iCMS 2015
INTERNATIONAL CONFERENCE ON COMPUTING, MATHEMATICS AND STATISTICS
Bridging Research Endeavour in Computer and Mathematical Sciences
For more information, please visit http://www.icms2015.org
Organized by : FACULTY of COMPUTER & MATHEMATICAL SCIENCES UiTM KEDAH
Jointly organized by : RESEARCH & INDUSTRIAL LINKAGES
4th - 5th November 2015
Langkawi Island, MALAYSIA
UNIVERSITI TEKNOLOGI MARA CAWANGAN KEDAH

PRE-CONFERENCE WORKSHOP "INTRODUCTION TO R AND DATA VISUALIZATION"

Ciprian Alexandru

R-omania Team | www.r-project.ro

Presentation

- The R platform provides a powerful and comprehensive platform for visualizing data, understanding and evaluating statistical models, and effectively communicating research results to both technical and nontechnical audiences. This 2 days workshop will provide practical review of R's major graphing capabilities; including base functionality and new capabilities provided by the lattice and ggplot2 packages.

Date & Location

- Date: 2 - 3 November 2015
- Time: 9 am - 5 pm
- Venue: Melur 1, Langkawi Lagoon, Langkawi Island, MALAYSIA

Who should attend?

- R is widely used within the academia especially in the fields of computational biology, applied science, quantitative finance and business intelligence. R is capable of solving challenging problems and among the strengths of R are its powerful built-in tools for inferential statistics, its compact modeling syntax, and its data visualization capabilities. In addition, R's open source nature and its extensibility via add-on "packages" has allowed it to keep up with the leading edge in academic research. This workshop on R and Data Visualization is suitable and relevant for:
- Lecturers, Researchers, Engineers, Students, Industry Professionals and Scientists of any discipline who wish to explore R. Prior experience with R is not required. Interested to join??? Please register here.

Speakers

Antoniade-Ciprian Alexandru is an Associate Professor at the Ecological University of Bucharest and the dean of the Faculty of Economics. He is also attached with the National Institute of Statistics, Bucharest as an expert trainer in data analyst using R environment. Dr Alexandru is one of the six members of the R-omania team, a team that promotes R projects for statistical computing by providing a free and open source software environment for data analysis and graphics. The team acts as a user community for development of R projects among individuals, institutions, commercial entities and non-profit organizations. Dr Alexandru participated in various research projects, workshops, and, national and international conferences. His research works were published in various international databases. Currently, he is working on a project that implements the use of R as a tool for analyzing the evolution of indices on the stock market.

Nicoleta Caragea is an Associate Professor at the Faculty of Economics, Ecological University of Bucharest and a senior expert at the National Institute of Statistics. Her teaching activity is focused mainly in the field of statistics, through courses and seminars and master degree programs (statistics, economic statistics, social statistics, economic and financial analysis). Dr Caragea participated as a national expert in various projects, workshops and conferences organized by EUROSTAT, OECD, WHO, World Bank and UNICEF-UIS. She is one of the other six members of the R-omania team, a team that promotes R projects for statistical computing by providing a free and open source software environment for data analysis and graphics. She also acted as a consultant in projects in Europe. Her latest work was as a technical assistance to a consultancy work in Turkey.

Course Outline 1/2

- Introduction to R Statistical Software
  - The beginning of R
  - R - Introducing the R Console
  - R - Installation, Packages, CRAN, Components
  - Graphical User Interfaces: R Console, R Studio, R Commander, R resources and online community
- Databases
  - Data manipulation
  - Queries
  - Using SQL within R
  - Data aggregation
  - Matching

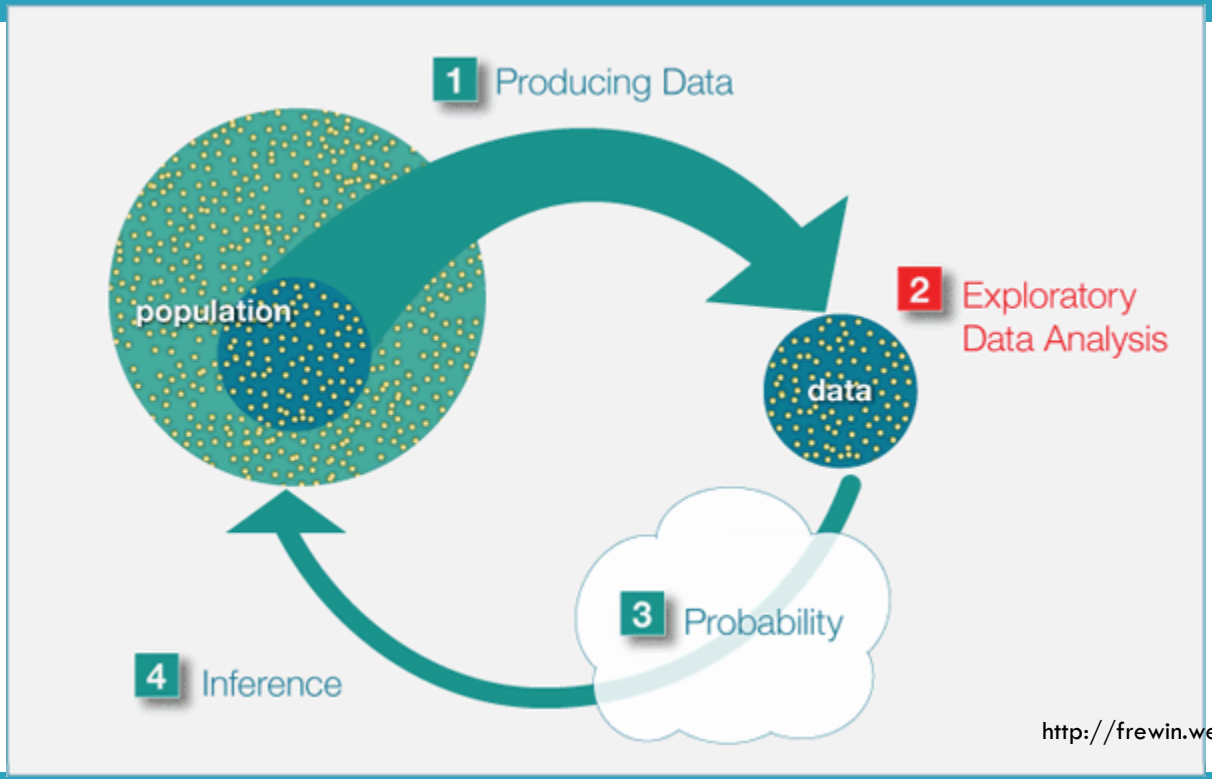
Course Outline 2/2

- Data Visualization & Graphics Environments
  - Base graphics (Scatterplot, Box-and-whiskers plot, Histogram)
  - Lattice
  - ggplot2
  - Interactive graphics in R
  - Reproducibility
- Regression Analysis with R
  - Linear regression models
  - OLS-ordinary least squares method for estimating the unknown parameters in a linear regression model
  - Interpreting the regression coefficients
  - Extensions to generalized linear models. Logistic regression
  - Parameter estimates – maximum likelihood method
  - Definition of the odds and odds ratio
  - Evaluating goodness of fit

Data Visualization & Graphics Environments

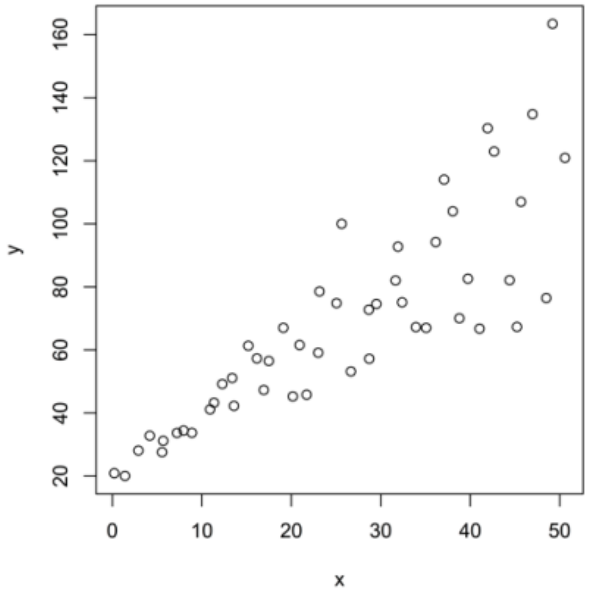
- Base graphics (Scatterplot, Box-and-whiskers plot, Histogram)
- Lattice
- ggplot2
- Interactive graphics in R
- Reproducibility

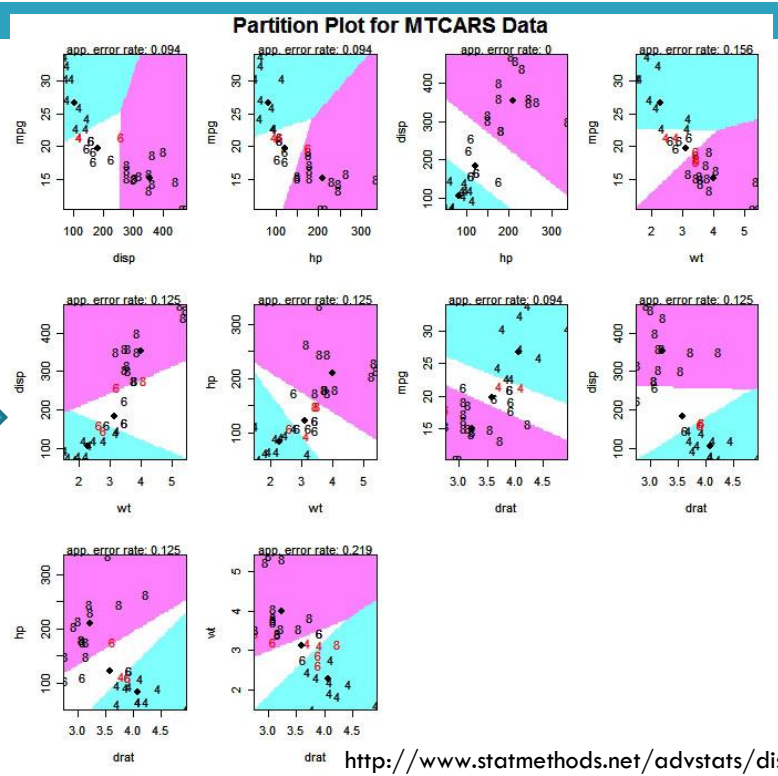
Before we start



Source:
http://frewin.weebly.com/ap-statistics-unit-1.html

exploratory graphs ~ final graphs





Source:
http://www.statmethods.net/advstats/discriminant.html

Base graphics

- density plots (histograms and kernel density plots)
- dot plots
- bar charts (simple, stacked, grouped)
- line charts
- pie charts (simple, annotated, 3D)
- boxplots (simple, notched, violin plots, bagplots)
- scatter plots (simple, with fit lines, scatterplot matrices, high density plots, and 3D plots)

Base graphics – quick example

- plot()
- abline()
- text()

```
> plot(mtcars$wt, mtcars$mpg)
> abline(lm(mtcars$mpg~mtcars$wt))
> title("Regression of MPG per Weight")

> boxplot(mtcars$wt, col = "blue")
```

Saving Graphs

Function	Output to
pdf("mygraph.pdf")	pdf file
win.metafile("mygraph.wmf")	windows metafile
png("mygraph.png")	png file
jpeg("mygraph.jpg")	jpeg file
bmp("mygraph.bmp")	bmp file
postscript("mygraph.ps")	postscript file

dev.off() – closing the device

Set or Query Graphical Parameters

- par(*tag* = *value*, *tag* = *value*, …)
- most used tags:
  - bg, fg
  - cex, cex.axis, cex.lab, cex.main, cex.sub
  - col, col.axis, col.lab, col.main, col.sub,
  - mar
  - mfrow, mfcol

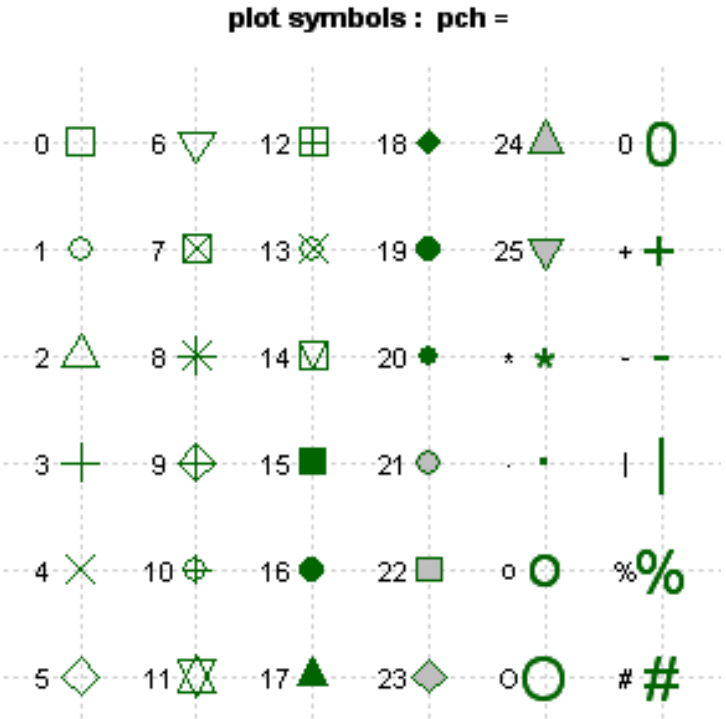
```
> par("col.lab")
"black"
> par("col.lab" = "red")
> save_gpar <- par()
> save(save_gpar, "my_gpar.RData")
> rm(save_gpar)
> load("my_gpar.RData")
> par(save_gpar)
> par("col.lab")
"red"
```

Text and Symbol Size

tag	description
cex	number indicating the amount by which plotting text and symbols should be scaled relative to the default. 1=default, 1.5 is 50% larger, 0.5 is 50% smaller, etc.
cex.axis	magnification of axis annotation relative to cex
cex.lab	magnification of x and y labels relative to cex
cex.main	magnification of titles relative to cex
cex.sub	magnification of subtitles relative to cex

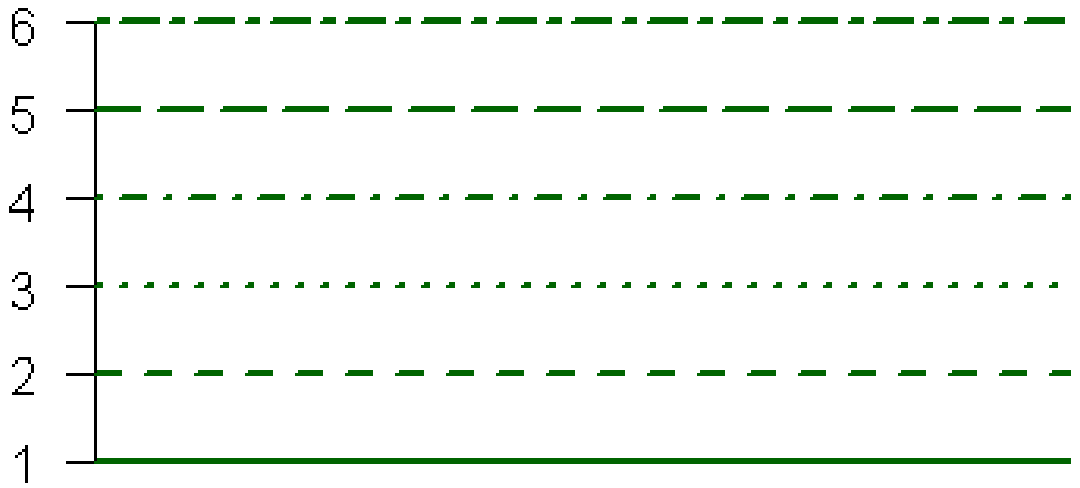
Plotting Symbols

- pch = *option*
- symbols 21 – 25:
  - border color (col=)
  - fill color (bg=)



Plotting Symbols

- lty = *option*
- lwd: line width (default 1)



Colors

- col=1, col="white", col="#FFFFFF"

tag	description
col	Default plotting color. Some functions (e.g. lines) accept a vector of values that are recycled.
col.axis	color for axis annotation
col.lab	color for x and y labels
col.main	color for titles
col.sub	color for subtitles
fg	plot foreground color (axes, boxes - also sets col= to same)
bg	plot background color

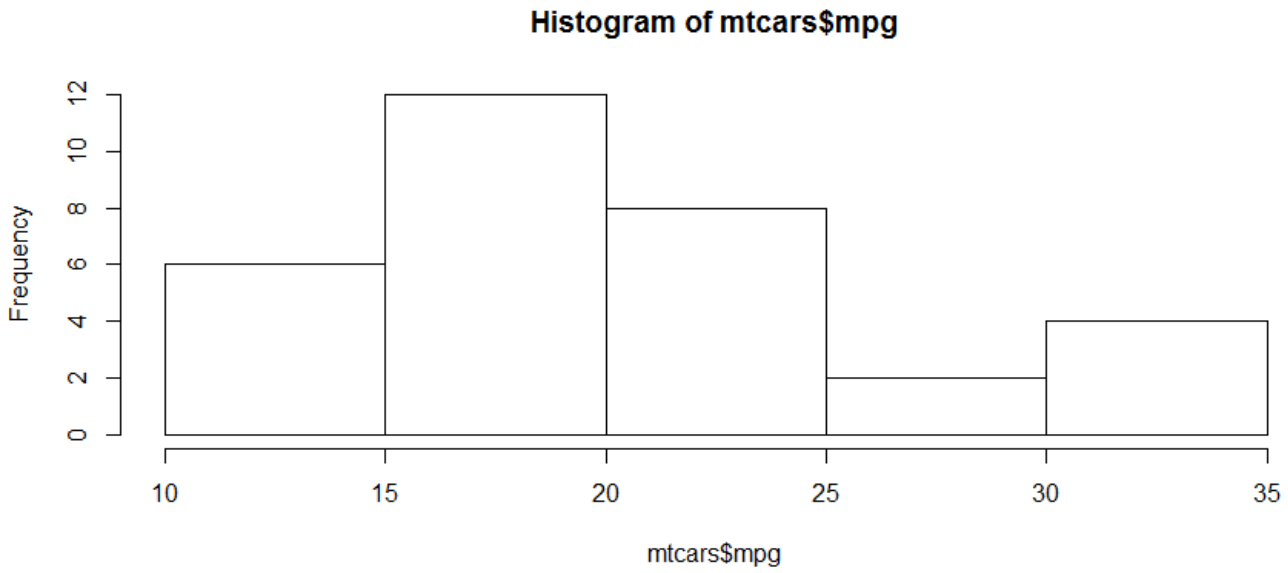
Fonts

- Map your font mapping
- windowsFonts(A=windowsFont("Arial Black"))
- text(4,4,family="A","Hello World from Arial Black")

tag	description
font	Integer specifying font to use for text. 1=plain, 2=bold, 3=italic, 4=bold italic, 5=symbol
font.axis	font for axis annotation
font.lab	font for x and y labels
font.main	font for titles
font.sub	font for subtitles
ps	font point size (roughly 1/72 inch) text size=ps*cex
family	font family for drawing text. Standard values are "serif", "sans", "mono", "symbol". Mapping is device dependent.

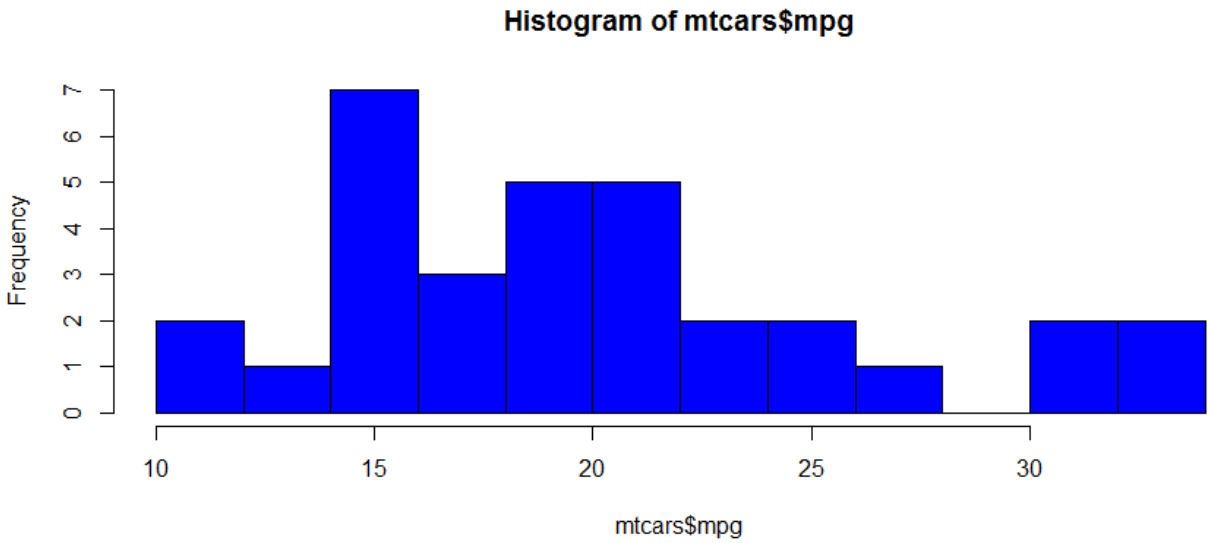
Histograms – default parameters

- hist(mtcars$mpg)



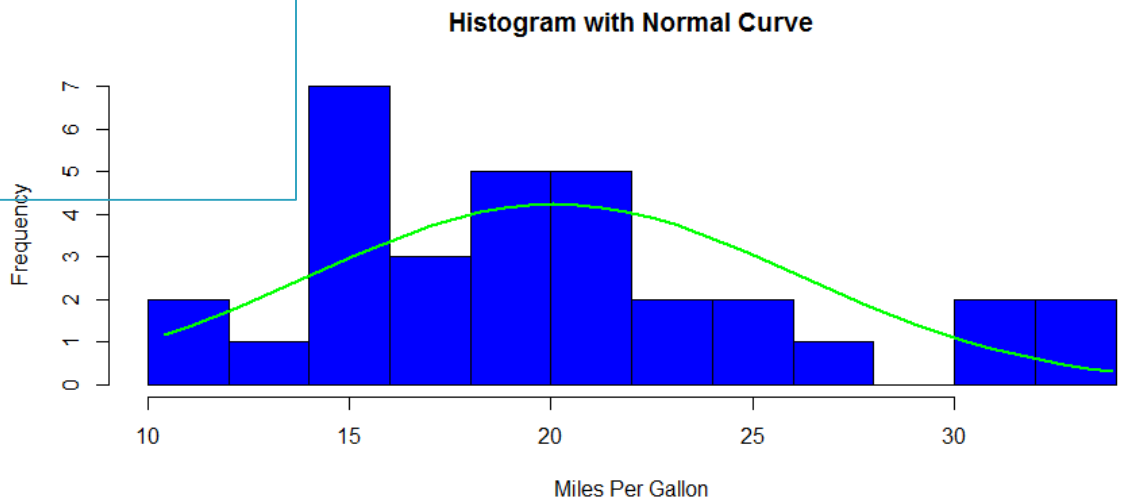
Histograms – custom parameters

- hist(mtcars$mpg, breaks=12, col="blue")



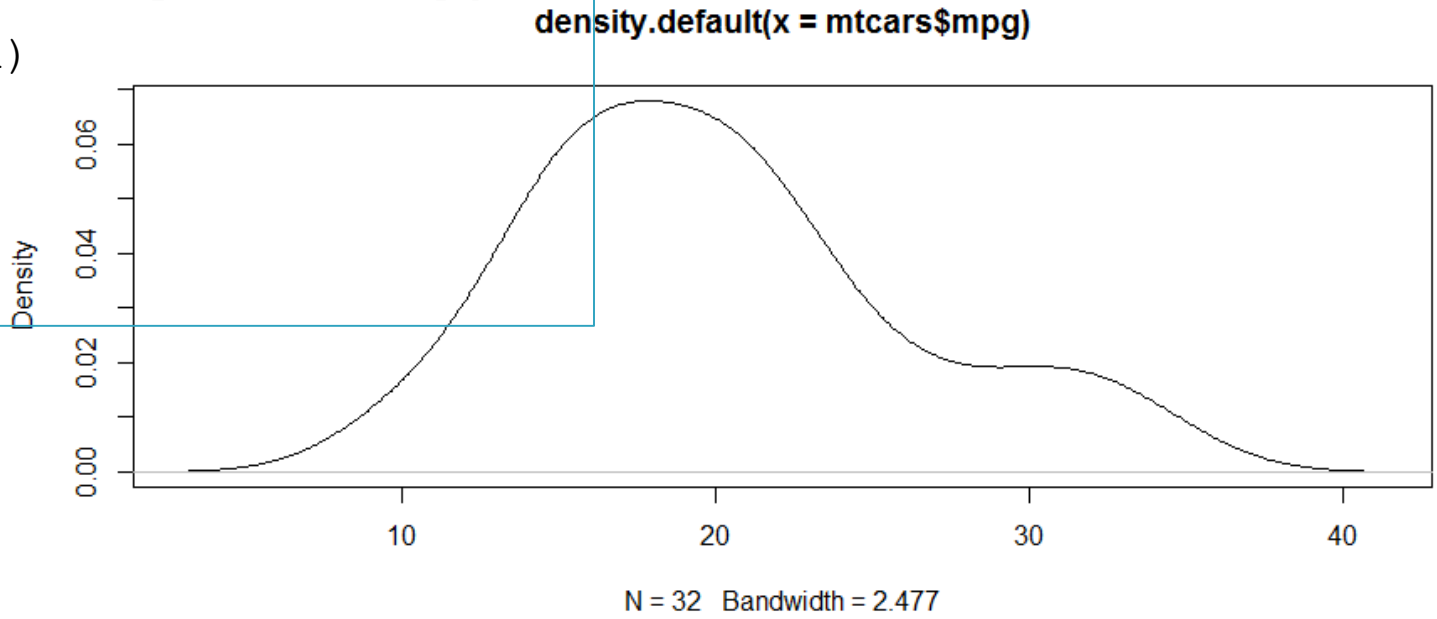
histograms and a normal curve


```
# Add a Normal Curve (Thanks to Peter Dalgaard)
x <- mtcars$mpg
h<-hist(x, breaks=10, col="blue", xlab="Miles Per Gallon",
        main="Histogram with Normal Curve")
xfit<-seq(min(x),max(x),length=40)
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="green", lwd=2)
```



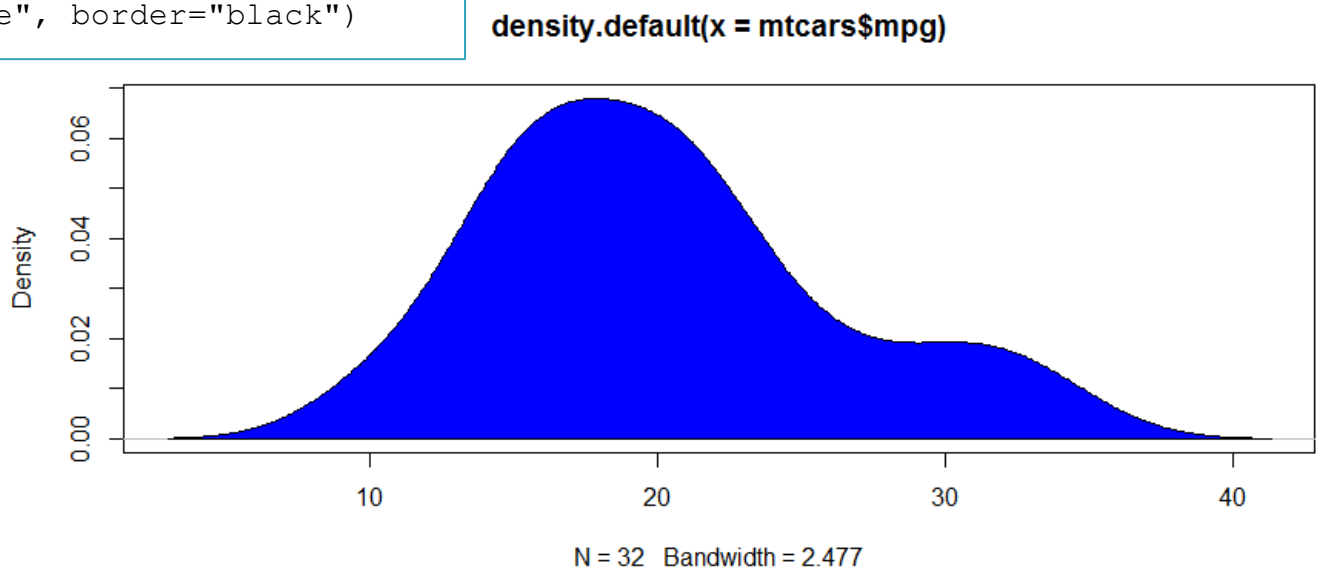
kernel density plots


```
> den <- density(mtcars$mpg)
> plot(den)
```



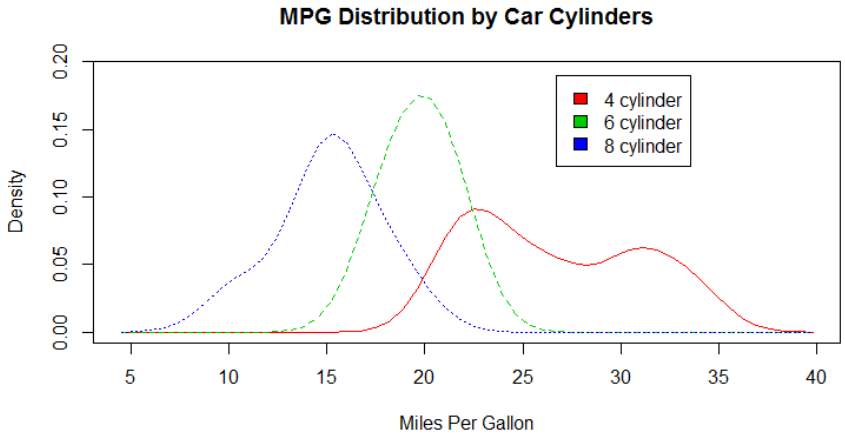
kernel density plots


```
> den <- density(mtcars$mpg)
> plot(den)
> polygon(d, col="blue", border="black")
```



kernel density – comparing groups – {sm} package

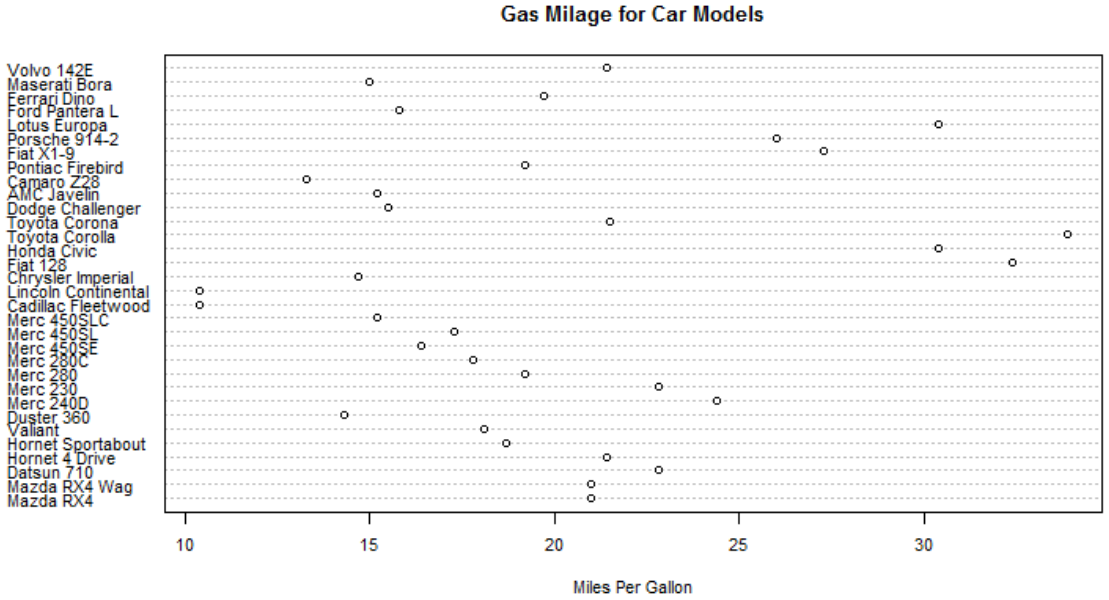

```
# Compare MPG distributions for cars with
# 4,6, or 8 cylinders
install.packages("sm")
library(sm)

# create value labels
cyl.f <- factor(mtcars$cyl, levels = c(4,6,8),
                labels = paste(c(4, 6, 8), "cylinder"))
#  labels = c("4 cylinder", "6 cylinder", "8 cylinder"))


# plot densities
sm.density.compare(mtcars$mpg, mtcars$cyl, xlab = "Miles Per Gallon")
title(main = "MPG Distribution by Car Cylinders")

# add legend via mouse click
colfill <- c(2:(2+length(levels(cyl.f))))
legend(locator(1), levels(cyl.f), fill=colfill)
```



Source: http://www.statmethods.net/graphs/density.html

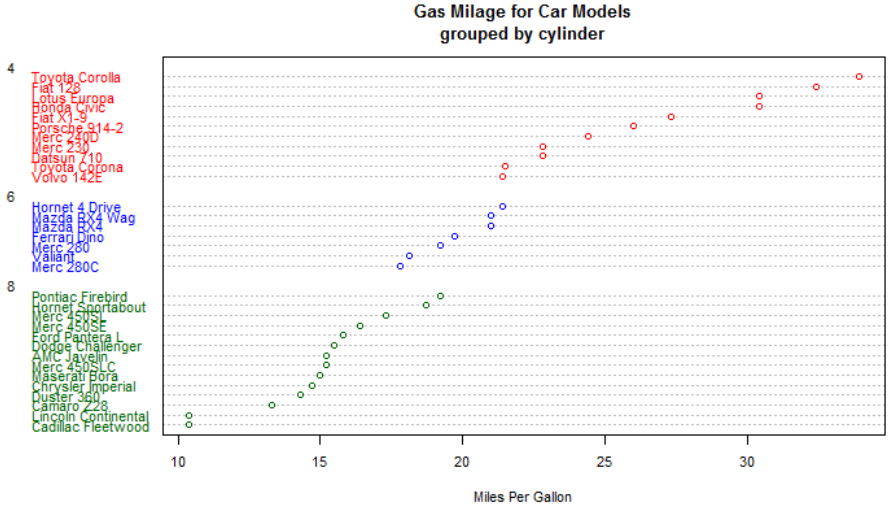
dot plots - dotchart(x, labels=)


```
> dotchart(mtcars$mpg, labels=row.names(mtcars), cex=.7,
         main="Gas Milage for Car Models",
         xlab="Miles Per Gallon")
```



dot plots - dotchart() – grouped an colored by a variable

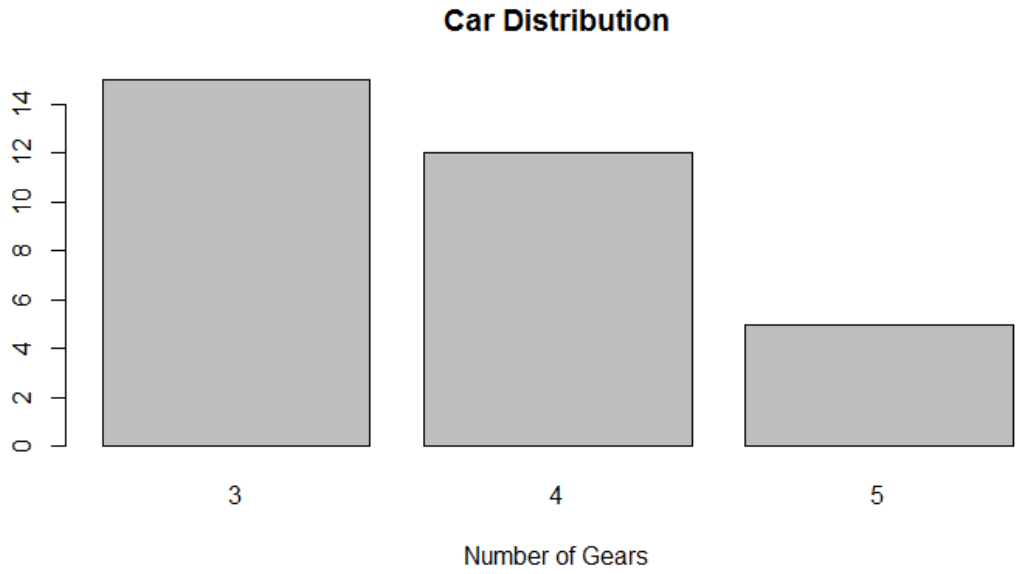

```
> x <- mtcars[order(mtcars$mpg),]
> x$cyl <- factor(x$cyl)
> x$color[x$cyl==4] <- "red"
> x$color[x$cyl==6] <- "blue"
> x$color[x$cyl==8] <- "darkgreen"
> dotchart(x$mpg, labels=row.names(x), cex=.7,
   groups= x$cyl,
   main="Gas Milage for Car Models\ngrouped by cylinder",
   xlab="Miles Per Gallon", gcolor="black", color=x$color)
```



Source: http://www.statmethods.net/graphs/density.html

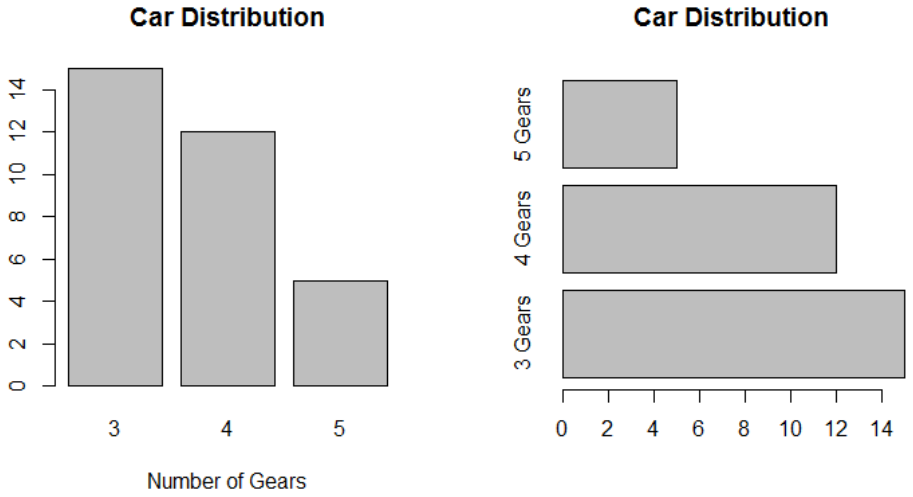
bar charts


```
> counts <- table(mtcars$gear)
> barplot(counts, main="Car Distribution",
        xlab="Number of Gears")
```



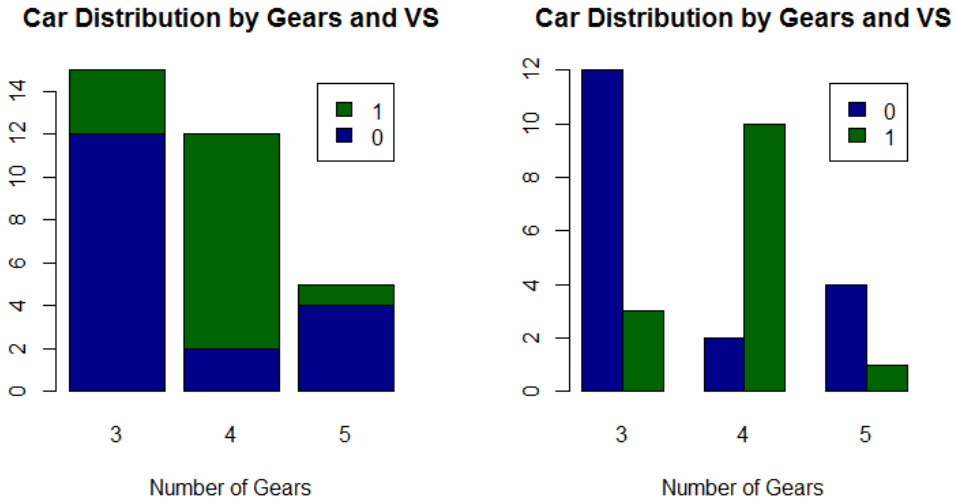
bar charts


```
> counts <- table(mtcars$gear)
> par(mfrow=c(1,2))
> barplot(counts, main="Car Distribution", xlab="Number of Gears")
> barplot(counts, main="Car Distribution", horiz=TRUE, names.arg=c("3 Gears", "4 Gears", "5 Gears"))
```



stacked bar plot vs. grouped bar plot


```
> counts <- table(mtcars$vs, mtcars$gear)
> barplot(counts, main="Car Distribution by Gears and VS", xlab="Number of Gears", col=c("darkblue","darkgreen"),
        legend = rownames(counts))
> barplot(counts, main="Car Distribution by Gears and VS", xlab="Number of Gears", col=c("darkblue","darkgreen"),
        legend = rownames(counts), beside = TRUE)
```

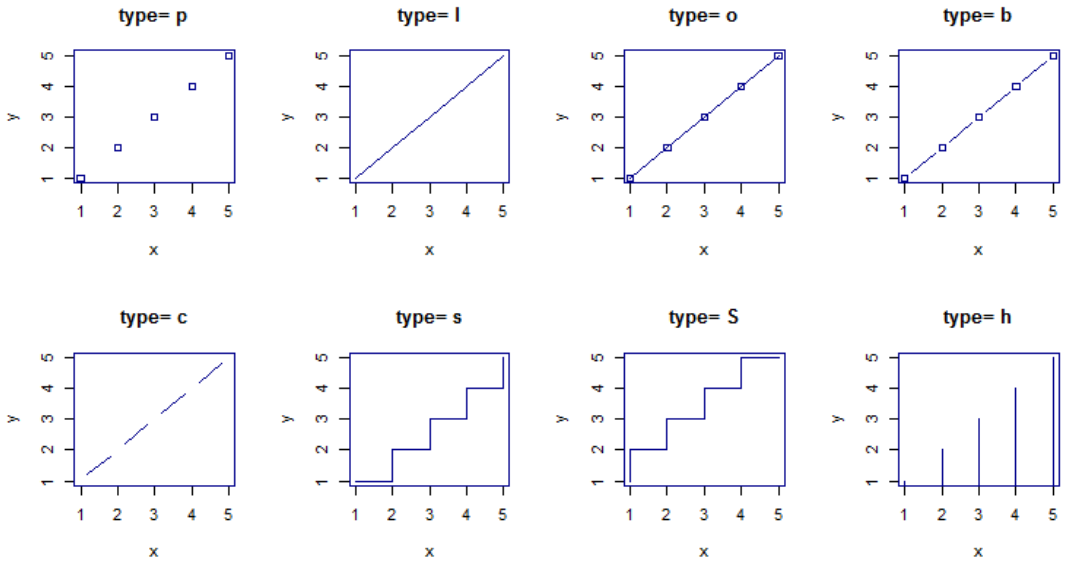


Source: http://www.statmethods.net/graphs/bar.html

line charts – without points


```
> x <- y <- 1:5
> par(pch=22, col="darkblue")
> par(mfrow=c(2,4))
> types = c("p","l","o","b","c","s","S","h")
> for(i in 1:length(types)){
        heading = paste("type=", types[i])
        plot(x, y, type="n", main=heading)
        lines(x, y, type=types[i])
}
```

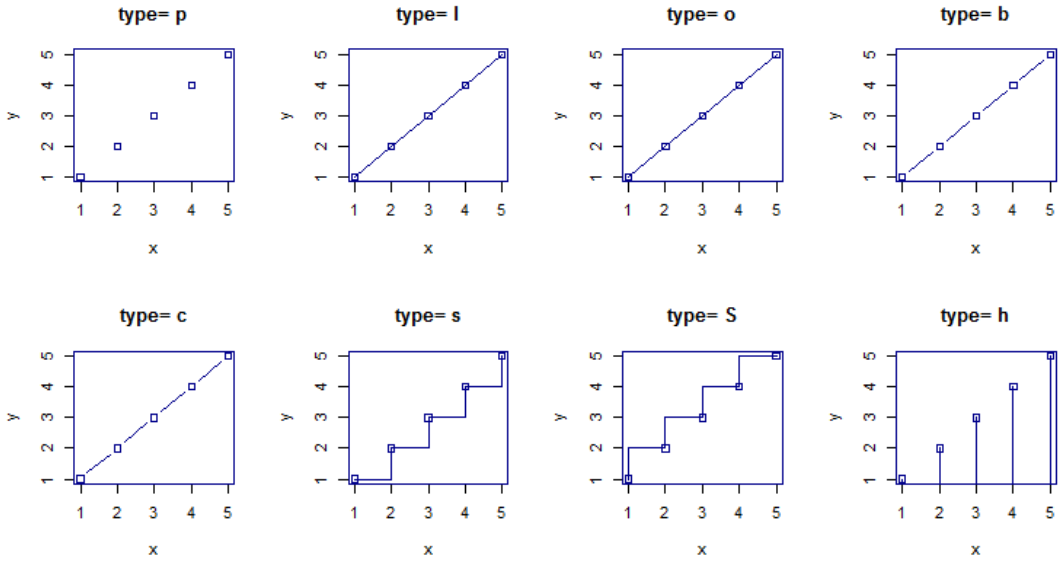
type	description
p	points
l	lines
o	overplotted points and lines
b, c	points (empty if "c") joined by lines
s, S	stair steps
h	histogram-like vertical lines
n	does not produce any points or lines



Source: http://www.statmethods.net/graphs/line.html

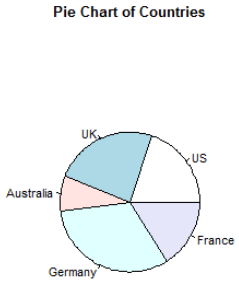
line charts – with points

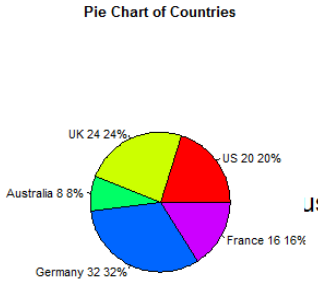

```
> x <- y <- 1:5
> par(pch=22, col="darkblue")
> par(mfrow=c(2,4))
> types = c("p","l","o","b","c","s","S","h")
> for(i in 1:length(types)){
        heading = paste("type=", types[i])
        plot(x, y, type="n", main=heading)
        lines(x, y, type=types[i])
}
```

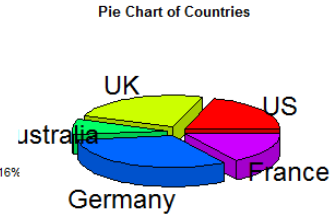


pie charts


```
> slices <- c(10, 12, 4, 16, 8)
> lbls <- c("US", "UK", "Australia", "Germany", "France")

> par(mfrow=c(1,3))

> pie(slices, labels = lbls, main="Pie Chart of Countries")

> lblsp <- paste(lbls,
      paste(round(slices/sum(slices)*100),"%", sep=""))
> pie(slices, labels = lblsp, col=rainbow(length(lblsp)),
      main="Pie Chart of Countries")

# 3D Pie Chart
> install.packages("plotrix")
> library(plotrix)
> pie3D(slices, labels=lbls, explode=0.1,
      main="Pie Chart of Countries ")
```

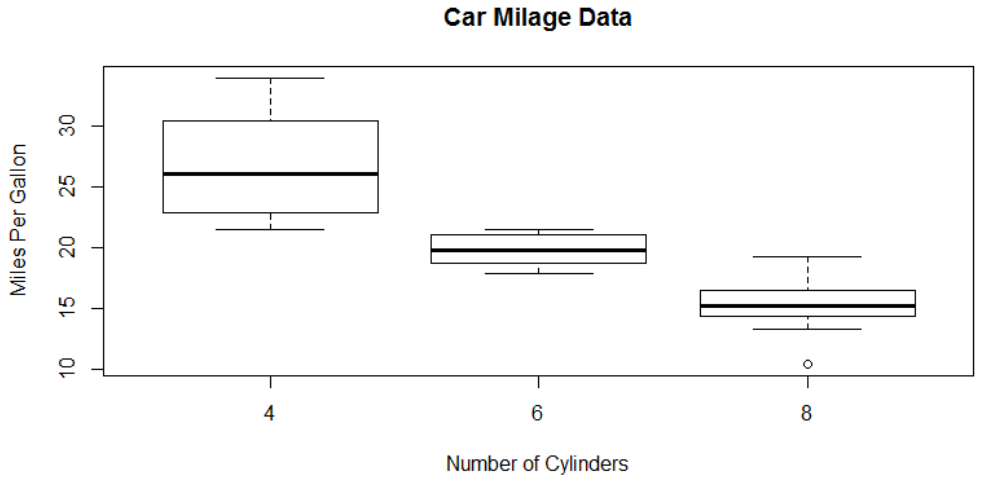






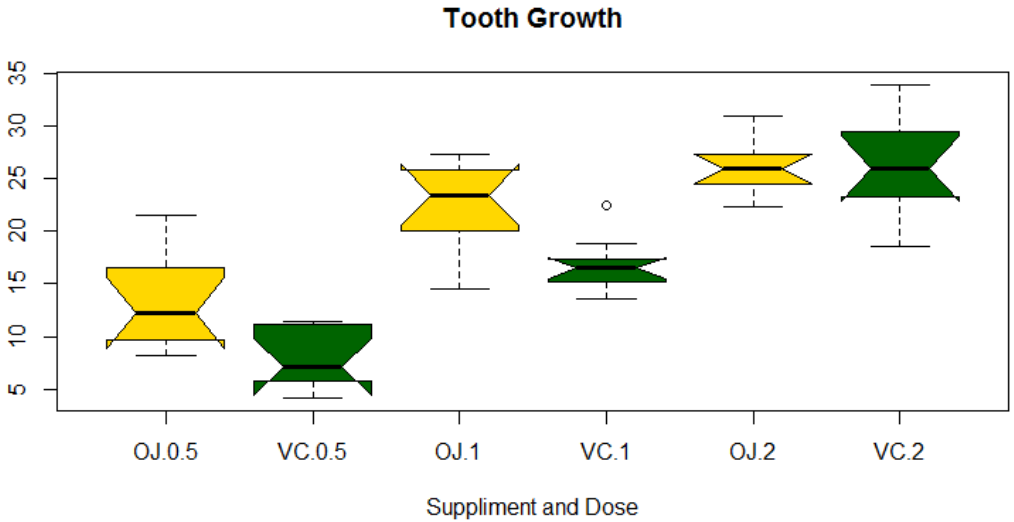
boxplots


```
> boxplot(mpg~cyl, data=mtcars, main="Car Milage Data",
      xlab="Number of Cylinders", ylab="Miles Per Gallon")
```



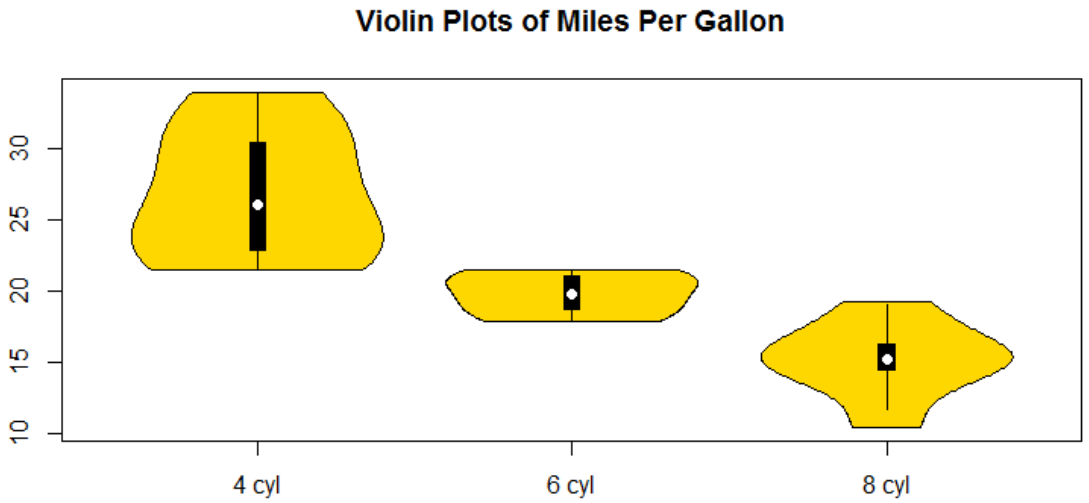
notched boxplots


```
> boxplot(len~supp*dose, data=ToothGrowth, notch=TRUE,
        col=(c("gold","darkgreen")),
        main="Tooth Growth", xlab="Suppliment and Dose")
```



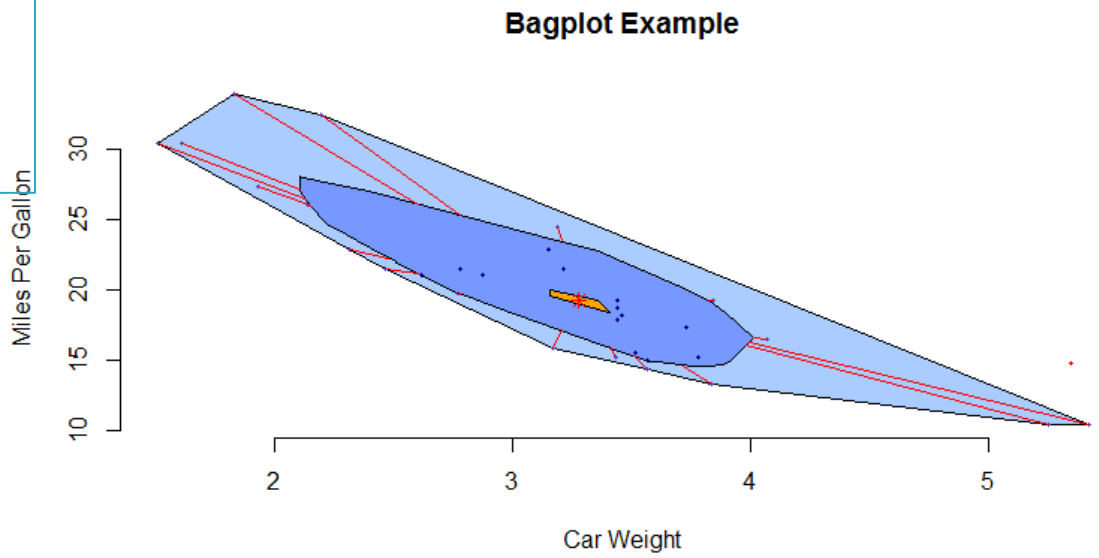
violin boxplots


```
> install.packages("vioplot")
> library(vioplot)
> x1 <- mtcars$mpg[mtcars$cyl==4]
> x2 <- mtcars$mpg[mtcars$cyl==6]
> x3 <- mtcars$mpg[mtcars$cyl==8]
> vioplot(x1, x2, x3,
    names=paste(c(4,6,8), "cyl"),
    col="gold")
> title("Violin Plots of Miles Per Gallon")
```



bagplot

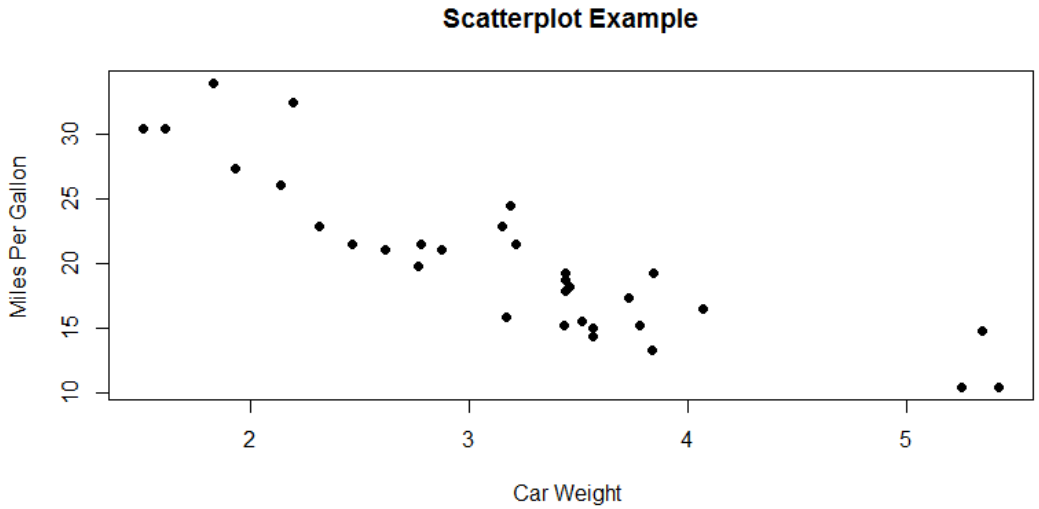

```
> install.packages("aplpack")
> library(aplpack)
> bagplot(mtcars$wt, mtcars$mpg,
    xlab="Car Weight",
    ylab="Miles Per Gallon",
    main="Bagplot Example")
```



Source: http://www.statmethods.net/graphs/boxplot.html

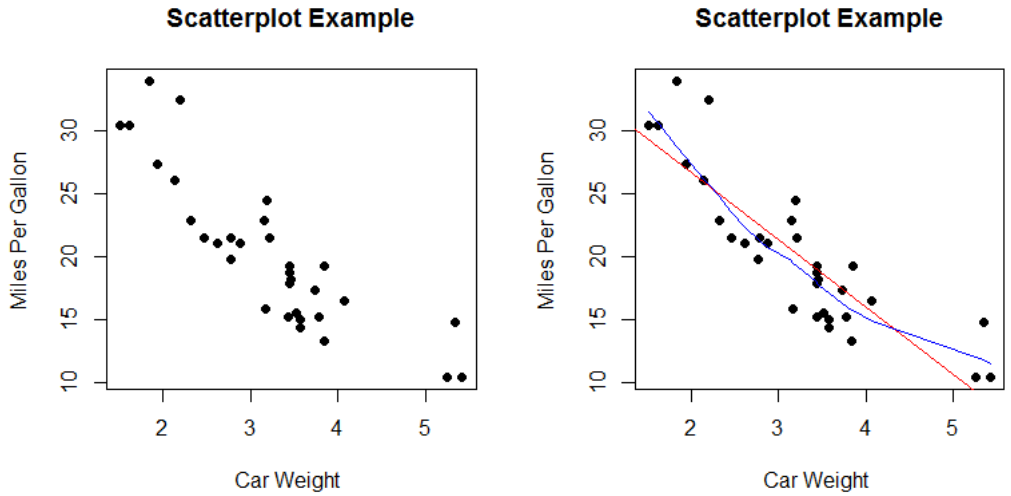
scatter plots


```
> plot(wt, mpg, main="Scatterplot Example",
          xlab="Car Weight ", ylab="Miles Per Gallon ", pch=19)
```



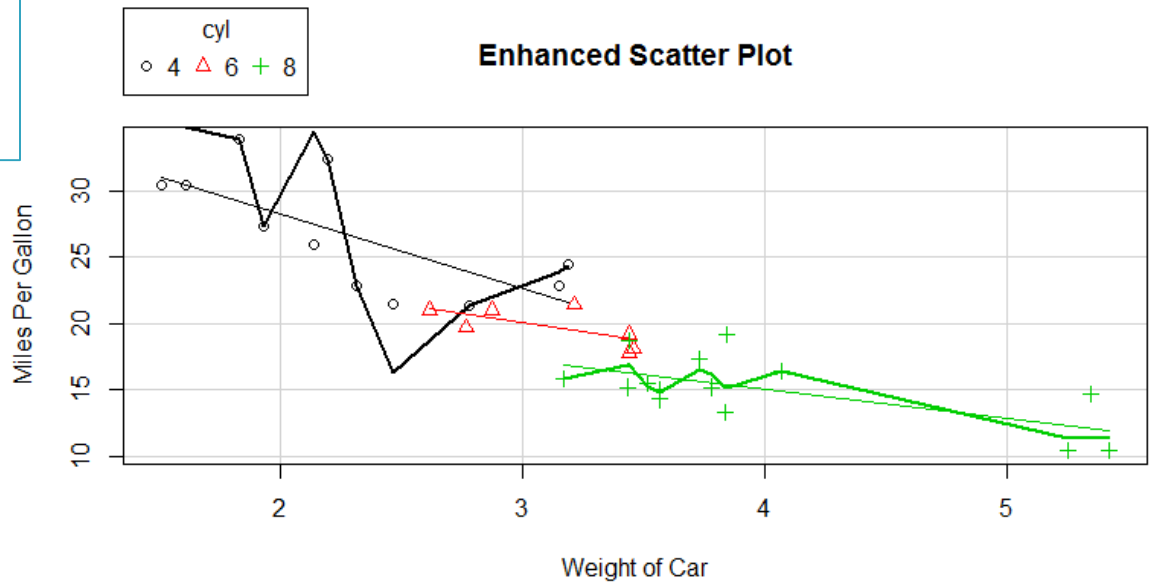
scatter plots + fit lines


```
> par(mfrow=c(1,2))
> plot(mtcars$wt, mtcars$mpg, main="Scatterplot Example", xlab="Car Weight ", ylab="Miles Per Gallon ", pch=19)
> abline(lm(mtcars$mpg~mtcars$wt), col="red")
> lines(lowess(mtcars$wt, mtcars$mpg), col="blue")
```



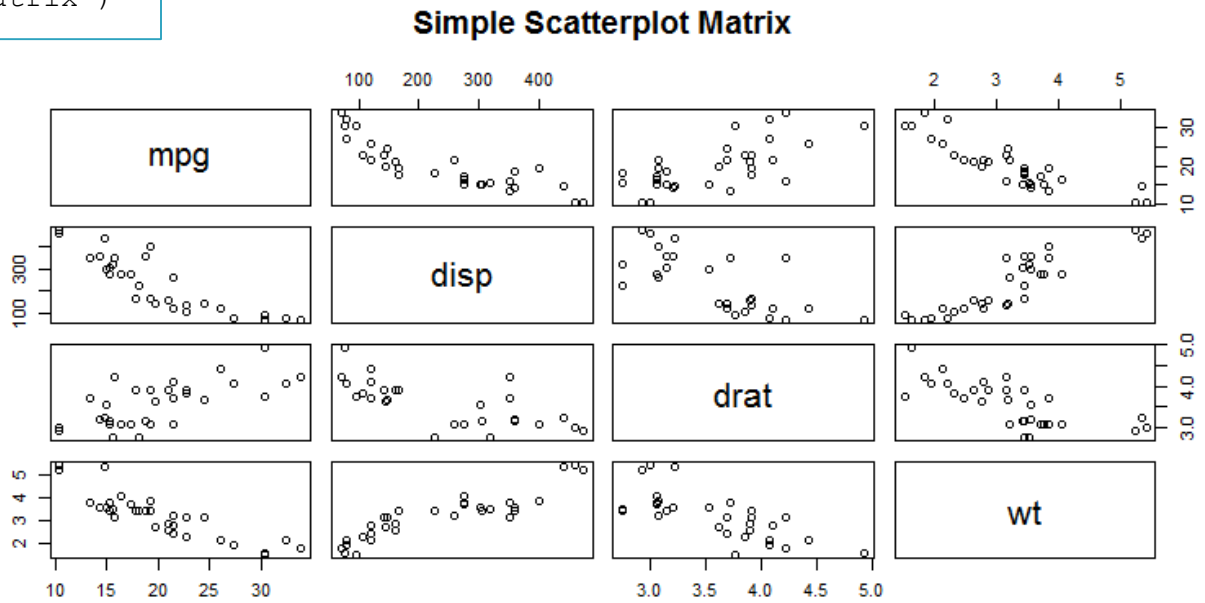
scatterplot() – {car} package


```
> install.packages("car")
> library(car)
> scatterplot(mpg ~ wt | cyl, data=mtcars,
              xlab="Weight of Car",
              ylab="Miles Per Gallon",
              main="Enhanced Scatter Plot",
              labels=row.names(mtcars))
```



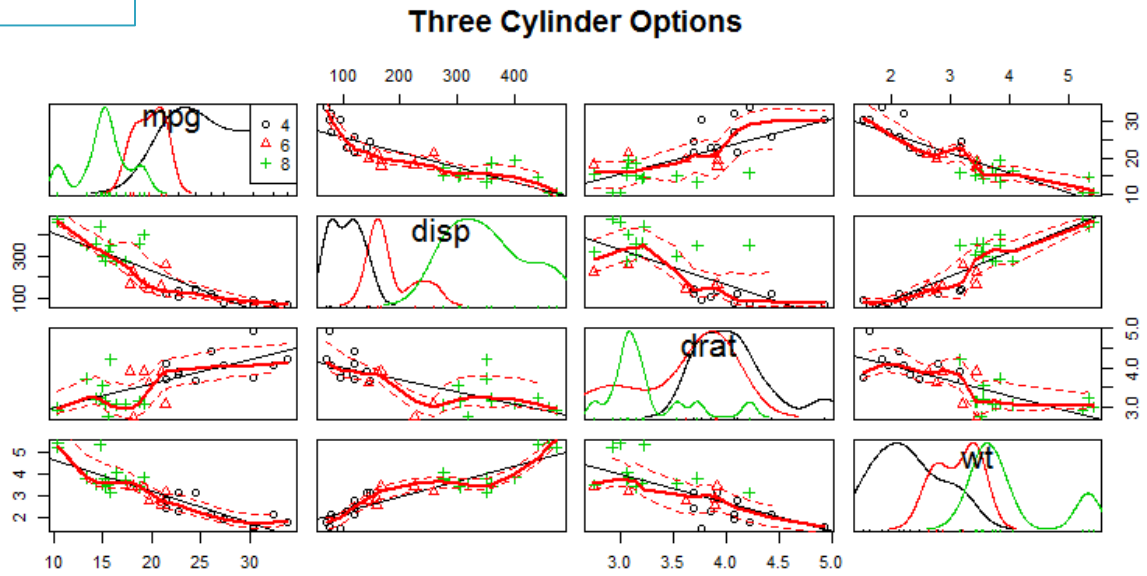
scatterplot matrices


```
> pairs(~mpg+disp+drat+wt,data=mtcars,
    main="Simple Scatterplot Matrix")
```



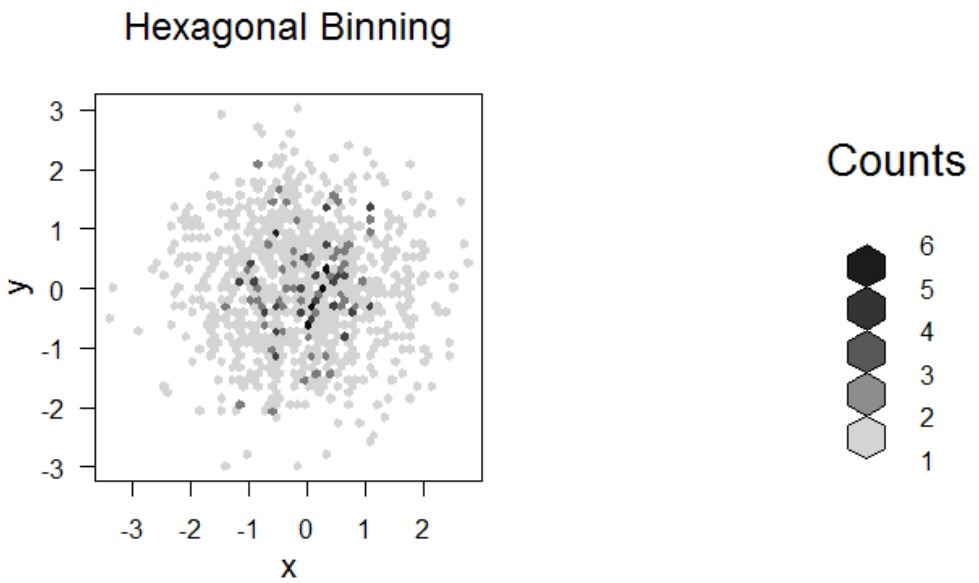
scatterplot matrices – {car} package


```
> library(car)
> scatterplot.matrix(~mpg+disp+drat+wt|cyl, data=mtcars,
                     main="Three Cylinder Options")
```



high density scatterplots – {hexbin} package

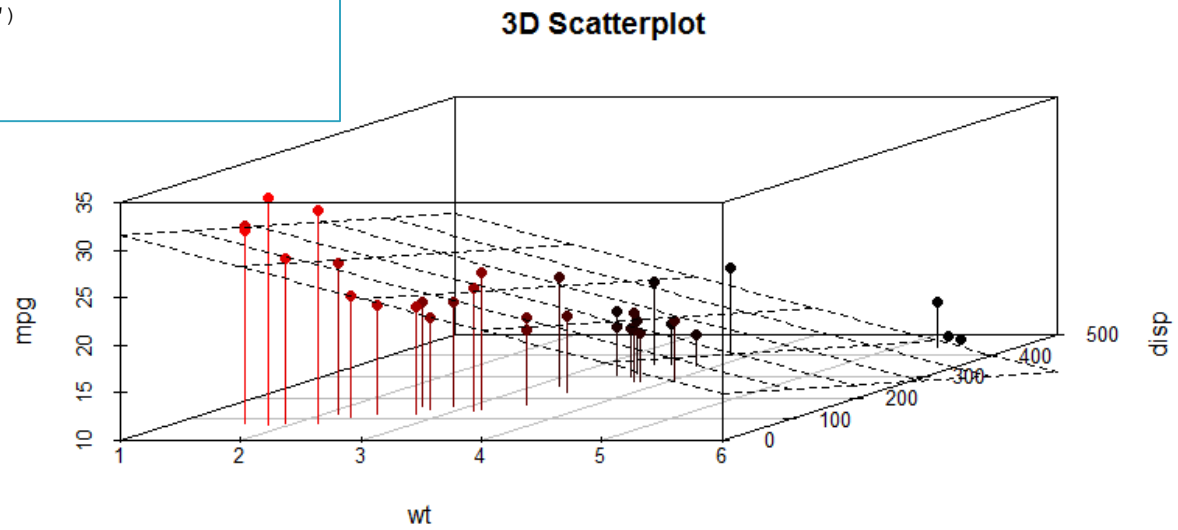

```
> install.packages("hexbin")
> library(hexbin)
> x <- rnorm(1000)
> y <- rnorm(1000)
> bin <- hexbin(x, y, xbins=50)
> plot(bin, main="Hexagonal Binning")
```



Source: http://www.statmethods.net/graphs/scatterplot.html

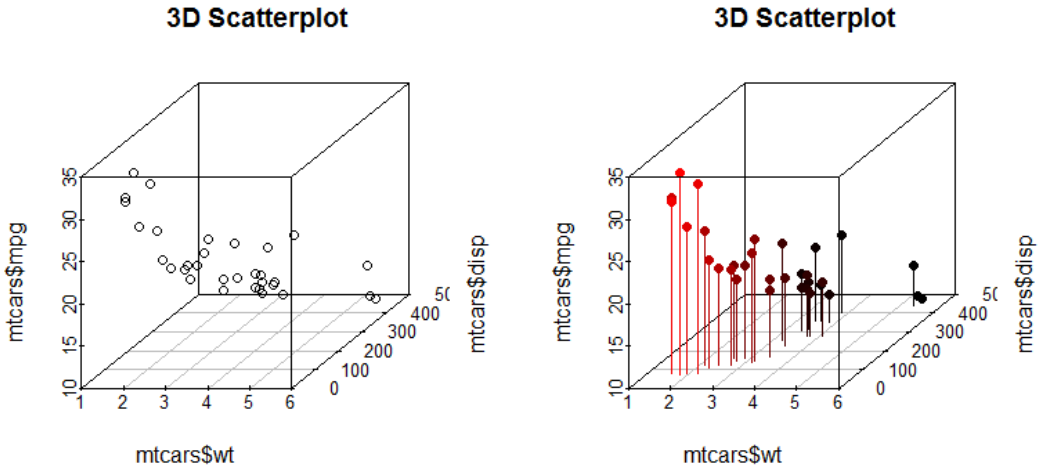
3D Scatterplots + regression plane


```
> install.packages("scatterplot3d")
> attach(mtcars)
> s3d <-scatterplot3d(wt,disp,mpg, pch=16, highlight.3d=TRUE,
        type="h", main="3D Scatterplot")
> fit <- lm(mpg ~ wt+disp)
> s3d$plane3d(fit)
```



3D Scatterplots – {scatterplot3d} package


```
> install.packages("scatterplot3d")
> library(scatterplot3d); par(mfrow=c(1,2))
> scatterplot3d(mtcars$wt, mtcars$disp, mtcars$mpg, main="3D Scatterplot")
> scatterplot3d(mtcars$wt, mtcars$disp, mtcars$mpg, pch=16, highlight.3d=TRUE, type="h", main="3D Scatterplot")
```



Data Visualization & Graphics Environments

- Base graphics (Scatterplot, Box-and-whiskers plot, Histogram)
- Lattice
- ggplot2
- Interactive graphics in R
- Reproducibility

{lattice} package

- written by Deepayan Sarkar
- improve on base R graphics
- provide better defaults
- enhance the ability to easily display multivariate relationships

- *graph_type*(*formula*, data=)

! complex functions require well understanding

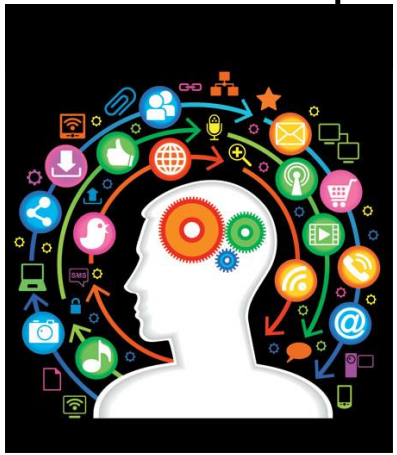


Image source: http://businesslife.ba.com/News-and-Blogs/Chris-Barez-Brown/How-to-have-a-great-idea.html

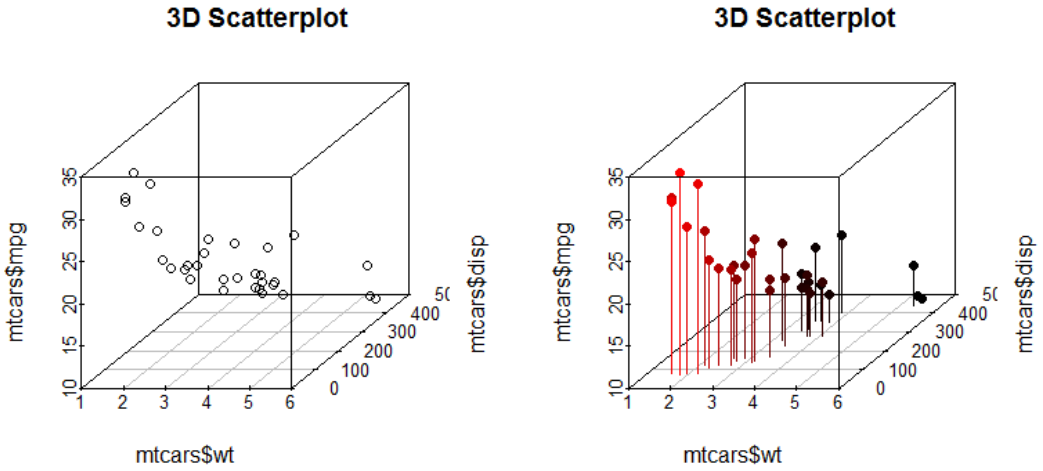
Lattice - graph type

- ~x display numeric variable x only
- ~x|A display numeric variable x for each level of factor A
- y~x | A*B display the relationship between numeric variables y and x separately for every combination of factor A and B levels

graph_type	description	formula examples
barchart	bar chart	x~A or A~x
bwplot	boxplot	x~A or A~x
cloud	3D scatterplot	z~x*y|A
contourplot	3D contour plot	z~x*y
densityplot	kernal density plot	~x|A*B
dotplot	dotplot	~x|A
histogram	histogram	~x
levelplot	3D level plot	z~y*x
parallel	parallel coordinates plot	data frame
splom	scatterplot matrix	data frame
stripplot	strip plots	A~x or x~A
xyplot	scatterplot	y~x|A
wireframe	3D wireframe graph	z~y*x

{lattice} package

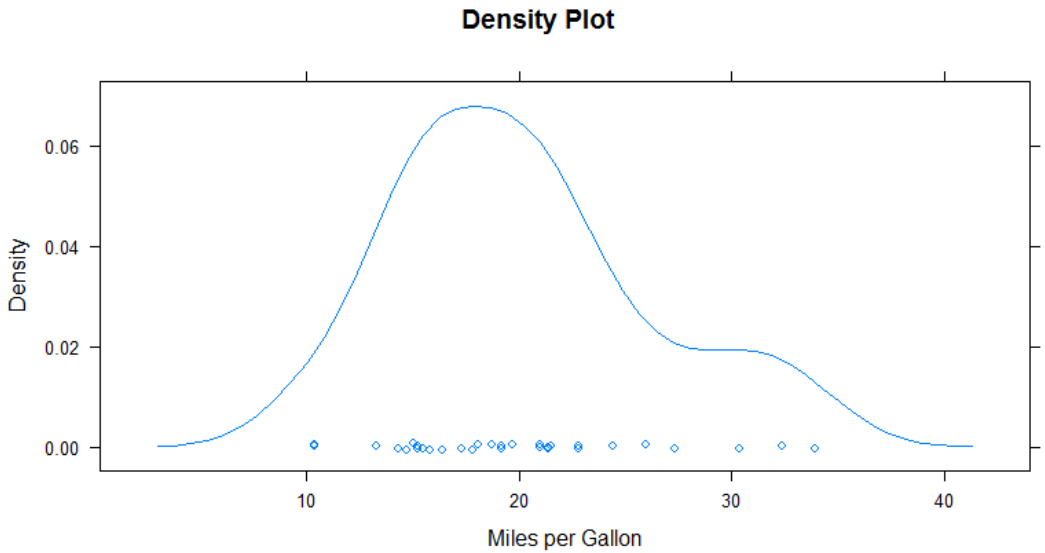

```
> install.packages("scatterplot3d")
> library(scatterplot3d); par(mfrow=c(1,2))
> scatterplot3d(mtcars$wt, mtcars$disp, mtcars$mpg, main="3D Scatterplot")
> scatterplot3d(mtcars$wt, mtcars$disp, mtcars$mpg, pch=16, highlight.3d=TRUE, type="h", main="3D Scatterplot")
```



Source: http://www.statmethods.net/advgraphs/trellis.html

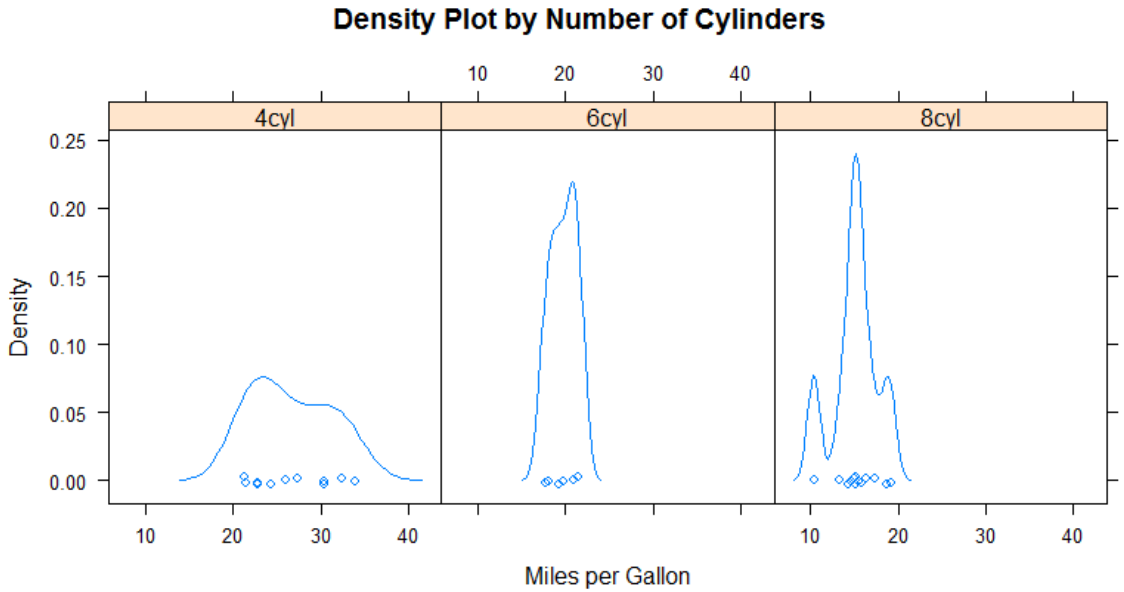
kernel density plot


```
> library(lattice)
> attach(mtcars)
> # create factors with value labels
> gear.f <- factor(gear, levels=c(3,4,5),
                   labels = c("3gears","4gears","5gears"))
> cyl.f <- factor(cyl, levels=c(4,6,8),
                   labels = c("4cyl","6cyl","8cyl"))
> densityplot(~mpg,
              main="Density Plot",
              xlab="Miles per Gallon")
```



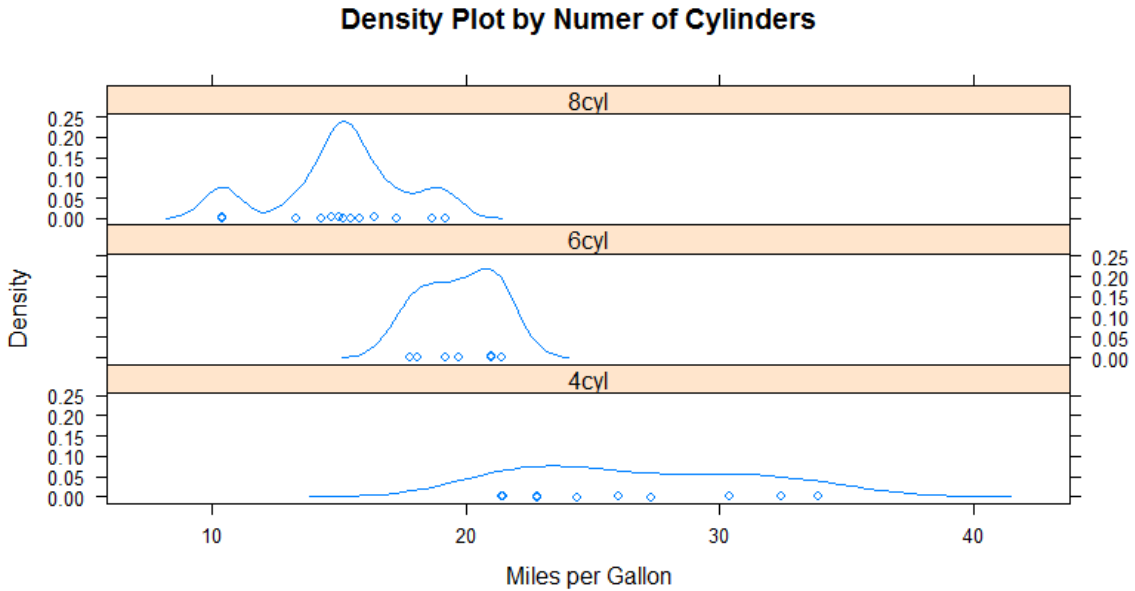
kernel density plots by factor level


```
> densityplot(~mpg|cyl.f,
              main = "Density Plot by Number of Cylinders",
              xlab = "Miles per Gallon")
```



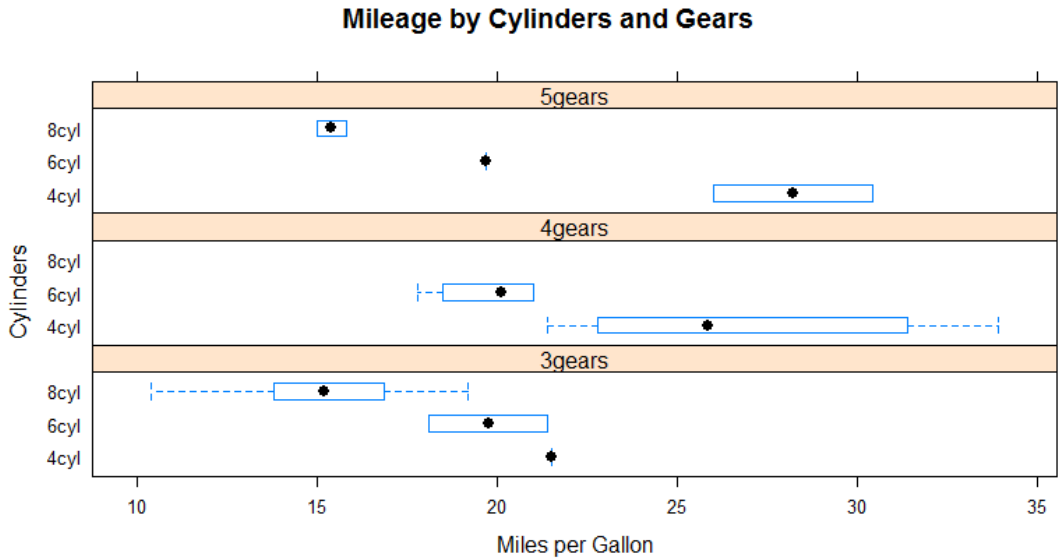
kernel density plots by factor level (alternate layout)


```
> densityplot(~mpg|cyl.f,
              main = "Density Plot by Numer of Cylinders",
              xlab = "Miles per Gallon",
              layout = c(1,3))
```



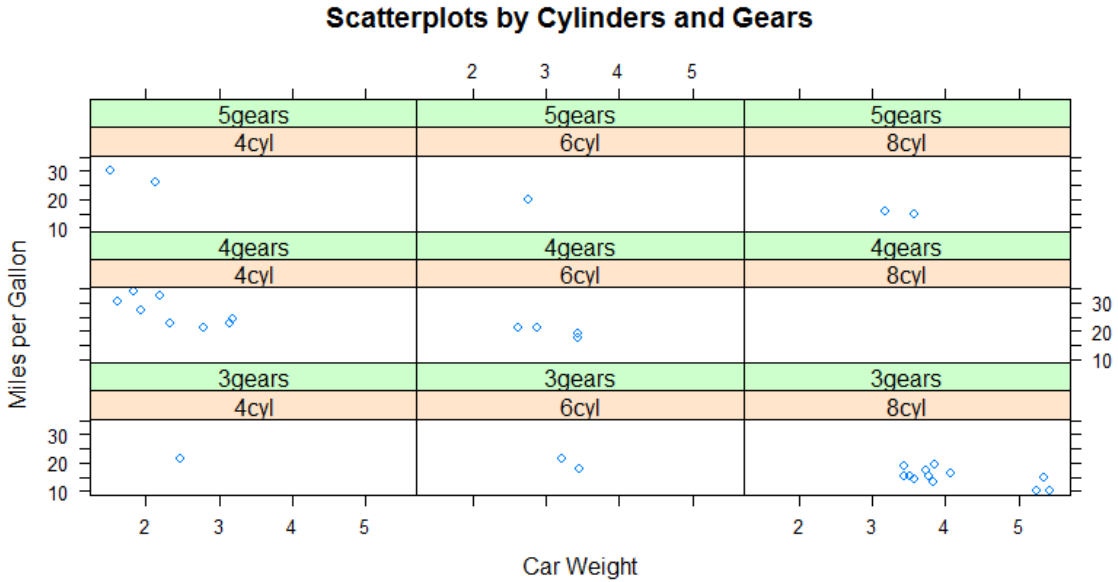
bwplot()


```
> bwplot(cyl.f~mpg|gear.f,
       ylab = "Cylinders", xlab = "Miles per Gallon",
       main = "Mileage by Cylinders and Gears",
       layout = (c(1,3)))
```



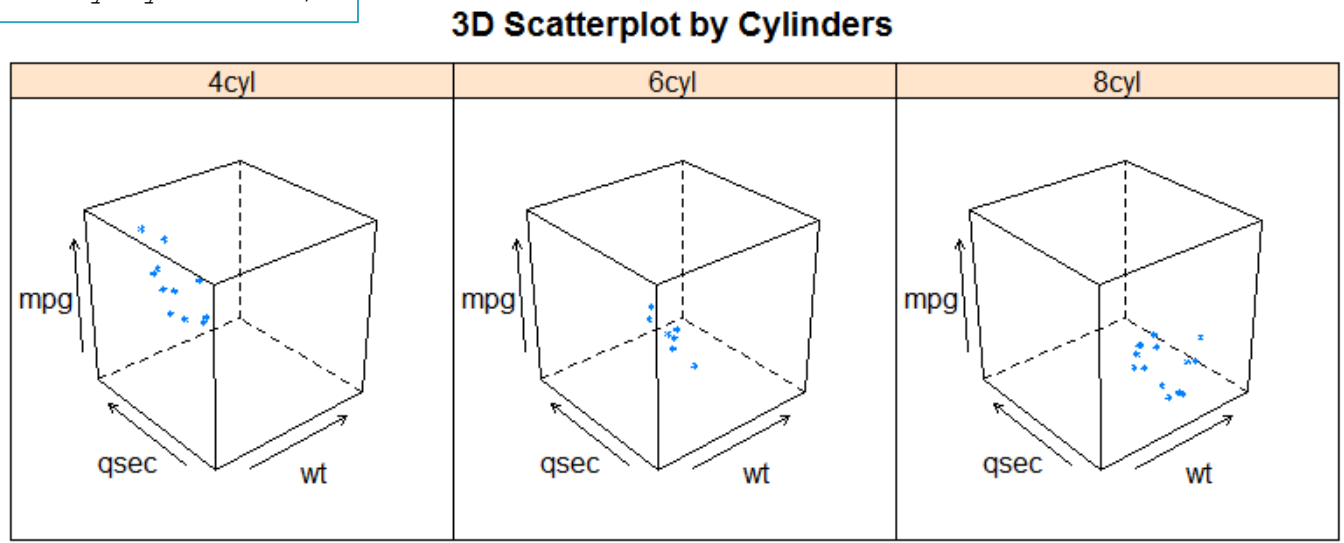
xyplot()


```
> xyplot(mpg~wt|cyl.f*gear.f,
      main = "Scatterplots by Cylinders and Gears",
      ylab = "Miles per Gallon", xlab = "Car Weight")
```



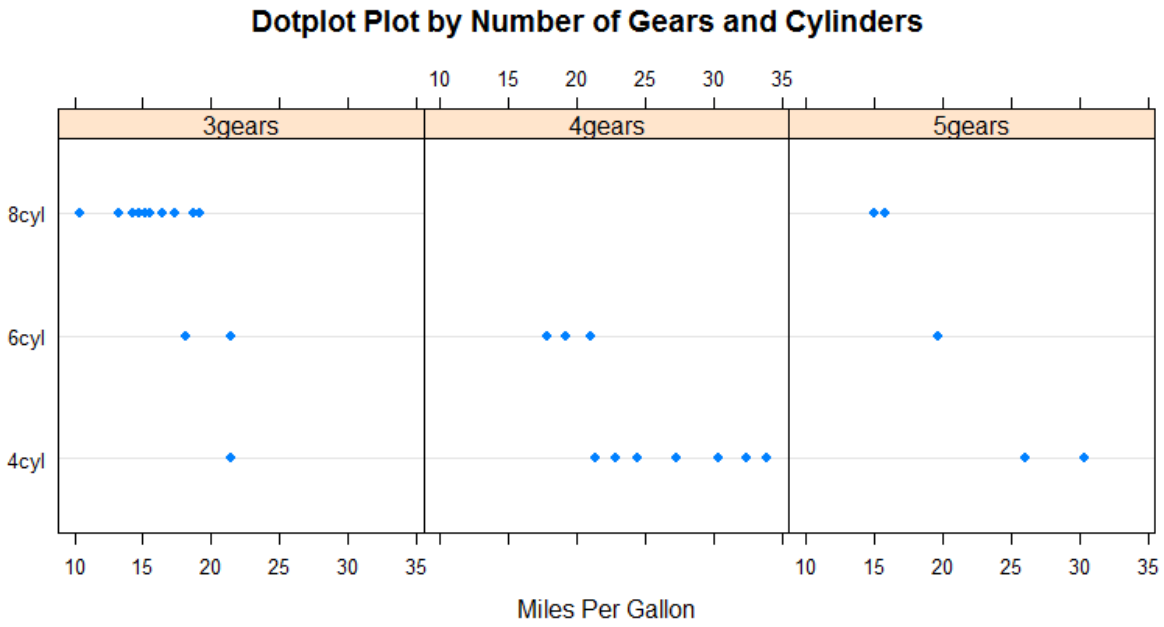
cloud()


```
> cloud(mpg~wt*qsec|cyl.f,
      main = "3D Scatterplot by Cylinders")
```



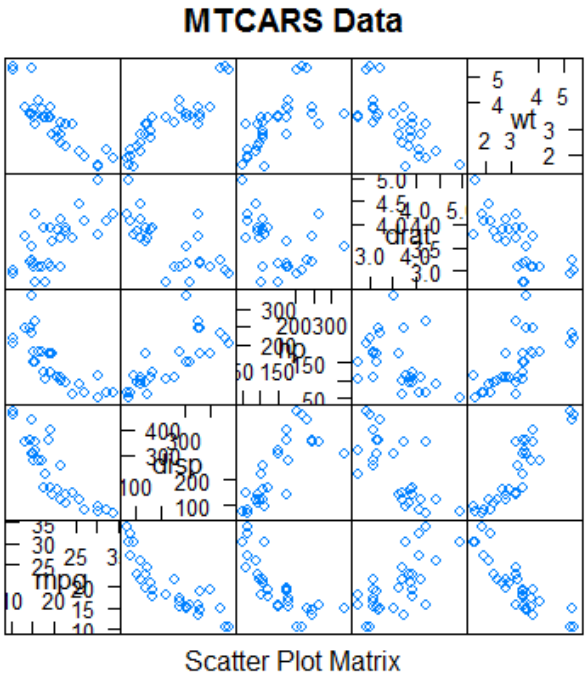
dotplot()


```
> dotplot(cyl.f~mpg|gear.f,
        main = "Dotplot Plot by Number of Gears and Cylinders",
        xlab = "Miles Per Gallon")
```



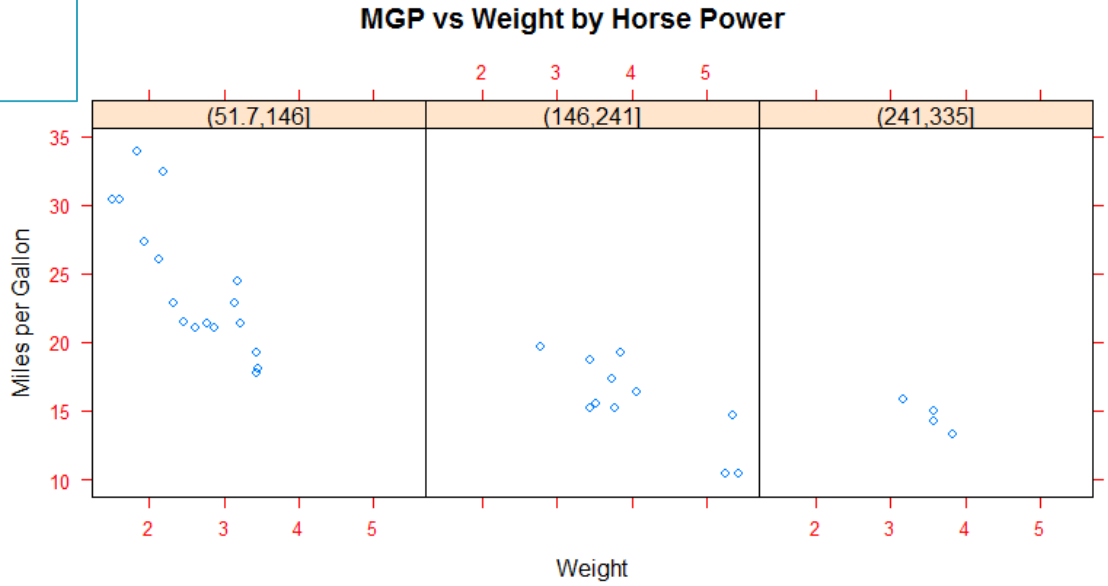
splom() - scatterplot matrix


```
> splom(mtcars[c(1,3,4,5,6)],
      main = "MTCARS Data")
```



Customizing Lattice Graphs


```
> attach(mtcars)
> hp <- cut(hp,3)
> xyplot(mpg ~ wt|hp, scales = list(cex=.8, col="red"),
       xlab = "Weight", ylab = "Miles per Gallon",
       main = "MGP vs Weight by Horse Power")
```



Data Visualization & Graphics Environments

- Base graphics (Scatterplot, Box-and-whiskers plot, Histogram)
- Lattice
- ggplot2
- Interactive graphics in R
- Reproducibility

{ggplot2} package

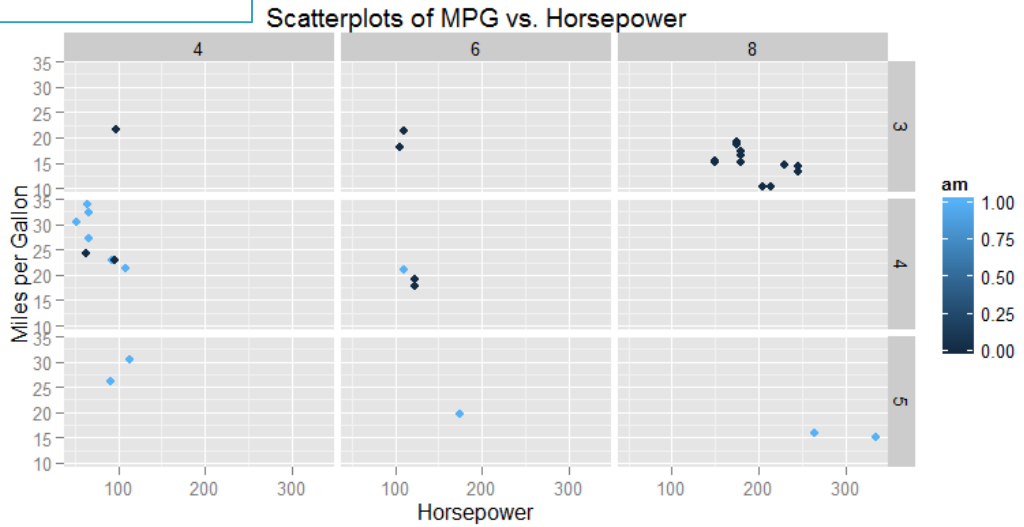
- created by Hadley Wickham
- represent both univariate and multivariate numerical and categorical data
- grouping can be represented by color, symbol, size, and transparency

- *qplot(x, y, data=, color=, shape=, size=, alpha=, geom=, method=, formula=, facets=, xlim=, ylim= xlab=, ylab=, main=, sub=)*

Source: http://www.statmethods.net/advgraphs/ggplot2.html

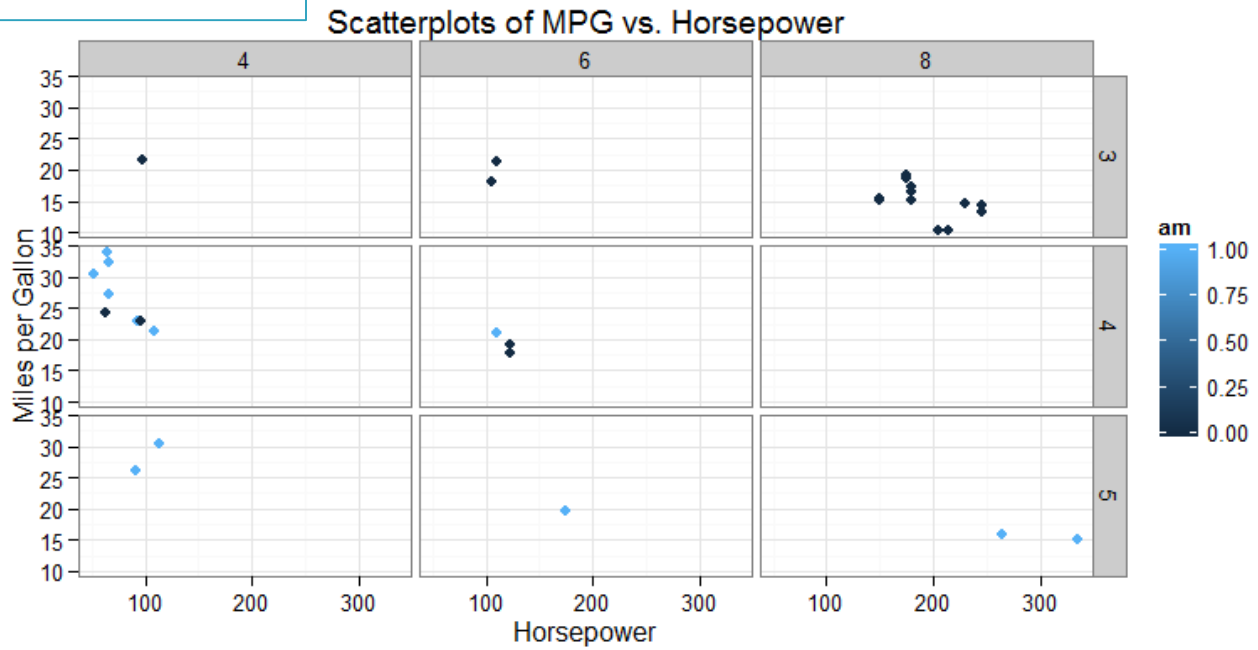
Customizing ggplot2 Graphs


```
> library(ggplot2)
> p <- qplot(hp, mpg, data = mtcars, color = am,
           facets = gear~cyl, main = "Scatterplots of MPG vs. Horsepower",
           xlab = "Horsepower", ylab = "Miles per Gallon")
> p
```



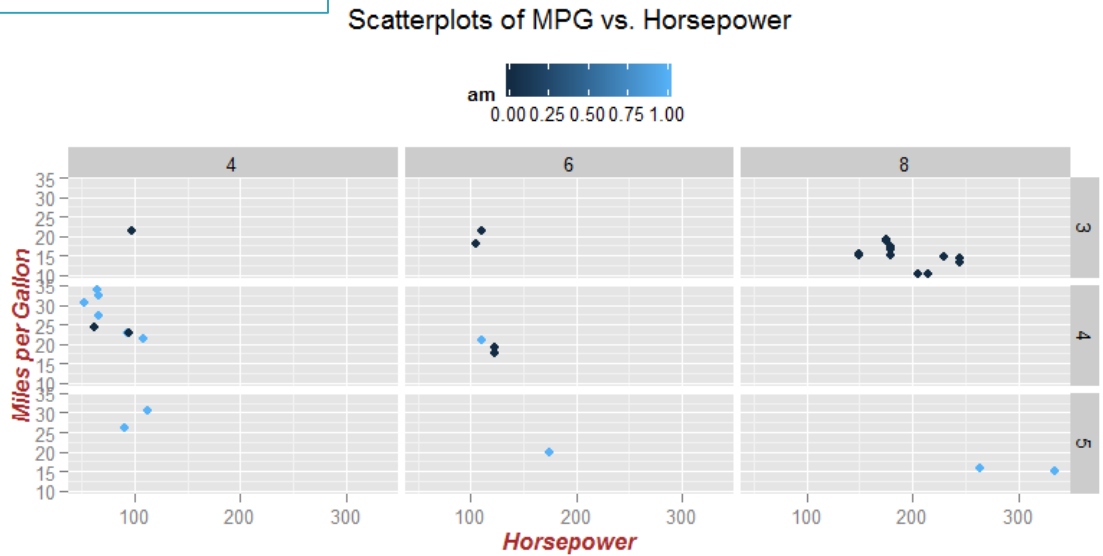
Customizing ggplot2 Graphs


```
# White background and black grid lines
> p + theme_bw()
```



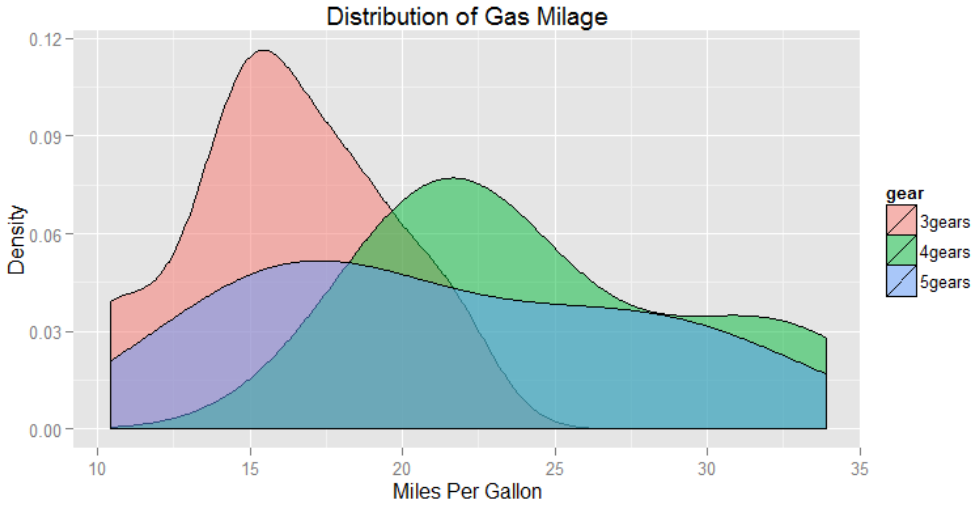
Customizing ggplot2 Graphs


```
# Large brown bold italics labels
# and legend placed at top of plot
> p + theme(axis.title=element_text(face="bold.italic",
       size="12", color="brown"), legend.position="top")
```



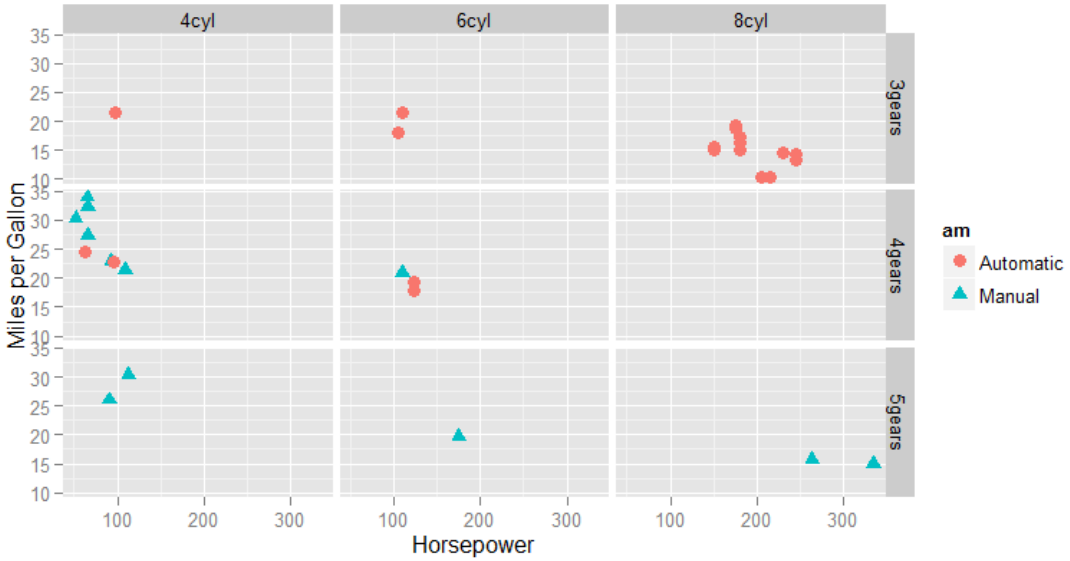
Kernel density plots


```
> library(ggplot2)
> mtcars$gear <- factor(mtcars$gear, levels = c(3,4,5), labels = c("3gears", "4gears",
"5gears"))
> mtcars$am <- factor(mtcars$am, levels = c(0, 1), labels = c("Automatic", "Manual"))
> mtcars$cyl <- factor(mtcars$cyl, levels = c(4, 6, 8), labels = c("4cyl", "6cyl", "8cyl"))


> qplot(mpg, data=mtcars, geom="density",
      fill=gear, alpha=I(.5),
      main = "Distribution of Gas Milage",
      xlab = "Miles Per Gallon",
      ylab = "Density")
```



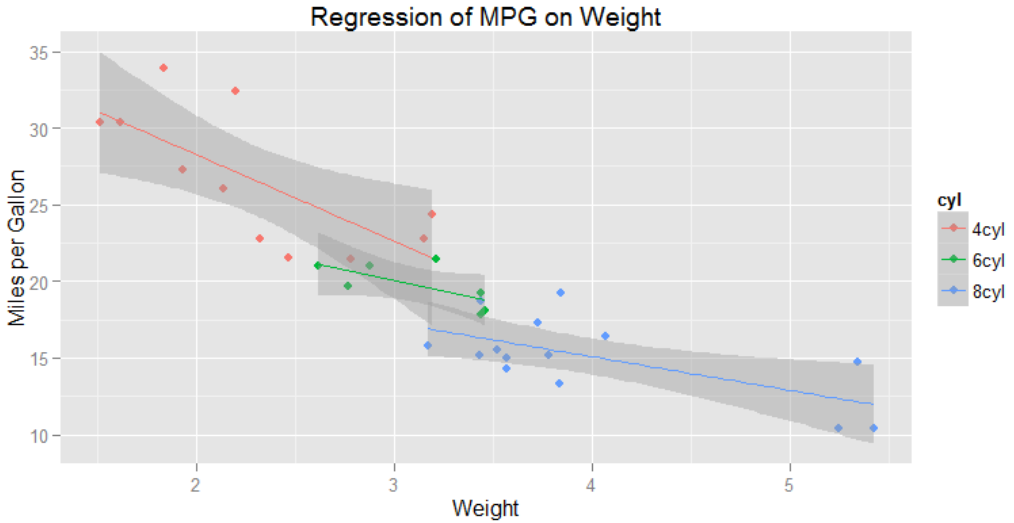
Scatterplot


```
> qplot(hp, mpg, data = mtcars, shape = am, color = am,
      facets = gear~cyl, size = I(3),
      xlab = "Horsepower", ylab = "Miles per Gallon")
```



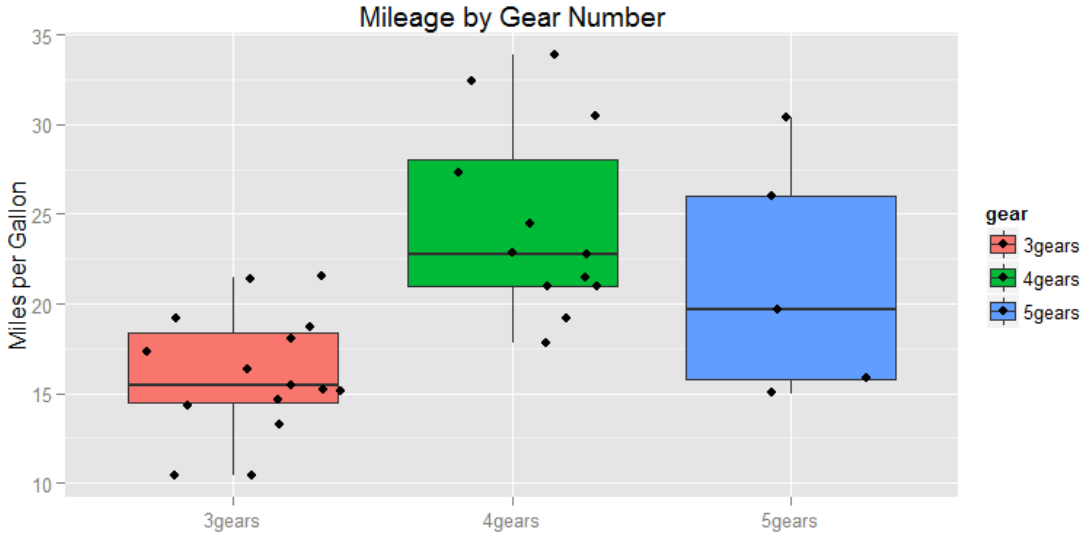
Regressions


```
> qplot(wt, mpg, data = mtcars, geom = c("point", "smooth"),
      method = "lm", formula = y~x, color = cyl,
      main = "Regression of MPG on Weight",
      xlab = "Weight", ylab = "Miles per Gallon")
```



Boxplot


```
> qplot(gear, mpg, data = mtcars, geom = c("boxplot", "jitter"),
      fill = gear, main = "Mileage by Gear Number",
      xlab = "", ylab = "Miles per Gallon")
```



Data Visualization & Graphics Environments

- Base graphics (Scatterplot, Box-and-whiskers plot, Histogram)
- Lattice
- ggplot2
- Interactive graphics in R
- Reproducibility

Interactive graphics in R

- hrsrrr

Source: http://www.statmethods.net/advgraphs/ggplot2.html

Data Visualization & Graphics Environments

- Base graphics (Scatterplot, Box-and-whiskers plot, Histogram)
- Lattice
- ggplot2
- Interactive graphics in R
- Reproducibility

Reproducibility

- Presentation from KeyNote speaker

Source: http://www.statmethods.net/advgraphs/ggplot2.html

Graphics – useful links

- GAR – Google Analytics Data [http://www.r-bloggers.com/query-your-google-analytics-data-with-the-gar-package/]
- Create interactive Powerpoint slides – [http://www.r-bloggers.com/programmatically-create-interactive-powerpoint-slides-with-r/]
- How to Change the Reference Map in Choroplethr [http://www.r-bloggers.com/how-to-change-the-reference-map-in-choroplethr/]
- Plotting regression curves with confidence intervals for LM, GLM and GLMM in R [http://www.r-bloggers.com/plotting-regression-curves-with-confidence-intervals-for-lm-glm-and-glmm-in-r/]

Thank you!

Ciprian Alexandru
alexcipro@yahoo.com