Bootstrap and Jackknife Calculations in R

Version 6 April 2004

These notes work through a simple example to show how one can program **R** to do both jackknife and bootstrap sampling. We start with bootstrapping.

Bootstrap Calculations

R has a number of nice features for easy calculation of bootstrap estimates and confidence intervals. To see how to use these features, consider the following 25 observations:

8.26	6.33	10.4	5.27	5.35	5.61	6.12	6.19	5.2	7.01	8.74	7.78
7.02	6	6.5	5.8	5.12	7.41	6.52	6.21	12.28	5.6	5.38	6.6
8.74											

Suppose we wish to estimate the coefficient of variation, CV = $\sqrt{\text{Var}}/\overline{x}$. Let's do this with a bootstrap estimator.

First, let's put the data into a vector, which we will call **x**,

```
> x <-c(8.26, 6.33, 10.4, 5.27, 5.35, 5.61, 6.12, 6.19, 5.2,
7.01, 8.74, 7.78, 7.02, 6, 6.5, 5.8, 5.12, 7.41, 6.52, 6.21,
12.28, 5.6, 5.38, 6.6, 8.74)
```

Now let's define a functon in **R**, which we will call **CV**, to compute the coefficient of variation,

```
> CV <- function(x) sqrt(var(x))/mean(x)
```

So, let's compute the CV

```
> CV(x)
[1] 0.2524712
```

To generate a single bootstrap sample from this data vector, we use the command

```
> sample(x,replace=T)
```

which generates a bootstrap sample of the data vector **x** by sampling with replacement. Hence, to compute the CV using a single bootstrap sample,

```
> CV(sample(x,replace=T))
[1] 0.2242572
```

The particular value that **R** returns for you will be different as the sample is random.

Some other useful commands:

- **> sum(x)** returns the sum of the elements in **x**
- **> mean(x)** returns the mean of the elements in **x**
- **> var(x)** returns the sample variance, i.e., $\sum_i (x - \overline{x})^2/(n-1)$
- **> length(x)** returns the number of items in **x** (i.e., the sample size n)

Note that the **sum** command is fairly general, for example

> sum((x-mean(x))^2) computes $\sum_i (x - \overline{x})^2$

So, lets now generate 1000 bootstrap samples. We first need to specify a vector of real values of lenght 1000, which we will call **boot**

```
> boot <-numeric(1000)
```

We now generate 1000 samples, and assign the CV for bootstrap sample i as the ith element in the vector **boot**, using a **for** loop

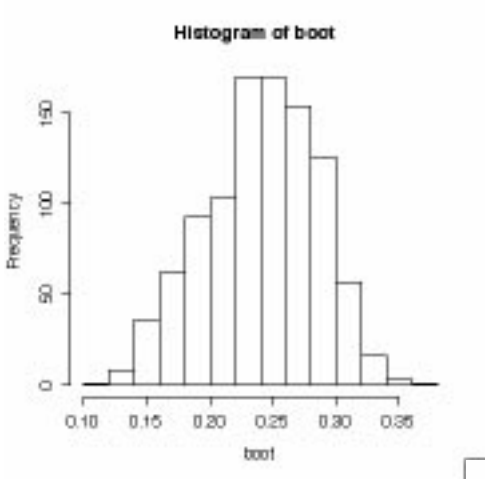
```
for (i in 1:1000) boot[i] <- CV(sample(x,replace=T))
```

The mean and variance of this collection of bootstrap samples are easily obtained using the **mean** and **var** commands (again, your values may differ),

```
> mean(boot)
[1] 0.2404653
> var(boot)
[1] 0.00193073
```

A plot of the histogram of these values follows using

```
hist(boot)
```



Likewise, the value corresponding to the (say) upper 97.5

```
> quantile(boot,0.975)
[1] 0.3176385
```

while the value corresponding to the lower 2.5% follows from

```
> quantile(boot,0.025)
[1] 0.153469
```

Recall from the notes that the estimate of the bias is given by the difference between the mean of the bootstrap values and the initial estimate,

```
> bias <- mean(boot) - CV(x)
```

and an bootstrap-corrected estimate of the CV is just the original estimate minus the bias,

```
> CV(x) - bias
[1] 0.2644771
```

Assuming normality, the approximate 95% confidence interval is given by

$$\widehat{CV} \pm 1.96\sqrt{\text{Var(bootstrap)}}$$

(or adjusting for the bias an lower and upper values of

```
> CV(x) - bias - 1.96*sqrt(var(boot))
[1] 0.1783546
> CV(x) - bias + 1.96*sqrt(var(boot))
[1] 0.3505997
```

Efron's confident limit (Equation 11 on resampling notes) has an upper and lower value of

```
> quantile(boot,0.975)
[1] 0.3176385
```

and

```
> quantile(boot,0.025)
[1] 0.153469
```

While Hall's confidence limits (Equation 12) has an upper and lower value of

```
> 2*CV(x) - quantile(boot,0.025)
[1] 0.3514734
```

and

```
> 2*CV(x) - quantile(boot,0.975)
[1] 0.1873039
```

Jackknife Calculations

We now turn to jackknifing the sample. Recall from the randomization notes that this involves two steps. First, we generate a jackknife sample which has value x_i removed and then compute the ith partial estimate of the test statistic using this sample,

$$\widehat{\theta}_i(x_1 \cdots x_{i-1}, x_i, \cdots x_n)$$

We then turn this ith partial estimate into the ith pseudovalue $\widehat{\theta}_i^*$ using (Equation 5c in random notes)

$$\widehat{\theta}_i^* = n\widehat{\theta} - (n-1)\widehat{\theta}_i$$

where $\widehat{\theta}$ is the estimate using the full data.

Let's see how to code this in **R** using the previous vector **x** of data with our test statistic again being the coefficient of variation (and hence our function **CV** previously defined). We first focus on generating the ith partial estimate and ith pseudovalue. We need to take the original data vector **x** and turn it into a vector (which we denote **jack**) of lenght $n-1$ as follows. First, we need to specify to **R** that we are creating the jackknife sample vector of the $n-1$ sampled points

```
jack <- numeric(length(x)-1)
```

As before, we will use the command **lenght(x)** in place of n. We also need to specify to **R** that we will be generating a vector **pseudo** of the n pseudovalues

```
pseudo <- numeric(length(x))
```

Next, we need to fill in the elements of the **jack** sample vector as follows. For $j < i$, the jth element of **jack** is the same as the jth element of **x**; for $j = i$ we exclude the value of **x**, while for $j > i$, the $j-1$th element of **jack** is the jth element of **x**. We can state all this using a logical **if .. else** statement within a **for** loop,

```
for (j in 1:length(x)) if(j < i) jack[j] <- x[j]
else if(j > i) jack[j-1] <- x[j]
```

We can then compute the ith pseudovalue (for the CV) as follows:

```
pseudo[i] <- lenght(x)*CV(x) -(lenght(x)-1)*CV(jack)
```

Finally, we top this all off by looping through the n possible i values, giving the final code as

```
jack <- numeric(length(x)-1)
pseudo <- numeric(length(x))
for (i in 1:length(x))
{ for (j in 1:length(x))
{if(j < i) jack[j] <- x[j] else if(j > i) jack[j-1] <- x[j]}
pseudo[i] <- length(x)*CV(x) -(length(x)-1)*CV(jack)}
```

Note the use of the parenthesis ({, }) to delimit the appropriate elements in each loop. The mean and variance of the pseudovalues are easily found using

```
> mean(pseudo)
[1] 0.2617376
> var(pseudo)
[1] 0.07262871
```

Likewise, a histogram of the pseudovalues is generated using

```
hist(pseudo)
```

Recall that the mean of the pseudovalues is the bootstrap estimator, while var(pseudo)/n is the variance of this estimator,

```
>var(pseudo)/length(x)
[1] 0.002905148
```

An approximate 95% confidence interval is given by
mean(pseudo) $\pm t_{0.975,n-1}\sqrt{\text{var(pseudo)}/n}$

Using **R**, the upper and lower limits become

```
> mean(pseudo) + qt(0.975,length(x)-1)*sqrt(var(pseudo)/length(x))
[1] 0.3729806
> mean(pseudo) - qt(0.975,length(x)-1)*sqrt(var(pseudo)/length(x))
[1] 0.1504947
```

Giving the approximate 95% jackknife confidence interval as 0.150 to 0.372.

Here's a summary of the various estimated values, variances, and confidence intervals

Method	Estimated CV	Variance	95% interval
Original Estimate	0.252		
Jackknife	0.262	0.0029	0.150 - 0.373
Bootstrap	0.264	0.0019	
Bootstrap (normality)			0.178 - 0.351
Bootstrap (Efron)			0.153 - 0.318
Bootstrap (Hall)			0.187 - 0.351