Bootstrapping

Simon Wood, University of Edinburgh, U.K.

Sampling distributions

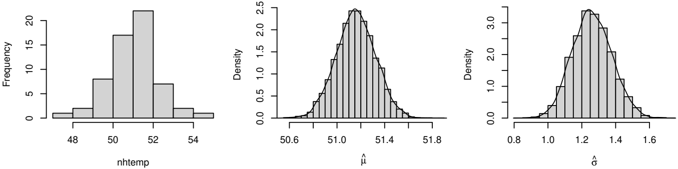
- When we talk about the distribution of an estimator, $\hat{\boldsymbol{\theta}}$, we are considering the randomness that $\hat{\boldsymbol{\theta}}$ inherits from the randomness in the data $\mathbf{y}$.
- If we repeatedly replicated the process of gathering $\mathbf{y}$ and estimating $\boldsymbol{\theta}$ we would get a somewhat different $\hat{\boldsymbol{\theta}}$ each time.
- This replicate to replicate variability is the *sampling distribution* of $\hat{\boldsymbol{\theta}}$. The basis for inferences about $\boldsymbol{\theta}$.
- When considering maximum likelihood estimation, we looked at theoretical approximations for the distribution of $\hat{\boldsymbol{\theta}}$.
- An alternative is to repeatedly simulate the data generation and estimation process to assess $\hat{\boldsymbol{\theta}}$'s distribution. *Bootstrapping*.

Parametric bootstrapping

- Suppose we have a model specifying a p.d.f. $\pi_\theta(\mathbf{y})$ for our data.
- Let $\hat{\boldsymbol{\theta}}$ be the MLE or other estimate of $\boldsymbol{\theta}$.
- Repeatedly simulate $\mathbf{y}^*$ from $\pi_{\hat{\theta}}(\mathbf{y})$, and for each $\mathbf{y}^*$ compute $\hat{\boldsymbol{\theta}}^*$.
- The set of $\hat{\boldsymbol{\theta}}^*$ are a *parametric bootstrap* approximation to the sampling distribution of $\hat{\boldsymbol{\theta}}$.
- Typically the variability between the $\hat{\boldsymbol{\theta}}^*$ is a reasonable approximation to the true sampling variability of $\hat{\boldsymbol{\theta}}$.
- But the distribution of the bootstrap sample will be centred on $\hat{\boldsymbol{\theta}}$, not the true $\boldsymbol{\theta}$.

Simple parametric bootstrap in R

```
mu <- mean(nhtemp)   ## mean temperature
sd <- sd(nhtemp)     ## standard dev temp
n <- length(nhtemp) ## number of data
nb <- 10000          ## number of bootstrap samples
sdb <- mub <- rep(0,nb) ## replicate estimates
for (i in 1:nb) { ## parametric bootstrap loop
  yb <- rnorm(n,mu,sd)  ## generate replicate data
  mub[i] <- mean(yb)    ## estimate mean
  sdb[i] <- sd(yb)      ## and standard deviation
}
```



Confidence intervals

- To illustrate that bootstrapping gets the variability right, compare the bootstrap standard deviation for $\hat{\mu}$ to its theoretical value.

  ```
  > c(sd(mub),sd(nhtemp)/sqrt(n))
  [1] 0.1643081 0.1633892
  ```

- For more or less symmetrical sampling distributions *bootstrap percentile confidence intervals* can be computed.

  ```
  > quantile(mub,c(.025,.975))
      2.5%    97.5%
  50.83482 51.48179
  > quantile(sdb,c(.025,.975))
      2.5%    97.5%
  1.042289 1.492834
  ```

- Obviously the parametric bootstrap also works with more complex models.

Generating random deviates in R

- R has built in functions for generating pseudorandom numbers from a wide variety of standard distributions.
- All these functions start with an '`r`', take the number of deviates required as the first argument and further parameters of the distribution as further arguments.
- Examples are: `rnorm`, `rgamma`, `rpois`, `rbinom`, `rgeom`...
- All work by starting with uniform deviates[*] and transforming them. There are a variety of fast methods for doing this for specific distributions, but there is also a generic method:
  1. Generate a deviate, u, from $U(0,1)$.
  2. Evaluate the inverse of the cumulative distribution function (CDF) for the distribution of interest (the *quantile function*) at u.
  3. The result is a random draw from the distribution of interest.

[*] see appendix C of *Core Statistics* for more on this.

CDFs and quantile functions

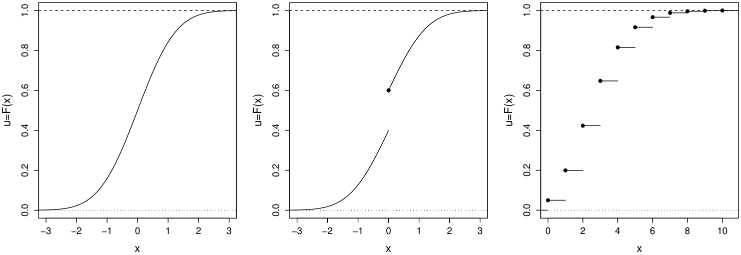
- For r.v. X, the CDF is defined as $F(x) = \Pr(X \le x)$.
- The *quantile function* is the inverse of the CDF defined as $F^{-}(u) = \min(x | F(x) \ge u)$ (usual inverse for continuous F).
- If $X \sim F$ and $F(x)$ is continuous then $F(X) \sim U(0,1)$.
  Proof: Defining $U = F(X)$ and $u = F(x)$, we have

  $$F(x) = \Pr(X \le x) = \Pr(F(X) \le F(x)) \Rightarrow u = \Pr(U \le u)$$

  but the r.h.s. is just the CDF of a $U(0,1)$ r.v.
- Similarly $U \sim U(0,1) \Rightarrow F^{-}(U) \sim F$, so we can generate from any distribution with a F^{-} function (continuous or not). In R
  - Quantile functions start with a '`q`'. e.g. `qnorm`, `qpois` …
    So `qnorm(runif(3),1,2)` equivalent to `rnorm(3,1,2)`.
  - CDFs start with a '`p`'. e.g. `pnorm`, `ppois`, `pgamma` …

Example CDFs

- Here are three example CDFs for
  1. $N(0, 1)$.
  2. A mixture: 0.8 probability of drawing from $N(0, 1)$ and 0.2 probability of a zero.
  3. A Poisson with mean 3.



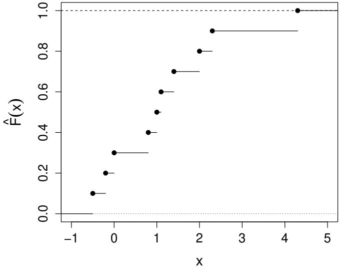
- Recall, $F(x) = \Pr(X \leq x)$ and $F^{-}(u) = \min(x|F(x) \geq u)$.

The empirical CDF

- Let $x_1, x_2, \dots x_n$ be sampled from a distribution with unknown CDF, $F(x)$.
- Since $F(x) = \Pr(X \leq x)$, we can estimate F by estimating the probabilities on the right, from our sample:

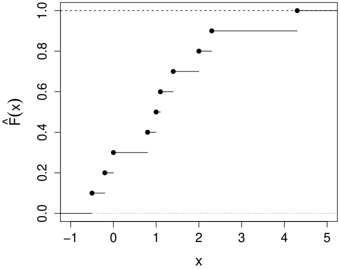
$$\hat{F}(x) = n^{-1} \sum_i \mathbb{I}(x_i \leq x)$$

  where $\mathbb{I}(\cdot)$ is the indicator function (1/0 when argument is T/F).
- e.g. for $\mathbf{x} = (1, 0, 4.3, -.5, .8, 2.3, -.2, 1.1, 1.4, 2)$



Empirical CDF sampling: non-parametric bootstrapping

- Can generate from the estimated F, by generating $u \sim U(0,1)$ and evaluating $\hat{F}^{-}(u)$ where $\hat{F}^{-}(u) = \min(x|\hat{F}(x) \geq u)$.

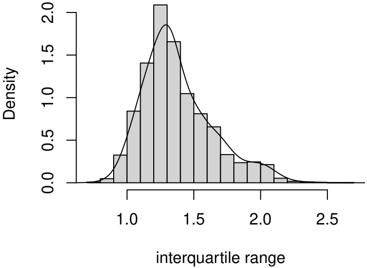


- …but CDF steps are evenly spaced $1/n$ apart on the vertical axis and the jumps occur at the original x_i values…
- We are just randomly selecting from the original sample of x_i values with equal probability. *Non-parametric bootstrapping*.

Non-parametric bootstrap example

Suppose we are interested in the interquartile range of the New Haven temperature distribution.

```
nb <- 10000; n <- length(nhtemp)
iqb <- rep(0,nb) ## storage for bs interquartile range
for (i in 1:nb) { ## bootstrap loop
  yb <- sample(nhtemp,n,replace=TRUE) ## non-parametric bs sample
  iqb[i] <- diff(quantile(yb,c(.25,.75))) ## bootstrap iqr
}
## examine distribution (notice the skew)...
hist(iqb,freq=FALSE);lines(density(iqb,adjust=2))
```



Basic bootstrap confidence interval

- IQR distribution not at all symmetric. Percentile CI dubious.
- Really BS gives distribution of $\hat{q}^*$ when true IQR is $\hat{q}$.
- Find interval containing 95% of $\hat{q}^*$. Write as $(\hat{q} - b, \hat{q} + c)$.

$$\begin{aligned} 0.95 = \Pr(\hat{q} - b < \hat{q}^* < \hat{q} + c) &= \Pr(-\hat{q}^* - b < -\hat{q} < -\hat{q}^* + c) \\ &= \Pr(\hat{q}^* + b > \hat{q} > \hat{q}^* - c) \end{aligned}$$

- View b and c as bootstrap estimates of constants such that

$$\Pr(\hat{q} + b > q > \hat{q} - c) = 0.95.$$

- That is $(\hat{q} - c, \hat{q} + b)$ is a 95% CI for q.
- Coincides with percentile interval for symmetric distribution.
- Obviously equally valid for parametric bootstrap.

Basic bootstrap CI R code

- Continuing the interquartile range example.

```
> iq <- diff(quantile(nhtemp,c(.25,.75))) ## original iqr
> ## 0.025 and 0.975 quantiles of b.s. irq...
> pci <- quantile(iqb,c(.025,.975))
> names(pci) <- names(iq) <- NULL ## avoid confusing names
> ## get upper and lower interval margins
> b <- iq - pci[1]; c <- pci[2] - iq
> c(iq-c,iq+b)  ## basic CI
0.625 1.650
> pci  ## equivalent percentile CI
1.000 2.025
```

- Notice the quite large difference between basic and percentile intervals in this case.

Bootstrapping multivariate data

- The bootstrapping idea generalizes readily to multivariate data.
- We simply resample *cases* - for tidy data this usually amounts to resampling rows. How?
- We can programme bootstrap resampling of univariate data by resampling either the data themselves, or their indices. That is

  ```
  xb <- sample(x,n,replace=TRUE)
  ```

  is equivalent to

  ```
  bsi <- sample(1:n,n,replace=TRUE)
  xb <- x[bsi]
  ```

- The second option generalizes immediately to resampling rows of a data matrix

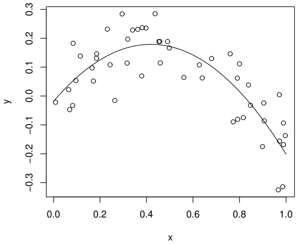
  ```
  bsi <- sample(1:n,n,replace=TRUE)
  Xb <- X[bsi,]
  ```

A more interesting example

- Consider a chemical experiment where you have measured yield, y, against flow rate of a reagent, x, and interest is in the x value maximizing yield.
- Suppose a quadratic model is appropriate

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i$$

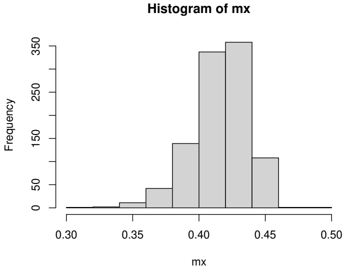
where the β_j are parameters and ϵ_i is a zero mean r.v.



A more interesting bootstrap

- We can bootstrap to find a CI for the optimum yield. Suppose x and y data are in dataframe dat.

```
n <- nrow(dat); nb <- 1000; mx <- rep(0,nb)
for (i in 1:nb) { ## bootstrap loop
  ii <- sample(1:n,n,replace=TRUE) ## resample row indices
  b <- coef(lm(y ~ x + I(x^2),data=dat[ii,])) ## fit to resample
  mx[i] <- -b[2]/(2*b[3]) ## compute location of maximum
}
quantile(mx,c(0.05,0.95)) ## 90% CI
  0.3785062 0.4458549
hist(mx)
```


Histogram of mx
Frequency
mx