**Ghement Statistical Consulting Company Ltd. © 2013**

**Customizing R graphics using the function text()**

**text()**

The R function text() allows you to place a specified text string in a particular location on your R plot.

**Example:**



plot(mpg ~ hp, data=mtcars)

text(x=200,y=25, labels="Some text")

As seen in the above example, text() has 3 required arguments:

1. x, which stands for the x-axis coordinate of the text;
2. y, which stands for the y-axis coordinate of the text;
3. labels, which lists the text string.

The arguments x and y can be replaced by the argument locator(1). This allows you to click once on your R plot and place the string text wherever you clicked. (If you used locator(2), you could specify two different text strings and click twice on your plot. Note that R will wait for you to click twice on the plot and then display the two text strings.)



**Examples:**

plot(mpg ~ hp, data=mtcars)

text(locator(1), labels="Some text")

plot(mpg ~ hp, data=mtcars)

text(locator(2), labels=c("Text 1","Text 2"))

It is possible to control the size and color of the text string fed to the function text(). Simply use the argument col to control text color and the argument cex to control text size. Recall that cex=1 means that the text will have default size, cex = 1.5 means that the text will have text size that is 1.5 times larger than the default text size in R and cex = 0.5 means that the text will have size that is half the size of the default text size in R.

**Example:**



plot(mpg ~ hp, data=mtcars)

text(x=200,y=25,

 labels="Some text",

 col="blue",

 cex=1.5)

If you check the help file for the text() function, you will note that this function has other optional arguments:

help(text)

One of these optional arguments is named pos and can take one of the following values: 1, 2, 3 or 4. The meaning of these values is explained below:

* text(x=200, y=25, labels="Some text", pos=1)

If **pos=1**, the text string will be positioned **below** the specified coordinates x and y.

* text(x=200, y=25, labels="Some text", pos=2)

If **pos=2**, the text string will be positioned to the **left** of the specified coordinates x and y.

* text(x=200, y=25, labels="Some text", pos=3)

If **pos=3**, the text string will be positioned **above** the specified coordinates x and y.

* text(x=200, y=25, labels="Some text", pos=4)

If **pos=4**, the text string will be positioned to the **right** of the specified coordinates x and y.

**Examples:**

par(mfrow=c(2,2))

plot(mpg ~ hp, data=mtcars)

points(x=200, y=25,pch=19,col="red")

text(x=200, y=25, labels="Some text", col="blue", pos=1)

title("pos=1 (below specified coordinates)")

plot(mpg ~ hp, data=mtcars)

points(x=200, y=25,pch=19,col="red")

text(x=200, y=25, labels="Some text", col="blue", pos=2)

title("pos=2 (to the left of specified coordinates)")

plot(mpg ~ hp, data=mtcars)

points(x=200, y=25,pch=19,col="red")

text(x=200, y=25, labels="Some text", col="blue", pos=3)

title("pos=3 (above the specified coordinates)")

plot(mpg ~ hp, data=mtcars)

points(x=200, y=25,pch=19,col="red")

text(x=200, y=25, labels="Some text", col="blue", pos=4)

title("pos=4 (to the right of specified coordinates)")



One final note – it is possible to add mathematical symbols or formulas to an R plot via the text() function. This is illustrated below.

**Example 1 [Adding a mathematical equation to a plot]:**

plot(mpg ~ hp, data=mtcars)

points(x=200, y=25,pch=19,col="red")

plotlabels <- **expression(paste(**"mpg = ", # Text

 beta[0], # Greek symbol

 " + ", # Text

 beta[1], # Greek symbol

 "\*hp", # Text

 " + ", # Text

 epsilon**)** # Greek Symbol

 **)**

text(x=200, y=25, labels=plotlabels, col="blue", pos=3)



Note:

- If you want to combine multiple mathematical expressions with text in R, you need to use paste() inside expression().

- In R, the Greek symbol for β is **beta**.

- In R, using square brackets allows the addition of subscripts to a Greek symbol, so that $β\_{1}$ is **beta[1]**.

**Example 2 [Adding a regression equation to a plot]:**

plot(mpg ~ hp, data=mtcars)

model <- lm(mpg ~ hp, data=mtcars) # regress mpg on hp

int <- coef(model)[1] # compute intercept from the data and call it **int**

int <- round(int, 2) # round int to 2 significant digits

slp <- coef(model)[2] # compute slope from the data and call it **slp**

slp <- round(slp, 2) # round slp to 2 significant digits

int

slp

plotlabels <- substitute(

 paste("mpg = ",

 b0, # b0 is a generic intercept

 " ",

 s, # s is the sign of the slope

 " ",

 b1, # b1 is a generic slope, stripped of sign

 "hp"),

 list(b0 = intercept, # bo is set to specific **int**

s = substr(sign(slp),1,1),# s is set to sign of **slp**

 b1 = abs(slp)) # b1 is set to absolute **slp**

 )

text(x=200, y=25, labels=plotlabels, col="blue", pos=3)

