11: Regression

John H Maindonald

September 6, 2015

```
fig11.1 <- function(){
## ---- plt-roller ----
plot(depression ~ weight, data=roller)
}
```

```
fig11.2 <- function(){
## ---- pltWline ----
plot(depression ~ weight, data=roller)
roller.lm <- lm(depression ~ weight, data=roller)
# For a line through the origin, specify
# depression ~ 0 + weight
abline(roller.lm)
}
```

```
fig11.3 <- function(){
## ---- fVSmTime ----
xyplot(timef~time, data=nihills, aspect=1,
       type=c("p","r"))
}
```

```
fig11.4 <- function(){
## ---- mfdensity ----
## Simplified code
densityplot(~ time+timef, data=nihills,
            ylab="Time (h)", auto.key=TRUE)
}
```

```
fig11.5 <- function(){
print("Run the separate functions fig11.5A() and fig11.5B()")
}
fig11.5A <- function(){
```

```
## ---- rec-logmf ----
xyplot(timef~time, data=nihills,
       scales=list(log=10),
       aspect=1, type=c("p","r"))
}
fig11.5B <- function(){
## ---- skewtime-log ----
densityplot(~ log(time)+log(timef), data=nihills,
           ylab="Time (h)", auto.key=TRUE)
}
```

```
fig11.6 <- function(){
print("Run the separate functions fig11.6A() and fig11.6B()")
}
fig11.6A <- function(){
## Panel A
plot(resid(mftime.lm)~time, data=nihills)
mtext(side=3, line=0.5, "A: Residuals, unlogged data")
}
fig11.6B <- function(){
## Panel B
plot(resid(mflogtime.lm) ~ log(time), data=nihills)
mtext(side=3, line=0.5, "B: Residuals, logged data")
}
```

```
fig11.7 <- function(){
plot(mftime.lm, cex.caption=0.8)
}
```

```
fig11.8 <- function(){
plot(mflogtime.lm, cex.caption=0.8)
}
```

```
fig11.9 <- function(){
## ---- simscat ----
gph <- plotSimScat(obj=mftime.lm, show="residuals",
                   type=c("p","smooth"),
                   layout=c(4,1))
update(gph, xlab="Time (h) for males",
      ylab="Residuals")
}
```

```
suppfig11.1 <- function(){
## ---- simdiag2 ----
plotSimDiags(obj=mftime.lm, which=2, layout=c(4,1))
}
```

```
suppfig11.2 <- function(){
## ---- simdiag3 ----
plotSimDiags(obj=mftime.lm, which=3, layout=c(4,1))
}
```

```
suppfig11.3 <- function(){
plotSimDiags(obj=mftime.lm, which=5, layout=c(4,1))
}
```

```
suppfig11.4 <- function(){
## ---- mftime-lm ----
mftime.lm <- lm(timef ~ time, data=nihills)
## ---- mftime-sims ----
gph <- plotSimScat(mftime.lm, layout=c(4,1))
update(gph, xlab="Male record times (h)",
       ylab="Female record times (h)")
}
```

```
fig11.10 <- function(){
print("Run the separate functions fig11.10A() and fig11.10B()")
}
fig11.10A <- function(){
## ---- tomato-aov ----
## Analysis of variance: tomato data (from DAAG)
tomato.aov <- aov(weight ~ trt, data=tomato)
## ---- termplot-aov-wt ----
## Panel A: Use weight as outcome variable
tomato.aov <- aov(weight ~ trt, data=tomato)
termplot(tomato.aov, xlab="Treatment",
         ylab="Partial for treatment",
         partial.resid=TRUE, se=TRUE, pch=16)
mtext(side=3, line=0.5, "A: weight", adj=0)
}
fig11.10B <- function(){
## ---- lev-tomato ----
lev <- c("Water", "A", "B", "C")
tomato[, "trt"] <- factor(rep(lev, rep(6,4)),
```

```
                          levels=lev)
## ---- termplot-aov-logwt ----
## Panel B: Use log(weight) as outcome variable
logtomato.aov <- aov(log(weight) ~ trt, data=tomato)
termplot(logtomato.aov, xlab="Treatment",
         ylab="Partial for treatment",
         partial.resid=TRUE, se=TRUE, pch=16)
mtext(side=3, line=0.5, "B: log(weight)", adj=0)
}
```

```
fig11.11 <- function(){
print("Run the separate functions fig11.16A() and fig11.16B()")
}
fig11.11A <- function(){
## ---- scatter-ni ----
## Unlogged data
library(lattice)
## Scatterplot matrix; unlogged data
splom(~nihills)
}
fig11.11B <- function(){
## ---- scatter-logni ----
lognihills <- log(nihills)
names(lognihills) <- paste0("l", names(nihills))
## Scatterplot matrix; log scales
splom(~ lognihills)
}
```

```
fig11.12 <- function(){
## ---- nireg-slope ----
nihills$gradient <- with(nihills, climb/dist)
lognihills <- log(nihills)
lognam <- paste0("l", names(nihills))
names(lognihills) <- lognam
lognigrad.lm <- lm(ltime ~ ldist + lgradient,
                   data=lognihills)
round(coef(lognigrad.lm),3)
## ---- tplot-ni ----
## Plot the terms in the model
termplot(lognigrad.lm, col.term="gray", partial=TRUE,
         col.res="black", smooth=panel.smooth)
}
```

```
fig11.13 <- function(){
## ---- bsnVary ----
set.seed(37)   # Use to reproduce graph shown
DAAG::bsnVaryNvar(m=100, nvar=3:50, nvmax=3)
}
```

```
fig11.14 <- function(){
## ---- Elec-spm ----
if(require('Ecdat', quietly=TRUE)){
  data(Electricity)
  if(requireNamespace('car'))
  car::spm(Electricity, smooth=TRUE, reg.line=NA,
           col=adjustcolor(rep("black",3), alpha.f=0.3)) else
      plot(Electricity, col=adjustcolor(rep("black",3), alpha.f=0.3))
} else print("Dataset Electricity is not available")
}
```

```
fig11.15 <- function(){
## ---- spm-cost-q ----
varlabs <- c("log(cost)", "log(q)")
if(requireNamespace('car'))
car::spm(log(Electricity[,1:2]), var.labels=varlabs,
    smooth=TRUE, reg.line=NA,
    col=adjustcolor(rep("black",3), alpha.f=0.5)) else
    plot(Electricity, col=adjustcolor(rep("black",3), alpha.f=0.3))
}
```

```
fig11.16 <- function(){
## ---- elec-me ----
elec.lm <- lm(log(cost) ~ log(q)+pl+sl+pk+sk+pf+sf,
              data=Electricity)
## ---- elec-me-tplot ----
termplot(elec.lm, partial=T, smooth=panel.smooth,
         transform.x=TRUE)
}
```

```
fig11.17 <- function(){
print("Run the separate functions fig11.17A() and fig11.17B()")
}
fig11.17A <- function(){
## ---- bronchitA ----
## ---- bronchit-ylim ----
```

```
ylim <- range(bronchit$poll)+c(0,2.5)
## Panel A
colr <- adjustcolor(c("red","blue"), alpha=0.5)
plot(poll ~ cig,
     xlab="# cigarettes per day", ylab="Pollution",
     col=colr[r+1], pch=(3:2)[r+1], data=bronchit,
     ylim=ylim)
legend(x="topright",
       legend=c("Non-sufferer","Sufferer"),
       ncol=2, pch=c(3,2), col=c(2,4), cex=0.8)
mtext(side=3, line=1.0,
      expression("A: Untransformed "*italic(x)*"-scale"), adj=0)
}
fig11.17B <- function(){
## ---- bronchitB ----
## ---- bronchit-ylim ----
ylim <- range(bronchit$poll)+c(0,2.5)
## Panel B
plot(poll ~ log(cig+1), col=c(2,4)[r+1], pch=(3:2)[r+1],
     xlab="log(# cigarettes per day + 1)", ylab="", data=bronchit, ylim=ylim)
xy1 <- with(subset(bronchit, r==0), cbind(x=log(cig+1), y=poll))
xy2 <- with(subset(bronchit, r==1), cbind(x=log(cig+1), y=poll))
if(requireNamespace('KernSmooth', quietly=TRUE)){
est1 <- bkde2D(xy1, bandwidth=c(0.7, 3))
est2 <- bkde2D(xy2, bandwidth=c(0.7, 3))
lev <- pretty(c(est1$fhat, est2$fhat),4)
contour(est1$x1, est1$x2, est1$fhat, levels=lev, add=TRUE, col=2)
contour(est2$x1, est2$x2, est2$fhat, levels=lev, add=TRUE, col=4, lty=2)
}
legend(x="topright", legend=c("Non-sufferer","Sufferer"), ncol=2, lty=1:2,
       col=c(2,4), cex=0.8)
mtext(side=3, line=1.0, expression("B: Log transformed "*italic(x)*"-scale"),
      adj=0)
}
```

```
fig11.18 <- function(){
## ---- cig2-glm ----
cig2.glm <- glm(r ~ log(cig+1) + poll, family=binomial, data=bronchit)
## ---- cig2-tplot ----
termplot(cig2.glm)
}
```

```
figset11 <- function(){
  if(!require(DAAG, quietly=TRUE))stop('DAAG must be installed')
```

```
  if(!require(KernSmooth, quietly=TRUE))
    print('KernSmooth is not installed. Fig 11.17B will not show contours')
  if(!require(splines, quietly=TRUE))stop('splines must be installed')
  }
```

```
figset11()
if(!exists("mftime.lm")) mftime.lm <- lm(timef ~ time, data=nihills)
if(!exists("mflogtime.lm"))
  mflogtime.lm <- lm(log(timef) ~ log(time), data=nihills)
  check4bronchit <- exists('bronchit')
  if(!check4bronchit)if(!require(DAAGviz))stop("Dataset 'bronchit' is not available"
if(!('rfac' %in% names(bronchit)))bronchit <-
within(bronchit,
       rfac <- factor(r, labels=c("abs","pres")))
```

```
fig11.1()
```

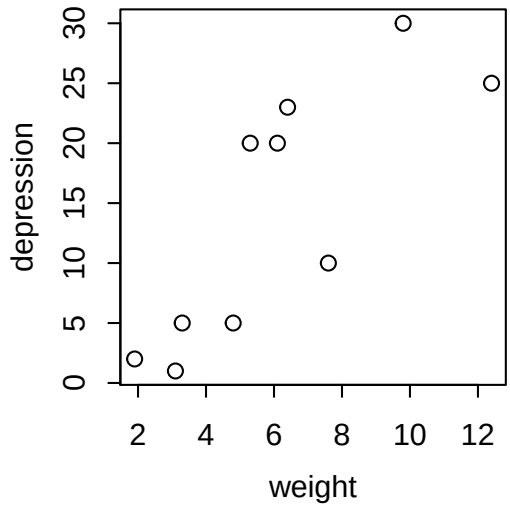


Figure 1: Plot of `depression` versus `weight`, using data from the data frame `roller` in the *DAAG* package.

```
fig11.2()
```

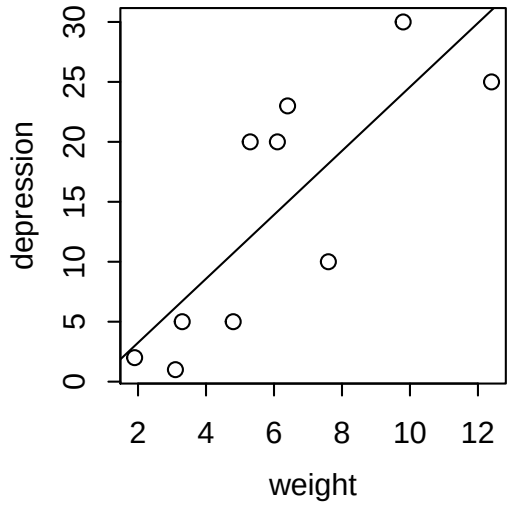


Figure 2: This repeats Figure 1, now adding a fitted line.

```
fig11.3()
```

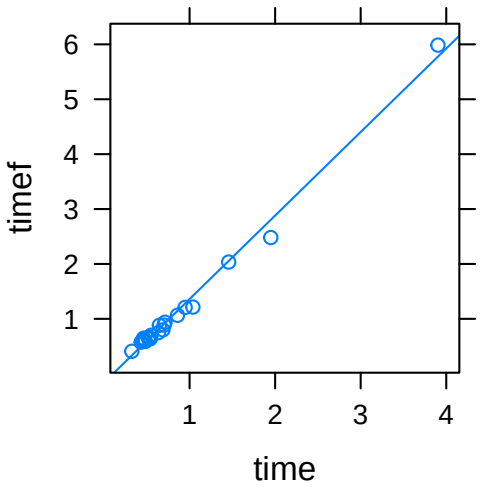


Figure 3: Data are for Northern Ireland hill races. Record times are compared – females versus males. A least squares line is added.

```
fig11.4()
```

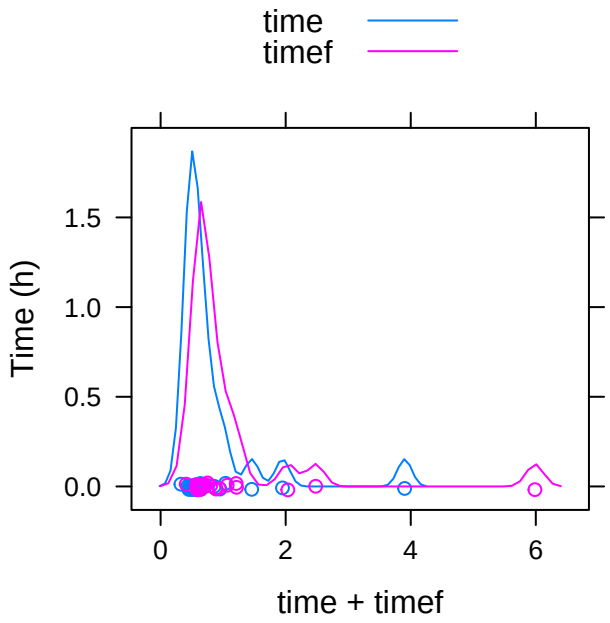


Figure 4: Density plot comparison of times between males and females. Note the long tail out to the right, already obvious from the diagnostic plots.

```
fig11.5A()
fig11.5B()
```

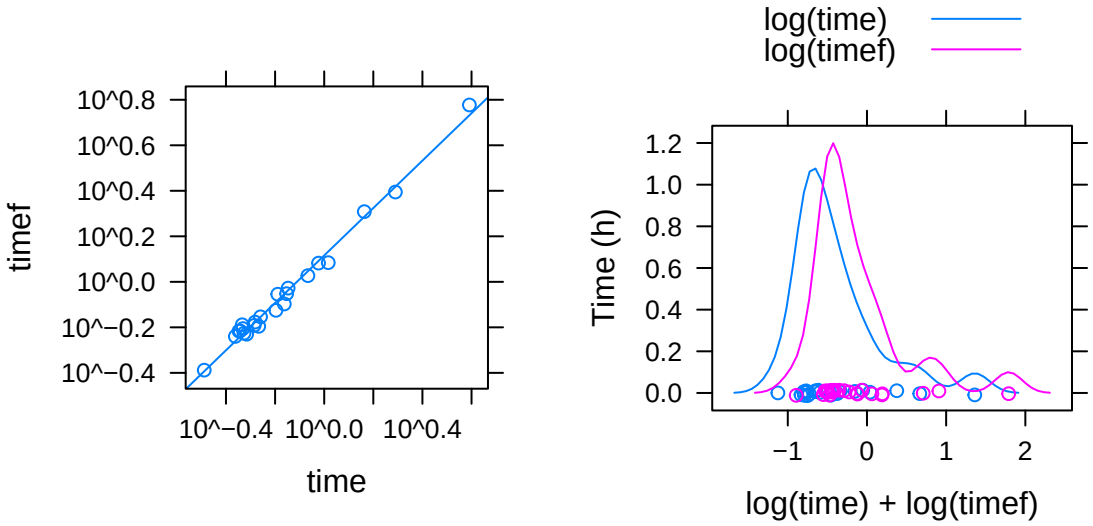


Figure 5: In these graphs, female and male times are shown on log scales. Panel A shows a density plot comparison. Panel B plots female versus male times.

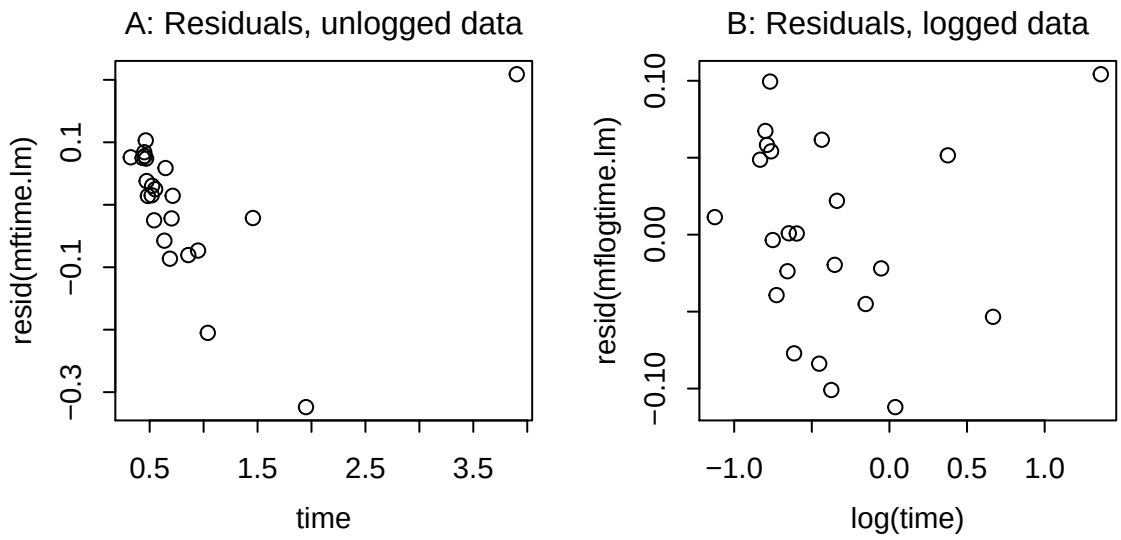


Figure 6: In Panel A, residuals from the line for the unlogged data have been plotted against male times. Panel B repeats the same type of plot, now for the regression for the logged data.

```
fig11.7()
```

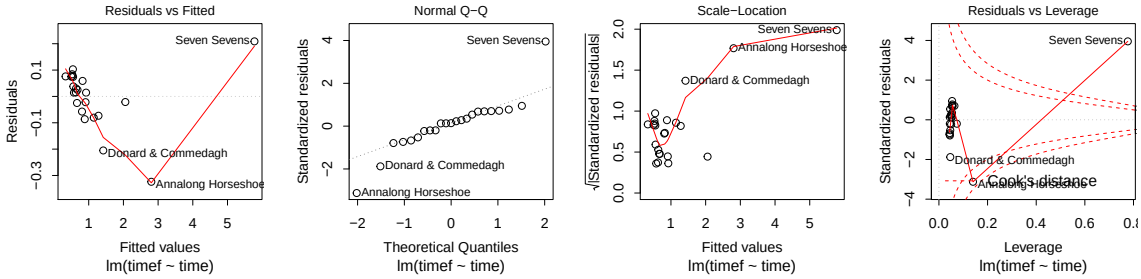


Figure 7: Diagnostic plots from the regession of `log(timef)` on `log(time)`.

```
fig11.8()
```

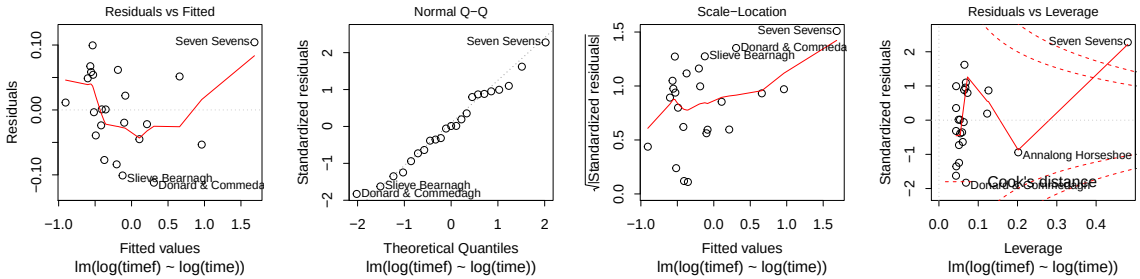


Figure 8: Diagnostic plots from the regession of `log(timef)` on `log(time)`.

```
fig11.9()
```

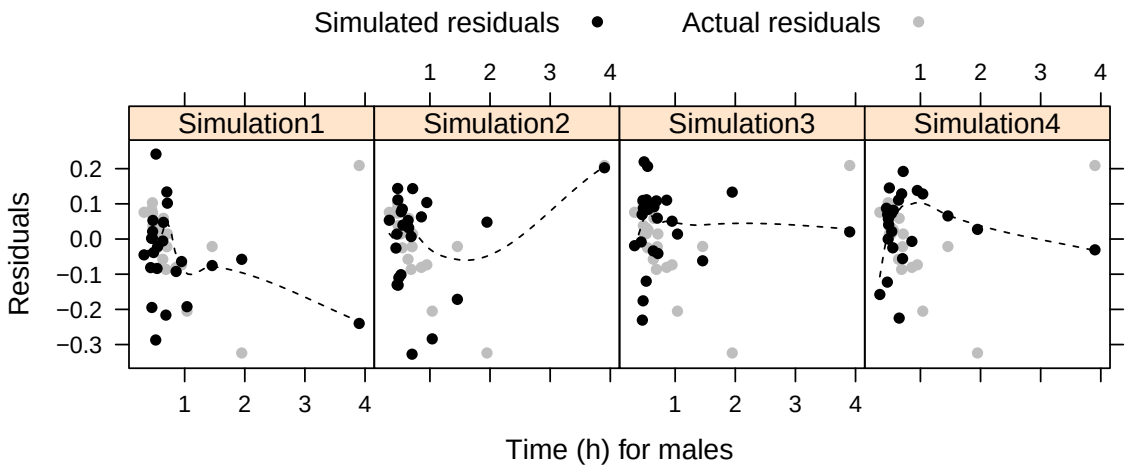
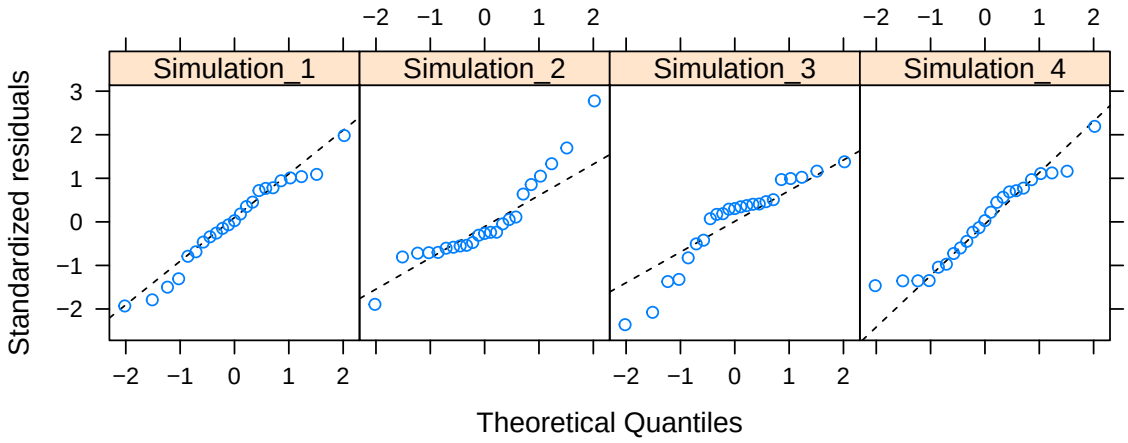


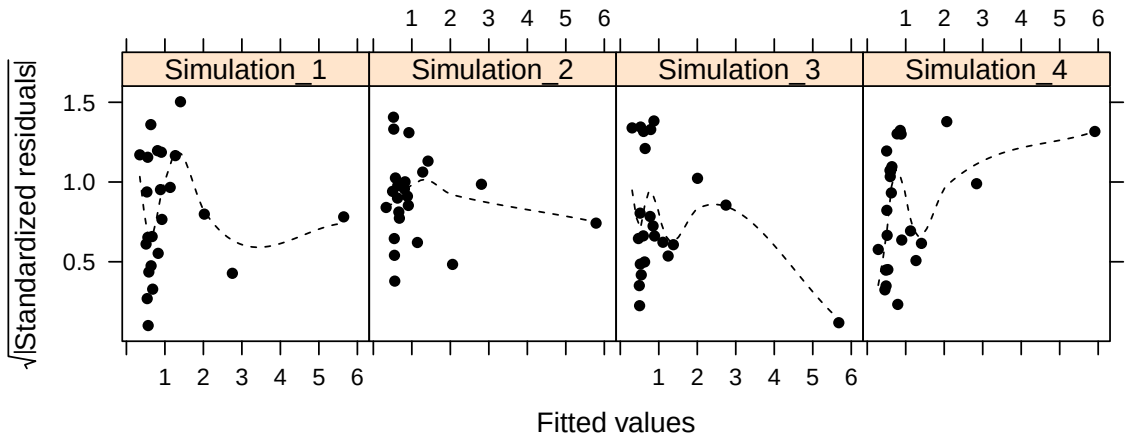
Figure 9: The plots are four simulations of residuals. The coefficients used, and the standard deviation, are from the fitted least squares line.

```
suppfig11.1()
```



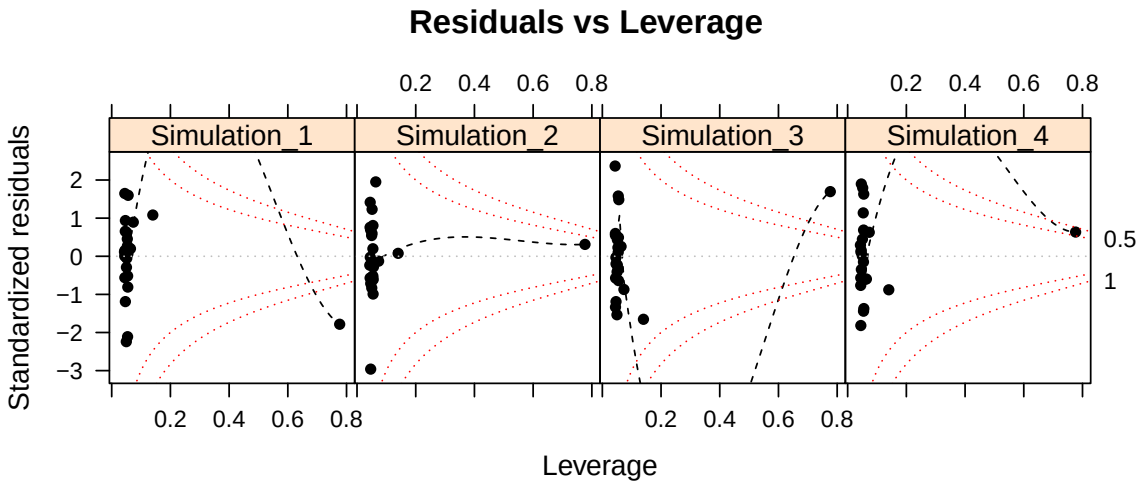
Supplementary Figure 1: Normal probability plots for four sets of simulated data.

```
suppfig11.2()
```



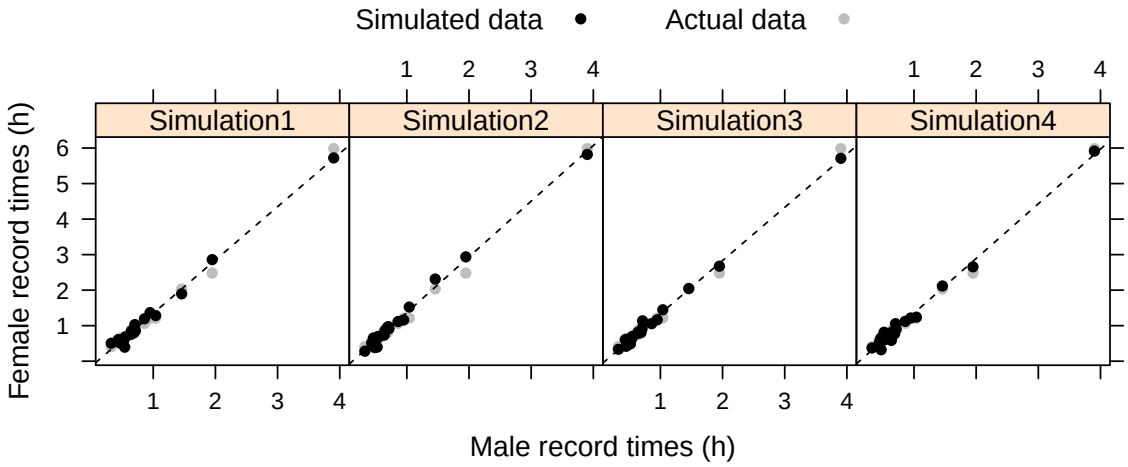
Supplementary Figure 2: These plots, here with simulated data, are designed to check for change in variance as the fitted values change.

```
suppfig11.3()
```



Supplementary Figure 3: Scale-location plots for four sets of simulated data.

```
suppfig11.4()
```



Supplementary Figure 4: The plots are four simulations of points. The coefficients used, and the standard deviation, are from the fitted least squares line. The gray points are the data values, which are of course the same in all 4 plots.

```
fig11.10A()
fig11.10B()
```

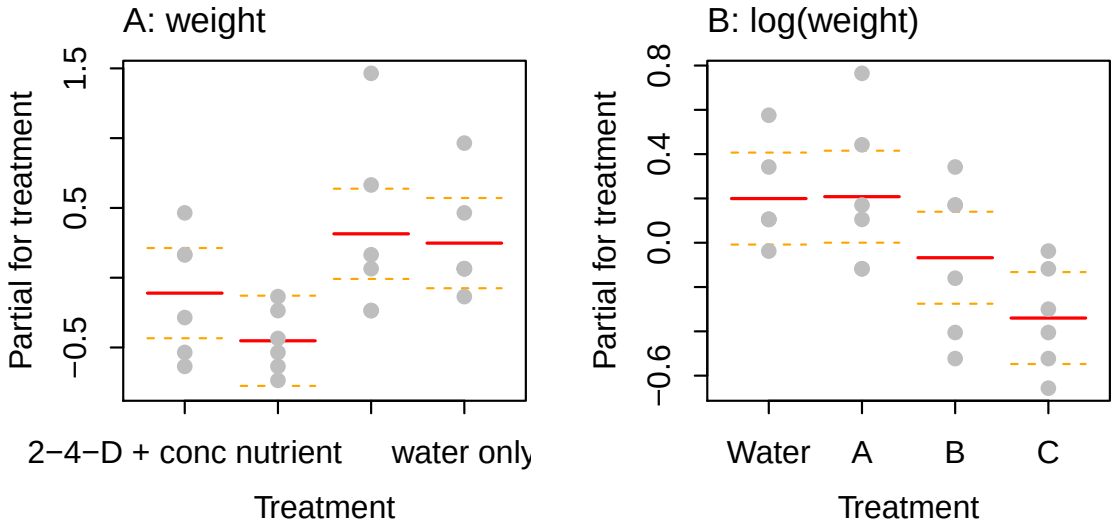


Figure 10: Termplot summary of the one-way analysis of variance result: (a) for the analysis that uses weights as the outcome variable, and (b) for the analysis that works with `log(weight)`

```
fig11.11A()
fig11.11B()
```

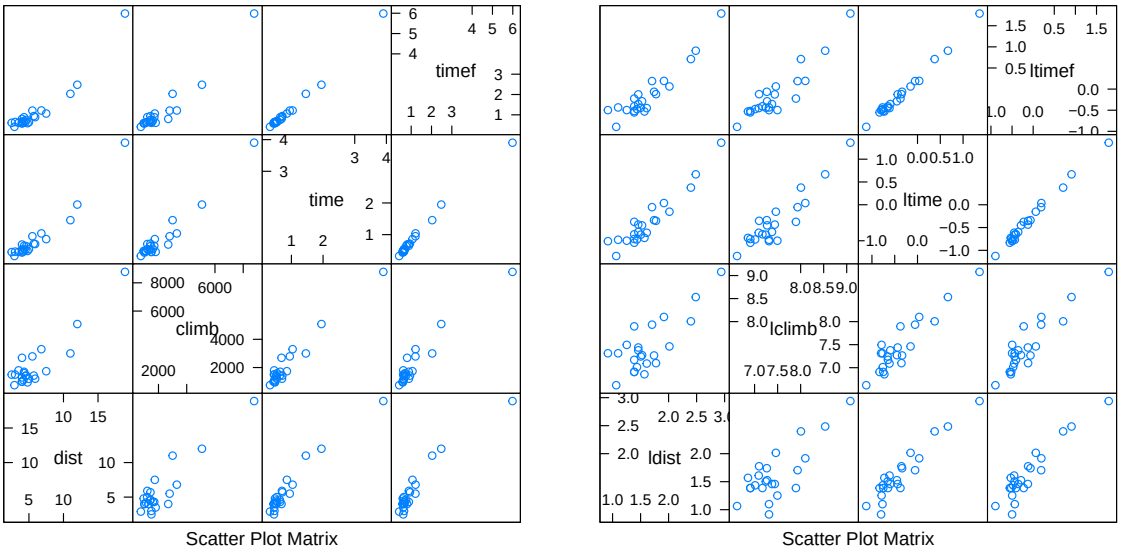


Figure 11: Scatterplot matrices for the Northern Ireland mountain racing data. In the right panel, code has been added that shows the correlations.

```
fig11.12()
```

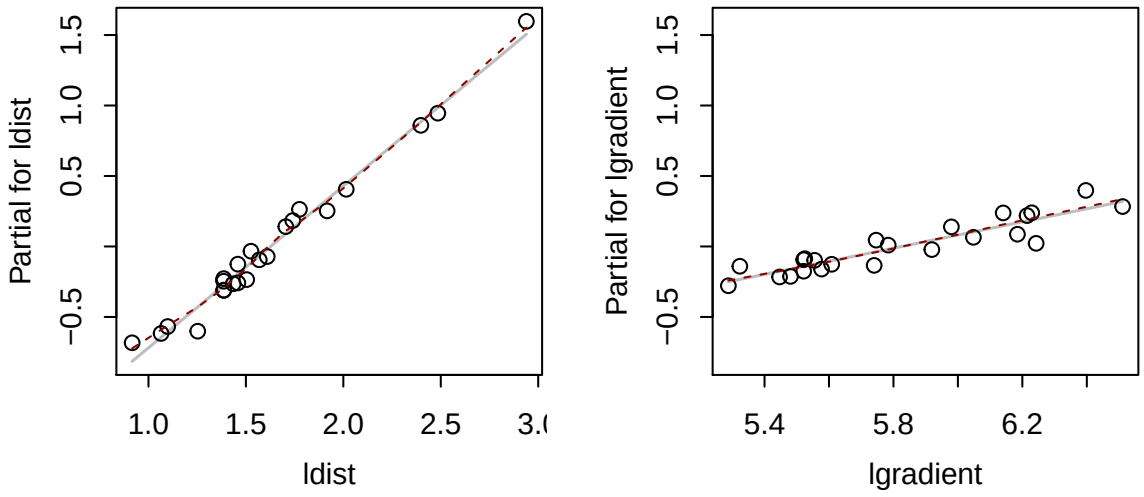


Figure 12: The vertical scales in both "term plot" panels show log(time), centered to a mean of zero. The partial residuals in the left panel are for the effect of `ldist`, while those in the right panel are for the effect of `lgradient`, i.e., `log(climb/dist)`. Smooth curves (dashes) have been passed through the points.

```
if(!requireNamespace("quantreg", quietly=TRUE))
    print("As quantreg is not available, trend curve will be omitted.")
fig11.13()
```

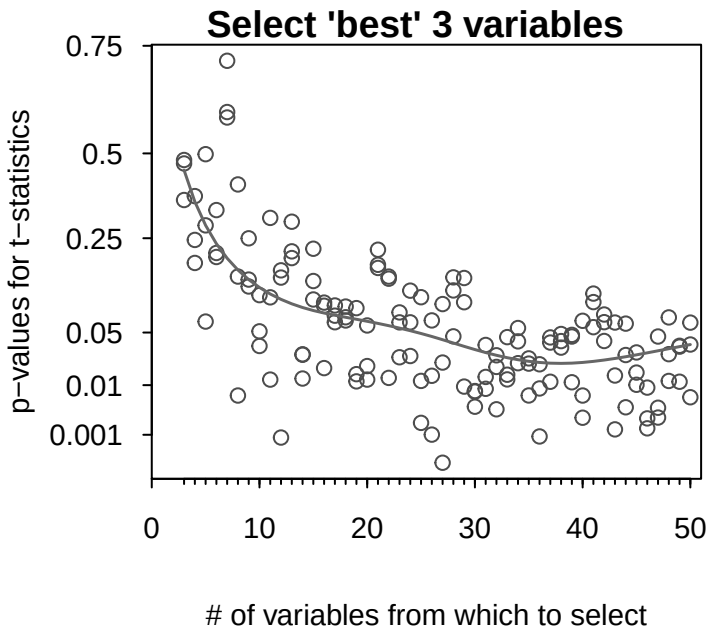


Figure 13: p-values, versus number of variables available for selection, when the "best" 3 variables were selected by exhaustive search. The fitted line estimates the median p-value.

```
check4Elec <- exists("Electricity")
if(!check4Elec){
    if(!require(Ecdat, quietly=TRUE))stop("Dataset Electricity is not available.")
}
check4Elec <- TRUE
data(Electricity)
```

```
if(check4Elec)fig11.14()
```

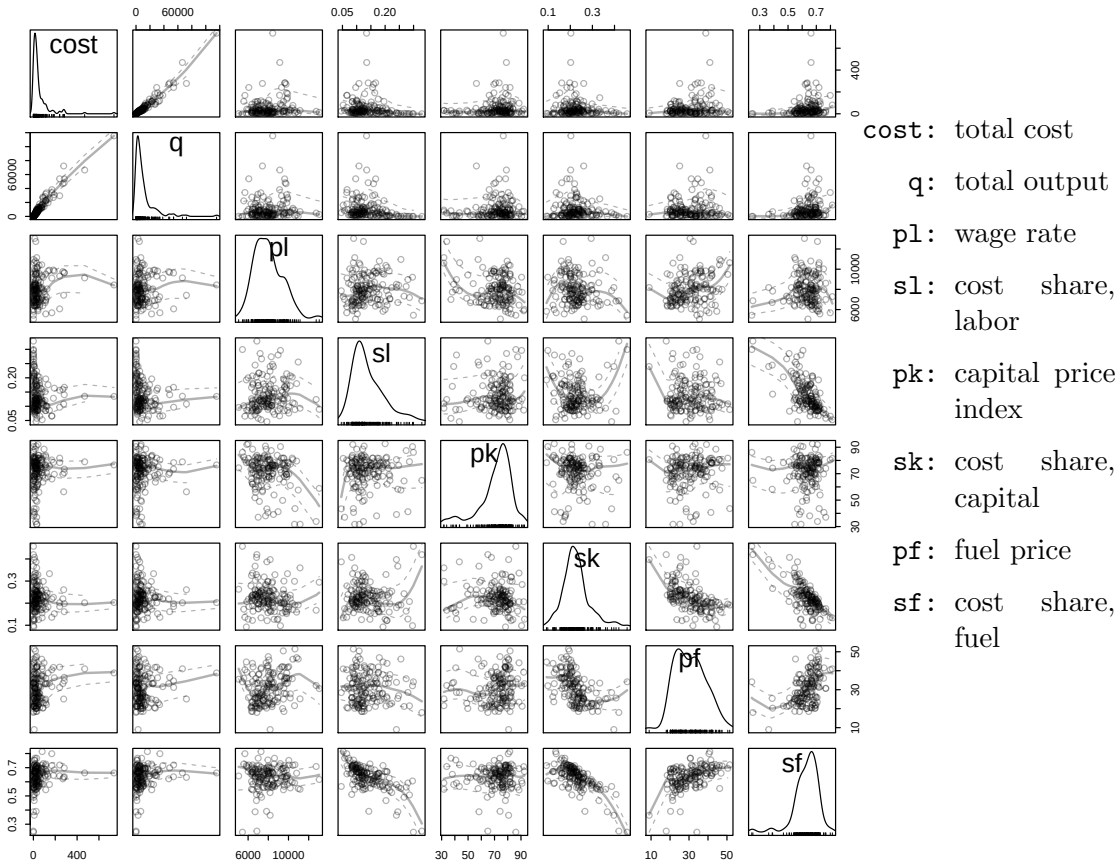


Figure 14: Scatterplot matrix, for the variables in the data set `Electricity`, in the *Ecdat* package. Density plots are shown in the diagonal.

```
if(check4Elec)fig11.15()
```

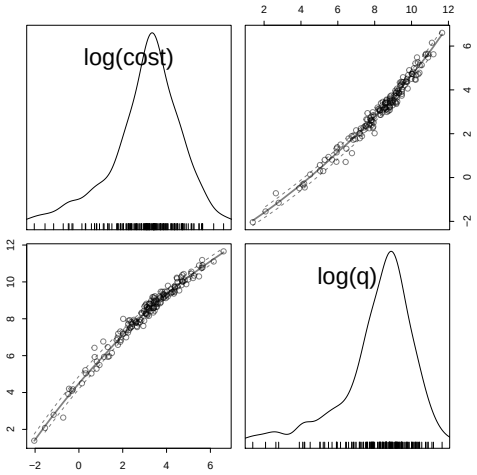


Figure 15: Scatterplot matrix for the logarithms of the variables `cost` and `q`. Density plots are shown in the diagonal.

```
if(check4Elec)fig11.16()
```

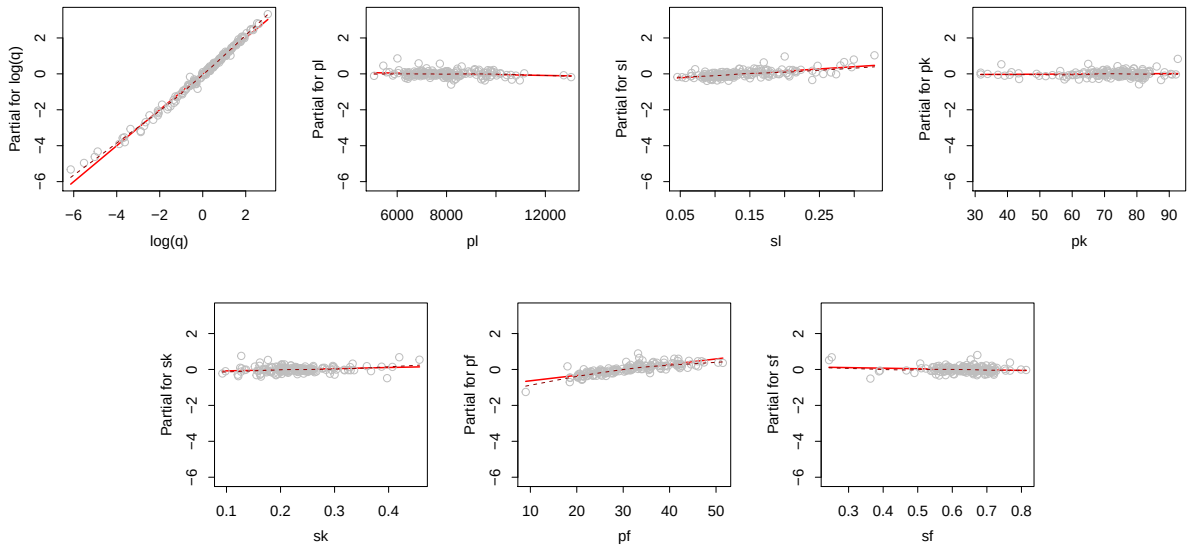


Figure 16: Termplot summary for the model that has been fitted to the `Electricity` dataset.

```
opar <- par(mar=c(4,4,2.5,0.6))

fig11.17A()
fig11.17B()
par(opar)
```

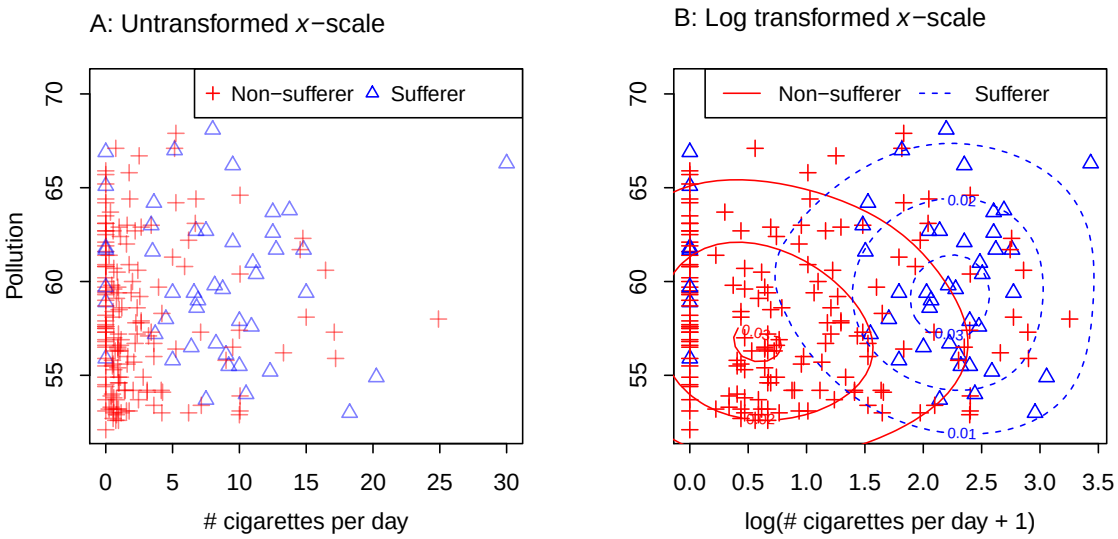


Figure 17: Panel A plots `poll` (pollution level) against `cig` (number of cigarettes per day). In panel B, the x-scale shows the logarithm of the number of cigarettes per day.

```
fig11.18()
```

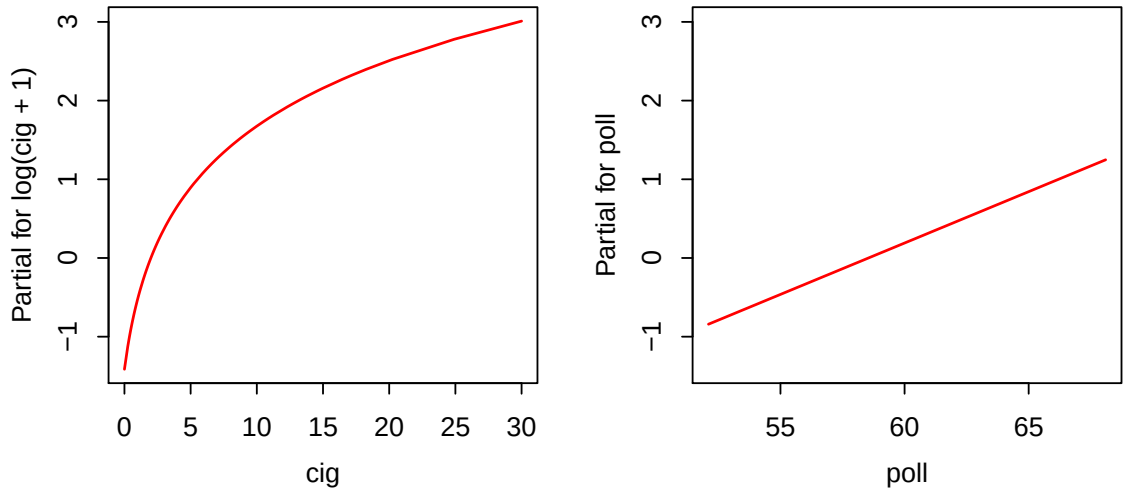


Figure 18: The panels show the contributions that the respective terms make to the fitted values (logit of probability of bronchitis), when the other term is held constant.