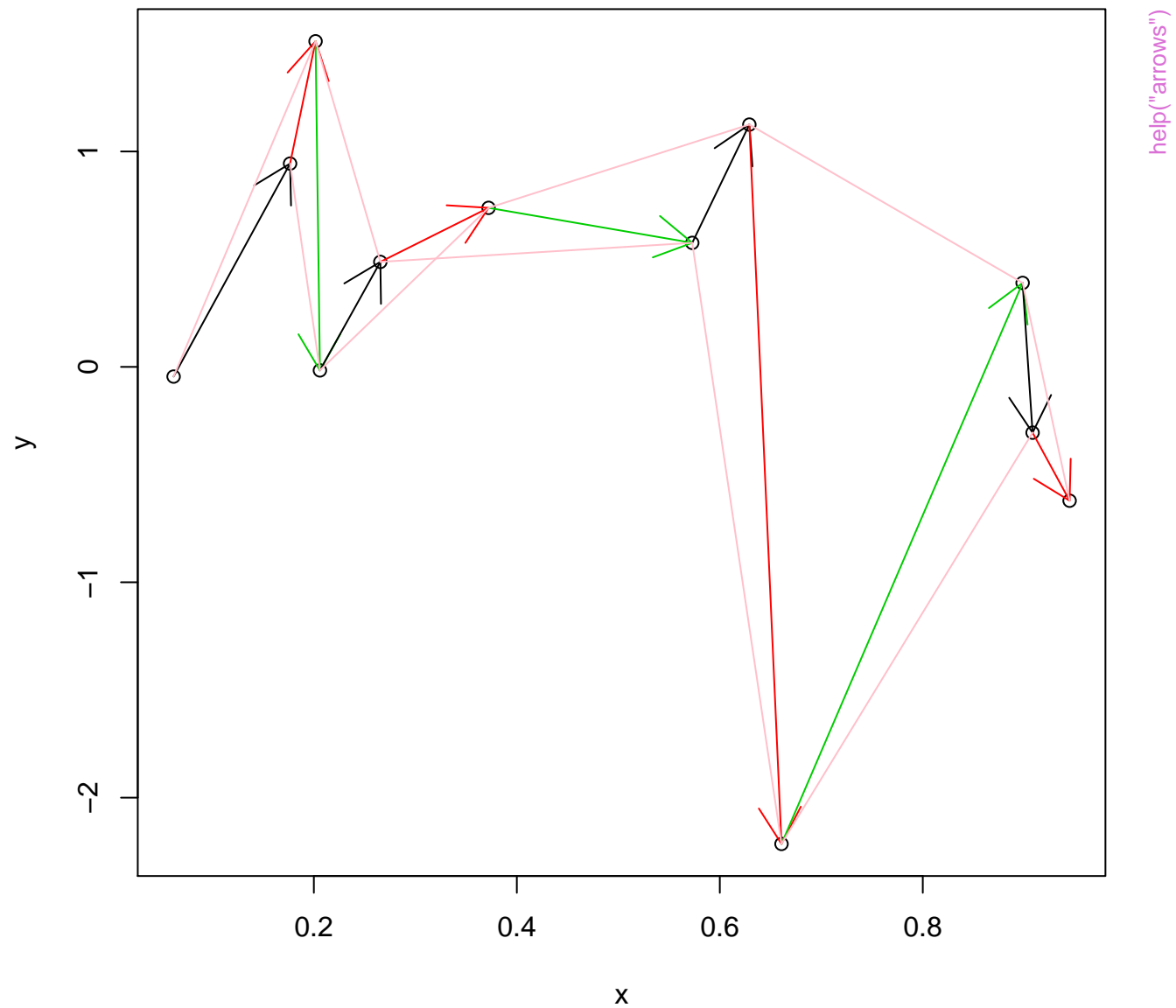
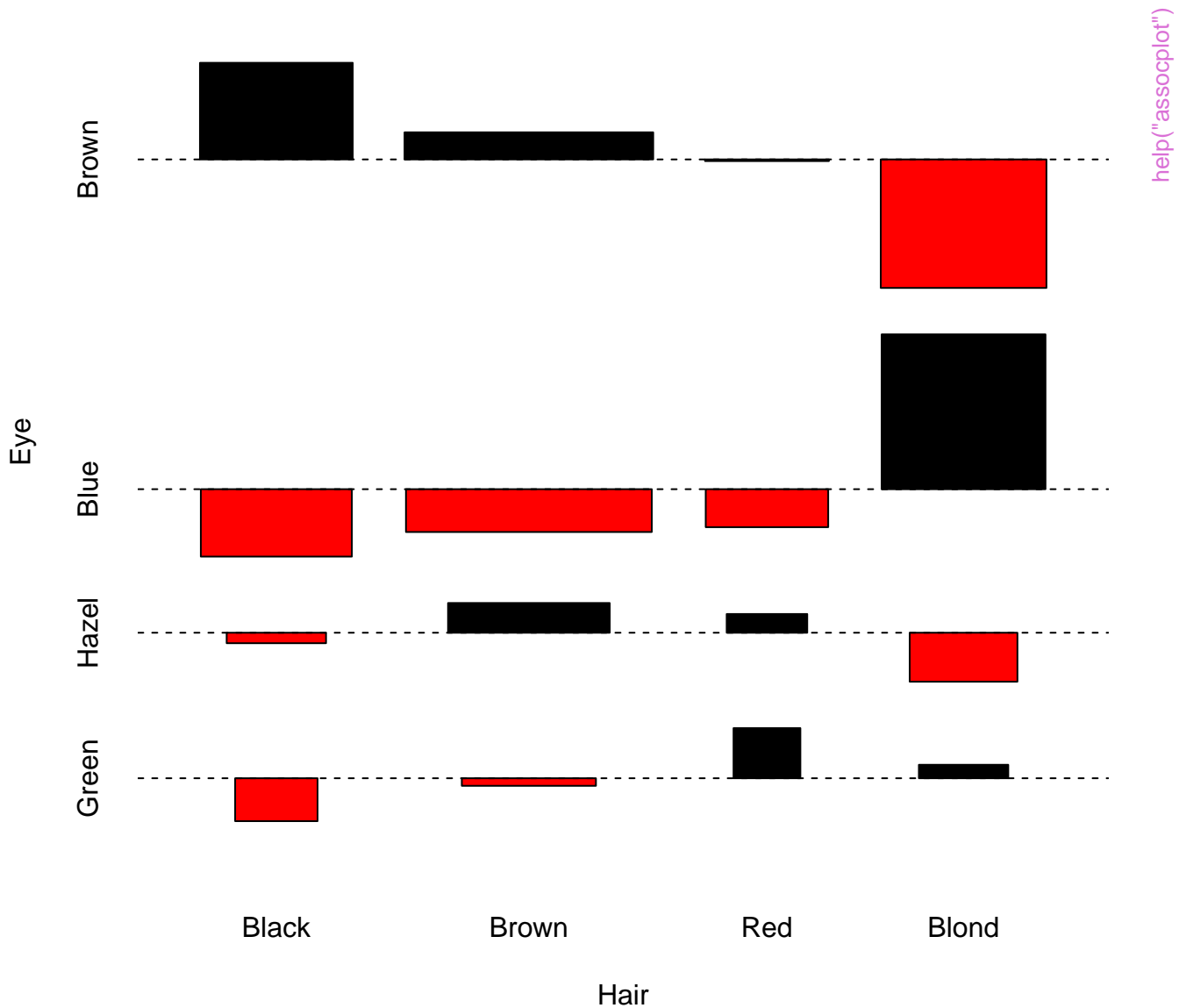
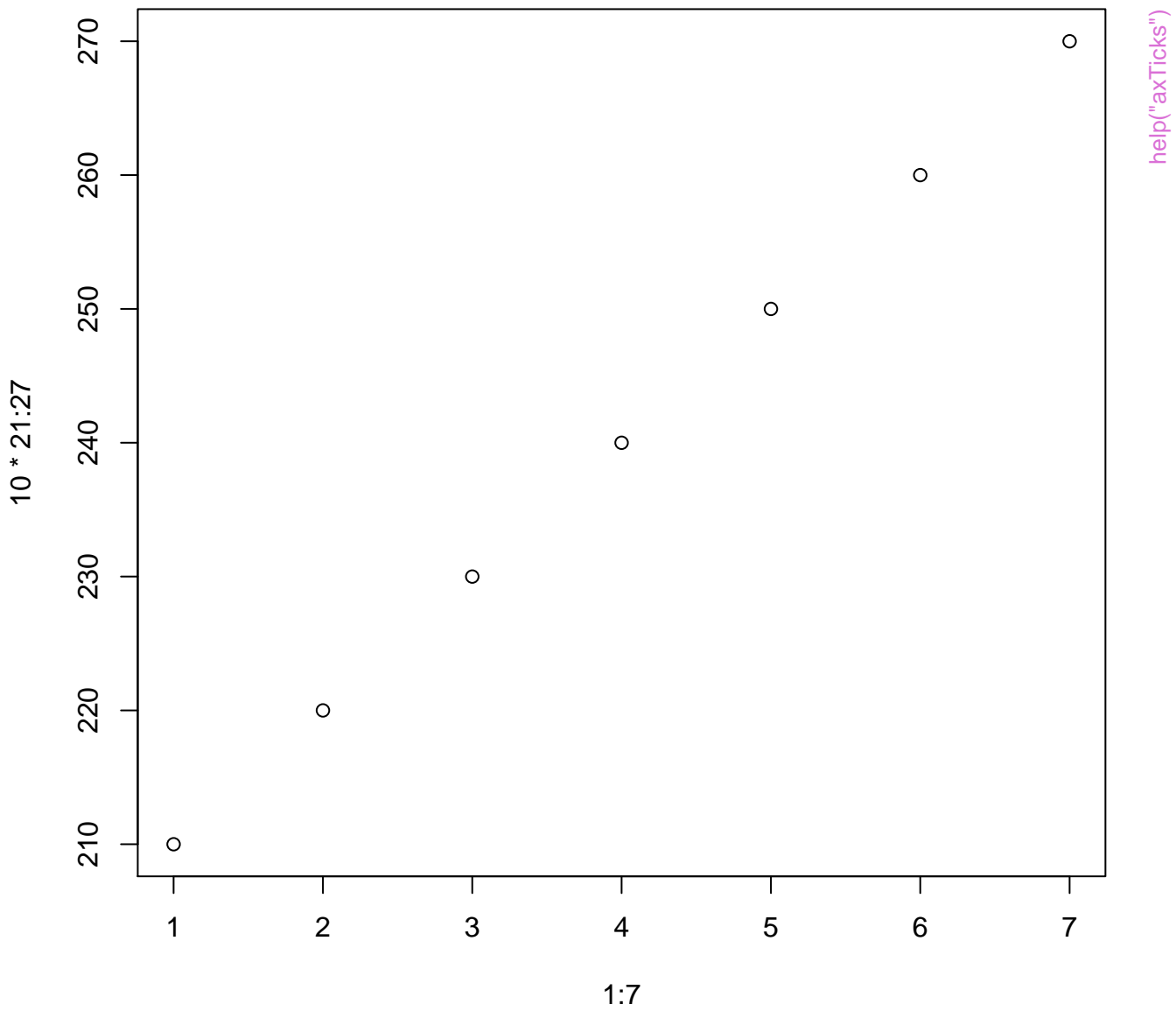


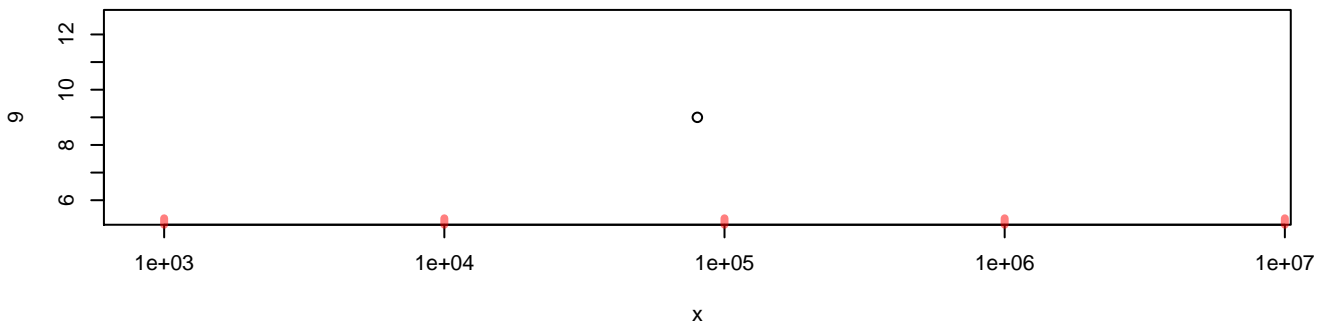
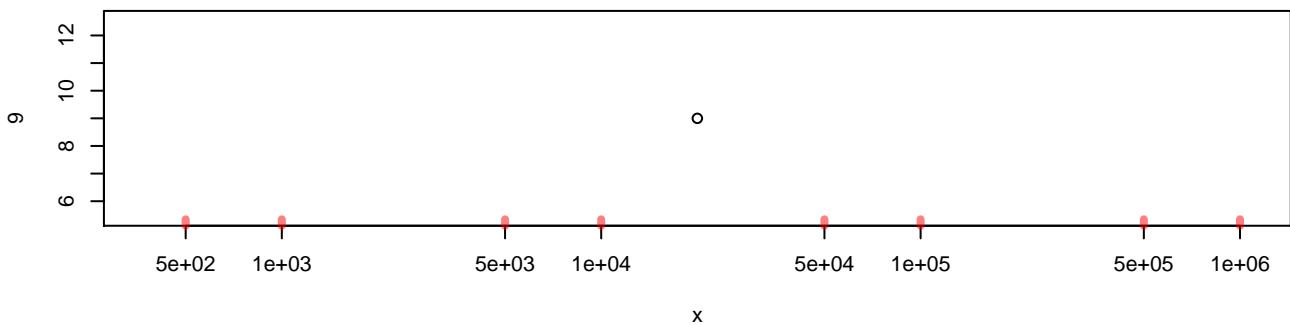
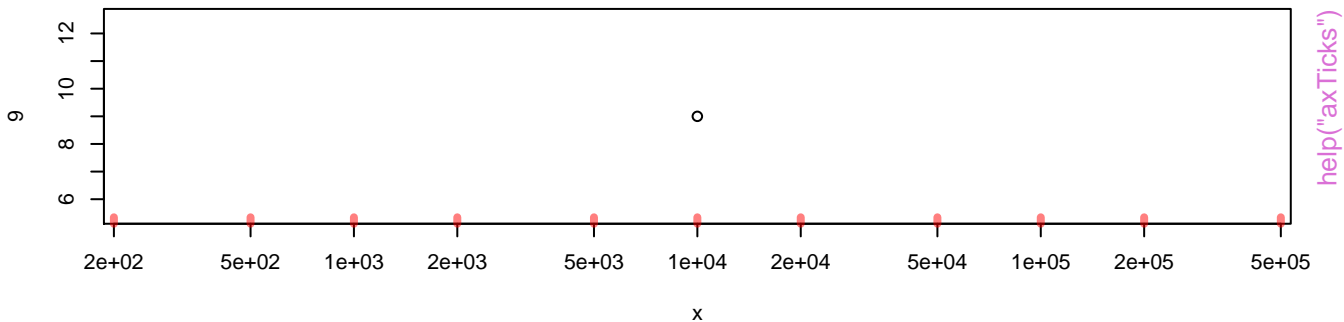
## arrows(.) and segments(.)



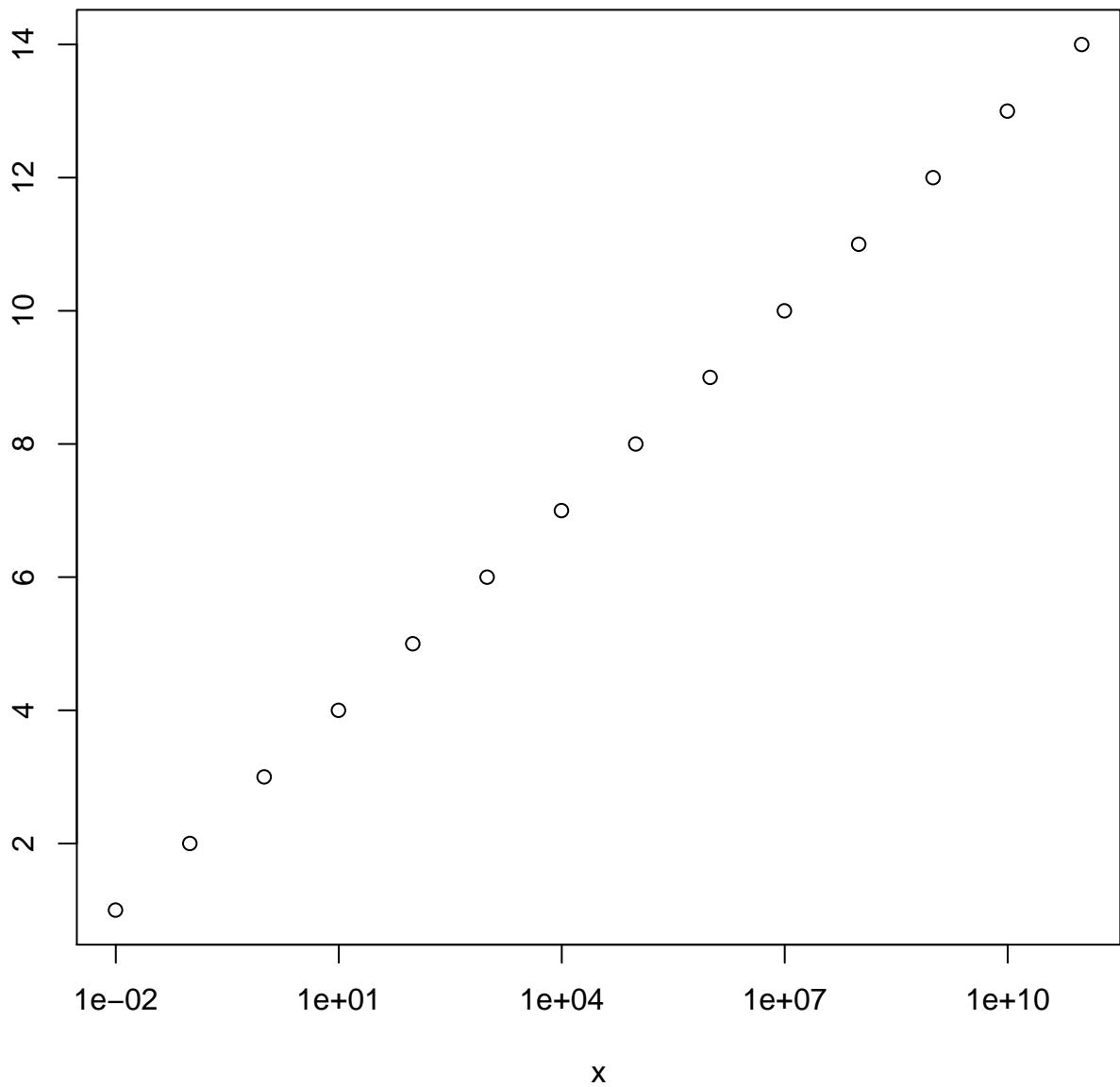
Relation between hair and eye color





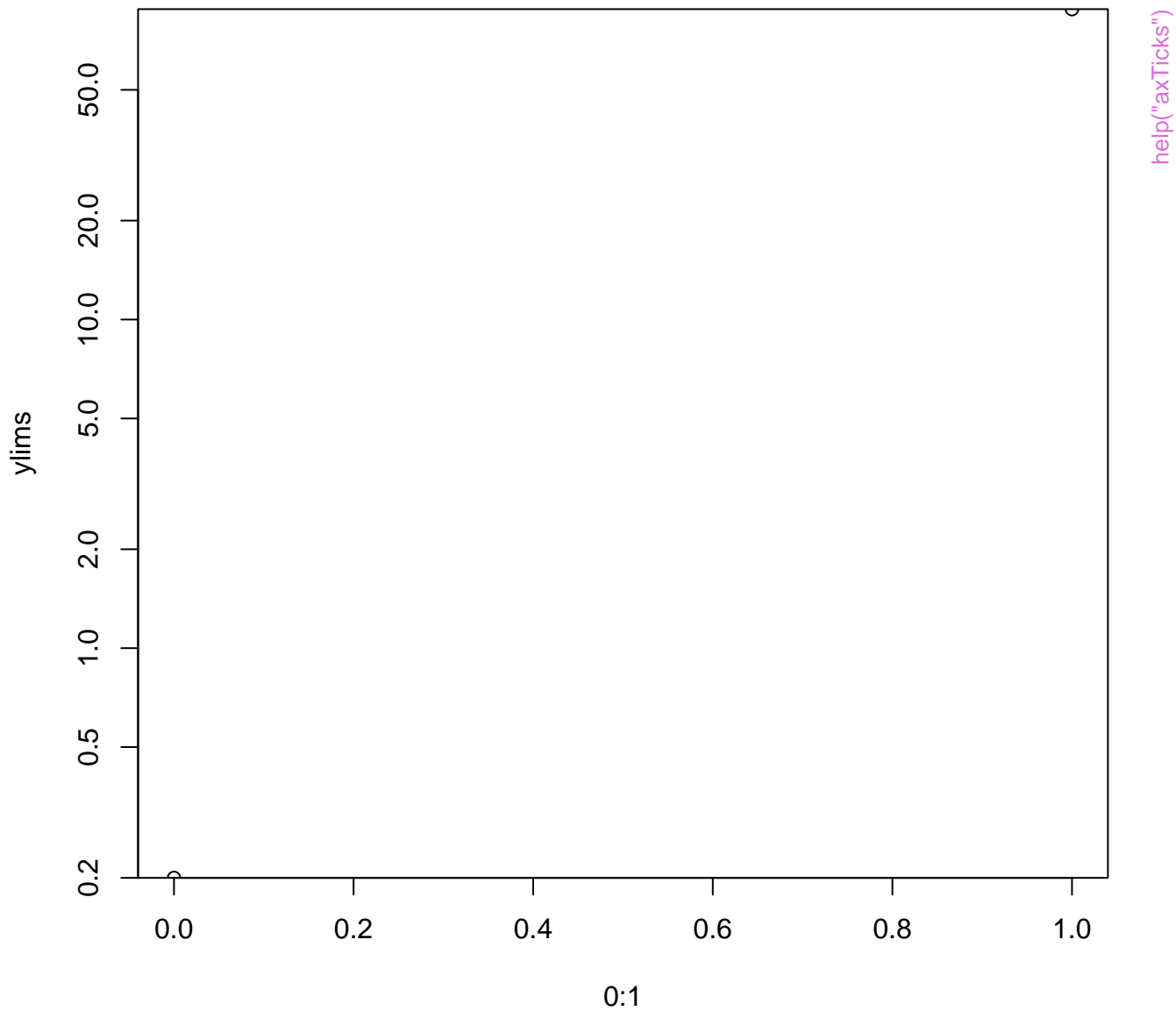


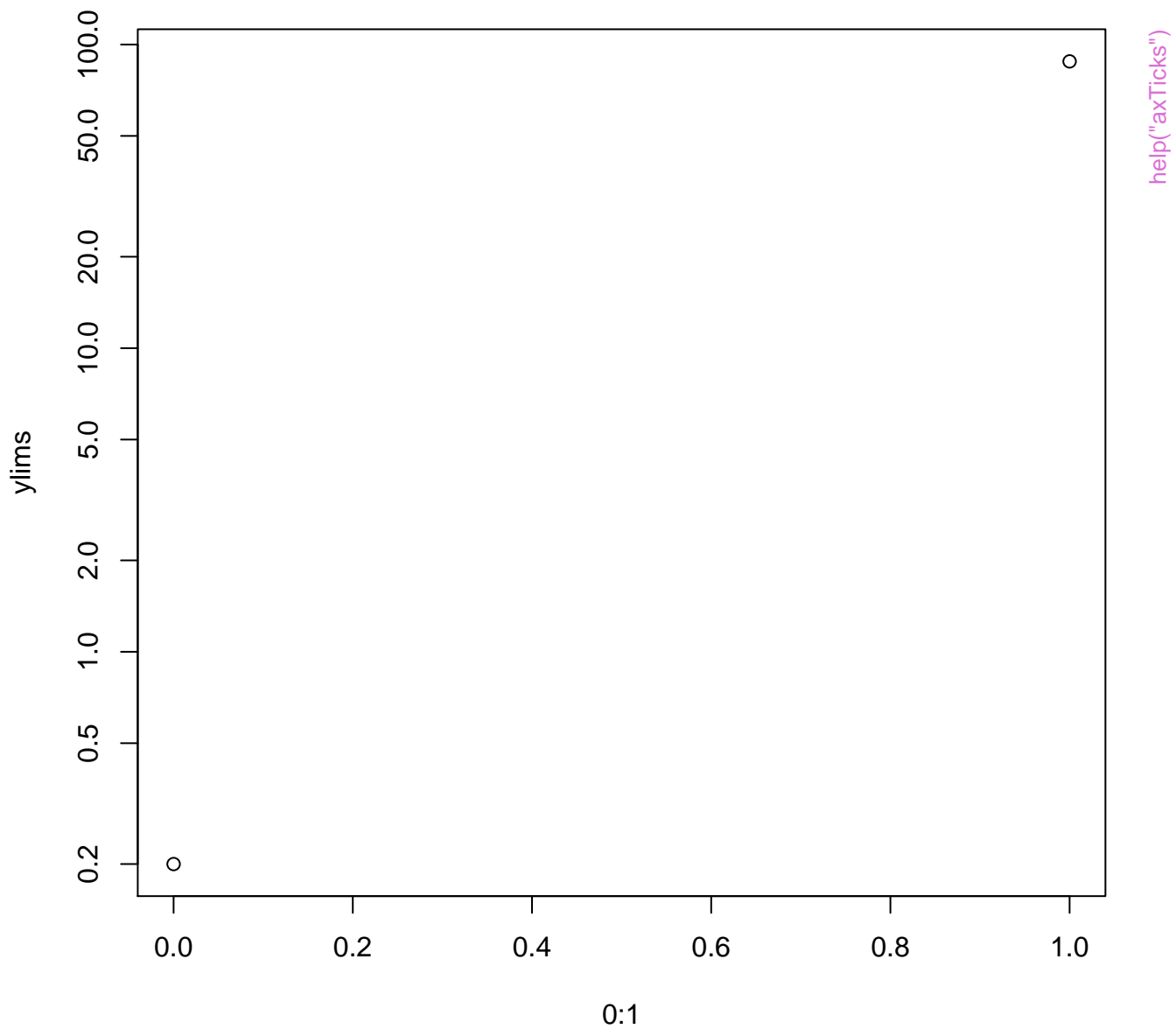
1:14

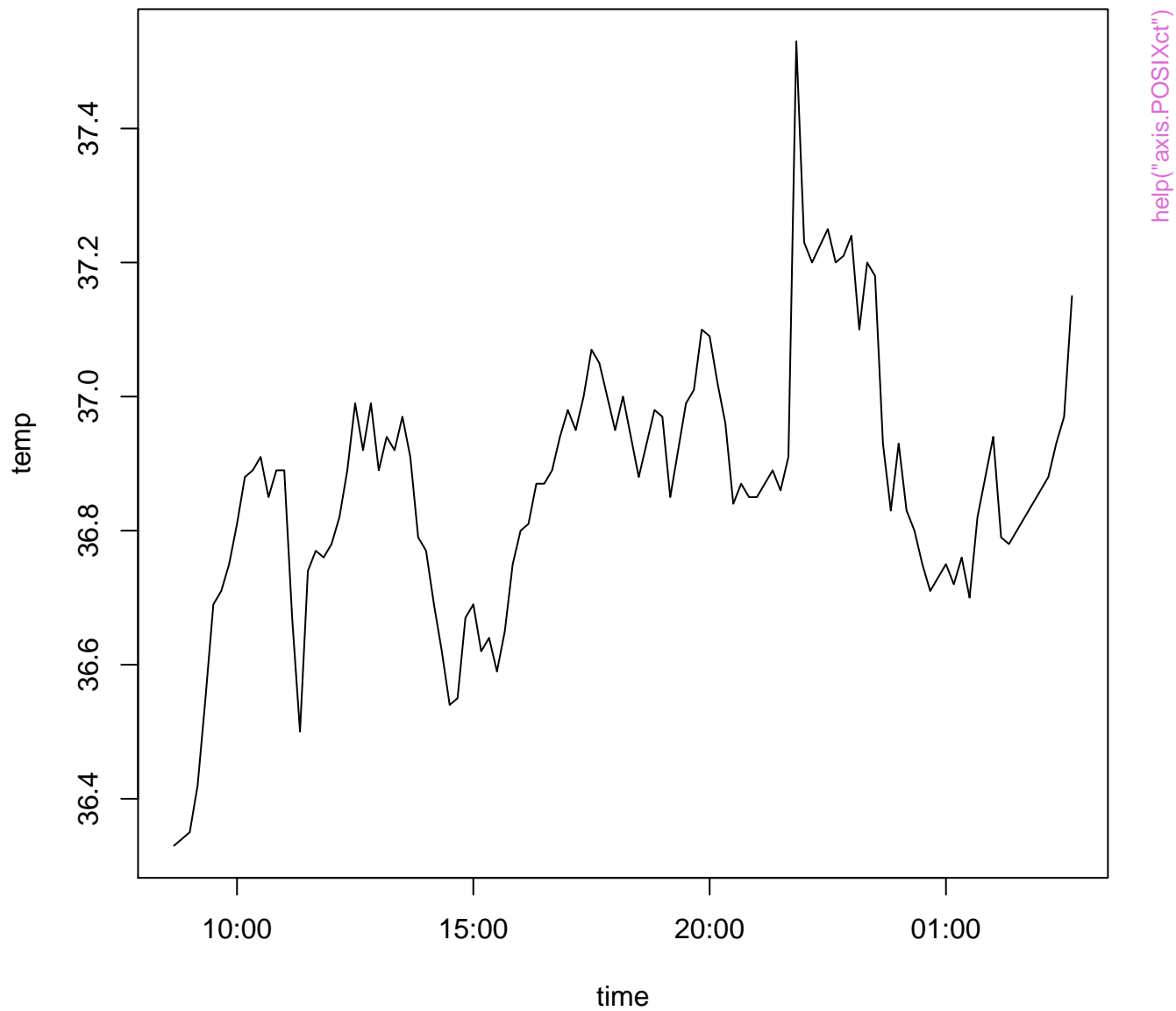


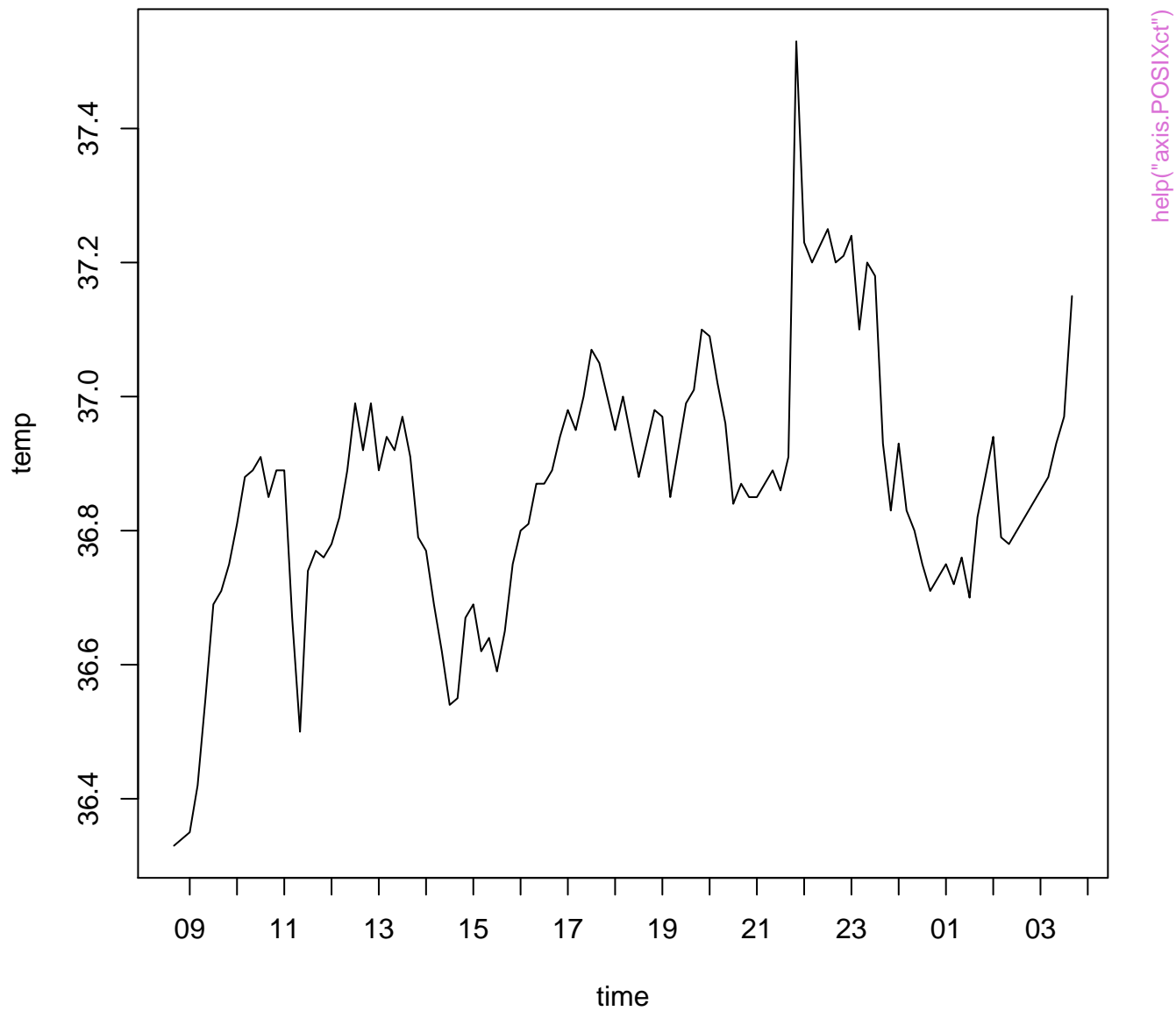
help("axTicks")

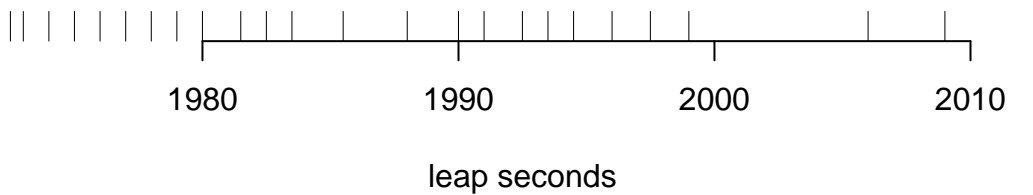




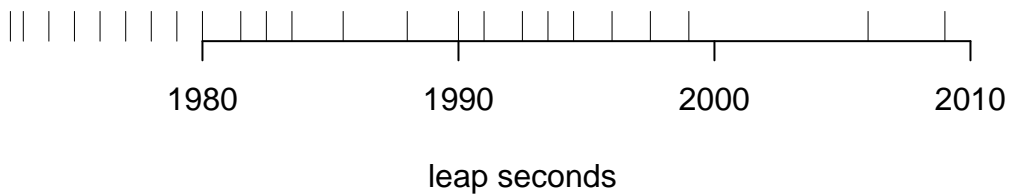




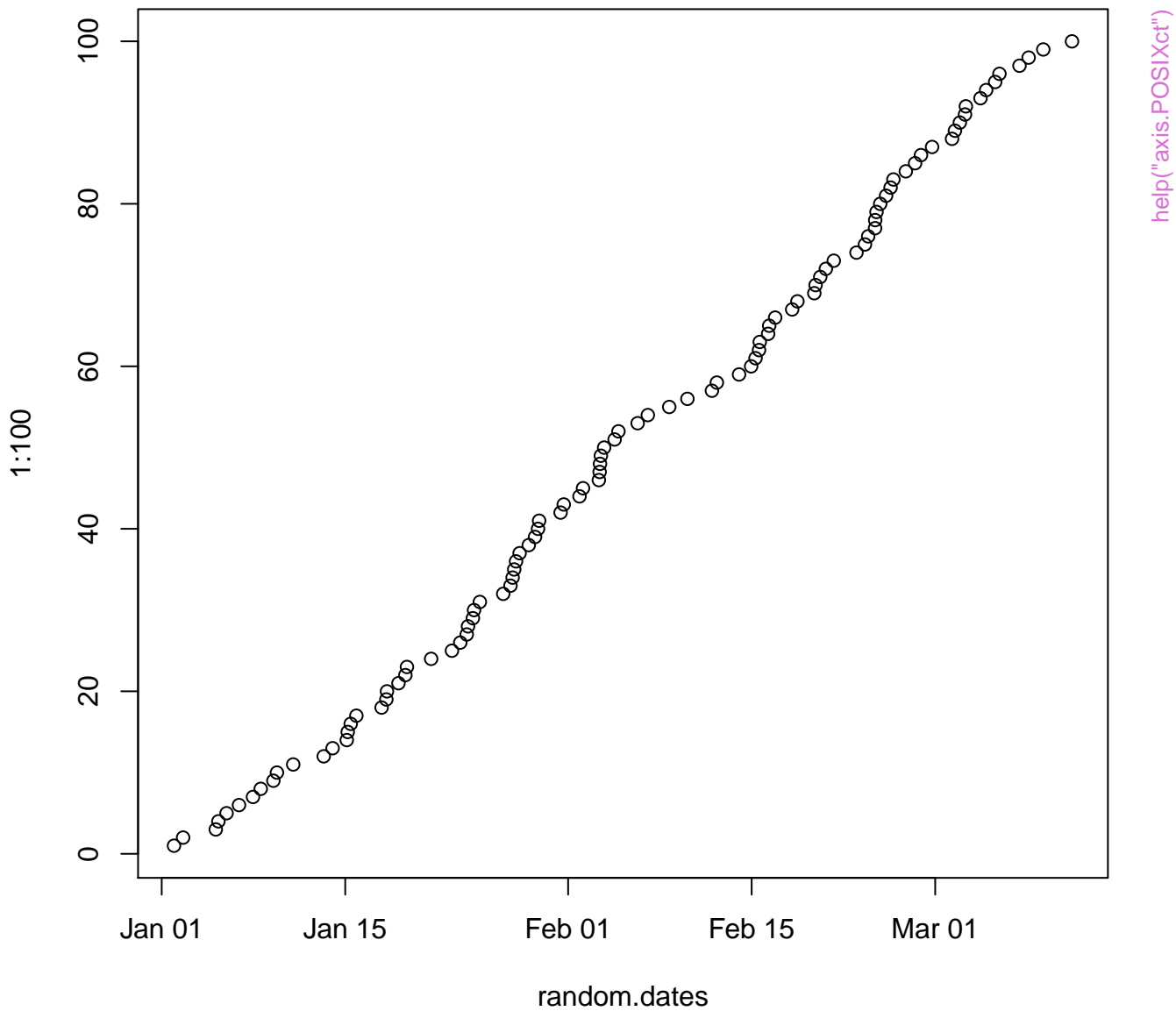


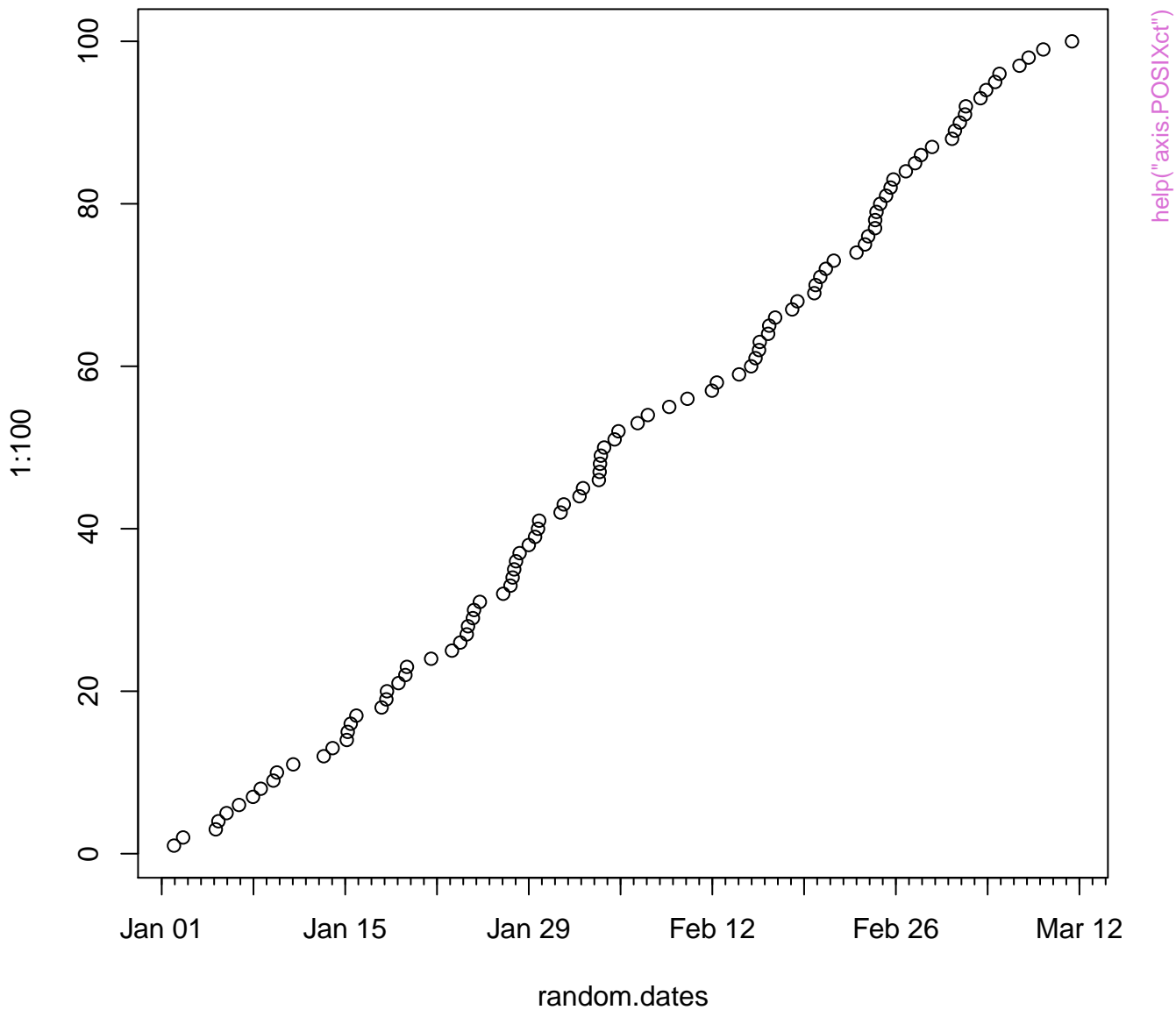


help("axis.POSIXct")

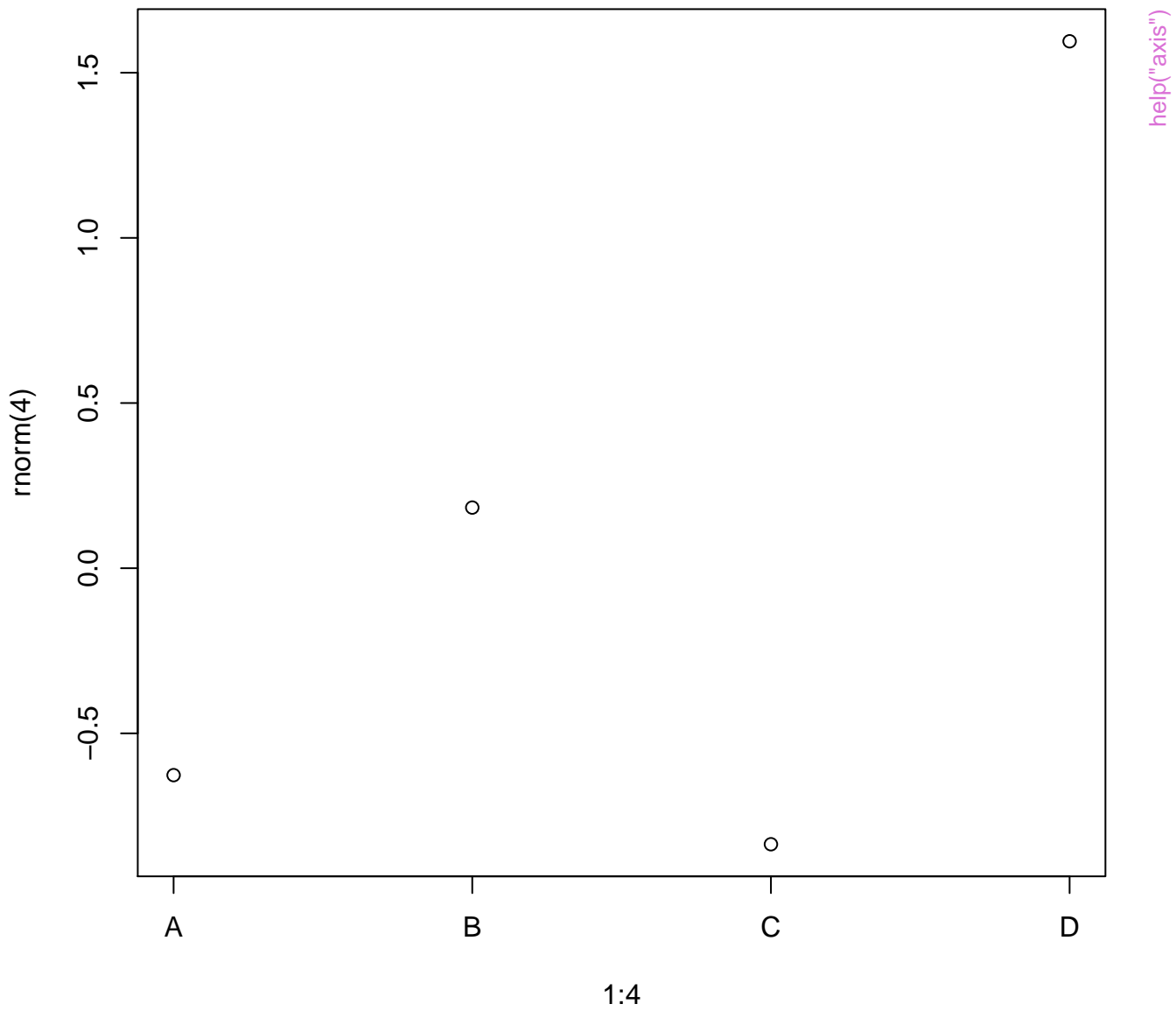


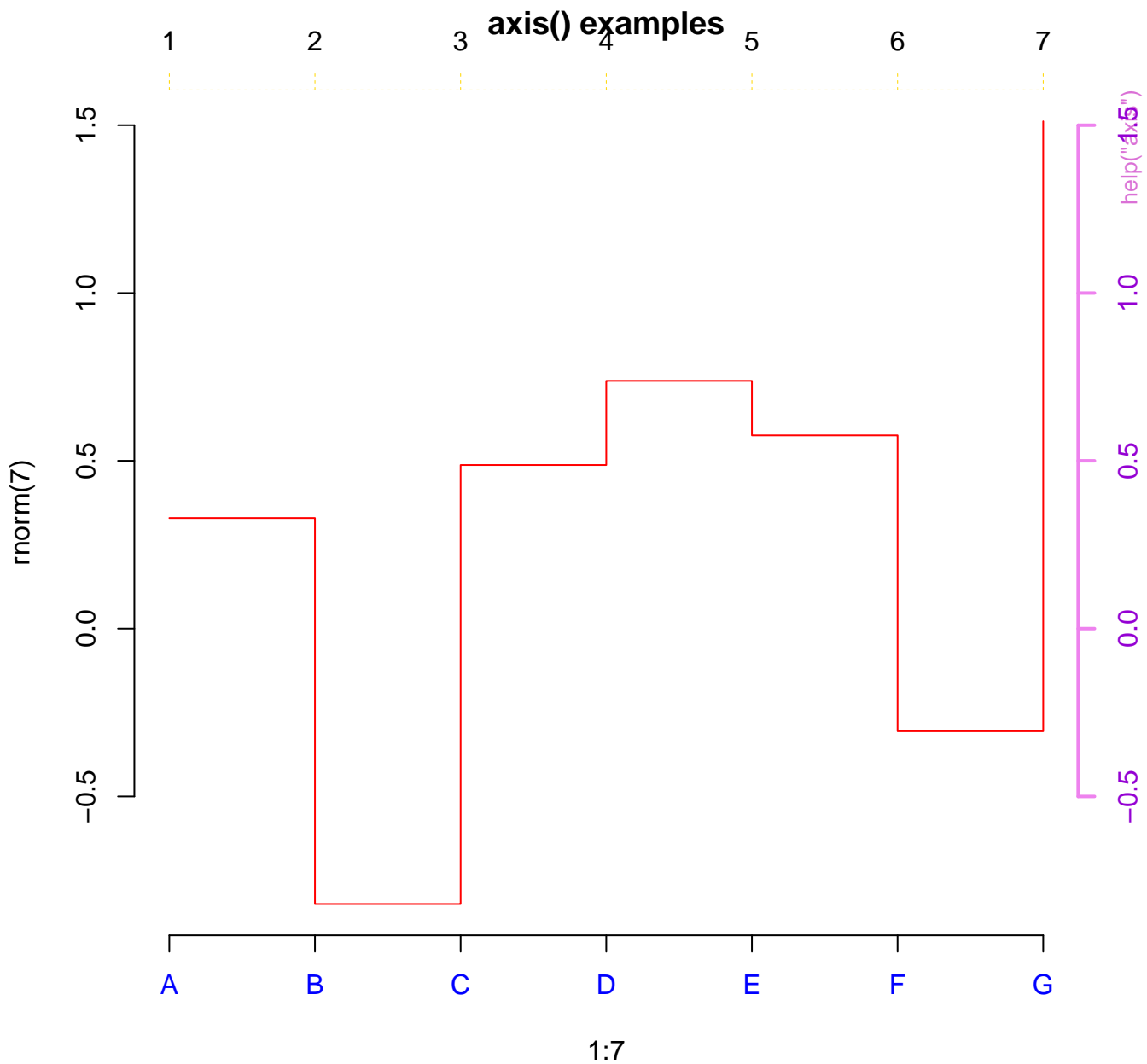
help("axis.POSIXct")

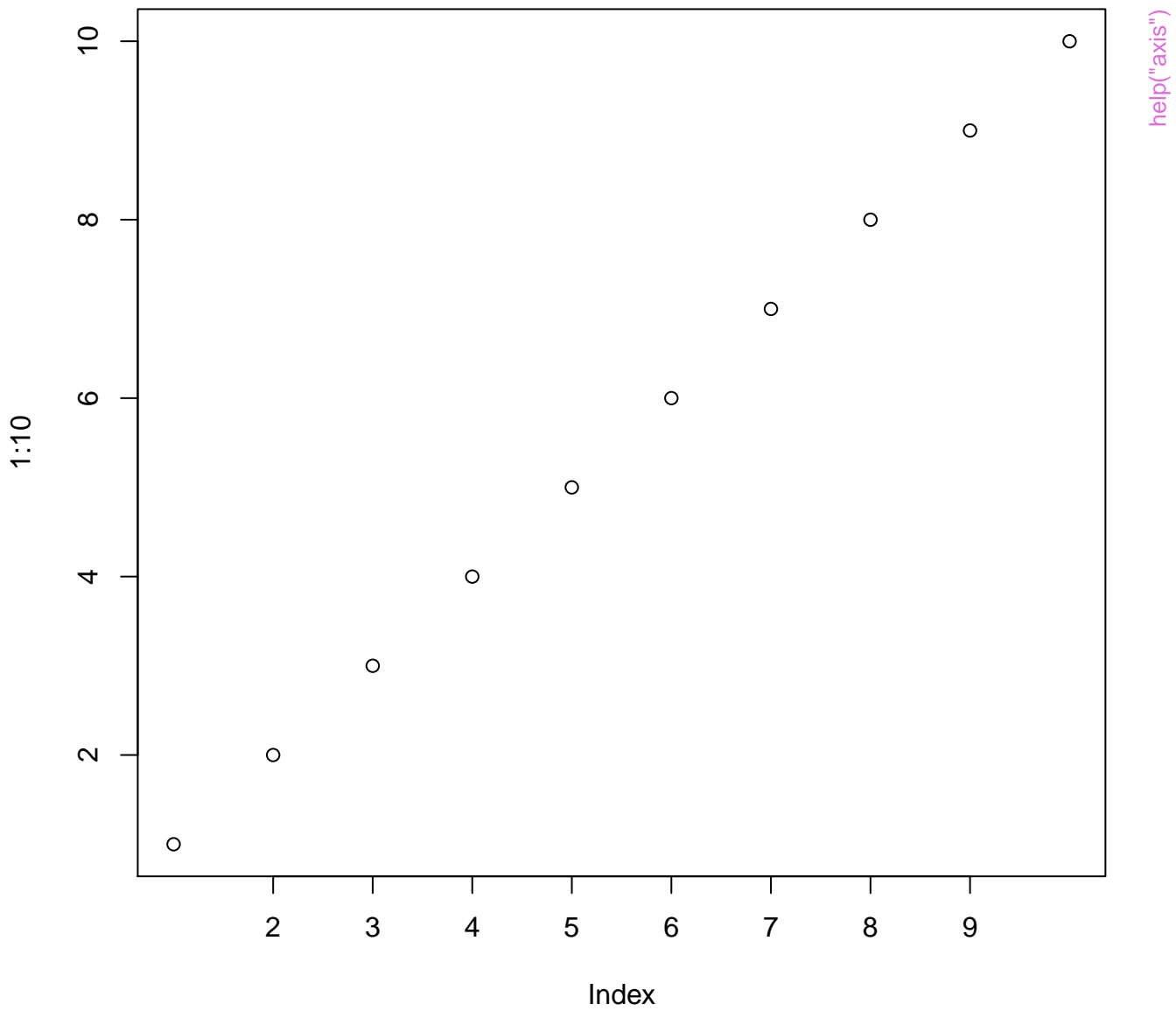


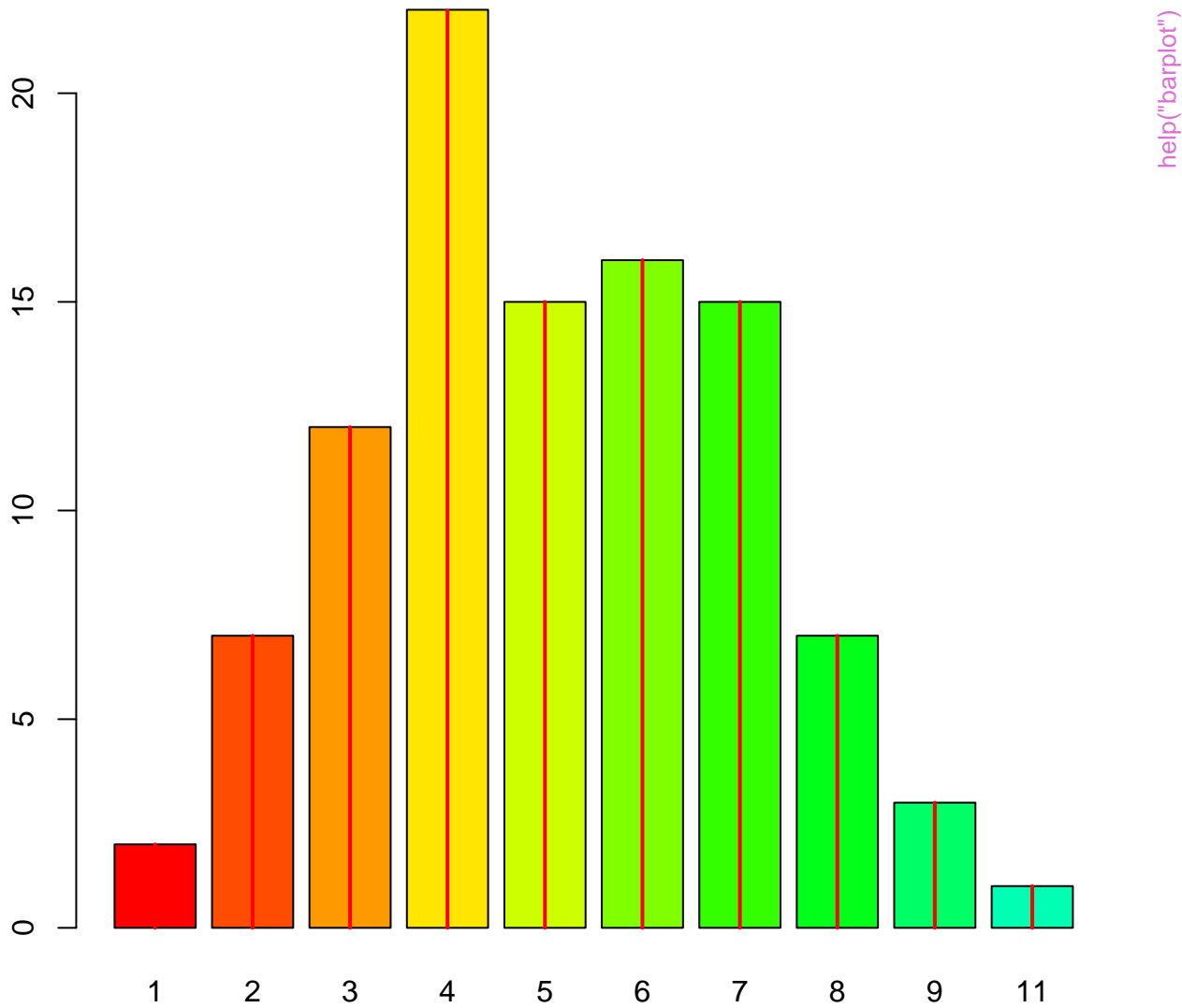


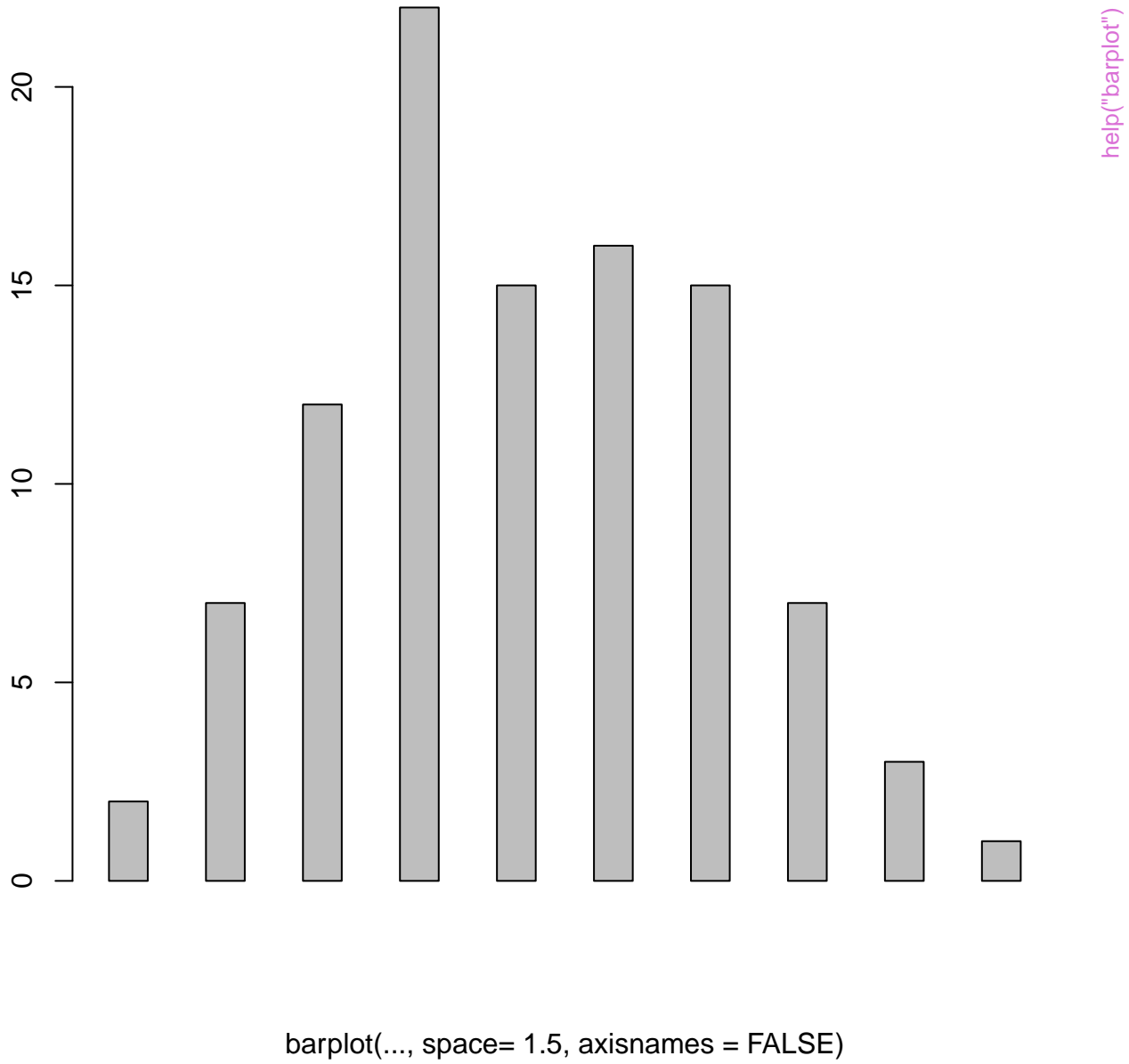


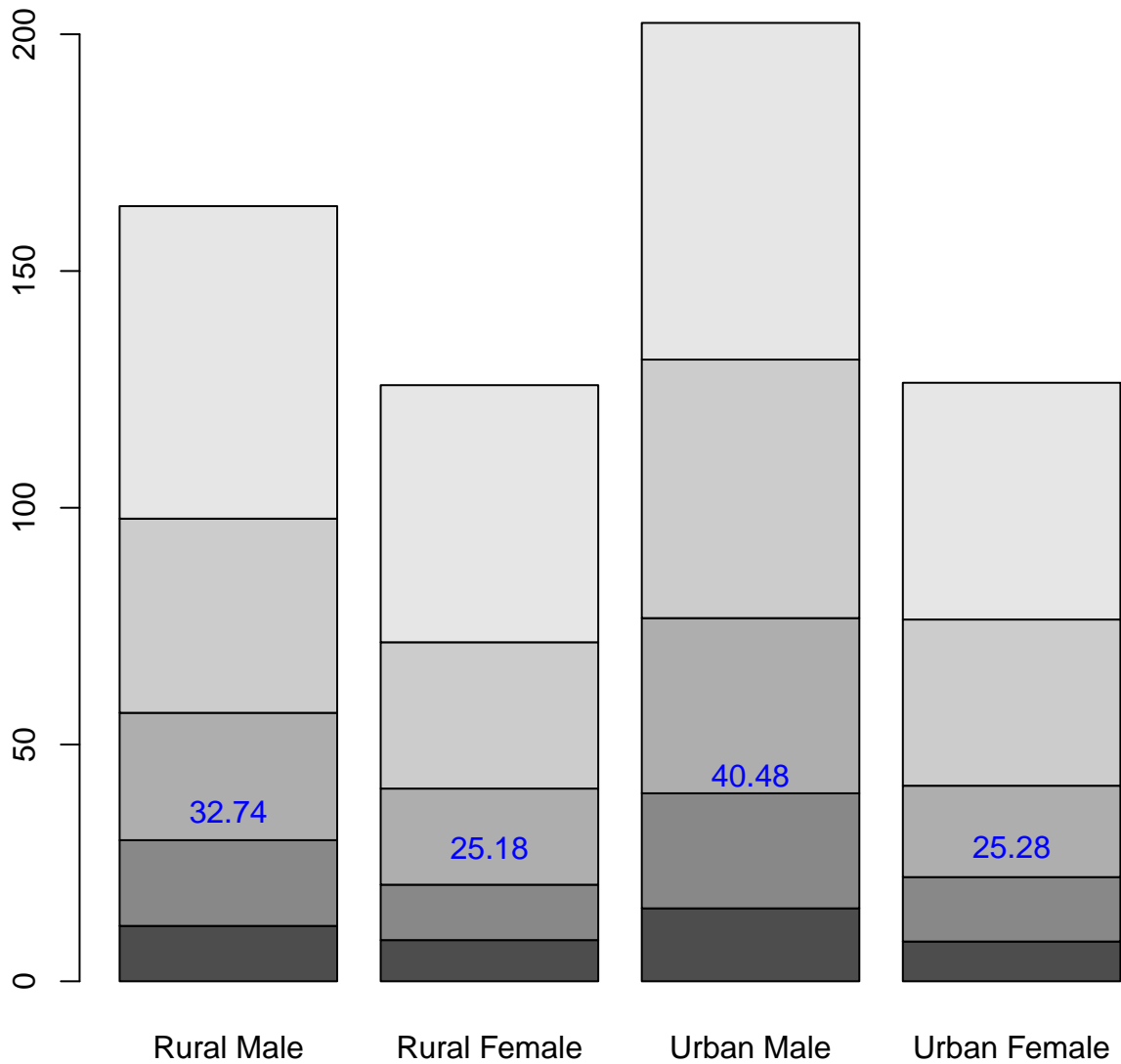






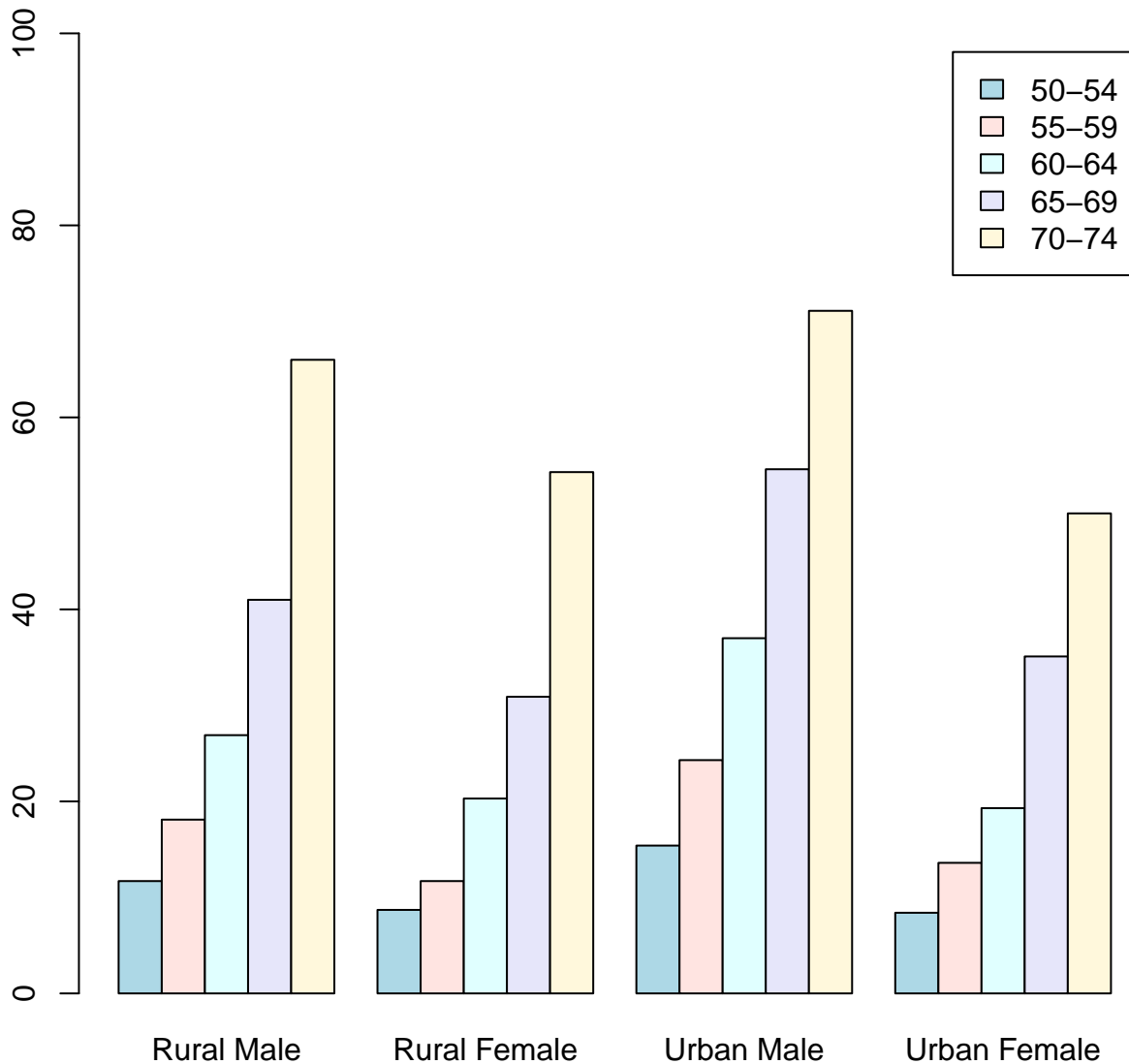






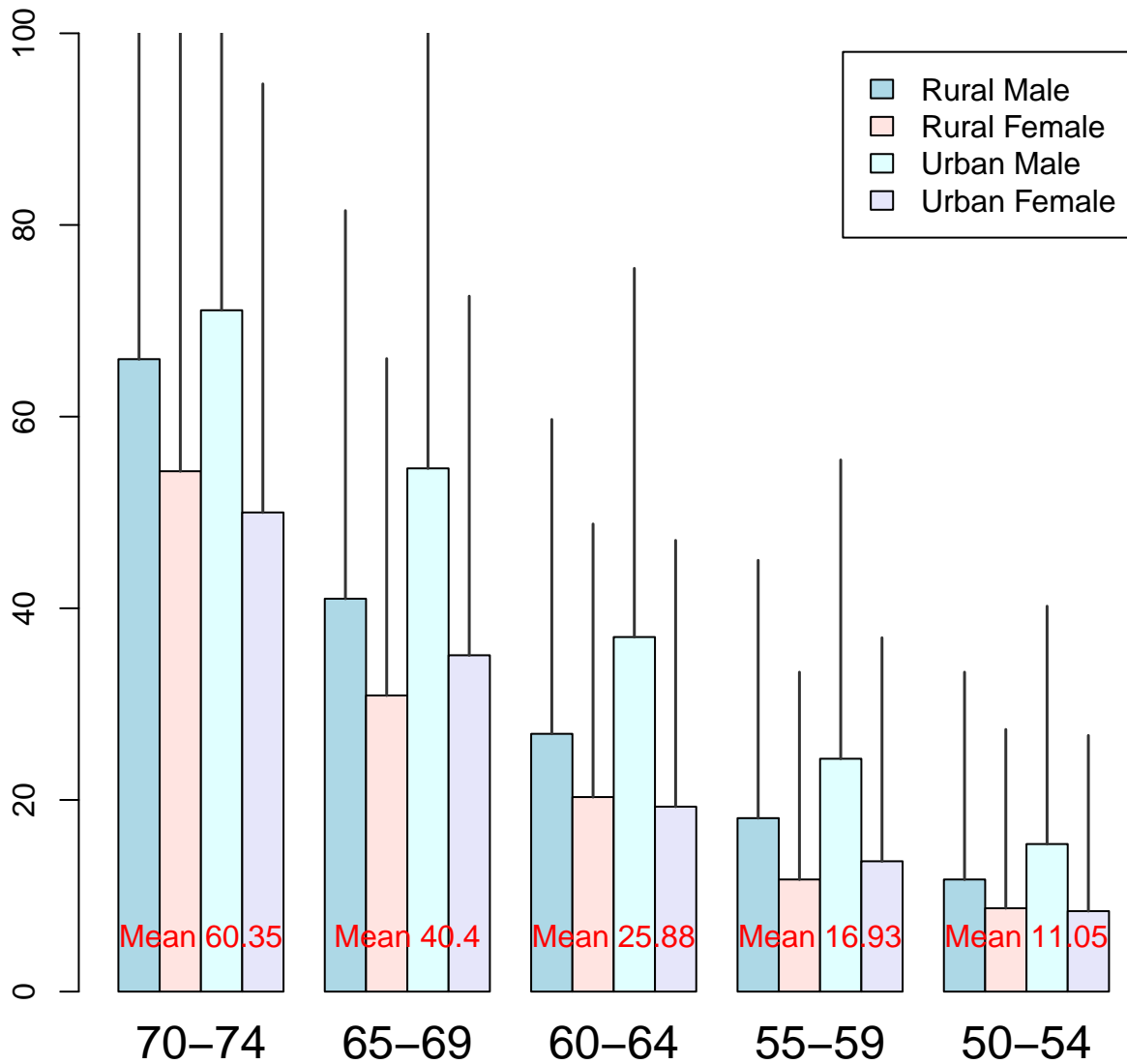
help("barplot")

## *Death Rates in Virginia*



help("barplot")

## Death Rates in Virginia

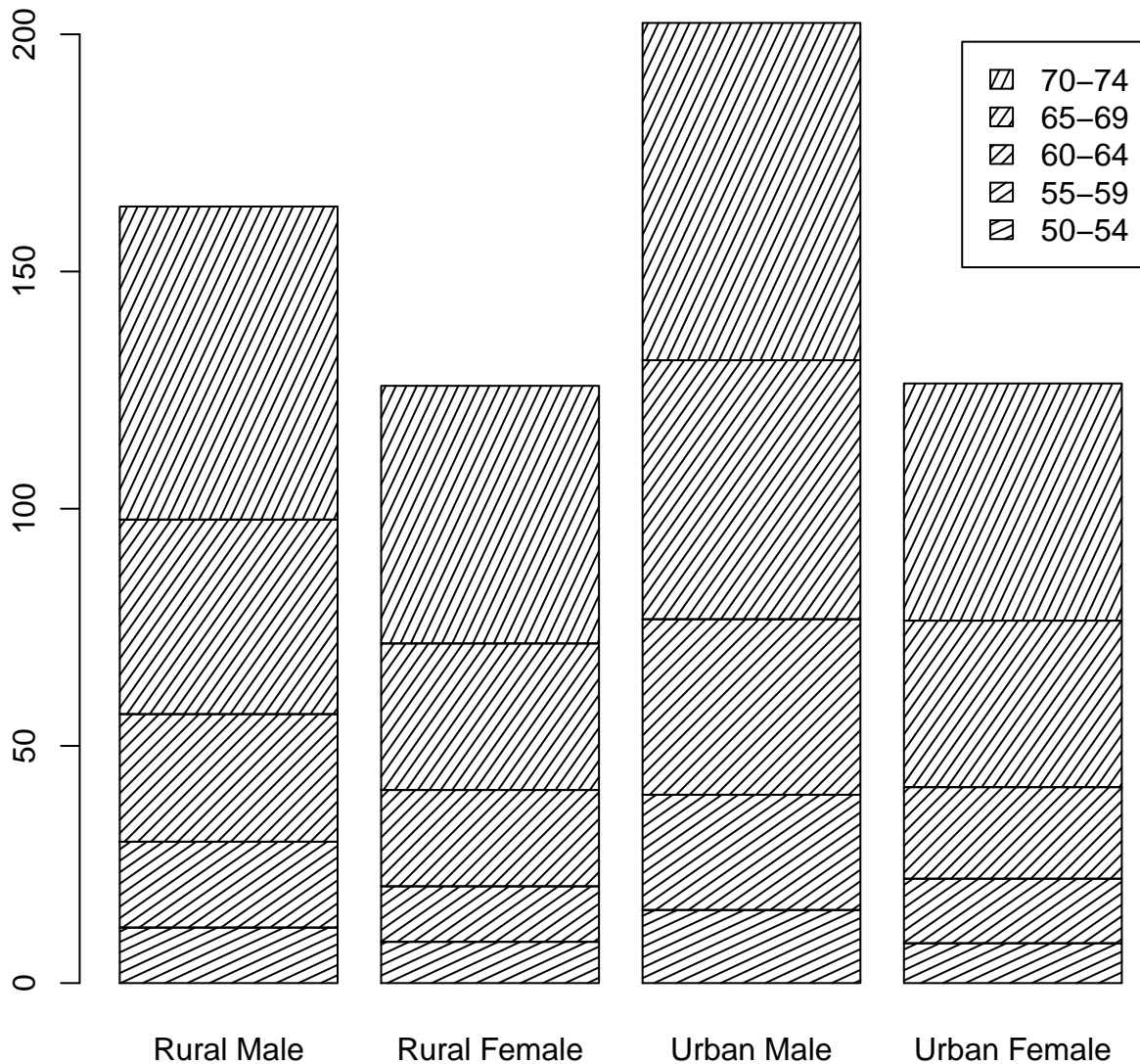


Faked upper 2\*sigma error bars

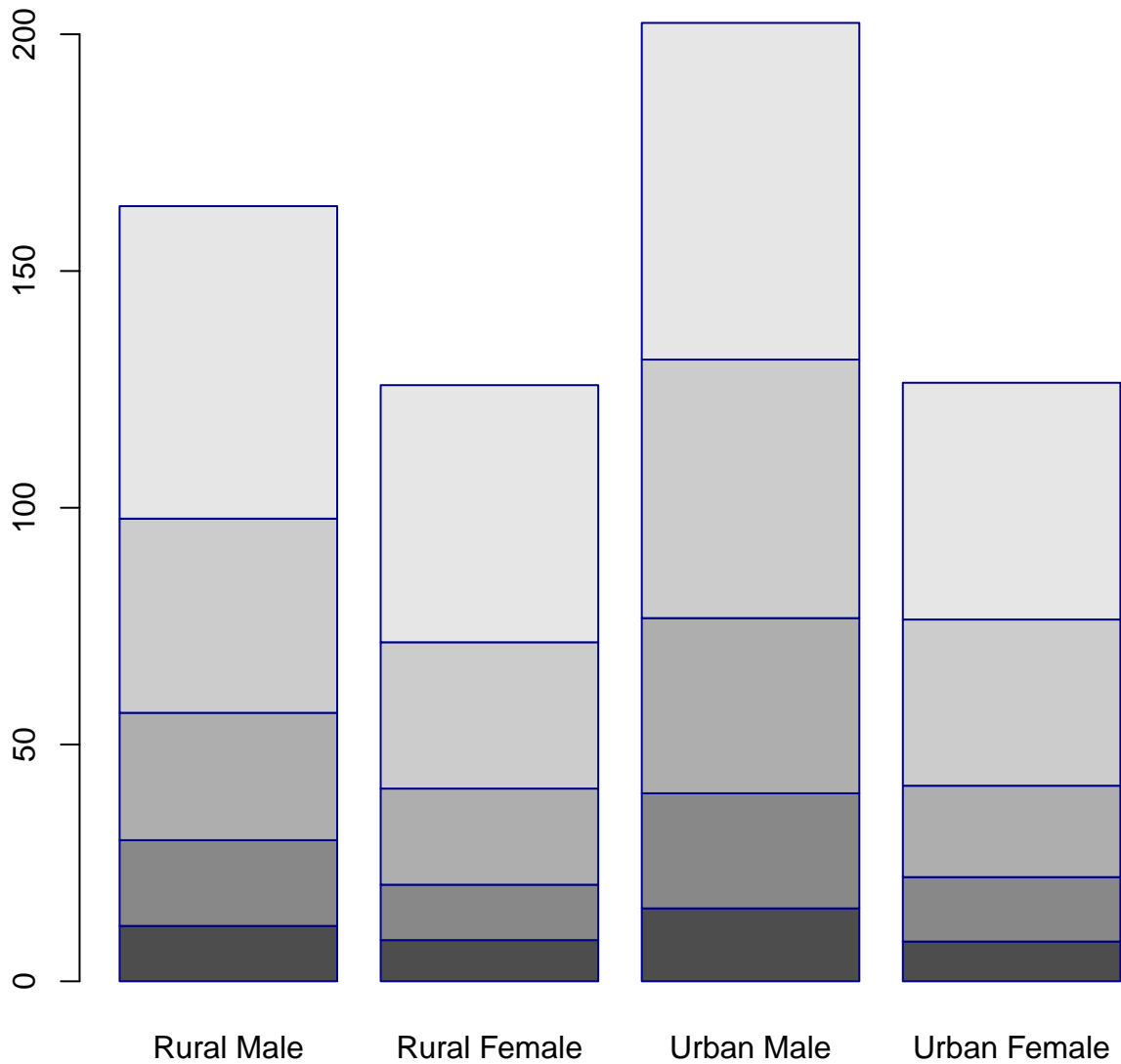
help("barplot")



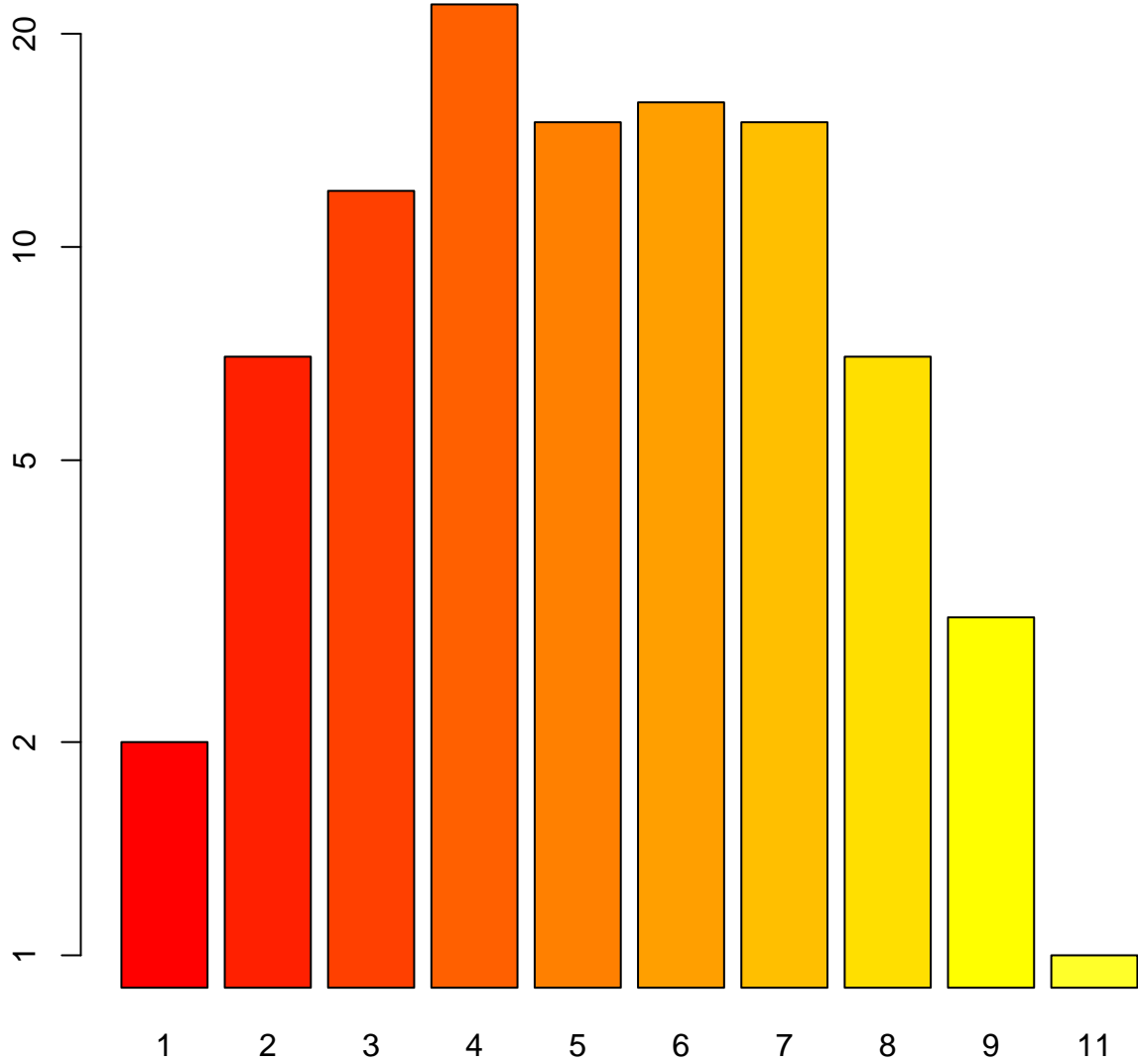
## ***Death Rates in Virginia***



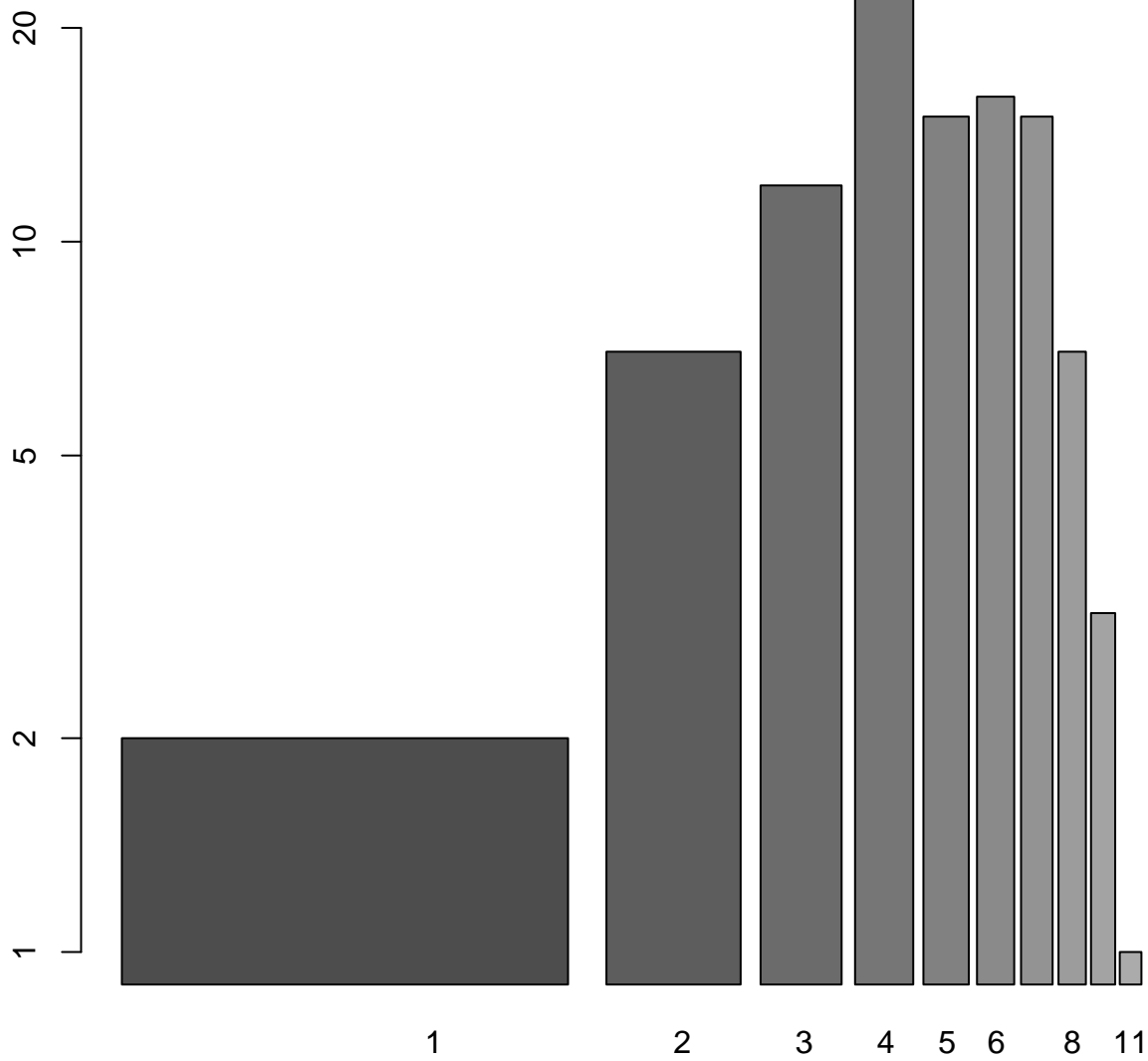
help("barplot")



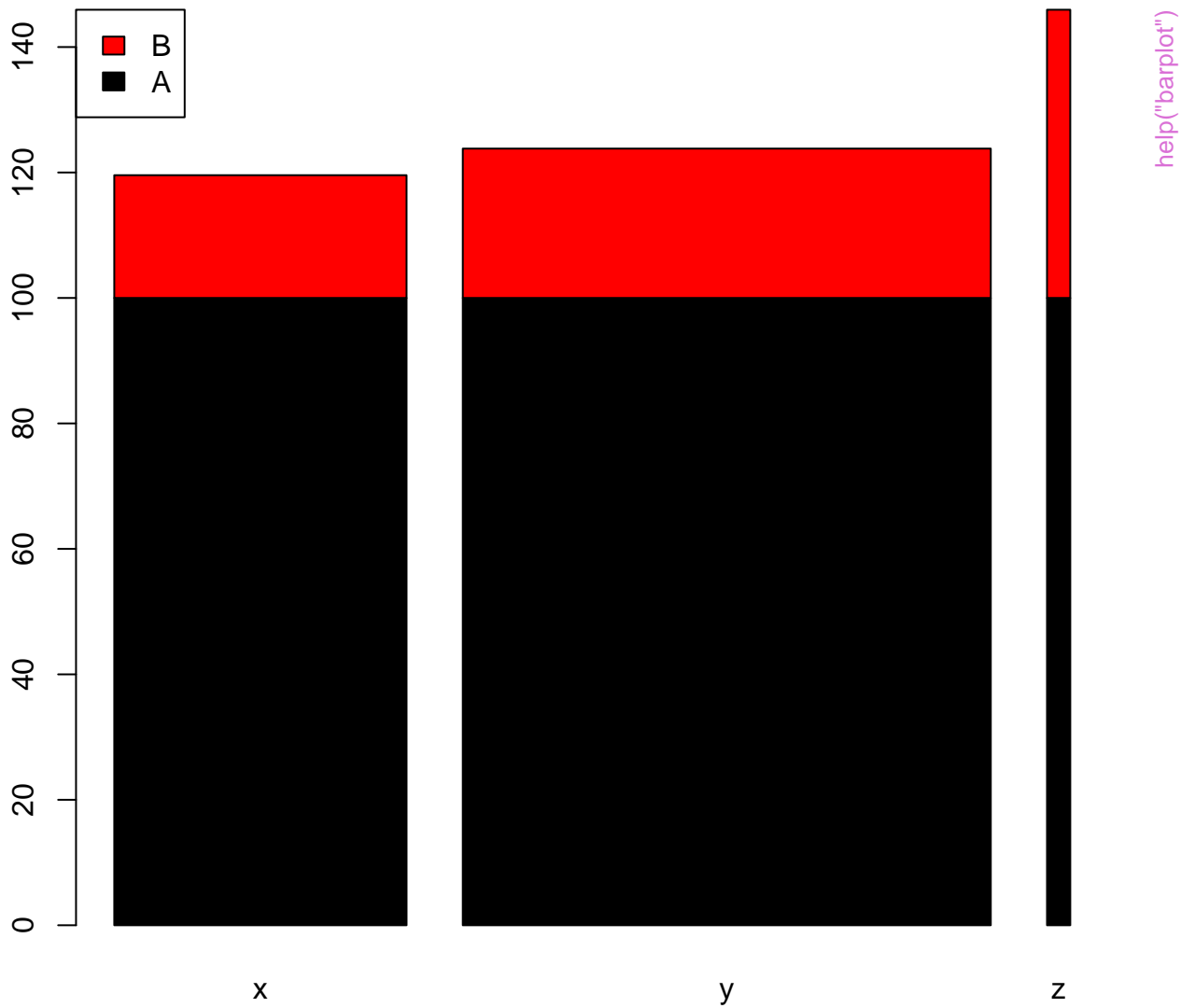
help("barplot")



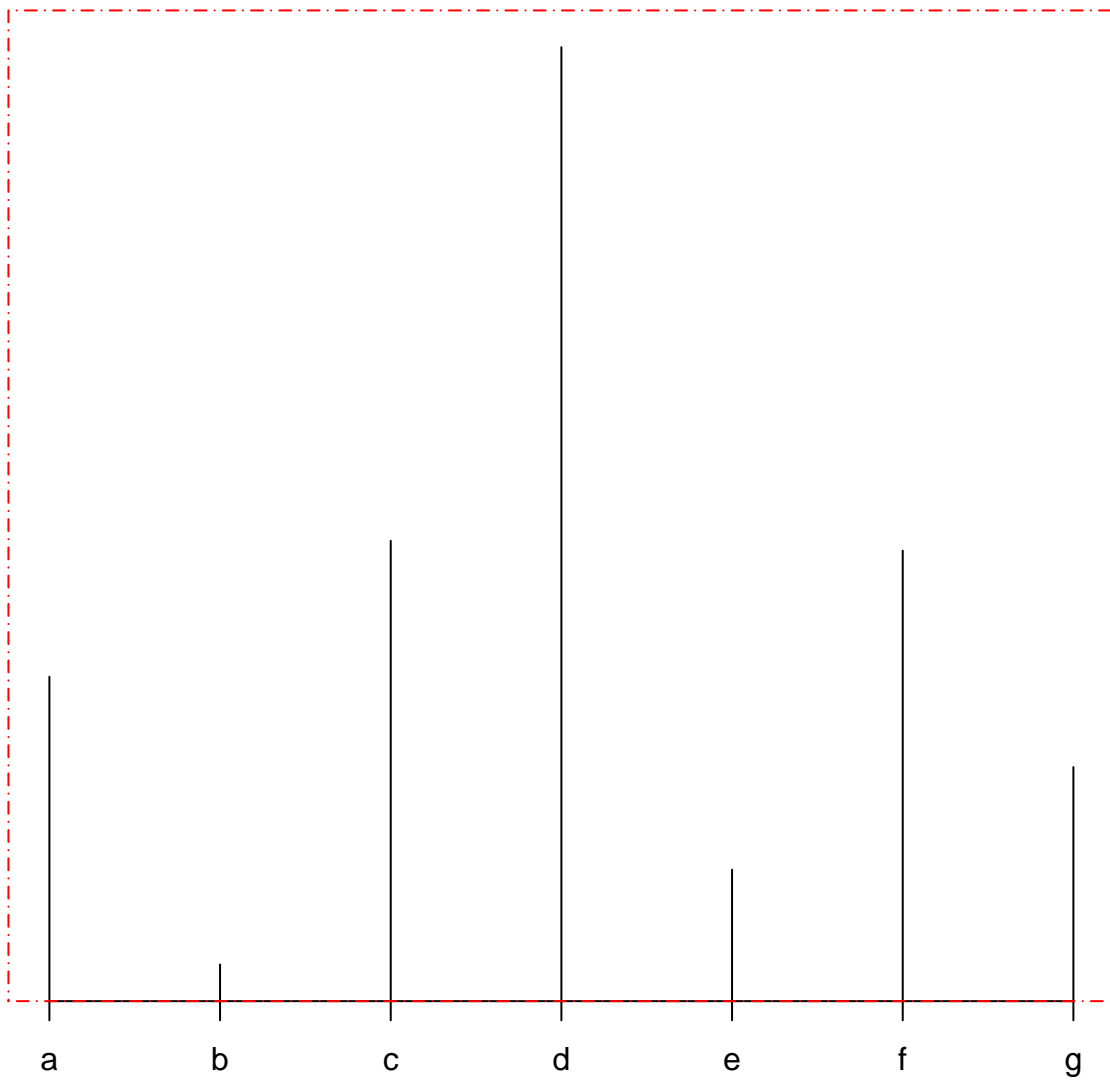
`help("barplot")`



`help("barplot")`

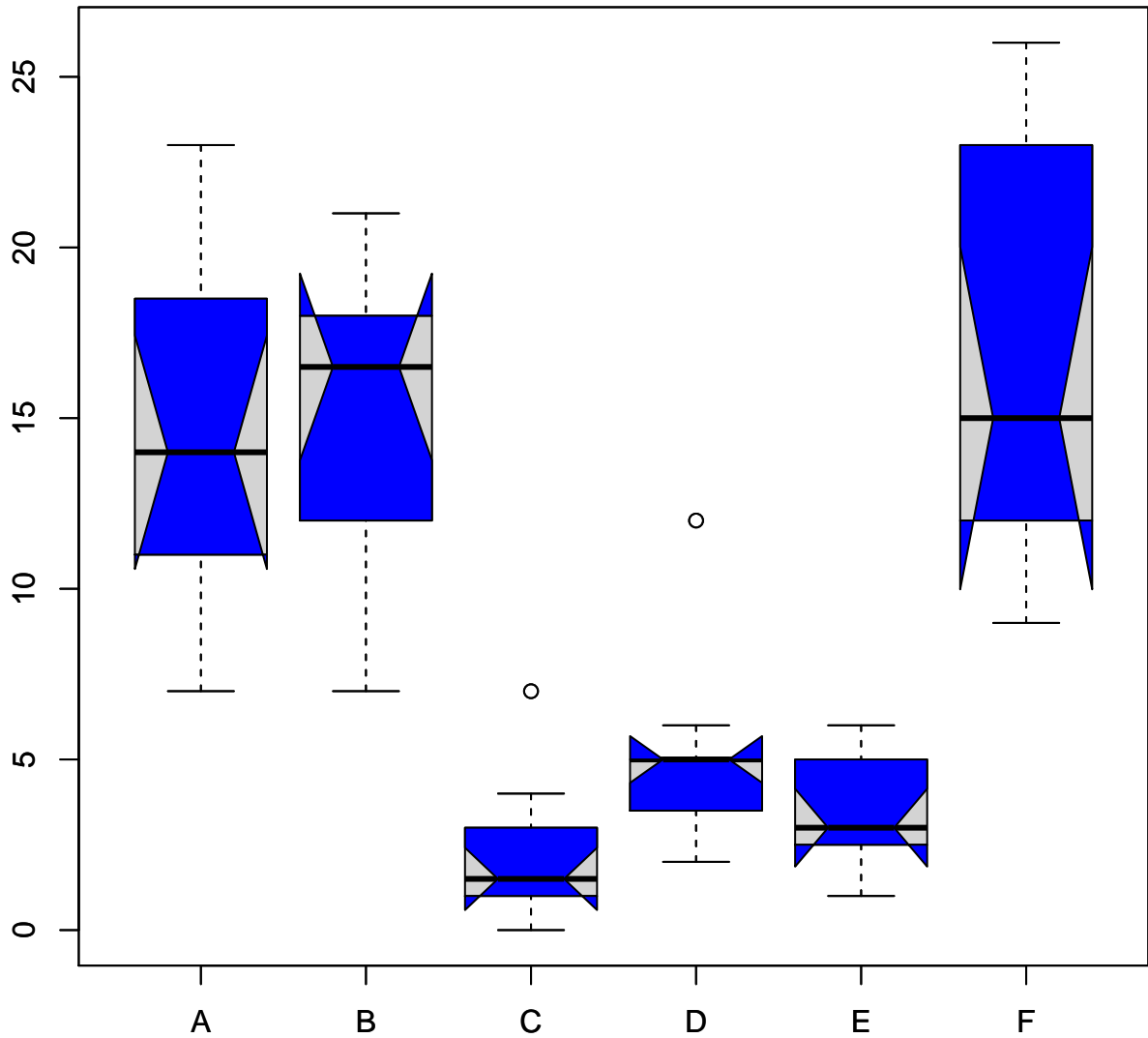


abs(stats::rnorm(7))

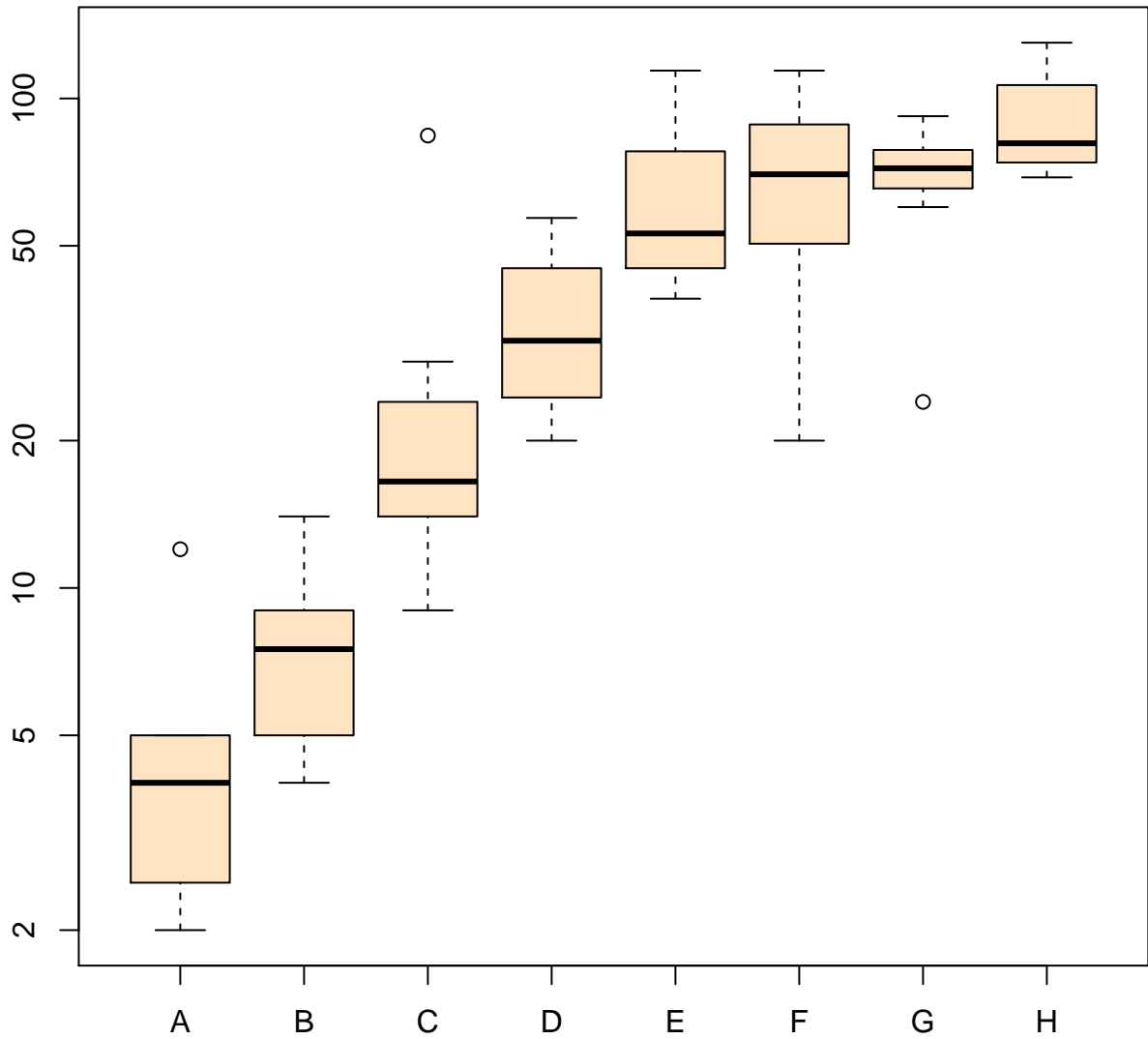


1:7

help("box")



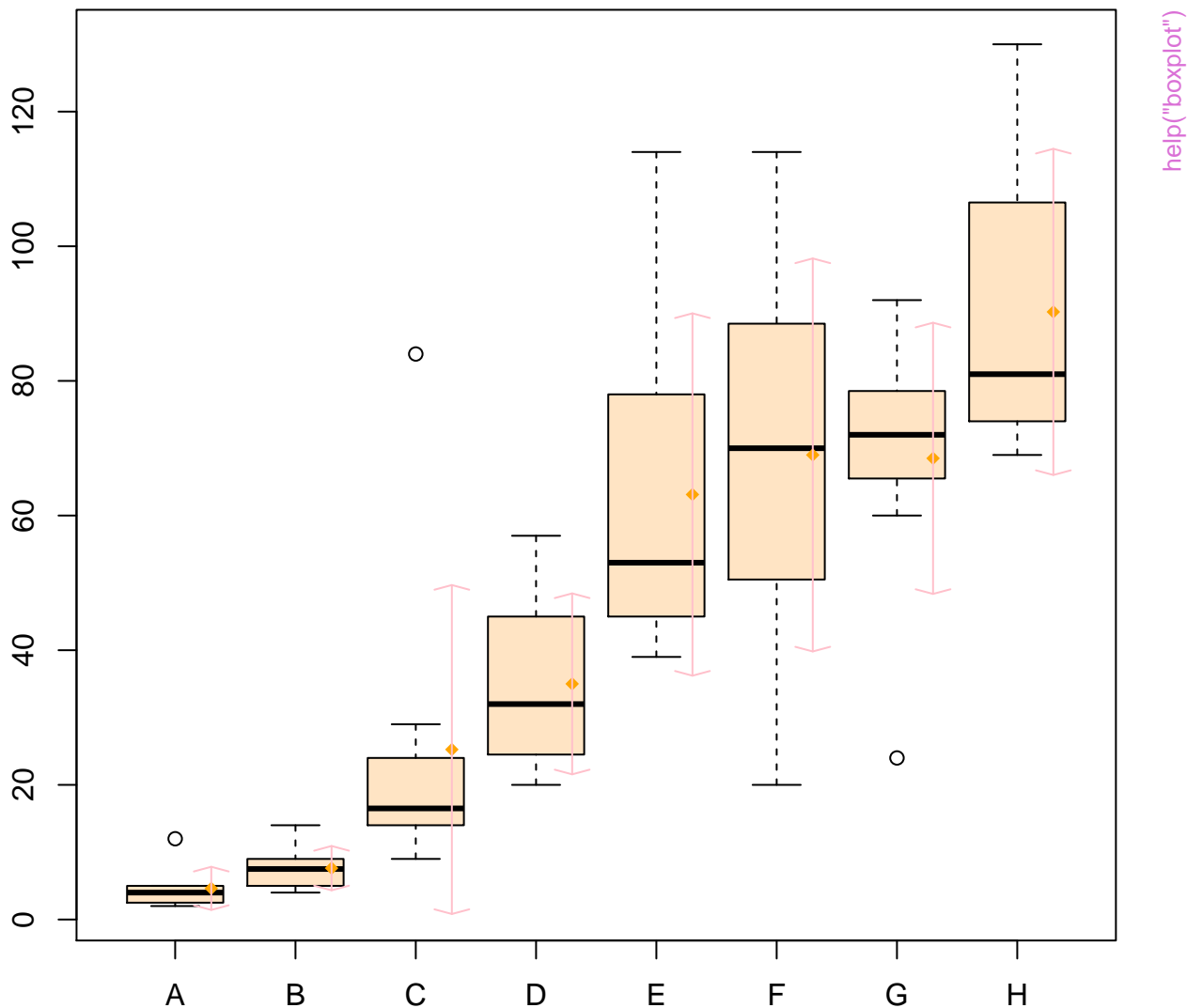
help("boxplot")



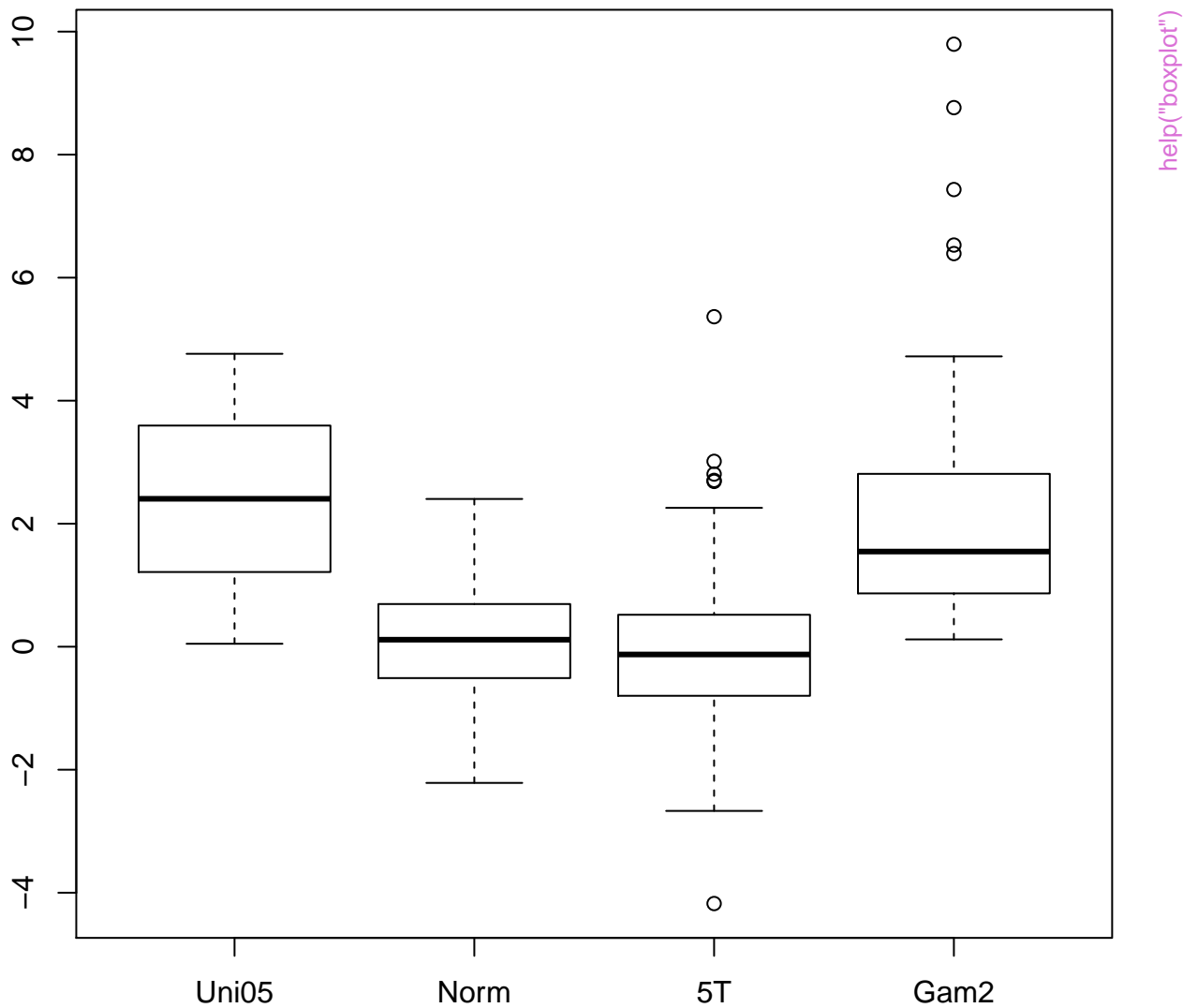
[help\("boxplot"\)](#)



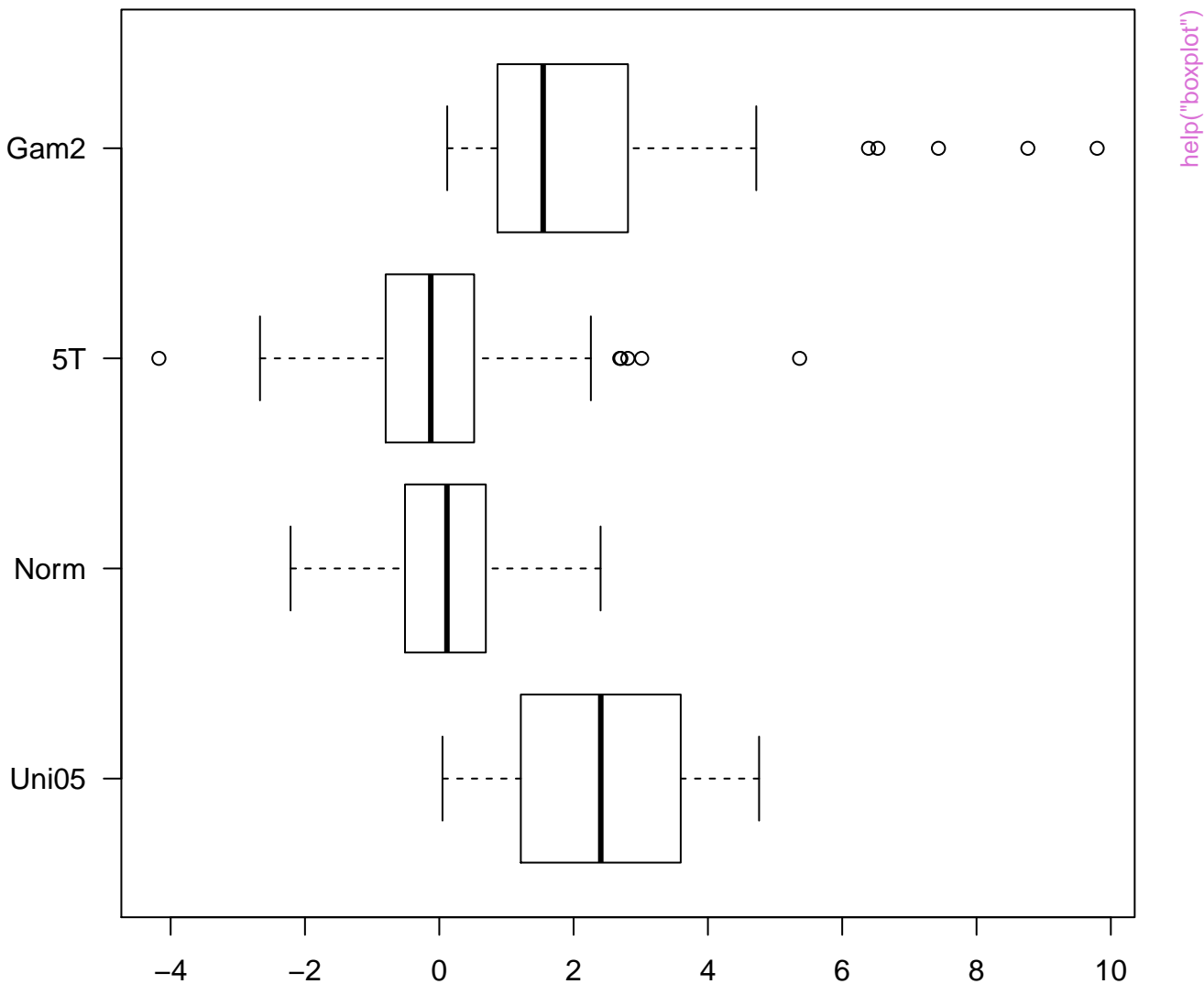
# Comparing boxplot(s) and non-robust mean $\pm$ SD



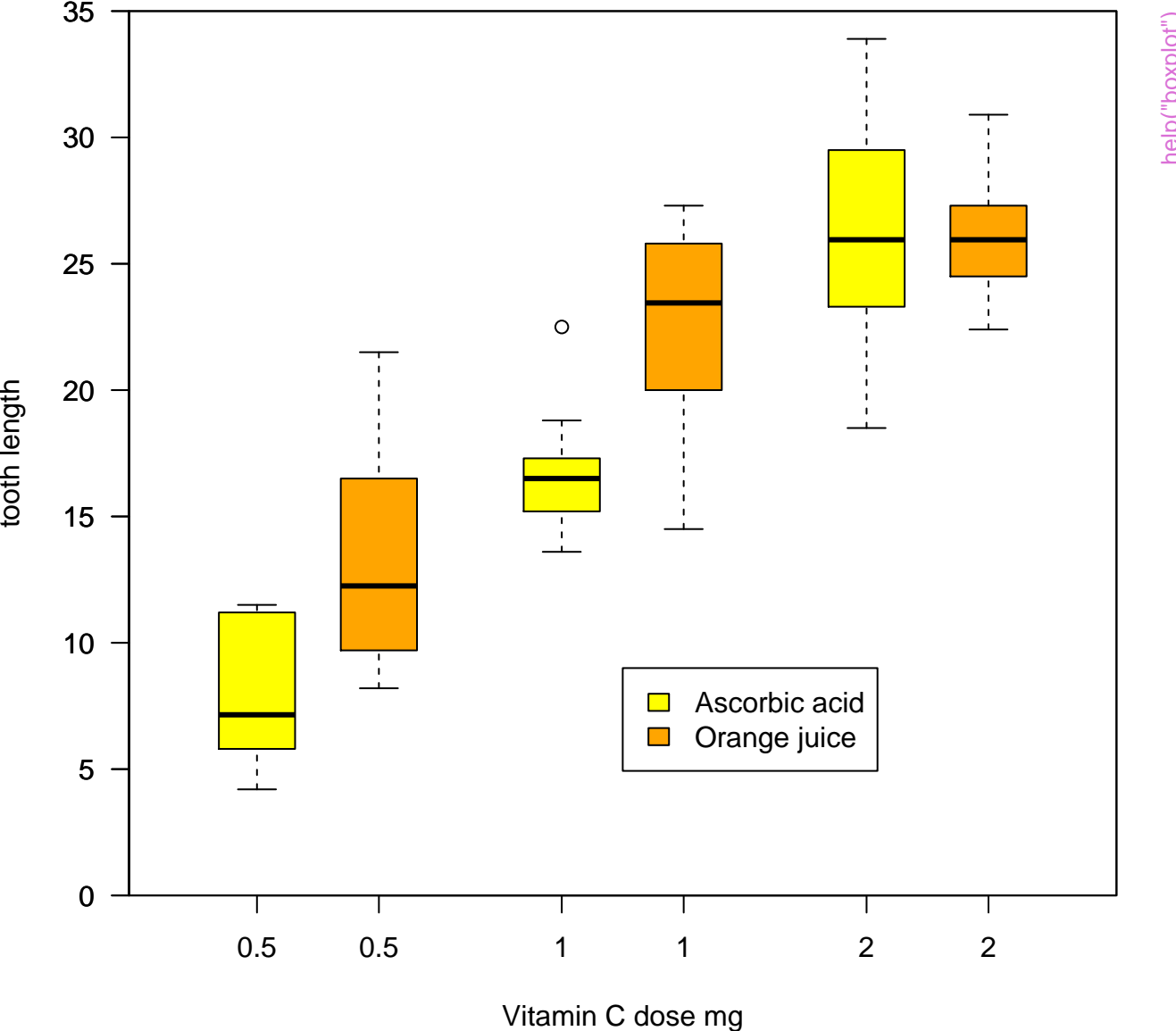
**boxplot(as.data.frame(mat), main = ...)**



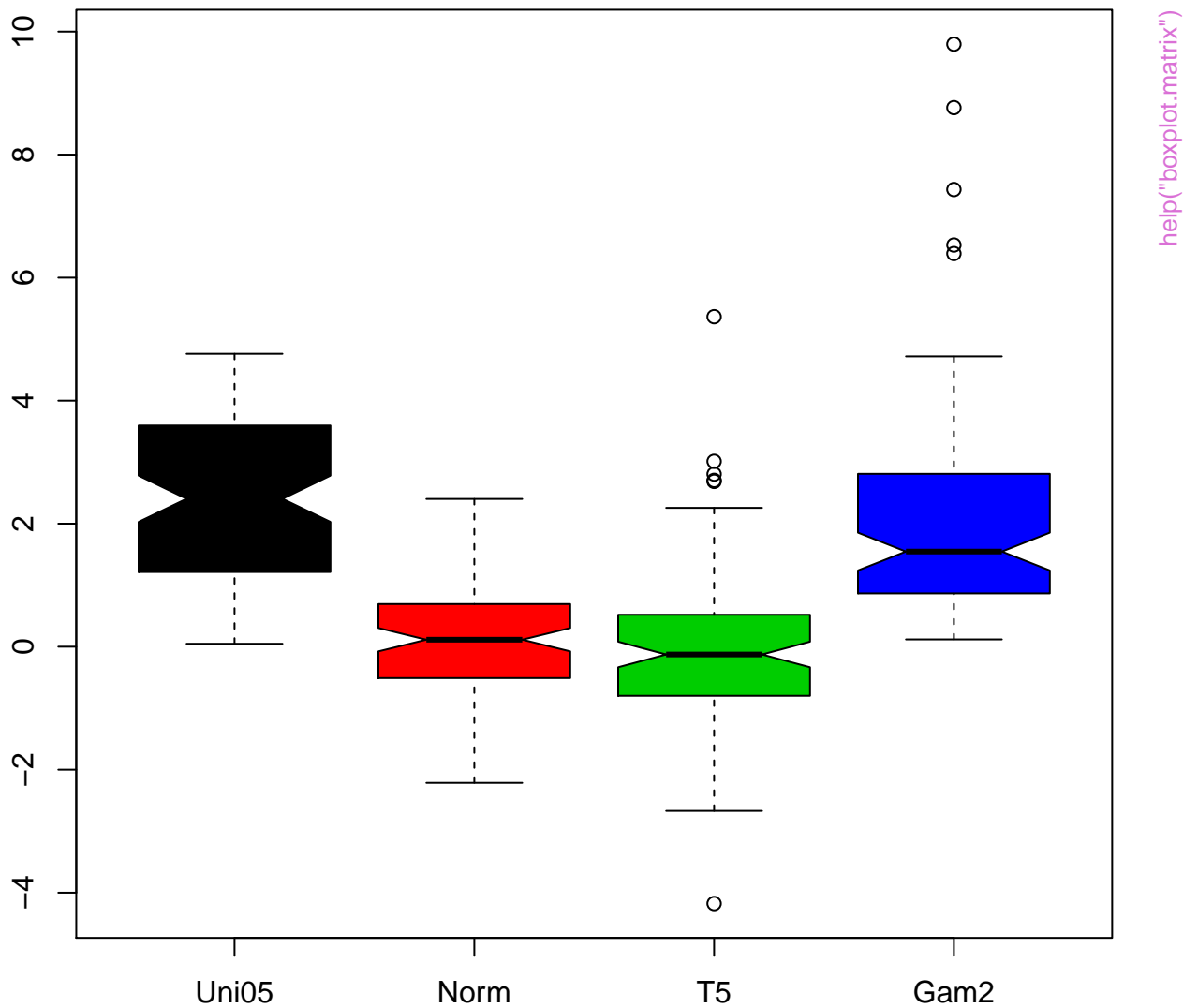
**boxplot(\*, horizontal = TRUE)**



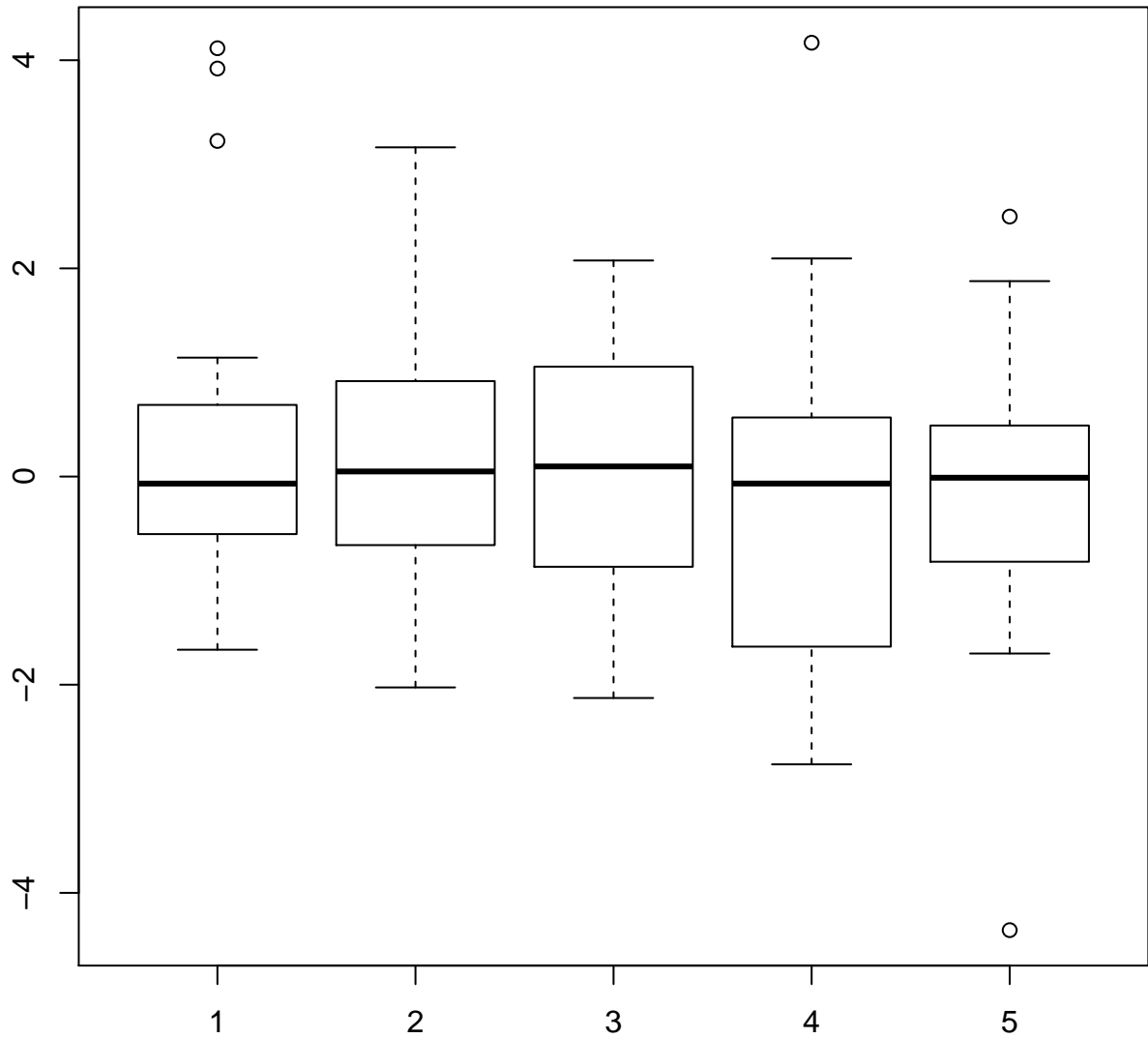
Guinea Pigs' Tooth Growth



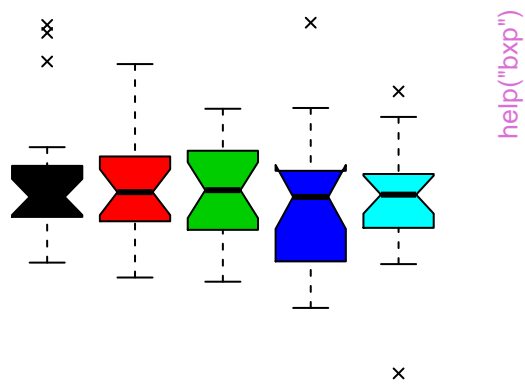
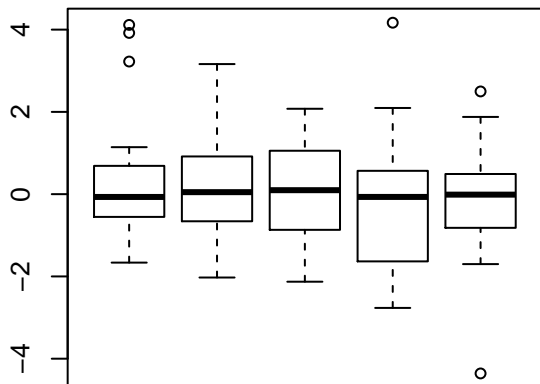
**boxplot.matrix(...., main = ...)**



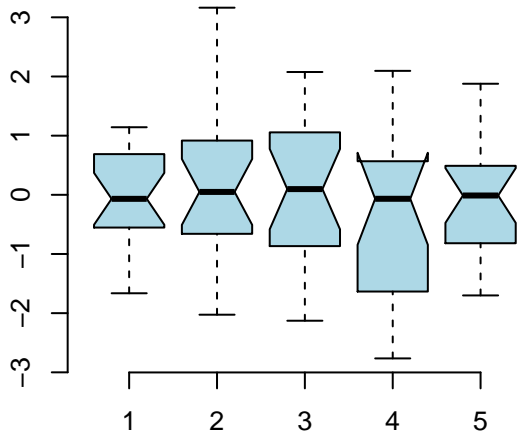
[help\("boxplot.matrix"\)](#)



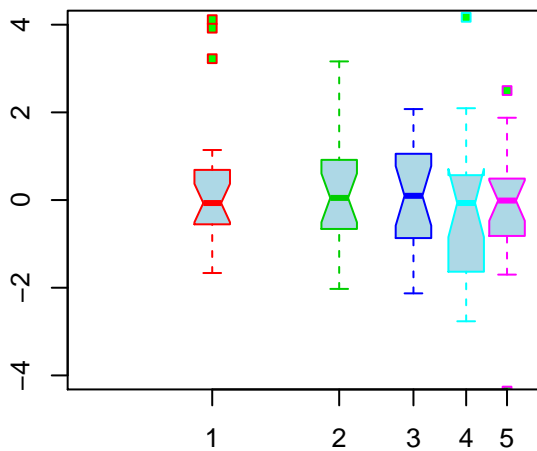
help("bxp")

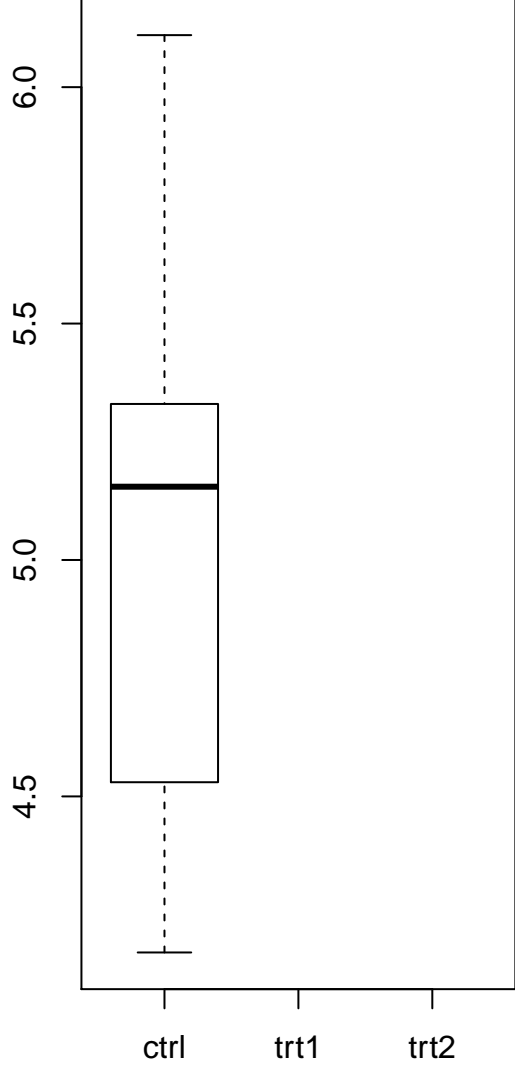
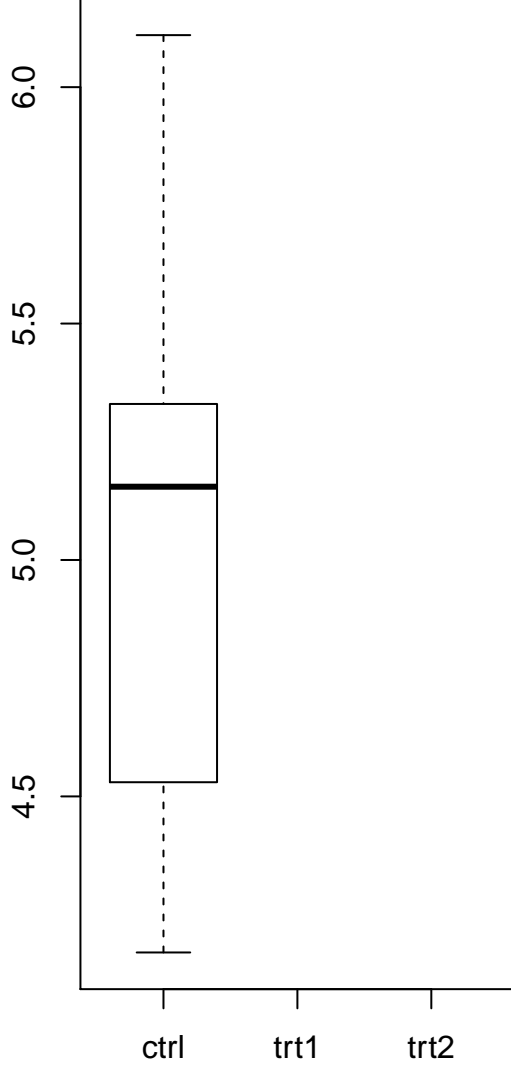


**bxp(\*, frame= FALSE, outl= FALSE)**



**... log = 'x', ylim = \***

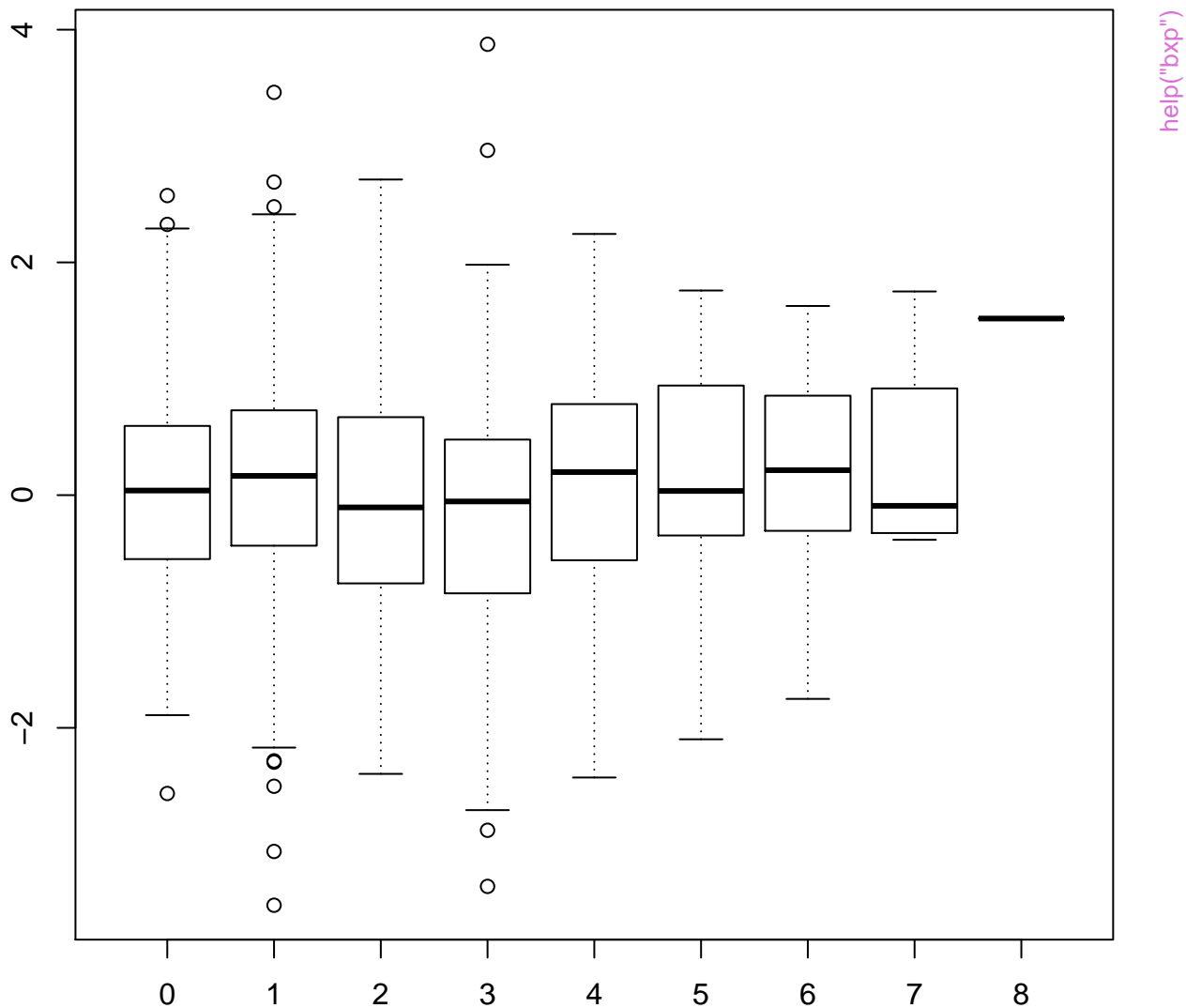




help("bxp")

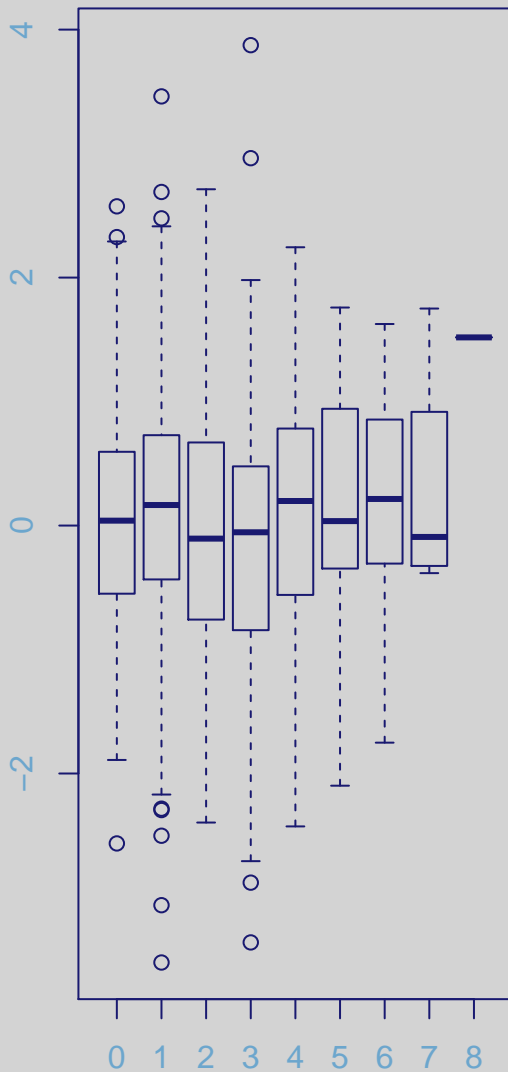


**boxplot(z, whisklty = 3)**

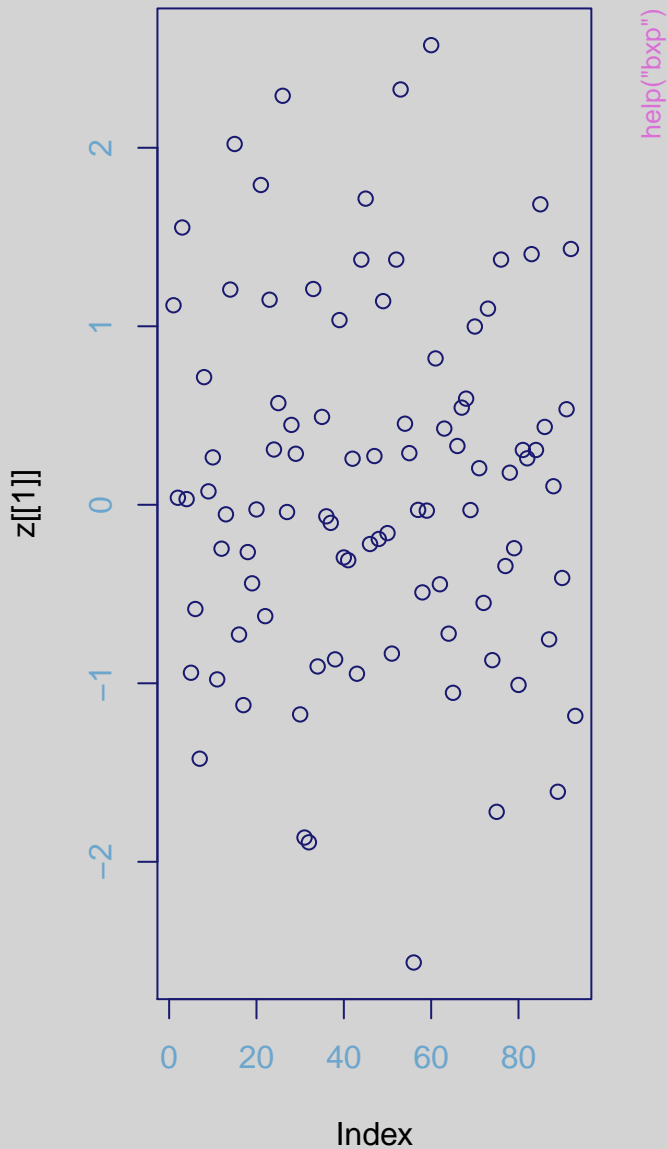


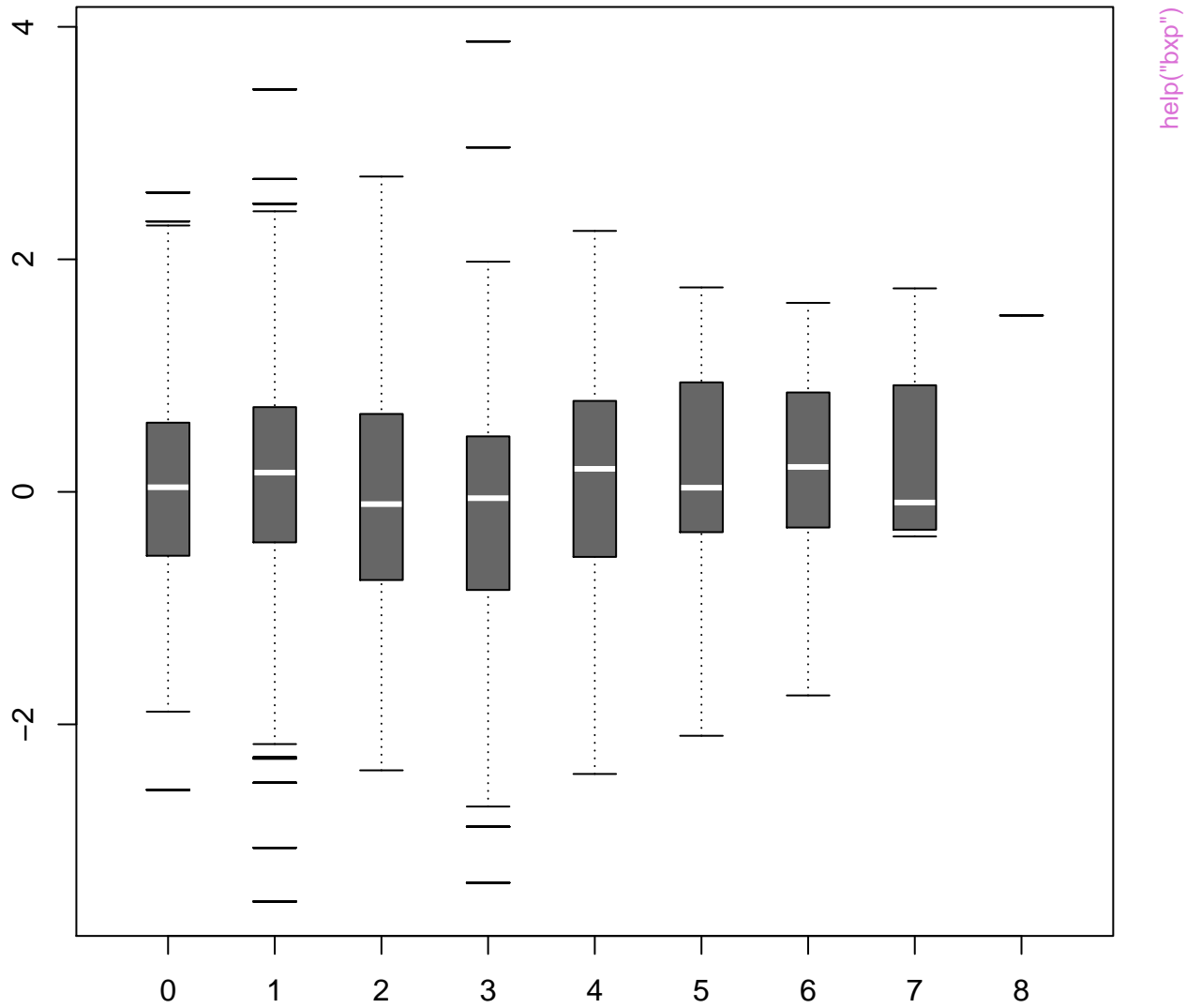
`par(bg="light gray", fg="midnight blue")`

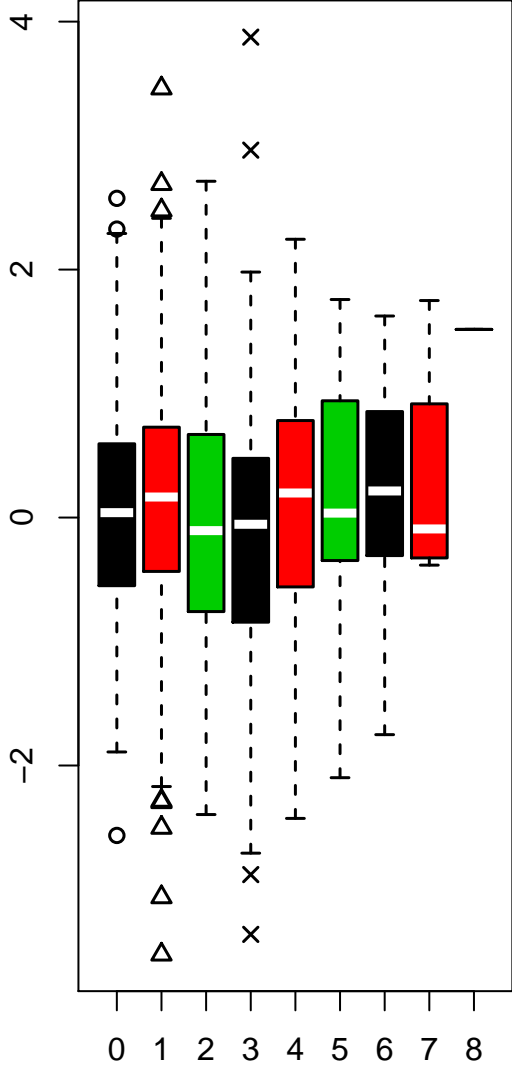
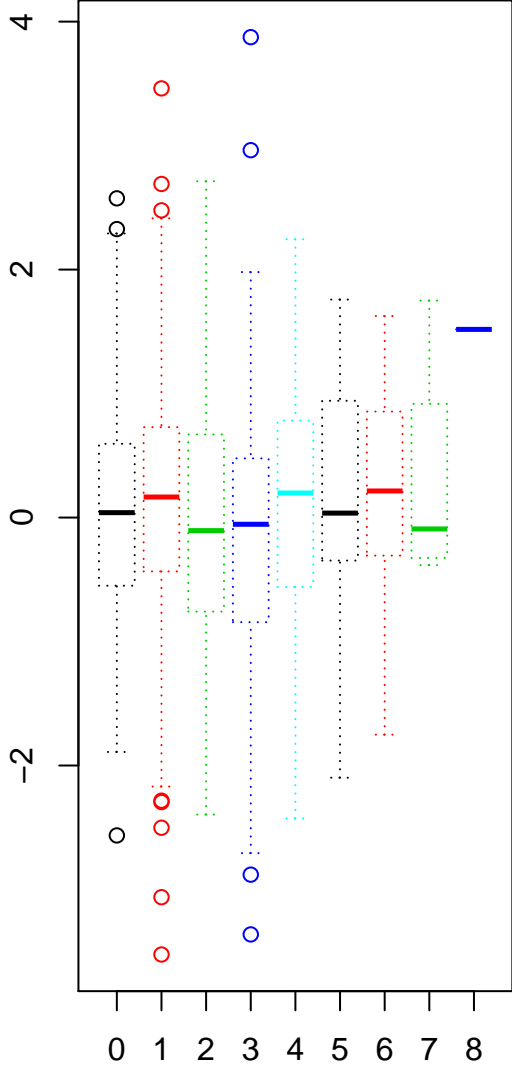
`boxplot(*, col.axis=.., main=..)`



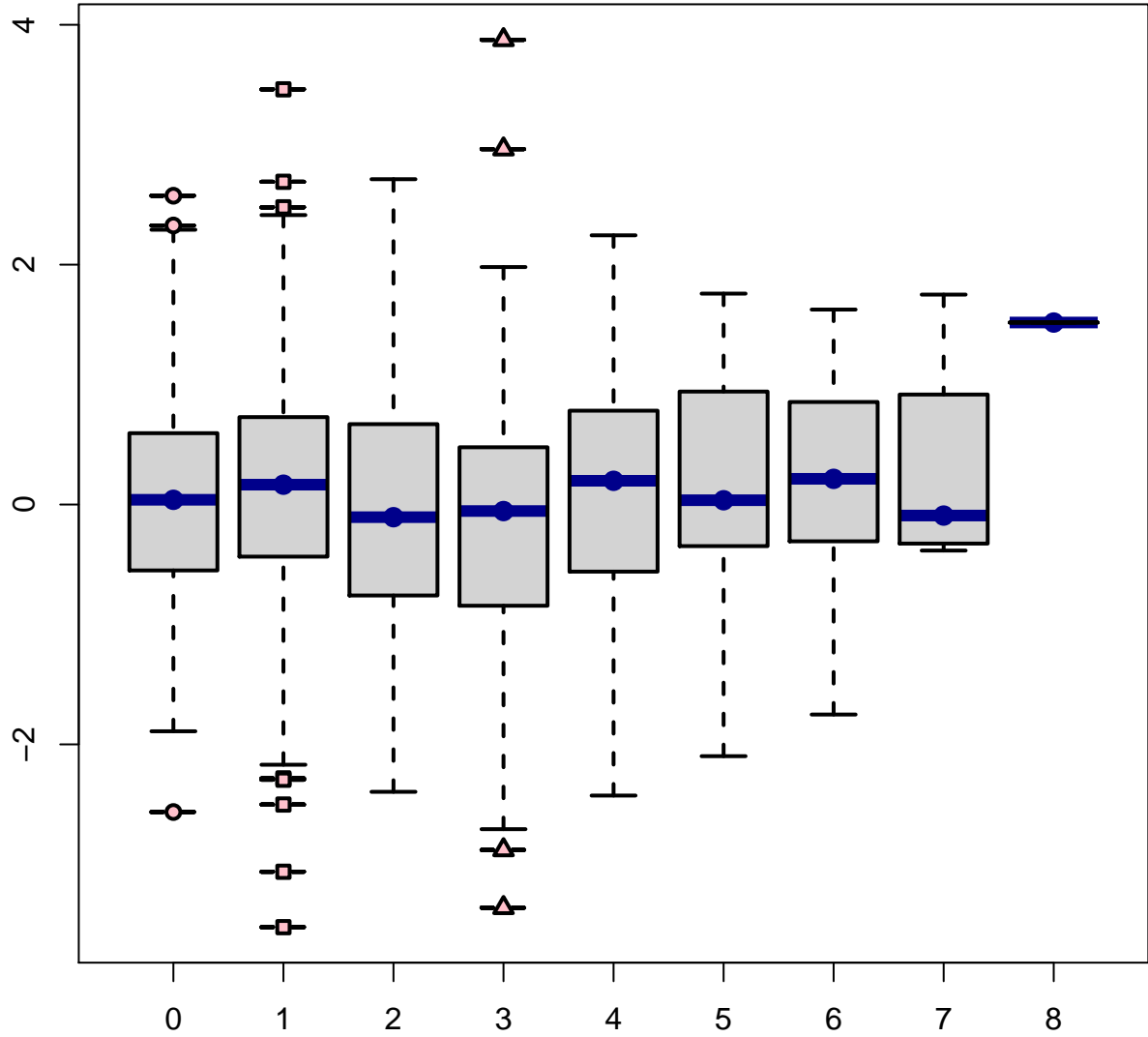
`plot(*, col.axis=.., main=..)`



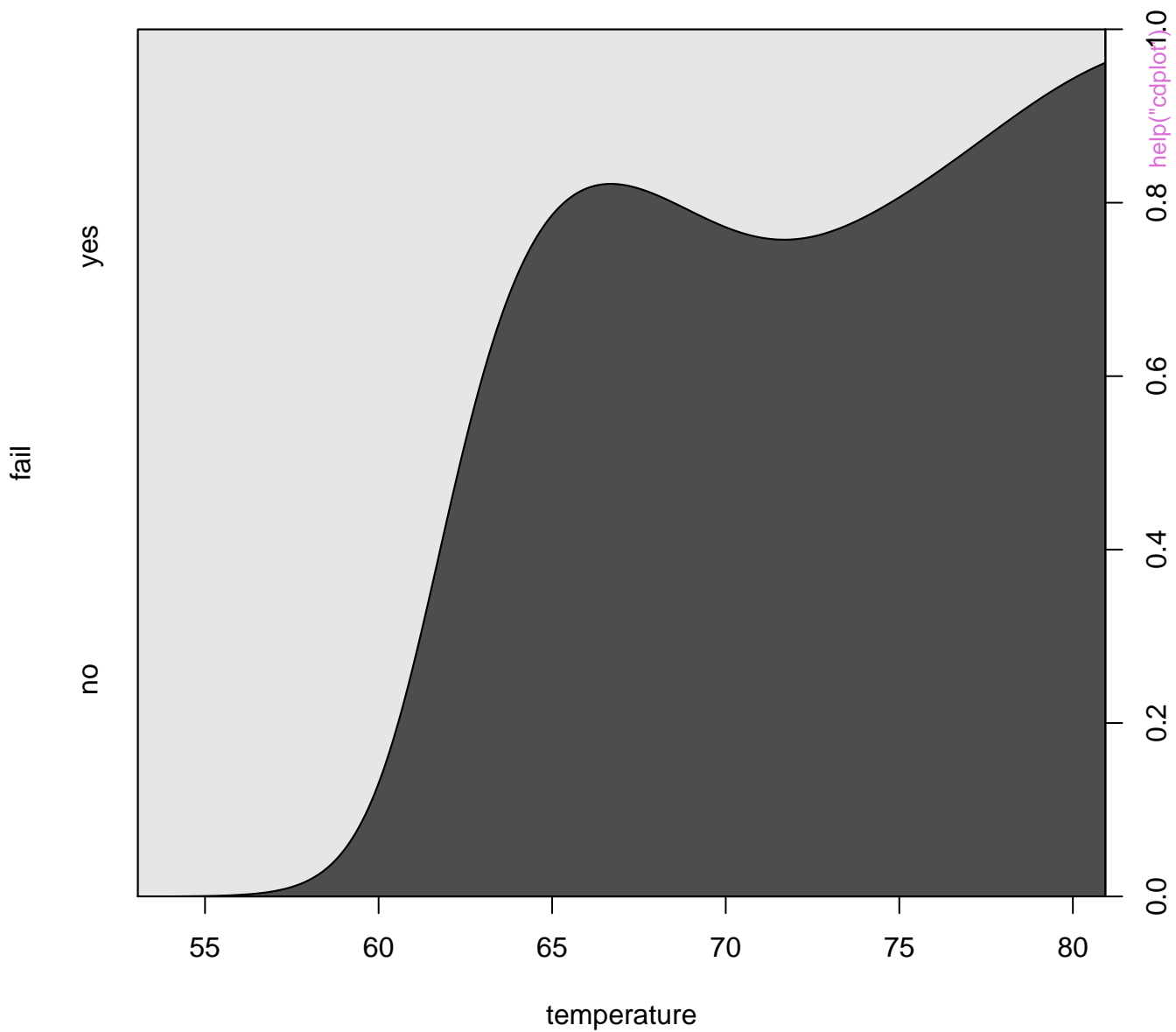


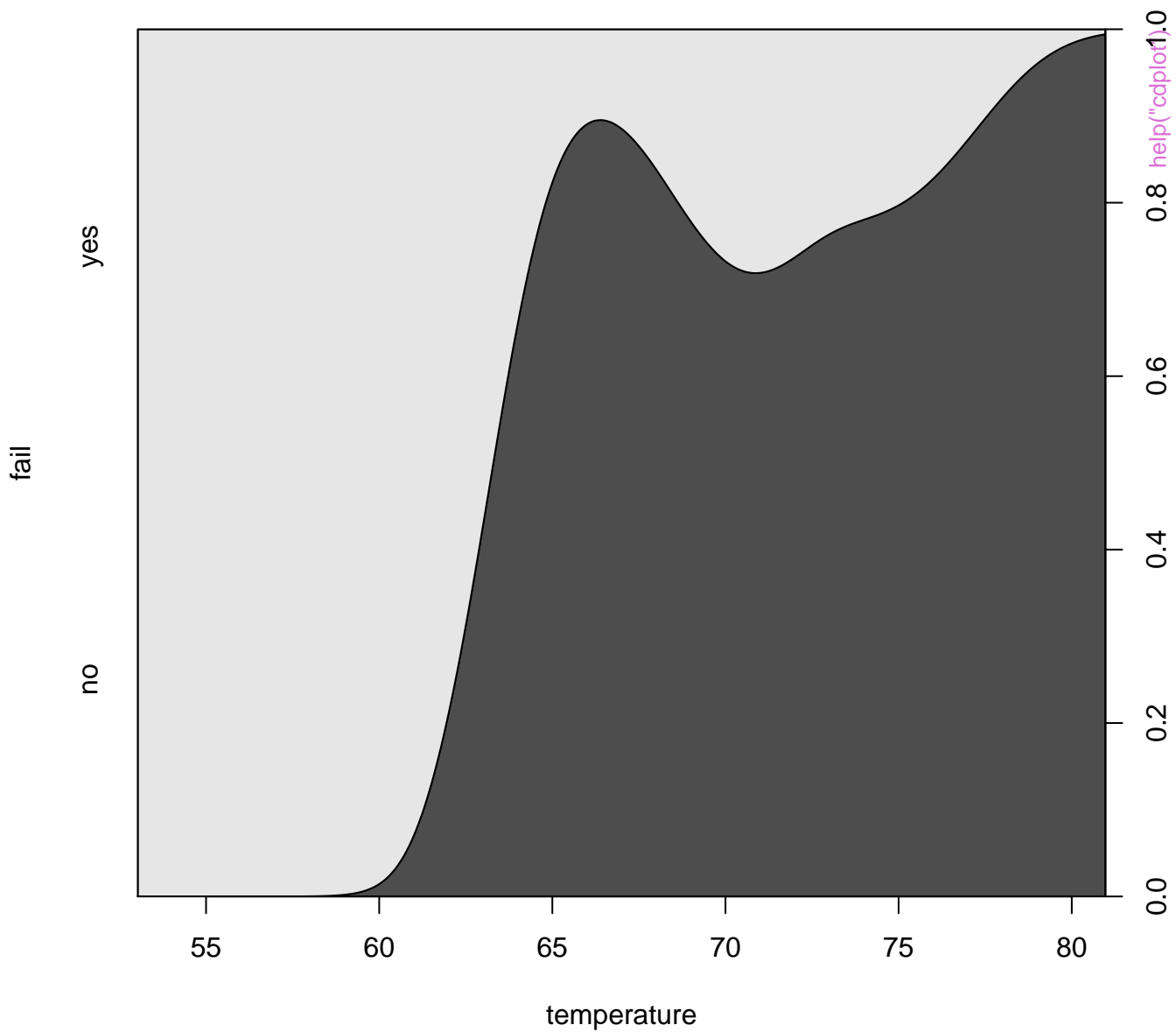


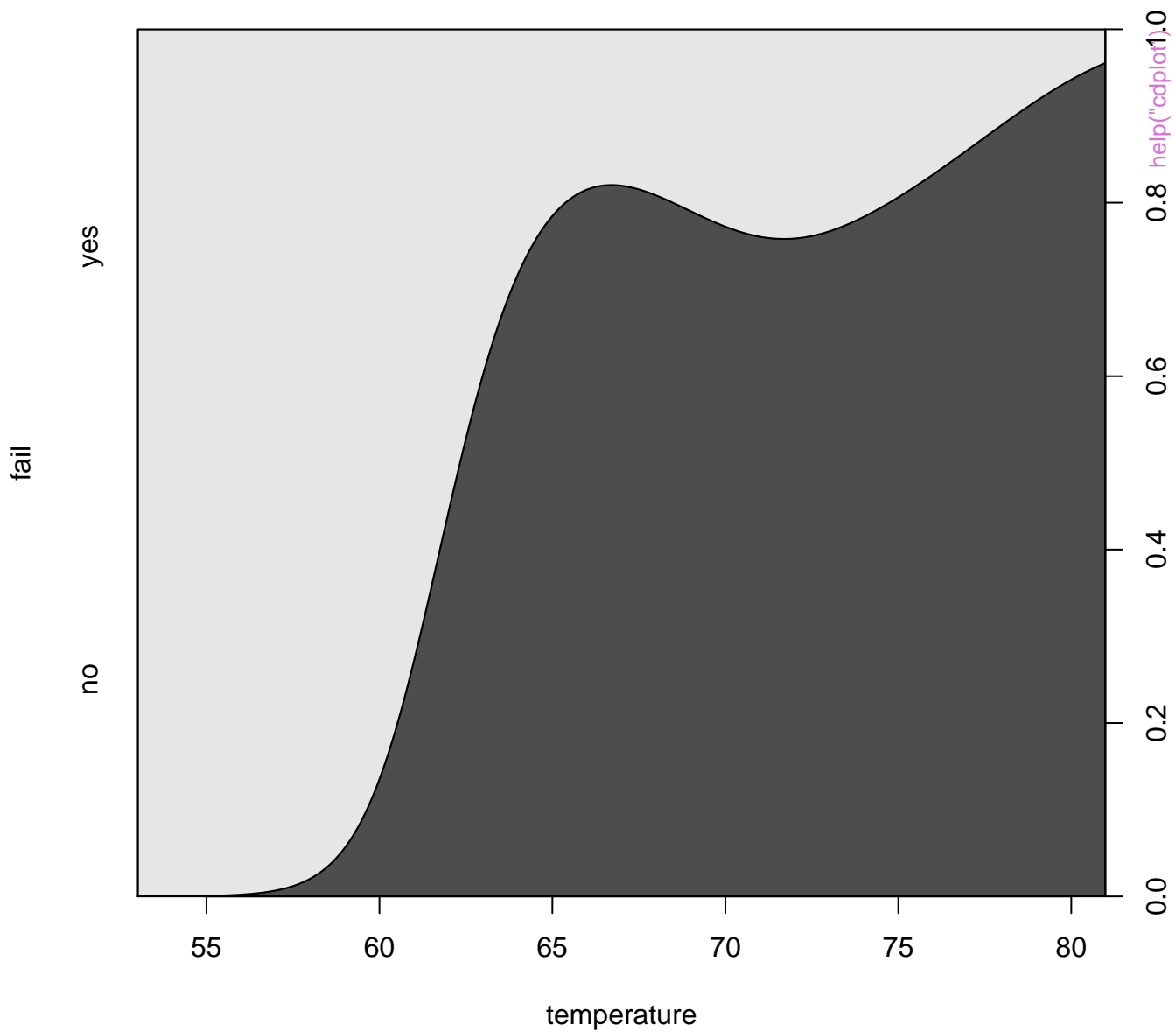
help("bxp")



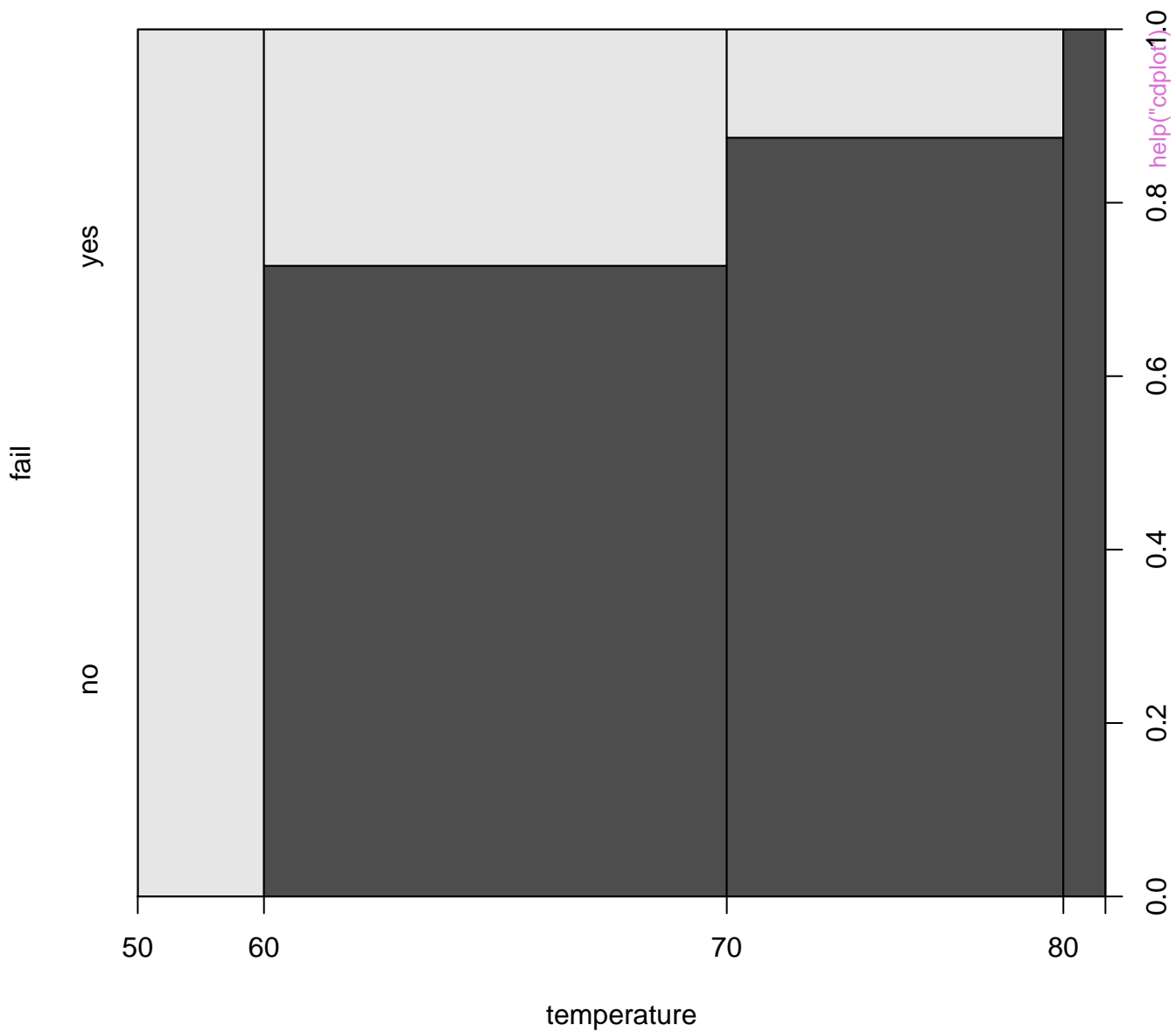
help("bxp")

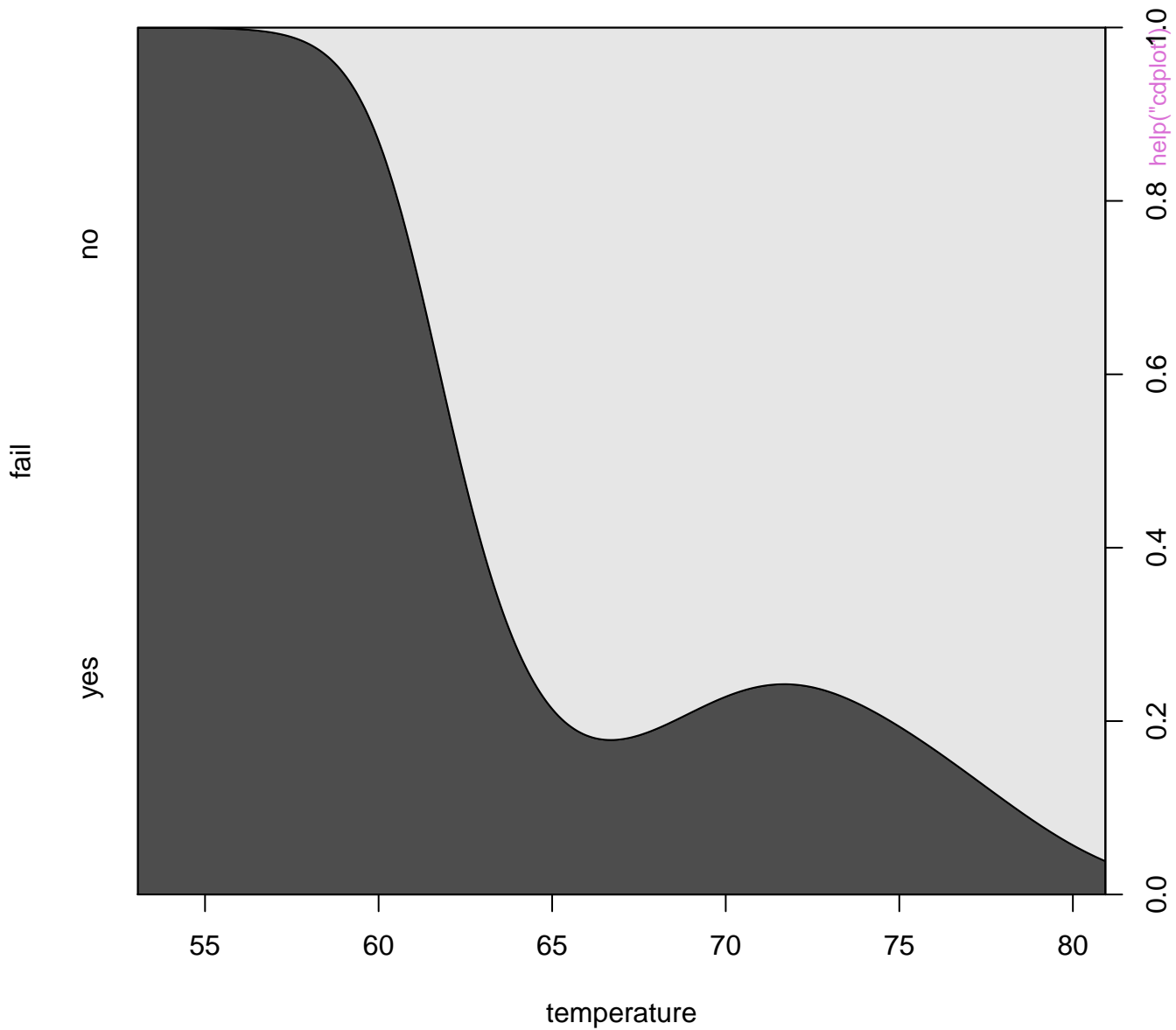


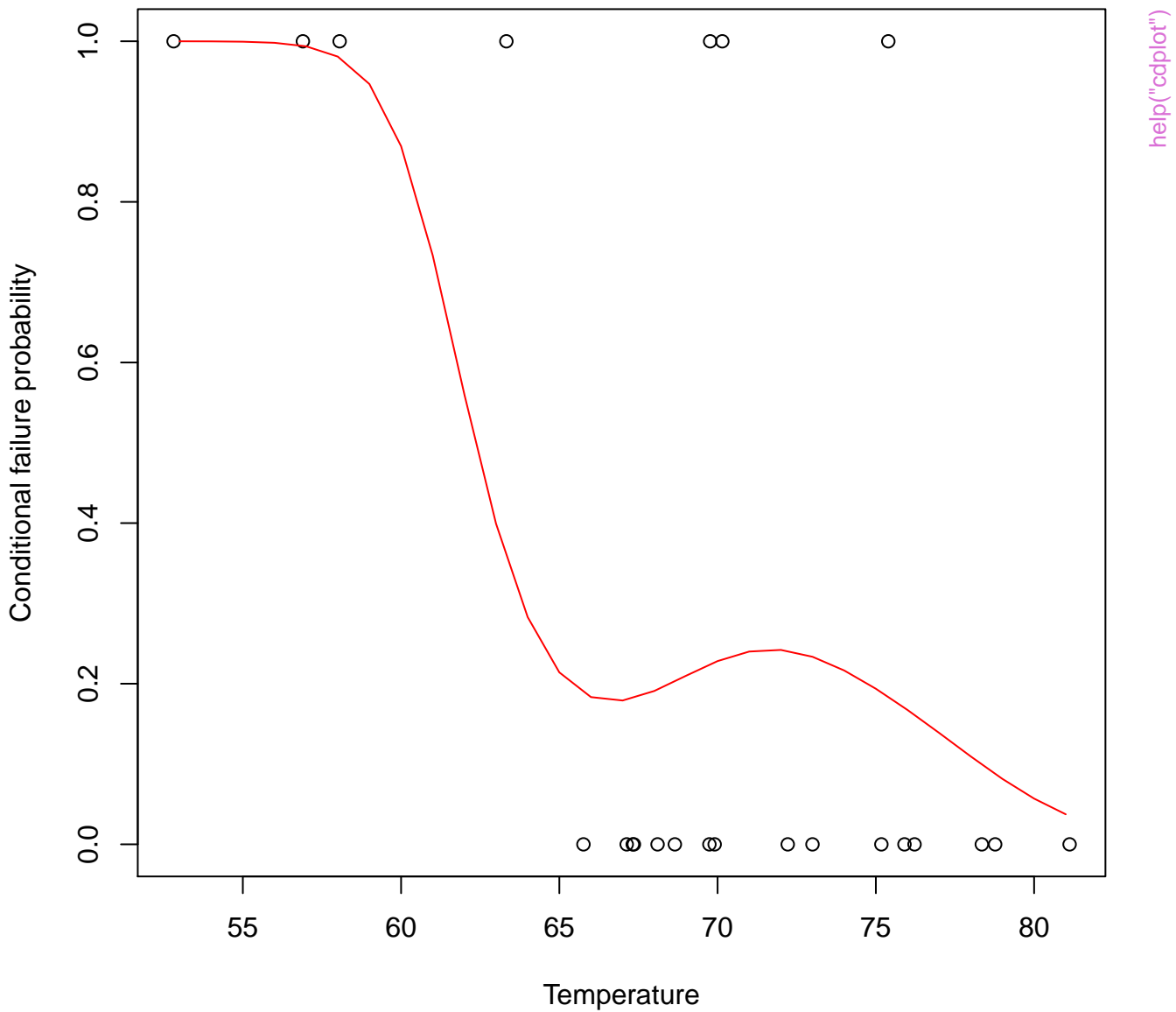




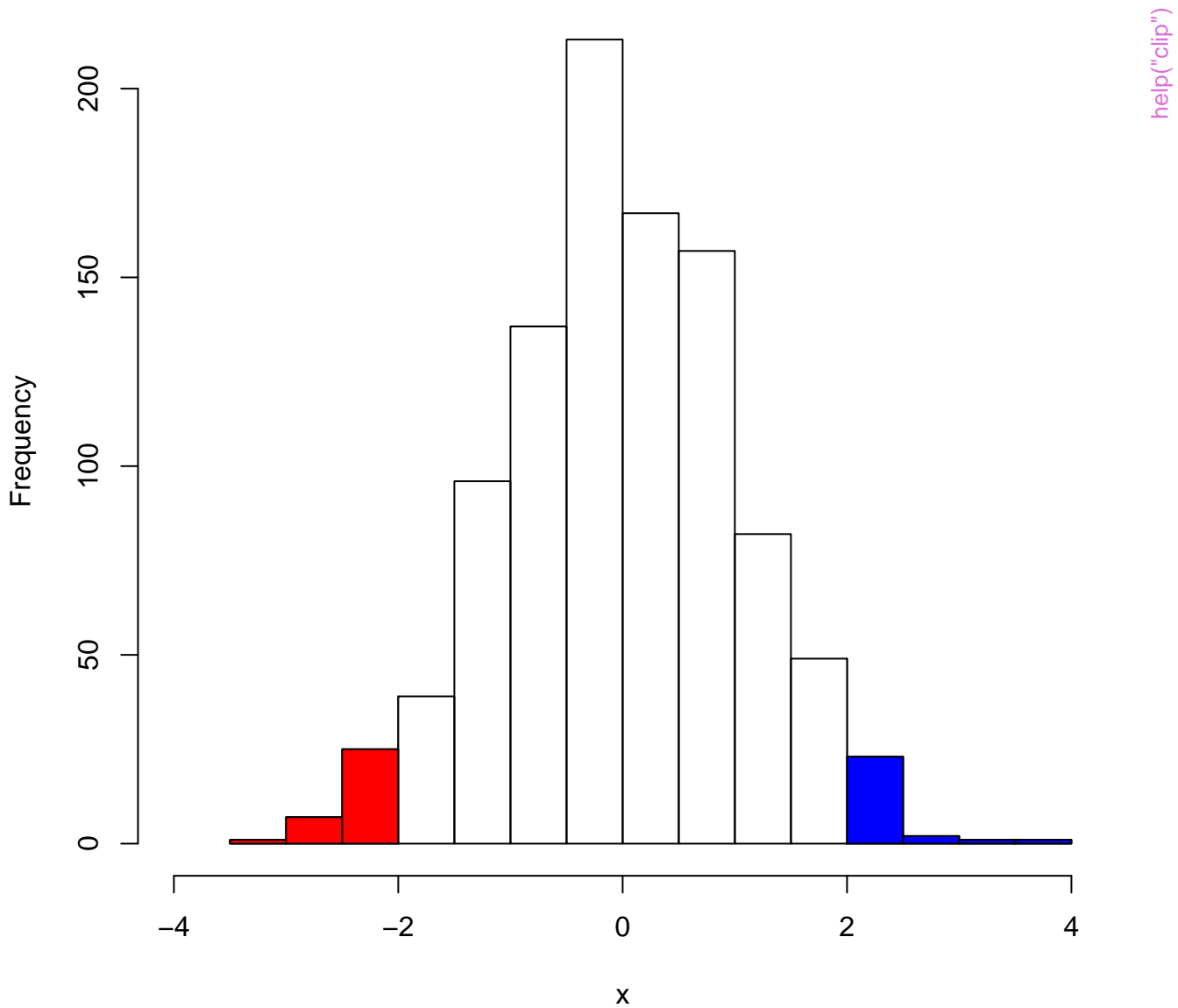


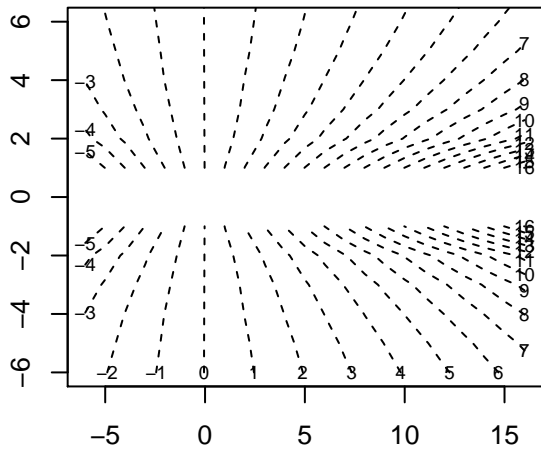
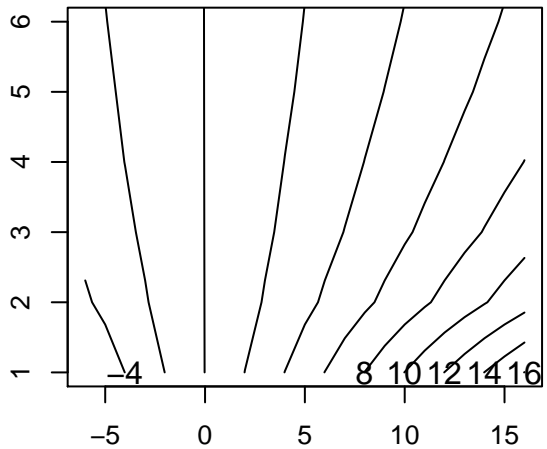
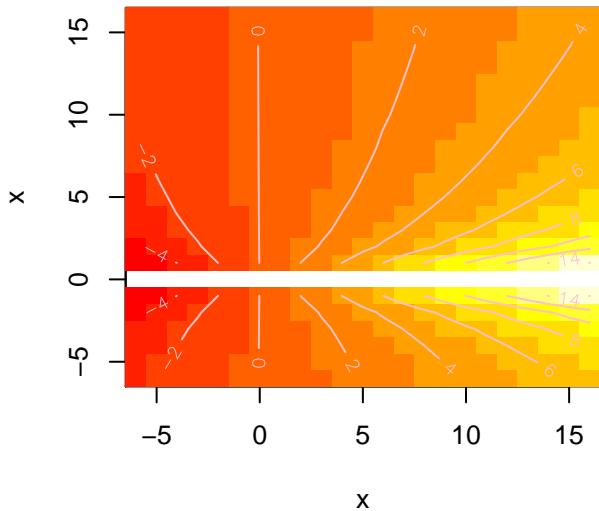
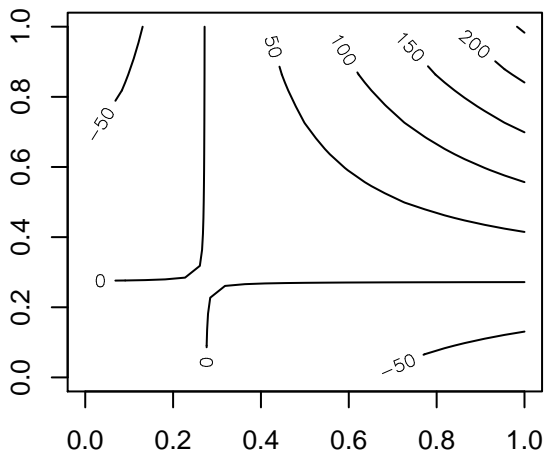




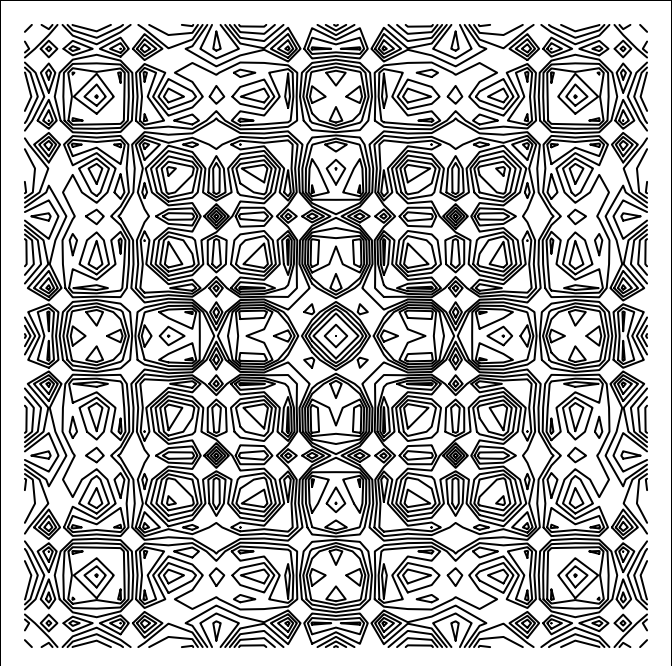
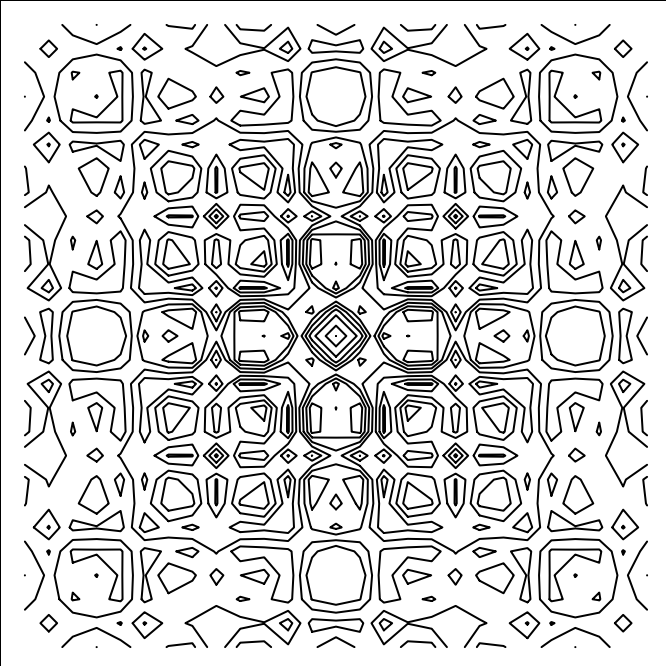
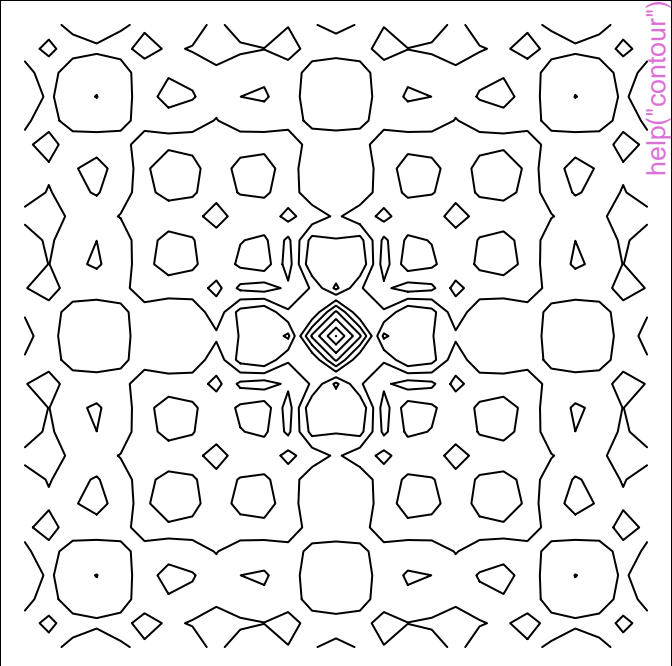
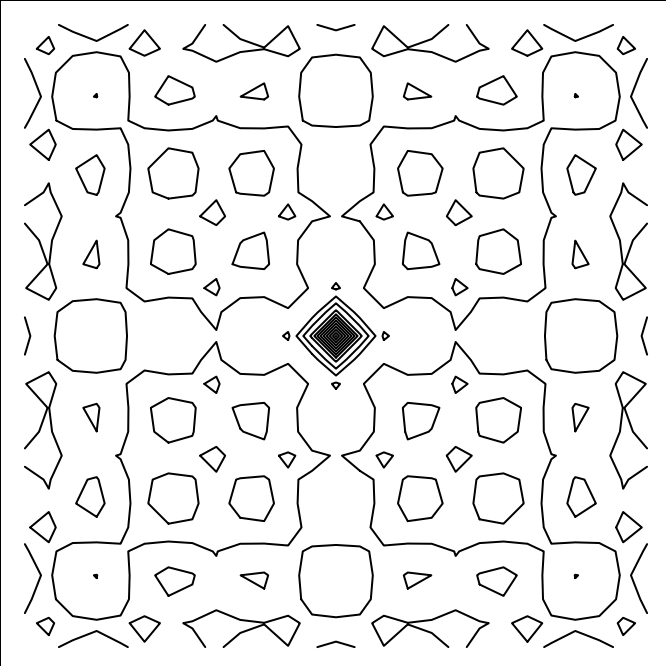


Histogram of x

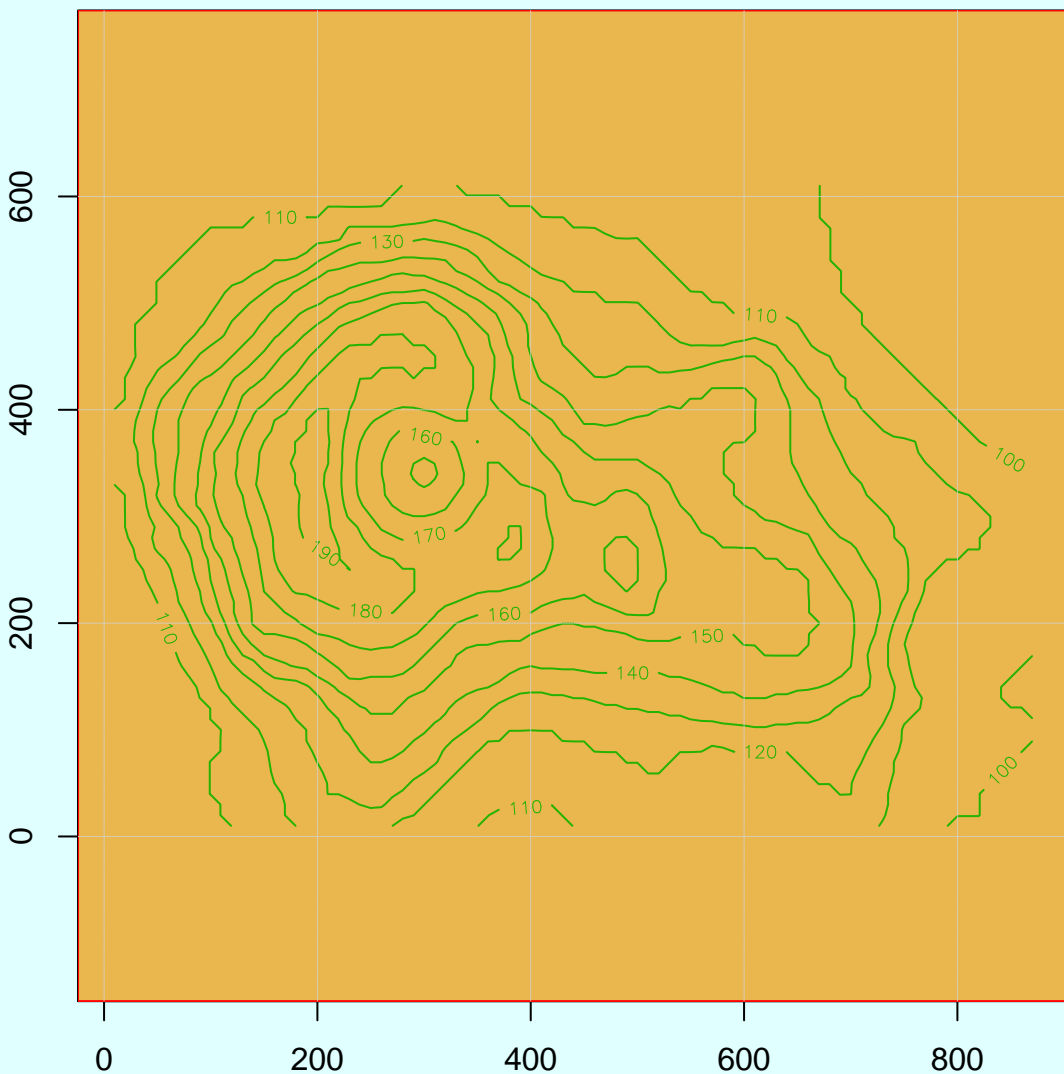




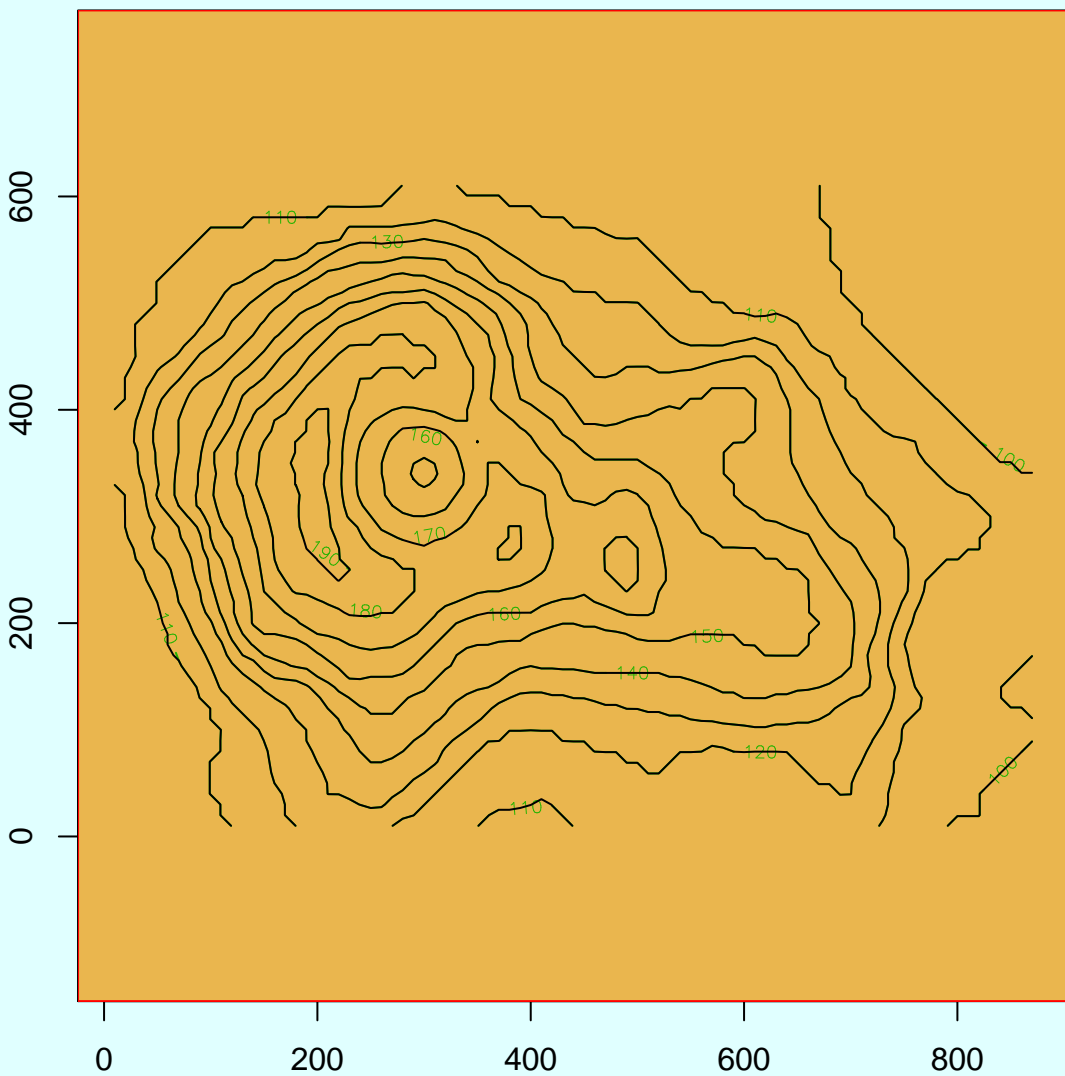
```
help("contour")
```



# A Topographic Map of Maunga Whau

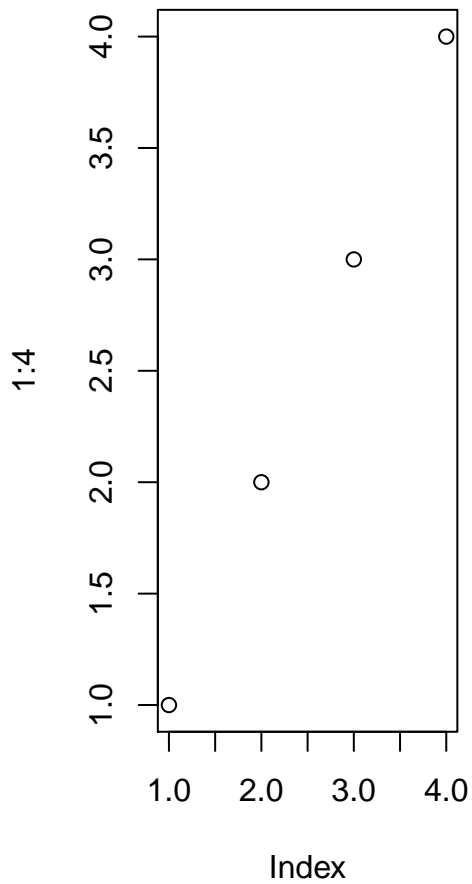


help("contour")

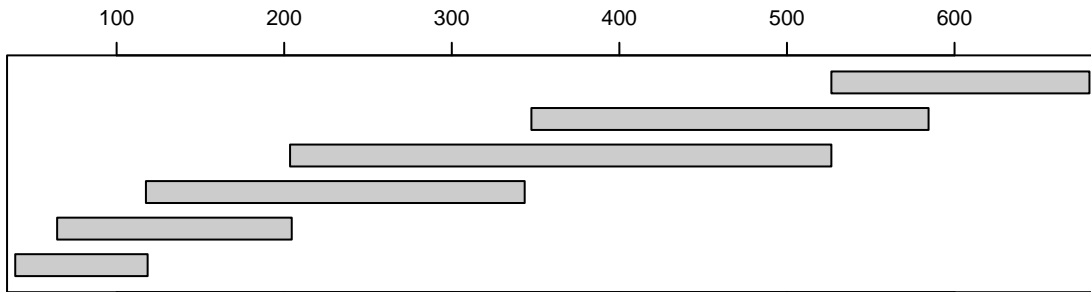


help("contour")

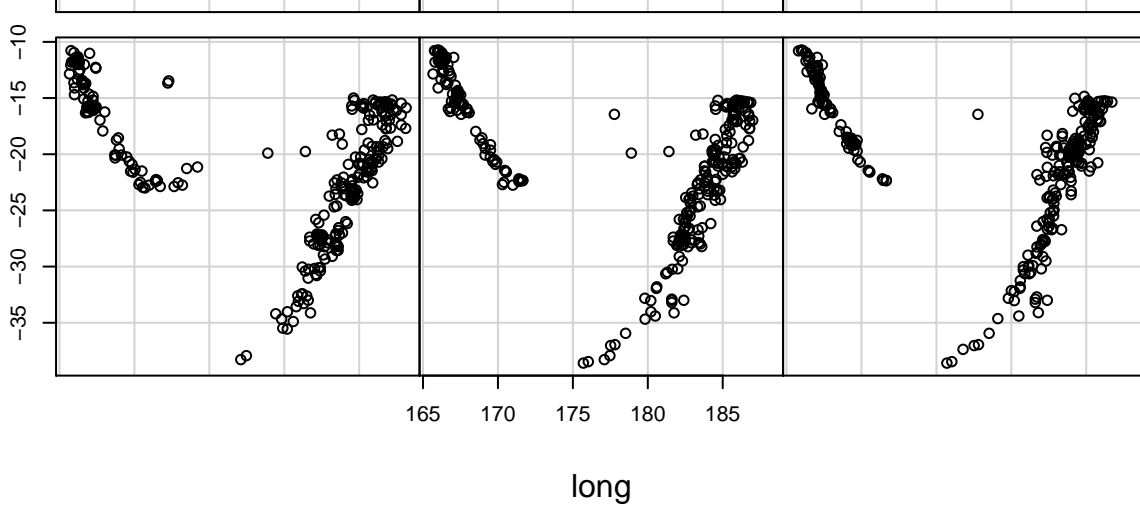
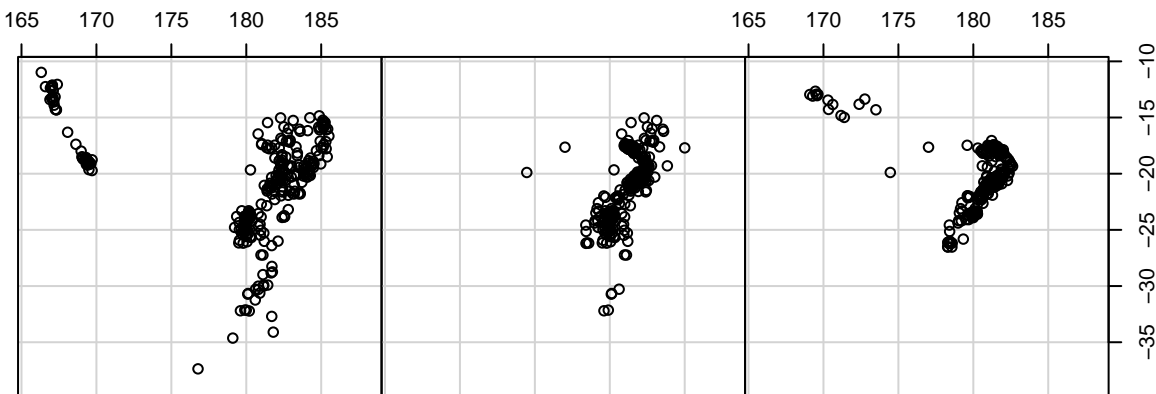




Given : depth

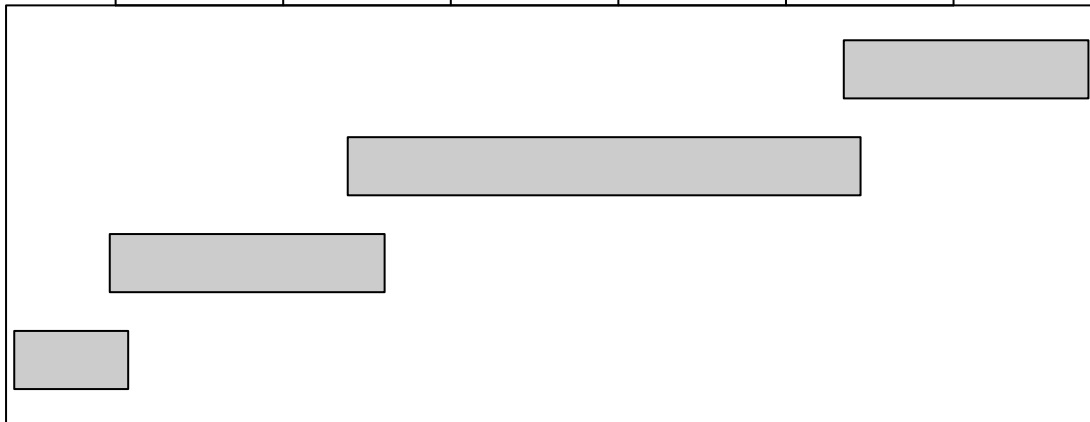


[help\("coplot"\)](#)

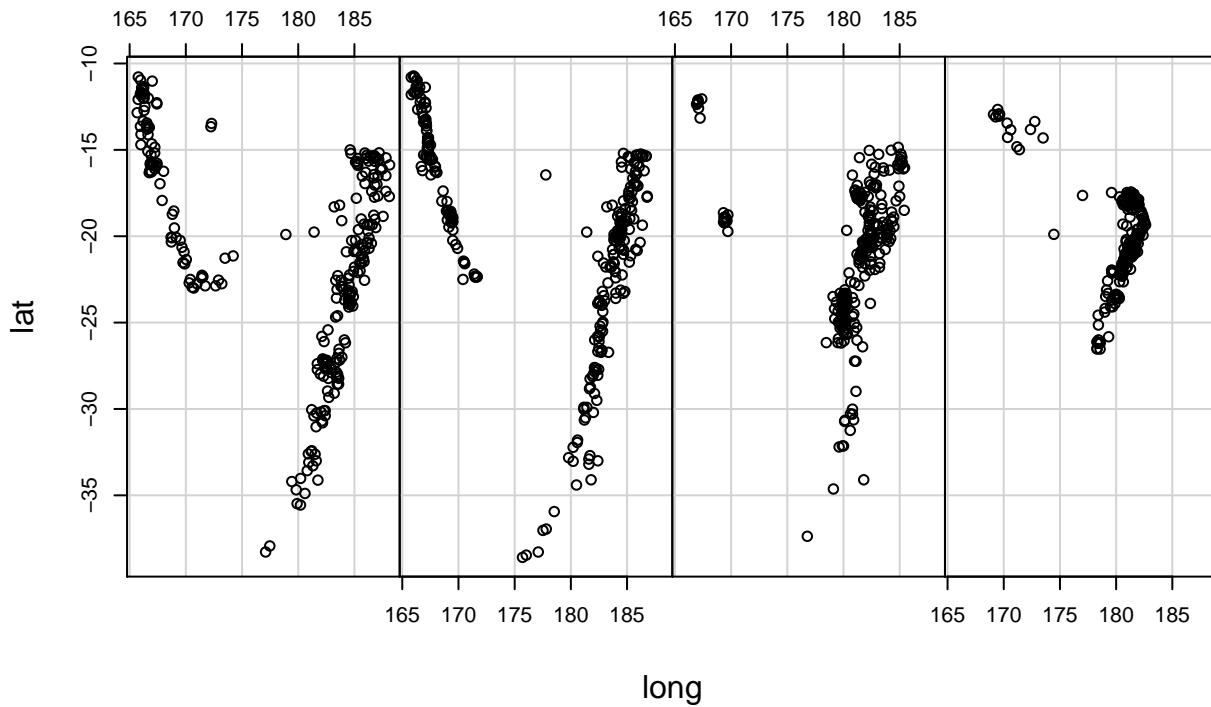


Given : depth

100 200 300 400 500 600

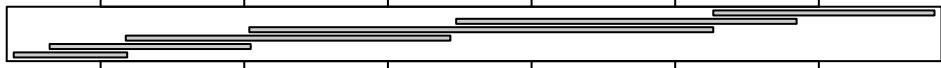


help("coplot")



Given : depth

100 200 300 400 500 600

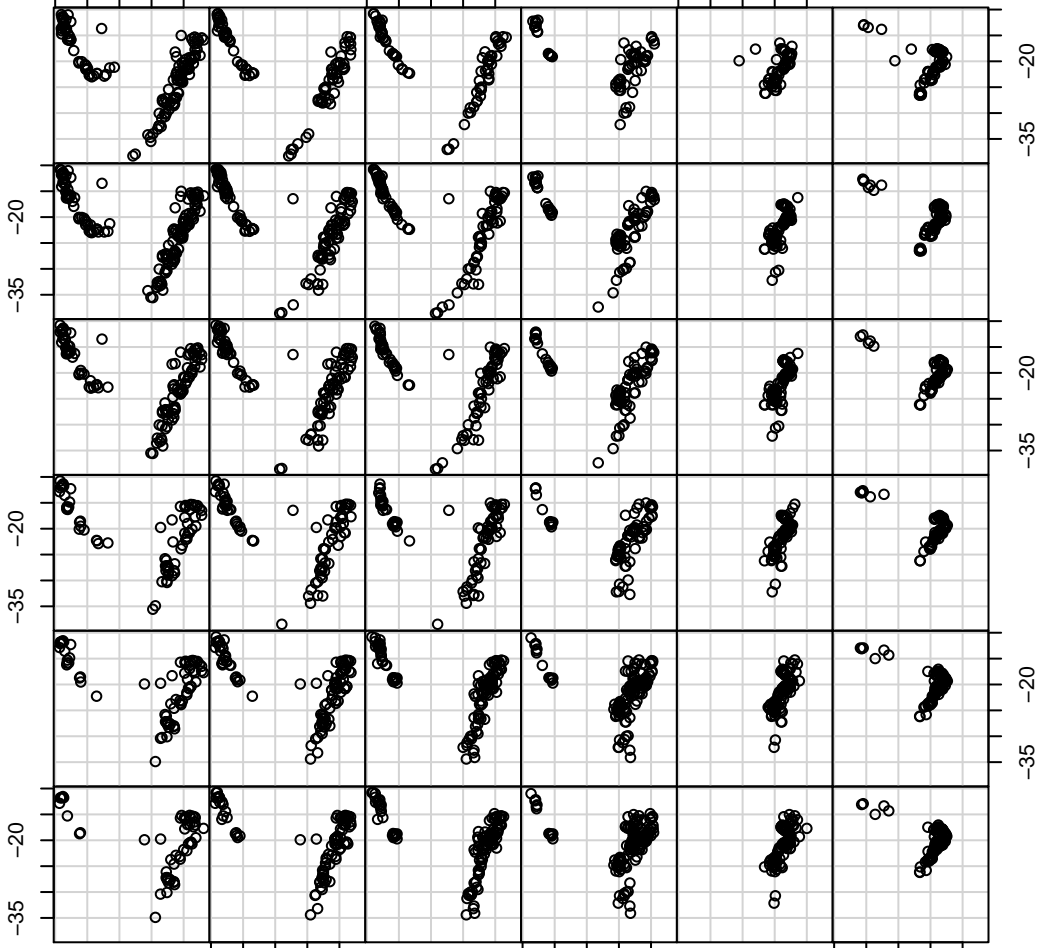


165 175 185

165 175 185

165 175 185

lat



165 175 185

165 175 185

165 175 185

long

Given : mag

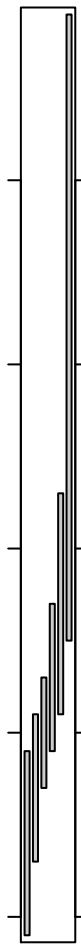
6.0

5.5

5.0

4.5

4.0



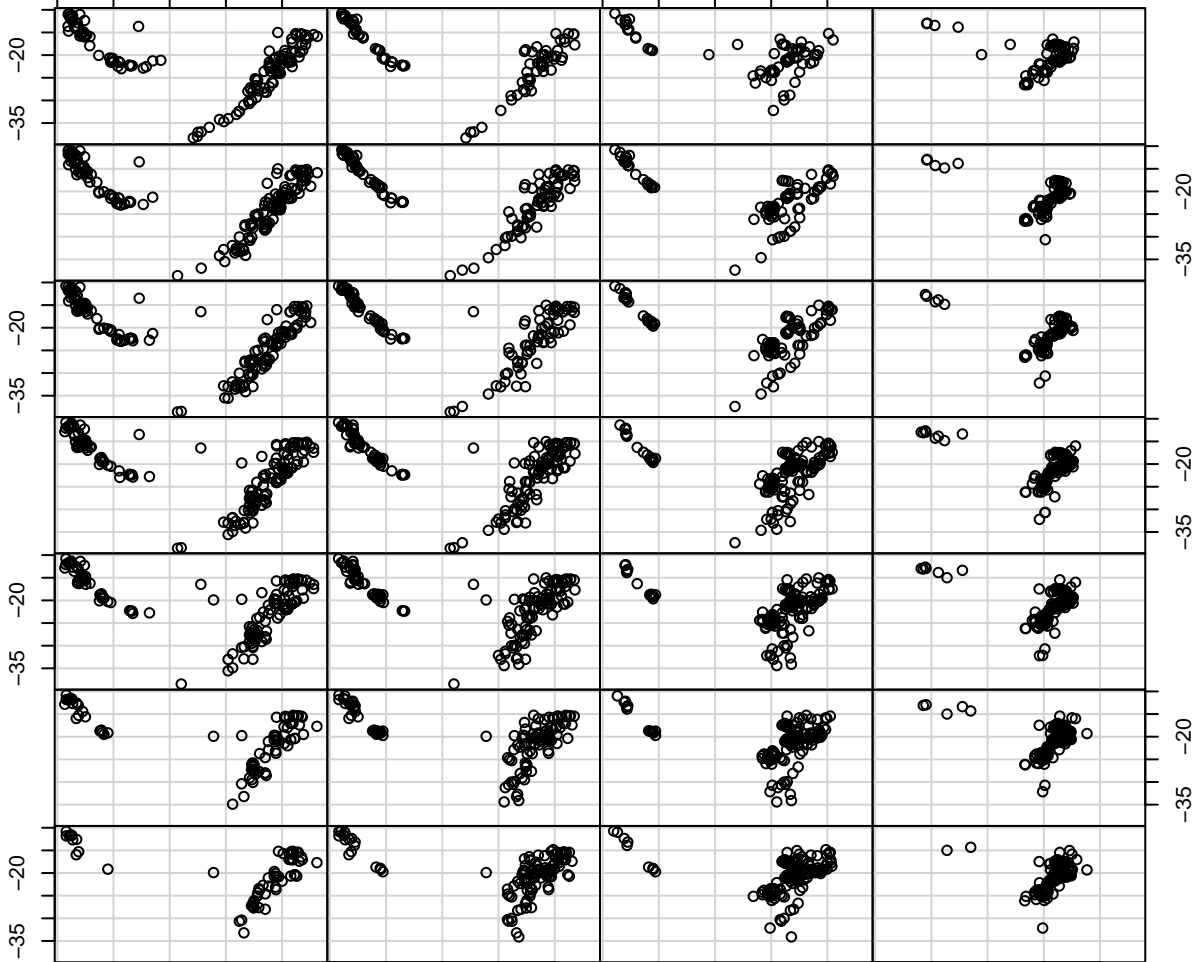
help("coplot")

Given : depth

100 200 300 400 500 600

165 170 175 180 185

165 170 175 180 185



help("coplot")

Given : mag

165 170 175 180 185

165 170 175 180 185

long

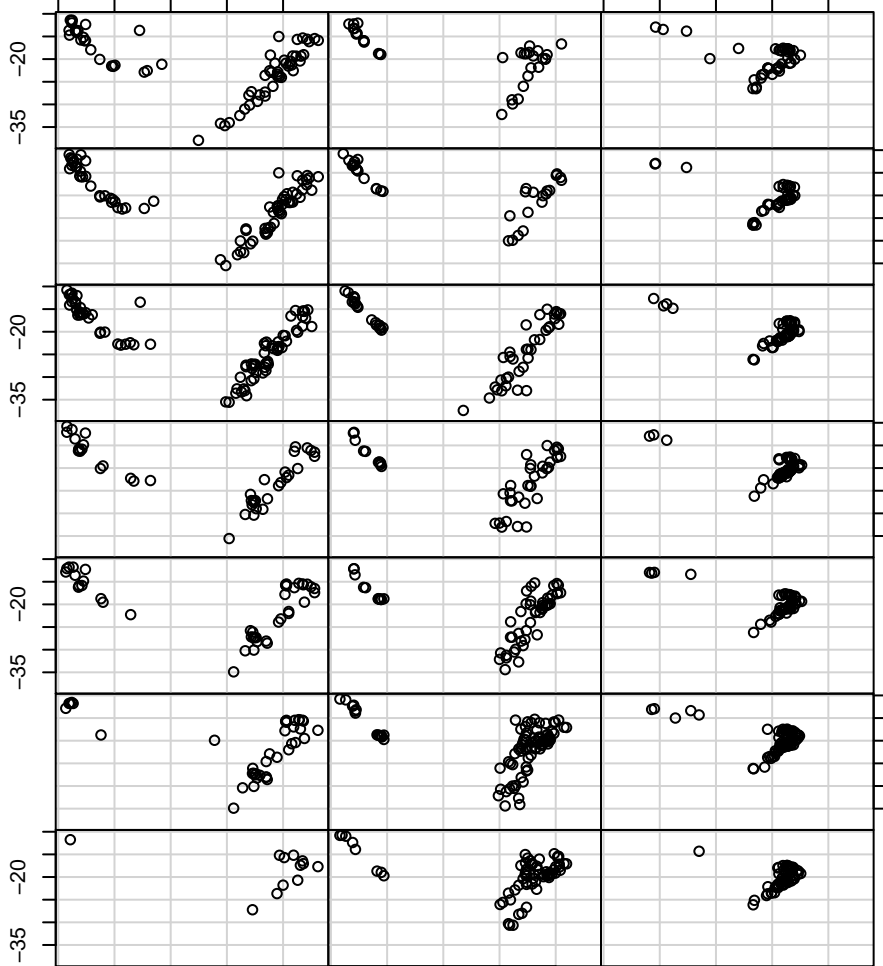
Given : depth

100 200 300 400 500 600



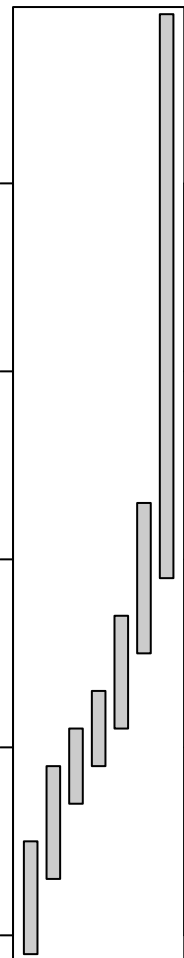
165 170 175 180 185

165 170 175 180 185



165 170 175 180 185

long

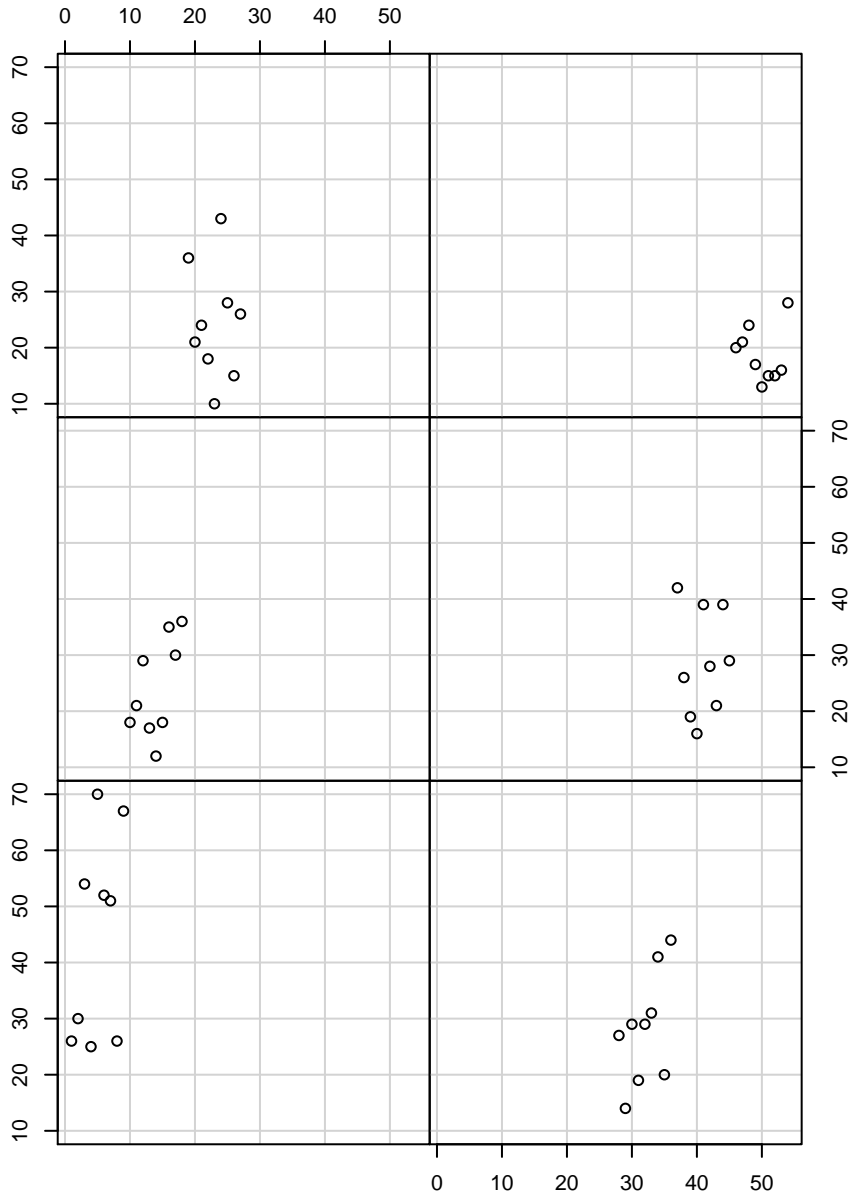


Given : mag

help("coplot")

Given : wool

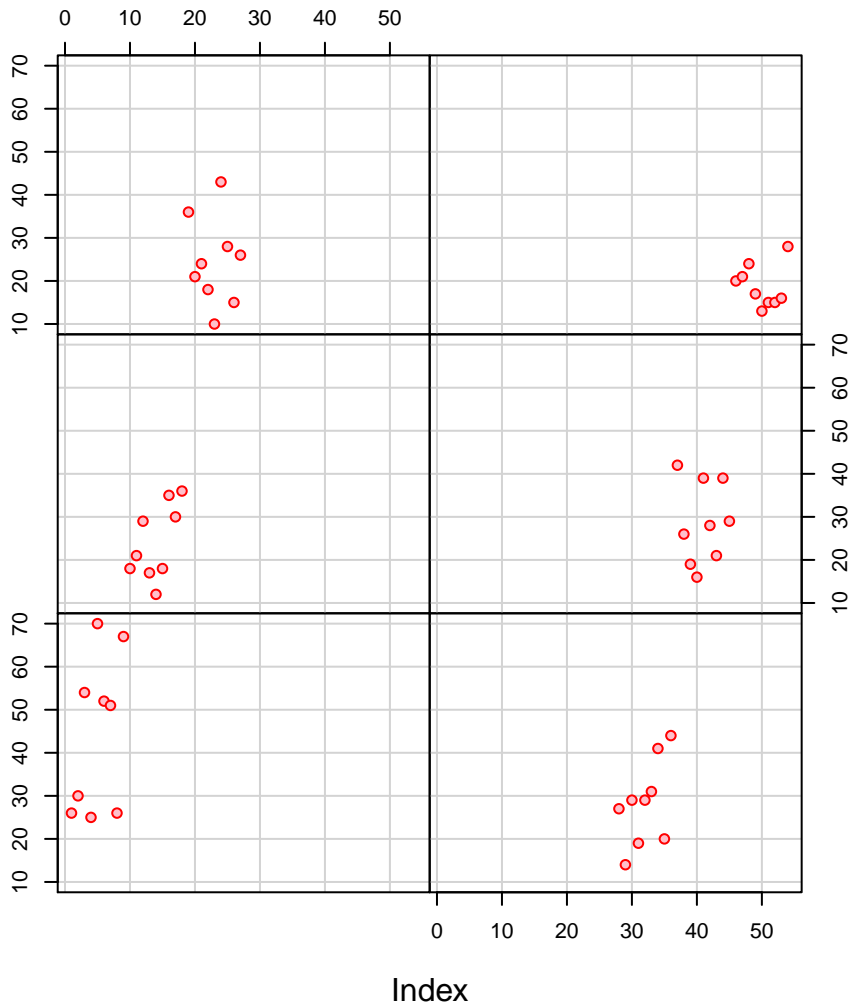
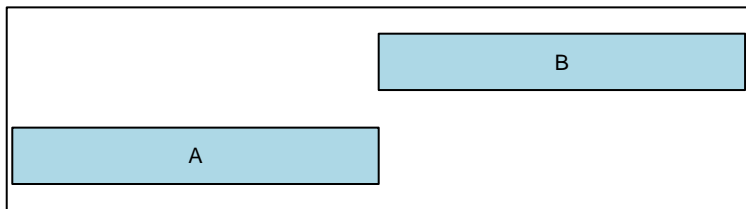
breaks



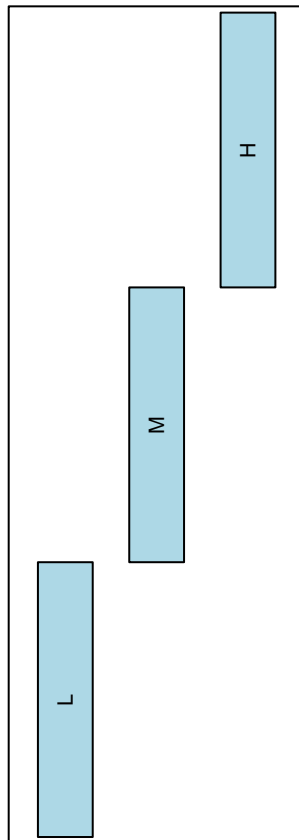
Given : tension

help("coplot")

Given : wool



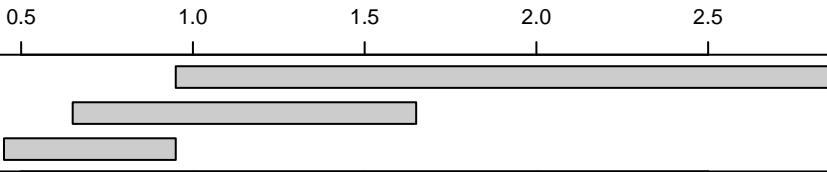
Given : tension



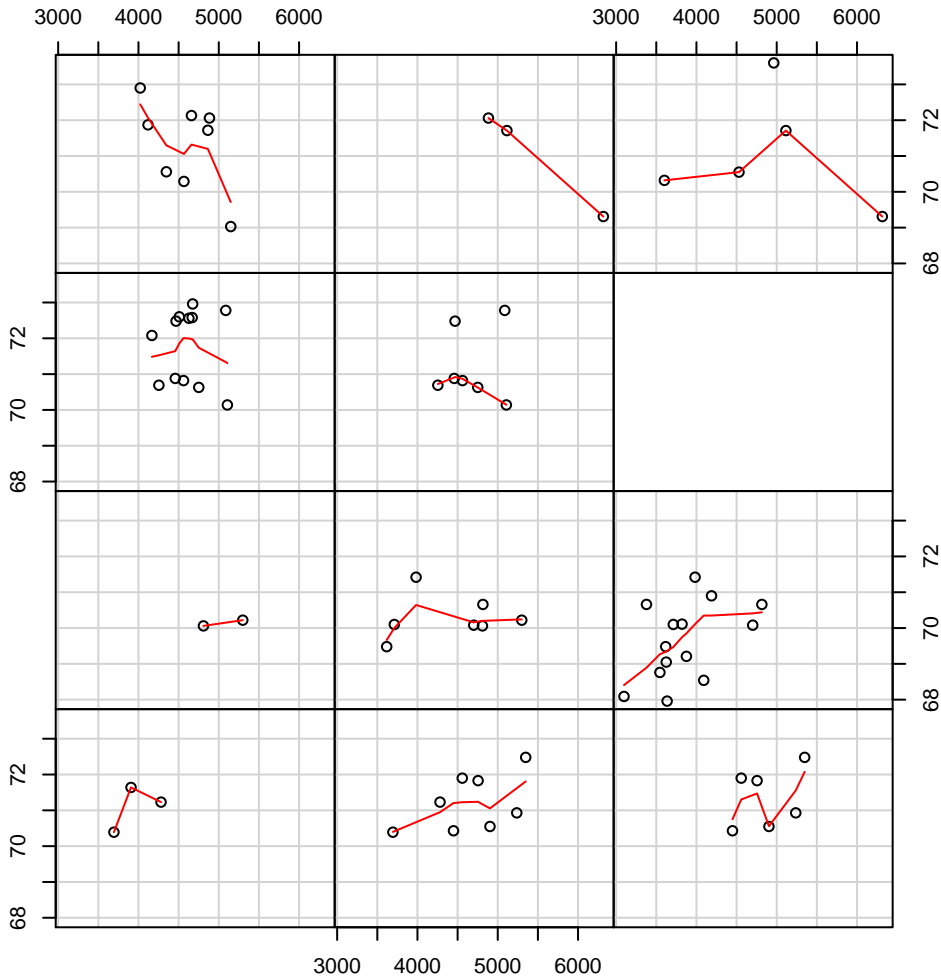
help("coplot")



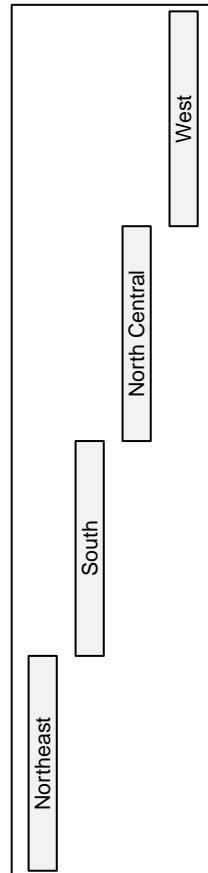
Given : Illiteracy



Life.Exp

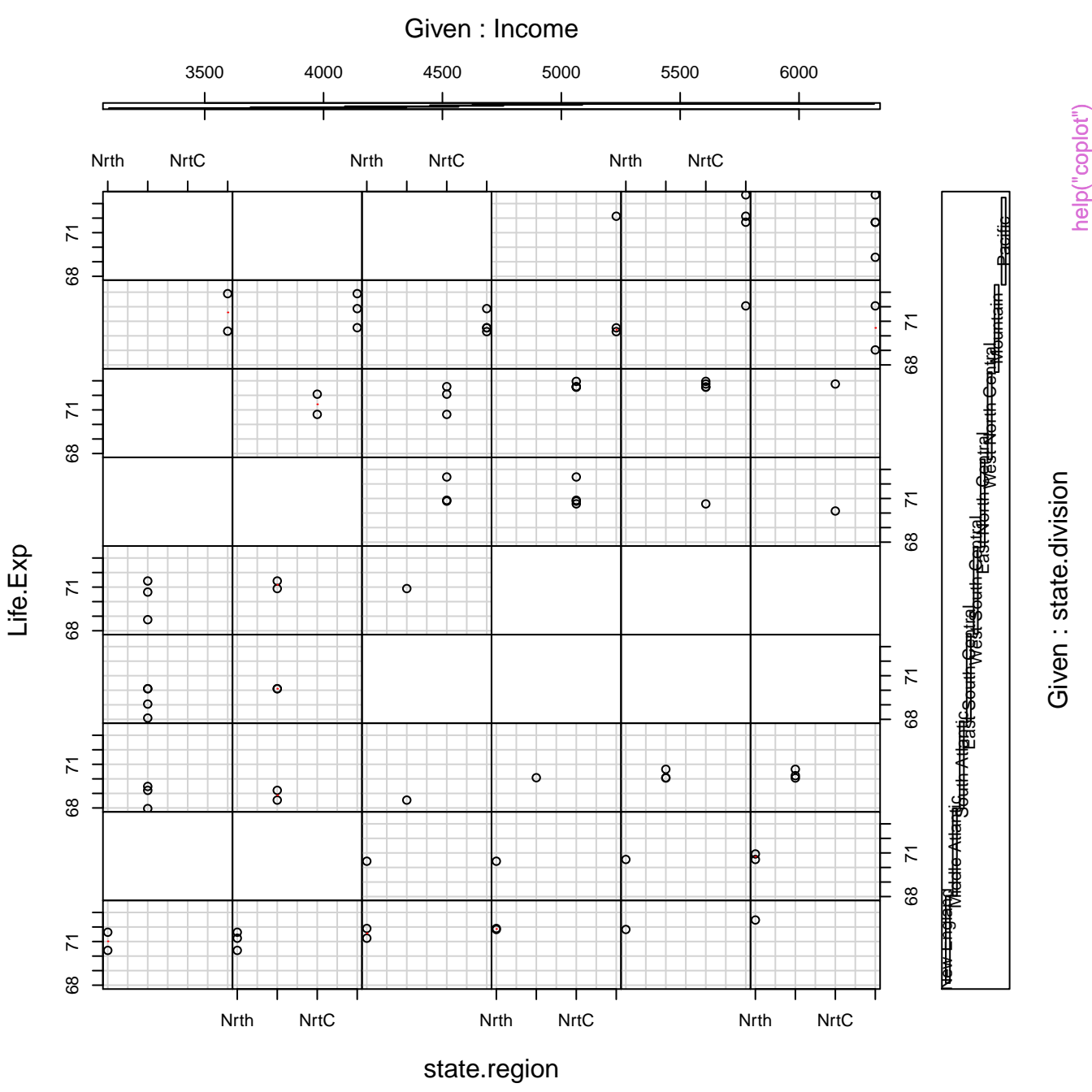


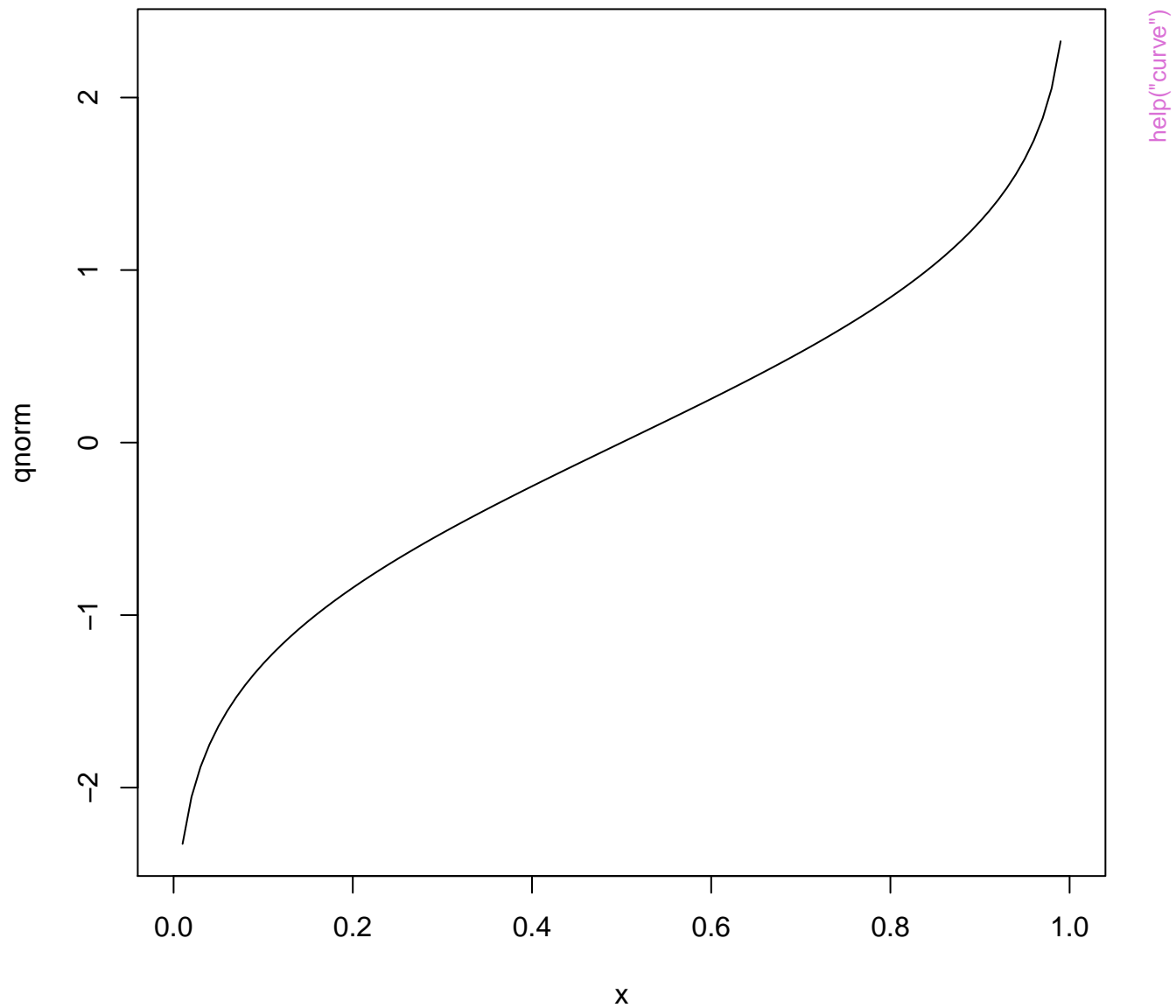
Income



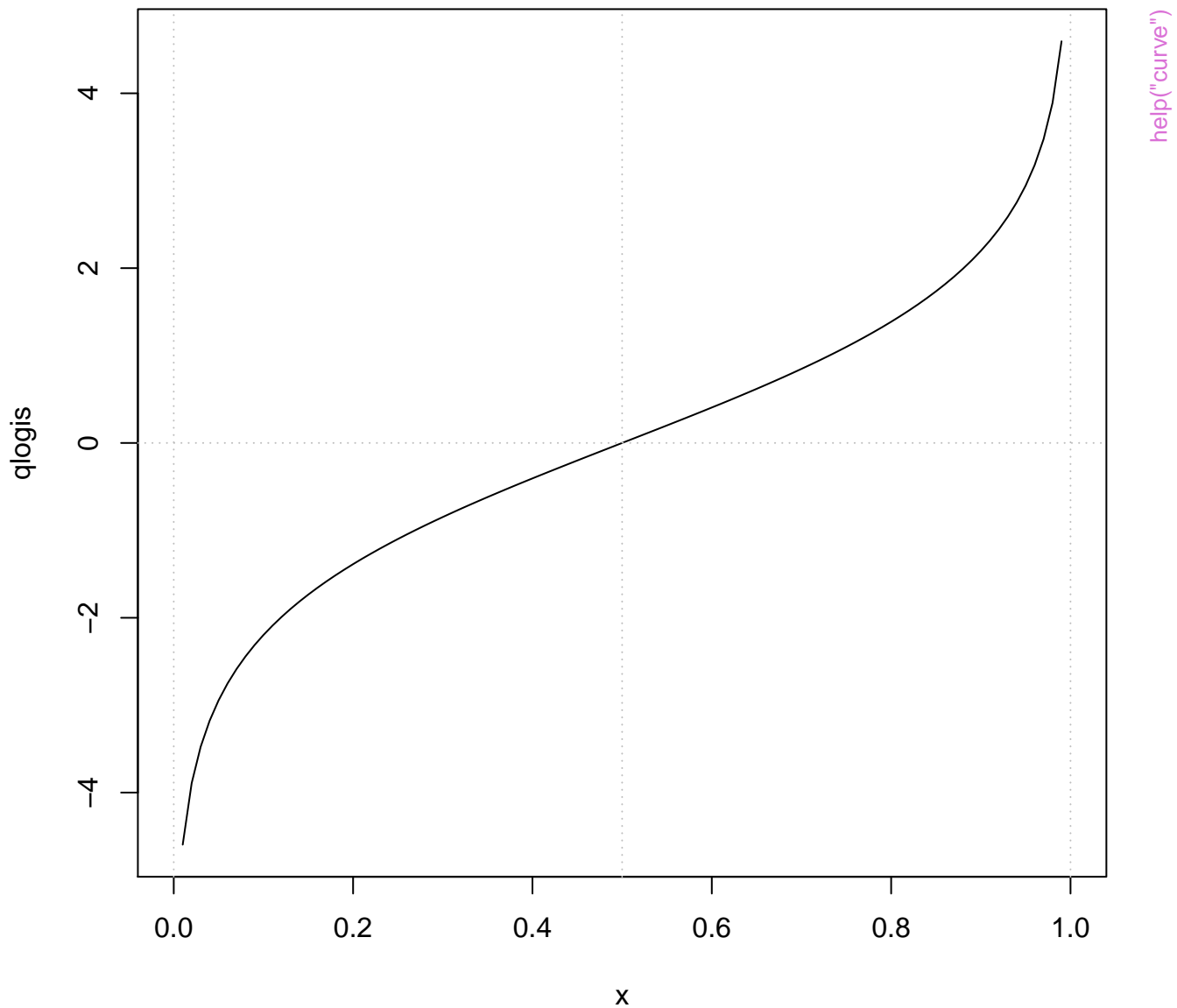
Given : state.region

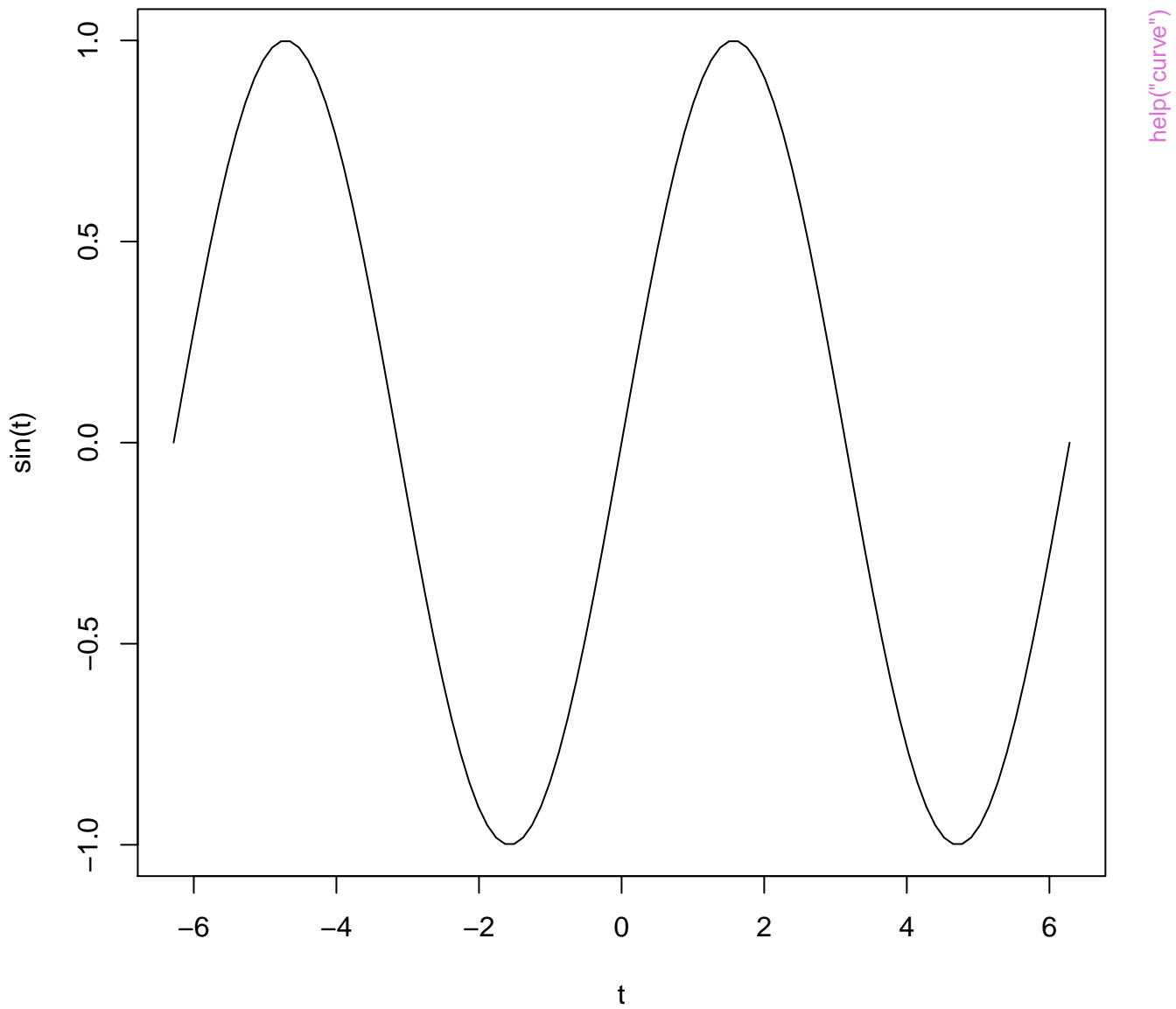
help("coplot")



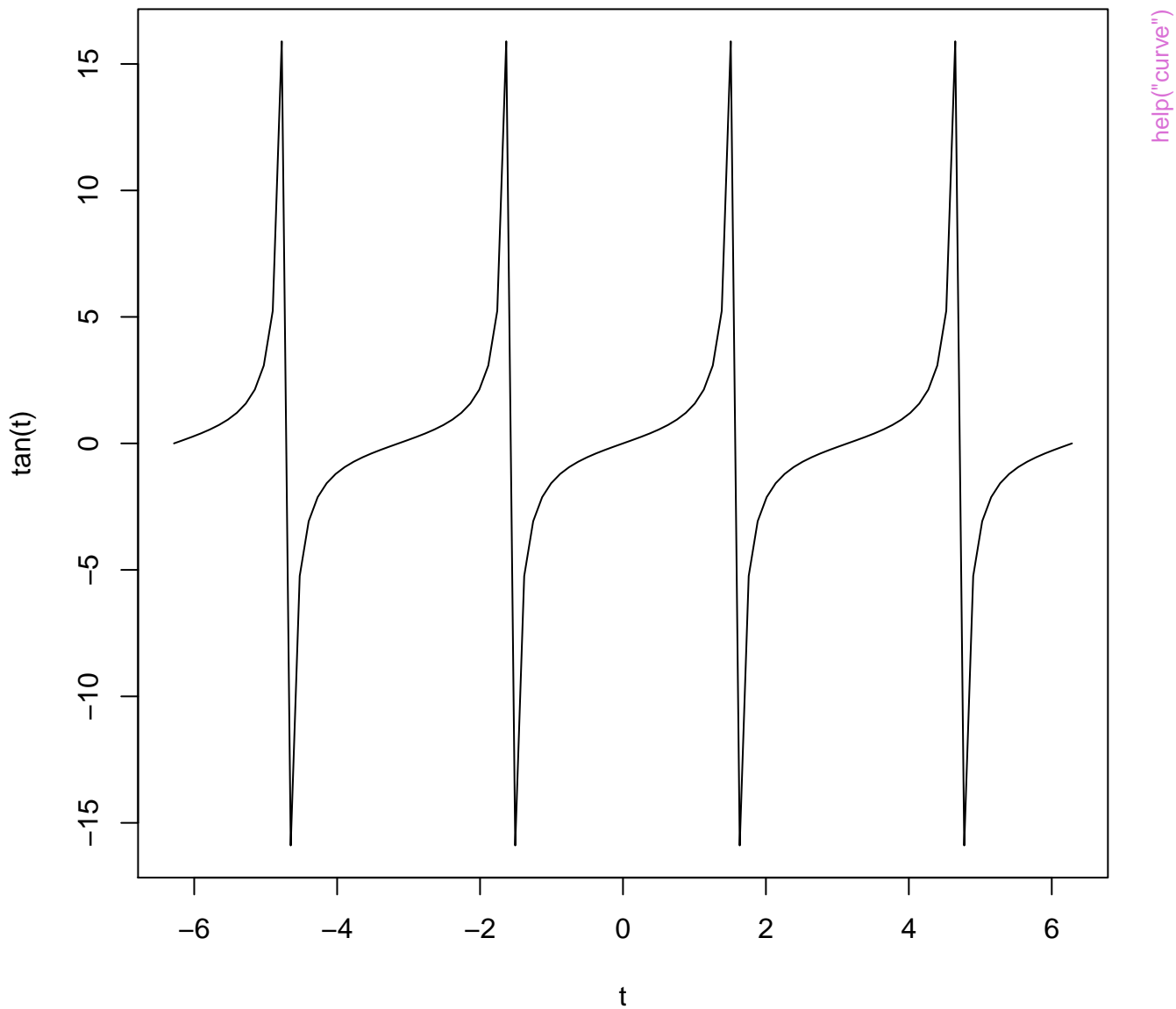


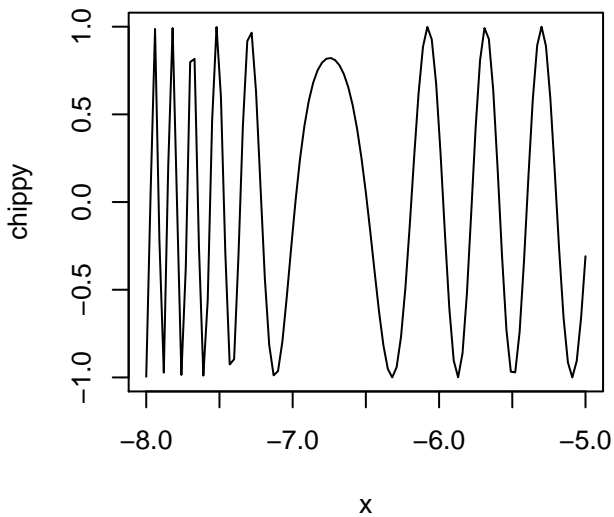
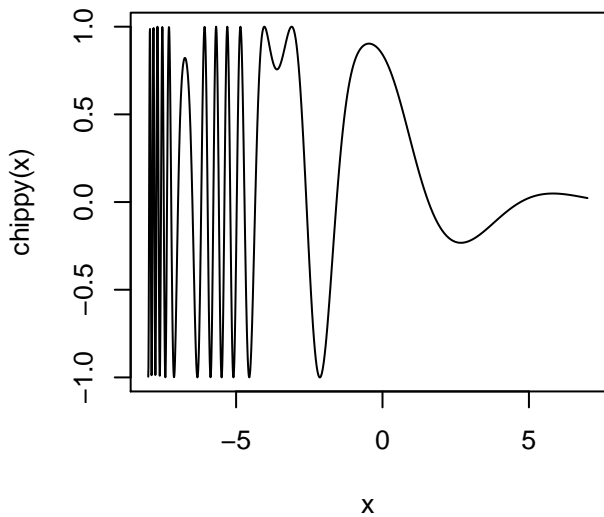
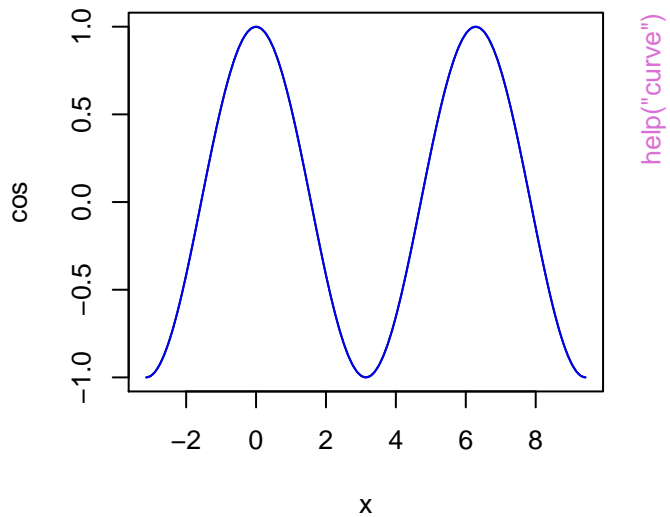
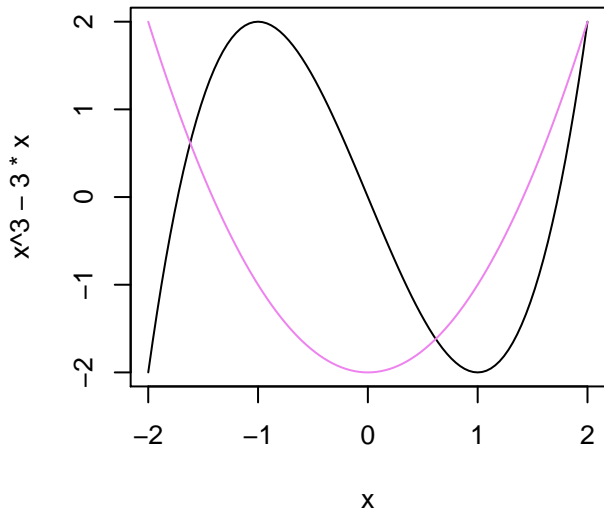
## The Inverse Logit : `qlogis()`

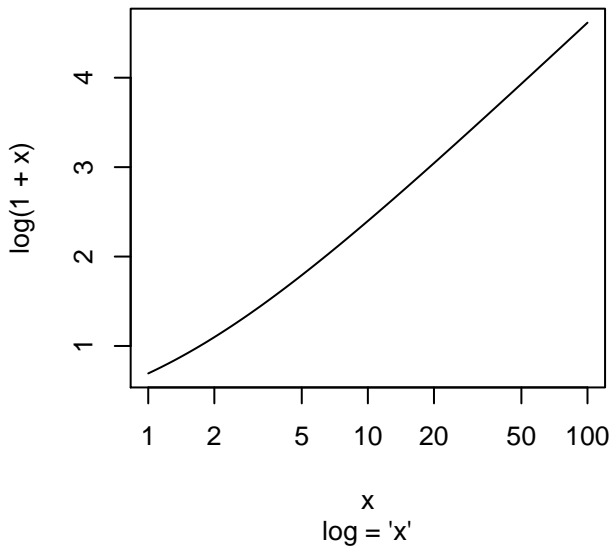
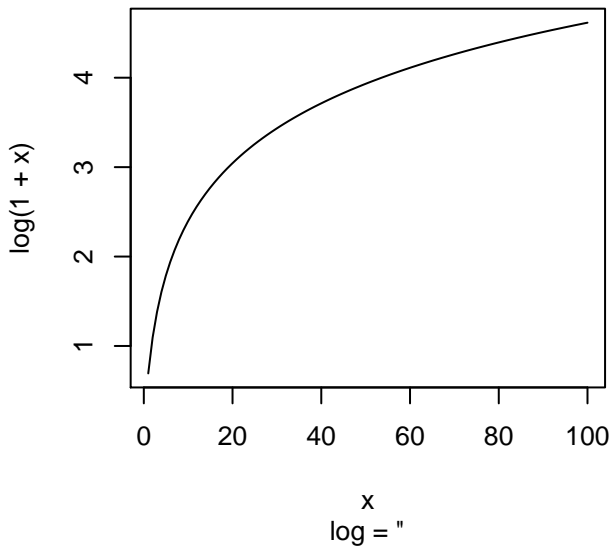




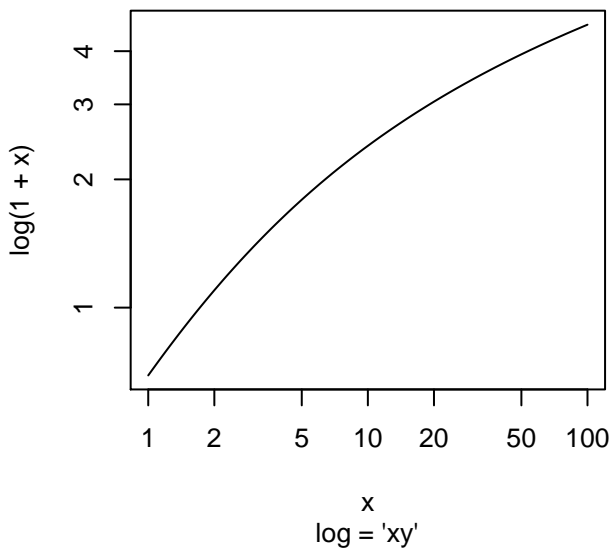
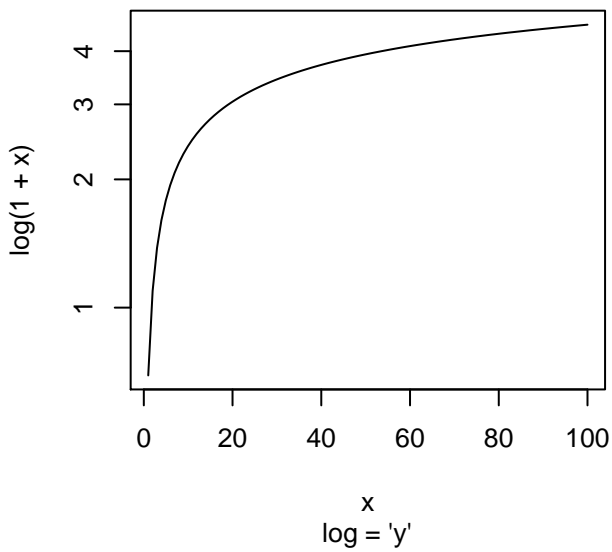
**curve(tan) --> same x-scale as previous plot**







help("curve")





# Death Rates in Virginia – 1940

## Rural Male

70–74

65–69

60–64

55–59

50–54

## Rural Female

70–74

65–69

60–64

55–59

50–54

## Urban Male

70–74

65–69

60–64

55–59

50–54

## Urban Female

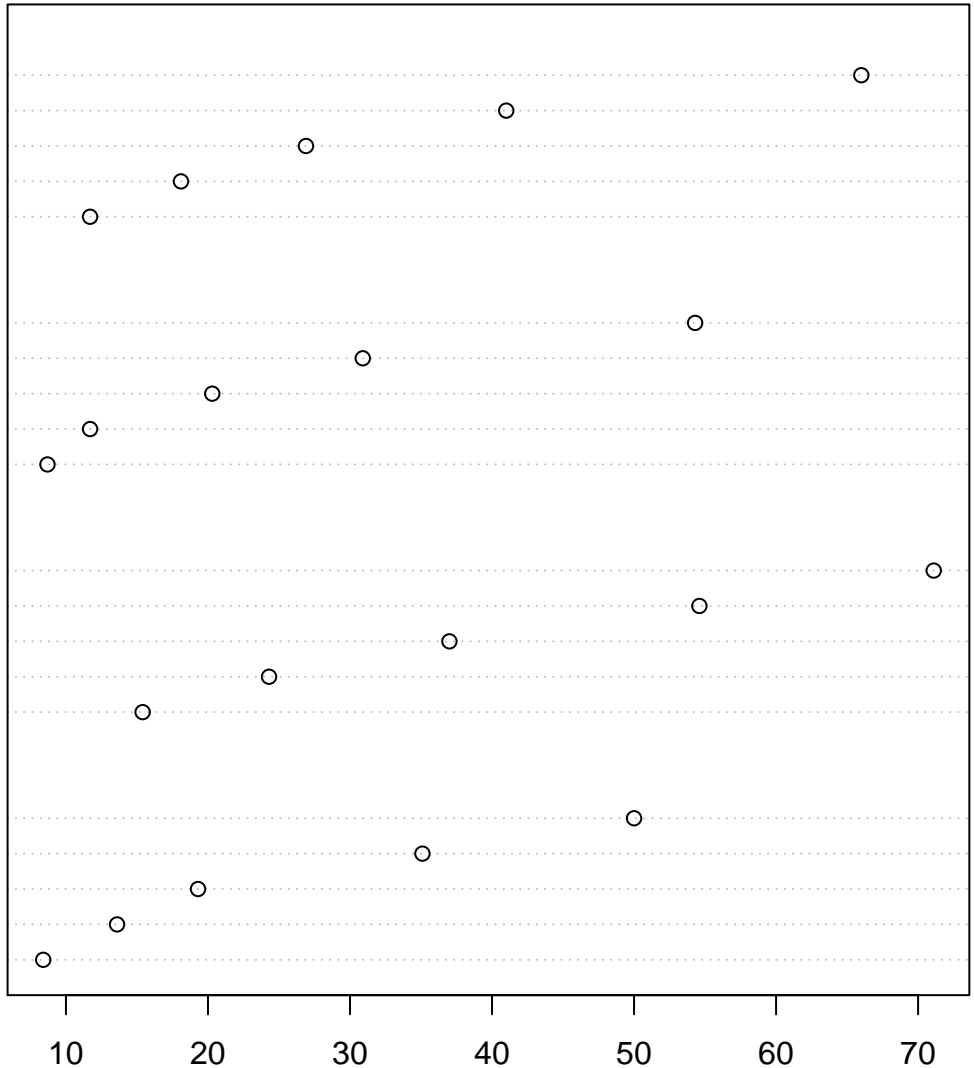
70–74

65–69

60–64

55–59

50–54



## Death Rates in Virginia – 1940

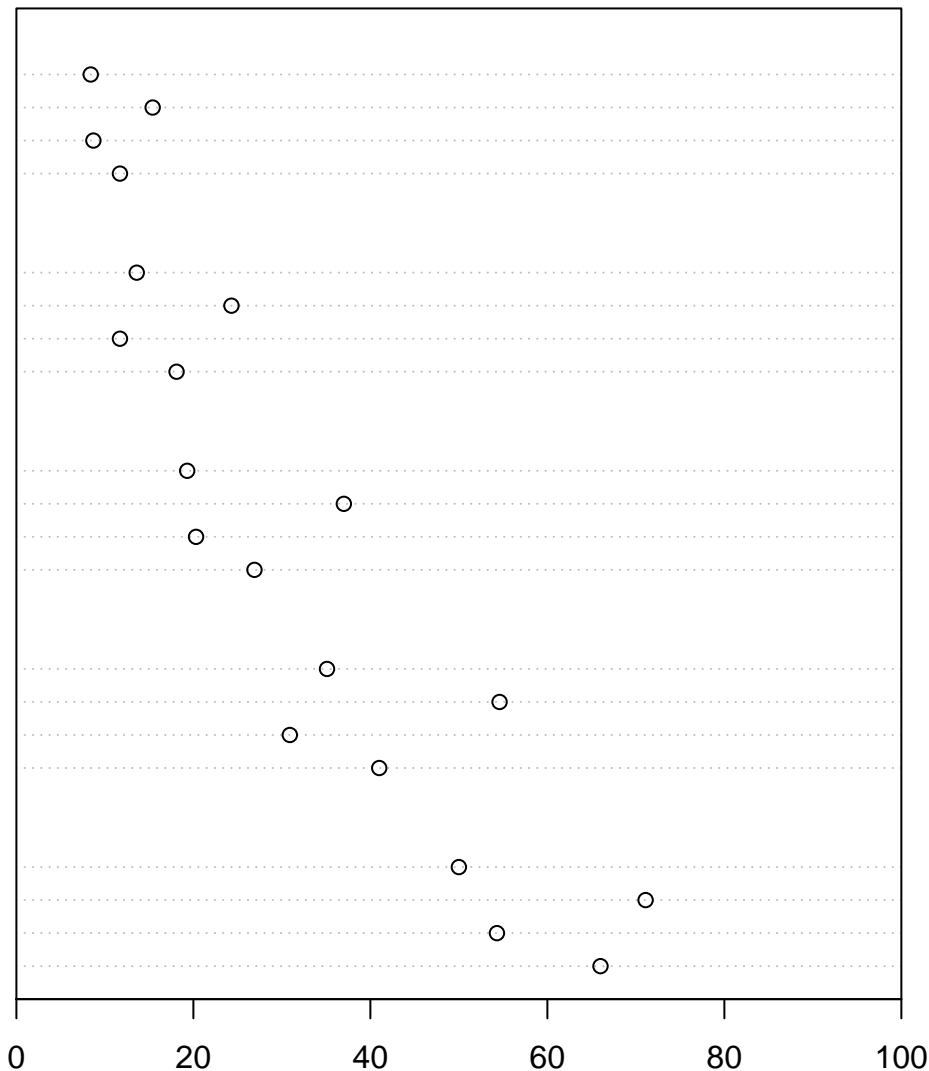
50–54  
Urban Female  
Urban Male  
Rural Female  
Rural Male

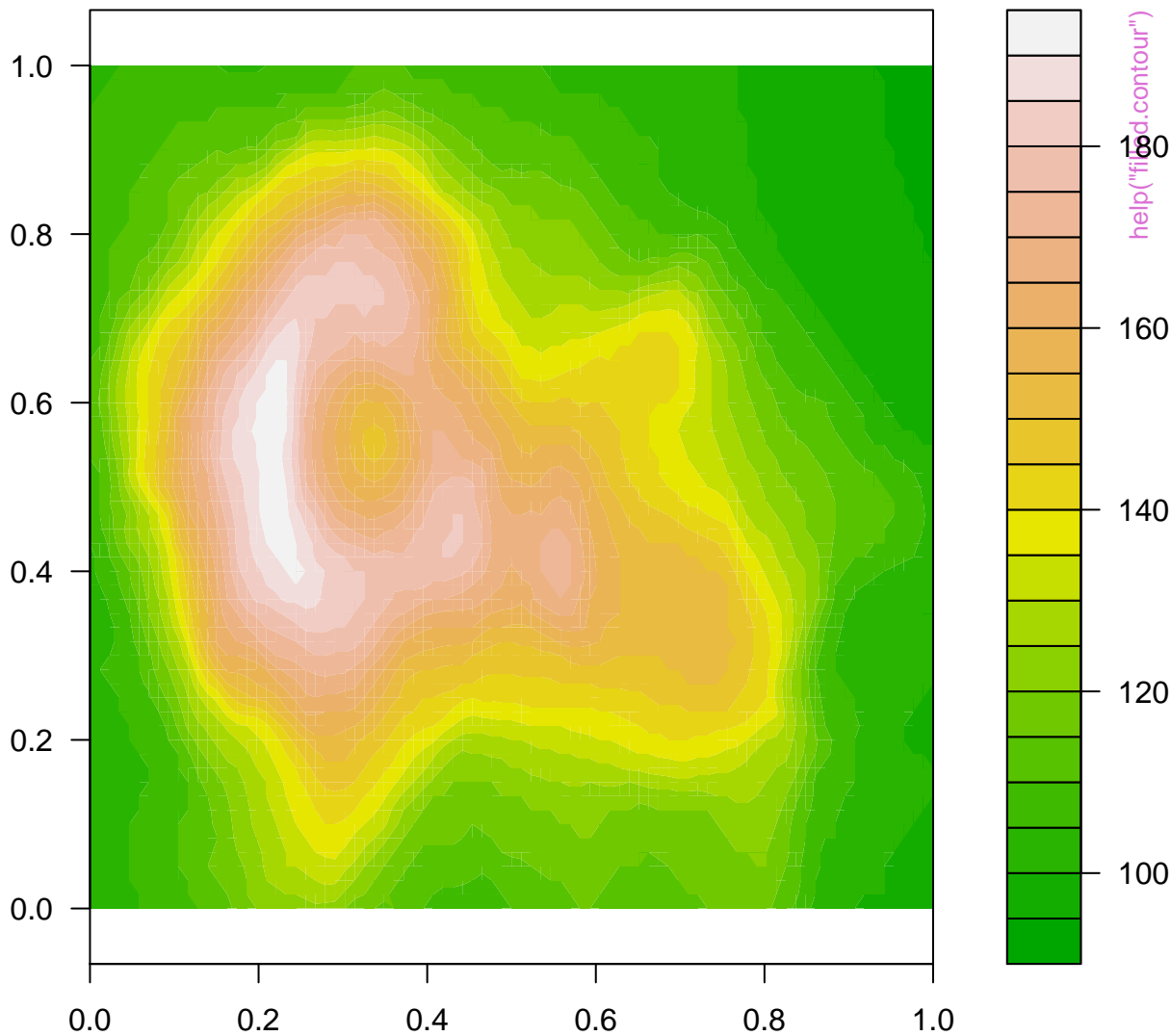
55–59  
Urban Female  
Urban Male  
Rural Female  
Rural Male

60–64  
Urban Female  
Urban Male  
Rural Female  
Rural Male

65–69  
Urban Female  
Urban Male  
Rural Female  
Rural Male

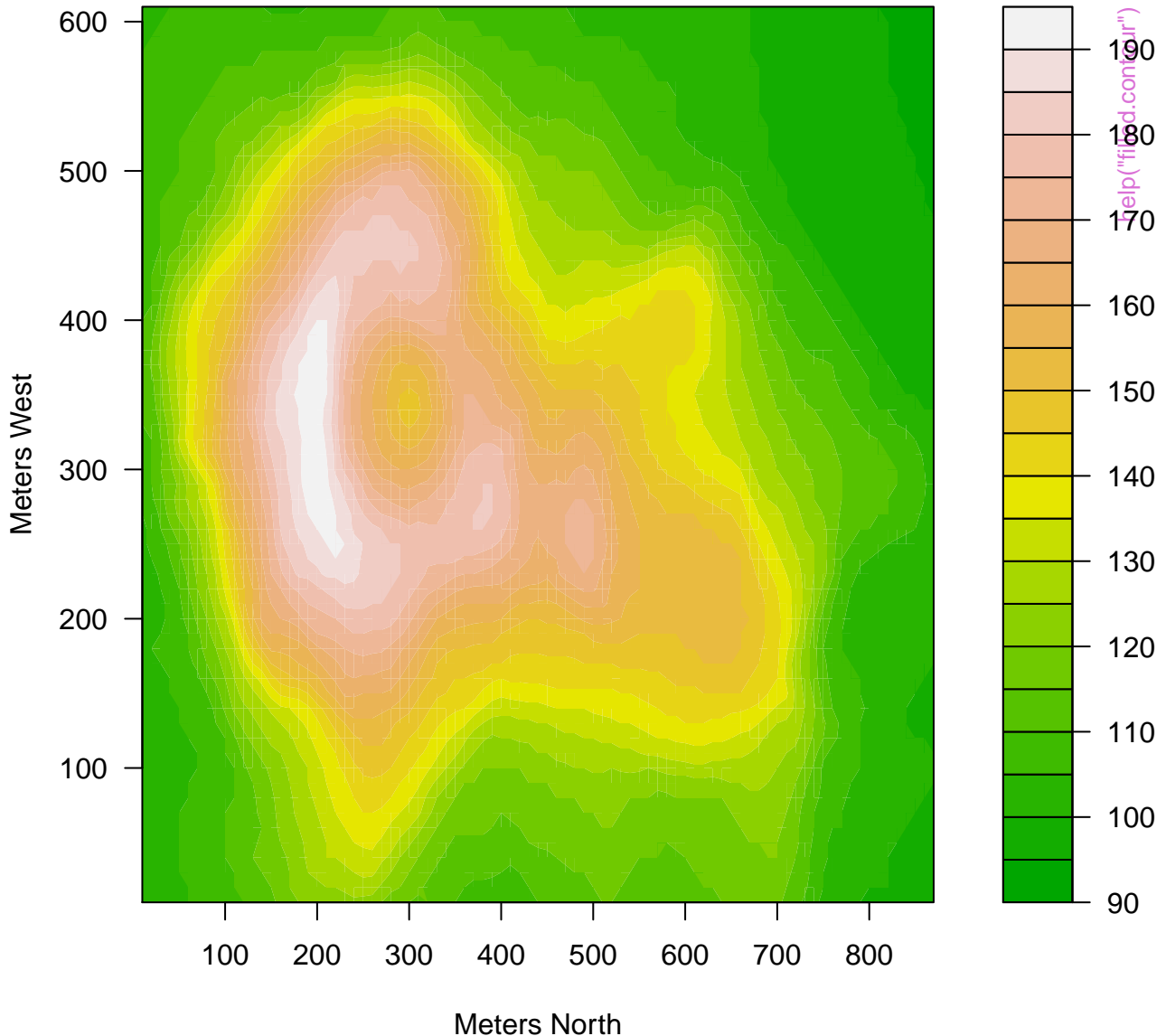
70–74  
Urban Female  
Urban Male  
Rural Female  
Rural Male

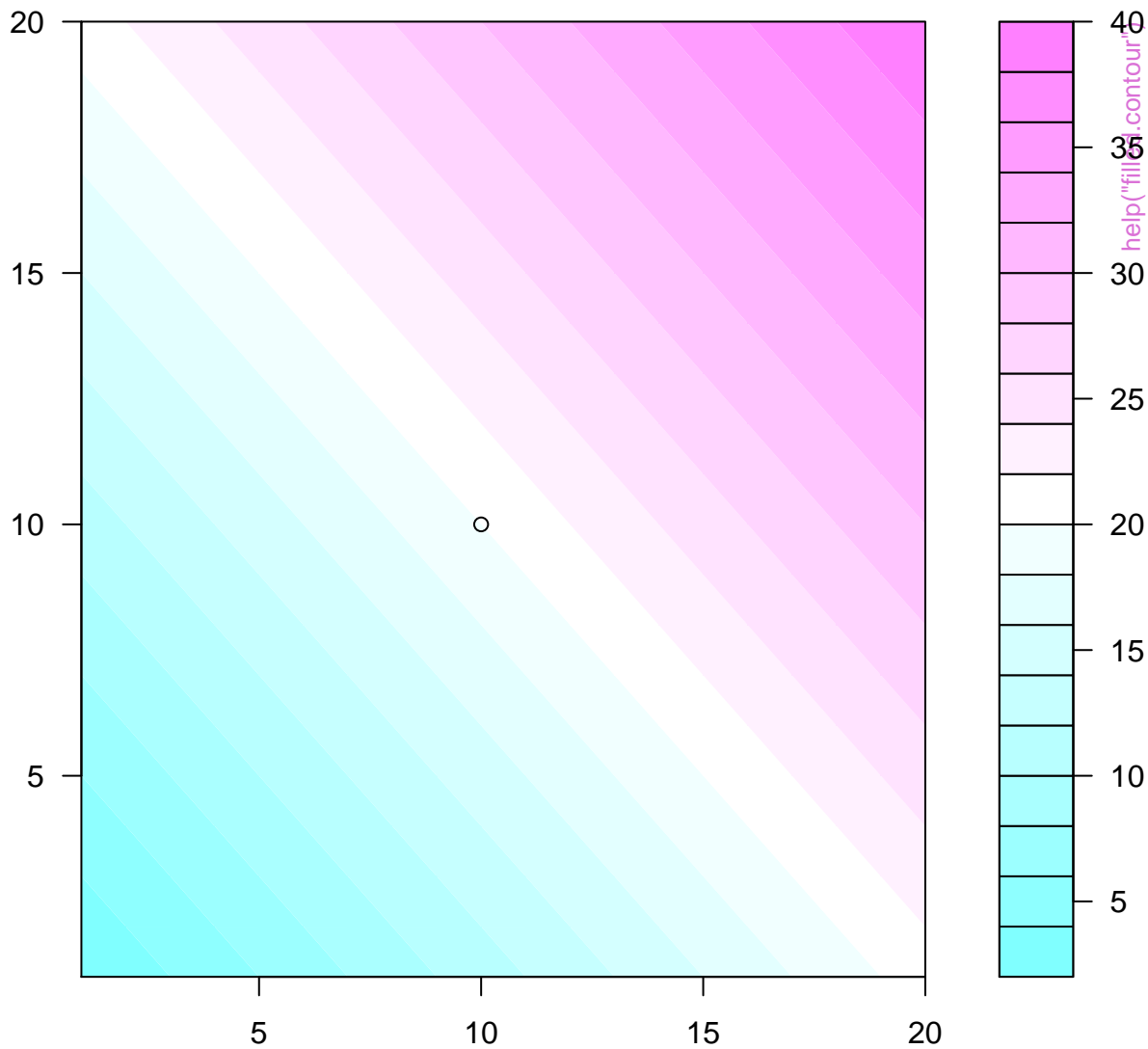


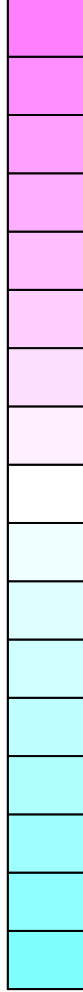
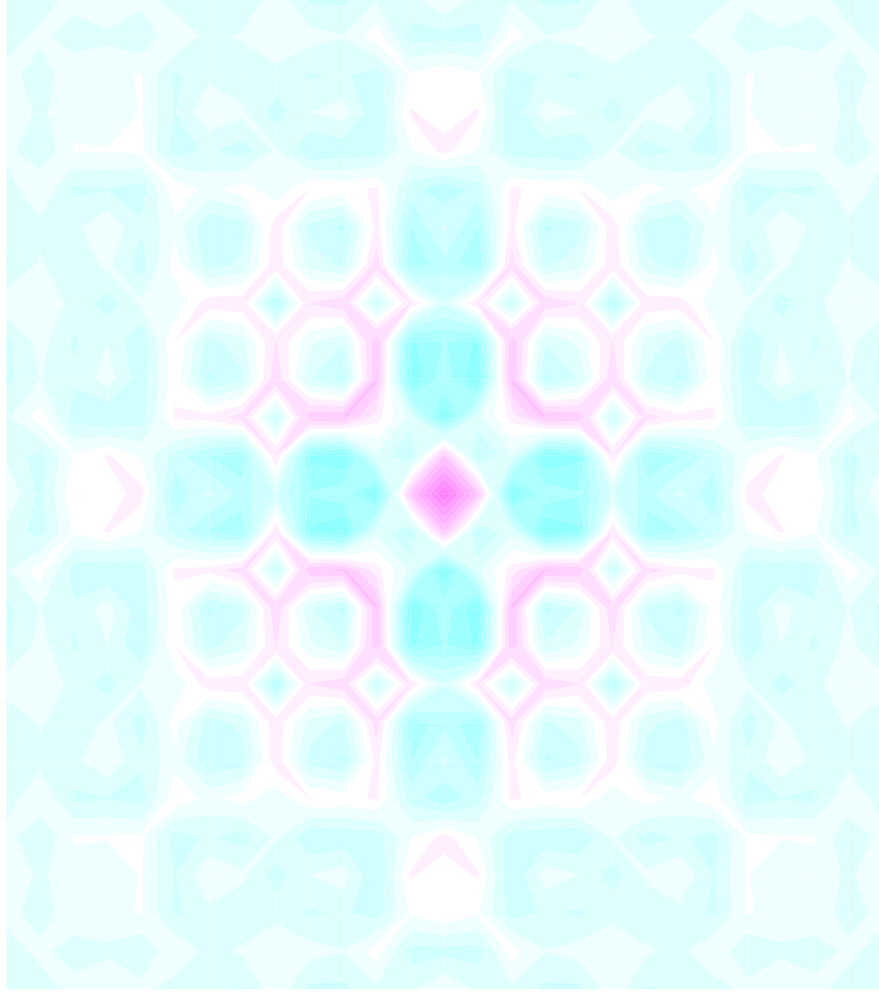


# The Topography of Maunga Whau

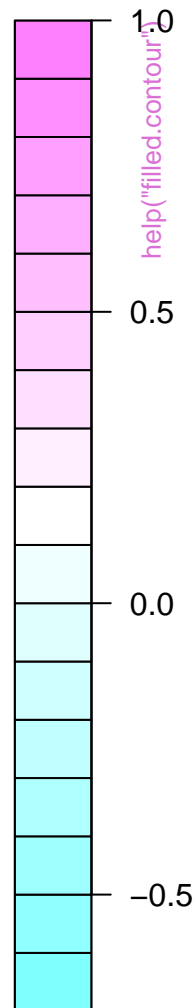
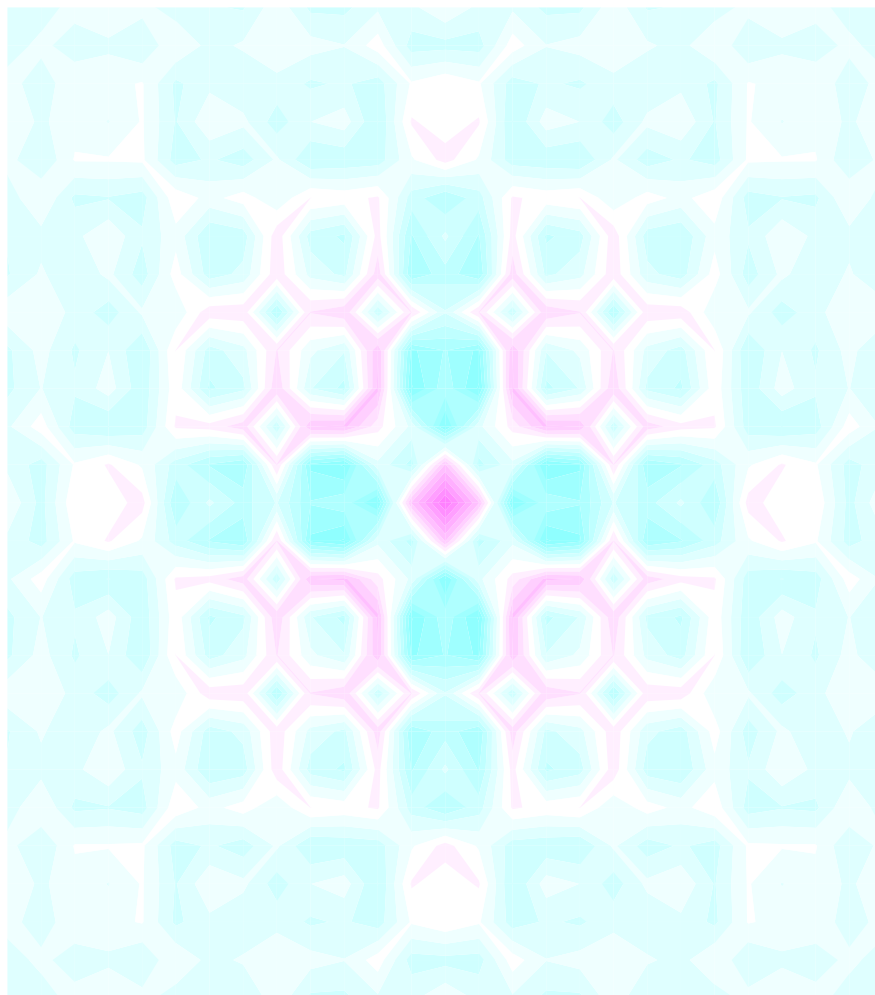
Height  
(meters)







help("filled.contour")



Sex: Male

1198

1493

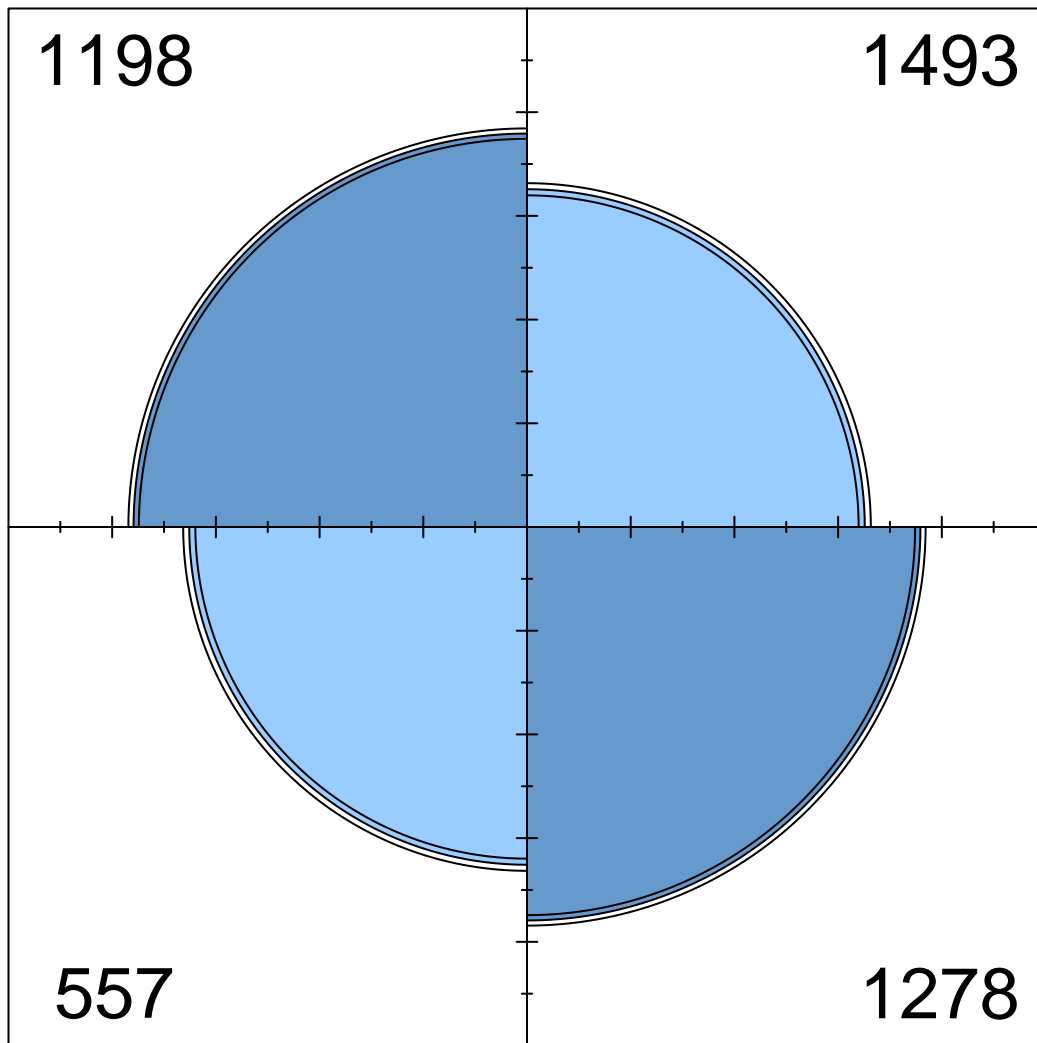
Admit?: Yes

Admit?: No

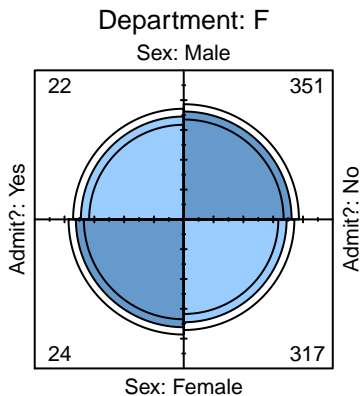
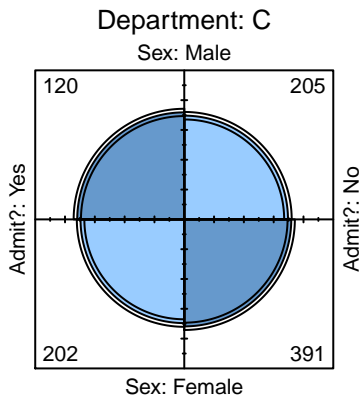
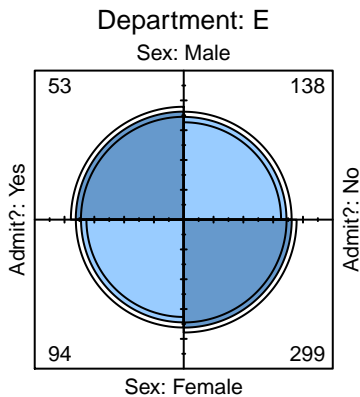
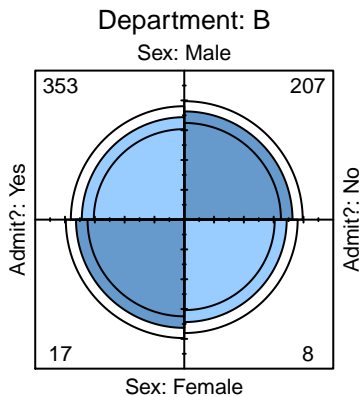
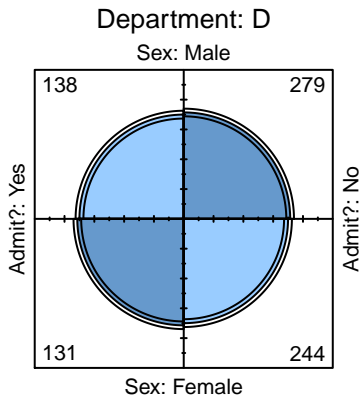
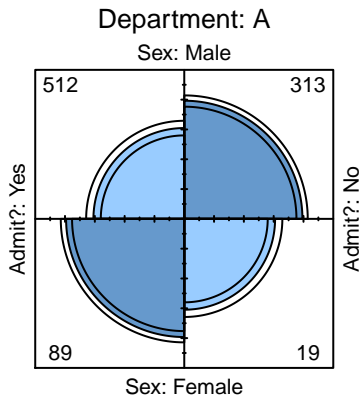
557

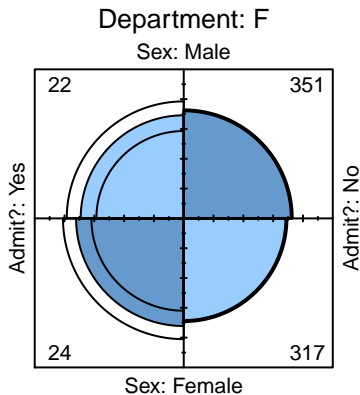
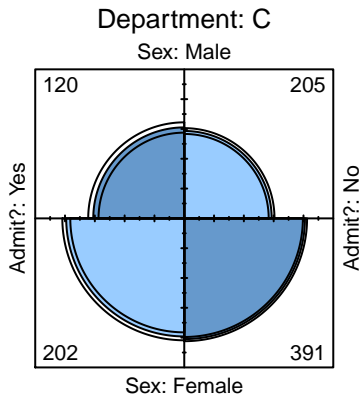
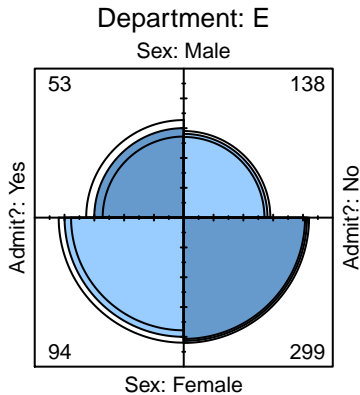
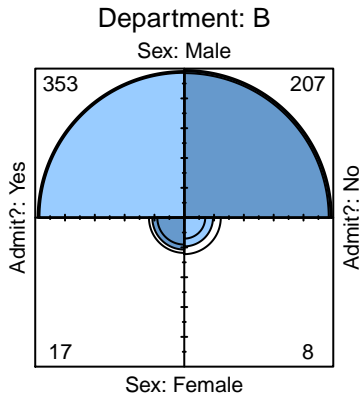
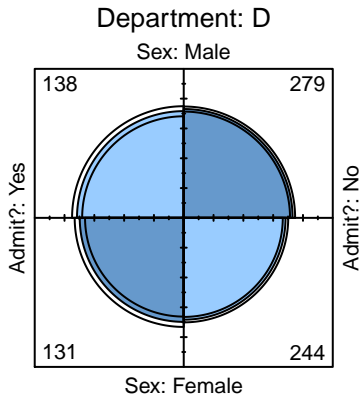
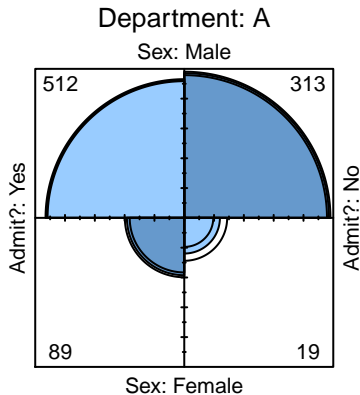
1278

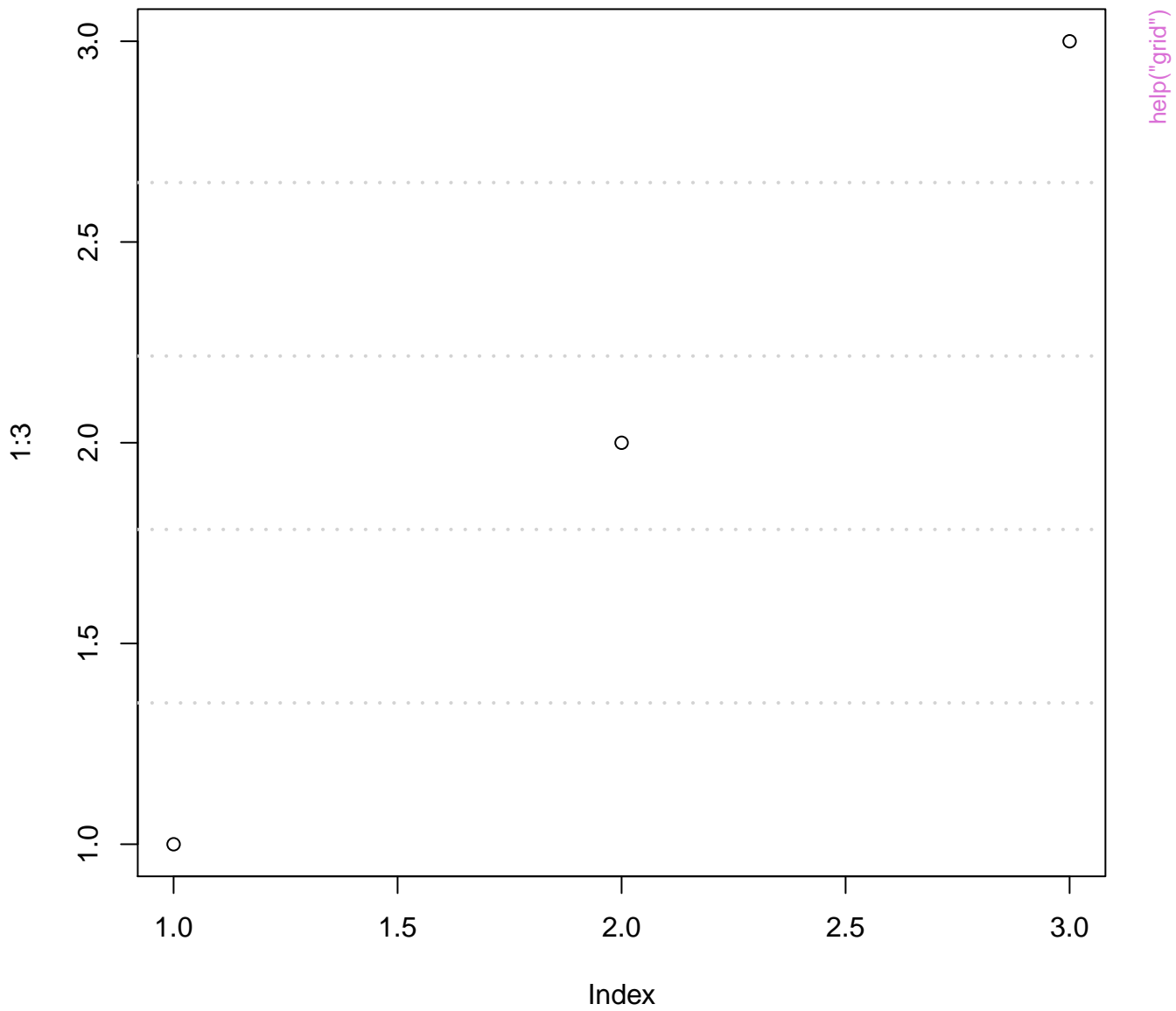
Sex: Female



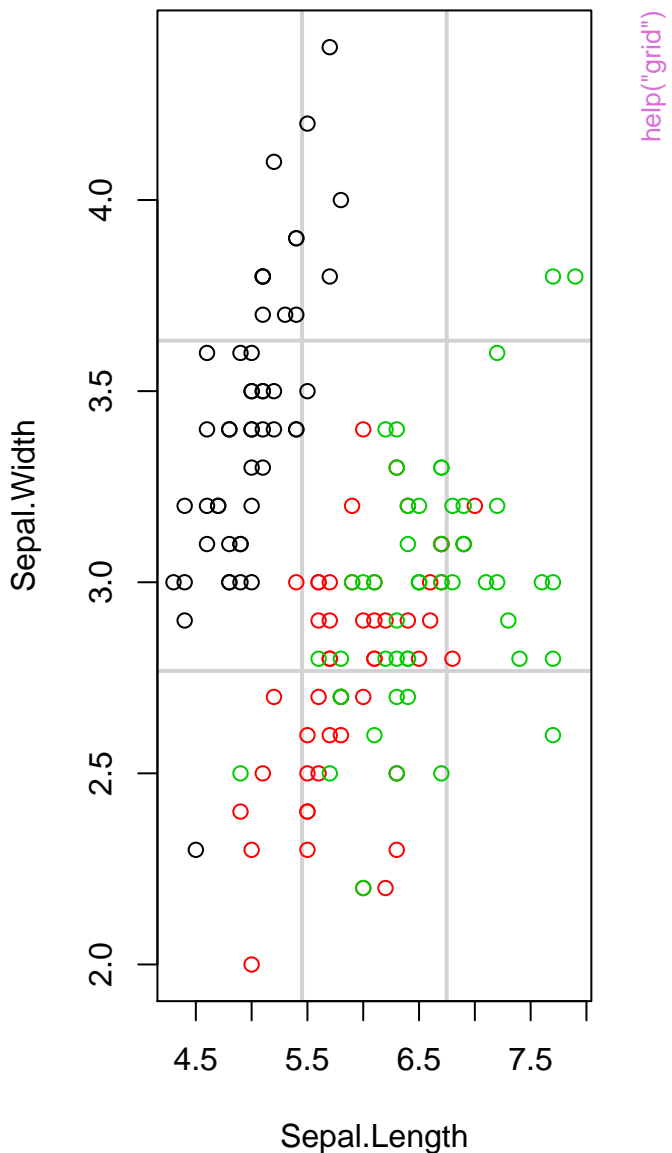
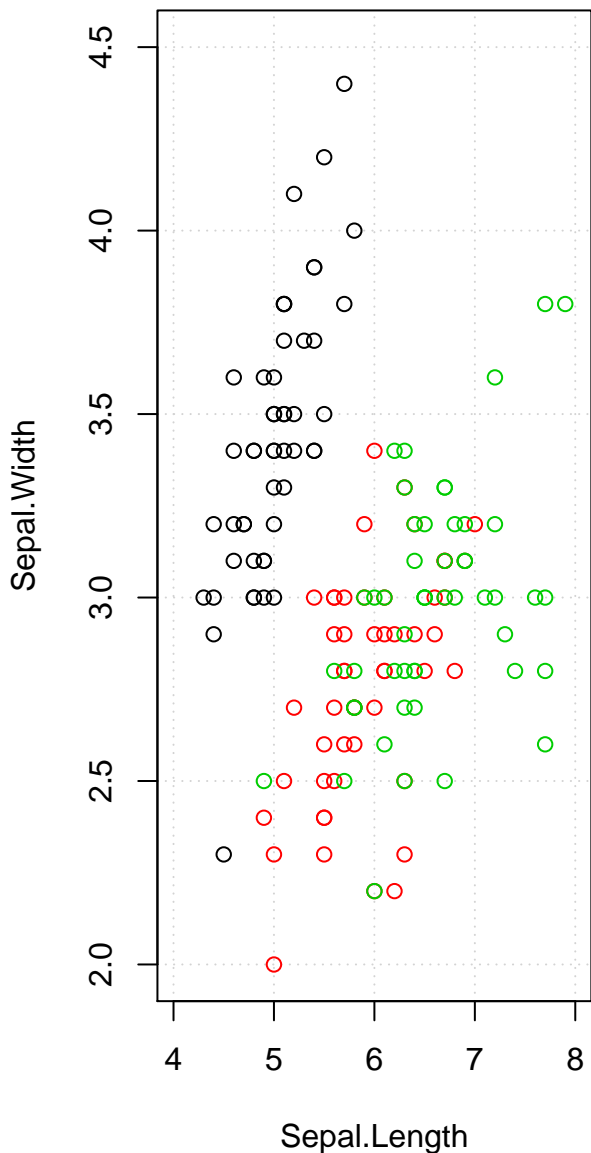




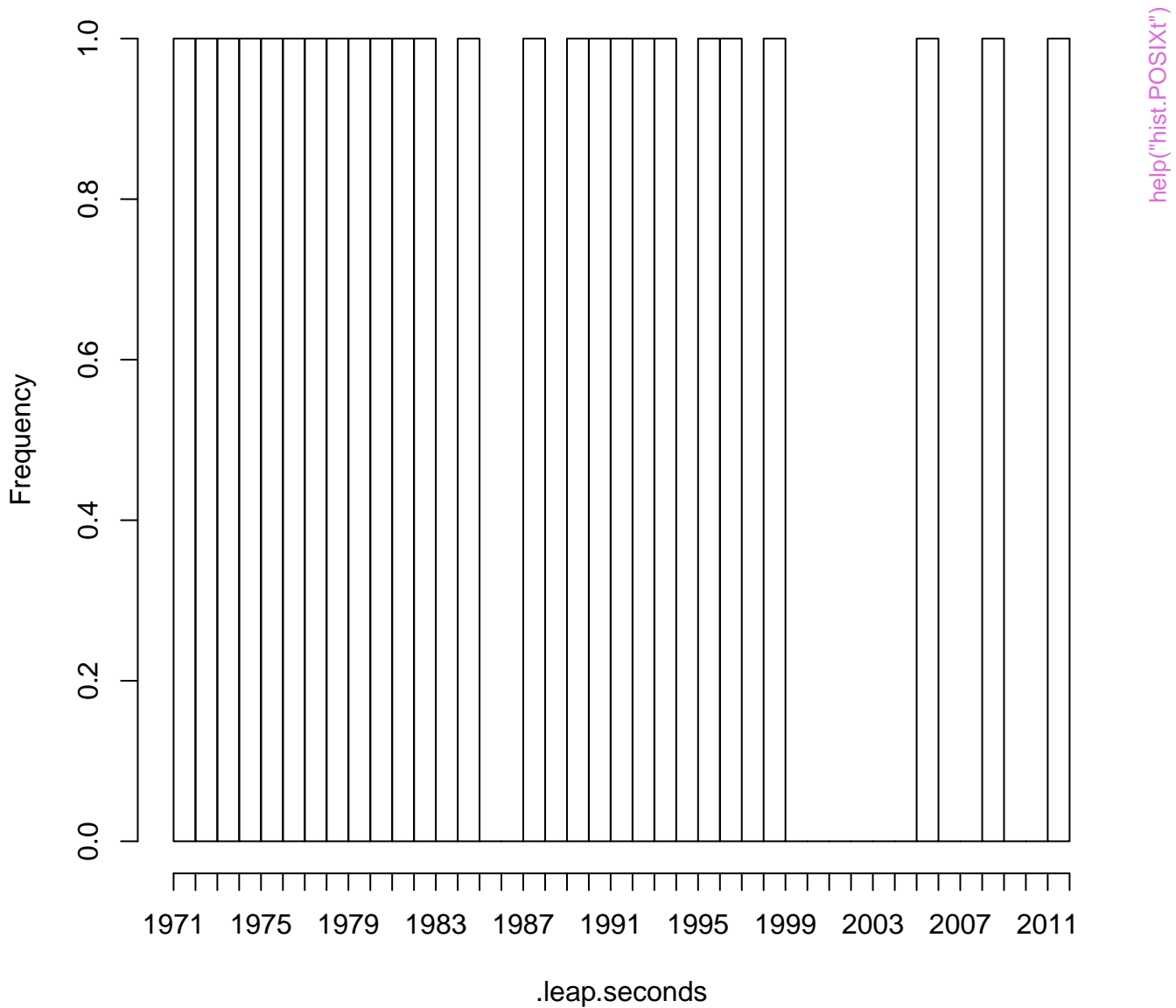




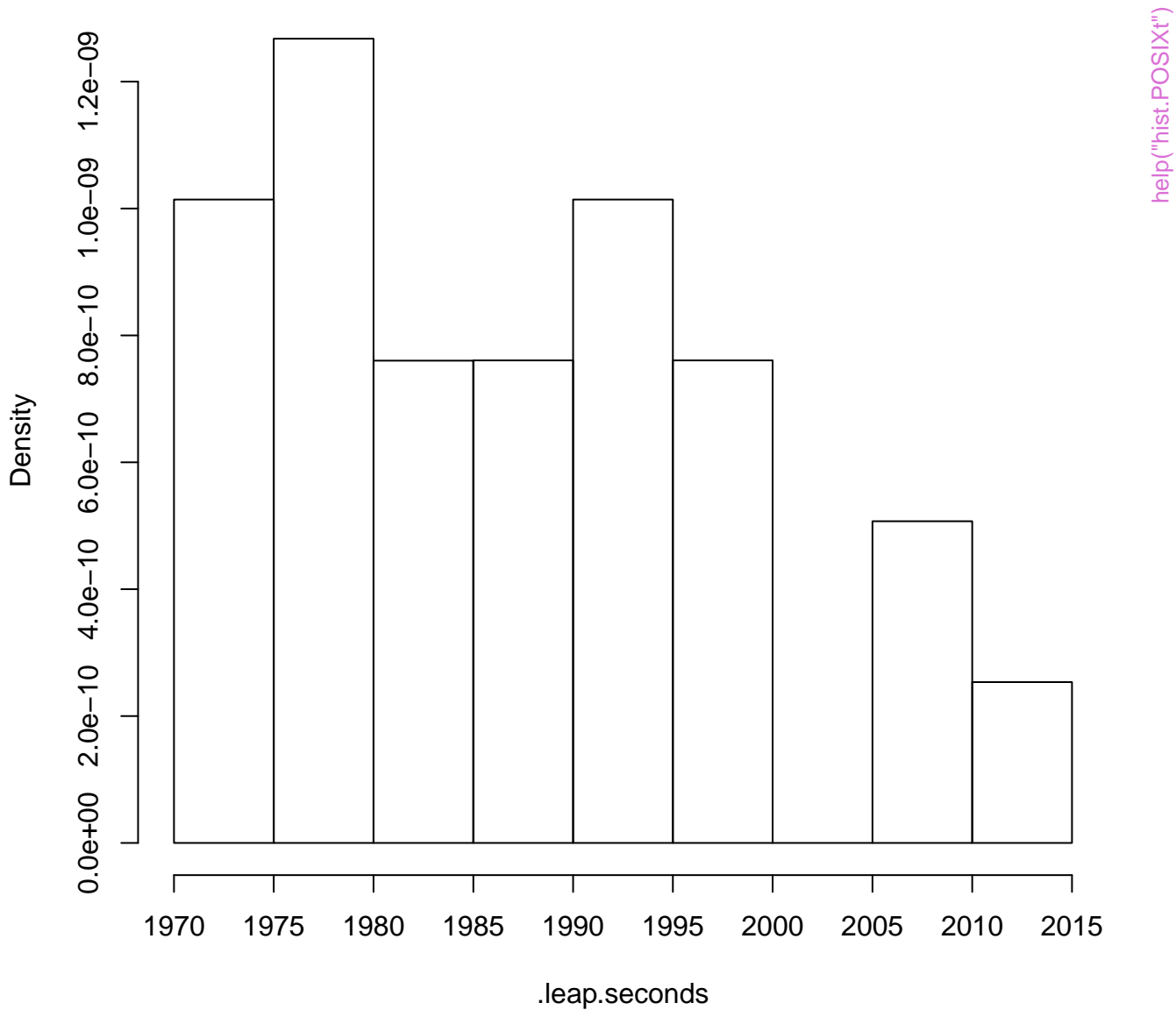
`with(iris, plot(...., panel.first = grid(), ... panel.first = grid(3, lty = 1, lwd = 2))`



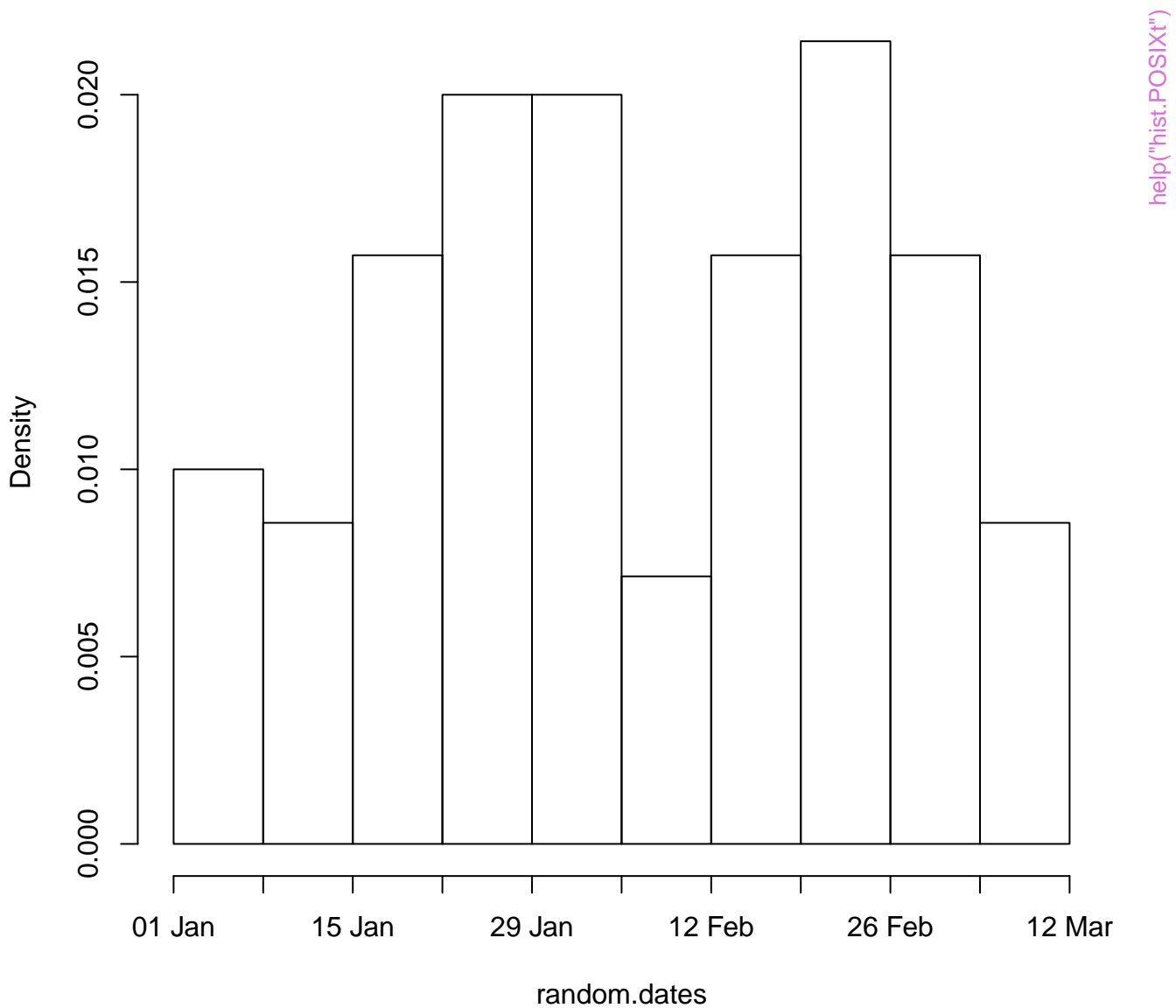
Histogram of .leap.seconds



Histogram of .leap.seconds

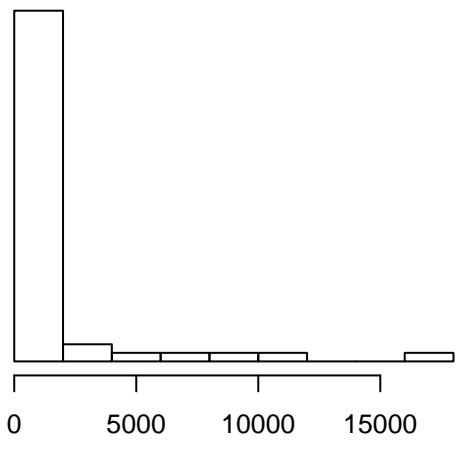


Histogram of random.dates



The histogram shows the frequency of word counts in the text. The x-axis is labeled 'Number of words' and ranges from 0 to 17,500. The y-axis represents frequency, with a maximum value of 10. The distribution is highly right-skewed, with the highest frequency (10) occurring in the first bin (0-2,500 words). The frequency drops sharply for subsequent bins, with most values being 1 or 2, and a single bin at the far right (15,000-17,500 words) having a frequency of 1.

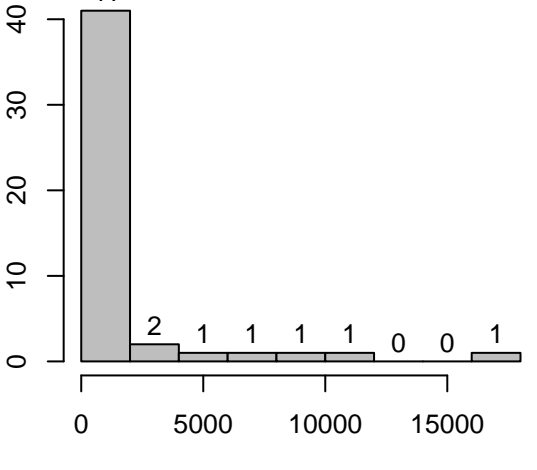
Number of words (bin range)	Frequency
0 - 2,500	10
2,500 - 5,000	1
5,000 - 7,500	1
7,500 - 10,000	1
10,000 - 12,500	1
12,500 - 15,000	0
15,000 - 17,500	1



Frequency

Number of non-zero elements

Number of non-zero elements	Frequency
0	40
1000	2
2000	1
3000	1
4000	1
5000	1
6000	0
7000	0
8000	1



The histogram shows the frequency of the number of children per family. The x-axis is labeled 'Number of children' and ranges from 0 to 140. The y-axis is labeled 'Frequency' and ranges from 0 to 100. The distribution is highly right-skewed, with a peak at 0 children (frequency ~100) and a long tail extending to 140 children.

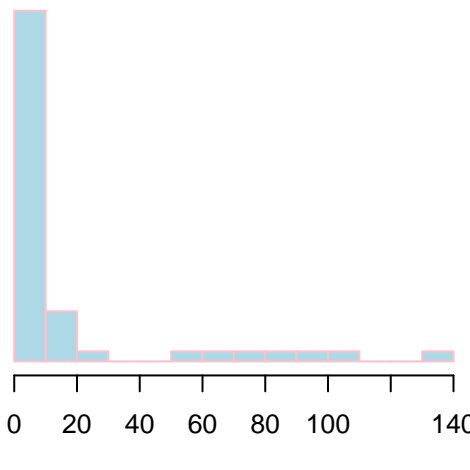
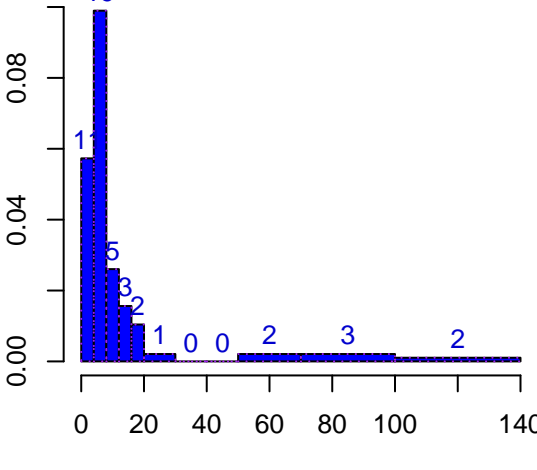
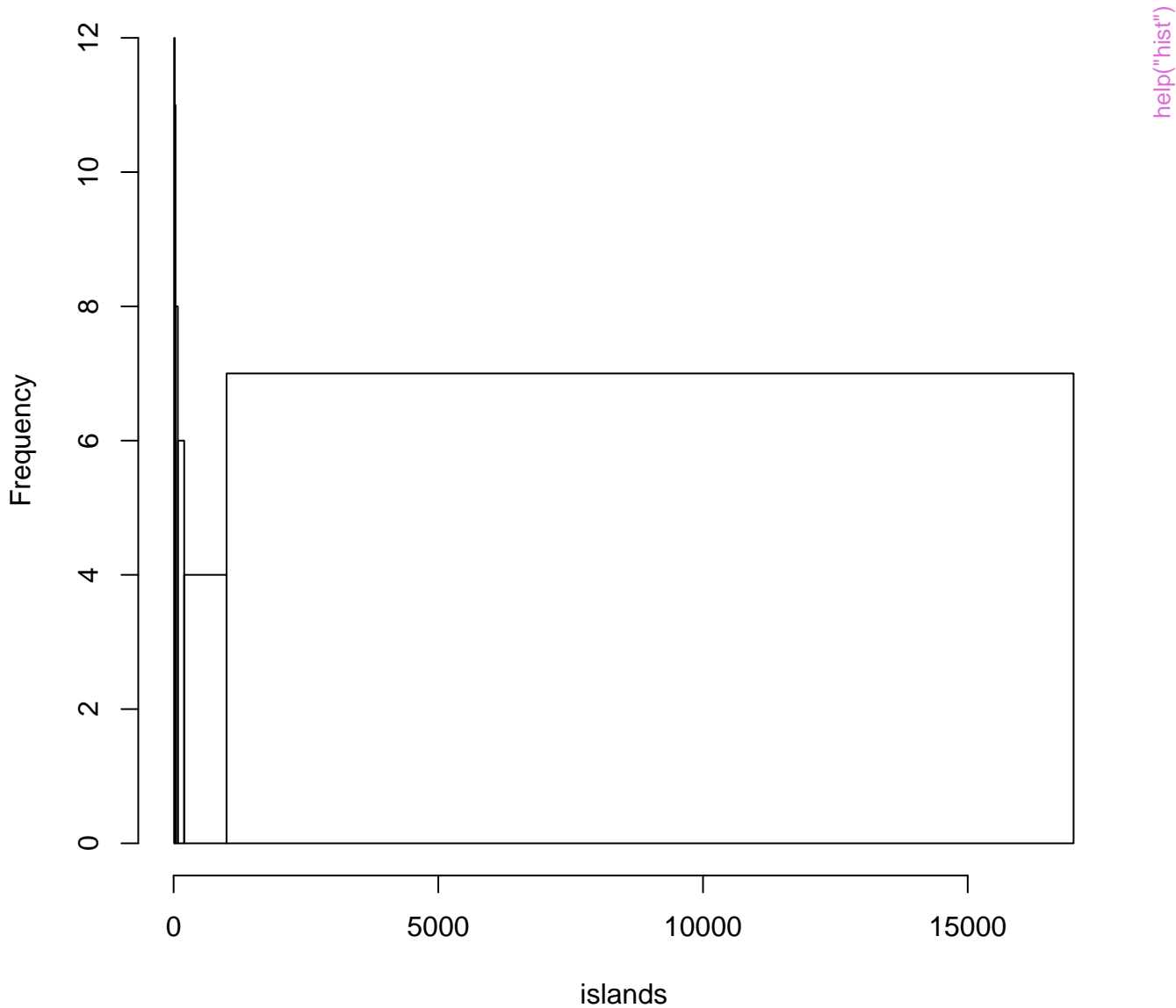


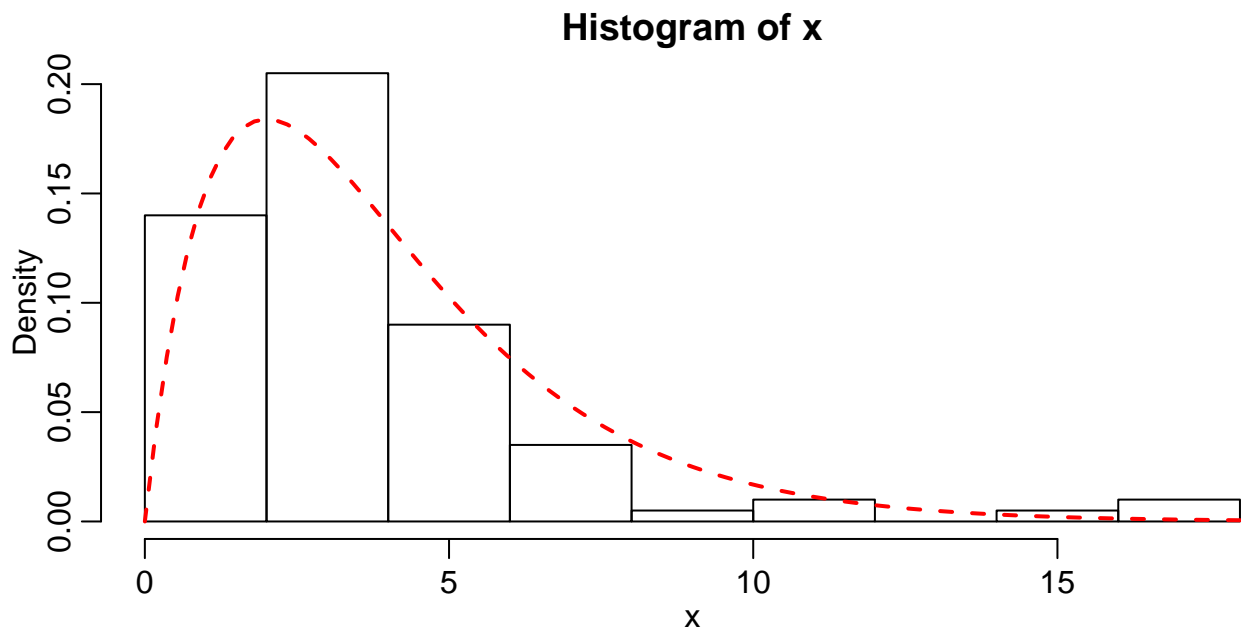
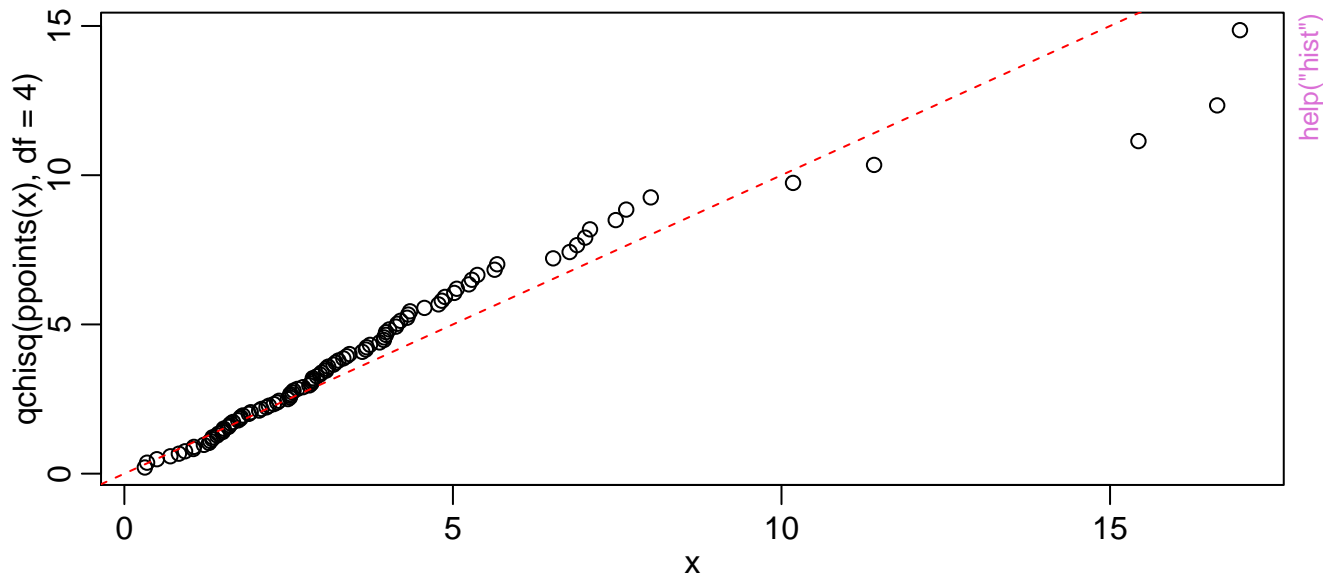
Figure 1: Histogram of the number of non-zero elements in the sparse matrix  $A$ . The x-axis represents the number of non-zero elements (0 to 140), and the y-axis represents the frequency (0.00 to 0.08). The distribution is highly skewed, with a peak at 5 non-zero elements. The counts for each bin are: 0: 1, 5: 5, 10: 3, 15: 2, 20: 1, 30: 0, 40: 0, 50: 2, 60: 3, 70: 3, 80: 2.

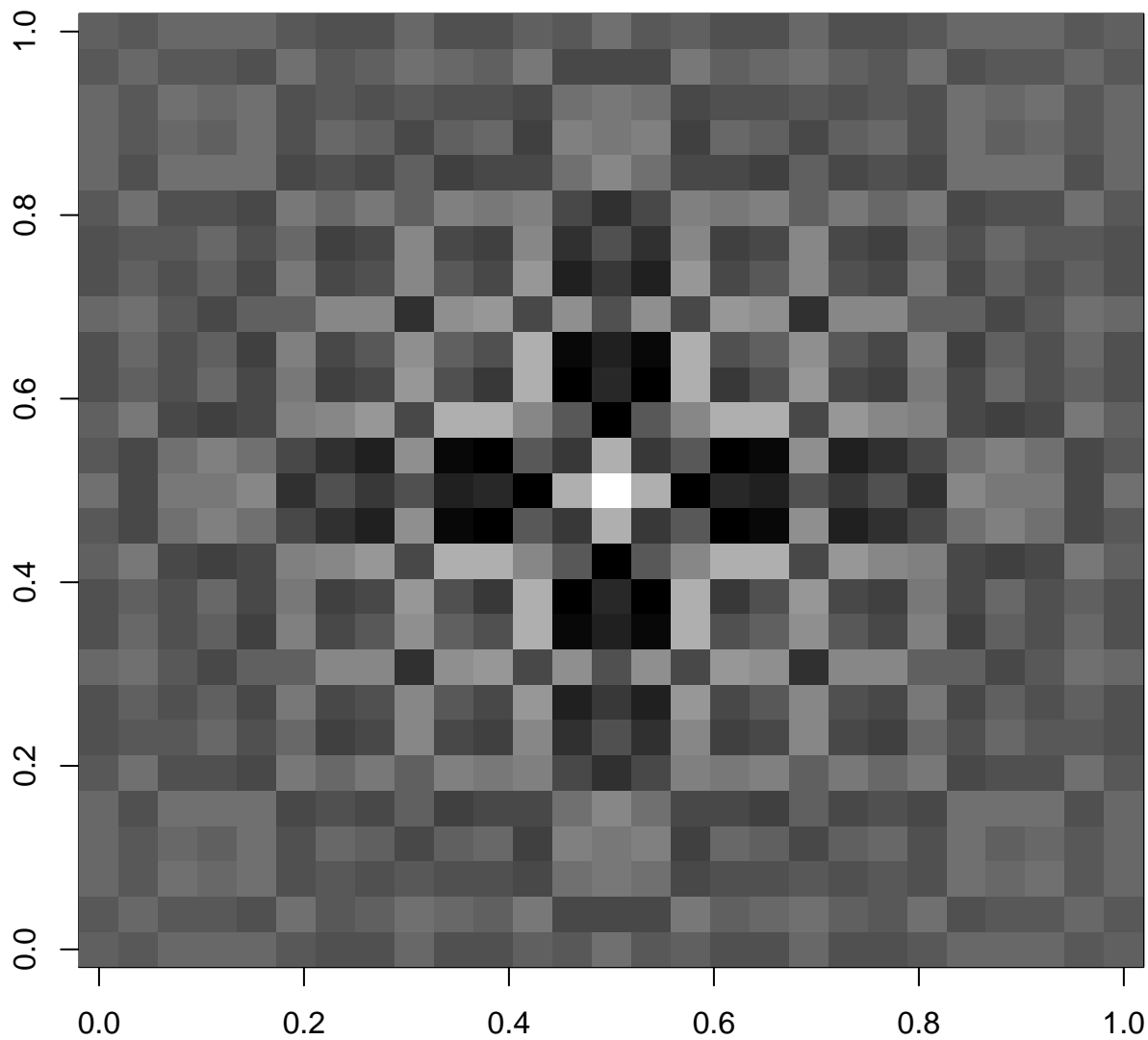




## WRONG histogram

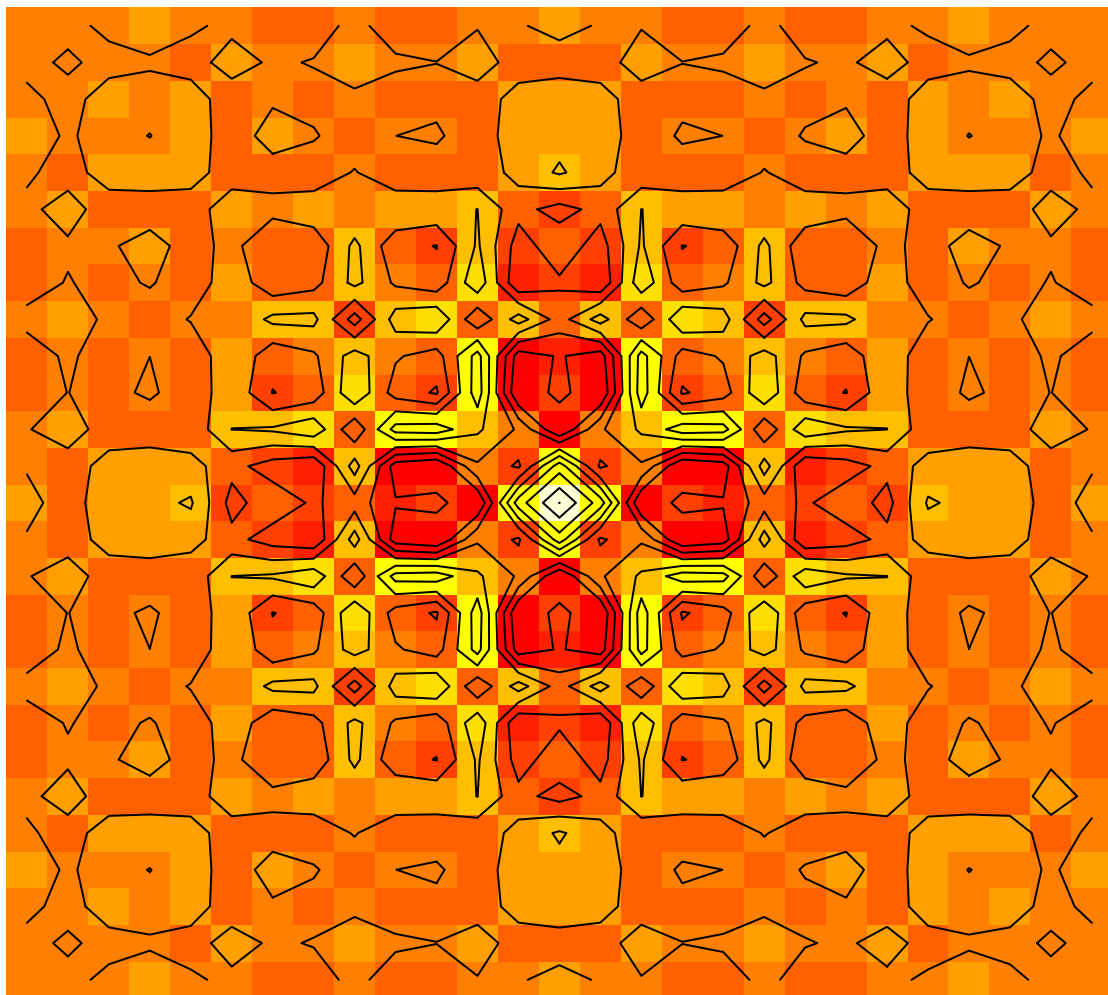






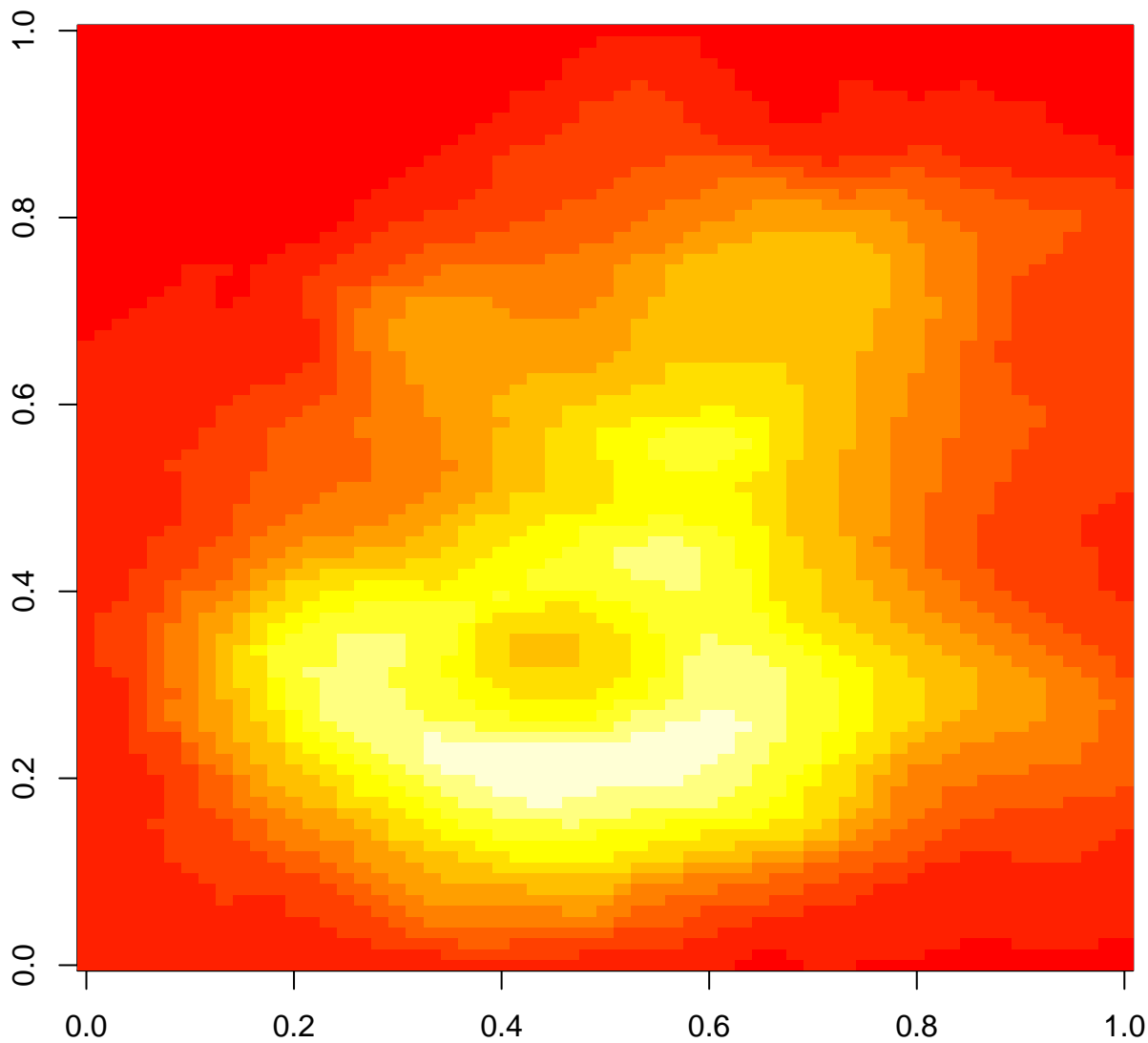
help("image")

Math can be beautiful ...



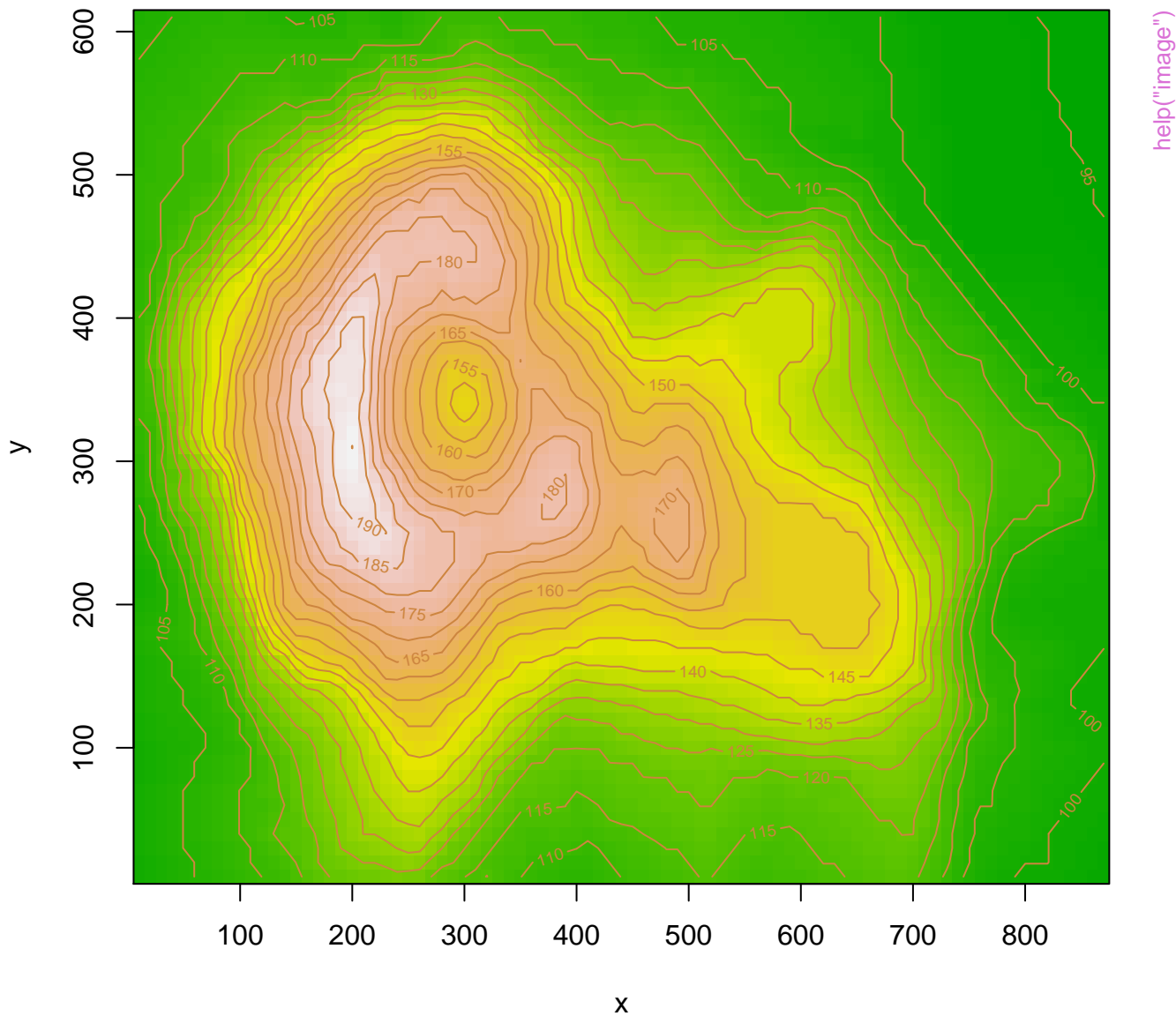
help("image")

$$\cos(r^2)e^{-r/6}$$



`help("image")`

# Maunga Whau Volcano



1

2

1

2



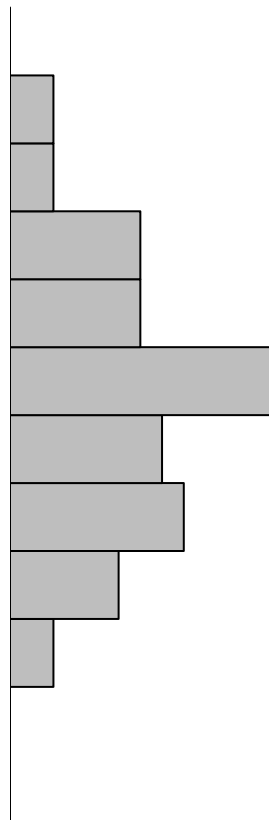
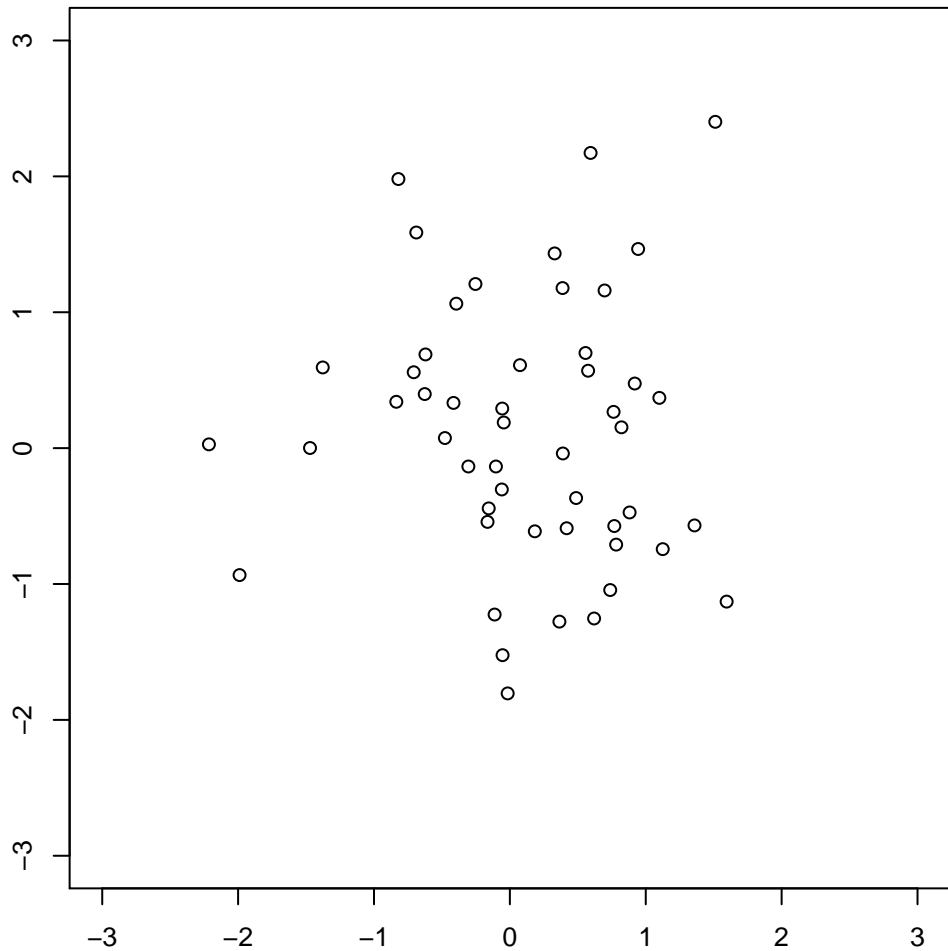
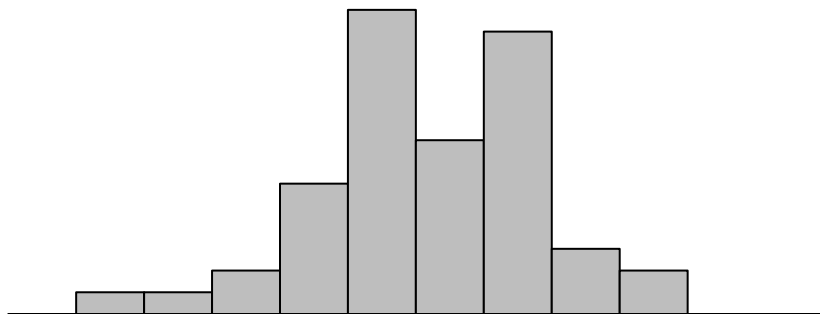
1

help("layout")

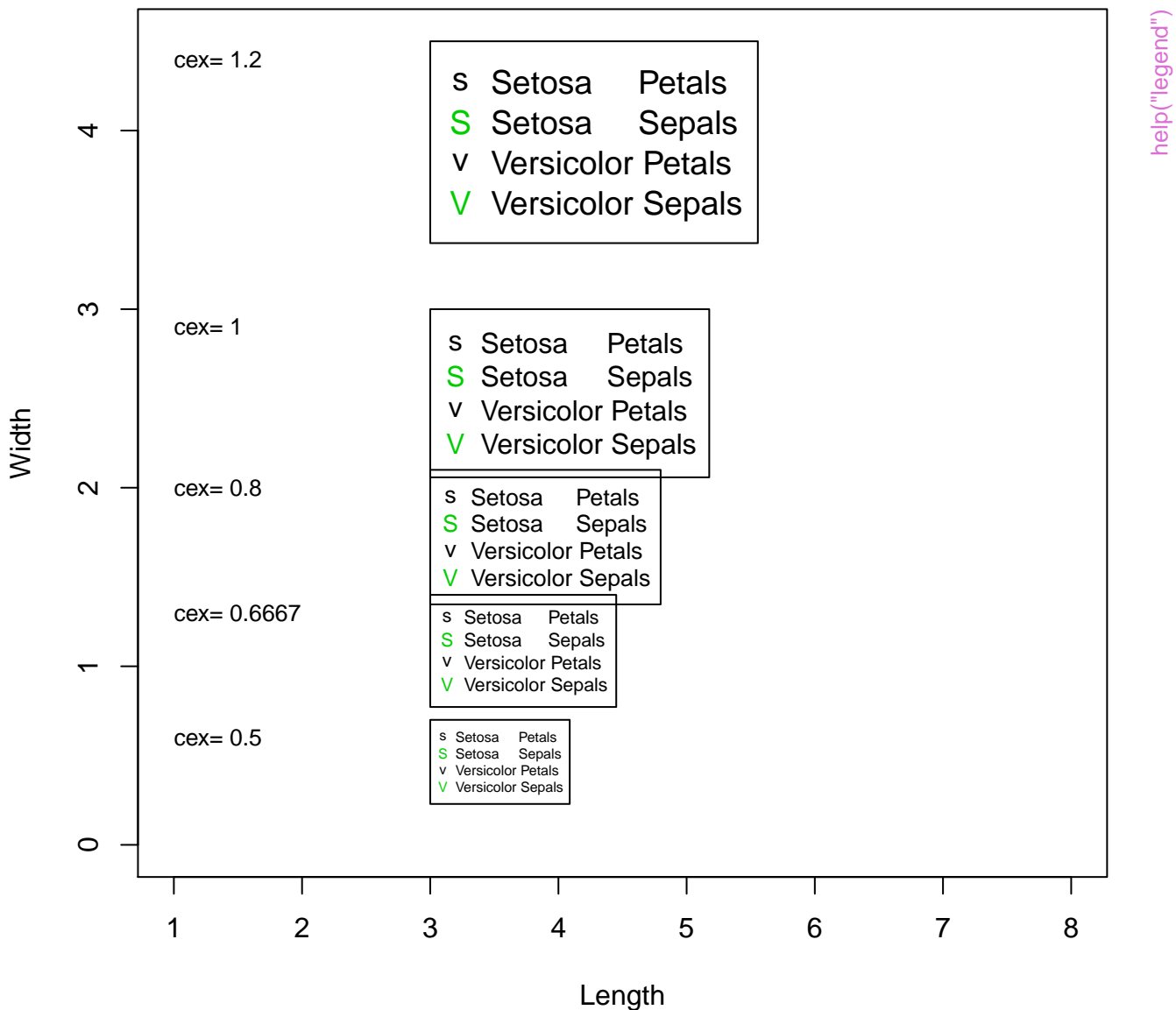
2

1

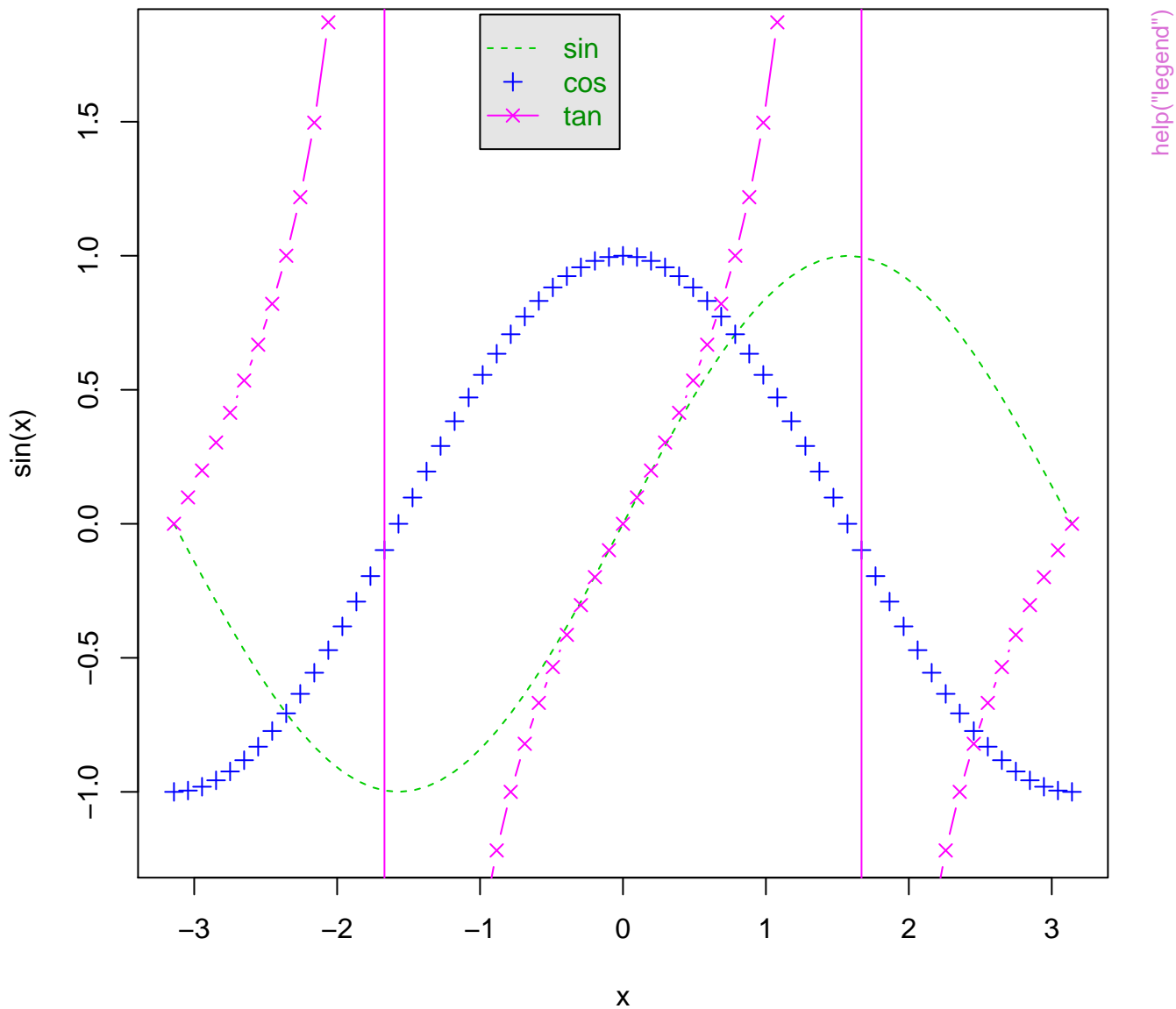
3

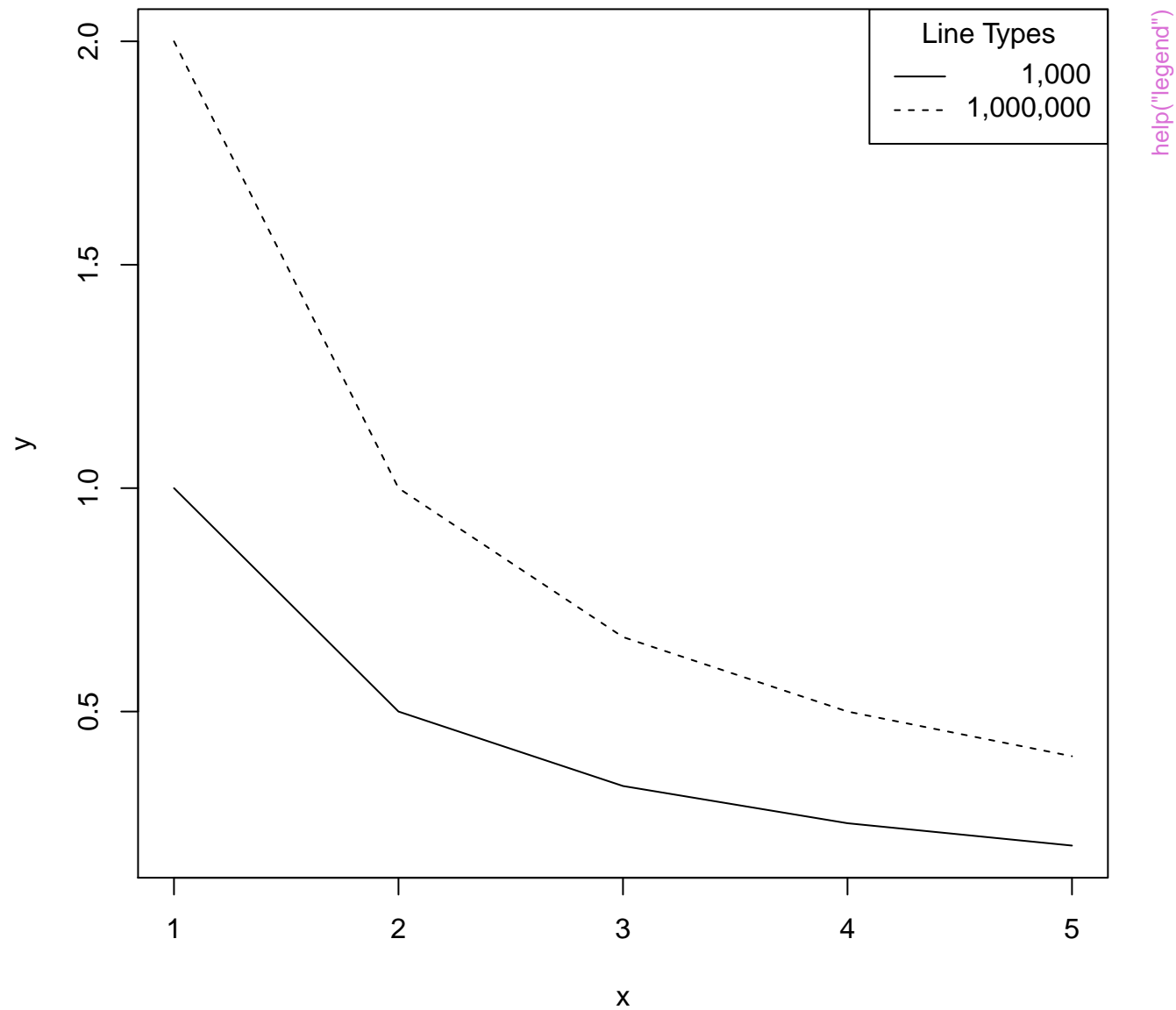


# Petal and Sepal Dimensions in Iris Blossoms

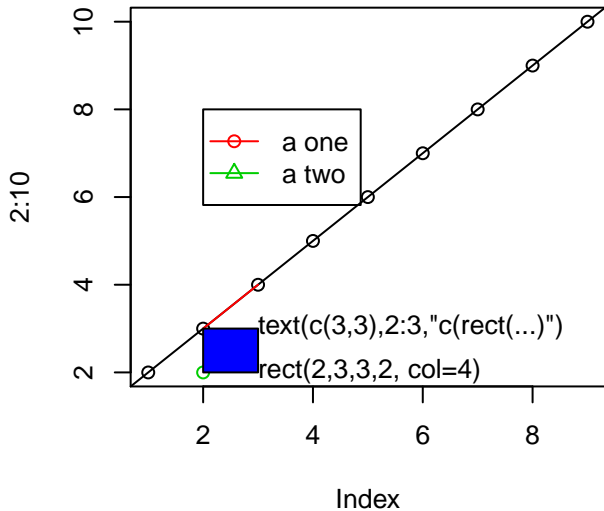


**legend(..., lty = c(2, -1, 1), pch = c(NA, 3, 4), merge = TRUE)**

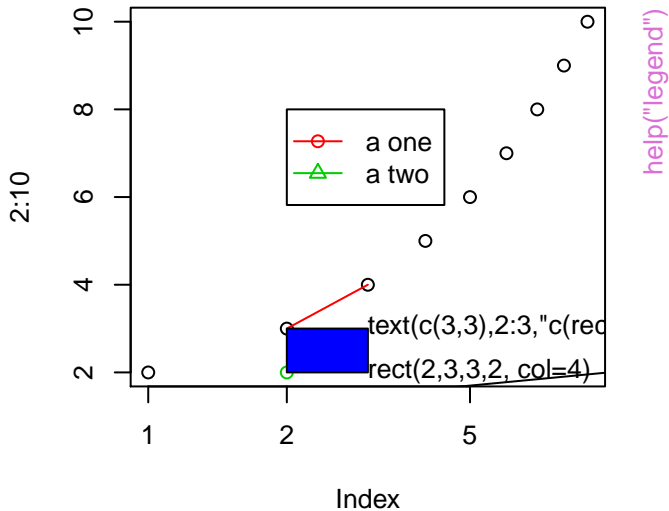




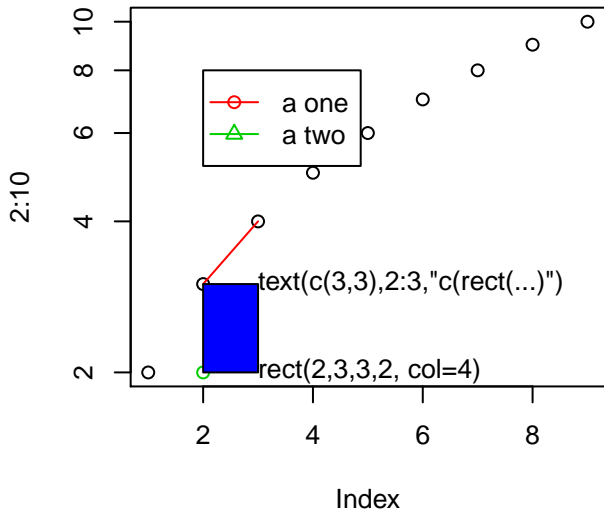
log = ""



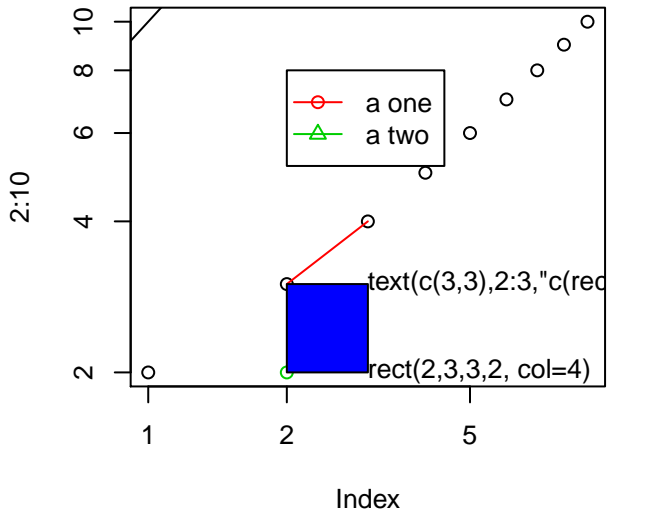
log = 'x'

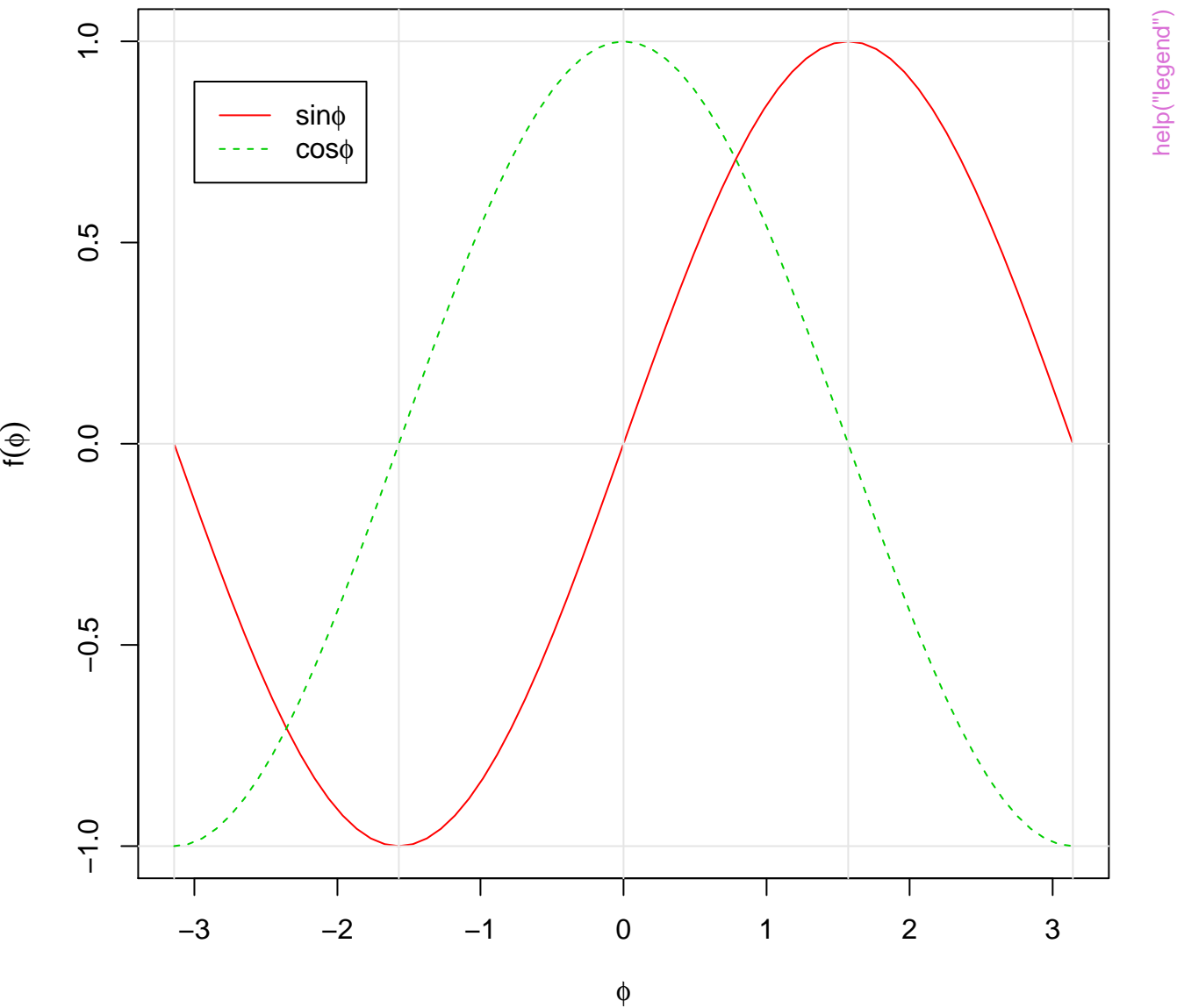


log = 'y'



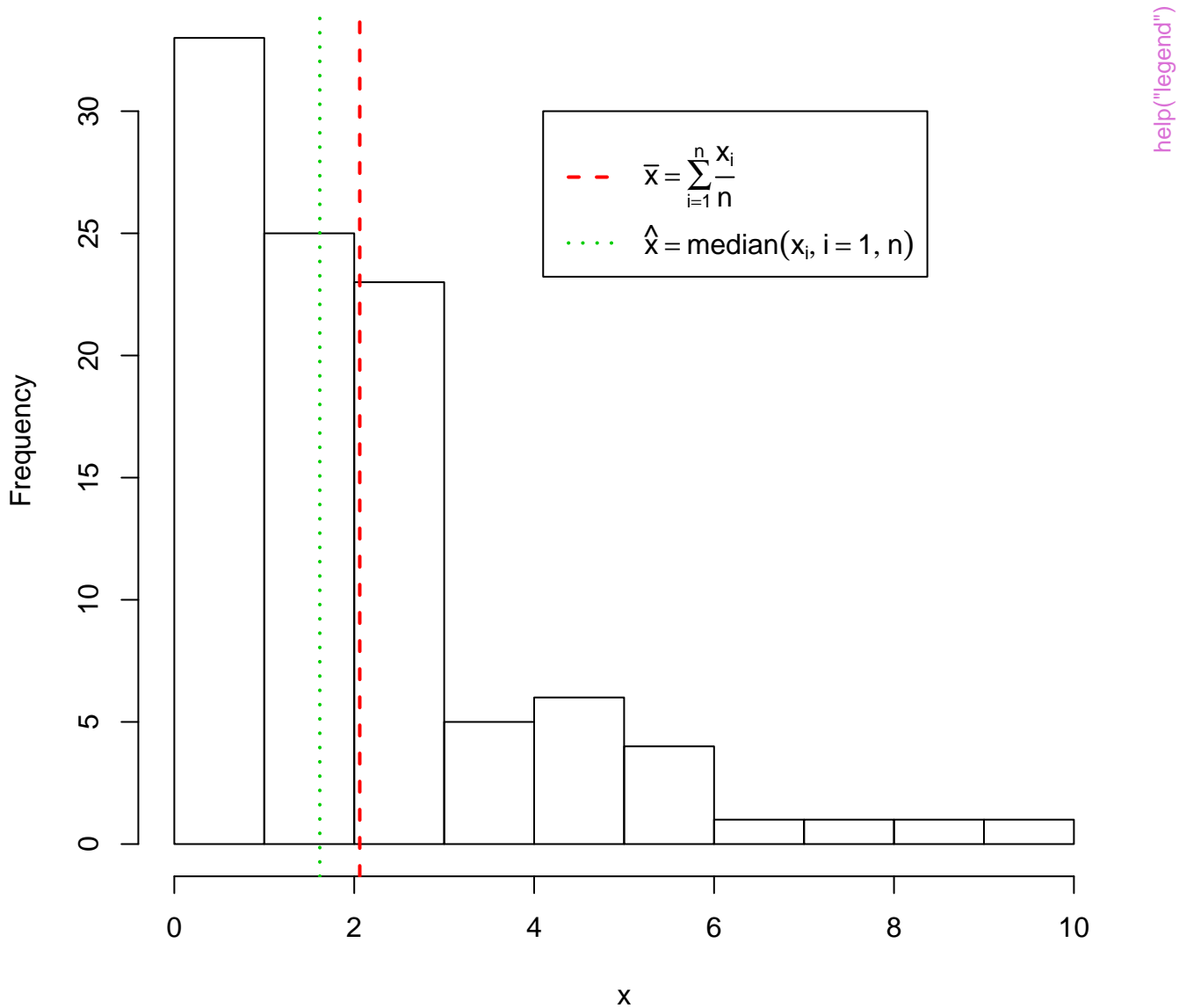
log = 'xy'

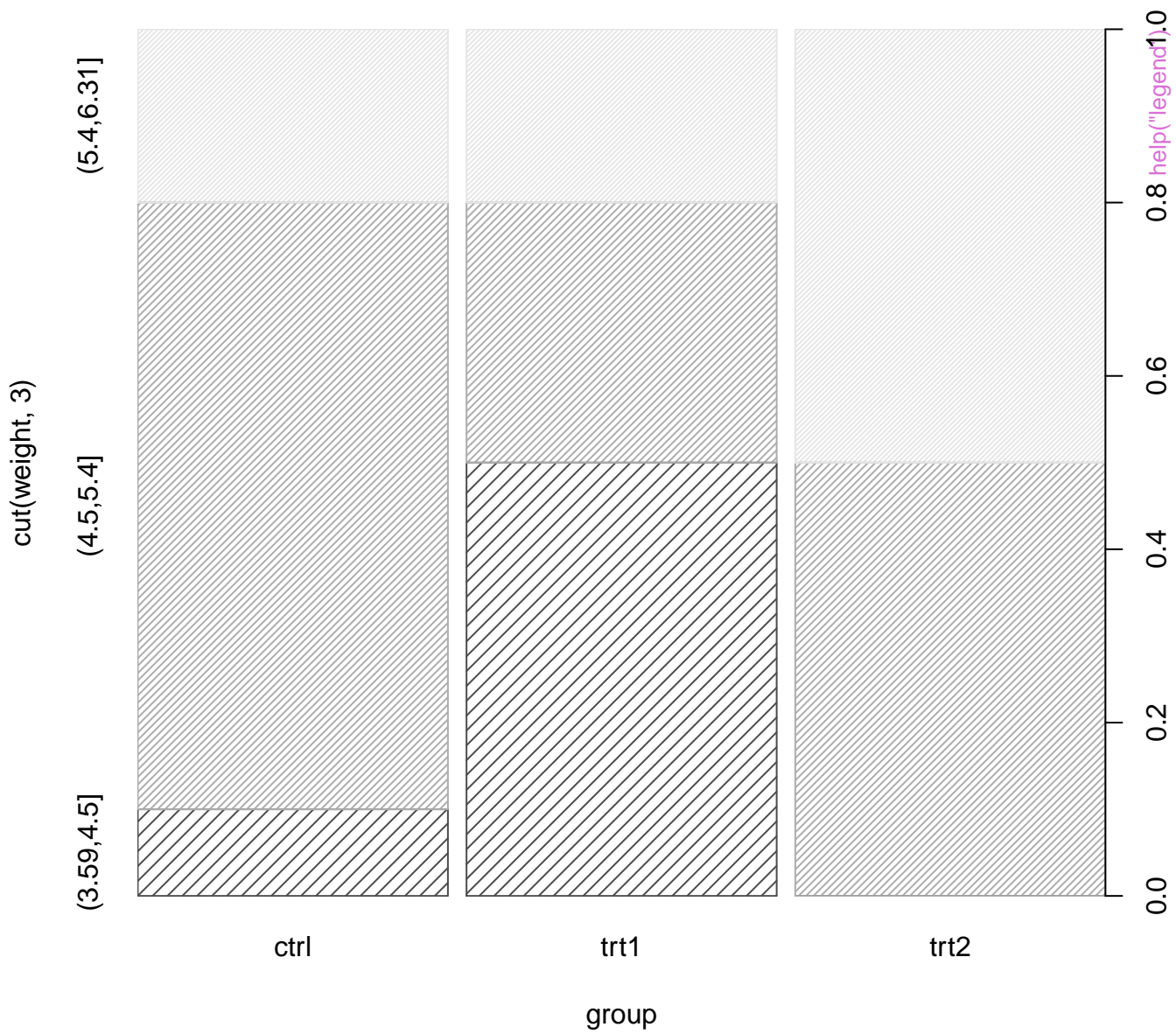


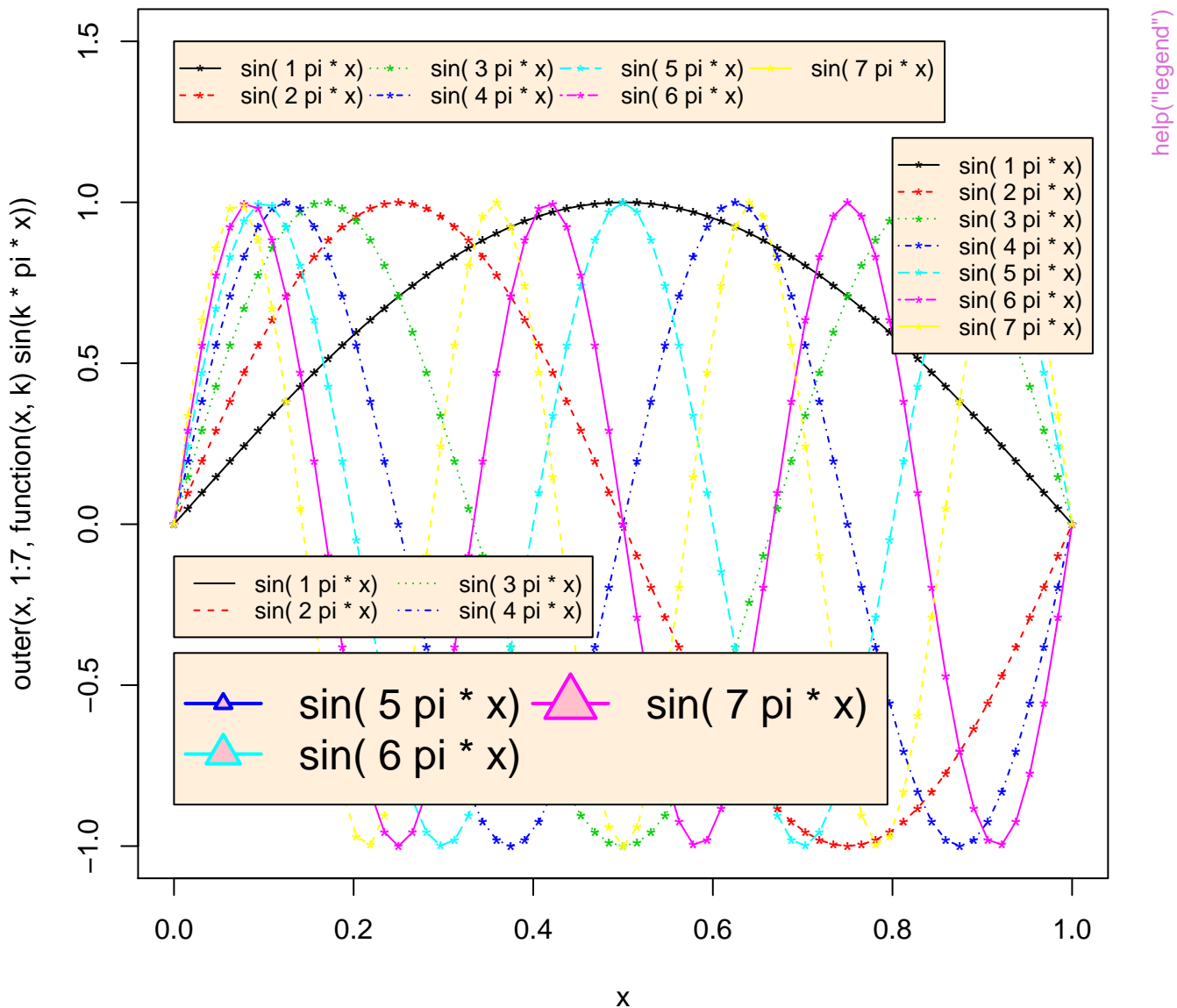




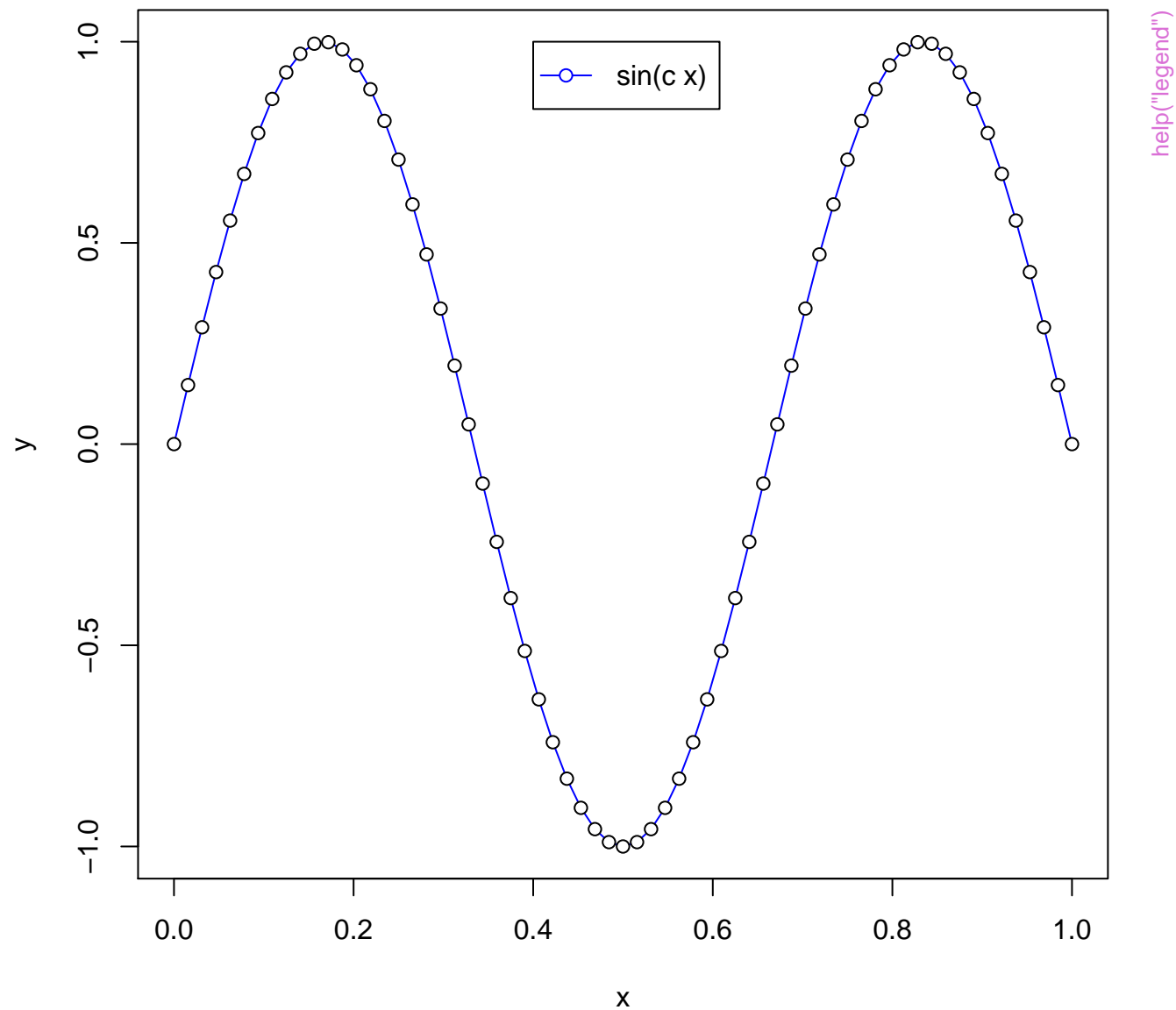
# Mean and Median of a Skewed Distribution

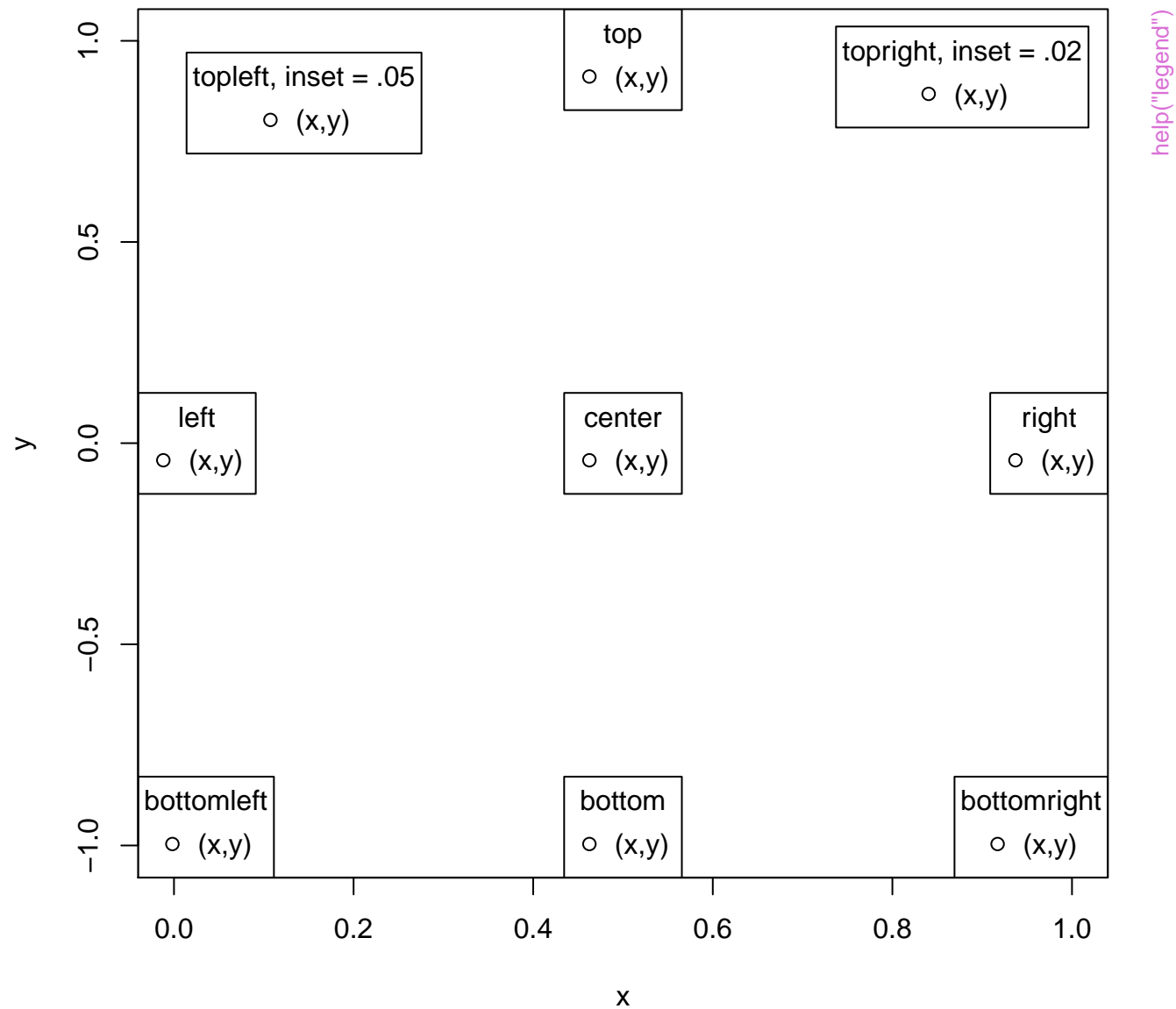






points with bg & legend(\*, pt.bg)





text.font = 1

——	A	——	D
——	B	——	E
——	C	——	F

text.font = 2

——	A	——	D
——	B	——	E
——	C	——	F

text.font = 3

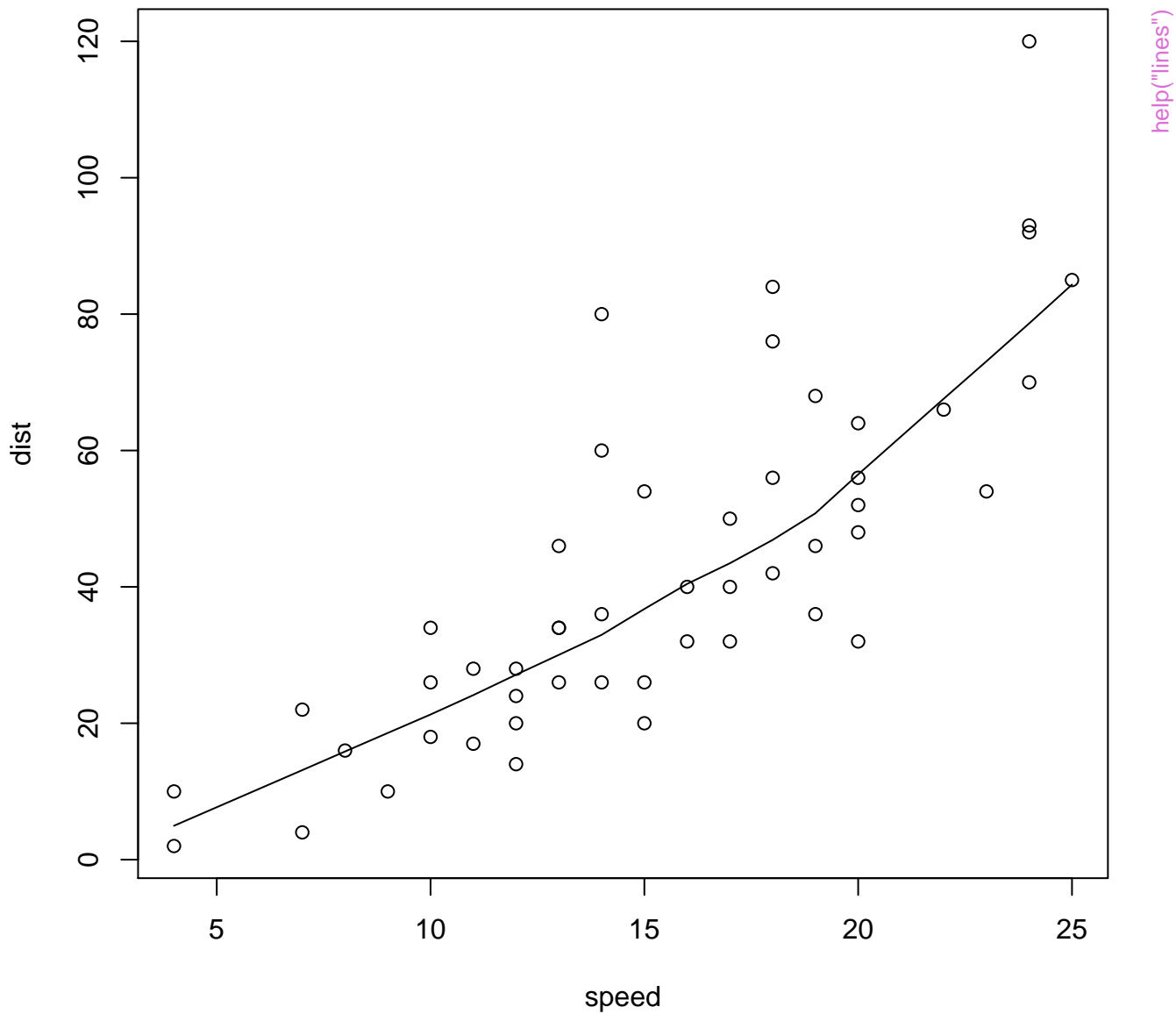
——	<i>A</i>	——	<i>D</i>
——	<i>B</i>	——	<i>E</i>
——	<i>C</i>	——	<i>F</i>

text.font = 4

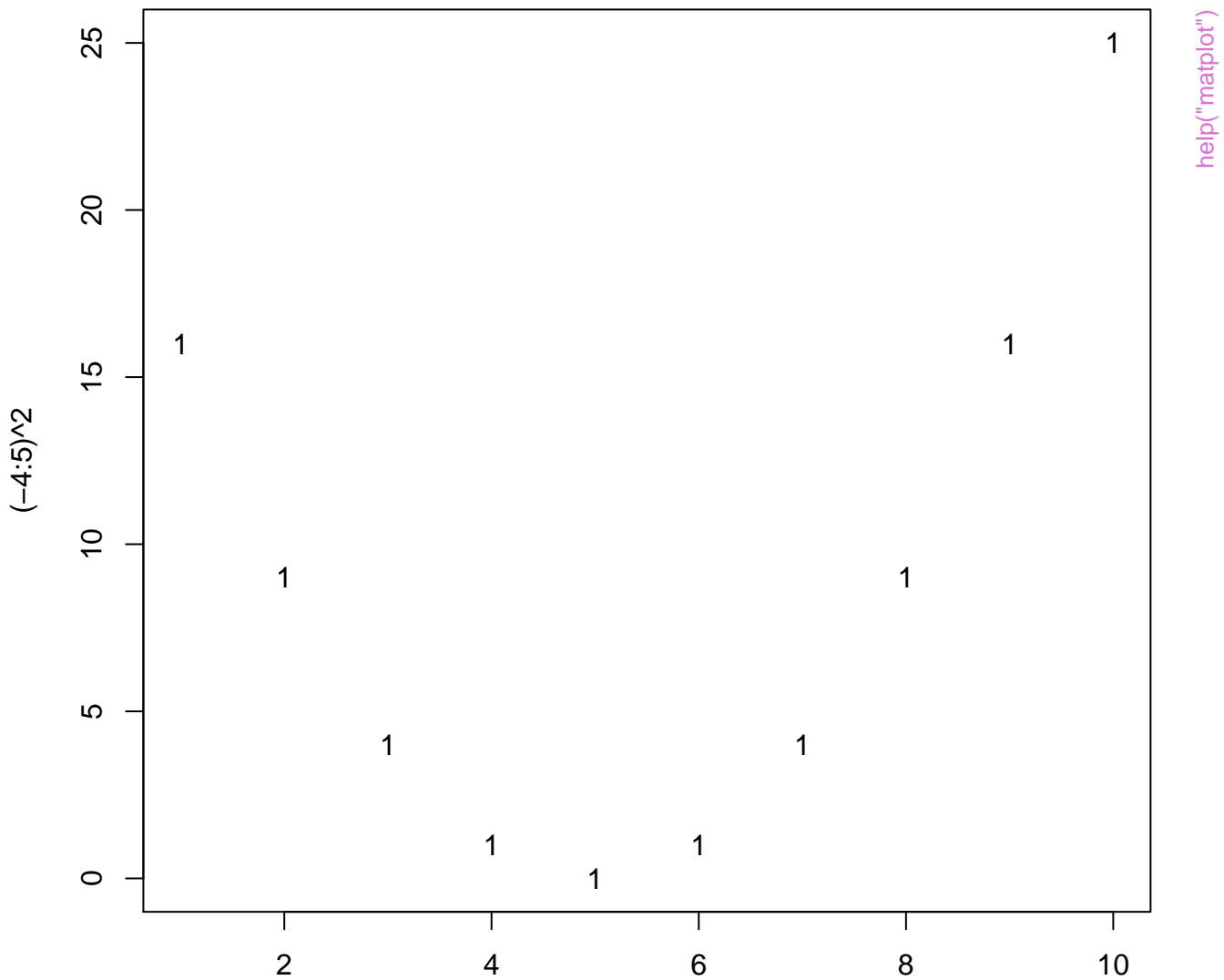
——	<b><i>A</i></b>	——	<b><i>D</i></b>
——	<b><i>B</i></b>	——	<b><i>E</i></b>
——	<b><i>C</i></b>	——	<b><i>F</i></b>

help("legend")

# Stopping Distance versus Speed

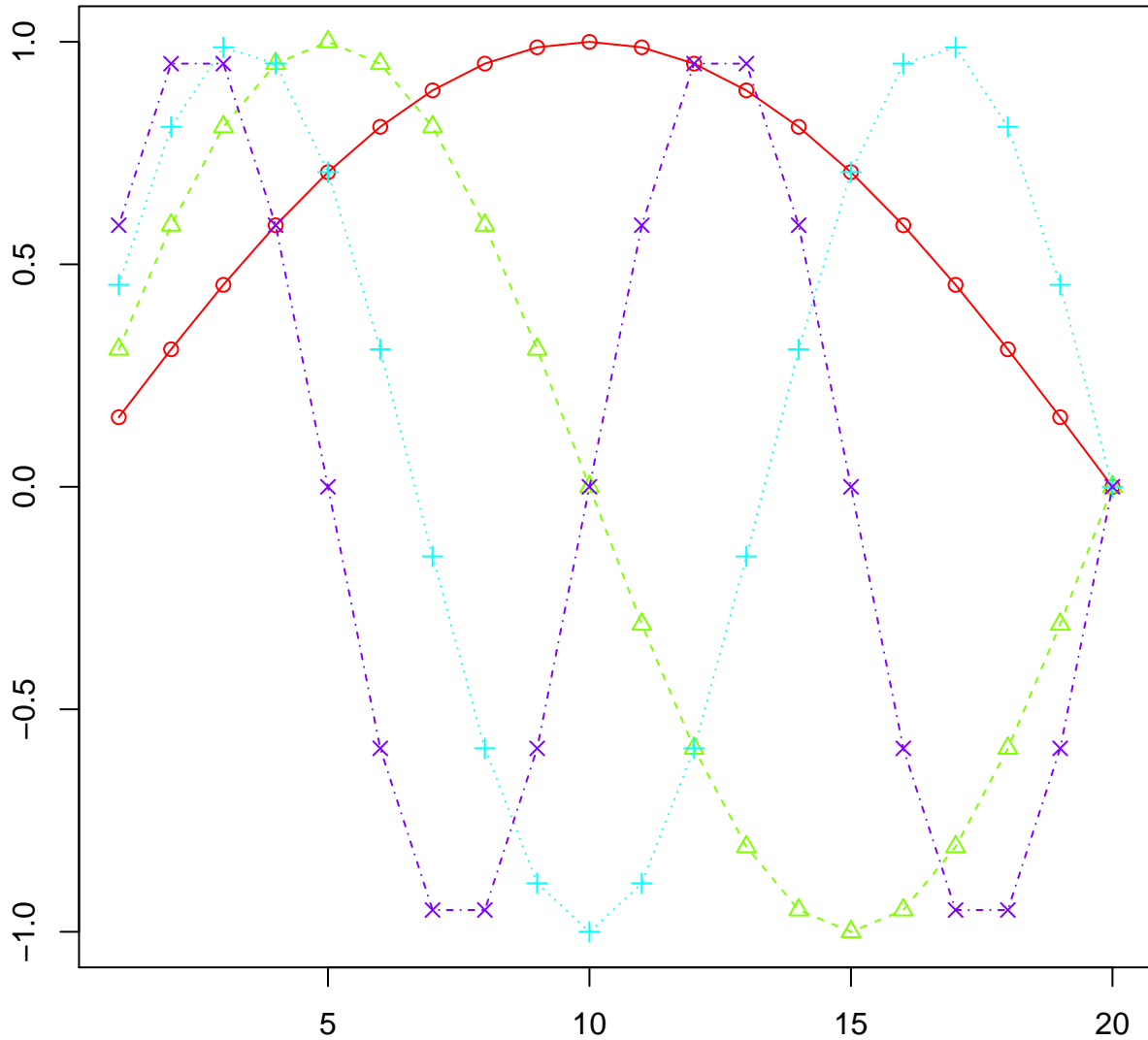


# Quadratic



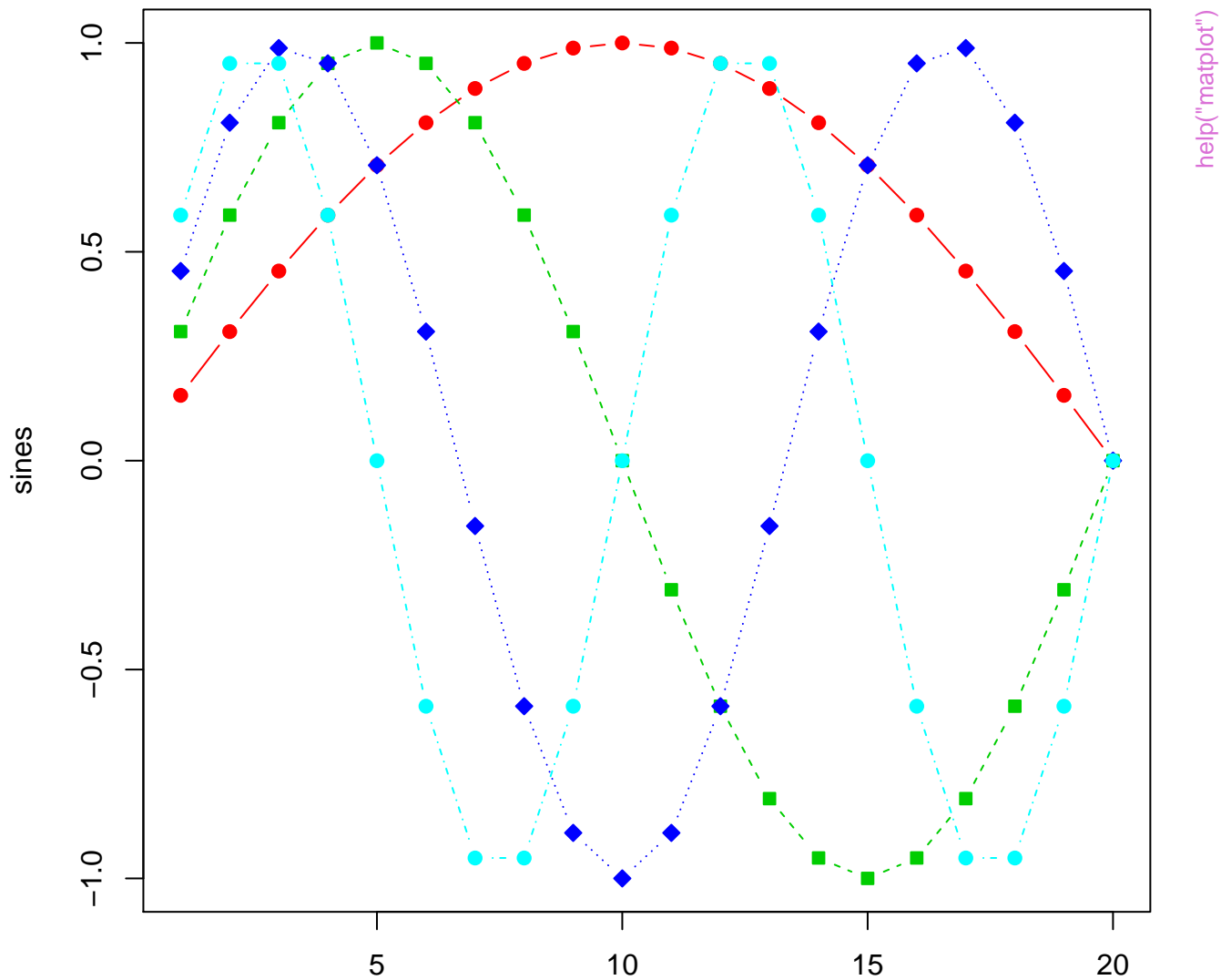


sines



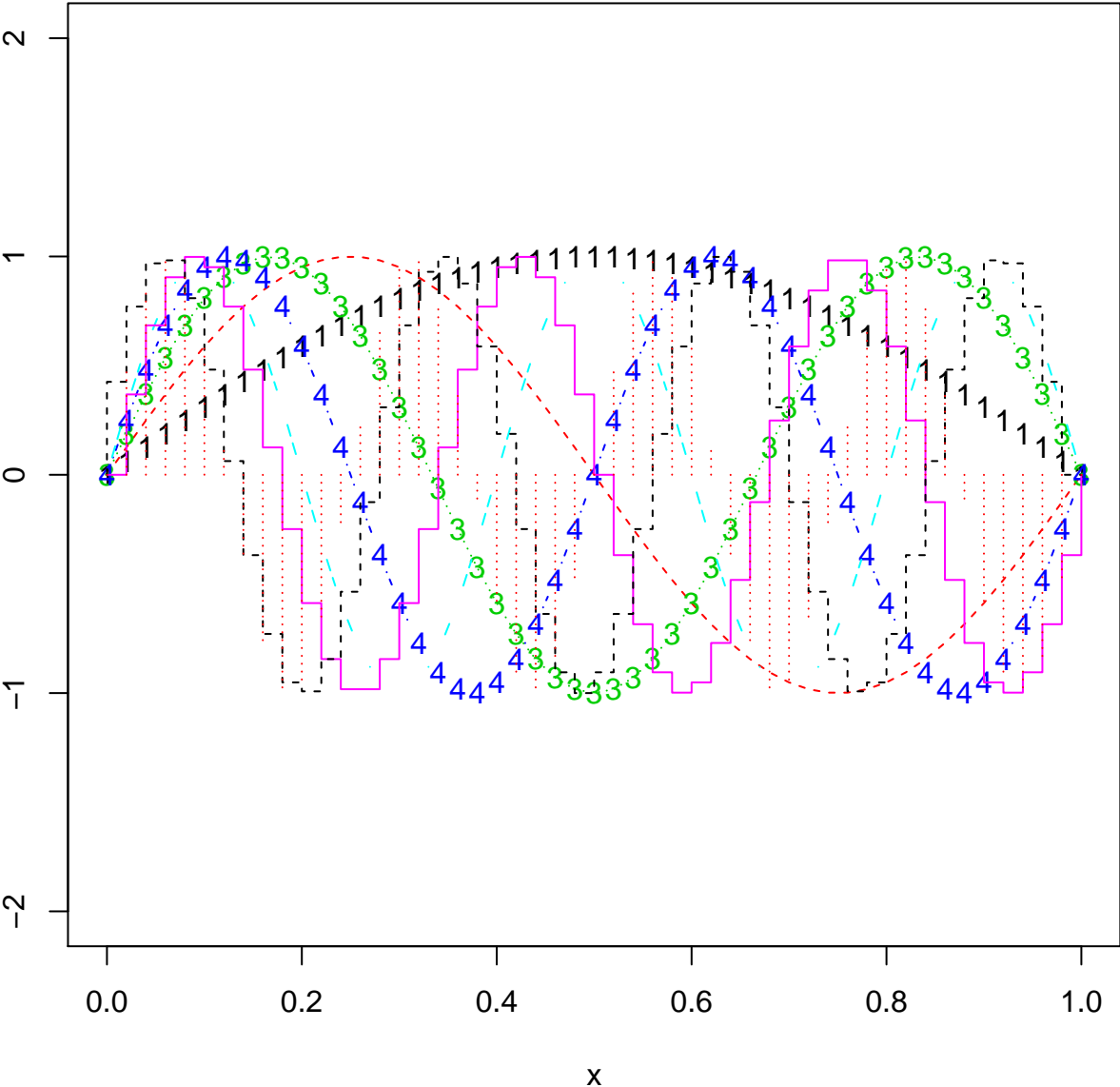
help("matplotlib")

`matplot(..., pch = 21:23, bg = 2:5)`

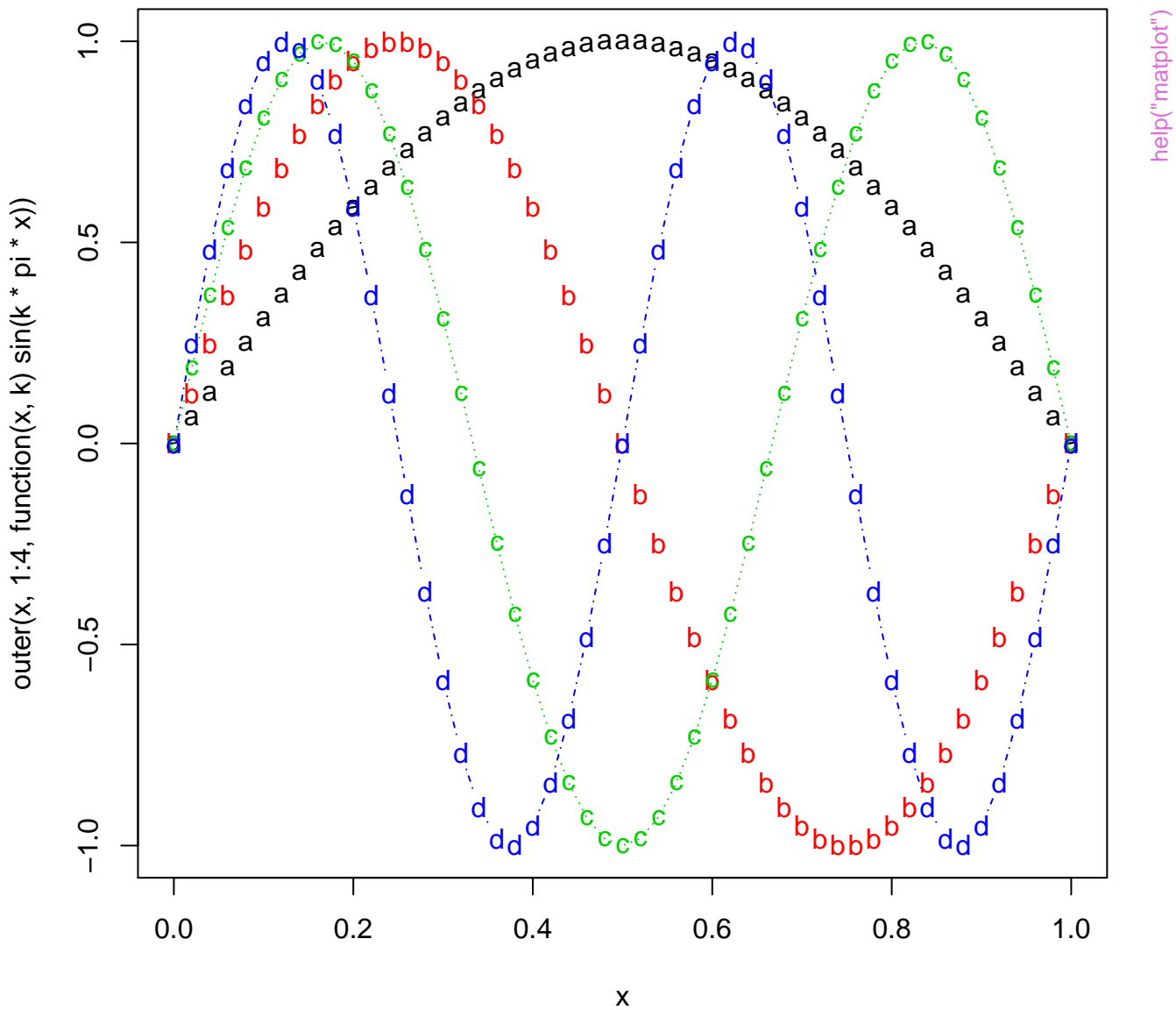


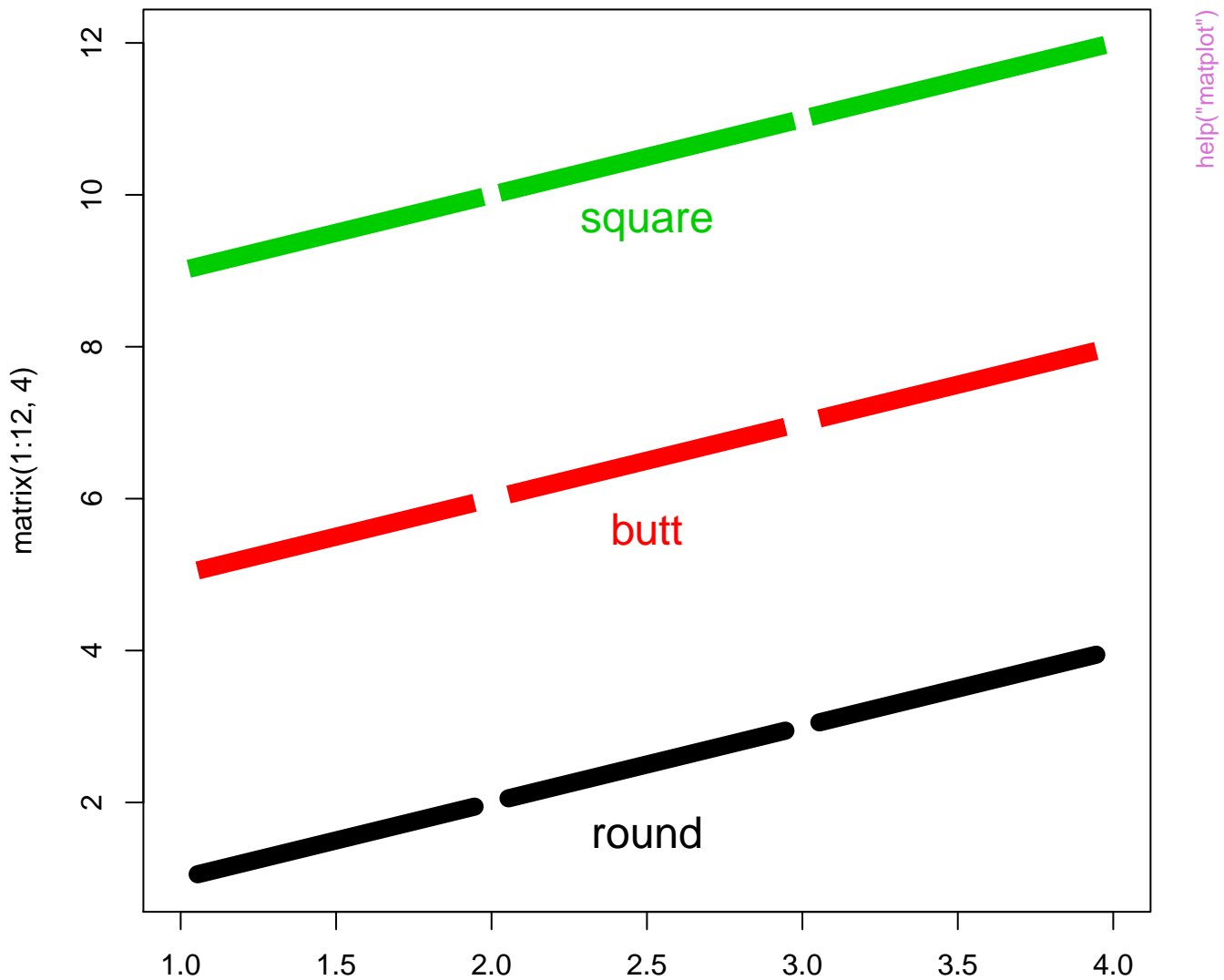
matplot(type = "plobcsSh" )

outer(x, 1:8, function(x, k) sin(k \* pi \* x))



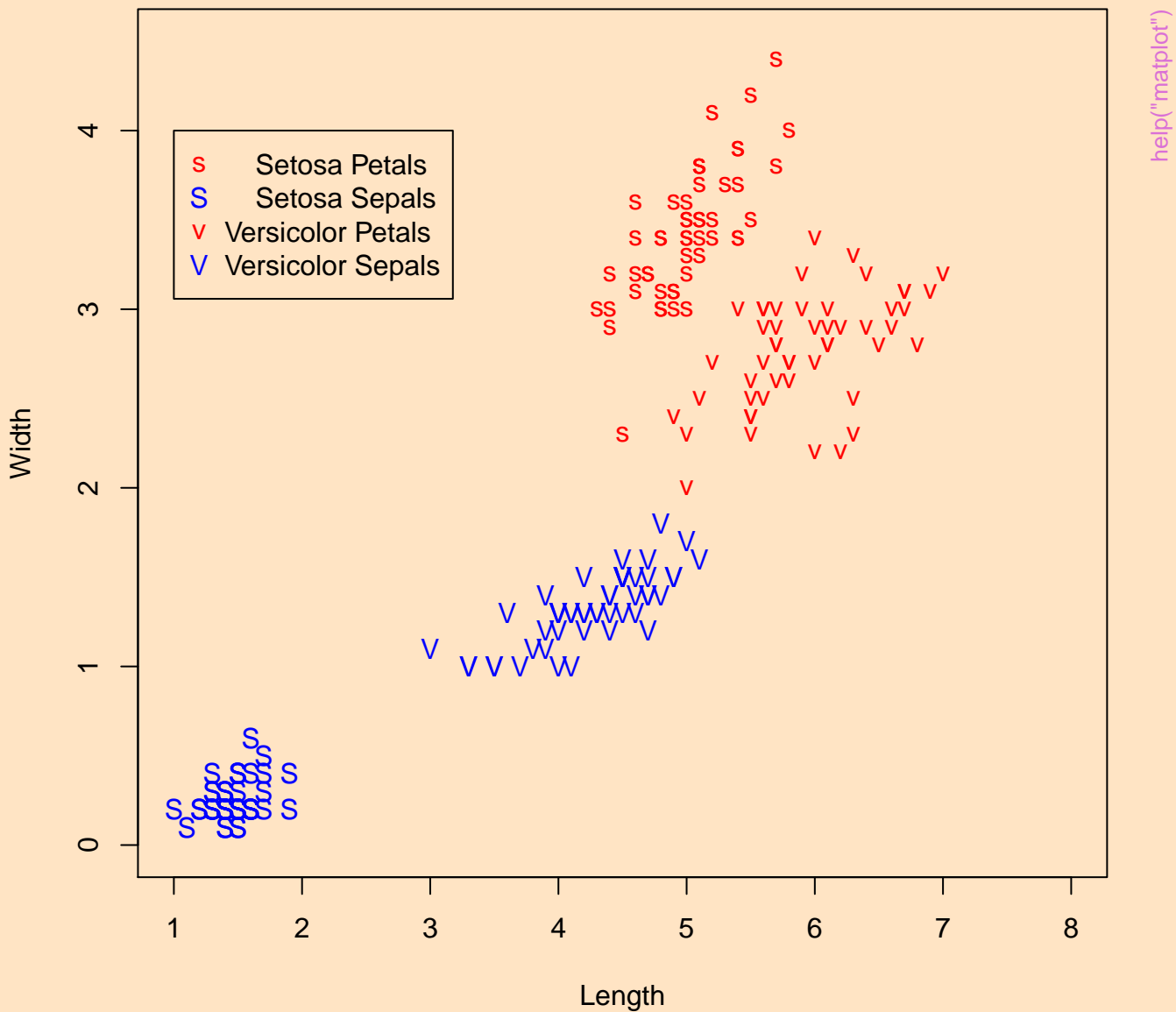
help("matplot")



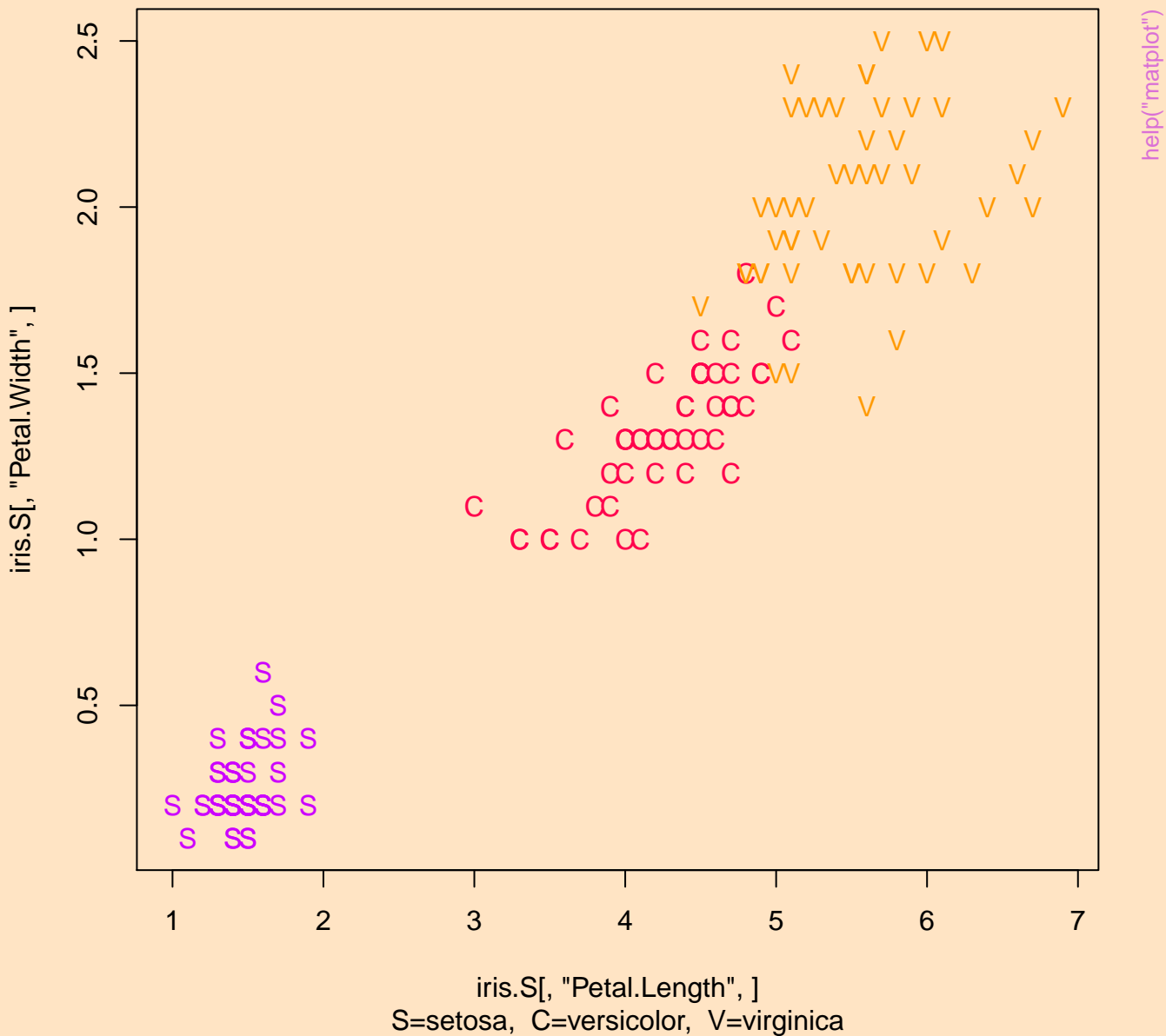


`help("matplot")`

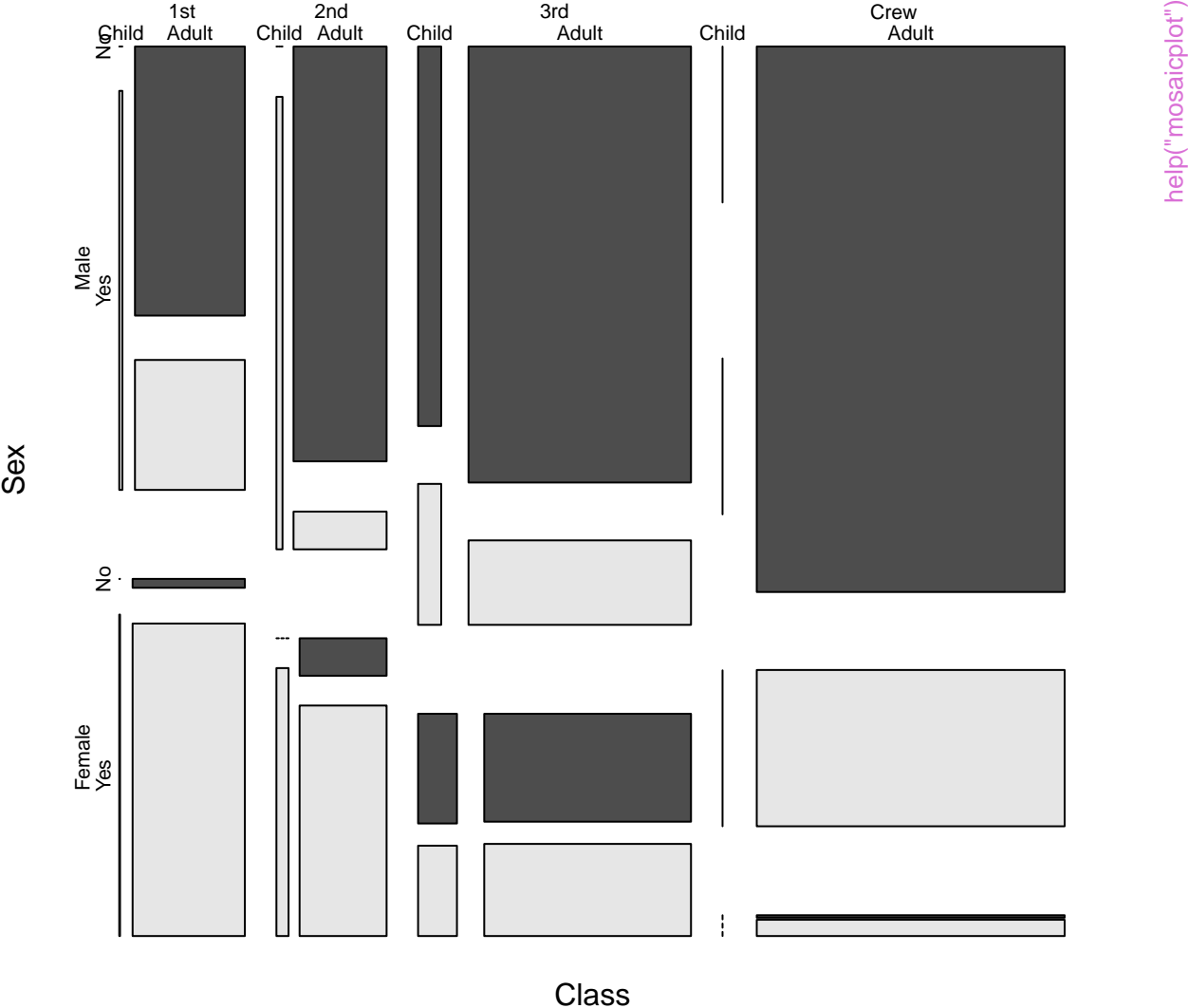
# Petal and Sepal Dimensions in Iris Blossoms



# Fisher's Iris Data

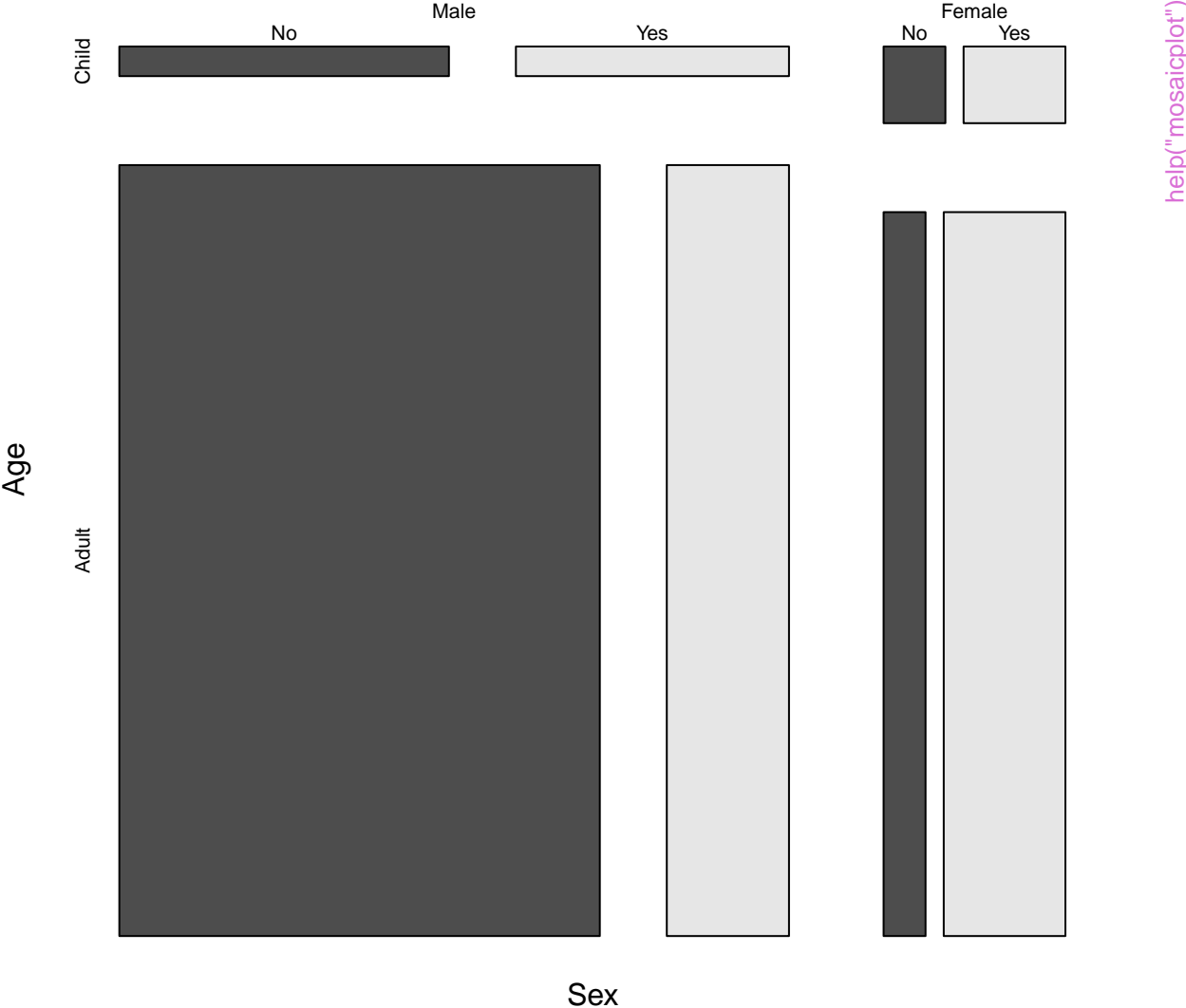


# Survival on the Titanic

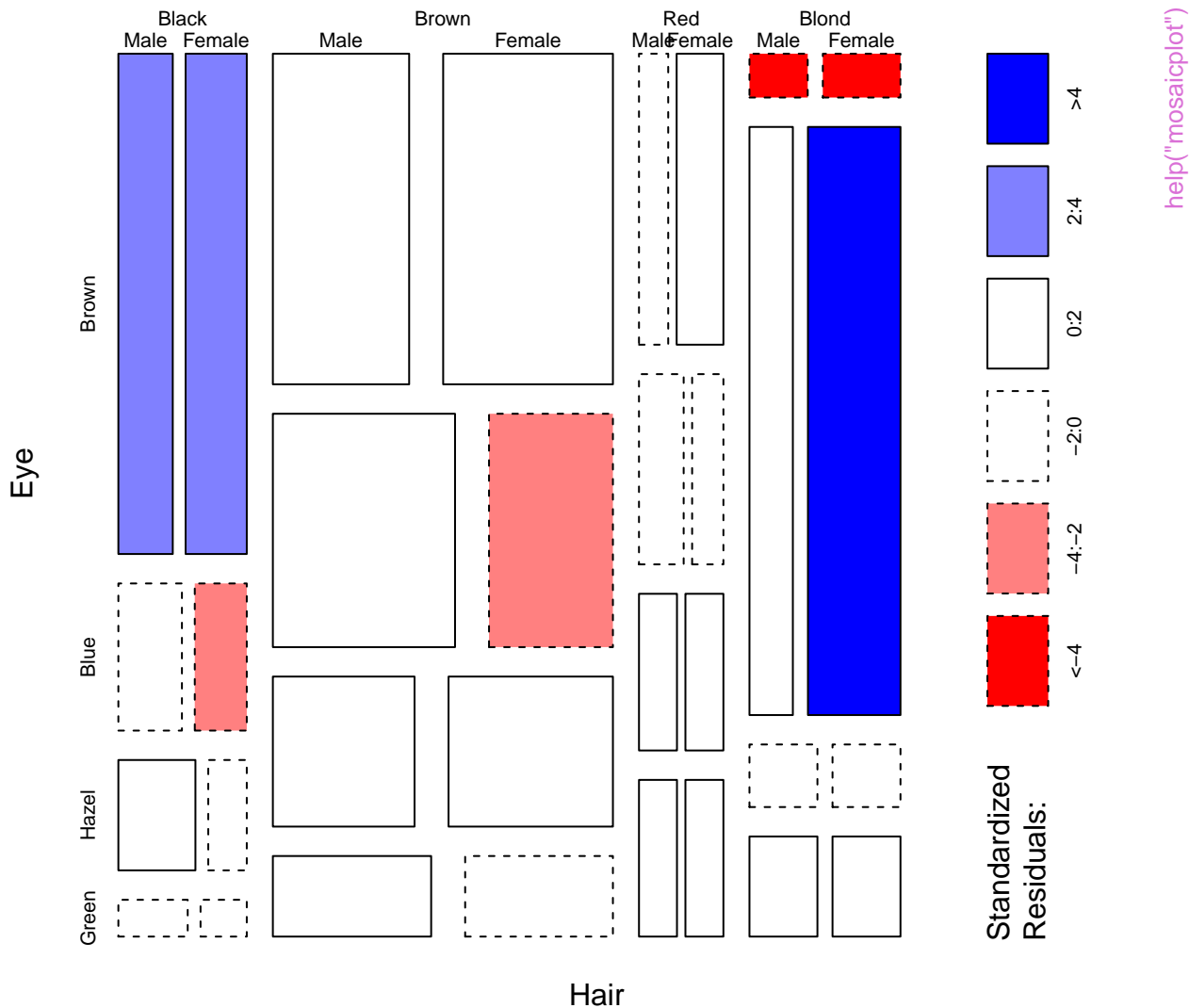




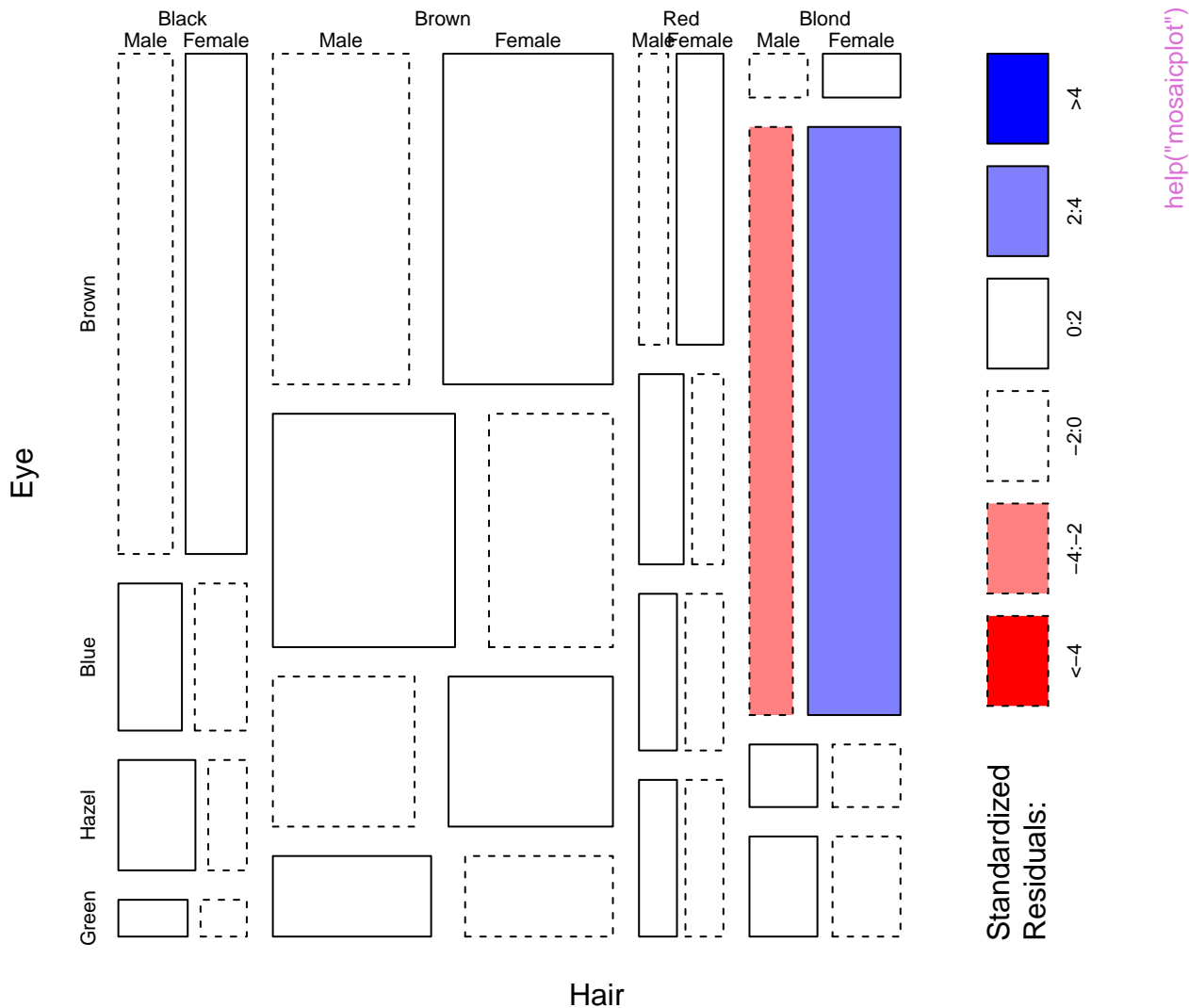
Titanic



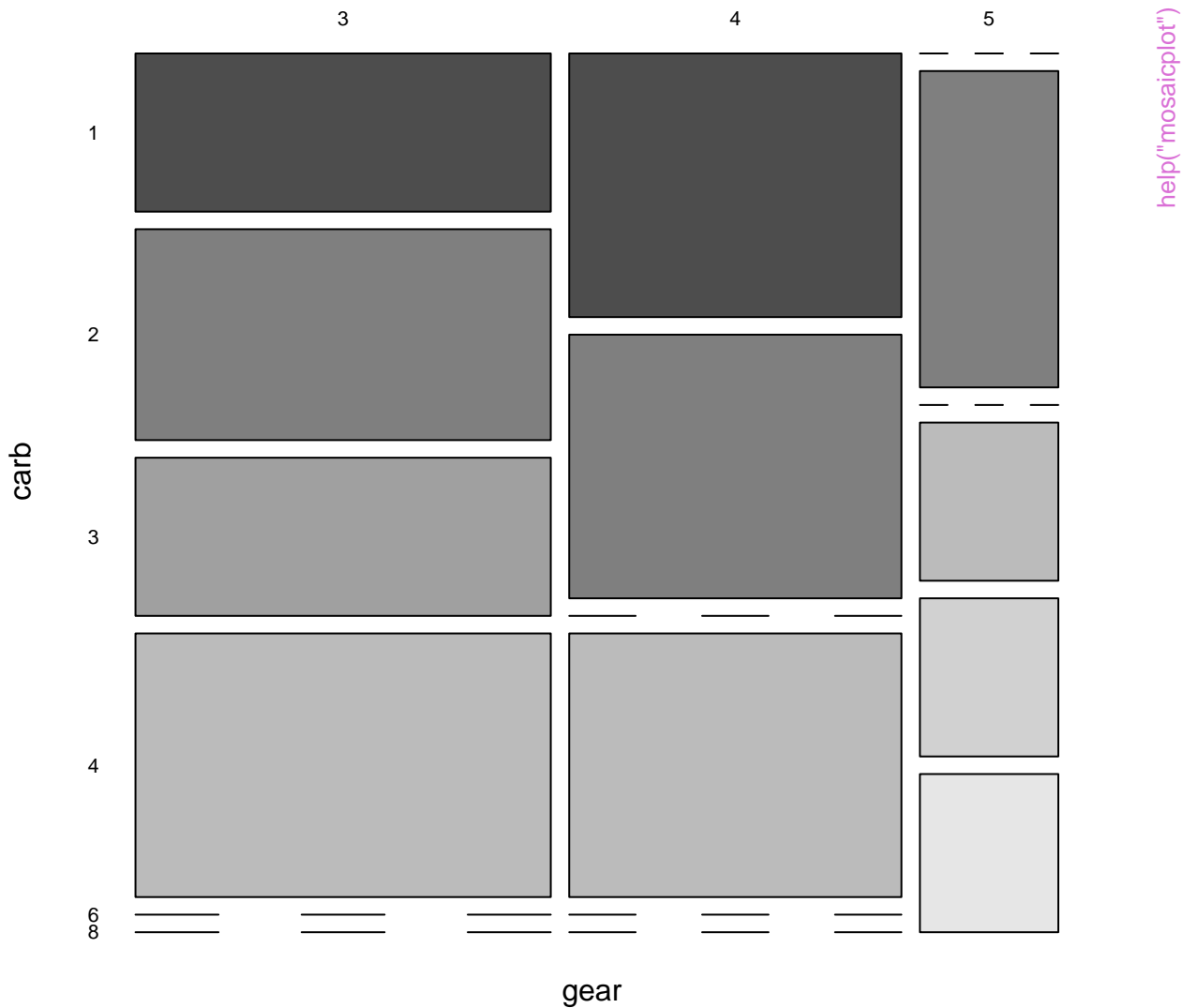
# HairEyeColor



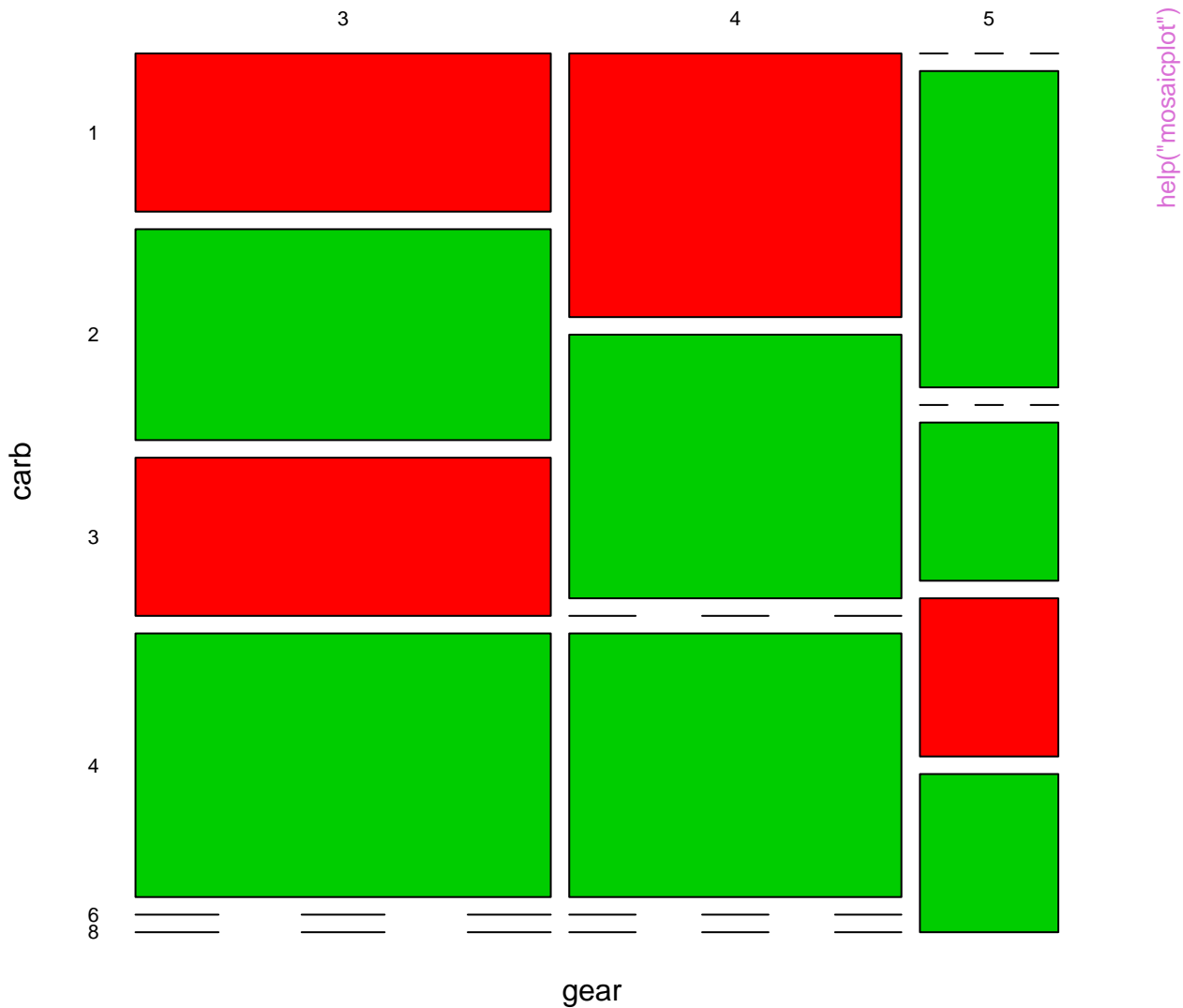
# HairEyeColor



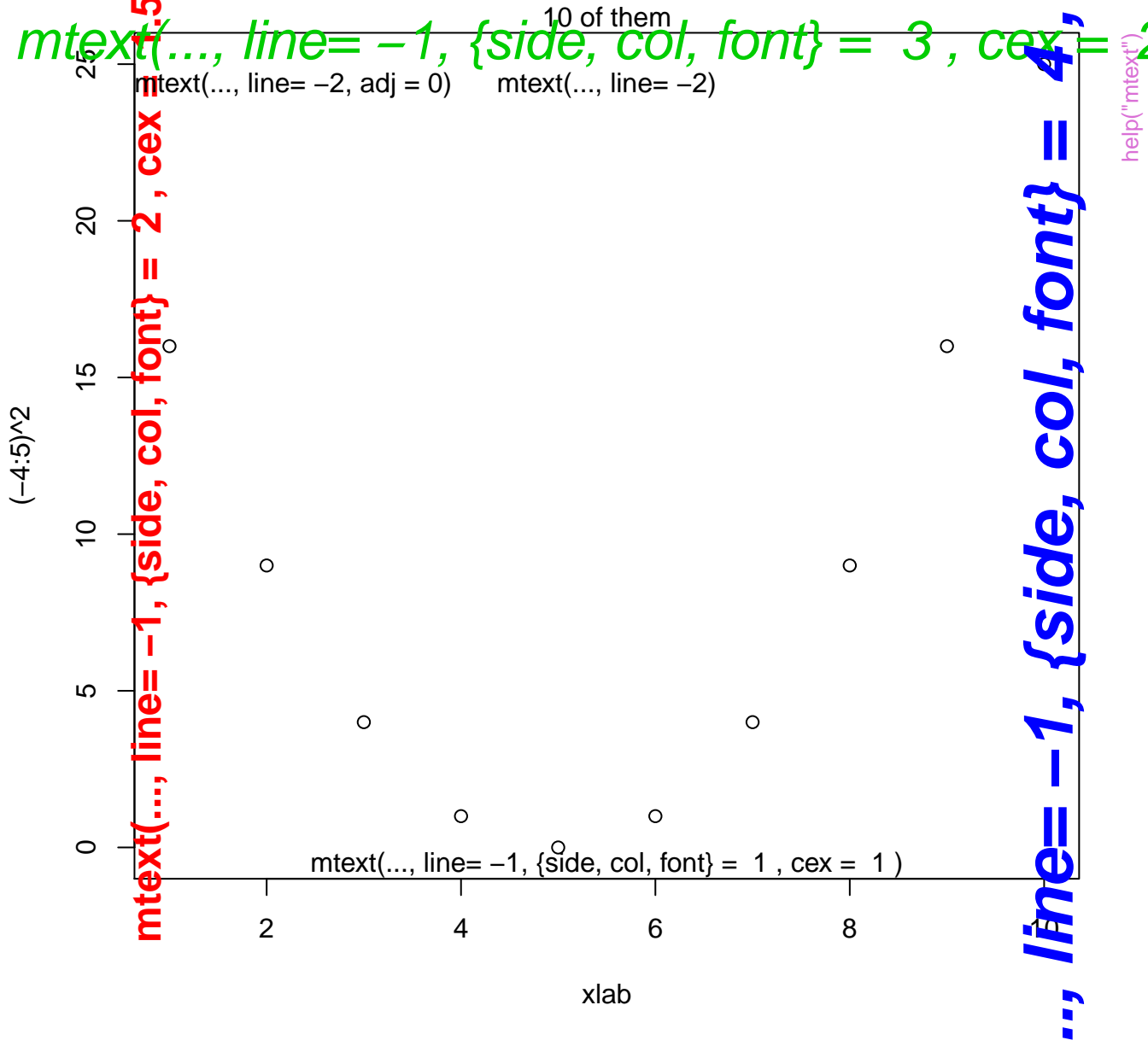
# mtcars



# mtcars

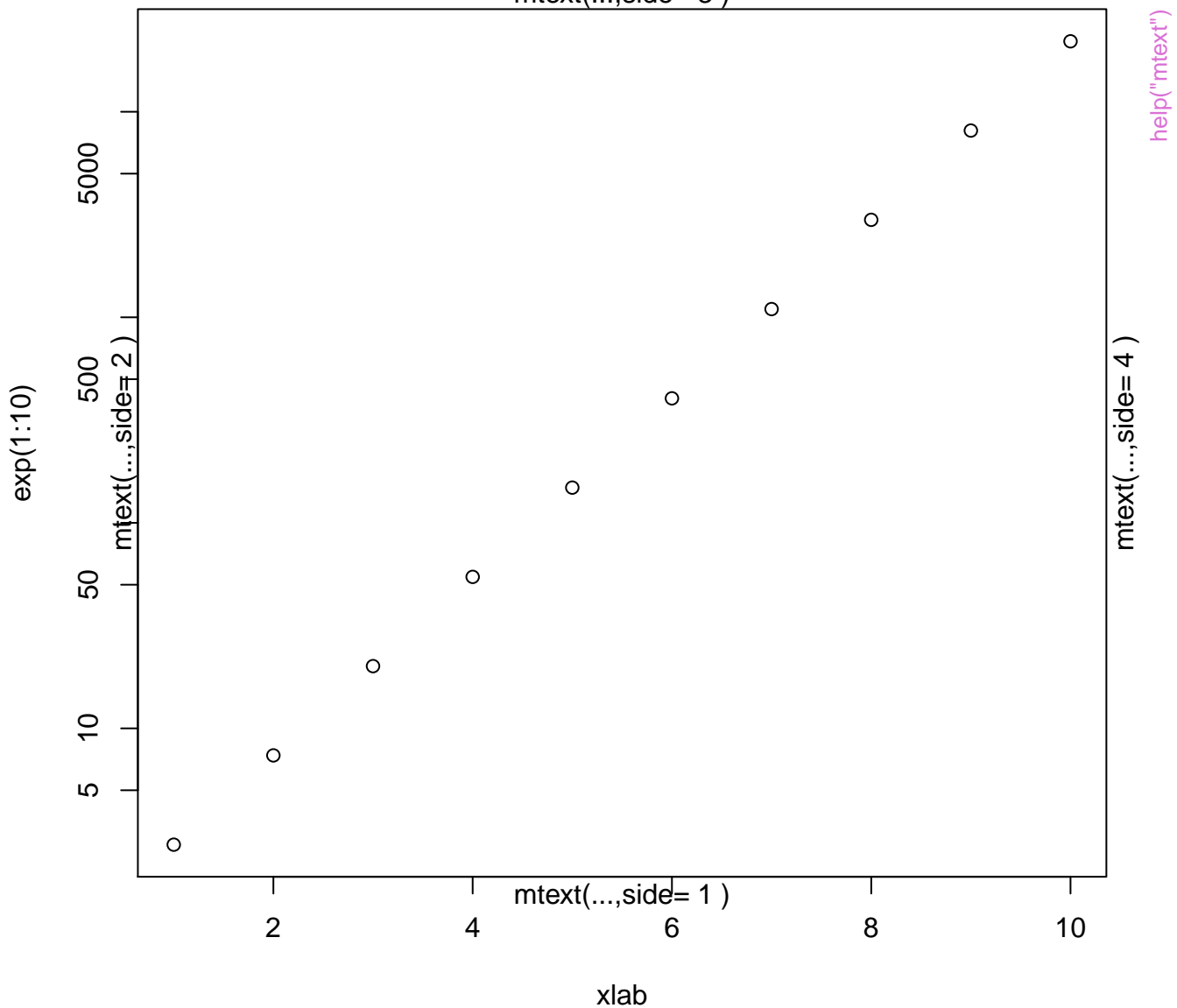


## Parabola Points

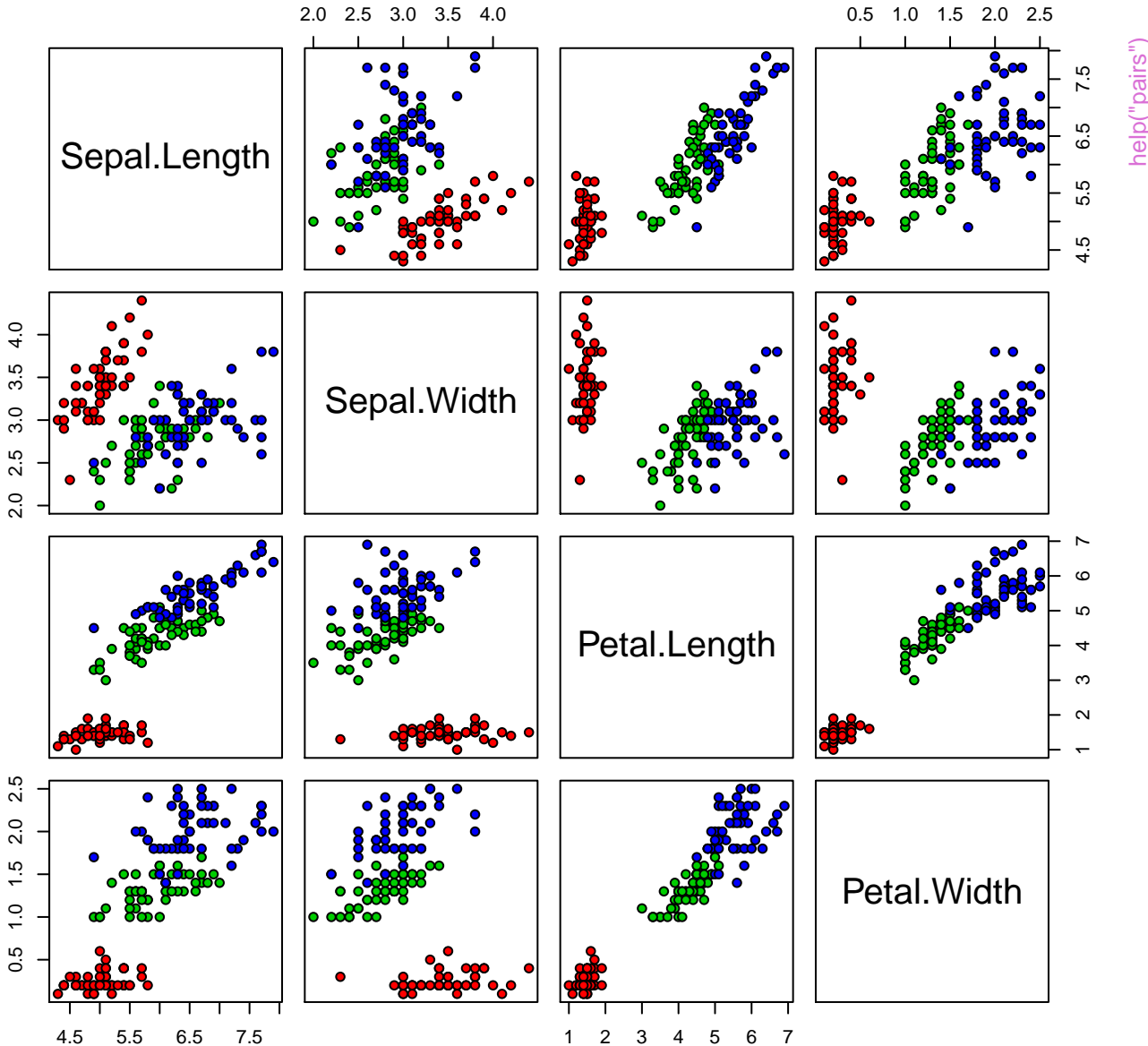


**log = "y"**

`mtext(...,side= 3 )`

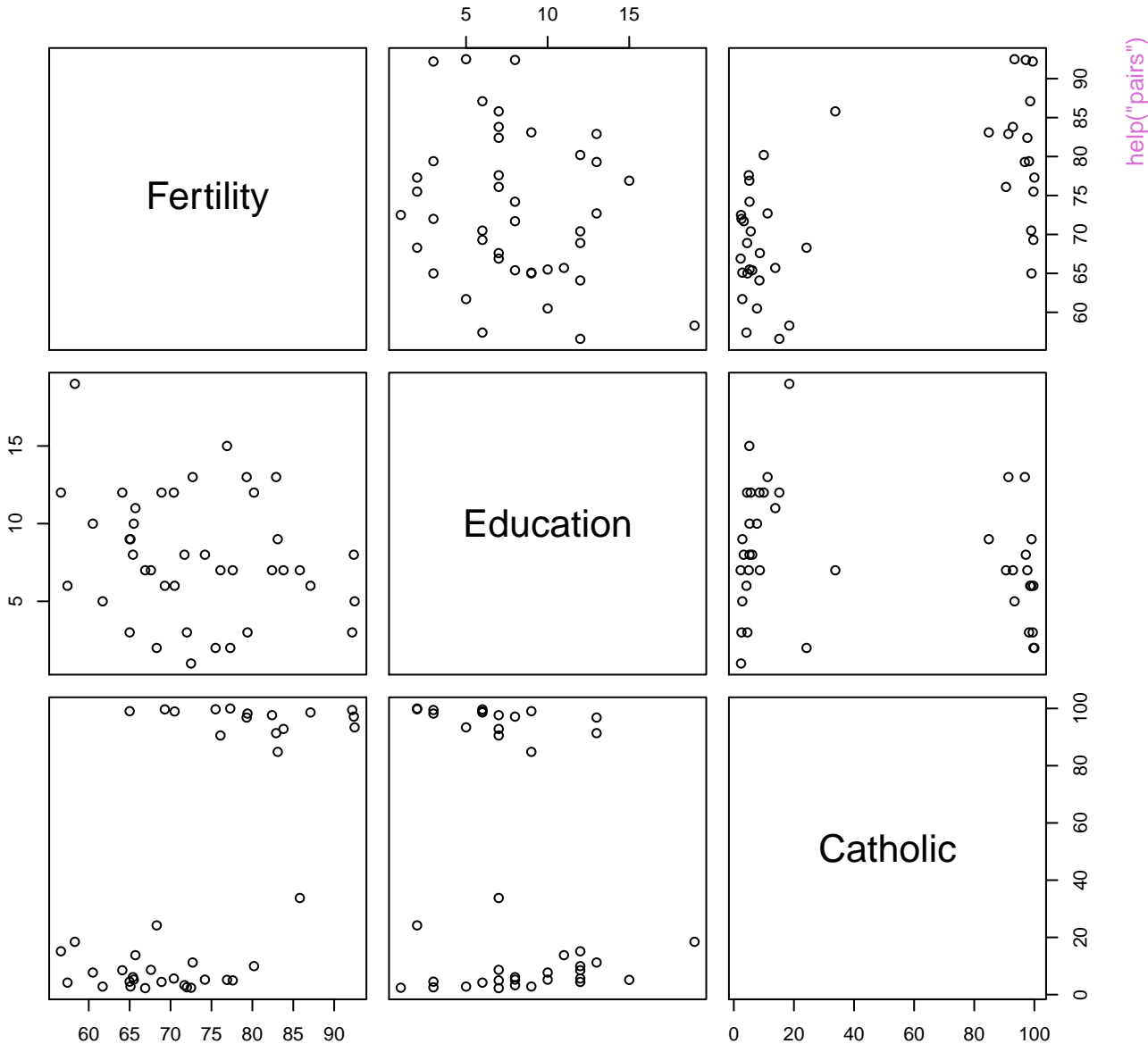


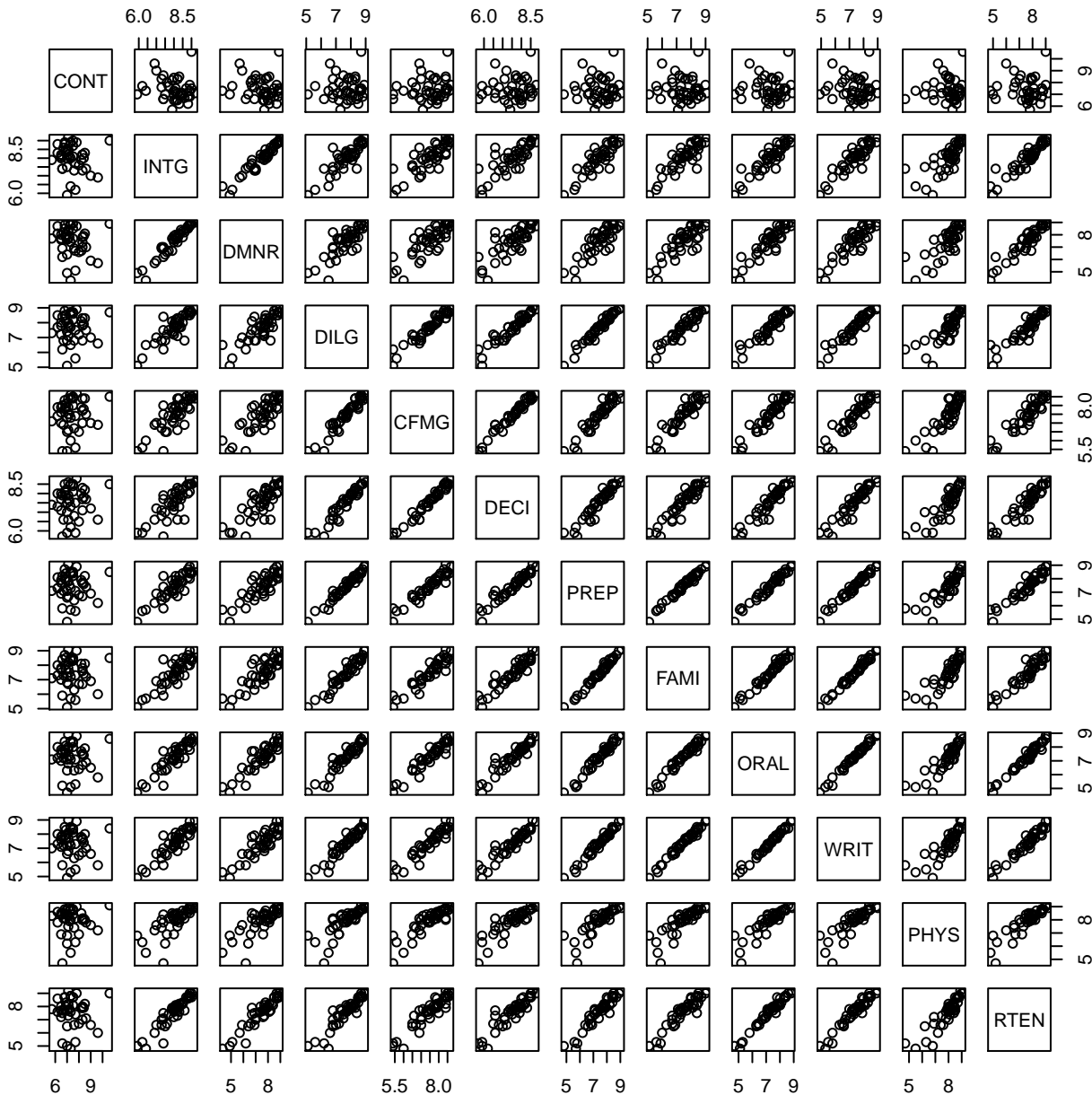
# Anderson's Iris Data -- 3 species



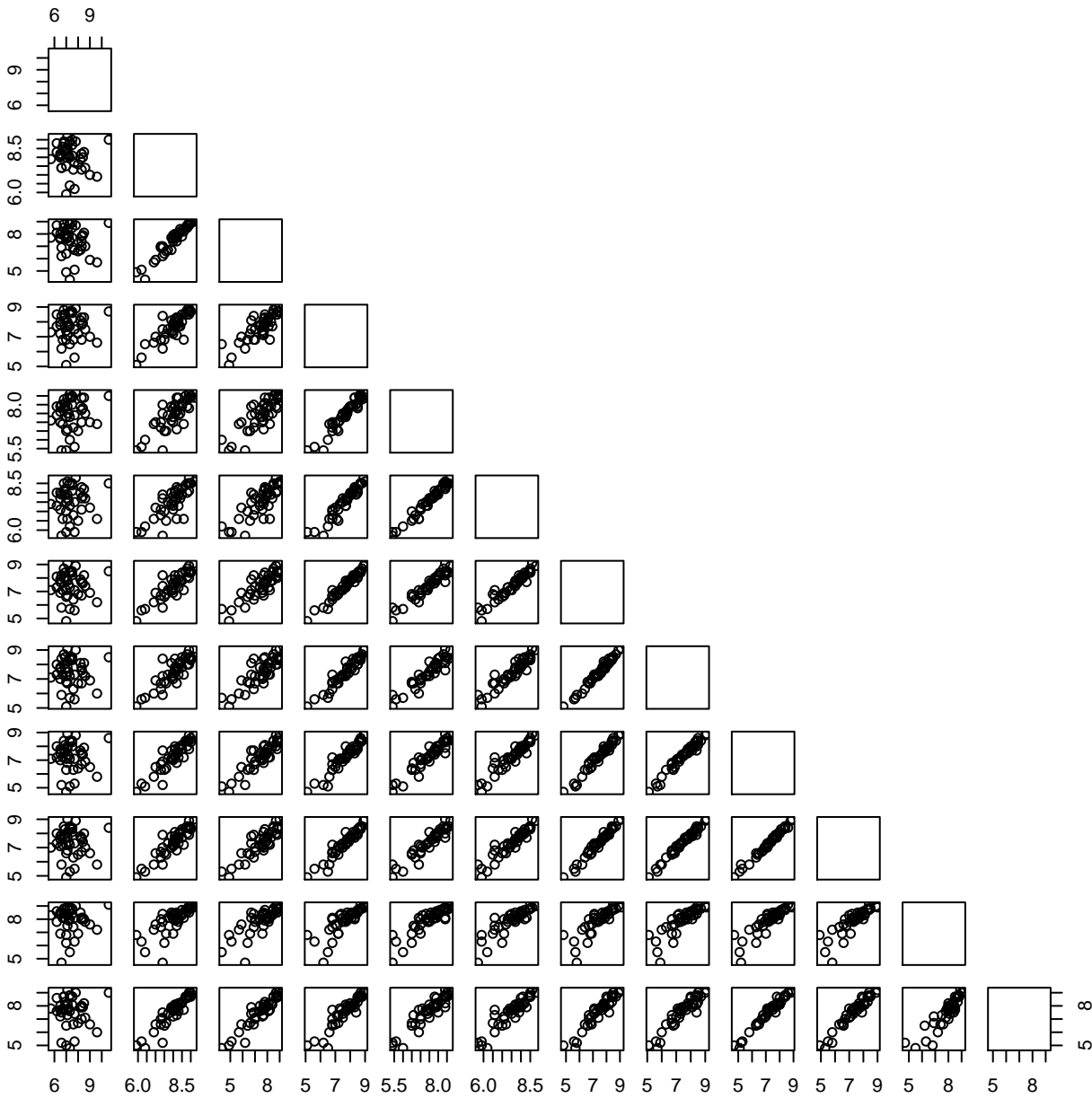


# Swiss data, Education < 20

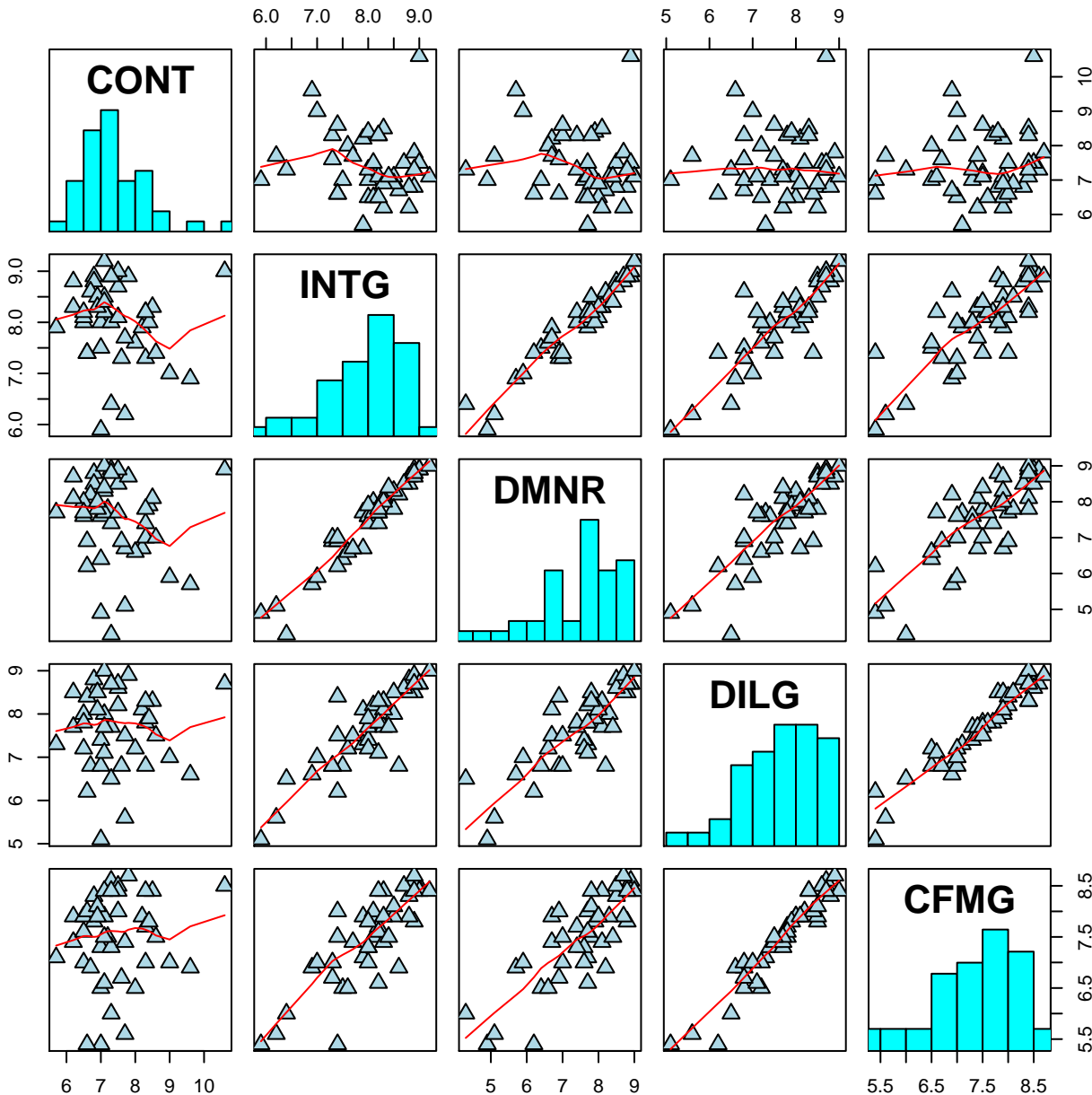




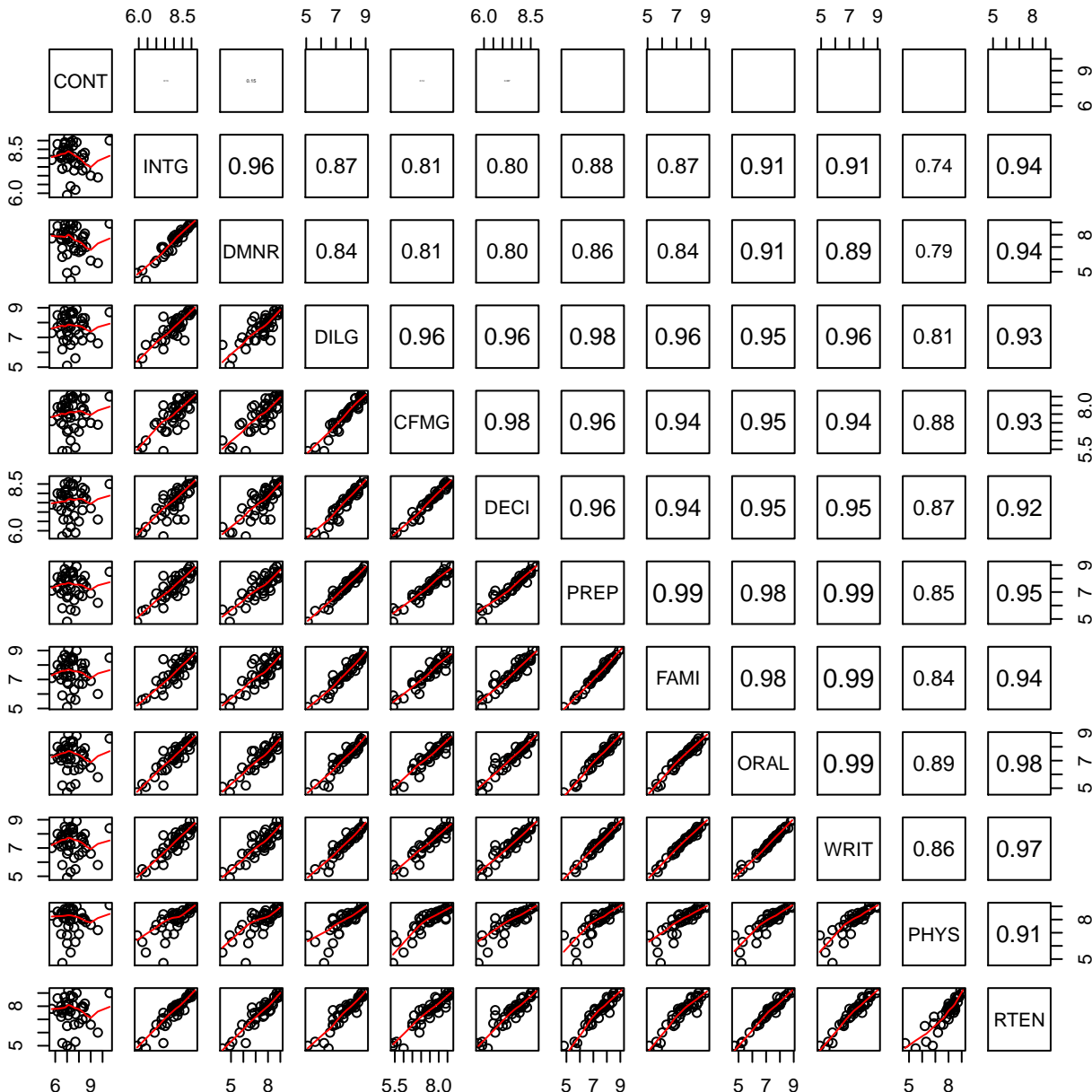
help("pairs")



help("pairs")

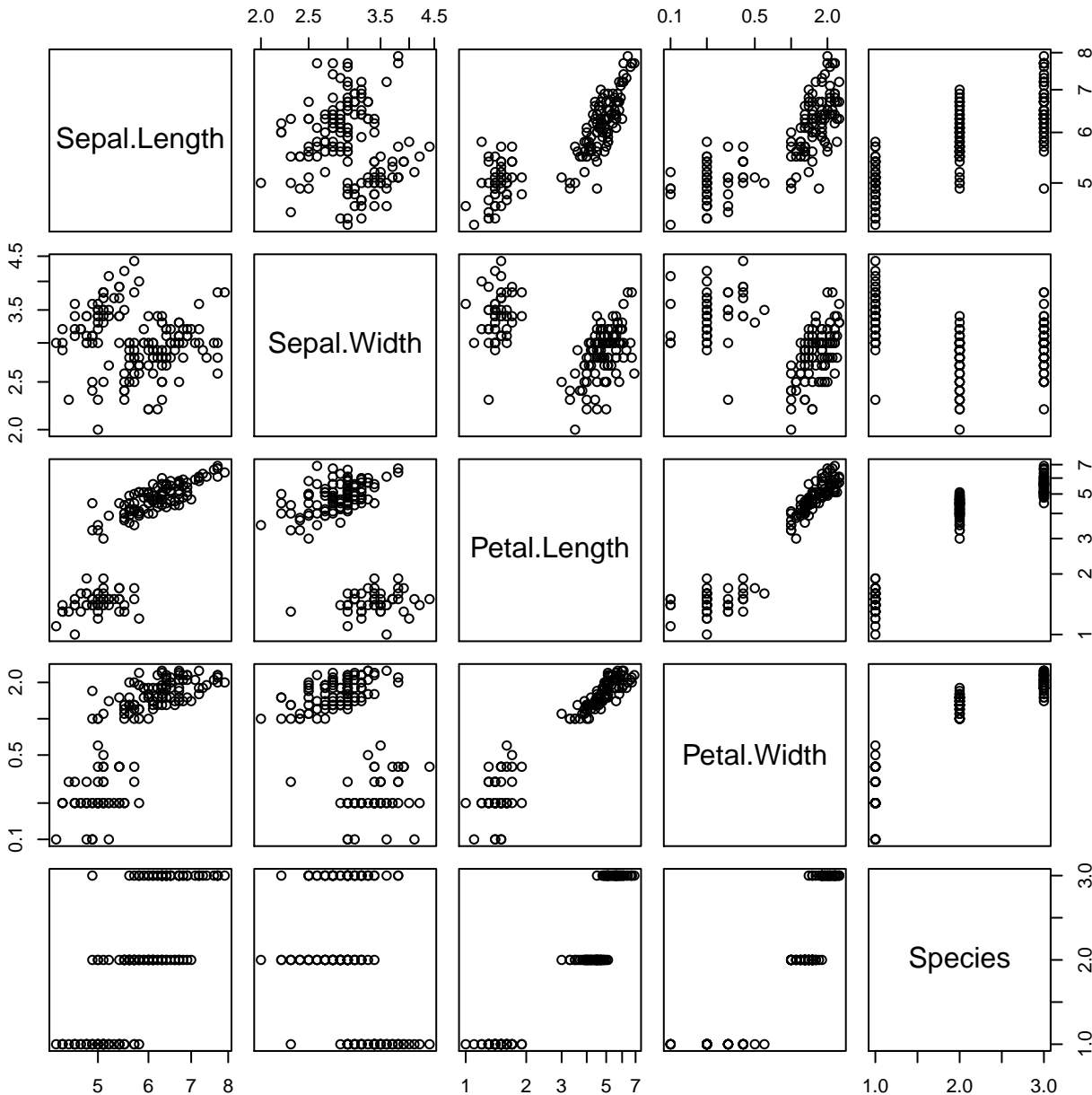


[help\("pairs"\)](#)

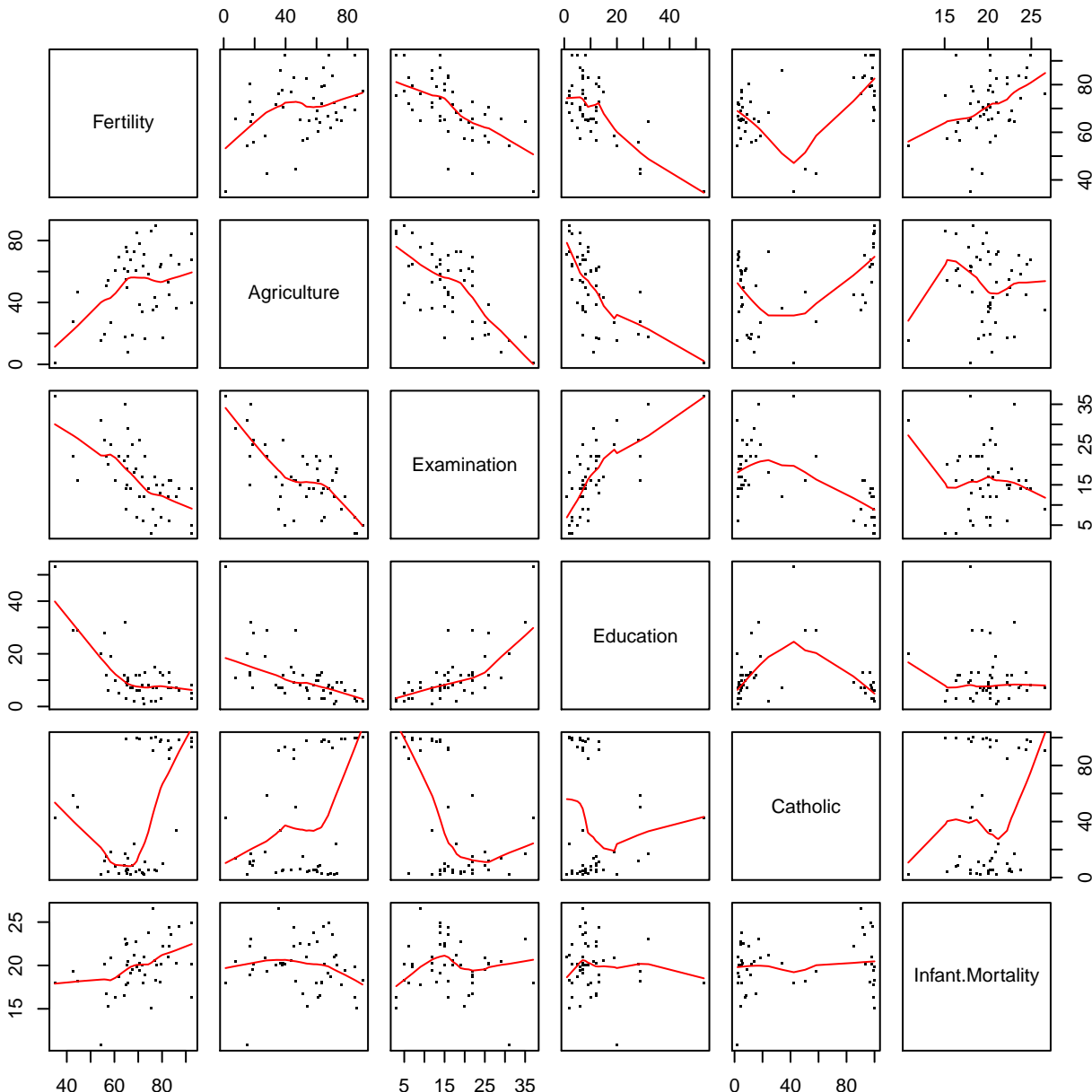


help("pairs")



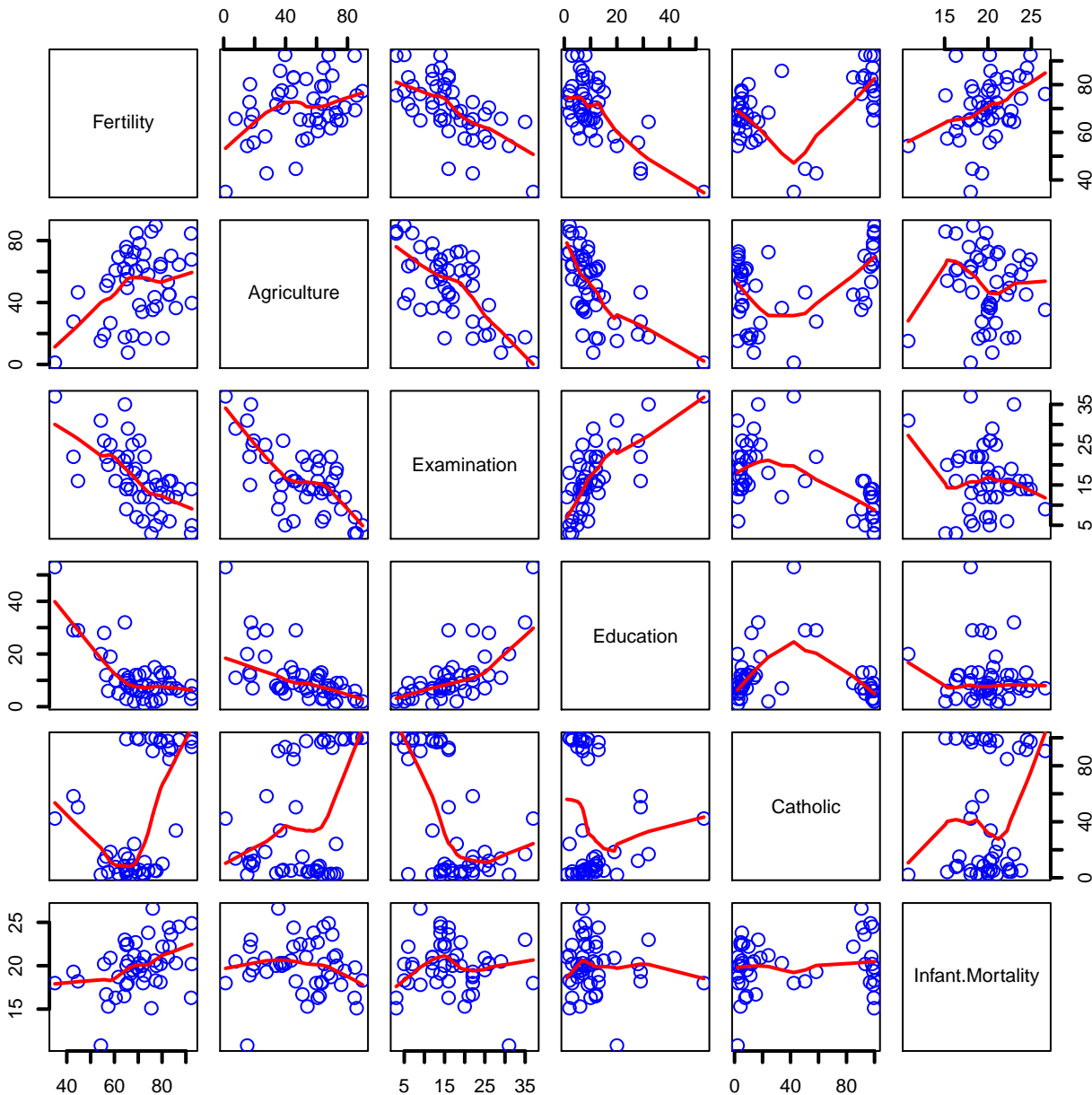


`help("pairs")`

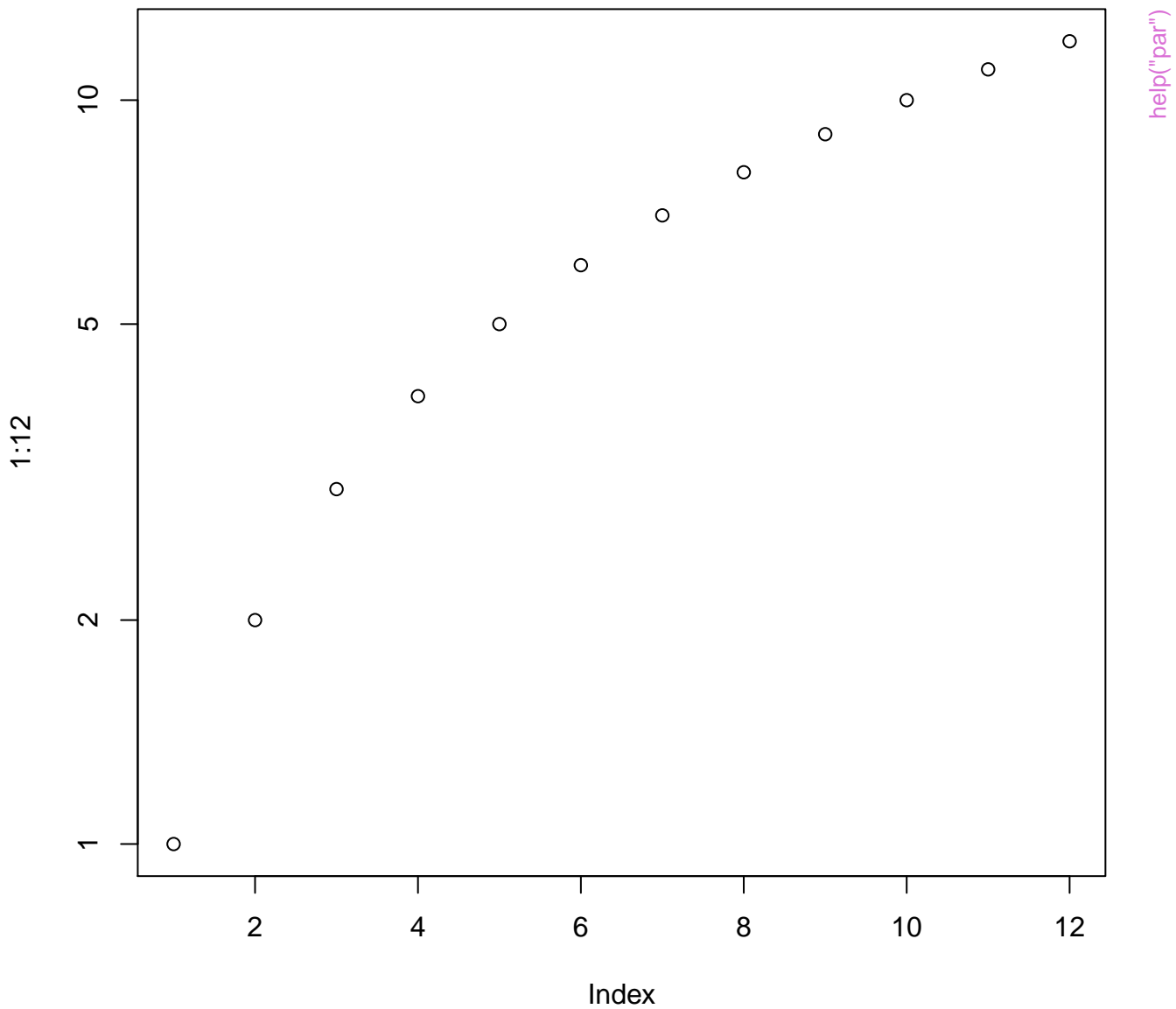


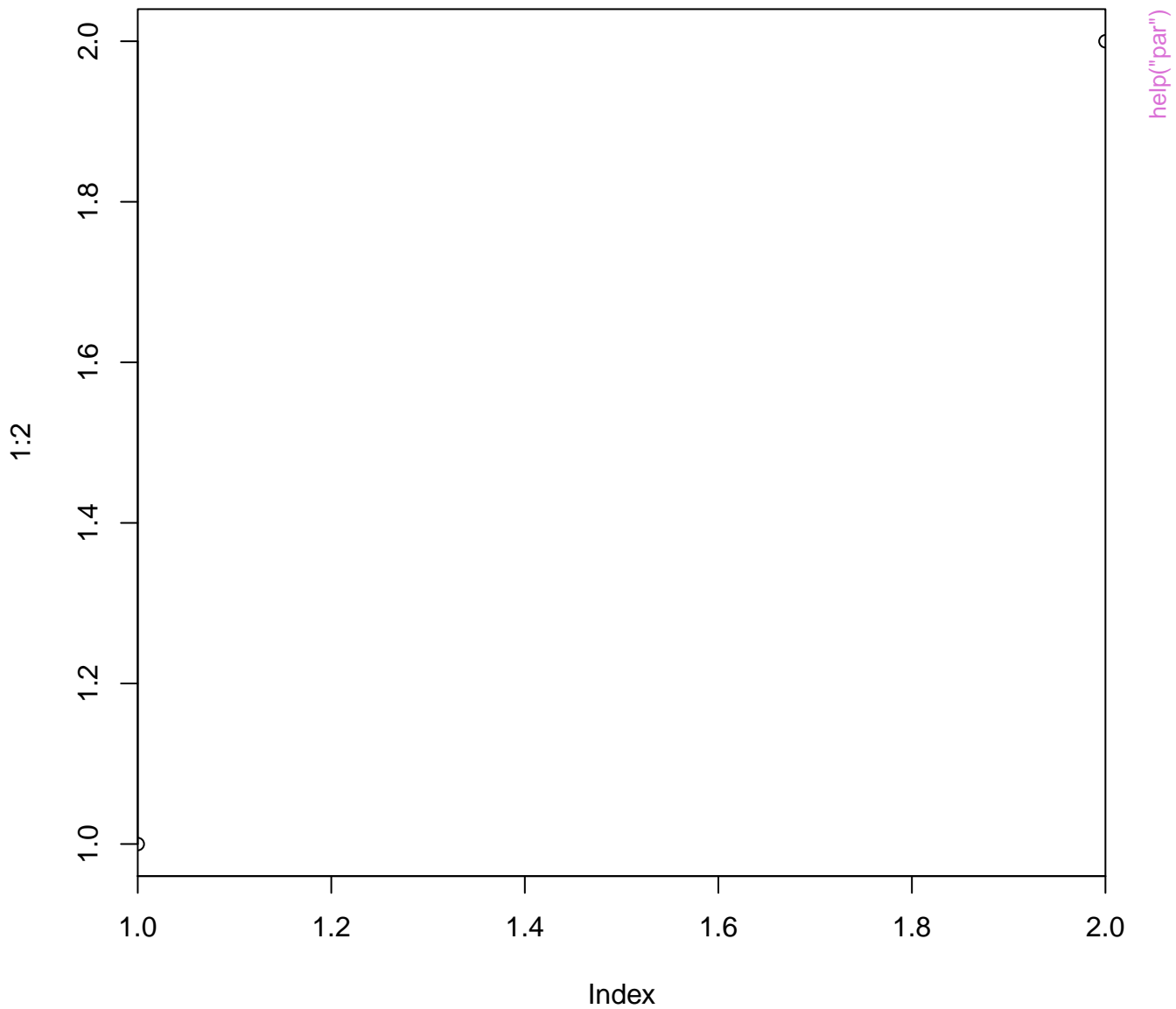
help("panel.smooth")



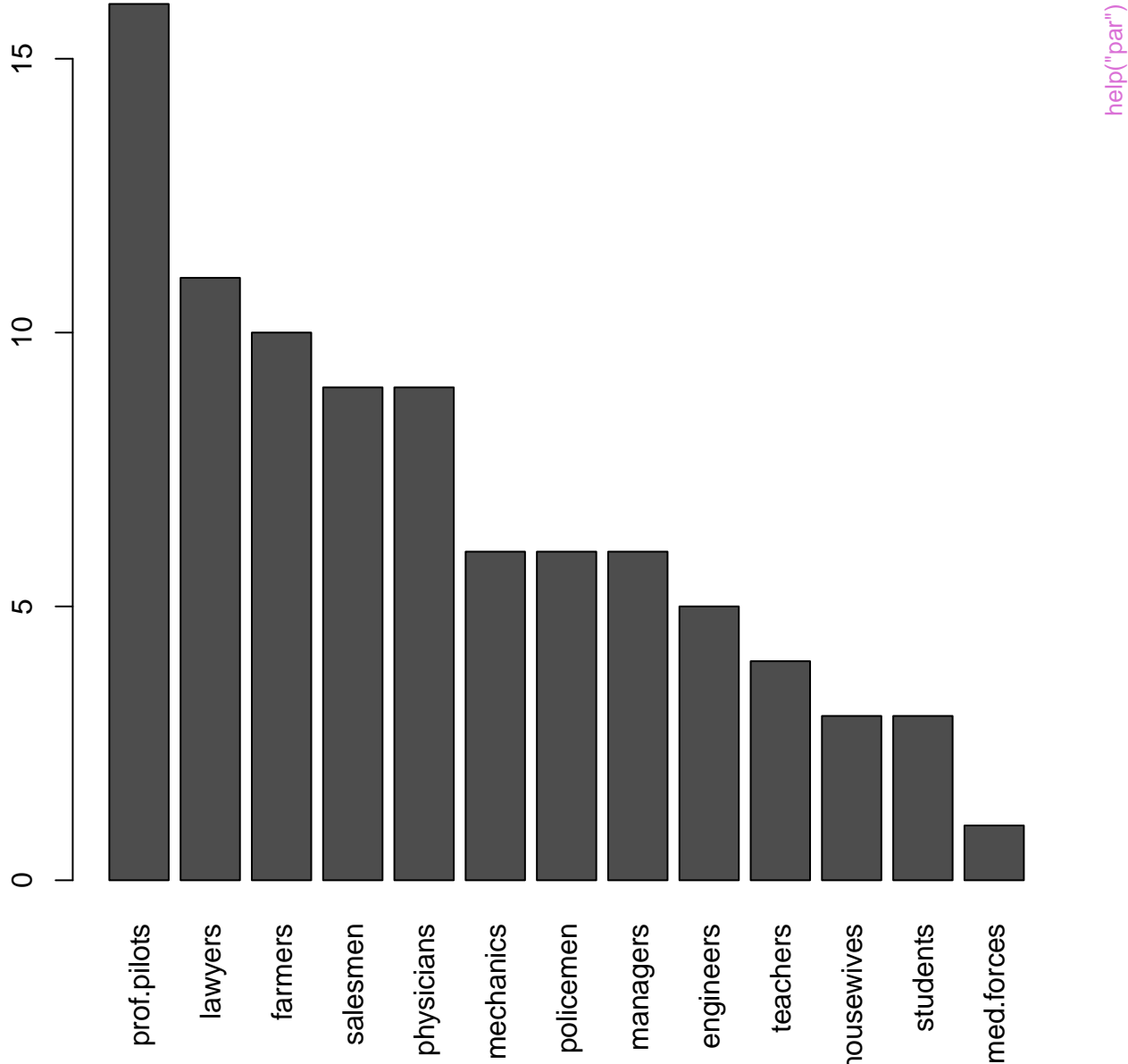


help("panel.smooth")

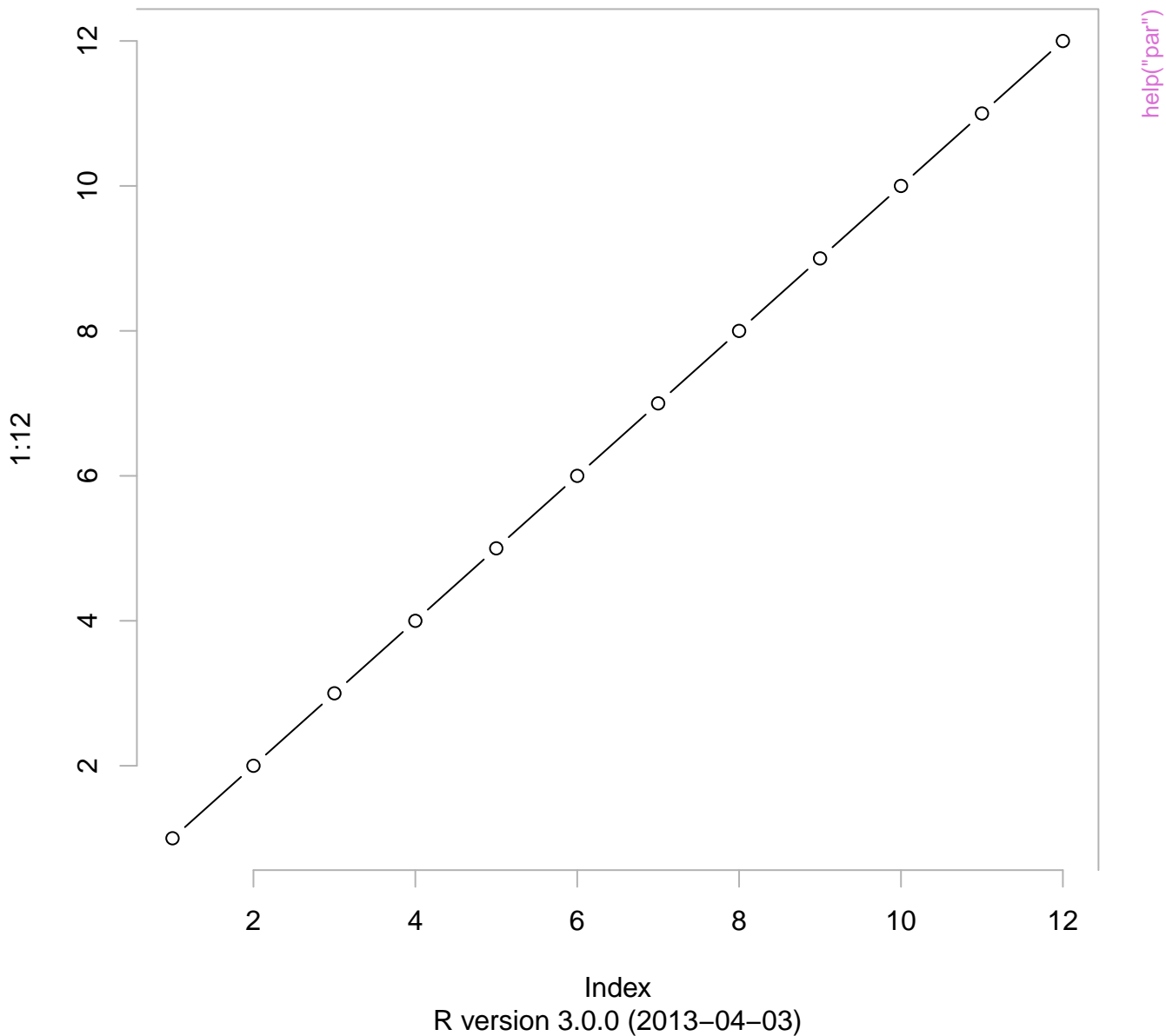




help("par")



**'fg' : axes, ticks and box in gray**



solid                      solid

solid lwd = 2              solid lwd = 2

dashed                      44

dashed lwd = 2              44 lwd = 2

dotted                      13

dotted lwd = 2              13 lwd = 2

dotdash                      1343

dotdash lwd = 2              1343 lwd = 2

longdash                      73

longdash lwd = 2              73 lwd = 2

twodash                      2262

twodash lwd = 2              2262 lwd = 2

11

11 lwd = 2

22

22 lwd = 2

33

33 lwd = 2

44

44 lwd = 2

12

12 lwd = 2

13

13 lwd = 2

14

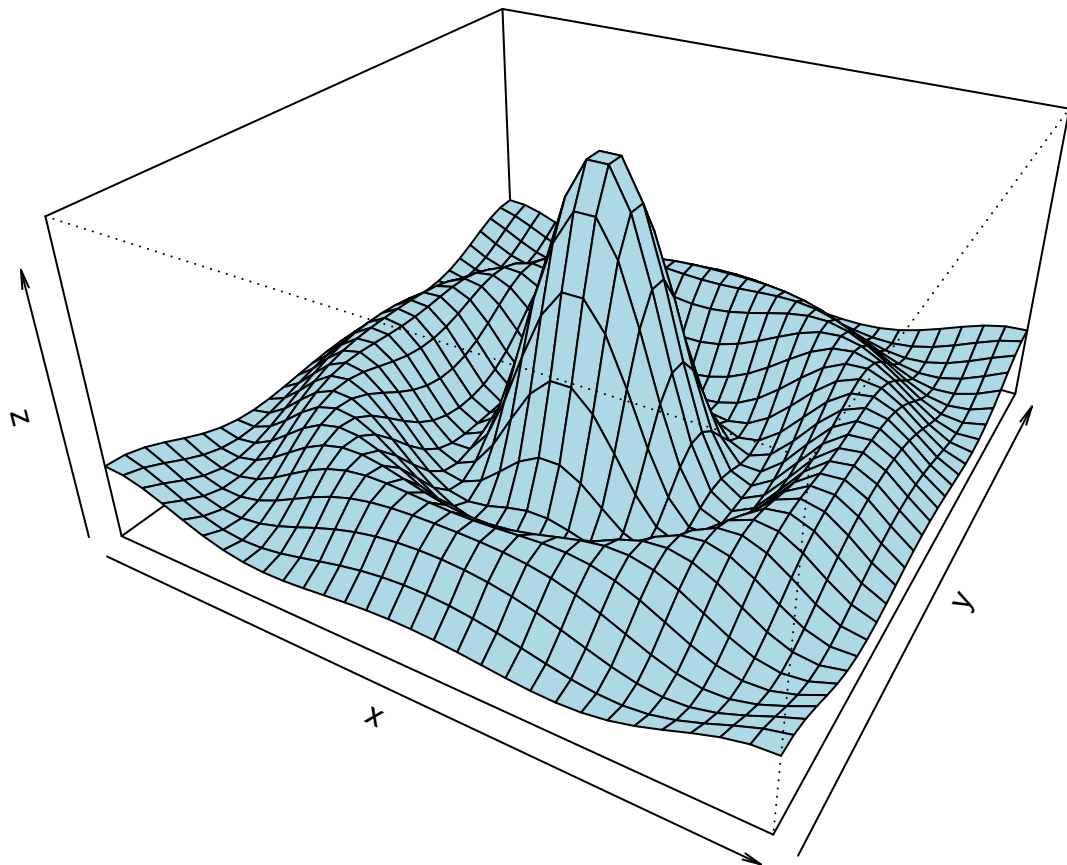
14 lwd = 2

21

21 lwd = 2

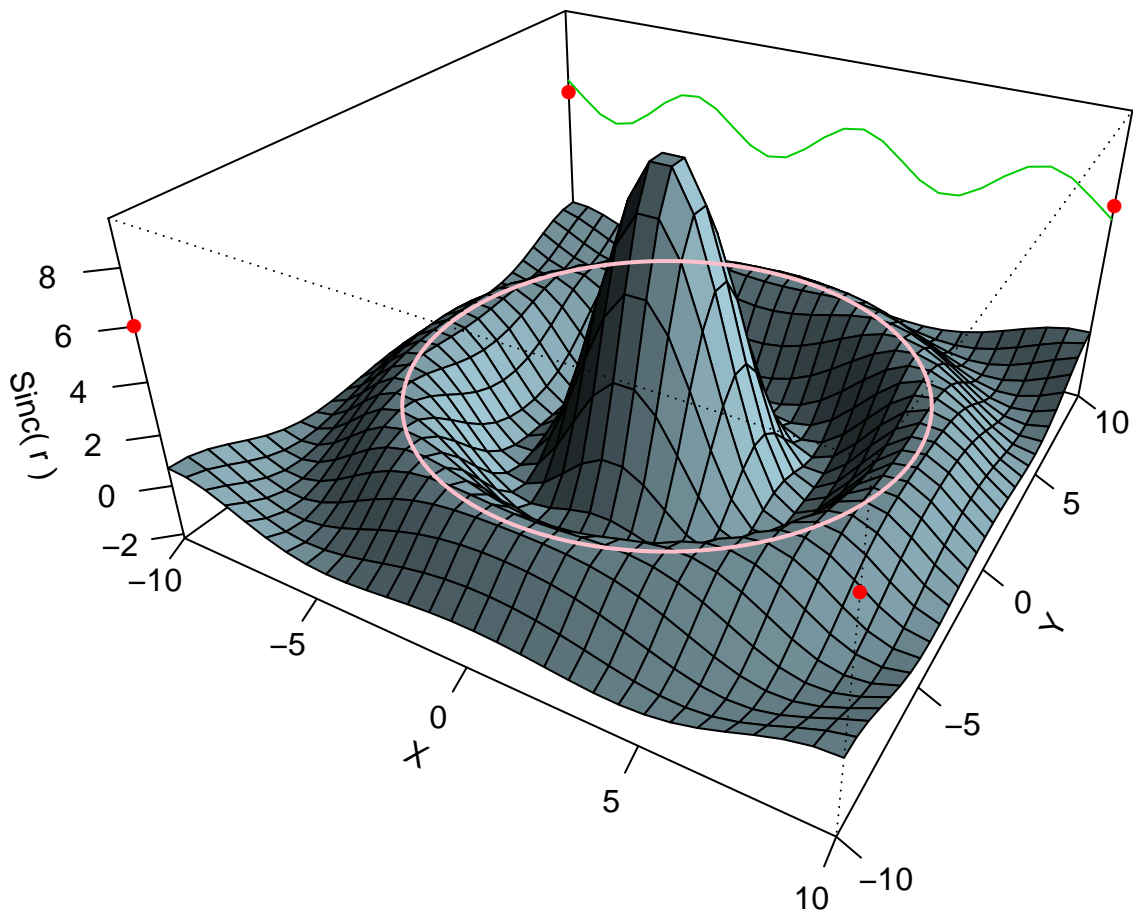
31

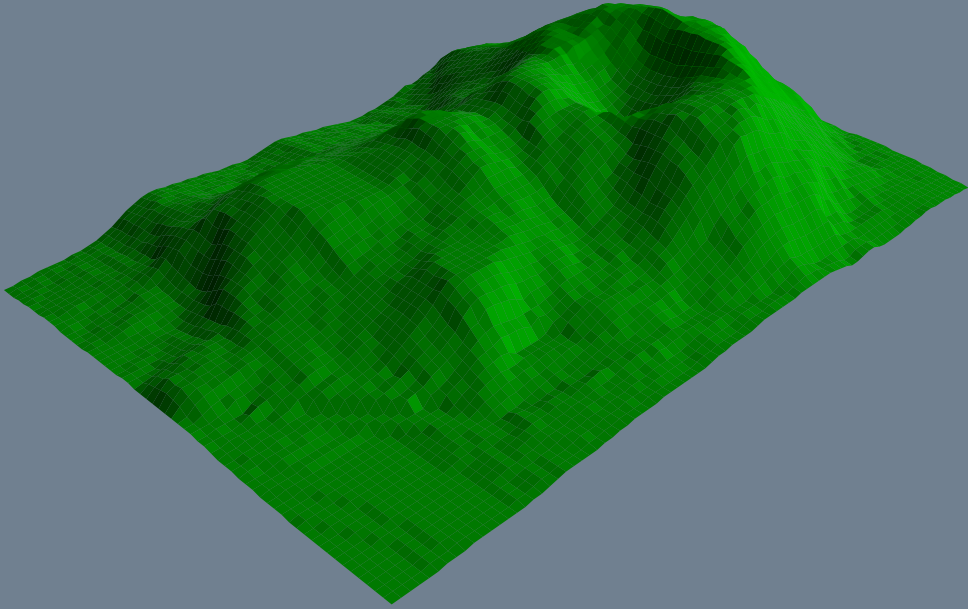
31 lwd = 2



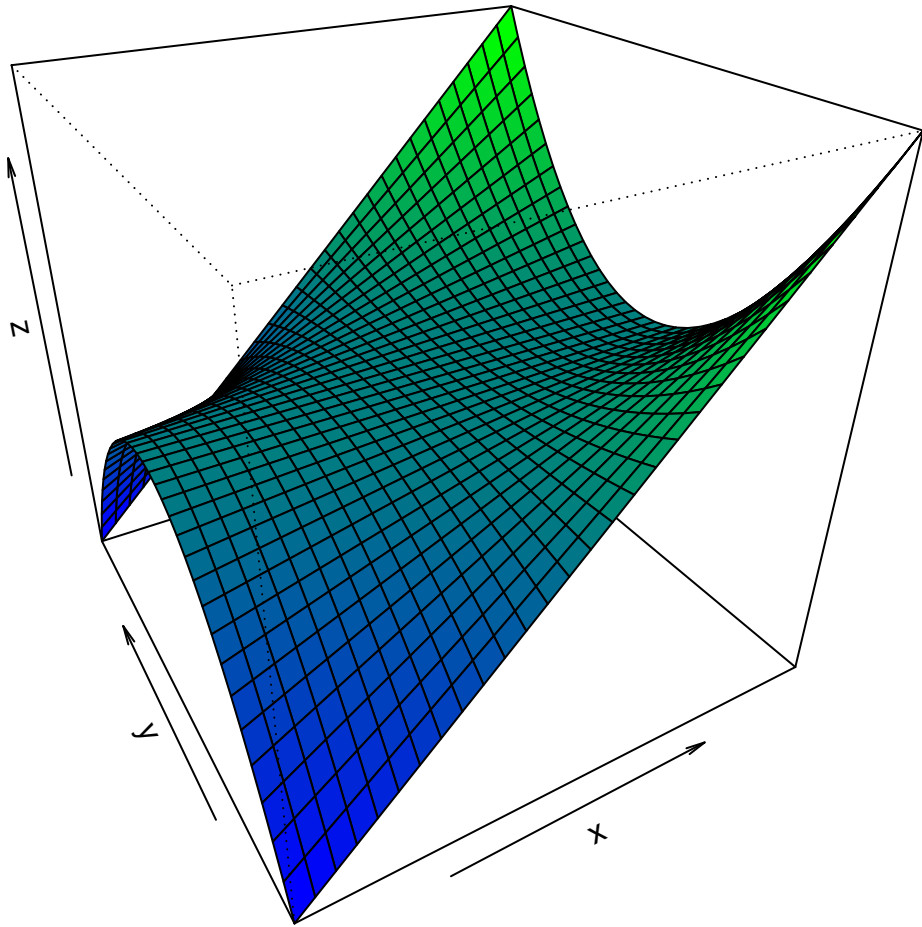
`help("persp")`



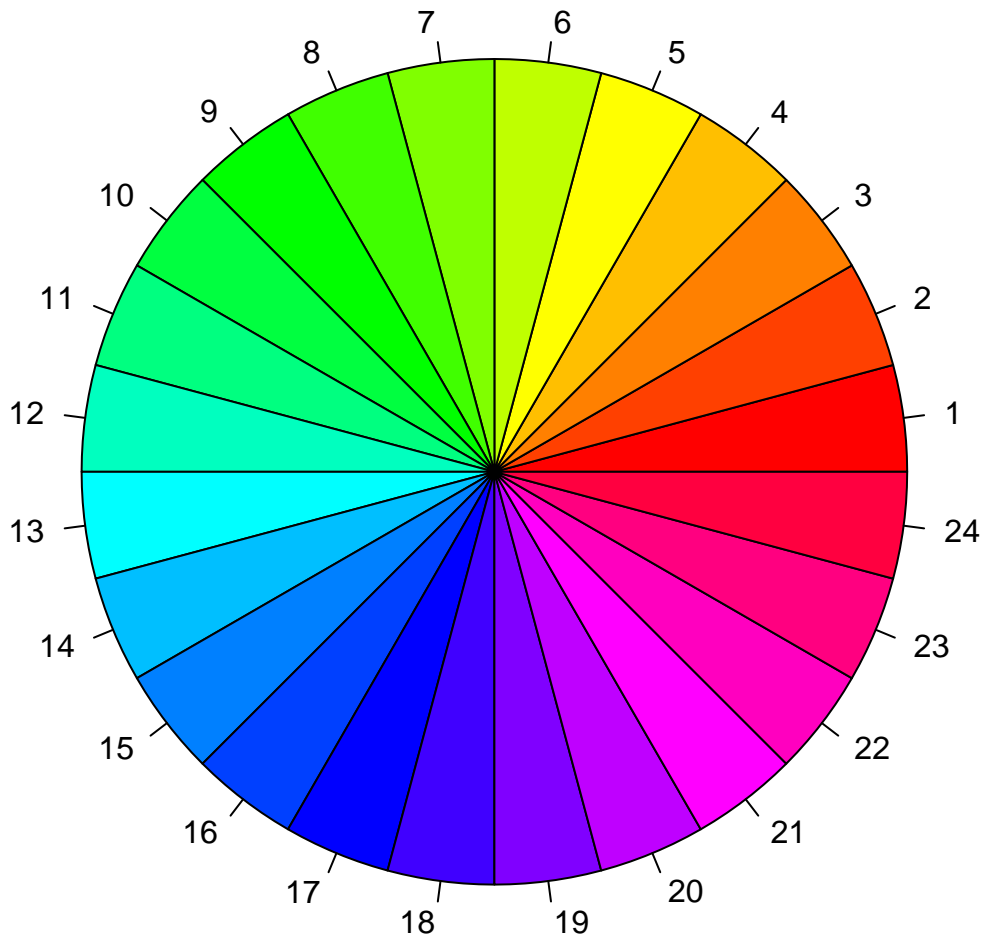




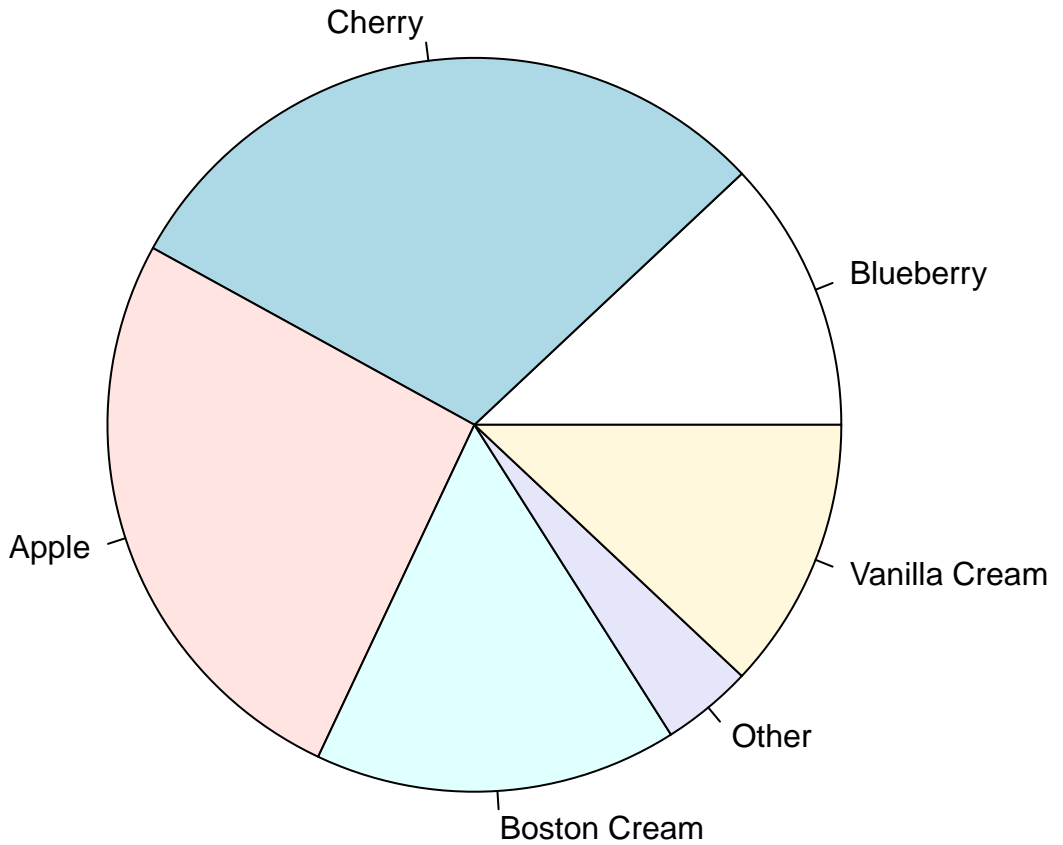
help("persp")

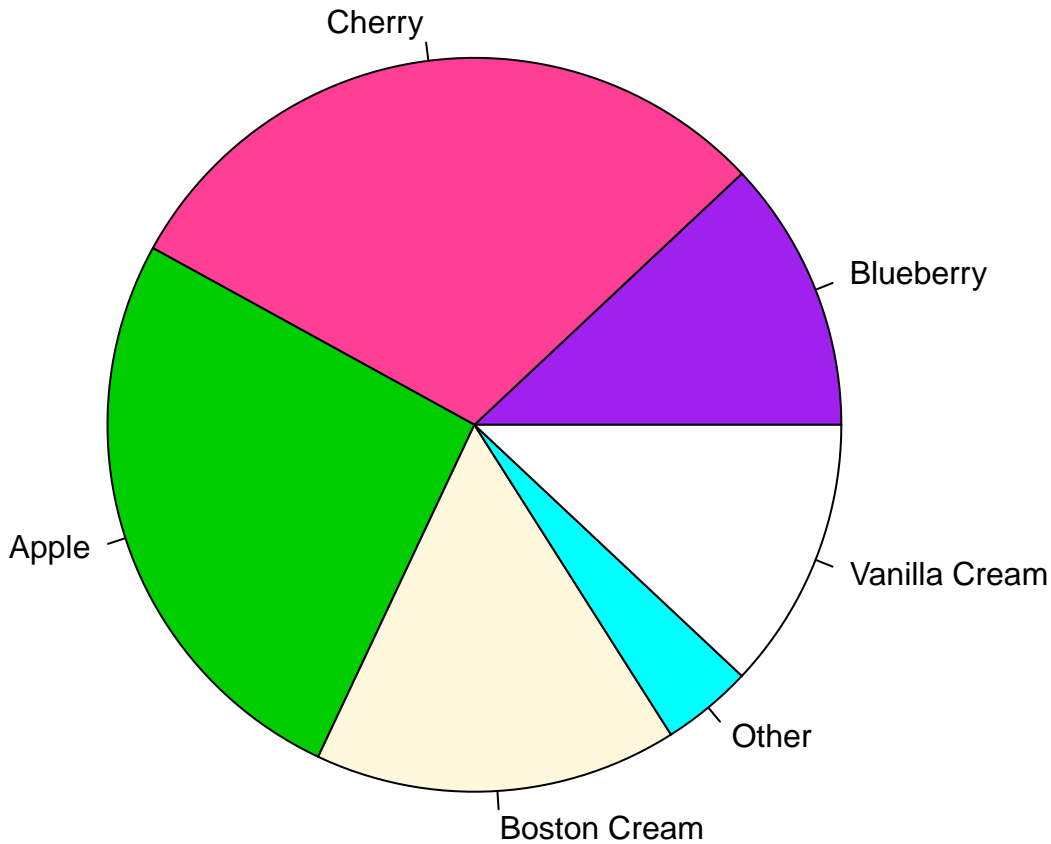


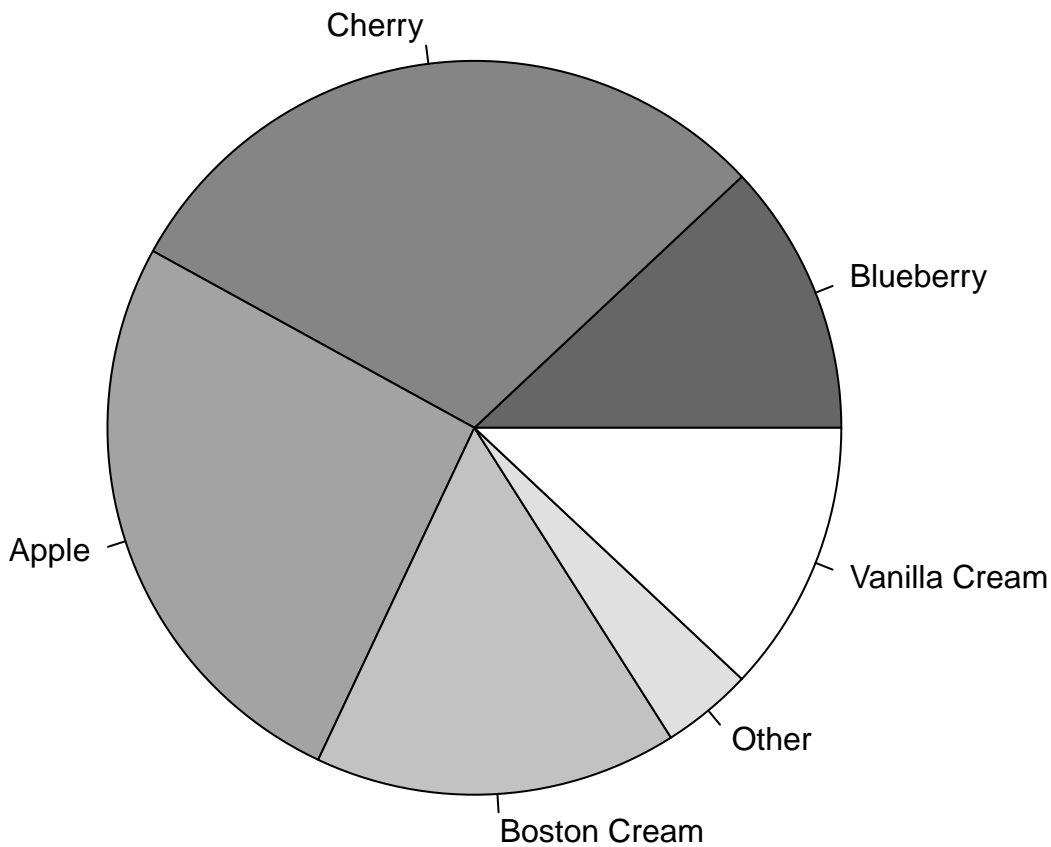
`help("persp")`

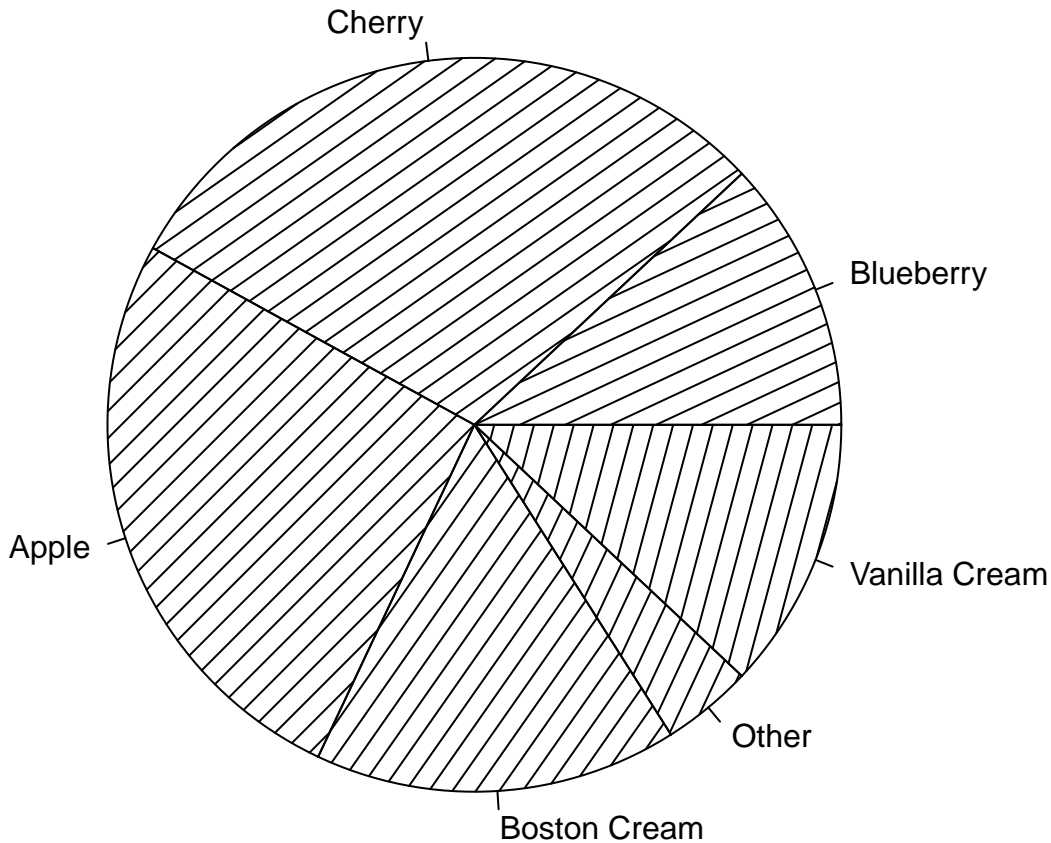


help("pie")





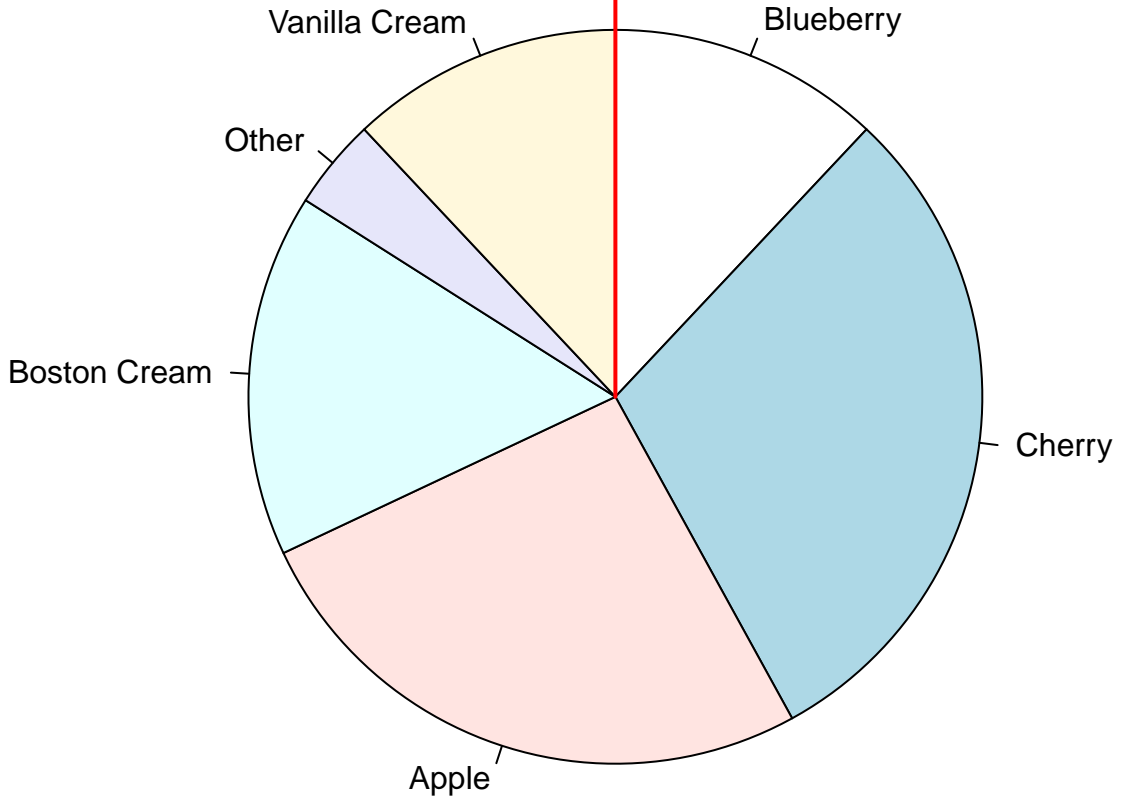






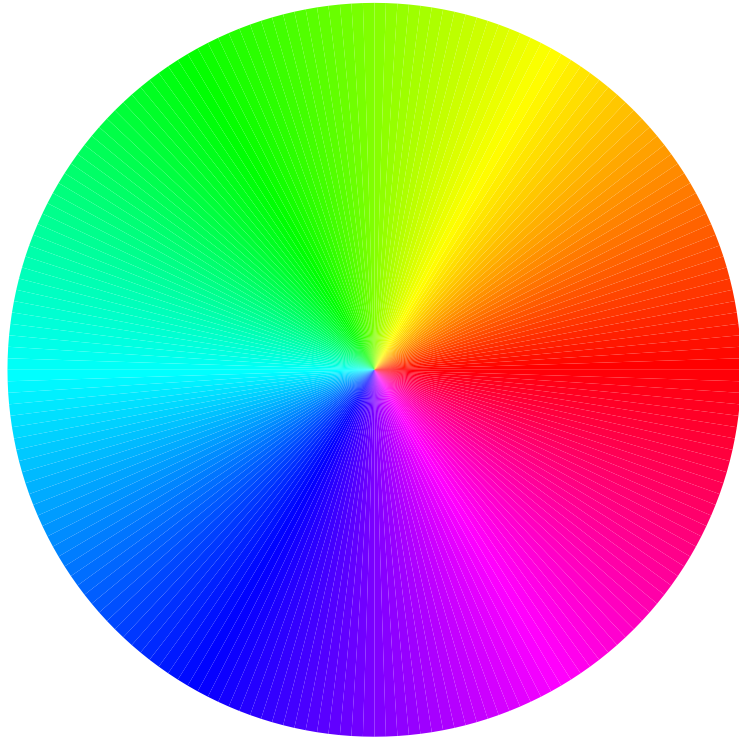
**pie(\*, clockwise = TRUE)**

init.angle = 90

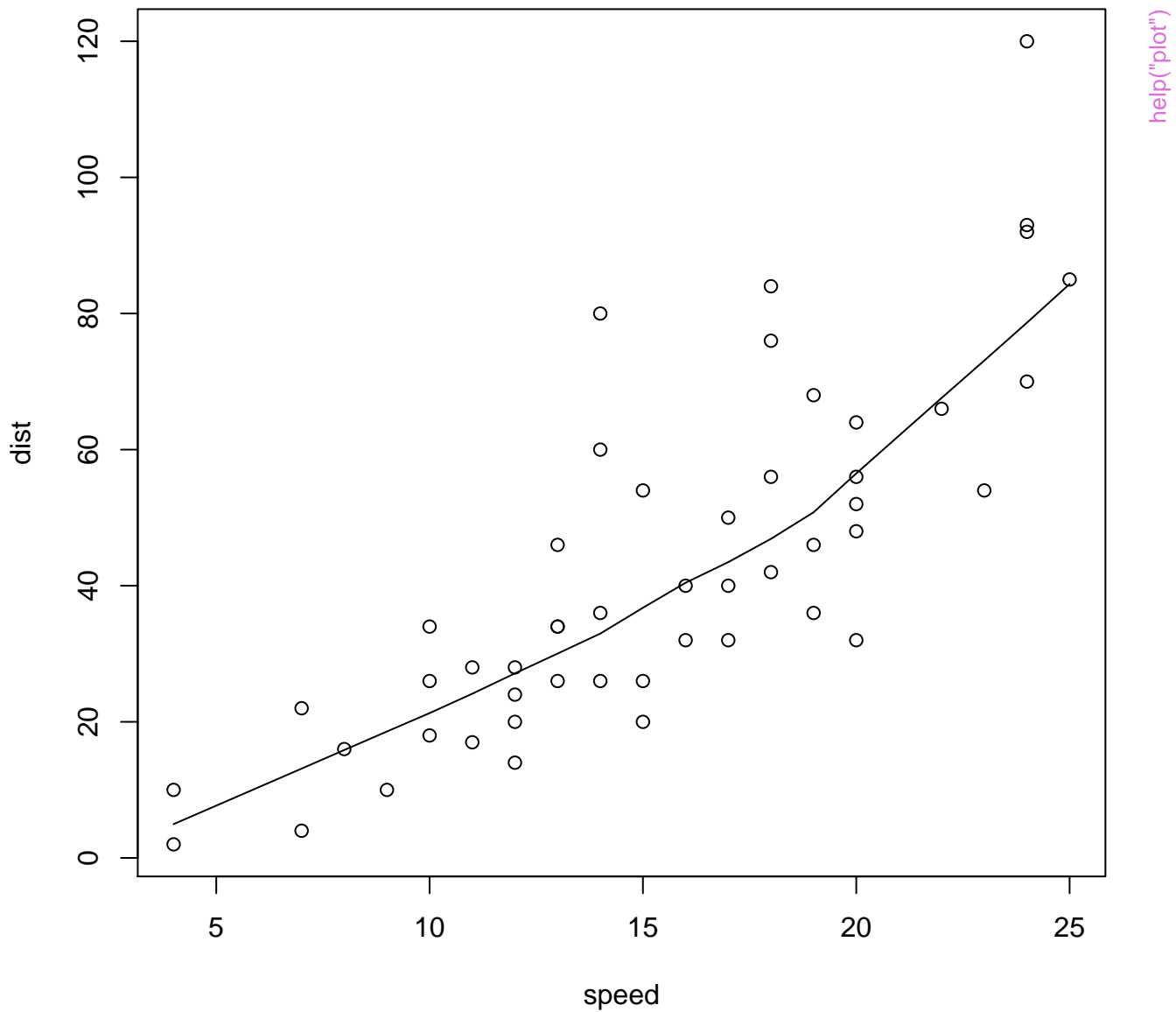


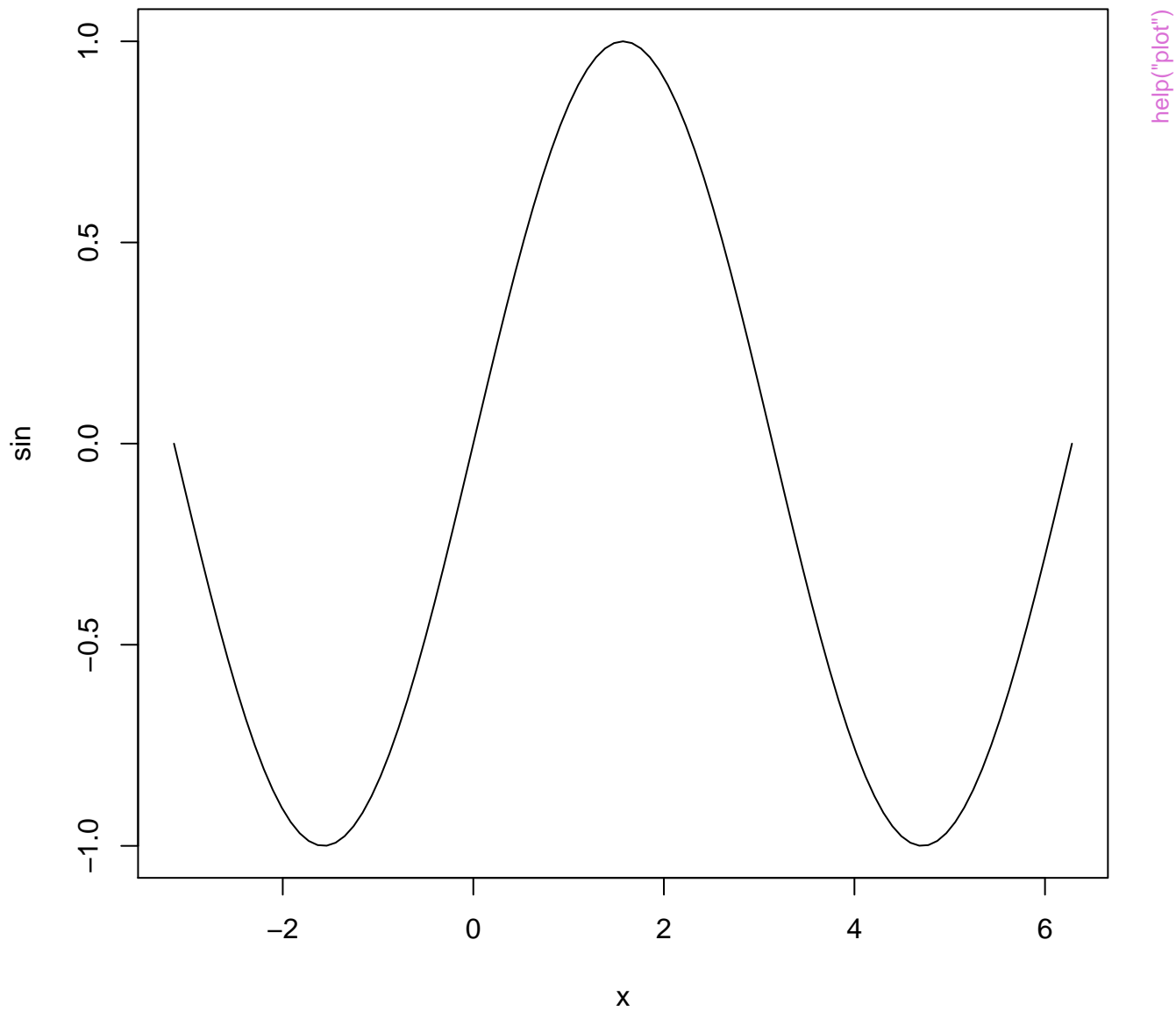
help("pie")

```
pie(*, labels="", col=rainbow(n), border=NA,..
```

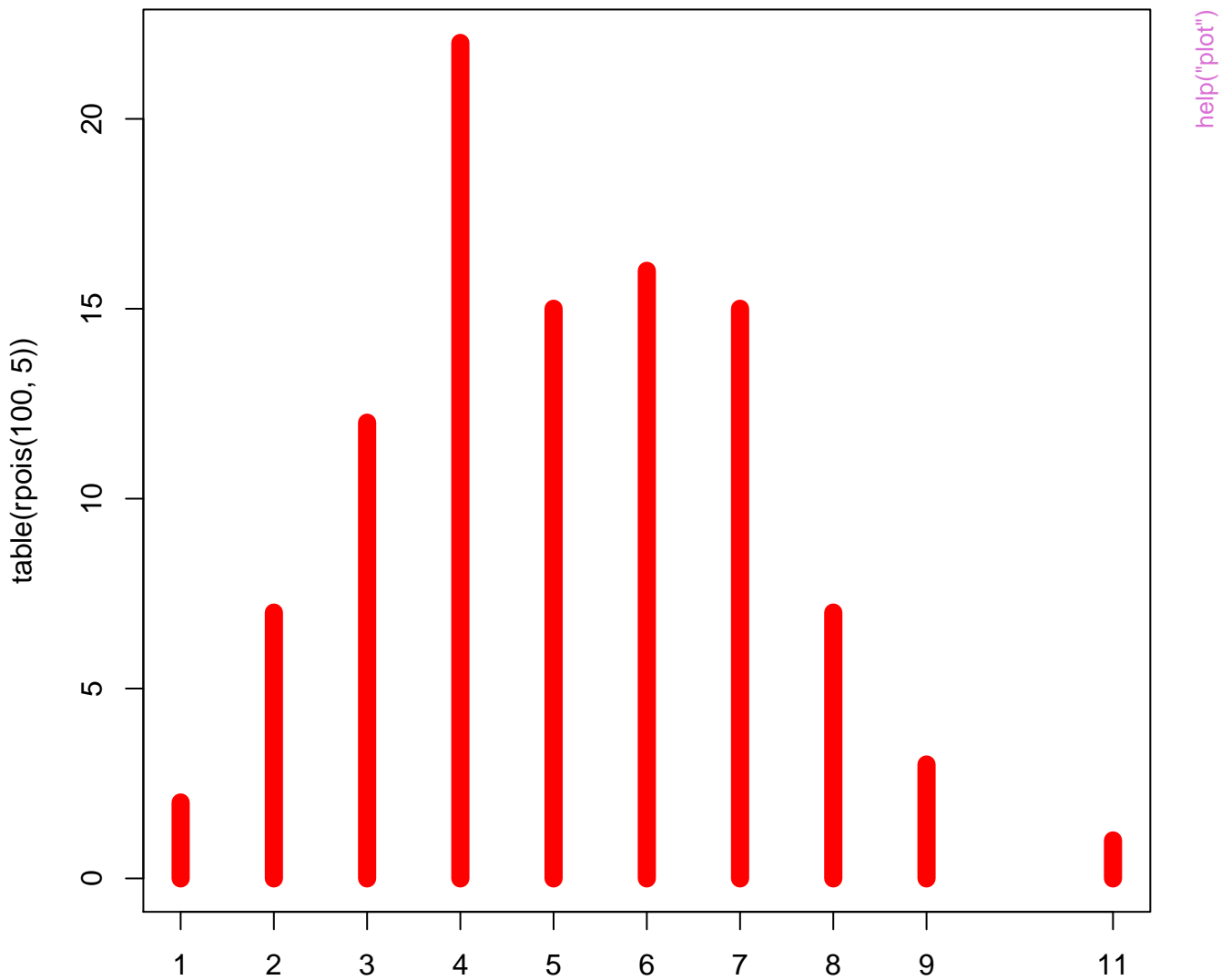


help("pie")



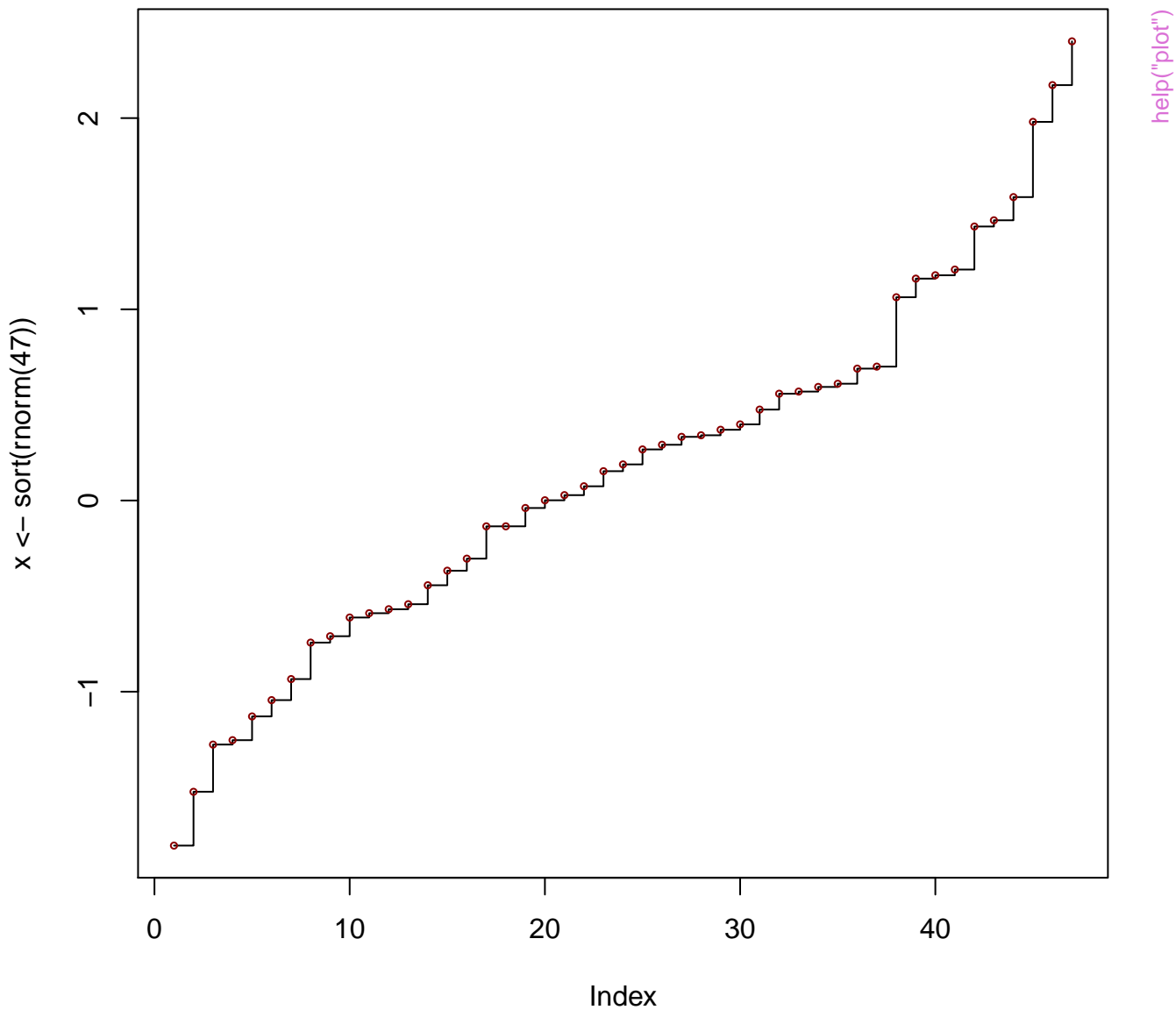


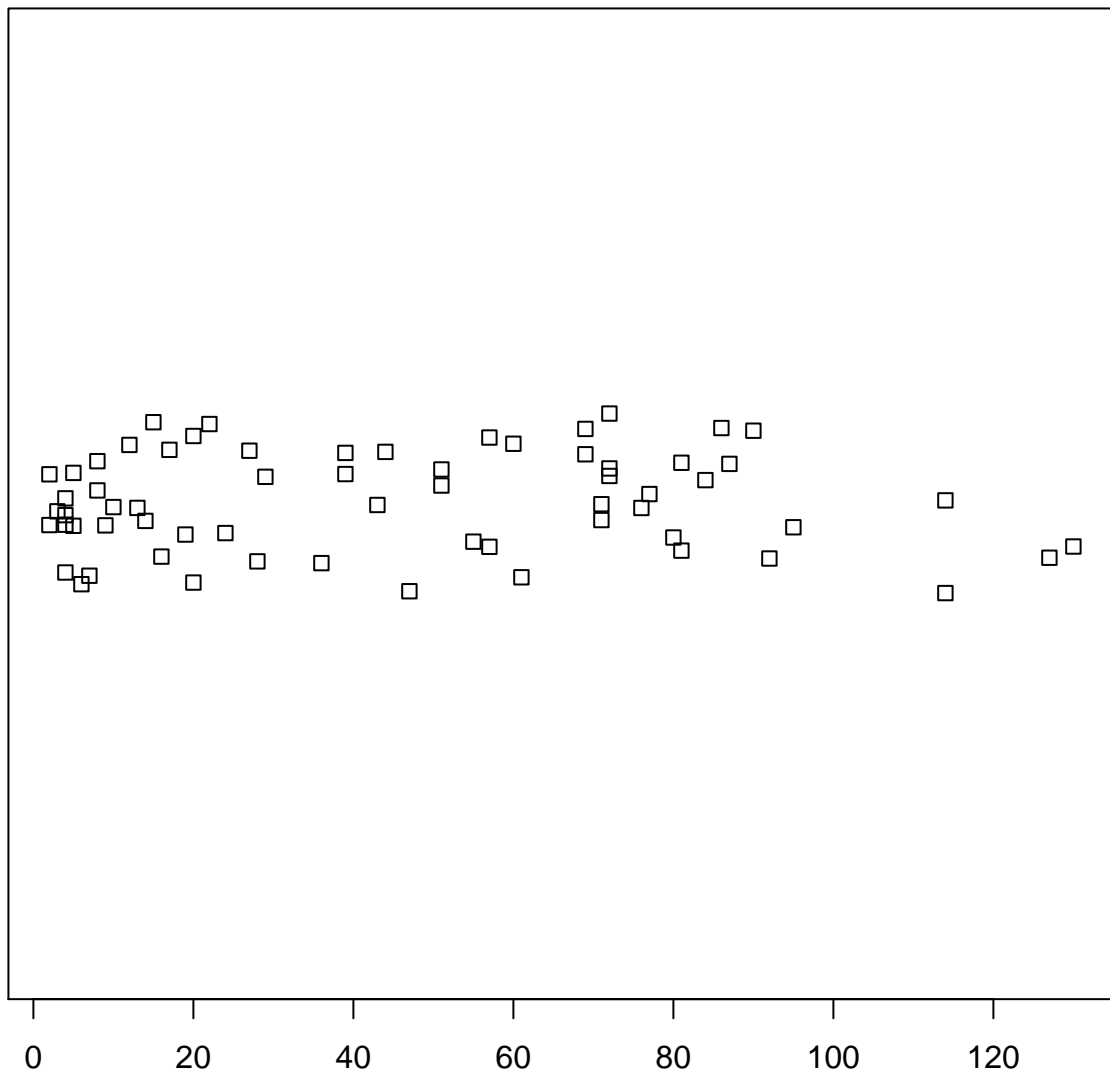
**rpois(100, lambda = 5)**



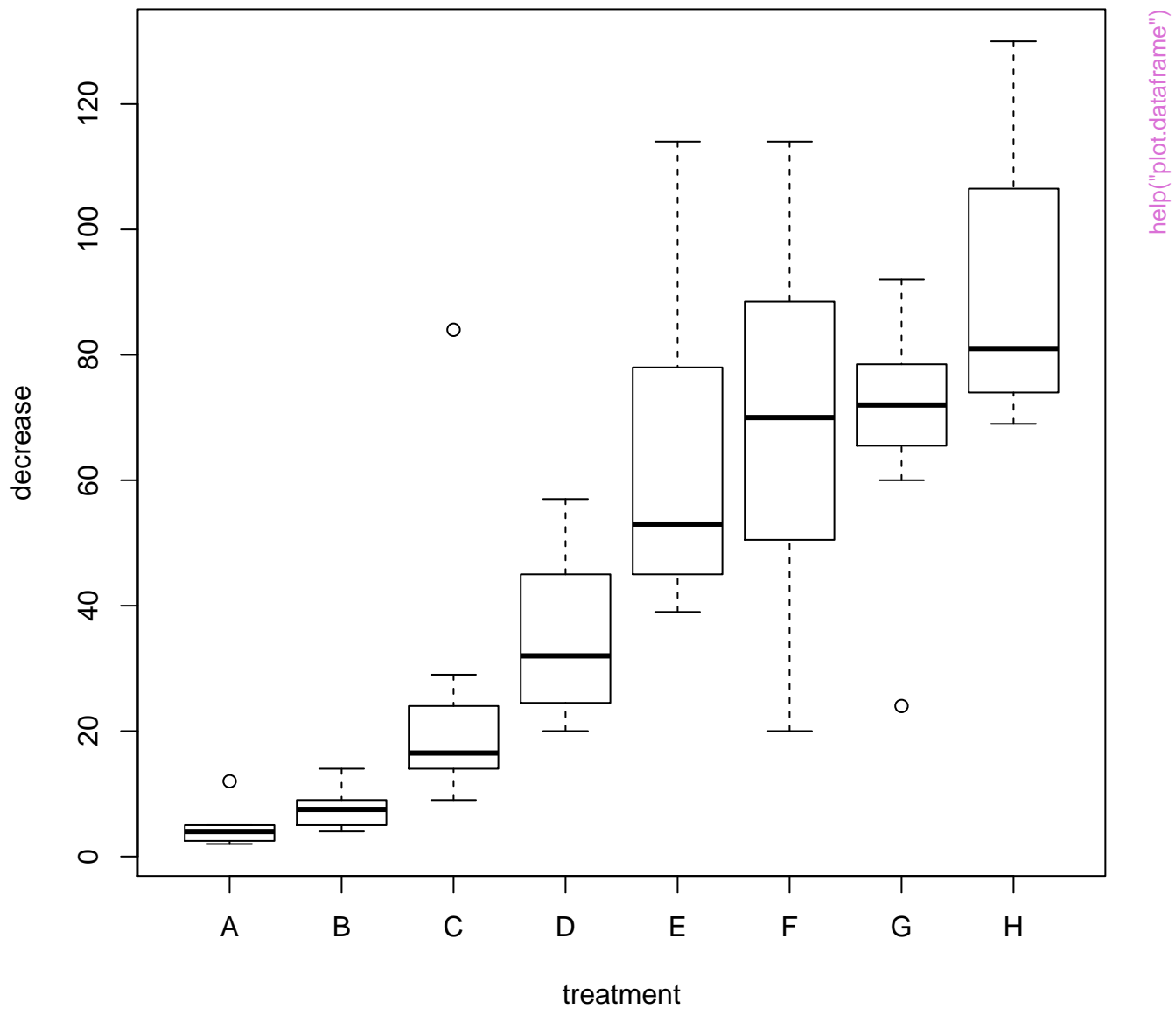
help("plot")

`plot(x, type = "s")`



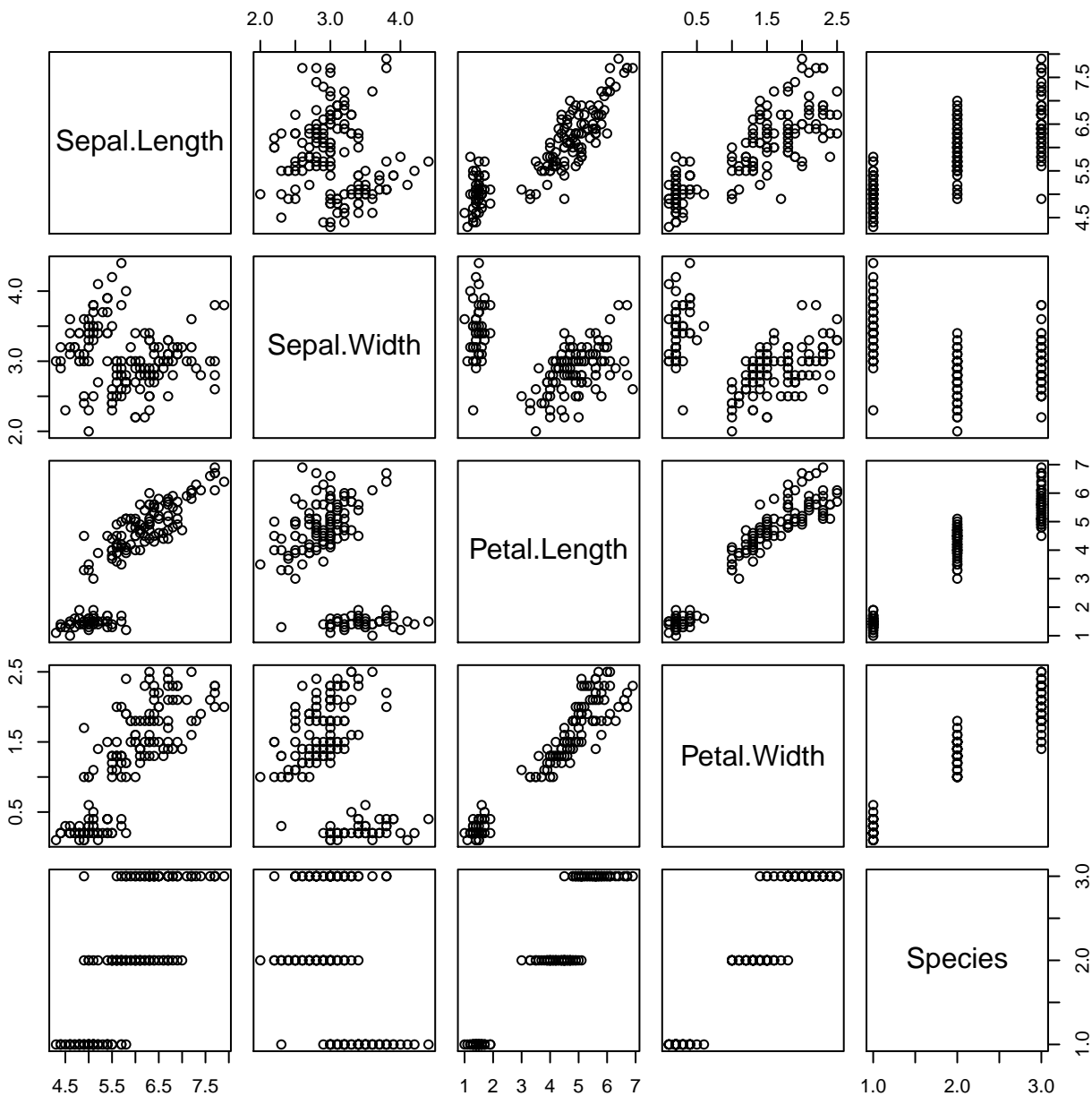


help("plot.dataframe")

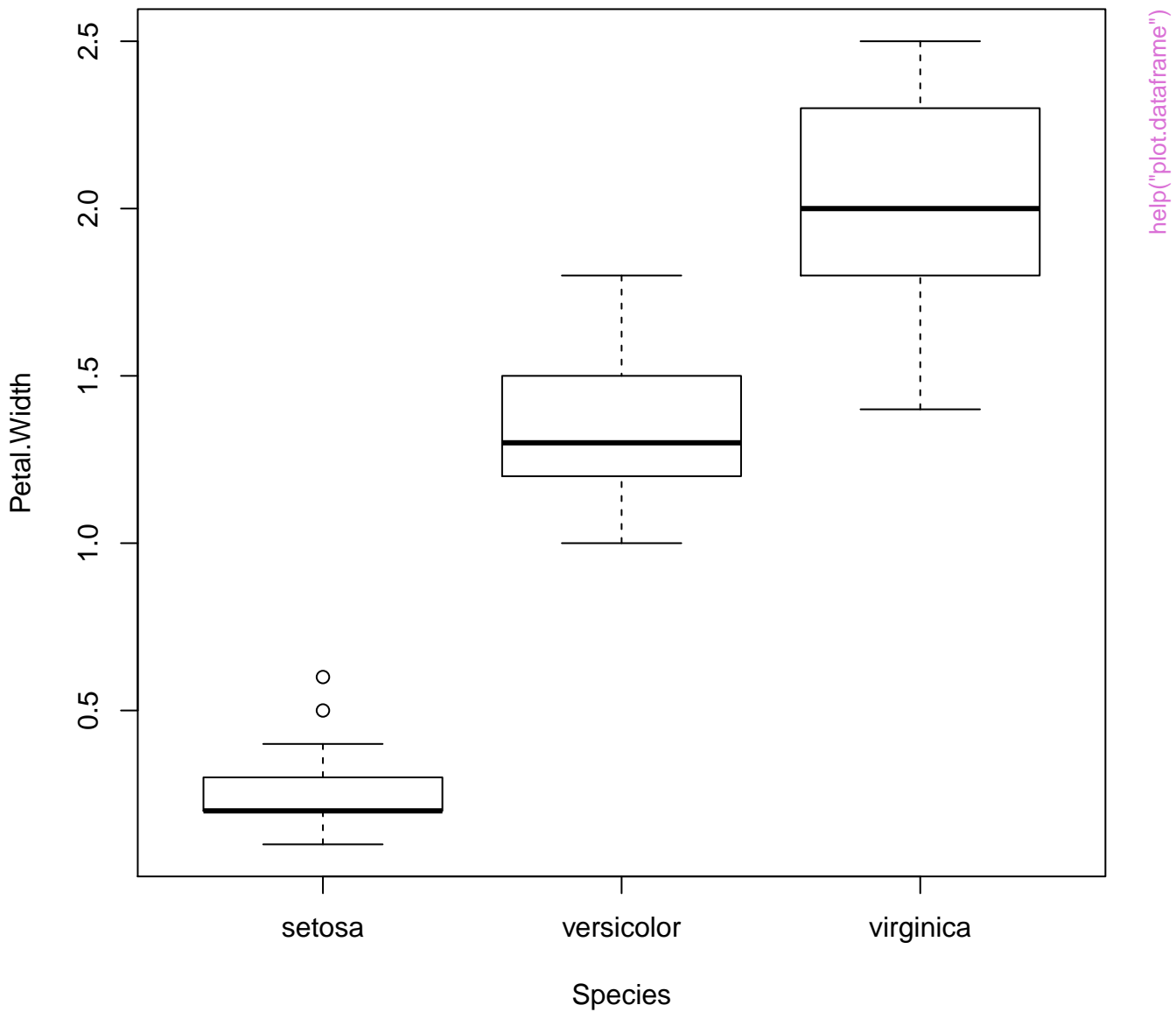


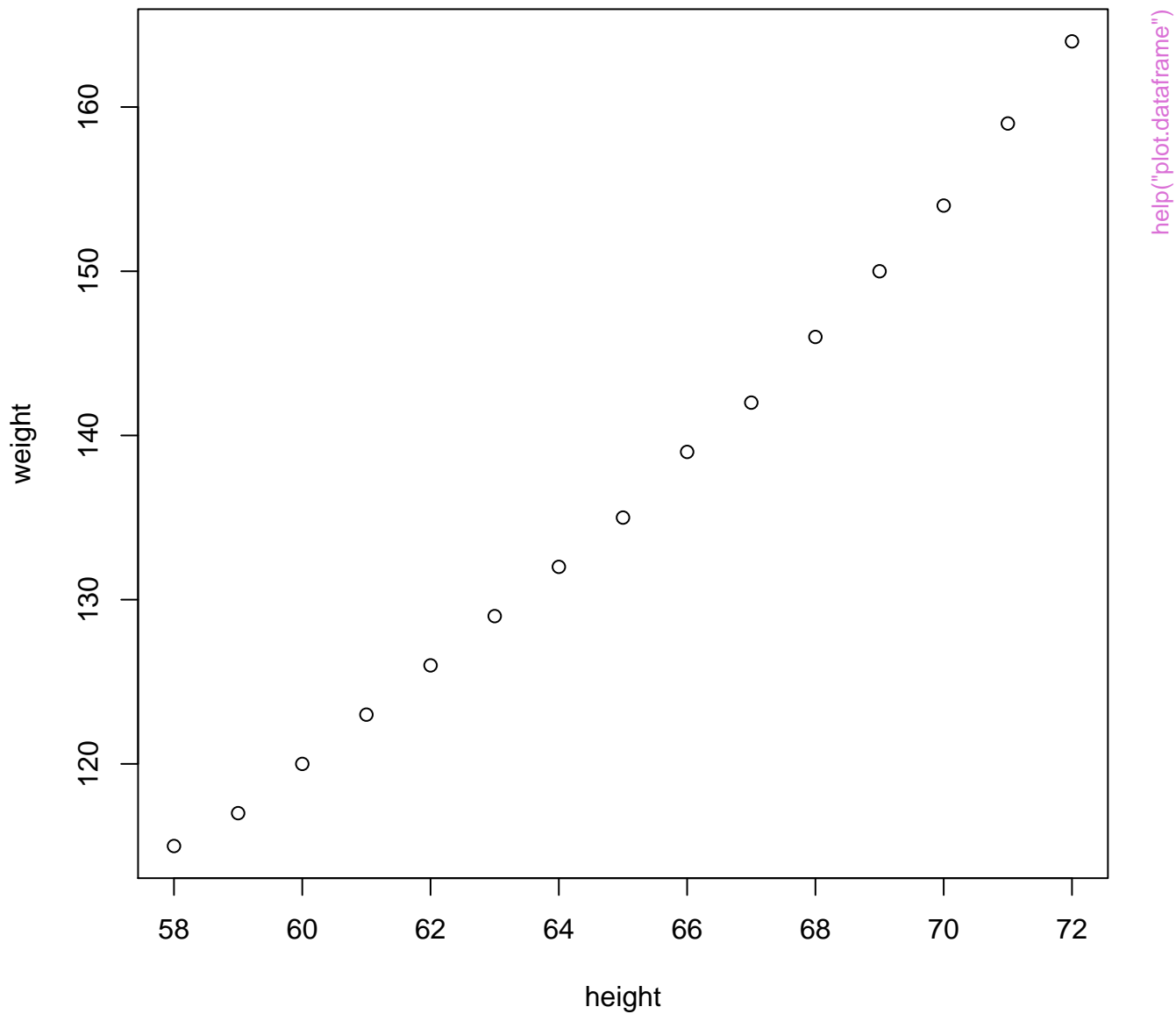


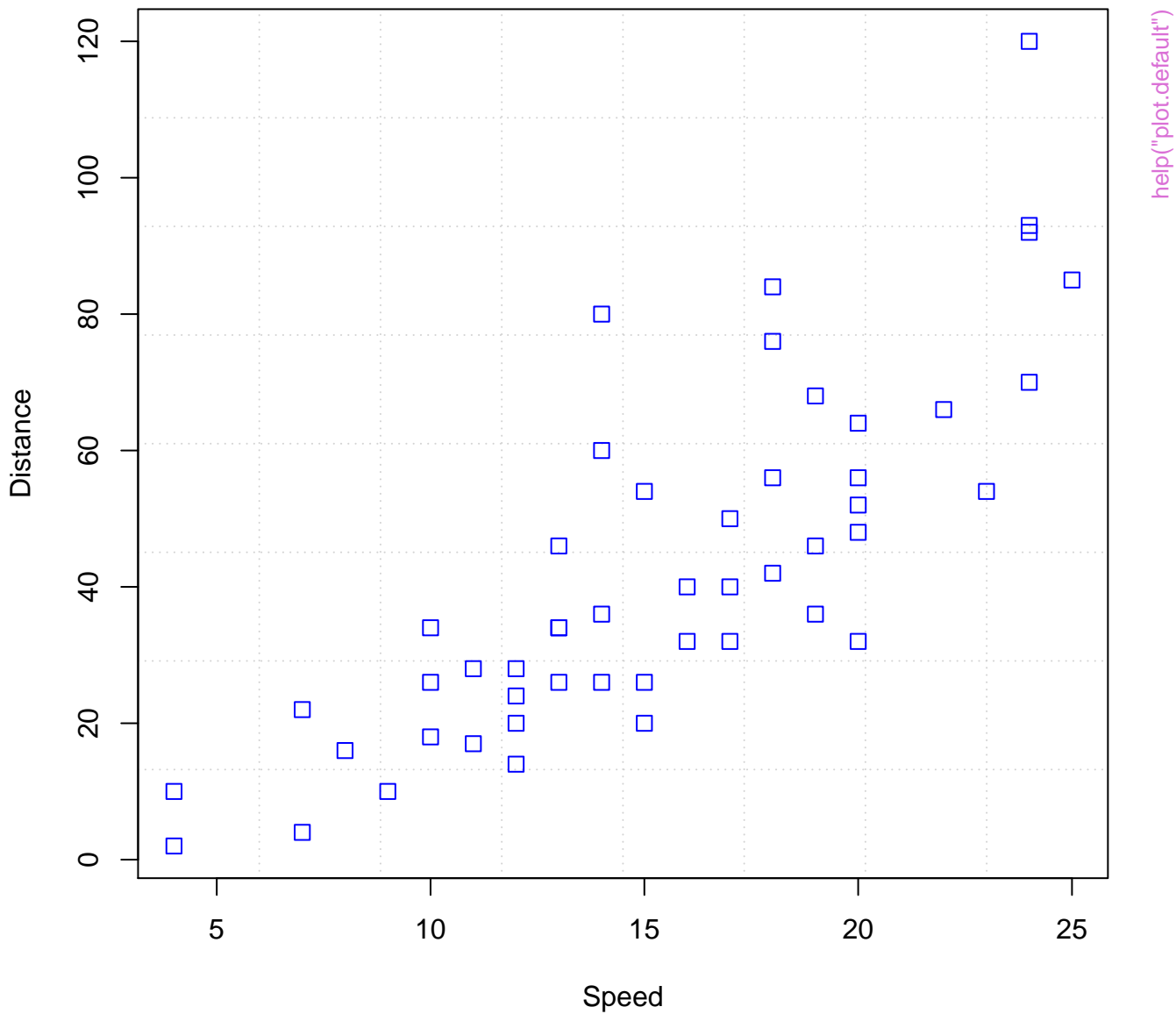


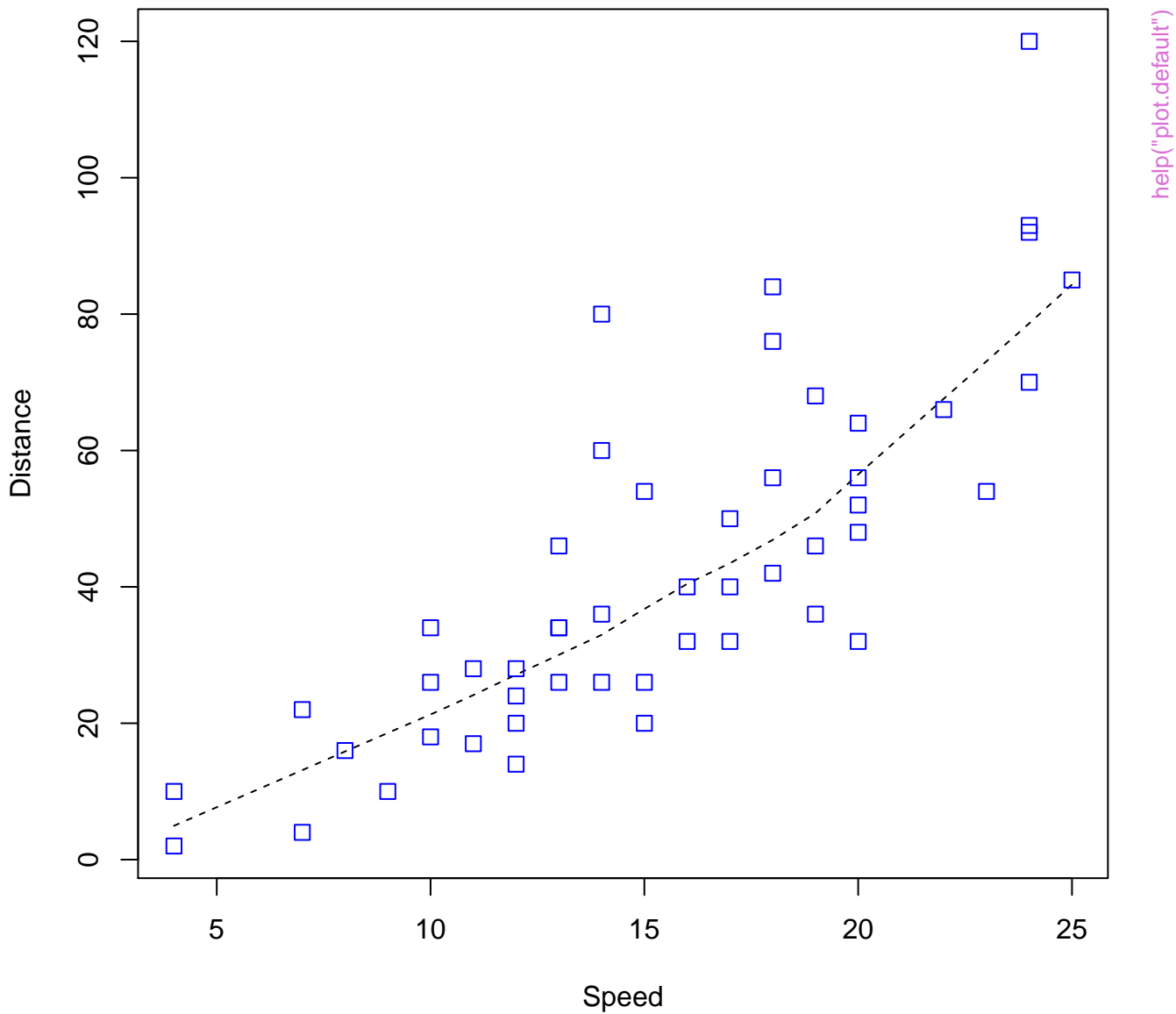


`help("plot.dataframe")`

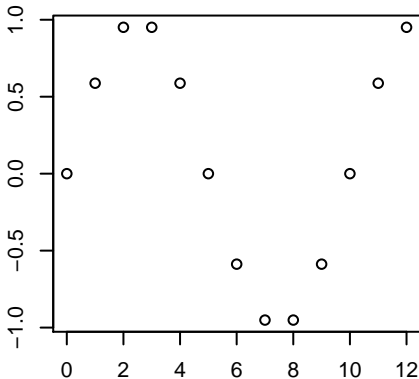




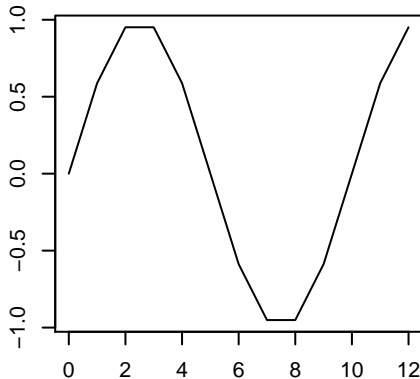




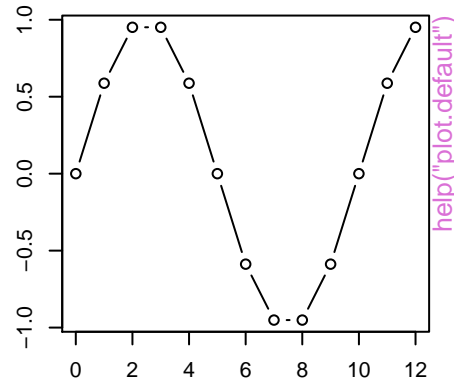
plot(\*, type = "p")



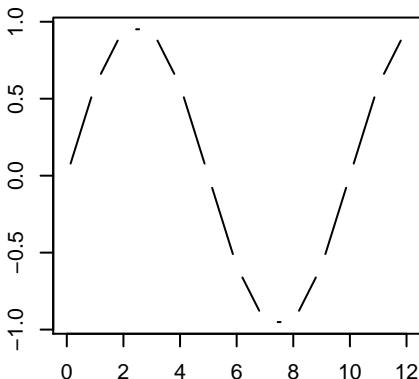
plot(\*, type = "l")



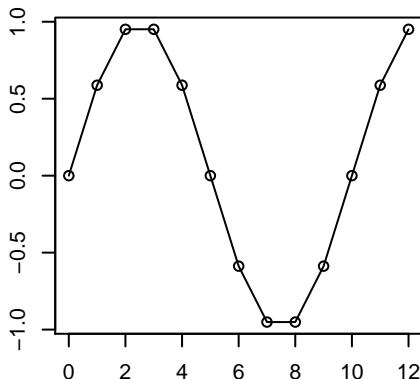
plot(\*, type = "b")



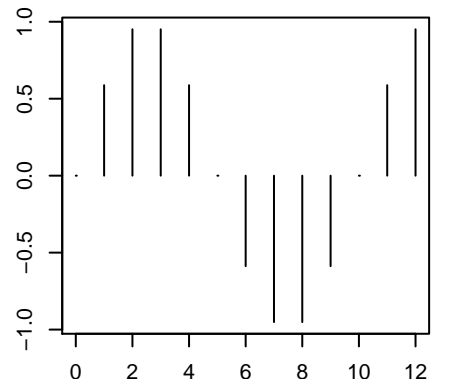
plot(\*, type = "c")



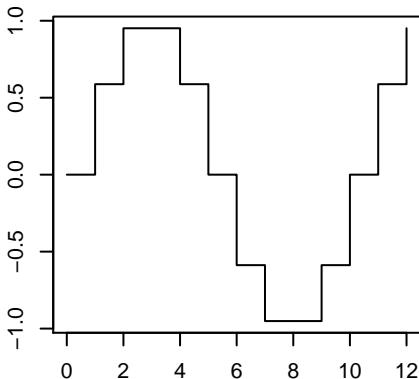
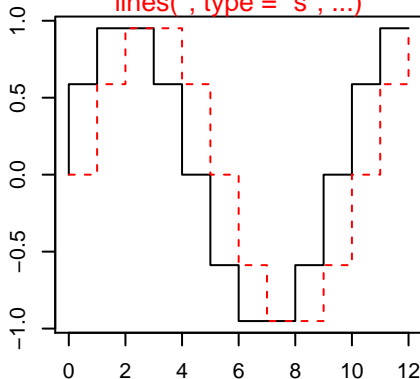
plot(\*, type = "o")



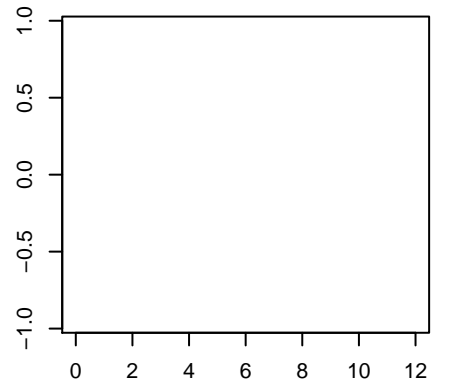
plot(\*, type = "h")



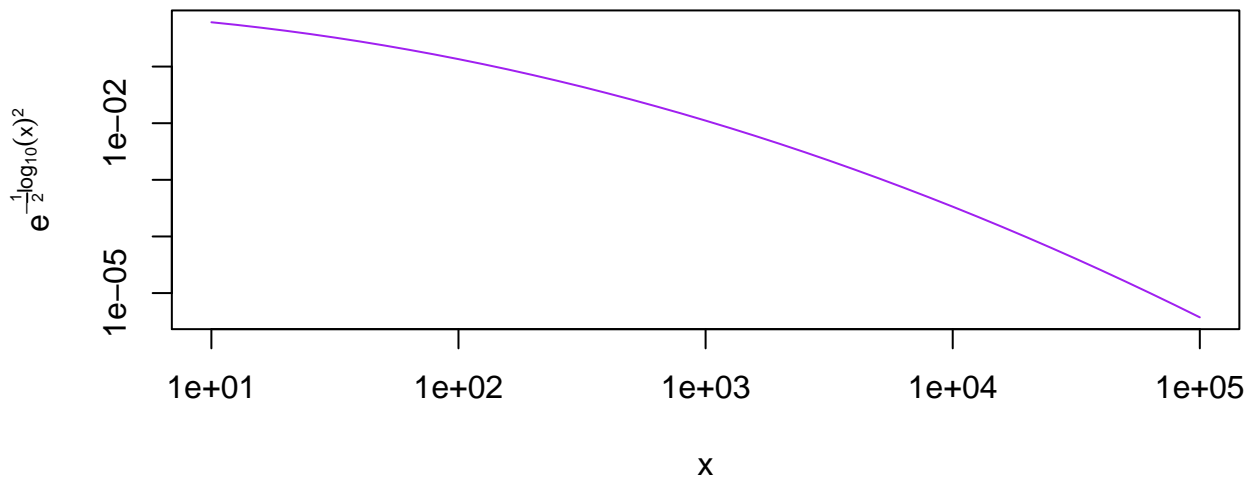
plot(\*, type = "s")

plot(\*, type = "S")  
lines(\*, type = "s", ...)

plot(\*, type = "n")

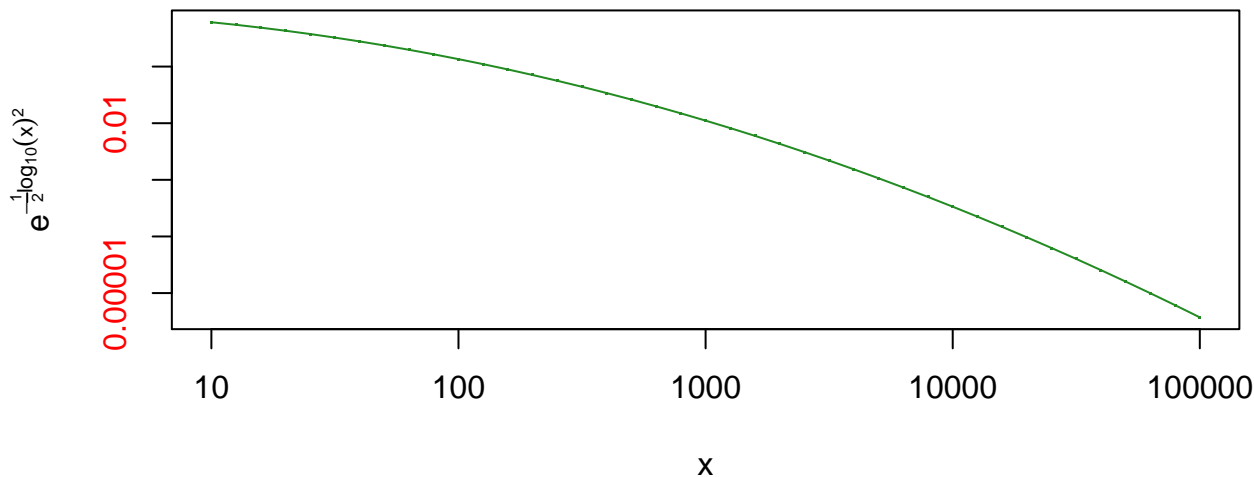


**Log-Log plot**

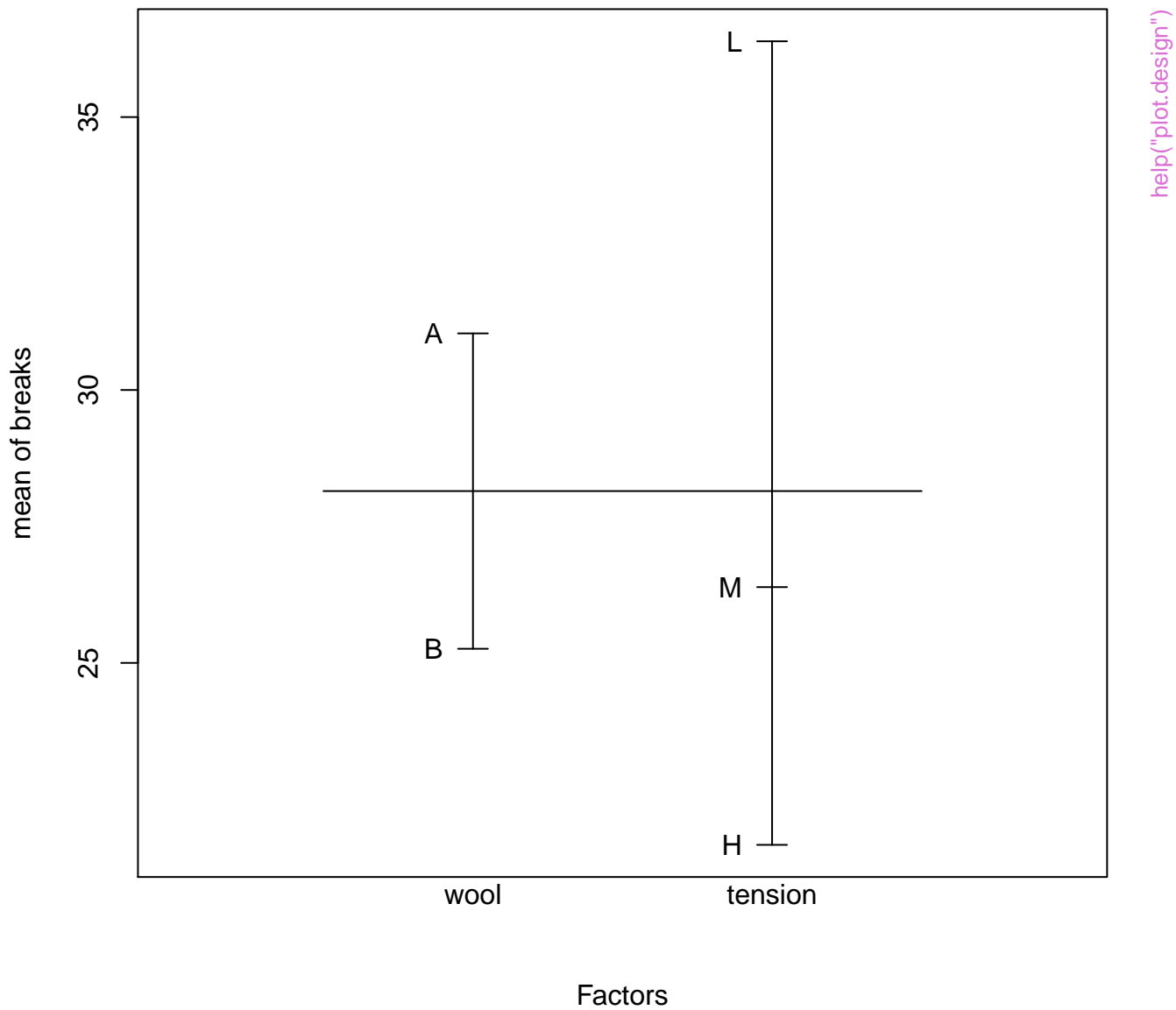


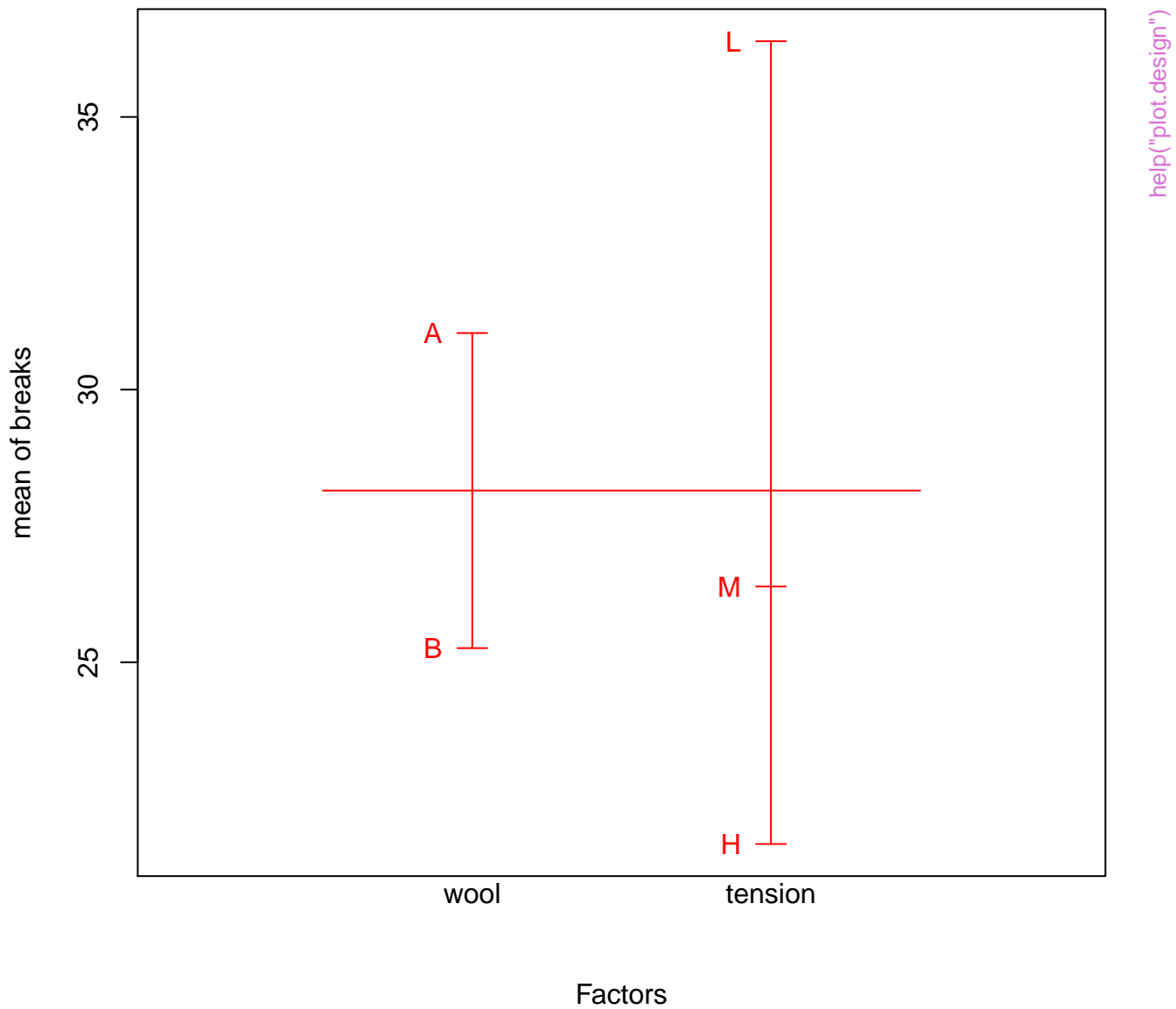
help("plot.default")

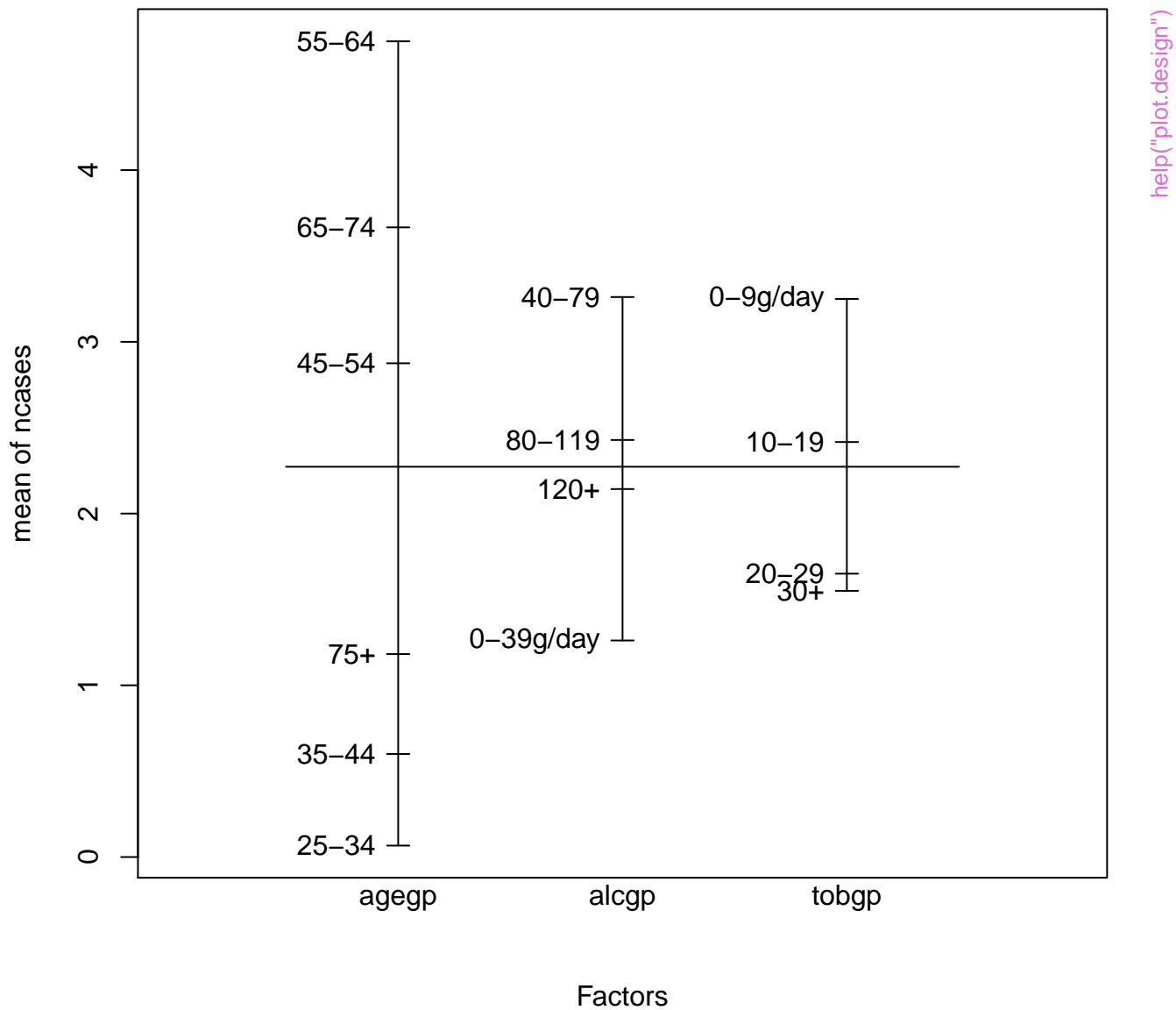
**Log-Log plot with custom axes**

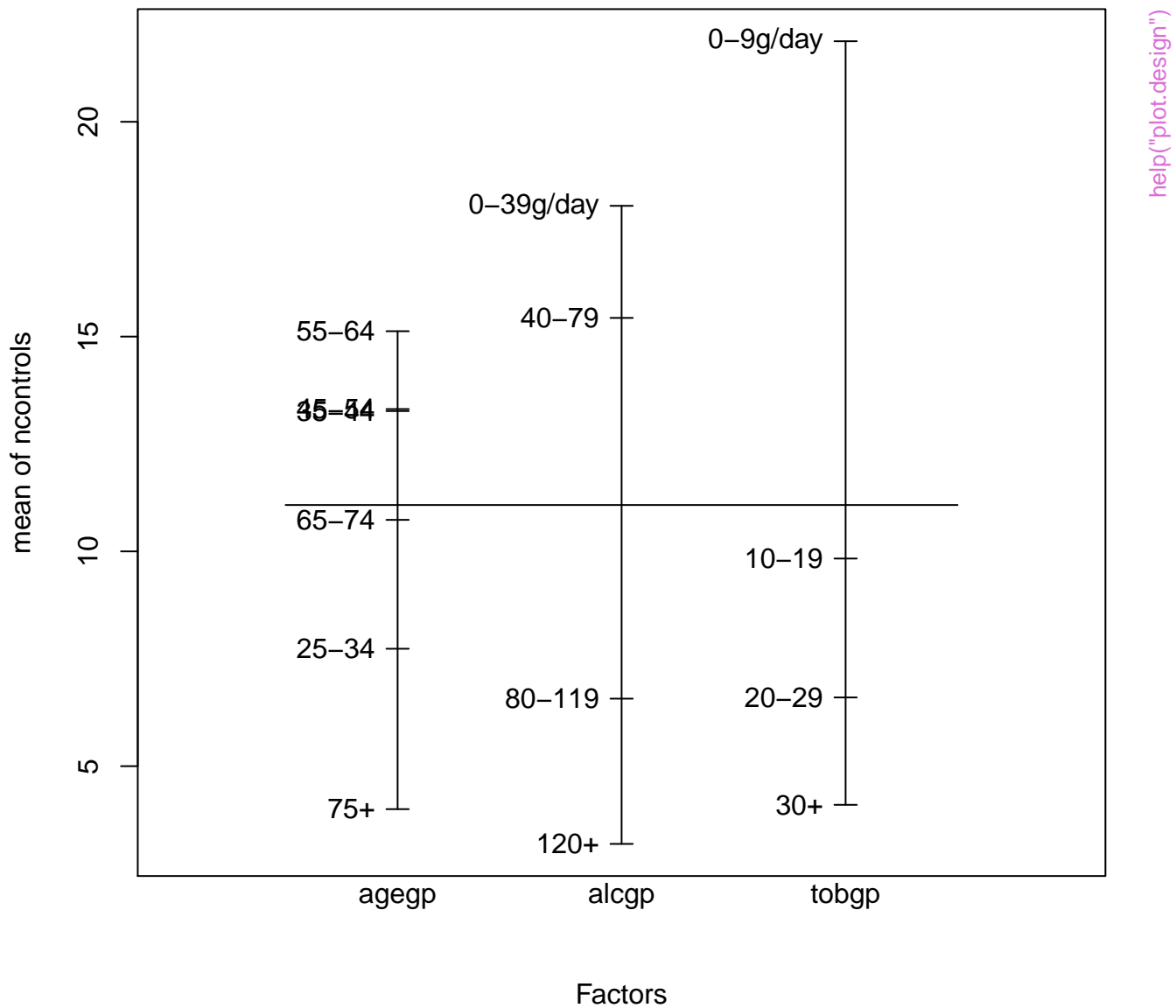


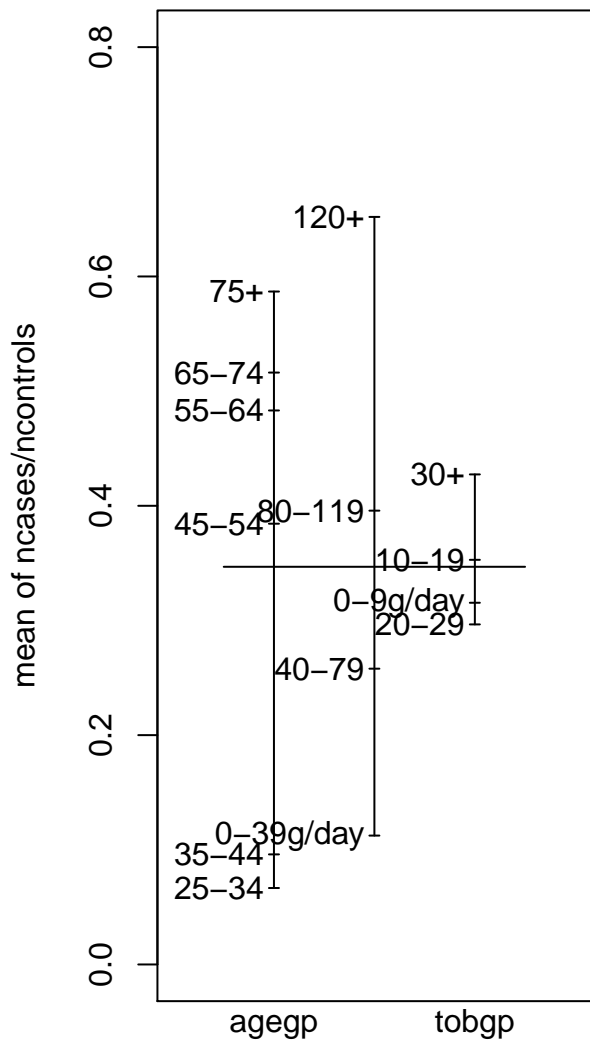




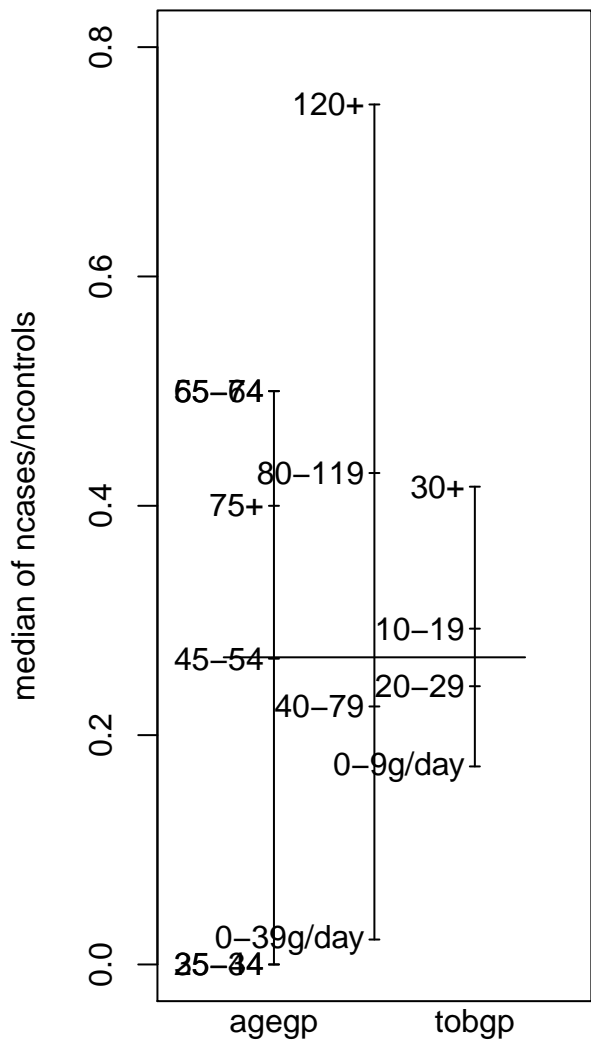




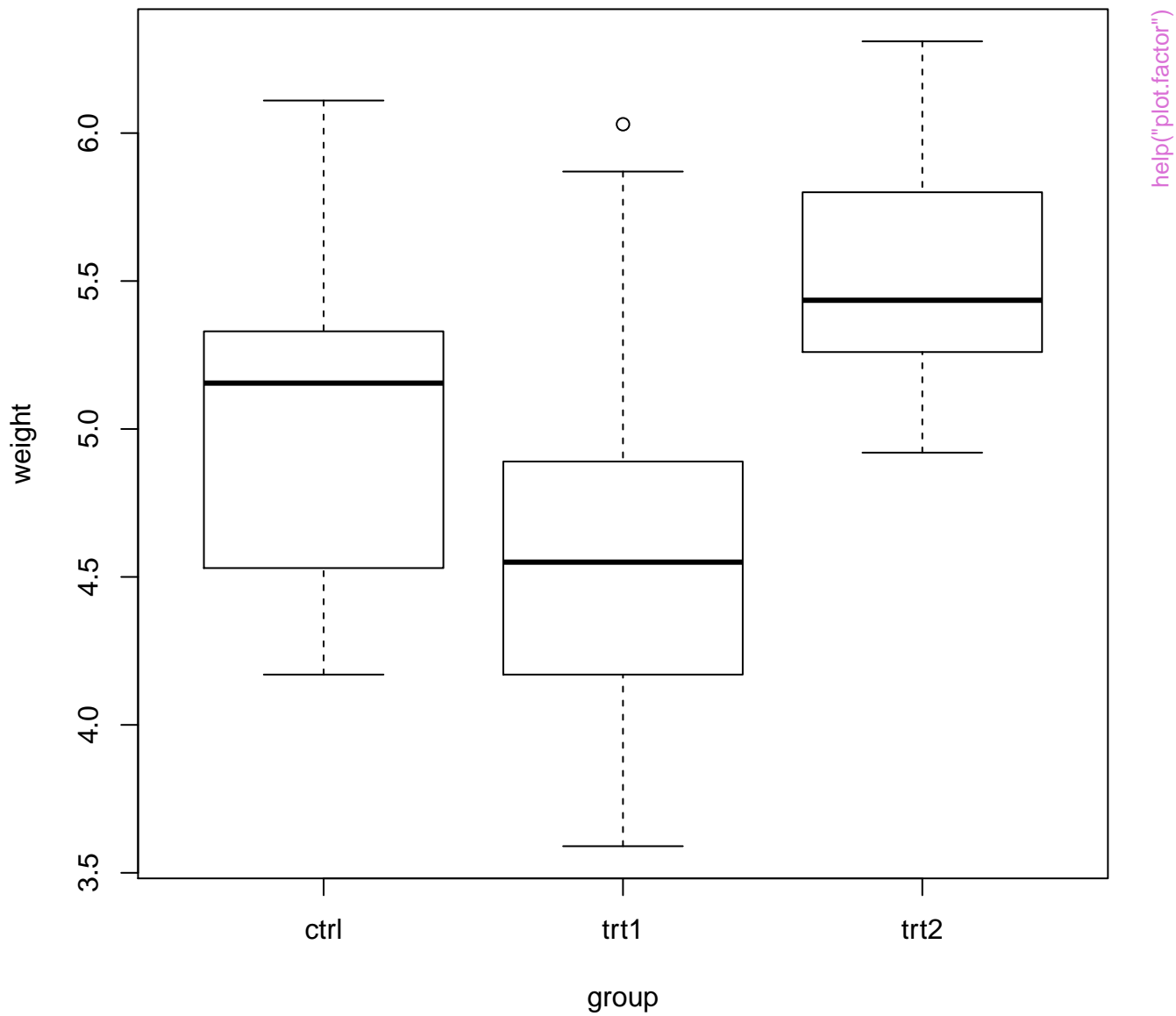


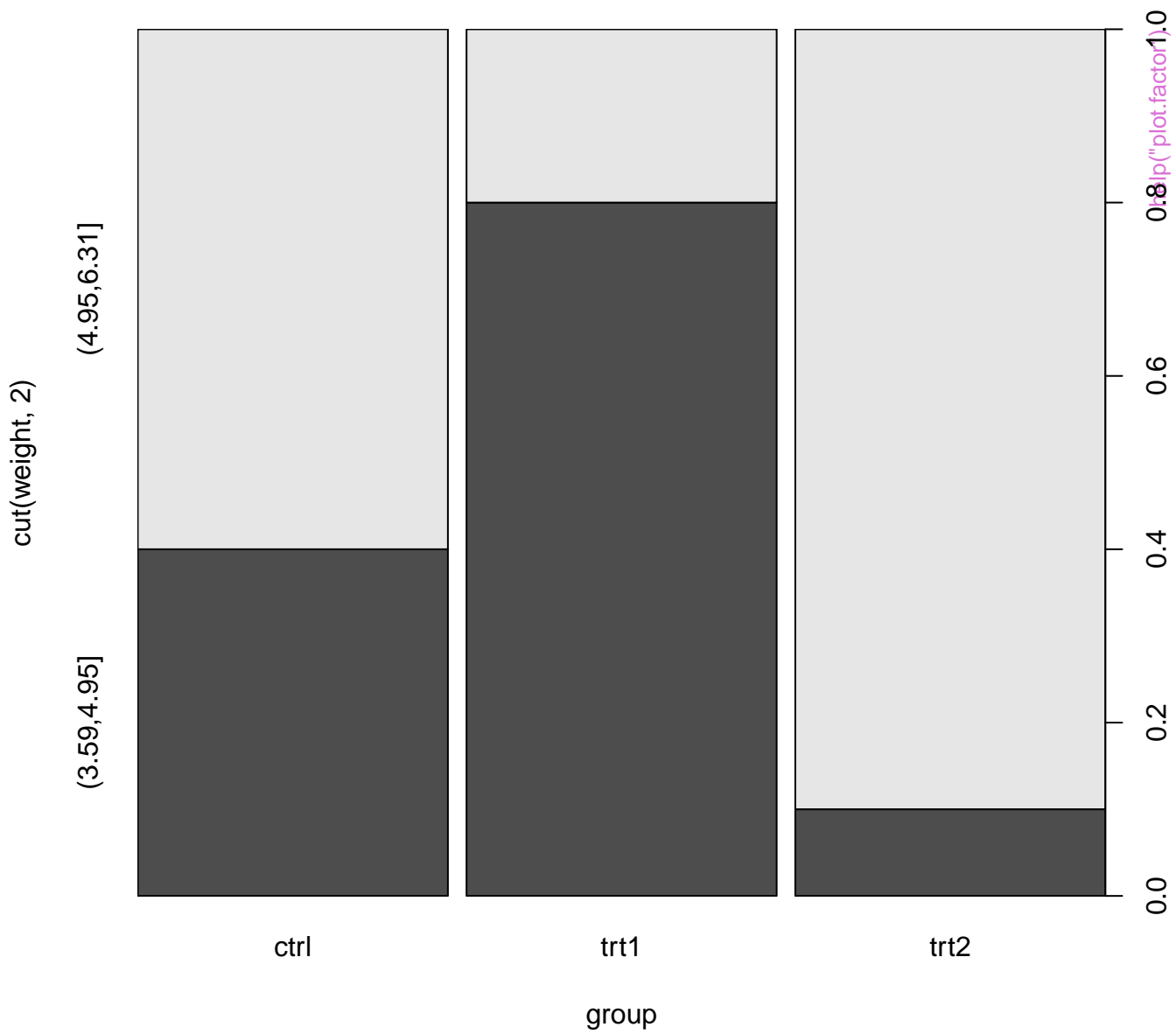


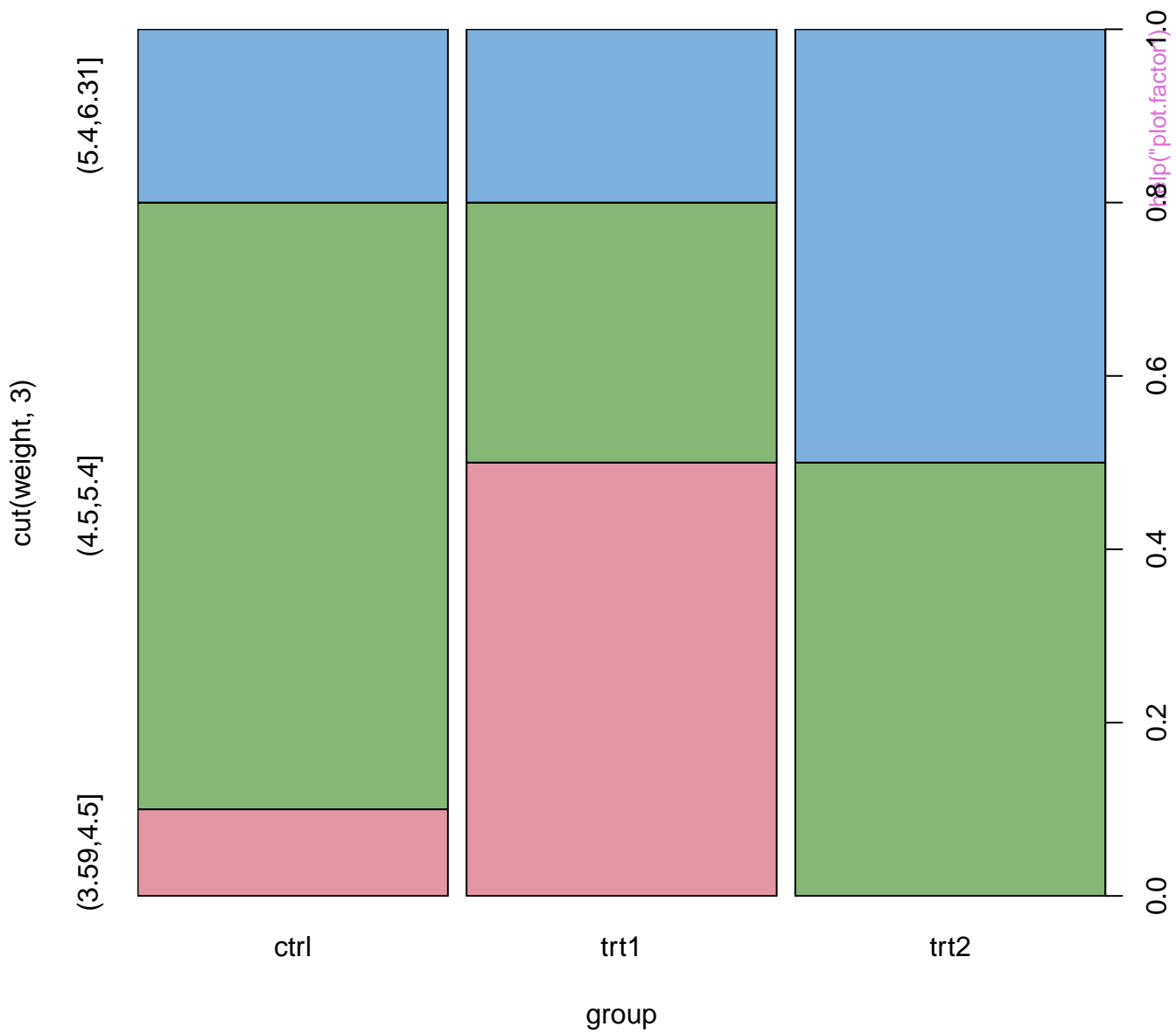
Factors



Factors

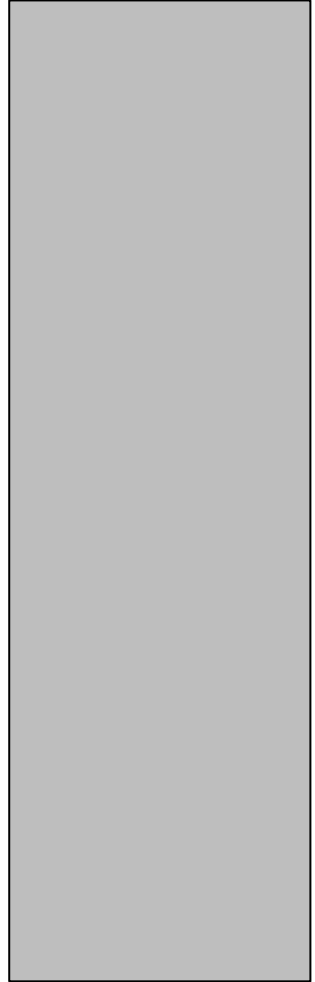
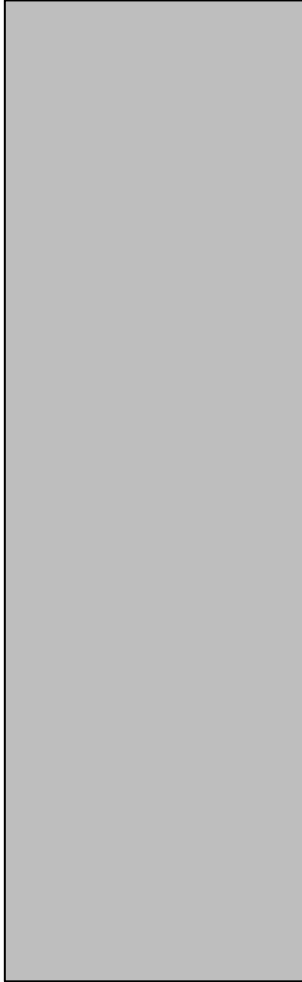




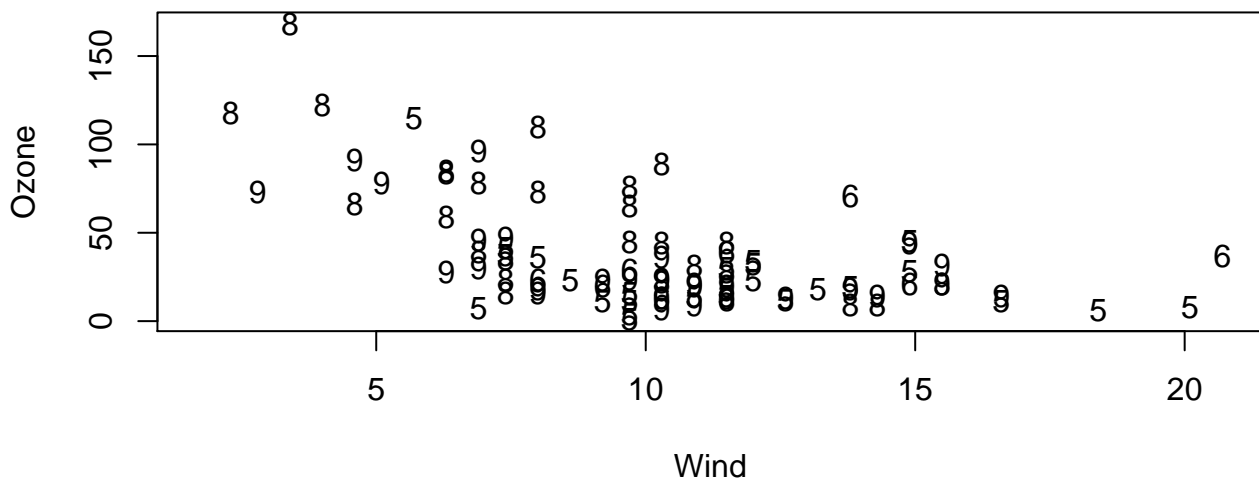
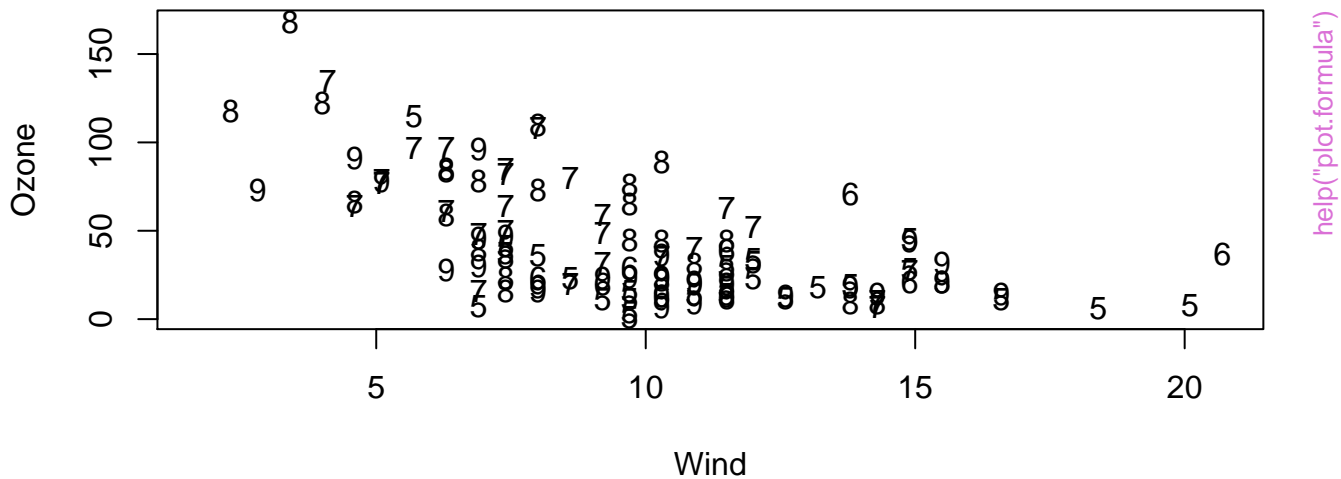




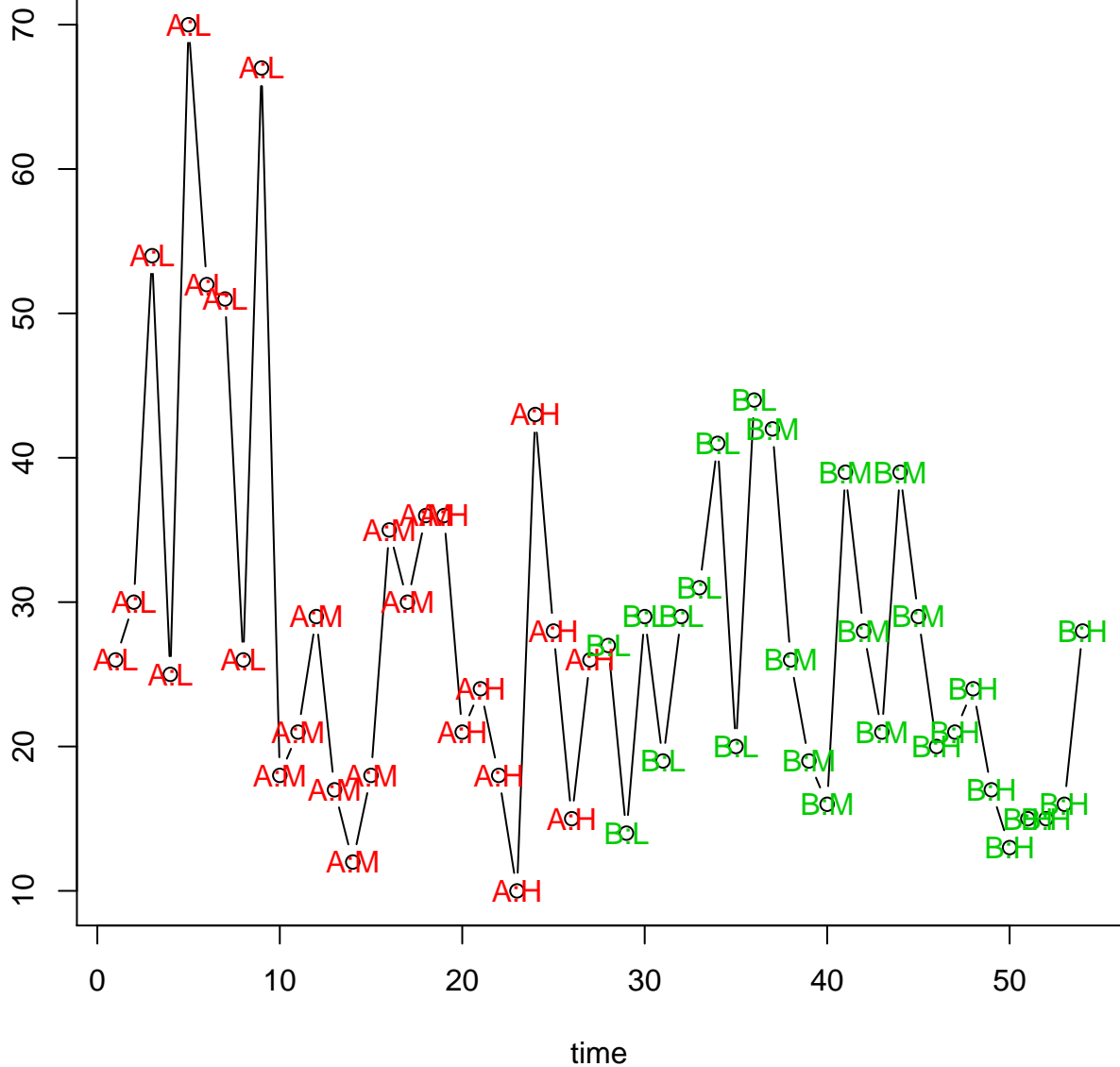
**no axes**



`help("plot.factor")`

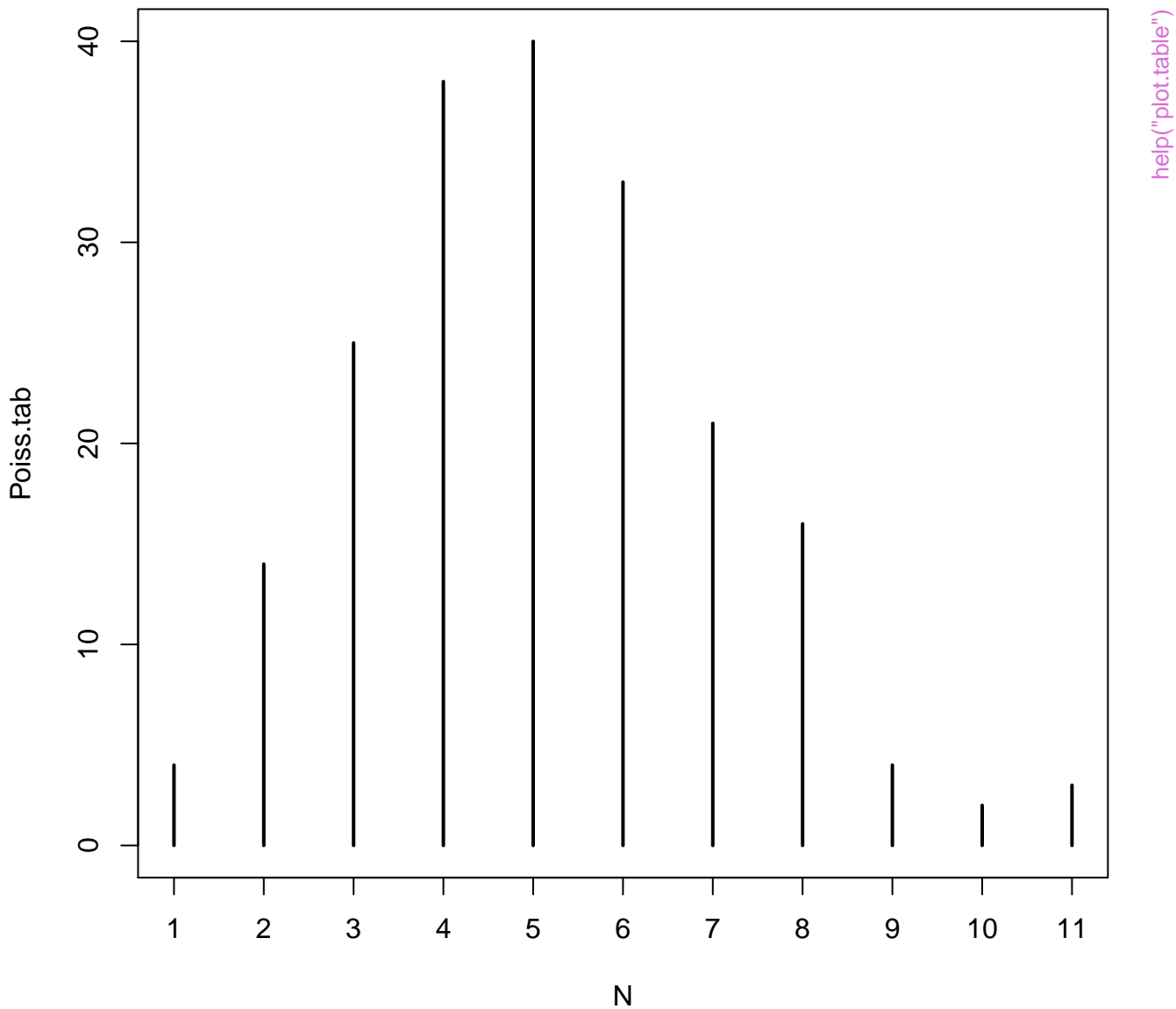


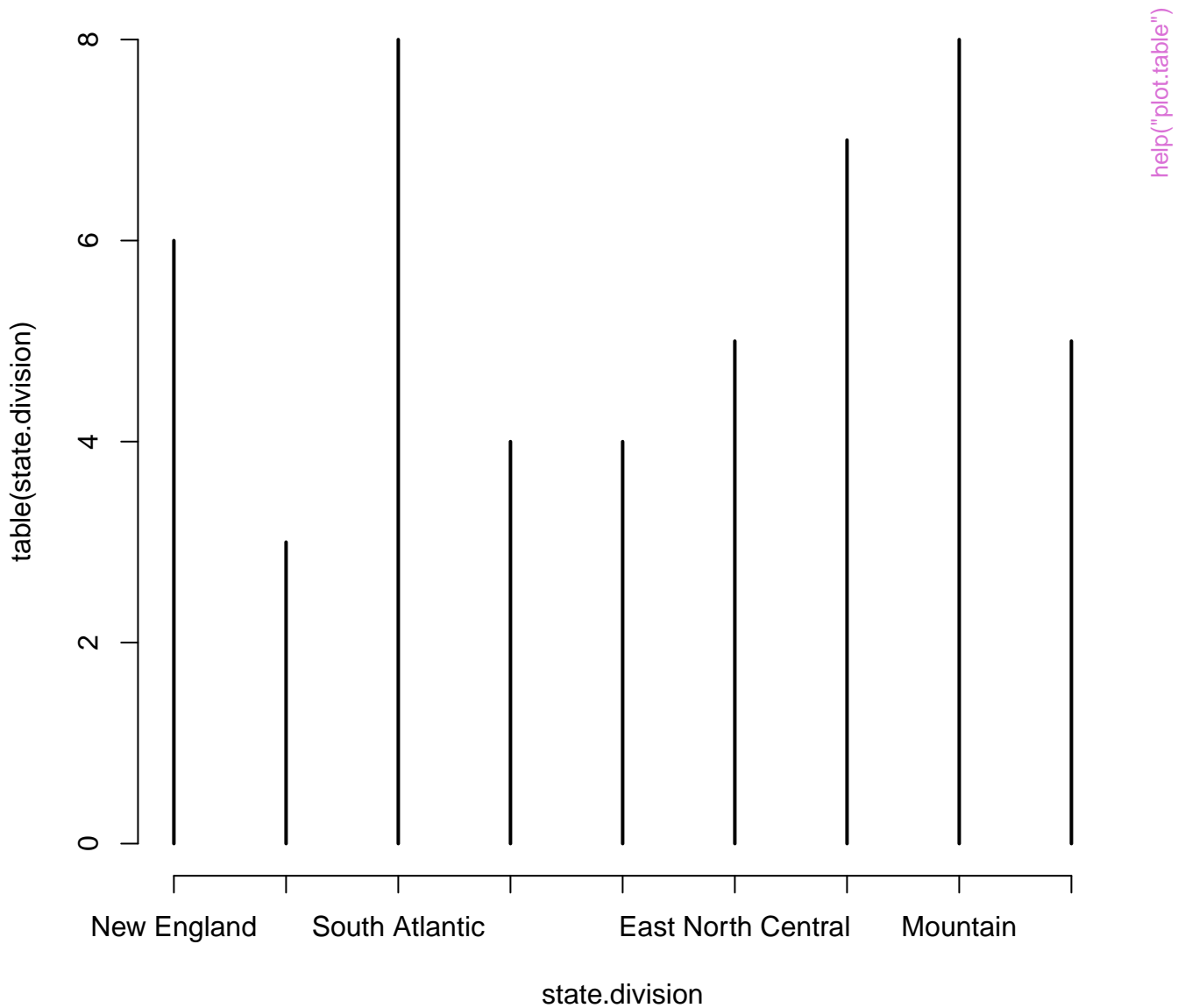
breaks



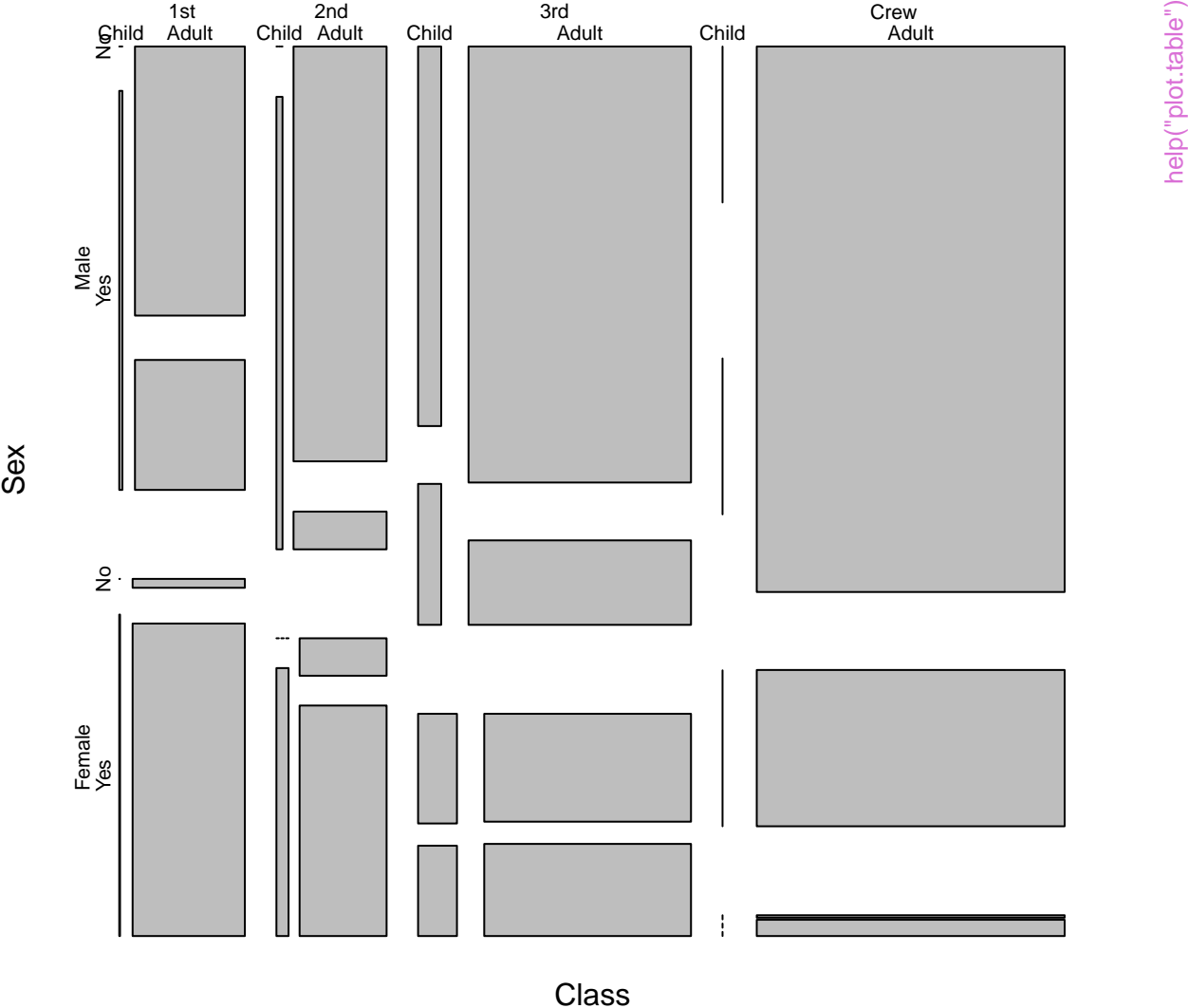
help("plot.formula")

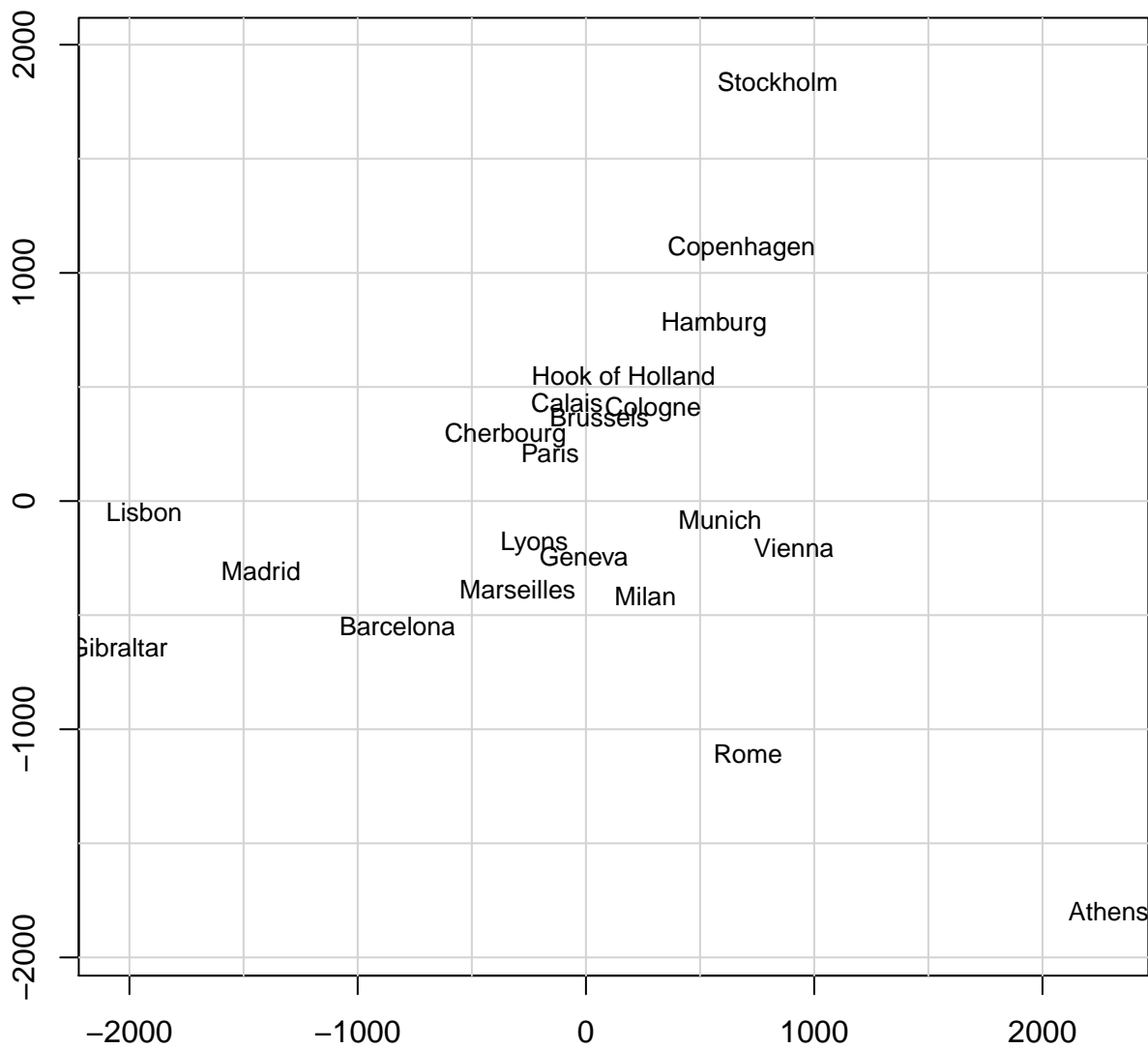
`plot(table(rpois(200, lambda = 5)))`





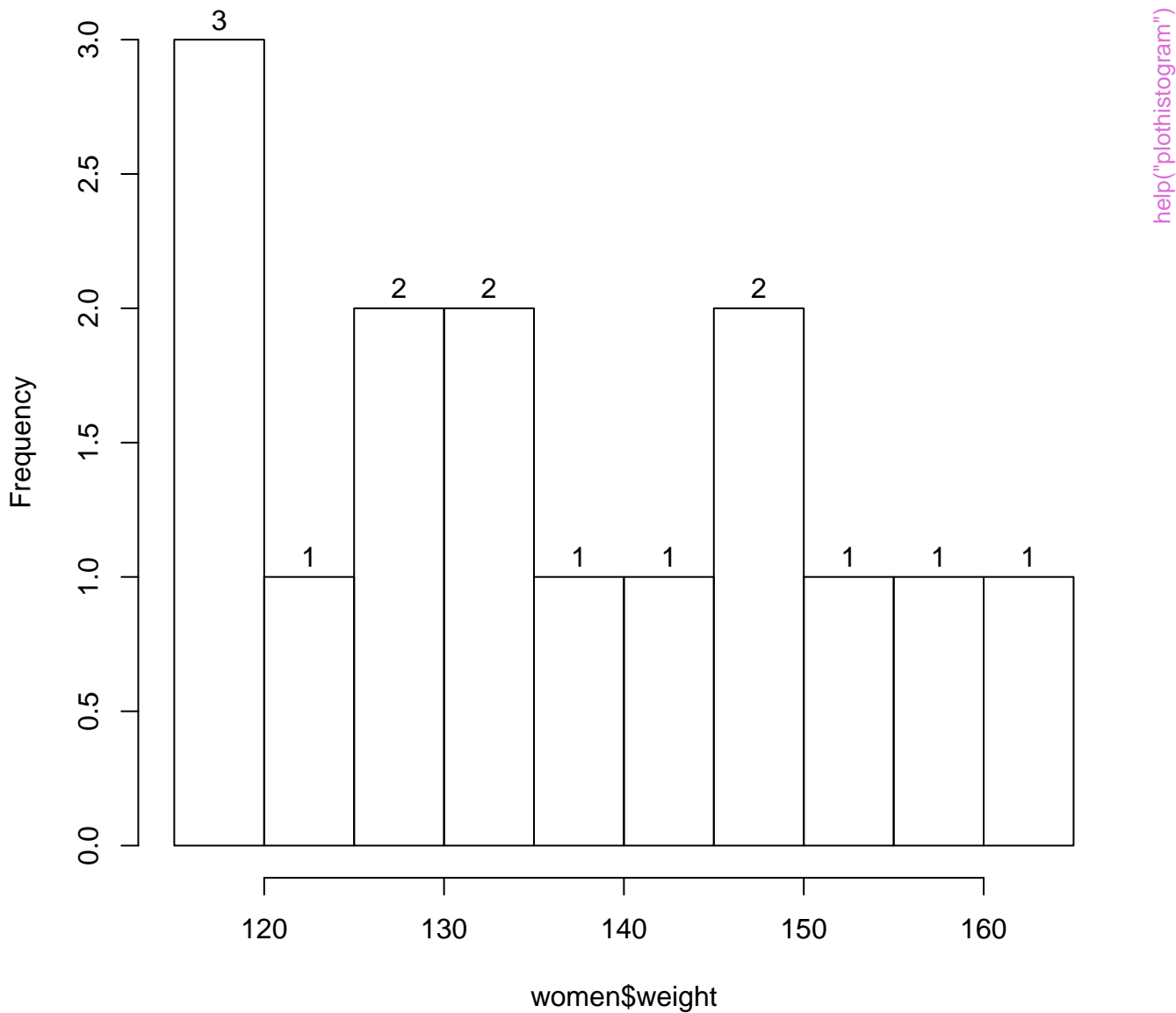
plot(Titanic, main= \*)





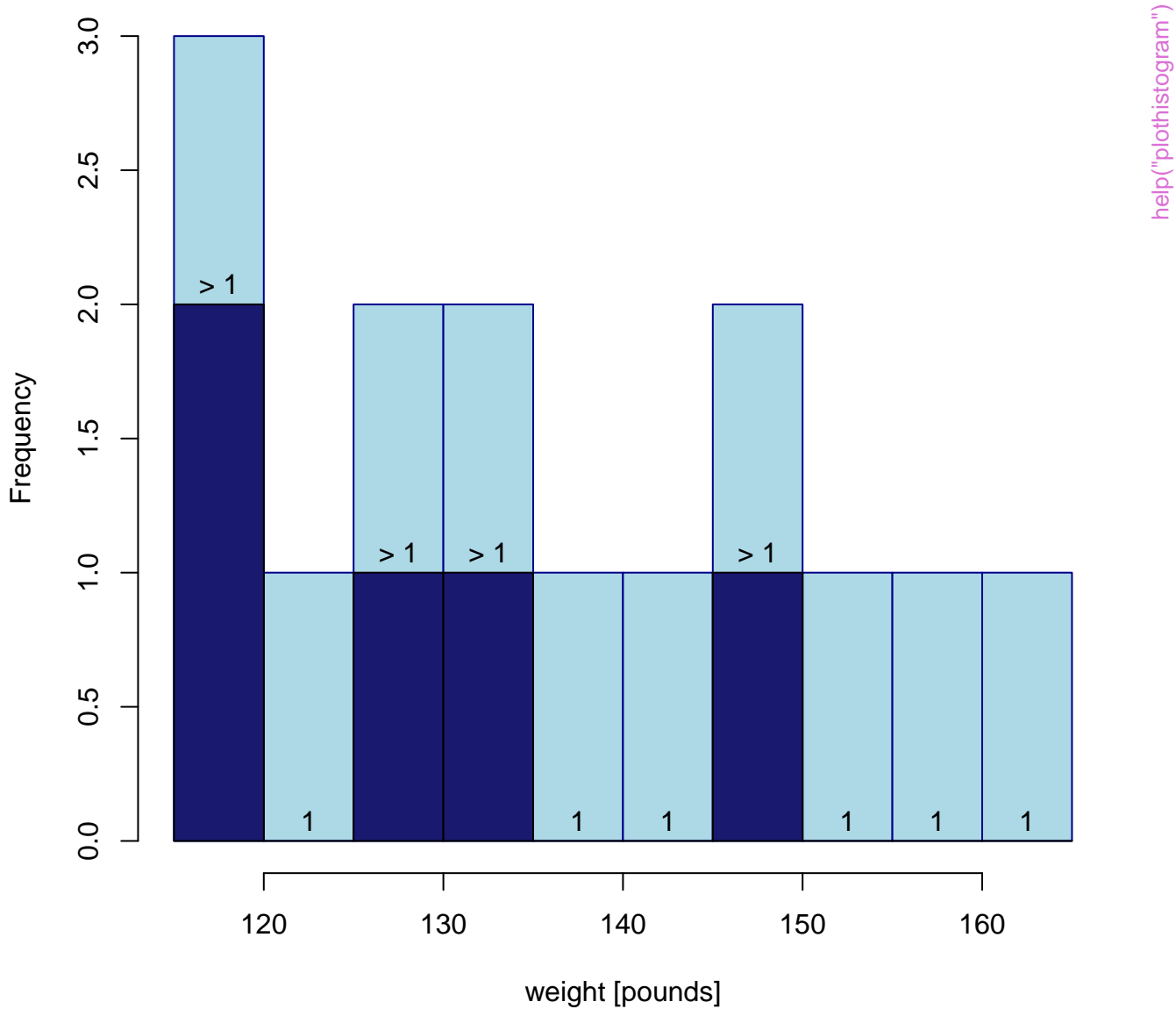
[help\("plot.window"\)](#)

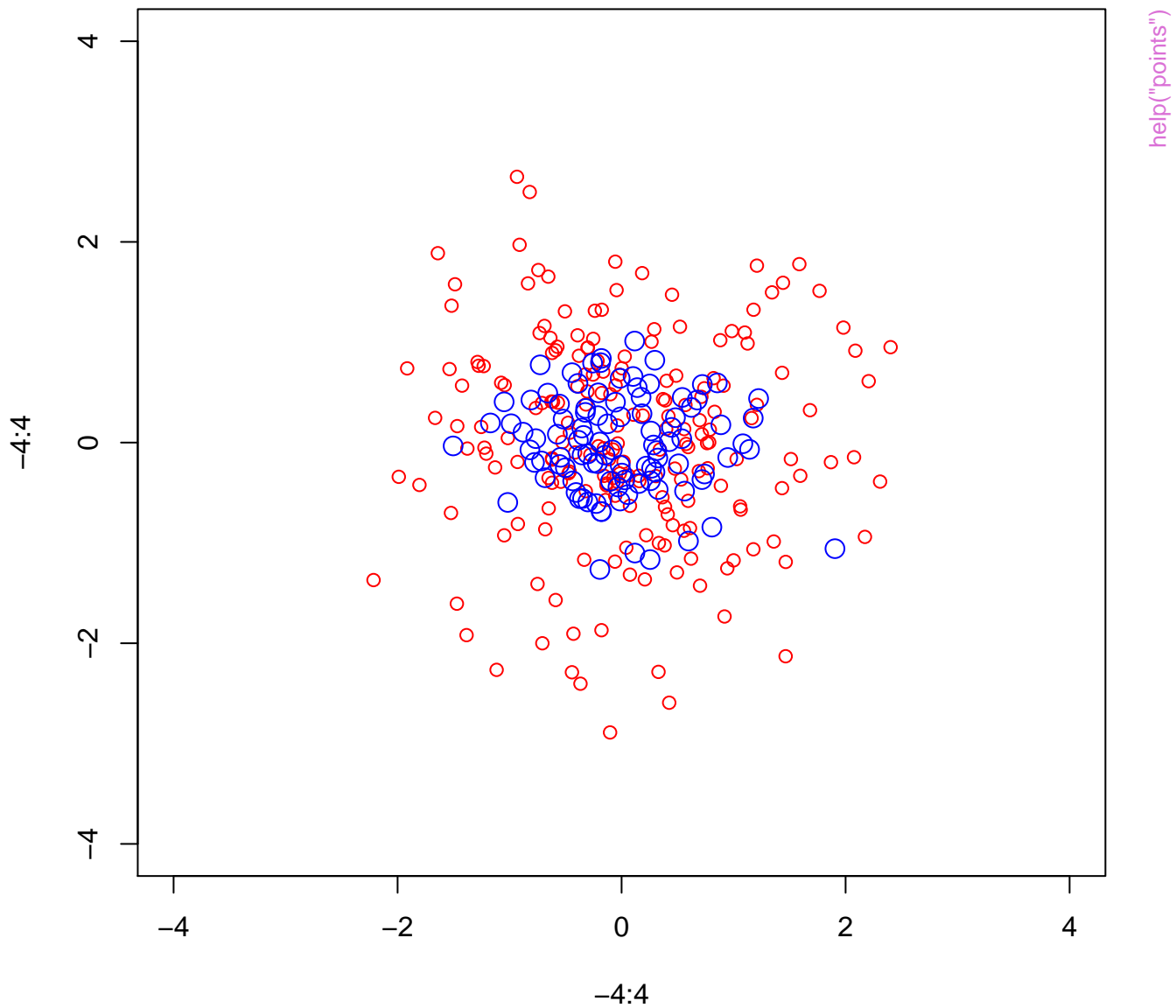
**Histogram of women\$weight**



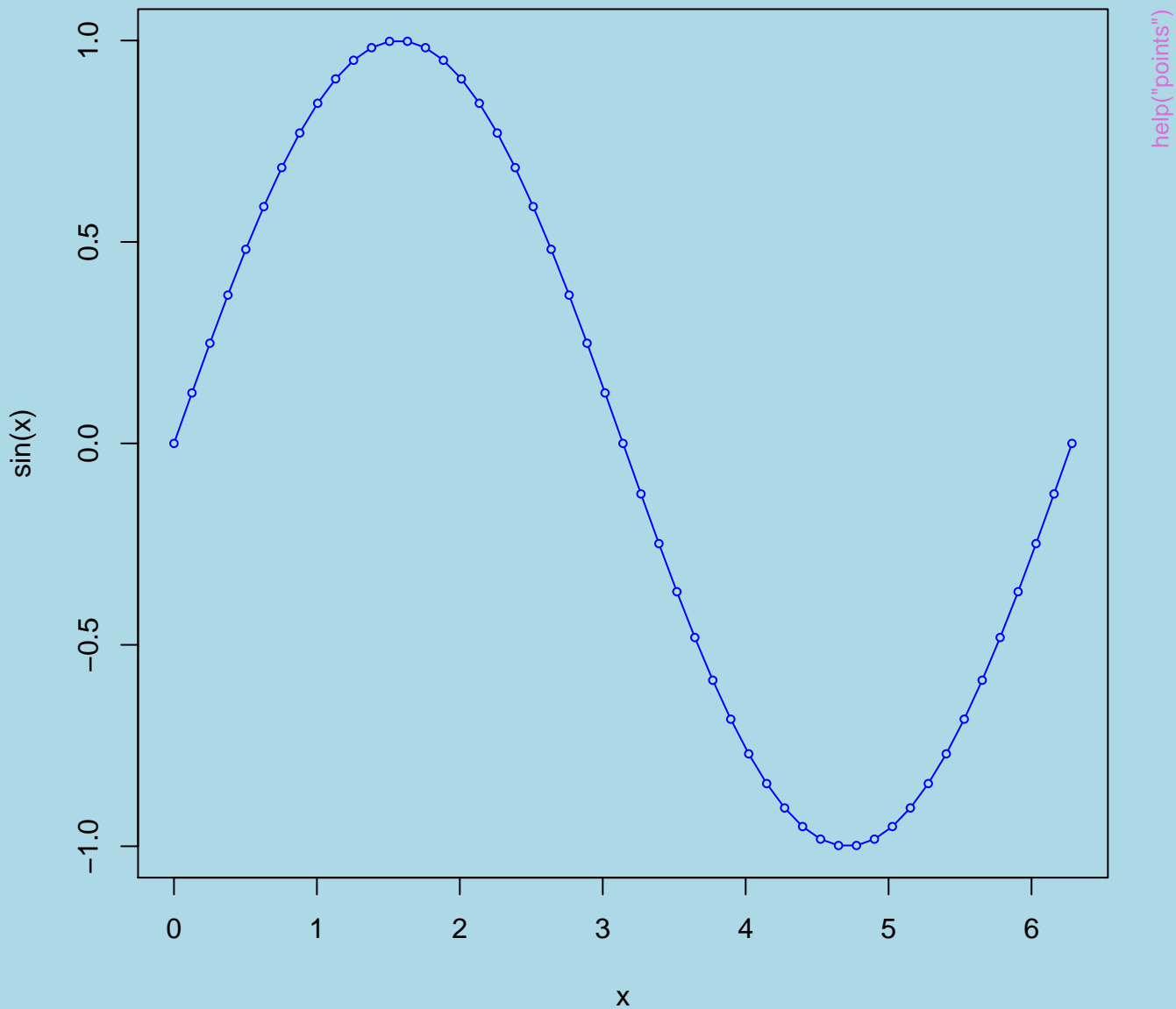


Histogram of 15 women's weights









































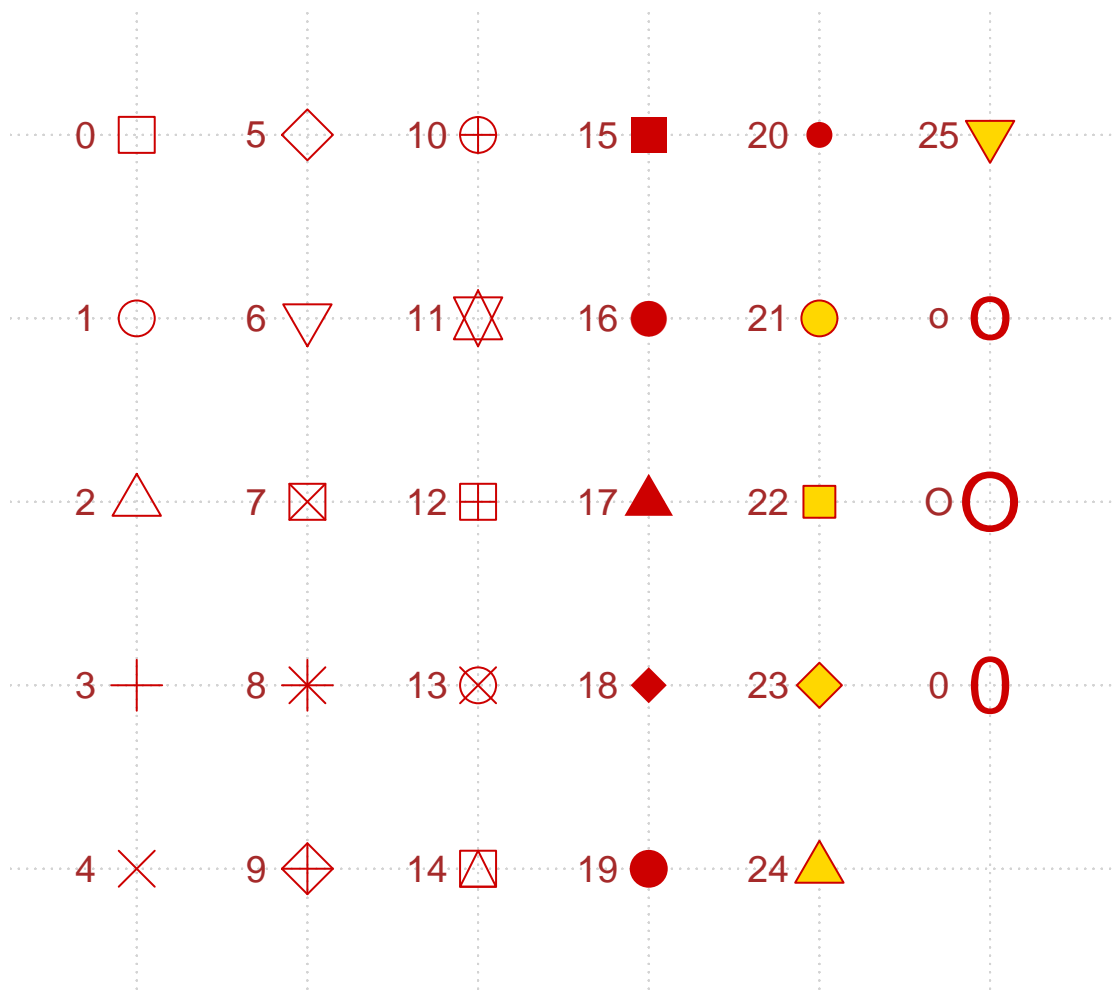
`plot(..., type="o", pch=21, bg=par("bg"))`



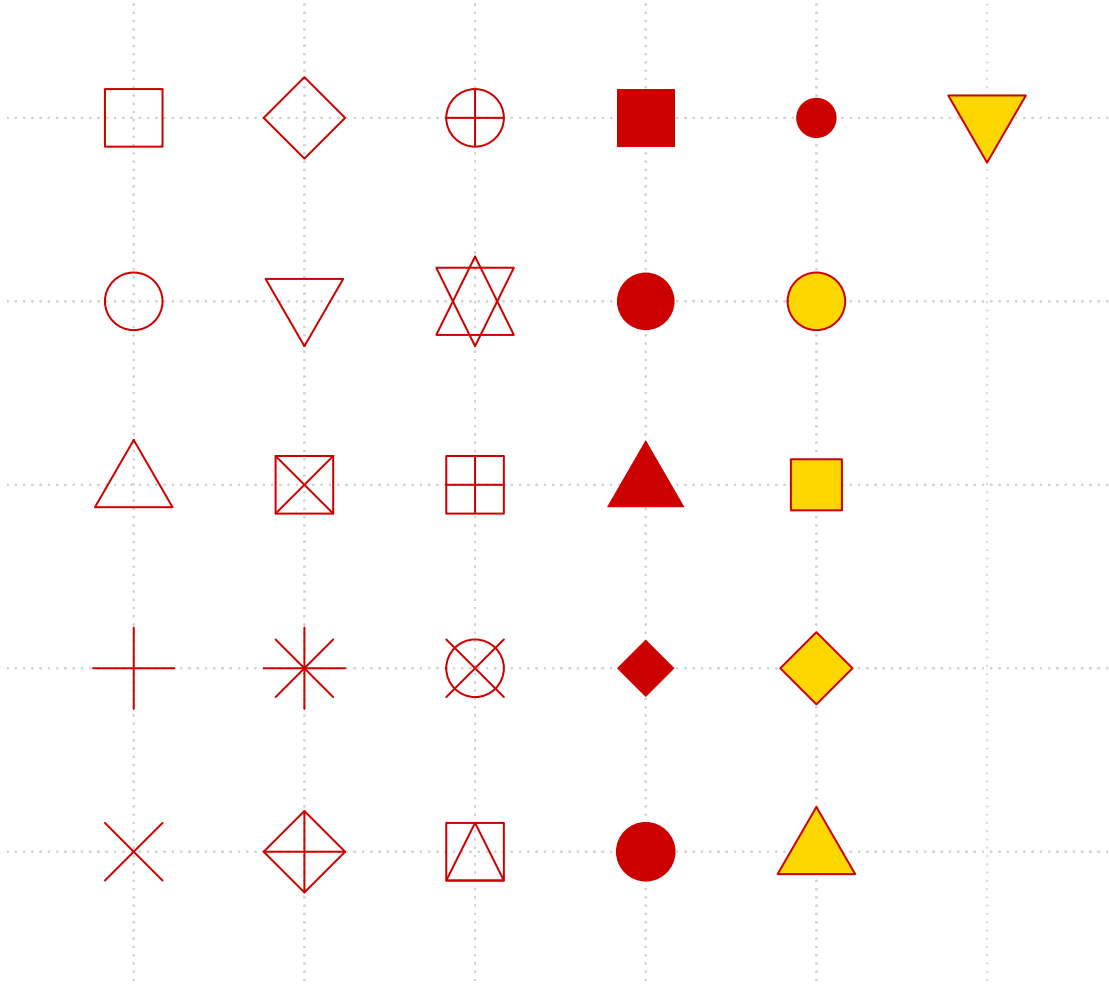
plot symbols : points (... pch = \*, cex = 3 )

0		6		12		18		24		0	
1		7		13		19		25		+	
2		8		14		20		*		-	
3		9		15		21		.			
4		10		16		22		o		%	
5		11		17		23		O		#	

plot symbols : points (... pch = \*, cex = 2.5 )

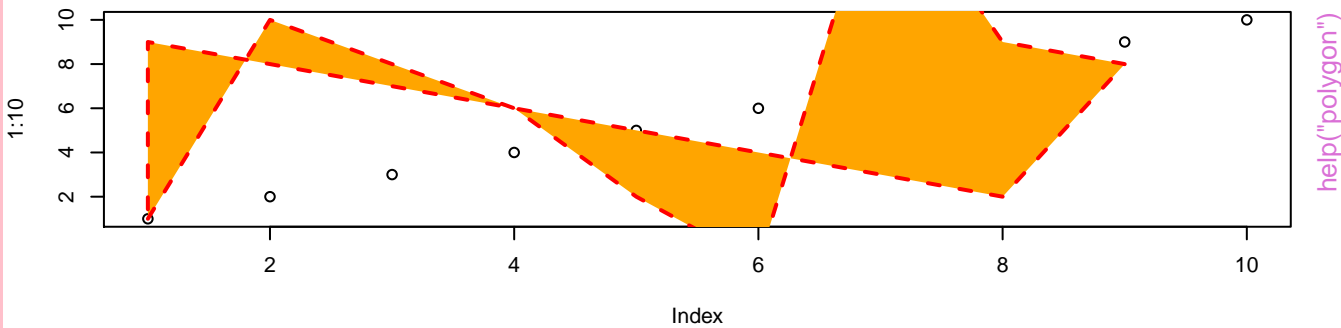


help("points")

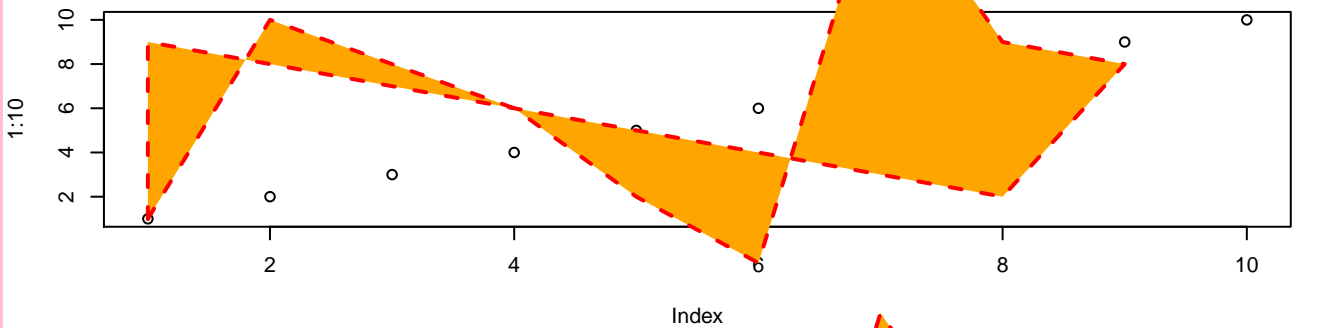


help("points")

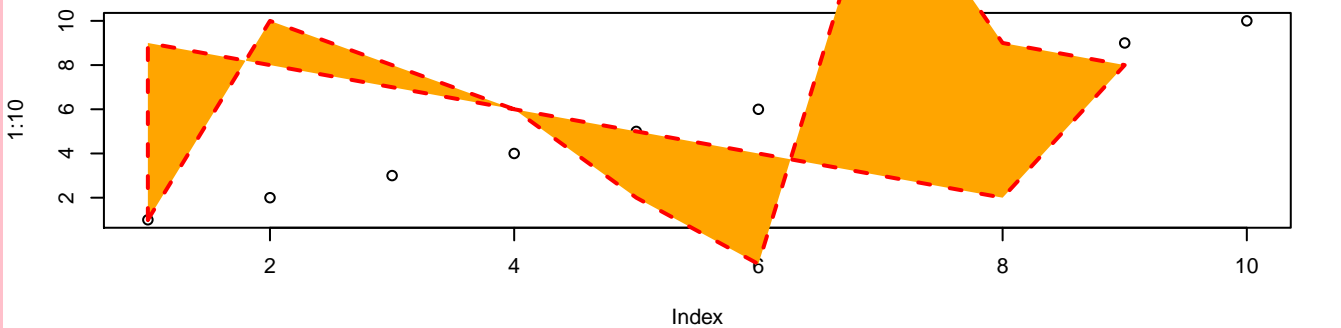
**xpd = FALSE**



**xpd = TRUE**

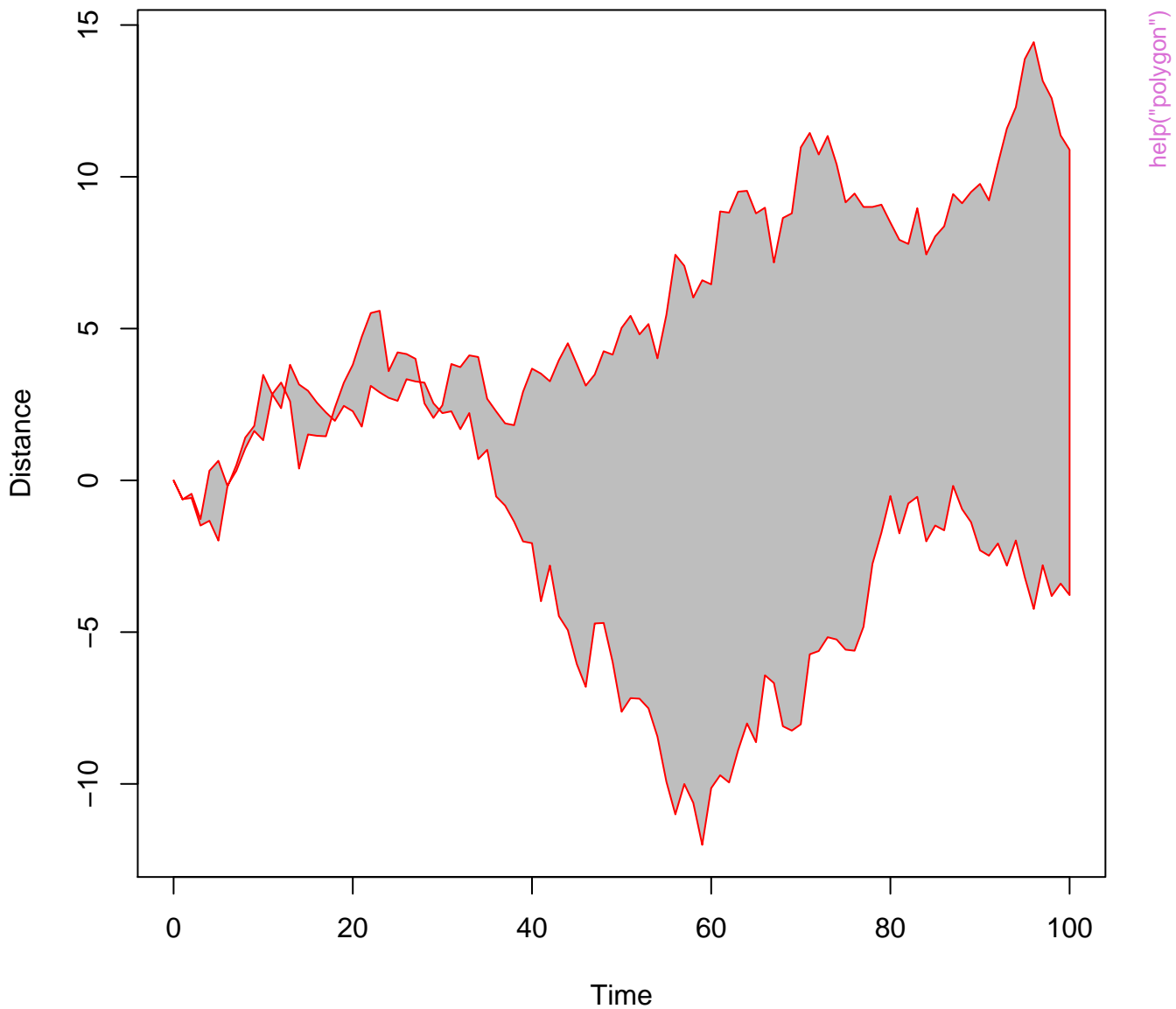


**xpd = NA**

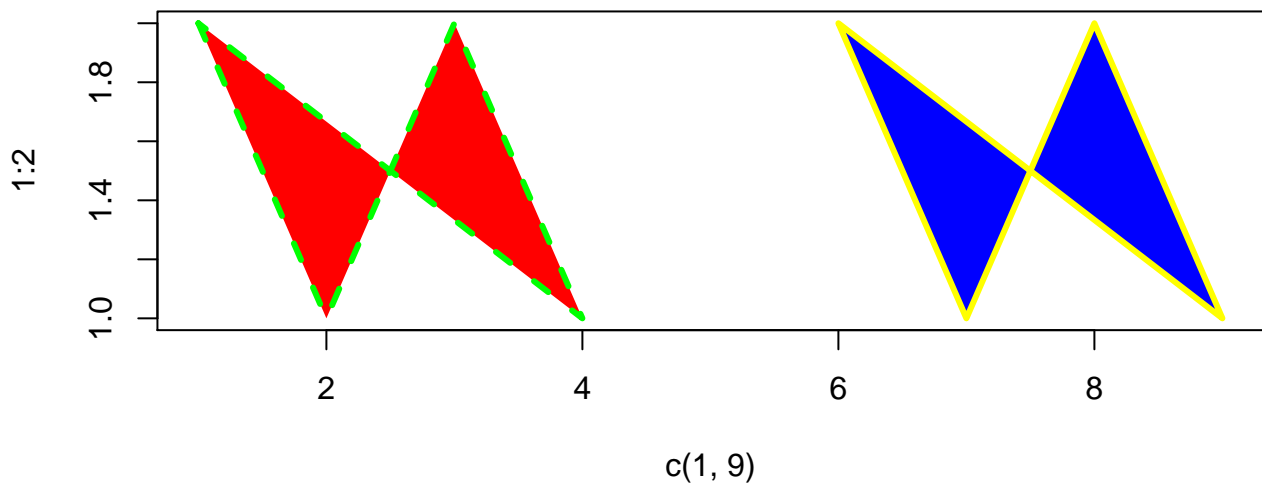
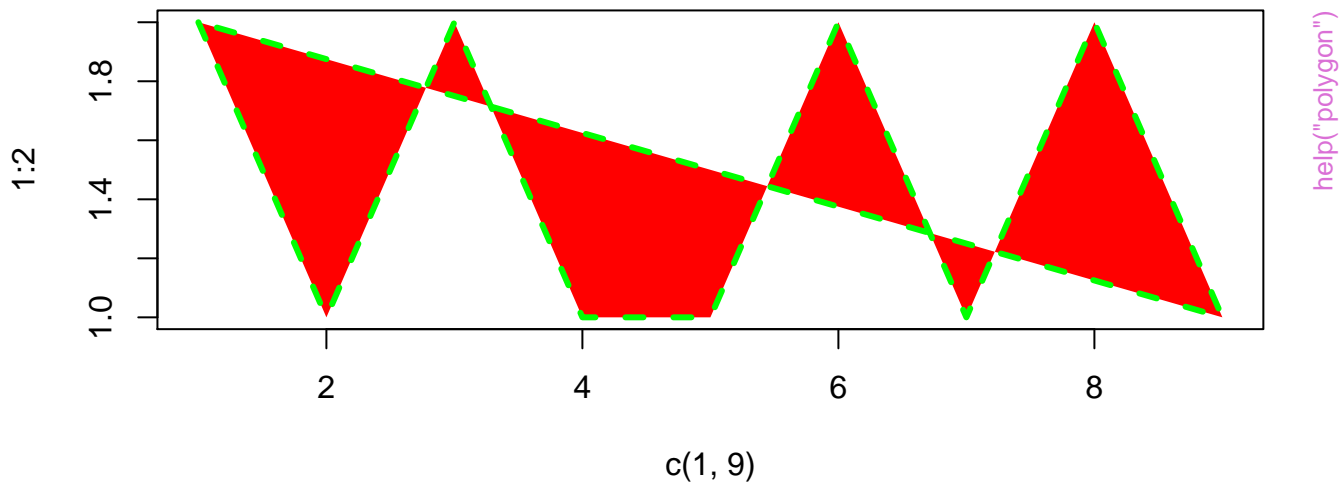


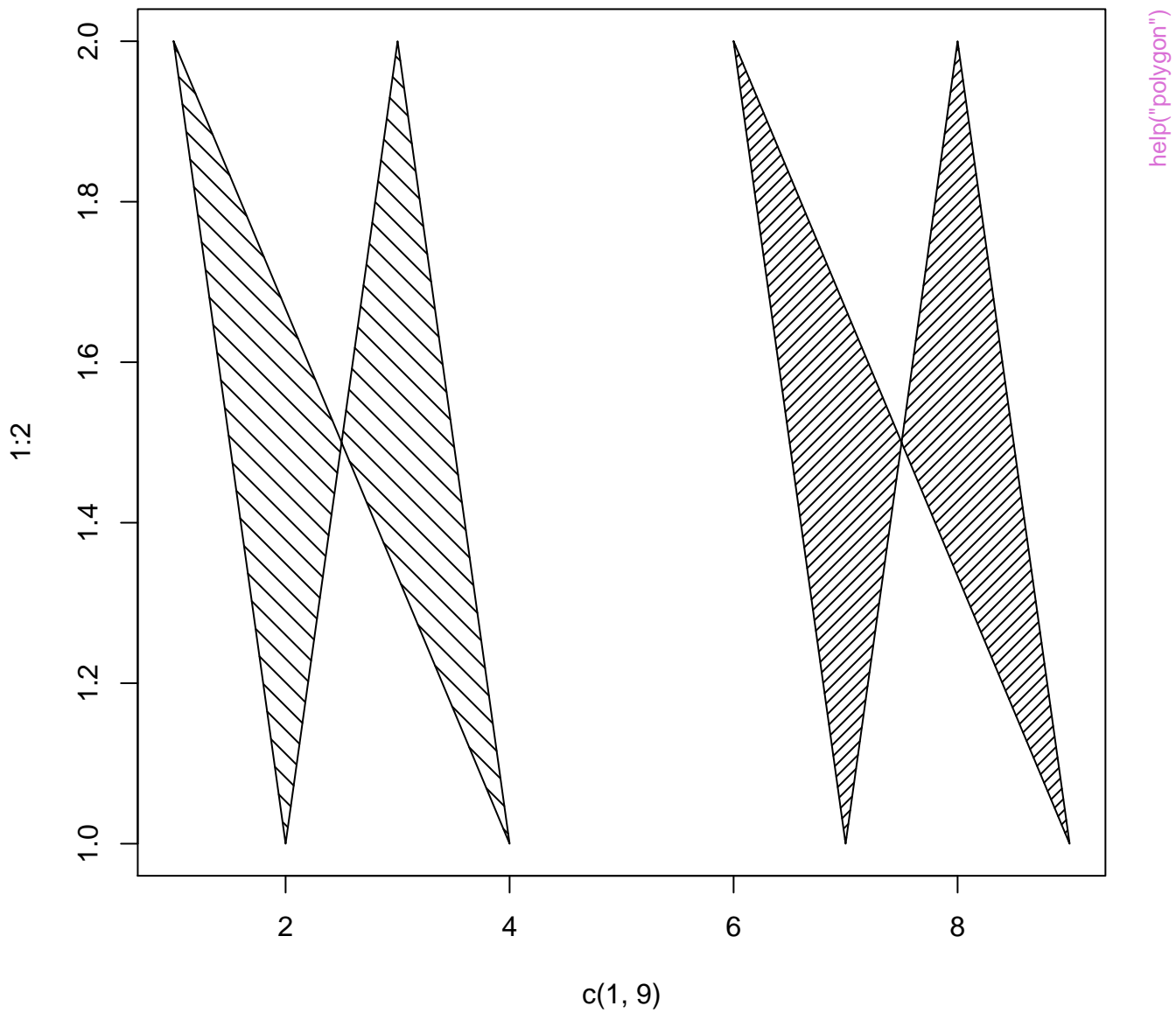
help("polygon")

## Distance Between Brownian Motions





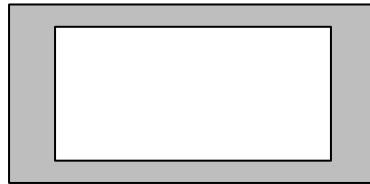




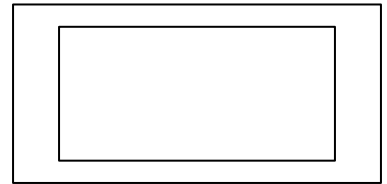
# Nested rectangles, both clockwise



Rule: winding



Rule: evenodd



help("polypath")

## Nested rectangles, outer clockwise, inner anti-clockwise



Rule: winding



Rule: evenodd



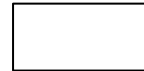
## Disjoint rectangles



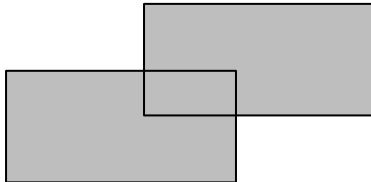
Rule: winding



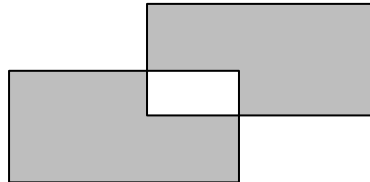
Rule: evenodd



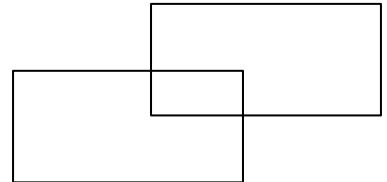
## Overlapping rectangles, both clockwise



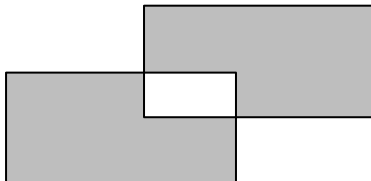
Rule: winding



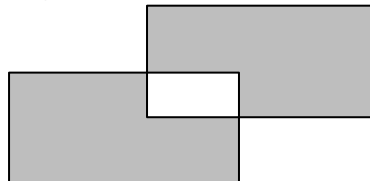
Rule: evenodd



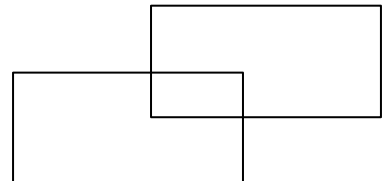
## Overlapping rectangles, one clockwise, other anti-clockwise

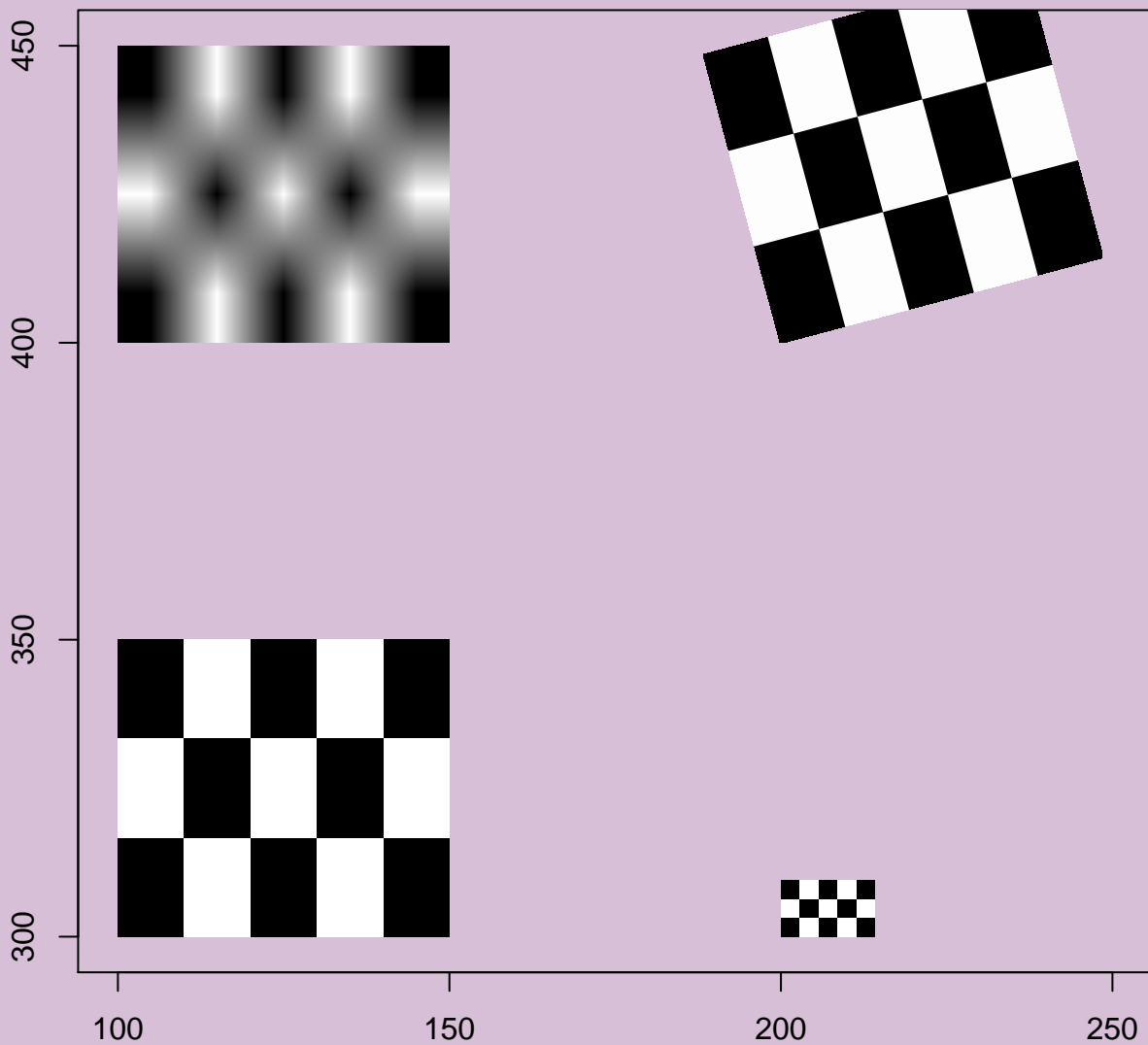


Rule: winding



Rule: evenodd

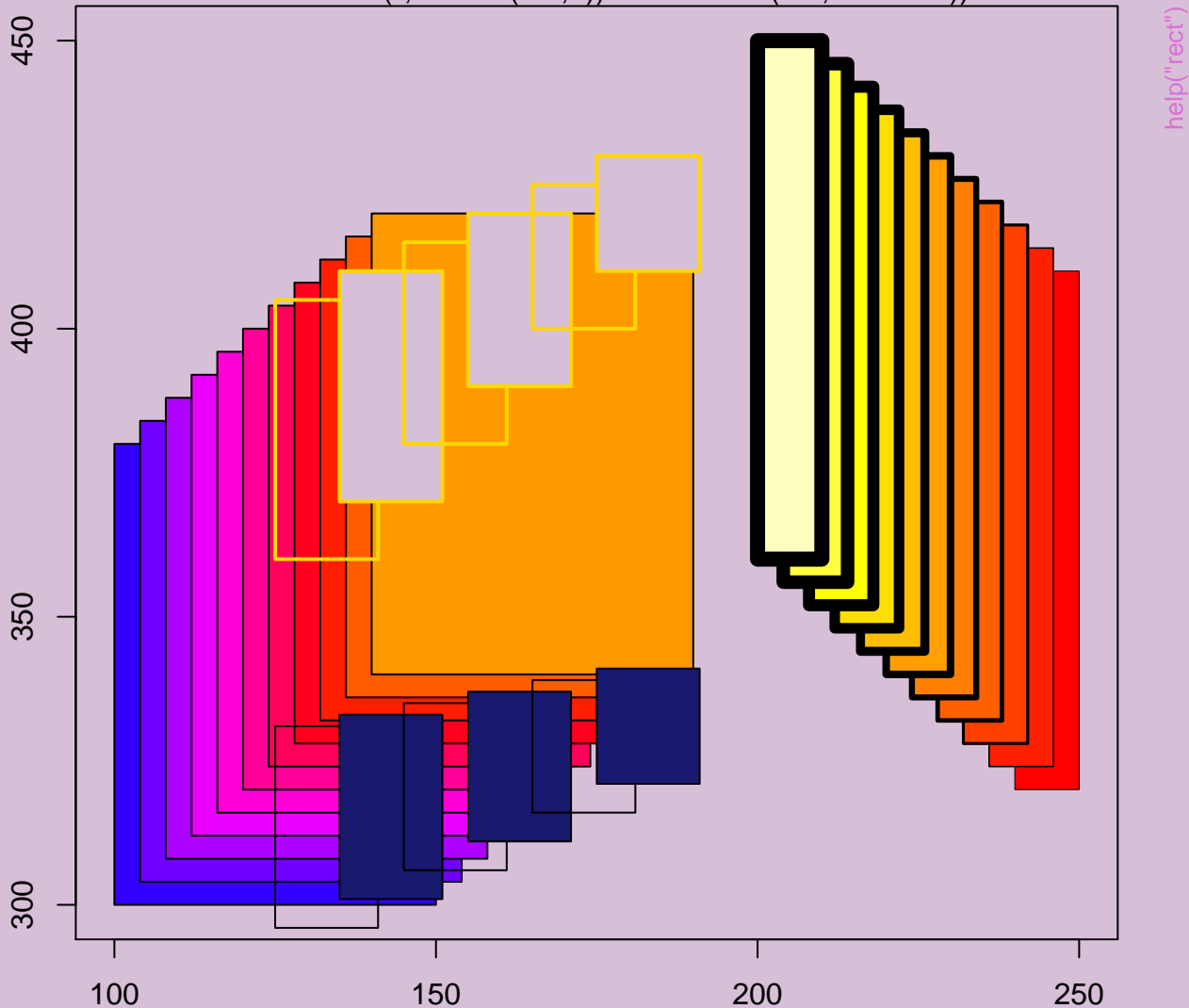


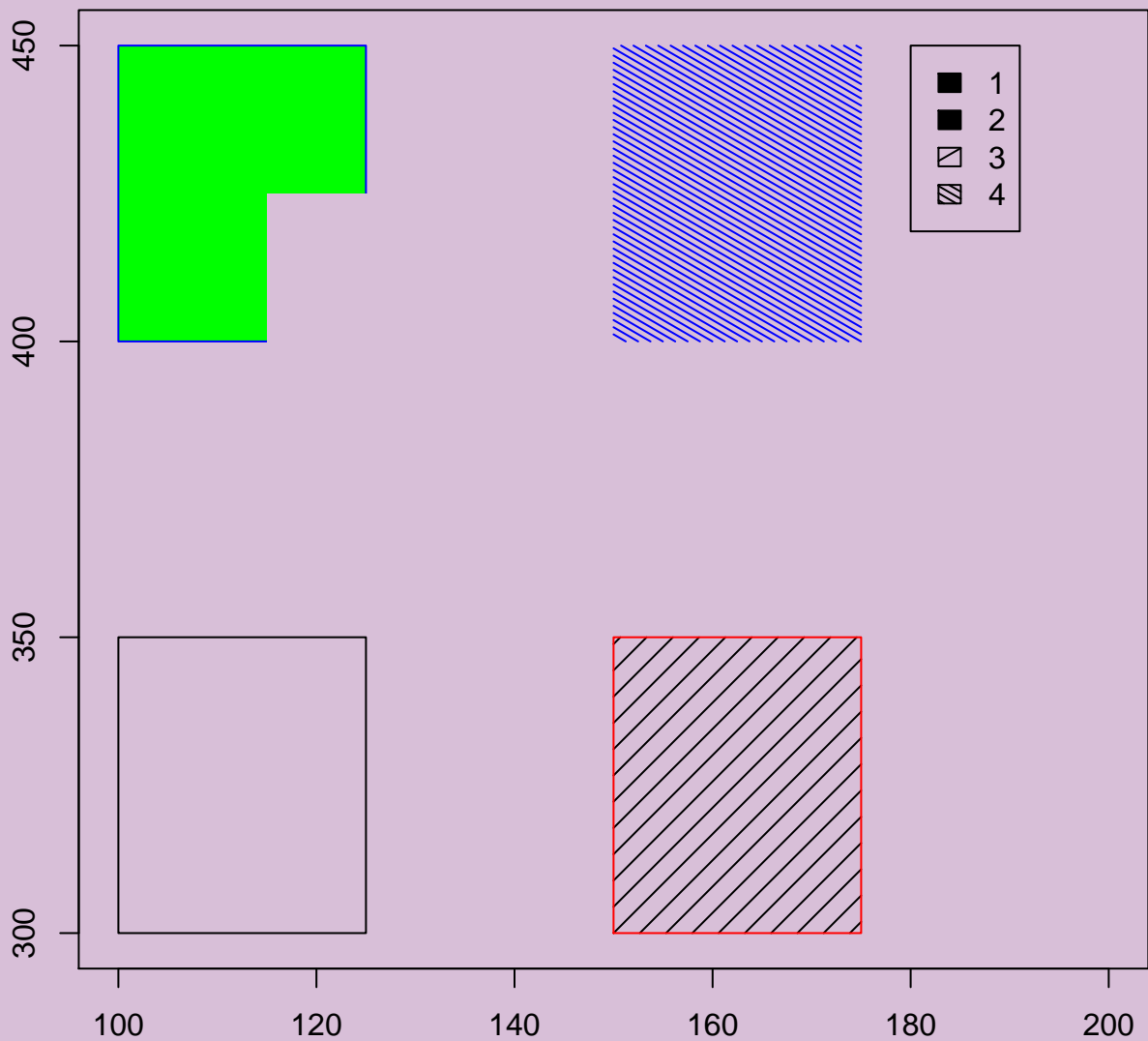


`help("rasterImage")`

**2 x 11 rectangles; 'rect(100+i,300+i, 150+i,380+i)'**

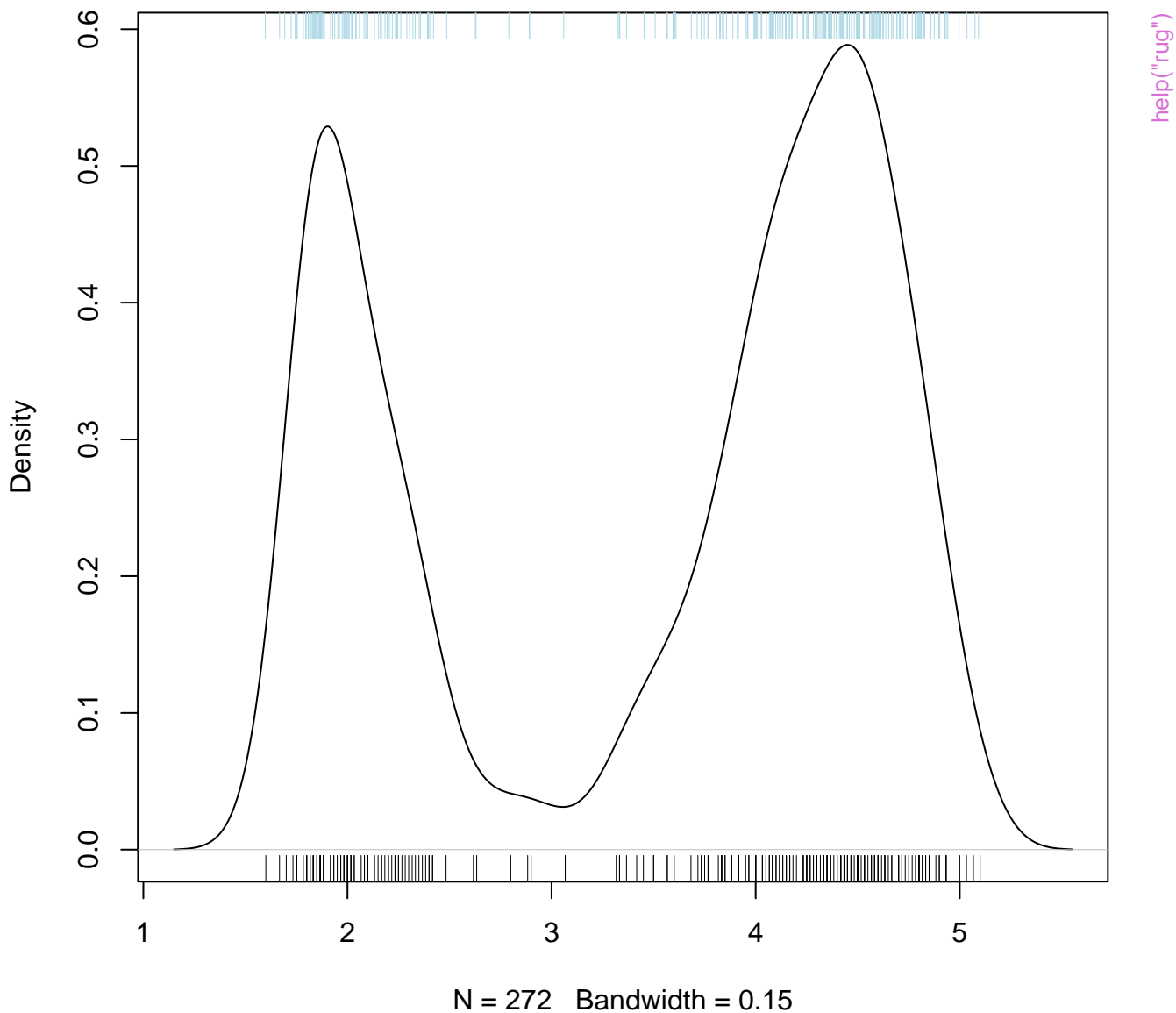
**+ 2 x 6 rect(\*, col = c(NA,0)) and col = c(NA,"m..blue")**



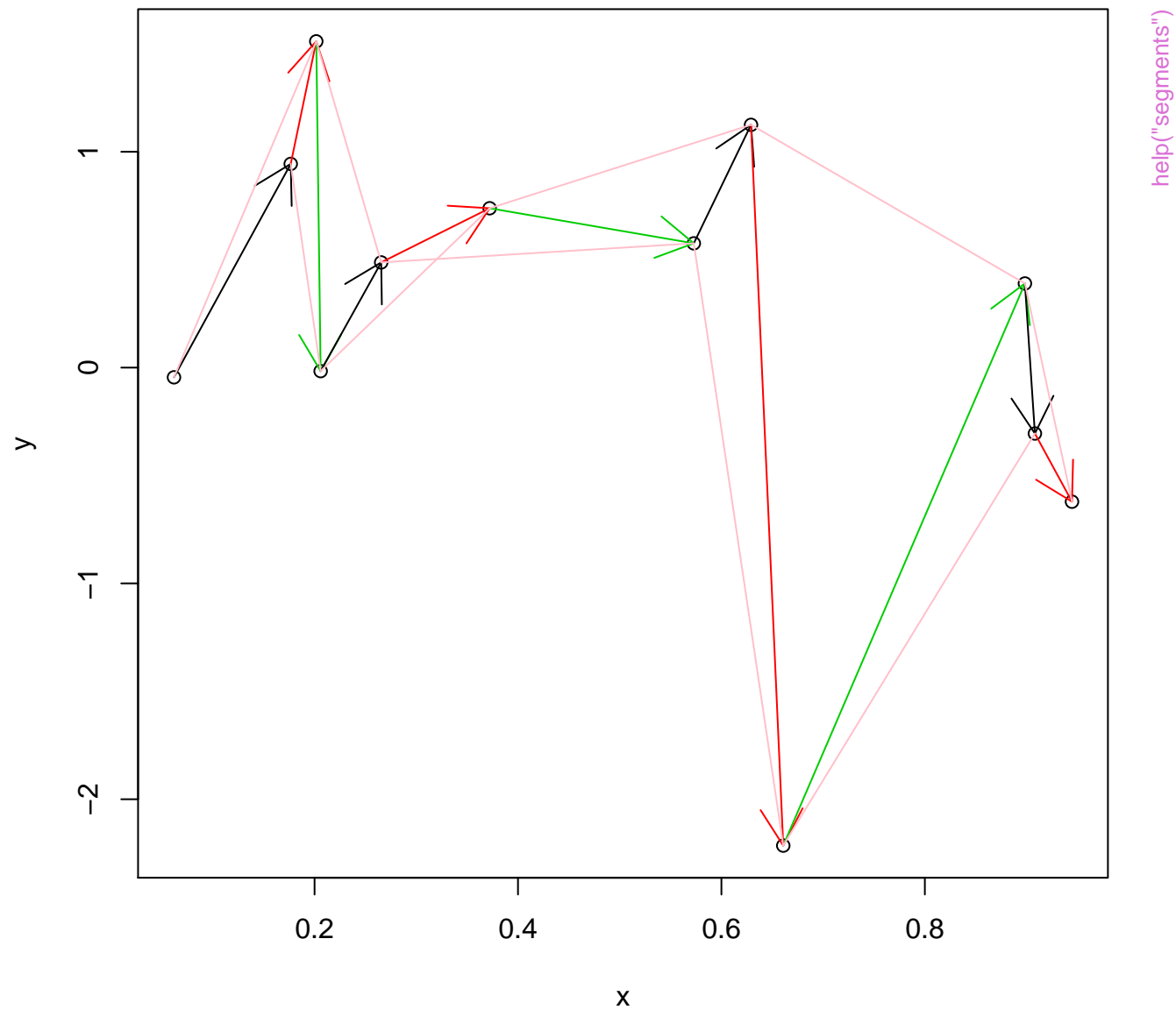


help("rect")

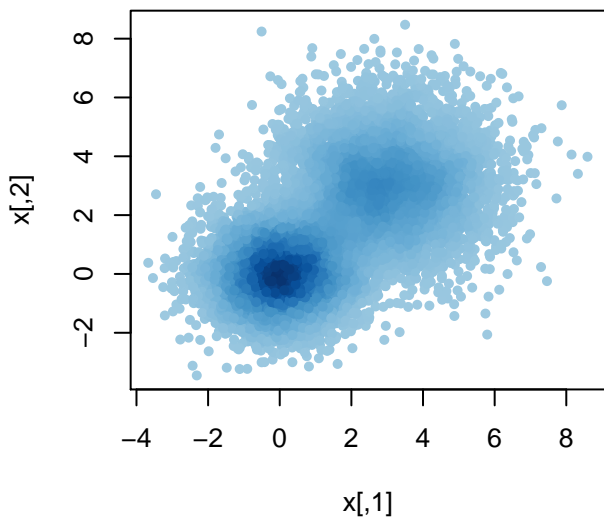
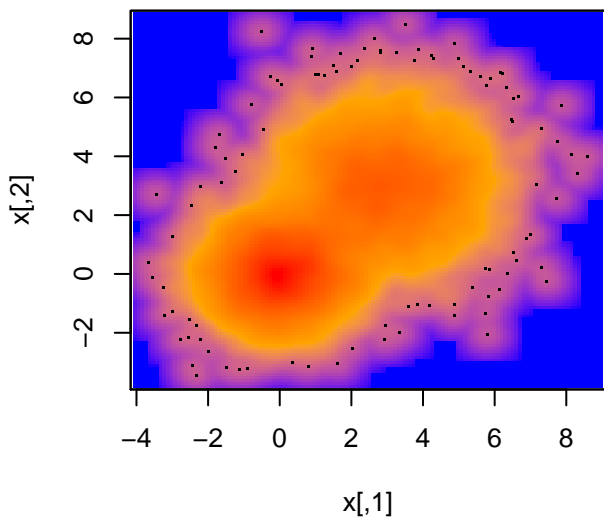
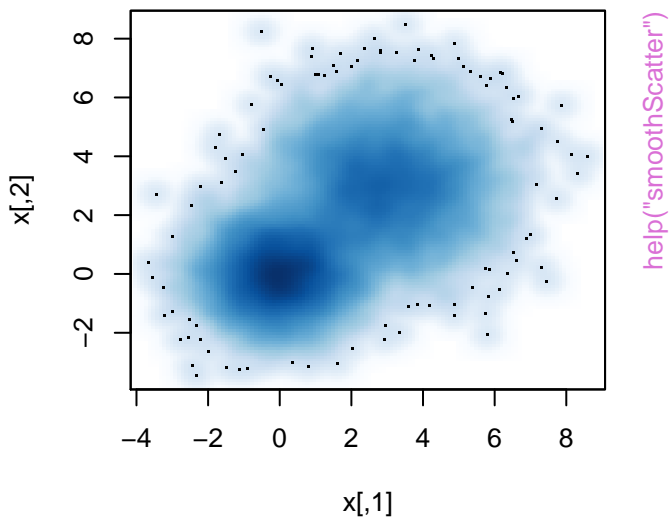
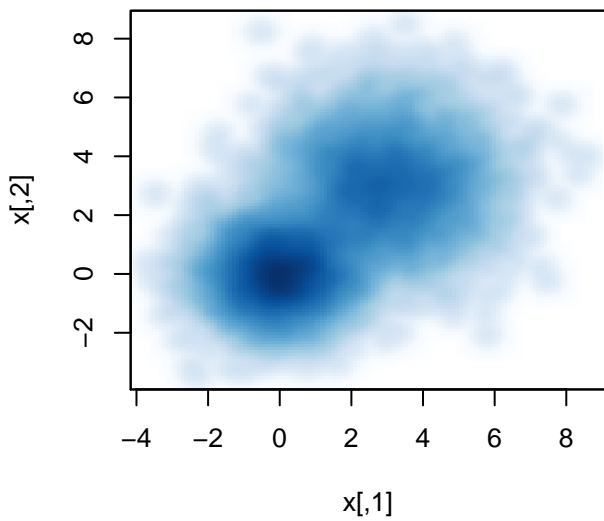
**density.default(x = eruptions, bw = 0.15)**

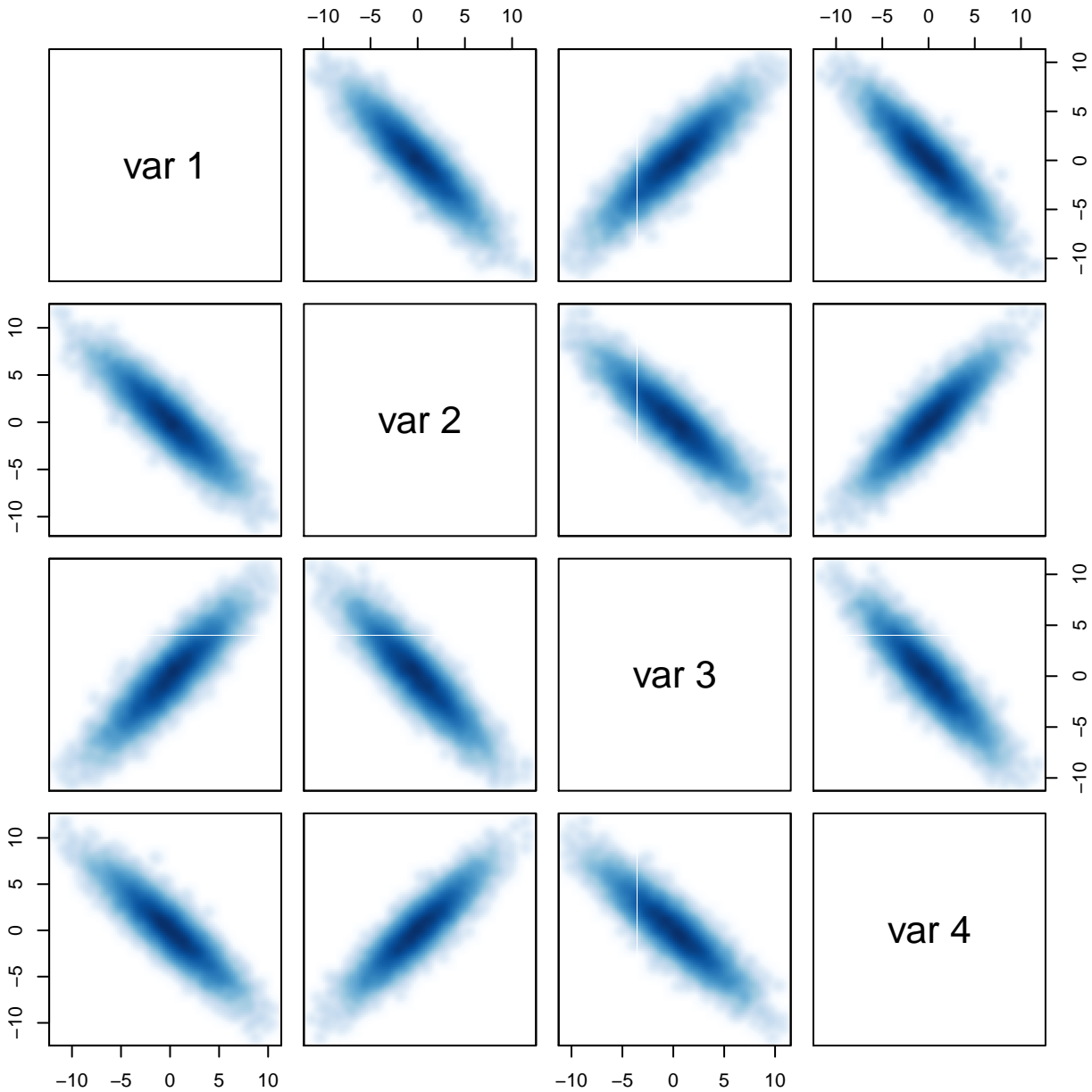


## arrows(.) and segments(.)

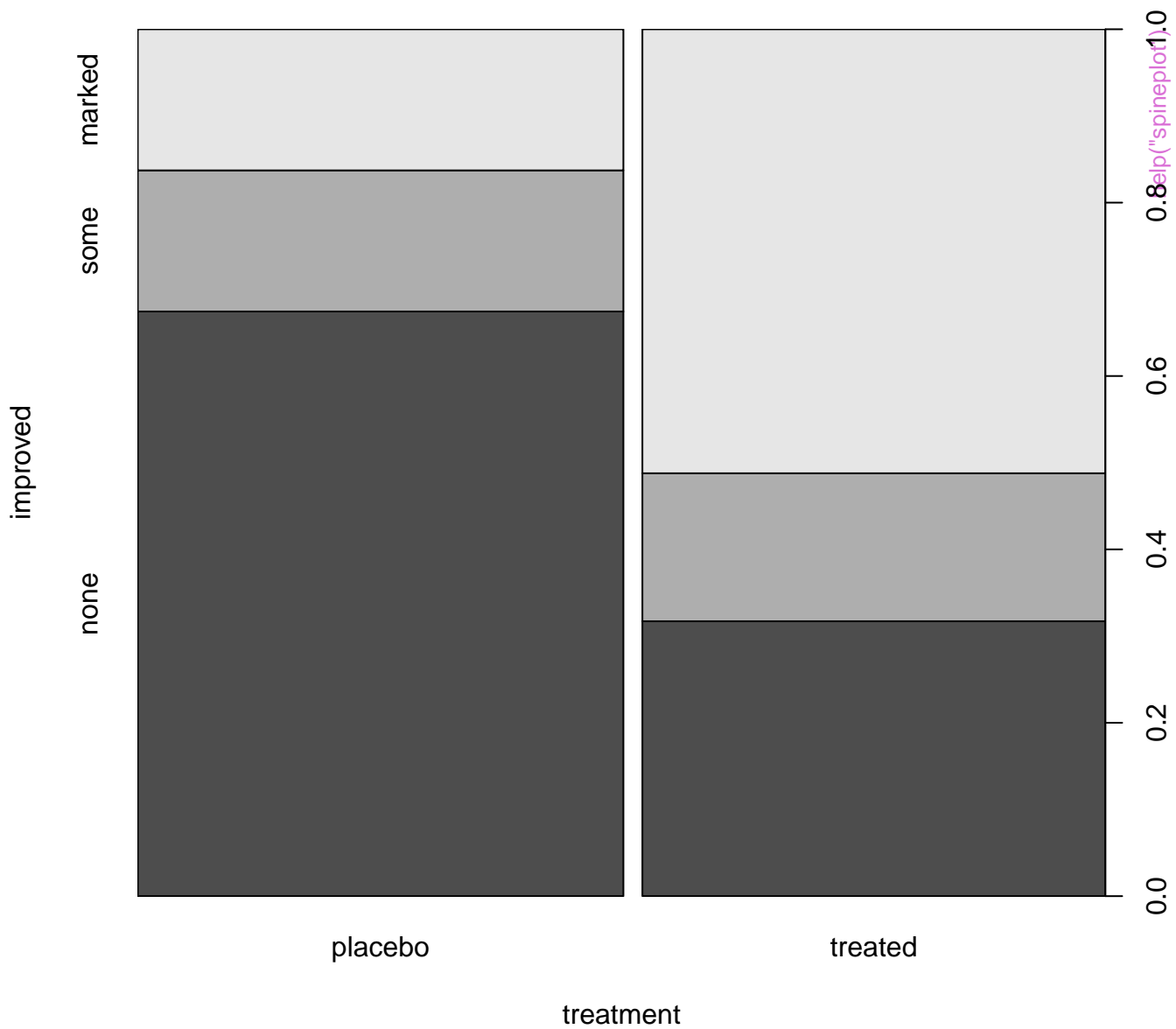




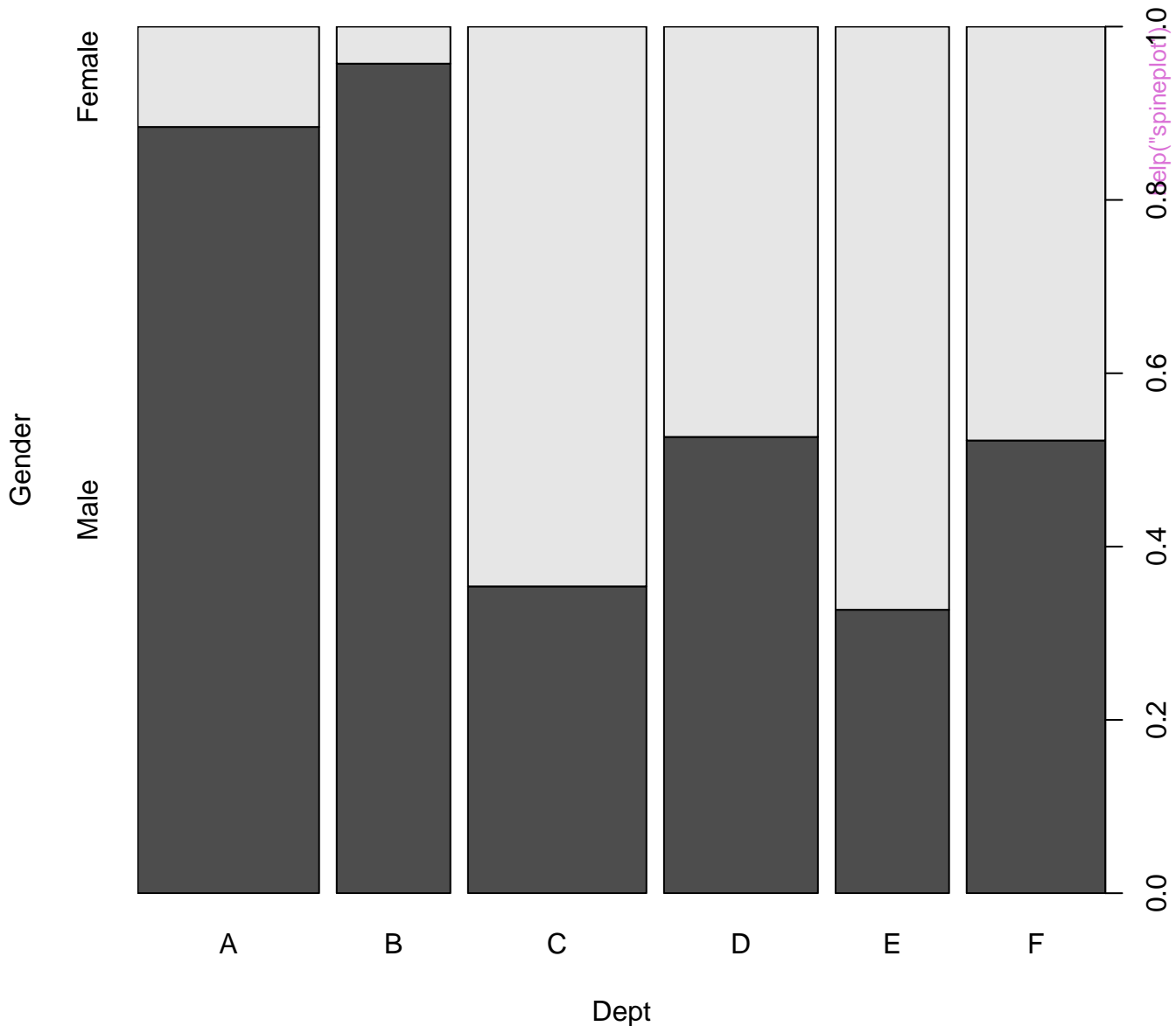




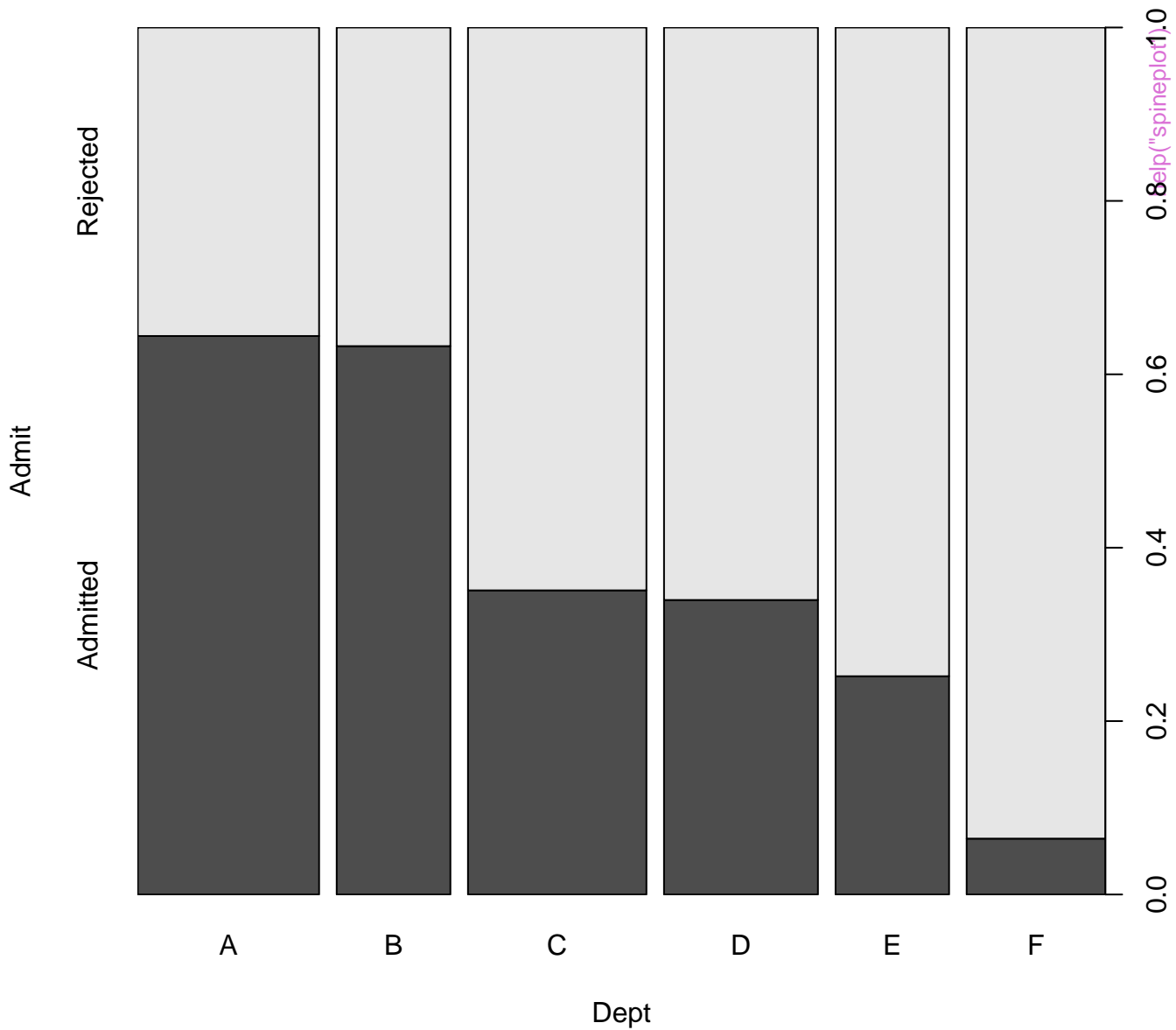
[help\("smoothScatter"\)](#)

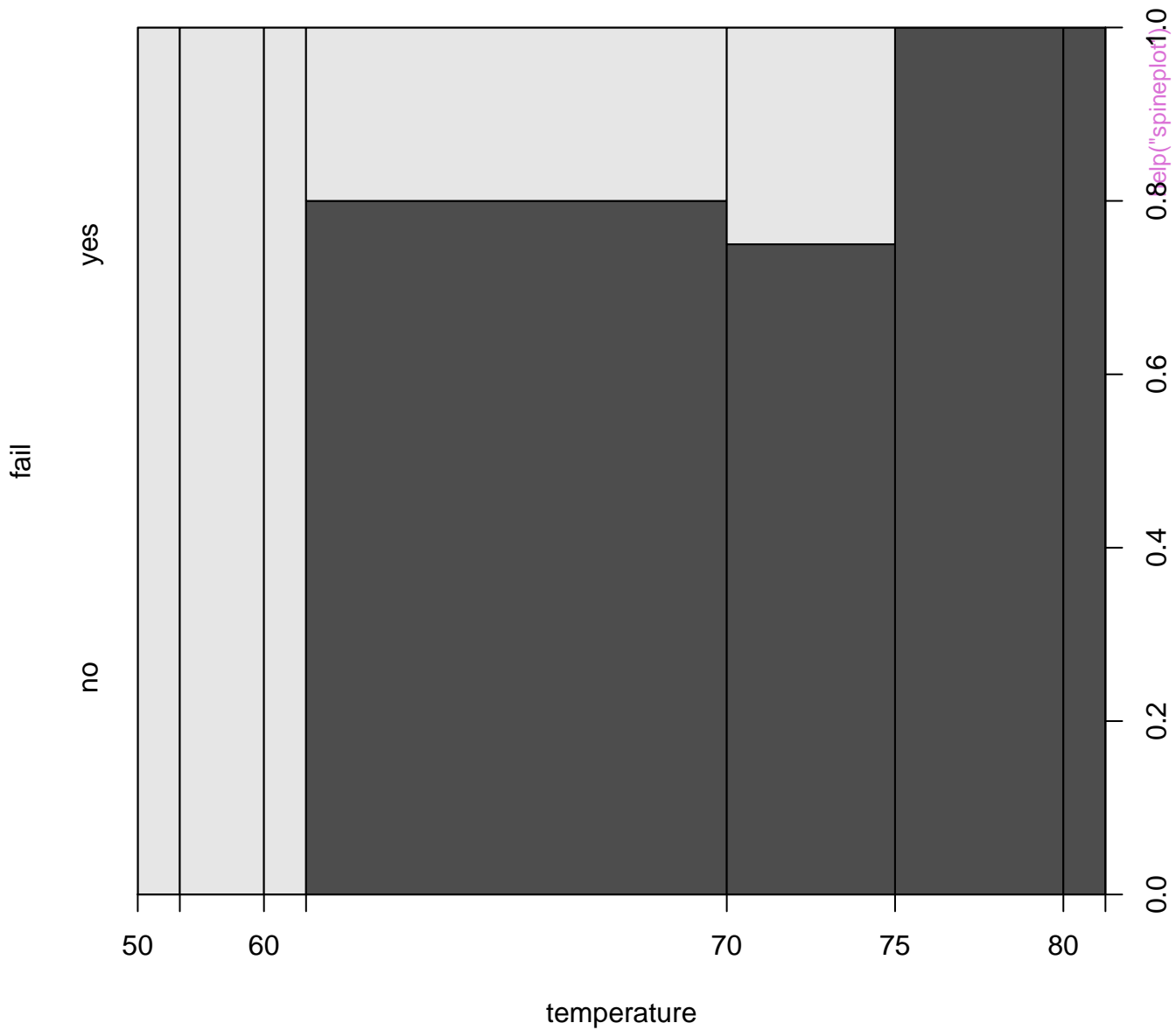


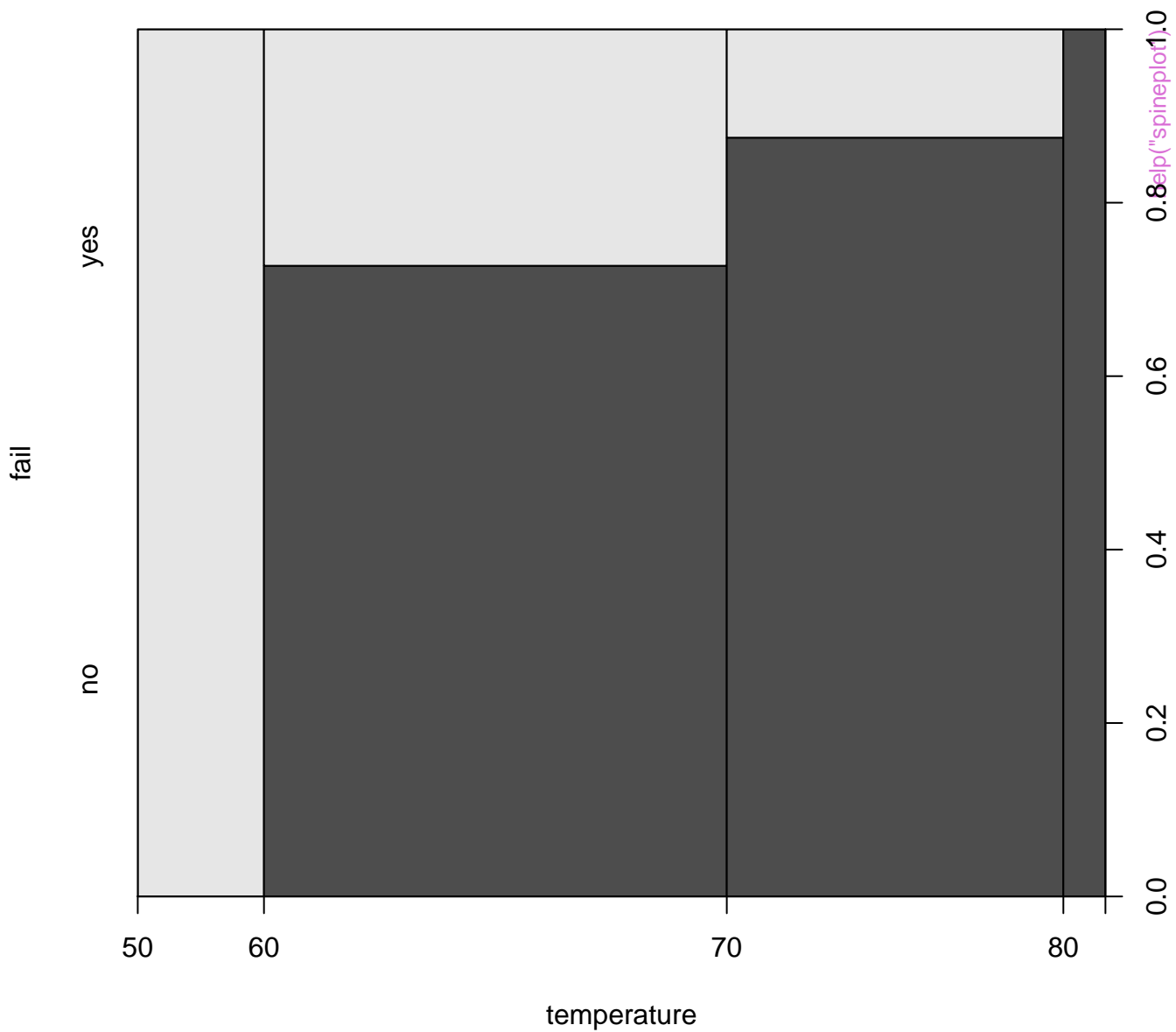
Applications at UCB

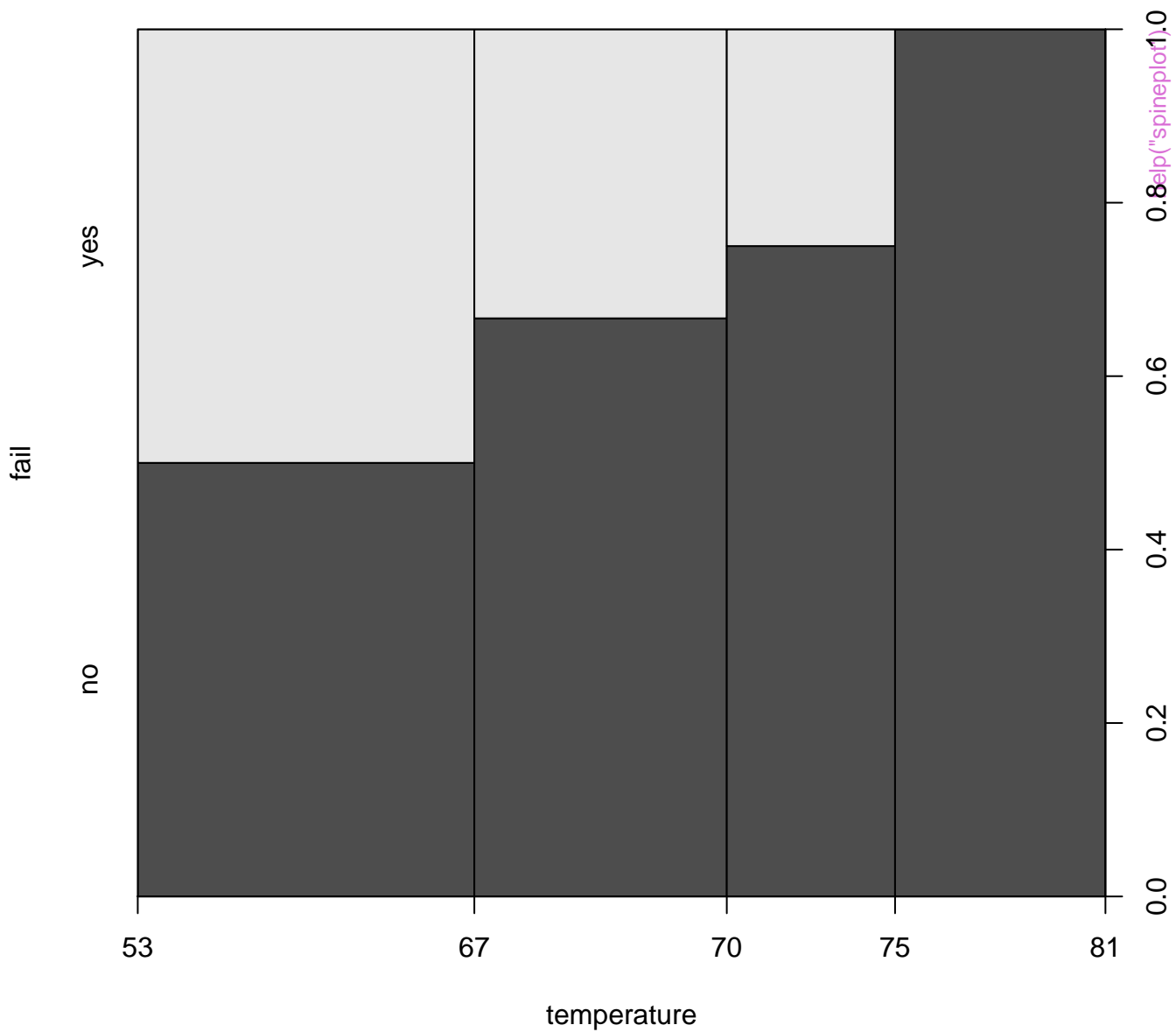


# Admissions at UCB

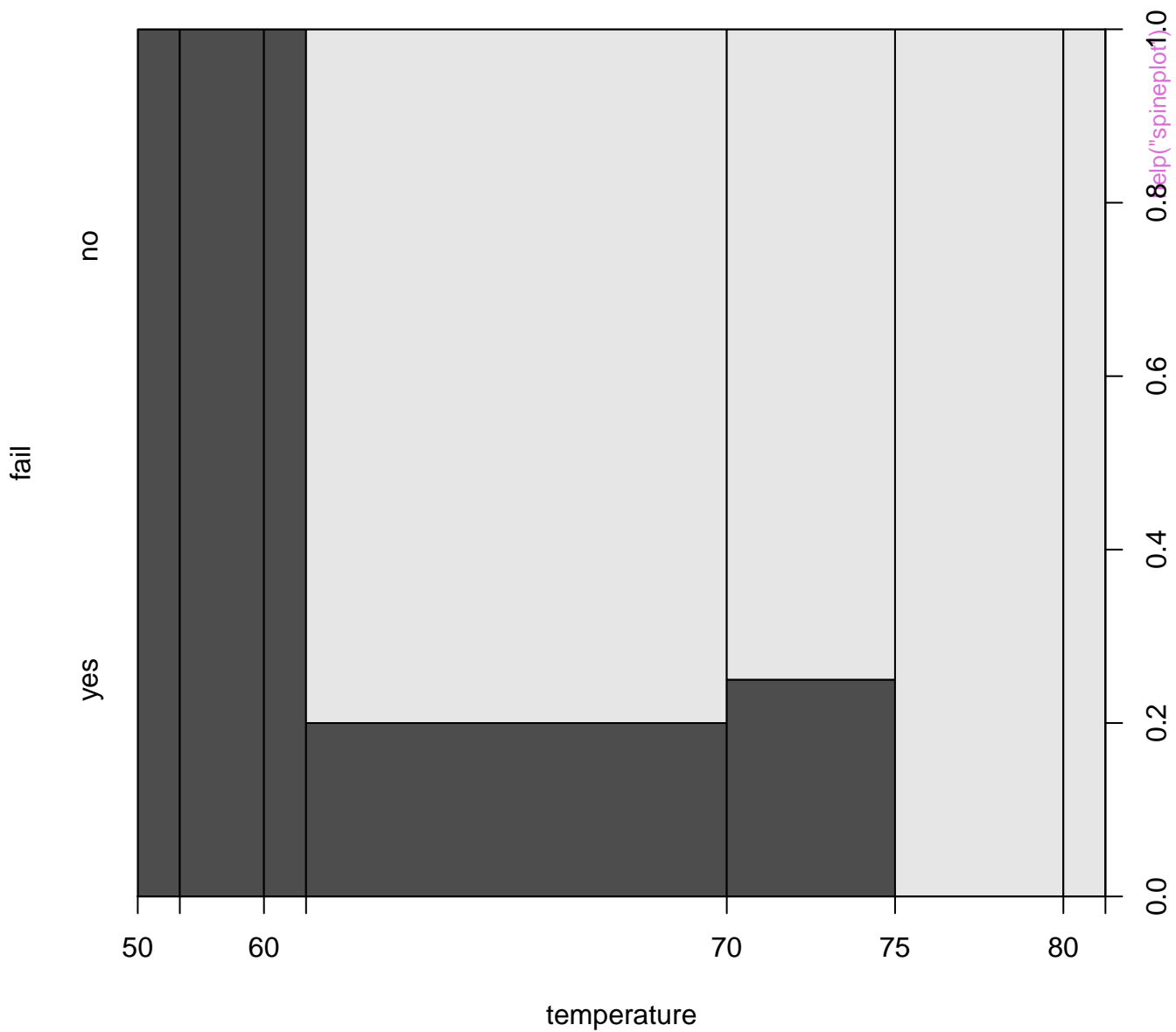












Motor Trend Cars : stars(\*, full = F)

help("stars")



Mazda RX4



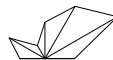
Mazda RX4 Wag



Datsun 710



Hornet 4 Drive



Hornet Sportabout



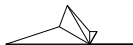
Valiant



Duster 360



Merc 240D



Merc 230



Merc 280



Merc 280C



Merc 450SE



Merc 450SL



Merc 450SLC



Cadillac Fleetwood



Lincoln Continental



Chrysler Imperial



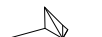
Fiat 128



Honda Civic



Toyota Corolla



Toyota Corona



Dodge Challenger



AMC Javelin



Camaro Z28



Pontiac Firebird



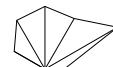
Fiat X1-9



Porsche 914-2



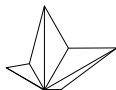
Lotus Europa



Ford Pantera L



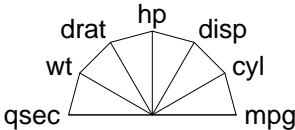
Ferrari Dino



Maserati Bora



Volvo 142E



# Motor Trend Cars : full stars()

help("stars")



Mazda RX4



Mazda RX4 Wag



Datsun 710



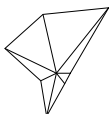
Hornet 4 Drive



Hornet Sportabout



Valiant



Duster 360



Merc 240D



Merc 230



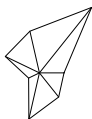
Merc 280



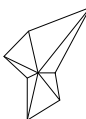
Merc 280C



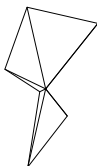
Merc 450SE



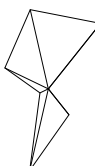
Merc 450SL



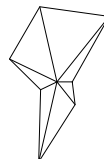
Merc 450SLC



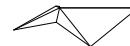
Cadillac Fleetwood



Lincoln Continental



Chrysler Imperial



Fiat 128



Honda Civic



Toyota Corolla



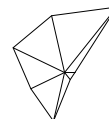
Toyota Corona



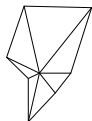
Dodge Challenger



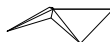
AMC Javelin



Camaro Z28



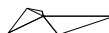
Pontiac Firebird



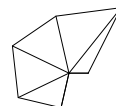
Fiat X1-9



Porsche 914-2



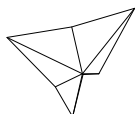
Lotus Europa



Ford Pantera L



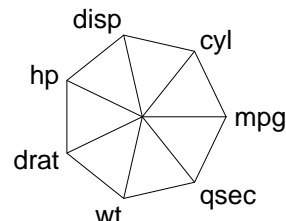
Ferrari Dino



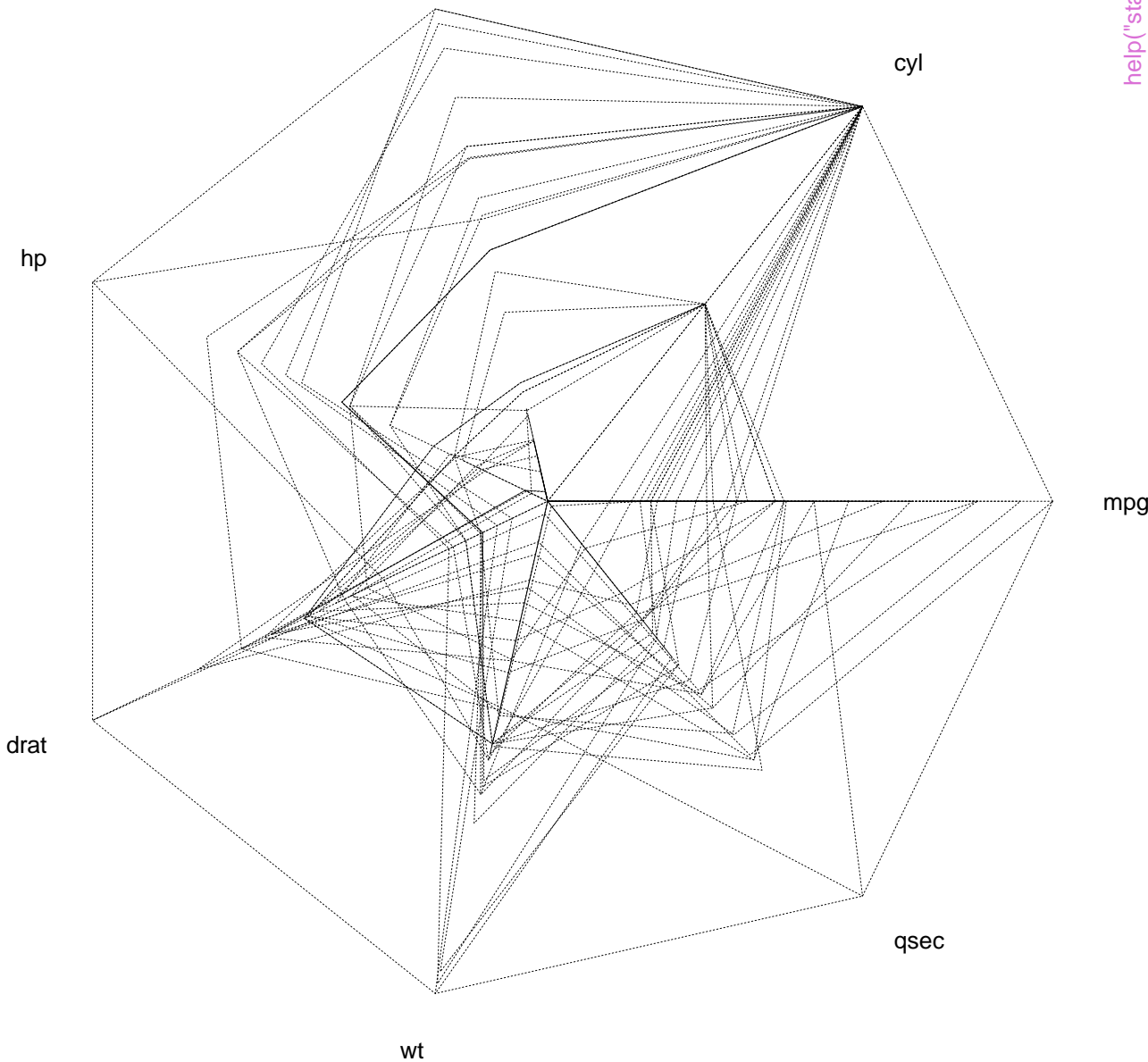
Maserati Bora



Volvo 142E

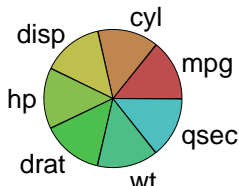
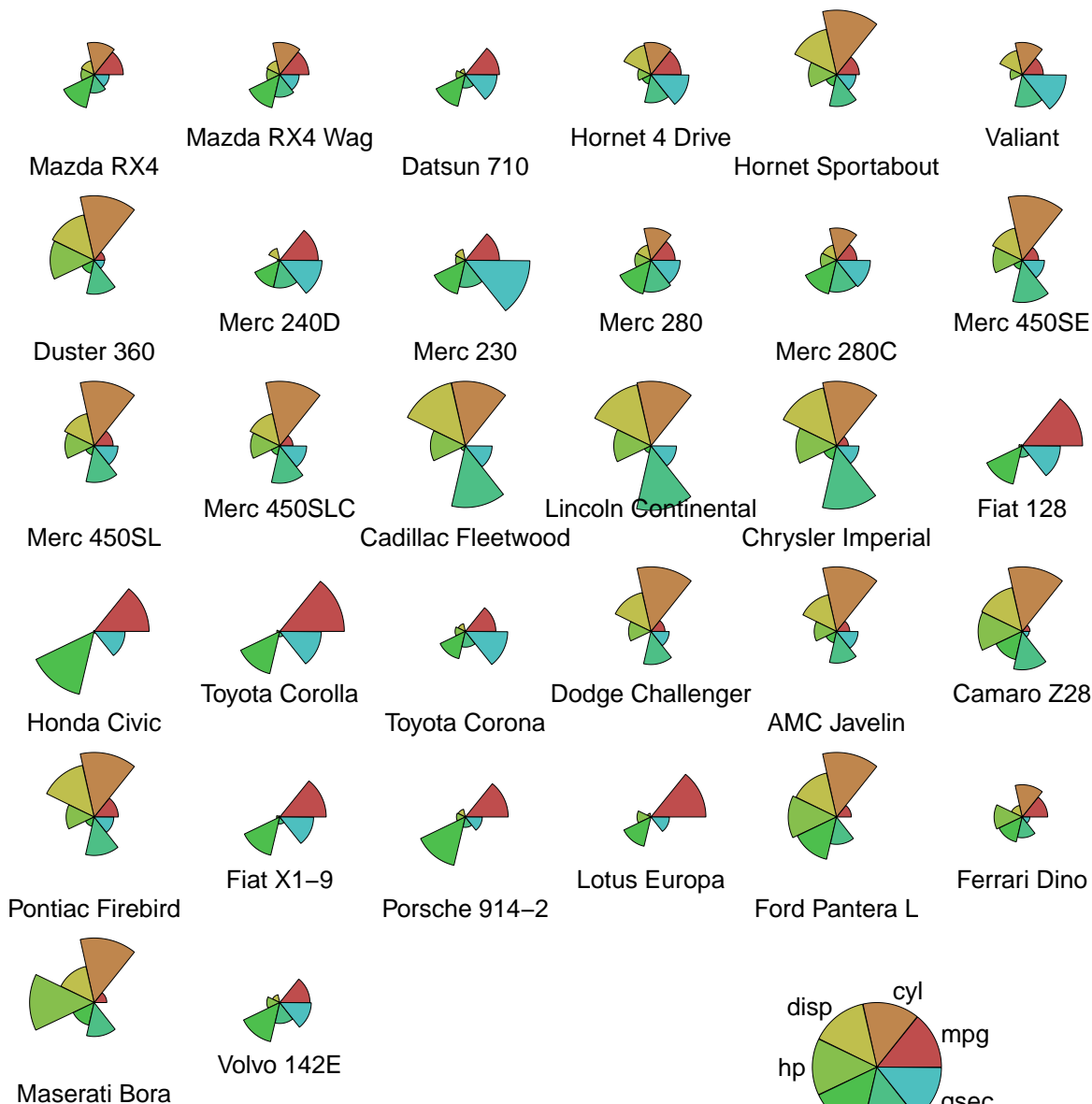


## disp

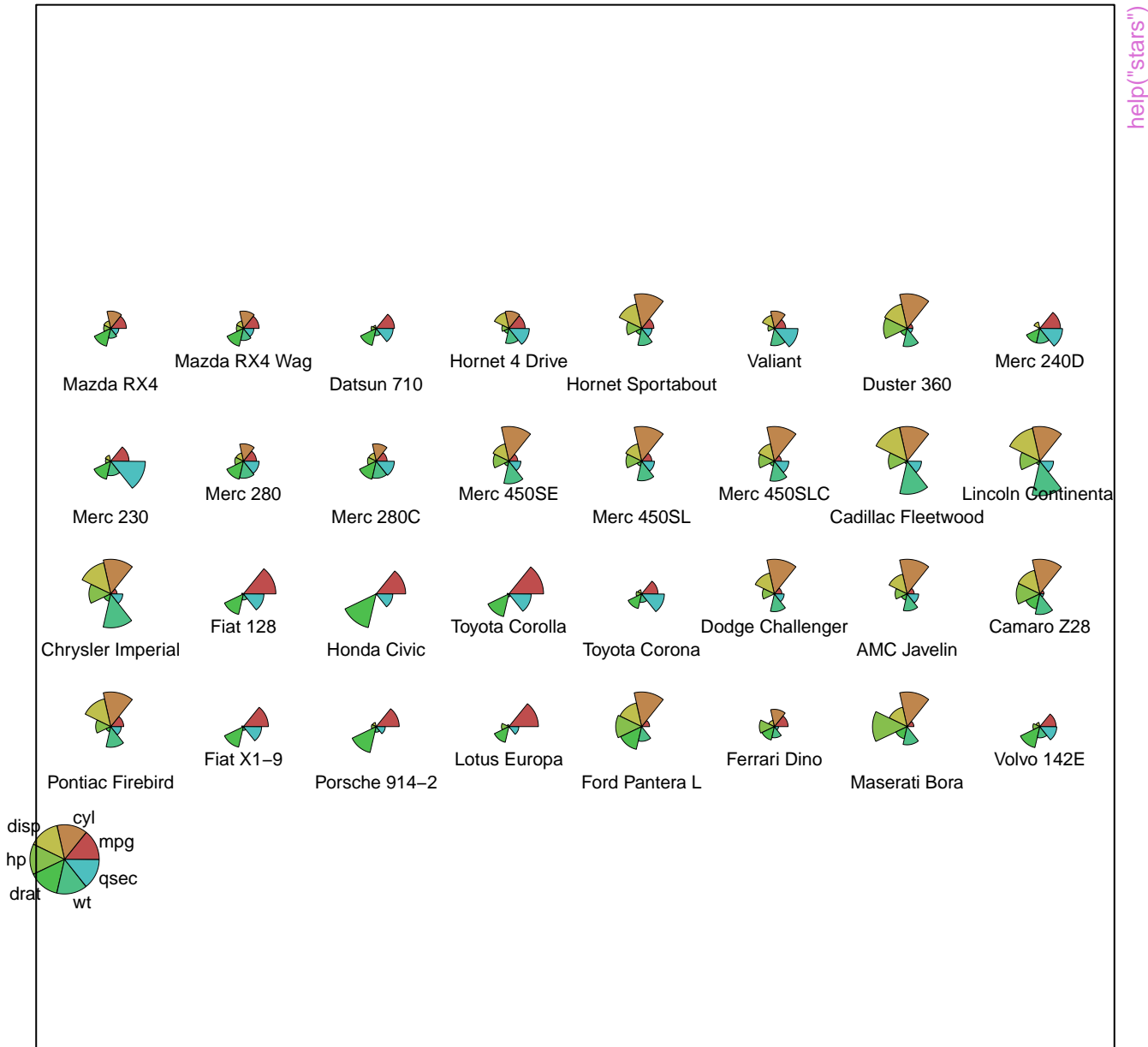


# Motor Trend Cars

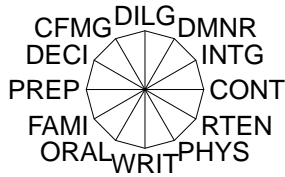
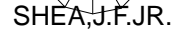
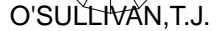
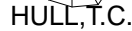
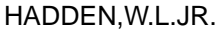
help("stars")

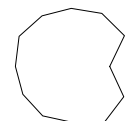


# Motor Trend Cars



# help("stars")

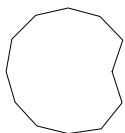




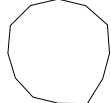
AARONSO



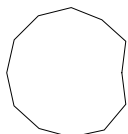
COHEN



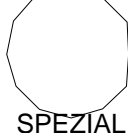
HADDEN



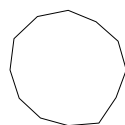
MCGRATH



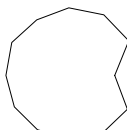
PASKEY



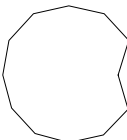
SPEZIAL



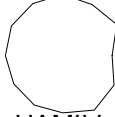
ZARRILL



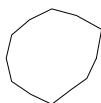
ALEXAND



DALY



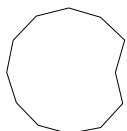
HAMILL



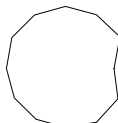
MIGNONE



RUBINOW



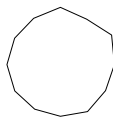
SPONZO



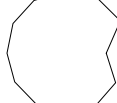
ARMENTA



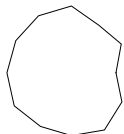
DANNEHY



HEALEY



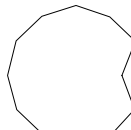
MISSAL



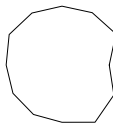
SADEN



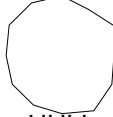
STAPLET



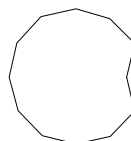
BERDON



DEAN



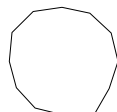
HULL



MULVEY



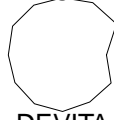
SATANIE



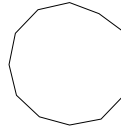
TESTO



BRACKEN



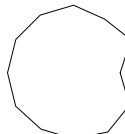
DEVITA



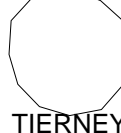
LEVINE



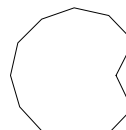
NARUK



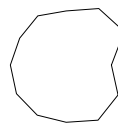
SHEA



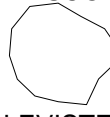
TIERNEY



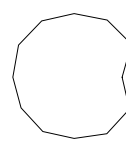
BURNS



DRISCOL



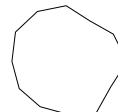
LEVISTE



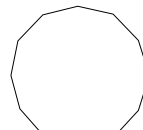
O'BRIEN



SHEA



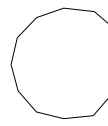
WALL



CALLAHA



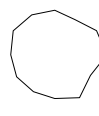
GRILLO



MARTIN



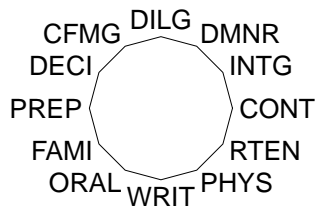
O'SULLI



SIDOR



WRIGHT



CFMG

DECI

PREP

FAMI

ORAL

DILG

DMNR

INTG

CONT

RTEN

PHYS

WRIT



# Judge not ...

help("stars")

AARONSO

ALEXAND

ARMENTA

BERDON

BRACKEN

BURNS

CALLAHA

COHEN

DALY

DANNEHY

DEAN

DEVITA

DRISCOL

GRILLO

HADDEN

HAMILL

HEALEY

HULL

LEVINE

LEVISTE

MARTIN

MCGRATH

MIGNONE

MISSAL

MULVEY

NARUK

O'BRIEN

O'SULLI

PASKEY

RUBINOW

SADEN

SATANIE

SHEA

SHEA

SIDOR

SPEZIAL

SPONZO

STAPLET

TESTO

TIERNEY

WALL

WRIGHT

ZARRILL

CFMG

DILG

DMNR

DECI

INTG

PREP

CONT

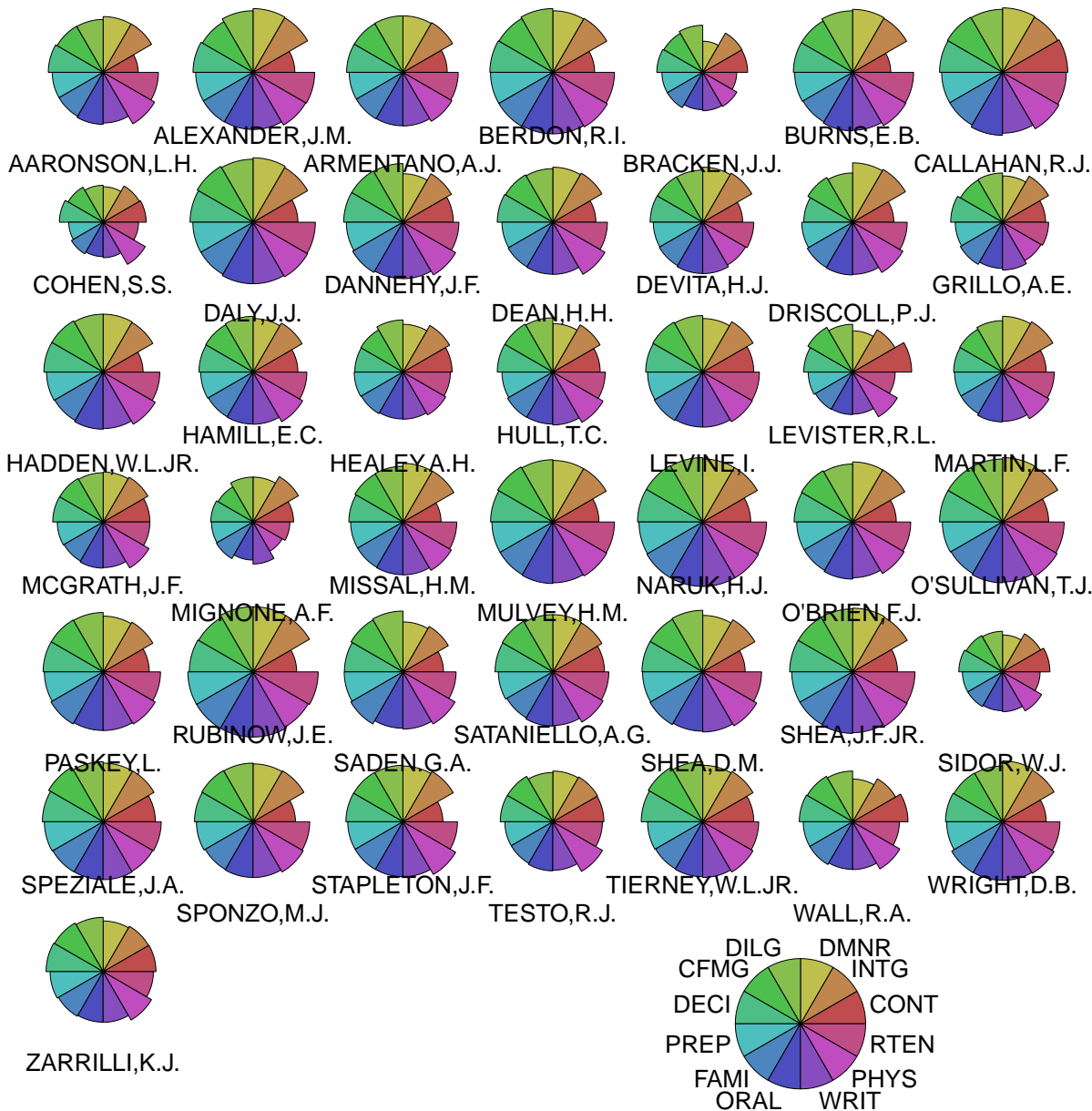
FAMI

RTEN

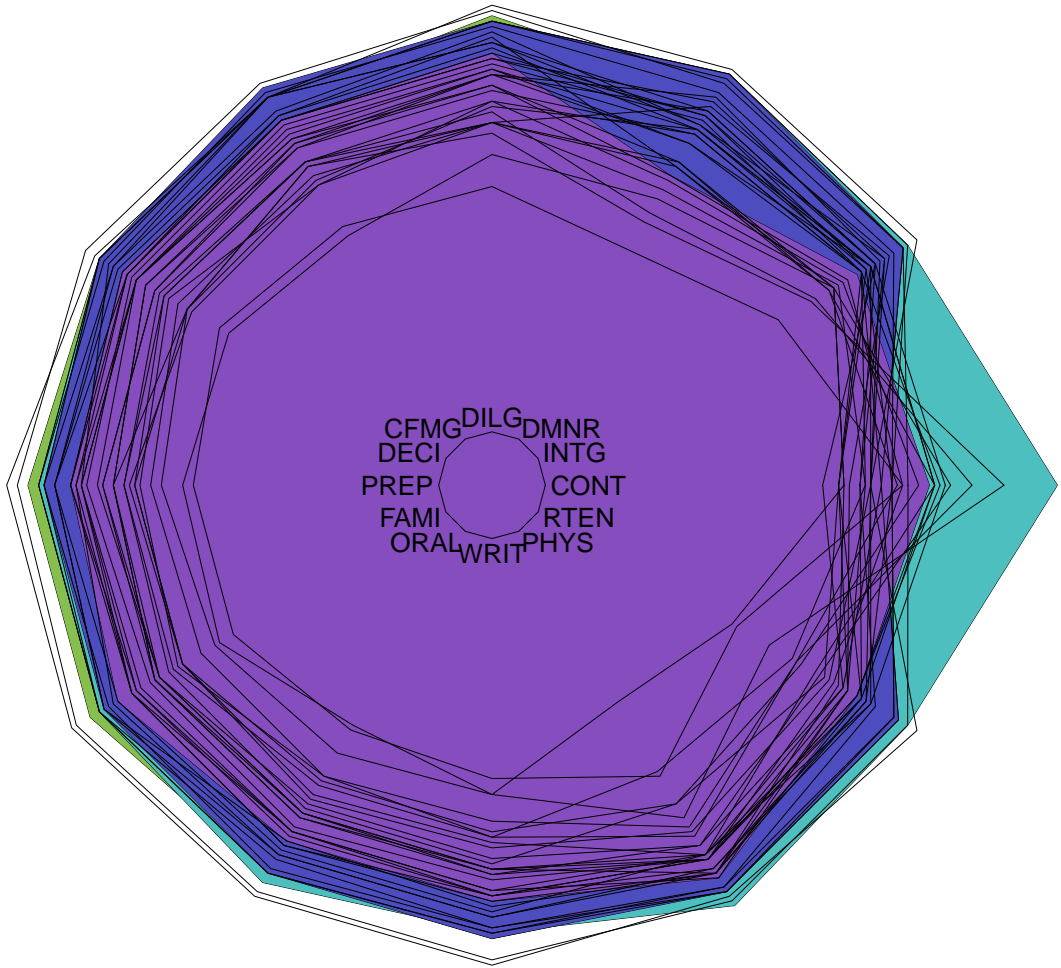
ORAL

WRIT

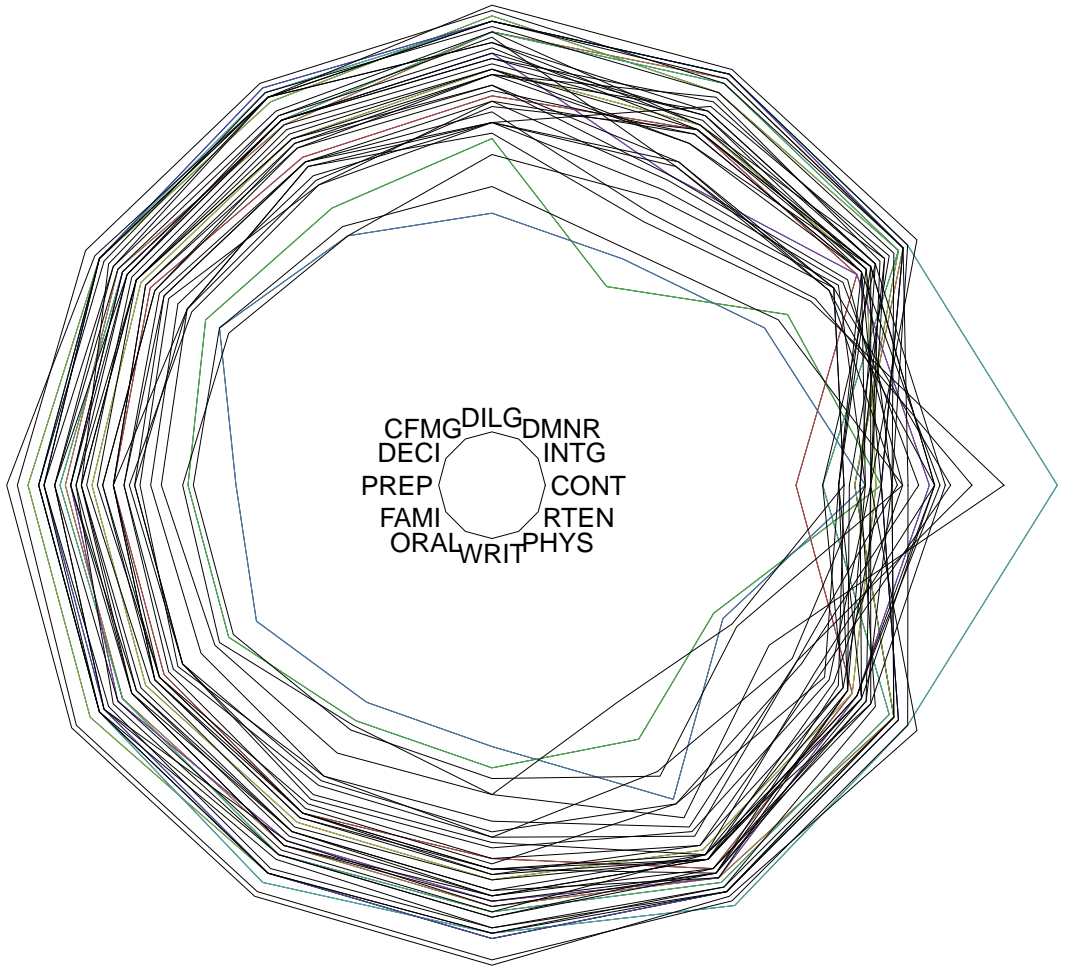
PHYS



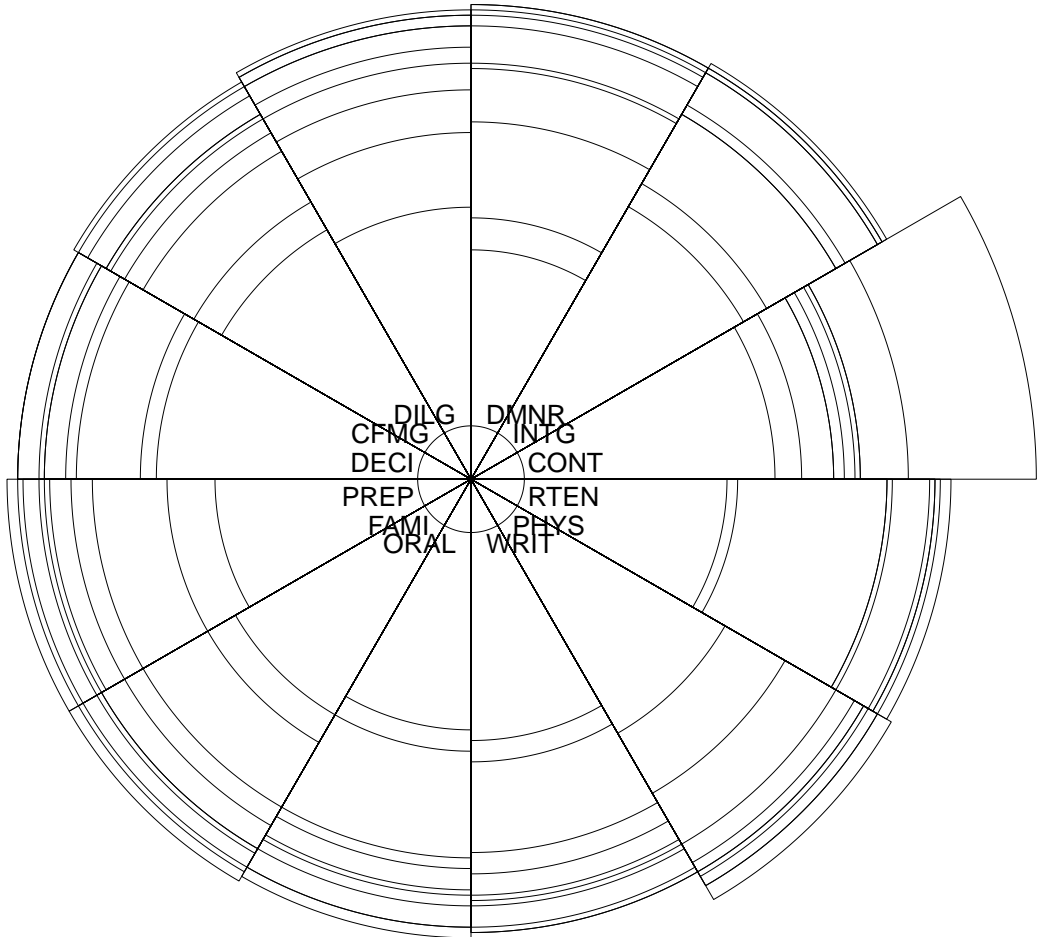
# US Judges rated



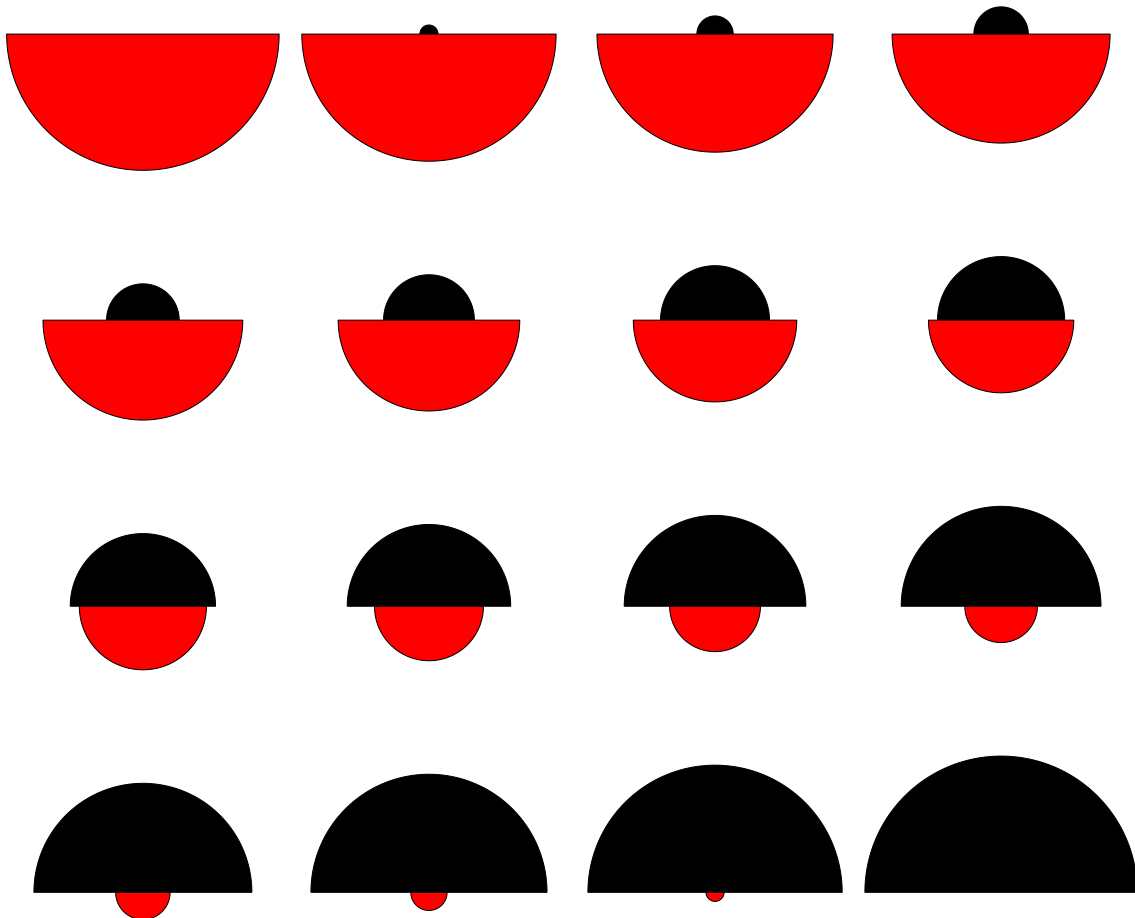
# US Judges rated



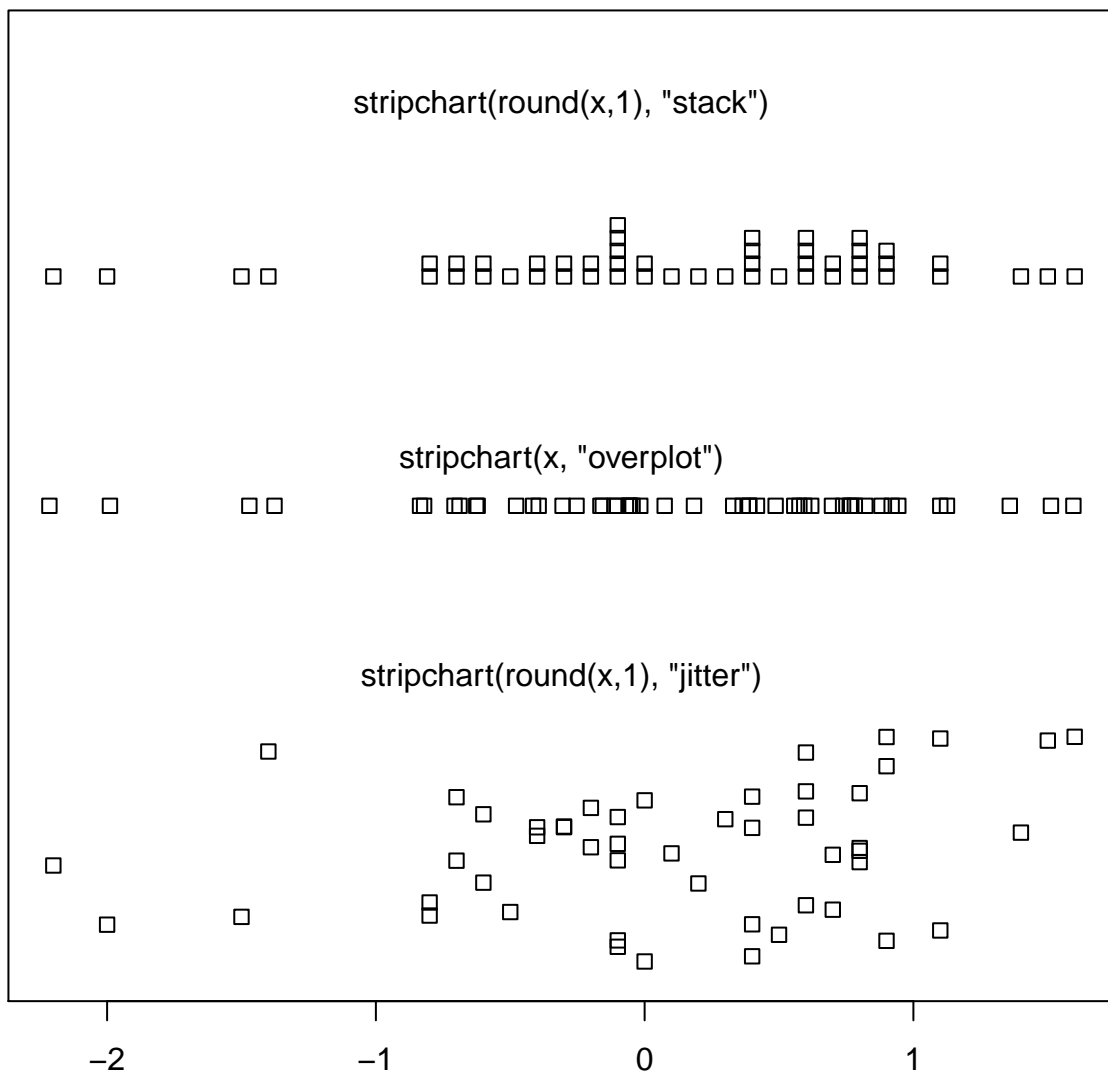
# US Judges 1–10



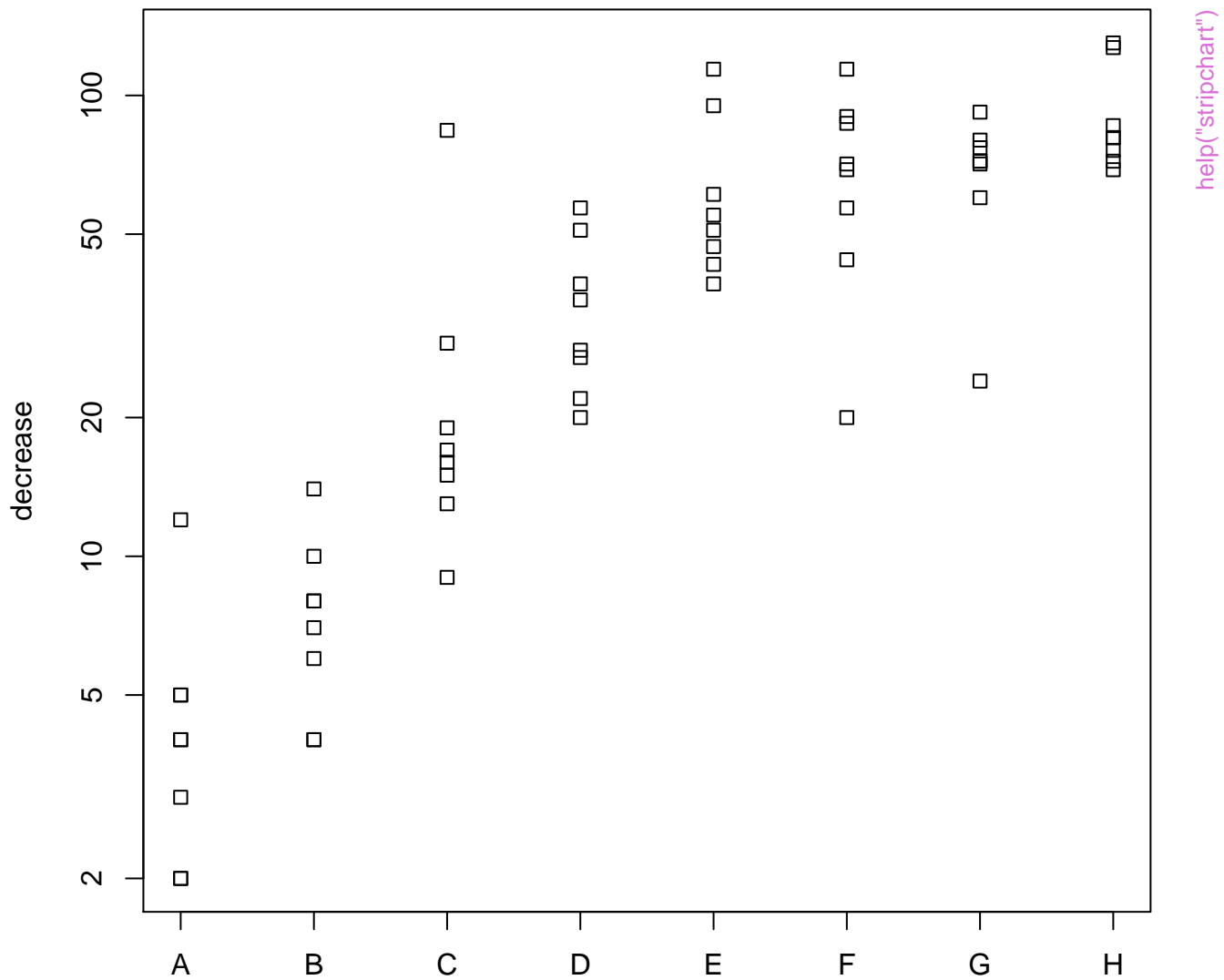
A Joke -- do *\*not\** use symbols on 2D data!



help("stars")



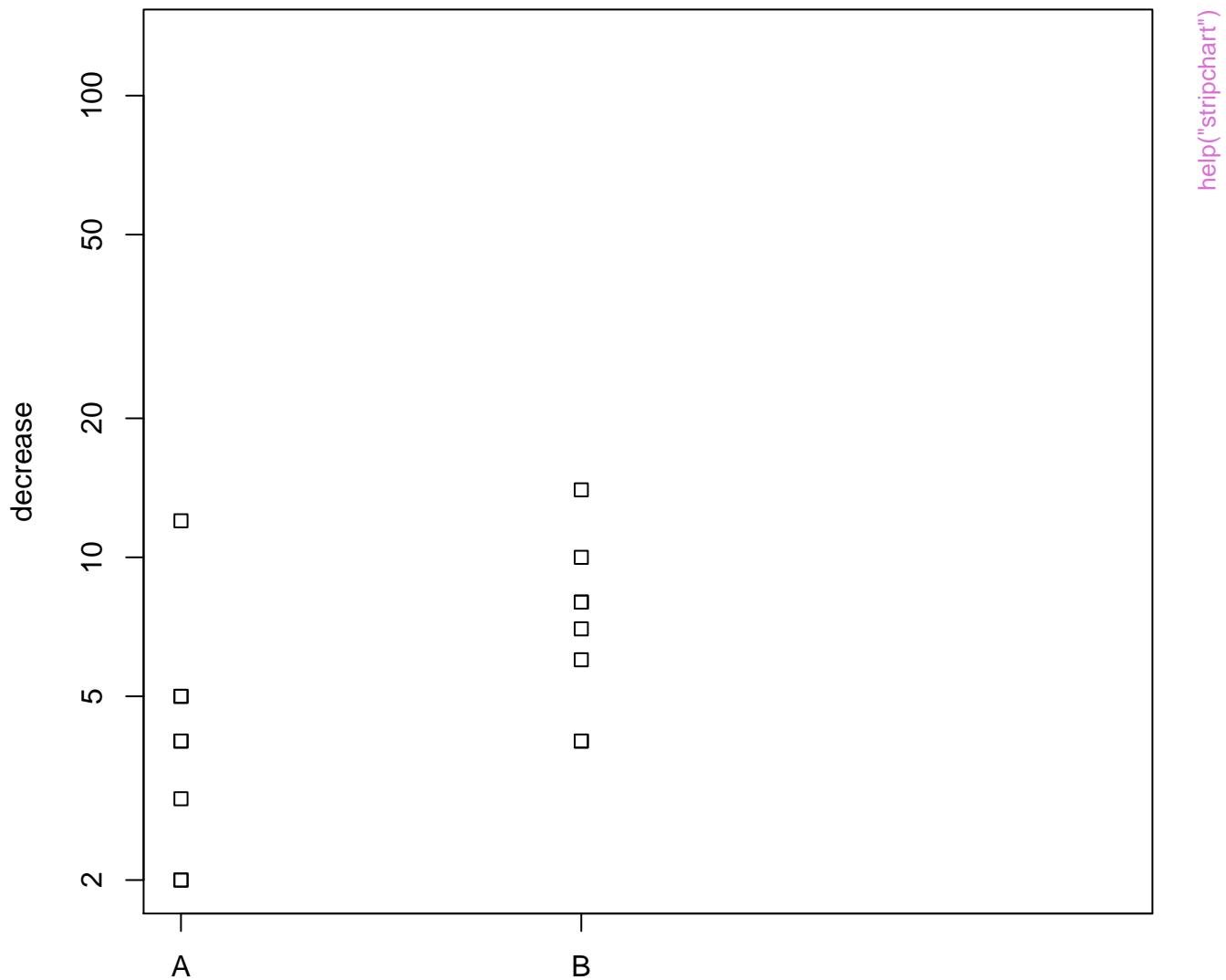
# stripchart(OrchardSprays)



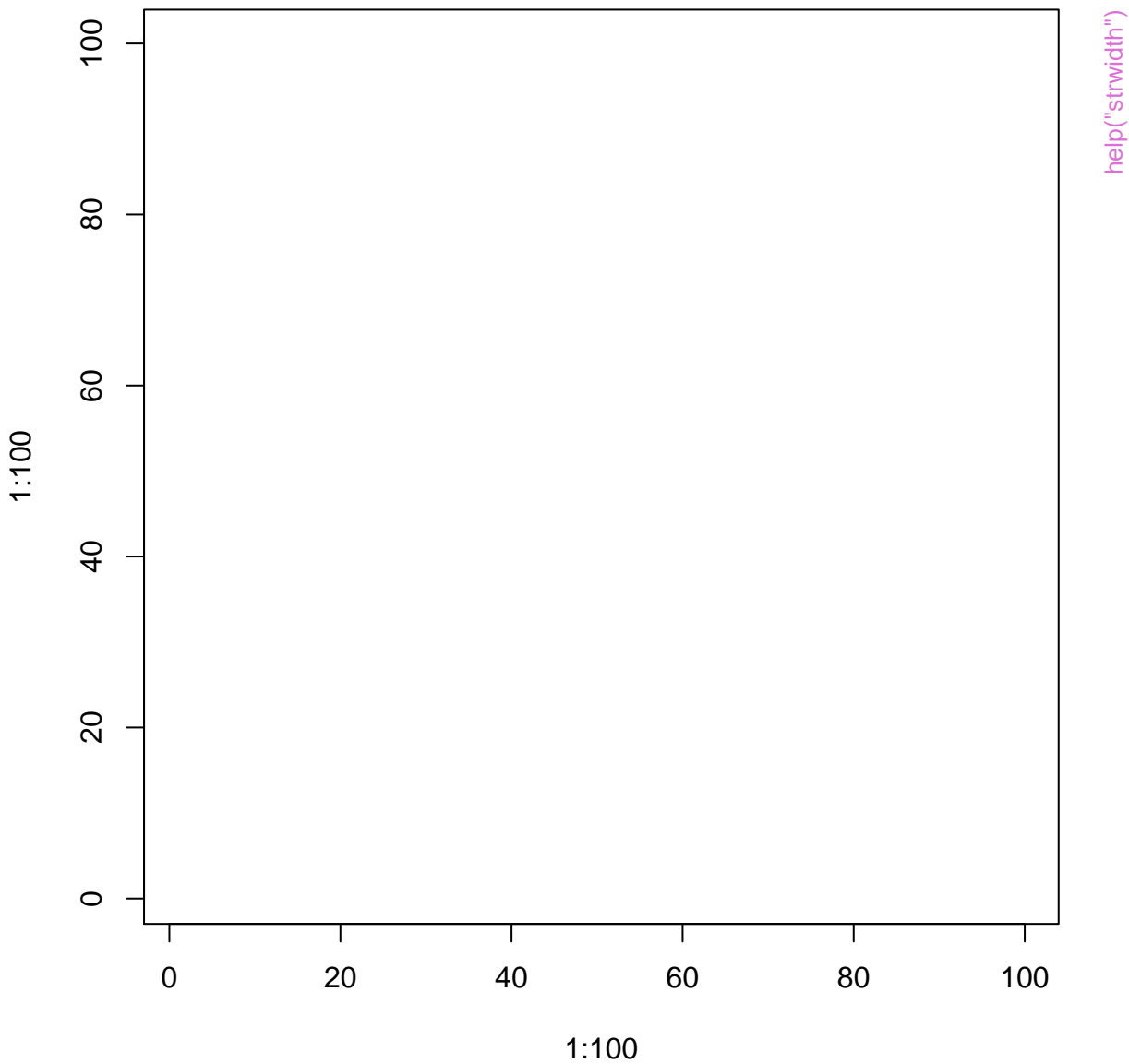
help("stripchart")

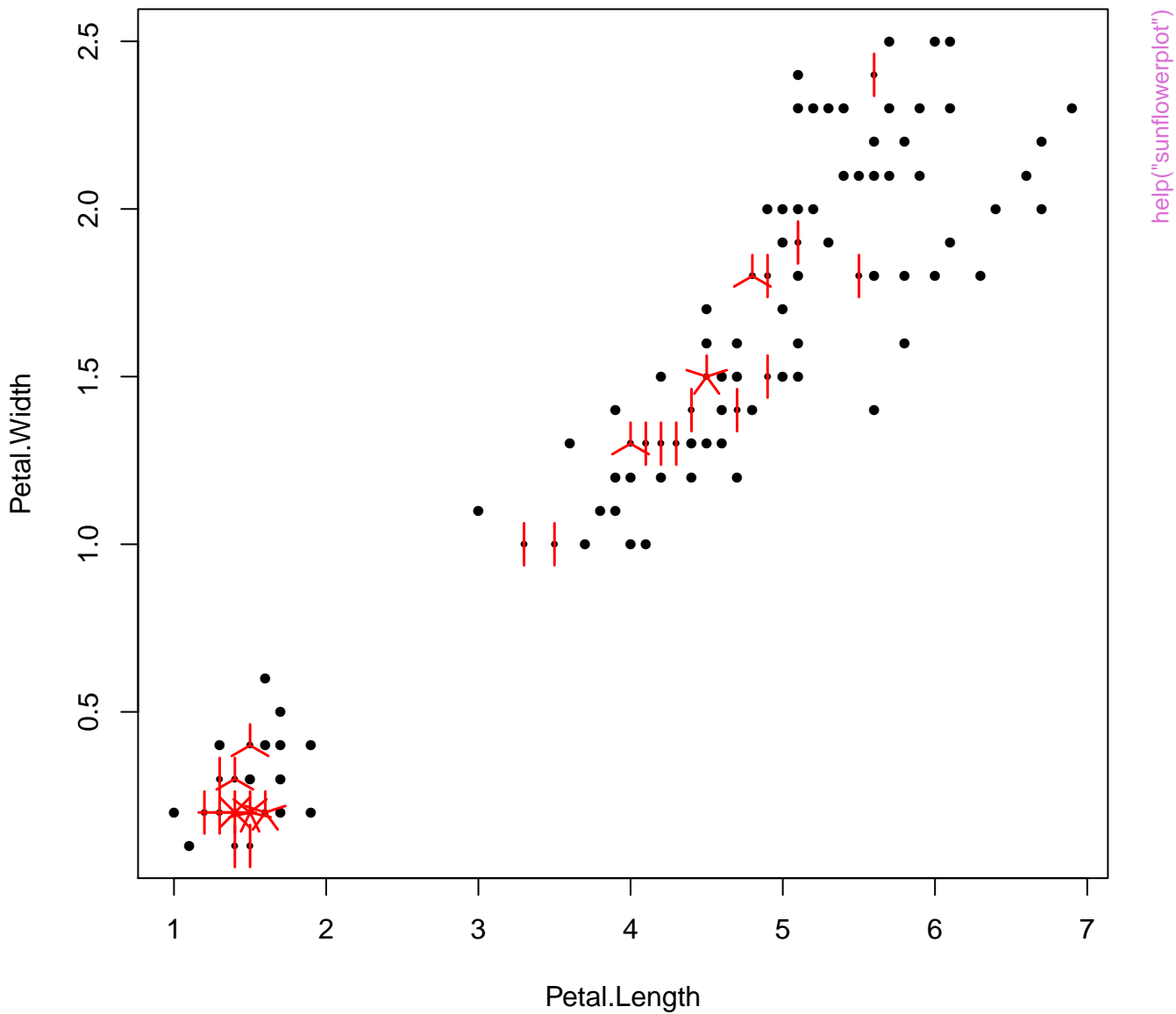


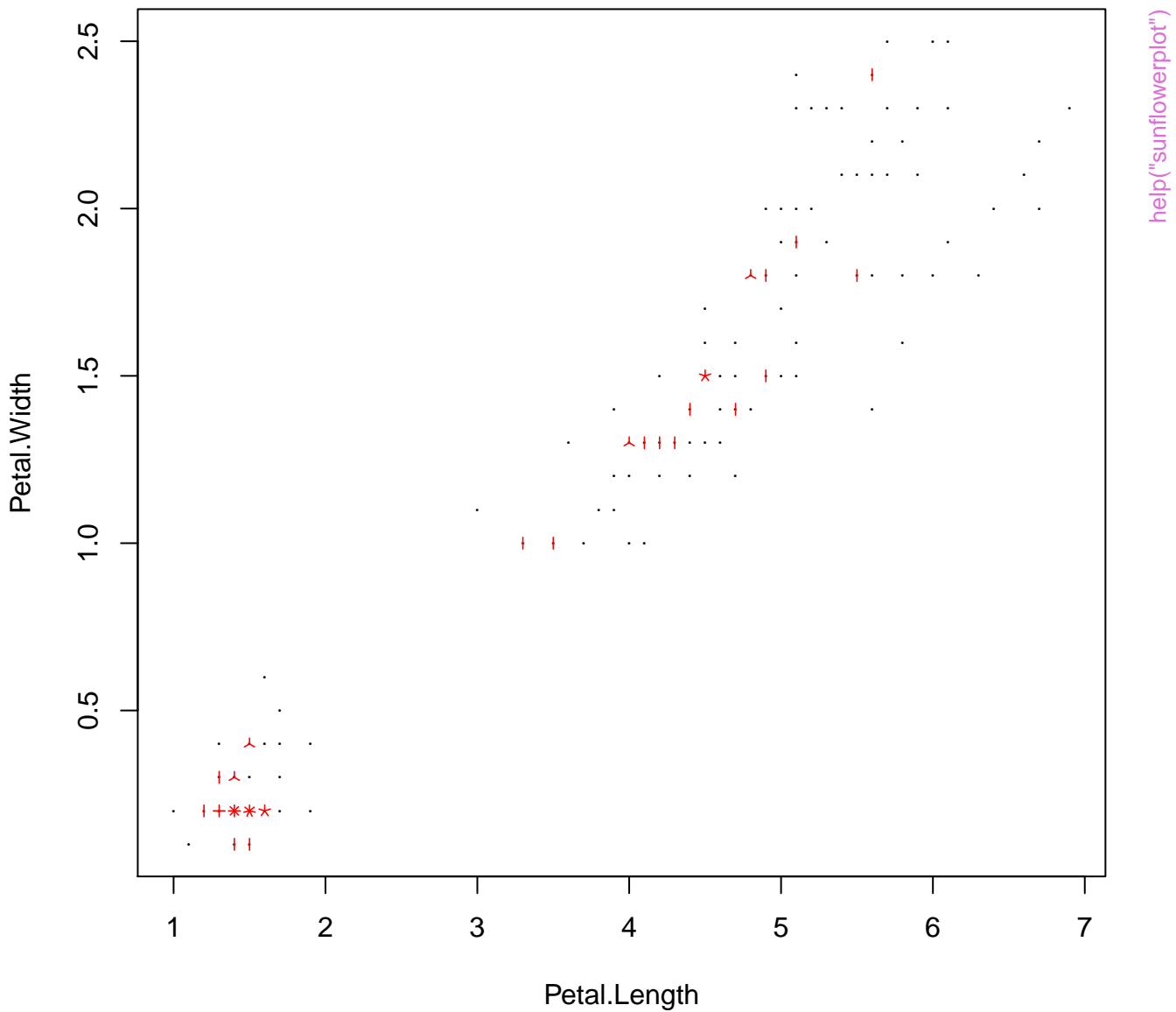
# stripchart(OrchardSprays)

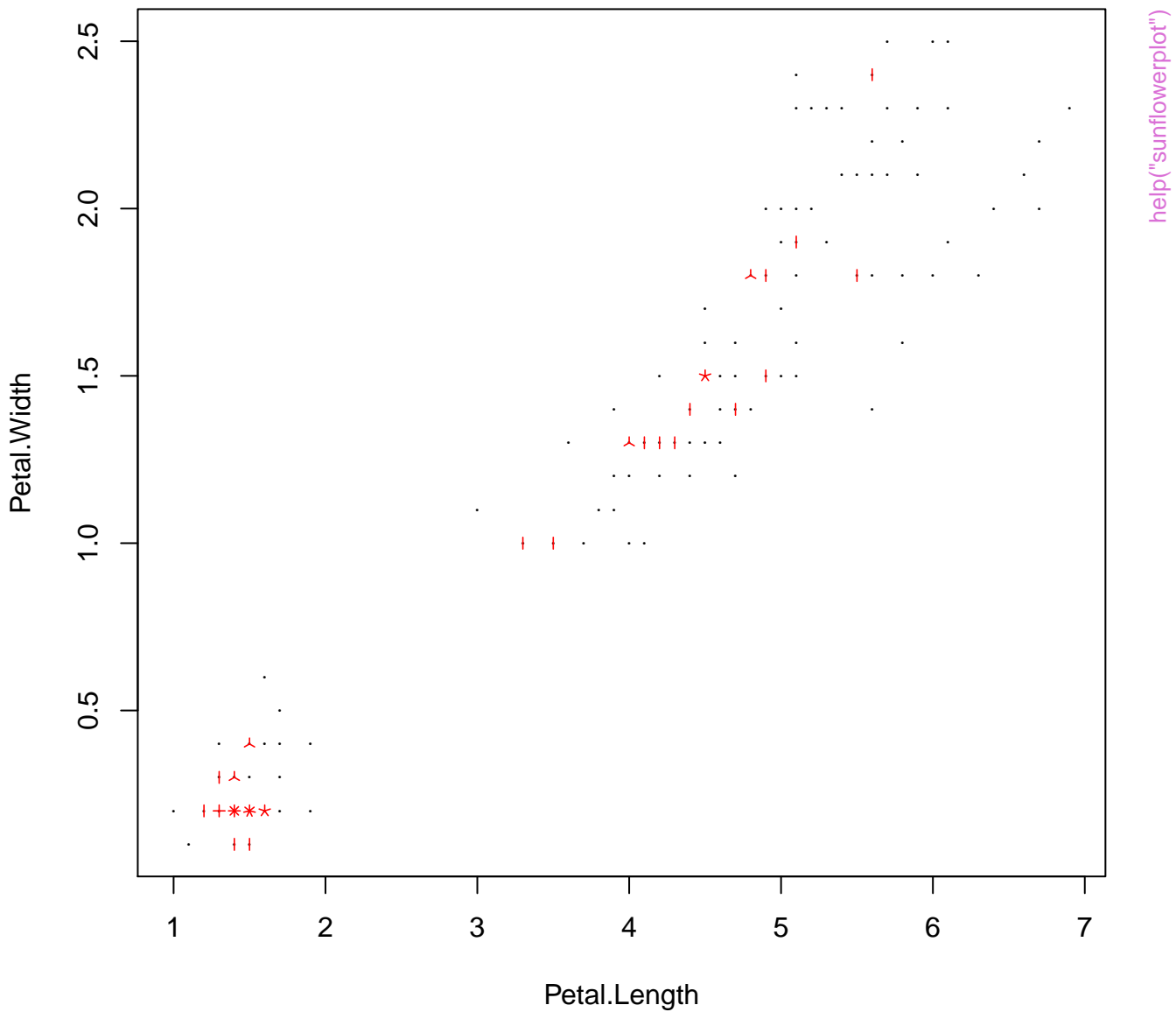


help("stripchart")

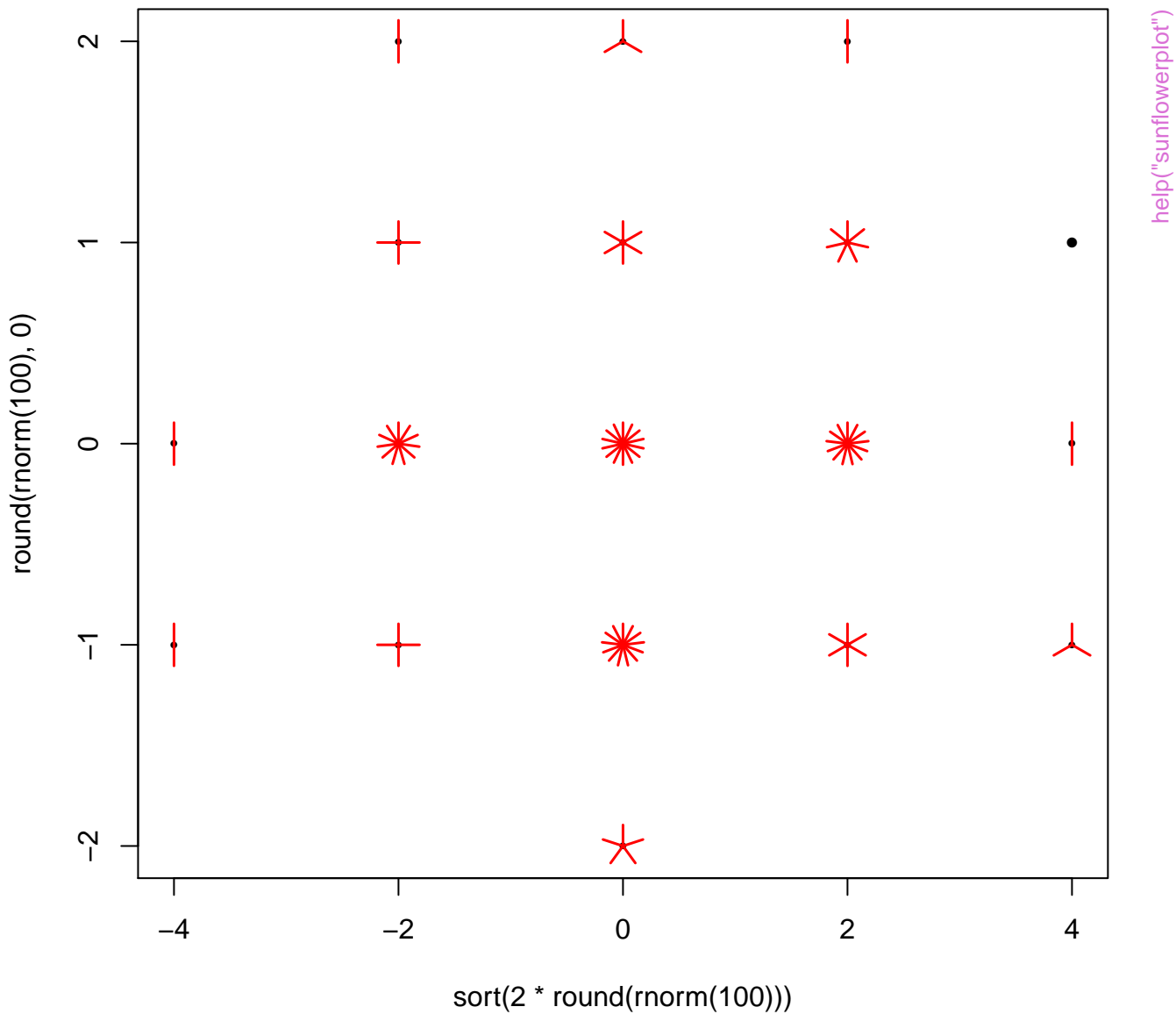




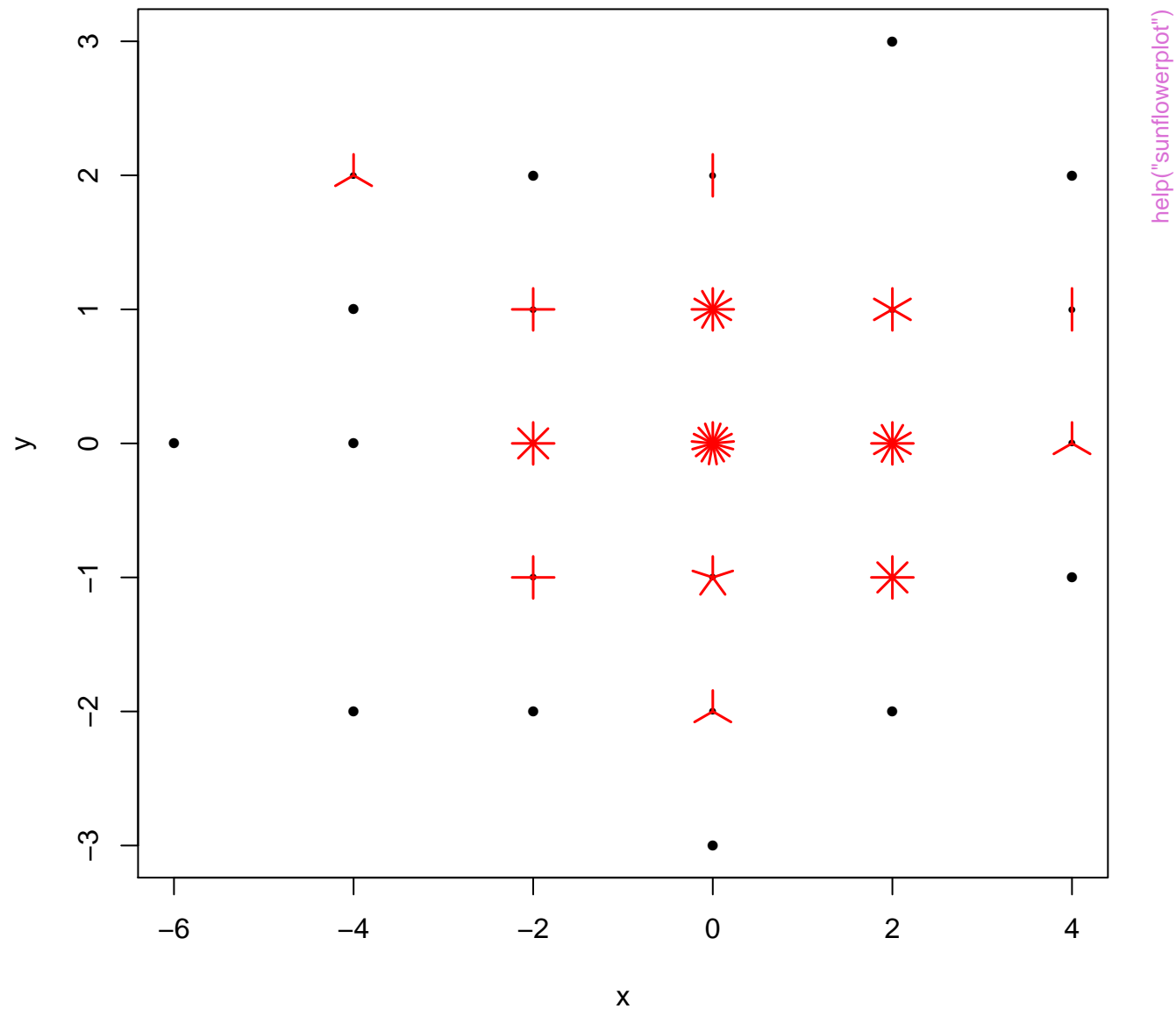




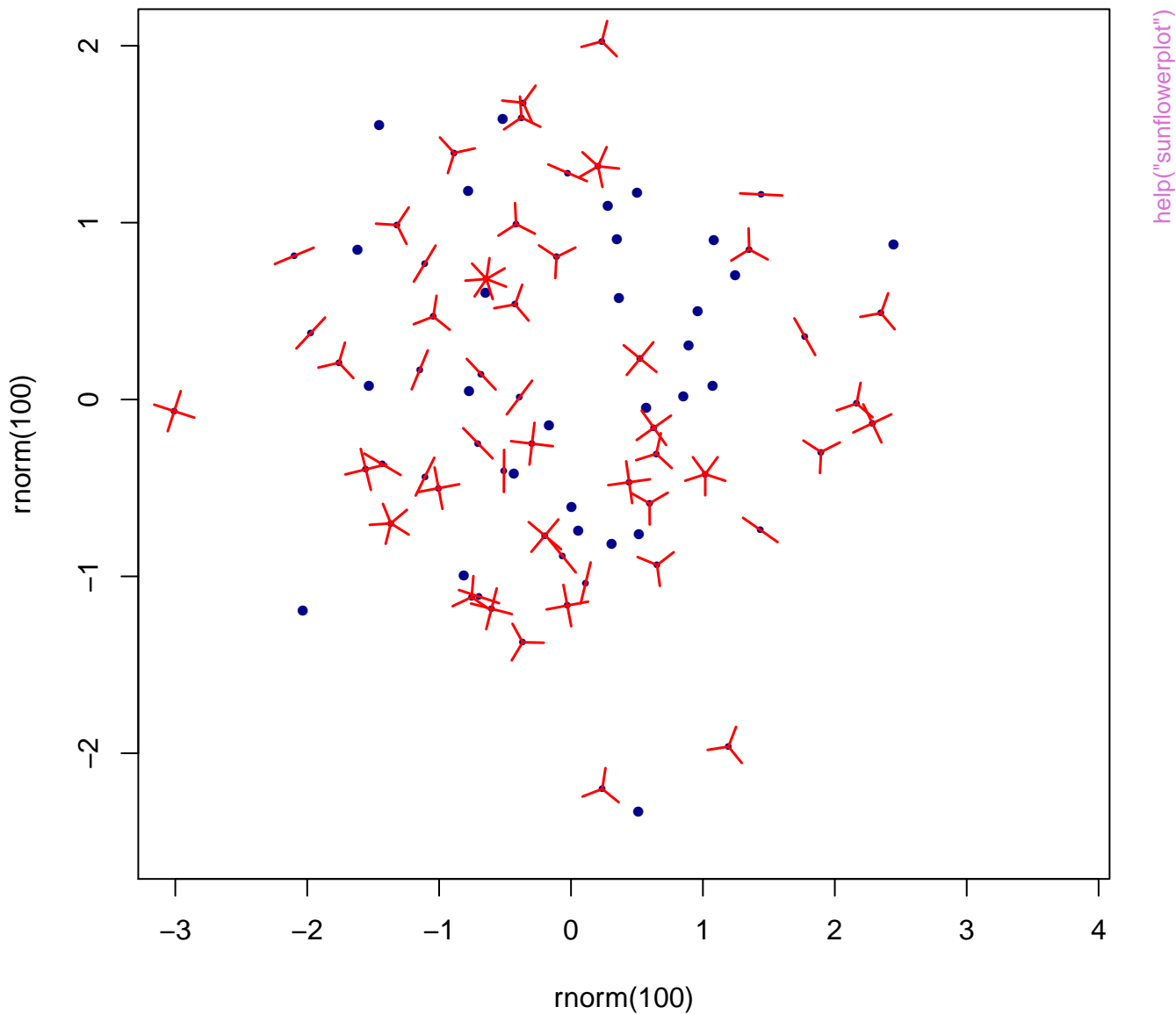
**Sunflower Plot of Rounded  $N(0,1)$**



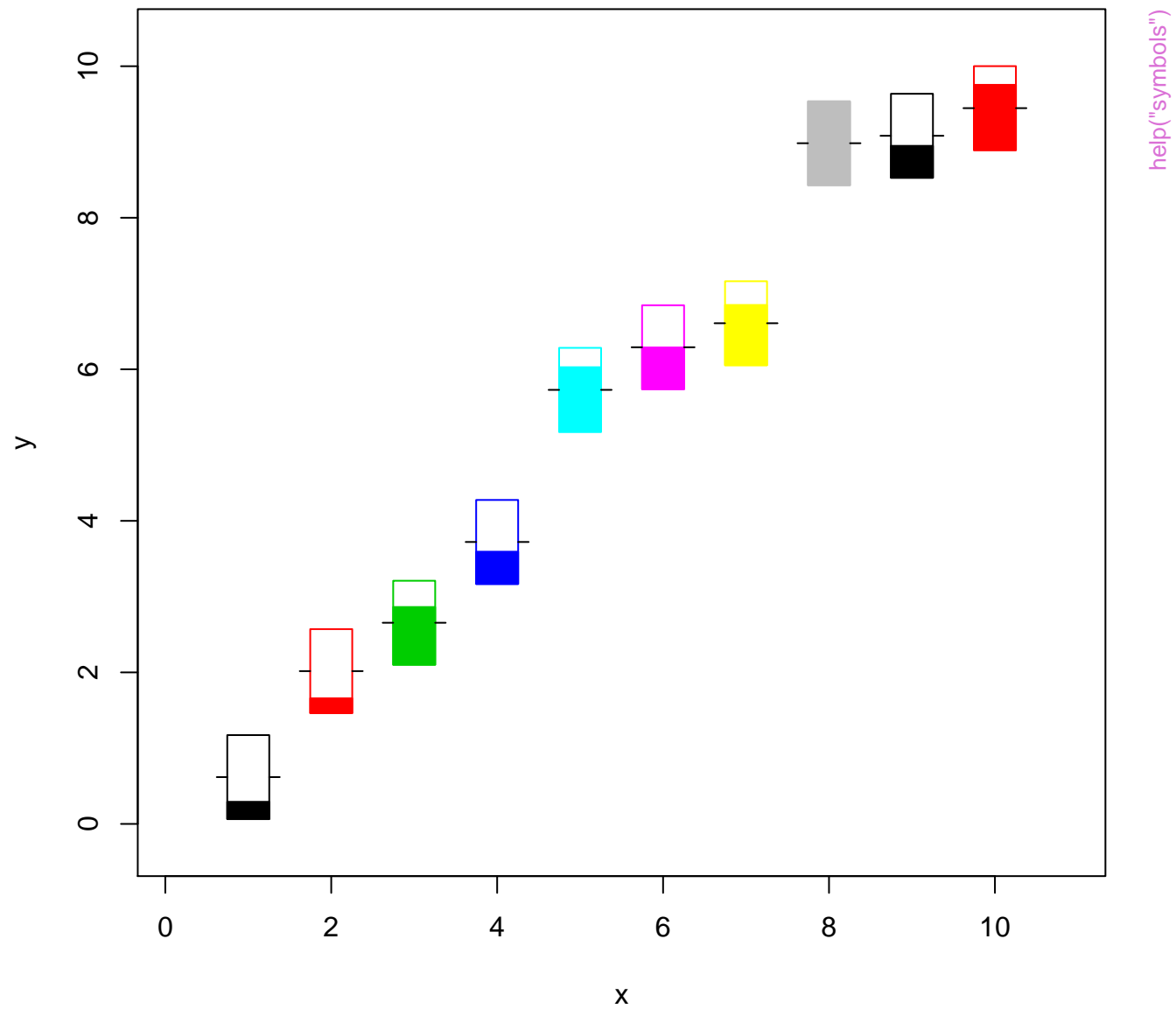
2nd Sunflower Plot of Rounded  $N(0,1)$

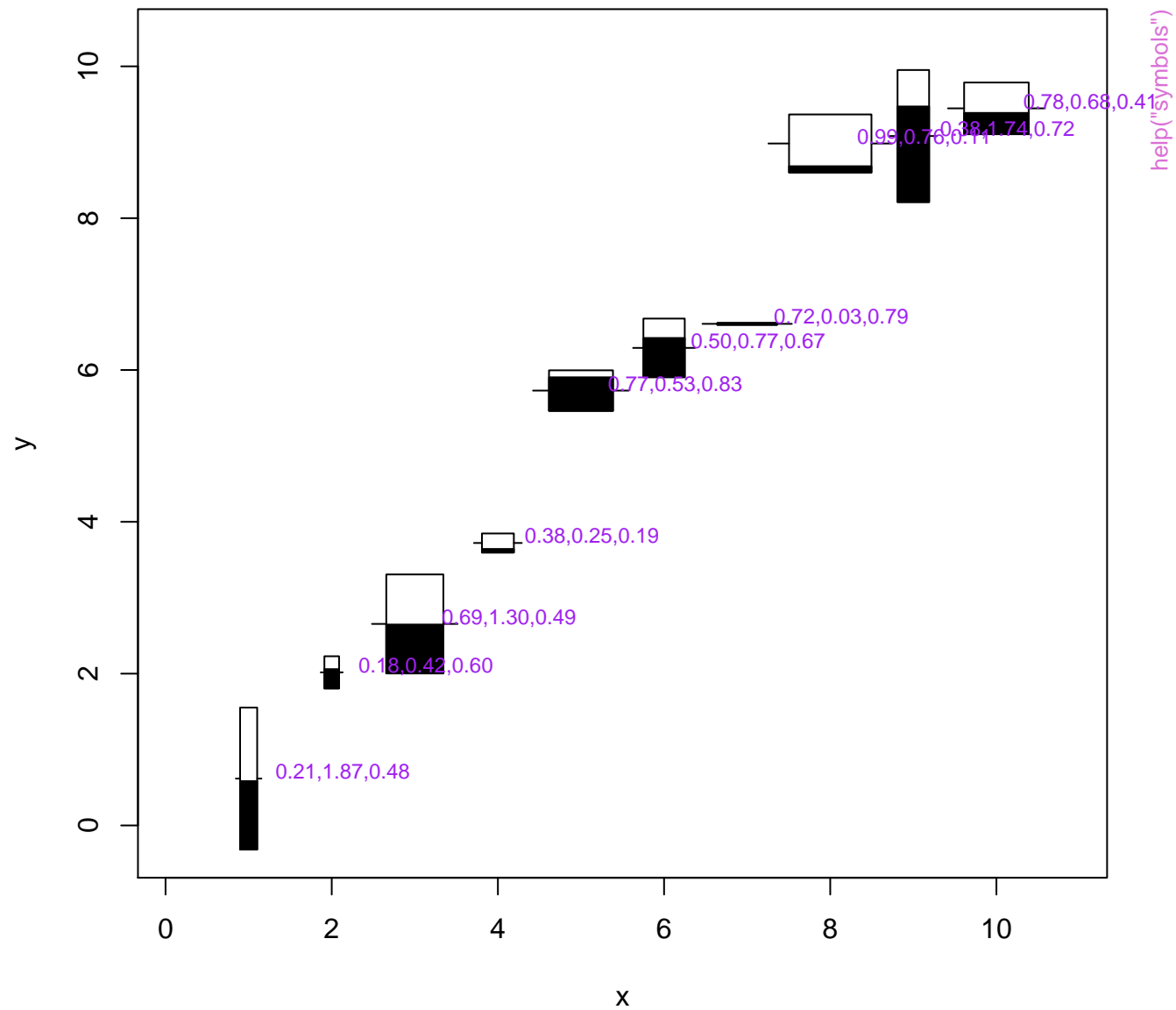


Sunflower plot (marked point process)

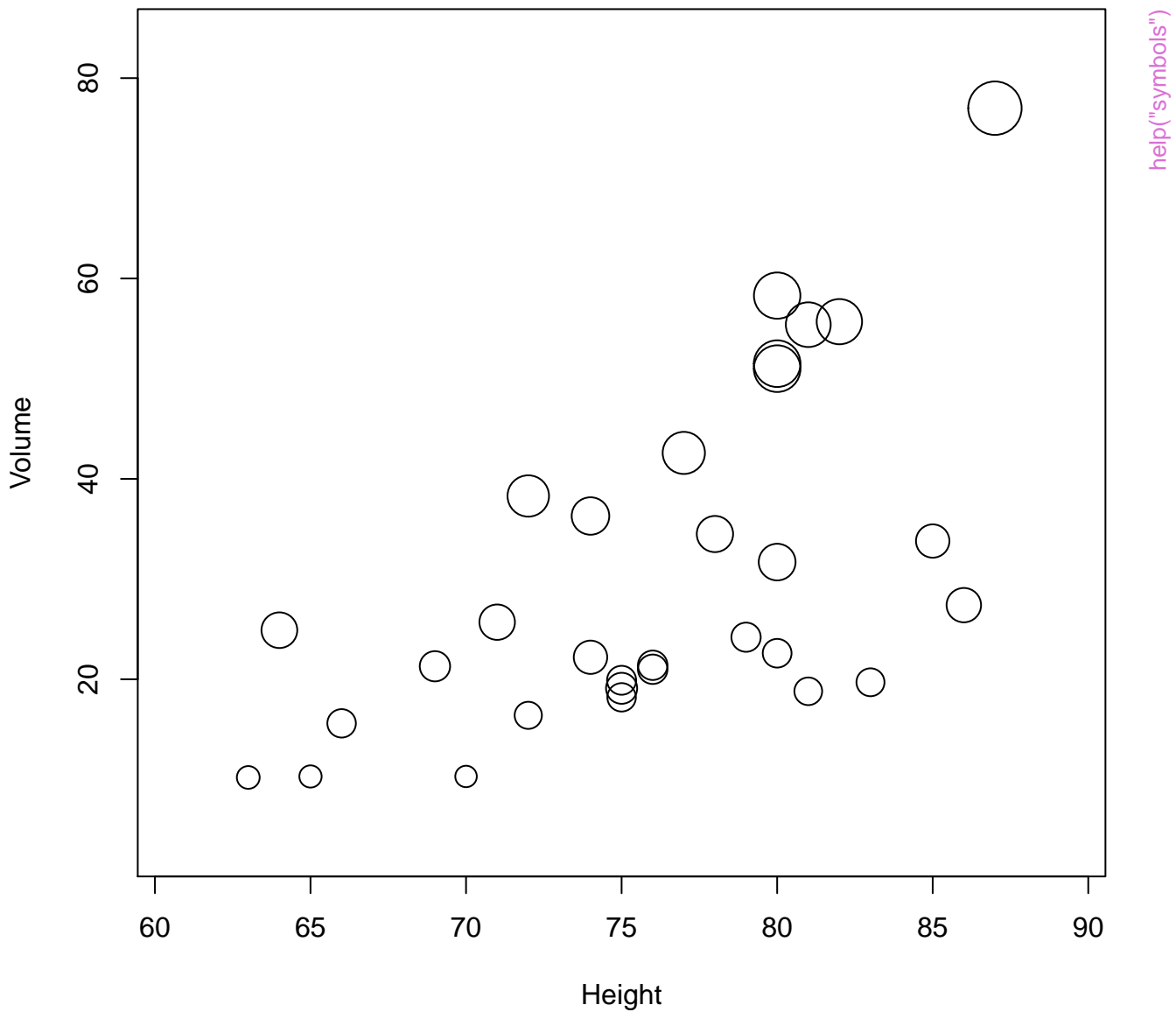




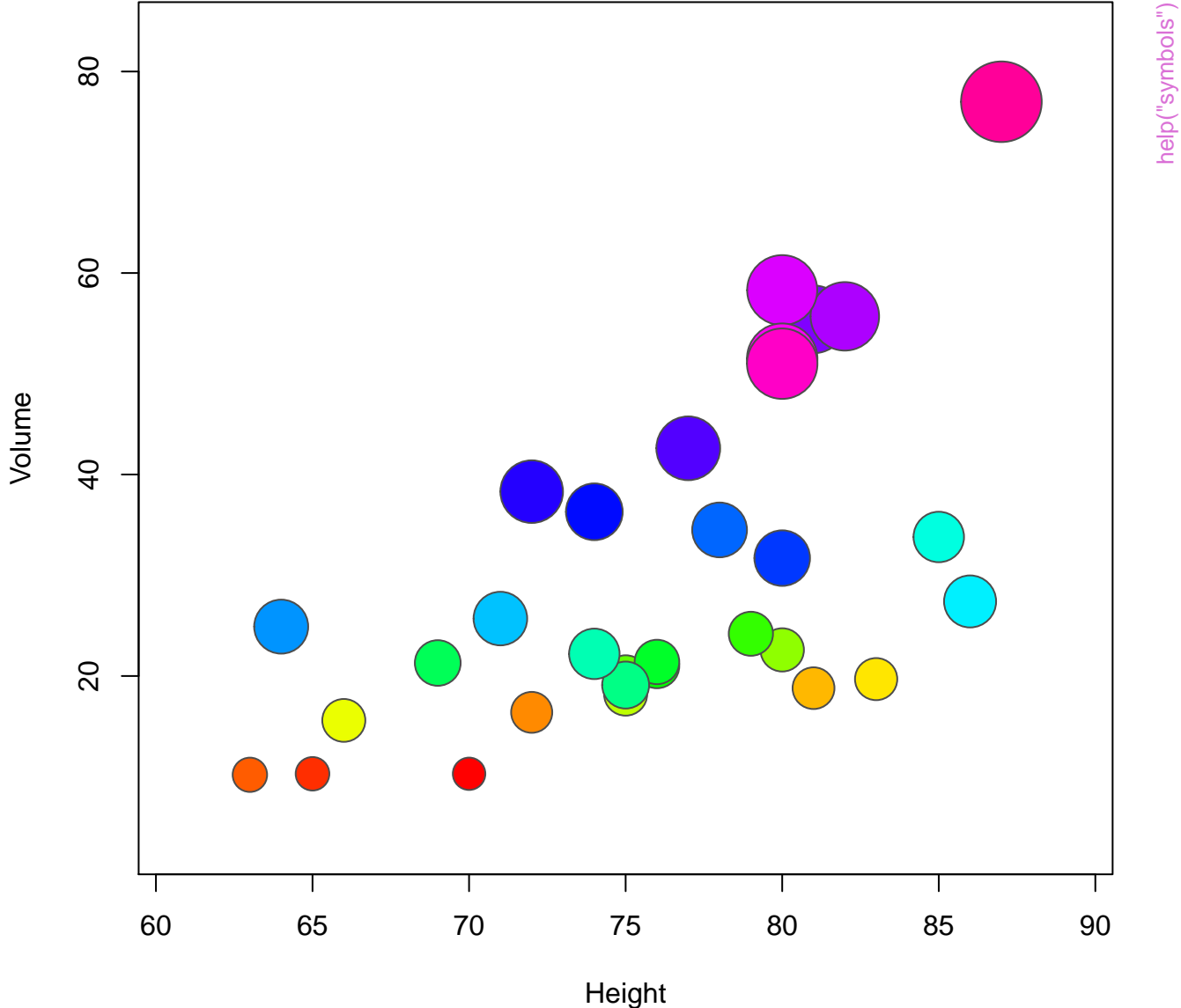




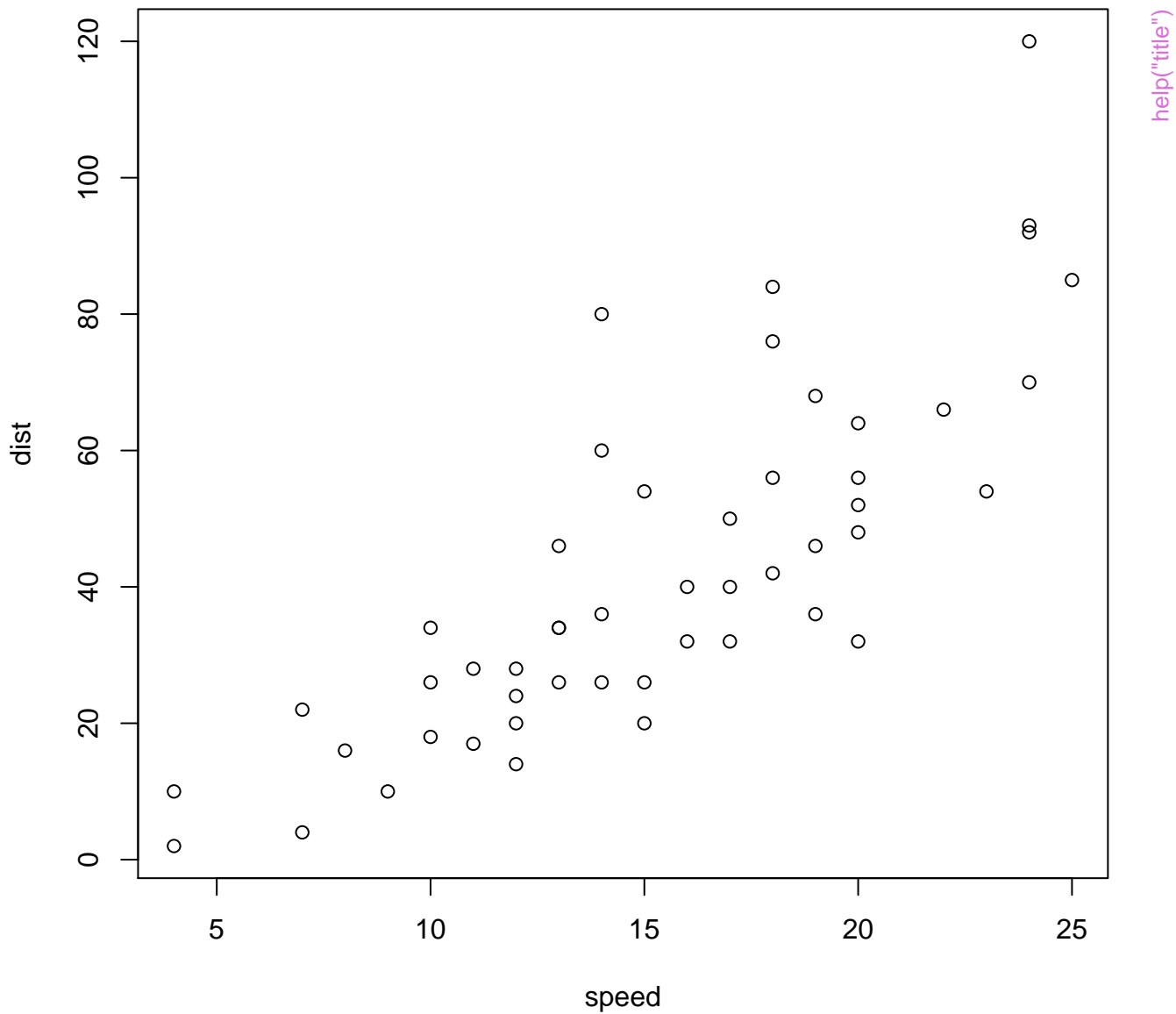
# Trees' Girth



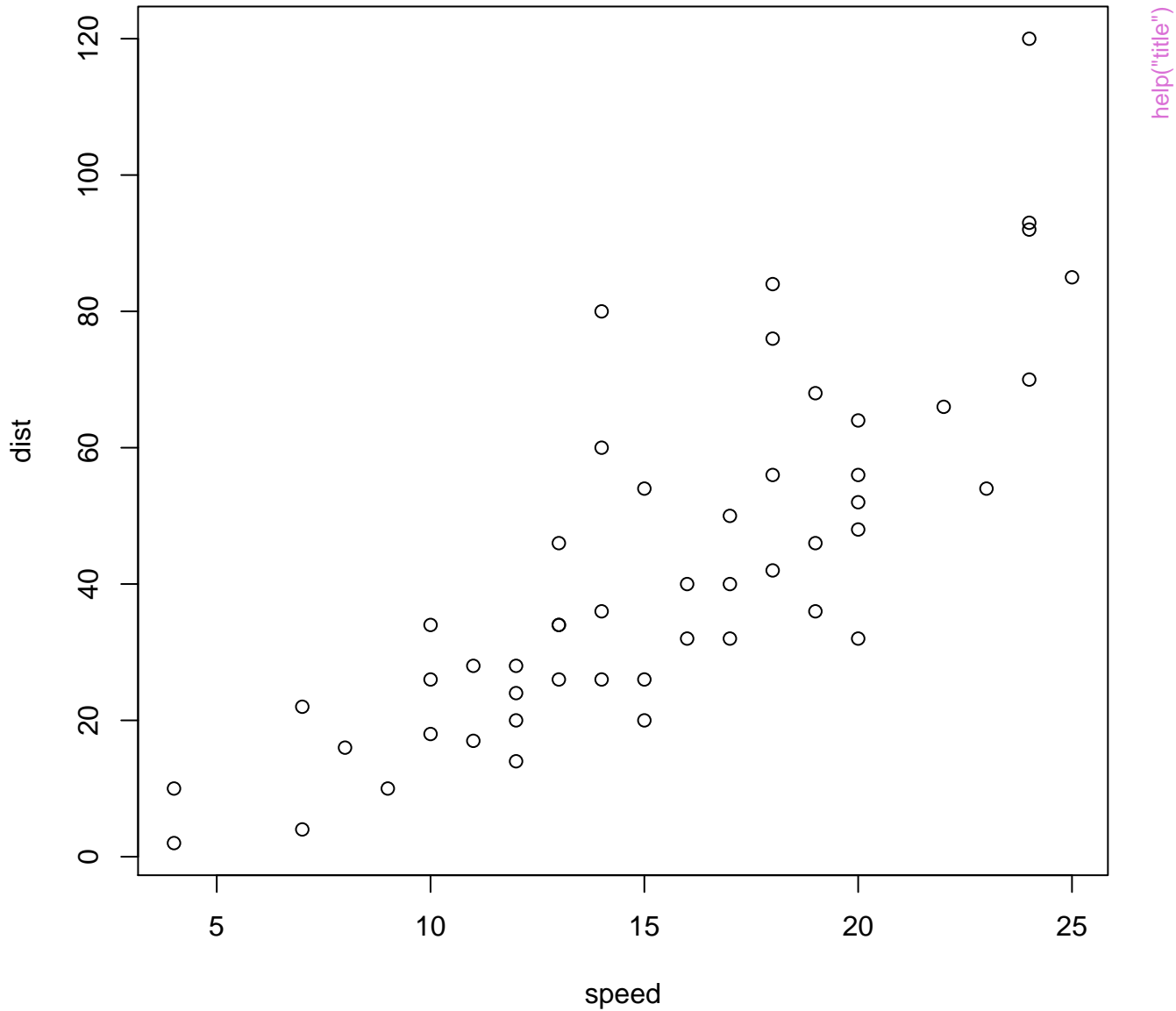
**symbols(\*, circles = Girth/16, bg = 1:N)**



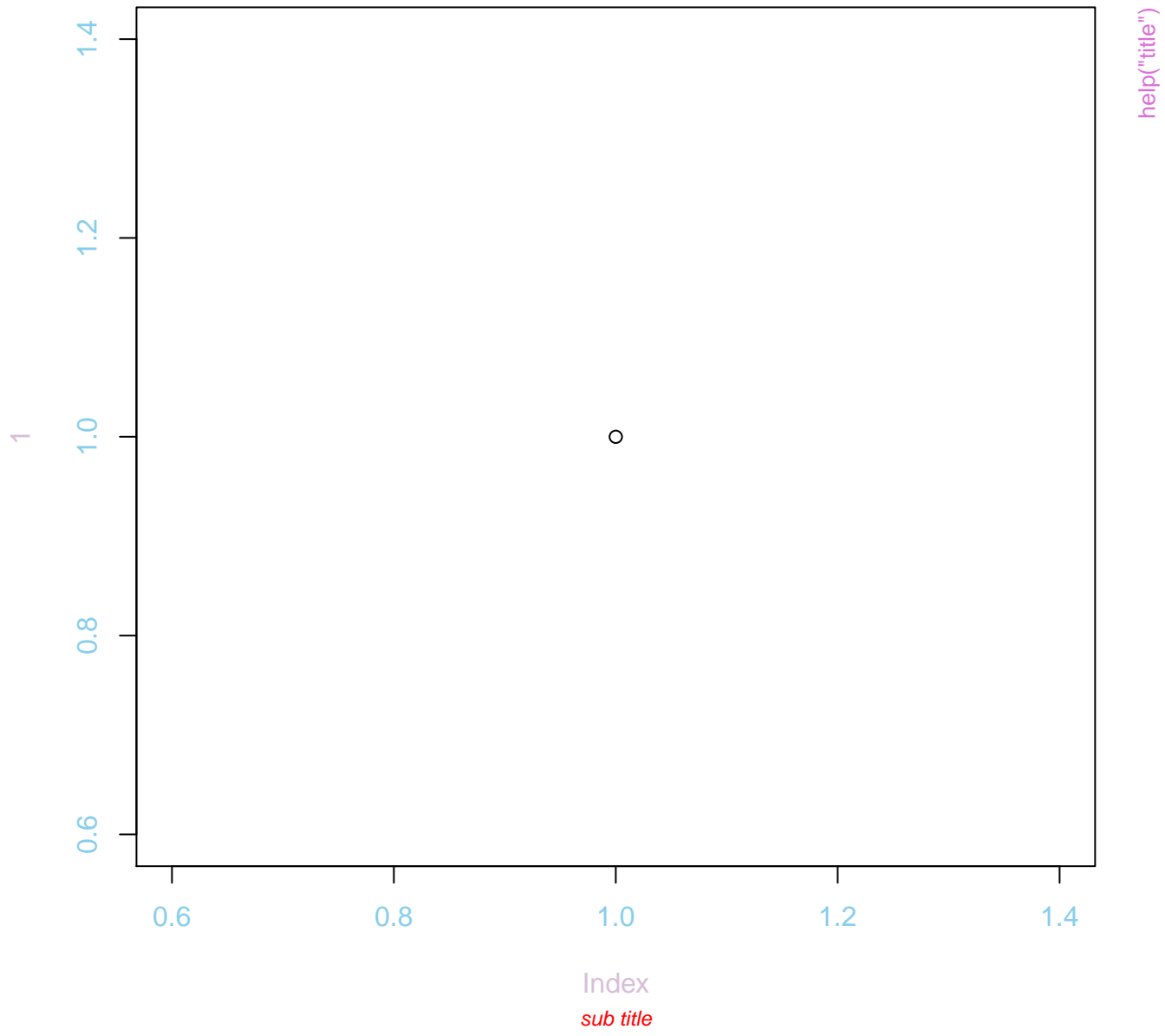
# Stopping Distance versus Speed



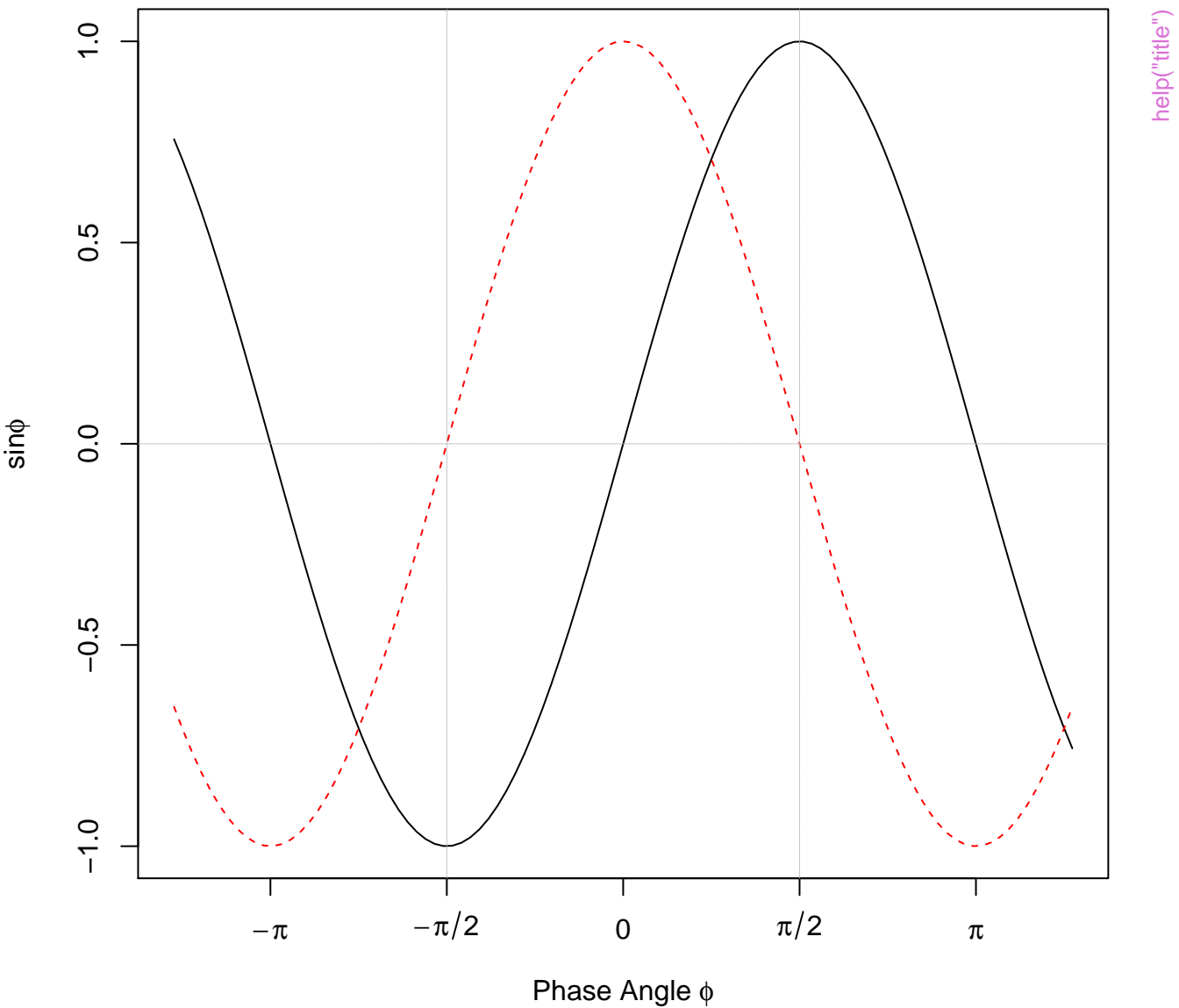
# *Stopping Distance versus Speed*



# ***Main Title***



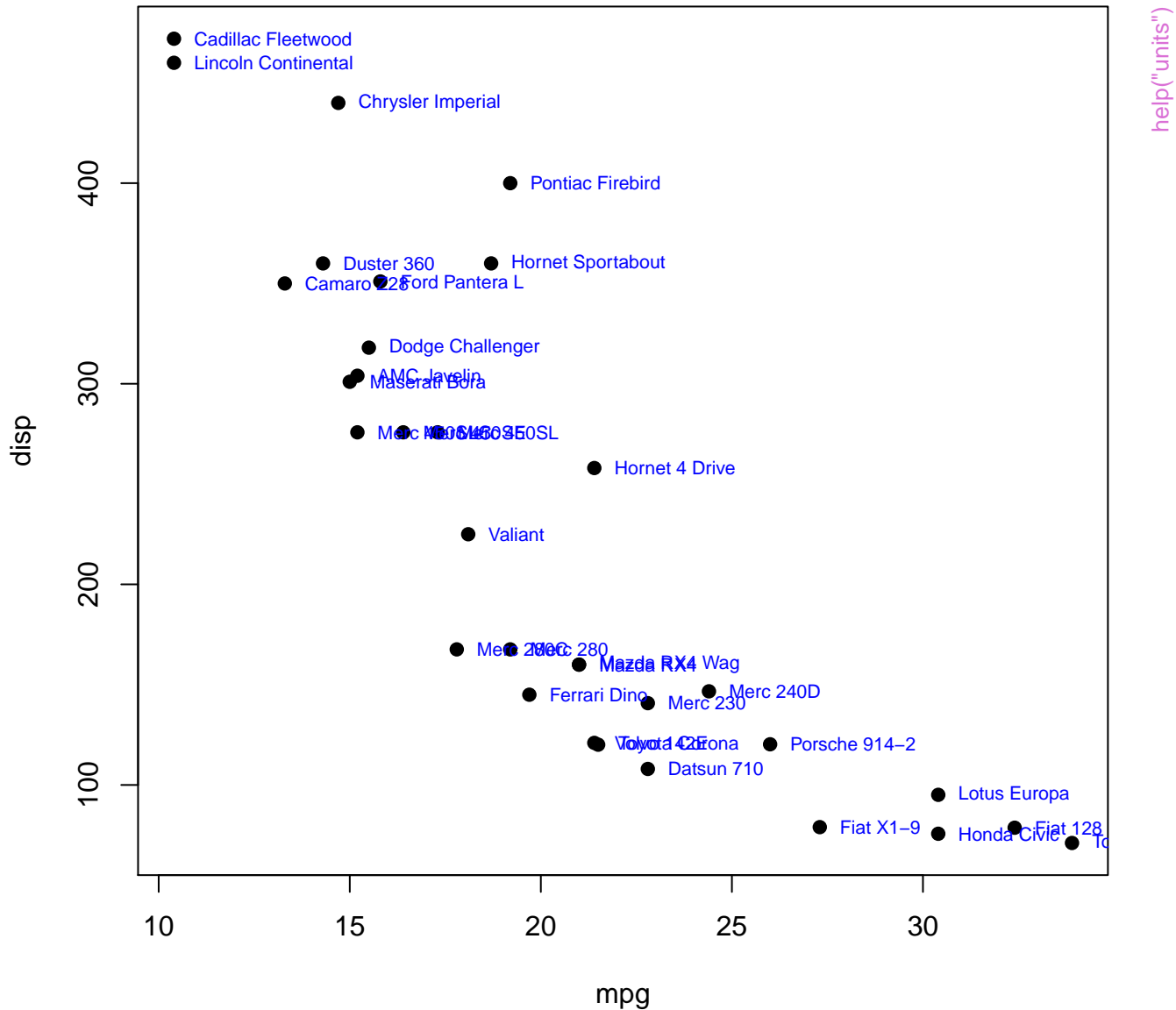
$\sin\phi$  and  $\cos\phi$



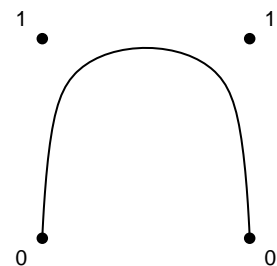
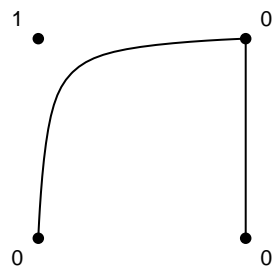
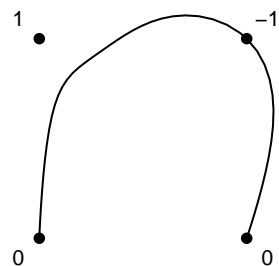
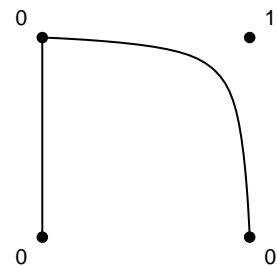
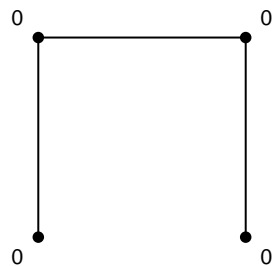
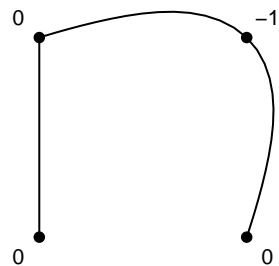
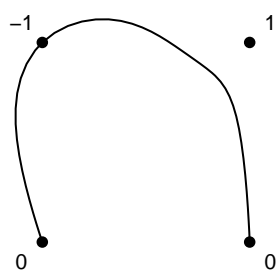
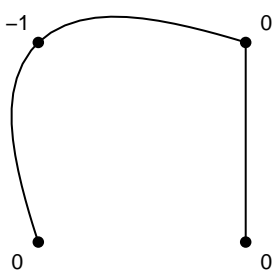
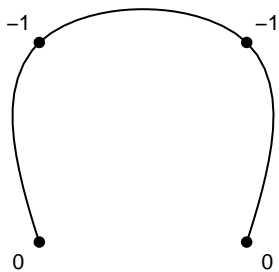
help("title")



## Motor Trend Cars



# Open X-splines



## Closed X-splines

