Estimating Totals

Estimating totals

Many attributes are either a total

$$a(\mathcal{P}) = \sum_{u \in \mathcal{P}} y_u$$

for some variate y_u defined for any unit $u \in \mathcal{P}$, or a function of such a total. Recall that a variate y is any function that when applied to any unit $u \in \mathcal{P}$, returns a value $y_u = y(u)$.

- e.g. a population average $\overline{z}$, can be expressed using $y_u = z_u/N$.
- e.g. a number of units with a certain property can be expressed as sum of indicators, $y_u = I_{\mathcal{A}}(u)$
- Sometimse, if an attribute focuses on a subpopulation $\mathcal{A} \subset \mathcal{P}$, we may represent the variates as

$$y_u = z_u \times I_{\mathcal{A}}(u).$$

  - a sub-population could be defined in a variety of ways, including having it depend on the value of another variate x_u as in $z_u \times I_B(x_u)$.

Examples

- The number of shark encounters from the sub-population of Australia, $\mathcal{A} \subset \mathcal{S}$:

$$a(\mathcal{P}) = \sum_{u \in \mathcal{P}} I\left(u \in \mathcal{A}\right) = \sum_{u \in \mathcal{P}} I_{\mathcal{A}}(u)$$

- The proportion of units with their vairate y less than or equal to 2:

$$a(\mathcal{P}) = \frac{1}{N} \sum_{u \in \mathcal{P}} I(y_u \leq 2) = \frac{1}{N} \sum_{u \in \mathcal{P}} I_{(-\infty,\ 2]}(y_u)$$

- The **cumulative distribution function** (cdf) $F_{\mathcal{P}}(y)$ at a specific value y defined as

$$F_{\mathcal{P}}(y) = \frac{1}{N} \sum_{u \in \mathcal{P}} I(y_u \leq y) = \frac{1}{N} \sum_{u \in \mathcal{P}} I_{(-\infty,\ y]}(y_u),$$

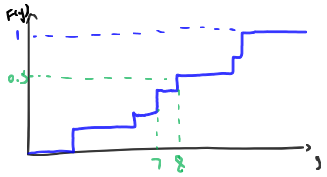
- We could define the quantile (and hence "inverse" cdf) as

$$Q_y(p) = \inf\left\{y_u \ : \ p \leq F_{\mathcal{P}}(y_u) \ \text{ and } \ u \in \mathcal{P}\right\}$$

  Since the cfd can be written as a total (sum), the quantiles can, implicitly, be written as totalls.

- In practice, instead of inf, we might choose to interpolate between two successive ordered values $y_{(i)} \leq y_{(i+1)}$ whenever $F_{\mathcal{P}}(y_{(i)}) \leq p \leq F_{\mathcal{P}}(y_{(i+1)})$.

Exercise: If $F_{\mathcal{P}}(y_{(i)}) \leq p \leq F_{\mathcal{P}}(y_{(i+1)})$ for $y_{(i)} < y_{(i+1)}$, give a mathematical expression for the value that would be returned by a simple linear interpolation.



An Estimate

To estimate $a(\mathcal{P}) = \sum_{u \in \mathcal{P}} y_u$, we use
$\hat{a}(\mathcal{P}) = a_{HT}(\mathcal{P}) = \sum_{u \in \mathcal{S}} \frac{y_u}{\pi_u}$

- A natural estimate of a population total $a(\mathcal{P}) = \sum_{u \in \mathcal{P}} y_u$, called the **Horvitz-Thompson estimate** after (Horvitz and Thompson 1952), is defined as

$$\widehat{a}(\mathcal{P}) = a_{HT}(\mathcal{S}) = \sum_{u \in \mathcal{S}} \frac{y_u}{\pi_u}$$

  where each value in the sample sum is weighted inversely to π_u, its probability of inclusion in $\mathcal{S}$.

- Note that the Horvitz-Thompson estimate is not necessarily Fisher consistent, so we use the subscript HT and the "hat" above $a(\mathcal{P})$ in the definition.

Exercise: Investigate the location-scale invariance and equivariance of Horvitz-Thompson estimator.

Exercise: Determine the sensitivity curve for the Horvitz-Thompson estimator (assume y has inclusion probability π). Draw the curve for various values of π. Comment on the results.

Sampling Properties - Expectation

- The repeated sampling properties of the Horvitz-Thompson estimat**or** can be derived.
  - It will be convenient to work with the random variate

$$D(u) = \begin{cases} 1 & \text{if } u \in \mathcal{S} \\ 0 & \text{otherwise.} \end{cases}$$

$$a_{HT}(\mathcal{S}) = \sum_{u \in \mathcal{S}} \frac{y_u}{\pi_u} = \sum_{u \in \mathcal{P}} D(u) \times \frac{y_u}{\pi_u}$$

The randomness comes from whether it's included in the sample

$a_{HT}(S) = \sum_{u \in S} \frac{y_u}{\pi_u} = \sum_{u \in \mathcal{P}} D(u) \frac{y_u}{\pi_u}$ where $D(u) = \begin{cases} 1 & \text{if } u \in S \\ 0 & \text{otherwise} \end{cases}$

prob of inclusion

$D(u) \sim \text{Bernu}(\pi_u)$

$E[D(u)] = 0 \times P(D(u)=0) + 1 \times P(D(u)=1) = \pi_u$

$\text{Var}(D_u) = E[D(u)^2] - (E[D_u])^2 = \pi_u(1-\pi_u)$

$E[a_{HT}(S)] = E\left[\sum_{u \in \mathcal{P}} D(u) \frac{y_u}{\pi_u}\right]$

$= \sum_{u \in \mathcal{P}} E[D(u)] \cdot \frac{y_u}{\pi_u}$

$= \sum_{u \in \mathcal{P}} y_u \Rightarrow E[a_{HT}(S)] = \sum_{u \in \mathcal{P}} y_u$

$\Rightarrow$ So $a_{HT}(S)$ is unbiased for $\sum_{u \in \mathcal{P}} y_u$

Therefore, MSE of HT-estimator is the variance

Let us investigate the **unbiasedness** of the HT estimator:

Sampling Properties - Variance

- The *variance of the Horvitz-Thompson estimator*,

$$Var\left(\widetilde{a}_{HT}(\mathcal{S})\right) = \sum_{u \in \mathcal{P}} \sum_{v \in \mathcal{P}} (\pi_{uv} - \pi_u \pi_v) \frac{y_u}{\pi_u} \frac{y_v}{\pi_v} = \sum_{u \in \mathcal{P}} \sum_{v \in \mathcal{P}} \Delta_{uv} \frac{y_u}{\pi_u} \frac{y_v}{\pi_v}$$

  - where $\Delta_{uv} = Cov(D(u), D(v)) = \pi_{uv} - \pi_u \pi_v$ is the **covariance** term.

$\text{Var}(a_{HT}(S)) = E\left[a_{HT}(S)^2\right] - \left(E[a_{HT}(S)]\right)^2$

$= E\left[\sum_{u \in S} \sum_{v \in S} \frac{y_u}{\pi_u} \frac{y_v}{\pi_v}\right] - \sum_{u \in \mathcal{P}} y_u \sum_{v \in \mathcal{P}} y_v$

$= E\left[\sum_{u \in \mathcal{P}} \sum_{v \in \mathcal{P}} D(u) D(v) \frac{y_u}{\pi_u} \frac{y_v}{\pi_v}\right] - \sum_{u \in \mathcal{P}} \sum_{v \in \mathcal{P}} y_u y_v$

$= \sum_{u \in \mathcal{P}} \sum_{v \in \mathcal{P}} \underbrace{E[D(u)D(v)]}_{1 \times P(D(u)=1, D(v)=1)} \frac{y_u}{\pi_u} \frac{y_v}{\pi_v} - \sum_{u \in \mathcal{P}} \sum_{v \in \mathcal{P}} \pi_u \pi_v \frac{y_u}{\pi_u} \frac{y_v}{\pi_v}$

π_{uv}: prob that u & v both in the sample

$\Rightarrow$ joint inclusion prob

$= \sum_{u \in \mathcal{P}} \sum_{v \in \mathcal{P}} \pi_{uv} \frac{y_u}{\pi_u} \frac{y_v}{\pi_v} - \sum_{u \in \mathcal{P}} \sum_{v \in \mathcal{P}} \pi_u \pi_v \frac{y_u}{\pi_u} \frac{y_v}{\pi_v}$

$= \sum_{u \in \mathcal{P}} \sum_{v \in \mathcal{P}} \underbrace{(\pi_{uv} - \pi_u \cdot \pi_v)}_{\text{cov}(D(u), D(v)) = \Delta_{uv}} \frac{y_u}{\pi_u} \frac{y_v}{\pi_v}$

QED

Variance Variations

- The variance of the Horvitz-Thompson estimator can equivalently written as

$$Var\left(\widetilde{a}_{HT}(\mathcal{S})\right) = \sum_{u \in \mathcal{P}} (1 - \pi_u) \frac{y_u^2}{\pi_u} + \sum_{u \in \mathcal{P}} \sum_{\substack{v \in \mathcal{P} \\ v \neq u}} \Delta_{uv} \frac{y_u}{\pi_u} \frac{y_v}{\pi_v}$$

- and equivalently the Yates-Grundy formulation, or the Sen-Yates-Grundy formula is

$$Var\left(\widetilde{a}_{HT}(\mathcal{S})\right) = -\frac{1}{2}\sum_{u\in\mathcal{P}}\sum_{v\in\mathcal{P}}\Delta_{uv}\left(\frac{y_u}{\pi_u} - \frac{y_v}{\pi_v}\right)^2$$

- Yates-Grundy formulation is insightful:
- If

$$\pi_u \propto y_u$$

  then $Var\left(\widetilde{a}_{HT}(\mathcal{S})\right) = 0$
- If y_u and y_ν are "close", choose π_u and π_ν "close".

SRS: sample with replacement
$\pi_u = \frac{\binom{N-1}{n-1}}{\binom{N}{n}} = \frac{n}{N}$

An Example

Consider the simple random sampling without replacement.

a. Show that the Horvitz-Thompson estimator of the population total is

$$a_{HT}(\mathcal{S}) = \frac{N}{n}\sum_{u\in\mathcal{S}} y_u$$

$a(\mathcal{P}) = \sum_{u\in\mathcal{P}} y_u$: total $\longrightarrow$ $a(\mathcal{P}) = \sum_{u\in\mathcal{P}} \frac{y_u}{N}$: average of y = $\frac{1}{N}a(\mathcal{P})$
$Var(a(\mathcal{P})) = N^2\left(1-\frac{n}{N}\right)\frac{1}{n}S^2$
where $S^2 = \frac{1}{N-1}\left(\sum_{u\in\mathcal{P}}(y_u-\bar{y})^2\right)$
$Var(a^*_{HT}(\mathcal{P})) = Var\left(\frac{1}{n}a_{HT}(\mathcal{P})\right) = \frac{1}{n^2}Var(a_{HT}(\mathcal{P})) = \left(1-\frac{n}{N}\right)\frac{S^2}{n}$
↑ variance of the mean
↑ finite population correction

b. Show that the variance of the Horvitz-Thompson estimator in part (a) is

$$Var(\widetilde{a}_{HT}(\mathcal{S})) = N^2\left(1-\frac{n}{N}\right)\frac{1}{n}\left(\frac{\sum_{u\in\mathcal{P}}(y_u-\overline{y})^2}{N-1}\right)$$

Sampling with replacement: $\pi_u = \frac{n}{N}$
$\pi_{uv} = \frac{n(n-1)}{N(N-1)}$

c. Note that dividing this by N^2 gives the variance of the sample average estimator $\sum_{u\in\mathcal{S}} y_u/n$ for the population average $\sum_{u\in\mathcal{P}} y_u/N$. The variance formula should look somewhat familiar except for a *finite population correction* $\left(1-\frac{n}{N}\right)$. Seeing this, explain the formula (divided by N^2) in words. What if $N >> n$?

Variance as a Total

- We consider the population $\mathcal{P}_{uv}$ of size N^2 consisting of all pairs (u,v) where $u,v\in\mathcal{P}$,
  - then the variance of the Horvitz-Thompson estimator can be written as

$$Var(\widetilde{a}_{HT}(\mathcal{S})) = \sum_{u\in\mathcal{P}}\sum_{v\in\mathcal{P}}\Delta_{uv}\frac{y_u}{\pi_u}\frac{y_v}{\pi_v} = \sum_{(u,v)\in\mathcal{P}_{uv}} q_{u,v}$$

    where

$$q_{u,v} = \Delta_{uv}\frac{y_u}{\pi_u}\frac{y_v}{\pi_v}.$$

- The $Var(\widetilde{a}_{HT}(\mathcal{S}))$ is itself a total!
  - The population is all pairs (u,v).
  - That means that we can use a Horvitz-Thompson estimator of this total to get an unbiased estimator!

Variance of HT estimator is in the form of a sum, so it can be estimated by using a HT estimator: $\sum_{u \in \mathcal{P}} y_u \quad \sum_{u \in \mathcal{S}} \frac{y_u}{\pi_u}$

parameter ← estimator
= attribute
r.v.
↳ s.d. of estimator
↳ ↑ s.d. estimate of s.d. = std error

standard error (std error) is the estimate of the standard deviation (s.d.) of the corresponding estimator

Estimating the Variance

- Sample from the population of pairs, $S_{uv} = \mathcal{S} \times \mathcal{S}$ by sampling $\mathcal{S}$ from $\mathcal{P}$ via $p(\mathcal{S})$.
- Then inclusion probability for each pair (u, v) is simply $\pi_{uv} > 0$ and
  - our variate is $q_{u,v} = \Delta_{uv} \frac{y_u}{\pi_u} \frac{y_v}{\pi_v}$

$$\widehat{Var}(\widetilde{a}_{HT}(\mathcal{S})) = \sum_{(u,v) \in \mathcal{S}_{uv}} \frac{q_{u,v}}{\pi_{uv}} = \sum_{u \in \mathcal{S}} \sum_{v \in \mathcal{S}} \frac{\Delta_{uv}}{\pi_{uv}} \frac{y_u}{\pi_u} \frac{y_v}{\pi_v} = \sum_{u \in \mathcal{S}} \sum_{v \in \mathcal{S}} \left(\frac{\pi_{uv} - \pi_u \pi_v}{\pi_{uv}} \right) \frac{y_u}{\pi_u} \frac{y_v}{\pi_v}.$$

- Using Horvitz-Thompson estimation, we are able to construct an
  - estimate of the population total *and* an
  - estimate of the variance of these estimators.
  - Both estimators are unbiased.

Summary

- From some population $\mathcal{P}$, we want to estimate some population total $\sum_{u \in \mathcal{P}} y_u$
  - We obtined a sample $\mathcal{S}$ from $\mathcal{P}$ using a sampling design $p(\mathcal{S})$.
  - from the sampling design we can determine the inclusion proabilities π_i and π_{ij}.
- To estimate the population total we use the Horvitz-Thompson estimate

$$a_{HT}(\mathcal{S}) = \sum_{u \in \mathcal{S}} \frac{y_u}{\pi_u}$$

- if we repeatedly sample from $\mathcal{P}$ using the sampling design $p(\mathcal{S})$, the variance of the estimator is

$$Var(\widetilde{a}_{HT}(\mathcal{S})) = \sum_{u \in \mathcal{P}} \sum_{v \in \mathcal{P}} (\pi_{uv} - \pi_u \pi_v) \frac{y_u}{\pi_u} \frac{y_v}{\pi_v}$$

  - based our sample an estimate of the variance is

$$\widehat{Var}(\widetilde{a}_{HT}(\mathcal{S})) = \sum_{u \in \mathcal{S}} \sum_{v \in \mathcal{S}} \left(\frac{\pi_{uv} - \pi_u \pi_v}{\pi_{uv}} \right) \frac{y_u}{\pi_u} \frac{y_v}{\pi_v}$$

- We are commonly interested in an estimate of the standard deviation which is

$$sterr(\widehat{a}_{HT}(\mathcal{S}) = \widehat{SD}(\widetilde{a}_{HT}(\mathcal{S})) = \sqrt{\widehat{Var}(\widetilde{a}_{HT}(\mathcal{S}))}$$

  - this is the estimate of the standard deviation of the corresponding estimator but some prefer calling it the **standard error** of an estimate.
- Many ettributes can be written as totals, e.g. mean, proportion, variance of a HT estimator, etc.
- **Refer to the posted course notes for more details and R codes.**

$\pi_u = P(\text{unit } u \text{ in sample}) = P(\text{unit } u \text{ in the sample \& sample includes unit } u)$
$= \sum_{\text{all samples } s} P(\text{unit } u \text{ in sample} \mid \text{sample } s \text{ chosen}) P(\text{sample } s \text{ chosen})$

$P(A) = P(A \cap B)$ where $A \subseteq B$

Example

Recall, our example with the population consists of five units. Interest lies in estimating the mean and providing the variance of the estimator.

```
set.seed(341)
x = round(rnorm(5),2)
x = sort(x)
x
```

```
## [1] -1.06 -0.99 -0.31  0.83  0.87
```

We can generate all samples of size 2

```
sam2 = combn(5,2)
sam2
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    1    1    1    2    2    2    3    3     4
## [2,]    2    3    4    5    3    4    5    4    5     5
```

Then calculate the average from each sample.

```
a2 <- apply(sam2, 2, function(s){mean(x[s])})
```

Two sampling design

```
p1 = rep(1/10,10)
p2 = 2*(abs(apply(sam2, 2, diff))-1)
p2 = p2/sum(p2)
round( rbind(p1,p2),2)
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## p1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1  0.1   0.1
## p2  0.0  0.1  0.2  0.3  0.0  0.1  0.2  0.0  0.1   0.0
```

We will focus on design p2 for the rest of the example.

Inclusion Probability ($p2$ design)

Compare the samples with $p(\mathcal{S})$

```
rbind( p2, combn(5, 2) )
```

```
##    [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## p2    0  0.1  0.2  0.3    0  0.1  0.2    0  0.1     0
##       1  1.0  1.0  1.0    2  2.0  2.0    3  3.0     4
##       2  3.0  4.0  5.0    3  4.0  5.0    4  5.0     5
```

For every sample, find which units are in the sample

```
inSample <- function(sam=NULL, N=NULL) {
  inSam = numeric(N) #inSam is a vecor of size N
  inSam[sam] = 1
  inSam
}

combn(5, 2, inSample, N=5)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    1    1    1    0    0    0    0    0     0
## [2,]    1    0    0    0    1    1    1    0    0     0
## [3,]    0    1    0    0    1    0    0    1    1     0
## [4,]    0    0    1    0    0    1    0    1    0     1
## [5,]    0    0    0    1    0    0    1    0    1     1
```

Weight each sample by the probability of selecting that sample.

```
weighted.sum <- function(x=NULL, w=NULL) { sum(x*w) }

pi2 = apply( combn(5, 2, inSample, N=5), 1, weighted.sum, w=p2)
pi2
```

```
## [1] 0.6 0.3 0.2 0.3 0.6
```

- Now, the sampling bias calculated using the sampling design and the inclusion probabilities.
- We do this based on distribution of the attribute $P(A = a)$ and based on the the inclusion probabilities (show on the board).

```
c( mean(a2)-sum(a2*p2) , mean(a2) - sum(x*pi2)/2  )
```

```
## [1] -0.02 -0.02
```

Joint Inclusion Probability

- To calculate the variance, we need joint inclusion probabilities $\pi_{u\nu}$
- For every sample, find which units are both in it.

```
inSample2 <- function(sam=NULL,  N=NULL) {
  inSam = numeric(N)
  inSam[sam] = 1
  inSam = outer(inSam, inSam)
  inSam
}

combn(5, 2, inSample, N=5) #this will generate a 5x5x10 array
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    1    1    1    0    0    0    0    0     0
## [2,]    1    0    0    0    1    1    1    0    0     0
## [3,]    0    1    0    0    1    0    0    1    1     0
## [4,]    0    0    1    0    0    1    0    1    0     1
## [5,]    0    0    0    1    0    0    1    0    1     1
```

```
combn(5,2, FUN=inSample2, N=5)[,,1:2]
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    1    0    0    0
## [2,]    1    1    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
##
## , , 2
```

```
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    0    1    0    0
## [2,]    0    0    0    0    0
## [3,]    1    0    1    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
```

Weight each matrix by the probability of selecting that sample to get the joint inclusion probabilities.

```
pij2 = apply( combn(5,2, FUN=inSample2, N=5), c(1,2), weighted.sum, w=p2)
pij2
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]  0.6  0.0  0.1  0.2  0.3
## [2,]  0.0  0.3  0.0  0.1  0.2
## [3,]  0.1  0.0  0.2  0.0  0.1
## [4,]  0.2  0.1  0.0  0.3  0.0
## [5,]  0.3  0.2  0.1  0.0  0.6
```

- Now, the sampling variance calculated using the sampling design and the inclusion probabitlies. The variance using the sampling design

```
  sum( ( a2 -  sum(a2*p2) )^2*p2 )
```

```
## [1] 0.048931
```

- The variance using the inclusion and joint probabilities

```
delta = ( pij2 - outer(pi2, pi2)  )
c( (x %*% delta %*% x)*(1/2)^2,  sum( delta * outer(x, x) )*(1/2)^2   )
```

```
## [1] 0.048931 0.048931
```

Using the Horvitz-Thompson Estimate

- We had $n = 2$ and gave varying probabilities to our samples. The estimate was

$$estimated\ average = \sum_{u \in \mathcal{S}} y_u/2$$

  - the corresponding estimator was biased.

- We can use the inclusion probabilities to make our estimator unbiased.

$$estimated\ average = \frac{1}{5} \sum_{u \in \mathcal{S}} y_u/\pi_u$$

- Now, the sampling bias calculated using the sampling design and the inclusion probabitlies

```
a2 = apply(sam2, 2, function(s, x){ mean(x[s]) }, x=x)
a2HT = apply(sam2, 2, function(s, x, wt){ sum(x[s]/wt[s]) }, x=x,wt=pi2 )/5
```

- The sampling bias is

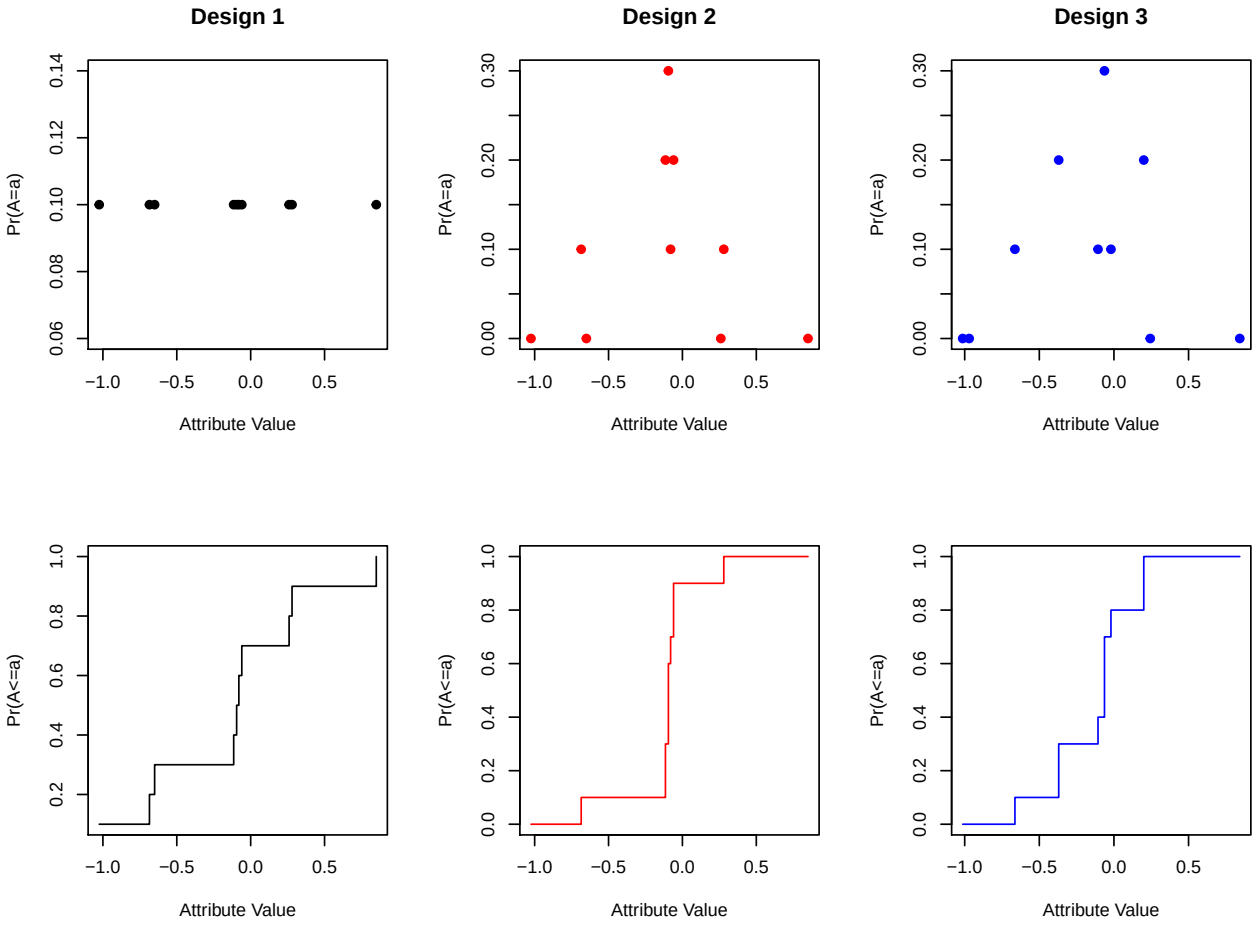
```
mean(x) - c(sum(a2*p2), sum(a2HT*p2) )
```

```
## [1] -0.02  0.00
```

Comparision

- Design 1
  - Simple random sampling with replacement
  - Estimate: $\frac{1}{5}\sum_{u\in\mathcal{S}} y_u/\pi_u = \sum_{u\in\mathcal{S}} y_u/n$
- Design 2
  - Sampling design with prob $= p2 = (0.0, 0.1, 0.2, 0.3, 0.0, 0.1, 0.2, 0.0, 0.1, 0.0)$
  - Estimate: $\sum_{u\in\mathcal{S}} y_u/n$ regular mean biased
- Design 3
  - Sample design with prob $= p2 = (0.0, 0.1, 0.2, 0.3, 0.0, 0.1, 0.2, 0.0, 0.1, 0.0)$
  - Estimate: $\frac{1}{5}\sum_{u\in\mathcal{S}} y_u/\pi_u$ HT est

```
##  [1] 0.0 0.1 0.2 0.3 0.0 0.1 0.2 0.0 0.1 0.0
```



- Reading the $PR(A = a)$ and CDF plots, what do you learn about the 3 designs?

Sampling MSE

```
##          Design 1 Design 2 Design 3
## bias       0.0000  -0.0200   0.0000
## samp.var   0.2669   0.0489   0.0643
## MSE        0.2669   0.0493   0.0643
```

biased but smaller MSE

Example - Shark Data

- We will estimate a few attributes and their variances based on the Sakrs data.

$$\hat{a}_{HT}(S) = \sum_{i \in S} y_i/\pi_i$$

$$\widehat{Var}\left[\hat{a}_{HT}(S)\right] = \sum_{u \in S} \sum_{v \in S} \left(\frac{\pi_{uv} - \pi_u \pi_v}{\pi_{uv}}\right) \frac{y_u}{\pi_u} \frac{y_v}{\pi_v}$$

- A random sample without replacement

```
N = nrow(sharks) # N =65
n = 10
```

- The inclusion probabilities are n/N

```
pi = rep(n/N, N)
sum(pi)
```

```
## [1] 10
```

- The joint inclusion probabilities are $\pi_{ij} = n/N \times (n-1)/(N-1)$ and the diagonal is $\pi_i = n/N$

```
pij = matrix(n/N * (n-1)/(N-1), nrow=N, ncol=N)
diag(pij) = pi
c( sum(pij[1,])*n, n*pi[1])
```

```
## [1] 15.384615  1.538462
```

```
#round(pij,4)
```

Shark Lengths HT estimate and population value

- We obtain a sample by selecting without replacement

```
set.seed(341)
sharkS <- sample(N, n)
```

- The HT estimate for the total is

```
c(sum(sharks$Length[sharkS]/pi[sharkS]),  sum(sharks$Length) )
```

```
## [1] 8508.5 9871.0
```

- The HT estimate for the average shark length and the population average is

```
c(sum(sharks$Length[sharkS]/pi[sharkS])/N, sum(sharks$Length/N) )
```

```
## [1] 130.9000 151.8615
```

- The HT estimate for the average age and the population average is

```
c(sum(sharks$Age[sharkS]/pi[sharkS])/N, sum(sharks$Age/N) )
```

```
## [1] 37.2 35.6
```

Average Shark Lengths HT estimate

- The HT estimate for the population average shark length is

$$\widetilde{a}_{HT}(\mathcal{S}) = \sum_{u \in \mathcal{S}} y_u/\pi_u = \sum_{u \in \mathcal{S}} \frac{z_u}{N} \frac{1}{\pi_u} = \frac{1}{N} \sum_{u \in \mathcal{S}} z_u/\pi_u$$

  where z_u is shark length,

```
yu = sharks$Length/N
pi = rep(n/N, N)

# Both computations below give the average shark length
c(sum(yu[sharkS]/pi[sharkS]), sum(sharks$Length[sharkS]/pi[sharkS])/N )
```

```
## [1] 130.9 130.9
```

- The variance of the HT estimator

$$Var\left(\widetilde{a}_{HT}(\mathcal{S})\right) = \sum_{u\in\mathcal{P}}\sum_{v\in\mathcal{P}}(\pi_{uv}-\pi_u\pi_v)\frac{y_u}{\pi_u}\frac{y_v}{\pi_v} = \sum_{u\in\mathcal{P}}\sum_{v\in\mathcal{P}}\Delta_{uv}\frac{y_u}{\pi_u}\frac{y_v}{\pi_v}$$

- The **estimate** of the variance of the HT estimator

$$\widehat{Var}(\widetilde{a}_{HT}(\mathcal{S})) = \sum_{u\in\mathcal{S}}\sum_{v\in\mathcal{S}}\left(\frac{\pi_{uv}-\pi_u\pi_v}{\pi_{uv}}\right)\frac{y_u}{\pi_u}\frac{y_v}{\pi_v}.$$

- The **standard error** of the HT estimator is the square root of the estimate of the variance.

```
HTestimate <- function(sam, yu, pi, pij ) {
  yu = yu[sam]
  pi = pi[sam]
  pij = pij[sam,sam]

  delta = pij - outer(pi, pi)


  estimate = sum(yu/pi)
  estimateVar =  sum( (delta/pij) * outer(yu/pi,yu/pi) )
  return(c(estimate, estimateVar))
}
HTestimate(sharkS, yu, pi, pij)
```

```
## [1] 130.9000 116.8905
```

The standard errror is

```
sqrt(HTestimate(sharkS, yu, pi, pij)[2])
```

```
## [1] 10.81159
```

Same Calculations Using the Rcode from Course notes

```
n = 10
N = nrow(sharks)
inclusionProb <- createInclusionProbFn(1:N, sampSize = n)
sharksHTestimator <- createHTestimator(inclusionProb)

### This is a new function modifed from createvariateFn
createvariateFnN <- function(popData, variate, N=1) {
  function (u){popData[u, variate]/N}
}

sharkAvgLength <- createvariateFnN(sharks, "Length", N=N)
sharksHTestimator(sharkS, sharkAvgLength)
```

```
## [1] 130.9
```

```
inclusionJointProb <- createJointInclusionProbFn(1:N,     sampSize = n)
HTVarianceEstimator <- createHTVarianceEstimator(1:N,
                         pi_u_fn = inclusionProb,
                        pi_uv_fn = inclusionJointProb)
HTVarianceEstimator(sharkS, sharkAvgLength)
```

```
## [1] 116.8905
```

From before

```
HTestimate(sharkS, yu, pi, pij)
```

```
## [1] 130.9000 116.8905
```

Shark Lengths HT proportions

- The HT estimate for the proportion of shark encounters from Australia (and the true proportion):

```
propAust <- createvariateFnN(sharks, "Australia", N=N)

c(sharksHTestimator(sharkS, propAust), sum(sharks$Australia)/N )
```

```
## [1] 0.3000000 0.4307692
```

- The HT estimate for the proportion of shark encounters that end in a fatality (and the true proportion):

```
propFat <- createvariateFnN(sharks, "Fatality", N=N)

c(sharksHTestimator(sharkS, propFat), sum(sharks$Fatality)/N )
```

```
## [1] 0.1000000 0.2615385
```

- The HT estimate for the proportion of shark encounters and person was Scuba diving (and the true proportion):

```
propScuba <- createvariateFnN(sharks, "Scuba", N=N)

c(sharksHTestimator(sharkS, propScuba), sum(sharks$Scuba)/N )
```

```
## [1] 0.00000000 0.09230769
```

- The HT estimate for the shark length less than or equal to 180 (and the true proportion):

```
createvariateFnNy <- function(popData, variate, N=1, y=NULL) {
  function (u) { (popData[u, variate] <= y)/N}
}

pSharky <- createvariateFnNy(sharks, "Length", N=N, y=180)

c(sharksHTestimator(sharkS, pSharky), sum(sharks$Length <= 180)/N )
```

```
## [1] 0.9000000 0.7692308
```

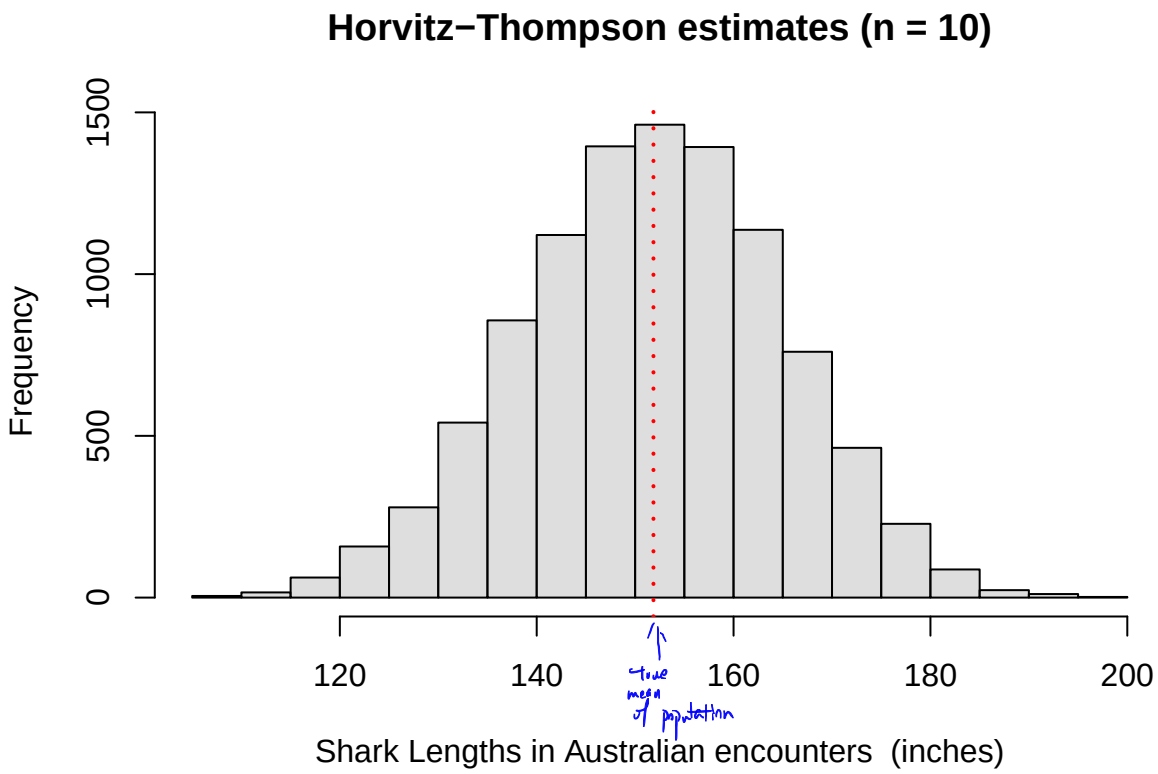
- The question is how far the estimates can be from the true values dues to sampling error?
- This is answered through the sampling distribution of the HT estimator

$\sqrt{\widehat{Var}(\bar{y}_{HT}(s))} = SE(\hat{\bar{y}})$

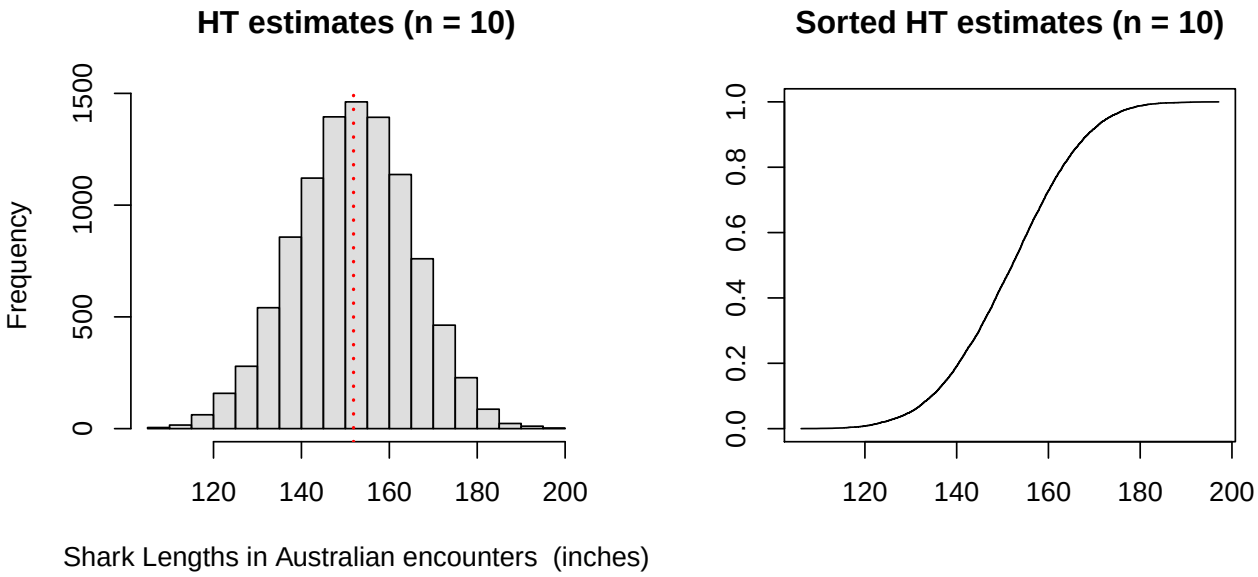
Sampling Distribution of the HT estimates (average)

- In this example, we look at the HT estimate for the average shark length.

```
### This is a new function modifed from createvariateFn
createvariateFnN <- function(popData, variate, N=1) {
  function (u){popData[u, variate]/N}
}

N= nrow(sharks)
n=10

sharkAvgLength <- createvariateFnN(sharks, "Length", N=N)
popAvg <- sum(sharks$Length)/N

set.seed(341)
avgs <- Map(function(rep) {
  sharksHTestimator(sample(N, n), sharkAvgLength)},  1:10000)

hist(as.numeric(avgs), col=adjustcolor("grey", alpha = 0.5),
     main="Horvitz-Thompson estimates (n = 10)",
     xlab="Shark Lengths in Australian encounters  (inches)",
     breaks=25
     )
### Mark the population attribute in red
abline(v=popAvg, col="red", lty=3, lwd=2)
```



Sampling Distribution of the HT estimates (average)



- If we use smapling without replacement and a sample size of size $n = 10$, how close is the sample estimate to the population value?
- To quantify this we might report
  - The intequartile range is 18.3
  - Our 25^{th} and 75^{th} quantiles are (142.6, 160.9)
  - The 25^{th} and 75^{th} quantiles as a range correspond to a range containing 50% of the averages obtained.

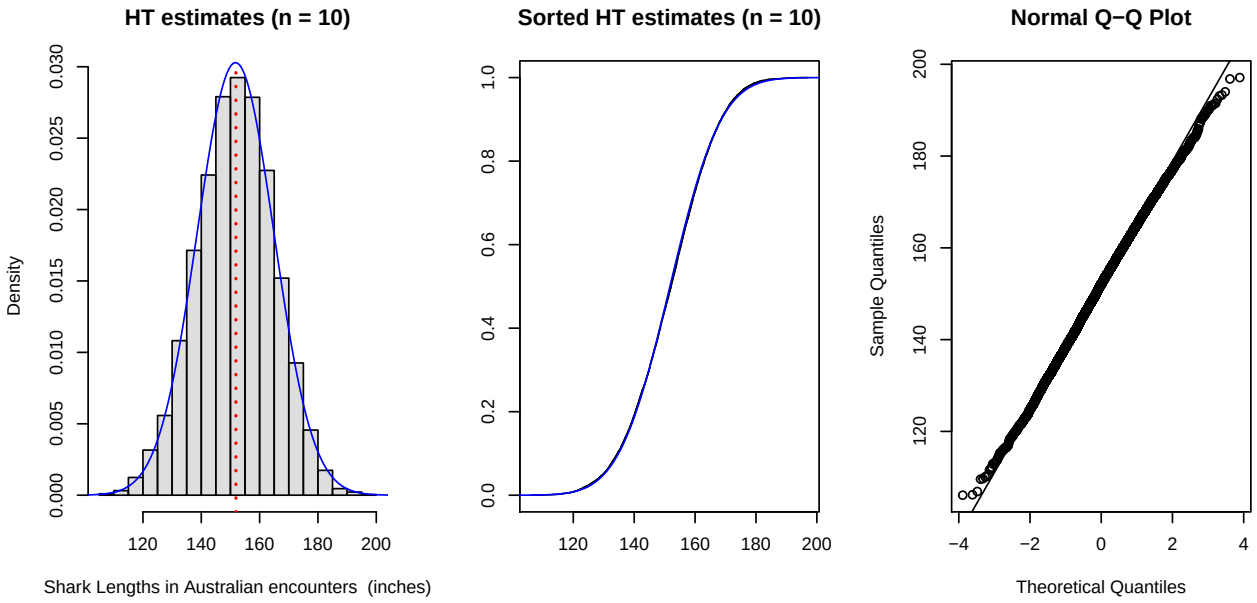
$$\Pr\left(\widehat{a}(\mathcal{S}) \in [Q(1/4), Q(3/4)]\right) = 0.5$$

- We might cast a wider net such as the trimmed range where we remove the smallest and largest 2.5% or effectively remove 5% of the sample
  - This trimmed range is $Q(3/4) - Q(1/4) = 51.3025$
  - The ranges of values (125.2, 176.5025) contain 95% percent of the HT estimates

$$\Pr\left(\widehat{a}(\mathcal{S}) \in [Q(0.025), Q(0.975)]\right) = 0.95$$

A Gaussian approximation

Using μ equal to the average 151.74145 and σ equal to the standard deviation 13.1722378 from the population of averages as parameters in the Gaussian desnity & cdf we can approximate the distribution of the HT estimator.



- We can approximate the range that contains 95% percent of the values with

$$\Pr\left(\widehat{a}(\mathcal{S}) \in [Q(0.025), Q(0.975)]\right) \approx \Pr\left(\frac{\widehat{a}(\mathcal{S}) - \mu}{\sigma} \in [Z_{0.025}, Z_{0.975}]\right) \approx 0.95$$

where Z is a standard normal random variable.

  - Using the Gaussian assumption a range that contains 95% percent of the values is

$$\mu \pm Z_{1-0.05/2}\sigma$$

  - which is (125.9, 177.6)

$$P\left(\frac{Q(0.025)-\mu}{\sigma} \leq Z \leq \frac{Q(0.975)-\mu}{\sigma}\right) = 95\%$$

$Z \sim N(\mu, \sigma)$

- Using the quantiles from the samples, a range that contains 95% percent of the values is (125.2, 176.5)

A Gaussian approximation

- Using the Gaussian assumption, a range that contains 95% percent of the values is

$$\mu \pm Z_{1-0.05/2}\sigma$$

  - but $Z_{1-0.05/2} = 1.959964 \approx 2$
  - and the HT estimator is unbiased,

$$\mu = \mathbb{E}\left[\widetilde{a}_{HT}(\mathcal{S})\right] = a_{HT}(\mathcal{P})$$

  - σ is an estimate of standard deviation of the HT estimator

$$\sigma = \mathrm{SD}\left[\widetilde{a}_{HT}(\mathcal{S})\right] = \sqrt{\mathbb{V}\mathrm{ar}\left[\widetilde{a}_{HT}(\mathcal{S})\right]}$$

- Putting this together we have,
  - Using the Gaussian assumption a range that contains 95% percent of the averages is

$$a_{HT}(\mathcal{P}) \pm 2\sqrt{\mathbb{V}\mathrm{ar}\left[\widetilde{a}_{HT}(\mathcal{S})\right]}$$

assume $1.96 \approx 2$

A Gaussian approximation

- Using the Gaussian assumption a range that contains 95% percent of the averages is

$$a_{HT}(\mathcal{P}) \pm 2\sqrt{\mathbb{V}\text{ar}\left[\widetilde{a}_{HT}(\mathcal{S})\right]}$$

From Stat 230, 231:
$\bar{x} \sim N(\mu, \frac{\sigma^2}{n})$
$P(\mu - 1.96\frac{\sigma}{\sqrt{n}} \le \bar{x} \le \mu + 1.96\frac{\sigma}{\sqrt{n}}) = 95\%$
$\Rightarrow P(\bar{x} - 1.96\frac{\sigma}{\sqrt{n}} \le \mu \le \bar{x} + 1.96\frac{\sigma}{\sqrt{n}}) = 95\%$

- But we typically only have a sample
  - an sample estimate of this is

$$\widehat{a}_{HT}(\mathcal{S}) \pm 2\sqrt{\widehat{\mathbb{V}\text{ar}}\left[\widetilde{a}_{HT}(\mathcal{S})\right]} = \widehat{a}_{HT}(\mathcal{S}) \pm 2\widehat{\text{SD}}\left[\widetilde{a}_{HT}(\mathcal{S})\right]$$

Sampling Distribution of the HT estimates (proportion)

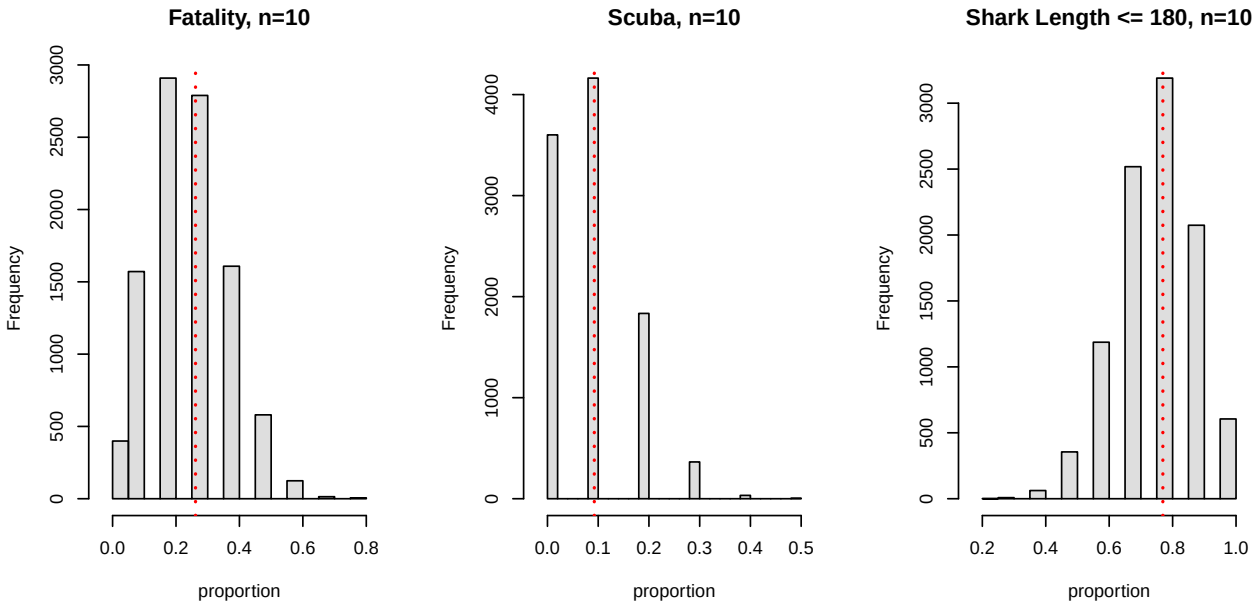
- The Gaussian approximation is not always good though.
  - Check the following cases

```
plotEstimators <- function( variate,  title="", xlab0="proportion", n=10, N=65, xlim0=NULL ) {
  set.seed(341)
  avgs <- Map(function(rep) {
    sharksHTestimator(sample(N, n), variate)},  1:10000)
  if (is.null(xlim0)) xlim0 = extendrange(avgs)

hist(as.numeric(avgs), col=adjustcolor("grey", alpha = 0.5),
     main=title,
     xlab=xlab0,
     breaks=25, xlim= xlim0 )
### Mark the population attribute in red
abline(v=sum(variate(1:N)), col="red", lty=3, lwd=2)
}
```

- For the proportion of shark encounters from Australia.

```
n=10

propFat <- createvariateFnN(sharks, "Fatality", N=65)
propScuba <- createvariateFnN(sharks, "Scuba", N=65)
pSharky <- createvariateFnNy(sharks, "Length", N=65, y=180)

par(mfrow=c(1,3),oma=c(0,0,0,0))
plotEstimators(propFat, "Fatality, n=10" )
plotEstimators(propScuba, "Scuba, n=10")
plotEstimators(pSharky, "Shark Length <= 180, n=10")
```

proportion in sharks data:
prop 1: Fatality prop of fatal encounters
prop 2: Scuba: the prop of encounters happened while scuba diving
prop 3: prop of encounters with shark length ≤180
↳ Pr(length ≤180)



- **What can we do here?**
  - clearly, Gaussian is NOT a good approximation
  - bootstrap methods?
  - there is a chance that bootstrap does not work either.
  - we will discuss resampling methods (e.g. bootstrap) later.

Sampling Design

- The pair $(\mathcal{P}_{\mathcal{S}}, p(\mathcal{S}))$ are called a **sampling design**.
  - The sampling design is ours to choose
  - the bias term for HT estimator is zero and the variance is

$$Var\left(a_{HT}(\mathcal{S})\right) = \sum_{u\in\mathcal{P}}\sum_{v\in\mathcal{P}} \Delta_{uv}\frac{y_u}{\pi_u}\frac{y_v}{\pi_v} = -\frac{1}{2}\sum_{u\in\mathcal{P}}\sum_{v\in\mathcal{P}} \Delta_{uv}\left(\frac{y_u}{\pi_u} - \frac{y_v}{\pi_v}\right)^2.$$

  - From this we can gain insight into how we might best choose a design.

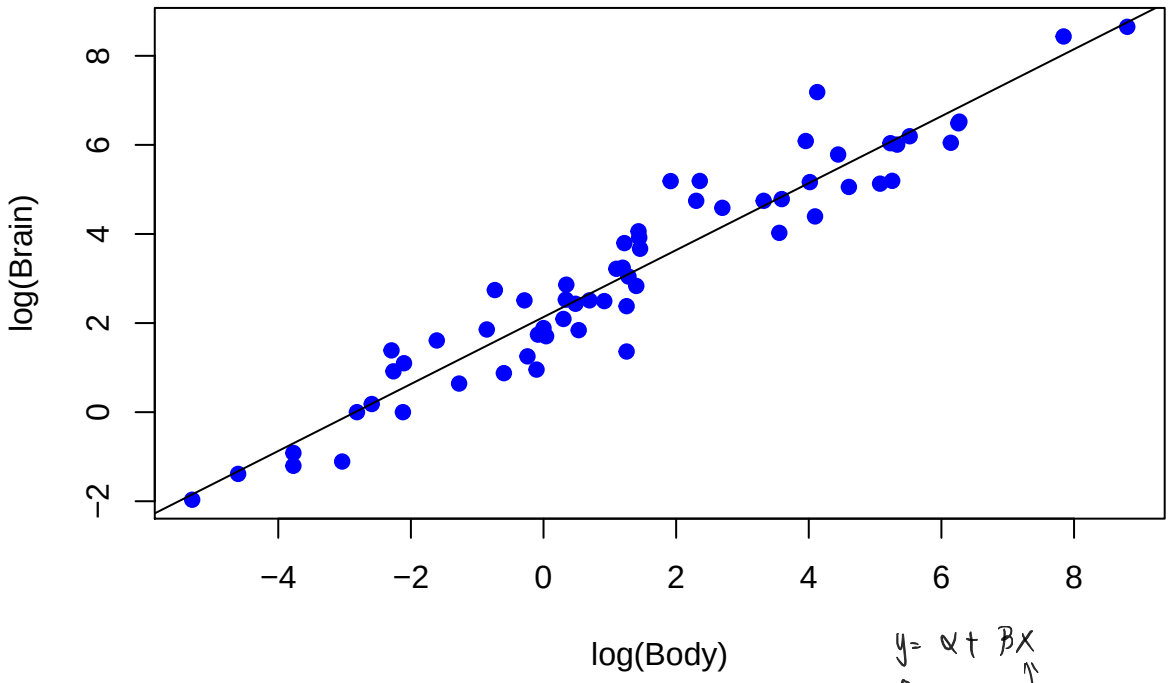
- For example, if we could choose $\pi_u \propto y_u$ then the variance will be zero!
  - Perhaps there is another variate x_u that is highly positively correlated with y_u for all $u \in \mathcal{P}$.
  - If we knew when $y_u \approx y_v$ we could arrange that $\pi_u \approx \pi_v$ and $\pi_{uv} = \pi_u\pi_v$ when y_u & y_v are different (e.g. stratified sampling tries to do this).

- Much of survey sampling is concerned with how best to choose the sampling design $(\mathcal{P}_{\mathcal{S}}, p(\mathcal{S}))$ to reduce the MSE.

Regression Example

- In this example, we show how sampling design can affect the precision of estimation.
- In particular, how it affects the estimation of the coefficients of a simple linear regression.
- Note that this example does not use HT estimator and only focuses on sampling design.

```
library(MASS)
data(mammals)
plot( log(mammals$body), log(mammals$brain), xlab="log(Body)", ylab="log(Brain)", pch=19, col=4)
abline( lm( log(brain) ~ log(body), mammals )$coef )
```



$y = \alpha + \beta x$

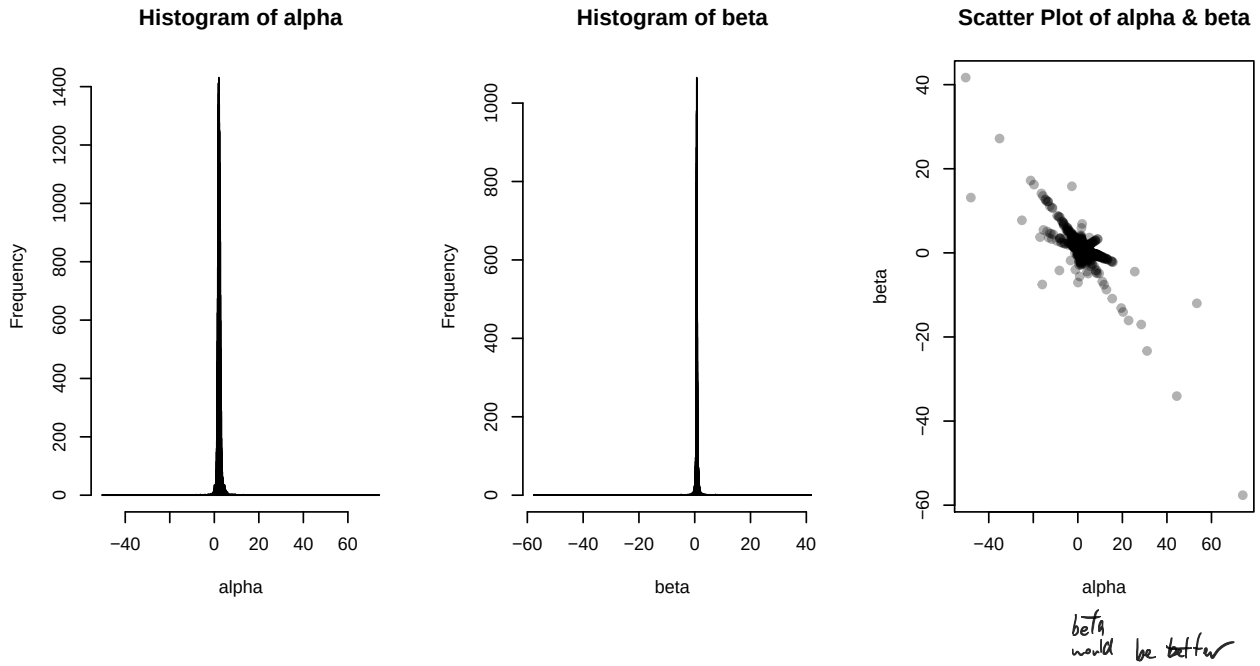
ln(Brain weight) ln(body weight)

All Samples of size 3

```
lmtest <- function(sam=NULL, data=NULL) {
  lm( log(brain) ~ log(body), data, subset=sam )$coef
}

choose(62,3)
```

```
## [1] 37820
```

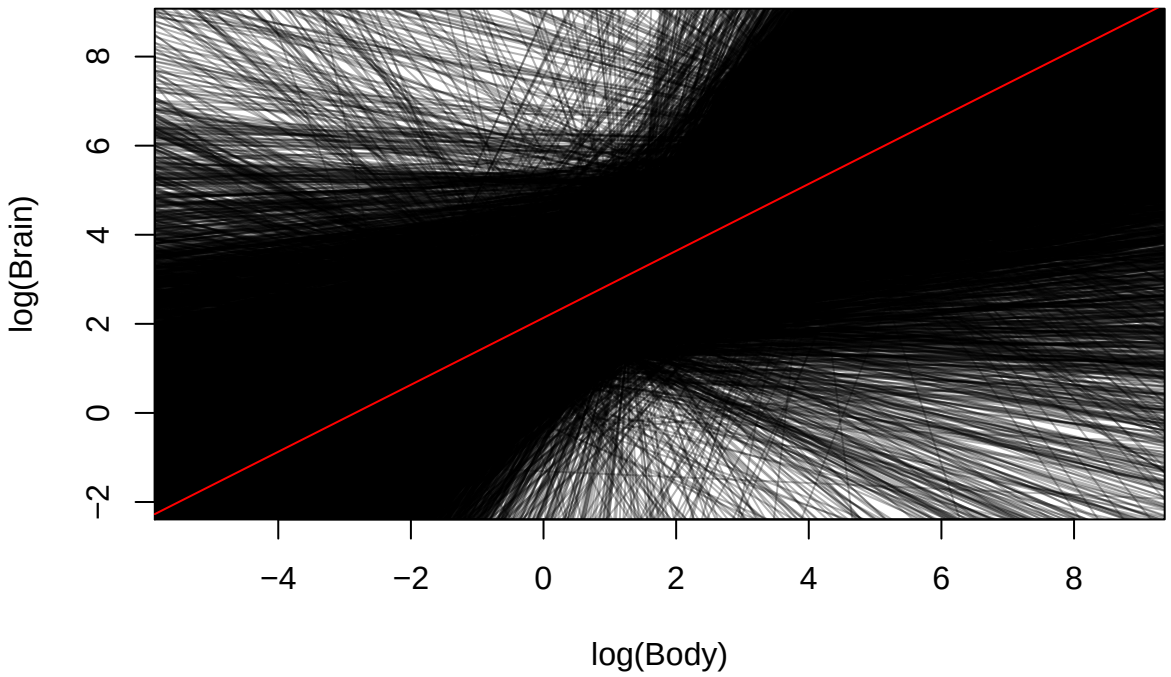
```
## This will take a while.
out = combn(62,3, FUN=lmtest, data=mammals)
```



All Samples of size 3

- The regression lines from all samples of size 3

```
plot( log(mammals$body), log(mammals$brain), xlab="log(Body)", ylab="log(Brain)")

for (i in 1:choose(62,3)) abline( out [,i], col=adjustcolor("black", alpha = 0.3)  )

abline( lm( log(brain) ~ log(body), mammals )$coef, col=2 )
points( log(mammals$body), log(mammals$brain), xlab="log(Body)", ylab="log(Brain)", pch=19, col=0)
```



Sample Size 6

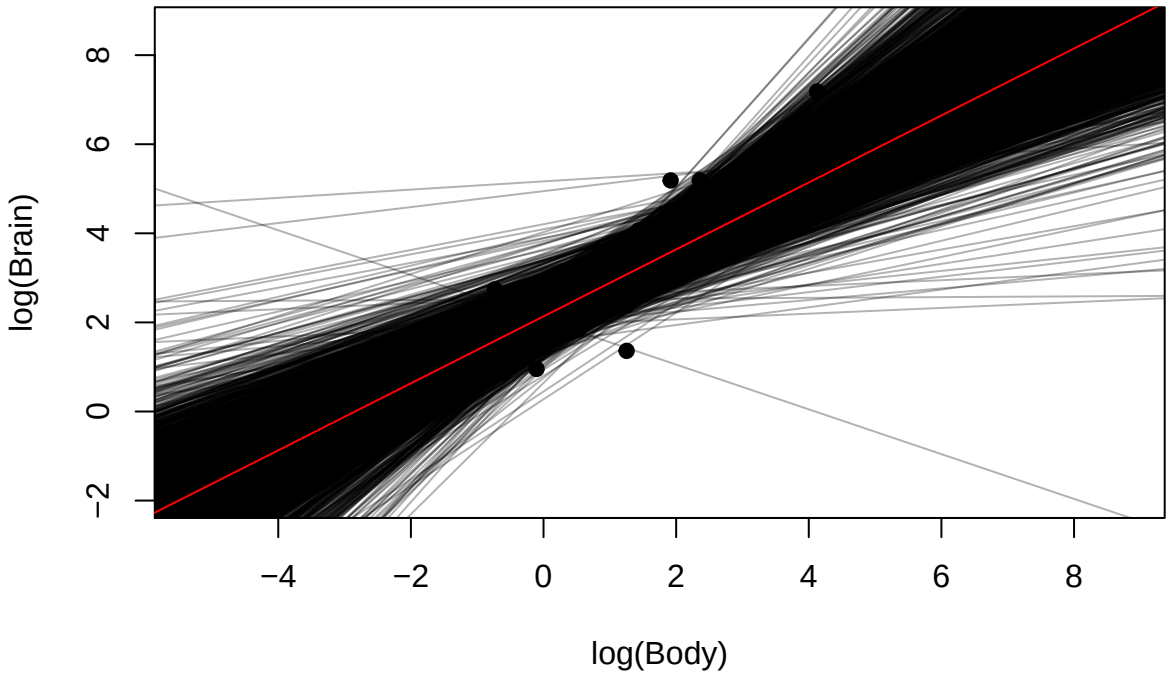
- For size 6, we have too many.

```
choose(63,6)
```

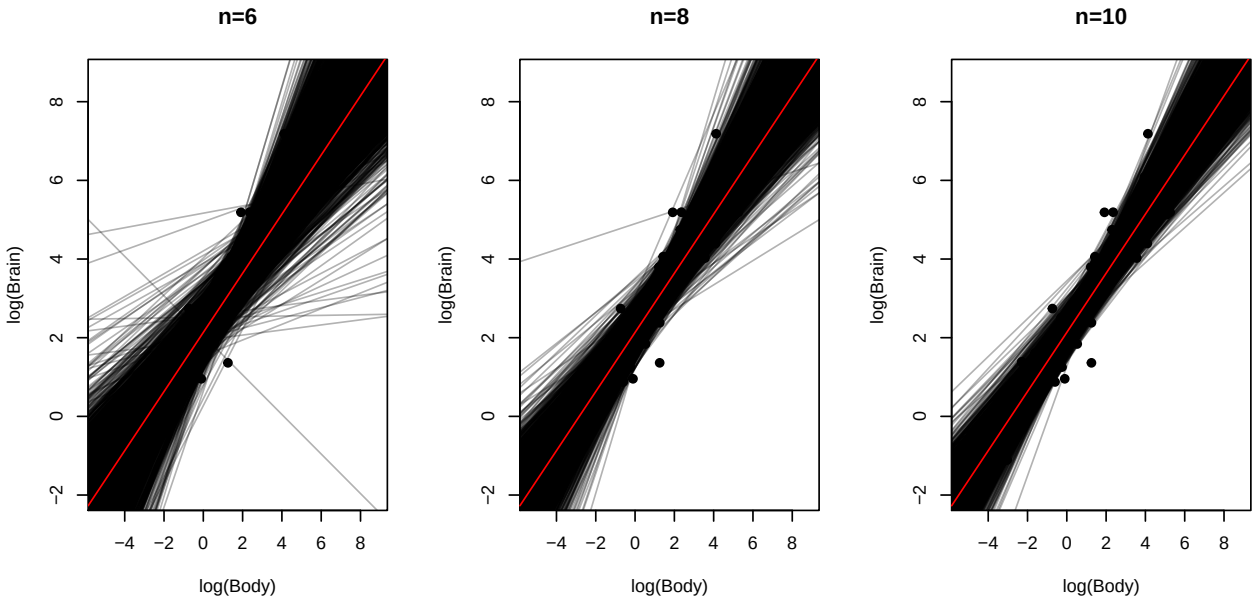
```
## [1] 67945521
```

- Here we use a random subset of the samples to quantify the sampling design.

```
N = nrow(mammals)
n = 6
m = 10000

set.seed(341)
reg.coef <- Map(function(rep) {
  lm( log(brain) ~ log(body), data=mammals, subset=sample(N, n) )$coef },  1:m)

plot( log(mammals$body), log(mammals$brain), xlab="log(Body)", ylab="log(Brain)", pch=19)

for (i in 1:m) abline( reg.coef[[i]], col=adjustcolor("black", alpha = 0.3) )

points( log(mammals$body), log(mammals$brain), pch=19, col=0)
abline( lm( log(brain) ~ log(body), mammals )$coef, col=2 )
```



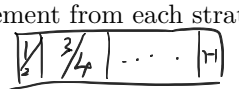
Sample Sizes 6, 8 and 10



- Can we construct a sampling design to reduce the variability?

Stratified Sampling

- The Population is split into H strata and we sample without replacement from each strata
  - Each strata has N_h units, $N_1 + \ldots + N_H = N$
  - We sample n_h from each strata, $n_1 + \ldots + n_H = n$



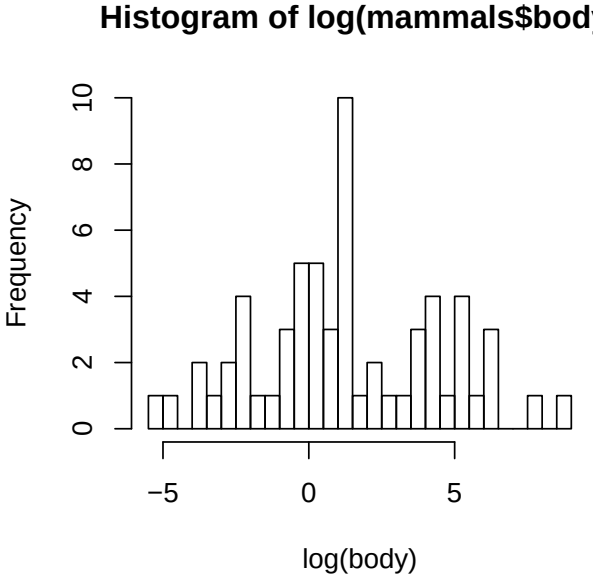
divide the population into H strata such that the differences within each stratum is negligible, but between strata the difference is significant

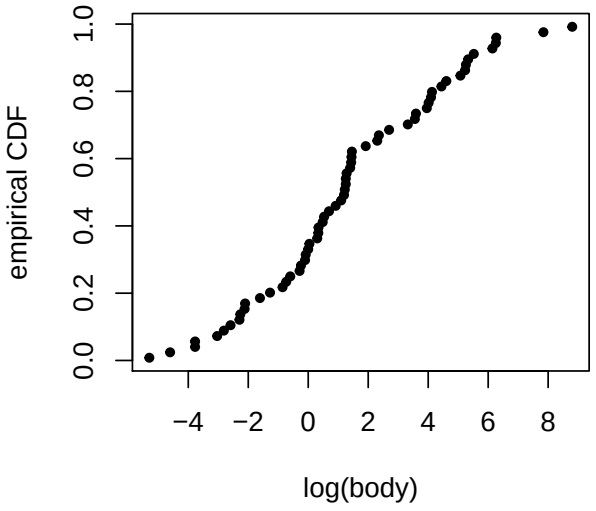
We take n_h sample from stratum h which has N_h units

the sample of size $n = n_1 + n_2 + \ldots + n_H$ from $N = N_1 + N_2 + \ldots + N_H$ is called a stratified sample

within each stratum a simple random sample without replacement is chosen.

- For our regression example, we will suppose that the body weight is the easier measurement to make and that the brain weights is expensive to obtain
  - Assumption: for each unit in the population, we have the body weight.



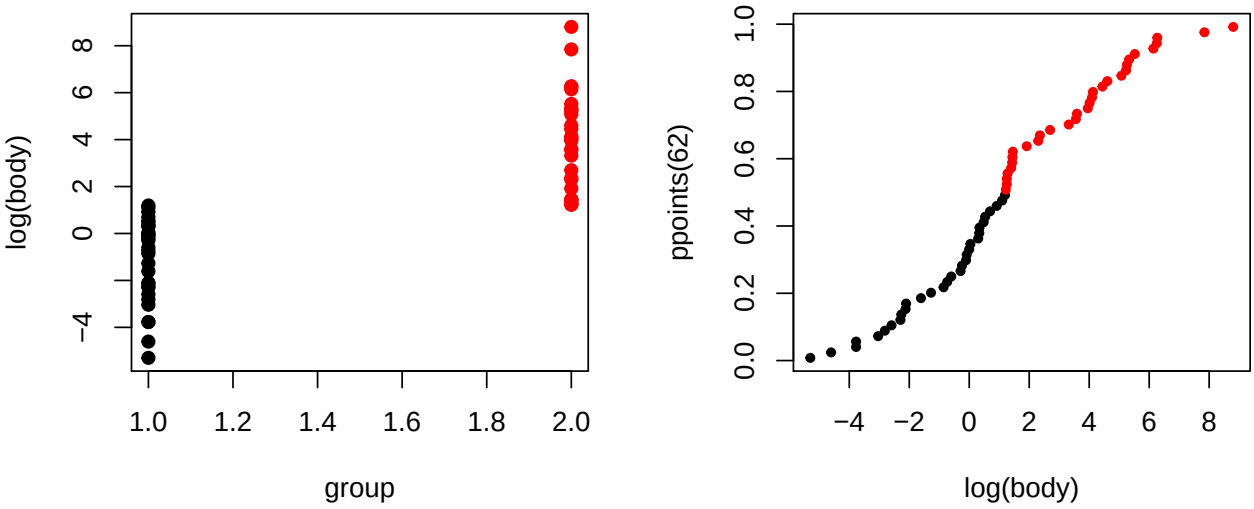


Regression with Stratified Sampling

- Assumption: for each unit we have the body weight.
  - The median log(body) weight is 1.21
  - Then two groups are below and above the median:

```
grp = numeric(nrow(mammals))
grp[log(mammals$body) <= median( log(mammals$body) )] = 1
grp[log(mammals$body) > median( log(mammals$body) )] = 2

summary( as.factor(grp))
```

```
##  1  2
## 31 31
```



Stratified Sampling Mechanism

```
createStrataMechanism <- function (pop, grp ) {

  method <- function(sampSize) {
    sam = list()
    for (h in 1:length(sampSize)) sam[[h]] = sample(pop[grp == h], sampSize[h])
    sam
    }
  return(method)
}

mammalStrata = createStrataMechanism(1:62, grp)
set.seed(341)
mammalStrata(c(2,3))
```

```
## [[1]]
## [1] 11 55
##
## [[2]]
## [1] 28  4 46
```

- Then we might want to apply unlist for compatibility

```
unlist(mammalStrata(c(2,3)))
```

```
## [1] 52 11 19 46  5
```

Notice that the 5 obervations above have been sampled using stratified sampling.

Stratified sampling (2 groups) for regression example

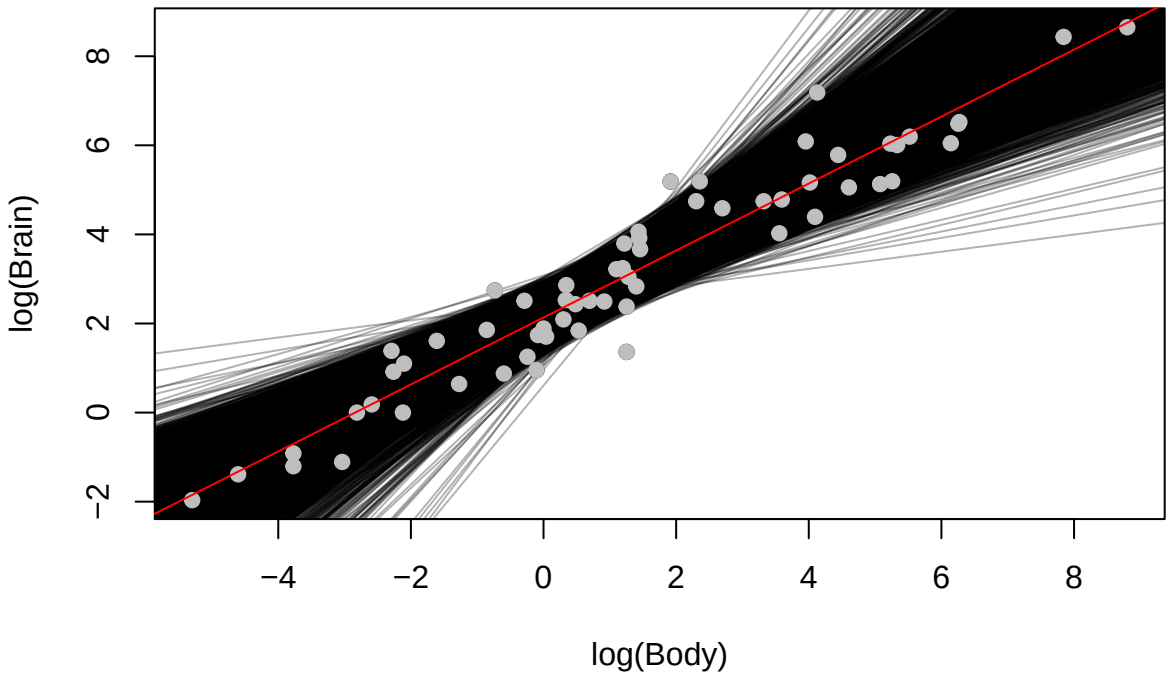
- For size 6 (stratified sampling), we have too many possibilities.

```
choose(31,3)*choose(31,3)
```

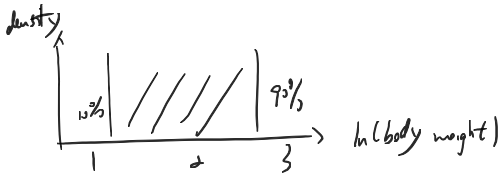
```
## [1] 20205025
```

- Here we use a random subset of the samples to quantify the sampling design.

```
N = nrow(mammals)
n = 6
m = 10000

set.seed(341)
reg.coef <- Map(function(rep) {
  lm( log(brain) ~ log(body), data=mammals, subset=unlist(mammalStrata(c(3,3))) )$coef },  1:m)

plot( log(mammals$body), log(mammals$brain), xlab="log(Body)", ylab="log(Brain)", pch=19)

for (i in 1:m) abline( reg.coef[[i]], col=adjustcolor("black", alpha = 0.3) )

points( log(mammals$body), log(mammals$brain), pch=19, col="grey")
abline( lm( log(brain) ~ log(body), mammals )$coef, col=2 )
```



Regression with 3 strata



- Limit the number in the middle

```
quantile(log(mammals$body), c(.1,.9) )
```

```
##       10%       90%
## -2.560504  5.325196
```

```
grp = numeric(nrow(mammals))
grp[log(mammals$body) <= -2.56 ] = 1
grp[log(mammals$body) <= 5.32 & log(mammals$body) > -2.56 ] =2
grp[ log(mammals$body) > 5.32 ] =3
```

The proportion of each group

```
##              1          2         3
## [1,] 7.0000000 48.0000000 7.0000000
## [2,] 0.1129032  0.7741935 0.1129032
```

- Construct the strata sampling mechanism

```
mammalStrata3 = createStrataMechanism(1:62, grp)
set.seed(341)
mammalStrata3(c(2,3,1))
```

```
## [[1]]
## [1] 15 53
##
## [[2]]
## [1] 25  3 50
##
## [[3]]
## [1] 33
```

Sample Size 6 with 3 strata

- For size 6, strata sampling, we have too many.

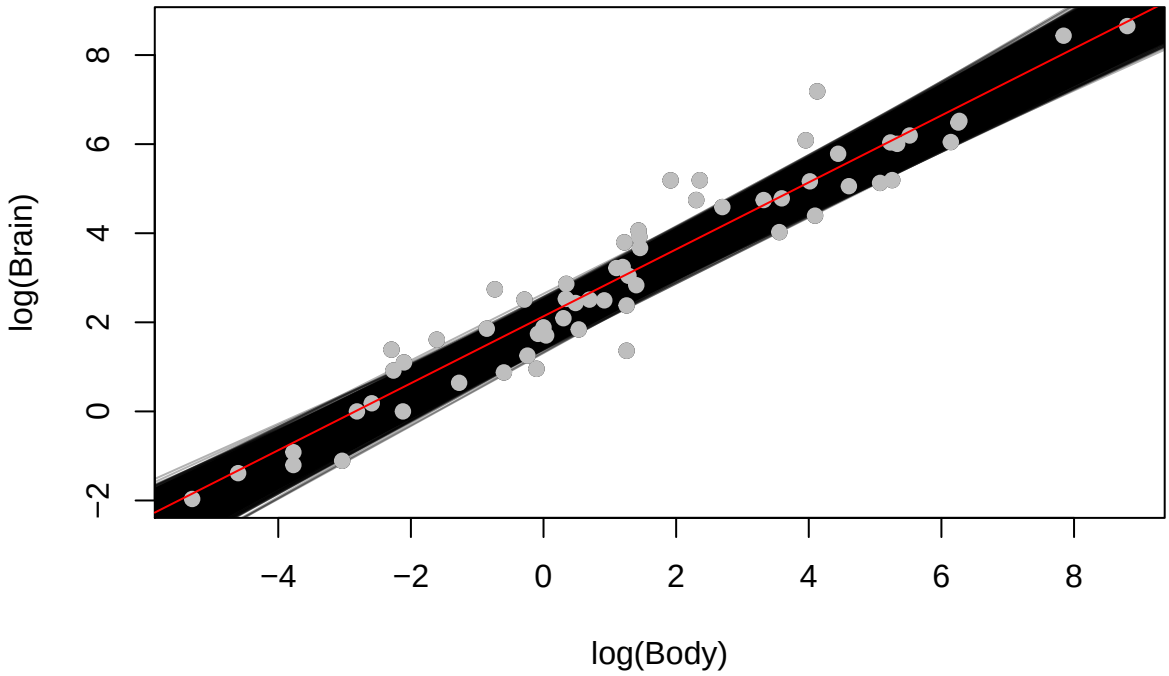
```
choose(7,2)*choose(48,2)*choose(7,2)
```

```
## [1] 497448
```

- Here we use a random subset of the samples to quantify the sampling design.

```
N = nrow(mammals)
n = 6
m = 10000

set.seed(341)
reg.coef <- Map(function(rep) {
  lm( log(brain) ~ log(body), data=mammals, subset=unlist(mammalStrata3(c(2,2,2))) )$coef },  1:m)

plot( log(mammals$body), log(mammals$brain), xlab="log(Body)", ylab="log(Brain)", pch=19)

for (i in 1:m) abline( reg.coef[[i]], col=adjustcolor("black", alpha = 0.3) )
```

```r
points( log(mammals$body), log(mammals$brain), pch=19, col="grey")
abline( lm( log(brain) ~ log(body), mammals )$coef, col=2 )
```

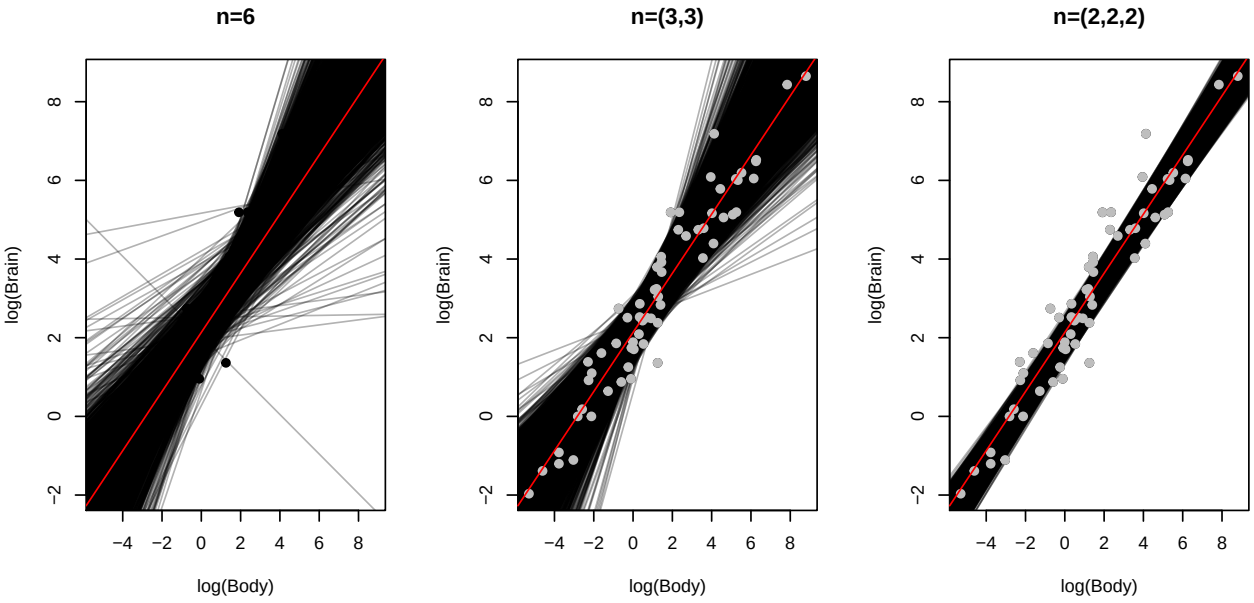


Comparing the sampling designs

- Using sample size 6
  - Random sampling without replacement (one strta)
  - Two strata with even split.
  - Three strata with varying split.

```r
par(mfrow=c(1,3),oma=c(0,0,0,0))

N = nrow(mammals)
n = 6
m = 10000

set.seed(341)
reg.coef <- Map(function(rep) {
  lm( log(brain) ~ log(body), data=mammals, subset=sample(N, n) )$coef },  1:m)

plot( log(mammals$body), log(mammals$brain), xlab="log(Body)", ylab="log(Brain)", pch=19, main="n=6")

for (i in 1:m) abline( reg.coef[[i]], col=adjustcolor("black", alpha = 0.3) )

points( log(mammals$body), log(mammals$brain), pch=19, col=0)
abline( lm( log(brain) ~ log(body), mammals )$coef, col=2 )



set.seed(341)
reg.coef <- Map(function(rep) {
```

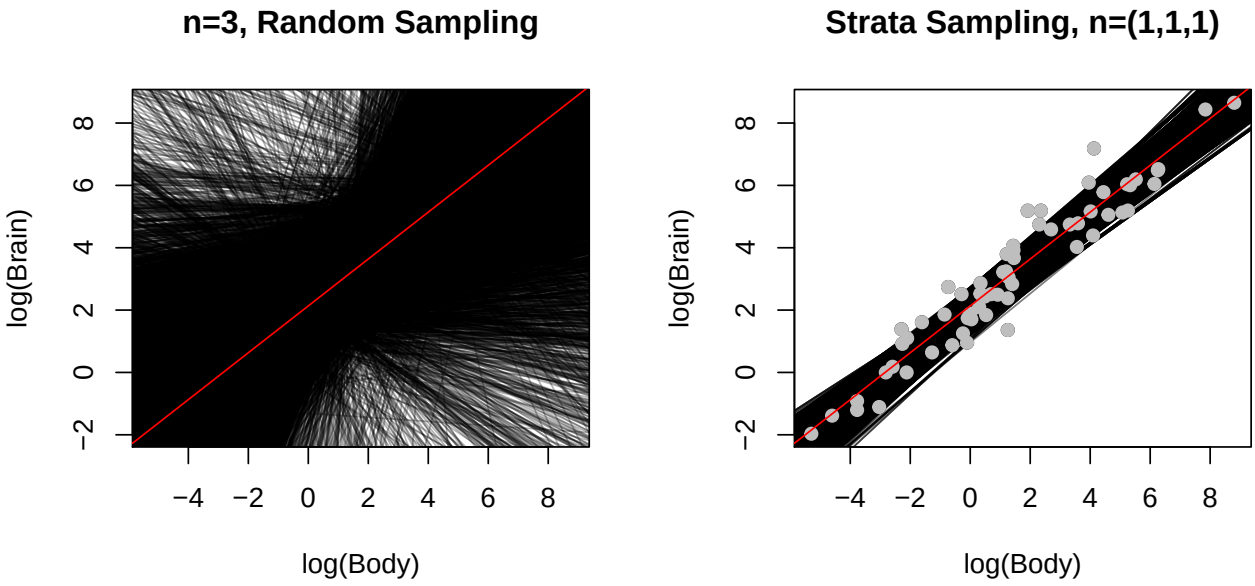
```
  lm( log(brain) ~ log(body), data=mammals, subset=unlist(mammalStrata(c(3,3))) )$coef },  1:m)

plot( log(mammals$body), log(mammals$brain),
      xlab="log(Body)", ylab="log(Brain)", pch=19, main="n=(3,3)")

for (i in 1:m) abline( reg.coef[[i]], col=adjustcolor("black", alpha = 0.3) )

points( log(mammals$body), log(mammals$brain), pch=19, col="grey")
abline( lm( log(brain) ~ log(body), mammals )$coef, col=2 )


N = nrow(mammals)
n = 6
m = 10000

set.seed(341)
reg.coef <- Map(function(rep) {
  lm( log(brain) ~ log(body), data=mammals, subset=unlist(mammalStrata3(c(2,2,2))) )$coef },  1:m)

plot( log(mammals$body), log(mammals$brain),
      xlab="log(Body)", ylab="log(Brain)", pch=19, main="n=(2,2,2)")

for (i in 1:m) abline( reg.coef[[i]], col=adjustcolor("black", alpha = 0.3) )

points( log(mammals$body), log(mammals$brain), pch=19, col="grey")
abline( lm( log(brain) ~ log(body), mammals )$coef, col=2 )
```



- **Note:** In all three plots we have taken a sample of size 6, and the difference is only due to sampling design.

Sample Size 3 with 3 strata

- For sample size $n = (1, 1, 1)$ with 3 strata compared to random sampling without replacement $n = 3$



This is the power of sampling design in improving estimation.