STAT 341 - Assignment 3

Due Friday Nov 8 at 9am - to be submitted through crowdmark

Estimating Different Age Attributes

For this question you will need the Titanic data which can be found in the `carData` package. Here we will focus on the male passengers and the relationship between age and survival.

```
library(carData)
data(TitanicSurvival)
Titanic = na.omit(TitanicSurvival)
Titanic = Titanic[Titanic$sex == "male",]
Titanic$survived1 = as.numeric(Titanic$survived == "yes")
head(Titanic)
```

```
##                                 survived  sex     age passengerClass
## Allison, Master. Hudson Trevor       yes male  0.9167            1st
## Allison, Mr. Hudson Joshua Crei       no male 30.0000            1st
## Anderson, Mr. Harry                  yes male 48.0000            1st
## Andrews, Mr. Thomas Jr                no male 39.0000            1st
## Artagaveytia, Mr. Ramon               no male 71.0000            1st
## Astor, Col. John Jacob                no male 47.0000            1st
##                                 survived1
## Allison, Master. Hudson Trevor          1
## Allison, Mr. Hudson Joshua Crei         0
## Anderson, Mr. Harry                     1
## Andrews, Mr. Thomas Jr                  0
## Artagaveytia, Mr. Ramon                 0
## Astor, Col. John Jacob                  0
```

Use the sample below from the Titanic data to answer the following questions.

```
set.seed(341)
TitanicSample = sample(658, 25)
TitanicSample
```

```
##  [1] 265 596 270 334 653 273  93  58 113 235 243 411 231 652 127 640 217
## [18] 626 279 482 395 410 162   7 603
```

to answer the questions below.

```
popSize <- function(pop) {nrow(as.data.frame(pop))}
sampSize <- function(samp) {popSize(samp)}

createInclusionProbFn <- function(pop, sampSize) {
  N <- popSize(pop)
  n <- sampSize
  function(u) { rep(n/N, length(u)) } # Changed to be vectorized!
}

createJointInclusionProbFn <- function(pop, sampSize) {
  N <- popSize(pop)
  n <- sampSize
  function(u,v) {
    ## Note that the answer depends on whether u and v
```

```r
    ## are the same or different
    # if (u == v) {n/N} else {(n * (n-1)) / (N * (N-1))}
    ifelse(u == v, n/N, (n * (n-1)) / (N * (N-1))) # Changed to be vectorized!
  }
}

createHTestimator <- function(pi_u_fn) {
  function(samp, variateFn) {
    Reduce(`+`,
           Map(function(u) {variateFn(u)/ pi_u_fn(u)}, samp),
           init = 0
    )
  }
}

createHTVarianceEstimator <- function(pop, pi_u_fn, pi_uv_fn) {
  function(samp, variateFn) {
    Reduce(`+`,
           Map(function(u) {
             pi_u <- pi_u_fn(u)
             y_u <- variateFn(u)
             Reduce(`+`,
                    Map(function(v) {
                      pi_v <- pi_u_fn(v)
                      pi_uv <- pi_uv_fn(u, v)
                      y_v <- variateFn(v)
                      Delta_uv <- pi_uv - pi_u * pi_v
                      result <- (Delta_uv  * y_u * y_v)
                      result <- result/(pi_uv * pi_u * pi_v)
                      result
                    },
                    samp),
                    init = 0)
           },
           samp
           ),
           init = 0)
  }
}
```

a) [**3 Marks**] Calculate the Horvitz-Thompson estimate of the average age and provide the standard error of the estimate.

```r
N = nrow(Titanic)
n = 25

inclusionProb       <- createInclusionProbFn(1:N, sampSize = n)
inclusionJointProb  <- createJointInclusionProbFn(1:N, sampSize = n)

titanicHTestimator   <- createHTestimator(inclusionProb)
HTVarianceEstimator <- createHTVarianceEstimator(1:N,
                                                 pi_u_fn = inclusionProb,
                                                 pi_uv_fn = inclusionJointProb)
```

```r
createvariateFnN <- function(popData, variate, N) {
  function (u) {popData[u, variate]/N}
}

titanicHTestimator(TitanicSample, createvariateFnN(Titanic, 'age', N))
```

```
## [1] 34.85667
```

```r
sqrt(HTVarianceEstimator(TitanicSample, createvariateFnN(Titanic, 'age', N)))
```

```
## [1] 3.306294
```

The Horvitz-Thompson estimate of the average age is 30.1 and the standard error of the estimate is 2.771284

b) [**3 Marks**] Calculate the Horvitz-Thompson estimate of the proportion of age less than or equal to 20 and provide the standard error of the estimate.

```r
createvariateFnN2 <- function(popData, variate, N, y) {
  function (u) {(popData[u, variate] <= y)/N}
}

titanicHTestimator(TitanicSample, createvariateFnN2(Titanic, 'age', N, 20))
```

```
## [1] 0.12
```

```r
sqrt(HTVarianceEstimator(TitanicSample, createvariateFnN2(Titanic, 'age', N, 20)))
```

```
## [1] 0.06506018
```

The Horvitz-Thompson estimate of the proportion of age less than or equal to 20 is 0.2, and the standard error of the estimate is 0.08008354.

c) [**3 Marks**] Calculate the Horvitz-Thompson estimate of the proportion of age less than or equal to 50 and provide the standard error of the estimate.

```r
createvariateFnN2 <- function(popData, variate, N, y) {
  function (u) {(popData[u, variate] <= y)/N}
}

titanicHTestimator(TitanicSample, createvariateFnN2(Titanic, 'age', N, 50))
```

```
## [1] 0.88
```

```r
sqrt(HTVarianceEstimator(TitanicSample, createvariateFnN2(Titanic, 'age', N, 50)))
```

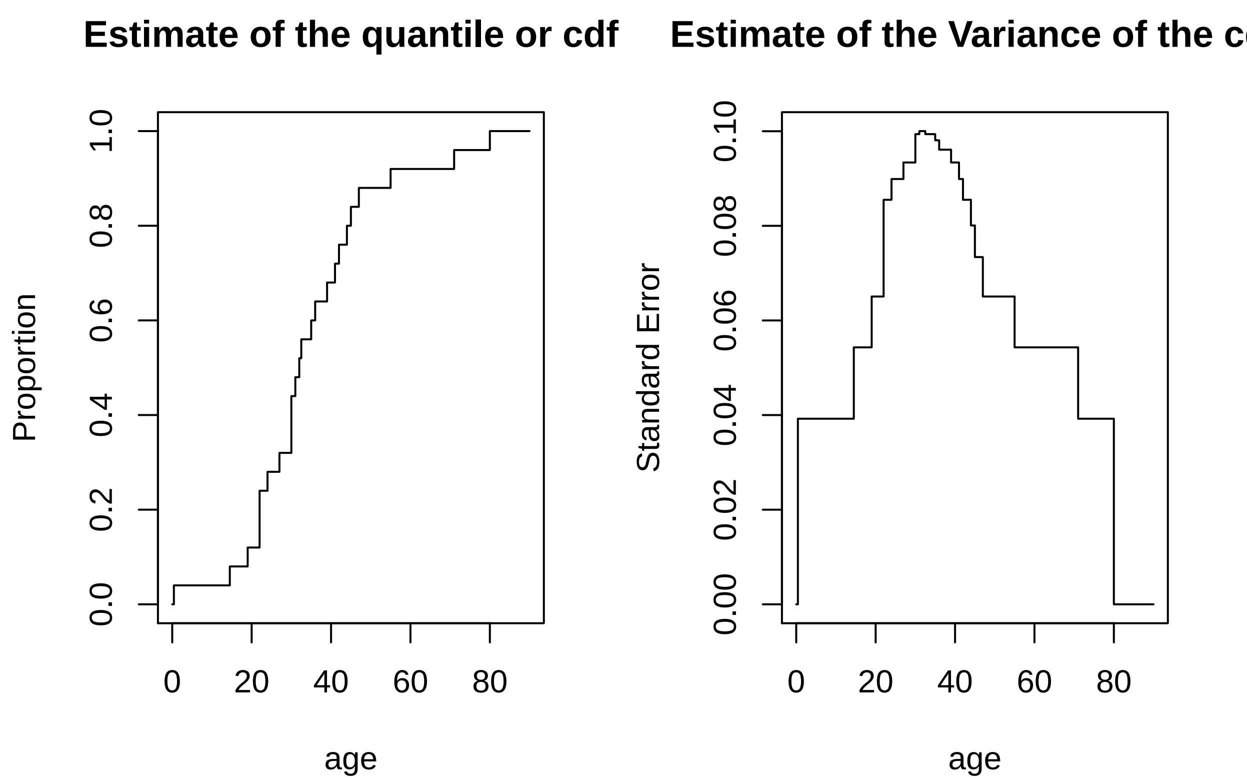
```
## [1] 0.06506018
```

The Horvitz-Thompson estimate of the proportion of age less than or equal to 50 is 0.92, and the standard error of the estimate is 0.0543153.

d) [**5 Marks**] In two separate graphs, plot the Horvitz-Thompson estimate of the cumulative distribution function of age and the standard error of the estimate. + **Note** Similar to the cdf, the standard error is also a function of age.

```r
yseq = c(0, sort(Titanic$age[TitanicSample]), 90)

cdfEstimate = sapply(yseq, function(y) {
  ptitanic <- createvariateFnN2(Titanic, 'age', N, y)
  titanicHTestimator(TitanicSample, ptitanic)
   } )
```

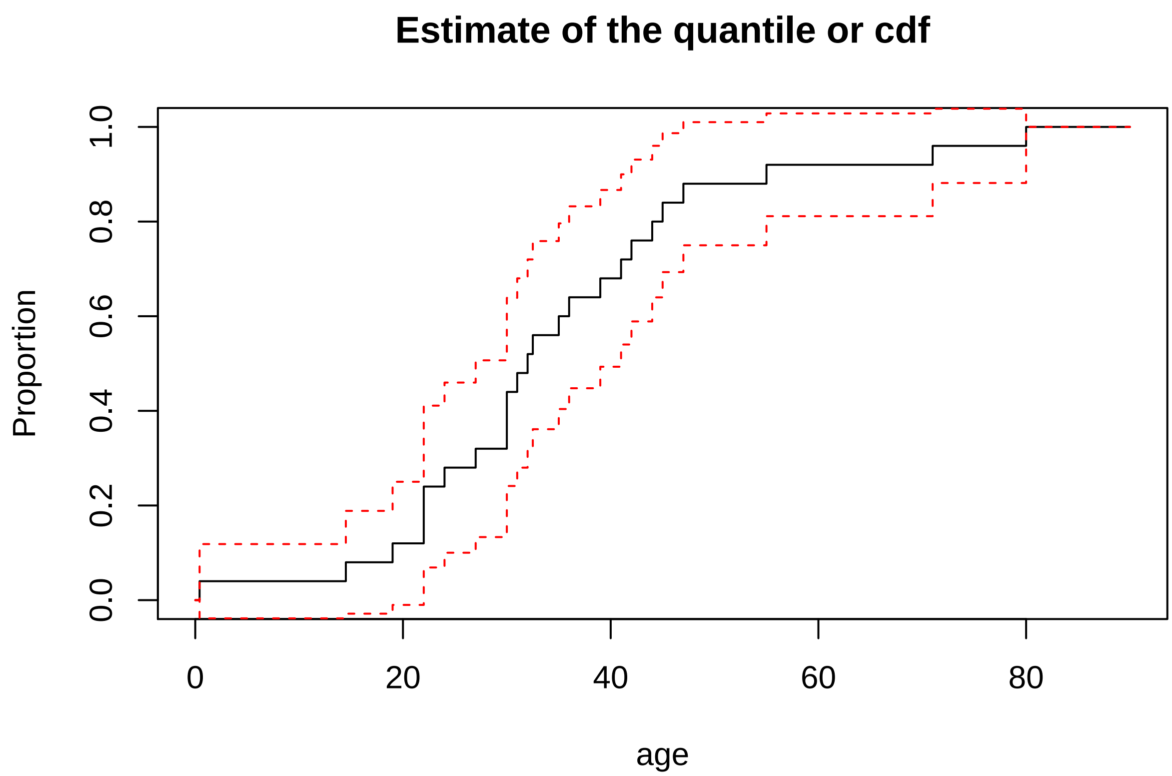
```
variancecdfEstimate = sapply(yseq, function(y) {
  ptitanic <- createvariateFnN2(Titanic, 'age', N, y)
  HTVarianceEstimator(TitanicSample, ptitanic)
   } )
stderr.cdf = sqrt(round(variancecdfEstimate,14))

par(mfrow=c(1,2) )
plot(yseq, cdfEstimate,
  type = 's',
  ylab = "Proportion",
  xlab = "age",
  main = "Estimate of the quantile or cdf"
)
plot(yseq, stderr.cdf,
  type = 's',
  ylab = "Standard Error",
  xlab = "age",
  main = "Estimate of the Variance of the cdf"
)
```



e) [**3 Marks**] Plot the Horvitz-Thompson estimate of the cdf of age, and overlay the lines of ± 2 times the standard error.

```
plot(yseq, cdfEstimate,
  type = 's',
  ylab = "Proportion",
```

```
  xlab = "age",
  main = "Estimate of the quantile or cdf"
)
lines(yseq, cdfEstimate - 2*stderr.cdf, type='s',col=2, lty=2)
lines(yseq, cdfEstimate + 2*stderr.cdf, type='s',col=2, lty=2)
```



Estimating the median for Radar Data

- The data were supplied by A. Frery. They are a part of a synthetic aperture satellite radar image corresponding to a suburb of Munich. Provided are coordinates and values corresponding to three frequency bands for each of 1573 pixels.
- The data can be found in the `robustbase` package.

```
data(radarImage, package="robustbase")
head(radarImage)
```

```
##   X.coord Y.coord  Band.1  Band.2  Band.3
## 1      59       1  157.20 -150.50  30.020
## 2      60       1   52.12  -72.61  -6.376
## 3      61       1 -188.10  -82.81 -55.630
## 4      62       1  -17.10   10.09 -21.230
## 5      52       2   18.39  -22.43  86.390
```

```
## 6      53       2 -144.20 -120.30 106.700
```

a) [10 Marks] For the variable `Band.2`, suppose we are interested in estimating the population median using the trimmed average. i) Generate $m = 10000$ samples from the radarImage data, each with sample size $n = 75$. For each sample, calculate the trimmed average with varying fractions of observations to be trimmed (0 to 0.5 by 0.05). Use help(mean) and the argument trim for more details.

```
m = 10000
band2tmeans = matrix(nrow = 10000, ncol = 11)
for(i in 1:m) {
  s = sample(nrow(radarImage), 75)
  for(j in 0:10) {
    band2tmeans[i, j + 1] = mean(radarImage$Band.2[s], trim = j * 0.05)
  }
}
```

```
ii) Report the sampling bias (SB), sampling standard deviation (SD) and square root mean square error (R
```

```
n.set = seq(0, 0.5, by=0.05)
result = matrix(nrow=length(n.set), ncol = 3,
                dimnames = list(n.set, c("Median SB", "Median SD", "Median RMSE")))
for (i in 1:length(n.set)){
  mu = mean(band2tmeans[,i])
  var = sum((band2tmeans[,i] - mu)^2) / m
  result[i, 1] = mu - median(radarImage$Band.2)
  result[i, 2] = sqrt(var)
  result[i, 3] = sqrt(var + result[i, 1]^2)
}
round(result,4)
```

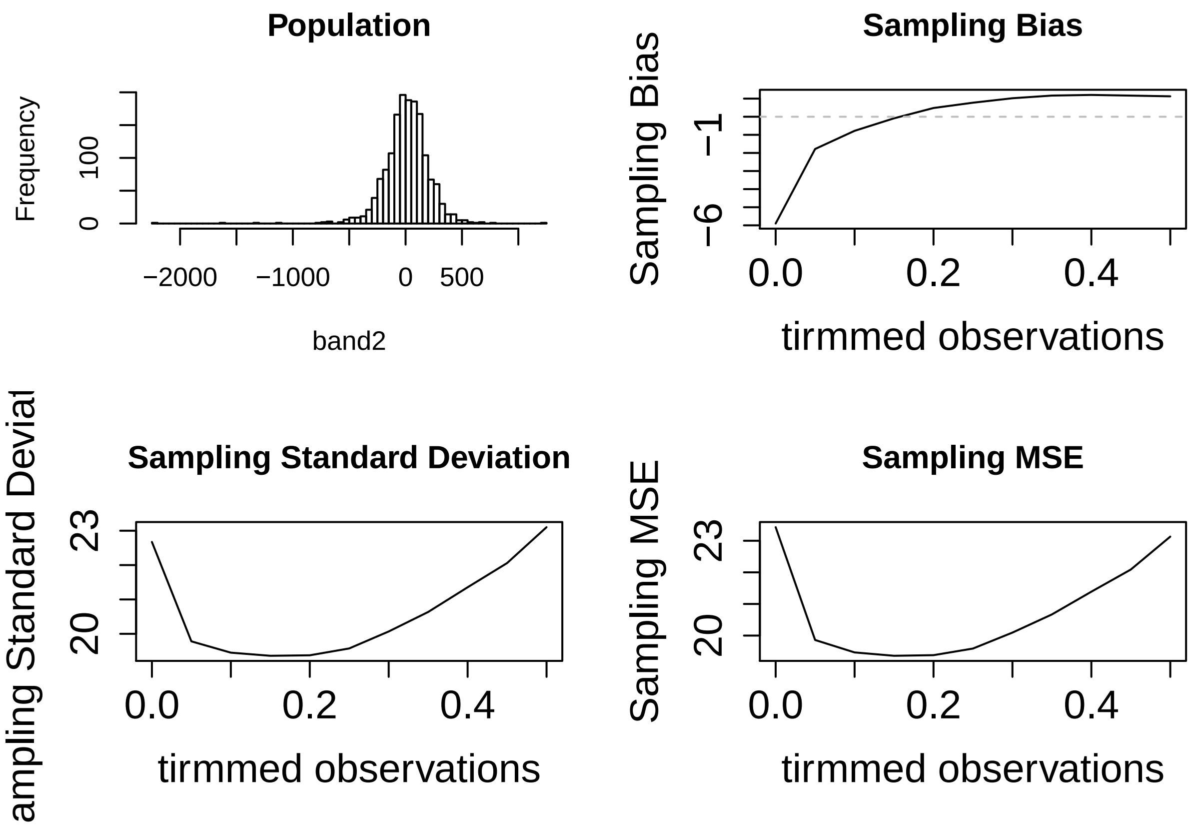
```
##      Median SB Median SD Median RMSE
## 0      -5.8988   22.6736     23.4283
## 0.05   -1.7827   19.7806     19.8608
## 0.1    -0.7784   19.4514     19.4670
## 0.15   -0.0871   19.3611     19.3612
## 0.2     0.4819   19.3746     19.3806
## 0.25    0.7745   19.5747     19.5900
## 0.3     1.0202   20.0683     20.0942
## 0.35    1.1681   20.6353     20.6684
## 0.4     1.2035   21.3543     21.3882
## 0.45    1.1668   22.0581     22.0889
## 0.5     1.1275   23.1025     23.1300
```

The absolute value of sampling bias is large at first, then it oscillates value 1. Standard deciation and square root mean square error do not change very much but both first decreased and increased.

```
iii) Plot a histogram of the population. In addition, plot the SB, SD and RMSE against the fraction of o
```

```
par(mfrow=c(2,2))
hist(radarImage$Band.2, breaks="FD", main="Population", xlab = "band2")

plot( result[,1]~n.set, main="Sampling Bias", type='l', ylim=range(result[,1]),
      ylab="Sampling Bias", xlab="tirmmed observations" ,cex.lab=1.5 , cex.axis=1.5)
abline(h=0, lty=2, col="grey")

plot(result[,2]~n.set, main="Sampling Standard Deviation", type='l', xlab="tirmmed observations",
      ylab="Sampling Standard Deviation",cex.lab=1.5 , cex.axis=1.5)
```

```r
plot( result[,3]~n.set, main="Sampling MSE", type='l',
      xlab="tirmmed observations", ylab="Sampling MSE",
      cex.lab=1.5 , cex.axis=1.5)
```



The histogram looks left skewed and there are many extremely small values. So it can affect the trimmed average a lot by not trim out enough extreme values. Average is not resistant to extreme values, which would make sampling bias negative.

b) [**10 Marks**] Repeat a) using the variable `Band.3`.

```r
m = 10000
band3tmeans = matrix(nrow = 10000, ncol = 11)
for(i in 1:m) {
  s = sample(nrow(radarImage), 75)
  for(j in 0:10) {
    band3tmeans[i, j + 1] = mean(radarImage$Band.3[s], trim = j * 0.05)
  }
}
```

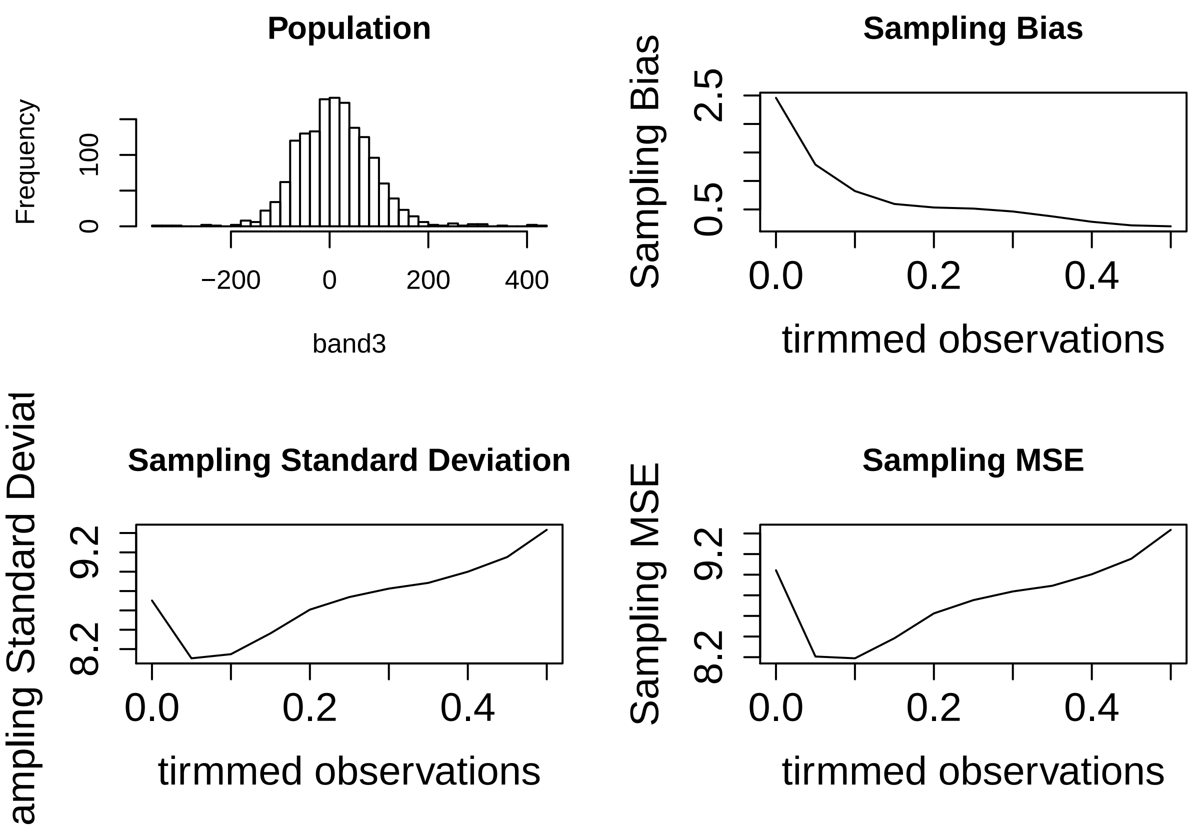
```r
n.set = seq(0, 0.5, by=0.05)
result2 = matrix(nrow=length(n.set), ncol = 3,
                dimnames = list(n.set, c("Median SB", "Median SD", "Median RMSE")))
for (i in 1:length(n.set)){
  mu = mean(band3tmeans[,i])
  var = sum((band3tmeans[,i] - mu)^2) / m
  result2[i, 1] = mu - median(radarImage$Band.3)
  result2[i, 2] = sqrt(var)
  result2[i, 3] = sqrt(var + result2[i, 1]^2)
}
```

```r
}
round(result2,4)
```

```
##      Median SB Median SD Median RMSE
## 0       2.4592    8.7023      9.0431
## 0.05    1.2858    8.1047      8.2061
## 0.1     0.8226    8.1474      8.1888
## 0.15    0.5959    8.3625      8.3837
## 0.2     0.5344    8.6081      8.6247
## 0.25    0.5150    8.7380      8.7531
## 0.3     0.4639    8.8252      8.8374
## 0.35    0.3779    8.8847      8.8927
## 0.4     0.2833    8.9998      9.0042
## 0.45    0.2208    9.1520      9.1547
## 0.5     0.2040    9.4331      9.4354
```

Sampling bias is decreasing towards 0 as the proporation trimmed observations increases. SD and RMSE is smaller comapre to SD and RMSE of band2.

```r
par(mfrow=c(2,2),oma=c(0,0,0,0))
hist(radarImage$Band.3, breaks="FD", main="Population", xlab = "band3")
plot( result2[,1]~n.set, main="Sampling Bias", type='l', ylim=range(result2[,1]),
      ylab="Sampling Bias", xlab="tirmmed observations" ,cex.lab=1.5 , cex.axis=1.5)
abline(h=0, lty=2, col="grey")

plot(result2[,2]~n.set, main="Sampling Standard Deviation", type='l', xlab="tirmmed observations",
      ylab="Sampling Standard Deviation",cex.lab=1.5 , cex.axis=1.5)

plot( result2[,3]~n.set, main="Sampling MSE", type='l',
      xlab="tirmmed observations", ylab="Sampling MSE",
      cex.lab=1.5 , cex.axis=1.5)
```



Compare to band2, band3 has both extremely large values and extremely small value. Trimmed averge can efftively remove those extreme values which makes sampling bias decreased as trimmed observation increases. Trimmed averge estimates meadian of band3 better.