Noen enkle simuleringsbetrakninger

La p være reell andel smittede, k antall prøver slått sammen i en pool (bulk) og $\pi(k)$ sannsynligheten for at en pool av k tester gir positivt utslag. Dersom vi for et første overslag ser bort fra redusert sensitivitet (antar altså inntil videre at sensitiviten og spesifisiteten er 100%) er

$$\pi(k) = 1 - (1-p)^k \approx kp \tag{1}$$

Vi kan estimere $\pi(k)$ ved N_+/N der N_+ er antall prøvepooler som er positive og N er antall prøvepooler. Totalt antall personer man tar prøver fra blir da $n_{pers} = kN$. Fra (1) får vi at

$$\hat{p} = 1 - (1 - N_+/N)^{1/k} \approx \frac{N_+}{kN}$$

(Tilnærmingen er kun god for små p og da er altså estimatoren tilnærmet forventningsrett.) Under er fordelingen til $\hat{p}$ basert på 100 000 simuleringer vist for ulike p og ulike valg av k. Har brukt $n_{pers} = 2000$ i alle simuleringer, dvs antall prøvepooler som må testes er da n_{pers}/k (denne er rapportert sammen med bias i tekstutskrift under hvert funksjonskall).

```
# Simulate the distribution of p-hat for given p, k, N
simp <- function(p,k,N,nsim=10^5){
  Npos <- rbinom(nsim,N,1-(1-p)^k)
  phat <- 1-(1-Npos/N)^(1/k)
  phat
}

# Simulate the distribution of p-hat for a sequence of k-values
# Keep number of persons (npers) fixed, thus number of pools  (N) changes
# as a function of k.
simkseq <- function(p,k,npers,nsim=10^5){
  phatmat <- matrix(nrow=nsim,ncol=length(k))
  for(i in 1:length(k))
    phatmat[,i] <- simp(p,k[i],round(npers/k[i]),nsim)
  boxplot(phatmat, use.cols=T, xaxt="n", ylab="p-hat", xlab="k",
          main=paste("p =",p,", ", " Antall personer =",npers))
  axis(side=1, at=1:length(k), labels = k)
  res <- rbind(k,round(npers/k), round(colMeans(phatmat)-p,digits=5))
  row.names(res) <- c("k", "N","bias")
  res
}

par(mfrow=c(2,2))
simkseq(p=0.001,k=c(1,2,4,8,16,32,64),npers=2000)
```

```
##      [,1] [,2] [,3] [,4] [,5]    [,6]    [,7]
## k       1    2    4    8   16 3.2e+01 6.4e+01
## N    2000 1000  500  250  125 6.2e+01 3.1e+01
## bias    0    0    0    0    0 1.0e-05 2.0e-05
```

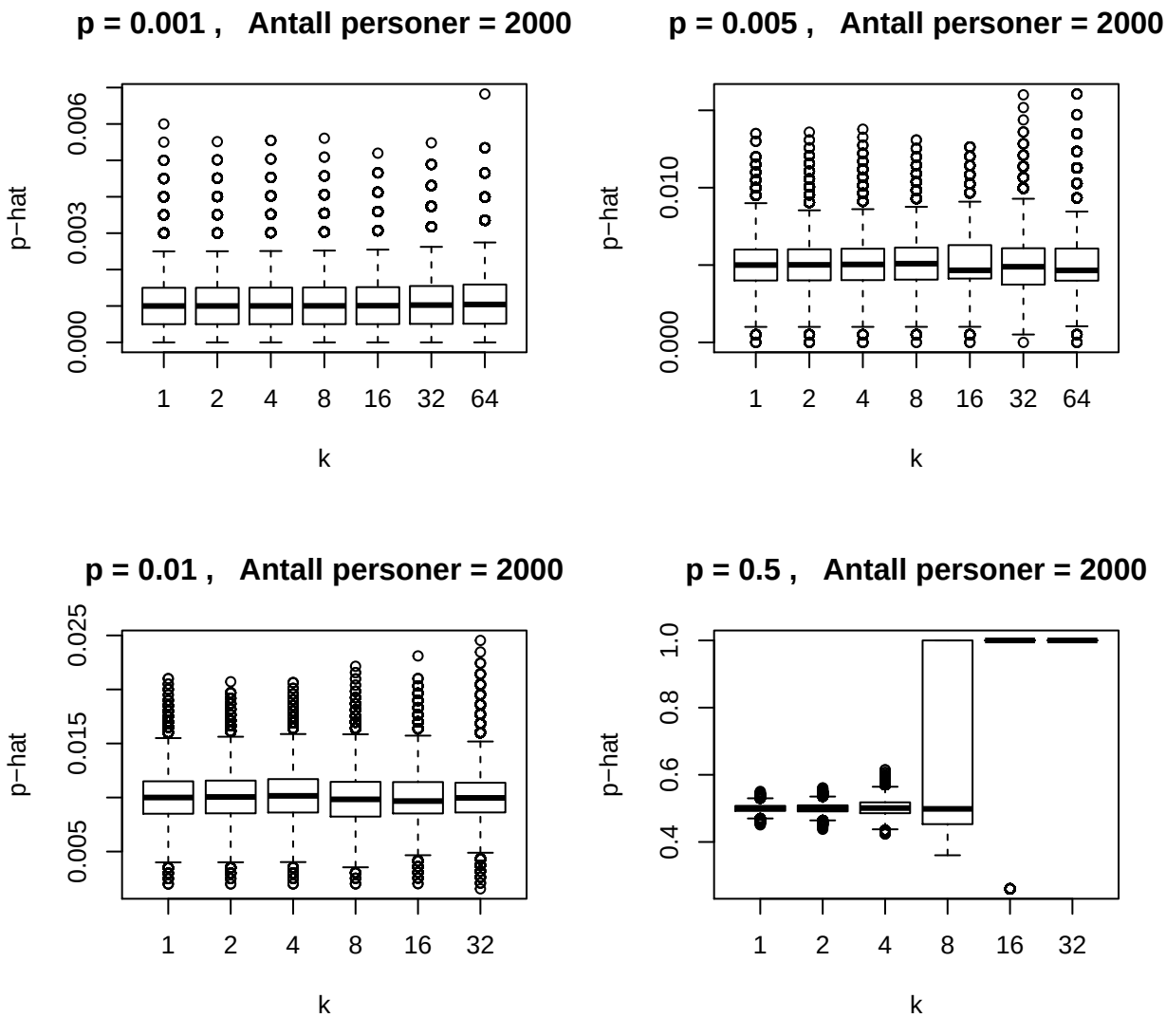
```
simkseq(p=0.005,k=c(1,2,4,8,16,32,64),npers=2000)
```

```
##      [,1] [,2]  [,3]    [,4]     [,5]    [,6]    [,7]
## k       1    2 4e+00 8.0e+00 1.60e+01 3.2e+01 64.0000
## N    2000 1000 5e+02 2.5e+02 1.25e+02 6.2e+01 31.0000
## bias    0    0 1e-05 1.0e-05 2.00e-05 5.0e-05  0.0001
```

```
simkseq(p=0.01,k=c(1,2,4,8,16,32),npers=2000)
```

```
##      [,1] [,2]  [,3]    [,4]     [,5]    [,6]
## k       1    2 4e+00 8.0e+00 1.60e+01 3.2e+01
## N    2000 1000 5e+02 2.5e+02 1.25e+02 6.2e+01
## bias    0    0 2e-05 2.0e-05 4.00e-05 9.0e-05
```

```
simkseq(p=0.5,k=c(1,2,4,8,16,32),npers=2000)
```




```
##        [,1]    [,2]    [,3]       [,4]       [,5] [,6]
## k     1e+00 2.0e+00 4.0e+00    8.00000   16.00000 32.0
```

```
## N     2e+03 1.0e+03 5.0e+02 250.00000 125.00000 62.0
## bias -3e-05 1.9e-04 1.5e-03   0.17204   0.49867  0.5
```

Legge til test-sensitivitet

La $s(k)$ være sensitivitet til testen ved pool av k tester. Da blir

$$\pi(k) = s(k)(1-(1-p)^k) \approx s(k)kp \tag{2}$$

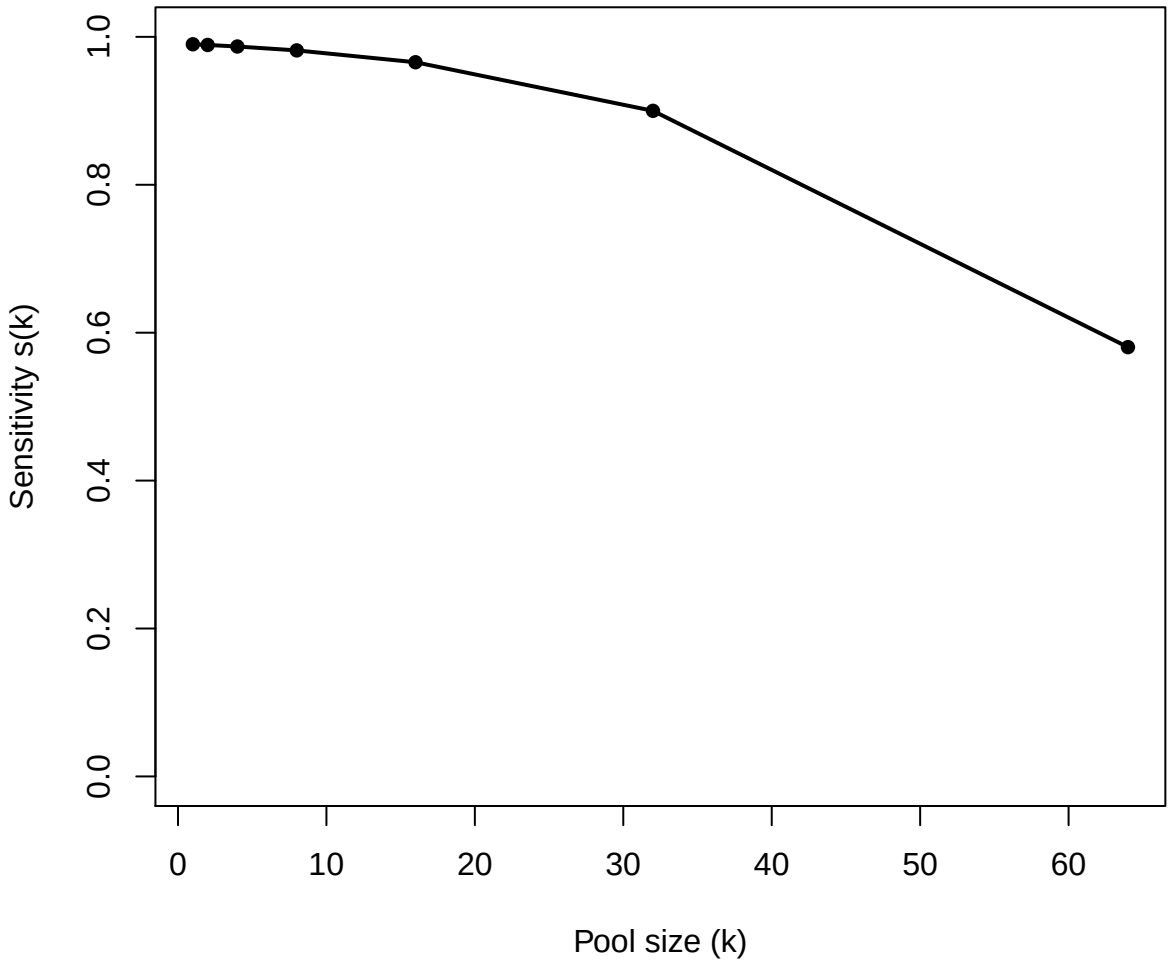
Ved å invertere og sette inn estimator får vi:

$$\hat{p} = 1-(1-N_+/(s(k)N))^{1/k} \approx \frac{N_+}{s(k)kN}$$

Ved å legge dette inn i simuleringen og anta en bestemt form på sensitivitetskurven får vi:

```r
# Simulate the distribution of p-hat for given p, k, s, N
simps <- function(p,k,s,N,nsim=10^5){
  Npos <- rbinom(nsim,N,s*(1-(1-p)^k))
  phat <- 1-(1-Npos/(s*N))^(1/k)
  phat
}

sens1 <- function(k){
  sens <- 0.98^(1+log(k))
  return(sens)
}

sens2 <- function(k){
  # sens 0.99 at k=1, 0.90 at k=32
  b1 <- (qnorm(0.9) - qnorm(0.99))/(32 - 1)
  b0 <- qnorm(0.99) - b1
  sens <- pnorm(b0 + b1*k)
  return(sens)
}

sens <- sens2

simksseq <- function(p,k,s,npers,nsim=10^5){
  phatmat <- matrix(nrow=nsim,ncol=length(k))
  for(i in 1:length(k))
    phatmat[,i] <- simps(p,k[i],s[i],round(npers/k[i]),nsim)
  boxplot(phatmat, use.cols=T, xaxt="n", ylab="p-hat", xlab="k",
          main=paste("p =",p,", ", " Antall personer =",npers))
  axis(side=1, at=1:length(k), labels = k)
  #
  .lower <- apply(phatmat, 2, quantile, probs = 0.025, na.rm = T) # noen missing her!
  .med <- apply(phatmat, 2, quantile, probs = 0.5, na.rm = T) # noen missing her!
  .upper <- apply(phatmat, 2, quantile, probs = 0.975, na.rm = T) # noen missing her!
  res <- rbind(k,s,round(colMeans(phatmat)-p,digits=5), .upper, .med, .lower) # kanskje median?
  row.names(res) <- c("k", "assumed sensitivity", "bias", "upper", "median", "lower")
  # res <- signif(res,2)
  temp <- t(res)
  matplot(temp[,"k"], temp[,c("lower", "median", "upper")], type = "o", lty = 1, lwd = 2, pch = 16, col
  abline(a = p, b = 0, lwd = 2, col = "red")
```

```
  res
}
k=c(1,2,4,8,16,32,64)
plot(k, sens(k), type = "o", pch = 16, lwd = 2, ylim = c(0,1),
     xlab = "Pool size (k)", ylab = "Sensitivity s(k)")
```



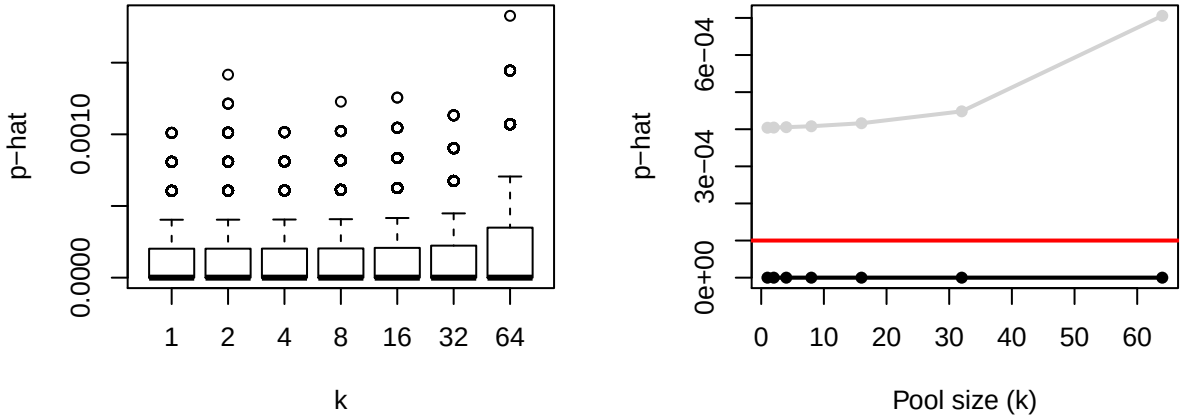

```
par(mfrow=c(2,2))
k=c(1,2,4,8,16,32,64)
simksseq(p=0.0001,k=k,s=sens(k),npers=5000)
```

```
##                            [,1]         [,2]         [,3]         [,4]
## k                  1.0000000000 2.0000000000 4.0000000000 8.0000000000
## assumed sensitivity 0.9900000000 0.9890657676 0.9869673860 0.9817102310
## bias               0.0000000000 0.0000000000 0.0000000000 0.0000000000
## upper              0.0004040404 0.0004045039 0.0004055285 0.0004080345
## median             0.0000000000 0.0000000000 0.0000000000 0.0000000000
## lower              0.0000000000 0.0000000000 0.0000000000 0.0000000000
```

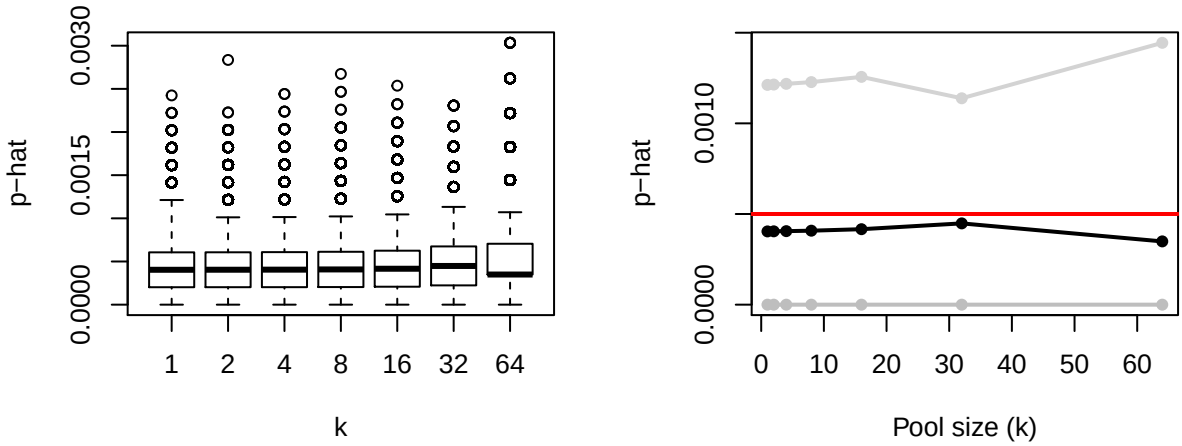
```
##                              [,5]         [,6]         [,7]
## k                    1.600000e+01 3.200000e+01 6.400000e+01
## assumed sensitivity 9.656815e-01 9.000000e-01 5.804529e-01
## bias                 0.000000e+00 0.000000e+00 0.000000e+00
## upper                4.161756e-04 4.482573e-04 7.056817e-04
## median               0.000000e+00 0.000000e+00 0.000000e+00
## lower                0.000000e+00 0.000000e+00 0.000000e+00
```

```
simksseq(p=0.0005,k=k,s=sens(k),npers=5000)
```

p = 1e−04 , Antall personer = 5000



p = 5e−04 , Antall personer = 5000



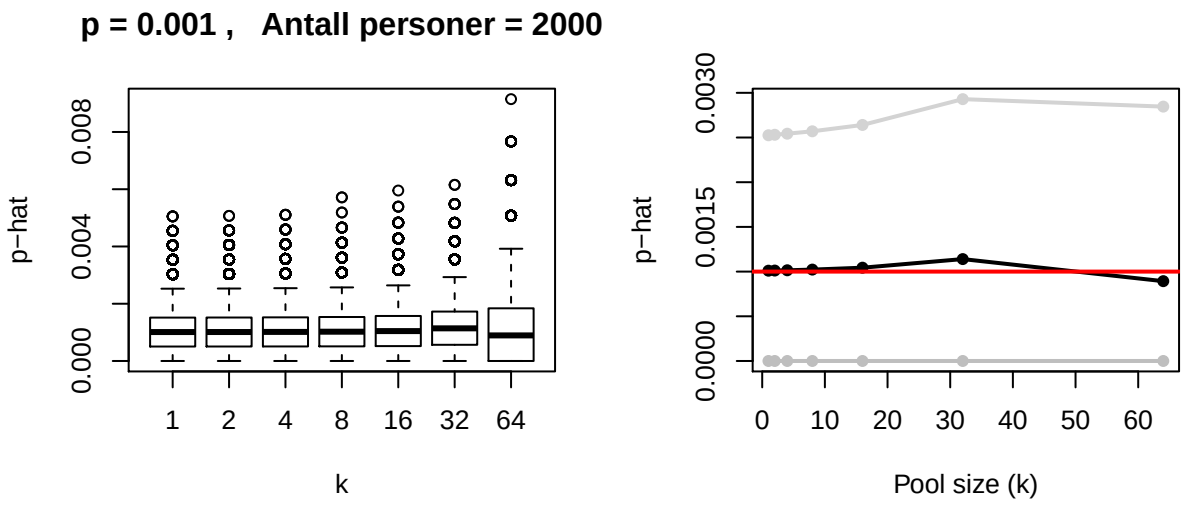

```
##                              [,1]         [,2]         [,3]         [,4]
## k                    1.0000000000 2.0000000000 4.0000000000 8.0000000000
## assumed sensitivity 0.9900000000 0.9890657676 0.9869673860 0.9817102310
## bias                 0.0000000000 0.0000000000 0.0000000000 0.0000000000
## upper                0.0012121212 0.0012140030 0.0012180694 0.0012276184
## median               0.0004040404 0.0004045039 0.0004055285 0.0004080345
## lower                0.0000000000 0.0000000000 0.0000000000 0.0000000000
##                              [,5]         [,6]         [,7]
```

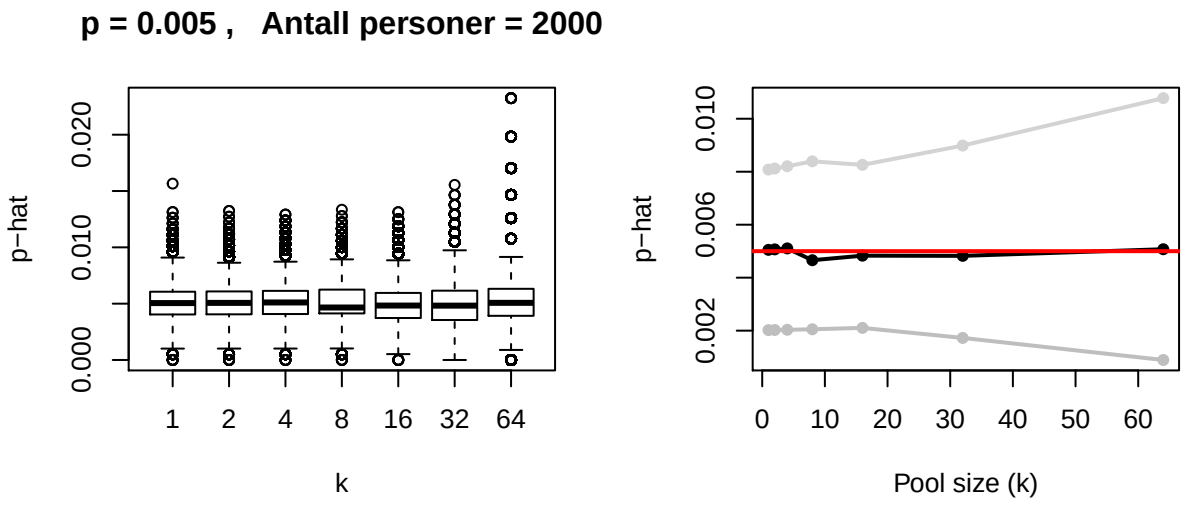
```
## k                    1.600000e+01 3.200000e+01 6.400000e+01
## assumed sensitivity 9.656815e-01 9.000000e-01 5.804529e-01
## bias                 0.000000e+00 0.000000e+00 1.000000e-05
## upper                1.256407e-03 1.138334e-03 1.444227e-03
## median               4.161756e-04 4.482573e-04 3.489182e-04
## lower                0.000000e+00 0.000000e+00 0.000000e+00
```

```
k=c(1,2,4,8,16,32,64)
simksseq(p=0.001,k=k,s=sens(k),npers=2000)
```

```
##                             [,1]        [,2]        [,3]        [,4]
## k                    1.000000000 2.000000000 4.000000000 8.000000000
## assumed sensitivity 0.990000000 0.989065768 0.986967386 0.981710231
## bias                 0.000000000 0.000000000 0.000000000 0.000000000
## upper                0.002525253 0.002530840 0.002542693 0.002569567
## median               0.001010101 0.001011567 0.001014748 0.001022281
## lower                0.000000000 0.000000000 0.000000000 0.000000000
##                             [,5]         [,6]         [,7]
## k                   16.000000000 32.000000000 6.400000e+01
## assumed sensitivity  0.965681465  0.900000000 5.804529e-01
## bias                 0.000000000  0.000010000 3.000000e-05
## upper                0.002640498  0.002929373 2.845753e-03
## median               0.001043668  0.001139987 8.930057e-04
## lower                0.000000000  0.000000000 0.000000e+00
```

```
simksseq(p=0.005,k=k,s=sens(k),npers=2000)
```



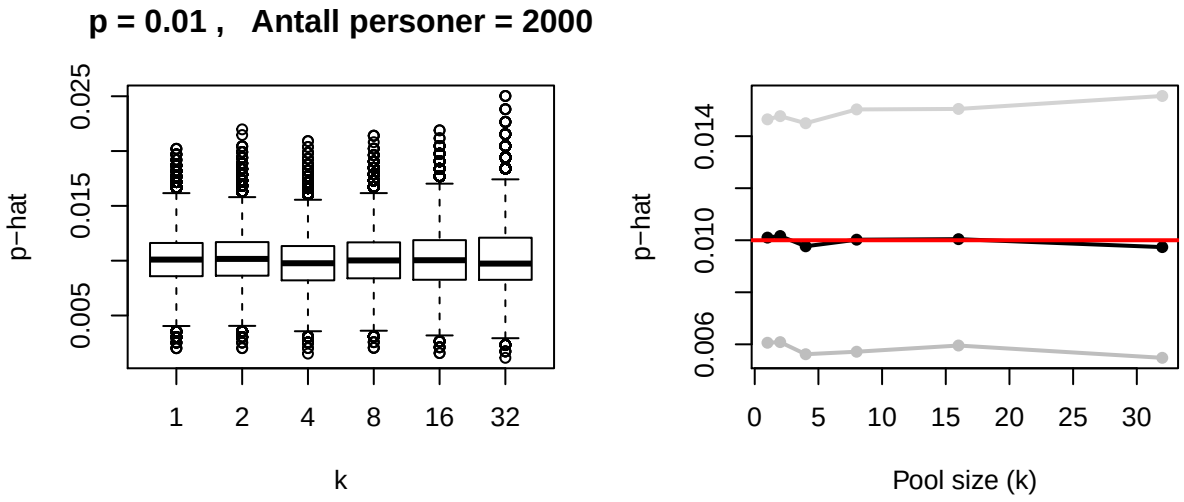


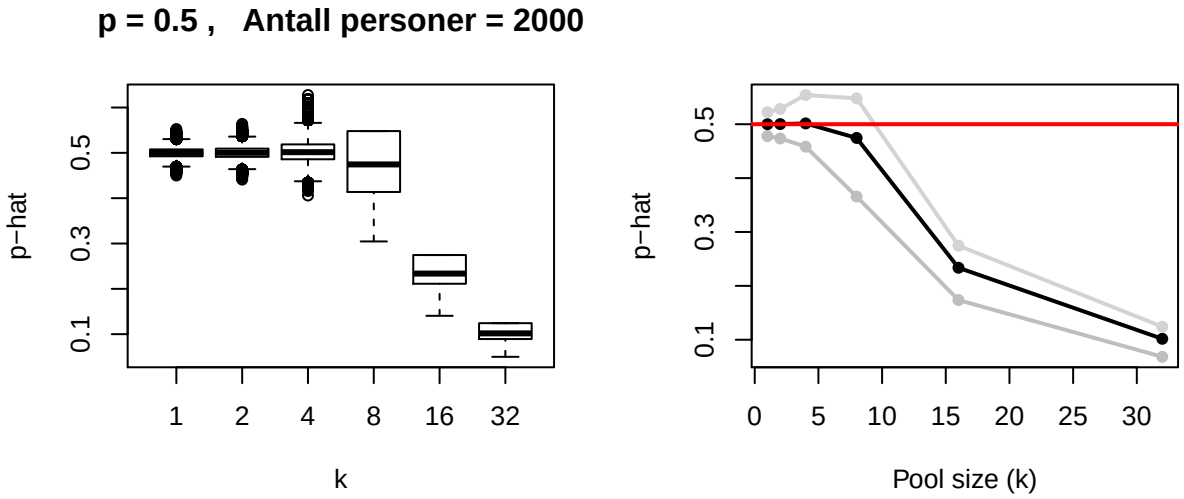

```
##                             [,1]        [,2]        [,3]        [,4]
## k                    1.000000000 2.000000000 4.000000000 8.000000000
## assumed sensitivity  0.990000000 0.989065768 0.986967386 0.981710231
## bias                 0.000000000 0.000000000 0.000010000 0.000010000
## upper                0.008080808 0.008121420 0.008206096 0.008391406
## median               0.005050505 0.005068118 0.005104982 0.004659109
## lower                0.002020202 0.002024159 0.002032598 0.002051937
##                             [,5]         [,6]         [,7]
## k                   16.000000000 32.000000000 6.400000e+01
## assumed sensitivity  0.965681465  0.900000000 5.804529e-01
## bias                 0.000020000  0.000040000 2.000000e-04
## upper                0.008258910  0.008986798 1.077711e-02
## median               0.004831082  0.004824208 5.073801e-03
## lower                0.002103952  0.001725468 8.930057e-04
```

```
k=c(1,2,4,8,16,32)
simksseq(p=0.01,k=k,s=sens(k),npers=2000)
```

```
##                             [,1]        [,2]        [,3]        [,4]
## k                     1.000000000 2.000000000 4.000000000 8.000000000
## assumed sensitivity   0.990000000 0.989065768 0.986967386 0.981710231
## bias                 -0.000010000 0.000000000 0.000010000 0.000020000
## upper                 0.014646465 0.014769366 0.014497078 0.015027948
## median                0.010101010 0.010162186 0.009767625 0.010021542
## lower                 0.006060606 0.006084843 0.005619822 0.005715505
##                             [,5]         [,6]
## k                    16.000000000 32.000000000
## assumed sensitivity   0.965681465  0.900000000
## bias                  0.000040000  0.000100000
## upper                 0.015046341  0.015543959
## median                0.010041817  0.009736410
## lower                 0.005954098  0.005481505
```

```
simksseq(p=0.5,k=k,s=sens(k),npers=2000)
```






```
##                             [,1]        [,2]        [,3]        [,4]        [,5]
```

```
## k                    1.0000000 2.0000000 4.0000000 8.0000000 16.0000000
## assumed sensitivity  0.9900000 0.9890658 0.9869674 0.9817102  0.9656815
## bias                -0.0000200 0.0002600 0.0017900       NaN        NaN
## upper                0.5222222 0.5283024 0.5542415 0.5480047  0.2745597
## median               0.5000000 0.5002029 0.5014615 0.4744866  0.2335988
## lower                0.4777778 0.4736013 0.4583268 0.3657429  0.1736165
##                            [,6]
## k                   32.00000000
## assumed sensitivity  0.90000000
## bias                        NaN
## upper                0.12423166
## median               0.10175471
## lower                0.06830277
```