8: Regression

John H Maindonald

December 7, 2018

```
doFigs <- FALSE
fig8.1 <- function(){
## ---- plt-roller ----
plot(depression ~ weight, data=roller)
}
```

```
fig8.2 <- 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)
}
```

```
fig8.3 <- function(){
## ---- fVSmTimeAB ----
def.par <- par(no.readonly = TRUE)
layout(matrix(c(1,3,0,0,2,4), 2, 3), respect = TRUE,
       widths=c(7.5,1,7.5), heights=c(7.15,2))
xlim <- range(DAAG::nihills$time)+c(-0.1,0.1)
par(mar=c(0.2,3,2,0), bty="o", cex.axis=1.1)
plot(timef~time, data=DAAG::nihills, xlim=xlim, xaxt='n',
     ylab="", fg="gray", main=" ")
abline(lm(timef~time, data=DAAG::nihills), col=2)
mtext(side=2,line=2.25,"Female times", las=0)
mtext(side=3,line=0.5,"A: Untransformed scales", adj=0)
plot(timef~time, data=DAAG::nihills, xlim=xlim, xaxt='n',
     log='xy', fg="gray", ylab="", main=" ")
abline(lm(log10(timef)~log10(time), data=DAAG::nihills), col=2)
mtext(side=2,line=2.25,"Female times (log scale)", las=0)
```

```
mtext(side=3,line=0.5,"B: Logarithmic scales", adj=0)
par(mar=c(3,2.75,0,0), bty="n")
pars = list(boxwex = 4.0, staplewex = 0.5, outwex = 0.5)
boxplot(DAAG::nihills$time, horizontal=T, xlim=xlim,
        pars=pars, at=2, axes=0)
axis(1, lwd = 0, lwd.ticks = 1)
mtext(side=1,line=2,"Male times")
boxplot(DAAG::nihills$time, horizontal=T, xlim=xlim,
        log='x', pars=pars, at=2, axes=0)
axis(1, lwd = 0, lwd.ticks = 1)
mtext(side=1,line=2,"Male times (log scale)")
par(def.par)
}
```

```
fig8.4 <- function(){
print("Run the separate functions fig8.4A() and fig8.4B()")
}
fig8.4A <- function(){
## Panel A
plot(resid(mftime.lm)~time, data=nihills)
mtext(side=3, line=0.5, "A: Residuals, unlogged data")
}
fig8.4B <- function(){
## Panel B
plot(resid(mflogtime.lm) ~ log(time), data=nihills)
mtext(side=3, line=0.5, "B: Residuals, logged data")
}
```

```
fig8.5 <- function(){
plot(mftime.lm, cex.caption=0.8)
}
```

```
fig8.6 <- function(){
plot(mflogtime.lm, cex.caption=0.8)
}
```

```
fig8.7 <- 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")
}
```

```
suppfig8.1 <- function(){
## ---- simdiag2 ----
plotSimDiags(obj=mftime.lm, which=2, layout=c(4,1))
}
```

```
suppfig8.2 <- function(){
## ---- simdiag3 ----
plotSimDiags(obj=mftime.lm, which=3, layout=c(4,1))
}
```

```
suppfig8.3 <- function(){
plotSimDiags(obj=mftime.lm, which=5, layout=c(4,1))
}
```

```
suppfig8.4 <- 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)")
}
```

```
fig8.8 <- function(){
print("Run the separate functions fig8.10A() and fig8.10B()")
}
fig8.8A <- 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)
}
fig8.8B <- 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)
}
```

```
fig8.9 <- function(){
print("Run the separate functions fig8.11A() and fig8.11B()")
}
fig8.9A <- function(){
## ---- scatter-ni ----
## Unlogged data
library(lattice)
## Scatterplot matrix; unlogged data
splom(~nihills)
}
fig8.9B <- function(){
## ---- scatter-logni ----
lognihills <- log(nihills)
names(lognihills) <- paste0("l", names(nihills))
## Scatterplot matrix; log scales
splom(~ lognihills)
}
```

```
fig8.10 <- 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)
}
```

```
fig8.11 <- function(){
## ---- bsnVary ----
set.seed(37)   # Use to reproduce graph shown
DAAG::bsnVaryNvar(m=100, nvar=3:50, nvmax=3)
}
```

```
fig8.12 <- function(){
## ---- Elec-spm ----
if(require('Ecdat', quietly=TRUE)){
  data(Electricity)
  if(requireNamespace('car'))
  car::spm(Electricity, smooth=TRUE, regLine=FALSE,
           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")
}
```

```
fig8.13 <- function(){
## ---- spm-cost-q ----
varlabs <- c("log(cost)", "log(q)")
if(requireNamespace('car'))
car::spm(log(Electricity[,1:2]), var.labels=varlabs,
    smooth=TRUE, regLine=FALSE,
    col=adjustcolor(rep("black",3), alpha.f=0.5)) else
    plot(Electricity, col=adjustcolor(rep("black",3), alpha.f=0.3))
}
```

```
fig8.14 <- 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)
}
```

```
fig8.15 <- function(){
print("Run the separate functions fig8.17A() and fig8.17B()")
}
fig8.15A <- function(){
## ---- bronchitA ----
## ---- bronchit-ylim ----
```

```
ylim <- range(bronchit$poll)+c(0,2.5)
## Panel A
colr <- adjustcolor(c("red","blue"), alpha=0.5)
plot(poll ~ cig,
     xlab="# cigarettes per day", ylab="Pollution",
     col=colr[r+1], pch=(3:2)[r+1], data=bronchit,
     ylim=ylim)
legend(x="topright",
       legend=c("Non-sufferer","Sufferer"),
       ncol=2, pch=c(3,2), col=c(2,4), cex=0.8)
mtext(side=3, line=1.0,
      expression("A: Untransformed "*italic(x)*"-scale"), adj=0)
}
fig8.15B <- function(){
## ---- bronchitB ----
## ---- bronchit-ylim ----
ylim <- range(bronchit$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)
}
```

```
fig8.16 <- function(){
## ---- cig2-glm ----
cig2.glm <- glm(r ~ log(cig+1) + poll, family=binomial, data=bronchit)
## ---- cig2-tplot ----
termplot(cig2.glm)
}
```

```
fig8.17 <- function(){
## ---- smooth-ohms ----
```

```
fig8.18A <- function(){
    if(!exists("eyeAmpM.gam"))
eyeAmpM.gam <- mgcv::gam(amp ~ s(x,y), data=subset(eyeAmp, Sex=="m"))
        if(!exists("eyeAmpF.gam"))
eyeAmpF.gam <- mgcv::gam(amp ~ s(x,y), data=subset(eyeAmp, Sex=="f"))
lims <- range(c(predict(eyeAmpF.gam), predict(eyeAmpM.gam)))
mgcv::vis.gam(eyeAmpM.gam, plot.type='contour', color="cm", zlim=lims, main="")
}
fig8.18B <- function(){
    if(!exists("eyeAmpM.gam"))
    eyeAmpM.gam <- mgcv::gam(amp ~ s(x,y), data=subset(eyeAmp, Sex=="m"))
    if(!exists("eyeAmpF.gam"))
    eyeAmpF.gam <- mgcv::gam(amp ~ s(x,y), data=subset(eyeAmp, Sex=="f"))
lims <- range(c(predict(eyeAmpF.gam), predict(eyeAmpM.gam)))
mgcv::vis.gam(eyeAmpF.gam, plot.type='contour', color="cm", zlim=lims, main="")
}
fig8.18 <- function(){
print("Run the separate figures fig8.18A() and fig8.18B()")
}
```

```
## Plot points
plot(ohms ~ juice, data=fruitohms)
## Add smooth curve, using default
## smoothing window
with(fruitohms,
     lines(lowess(ohms ~ juice), col="gray", lwd=2))
}
```

```
figset8 <- 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')
  }
```

```
figset8()
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")))
```

```
fig8.1()
```


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

```
fig8.2()
```


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

```
fig8.3()
```


Figure 3: Graphs compare female with male record times, for Northern Ireland hill races. Least squares lines are added, and marginal boxplots are shown on the horizontal axis. Panel A has untransformed scales, while Panel B has log transformed scales. For the code, see the script file for this chapter.

Figure 4: 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.

```
fig8.5()
```


Figure 5: Diagnostic plots from the regession of `log(timef)` on `log(time)`.

```
fig8.6()
```


Figure 6: Diagnostic plots from the regession of `log(timef)` on `log(time)`.

```
fig8.7()
```


Figure 7: The plots are four simulations of residuals. The coefficients used, and the standard deviation, are from the fitted least squares line.

```
suppfig8.1()
```

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

```
suppfig8.2()
```

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

```
suppfig8.3()
```

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

```
suppfig8.4()
```

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.

```
fig8.8A()
fig8.8B()
```

Figure 8: 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)`

```
fig8.9A()
fig8.9B()
```

Figure 9: Scatterplot matrices for the Northern Ireland mountain racing data. In the right panel, code has been added that shows the correlations.

```
fig8.10()
```

Figure 10: 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.")
fig8.11()
```

Figure 11: 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)
```

`cost`: total cost

`q`: total output

`pl`: wage rate

`sl`: cost share, labor

`pk`: capital price index

`sk`: cost share, capital

`pf`: fuel price

`sf`: cost share, fuel

```
if(check4Elec)fig8.12()
```

Figure 12: Scatterplot matrix, for the variables in the data set `Electricity`, in the *Ecdat* package. Density plots are shown in the diagonal.

```
if(check4Elec)fig8.13()
```

Figure 13: Scatterplot matrix for the logarithms of the variables `cost` and `q`. Density plots are shown in the diagonal.

```
if(check4Elec)fig8.14()
```

Figure 14: Termplot summary for the model that has been fitted to the `Electricity` dataset.

```
fig8.17()
```

Figure 17: Resistance in ohms is plotted against apparent juice content. A smooth curve (in gray) has been added, using the `lowess` smoother. The width of the smoothing window was the default fraction $f = \frac{2}{3}$ of the range of values of the x-variable.

```
opar <- par(mfrow=c(1,2))
fig8.18A()
mtext(side=3, line=0.5, adj=0, "A: Response amplitudes, Males")
fig8.18B()
mtext(side=3, line=0.5, adj=0, "B: Response amplitudes, Females")
par(opar)
```

Figure 18: Estimated contours of left eye responses to visual stimulae, projected onto the plane.

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

fig8.15A()
fig8.15B()
par(opar)
```

Figure 15: 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.

```
fig8.16()
```

Figure 16: 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.