Figures for Chapter 1

John H Maindonald

October 30, 2012

```
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", ...)
}
```

```
fig1.2 <-
function ()
{
    library(MASS)
    par(fig = c(0, 0.5, 0, 1))
    plot(brain ~ body, data = mammals, pty = "s")
    mtext(side = 3, line = 0.5, adj = 0, "A: Unlogged data")
    par(fig = c(0.5, 1, 0, 1), new = TRUE)
    plot(brain ~ body, data = mammals, log = "xy", pty = "s")
    mtext(side = 3, line = 0.5, adj = 0, "B: Log scales on both axes")
    par(fig = c(0, 1, 0, 1))
}
```

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

```
fig1.2B <-
function ()
```

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

fig1.3 <-
function ()
{
    pairs(log(mammals), labels = c("log(body)", "log(brain)"))
}

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 = "Plot shows 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 Population", 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 Population", 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")
    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)
}


plotSimScat <-
    function(obj, sigma=NULL, layout=c(4,1), type=c("p","r"),
             show=c("points","residuals")){
        nsim <- prod(layout)
```

```
        if(is.null(sigma))sigma <- summary(obj)[["sigma"]]
        hat <- fitted(obj)
        xnam <- all.vars(formula(obj))[2]
        ynam <- all.vars(formula(obj))[1]
        df <- data.frame(sapply(1:nsim,
                                function(x)rnorm(length(hat), sd=sigma)))
        if(show[1]=="points")df <- df + hat
        simnam <- names(df) <- paste("Simulation", 1:nsim, sep="")
        df[, c(xnam, ynam)] <- model.frame(obj)[, c(xnam, ynam)]
        if(show[1]!="points"){df[, "Residuals"] <- df[, ynam] - hat
                              ynam <- "Residuals"
                              legadd <- "residuals"
                            } else legadd <- "data"
        leg <- list(text=paste(c("Simulated", "Actual"), legadd),
                    columns=2)
        formula <- formula(paste(paste(simnam, collapse="+"),
                                 "~", xnam))
        parset <- simpleTheme(pch=c(16,16), lty=2,
                              col=c("black","gray"))
        gph <- xyplot(formula, data=df, outer=TRUE,
                      par.settings=parset, auto.key=leg, lty=2,
                      layout=layout, type=type)
        formxy <- formula(paste(ynam, "~", xnam))
        addgph <- xyplot(formxy, data=df, pch=16, col="gray")
        gph+as.layer(addgph, under=TRUE)
    }

fig1.8 <- function(){
    obj <- lm(depression ~ weight, data=roller)
    gph <- plotSimScat(obj, sigma=6.4, layout=c(4,1))
    gph <- update(gph, xlab="Roller weight (t)", ylab="Depression (mm)")
    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")
    gph
}

fig1.9 <- 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")
}

fig1.10 <- function(){
     plot(mftime.lm, which=1, sub.caption="")
}

fig1.11 <- function(){
    obj <- lm(timef ~ time, data=nihills)
    gph <- plotSimScat(obj=mftime.lm, show="residuals",
                       type=c("p","smooth"), layout=c(4,1))
    gph <- update(gph, xlab="Time (h) for males", ylab="Residuals")
    gph
}

fig1.12 <- function(){
    plot(mftime.lm, which=2, sub.caption="")
}

fig1.13 <- function(){
    gph <- plotSimDiags(obj=mftime.lm, which=2, layout=c(4,1))
    gph
}

fig1.14 <- function(){
    plot(mftime.lm, which=3, sub.caption="")
}

fig1.15 <- function(){
    gph <- plotSimDiags(obj=mftime.lm, which=3, layout=c(4,1))
    gph
}

fig1.16 <- function(){
    plot(mftime.lm, which=5, sub.caption="")
}

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)
    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")
    gph
}

fig1.19 <- function(){
    obj <- lm(log(timef) ~ log(time), data=nihills)
    opar <- par(mfrow=c(1,4), mex=0.75, mar=c(4.1,4.1,2.1,0.6), pty="s")
    plot(obj, cex.caption=0.8)
    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))
    gph
}
```