

Question 2 c)

Sheen Thusoo

14/11/2021

```
data <- read.csv("Infectious.csv")

# From Tutorial 7 but modified for with replacement and weighted
createInclusionProbFn <- function(N, sampSize, w) {
  n <- sampSize
  f <- function(u) rep( 1 - (1 - w[u])^n, length(u))
  return(f)
}

createJointInclusionProbFn <- function(N, sampSize, w) {
  n <- sampSize
  f <- function(u, v) ifelse(u == v, 1 - (1 - w[u])^n, 1 - (1 - w[u])^n -
                             (1 - w[v])^n + (1 - w[u] - w[v])^n )
  return(f)
}

createHTestimator <- function(pi_u_fn) {
  f <- function(sample_idx, variateFn)
    sum(sapply(sample_idx, function(u) variateFn(u)/ pi_u_fn(u)))
  return(f)
}

createHTVarianceEstimator <- function(pi_u_fn, pi_uv_fn) {
  f = function(sample_idx, variateFn) {
    sum(outer(sample_idx, sample_idx, Vectorize(function(u, v) {
      pi_u <- pi_u_fn(u)
      pi_v <- pi_u_fn(v)
      y_u <- variateFn(u)
      y_v <- variateFn(v)
      pi_uv <- pi_uv_fn(u, v)
      Delta_uv <- pi_uv - pi_u * pi_v
      return((Delta_uv * y_u * y_v) / (pi_uv * pi_u * pi_v))
    })))
  }
  return(f)
}

createGenericVariateFn <- function(popData, expression, ...) {
  # Save extra arguments to extra_args
  extra_args <- list(...)
  # A formality; instead of evaluating, return the unevaluated expression.
```

```

evalable <- substitute(expression)
# Evaluate expression in the context of popData, restricted to indices u, and any extra_args.
f <- function(u) with(extra_args, eval(evalable, popData[u,]))
return(f)
}

```

```

w = c(0.0027, 0.016, 0.0069, 0.0011, 0.0066, 0.0108, 0.003, 0.0043,
      0.0142, 0.0016, 0.0122, 4e-04, 0.0047, 0.014, 0.0086, 0.0169,
      0.0165, 0.0118, 0.0043, 2e-04, 0.0142, 0.0092, 0.0162, 0.0106,
      0.0588, 0.0135, 0.0025, 0.0011, 0.0109, 0.0085, 0.0027, 0.0112,
      0.0127, 5e-04, 0.0082, 0.0085, 0.0066, 0.0125, 0.012, 0.0116,
      0.0089, 0.016, 0.0108, 9e-04, 0.0088, 0.0066, 0.0588, 0.0044,
      8e-04, 0.002, 0.01, 0.0101, 0.0012, 0.0135, 0.0103, 0.0058,
      0.004, 0.0088, 0.0057, 0.0049, 0.0111, 0.0117, 0.0081, 0.014,
      0.0079, 0.0134, 0.0149, 0.0042, 0.0109, 0.0072, 0.0109, 0.0082,
      5e-04, 0.002, 0.0588, 0.0025, 0.0018, 0.0105, 0.015, 0.0148,
      0.0042, 0.0025, 0.0061, 0.0111, 0.017, 0.015, 0.0056, 0.0011,
      0.0072, 0.007, 0.015, 0.0081, 0.016, 0.0057, 0.0029, 0.0012,
      0.0588, 8e-04, 2e-04, 0.005)

```

Provide an estimate of the total number of individuals who died in country A because of the infectious disease

```

n = 100 # samples
N = 486 # population

inclusionProb      <- createInclusionProbFn(N, sampSize = n, w=w)
inclusionJointProb <- createJointInclusionProbFn(N, sampSize = n, w=w)

deceasedHTEstimator <- createHTEstimator(inclusionProb)
HTVarianceEstimator <- createHTVarianceEstimator(pi_u_fn = inclusionProb,
                                                  pi_uv_fn = inclusionJointProb)

deceased.infected <- createGenericVariateFn(data, Deceased.Prop * Infected * 12.5)

estimated.deceased.infected <- deceasedHTEstimator(seq(100), deceased.infected)
cat('Estimate of total number of individuals who died in country A because
    of the infectious disease: ', estimated.deceased.infected)

## Estimate of total number of individuals who died in country A because
## of the infectious disease: 92713.64

```

Provide the standard error of your estimate as well as a 95% confidence interval for the number of individuals who died of the infectious

```

standard.error <- sqrt(HTVarianceEstimator(seq(100), deceased.infected))
cat('Standard Error of Estimate: ', standard.error)

## Standard Error of Estimate: 13230.55

```

```
confidence.interval <- estimated.deceased.infected + 2*standard.error*c(-1,1)
cat('95% Confidence Interval: ', confidence.interval)
```

```
## 95% Confidence Interval: 66252.53 119174.7
```