R for Bioinformatics

Introduction, Programming, Data Analysis and Visualization

Basic data analysis and plotting using R

Gang Chen
chengang@bgitechsolutions.com

November 16, 2013

Outline

Next

Data Analysis and Visualization

06-CA-VS-CFHR4	NM_006684	nonsynonymous SNV	c.510C>G	p.D170E	1	1.97E+08	1.97E+08	C>G	39
06-CA-VS-NBPF9		splice_site			1	1.45E+08	1.45E+08	G>C	190
06-CA-VS-HIVEP3	NM_001127714	nonsynonymous SNV	c.2681C>T	p.A894V	1	42047788	42047788	G>A	18
06-CA-VS-TTLL7	NM_024686	nonsynonymous SNV	c.1367G>A	p.R456Q	1	84385515	84385515	C>T	48
06-CA-VS-PRKACB	NM_001242862	nonsynonymous SNV	c.605G>A	p.G202D	1	84668367	84668367	G>A	58
06-CA-VS-SMG5	NM_015327	nonsynonymous SNV	c.2213C>A	p.P738H	1	1.56E+08	1.56E+08	G>T	55
06-CA-VS-CREG1	NM_003851	stopgain SNV	c.386T>A	p.L129X	1	1.68E+08	1.68E+08	A>T	57
06-CA-VS-CRB1	NM_201253	nonsynonymous SNV	c.952G>T	p.D318Y	1	1.97E+08	1.97E+08	G>T	66
06-CA-VS-LYST	NM_000081	nonsynonymous SNV	c.3509G>A	p.G1170E	1	2.36E+08	2.36E+08	C>T	53
06-CA-VS-CCDC147	NM_001008723	nonsynonymous SNV	c.1855C>T	p.R619W	10	1.06E+08	1.06E+08	C>T	85
06-CA-VS-DOCK1	NM_001380	nonsynonymous SNV	c.260C>T	p.P87L	10	1.29E+08	1.29E+08	C>T	53
06-CA-VS-VWCE	NM_152718	nonsynonymous SNV	c.2428A>G	p.T810A	11	61026587	61026587	T>C	31
06-CA-VS-CTTN	NM_001184740	nonsynonymous SNV	c.1381G>A	p.V461I	11	70279800	70279800	G>A	26
06-CA-VS-GRIA4	NM_000829	nonsynonymous SNV	c.2617G>A	p.V873I	11	1.06E+08	1.06E+08	G>A	44
06-CA-VS-MLL	NM_001197104	nonsynonymous SNV	c.11065A>T	p.I3689F	11	1.18E+08	1.18E+08	A>T	55
06-CA-VS-C11orf63	NM_024806	nonsynonymous SNV	c.1895A>C	p.K632T	11	1.23E+08	1.23E+08	A>C	56
06-CA-VS-OR8G2	NM_001007249	nonsynonymous SNV	c.339G>C	p.Q113H	11	1.24E+08	1.24E+08	G>C	83
06-CA-VS-CACNA1C	NM_001129837	nonsynonymous SNV	c.4559G>A	p.G1520E	12	2775917	2775917	G>A	46
06-CA-VS-TAS2R10	NM_023921	nonsynonymous SNV	c.761A>G	p.E254G	12	10978108	10978108	T>C	58
06-CA-VS-ABCC9	NM_005691	nonsynonymous SNV	c.1886A>T	p.E629V	12	22040785	22040785	T>A	78
06-CA-VS-ITPR2	NM_002223	nonsynonymous SNV	c.5698A>T	p.N1900Y	12	26639150	26639150	T>A	67
06-CA-VS-C12orf35	NM_018169	nonsynonymous SNV	c.905C>T	p.S302F	12	32134794	32134794	C>T	54
06-CA-VS-KRT83	NM_002282	nonsynonymous SNV	c.673C>T	p.L225F	12	52711542	52711542	G>A	29
06-CA-VS-CRY1	NM_004075	nonsynonymous SNV	c.815T>C	p.L272P	12	1.07E+08	1.07E+08	A>G	50
06-CA-VS-RPH3A	NM_014954	nonsynonymous SNV	c.1378G>A	p.V460M	12	1.13E+08	1.13E+08	G>A	31
06-CA-VS-NAA16	NM_024561	nonsynonymous SNV	c.1426A>G	p.M476V	13	41936182	41936182	A>G	58
06-CA-VS-SLC24A4	NM_153646	nonsynonymous SNV	c.695T>G	p.L232R	14	92913726	92913726	T>G	49
06-CA-VS-EIF5	NM_183004	nonsynonymous SNV	c.317A>C	p.E106A	14	1.04E+08	1.04E+08	A>C	76
06-CA-VS-TMC5	NM_024780	nonsynonymous SNV	c.899G>T	p.S300I	16	19477555	19477555	G>T	84
06-CA-VS-GPR139	NM_001002911	nonsynonymous SNV	c.802G>A	p.D268N	16	20043317	20043317	C>T	105
06-CA-VS-SPATA22	NM_001170696	nonsynonymous SNV	c.710A>T	p.Y237F	17	3346529	3346529	T>A	79
06-CA-VS-CAMTA2	NM_001171166	nonsynonymous SNV	c.2134C>T	p.R712W	17	4876953	4876953	G>A	34
06-CA-VS-TP53	NM_001126115	nonsynonymous SNV	c.263A>G	p.Y88C	17	7578190	7578190	T>C	7

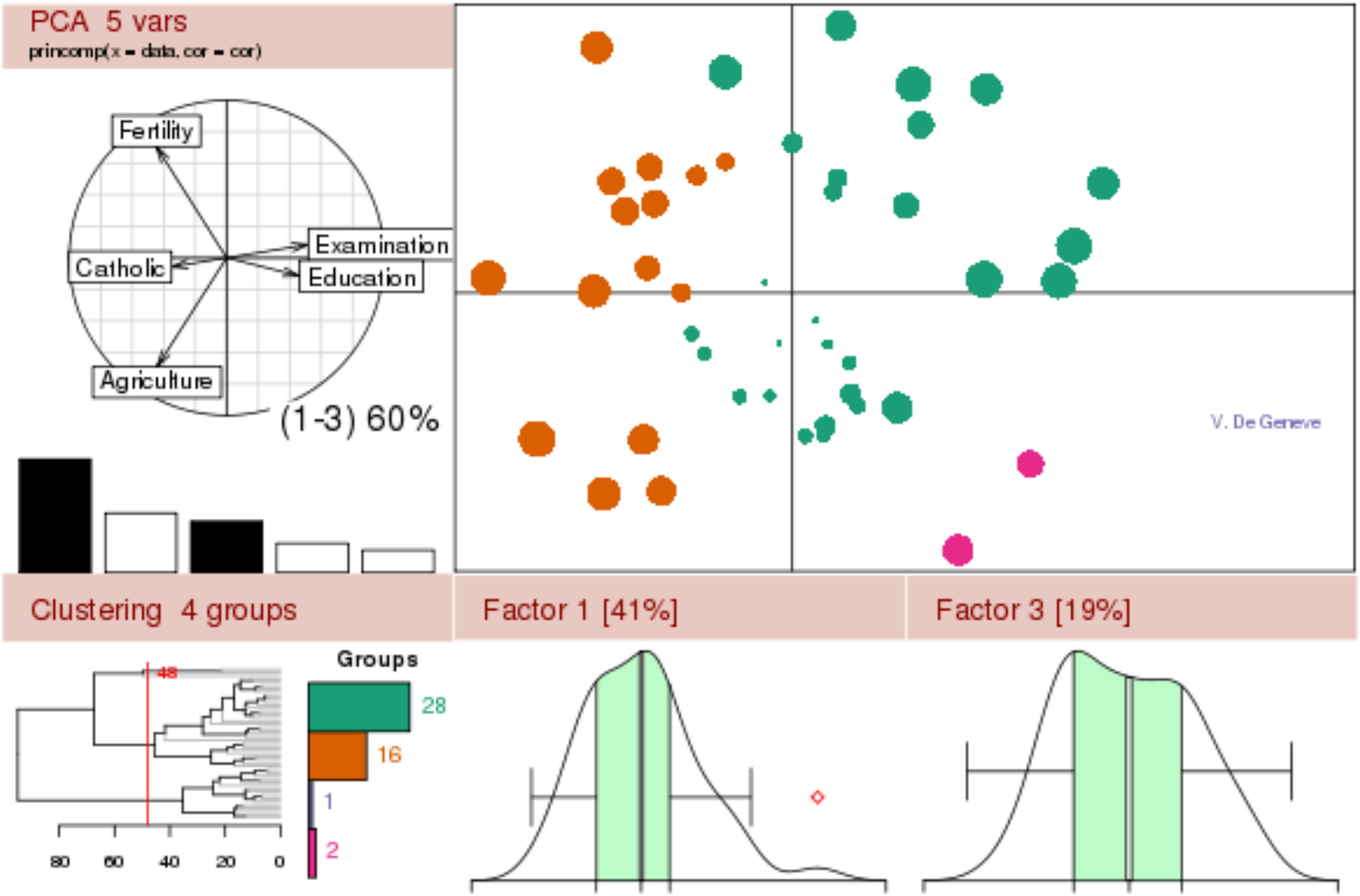
Data Visualization

What is Data Visualization

Data Visualization

What is Data Visualization

Statistical Plotting in R



Graphic Systems in R

Graphic Systems

- Traditional Plotting
- grid and lattice
- ggplot2
- Graphic Systems for specific purposes: rgl,ggbio, scatterplot3d

Next

Functions

- plot
- boxplot, barplot, hist, ...
- par
- output: png, jpg, pdf, ...

mtcars

```
data(mtcars)
mtcars
```

```
##                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4          21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag      21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710         22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive     21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout  18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant            18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360         14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D          24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230           22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280           19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C          17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE         16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL         17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC        15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood 10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4  8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial  14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128           32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic        30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla     33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona      21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger   15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin        15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28         13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird   19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
```

plot

plot function

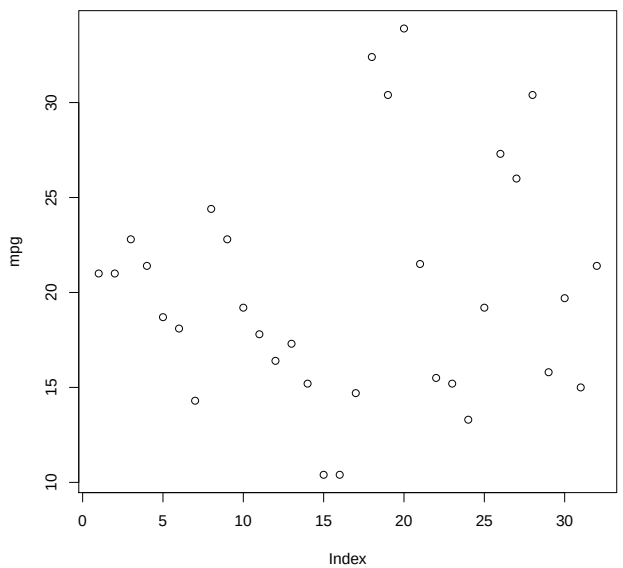
- plot() is the main graphing function
- Automatically produces simple plots for vectors, functions or data frames (OOP, S3)
- Many useful customization options...

plot

```
help(plot)
```

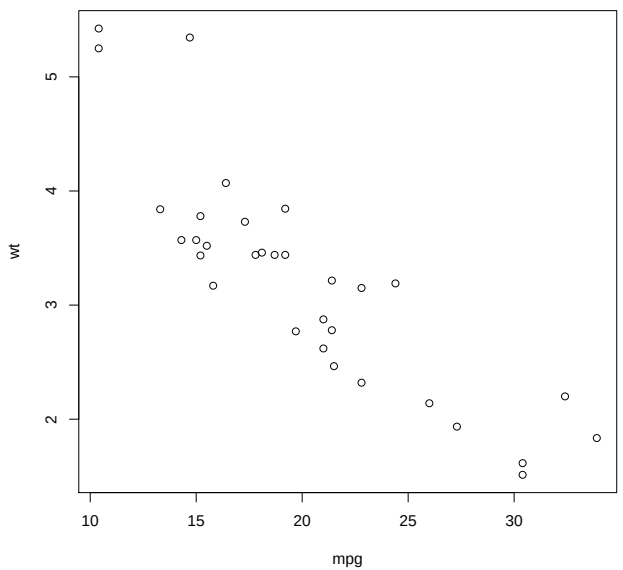
plot

```
plot(mpg)
```



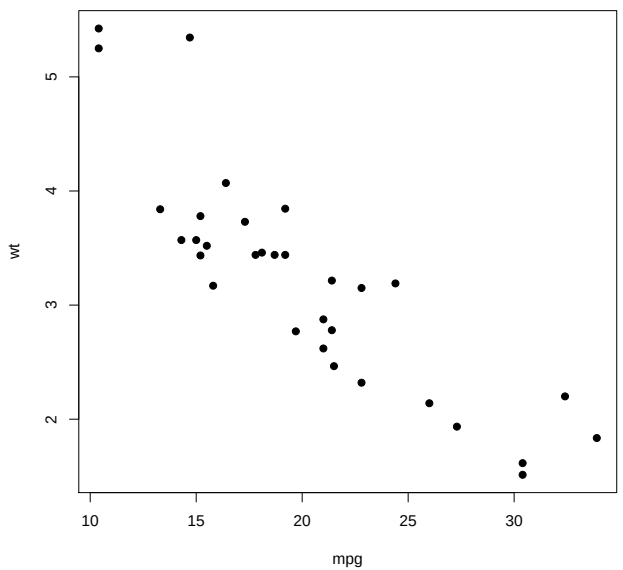
plot

```
plot(mpg, wt)
```



plot

```
plot(mpg, wt, pch = 19)
```

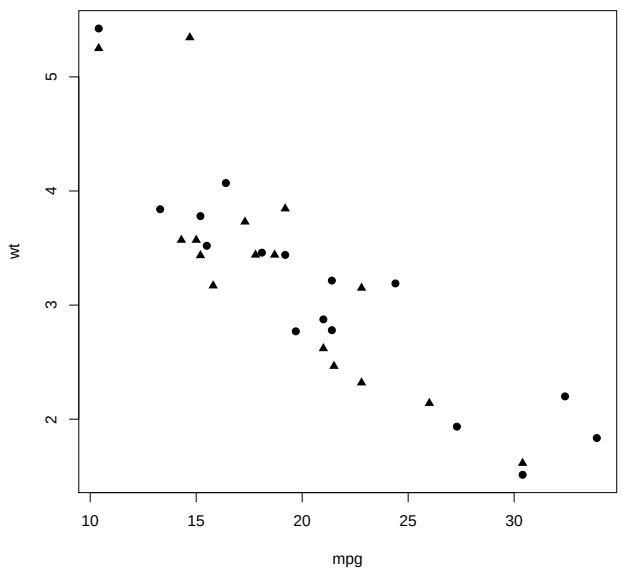


par and points

```
help(par)
help(points)
```

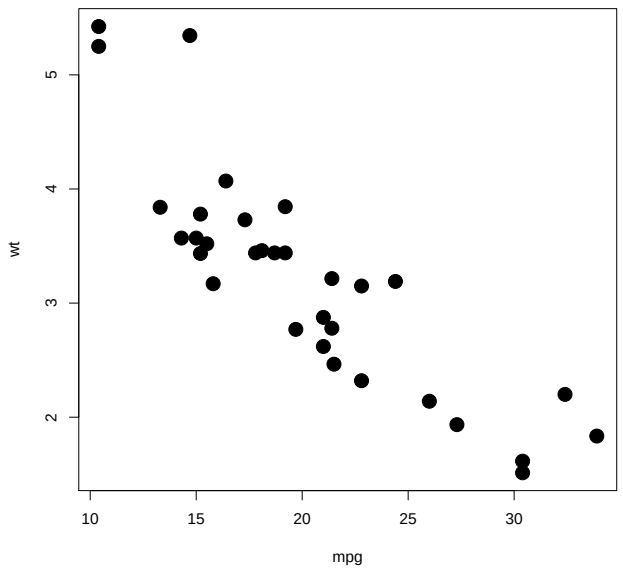
plot

```
plot(mpg, wt, pch = c(17, 19))
```



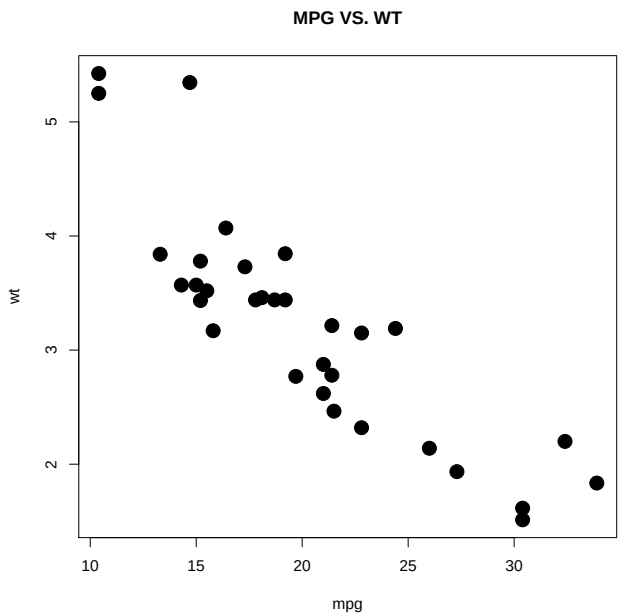
plot

```
plot(mpg, wt, pch = 19, cex = 2)
```



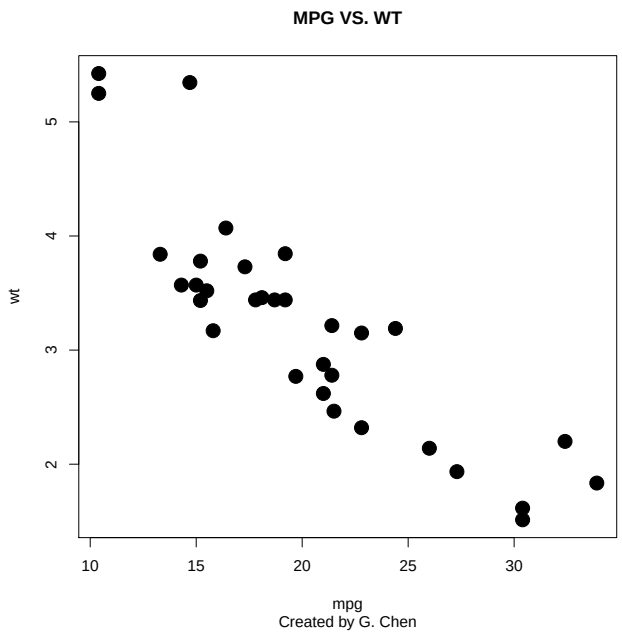
plot

```
plot(mpg, wt, pch = 19, cex = 2, main = "MPG VS. WT")
```



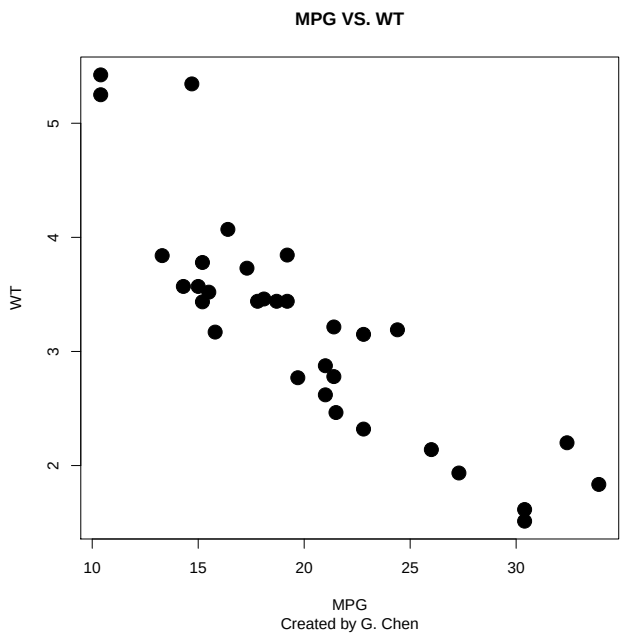
plot

```
plot(mpg, wt, pch = 19, cex = 2, main = "MPG VS. WT", sub = "Cre
```



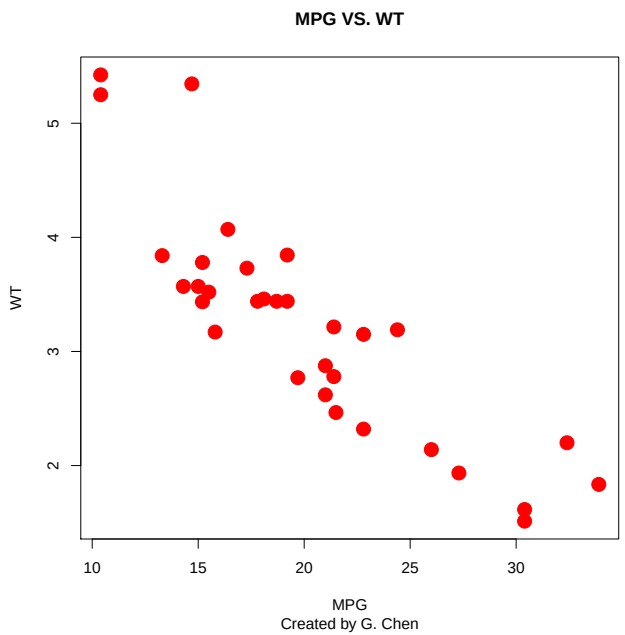
plot

```
plot(mpg, wt, pch = 19, cex = 2, main = "MPG VS. WT", sub = "Created b
    xlab = "MPG", ylab = "WT")
```



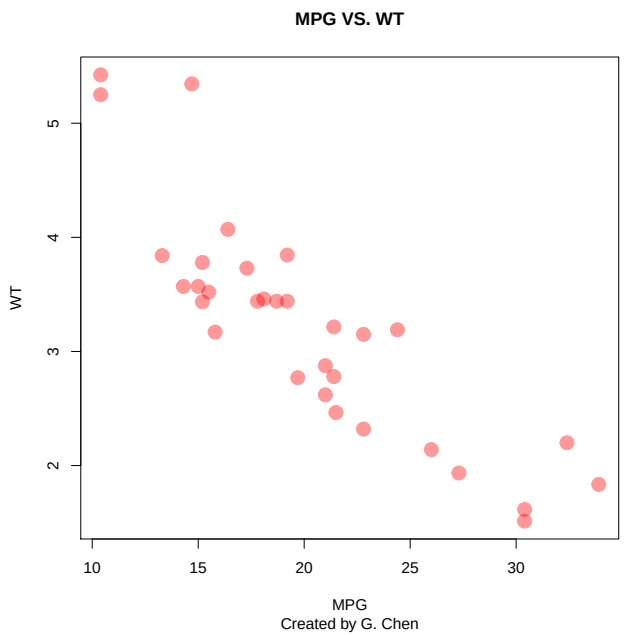
plot

```
plot(mpg, wt, pch = 19, cex = 2, main = "MPG VS. WT", sub = "Created b
    xlab = "MPG", ylab = "WT", col = "red")
```



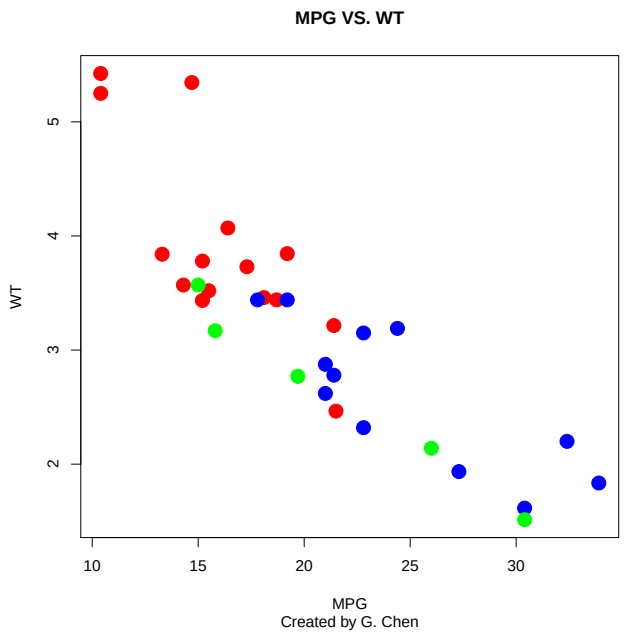
plot

```
plot(mpg, wt, pch = 19, cex = 2, main = "MPG VS. WT", sub = "Created b
    xlab = "MPG", ylab = "WT", col = rgb(1, 0, 0, 0.4))
```



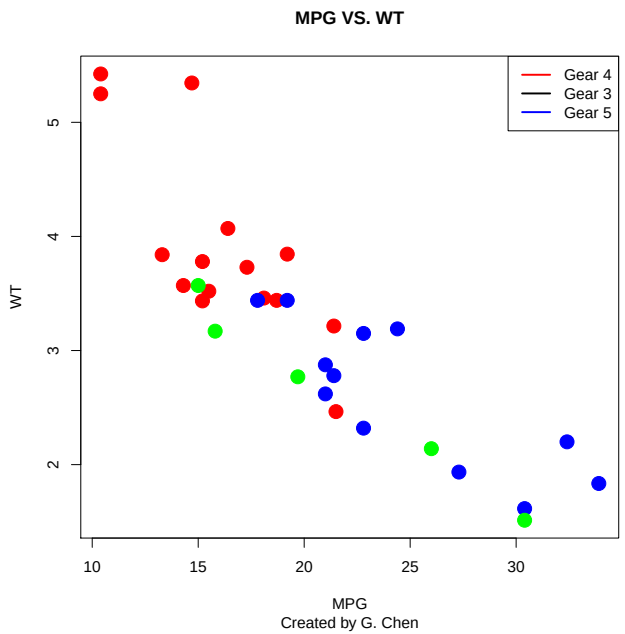
plot

```
plot(mpg, wt, pch = 19, cex = 2, main = "MPG VS. WT", sub = "Created b
    xlab = "MPG", ylab = "WT", col = c("red", "blue", "green")
[as.factor(gear)])
```



plot

```
plot(mpg, wt, pch = 19, cex = 2, main = "MPG VS. WT", sub = "Created by G. Chen",
    xlab = "MPG", ylab = "WT", col = c("red", "blue", "green")[as.factor(gear)])
legend("topright", legend = paste("Gear", unique(gear)), lwd = 2, col = c("red",
    "black", "blue", "green"))
```

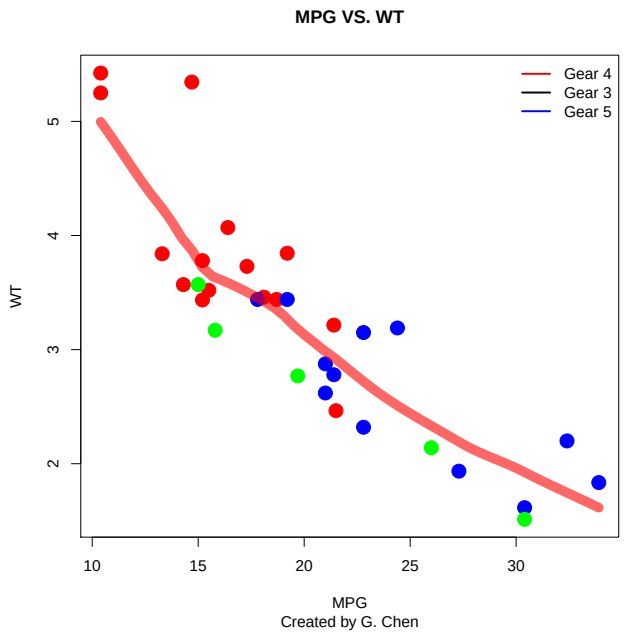


legend

```
help(legend)
```

plot

```
plot(mpg, wt, pch = 19, cex = 2, main = "MPG VS. WT", sub = "Created by G. Chen",
    xlab = "MPG", ylab = "WT", col = c("red", "blue", "green")[as.factor(gear)])
legend("topright", legend = paste("Gear", unique(gear)), lwd = 2, col = c("red",
    "black", "blue", "green"), bty = "n")
lines(loess.smooth(mtcars$mpg, mtcars$wt), col = rgb(1, 0, 0, 0.6), lwd = 10)
```

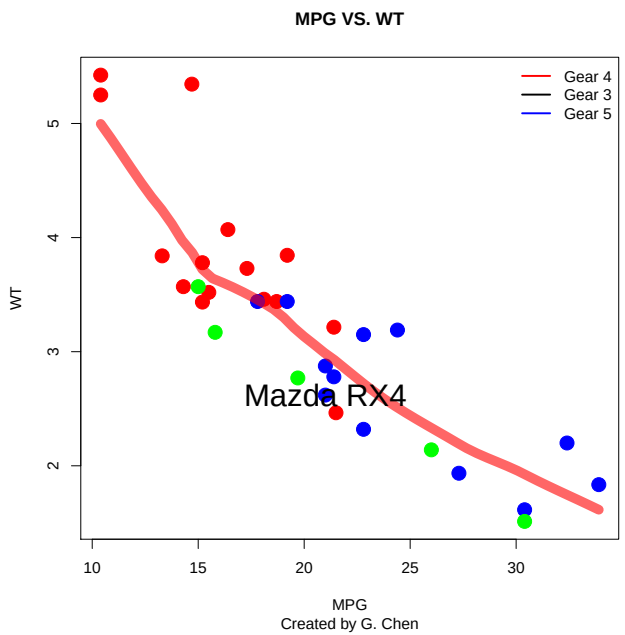


lines

```
help(lines)
help(abline)
```

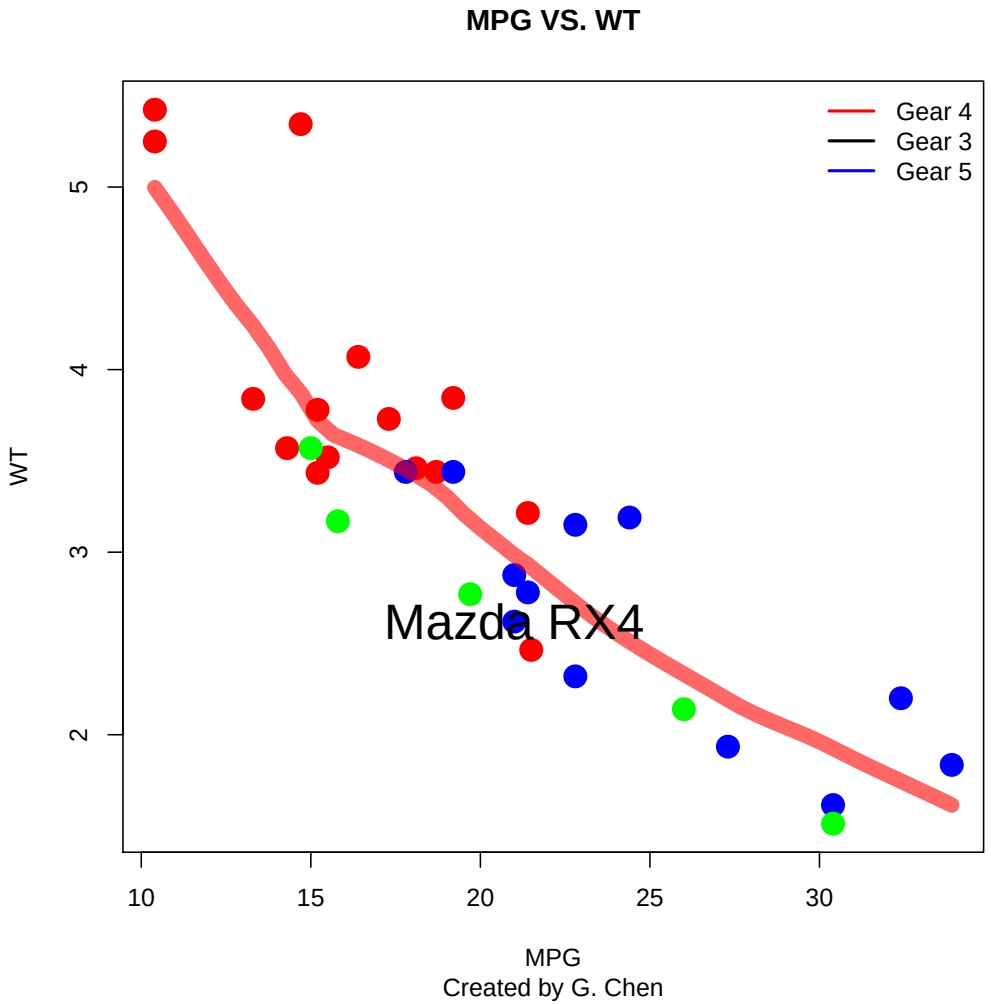
plot

```
plot(mpg, wt, pch = 19, cex = 2, main = "MPG VS. WT", sub = "Created by G. Chen",
    xlab = "MPG", ylab = "WT", col = c("red", "blue", "green")[as.factor(gear)])
legend("topright", legend = paste("Gear", unique(gear)), lwd = 2, col = c("red",
    "black", "blue", "green"), bty = "n")
lines(loess.smooth(mtcars$mpg, mtcars$wt), col = rgb(1, 0, 0, 0.6), lwd = 10)
text(mpg[1], wt[1], rownames(mtcars)[1], cex = 2)
```



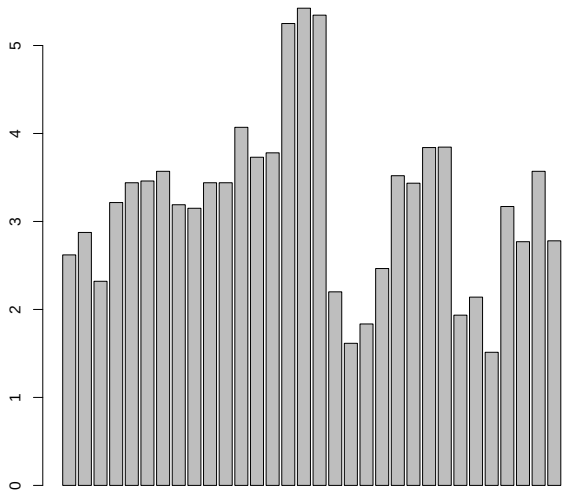
text

```
help(text)
```


MPG VS. WT
Gear 4
Gear 3
Gear 5
Mazda RX4
WT
MPG
Created by G. Chen
2
3
4
5
10
15
20
25
30

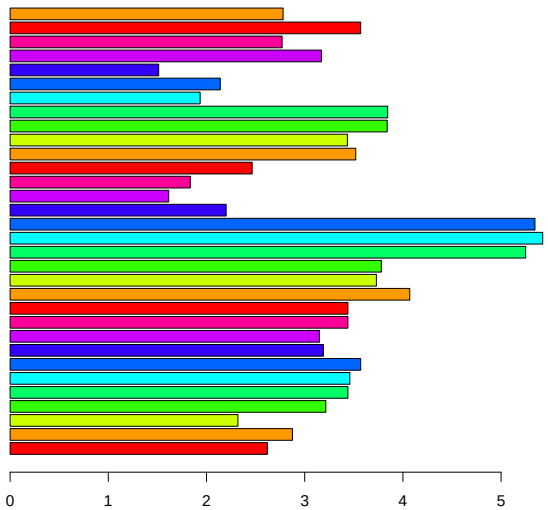
barplot

```
barplot(wt)
```



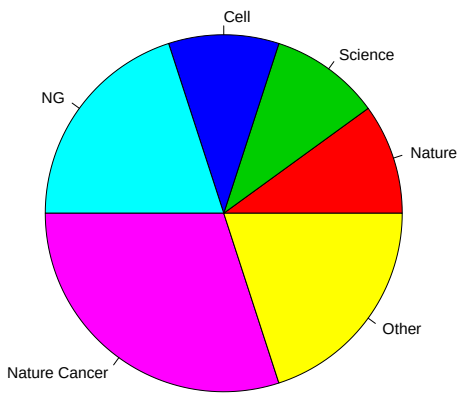
barplot

```
barplot(wt, width = 2, horiz = T, col = rainbow(10))
```



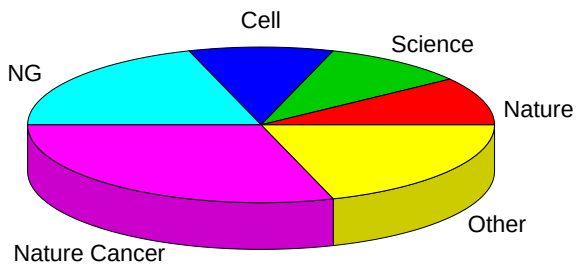
pie

```
pie(c(10, 10, 10, 20, 30, 20), c("Nature", "Science",
"Cell", "NG", "Nature Cancer", "Other"), col = 2:7)
```



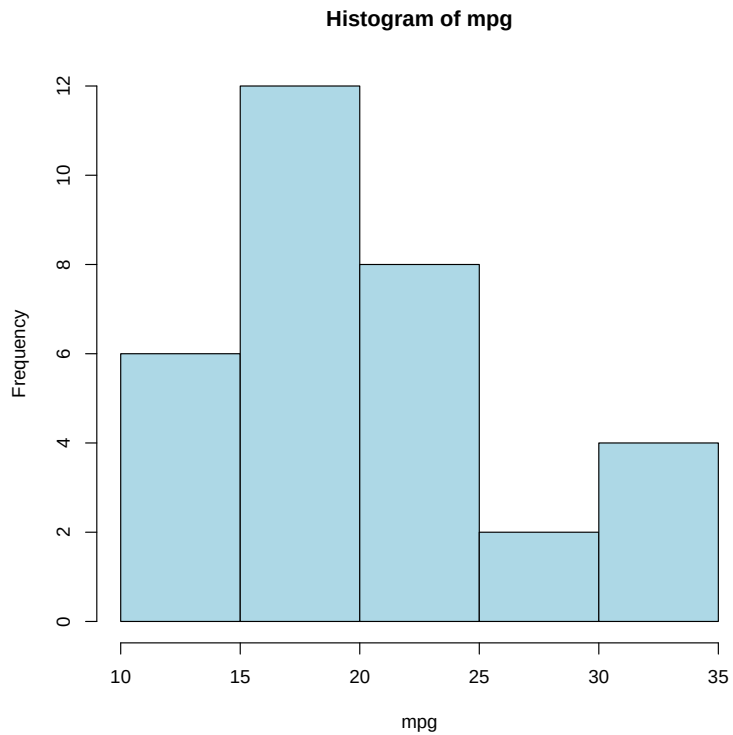
pie

```
library(plotrix)
pie3D(c(10, 10, 10, 20, 30, 20), labels = c("Nature",
"Science", "Cell", "NG", "Nature Cancer", "Other"), col
= 2:7)
```



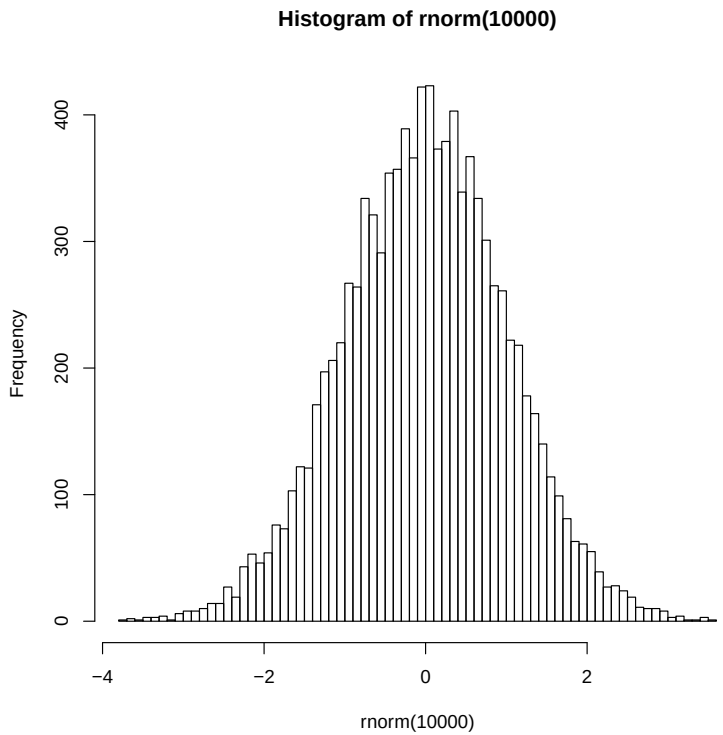
hist

```
hist(mpg, col = "lightblue")
```



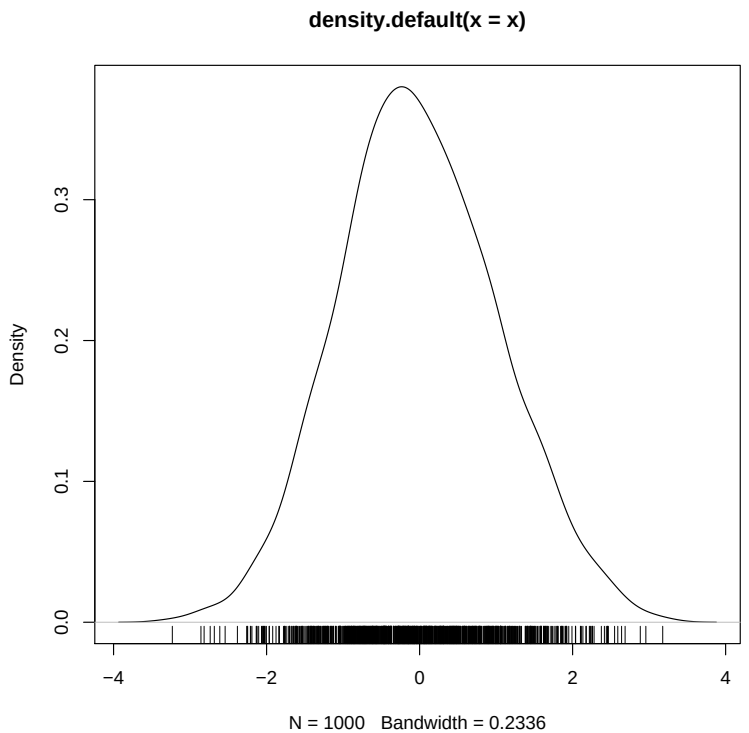
hist

```
hist(rnorm(10000), breaks = 100)
```



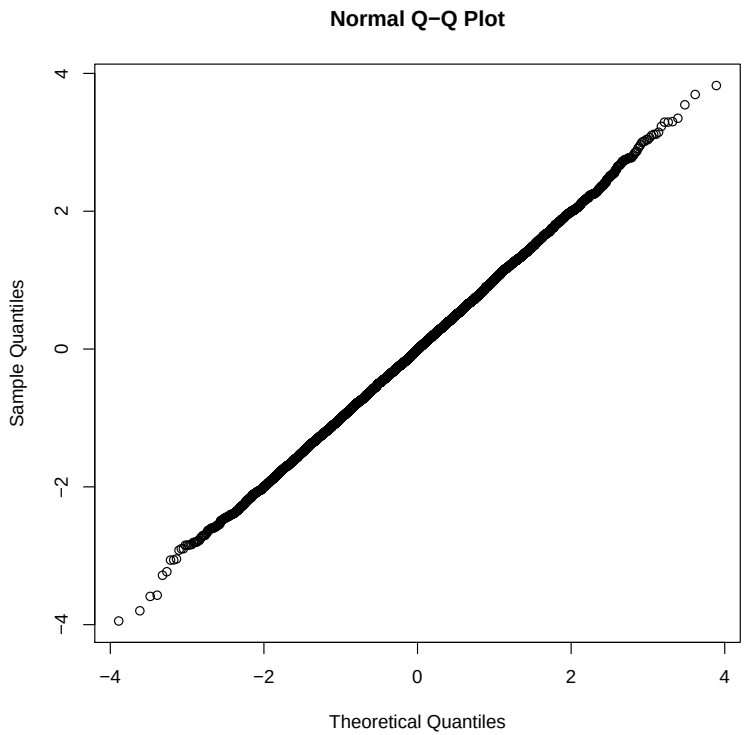
density + rug

```
x = rnorm(1000)
plot(density(x))
rug(x)
```



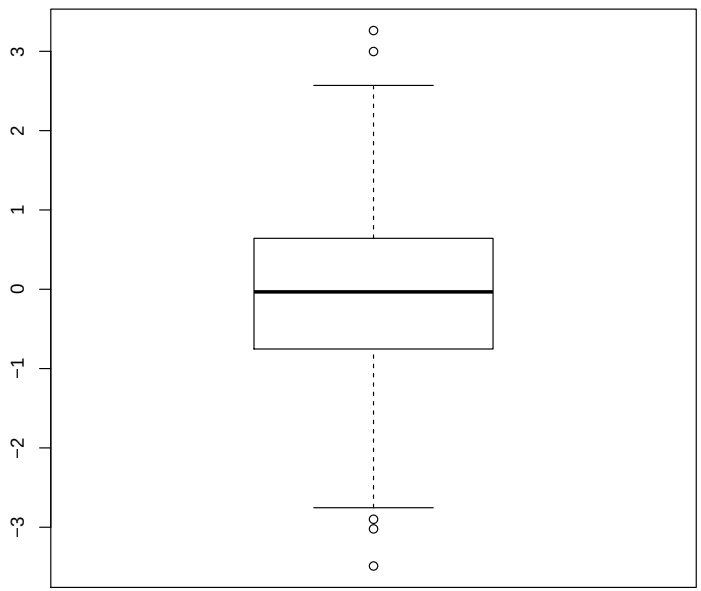
Q-Q plot

```
qqnorm(rnorm(10000))
```



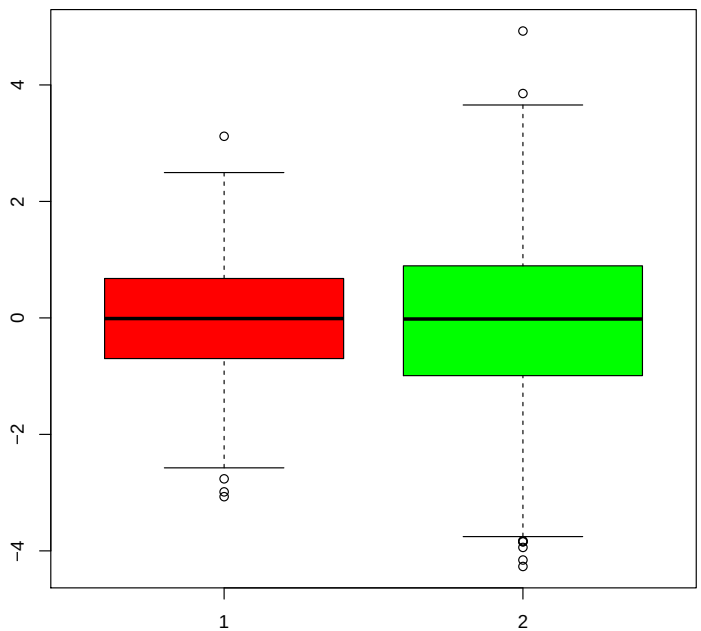
boxplot

```
boxplot(rnorm(1000))
```



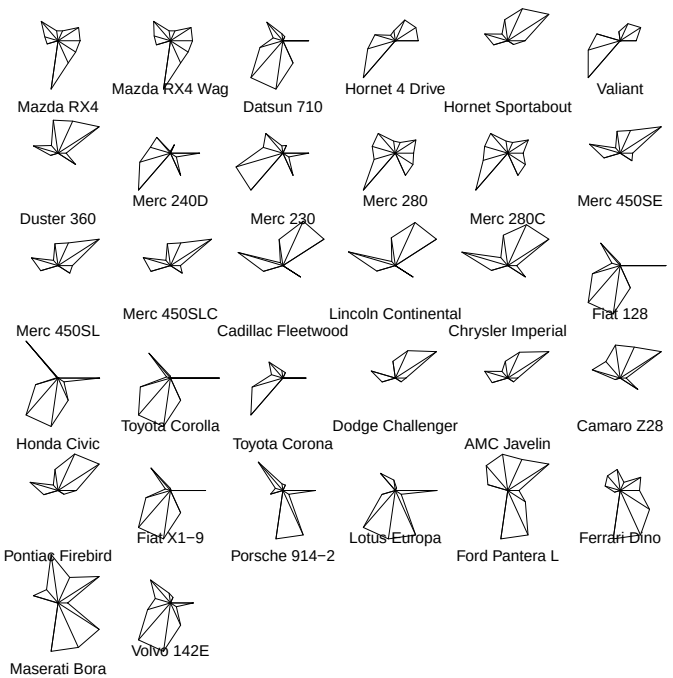
boxplot

```
boxplot(cbind(rnorm(1000), rnorm(1000) + rnorm(1000)), col =
```



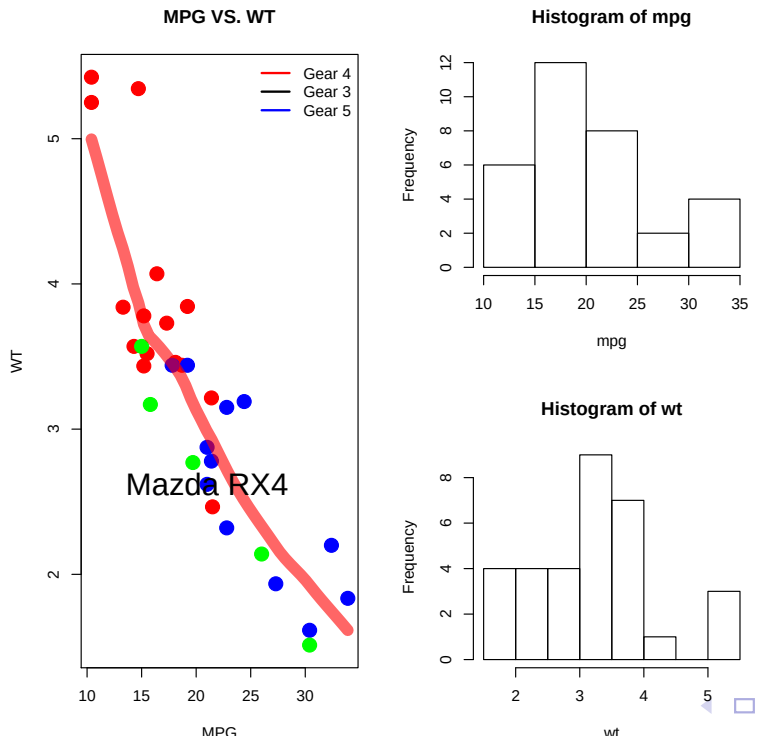
stars

```
stars(mtcars)
```



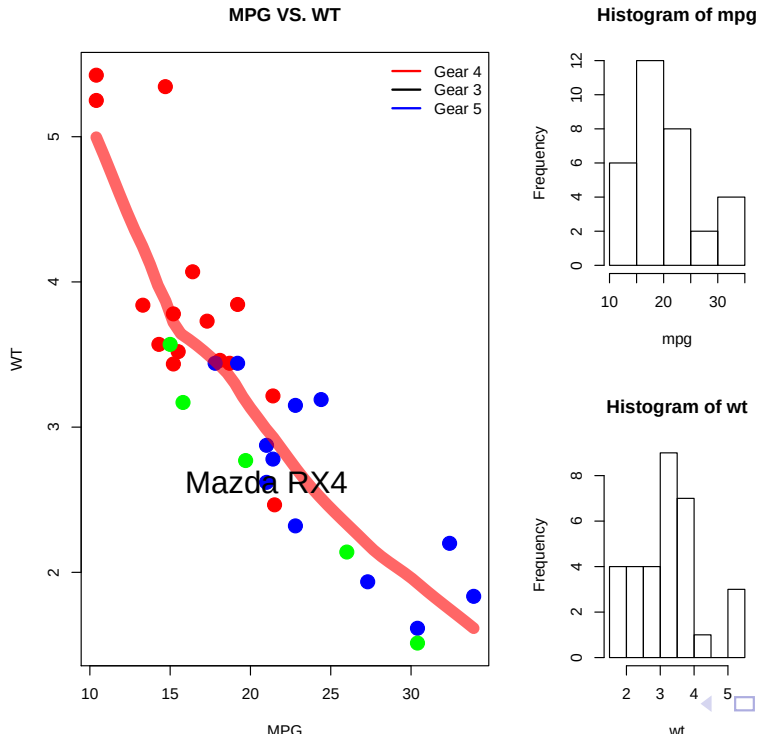
layout

```
layout(matrix(c(1, 1, 2, 3), ncol = 2))
plot(mpg, wt, pch = 19, cex = 2, main = "MPG VS. WT", sub = "Created by G. Chen",
    xlab = "MPG", ylab = "WT", col = c("red", "blue", "green")[as.factor(gear)])
legend("topright", legend = paste("Gear", unique(gear)), lwd = 2, col = c("red",
    "black", "blue", "green"), bty = "n")
lines(loess.smooth(mtcars$mpg, mtcars$wt), col = rgb(1, 0, 0, 0.6), lwd = 10)
text(mpg[1], wt[1], rownames(mtcars)[1], cex = 2)
hist(mpg)
hist(wt)
```



layout

```
layout(matrix(c(1, 1, 2, 3), ncol = 2), widths = c(2, 1))
plot(mpg, wt, pch = 19, cex = 2, main = "MPG VS. WT", sub = "Created by G. Chen",
    xlab = "MPG", ylab = "WT", col = c("red", "blue", "green")[as.factor(gear)])
legend("topright", legend = paste("Gear", unique(gear)), lwd = 2, col = c("red",
    "black", "blue", "green"), bty = "n")
lines(loess.smooth(mtcars$mpg, mtcars$wt), col = rgb(1, 0, 0, 0.6), lwd = 10)
text(mpg[1], wt[1], rownames(mtcars)[1], cex = 2)
hist(mpg)
hist(wt)
```



Graphic Devices

Graphic Devices

- Everything you draw in R must be drawn on a graphic device
- Different devices save graphical input in different ways
- raster device vs. vecter device

x11, windows(), quartz()

vecter device: pdf, postscript

raster device: png, bmp, jpeg, tiff

Next

ggplot2

ggplot2

ggplot2 is an R implementation of the grammar of grahics.

Installation

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

Website

http://had.co.nz/ggplot2

```
library(ggplot2)

##
## Attaching package: 'ggplot2'
##
## The following object is masked from 'mtcars':
##
##     mpg
```

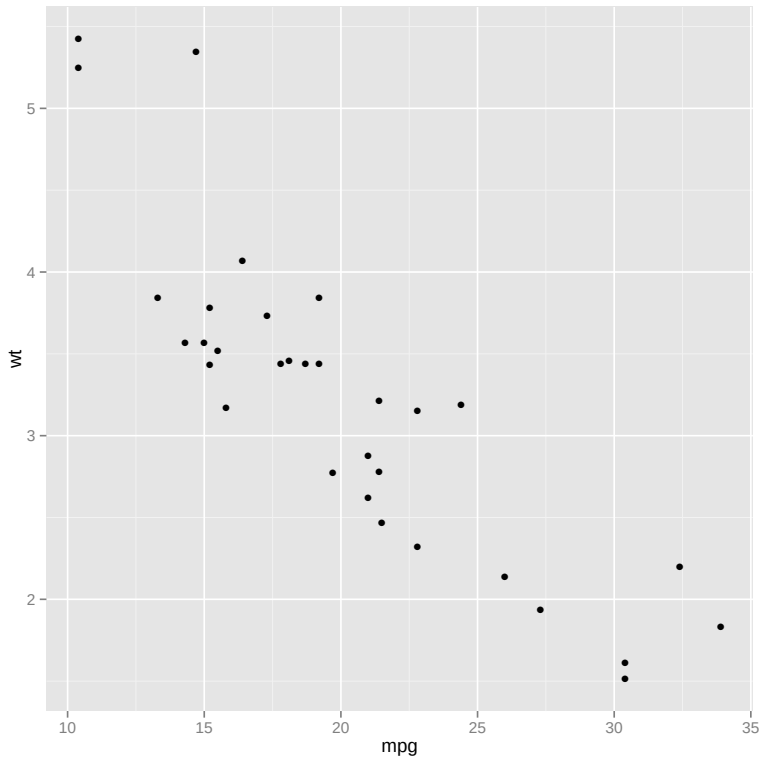
qplot

Wraps up all the details of ggplot with a familiar syntax borrowed from plot.
Additional features:

- Automatically scales data
- Can produce any type of plot
- Facetting and margins
- Creates objects that can be saved and modified

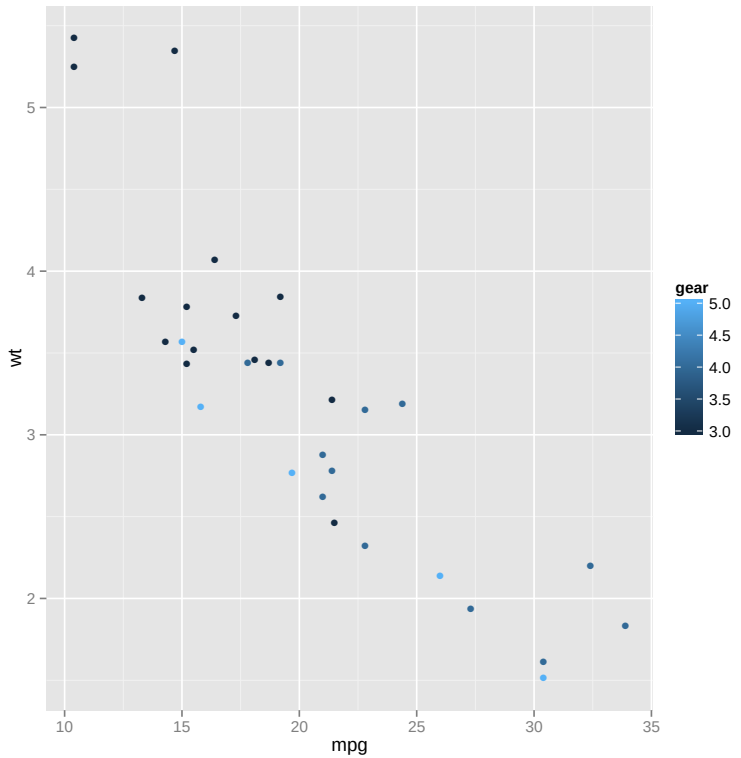
qplot

```
library(ggplot2)
qplot(mpg, wt, data = mtcars)
```



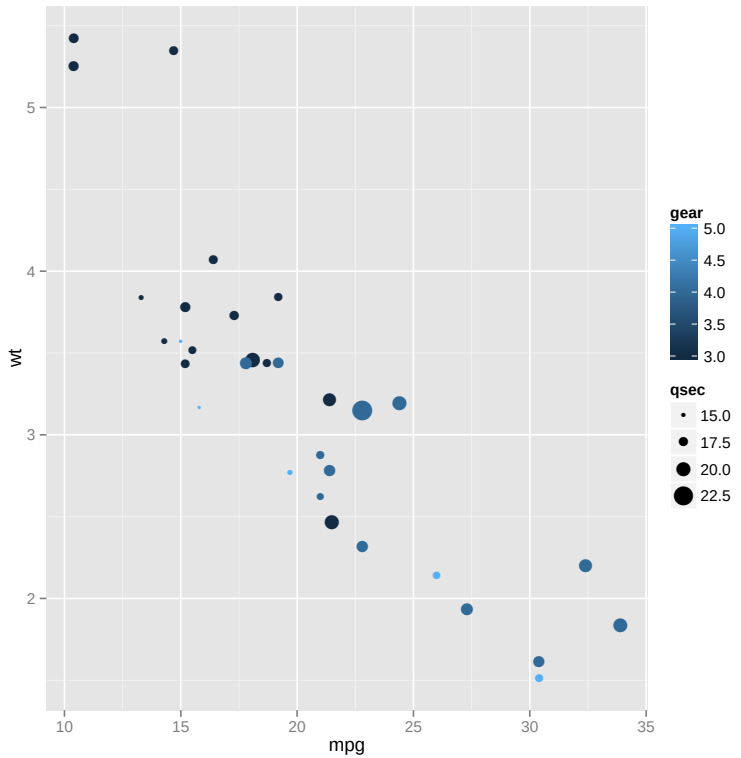
qplot

```
qplot(mpg, wt, data = mtcars, color = gear)
```



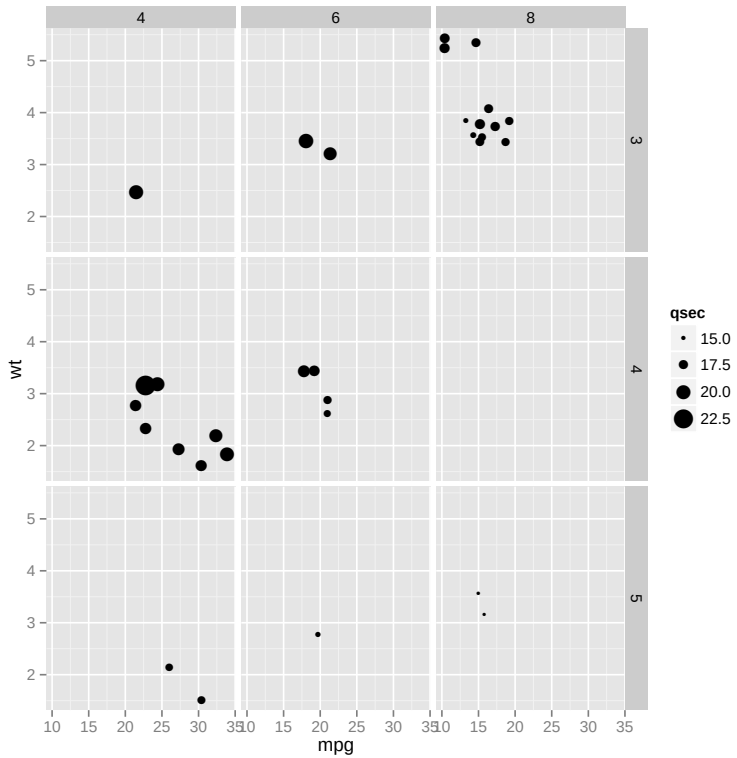
qplot

```
qplot(mpg, wt, data = mtcars, color = gear, size = qsec)
```



qplot

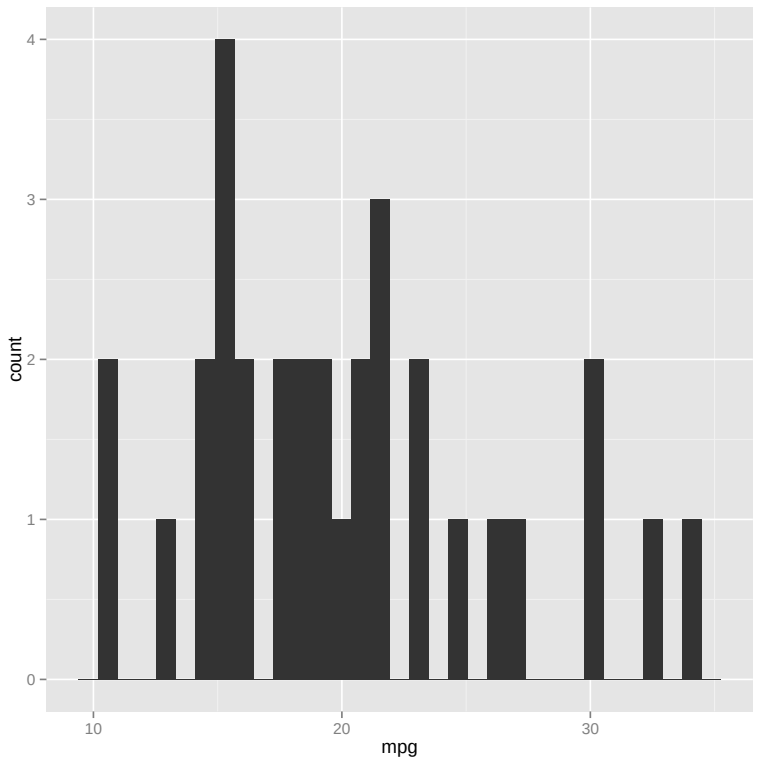
```
qplot(mpg, wt, data = mtcars, facets = gear ~ cyl, size = qsec)
```



qplot

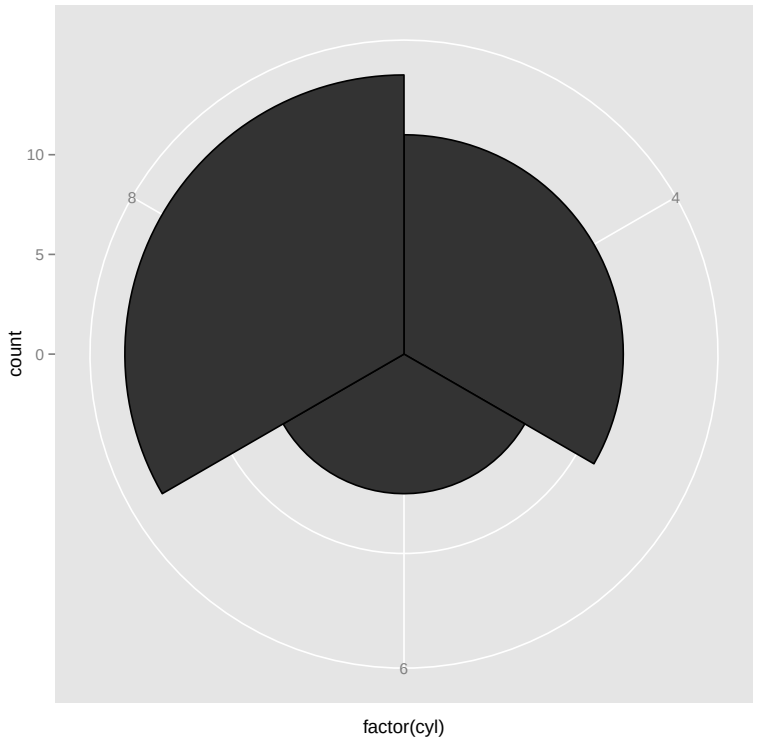
```
qplot(mpg, data = mtcars, geom = "bar")

## stat_bin: binwidth defaulted to range/30. Use
'binwidth = x' to adjust this.
```



qplot

```
cxc <- ggplot(mtcars, aes(x = factor(cyl))) + geom_bar(width
cxc + coord_polar()
```



Reference

http://ygc.name/stats/ggplot2.html

Next

rgl

rgl package

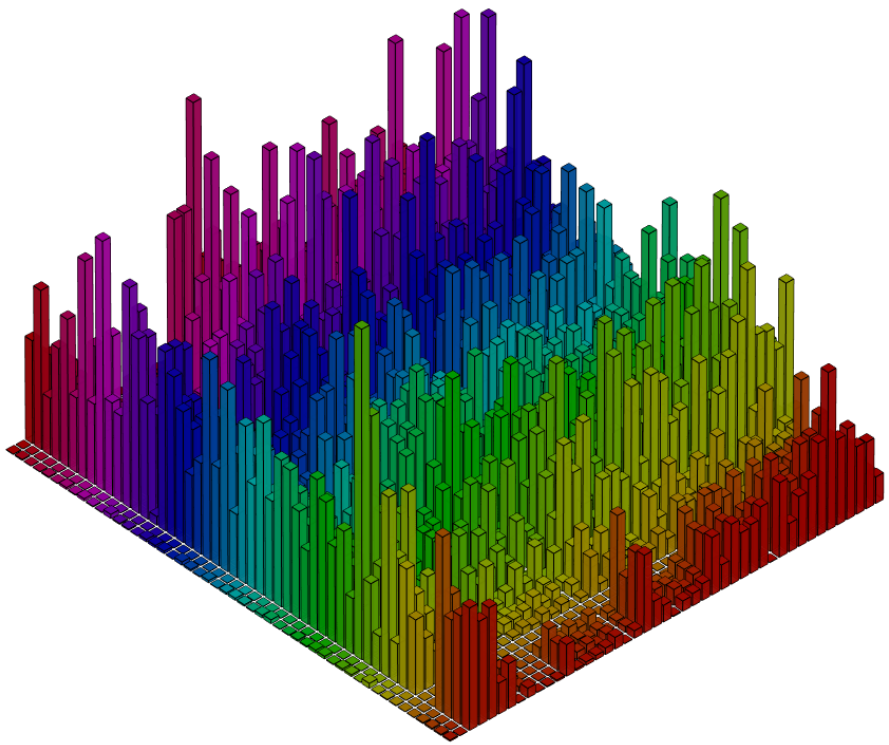
The core of RGL is a shared library that acts as an interface between R and OpenGL. In order to provide convenient access to OpenGL–features, a set of R–functions which act as an API (Application Programming Interface) was written.

Shape Fcuntions

- rgl.points
- rgl.lines
- rgl.triangles
- rgl.surface
- rgl.quads
- rgl.spheres

Reference: RGL: A R-library for 3D visualization with OpenGL (RGL_REF.pdf)

3D Barplot



bgiR

```
library(devtools)
install_github("bgiR", "gangchen")
library(bgiR)
bgiR()
```

Next

Assignments. 1

- **Data:** A subset of microarray dataset (exprData.tsv on eLearning)
  - Each column is a sample
  - Each row is a probeset
- **To Do:**
  - Read the file into R (Tip: read.table());
  - Calculate mean and median of each column and each row (Tip: apply());
  - *Calculate correlation between each pair of samples (Tip: apply());
  - Draw boxplot of each sample, and output the plot to a pdf file (Tip: boxplot);
  - * Read help information of heatmap function, plot a heatmap for this data (Tip: help(heatmap));
  - Try your best to optimize the script and plots (Tip: ggplot2).