BIOS 591R - 2019 - Lecture 9: Graphics I: base graphics

author: Ziyi Li ziyi.li@emory.edu date: autosize: true font-family: 'Helvetica'

Content today

- Features and history of R graphics
- Brief overview of BASE graphics

Features of R graphics

- R has a powerful environment for visualization of scientific data
- It provides publication quality graphics, which are fully programmable
- Easily reproducible
- Full LaTeX and Sweave support
- Lots of packages and functions with built-in graphics support
- On-screen graphics
- Postscript, PDF, jpeg, png, SVG

History of R graphics

- R Graph Gallery
  - https://www.r-graph-gallery.com/network/
- R Graphic Manual and Gallery
  - https://www.imsbio.co.jp/RGM/R_image_list?page=277&init=true
- Grid Graphics - Paul Murrell
  - https://stat.ethz.ch/R-manual/R-devel/library/grid/doc/grid.pdf
- Lattic Graphics
  - http://lattice.r-forge.r-project.org/Vignettes/src/lattice-intro/lattice-intro.pdf
- ggplot2 - Tidyverse

    - https://ggplot2.tidyverse.org/

History of R graphics

R graphics can be confusing because there are no less than 4 different systems. Let's list them out here and talk about which one(s) to use.

Low-Level Capability

- Base Graphics (Has Low and High Level functions)
- Grid Graphics

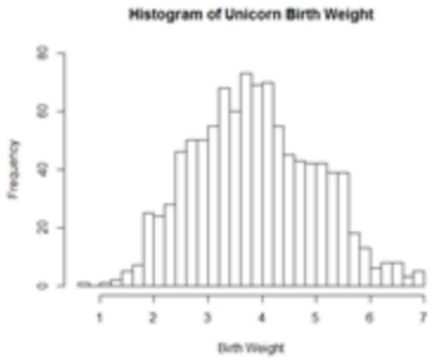
High-Level Capability

- Lattice Graphics
- ggplot2

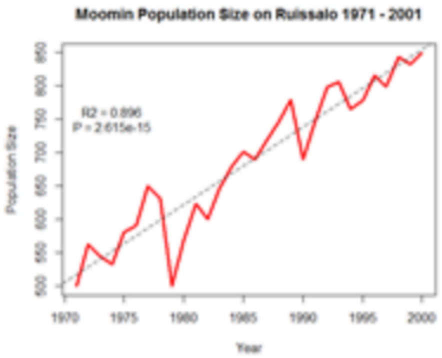
History: BASE graphics

- Oldest and most commonly used
- Uses a "pen-on-paper" model. You can only draw on top of the object. Cannot erase, modify, or delete what has already been drawn. Mimicing the thinking process.
- Has both high and low level plotting routines (unique to Base)
- Base graphics are fast.
- Lots of documentation and "google" support

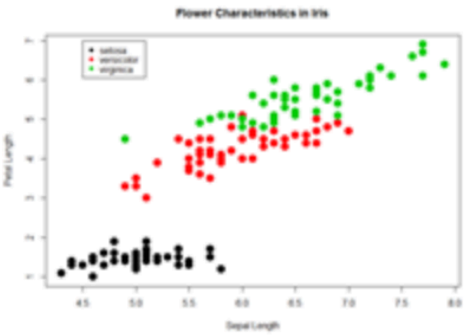
BASE graphics



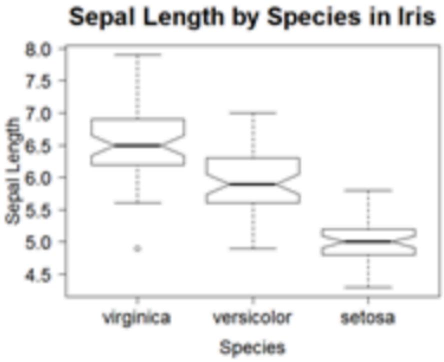
1. Basic Histogram



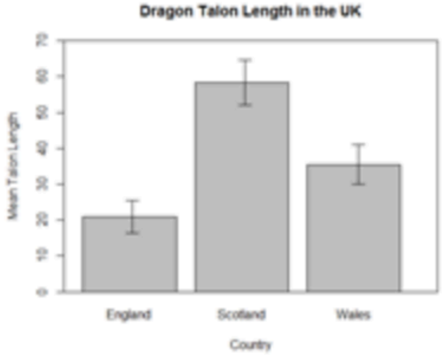
2. Line Graph with Regression



3. Scatterplot with Legend



4. Boxplot with reordered/ formatted axes

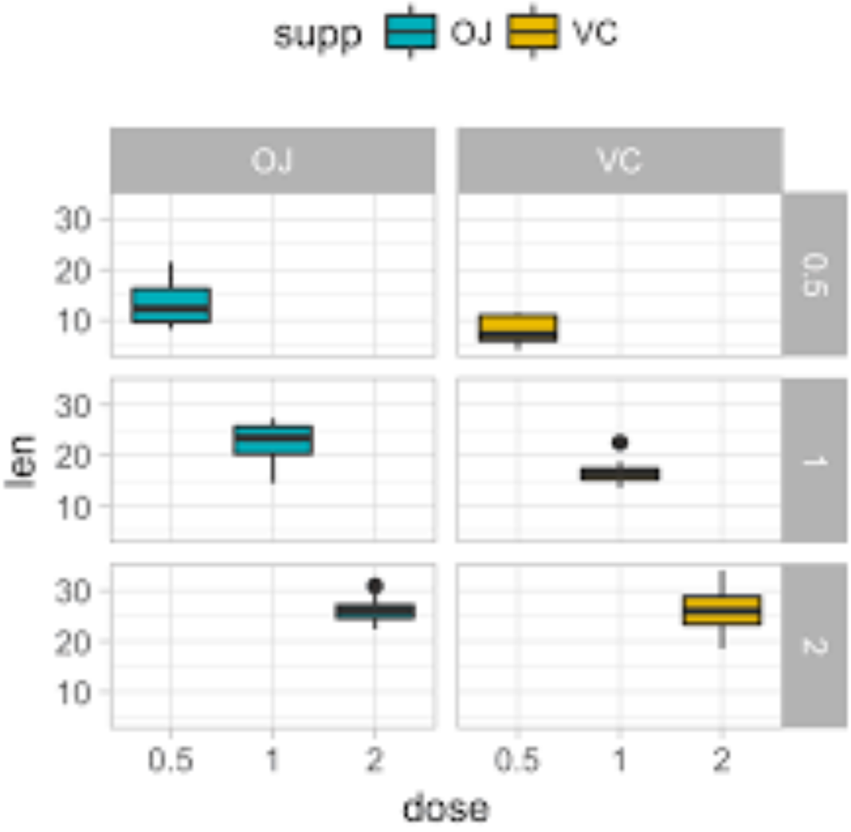


5. Boxplot with Error Bars

History: Grid graphics

- Developed in 2000 by Paul Murrell
- Provides a rich set of graphics primitives
- Uses a system of objects and view ports to make complex objects easier.
- You will almost never use this directly unless you want to do in-depth programming

Grid graphics

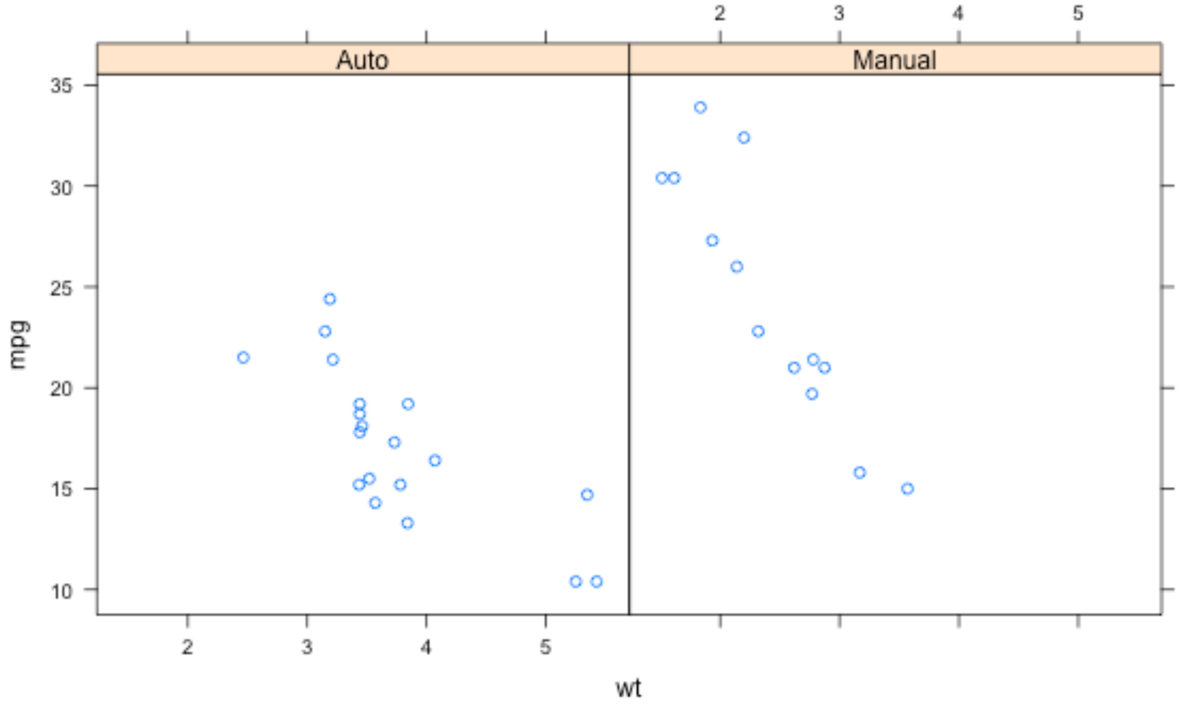


History: Lattice package

- Developed by Deepayan Sarkar to implement the trellis graphics system described in "Visualizing Data" by Cleveland.
- Independent from the BASE graphics. Include its own functions such as `xyplot` , `bwplot` , `levelplot`
- Easy to create conditioned plots with automatic creation of axes, legends, and other annotations
- Usually considered to be an improvement over Base graphics.

Lattice package

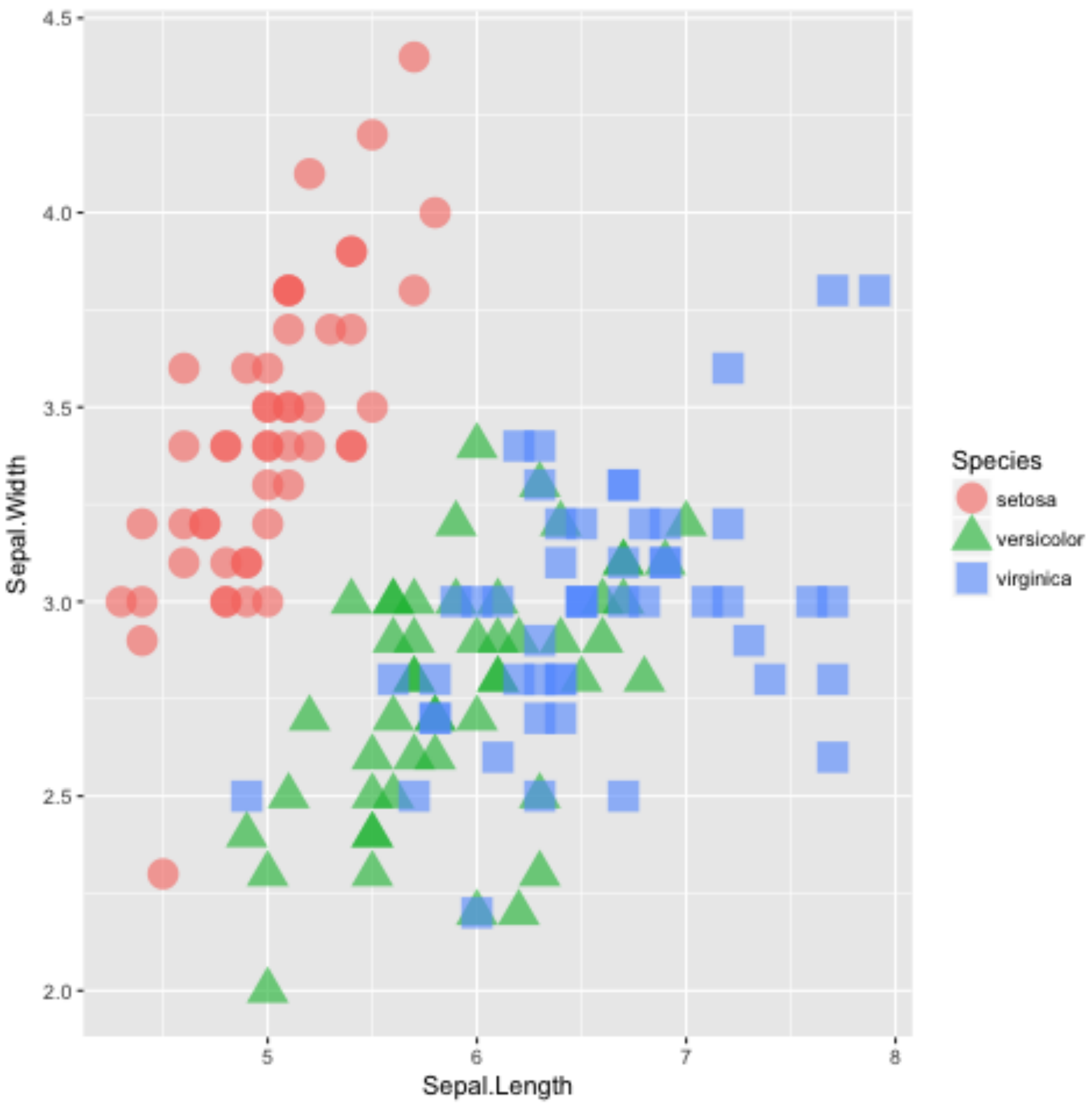
```
library(lattice)
xyplot(mpg~wt | factor(am,labels=c("Auto","Manual")), data=mtcars)
```



History: ggplot2

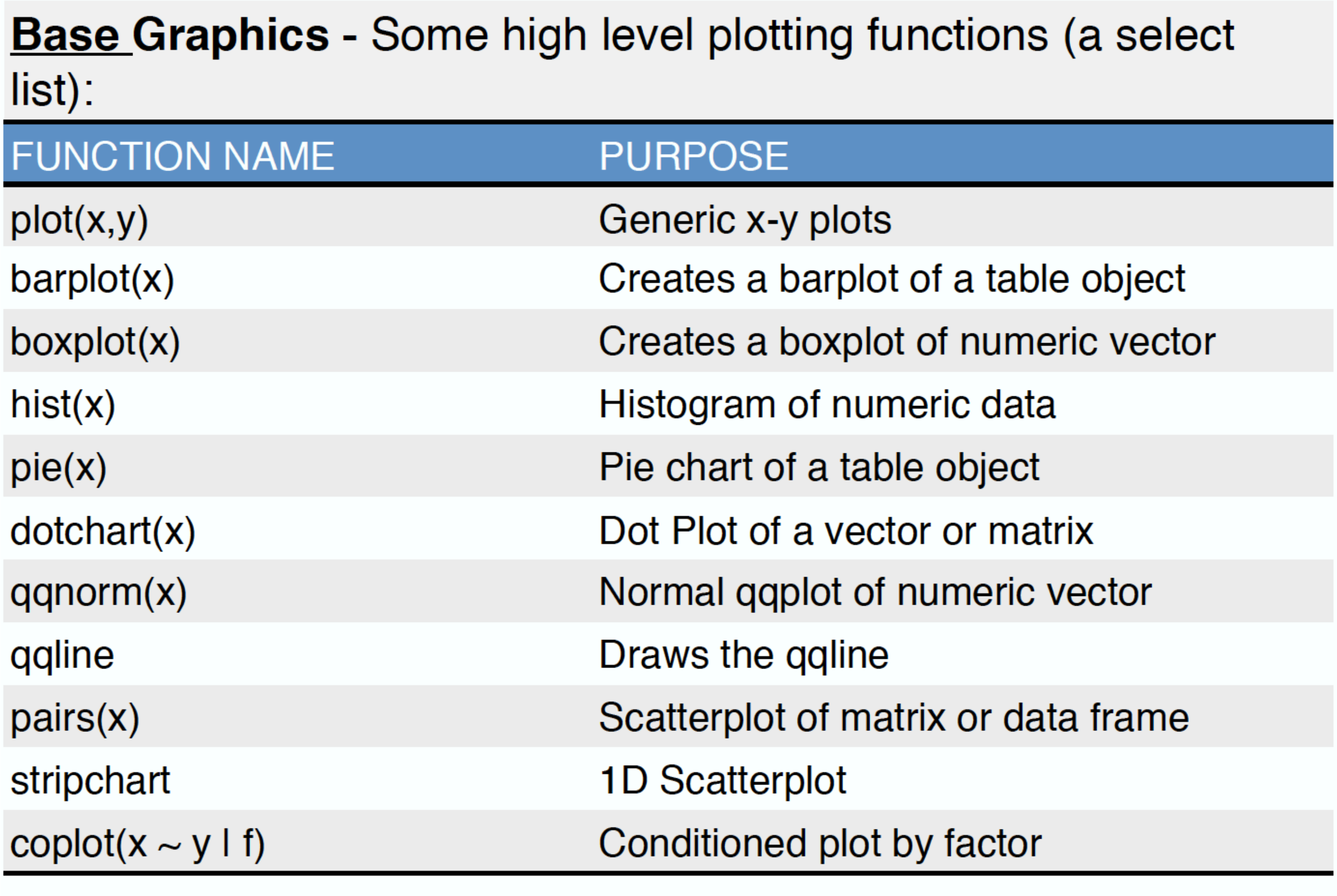
- Developed starting in 2005 by Hadley Wickham
- ggplot2 is an implementation of Leland Wilkinson's Grammar of Graphics--a general scheme for data visualization which breaks up graph into semantic components such as scales and layers.
- ggplot2 can serve as a replacement for the base graphics in R and contains a number of defaults for web and print display of common scales.
- Is said to be much slower than Base graphics but this isn't a major thing (in my opinion)
- Part of Tidyverse

ggplot2



BASE Graphics

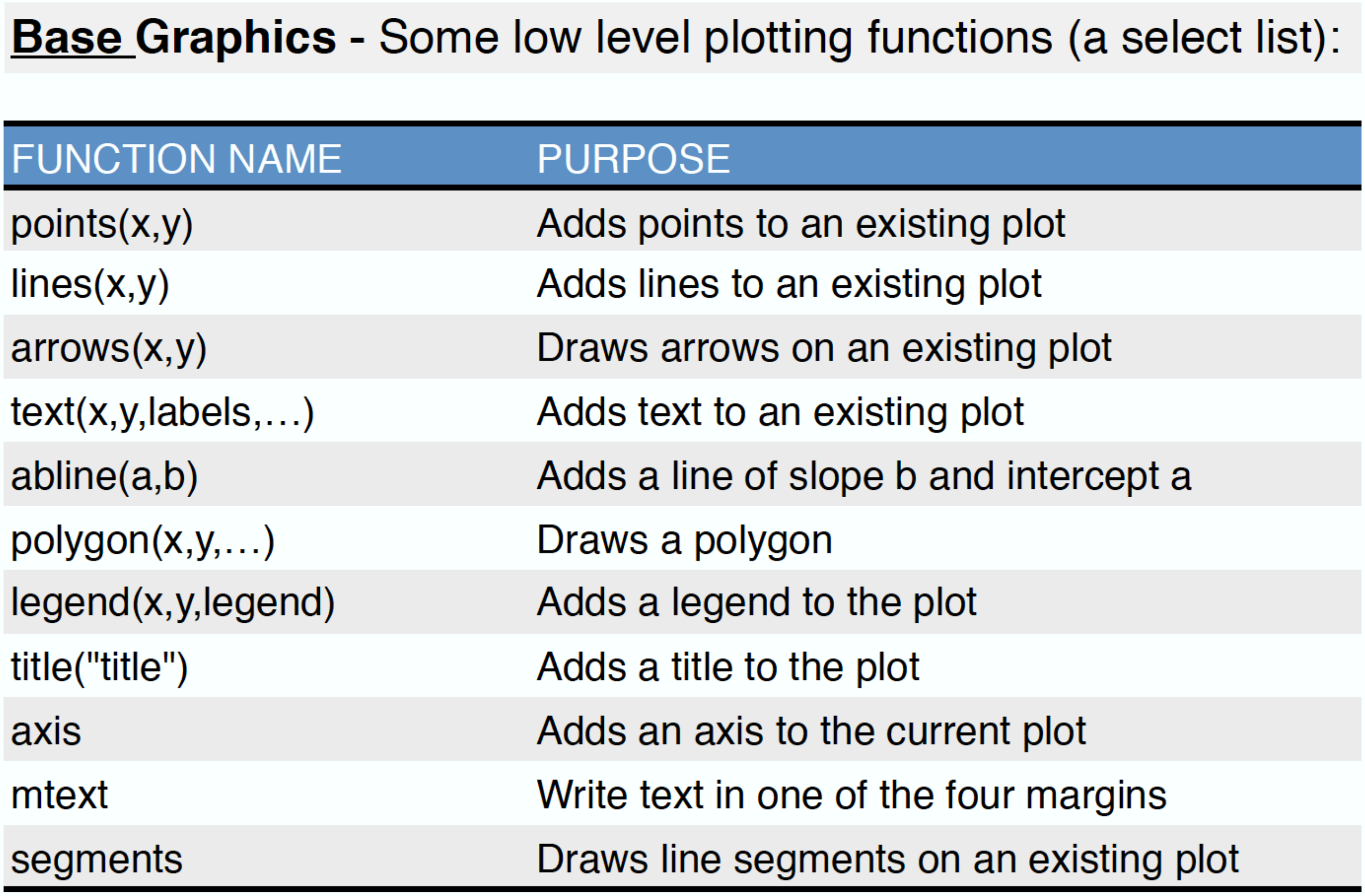
Brief overview of BASE graphics



Base Graphics - Some high level plotting functions (a select list):

FUNCTION NAME	PURPOSE
plot(x,y)	Generic x-y plots
barplot(x)	Creates a barplot of a table object
boxplot(x)	Creates a boxplot of numeric vector
hist(x)	Histogram of numeric data
pie(x)	Pie chart of a table object
dotchart(x)	Dot Plot of a vector or matrix
qqnorm(x)	Normal qqplot of numeric vector
qqline	Draws the qqline
pairs(x)	Scatterplot of matrix or data frame
stripchart	1D Scatterplot
coplot(x ~ y I f)	Conditioned plot by factor

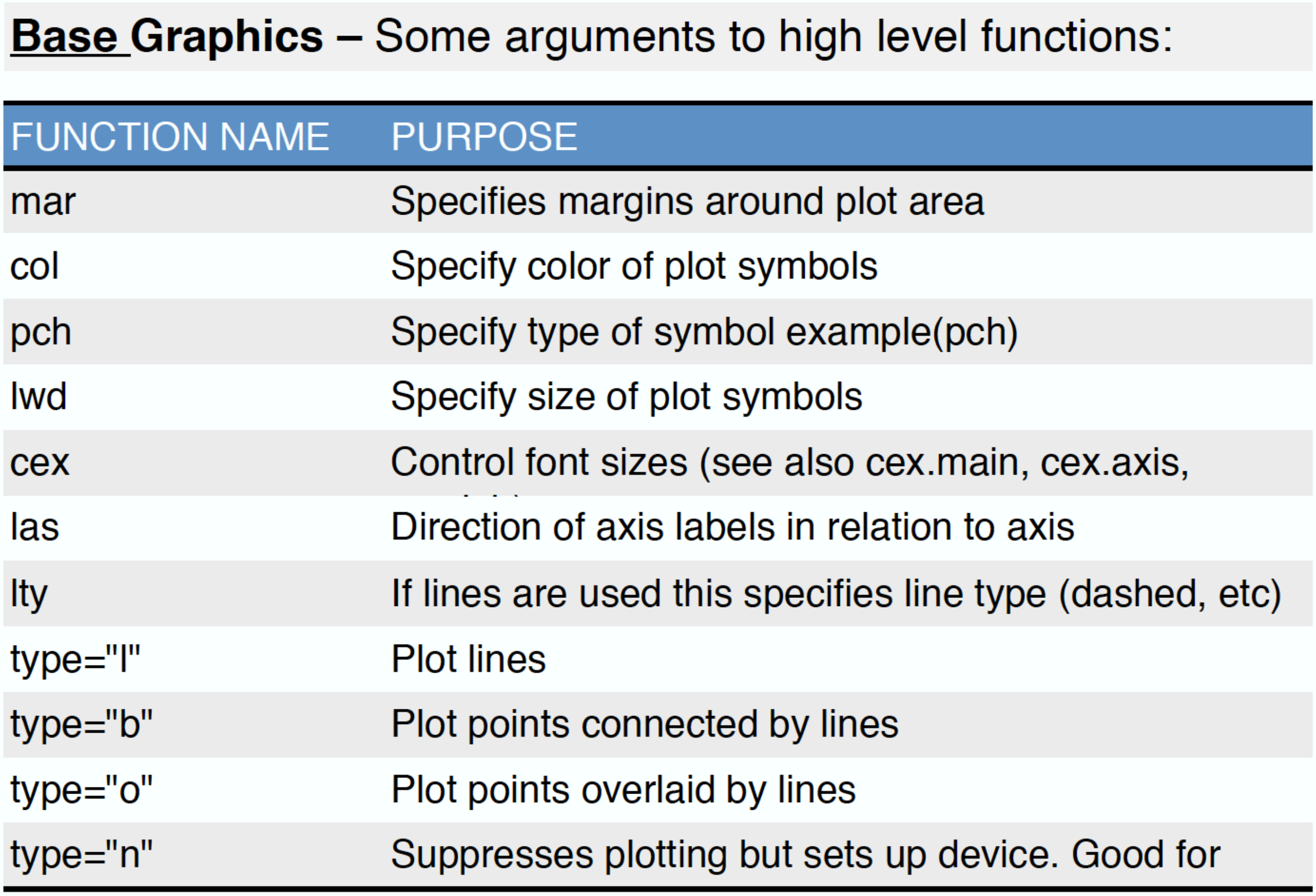
Brief overview of BASE graphics



Base Graphics - Some low level plotting functions (a select list):

FUNCTION NAME	PURPOSE
points(x,y)	Adds points to an existing plot
lines(x,y)	Adds lines to an existing plot
arrows(x,y)	Draws arrows on an existing plot
text(x,y,labels,…)	Adds text to an existing plot
abline(a,b)	Adds a line of slope b and intercept a
polygon(x,y,…)	Draws a polygon
legend(x,y,legend)	Adds a legend to the plot
title("title")	Adds a title to the plot
axis	Adds an axis to the current plot
mtext	Write text in one of the four margins
segments	Draws line segments on an existing plot

Brief overview of BASE graphics



Base Graphics – Some arguments to high level functions:

FUNCTION NAME	PURPOSE
mar	Specifies margins around plot area
col	Specify color of plot symbols
pch	Specify type of symbol example(pch)
lwd	Specify size of plot symbols
cex	Control font sizes (see also cex.main, cex.axis,
las	Direction of axis labels in relation to axis
lty	If lines are used this specifies line type (dashed, etc)
type="l"	Plot lines
type="b"	Plot points connected by lines
type="o"	Plot points overlaid by lines
type="n"	Suppresses plotting but sets up device. Good for

Brief overview of BASE graphics

Base Graphics – Some arguments to high level functions:

FUNCTION NAME	PURPOSE
add=TRUE	Adds a new plot on top of another (kind of)
axes=FALSE	Suppresses axis creation – you then make your
xlab="STRING"	Makes the X label
ylab="STRING"	Makes the y-label
main="STRING"	Gives the plot a main title
sub="STRING"	Gives a subtitle
type="p"	Plot individual points
type="l"	Plot lines
type="b"	Plot points connected by lines
type="o"	Plot points overlaid by lines
type="n"	Suppresses plotting but sets up device. Good for

Brief overview of BASE graphics

You can save your on-screen graphics to a popular file type for use within a program. You can always do screen grabs too.

FUNCTION	RESULT OUTPUT
`pdf("file.pdf")`	Creates a PDF file called "file.pdf"
`png("file.png")`	Createa a PNG file
`jpeg("file.jpg")`	Creates a JPG file
`bmp(""file.bmp")`	Creates a BMP file
`postscript("file.ps")`	Creates a Postscript file
`win.meta("file.wmf")`	Creates a Windows meta file

```
> png("mytest.png")

> plot(mtcats$mpg)  # Simple, but you get the point

> dev.off()
```