Data Analysis & Graphics Using R, 3^{rd} edn – Solutions to Exercises (April 29, 2010)

Preliminaries

```
> library(DAAG)
```

Exercise 1
Use the lattice function `bwplot()` to display, for each combination of `site` and `sex` in the data frame `possum` (*DAAG* package), the distribution of ages. Show the different sites on the same panel, with different panels for different sexes.

```
> library(lattice)
> bwplot(age ~ site | sex, data=possum)
```

Exercise 3
Plot a histogram of the `earconch` measurements for the `possum` data. The distribution should appear *bimodal* (two peaks). This is a simple indication of clustering, possibly due to sex differences. Obtain side-by-side boxplots of the male and female `earconch` measurements. How do these measurement distributions differ? Can you predict what the corresponding histograms would look like? Plot them to check your answer.

```
> par(mfrow=c(1,2), mar=c(3.6,3.6,1.6,0.6))
> hist(possum$earconch, main="")
> boxplot(earconch ~ sex, data=possum, boxwex=0.3, horizontal=TRUE)
> par(mfrow=c(1,1))
```

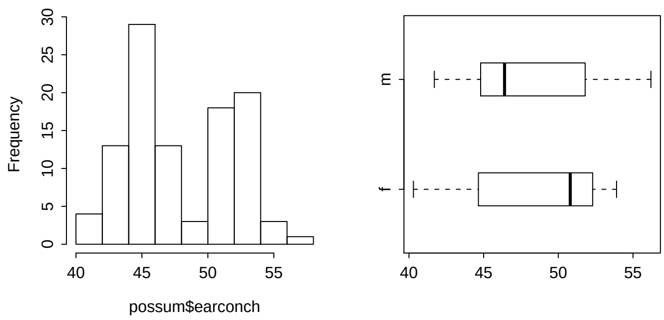

Figure 1: The left panel shows a histogram of possum ear conch measurements. The right panel shows side by side boxplots, one for each sex. A horizontal layout is often advantageous.

Note the alternative to `boxplot()` that uses the *lattice* function `bwplot()`. Placing `sex` on the left of the graphics formula leads to horizontal boxplots.

```
bwplot(sex ~ earconch, data=possum)
```

The following gives side by side histograms:

```
> par(mfrow=c(1,2))
> hist(possum$earconch[possum$sex == "f"], border="red", main="")
> hist(possum$earconch[possum$sex == "m"], border="blue", main="")
> par(mfrow=c(1,1))
```

The histograms make it clear that sex differences are not the whole of the explanation for the bimodality.

Alternatively, use the *lattice* function `histogram()`

```
> library(lattice)
> histogram(~ earconch | sex, data=possum)
```

Note: We note various possible alternative plots.

Density plots, in addition to their other advantages, are easy to overlay. Alternatives 1 & 2 obtain overlaid density plots:

```
> "Alternative 1: Overlaid density plots"
> fden <- density(possum$earconch[possum$sex == "f"])
> mden <- density(possum$earconch[possum$sex == "m"])
> xlim <- range(c(fden$x, mden$x))
> ylim <- range(c(fden$y, mden$y))
> plot(fden, col="red", xlim=xlim, ylim=ylim, main="")
> lines(mden, col="blue", lty=2)

> library(lattice)
> "Alternative 2: Overlaid density plots, using the lattice package"
> print(densityplot(~earconch, data=possum, groups=sex), main="")
```

Alternatives 3 and 4 give alternative forms of histogram plot.

```
> "Alternative 3: Overlaid histograms, using regular graphics"
> fhist <- hist(possum$earconch[possum$sex=="f"], plot=F,
+               breaks=seq(from=40,to=58,by=2))
> mhist <- hist(possum$earconch[possum$sex=="m"], plot=F,
+               breaks=seq(from=40,to=58,by=2))
> ylim <- range(fhist$density, mhist$density)
> plot(fhist, freq=F, xlim=c(40,58), ylim=ylim, border="red", main="")
> lines(mhist, freq=F, border="blue", lty=2)
```

Note the use of `border="red"` to get the histogram for females outlined in red. The parameter setting `col="red"` gives a histogram with the rectangles filled in red.

Unfortunately, `histogram()` in the lattice package ignores the parameter `groups`. With `histogram()`, we are limited to side by side histograms:

```
> "Alternative 4: Side by side histograms, using the lattice package"
> print(histogram(~earconch | sex, data=possum), main="")
```

Both for density plots and for histograms, do we really want the separate total areas to be scaled to 1, as happens with the setting `freq=FALSE`, rather than to the total frequencies in the respective populations? This will depend on the specific application.

Exercise 4
For the data frame `ais` (*DAAG* package), draw graphs that show how the values of the hematological measures (red cell count, hemoglobin concentration, hematocrit, white cell count and plasma ferritin concentration) vary with the sport and sex of the athlete.

Use for example

```
> bwplot(sport ~ rcc | sex, data=ais)
```

Exercise 5
Using the data frame `cuckoohosts`, plot `clength` against `cbreadth`, and `hlength` against `hbreadth`, all on the same graph and using a different color to distinguish the first set of points (for the cuckoo eggs) from the second set (for the host eggs). Join the two points that relate to the same host species with a line. What does a line that is long, relative to other lines, imply? Here is code that you may wish to use:

```
attach(cuckoohosts)
plot(c(clength, hlength), c(cbreadth, hbreadth),
     col=rep(1:2,c(12,12)))
for(i in 1:12)lines(c(clength[i], hlength[i]),
                    c(cbreadth[i], hbreadth[i]))
text(hlength, hbreadth, abbreviate(rownames(cuckoohosts),8))
detach(cuckoohosts)
```

A line that is long relative to other lines, as for the wren, is indicative of an unusually large difference in egg dimensions.

Exercise 7
Install and attach the package *Devore5*, available from the CRAN sites. Then gain access to data on tomato yields by typing

```
library(Devore5)
tomatoes <- ex10.22
```

This data frame gives tomato yields at four levels of salinity, as measured by electrical conductivity (`EC`, in nmhos/cm).

(a) Obtain a scatterplot of `yield` against `EC`.

(b) Obtain side-by-side boxplots of `yield` for each level of `EC`.

(c) The third column of the data frame is a factor representing the four different levels of `EC`. Comment upon whether the yield data are more effectively analyzed using `EC` as a quantitative or qualitative factor.

```
> library(Devore6)
> tomatoes <- ex10.22
> plot(yield ~ EC, data=tomatoes)
> boxplot(split(tomatoes$yield, tomatoes$EC))
```

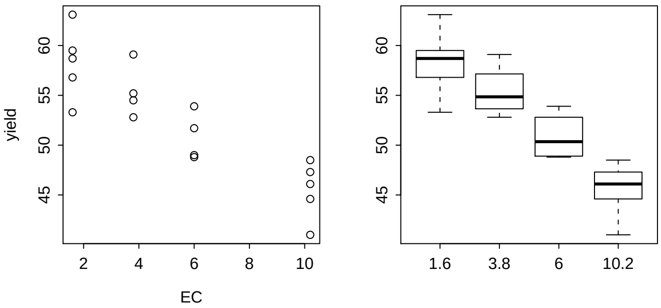

Figure 2: The left panel plots `yield` against `EC`. The right panel shows boxplots of `yield` for each distinct value of `EC`.

The data are more effectively analyzed using `EC` as a quantitative factor. Treating `EC` as a factor would ignore the linear or near linear dependence of `yield` on `EC`.

Exercise 8
Examine the help for the function `mean()`, and use it to learn about the trimmed mean. For the total lengths of female possums, calculate the mean, the median, and the 10% trimmed mean. How does the 10% trimmed mean differ from the mean for these data? Under what circumstances will the trimmed mean differ substantially from the mean?

```
> fossum <- possum[possum$sex=="f", ]
> mean(fossum$totlngth)

[1] 87.90698

> c(median=median(fossum$totlngth),
+   "trim-mean-0.1"= mean(fossum$totlngth, trim=0.1))

     median trim-mean-0.1
   88.50000      88.04286
```

The following gives an indication of the shape of the distribution:

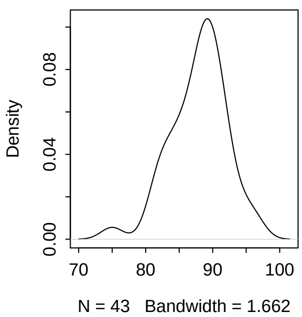

Figure 3: Density plot of female possum lengths.

```
> totlngth <- fossum[, "totlngth"]
> plot(density(totlngth), main="")
```

The distribution is negatively skewed, i.e., it has a tail to the left. As a result, the mean is substantially less than the mean. Removal of the smallest and largest 10% of values leads to a distribution that is more nearly symmetric. The mean is then similar to the median. (Note that trimming the same amount off both tails of the distribution does not affect the median.)

The trimmed mean will differ substantially from the mean when the distribution is positively or negatively skewed.

Exercise 9
Assuming that the variability in egg length for the cuckoo eggs data is the same for all host birds, obtain an estimate of the pooled standard deviation as a way of summarizing this variability. [Hint: Remember to divide the appropriate sums of squares by the number of degrees of freedom remaining after estimating the six different means.]

```
> sapply(cuckoos, is.factor)   # Check which columns are factors

 length breadth species      id
  FALSE   FALSE    TRUE   FALSE

> specnam <- levels(cuckoos$species)
> ss <- 0
> ndf <- 0
> for(nam in specnam){
+   lgth <- cuckoos$length[cuckoos$species==nam]
+   ss <- ss + sum((lgth - mean(lgth))^2)
+   ndf <- ndf + length(lgth) - 1
+ }
> sqrt(ss/ndf)

[1] 0.9051987
```

A more cryptic solution is:

```
> diffs <- unlist(sapply(split(cuckoos$length, cuckoos$species),
+                 function(x)x-mean(x)))
> df <- unlist(sapply(split(cuckoos$length, cuckoos$species),
+              function(x)length(x) - 1))
> sqrt(sum(diffs^2)/sum(df))
```