Data Visualization

https://blog.modeanalytics.com/r-data-visualization-packages/

1. Line and Scatter Plots
2. Bar Plots
3. Pie Charts
4. Histograms
5. Bubble Plots
6. Box Plots
7. 3D Plots
8. Saving Plots

References

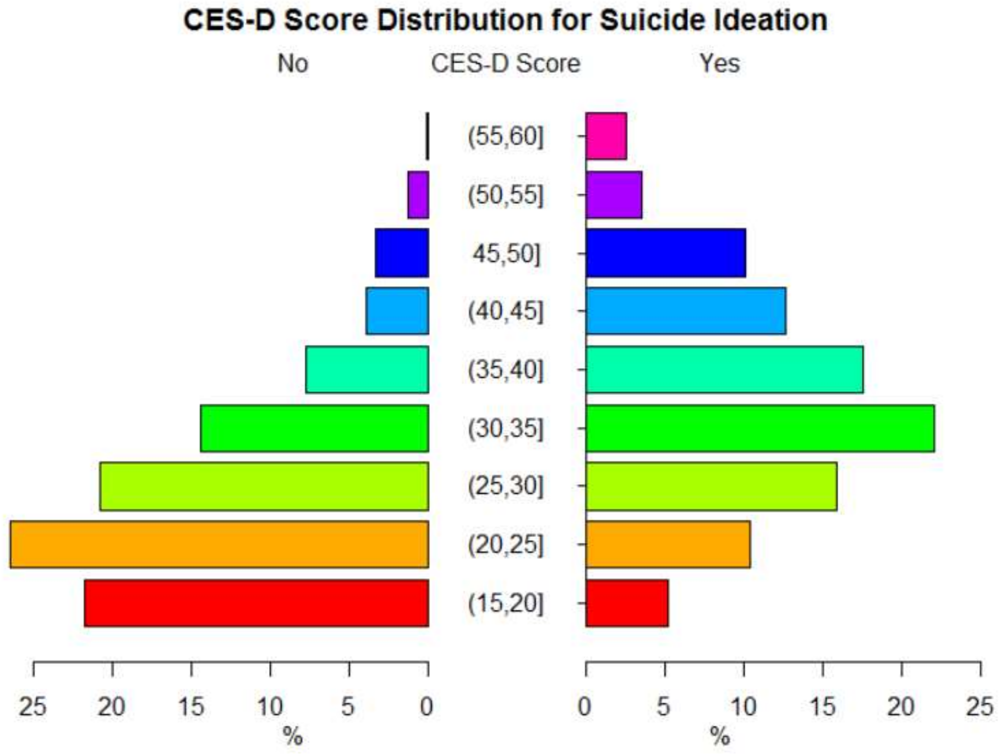
[1] https://www.r-graph-gallery.com/all-graphs/

Data Visualization: What it is and why it matters

[Source] https://www.sas.com/en_us/insights/big-data/data-visualization.html

►It enables decision makers to see analytics presented visually, so they can grasp difficult concepts or identify new patterns.

►It makes us to be able to dig for more insights – look at data differently, more imaginatively.



1. Line and Scatter Plots

plot(x, y, pch, type, lty, ...) {graphics}
Generic function for plotting of R objects.

points(x, y, type = "p", ...) {graphics}
Add Points to a Plot

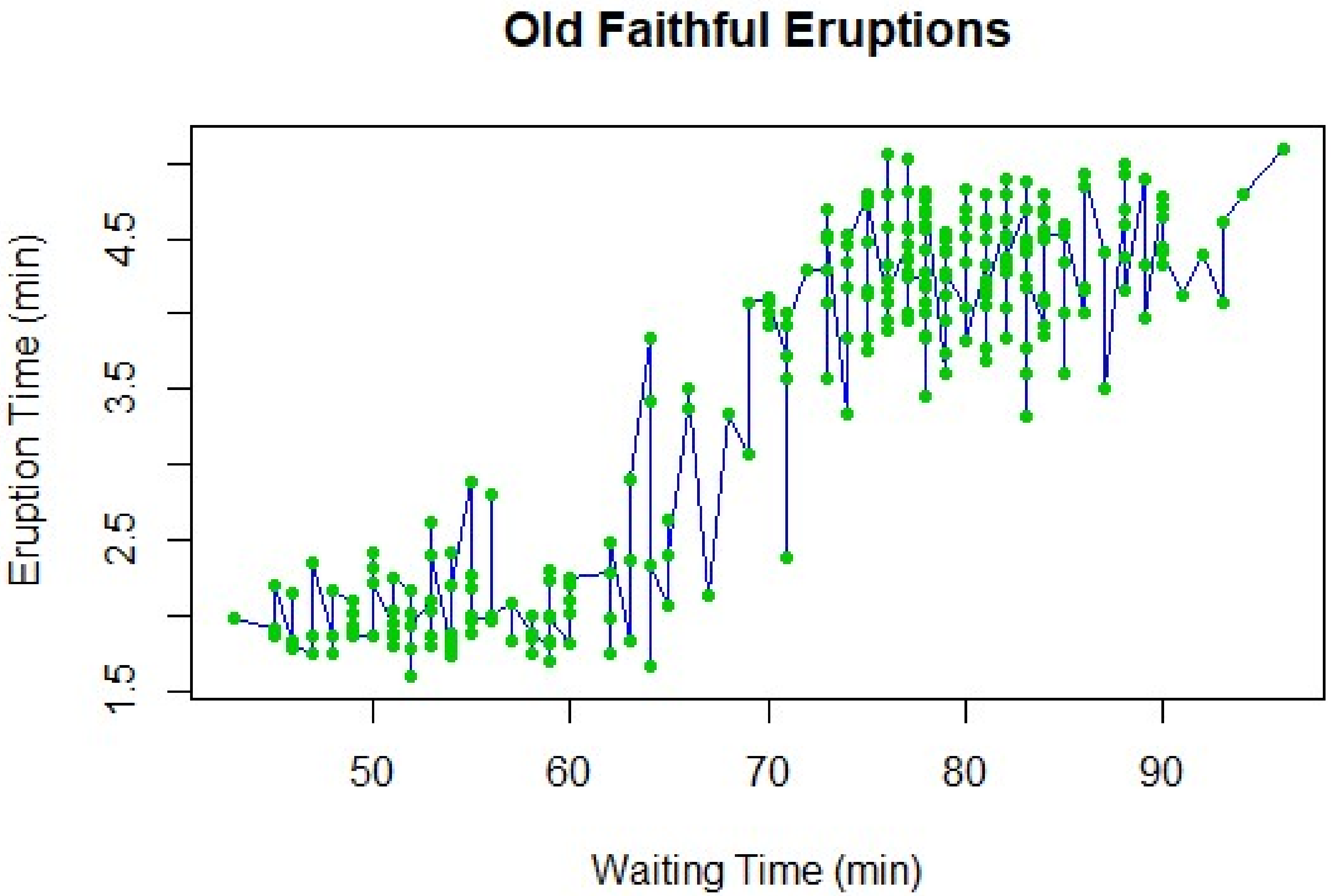
pch= R plotting symbols.

type = "p" for **p**oints, "l" for **l**ines, "b" for **b**oth,
"c" for the lines part alone of "b", "o" for both '**o**verplotted',
"h" for '**h**istogram' like vertical lines, "s" for stair **s**teps,
"S" for other **s**teps, "n" for no plotting.

faithful {datasets} Old Faithful Geyser Data
Waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park, Wyoming, USA.
eruptions : Eruption time (in mins)
waiting : Waiting time to next eruption (in mins)

```
> head(faithful)
  eruptions waiting
1     3.600      79
2     1.800      54
3     3.333      74
4     2.283      62
5     4.533      85
6     2.883      55
> #sorting faithful data.frame by waiting
> fa <- faithful[order(faithful$waiting),]
> head(fa)
    eruptions waiting
265     1.983      43
127     1.917      45
131     1.867      45
161     2.200      45
135     1.833      46
188     1.833      46
```

```
x <- fa[,2]; y <- fa[,1]
plot(x, y, type='l', col=4, xlab='Waiting Time (min)',
     ylab='Eruption Time (min)', main='Old Faithful Eruptions')
points(x,y,pch=20,col=3)
```



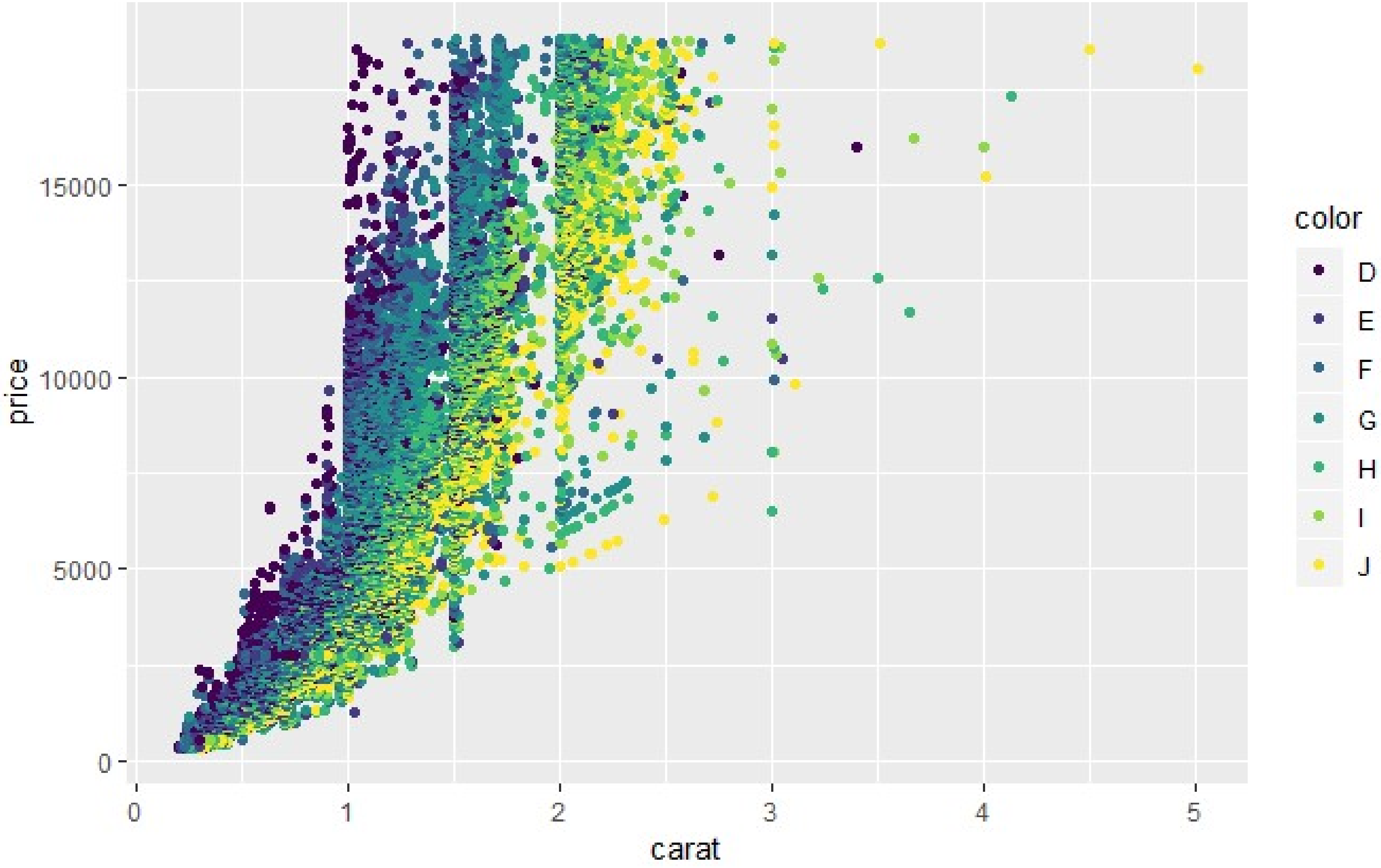
ggplot(data, mapping=aes(), ...) {ggplot2}
Create a new ggplot
  geom_line() Connect the observations, ordered by x value.
  geom_point() Points, as for a scatterplot.

```
#(2)
library(ggplot2)
ggplot(fa, aes(x, y), xtitle='Waiting Time (min)') +
  geom_point(col=3) + geom_line(col=4) +
  xlab('Waiting Time (min)') +
  ylab('Eruption Time (min)') +
  ggtitle('Old Faithful Eruptions')
```

diamonds {ggplot2}

A dataset containing the prices and other attributes of almost 54,000 diamonds.

```
#(3)
head(diamonds)
ggplot(data=diamonds,aes(x=carat,y=price)) +
  geom_point(aes(color=color))
```



2. Bar Plots

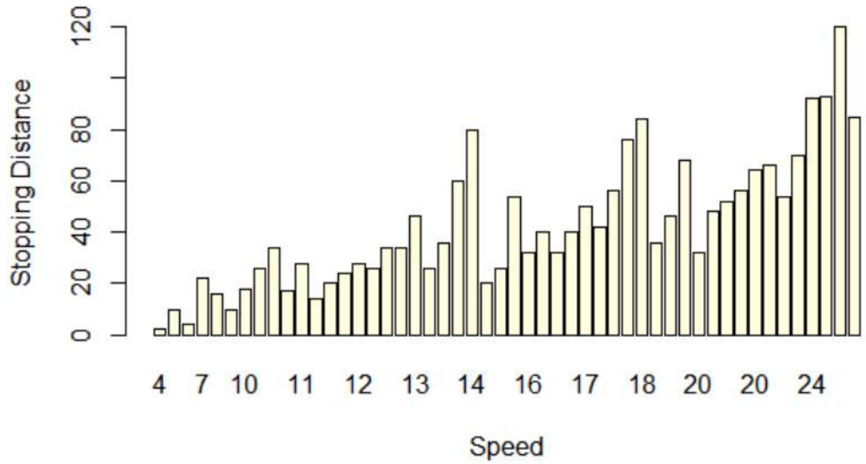
barplot(height, ...) {graphics}
Creates a bar plot with vertical or horizontal bars.

(1) Simple Bar Plot

cars {datasets}
Speed and Stopping Distances of Cars
speed : speed (mph)
dist : stopping distance (ft)
[**Source**] Ezekiel, M. (1930) Methods of Correlation Analysis. Wiley.

```
> head(cars)
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
```

```
barplot(cars[,2],col="cornsilk",
        names.arg=cars[,1],
        xlab="Speed",
        ylab="Stopping Distance")
```

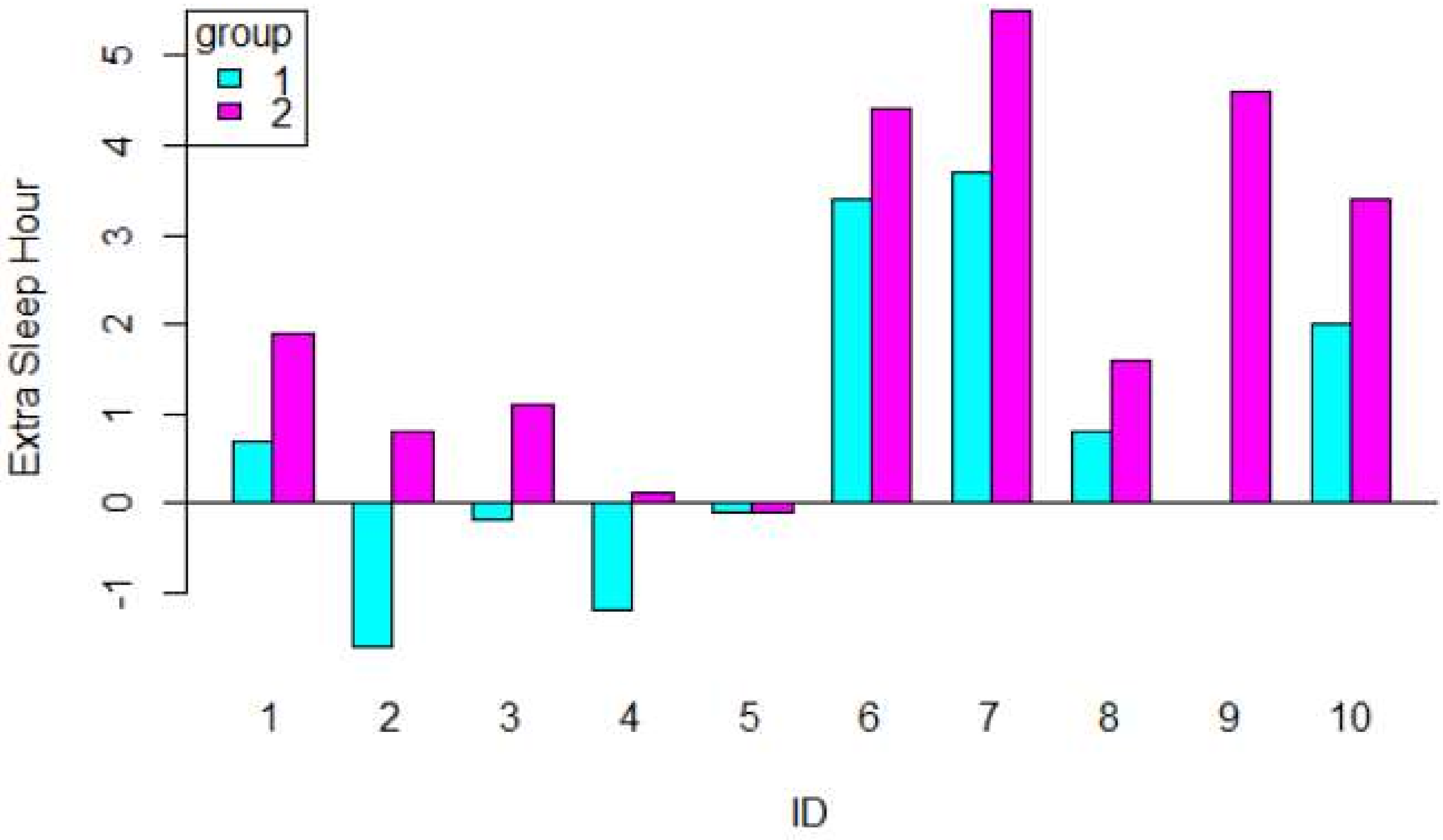


(2) Grouped Bar plot

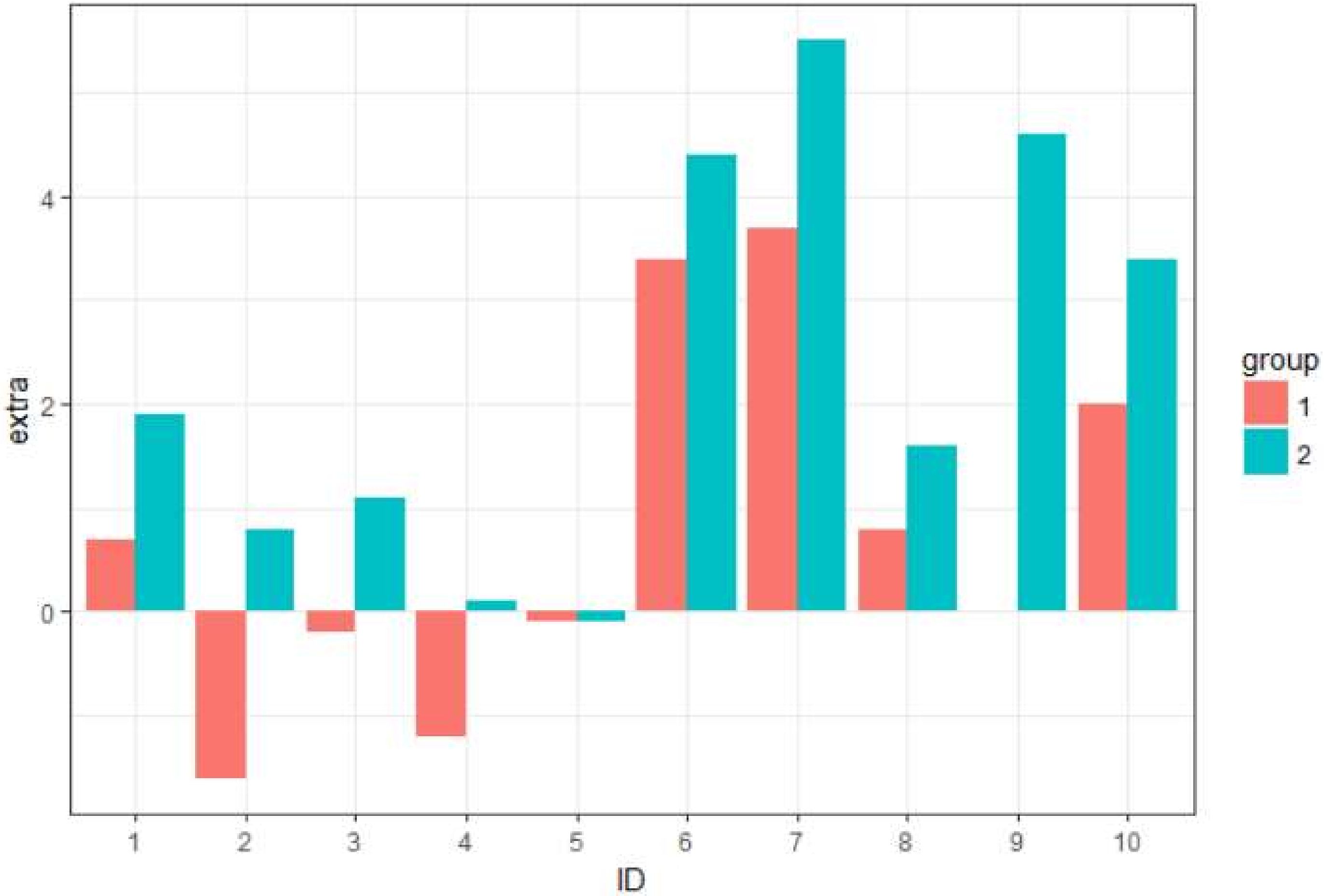
sleep {datasets}
Data which show the effect of two soporific drugs (increase in hours of sleep compared to control) on 10 patients.
extra : increase in hours of sleep
group : drug given
ID : patient ID

```
> attach(sleep)
> sleep #list data
   extra group ID
1    0.7     1  1
2   -1.6     1  2
3   -0.2     1  3
4   -1.2     1  4
5   -0.1     1  5
6    3.4     1  6
7    3.7     1  7
8    0.8     1  8
9    0.0     1  9
10   2.0     1 10
11   1.9     2  1
12   0.8     2  2
13   1.1     2  3
14   0.1     2  4
15  -0.1     2  5
16   4.4     2  6
17   5.5     2  7
18   1.6     2  8
19   4.6     2  9
20   3.4     2 10
> y <- rbind(extra[1:10],extra[11:20])
> y
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]  0.7 -1.6 -0.2 -1.2 -0.1  3.4  3.7  0.8  0.0   2.0
[2,]  1.9  0.8  1.1  0.1 -0.1  4.4  5.5  1.6  4.6   3.4
```

```
barplot(y, beside=T, col=5:6, names.arg=ID[1:10],
        xlab="ID",ylab="Extra Sleep Hour")
abline(h=0)
legend('topleft', title='group', legend=1:2, fill=5:6)
```



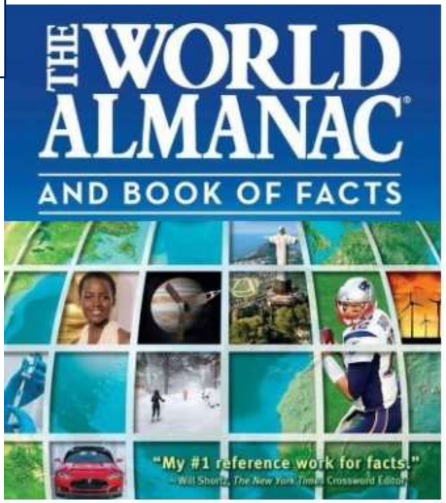
```
#(2) visualization in ggplot2
library(ggplot2)
ggplot(sleep, aes(x=ID, y=extra, fill=group)) +
    geom_bar(stat="identity",position="dodge") + theme_bw()
```



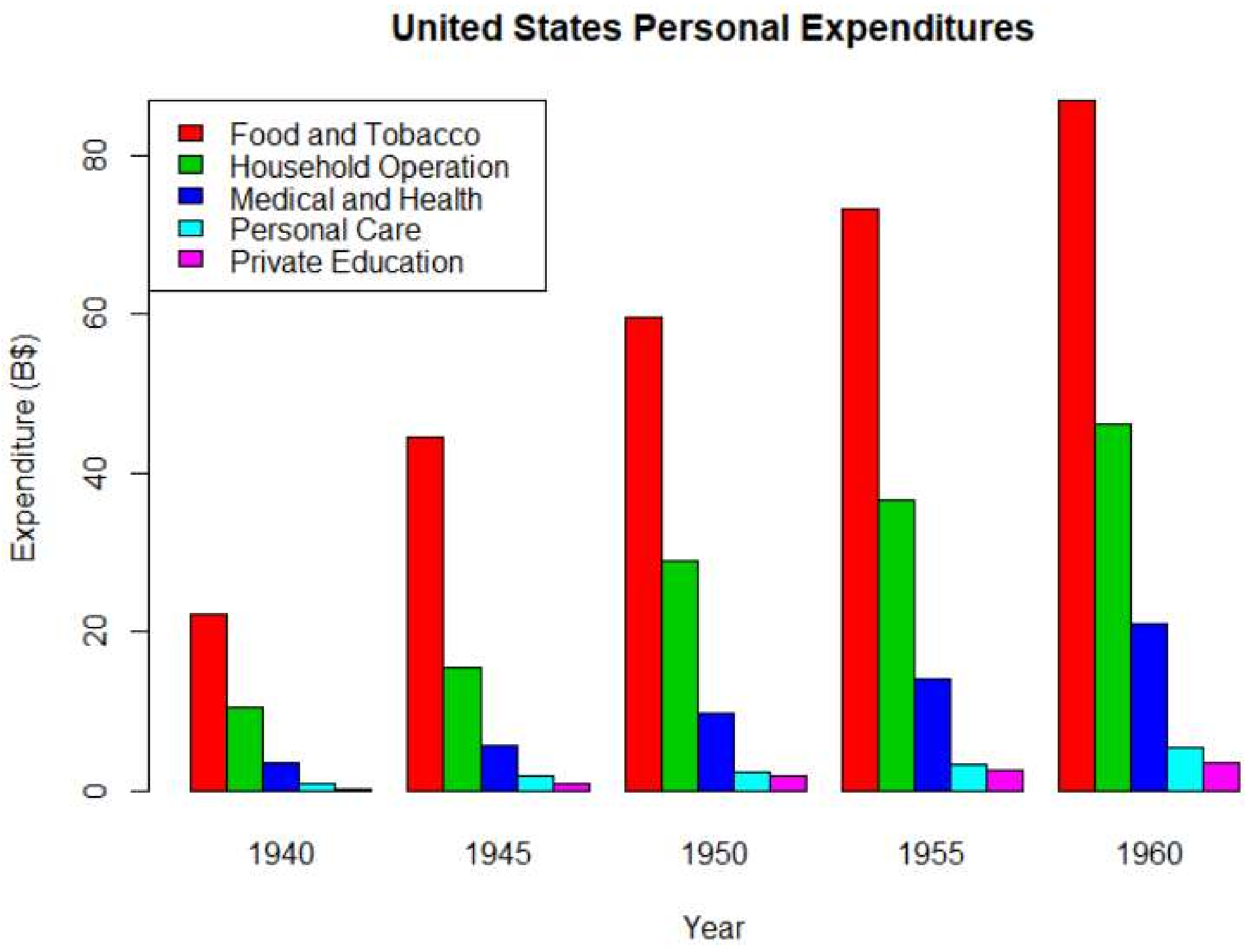
(3) Bar Plot of Matrix/Table Data

USPersonalExpenditure {datasets} Personal Expenditure Data
This data set consists of United States personal expenditures (in billions of dollars) in the categories; food and tobacco, household operation, medical and health, personal care, and private education for the years 1940, 1945, 1950, 1955 and 1960.
[**Source**] The World Almanac and Book of Facts, 1962, page 756.

```
> data(USPersonalExpenditure)
> UPE <- USPersonalExpenditure
> UPE
                      1940   1945  1950 1955  1960
Food and Tobacco    22.200 44.500 59.60 73.2 86.80
Household Operation 10.500 15.500 29.00 36.5 46.20
Medical and Health   3.530  5.760  9.71 14.0 21.10
Personal Care        1.040  1.980  2.45  3.4  5.40
Private Education    0.341  0.974  1.80  2.6  3.64
```



```
barplot(UPE, beside=T, col=2:6, ylab="Expenditure (B$)",
        xlab='Year', main='United States Personal Expenditures')
legend('topleft',legend=row.names(UPE), fill=2:6)
```



3. Pie Charts

pie(x, labels, ...) {graphics}
Draw a pie chart.

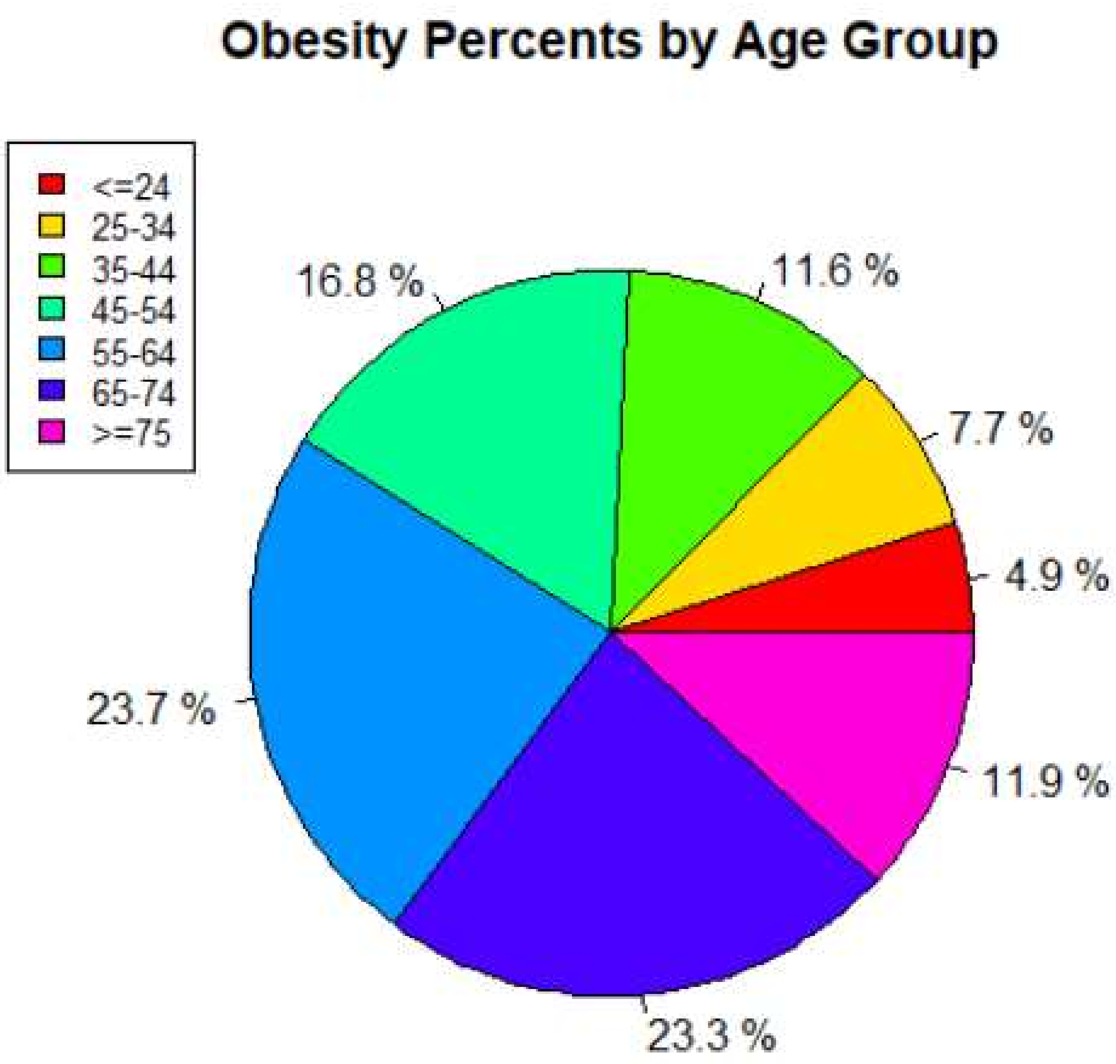
Table 1 Description of the demographic factors (%) of individuals whose body mass index (BMI) was ≥30

Variable	*Obese sample (BMI ≥30)*
Age	
≤24	4.9
25–34	7.7
35–44	11.6
45–54	16.8
55–64	23.7
65–74	23.3
≥75	11.9

[**Source**] M. A. Green et al., *Journal of Public Health*, **38** (2016) pp. 258 –264

```
#(1) Simple pie chart
#sample data
Age <- c('<=24','25-34','35-44','45-54','55-64','65-74','>=75')
Obese <- c(4.9,7.7,11.6,16.8,23.7,23.3,11.9)
```

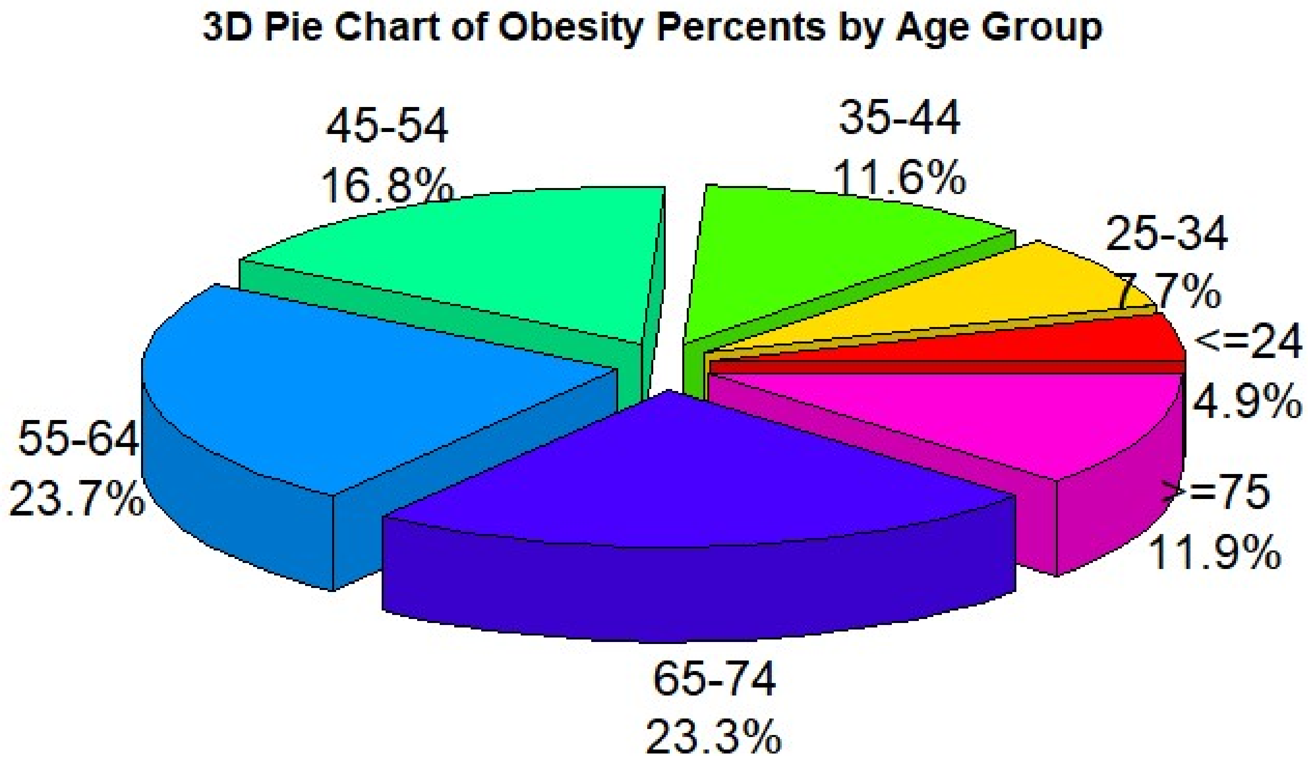
```
#plot pie chart
pie(Obese, labels=paste(Obese,'%'), main='Obesity Percents by Age Group',
    col=rainbow(length(Age)))
legend("topleft", Age, cex=0.8, fill=rainbow(length(Age)))
```



pie3D(x, labels, explode, col, ...) {plotrix}

Displays a 3D pie chart with optional labels.

```
library(plotrix)
xb <- paste(Age,"\n",Obese,'%',sep="")
pie3D(Obese, labels=xb, explode=0.1,
      col=rainbow(length(Age)),
      main="3D Pie Chart of Obesity Percents by Age Group")
```



4. Histograms

hist(x, breaks, freq, ...) {graphics}

The hist( ) function computes a histogram of the given data values.

cane {boot}   Sugar-cane Disease Data

n : The total number of shoots in each plot.

r : The number of diseased shoots.

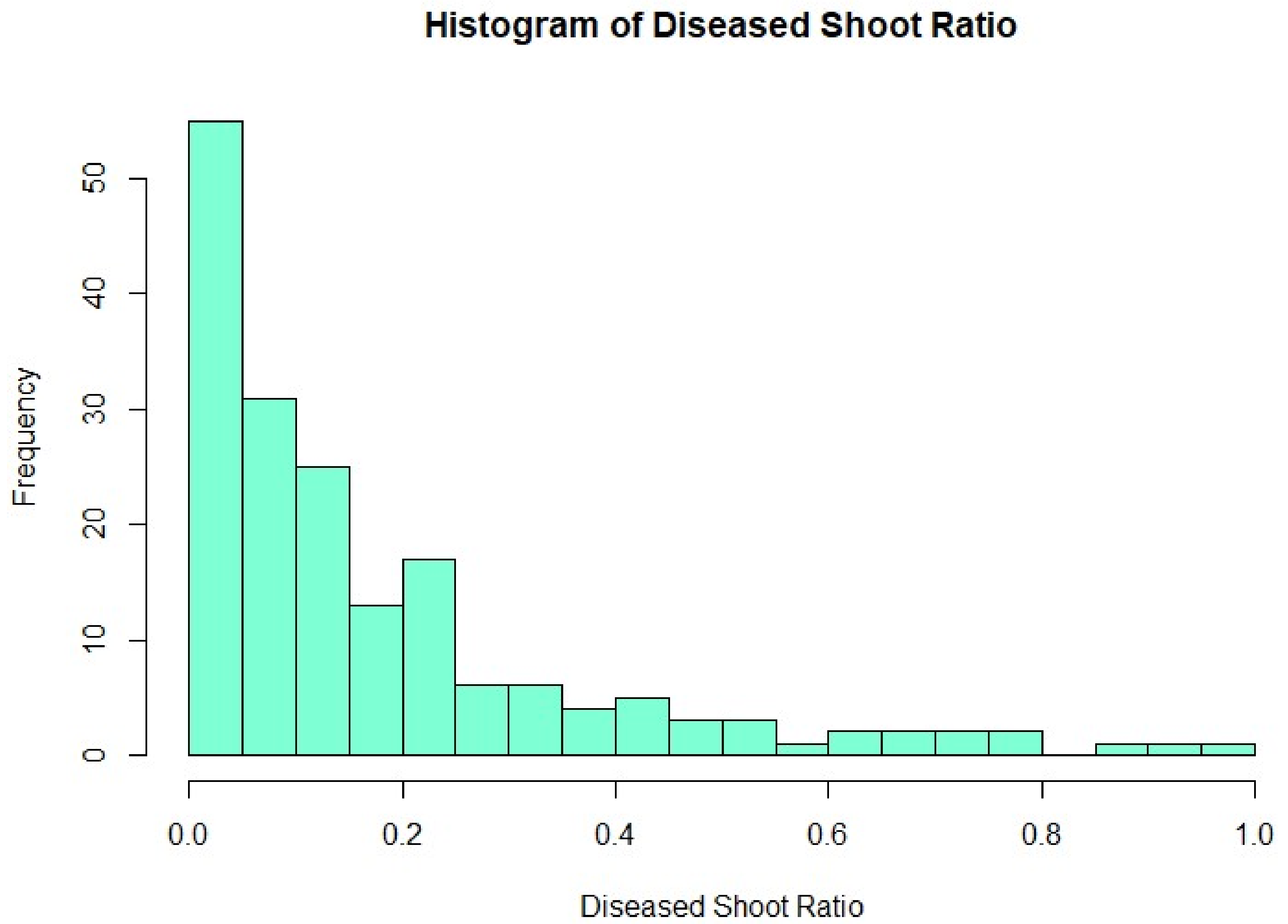
x : The number of pieces of the stems, out of 50

var : A factor indicating the variety of sugar-cane in each plot.

block : A factor for the blocks.

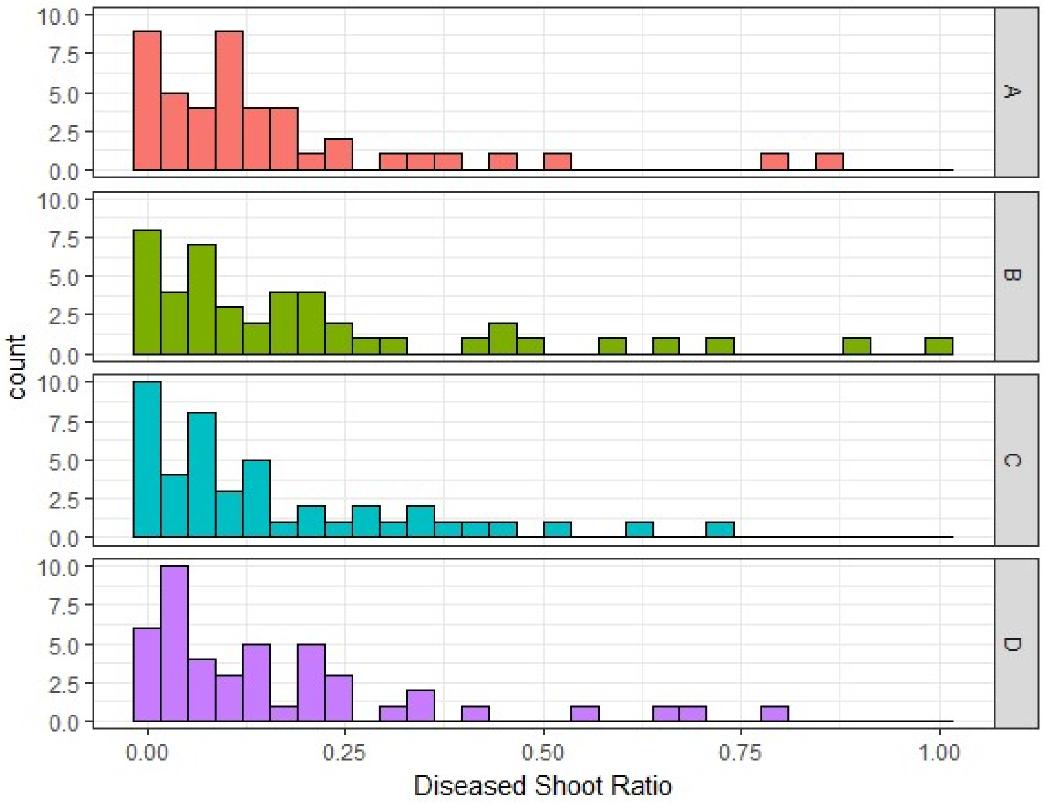
```
> #sample data: cane
> library(boot)
> dim(cane)
[1] 180   5
> str(cane)
'data.frame':   180 obs. of  5 variables:
 $ n    : num  87 119 94 95 134 92 118 70 128 85 ...
 $ r    : num  76 8 74 11 0 0 11 32 33 14 ...
 $ x    : num  19 14 9 12 12 3 17 3 3 21 ...
 $ var  : Factor w/ 45 levels "1","10","11",..: 1 12 23 34 41 42 43 44 45 2 ...
 $ block: Factor w/ 4 levels "A","B","C","D": 1 1 1 1 1 1 1 1 1 1 ...
> head(cane)
    n  r  x var block
1  87 76 19   1     A
2 119  8 14   2     A
3  94 74  9   3     A
4  95 11 12   4     A
5 134  0 12   5     A
6  92  0  3   6     A
```

```
#(1) Simple Histogram
ratio <- cane$r/cane$n
hist(ratio, breaks=20, xlab='Diseased Shoot Ratio', col='aquamarine',
     main='Histogram of Diseased Shoot Ratio')
```



facet_grid( ) {ggplot2}
Lay out panels in a rectangular/tabular manner.

```
#(2)Histograms of Diseased Shoot Ratio by block
library(ggplot2)
ggplot(cane, aes(x=r/n, fill=block)) +
  geom_histogram(colour="black") + theme_bw() +
  facet_grid(block ~ .) + xlab('Diseased Shoot Ratio') +
  theme(legend.position="none")
```



5. Bubble Plots

Sample Data: USArrests

USArrests {datasets} Violent Crime Rates by US State

This data set contains statistics, in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973.

Murder : Murder arrests (per 100,000)

Assault : Assault arrests (per 100,000)

UrbanPop : Percent urban population

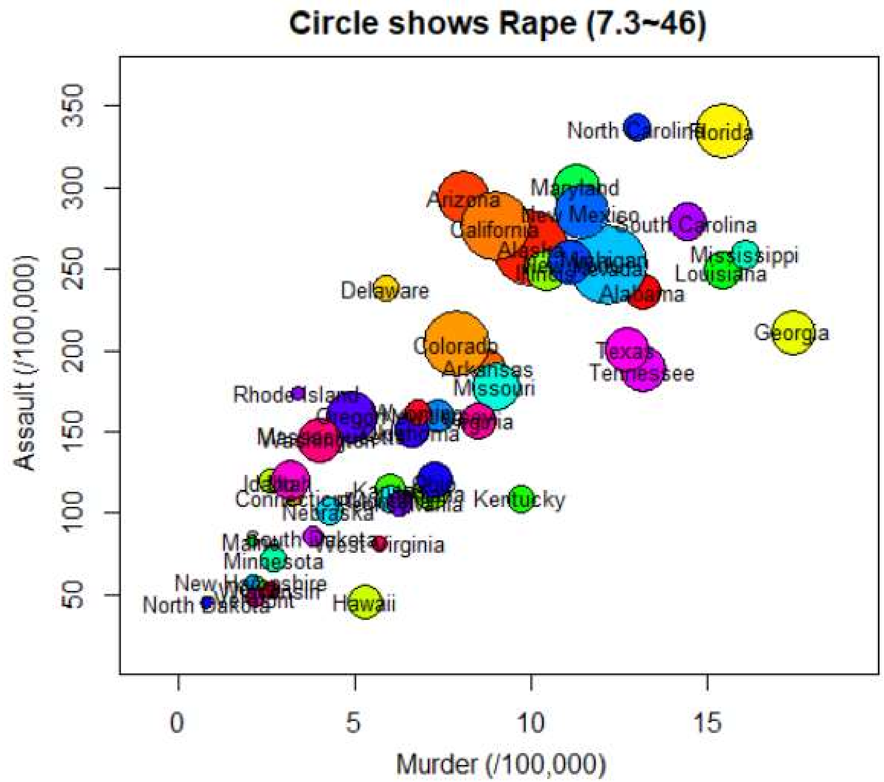
Rape : Rape arrests (per 100,000)

```
> attach(USArrests)
> head(USArrests)
           Murder Assault UrbanPop Rape
Alabama      13.2     236       58 21.2
Alaska       10.0     263       48 44.5
Arizona       8.1     294       80 31.0
Arkansas      8.8     190       50 19.5
California    9.0     276       91 40.6
Colorado      7.9     204       78 38.7
> summary(USArrests)
     Murder          Assault         UrbanPop          Rape      
 Min.   : 0.800   Min.   : 45.0   Min.   :32.00   Min.   : 7.30  
 1st Qu.: 4.075   1st Qu.:109.0   1st Qu.:54.50   1st Qu.:15.07  
 Median : 7.250   Median :159.0   Median :66.00   Median :20.10  
 Mean   : 7.788   Mean   :170.8   Mean   :65.54   Mean   :21.23  
 3rd Qu.:11.250   3rd Qu.:249.0   3rd Qu.:77.75   3rd Qu.:26.18  
 Max.   :17.400   Max.   :337.0   Max.   :91.00   Max.   :46.00  
```

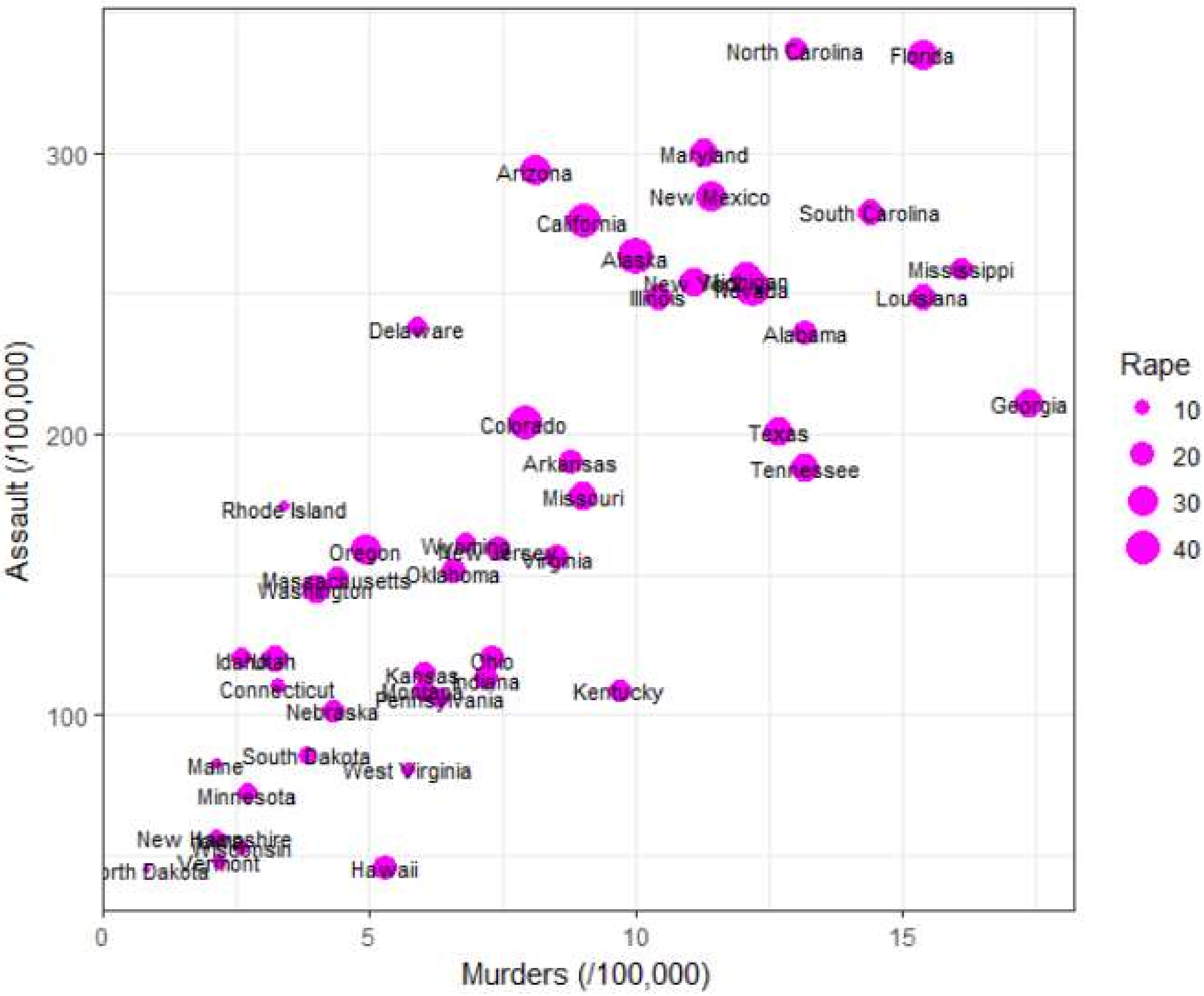
symbols(x, y, circles, ...) {graphics}
Draw Symbols (Circles, Squares, Stars, Thermometers, Boxplots)

palette(value) {grDevices}
View or manipulate the color palette which is used when a col= has a numeric index.

```
#(1)
radius <- Rape/max(Rape) #circle size
N <- nrow(USArrests)
op <- palette(rainbow(N))
symbols(Murder, Assault, circles=radius,
        inches=0.25, fg='black', bg=1:N,
        xlab='Murder (/100,000)',
        ylab='Assault (/100,000)',
        main='Circle shows Rape (7.3~46)')
text(Murder, Assault, row.names(USArrests),
     cex=0.8)
palette(op)
```



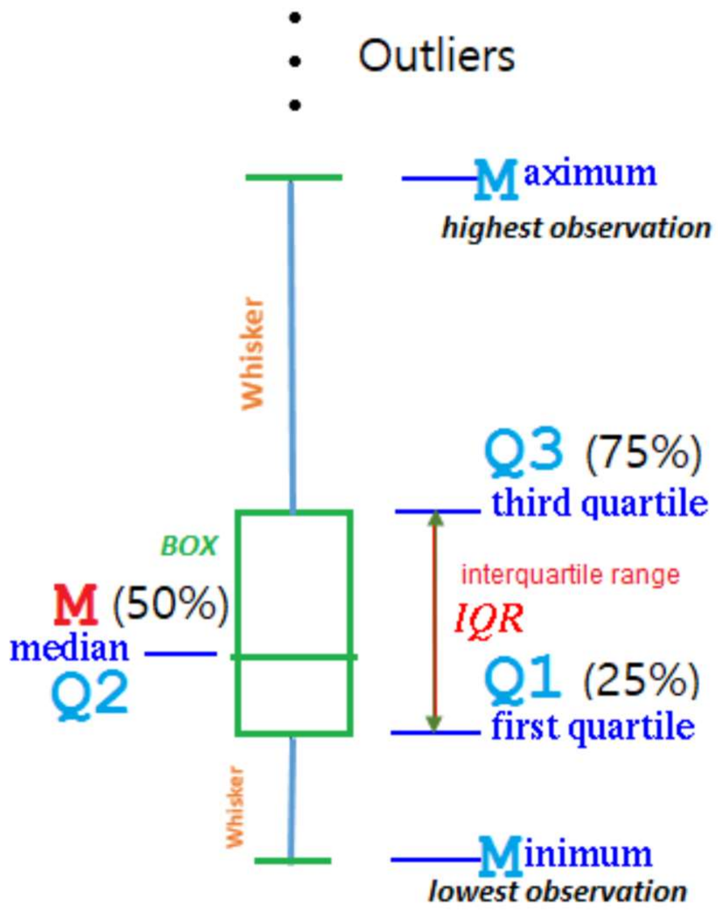
```
#(2)
library(ggplot2)
ggplot(USArrests, aes(Murder,Assault,size=Rape,label=row.names(USArrests))) +
geom_point(colour="magenta")  + geom_text(size=3) + theme_bw() +
xlab("Murders (/100,000)") + ylab("Assault (/100,000)")
```



6. Box Plots

boxplot(x, data, ...) {graphics}
Produce box-and-whisker plot(s) of the given (grouped) values.

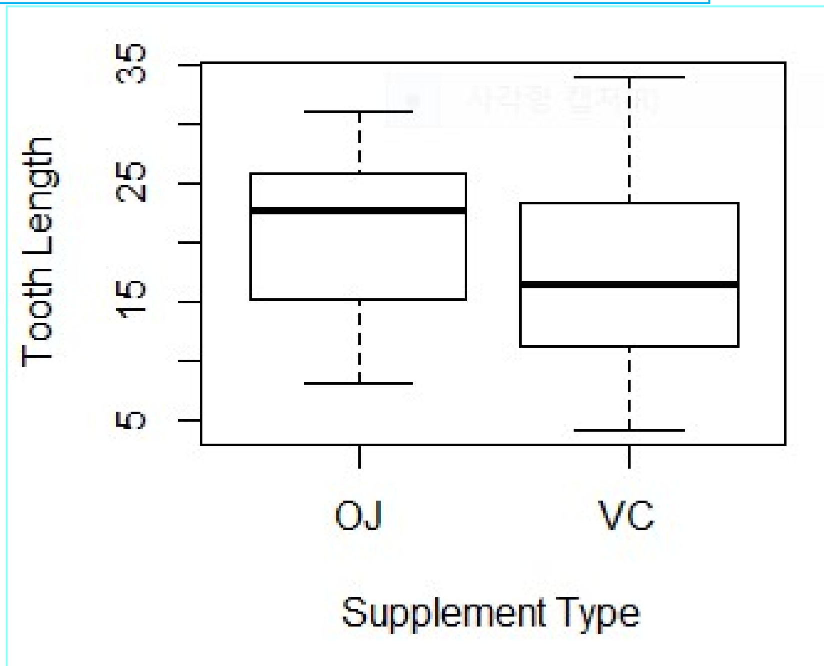
A **box plot** is used to depict groups of numerical data through their quartiles.



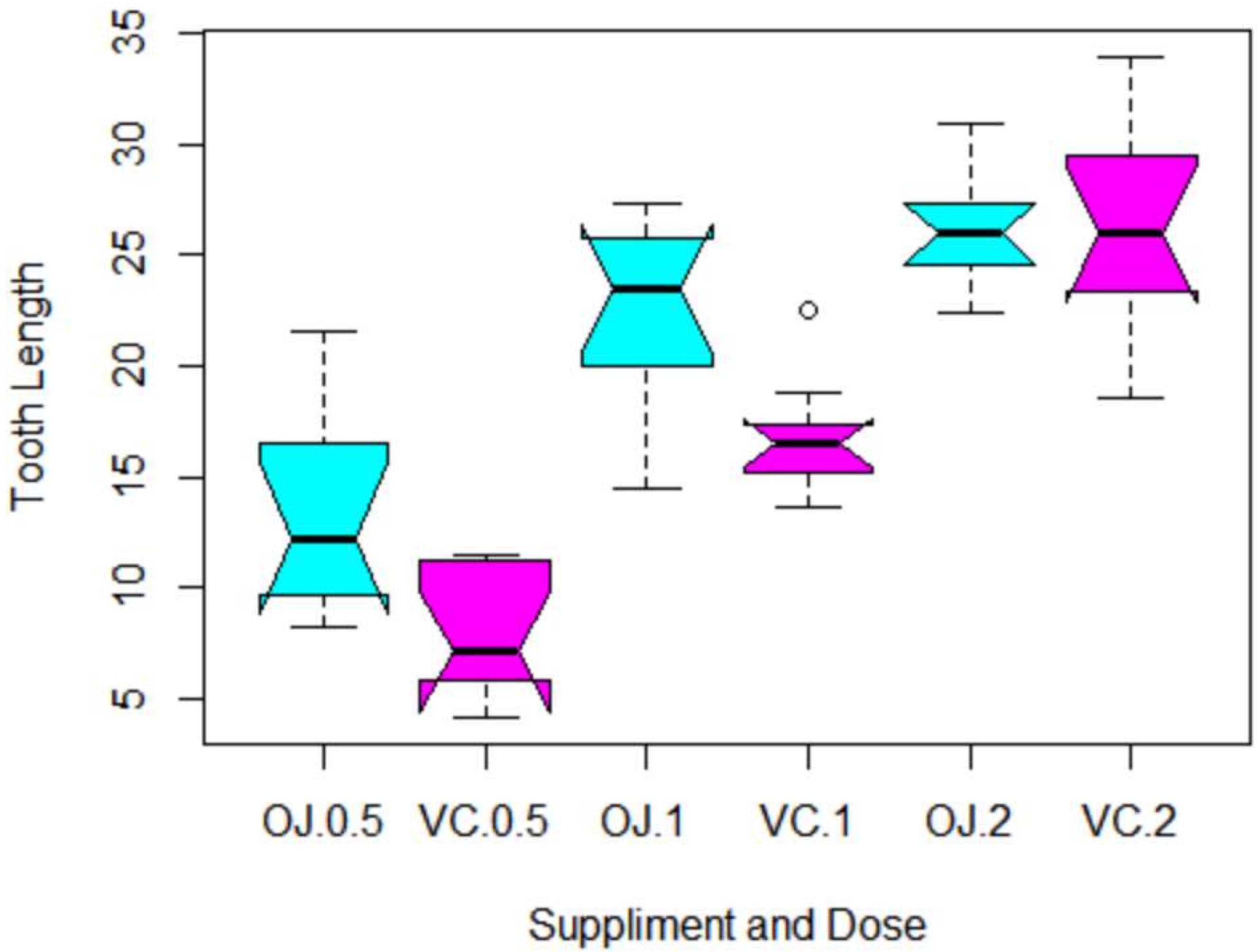
ToothGrowth {datasets}
The Effect of Vitamin C on Tooth Growth in Guinea Pigs
len : Tooth length
supp : Supplement type (VC or OJ).
dose : Dose in milligrams/day

```
> head(ToothGrowth)
   len supp dose
1  4.2   VC  0.5
2 11.5   VC  0.5
3  7.3   VC  0.5
4  5.8   VC  0.5
5  6.4   VC  0.5
6 10.0   VC  0.5
```

```
#(1) Simple boxplot
boxplot(len~supp, data=ToothGrowth,
  xlab="Supplement Type",ylab="Tooth Length")
```

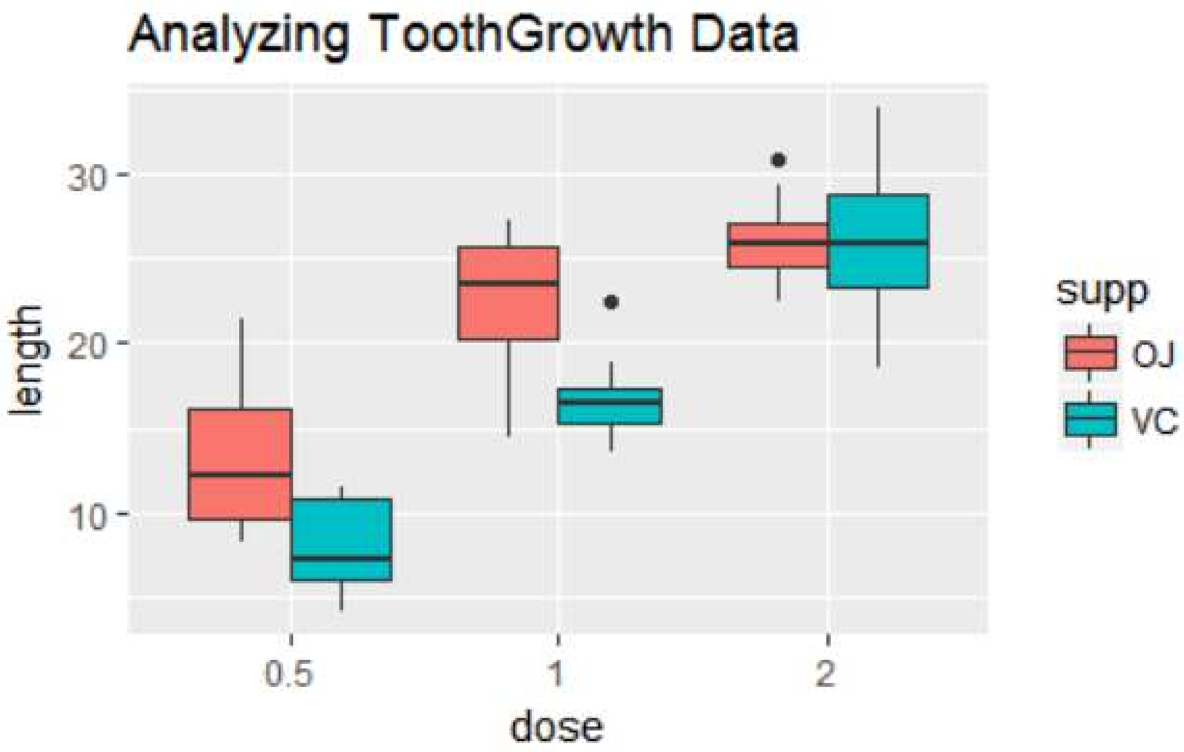


```
#(2) Boxplot of len against dose and supp factors
boxplot(len~supp*dose, data=ToothGrowth, notch=T,
 xlab="Suppliment and Dose",ylab="Tooth Length",
 col=c("cyan","magenta"))
```



geom_boxplot(mapping, data, ...) {ggplot2}
A box and whiskers plot (in the style of Tukey)

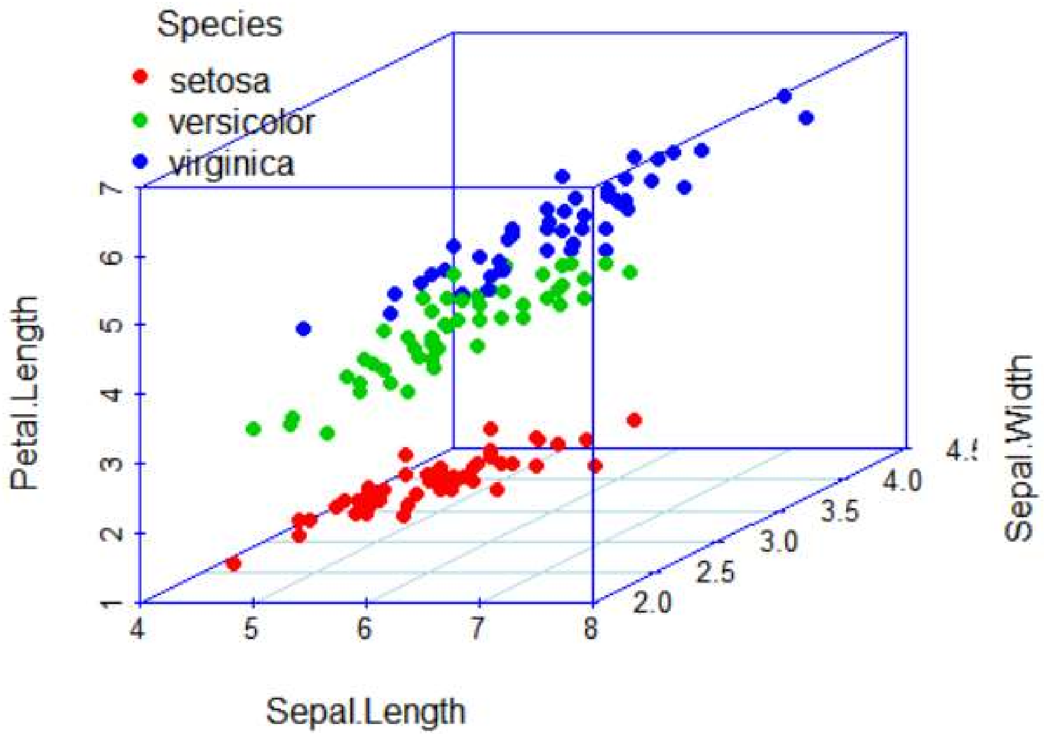
```
#(3) boxplot in ggplot2
library(ggplot2)
ggplot(ToothGrowth, aes(x=factor(dose), y=len)) +
  geom_boxplot(aes(fill=supp)) +
  xlab("dose") + ylab("length") +
  ggtitle("Analyzing ToothGrowth Data")
```



scatterplot3d(x, y, z, ...) {scatterplot3d}

Plots a three dimensional (3D) point cloud.

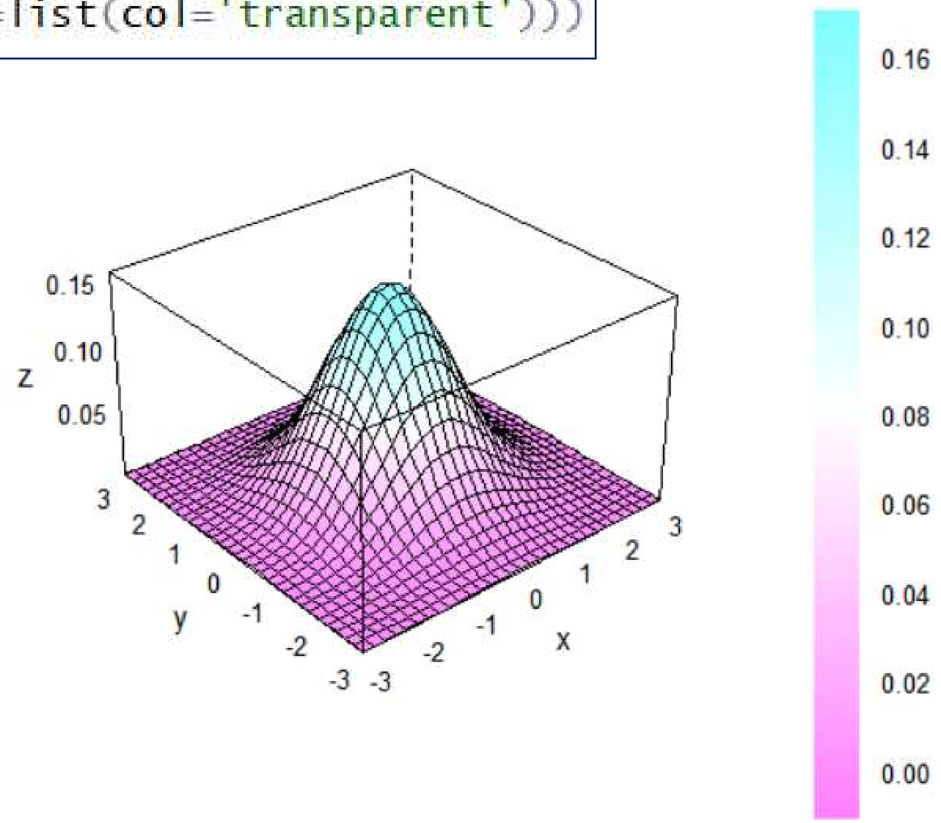
```
#(1)  3D Scatter Plot
library(scatterplot3d)
s3d <- scatterplot3d(iris[,1:3], color=c(2:4)[iris$Species],
       col.axis="blue", col.grid="lightblue", pch=16, cex.symbols=1)
legend(s3d$xyz.convert(2.5,3.1,8.8), legend=levels(iris$Species),
       col=2:4, pch=16, bty="n", title="Species")
```



wireframe(x, data, ...) {lattice}

Generic functions to draw 3d scatter plots and surfaces.

```
#(2) 3D Wireframe Plot
library(lattice)
x <- seq(-3, 3, .2); y=x
dat <- expand.grid(x,y)
dat$z <- dnorm(dat[,1])*dnorm(dat[,2])
names(dat) <- c('x','y','z')
wireframe(z ~ x*y, data=dat,  scales=list(arrows=FALSE),
          aspect=c(1,.6), drape=TRUE,
          par.settings=list(axis.line=list(col='transparent')))
```



8. Saving Plots

Since R runs on so many different operating systems, and supports so many different graphics formats, it's not surprising that there are a variety of ways of saving your plots.

bmp, **jpeg**, **png**(filename, width, height, pointsize, ...)

Graphics devices for BMP, JPEG and PNG format bitmap files.

pdf(file, width, height, pointsize, ...)

pdf starts the graphics device driver for producing PDF graphics.

- jpeg("test.jpg"); plot(x,y); dev.off()

  # After the 'jpeg("test.jpg")' command all graphs are redirected to the file "test.jpg" in JPEG format.
  The actual image data are not written to the file until the 'dev.off()' command is executed!

```
# Example: Saving Plots
png('SampleSavePlot.png',width=580,height=640)
x = rnorm(100, mean=0, sd=1)
hist(x, freq=F, col='cyan')
lines(density(x), col='red')
dev.off()
```

SamplesavePlot.png →