Supplement to: MacQuarrie CJK and BJ Cooke (2011)

Density-dependant population dynamics of mountain pine beetle in natural and thinned forests. Can J For Res 41:1031-1046.

Chris J K MacQuarrie cjkmacquarrie@gmail.com

July 2009

Contents

This file contains the code necessary to reproduce our analysis and construct figures 2 and 3 (note the publication version of Figures 2 and 3 were touched up a bit in Inkscape prior to publication. Currently, the `lattice` package does not make it easy to add supplemental text to the panels, so I added the figure identification labels (*a*, *b*, *c*, etc.) and the asterisks (*) to Figure 3 by hand.

The script was built and tested in R version 2.8.0 and requires additional packages available from CRAN. Please note the script attempts to load these packages at start-up and R will produce warnings if the packages are not installed on your system. The script will also attempt to load an accessory file of utility functions. Again, if these are not found running the script will produce errors.

If you find our functions useful, please feel free to use them but include a reference to our paper.

This script will attempt to load the two datasets that accompanied this file, if you neglected to download these files, please do so now. Permission is granted to use these data in other applications as long as the data are attributed to the original authors/collectors. See the meta-data for the two datasets for this information.

This script was written at the Northern Forestry Centre of Natural Resources Canada Canadian Forest Service in Edmonton, Alberta, Canada.

Original file produced using `Notepad++` ver. 4.8.2 (http://notepad-plus.sourceforge.net) with syntax highlighting via `NppToR` (http://sourceforge.net/projects/npptor/).

Prologue

Establish the analysis environment

ATTENTION! Please set the location of the `helperfunctions.R` and data files before continuing

```
cwd <- file.path("~/GitHub/MPB/growth-curves") # e.g. "c:\\mystuff\\my_R\\cooldata\\"

infunctions <- file.path(cwd, "helperfunctions.R")
indatafile1 <- file.path(cwd, "StudyData.csv")
indatafile2 <- file.path(cwd, "USDAFSData.csv")

# load the neccesary packages and functions
library(nlme)
library(lattice)
library(grid)

source(infunctions)

# read in the data extracted from the 8 pine beetle tree mortality studies
treemort <- read.csv(indatafile1, header = TRUE, fill = TRUE)

# read in the data extracted from the US Forest Service pine mortality surveys
usforests <- read.csv(indatafile2, header = TRUE, fill = TRUE)
```

Part 1

Model the natural and thinned forest data from the 8 pine beetle tree mortality studies

```
# declare a useful vector
treats <- c("control", "monitor", "limit", "basal")

# Calculate the reproduction rate (r) for all records in the treemort data
attach(treemort)

reprod.rate <- numeric()

for (i in 1:length(yearabs)) {
  # if two adjacent rows in the data frame are from the same study and are from consequtive years, calc
  # if not, set r = 0
  x <- ifelse(Study[i + 1] == Study[i] & yearabs[i + 1] == (yearabs[i] + 1),
              per_ha_dead[i + 1] / per_ha_dead[i], # if true, set x = r
              0)                                   # if not, set x = 0

  reprod.rate <- append(reprod.rate, x, i)
}

detach(treemort)
```

```
# add the vector of r values to the treemort data frame and clean up
treemort <- cbind(treemort, reprod.rate)
```

```r
rm(reprod.rate, x, i)

# calculate the natural log of the reproduction rate (R) and population size (X)
treemort$lnr <- log(treemort$reprod.rate)
treemort$lnxt <- log(treemort$per_ha_dead)

# remove any 0 values
treemort.no.r0 <- treemort[which( treemort$reprod.rate > 0 & treemort$per_ha_dead > 0 ), ]

# partition the data by treatment into four datasets.

cntrl <- treemort.no.r0[which(treemort.no.r0$Treat == "control"), ]
limit <- treemort.no.r0[which(treemort.no.r0$Treat == "limit"), ]
basal <- treemort.no.r0[which(treemort.no.r0$Treat == "basal"), ]
monitor <- treemort.no.r0[which(treemort.no.r0$Treat == "monitor"), ]

# fit the three parameter model to each data set.
# extract the parameter values and show them in the console
fm1control.nls <- nls(lnr ~ a - b * exp( c * lnxt), data = cntrl,
                      start = c(a = 1, b = 1, c = 1))

fm1limit.nls <- nls(lnr ~ a - b * exp( c * lnxt ), data = limit,
                    start = c(a = 1, b = 1, c = 1))

fm1basal.nls <- nls(lnr ~ a - b * exp( c * lnxt ), data = basal,
                    start = c(a = 0.5, b = 0.5, c = 0.5))

fm1monitor.nls <- nls( lnr ~ a - b * exp(c * lnxt), data = monitor,
                       start = c( a = 1, b = 1, c = 1))

# extract the three parameter values from each fitted model and show in console

avalues <- rbind(summary(fm1control.nls)$coef[1, 1:2],
                 summary(fm1monitor.nls)$coef[1, 1:2],
                 summary(fm1limit.nls)$coef[1, 1:2],
                 summary(fm1basal.nls)$coef[1, 1:2])

bvalues <- rbind(summary(fm1control.nls)$coef[2, 1:2],
                 summary(fm1monitor.nls)$coef[2, 1:2],
                 summary(fm1limit.nls)$coef[2, 1:2],
                 summary(fm1basal.nls)$coef[2, 1:2])

cvalues <- rbind(summary(fm1control.nls)$coef[3, 1:2], summary(fm1monitor.nls)$coef[3, 1:2],
                 summary(fm1limit.nls)$coef[3, 1:2], summary(fm1basal.nls)$coef[3, 1:2])

nlssummary.table <- as.data.frame(cbind(avalues, bvalues, cvalues), row.names = treats)

dimnames(nlssummary.table)[[2]] <- c("a", "a_std", "b", "b_std", "c", "c_std")

rm(avalues, bvalues, cvalues)

nlssummary.table
```

```
##                  a     a_std          b      b_std          c     c_std
```

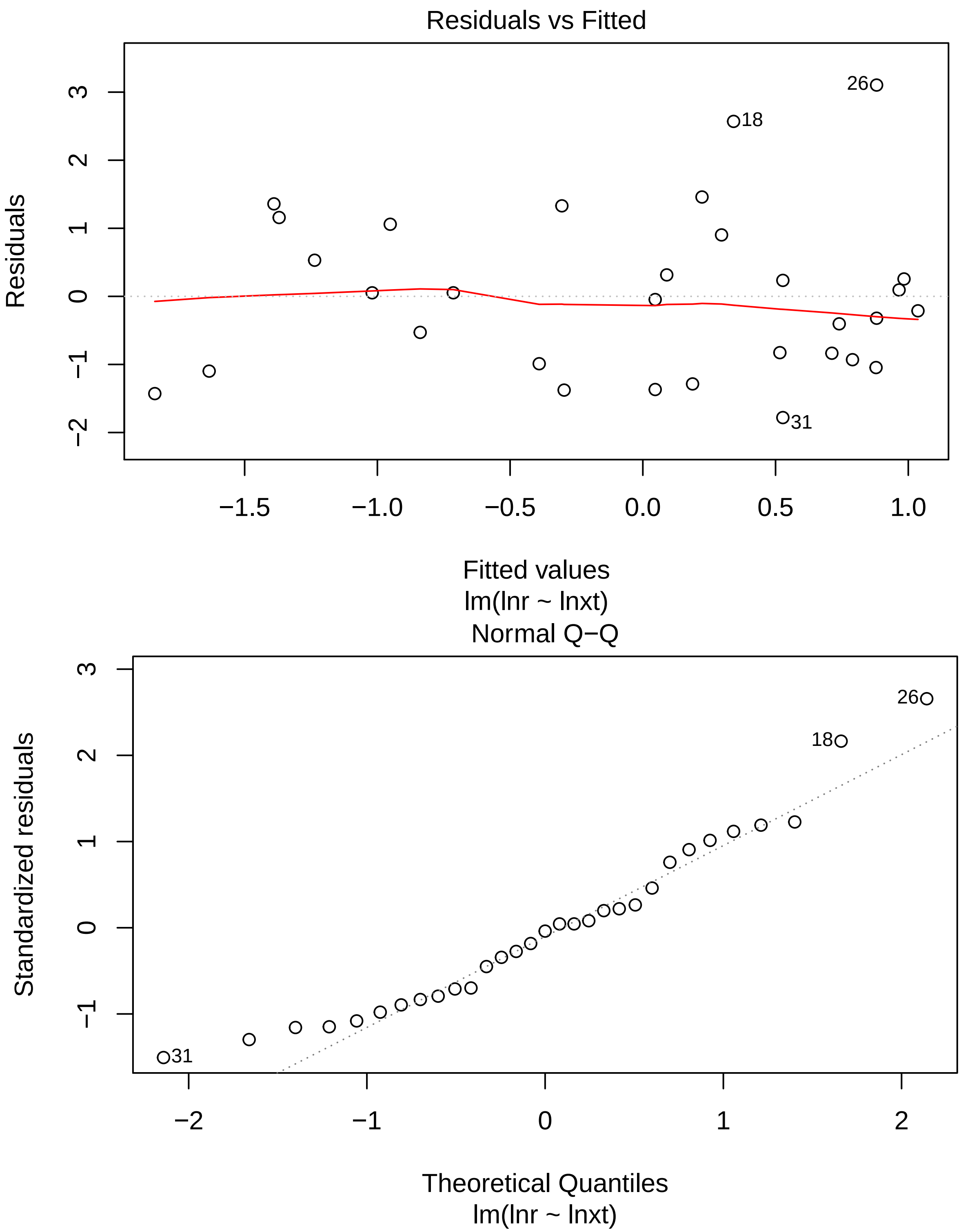
```
## control 1.8043681 2.8060302 0.552176964 1.84095656 0.3134522 0.4388646
## monitor 0.7768605 0.5056261 0.173752395 0.25527045 0.4911593 0.2465532
## limit   0.3319765 0.3337334 0.028747829 0.05290710 1.1037189 0.3947633
## basal   0.7577094 0.3293860 0.005476768 0.01427722 1.3476168 0.5111339
```

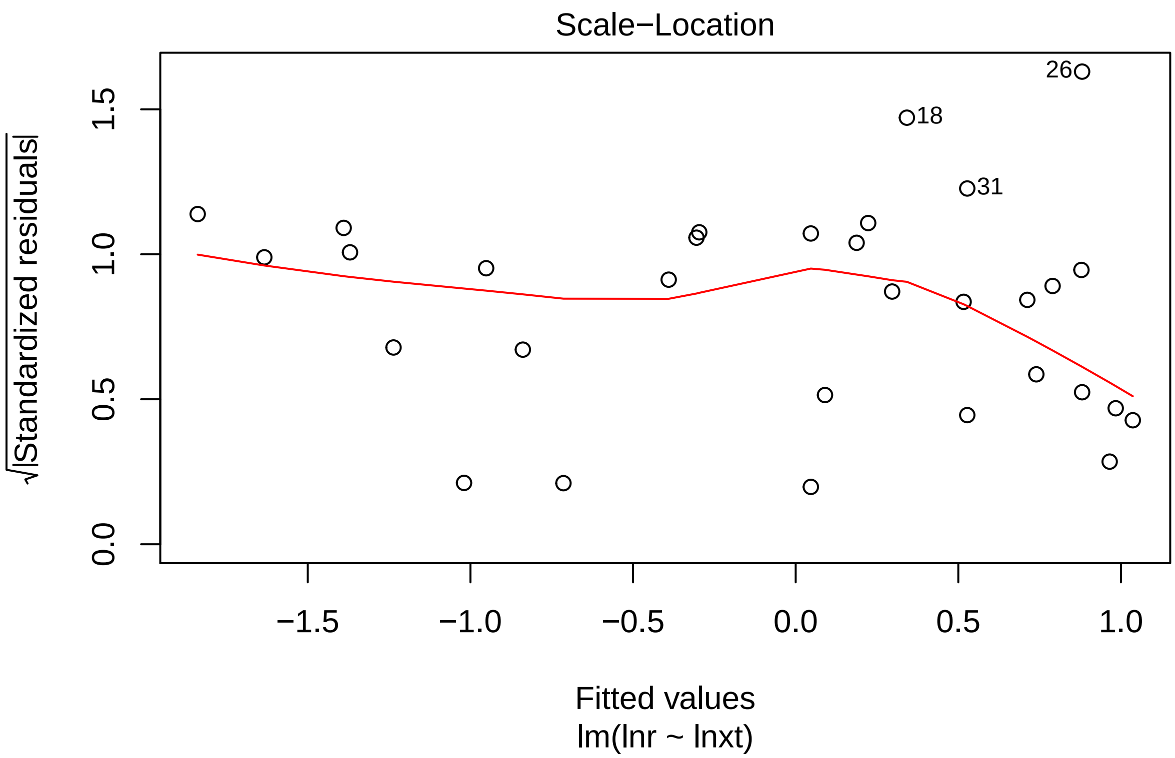
Model the control data

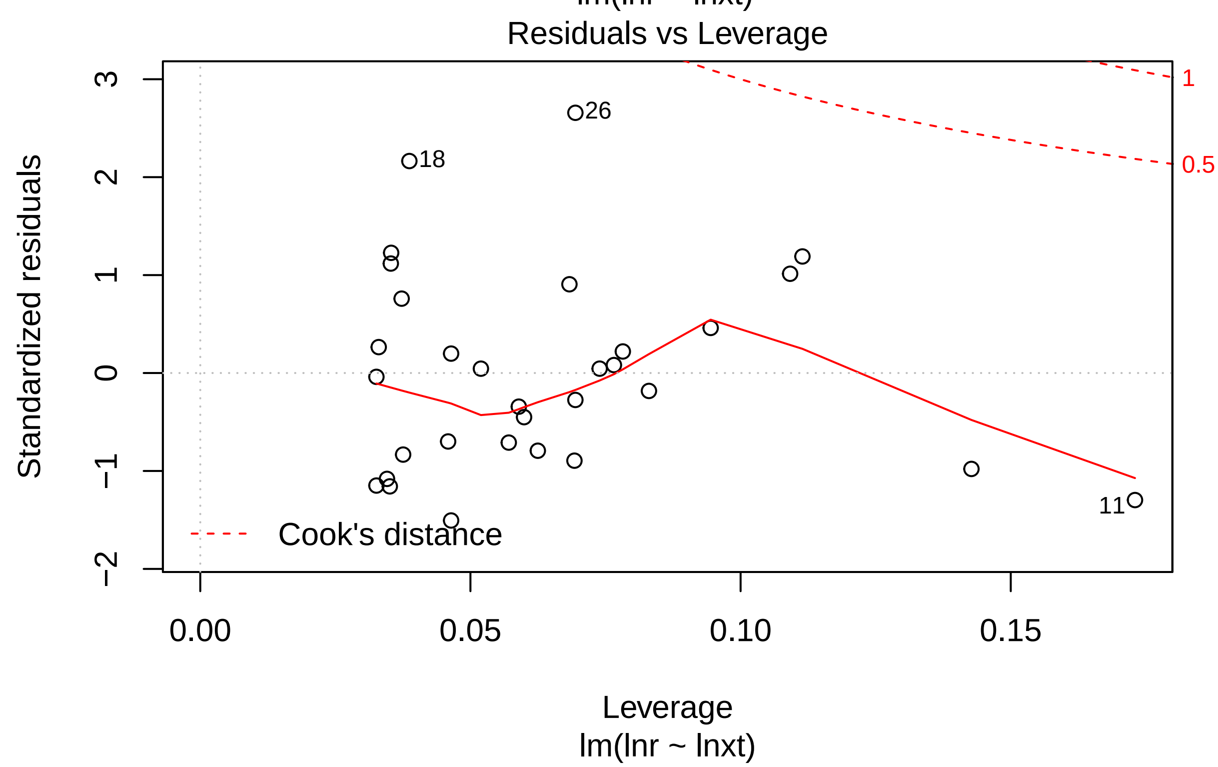
```r
attach(cntrl)

# fit a linear regression model

fm1control.lm <- lm(lnr ~ lnxt)
summary(fm1control.lm) # check the summary
```

```
##
## Call:
## lm(formula = lnr ~ lnxt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.78028 -0.95977 -0.04658  0.71651  3.10368
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1933     0.6058   3.620 0.001109 **
## lnxt         -0.6310     0.1596  -3.954 0.000453 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.211 on 29 degrees of freedom
## Multiple R-squared:  0.3503, Adjusted R-squared:  0.3279
## F-statistic: 15.63 on 1 and 29 DF,  p-value: 0.0004534
```

```r
plot(fm1control.lm) # look at the plots
```


Residuals vs Fitted
Residuals
Fitted values
lm(lnr ~ lnxt)
26
18
31
Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(lnr ~ lnxt)
26
18
31






```r
# is the linear model significant
anova( fm1control.lm ) # yes
```

```
## Analysis of Variance Table
##
## Response: lnr
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)    
## lnxt       1 22.923 22.9231  15.633 0.0004534 ***
## Residuals 29 42.523  1.4663                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

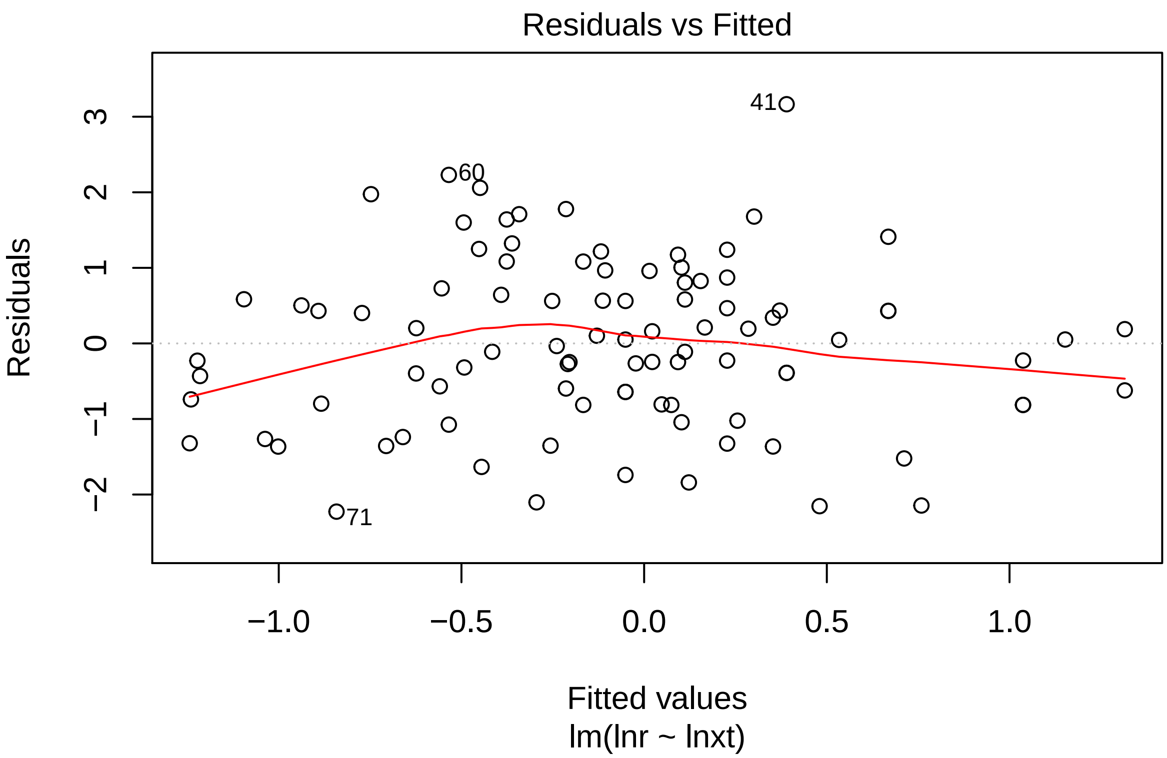
```r
detach(cntrl)
```

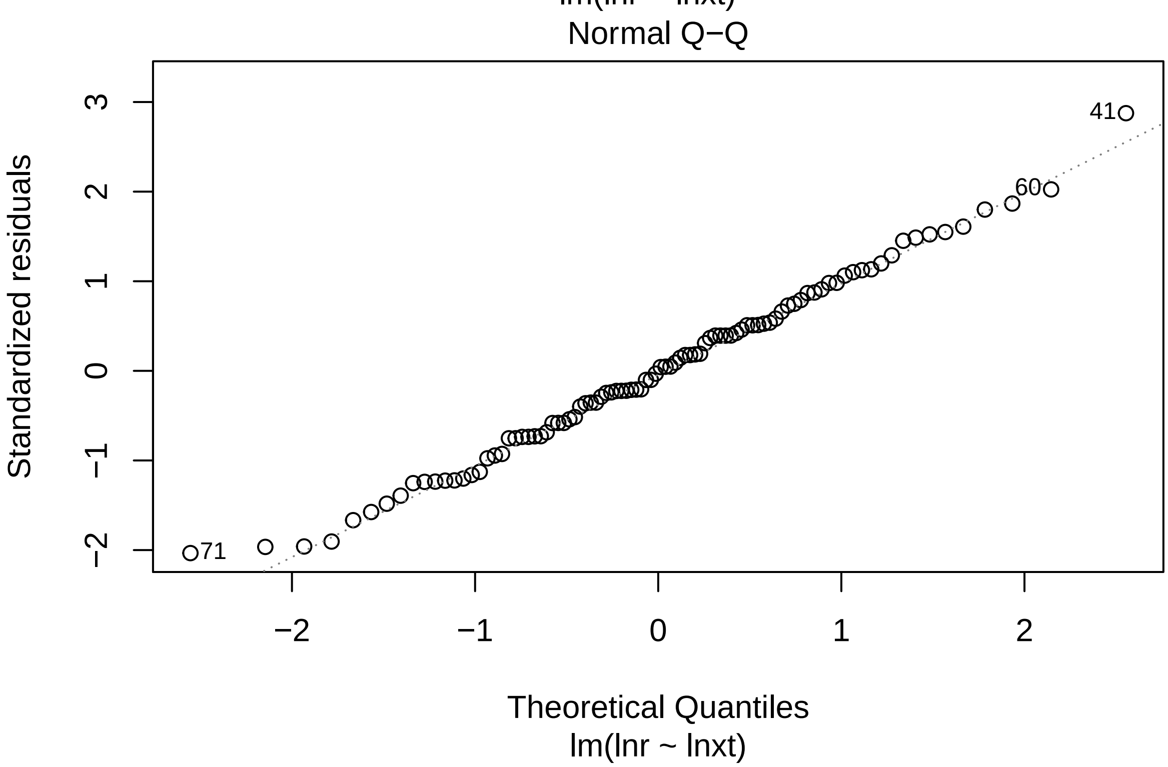
Model the monitoring data

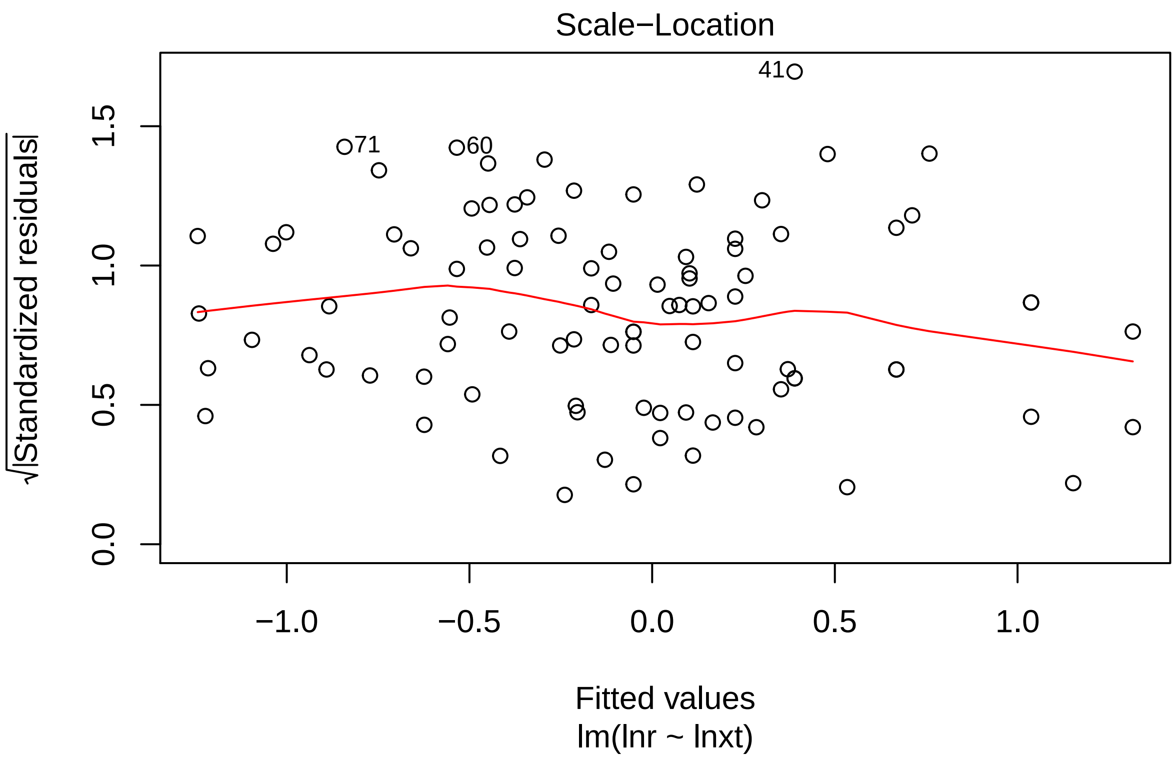
```r
attach(monitor)

# fit a linear regression
fm1monitor.lm <- lm(lnr ~ lnxt)
summary(fm1monitor.lm) # check the summary
```

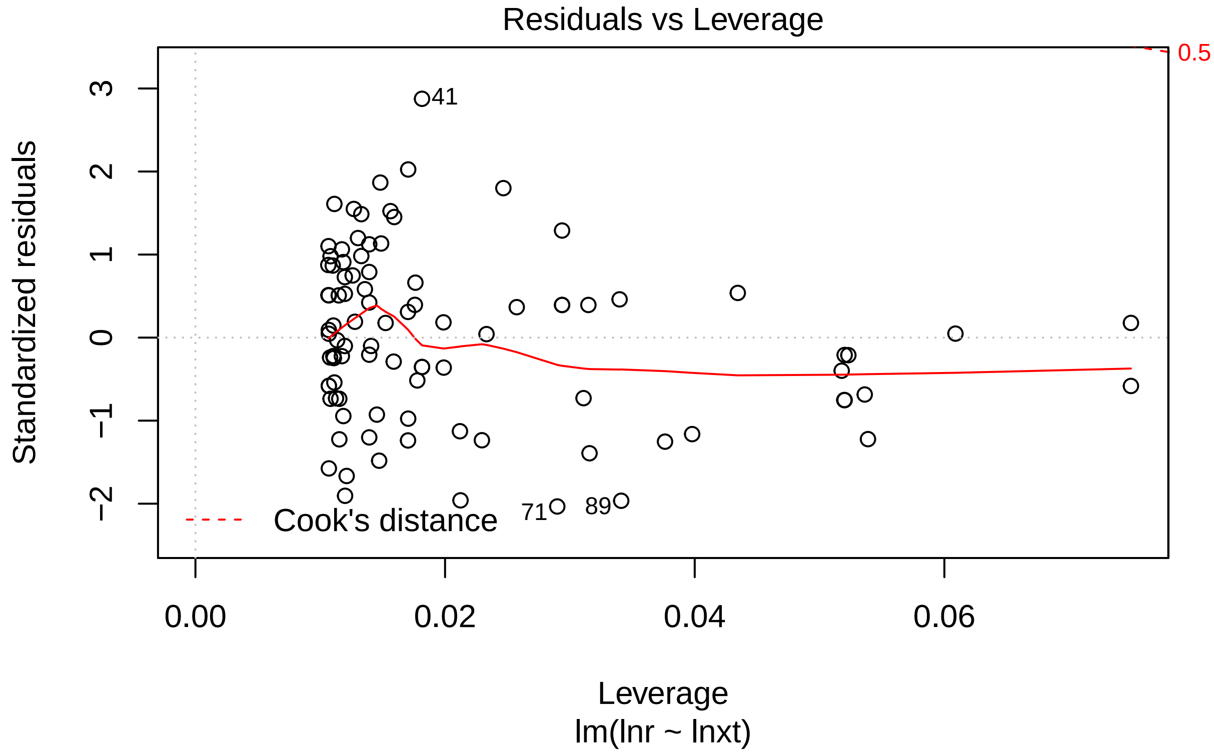
```
## 
## Call:
## lm(formula = lnr ~ lnxt)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -2.2262 -0.8046  0.0057  0.7070  3.1654 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  1.03219    0.25246   4.089 9.28e-05 ***
## lnxt        -0.40151    0.08044  -4.992 2.82e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1.111 on 92 degrees of freedom
## Multiple R-squared:  0.2131, Adjusted R-squared:  0.2046 
## F-statistic: 24.92 on 1 and 92 DF,  p-value: 2.825e-06
```

```r
plot(fm1monitor.lm)
```


Residuals vs Fitted
Residuals
Fitted values
lm(lnr ~ lnxt)
41
60
71
3
2
1
0
−1
−2
−1.0
−0.5
0.0
0.5
1.0


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(lnr ~ lnxt)
41
60
71
3
2
1
0
−1
−2
−2
−1
0
1
2






```
# is the model sig?
anova(fm1monitor.lm) # Yes (marginal)
```

```
## Analysis of Variance Table
##
## Response: lnr
```

```
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## lnxt       1  30.748 30.7484  24.916 2.825e-06 ***
## Residuals 92 113.536  1.2341
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

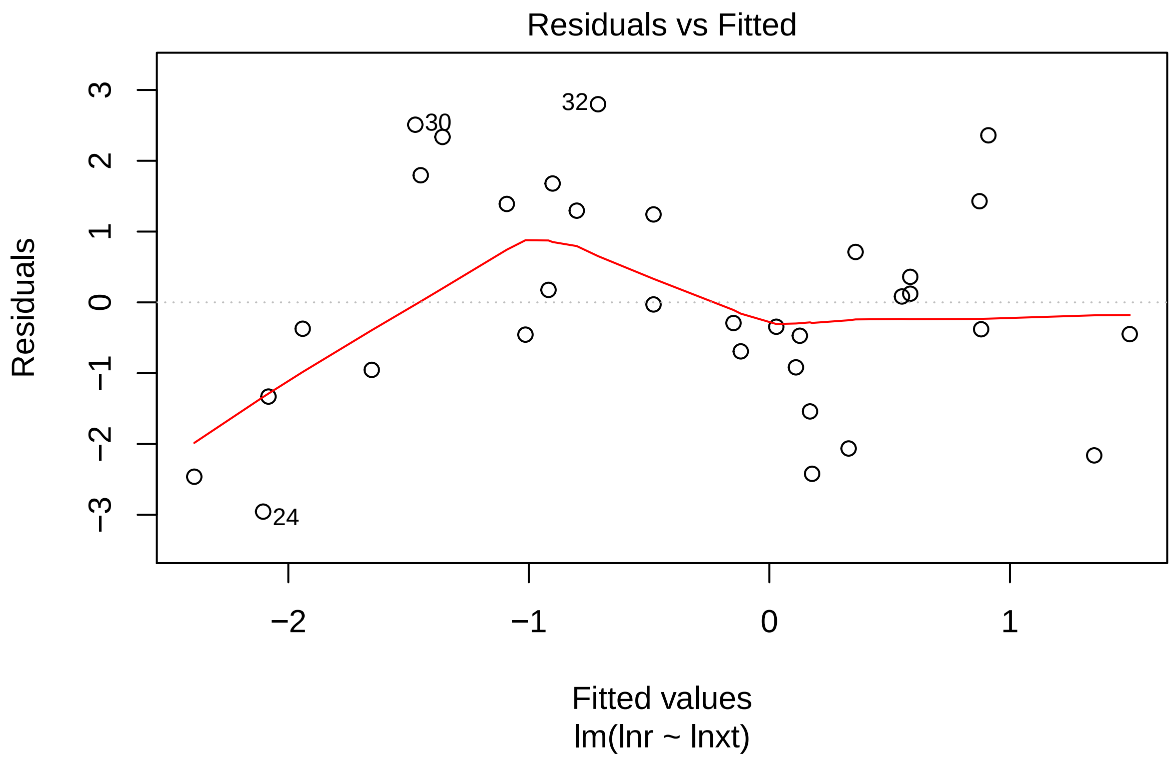
```r
detach(monitor)
```

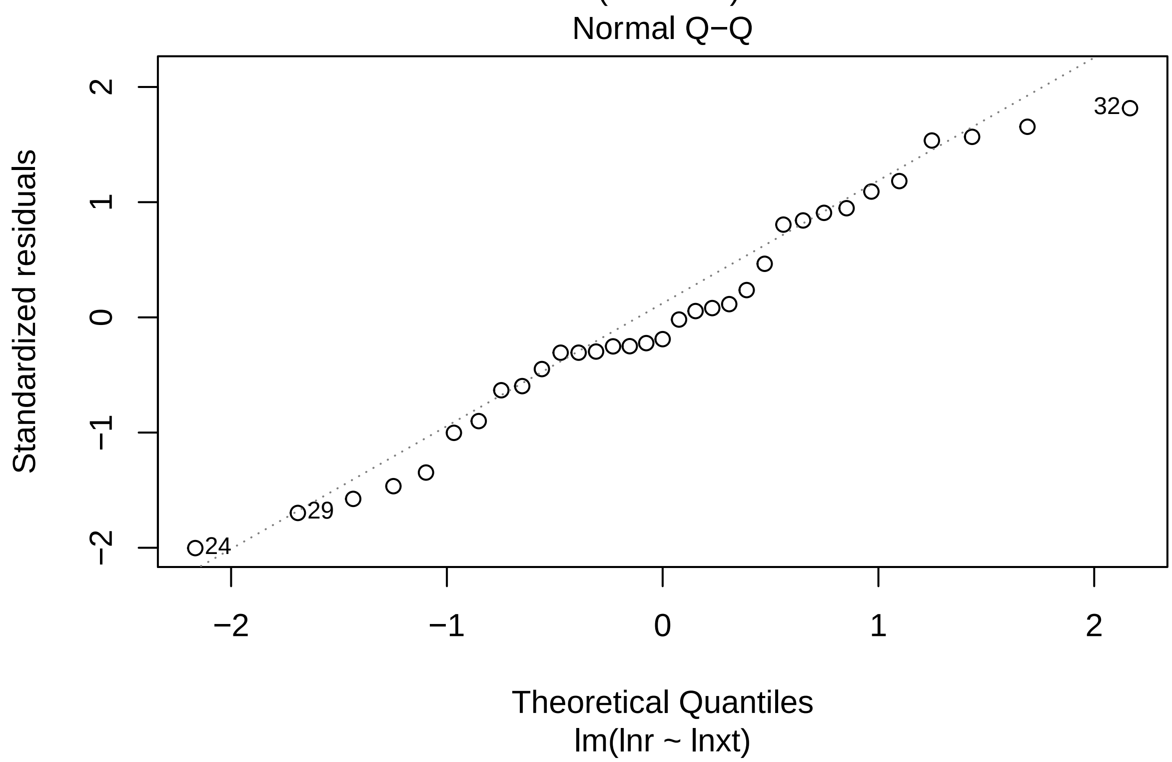
Model the limit data

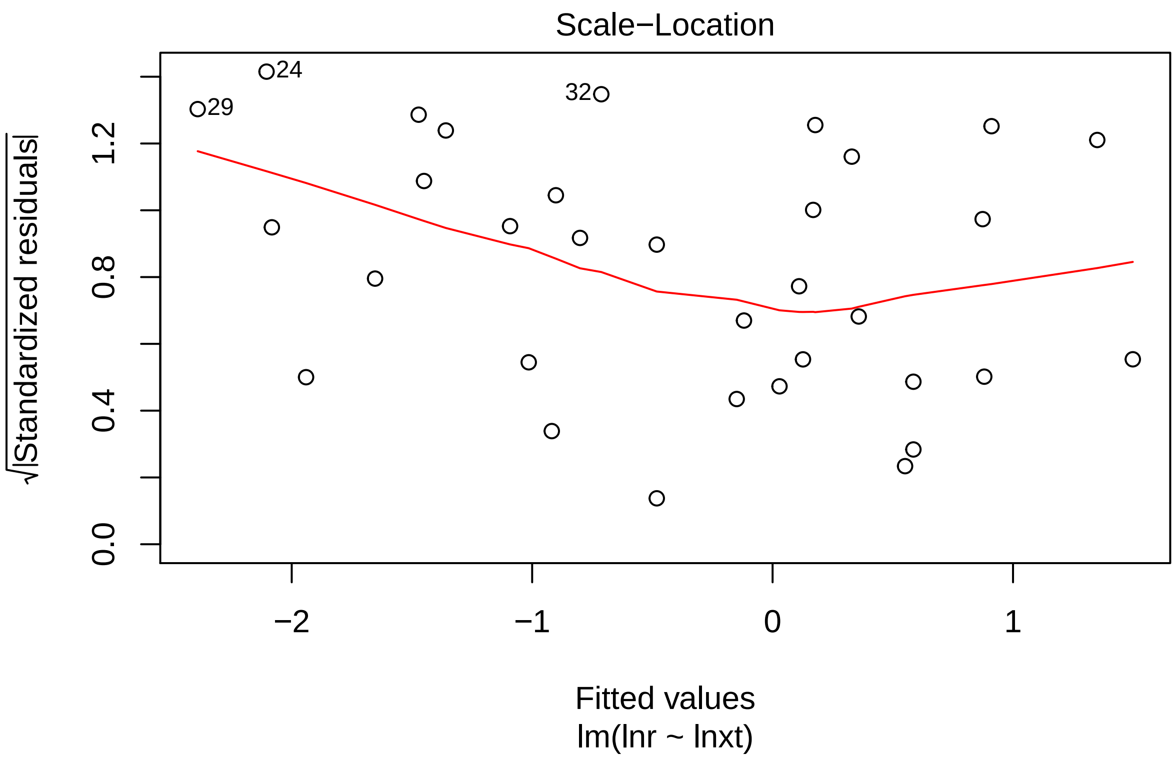
```r
attach(limit)

# fit a linear  model
fm1limit.lm <- lm(lnr ~ lnxt)

summary(fm1limit.lm) # check the summary
```

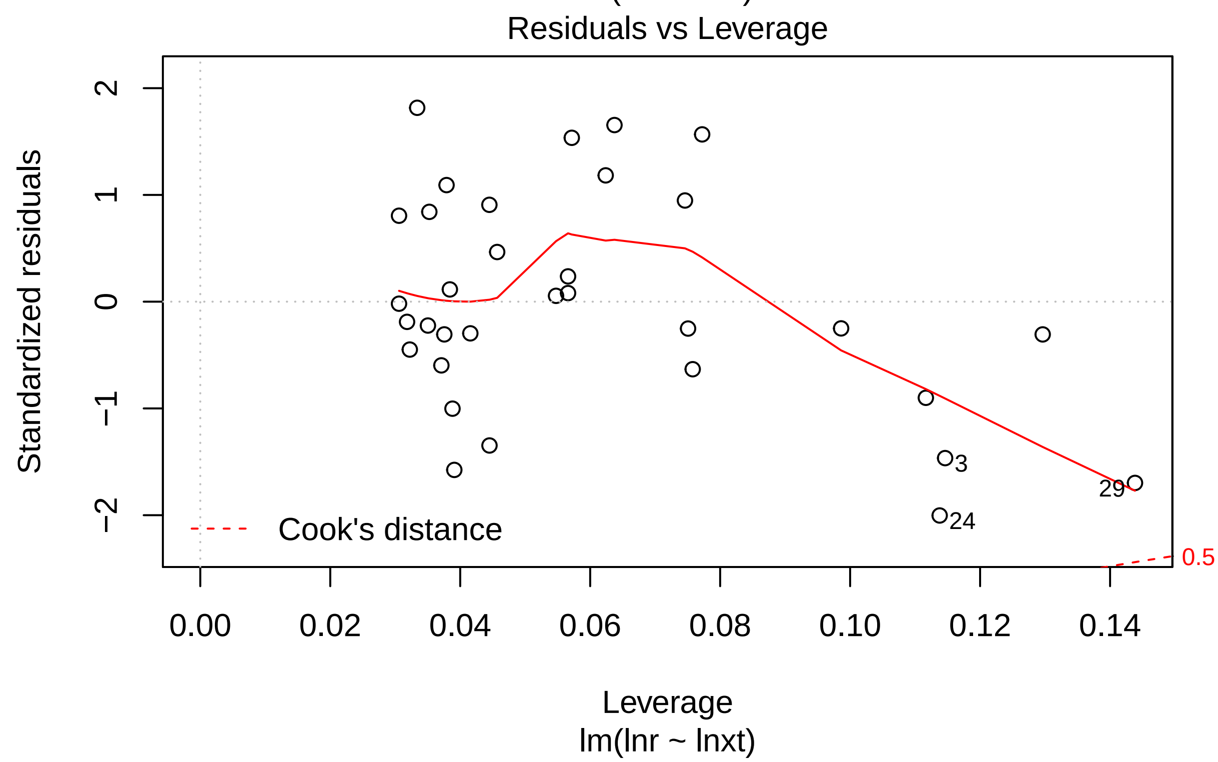
```
##
## Call:
## lm(formula = lnr ~ lnxt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.9554 -0.9178 -0.2917  1.2953  2.7986
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.4657     0.3522   1.322 0.195760
## lnxt         -0.5886     0.1547  -3.805 0.000626 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.567 on 31 degrees of freedom
## Multiple R-squared:  0.3184, Adjusted R-squared:  0.2964
## F-statistic: 14.48 on 1 and 31 DF,  p-value: 0.0006261
```

```r
plot(fm1limit.lm)
```


Residuals vs Fitted
Residuals
Fitted values
lm(lnr ~ lnxt)
30
32
24


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(lnr ~ lnxt)
32
29
24



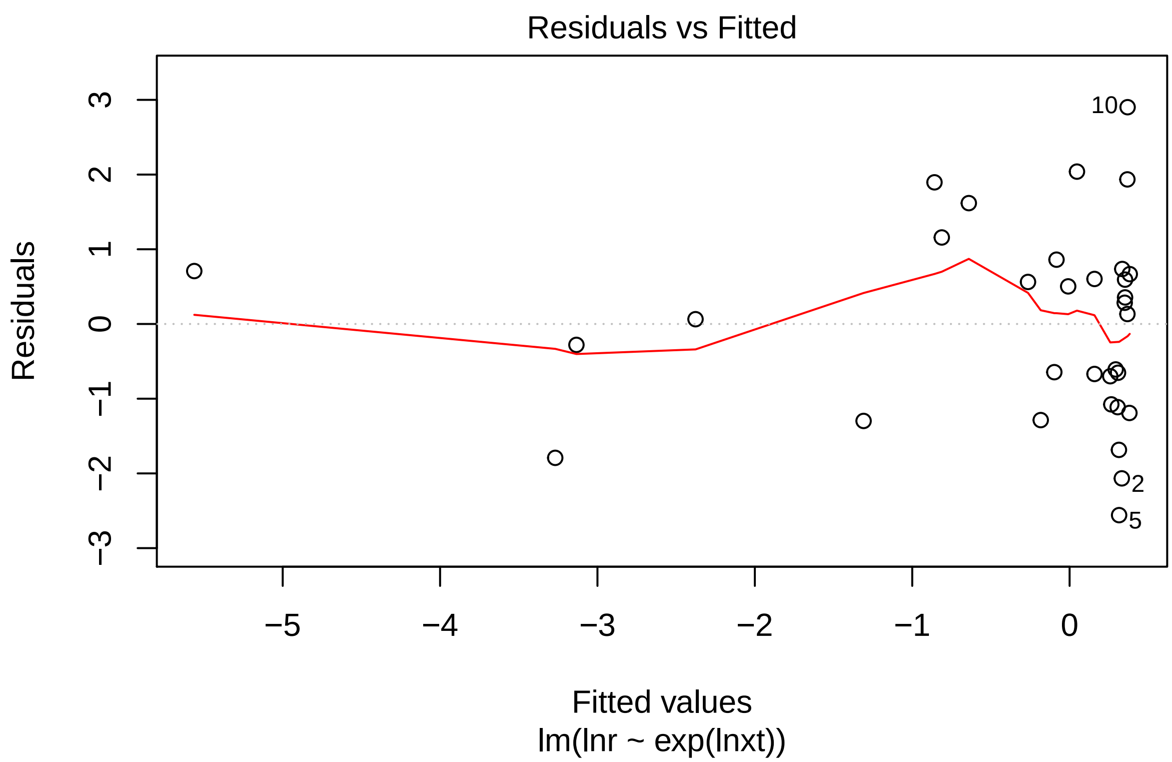


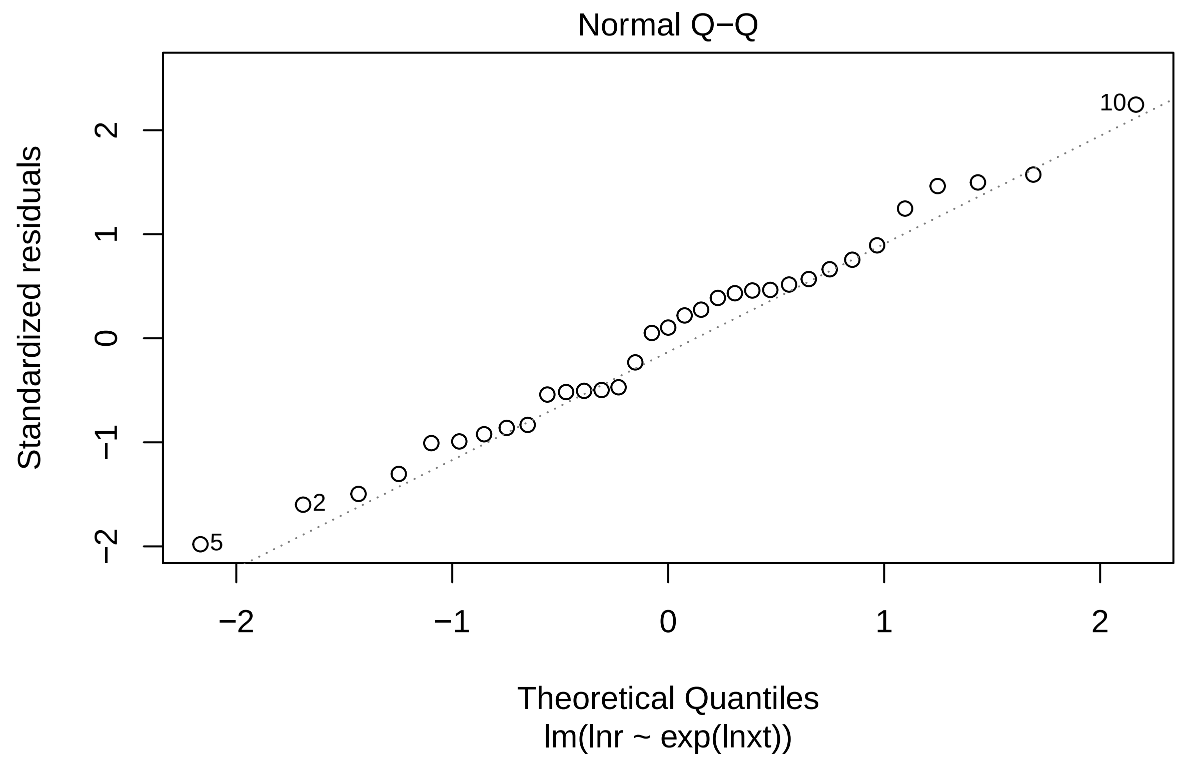

```r
# fit the exponential model.
fm2limit.lm <- lm(lnr ~ exp(lnxt))

summary(fm2limit.lm)
```

```
##
```

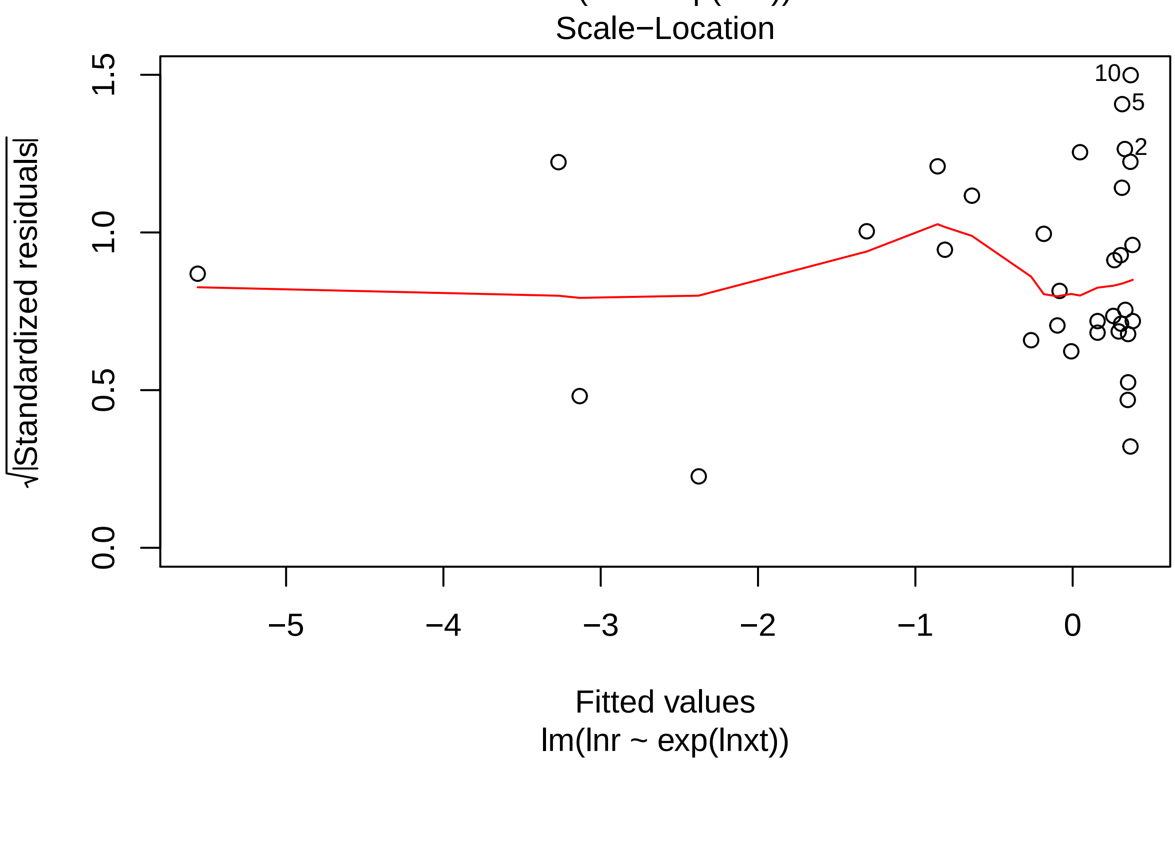
```
## Call:
## lm(formula = lnr ~ exp(lnxt))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5584 -1.0760  0.1333  0.7084  2.9016
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.390418   0.265601   1.470    0.152
## exp(lnxt)   -0.046431   0.008045  -5.771 2.36e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.318 on 31 degrees of freedom
## Multiple R-squared:  0.5179, Adjusted R-squared:  0.5024
## F-statistic: 33.31 on 1 and 31 DF,  p-value: 2.356e-06
```

```r
plot(fm2limit.lm)
```

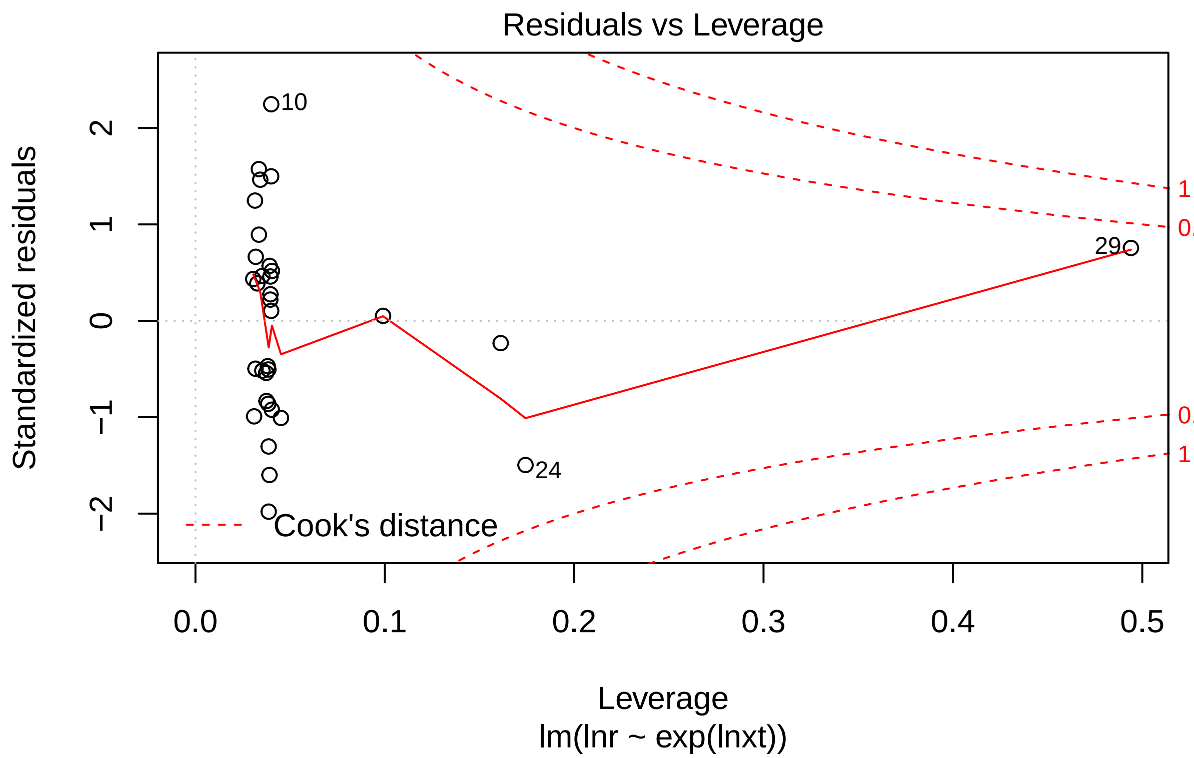




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(lnr ~ exp(lnxt))
10
2
5
−2
−1
0
1
2



Scale−Location
√|Standardized residuals|
Fitted values
lm(lnr ~ exp(lnxt))
10
5
2
0.0
0.5
1.0
1.5
−5
−4
−3
−2
−1
0




```r
# is the exponential model significant
anova(fm2limit.lm) ## Yes
```

```
## Analysis of Variance Table
##
## Response: lnr
##           Df Sum Sq Mean Sq F value    Pr(>F)
## exp(lnxt)  1 57.876  57.876  33.307 2.356e-06 ***
## Residuals 31 53.866   1.738
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
detach(limit)
```

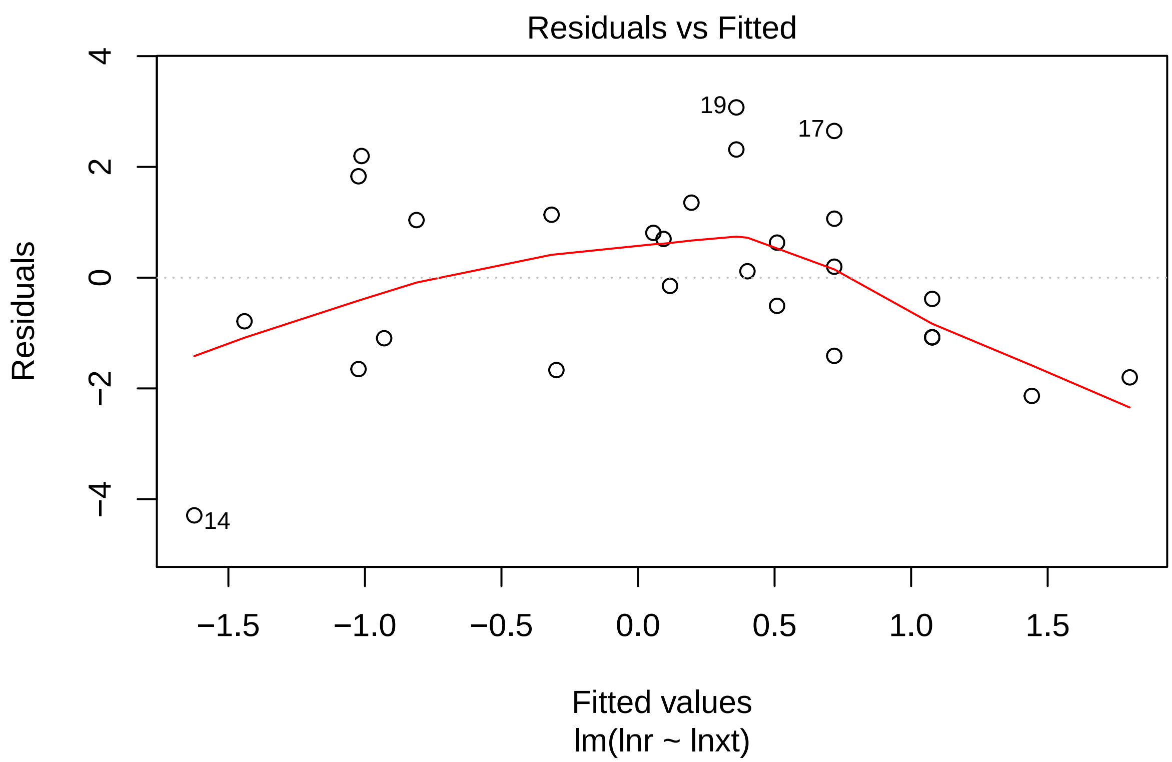
Model the basal area data

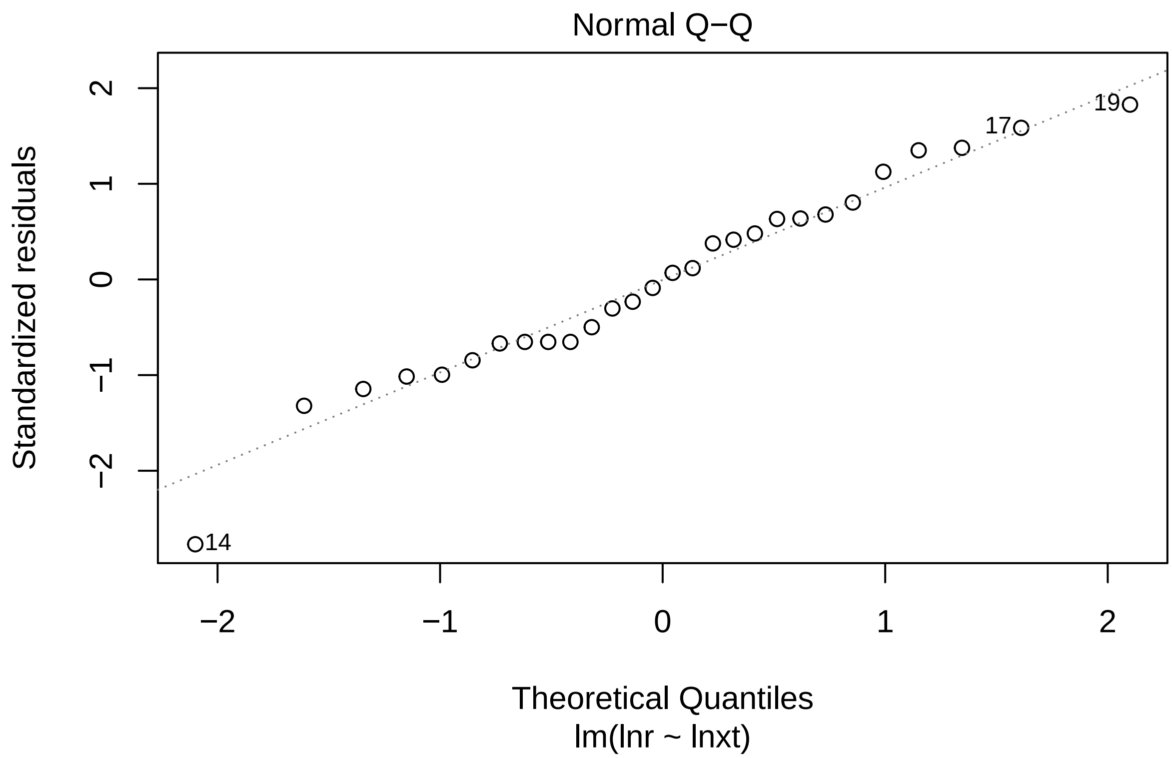
```r
attach(basal)

# fit a linear model
fm1basal.lm <- lm(lnr ~ lnxt)

summary(fm1basal.lm)
```

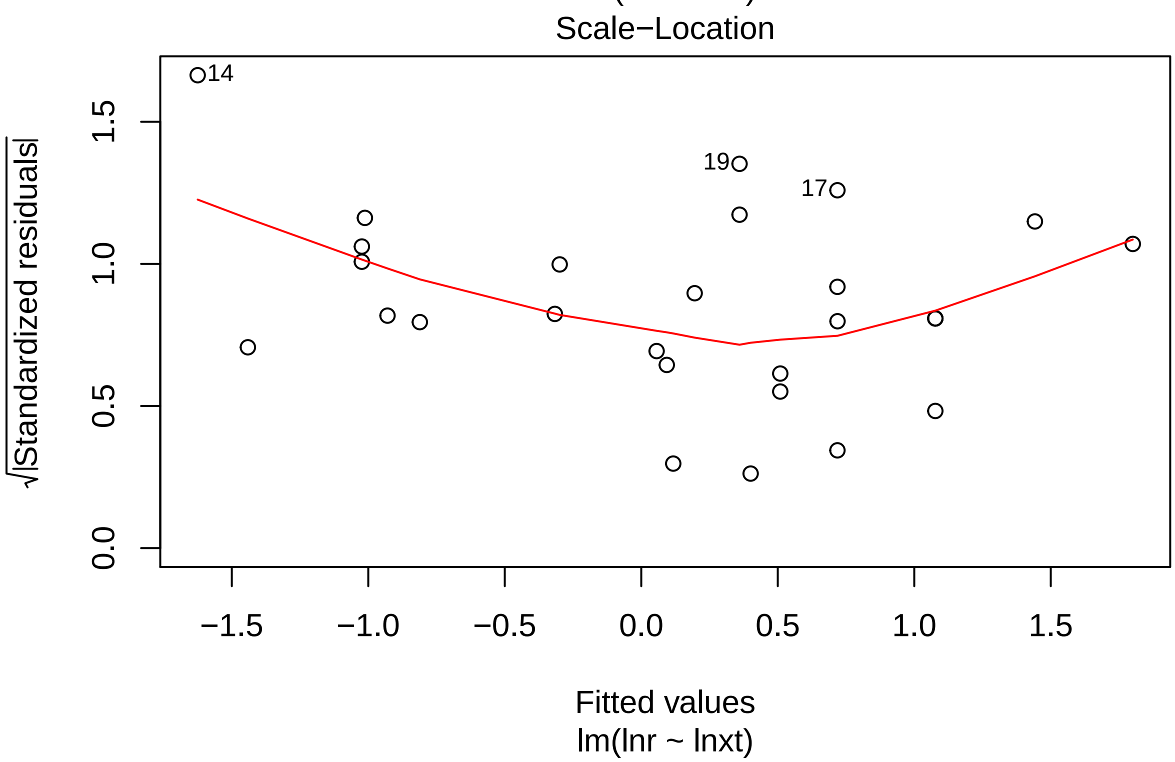
```
##
## Call:
## lm(formula = lnr ~ lnxt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.2918 -1.0813 -0.0167  1.0825  3.0740
```

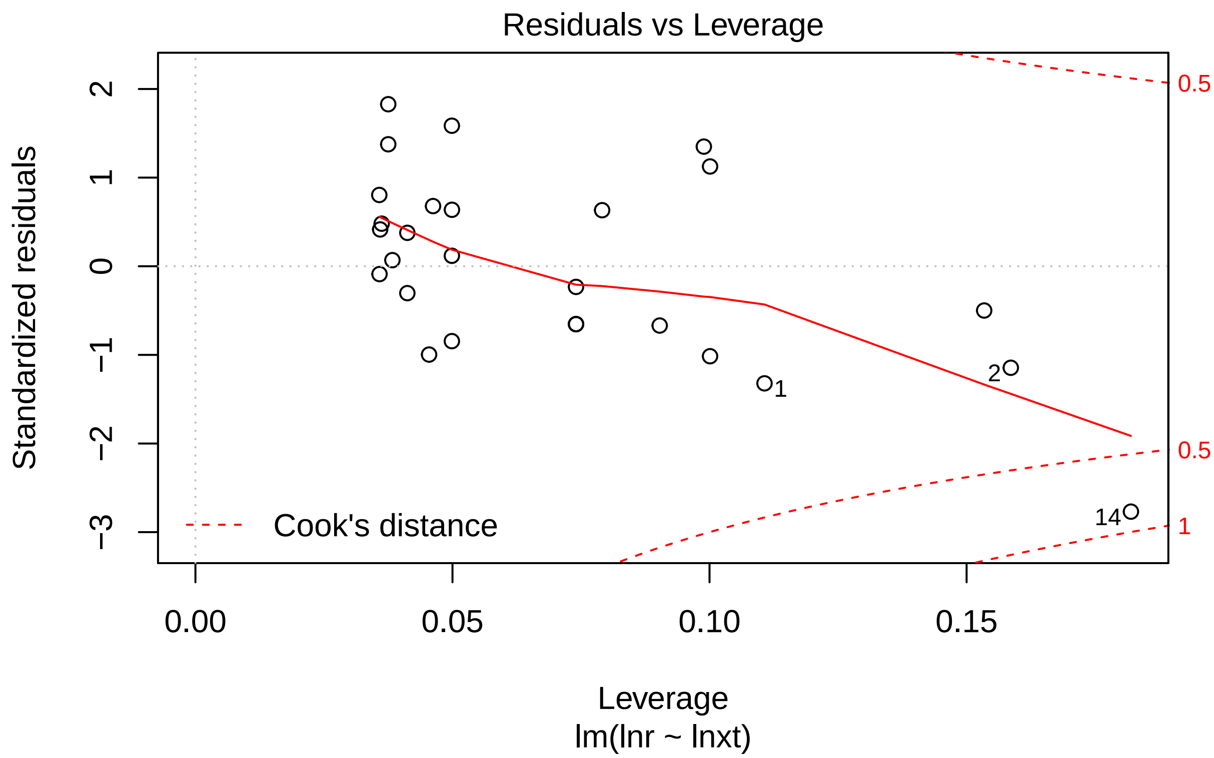
```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.0775     0.4664   2.310   0.0291 *
## lnxt         -0.5174     0.1897  -2.727   0.0113 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.714 on 26 degrees of freedom
## Multiple R-squared:  0.2224, Adjusted R-squared:  0.1925
## F-statistic: 7.435 on 1 and 26 DF,  p-value: 0.0113
```

```
plot(fm1basal.lm)
```




Normal Q−Q
Standardized residuals
−2
−1
0
1
2
14
17
19
Theoretical Quantiles
lm(lnr ~ lnxt)

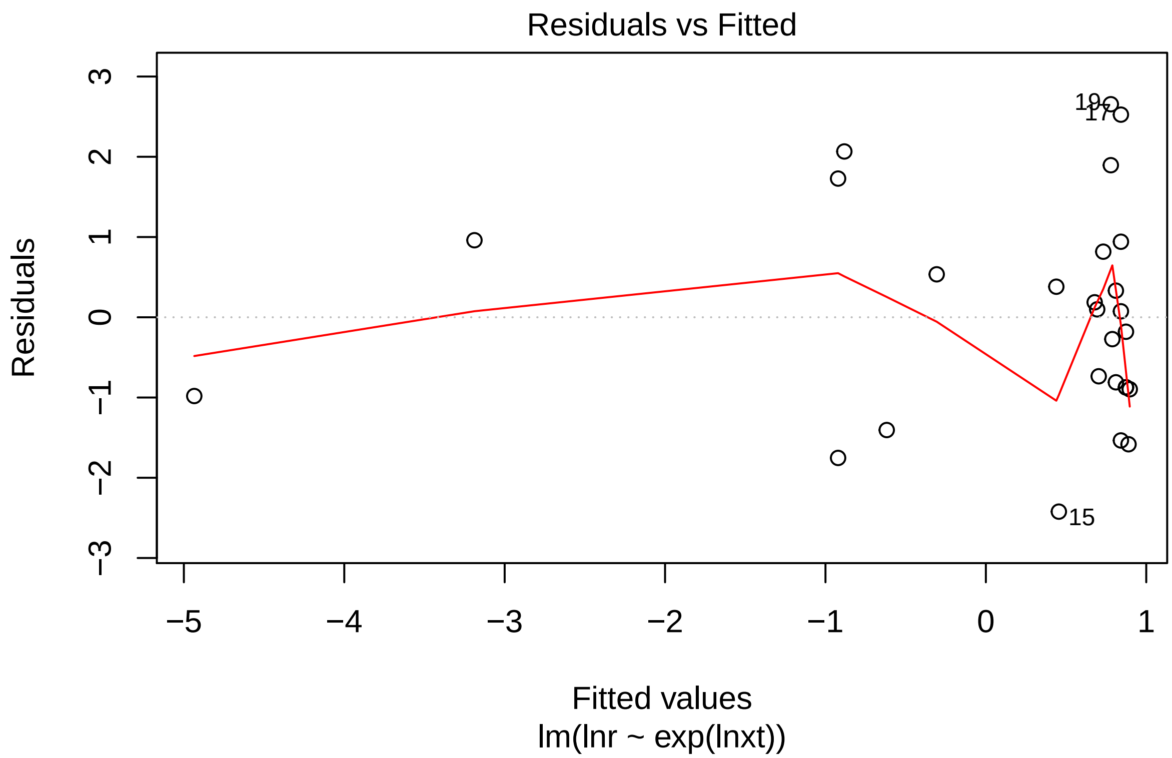

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
−1.5
−1.0
−0.5
14
19
17
Fitted values
lm(lnr ~ lnxt)

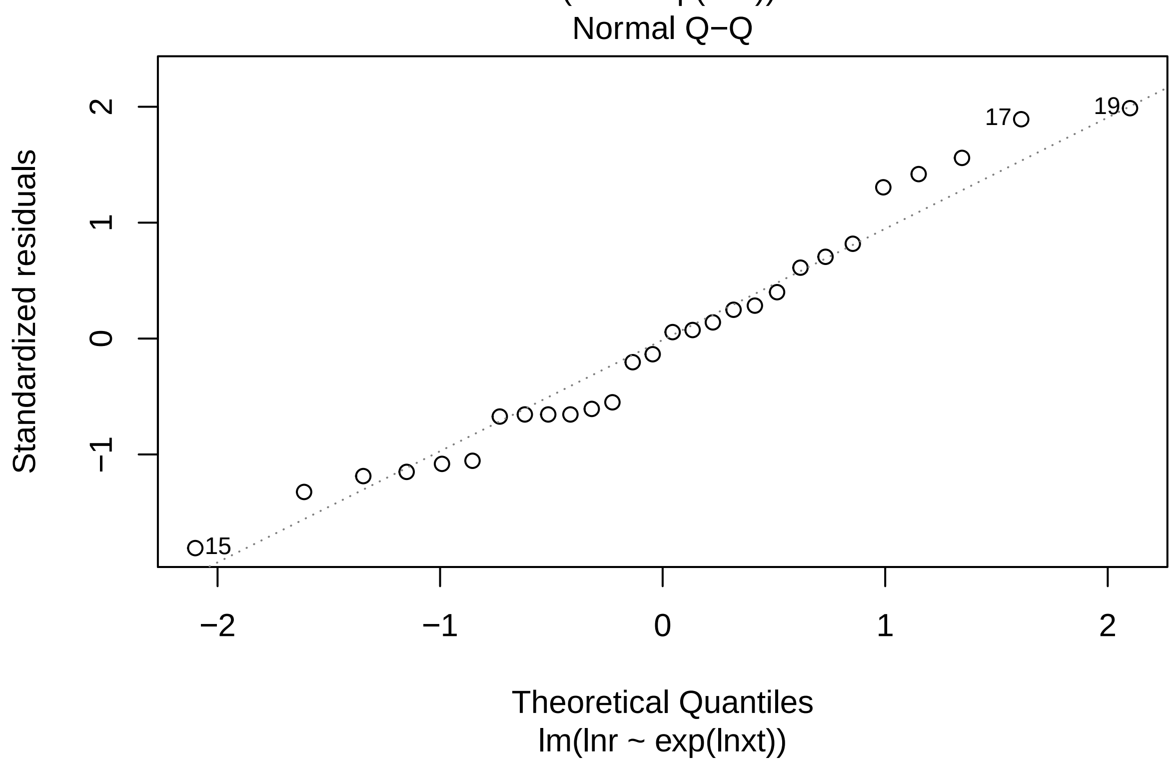


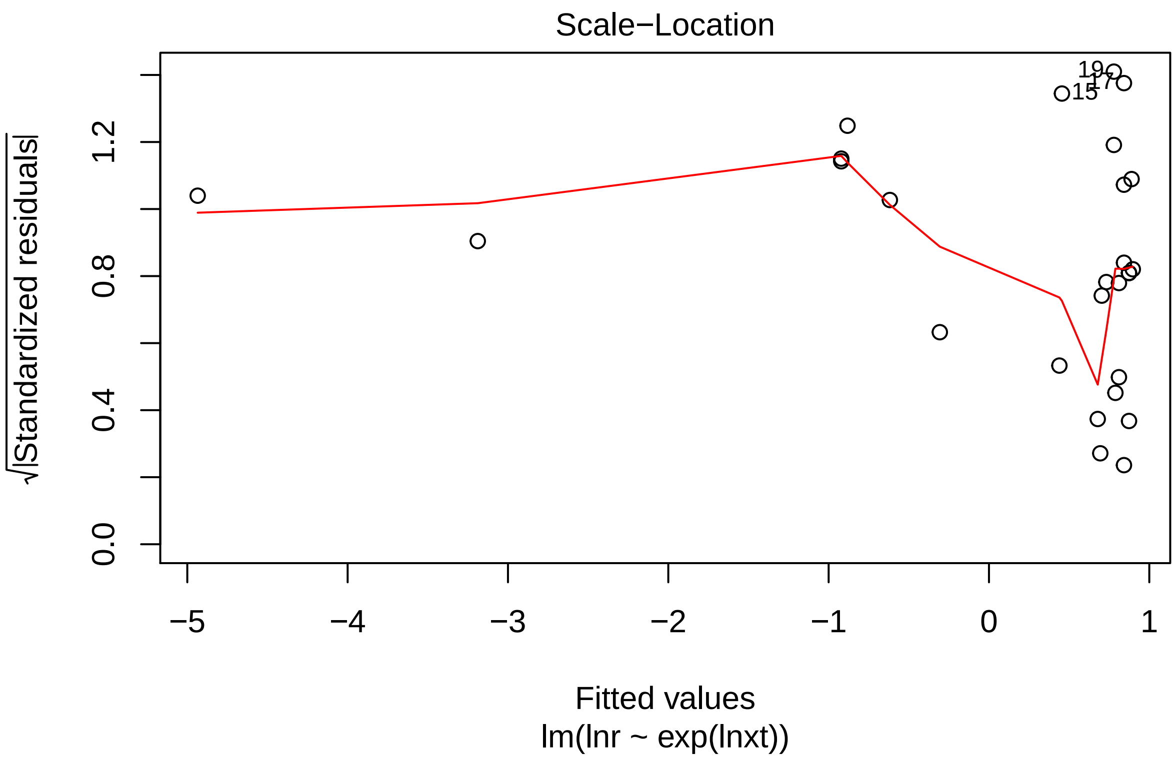

```r
#fit the exponential model
fm2basal.lm <- lm(lnr ~ exp(lnxt))

summary(fm2basal.lm)
```

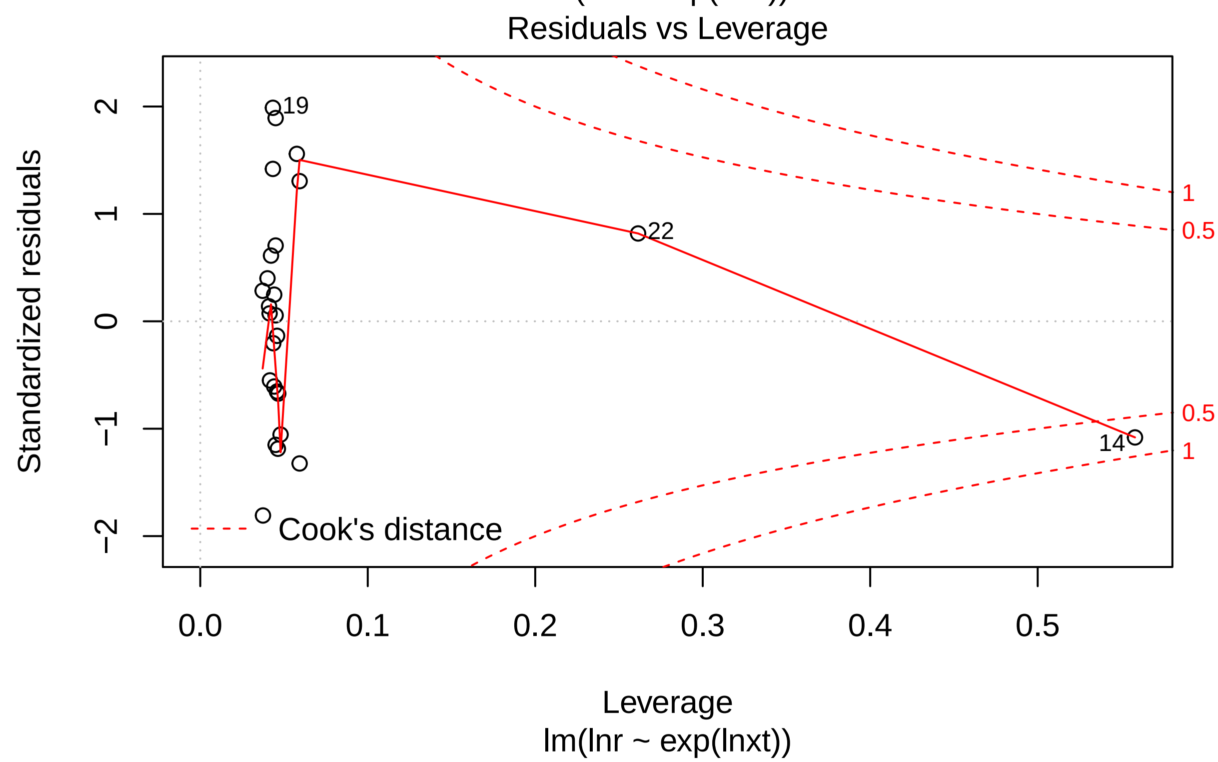
```
##
## Call:
## lm(formula = lnr ~ exp(lnxt))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.42211 -0.87941 -0.05302  0.84853  2.65496
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.904973   0.295404   3.064  0.00504 **
## exp(lnxt)   -0.031467   0.006092  -5.165 2.18e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.365 on 26 degrees of freedom
## Multiple R-squared:  0.5064, Adjusted R-squared:  0.4874
## F-statistic: 26.68 on 1 and 26 DF,  p-value: 2.175e-05
```

```r
plot(fm2basal.lm)
```


Residuals vs Fitted
Residuals
3
2
1
0
−1
−2
−3
19
17
15
−5
−4
−3
−2
−1
0
1
Fitted values
lm(lnr ~ exp(lnxt))


Normal Q−Q
Standardized residuals
2
1
0
−1
17
19
15
−2
−1
0
1
2
Theoretical Quantiles
lm(lnr ~ exp(lnxt))






```
#is the exponential model sig?
anova(fm2basal.lm) ## Yes
```

```
## Analysis of Variance Table
##
## Response: lnr
##             Df Sum Sq Mean Sq F value    Pr(>F)
```

```
## exp(lnxt)  1 49.742  49.742  26.678 2.175e-05 ***
## Residuals 26 48.479   1.865
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
detach(basal)
```

Compare the four linear models

```r
# extract and plot the slope and intercepts of the linear model with standard errors,
# show in console
intercepts <- rbind(summary(fm1control.lm)$coef[1, 1:2],
                    summary(fm1monitor.lm)$coef[1, 1:2],
                    summary(fm1limit.lm)$coef[1, 1:2],
                    summary(fm1basal.lm)$coef[1, 1:2])

slopes <- rbind(summary(fm1control.lm)$coef[2, 1:2],
                summary(fm1monitor.lm)$coef[2, 1:2],
                summary(fm1limit.lm)$coef[2, 1:2],
                summary(fm1basal.lm)$coef[2, 1:2])

lmsummary.table <- as.data.frame(cbind(intercepts, slopes), row.names = treats)

dimnames(lmsummary.table)[[2]] <- c("i", "i_std", "slope", "slope_std")

rm(intercepts, slopes)

lmsummary.table
```

```
##                 i     i_std      slope slope_std
## control 2.1932856 0.6058146 -0.6310441 0.1596007
## monitor 1.0321923 0.2524582 -0.4015135 0.0804381
## limit   0.4656501 0.3521658 -0.5885993 0.1546885
## basal   1.0775078 0.4664349 -0.5173533 0.1897348
```

Compare the four exponential decline models

```r
# fit the exponential model to the control and limit data
fm2control.lm <- lm(lnr ~ exp(lnxt), data = cntrl)
fm2monitor.lm <- lm(lnr ~ exp(lnxt), data = monitor)

# extract and plot the slope and intercepts of the linear model with standard errors
# display in console

intercepts <- rbind(summary(fm2control.lm)$coef[1, 1:2],
                    summary(fm2monitor.lm)$coef[1, 1:2],
                    summary(fm2limit.lm)$coef[1, 1:2],
                    summary(fm2basal.lm)$coef[1, 1:2])

slopes <- rbind(summary(fm2control.lm)$coef[2, 1:2],
                summary(fm2monitor.lm)$coef[2, 1:2],
                summary(fm2limit.lm)$coef[2, 1:2],
```

```r
                summary(fm2basal.lm)$coef[2, 1:2])

lmexpsummary.table <- as.data.frame(cbind(intercepts, slopes), row.names = treats)

dimnames(lmexpsummary.table)[[2]] <- c("i", "i_std", "slope", "slope_std")

rm( intercepts, slopes )

lmexpsummary.table
```

```
##                 i     i_std        slope   slope_std
## control 0.5144092 0.2627504 -0.006130507 0.001591242
## monitor 0.2923428 0.1346793 -0.009369376 0.001782170
## limit   0.3904178 0.2656011 -0.046430987 0.008045204
## basal   0.9049725 0.2954044 -0.031466652 0.006092242
```

Plot Figure 2

```r
# a 2x2 layout with marginal y and x axis, subtitles above panels a+b and c+d;
nf <- layout(matrix(c( 0, 1, 1, 2, 3, 4, 2, 5, 5, 2, 6, 7, 0, 8, 8),
                    nrow = 5, ncol = 3, byrow = TRUE),
             widths = c(1, 4, 4), heights = c(1, 8, 1, 8, 2))

par(mar = c(0, 0, 0, 0))

# plot a+b subtitle
text.box(label.text = expression(bold("natural forests")), cex = 2.0)

# y-axis label
text.box(label.text = expression(bold(paste(log[e], " (", bolditalic( x[t + 1] / x[t] ), " )"))),
         srt = 90, cex = 2.0)

par(mar = c(1, 2, 1, 1))

# monitoring data
#   determine the predicted values from fm1monitor.lm for the values given in the pred.seq vector
plot.box(pred.seq = seq(-1, 6, 0.25), model = fm1monitor.lm,
         x.data = monitor$lnxt, y.data = monitor$lnr, # the raw data
         IDletter.plot = "a", name.plot = "monitor", # label the plot
         equation.text = expression(paste(italic(y), " = 1.01 - 0.40", italic(x), "; ",
                                          italic(r) ^ 2, " = 0.20; ",
                                          italic(F)["1,92"], " = 24.92; ",
                                          italic(p), " < 0.01")), cex = 1.5)

# control data
plot.box(pred.seq = seq(1, 6, 0.25), model = fm1control.lm, x.data = cntrl$lnxt,
         y.data = cntrl$lnr, cex = 1.5, IDletter.plot = "b", name.plot = "control",
         equation.text = expression(paste(italic(y), " = 2.19 - 0.63", italic(x), "; ",
                                          italic(r)^2, " = 0.35; ",
                                          italic(F)["1,29"], " = 15.63; ",
                                          italic(p), " < 0.01" )))

par( mar = c( 0, 0, 0, 0 ) )
```

```
# plot c+d subtitle
text.box(label.text = expression(bold("thinned forests")), cex = 2.0)

par(mar = c(1, 2, 1, 1))

# limit data
plot.box(pred.seq = seq(-2, 6, 0.25), model = fm2limit.lm, x.data = limit$lnxt, y.data = limit$lnr,
         cex = 1.5, IDletter.plot = "c", name.plot = "limit",
         equation.text = expression( paste(   italic(y), " = 0.39 - 0.04e"^italic(x), "; ",
                                              italic(r)^2, " = 0.50; ",
                                              italic(F)["1,31"], " = 33.31; ",
                                              italic(p), " < 0.01" )))

# basal data
plot.box(pred.seq = seq(-2, 6, 0.25), model = fm2basal.lm, x.data = basal$lnxt,
         y.data = basal$lnr, cex = 1.5, IDletter.plot = "d", name.plot = "basal",
         equation.text = expression(paste(italic(y), " = 0.90 - 0.03e"^italic(x), "; ",
                                          italic(r)^2, " = 0.48; ",
                                          italic(F)["1,26"], " = 26.68; ",
                                          italic(p), " < 0.01" )))

# x-axis
par(mar = c(0, 0, 0, 0))
text.box(label.text = expression(bold(paste(log[e], bolditalic(" n"[t])))), cex = 2.0)
```

Part 2

Test reproduction rates predicted from the four models against observed tree mortality from 27 locations in the United States

NOTES:

1. This analysis implements a chi-square goodness-of-fit test via a custom function, `multipleX2`, that calls `chisq.test` internally. `multipleX2` computes values of the Goodness-of-fit chi-square statistic comparing observed `r` values to `r` values predicted from `Xt` using the 2 linear and 2 exponential models developed in part 1. Within the function data are transformed from `Rt` to `r` and `Xt` to `xt` (using `exp`). This procedure was necessary to meet the assumption of no negative values inherent to a chi-square test and because applying `chisq.test` to negative values produces a critical error (`chisq.test` will fail to run).

2. Chi-square tests for < 5 observations are typically not advised and therefore applying `multipleX2` for forests with < 5 observations will cause R to throw warnings even though it will calculate the test statistics. A different test may be more appropriate here (*e.g.*, Fisher's exact test, or a G test), for our purposes the chi-square approach was sufficient.

```
# declare 2 useful vectors. Note that districts are administrative subunits of forests, see Appendix B.
forests <- c("BH", "DL", "FL", "GL", "KO", "LO", "NZ")

districts <- c("BLA", "JEF", "GVW", "HHR", "SPB", "SWL", "TAL", "FLA", "CAB", "FIR",
               "FOR", "LIB", "REX", "YAK", "NIM", "PLA", "SUP", "THF", "ELK", "RED")
```

Pre-analysis Data manipulation

```r
# calculate some values
usforests$r <- usforests$xtplusone / usforests$xt
usforests$lnr <- log(usforests$r)
usforests$lnxt <- log(usforests$xt)

# limit the analysis to data points where xt != 0
usforests <- usforests[which(usforests$xt > 0 & usforests$r > 0 ), ]

# lose all records for "AVG" districts = Forest averages, are legacy values in the dataset
usforests <- usforests[which(usforests$district != "AVG"), ]
```

```r
attach(usforests)

## test forests

# Beaverhead NF
BH <- multipleX2(x = lnxt[which(forest == "BH")], obs <- lnr[which(forest == "BH")])
# Deerlodge NF
DL <- multipleX2(x = lnxt[which(forest == "DL")], obs <- lnr[which(forest == "DL")])
# Flathead NF
FL <- multipleX2(x = lnxt[which(forest == "FL")], obs <- lnr[which(forest == "FL")])
# Gallatin NF
GL <- multipleX2(x = lnxt[which(forest == "GL")], obs <- lnr[which(forest == "GL")])
# Kootenai NF
KO <- multipleX2(x = lnxt[which(forest == "KO")], obs <- lnr[which(forest == "KO")])
# Lolo NF
LO <- multipleX2(x = lnxt[which(forest == "LO")], obs <- lnr[which(forest == "LO")])
# Nez Perce NF
NZ <- multipleX2(x = lnxt[which(forest == "NZ")], obs <- lnr[which(forest == "NZ")])

## test districts

# Blackfoot IR
BLA <- multipleX2(x = lnxt[which(district == "BLA")], obs <- lnr[which(district == "BLA")])
# Jefferson RD
JEF <- multipleX2(x = lnxt[which(district == "JEF")], obs <- lnr[which(district == "JEF")])
# Flathead IR
FLA <- multipleX2(x = lnxt[which(district == "FLA")], obs <- lnr[which(district == "FLA")])
# Glacier view RD
GVW <- multipleX2(x = lnxt[which(district == "GVW")], obs <- lnr[which(district == "GVW")])
# Hungry Horse RD
HHR <- multipleX2(x = lnxt[which(district == "HHR")], obs <- lnr[which(district == "HHR")])
# Spotted Bear RD
SPB <- multipleX2(x = lnxt[which(district == "SPB")], obs <- lnr[which(district == "SPB")])
# Swan Lake
SWL <- multipleX2(x = lnxt[which(district == "SWL")], obs <- lnr[which(district == "SWL")])
# Tally Lake
TAL <- multipleX2(x = lnxt[which(district == "TAL")], obs <- lnr[which(district == "TAL")])
# Cabinet RD
CAB <- multipleX2(x = lnxt[which(district == "CAB")], obs <- lnr[which(district == "CAB")])
# Fisher River RD
FIR <- multipleX2(x = lnxt[which(district == "FIR")], obs <- lnr[which(district == "FIR")])
```

```r
# Fortine RD
FOR <- multipleX2(x = lnxt[which(district == "FOR")], obs <- lnr[which(district == "FOR")])
# Libby RD
LIB <- multipleX2(x = lnxt[which(district == "LIB")], obs <- lnr[which(district == "LIB")])
# Rexford RD
REX <- multipleX2(x = lnxt[which(district == "REX")], obs <- lnr[which(district == "REX")])
# Yaak RD
YAK <- multipleX2(x = lnxt[which(district == "YAK")], obs <- lnr[which(district == "YAK")])
# Ninemile RD
NIM <- multipleX2(x = lnxt[which(district == "NIM")], obs <- lnr[which(district == "NIM")])
# Plains RD
PLA <- multipleX2(x = lnxt[which(district == "PLA")], obs <- lnr[which(district == "PLA")])
# Superior RD
SUP <- multipleX2(x = lnxt[which(district == "SUP")], obs <- lnr[which(district == "SUP")])
# Thompson Falls RD
THF <- multipleX2(x = lnxt[which(district == "THF")], obs <- lnr[which(district == "THF")])
# Elk RD
ELK <- multipleX2(x = lnxt[which(district == "ELK")], obs <- lnr[which(district == "ELK")])
# Red River RD
RED <- multipleX2(x = lnxt[which(district == "RED")], obs <- lnr[which(district == "RED")])

detach(usforests)
```

```r
# combine the test statistics from the chi-square tests for all the forests into one data frame,
# do the same for the statistics from the tests of the individual districts

# forests
summary.forests <- NULL

cntrl <- NULL
monitor <- NULL
limit <- NULL
basal <- NULL
limit2 <- NULL

for (i in 1:length(forests)) {
  cntrl <- rbind(cntrl, as.list(get(forests[i])[1, ]))
  monitor <- rbind(monitor, as.list(get(forests[i])[2, ]))
  limit <- rbind(limit, as.list(get(forests[i])[3, ]))
  basal <- rbind(basal, as.list(get(forests[i])[4, ]))
  limit2 <- rbind(limit2, as.list(get(forests[i])[5, ]))
  summary.forests <- cbind(cntrl, monitor, limit, basal, limit2)
}
summary.forests <- cbind(forests, summary.forests)

# districts
summary.districts <- NULL

cntrl <- NULL
monitor <- NULL
limit <- NULL
basal <- NULL
limit2 <- NULL
```

```
for (i in 1:length(districts)) {
  cntrl <- rbind(cntrl, as.list(get(districts[i])[1, ]))
  monitor <- rbind(monitor, as.list(get(districts[i])[2, ]))
  limit <- rbind(limit, as.list(get(districts[i])[3, ]))
  basal <- rbind(basal, as.list(get(districts[i])[4, ]))
  limit2 <- rbind(limit2, as.list(get(districts[i])[5, ]))
  summary.districts <- cbind(cntrl, monitor, limit, basal, limit2)
}
summary.districts <- cbind(districts, summary.districts)

# write the summaries to the console

# columns 2-4 are for tests of the 'monitor' model;
# columns 5-7 are tests of the 'control' model;
# columns 8-10 are tests of the  'limit' model;
# columns 11-13 are tests of the 'basal' model;
# columns 14-16 are tests of the 'limit+2' model.

summary.forests
```

```
##      forests X2        degfree pvalue    X2        degfree pvalue
## [1,] "BH"    0.8896695 3       0.82792   0.5838979 3       0.9001085
## [2,] "DL"    11.36158  8       0.1820413 15.11728  8       0.05690483
## [3,] "FL"    25.1377   37      0.9308515 25.25752  37      0.9284194
## [4,] "GL"    1.478832  8       0.9930538 1.52465   8       0.9922909
## [5,] "KO"    59.97578  50      0.1577584 84.73259  50      0.001565023
## [6,] "LO"    18.98032  34      0.9824271 18.40759  34      0.9864838
## [7,] "NZ"    2.095416  10      0.9955552 3.134466  10      0.97808
##      X2       degfree pvalue       X2       degfree pvalue       X2
## [1,] 2.508037 3       0.4738404    1.449301 3       0.69402      2.508037
## [2,] 10.01474 8       0.263993     10.44282 8       0.2353104    10.01474
## [3,] 250.555  37      1.564863e-33 77.81751 37      9.943939e-05 250.555
## [4,] 3.966814 8       0.8601051    2.293142 8       0.9706804    3.966814
## [5,] 423.1893 50      1.503203e-60 146.9879 50      1.766087e-11 423.1893
## [6,] 143.3544 34      2.208172e-15 47.49807 34      0.06198322   143.3544
## [7,] 4.977745 10      0.8926597    4.708836 10      0.9097604    4.977745
##      degfree pvalue
## [1,] 3       0.4738404
## [2,] 8       0.263993
## [3,] 37      1.564863e-33
## [4,] 8       0.8601051
## [5,] 50      1.503203e-60
## [6,] 34      2.208172e-15
## [7,] 10      0.8926597
```

```
summary.districts
```

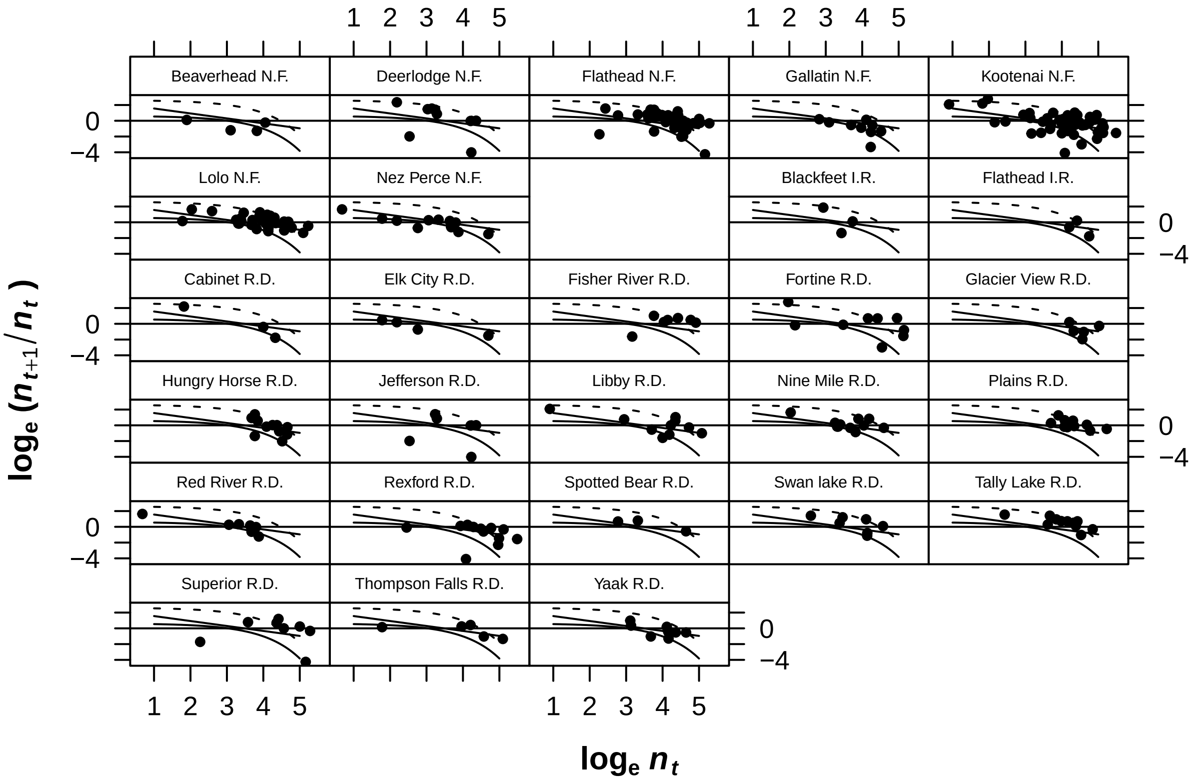
```
##       districts X2        degfree pvalue     X2        degfree
##  [1,] "BLA"     5.20934   2       0.07392753 6.269824  2
##  [2,] "JEF"     6.995401  5       0.2209826  7.053938  5
##  [3,] "GVW"     1.161087  4       0.8844629  1.166366  4
##  [4,] "HHR"     7.276728  10      0.6990873  8.56312   10
##  [5,] "SPB"     0.2749253 2       0.8715669  0.3584634 2
##  [6,] "SWL"     3.179039  10      0.976891   3.795631  10
```

```
##  [7,] "TAL"     10.68888  7       0.1527774  7.862952  7
##  [8,] "FLA"     0.8011257 2       0.6699428  0.8264221 2
##  [9,] "CAB"     2.441259  2       0.2950444  5.034153  2
## [10,] "FIR"     3.921121  6       0.6873501  3.061663  6
## [11,] "FOR"     19.83353  8       0.01098454 29.23947  8
## [12,] "LIB"     6.564011  9       0.6824087  6.853581  9
## [13,] "REX"     3.234761  11      0.9872453  2.593018  11
## [14,] "YAK"     1.901529  7       0.9650888  2.512079  7
## [15,] "NIM"     5.241905  10      0.8744437  5.939507  10
## [16,] "PLA"     3.625752  10      0.9626551  4.159551  10
## [17,] "SUP"     3.277431  6       0.7733067  3.744191  6
## [18,] "THF"     3.064314  4       0.5471209  1.70341   4
## [19,] "ELK"     0.1355912 3       0.9872483  0.3400579 3
## [20,] "RED"     0.7055155 6       0.9943713  1.404631  6
##       pvalue       X2        degfree pvalue       X2        degfree
##  [1,] 0.04350357   3.697117  2       0.157464     4.553932  2
##  [2,] 0.2166597    8.655655  5       0.1236126    6.450898  5
##  [3,] 0.8836045    12.83246  4       0.01212409   5.805005  4
##  [4,] 0.5740045    5.71374   10      0.8387129    4.643196  10
##  [5,] 0.8359122    3.073008  2       0.2151319    0.9710359 2
##  [6,] 0.9560958    15.53574  10      0.1137162    6.114609  10
##  [7,] 0.3448337    223.6109  7       1.129309e-44 56.61377  7
##  [8,] 0.6615226    0.9357325 2       0.6263373    0.7753907 2
##  [9,] 0.08069516   0.5190535 2       0.7714166    1.620417  2
## [10,] 0.8010686    50.60635  6       3.552914e-09 21.17963  6
## [11,] 0.0002878272 127.5131  8       9.265871e-24 44.03358  8
## [12,] 0.6523606    33.5892   9       0.000105406  14.54613  9
## [13,] 0.9950863    168.7485  11      2.134626e-30 34.17163  11
## [14,] 0.9261847    5.250413  7       0.6294336    2.709586  7
## [15,] 0.8203197    16.21683  10      0.09359146   10.18282  10
## [16,] 0.9398624    44.40645  10      2.781921e-06 13.76972  10
## [17,] 0.7112477    9.363327  6       0.1541522    4.970266  6
## [18,] 0.7900981    17.62328  4       0.001461822  5.792579  4
## [19,] 0.9523267    2.132969  3       0.5452718    0.7755629 3
## [20,] 0.9655761    0.9011257 6       0.9890843    1.760174  6
##       pvalue       X2        degfree pvalue
##  [1,] 0.102595     3.697117  2       0.157464
##  [2,] 0.264782     8.655655  5       0.1236126
##  [3,] 0.2141916    12.83246  4       0.01212409
##  [4,] 0.9137043    5.71374   10      0.8387129
##  [5,] 0.6153784    3.073008  2       0.2151319
##  [6,] 0.8055441    15.53574  10      0.1137162
##  [7,] 7.133151e-10 223.6109  7       1.129309e-44
##  [8,] 0.6786191    0.9357325 2       0.6263373
##  [9,] 0.4447654    0.5190535 2       0.7714166
## [10,] 0.001703124  50.60635  6       3.552914e-09
## [11,] 5.60707e-07  127.5131  8       9.265871e-24
## [12,] 0.1041816    33.5892   9       0.000105406
## [13,] 0.0003390399 168.7485  11      2.134626e-30
## [14,] 0.9105063    5.250413  7       0.6294336
## [15,] 0.424603     16.21683  10      0.09359146
## [16,] 0.183757     44.40645  10      2.781921e-06
## [17,] 0.5476323    9.363327  6       0.1541522
## [18,] 0.2151833    17.62328  4       0.001461822
```

```
## [19,] 0.8553027    2.132969  3       0.5452718
## [20,] 0.9403819    0.9011257 6       0.9890843
```

Plot Figure 3

```
# make two datasets, one containing the data for the forests that were tested,
# the other containing the districts that were tested

surv_for <- subset(usforests, forest %in% forests)
surv_dist <- subset(usforests, district %in% districts)
surv_dist$district <- ordered(surv_dist$district, levels = districts)

# bind the two datasets together and add a column of ID numbers,
# ID values are set so that plots are layed out in alphabetical order in the final plot.
# This is not the most elegant way of doing this.

surveyareas <- rbind(
  cbind(surv_for,
        ID = rep(c(1:7), times = as.vector(unlist(lapply(
          split(surv_for$forest, f = surv_for$forest, drop = TRUE), length))))
  ),
  cbind(surv_dist,
        ID = rep(c(8, 16, 14, 15, 22, 24, 25, 9, 10, 12, 13, 17, 21, 27, 18, 19, 23, 26, 11, 20),
                 times = as.vector(unlist(lapply(
                   split(surv_dist$district, f = surv_dist$district, drop = TRUE), length))))
  )
)

surveyareas$ID <- ordered(surveyareas$ID)

strip.titles <- c("Beaverhead N.F.", "Deerlodge N.F.", "Flathead N.F.", "Gallatin N.F.",
                  "Kootenai N.F.", "Lolo N.F.", "Nez Perce N.F.", "Blackfeet I.R.", "Flathead I.R.",
                  "Cabinet R.D.", "Elk City R.D.", "Fisher River R.D.", "Fortine R.D.",
                  "Glacier View R.D.", "Hungry Horse R.D.", "Jefferson R.D.", "Libby R.D.",
                  "Nine Mile R.D.","Plains R.D.", "Red River R.D.", "Rexford R.D.",
                  "Spotted Bear R.D.", "Swan lake R.D.", "Tally Lake R.D.",  "Superior R.D.",
                  "Thompson Falls R.D.", "Yaak R.D.")

all_areas_alt <- xyplot(lnr ~ lnxt | ID, data = surveyareas, panel = panelformat.2, as.table = TRUE,
                        layout = c(5, 6), skip = c(rep(FALSE, 7), TRUE, rep(FALSE, 22)),
                        strip = strip.custom(factor.levels = strip.titles, bg = "white",
                                             par.strip.text = list(col = "black", cex = 0.50)),
                        ylab = list(label = expression(
                          bold(paste(log[e], " (", bolditalic(n[t + 1] / n[t]), " )")))),
                        xlab = list(label = expression(
                          bold(paste(log[e], bolditalic(" n"[t]))))))

print(all_areas_alt)
```


Beaverhead N.F.
Deerlodge N.F.
Flathead N.F.
Gallatin N.F.
Kootenai N.F.
Lolo N.F.
Nez Perce N.F.
Blackfeet I.R.
Flathead I.R.
Cabinet R.D.
Elk City R.D.
Fisher River R.D.
Fortine R.D.
Glacier View R.D.
Hungry Horse R.D.
Jefferson R.D.
Libby R.D.
Nine Mile R.D.
Plains R.D.
Red River R.D.
Rexford R.D.
Spotted Bear R.D.
Swan lake R.D.
Tally Lake R.D.
Superior R.D.
Thompson Falls R.D.
Yaak R.D.
$\log_e (n_{t+1}/n_t)$
$\log_e n_t$