

1: Key Ideas and Issues

John H Maindonald

August 19, 2014

Ideas and issues illustrated by the graphs in this vignette

Ideas and issues that the graphs given here are designed to illustrate can be summarized under the headings:

- Data Issues
 - Data Exploration
 - Source/target issues
 - Data validity, accuracy and relevance
- Models and Model Assumptions
 - Model assumptions
 - Least squares, maximum likelihood and Bayesian estimation
 - Simulation from an assumed model
 - Model diagnostics
 - Weighting biases – problems for interpretation of model parameters

1 Code for Functions that Give the Figures

```
fig1.1 <-  
function (form = depression ~ weight, data = roller, ...)  
{  
  yvar <- all.vars(form)[1]  
  xvar <- all.vars(form)[2]  
  x <- data[, xvar]  
  y <- data[, yvar]  
  maxx <- max(x)  
  maxy <- max(y)  
  plot(form, data = roller, xlim = c(0, 1.04 * maxx), ylim = c(0,
```

```

    1.04 * maxy), xaxs = "i", yaxs = "i", ...,
    main="1.1: Depression vs weight")
}

```

```

fig1.2 <-
function ()
{
  print("Run the separate functions fig1.2A() and fig1.2B()")
}

```

```

fig1.2A <-
function ()
{
  require(MASS)
  plot(brain ~ body, data = mammals, pty = "s")
  mtext(side = 3, line = 0.5, adj = 0, "1.2A: Unlogged data")
}

```

```

fig1.2B <-
function ()
{
  library(MASS)
  plot(brain ~ body, data = mammals, log = "xy", pty = "s")
  mtext(side = 3, line = 0.5, adj = 0, "1.2B: Log scales on both axes")
}

```

```

fig1.3 <-
function ()
{
  opar <- par(mar=rep(0.6,4), oma=c(0,0,2,0))
  pairs(log(mammals), labels = c("log(body)", "log(brain)"))
  mtext(side=3, line=0.75, outer=TRUE, "1.3: Pairs plot")
}

```

```

fig1.4 <-
function (parset = simpleTheme(pch = 1:10, alpha = 0.6, cex = 1),
  fontsize = list(text = 14, points = 10))
{
  if (!is.null(parset))
    parset$fontsize <- fontsize
}

```

```

library(MASS)
droplevs <- fgl$type %in% c("Tabl", "Con")
usefgl <- droplevels(subset(fgl, !droplevs))
fgl.hat <- predict(lda(type ~ ., data = usefgl))
gph <- xyplot(fgl.hat$x[, 2] ~ fgl.hat$x[, 1],
              groups = usefgl$type,
              auto.key = list(columns = 2),
              xlab = "Axis 1", ylab = "Axis 2",
              aspect = 1, scales = list(tck = 0.4),
              par.settings = parset,
              title = "1.4: Plot of first two linear discriminant scores")

gph
}

```

```

fig1.5 <-
function ()
{
  opar <- par(mar=rep(0.5,4))
  if(!require(diagram))stop("Package 'diagram' must be installed")
  openplotmat(xlim = c(-0.1, 1.1))
  textellipse(mid=c(.5, .8), radx=0.6, rady=0.25,
              lab="Source", adj=c(.5,-2),
              box.col="gray95")
  textellipse(mid=c(.5, .7), radx=0.3, rady=0.1,
              lab="Source Sample", adj=c(.5,.5),
              box.col="gray90")
  textellipse(mid=c(.5, .2), radx=0.6, rady=0.25,
              lab="Target", adj=c(.5,-2),
              box.col="gray95")
  textellipse(mid=c(.5, .1), radx=0.3, rady=0.1,
              lab="Target Sample?", adj=c(.5,.5),
              box.col="gray90")
  par(opar)
}

```

```

fig1.6 <-
function ()
{
  library(DAAG)
  roller.obj <- lm(depression ~ weight, data = roller)
  yhat <- predict(roller.obj)
  ymax <- max(c(roller$depression, yhat))
  plot(depression ~ weight, data = roller, xlab = "Roller weight (t)",

```

```

        ylab = "Depression in lawn (mm)", pch = 4, xlim = c(0,
            max(roller$weight) * 1.01), ylim = c(0, ymax * 1.01),
        xaxs = "i", yaxs = "i", main="")
abline(roller.obj)
b <- summary(roller.obj)$coef
topleft <- par()$usr[c(1, 4)]
chw <- par()$cxy[1]
chh <- par()$cxy[2]
legend(topleft[1], topleft[2] + 0.25 * chh, pch = c(1, 4),
        legend = c("Fitted values", "Data values"), adj = 0,
        cex = 0.8, x.intersp = 0.8, y.intersp = 0.8, bty = "n")
df <- cbind(roller, above = as.numeric(roller$depression >
        yhat))
with(df, segments(weight, depression, weight, yhat, col = c("gray45",
        "black")[above + 1]))
n <- nrow(roller)
ns <- with(roller, min((1:n)[depression - yhat >= 0.75 *
        max(depression - yhat)]))
ypos <- 0.5 * (roller$depression[ns] + yhat[ns])
text(roller$weight[ns], ypos, "+ve residual", pos = 2, cex = 0.8)
points(roller$weight, yhat, pch = 1)
ns <- with(roller, (1:n)[depression - yhat == min(depression -
        yhat)][1])
ypos <- 0.5 * (roller$depression[ns] + yhat[ns])
text(roller$weight[ns], ypos, "-ve residual", pos = 4, cex = 0.8)
mtext(side=3, line=0.75,
        "1.6: Lawn roller plot + line & annotation")
}

```

```

fig1.7 <- function(){
  obj <- lm(depression ~ weight, data=roller)
  gph <- plotSimScat(obj, sigma=6.4, layout=c(4,1), aspect=1)
  gph <- update(gph, xlab="Roller weight (t)", ylab="Depression (mm)",
        main="1.7: Lawn roller data")
  gph
}

```

```

fig1.8 <- function(){
  pset <- simpleTheme(col.line="gray")
  gph <- xyplot(timef~time,
        data=nihills,
        aspect=1,
        type=c("p", "r"),

```

```

        par.settings=pset)
gph <- update(gph, xlab="Male record times",
             ylab="Female record times",
             main="1.8: f vs m times")
gph
}

```

```

fig1.9 <- function(obj=mftime.lm){
  gph <- plotSimScat(obj, layout=c(4,1), aspect=1)
  update(gph, xlab="Record times for males (h)",
         ylab="Record times for females (h)",
         main="1.9: f vs m times, simulation")
}

```

```

fig1.10 <- function(obj=mftime.lm){
  plot(obj, which=1, caption=NULL,
       sub.caption=NULL,
       main="1.10: Diagnostic plot 1")
}

```

```

fig1.11 <- function(obj=mftime.lm){
  gph <- plotSimScat(obj, show="residuals",
                   type=c("p","smooth"), layout=c(4,1))
  gph <- update(gph, xlab="Time (h) for males", ylab="Residuals",
               title="1.11: Diagnostic plot 1; 4 simulations",
               aspect=1)
  gph
}

```

```

fig1.12 <- function(obj=mftime.lm){
  plot(obj, which=2, caption=NULL,
       sub.caption=NULL,
       main="1.12: Diagnostic plot 2")
}

```

```

fig1.13 <- function(){
  gph <- plotSimDiags(obj=mftime.lm, which=2, layout=c(4,1),
                   aspect=1,
                   title="1.13: Diagnostic plot 2; 4 simulations")
  gph
}

```

```
fig1.14 <- function(obj=mftime.lm){
  plot(obj, which=3, caption=NULL,
        sub.caption=NULL,
        main="1.14: Diagnostic plot 3")
}
```

```
fig1.15 <- function(obj=mftime.lm){
  gph <- plotSimDiags(obj, which=3, layout=c(4,1),
                      aspect=1,
                      title="1.15: Diagnostic plot 3; 4 simulations")
  gph
}
```

```
fig1.16 <- function(){
  plot(mftime.lm, which=5, caption=NULL,
        sub.caption=NULL,
        main="1.16: Leverage plot")
}
```

```
fig1.17 <- function(){
  pset <- simpleTheme(lty=c(1,2))
  key <- list(text=c("Males", "Females"), columns=2)
  gph <- densityplot(~ time+timef, data=nihills, par.settings=pset,
                     ylab="Time (h)", auto.key=key,
                     main="1.17: Overlaid F and M densities")
  gph
}
```

```
fig1.18 <- function(){
  pset <- simpleTheme(col.line="gray")
  gph <- xyplot(timef ~ time,
                data=nihills,
                scales=list(log=10),
                aspect=1,
                type=c("p", "r"),
                par.settings=pset)
  gph <- update(gph, xlab="Male record times",
                ylab="Female record times",
                main="1.18: F vs M record times; log10 scales")
  gph
}
```

```
fig1.19 <- function(){
  obj <- lm(log(timef) ~ log(time), data=nihills)
  opar <- par(mfrow=c(1,4), mex=0.75, oma=c(0,0,2,0),
             mar=c(4.1,4.1,2.1,0.6), pty="s")
  plot(obj, cex.caption=0.75, cex.main=1.2,
       sub.caption="1.19: F vs M record times, diagnostic plots")
  par(opar)
}
```

```
fig1.20 <- function(){
  library(lattice)
  parset <- simpleTheme(cex=1.35, pch=16,
                      col=c("darkblue", "turquoise"))
  gabalong <- data.frame(values=unlist(gaba["30",])[-1],
                       sex=rep(c("male", "female", "all"), rep(2,3)),
                       trt=rep(c("Baclofen", "No baclofen"), 3))
  gph <- stripplot(sex~values, groups=trt, data=gabalong,
                 par.settings=parset,
                 xlab=list("Average reduction: 30 min vs 0 min",
                          cex=1.0),
                 scales=list(cex=1.0),
                 panel=function(x,y,...){
                   panel.stripplot(x,y,...)
                   ltext(x,y,paste(c(3,9,15,7,22,12)), pos=1,
                        cex=0.8)
                 }, auto.key=list(columns=2, points=TRUE, cex=1.0),
                 title="1.20: Pain reduction scores")

  gph
}
```

2 Code for the Figures

Unless `doFigs` is found in the workspace and is `FALSE`, then subject to checks that all necessary datasets and packages are available, the figures are now shown.

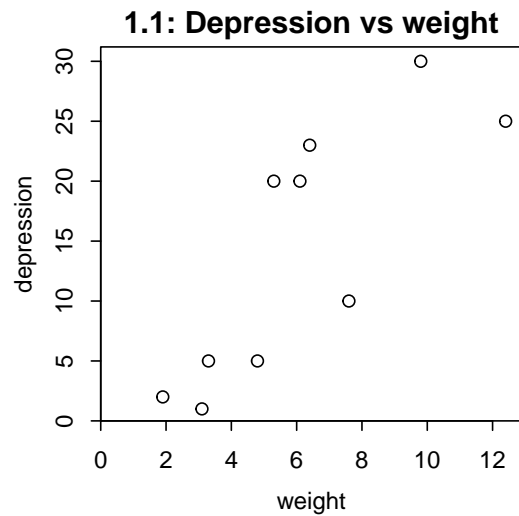
```
if(!exists("doFigs")) doFigs <- TRUE
```

```
library(DAAG)
```

```
Loading required package: lattice
```

```
mftime.lm <- lm(timef ~ time, data=nihills)
```

```
fig1.1()
```



```
fig1.2()
```

```
[1] "Run the separate functions fig1.2A() and fig1.2B()"
```

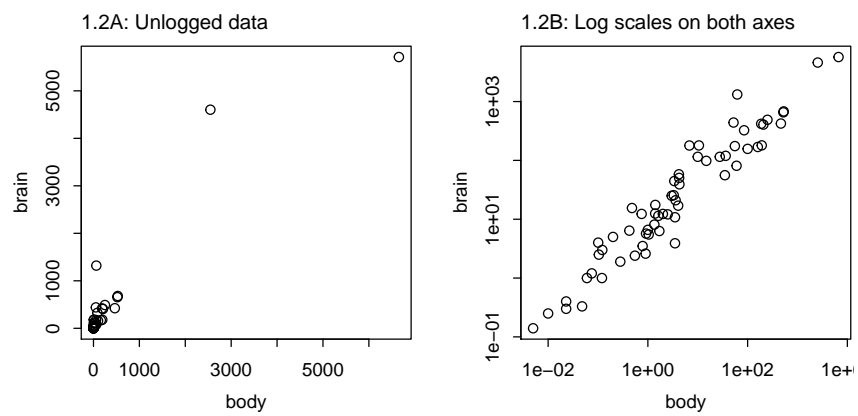
```
fig1.2A()
```

Loading required package: MASS

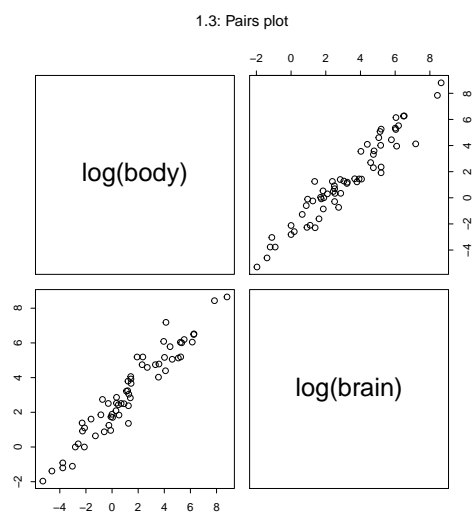
Attaching package: 'MASS'

The following object is masked from 'package:DAAG':
hills

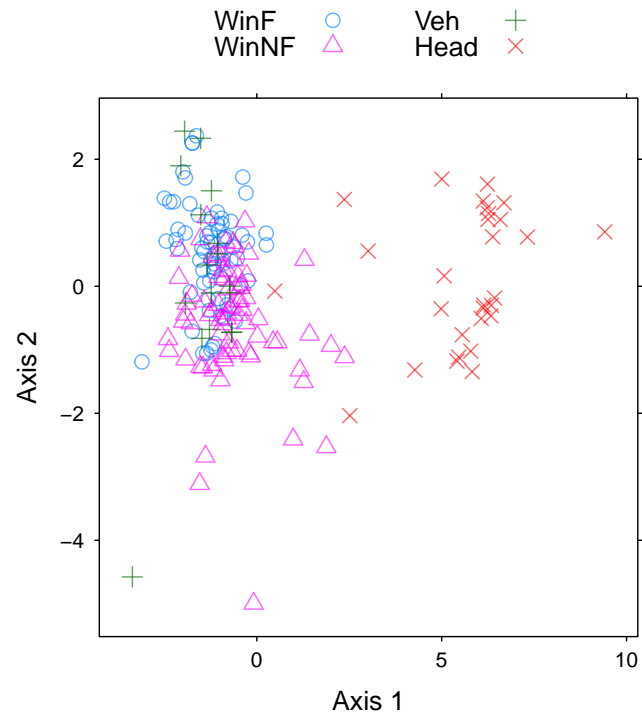
```
fig1.2B()
```



```
fig1.3()
```



```
fig1.4()
```



```
fig1.5()
```

Loading required package: *diagram*

Loading required package: *shape*

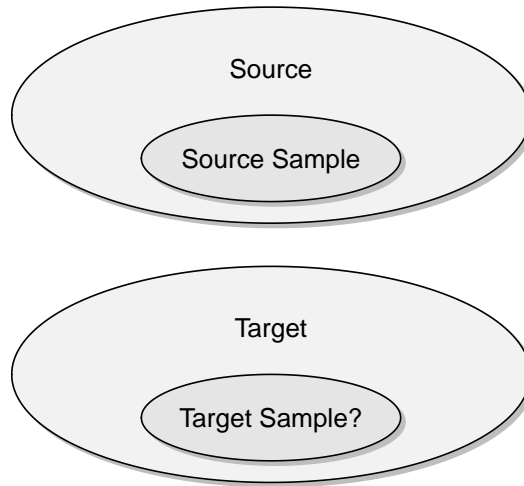
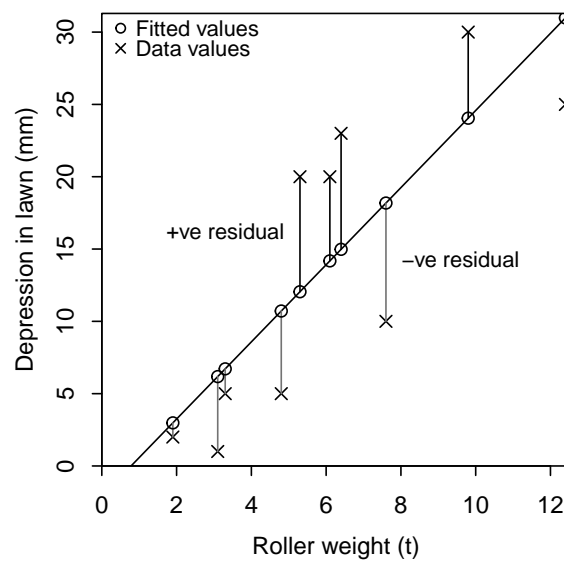
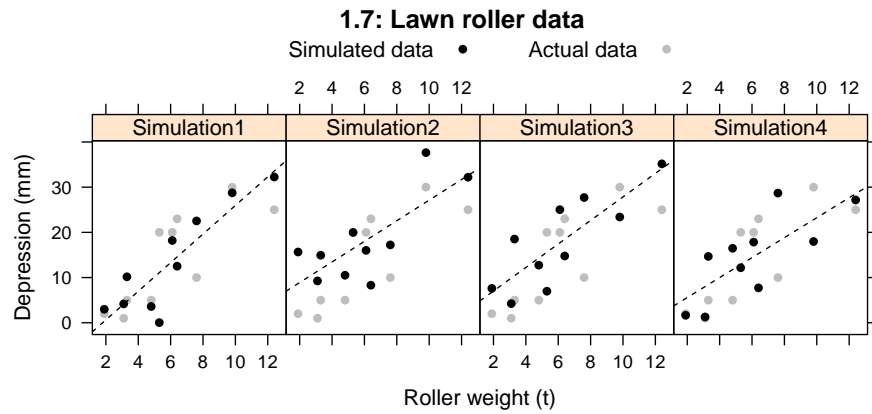


fig1.6()

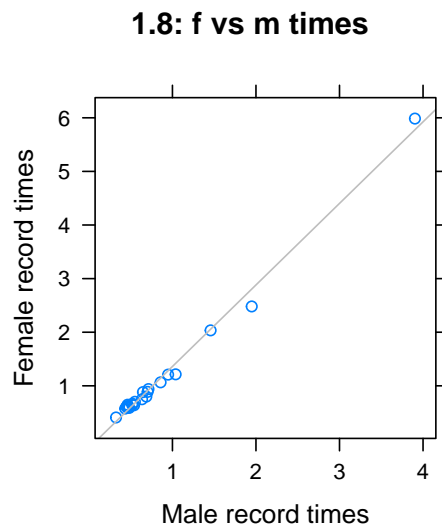
1.6: Lawn roller plot + line & annotation



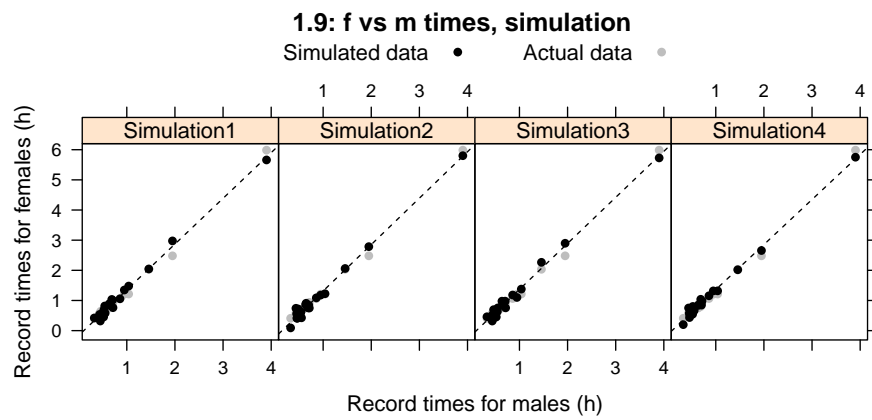
```
fig1.7()
```



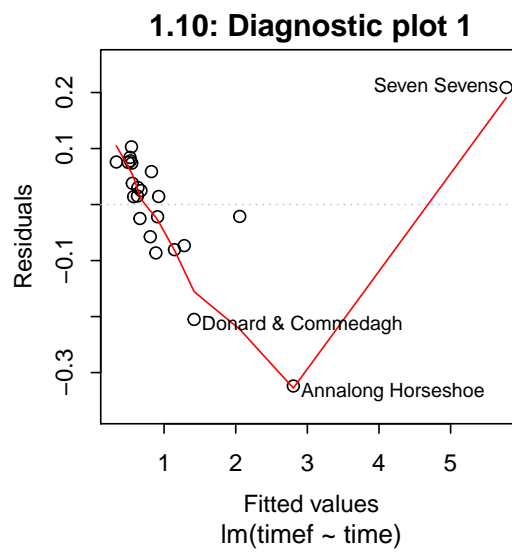
```
fig1.8()
```



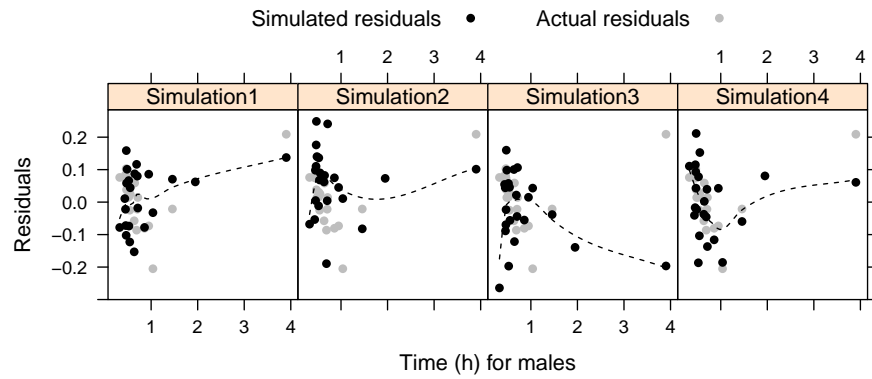
```
fig1.9()
```



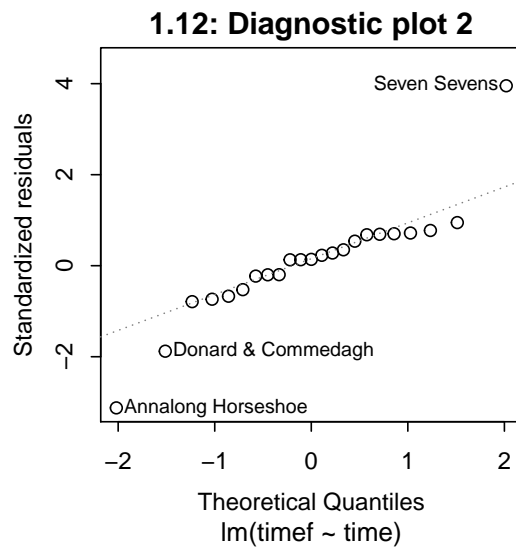
```
fig1.10()
```



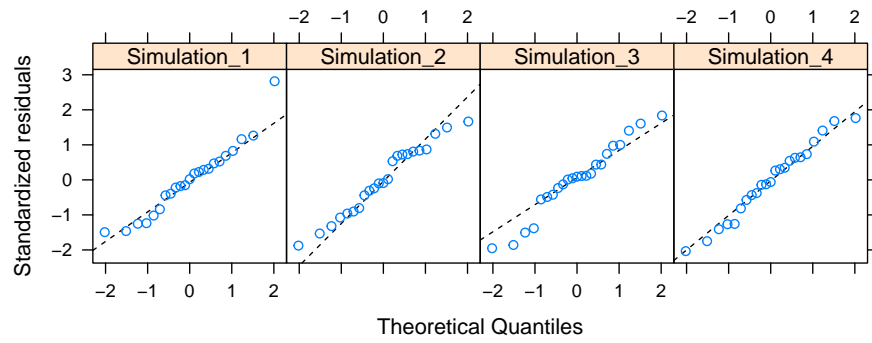
```
fig1.11()
```



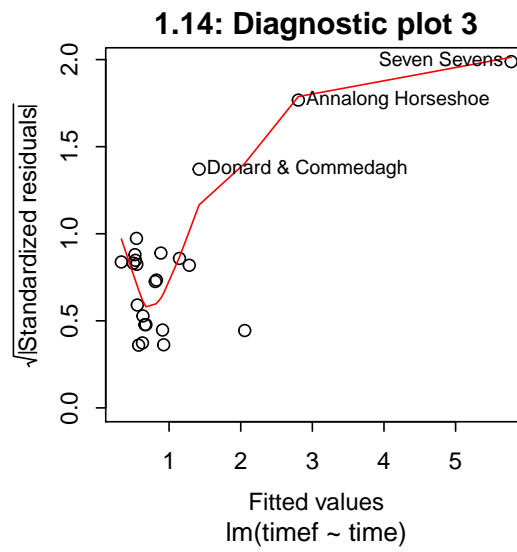
`fig1.12()`



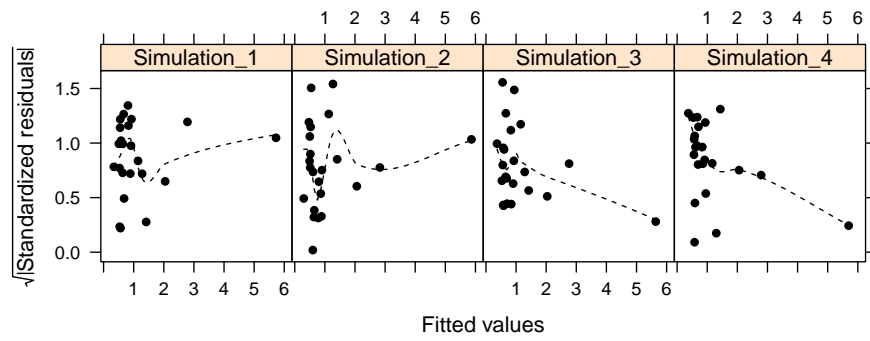
`fig1.13()`



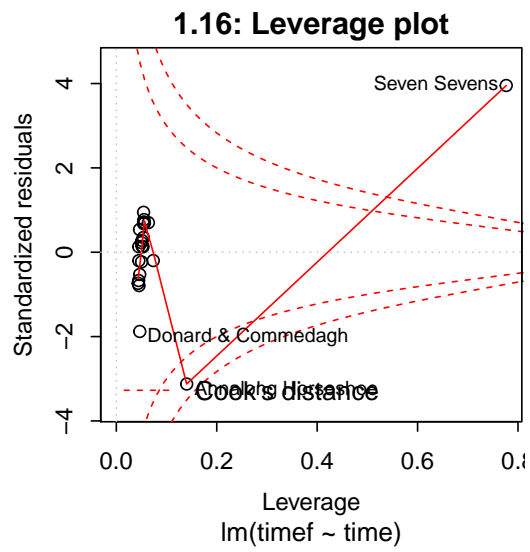
```
fig1.14()
```



```
fig1.15()
```

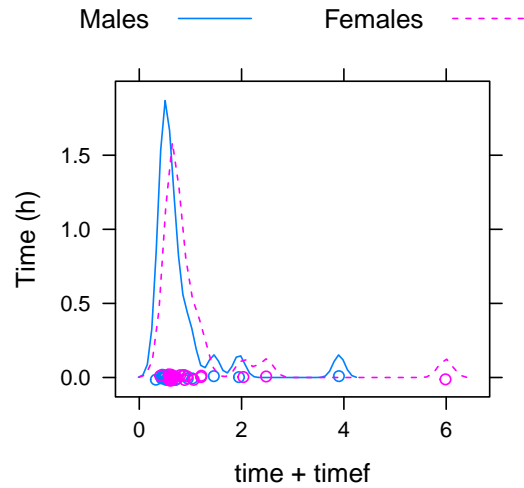


```
fig1.16()
```



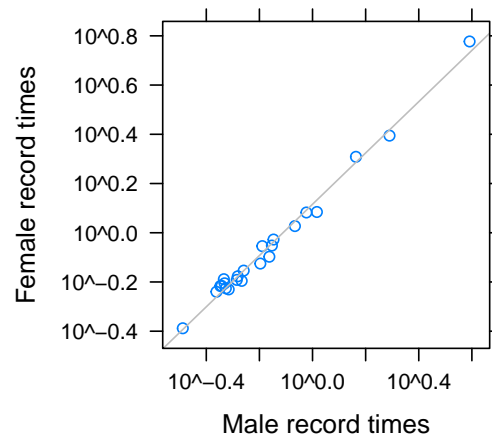
```
fig1.17()
```

1.17: Overlaid F and M densities



```
fig1.18()
```

1.18: F vs M record times; log10 scales



```
fig1.19()
```

1.19: F vs M record times, diagnostic plots

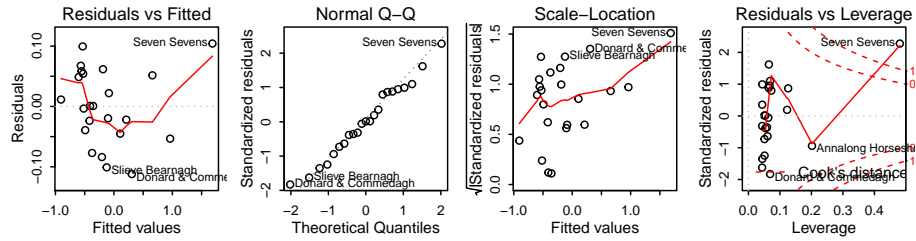


fig1.20()

