Figures for Chapter 3

John H Maindonald

October 28, 2012

```
fig3.1 <-
function (plotit=TRUE){
library(MASS)             # MASS has the cats data set
wts <- with(cats, na.omit(Bwt[Sex=="F"]))
av <- mean(wts); sdev <- sd(wts); sampsize <- length(wts)
simmat <- cbind(wts, matrix(rnorm(sampsize*5, mean=av, sd=sdev),
                            ncol=5))
simdf <- as.data.frame(simmat)
names(simdf) <- c("Source", paste("normal", 1:5, sep=""))
simdf <- stack(simdf)
names(simdf) <- c("height", "Sample")
library(MASS)             # MASS has the cats data set
wts <- with(cats, na.omit(Bwt[Sex=="F"]))
av <- mean(wts); sdev <- sd(wts); sampsize <- length(wts)
simmat <- cbind(wts, matrix(rnorm(sampsize*5, mean=av, sd=sdev),
                            ncol=5))
simdf <- as.data.frame(simmat)
names(simdf) <- c("Source", paste("normal", 1:5, sep=""))
simdf <- stack(simdf)
names(simdf) <- c("height", "Sample")
denplotSimple <- densityplot(~height, groups=Sample, data=simdf,
                             xlab="Body weight (kg)")
denplotn <- update(denplotSimple, scales=list(tck=0.5),
                   main=list(expression(plain("A: Density plots")),
                             x=0.05, just="left"),
                   par.settings=simpleTheme(lty=1:6))
bwpltBasic <- bwplot(Sample ~ height, data=simdf,
                     xlab="Body weight (kg)",
                     auto.key=list(columns=3))
bwplotn <- update(bwpltBasic, scales=list(tck=0.5),
                  main=list(expression(plain("B: Boxplots")),
                            x=0.05, just="left"))
if(plotit){
    print(denplotn, position=c(0,0,0.5,1))
    print(bwplotn, position=c(0.5,0,1,1),newpage=FALSE)
```

```
}
invisible(list(denplotn, bwplotn))
}

fig3.2 <-
function (plotit=TRUE)
{
    wts <- with(cats, na.omit(Bwt[Sex=="F"]))
    sampsize <- length(wts)
    bootmat <- cbind(wts, matrix(0, ncol=5, nrow=sampsize))
    for(i in 2:6) bootmat[,i] <- sample(wts, replace=TRUE)
    colnames(bootmat) <- c("Source", paste("normal", 1:5, sep=""))
    bootdf <- stack(as.data.frame(bootmat))
    names(bootdf) <- c("height", "Sample")
    denplotSimple <- densityplot(~ height, groups=Sample, data=bootdf,
                                 xlab="Body weight (kg)")
    legendA <- expression(plain("A: Density plots (bootstrap samples)"))
    denplot <- update(denplotSimple, scales=list(tck=0.5),
                      main=list(legendA, x=0.05, just="left"),
                      par.settings=simpleTheme(lty=1:6))
    bwpltBasic <- bwplot(Sample ~ height, data=bootdf,
                         xlab="Body weight (kg)",
                         auto.key=list(columns=3))
    legendB <- expression(plain("B: Boxplots"))
    bwplot <- update(bwpltBasic, scales=list(tck=0.5),
                     main=list(legendB, x=0.05, just="left"))
    if(plotit){
        print(denplot, position=c(0,0,0.5,1))
        print(bwplot, position=c(0.5,0,1,1),newpage=FALSE)
    }
    invisible(list(denplot, bwplot))
}

fig3.3 <-
function ()
{
    opar <- par(mgp=c(2,.75,0), mfrow=c(1,2))
    curve(dnorm(x), from = -3, to = 3,
          ylab=expression("dnorm("*italic(x)*")"),
          xlab=expression("Normal deviate "*italic(x)))
    curve(pnorm(x), from = -3, to = 3,
          ylab=expression("pnorm("*italic(x)*")"),
          xlab=expression("Normal deviate "*italic(x)))
    par(opar)
}

fig3.4 <-
```

```
function (){
    library(MASS)
    heights <- na.omit(subset(survey, Sex=="Female")$Height)
    plot(density(heights), bty="l", main="",
         cex.axis=1.15, cex.lab=1.15)
    av <- mean(heights); sdev <- sd(heights)
    abline(v=c(av-sdev, av, av+sdev), col="gray", lty=c(2,1,2))
    ## Show fitted normal curve
    xval <- pretty(heights, n=40)
    normal_den <- dnorm(xval, mean=av, sd=sdev)
    lines(xval, normal_den, col="gray40", lty=2)
    ytop <- par()$usr[4]-0.25*par()$cxy[2]
    text(c(av-sdev, av+sdev), ytop,
         labels=c("mean-SD","mean+SD"), col="gray40", xpd=TRUE)
}

fig3.5 <-
function (){
    library(MASS)
    y <- with(cats, na.omit(Bwt[Sex=="F"]))
    opar <- par(pty="s")
    qqnorm(y)
    par(opar)
}

fig3.6 <-
function ()
{
    opar <- par(fig=c(0, 1, 0.465, 1))
    av <- numeric(1000)
    for (i in 1:1000)
        av[i] <- mean(rnorm(47, mean=2.36, sd=0.27))
    avdens <- density(av)
    xval <- pretty(c(2.36-3*0.27, 2.36+3*0.27), 50)
    den <- dnorm(xval, mean=2.36, sd=0.27)
    plot(xval, den, type="l", xlab="",  xlim=c(1.5, 3.75),
         ylab="Density", ylim=c(0,max(avdens$y)),
         col="gray", lwd=2, lty=2)
    lines(avdens)
    title(main="A: Simulation (from a normal distribution)", adj=0)
    legend("topleft", legend=c("Source", "Sampling\ndistribution\nof mean"),
           col=c("gray", "black"), lty=c(2,1), lwd=c(2,1), bty="n", cex=0.8)
    par(fig=c(0, 1, 0, 0.535), new=TRUE)
    y <- with(cats, na.omit(Bwt[Sex=="F"]))
    av <- numeric(1000)
    for (i in 1:1000)
```

```
        av[i] <- mean(sample(y, size=length(y), replace=TRUE))
    avdens <- density(av)
    plot(density(y), ylim=c(0, max(avdens$y)),
         xlab="", ylab="Density", xlim=c(1.5, 3.75),
         col="gray", lwd=2, lty=2, main="")
    lines(avdens)
    title(main="B: Bootstrap samples (from the sample data)", adj=0)
    legend("topleft", legend=c("Source",
                      "Sampling\ndistribution\nof mean"),
           col=c("gray", "black"), lty=c(2,1), lwd=c(2,1), bty="n",
           cex=0.8)
    par(fig=c(0,1,0,1))
}

fig3.7 <-
function ()
{
    xleft <- 0:3; xrt <- 1:4
    ybot <- rep(0,4); ytop <- rep(1,4) - 0.05
    opar <- par(mar=rep(0.1,4))
    plot(c(0,5), c(-1,4), xlab="", ylab="", axes=F, type="n")
    for(i in 0:3){
        i1 <- i+1
        rect(xleft, ybot+i, xrt, ytop+i)
        xli <- xleft[i+1]; xri <- xrt[i+1];
        yboti <- (ybot+i)[i+1]; ytopi <- (ytop+i)[i+1]
        rect(xli, yboti, xri, ytopi, col="gray80")
        text(0.5*(xli+xri), 0.5*(yboti+ytopi), "TEST")
        text(0.5*(xleft[-i1]+xrt[-i1]), 0.5*(ybot[-i1]+ytop[-i1])+i, "Training")
        text(4+strwidth("TE"), i+0.475, paste("Fold", i1), adj=0)
}
}

fig3.8 <-
function (plotit=TRUE)
{
    library(DAAG)
    library(grid)
    parset1 <- simpleTheme(pch=1:6, alpha=0.8)
    plt1 <- xyplot(length ~ breadth, groups=species, data=cuckoos,
                   par.settings=parset1, aspect=1,
                   scales=list(tck=0.5),
                   auto.key=list(columns=2, alpha=1),
                   main=textGrob("A:", x=unit(.025, "npc"),
                   y = unit(.25, "npc"), just="left",
                   gp=gpar(cex=1))
```

```
                )
    Species <- factor(c(rep("other", 5), "wren")[unclass(cuckoos$species)])
    parset2 <- simpleTheme(pch=c(0,6), alpha=0.8,
                           col=trellis.par.get()$superpose.symbol$col[c(7,6)])
    plt2 <- xyplot(length ~ breadth, groups=Species, data=cuckoos,
                   par.settings=parset2,
                   aspect=1, ylab="", scales=list(tck=0.25),
                   auto.key=list(columns=1, alpha=1),
                   main=textGrob("B:", x=unit(.05, "npc"),
                   y = unit(.25, "npc"), just="left",
                   gp=gpar(cex=1))
                   )
    plt2 <- update(plt2,
                   par.settings=list(layout.heights=list(key.top=1.5)))
    if(plotit){
        print(plt1, position=c(0,0,0.515,1))
        print(plt2, position=c(0.485,0,1,1), newpage=FALSE)
    }
    invisible(list(plt1, plt2))
}

fig3.9 <-
function ()
{
    parset <- list(dot.symbol=list(pch=1, alpha=0.6))
    dotwren <- dotplot(species %in% "wren" ~ length, data=cuckoos,
                       scales=list(y=list(labels=c("Other", "Wren"))),
                       par.settings=parset, xlab="Length (mm)")
    dotwren
}

fig3.10 <-
function ()
{
    avdiff <- numeric(100)
    for(i in 1:100){
        avs <- with(cuckoos, sapply(split(length, species %in% "wren"),
                                    function(x)mean(sample(x, replace=TRUE))))
        avdiff[i] <- avs[1] - avs[2]  # FALSE (non-wren) minus TRUE (wren)
    }
    xtxt <- paste("Means of bootstrap samples of length difference,",
                  "non-wren - wren (mm)")
    dotdiff <- dotplot(~ avdiff, xlab=xtxt,
                       par.settings=list(dot.symbol=list(pch=1, alpha=0.6)))
    dotdiff
}
```

```
fig3.11 <-
function ()
{
    mcats <- subset(cats, Sex=="M")
    xyplot(Hwt ~ Bwt, data=mcats,
           type=c("p","r"))
}

fig3.12 <-
function ()
{
    mcats <- subset(cats, Sex=="M")
    mcats.lm <- lm(Hwt ~ Bwt, data=mcats)
    res <- resid(mcats.lm)
    plot(density(res), main="")
    rug(res, col="gray")
}

fig3.13 <-
function ()
{
    if(!require(car))stop("Package 'car' must be installed")
    mcats <- subset(cats, Sex=="M")
    bootmat <- bootreg(formula = Hwt ~ Bwt,
                       data = mcats,
                       n = 1000)
    bootdf <- as.data.frame(bootmat)
    names(bootdf) <- c("Intercept","Slope")
    colr <- adjustcolor(rep("black",3),
                        alpha.f=0.25)
    scatterplot(Slope ~ Intercept, col=colr,
                data=bootdf, boxplots="xy",
                reg.line=NA, smooth=FALSE)
}

fig3.14 <-
function (plotit=TRUE)
{
    mcats <- subset(cats, Sex=="M")
    bootmat <- bootreg(formula = Hwt ~ Bwt,
                       data = mcats[-97, ],
                       n = 1000)
    bootdf0 <- as.data.frame(bootmat)
    names(bootdf0) <- c("Intercept","Slope")
    gphA <- xyplot(Slope ~ Intercept, data=bootdf0, alpha=0.25)
        simmat <- simreg(formula = Hwt ~ Bwt,
                         data=mcats[-97, ], n=1000)
```

```
    simdf <- as.data.frame(simmat)
    names(simdf) <- c("Intercept","Slope")
    gphB <- xyplot(Slope ~ Intercept, data=simdf, alpha=0.25)
    if(plotit){
        print(gphA, position=c(0,0,0.515,1))
        print(gphB, position=c(0.485,0,1,1), newpage=FALSE)
    }
    invisible(list(gphA, gphB))
}
```