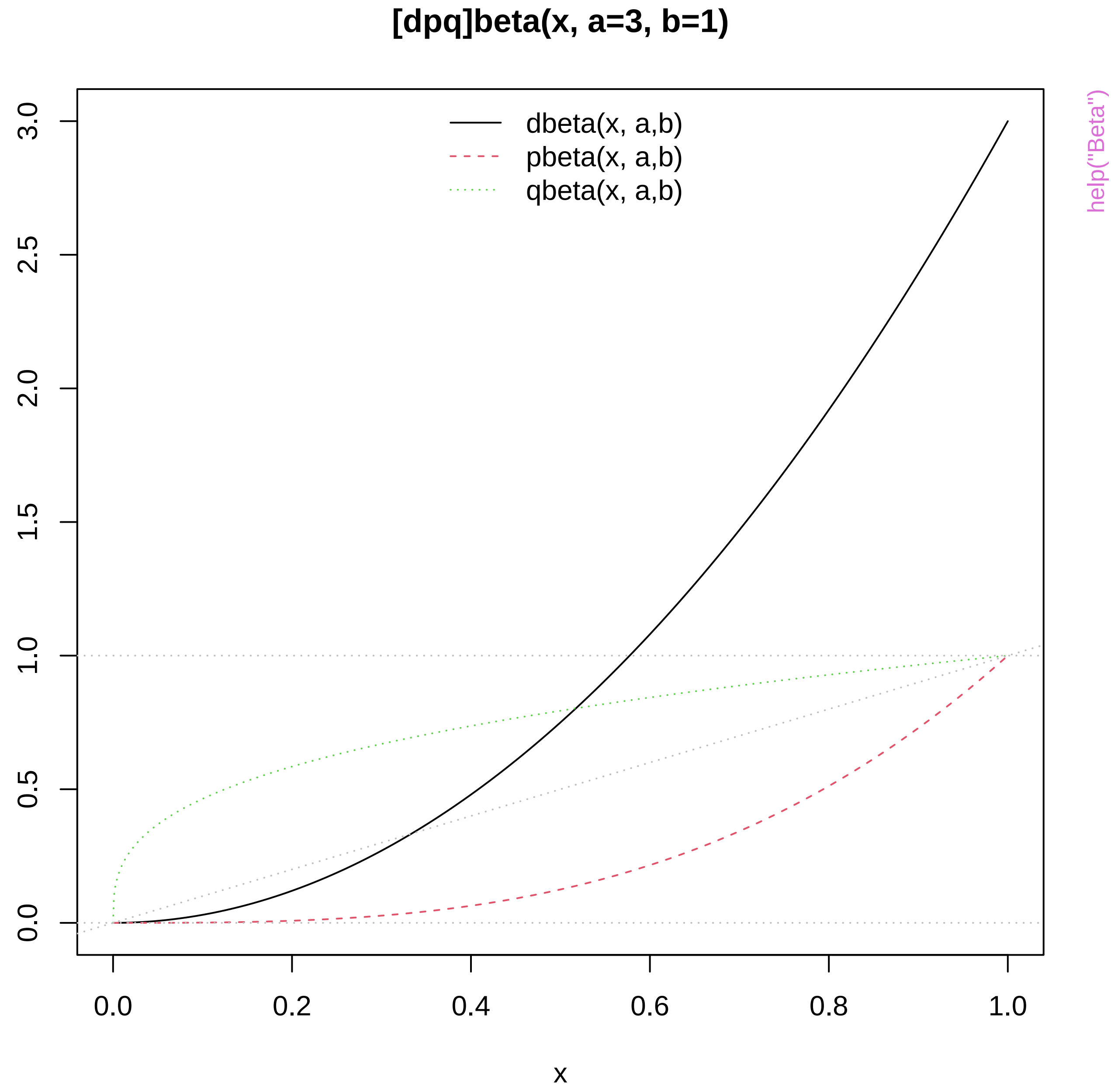
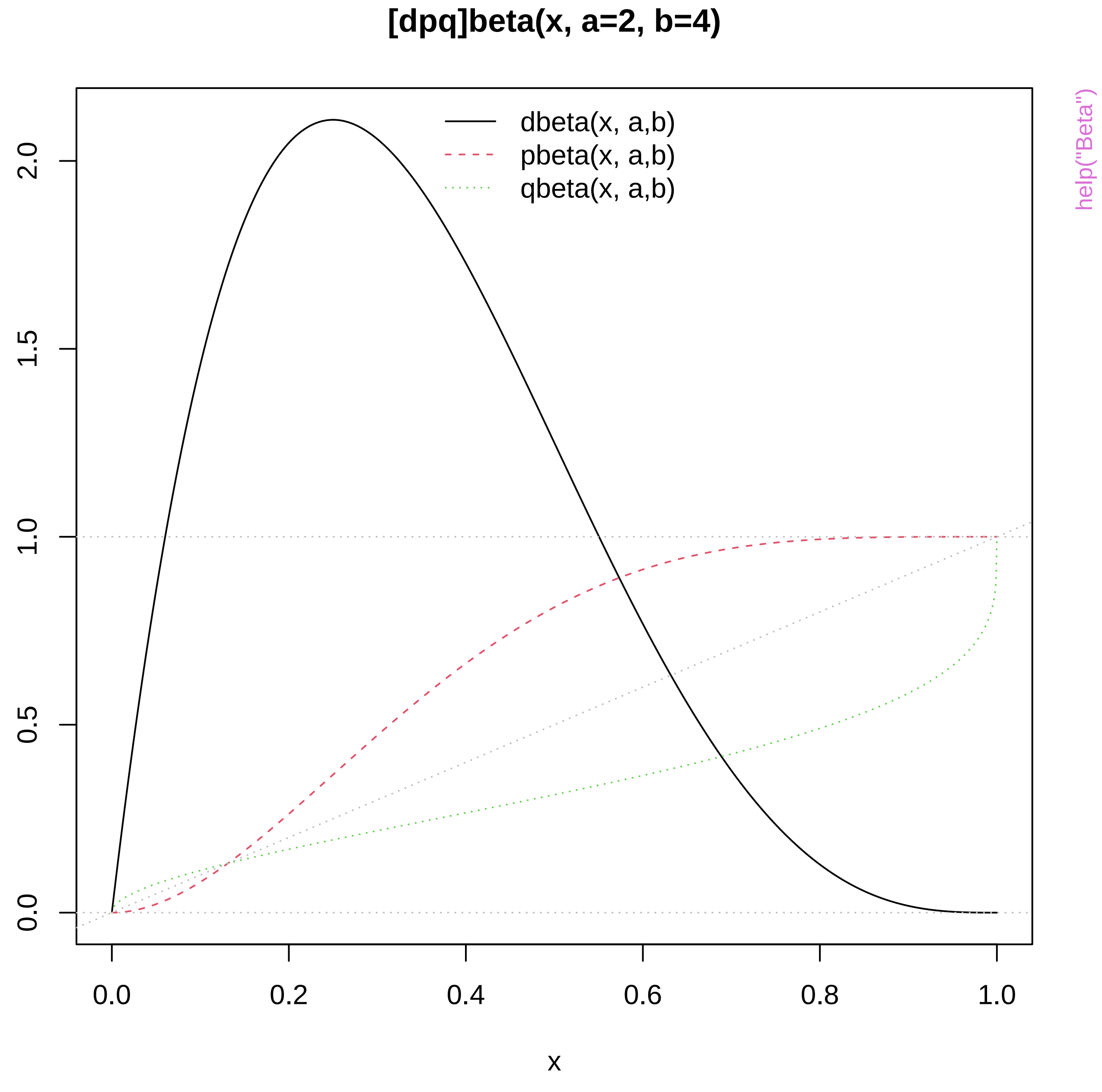
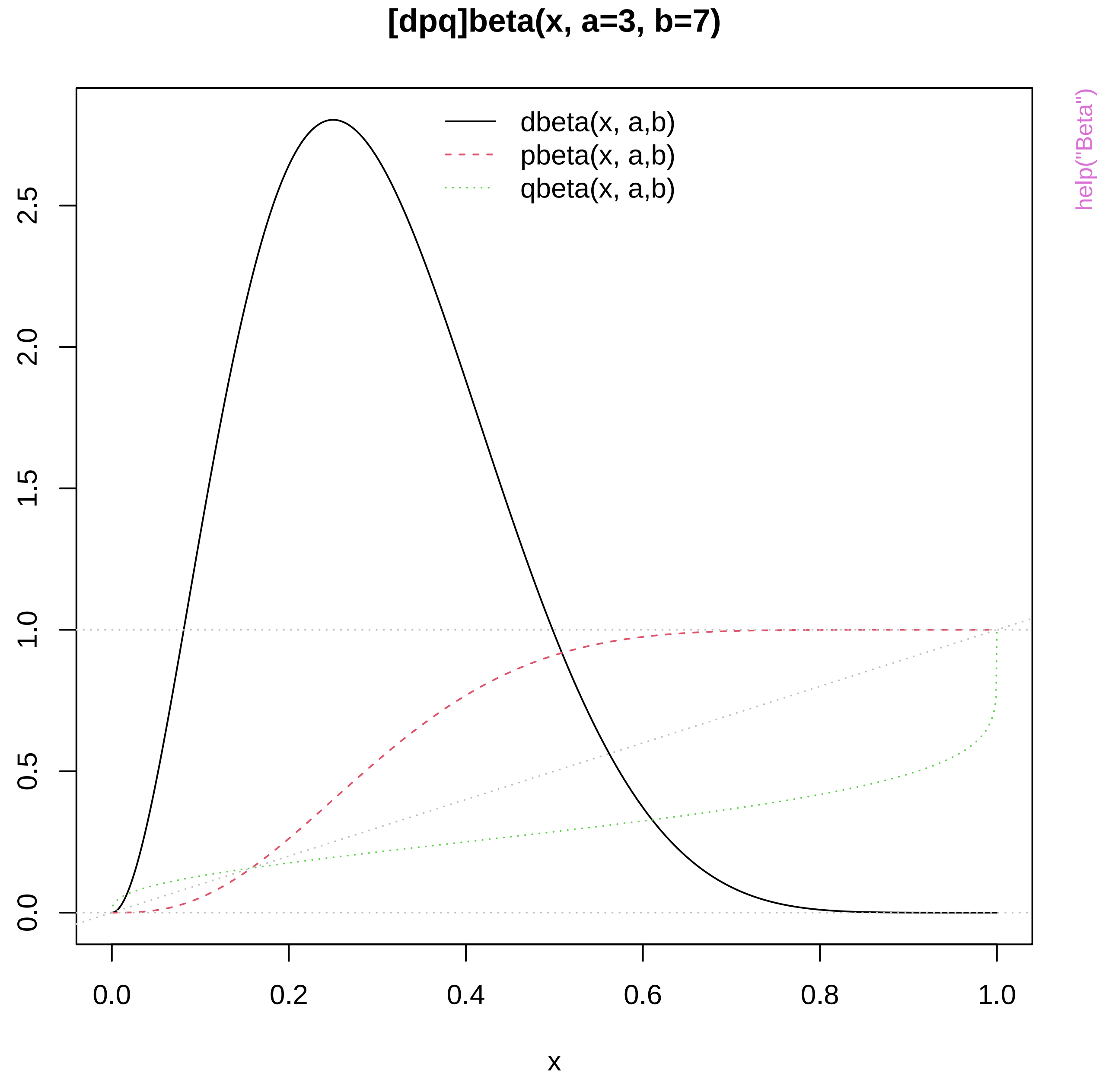
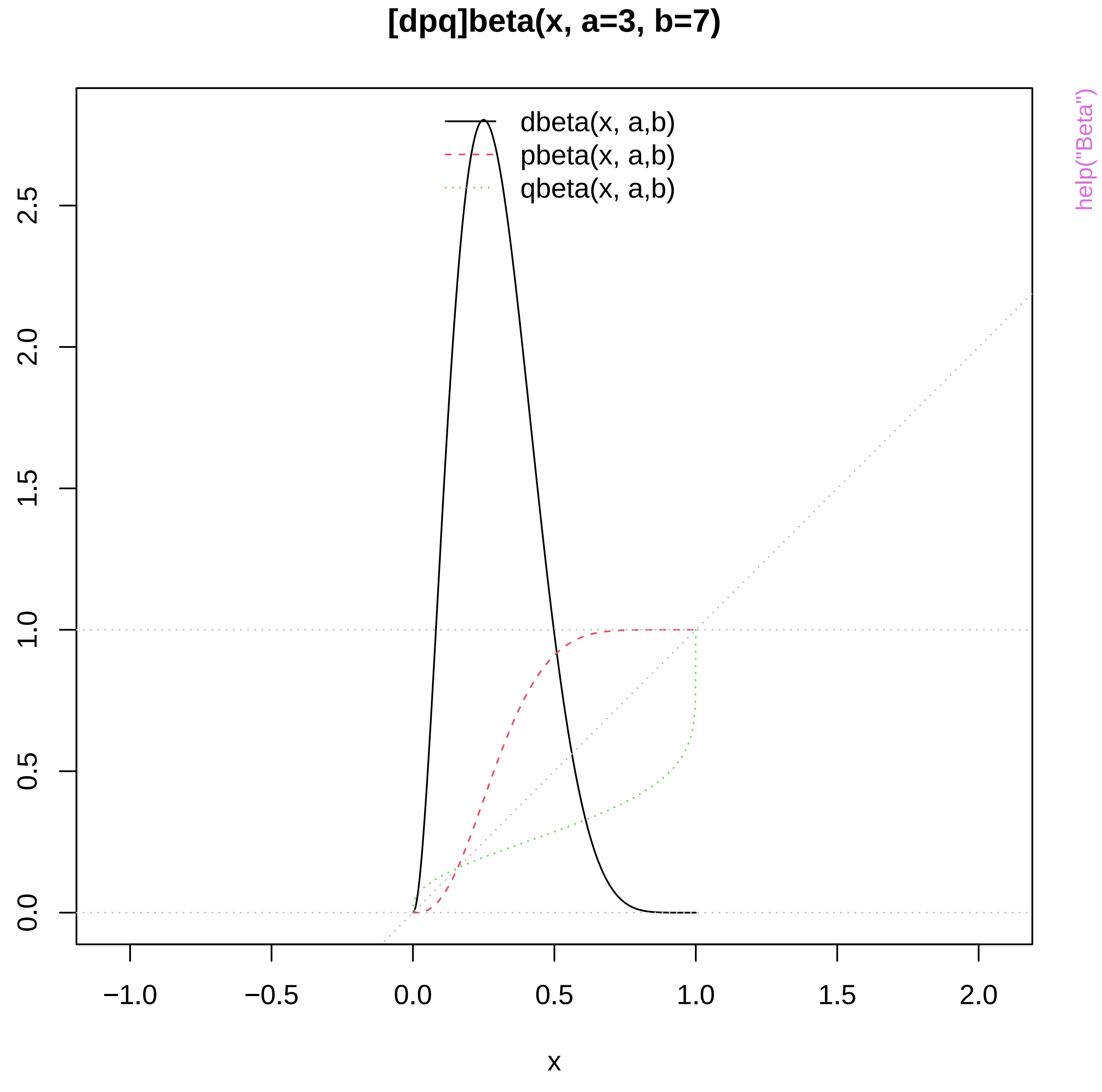


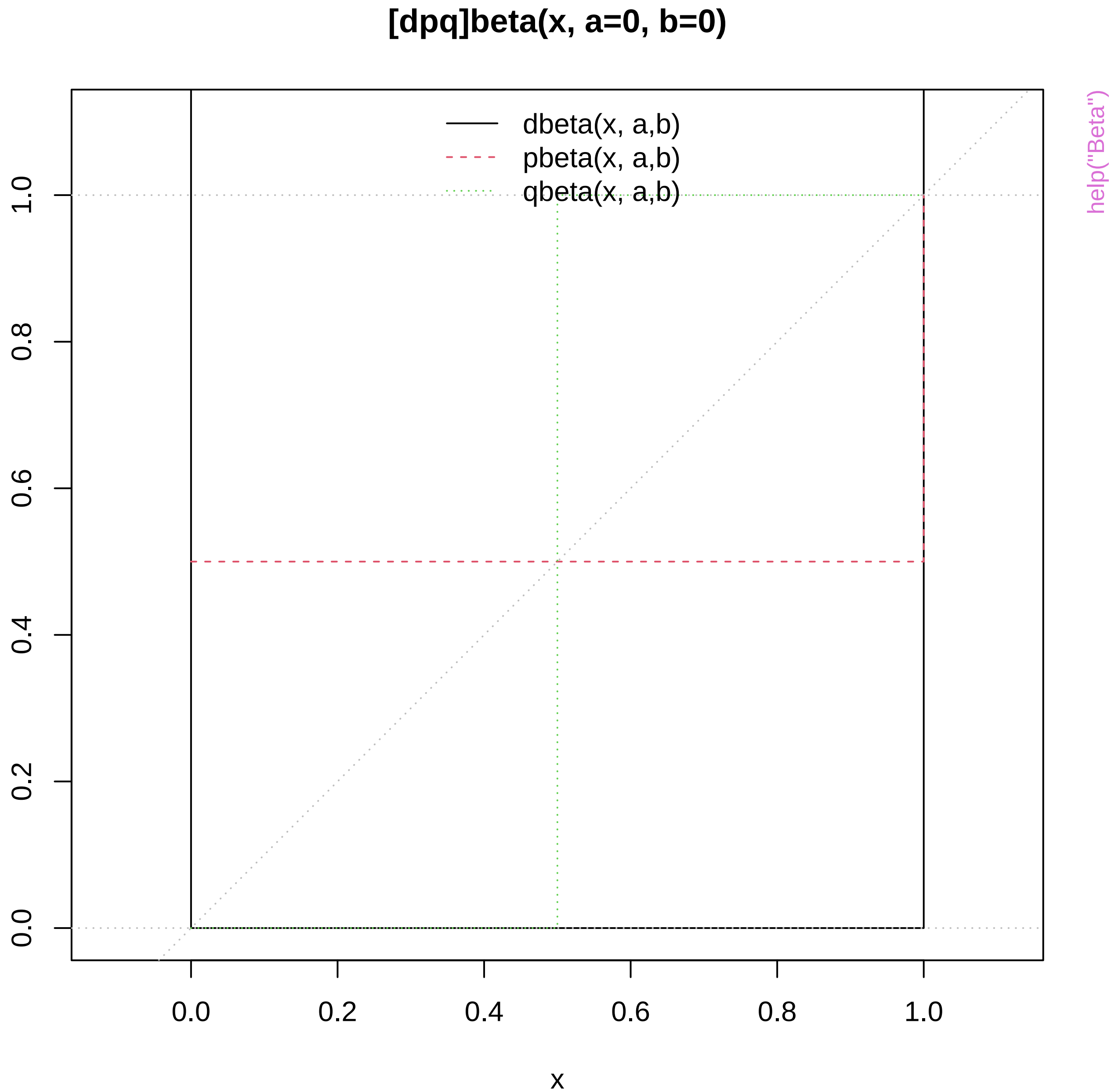
[dpq]beta(x, a=3, b=1)
dbeta(x, a,b)
pbeta(x, a,b)
qbeta(x, a,b)
help("Beta")
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
1.0
x


[dpq]beta(x, a=2, b=4)
dbeta(x, a,b)
pbeta(x, a,b)
qbeta(x, a,b)
help("Beta")
0.0
0.5
1.0
1.5
2.0
0.0
0.2
0.4
0.6
0.8
1.0
x

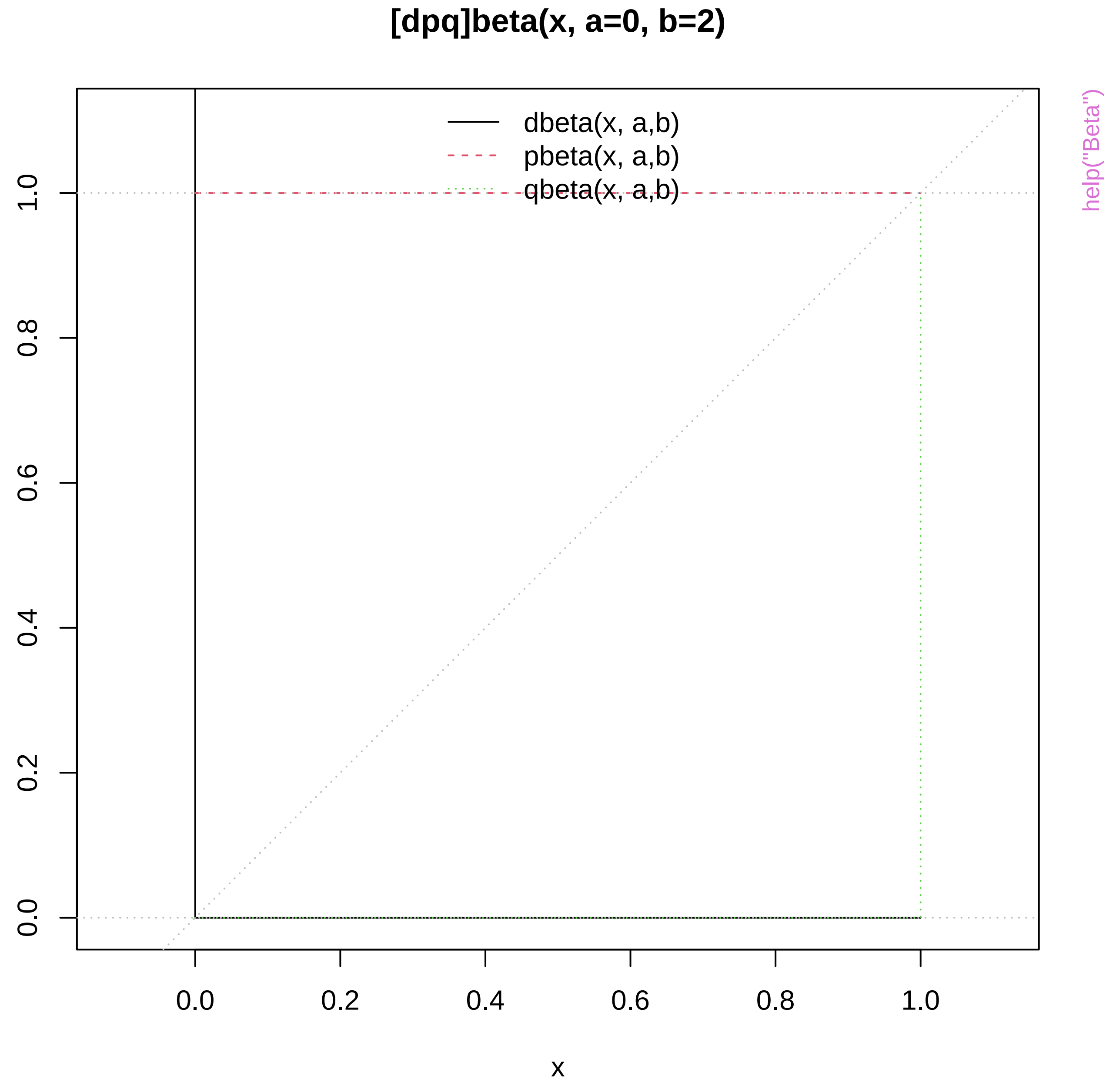

[dpq]beta(x, a=3, b=7)
dbeta(x, a,b)
pbeta(x, a,b)
qbeta(x, a,b)
help("Beta")
0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.2
0.4
0.6
0.8
1.0
x

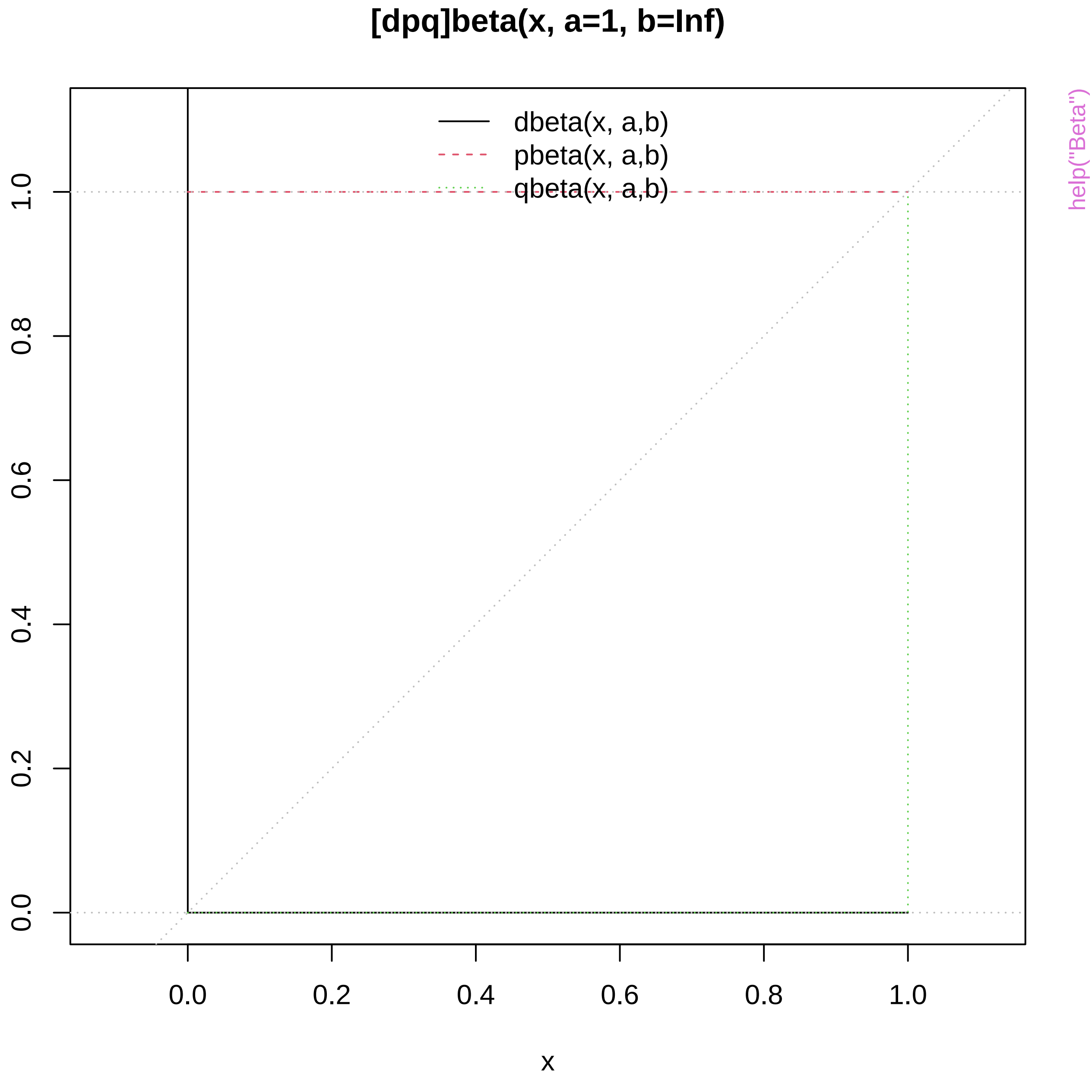


[dpq]beta(x, a=3, b=7)
dbeta(x, a,b)
pbeta(x, a,b)
qbeta(x, a,b)
help("Beta")
0.0
0.5
1.0
1.5
2.0
2.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
x

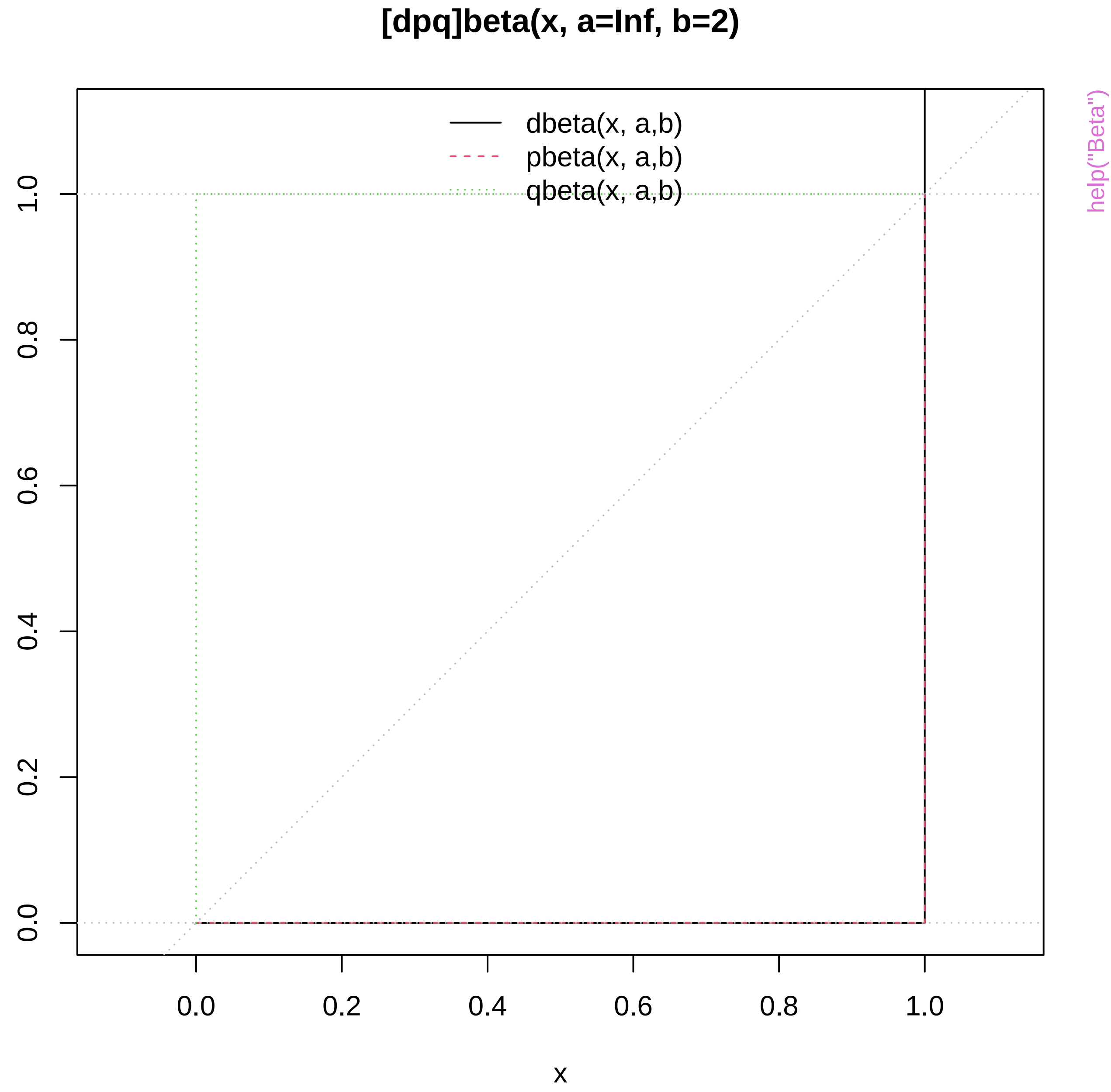


[dpq]beta(x, a=0, b=0)
dbeta(x, a,b)
pbeta(x, a,b)
qbeta(x, a,b)
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
x
help("Beta")

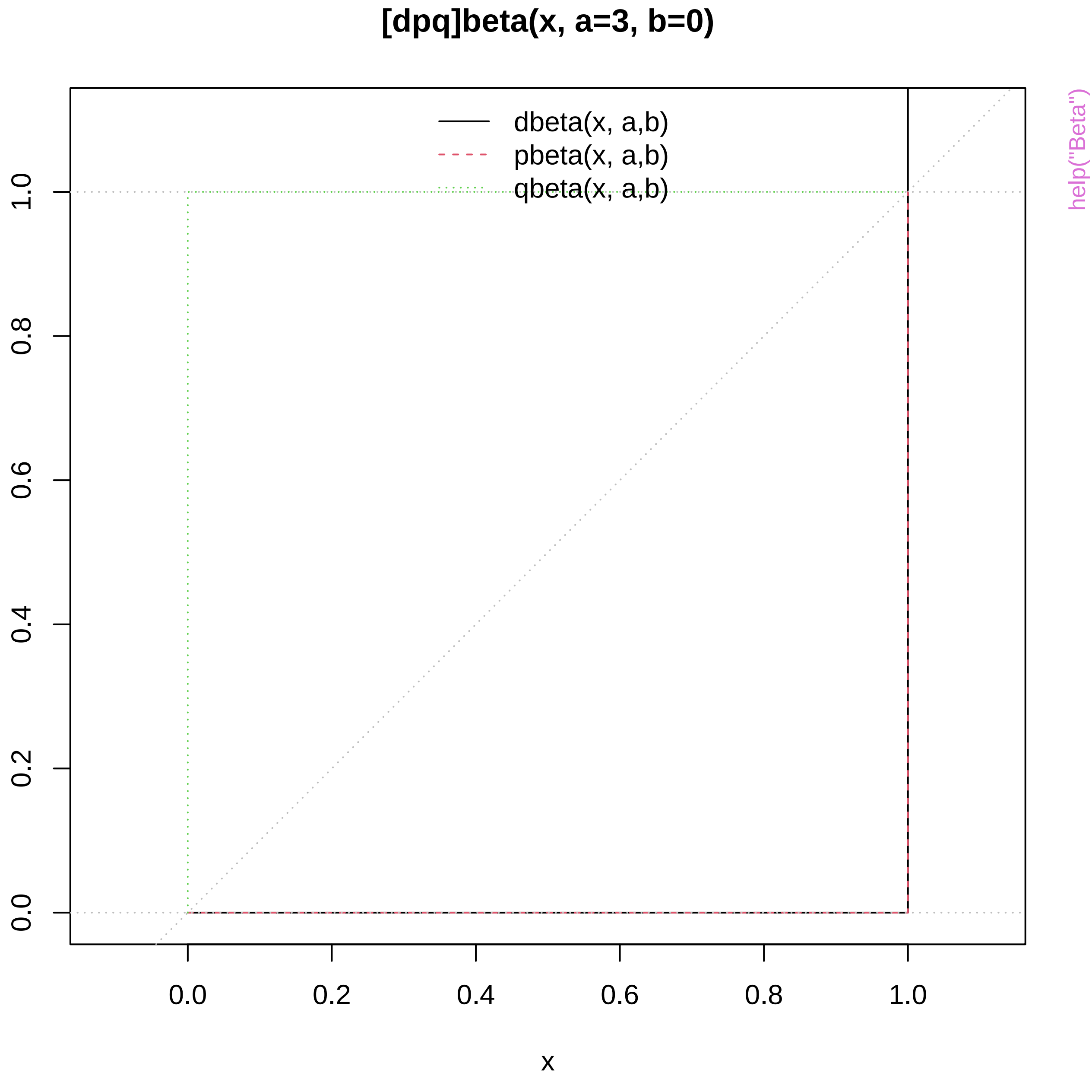

[dpq]beta(x, a=0, b=2)
dbeta(x, a,b)
pbeta(x, a,b)
qbeta(x, a,b)
0.0
0.2
0.4
0.6
0.8
1.0
x
help("Beta")



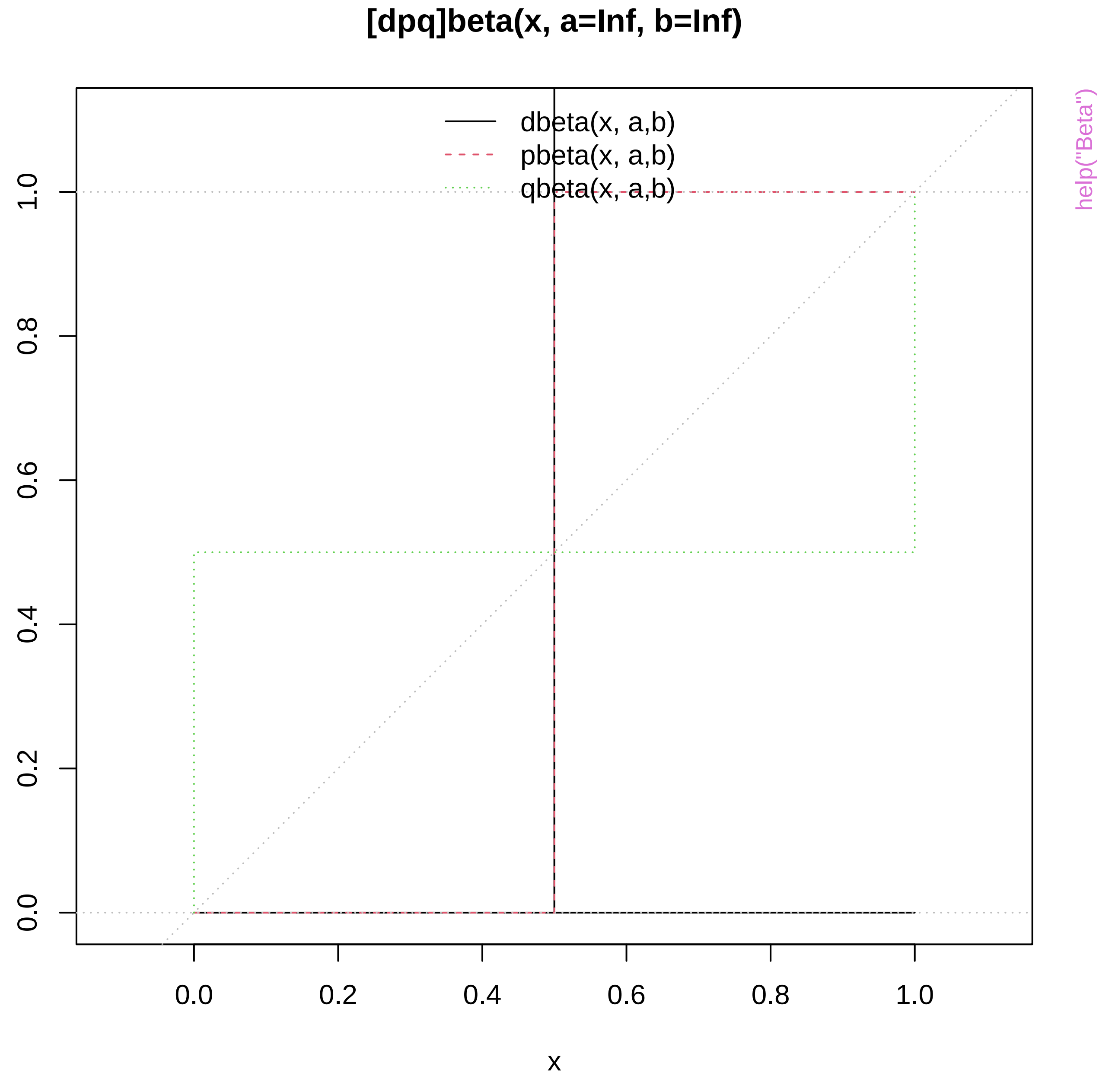
[dpq]beta(x, a=1, b=Inf)
dbeta(x, a,b)
pbeta(x, a,b)
qbeta(x, a,b)
help("Beta")
0.0
0.2
0.4
0.6
0.8
1.0
x



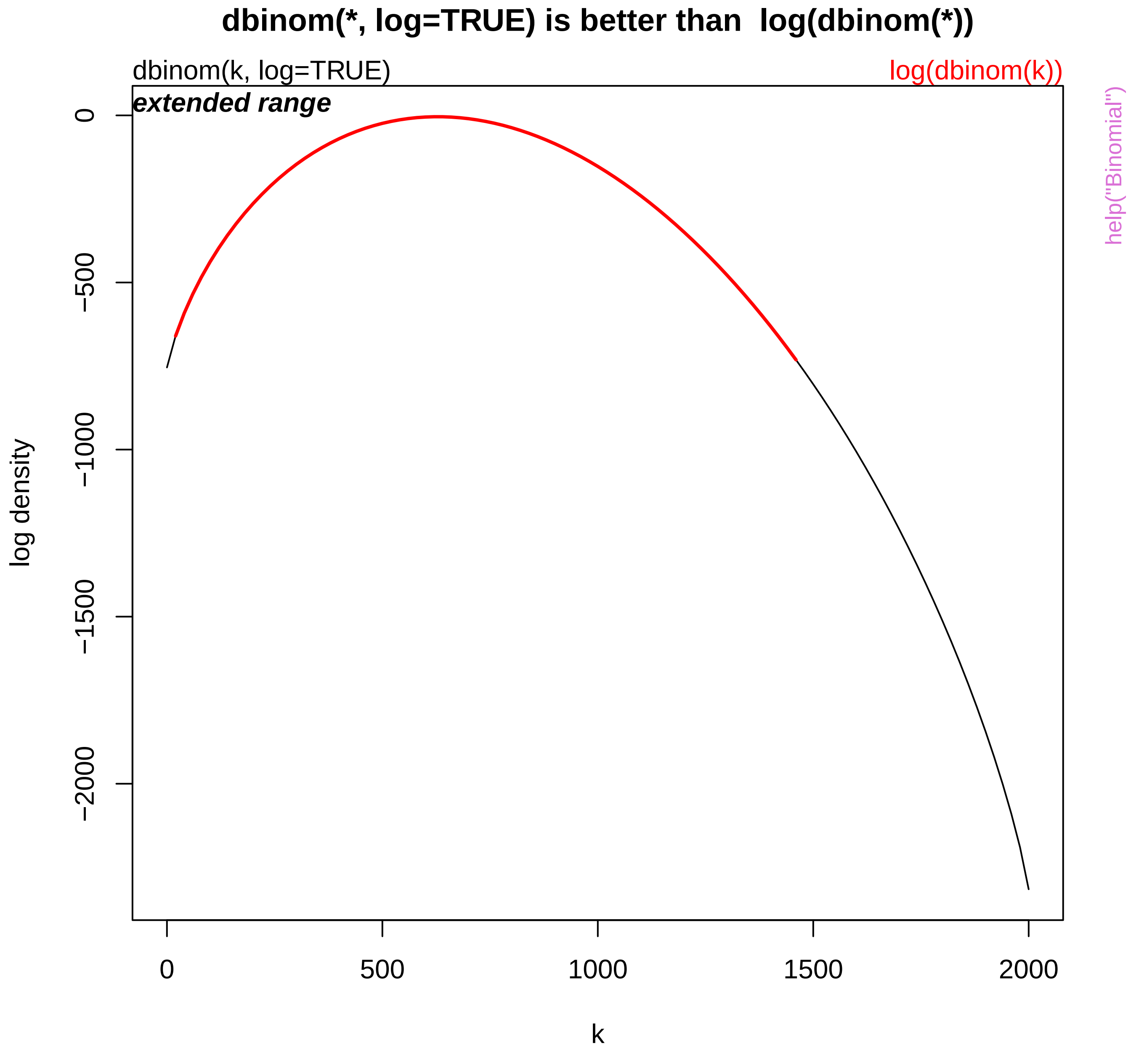
[dpq]beta(x, a=Inf, b=2)
dbeta(x, a,b)
pbeta(x, a,b)
qbeta(x, a,b)
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
x
help("Beta")

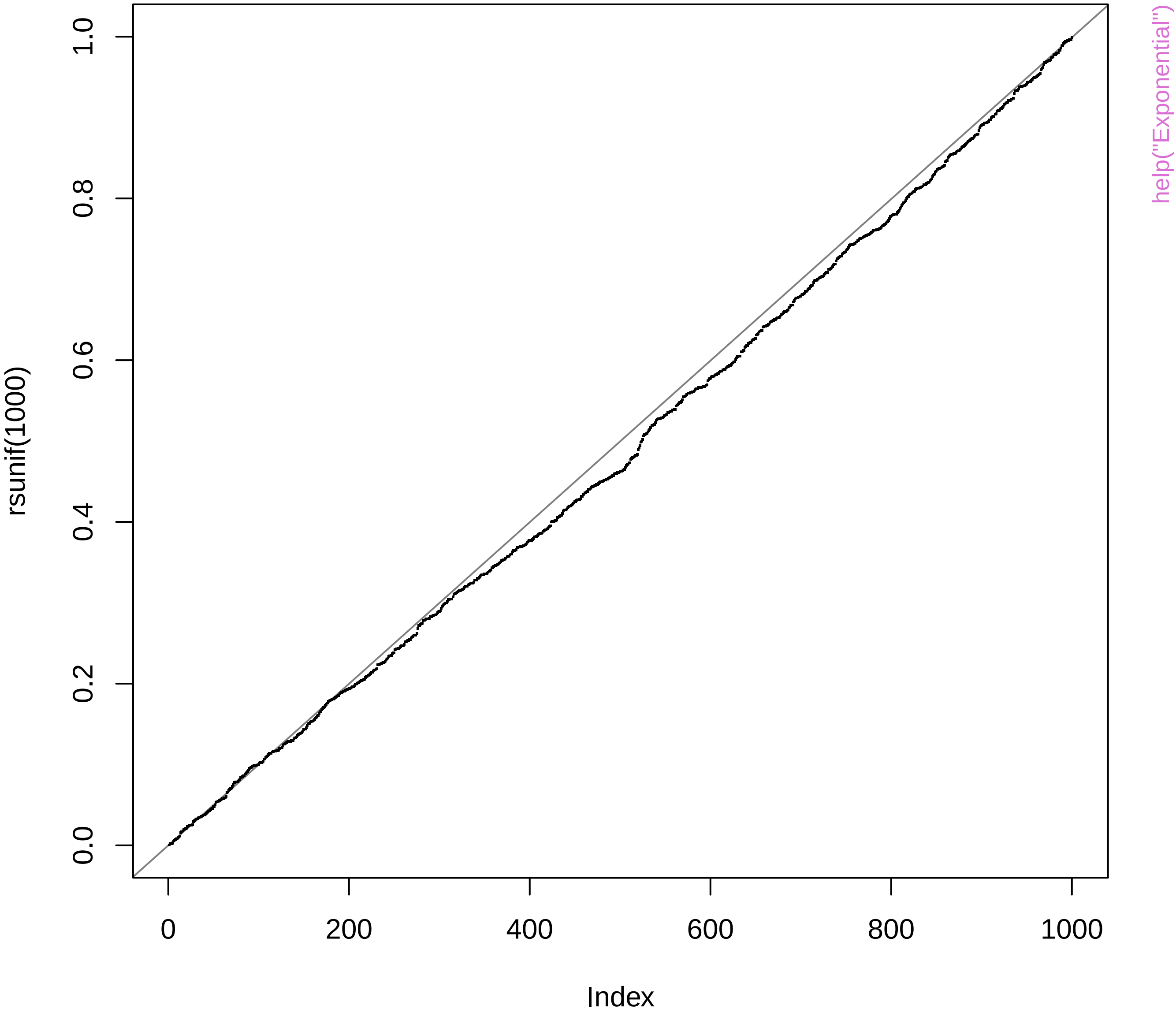


[dpq]beta(x, a=3, b=0)
dbeta(x, a,b)
pbeta(x, a,b)
qbeta(x, a,b)
help("Beta")
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
x

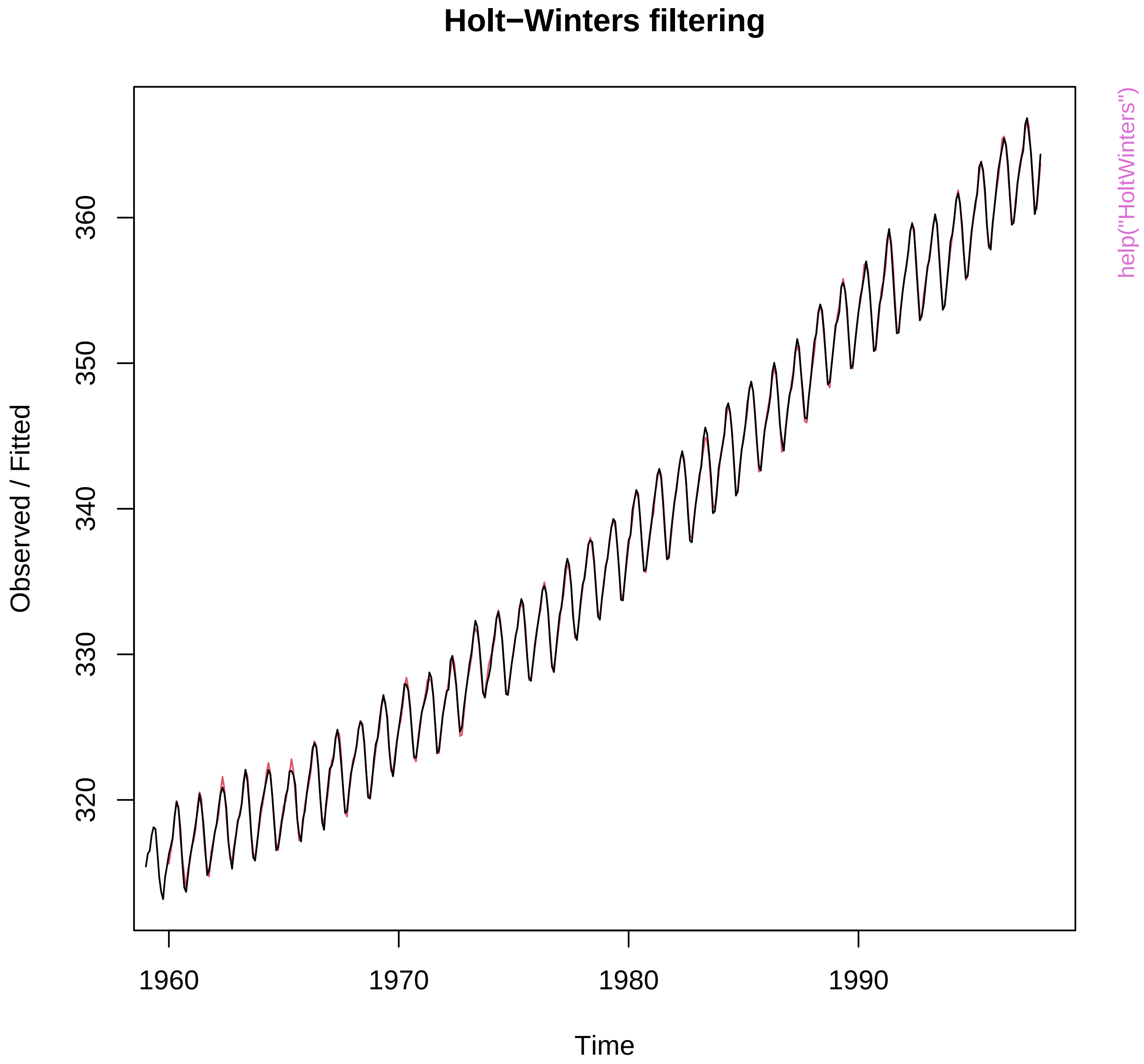


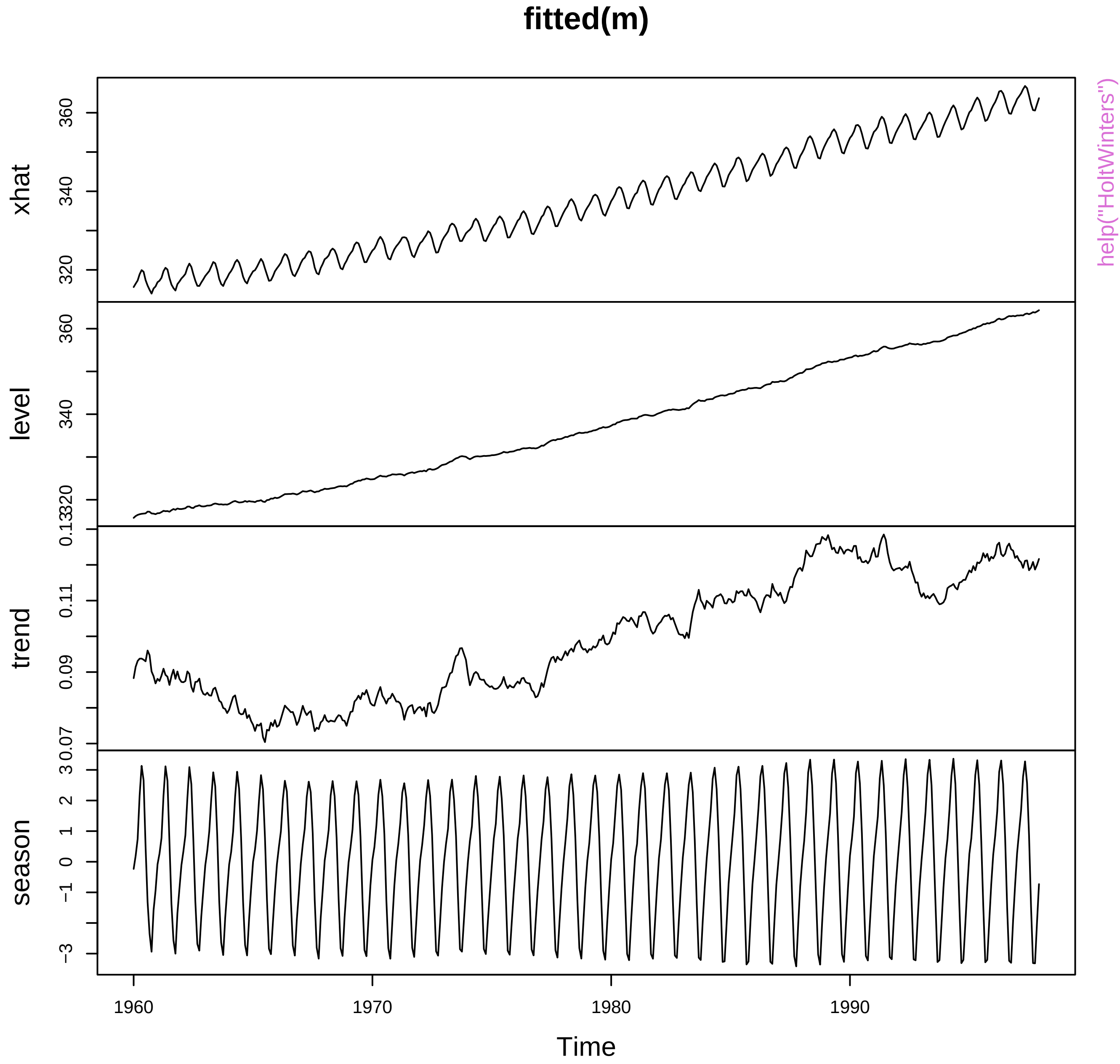
[dpq]beta(x, a=Inf, b=Inf)
dbeta(x, a,b)
pbeta(x, a,b)
qbeta(x, a,b)
0.0
0.2
0.4
0.6
0.8
1.0
x
help("Beta")

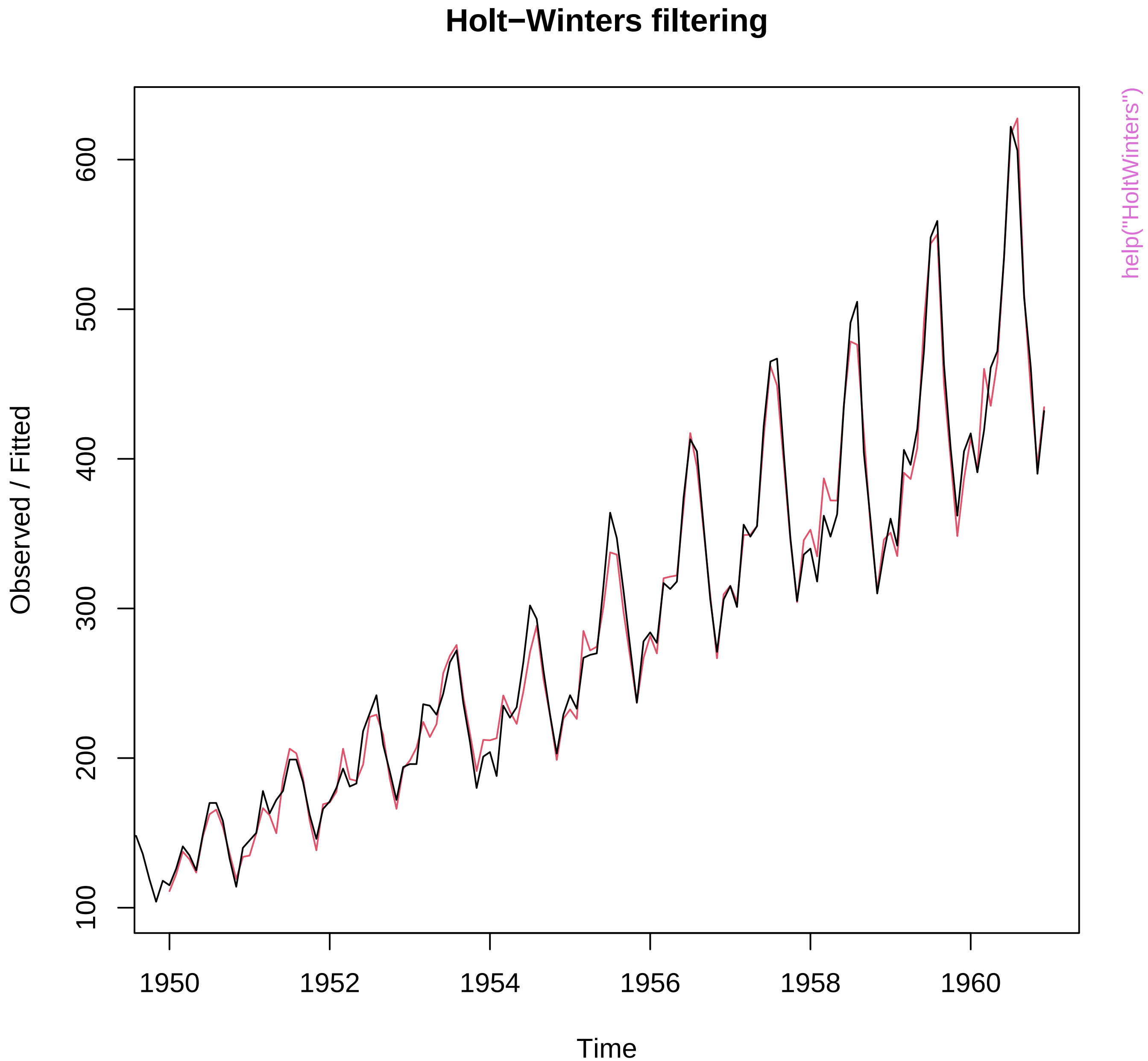

dbinom(*, log=TRUE) is better than log(dbinom(*))
dbinom(k, log=TRUE)
log(dbinom(k))
extended range
log density
0
−500
−1000
−1500
−2000
0
500
1000
1500
2000
k
help("Binomial")



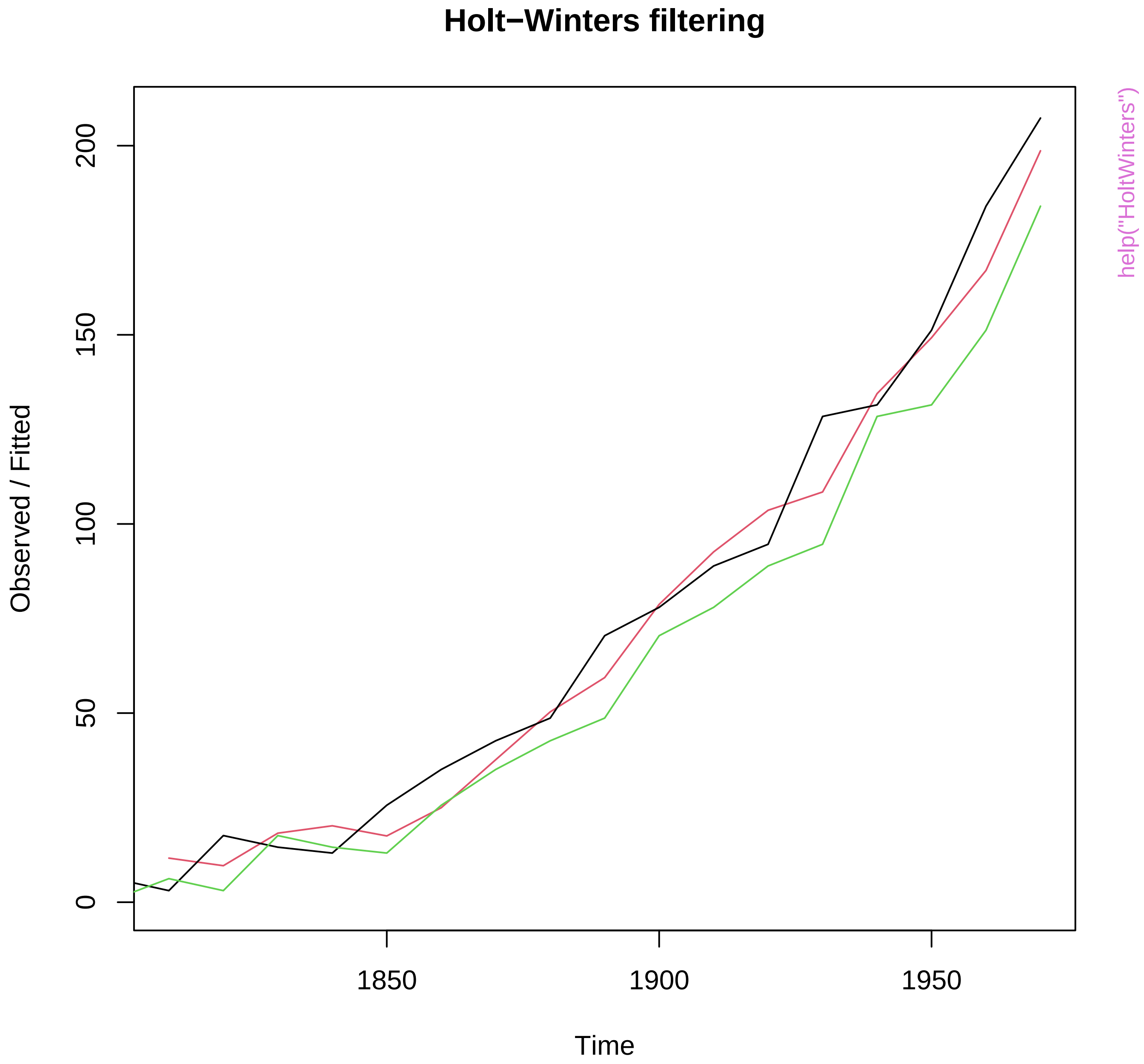
rsunif(1000)
0.0
0.2
0.4
0.6
0.8
1.0
0
200
400
600
800
1000
Index
help("Exponential")


Holt−Winters filtering
Observed / Fitted
360
350
340
330
320
1960
1970
1980
1990
Time
help("HoltWinters")

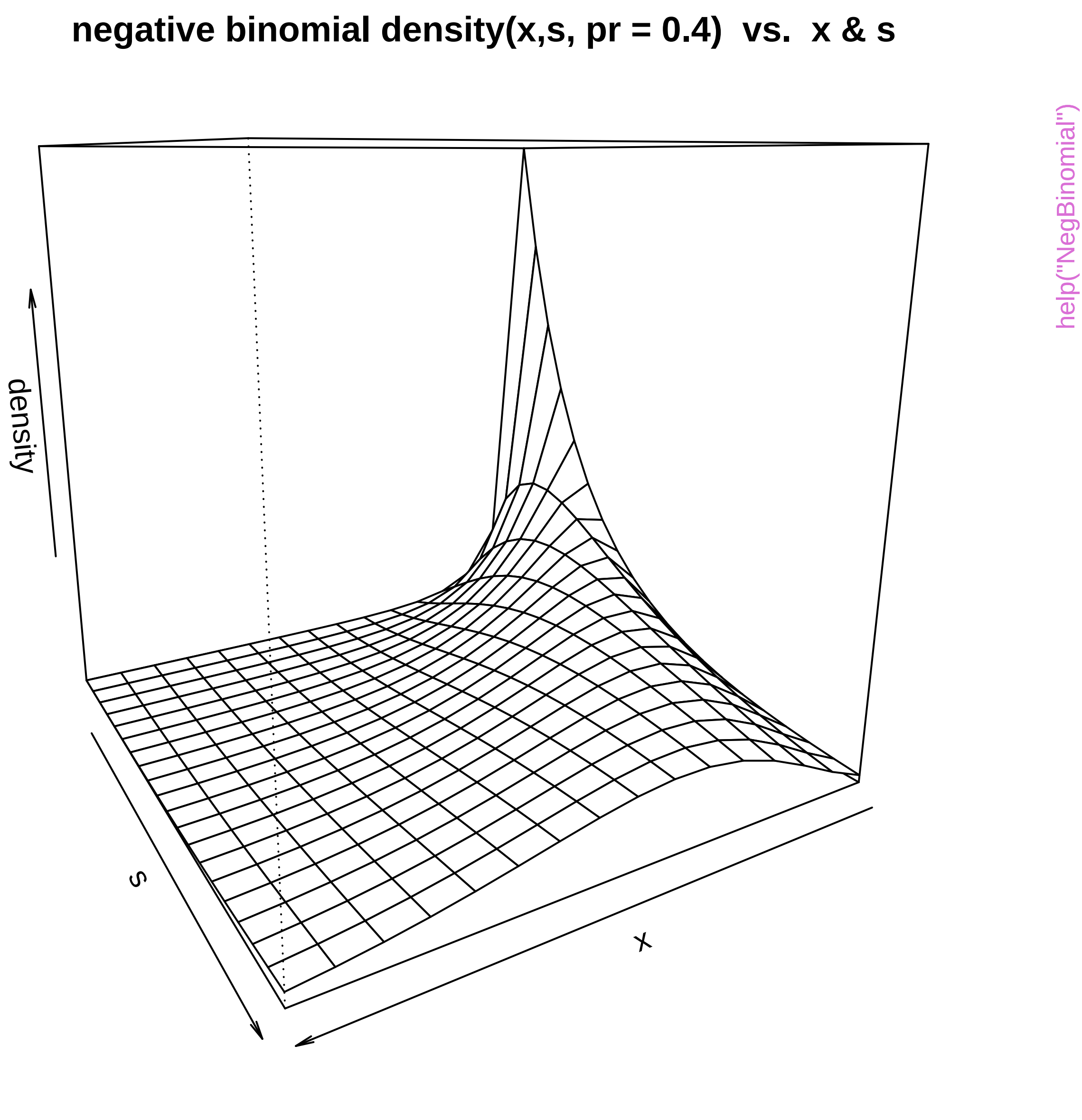

fitted(m)
xhat
level
trend
season
Time
1960
1970
1980
1990
help("HoltWinters")

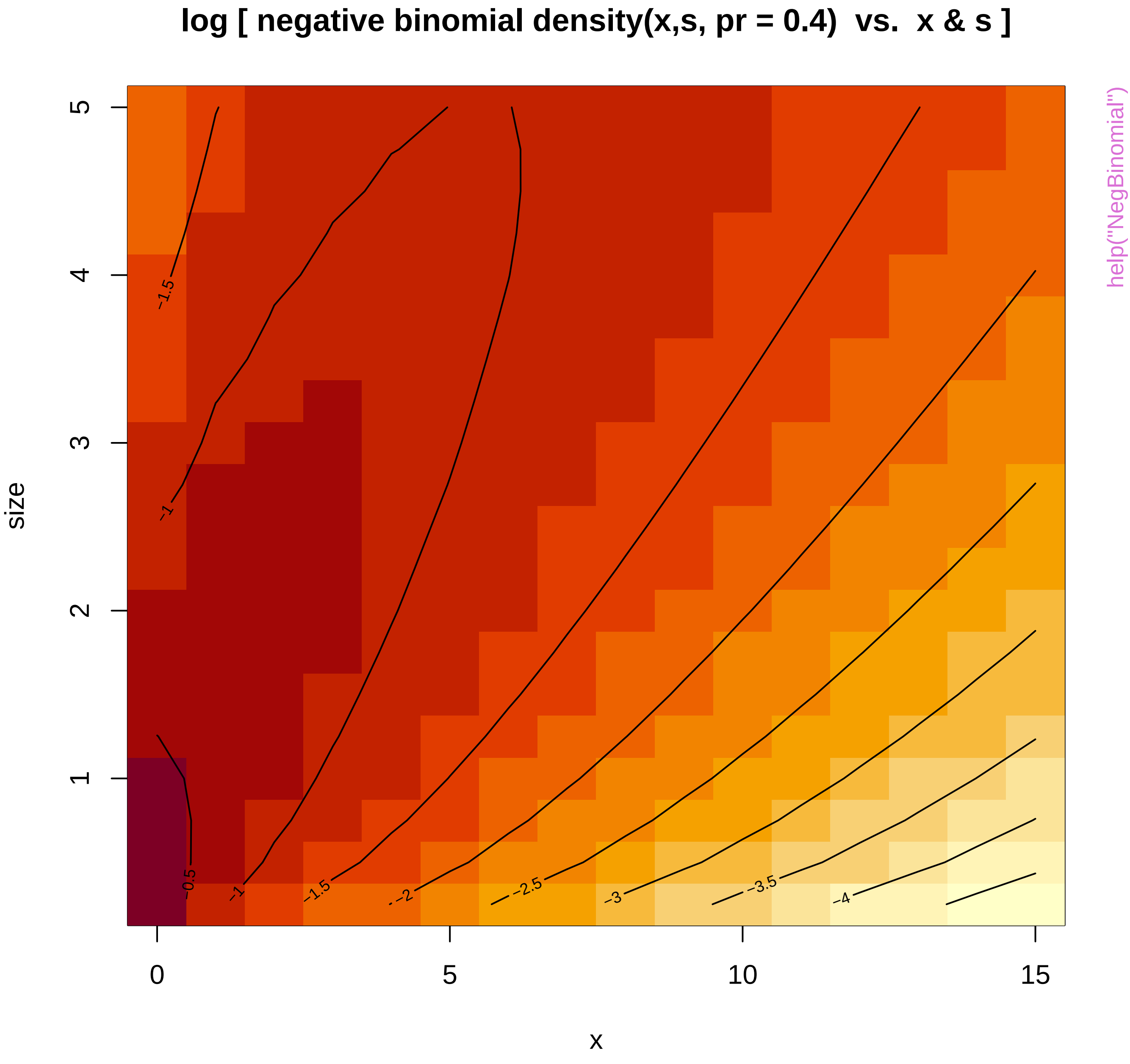


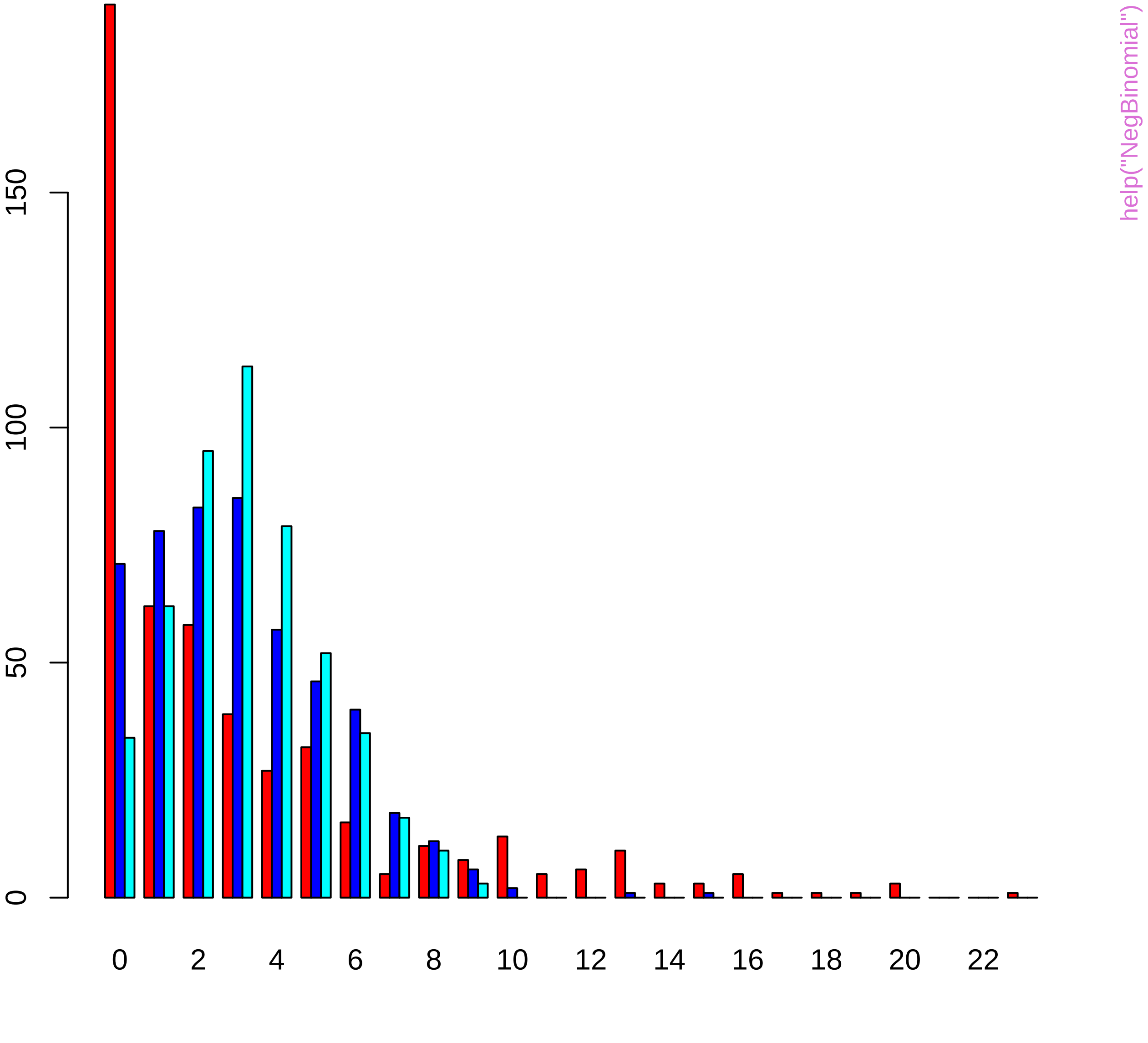
Holt−Winters filtering
Observed / Fitted
100
200
300
400
500
600
1950
1952
1954
1956
1958
1960
Time
help("HoltWinters")



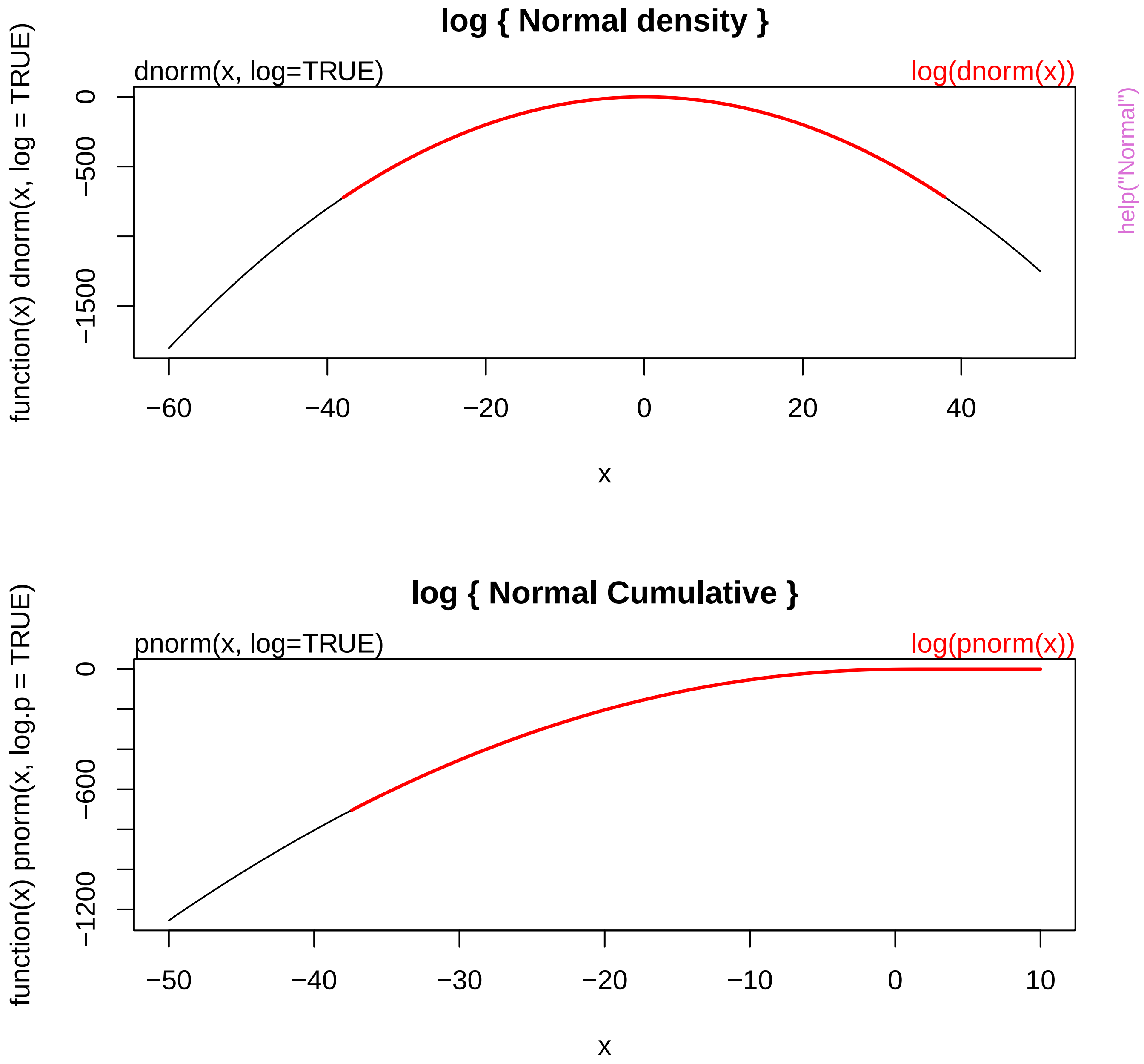
Holt−Winters filtering
Observed / Fitted
0
50
100
150
200
1850
1900
1950
Time
help("HoltWinters")


negative binomial density(x,s, pr = 0.4) vs. x & s
density
s
x
help("NegBinomial")


log [negative binomial density(x,s, pr = 0.4) vs. x & s]
size
x
0
5
10
15
1
2
3
4
5
−0.5
−1
−1.5
−2
−2.5
−3
−3.5
−4
help("NegBinomial")

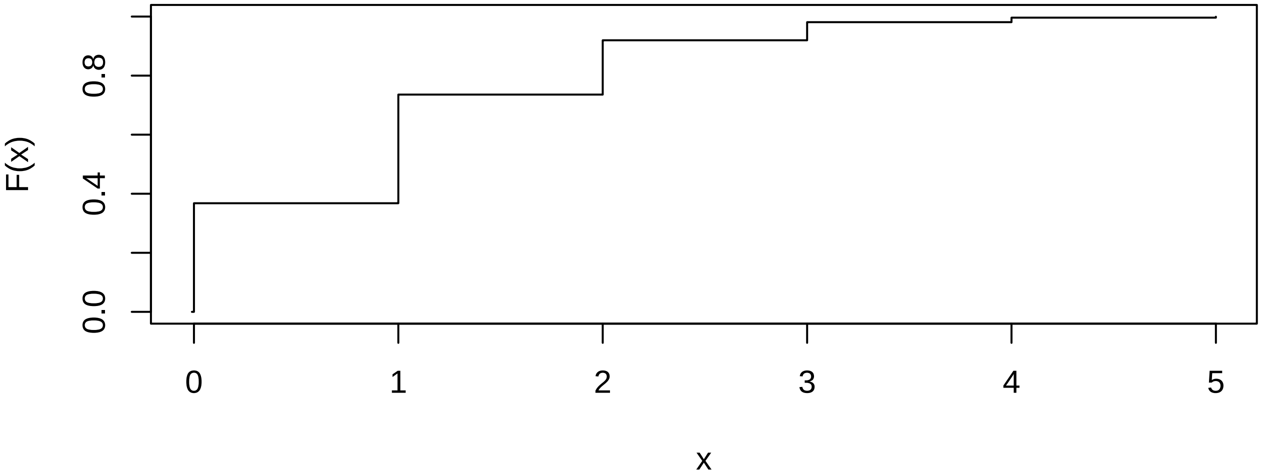


0
50
100
150
0
2
4
6
8
10
12
14
16
18
20
22
help("NegBinomial")



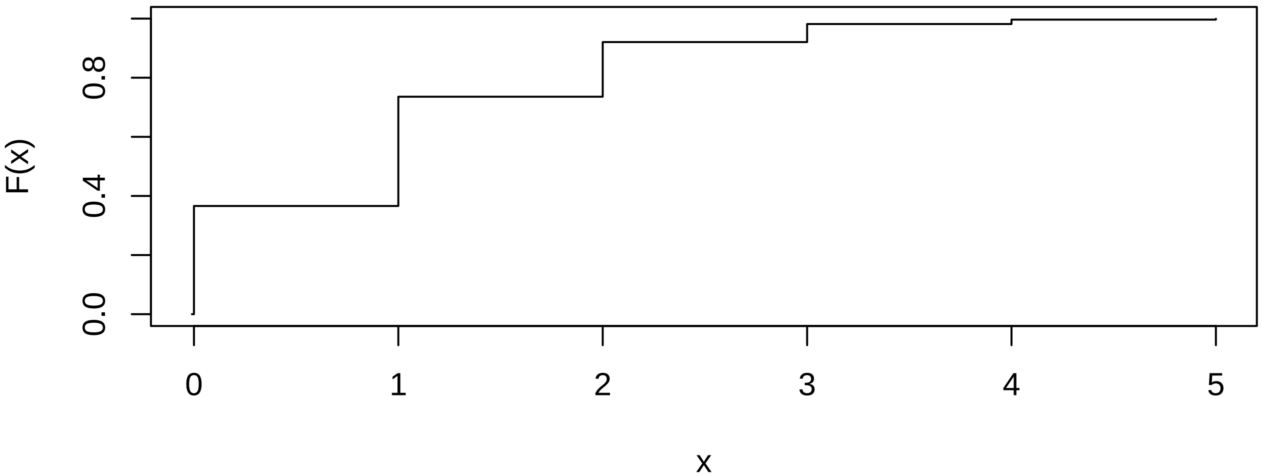
log { Normal density }
dnorm(x, log=TRUE)
log(dnorm(x))
function(x) dnorm(x, log = TRUE)
0
−500
−1500
−60
−40
−20
0
20
40
x
help("Normal")
log { Normal Cumulative }
pnorm(x, log=TRUE)
log(pnorm(x))
function(x) pnorm(x, log.p = TRUE)
0
−600
−1200
−50
−40
−30
−20
−10
0
10
x

Poisson(1) CDF



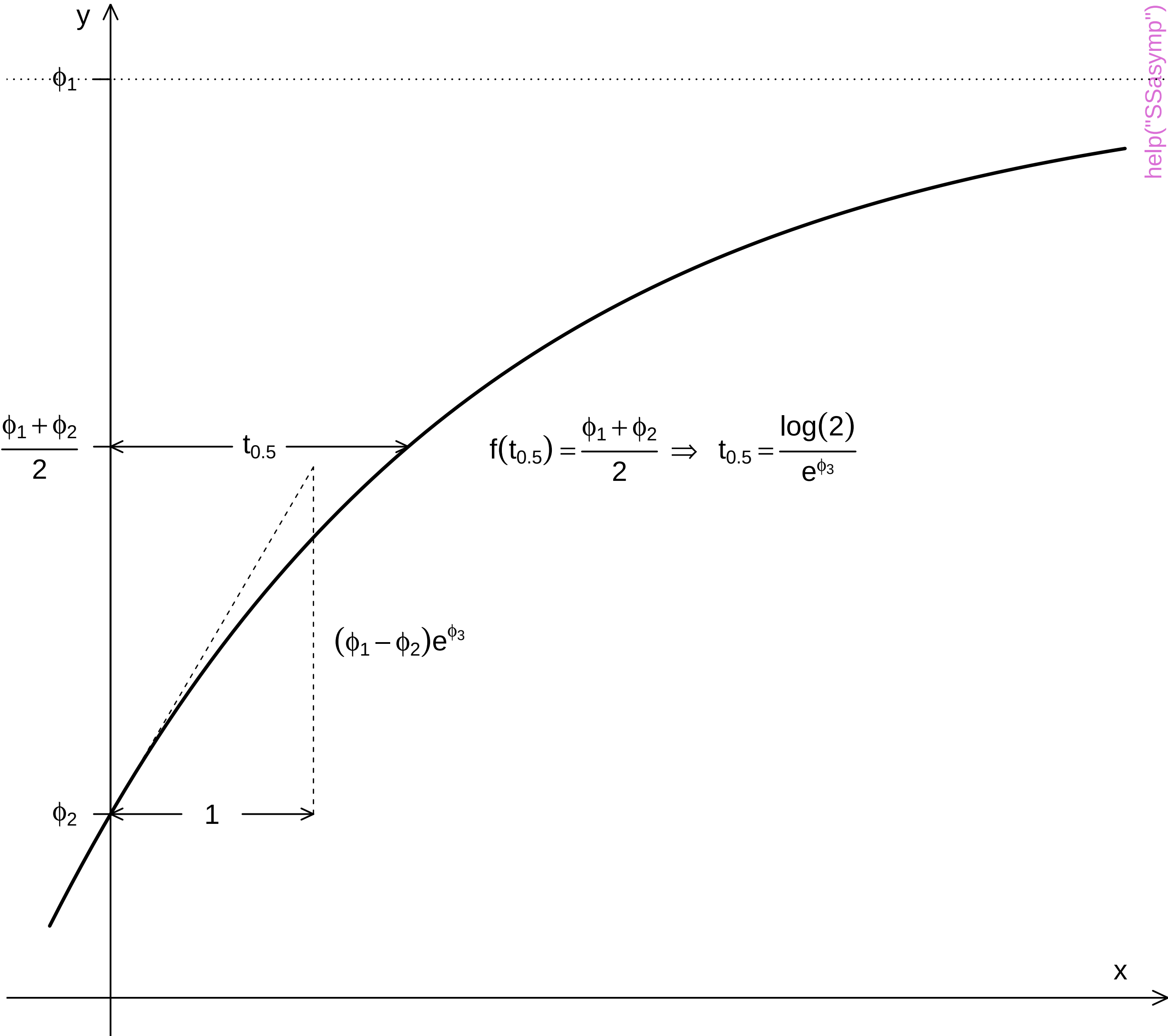
help("Poisson")

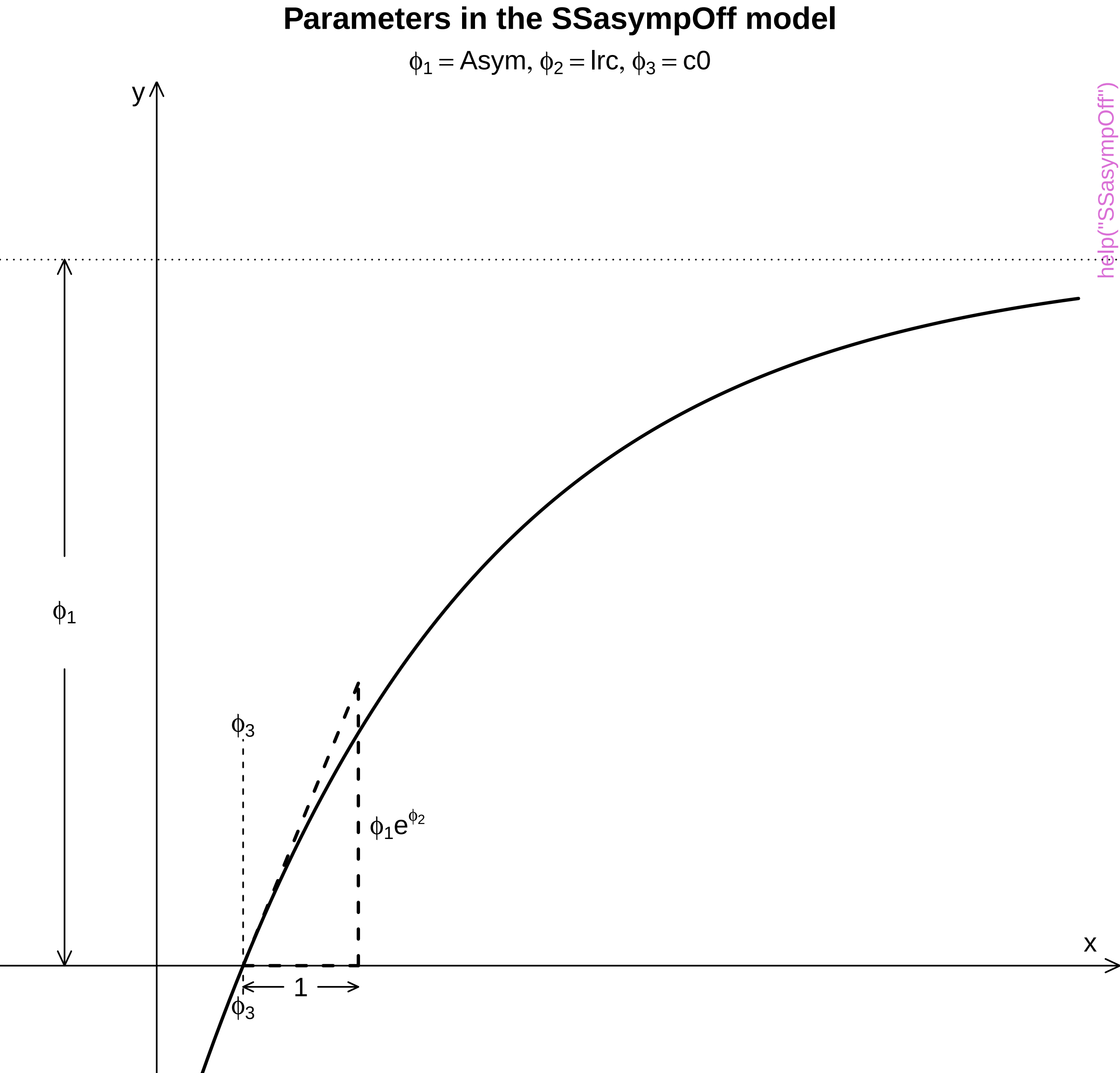
Binomial(100, 0.01) CDF



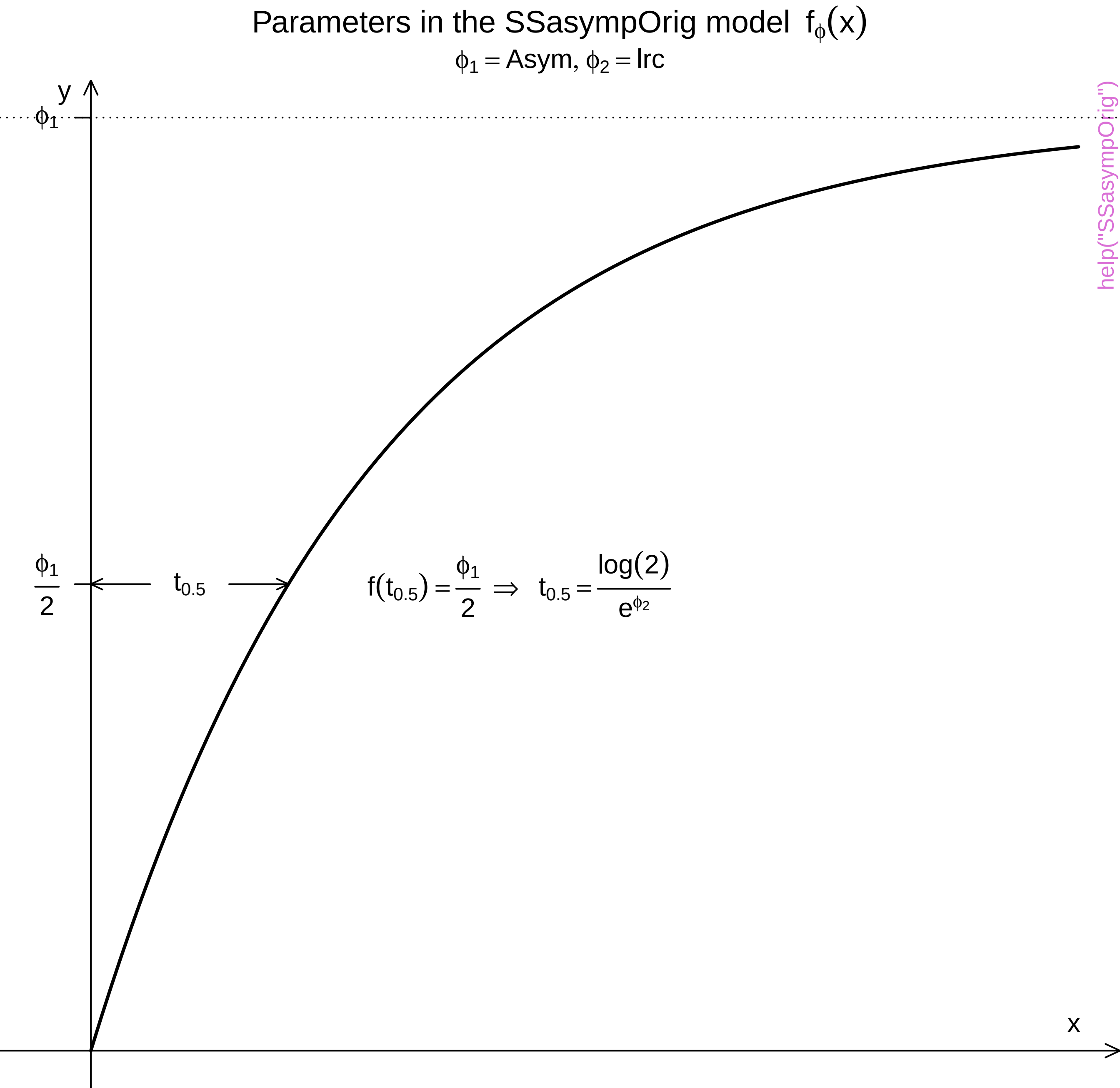
Parameters in the SSasymp model $f_\phi(x) = \phi_1 + (\phi_2 - \phi_1)\, e^{-e^{\phi_3} x}$

$\phi_1 = \text{Asym},\ \phi_2 = \text{R0},\ \phi_3 = \text{lrc}$

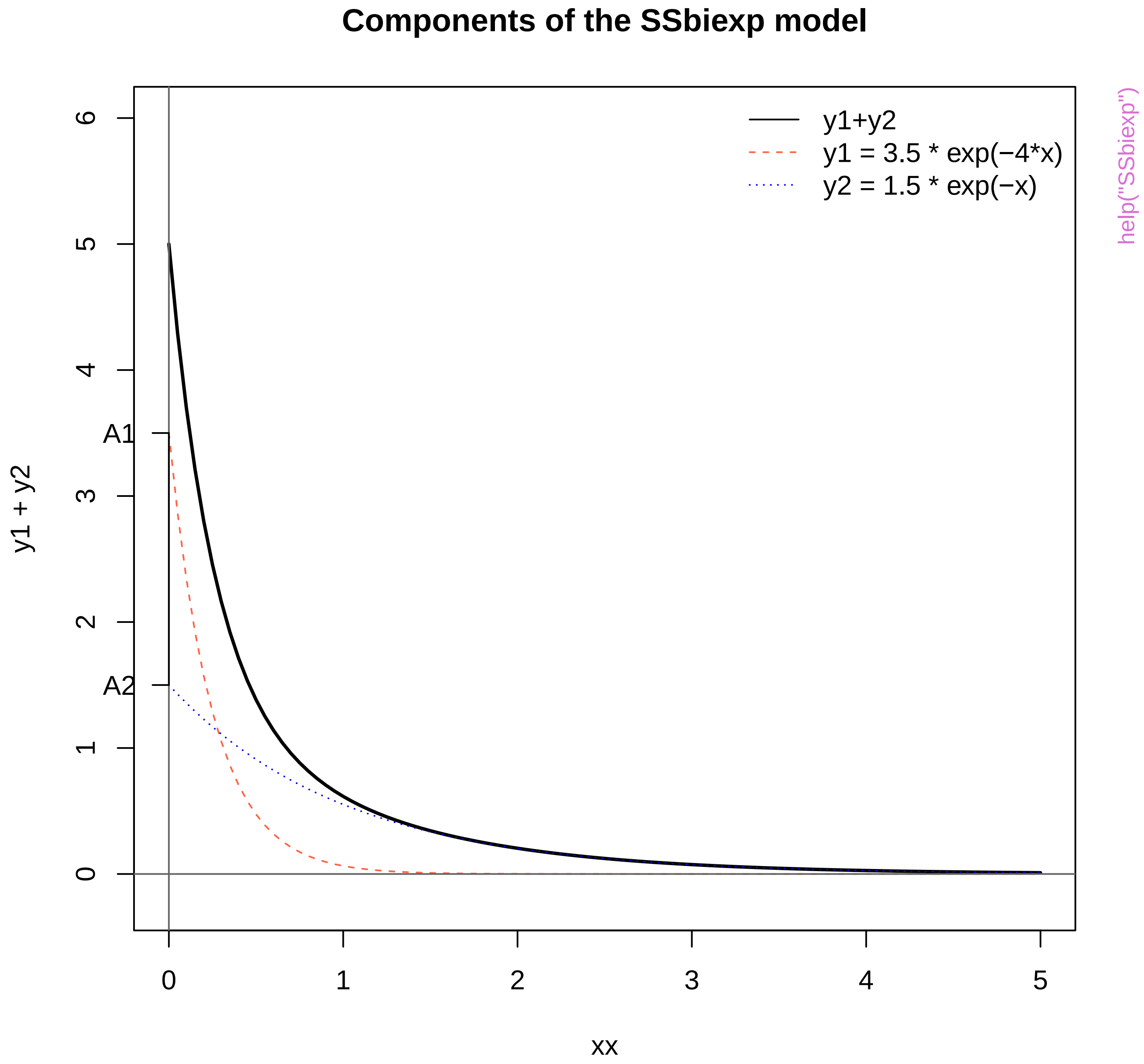


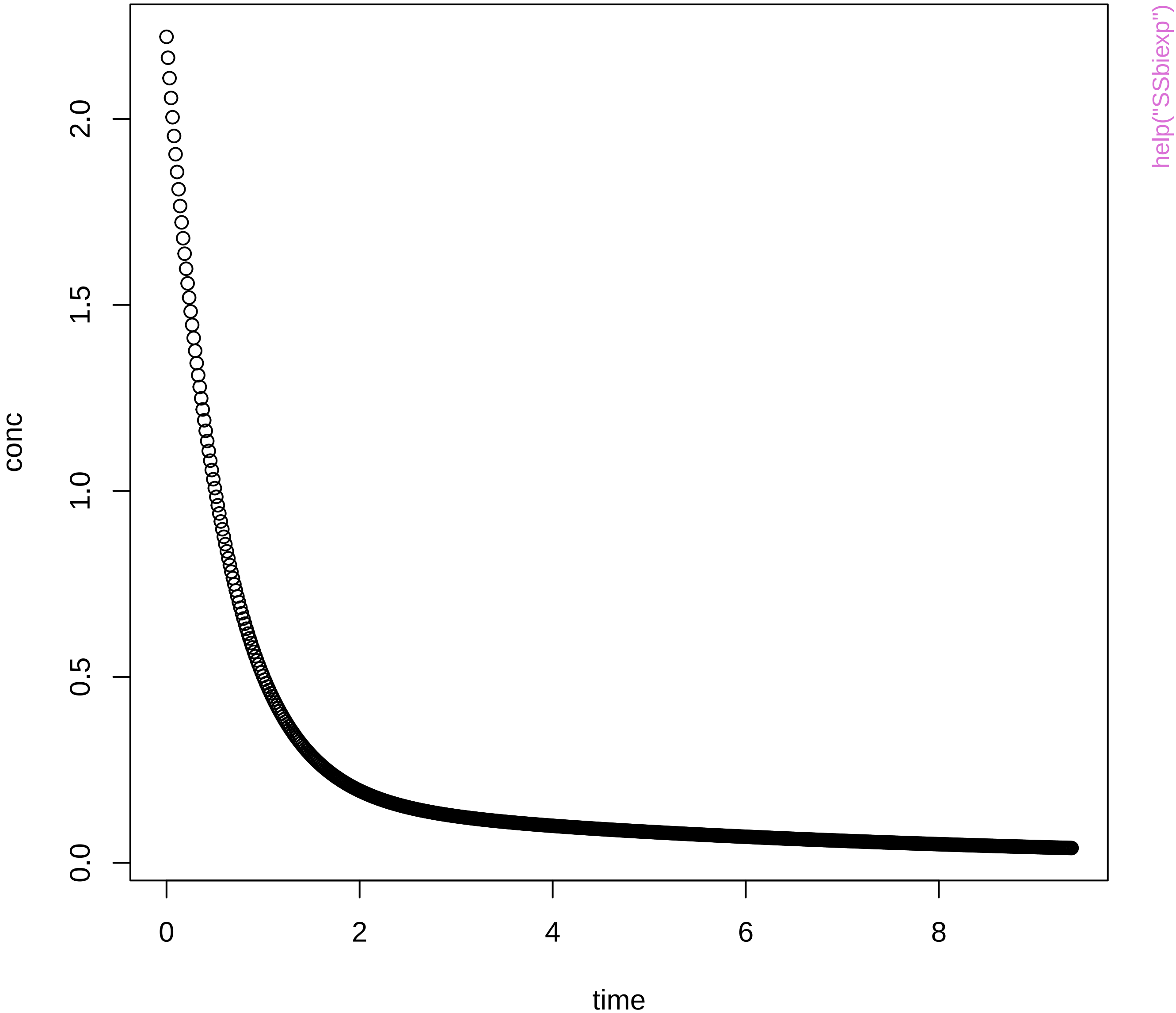


Parameters in the SSasympOff model
$\phi_1 = \text{Asym}, \phi_2 = \text{lrc}, \phi_3 = \text{c0}$
y
x
ϕ_1
ϕ_3
$\phi_1 e^{\phi_2}$
1
ϕ_3
help("SSasympOff")

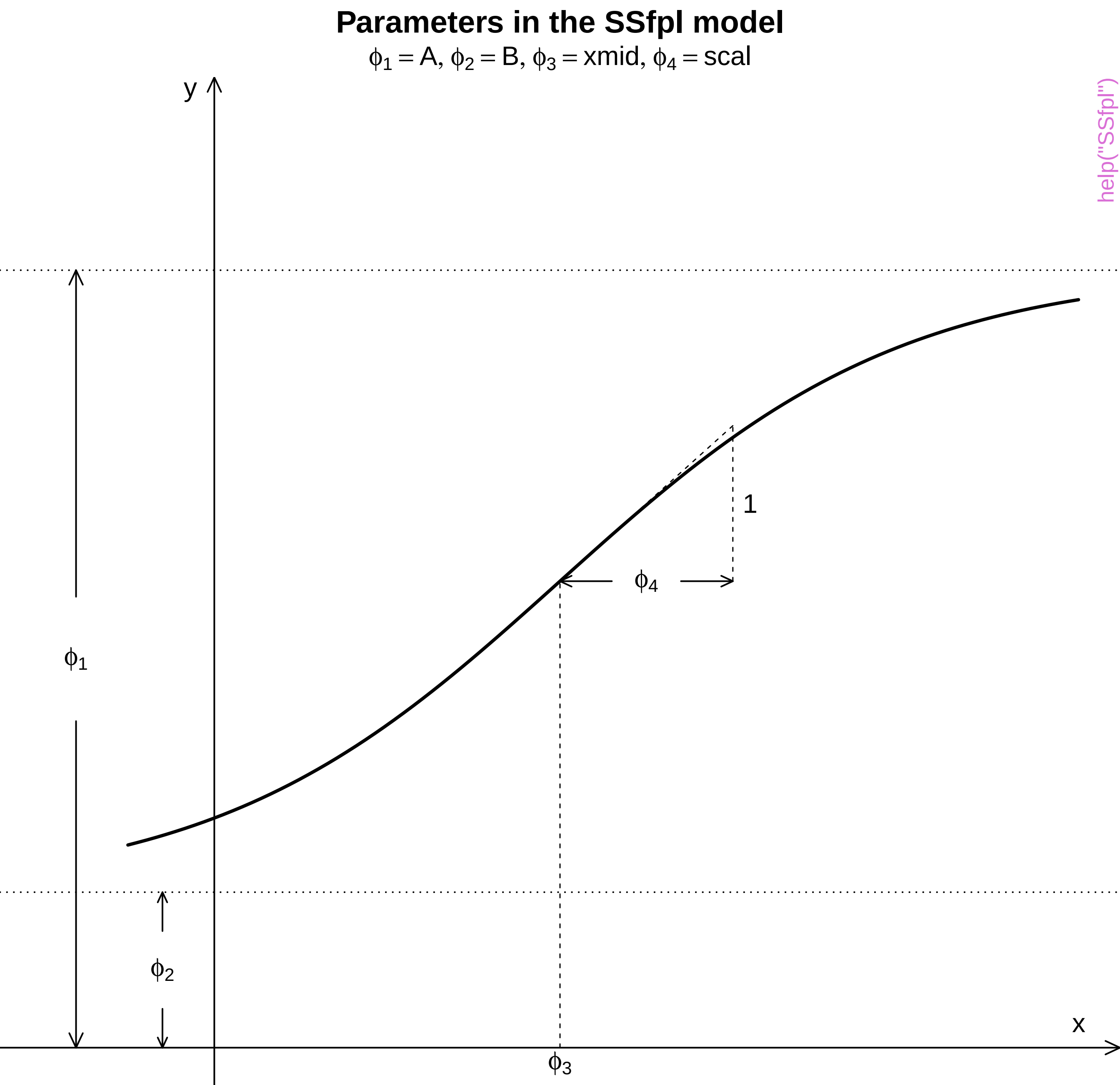


Parameters in the SSasympOrig model $f_\phi(x)$
$\phi_1 = \text{Asym}, \phi_2 = \text{lrc}$
y
ϕ_1
$\frac{\phi_1}{2}$
$t_{0.5}$
$f(t_{0.5}) = \frac{\phi_1}{2} \Rightarrow t_{0.5} = \frac{\log(2)}{e^{\phi_2}}$
x
help("SSasympOrig")

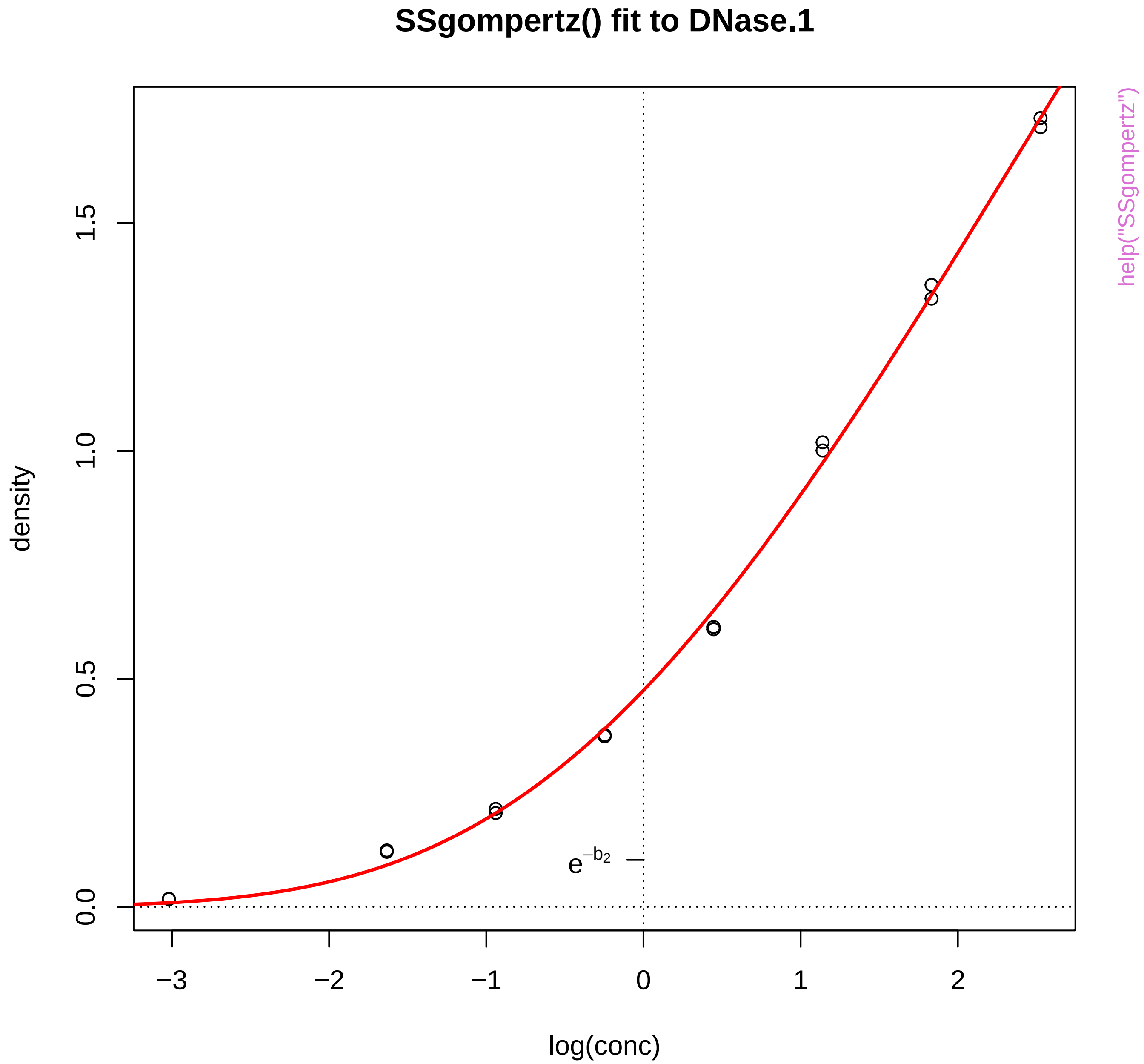

Components of the SSbiexp model
y1+y2
y1 = 3.5 * exp(−4*x)
y2 = 1.5 * exp(−x)
A1
A2
0
1
2
3
4
5
6
y1 + y2
xx
help("SSbiexp")



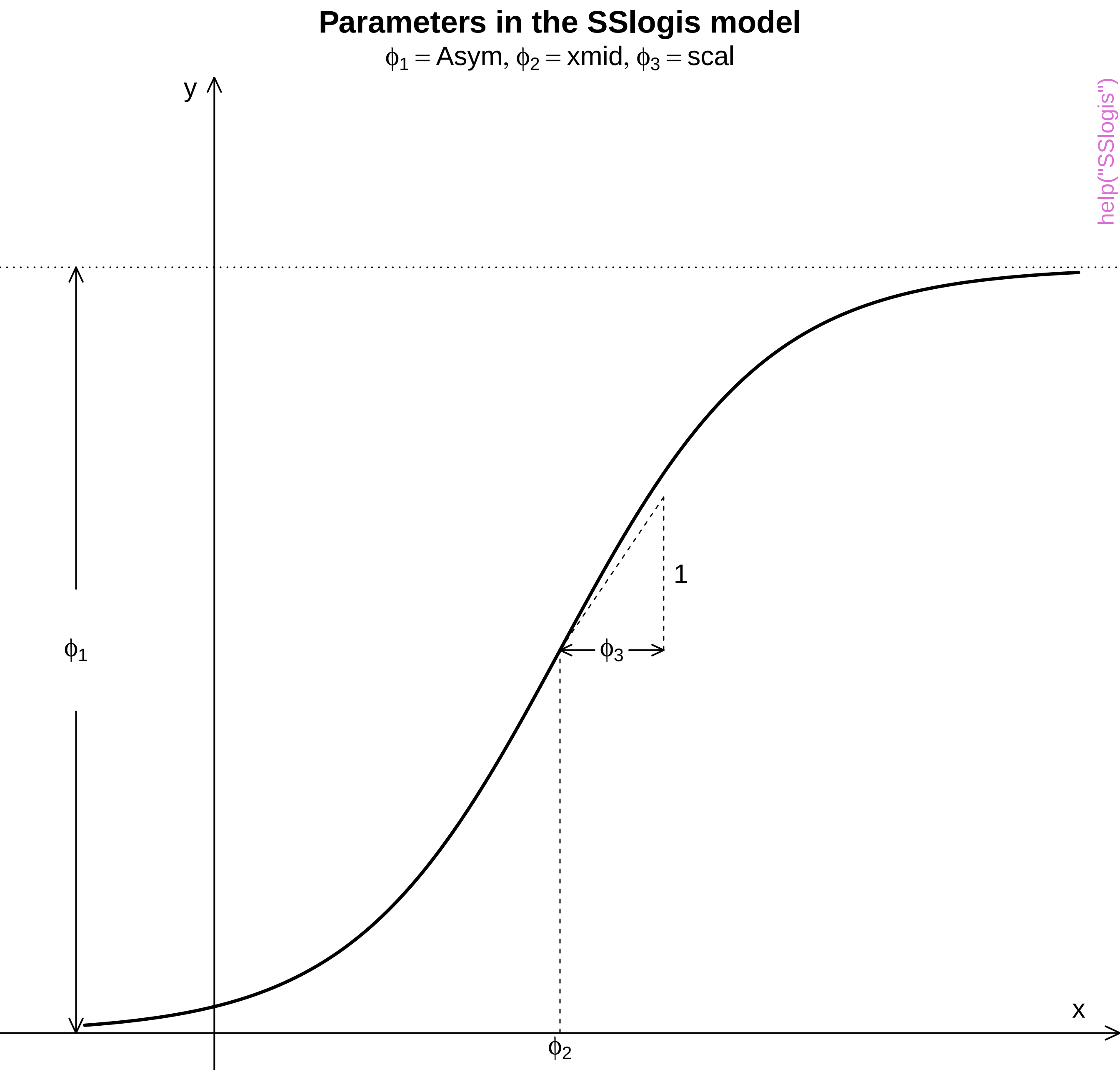
conc
time
0.0
0.5
1.0
1.5
2.0
0
2
4
6
8
help("SSbiexp")



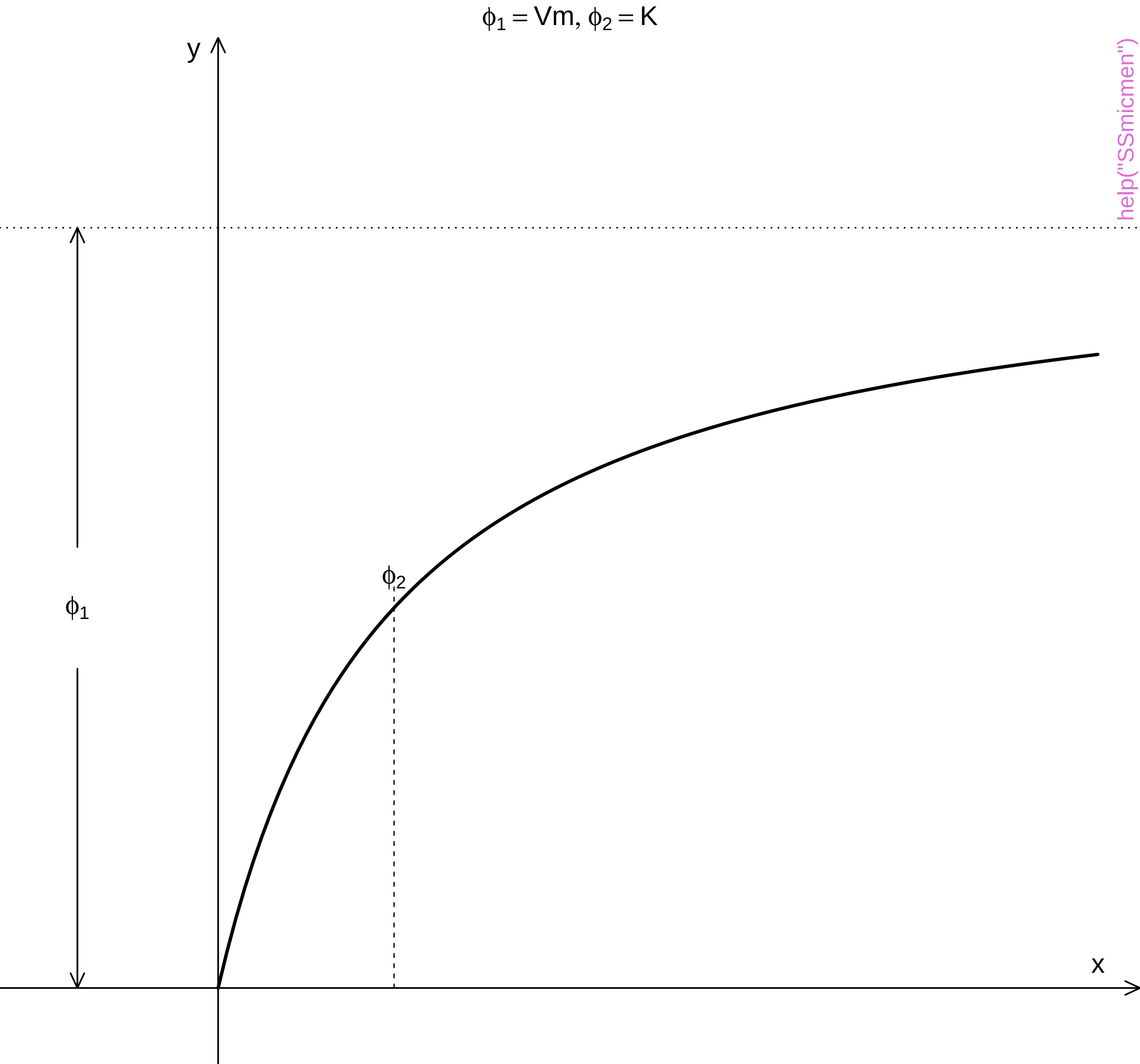
Parameters in the SSfpl model
$\phi_1 = \text{A}, \phi_2 = \text{B}, \phi_3 = \text{xmid}, \phi_4 = \text{scal}$
y
x
ϕ_1
ϕ_2
ϕ_3
ϕ_4
1
help("SSfpl")



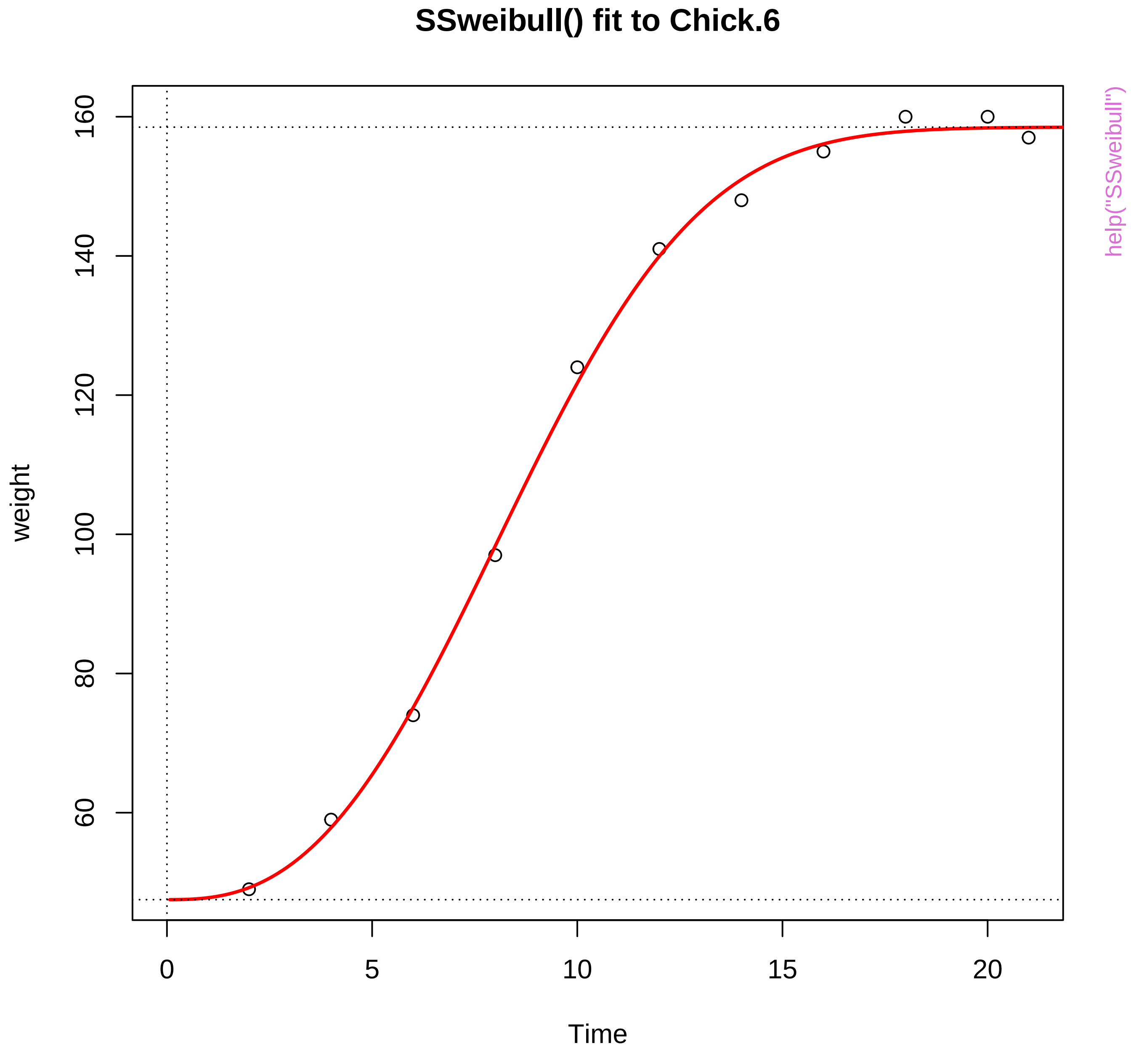
SSgompertz() fit to DNase.1
density
log(conc)
help("SSgompertz")
0.0
0.5
1.0
1.5
−3
−2
−1
0
1
2
e^{-b_2}

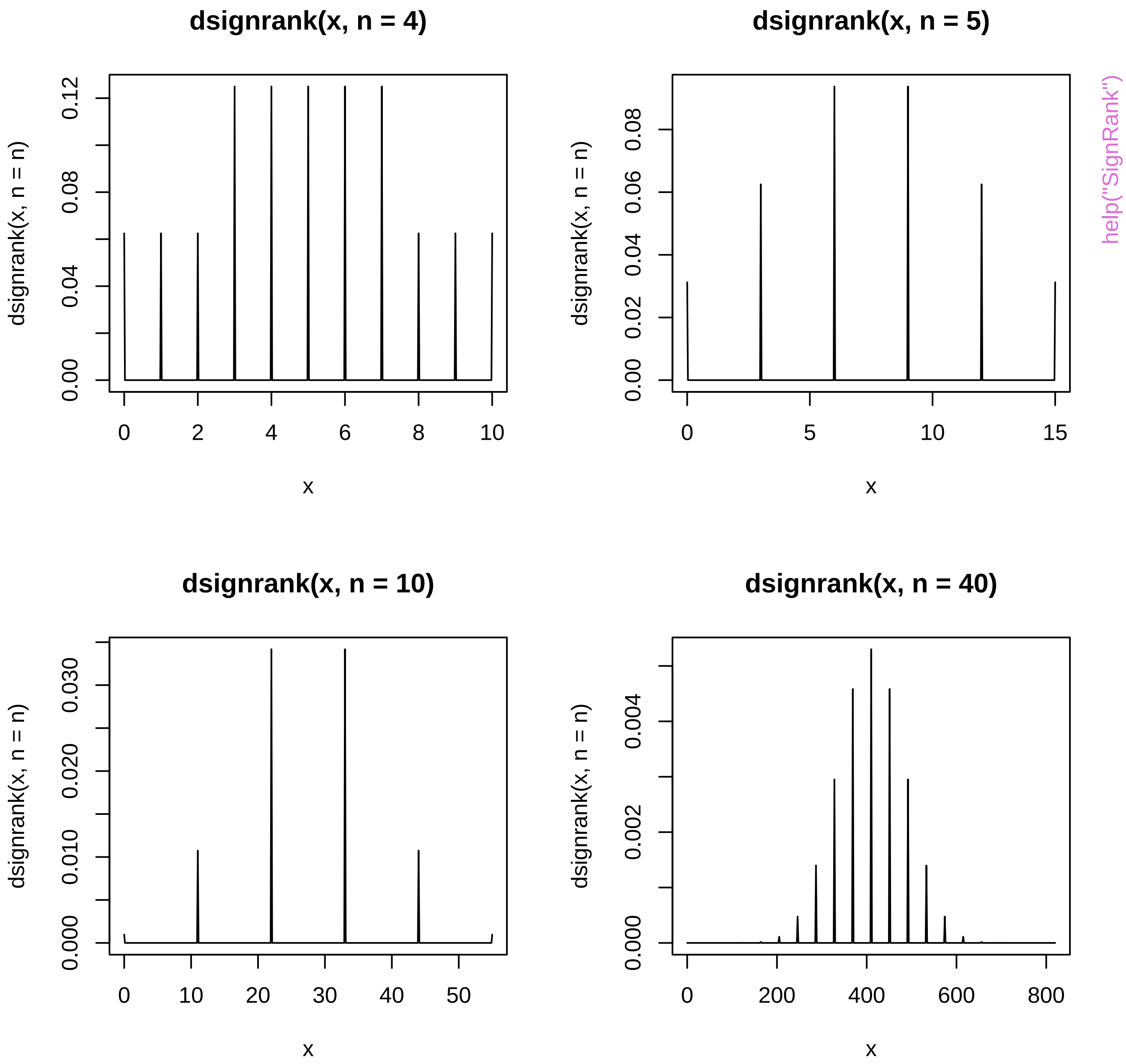


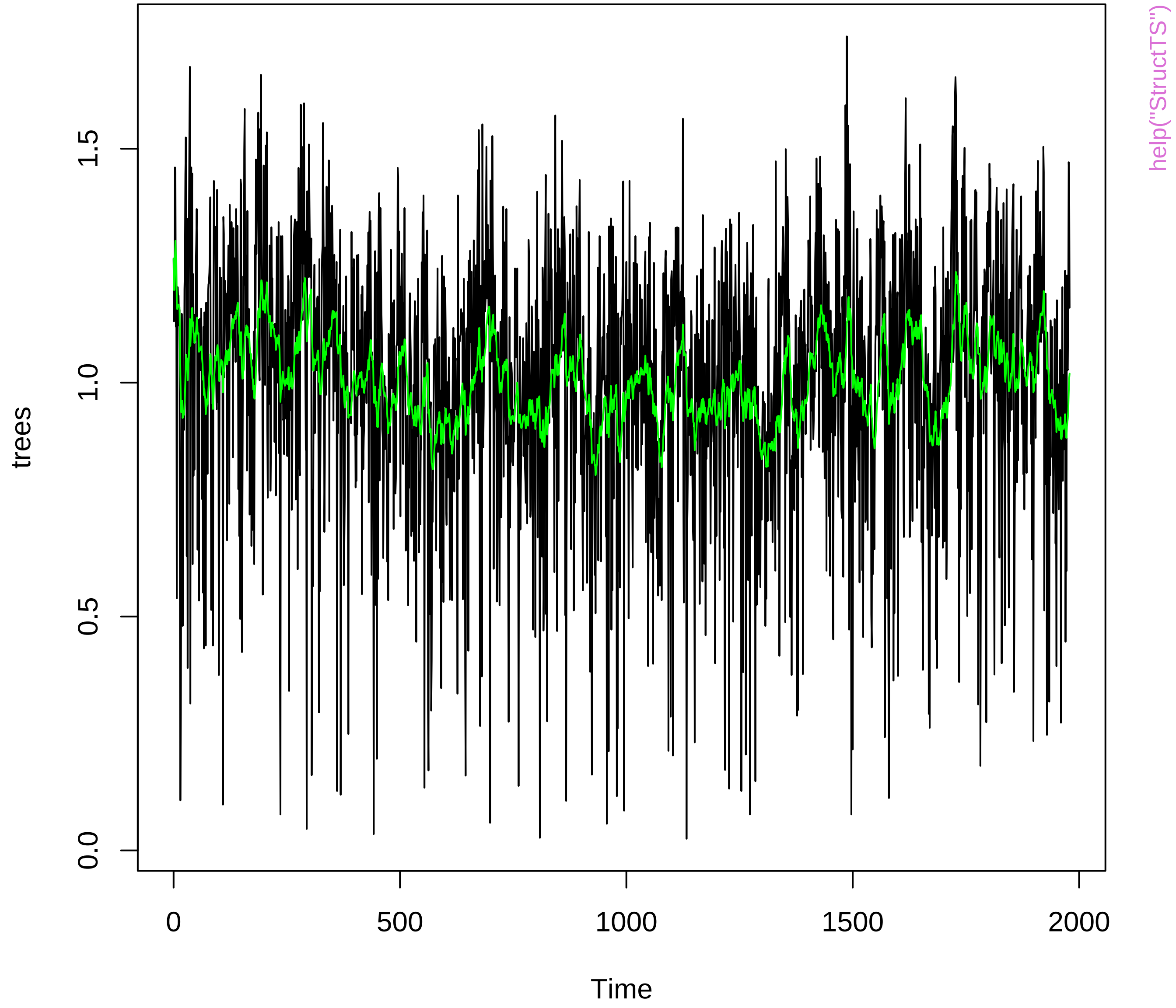
Parameters in the SSlogis model
$\phi_1 = \text{Asym}, \phi_2 = \text{xmid}, \phi_3 = \text{scal}$
y
x
ϕ_1
ϕ_2
ϕ_3
1
help("SSlogis")



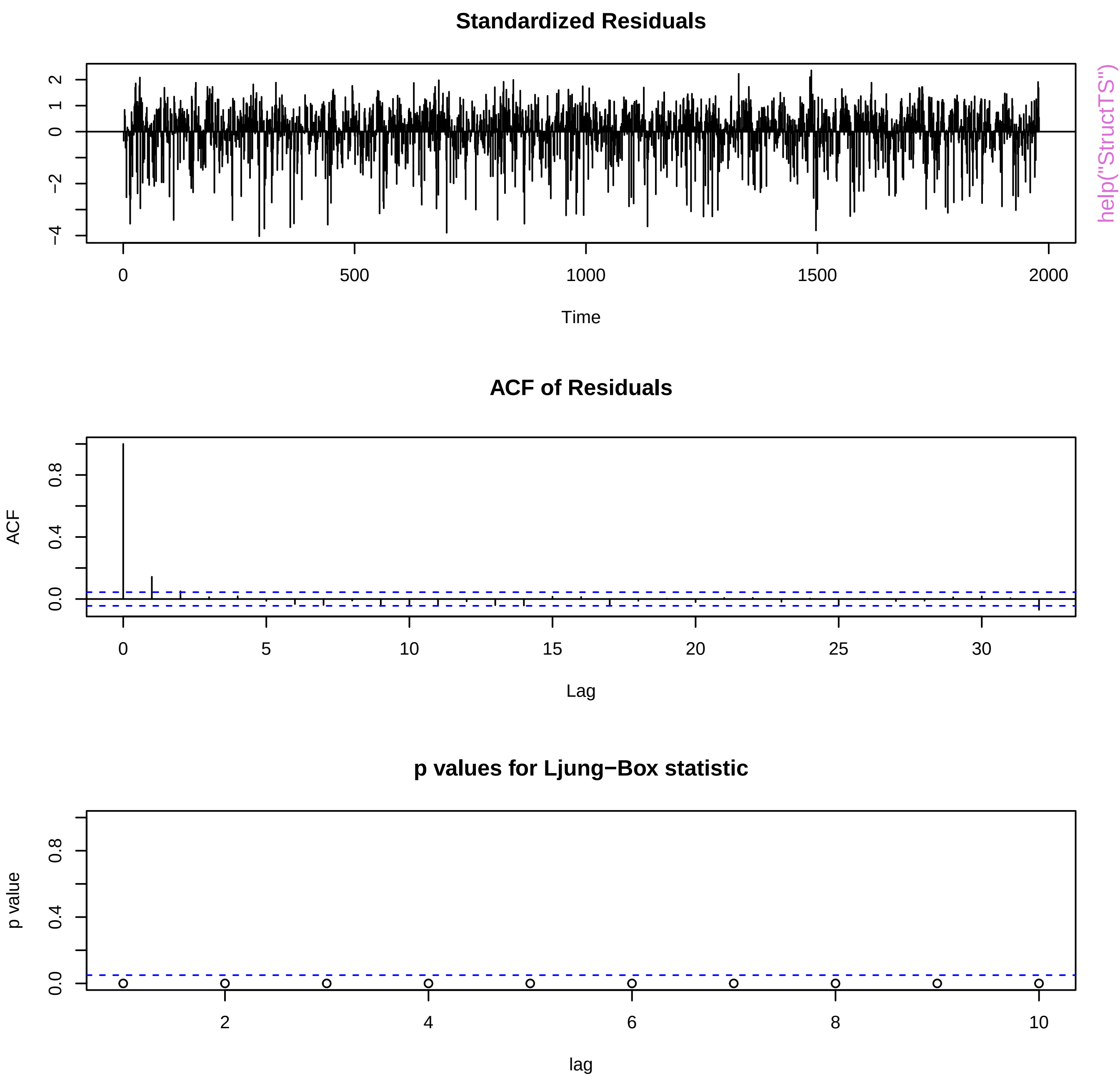
$\phi_1 = \mathrm{Vm}, \phi_2 = \mathrm{K}$
y
x
ϕ_1
ϕ_2
help("SSmicmen")

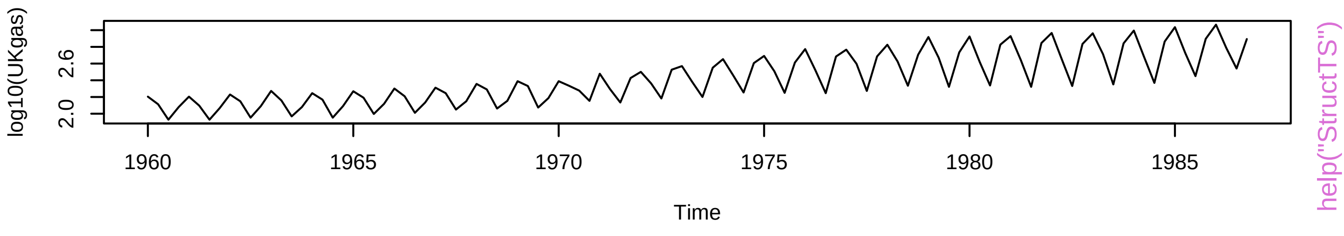

SSweibull() fit to Chick.6
weight
Time
160
140
120
100
80
60
0
5
10
15
20
help("SSweibull")

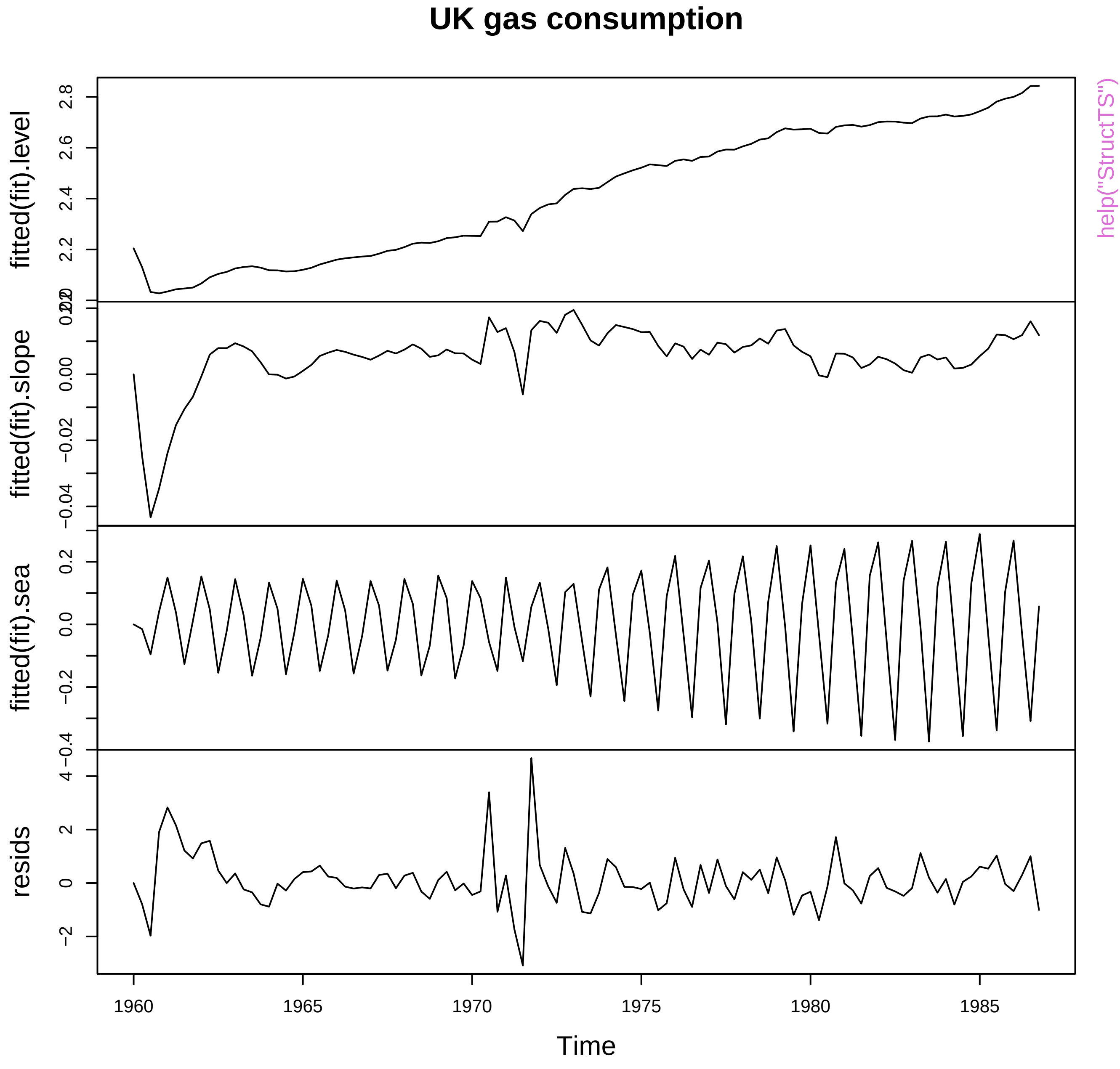

dsignrank(x, n = 4)
dsignrank(x, n = 5)
dsignrank(x, n = 10)
dsignrank(x, n = 40)
dsignrank(x, n = n)
x
0.00
0.04
0.08
0.12
0
2
4
6
8
10
0.00
0.02
0.04
0.06
0.08
0
5
10
15
0.000
0.010
0.020
0.030
0
10
20
30
40
50
0.000
0.002
0.004
0
200
400
600
800
help("SignRank")



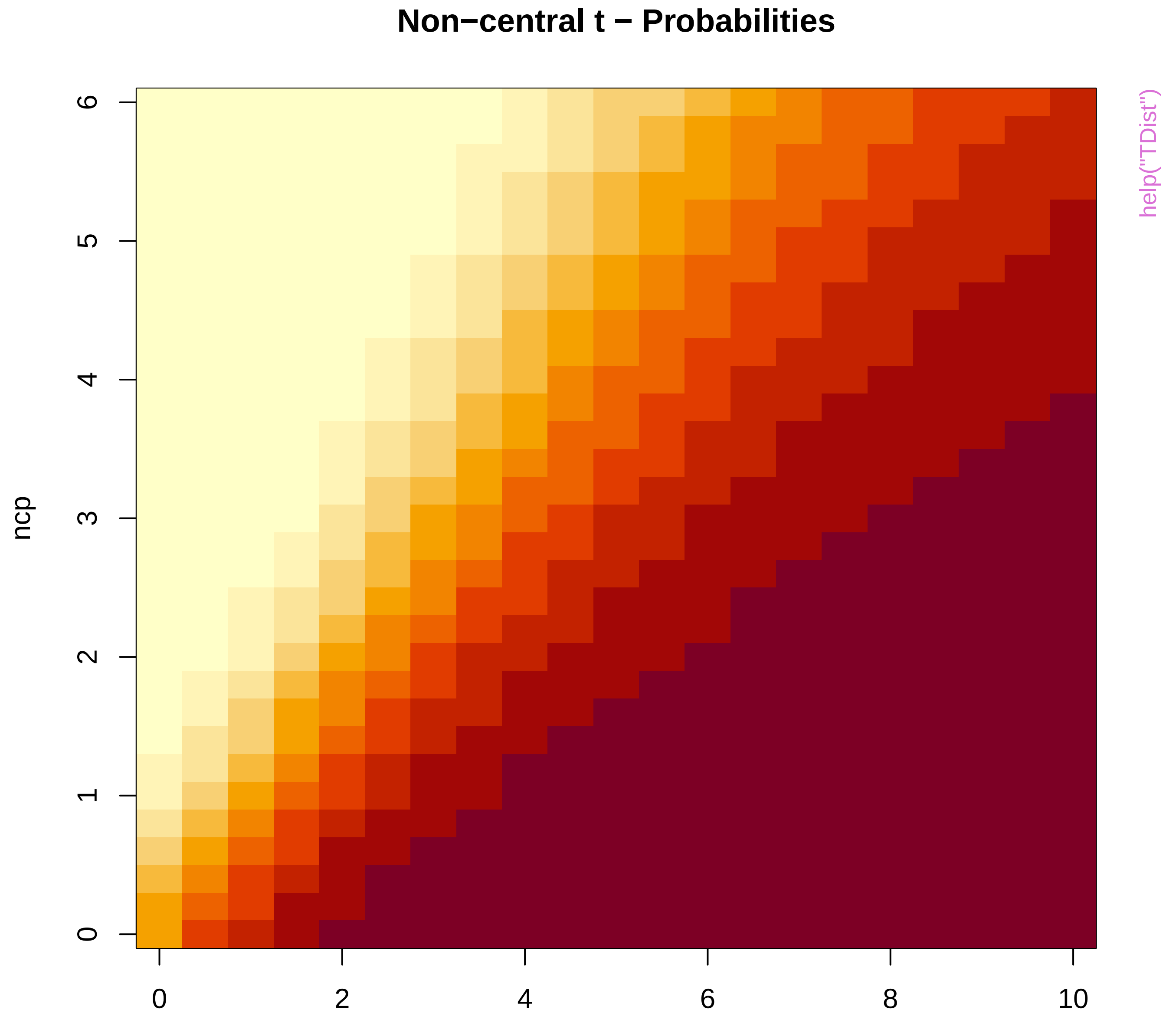
trees
0.0
0.5
1.0
1.5
0
500
1000
1500
2000
Time
help("StructTS")


Standardized Residuals
2
1
0
−2
−4
0
500
1000
1500
2000
Time
help("StructTS")
ACF of Residuals
ACF
0.0
0.4
0.8
0
5
10
15
20
25
30
Lag
p values for Ljung−Box statistic
p value
0.0
0.4
0.8
2
4
6
8
10
lag

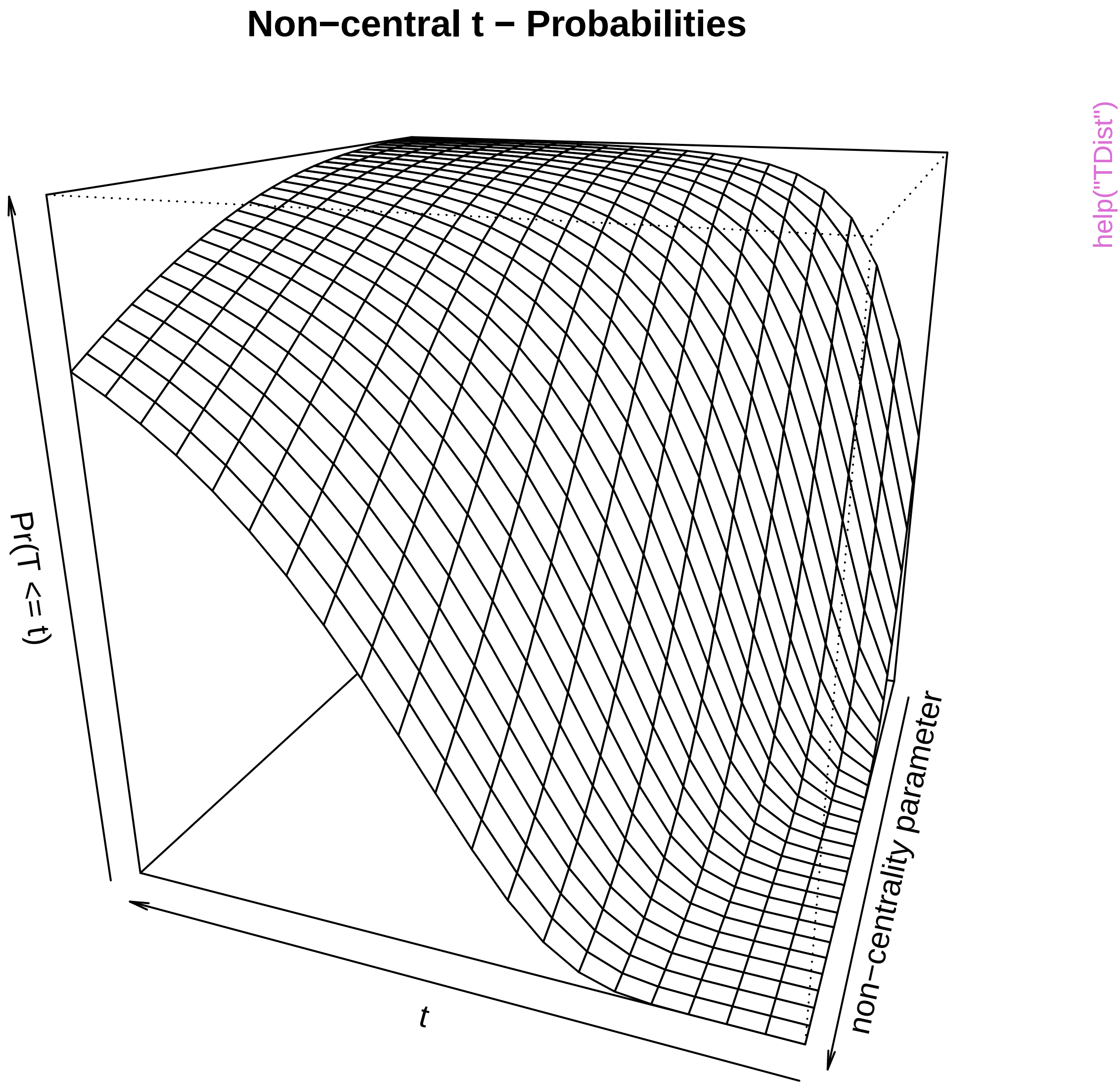

log10(UKgas)
2.0
2.6
1960
1965
1970
1975
1980
1985
Time
help("StructTS")

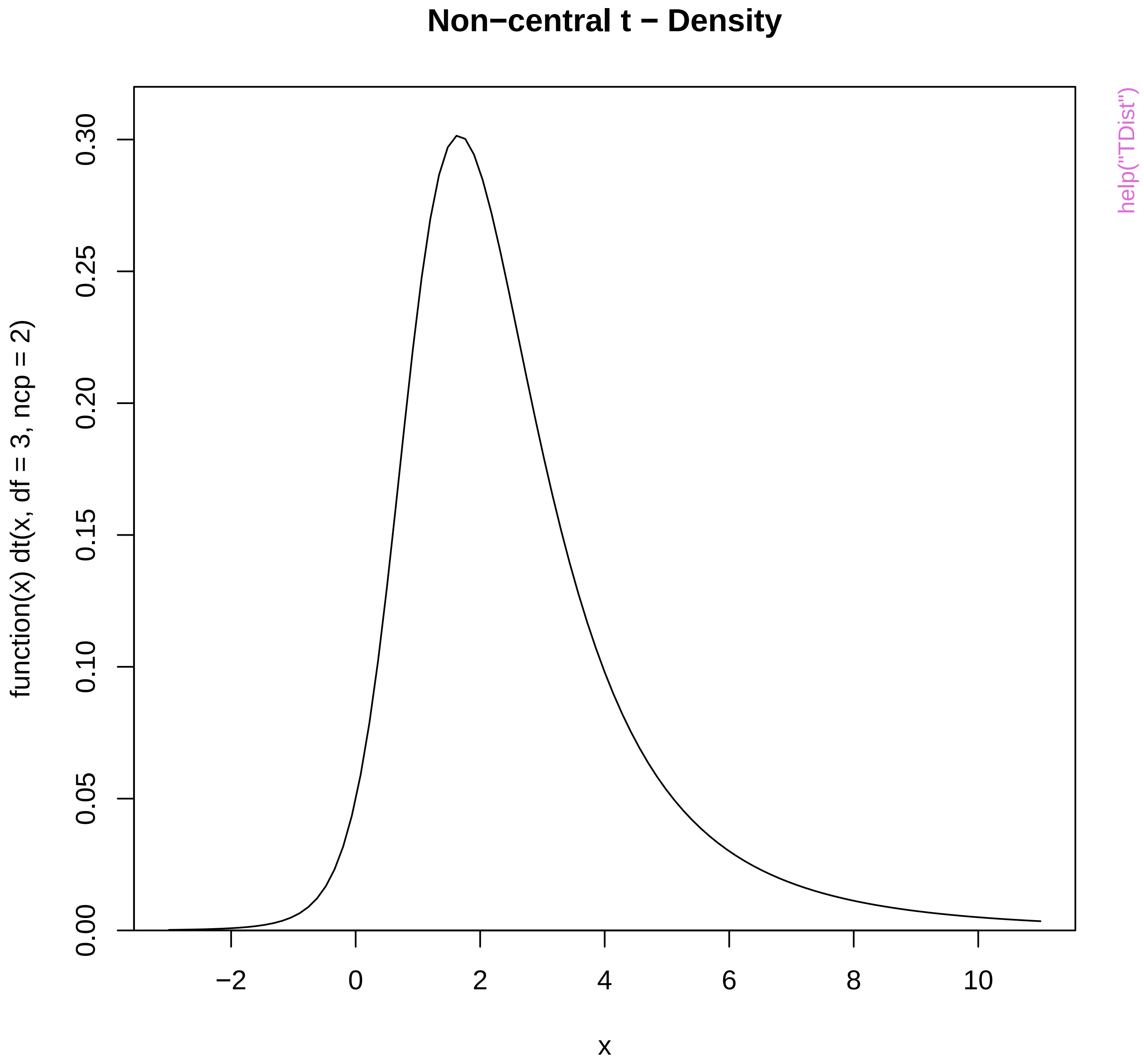


UK gas consumption
fitted(fit).level
fitted(fit).slope
fitted(fit).sea
resids
Time
1960
1965
1970
1975
1980
1985
help("StructTS")

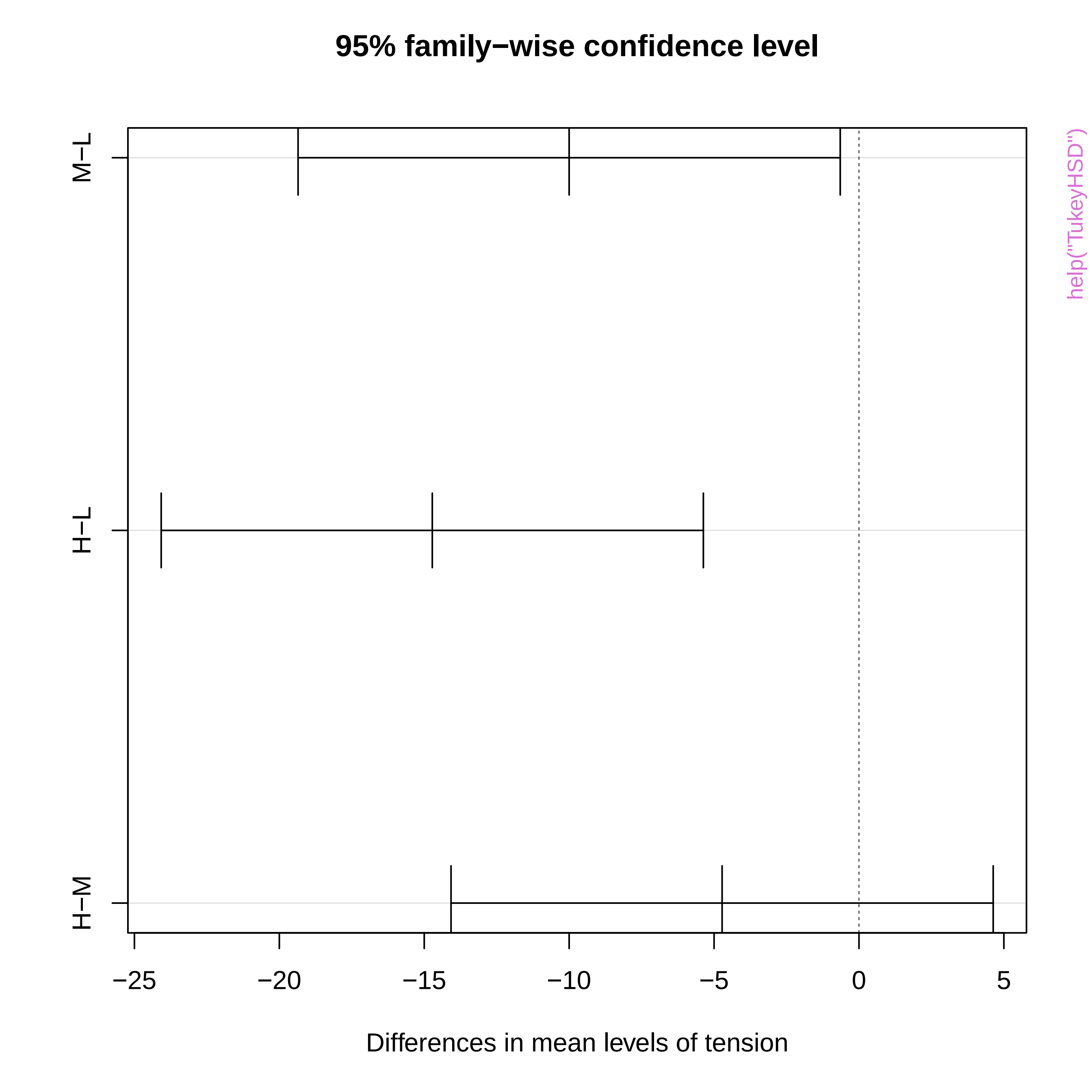


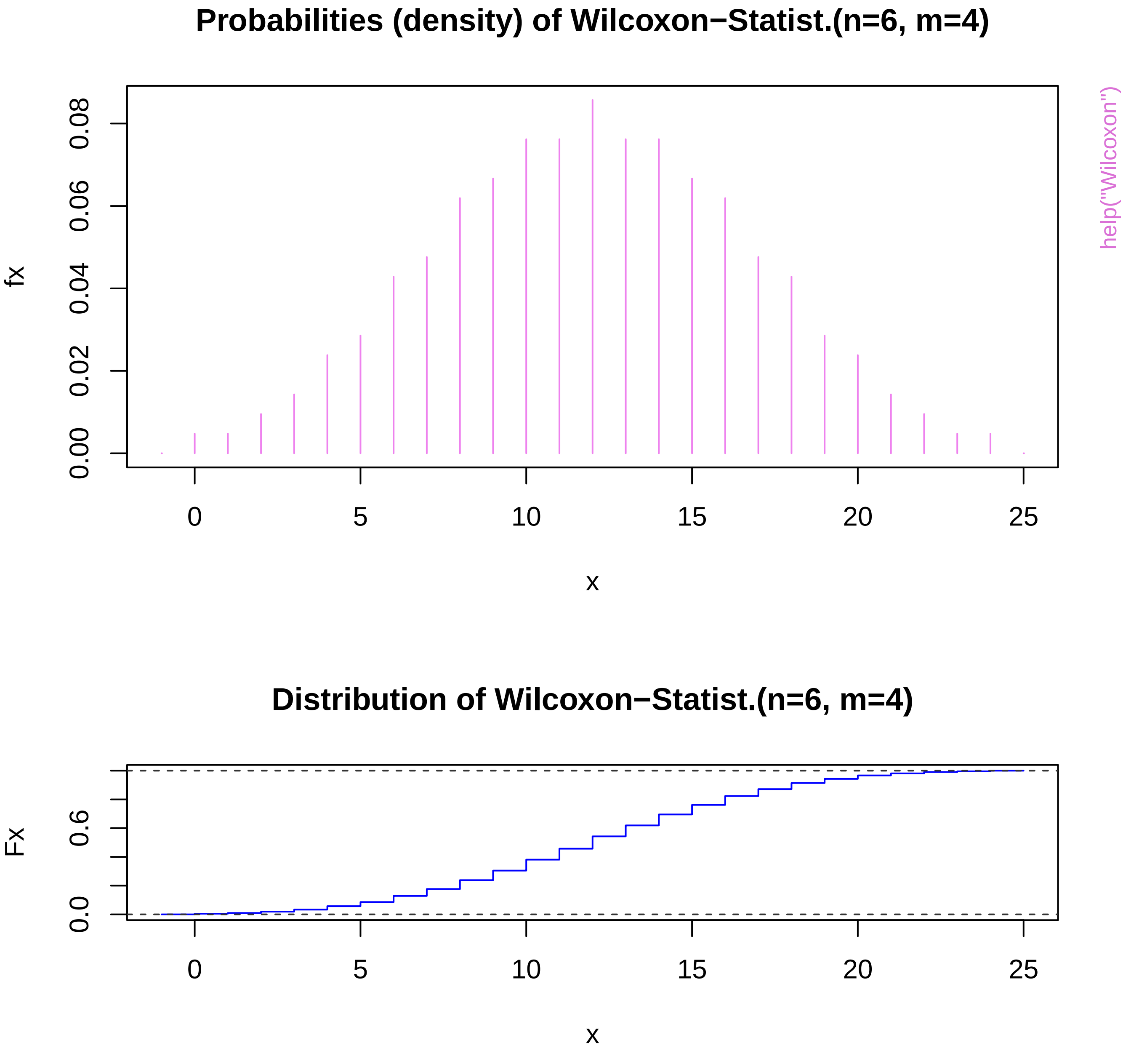
Non−central t − Probabilities
ncp
0
1
2
3
4
5
6
0
2
4
6
8
10
tt
help("TDist")

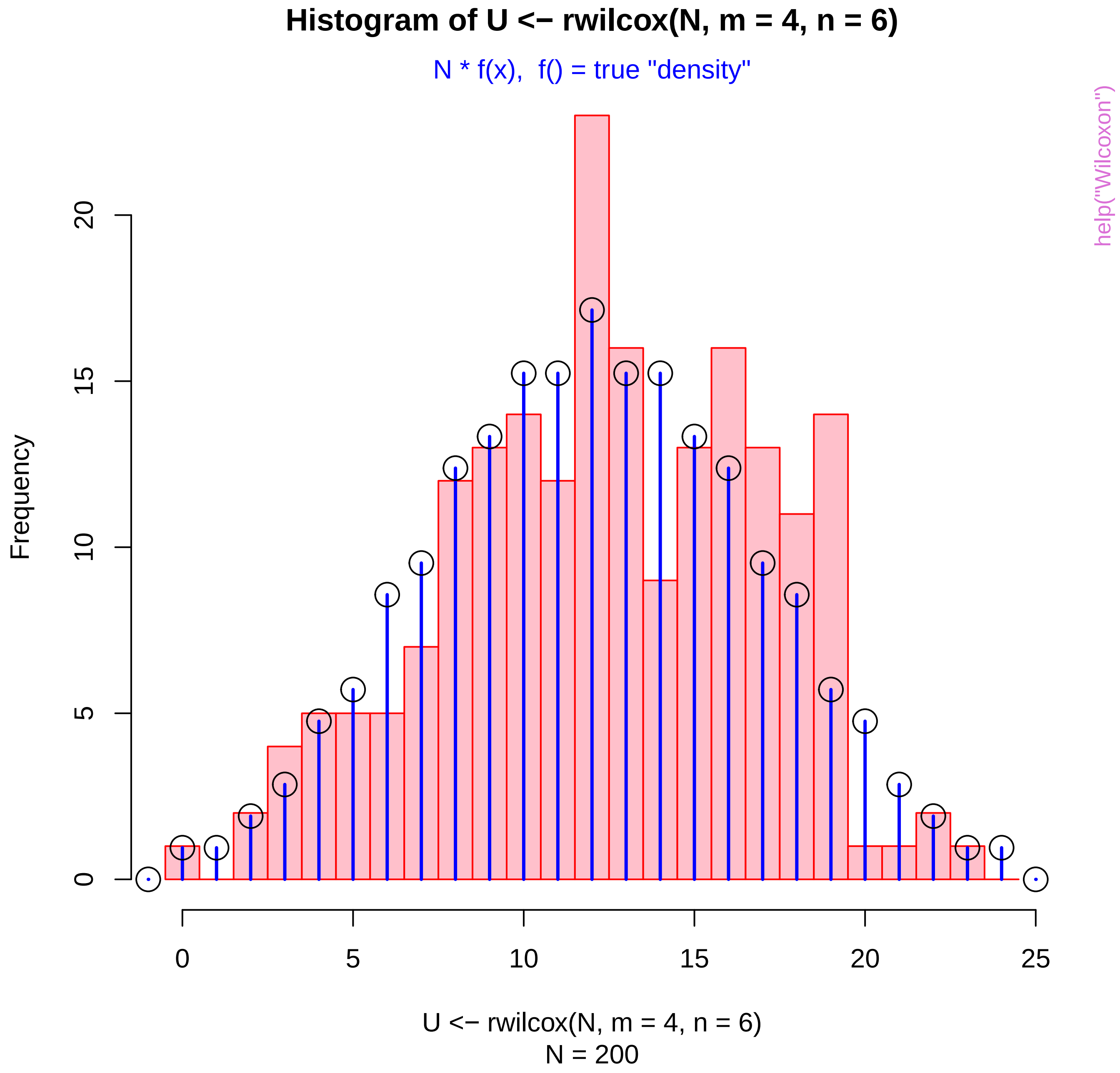

Non−central t − Probabilities
Pr(T <= t)
t
non−centrality parameter
help("TDist")



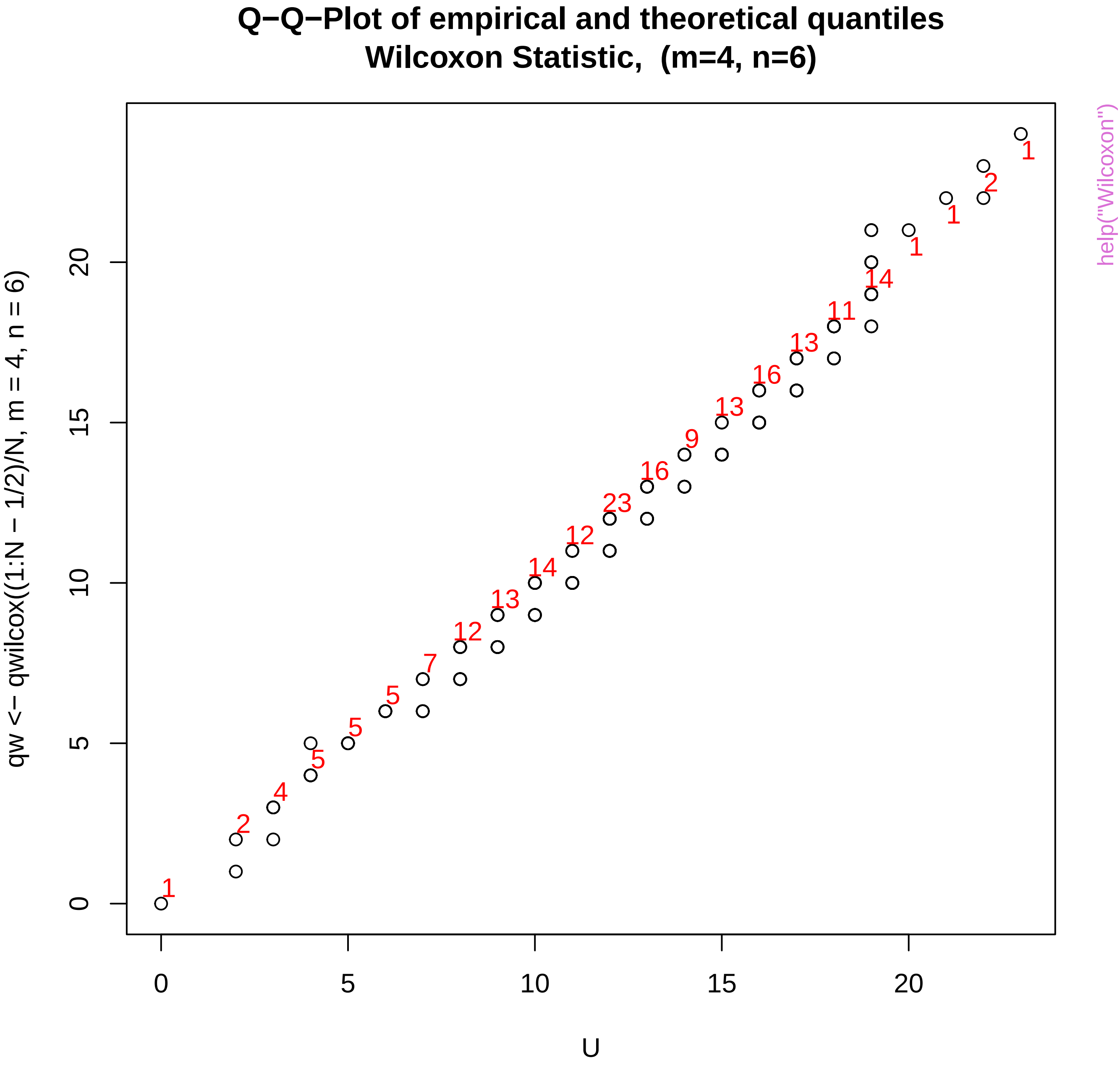
Non−central t − Density
function(x) dt(x, df = 3, ncp = 2)
0.00
0.05
0.10
0.15
0.20
0.25
0.30
−2
0
2
4
6
8
10
x
help("TDist")

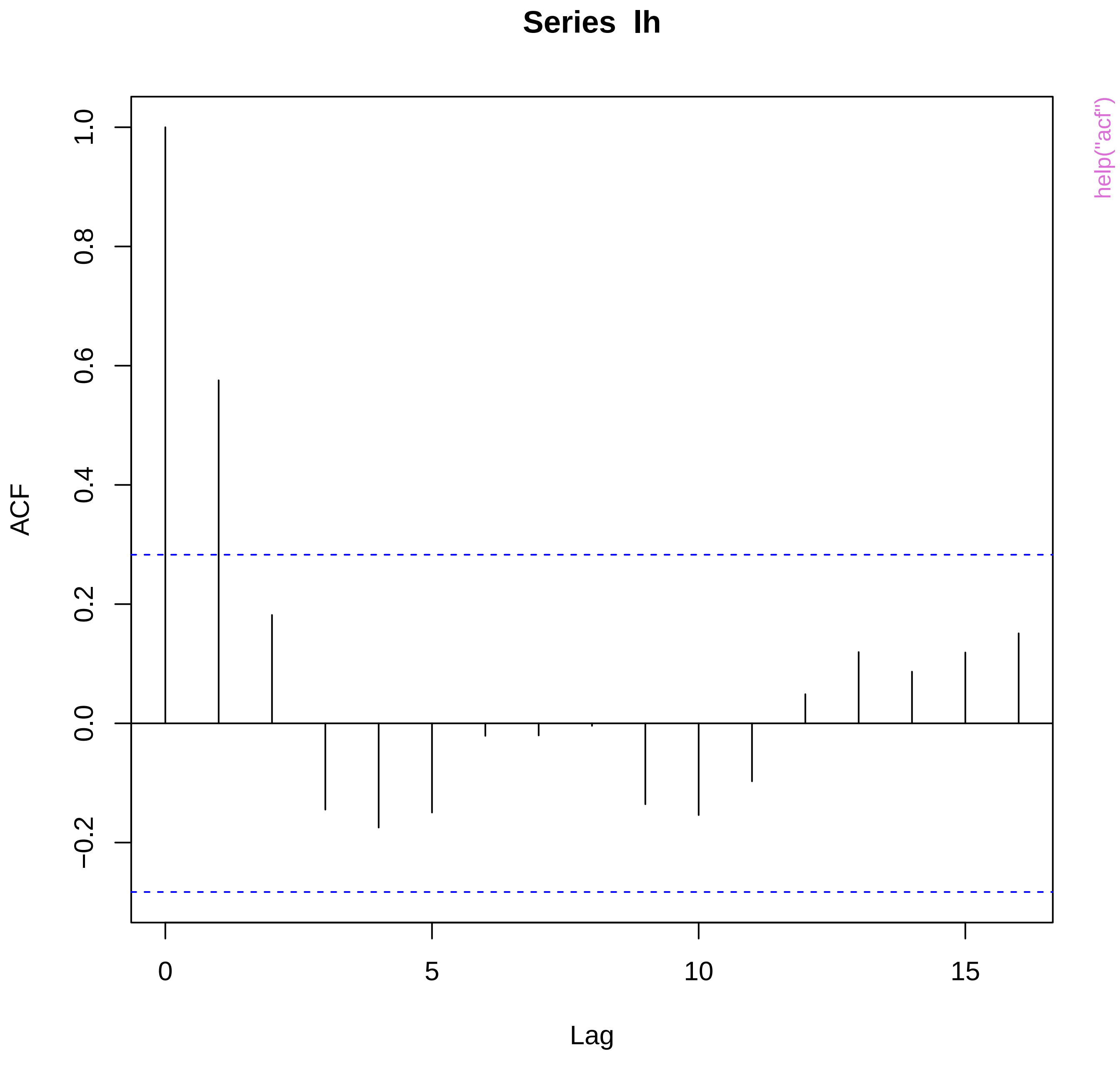

95% family−wise confidence level
M−L
H−L
H−M
−25
−20
−15
−10
−5
0
5
Differences in mean levels of tension
help("TukeyHSD")


Probabilities (density) of Wilcoxon−Statist.(n=6, m=4)
fx
0.00
0.02
0.04
0.06
0.08
0
5
10
15
20
25
x
help("Wilcoxon")
Distribution of Wilcoxon−Statist.(n=6, m=4)
Fx
0.0
0.6
0
5
10
15
20
25
x

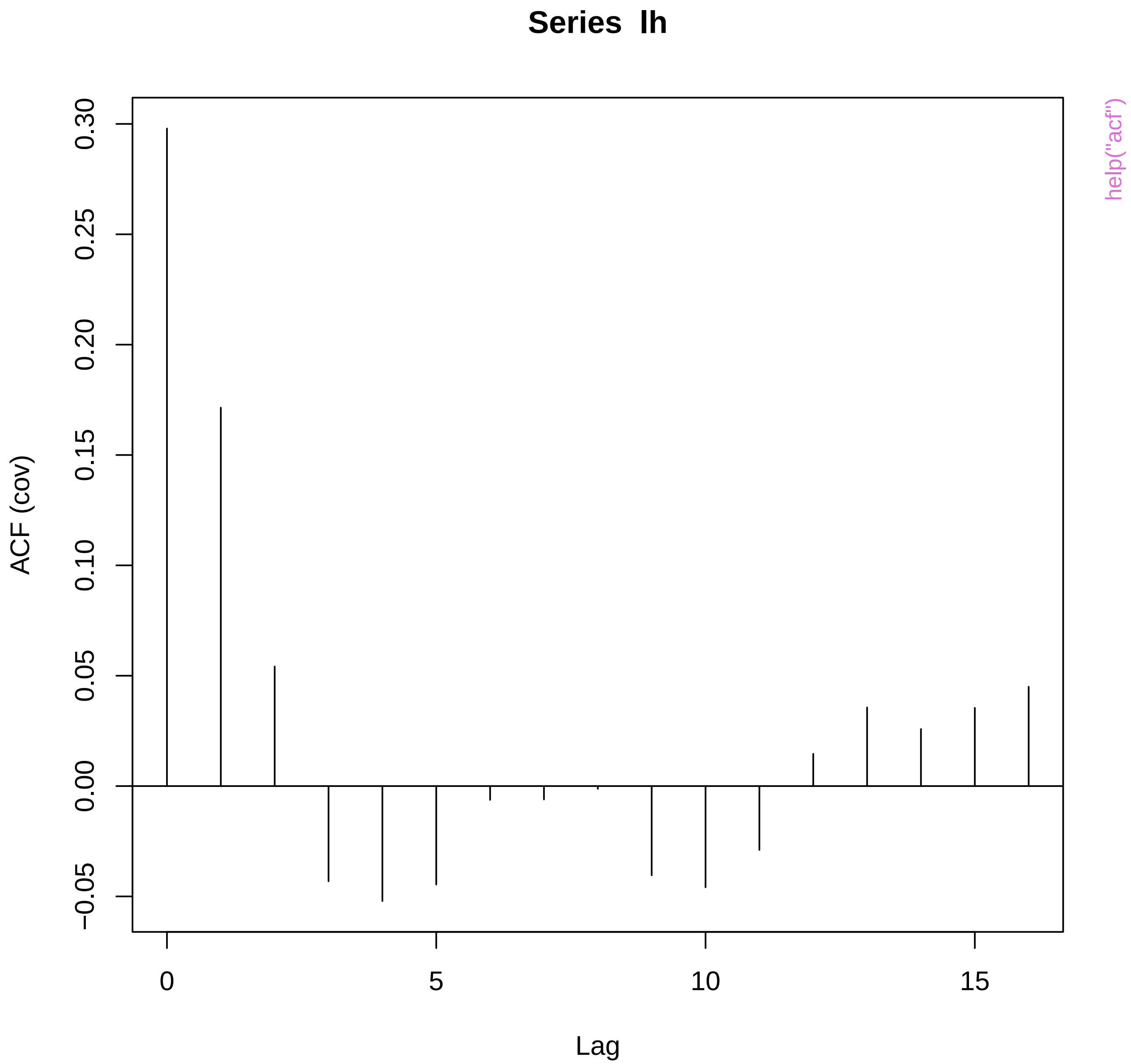


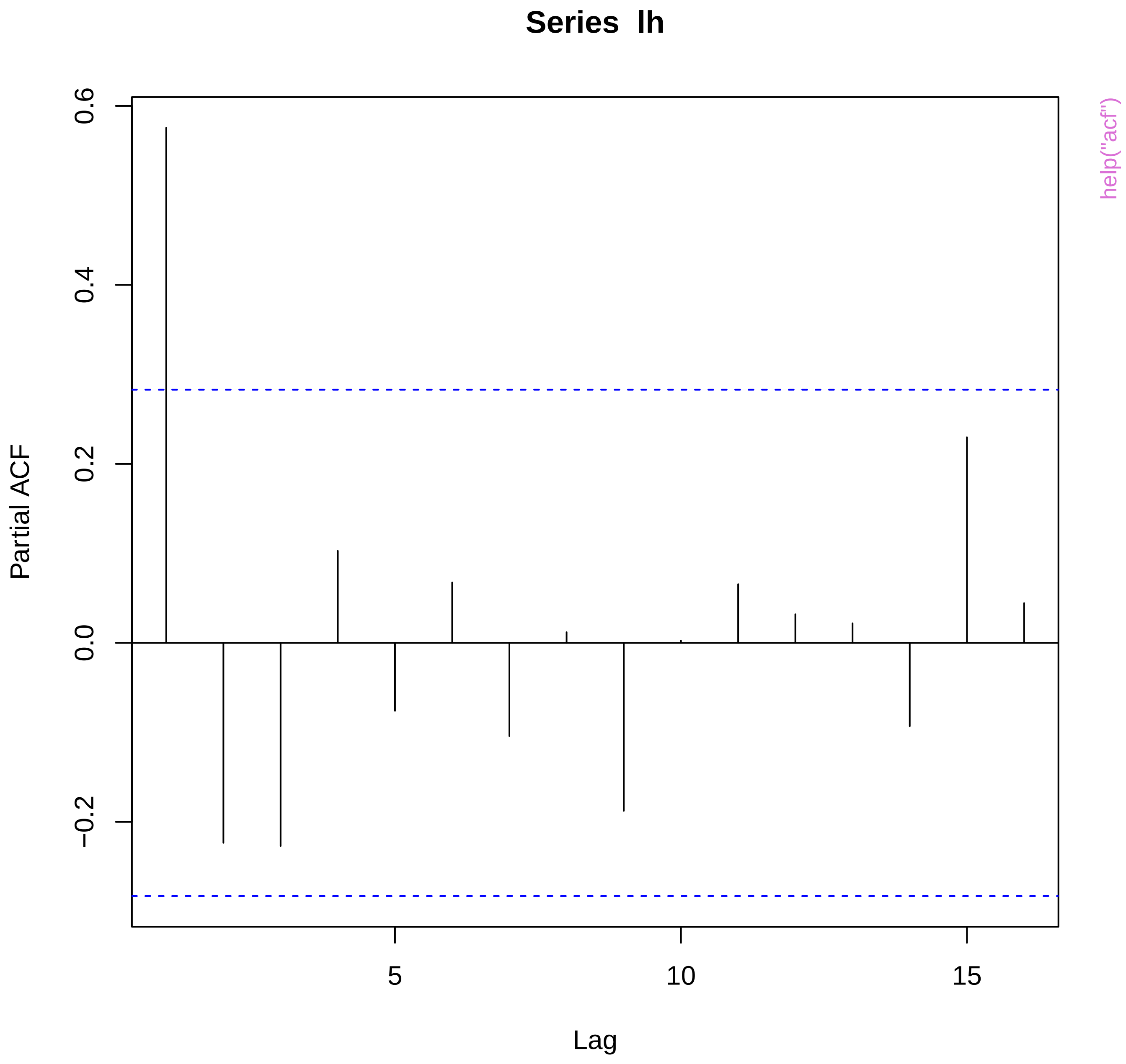
Histogram of U <− rwilcox(N, m = 4, n = 6)
N * f(x), f() = true "density"
help("Wilcoxon")
Frequency
0
5
10
15
20
0
5
10
15
20
25
U <− rwilcox(N, m = 4, n = 6)
N = 200


Q−Q−Plot of empirical and theoretical quantiles
Wilcoxon Statistic, (m=4, n=6)
qw <− qwilcox((1:N − 1/2)/N, m = 4, n = 6)
U
help("Wilcoxon")
0
5
10
15
20
1
2
4
5
5
5
7
12
13
14
12
23
16
9
13
16
13
11
14
1
1
2
1

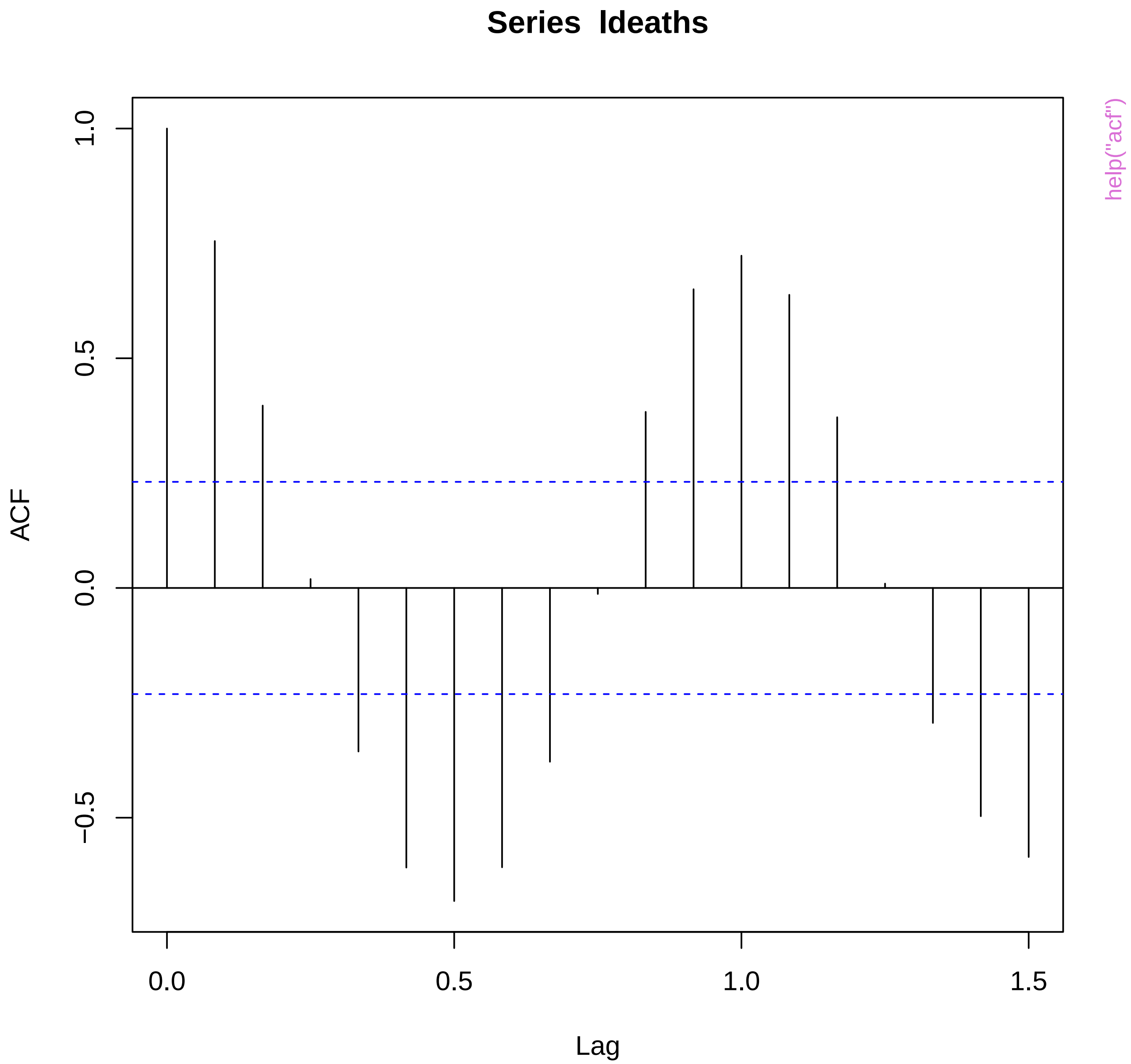


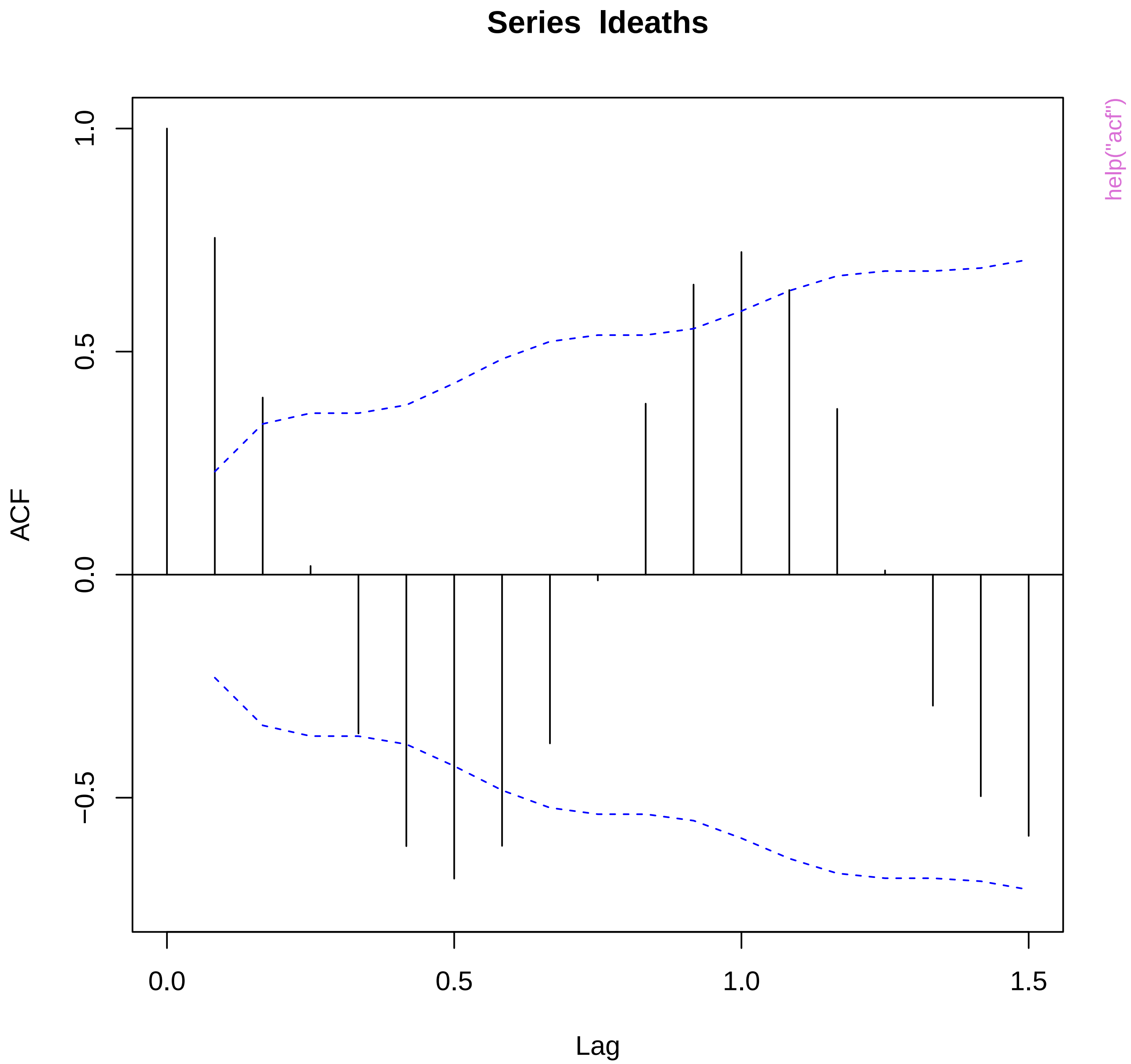
Series lh
ACF
Lag
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
5
10
15
help("acf")


Series lh
ACF (cov)
Lag
0
5
10
15
−0.05
0.00
0.05
0.10
0.15
0.20
0.25
0.30
help("acf")

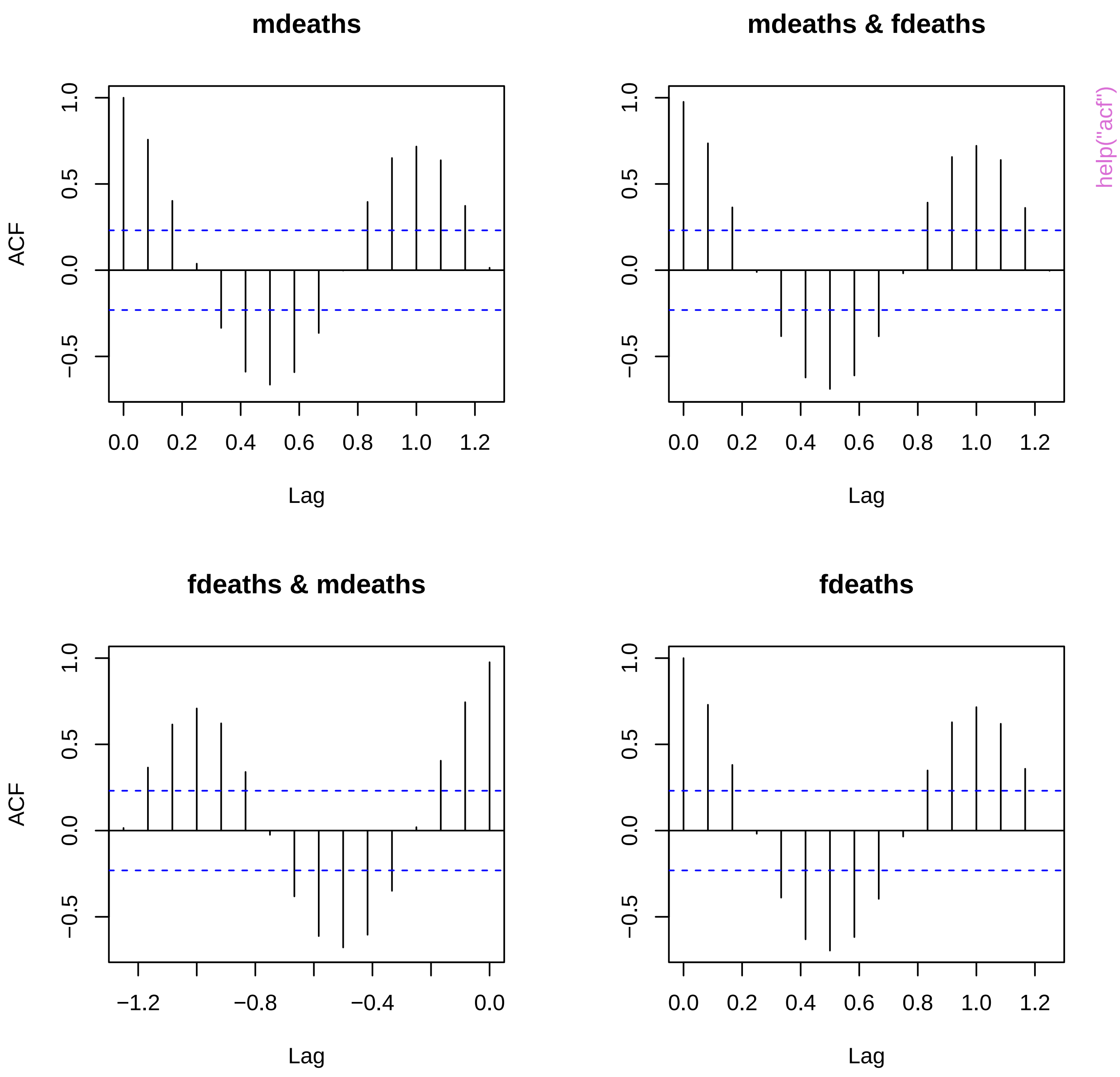


Series lh
Partial ACF
Lag
0.6
0.4
0.2
0.0
−0.2
5
10
15
help("acf")

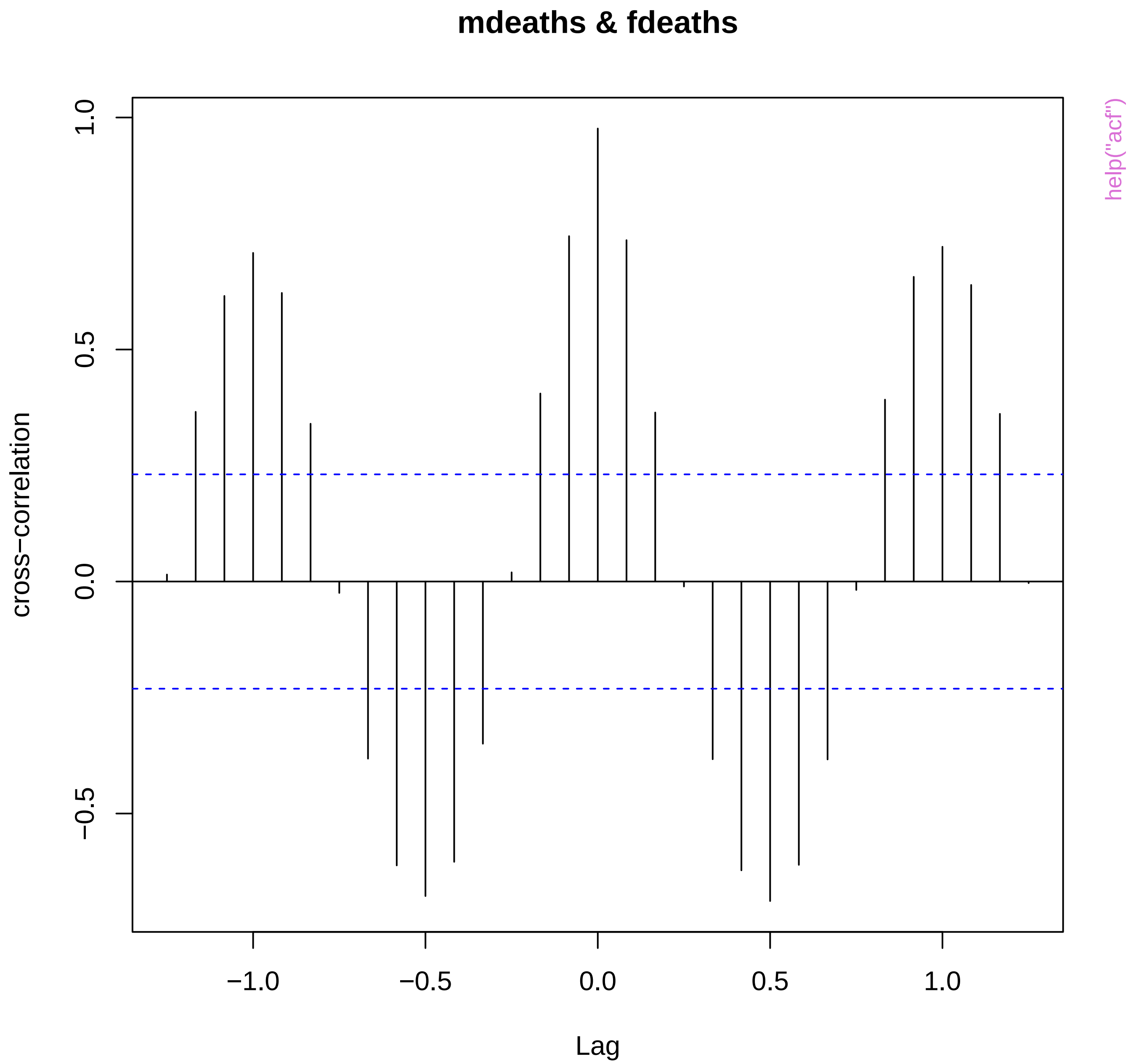

Series ldeaths
ACF
Lag
1.0
0.5
0.0
−0.5
0.0
0.5
1.0
1.5
help("acf")

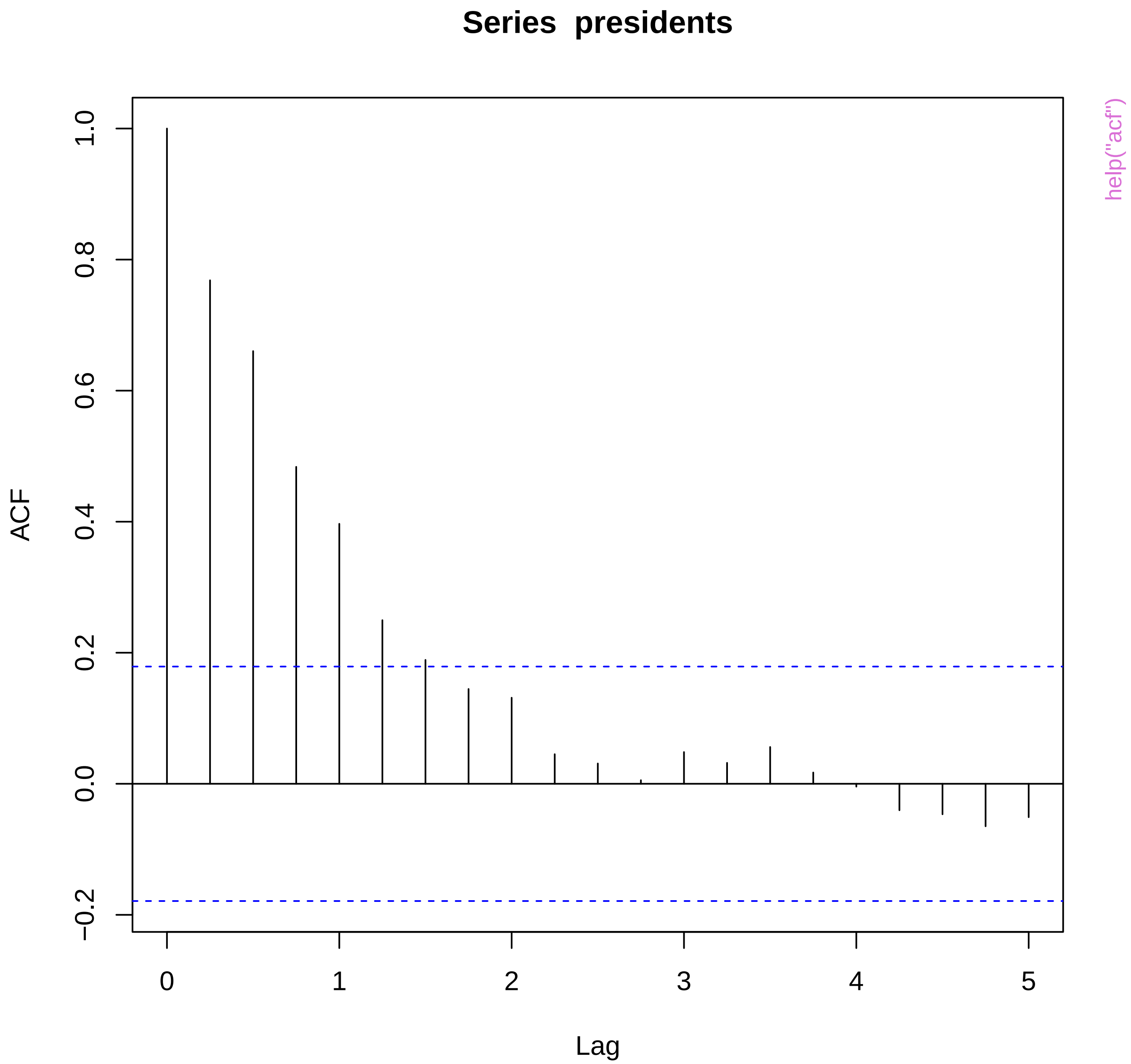


Series ldeaths
ACF
1.0
0.5
0.0
−0.5
0.0
0.5
1.0
1.5
Lag
help("acf")

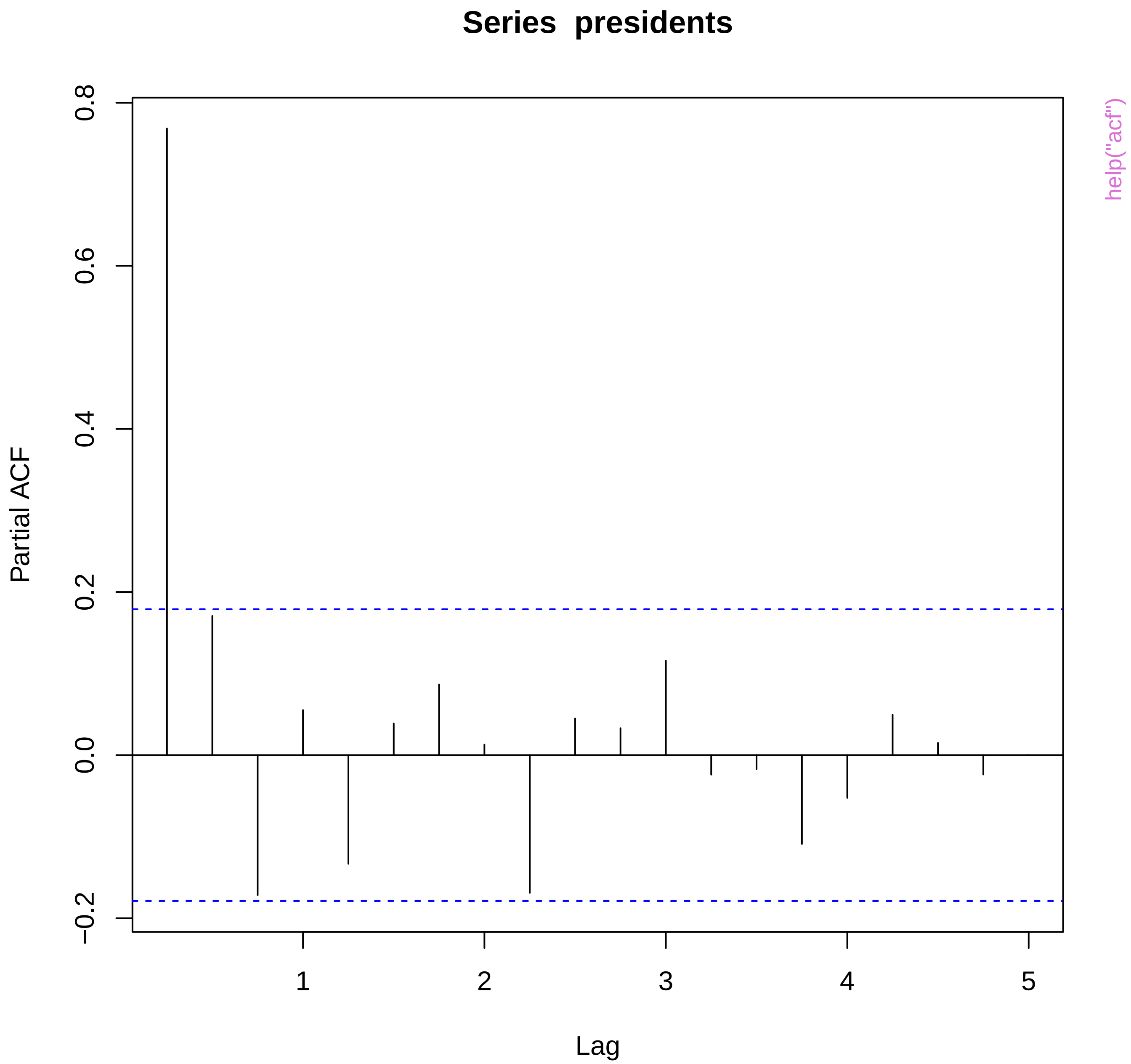


help("acf")
mdeaths
mdeaths & fdeaths
fdeaths & mdeaths
fdeaths
ACF
1.0
0.5
0.0
−0.5
0.0
0.2
0.4
0.6
0.8
1.0
1.2
−1.2
−0.8
−0.4
0.0
Lag

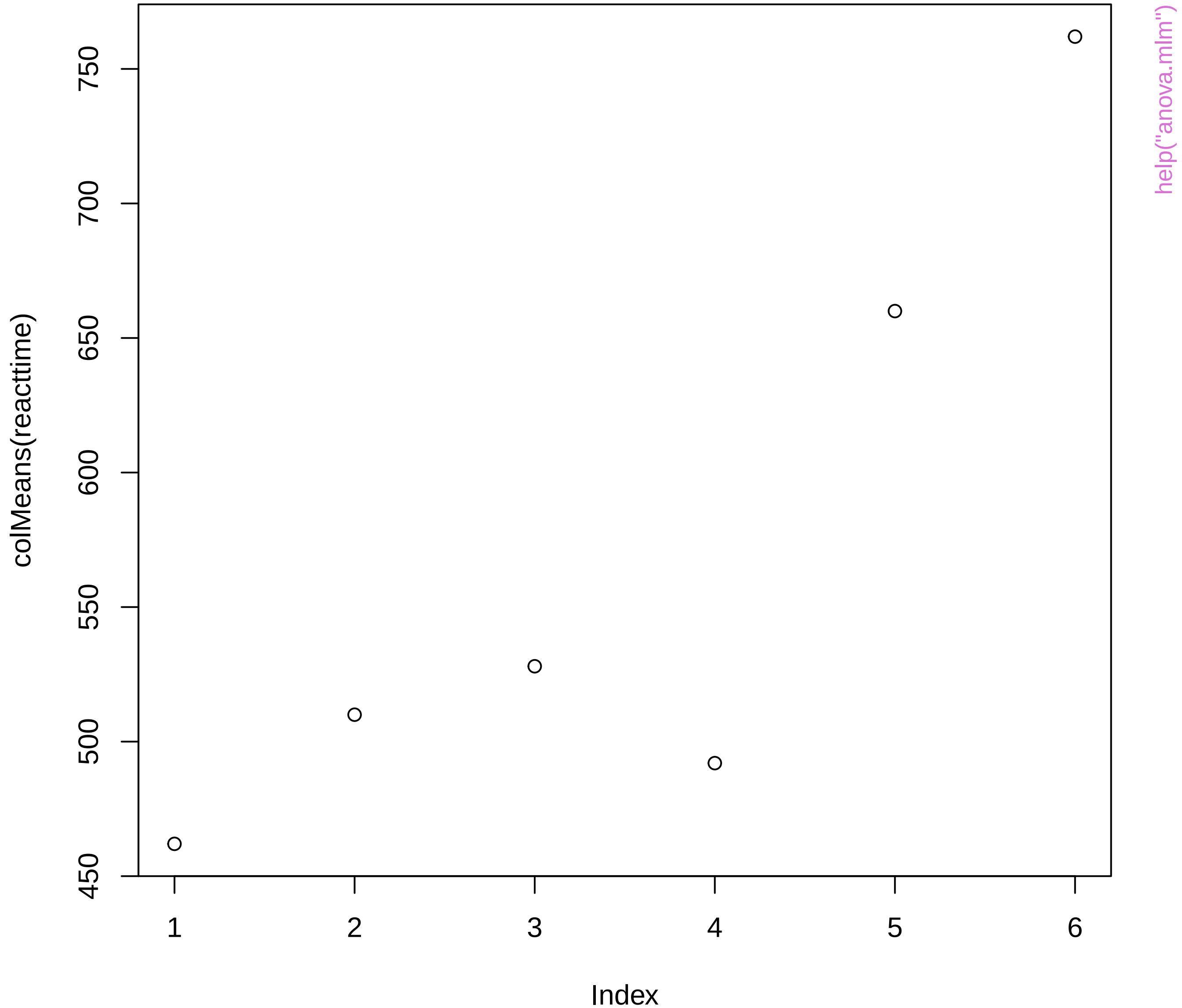

mdeaths & fdeaths
1.0
0.5
0.0
−0.5
cross−correlation
−1.0
−0.5
0.0
0.5
1.0
Lag
help("acf")

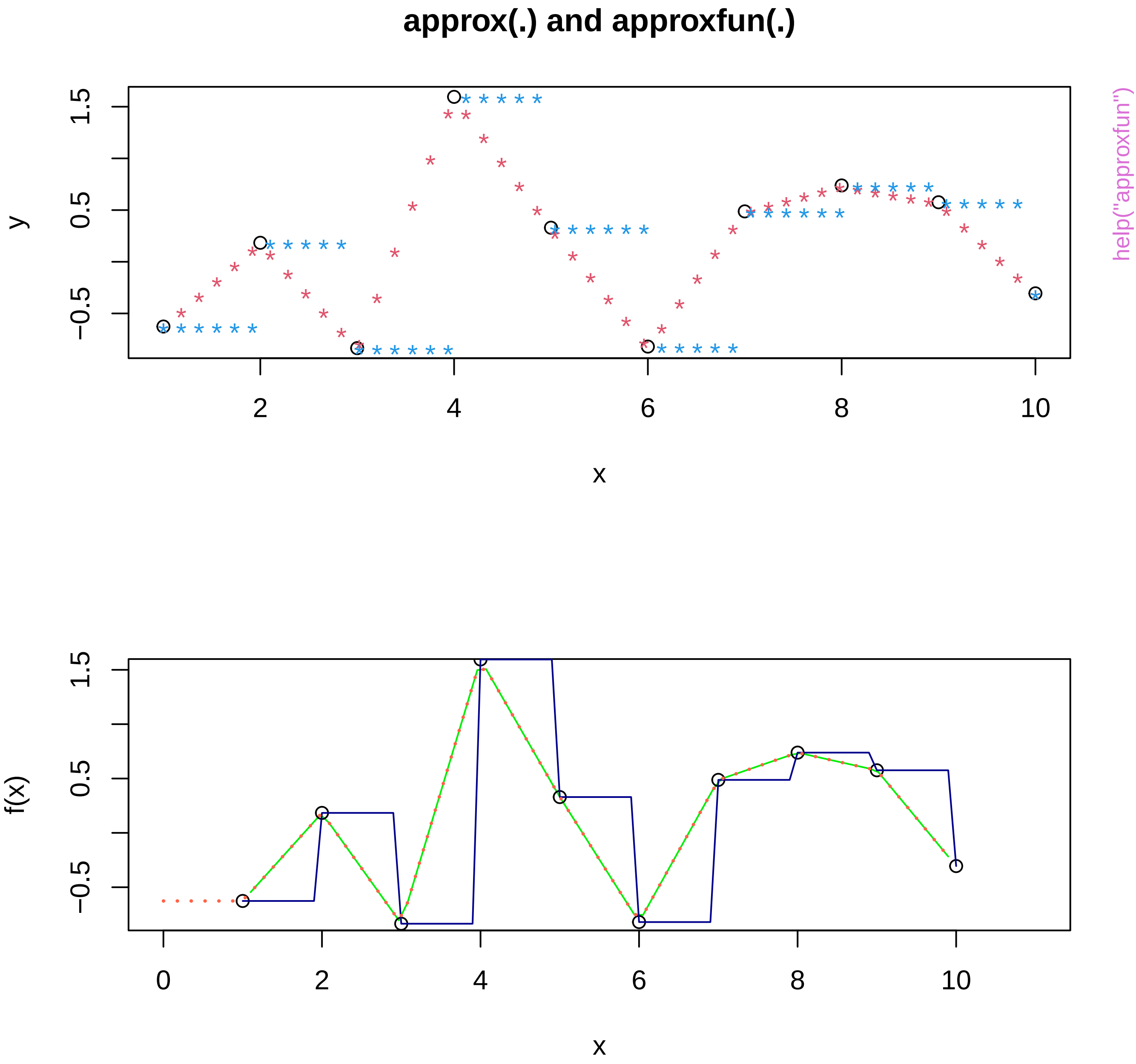


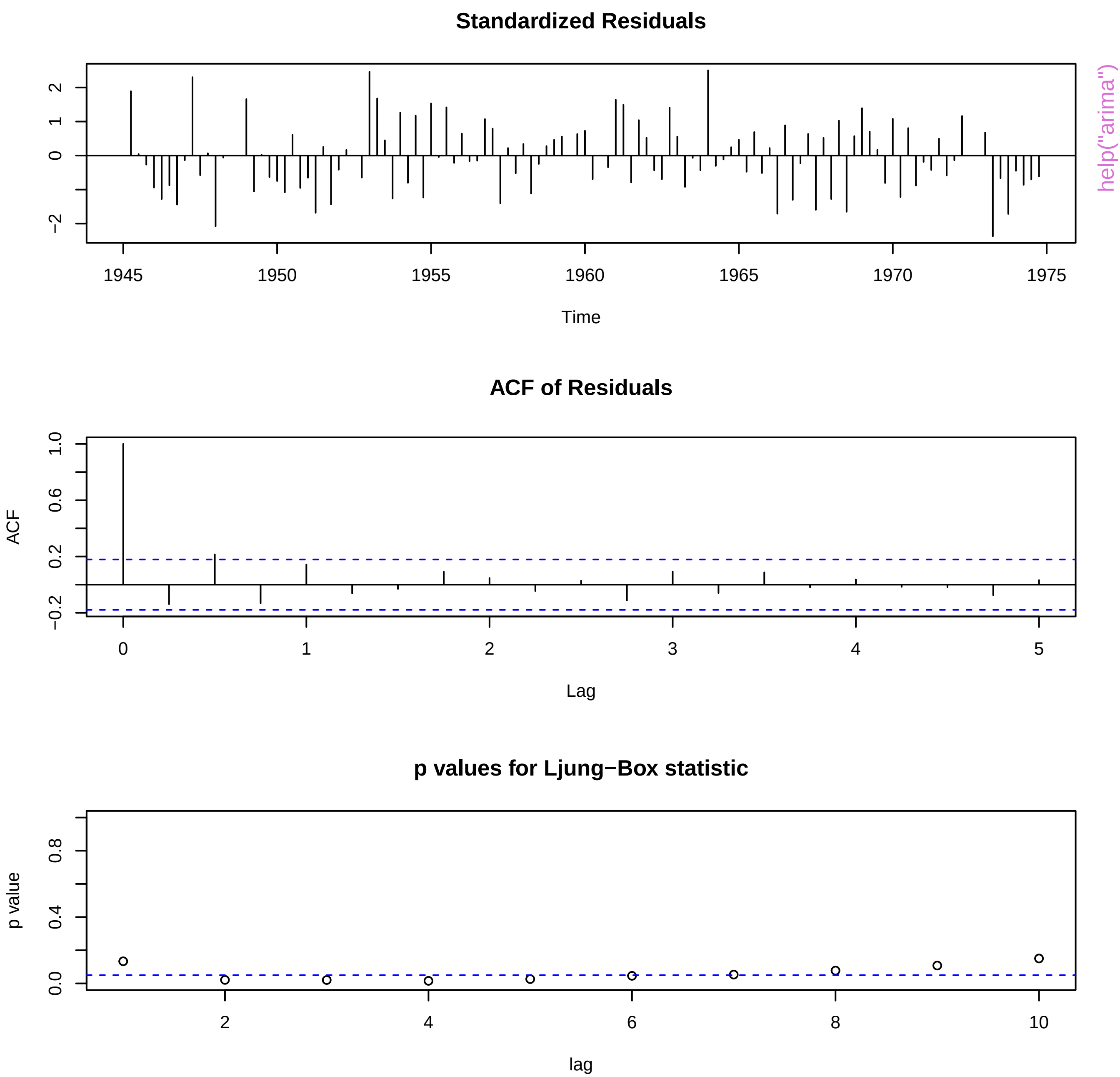
Series presidents
ACF
Lag
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
1
2
3
4
5
help("acf")

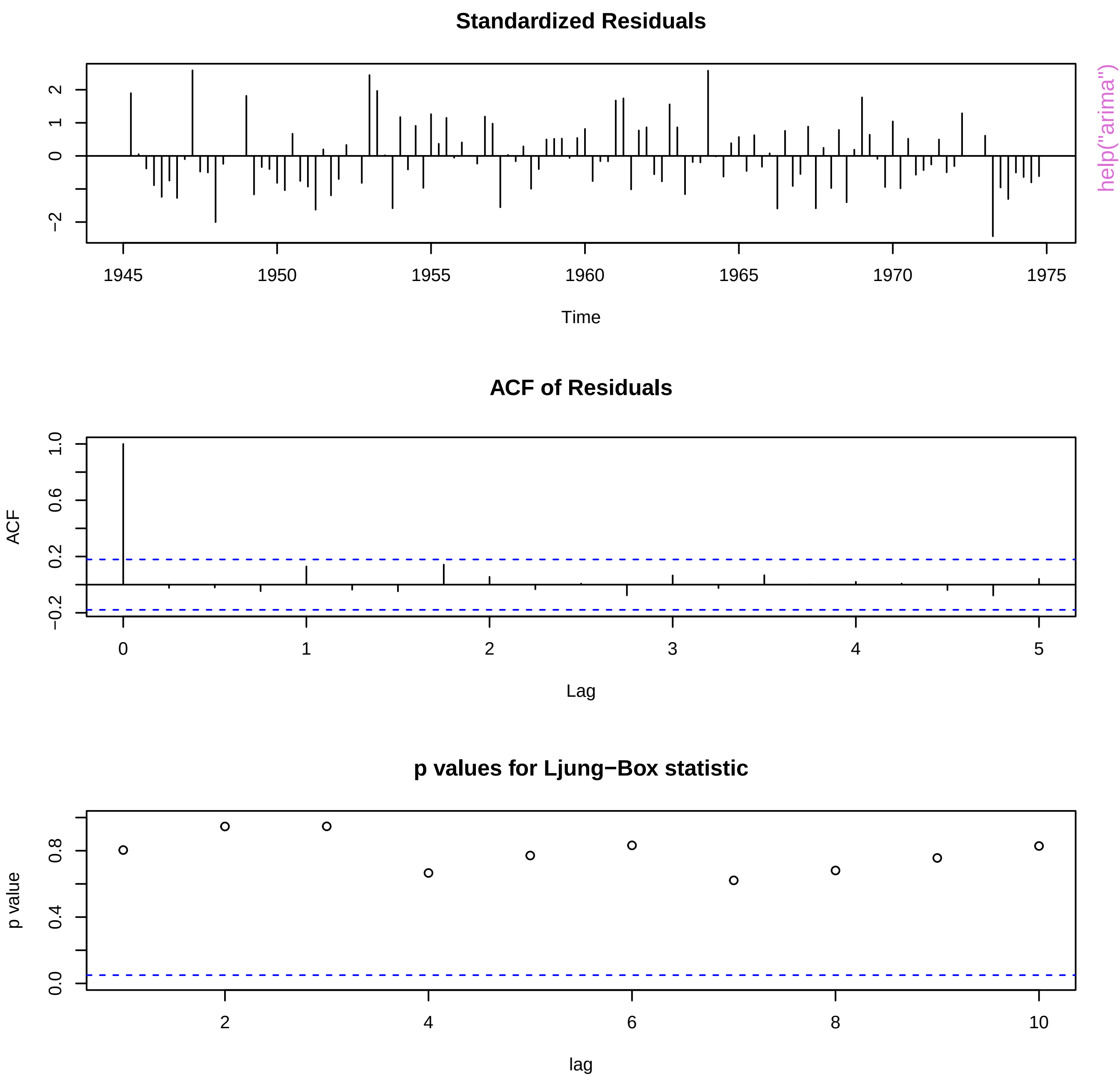


Series presidents
Partial ACF
0.8
0.6
0.4
0.2
0.0
−0.2
1
2
3
4
5
Lag
help("acf")

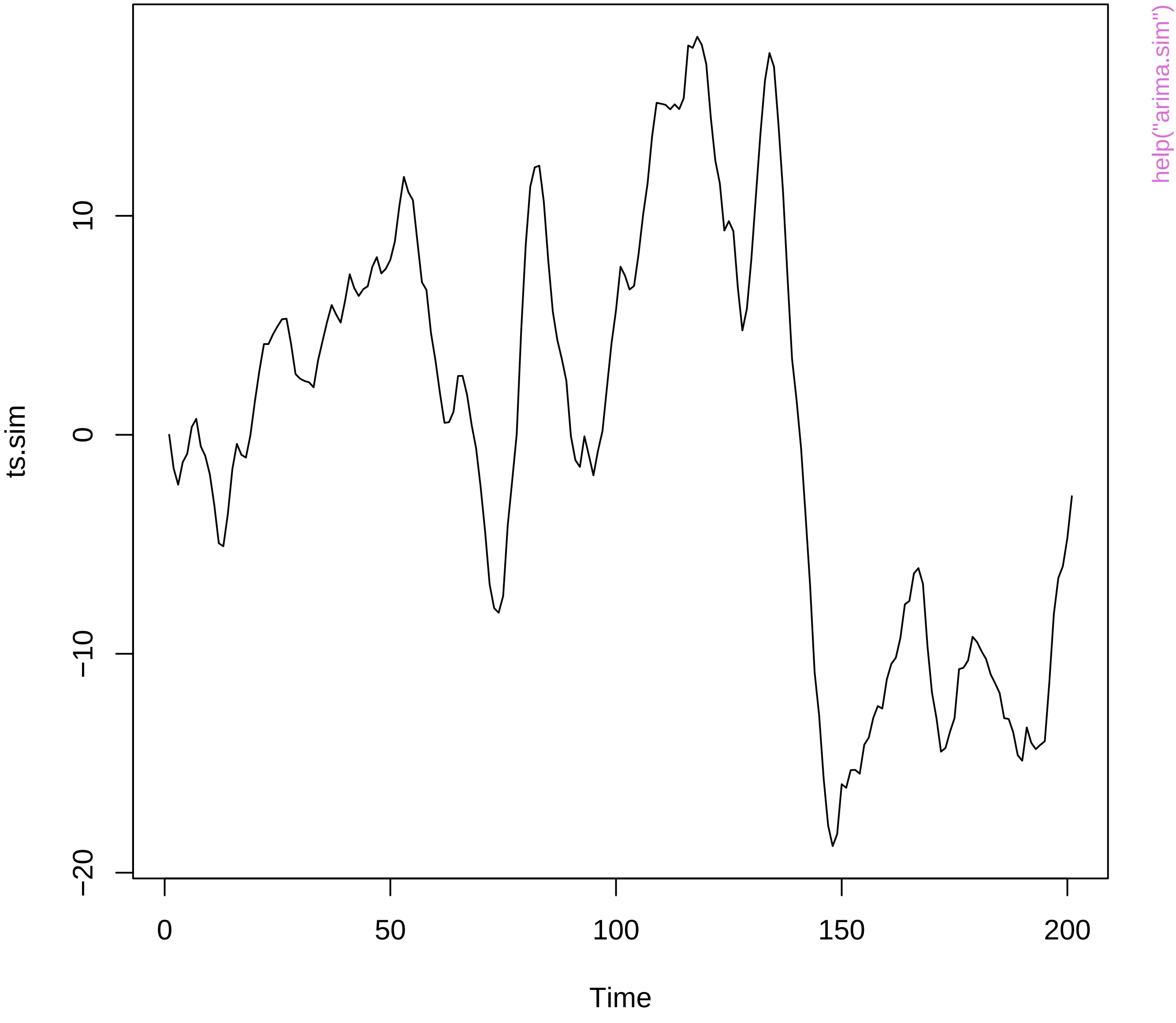

colMeans(reacttime)
450
500
550
600
650
700
750
1
2
3
4
5
6
Index
help("anova.mlm")


approx(.) and approxfun(.)
y
1.5
0.5
−0.5
2
4
6
8
10
x
help("approxfun")
f(x)
1.5
0.5
−0.5
0
2
4
6
8
10
x

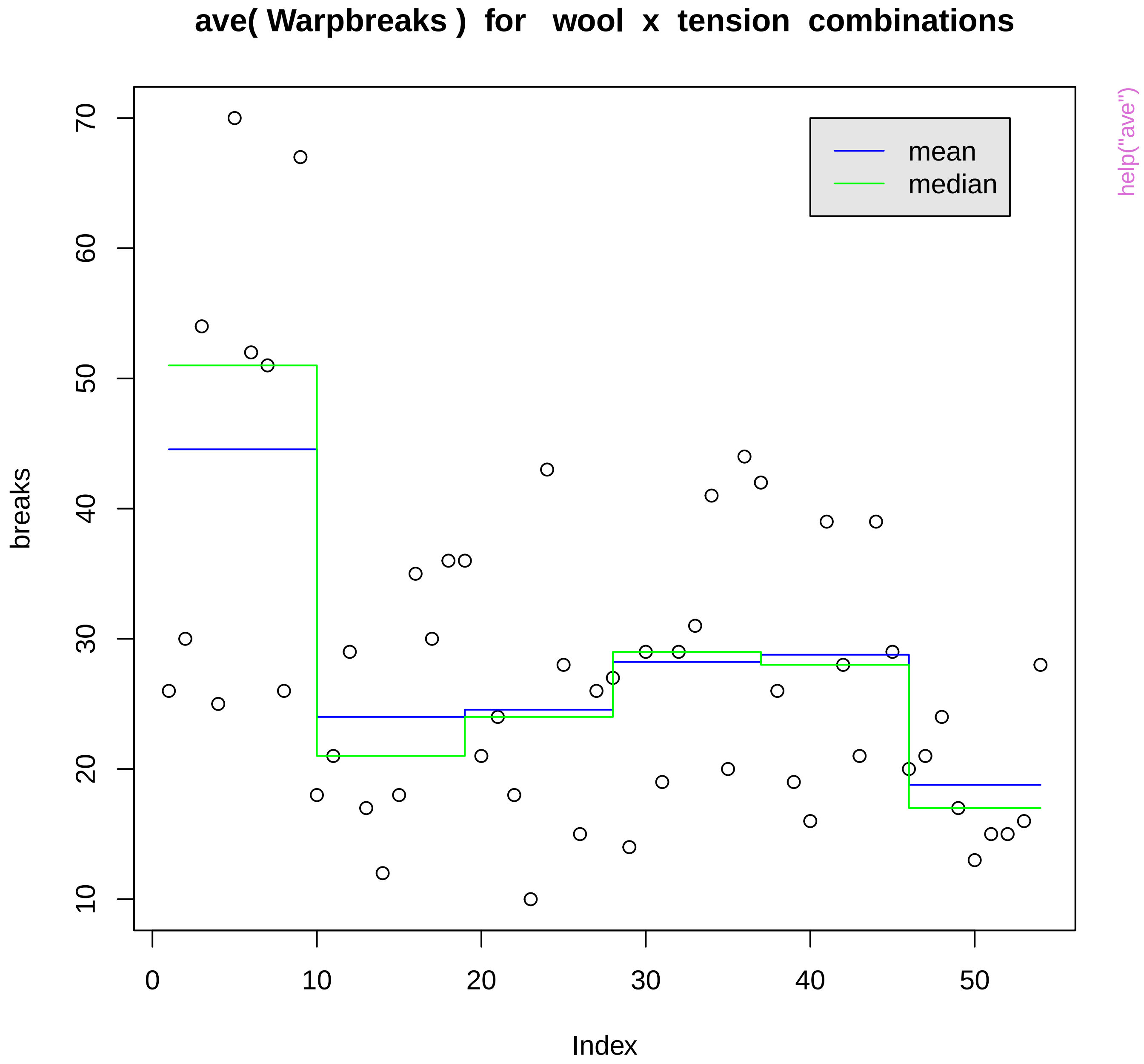

Standardized Residuals
−2
0
1
2
1945
1950
1955
1960
1965
1970
1975
Time
help("arima")
ACF of Residuals
ACF
−0.2
0.2
0.6
1.0
0
1
2
3
4
5
Lag
p values for Ljung−Box statistic
p value
0.0
0.4
0.8
2
4
6
8
10
lag

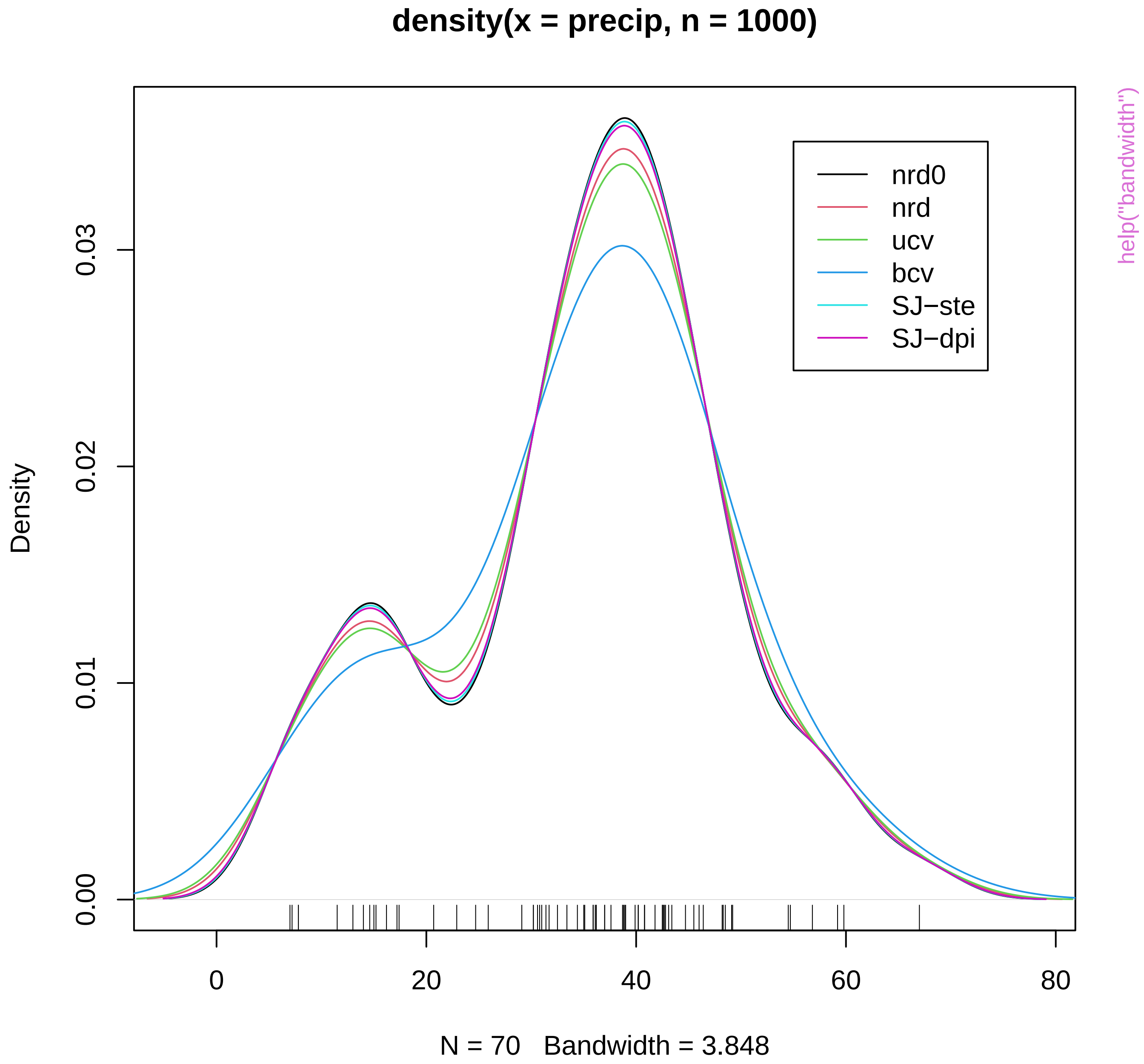


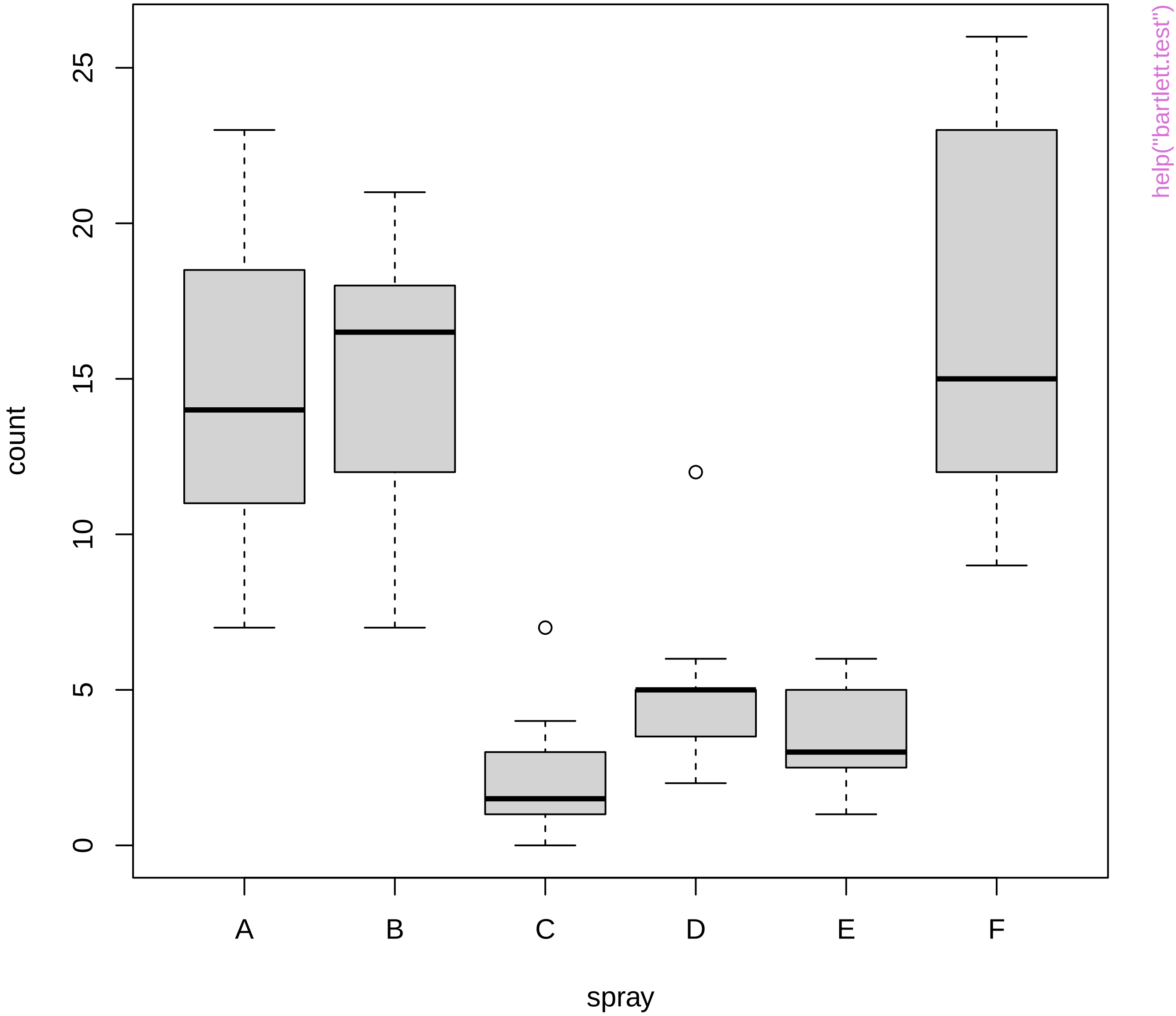
Standardized Residuals
2
1
0
−2
1945
1950
1955
1960
1965
1970
1975
Time
help("arima")
ACF of Residuals
ACF
1.0
0.6
0.2
−0.2
0
1
2
3
4
5
Lag
p values for Ljung−Box statistic
p value
0.8
0.4
0.0
2
4
6
8
10
lag

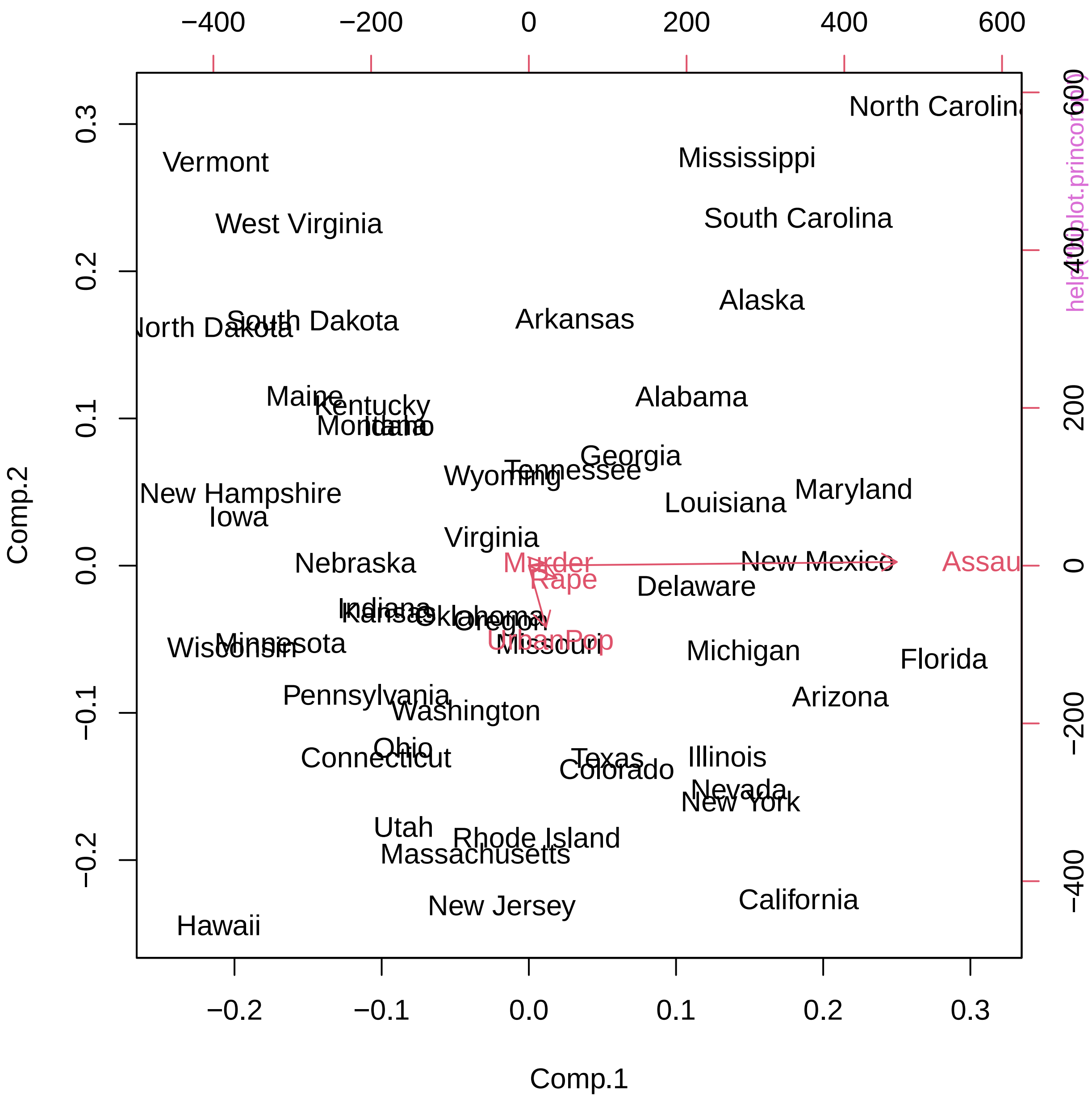


ts.sim
10
0
−10
−20
0
50
100
150
200
Time
help("arima.sim")

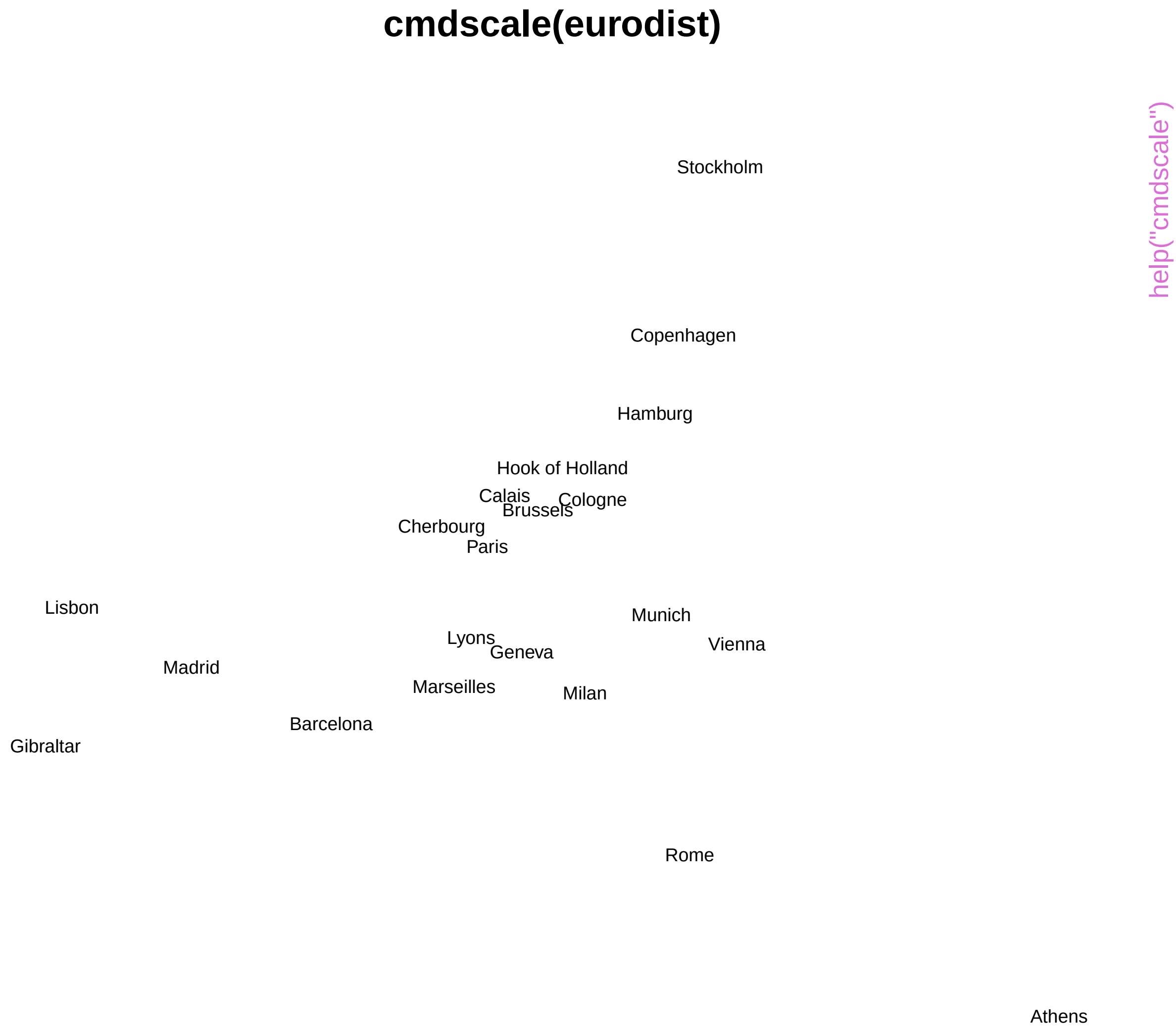

ave(Warpbreaks) for wool x tension combinations
mean
median
breaks
Index
0
10
20
30
40
50
70
60
50
40
30
20
10
help("ave")

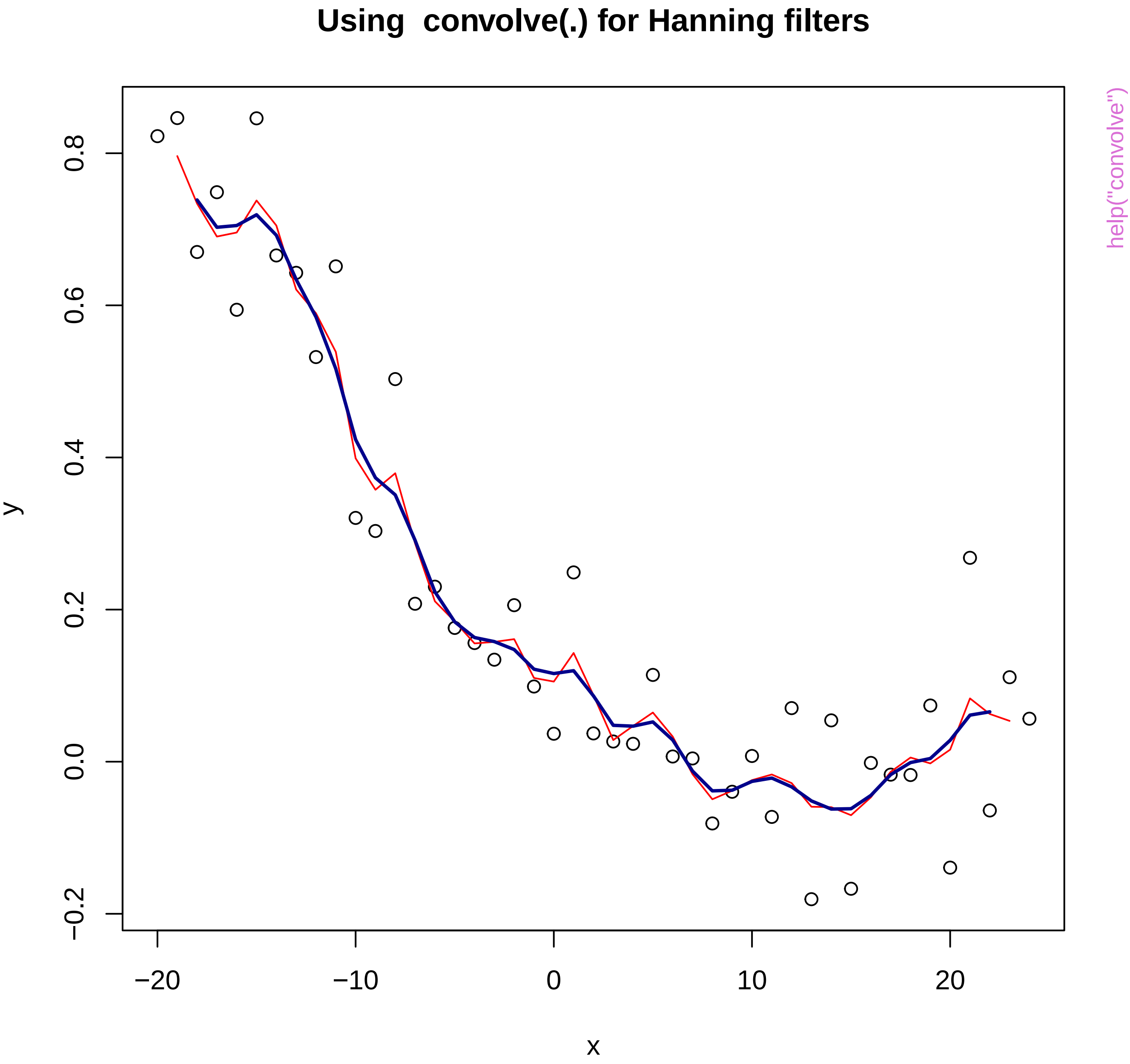

density(x = precip, n = 1000)
Density
0.00
0.01
0.02
0.03
0
20
40
60
80
N = 70 Bandwidth = 3.848
nrd0
nrd
ucv
bcv
SJ−ste
SJ−dpi
help("bandwidth")

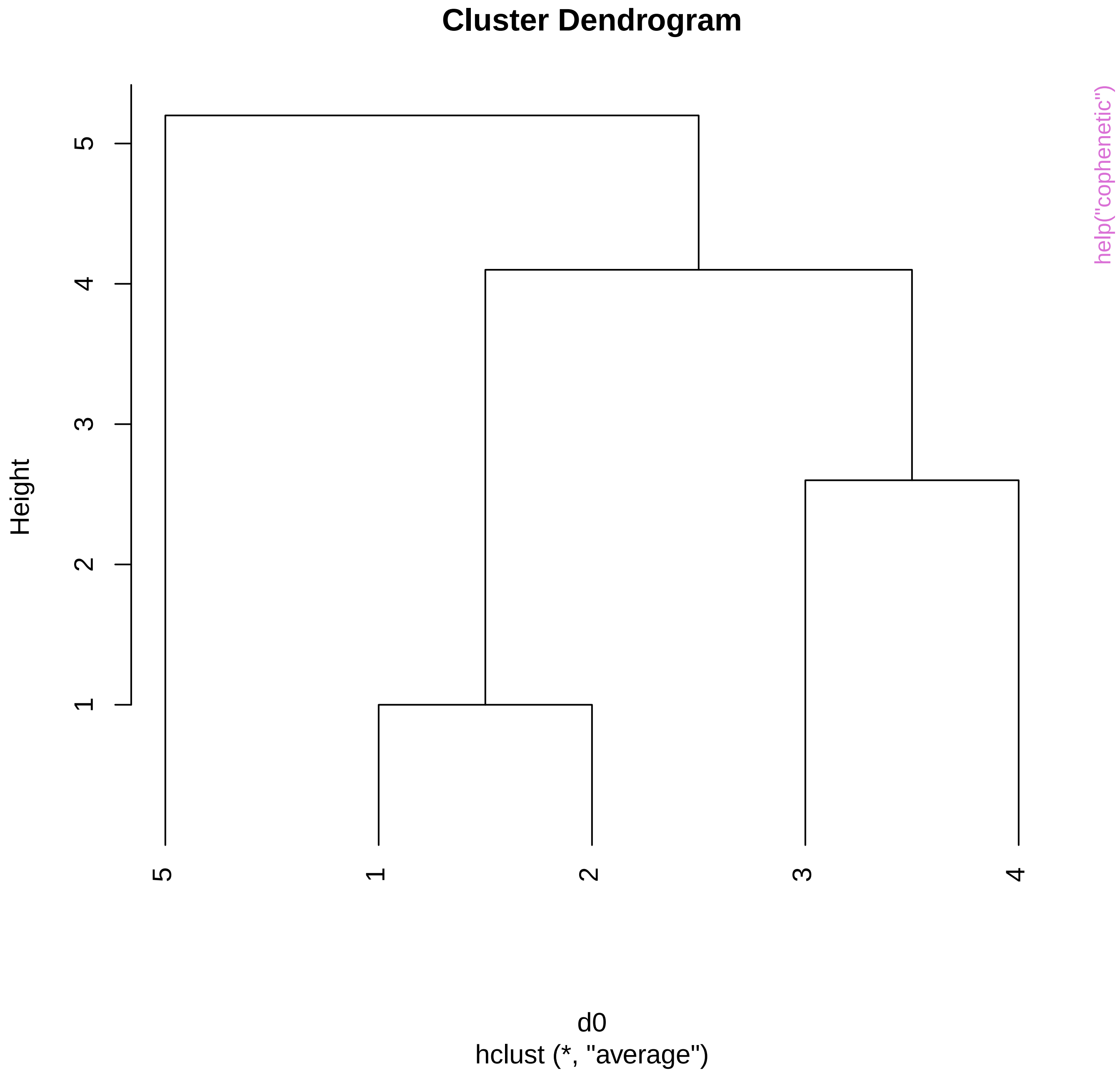

count
spray
0
5
10
15
20
25
A
B
C
D
E
F
help("bartlett.test")

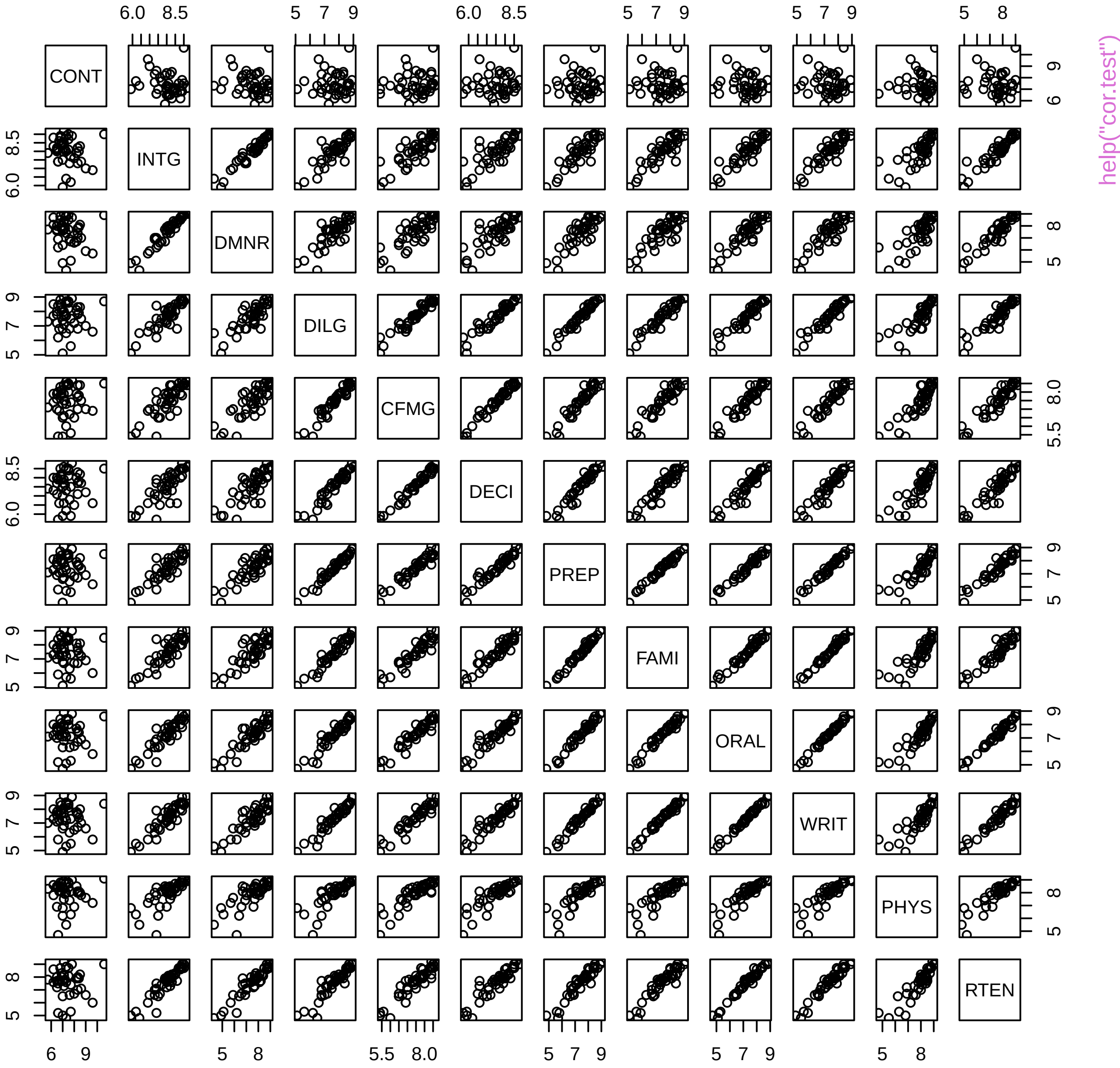


−400
−200
0
200
400
600
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.2
−0.1
0.0
0.1
0.2
0.3
600
400
200
0
−200
−400
Comp.1
Comp.2
help(biplot.princomp)
North Carolina
Vermont
Mississippi
West Virginia
South Carolina
Alaska
North Dakota
South Dakota
Arkansas
Maine
Kentucky
Montana
Idaho
Alabama
Georgia
Tennessee
Wyoming
New Hampshire
Maryland
Louisiana
Iowa
Virginia
Nebraska
Murder
Rape
New Mexico
Assault
Delaware
Indiana
Kansas
Oklahoma
Oregon
UrbanPop
Missouri
Wisconsin
Minnesota
Michigan
Florida
Pennsylvania
Washington
Arizona
Ohio
Connecticut
Texas
Colorado
Illinois
Nevada
New York
Utah
Rhode Island
Massachusetts
New Jersey
California
Hawaii

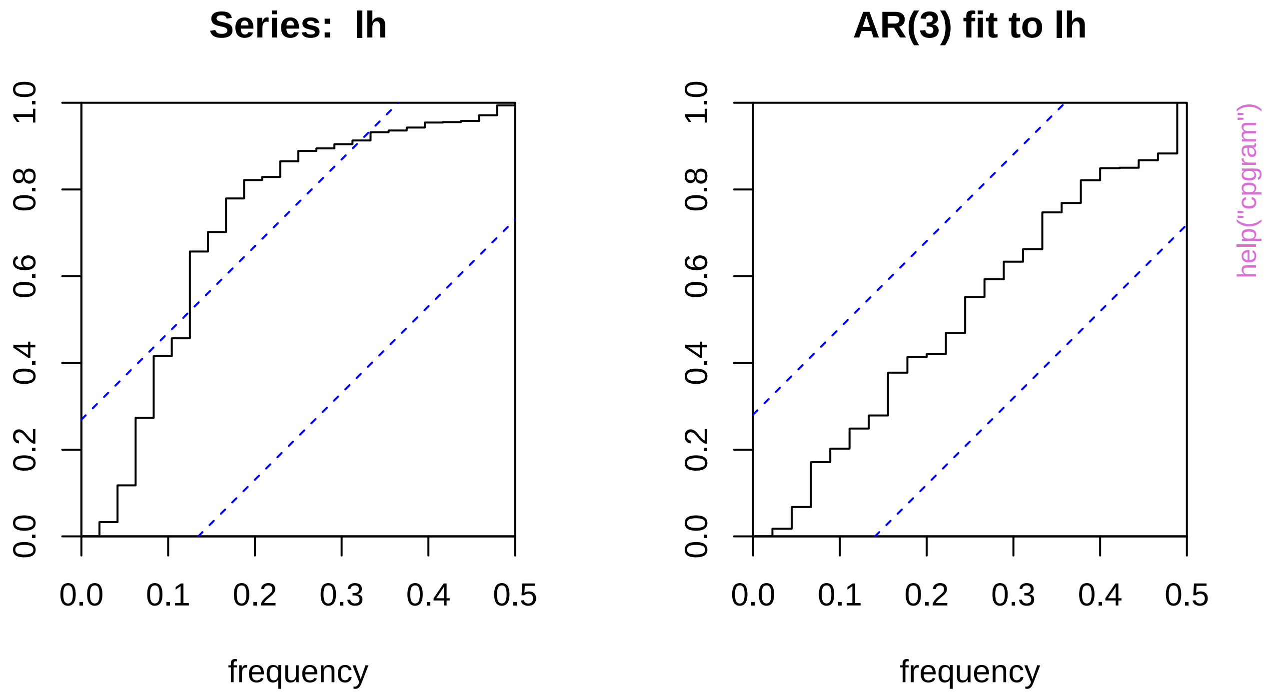

cmdscale(eurodist)
Stockholm
Copenhagen
Hamburg
Hook of Holland
Calais
Cologne
Brussels
Cherbourg
Paris
Lisbon
Munich
Lyons
Vienna
Geneva
Madrid
Marseilles
Milan
Barcelona
Gibraltar
Rome
Athens
help("cmdscale")


Using convolve(.) for Hanning filters
help("convolve")
y
x
−0.2
0.0
0.2
0.4
0.6
0.8
−20
−10
0
10
20

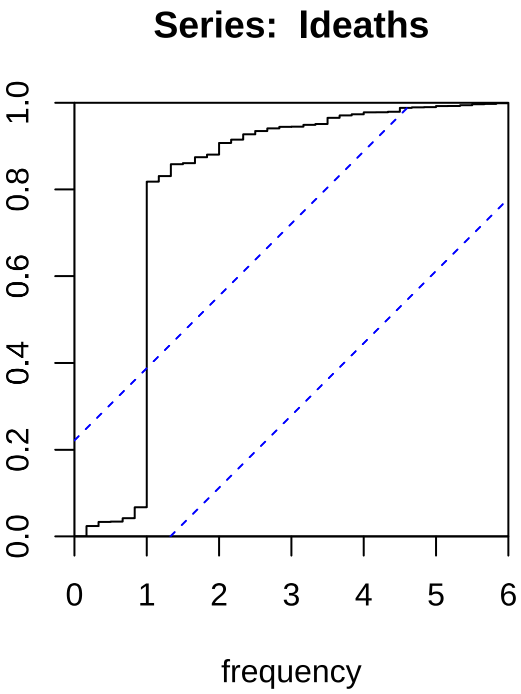

Cluster Dendrogram
Height
5
4
3
2
1
5
1
2
3
4
d0
hclust (*, "average")
help("cophenetic")



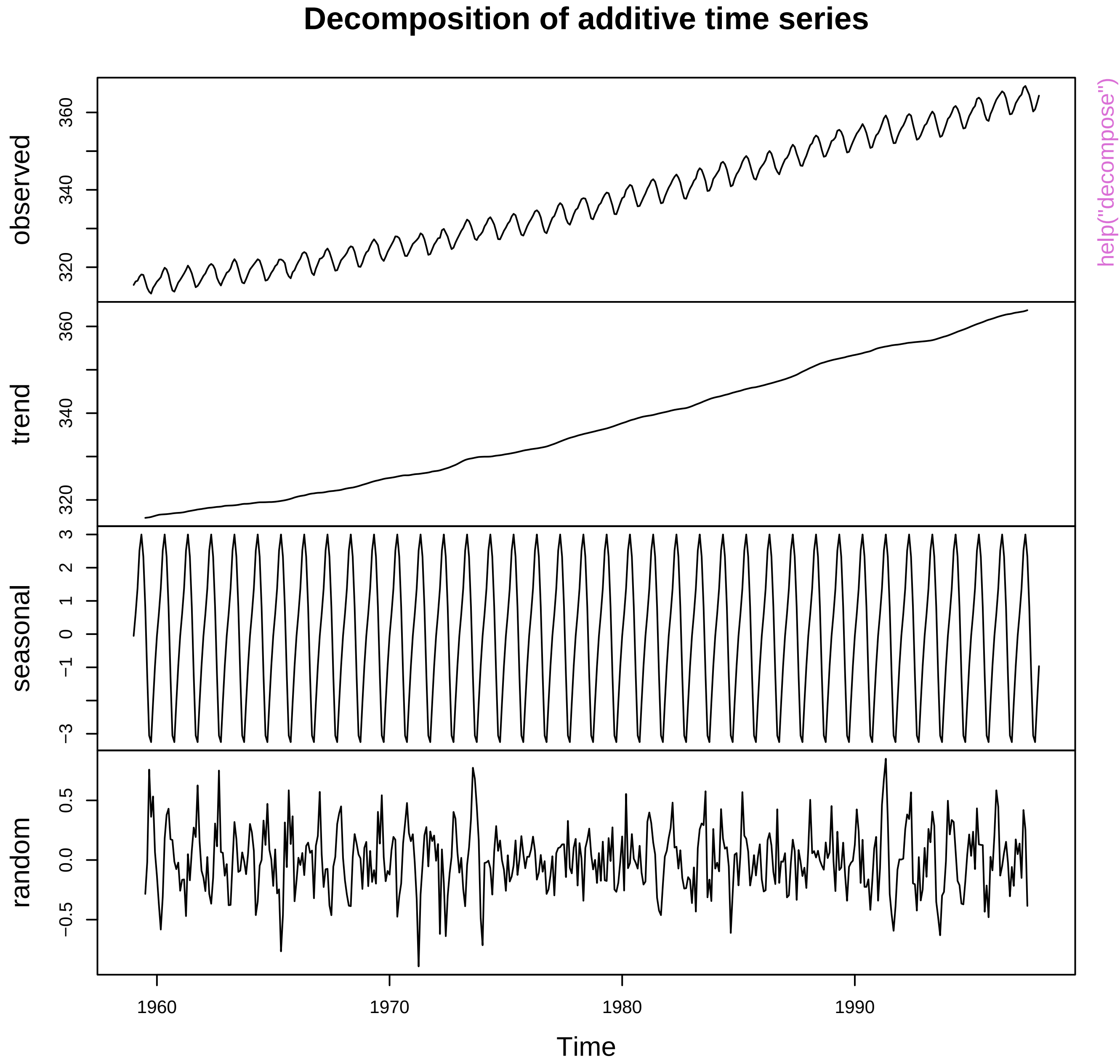
CONT
INTG
DMNR
DILG
CFMG
DECI
PREP
FAMI
ORAL
WRIT
PHYS
RTEN
help("cor.test")



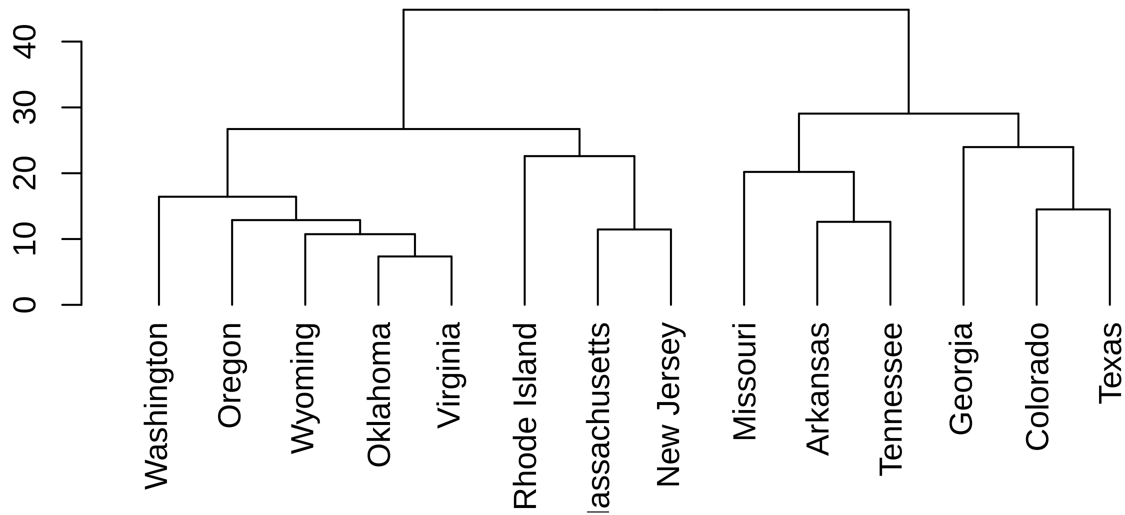
Series: lh
AR(3) fit to lh
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.1
0.2
0.3
0.4
0.5
frequency
frequency
help("cpgram")

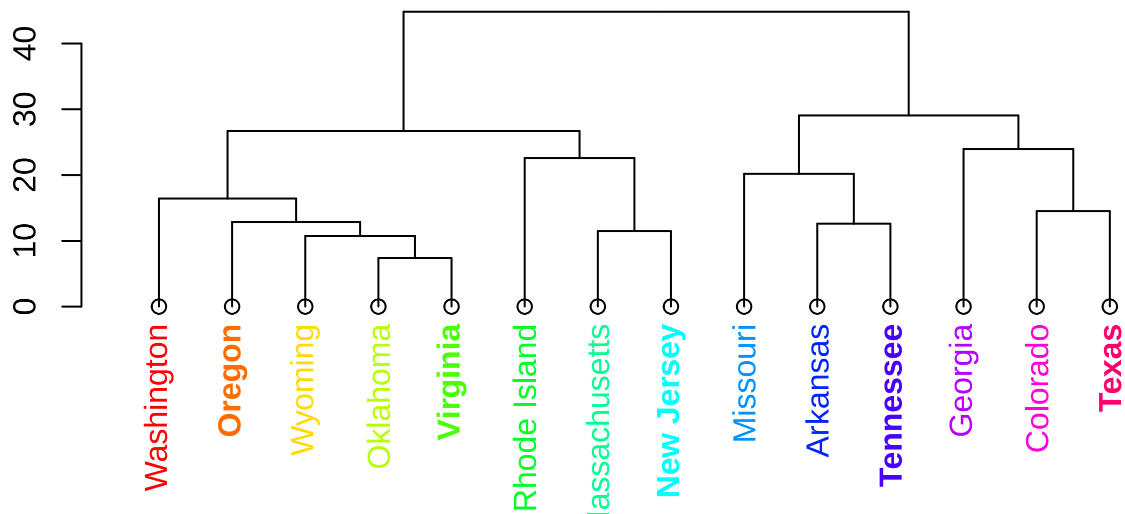


Series: ldeaths
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
frequency

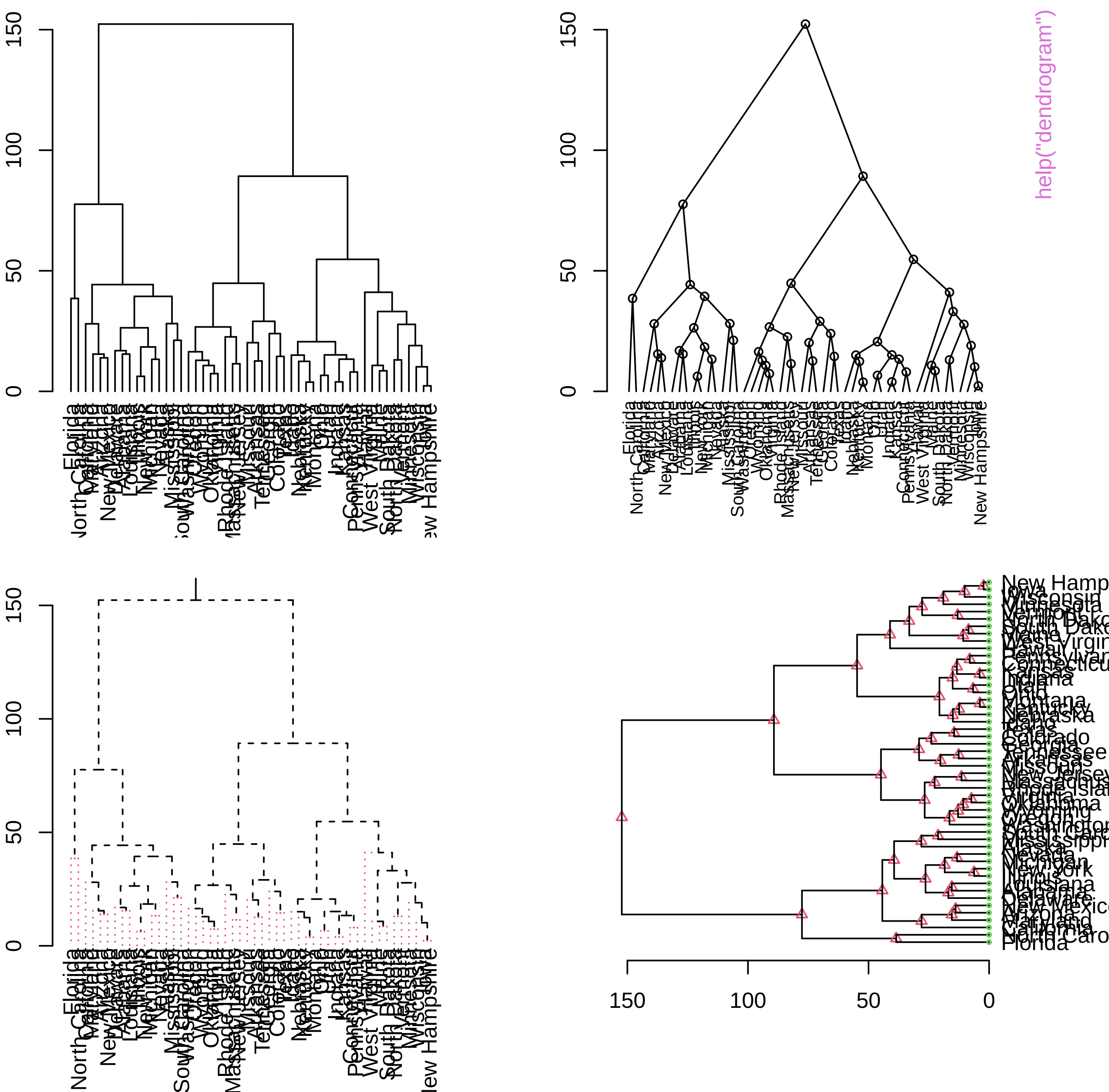


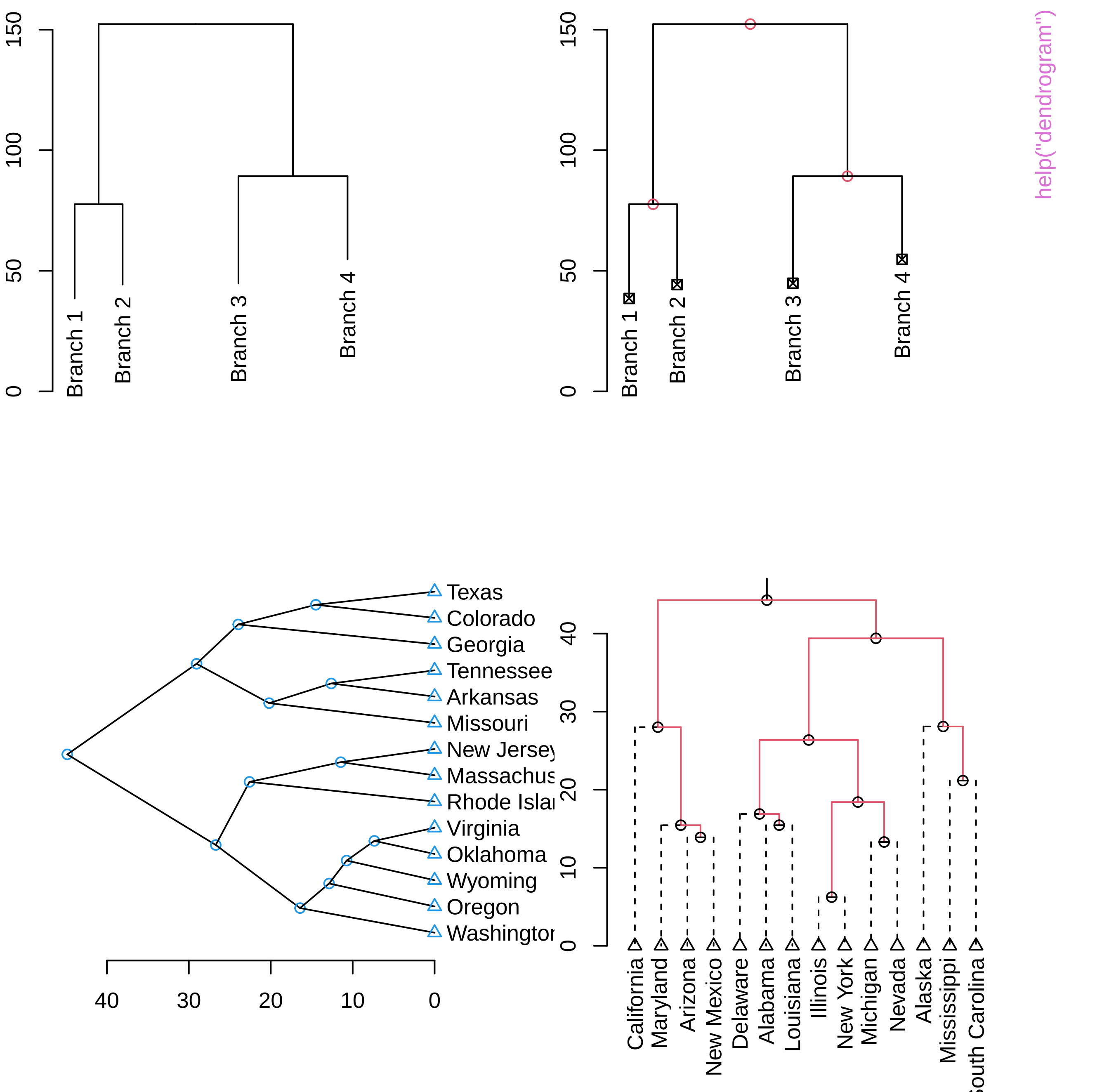
Decomposition of additive time series
observed
trend
seasonal
random
320
340
360
320
340
360
3
2
1
0
−1
−3
0.5
0.0
−0.5
1960
1970
1980
1990
Time
help("decompose")





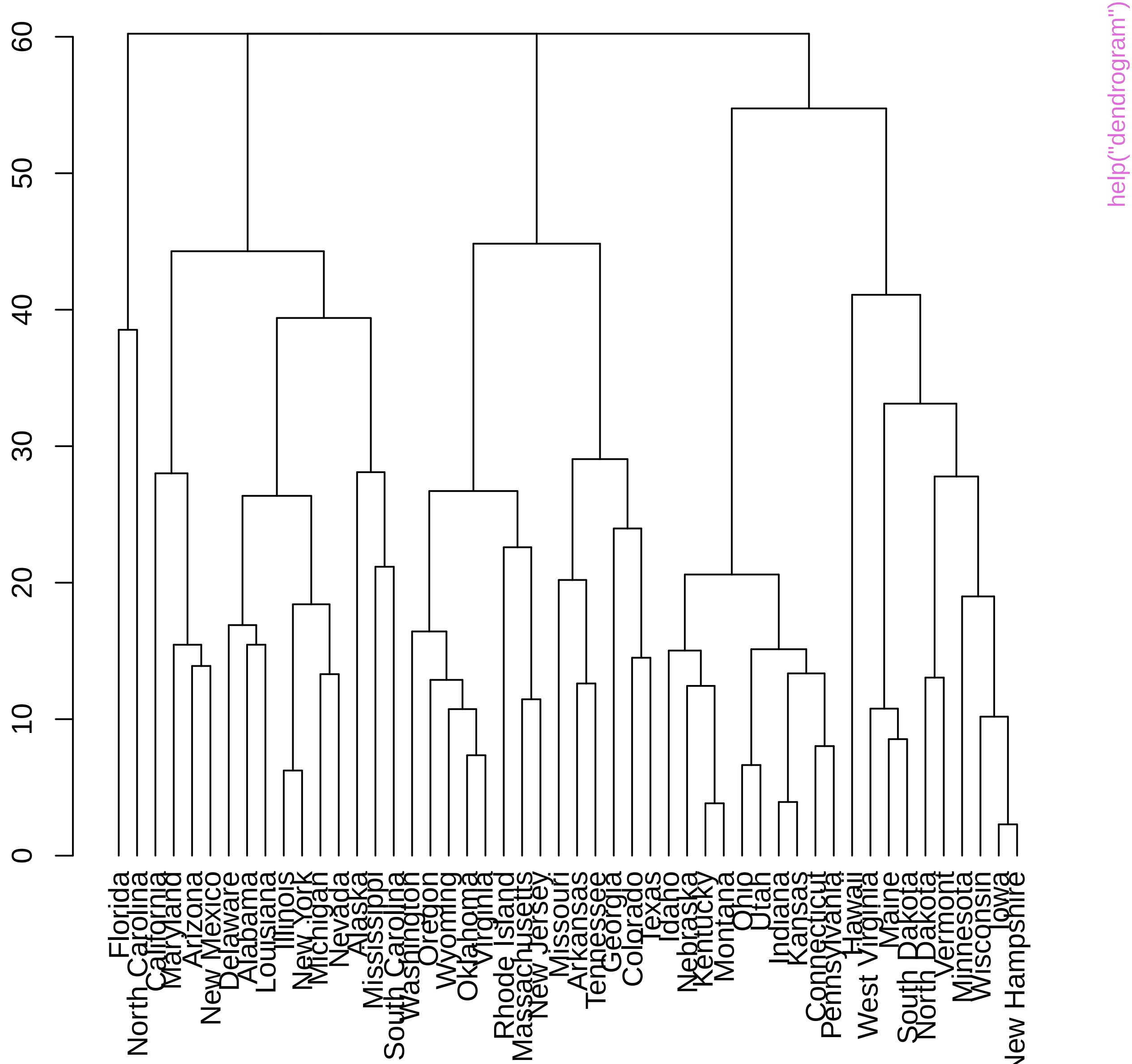
help("dendrapply")

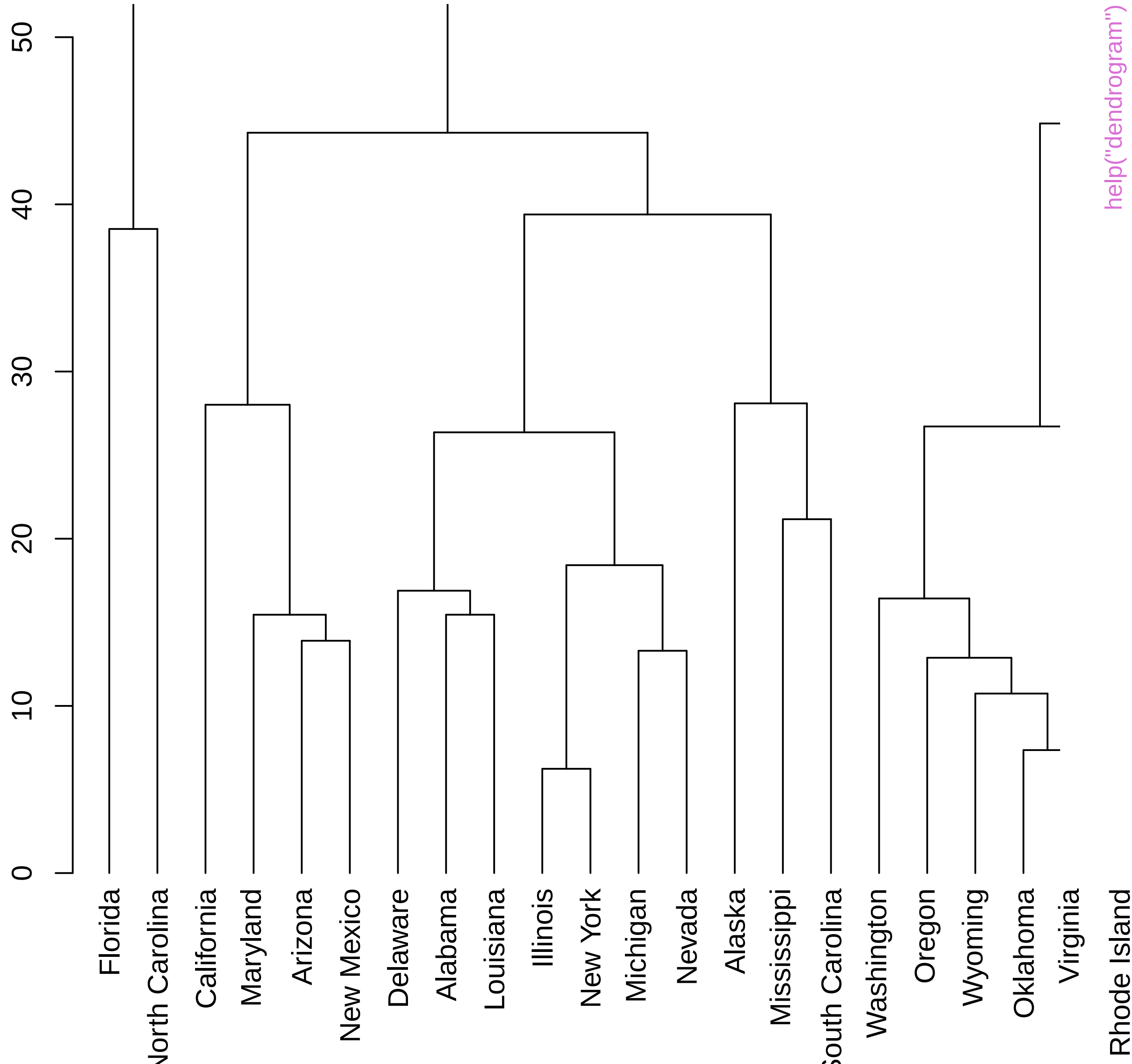

help("dendrogram")
150
100
50
0



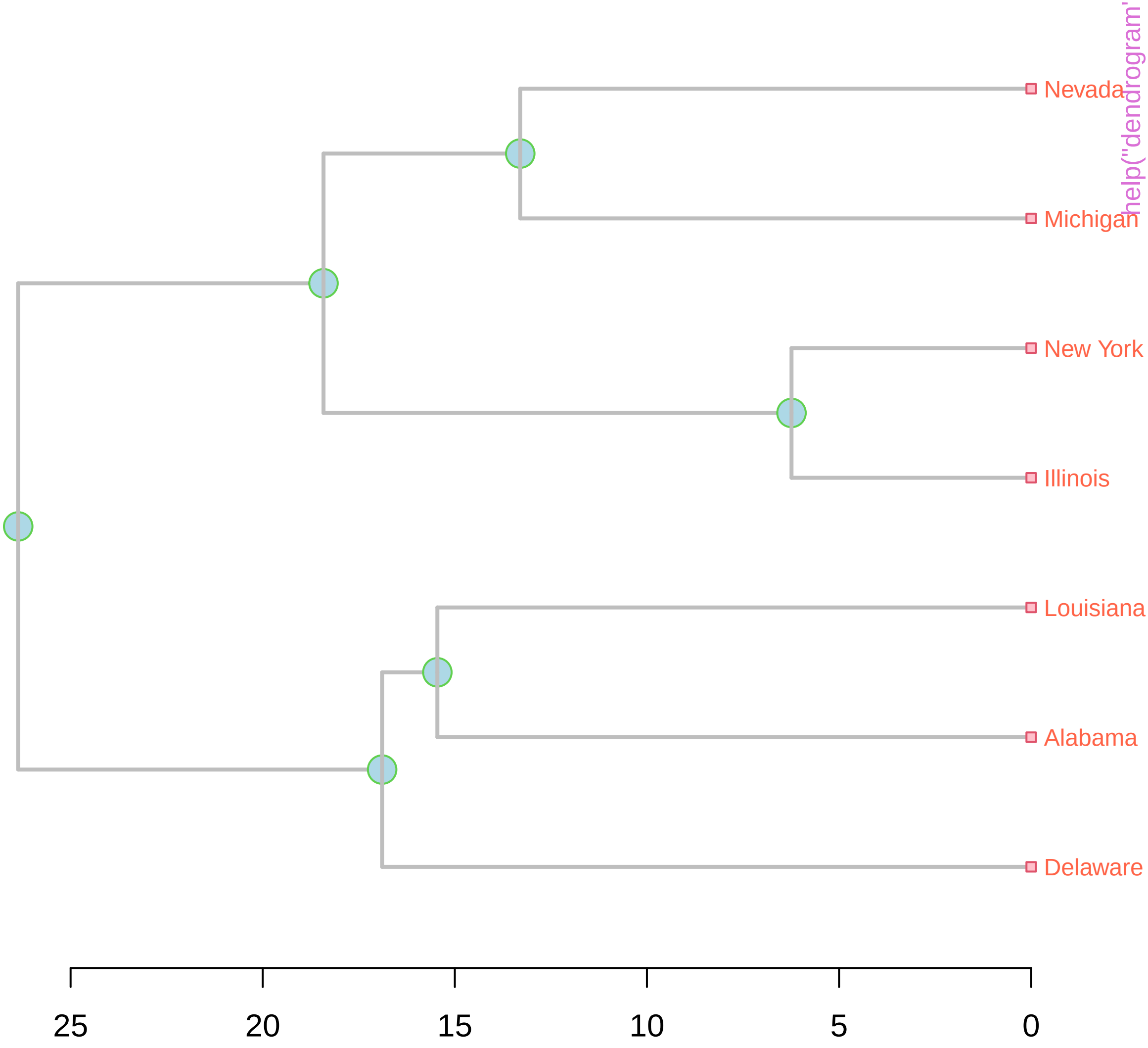
150
100
50
0
Branch 1
Branch 2
Branch 3
Branch 4
help("dendrogram")
Texas
Colorado
Georgia
Tennessee
Arkansas
Missouri
New Jersey
Massachus
Rhode Islan
Virginia
Oklahoma
Wyoming
Oregon
Washington
40
30
20
10
0
California
Maryland
Arizona
New Mexico
Delaware
Alabama
Louisiana
Illinois
New York
Michigan
Nevada
Alaska
Mississippi
South Carolina

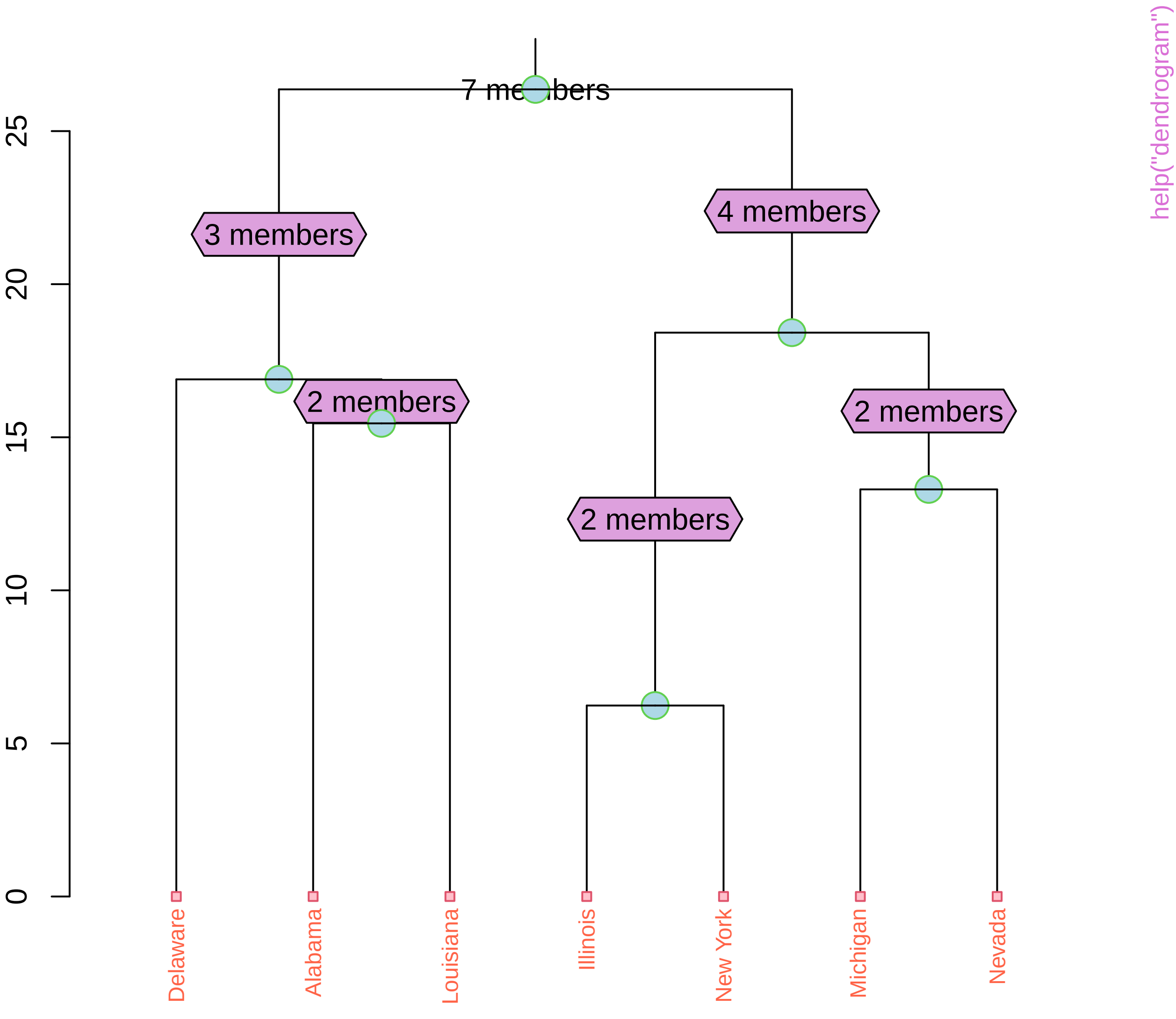
merge(d1, d2, d3, d4) |−> dendrogram with a 4−split



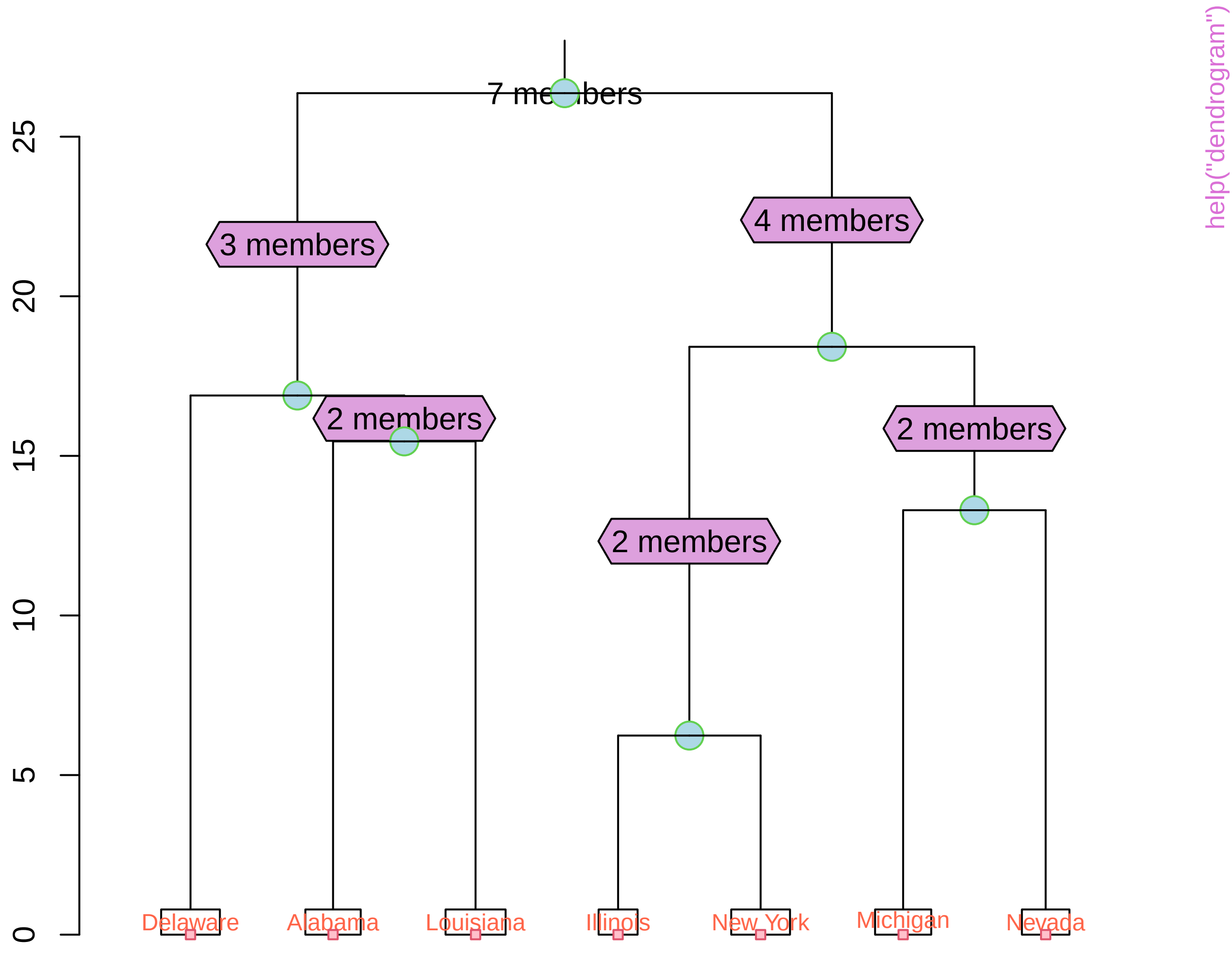


0
10
20
30
40
50
Florida
North Carolina
California
Maryland
Arizona
New Mexico
Delaware
Alabama
Louisiana
Illinois
New York
Michigan
Nevada
Alaska
Mississippi
South Carolina
Washington
Oregon
Wyoming
Oklahoma
Virginia
Rhode Island
help("dendrogram")

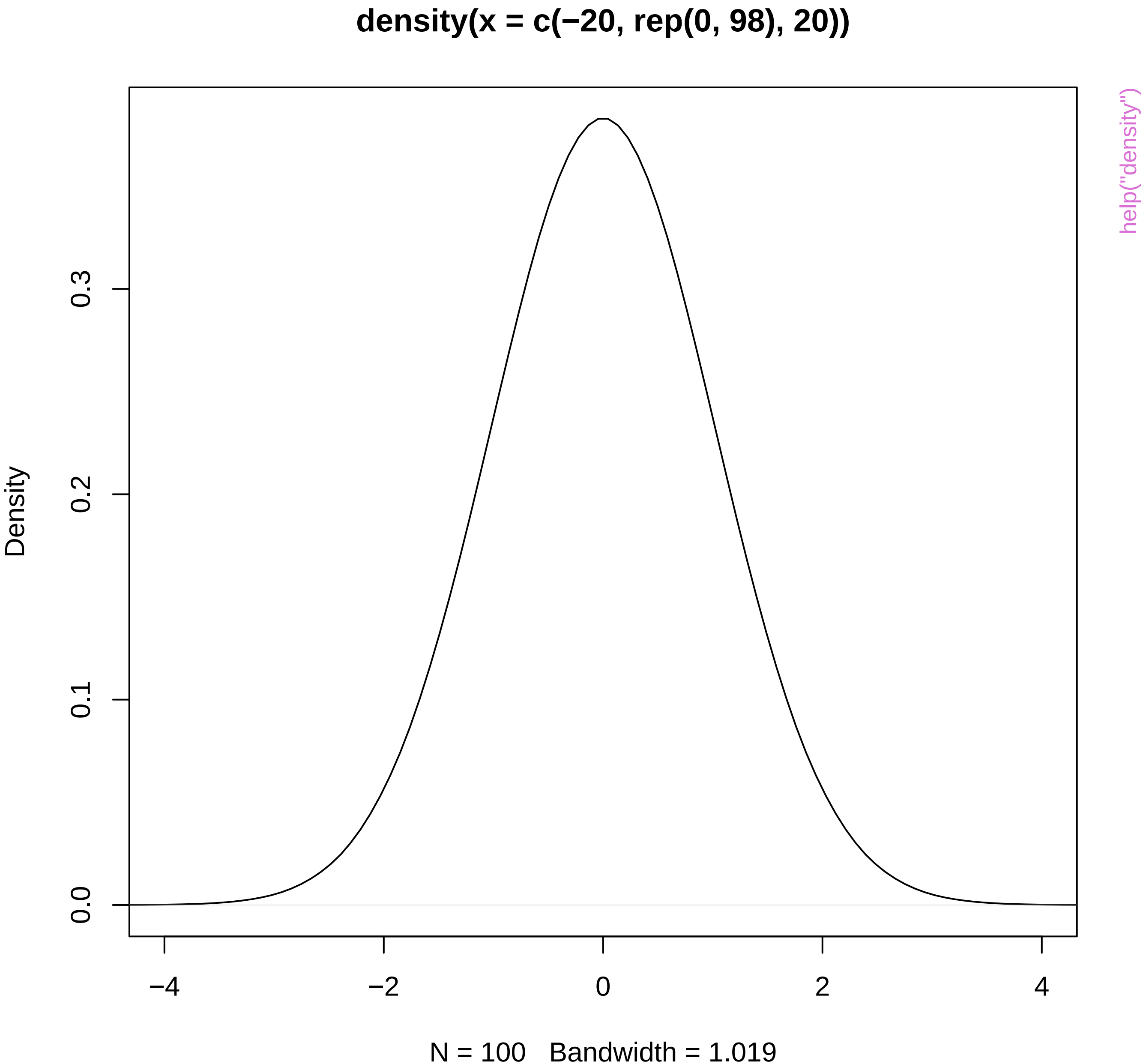

help("dendrogram")
Nevada
Michigan
New York
Illinois
Louisiana
Alabama
Delaware
25
20
15
10
5
0



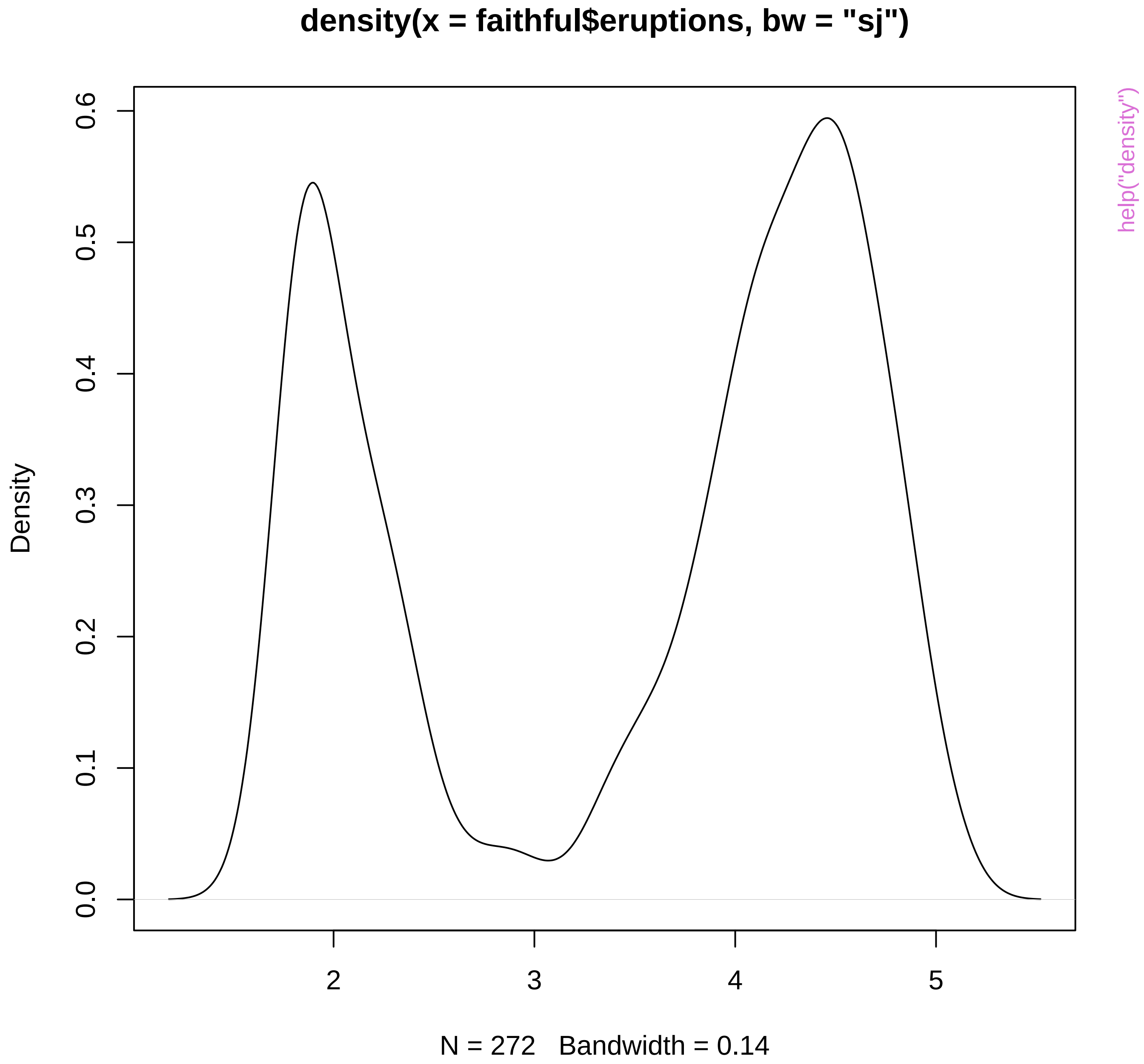
help("dendrogram")
7 members
3 members
4 members
2 members
2 members
2 members
0
5
10
15
20
25
Delaware
Alabama
Louisiana
Illinois
New York
Michigan
Nevada



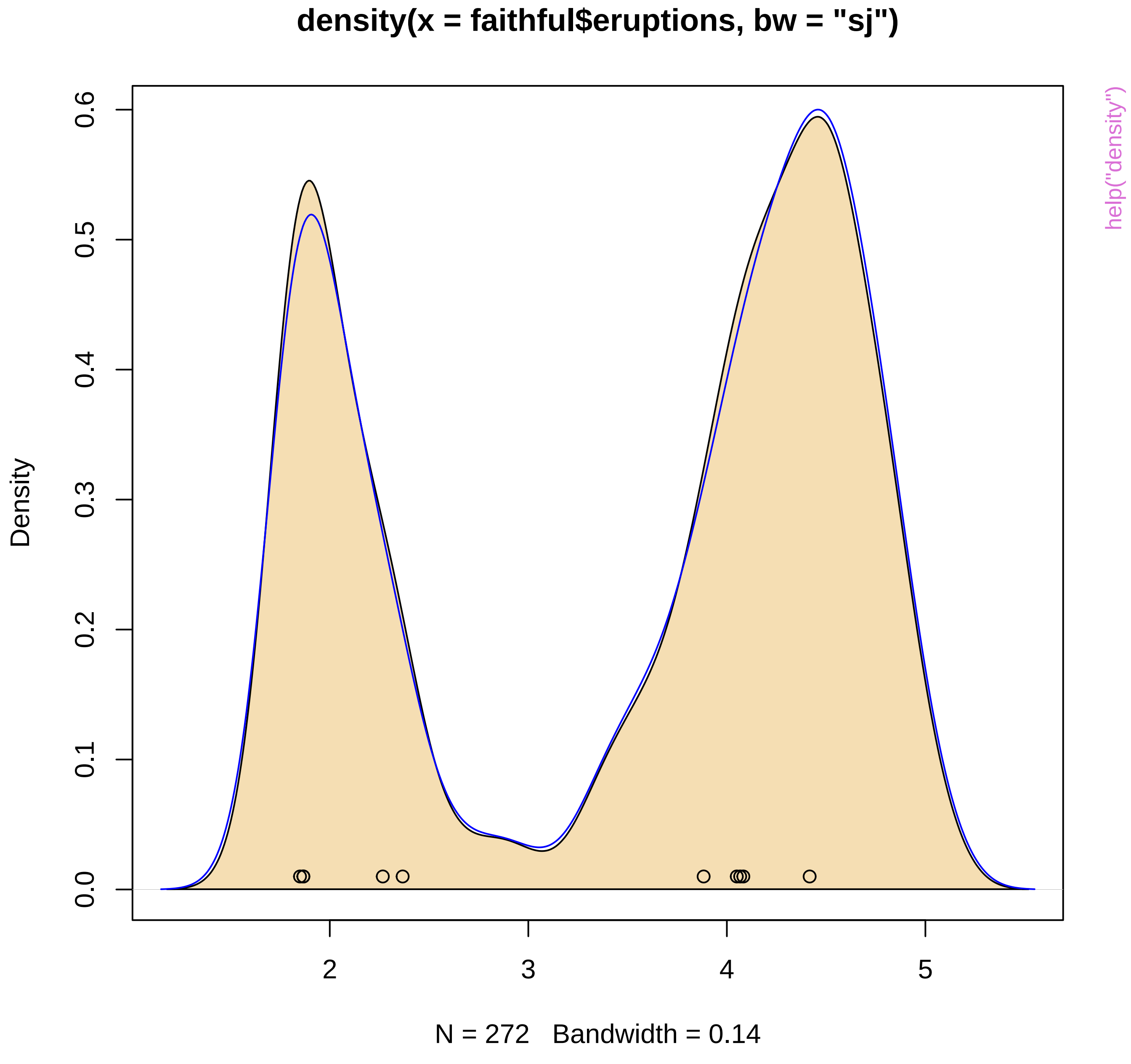
7 members
3 members
4 members
2 members
2 members
2 members
0
5
10
15
20
25
Delaware
Alabama
Louisiana
Illinois
New York
Michigan
Nevada
help("dendrogram")

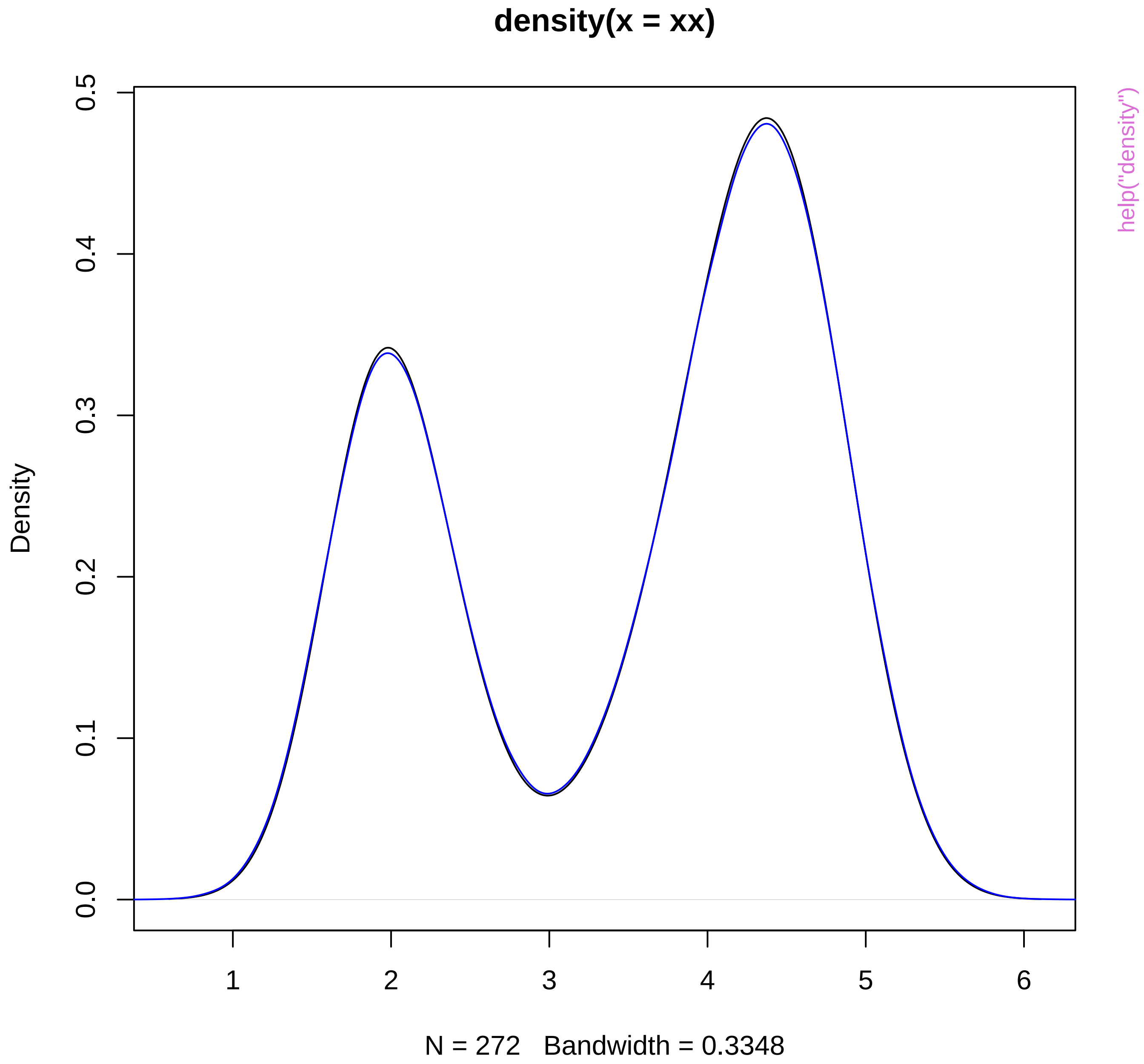


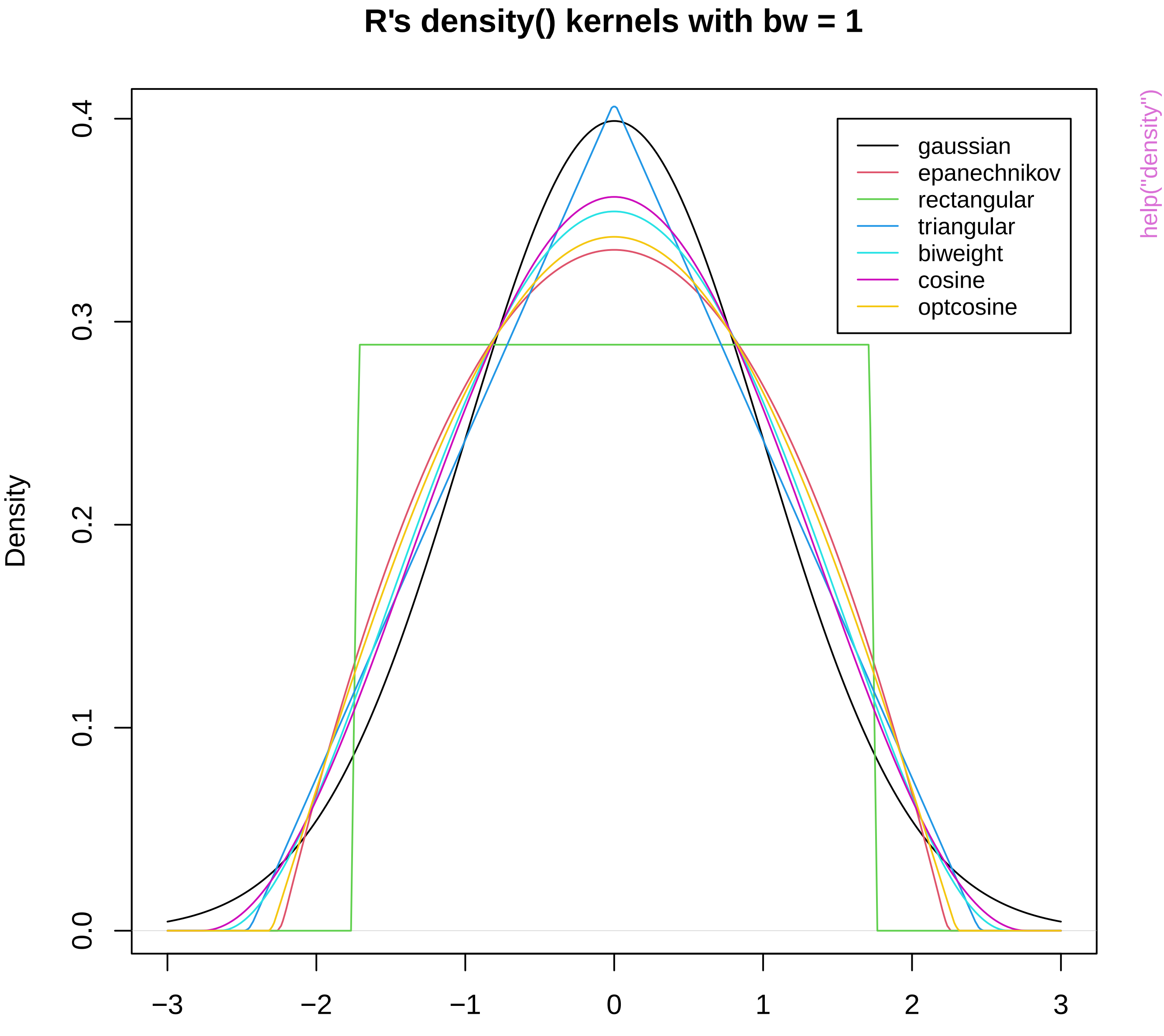
density(x = c(−20, rep(0, 98), 20))
Density
0.0
0.1
0.2
0.3
−4
−2
0
2
4
N = 100 Bandwidth = 1.019
help("density")



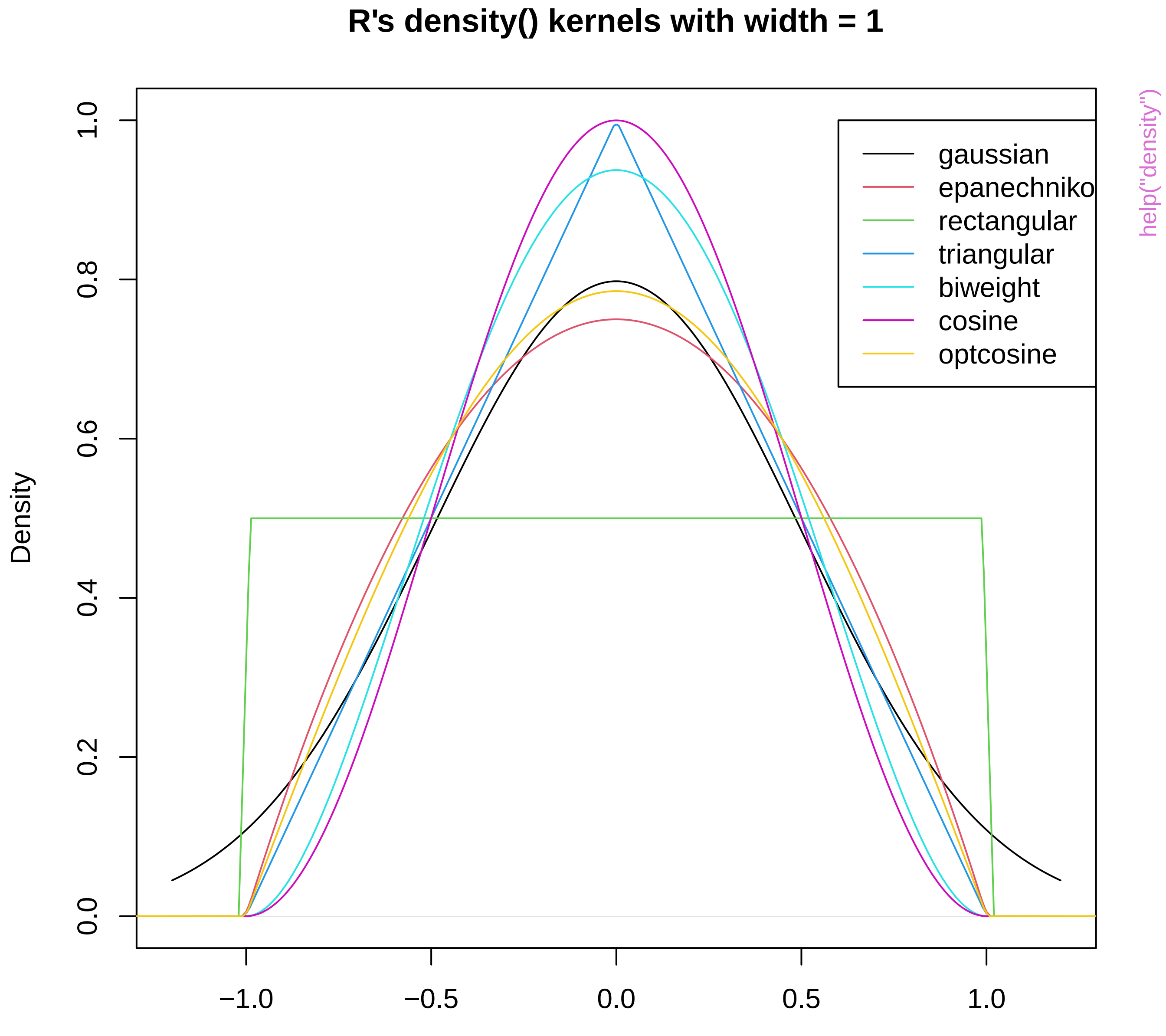
density(x = faithful$eruptions, bw = "sj")
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
2
3
4
5
N = 272 Bandwidth = 0.14
help("density")


density(x = faithful$eruptions, bw = "sj")
Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
2
3
4
5
N = 272   Bandwidth = 0.14
help("density")

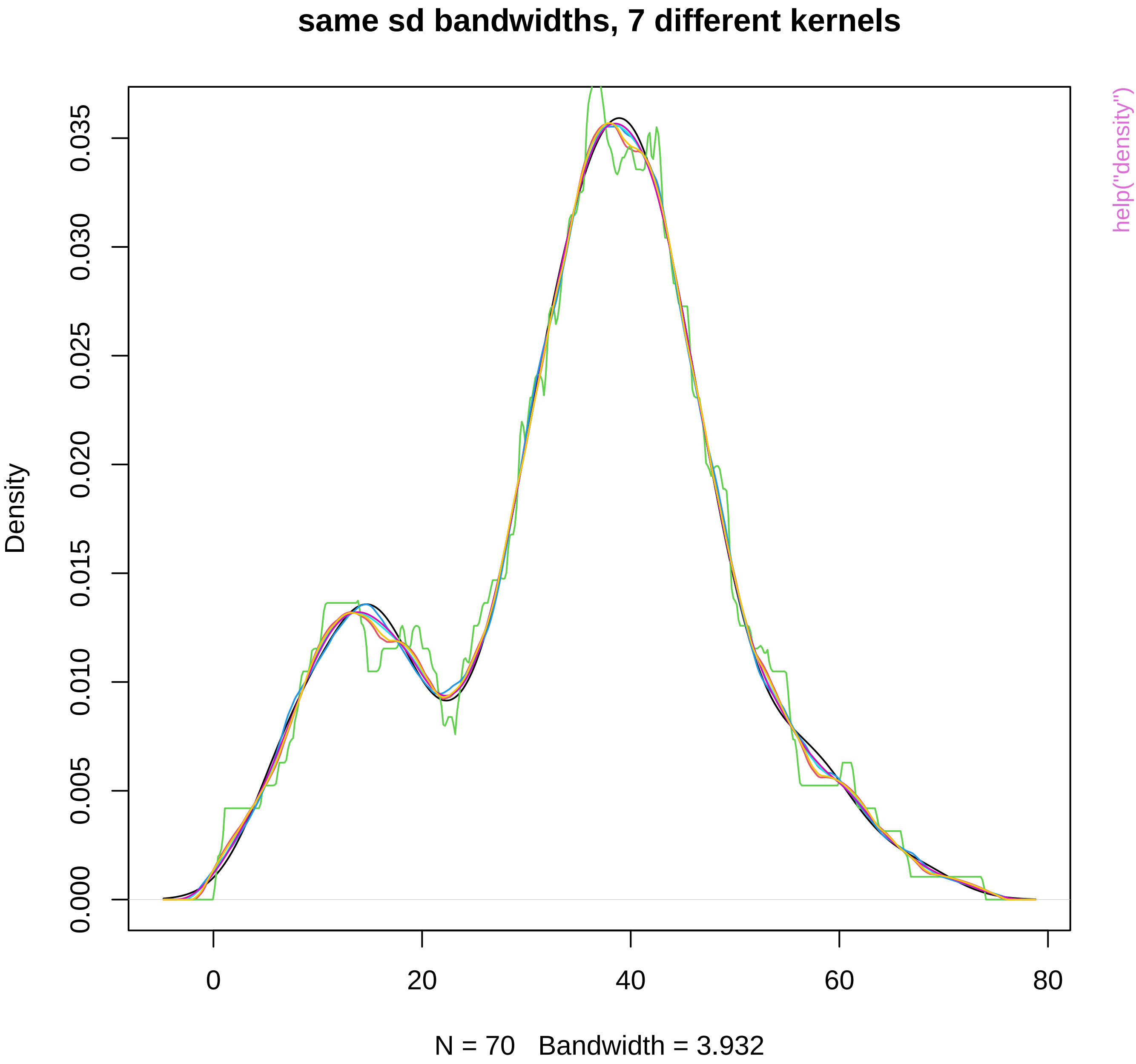

density(x = xx)
Density
0.0
0.1
0.2
0.3
0.4
0.5
1
2
3
4
5
6
N = 272   Bandwidth = 0.3348
help("density")

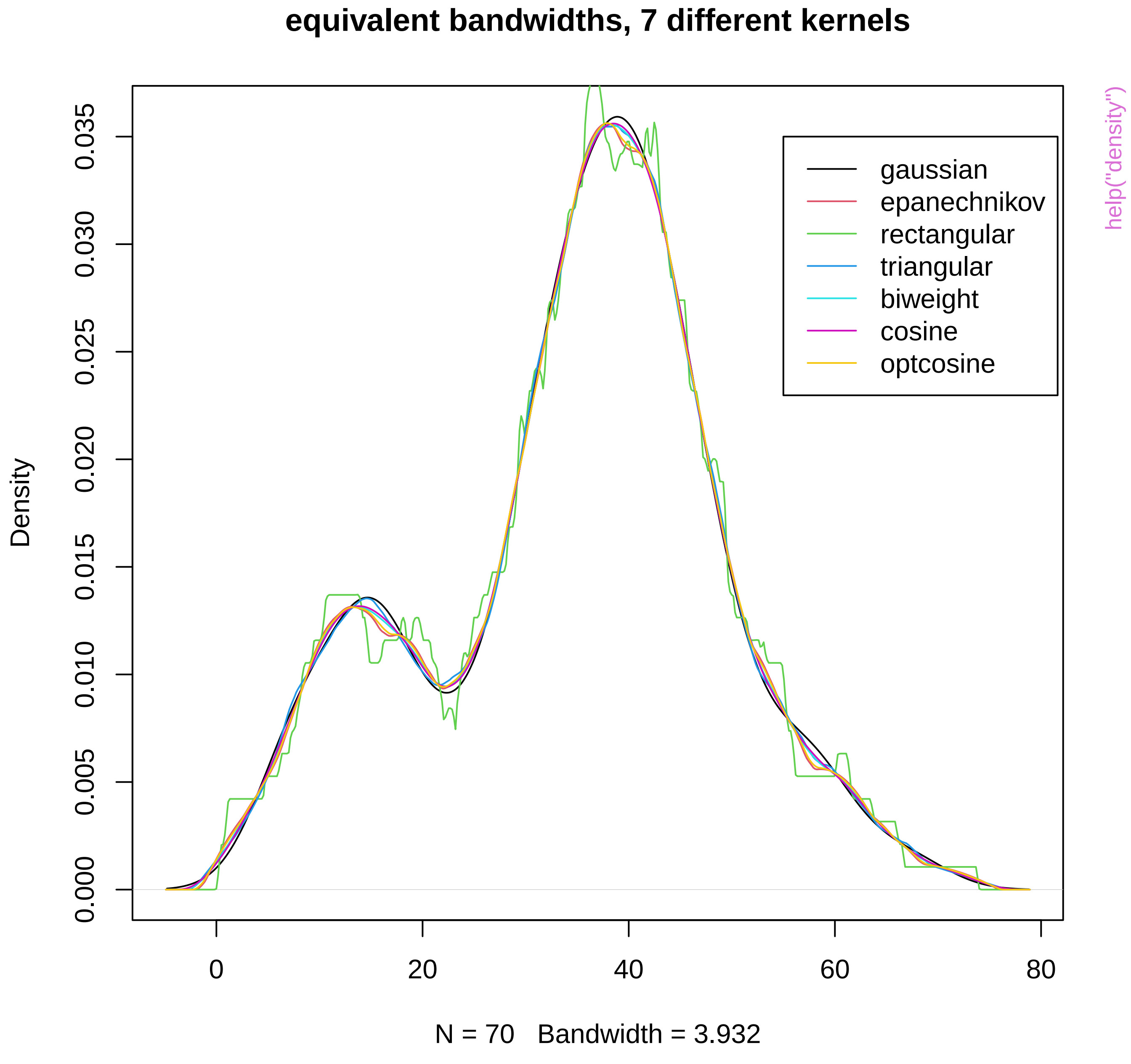


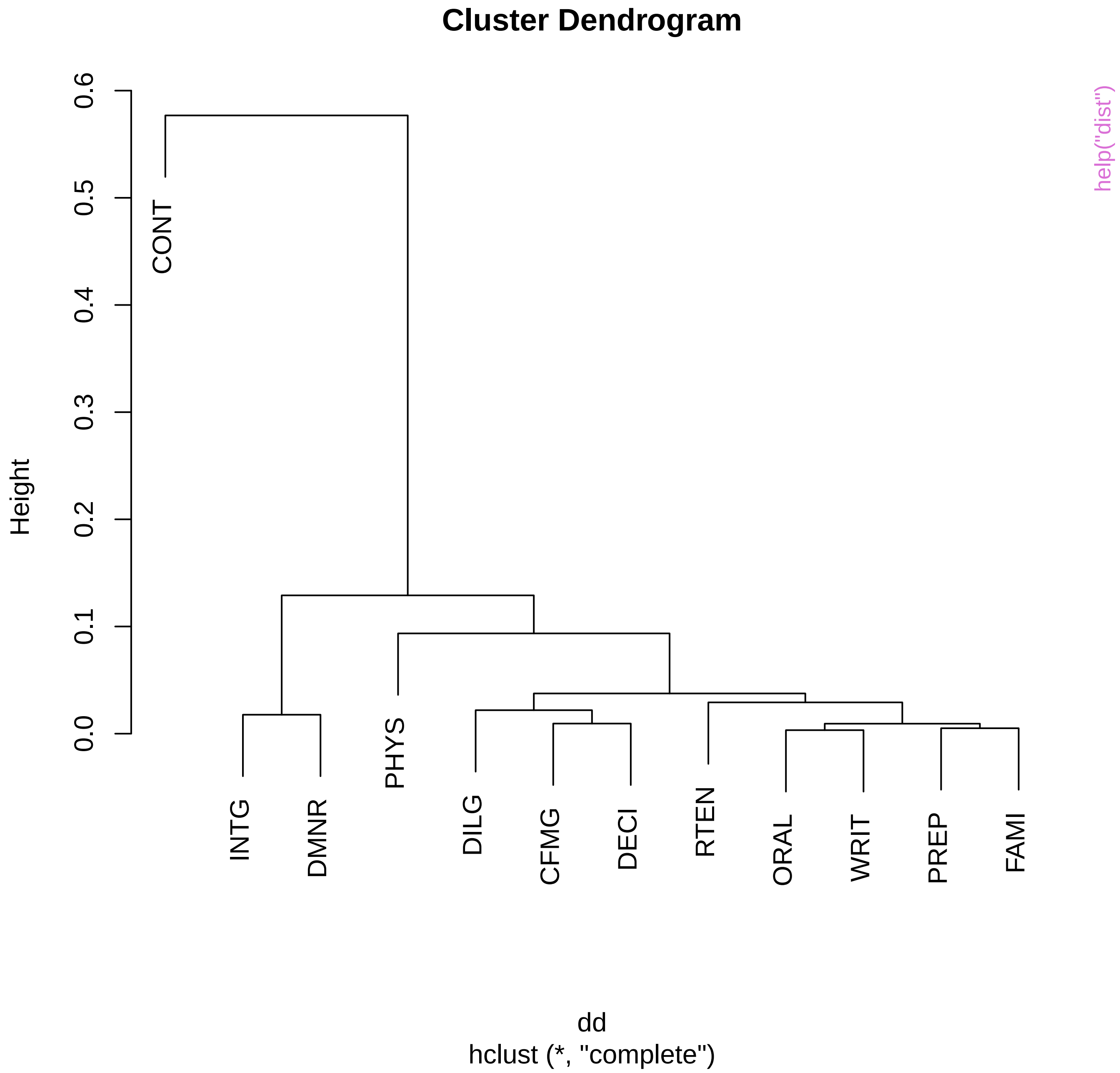
R's density() kernels with bw = 1
Density
0.0
0.1
0.2
0.3
0.4
−3
−2
−1
0
1
2
3
gaussian
epanechnikov
rectangular
triangular
biweight
cosine
optcosine
help("density")



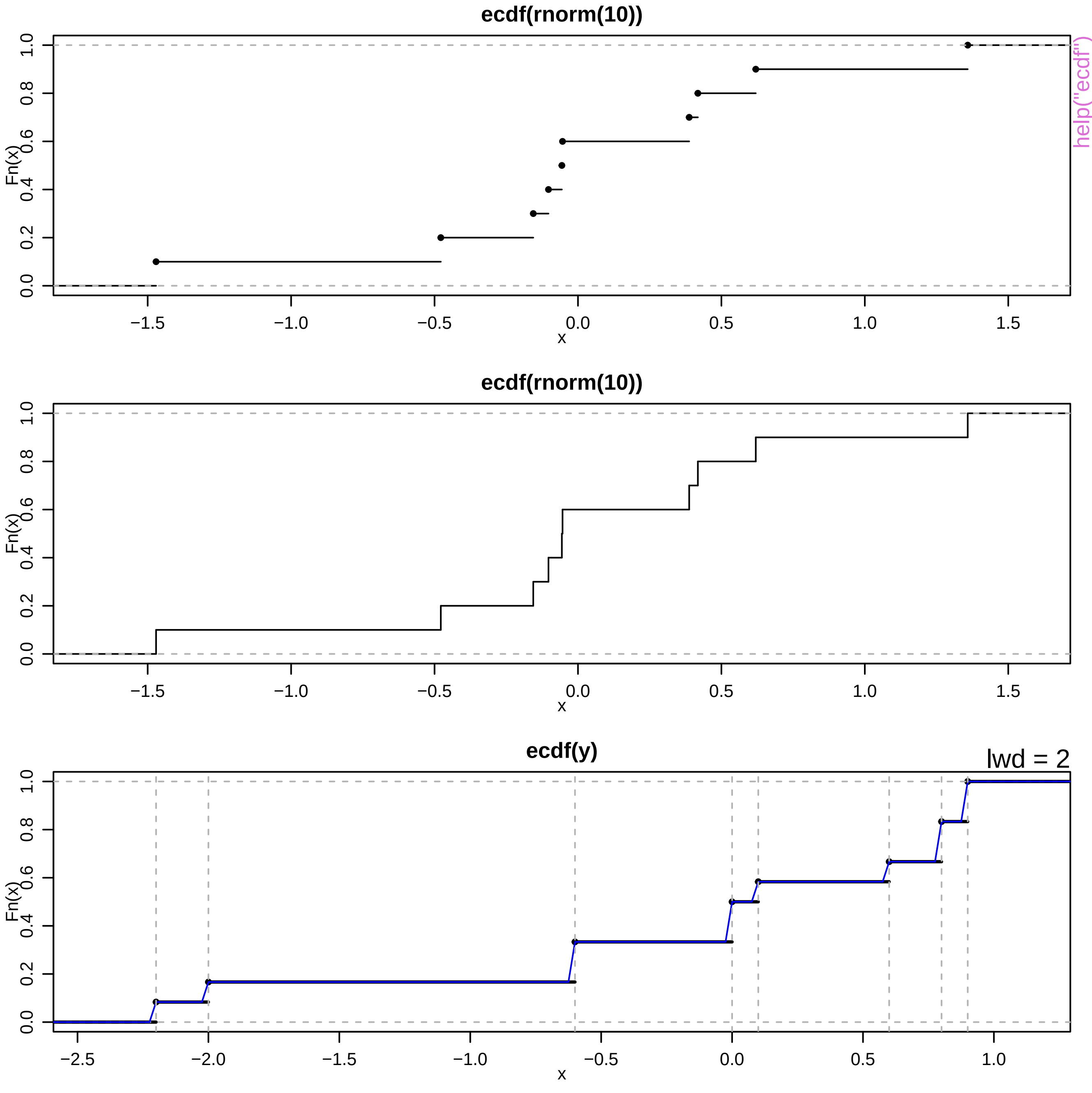
R's density() kernels with width = 1
Density
1.0
0.8
0.6
0.4
0.2
0.0
−1.0
−0.5
0.0
0.5
1.0
gaussian
epanechniko
rectangular
triangular
biweight
cosine
optcosine
help("density")


same sd bandwidths, 7 different kernels
Density
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0
20
40
60
80
N = 70   Bandwidth = 3.932
help("density")

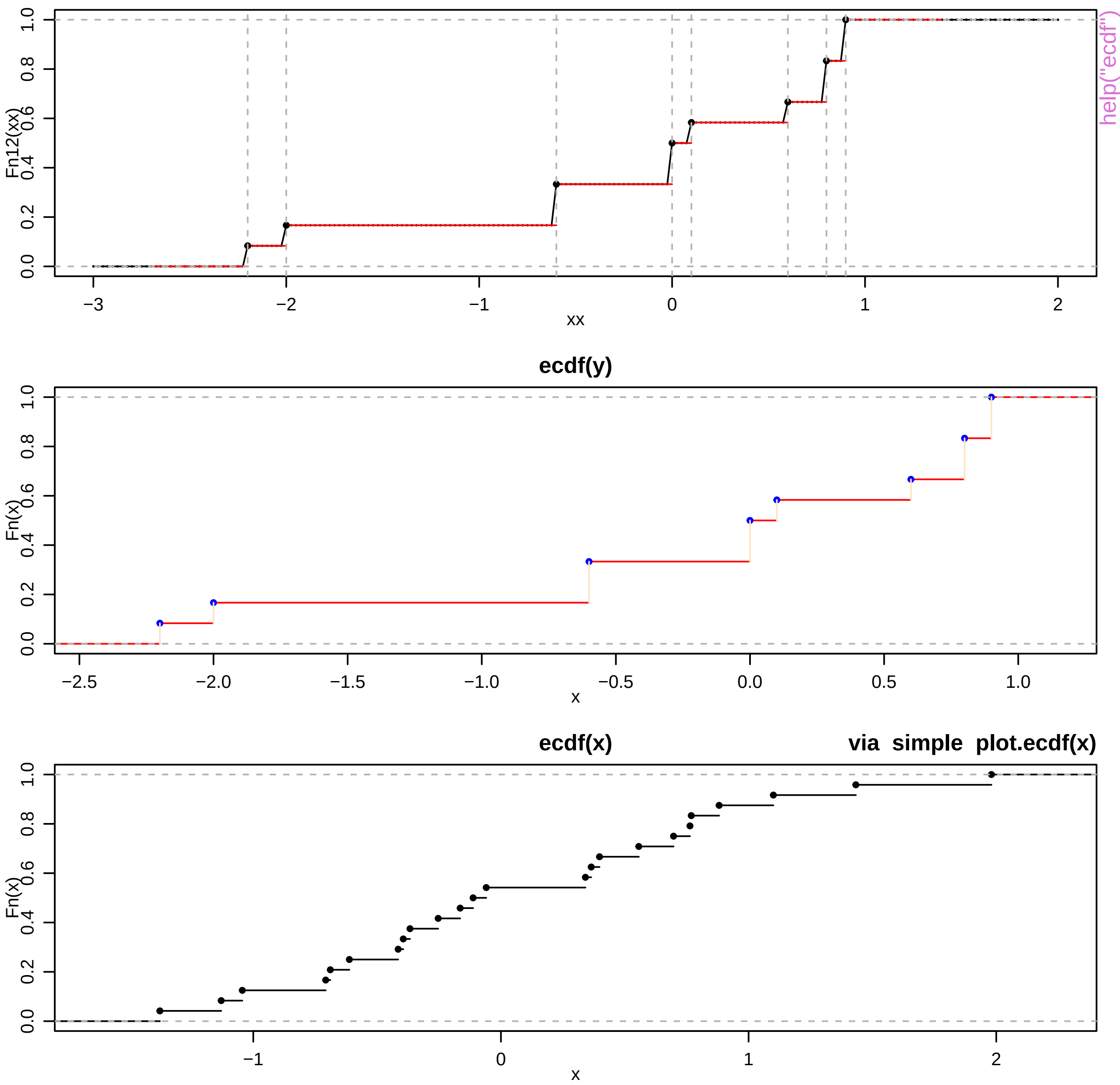

equivalent bandwidths, 7 different kernels
Density
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0
20
40
60
80
N = 70 Bandwidth = 3.932
gaussian
epanechnikov
rectangular
triangular
biweight
cosine
optcosine
help("density")

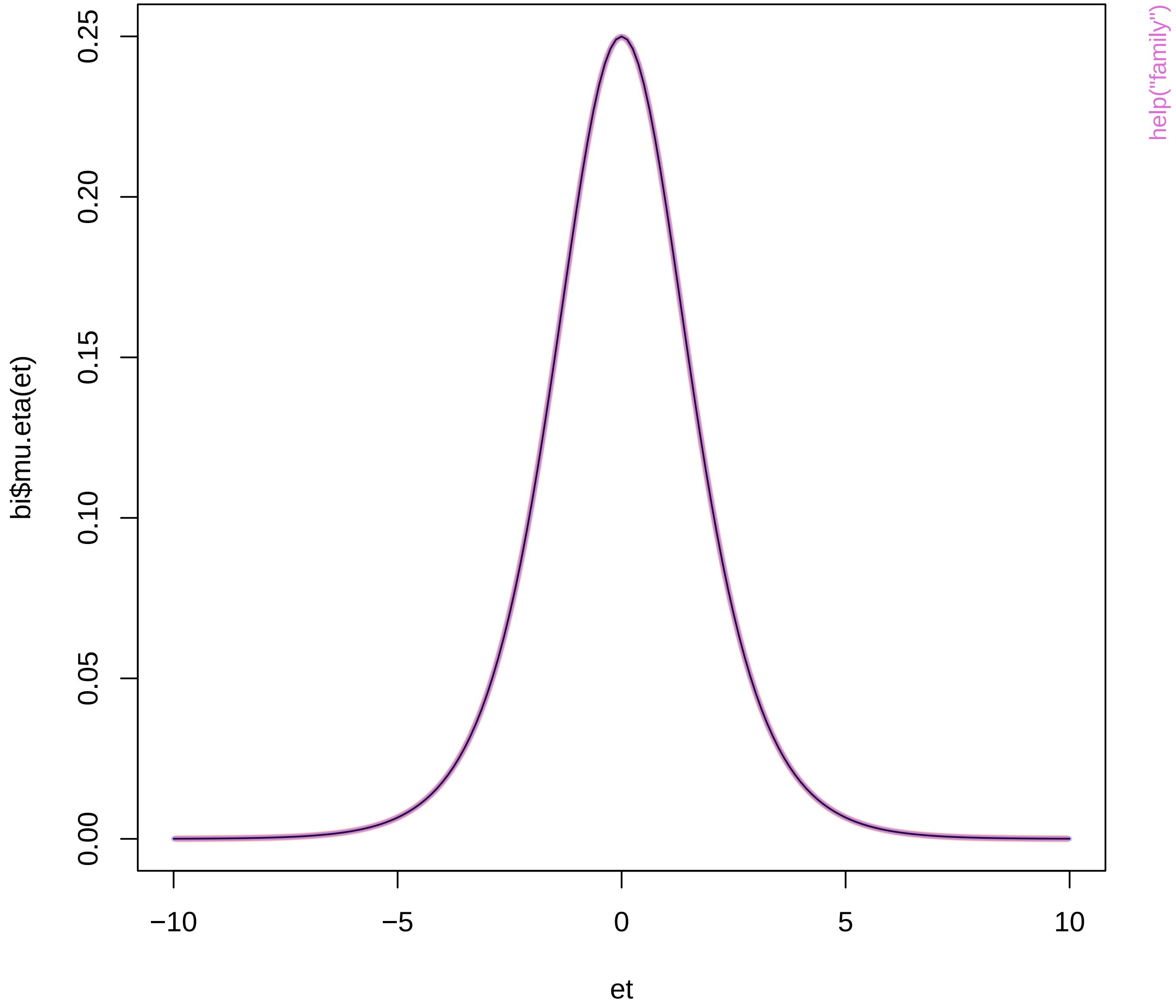


Cluster Dendrogram
Height
0.0
0.1
0.2
0.3
0.4
0.5
0.6
CONT
INTG
DMNR
PHYS
DILG
CFMG
DECI
RTEN
ORAL
WRIT
PREP
FAMI
dd
hclust (*, "complete")
help("dist")

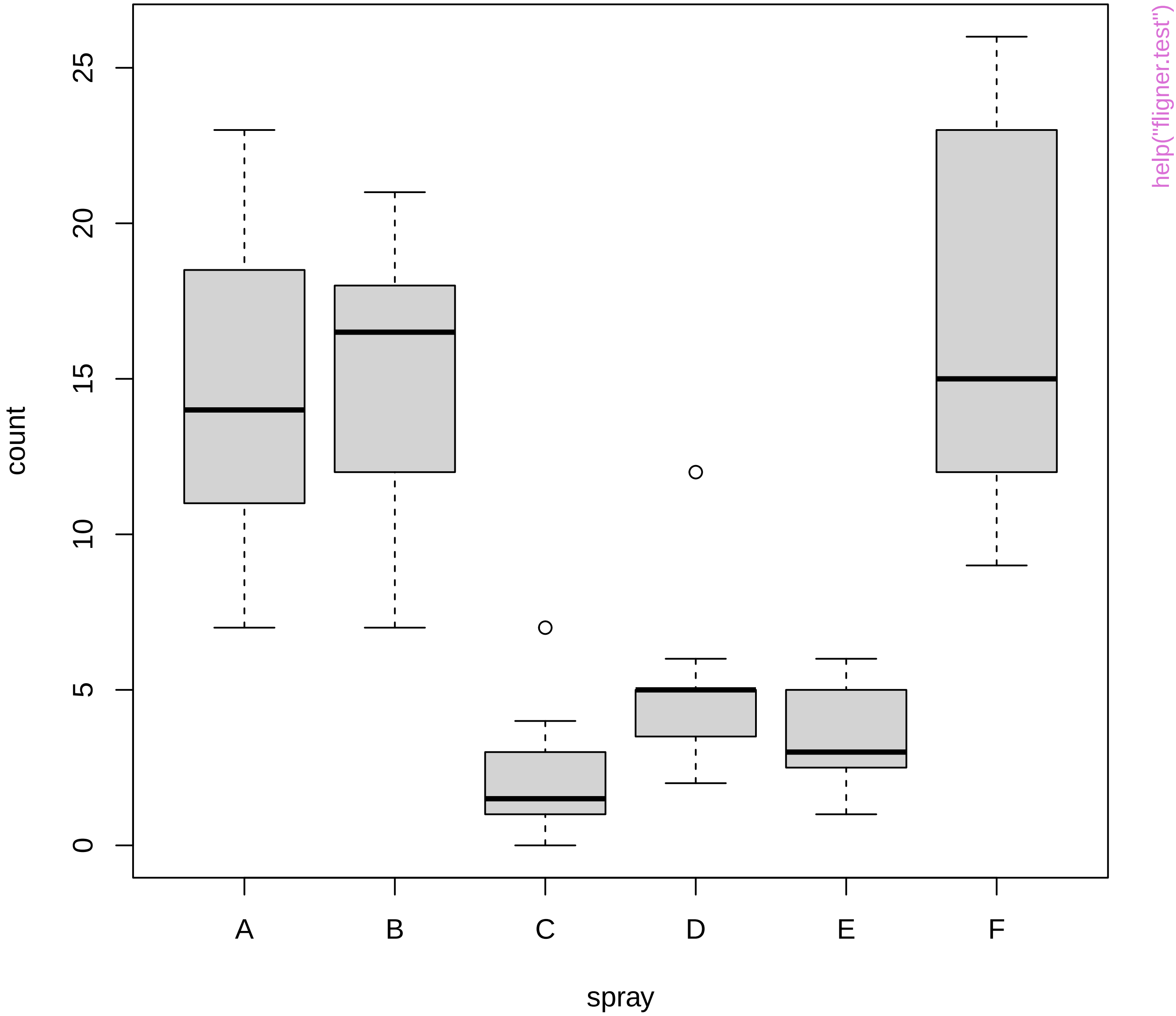


ecdf(rnorm(10))
Fn(x)
x
help("ecdf")
ecdf(rnorm(10))
Fn(x)
x
ecdf(y)
lwd = 2
Fn(x)
x

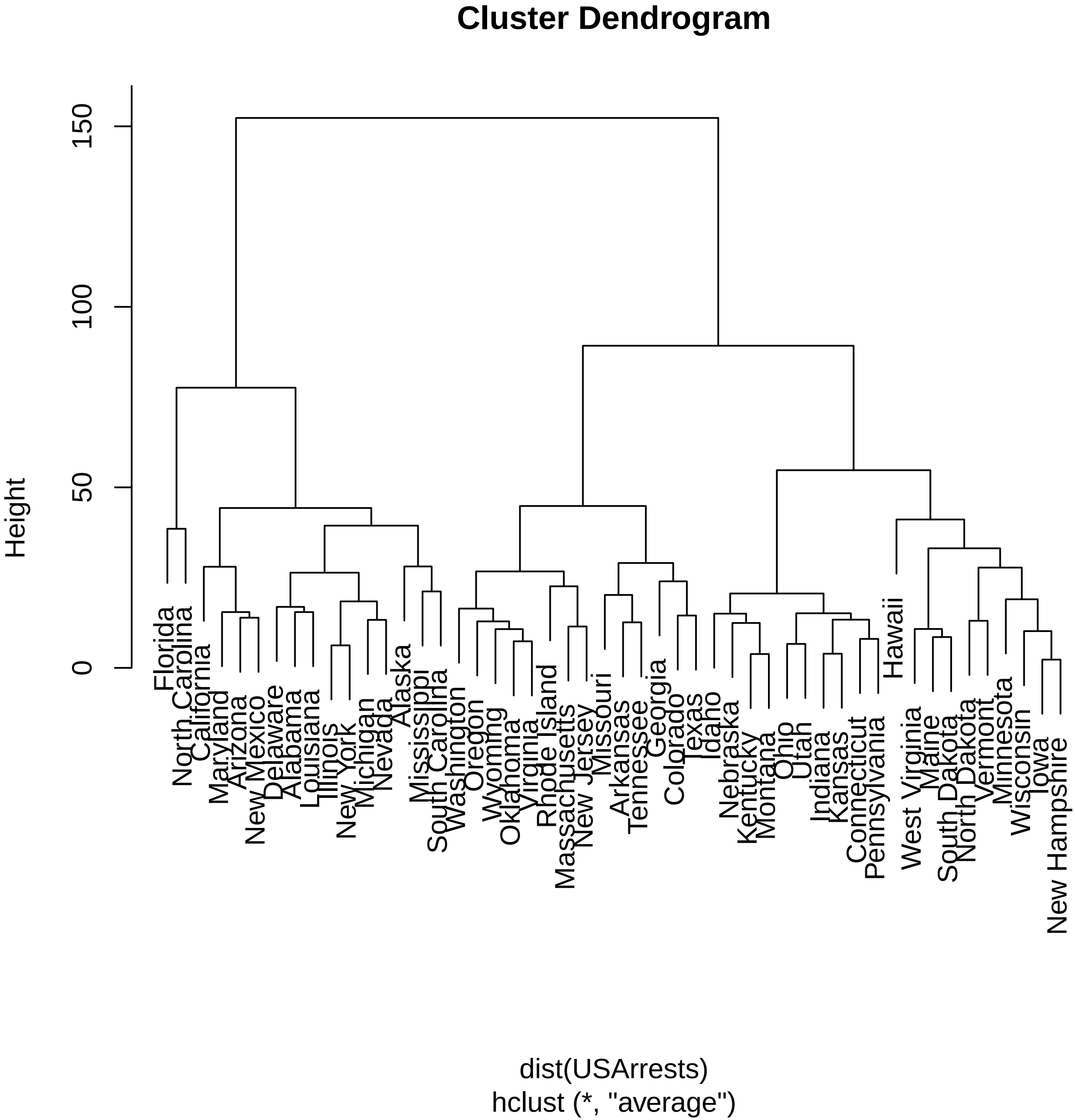

help("ecdf")
Fn12(xx)
xx
ecdf(y)
Fn(x)
x
ecdf(x)
via simple plot.ecdf(x)
Fn(x)
x



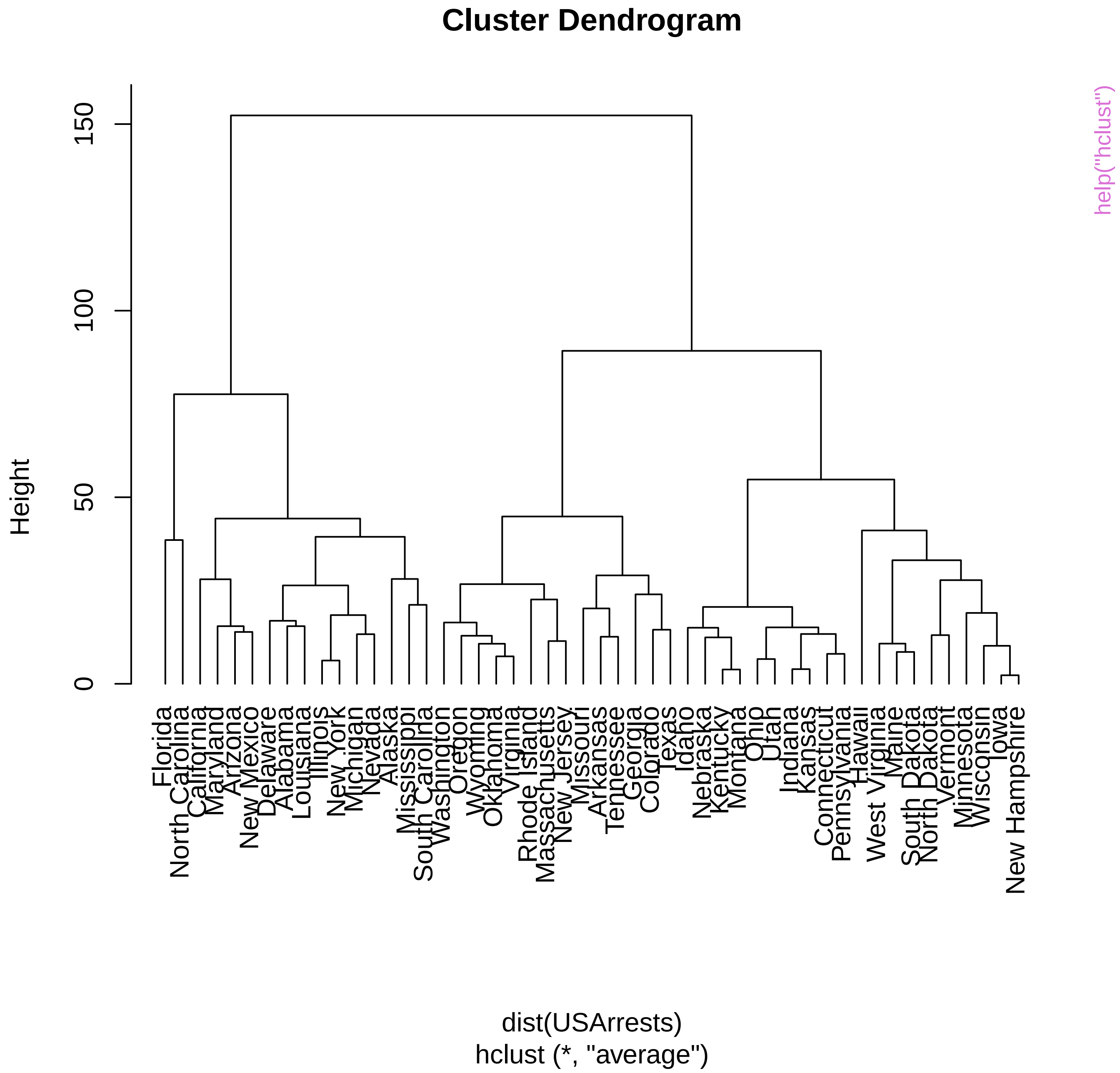
0.00
0.05
0.10
0.15
0.20
0.25
bi$mu.eta(et)
−10
−5
0
5
10
et
help("family")



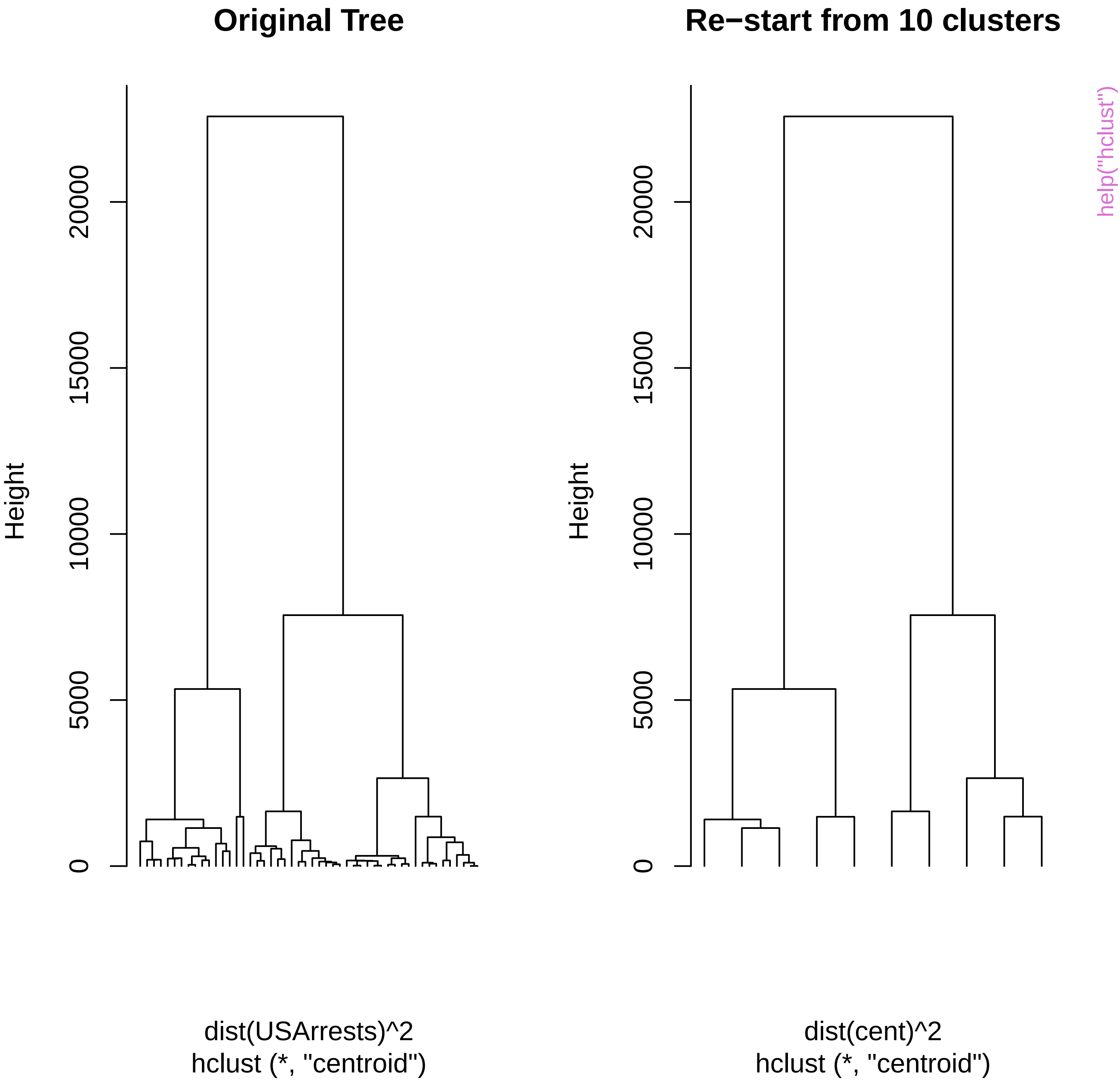
count
0
5
10
15
20
25
A
B
C
D
E
F
spray
help("fligner.test")


Cluster Dendrogram
Height
0
50
100
150
Florida
North Carolina
California
Maryland
Arizona
New Mexico
Delaware
Alabama
Louisiana
Illinois
New York
Michigan
Nevada
Alaska
Mississippi
South Carolina
Washington
Oregon
Wyoming
Oklahoma
Virginia
Rhode Island
Massachusetts
New Jersey
Missouri
Arkansas
Tennessee
Georgia
Colorado
Texas
Idaho
Nebraska
Kentucky
Montana
Ohio
Utah
Indiana
Kansas
Connecticut
Pennsylvania
Hawaii
West Virginia
Maine
South Dakota
North Dakota
Vermont
Minnesota
Wisconsin
Iowa
New Hampshire
dist(USArrests)
hclust (*, "average")

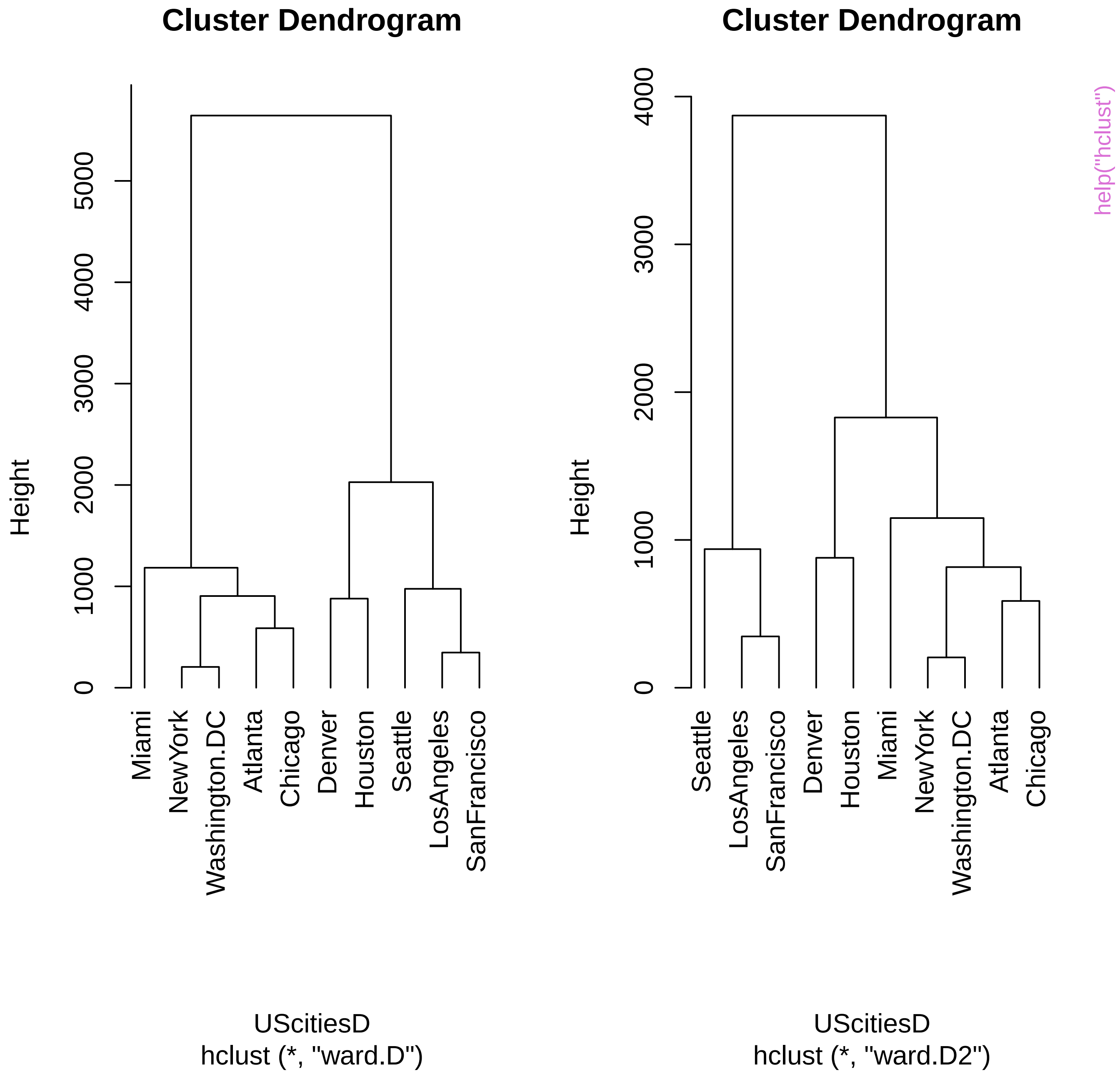
help("hclust")

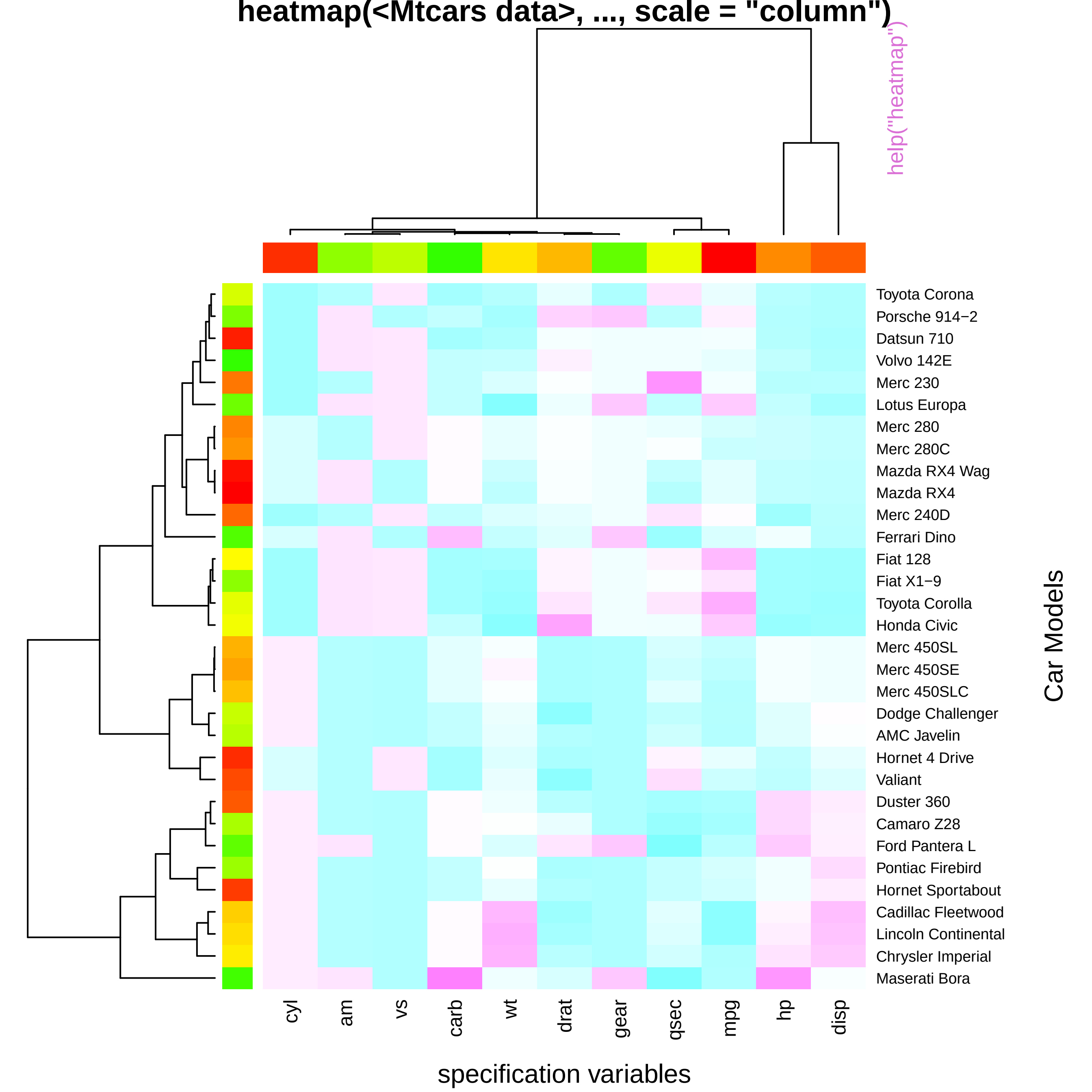


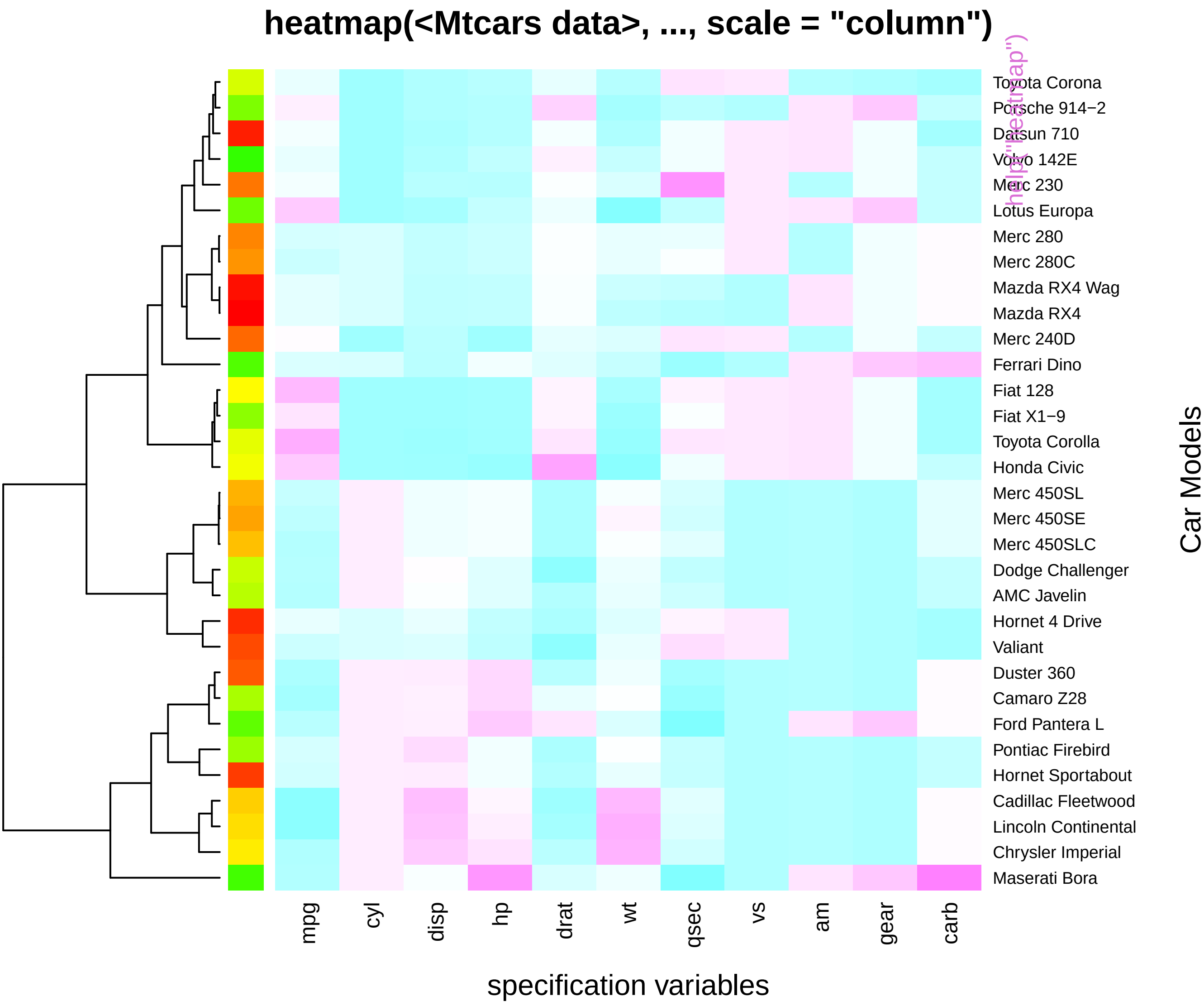
Cluster Dendrogram
Height
0
50
100
150
Florida
North Carolina
California
Maryland
Arizona
New Mexico
Delaware
Alabama
Louisiana
Illinois
New York
Michigan
Nevada
Alaska
Mississippi
South Carolina
Washington
Oregon
Wyoming
Oklahoma
Virginia
Rhode Island
Massachusetts
New Jersey
Missouri
Arkansas
Tennessee
Georgia
Colorado
Texas
Idaho
Nebraska
Kentucky
Montana
Ohio
Utah
Indiana
Kansas
Connecticut
Pennsylvania
Hawaii
West Virginia
Maine
South Dakota
North Dakota
Vermont
Minnesota
Wisconsin
Iowa
New Hampshire
dist(USArrests)
hclust (*, "average")
help("hclust")

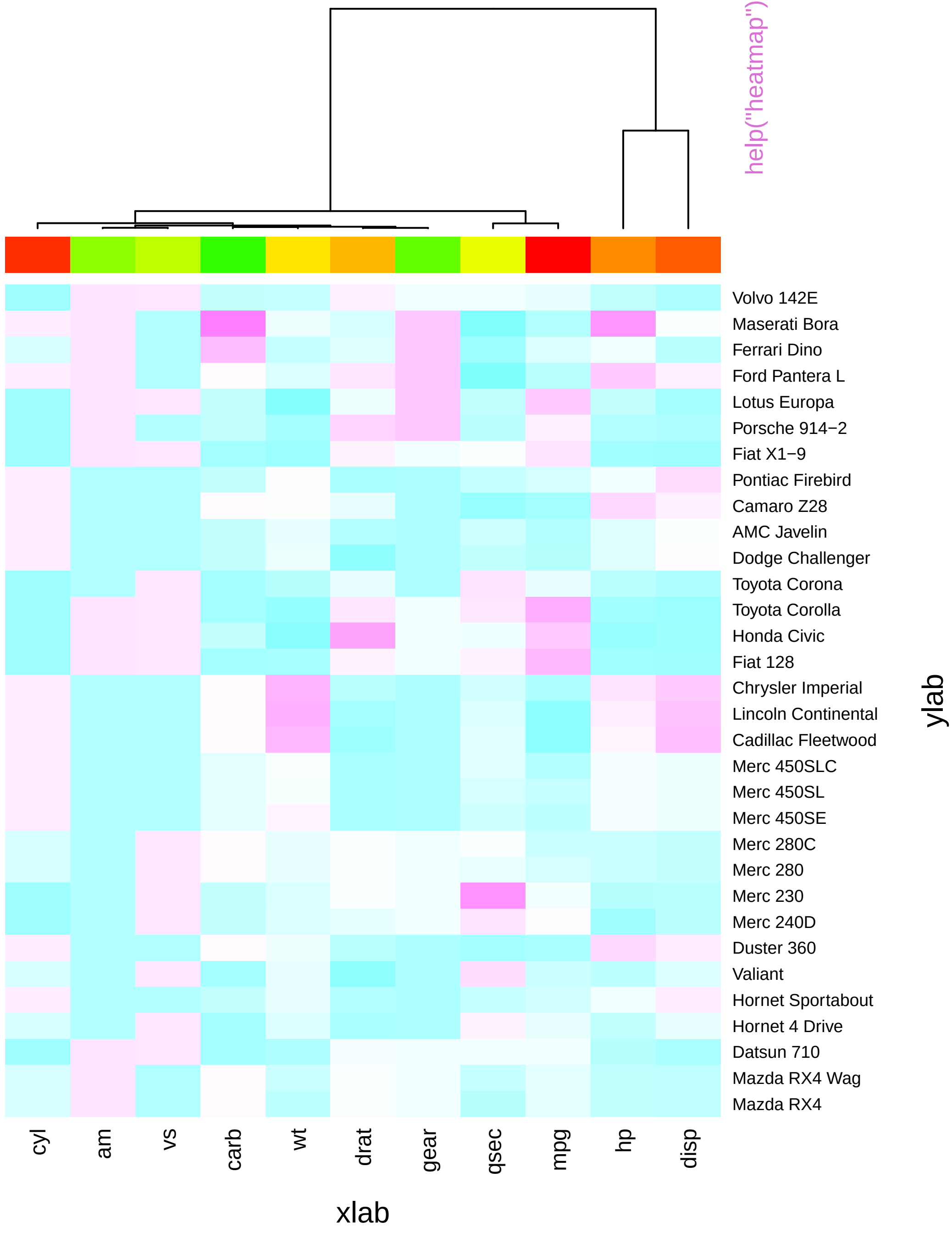

Original Tree
Re−start from 10 clusters
Height
Height
0
5000
10000
15000
20000
0
5000
10000
15000
20000
dist(USArrests)^2
hclust (*, "centroid")
dist(cent)^2
hclust (*, "centroid")
help("hclust")


Seattle
NewYork
Chicago
Washington.DC
Denver
SanFrancisco
Atlanta
LosAngeles
Houston
Miami
help("hclust")


Cluster Dendrogram
Height
0
1000
2000
3000
4000
5000
Miami
NewYork
Washington.DC
Atlanta
Chicago
Denver
Houston
Seattle
LosAngeles
SanFrancisco
UScitiesD
hclust (*, "ward.D")
Cluster Dendrogram
Height
0
1000
2000
3000
4000
Seattle
LosAngeles
SanFrancisco
Denver
Houston
Miami
NewYork
Washington.DC
Atlanta
Chicago
UScitiesD
hclust (*, "ward.D2")
help("hclust")

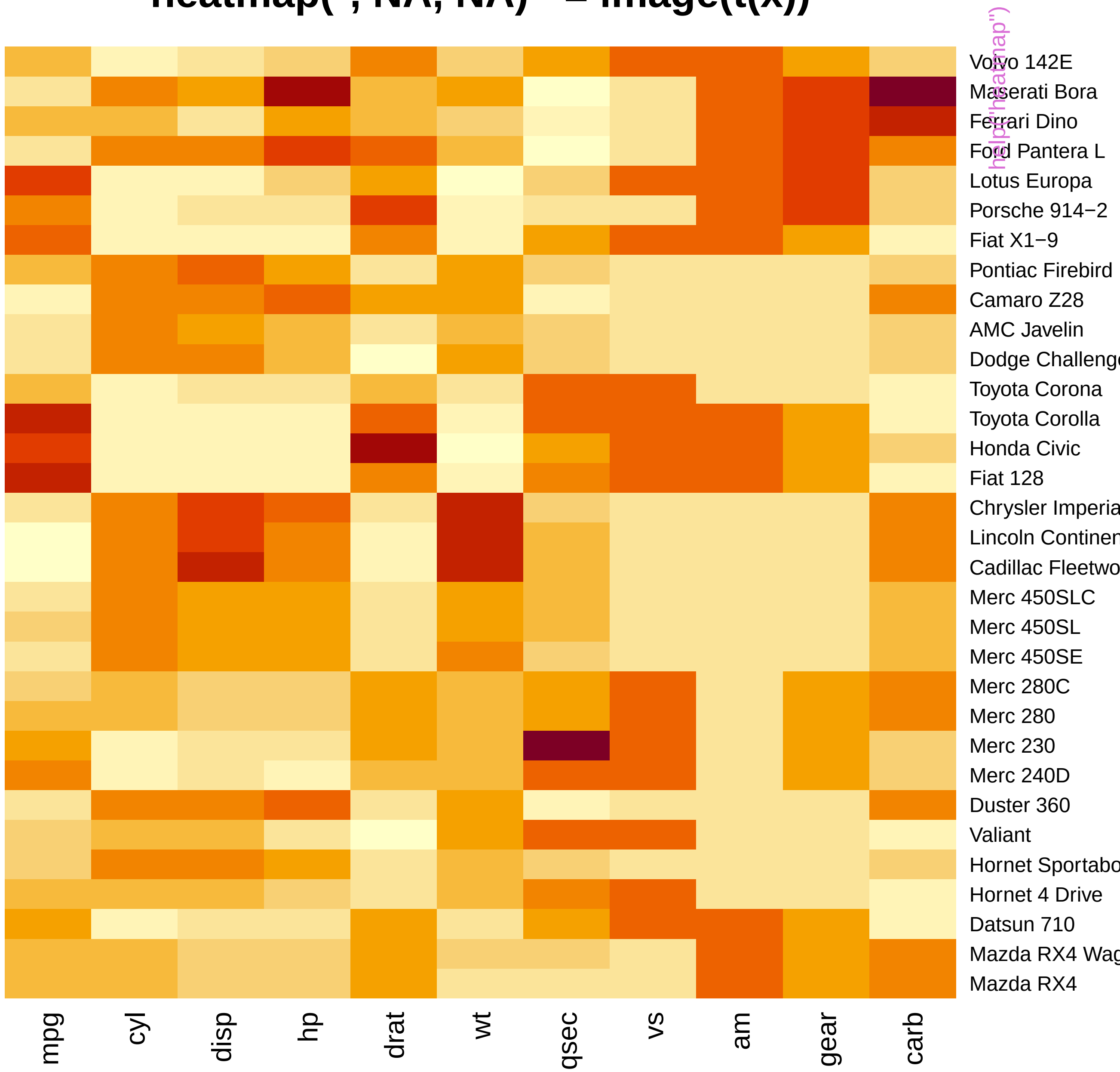

heatmap(<Mtcars data>, ..., scale = "column")
help("heatmap")
Toyota Corona
Porsche 914−2
Datsun 710
Volvo 142E
Merc 230
Lotus Europa
Merc 280
Merc 280C
Mazda RX4 Wag
Mazda RX4
Merc 240D
Ferrari Dino
Fiat 128
Fiat X1−9
Toyota Corolla
Honda Civic
Merc 450SL
Merc 450SE
Merc 450SLC
Dodge Challenger
AMC Javelin
Hornet 4 Drive
Valiant
Duster 360
Camaro Z28
Ford Pantera L
Pontiac Firebird
Hornet Sportabout
Cadillac Fleetwood
Lincoln Continental
Chrysler Imperial
Maserati Bora
Car Models
cyl
am
vs
carb
wt
drat
gear
qsec
mpg
hp
disp
specification variables

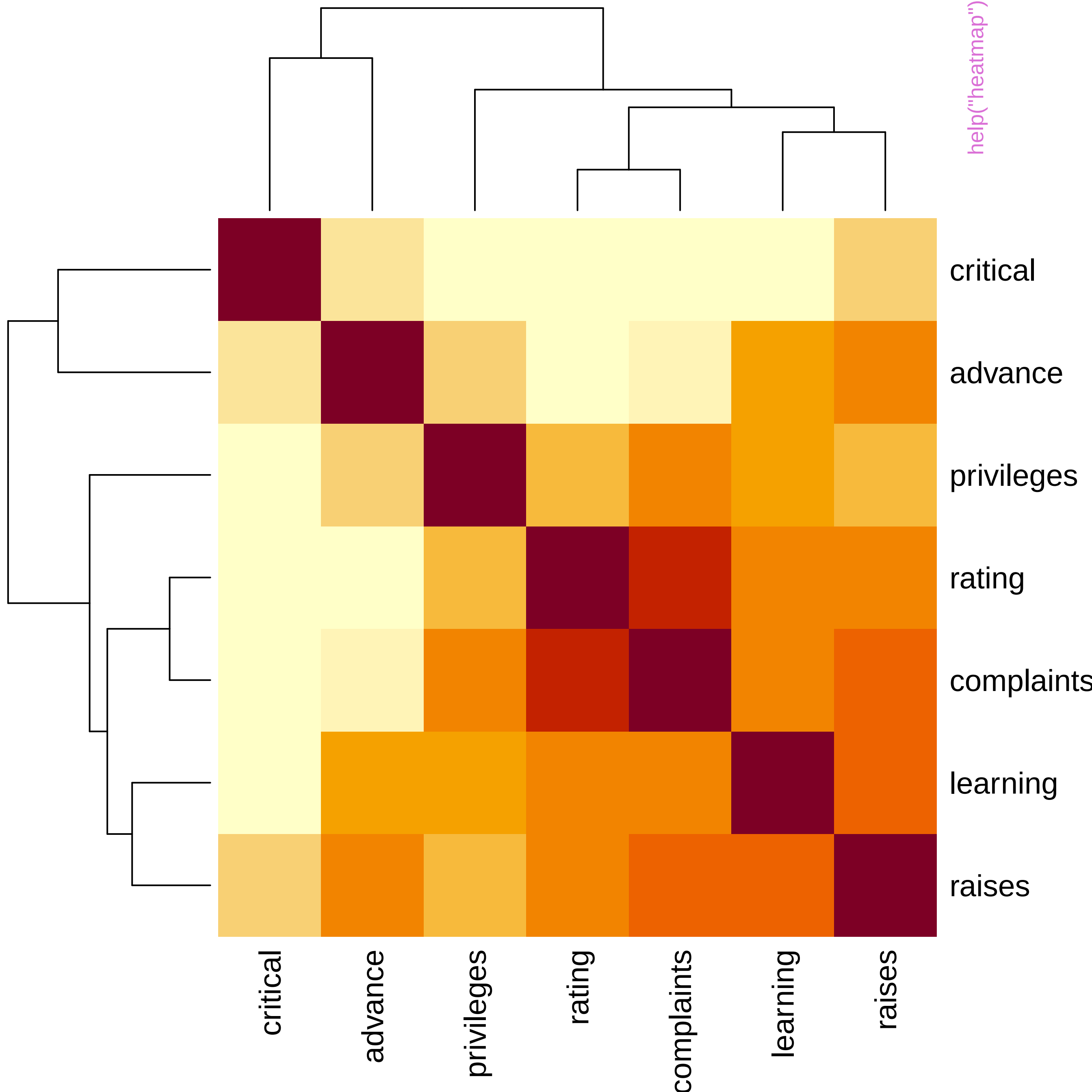

heatmap(<Mtcars data>, ..., scale = "column")
help("heatmap")
Toyota Corona
Porsche 914−2
Datsun 710
Volvo 142E
Merc 230
Lotus Europa
Merc 280
Merc 280C
Mazda RX4 Wag
Mazda RX4
Merc 240D
Ferrari Dino
Fiat 128
Fiat X1−9
Toyota Corolla
Honda Civic
Merc 450SL
Merc 450SE
Merc 450SLC
Dodge Challenger
AMC Javelin
Hornet 4 Drive
Valiant
Duster 360
Camaro Z28
Ford Pantera L
Pontiac Firebird
Hornet Sportabout
Cadillac Fleetwood
Lincoln Continental
Chrysler Imperial
Maserati Bora
Car Models
mpg
cyl
disp
hp
drat
wt
qsec
vs
am
gear
carb
specification variables

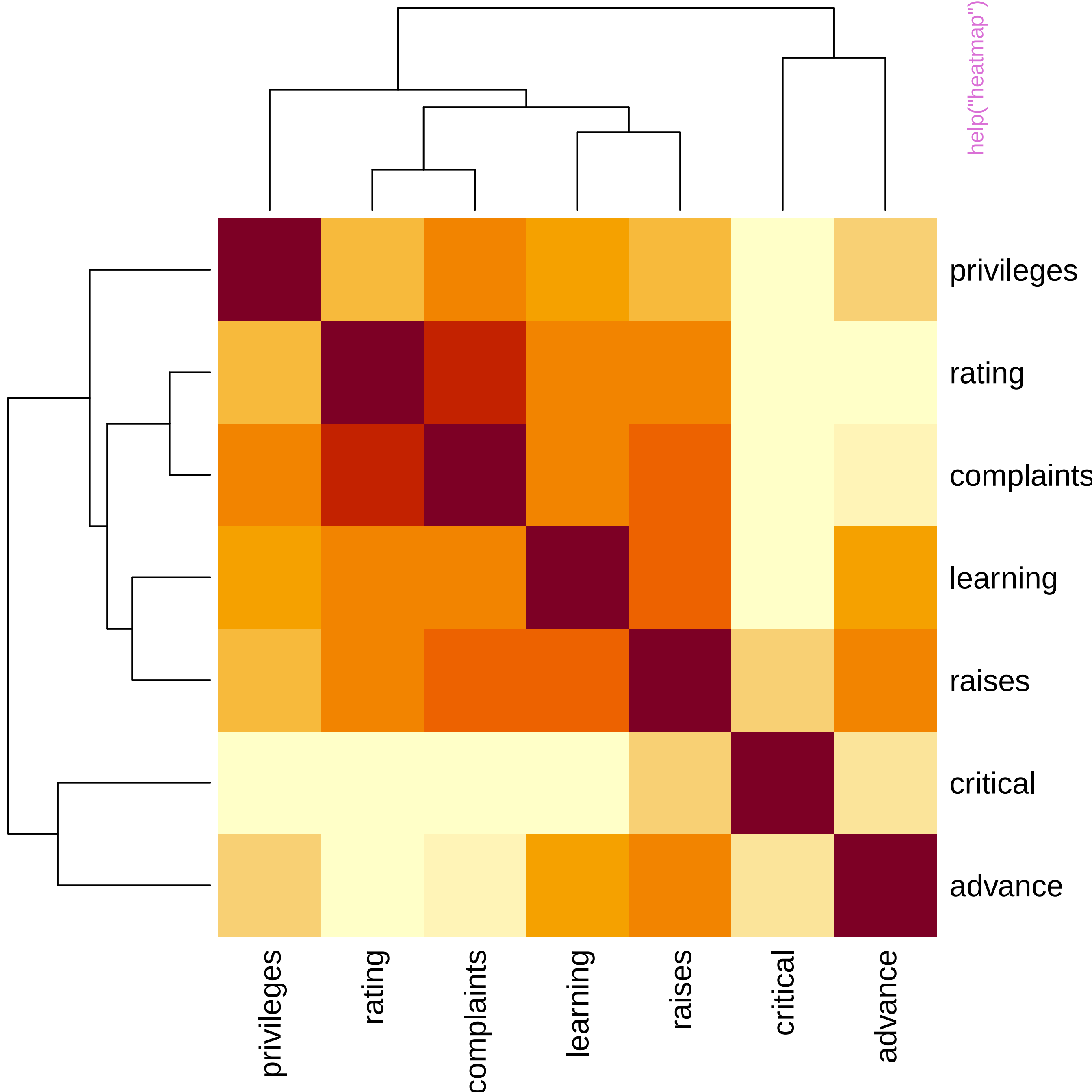


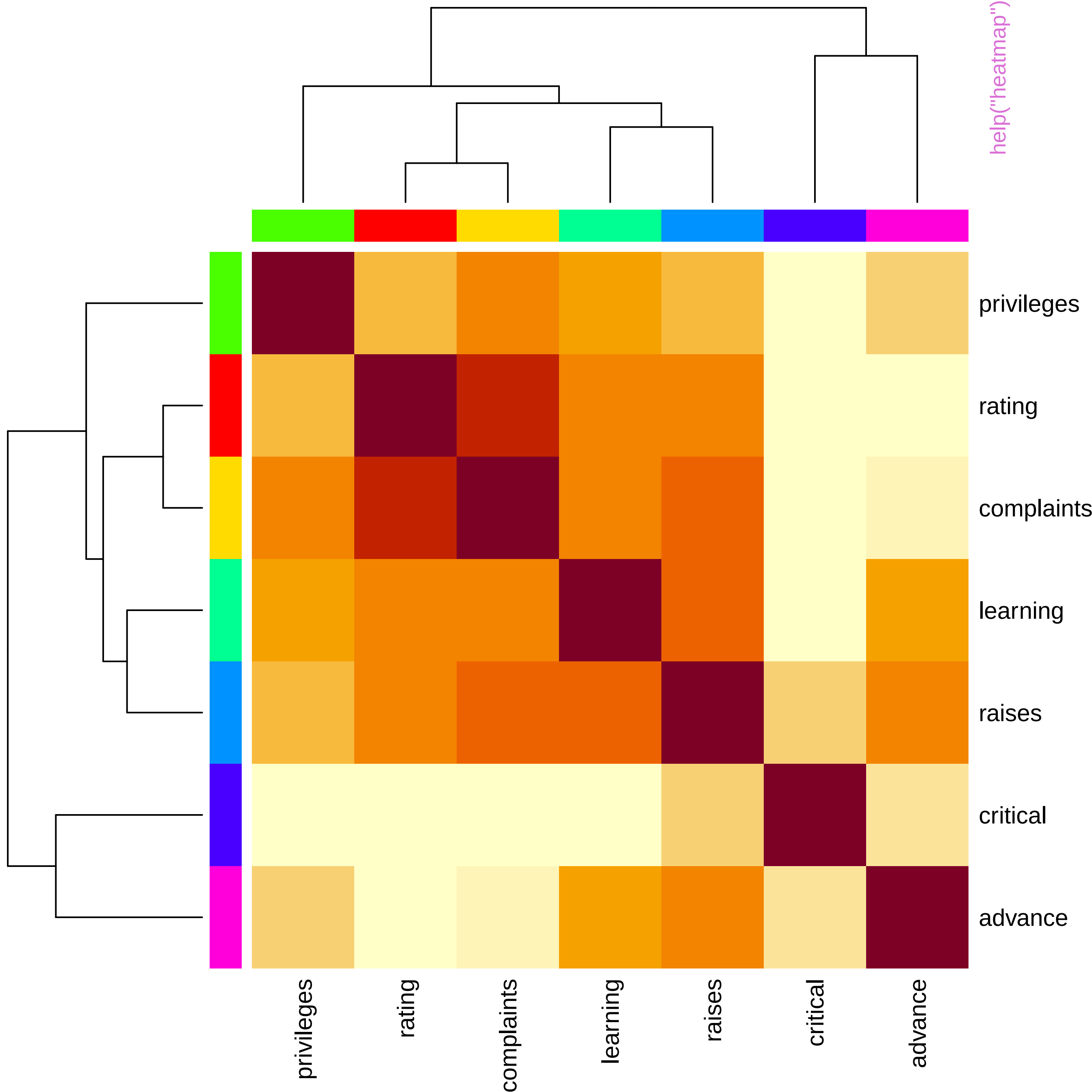
help("heatmap")
Volvo 142E
Maserati Bora
Ferrari Dino
Ford Pantera L
Lotus Europa
Porsche 914−2
Fiat X1−9
Pontiac Firebird
Camaro Z28
AMC Javelin
Dodge Challenger
Toyota Corona
Toyota Corolla
Honda Civic
Fiat 128
Chrysler Imperial
Lincoln Continental
Cadillac Fleetwood
Merc 450SLC
Merc 450SL
Merc 450SE
Merc 280C
Merc 280
Merc 230
Merc 240D
Duster 360
Valiant
Hornet Sportabout
Hornet 4 Drive
Datsun 710
Mazda RX4 Wag
Mazda RX4
ylab
cyl
am
vs
carb
wt
drat
gear
qsec
mpg
hp
disp
xlab

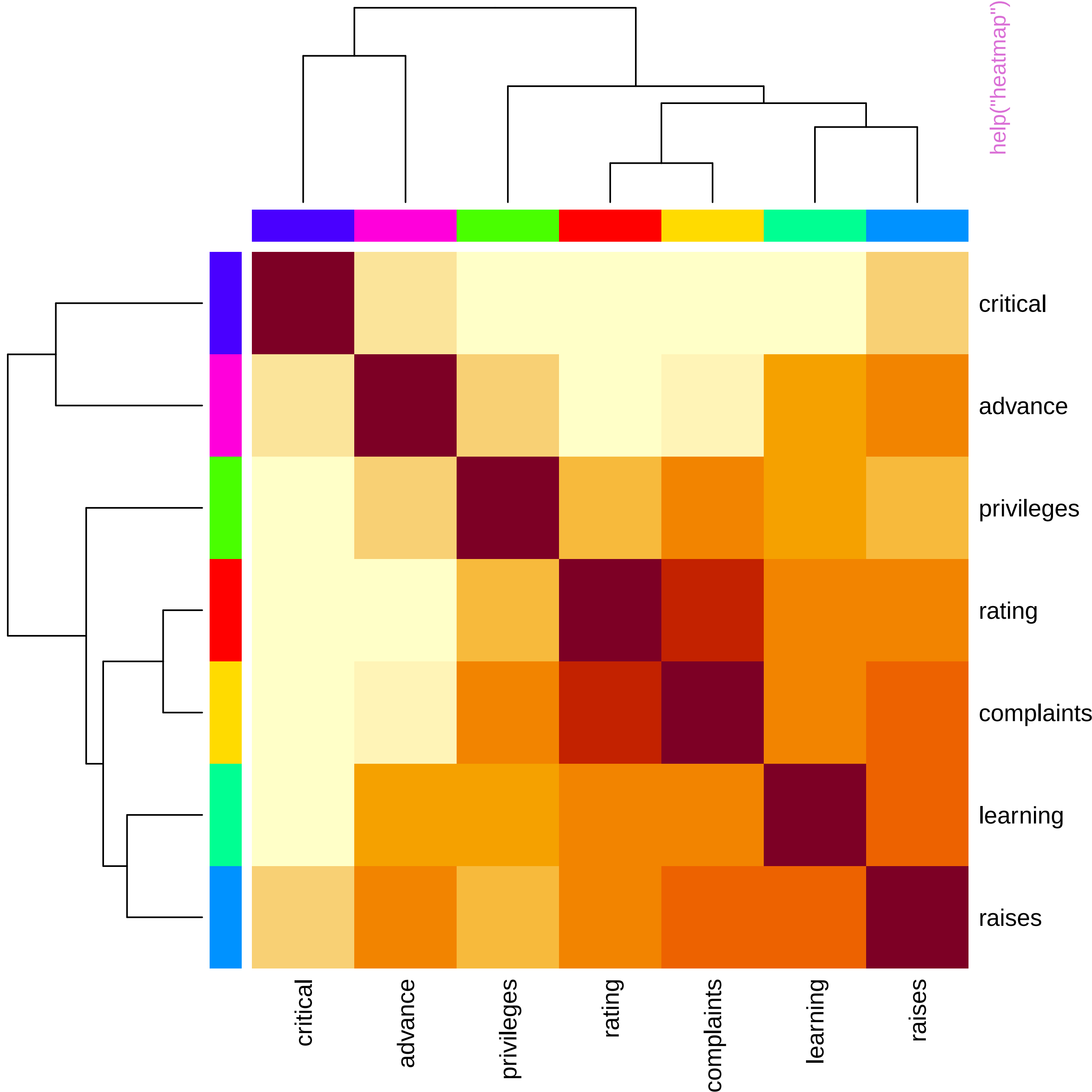
heatmap(*, NA, NA) ~= image(t(x))

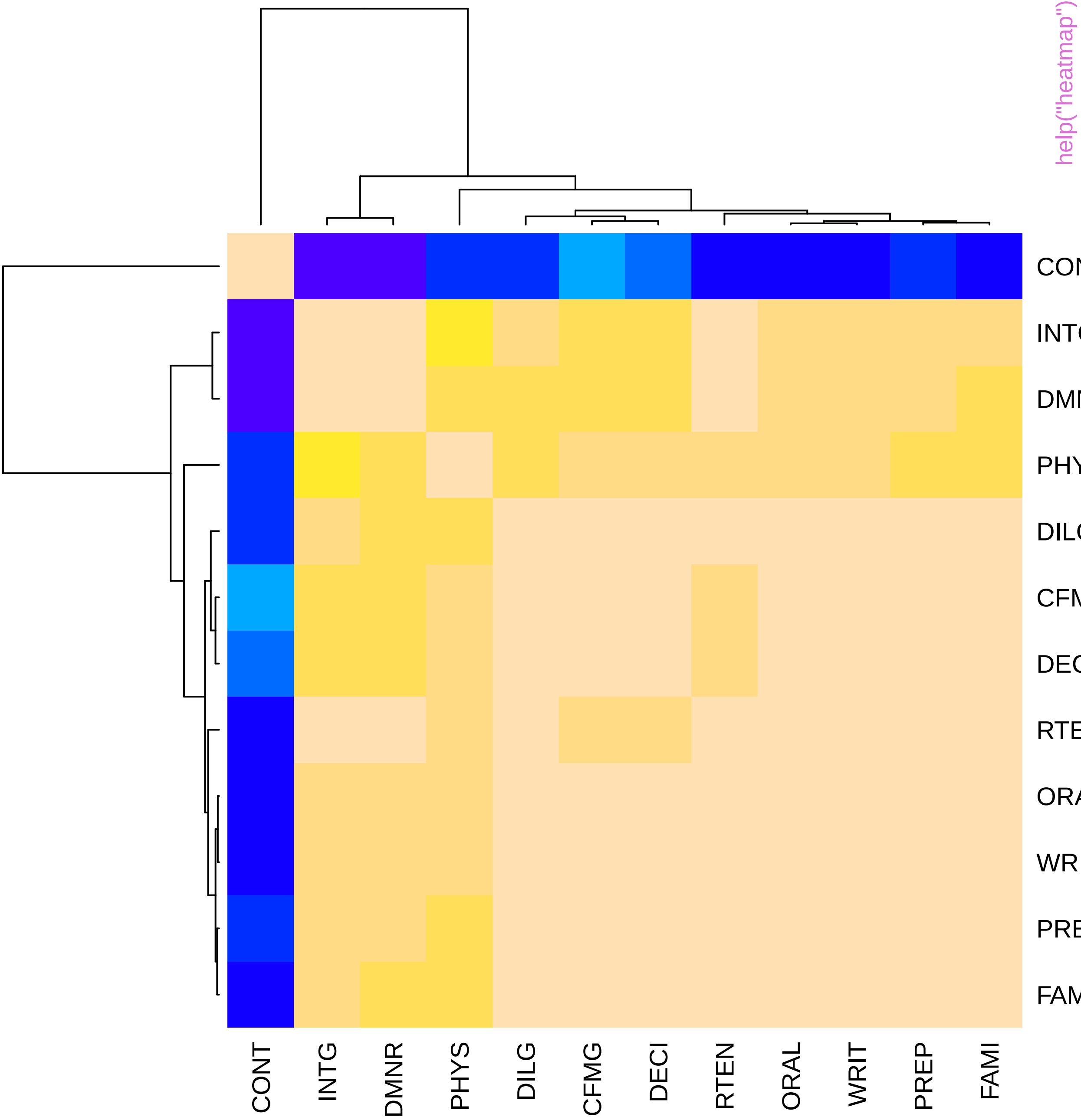


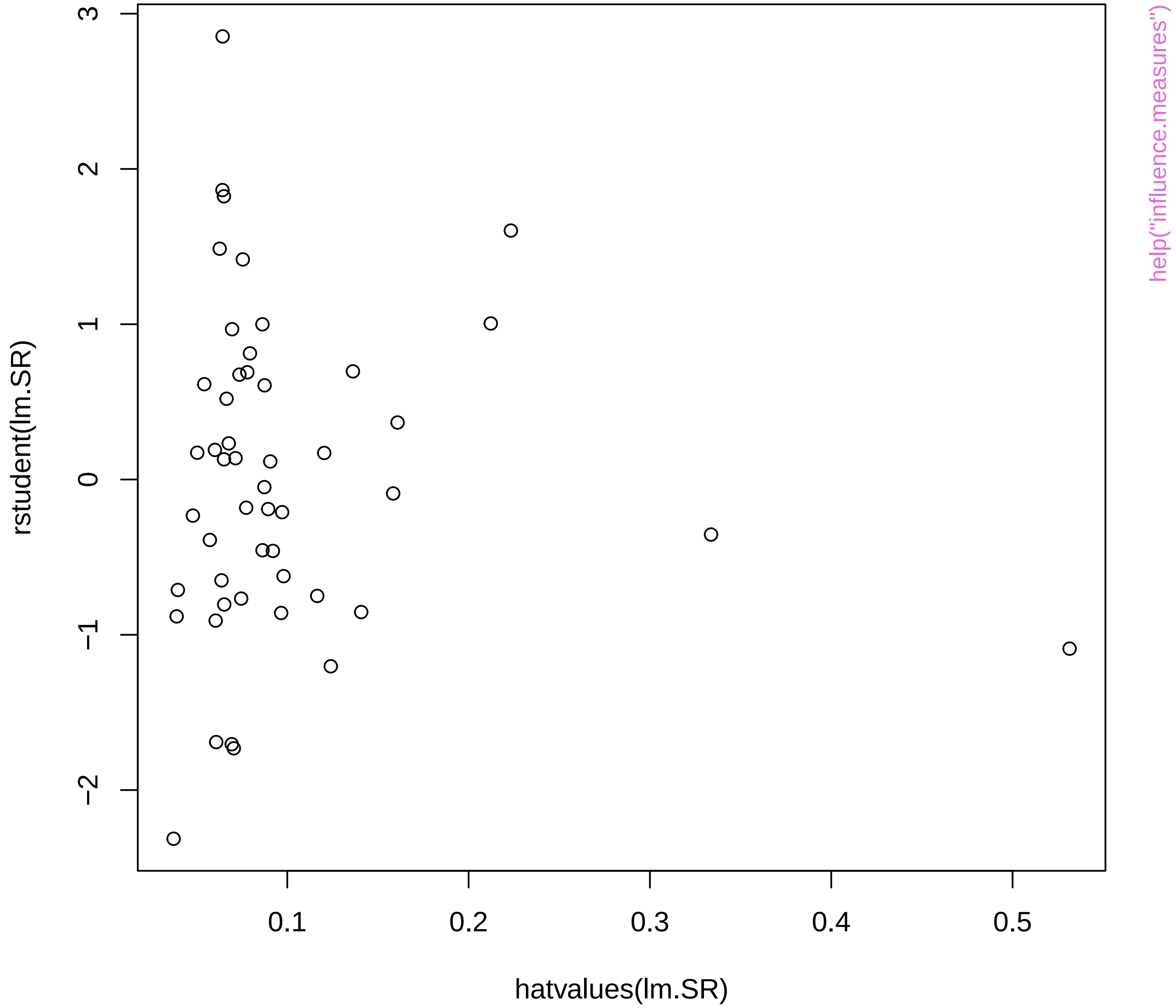

help("heatmap")
critical
advance
privileges
rating
complaints
learning
raises


help("heatmap")
privileges
rating
complaints
learning
raises
critical
advance
privileges
rating
complaints
learning
raises
critical
advance

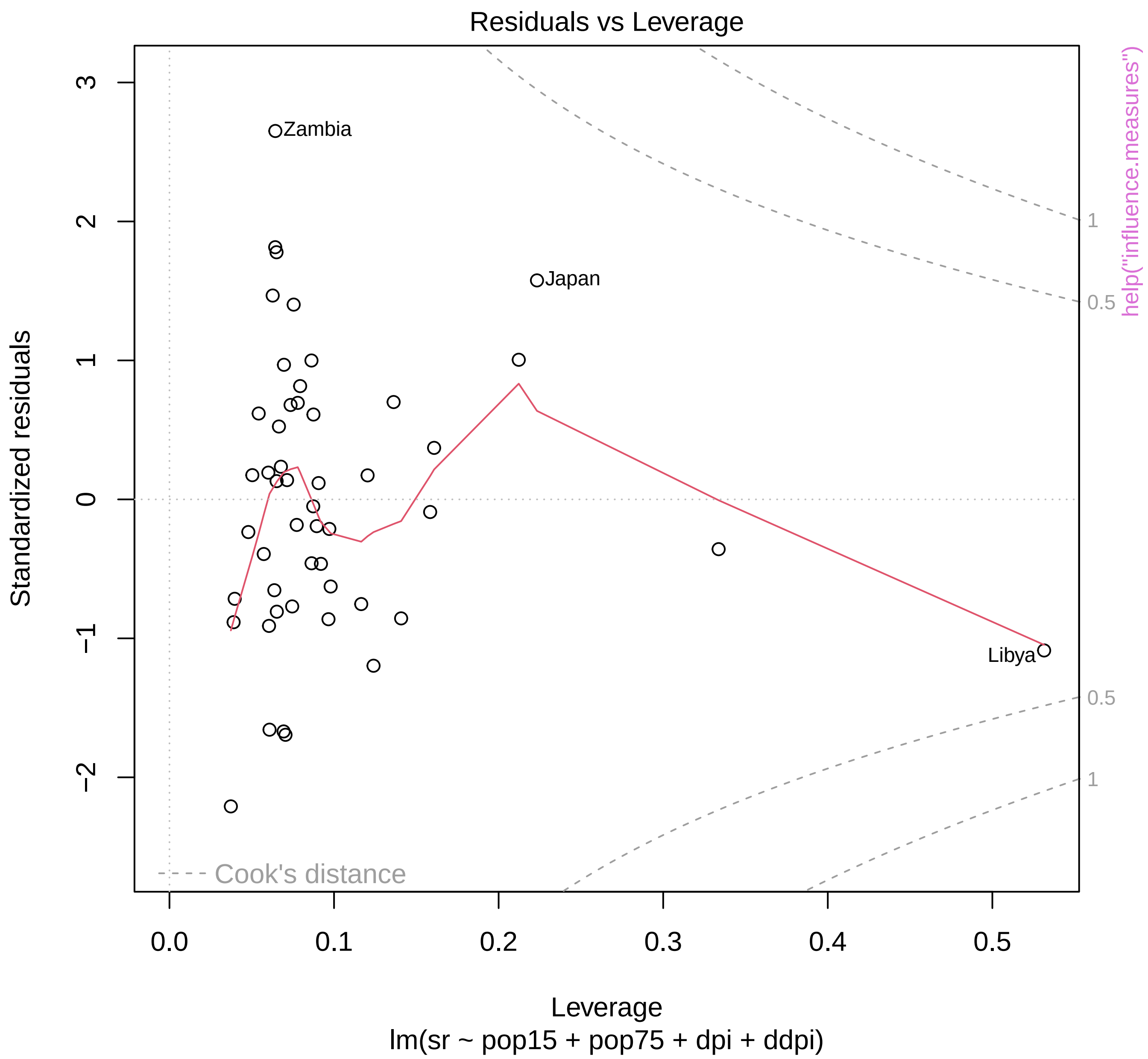

help("heatmap")
privileges
rating
complaints
learning
raises
critical
advance
privileges
rating
complaints
learning
raises
critical
advance

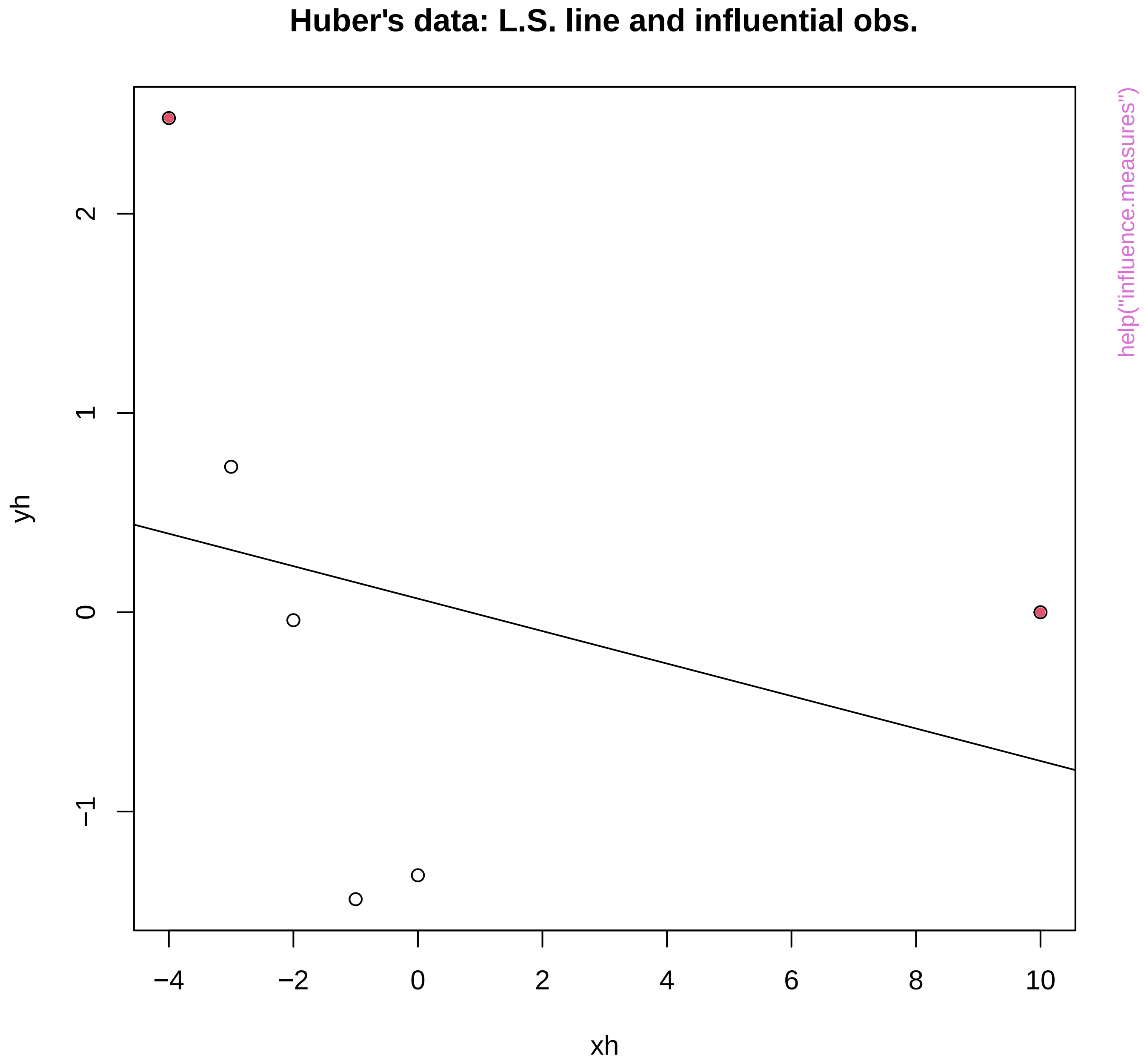

help("heatmap")
critical
advance
privileges
rating
complaints
learning
raises
critical
advance
privileges
rating
complaints
learning
raises

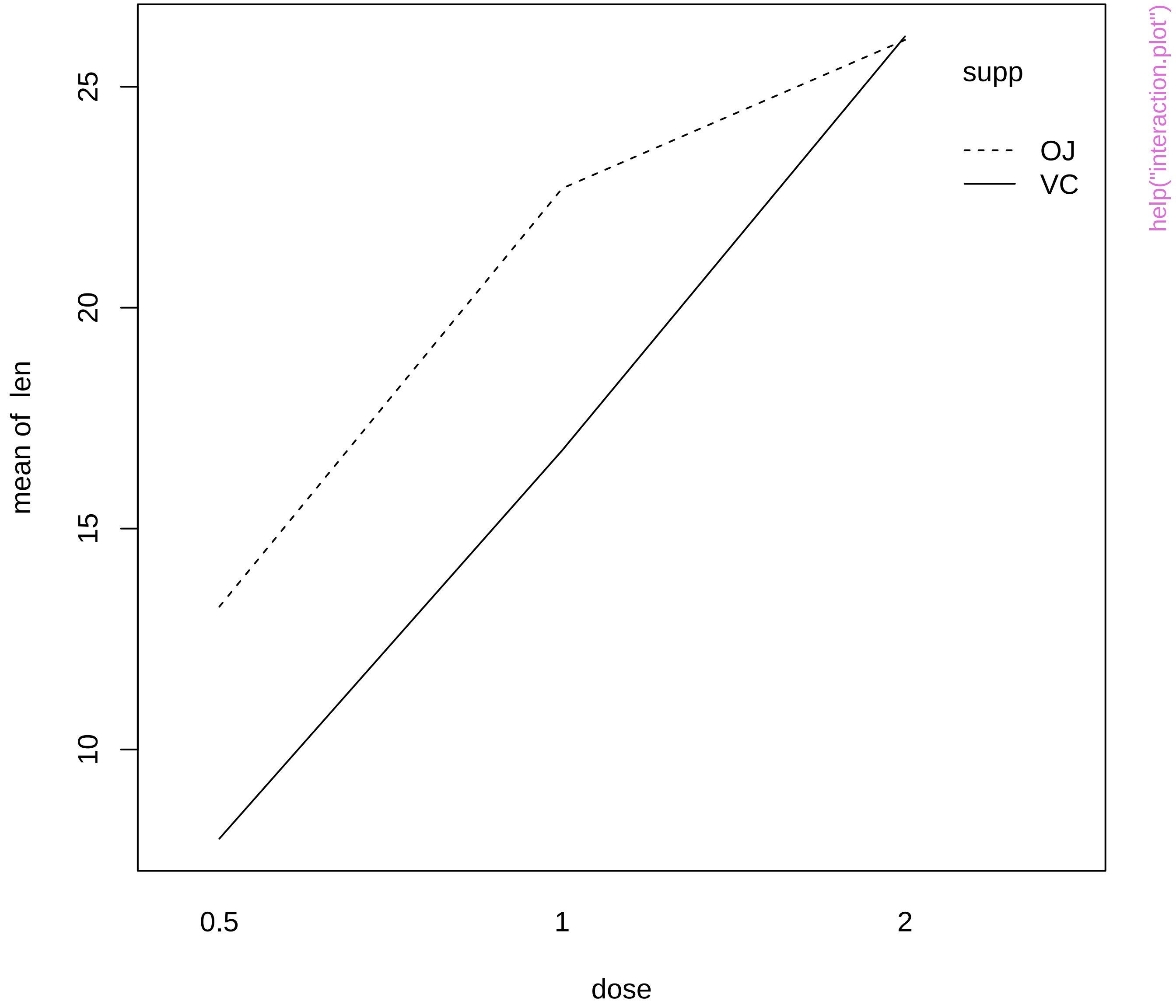

help("heatmap")
CONT
INTG
DMNR
PHYS
DILG
CFMG
DECI
RTEN
ORAL
WRIT
PREP
FAMI
CONT
INTG
DMNR
PHYS
DILG
CFMG
DECI
RTEN
ORAL
WRIT
PREP
FAMI



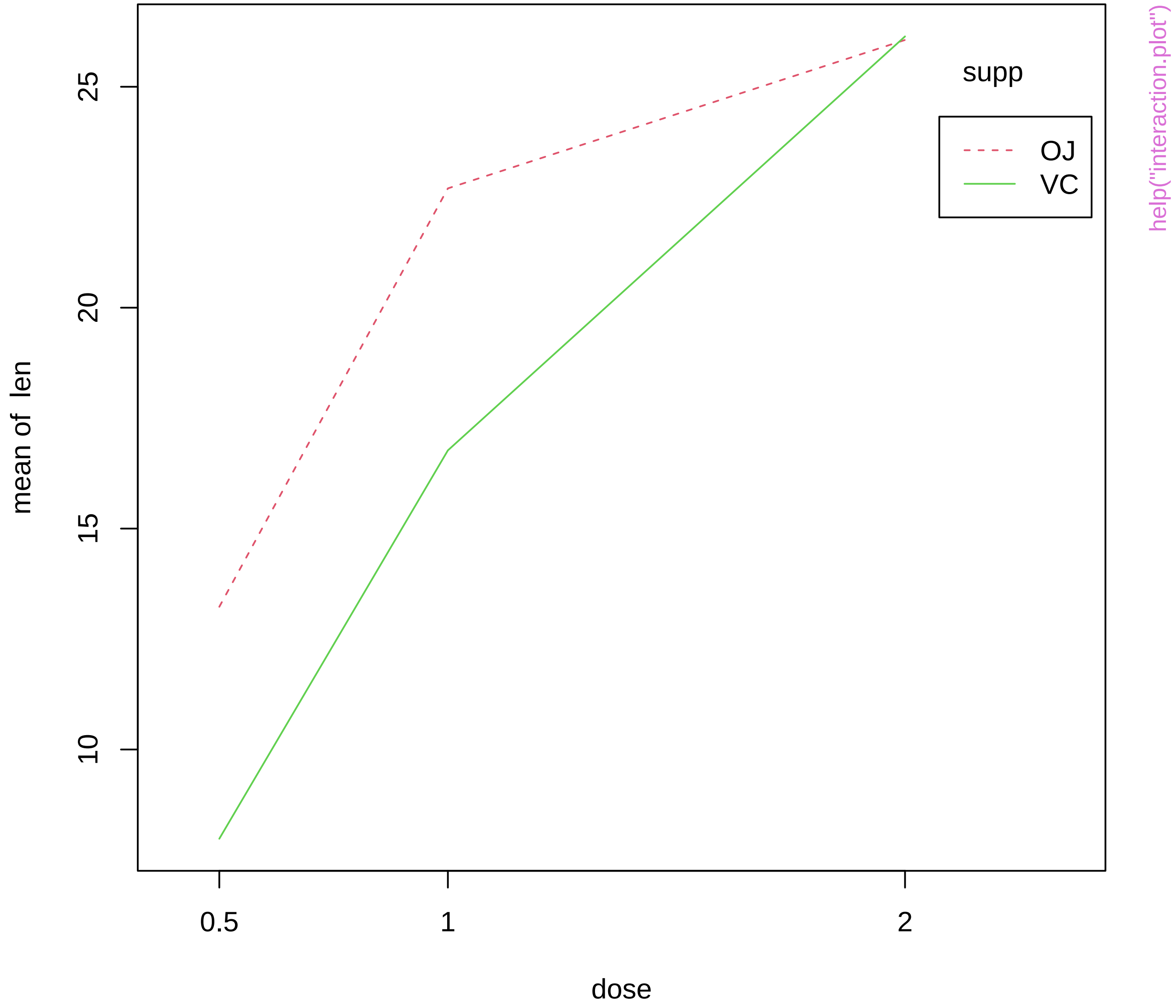
rstudent(lm.SR)
hatvalues(lm.SR)
3
2
1
0
−1
−2
0.1
0.2
0.3
0.4
0.5
help("influence.measures")


Residuals vs Leverage
Standardized residuals
Leverage
lm(sr ~ pop15 + pop75 + dpi + ddpi)
Zambia
Japan
Libya
Cook's distance
help("influence.measures")
0.0
0.1
0.2
0.3
0.4
0.5
3
2
1
0
−1
−2
1
0.5
0.5
1

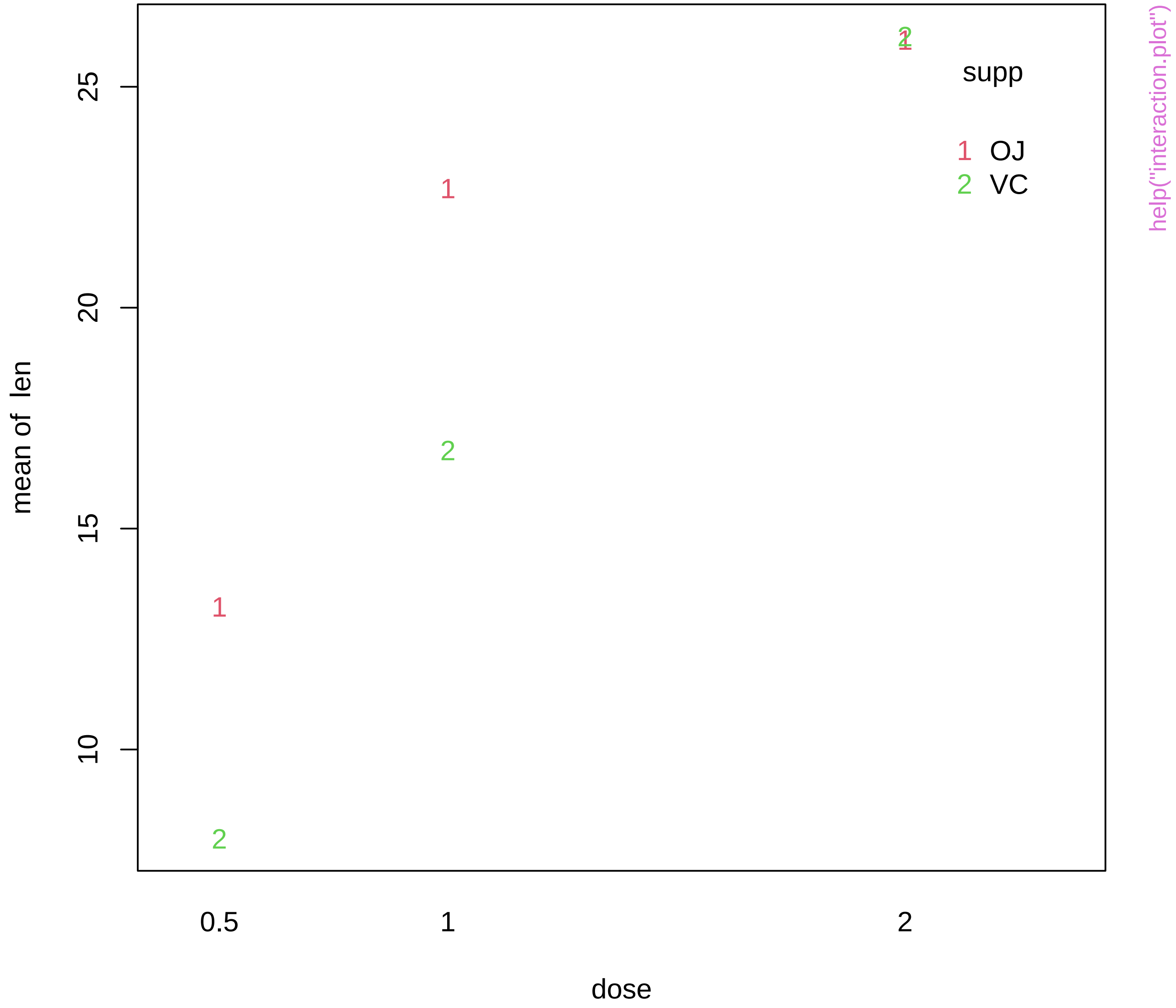

Huber's data: L.S. line and influential obs.
yh
xh
−4
−2
0
2
4
6
8
10
−1
0
1
2
help("influence.measures")



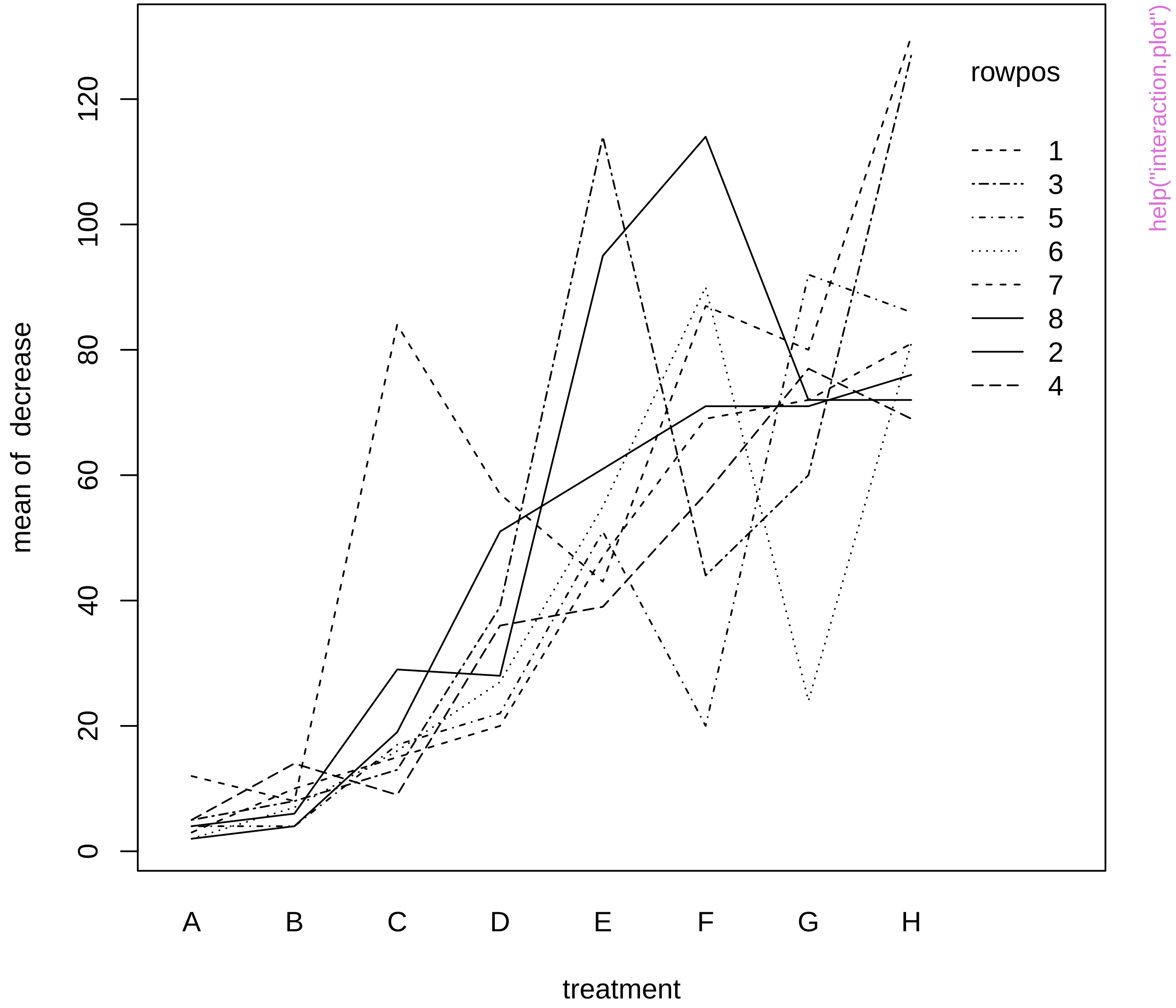
supp
OJ
VC
25
20
15
10
mean of len
0.5
1
2
dose
help("interaction.plot")



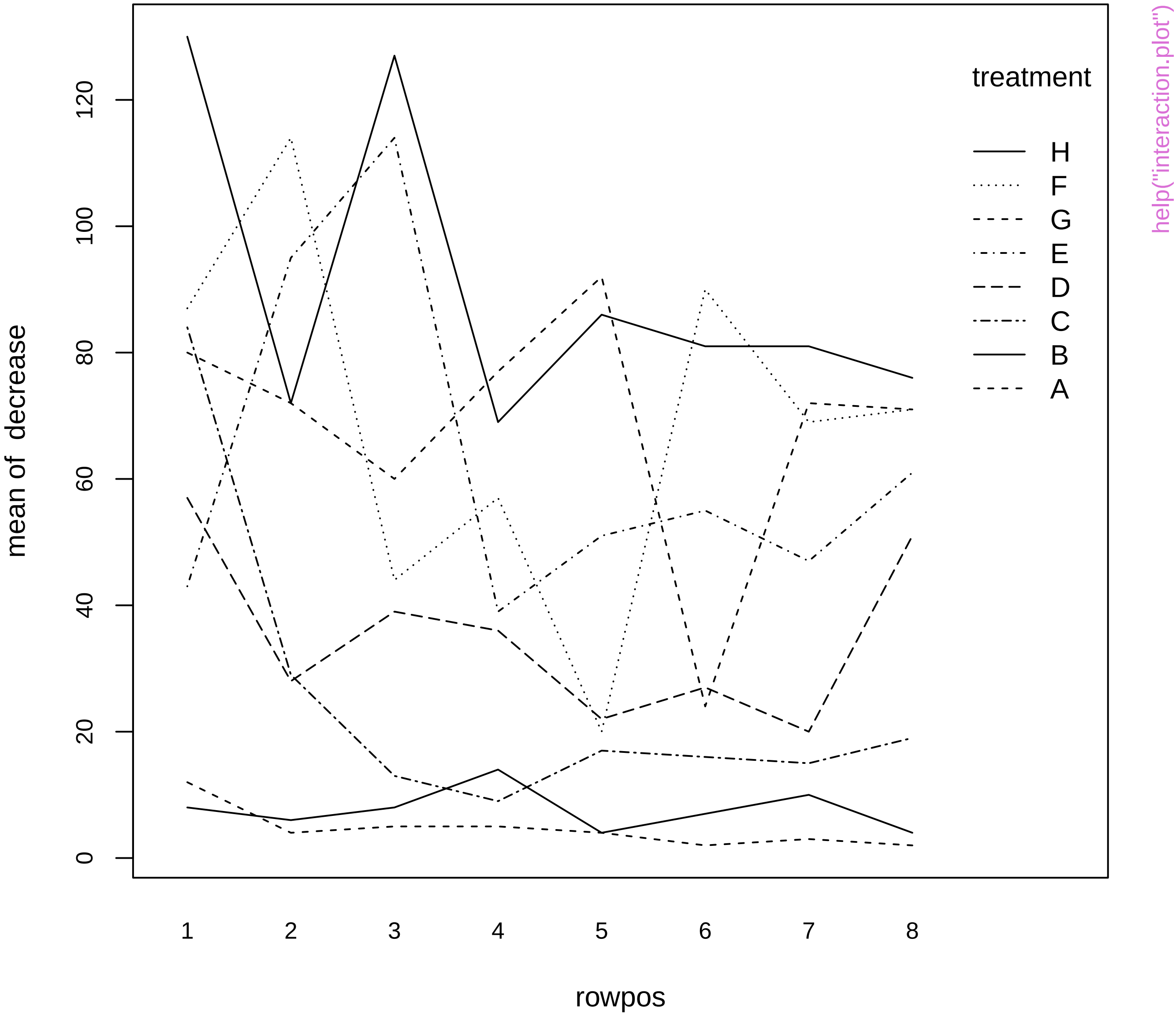
supp
OJ
VC
mean of len
dose
25
20
15
10
0.5
1
2
help("interaction.plot")



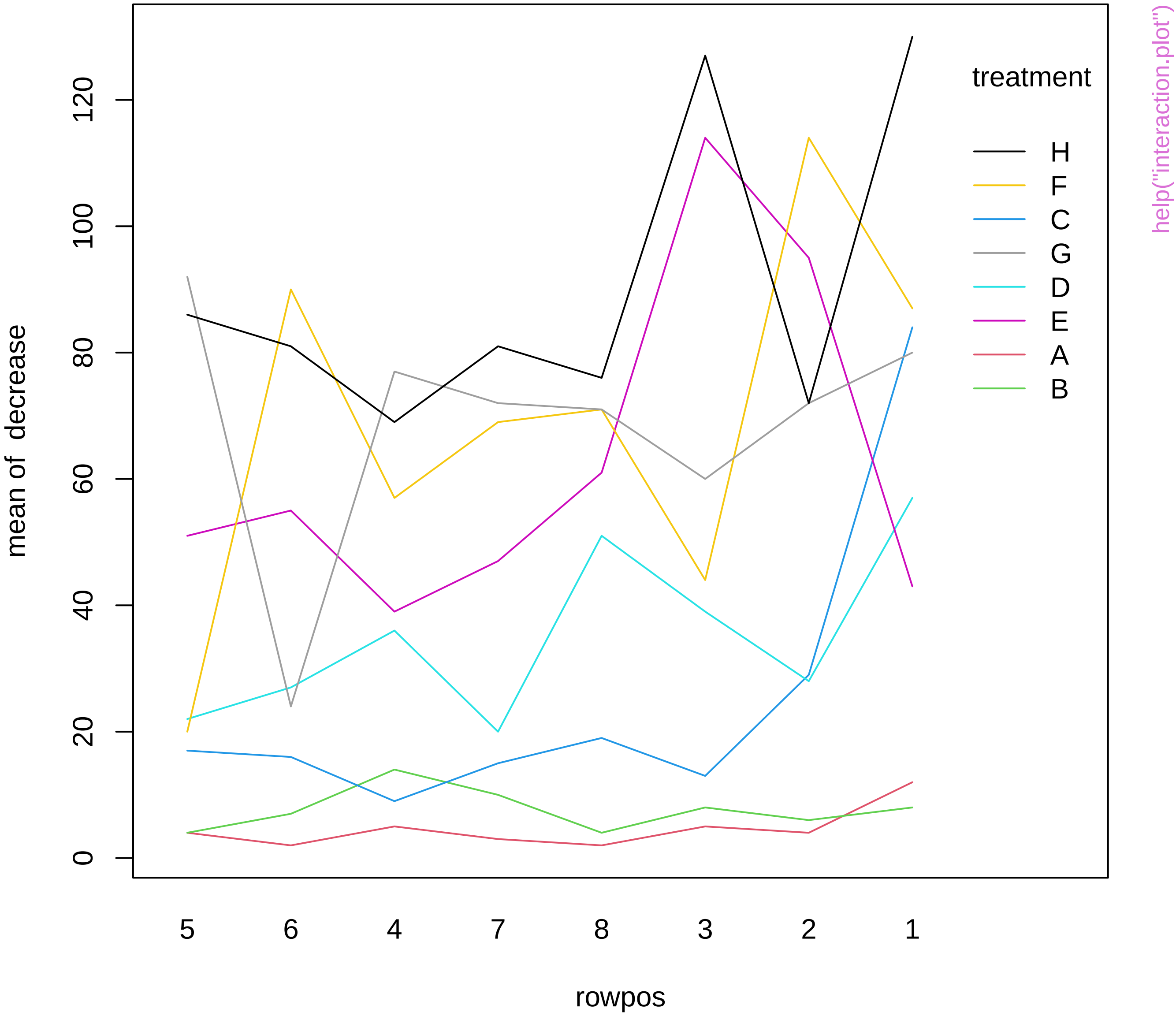
mean of len
dose
25
20
15
10
0.5
1
2
supp
1 OJ
2 VC
help("interaction.plot")



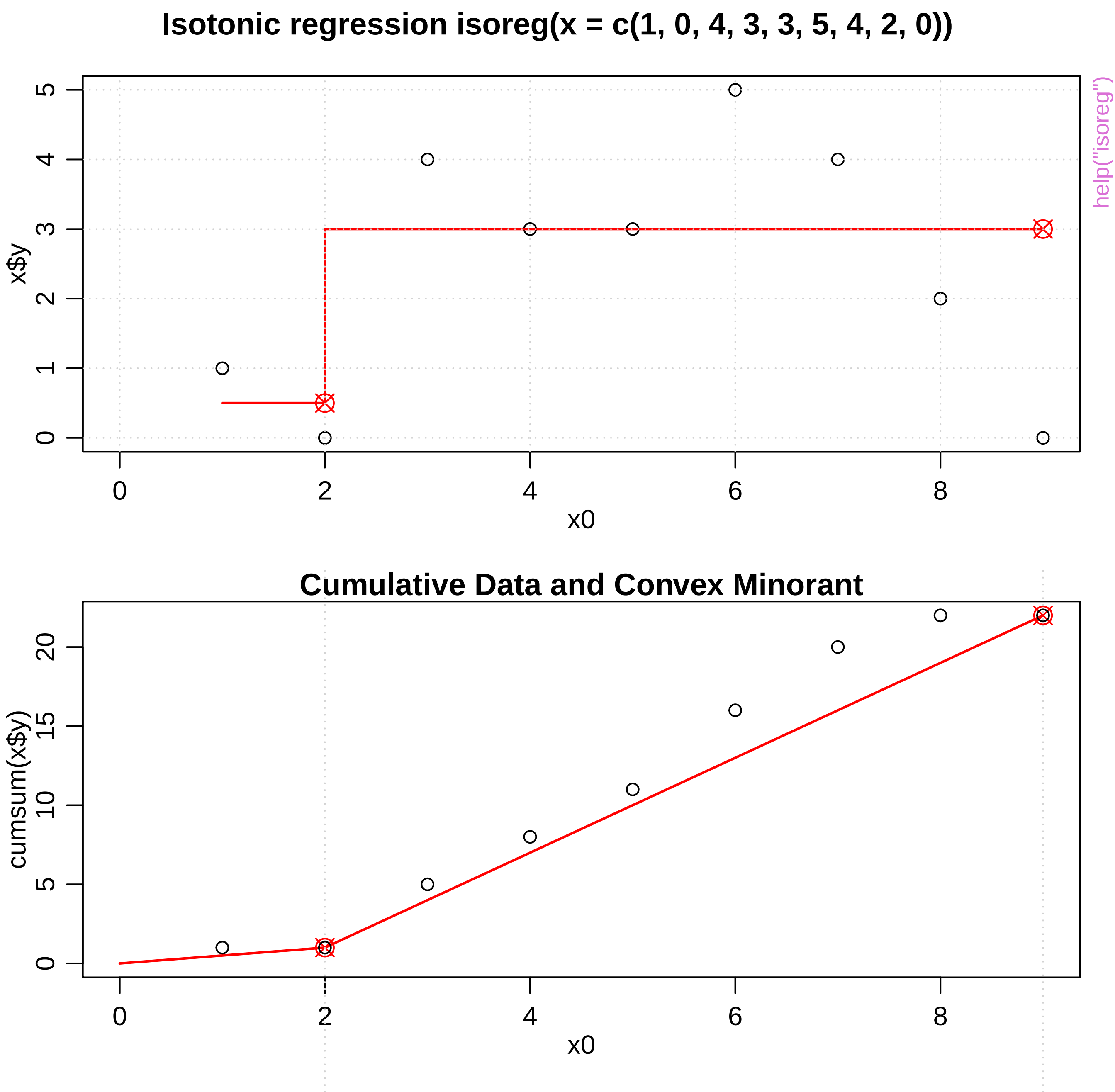
help("interaction.plot")
rowpos
1
3
5
6
7
8
2
4
mean of decrease
0
20
40
60
80
100
120
A
B
C
D
E
F
G
H
treatment

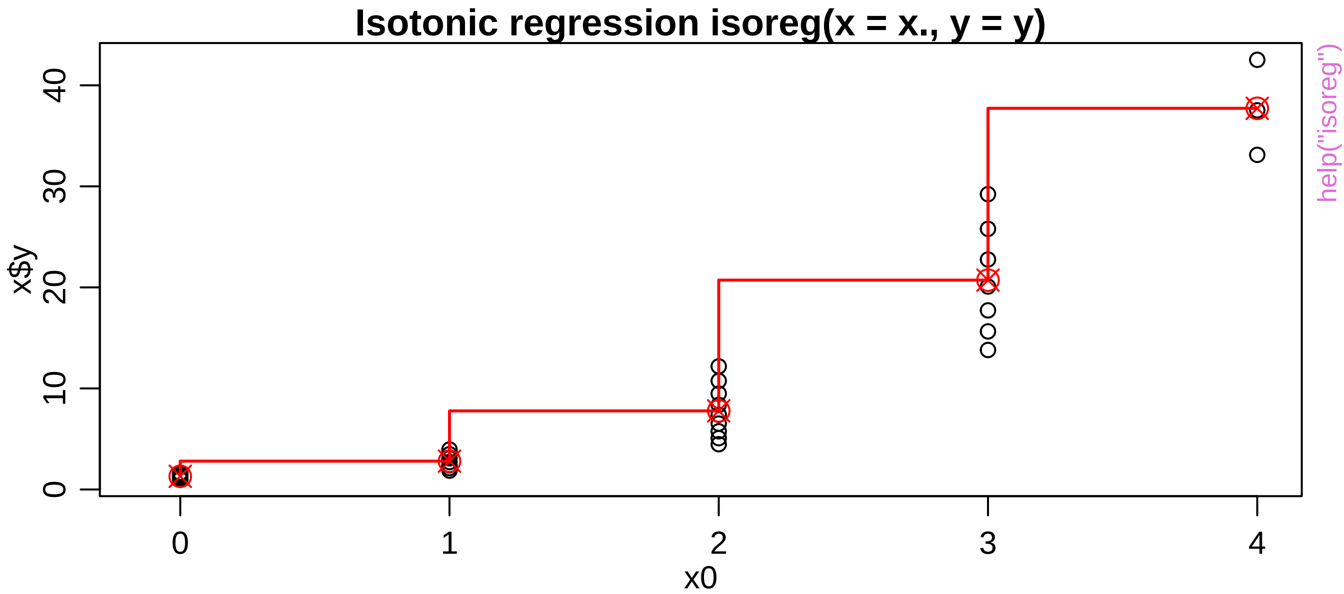


treatment
H
F
G
E
D
C
B
A
mean of decrease
rowpos
0
20
40
60
80
100
120
1
2
3
4
5
6
7
8
help("interaction.plot")

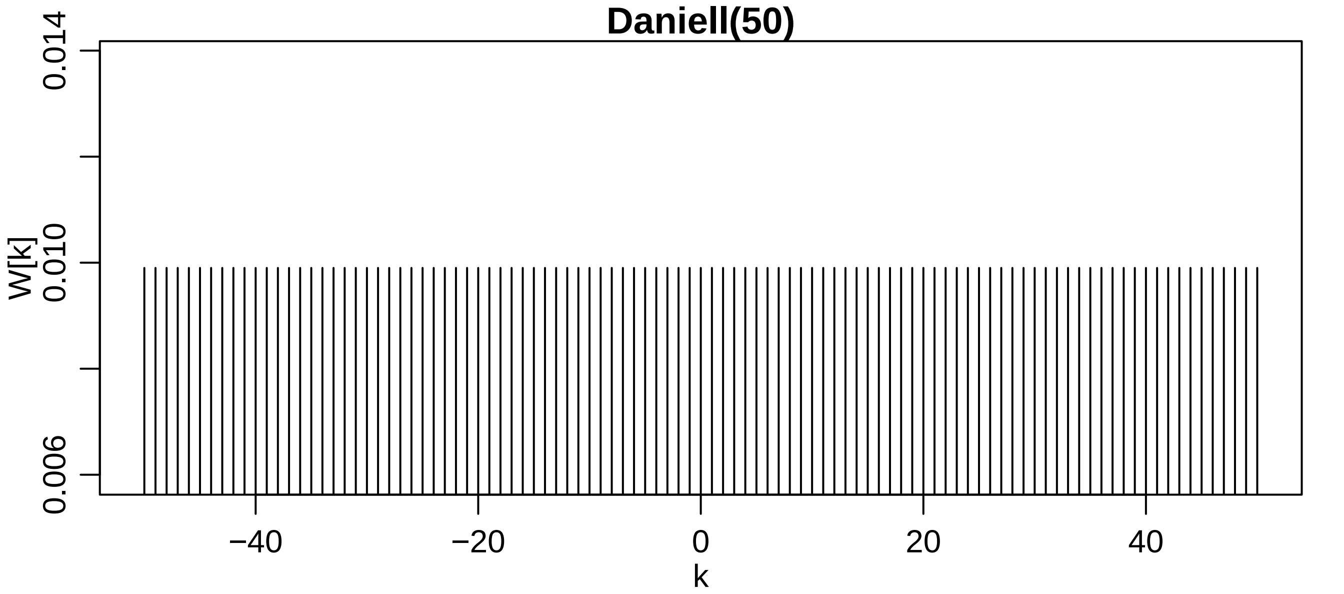


treatment
H
F
C
G
D
E
A
B
mean of decrease
0
20
40
60
80
100
120
5
6
4
7
8
3
2
1
rowpos
help("interaction.plot")

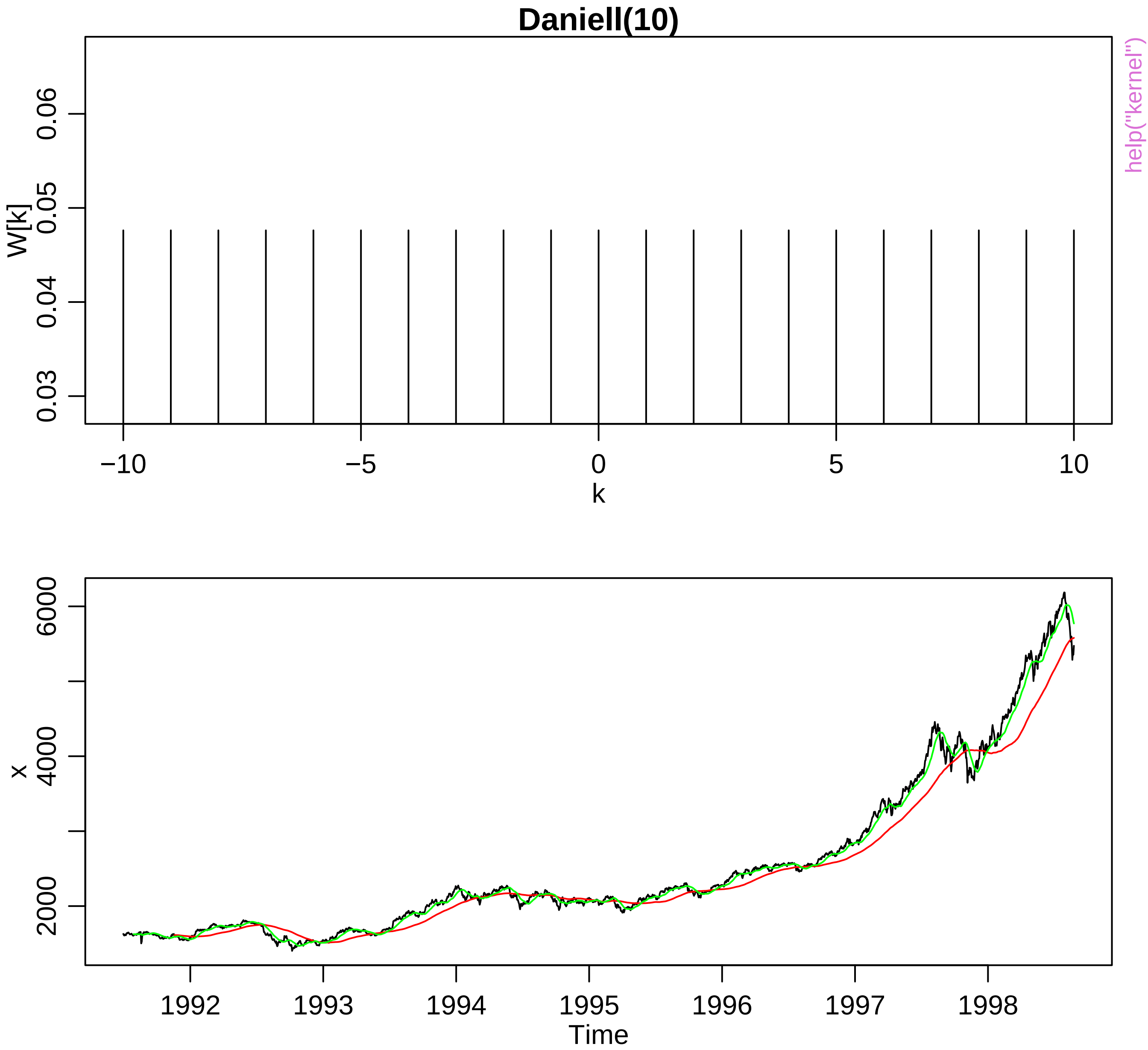

Isotonic regression isoreg(x = c(1, 0, 4, 3, 3, 5, 4, 2, 0))
x$y
x0
help("isoreg")
Cumulative Data and Convex Minorant
cumsum(x$y)
x0



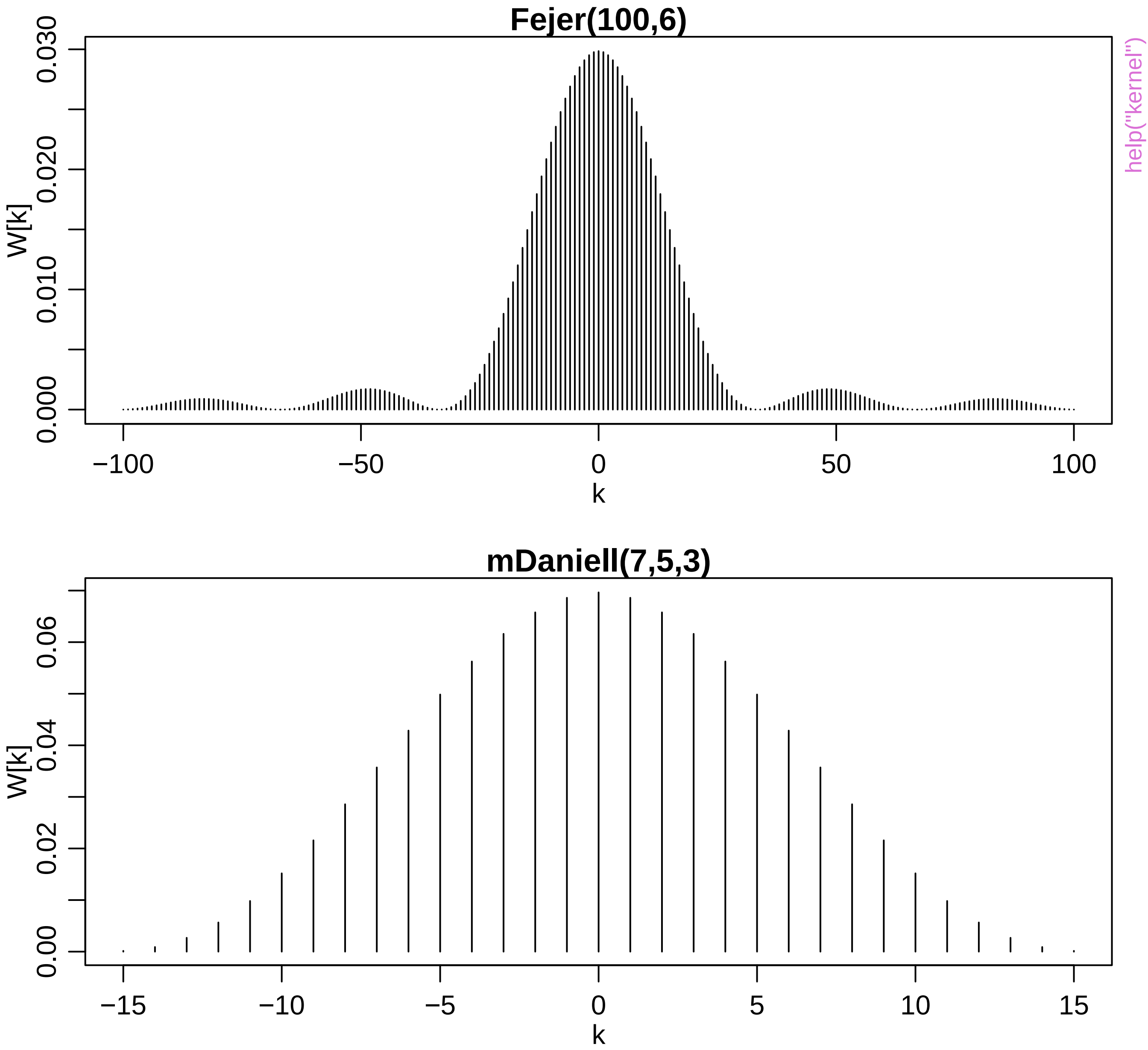
Isotonic regression isoreg(x = x., y = y)
x$y
0
10
20
30
40
0
1
2
3
4
x0
help("isoreg")



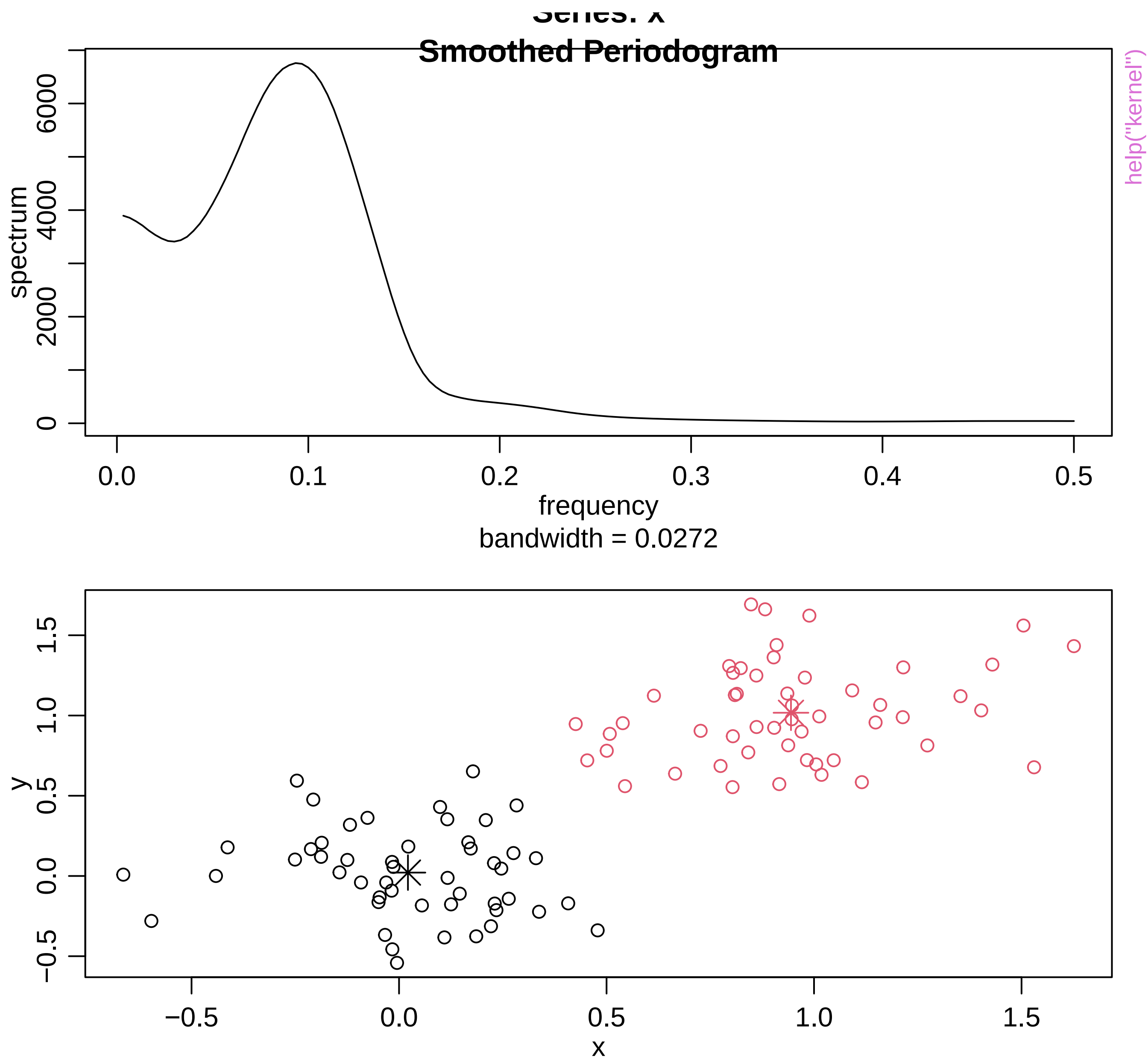
Daniell(50)
W[k]
0.006
0.010
0.014
−40
−20
0
20
40
k



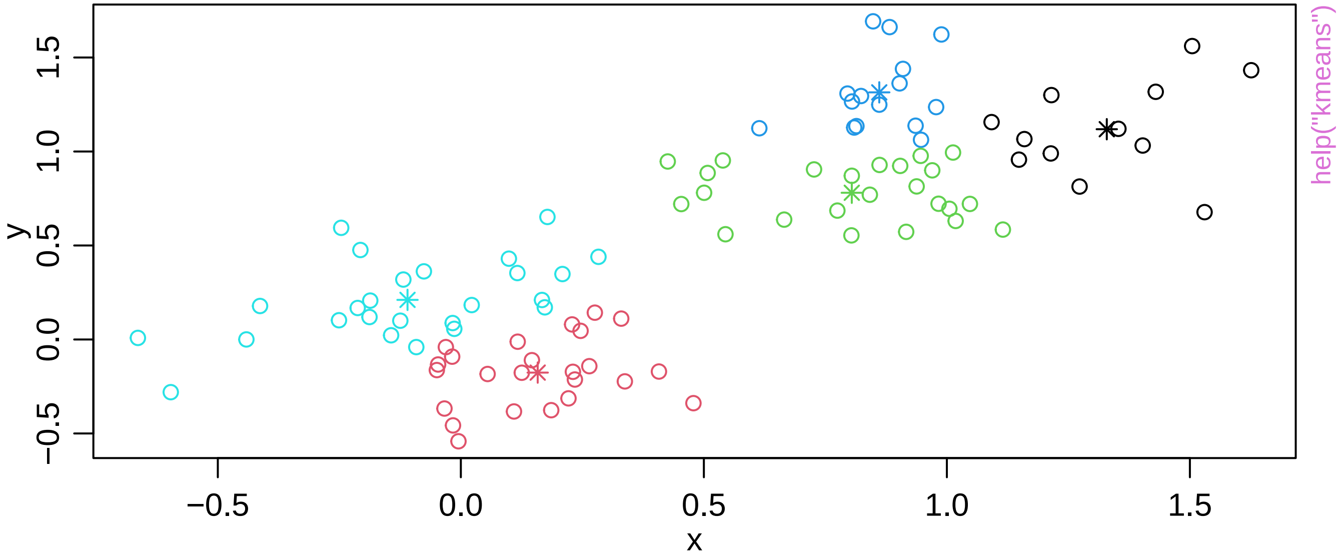
Daniell(10)
help("kernel")
W[k]
0.03
0.04
0.05
0.06
−10
−5
0
5
10
k
x
2000
4000
6000
1992
1993
1994
1995
1996
1997
1998
Time

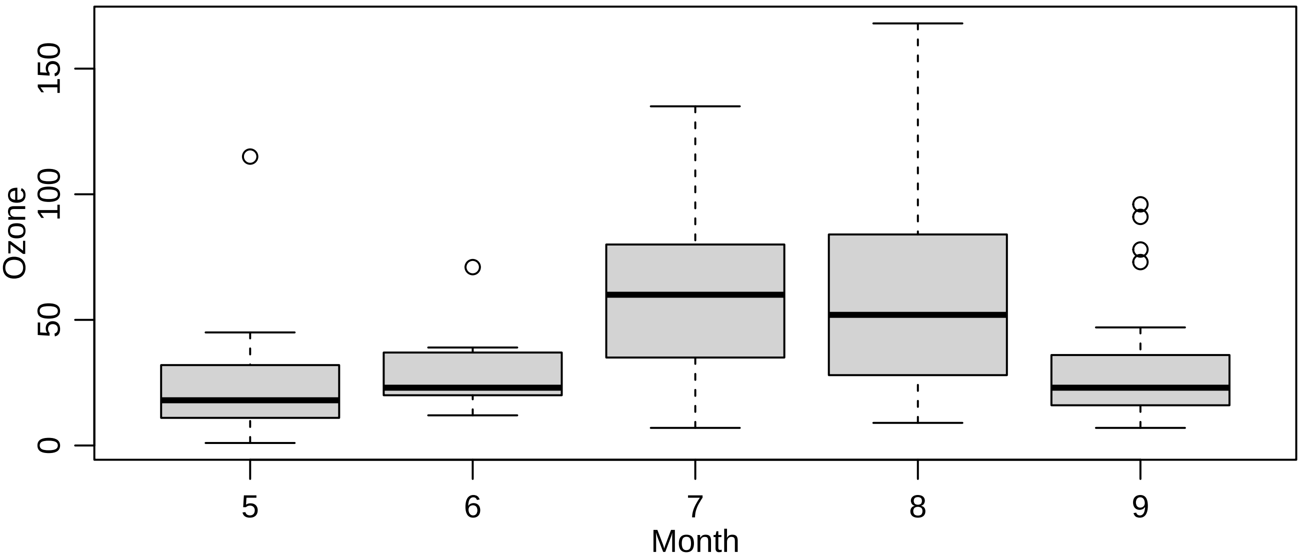


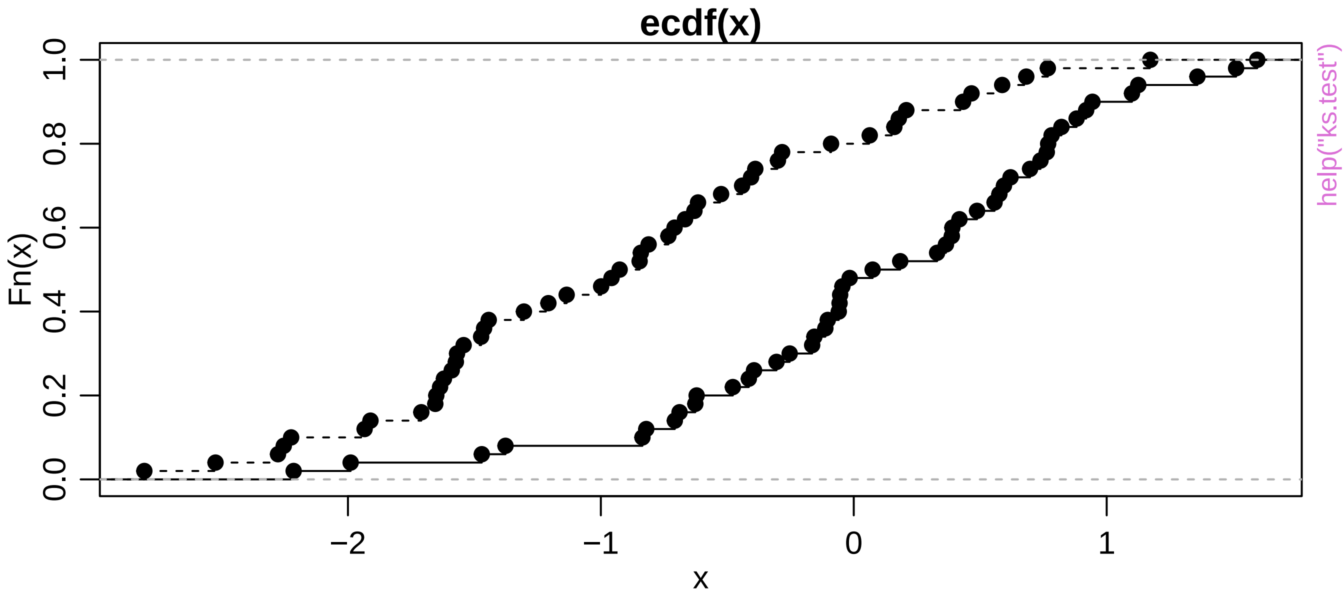
Fejer(100,6)
0.000
0.010
0.020
0.030
W[k]
−100
−50
0
50
100
k
help("kernel")
mDaniell(7,5,3)
0.00
0.02
0.04
0.06
W[k]
−15
−10
−5
0
5
10
15
k



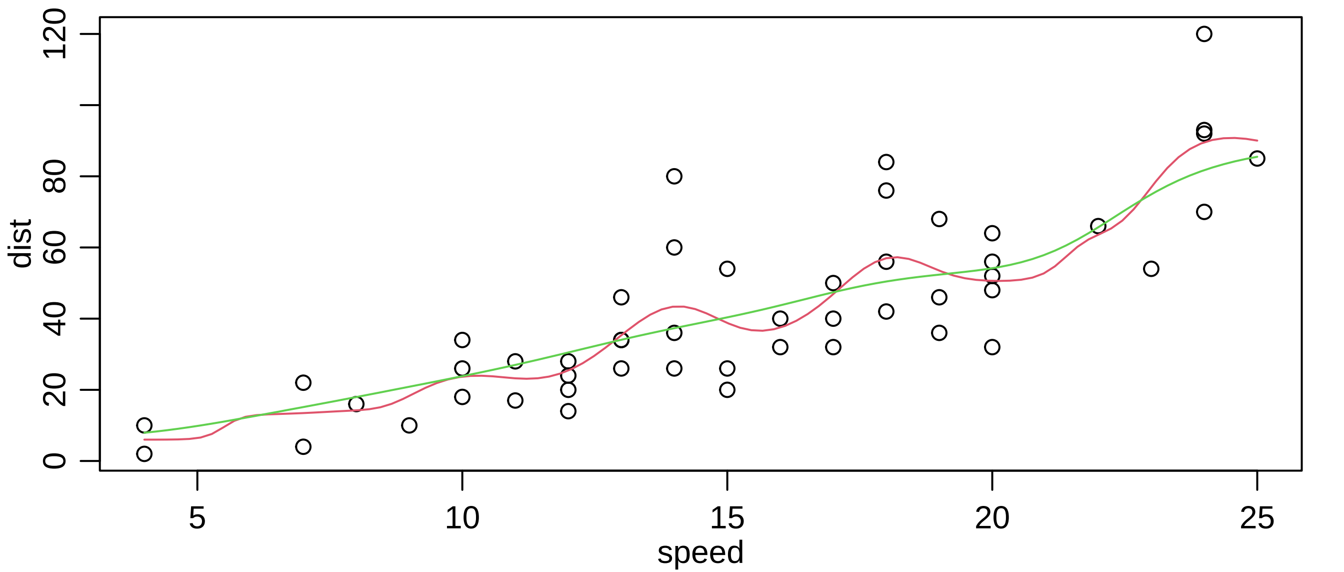
Series: x
Smoothed Periodogram
spectrum
0
2000
4000
6000
0.0
0.1
0.2
0.3
0.4
0.5
frequency
bandwidth = 0.0272
help("kernel")
y
−0.5
0.0
0.5
1.0
1.5
−0.5
0.0
0.5
1.0
1.5
x


help("kmeans")
y
x
−0.5
0.0
0.5
1.0
1.5

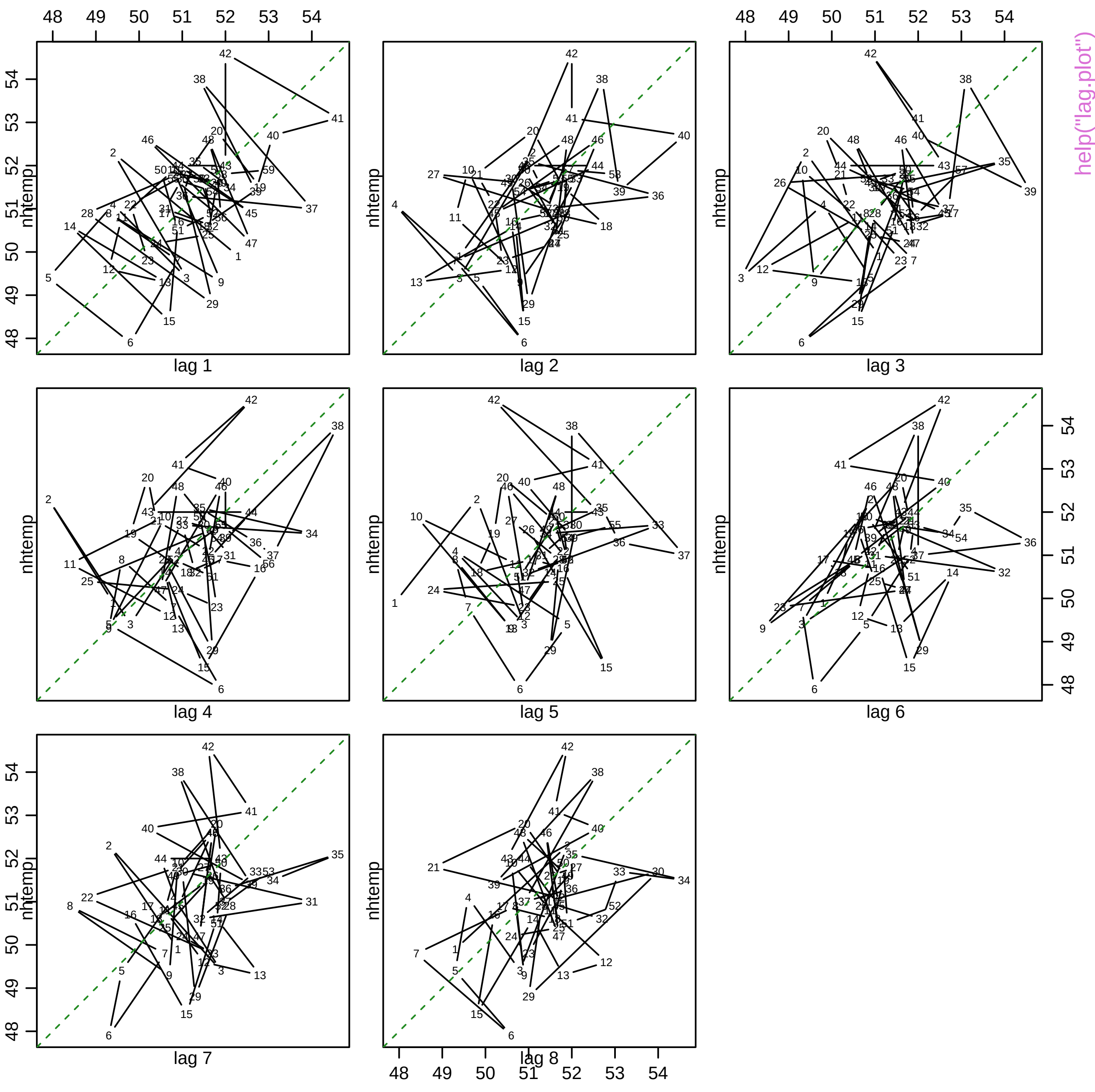

Ozone
Month
0
50
100
150
5
6
7
8
9



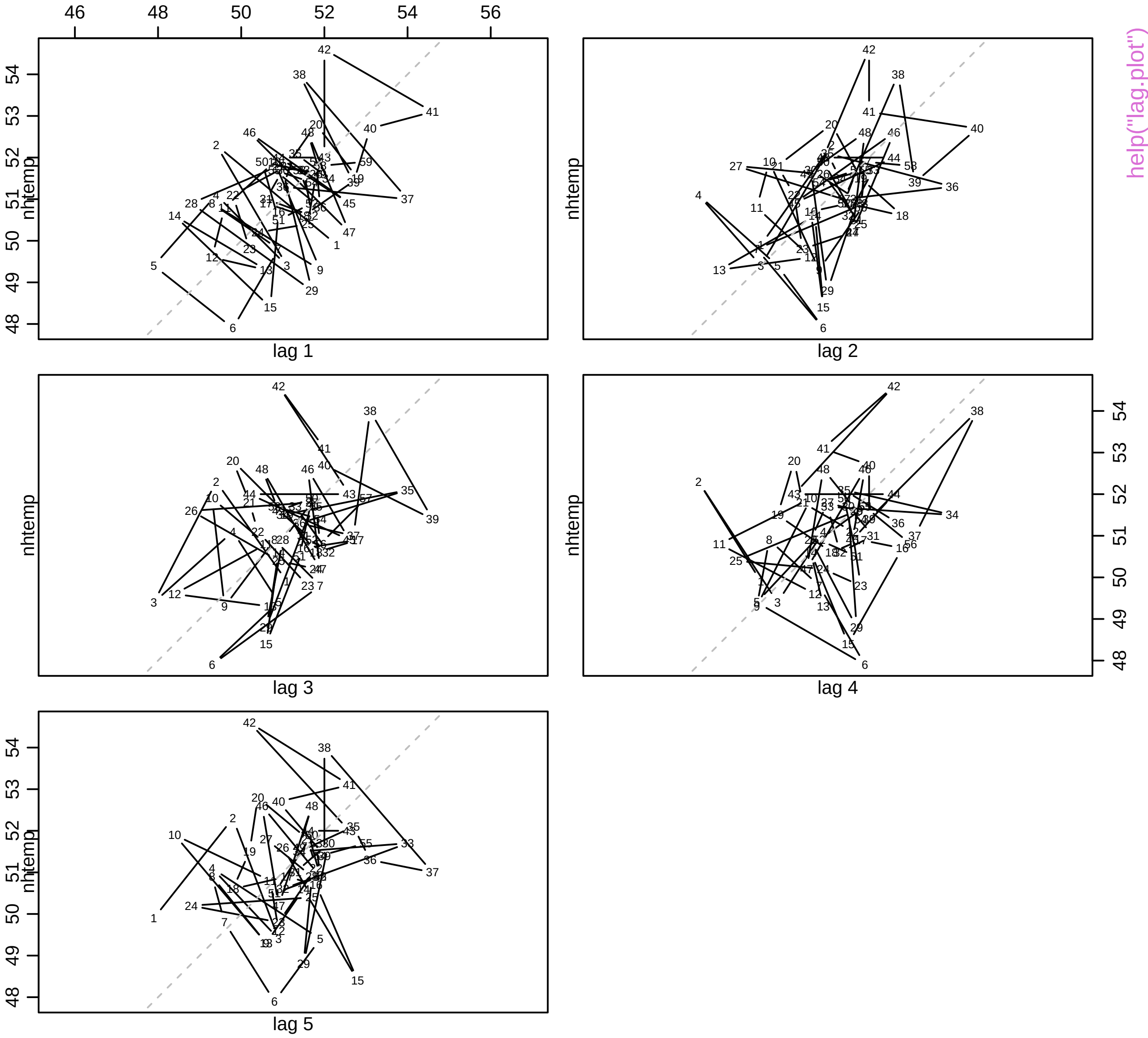
ecdf(x)
Fn(x)
x
help("ks.test")

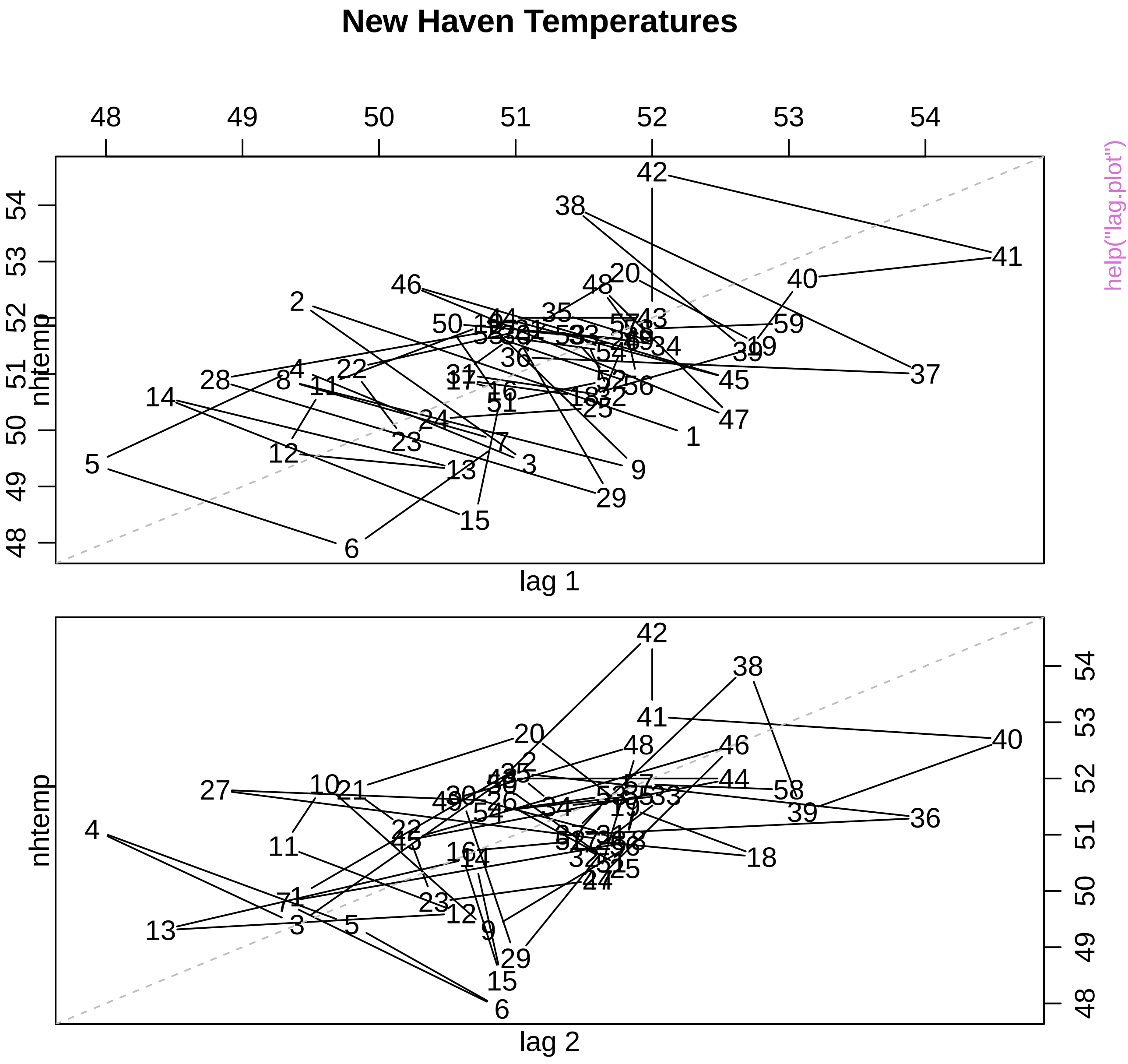


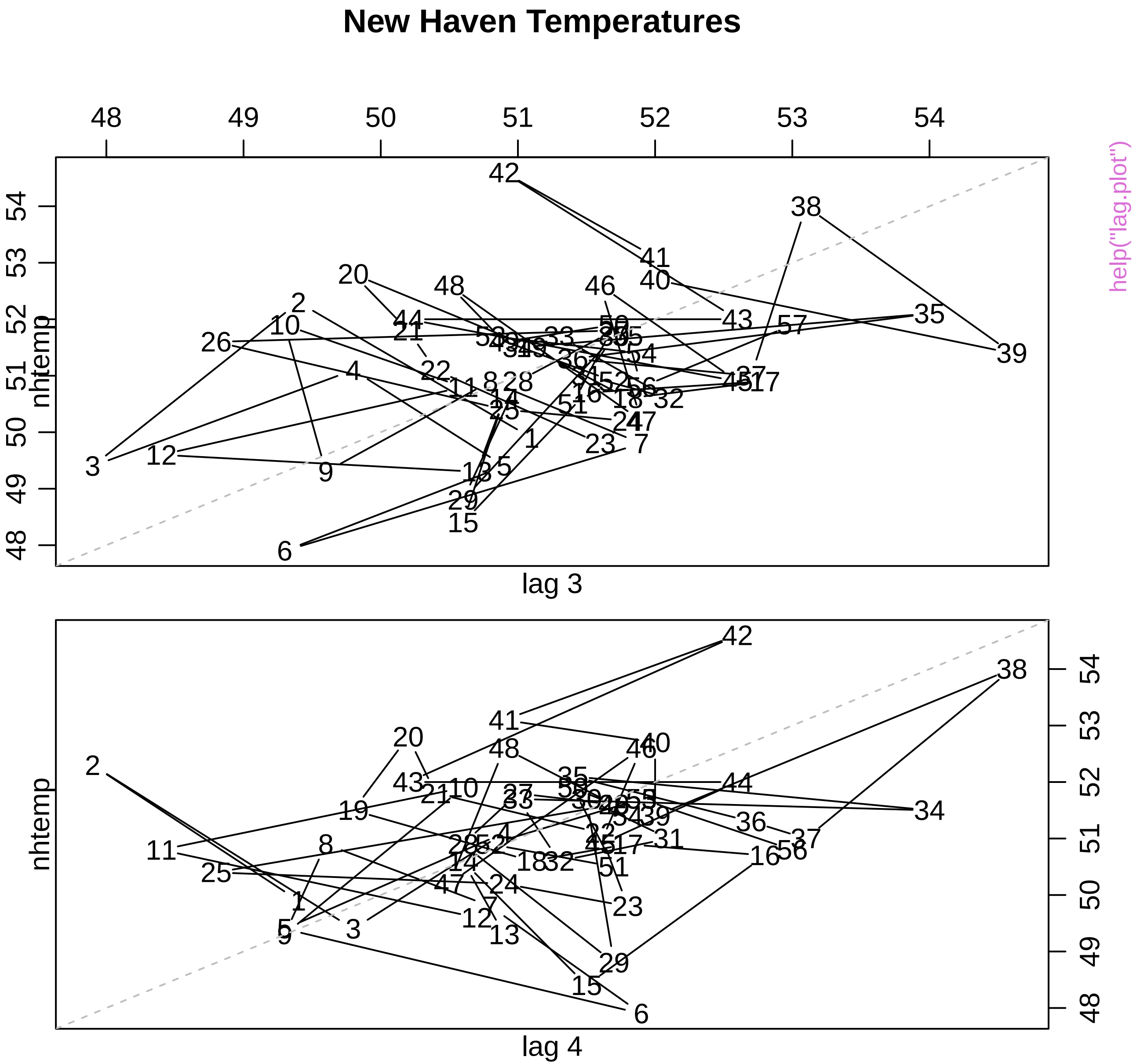
dist
speed

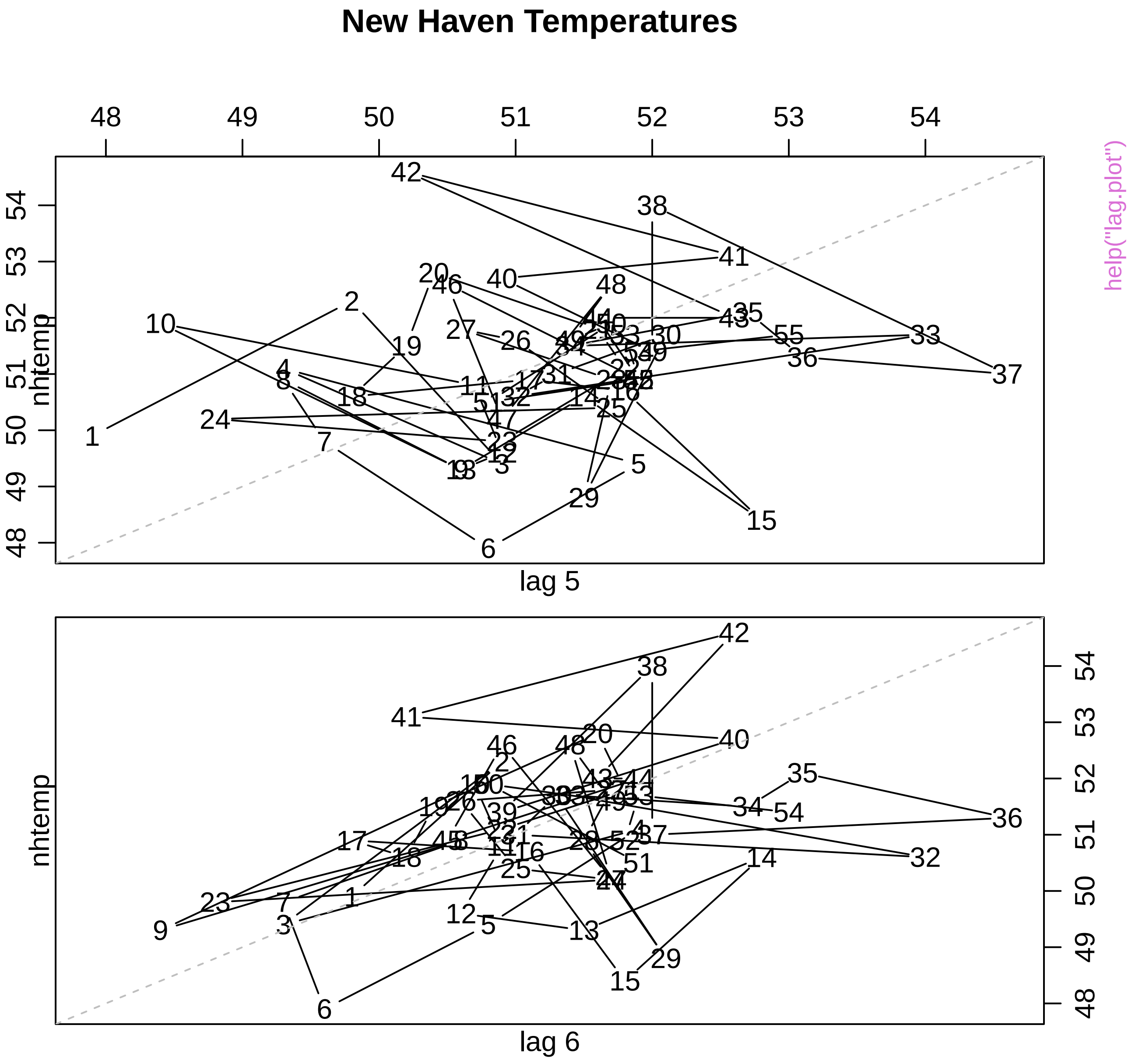

nhtemp
lag 1
lag 2
lag 3
lag 4
lag 5
lag 6
lag 7
lag 8
48 49 50 51 52 53 54
help("lag.plot")

Average Temperatures in New Haven

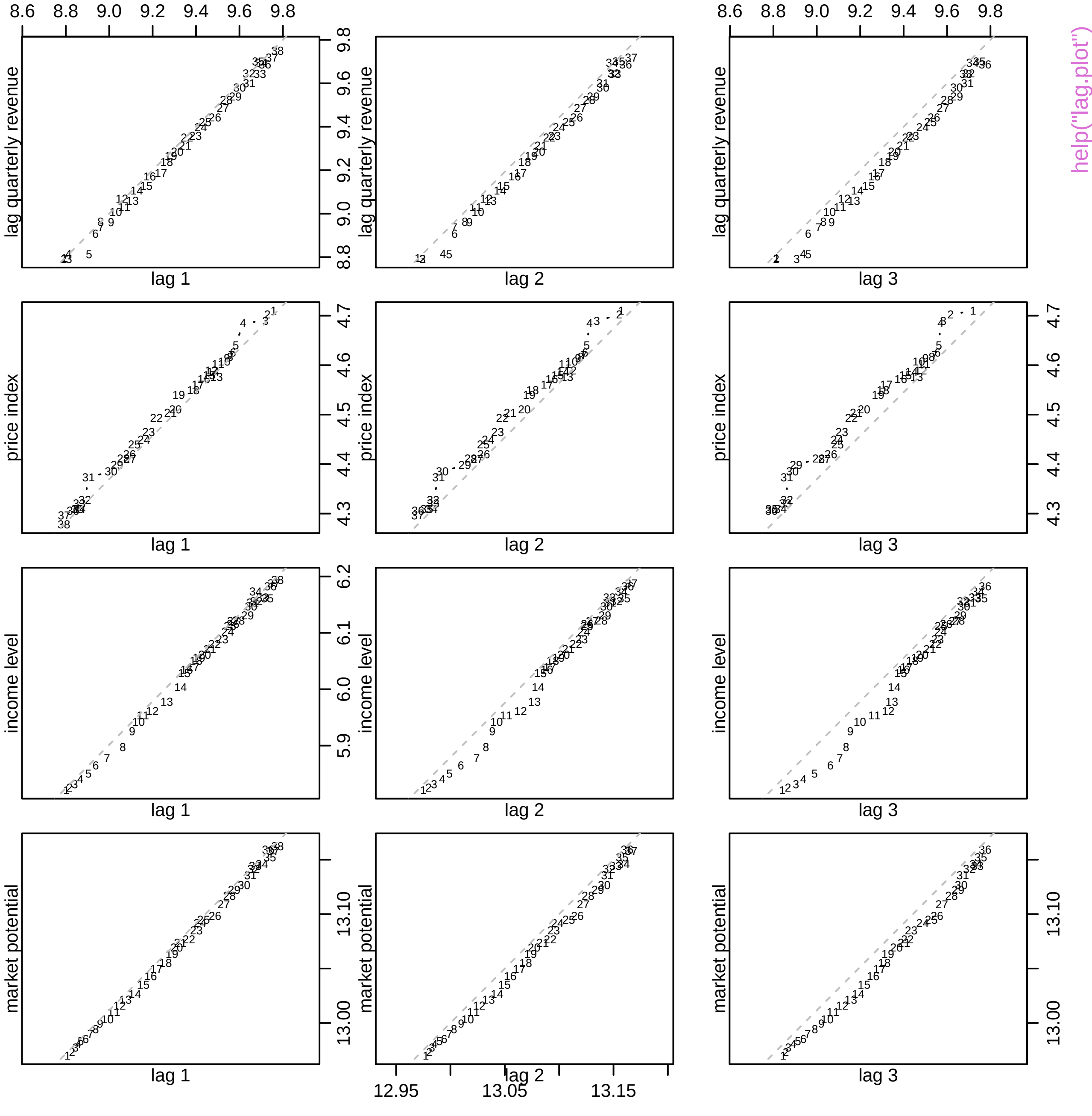



New Haven Temperatures
48
49
50
51
52
53
54
nhtemp
lag 1
lag 2
help("lag.plot")

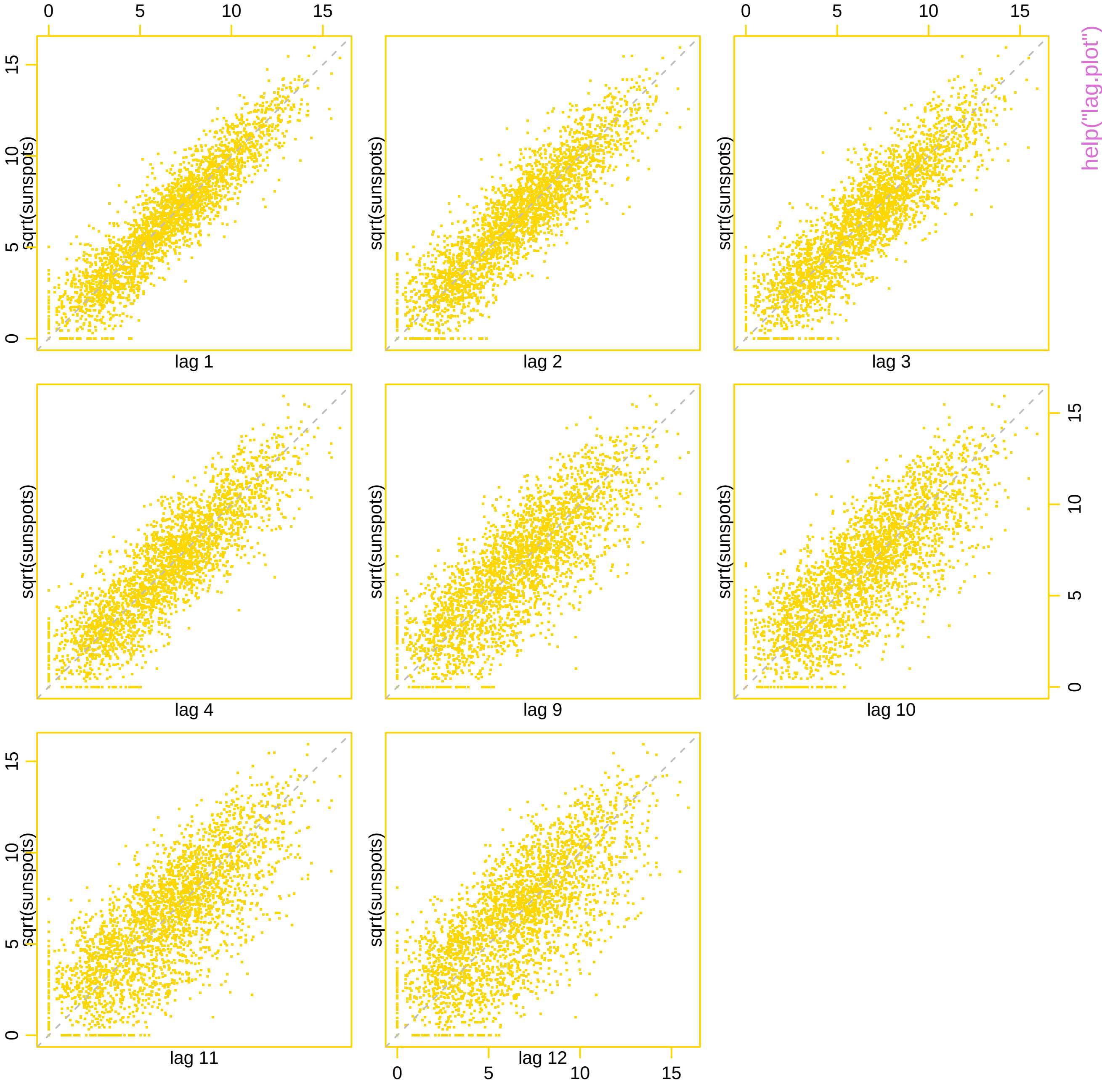

New Haven Temperatures
48 49 50 51 52 53 54
lag 3
lag 4
nhtemp
help("lag.plot")



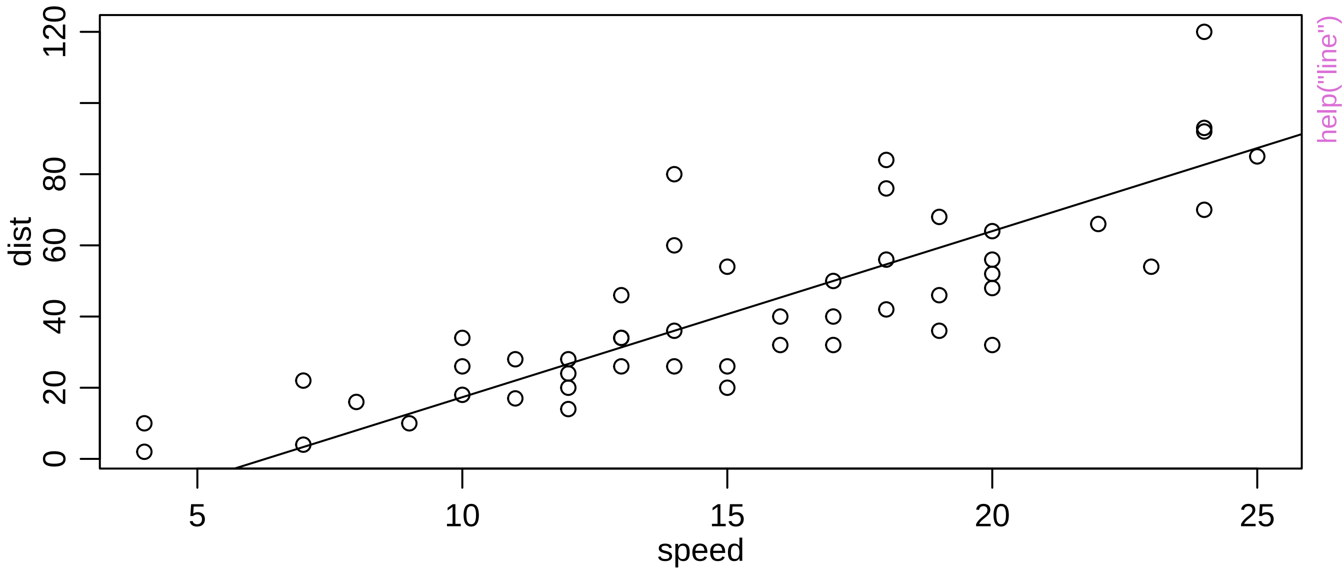
New Haven Temperatures
48
49
50
51
52
53
54
nhtemp
lag 5
nhtemp
lag 6
help("lag.plot")



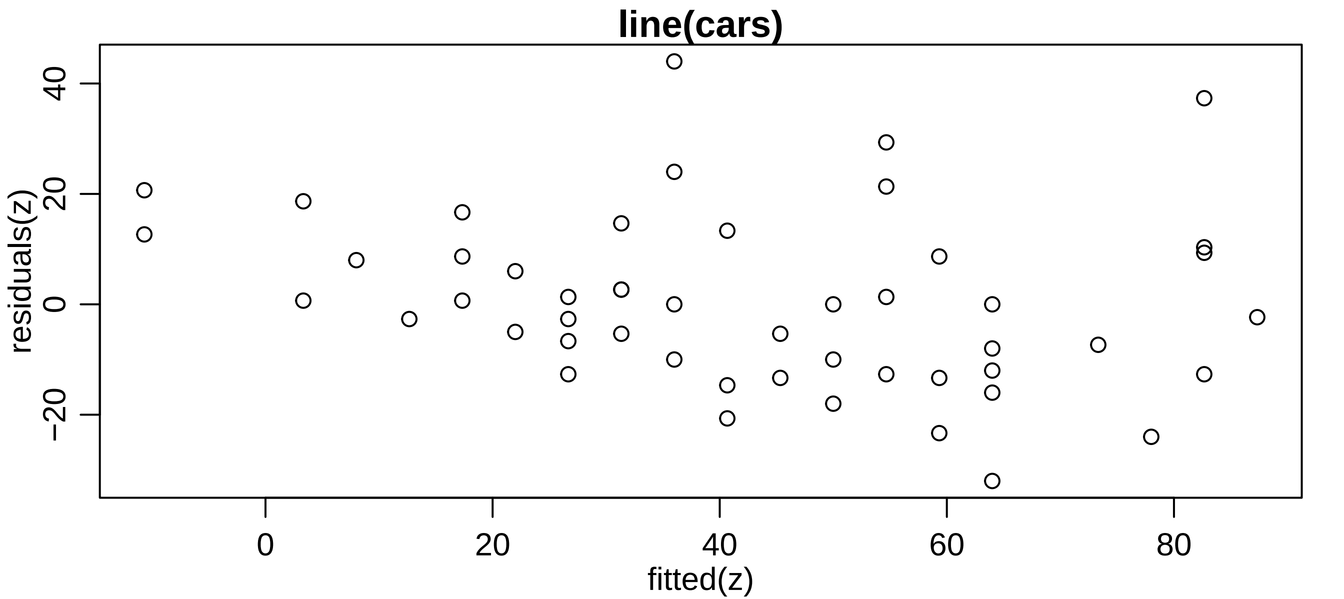
8.6 8.8 9.0 9.2 9.4 9.6 9.8
lag quarterly revenue
price index
income level
market potential
lag 1
lag 2
lag 3
8.8 9.0 9.2 9.4 9.6 9.8
4.3 4.4 4.5 4.6 4.7
5.9 6.0 6.1 6.2
13.00 13.10
12.95 13.05 13.15
help("lag.plot")



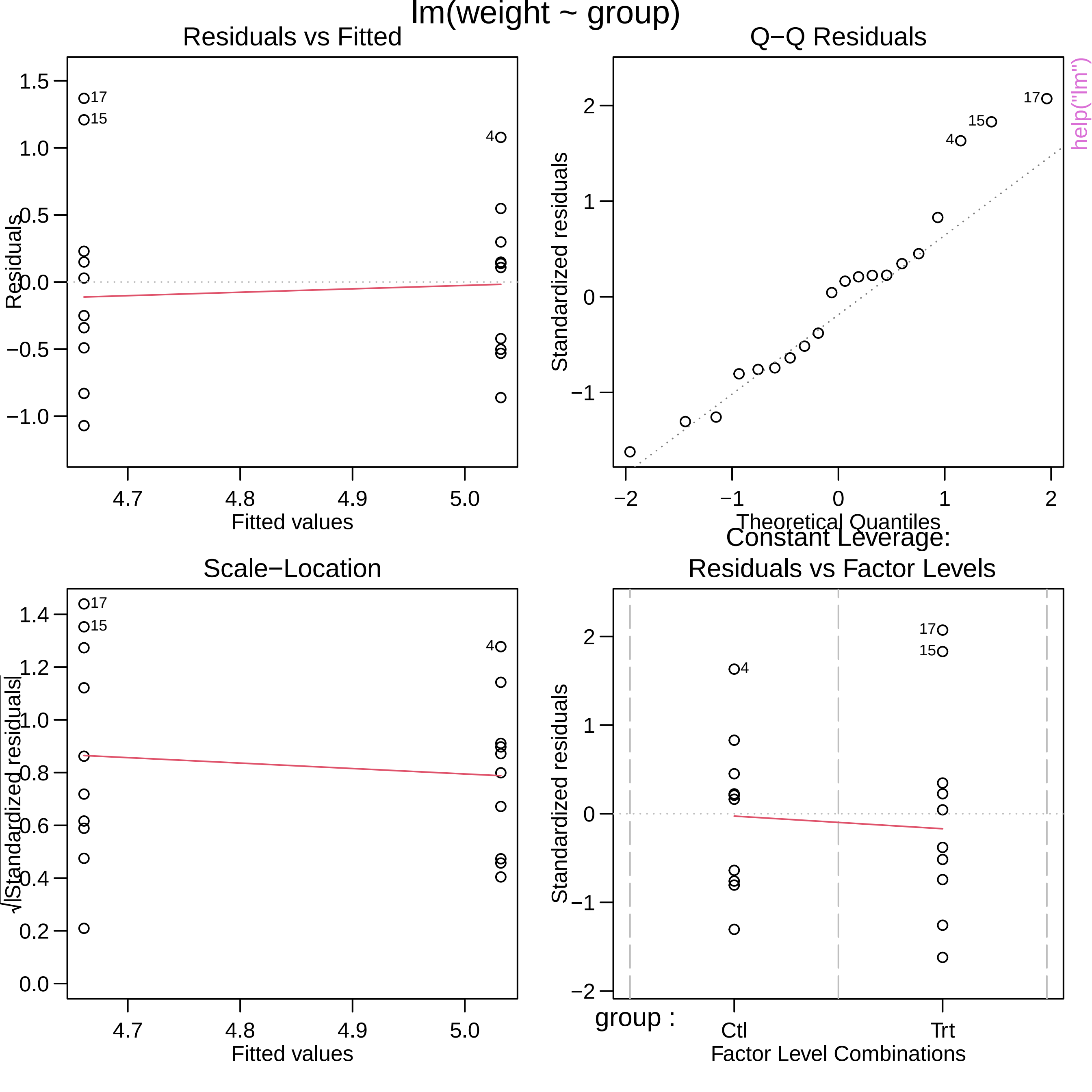
help("lag.plot")
sqrt(sunspots)
lag 1
lag 2
lag 3
lag 4
lag 9
lag 10
lag 11
lag 12
0
5
10
15

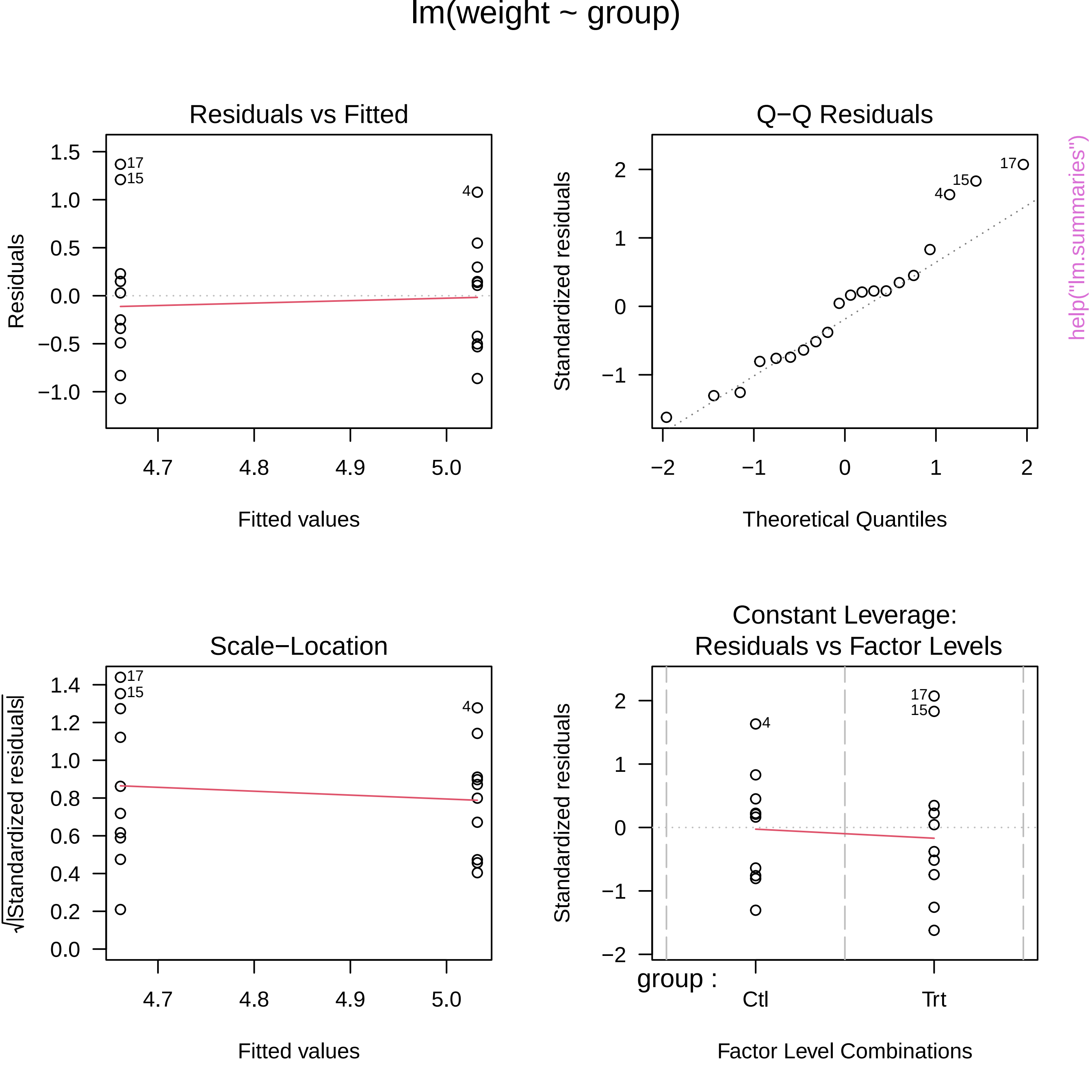


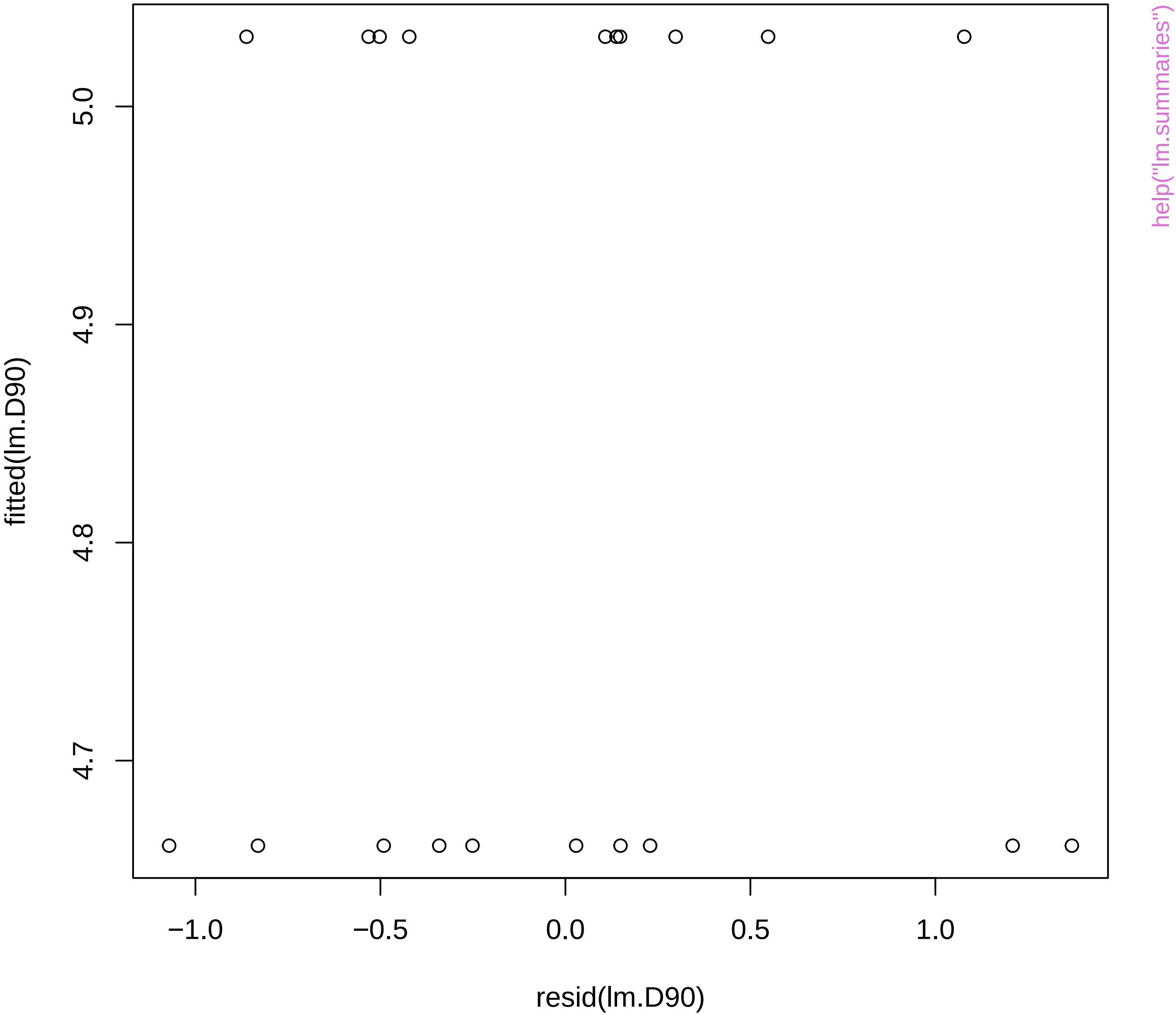
dist
0
20
40
60
80
120
5
10
15
20
25
speed
help("line")



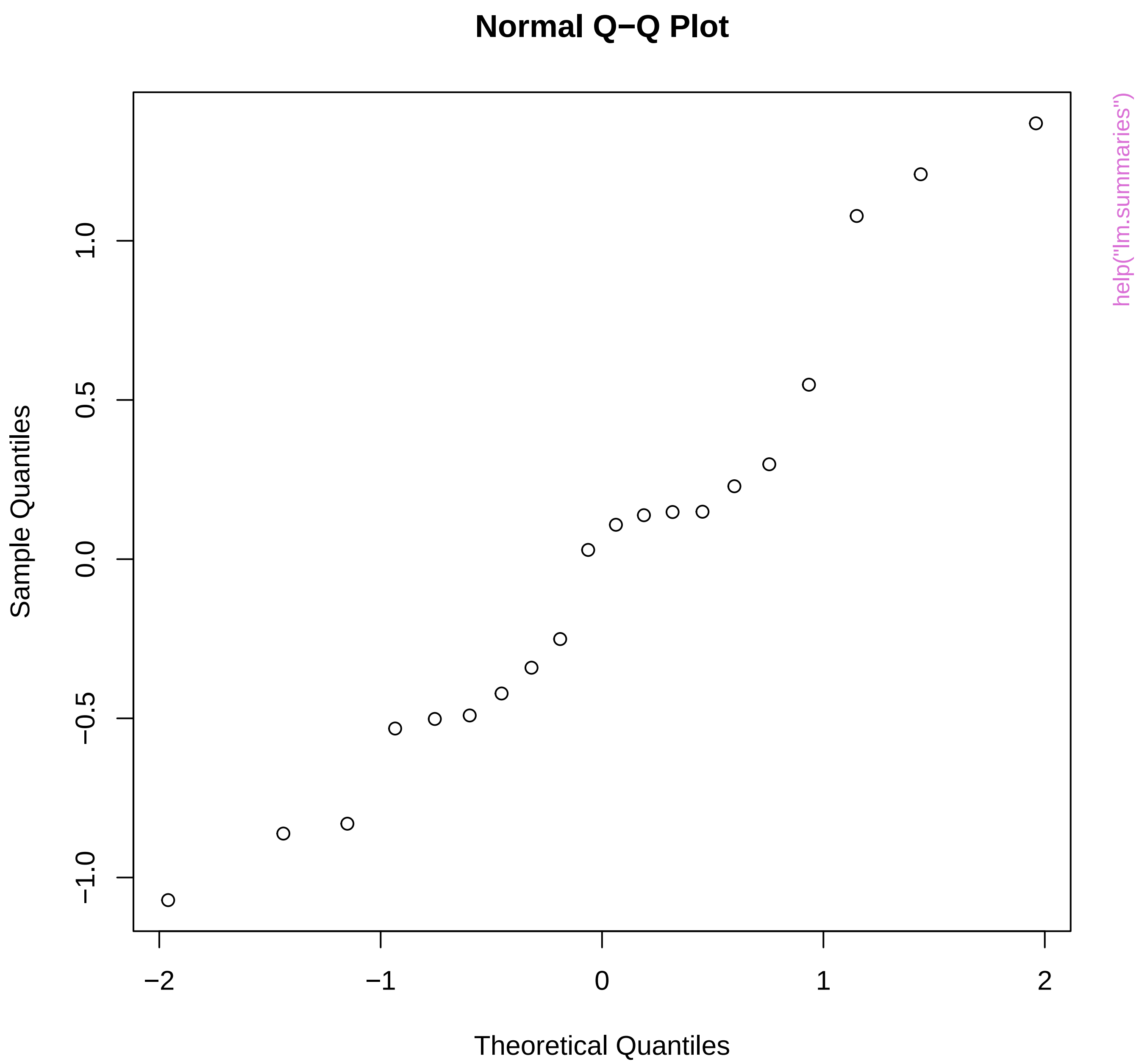
line(cars)
residuals(z)
−20
0
20
40
0
20
40
60
80
fitted(z)

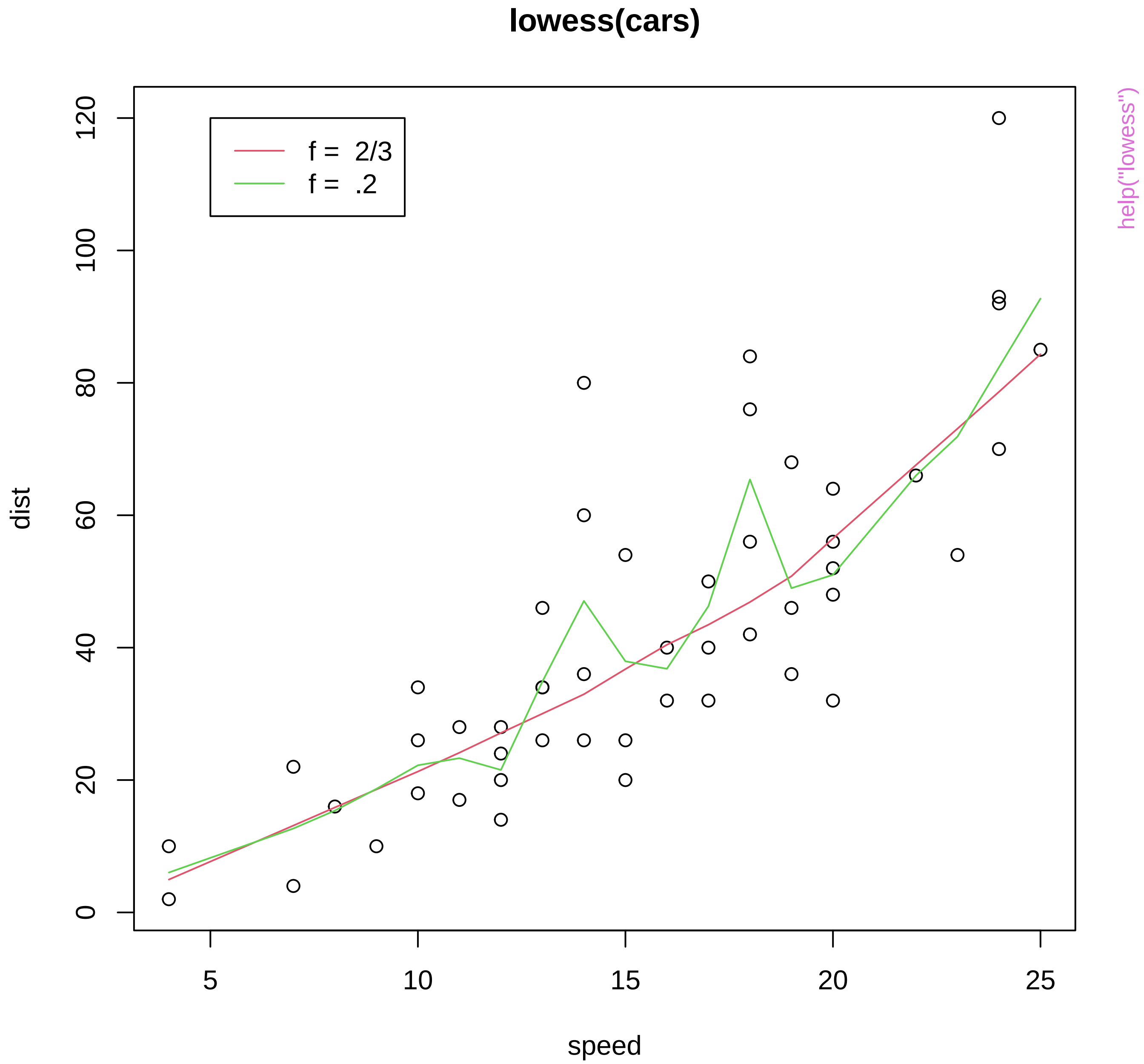

lm(weight ~ group)
Residuals vs Fitted
Residuals
Fitted values
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
help("lm")
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
group :
Ctl
Trt
Factor Level Combinations

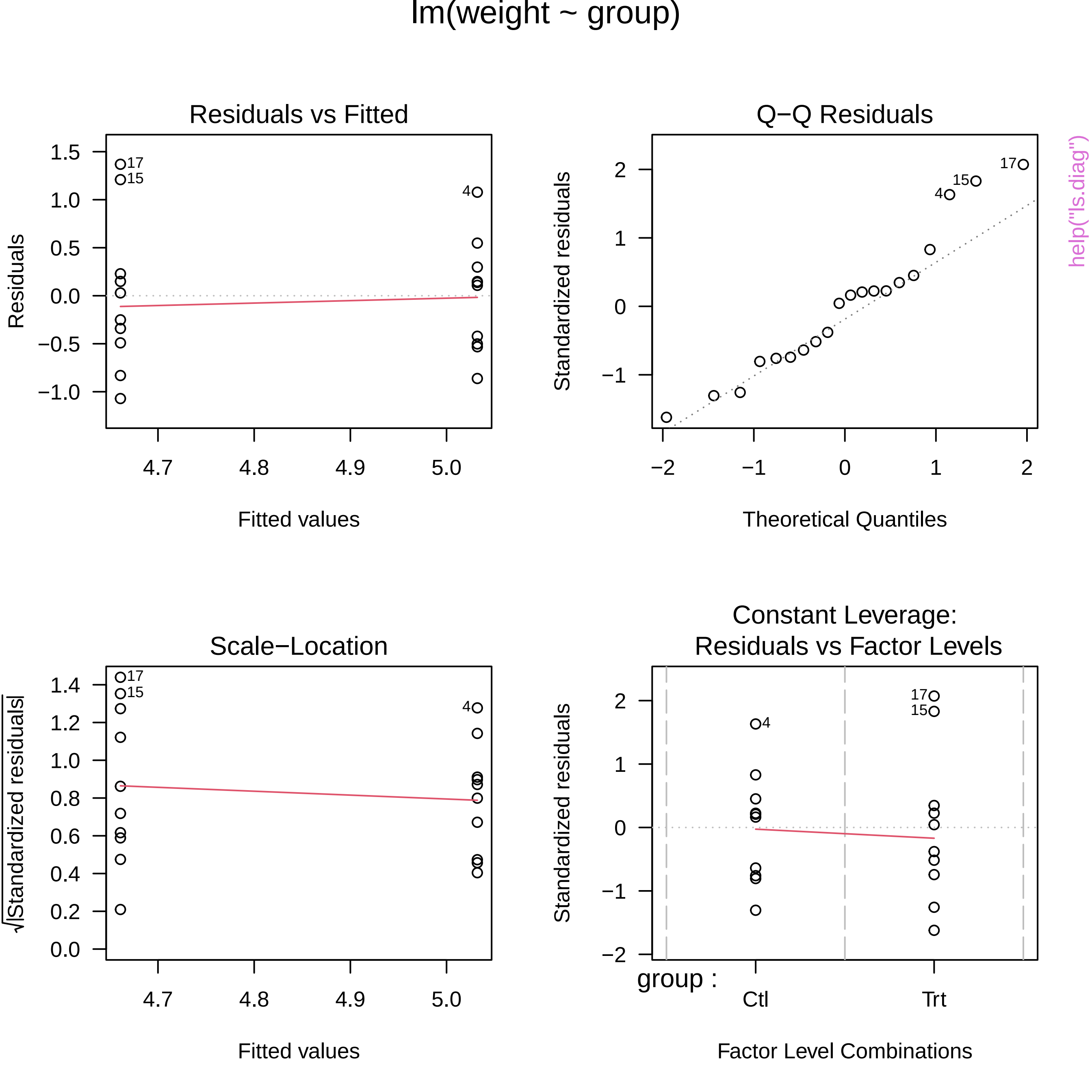

lm(weight ~ group)
Residuals vs Fitted
Residuals
Fitted values
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
help("lm.summaries")
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
group :
Ctl
Trt
Factor Level Combinations

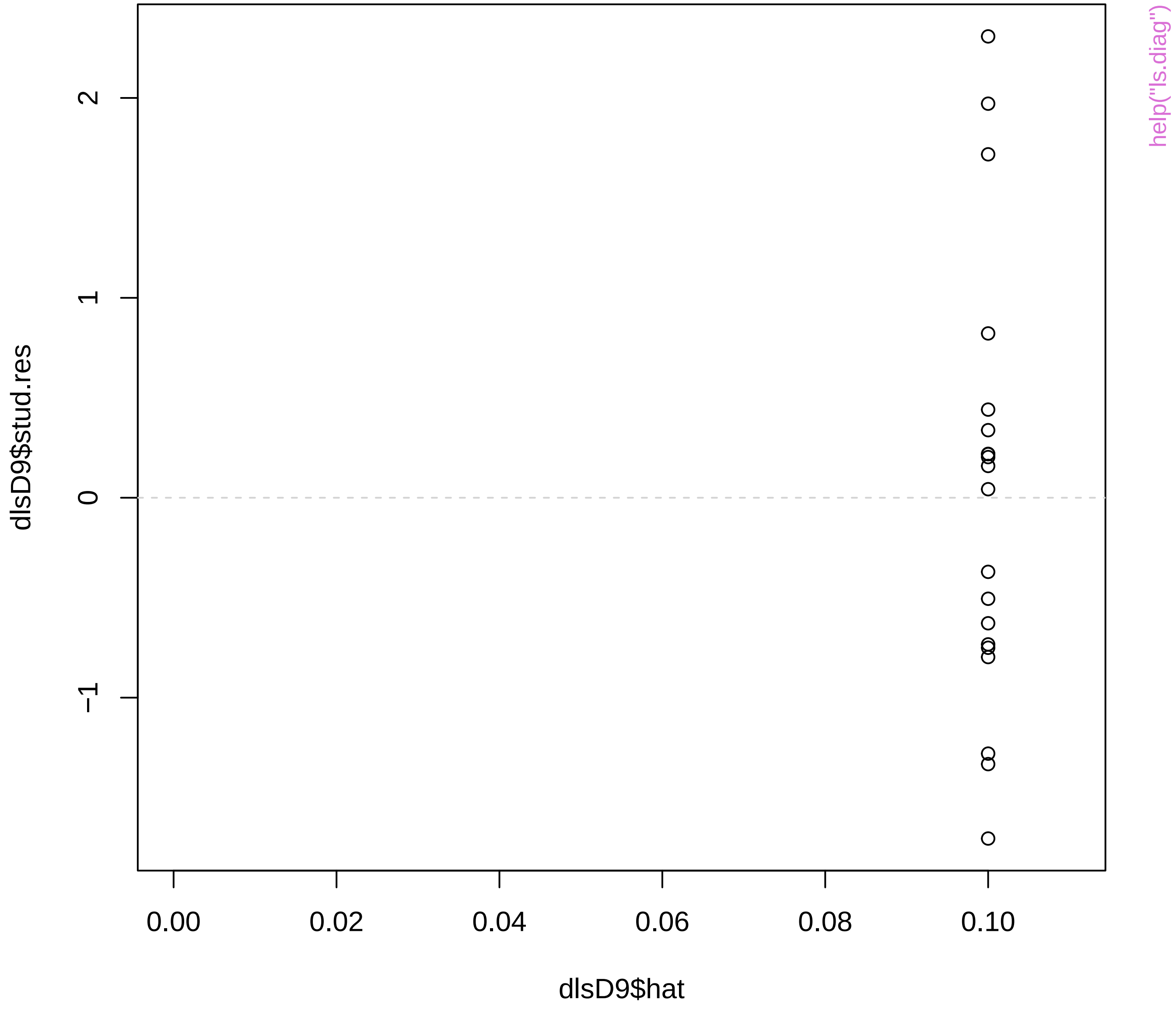


fitted(lm.D90)
5.0
4.9
4.8
4.7
−1.0
−0.5
0.0
0.5
1.0
resid(lm.D90)
help("lm.summaries")

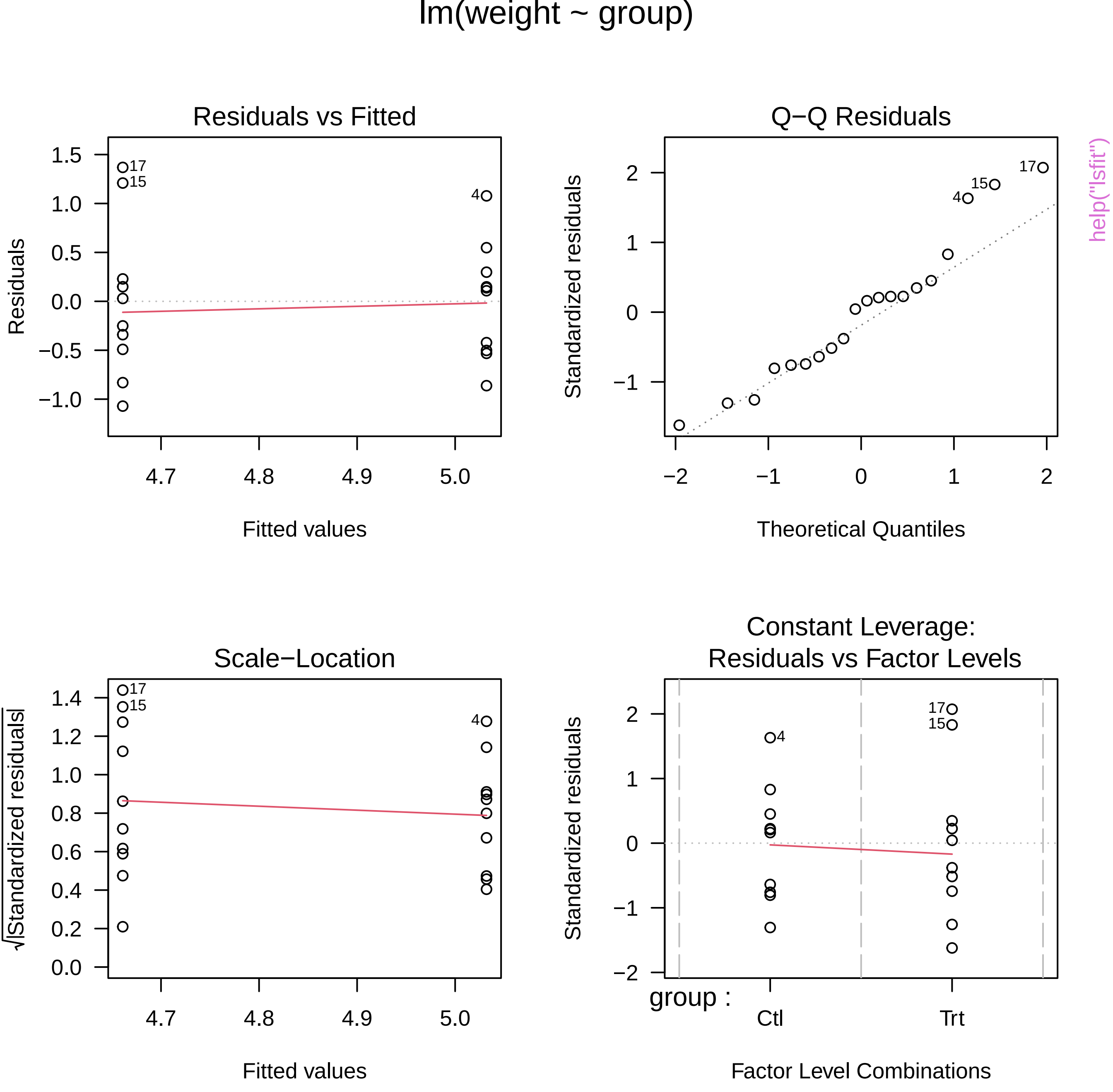

Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
−1.0
−0.5
0.0
0.5
1.0
−2
−1
0
1
2
help("lm.summaries")

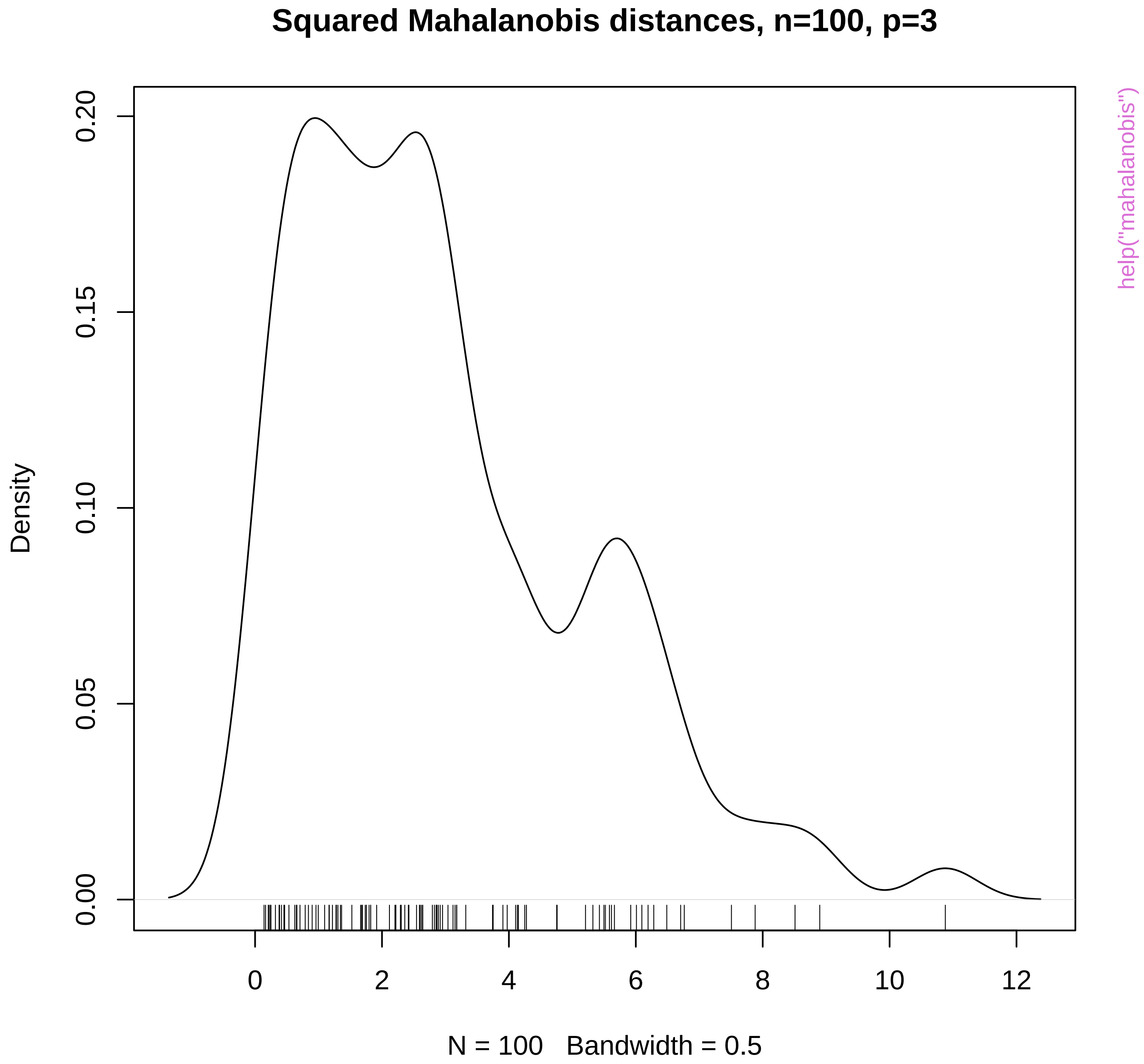

lowess(cars)
f = 2/3
f = .2
dist
speed
0
20
40
60
80
100
120
5
10
15
20
25
help("lowess")

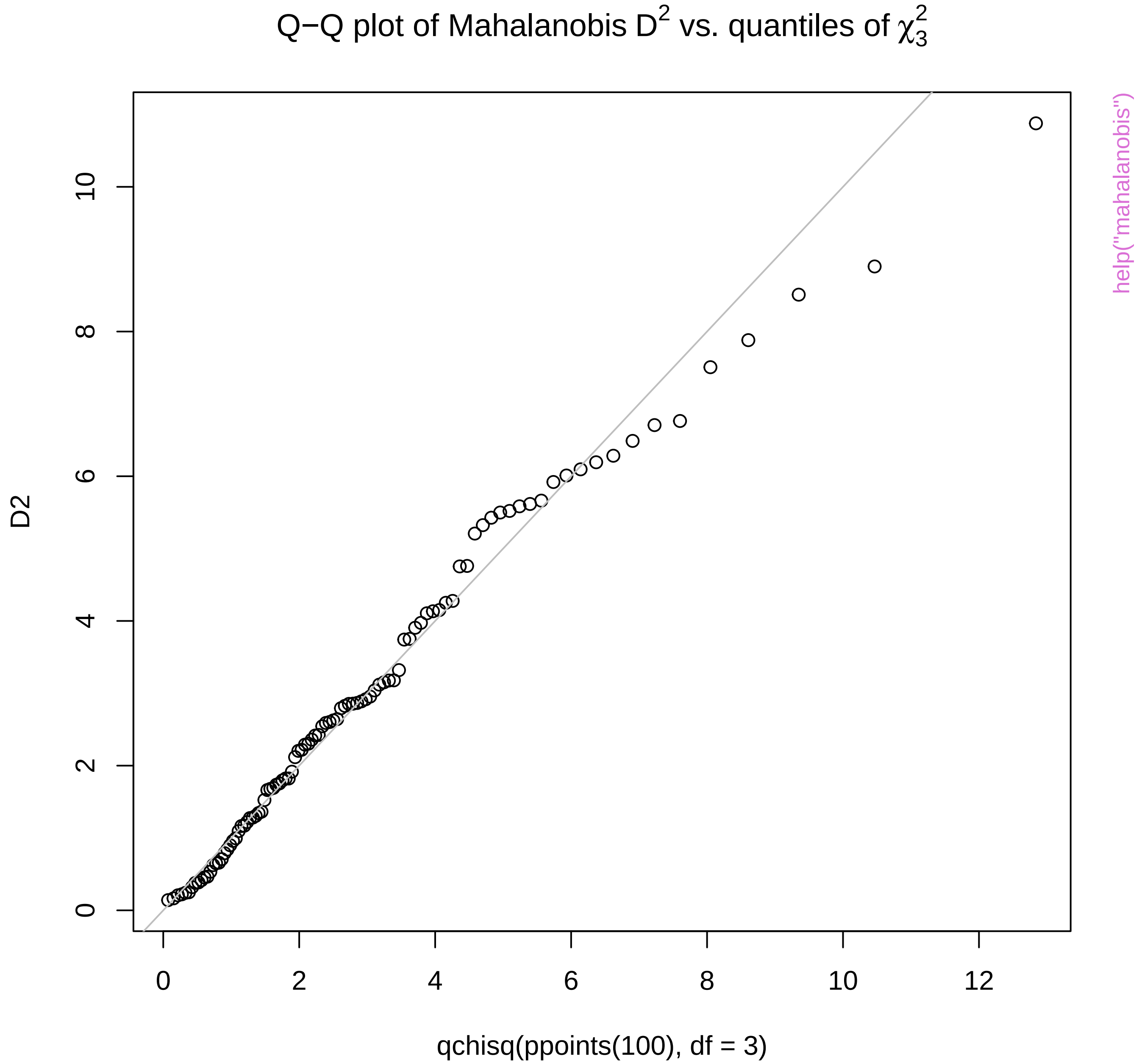

lm(weight ~ group)
Residuals vs Fitted
Residuals
Fitted values
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
help("ls.diag")
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
group :
Ctl
Trt
Factor Level Combinations



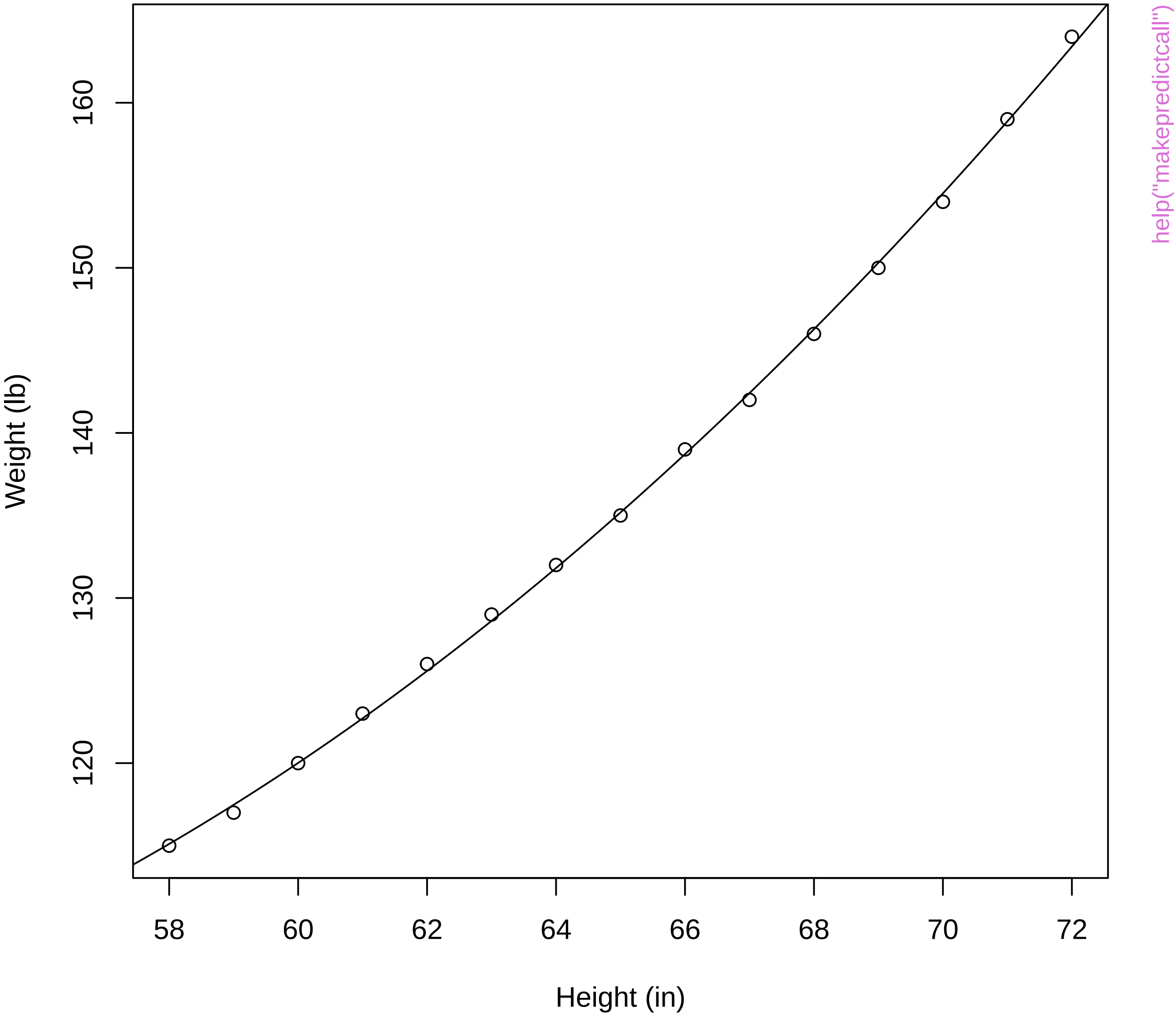
dlsD9$stud.res
dlsD9$hat
0.00
0.02
0.04
0.06
0.08
0.10
−1
0
1
2
help("ls.diag")


lm(weight ~ group)
Residuals vs Fitted
Residuals
Fitted values
1.5
1.0
0.5
0.0
−0.5
−1.0
4.7
4.8
4.9
5.0
17
15
4
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
2
1
0
−1
−2
17
15
4
help("lsfit")
Scale−Location
√|Standardized residuals|
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Fitted values
17
15
4
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
group :
Ctl
Trt
Factor Level Combinations
17
15
4

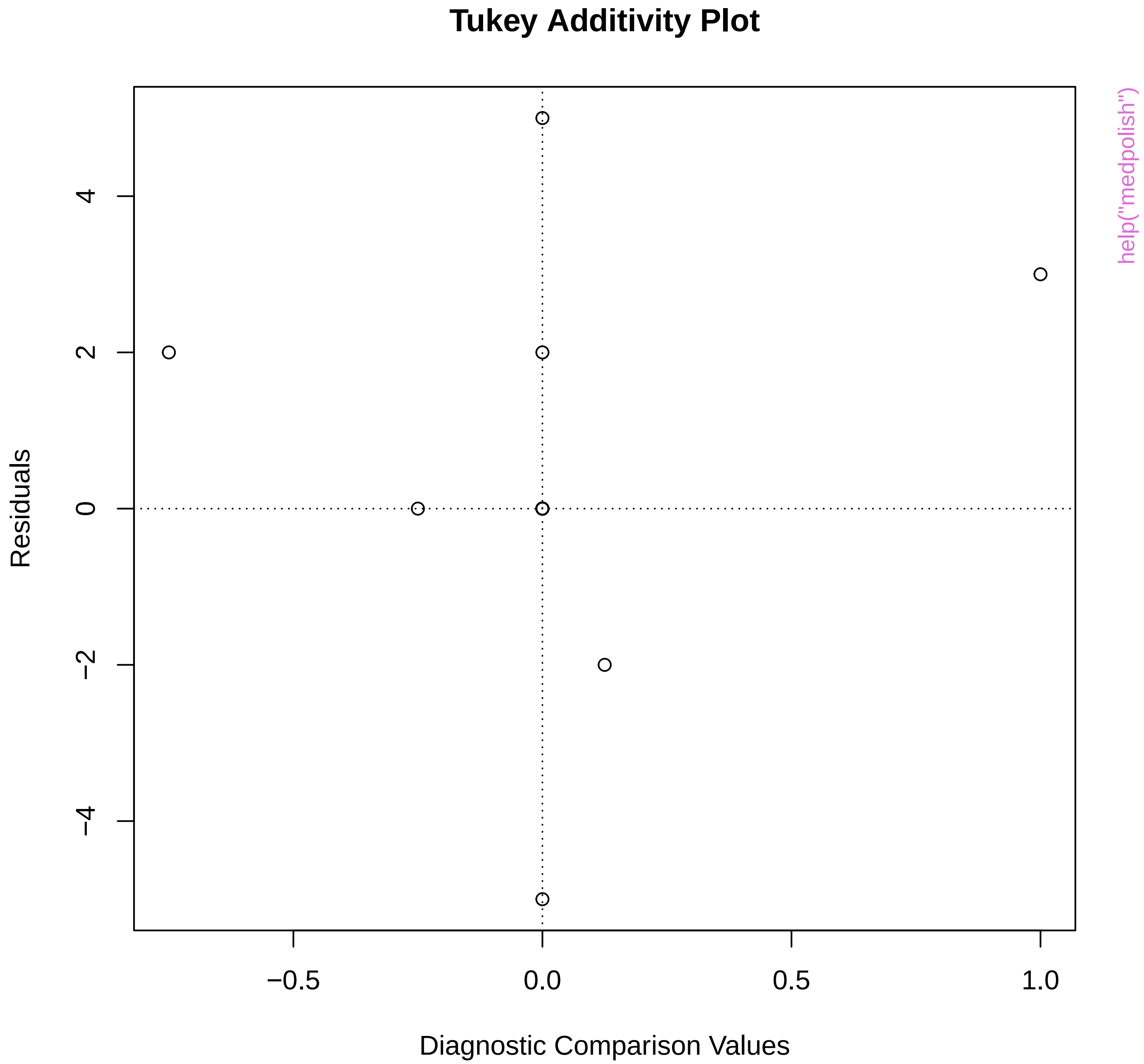

Squared Mahalanobis distances, n=100, p=3
Density
0.00
0.05
0.10
0.15
0.20
0
2
4
6
8
10
12
N = 100 Bandwidth = 0.5
help("mahalanobis")



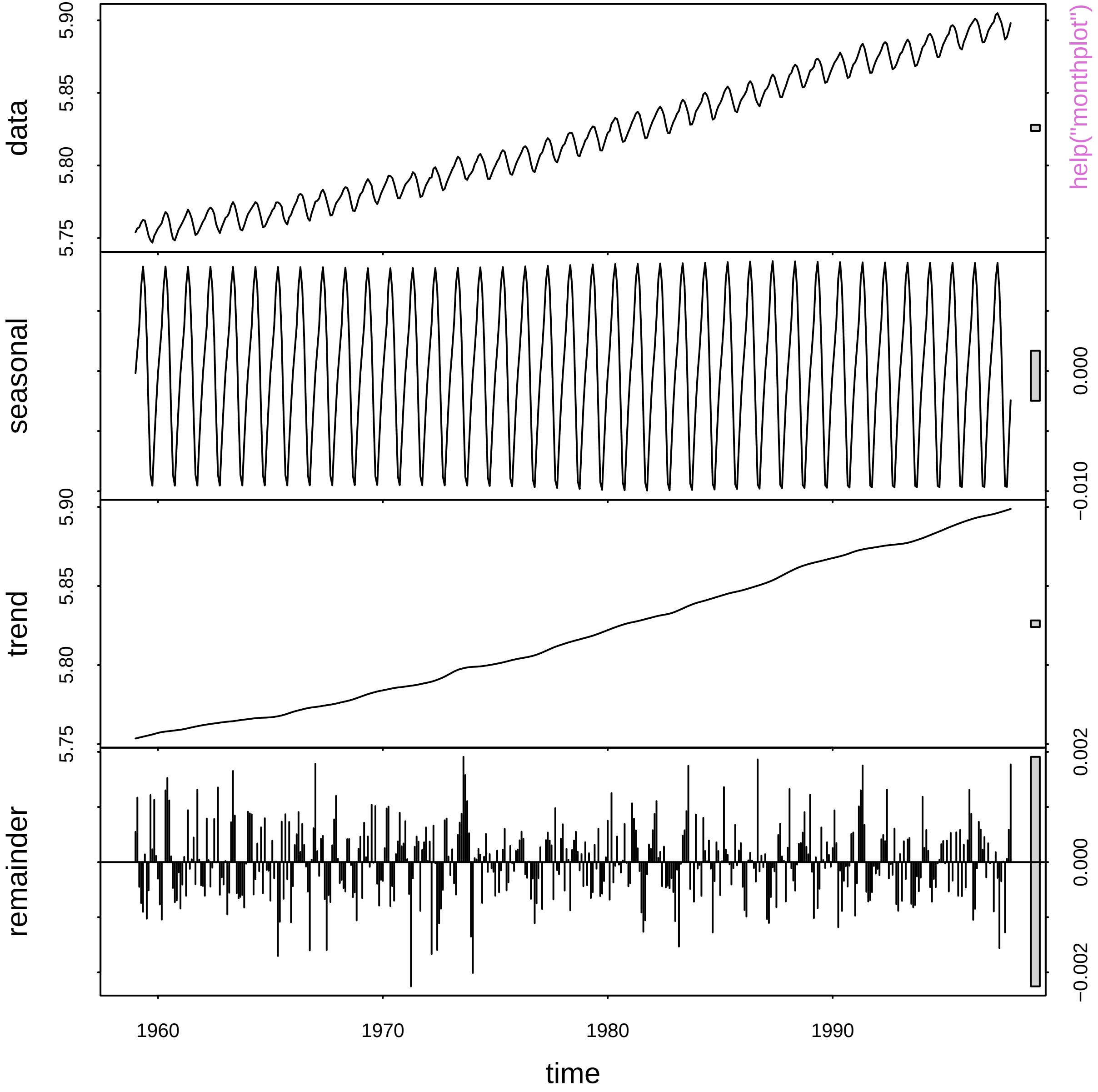
Q−Q plot of Mahalanobis D^2 vs. quantiles of χ^2_3
D2
qchisq(ppoints(100), df = 3)
0
2
4
6
8
10
12
help("mahalanobis")



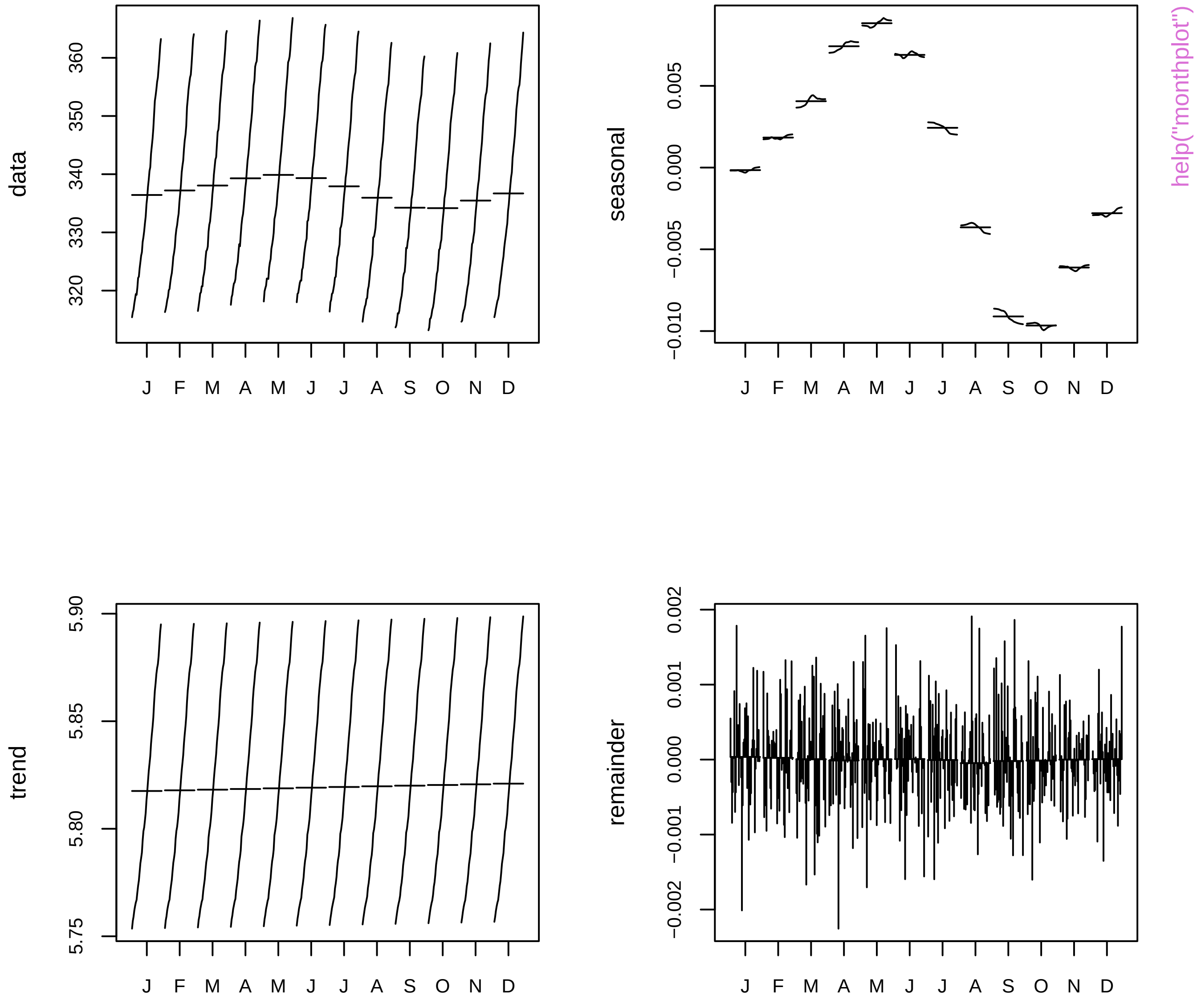
Weight (lb)
120
130
140
150
160
58
60
62
64
66
68
70
72
Height (in)
help("makepredictcall")



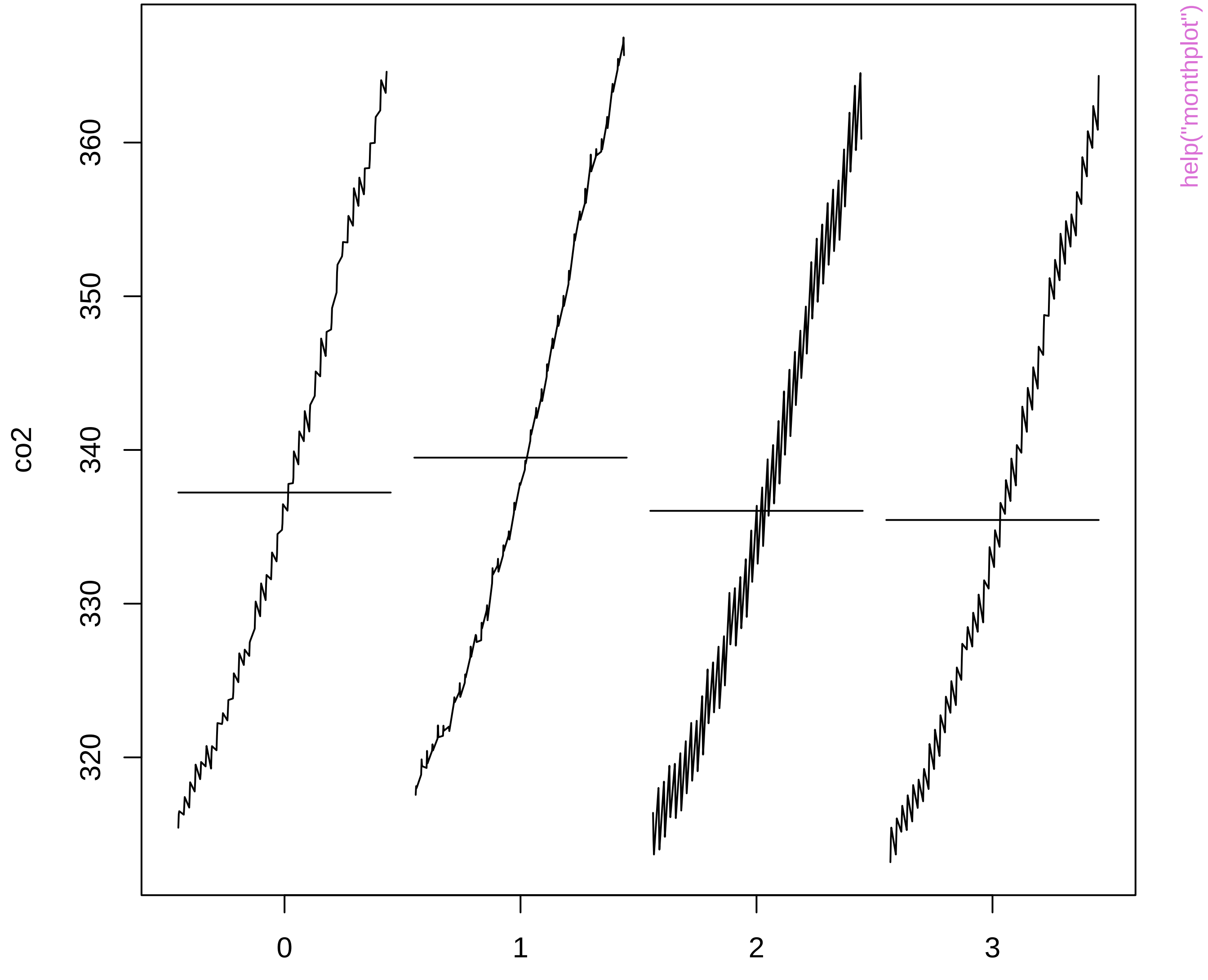
Tukey Additivity Plot
Residuals
Diagnostic Comparison Values
−0.5
0.0
0.5
1.0
−4
−2
0
2
4
help("medpolish")

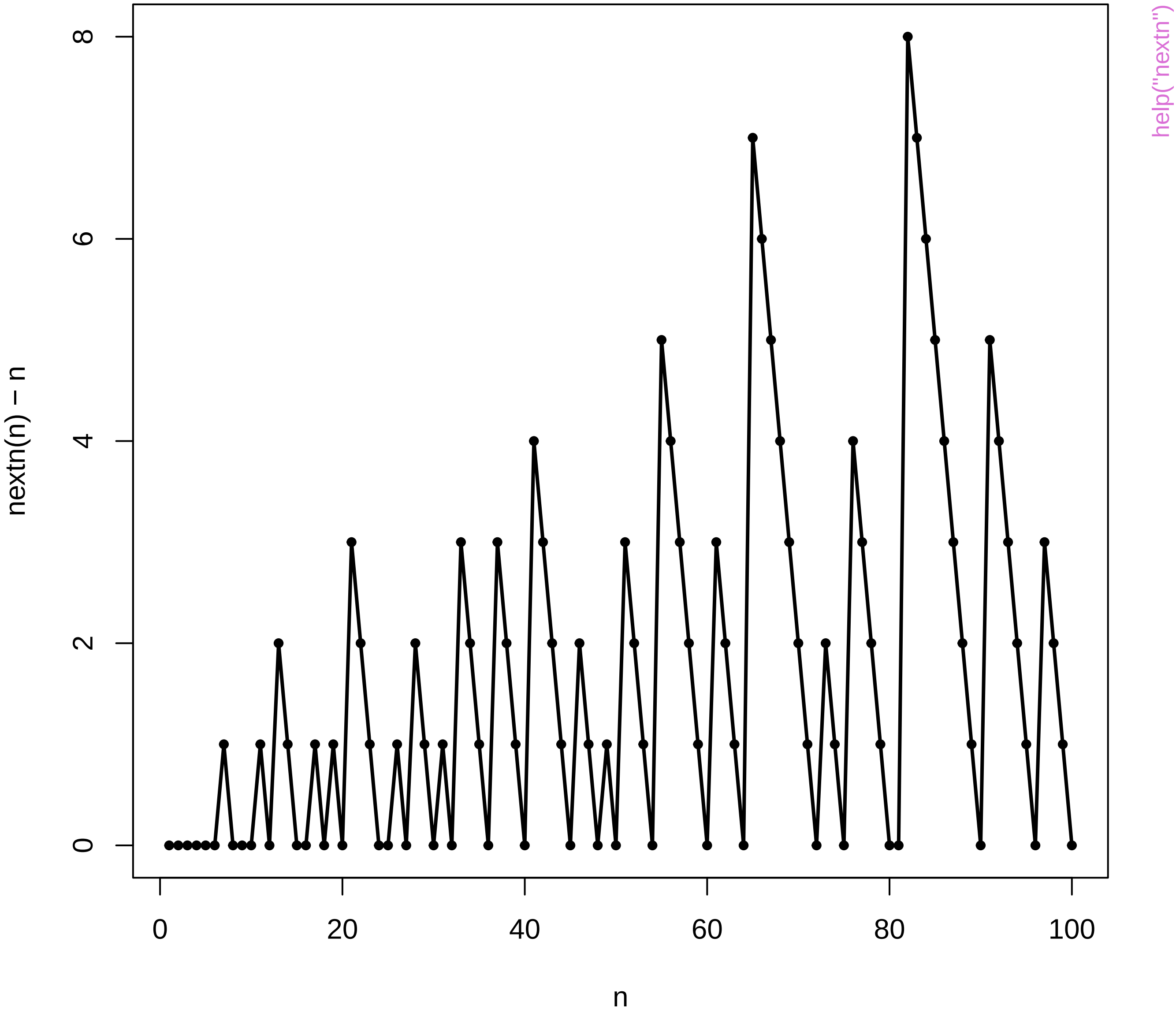


data
seasonal
trend
remainder
5.90
5.85
5.80
5.75
0.000
−0.010
0.002
−0.002
help("monthplot")
1960
1970
1980
1990
time

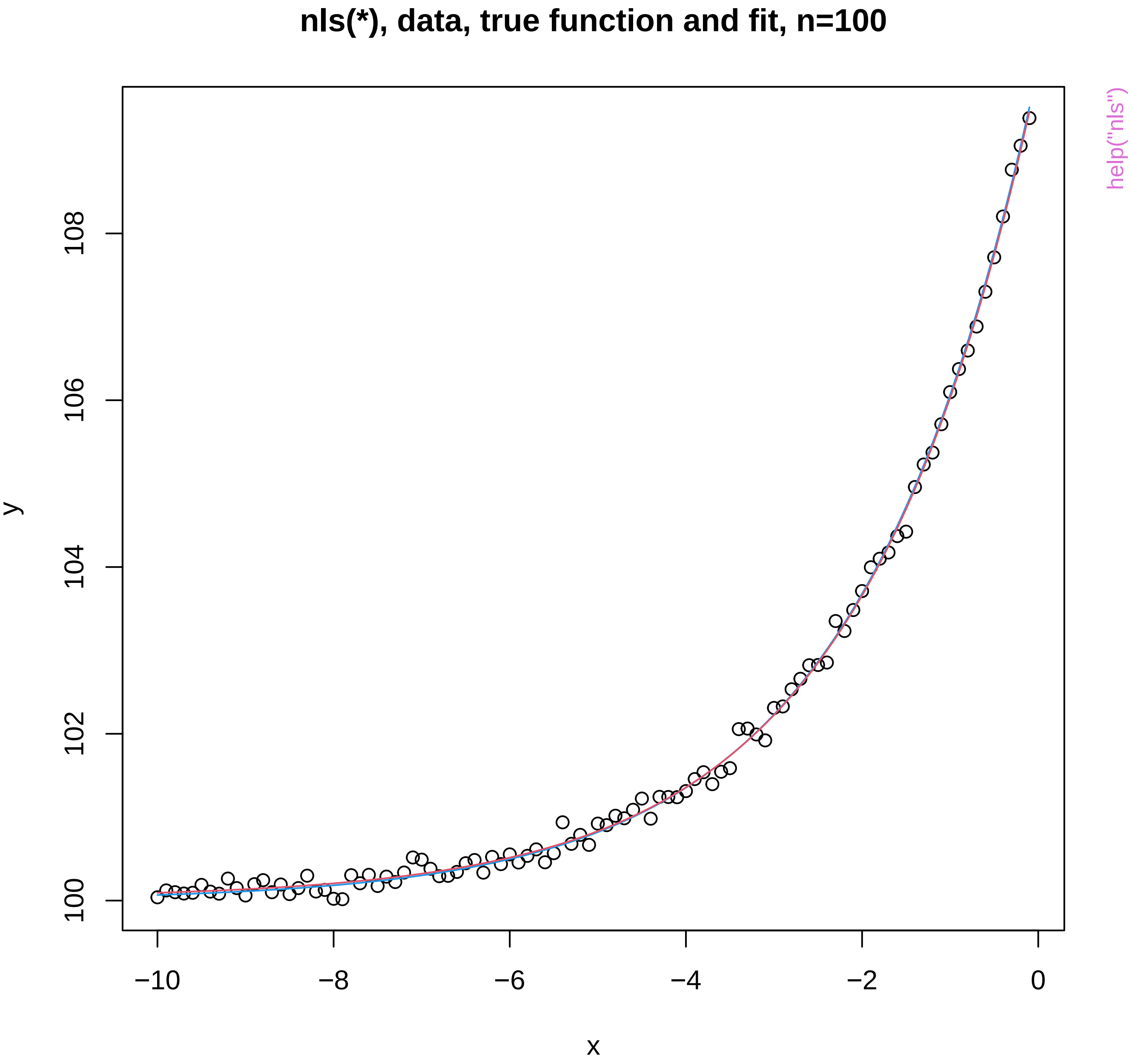


data
seasonal
trend
remainder
help("monthplot")
J F M A M J J A S O N D

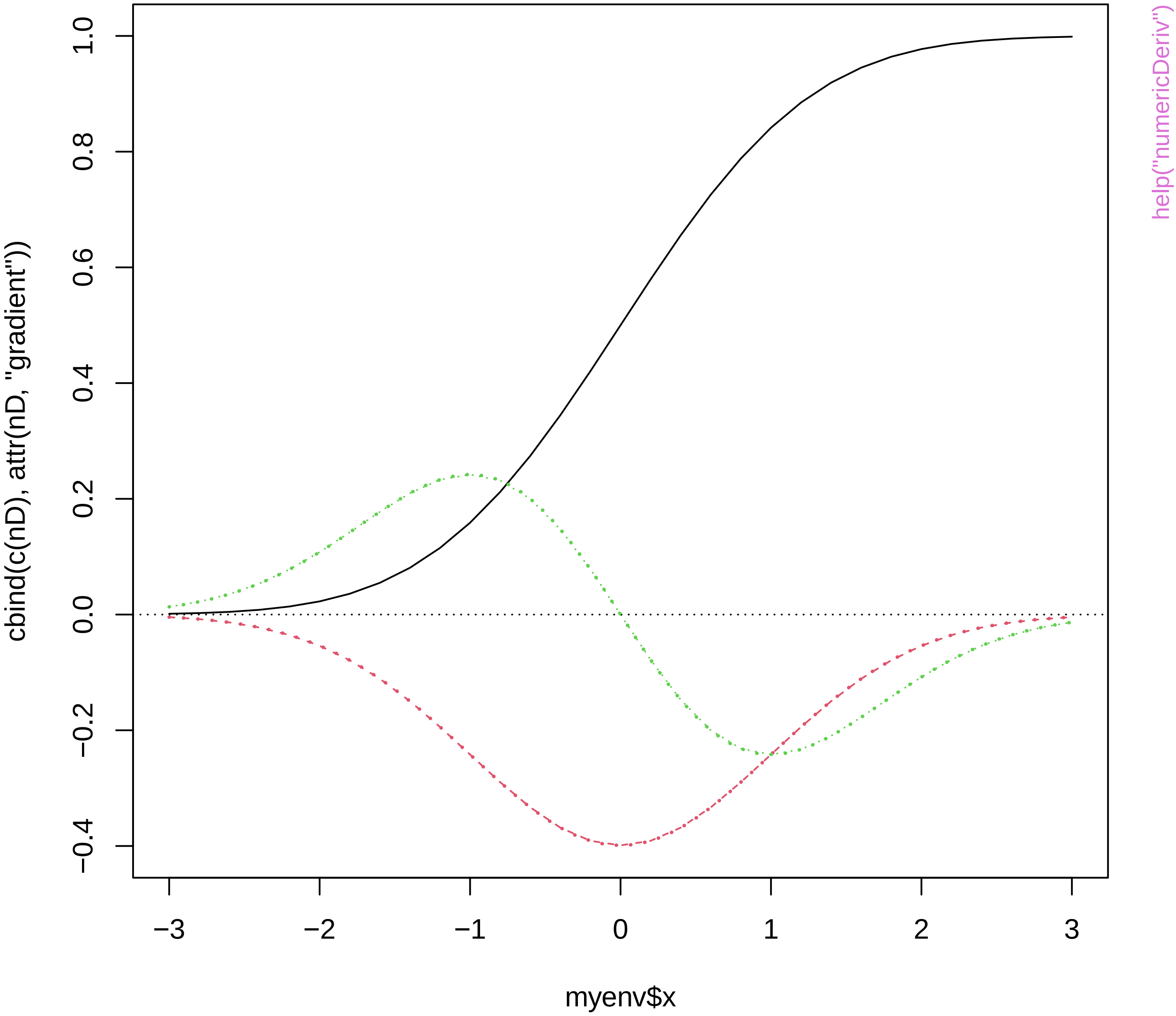

co2
320
330
340
350
360
0
1
2
3
help("monthplot")

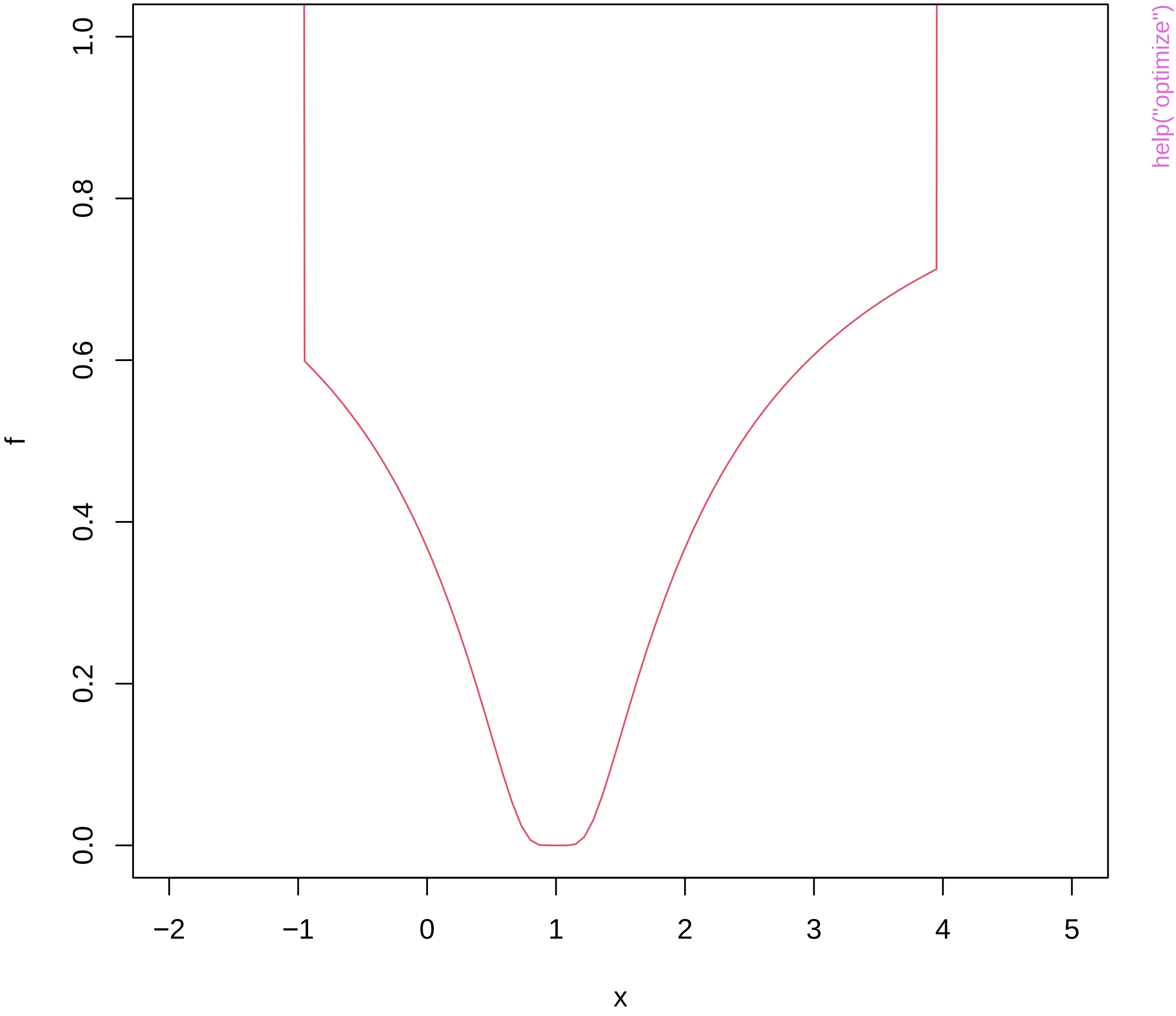


help("nextn")
nextn(n) − n
n
0
2
4
6
8
0
20
40
60
80
100

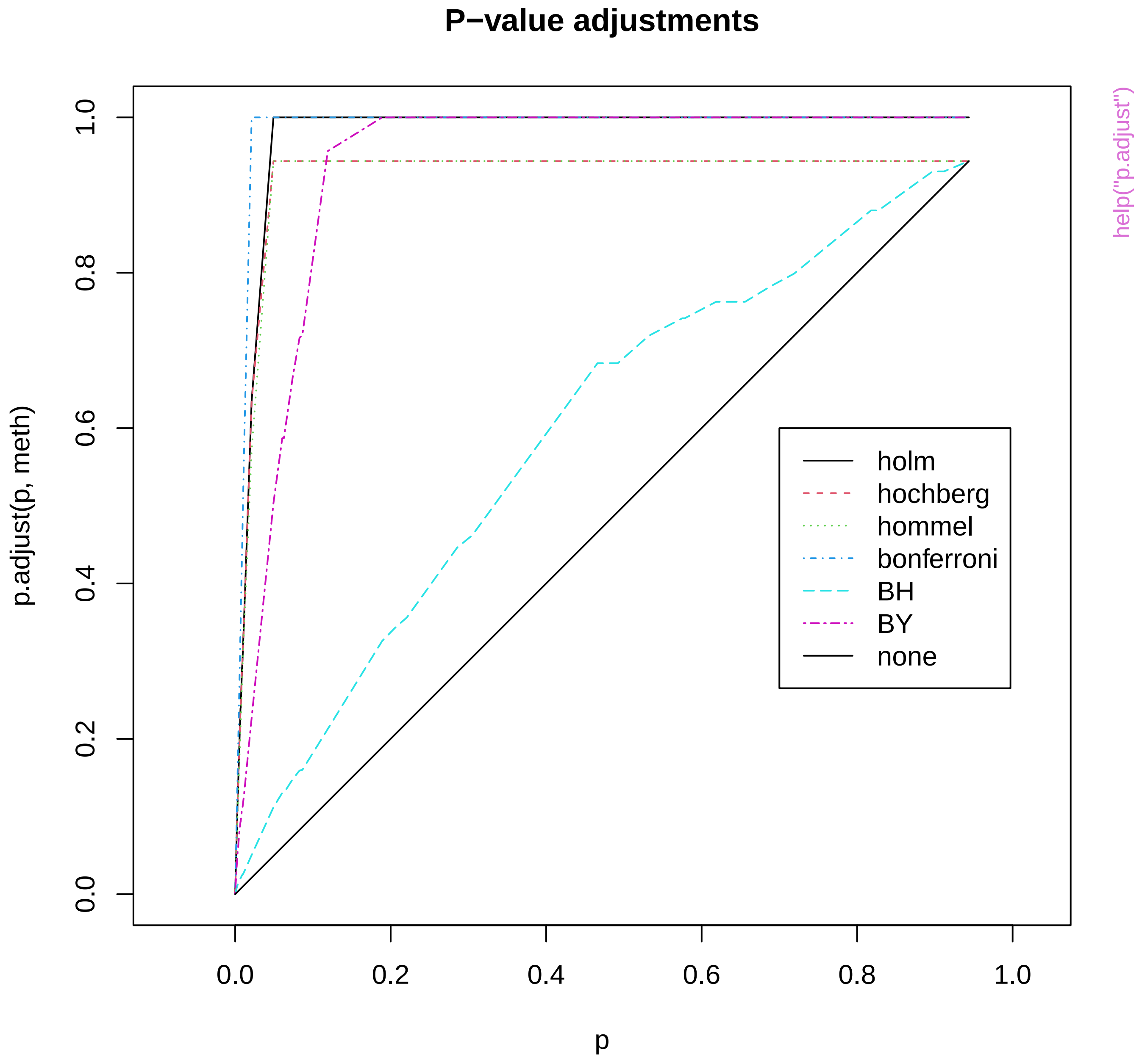


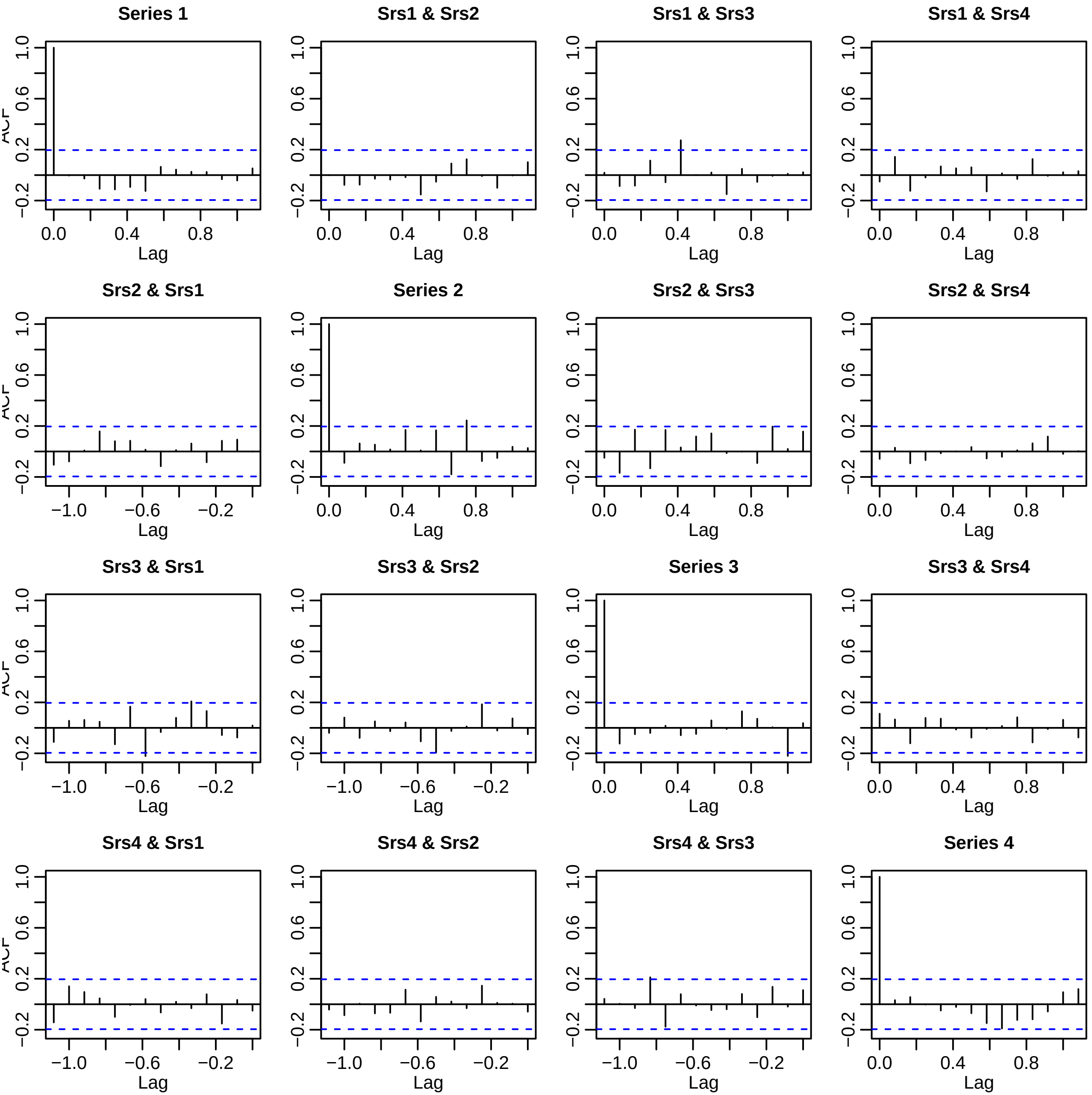
nls(*), data, true function and fit, n=100
help("nls")
y
x
100
102
104
106
108
−10
−8
−6
−4
−2
0


cbind(c(nD), attr(nD, "gradient"))
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−3
−2
−1
0
1
2
3
myenv$x
help("numericDeriv")

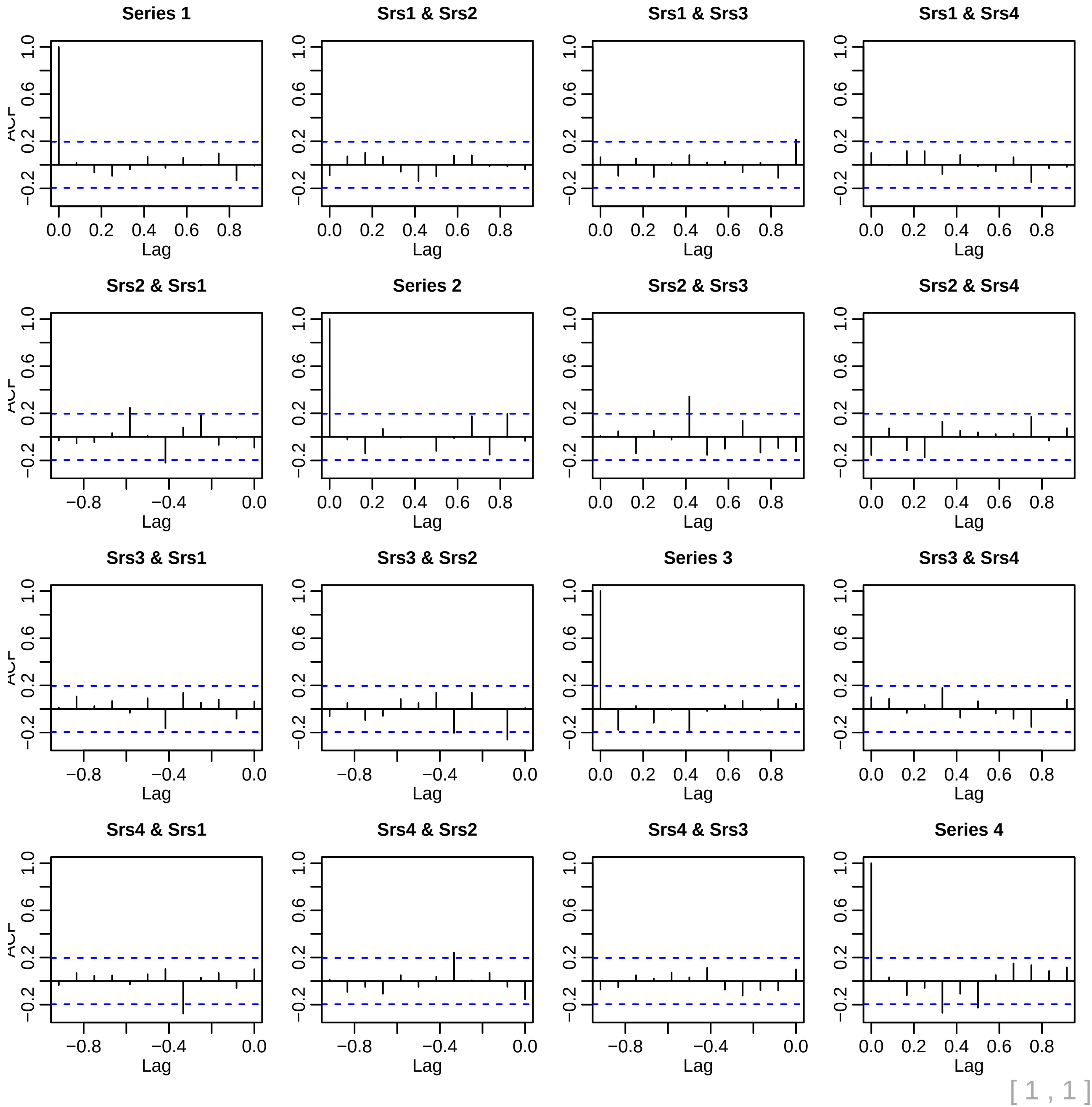


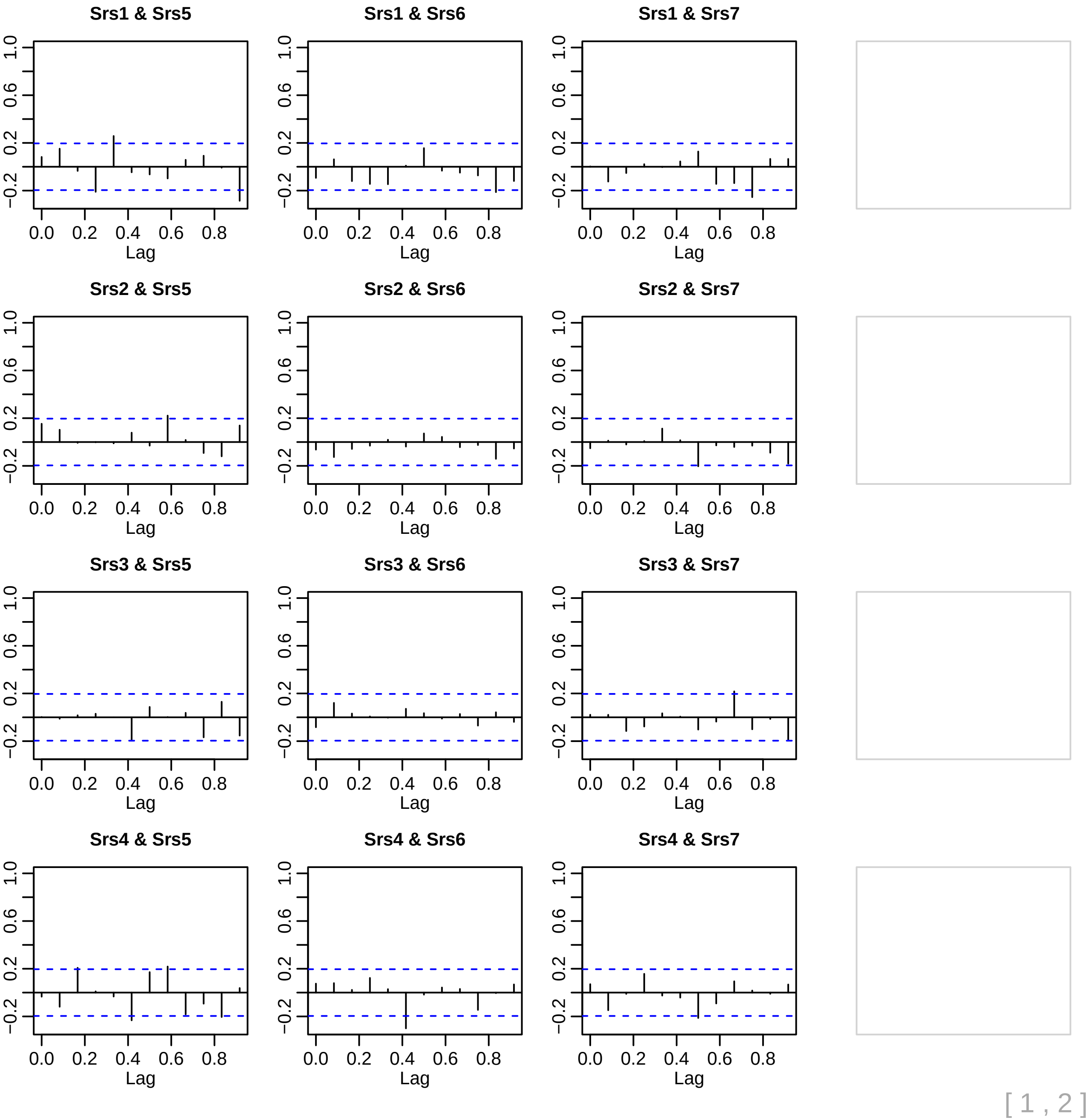
1.0
0.8
0.6
0.4
0.2
0.0
f
−2
−1
0
1
2
3
4
5
x
help("optimize")

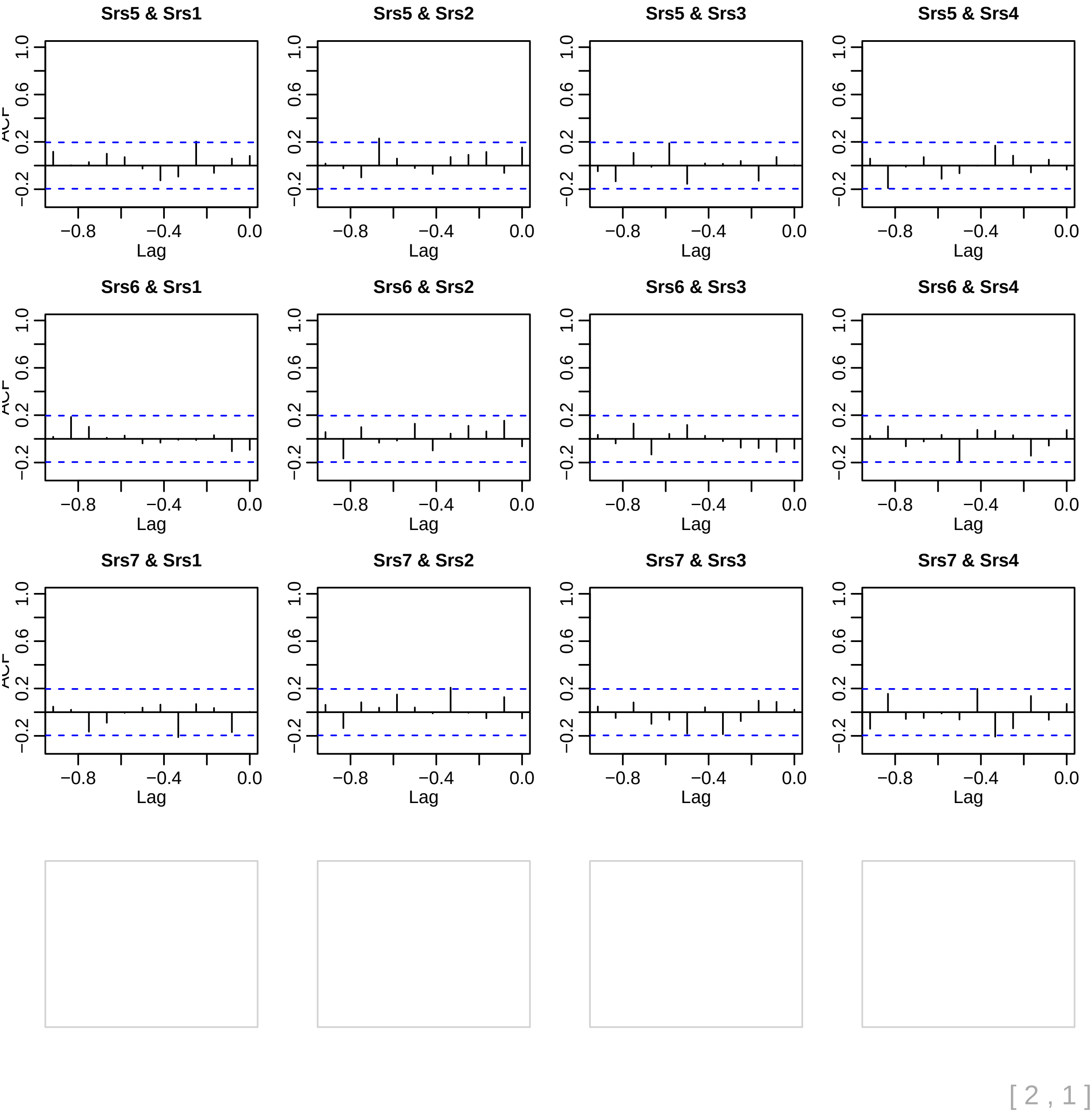

P−value adjustments
p.adjust(p, meth)
p
0.0
0.2
0.4
0.6
0.8
1.0
holm
hochberg
hommel
bonferroni
BH
BY
none
help("p.adjust")

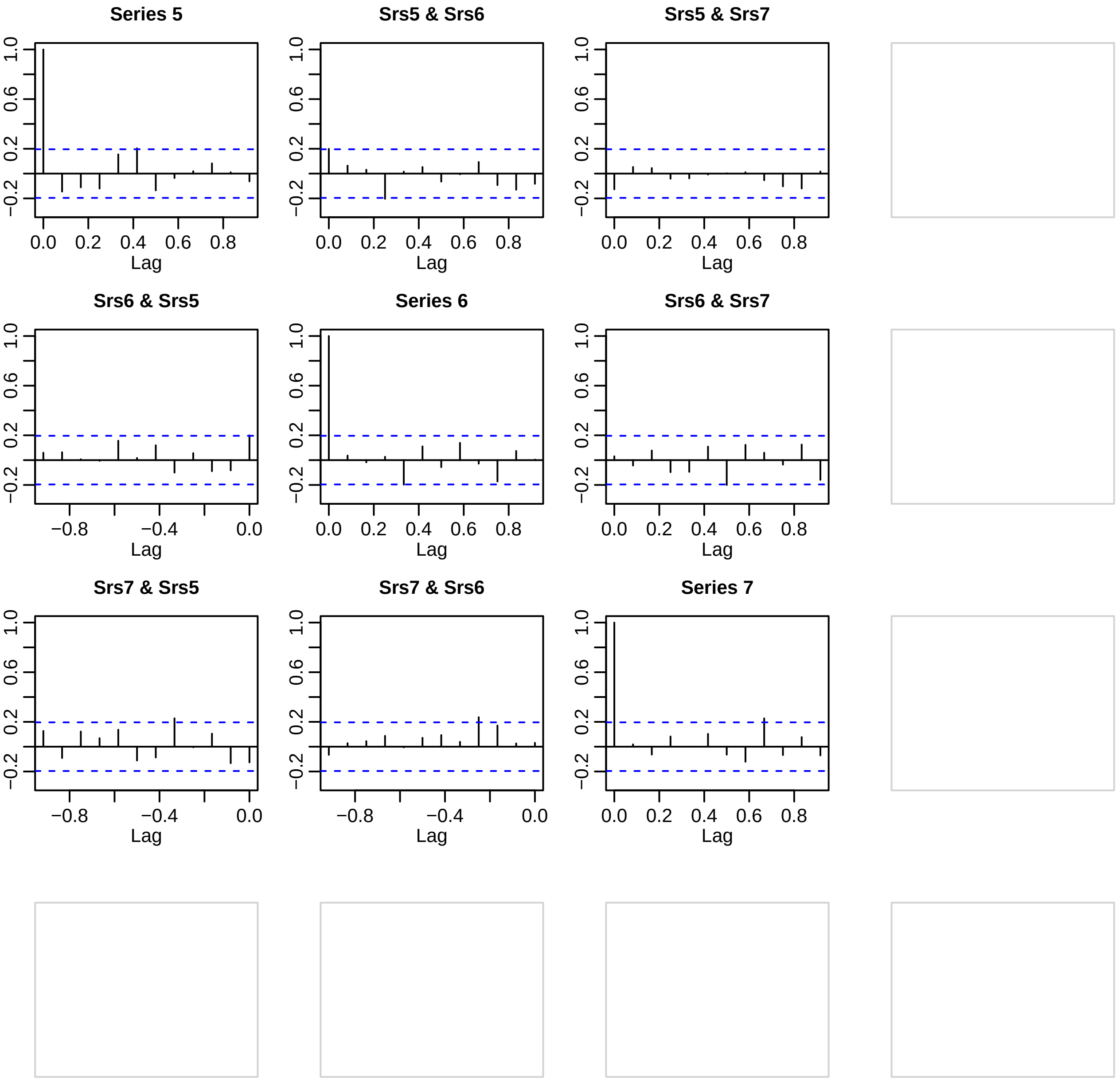

Series 1
Srs1 & Srs2
Srs1 & Srs3
Srs1 & Srs4
Srs2 & Srs1
Series 2
Srs2 & Srs3
Srs2 & Srs4
Srs3 & Srs1
Srs3 & Srs2
Series 3
Srs3 & Srs4
Srs4 & Srs1
Srs4 & Srs2
Srs4 & Srs3
Series 4
Lag
−0.2
0.2
0.6
1.0
0.0
0.4
0.8
−1.0
−0.6
−0.2

Series 1
Srs1 & Srs2
Srs1 & Srs3
Srs1 & Srs4
Srs1 & Srs5
Srs1 & Srs6
Srs1 & Srs7
Srs2 & Srs1
Series 2
Srs2 & Srs3
Srs2 & Srs4
Srs2 & Srs5
Srs2 & Srs6
Srs2 & Srs7
Srs3 & Srs1
Srs3 & Srs2
Series 3
Srs3 & Srs4
Srs3 & Srs5
Srs3 & Srs6
Srs3 & Srs7
Srs4 & Srs1
Srs4 & Srs2
Srs4 & Srs3
Series 4
Srs4 & Srs5
Srs4 & Srs6
Srs4 & Srs7
Srs5 & Srs1
Srs5 & Srs2
Srs5 & Srs3
Srs5 & Srs4
Series 5
Srs5 & Srs6
Srs5 & Srs7
Srs6 & Srs1
Srs6 & Srs2
Srs6 & Srs3
Srs6 & Srs4
Srs6 & Srs5
Series 6
Srs6 & Srs7
Srs7 & Srs1
Srs7 & Srs2
Srs7 & Srs3
Srs7 & Srs4
Srs7 & Srs5
Srs7 & Srs6
Series 7
ACF
−0.2
0.2
0.0
0.6
−0.8
Lag


Series 1
Srs1 & Srs2
Srs1 & Srs3
Srs1 & Srs4
Srs2 & Srs1
Series 2
Srs2 & Srs3
Srs2 & Srs4
Srs3 & Srs1
Srs3 & Srs2
Series 3
Srs3 & Srs4
Srs4 & Srs1
Srs4 & Srs2
Srs4 & Srs3
Series 4
Lag
[1 , 1]

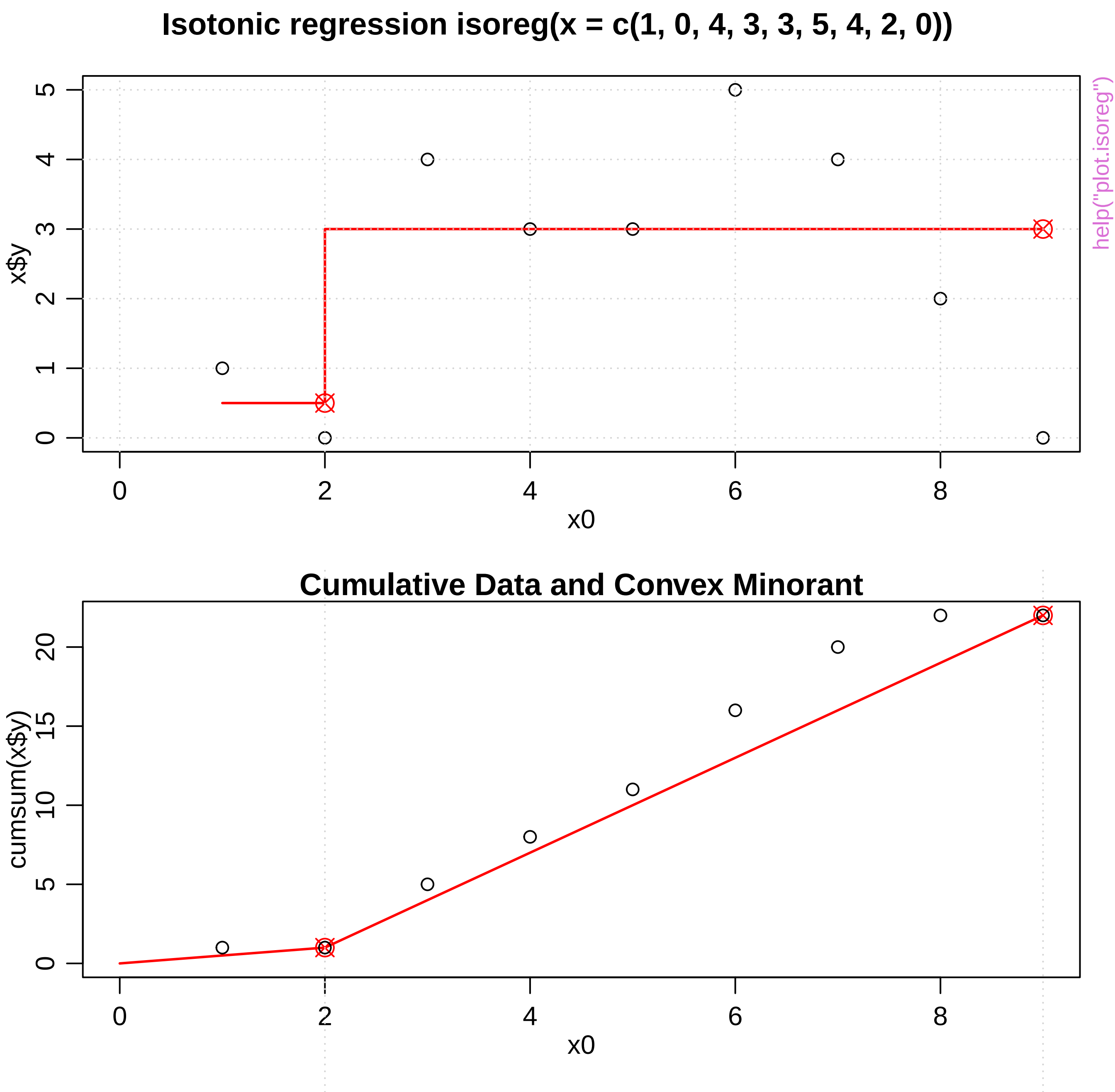

Srs1 & Srs5
Srs1 & Srs6
Srs1 & Srs7
Srs2 & Srs5
Srs2 & Srs6
Srs2 & Srs7
Srs3 & Srs5
Srs3 & Srs6
Srs3 & Srs7
Srs4 & Srs5
Srs4 & Srs6
Srs4 & Srs7
Lag
[1 , 2]

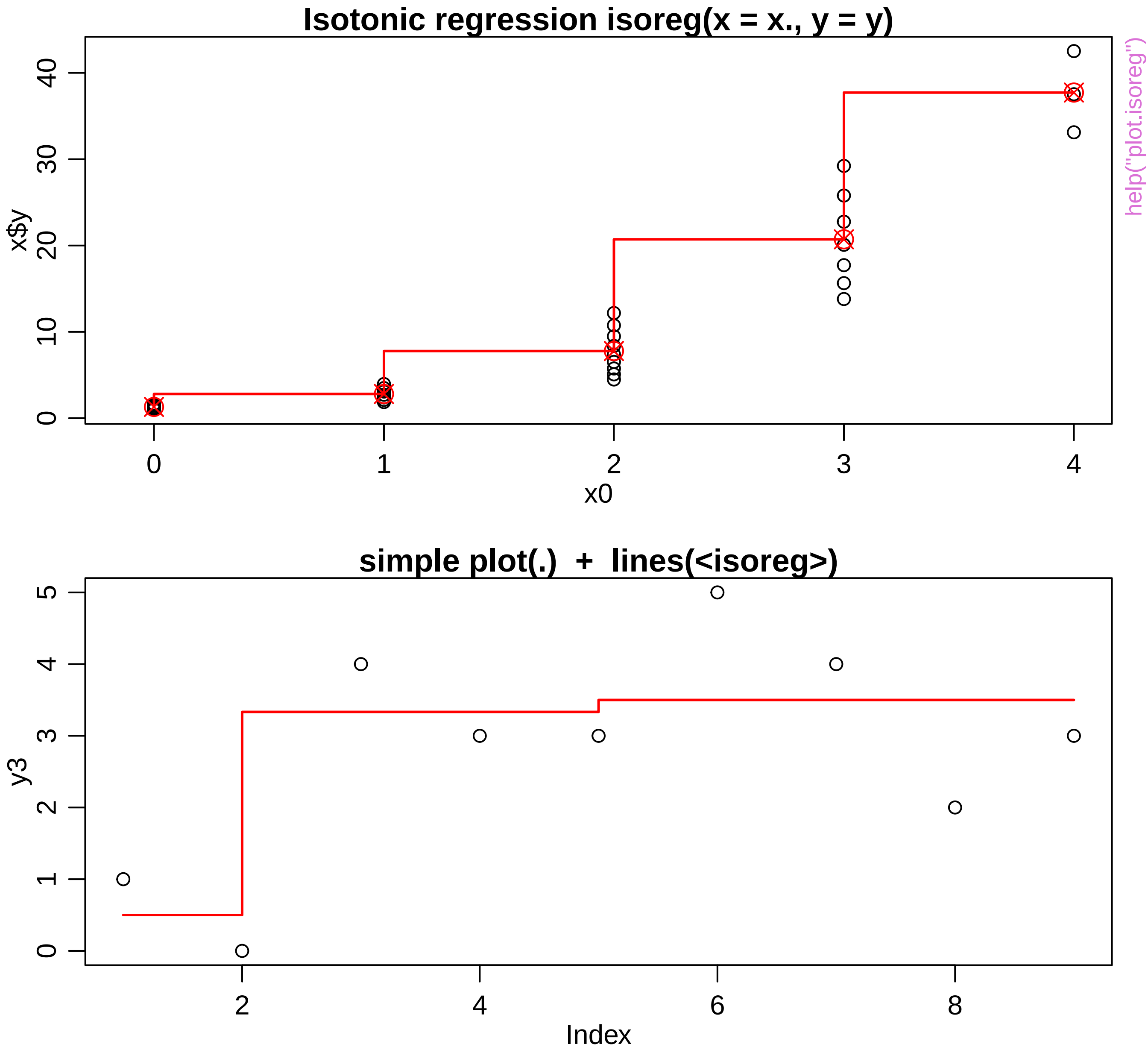

Srs5 & Srs1
Srs5 & Srs2
Srs5 & Srs3
Srs5 & Srs4
Srs6 & Srs1
Srs6 & Srs2
Srs6 & Srs3
Srs6 & Srs4
Srs7 & Srs1
Srs7 & Srs2
Srs7 & Srs3
Srs7 & Srs4
ACF
1.0
0.6
0.2
−0.2
−0.8
−0.4
0.0
Lag
[2 , 1]

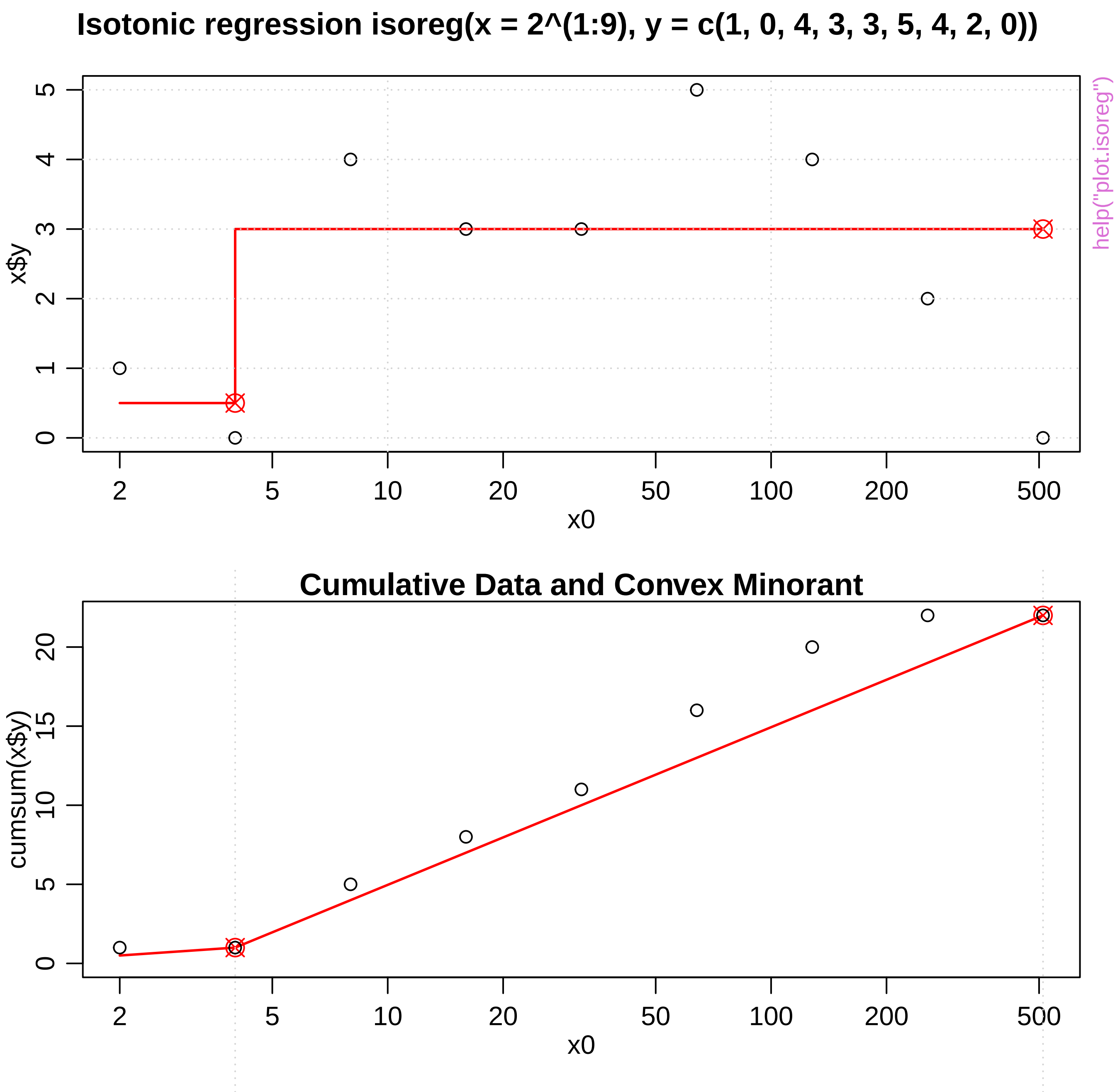


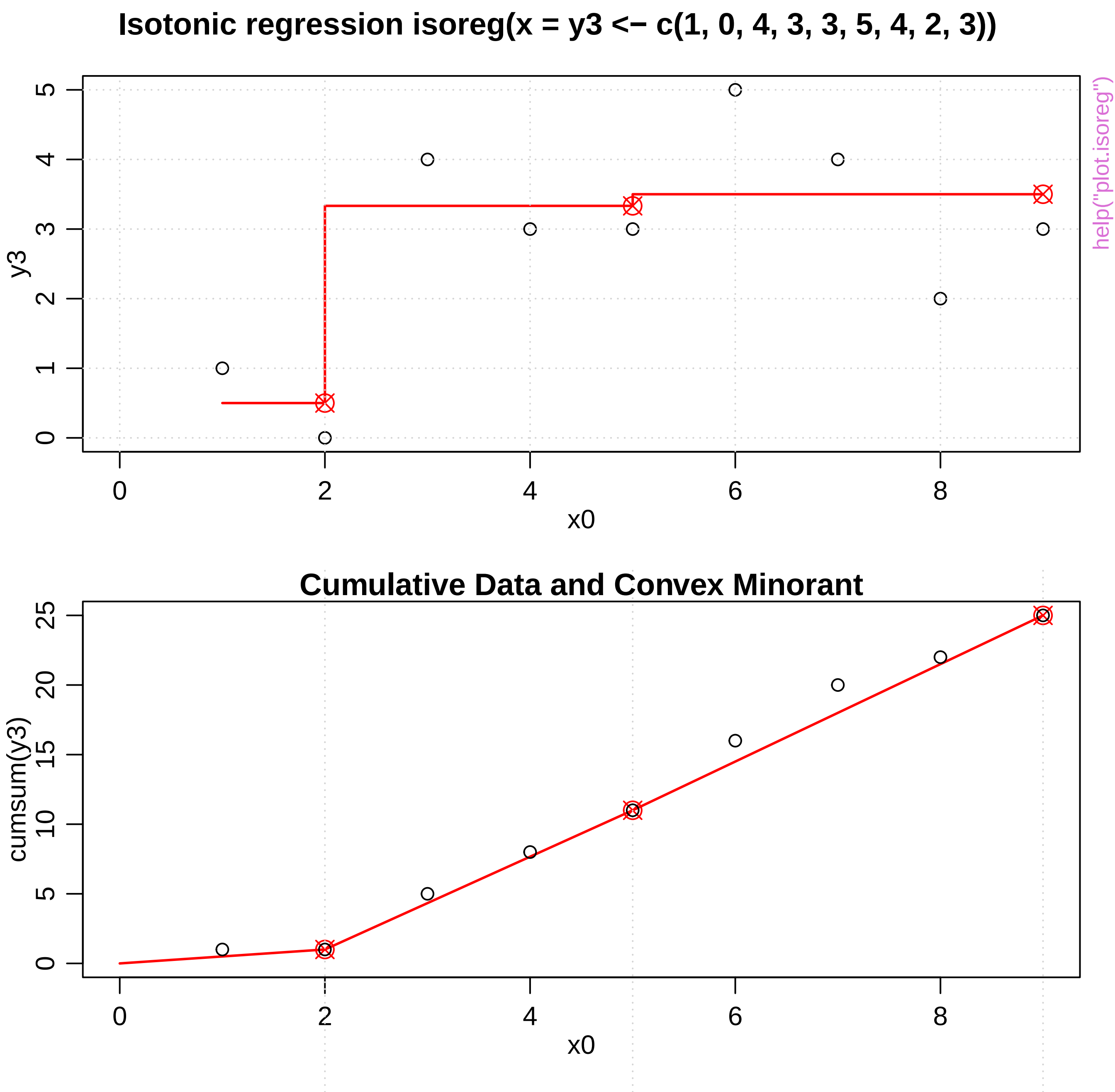
Series 5
Srs5 & Srs6
Srs5 & Srs7
Srs6 & Srs5
Series 6
Srs6 & Srs7
Srs7 & Srs5
Srs7 & Srs6
Series 7
Lag

[2 , 2]

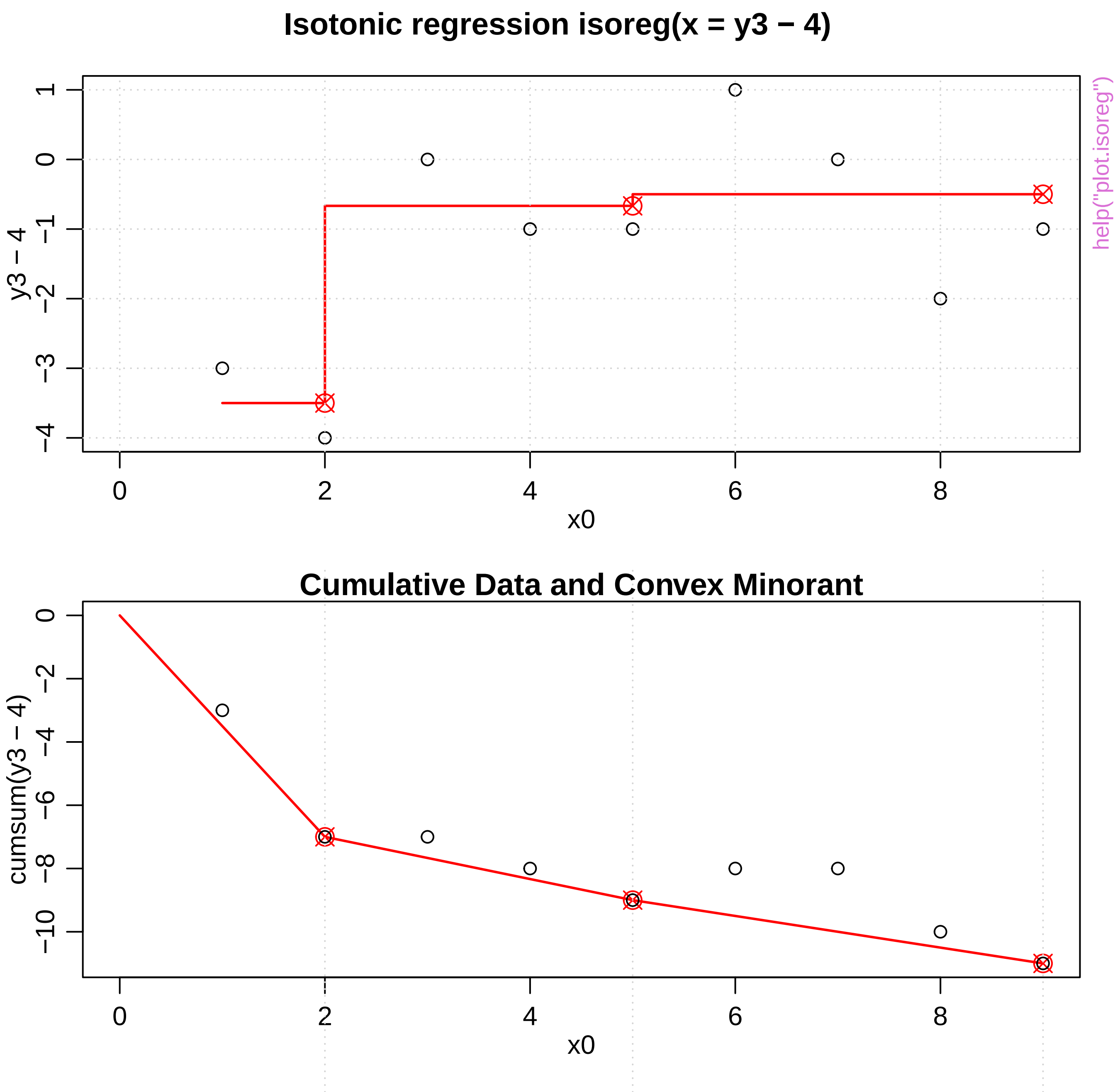

Isotonic regression isoreg(x = c(1, 0, 4, 3, 3, 5, 4, 2, 0))
x$y
x0
help("plot.isoreg")
Cumulative Data and Convex Minorant
cumsum(x$y)
x0

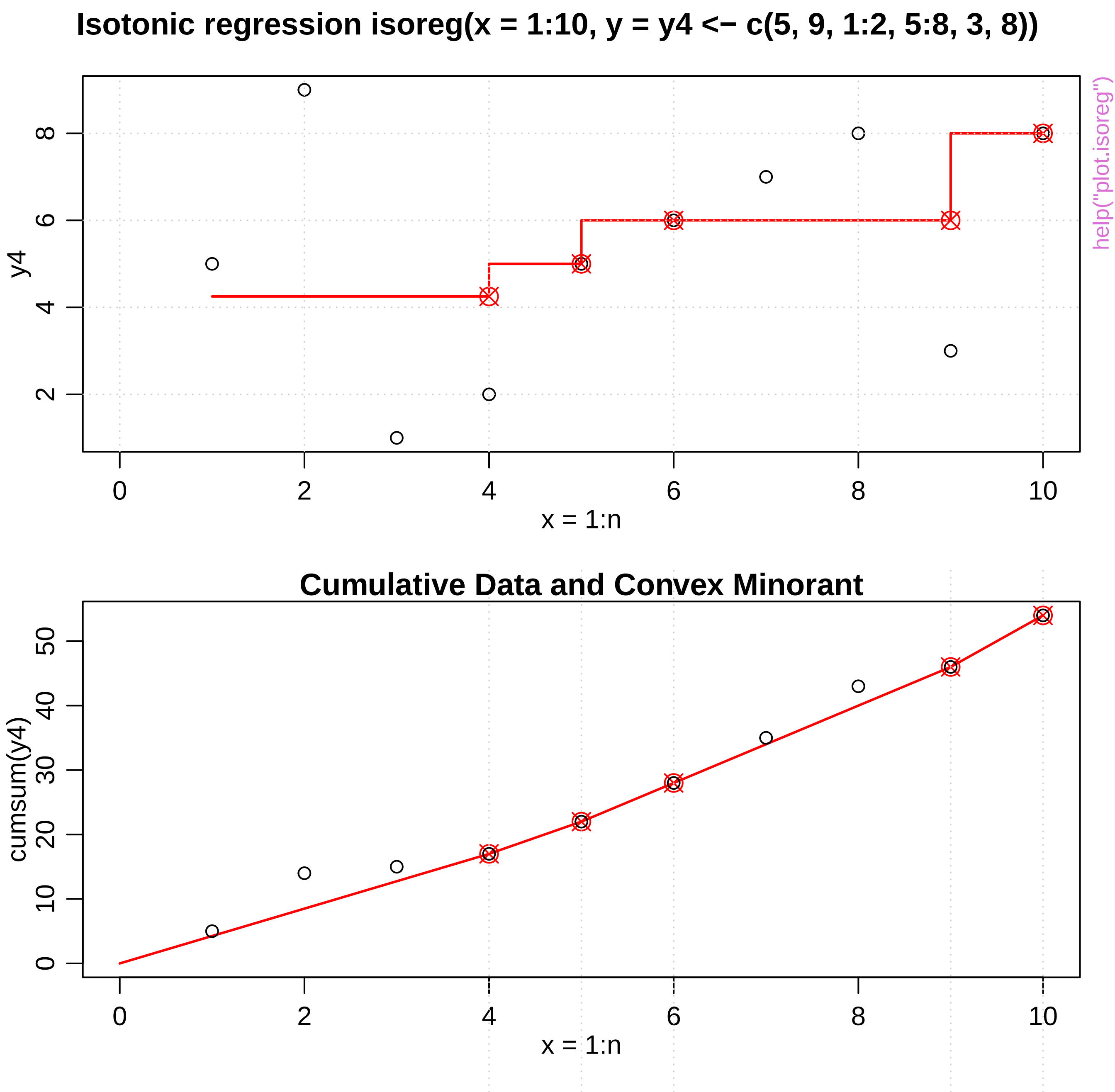

Isotonic regression isoreg(x = x., y = y)
x$y
0
10
20
30
40
0
1
2
3
4
x0
help("plot.isoreg")
simple plot(.) + lines(<isoreg>)
y3
0
1
2
3
4
5
2
4
6
8
Index

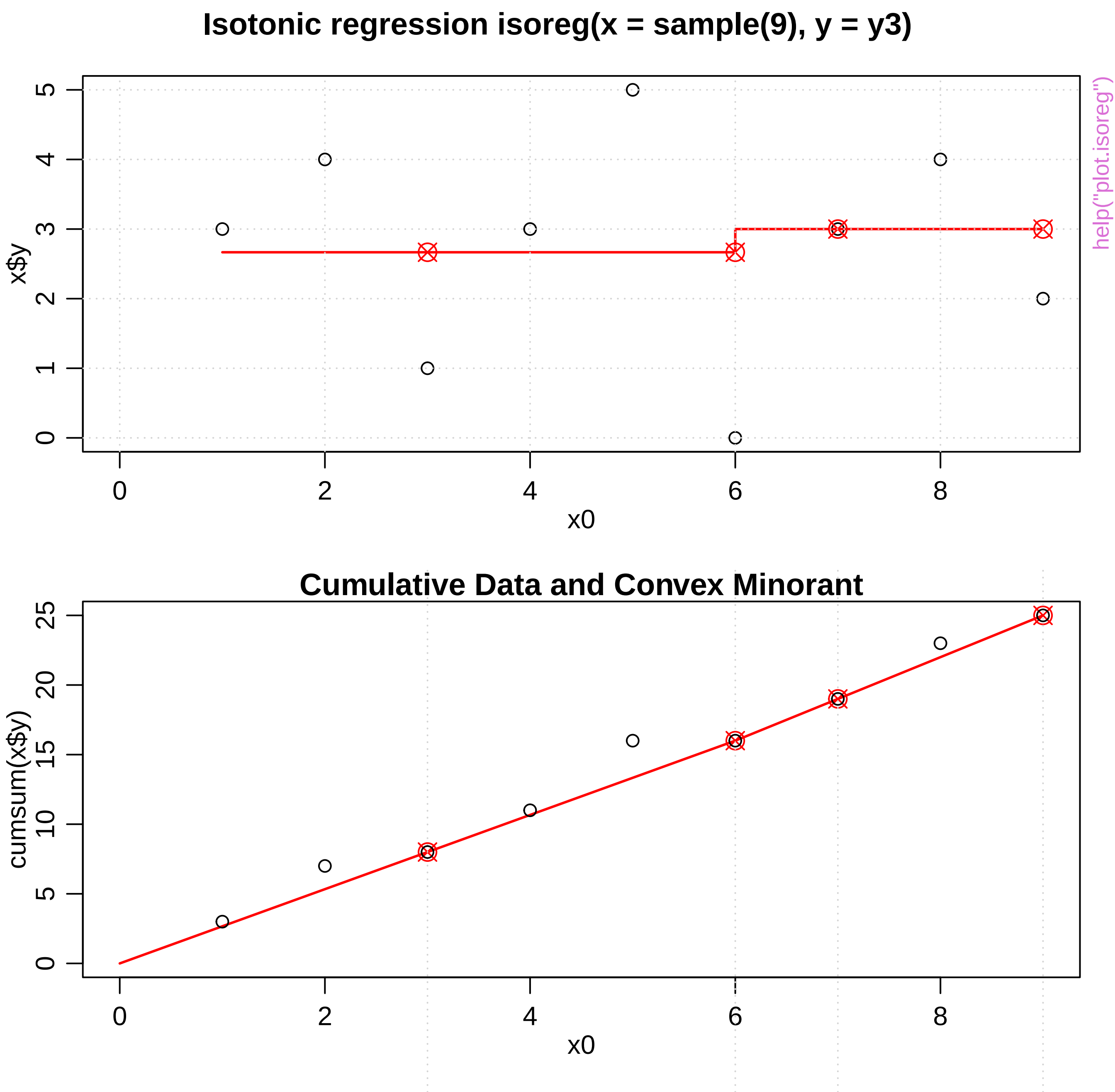

Isotonic regression isoreg(x = 2^(1:9), y = c(1, 0, 4, 3, 3, 5, 4, 2, 0))
x$y
x0
help("plot.isoreg")
Cumulative Data and Convex Minorant
cumsum(x$y)
x0



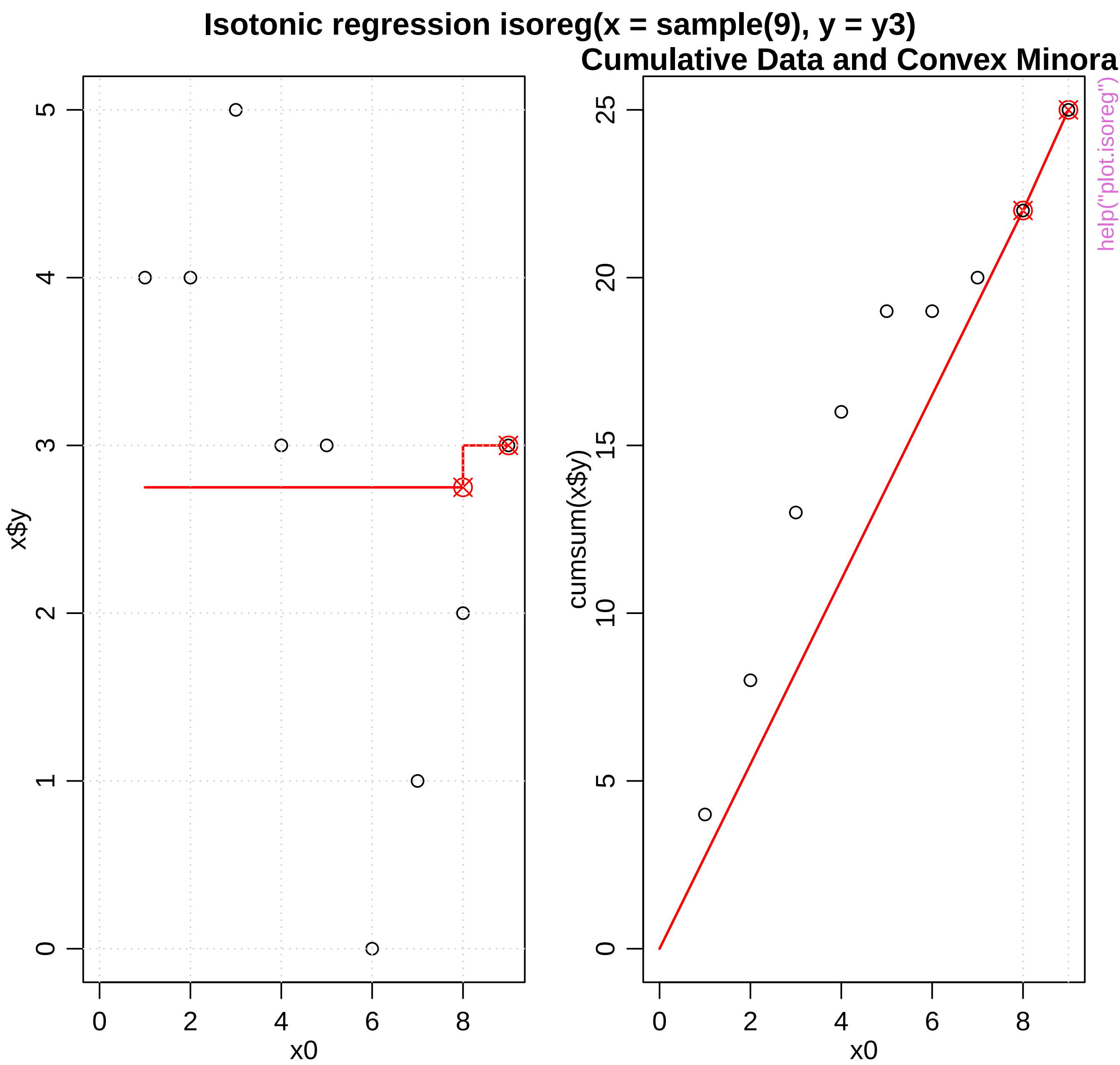
Isotonic regression isoreg(x = y3 <− c(1, 0, 4, 3, 3, 5, 4, 2, 3))
y3
0
1
2
3
4
5
0
2
4
6
8
x0
help("plot.isoreg")
Cumulative Data and Convex Minorant
cumsum(y3)
0
5
10
15
20
25
0
2
4
6
8
x0

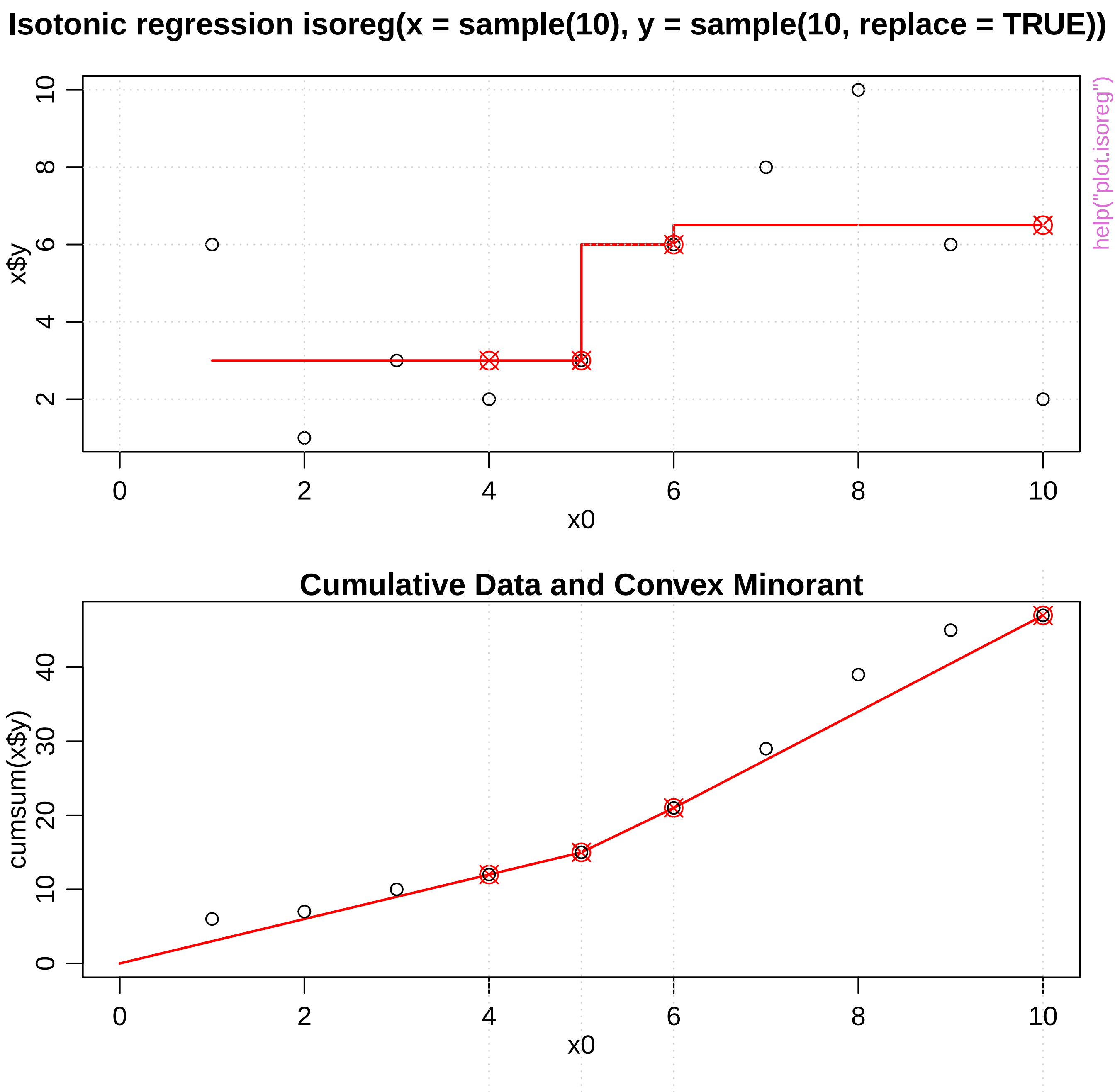

Isotonic regression isoreg(x = y3 − 4)
y3 − 4
1
0
−1
−2
−3
−4
0
2
4
6
8
x0
help("plot.isoreg")
Cumulative Data and Convex Minorant
cumsum(y3 − 4)
0
−2
−4
−6
−8
−10
0
2
4
6
8
x0

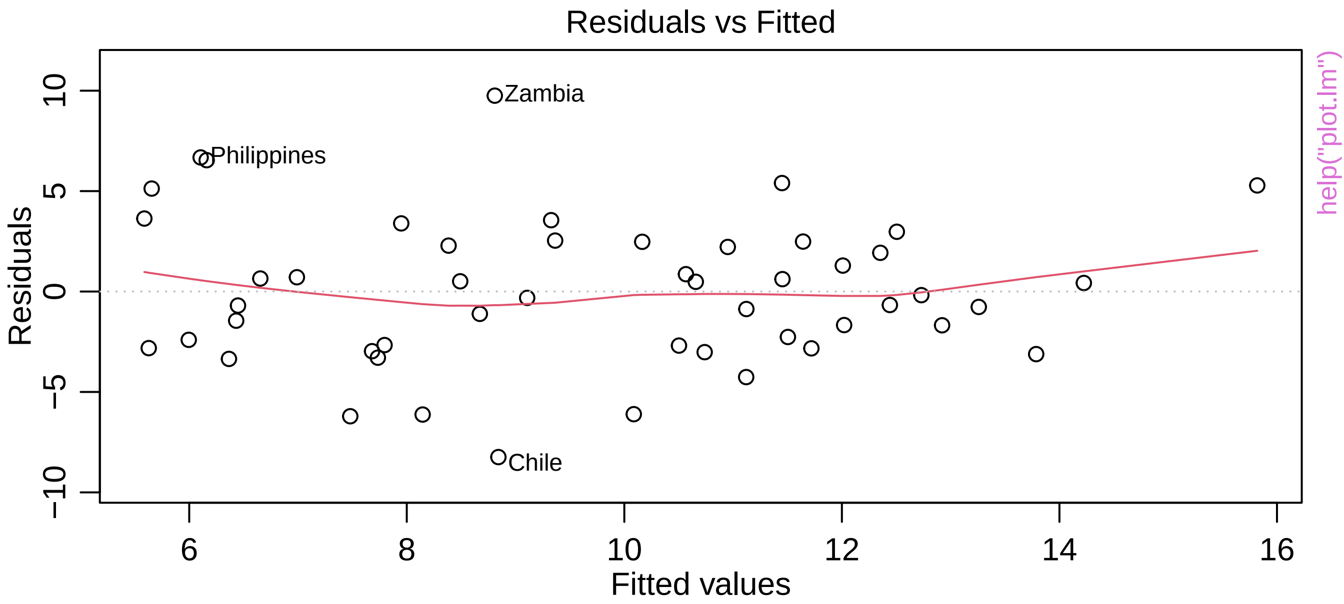

Isotonic regression isoreg(x = 1:10, y = y4 <− c(5, 9, 1:2, 5:8, 3, 8))
y4
x = 1:n
help("plot.isoreg")
Cumulative Data and Convex Minorant
cumsum(y4)
x = 1:n

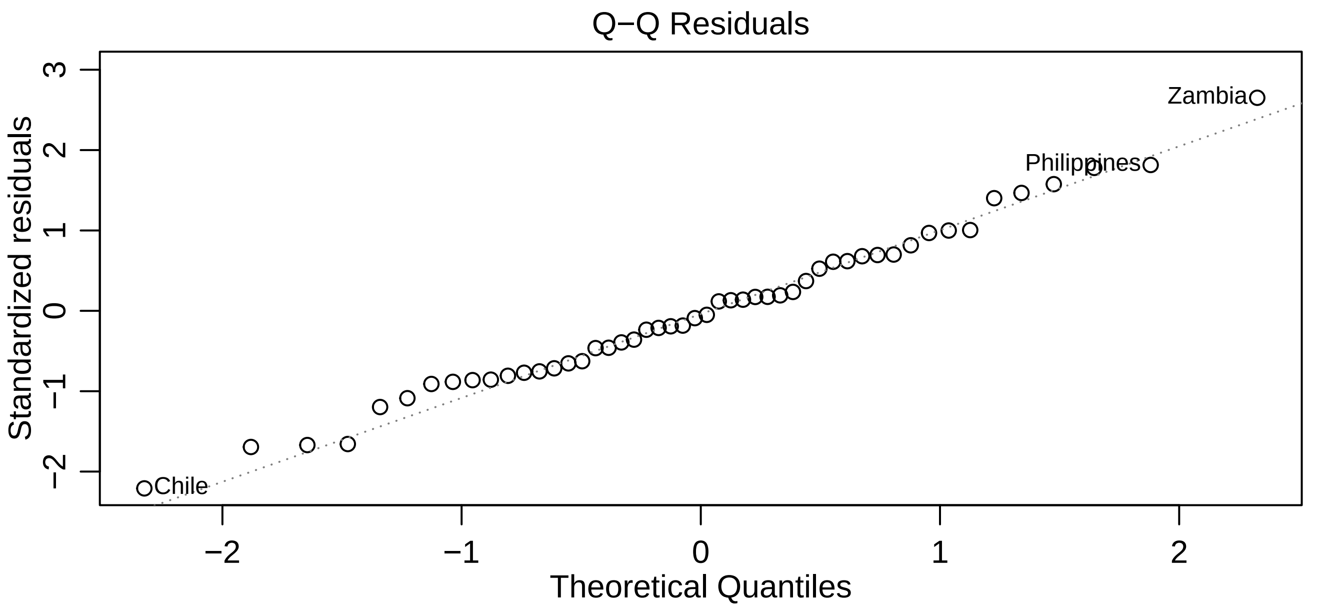


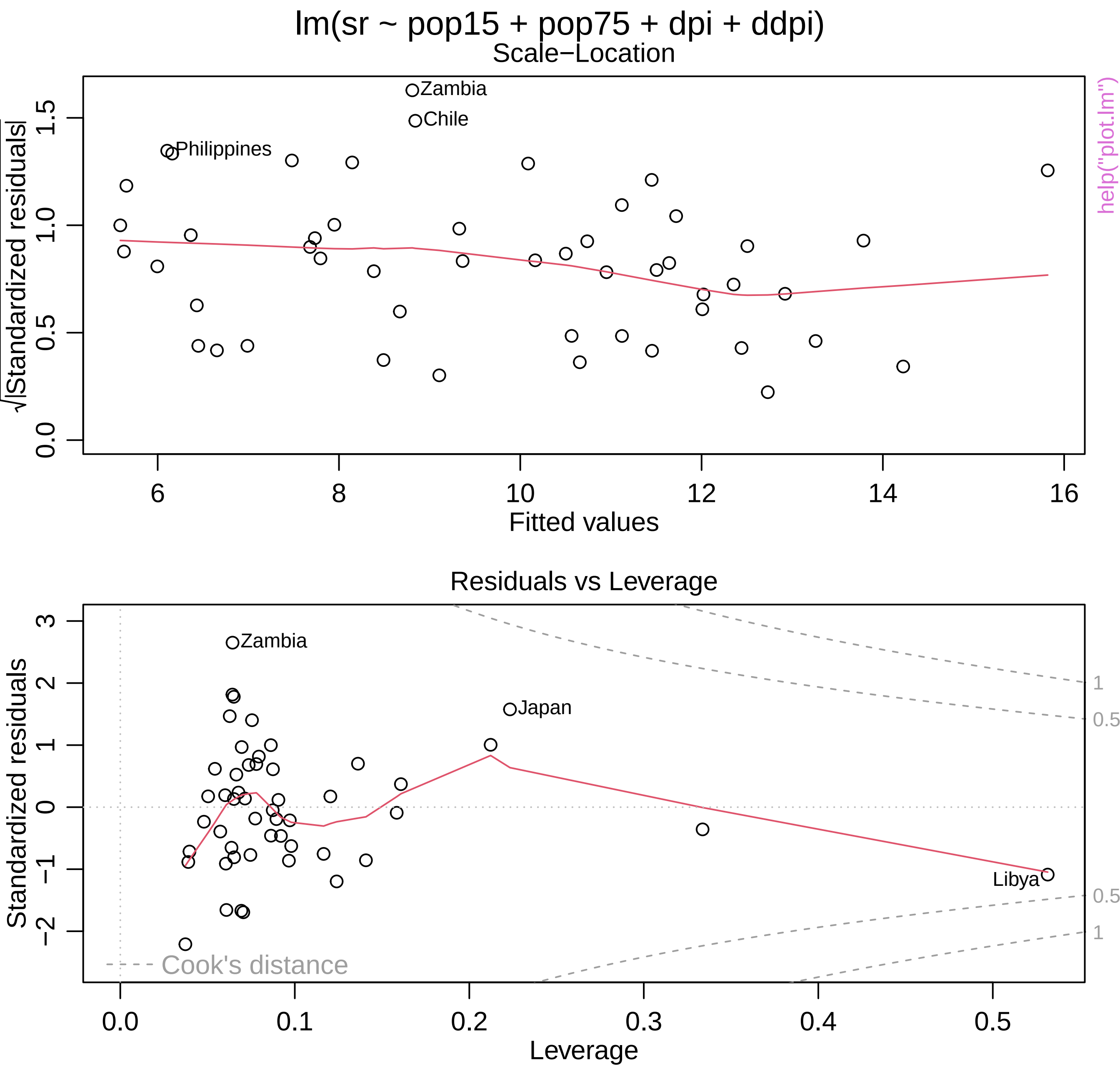
Isotonic regression isoreg(x = sample(9), y = y3)
x$y
x0
help("plot.isoreg")
Cumulative Data and Convex Minorant
cumsum(x$y)
x0

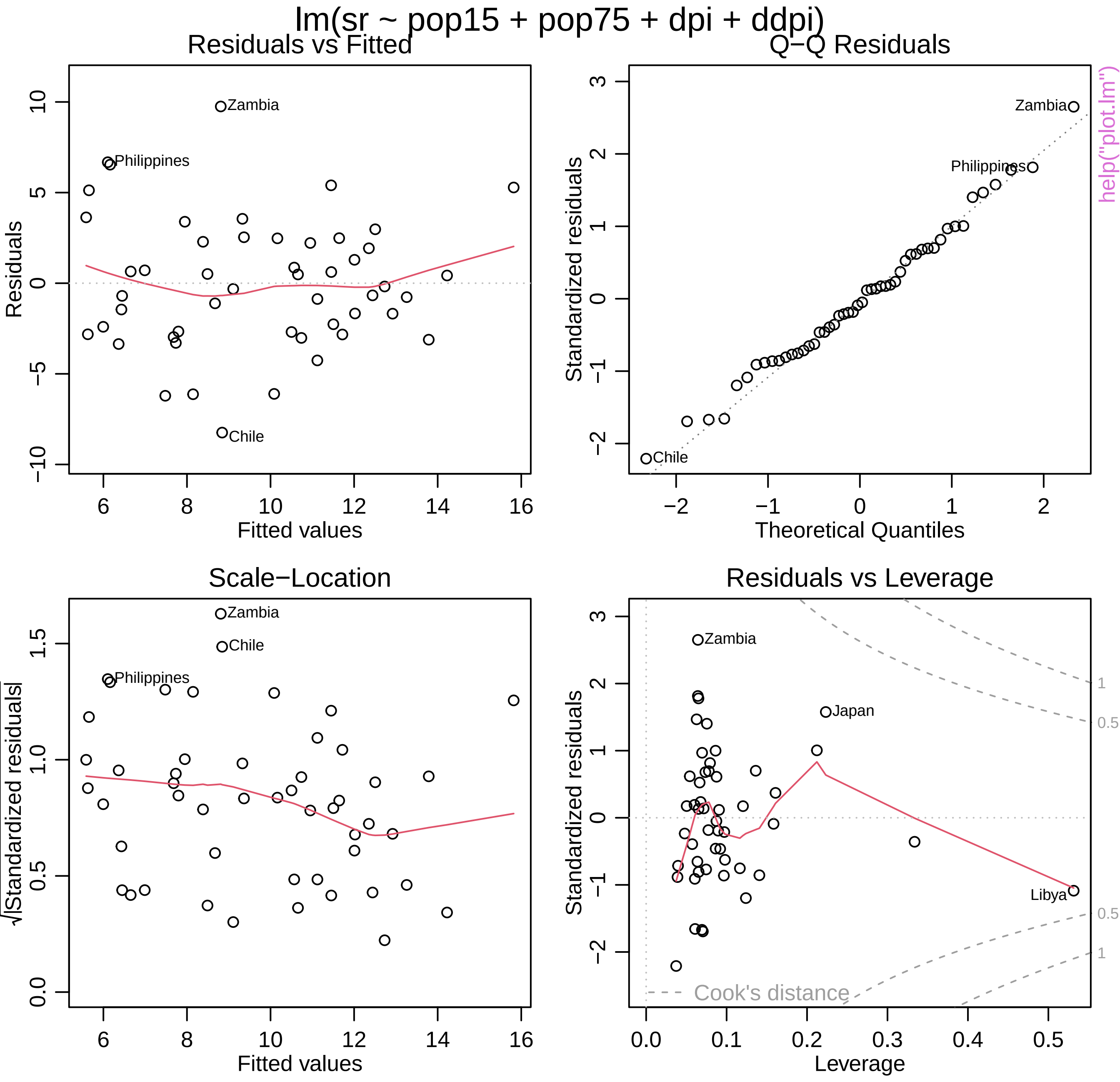

Isotonic regression isoreg(x = sample(9), y = y3)
Cumulative Data and Convex Minora
x$y
x0
cumsum(x$y)
x0
help("plot.isoreg")

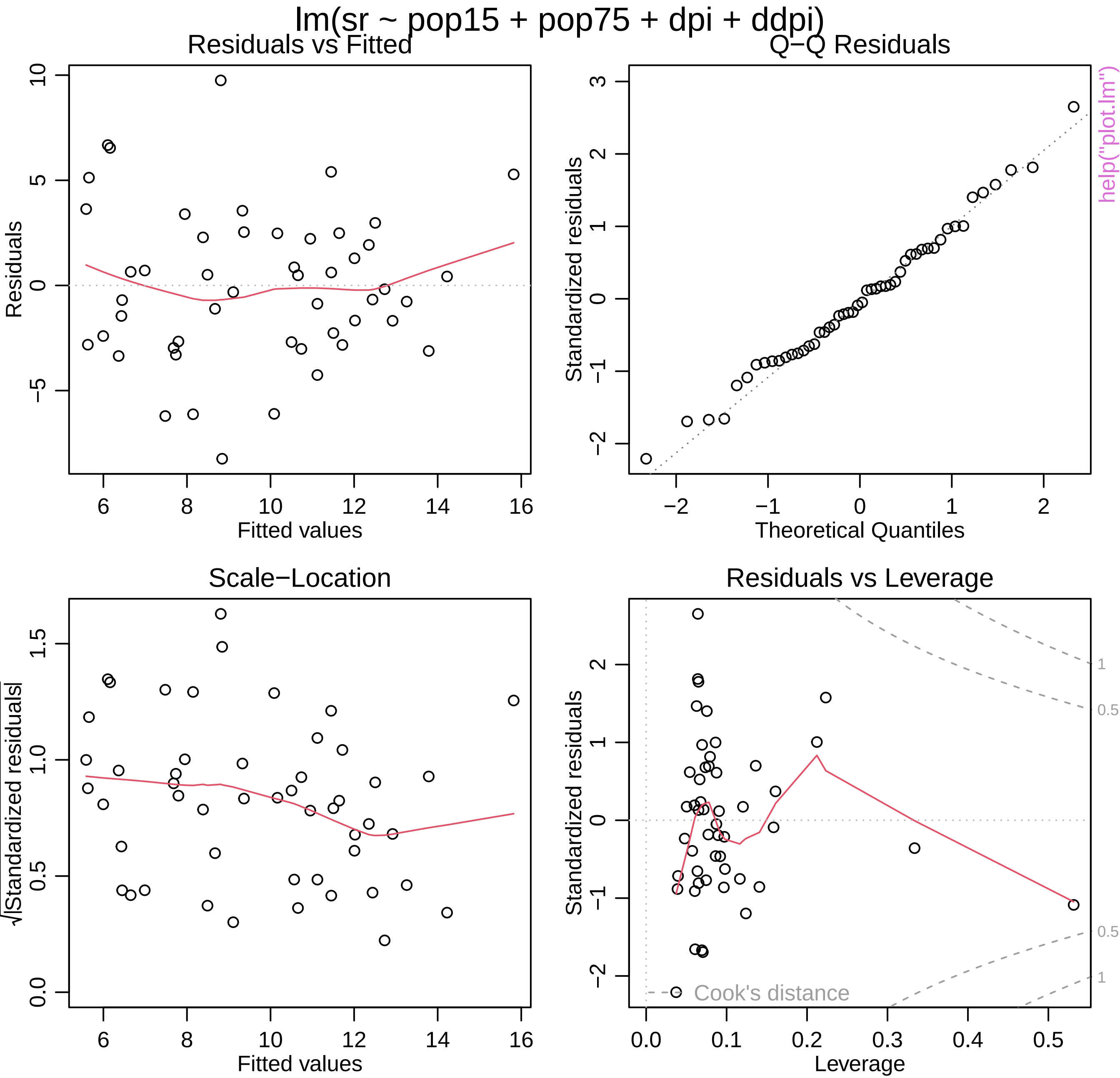

Isotonic regression isoreg(x = sample(10), y = sample(10, replace = TRUE))
help("plot.isoreg")
x$y
x0
Cumulative Data and Convex Minorant
cumsum(x$y)
x0
0
2
4
6
8
10
0
10
20
30
40

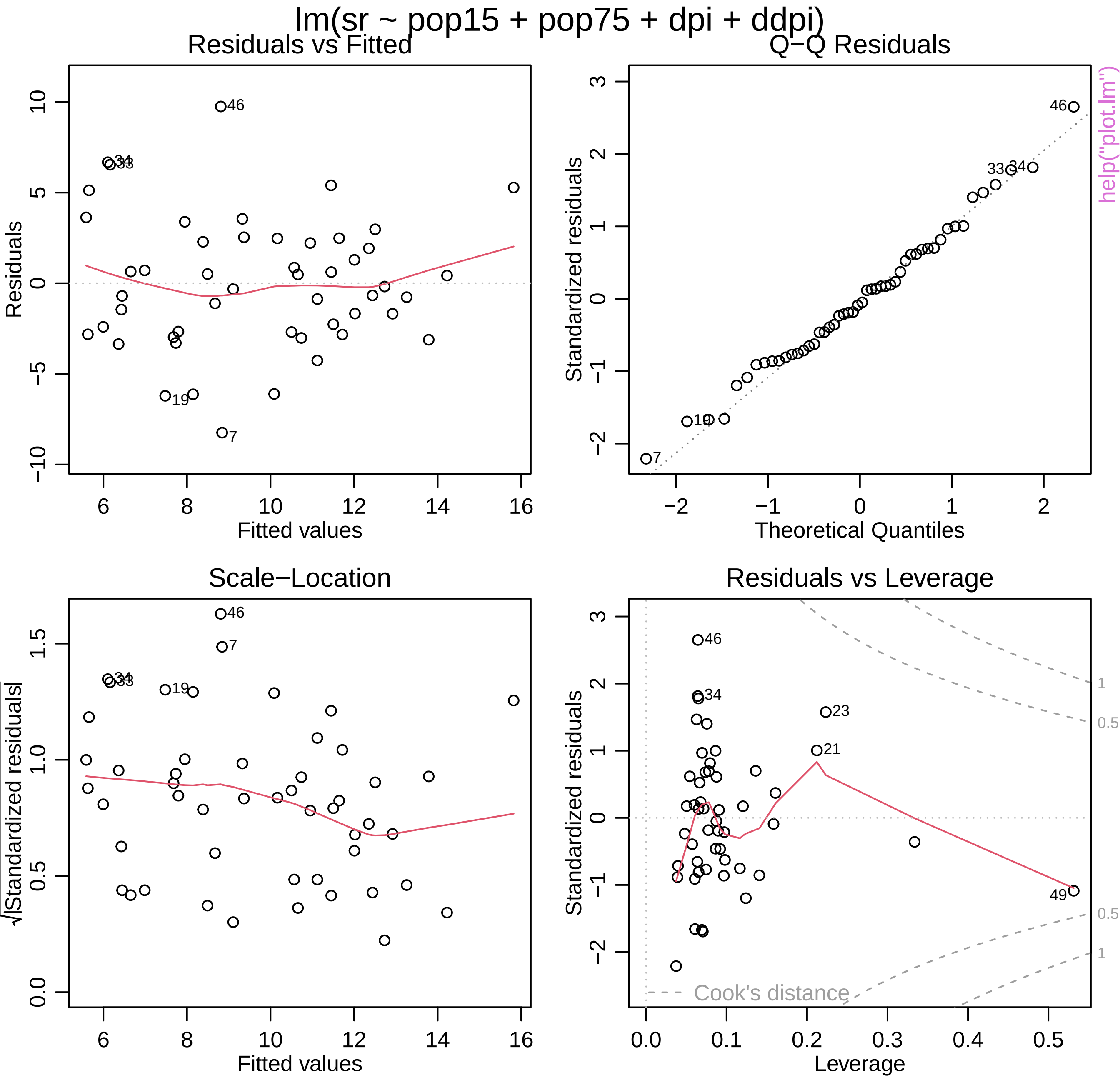

Residuals vs Fitted
Residuals
Fitted values
−10
−5
0
5
10
6
8
10
12
14
16
Zambia
Philippines
Chile
help("plot.lm")

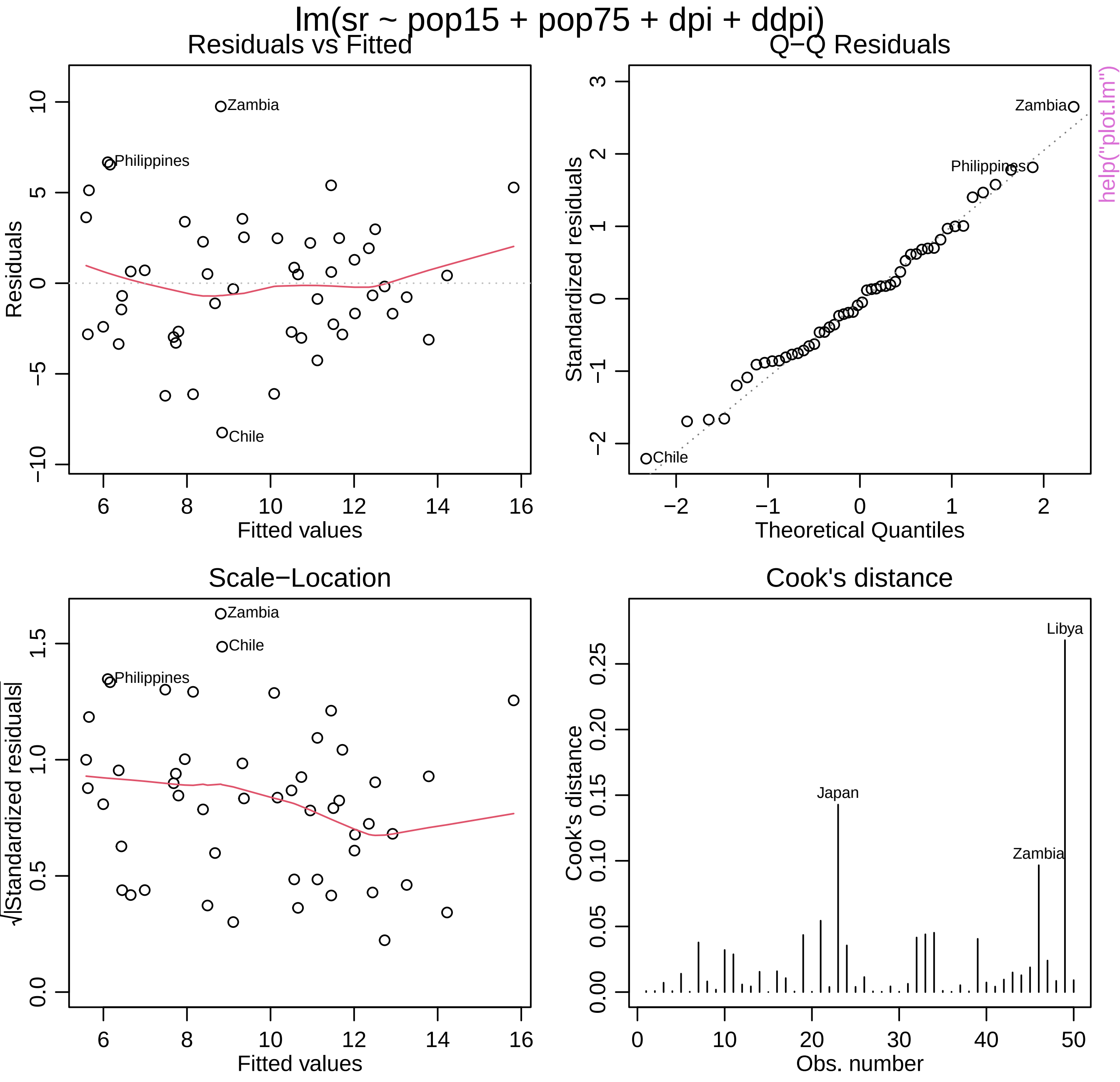

Q−Q Residuals
Standardized residuals
Theoretical Quantiles
−2
−1
0
1
2
3
Zambia
Philippines
Chile

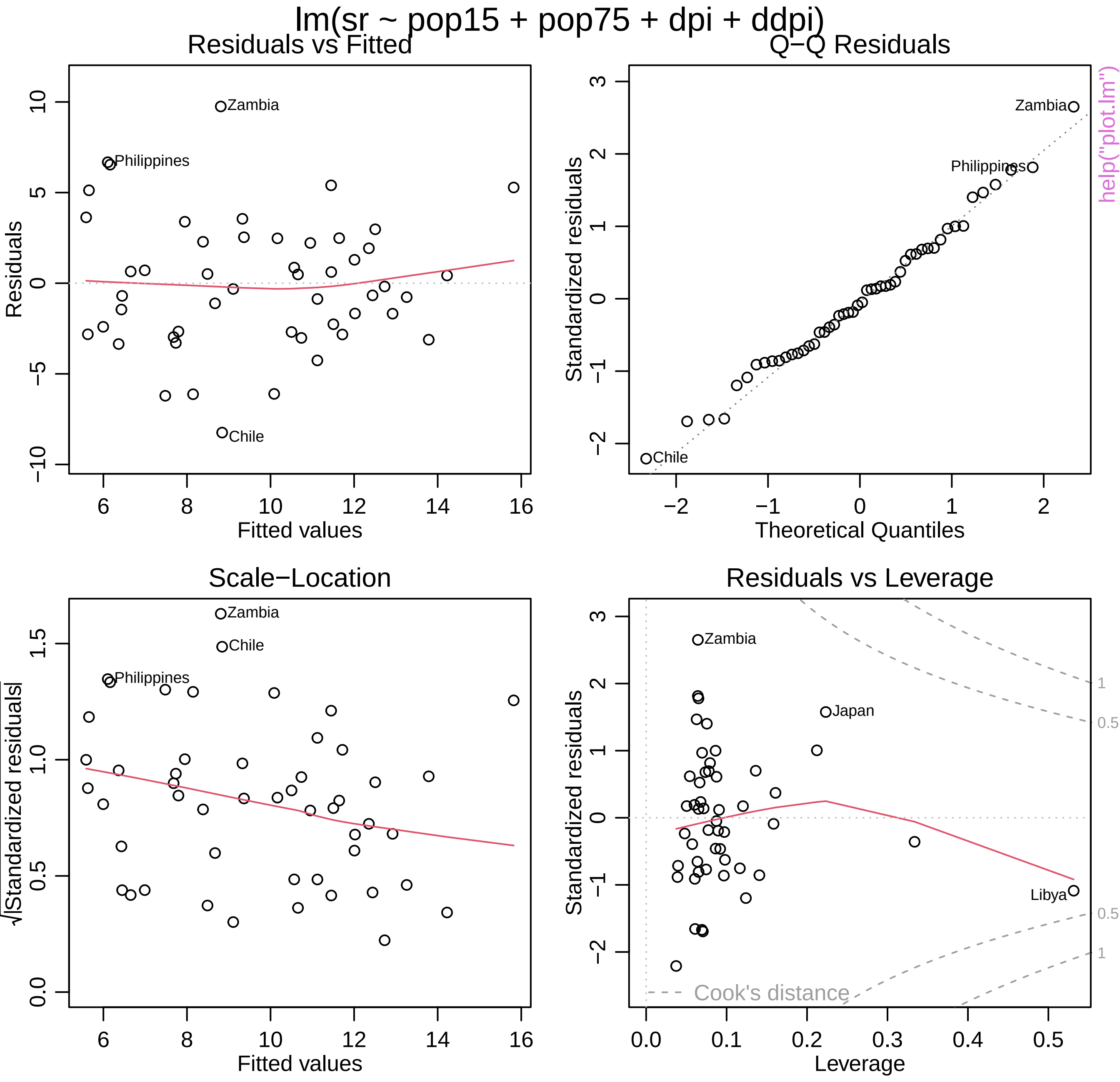

lm(sr ~ pop15 + pop75 + dpi + ddpi)
Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
Zambia
Chile
Philippines
6
8
10
12
14
16
Fitted values
help("plot.lm")
Residuals vs Leverage
Standardized residuals
−2
−1
0
1
2
3
Zambia
Japan
Libya
1
0.5
0.5
1
Cook's distance
0.0
0.1
0.2
0.3
0.4
0.5
Leverage


lm(sr ~ pop15 + pop75 + dpi + ddpi)
Residuals vs Fitted
Residuals
Fitted values
Zambia
Philippines
Chile
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
Zambia
Philippines
Chile
help("plot.lm")
Scale−Location
√|Standardized residuals|
Fitted values
Zambia
Chile
Philippines
Residuals vs Leverage
Standardized residuals
Leverage
Zambia
Japan
Libya
Cook's distance
1
0.5
0.5
1

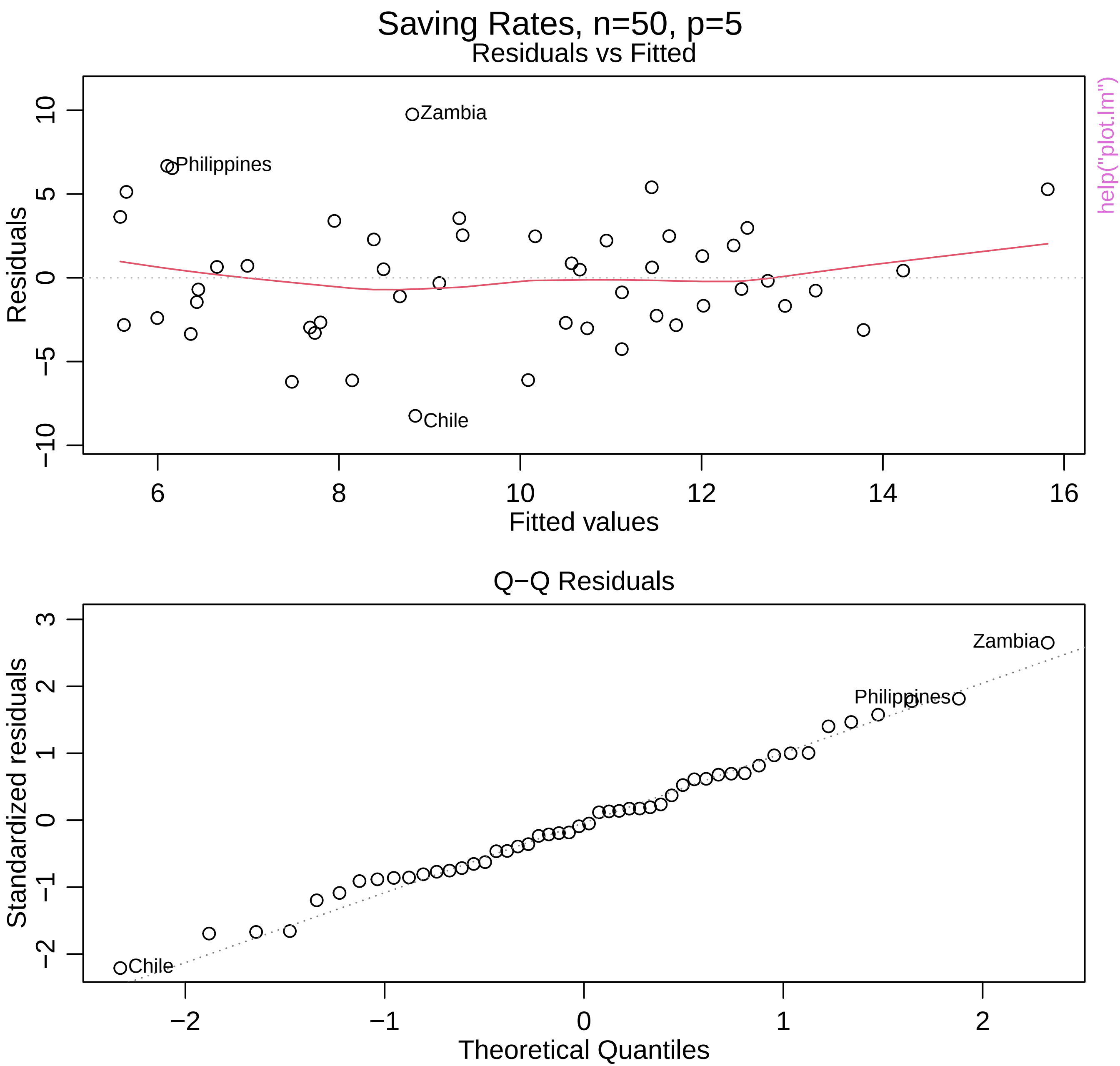

lm(sr ~ pop15 + pop75 + dpi + ddpi)
Residuals vs Fitted
Residuals
Fitted values
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
help("plot.lm")
Scale−Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
1
0.5
0.5
1

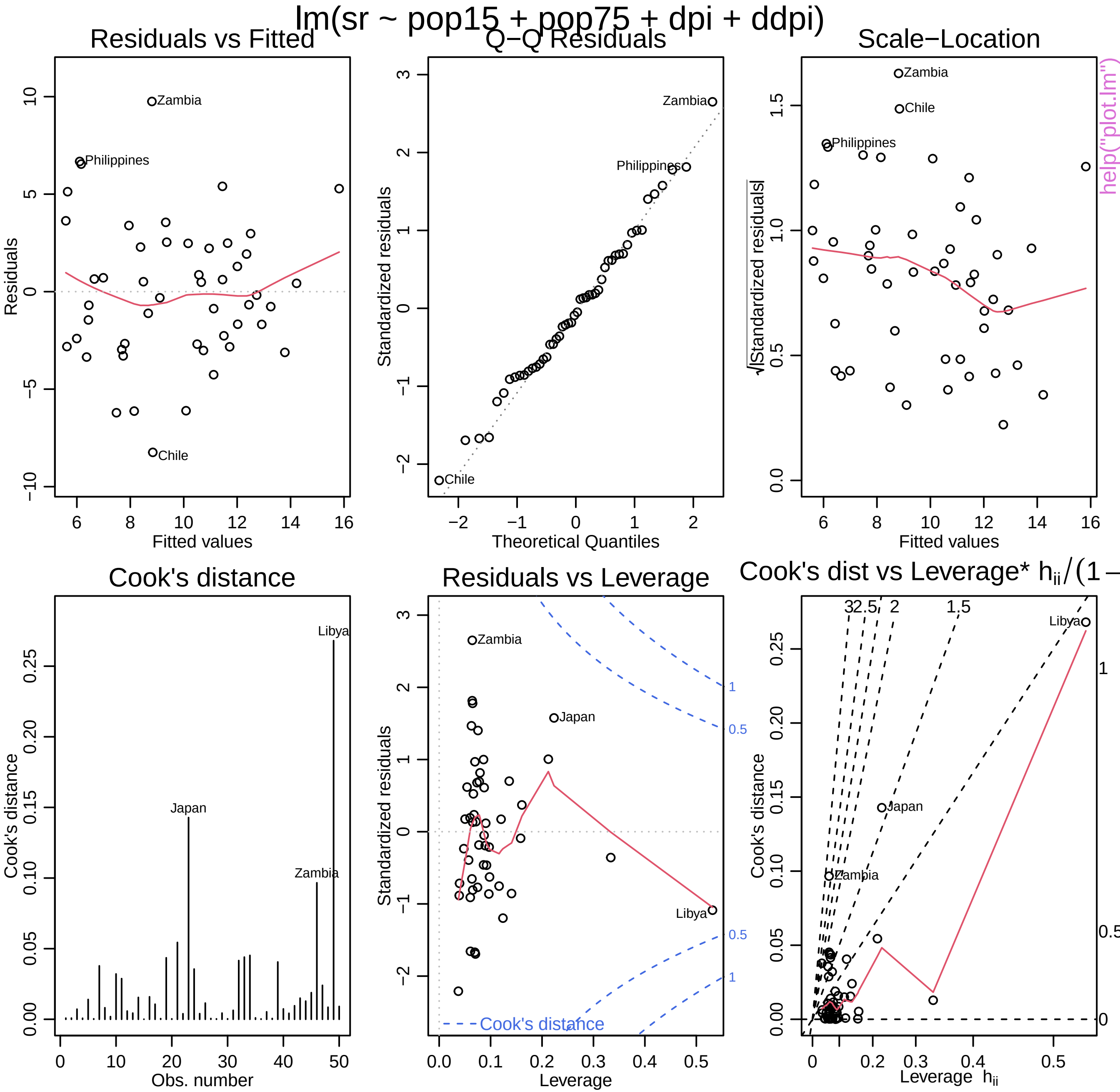

lm(sr ~ pop15 + pop75 + dpi + ddpi)
Residuals vs Fitted
Residuals
Fitted values
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
help("plot.lm")
Scale−Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance

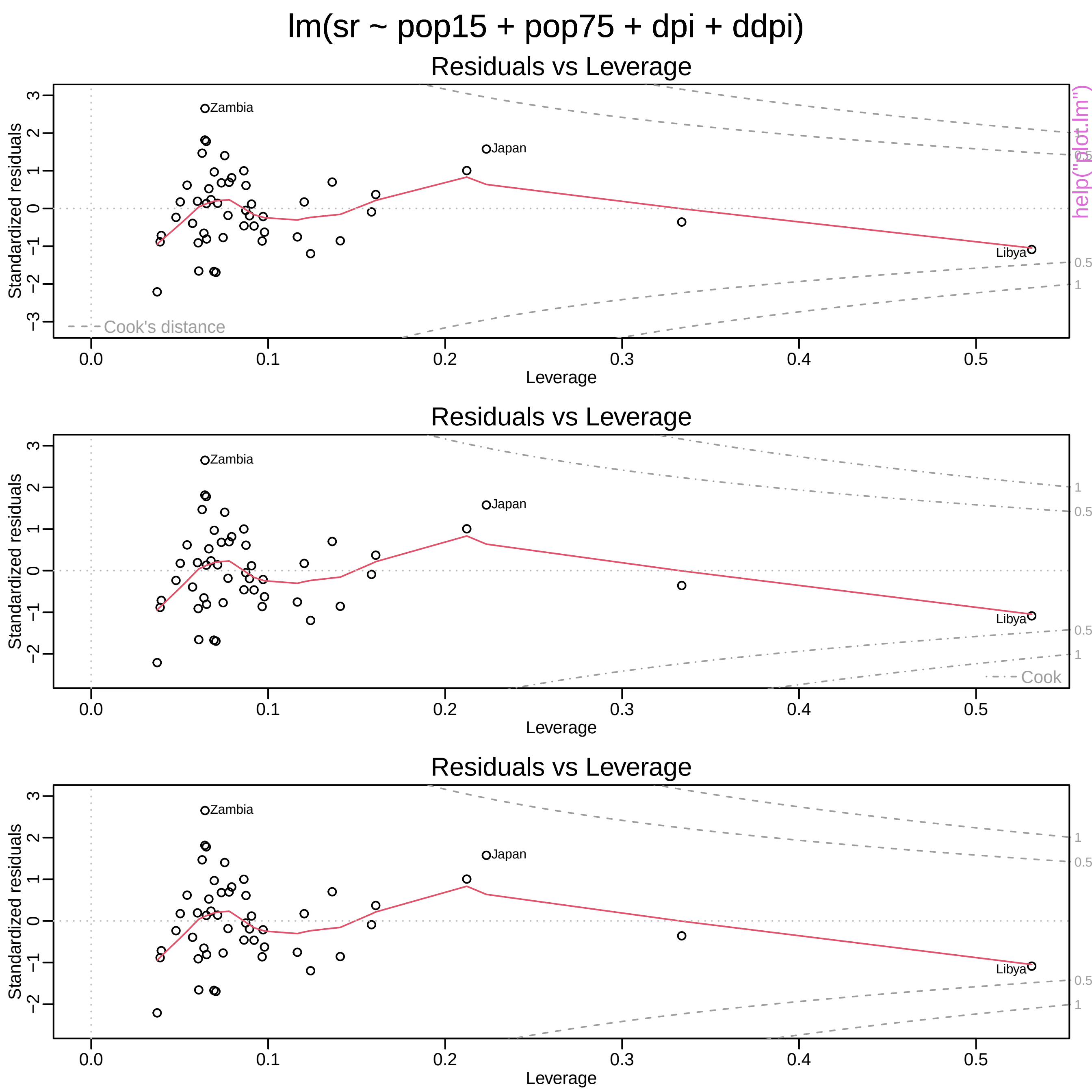

lm(sr ~ pop15 + pop75 + dpi + ddpi)
Residuals vs Fitted
Residuals
Fitted values
Zambia
Philippines
Chile
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
help("plot.lm")
Scale−Location
√|Standardized residuals|
Cook's distance
Obs. number
Japan
Libya

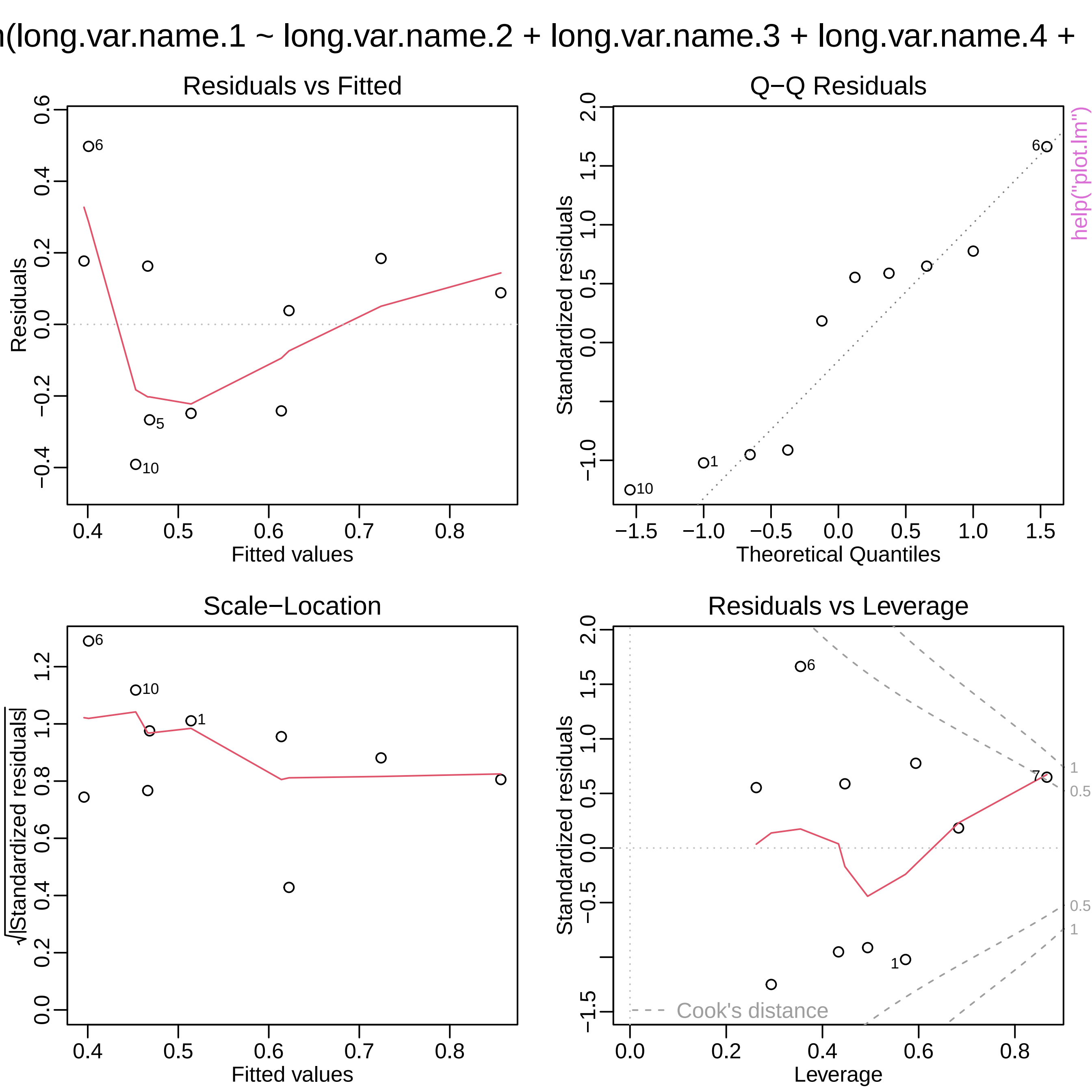


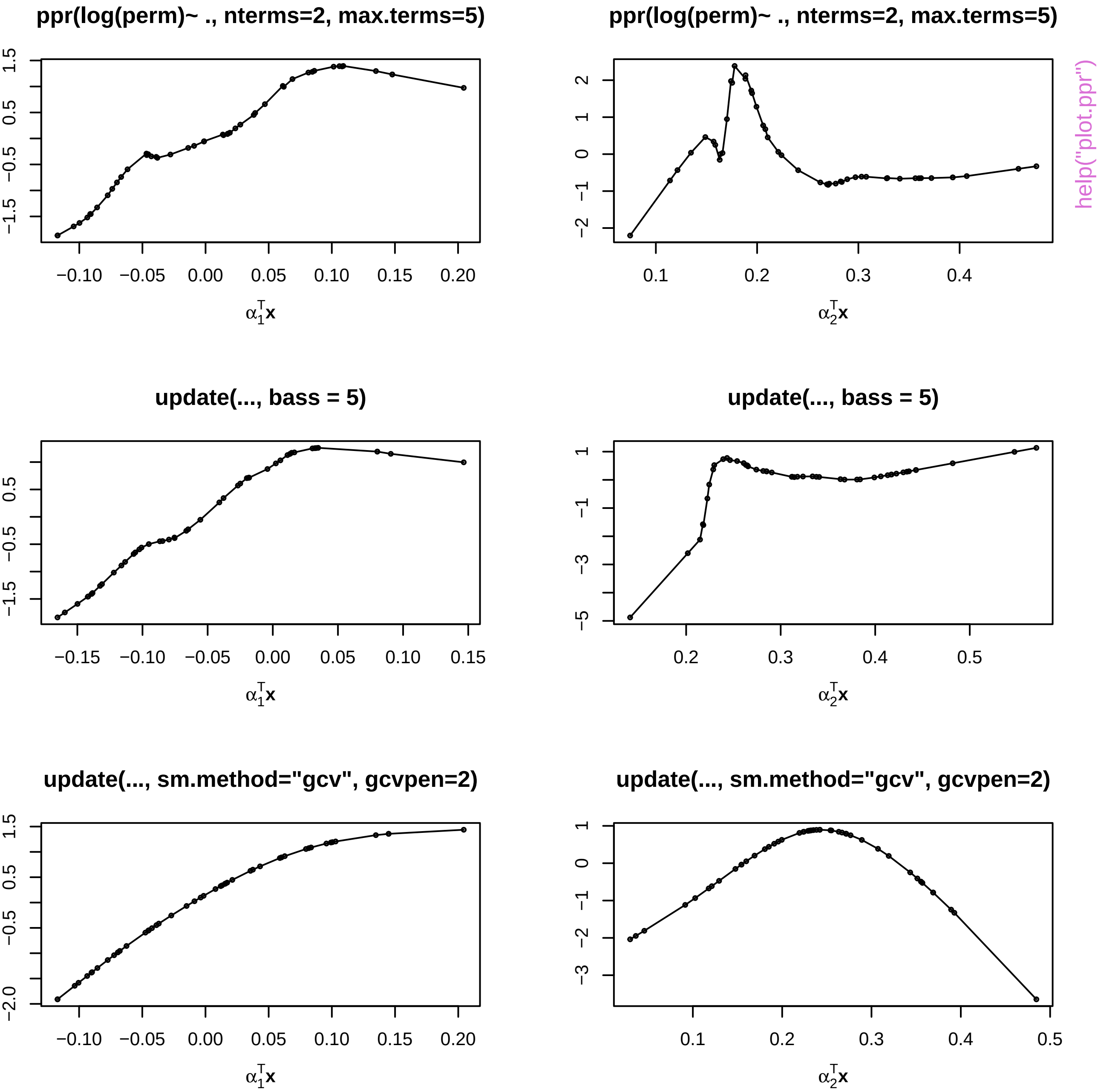
lm(sr ~ pop15 + pop75 + dpi + ddpi)
Residuals vs Fitted
Residuals
Fitted values
Zambia
Philippines
Chile
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
Zambia
Philippines
Chile
help("plot.lm")
Scale−Location
√|Standardized residuals|
Fitted values
Zambia
Chile
Philippines
Residuals vs Leverage
Standardized residuals
Leverage
Zambia
Japan
Libya
Cook's distance

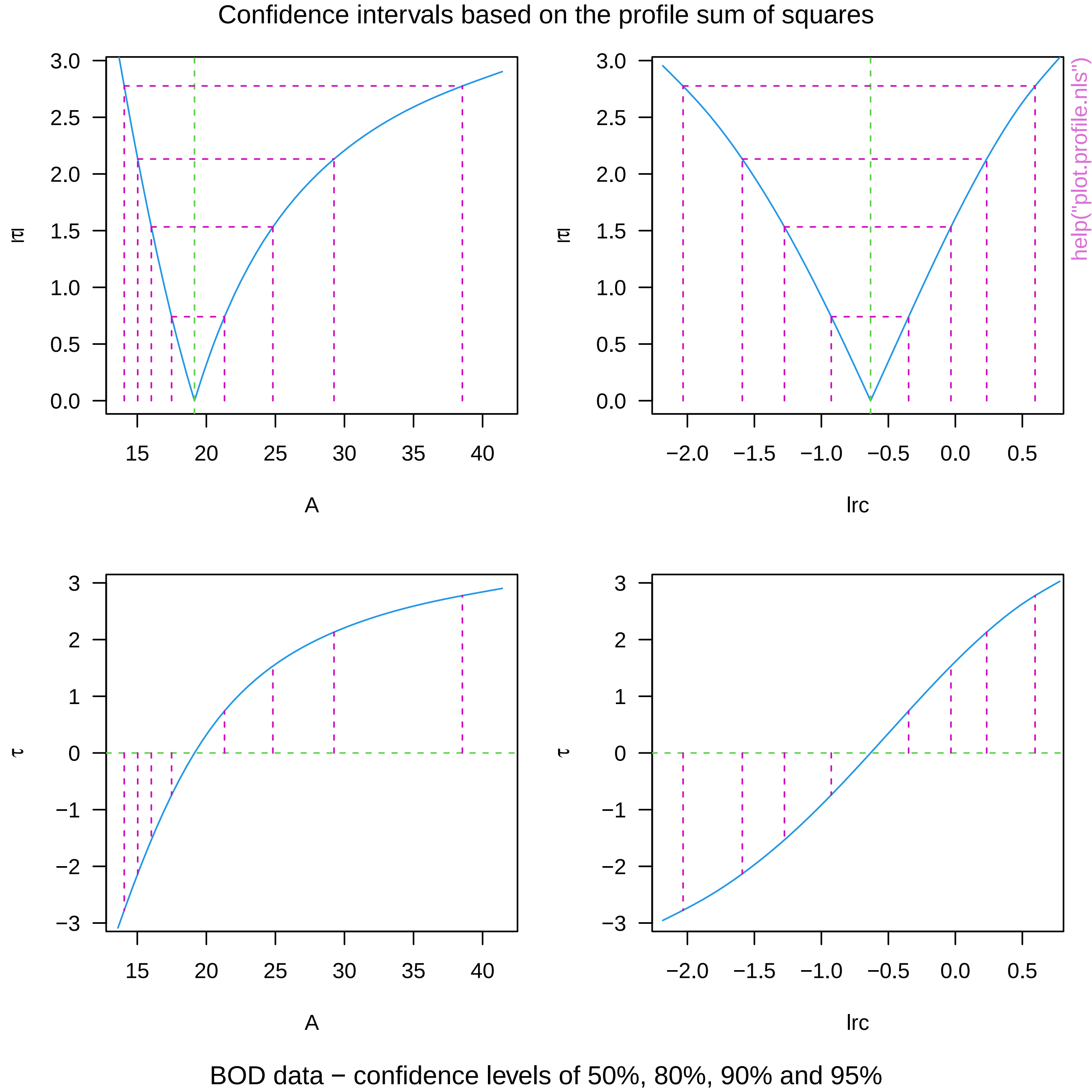

Saving Rates, n=50, p=5
Residuals vs Fitted
Residuals
Fitted values
Zambia
Philippines
Chile
help("plot.lm")
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
Zambia
Philippines
Chile


lm(sr ~ pop15 + pop75 + dpi + ddpi)
Residuals vs Fitted
Residuals
Fitted values
Zambia
Philippines
Chile
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
help("plot.lm")
Cook's distance
Obs. number
Libya
Japan
Residuals vs Leverage
Leverage
Cook's distance
Cook's dist vs Leverage* h_ii/(1 −
Leverage h_ii

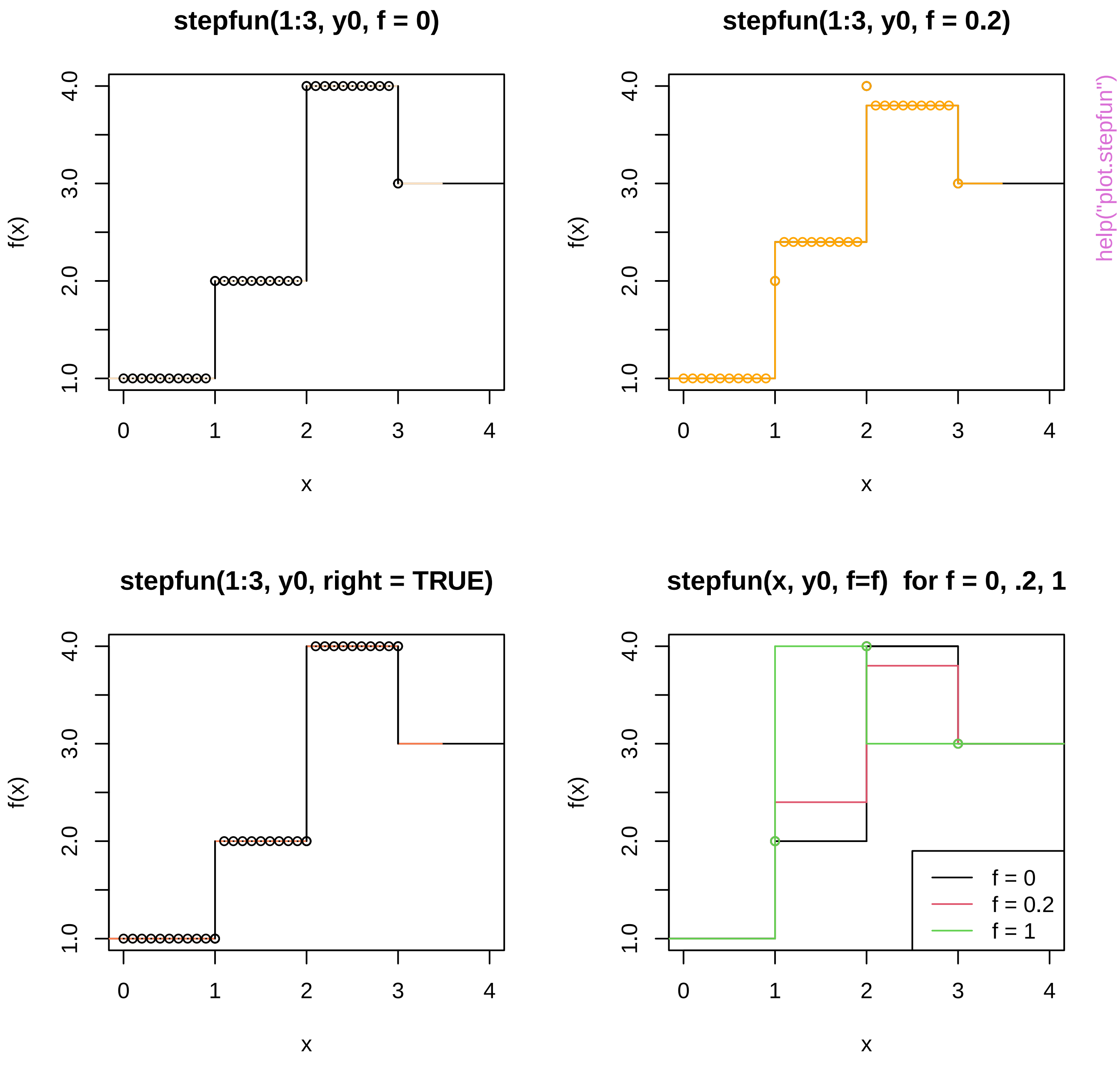

lm(sr ~ pop15 + pop75 + dpi + ddpi)
Residuals vs Leverage
Standardized residuals
Leverage
Zambia
Japan
Libya
Cook's distance
help("plot.lm")
0.5
1
Residuals vs Leverage
Standardized residuals
Leverage
Zambia
Japan
Libya
Cook
1
0.5
0.5
1
Residuals vs Leverage
Standardized residuals
Leverage
Zambia
Japan
Libya
1
0.5
0.5
1


n(long.var.name.1 ~ long.var.name.2 + long.var.name.3 + long.var.name.4 +
Residuals vs Fitted
Residuals
Fitted values
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
help("plot.lm")
Scale−Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance

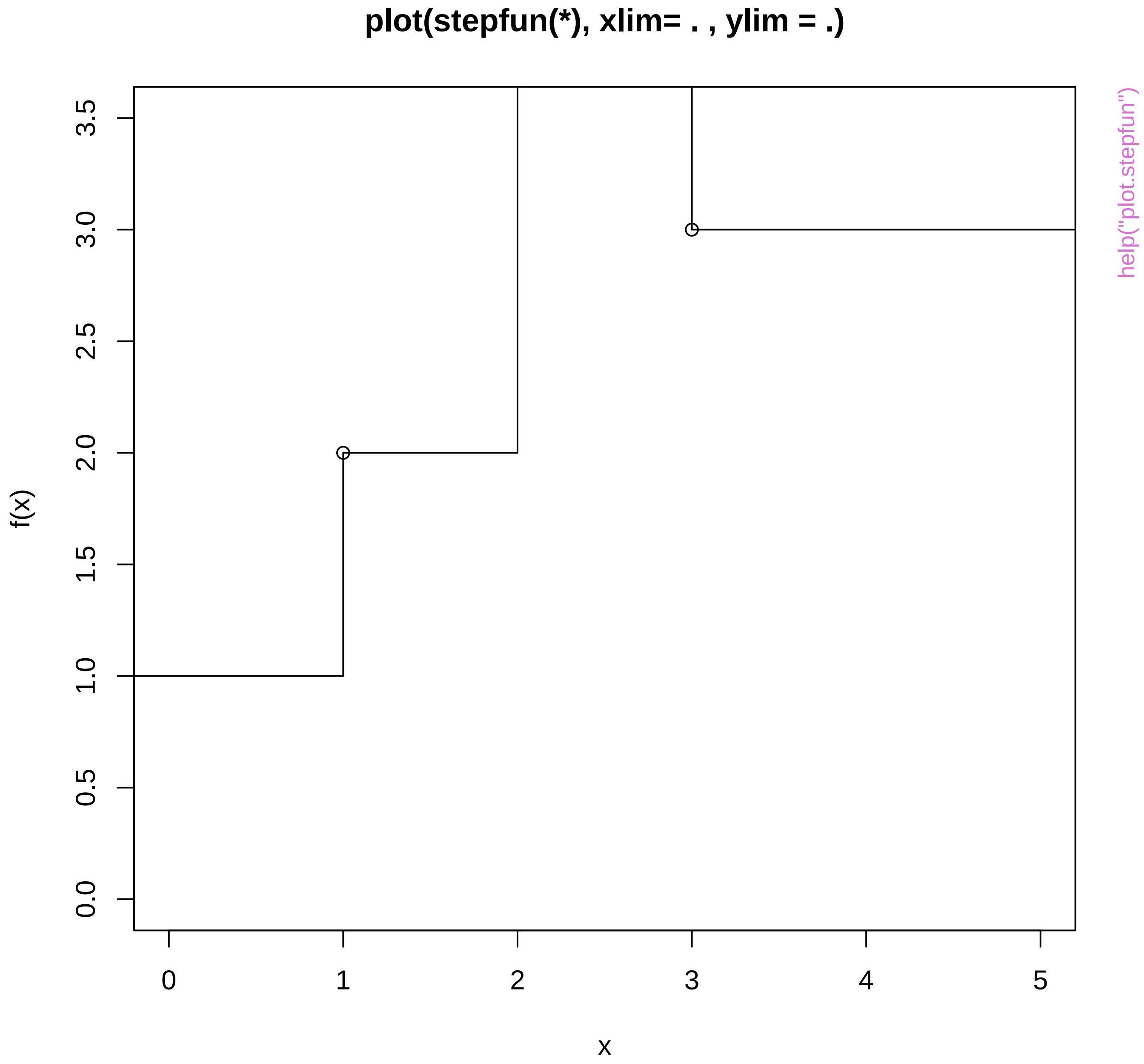

ppr(log(perm)~ ., nterms=2, max.terms=5)
ppr(log(perm)~ ., nterms=2, max.terms=5)
help("plot.ppr")
update(..., bass = 5)
update(..., bass = 5)
update(..., sm.method="gcv", gcvpen=2)
update(..., sm.method="gcv", gcvpen=2)
$\alpha_1^T\mathbf{x}$
$\alpha_2^T\mathbf{x}$

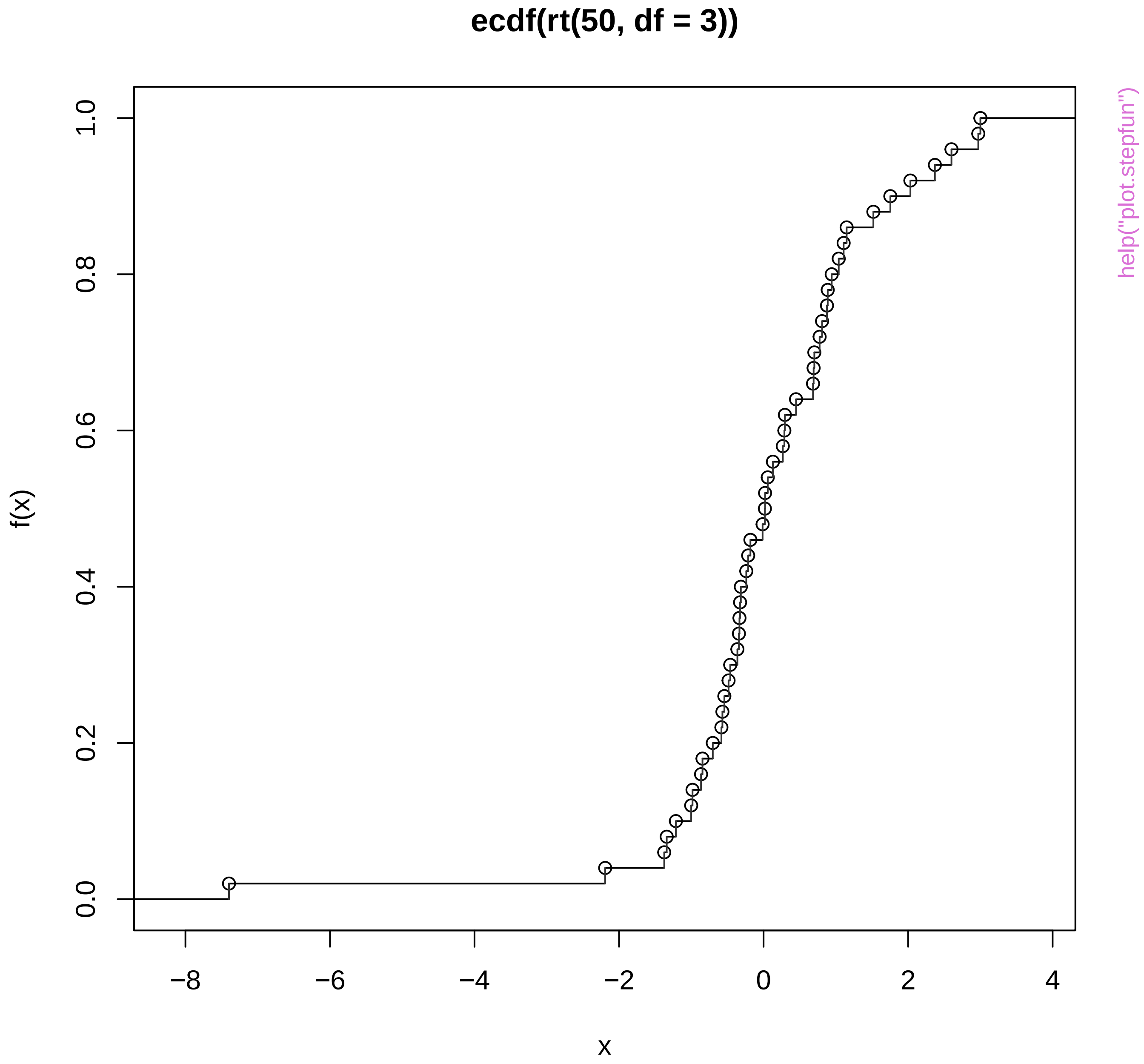


Confidence intervals based on the profile sum of squares
|τ|
τ
A
lrc
help("plot.profile.nls")
BOD data − confidence levels of 50%, 80%, 90% and 95%

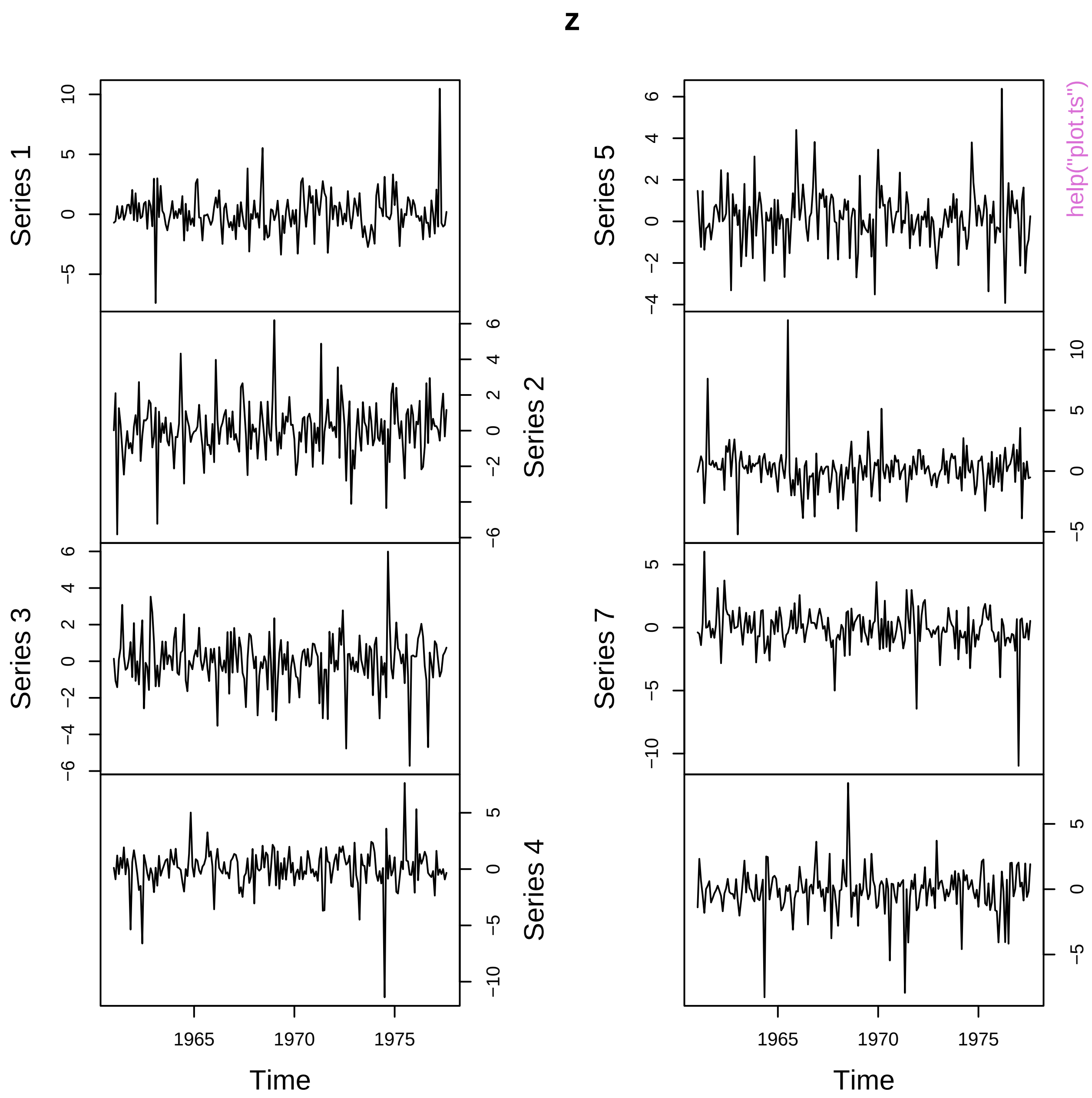


stepfun(1:3, y0, f = 0)
stepfun(1:3, y0, f = 0.2)
stepfun(1:3, y0, right = TRUE)
stepfun(x, y0, f=f) for f = 0, .2, 1
f(x)
x
1.0
2.0
3.0
4.0
0
1
2
3
4
f = 0
f = 0.2
f = 1
help("plot.stepfun")


plot(stepfun(*), xlim= . , ylim = .)
f(x)
x
0
1
2
3
4
5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
help("plot.stepfun")

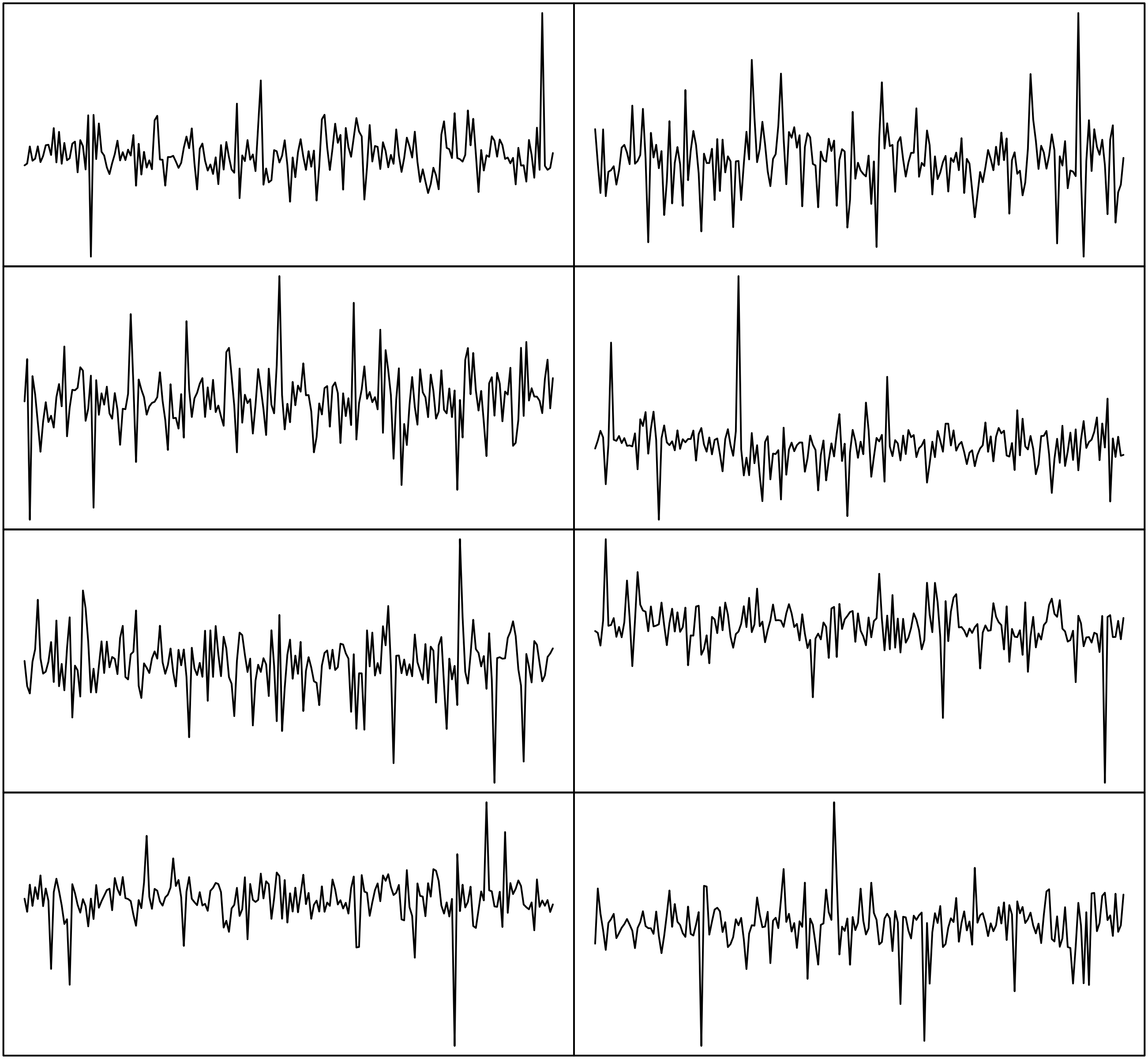


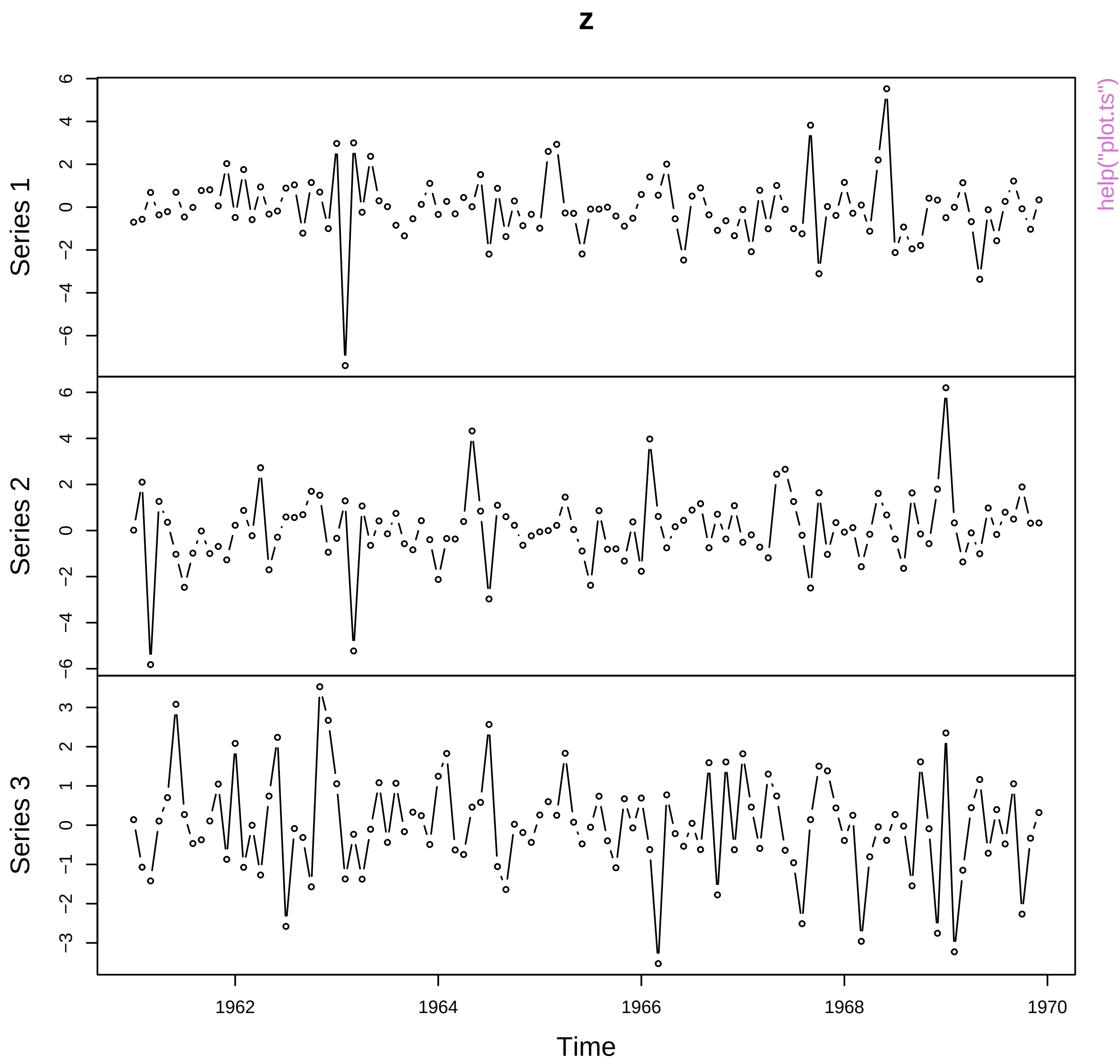
ecdf(rt(50, df = 3))
f(x)
x
0.0
0.2
0.4
0.6
0.8
1.0
−8
−6
−4
−2
0
2
4
help("plot.stepfun")

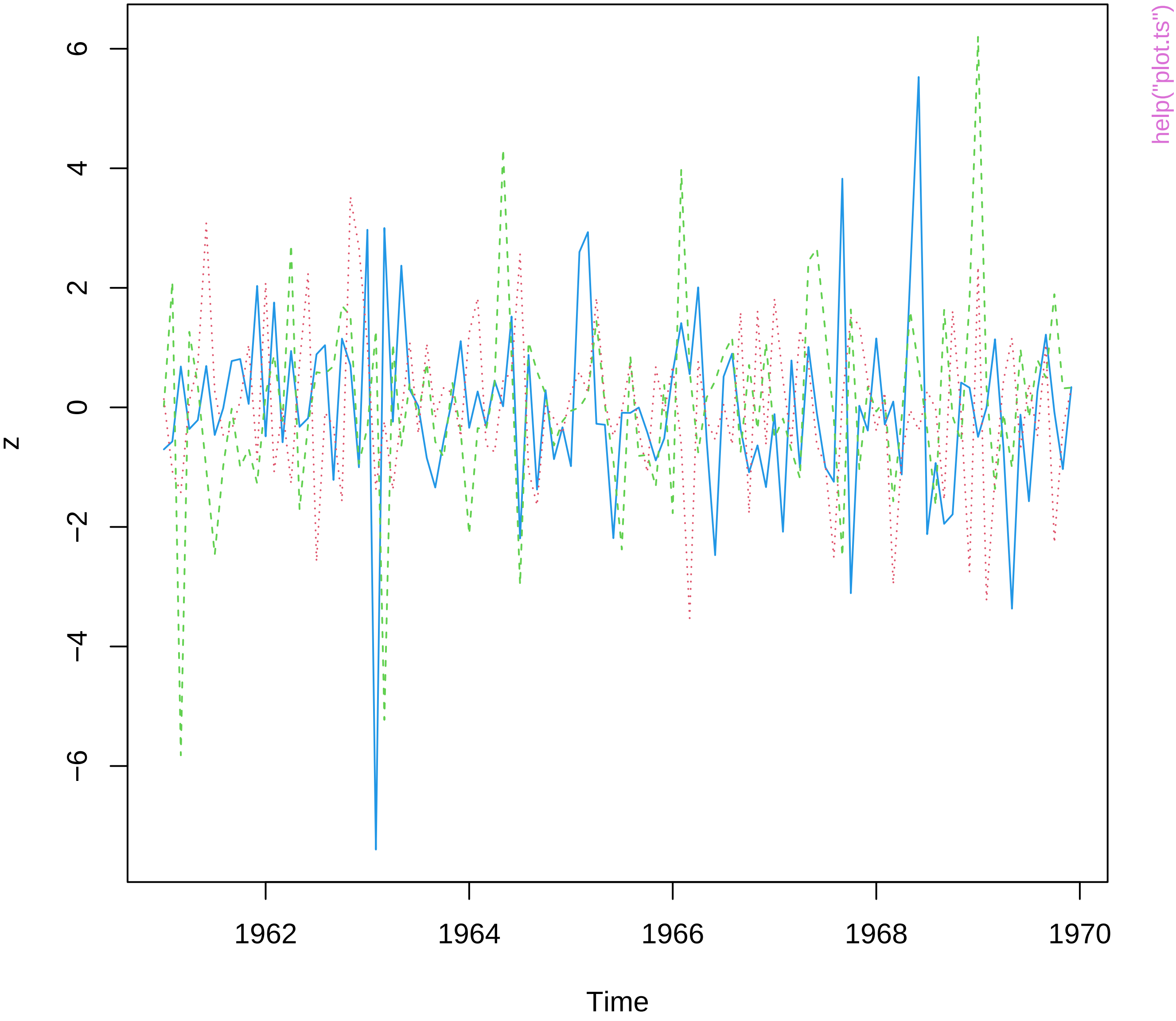


z
Series 1
Series 2
Series 3
Series 4
Series 5
Series 6
Series 7
Series 8
Time
Time
1965
1970
1975
help("plot.ts")

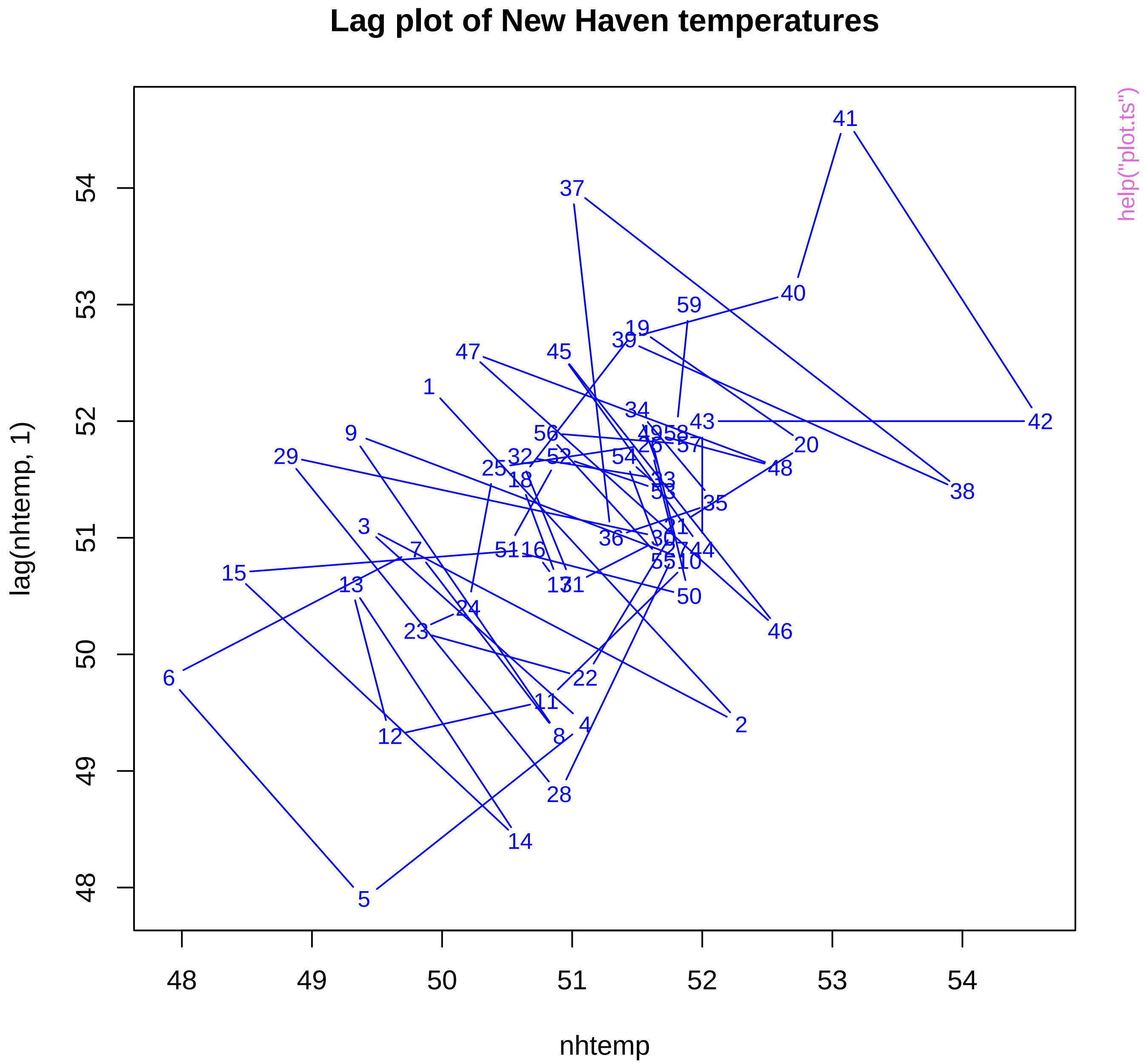
plot(ts(..), axes=FALSE, ann=FALSE, frame.plot=TRUE, mar..., oma...)

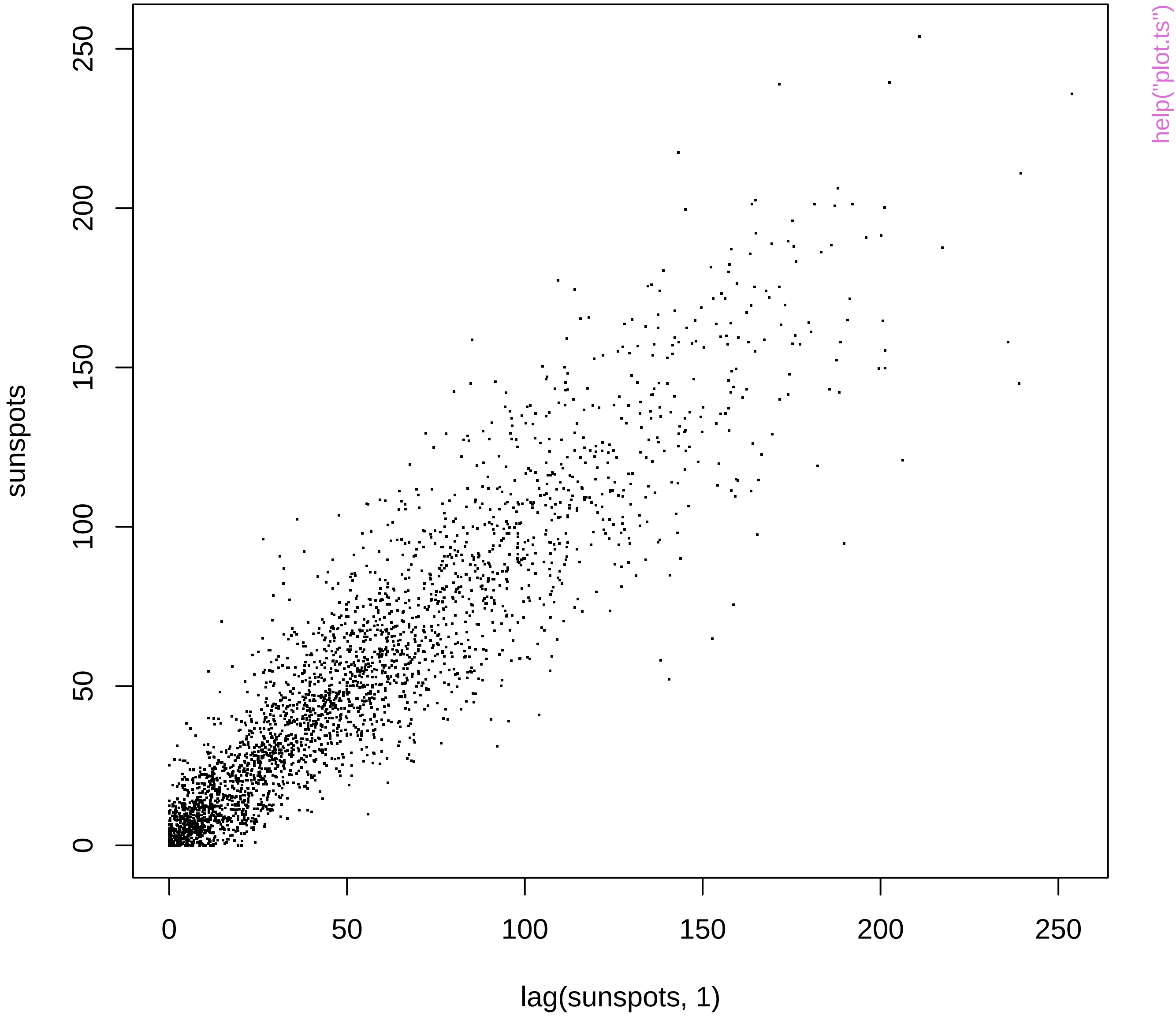



z
Series 1
Series 2
Series 3
Time
1962
1964
1966
1968
1970
help("plot.ts")

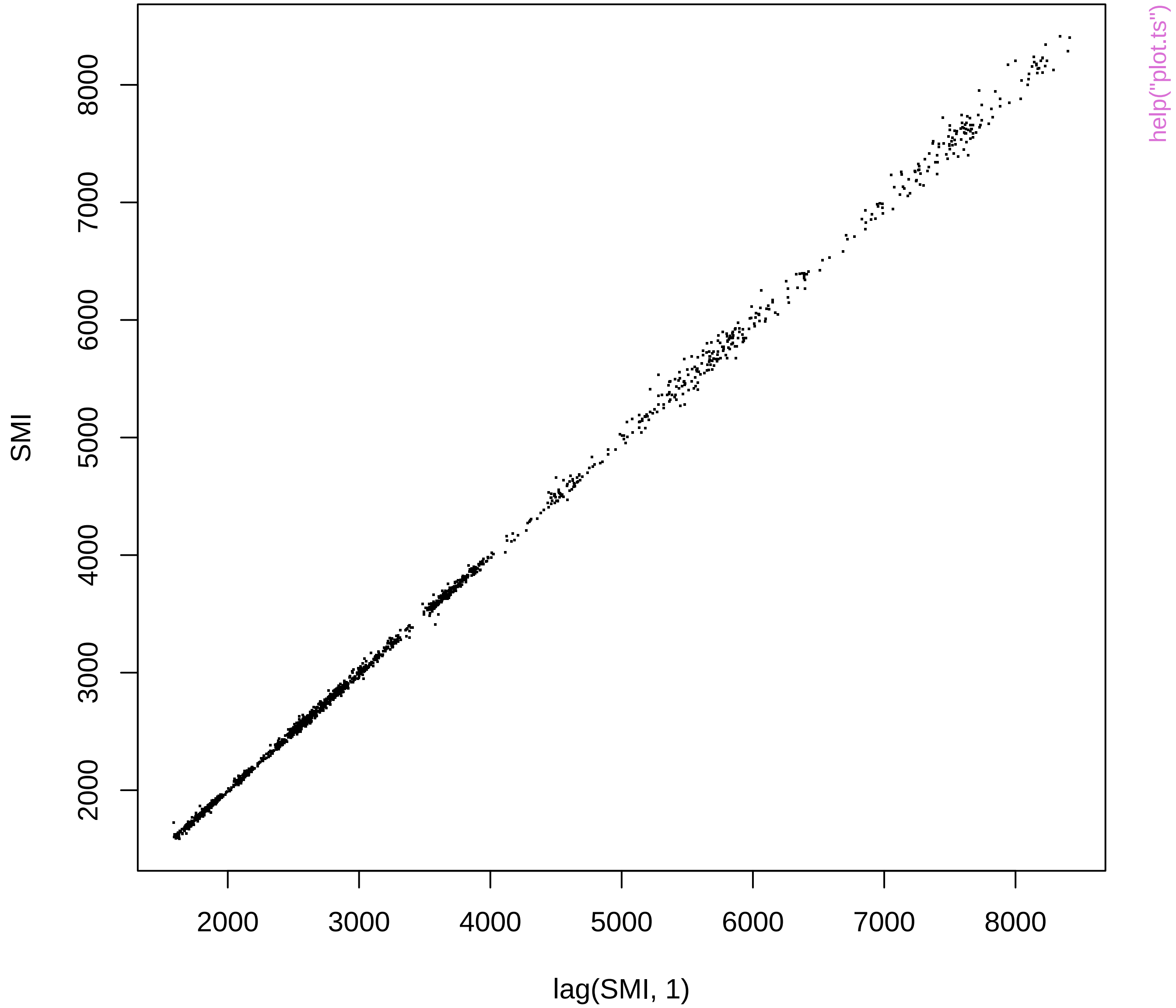


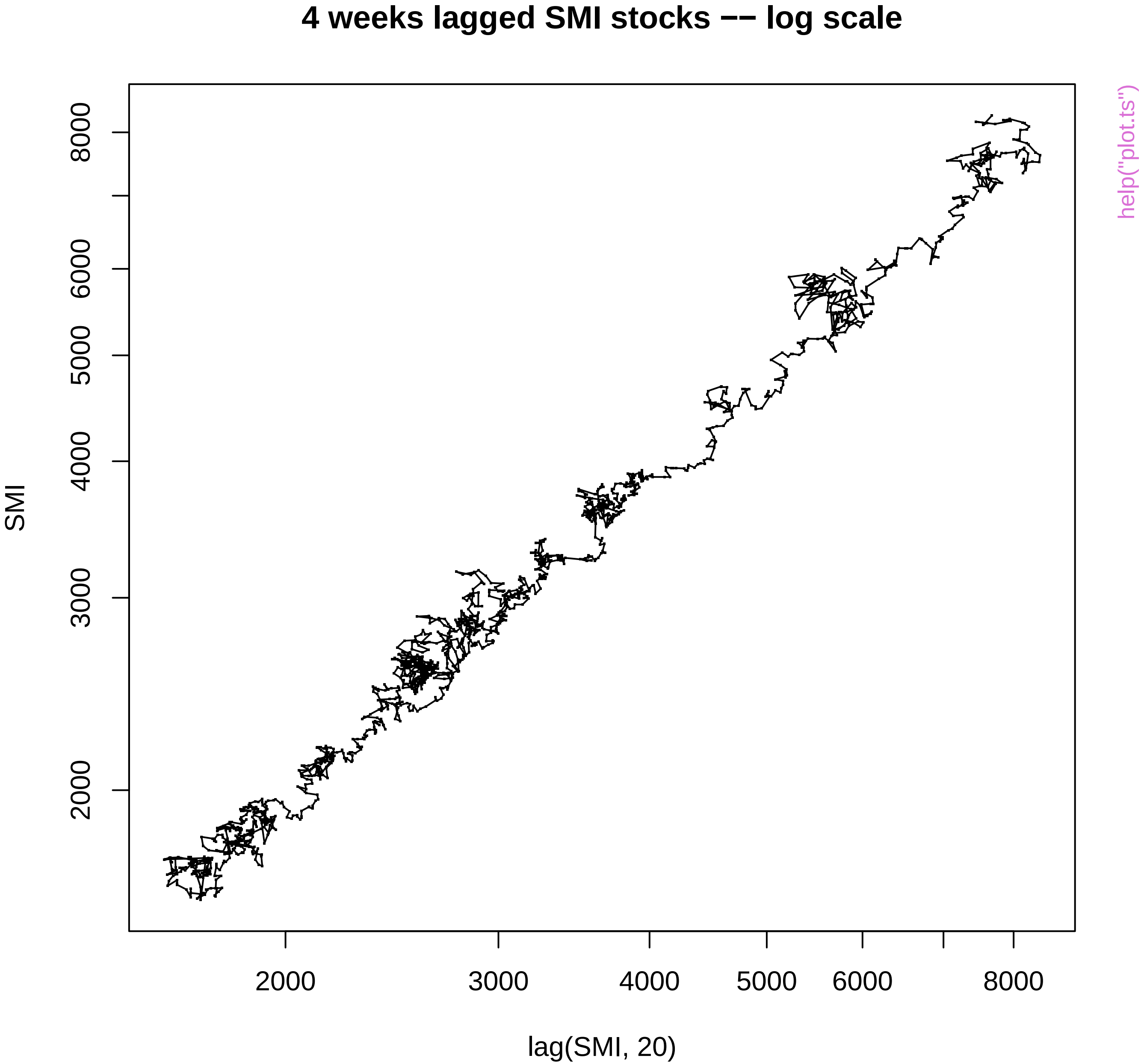
z
6
4
2
0
−2
−4
−6
1962
1964
1966
1968
1970
Time
help("plot.ts")


Lag plot of New Haven temperatures
lag(nhtemp, 1)
nhtemp
48 49 50 51 52 53 54
help("plot.ts")

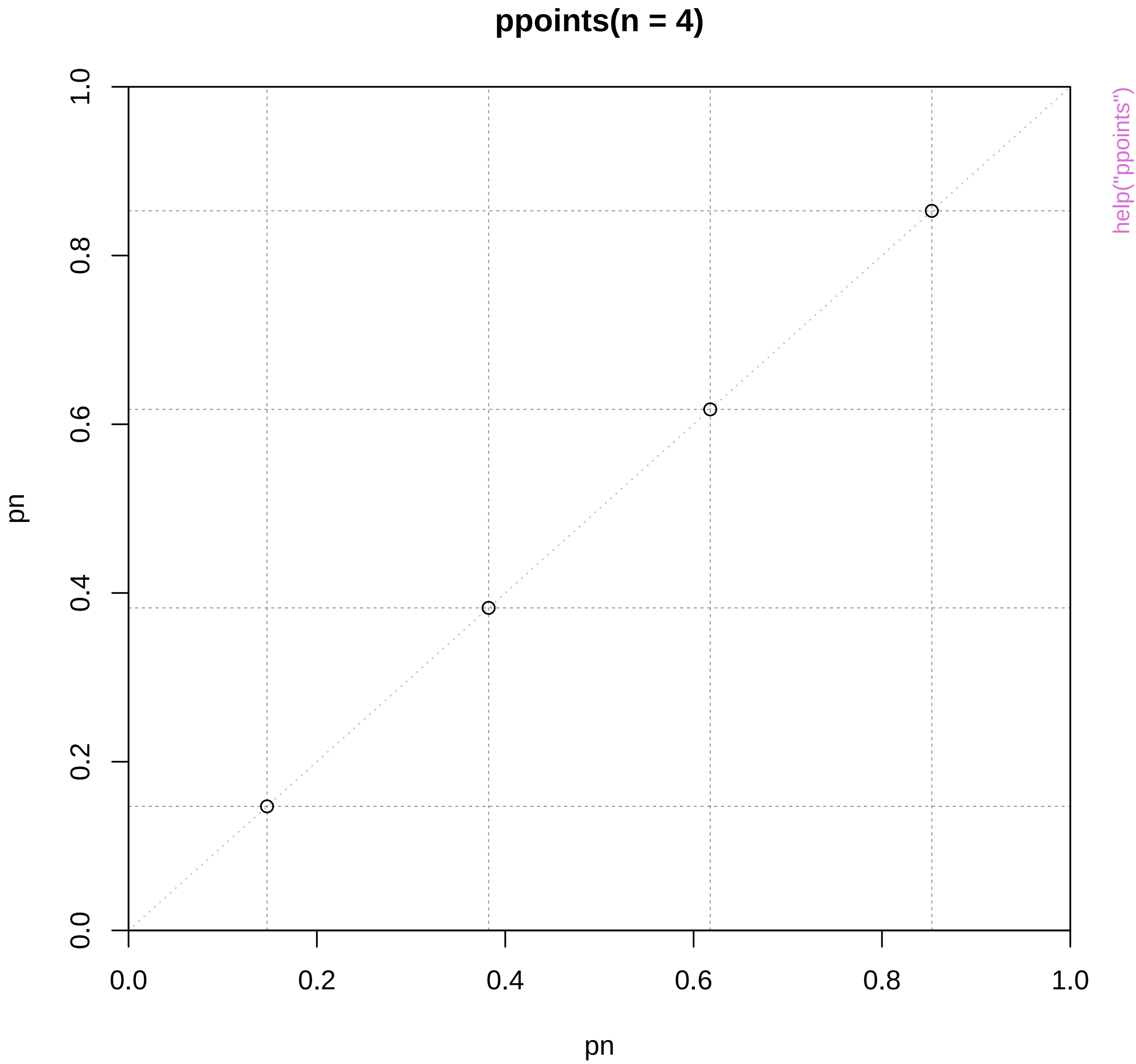


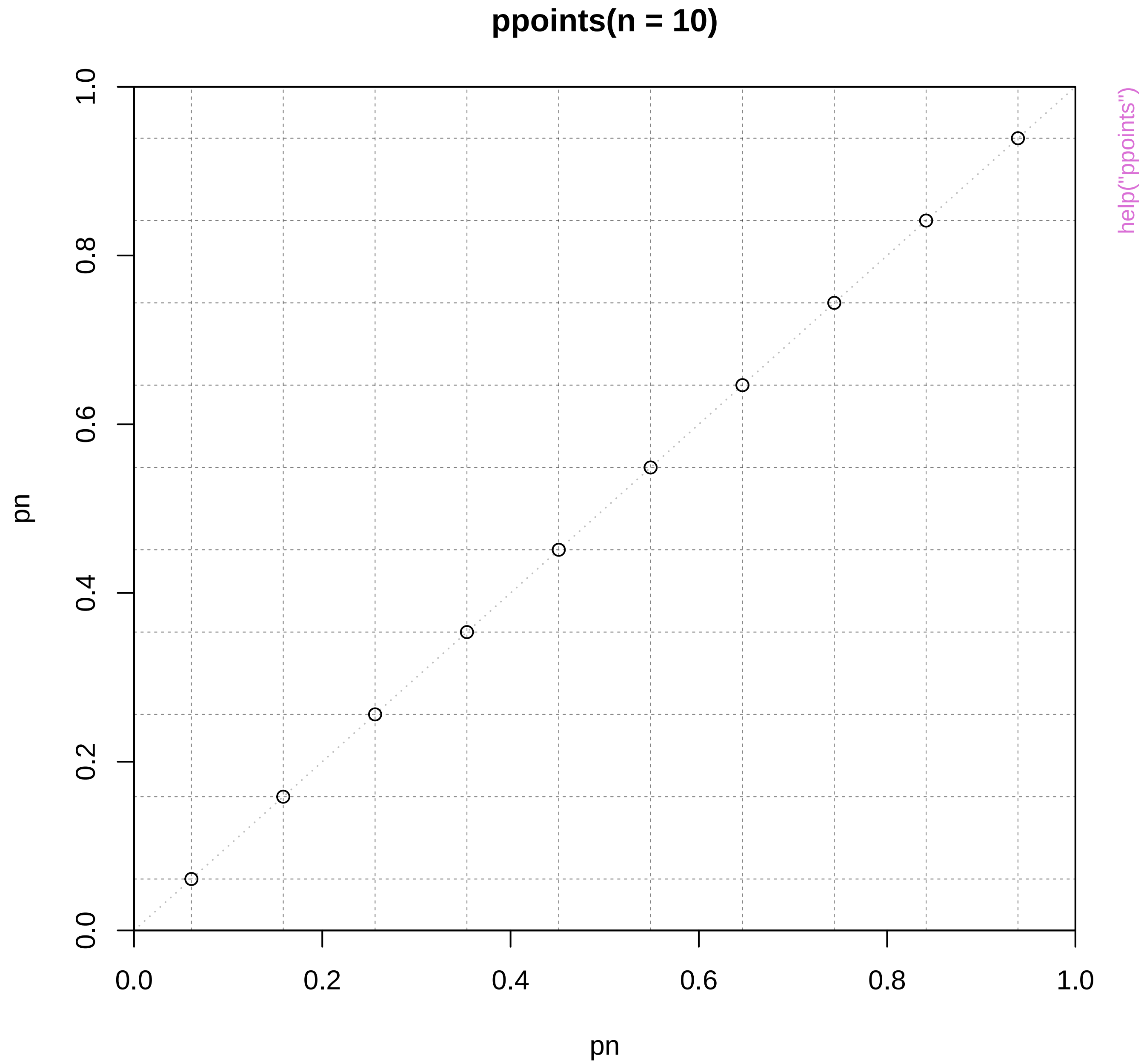
sunspots
0
50
100
150
200
250
lag(sunspots, 1)
help("plot.ts")

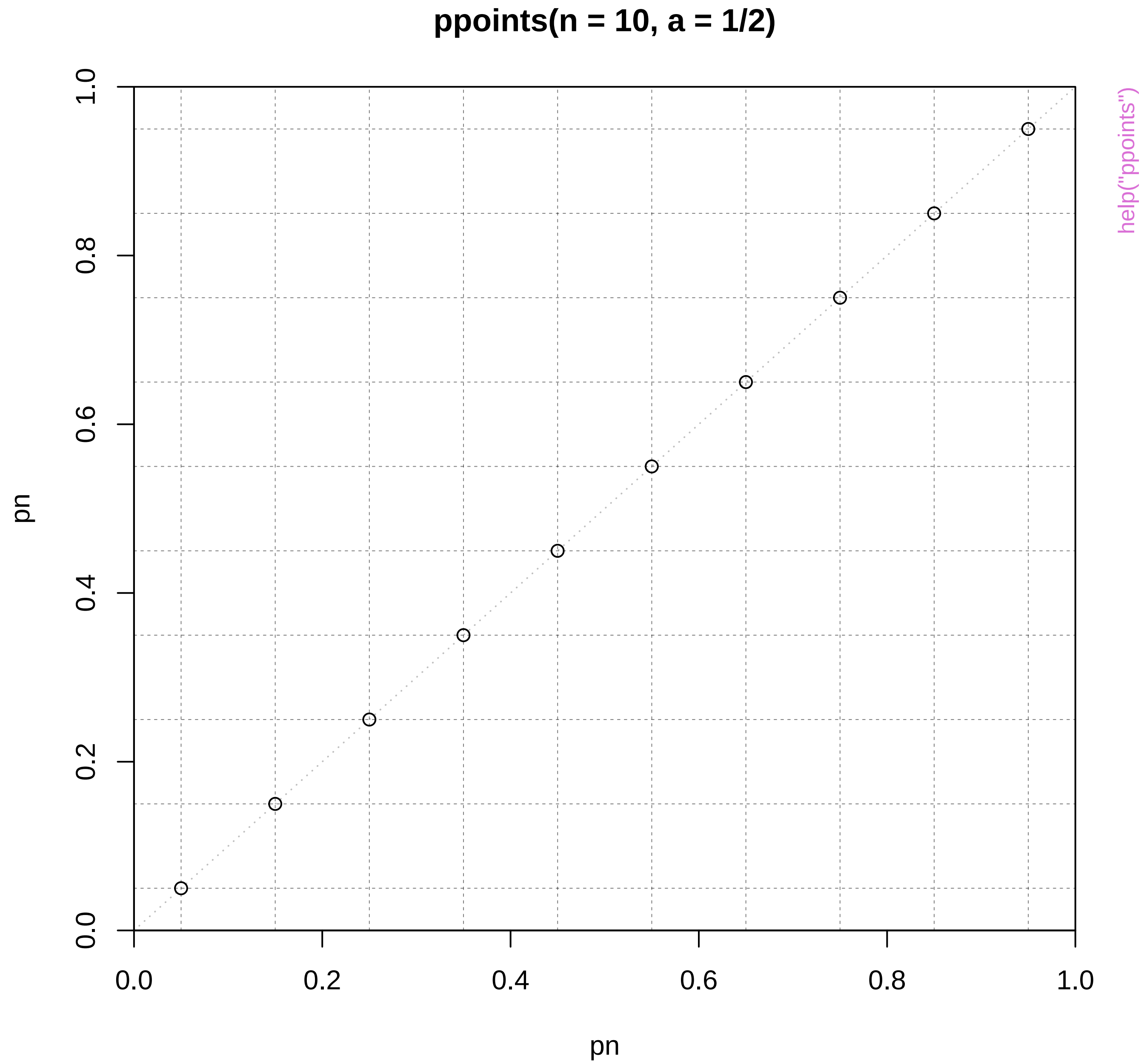

SMI
lag(SMI, 1)
2000
3000
4000
5000
6000
7000
8000
help("plot.ts")



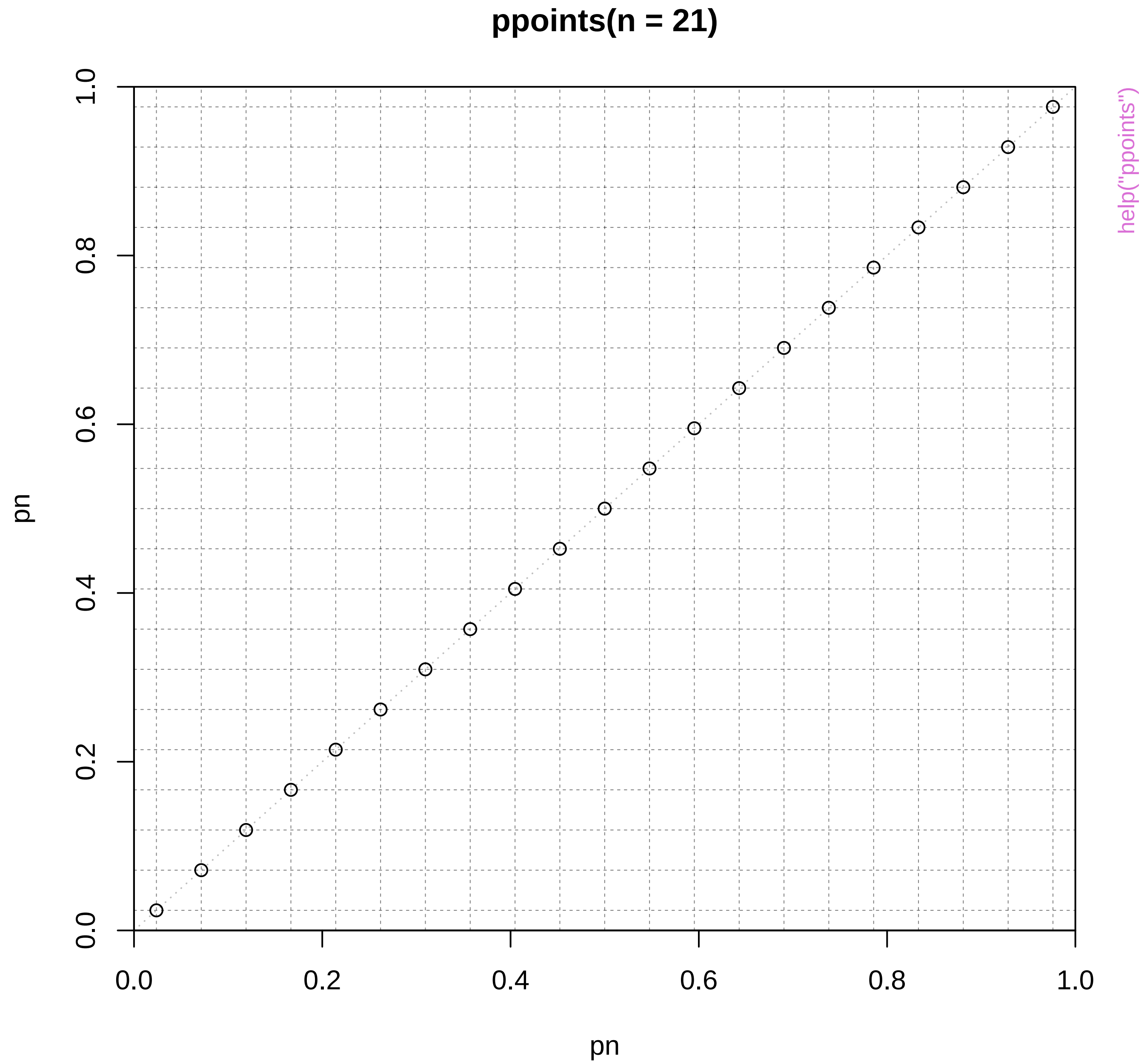
4 weeks lagged SMI stocks -- log scale
SMI
2000
3000
4000
5000
6000
8000
lag(SMI, 20)
help("plot.ts")

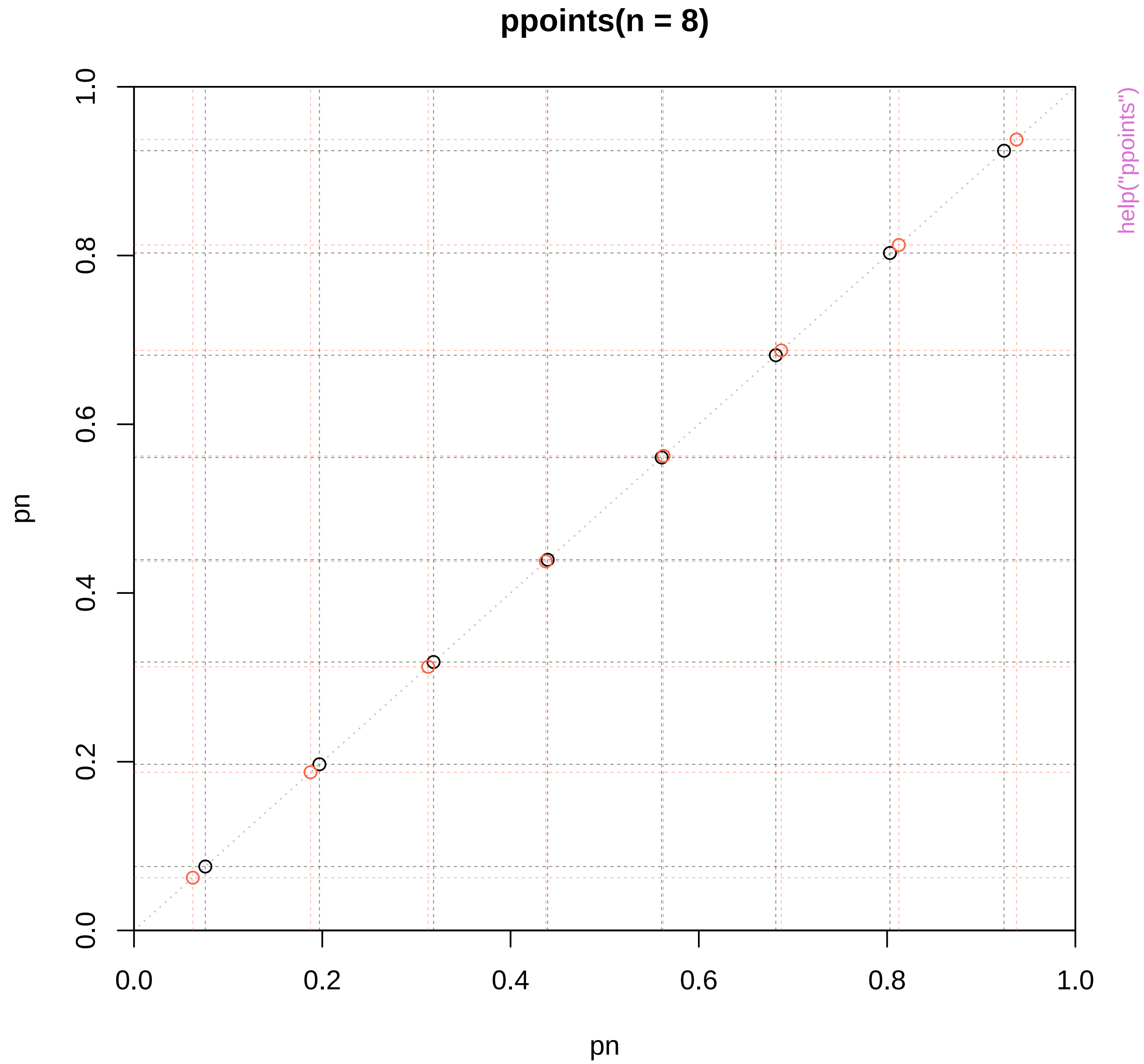

ppoints(n = 4)
0.0
0.2
0.4
0.6
0.8
1.0
pn
help("ppoints")


ppoints(n = 10)
help("ppoints")
pn
pn
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

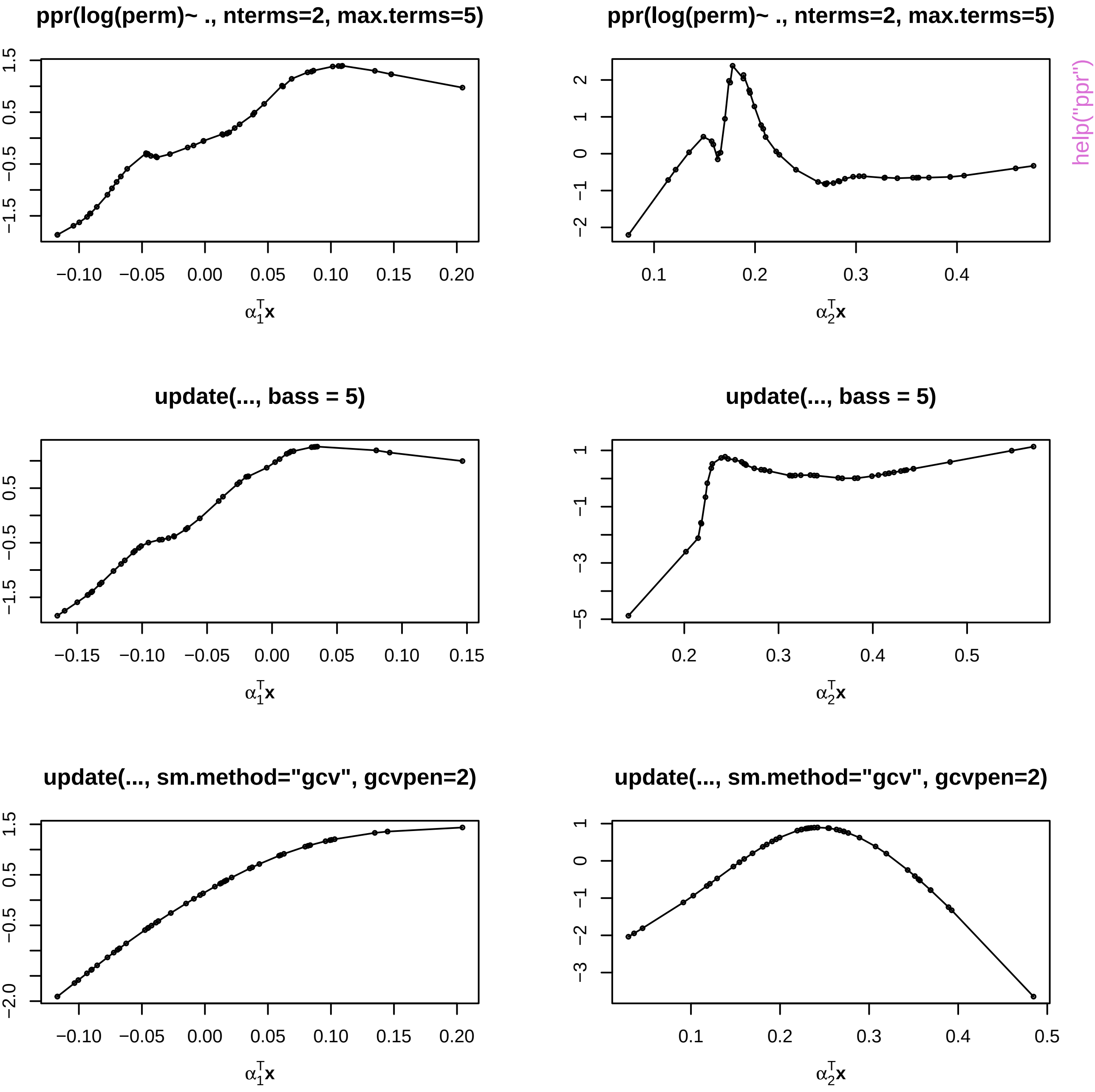


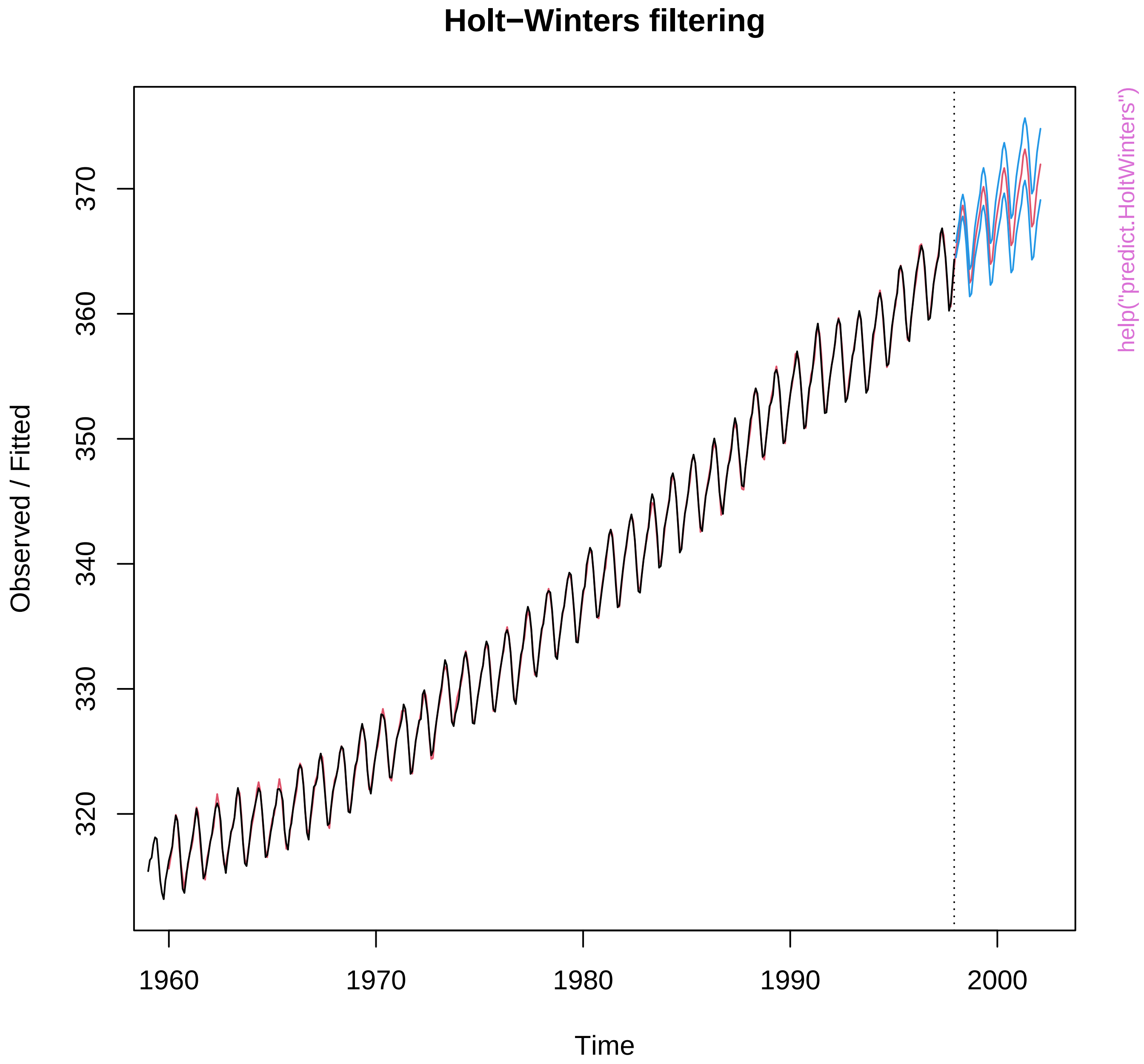
ppoints(n = 10, a = 1/2)
pn
pn
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
help("ppoints")


ppoints(n = 21)
pn
pn
0.0
0.2
0.4
0.6
0.8
1.0
help("ppoints")

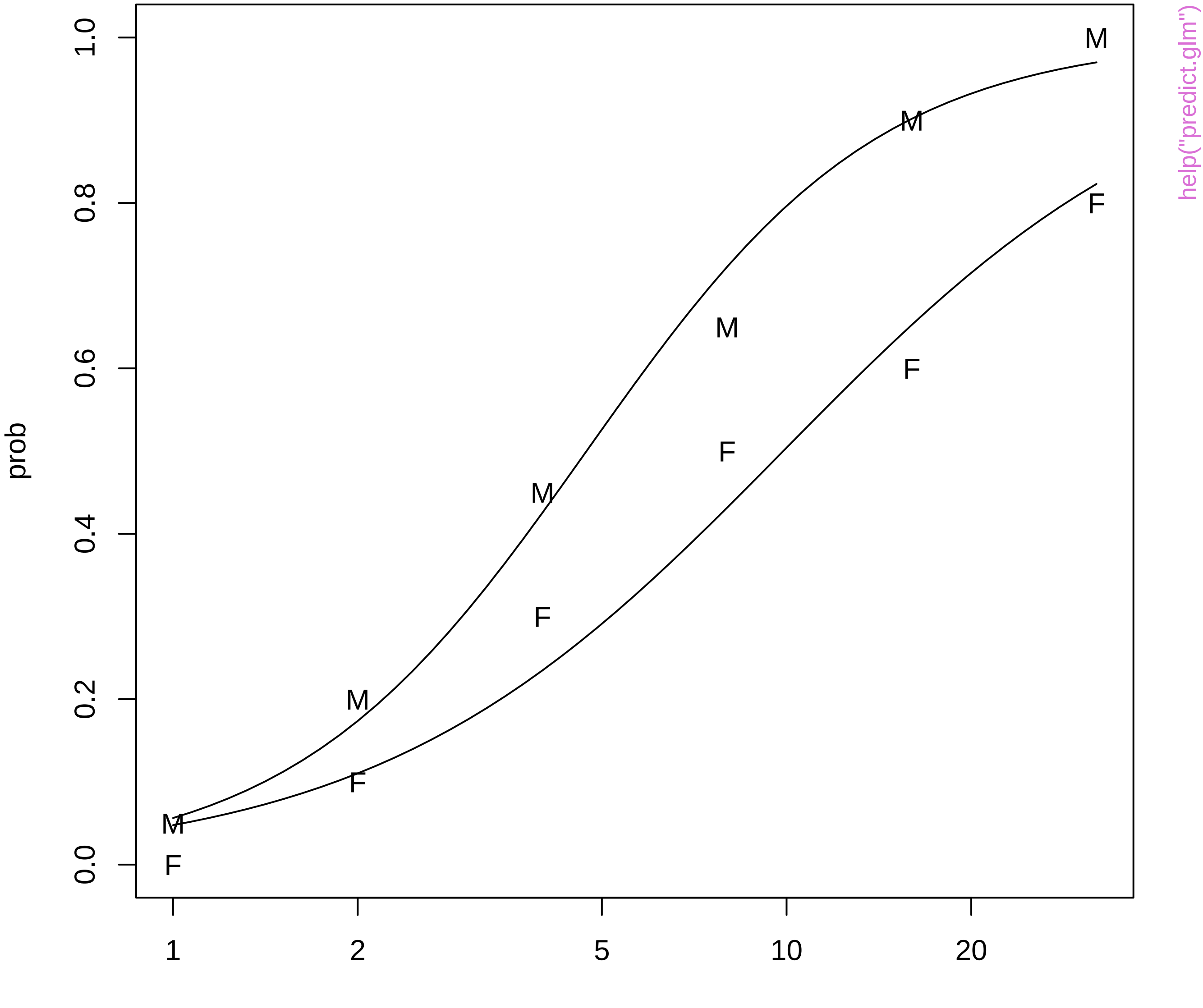


ppoints(n = 8)
0.0
0.2
0.4
0.6
0.8
1.0
pn
pn
help("ppoints")

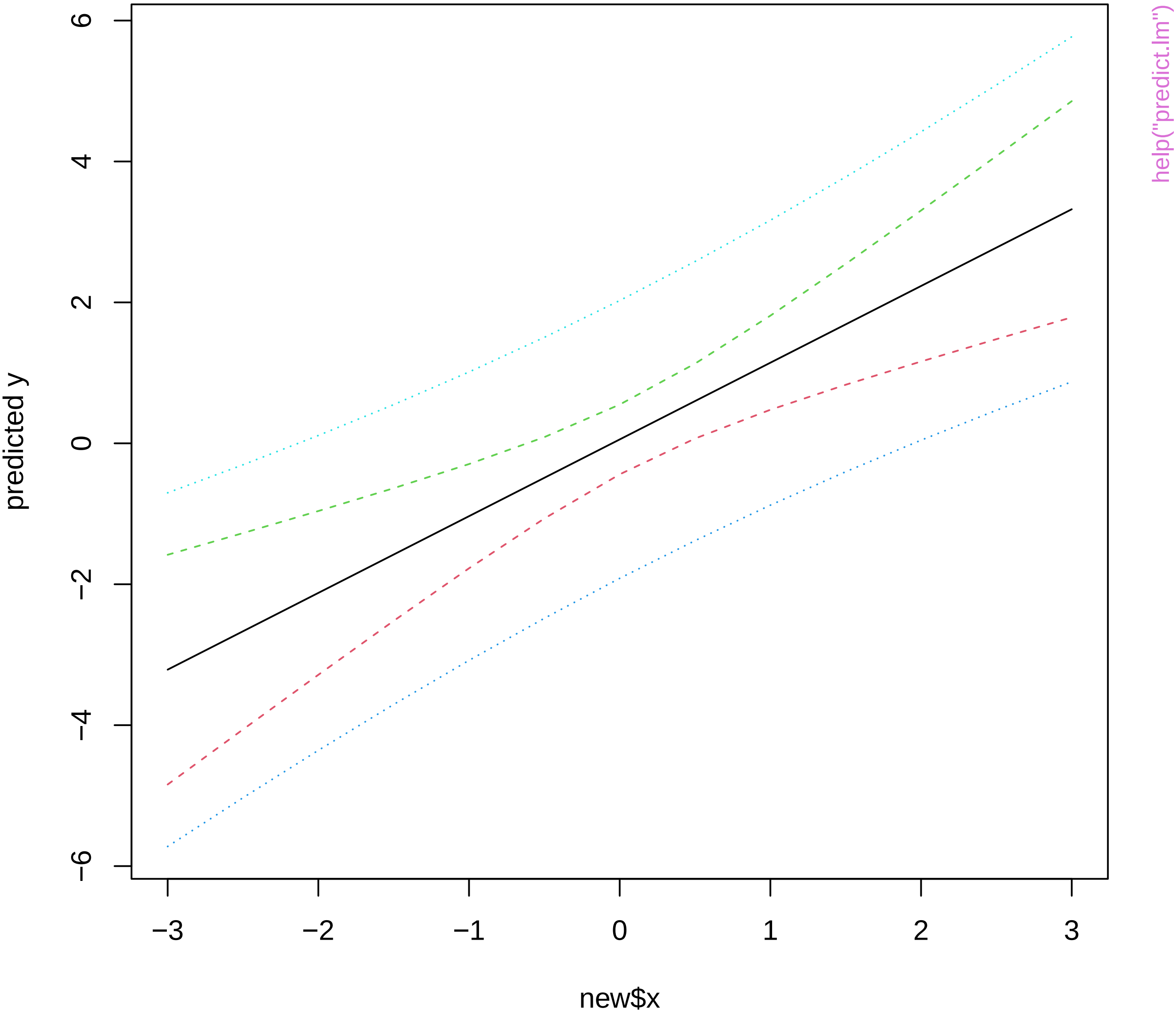

ppr(log(perm)~ ., nterms=2, max.terms=5)
ppr(log(perm)~ ., nterms=2, max.terms=5)
update(..., bass = 5)
update(..., bass = 5)
update(..., sm.method="gcv", gcvpen=2)
update(..., sm.method="gcv", gcvpen=2)
$\alpha_1^T\mathbf{x}$
$\alpha_2^T\mathbf{x}$
−0.10 −0.05 0.00 0.05 0.10 0.15 0.20
−1.5 −0.5 0.5 1.5
0.1 0.2 0.3 0.4
−2 −1 0 1 2
−0.15 −0.10 −0.05 0.00 0.05 0.10 0.15
−1.5 −0.5 0.5
0.2 0.3 0.4 0.5
−5 −3 −1 1
−2.0 −0.5 0.5 1.5
0.1 0.2 0.3 0.4 0.5
−3 −2 −1 0 1
help("ppr")



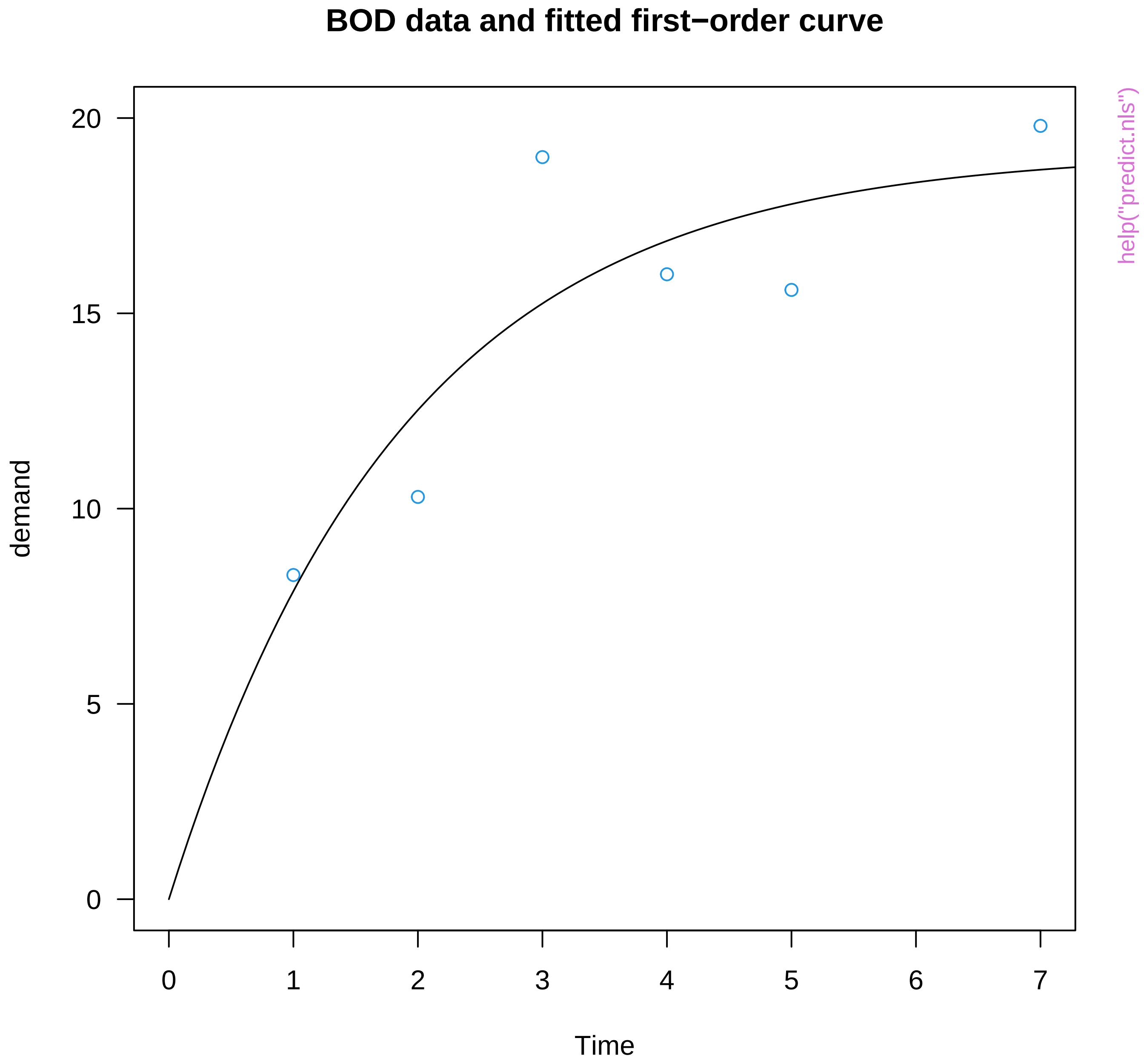
Holt−Winters filtering
Observed / Fitted
Time
320
330
340
350
360
370
1960
1970
1980
1990
2000
help("predict.HoltWinters")

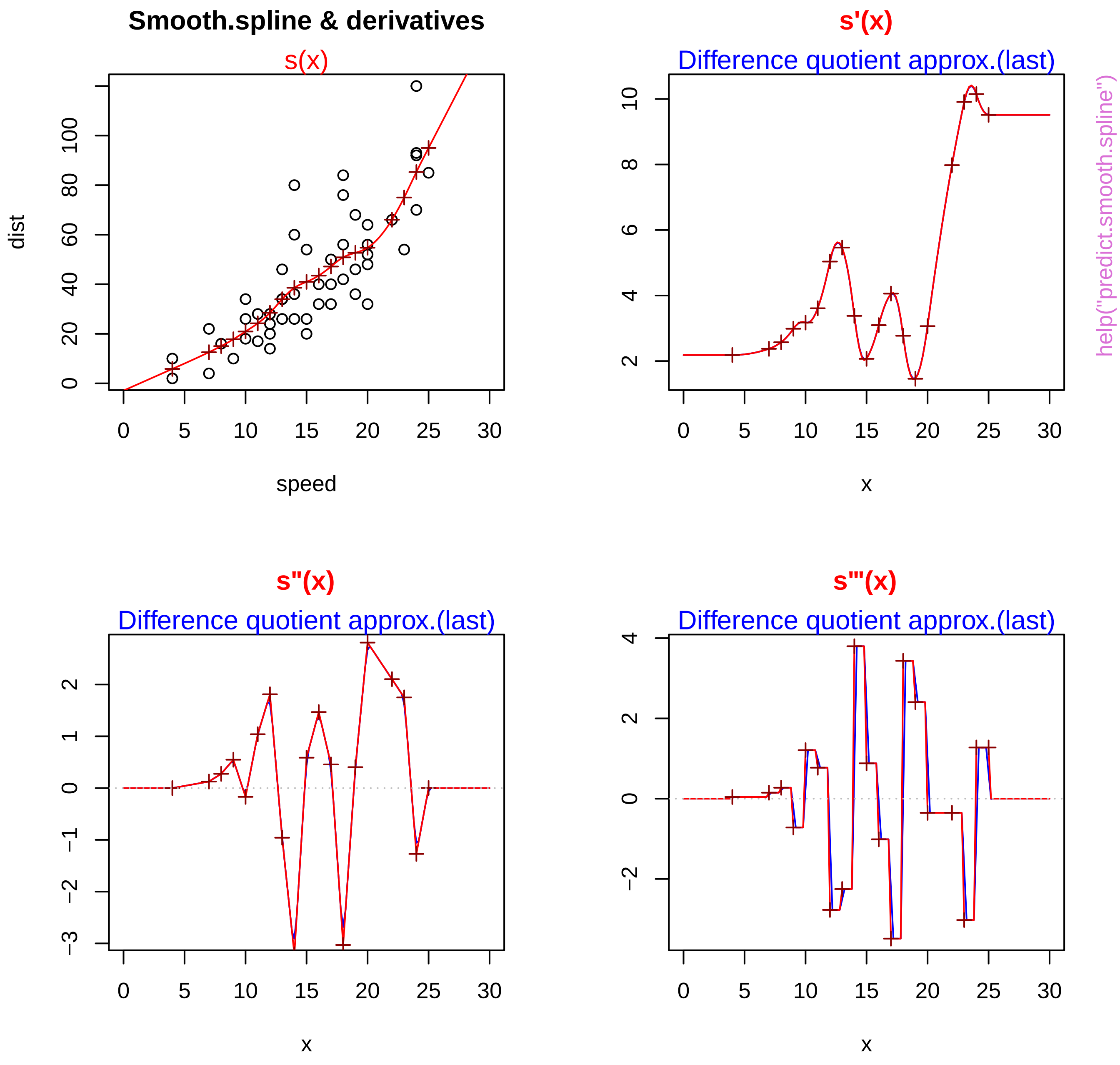


prob
1.0
0.8
0.6
0.4
0.2
0.0
1
2
5
10
20
dose
M
F
help("predict.glm")

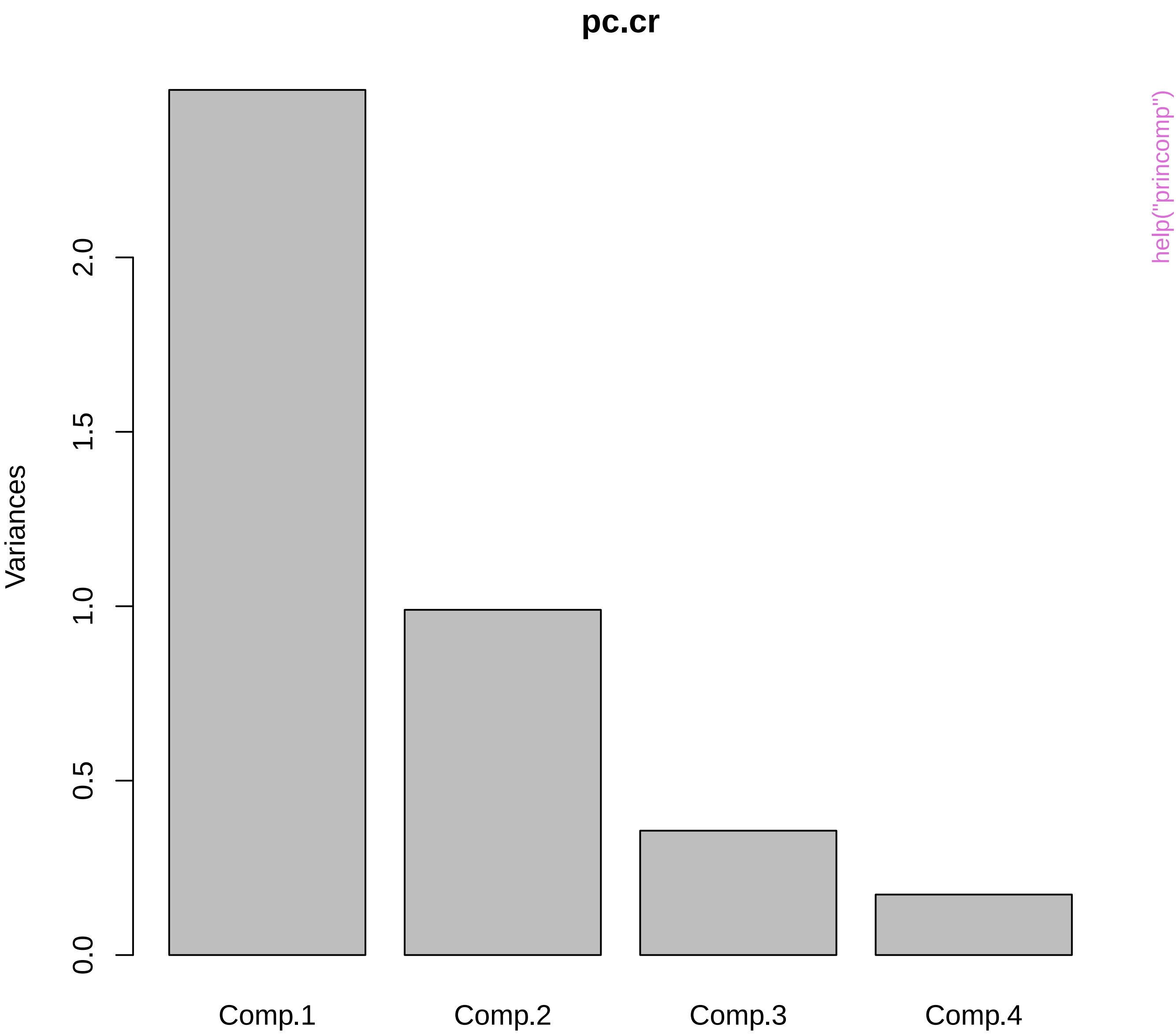


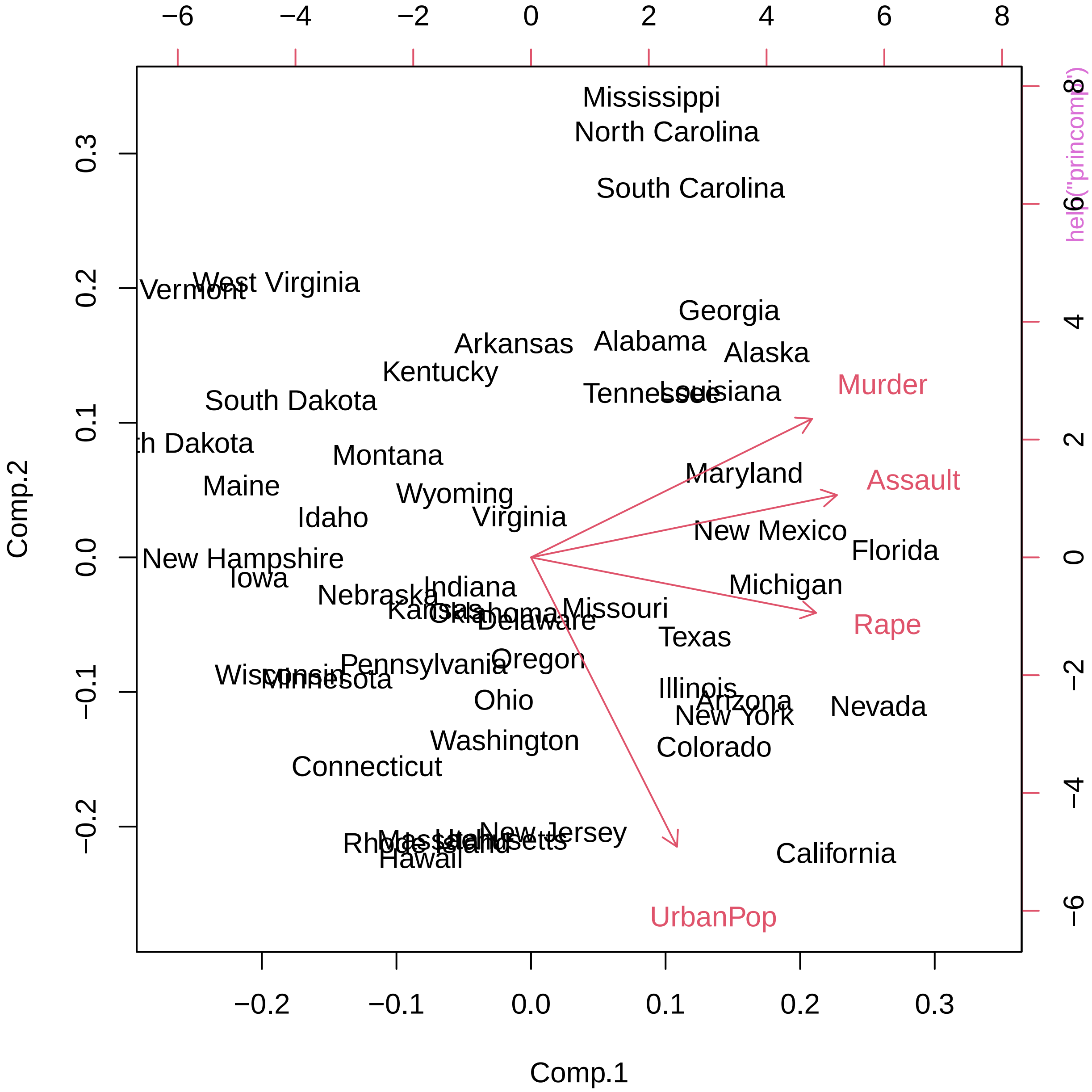
6
4
2
0
−2
−4
−6
predicted y
−3
−2
−1
0
1
2
3
new$x
help("predict.lm")


BOD data and fitted first−order curve
20
15
10
5
0
demand
0
1
2
3
4
5
6
7
Time
help("predict.nls")

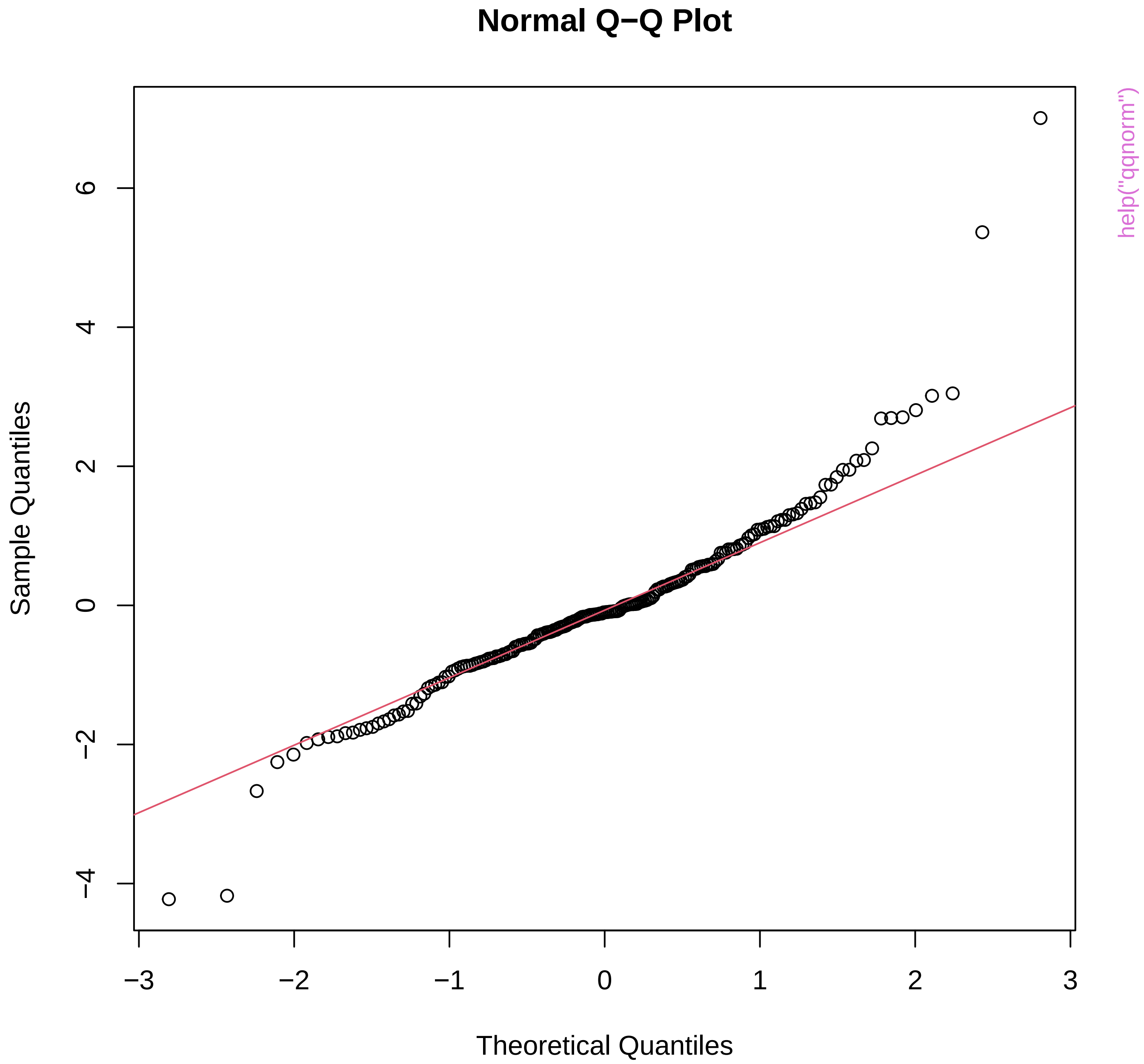


Smooth.spline & derivatives
s(x)
dist
speed
s'(x)
Difference quotient approx.(last)
help("predict.smooth.spline")
s"(x)
Difference quotient approx.(last)
s'''(x)
Difference quotient approx.(last)
x

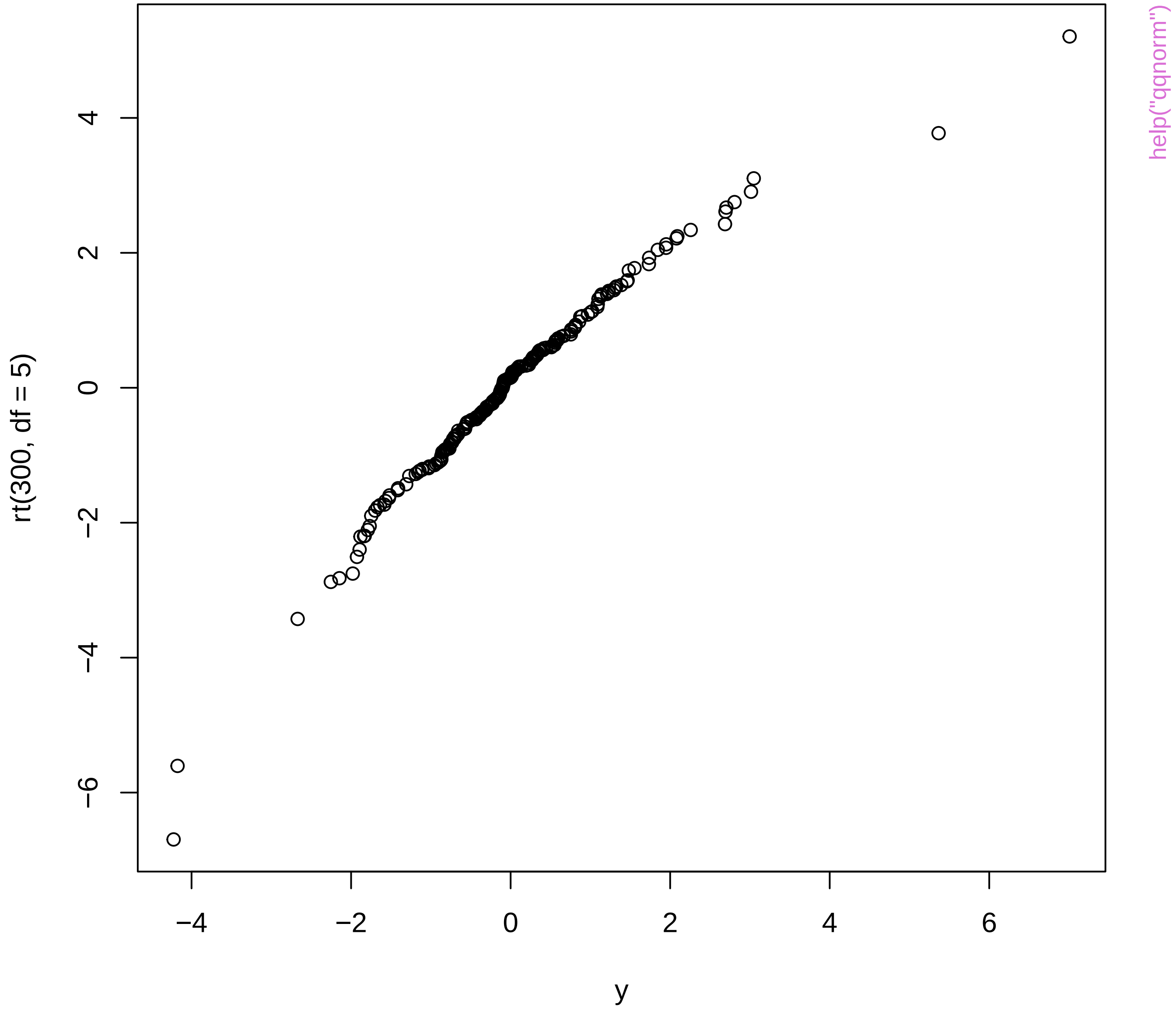

pc.cr
Variances
0.0
0.5
1.0
1.5
2.0
Comp.1
Comp.2
Comp.3
Comp.4
help("princomp")

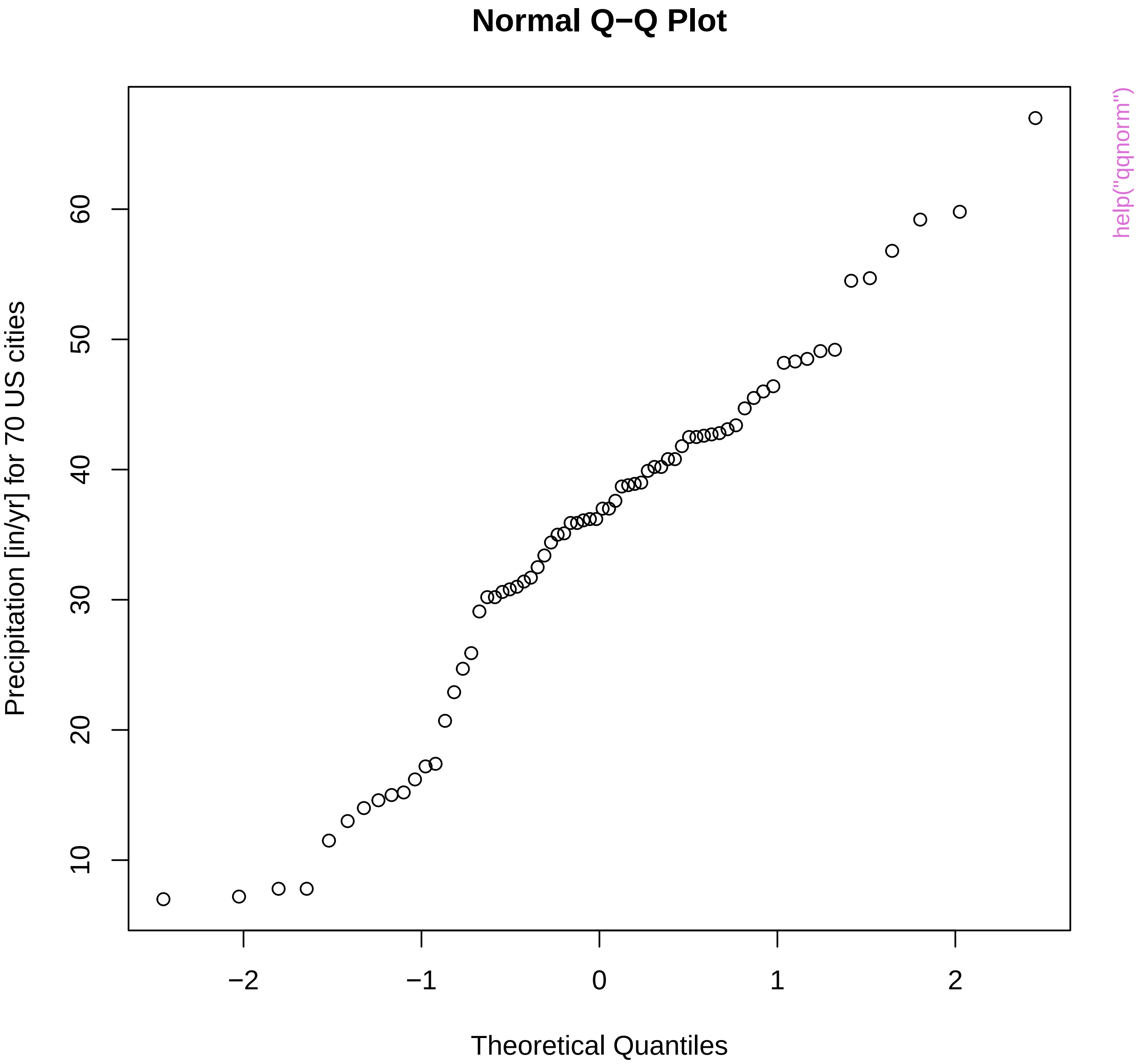


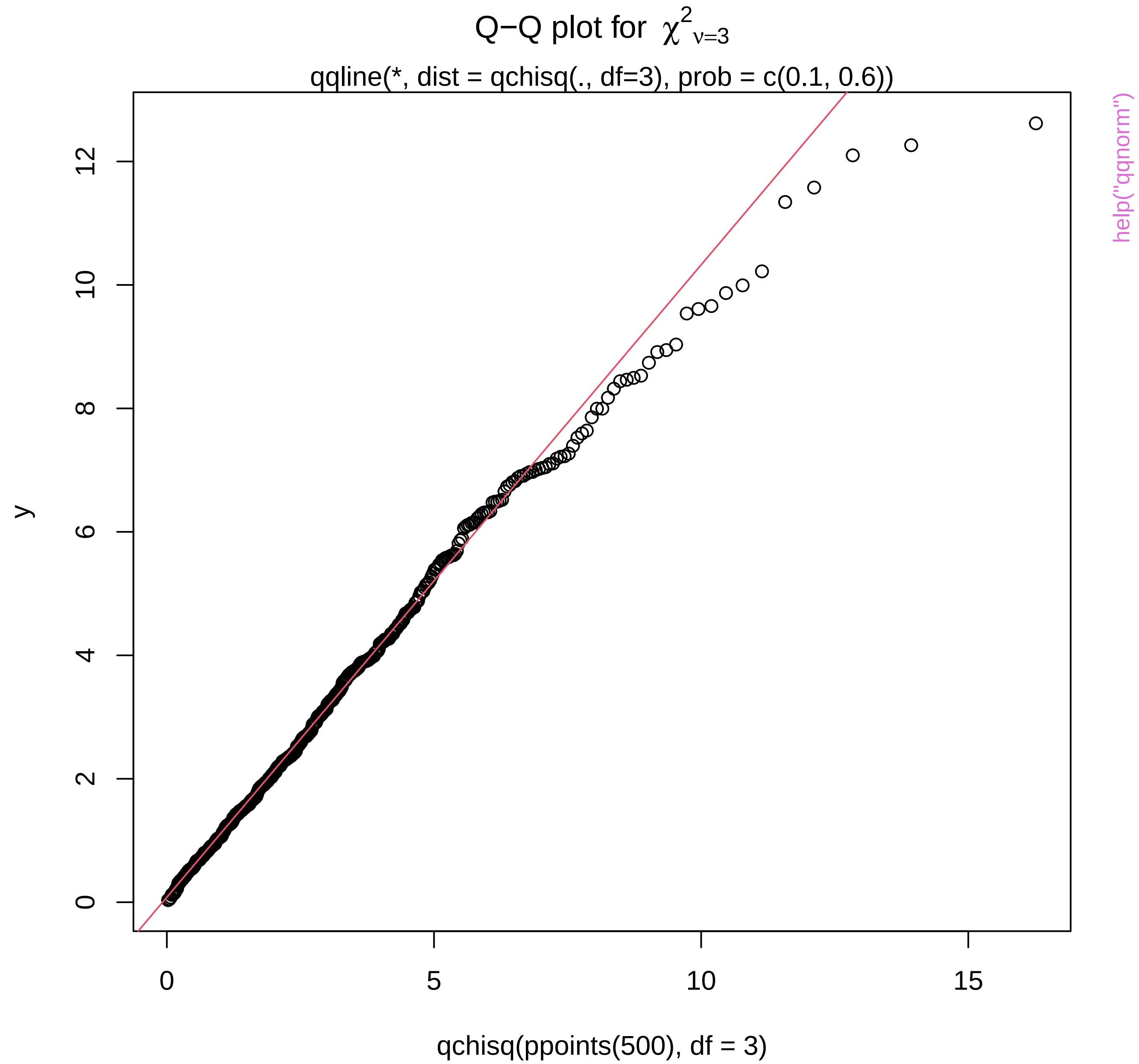
−6
−4
−2
0
2
4
6
8
Mississippi
North Carolina
South Carolina
Vermont
West Virginia
Georgia
Arkansas
Alabama
Alaska
Kentucky
Tennessee
Louisiana
Murder
South Dakota
th Dakota
Montana
Maryland
Assault
Maine
Wyoming
Idaho
Virginia
New Mexico
Florida
New Hampshire
Iowa
Indiana
Michigan
Nebraska
Kansas
Oklahoma
Missouri
Delaware
Rape
Texas
Oregon
Pennsylvania
Wisconsin
Minnesota
Illinois
Arizona
Ohio
New York
Nevada
Washington
Colorado
Connecticut
New Jersey
Rhode Island
Massachusetts
Utah
Hawaii
California
UrbanPop
0.3
0.2
0.1
0.0
−0.1
−0.2
Comp.2
−0.2
−0.1
0.0
0.1
0.2
0.3
Comp.1
help("princomp")



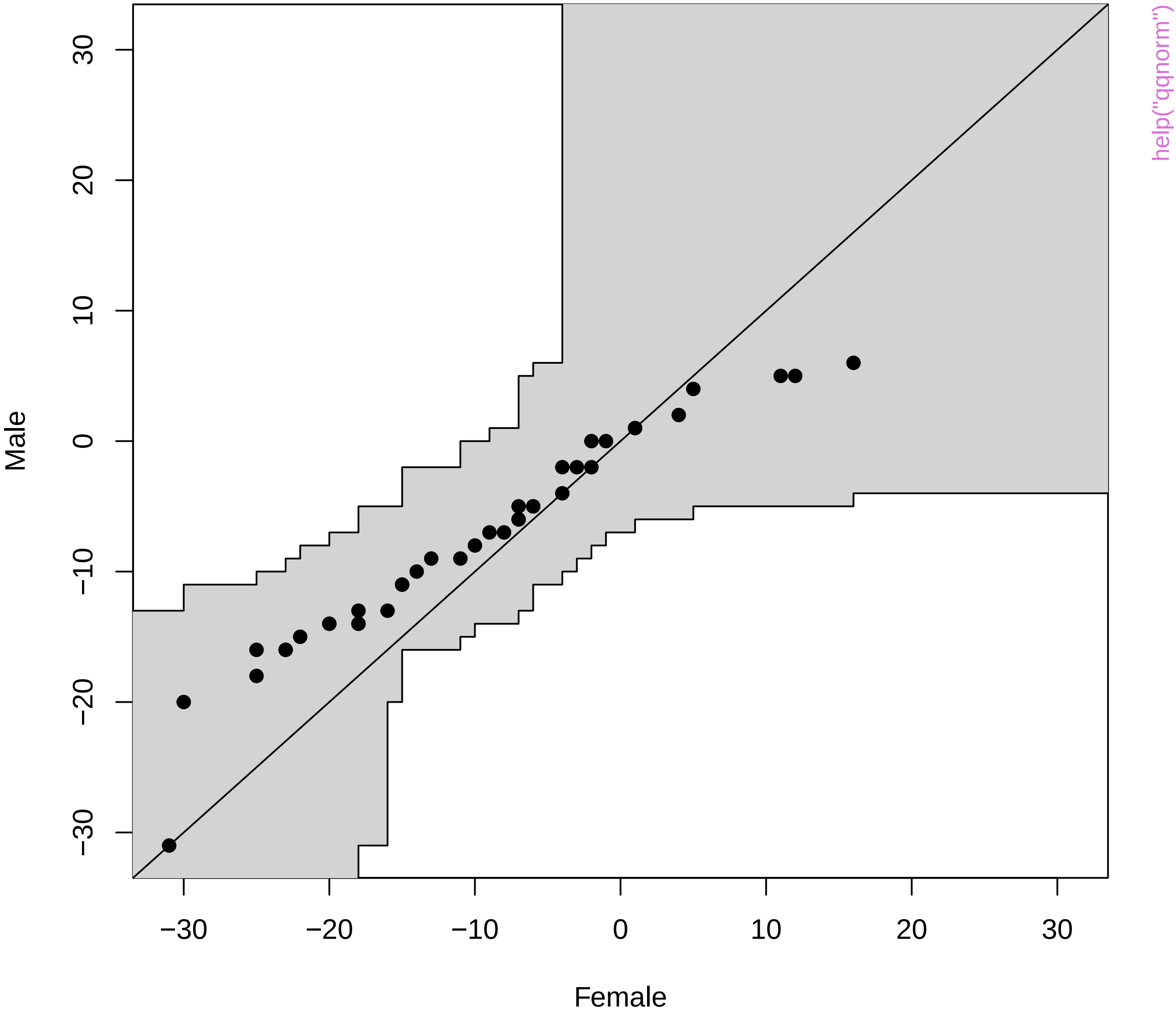
Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
−4
−2
0
2
4
6
−3
−2
−1
0
1
2
3
help("qqnorm")


rt(300, df = 5)
4
2
0
−2
−4
−6
−4
−2
0
2
4
6
y
help("qqnorm")

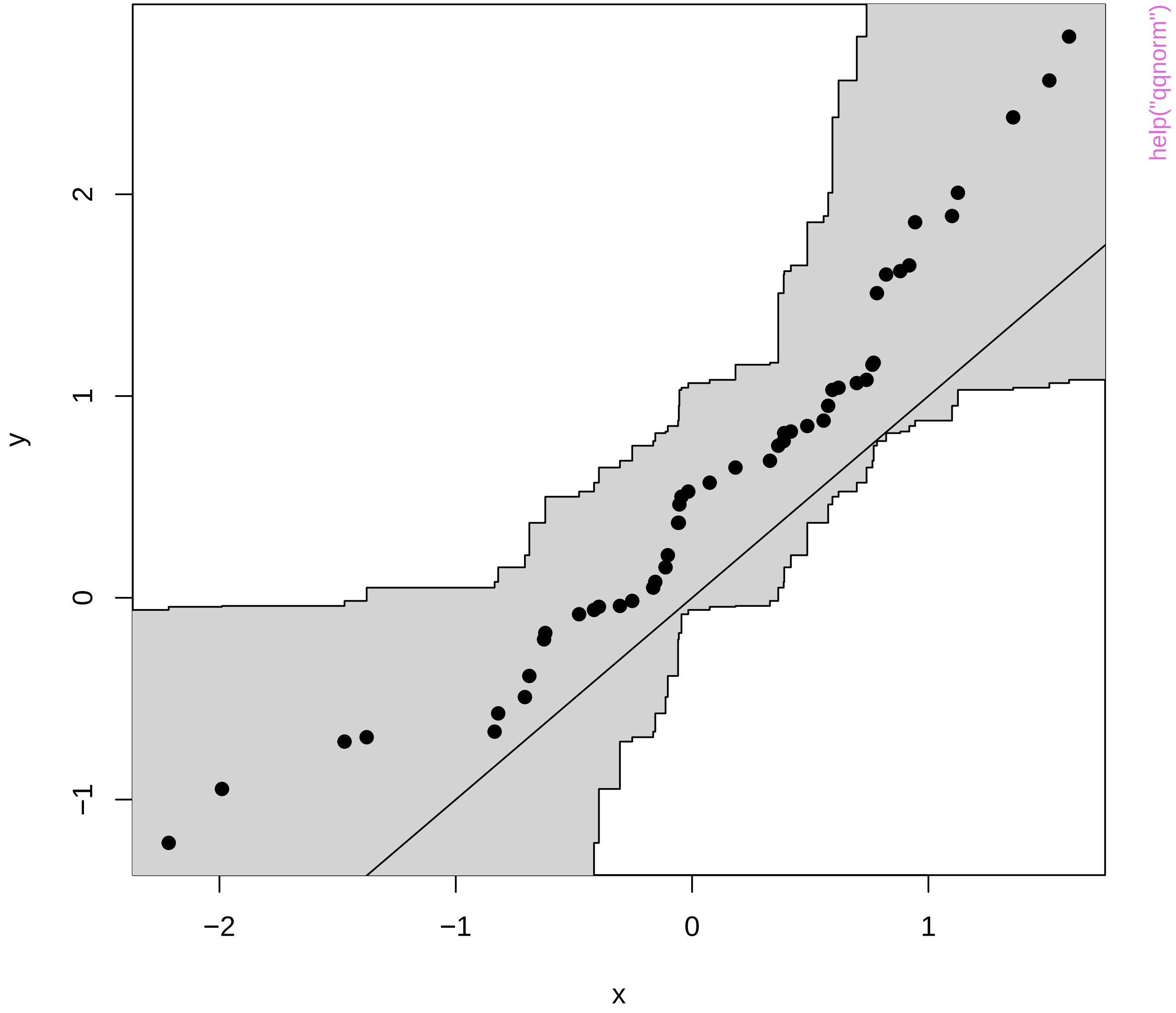

Normal Q−Q Plot
Precipitation [in/yr] for 70 US cities
Theoretical Quantiles
−2
−1
0
1
2
10
20
30
40
50
60
help("qqnorm")



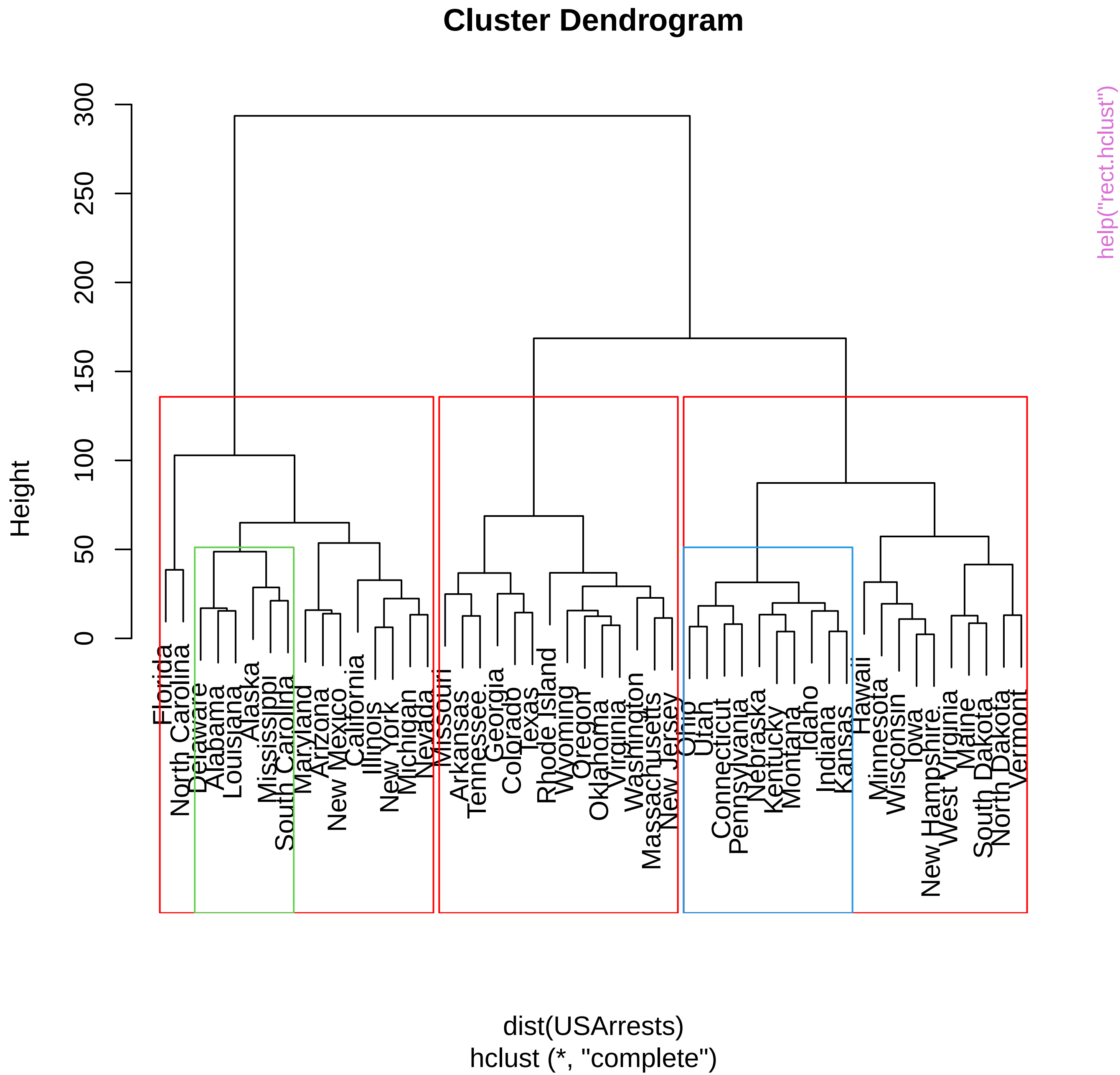
Q−Q plot for $\chi^2_{\nu=3}$
qqline(*, dist = qchisq(., df=3), prob = c(0.1, 0.6))
help("qqnorm")
y
0
2
4
6
8
10
12
0
5
10
15
qchisq(ppoints(500), df = 3)

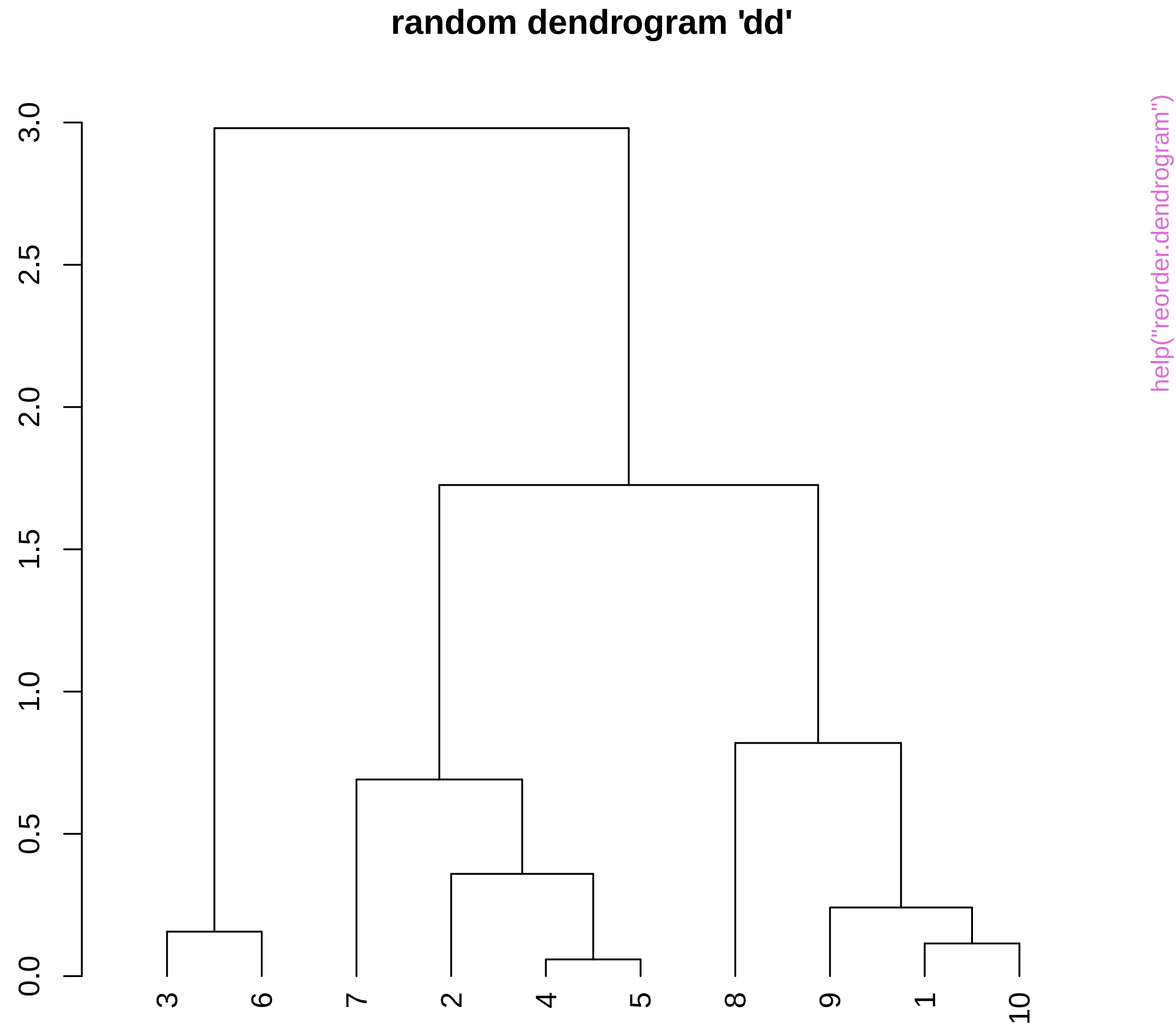


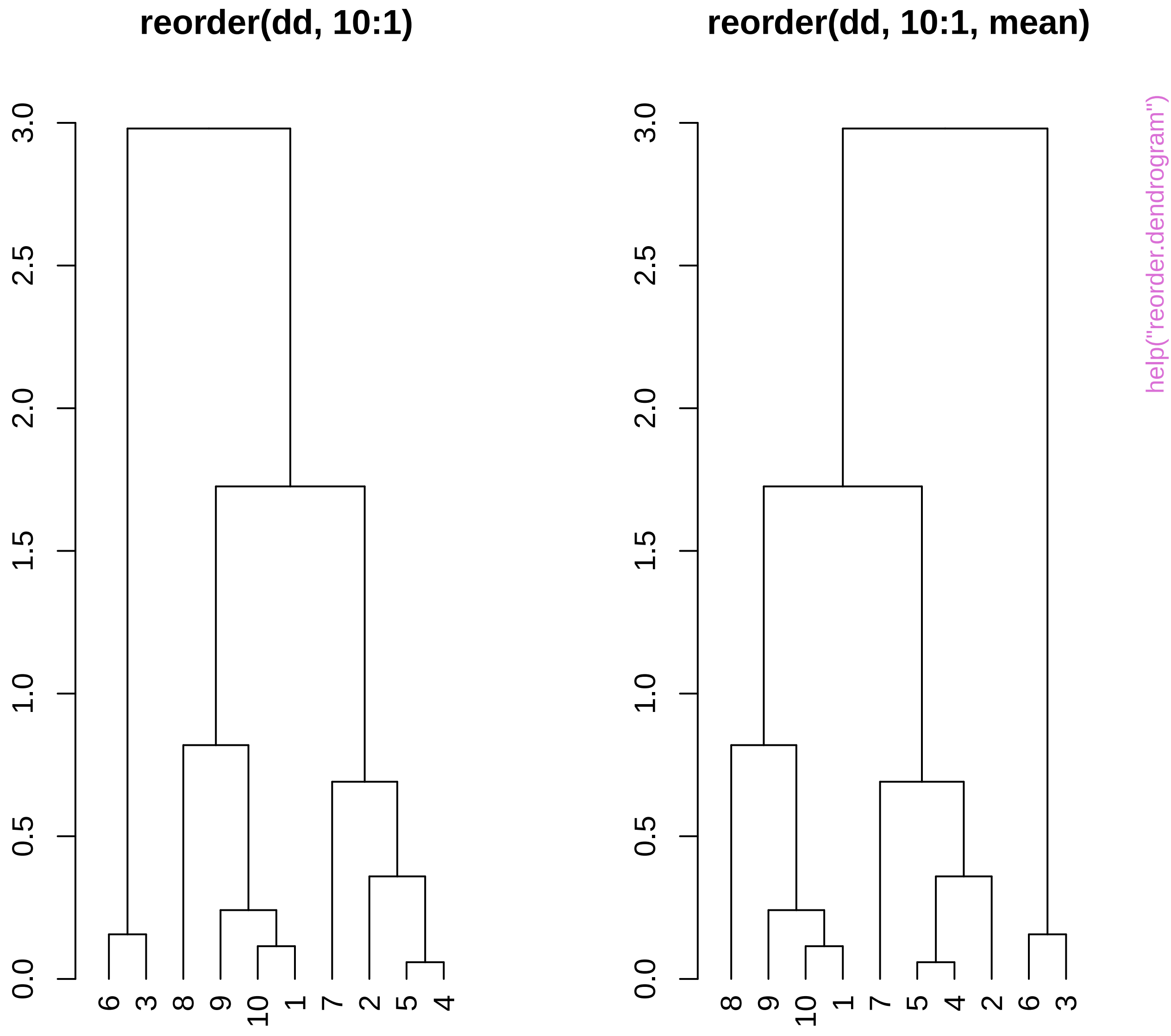
30
20
10
0
−10
−20
−30
Male
−30
−20
−10
0
10
20
30
Female
help("qqnorm")

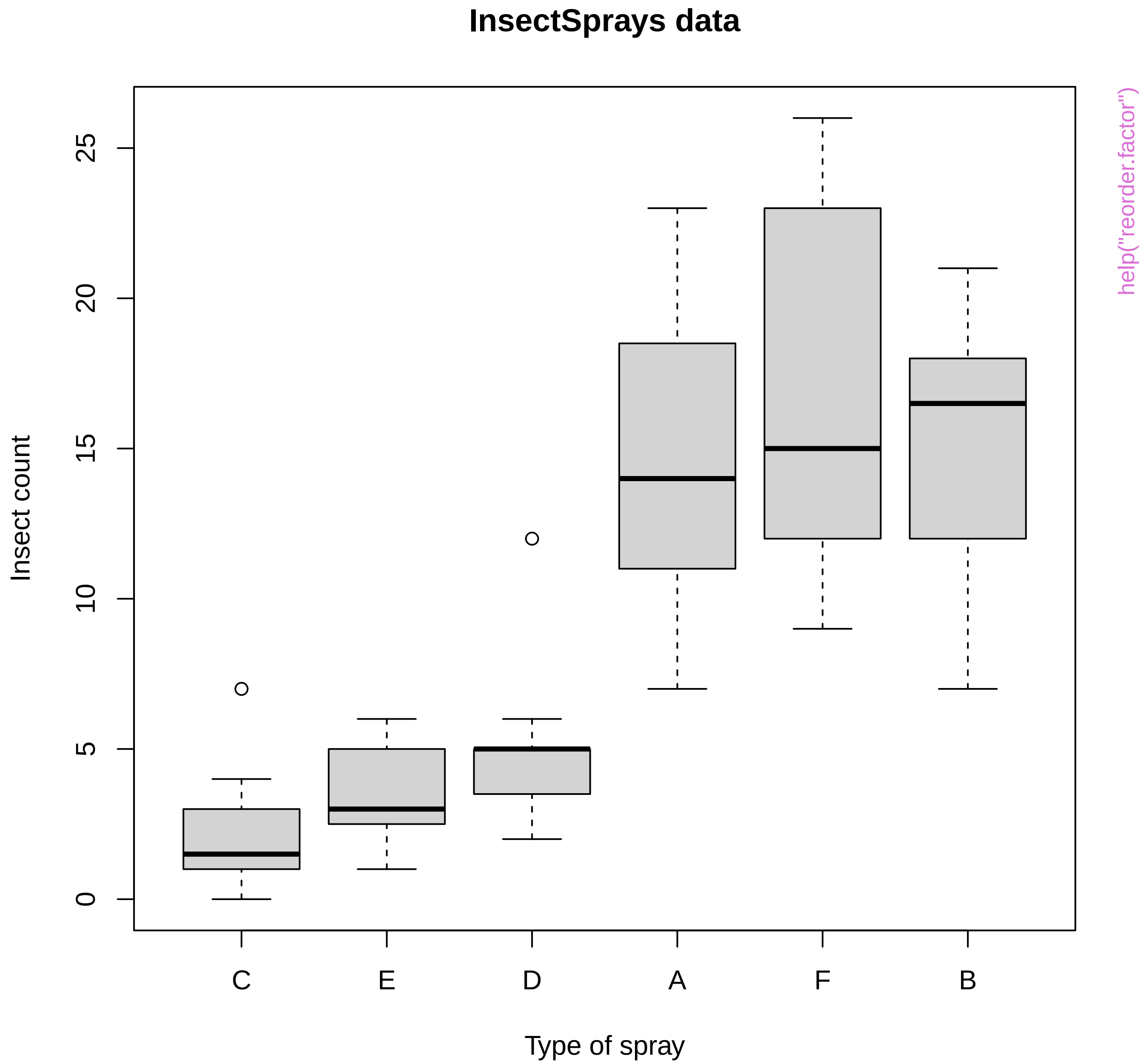


y
x
−2
−1
0
1
2
help("qqnorm")

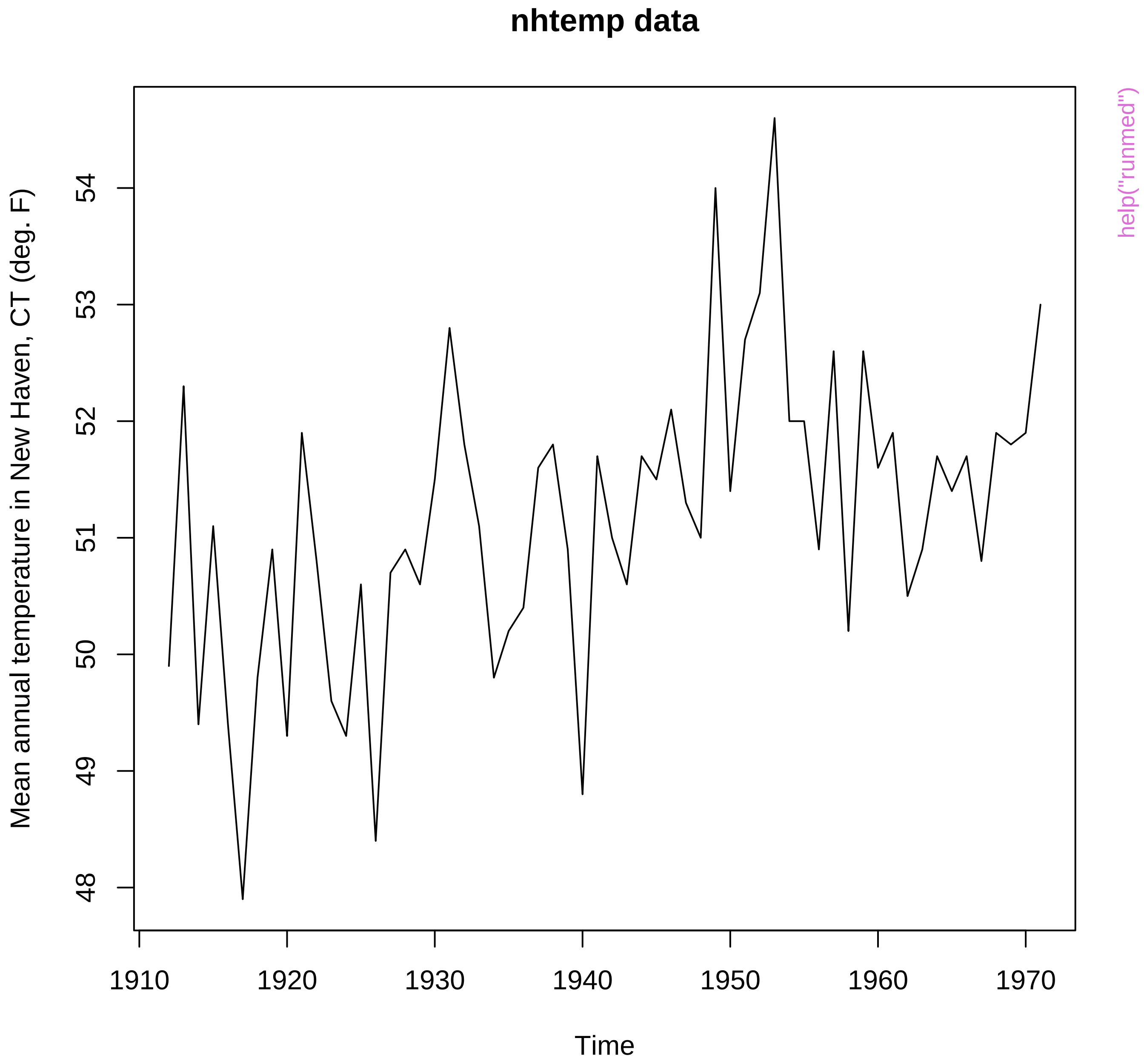

Cluster Dendrogram
Height
300
250
200
150
100
50
0
help("rect.hclust")
Florida
North Carolina
Delaware
Alabama
Louisiana
Alaska
Mississippi
South Carolina
Maryland
Arizona
New Mexico
California
Illinois
New York
Michigan
Nevada
Missouri
Arkansas
Tennessee
Georgia
Colorado
Texas
Rhode Island
Wyoming
Oregon
Oklahoma
Virginia
Washington
Massachusetts
New Jersey
Ohio
Utah
Connecticut
Pennsylvania
Nebraska
Kentucky
Montana
Idaho
Indiana
Kansas
Hawaii
Minnesota
Wisconsin
Iowa
New Hampshire
West Virginia
Maine
South Dakota
North Dakota
Vermont
dist(USArrests)
hclust (*, "complete")

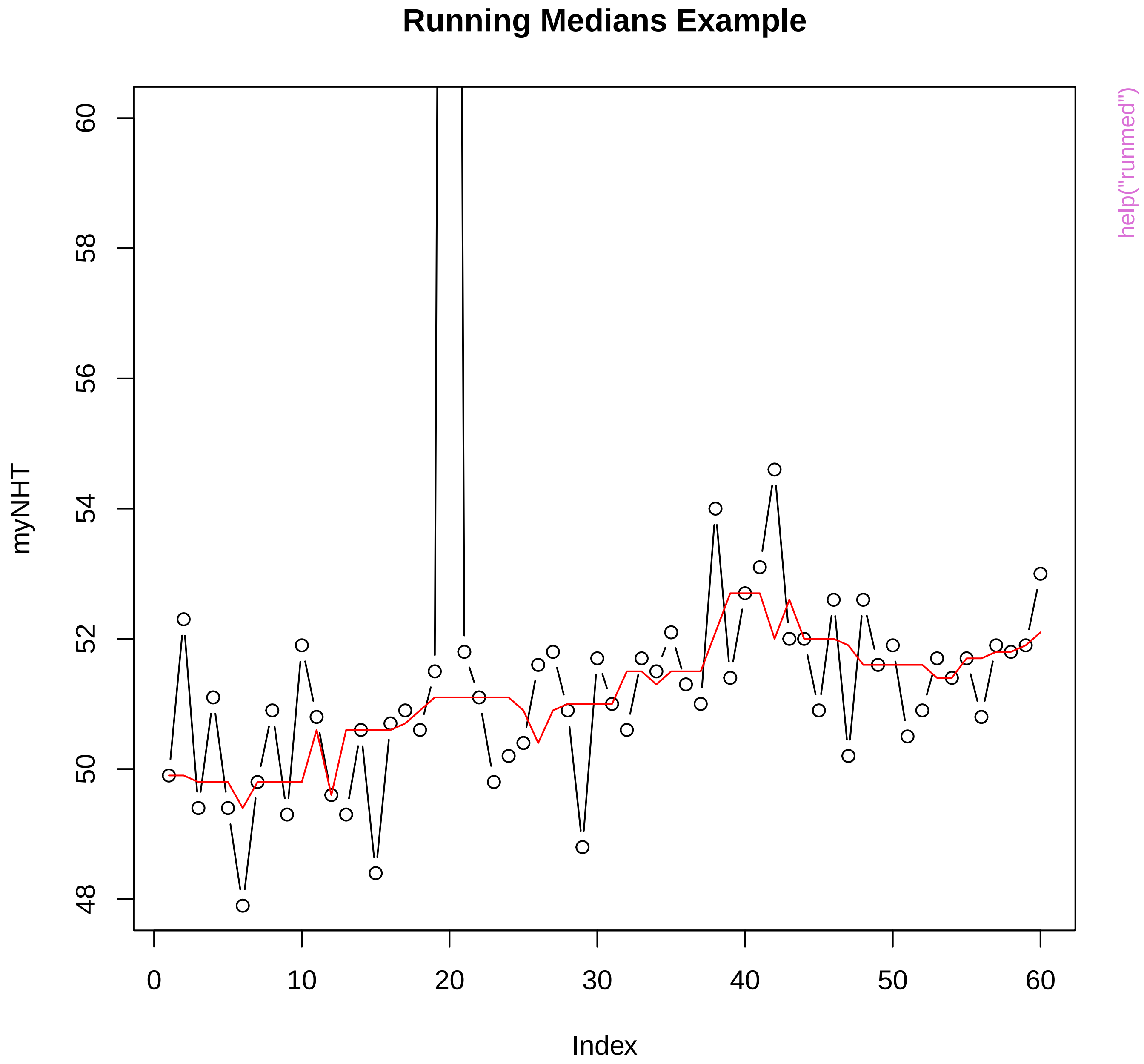

random dendrogram 'dd'
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3
6
7
2
4
5
8
9
1
10
help("reorder.dendrogram")

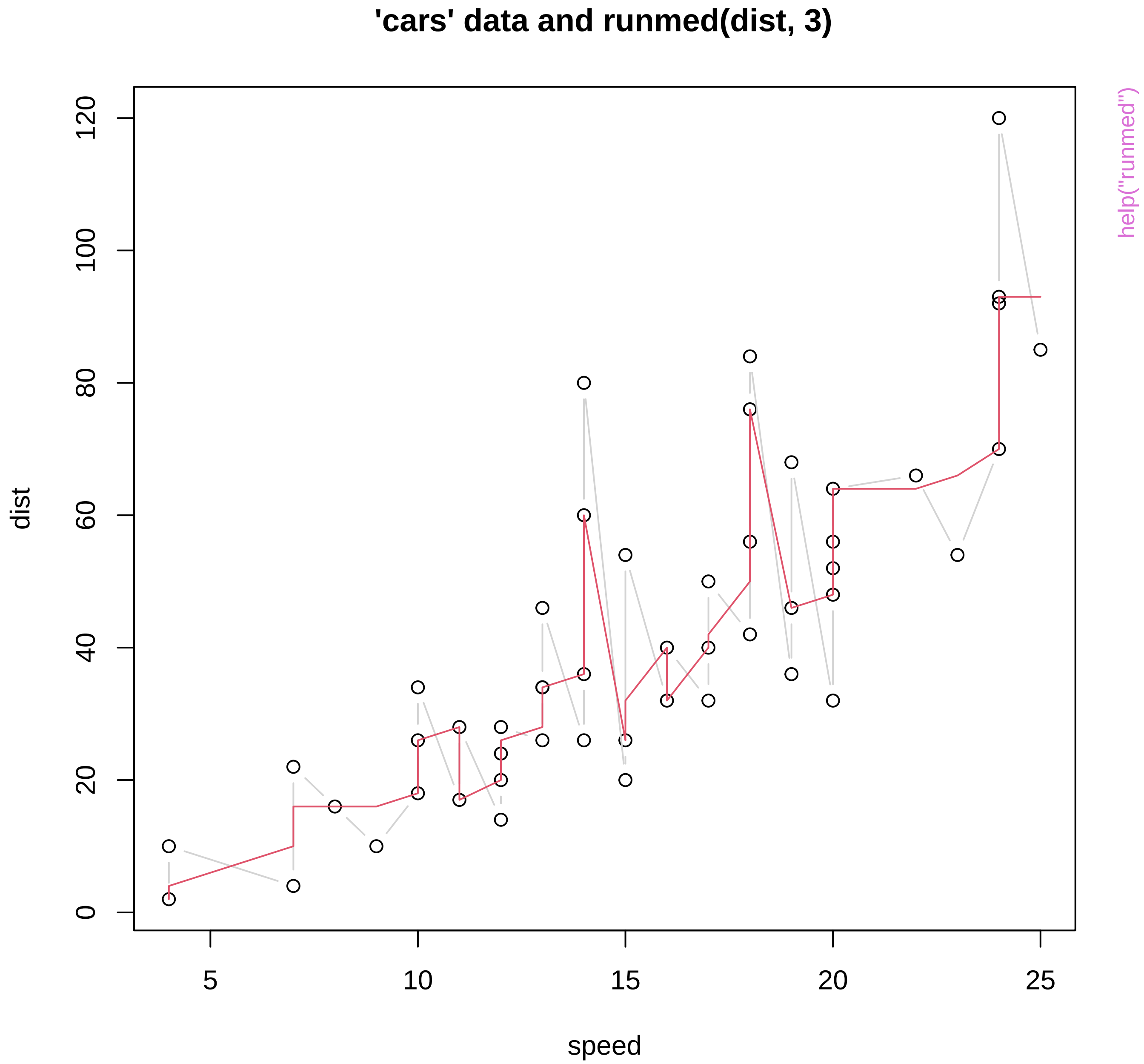

reorder(dd, 10:1)
reorder(dd, 10:1, mean)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
6 3 8 9 10 1 7 2 5 4
8 9 10 1 7 5 4 2 6 3
help("reorder.dendrogram")

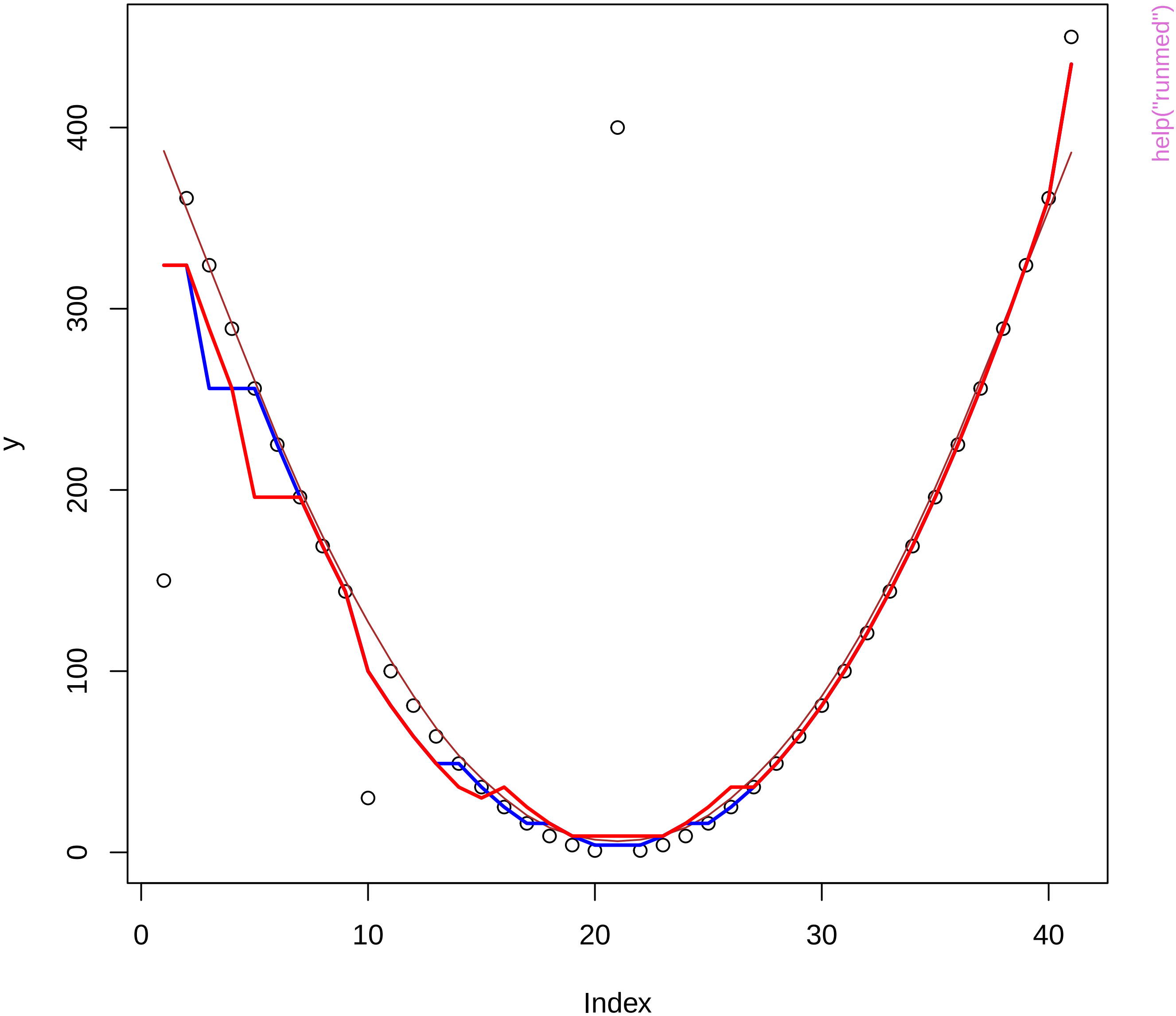


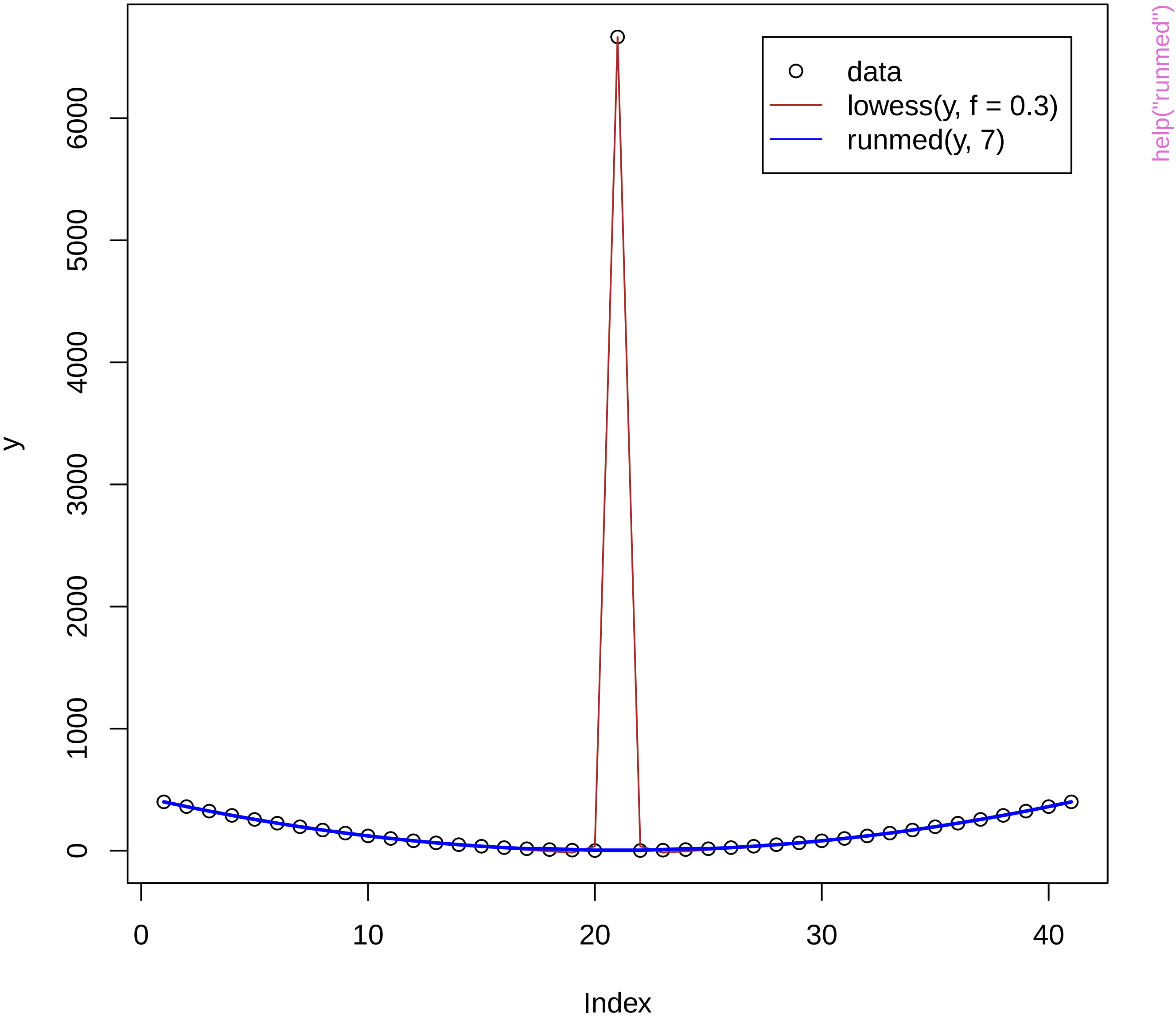
InsectSprays data
Insect count
0
5
10
15
20
25
C
E
D
A
F
B
Type of spray
help("reorder.factor")


nhtemp data
Mean annual temperature in New Haven, CT (deg. F)
48
49
50
51
52
53
54
1910
1920
1930
1940
1950
1960
1970
Time
help("runmed")

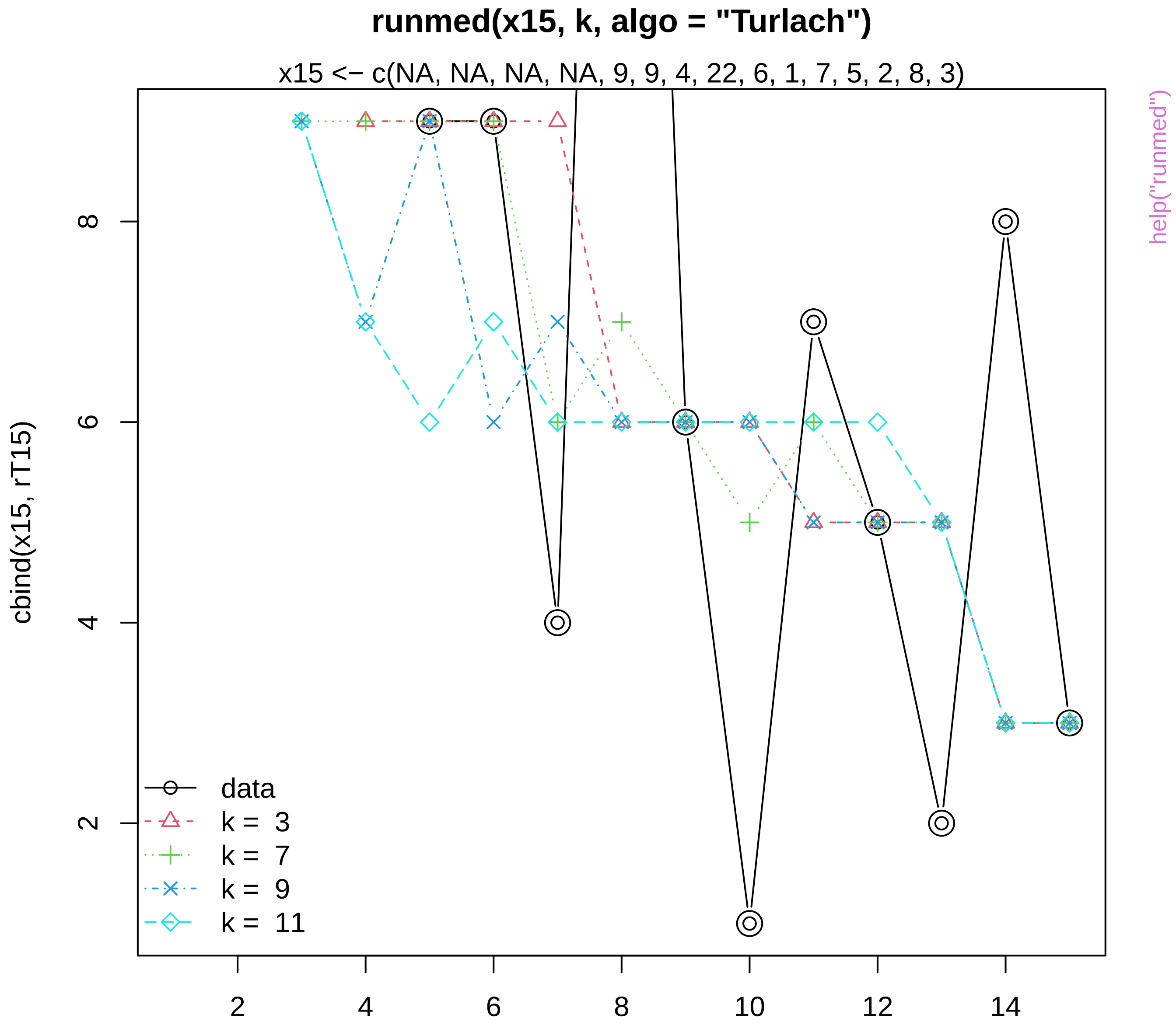

Running Medians Example
myNHT
Index
48
50
52
54
56
58
60
0
10
20
30
40
50
60
help("runmed")

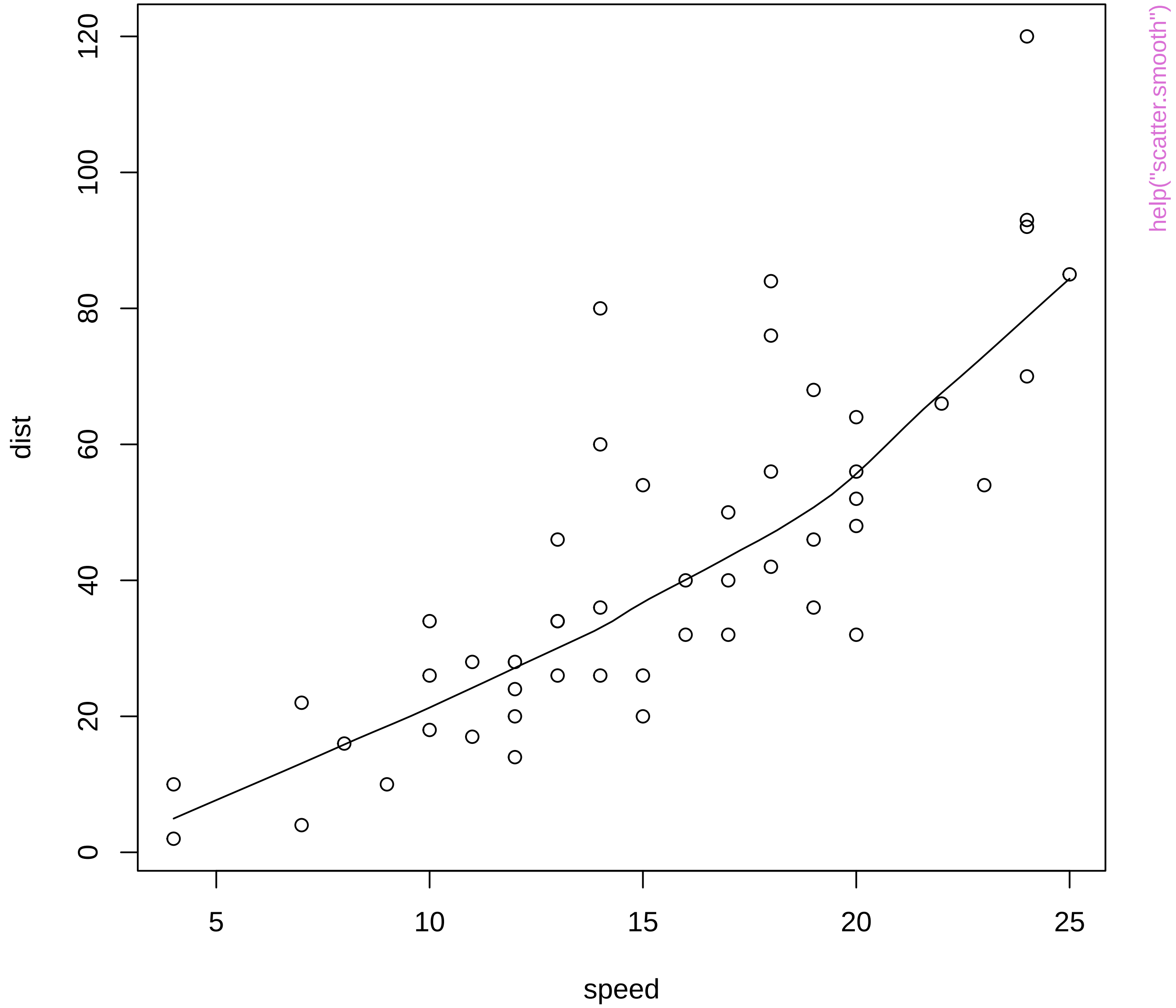

'cars' data and runmed(dist, 3)
dist
speed
0
20
40
60
80
100
120
5
10
15
20
25
help("runmed")


help("runmed")
y
0
100
200
300
400
0
10
20
30
40
Index

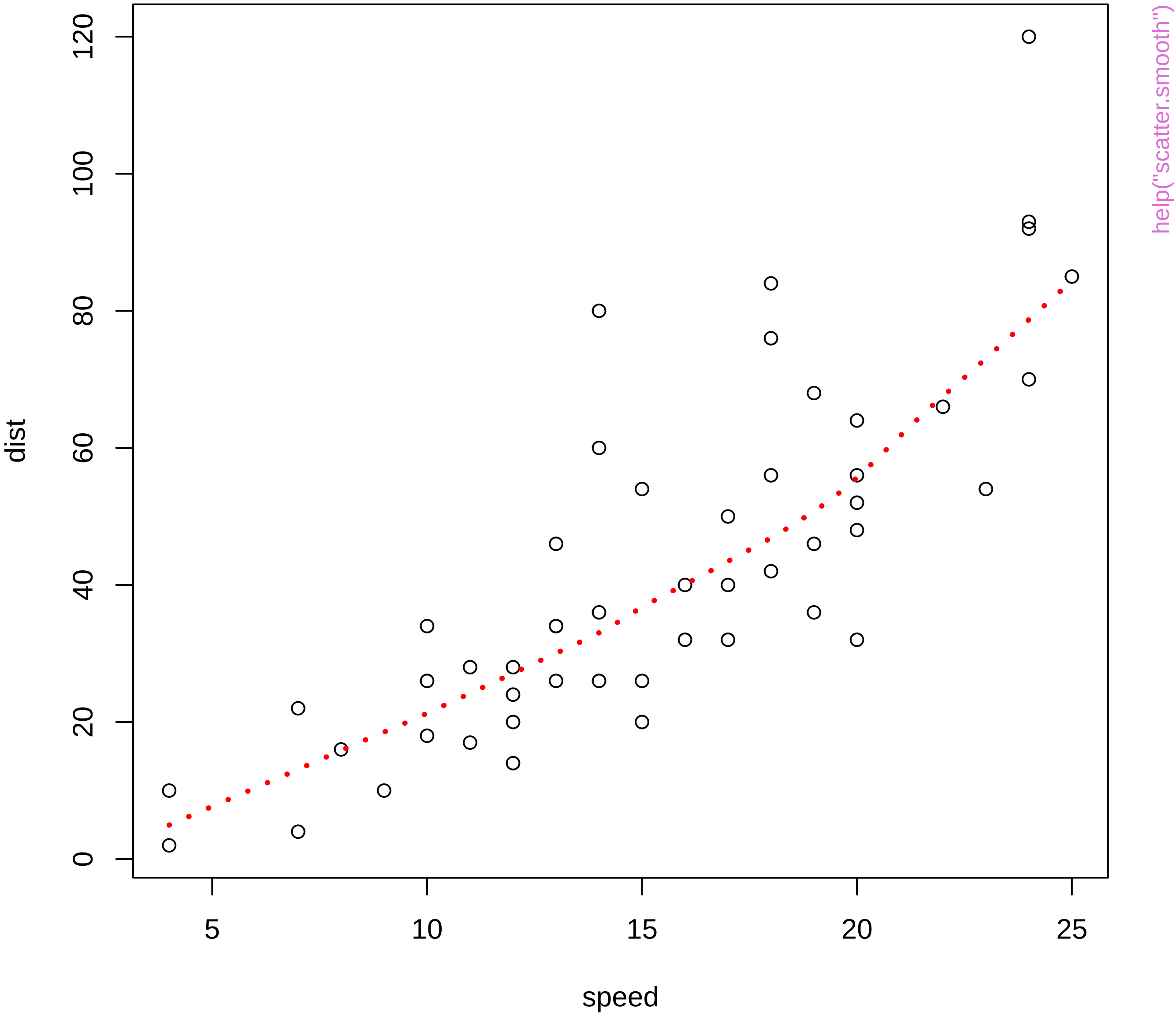


data
lowess(y, f = 0.3)
runmed(y, 7)
y
0
1000
2000
3000
4000
5000
6000
0
10
20
30
40
Index
help("runmed")

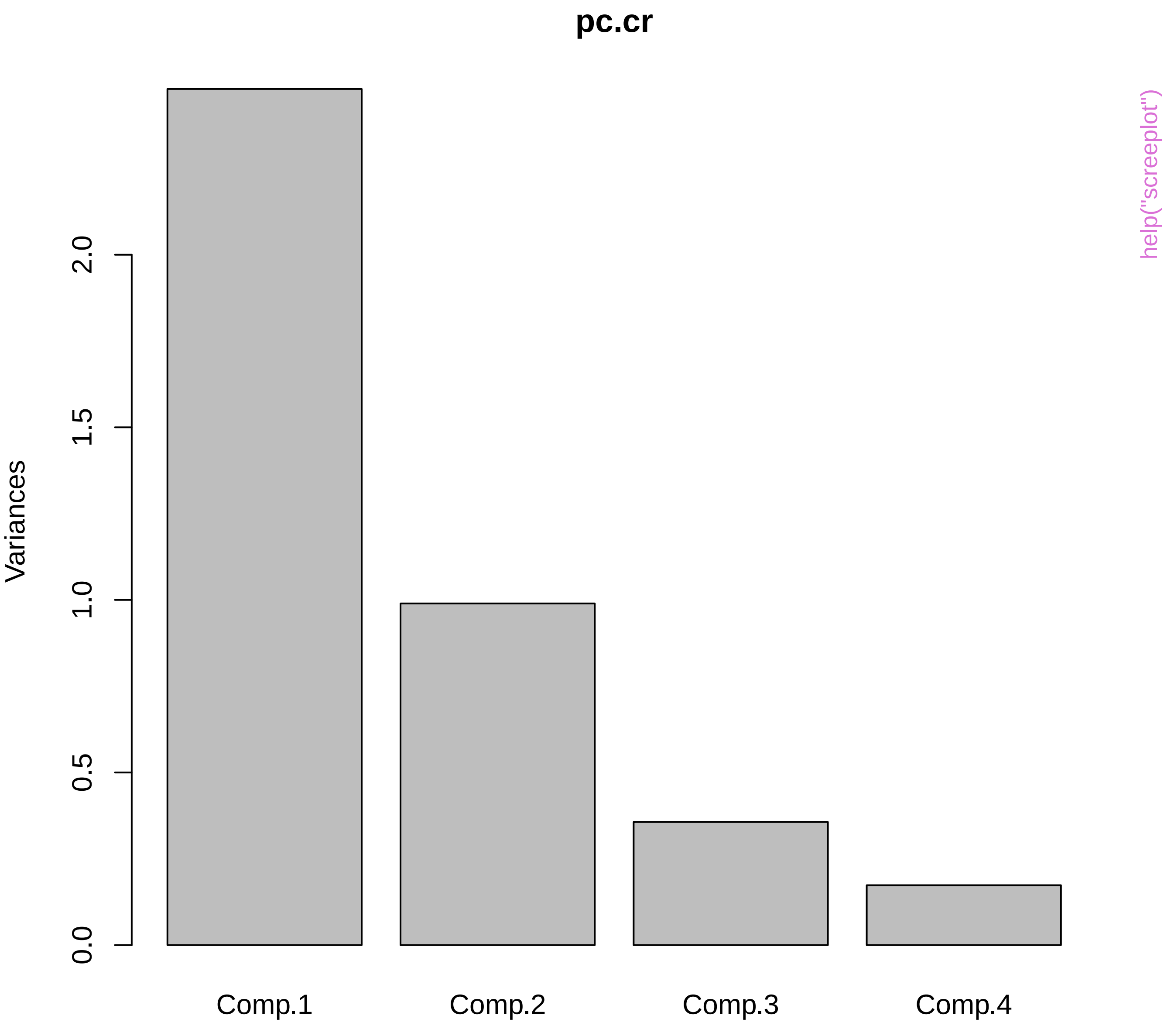

runmed(x15, k, algo = "Turlach")
x15 <- c(NA, NA, NA, NA, 9, 9, 4, 22, 6, 1, 7, 5, 2, 8, 3)
cbind(x15, rT15)
help("runmed")
data
k = 3
k = 7
k = 9
k = 11
2
4
6
8
10
12
14



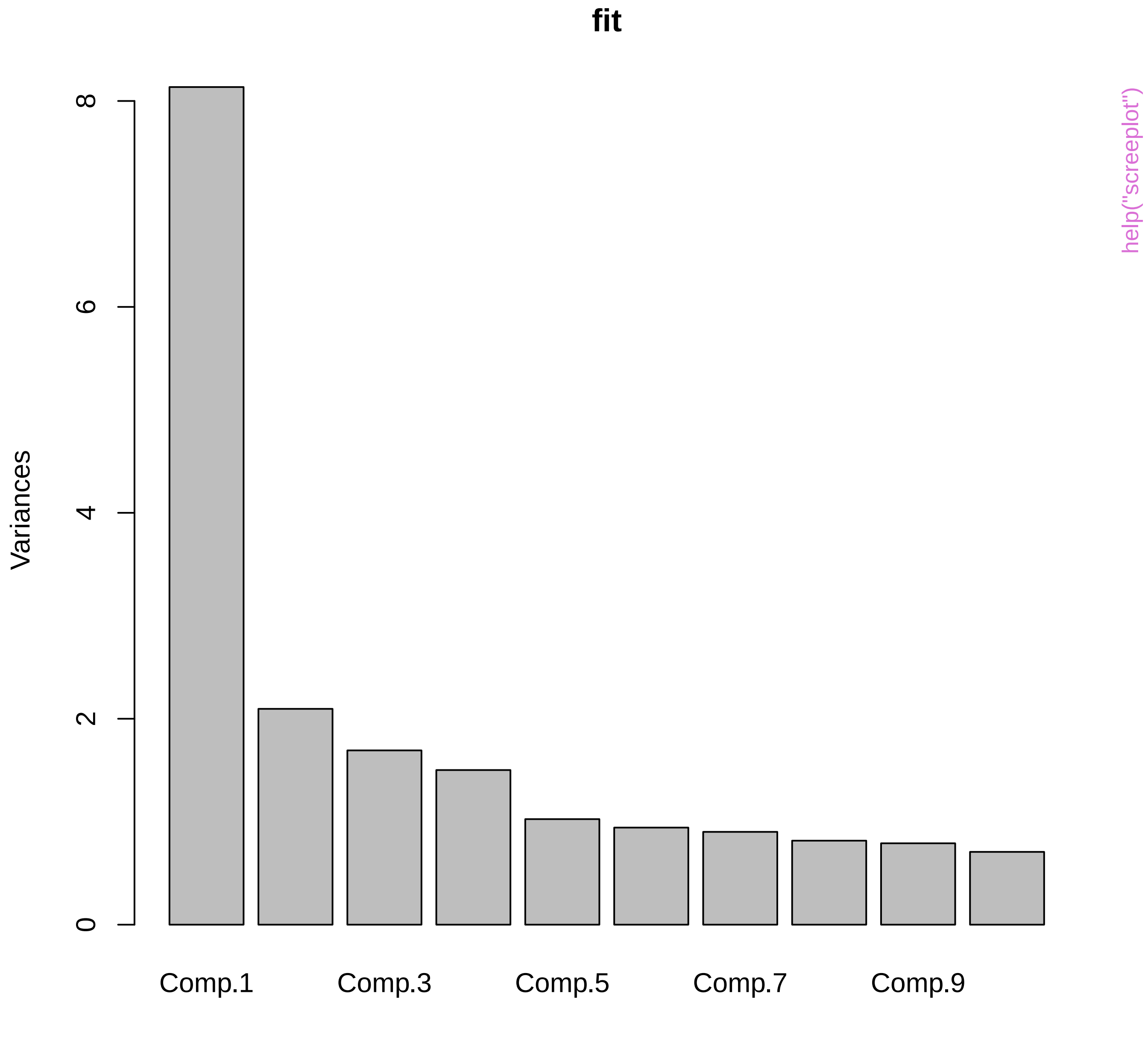
0
20
40
60
80
100
120
dist
5
10
15
20
25
speed
help("scatter.smooth")

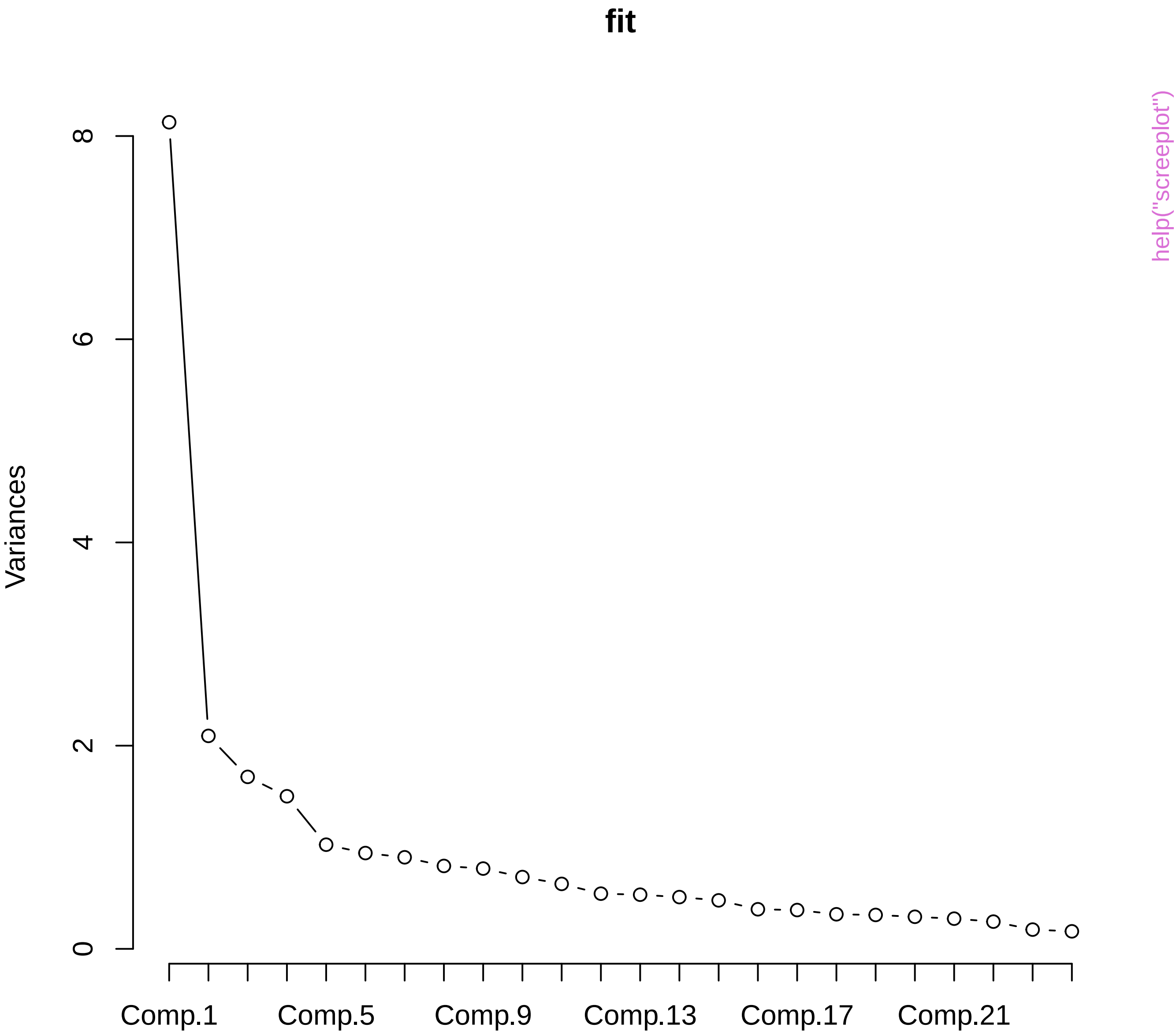


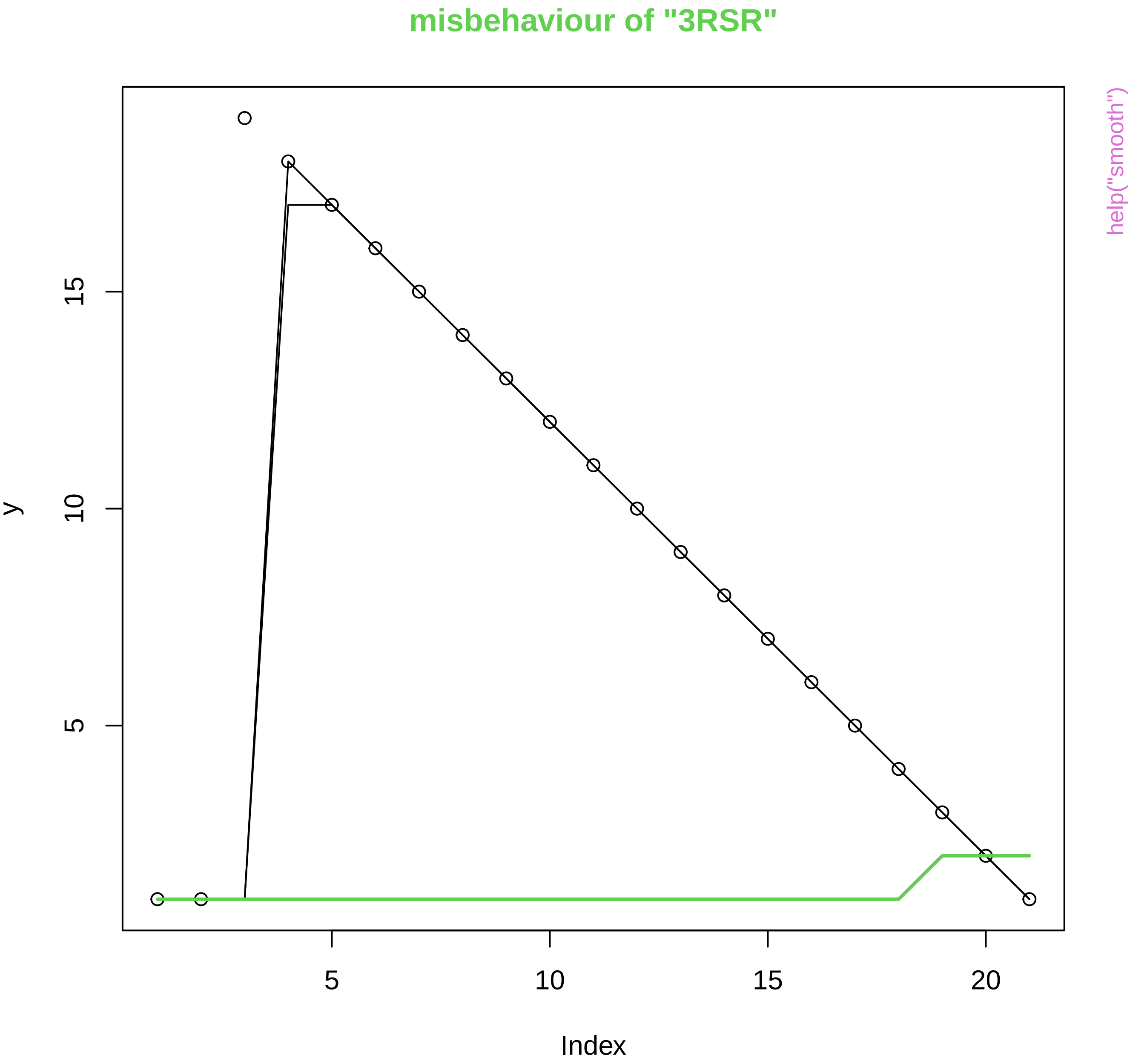
dist
speed
0
20
40
60
80
100
120
5
10
15
20
25
help("scatter.smooth")

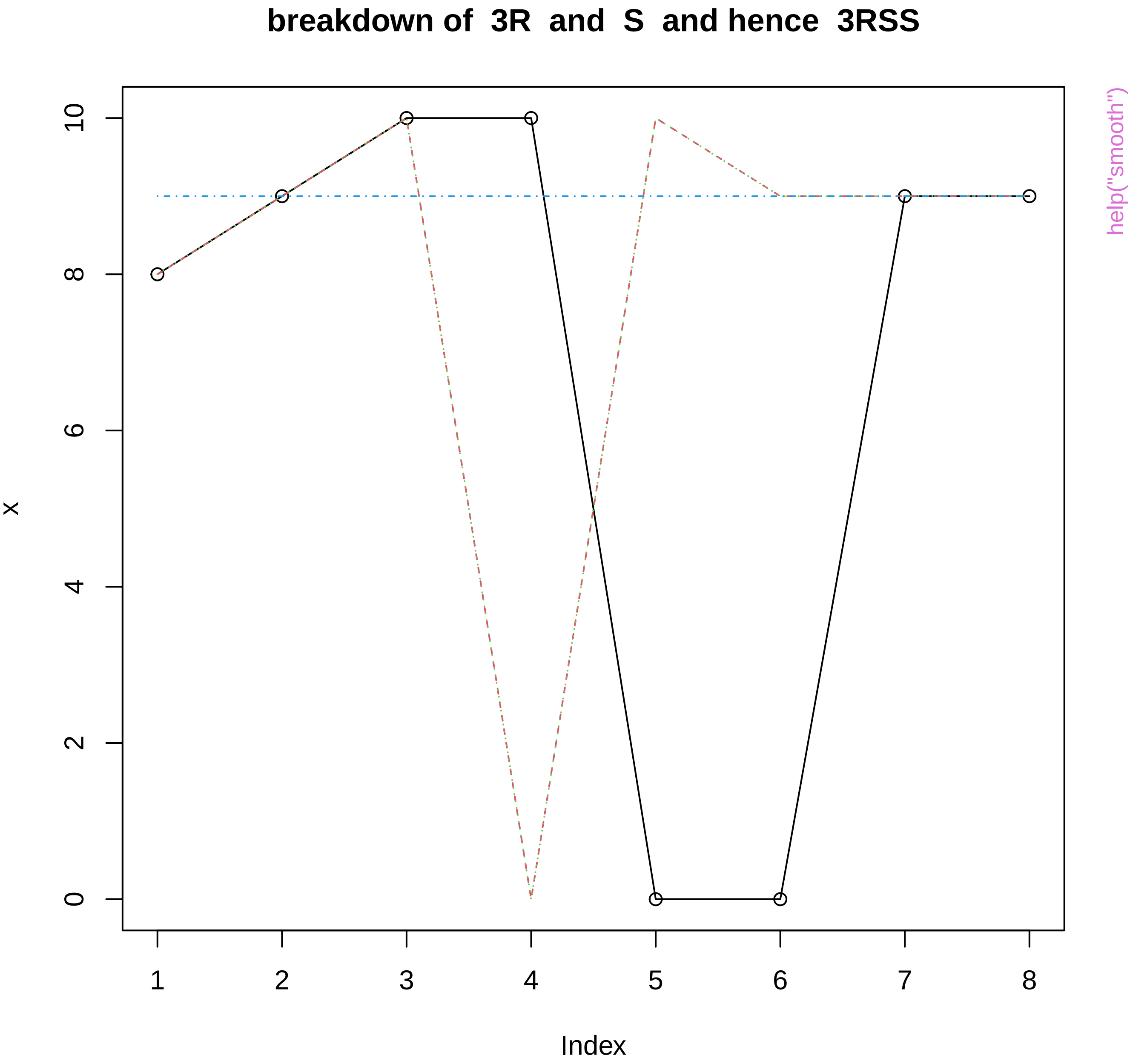


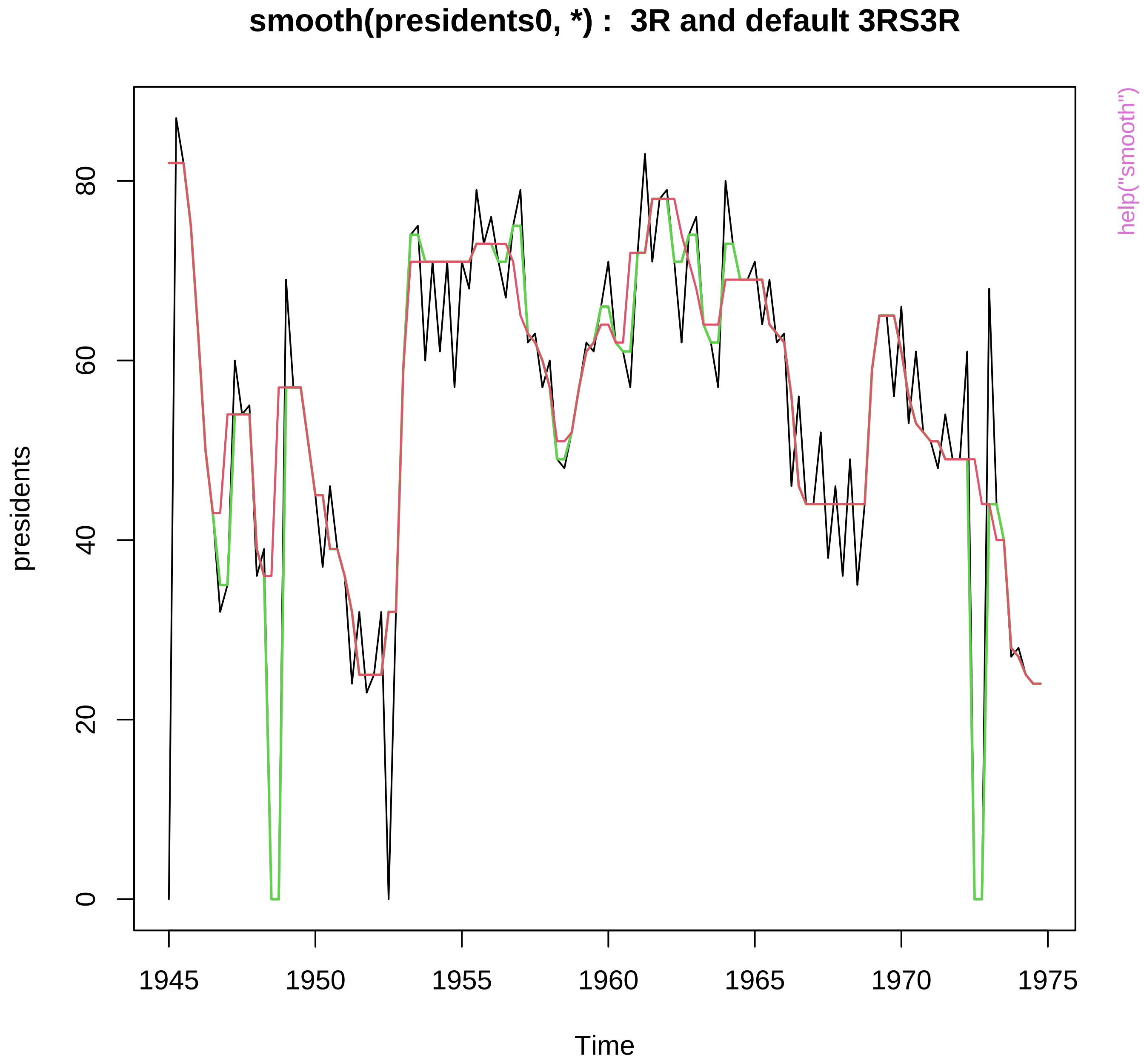
pc.cr
Variances
0.0
0.5
1.0
1.5
2.0
Comp.1
Comp.2
Comp.3
Comp.4
help("screeplot")

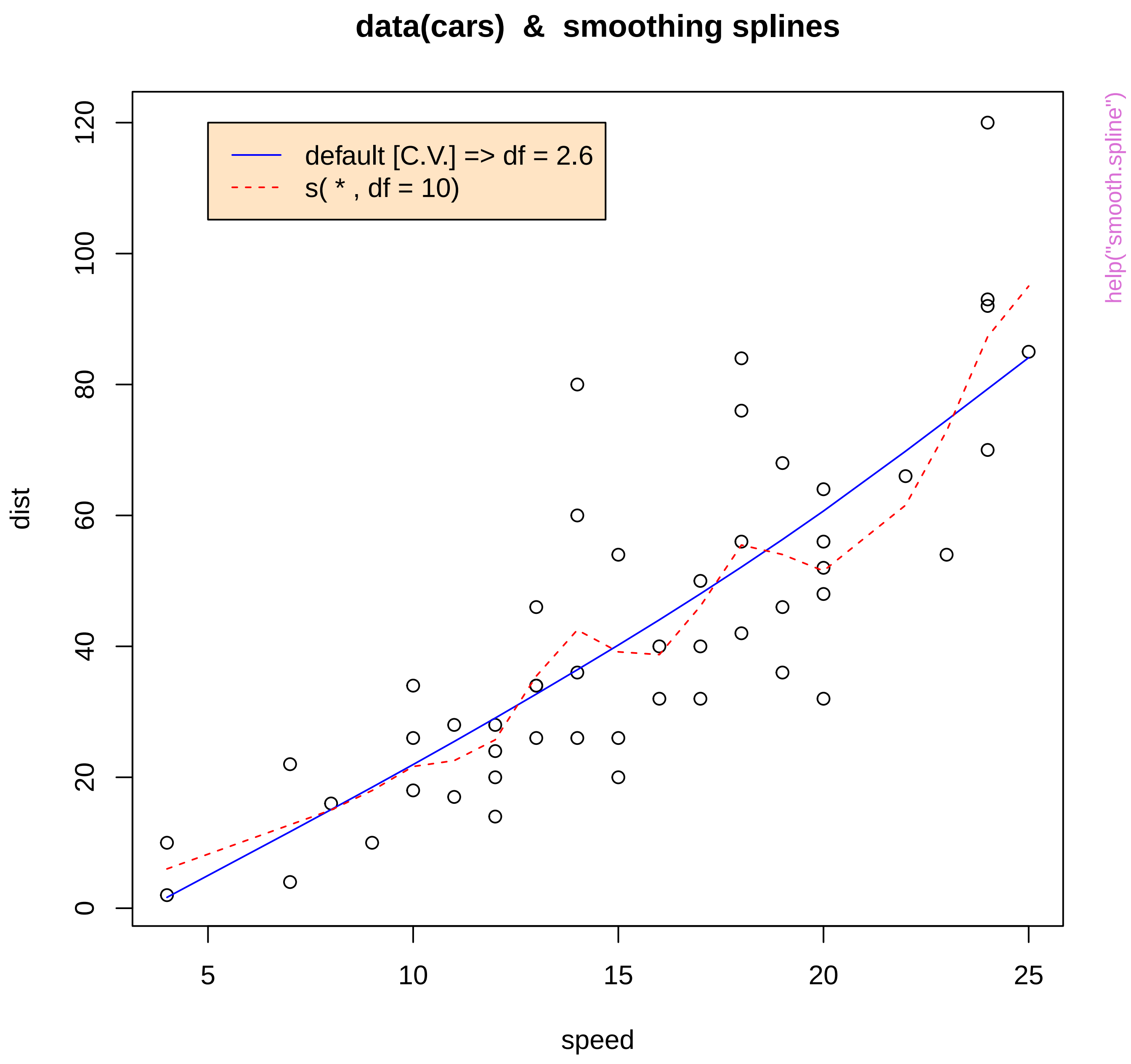

fit
Variances
0
2
4
6
8
Comp.1
Comp.3
Comp.5
Comp.7
Comp.9
help("screeplot")

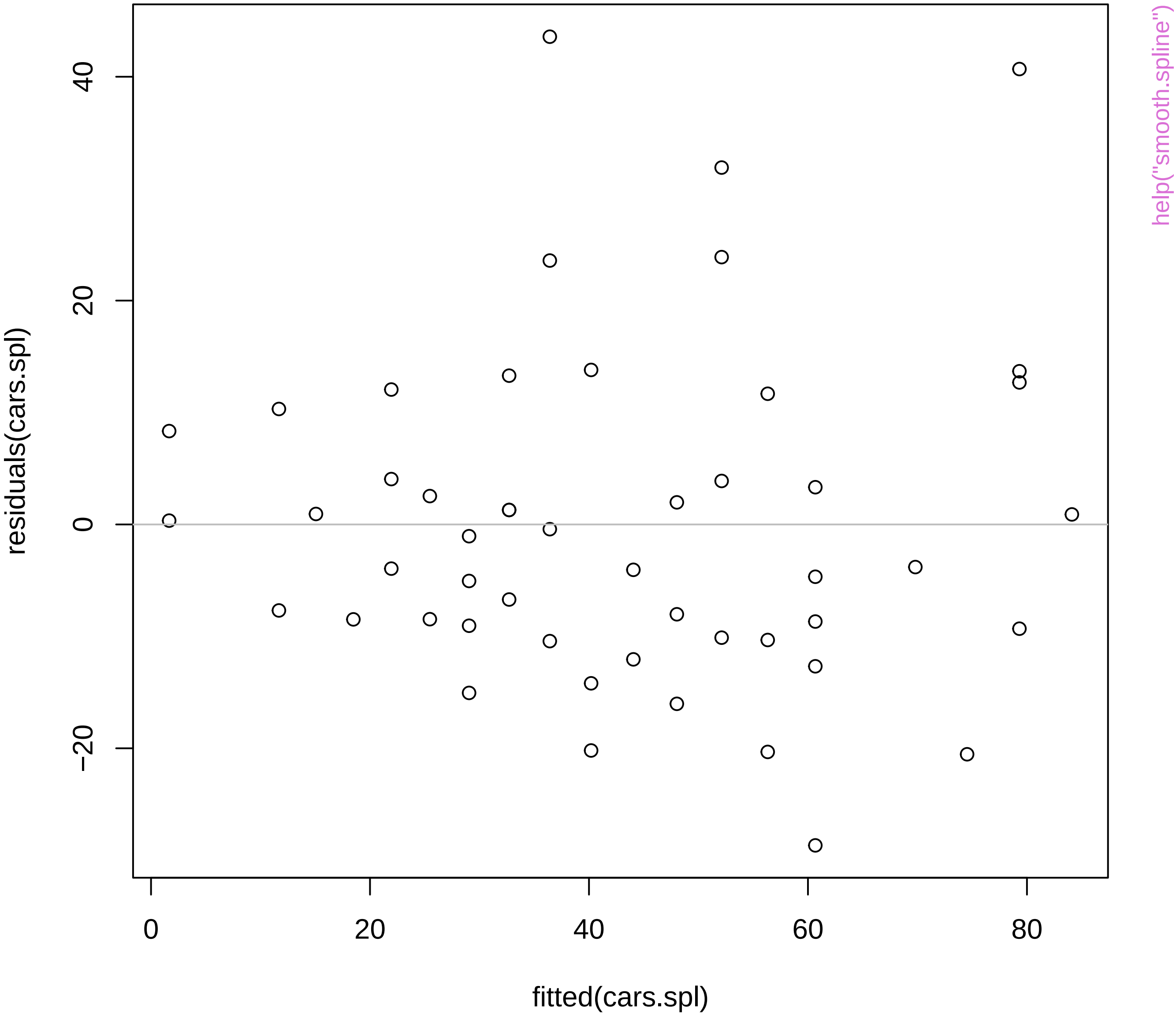

fit
Variances
0
2
4
6
8
Comp.1
Comp.5
Comp.9
Comp.13
Comp.17
Comp.21
help("screeplot")


misbehaviour of "3RSR"
help("smooth")
y
Index
5
10
15
20
5
10
15

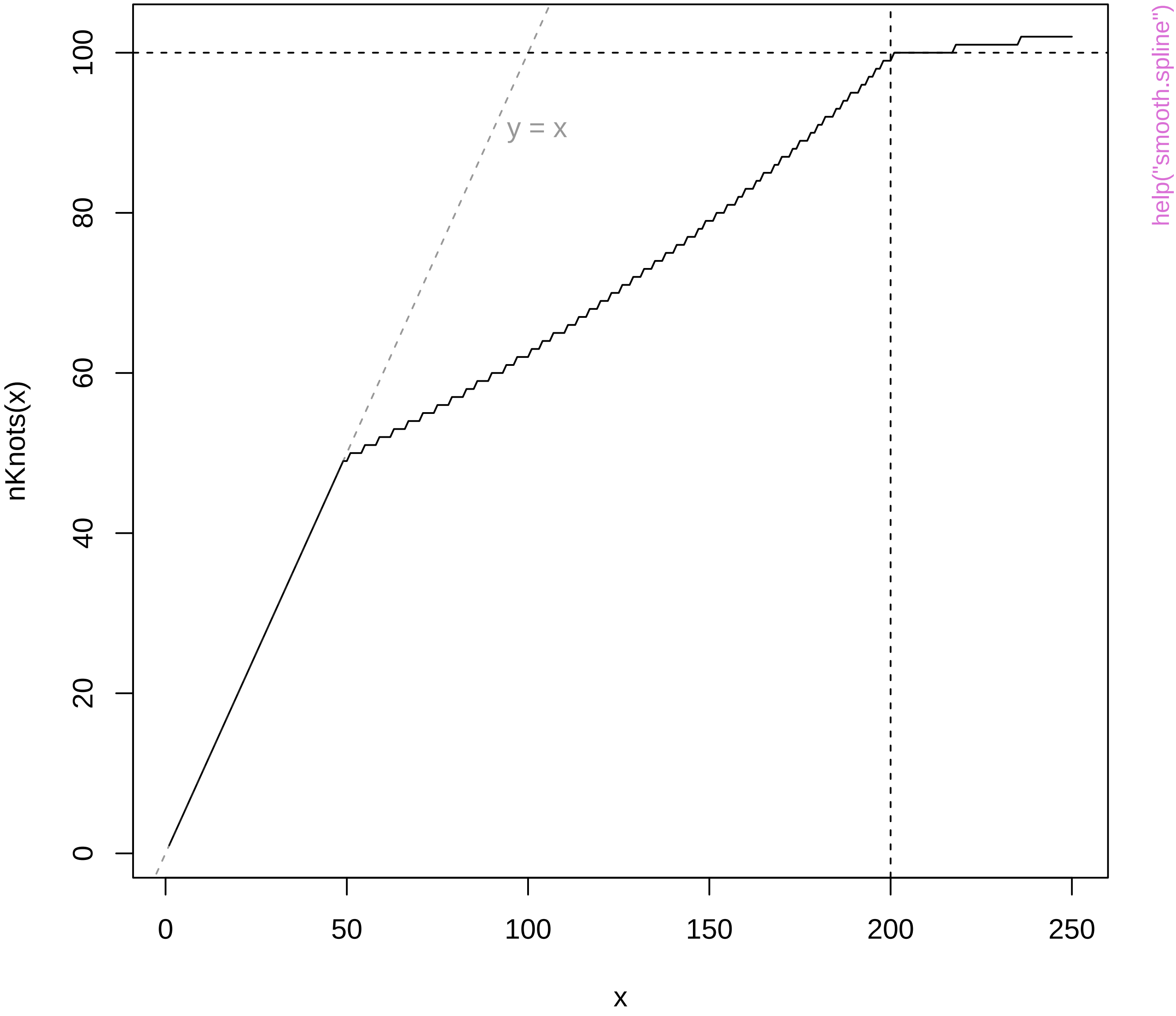

breakdown of 3R and S and hence 3RSS
x
Index
0
2
4
6
8
10
1
2
3
4
5
6
7
8
help("smooth")


smooth(presidents0, *) : 3R and default 3RS3R
presidents
0
20
40
60
80
Time
1945
1950
1955
1960
1965
1970
1975
help("smooth")

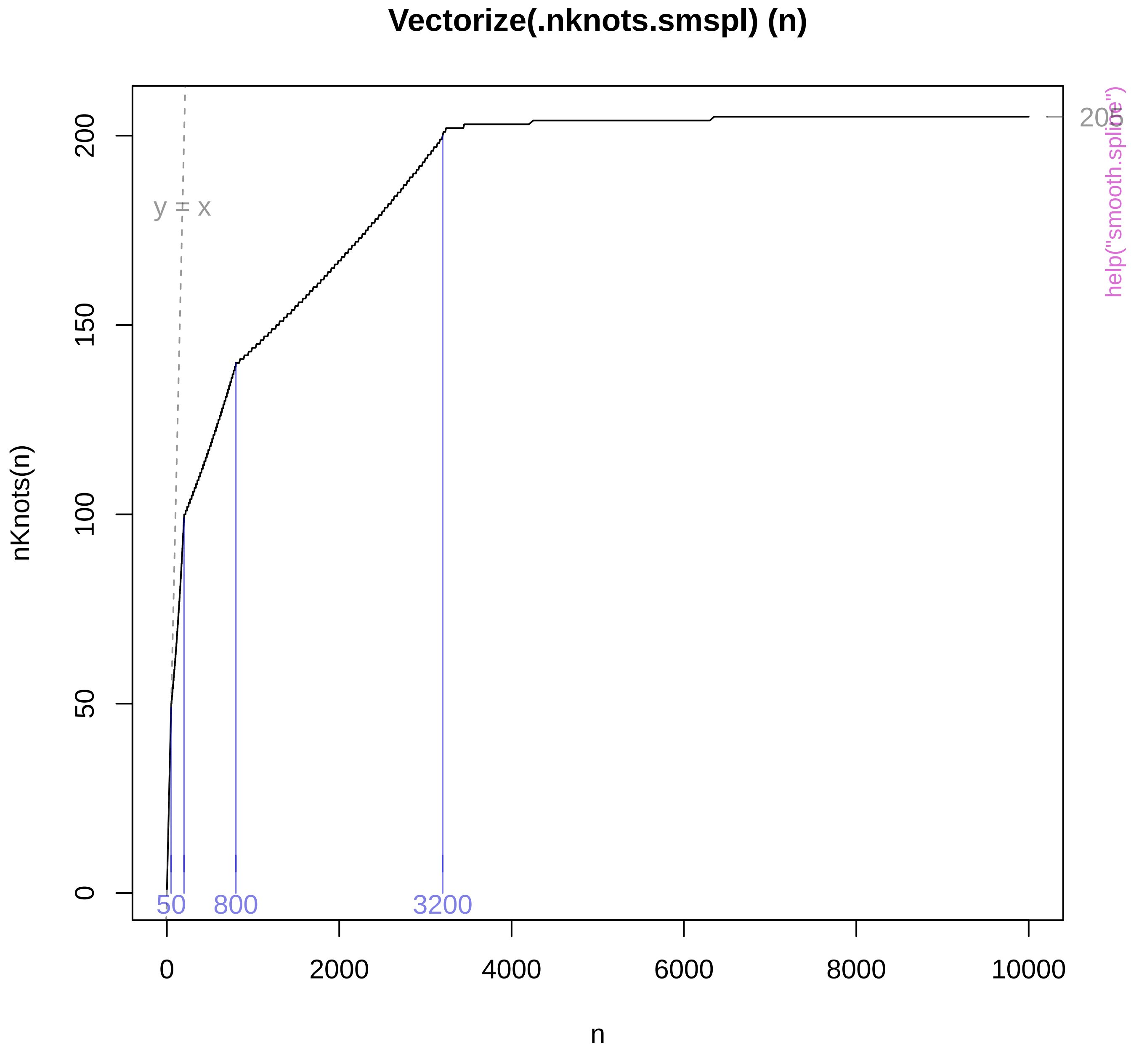

data(cars) & smoothing splines
default [C.V.] => df = 2.6
s(* , df = 10)
dist
speed
0
20
40
60
80
100
120
5
10
15
20
25
help("smooth.spline")



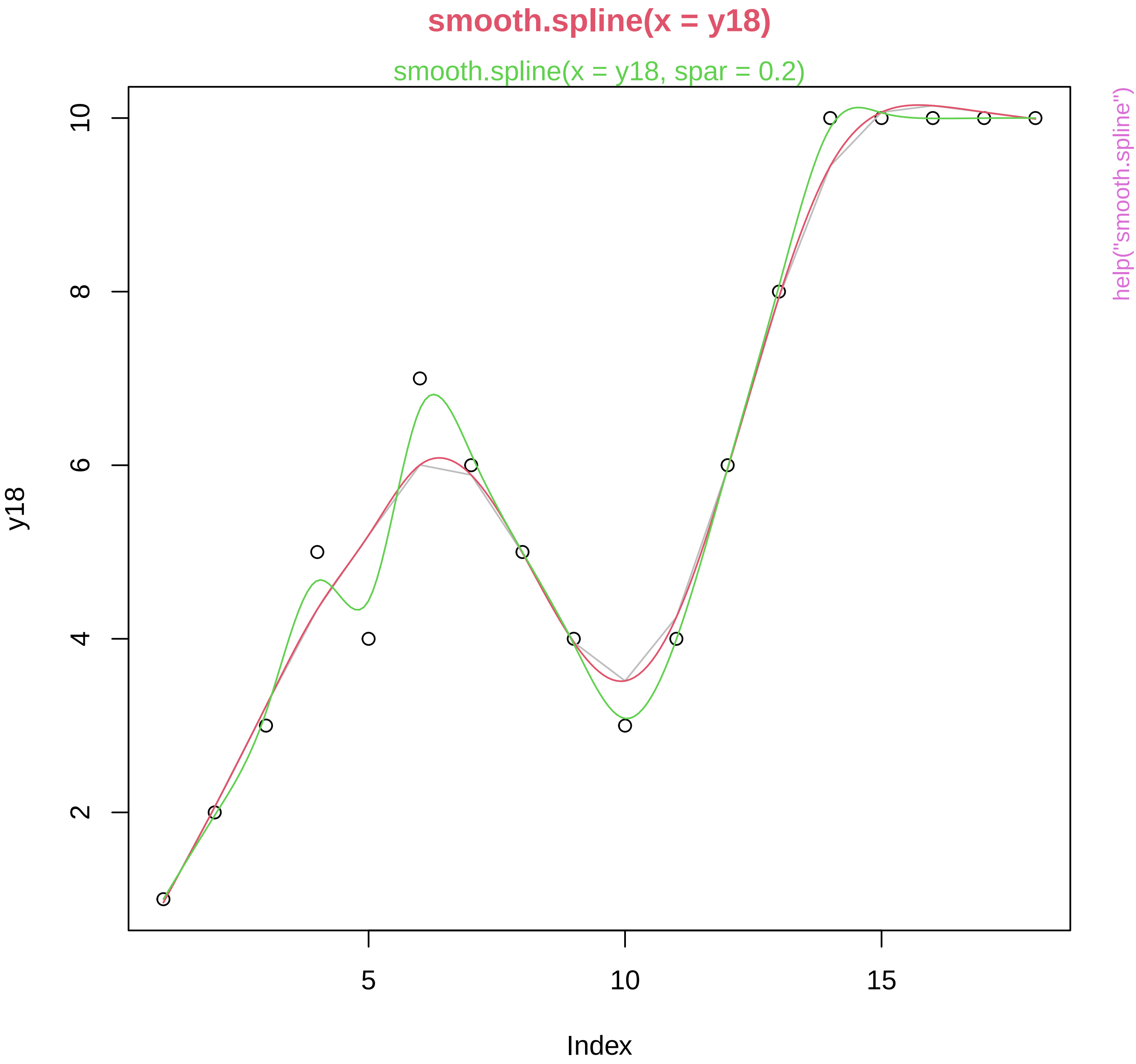
residuals(cars.spl)
fitted(cars.spl)
-20
0
20
40
0
20
40
60
80
help("smooth.spline")

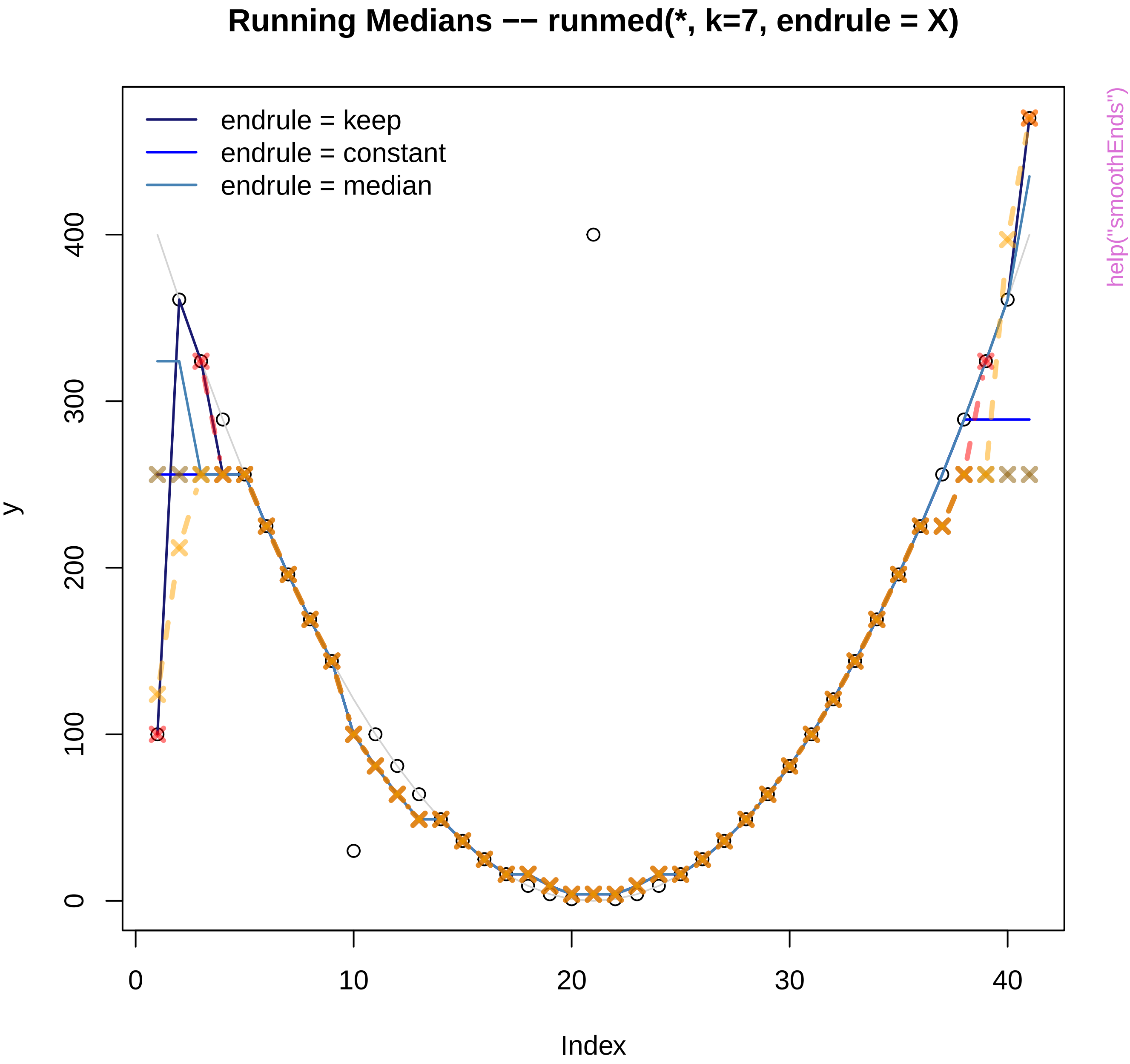


y = x
nKnots(x)
x
0
20
40
60
80
100
0
50
100
150
200
250
help("smooth.spline")

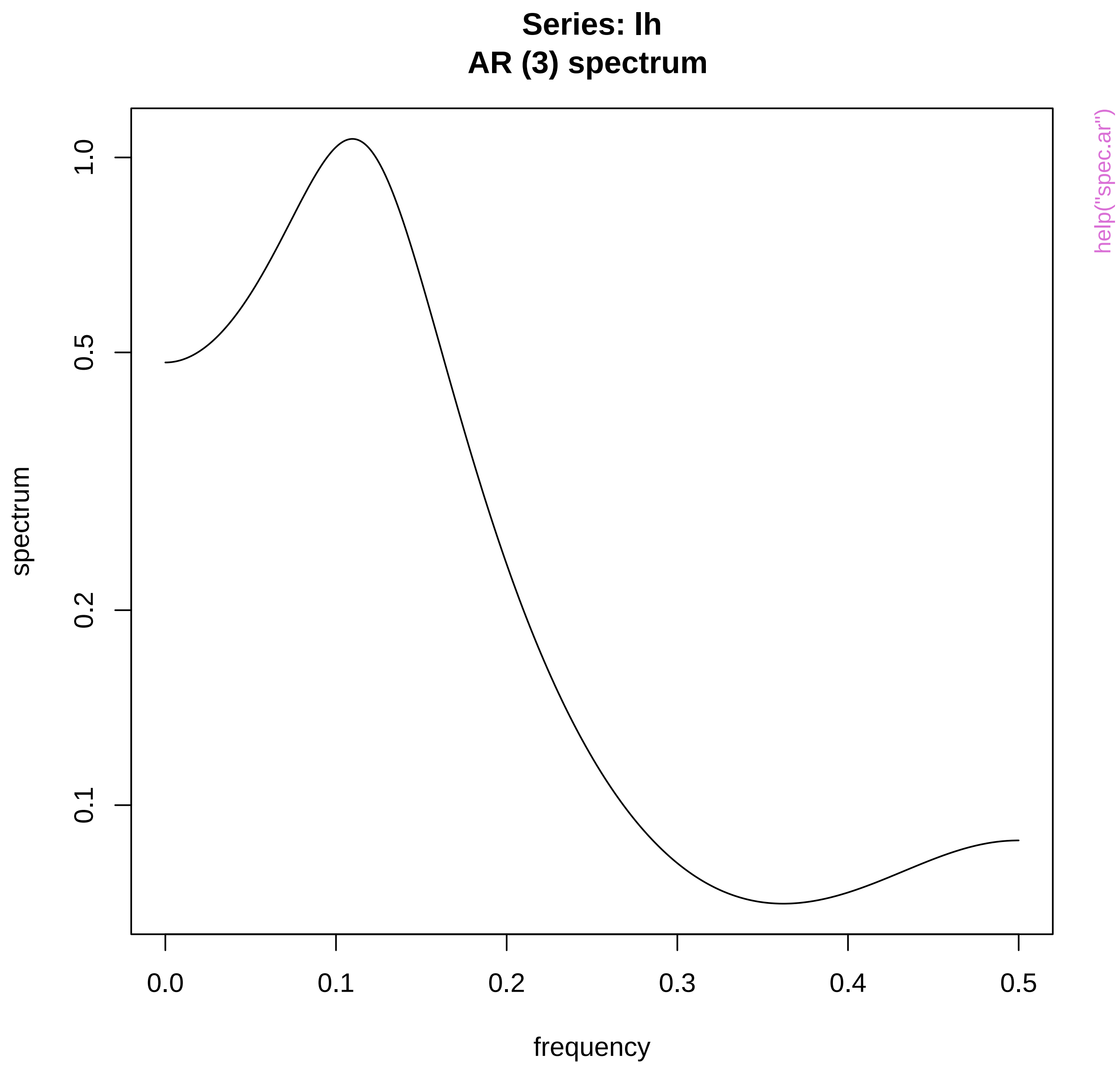


Vectorize(.nknots.smspl) (n)
nKnots(n)
n
y = x
205
help("smooth.spline")
50
800
3200
0
50
100
150
200
0
2000
4000
6000
8000
10000

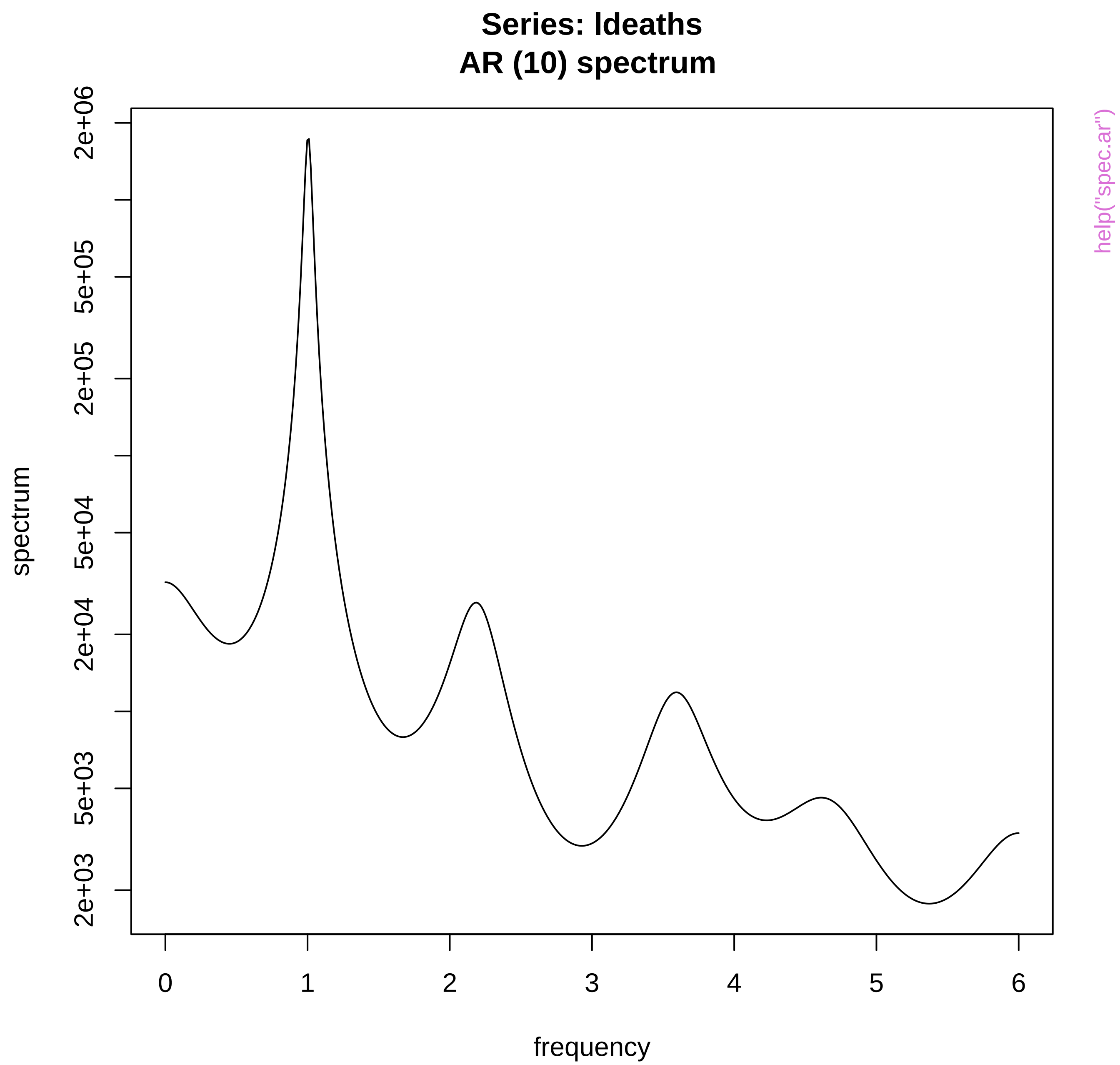

smooth.spline(x = y18)
smooth.spline(x = y18, spar = 0.2)
help("smooth.spline")
y18
Index
2
4
6
8
10
5
10
15

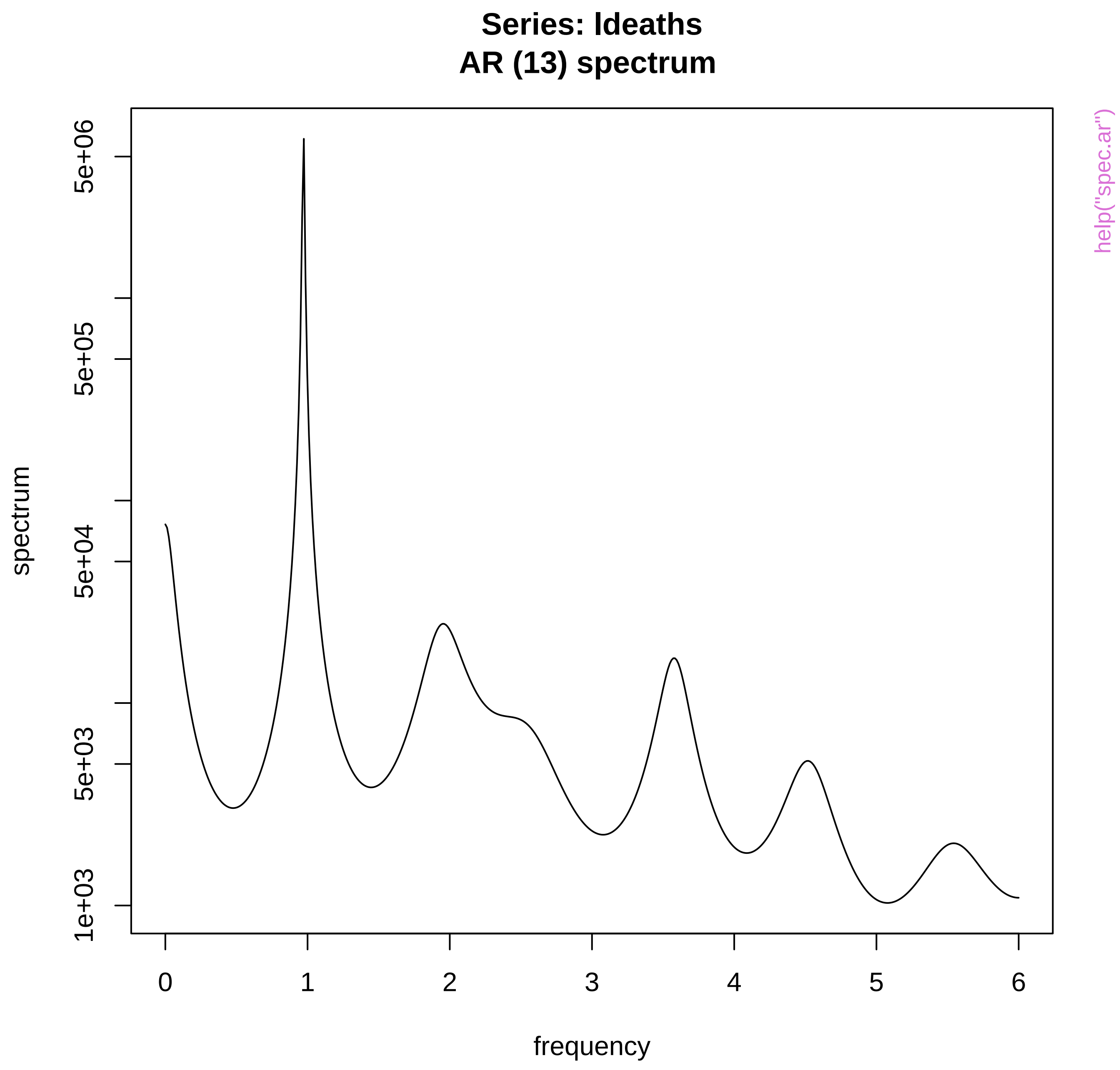


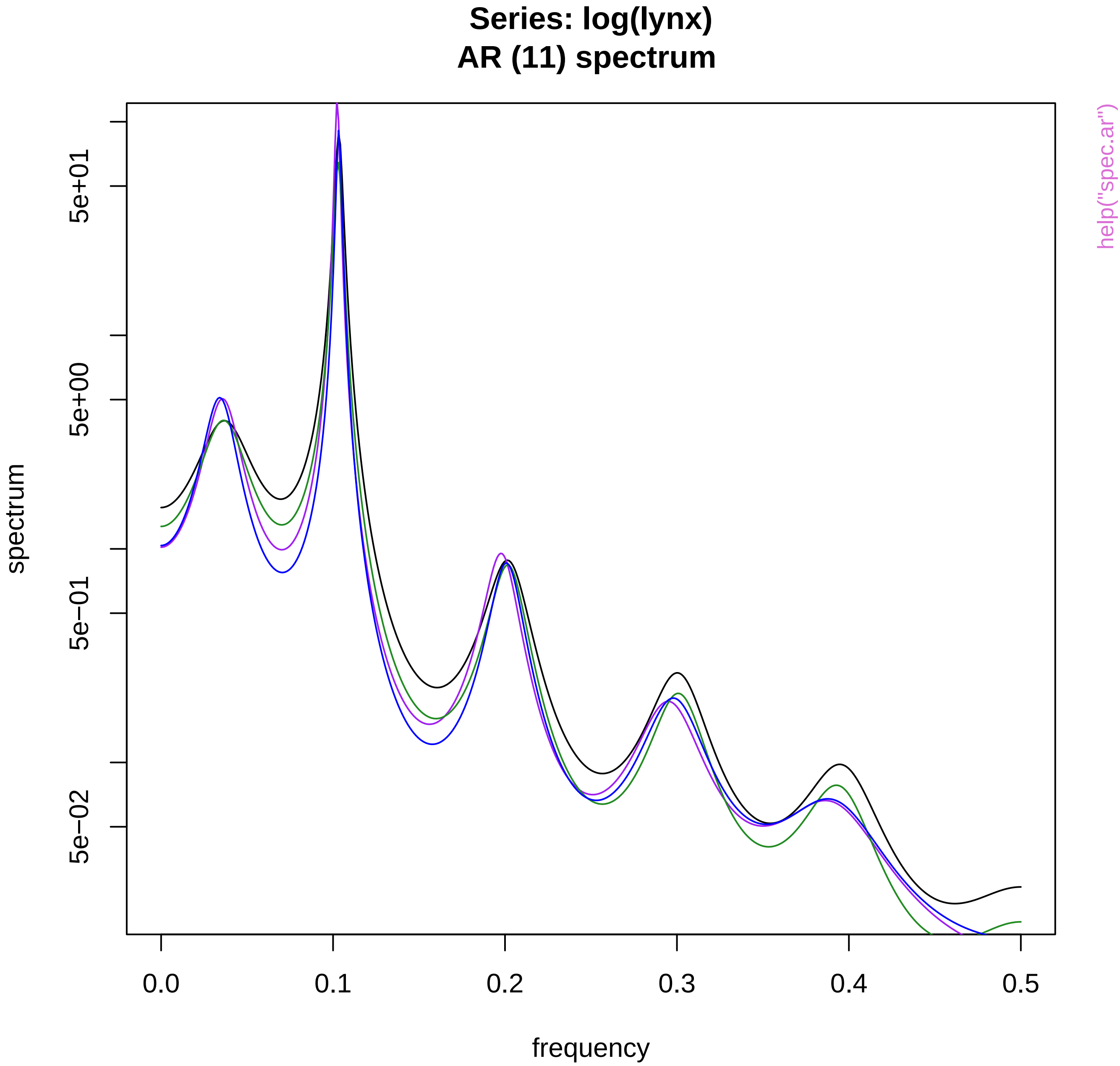
Running Medians -- runmed(*, k=7, endrule = X)
endrule = keep
endrule = constant
endrule = median
help("smoothEnds")
y
Index
0
100
200
300
400
0
10
20
30
40



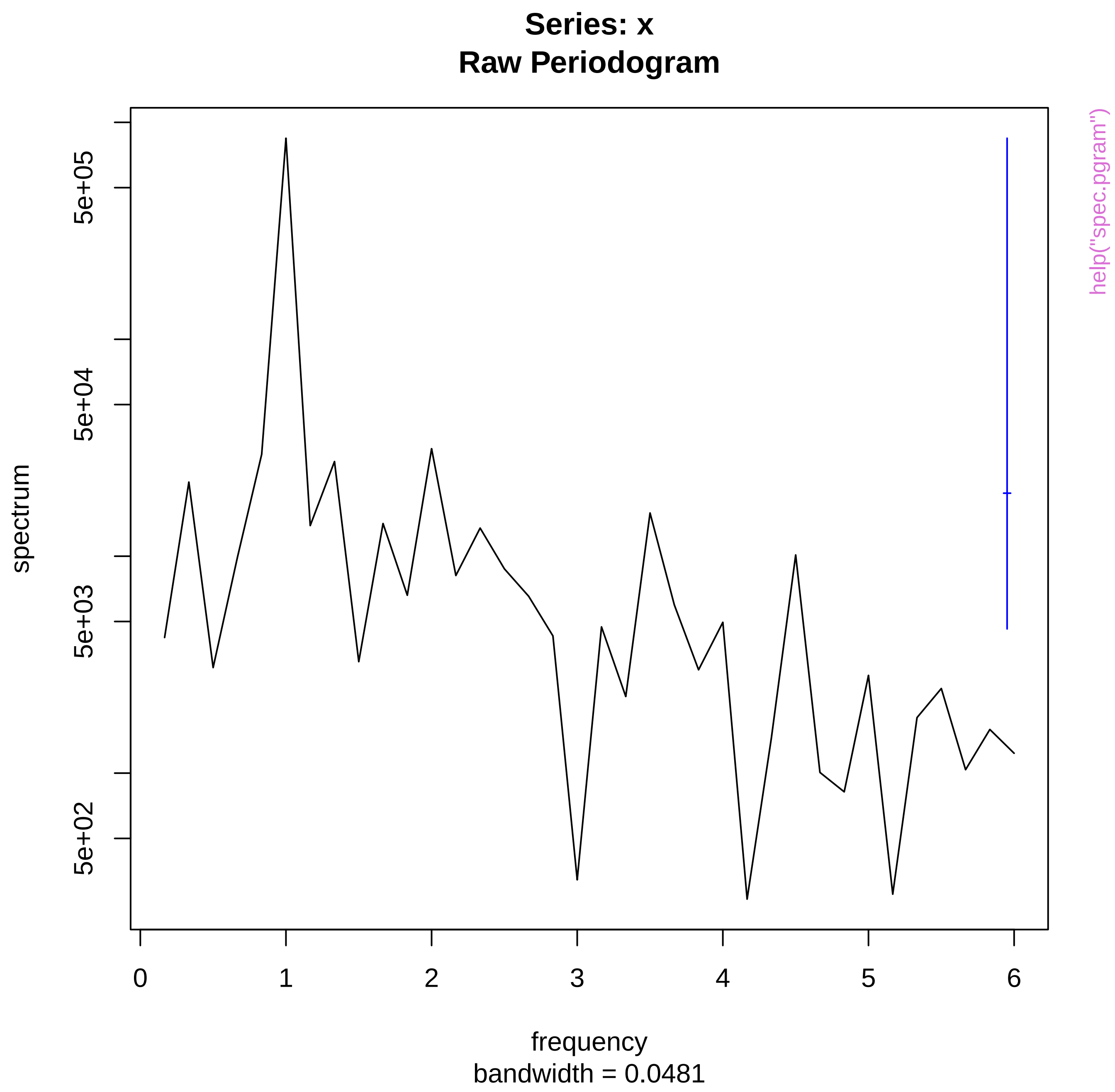
Series: lh
AR (3) spectrum
spectrum
1.0
0.5
0.2
0.1
0.0
0.1
0.2
0.3
0.4
0.5
frequency
help("spec.ar")


Series: ldeaths
AR (10) spectrum
spectrum
frequency
2e+06
5e+05
2e+05
5e+04
2e+04
5e+03
2e+03
0
1
2
3
4
5
6
help("spec.ar")

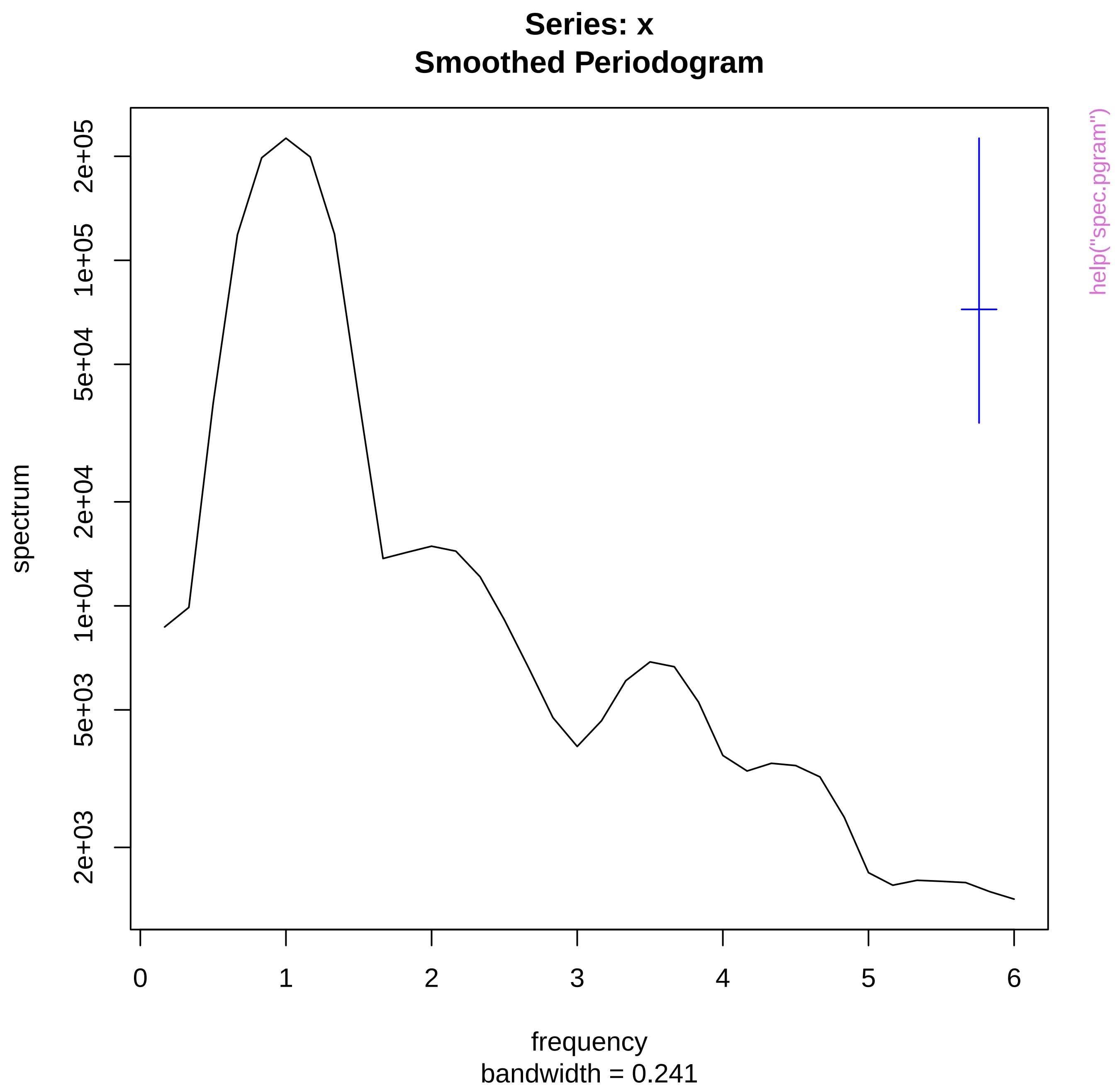

Series: ldeaths
AR (13) spectrum
spectrum
frequency
5e+06
5e+05
5e+04
5e+03
1e+03
0
1
2
3
4
5
6
help("spec.ar")



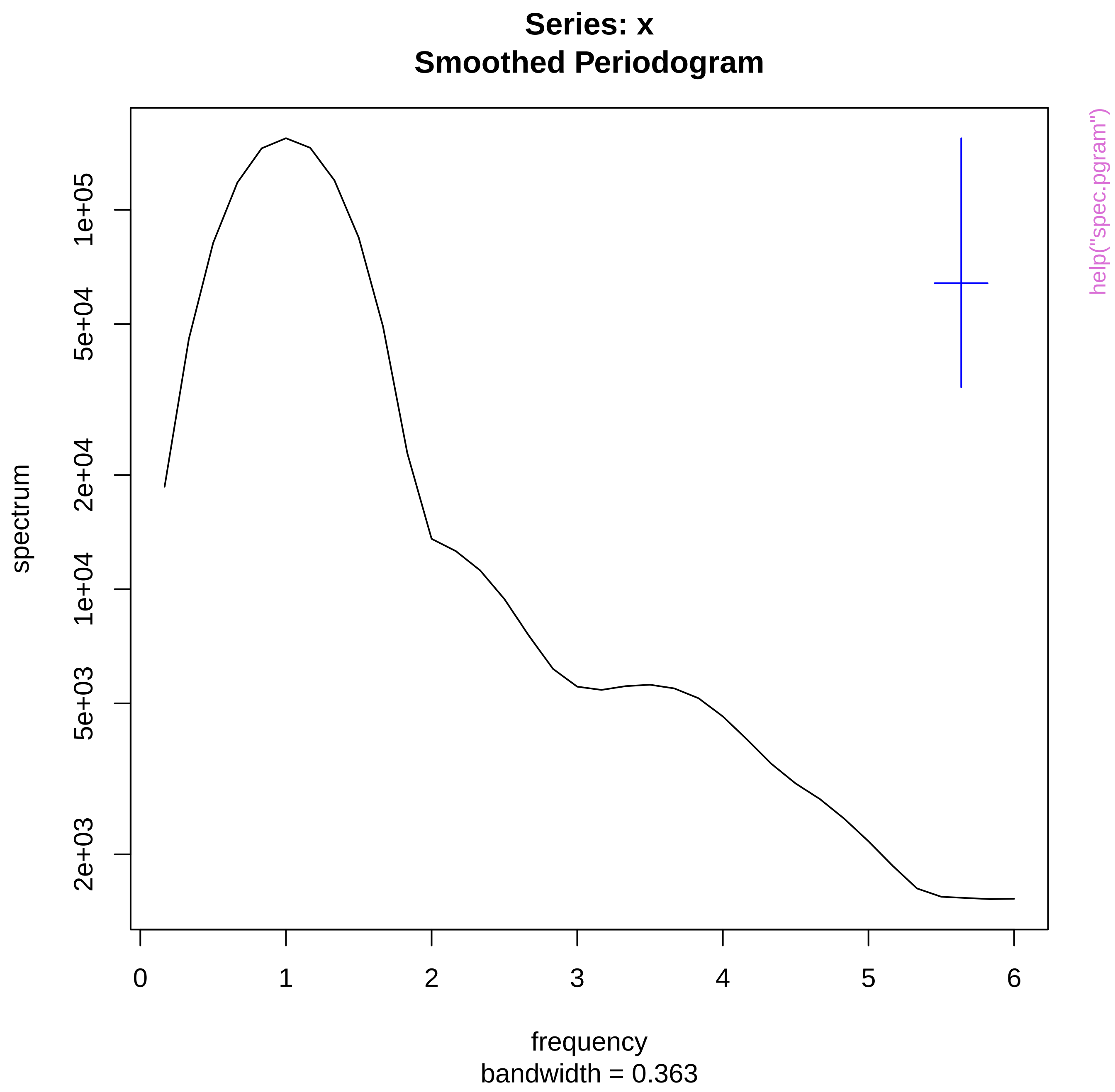
Series: log(lynx)
AR (11) spectrum
5e+01
5e+00
5e−01
5e−02
spectrum
0.0
0.1
0.2
0.3
0.4
0.5
frequency
help("spec.ar")

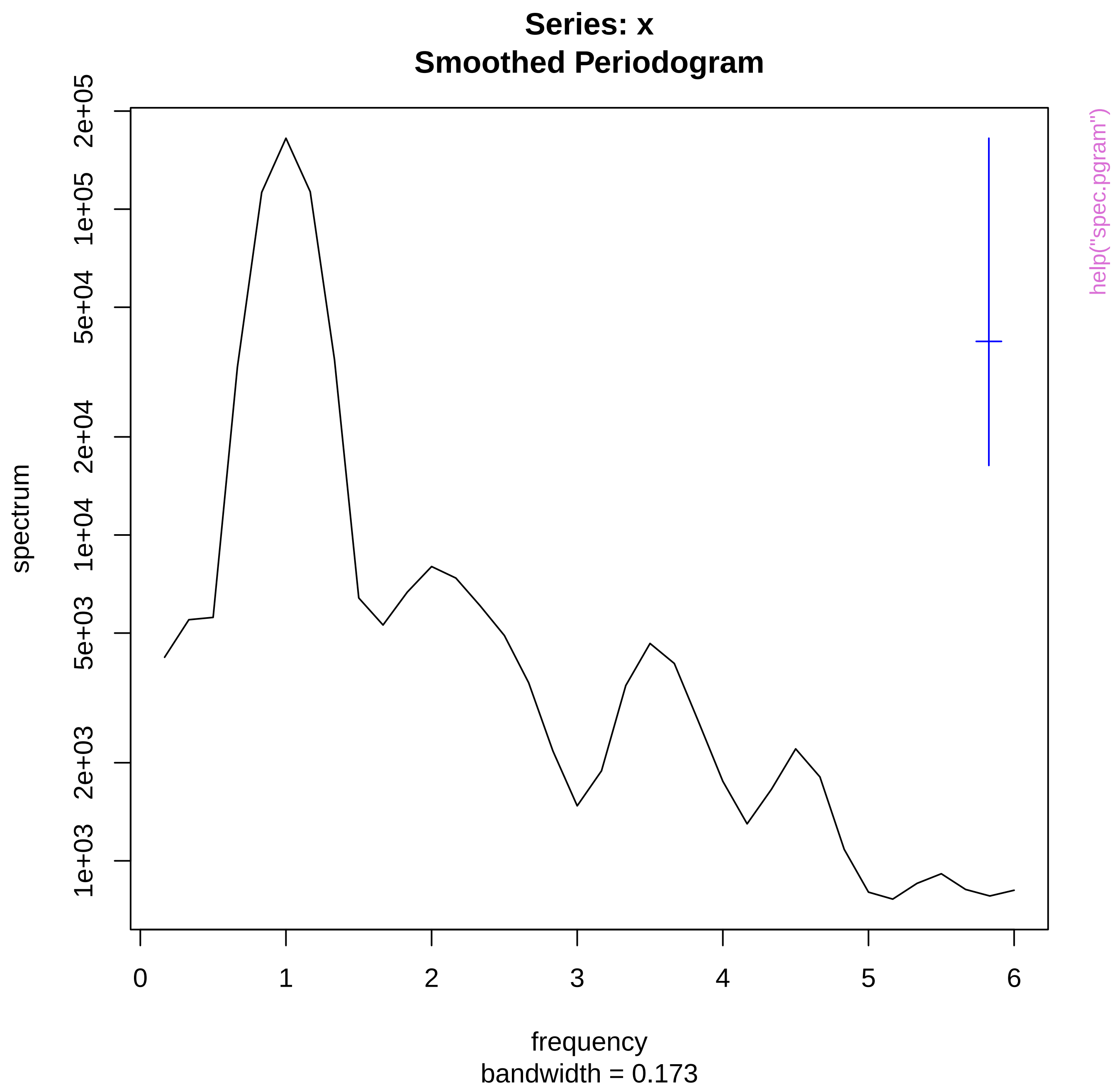


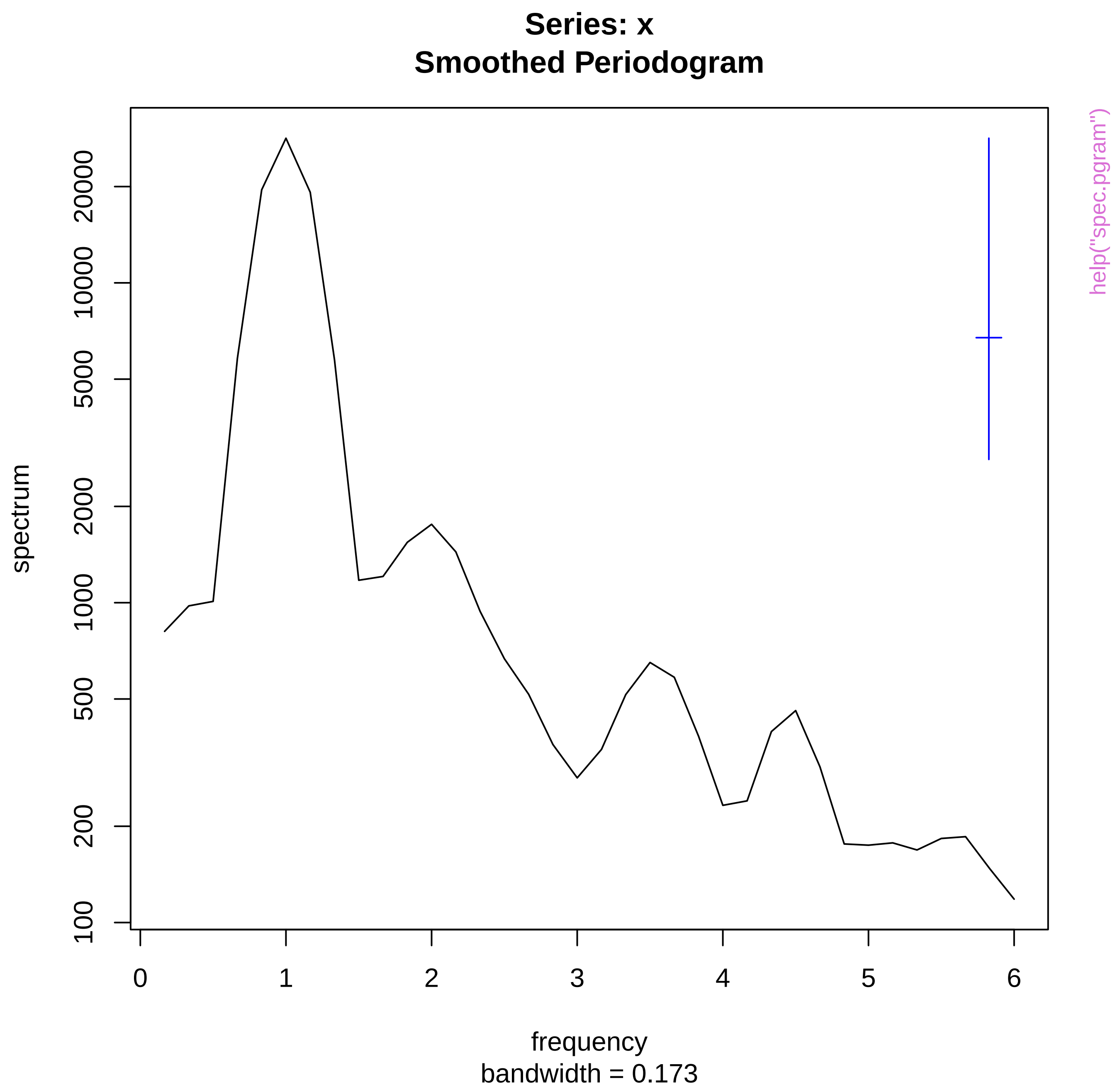
Series: x
Raw Periodogram
5e+02
5e+03
5e+04
5e+05
spectrum
0
1
2
3
4
5
6
frequency
bandwidth = 0.0481
help("spec.pgram")



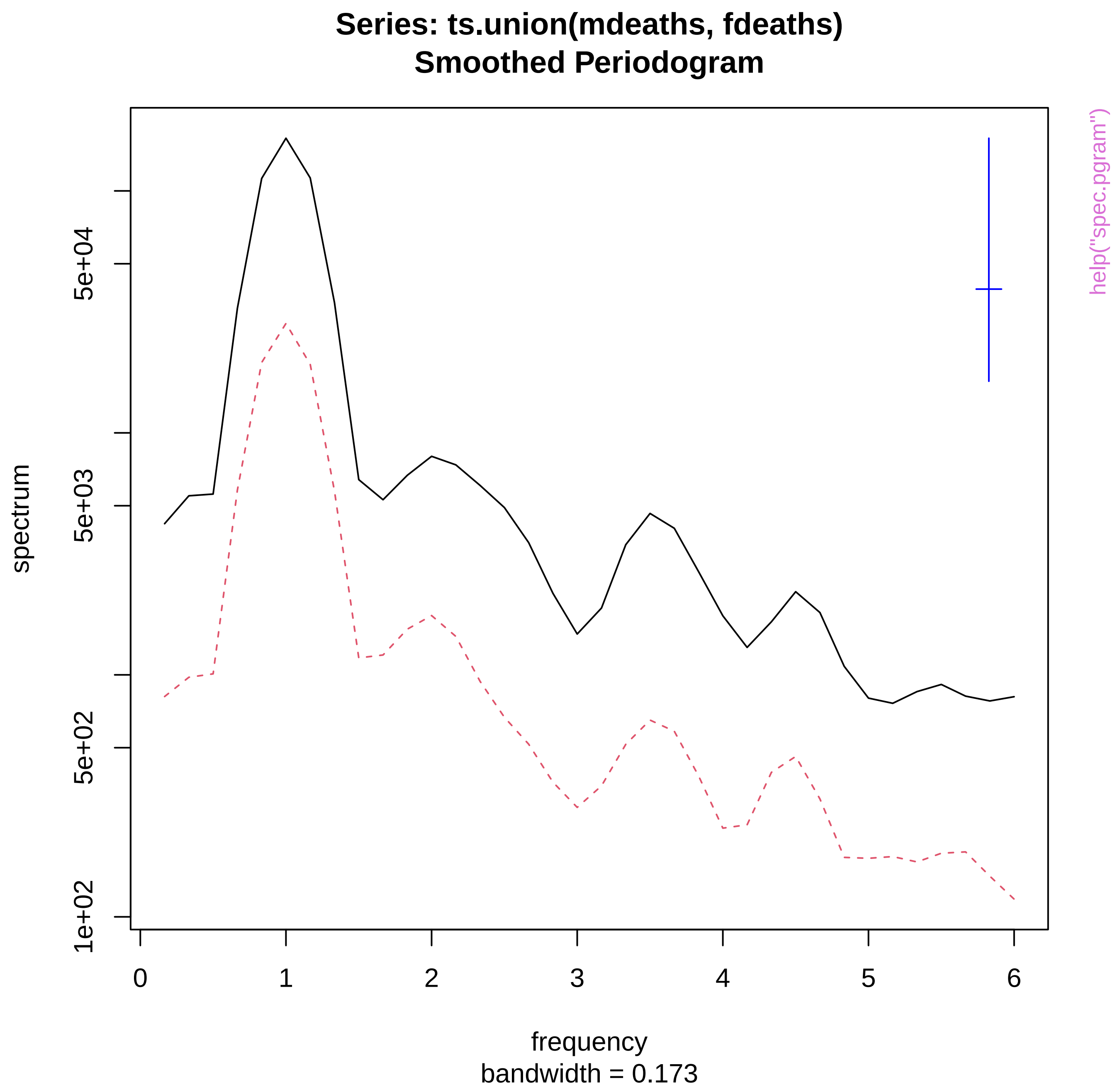
Series: x
Smoothed Periodogram
spectrum
2e+05
1e+05
5e+04
2e+04
1e+04
5e+03
2e+03
0
1
2
3
4
5
6
frequency
bandwidth = 0.241
help("spec.pgram")

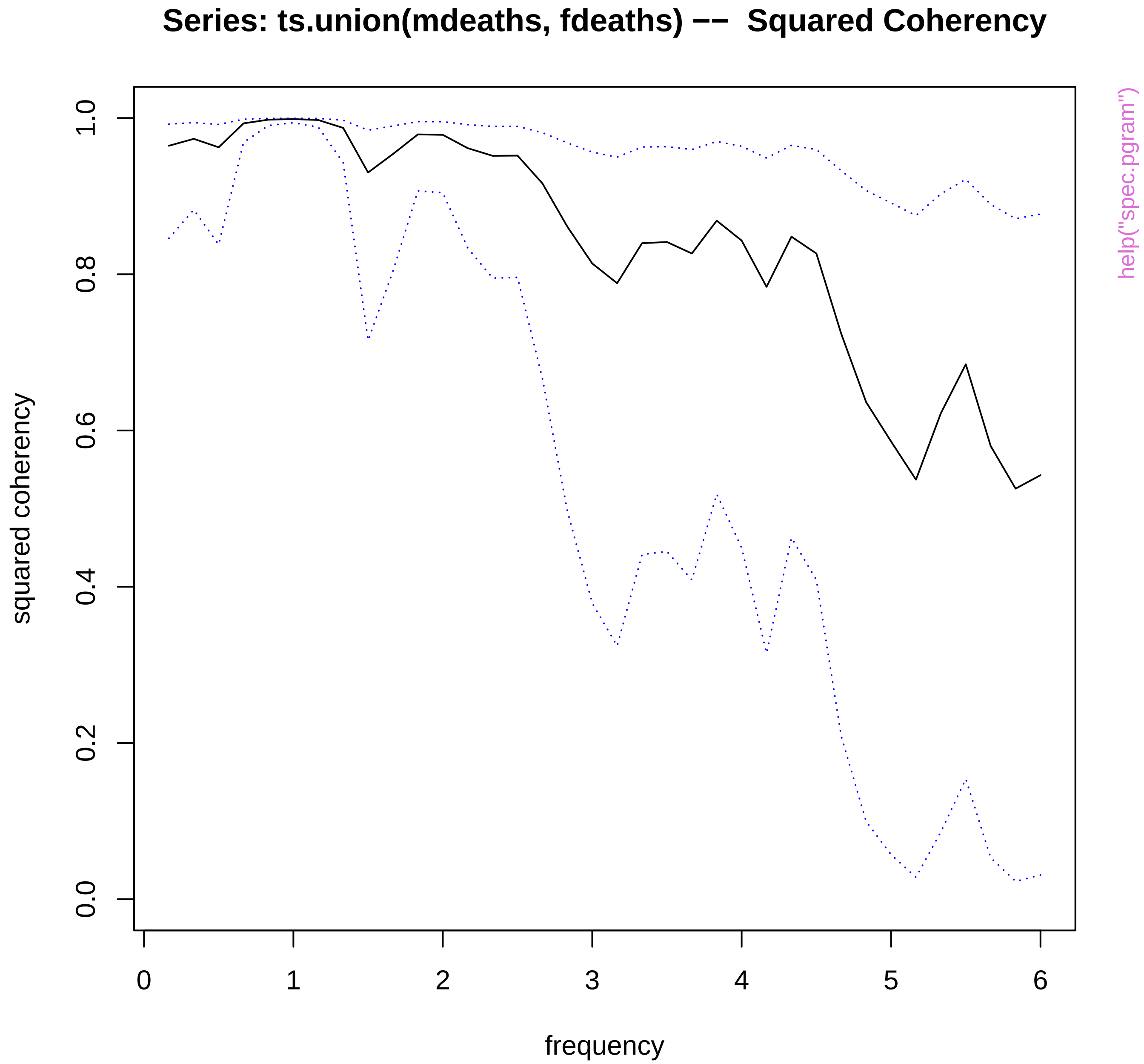

Series: x
Smoothed Periodogram
spectrum
2e+03
5e+03
1e+04
2e+04
5e+04
1e+05
0
1
2
3
4
5
6
frequency
bandwidth = 0.363
help("spec.pgram")

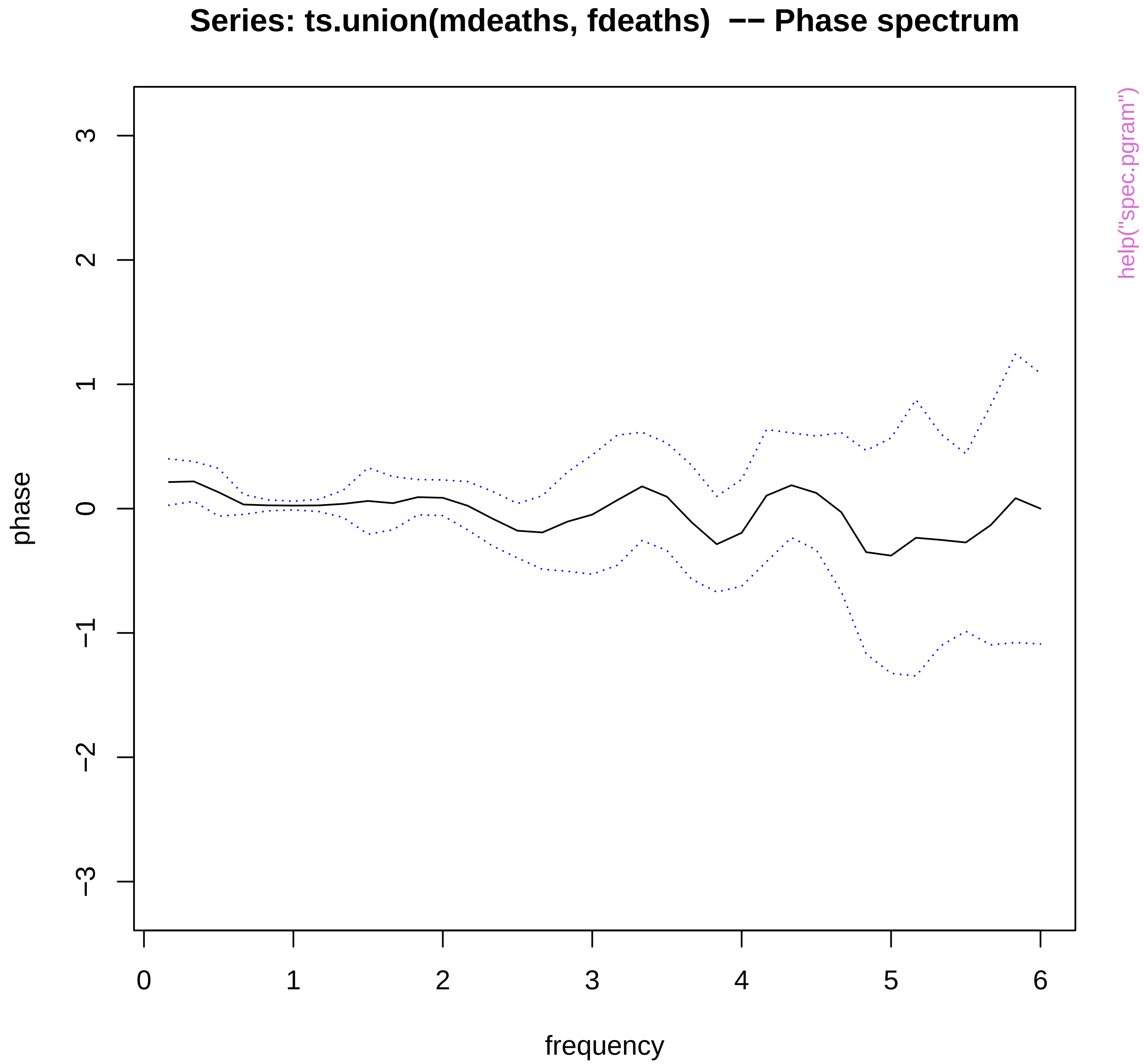

Series: x
Smoothed Periodogram
spectrum
2e+05
1e+05
5e+04
2e+04
1e+04
5e+03
2e+03
1e+03
0
1
2
3
4
5
6
frequency
bandwidth = 0.173
help("spec.pgram")

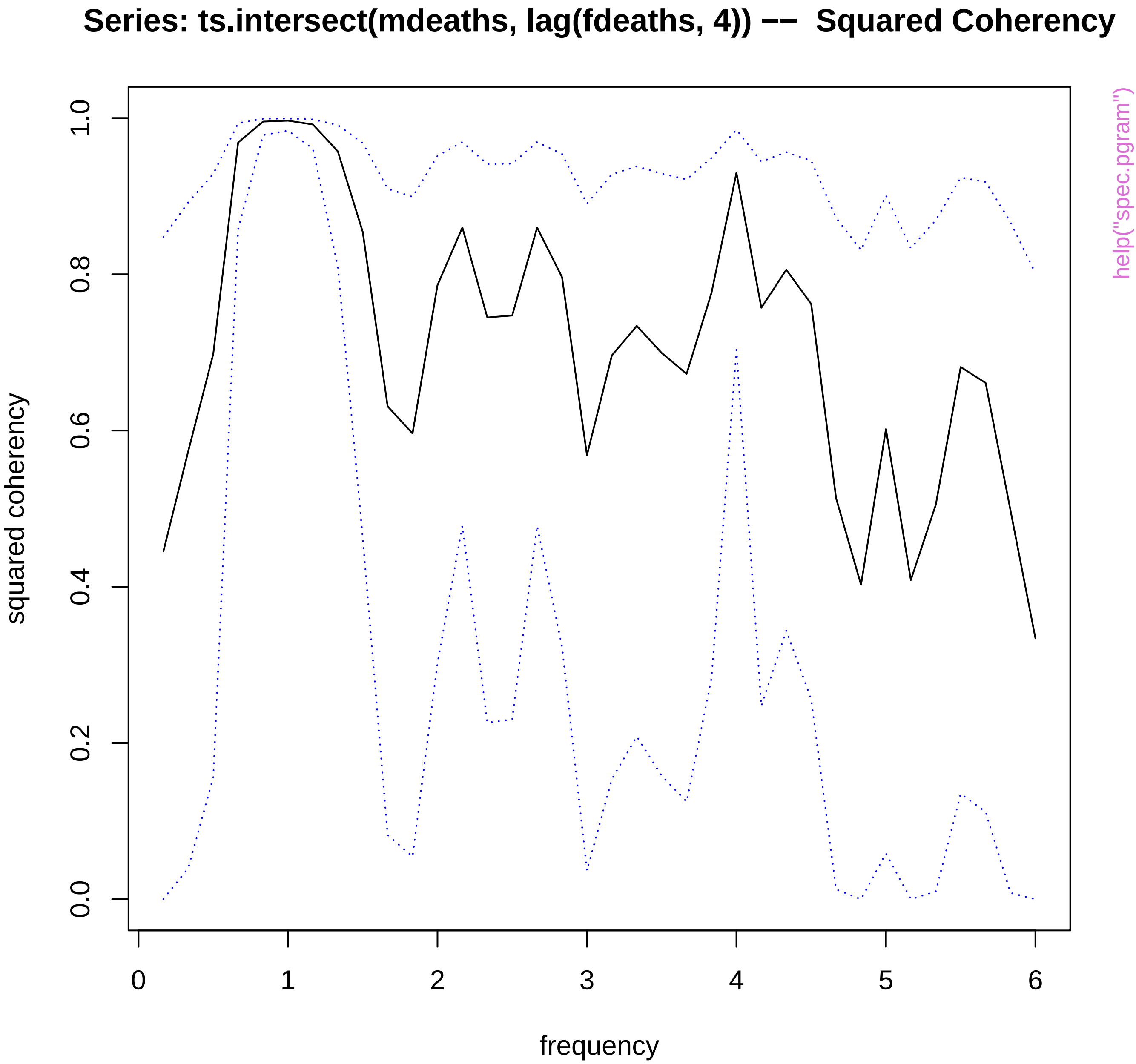


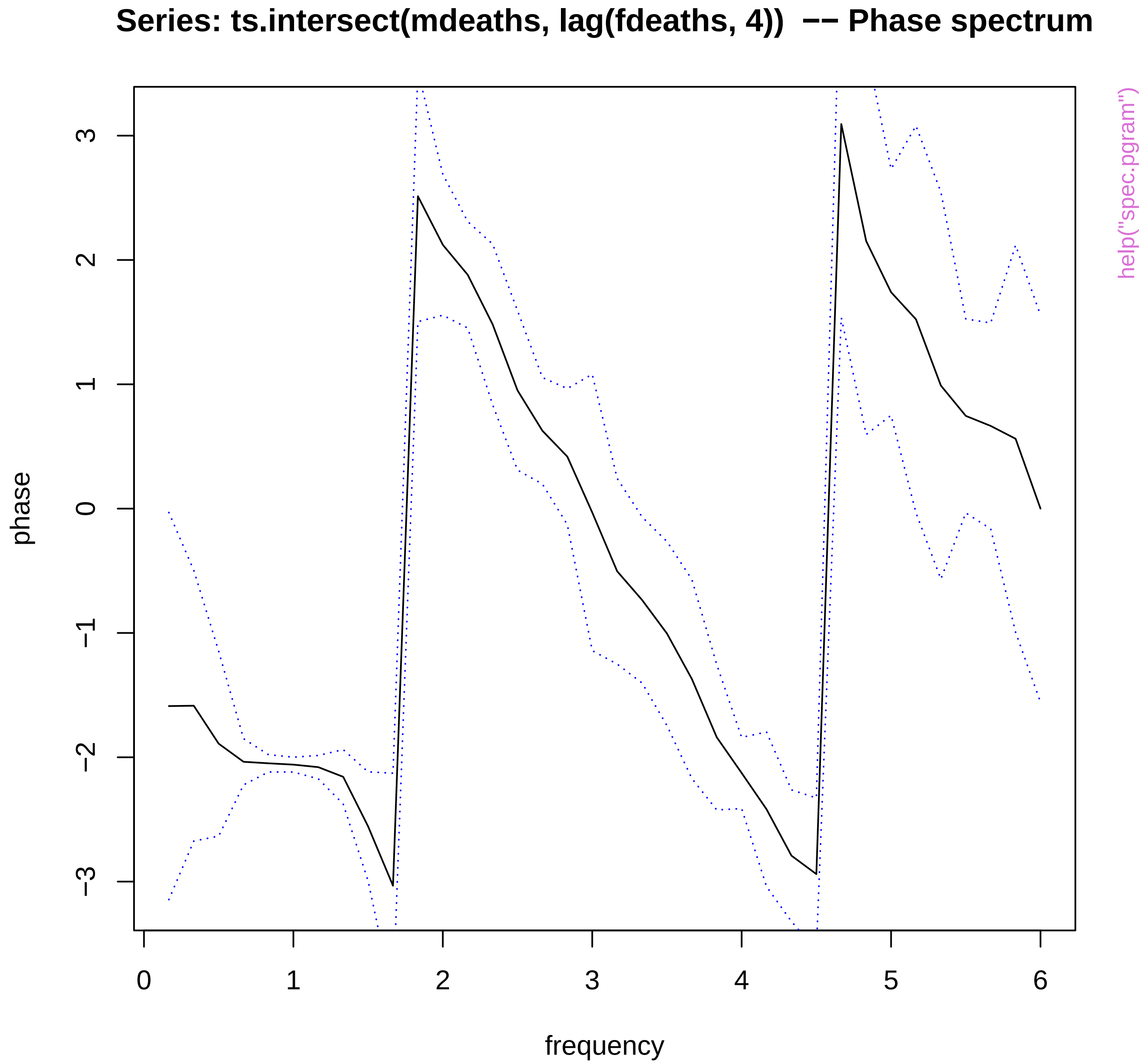
Series: x
Smoothed Periodogram
spectrum
20000
10000
5000
2000
1000
500
200
100
0
1
2
3
4
5
6
frequency
bandwidth = 0.173
help("spec.pgram")


Series: ts.union(mdeaths, fdeaths)
Smoothed Periodogram
spectrum
5e+04
5e+03
5e+02
1e+02
0
1
2
3
4
5
6
frequency
bandwidth = 0.173
help("spec.pgram")

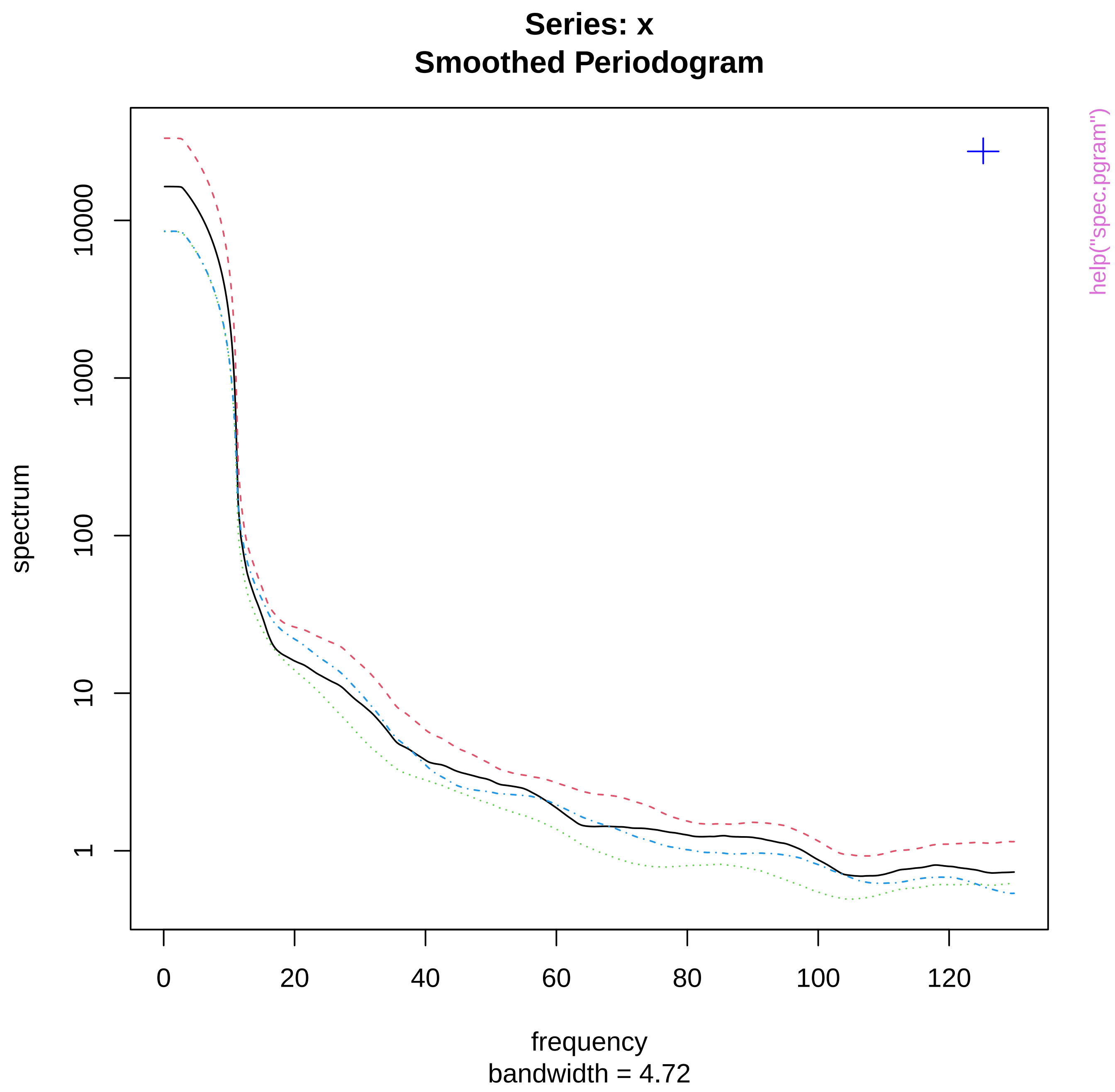

Series: ts.union(mdeaths, fdeaths) -- Squared Coherency
squared coherency
frequency
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
6
help("spec.pgram")

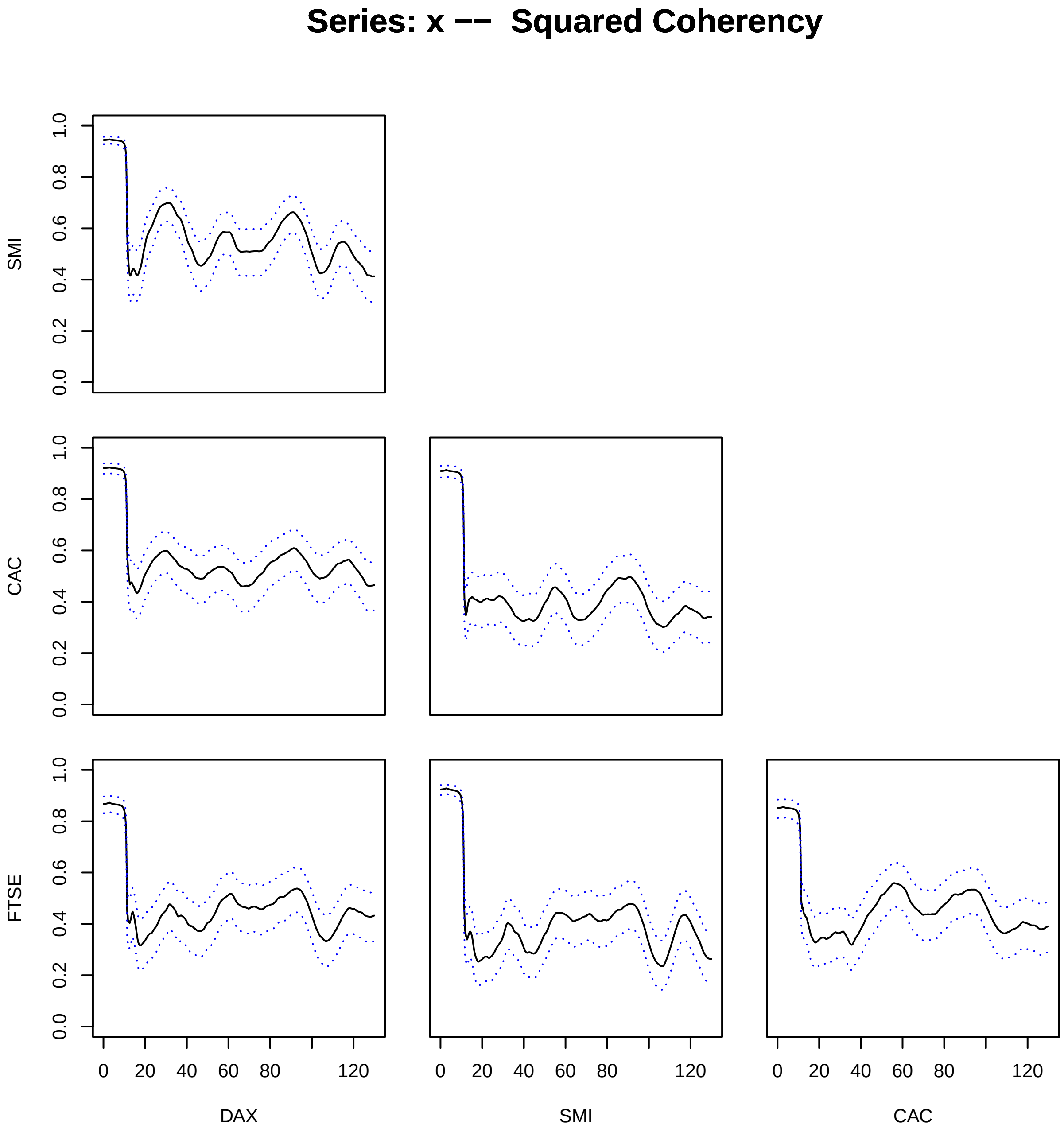

Series: ts.union(mdeaths, fdeaths) -- Phase spectrum
phase
frequency
3
2
1
0
−1
−2
−3
0
1
2
3
4
5
6
help("spec.pgram")

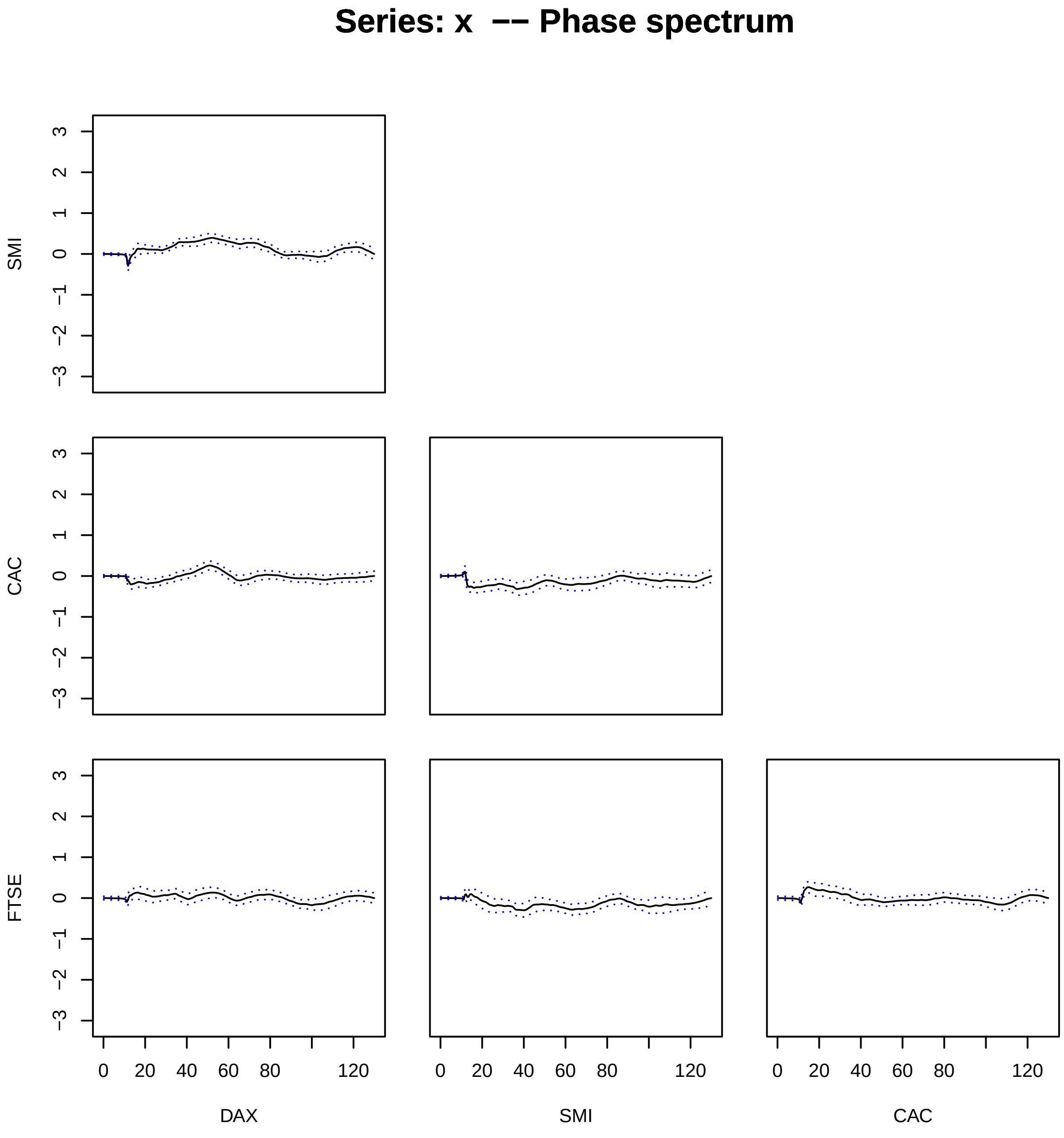

Series: ts.intersect(mdeaths, lag(fdeaths, 4)) -- Squared Coherency
squared coherency
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
6
frequency
help("spec.pgram")

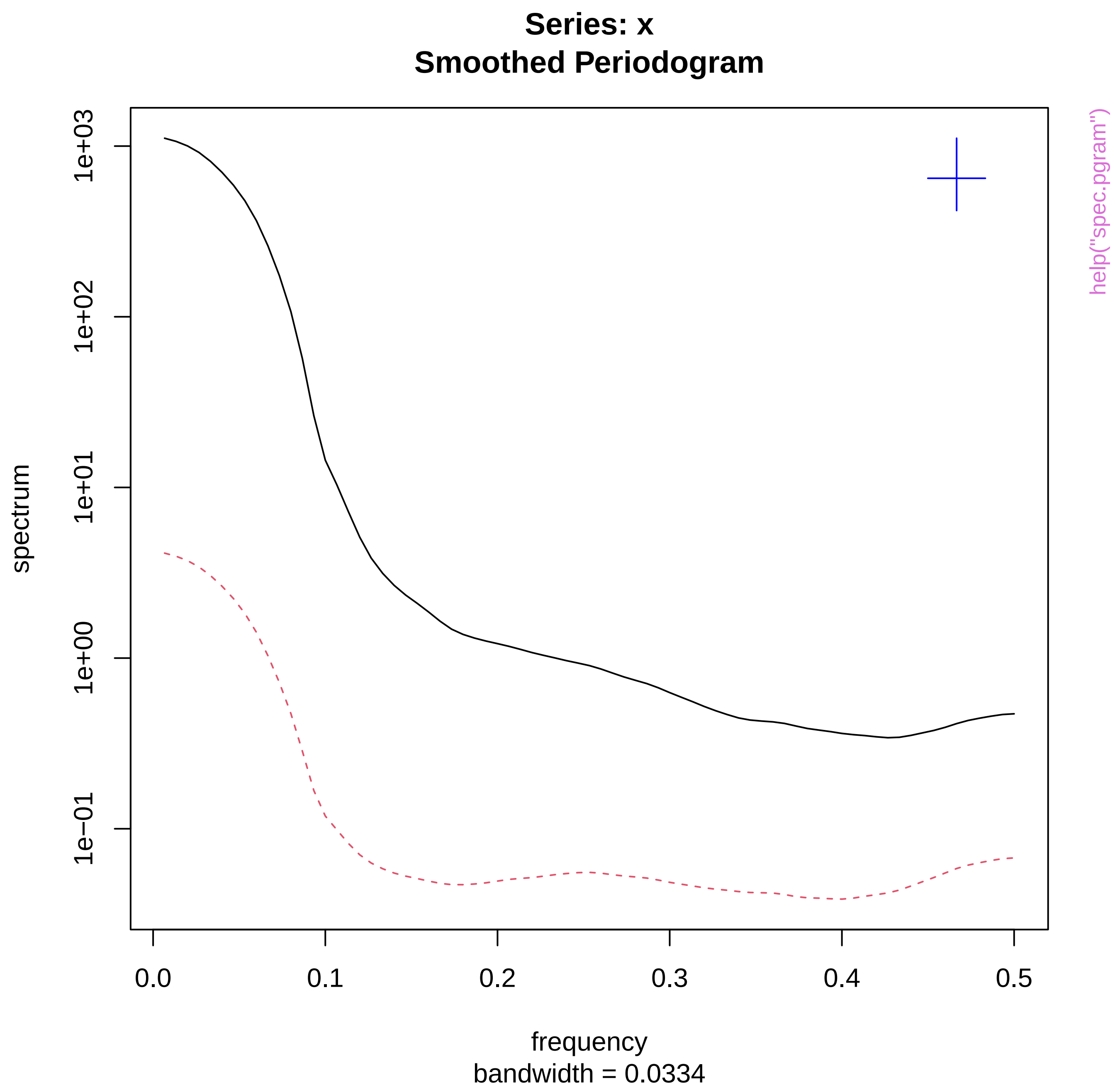


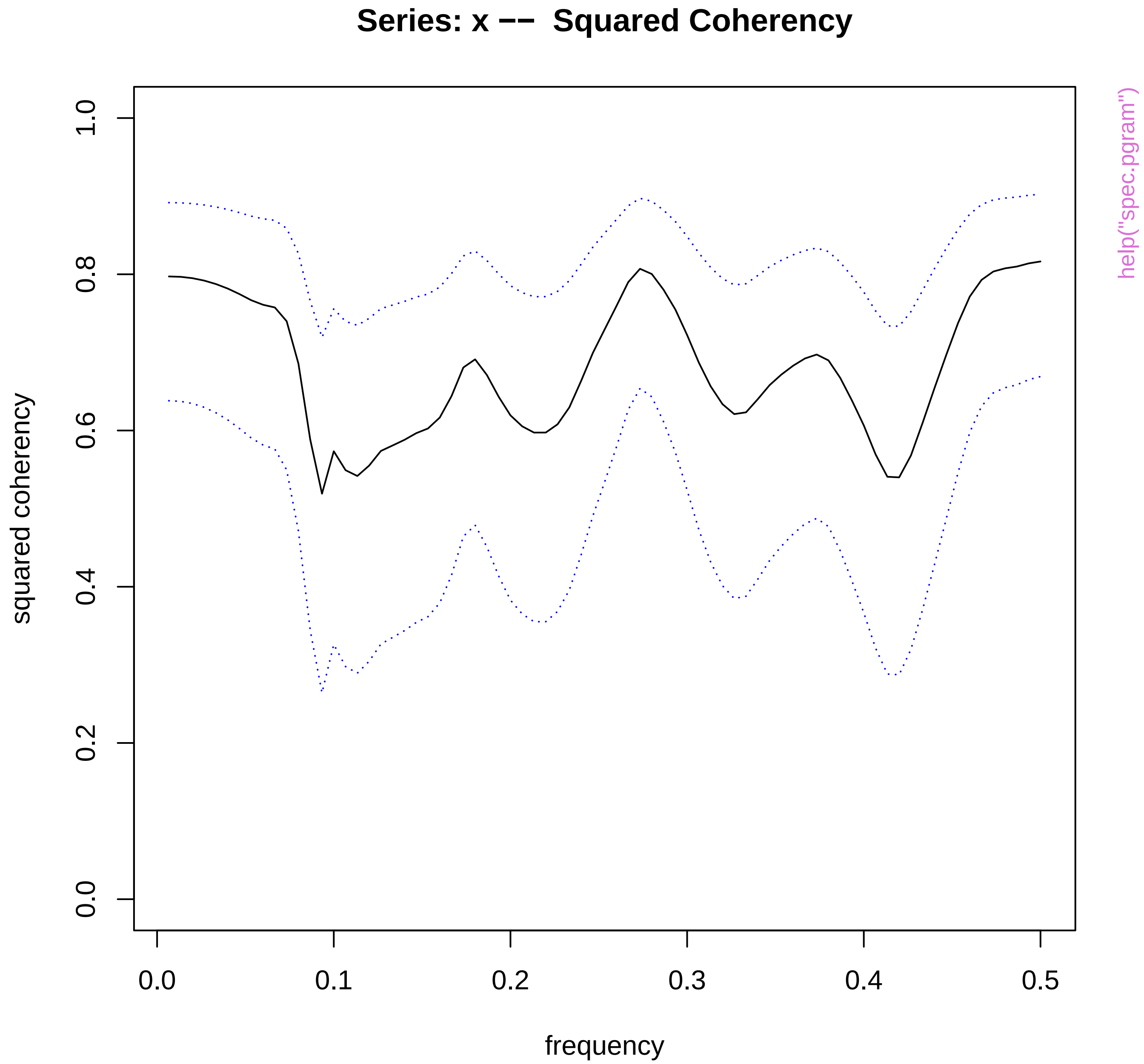
Series: ts.intersect(mdeaths, lag(fdeaths, 4)) -- Phase spectrum
phase
frequency
3
2
1
0
−1
−2
−3
0
1
2
3
4
5
6
help("spec.pgram")

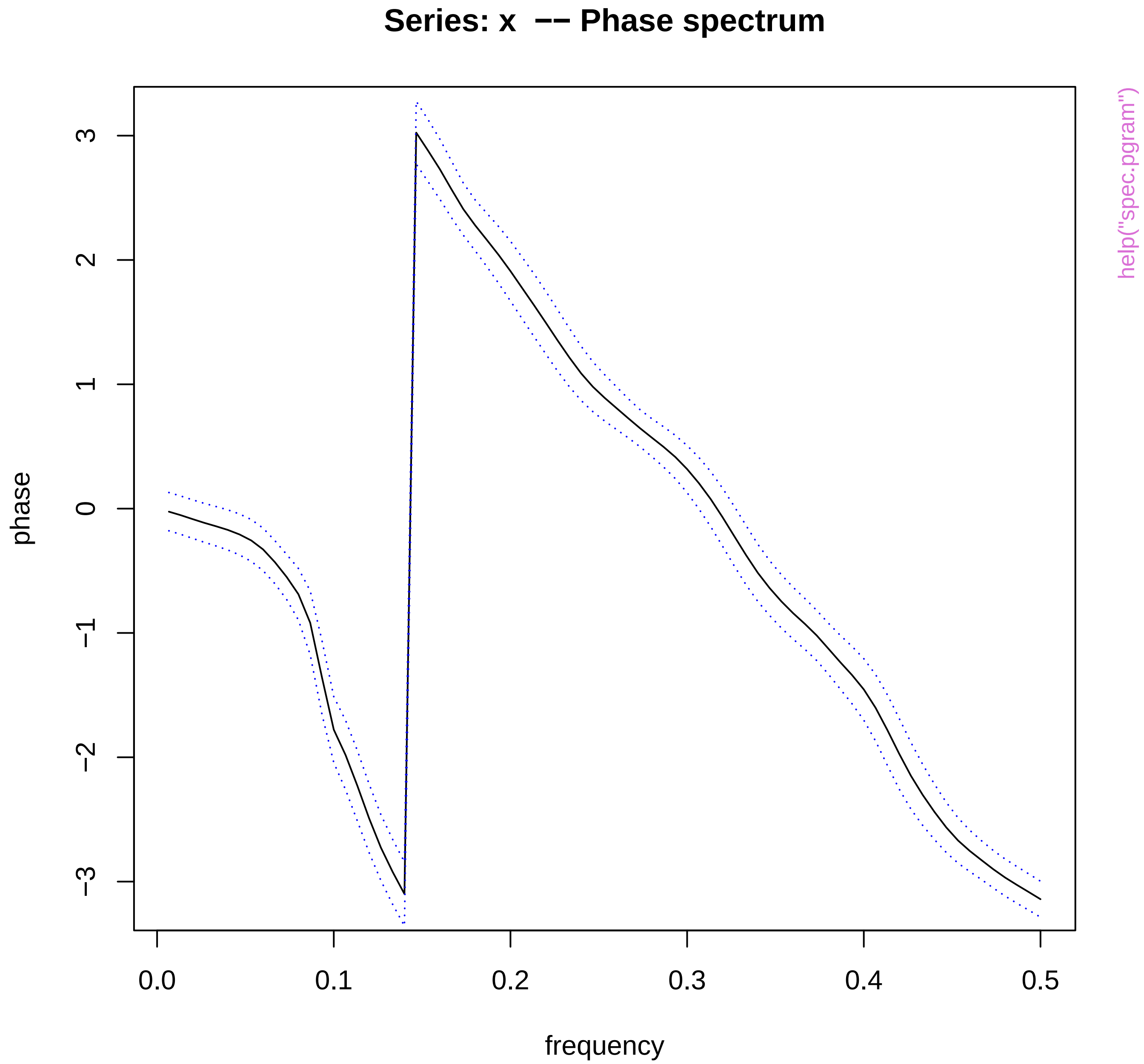

Series: x
Smoothed Periodogram
spectrum
10000
1000
100
10
1
0
20
40
60
80
100
120
frequency
bandwidth = 4.72
help("spec.pgram")

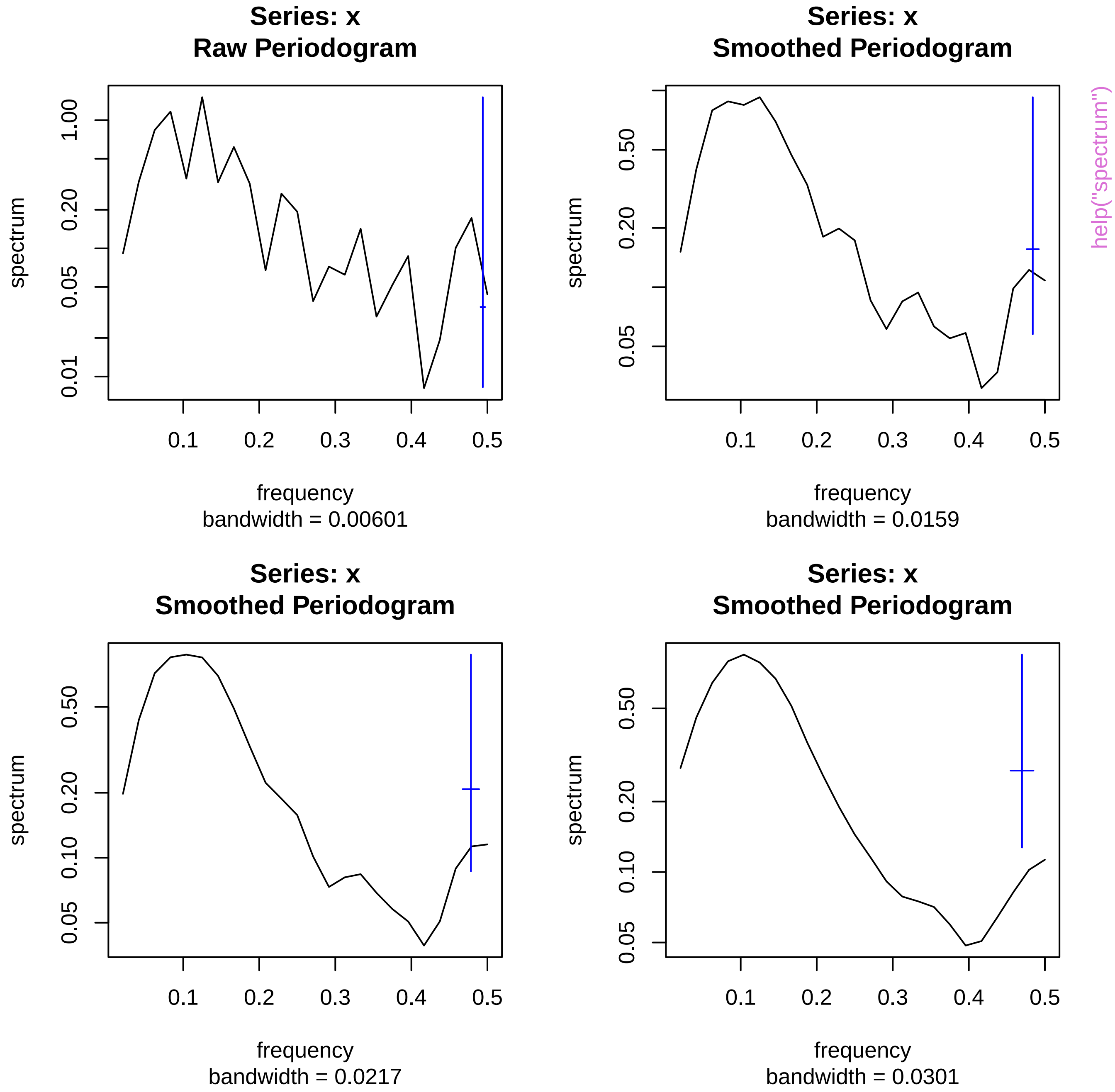

Series: x -- Squared Coherency
SMI
CAC
FTSE
DAX
SMI
CAC
0.0
0.2
0.4
0.6
0.8
1.0
0
20
40
60
80
120

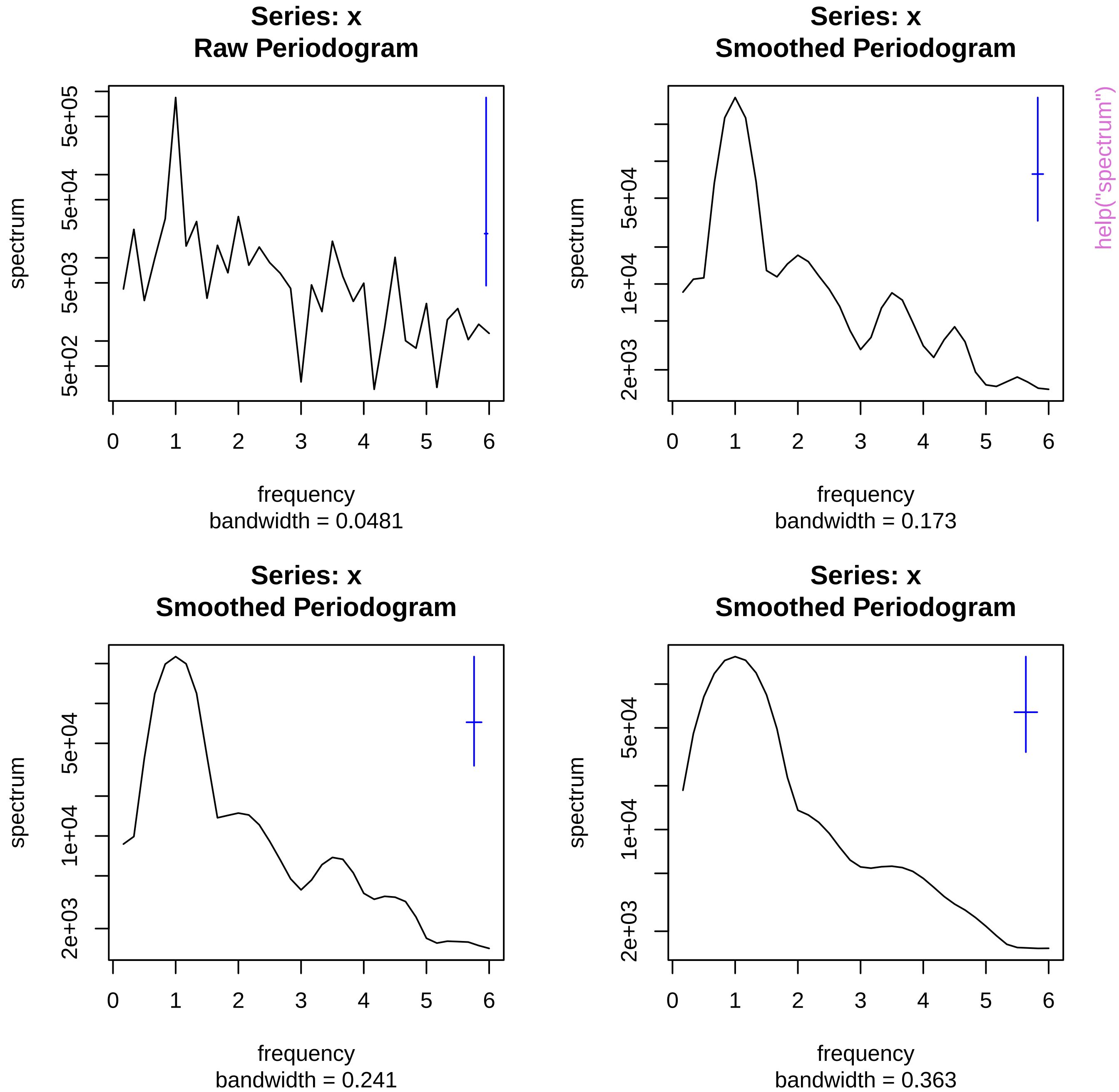

Series: x -- Phase spectrum
SMI
CAC
FTSE
DAX
SMI
CAC
3
2
1
0
−1
−2
−3
0
20
40
60
80
120


Series: x
Smoothed Periodogram
spectrum
1e+03
1e+02
1e+01
1e+00
1e−01
0.0
0.1
0.2
0.3
0.4
0.5
frequency
bandwidth = 0.0334
help("spec.pgram")

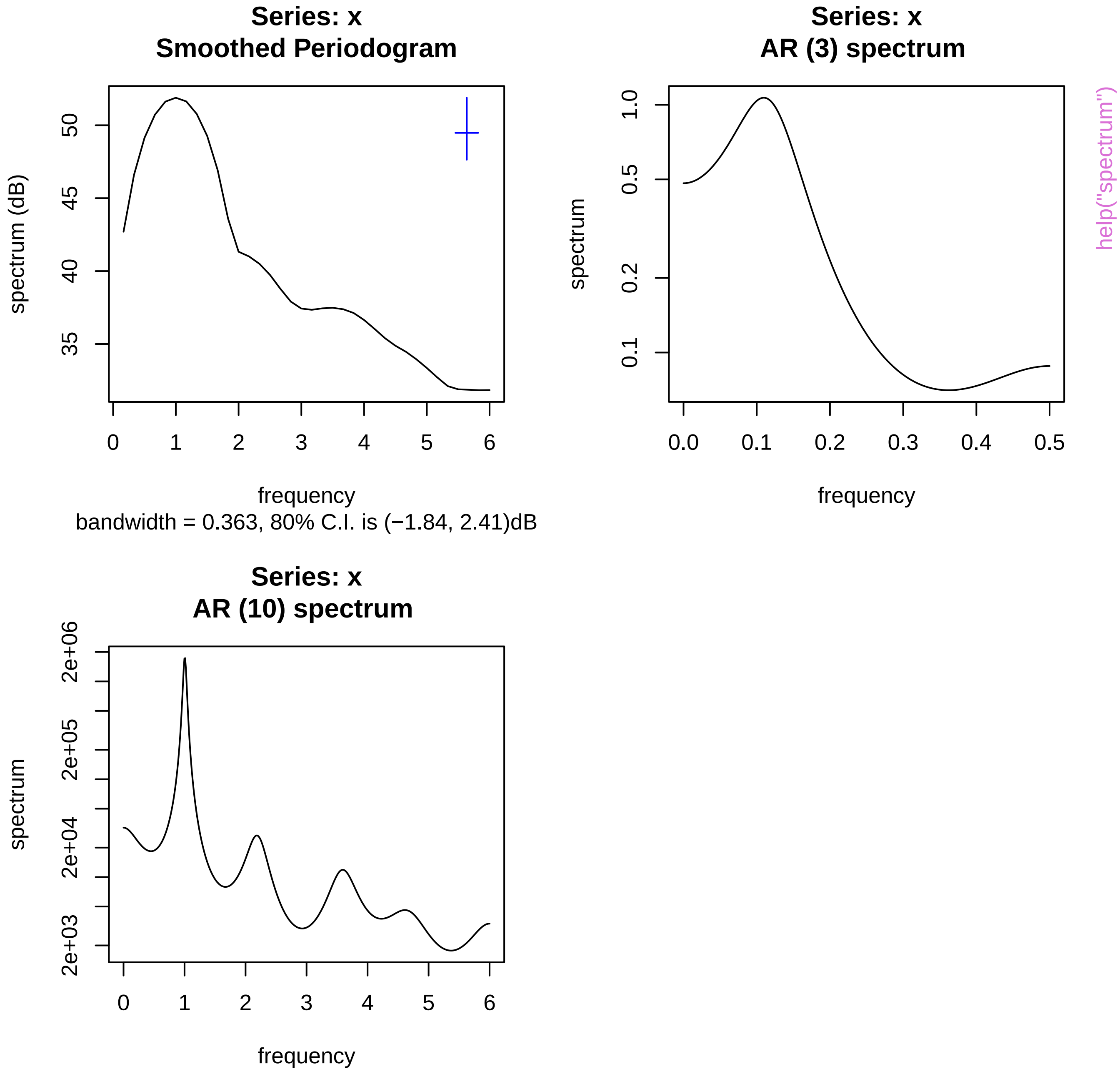

Series: x -- Squared Coherency
squared coherency
frequency
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.1
0.2
0.3
0.4
0.5
help("spec.pgram")

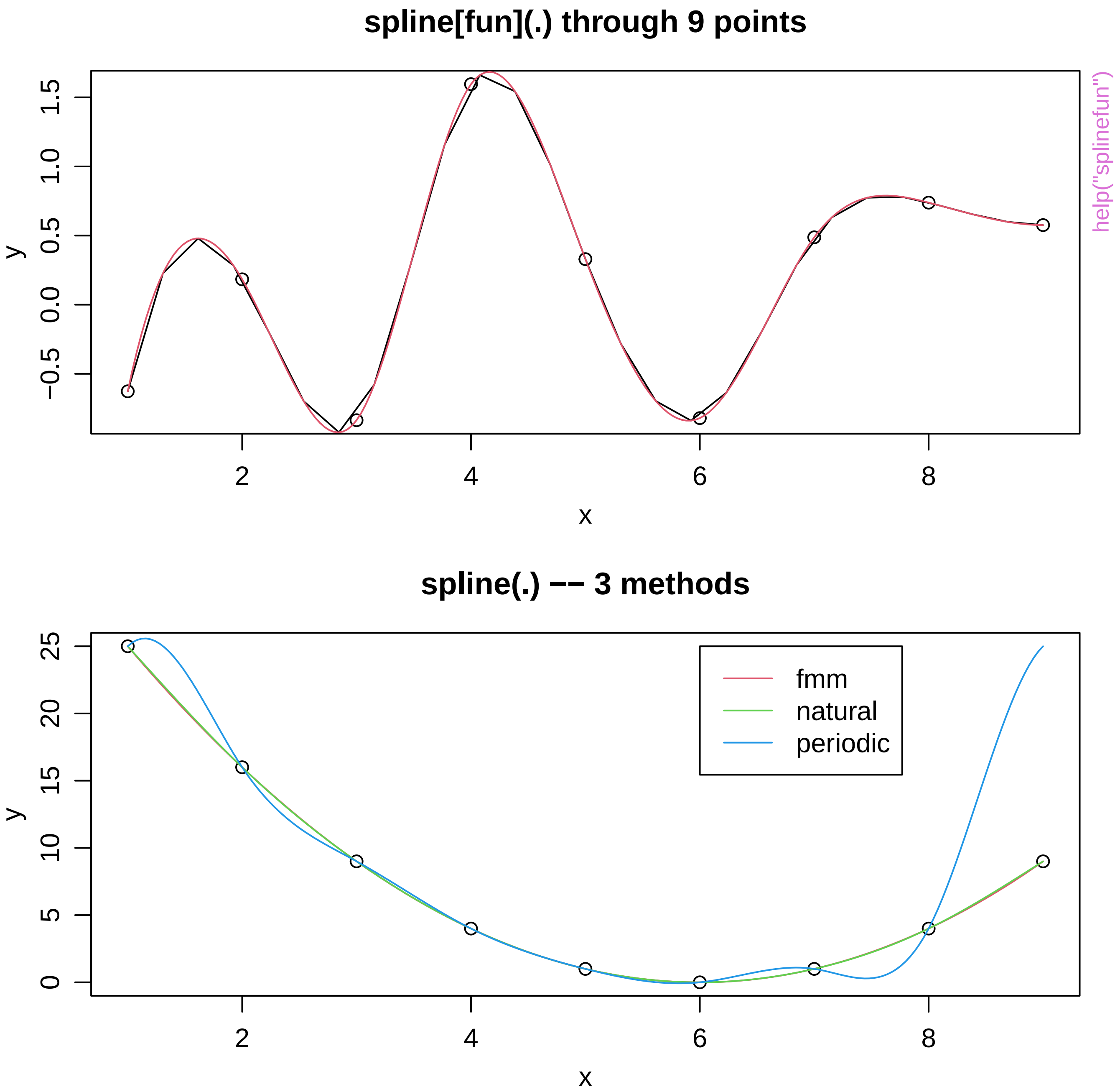

Series: x -- Phase spectrum
phase
3
2
1
0
−1
−2
−3
0.0
0.1
0.2
0.3
0.4
0.5
frequency
help("spec.pgram")

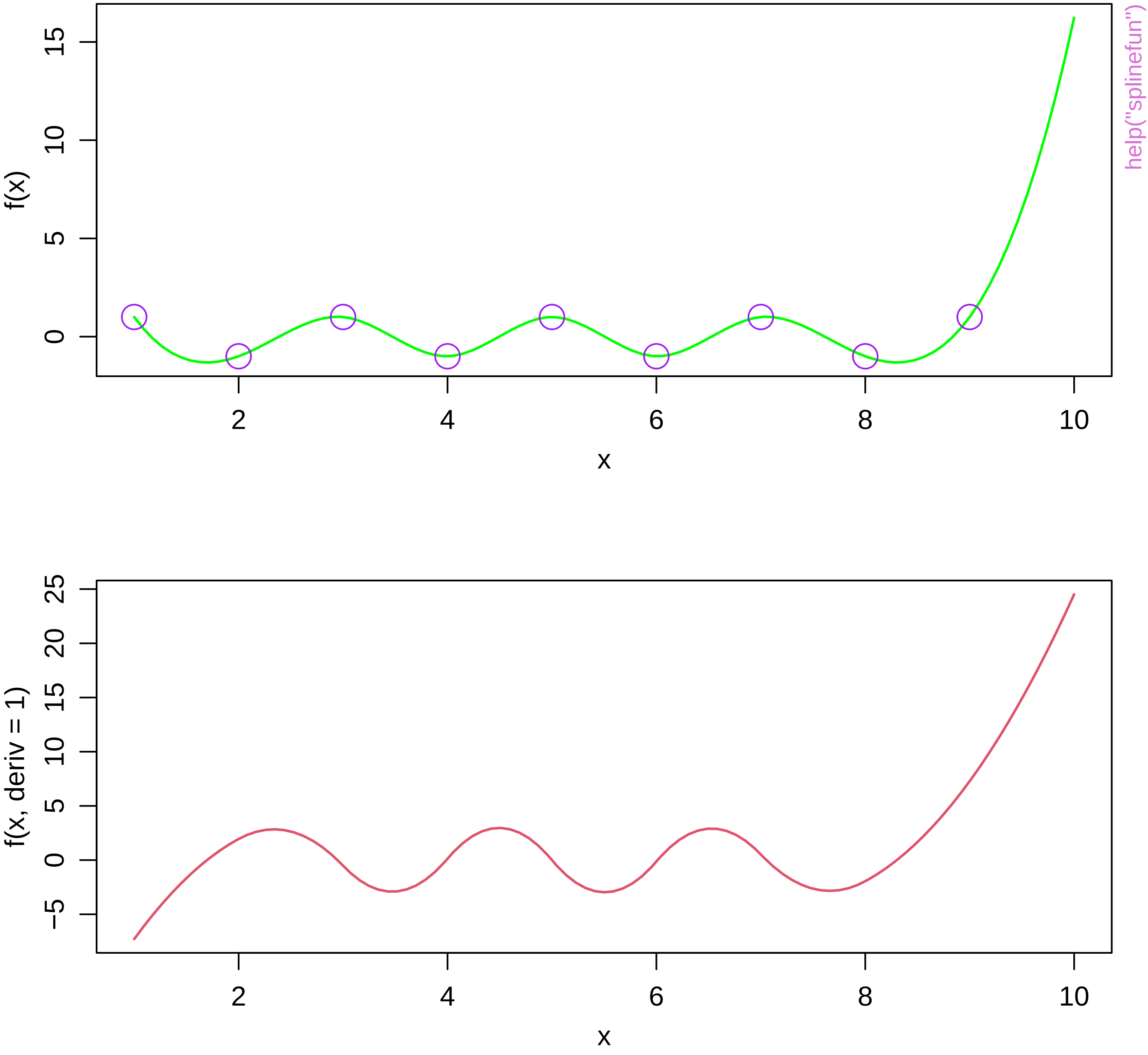

Series: x
Raw Periodogram
spectrum
1.00
0.20
0.05
0.01
0.1
0.2
0.3
0.4
0.5
frequency
bandwidth = 0.00601
Series: x
Smoothed Periodogram
spectrum
0.50
0.20
0.05
0.1
0.2
0.3
0.4
0.5
frequency
bandwidth = 0.0159
help("spectrum")
Series: x
Smoothed Periodogram
spectrum
0.50
0.20
0.10
0.05
0.1
0.2
0.3
0.4
0.5
frequency
bandwidth = 0.0217
Series: x
Smoothed Periodogram
spectrum
0.50
0.20
0.10
0.05
0.1
0.2
0.3
0.4
0.5
frequency
bandwidth = 0.0301



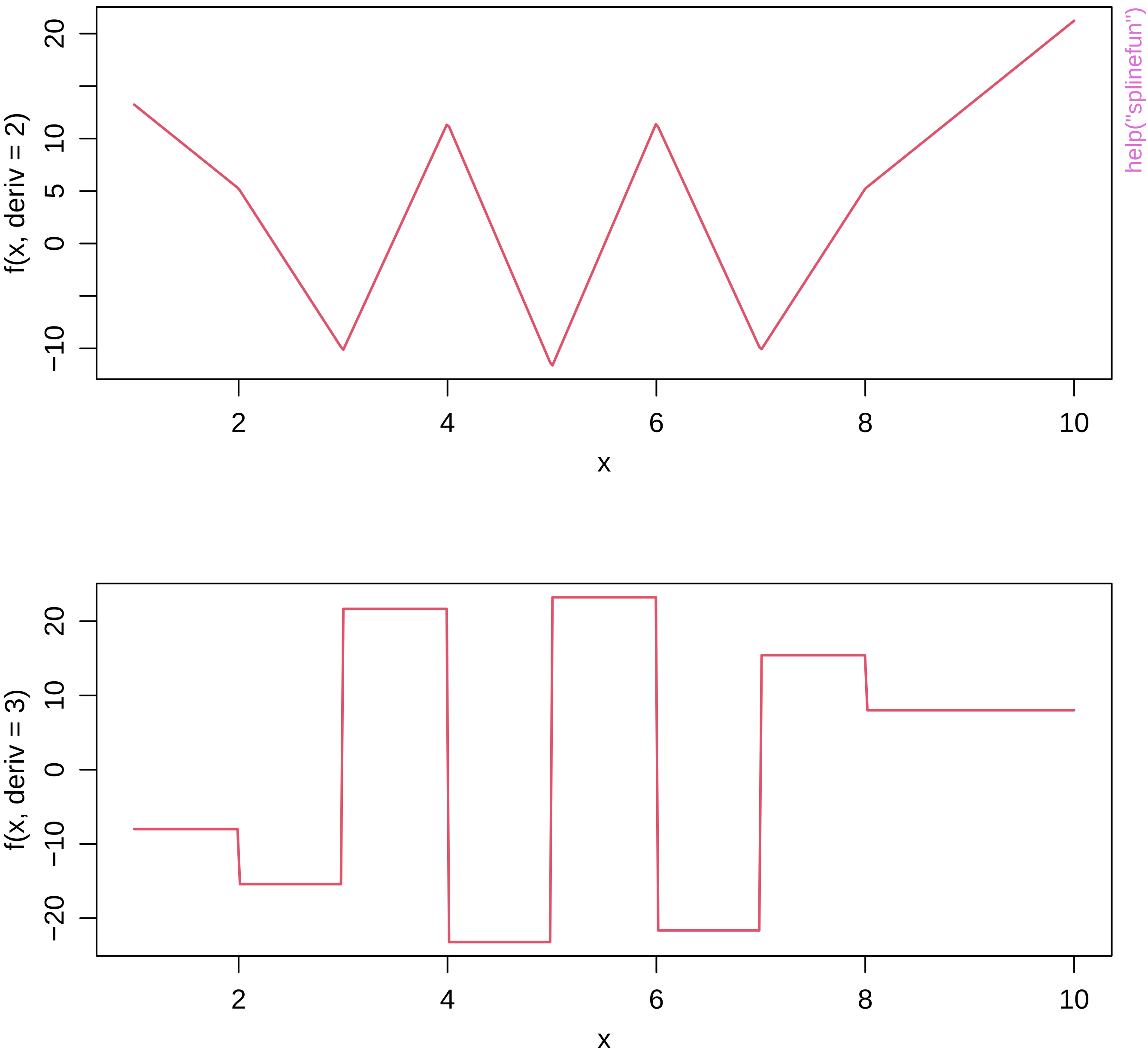
Series: x
Raw Periodogram
spectrum
5e+02
5e+03
5e+04
5e+05
0
1
2
3
4
5
6
frequency
bandwidth = 0.0481
Series: x
Smoothed Periodogram
spectrum
2e+03
1e+04
5e+04
0
1
2
3
4
5
6
frequency
bandwidth = 0.173
help("spectrum")
Series: x
Smoothed Periodogram
spectrum
2e+03
1e+04
5e+04
0
1
2
3
4
5
6
frequency
bandwidth = 0.241
Series: x
Smoothed Periodogram
spectrum
2e+03
1e+04
5e+04
0
1
2
3
4
5
6
frequency
bandwidth = 0.363


Series: x
Smoothed Periodogram
spectrum (dB)
frequency
bandwidth = 0.363, 80% C.I. is (−1.84, 2.41)dB
Series: x
AR (3) spectrum
spectrum
frequency
help("spectrum")
Series: x
AR (10) spectrum
spectrum
frequency

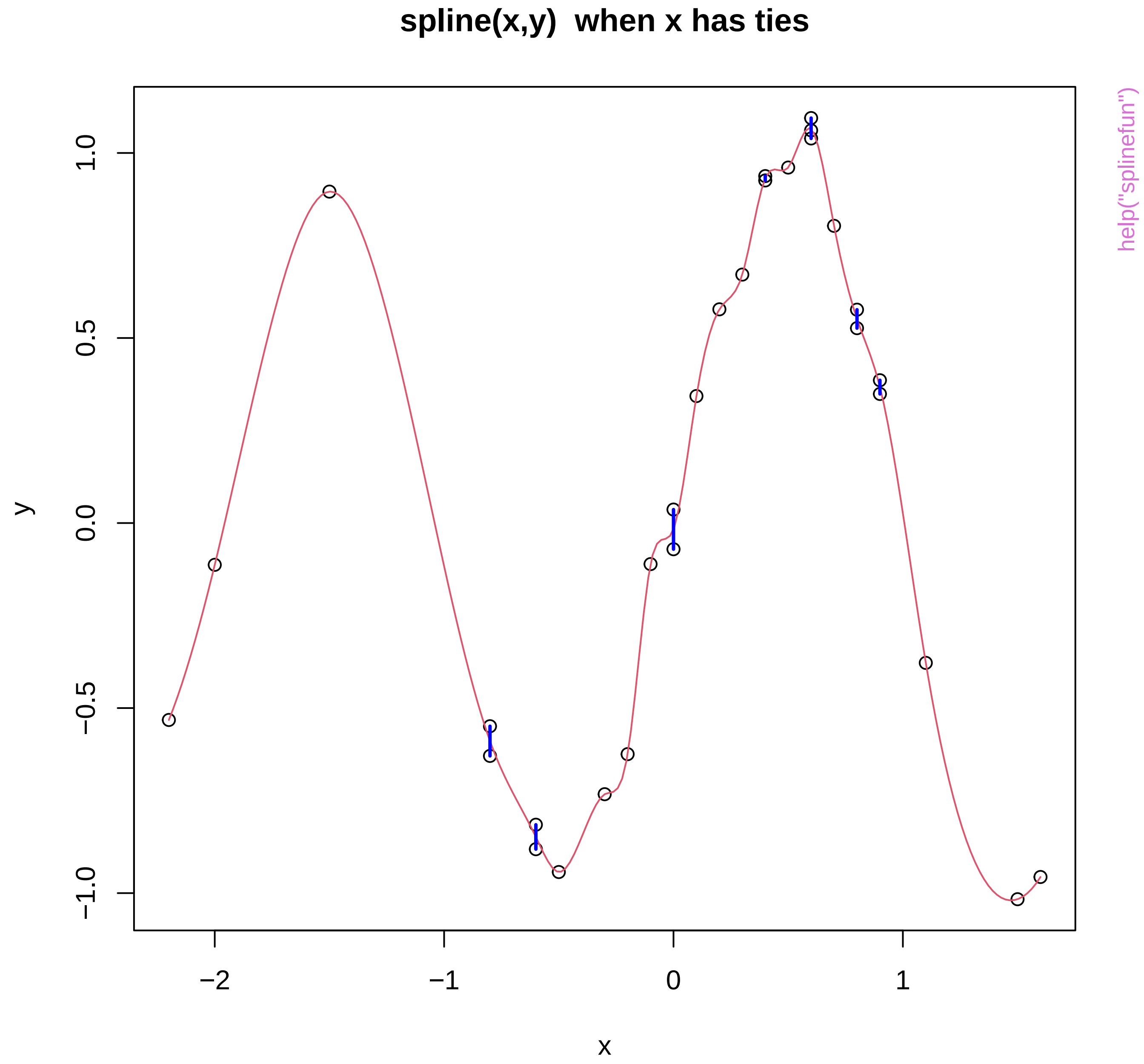

spline[fun](.) through 9 points
help("splinefun")
y
x
−0.5
0.0
0.5
1.0
1.5
2
4
6
8
spline(.) −− 3 methods
fmm
natural
periodic
0
5
10
15
20
25
y
x
2
4
6
8

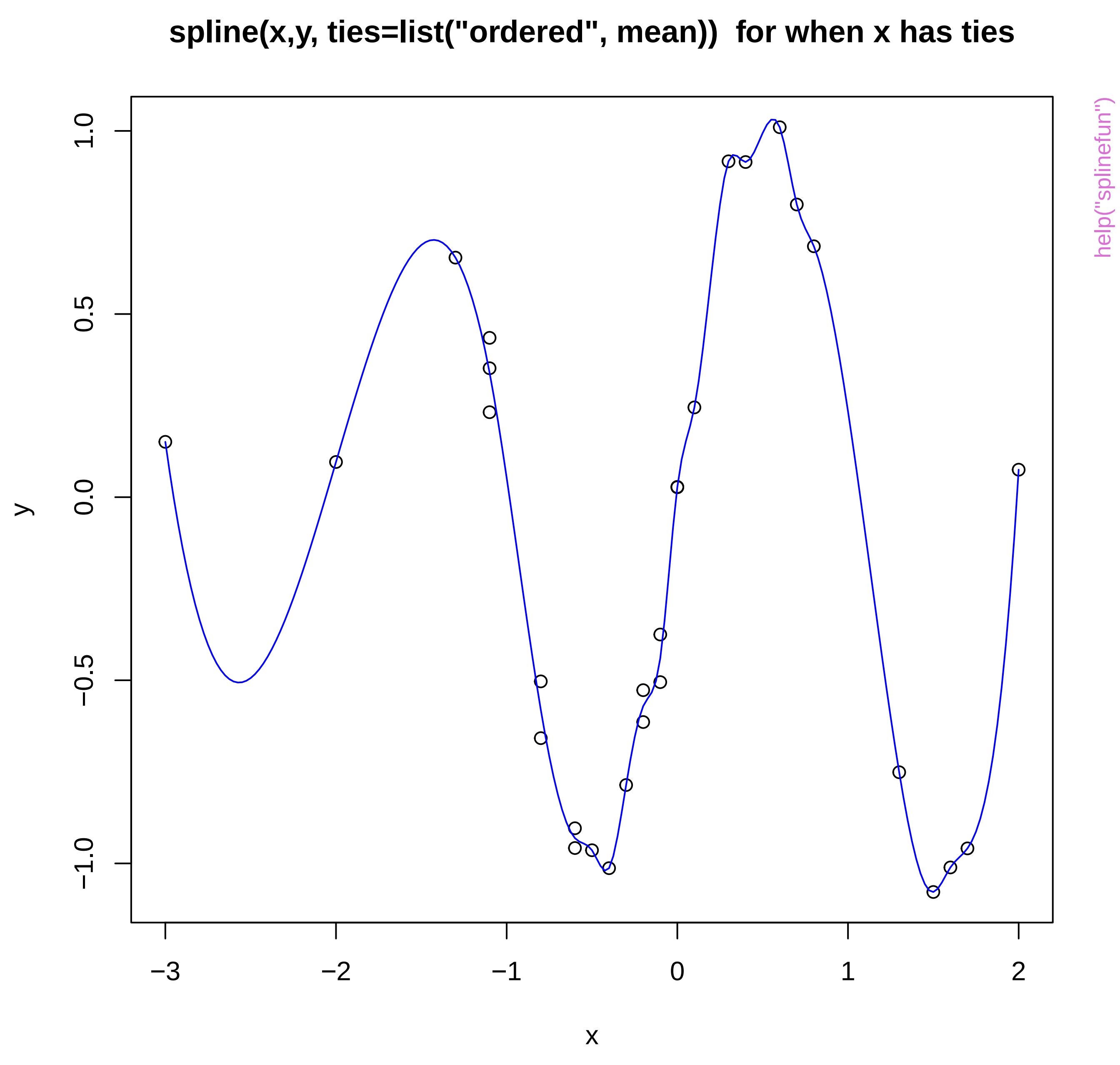


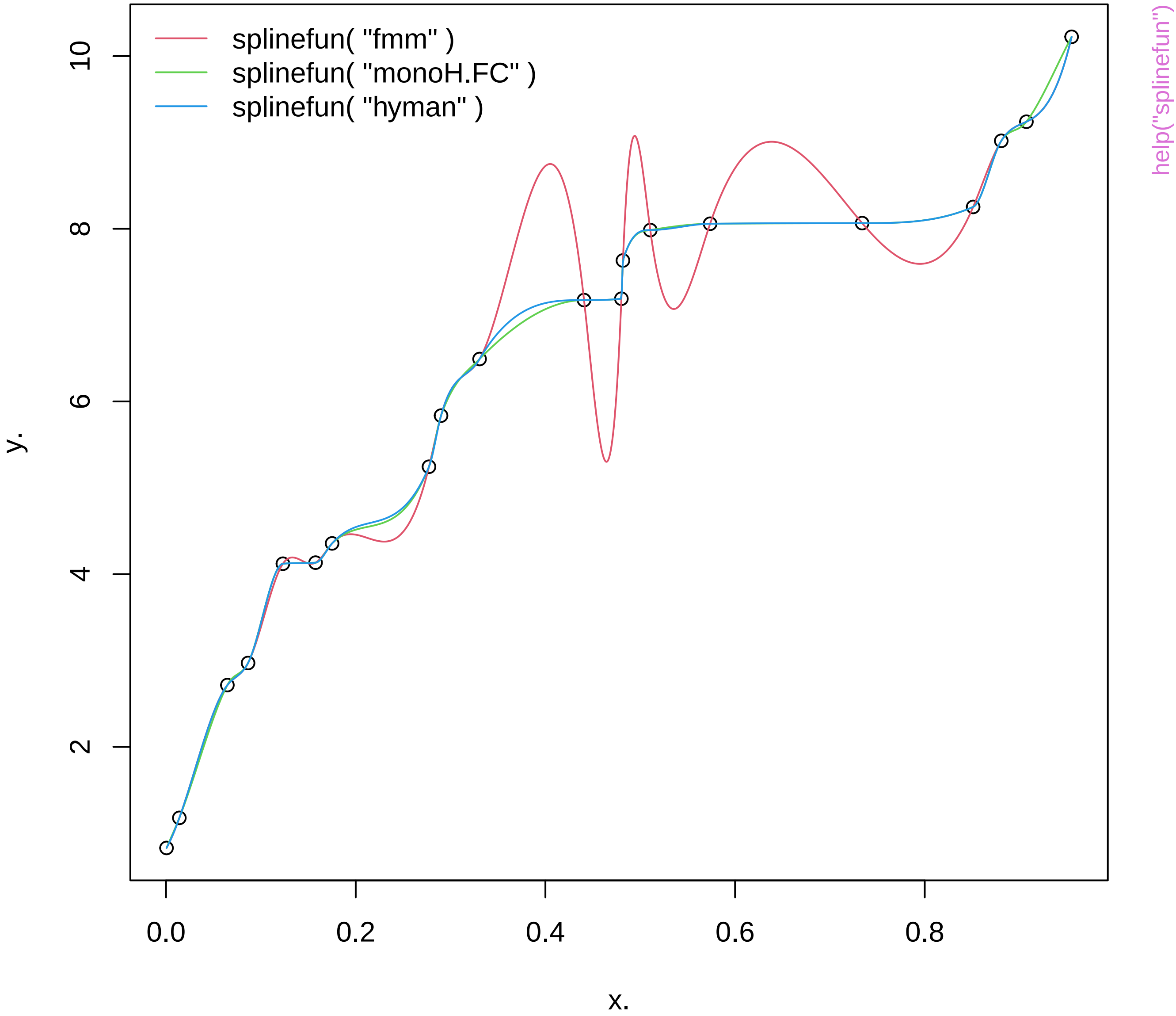
f(x)
0
5
10
15
2
4
6
8
10
x
help("splinefun")
f(x, deriv = 1)
−5
0
5
10
15
20
25
2
4
6
8
10
x



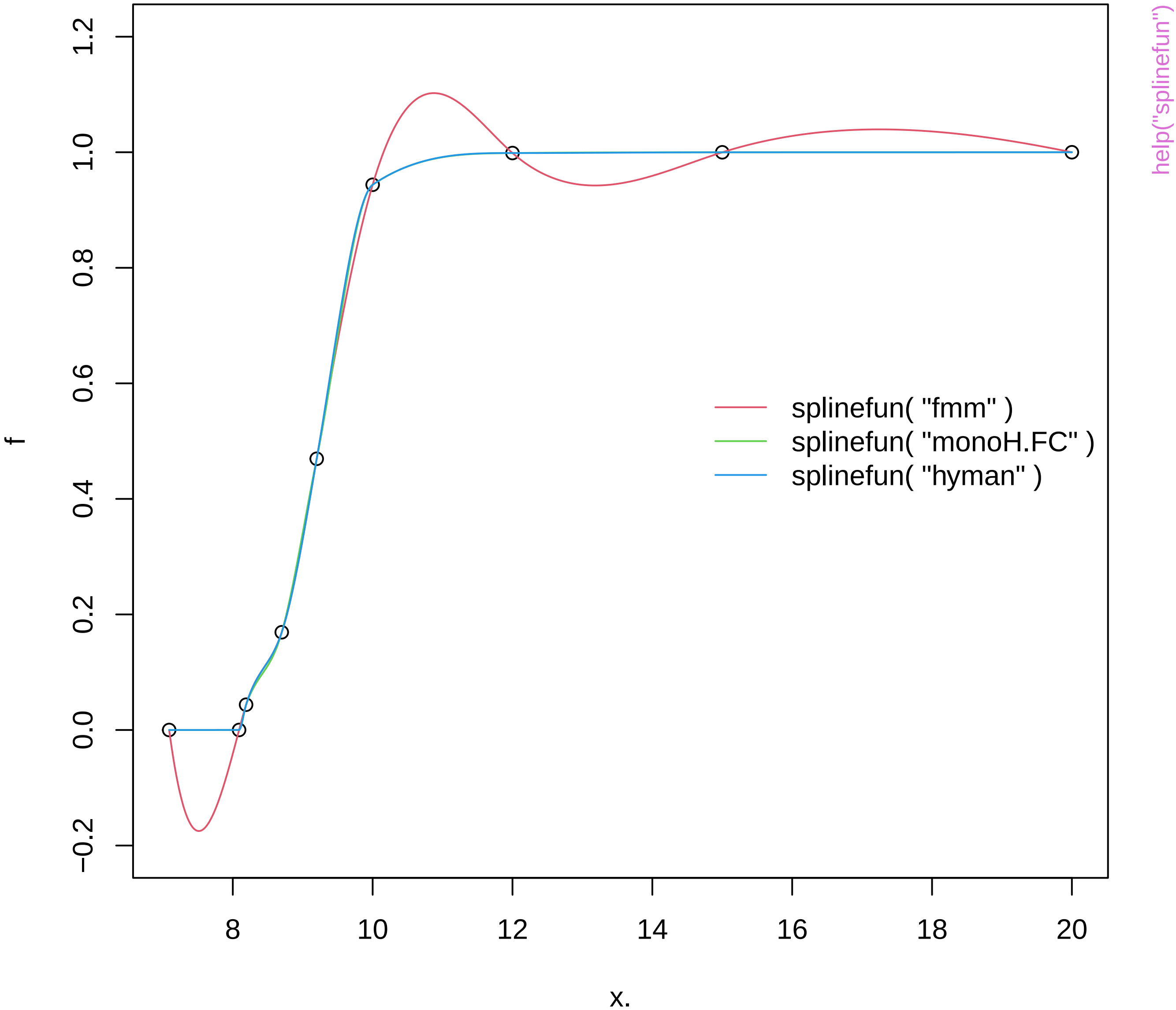
help("splinefun")
f(x, deriv = 2)
20
10
5
0
−10
2
4
6
8
10
x
f(x, deriv = 3)
20
10
0
−10
−20
2
4
6
8
10
x


spline(x,y) when x has ties
y
1.0
0.5
0.0
−0.5
−1.0
−2
−1
0
1
x
help("splinefun")

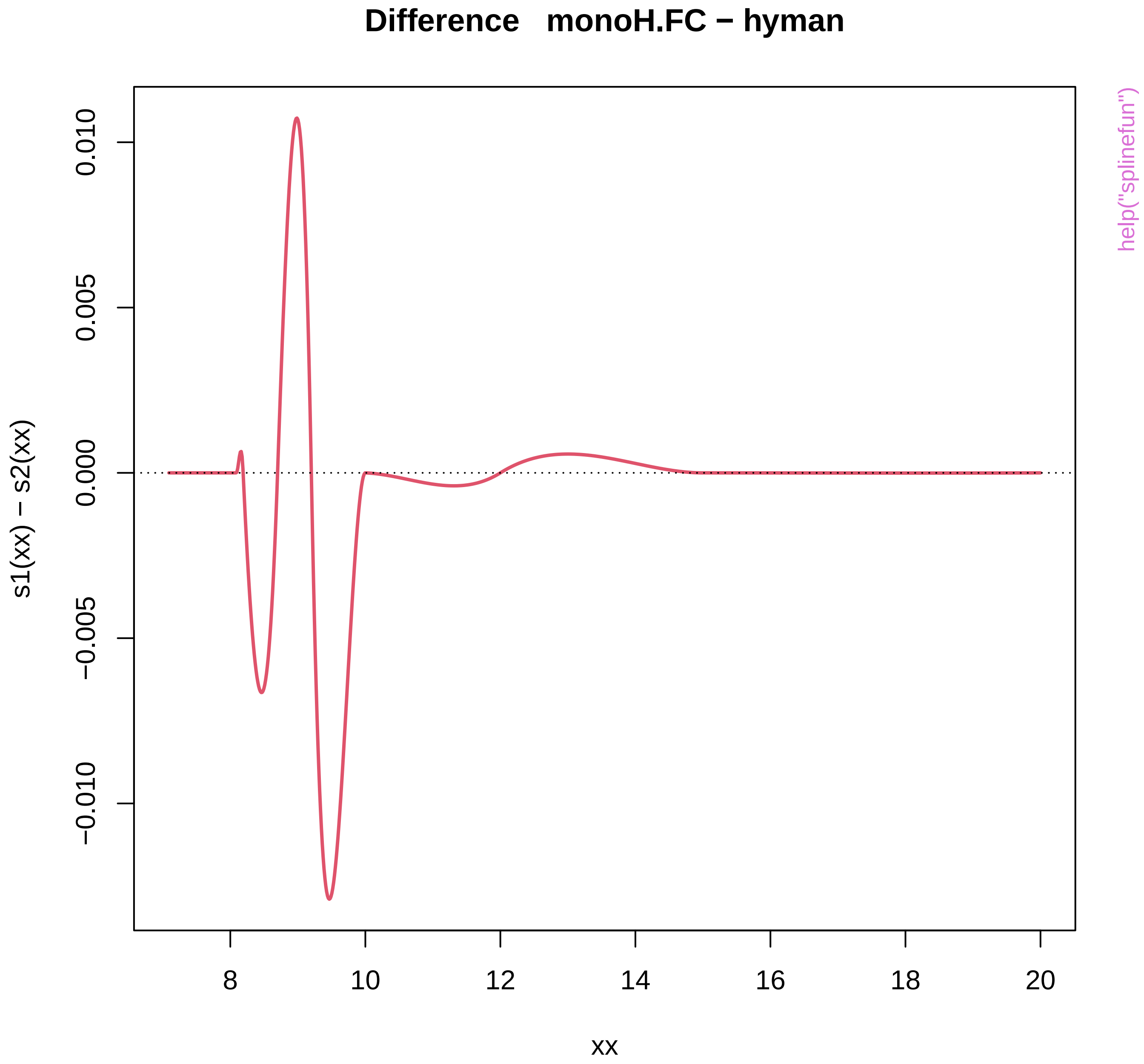

spline(x,y, ties=list("ordered", mean)) for when x has ties
help("splinefun")
y
x
1.0
0.5
0.0
−0.5
−1.0
−3
−2
−1
0
1
2



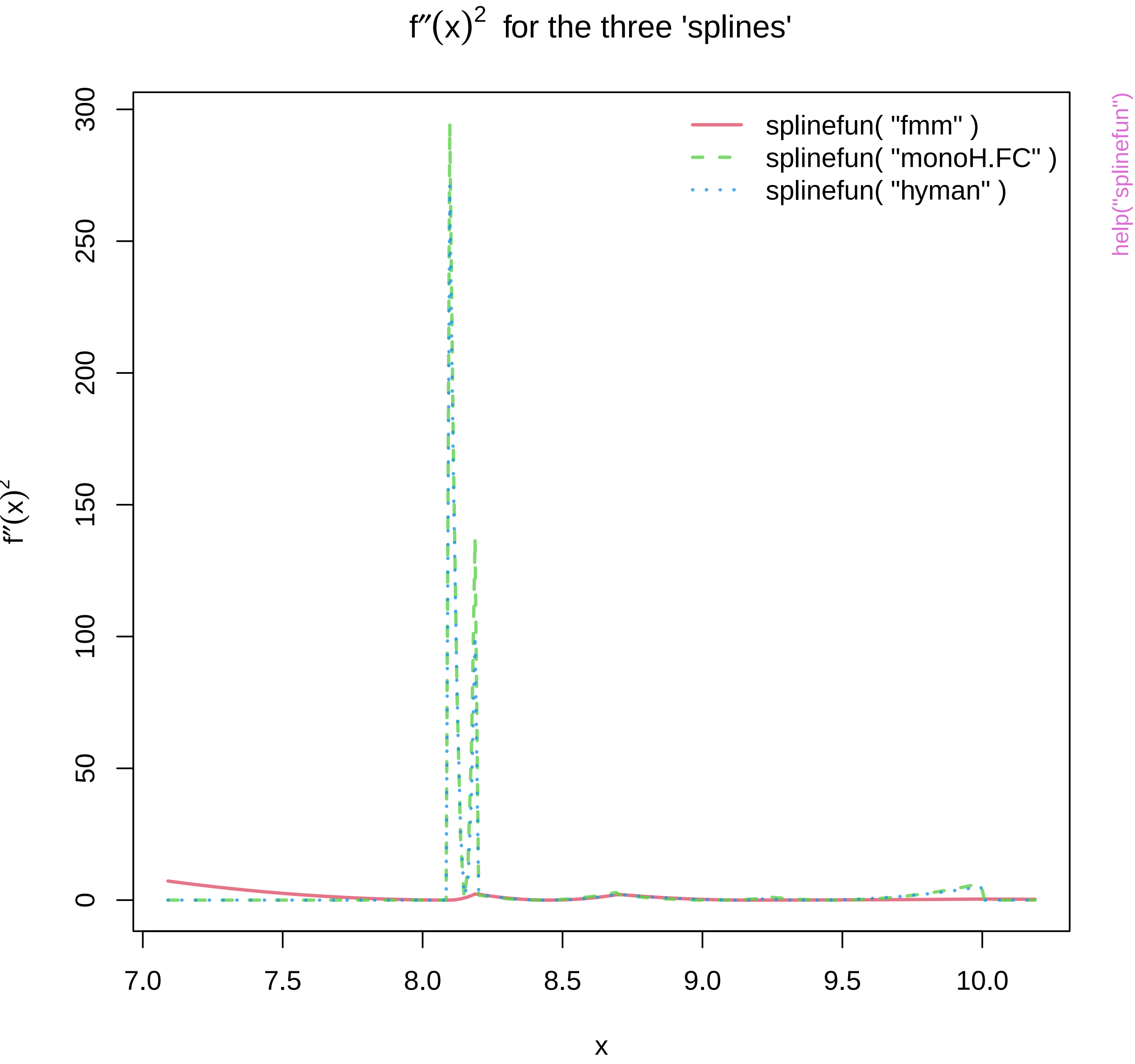
splinefun("fmm")
splinefun("monoH.FC")
splinefun("hyman")
help("splinefun")
y.
x.
2
4
6
8
10
0.0
0.2
0.4
0.6
0.8



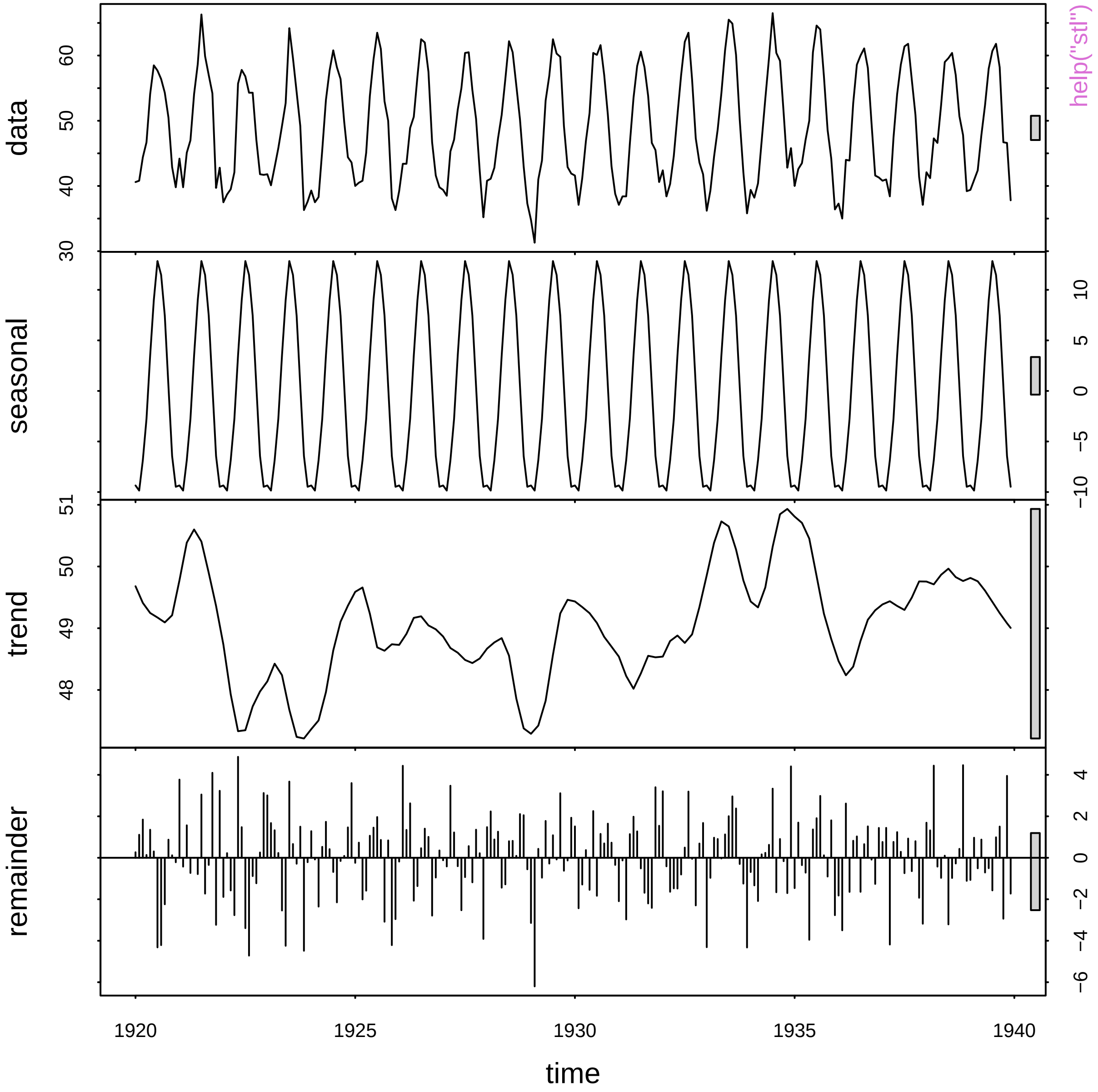
splinefun("fmm")
splinefun("monoH.FC")
splinefun("hyman")
help("splinefun")
f
x.
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
8
10
12
14
16
18
20



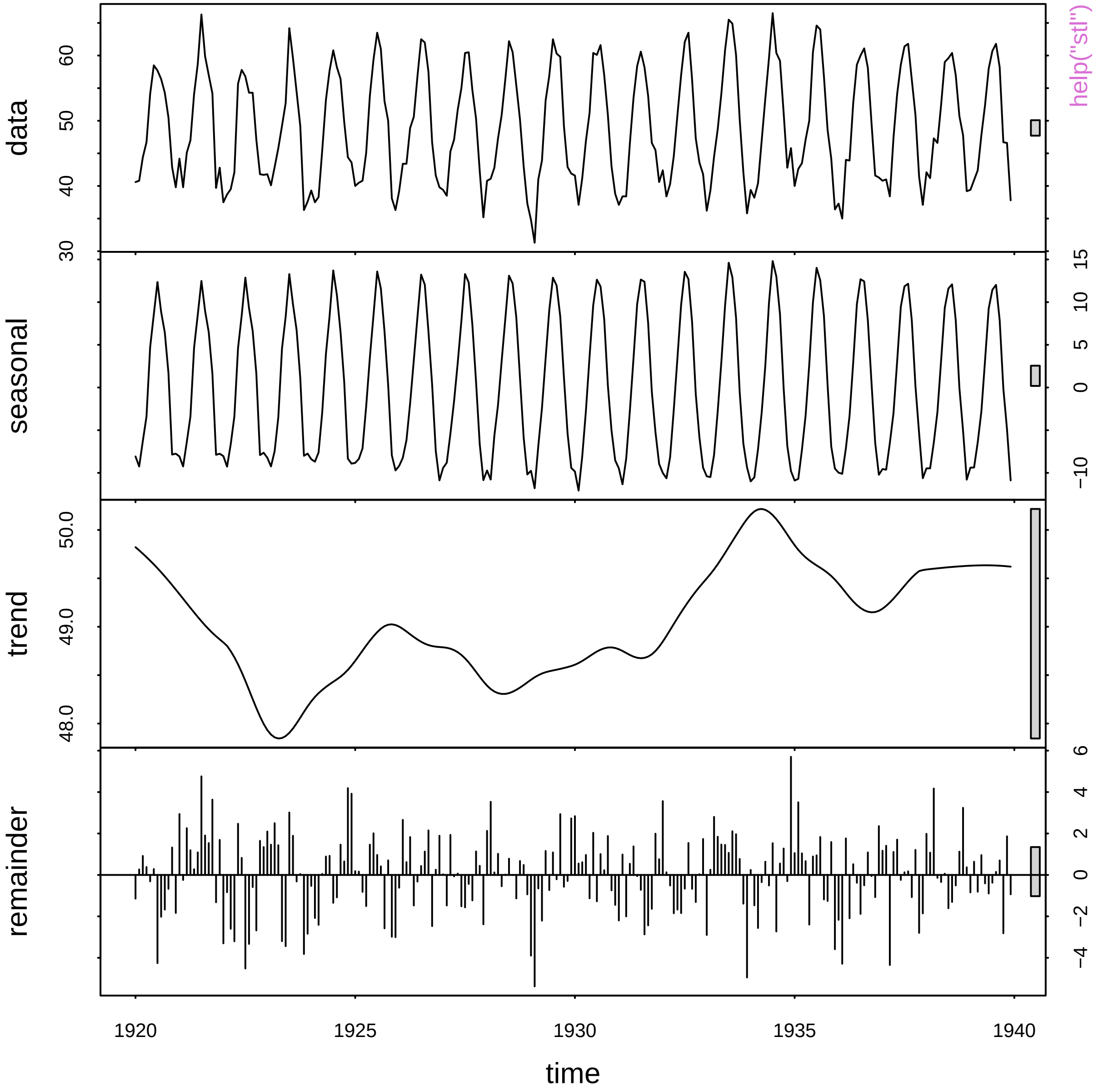
Difference monoH.FC − hyman
s1(xx) − s2(xx)
0.010
0.005
0.000
−0.005
−0.010
8
10
12
14
16
18
20
xx
help("splinefun")



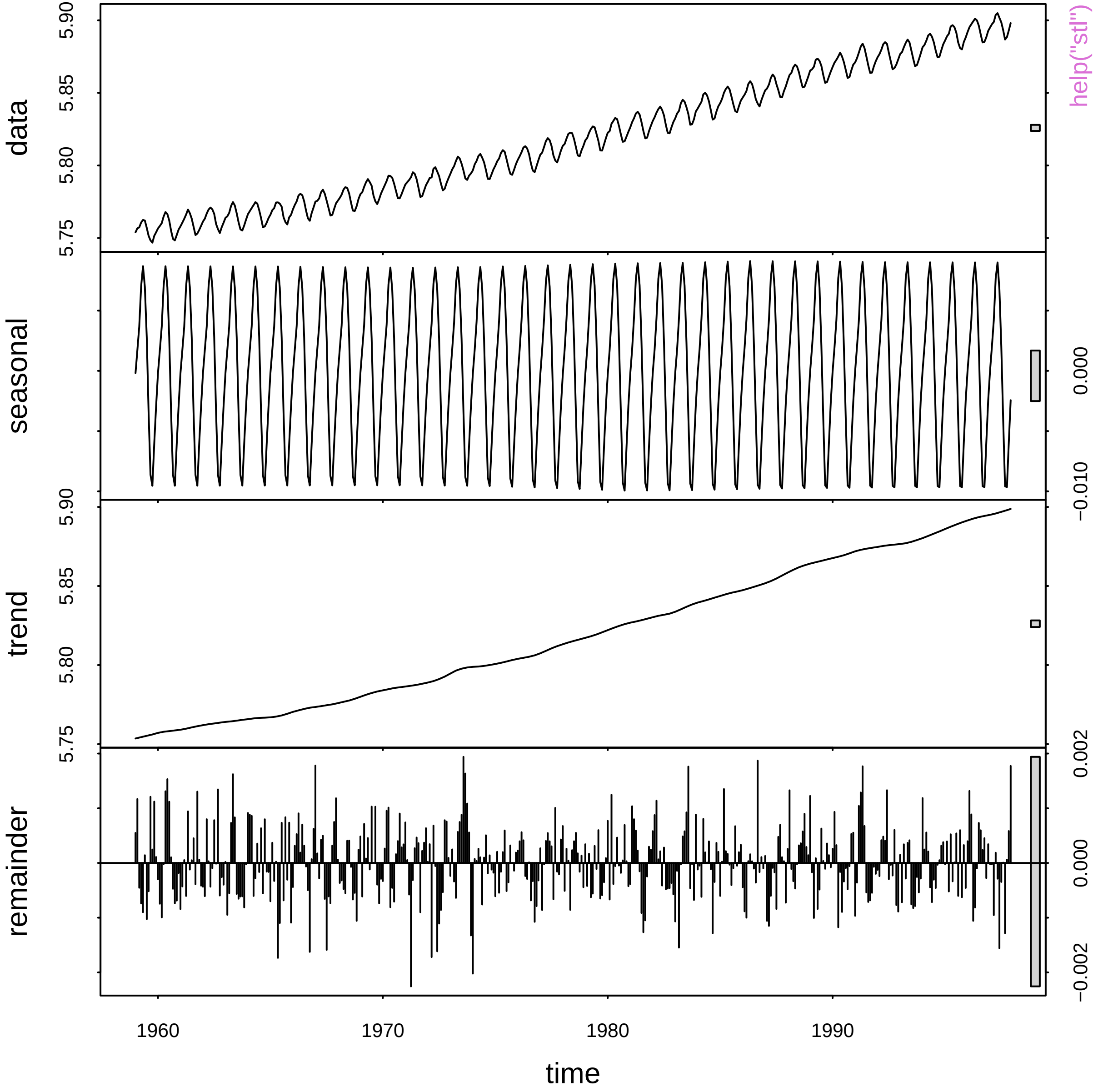
f″(x)^2 for the three 'splines'
splinefun("fmm")
splinefun("monoH.FC")
splinefun("hyman")
help("splinefun")
0
50
100
150
200
250
300
7.0
7.5
8.0
8.5
9.0
9.5
10.0
x
f″(x)^2



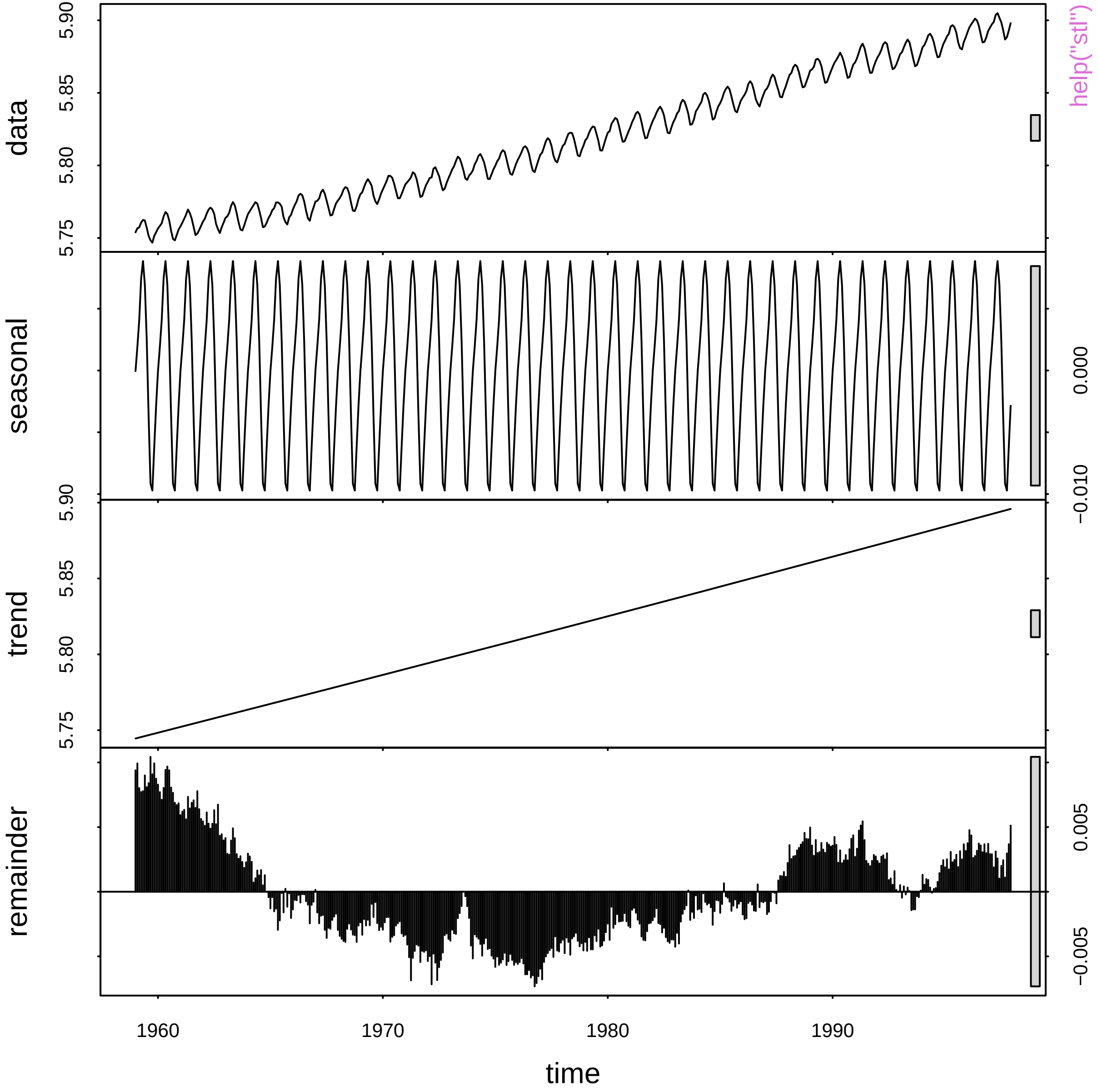
data
seasonal
trend
remainder
help("stl")
30
40
50
60
−10
−5
0
5
10
48
49
50
51
−6
−4
−2
0
2
4
1920
1925
1930
1935
1940
time



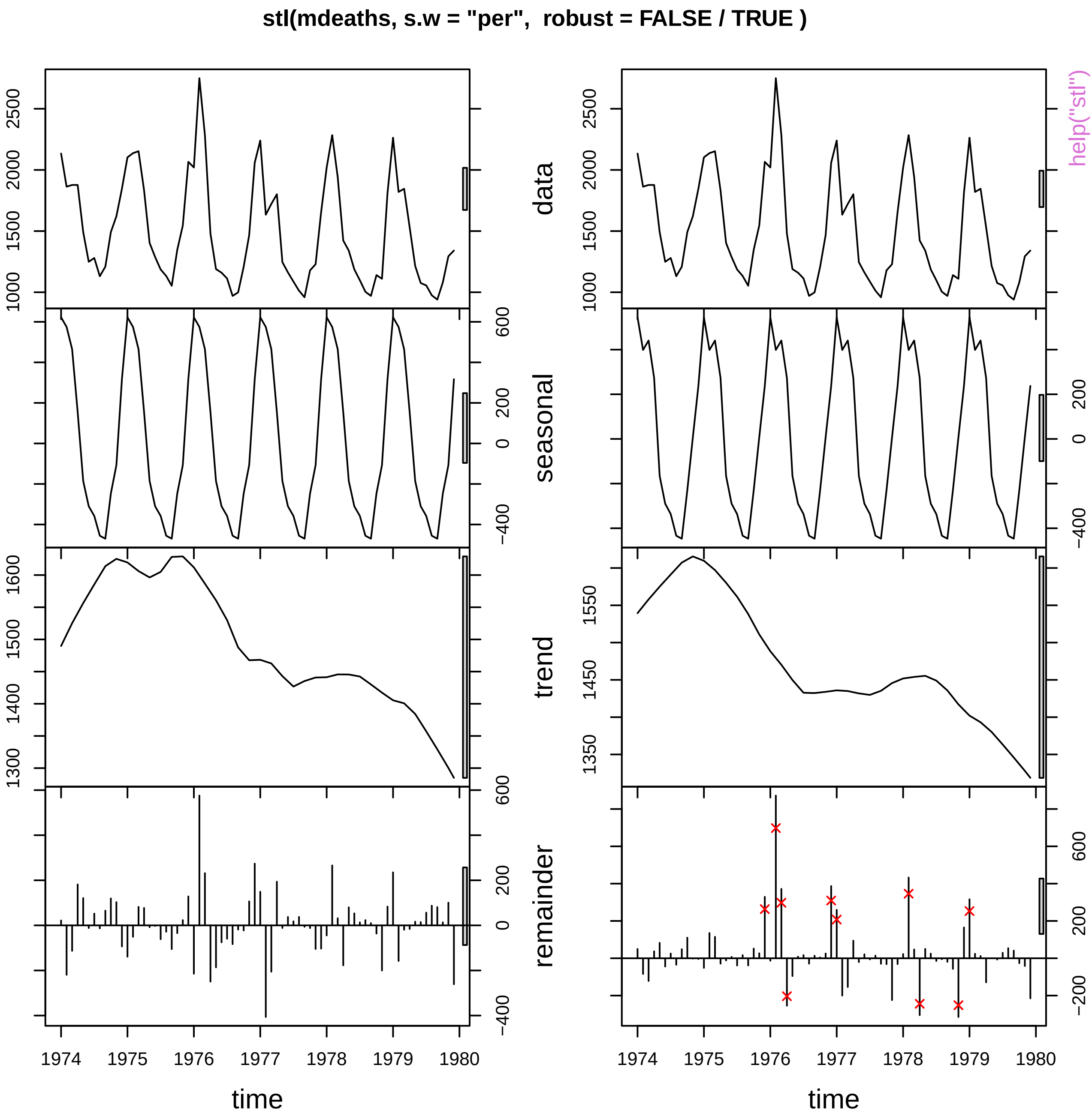
data
seasonal
trend
remainder
time
help("stl")
30
40
50
60
15
10
5
0
−10
48.0
49.0
50.0
6
4
2
0
−2
−4
1920
1925
1930
1935
1940

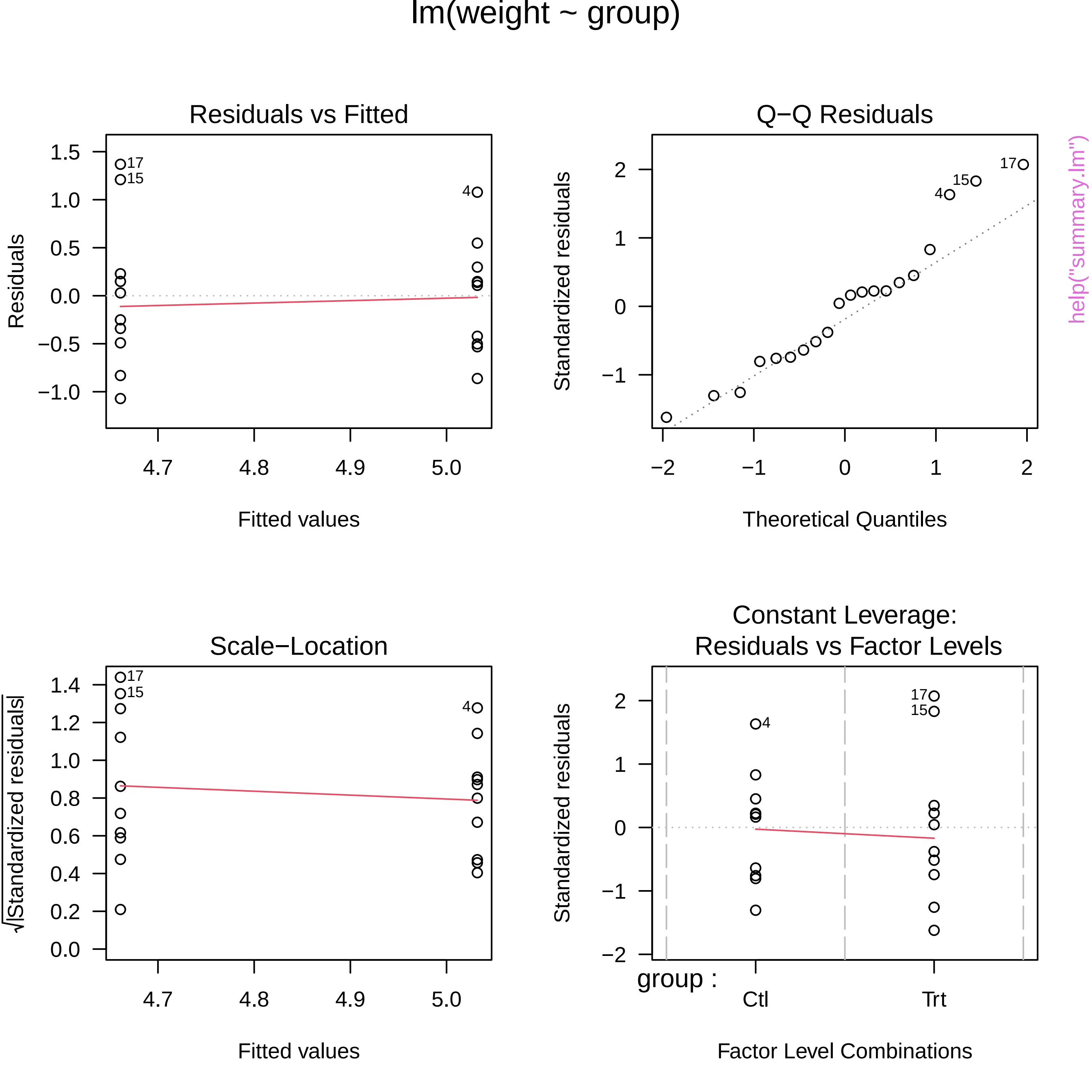


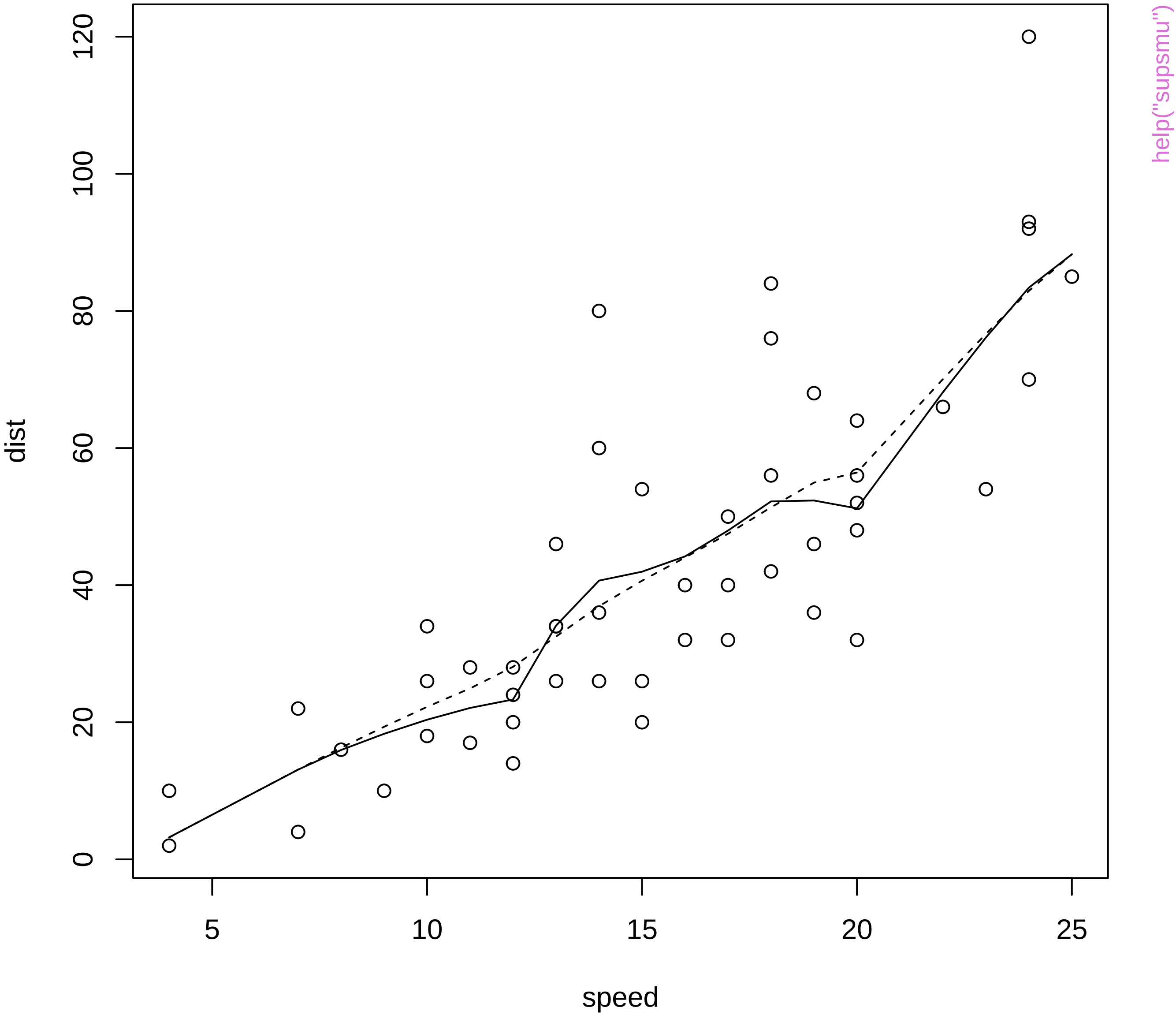
data
seasonal
trend
remainder
5.75
5.80
5.85
5.90
0.000
−0.010
0.002
−0.002
help("stl")
1960
1970
1980
1990
time



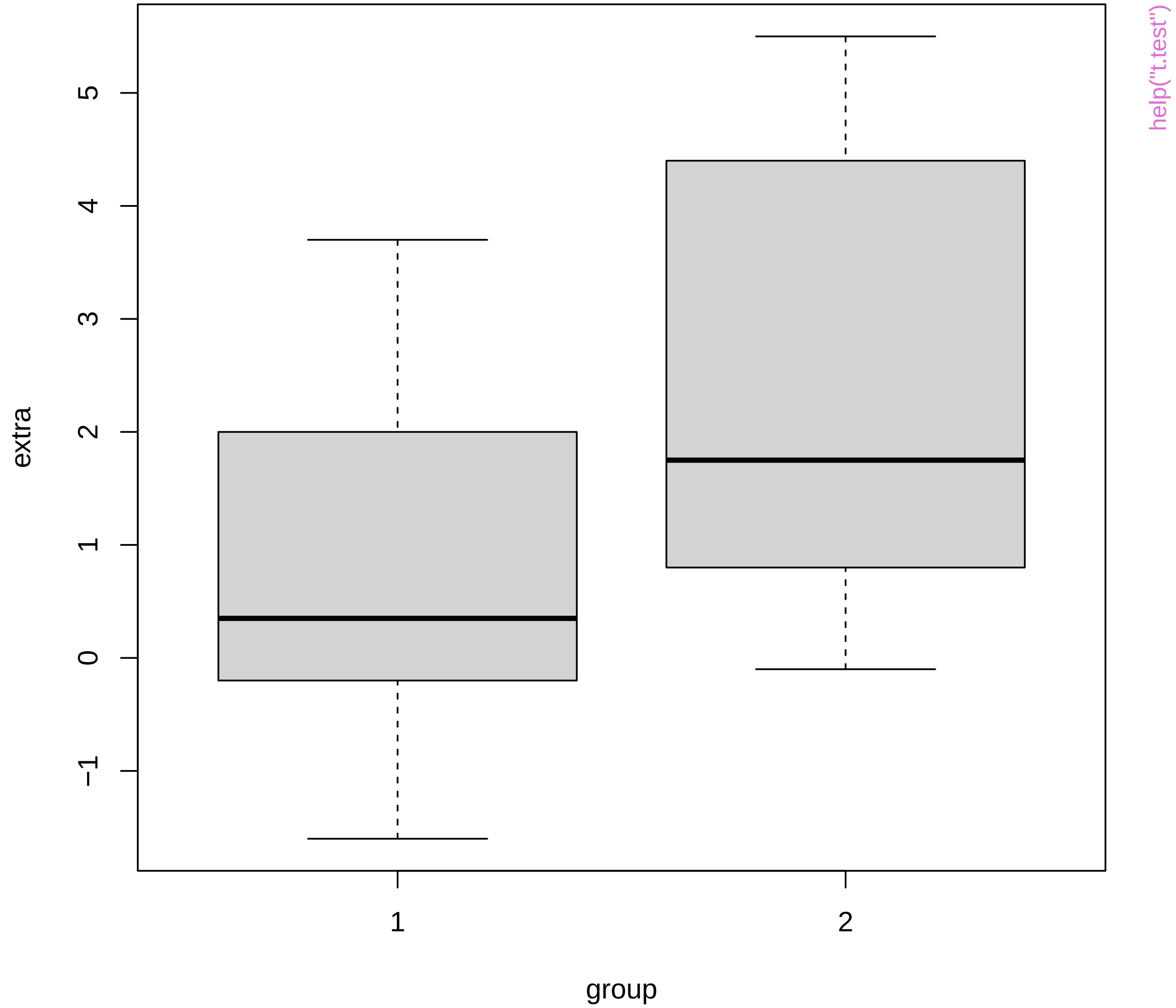
data
seasonal
trend
remainder
time
help("stl")
5.75
5.80
5.85
5.90
0.000
−0.010
0.005
−0.005
1960
1970
1980
1990


stl(mdeaths, s.w = "per", robust = FALSE / TRUE)
data
seasonal
trend
remainder
time
time
help("stl")
1974
1975
1976
1977
1978
1979
1980
1000
1500
2000
2500
-400
0
200
600
1300
1400
1500
1600
1350
1450
1550
-200

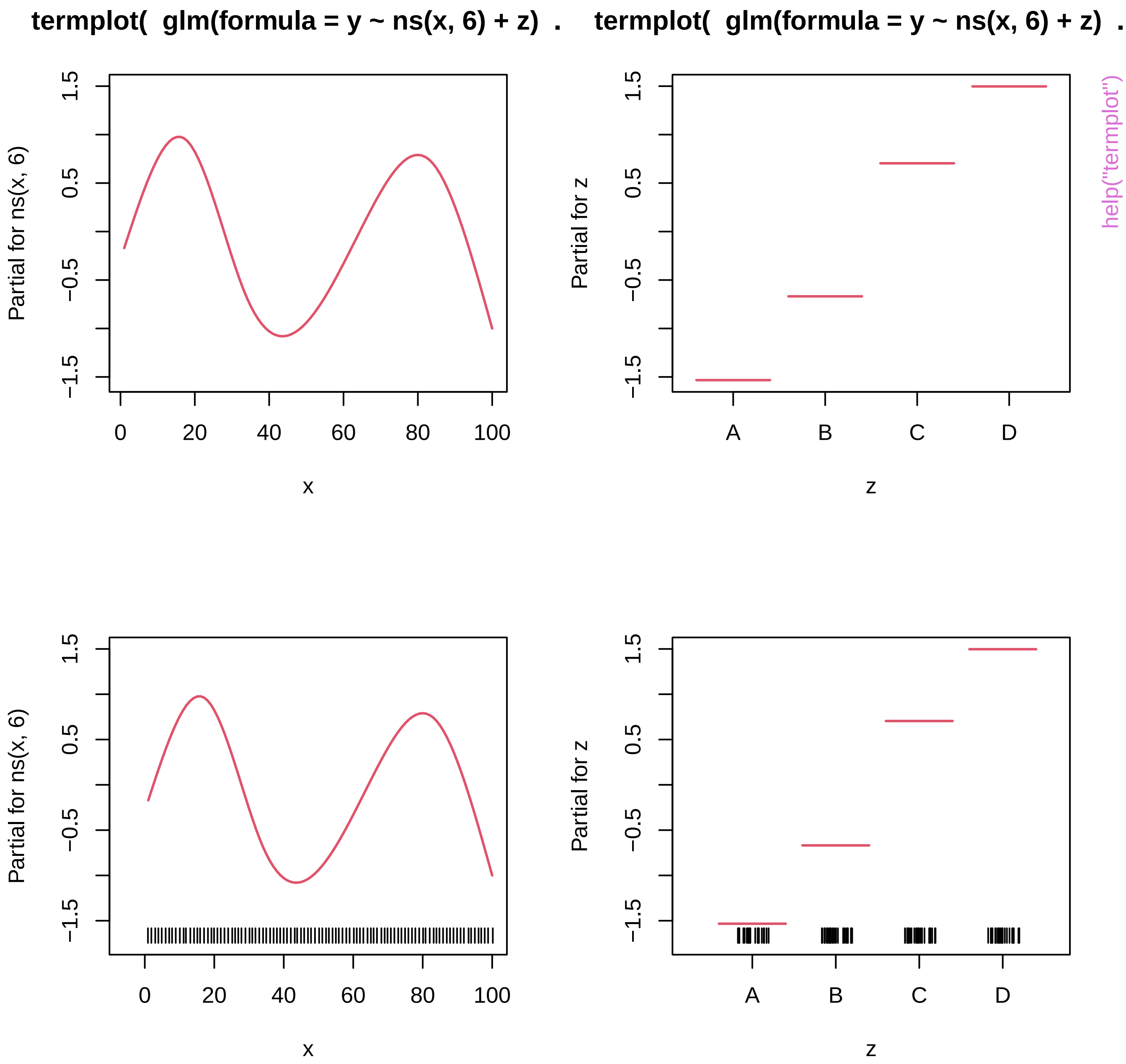

lm(weight ~ group)
Residuals vs Fitted
Residuals
Fitted values
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
help("summary.lm")
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
group :
Ctl
Trt
Factor Level Combinations

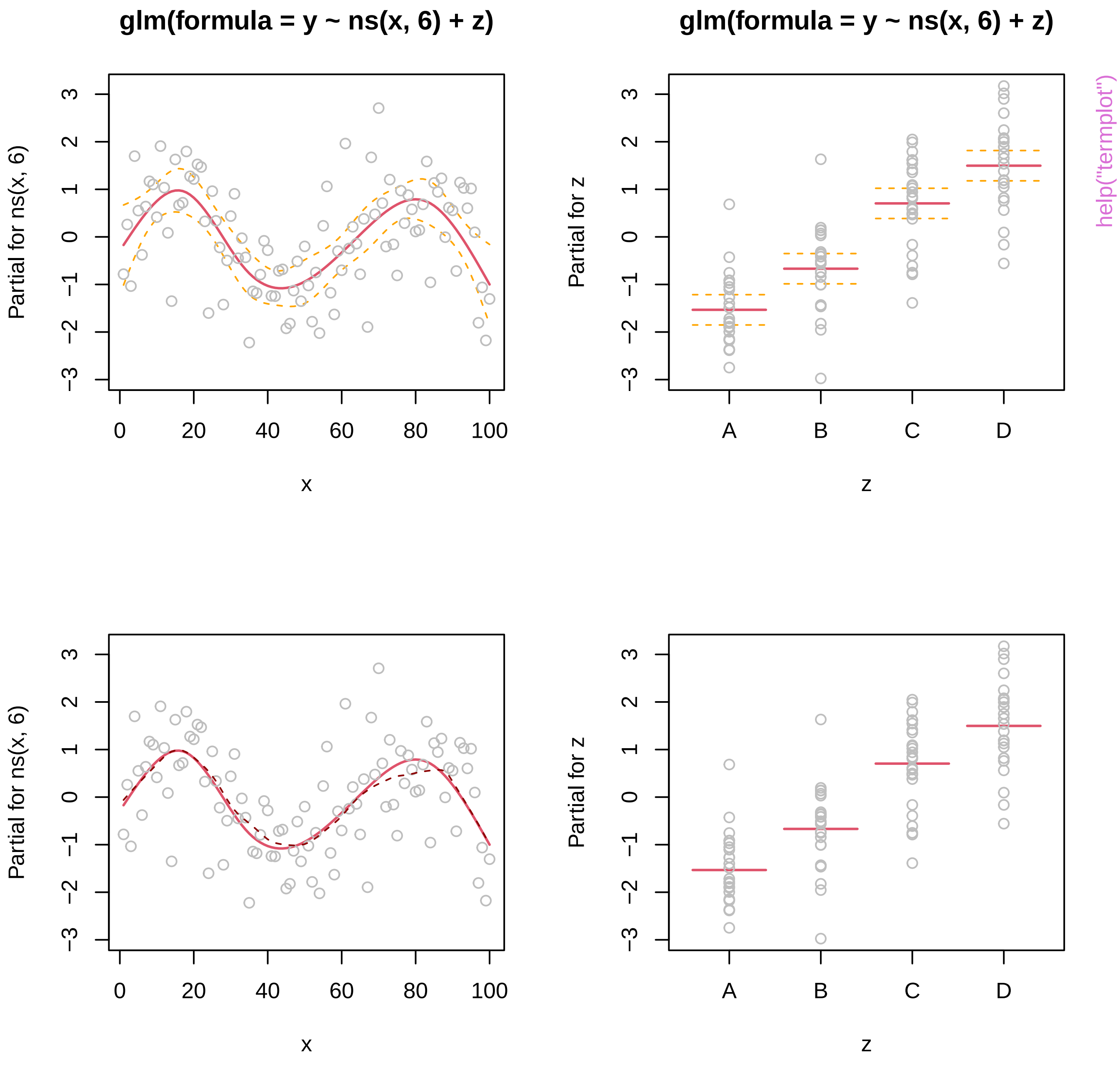


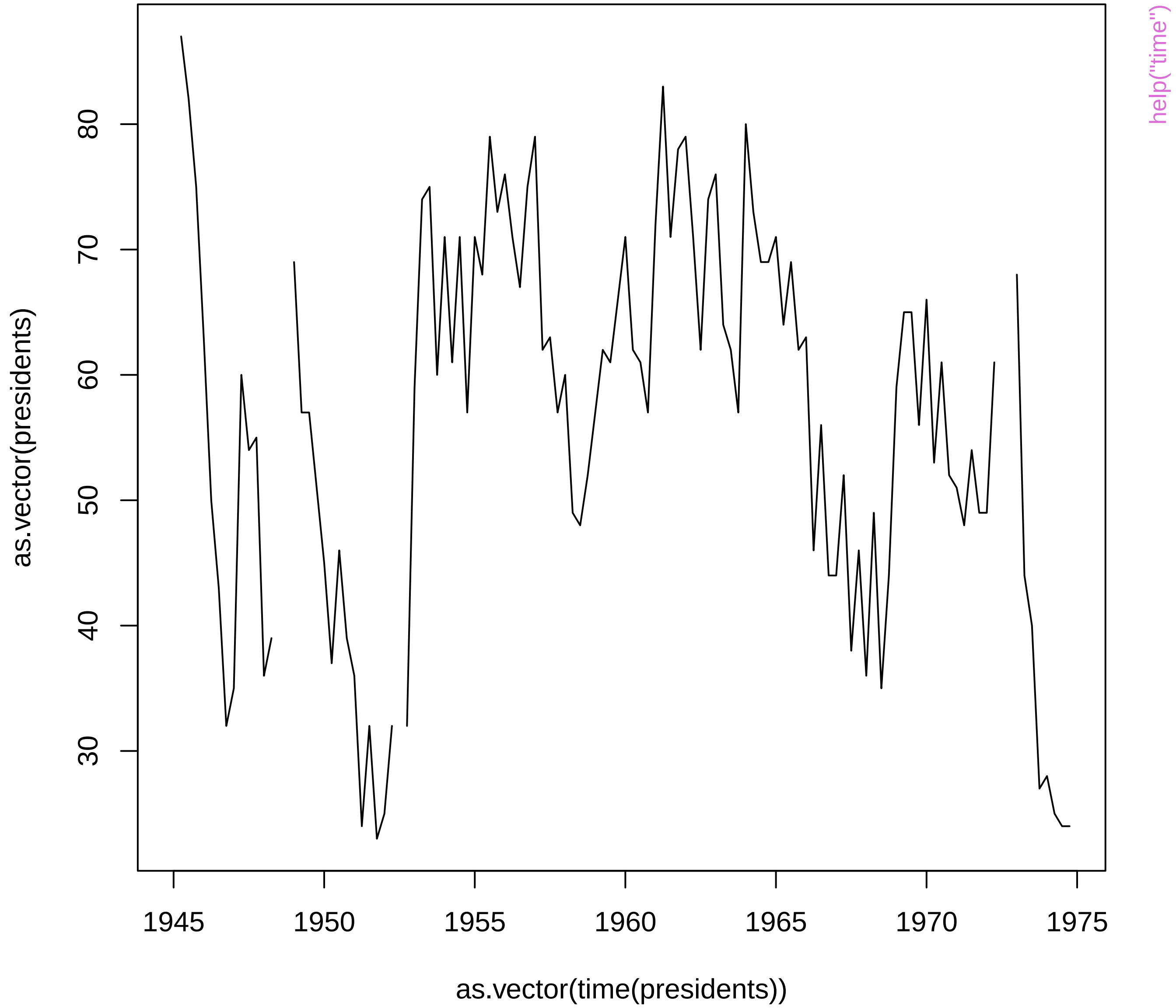
0
20
40
60
80
100
120
dist
5
10
15
20
25
speed
help("supsmu")



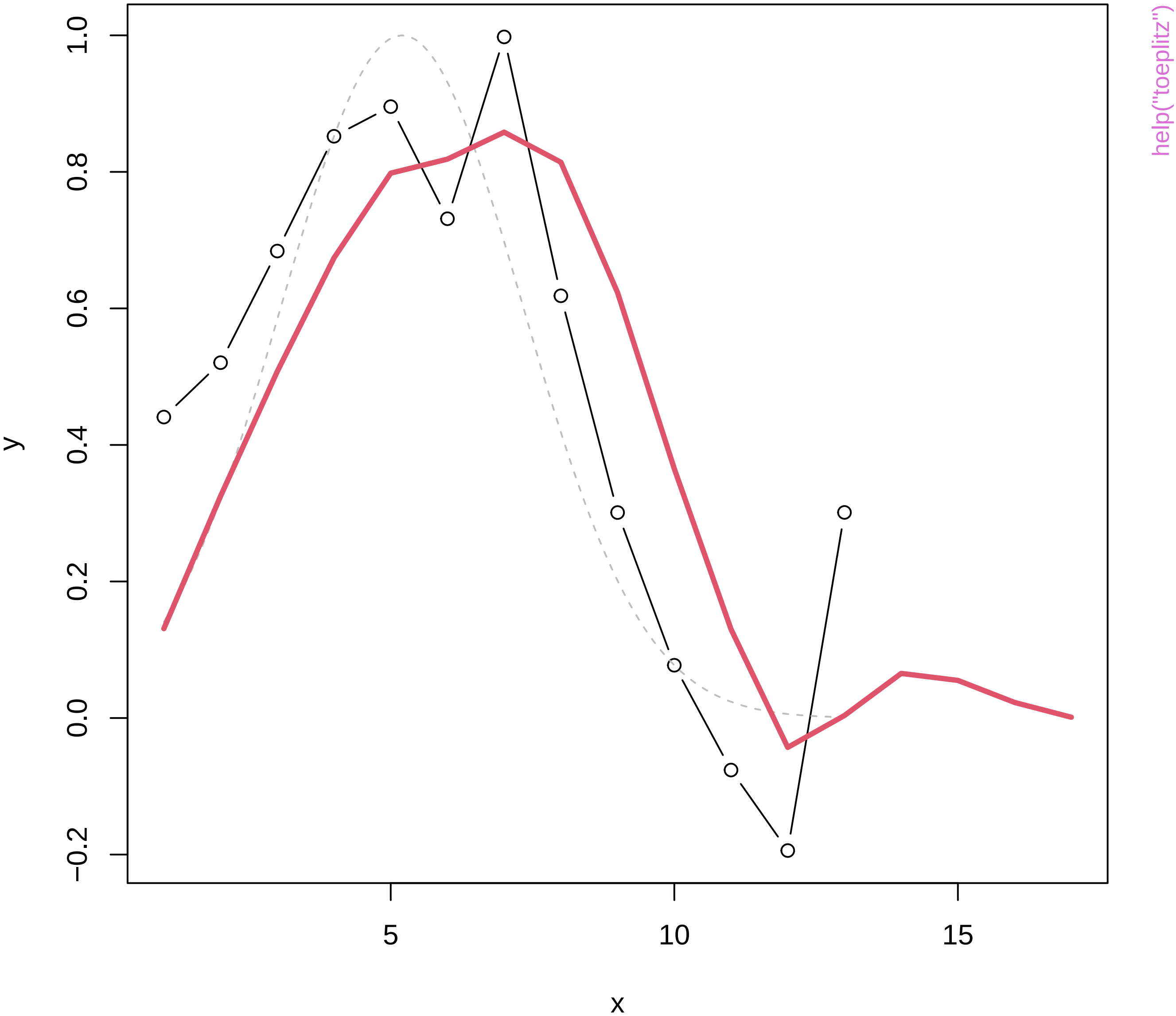
extra
−1
0
1
2
3
4
5
1
2
group
help("t.test")


termplot(glm(formula = y ~ ns(x, 6) + z) .
termplot(glm(formula = y ~ ns(x, 6) + z) .
help("termplot")
Partial for ns(x, 6)
Partial for z
x
z
1.5
0.5
−0.5
−1.5
0
20
40
60
80
100
A
B
C
D

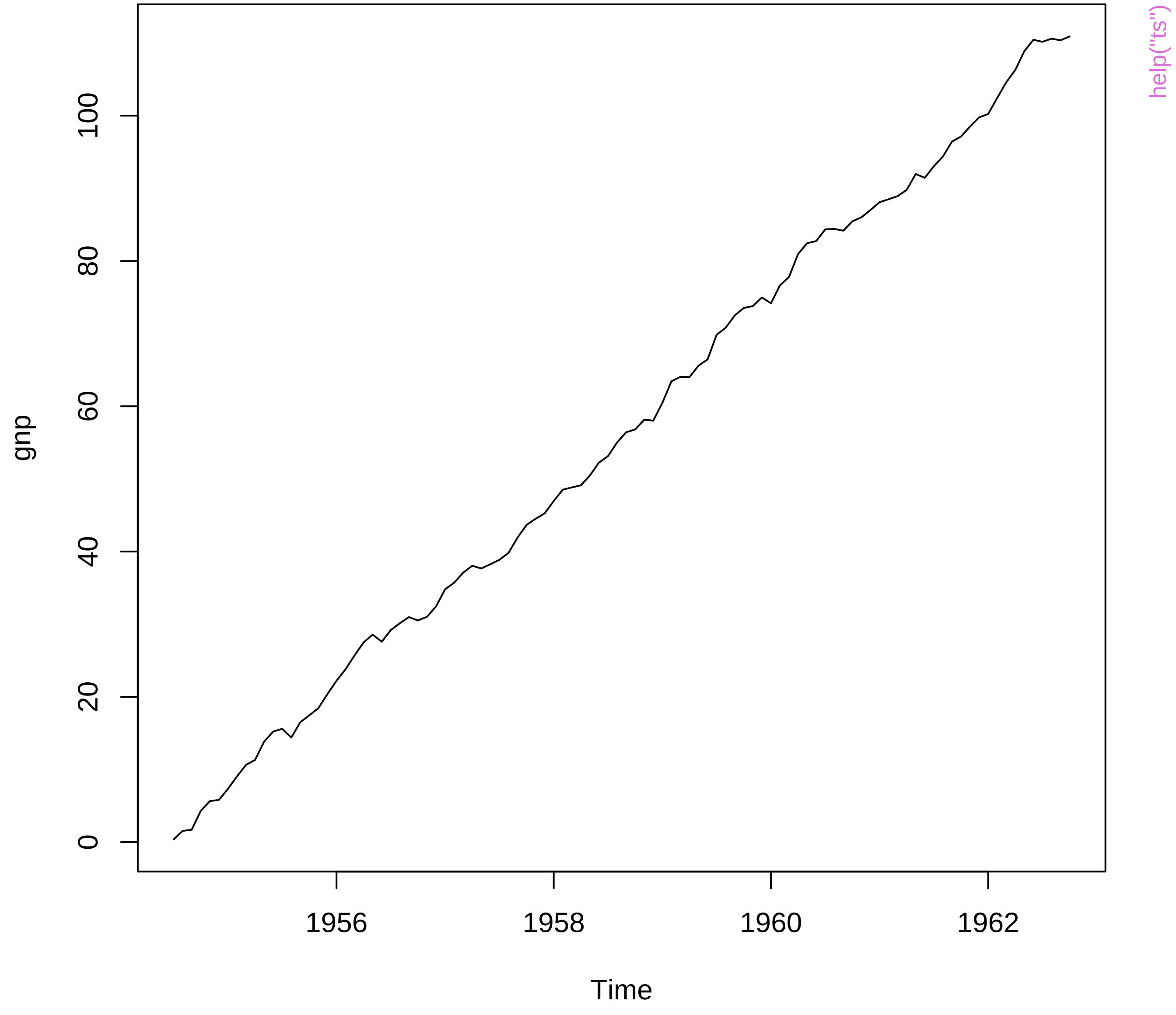

glm(formula = y ~ ns(x, 6) + z)
glm(formula = y ~ ns(x, 6) + z)
Partial for ns(x, 6)
Partial for z
x
z
0
20
40
60
80
100
A
B
C
D
−3
−2
−1
0
1
2
3
help("termplot")



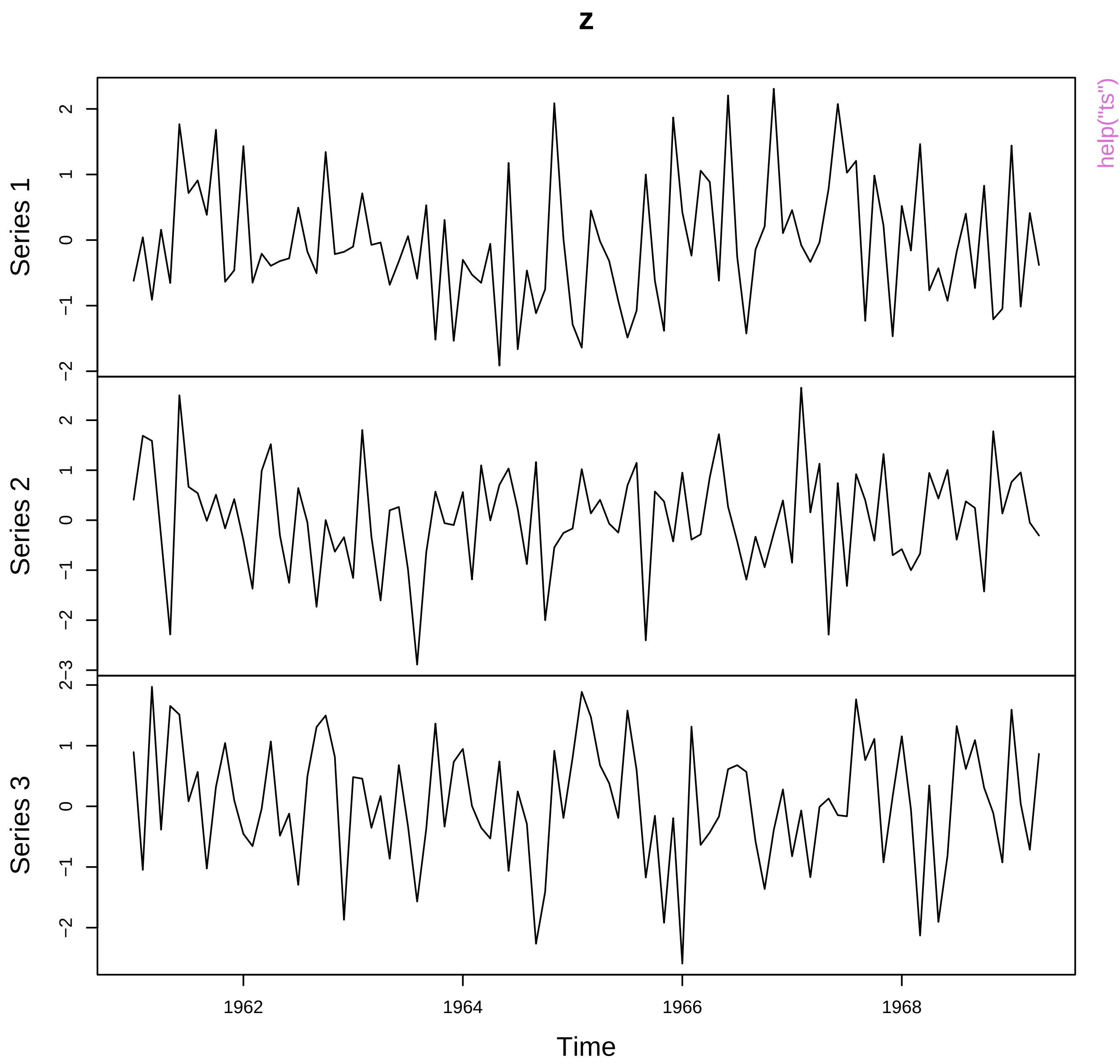
80
70
60
50
40
30
as.vector(presidents)
1945
1950
1955
1960
1965
1970
1975
as.vector(time(presidents))
help("time")



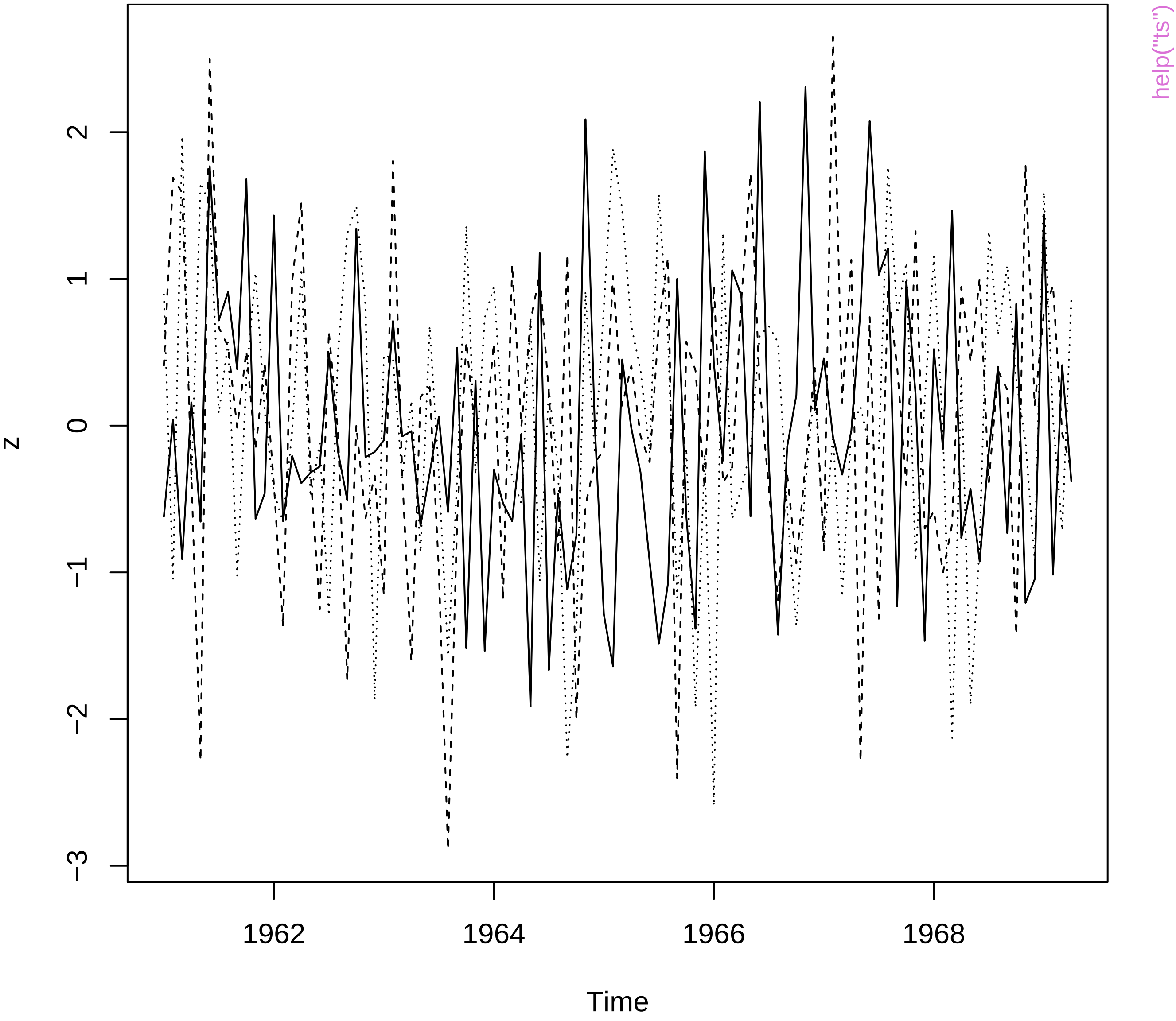
help("toeplitz")
y
x
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
5
10
15



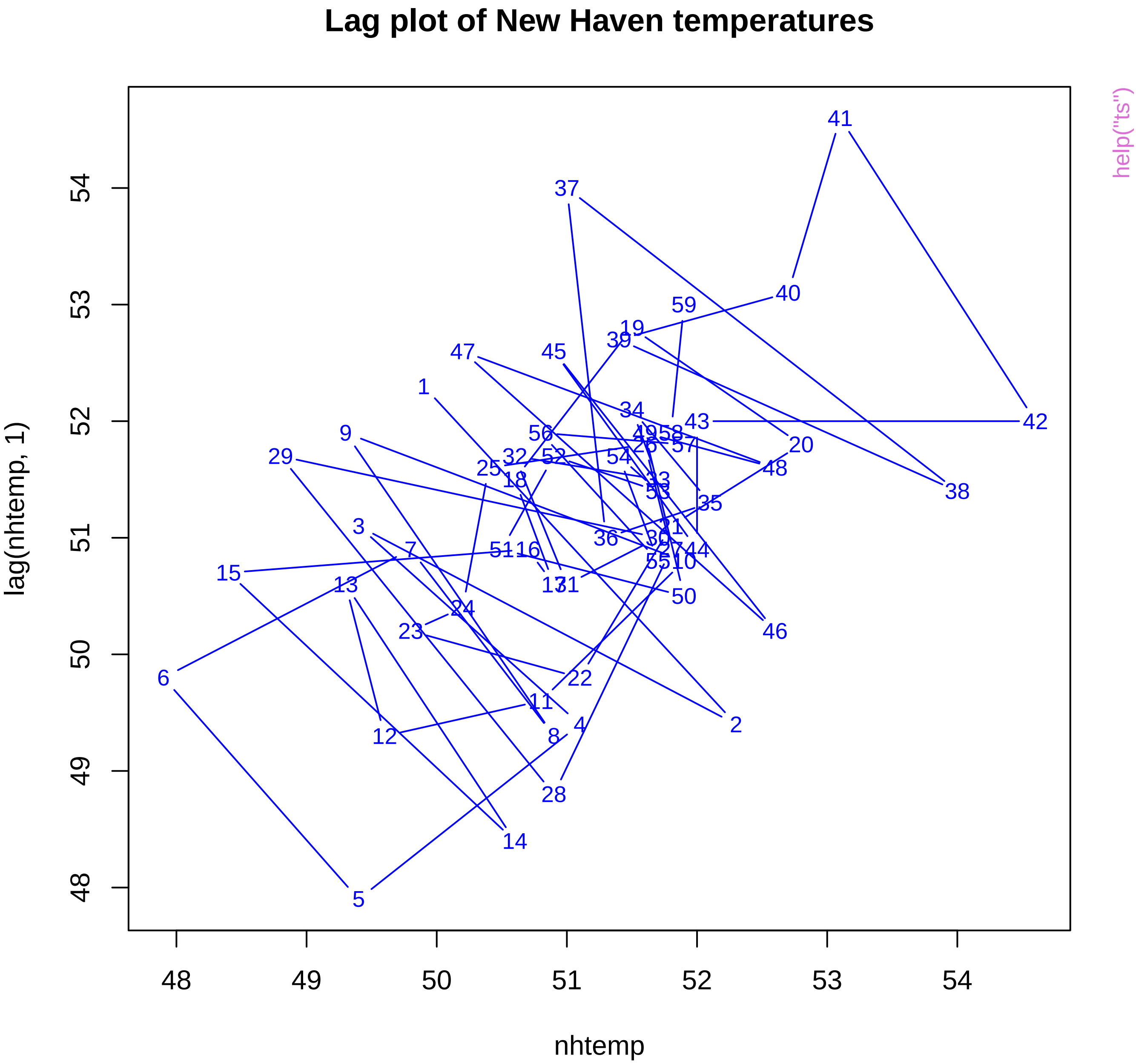
help("ts")
gnp
0
20
40
60
80
100
1956
1958
1960
1962
Time

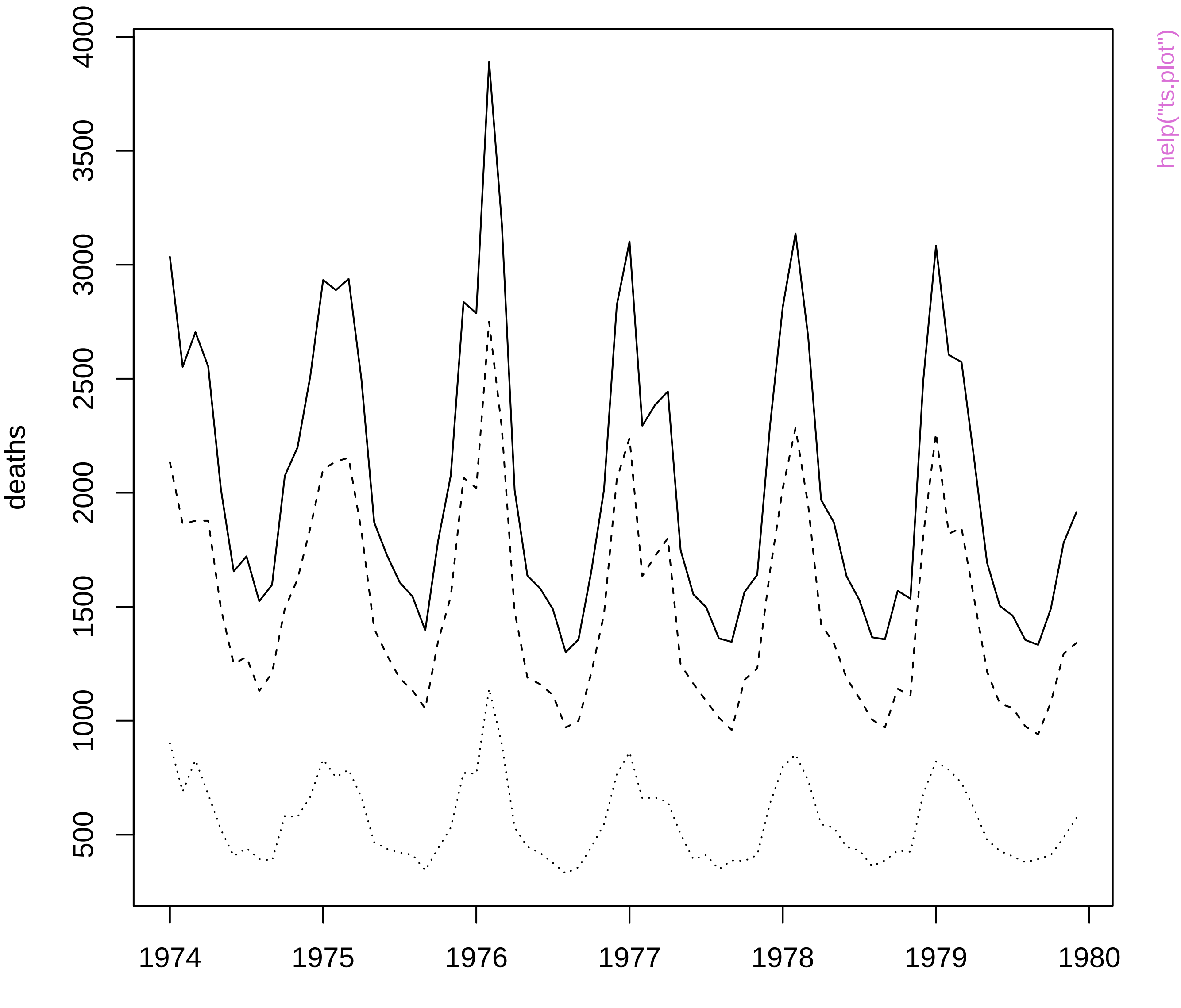


z
Series 1
Series 2
Series 3
2
1
0
−1
−2
2
1
0
−1
−2
−3
2
1
0
−1
−2
1962
1964
1966
1968
Time
help("ts")

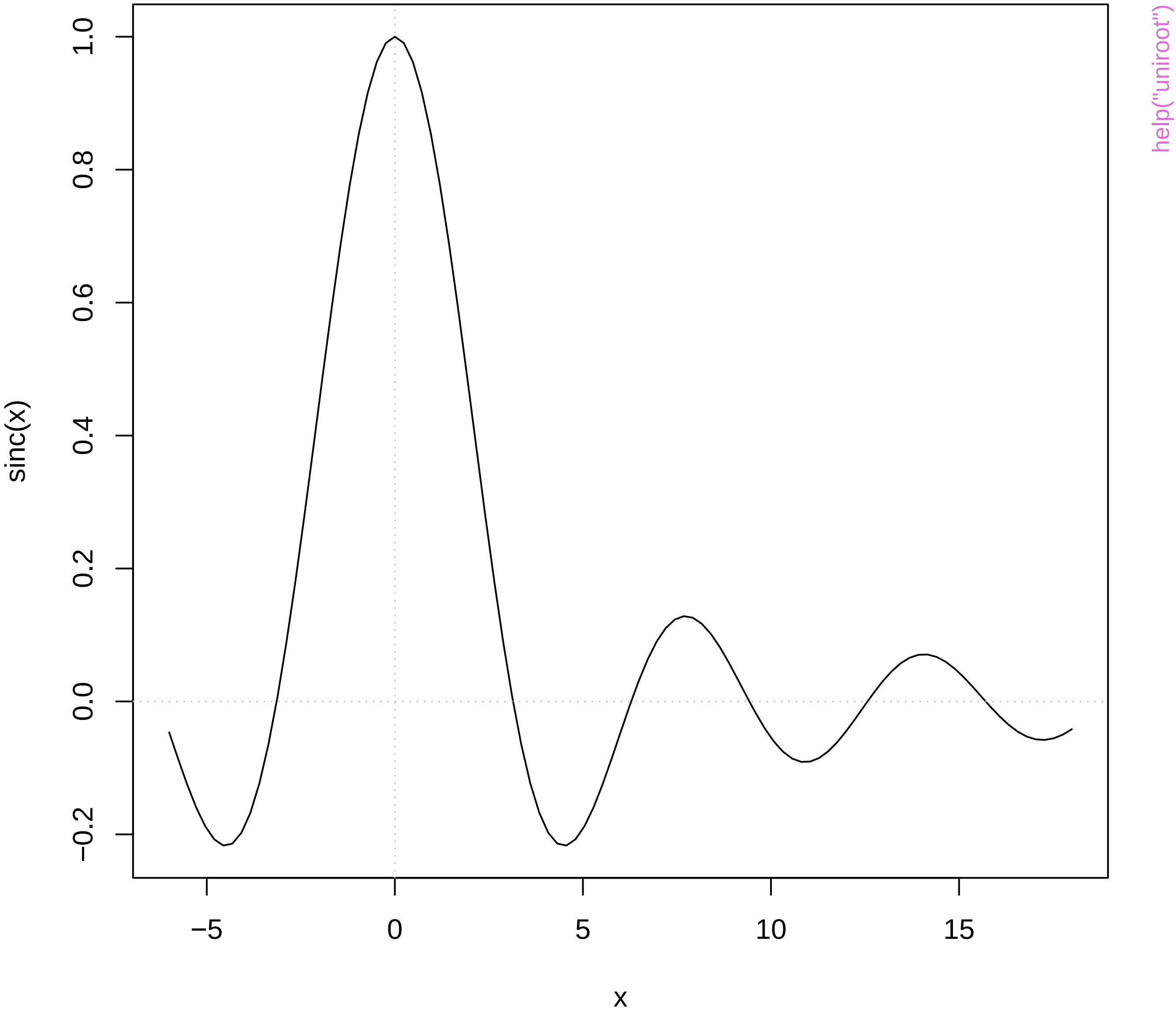


z
2
1
0
−1
−2
−3
1962
1964
1966
1968
Time
help("ts")

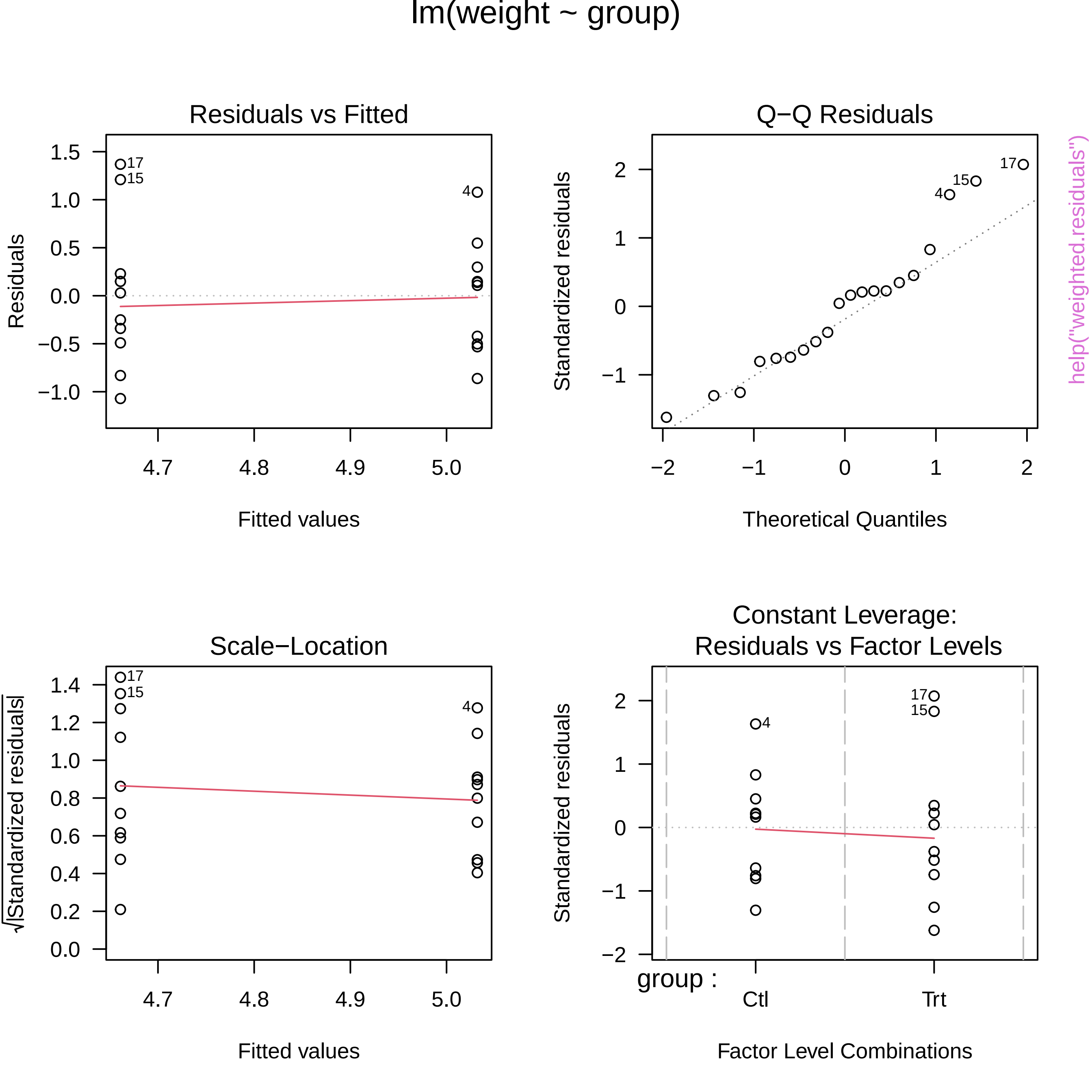

Lag plot of New Haven temperatures
lag(nhtemp, 1)
48
49
50
51
52
53
54
nhtemp
help("ts")



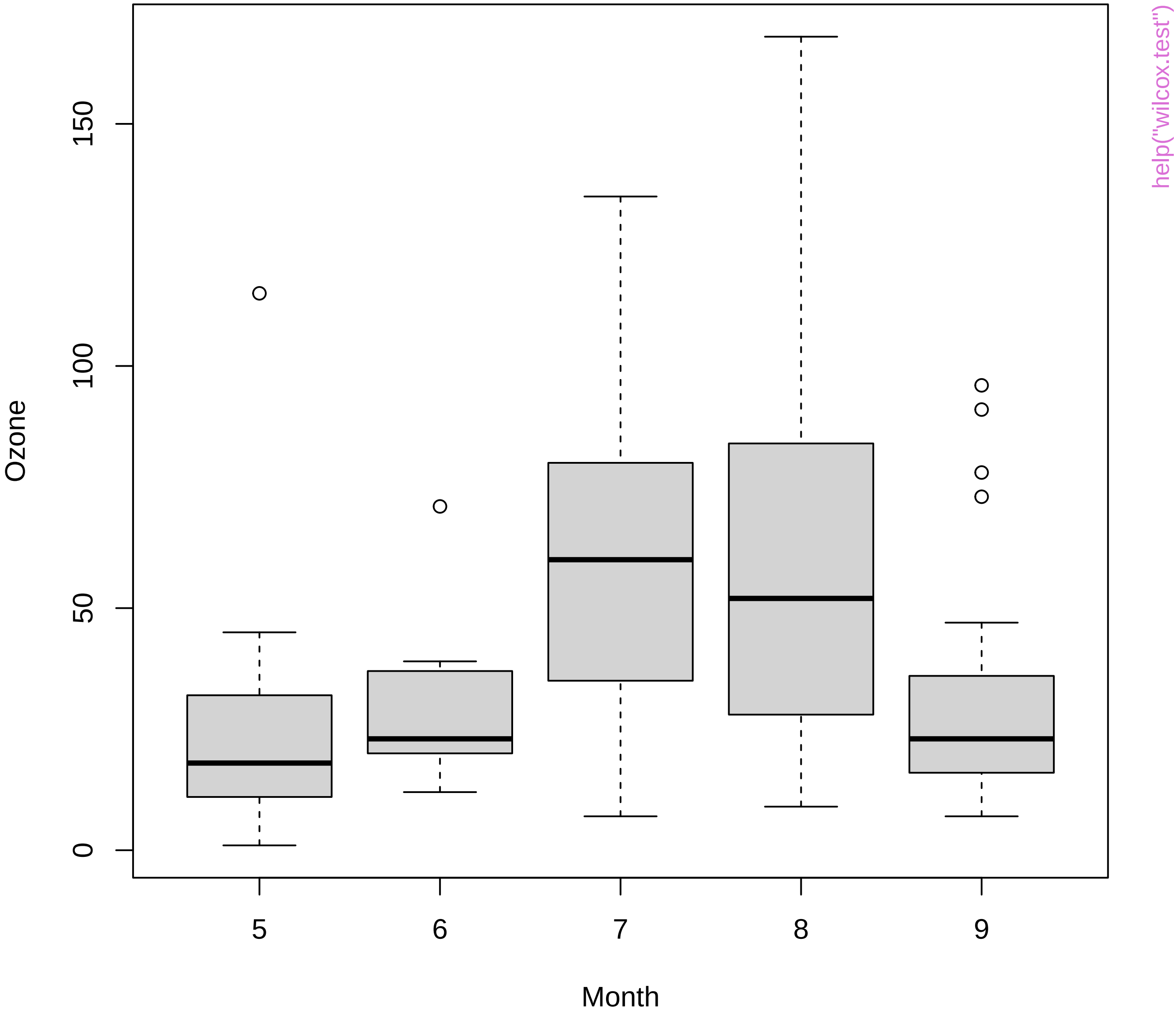
4000
3500
3000
2500
2000
1500
1000
500
deaths
1974
1975
1976
1977
1978
1979
1980
year
help("ts.plot")



1.0
0.8
0.6
0.4
0.2
0.0
−0.2
sinc(x)
−5
0
5
10
15
x
help("uniroot")



lm(weight ~ group)
Residuals vs Fitted
Residuals
Fitted values
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
help("weighted.residuals")
Scale−Location
√|Standardized residuals|
Fitted values
Constant Leverage:
Residuals vs Factor Levels
Standardized residuals
group :
Ctl
Trt
Factor Level Combinations


Ozone
150
100
50
0
5
6
7
8
9
Month
help("wilcox.test")