Exercises – R-based Data Analysis and Statistical Learning

John Maindonald

August 1, 2010

Contents

Part VI
Data Summary – Traps for the Unwary

Package: `DAAGxtras`

1 Multi-way Tables

	Small (<2cm)			Large (>=2cm)			Total	
	open	ultrasound		open	ultrasound		open	ultrasound
yes	81	234	yes	192	55	yes	273	289
no	6	36	no	71	25	no	77	61
Success rate	93%	87%		73%	69%		78%	83%

Table 1: Outcomes for two different types of surgery for kidney stones. The overall success rates (78% for open surgery as opposed to 83% for ultrasound) favor ultrasound. Comparison of the success rates for each size of stone separately favors, in each case, open surgery.

Exercise 1
Table 1 illustrates the potential hazards of adding a multiway table over one of its margins. Data are from a study[a] that compared outcomes for two different types of surgery for kidney stones; A: `open`, which used open surgery, and B: `ultrasound`, which used a small incision, with the stone destroyed by ultrasound. The data can be entered into R, thus:

```
> stones <- array(c(81, 6, 234, 36, 192, 71, 55, 25), dim = c(2,
+     2, 2), dimnames = list(Sucess = c("yes", "no"), Method = c("open",
+     "ultrasound"), Size = c("<2cm", ">=2cm")))
```

(a) Determine the success rate that is obtained from combining the data for the two different sizes of stone. Also determine the success rates for the two different stone sizes separately.

(b) Use the following code to give a visual representation of the information in the three-way table:

```
mosaicplot(stones, sort=3:1)
   # Re-ordering the margins gives a more interpretable plot.
```

Annotate the graph to show the success rates?

(c) Observe that the overall rate is, for open surgery, biased toward the open surgery outcome for large stones, while for ultrasound it is biased toward the outcome for small stones. What are the implications for the interpretation of these data?

[Without additional information, the results are impossible to interpret. Different surgeons will have preferred different surgery types, and the prior condition of patients will have affected the choice of surgery type. The consequences of unsuccessful surgery may have been less serious than for ultrasound than for open surgery.]

[a] Charig, C. R., 1986. Comparison of treatment of renal calculi by operative surgery, percutaneous nephrolithotomy, and extracorporeal shock wave lithotripsy. *British Medical Journal*, 292:879–882

The relative success rates for the two different types of surgery, for the two stone sizes separately, can be calculated thus:

```
> stones[1, , ]/(stones[1, , ] + stones[2, , ])
```

To perform the same calculation after adding over the two stone sizes (the third dimension of the table), do

```
> stones2 <- stones[, , 1] + stones[, , 2]
> stones2[1, ]/(stones2[1, ] + stones2[2, ])
```

1.1 Which multi-way table? It can be important!

Each year the National Highway Traffic Safety Administration (NHTSA) in the USA collects, using a random sampling method, data from all police-reported crashes in which there is a harmful event (people or property), and from which at least one vehicle is towed. The data frame `nassCDS` (*DAAGxtras*) is derived from NHTSA data.[1]

The data are a sample. The use of a complex sampling scheme has the consequence that the sampling fraction differs between observations. Each point has to be multiplied by the relevant sampling fraction, in order to get a proper estimate of its contribution to the total number of accidents. The column `weight` (`national` = *national inflation factor* in the SAS dataset) gives the relevant multiplier.

Other variables than those included in `nassCDS` might be investigated – those extracted into `nassCDS` are enough for present purposes.

The following uses `xtabs()` to estimate numbers of front seat passengers alive and dead, classified by airbag use:

```
library(DAAGxtras)
> abtab <- xtabs(weight ~ dead + airbag, data=nassCDS)
> abtab
       airbag
dead          none     airbag
  alive 5445245.90 6622690.98
  dead    39676.02   25919.11
```

The function `prop.table()` can then be used to obtain the proportions in margin 1, i.e., the proportions dead, according to airbag use:

```
> round(prop.table(abtab, margin=2)["dead", ], 4)
  none airbag
0.0072 0.0039
## Alternatively, the following gives proportions alive & dead
## round(prop.table(abtab, margin=2), 4)
```

The above might suggest that the deployment of an airbag substantially reduces the risk of mortality. Consider however:

```
> abSBtab <- xtabs(weight ~ dead + seatbelt + airbag, data=nassCDS)
> ## Take proportions, retain margins 2 & 3, i.e. airbag & seatbelt
> round(prop.table(abSBtab, margin=2:3)["dead", , ], 4)
        seatbelt
airbag     none belted
  none   0.0176 0.0038
  airbag 0.0155 0.0021
```

The results are now much less favorable to airbags. The clue comes from examination of:

[1] They hold a subset of the columns from a corrected version of the data analyzed in the Meyer (2005) paper that is referenced on the help page for `nassCDS`. More complete data are available from one of the web pages
`http://www.stat.uga.edu/~mmeyer/airbags.htm` (SAS transport file)
or `http://www.maths.anu.edu.au/~johnm/datasets/airbags/` (R image file).

```
> margin.table(AStab, margin=2:3)  # Add over margin 1
        airbag
seatbelt      none    airbag
  none   1366088.6  885635.3
  belted 4118833.4 5762974.8
```

In the overall table, the results without airbags are mildly skewed (4.12:1.37) to the results for `belted`, while with airbags they are highly skewed (57.6:8.86) to the results for `belted`.

Exercise 2
Do an analysis that accounts, additionally, for estimated force of impact (`dvcat`):

```
ASdvtab <- xtabs(weight ~ dead + seatbelt + airbag + dvcat,
                      data=nassCDS)
round(prop.table(ASdvtab, margin=2:4)["dead", , , ], 6)
## Alternative: compact, flattened version of the table
round(ftable(prop.table(ASdvtab, margin=2:4)["dead", , , ]), 6)
```

It will be apparent that differences between `none` and `airbag` are now below any reasonable threshold of statistical detectability.

Exercise 3
The package *DAAGxtras* includes the function `excessRisk()`. Run it with the default arguments, i.e. type

```
> excessRisk()
```

Compare the output with that obtained in Exercise 2 when the classification was a/c seatbelt (and airbag), and check that the output agrees.
Now do the following calculations, in turn:

(a) Classify according to `dvcat` as well as `seatbelt`. All you need do is add `dvcat` to the first argument to `excessRisk()`. What is now the total number of excess deaths?
[The categories are 0-9 kph, 10-24 kph, 25-39 kph, 40-54 kph, and 55+ kph]

(b) Classify according to `dvcat`, `seatbelt` and `frontal`, and repeat the calculations. What is now the total number of excess deaths?

Explain the dependence of the estimates of numbers of excess deaths on the choice of factors for the classification.

Note: ? argues that these data, tabulated as above, have too many uncertainties and potential sources of bias to give reliable results. He presents a different analysis, based on the use of front seat passenger mortality as a standard against which to compare driver mortality, and limited to cars without passenger airbags. In the absence of any effect from airbags, the ratio of driver mortality to passenger mortality should be the same, irrespective of whether or not there was a driver airbag. In fact the ratio of driver fatalities to passenger fatalities was 11% lower in the cars with driver airbags.

2 Weighting Effects – Example with a Continous Outcome

Exercise 4
Table 2, shows data from the data frame `gaba` (*DAAGxtras*). For background, see the Gordon (1995) paper that is referenced on the help page for `gaba`. [Image files that hold the functions `plotGaba()` and `compareGaba()` are in the subdirectory `http://www.maths.anu.edu.au/~johnm/r/functions/`]

	min	mbac	mpl	fbac	fpl
2	10	1.76	1.76	2.18	2.55
3	30	1.31	1.65	3.48	4.15
4	50	0.05	0.67	3.13	3.66
5	70	−0.57	−0.25	3.03	2.05
6	90	−1.26	−0.50	2.08	0.61
7	110	−2.15	−2.22	1.60	0.34
8	130	−1.65	−2.18	1.38	0.67
9	150	−1.68	−2.86	1.76	0.76
10	170	−1.68	−3.23	1.06	0.39

Table 2: Data (average VAS pain scores) are from a trial that investigated the effect of pentazocine on post-operative pain, with (mbac and fbac) and without (mpl and fpl) preoperatively administered baclofen. Data are in the data frame `gaba` (*DAAGxtras* package). Numbers of males and females on the two treatments were:

	baclofen	placebo
females	15	7
males	3	16

Exercise 4, continued

(a) What do you notice about the relative numbers on the two treatments?

(b) For each treatment, obtain overall weighted averages at each time point, using the numbers in Table 2 as weights. (These are the numbers you would get if you divided the total over all patients on that treatment by the total number of patients.) This will give columns `avbac` and `avplac` that can be added the data frame.

(c) Plot `avbac` and `avplac` against time, on the same graph. On separate graphs, repeat the comparisons (a) for females alone and (b) for males alone. Which of these graphs make a correct and relevant comparison between baclofen and placebo (albeit both in the presence of pentazocine)?

3 Extraction of nassCDS

Here are details of the code used to extract these data.

```
nassCDS <- nass9702cor[,c("dvcat", "national", "dead", "airbag", "seatbelt",
                          "frontal", "male", "age.of.o", "yearacc")]
nassCDS$dead <- 2-nass_cds$dead  # Ensures 0 = survive; 1 = dead
## Now make dead a factor
nassCDS$dead <- factor(c("alive", "dead")[nassCDS$dead+1])
names(nassCDS)[8] <- "ageOFocc"
names(nassCDS)[2] <- "weight"
table(nassCDS$seatbelt)  # Check the values of seatbelt, & their order
## Now make seatbelt a factor.  The first value, here 0, becomes "none"
## The second value, here 1, becomes "belted"
nassCDS$seatbelt <- factor(nassCDS$seatbelt, labels=c("none","belted"))
  # NB labels (unlike levels) matches only the order, not the values
nassCDS$airbag <- factor(nassCDS$airbag, labels=c("none","airbag"))
```

Part VII
Populations & Samples – Theoretical & Empirical Distributions

R functions that will be used in this laboratory include:

(a) `dnorm()`: Obtain the density values for the theoretical normal distribution;

(b) `pnorm()`: Given a normal deviate or deviates, obtain the cumulative probability;

(c) `qnorm()`: Given the cumulative probabilty. calculate the normal deviate;

(d) `sample()`: take a sample from a vector of values. Values may be taken without replacement (once taken from the vector, the value is not available for subsequent draws), or with replacement (values may be repeated);

(e) `density()`: fit an empirical density curve to a set of values;

(f) `rnorm()`: Take a random sample from a theoretical normal distribution;

(g) `runif()`: similar to `rnorm()`, but sampling is from a uniform distribution;

(h) `rt()`: similar to `rnorm()`, but sampling is from a t-distribution (the degrees of freedom must be given as the second parameter);

(i) `rexp()`: similar to `rnorm()`, but sampling is from an exponential distribution;

(j) `qqnorm()`: Compare the empirical distribution of a set of values with the empirical normal distribution.

1 Populations and Theoretical Distributions

Exercise 1

(a) Plot the density and the cumulative probability curve for a normal distribution with a mean of 2.5 and SD = 1.5.
Code that will plot the curve is

```
> curve(dnorm(x, mean = 2.5, sd = 1.5), from = 2.5 - 3 * 1.5, to = 2.5 +
+     3 * 1.5)
> curve(pnorm(x, mean = 2.5, sd = 1.5), from = 2.5 - 3 * 1.5, to = 2.5 +
+     3 * 1.5)
```

(b) From the cumulative probability curve in (a), read off the area under the density curve between `x=0.5` and `x=4`. Check your answer by doing the calculation

```
> pnorm(4, mean = 2.5, sd = 1.5) - pnorm(0.5, mean = 2.5, sd = 1.5)

[1] 0.7501335
```

Exercise 1, continued

(a) The density for the distribution in items (i) and (ii), given by `dnorm(x, 2.5, 1.5)`, gives the relative number of observations per unit interval that can be expected at the value `x`. For example `dnorm(x=2, 2.5, 1.5)` $\simeq 0.2516$. Hence

(i) In a sample of 100 the expected number of observations per unit interval, in the immediate vicinity of `x = 2`, is 25.16

(ii) In a sample of 1000 the expected number of observations per unit interval, in the immediate vicinity of `x = 2`, is 251.6

(iii) The expected number of values from a sample of 100, between 1.9 amd 2.1, is approximately $0.2 \times 251.6 = 50.32$

[The number can be calculated more exactly as
`(pnorm(2.1, 2.5, 1.5) - pnorm(1.9, 2.5, 1.5)) * 1000`]

Repeat the calculation to get approximate and more exact values for the expected number

(i) between 0.9 and 1.1

(ii) between 2.9 and 3.1

(iii) between 3.9 and 4.1

By way of example, here is the code for (a):

```
> curve(dnorm(x, mean = 2.5, sd = 1.5), from = 2.5 - 3 * 1.5, to = 2.5 +
+     3 * 1.5)
> curve(pnorm(x, mean = 2.5, sd = 1.5), from = 2.5 - 3 * 1.5, to = 2.5 +
+     3 * 1.5)
```

Exercise 2

(a) Plot the density and the cumulative probability curve for a t-distribution with 3 degrees of freedom. Overlay, in each case, a normal distribution with a mean of 0 and SD=1.
[Replace `dnorm` by `dt`, and specify `df=10`]

(b) Plot the density and the cumulative probability curve for an exponential distribution with a rate parameter equal to 1 (the default). Repeat, with a rate parameter equal to 2. (When used as a failure time distribution; the rate parameter is the expected number of failures per unit time.)

2 Samples and Estimated Density Curves

Exercise 3
Use the function `rnorm()` to draw a random sample of 25 values from a normal distribution with a mean of 0 and a standard deviation equal to 1.0. Use a histogram, with `probability=TRUE` to display the values. Overlay the histogram with: (a) an estimated density curve; (b) the theoretical density curve for a normal distribution with mean 0 and standard deviation equal to 1.0. Repeat with samples of 100 and 500 values, showing the different displays in different panels on the same graphics page.

```
> par(mfrow = c(1, 3), pty = "s")
> x <- rnorm(50)
> hist(x, probability = TRUE)
```

```
> lines(density(x))
> xval <- pretty(c(-3, 3), 50)
> lines(xval, dnorm(xval), col = "red")
```

Exercise 4
Data whose distribution is close to lognormal are common. Size measurements of biological organisms often have this character. As an example, consider the measurements of body weight (`body`), in the data frame `Animals` (*MASS*). Begin by drawing a histogram of the untransformed values, and overlaying a density curve. Then

(a) Draw an estimated density curve for the logarithms of the values. Code is given immediately below.

(b) Determine the mean and standard deviation of `log(Animals$body)`. Overlay the estimated density with the theoretical density for a normal distribution with the mean and standard deviation just obtained.

Does the distribution seem normal, after transformation to a logarithmic scale?

```
> library(MASS)
> plot(density(Animals$body))
> logbody <- log(Animals$body)
> plot(density(logbody))
> av <- mean(logbody)
> sdev <- sd(logbody)
> xval <- pretty(c(av - 3 * sdev, av + 3 * sdev), 50)
> lines(xval, dnorm(xval, mean = av, sd = sdev))
```

Exercise 5
The following plots an estimated density curve for a random sample of 50 values from a normal distribution:

```
> plot(density(rnorm(50)), type = "l")
```

(a) Plot estimated density curves, for random samples of 50 values, from (a) the normal distribution; (b) the uniform distribution (`runif(50)`); (c) the t-distribution with 3 degrees of freedom. Overlay the three plots (use `lines()` in place of `plot()` for densities after the first).

(b) Repeat the previous exercise, but taking random samples of 500 values.

Exercise 6
There are two ways to make an estimated density smoother:

(a) One is to increase the number of samples, For example:

```
> plot(density(rnorm(500)), type = "l")
```

Exercise 6, continued

(b) The other is to increase the bandwidth. For example

```
> plot(density(rnorm(50), bw = 0.2), type = "l")
> plot(density(rnorm(50), bw = 0.6), type = "l")
```

Repeat each of these with bandwidths (`bw`) of 0.15, with the default choice of bandwidth, and with the bandwidth set to 0.75.

Exercise 7

Here we experiment with the use of `sample()` to take a sample from an empirical distribution, i.e., from a vector of values that is given as argument. Here, the sample size will be the number of values in the argument. Any size of sample is however permissible.

```
> sample(1:5, replace = TRUE)
> for (i in 1:10) print(sample(1:5, replace = TRUE))
> plot(density(log10(Animals$body)))
> lines(density(sample(log10(Animals$body), replace = TRUE)), col = "red")
```

Repeat the final density plot several times, perhaps using different colours for the curve on each occasion. This gives an indication of the stability of the estimated density curve with respect to sample variation.

3 *Normal Probability Plots

Exercise 8

Partly because of the issues with bandwidth and choice of kernel, and partly because it is hard to density estimates are not a very effective means for judging normality. A much better tool is the normal probability plot, which works with cumulative probability distributions. Try

```
> qqnorm(rnorm(10))
> qqnorm(rnorm(50))
> qqnorm(rnorm(200))
```

For samples of modest to large sizes, the points lie close to a line.
The function `qreference()` (*DAAG*) takes one sample as a reference (by default it uses a random sample) and by default provides 5 other random normal samples for comparison. For example:

```
> library(DAAG)
> qreference(m = 10)
> qreference(m = 50)
> qreference(m = 200)
```

Exercise 9

The intended use of `qreference()` is to draw a normal probability for a set of data, and place alongside it some number of normal probability plots for random normal data. For example

```
> qreference(possum$totlngth)
```

Obtain similar plots for each of the variables `taill`, `footlgth` and `earconch` in the possum data. Repeat the exercise for males and females separately

Exercise 10
Use normal probability plots to assess whether the following sets of values, all from data sets in the `DAAG` package, have distributions that seem consistent with the assumption that they have been sampled from a normal distribution?

(a) the difference `heated - ambient`, in the data frame `pair65` (*DAAG*)?

(b) the values of `earconch`, in the `possum` data frame (*DAAG*)?

(c) the values of `body`, in the data frame `Animals` (*MASS*)?

(d) the values of `log(body)`, in the data frame `Animals` (*MASS*)?

4 Boxplots – Simple Summary Information on a Distribution

In the data frame `cfseal` (*DAAG*), several of the columns have a number of missing values. A relevant question is: "Do missing and non-missing rows have similar values, for columns that are complete?"

Exercise 11
Use the following to find, for each column of the data frame `cfseal`, the number of missing values:

```
sapply(cfseal, function(x)sum(is.na(x)))
```

Observe that for `lung`, `leftkid`, `rightkid`, and `intestines` values are missing in the same six rows. For each of the remaining columns compare, do boxplots that compare the distribution of values for the 24 rows that had no missing values with the distribution of values for the 6 rows that had missing values.

Here is code that can be used to get started:

```
present <- complete.cases(cfseal)
boxplot(age ~ present, data=cfseal)
```

Or you might use the lattice function and do the following:

```
present <- complete.cases(cfseal)
library(lattice)
present <- complete.cases(cfseal)
bwplot(present ~ age, data=cfseal)
```

Exercise 12
Tabulate, for the same set of columns for which boxplots were obtained in Exercise 2, differences in medians, starting with:

```
median(age[present]) - median(age[!present]))
```

Calculate also the ratios of the two interquartile ranges, i.e.

```
IQR(age[present]) - IQR(age[!present]))
```

Part VIII
Informal Uses of Resampling Methods

1 Bootstrap Assessments of Sampling Variability

Exercise 1
The following takes a with replacement sample of the rows of `Pima.tr2`.

```
> rows <- sample(1:dim(Pima.tr2)[1], replace = TRUE)
> densityplot(~bmi, groups = type, data = Pima.tr2[rows, ], scales = list(relation = "free"),
+     xlab = "Measure")
```

Repeat, but using `anymiss` as the grouping factor, and with different panels for the two levels of `type`. Repeat for several different bootstrap samples. Are there differences between levels of `anymiss` that seem consistent over repeated bootstrap samples?

Exercise 2
The following compares density plots, for several of the variables in the data frame `Pima.tr2`, between rows that had one or more missing values and those that had no missing values.

```
> missIND <- complete.cases(Pima.tr2)
> Pima.tr2$anymiss <- c("miss", "nomiss")[missIND + 1]
> library(lattice)
> stripplot(anymiss ~ npreg + glu | type, data = Pima.tr2, outer = TRUE,
+     scales = list(relation = "free"), xlab = "Measure")
```

The distribution for `bmi` gives the impression that it has a different shape, between rows where one or more values was missing and rows where no values were missing, at least for `type=="Yes"`. The bootstrap methodology can be used to give a rough check of the consistency of apparent differences under sampling variation. The idea is to treat the sample as representative of the population, and takes repeated with replacement ("bootstrap") samples from it. The following compares the qq-plots between rows that had missing data (`anymiss=="miss"`) and rows that were complete (`anymiss=="nomiss"`), for a single bootstrap sample, separately for the non-diabetics (`type=="No"`) and the diabetics (`type=="Yes"`).

```
> rownum <- 1:dim(Pima.tr2)[1]
> chooserows <- sample(rownum, replace = TRUE)
> qq(anymiss ~ bmi | type, data = Pima.tr2[chooserows, ], auto.key = list(columns = 2))
```

Wrap these lines of code in a function. Repeat the formation of the bootstrap samples and the plots several times. Does the shift in the distribution seem consistent under repeating sampling?

Judgements based on examination of graphs are inevitably subjective. They do however make it possible to compare differences in the shapes of distributions. Here, the shape difference is of more note than any difference in mean or median.

Exercise 3
In the data frame `nswdemo` (*DAAGxtras* package), compare the distribution of `re78` for those who received work training (`trt==1`) with controls (`trt==0`) who did not.

```
> library(DAAGxtras)
> densityplot(~re78, groups = trt, data = nswdemo, from = 0, auto.key = list(columns = 2))
```

The distributions are highly skew. A few very large values may unduly affect the comparison.

Exercise 3, continued
A reasonable alternative is to compare values of `log(re78+23)`. The value 23 is chosen because it is half the minimum non-zero value of `re78`. Here is the density plot.

```
> unique(sort(nswdemo$re78))[1:3]
> densityplot(~log(re78 + 23), groups = trt, data = nswdemo, auto.key = list(columns = 2))
```

Do the distribution for control and treated have similar shapes?

Exercise 4
Now examine the displacement, under repeated bootstrap sampling, of one mean relative to the other. Here is code for the calculation:

```
> twoBoot <- function(n = 999, df = nswdemo, ynam = "re78", gp = "trt") {
+     fac <- df[, gp]
+     if (!is.factor(fac))
+         fac <- factor(fac)
+     if (length(levels(fac)) != 2)
+         stop(paste(gp, "must have 2 levels"))
+     y <- df[, ynam]
+     d2 <- c(diff(tapply(y, fac, mean)), rep(0, n))
+     for (i in 1:n) {
+         chooserows <- sample(1:length(y), replace = TRUE)
+         faci <- fac[chooserows]
+         yi <- y[chooserows]
+         d2[i + 1] <- diff(tapply(yi, faci, mean))
+     }
+     d2
+ }
> d2 <- twoBoot()
> quantile(d2, c(0.025, 0.975))
```

Note that a confidence interval should not be interpreted as a probability statement. It takes no account of prior probability. Rather, 95% of intervals that are calculated in this way can be expected to contain the true probability.

2 Use of the Permutation Distribution as a Standard

Exercise 5
If the difference is entirely due to sampling variation, then permuting the treatment labels will give another sample from the same null distribution. The permutation distribution is the distribution of differences of means from repeated samples, obtained by permuting the labels.
This offers a standard against which to compare the difference between treated and controls. Does the observed difference between treated and controls seem "extreme", relative to this permutation distribution? Note that the difference between `treat==1` and `treat==1` might go in either direction. Hence the multiplication of the tail probability by 2. Here is code:

```
> dnsw <- numeric(1000)
> y <- nswdemo$re78
> treat <- nswdemo$trt
> dnsw[1] <- mean(y[treat == 1]) - mean(y[treat == 0])
> for (i in 2:1000) {
+     trti <- sample(treat)
+     dnsw[i] <- mean(y[trti == 1]) - mean(y[trti == 0])
+ }
> 2 * min(sum(d2 < 0)/length(d2), sum(d2 > 0)/length(d2))
```

Replace `re78` with `log(re78+23)` and repeat the calculations.

Part IX
Sampling Distributions, & the Central Limit Theorem

Package: `DAAGxtras`

1 Sampling Distributions

The exercises that follow demonstrate the sampling distribution of the mean, for various different population distributions. More generally, sampling distributions of other statistics may be important.

Inference with respect to means is commonly based on the sampling distribution of the mean, or of a difference of means, perhaps scaled suitably. The ideas extend to the statistics (coefficients, etc) that arise in regression or discriminant or other such calculations. These ideas are important in themselves, and will be useful background for later laboratories and lectures.

Here, it will be assumed that sample values are independent. There are several ways to proceed.

- The distribution from which the sample is taken, although not normal, is assumed to follow a common standard form. For example, in the life testing of industrial components, an exponential or Weibull distribution might be assumed. The relevant sampling distribution can be estimated by taking repeated random samples from this distribution, and calculating the statistic for each such sample.
- If the distribution is normal, then the sample distribution of the mean will also be normal. Thus, taking repeated random samples is unnecessary; theory tells us the shape of the distribution.
- Even if the distribution is not normal, the Central Limit Theorem states that, by taking a large enough sample, the sampling distribution can be made arbitrarily close to normal. Often, given a population distribution that is symmetric, a sample of 4 or 5 is enough, to give a sampling distribution that is for all practical purposes normal.
- The final method [the "bootstrap"] that will be described is empirical. The distribution of sample values is treated as if it were the population distribution. The form of the sampling distribution is then determined by taking repeated random with replacement samples (bootstrap samples), of the same size as the one available sample, from that sample. The value of the statistic is calculated for each such bootstrap sample. The repeated bootstrap values of the statistic are used to build a picture of the sampling distribution.

  With replacement samples are taken because this is equivalent to sampling from a population in which each of the avaialble sample values is repeated an infinite number of times.

The bootstrap method obviously works best if the one available sample is large, thus providing an accurate estimate of the population distribution. Likewise, the assumption that the sampling distribution is normal is in general most reasonable if the one available sample is of modest size, or large. Inference is inevitably hazardous for small samples, unless there is prior information on the likely form of the distribution. As a rough summary:

- Simulation (repeated resampling from a theoretical distribution or distributions) is useful
  - as a check on theory (the theory may be approximate, or of doubtful relevance)
  - where there is no adequate theory
  - to provide insight, especially in a learning context.
- The bootstrap (repeated resampling from an empirical distribution or distributions) can be useful

- when the sample size is modest and uncertainty about the distributional form may materially affect the assessment of the shape of the sampling distribution;
- when standard theoretical models for the population distribution seem unsatisfactory.

The idea of a sampling distribution is wider than that of a sampling distribution of a statistic. It can be useful to examine the sampling distribution of a graph, i.e., to examine how the shape of a graph changes under repeated bootstrap sampling.

Exercise 1
First, take a random sample from the normal distribution, and plot the estimated density function:

```
> y <- rnorm(100)
> plot(density(y), type="l")
```

Now take repeated samples of size 4, calculate the mean for each such sample, and plot the density function for the distribution of means:

```
> av <- numeric(100)  # will take 100 means
> for (i in 1:100){
+ av[i] <- mean(rnorm(4))
+ }
> lines(density(av), col="red")
```

Repeat the above: taking samples of size 9, and of size 25.

Exercise 2
It is also possible to take random samples, usually with replacement, from a vector of values, i.e., from an empirical distribution. This is the bootstrap idea. Again, it may of interest to study the sampling distributions of means of different sizes. Consider the distribution of heights of female Adelaide University students, in the data frame `survey` (*MASS* package). The following takes 100 bootstrap samples of size 4, calculating the mean for each such sample:

```
> library(MASS)
> y <- na.omit(survey[survey$Sex=="Female", "Height"])
> av <- numeric(100)
> for(i in 1:100)av[i] <- mean(sample(y, 4, replace=TRUE))
```

Repeat, taking samples of sizes 9 and 16. In each case, use a density plot to display the (empirical) sampling distribution.

Exercise 3
Repeat exercise 1 above: (a) taking values from a uniform distribution (replace `rnorm(4)` by `runif(4)`); (b) from an exponential distribution with rate 1 (replace `rnorm(4)` by `rexp(4, rate=1)`).
[As noted above, density plots are not a good tool for assessing distributional form. They are however quite effective, as here, for showing the reduction in the standard deviation of the sampling distribution of the mean as the sample size increases. The next exercise but one will repeat the comparisons, using normal probability plots in place of density curves.]

Exercise 4
Laboratory 3 examined the distribution of `bmi` in the data frame `Pima2` (*MASS* package). The distribution looked as though it might have shifted, for data where one or more rows was missing, relative to other rows. We can check whether this apparent shift is consistent under repeated sampling. Here again is code for the graph for `bmi`

```
> library(MASS)
> library(lattice)
> complete <- complete.cases(Pima.tr2)
> completeF <- factor(c("oneORmore", "none")[as.numeric(complete)+1])
> Pima.tr2$completeF <- completeF
> densityplot(~bmi, groups=completeF, data=Pima.tr2, auto.key=list(columns=2))
```

Now take one bootstrap sample from each of the two categories of row, then repeating the density plot.

```
> rownum <- seq(along=complete)  # generate row numbers
> allpresSample <- sample(rownum[complete], replace=TRUE)
>   # By default, sample size is the number of vales of the first argument
> NApresSample <- sample(rownum[!complete], replace=TRUE)
> densityplot(~bmi, groups=completeF, data=Pima.tr2,
+              auto.key=list(columns=2), subset=c(allpresSample, NApresSample))
```

Wrap these lines of code in a function. Repeat the formation of the bootstrap samples and the plots several times. Does the shift in the distribution seem consistent under repeating sampling?

Exercise 5
More commonly, one compares examines the displacement, under repeated sampling, of one mean relative to the other. Here is code for the calculation:

```
> twot <- function(n=99){
+   complete <- complete.cases(Pima.tr2)
+   rownum <- seq(along=complete)  # generate row numbers
+   d2 <- numeric(n+1)
+   d2[1] <- with(Pima.tr2, mean(bmi[complete], na.rm=TRUE)-
+                                  mean(bmi[!complete], na.rm=TRUE))
+   for(i in 1:n){
+     allpresSample <- sample(rownum[complete], replace=TRUE)
+     NApresSample <- sample(rownum[!complete], replace=TRUE)
+     d2[i+1] <- with(Pima.tr2, mean(bmi[allpresSample], na.rm=TRUE)-
+                                    mean(bmi[NApresSample], na.rm=TRUE))
+ }
+   d2
+ }
>   # NB: There are n+1 values in all; the mean difference for the
>   # actual sample values, plus differences for n bootstrap samples
> ## Now estimate and plot the density function
> d2 <- twot(n=999)
> dens <- density(d2)
> plot(dens)
> ## Check the proportion of values of d2 that are less than or equal to 0
> sum(d2<0)/length(d2)

[1] 0.171
```

Those who are familiar with t-tests may recognize the final calculation as a bootstrap equivalent of the t-test.

Exercise 6
The range that contains the central 95% of values of `d2` gives a 95% confidence (or coverage) interval for the mean difference. Given that there are 1000 values in total, the interval is the range from the 26th to the 975th value, when values are sorted in order of magnitude, thus:

```
> round(sort(d2)[c(26, 975)], 2)

[1] -0.86  2.50
```

Repeat the calculation of `d2` and the calculation of the resulting 95% confidence interval, several times.

2 The Central Limit Theorem

Theoretically based t-statistic and related calculations rely on the assumption that the sampling distribution of the mean is normal. The Central Limit Theorem assures that the distribution will for a large enough sample be arbitrarily close to normal, providing only that the population distribution has a finite variance. Simulation of the sampling distribution is especially useful if the population distribution is not normal, providing an indication of the size of sample needed for the sampling distribution to be acceptably close to normal.

Exercise 7
The function `simulateSampDist()` (*DAAGxtras*) allows investigation of the sampling distribution of the mean or other stastistic, for an arbitrary population distribution and sample size. Figure 1 shows sampling distributions for samples of sizes 4 and 9, from a normal population. The function call is

```
> library(DAAGxtras)
> sampvalues <- simulateSampDist(numINsamp=c(4,9))
> plotSampDist(sampvalues=sampvalues, graph="density", titletext=NULL)
```

Experiment with sampling from normal, uniform, exponential and t_2-distributions. What is the effect of varying the value of `numsamp`?
[To vary the kernel and/or the bandwidth used by `density()`, just add the relevant arguments in the call to to `simulateSampDist()`, e.g. `sampdist(numINsamp=4, bw=0.5)`. Any such additional arguments (here, `bw`) are passed via the `...` part of the parameter list.]

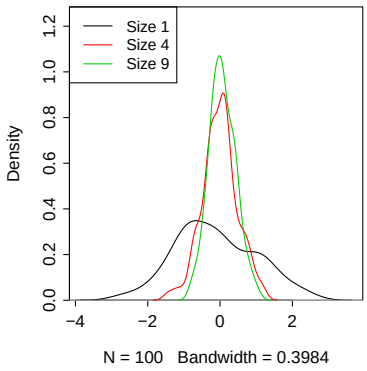

Figure 1: Empirical density curve, for a normal population and for the sampling distributions of means of samples of sizes 4 and 9 from that population.

Exercise 8
The function `simulateSampDist()` has an option (`graph="qq"`) that allows the plotting of a normal probability plot. Alternatively, by using the argument `graph=c("density","qq")`, the two types of plot appear side by side, as in Figure 2. Figure 2 is an example of its use.

```
> sampvalues <- simulateSampDist()
> plotSampDist(sampvalues=sampvalues, graph=c("density", "qq"))
```

In the right panel, the slope is proportional to the standard deviation of the distribution. For means of a sample size equal to 4, the slope is reduced by a factor of 2, while for a sample size equal to 9, the slope is reduced by a factor of 3.
Comment in each case on how the spread of the density curve changes with increasing sample size. How does the qq-plot change with increasing sample size? Comment both on the slope of a line that might be passed through the points, and on variability about that line.

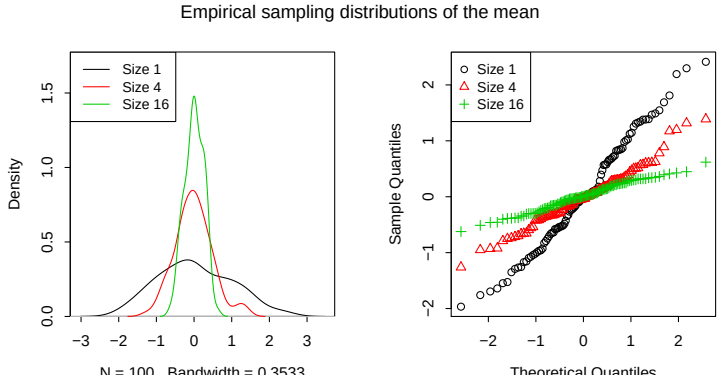

Figure 2: Empirical density curves, for a normal population and for the sampling distributions of means of samples of sizes 4 and 9, are in the left panel. The corresponding normal probability plots are shown in the right panel.

Exercise 9
How large is "large enough", so that the sampling distribution of the mean is close to normal? This will depend on the population distribution. Obtain the equivalent for Figure 2, for the following populations:

(a) A t-distribution with 2 degrees of freedom
[`rpop = function(n)rt(n, df=2)`]

(b) A log-normal distribution, i.e., the logarithms of values have a normal distribution
[`rpop = function(n, c=4)exp(rnorm(n)+c)`]

(c) The empirical distribution of heights of female Adelaide University students, in the data frame `survey` (*MASS* package). In the call to `simulateSampDist()`, the parameter `rpop` can specify a vector of numeric values. Samples are then obtained by sampling with replacement from these numbers. For example:

```
> library(MASS)
> y <- na.omit(survey[survey$Sex=="Female", "Height"])
> sampvalues <- simulateSampDist(y)
> plotSampDist(sampvalues=sampvalues)
```

How large a sample seems needed, in each instance, so that the sampling distribution is approximately normal – around 4, around 9, or greater than 9?

Appendix A
Use of the Sweave (.Rnw) Exercise Files

The following is a wrapper file, called **wrap-basic.Rnw**, for the sets of exercises generated by processing **rbasics.Rnw** and **rpractice.Rnw**. It is actually pure LaTeX, so that it is not strictly necessary to process it through R's `Sweave()` function.

```
\documentclass[a4paper]{article}

\usepackage{url}
\usepackage{float}
\usepackage{exercises}
\usepackage{nextpage}
\pagestyle{headings}
\title{``Basic R Exercises'' and ``Further Practice with R''}
\author{John Maindonald}
\usepackage{Sweave}
\begin{document}

\maketitle
\tableofcontents

\cleartooddpage

\include{rbasics}
\cleartooddpage
\setcounter{section}{0}
\include{rpractice}
\end{document}
```

To create a LaTeX file from this, ensure that **wrap-basic.Rnw** is in the working directory, and do:

```
> Sweave("wrap-basic")
```

This generates the file **wrap-basic.tex**

Now process **rbasic.Rnw** and **rpractice.Rnw** through `Sweave()`:

```
> Sweave("rbasics", keep.source=TRUE)
> Sweave("rpractice", keep.source=TRUE)
```

This generates files **rbasics.tex** and **rpractice.tex**, plus pdf and postscript versions of the graphics files. Specifying `keep.source=TRUE` ensures that comments will be retained in the code that appears in the LaTeX file that is generated.

Make sure that the file **Sweave.sty** is in the LaTeX path. A simple way to ensure that it is available is to copy it into your working directory. Process **wrap-basic.tex** through LaTeX to obtain the pdf file **wrap-basic.pdf**.

You can find the path to the file **Sweave.sty** that comes with your R installation by typing:

```
> paste(R.home(), "share/texmf/", sep="/")  # NB: Output is for a MacOS X system

[1] "/Library/Frameworks/R.framework/Resources/share/texmf/"
```