STAT 5201 Final Exam Tae Uk You

1. Find a recent survey reported in a newspaper, magazine or on the web. Briefly describe the survey. What are the target population and sampled population? What conclusions are drawn from the survey in the article. Do you think these conclusions are justified? What are the possible sources of bias in the survey? Please be specific but brief.

(Answer)

President Trump's tweets are often seen in many mainstream and social media these days. Using Twitter as a tool to speak directly to American people and his supporters appears to be effective for Trump; his Twitter account (@realDonaldTrump) is followed by 52 million people, which is not a small number when comparing to his 63 million votes he received in 2016 President Election.

However, there was a recent study that measured how many Americans of voting age actually read Trump's Tweets. In May 2018, Gallup U.S. Poll conducted telephone interviews with a random sample of 2,806 adults, aged 18 and older, living in all 50 U.S. states and the District of Columbia. Each sample of national adults included a minimum quota of 70% cellphone responders and 30% landline respondents. Telephone numbers were selected using random-digit-dial methods. Target population was all U.S. adults who age 18 and older, and sampled population were randomly picked U.S. adults through randomly-dialed telephone numbers.

After the sampling survey, the study introduced conclusions that estimate among all U.S. adults population that 26% Americans have a Twitter account, 8% have a Twitter account and follow Trump's Twitter account, and 4% have a Twitter account, follow his account, and read all or most Trump's tweets. So based on this Gallup's data, there are 20 millions of Americans of voting age follow Trump's Twitter account, and the real consumers of his tweets are about 10 millions.

The report concluded that 20 million and 10 million are not small numbers at all, but it is nowhere close to the 52 million followers of his Twitter account. The study agreed that Trump's unprecedented use of Twitter as a means of presidential communication serves him well as a mean to get his thoughts and messages out to a broad majority of the American public. However, the study indicated that his tweets do not appear to follow a direct pathway to most Americans by subscribing to his Twitter feed. Instead, the study analyzed that Trump's tweet reach Americans because news and social media propagate and rebroadcast them into news and social media streams.

Of course, this Gallup poll is just one survey, but its conclusion has more realistic estimate of Trump's Twitter audience than his official number of followers. People of all ages and even outside of the U.S. can use Twitter so Trump's 52 million followers may include teenagers and foreigners who care about his thoughts and decisions. And the result of this survey that 26 percent of American adults follow Twitter accounts has similar estimate of other same-purposed surveys, like 2016 Pew Research Center poll that found 21 percent of American adults have Twitter accounts.

Despite decent procedure and conclusion of this survey, a few possible limitations are found. One of the questions that the survey asked was which party—Republicans and Democrats—are you for and how much do you follow President Trump's tweets. And the result showed that Democrats pay more attention than Republicans, 64% and 50% respectively. However, Republicans may voluntarily respond that they do not care his tweets as most of developed stories and coverage about his remarks in Twitter tend to be negative. Democrats, on the other hand, may voluntarily respond that they pay attention to them because they may be more exposed to negative attention about his tweets. Therefore, this particular question and conclusion cannot be considered valid or scientific as the responses were susceptible to being biased.

2. Information for a population with four strata is given just below. The second column of the table gives the sizes of the four strata, the next is our guess for the variance for each strata and the last, the cost per sampling an observation in each strata.

stratum	size	Est var	cost
1	2000	450	10
2	1500	300	15
3	1000	500	20
4	1200	200	10

1) Using this information, find the optimal allocation of a sample of size 100.

```
N = c(2000,1500,1000,1200)
v = c(450, 300, 500, 200)
c = c(10, 15, 20, 10)

dummy = sum(N*sqrt(v))

optall = function(N, v){
  size_var = N*sqrt(v)
  n = N*sqrt(v) / dummy
  return(n*100)
}

for(i in 1:4){
  n = optall(N[i], v[i])
  cat("The stratum", i, " will get ", round(n),  "\n")
}


Results:
The stratum 1 will have 39 sample size.
The stratum 2 will have 24 sample size.
The stratum 3 will have 21 sample size.
The stratum 4 will have 16 sample size.
```

2) Suppose we want an estimate of the population mean with a variance of approximately 10. Based on the information in the table what would be the optimal allocation.

```
findcost = function(N, v, c, fix_v){
  w = N/sum(N)
  dum1 = sum(w*sqrt(v)*sqrt(c))
  dum2 = sum((w^2)*v/N)
  lam = dum1/(fix_v + dum2)
  cost = (dum1^2) / (fix_v + dum2)
  n = lam*w*sqrt(v)/sqrt(c)
  ans = c(cost,n)
  return(ans)
}

answer = findcost(N, v, c, fix_v=10)

> answer
[1] 461.468405  15.938227   7.969113   5.939826   6.375291


Results:
The stratum 1 will cost 440.1 to allocate 16.
The stratum 2 will cost 441.18 to allocate 8.
The stratum 3 will cost 952.38 to allocate 6.
The stratum 4 will cost 196.72 to allocate 7.
```

3. From a population of 20 clusters a simple random sample of 4 clusters was taken. Within in each sampled cluster a further random sample was taken. The data are in a matrix denoted by mxclus. The cluster sizes are only known for the clusters in the sample. Find the value of the ratio estimator for estimating the population mean and give an estimate of its variance.

```
library("RCurl")
load(url("http://users.stat.umn.edu/~gmeeden/classes/5201/moredata/clusf18.RData"))
mxclus

yval = mxclus[,3]
size = mxclus[,2]
clus = mxclus[,1]
foo = split(yval,clus)

nclus = length(foo)
nclus

dum = split(size,clus)
mi = sapply(dum,length)

Mi = sapply(dum,mean)

ssufpc = 1-mi/Mi
clusmean = sapply(foo,mean)
clusvar = sapply(foo,var)
N = 20 #first-stage cluster size
that = (N/nclus)*sum(Mi*clusmean)
ybarratio = sum(Mi*clusmean)/sum(Mi)
s2t = var(Mi*clusmean)
varterm2 = sum(ssufpc*(Mi^2)*clusvar/mi)
s2r = var(Mi*(clusmean - ybarratio))
varthat = N^2*(1-nclus/N)*s2t/nclus + (N/nclus)*varterm2
varybarr = ( (1-nclus/N)*s2r/nclus + varterm2/(nclus*N))/(mean(Mi))^2
list(that=that, varthat=varthat, yratio=ybarratio,  varyratio=varybarr)
```

```
Results:
$that
[1] 7562.099

$varthat
[1] 3599200

$yratio
[1] 25.207

$varyratio
[1] 19.51253
```

Comment:
From the above R code, I computed to estimate the population mean (=25.207) and its estimate variance (=19.51253) using ratio estimator in two-stage cluster sample settings.

4. Consider the following table of sums of weights from a sample; each entry in the table is the sum of the sampling weights for persons in the sample falling in that classification (for example, the sum of the sampling weights for the number of women between the ages of 20 and 29 is 98.) Assume it is known the that the population contains 1800 men and 1200 women and 300 persons between the ages of 20-29, 900 between 30-39, 1000 between 40-49 and 800 between 50-59. Readjust the cell weights so that in the new table the marginal weights agree with the known marginal population weights.

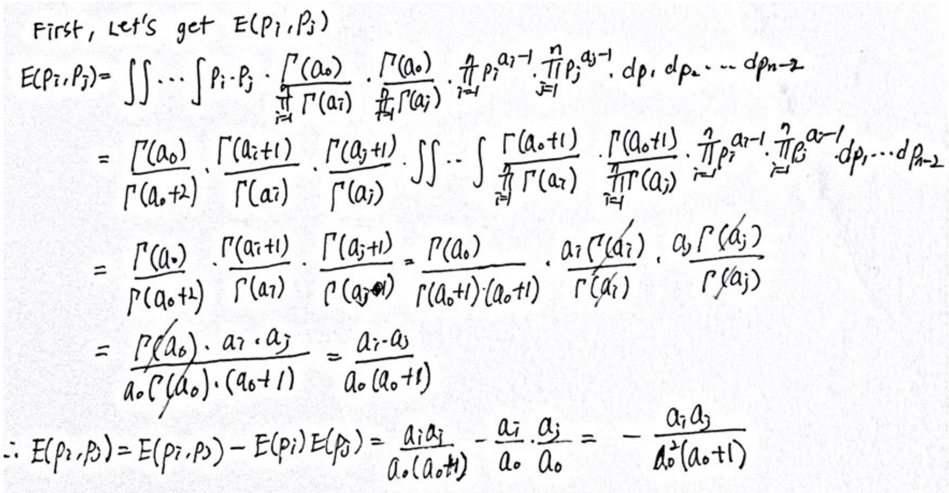
```
male =c(183,447,522,416);female =c(98,377,395,467);truemale = 1800;truefemale = 1200;trueage
=c(300,900,1000,800)

rbind(male,female)
       [,1] [,2] [,3] [,4]
male    183  447  522  416
female   98  377  395  467

# First Trial (raking row) => result: not age group yet
male1 = male*(truemale/sum(male)
female1 = female*(truefemale/sum(female))
rbind(male1,female1)
             [,1]     [,2]     [,3]     [,4]
male1   210.07653 513.1378 599.2347 477.5510
female1  87.95812 338.3695 354.5251 419.1473

# Second Trial (raking columns) => result: make the gender total off again
male2 = numeric(4)
for(i in 1:4){
  male2[i] = male1[i] * (trueage[i]/sum(male1[i]+female1[i]))
}
female2 = numeric(4)
for(i in 1:4){
  female2[i] = female1[i] * (trueage[i]/sum(male1[i]+female1[i]))
}
rbind(male2,female2)
male2   211.46186 542.3606 628.2868 426.0528
female2  88.53814 357.6394 371.7132 373.9472

# Third Trial (raking row) => result: make the age off a bit again, but noticing it's getting close
male3 = male2*(truemale/sum(male2))
female3 = female2*(truefemale/sum(female2))
rbind(male3,female3)
> rbind(male3,female3)
             [,1]     [,2]     [,3]     [,4]
male3   210.50732 539.9124 625.4507 424.1296
female3  89.14448 360.0886 374.2588 376.5081

#Fourth Trial (raking columns) => result: It's getting close but still needs to rake the row again
male4 = numeric(4)
for(i in 1:4){
  male4[i] = male3[i] * (trueage[i]/sum(male3[i]+female3[i]))
}
female4 = numeric(4)
for(i in 1:4){
  female4[i] = female3[i] * (trueage[i]/sum(male3[i]+female3[i]))
}
rbind(male4,female4)
             [,1]     [,2]     [,3]     [,4]
male4   210.75193 539.9118 625.6325 423.7917
female4  89.24807 360.0882 374.3675 376.2083

#Fifth Trial (raking row) => result: the age groups are off just a bit.
male5 = male4*(truemale/sum(male4))
female5 = female4*(truefemale/sum(female4))
rbind(male5,female5)
             [,1]     [,2]     [,3]     [,4]
male5   210.74164 539.8854 625.6019 423.7710
female5  89.25461 360.1146 374.3950 376.2358

#Sixth Trial (raking columns) => result: very close, just a bit less than 1.5
male6 = numeric(4)
for(i in 1:4){
  male6[i] = male5[i] * (trueage[i]/sum(male5[i]+female5[i]))
}
female6 = numeric(4)
for(i in 1:4){
  female6[i] = female5[i] * (trueage[i]/sum(male5[i]+female5[i]))
}
rbind(male6,female6)
             [,1]     [,2]     [,3]     [,4]
male6   210.74427 539.8854 625.6039 423.7674
female6  89.25573 360.1146 374.3961 376.2326

> sum(male6)
[1] 1800
> sum(female6)
[1] 1200
> cbind(sum([,1]),sum([,2]),sum([,3]),sum([,4]))
[1] 300 900 1000 800
```

Comment:

Raking is a poststratification method that may be used when poststrata are formed using more than one variable, but only the marginal population totals are known. Now, we found that the entries in the last table may be better estimates of the cell populations than the original weighted estimates, simply because they use more information about the population.

5. Let $p = (p_1, \dots, p_n)$ be a probability vector. That is each $p_i > 0$ and they sum to one. Let Δ be the set of all such possible probability vectors. Let $\alpha = (\alpha_1, \dots, \alpha_n)$ where each $\alpha_i > 0$. We say that p has the Dirichlet distribution with parameter α if its probability density function over Δ is given by $f(p_1, \dots, p_n) = \frac{\Gamma(\sum_{i=1}^{n} \alpha_i)}{\prod_{i=1}^{n} \Gamma(\alpha_i)} \prod_{i=1}^{n} {p_i}^{\alpha_1 - 1}$. Let $\alpha_0 = \sum_{i=1}^{n} \alpha_i$. In class, it was stated that $E(p_i) = \alpha_i / \alpha_0$ and $Var(p_i) = \frac{\alpha_i(\alpha_0 - \alpha_i)}{{\alpha_i}^2(\alpha_0+1)}$.

1) Show that $cov(p_i, p_j) = -\frac{\alpha_i \alpha_j}{{\alpha_0}^2(\alpha_0+1)}$.



First, Let's get $E(p_i, p_j)$

$$E(p_i, p_j) = \int\int\cdots\int p_i \cdot p_j \cdot \frac{\Gamma(a_0)}{\prod_{i=1}^{n} \Gamma(a_i)} \cdot \frac{\Gamma(a_0)}{\prod_{j=1}^{n} \Gamma(a_j)} \cdot \prod_{i=1}^{n} {p_i}^{a_i-1} \cdot \prod_{j=1}^{n} {p_j}^{a_j-1} \cdot dp_1\, dp_2 \cdots dp_{n-2}$$

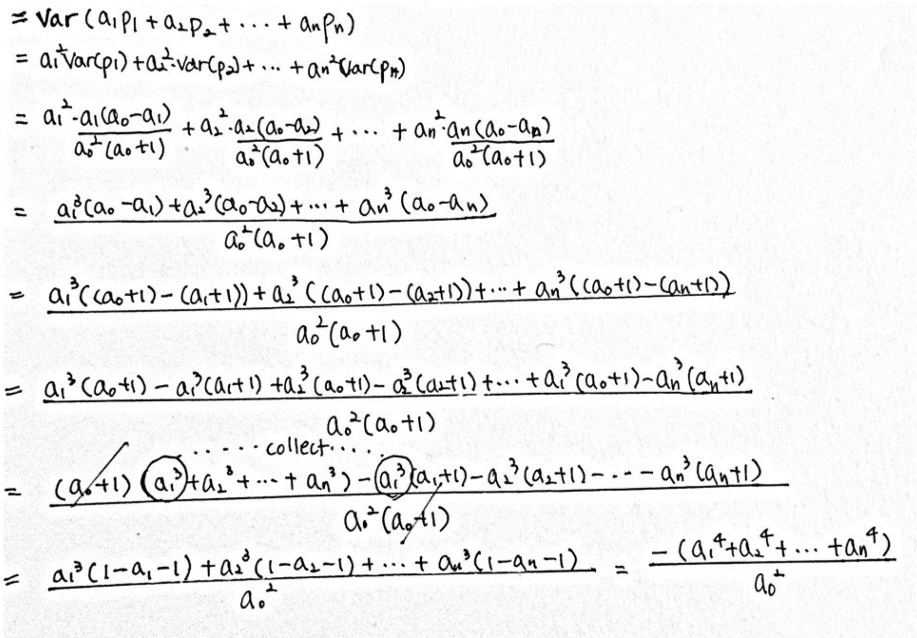
$$= \frac{\Gamma(a_0)}{\Gamma(a_0+2)} \cdot \frac{\Gamma(a_i+1)}{\Gamma(a_i)} \cdot \frac{\Gamma(a_j+1)}{\Gamma(a_j)} \cdot \int\int\cdots\int \frac{\Gamma(a_0+1)}{\prod_{i=1}^{n} \Gamma(a_i)} \cdot \frac{\Gamma(a_0+1)}{\prod_{i=1}^{n} \Gamma(a_j)} \cdot \prod_{i=1}^{n} {p_i}^{a_i-1} \cdot \prod_{i=1}^{n} {p_j}^{a_i-1} \cdot dp_1 \cdots dp_{n-2}$$

$$= \frac{\Gamma(a_0)}{\Gamma(a_0+2)} \cdot \frac{\Gamma(a_i+1)}{\Gamma(a_i)} \cdot \frac{\Gamma(a_j+1)}{\Gamma(a_j)} = \frac{\Gamma(a_0)}{\Gamma(a_0+1) \cdot (a_0+1)} \cdot \frac{a_i \cancel{\Gamma(a_i)}}{\cancel{\Gamma(a_i)}} \cdot \frac{a_j \cancel{\Gamma(a_j)}}{\cancel{\Gamma(a_j)}}$$

$$= \frac{\cancel{\Gamma(a_0)} \cdot a_i \cdot a_j}{a_0 \cancel{\Gamma(a_0)} \cdot (a_0+1)} = \frac{a_i \cdot a_j}{a_0(a_0+1)}$$

$$\therefore E(p_i, p_j) = E(p_i \cdot p_j) - E(p_i)E(p_j) = \frac{a_i a_j}{a_0(a_0+1)} - \frac{a_i}{a_0} \cdot \frac{a_j}{a_0} = -\frac{a_i a_j}{a_0^2(a_0+1)}$$

2) Find an expression for $Var(\sum_{i=1}^{n} \alpha_i\, p_i)$.



$$= Var(a_1 p_1 + a_2 p_2 + \cdots + a_n p_n)$$

$$= a_1^2 Var(p_1) + a_2^2 \cdot Var(p_2) + \cdots + a_n^2 Var(p_n)$$

$$= a_1^2 \cdot \frac{a_1(a_0 - a_1)}{a_0^2(a_0+1)} + a_2^2 \cdot \frac{a_2(a_0 - a_2)}{a_0^2(a_0+1)} + \cdots + a_n^2 \cdot \frac{a_n(a_0 - a_n)}{a_0^2(a_0+1)}$$

$$= \frac{a_1^3(a_0 - a_1) + a_2^3(a_0 - a_2) + \cdots + a_n^3(a_0 - a_n)}{a_0^2(a_0+1)}$$

$$= \frac{a_1^3((a_0+1) - (a_1+1)) + a_2^3((a_0+1) - (a_2+1)) + \cdots + a_n^3((a_0+1) - (a_n+1))}{a_0^2(a_0+1)}$$

$$= \frac{a_1^3(a_0+1) - a_1^3(a_1+1) + a_2^3(a_0+1) - a_2^3(a_2+1) + \cdots + a_i^3(a_0+1) - a_n^3(a_n+1)}{a_0^2(a_0+1)}$$

. . . . . collect . . . .

$$= \frac{(\cancel{a_0+1})(a_1^3 + a_2^3 + \cdots + a_n^3) - a_1^3(a_1+1) - a_2^3(a_2+1) - \cdots - a_n^3(a_n+1)}{a_0^2(\cancel{a_0+1})}$$

$$= \frac{a_1^3(1 - a_1 - 1) + a_2^3(1 - a_2 - 1) + \cdots + a_n^3(1 - a_n - 1)}{a_0^2} = \frac{-(a_1^4 + a_2^4 + \cdots + a_n^4)}{a_0^2}$$

6. In this problem you must perform a small simulation study to compare the ratio estimator and the regression estimator when the model generating the population is the correct one for the regression estimator. Use the following code to generate two different populations; the first with sd = 10 and the second with sd = 5. For each of the two populations take 400 simple random samples of size 10 and then of size 50. Compare both the point and interval estimators for the population total for the two estimators in the four cases. Discuss your results. Finally, construct a population such that when the sample size is 10 the average absolute error of the ratio estimator is at least for times larger that of the regression estimator.

```
set.seed(22334455)

# First, let's notice that popy1(sd=10) has bigger variability and smaller coefficient correlation.
cor(popx,popy1) # 0.3810313
cor(popx,popy2) # 0.6291245

# Simulation Case 1: sample size is 10
    ratio_size10_sd10 = matrix(data=NA, nrow=400, ncol=5)   ## ratio estimator, n=10, sd=10
    reg_size10_sd10 = matrix(data=NA, nrow=400, ncol=5)     ## reg estimator, n=10, sd=10
    ratio_size10_sd5 = matrix(data=NA, nrow=400, ncol=5)    ## ratio estimator, n=10, sd=5
    reg_size10_sd5 = matrix(data=NA, nrow=400, ncol=5)      ## reg estimator, n=10, sd=5

    for(i in 1:400){
      # generate the population
      popx = rgamma(1000,4)+40
      popy1 = rnorm(1000,100+2*popx,10)
      popy2 = rnorm(1000,100+2*popx,5)
      smp1 = sample(1:1000,10)
      smp2 = sample(1:1000,50)

      # when sample size = 10, store each simulation to the matrix
      ratio_size10_sd10[i,] = ratiotot(smp1,popy1,popx)
      reg_size10_sd10[i,] = regtot(smp1,popy1,popx)
      ratio_size10_sd5[i,] = ratiotot(smp1,popy2,popx)
      reg_size10_sd5[i,] = regtot(smp1,popy2,popx)
    }

    # take the average of the 400 simulation
    apply(ratio_size10_sd10,2,mean)
    apply(reg_size10_sd10,2,mean)
    apply(ratio_size10_sd5,2,mean)
    apply(reg_size10_sd5,2,mean)

>     apply(ratio_size10_sd10,2,mean)
[1] 188071.8895   2817.6482 181523.5872  13096.6046      0.9375
>     apply(reg_size10_sd10,2,mean)
[1] 188206.020   2824.324 182259.220  11893.599      0.875
>     apply(ratio_size10_sd5,2,mean)
[1] 187917.1414   1706.9244 183881.5277   8071.2274      0.9075
>     apply(reg_size10_sd5,2,mean)
[1] 187962.1293   1276.3046 184856.7225   6210.8137      0.9075

# Simulation Case 2: sample size is 50
    ratio_size50_sd10 = matrix(data=NA, nrow=400, ncol=5)   ## ratio estimator, n=10, sd=10
    reg_size50_sd10 = matrix(data=NA, nrow=400, ncol=5)     ## reg estimator, n=10, sd=10
    ratio_size50_sd5 = matrix(data=NA, nrow=400, ncol=5)    ## ratio estimator, n=10, sd=5
    reg_size50_sd5 = matrix(data=NA, nrow=400, ncol=5)      ## reg estimator, n=10, sd=5

    for(i in 1:400){
      # generate the population
      popx = rgamma(1000,4)+40
      popy1 = rnorm(1000,100+2*popx,10)
      popy2 = rnorm(1000,100+2*popx,5)
      smp1 = sample(1:1000,10)
      smp2 = sample(1:1000,50)

      # when sample size = 10, store each simulation to the matrix
      ratio_size50_sd10[i,] = ratiotot(smp2,popy1,popx)
      reg_size50_sd10[i,] = regtot(smp2,popy1,popx)
      ratio_size50_sd5[i,] = ratiotot(smp2,popy2,popx)
      reg_size50_sd5[i,] = regtot(smp2,popy2,popx)
    }

    # take the average of the 400 simulation
    apply(ratio_size50_sd10,2,mean)
    apply(reg_size50_sd10,2,mean)
    apply(ratio_size50_sd5,2,mean)
    apply(reg_size50_sd5,2,mean)

>     apply(ratio_size50_sd10,2,mean)
[1] 188022.3322   1215.3450 185095.8012   5853.0622      0.9325
>     apply(reg_size50_sd10,2,mean)
[1] 188022.4257   1121.2836 185349.8340   5345.1834      0.9425
>     apply(ratio_size50_sd5,2,mean)
[1] 188083.1638    733.5318 186269.4407   3627.4462      0.9475
>     apply(reg_size50_sd5,2,mean)
[1] 188065.0474    541.9611 186719.5693   2690.9563      0.9575
```

```
# Construct a population when n=10
set.seed(22334455)
popnew = rnorm(1000, 100+2*popx, 1000)
smpnew = sample(1:1000,10)
cbind(ratiotot(smpnew,popnew,popx)[2],regtot(smpnew,popnew,popx)[2])
```

Comment:

In this simulation study, we performed to compare the ratio estimator and the regression estimator when the model generating the population is the correct one for the regression estimator. Before I started this simulation study, the proposition that I wanted to check if it is true was that even if the model is supposed to work better with the regression estimator, when the sample size is small the ratio estimator does as equivalently good as the regression estimator; when the sample size is getting bigger, the regression estimator would outdo the ratio estimator. (Cochran)

In the simulation, I divided into two different cases where when sample size is 10 and 50 and different standard deviation with both estimator. When sample size is 10, the result show that the point estimate and the interval estimators are pretty similar on both ratio and regression estimator cases. When you see the absolute error, the ratio estimator indeed did a nice job compared to the regression estimator. This finding confirms what Cochran said.

On the other hand, when the sample size gets larger n = 50, the regression estimator seems to take the job over. Plus, as I tried to construct a population when the sample size is 10 the average absolute error of the ratio estimator is at least for times larger than the regression estimator, I set the standard deviation extremely large. From this, we can know that when the standard deviation gets larger, the regression estimator does better estimation job over the ratio estimator. This fortify what Cochran said.

7. Using the same popx as in problem 6 we now get popy by doing the following `popy<-rnorm(1000,100 + 4*popx,5)`. For this population cor(popy,popx) = 0.865. Here you need to do a simulation study where you are estimating this correlation. You will take 300 samples of size n = 60 doing pps sampling proportional to popx. Using these 500 simulate correlations, find your estimate, its relative bias, its absolute error, the length of the approximated 0.95 credible interval and whether or not this interval contains the true correlation. Finally present the average of these quantities over the 300 samples.

```
set.seed(22334455)
popx = rgamma(1000,4) + 40
popy = rnorm(1000, 100+4*popx,5)
cor(popy,popx)

correlation.est = function(popx,popy,n,Rsim,Rsmp){
  N = length(popy)
  pop.cor = cor(popx,popy)
  inprb = n*popx/sum(popx)
  wt = 1/inprb
  ans= rep(0,5)

  for(i in 1:Rsmp){
    z=1.96
    smp = sort(sample(1:N,n,prob=popx))
    xsmp = popx[smp]
    ysmp = popy[smp]

    wts = wt[smp]
    wts = n*wts/sum(wts)
    dcor = rep(0,Rsim)
    for(k in 1:Rsim){
      simpop = wtpolyap(ysmp,wts,N-n)
      dcor[k] = cor(xsmp,ysmp)
    }

    est.cor = mean(dcor)
    abserr = abs(est.cor - pop.cor)
    relabias = (est.cor - pop.cor)/pop.cor
    estvr = (1-n/N)*var(xsmp)/n
    lwbd = est.cor - z*sqrt(estvr)
    upbd = est.cor + z*sqrt(estvr)
    if(lwbd <= pop.cor & pop.cor <= upbd) {cov <- 1}
    else{cov <- 0}
    ans = ans + c(est.cor,relabias, abserr, 2*1.96*sqrt(estvr), cov)
  }
  ans = round(ans/Rsmp, digits=6)
  names(ans) =c("estimation", "relative bias", "absolute error", "0.95 credible interval",
"frequency of coverage")
  print(ans)
}

set.seed(22334455)
correlation.est(popx=popx, popy=popy, n=60, Rsim=500, Rsmp=300)
```

```
            estimation          relative bias         absolute error
              0.866077               0.000706               0.027347
0.95 credible interval  frequency of coverage
              1.024910               1.000000
```

Comment:

In this simulation study, I estimated the correlation. I took 300 samples of size n=60 doing pps sampling proportional to pox. Using the R handout polyapost on the class web page, I found the estimate, its relative bias, its absolute error, the length of the approximated 0.95 credible interval and frequency of coverage.