Selecting Samples

Contents

3.2 Selecting Samples

Interest still lies in summarizing a population $\mathcal{P}$ with an attribute $a(\cdot)$ but now we do so by calculating the value of that attribute on a sample $\mathcal{S} \subset \mathcal{P}$:

$$a(\mathcal{S}) = \widehat{a(\mathcal{P})}$$

However, we must now acknowledge the existence of **sample error** and the possibility that:

$$a(\mathcal{S}) \neq a(\mathcal{P})$$

For any particular sample

- the attribute calculated based on the sample *could be* identical to the population attribute, or
- it might be so different that we would be completely misled about the true nature of the population attribute

We never which situation we're in

This is why it is important to think carefully about **how** to select the sample and – if possible – do so in such a way to mitigate sample error.

- Even when the latter is possible, enormous care must be taken so that our own prejudices and pre-conceptions about the population do not render a sample that is misleading.

Given the reality that is sample error, it would be nice to understand the **magnitude** of error that can be expected.

- The **sampling distribution** of the attribute $a(\mathcal{S})$ gives insight into this. Properties of this distribution can be determined
  - *exactly*, when all possible samples are available
  - *approximately*, when a subset of all possible samples is considered
  - *in expectation*, when a probabilistic sampling mechanism is used to draw a single sample

- We've already considered the first case. We briefly discuss the second below, but the majority of this section is devoted to third case – which is the most realistic of the three.

What's the probability that $p < a(S) < q$?

3.2.0 Randomly Selecting m Samples

Consider drawing samples of size n from the population $\mathcal{P}$. The population $\mathcal{P}_{\mathcal{S}}$ of *all such samples* has size $M = \binom{N}{n}$ and is denoted

$$\mathcal{P}_{\mathcal{S}} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_M\}$$

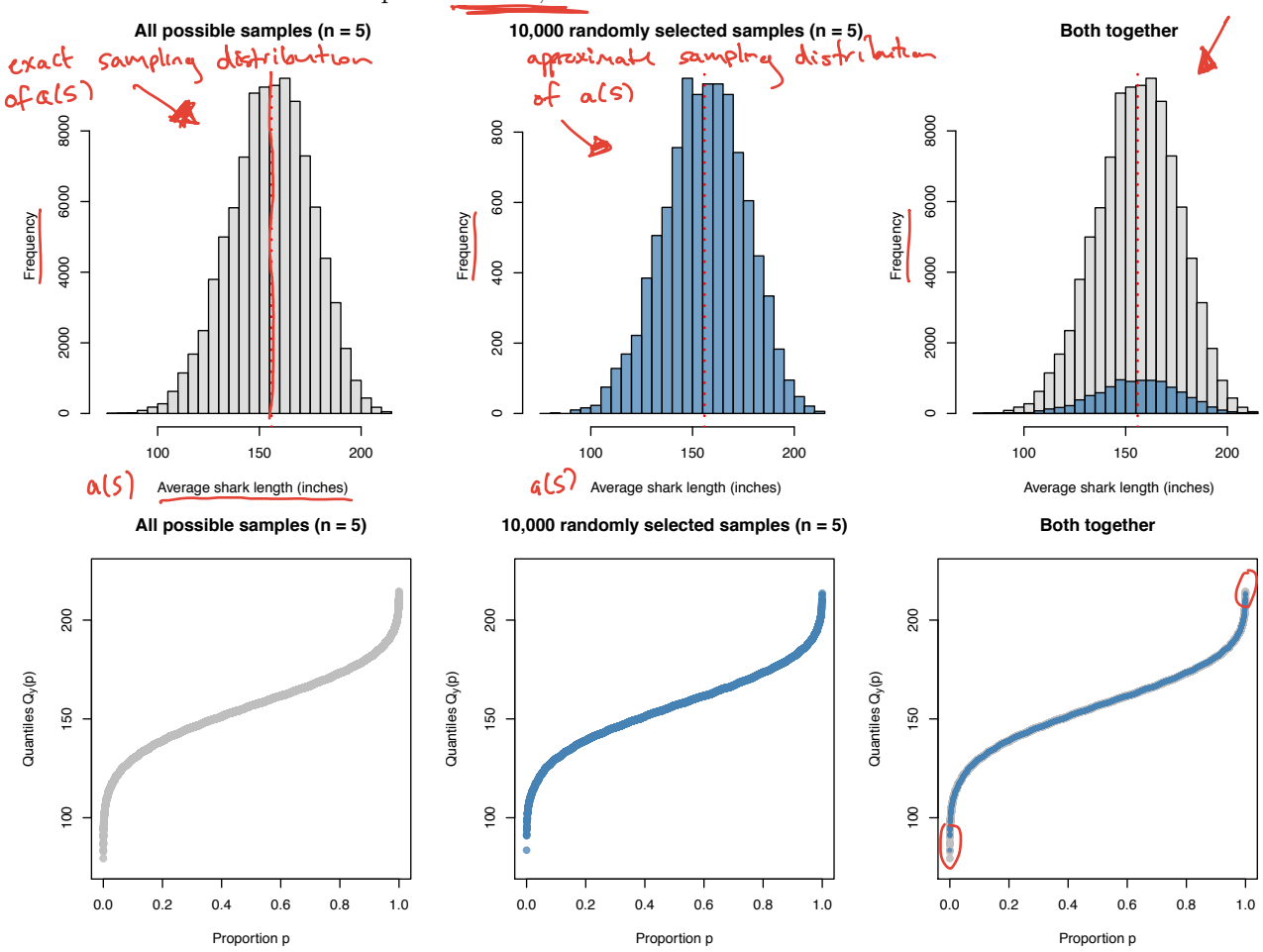
- Any attribute $a(\mathcal{S}_i)$ is now just a variate on that unit!
  - We then have a population of attributes

$$\mathcal{P}_{a(\mathcal{S})} = \{a(\mathcal{S}_1), a(\mathcal{S}_2), \ldots, a(\mathcal{S}_M)\}$$ ← all possible attribute values

In general, it computationally expensive to calculate all possible samples.

- Instead, suppose we randomly select m samples $\mathcal{S}_{u_1}, \ldots, \mathcal{S}_{u_m}$ from the population $\mathcal{P}_{\mathcal{S}}$ of $M = \binom{N}{n}$ possible samples.

- Let's revisit the Shark Data and in particular the subpopulation of shark encounters in Australian waters.
- Here there are $M = \binom{28}{5} = 98,280$ possible samples of size $n = 5$
- Let's consider a sample of $m = 10,000$ of these



- What do we learn from these plots?

We don't need all possible samples. Looking at a subset of them provides a very good approximation of the sampling distribution of a(s) and hence the sampling distribution of the errors a(s)-a(P).

Exercise: Regenerate the plots above. Note that using the argument `add=TRUE` in the `hist` function will be handy.

This was not a coincidence

Interesting Aside: The Distribution of a Histogram

- Suppose the histogram based on all possible samples has K bins

$$B_1 = (b_0, b_1], B_2 = (b_1, b_2], \ldots, B_K = (b_{K-1}, b_K]$$

where the k^{th} bin B_k contains $M_k \geq 0$ of the attribute values $a(S_i)$ $i = 1, \ldots, M$

- The bins contain the attribute values of all of the $\mathcal{S}_i \in \mathcal{P}_\mathcal{S}$ so that $\sum_{k=1}^{K} M_k = M$.

- Now suppose that we select m samples at random from $\mathcal{P}_\mathcal{S}$ such that probability that any given sample is selected is

$$p(\mathcal{S}) = \frac{1}{M}$$

  - Let m_k be the number of the m selected samples whose attribute value falls in B_k, with $m = \sum_{k=1}^{K} m_k$.

- With this notation,
  - the histogram based on all possible samples has heights $M_1, \ldots, M_K$ and
  - the histogram based on a sample of m of the possible samples has heights $m_1, \ldots, m_K$.

heights in grey histogram

heights in blue histogram

- The probability of any particular histogram arising from a random selection of m samples is therefore a **multivariate hypergeometric** probability

$$\frac{\binom{M_1}{m_1}\binom{M_2}{m_2}\cdots\binom{M_K}{m_K}}{\binom{M}{m}}$$

which, when $m << M$ can be approximated by the **multinomial** probability

$$\frac{m!}{m_1!\, m_2! \cdots m_K!} \binom{m}{m_1 \;\; m_2 \;\; \cdots \;\; m_K} p_1^{m_1} p_2^{m_2} \cdots p_K^{m_K}$$

this is approximately equal to this

with probabilities $p_k = \frac{M_k}{M}$ for $k = 1, \ldots, K$.

- From the multinomial, the expected value of the number of attribute values in each bin B_k is proportional to M_k (i.e. $mp_k = \frac{m}{M} M_k$). $\rightarrow$ $E[m_k] \propto M_k$

  - The *frequency* histogram based on m samples is (in expectation) a scaled version of that of all possible samples.

$E[m_k]$

  - The *density* histogram based on m samples is (in expectation) identical to that of all possible samples.

3.2.1 Quantifying Sample Error

In principle we select a sample $\mathcal{S}$ from the population $\mathcal{P}_\mathcal{S}$ containing all available samples

- We do so with some probability $p(\mathcal{S}) \geq 0$ of being selected. We require of course that

$$\sum_{\mathcal{S} \in \mathcal{P}_\mathcal{S}} p(\mathcal{S}) = 1.$$

- For any sample $\mathcal{S} \in \mathcal{P}_{\mathcal{S}}$, we have its **sample error**

$$\text{Sample Error} = a(\mathcal{S}) - a(\mathcal{P}).$$

  - Recall, for any collection of samples (or population of samples) $\mathcal{P}_{\mathcal{S}}$ containing M samples, we can calculate the **average sample error**

$$\text{Average Sample Error} = \frac{1}{M} \sum_{\mathcal{S} \in \mathcal{P}_{\mathcal{S}}} \left(a(\mathcal{S}) - a(\mathcal{P}) \right).$$

We can quantify the concentration of sample errors in expectation using quantities such as **sampling bias**, **sampling variance**, and **sampling mean squared error**.

- By sampling $\mathcal{S}$ randomly from $\mathcal{P}_{\mathcal{S}}$ with probability $p(\mathcal{S})$ we define the **sampling bias** as

$$\begin{aligned} \text{Sampling Bias} &= E[a(\mathcal{S})] - a(\mathcal{P}) \\ &= \sum_{\mathcal{S} \in \mathcal{P}_{\mathcal{S}}} a(\mathcal{S}) p(\mathcal{S}) - a(\mathcal{P}) \\ &= \sum_{\mathcal{S} \in \mathcal{P}_{\mathcal{S}}} \left(a(\mathcal{S}) - a(\mathcal{P}) \right) p(\mathcal{S}) \end{aligned}$$

  - Sampling bias is just the **expected** sample error induced by the repeated random sampling of $\mathcal{S}$ from $\mathcal{P}_{\mathcal{S}}$. If $p(\mathcal{S}) = \frac{1}{M}$, the sampling bias is identical to the average sample error of $a(\mathcal{P})$.
  - The sampling bias depends on the attribute $a(\cdot)$, the set of possible samples $\mathcal{P}_{\mathcal{S}}$, and the sample probabilities $p(\mathcal{S})$.
  - **Note:** If sampling bias is 0, then $a(\mathcal{S})$ is called an **unbiased** estimator of $a(\mathcal{P})$.

- The **sampling variance** is defined as

$$Var\left[a(\mathcal{S})\right] = E\left[\left(a(\mathcal{S}) - E[a(\mathcal{S})] \right)^2 \right]$$

  - This quantifies dispersion in the sample errors
  - We may also use the **sampling standard deviation** defined as the square root the variance

- Given a sample $\mathcal{S}$, we would like $a(\mathcal{S})$ and $a(\mathcal{P})$ to be as close as possible. We can use the **mean squared error** to quantify the expected squared distance between these two quantities

$$\begin{aligned} MSE\left[a(\mathcal{S})\right] &= E\left[(a(\mathcal{S}) - a(\mathcal{P}))^2\right] \quad (1) \\ &= Var\left[a(\mathcal{S})\right] + [\text{Sampling Bias}]^2 \quad (2) \end{aligned}$$

  - Ideally, we would like to choose $p(\mathcal{S})$ and/or $\mathcal{P}_{\mathcal{S}}$, so that both the square of sampling bias and the sampling variance are as small as possible.

$$(1) = E\left[\left(a(\mathcal{S}) - E[a(\mathcal{S})] + E[a(\mathcal{S})] - a(\mathcal{P}) \right)^2 \right]$$

$$= E\left[\left\{ \left(a(\mathcal{S}) - E[a(\mathcal{S})] \right) + \left(E[a(\mathcal{S})] - a(\mathcal{P}) \right) \right\}^2 \right]$$

$$= E\left[\left(a(\mathcal{S}) - E[a(\mathcal{S})] \right)^2 + \left(E[a(\mathcal{S})] - a(\mathcal{P}) \right)^2 + 2 \left(a(\mathcal{S}) - E[a(\mathcal{S})] \right) \left(E[a(\mathcal{S})] - a(\mathcal{P}) \right) \right]$$

$= E[(a(S) - E[a(S)])^2] + (E[a(S)] - a(P))^2 + 2(E[a(S)] - a(P))\, E[a(S) - E[a(S)]]$ (the last factor $\to 0$)

$= Var[a(S)] + Bias^2$ ← this decomposition is the basis for what's called the "bias-variance trade off"

$= (2)$

*Note that all expectations are taken with respect to the probabilities $p(\mathcal{S})$ of choosing a sample $\mathcal{S}$ from $\mathcal{P}_\mathcal{S}$.

Attribute as an Estimator (Using UWaterloo notation we can this $\widetilde{a(S)}$)

- Thinking of the sampling distribution of an attribute $a(\mathcal{S})$ gives rise to the notion of an attribute as an **estimator** (i.e., as a random variable).
- We can introduce a **random variable**, say A ($= \widetilde{a(S)}$), that takes values a from the distinct values of $a(\mathcal{S})$ for all $\mathcal{S} \in \mathcal{P}_\mathcal{S}$. The induced probability distribution is

$$\Pr(A = a) = \sum_{\mathcal{S} \in \mathcal{P}_\mathcal{S}} p(\mathcal{S}) \times I_{\{a\}}(a(\mathcal{S}))$$

← Proportion of all possible samples whose attribute value was a.

where $I_X(x)$ is the usual indicator function defined for any x and set X as

$$I_X(x) = \begin{cases} 1 & \text{if } x \in X \\ 0 & \text{otherwise.} \end{cases}$$

  - It follows that A is a discrete random variable.
  - Probability statements about its values can be made using its distribution, including its expectation, variance, etc.
  - Each of the definitions above (sampling bias, sampling variance, sample MSE) can be defined in terms of this random variable and the corresponding probability distribution.

Exercise: Express the sampling bias and the sampling variance in terms of this random variate.

Example – Comparing Different Sampling Designs

In this toy example we illustrate two different **sampling designs** (i.e., two different ways of defining $p(\mathcal{S})$ for $\mathcal{S} \subset \mathcal{P}$). We compare these methods of selecting $\mathcal{S}$ on the basis of bias, variance and mean squared error.

- Suppose that the population $\mathcal{P}$ consists of $N = 5$ units:

```
set.seed(341)
pop5 = round(rnorm(5),2)
pop5 = sort(pop5)
pop5 # our population
```

```
## [1] -1.06 -0.99 -0.31  0.83  0.87
```
← $\mathcal{P}$

- Now suppose we're interested in taking a sample of size $n = 2$. All possible samples of size 2 are:

```
sam2 = combn(5,2)
colnames(sam2) = paste("S", 1:10, sep="")
sam2
```

```
##       S1 S2 S3 S4 S5 S6 S7 S8 S9 S10
## [1,]  1  1  1  1  2  2  2  3  3   4
## [2,]  2  3  4  5  3  4  5  4  5   5
```

Note that the elements of the matrix above are the indices of the samples, not the actual values of the units.

- Next, we calculate the attribute (the mean) on theses samples.

```
sam.avg <- apply(sam2, MARGIN = 2, FUN = function(s){mean(pop5[s])})
round(sam.avg,3)
```

```
##     S1     S2     S3     S4     S5     S6     S7     S8     S9    S10
## -1.025 -0.685 -0.115 -0.095 -0.650 -0.080 -0.060  0.260  0.280  0.850
```

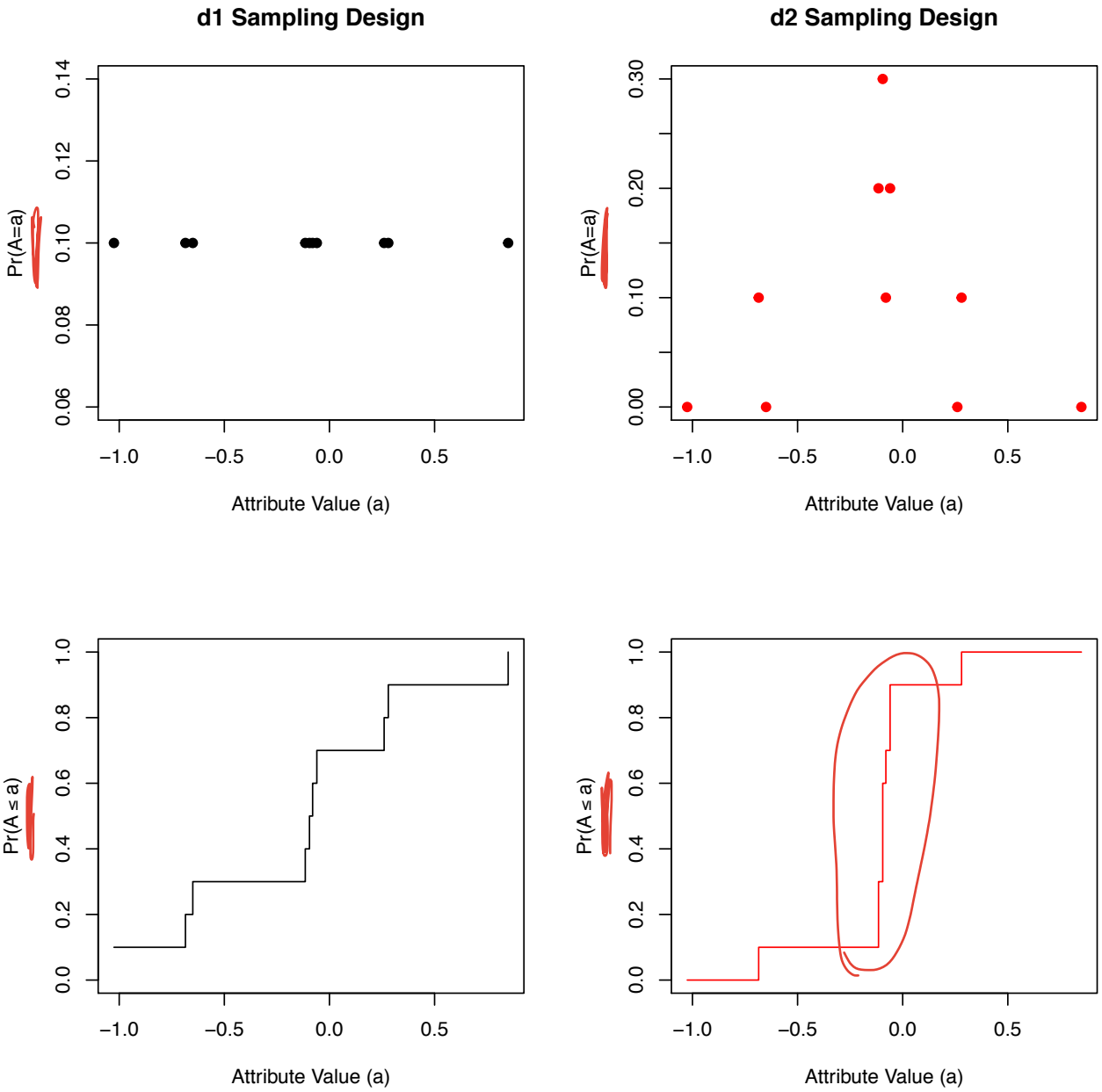
- Now consider two sampling designs:
  - d1: each sample is selected from the 10 possible samples with the same probability: $p(\mathcal{S}) = \frac{1}{10}$ $\forall \mathcal{S}$.
  - d2: different samples have different probabilities of being selected. There is no intuition behind this design, we use it simply for illustration.

```
d1 = rep(1/10,10)
d2 = 2*(abs(apply(sam2, 2, diff))-1)
d2 = d2/sum(d2)
designs = rbind(d1,d2)
colnames(designs) = paste('S', 1:10, sep="")
round( designs,2)
```

```
##     S1  S2  S3  S4  S5  S6  S7  S8  S9 S10
## d1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
## d2 0.0 0.1 0.2 0.3 0.0 0.1 0.2 0.0 0.1 0.0
```

- The distribution of the attribute (here, the mean) induced by the sampling design, i.e. $\Pr(A = a_i)$ is:

```
avg.ord = order(sam.avg)

par(mfrow=c(2,2),oma=c(0,0,0,0))
plot(sam.avg[avg.ord], d1[avg.ord], xlab="Attribute Value (a)",
     ylab="Pr(A=a)", pch=19,main="d1 Sampling Design")
plot(sam.avg[avg.ord], d2[avg.ord], xlab="Attribute Value (a)",
     ylab="Pr(A=a)", pch=19,col=2,main="d2 Sampling Design")

plot(sam.avg[avg.ord], cumsum(d1[avg.ord]), xlab="Attribute Value (a)",
     ylab=expression("Pr(A<=a)"), pch=19, type='s',ylim=c(0,1))
plot(sam.avg[avg.ord], cumsum(d2[avg.ord]), xlab="Attribute Value (a)",
     ylab=expression("Pr(A<=a)"), pch=19,col=2, type='s',ylim=c(0,1))
```



Note that the distribution of the attribute with respect to design `d2` is more concentrated.

Let's evaluate the sampling error associated with the attribute (here, the mean) with respect to both sampling designs.

- We compare the two sampling designs `d1` and `d2` numerically using the sampling bias, sampling variance and sampling MSE

Sampling bias

```
exp1 = sum(sam.avg*d1)
exp2 = sum(sam.avg*d2)
```

a(s) is unbiased for a(P) under sampling design d1

```
sam.bias=  c(exp1, exp2) - mean(pop5)
round(sam.bias,5)
```

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

Sampling Variance

```
sam.var = c( sum( (sam.avg-exp1)^2 * d1 ), sum( (sam.avg- exp2)^2*d2 ) )
round(sam.var,5)
```

```
## [1] 0.26689 0.04893
```

the sampling distribution of a(s) is much less variable under sampling design d2

Sampling MSE

```
sam.MSE = sam.var + sam.bias^2
round(sam.MSE,5)
```

```
## [1] 0.26689 0.04933
```

Alternatively, for our MSE calculation we could use the formula

$$MSE\left[a(\mathcal{S})\right] = E\left[(a(\mathcal{S}) - a(\mathcal{P}))^2\right]$$

```
MSE = c( sum( (sam.avg-mean(pop5))^2 * d1 ), sum( (sam.avg- mean(pop5))^2*d2 ) )
round(MSE,5)
```

```
## [1] 0.26689 0.04933
```

this is much smaller

Question: So which sampling design is better?

Since the MSE associated with d2 is smaller, it is to be preferred since sample values arising in this way tend to be much closer to the true population value than with d1.

3.2.2 A Note on Large Populations

- As we've discussed, as the population size increases constructing all possible samples becomes prohibitive
  - For example, consider the agricultural census of US counties whose population consists of only $N = 3078$ counties. For $n = 100$, there are $\binom{3078}{100}$ or about 1.4×10^{190} possible samples.

- The combinatorial explosion is avoided if we examine only m, say $m = 10{,}000$, samples.
  - Unfortunately, if we have to enumerate all possible samples just to select from them we are no farther ahead.

- We have discussed methods of quantifying the distribution of sample errors, but at present these definitions assume that either $\mathcal{P}_{\mathcal{S}}$ is available or that $\Pr(A = a)$ is known. Neither of which is very realistic.

✳ Considering all possible samples a useful means to quantify sample error and a valid means for selecting a sample
  - But we require an approach to selecting samples and evaluating sample errors that reflects the realistic scenario in which *just one* sample is ever observed.

3.2.3 Sampling Mechanisms

Rather than select samples at random from all possible samples, the same outcome is effected by sampling the units that will appear in any particular sample

- In other words: rather than select $\mathcal{S}$ with probability $p(\mathcal{S})$ from $\mathcal{P}_\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N\}$ we form $\mathcal{S}$ by selecting n units $u_{i_1}, u_{i_2}, \ldots, u_{i_n}$ directly from the population of units $\mathcal{P} = \{u_1, u_2, \ldots, u_N\}$.
- Each unit u in a sample $\mathcal{S}$ is selected one at a time from the population $\mathcal{P}$.
- A *sequence* of the first k units u_i selected from $\mathcal{P}$ is

$$s_k = (u_{i_1}, u_{i_2}, \ldots, u_{i_k})$$

- A **sampling mechanism** is defined by the probabilities

$$\Pr(u) \quad \text{and} \quad \Pr(u \mid k, s_{k-1}).$$

  where
  - the first unit is selected with probability $\Pr(u)$, and
  - the probability of the sequence of the first k units selected is

$$\Pr(s_k) = \Pr(u_{i_1}) \times \Pr(u_{i_2} \mid 2, s_1) \times \Pr(u_{i_3} \mid 3, s_2) \times \cdots \times \Pr(u_{i_k} \mid k, s_{k-1}).$$

- To determine $p(\mathcal{S})$ from a sampling mechanism:
  - Recognize that the order in which the units appear does not matter, i.e. any of the $n!$ permutations of the elements of s_n counts as $\mathcal{S}$
  - $p(\mathcal{S})$ is thus the sum of $\Pr(s_n)$ over all permutations s_n.

Simple Random Sampling Without Replacement (SRSWOR)

- The **sampling mechanism** is

$$\Pr(u) = \frac{1}{N} \quad \text{and} \quad \Pr(u \mid k, s_{k-1}) = \frac{1}{N-k+1}$$

- The probability of the sequence s_n is

$$\Pr(s_n) = \frac{1}{N} \times \frac{1}{N-1} \times \frac{1}{N-2} \times \cdots \times \frac{1}{N-n+1}$$

  which is the same for all $n!$ permutations, so

$$p(\mathcal{S}) = \frac{n!}{N(N-1)(N-2)\cdots(N-n+1)} = \frac{1}{\binom{N}{n}}$$

  This probability is the same as we had before for selecting n distinct units from a population of N distinct units.

However, we now have a mechanism that allows us to select a sample *without first enumerating* all $M = \binom{N}{n}$ possible samples in $\mathcal{P}_{\mathcal{S}}$.

- In R, the indices of a simple random sample without replacement of size n from indices $1, \ldots, N$ is returned from the function call `sample(N,n)`.

```
set.seed(341)
pop15 = round(rnorm(15),2)
pop15
```

```
##  [1] -1.06 -0.31  0.87 -0.99  0.83  0.47 -0.66 -0.05  1.46 -0.72  0.82
## [12]  1.34  1.78 -1.12 -0.76
```

```
set.seed(341)
sample5 = sample(15, 5)
sample5
```

```
## [1]  3 13  5  1  9
```

```
pop15[sample5]
```

```
## [1]  0.87  1.78  0.83 -1.06  1.46
```

- If rather than indices, the units were identified by the (assumed unique) contents of a vector `Pop`, then `sample(Pop,n)` would return the vector of units in the sample.

```
set.seed(341)
sample(pop15, 5)
```

```
## [1]  0.87  1.78  0.83 -1.06  1.46
```

Simple Random Sampling With Replacement (SRSWR)

- The **sampling mechanism** is

$$\Pr(u) = \frac{1}{N} = \Pr(u \mid k, s_{k-1})$$

and thus a sample $\mathcal{S}$ can contain have one or replicated units

- The probability of the sequence s_n is

$$\Pr(s_n) = \Pr(u_{i_1}) \times \Pr(u_{i_2} \mid 2, s_1) \times \Pr(u_{i_3} \mid 3, s_2) \times \cdots \times \Pr(u_{i_n} \mid n, s_{n-1}) = \left(\frac{1}{N}\right)^n$$

- Note that unlike in the case of SRSWOR, because each s_n contains possibly duplicated values we treat each s_n as a distinct sample and so

$$\Pr(\mathcal{S}) = \Pr(s_n) = \frac{1}{N^n}$$

  - The population of all samples $\mathcal{P}_{\mathcal{S}}$ in this case contains $M = N^n$ different samples.

- To generate simple random samples with replacement in R, we use `sample` as before except now with the argument `replace = TRUE` as in `sample(N, n, replace = TRUE)`.

```
set.seed(341)
pop15 = round(rnorm(15),3)

set.seed(341)
sample5 = sample(15, 5, replace=TRUE)
sample5
```

```
## [1]  3 14  6  1 13
```

```
pop15[sample5]
```

```
## [1]  0.866 -1.115  0.473 -1.060  1.783
```

```
sample(pop15, 5)
```

```
## [1]  1.338  0.866 -0.993 -0.720 -1.060
```

A Weird Hybrid Sampling Mechanism (Let's call it SRSWH)

- The following mechanism was first explored by Basu (1958).
- Suppose we perform simple random sampling with replacement except that we *remove* any duplicate units.
  - The samples produced will have sizes anywhere from 1 to n according to how many distinct units were selected in a sample (sampling with replacement).

```
set.seed(341)
pop10 = round(rnorm(10),3)
set.seed(341)
sample5 = sample(10, 5, replace=TRUE)
sample5
```

```
## [1] 2 9 4 1 9
```

- Simple random sample with replacement yields

```
pop10[sample5]
```

```
## [1] -0.308  1.462 -0.993 -1.060  1.462
```

- Simple random sample with replacement removing duplicate units yields

```
unique(pop10[sample5])
```

```
## [1] -0.308  1.462 -0.993 -1.060
```

*Note that since the number of duplicates is a random variable, the actual sample size (n minus the number of duplicates) is also a random variable here!

Connections to Balls in an Urn

- Suppose that we have an urn containing N different balls that are either white or black.
  - We would like to estimate the proportion of balls in the urn which are black by drawing n balls at random from the urn.

1. Simple random sampling **without** replacement (SWSWOR).
   - Randomly draw n balls from the urn one after another, **without replacing** any at any time.
   - The estimate is the proportion of black balls in your sample.
2. Simple random sampling **with** replacement (SRSWR).
   - Randomly draw n balls from the urn one after another, **each time replacing** the ball after drawing it.
   - The estimate is again the proportion of black balls in your sample.
3. Basu's Weird Hybrid (SRSWH).
   - Select one ball at a time and record its colour, mark it with an X and return it to the urn.
   - If a ball drawn already has an X marked on it, then it counts as a draw, but is returned to the box without recording its colour.
   - Continue in this way until n draws have been made.
   - The estimate is the proportion of black balls observed with X's.

Comparing Sampling Mechanisms – Australian Shark Encounters

- For a population of size N
  - there exist $\binom{N}{n}$ samples **without replacement**
  - there exist N^n samples **with replacement**
  - there exist somewhere between $\binom{N}{n}$ and N^n samples **with replacement but no duplicates**

- Using the Australian shark encounter population, if we take $n = 15$ samples,
  1. sampling without replacement yields a population $\mathcal{P}_S$ of size $M = \binom{28}{15} = 37{,}442{,}160.$
  2. for sampling with replacement, $\mathcal{P}_S$ is much larger, containing $M = 28^{15} = 5.097655 \times 10^{21} =$ 5,097,655,000,000,000,000,000 different possibilities.

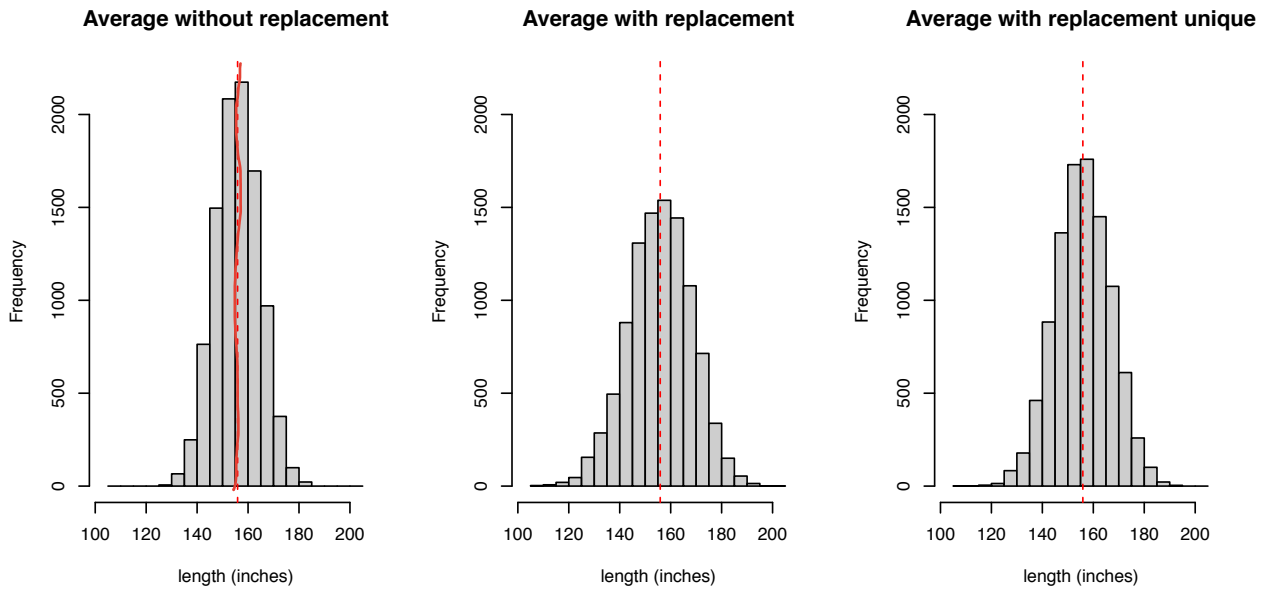
- Using each mechanism we construct $m = 10,000$ samples and for each sample calculate the average.

```
popSharks <- rownames(sharks)
popSharksAustralia <- popSharks[sharks$Australia == 1]
avePop <- mean(sharks[popSharksAustralia, "Length"])
```

```
### sample size
n <- 15
### number of samples
m <- 10000
```

```r
### reproducibiity
set.seed(123415)
### samples without replacement
sampsWithout <- Map(function(i){sample(popSharksAustralia, size=n, replace = FALSE)},
                    1:m)
### attribute evaluated on each sample
aveWithout <- Map(function(s){mean(sharks[s,"Length"])}, sampsWithout)


### samples with replacement
sampsWith <- Map(function(i){sample(popSharksAustralia, size=n, replace = TRUE)},
                 1:m)
### attribute evaluated on each sample
aveWith <- Map(function(s){mean(sharks[s,"Length"])}, sampsWith)

### samples with replacement but no duplicates
aveWithUnique <- Map(function(s){mean(sharks[unique(s),"Length"])}, sampsWith)

### Note that in both cases, there are so many samples to choose from
### that we are not going to worry about whether we have repeated any
### in the m we have selected from M
###

### Now prepare to plot histograms
###
### Use the same x scale in the plots
xlim <- extendrange(c(aveWith, aveWithout))
### and bins
bins <- hist(c(as.numeric(aveWithout), as.numeric(aveWith)),
             breaks = 30, plot=FALSE)
### And heights
ylim <- c(0, 2200)

### Without replacement
###
par(mfrow=c(1,3))
hist(as.numeric(aveWithout), main = "Average without replacement",
     xlim = xlim, xlab = "length (inches)", ylim = ylim,
     breaks = bins$breaks, col =  adjustcolor("grey", 0.75))
abline(v=avePop, col="red", lty=2)

### and with
hist(as.numeric(aveWith), main = "Average with replacement",
     xlim = xlim, xlab = "length (inches)", ylim = ylim,
     breaks = bins$breaks, col =  adjustcolor("grey", 0.75))
abline(v=avePop, col="red", lty=2)

### and with but no duplicates
hist(as.numeric(aveWithUnique), main = "Average with replacement unique",
     xlim = xlim, xlab = "length (inches)", ylim = ylim,
     breaks = bins$breaks, col =  adjustcolor("grey", 0.75))
abline(v=avePop, col="red", lty=2)
```

Approximate sampling distributions for a(S) = average shark length



- Here is a numerical summary of these sampling distributions:

```
##                                      Min 1st Qu. Median 3rd Qu.   Max
## Without Replacement                128.0   149.8  155.7   161.7 183.6
## With Replacement                   106.8   147.5  156.0   164.4 203.6
## With Replacement but no duplicates 107.9   148.4  155.8   163.4 192.2
```

- To compare these sampling mechanisms numerically we might also use bias, variance and MSE:

```
population=sharks[popSharksAustralia,]$Length

# the 10,000 averages based on different sampling designs
aveWithout = unlist(aveWithout)
aveWith = unlist(aveWith)
aveWithUnique = unlist(aveWithUnique)

average = matrix(0, nrow=length(aveWith), ncol=3)
average[,1] <- aveWithout
average[,2] <- aveWith
average[,3] <- aveWithUnique

temp = rbind(apply(average,2,mean) -mean(population),
apply(average,2,var), (apply(average,2,mean)-mean(population))^2 + apply(average,2,var) )
dimnames(temp)[[1]] = c("Bias", "Var", "MSE")
dimnames(temp)[[2]] = c("Without Replacement",
                        "With Replacement", "With Replacement but no duplicates")
round(t(temp),4)
```

```
##                                      Bias      Var      MSE
## Without Replacement                -0.0704  74.8888  74.8937
## With Replacement                   -0.0365 155.1258 155.1272
## With Replacement but no duplicates -0.0340 120.6517 120.6528
```

- What conclusions can we draw?

- SRSWOR is superior in terms of MSE
- SRSWR is (perhaps surprisingly) the worst in terms of MSE

* We see that samples with no duplicates are in some sense of higher quality, i.e., they better represent the population

Implementation of Sampling Mechanisms in R

We could implement any of the above sampling mechanisms as a single call to a factory function.

```
### This will create a sampling mechanism
createSamplingMechanism <- function (pop, method = c("withoutReplacement","withReplacement",
 "withUnique")) {

  if(method == "withoutReplacement"){
   function (sampSize) { sample(pop, sampSize, replace=TRUE) }
  }else if(method == "withReplacement"){
   function (sampSize) { sample(pop, sampSize, replace=FALSE) }
  }else if(method == "withUnique"){
   function (sampSize) { unique(sample(pop, sampSize, replace=TRUE)) }
  }else{
   stop(paste("No sampling mechanism:", method))
  }
}
```

For example, for simple random sampling without replacement on the population of all sharks, we might define a function `srswor(sampSize)` as

```
### without replacement is the default method.
srswor <- createSamplingMechanism(popSharks)
```

which now allows us to generate a sample of any size containing **units selected without replacement** from the population of all sharks.

- A sample of size 5, 10 and 30

```
set.seed(341)
srswor(5)
```

```
## [1] "10" "58" "24" "4"  "50"
```

```
srswor(10)
```

```
##  [1] "50" "11" "16" "65" "5"  "41" "1"  "15" "45" "27"
```

```
srswor(30)
```

```
##  [1] "62" "60" "42" "15" "1"  "48" "31" "53" "34" "54" "63" "8"  "3"  "12"
## [15] "25" "56" "59" "35" "61" "21" "22" "16" "30" "29" "27" "20" "36" "23"
## [29] "2"  "44"
```

- Similarly, for the unique units from a sample with replacement of some size:

```
set.seed(354661)
### create the sampling mechnism
uniquewr <- createSamplingMechanism(popSharks, method="withUnique")
### A sample uniquely from size 30 with replacement
uniquewr(30)
```

```
##  [1] "45" "31" "5"  "48" "42" "1"  "51" "56" "32" "65" "60" "36" "26" "30"
## [15] "22" "4"  "3"  "8"  "62" "21" "57" "35" "41" "29" "49"
```

```
uniquewr(30)
```

```
##  [1] "37" "55" "21" "61" "29" "11" "46" "16" "62" "13" "30" "51" "53" "5"
## [15] "65" "63" "39" "12" "58" "15" "1"  "19" "43" "36" "52" "20"
```

n=28

```
uniquewr(30)
```

```
##  [1] "15" "3"  "28" "55" "12" "25" "2"  "56" "49" "50" "45" "22" "6"  "61"
## [15] "20" "26" "8"  "48" "40" "1"  "36" "47" "13" "30"
```

n=24

Note that different sample sizes can result for this method.

- The created function will only generate samples from the population `pop` which allows us to write different sampling mechanisms that might actually depend on some features of the population.

3.2.4 Unit Inclusion Probabilities

- The **probability of inclusion** for unit u is the probability of unit u being included in the sample

$\pi_u = P(u \in S)$

- Such inclusion probabilities are of interest to calculate in addition to $p(\mathcal{S})$ and they may in fact be derived from $p(\mathcal{S})$.

- Consider the indicator function

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

D_u is a binary random variable that takes value 1 with probability $\Pr(u \in \mathcal{S})$ if the probability that the sample $\mathcal{S}$ contains u, and 0 otherwise.

- The probability that unit u is in $\mathcal{S}$ is

$$\begin{aligned} \pi_u &= E[D_u] \\ &= 1 \times \Pr(D_u = 1) + 0 \times \Pr(D_u = 0) \\ &= \Pr(u \in \mathcal{S}) \\ &= \textstyle\sum_{\mathcal{S}: u \in \mathcal{S}} p(\mathcal{S}) \end{aligned}$$

(marginal)

This is called the **inclusion probability** of u in the sample $\mathcal{S}$; it is the probability that the unit u will be in a sample $\mathcal{S}$ selected according to $p(\mathcal{S})$.

- The probability that u and v are in the sample $\mathcal{S}$ is called the **joint inclusion probability** and is given by

$$\begin{aligned} \pi_{uv} &= Pr(u \in \mathcal{S} \text{ and } v \in \mathcal{S}) \\ &= E[\, D_u \times D_v \,] \\ &= \textstyle\sum_{\mathcal{S}: u, v \in \mathcal{S}} p(\mathcal{S}) \end{aligned}$$

$1 \times P(u \in S, v \in S) + 0 \times P(u \in S, v \in S)$

Note that the sums in the preceding equations are over all $\mathcal{S} \in \mathcal{P}_{\mathcal{S}}$ containing the designated units.

Simple Random Sampling Without Replacement

- The inclusion probability is

$$\pi_u = \Pr(u \in \mathcal{S}) = \frac{1 \times \binom{N-1}{n-1}}{\binom{N}{n}} = \frac{1 \times \binom{N-1}{n-1}}{\frac{N}{n} \times \binom{N-1}{n-1}} = \frac{n}{N}$$

- The joint inclusion probability is

$$\pi_{uv} = \Pr(u \in \mathcal{S} \cap v \in \mathcal{S}) = \frac{1 \times 1 \times \binom{N-2}{n-2}}{\binom{N}{n}} = \frac{n(n-1)}{N(N-1)}$$

Simple Random Sampling With Replacement

- The inclusion probability is

$$\pi_u = 1 - \left(\frac{N-1}{N}\right)^n$$

- The joint inclusion probability is

$$\pi_{uv} = 1 - 2\left(\frac{N-1}{N}\right)^n + \left(\frac{N-2}{N}\right)^n$$

The Weird Hybrid Sampling Mechanism

The inclusion probabilities for sampling with replacement but using only the unique units selected (i.e. the "weird hybrid" mechanism discussed earlier due to Basu) are identical to simple random sampling with replacement.

3.2.5 Estimating Totals

Many attributes are either a total

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

of some variate y_u observed on every unit $u \in \mathcal{P}$, or a function of such a total

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

(Recall that a variate y is any function that when applied to any unit $u \in \mathcal{P}$ returns a value $y(u) = y_u$)

Common Totals and Functions Thereof

- The **population average** is a total:

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

- The **population variance** is a total:

$$a(\mathcal{P}) = \sum_{u \in \mathcal{P}} \frac{(y_u - \bar{y})^2}{N}$$

- The **population standard deviation** is a function of a total:

$$a(\mathcal{P}) = \sqrt{\sum_{u \in \mathcal{P}} \frac{(y_u - \bar{y})^2}{N}}$$

- The **number of** shark encounters from the sub-population of Australia ($\mathcal{A} \subset \mathcal{P}$) is a total:

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

$$I_X(x) = \begin{cases} 1 & \text{if } x \in X \\ 0 & \text{otherwise} \end{cases}$$

- The **proportion** of units with their variate value lying in the interval $[a, b]$ is a total:

$$a(\mathcal{P}) = \sum_{u \in \mathcal{P}} \frac{I_{[a,b]}(y_u)}{N}$$

- The **cumulative distribution function** (CDF) at a specific value y, $F_{\mathcal{P}}(y)$, is a total:

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

- A **quantile** (the inverse of the CDF) is thus an implicit function of a total:

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

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

The Horvitz-Thompson Estimate

- A natural (and very commonly used) estimate of a population total

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

is the **Horvitz-Thompson estimate** (due to Daniel G. Horvitz and Donovan J. Thompson, 1952) defined as

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

where the contribution for each unit in the sample is weighted inversely by π_u, its probability of inclusion in $\mathcal{S}$.

- if the probability of inclusion is small, then the weight will be high
- if the probability of inclusion is large, then the weight will be low

- Note that we use the subscript HT and to distinguish the Horvitz-Thompson estimate of $a(\mathcal{P})$ from other estimates based on the sample $\mathcal{S}$.

$a(S)$ ← sample attribute

The Horvitz-Thompson Estimator

Here we consider properties of the Horvitz-Thompson estimat**or** (i.e., the random variable), $\widetilde{a}_{HT}(\mathcal{S})$. Such properties inform what can be expected under repeated sampling.

Bias, Variance, and Mean Squared Error

In what follows tt will be convenient to work with the random variable

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

- Note that with this defined the Horvitz-Thompson estimator can be written as

$$\widetilde{a}_{HT}(\mathcal{S}) = \sum_{u \in \mathcal{S}} \frac{y_u}{\pi_u} = \sum_{u \in \mathcal{P}} D_u \times \frac{y_u}{\pi_u}$$

The following properties of D_u will also be useful:

- $E[D_u] = P(u \in S) = \pi_u$
- $Var[D_u] = E[D_u^2] - E[D_u]^2$

$= E[D_u] - E[D_u]^2$

$= \pi_u (1 - \pi_u)$

- $Cov[D_u, D_v] =$ $E[D_u D_v] - E[D_u]E[D_v] = \pi_{uv} - \pi_u \pi_v = \Delta_{uv}$

The HT estimator is unbiased !!

Let us derive the **bias** of the HT estimator:

$$\text{Bias} = E[\tilde{a}_{HT}(S) - a(P)] = E[\tilde{a}_{HT}(S)] - a(P)$$

$$= E\left[\sum_{u \in P} D_u \frac{y_u}{\pi_u}\right] - a(P) = \left(\sum_{u \in P} E[D_u] \times \frac{y_u}{\pi_u}\right) - a(P) = \left(\sum_{u \in P} y_u\right) - a(P) = 0$$

(π_u)

It can be shown that **variance** of the HT estimator is

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

$$Var\left[\sum_{u \in P} D_u \cdot \frac{y_u}{\pi_u}\right] = \sum_{u \in P} \sum_{v \in P} Cov(D_u, D_v) \frac{y_u}{\pi_u} \frac{y_v}{\pi_v} = \sum_{u \in P} \sum_{v \in P} \Delta_{uv} \frac{y_u}{\pi_u} \frac{y_v}{\pi_v}$$

Like $Var\left[\sum_i a_i X_i\right] = \sum_i \sum_j Cov(a_i X_i, a_j X_j) = \sum_i \sum_j a_i a_j Cov(X_i, X_j)$

This variance can be equivalently written in the **Yates-Grundy** or the **Sen-Yates-Grundy** formulation:

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

- The Yates-Grundy formulation provides an intuition for choosing sampling mechanisms that minimize the variance (as will be discussed further in Section 3.2.6).

* Note that because the HT estimator is unbiased the **mean squared error** is simply equal to the variance.

Exercise (SRSWOR)

Consider the simple random sampling without replacement mechanism.

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

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

Just plug in SRSWOR π_u and π_{uv}

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

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

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

This matches the formulae derived in infinite in intuition. This is just the finite population analog.

- If the last term in the R.H.S. of the equation in part (b) had the denominator $N-1$ instead of N, the *finite population correction* would be defined as $\left(1-\frac{n}{N}\right)=\left(\frac{N-n}{N}\right)$.

The HT Estimate of the Variance of the HT Estimator

- Define
$$q_{u,v}=\Delta_{uv}\frac{y_u}{\pi_u}\frac{y_v}{\pi_v}$$
and $\mathcal{P}_{uv}$ to be the population of all pairs (u,v) where $u,v\in\mathcal{P}$.
- The the variance of the HT estimator can be written as
$$Var\left[\widetilde{a}_{HT}(\mathcal{S})\right]=\sum_{u\in\mathcal{P}}\sum_{v\in\mathcal{P}}\Delta_{uv}\frac{y_u}{\pi_u}\frac{y_v}{\pi_v}=\sum_{(u,v)\in\mathcal{P}_{uv}}q_{u,v}$$
- Written this way we can see that this variance is in fact just a total!
- So.... we can estimate the variance of the HT estimator of a(P) with a HT estimate.

- The HT estimate of the variance of the HT estimator is:
$$\widehat{Var}\left[\widetilde{a}_{HT}(\mathcal{S})\right]=\sum_{(u,v)\in\mathcal{S}_{uv}}\frac{q_{u,v}}{\pi_{uv}}=\sum_{u\in\mathcal{S}}\sum_{v\in\mathcal{S}}\frac{\Delta_{uv}}{\pi_{uv}}\frac{y_u}{\pi_u}\frac{y_v}{\pi_v}=\sum_{u\in\mathcal{S}}\sum_{v\in\mathcal{S}}\left(\frac{\pi_{uv}-\pi_u\pi_v}{\pi_{uv}}\right)\frac{y_u}{\pi_u}\frac{y_v}{\pi_v}.$$
($\mathcal{S}_{uv}$: sample)

  - Note that the sample $\mathcal{S}_{uv}$ is obtained by sampling from the population $\mathcal{P}_{uv}$ of all possible pairs (u,v). The probability that any particular pair (u,v) is included in the sample is $\pi_{uv}>0$

  - Note also that the square root of this variance estimate is what is commonly referred to the **standard error** of the estimate $\widehat{SD}\left[\widetilde{a}_{HT}(S)\right]=\sqrt{\widehat{Var}\left[\widetilde{a}_{HT}(S)\right]}$ ← estimate of the standard deviation of $\widetilde{a}_{HS}(S)$

- Thus, using Horvitz-Thompson estimation we are able to construct
  - an estimate of the population total *and*
  - an estimate of the variance of this estimator *and*
  - both estimators are unbiased.

- Why is this useful?

1. Recall, many many attributes are totals, and so the HT framework gives us a intuitive and effective means of estimation.

2. Understanding sampling error requires just one sample.

HT Toy Example

Recall our example with the population consisting of $N = 5$ units.

```
pop5
```

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

Here we will explore Horvitz-Thompson estimation of the population mean using samples of size $n = 2$. Recall that the sample means in each of the $\binom{N}{n} = 10$ possible samples are:

```
round(sam.avg, 3)
```

```
##     S1     S2     S3     S4     S5     S6     S7     S8     S9    S10
## -1.025 -0.685 -0.115 -0.095 -0.650 -0.080 -0.060  0.260  0.280  0.850
```

Recall also that we had two different sampling designs:

```
round(designs, 2)
```

```
##     S1  S2  S3  S4  S5  S6  S7  S8  S9 S10
## d1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
## d2 0.0 0.1 0.2 0.3 0.0 0.1 0.2 0.0 0.1 0.0
```

And the sampling bias, variance and MSE associated with these designs were:

```
exp1 = sum(sam.avg*d1)
exp2 = sum(sam.avg*d2)

sam.bias=  c(exp1, exp2) - mean(pop5)

sam.var = c( sum( (sam.avg-exp1)^2 * d1 ), sum( (sam.avg- exp2)^2*d2 ) )


designs.MSE = rbind( sam.bias, sam.var, MSE=sam.var + sam.bias^2)
colnames(designs.MSE) = c("d1", "d2")
round(designs.MSE,3)
```

```
##             d1    d2
## sam.bias 0.000 0.020
## sam.var  0.267 0.049
## MSE      0.267 0.049
```

For the HT calculations below we will focus on design `d2` (because HT estimation in `d1` is no different than ordinary estimation via the sample mean [Why?]).

- In order to perform Horvitz-Thompson estimation we require the **marginal inclusion probabilities** for each unit in the population and the **joint inclusion probabilities** for each possible pair.

- Recall that in order to calculate these we sum $p(\mathcal{S})$ over all samples that contain the unit (or pair of units) of interest.

Marginal Inclusion Probabilities (`d2` design)

For every sample, find which units are in the sample

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

pop5sam2.units = combn(5, 2, inSample, N=5)
rownames(pop5sam2.units) = paste("unit", 1:5, sep="")
colnames(pop5sam2.units) = paste("S", 1:10, sep="")
pop5sam2.units
```

```
##       S1 S2 S3 S4 S5 S6 S7 S8 S9 S10
## unit1  1  1  1  1  0  0  0  0  0   0
## unit2  1  0  0  0  1  1  1  0  0   0
## unit3  0  1  0  0  1  0  0  1  1   0
## unit4  0  0  1  0  0  1  0  1  0   1
## unit5  0  0  0  1  0  0  1  0  1   1
```

Now for each unit, add up $p(\mathcal{S})$ for each sample it appears in and that will yield π_u:

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

pi2 = apply( pop5sam2.units, 1, weighted.sum, w=d2)
pi2
```

weighted row sum where the weights are the probability that a particular sample would be observed.

```
## unit1 unit2 unit3 unit4 unit5
##   0.6   0.3   0.2   0.3   0.6
```

π_1 π_2 π_3 π_4 π_5

Joint Inclusion Probabilities (d2 design)

For every sample, find which pairs of units are in the sample

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

#this will generate a 5x5x10 tensor
jsample.incl = combn(5,2, FUN=inSample2, N=5)
dimnames(jsample.incl) = list(paste("unit", 1:5, sep=""),
                              paste("unit", 1:5, sep=""),
                              paste("S", 1:10, sep="") )
jsample.incl[,,1:3]
```

```
## , , S1
##
##       unit1 unit2 unit3 unit4 unit5
## unit1     1     1     0     0     0
## unit2     1     1     0     0     0
## unit3     0     0     0     0     0
```

```
## unit4     0     0     0     0     0
## unit5     0     0     0     0     0
##
## , , S2
##
##       unit1 unit2 unit3 unit4 unit5
## unit1     1     0     1     0     0
## unit2     0     0     0     0     0
## unit3     1     0     1     0     0
## unit4     0     0     0     0     0
## unit5     0     0     0     0     0
##
## , , S3
##
##       unit1 unit2 unit3 unit4 unit5
## unit1     1     0     0     1     0
## unit2     0     0     0     0     0
## unit3     0     0     0     0     0
## unit4     1     0     0     1     0
## unit5     0     0     0     0     0
```

Now for each pair, add up $p(\mathcal{S})$ for each sample they appear together in and that will yield π_{uv}:

```
pij2 = apply( jsample.incl, c(1,2), weighted.sum, w=d2)
pij2
```

Weighted sum through the tensor

```
##       unit1 unit2 unit3 unit4 unit5
## unit1   0.6   0.0   0.1   0.2   0.3
## unit2   0.0   0.3   0.0   0.1   0.2
## unit3   0.1   0.0   0.2   0.0   0.1
## unit4   0.2   0.1   0.0   0.3   0.0
## unit5   0.3   0.2   0.1   0.0   0.6
```

π_{12} π_{51} marginal inclusion probabilities

Now let's use the Horvitz-Thompson estimate

- Recall we had samples of size $n = 2$ and each sample was assigned a different probability $p(\mathcal{S})$. The sample estimate of $a(\mathcal{P}) = \sum_{u \in \mathcal{P}} \frac{y_u}{5}$ was defined as:

N=5

$$\textit{estimated average} = \sum_{u \in \mathcal{S}} \frac{y_u}{2}$$ ← previous naive estimate

  - Recall also that the corresponding estimator was biased.
- We can use the inclusion probabilities to make our estimator unbiased:

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

Here are the HT estimates of the population mean in each of the $\binom{N}{n} = 10$ possible samples:

```
sam.HT = apply(sam2, 2, function(s, x, wt){ sum(x[s]/wt[s]) }, x=pop5, wt=pi2 )/5
sam.HT
```

```
##          S1          S2          S3          S4          S5          S6
```

```
## -1.01333333 -0.66333333  0.20000000 -0.06333333 -0.97000000 -0.10666667
##          S7          S8          S9         S10
## -0.37000000  0.24333333 -0.02000000  0.84333333
```

HT estimates of a(P) in all possible samples

The sampling bias is then $\sum_{\mathcal{S}\in\mathcal{P}_\mathcal{S}} \left(a_{HT}(\mathcal{S}) - a(\mathcal{P})\right) p(\mathcal{S})$:

```
sum((sam.HT - mean(pop5))*d2)
```

```
## [1] -1.474515e-17
```

= 0

The sampling variance is then $\sum_{\mathcal{S}\in\mathcal{P}_\mathcal{S}} \left(a_{HT}(\mathcal{S}) - E[a_{HT}(\mathcal{S})]\right)^2 p(\mathcal{S})$:

```
sum((sam.HT - sum(sam.HT*d2))^2*d2)
```

```
## [1] 0.06433822
```

Exercise: Using the joint inclusion probabilities, calculate this sampling variance using this formula instead:

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

A comparison across sampling designs and methods of estimation is shown below.

- **Design 1**
  - Simple random sampling without replacement $p(\mathcal{S}) = 0.1 \quad \forall\mathcal{S}$
  - Estimate:

$$\begin{aligned} \widehat{a}(\mathcal{P}) &= \sum_{u\in\mathcal{S}} \frac{y_u}{2} = a(\mathcal{S}) \\ &= \frac{1}{5}\sum_{u\in\mathcal{S}} \frac{y_u}{\pi_u} = a_{HT}(\mathcal{S}) \end{aligned}$$

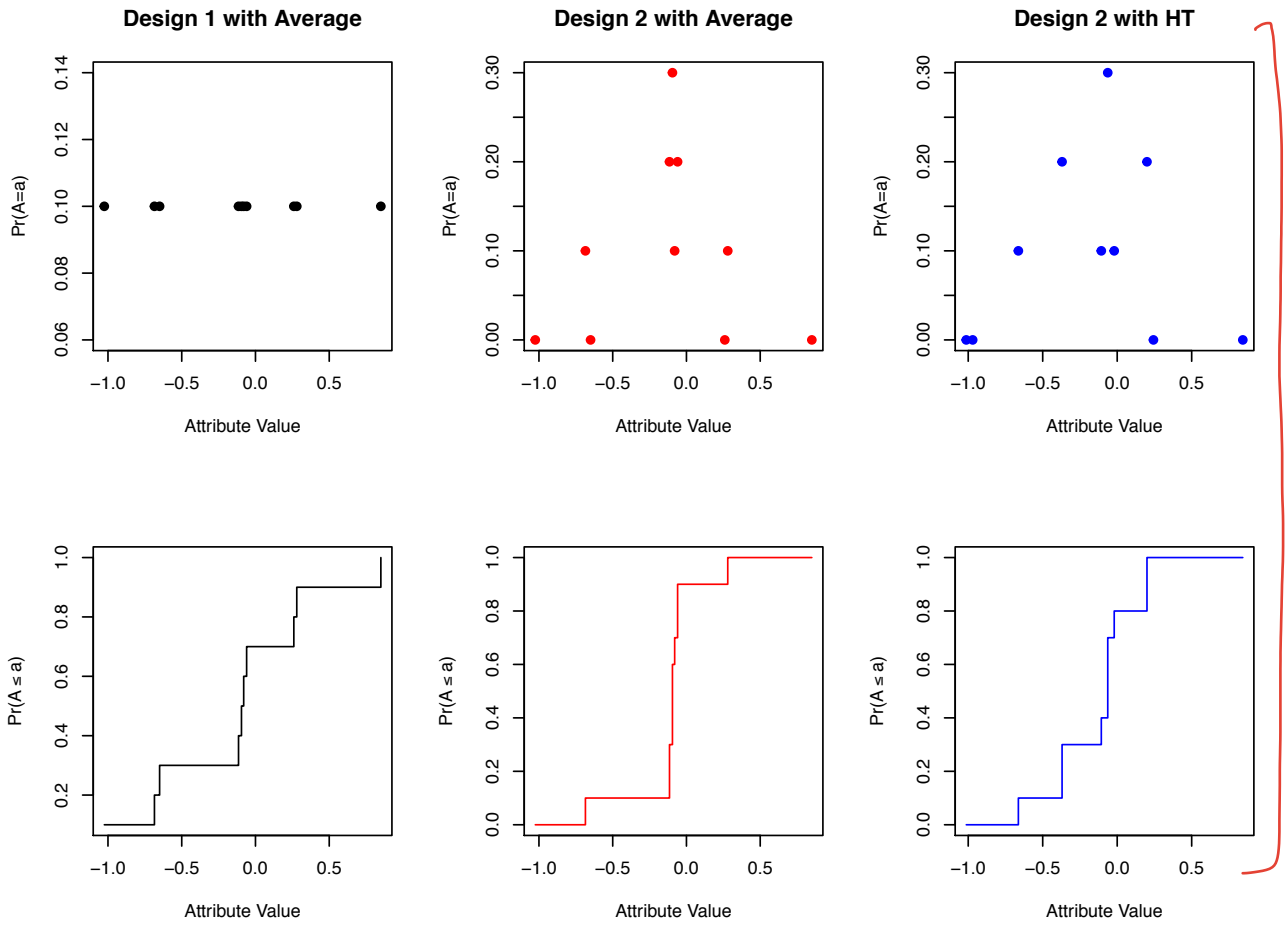
$\pi_u = \frac{n}{N} = \frac{2}{5}$

- **Design 2**
  - Sampling design with $\{p(\mathcal{S}_1), p(\mathcal{S}_2), \ldots, p(\mathcal{S}_{10})\} = \{0.0, 0.1, 0.2, 0.3, 0.0, 0.1, 0.2, 0.0, 0.1, 0.0\}$
  - Estimate:

$$\widehat{a}(\mathcal{P}) = \sum_{u\in\mathcal{S}} \frac{y_u}{2} = a(\mathcal{S})$$

- **Design 2***
  - Sampling design with $\{p(\mathcal{S}_1), p(\mathcal{S}_2), \ldots, p(\mathcal{S}_{10})\} = \{0.0, 0.1, 0.2, 0.3, 0.0, 0.1, 0.2, 0.0, 0.1, 0.0\}$
  - Estimate:

$$\widehat{a}(\mathcal{P}) = \frac{1}{5}\sum_{u\in\mathcal{S}} \frac{y_u}{\pi_u} = a_{HT}(\mathcal{S})$$



Visual depictions of the sampling distribution of $\hat{a}(P)$

```
##                   bias samp.var    MSE
## Design 1, Averge 0.00   0.2669 0.2669
## Design 2, Averge 0.02   0.0489 0.0493
## Design 2, HT     0.00   0.0643 0.0643
```

unbiased, but variance is large

biased, but variance is small

unbiased, variance is much smaller than in d1 but slightly larger than in d2

Strictly speaking (in terms of MSE) d2 is best, but d2* is unbiased and the MSE is very close to that of d2 and so may be a better alternative in practice.

HT Shark Example

Here we will explore the use of the HT estimator in the context of the Shark data. To begin with it will be useful to define the sampling mechanism as well as the marginal and joint inclusion probabilities.

- We will using sampling without replacement and hence use the `srswor` function already defined. Recall that this takes a sample of size n from the Sharks population.

```
set.seed(341)
sample_indx <- as.numeric(srswor(10))
sharkS <- sharks[sample_indx,]
```

n = 10

- Recall the marginal inclusion probability π_u when sampling without replacement is n/N for all u:

```
N <- nrow(sharks)
n <- 10
```

N = 65

```
pi <- rep(n/N, N)
```

- Recall also that the joint inclusion probability π_{uv} when sampling without replacement is $\frac{n(n-1)}{N(N-1)}$ for any pair (u, v):

```
pij <- matrix((n*(n-1)) / (N*(N-1)), nrow=N, ncol=N)
diag(pij) <- pi
```

What follows are a variety of attributes for which the HT estimate is calculated. In each case the left number is the HT estimate of the attribute and the right number is the true population value of the attribute.

- **Total** shark length:

```
y_u <- sharkS$Length
pi_u <- pi[sample_indx]
c(sum(y_u/pi_u), sum(sharks$Length))
```

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

- **Average** shark length:

```
y_u <- sharkS$Length/N
pi_u <- pi[sample_indx]
c(sum(y_u/pi_u), sum(sharks$Length/N))
```

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

- **Average** victim age:

```
y_u <- sharkS$Age/N
pi_u <- pi[sample_indx]
c(sum(y_u/pi_u), sum(sharks$Age/N))
```

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

- **Proportion** of shark encounters from Australia:

```
y_u <- sharkS$Australia/N
pi_u <- pi[sample_indx]
c(sum(y_u/pi_u), sum(sharks$Australia/N))
```

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

- **Proportion** of shark encounters ending in a fatality:

```
y_u <- sharkS$Fatality/N
pi_u <- pi[sample_indx]
c(sum(y_u/pi_u), sum(sharks$Fatality/N))
```

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

- **Proportion** of sharks with length less than or equal to 180 inches:

```
newVariate <- (sharks$Length <= 180)*1
y_u <- newVariate[sample_indx]/N
pi_u <- pi[sample_indx]
c(sum(y_u/pi_u), sum(newVariate/N) )
```

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

In order to quantify the **sampling variability** of these estimators we can use the HT estimate of the variance. In what follows we calculate the estimate of the variance and the estimate of the standard deviation (also called the **standard error**) for each of the HT estimates calculated above.

But first, the following function will be helpful:

```
estVarHT <- function(sam, yu, pi, pij){
  pi = pi[sam]
  pij = pij[sam,sam]
  delta = pij - outer(pi, pi)
  estimateVar =  sum( (delta/pij) * outer(yu/pi,yu/pi) )
  return(estimateVar)
}
```

In each case below the left number is the estimate of the variance of the HT estimator and the right number is the estimate of the standard deviation of the HT estimator (the standard error).

- **Total** shark length:

```
y_u <- sharkS$Length
v <- estVarHT(sam = sample_indx, y_u, pi, pij)
c(v, sqrt(v))
```

```
## [1] 493862.4167    702.7535
```

- **Average** shark length:

```
y_u <- sharkS$Length/N
v <- estVarHT(sam = sample_indx, y_u, pi, pij)
c(v, sqrt(v))
```

```
## [1] 116.89051  10.81159
```

- **Average** victim age:

```
y_u <- sharkS$Age/N
v <- estVarHT(sam = sample_indx, y_u, pi, pij)
c(v, sqrt(v))
```

```
## [1] 12.406496  3.522286
```

- **Proportion** of shark encounters from Australia:

```
y_u <- sharkS$Australia/N
v <- estVarHT(sam = sample_indx, y_u, pi, pij)
c(v, sqrt(v))
```

```
## [1] 0.01974359 0.14051188
```

- **Proportion** of shark encounters ending in a fatality:

```
y_u <- sharkS$Fatality/N
v <- estVarHT(sam = sample_indx, y_u, pi, pij)
c(v, sqrt(v))
```

```
## [1] 0.008461538 0.091986621
```

- **Proportion** of sharks with length less than or equal to 180 inches:

```r
y_u <- newVariate[sample_indx]/N
v <- estVarHT(sam = sample_indx, y_u, pi, pij)
c(v, sqrt(v))
```

```
## [1] 0.008461538 0.091986621
```

Note that calculations like the ones we've just done may also be automated using the series of general purpose functions defined in Section 3.2.5 of this version of the STAT 341 notes.

hyperlink

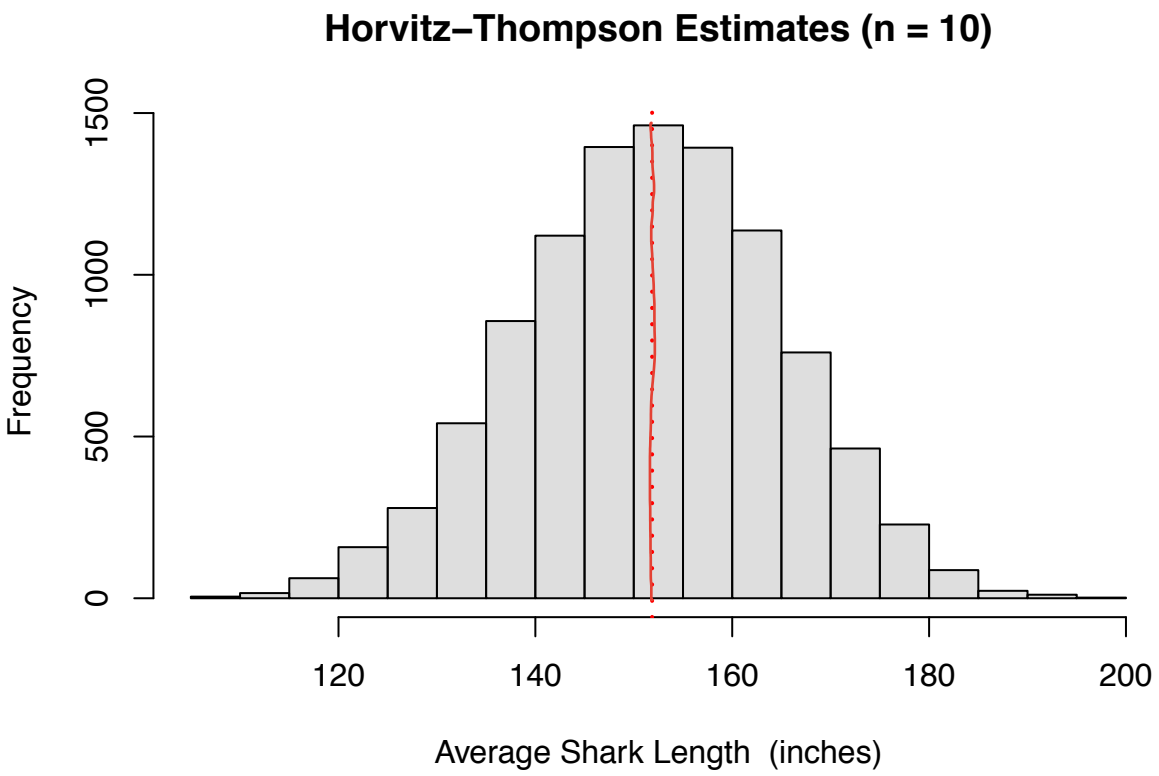
Such measures of variability are most commonly used in the context of **interval estimates** (e.g., **confidence intervals**) such as:

$$\left[a_{HT}(\mathcal{S}) - 2\widehat{SD}[\widetilde{a}_{HT}(\mathcal{S})], a_{HT}(\mathcal{S}) + 2\widehat{SD}[\widetilde{a}_{HT}(\mathcal{S})]\right]$$

Let's illustrate the intuition behind this formula, and let's do so in the context of the sampling distribution for the average shark length.

```r
### This is a new function modifed from createvariateFn
createvariateFnN <- function(popData, variate, N=1) {
  function (u){popData[u, variate]/N}
}
inclusionProb <- createInclusionProbFn(1:N, sampSize = n)
sharksHTestimator <- createHTestimator(inclusionProb)

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)
avgs <- as.numeric(avgs)

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

10,000 samples



- If we use sampling without replacement and a sample size of size $n = 10$, how close is the sample estimate to the population value?

  We're on average close to the true value, but can be wrong in either direction by ~50 inches.

- To quantify this we might report some measure of dispersion for the sampling distribution, such as:
  - The interquartile range is 18.3
  - The 25^{th} and 75^{th} quantiles are (142.6, 160.9)
  - The 25^{th} and 75^{th} quantiles are the endpoints of an interval containing 50% of the HT estimates

$$\Pr\left(\widetilde{a}(\mathcal{S}) \in [Q(0.25), Q(0.75)]\right) = 0.5$$

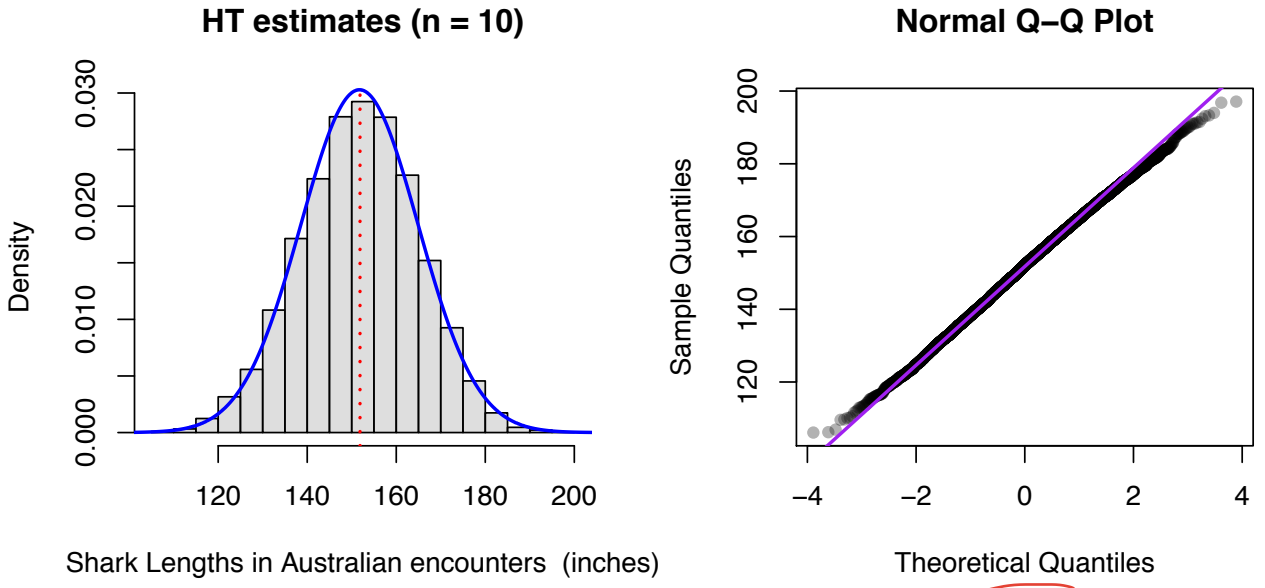
- We might also cast a wider net and use the trimmed range where we remove the smallest and largest 2.5% (for example):
  - This trimmed range is $Q(0.975) - Q(0.025) = 51.3025$
  - The interval $(Q(0.025), Q(0.975)) = (125.2, 176.5025)$ contains 95% of the HT estimates

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

BUT in order to calculate such intervals we need a reasonably good estimate of the sampling distribution, which above was obtained by drawing 10,000 samples of size $n = 10$ from the population. This is not realistic. Instead we could recognize that the sampling distribution we obtained when we did this appears bell-shaped and symmeteric. In other words, it could be **approximated by a normal distribution**.

Letting μ equal the average 151.74145 and σ equal the standard deviation 13.1722378 from the population of averages as parameters in the $N(\mu, \sigma^2)$ distribution we can approximate the sampling distribution of the HT estimator as depicted below.

```r
### This is a new function modifed from createvariateFn
par(mfrow=c(1,2),oma=c(0,0,0,0))
hist(avgs, col=adjustcolor("grey", alpha = 0.5),
     main="HT estimates (n = 10)",
     xlab="Shark Lengths in Australian encounters  (inches)",
     breaks=25, prob=TRUE
     )
### Mark the population attribute in red
abline(v=popAvg, col="red", lty=3, lwd=2)

### Add the approximating normal curve
xseq= seq(10, 250, .01)
lines(xseq, dnorm(xseq, mean(avgs), sd(avgs)), col=4, lwd = 2)


### Compare the observed distribution to the approximating normal distributions

qqnorm(avgs, pch = 16, col = adjustcolor("black", alpha = 0.3))
qqline(avgs, col = "purple", lwd = 2)
```



- As was determined using the quantiles from the samples, the interval (125.2, 176.5) contains 95% percent of the HT estimates.
- Since the histogram is approximately normal, we have that $\tilde{a}_{HT}(S) \sim N\left(E[\tilde{a}_{HT}(S)], Var[\tilde{a}_{HT}(S)]\right)$

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

  - Applying this result yields the following approximate interval (125.9, 177.6)

- Under an assumption of normality, an interval that contains 95% percent of the distribution's values is

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

where $Z_{1-0.05/2} = 1.959964 \approx 2$

- Letting μ equal $a(\mathcal{P})$ (the expecation of the unbiased HT estimator)

$$\mu = a(\mathcal{P}) = \mathrm{E}[\widetilde{a}_{HT}(\mathcal{S})]$$

and σ be the standard deviation of the HT estimator

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

we obtain the following interval which (under an assumption of normality) approximately contains 95% of the HT estimates

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

- A sample estimate of this interval is

$$a_{HT}(\mathcal{S}) \pm 2\sqrt{\widehat{\mathrm{Var}}\left[\widetilde{a}_{HT}(\mathcal{S})\right]} = a_{HT}(\mathcal{S}) \pm 2\widehat{\mathrm{SD}}\left[\widetilde{a}_{HT}(\mathcal{S})\right]$$

This interval calculated for the average shark length based on our SRSWOR of size $n = 10$ is

```
y_u <- sharkS$Length/N
pi_u <- pi[sample_indx]
estHT <- sum(y_u/pi_u)
sterrorHT <- sqrt(estVarHT(sam = sample_indx, y_u, pi, pij))
estHT + 2*c(-1,1)*sterrorHT
```

```
## [1] 109.2768 152.5232
```

Exercise: Calculate this interval for each of the other HT estimates investigated in this example.

Exercise: For each such HT estimate generate 10,000 samples and for each $\mathcal{S}$ determine the interval

$$a_{HT}(\mathcal{S}) \pm 2\widehat{\mathrm{SD}}\left[\widetilde{a}_{HT}(\mathcal{S})\right]$$

and determine the proportion of these intevals that contain $a(\mathcal{P})$. Comment on your findings.

3.2.6 Sampling Design

The pair $(\mathcal{P}_{\mathcal{S}}, p(\mathcal{S}))$ is called a **sampling design**

- Together they determine which samples are possible and with what probability they are selected
- The SRSWOR, SRSWR amd SRSWH frameworks provide examples of different sampling designs
- The sampling design is ours to choose
  - We may choose $\mathcal{P}_{\mathcal{S}}$ so that the values $a(\mathcal{S})$ for $\mathcal{S} \in \mathcal{P}_{\mathcal{S}}$ are constrained to be near $a(\mathcal{P})$
  - We may choose $p(\mathcal{S})$ so that samples $\mathcal{S} \in \mathcal{P}_{\mathcal{S}}$ that have $a(\mathcal{S})$ close to $a(\mathcal{P})$ have higher probability $p(\mathcal{S})$ of being selected

– Within the Horvitz-Thompson framework we know that

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

which provides insight into how we might best choose a sampling design.

* For example, if we could choose $\pi_u \propto y_u$ then the variance (and MSE) will be zero!
* Perhaps there is another variate x_u that is highly positively correlated with y_u for all $u \in \mathcal{P}$. Then choosing $\pi_u \propto x_u$ could reduce MSE.
* If we knew when $y_u \approx y_v$ we could choose $\pi_u \approx \pi_v$ and this might reduce MSE (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 of an estimator (attribute) of interest.