Bootstrap

true

Introduction

Both Bootstrap and Jackknife are resampling methods used for estimating standard errors and building confidence intervals.

Jackknife

The jackknife is a linear approximation of the bootstrap. The jackknife estimation of bias is defined by:

$$\hat{bias}_{jack} = (n-1)(\hat{\theta}_{(\cdot)} - \hat{\theta})$$

where

$$\hat{\theta}_{(\cdot)} = \sum_{i=1}^{n} \hat{\theta}_{(i)}/n$$

and

$$\hat{\theta}_{(i)} = T(x_1, x_2, ..., x_{i-1}, x_{i+1}, ..., x_n)$$

where T is a statistic. The standard error for jackknife is:

$$\hat{se}_{jack} = [\frac{n-1}{n}\sum(\hat{\theta}_{(i)} - \hat{\theta}_{(\cdot)})]$$

The *pseudovalue* is defined as: (See An Introduction to the Bootstrap, 1993, Bradley Efron, R.J. Tibshirani)

$$\tilde{\theta}_i = n\tilde{\theta} - (n-1)\hat{\theta}_i$$

And the standard deviation expressed in terms of psudovalue is:

$$\hat{se}_{jack} = \sqrt{\sum_{i=1}^{n}(\tilde{\theta}_i - \tilde{\theta})^2/(n-1)n}$$

Bias-corrected jackknife estimate is given by:

$$\hat{\theta}_{jack} = n\hat{\theta} - (n-1)\hat{\theta}_{(\cdot)}$$

And one jacknife confidence interval is given by :(See here)

$$\tilde{\theta} \pm t_{n-1}^{1-\alpha}\hat{se}_{jack}$$

```r
Jackknife<-function(v1,statfunc=sd, alpha=0.05)
{
  n1<-length(v1)
  jackvec<-NULL
  mu0<-statfunc(v1)
  for(i in 1:n1){
    mua<-statfunc(v1[-i])
    jackvec<-c(jackvec, n1*(mu0)-(n1-1)*mua) # psudovalue
  }
  jackbias<-(n1-1)*(mean(jackvec)-mu0)
  jacksd<-sqrt(sum( (jackvec - mean(jackvec) )^2)  / (n1*(n1-1)) )

  list(mu0=mu0,jackbias=jackbias,jacksd=jacksd, interval=c(mu0-qt(1-alpha, df=n1-1)*jacksd,
                                                           mu0+qt(1-alpha, df=n1-1)*jacksd))
}
```

Bootstrap

For Bootstrap, the variance is defined as:(From *All of Statistics*)

$$v_{boot} = \frac{1}{B}\sum_{b=1}^{B}(T^*_{n,b} - \frac{1}{B}\sum_{r=1}^{B}T^*_{n,r})^2$$

and $\hat{se}_{boot} = \sqrt{v_{boot}}$ where B is the number of boostrap iteration, $T^*_{n,r}$ is the statistic calculated from rth sampling.

We use Pivital Confidence Sets, that is, given bootstrap results, our condifence interval is expressed as:

$$C_n = (2\hat{\theta}_n - \hat{\theta}^*_{1-\alpha/2}, 2\hat{\theta}_n - \hat{\theta}^*_{\alpha/2})$$

where $\hat{\theta}_n = T(\hat{F}_n)$,

```
my.bootstrapci<-function(vec0,nboot=10000,alpha=0.05,statfun=mean)
{
  #extract sample size, mean and standard deviation from the original data
  n0<-length(vec0)
  mean0<-statfun(vec0)
  #sd0<-sqrt(var(vec0))
  # create a vector to store the location of the bootstrap studentized deviation vector
  bootvec<-NULL

  #create the bootstrap distribution using a for loop
  for( i in 1:nboot){
    vecb<-sample(vec0,replace=T)
    #create mean and standard deviation to studentize
    meanb<-statfun(vecb)
    # sdb<-sqrt(var(vecb))
    #note since resampling full vector we can use n0 for sample size of vecb
    bootvec<-c(bootvec, meanb)
  }
  bias<- mean(bootvec)-mean0
  #Calculate lower and upper quantile of the bootstrap distribution
  lq<-quantile(bootvec,alpha/2)
  uq<-quantile(bootvec,1-alpha/2)
  #ADD the other two confidence intervals.
  #incorporate into the bootstrap confidence interval (what algebra supports this?) and output result
  #LB<-mean0-(sd0/sqrt(n0))*uq
  #UB<-mean0-(sd0/sqrt(n0))*lq
  #since I have the mean and standard deviation calculate the normal confidence interval here as well
  se=sqrt(sum((bootvec - mean(bootvec))^2) / nboot)
  NLB<-mean0-sd(vec0)*qnorm(1-alpha/2)
  NUB<-mean0+sd(vec0)*qnorm(1-alpha/2)
  list(bias=mean(bootvec)-mean0
    ,normal.confidence.interval=c(NLB,NUB), Pivotal.Interval=c(2*mean0-uq, 2*mean0-lq),
       Percentile.Interval=c(quantile(bootvec, alpha/2), quantile(bootvec, 1-alpha/2)))
}
```

Simulation

```
simu<-function(mu.val=3,n=30,nsim=1000)
{
  #create coverage indicator vectors for bootstrap and normal
  cvec.boot<-NULL
  cvec.norm<-NULL
  cvec.perc<-NULL
  cvec.jack<-NULL
  #calculate real mean
  mulnorm<-(exp(mu.val+1/2))
  #run simulation
  for(i in 1:nsim){
    if((i/100)==floor(i/100)){
      print(i)
      #let me know computer hasnt died
    }
    #sample the simulation vector
    vec.sample<-rlnorm(n,mu.val)
    #bootstrap it
    boot.list<-my.bootstrapci(vec.sample)
    boot.conf<-boot.list$Pivotal.Interval #aka pivot interval
    norm.conf<-boot.list$normal.confidence.interval
    percentile.conf<-boot.list$Percentile.Interval
    jack<-Jackknife(vec.sample, statfun=mean)
    jack.interval<-jack$interval
    #calculate if confidence intervals include mu
    #count up the coverage by the bootstrap interval
    cvec.boot<-c(cvec.boot,(boot.conf[1]<mulnorm)*(boot.conf[2]>mulnorm))
    #count up the coverage by the normal theory interval
    cvec.norm<-c(cvec.norm,(norm.conf[1]<mulnorm)*(norm.conf[2]>mulnorm))
    cvec.perc<-c(cvec.perc, (percentile.conf[1]< mulnorm)*(percentile.conf[2]>mulnorm) )
    cvec.jack<-c(cvec.jack, (jack.interval[1]< mulnorm)*(jack.interval[2]>mulnorm))
  }
  #calculate and output coverage probability estimates
  return(list(boot.coverage=(sum(cvec.boot)/nsim),norm.coverage=(sum(cvec.norm)/nsim), perc.coverage
       =(sum(cvec.perc)/nsim), jack.coverage=(
         sum(cvec.jack)/nsim) ) )
}
```

Let's test it against lognormal distribution:

```
cat("n=10\n")
```

```
## n=10
```

```r
simu(n=10)
```

```
## [1] 100
## [1] 200
## [1] 300
## [1] 400
## [1] 500
## [1] 600
## [1] 700
## [1] 800
## [1] 900
## [1] 1000

## $boot.coverage
## [1] 0.764
##
## $norm.coverage
## [1] 0.991
##
## $perc.coverage
## [1] 0.818
##
## $jack.coverage
## [1] 0.806
```

```r
cat("n=30\n")
```

```
## n=30
```

```r
simu(n=30)
```

```
## [1] 100
## [1] 200
## [1] 300
## [1] 400
## [1] 500
## [1] 600
## [1] 700
## [1] 800
## [1] 900
## [1] 1000

## $boot.coverage
## [1] 0.852
##
## $norm.coverage
## [1] 1
##
## $perc.coverage
## [1] 0.892
##
## $jack.coverage
## [1] 0.841
```

```r
cat("n=100\n")
```

```
## n=100
```

```r
simu(n=100)
```

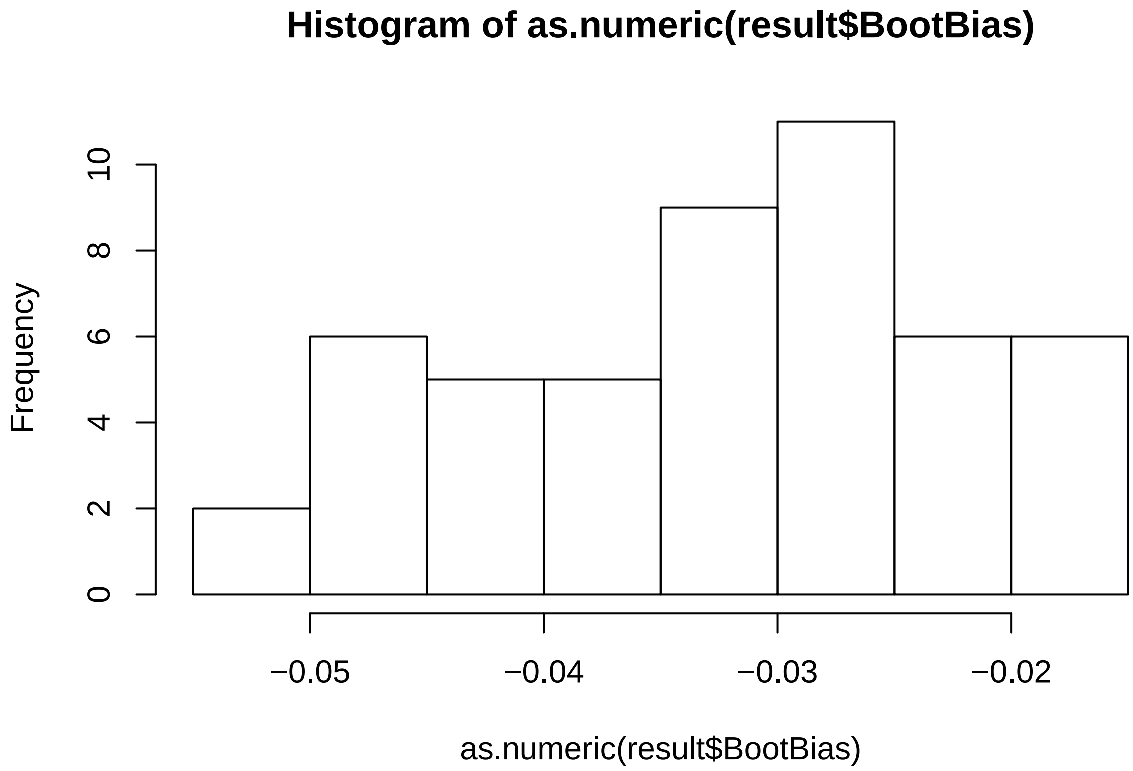
```
## [1] 100
## [1] 200
## [1] 300
## [1] 400
## [1] 500
## [1] 600
## [1] 700
## [1] 800
## [1] 900
## [1] 1000
```

```
## $boot.coverage
## [1] 0.881
##
## $norm.coverage
## [1] 1
##
## $perc.coverage
## [1] 0.9
##
## $jack.coverage
## [1] 0.853
```

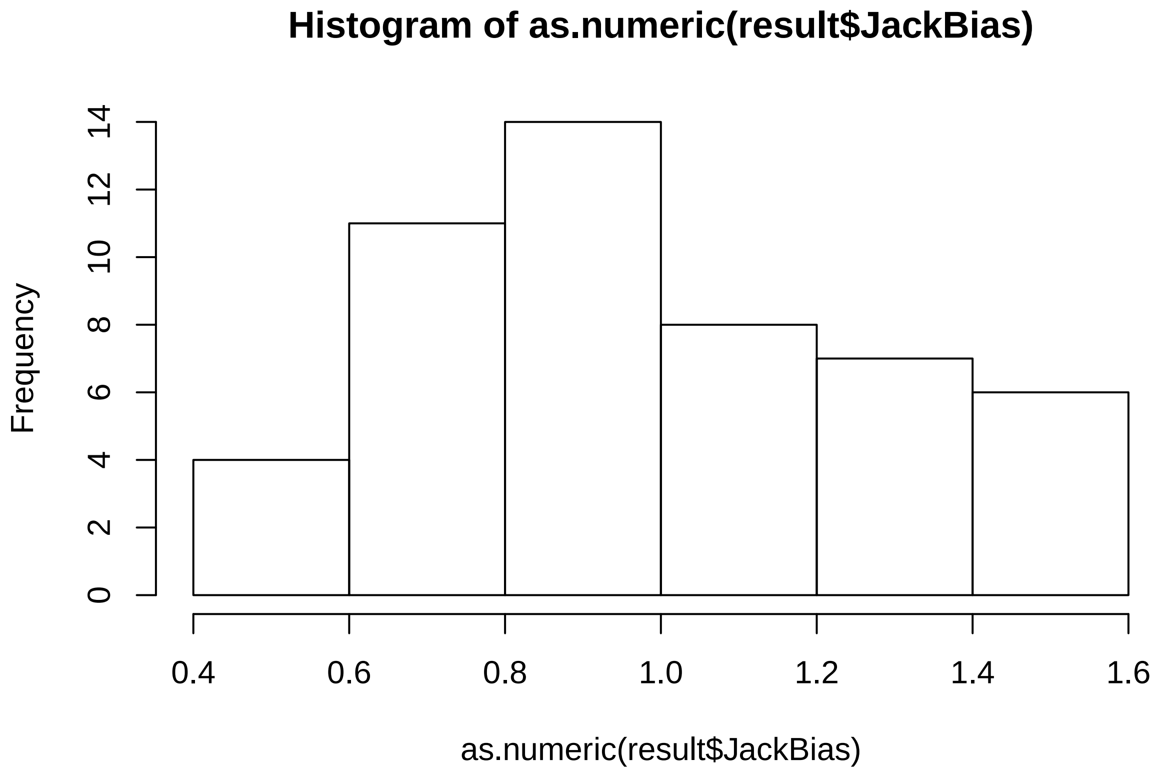
```r
#sample variance divided by n
sam_var<-function(x){
  return( sum( (x-mean(x))^2) / length(x)
    )
}
```

```r
simu2<-function(mu=0, sd=1 , n=30, nsim=1000){
  boot.bias<-NULL
  jack.bias<-NULL
  for(i in 1:nsim){
    vec.sample=rnorm(n, mean = mu, sd)
    boot.bias<-c(boot.bias, my.bootstrapci(vec0 = vec.sample, statfun = sam_var)$bias )
    jack.bias<-c(jack.bias, Jackknife(vec.sample, statfunc = sam_var)$jackbias )


  }
  return( list(
    BootBias=boot.bias, JackBias=jack.bias

  ) )
}
```

```
result<-simu2(nsim=50)
hist( as.numeric(result$BootBias) )
```



```
hist( as.numeric(result$JackBias) )
```


Histogram of as.numeric(result$JackBias)
Frequency
0
2
4
6
8
10
12
14
0.4
0.6
0.8
1.0
1.2
1.4
1.6
as.numeric(result$JackBias)