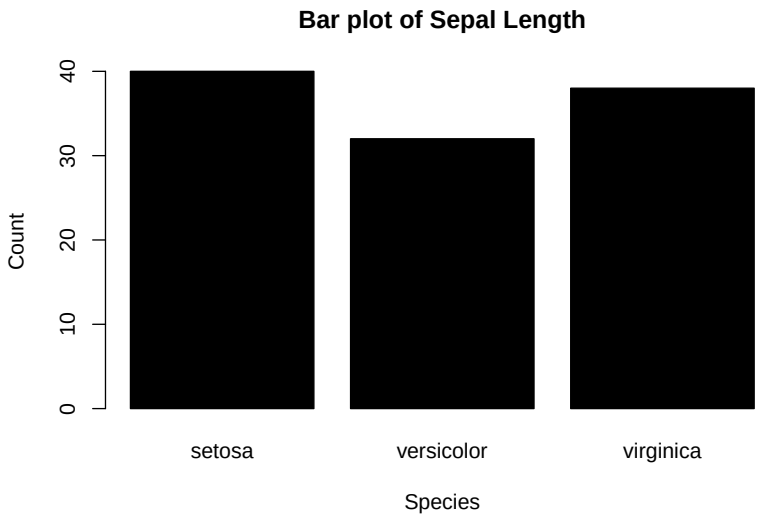


2.4 Bar chart

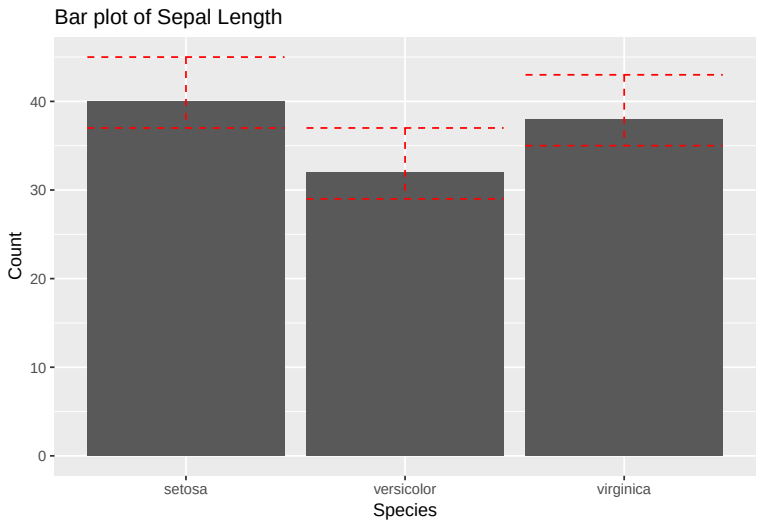
```
# Bar Chart 1

set.seed(1234)

iris1 = iris[sample(1:nrow(iris), 110), ]

hline = data.frame(Species = c("setosa", "versicolor", "virginica"),
                   hline1 = as.vector(table(iris1$Species) - 3),
                   hline2 = as.vector(table(iris1$Species) + 5))

hline
```

```
##      Species hline1 hline2
## 1     setosa     37     45
## 2 versicolor     29     37
## 3  virginica     35     43
```

```
barplot(table(iris1$Species), col = "black",
        xlab = "Species", ylab = "Count", main = "Bar plot of Sepal Length")
```




```
bar = ggplot(data = iris1, aes(x = Species))
bar + geom_bar() + xlab("Species") +  ylab("Count") +
  ggtitle("Bar plot of Sepal Length") +
  geom_errorbar(data = hline, aes(ymin = hline1, ymax = hline2),
                col = "red", linetype = "dashed")
```

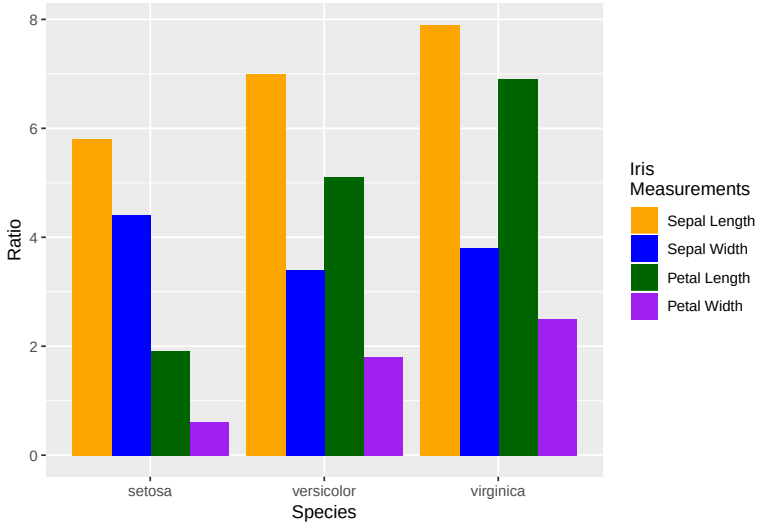



```
# Bar chart 2
library(reshape2)
iris2 = melt(iris, id.vars = "Species")
iris2[1:3,]
```

```
##   Species     variable value
## 1  setosa Sepal.Length   5.1
```

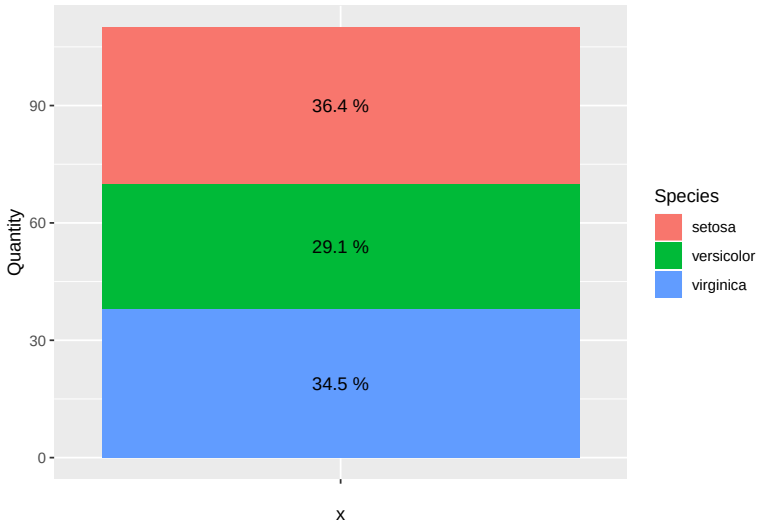
```
## 2  setosa Sepal.Length   4.9
## 3  setosa Sepal.Length   4.7
```

```
bar1 = ggplot(data = iris2, aes(x = Species, y = value, fill = variable))
bar1 + geom_bar(stat = "identity", position = "dodge") + ylab("Ratio") +
  scale_fill_manual(values = c("orange", "blue", "darkgreen", "purple"),
                    name = "Iris\nMeasurements",
                    breaks = c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width"),
                    labels = c("Sepal Length", "Sepal Width", "Petal Length", "Petal Width"))
```

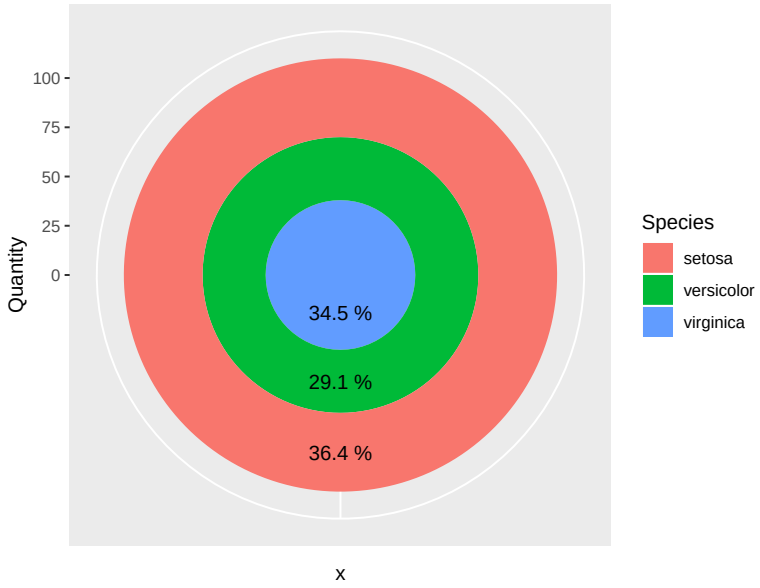


2.5 Pie chart

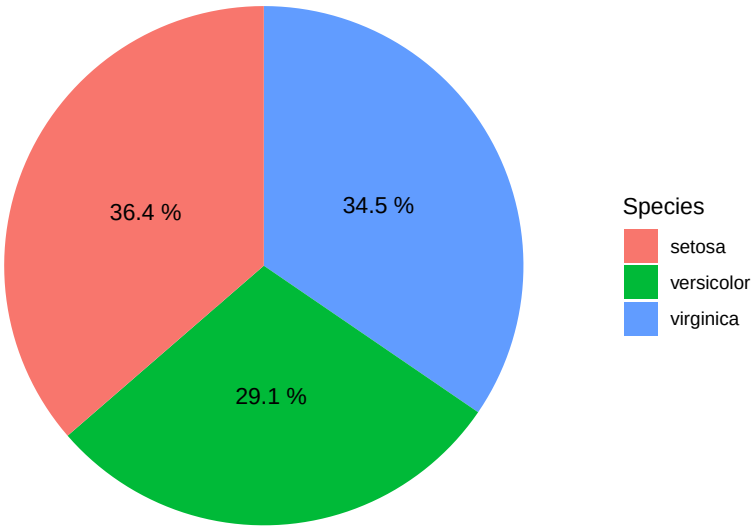
```
# Frequency table
quan = as.vector(table(iris1$Species))
prop = round(prop.table(quan), 3)
quantity = data.frame(Species = c("setosa", "versicolor", "virginica"),
                      Quantity = quan, Proportion = prop)
# Create a basic bar
pie = ggplot(quantity, aes(x = "", y = Quantity, fill = Species)) +
  geom_bar(stat = "identity", width = 1) +
  geom_text(aes(label = paste(Proportion*100, "%")), position = position_stack(vjust = 0.5))
pie
```




```
pie + coord_polar()
```




```
pie + coord_polar("y", start = 0) +
  labs(x = NULL, y = NULL, fill = "Species", title = NULL) +
  theme_classic() + theme(axis.line = element_blank(),
          axis.text = element_blank(), axis.ticks = element_blank())
```



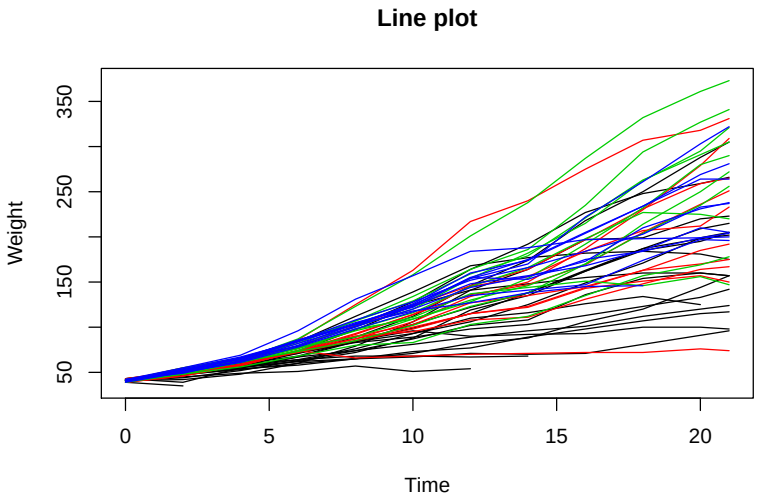
2.6 Line plot

```
# Line Plot 1
head(ChickWeight)
```

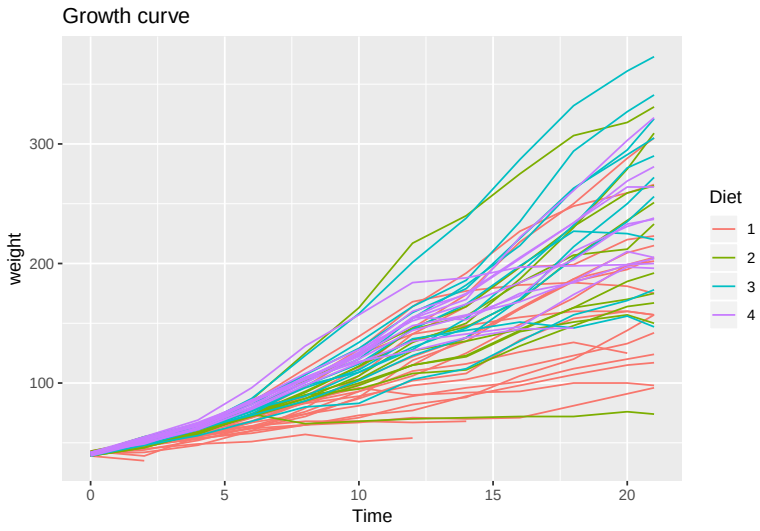
```
##   weight Time Chick Diet
## 1     42    0     1    1
## 2     51    2     1    1
## 3     59    4     1    1
## 4     64    6     1    1
## 5     76    8     1    1
## 6     93   10     1    1
```

```
chick = unique(ChickWeight$Chick)
dat = ChickWeight[ChickWeight$Chick == chick[1],]
color = as.vector(dat$Diet[1])
plot(dat$Time, dat$weight, type = "l", ylim = range(ChickWeight$weight), col = color,
     xlab = "Time", ylab = "Weight", main = "Line plot")


for (i in 2:length(chick))
{
  dat = ChickWeight[ChickWeight$Chick == chick[i],]
```

```
  color = as.vector(dat$Diet[1])

  lines(dat$Time, dat$weight, col = color)
}
```




```
ggplot(data = ChickWeight, aes(x = Time, y = weight)) +

  geom_line(aes(color = Diet, group = Chick)) + ggtitle("Growth curve")
```




```
# Line Plot 2

library(plyr)

sepal.min = ddply(iris, "Species", summarise,

                  xval = min(Sepal.Length), yval = min(Sepal.Width))

sepal.max = ddply(iris, "Species", summarise,

                  xval = max(Sepal.Length), yval = max(Sepal.Width))

sepal = rbind(sepal.min, sepal.max)
```

```
sepal
```

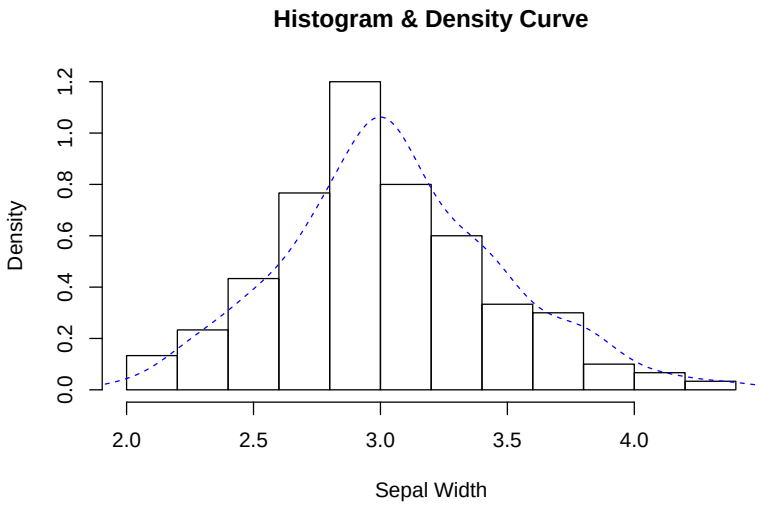
```
##      Species xval yval
## 1     setosa  4.3  2.3
## 2 versicolor  4.9  2.0
## 3  virginica  4.9  2.2
## 4     setosa  5.8  4.4
## 5 versicolor  7.0  3.4
## 6  virginica  7.9  3.8
```

```
ggplot(sepal, aes(x = xval, y = yval, color = Species)) +
  geom_line(aes(linetype = Species), size = 1.2) +
  geom_point(aes(shape = Species), size = 4) +
  scale_shape_manual(values = c(6, 5, 4)) +
  scale_linetype_manual(values = c("dotdash", "solid", "dotted")) +
  xlab("Sepal Length") + ylab("Sepal Width") + ggtitle("Line plot of sepal length & width")
```

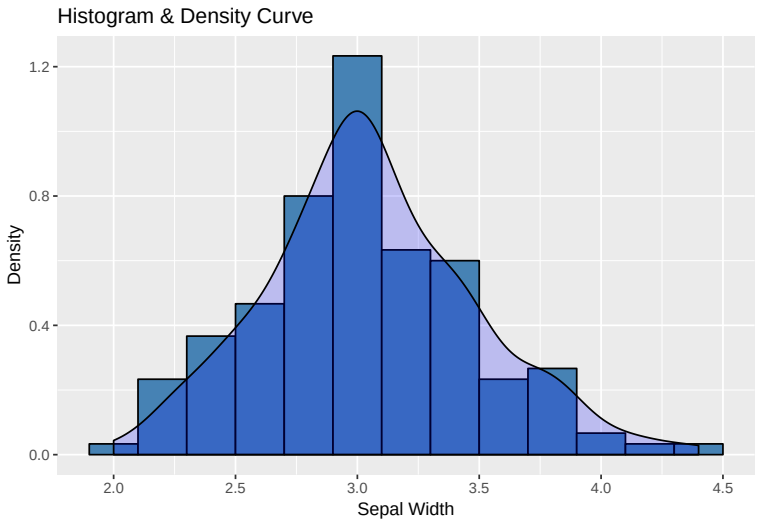
2.7 Density curve

```
# Density Curve 1
d = density(iris$Sepal.Width)
hist(iris$Sepal.Width, breaks = 12, prob = TRUE,
     xlab = "Sepal Width", main = "Histogram & Density Curve")
```

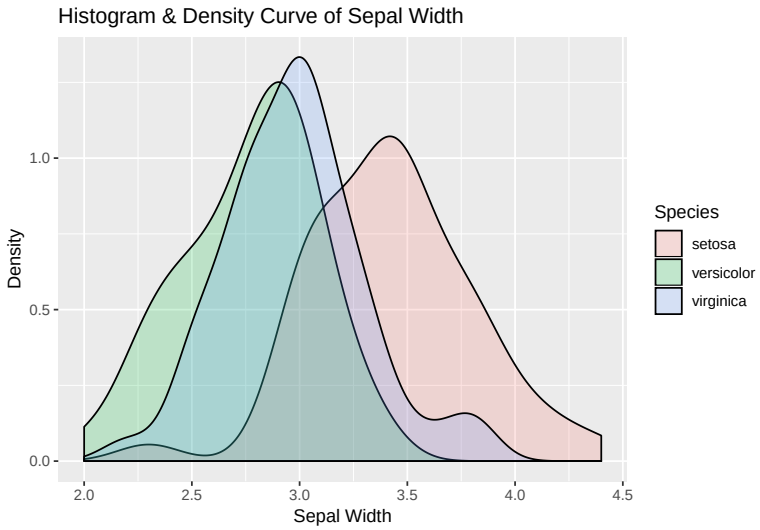
```
lines(d, lty = 2, col = "blue")
```




```
#polygon(d, col = "yellow", border = "blue")


density = ggplot(data = iris, aes(x = Sepal.Width))
density + geom_histogram(binwidth = 0.2, color = "black",
                         fill = "steelblue", aes(y = ..density..)) +
  geom_density(stat = "density", alpha = I(0.2), fill = "blue") +
  xlab("Sepal Width") +  ylab("Density") + ggtitle("Histogram & Density Curve")
```



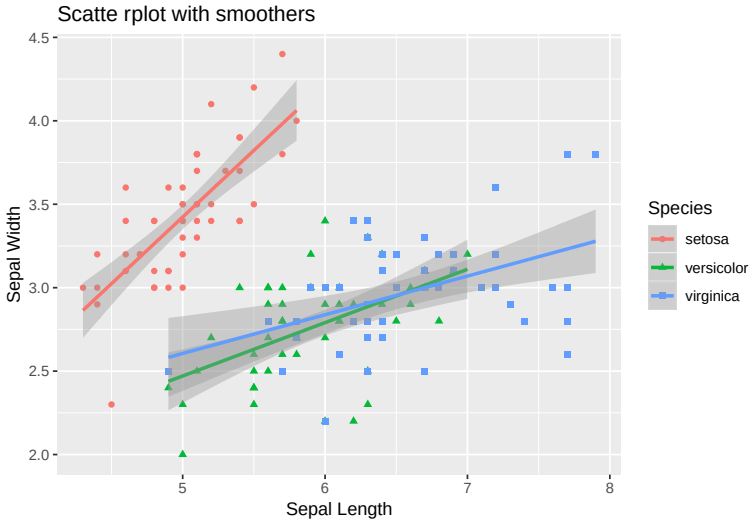

```
# Density Curve 2
density2 = ggplot(data = iris, aes(x = Sepal.Width, fill = Species))
density2 + geom_density(stat = "density", alpha = I(0.2)) +
  xlab("Sepal Width") + ylab("Density") + ggtitle("Histogram & Density Curve of Sepal Width")
```



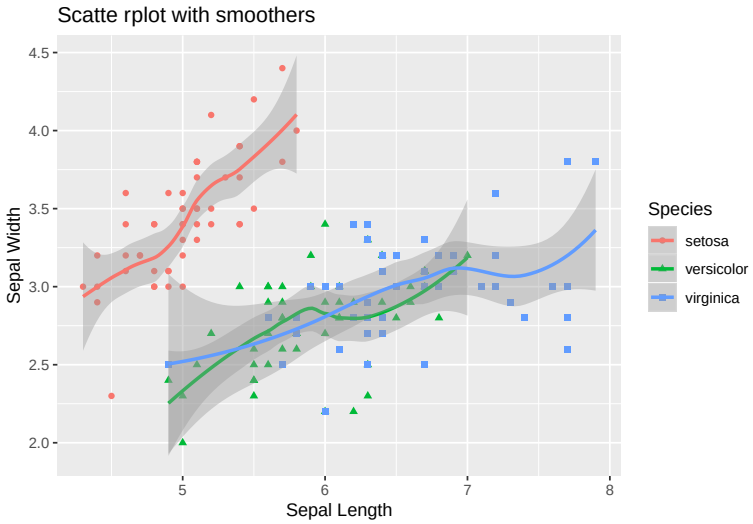
3 Elaboration

3.1 Adding smoothers

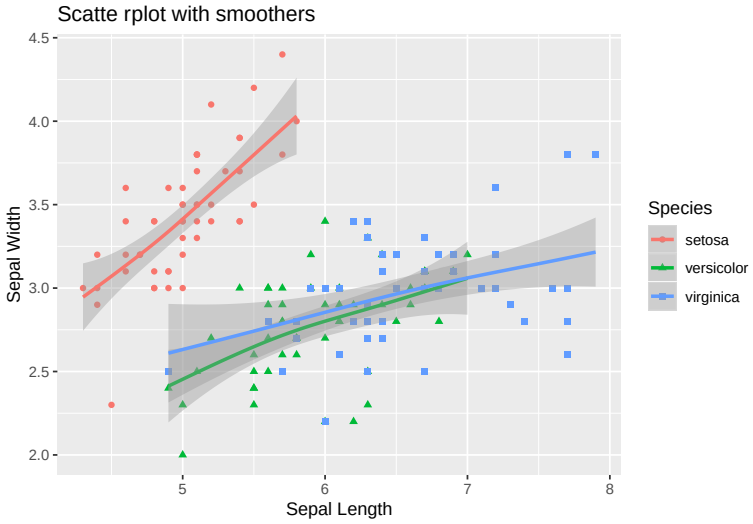
```
smooth = ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  geom_point(aes(shape = Species), size = 1.5) +
  xlab("Sepal Length") + ylab("Sepal Width") + ggtitle("Scatte rplot with smoothers")
# Linear model
smooth + geom_smooth(method = "lm")
```



```r
# Local polynomial regression

smooth + geom_smooth(method = "loess")
```



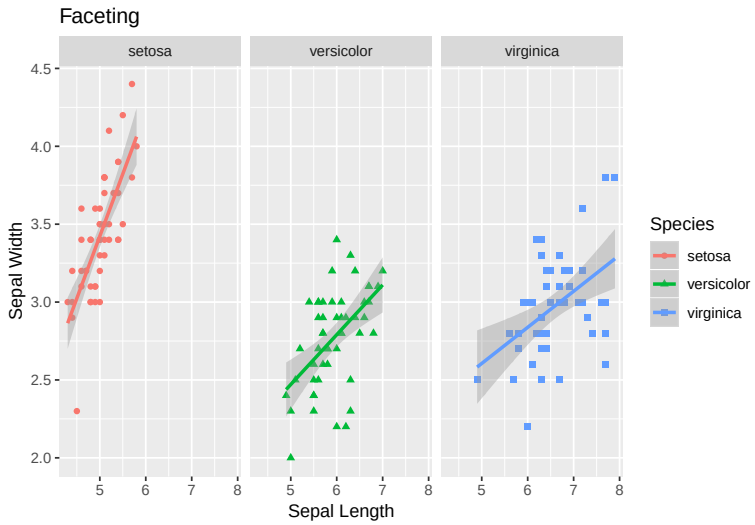

```r
# Generalised additive model

smooth + geom_smooth(method = "gam", formula = y ~ s(x, bs = "cs"))
```



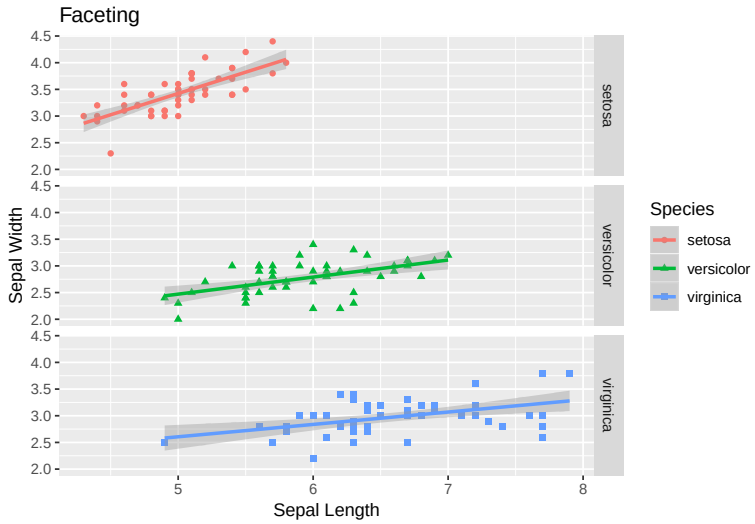
3.2 Faceting

```r
facet = ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species)) +
  geom_point(aes(shape = Species), size = 1.5) + geom_smooth(method = "lm") +
  xlab("Sepal Length") + ylab("Sepal Width") + ggtitle("Faceting")

# Along rows
```

```
facet + facet_grid(. ~ Species)
```

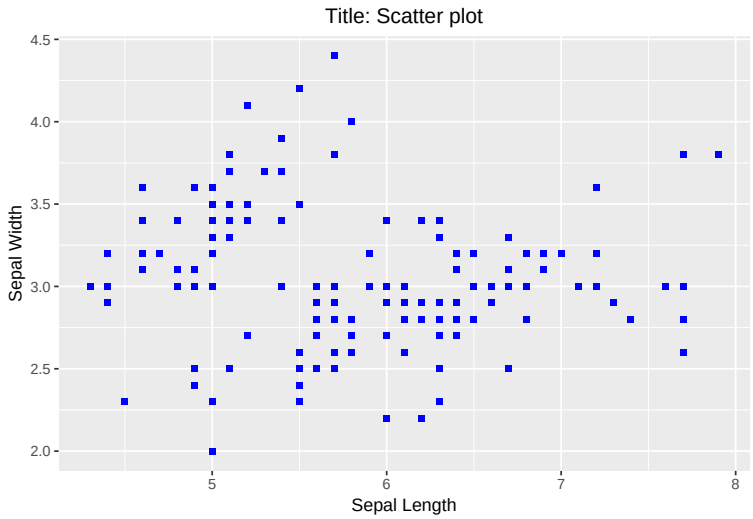



```
# Along columns

facet + facet_grid(Species ~ .)
```



3.3 Placing the title in the center

```
scatter = ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width))

scatter + geom_point(color = "blue", shape = 15) +

  xlab("Sepal Length") + ylab("Sepal Width") + ggtitle("Title: Scatter plot") +

  theme(plot.title = element_text(hjust = 0.5))
```



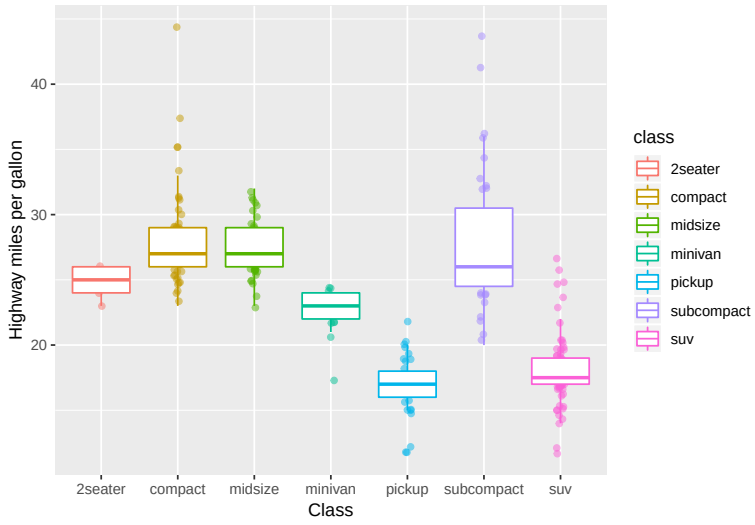
4 Additionally on ggplot2

4.1 Jitter

```
head(mpg)
```

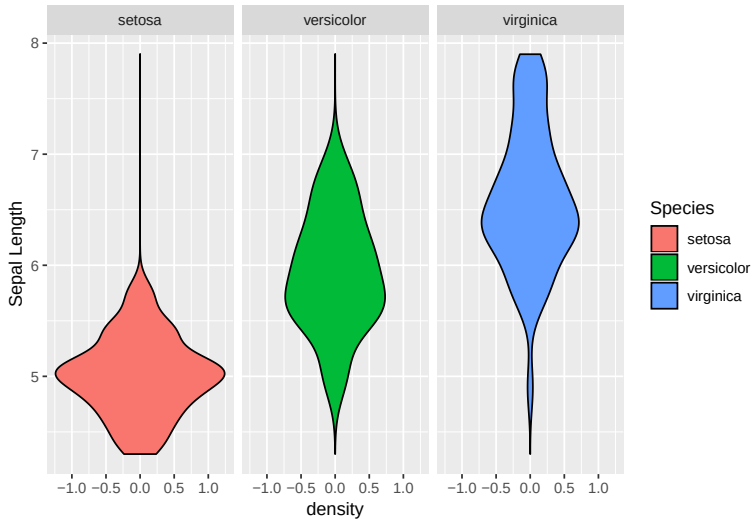
```
## # A tibble: 6 x 11
##   manufacturer model displ  year   cyl trans drv     cty   hwy fl    class
##   <chr>        <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
## 1 audi         a4      1.8  1999     4 auto~ f        18    29 p     comp~
## 2 audi         a4      1.8  1999     4 manu~ f        21    29 p     comp~
## 3 audi         a4      2    2008     4 manu~ f        20    31 p     comp~
## 4 audi         a4      2    2008     4 auto~ f        21    30 p     comp~
## 5 audi         a4      2.8  1999     6 auto~ f        16    26 p     comp~
## 6 audi         a4      2.8  1999     6 manu~ f        18    26 p     comp~
```

```
jitter = ggplot(mpg, aes(x = class, y = hwy))
jitter + scale_x_discrete() +
  geom_jitter(aes(x = class, color = class),
              position = position_jitter(width = .05), alpha = 0.5) +
  geom_boxplot(aes(color = class), outlier.colour = NA, position = "dodge") +
  xlab("Class") + ylab("Highway miles per gallon")
```



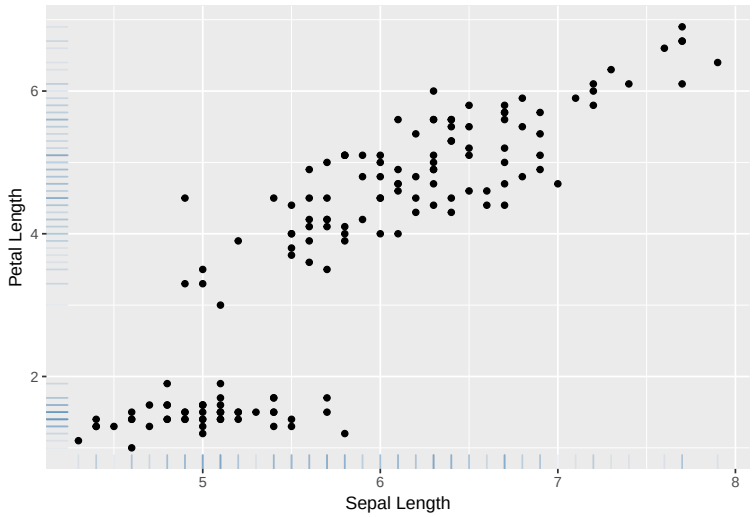
4.2 Volcano plot

```
vol = ggplot(data = iris, aes(x = Sepal.Length))
vol + stat_density(aes(ymax = ..density..,  ymin = -..density.., fill = Species),
                   color = "black", geom = "ribbon", position = "identity") +
  facet_grid(. ~ Species) + coord_flip() + xlab("Sepal Length")
```



4.3 Rug plot

```
ggplot(data = iris, aes(x = Sepal.Length, y = Petal.Length)) + geom_point() +
  geom_rug(col = "steelblue",alpha = 0.1) + xlab("Sepal Length") + ylab("Petal Length")
```

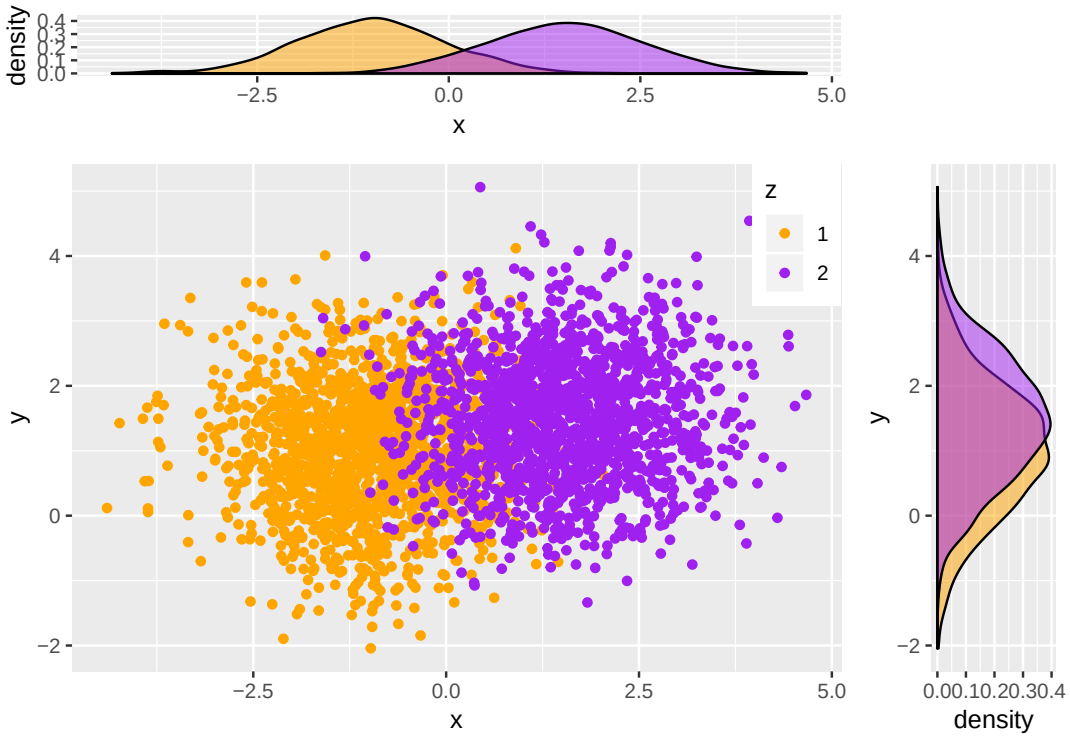


4.4 Density curves

(ggplot2 Cheatsheet from R for Public Health: http://http://felixfan.github.io/ggplot2-cheatsheet/)

```
library(gridExtra)
set.seed(1234)
x = c(rnorm(1500, mean = -1), rnorm(1500, mean = 1.5))
y = c(rnorm(1500, mean = 1), rnorm(1500, mean = 1.5))
z = as.factor(c(rep(1, 1500), rep(2, 1500)))
xy = data.frame(x, y, z)

# Scatterplot of x and y
scatter = ggplot(data = xy,aes(x = x, y = y)) + geom_point(aes(color = z)) +
  scale_color_manual(values = c("orange", "purple")) +
  theme(legend.position = c(1,1),legend.justification = c(1,1))

# Marginal density of x - plot on top
plot_top = ggplot(data = xy, aes(x = x, fill = z)) +
  geom_density(alpha = .5) +
  scale_fill_manual(values = c("orange", "purple")) +
  theme(legend.position = "none")
```

```r
# Marginal density of y - plot on the right
plot_right = ggplot(data = xy, aes(x = y, fill = z)) +
  geom_density(alpha = .5) + coord_flip() +
  scale_fill_manual(values = c("orange", "purple")) +
  theme(legend.position = "none")

# Empty plot
empty = ggplot() + geom_point(aes(1,1), color = "white") +
theme(
plot.background = element_blank(),
panel.grid.major = element_blank(),
panel.grid.minor = element_blank(),
panel.border = element_blank(),
panel.background = element_blank(),
axis.title.x = element_blank(),
axis.title.y = element_blank(),
axis.text.x = element_blank(),
axis.text.y = element_blank(),
axis.ticks = element_blank()
)

# Arrange the plots together
grid.arrange(plot_top, empty, scatter, plot_right, ncol = 2, nrow = 2,
             widths = c(4, 1), heights = c(1, 4))
```



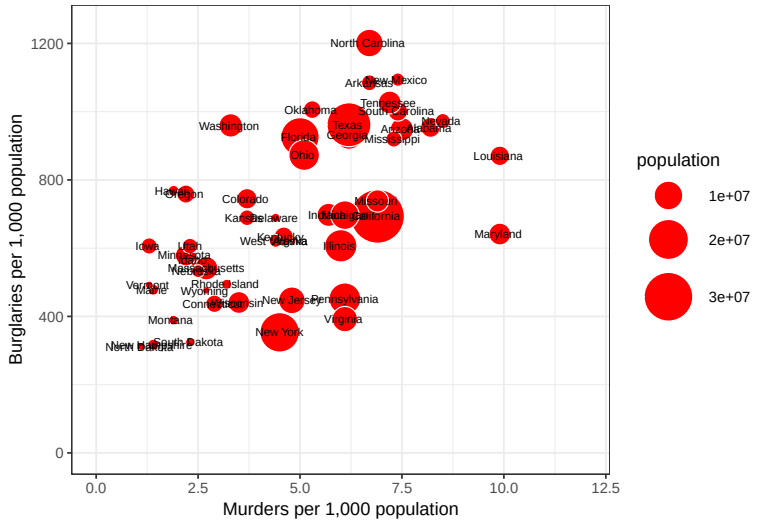
4.5 Bubble chart

```
crime = read.csv("http://datasets.flowingdata.com/crimeRatesByState2005.tsv",
                 header = TRUE, sep = "\t")
head(crime)
```

```
##         state murder Forcible_rate Robbery aggravated_assult burglary
## 1     Alabama    8.2          34.3   141.4             247.8    953.8
## 2      Alaska    4.8          81.1    80.9             465.1    622.5
## 3     Arizona    7.5          33.8   144.4             327.4    948.4
## 4    Arkansas    6.7          42.9    91.1             386.8   1084.6
## 5  California    6.9          26.0   176.1             317.3    693.3
## 6    Colorado    3.7          43.4    84.6             264.7    744.8
##   larceny_theft motor_vehicle_theft population
## 1        2650.0               288.3    4627851
## 2        2599.1               391.0     686293
## 3        2965.2               924.4    6500180
## 4        2711.2               262.1    2855390
```

```
## 5        1916.5                712.8   36756666
## 6        2735.2                559.5    4861515
```

```
ggplot(data = crime, aes(x = murder, y = burglary, size = population, label = state)) +
geom_point(color = "white", fill = "red", shape = 21) + scale_size_area(max_size = 15) +
scale_x_continuous(name = "Murders per 1,000 population", limits = c(0,12)) +
scale_y_continuous(name = "Burglaries per 1,000 population", limits = c(0,1250)) +
geom_text(size = 2.5) + theme_bw()
```

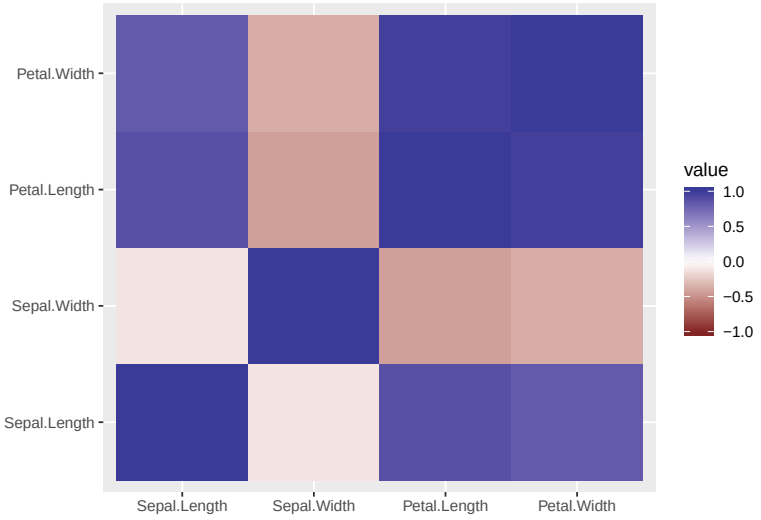


4.6 Heat map

```
# Heat Map 1
dat = iris[,1:4]
cor = melt(cor(dat, use = "p"))
head(cor)
```

```
##          Var1         Var2      value
## 1 Sepal.Length Sepal.Length  1.0000000
## 2  Sepal.Width Sepal.Length -0.1175698
## 3 Petal.Length Sepal.Length  0.8717538
## 4  Petal.Width Sepal.Length  0.8179411
## 5 Sepal.Length  Sepal.Width -0.1175698
## 6  Sepal.Width  Sepal.Width  1.0000000
```

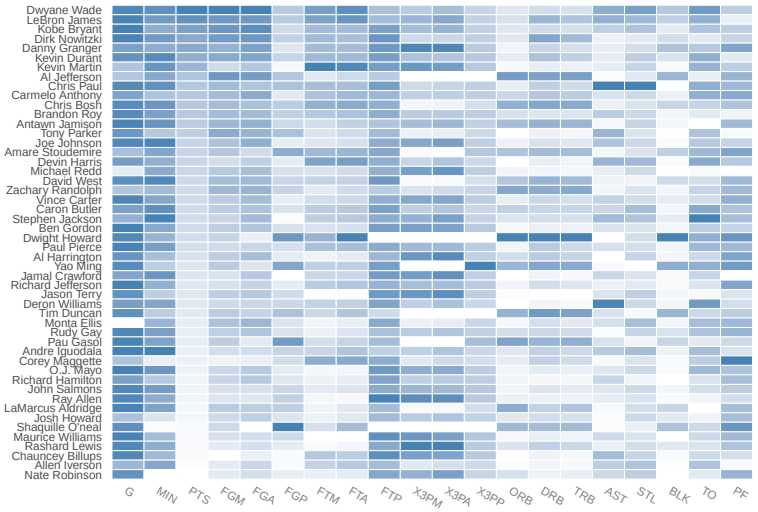
```
heat = ggplot(data = cor, aes(x = Var1, y = Var2, fill = value))
heat + geom_tile() + labs(x = "", y = "") + scale_fill_gradient2(limits = c(-1, 1))
```




```
# Heat Map 2
# (Learning R: https://learnr.wordpress.com)


nba = read.csv("http://datasets.flowingdata.com/ppg2008.csv")
head(nba)
```

```
##             Name  G  MIN  PTS  FGM  FGA   FGP FTM FTA   FTP X3PM X3PA
## 1   Dwyane Wade  79 38.6 30.2 10.8 22.0 0.491 7.5 9.8 0.765  1.1  3.5
## 2  LeBron James  81 37.7 28.4  9.7 19.9 0.489 7.3 9.4 0.780  1.6  4.7
## 3   Kobe Bryant  82 36.2 26.8  9.8 20.9 0.467 5.9 6.9 0.856  1.4  4.1
## 4 Dirk Nowitzki  81 37.7 25.9  9.6 20.0 0.479 6.0 6.7 0.890  0.8  2.1
## 5 Danny Granger  67 36.2 25.8  8.5 19.1 0.447 6.0 6.9 0.878  2.7  6.7
## 6  Kevin Durant  74 39.0 25.3  8.9 18.8 0.476 6.1 7.1 0.863  1.3  3.1
##    X3PP ORB DRB TRB AST STL BLK  TO  PF
## 1 0.317 1.1 3.9 5.0 7.5 2.2 1.3 3.4 2.3
## 2 0.344 1.3 6.3 7.6 7.2 1.7 1.1 3.0 1.7
## 3 0.351 1.1 4.1 5.2 4.9 1.5 0.5 2.6 2.3
## 4 0.359 1.1 7.3 8.4 2.4 0.8 0.8 1.9 2.2
## 5 0.404 0.7 4.4 5.1 2.7 1.0 1.4 2.5 3.1
## 6 0.422 1.0 5.5 6.5 2.8 1.3 0.7 3.0 1.8
```

```
library(scales)


nba$Name = with(nba, reorder(Name, PTS))
nba.m = melt(nba)

## Using Name as id variables
nba.m = ddply(nba.m, .(variable), transform, rescale = rescale(value))


heat = ggplot(data = nba.m, aes(x = variable, y = Name)) +
  geom_tile(aes(fill = rescale), color = "white") +
  scale_fill_gradient(low = "white", high = "steelblue")


base_size = 9
heat + theme_grey(base_size = base_size) + labs(x = "", y = "") +
  scale_x_discrete(expand = c(0, 0)) + scale_y_discrete(expand = c(0, 0)) +
  theme(legend.position = "none", axis.ticks = element_blank(),
        axis.text.x = element_text(size = base_size * 0.8,
                                   angle = 330, hjust = 0, color = "grey50"))
```



4.7 Exporting

```
plot = ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width)) +
  geom_point(aes(shape = Species, color = Species))


ggsave("plot1.png")
ggsave(plot, file = "plot2.png")
ggsave(plot, file = "plot3.png", width = 6, height = 4)
```

5 Useful Resources

- R Cookbook: http://www.cookbook-r.com
- ggplot2 geoms: http://docs.ggplot2.org/current/
- Be Colorful!: http://tools.medialab.sciences-po.fr/iwanthue
- Christophe Ladroue: http://chrisladroue.com