par exercises

Table of Contents

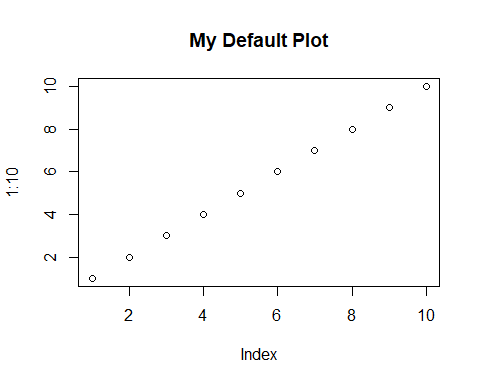
This document accompanies the [LinkedIn Learning Video](http://niemannross.com/link/mnratlil).

par is short for “Graphical Parameters” They get or set parameters before the plot or are passed through the plot command to par.

Here is a list of all the par commands and their effects.

First, here’s a normal baseline plot for comparison

# devnull captures the unneeded return value from plot  
devnull <- plot(1:10)+  
title( main = "My Default Plot")

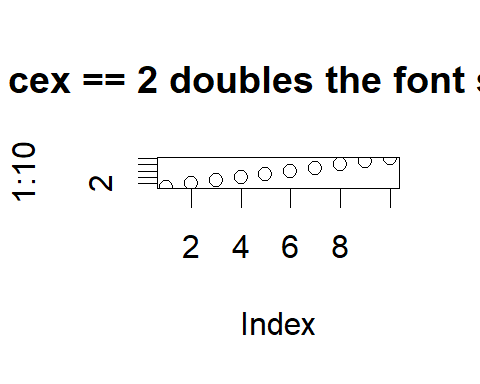


# Text - size

## cex

Font size relative to default value of 1

oldParValues <- par(no.readonly = TRUE)  
  
par(cex = 2)  
  
devnull <- plot(1:10)+  
title( main = "cex == 2 doubles the font size")

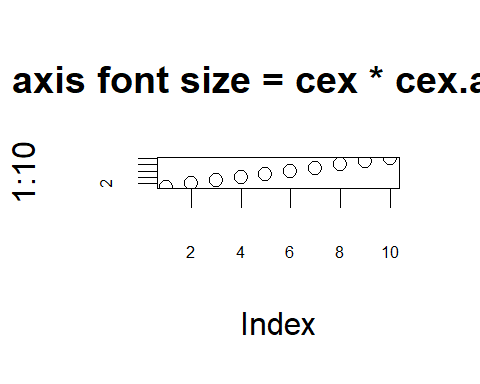


par(oldParValues)

## cex.axis

Axis font size relative to cex

oldParValues <- par(no.readonly = TRUE)  
  
par(cex = 2, cex.axis = .5)  
  
devnull <- plot(1:10)+  
title( main = "axis font size = cex \* cex.axis")

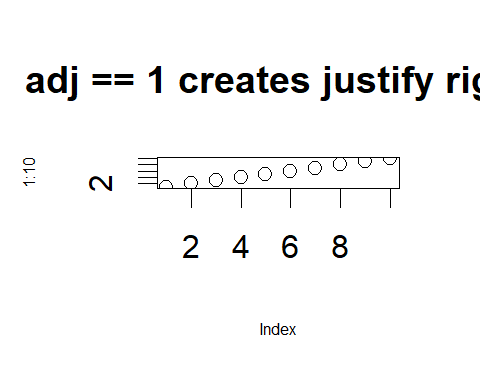


par(oldParValues)

## cex.lab

label font size relative to cex

oldParValues <- par(no.readonly = TRUE)  
  
par(cex = 2, cex.lab = .5)  
  
devnull <- plot(1:10)+  
title( main = "adj == 1 creates justify right")

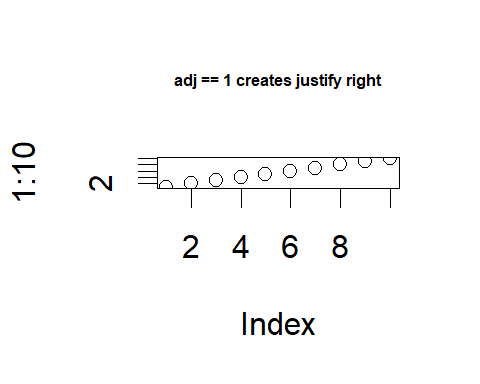


par(oldParValues)

## cex.main

main title font size relative to cex

oldParValues <- par(no.readonly = TRUE)  
  
par(cex = 2, cex.main = .5)  
  
devnull <- plot(1:10)+  
title( main = "adj == 1 creates justify right")

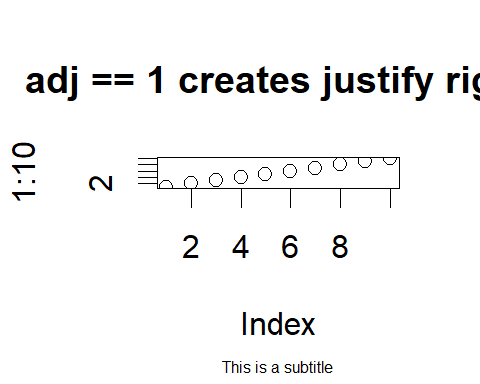


par(oldParValues)

## cex.sub

sub-title font size relative to cex

oldParValues <- par(no.readonly = TRUE)  
  
par(cex = 2, cex.sub = .5)  
  
devnull <- plot(1:10)+  
title( main = "adj == 1 creates justify right",  
 sub = "This is a subtitle")

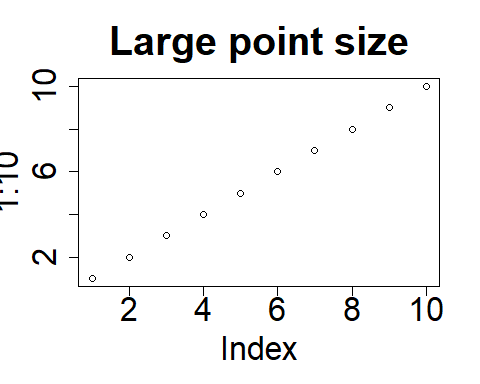


par(oldParValues)

## ps

Character size in points. Varies from device to device, but probably multiples of 1/72 of an inch

oldParValues <- par(no.readonly = TRUE)  
  
par( ps = 25)  
  
devnull <- plot(1:10, main = "Large point size")



par(oldParValues)

## cin

Character size (width, height) in inches. Read only.

par("cin")

## [1] 0.15 0.20

## cra

Character size (width, height) in pixels. Possibly 1/72 of an inch, depending on the device. read only

par("cra")

## [1] 14.4 19.2

## csi

Character height in inches. read only

par("csi")

## [1] 0.2

# aka...  
par("cin")[2]

## [1] 0.2

## cxy

Character size (width, height) in user coordinates. read only

par("cxy")

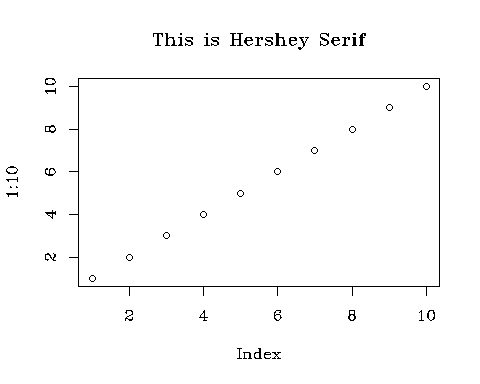
## [1] 0.03989363 0.09259264

# Text - Font

## family

font family by name. "" is default. Values include “serif”, “sans”, “mono”, and Hershey fonts. Use “?hershey” for details on the hershey font family.

oldParValues <- par(no.readonly = TRUE)  
  
par(family = "HersheySerif")  
  
devnull <- plot(1:10)+  
title( main = "This is Hershey Serif")

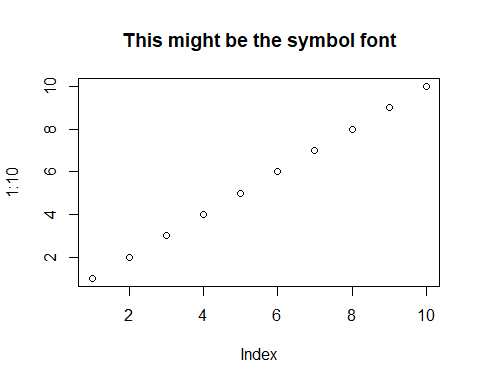


par(oldParValues)

## font

1 = plain 2 = bold 3 = italic 4 = bold italic 5 = symbol This is dicey, depending on the device and operating system.

oldParValues <- par(no.readonly = TRUE)  
  
par( font = 5)  
  
devnull <- plot(1:10)+  
title( main = "This might be the symbol font")

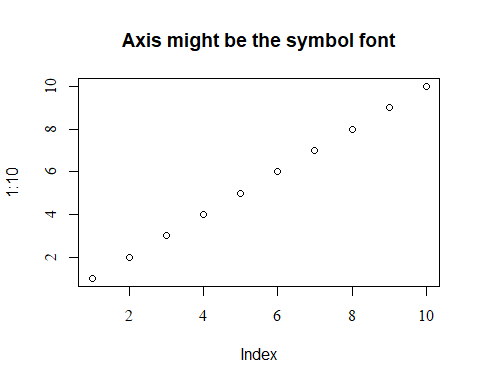


par(oldParValues)

## font.axis

The font to be used for axis annotation. Dependent on the operating system

oldParValues <- par(no.readonly = TRUE)  
  
par( font.axis = 5)  
  
devnull <- plot(1:10)+  
title( main = "Axis might be the symbol font")

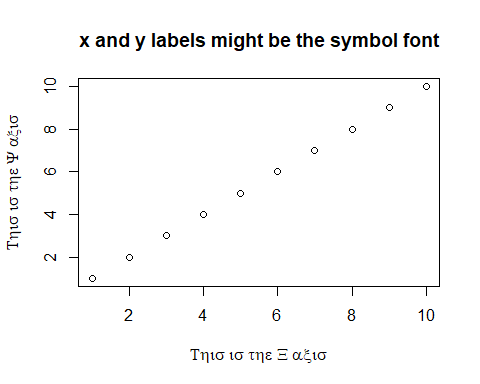


par(oldParValues)

## font.lab

The font to be used for x and y labels. Dependent on the operating system

oldParValues <- par(no.readonly = TRUE)  
  
par( font.lab = 5)  
  
plot(1:10, xlab = "This is the X axis", ylab = "This is the Y axis", main = "x and y labels might be the symbol font")

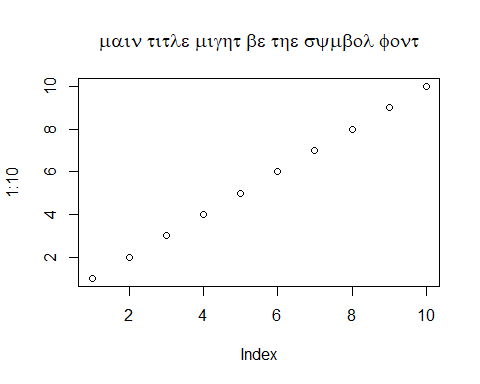


par(oldParValues)

## font.main

The font to be used for plot main titles.

oldParValues <- par(no.readonly = TRUE)  
  
par( font.main = 5)  
  
devnull <- plot(1:10)+  
title( main = "main title might be the symbol font")

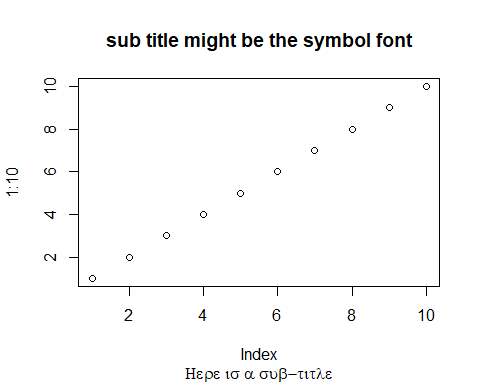


par(oldParValues)

## font.sub

The font to be used for plot sub-titles.

oldParValues <- par(no.readonly = TRUE)  
  
par( font.sub = 5)  
  
devnull <- plot(1:10)+  
title( main = "sub title might be the symbol font",  
 sub = "Here is a sub-title")



par(oldParValues)

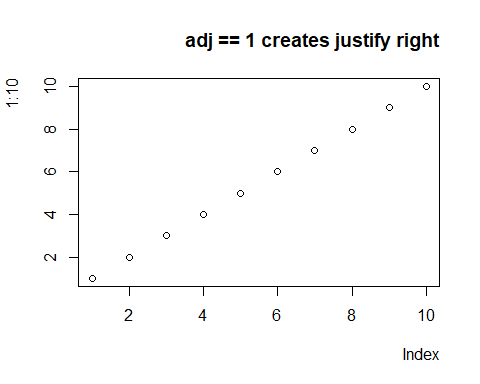
# Text - justification & rotation

## adj

Text Justification. “0” is left, “0.5” is centered, 1 is right

Affects text(), mtext(), and title( main = )

oldParValues <- par(no.readonly = TRUE)  
  
par(adj = 1)  
  
devnull <- plot(1:10)+  
title( main = "adj == 1 creates justify right")

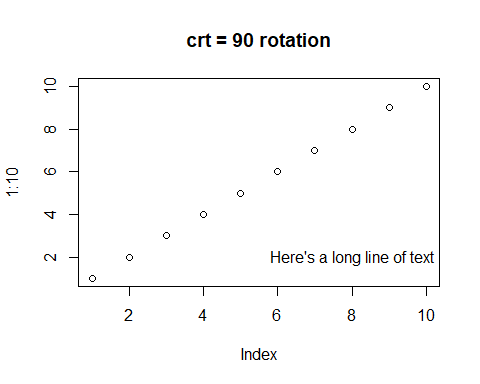


par(oldParValues) # this resets par to default values

## crt

Text Rotation in degrees. Probably only 90 degrees, depending on your device. Possibly not at all, depending on your device. Also see srt for string rotation

oldParValues <- par(no.readonly = TRUE)  
  
par(crt = 90)  
  
devnull <- plot(1:10)+  
title( main = "crt = 90 rotation")+  
text(x = 8, y = 2, labels = "Here's a long line of text")

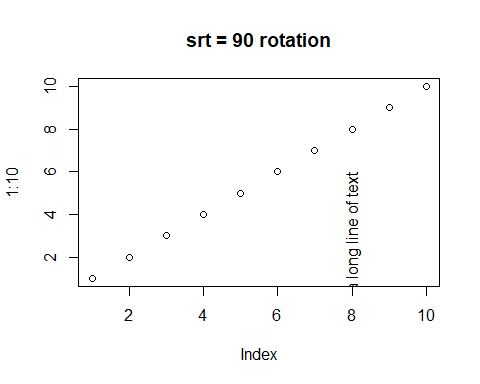


par(oldParValues)

## srt

The string rotation in degrees. Only supported by text.

oldParValues <- par(no.readonly = TRUE)  
  
par(srt = 90)  
  
devnull <- plot(1:10)+  
title( main = "srt = 90 rotation")+  
text(x = 8, y = 2, labels = "Here's a long line of text")

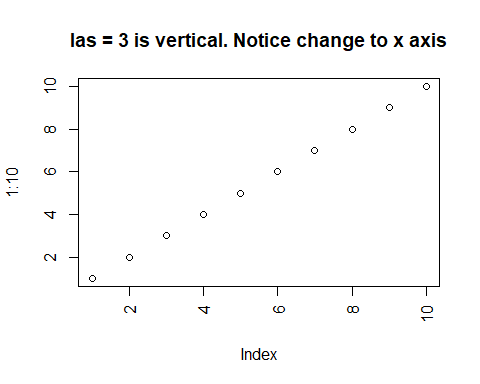


par(oldParValues)

## las

Axis label rotation. 0 = parallel (default), 1 = horizontal, 2 = perpendicular, 3 = vertical

oldParValues <- par(no.readonly = TRUE)  
  
par(las = 3)  
  
devnull <- plot(1:10)+  
title( main = "las = 3 is vertical. Notice change to x axis")



par(oldParValues)

# Axis control

## xaxt & yaxt

“n” suppresses plotting of the x or y axis.

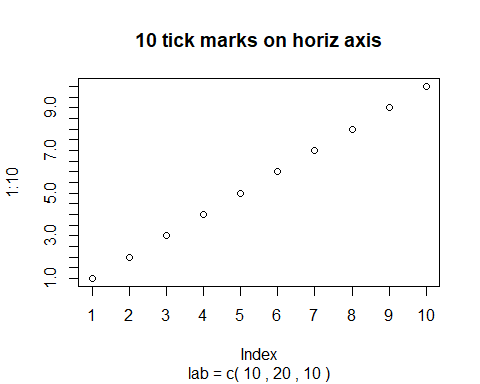
## xlog & ylog

If TRUE, a logarithmic scale is in use (e.g., after plot(\*, log = “x”)). For a new device, it defaults to FALSE, i.e., linear scale.

## lab

Axis labels. x and y are tick marks. len is the label length. Default is c(5,5,7)

oldParValues <- par(no.readonly = TRUE)  
  
xTickMarks <- 10  
yTickMarks <- 20  
labLength <- 10  
  
par(lab = c(xTickMarks, yTickMarks, labLength))  
  
devnull <- plot(1:10)+  
title( main = paste(xTickMarks,"tick marks on horiz axis"),  
 sub = paste("lab = c(",xTickMarks,",",yTickMarks,",",labLength,")"))

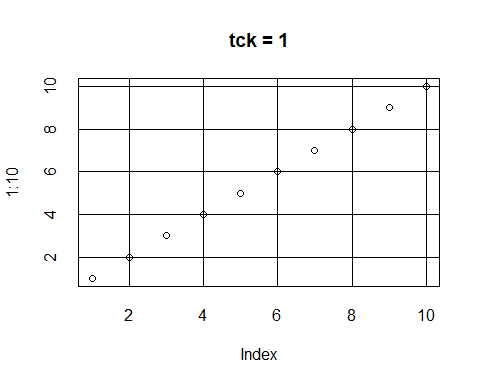


par(oldParValues)

## tck

The length of tick marks as a fraction of the smaller of the width or height of the plotting region. If tck >= 0.5 it is interpreted as a fraction of the relevant side, so if tck = 1 grid lines are drawn. The default setting (tck = NA) is to use tcl = -0.5.

oldParValues <- par(no.readonly = TRUE)  
  
par(tck = 1)  
  
devnull <- plot(1:10, main = "tck = 1")

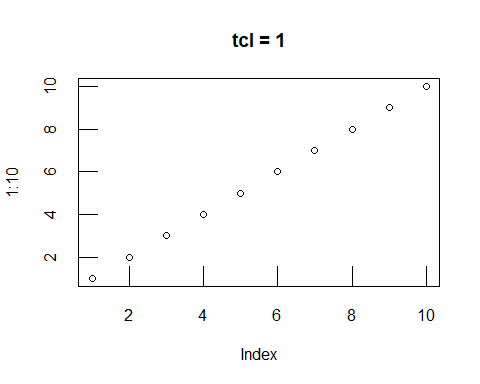


par(oldParValues)

## tcl

The length of tick marks as a fraction of the height of a line of text. The default value is -0.5; setting tcl = NA sets tck = -0.01 which is S’ default.

oldParValues <- par(no.readonly = TRUE)  
  
par(tcl = 1)  
  
devnull <- plot(1:10, main = "tcl = 1")



par(oldParValues)

# Decoration

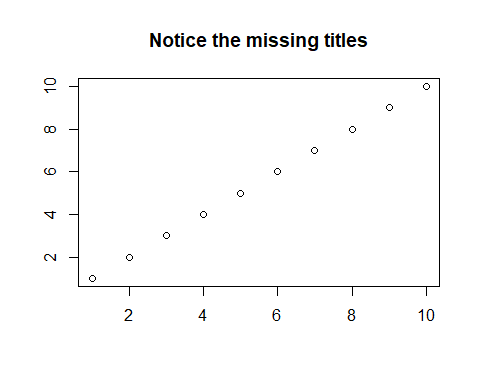
## pch

The character used to plot a point in a graph. Refer to “?points” for a lengthy discussion.

## ann

annotation on or off. Defaults to on (TRUE)

oldParValues <- par(no.readonly = TRUE)  
  
par(ann = FALSE)  
  
devnull <- plot(1:10)+  
title( main = "Notice the missing titles")

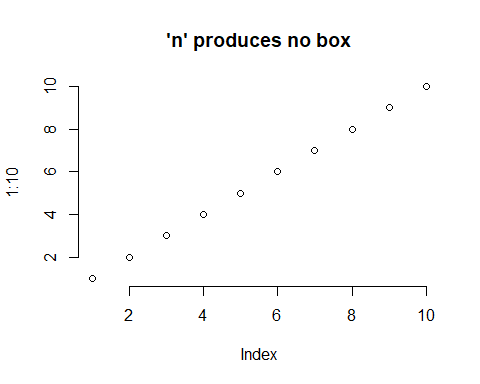


par(oldParValues)

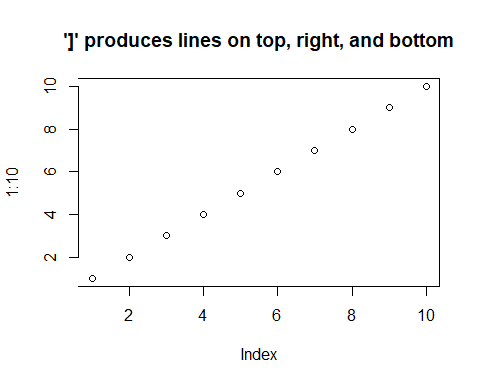
## bty

Box Type - the box surrounding the plot. o - default l - lines on left and bottom 7 - lines on top and right c - lines on top, left, and right u - lines on left, bottom, and right ] - lines on top, right, and bottom n - no box

oldParValues <- par(no.readonly = TRUE)  
  
par(bty = "n")  
  
devnull <- plot(1:10)+  
title( main = "'n' produces no box")



par(bty = "]")  
  
devnull <- plot(1:10)+  
title( main = "']' produces lines on top, right, and bottom")



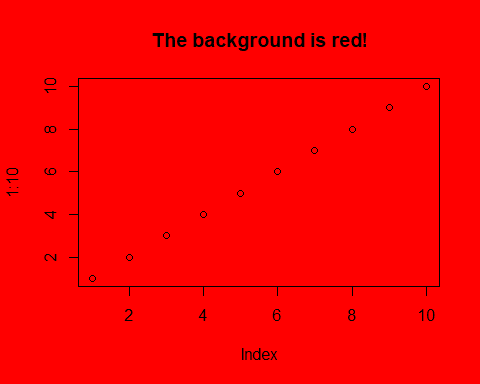
par(oldParValues)

# Color

## bg

background color

oldParValues <- par(no.readonly = TRUE)  
  
par(bg = "Red")  
  
devnull <- plot(1:10)+  
title( main = "The background is red!")

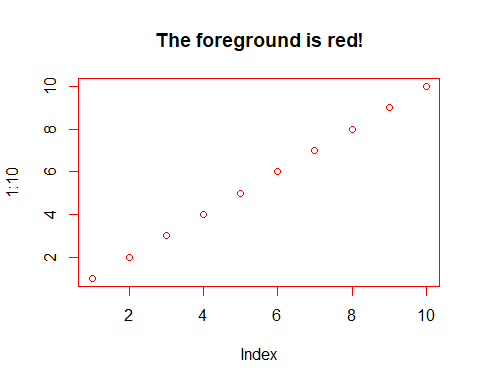


par(oldParValues)

## fg

foreground color

oldParValues <- par(no.readonly = TRUE)  
  
par(fg = "Red")  
  
devnull <- plot(1:10)+  
title( main = "The foreground is red!")

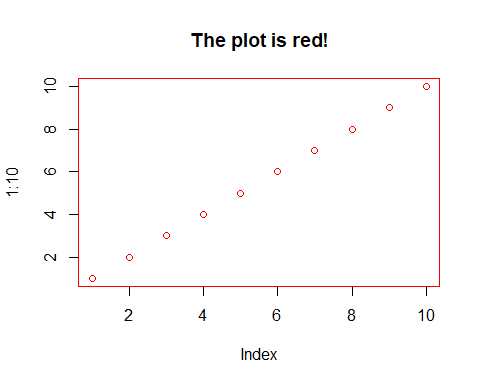


par(oldParValues)

## col

default plotting color

oldParValues <- par(no.readonly = TRUE)  
  
par(col = "Red")  
  
devnull <- plot(1:10)+  
title( main = "The plot is red!")

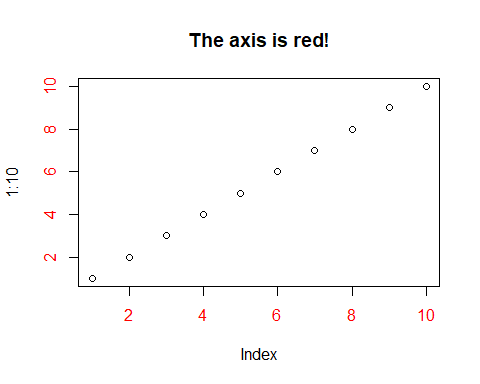


par(oldParValues)

## col.axis

The color to be used for axis annotation. Defaults to “black”.

oldParValues <- par(no.readonly = TRUE)  
  
par(col.axis = "Red")  
  
devnull <- plot(1:10)+  
title( main = "The axis is red!")

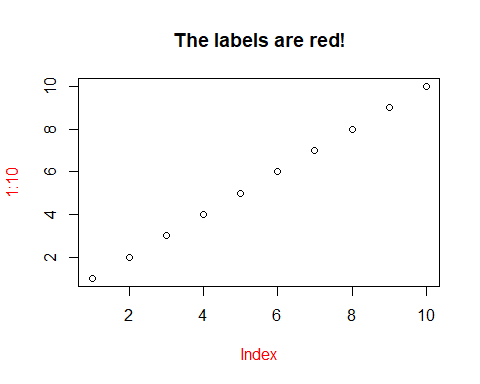


par(oldParValues)

## col.lab

The color to be used for x and y labels. Defaults to “black”.

oldParValues <- par(no.readonly = TRUE)  
  
par(col.lab = "Red")  
  
devnull <- plot(1:10)+  
title( main = "The labels are red!")

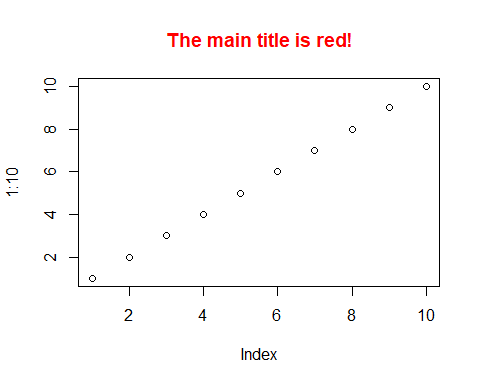


par(oldParValues)

## col.main

The color to be used for plot main titles. Defaults to “black”.

oldParValues <- par(no.readonly = TRUE)  
  
par(col.main = "Red")  
  
devnull <- plot(1:10)+  
title( main = "The main title is red!")

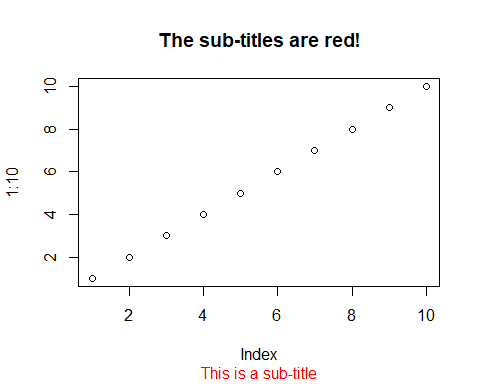


par(oldParValues)

## col.sub

The color to be used for plot sub-titles. Defaults to “black”.

oldParValues <- par(no.readonly = TRUE)  
  
par(col.sub = "Red")  
  
devnull <- plot(1:10)+  
title( main = "The sub-titles are red!",  
 sub = "This is a sub-title")



par(oldParValues)

# Device information

## din

Width and height of device in inches. dev.size is similiar, but dynamic when window is resized.

par("din")

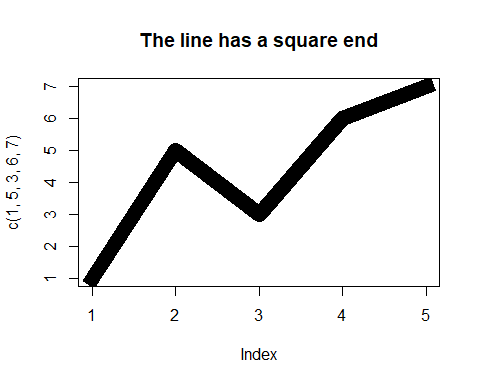
## [1] 4.999999 3.999999

# lines

## lend

The line end (line cap) style. 0 = round (default), 1 = butt, 2 = square.

oldParValues <- par(no.readonly = TRUE)  
  
par(lend = 2)  
  
devnull <- plot(c(1,5,3,6,7), type = "l", lwd = "15")+  
title( main = "The line has a square end")

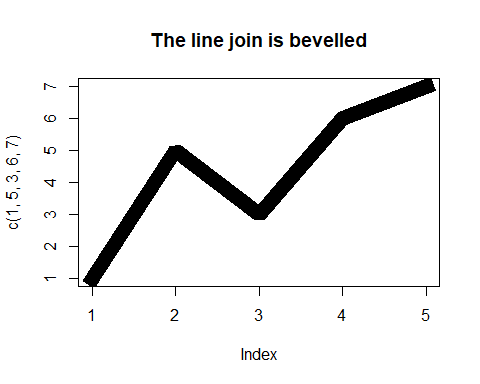


par(oldParValues)

## ljoin

The line join style. This can be specified as an integer or string. 0 and “round” = rounded line joins [default]. 1 and “mitre” = mitred. 2 and “bevel” = bevelled.

oldParValues <- par(no.readonly = TRUE)  
  
par(ljoin = 2)  
  
devnull <- plot(c(1,5,3,6,7), type = "l", lend = "square", lwd = "15")+  
title( main = "The line join is bevelled")

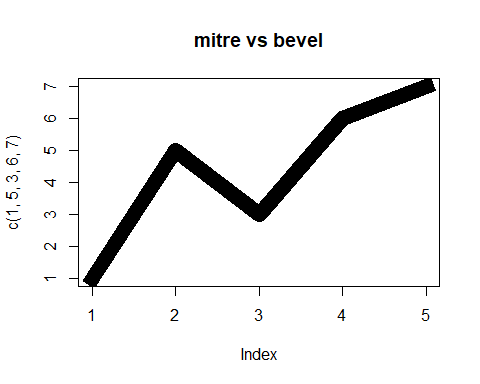


par(oldParValues)

## lmitre

The line mitre limit. This controls when mitred line joins are automatically converted into bevelled line joins. The value must be larger than 1 and the default is 10. Not all devices will honour this setting.

oldParValues <- par(no.readonly = TRUE)  
  
par(lmitre = 40)  
  
devnull <- plot(c(1,5,3,6,7), type = "l", lend = "square", lwd = "15")+  
title( main = "mitre vs bevel")

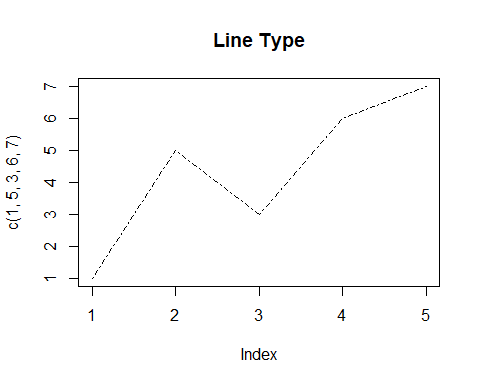


par(oldParValues)

## lty

The line type, specified as integer/string (choose one). 0/blank, 1/solid (default), 2/dashed, 3/dotted, 4/dotdash, 5/longdash, 6/twodash

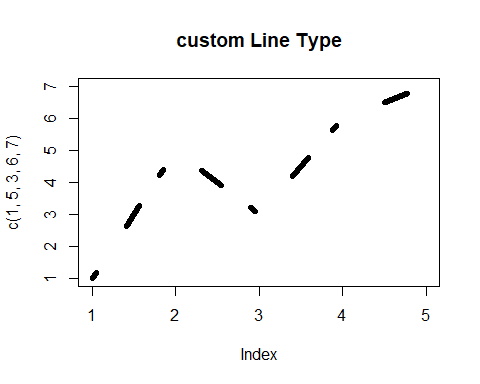
oldParValues <- par(no.readonly = TRUE)  
  
devnull <- plot(c(1,5,3,6,7), type = "l", lty = "dotdash")+  
title( main = "Line Type")



par(oldParValues)

or…do it yourself by specifying up to 8 characters - 1:9, or A:F

oldParValues <- par(no.readonly = TRUE)  
  
devnull <- plot(c(1,5,3,6,7), type = "l", lty = "1946", lwd = "6")+  
title( main = "custom Line Type")

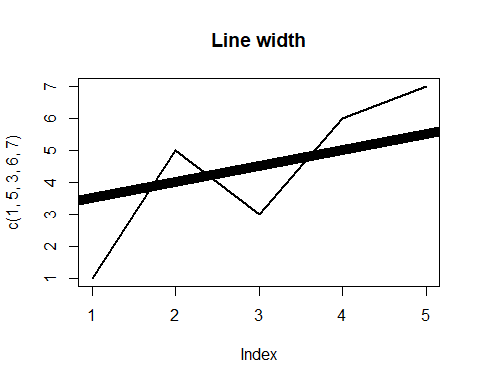


par(oldParValues)

## lwd

The line width. Default = 1

oldParValues <- par(no.readonly = TRUE)  
  
devnull <- plot(c(1,5,3,6,7), type = "l", lwd = "2")+  
title( main = "Line width")+  
 abline(c(3,.5), lwd = 10)



par(oldParValues)

# Layout

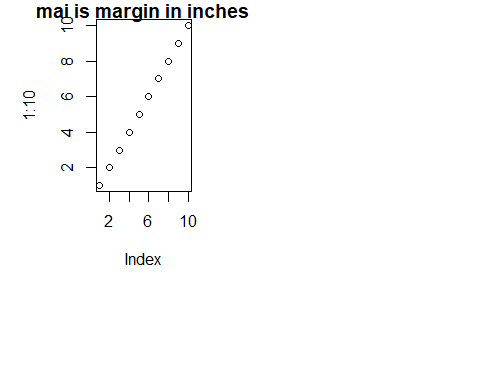
## mai

margin (in inches). Format is c(bottom, left, top, right)

oldParValues <- par(no.readonly = TRUE)  
  
par("mai") # what is the default margin size in inches?

## [1] 1.02 0.82 0.82 0.42

par(mai = c(2,1,.2,3))  
  
devnull <- plot(1:10)+  
title( main = "mai is margin in inches")



par(oldParValues) # this resets par to default values

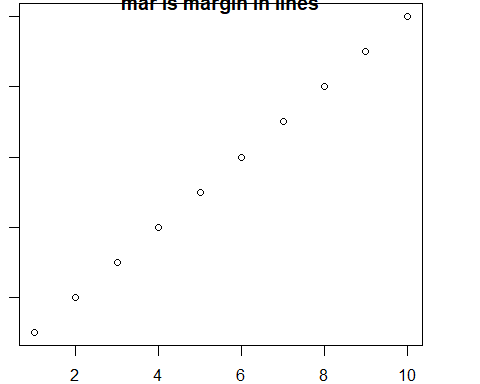
## mar

margin (in lines)

oldParValues <- par(no.readonly = TRUE)  
  
par("mar") # what is the default margin size in lines?

## [1] 5.1 4.1 4.1 2.1

par(mar = c(2,1,.2,3))  
  
devnull <- plot(1:10)+  
title( main = "mar is margin in lines")



par(oldParValues) # this resets par to default values

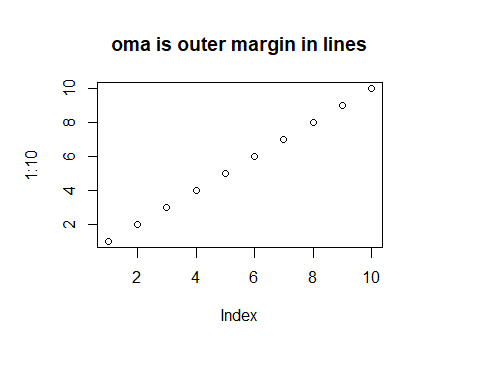
## oma

The outer margin in lines of text.

oldParValues <- par(no.readonly = TRUE)  
  
par("oma") # what is the default outer margin size in lines?

## [1] 0 0 0 0

par(oma = c(2,1,.2,3))  
  
devnull <- plot(1:10, main = "oma is outer margin in lines")



par(oldParValues) # this resets par to default values

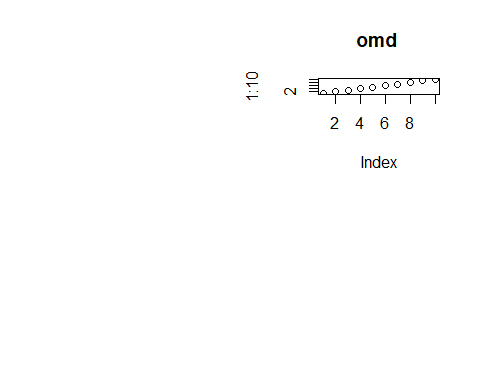
## omd

The region inside outer margins of a plot, relative to device region. The values are c(x1, x2, y1, y2). So the bottom left is (x1,y1). Top right is (x2, y2)

oldParValues <- par(no.readonly = TRUE)  
  
par("omd")

## [1] 0 1 0 1

par(omd = c(.5,1,.5,1))  
devnull <- plot(1:10, main = "omd ")



par(oldParValues) # this resets par to default values

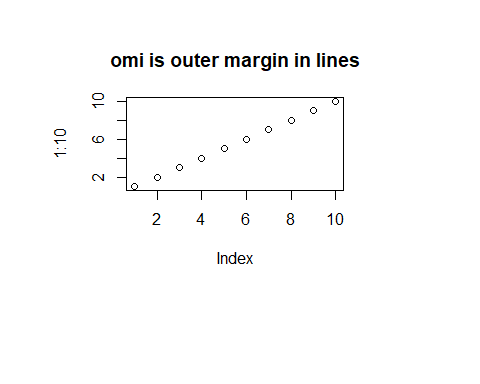
## omi

A vector of the form c(bottom, left, top, right) giving the size of the outer margins in inches.

oldParValues <- par(no.readonly = TRUE)  
  
par("omi") # what is the default outer margin size in lines?

## [1] 0 0 0 0

par(omi = c(1,.5,.2,1))  
  
devnull <- plot(1:10, main = "omi is outer margin in lines")



par(oldParValues) # this resets par to default values

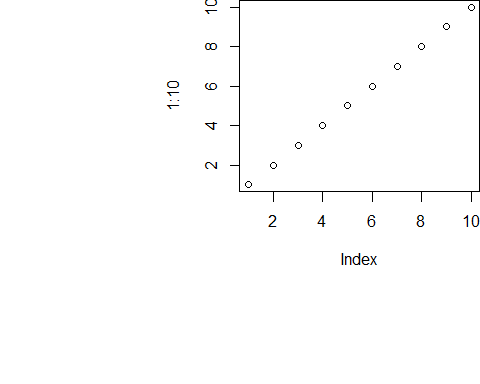
## usr

A vector of the form c(x1, x2, y1, y2) giving the extremes of the user coordinates of the plotting region. When a logarithmic scale is in use (i.e., par(“xlog”) is true, see below), then the x-limits will be 10 ^ par(“usr”)[1:2]. Similarly for the y-axis.

oldParValues <- par(no.readonly = TRUE)  
  
par("usr")

## [1] 0 1 0 1

par(plt = c(.5,1,.5,1))  
  
devnull <- plot(1:10, main = "par(plt = c(.5,1,.5,1))")



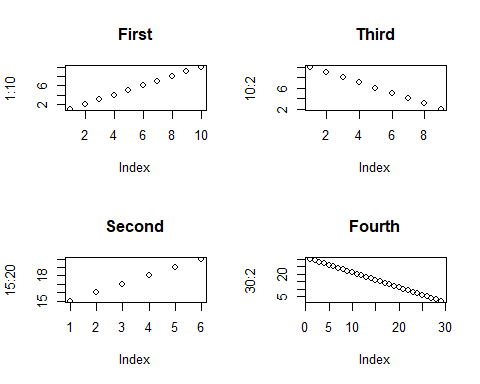
par(oldParValues) # this resets par to default values

## mfcol, mfrow

Causes plots to be drawn on a grid. Grid size is specified as c(rows,columns). Set mfcol to draw columns first. Set mfrow to draw rows first. Also look at the functions “layout” and “split.screen”.

For example: fill the columns first, then the rows

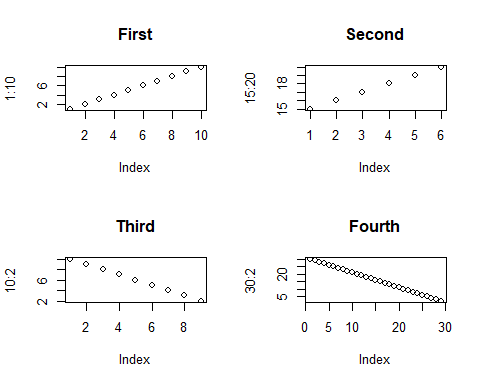
oldParValues <- par(no.readonly = TRUE)  
  
par(mfcol = c(2,2))  
  
devnull <- plot(1:10, main = "First")  
devnull <- plot(15:20, main = "Second")  
devnull <- plot(10:2, main = "Third")  
devnull <- plot(30:2, main = "Fourth")



par(oldParValues) # this resets par to default values

Second, fill the rows first, then columns

oldParValues <- par(no.readonly = TRUE)  
  
par(mfrow = c(2,2))  
  
devnull <- plot(1:10, main = "First")  
devnull <- plot(15:20, main = "Second")  
devnull <- plot(10:2, main = "Third")  
devnull <- plot(30:2, main = "Fourth")



par(oldParValues) # this resets par to default values

## mfg

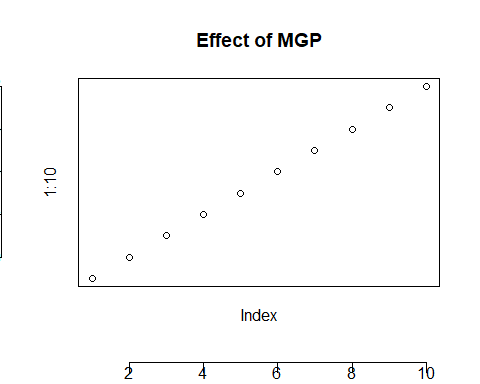
A numerical vector of the form c(i, j) where i and j indicate which figure in an array of figures is to be drawn next (if setting) or is being drawn (if enquiring). The array must already have been set by mfcol or mfrow.

For compatibility with S, the form c(i, j, nr, nc) is also accepted, when nr and nc should be the current number of rows and number of columns. Mismatches will be ignored, with a warning.

## mgp

Margins in mex units for axis title, axis label, and axis line. Default is c(3,1,0)

oldParValues <- par(no.readonly = TRUE)  
  
par(mgp = c(1,4,4))  
  
devnull <- plot(1:10, main = "Effect of MGP")

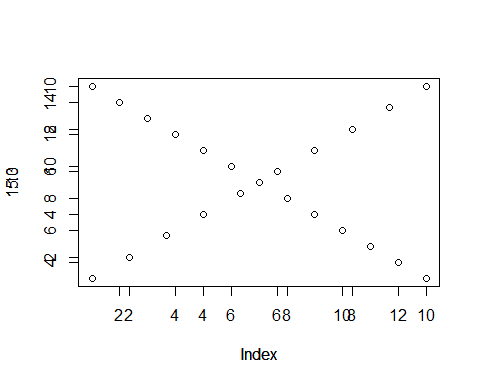


par(oldParValues) # this resets par to default values

## new

Prevents cleaning the frame of an output device between plot commands.

oldParValues <- par(no.readonly = TRUE)  
  
devnull <- plot(1:10)  
par(new = TRUE)  
devnull <- plot(15:3)



par(oldParValues) # this resets par to default values

## page

Indicates if the next plot is on a new page. Read only.

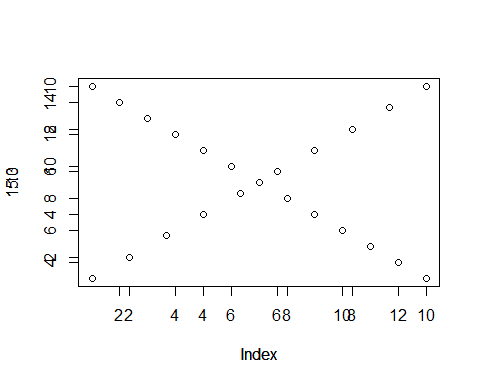
oldParValues <- par(no.readonly = TRUE)  
  
par("page")

## [1] TRUE

devnull <- plot(1:10)  
par(new = TRUE)  
par("page")

## [1] FALSE

devnull <- plot(15:3)



par(oldParValues) # this resets par to default values

# Regions

## xpd

A logical value or NA. If FALSE, all plotting is clipped to the plot region, if TRUE, all plotting is clipped to the figure region, and if NA, all plotting is clipped to the device region. See also clip.

## Plot Region

### pin

The current plot dimensions, (width, height), in inches.

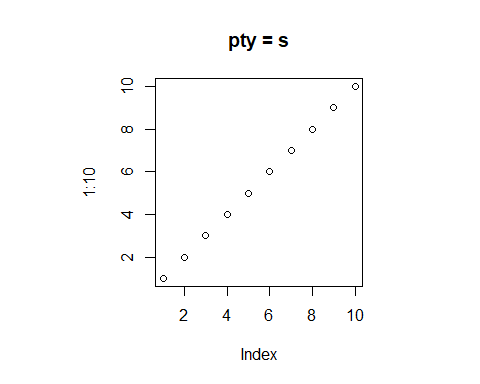
par("pin")

## [1] 3.759999 2.159999

### pty

A character specifying the type of plot region to be used; “s” generates a square plotting region and “m” generates the maximal plotting region.

oldParValues <- par(no.readonly = TRUE)  
  
par(pty = "s")  
  
devnull <- plot(1:10, main = "pty = s")



par(oldParValues) # this resets par to default values

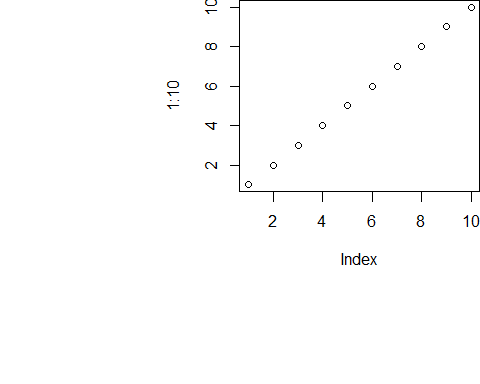
### plt

A vector of the form c(x1, x2, y1, y2) giving the coordinates of the plot region as fractions of the current figure region.

oldParValues <- par(no.readonly = TRUE)  
  
par("plt")

## [1] 0.1640000 0.9160000 0.2550001 0.7949999

par(plt = c(.5,1,.5,1))  
devnull <- plot(1:10, main = "plt ")



par(oldParValues) # this resets par to default values

## Figure Region

### fig

A numerical vector of the form c(x1, x2, y1, y2) which gives the (NDC) coordinates of the figure region in the display region of the device. If you set this, unlike S, you start a new plot, so to add to an existing plot use new = TRUE as well.

### fin

The figure region dimensions, (width, height), in inches. If you set this, unlike S, you start a new plot.

# These don’t seem to work

## xaxs & yaxs

The style of axis interval calculation to be used for the x-axis. Only “r” and “i” styles have been implemented in R. Style “r” (regular) first extends the data range by 4 percent at each end and then finds an axis with pretty labels that fits within the extended range. Style “i” (internal) just finds an axis with pretty labels that fits within the original data range.

## ylbias

A positive real value used in the positioning of text in the margins by axis and mtext. The default is in principle device-specific, but currently 0.2 for all of R’s own devices. Set this to 0.2 for compatibility with R < 2.14.0 on x11 and windows() devices.

## mex

mex is only used with plots that change shape.

## xaxp & yaxp

Coordinates c(x,y,n) of extreme tick marks and number of intervals between tick marks when par(“xlog”) == FALSE.

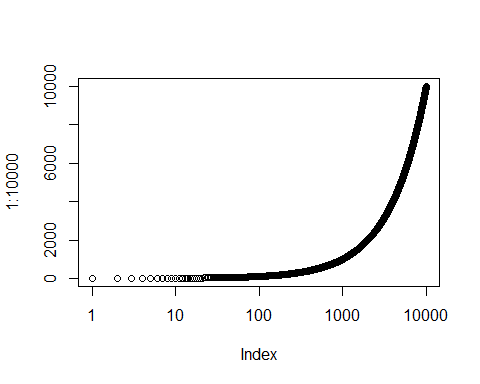
If par(“xlog”) == TRUE then for c(x,y,n), x & y are lowest and highest power of 10. If n=1 tick marks at 10^j for integer j; n = 2 gives marks k 10^j with k in {1,5}; n = 3 gives marks k 10^j with k in {1,2,5}.

See axTicks() for a pure R implementation of this.

oldParValues <- par(no.readonly = TRUE)  
  
par("xaxp")

## [1] 0 1 5

devnull <- plot(1:10000, log = "x")

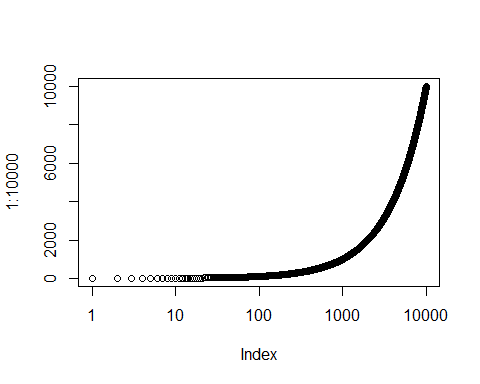


par("xlog")

## [1] TRUE

par(oldParValues)

oldParValues <- par(no.readonly = TRUE)  
  
par(xaxp = c(20,20,2))  
devnull <- plot(1:10000, log = "x")



par("xlog")

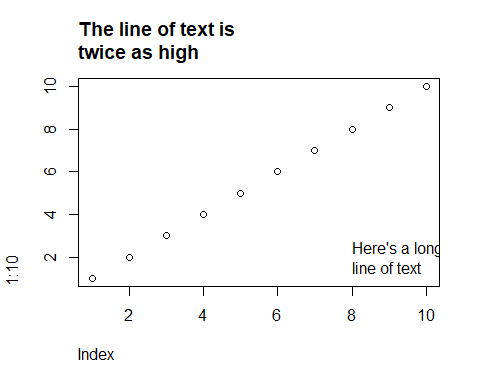
## [1] TRUE

par(oldParValues)

## lheight

Line of text height multiplier. Relative to character height. Default is 1. (The same inter-line spacing multiplied by lheight is used for multi-line strings in text and strheight.)

oldParValues <- par(no.readonly = TRUE)  
  
par( lheight = 4, adj = 0)  
  
devnull <- plot(1:10)+  
title( main = "The line of text is \ntwice as high") +  
text(x = 8, y = 2, labels = "Here's a long \nline of text")



par(oldParValues)