Lab 4

36-350, Statistical Computing

SOLUTIONS

Load the data as before:

```
library(MASS)
data(cats)
hearts <- cats$Hwt
```

The last line is not necessary, but saves typing later.

1. SOLUTION: Remember that when we have a vector, we can form a new vector which omits one or more positions by using a negative index. (See chapter 2 of the textbook.)

   ```
   > hearts.3 <- hearts[1:3]
   > hearts.3
   [1] 7.0 7.4 9.5
   > hearts.3[-2]
   [1] 7.0 9.5
   ```

   Now we just do this three times:

   ```
   > fit.1 <- gamma.est(hearts.3[-1])
   > fit.2 <- gamma.est(hearts.3[-2])
   > fit.3 <- gamma.est(hearts.3[-3])
   > fitted.a.values <- c(fit.1$a,fit.2$a,fit.3$a)
   > fitted.s.values <- c(fit.1$s,fit.2$s,fit.3$s)
   > n <- length(hearts.3)
   > sqrt(((n-1)^2/n)*var(fitted.a.values)) # Jackknife standard error for a
   [1] 413.9912
   > sqrt(((n-1)^2/n)*var(fitted.s.values)) # and for s
   [1] 0.21679
   ```

2. SOLUTION: We basically wrap what we just did up into a function, with a little loop inside.

   ```
   # Jackknife standard errors for estimates of gamma distribution parameters
   # Inputs: vector of data values (data)
   ```

```
# Outputs: vector with two components, giving standard errors in shape and
  # scale parameters
gamma.jackknife <- function(data) {
  n <- length(data)
  # Prepare a matrix to store the re-estimates on partial data
  jackknifed.ests <- matrix(NA,nrow=2,ncol=n)
  rownames(jackknifed.ests) = c("a","s")
  # Omit each point in turn, and do estimation on the rest
  for (omitted.point in 1:n) {
    # Presumes gamma.est() exists in this environment and takes a data vector
    fit <- gamma.est(data[-omitted.point])
    # Presumes that gamma.est returns a list of two numbers named "a" and "s"
    jackknifed.ests["a",omitted.point] <- fit$a
    jackknifed.ests["s",omitted.point] <- fit$s
  }
  # Take the variance of each row of the re-estimates
  variance.of.ests <- apply(jackknifed.ests,1,var)
  # Scale up the variances to get the jackknife variance-of-estimation
  jackknifed.vars <- ((n-1)^2/n)*variance.of.ests
  # Standard error is the  square root of variance-of-estimation
  jackknifed.stderrs <- sqrt(jackknifed.vars)
  return(jackknifed.stderrs)
}
```

There are several ways to do this without using an explicit loop, but the easiest of them use functions like `sapply`, which we will come to later.

3. Solution:

```
> gamma.jackknife(hearts.3)
        a         s
413.99123   0.21679
```

The function thus works as it should on the little test case where we calculated everything ourselves.

4. Solution

```
> gamma.jackknife(hearts)
         a          s
2.74062279 0.07829436
```

Realistically, there's no point in reporting these beyond three significant digits (since the data is no more precise):

```
> signif(gamma.jackknife(hearts),3)
     a      s
2.7400 0.0783
```

5. Solution: First the estimates (which you also got in the last lab).

```
> female.est <- gamma.est(hearts[cats$Sex=="F"])
> male.est <- gamma.est(hearts[cats$Sex=="M"])
> female.est
$a
[1] 45.93998
$s
[1] 0.2003076
> male.est
$a
[1] 19.83576
$s
[1] 0.5708216
```

And now the standard errors:

```
> female.stderrs <- gamma.jackknife(hearts[cats$Sex=="F"])
> male.stderrs <- gamma.jackknife(hearts[cats$Sex=="M"])
> female.stderrs
          a           s
10.69733598  0.04223307
> male.stderrs
         a          s
3.39861612 0.09279273
```

6. Solution:

```
> shape.diff <- female.est$a - male.est$a
> scale.diff <- female.est$s - male.est$s
> shape.diff.stderr <- sqrt(female.stderrs["a"]^2+male.stderrs["a"]^2)
> scale.diff.stderr <- sqrt(female.stderrs["s"]^2+male.stderrs["s"]^2)
> shape.diff
[1] 26.10422
> scale.diff
[1] -0.370514
> shape.diff.stderr
       a
11.22424
> scale.diff.stderr
        s
0.1019516
```

7. Solution: Strictly speaking, we do not have enough information to say anything about statistical significance. *If* the fluctuations in the estimates are roughly Gaussian, then seeing differences like these, of two or three standard errors,

```
> shape.diff/shape.diff.stderr
     a
2.3257
> scale.diff/scale.diff.stderr
        s
-3.634215
```

is reasonably unlikely, at the 5% or even 1% level. But why should we expect Gaussian fluctuations here?