APTS 2010–11 preliminary material

Computer-Intensive Statistics

Hints for Preliminary Exercises

This document gives some hints on how to approach the exercises – a step-by-step guide is also available.

Bootstrapping

Efron (1979, 1982) gives the following data on admissions to 15 US Law Schools

```
LSAT:
576 635 558 578 666 580 555 661 651 605 653 575 545 571 594
GPA:
3.39 3.30 2.81 3.03 3.44 3.07 3.00 3.43 3.36 3.13 3.12 2.74 2.76 2.88 2.96
```

Ex 1 Enter these data into R as a data frame `law`, plot them and compute the correlation coefficient.

[There are many ways to enter the data. Here is one of the simplest that you can copy-and-paste to R. Do not omit the blank lines.

```
LSAT <- scan()
576 635 558 578 666 580 555 661 651 605 653 575 545 571 594

GPA <- scan()
3.39 3.30 2.81 3.03 3.44 3.07 3.00 3.43 3.36 3.13 3.12 2.74 2.76 2.88 2.96

law <- data.frame(LSAT, GPA)
law # print it for checking.
cor(law) # gives the full correlation matrix
cor(law)[1,2] # the element we want, or
cor(LSAT, GPA)
```

]

Use Fisher's theory to give a 95% confidence interval for ρ assuming normality.

[You can do this by hand, or use R as your calculator.]

Ex 2 Create a bootstrap sample. Here is one simple way to do it:

```
law[sample(1:15, replace=TRUE), ]
```

Now compute its correlation coefficient.

Repeat 1000 times and put in a vector, making use of the `replicate` function.

[

```
rhostar <- replicate(1000, cor(law[sample(1:15, replace=TRUE), ])[1,2])
```

]

Now take a look at the bootstrap distribution on both correlation and `atanh` scales.

[Here are some ways to look at the results using a histogram and a kernel density estimate. If you are unfamiliar with these, look them up in Venables & Ripley (2002).

```
hist(rhostar, prob=TRUE)
lines(density(rhostar, bw="SJ"))
hist(atanh(rhostar), prob=TRUE)
lines(density(atanh(rhostar), bw="SJ"))
```

]

Ex 3 [The hint is in the question itself.]

Ex 4 How about a confidence interval for ρ? We can use function `boot.ci` to produce several different confidence intervals – do so on both correlation and `atanh` scale.

[You need to use the function `boot.ci`. Here's a start

```
boot.ci(out, type=c("norm", "basic", "perc", "bca"))
```

You may well get a warning, and need to do more runs.

Function `boot.ci` can do the transformations for you.]

Ex 5 Cox and Lewis (1966) reported 799 time intervals between pulses on a nerve fibre. The dataset can be downloaded from
`http://www.stat.cmu.edu/~larry/all-of-statistics/=data/nerve.dat`
and is used by Wasserman (2004, pp. 98, 111).

[To get the data into R use

```
nerve <-
scan("http://www.stat.cmu.edu/~larry/all-of-statistics/=data/nerve.dat")
summary(nerve) # a quick check
```

]

Use the bootstrap to get confidence intervals for the median and skewness of these data. You can either write your own R function to compute the skewness, or get one from contributed package `e1071` and about ten others. To see if you have one installed, use

```
help.search("skewness")
```

[If you do not, here is a simple version

```
skewness <- function (x) mean((x - mean(x))^3)/sd(x)^3
```

You need to use functions `boot` and `boot.ci` as in the the previous exercises.

The data have clearly been recorded to the nearest `0.01`. See `jitter` for how to fake the continuous measurements.

Hint: note the comments about the smoothed bootstrap.]

Spatial patterns and MCMC

Ex 6 The Strauss process (1) and ways to simulate it are contained in R packages `spatial` and `spatstat`.

Let us consider the Swedish pines data from Ripley (1981), described in Venables & Ripley (2002, §15.3). Retrieve and plot it in R by

```
library(MASS)
library(spatial)
pines <- ppinit("pines.dat")
eqscplot(pines, xlim = c(0, 10), ylim = c(0, 10), xlab = "", ylab = "")
```

How many points are there? How might you describe the pattern? By the way, the coordinates are in metres.

Venables & Ripley (2002, p. 443) suggest that $R = 0.7$ and $c = 0.15$ are reasonable estimates. Use function `Strauss` to simulate with these parameter values and compare a plot with the real data.

[Let's compare the data and 3 simulations:

```
par(mfrow=c(2,2), mar=c(3,3,1,1))
eqscplot(pines, xlim = c(0, 10), ylim = c(0, 10), xlab = "", ylab = "")
for (i in 1:3) {
    sp <- Strauss(n = 72, c = 0.15, r = 0.7)
    eqscplot(sp, xlim = c(0, 10), ylim = c(0, 10), xlab = "", ylab = "")
}
```

]

Ex 7 [This question contains the code to be used. For part (f) look at the help page for function `rcauchy`, or in the references in Appendix A.]