

Initial Set Up, a) to e)

```
# set up
N = 100000
Yvalues = rgamma(100000, 20, 0.5)

# summary
true.t_y = sum(Yvalues)

# poisson sampling probabilities
P = rep(.5, N)
P[Yvalues <= 50] = 0.3
P[Yvalues <= 30] = 0.1

# repetitions
T = 5000
data = matrix(NA, nrow = T, ncol = 2)
colnames(data) = c('SRS', 't_y.HT')

for (i in seq(1:T)) {

  # sampling
  insample = rbinom(N, 1, P)
  thesampley = Yvalues[insample == 1]
  thesamplep = P[insample == 1]

  # calculate estimators
  t_y.HT = sum(thesampley / thesamplep)
  SRS_estimator = N * sum(thesampley/length(thesampley))

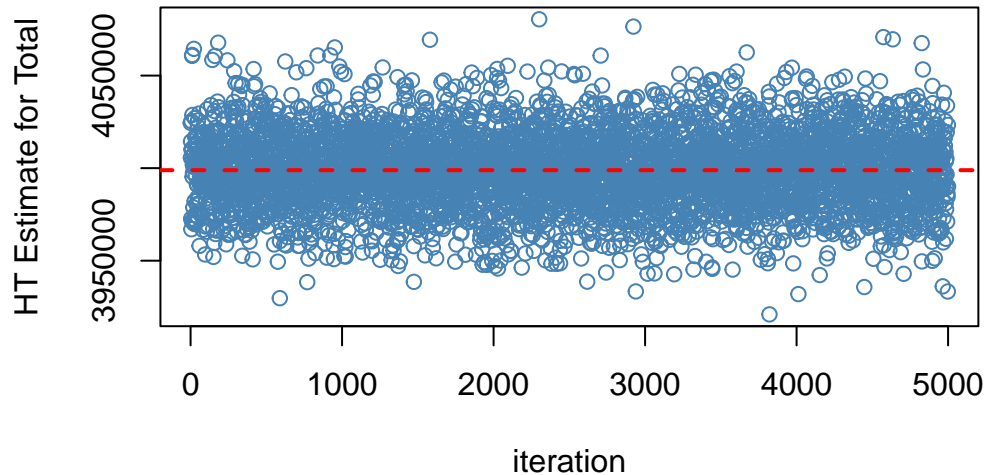
  # store results
  data[i,] = c(SRS_estimator, t_y.HT)
}
```

f)

Yes, the HT estimator is unbiased. We know from the theory that it is, so we just have to run the code to support that. I have `true.t_y` as the truth from the simulated population, and below I've plotted the individual iterations against each iteration's \hat{t}_y^{HT} with a horizontal red line for the true t_y .

This line passes right through the middle of the data points, so we know this supports that $E(\hat{t}_y^{HT}) - t_y = 0$.

```
plot(seq(1:T), data[, 't_y.HT'],
     col='steelblue',
     xlab = 'iteration',
     ylab = 'HT Estimate for Total')
abline(h = true.t_y, col='red', lty=2, lwd=2)
```



g)

Since we're Poisson sampling instead of an SRS, $N\bar{y}$ is not unbiased. By definition, we should see (approximately) that $E(N\bar{y}) - t_y = 0$, but below we see that is not the case. However, as a double check on the Horwitz-Thompson estimator being unbiased, we see that this difference is close to 0 (though is not 0 since we're performing a Monte Carlo calculation).

I didn't plot this since the scale is off and the line does not pass anywhere near the data points, additionally confirming that $N\bar{y}$ is biased.

```
mean(data[, 't_y.HT']) - true.t_y
```

```
[1] -224.0549
```

```
mean(data[, 'SRS']) - true.t_y
```

```
[1] 248740.4
```