Effect of bottom coverage to larval presence

Week4-ex3, solution

```
library(ggplot2)
library(StanHeaders)
library(rstan)
```

```
## rstan (Version 2.19.3, GitRev: 2e1f913d3ca3)

## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

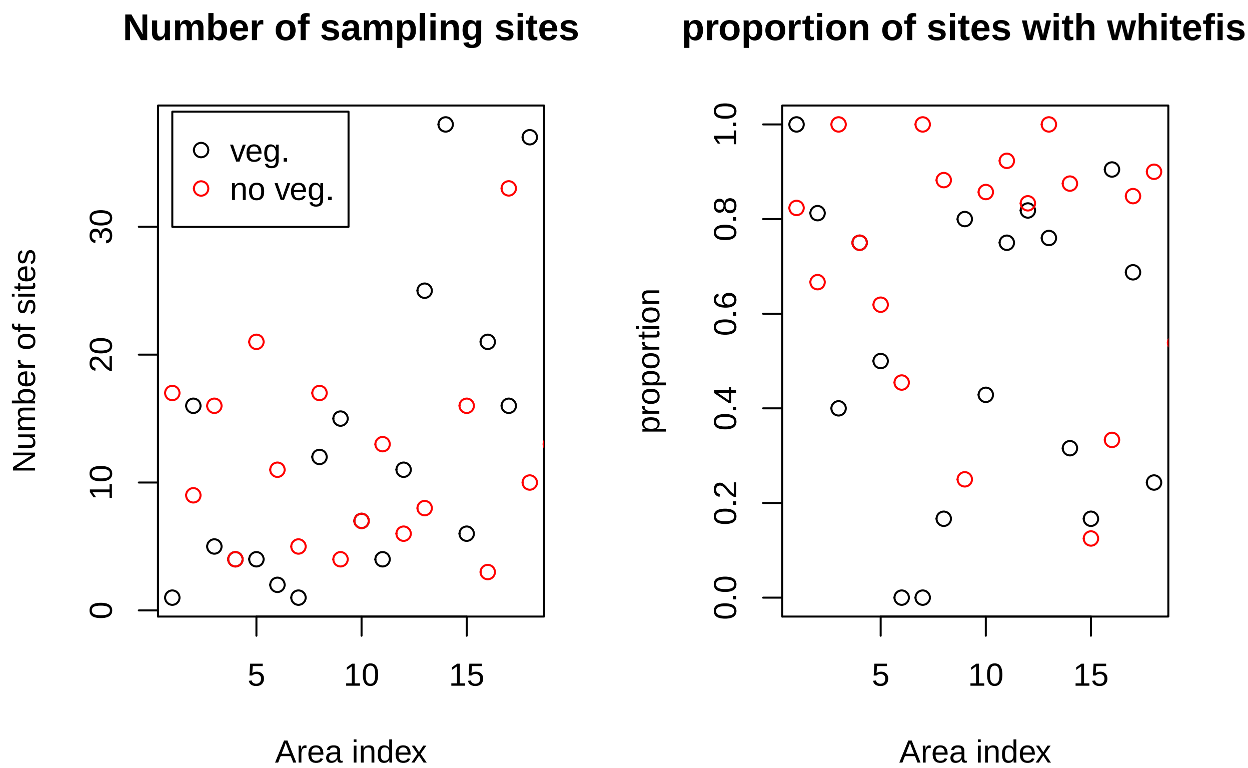
```
library(coda)
```

```
##
## Attaching package: 'coda'

## The following object is masked from 'package:rstan':
##
##     traceplot
```

```
set.seed(123)
options(mc.cores = parallel::detectCores())
rstan_options(auto_write = TRUE)
```

Let's first explore the data a bit more.

```
# Read the data
data = read.csv("white_fishes_data.csv")
# Form a data table for sites without bottom vegetation
y.noveg = table(data$AREANAME[data$BOTTOMCOV==0], data$WHIBIN[data$BOTTOMCOV==0])
colnames(y.noveg) <- c("y=0","y=1")
N.noveg = rowSums(y.noveg)
# Form a data table for sites with bottom vegetation
y.veg = table(data$AREANAME[data$BOTTOMCOV==1], data$WHIBIN[data$BOTTOMCOV==1])
colnames(y.veg) <- c("y=0","y=1")
N.veg = rowSums(y.veg)

par(mfrow=c(1,2))
plot(N.veg, main="Number of sampling sites", xlab="Area index", ylab="Number of sites")
points(N.noveg, col="red")
legend(1, 39, c("veg.","no veg."),col=c("black","red"), pch=1, cex=1, box.lty=1)
plot(y.veg[,2]/N.veg, main="proportion of sites with whitefish", xlab="Area index", ylab="proportion")
points(y.noveg[,2]/N.noveg, col="red")
```




```
print(y.veg)
```

```
##
##                  y=0 y=1
##   Bjuroklubb       0   1
##   Bygdea           3  13
##   Haaparanta       3   2
##   Hailuoto         1   3
##   Harnosand        2   2
##   Hornslandet      2   0
##   Lohtaja          1   0
##   Luvia           10   2
##   Mikkelinsaaret   3  12
##   Mjolefjarden     4   3
##   Nordingra        1   3
##   Pietarsaari      2   9
##   Pitea            6  19
##   Pori            26  12
##   Siipyy           5   1
##   Storsand         2  19
##   Tore             5  11
##   Vaasa           28   9
```

Part 1

The model

```
model="
  data{
   int i;
   int c;
   int <lower = 0> N[i,c];
   int <lower = 0> y[i,c];

  }

  parameters{
    real <lower = 0, upper = 1> mu[c];
    real <lower = 0> s[c];
    real <lower = 0,upper = 1>  theta[i,c];
  }

  model{

    for (j in 1:c){
      mu[j] ~ uniform(0,1);
    }

    for (j in 1:c){
      s[j] ~ lognormal(4,2);
    }

    for(j in 1:i){
      for(k in 1:c){
        theta[j,k] ~ beta( mu[k] * s[k] , s[k] - mu[k] * s[k]);
      }
    }

    for(j in 1:i){
      for(k in 1:c){
        y[j,k] ~ binomial(N[j,k],theta[j,k]);
      }
    }

  }

"
```

```
dataset <- list("c"=2,"i"=19,"y" = cbind(y.noveg[,2],append(y.veg[,2], 0, after = 6)),"N"=cbind(N.noveg
#give

init1 = list(y=0)
init2 = list(y=10)
init3 = list(y=20)
```

```r
inits <- list(init1, init2, init3)
# stan function does all of the work of fitting a Stan model and
# returning the results as an instance of stanfit = post in our exercises.
post=stan(model_code=model,data=dataset,init=inits,warmup=500,iter=1000,chains=3,thin=1)
```

```
## Trying to compile a simple C file
```

```
## Running /usr/lib/R/bin/R CMD SHLIB foo.c
## gcc -std=gnu99 -I"/usr/share/R/include" -DNDEBUG   -I"/usr/lib/R/site-library/Rcpp/include/"  -I"/us
## In file included from /usr/lib/R/site-library/RcppEigen/include/Eigen/Core:88,
##                  from /usr/lib/R/site-library/RcppEigen/include/Eigen/Dense:1,
##                  from /usr/lib/R/site-library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp:1
##                  from <command-line>:
## /usr/lib/R/site-library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:1: error: unknown type na
##   613 | namespace Eigen {
##       | ^~~~~~~~~
## /usr/lib/R/site-library/RcppEigen/include/Eigen/src/Core/util/Macros.h:613:17: error: expected '=',
##   613 | namespace Eigen {
##       |                 ^
## In file included from /usr/lib/R/site-library/RcppEigen/include/Eigen/Dense:1,
##                  from /usr/lib/R/site-library/StanHeaders/include/stan/math/prim/mat/fun/Eigen.hpp:1
##                  from <command-line>:
## /usr/lib/R/site-library/RcppEigen/include/Eigen/Core:96:10: fatal error: complex: No such file or di
##    96 | #include <complex>
##       |          ^~~~~~~~~
## compilation terminated.
## make: *** [/usr/lib/R/etc/Makeconf:167: foo.o] Error 1
```

Summary posterior mu

```r
summary(post,pars="mu", probs = c(0.025, 0.975))
```

```
## $summary
##            mean     se_mean         sd      2.5%     97.5%    n_eff      Rhat
## mu[1] 0.7142182 0.001300077 0.05532249 0.5979124 0.8110573 1810.778 0.9984905
## mu[2] 0.5449345 0.001634402 0.06105875 0.4197484 0.6609843 1395.655 0.9998357
##
## $c_summary
## , , chains = chain:1
##
##          stats
## parameter      mean         sd      2.5%     97.5%
##     mu[1] 0.7126846 0.05462296 0.5984645 0.8092906
##     mu[2] 0.5488468 0.05866899 0.4288502 0.6596422
##
## , , chains = chain:2
##
##          stats
## parameter      mean         sd      2.5%     97.5%
##     mu[1] 0.7141387 0.05733942 0.5964158 0.8106541
```

```
##     mu[2] 0.5406333 0.06275255 0.4124922 0.6538475
##
## , , chains = chain:3
##
##          stats
## parameter      mean          sd      2.5%     97.5%
##     mu[1] 0.7158314 0.05401437 0.6023117 0.8112631
##     mu[2] 0.5453233 0.06152640 0.4245339 0.6626200
```

Summary posterior s

```r
summary(post,pars="s", probs = c(0.025, 0.975))
```

```
## $summary
##          mean    se_mean       sd     2.5%     97.5%    n_eff      Rhat
## s[1] 4.283778 0.09980674 2.288285 1.489197  9.878245 525.6547 0.9990063
## s[2] 4.583519 0.09834979 2.290161 1.692722 10.407492 542.2320 1.0019213
##
## $c_summary
## , , chains = chain:1
##
##          stats
## parameter     mean       sd     2.5%    97.5%
##      s[1] 4.334745 2.397479 1.504329 10.94471
##      s[2] 4.709489 2.434941 1.812477 10.84938
##
## , , chains = chain:2
##
##          stats
## parameter     mean       sd     2.5%    97.5%
##      s[1] 4.238794 2.398169 1.401489 9.973716
##      s[2] 4.376029 2.055336 1.846785 8.901410
##
## , , chains = chain:3
##
##          stats
## parameter     mean       sd     2.5%     97.5%
##      s[1] 4.277796 2.055711 1.621444  9.372532
##      s[2] 4.665040 2.352976 1.561398 10.558125
```

Summary posterior theta

```r
summary(post,pars="theta", probs = c(0.025, 0.975))
```

```
## $summary
##                  mean      se_mean         sd       2.5%     97.5%    n_eff
## theta[1,1]  0.8011518 0.0019051367 0.08890084 0.59963107 0.9426752 2177.506
## theta[1,2]  0.6357775 0.0047123995 0.20085908 0.20663584 0.9631411 1816.765
## theta[2,1]  0.6847837 0.0027197246 0.12275271 0.41769390 0.8948484 2037.100
```

```
## theta[2,2]  0.7546449 0.0020048538 0.09547146 0.54543121 0.9092331 2267.680
## theta[3,1]  0.9416678 0.0013810682 0.05791858 0.79447888 0.9995839 1758.756
## theta[3,2]  0.4676376 0.0036260611 0.16695476 0.14656040 0.7865234 2119.959
## theta[4,1]  0.7331748 0.0031731446 0.14418307 0.40372562 0.9561518 2064.661
## theta[4,2]  0.6459348 0.0033683224 0.15759393 0.32730762 0.9135925 2189.030
## theta[5,1]  0.6372895 0.0020118689 0.09511360 0.44680328 0.8156651 2235.043
## theta[5,2]  0.5288782 0.0035951472 0.16516345 0.19339474 0.8363498 2110.545
## theta[6,1]  0.5246877 0.0032137738 0.12924968 0.27302073 0.7629462 1617.441
## theta[6,2]  0.3607349 0.0049096421 0.19724728 0.03189376 0.7461568 1614.070
## theta[7,1]  0.8779103 0.0027221906 0.10551959 0.61727065 0.9982784 1502.551
## theta[7,2]  0.5390356 0.0053623747 0.23645901 0.09211266 0.9425840 1944.453
## theta[8,1]  0.8530236 0.0018087404 0.07493474 0.68463115 0.9675258 1716.382
## theta[8,2]  0.4282120 0.0048857687 0.21000656 0.05500701 0.8227385 1847.566
## theta[9,1]  0.4736964 0.0042699694 0.18309843 0.12117737 0.8124097 1838.738
## theta[9,2]  0.2665213 0.0027866775 0.11415018 0.08470632 0.5112890 1677.953
## theta[10,1] 0.8120103 0.0023122250 0.10835882 0.57080518 0.9699907 2196.182
## theta[10,2] 0.7406714 0.0022501208 0.10558577 0.51434440 0.9180403 2201.908
## theta[11,1] 0.8733166 0.0017187686 0.07815260 0.69081480 0.9819048 2067.531
## theta[11,2] 0.4733249 0.0029392542 0.13729801 0.22428642 0.7410766 2181.997
## theta[12,1] 0.7878838 0.0029605783 0.12724127 0.49721951 0.9695200 1847.153
## theta[12,2] 0.6531249 0.0031828640 0.16037783 0.31528827 0.9241367 2538.940
## theta[13,1] 0.9094959 0.0021392354 0.08426407 0.68266612 0.9990666 1551.557
## theta[13,2] 0.7444226 0.0025258172 0.10790025 0.51084671 0.9275569 1824.908
## theta[14,1] 0.8656877 0.0009251707 0.04077290 0.77612437 0.9346396 1942.224
## theta[14,2] 0.7302343 0.0015557482 0.07754079 0.57045296 0.8667445 2484.173
## theta[15,1] 0.2443982 0.0028453725 0.10886946 0.06961487 0.4808473 1463.976
## theta[15,2] 0.3429326 0.0015903428 0.07294342 0.20211468 0.4881143 2103.734
## theta[16,1] 0.5535810 0.0045480905 0.19077431 0.16631420 0.8817348 1759.469
## theta[16,2] 0.3200352 0.0033958097 0.15065185 0.07509564 0.6321765 1968.168
## theta[17,1] 0.8352206 0.0013641115 0.05719926 0.71292150 0.9305547 1758.252
## theta[17,2] 0.8421044 0.0018442236 0.07731340 0.66806343 0.9640888 1757.447
## theta[18,1] 0.8491566 0.0020612747 0.09524105 0.61969130 0.9807312 2134.896
## theta[18,2] 0.6555121 0.0021416245 0.09970342 0.45107767 0.8392252 2167.372
## theta[19,1] 0.5836034 0.0027607704 0.12281490 0.35025923 0.8125010 1978.981
## theta[19,2] 0.2784447 0.0015796673 0.07497942 0.14427609 0.4344253 2252.957
##                  Rhat
## theta[1,1]  0.9992144
## theta[1,2]  0.9995102
## theta[2,1]  1.0006526
## theta[2,2]  0.9987669
## theta[3,1]  1.0006912
## theta[3,2]  1.0000179
## theta[4,1]  0.9996046
## theta[4,2]  1.0001400
## theta[5,1]  0.9987670
## theta[5,2]  0.9989549
## theta[6,1]  0.9991872
## theta[6,2]  1.0000211
## theta[7,1]  0.9986273
## theta[7,2]  1.0004417
## theta[8,1]  0.9987411
## theta[8,2]  0.9987843
## theta[9,1]  1.0004452
## theta[9,2]  0.9989443
```

```
## theta[10,1] 0.9994640
## theta[10,2] 0.9995846
## theta[11,1] 0.9990509
## theta[11,2] 0.9988052
## theta[12,1] 0.9988894
## theta[12,2] 1.0000712
## theta[13,1] 0.9990890
## theta[13,2] 0.9999312
## theta[14,1] 0.9996330
## theta[14,2] 0.9984477
## theta[15,1] 0.9986508
## theta[15,2] 0.9989929
## theta[16,1] 0.9997429
## theta[16,2] 1.0000763
## theta[17,1] 0.9989218
## theta[17,2] 1.0005970
## theta[18,1] 0.9998294
## theta[18,2] 0.9997319
## theta[19,1] 0.9997291
## theta[19,2] 0.9989122
##
## $c_summary
## , , chains = chain:1
##
##              stats
## parameter          mean         sd       2.5%     97.5%
##   theta[1,1]  0.8004219 0.08466408 0.63236930 0.9405299
##   theta[1,2]  0.6282106 0.19023120 0.25471000 0.9358053
##   theta[2,1]  0.6906326 0.12428726 0.40271177 0.8983860
##   theta[2,2]  0.7537284 0.09571884 0.53989308 0.9178754
##   theta[3,1]  0.9425314 0.05778322 0.79406418 0.9997254
##   theta[3,2]  0.4782940 0.15331128 0.17228330 0.7669835
##   theta[4,1]  0.7248889 0.13761539 0.41576687 0.9438182
##   theta[4,2]  0.6537649 0.15105003 0.35542372 0.9076937
##   theta[5,1]  0.6394476 0.09585875 0.44996573 0.8107028
##   theta[5,2]  0.5322945 0.16223036 0.19519360 0.8396490
##   theta[6,1]  0.5286964 0.12477912 0.30016376 0.7746909
##   theta[6,2]  0.3667654 0.19461177 0.03906740 0.7700864
##   theta[7,1]  0.8787127 0.10661234 0.62167442 0.9987354
##   theta[7,2]  0.5457211 0.24213577 0.09211266 0.9385242
##   theta[8,1]  0.8548675 0.07723492 0.68851603 0.9708916
##   theta[8,2]  0.4339671 0.20599374 0.08094946 0.8014447
##   theta[9,1]  0.4777807 0.18567470 0.12297728 0.8212911
##   theta[9,2]  0.2696495 0.10993270 0.09643735 0.4982771
##   theta[10,1] 0.8090893 0.10923419 0.57097539 0.9705844
##   theta[10,2] 0.7357285 0.11194874 0.49070074 0.9282757
##   theta[11,1] 0.8709721 0.07573396 0.69081480 0.9782374
##   theta[11,2] 0.4733877 0.14178651 0.22376969 0.7426315
##   theta[12,1] 0.7857970 0.12798693 0.51193392 0.9652420
##   theta[12,2] 0.6572152 0.15962042 0.32780536 0.9271449
##   theta[13,1] 0.9090078 0.08453208 0.67586929 0.9984700
##   theta[13,2] 0.7396511 0.10482085 0.51228110 0.9179300
##   theta[14,1] 0.8678545 0.03984054 0.77918461 0.9378774
##   theta[14,2] 0.7312917 0.07676419 0.57260726 0.8666420
```

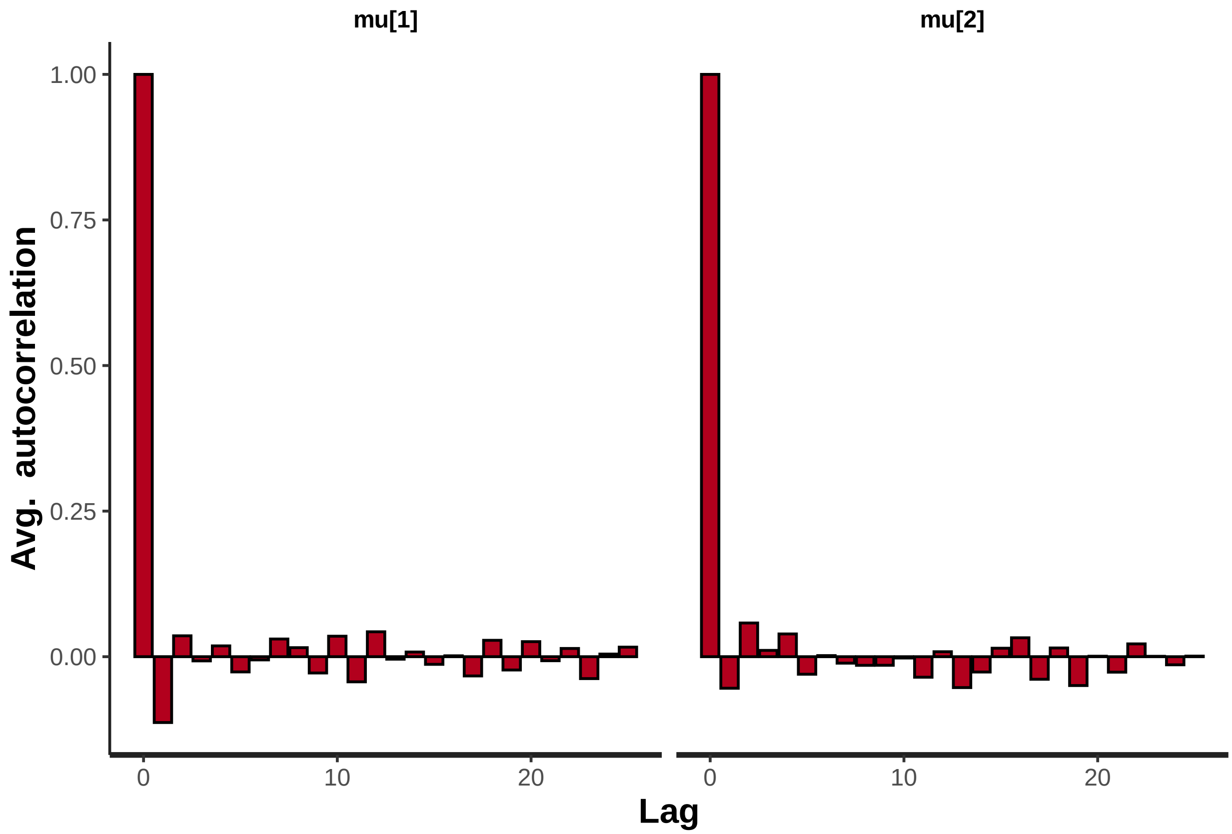
```
##   theta[15,1] 0.2449154 0.11104510 0.06886355 0.4996136
##   theta[15,2] 0.3441113 0.07141225 0.21273084 0.4840339
##   theta[16,1] 0.5488141 0.18626575 0.17110905 0.8790792
##   theta[16,2] 0.3266768 0.15583073 0.08025534 0.6488062
##   theta[17,1] 0.8337077 0.05721207 0.71266542 0.9240149
##   theta[17,2] 0.8412862 0.08078597 0.66298692 0.9624221
##   theta[18,1] 0.8500830 0.08856821 0.63729027 0.9780682
##   theta[18,2] 0.6543420 0.09236392 0.47012444 0.8206576
##   theta[19,1] 0.5910510 0.11896747 0.36235509 0.8112942
##   theta[19,2] 0.2798204 0.07460302 0.15178840 0.4281938
##
## , , chains = chain:2
##
##              stats
## parameter          mean         sd       2.5%     97.5%
##   theta[1,1]  0.8019839 0.09096976 0.59861117 0.9437209
##   theta[1,2]  0.6383590 0.20051922 0.18379167 0.9687188
##   theta[2,1]  0.6773196 0.12450615 0.41662901 0.8886485
##   theta[2,2]  0.7531699 0.09609424 0.55327803 0.9097354
##   theta[3,1]  0.9431082 0.06159703 0.78207931 0.9996370
##   theta[3,2]  0.4581838 0.17210123 0.13328131 0.7853899
##   theta[4,1]  0.7354178 0.14959803 0.39752275 0.9567756
##   theta[4,2]  0.6465300 0.15740926 0.33532786 0.8992561
##   theta[5,1]  0.6383093 0.09296702 0.44751059 0.8086461
##   theta[5,2]  0.5231393 0.16430487 0.18442763 0.8265172
##   theta[6,1]  0.5267009 0.12780845 0.28242591 0.7613378
##   theta[6,2]  0.3583247 0.19425781 0.03472556 0.7410369
##   theta[7,1]  0.8776454 0.11200149 0.59052684 0.9988049
##   theta[7,2]  0.5335763 0.22958998 0.09789591 0.9372097
##   theta[8,1]  0.8528196 0.07258427 0.68543181 0.9636834
##   theta[8,2]  0.4235557 0.21449894 0.04958109 0.8330921
##   theta[9,1]  0.4676999 0.18391015 0.11896412 0.8111706
##   theta[9,2]  0.2657019 0.11203230 0.09386885 0.5129865
##   theta[10,1] 0.8157037 0.10673231 0.55768166 0.9702394
##   theta[10,2] 0.7431583 0.10290441 0.54315749 0.9126584
##   theta[11,1] 0.8739874 0.07972261 0.69314125 0.9838211
##   theta[11,2] 0.4730729 0.13790053 0.23051007 0.7389787
##   theta[12,1] 0.7865567 0.12691939 0.48439235 0.9723119
##   theta[12,2] 0.6574900 0.14949730 0.33649105 0.9108066
##   theta[13,1] 0.9109242 0.08372002 0.70112738 0.9993320
##   theta[13,2] 0.7480887 0.10954745 0.51018454 0.9271594
##   theta[14,1] 0.8659416 0.04135978 0.77299488 0.9346396
##   theta[14,2] 0.7300497 0.08202122 0.55916187 0.8714267
##   theta[15,1] 0.2439225 0.11115967 0.07260715 0.4790359
##   theta[15,2] 0.3399032 0.07657893 0.19262364 0.4952021
##   theta[16,1] 0.5654843 0.19377956 0.14411141 0.8843607
##   theta[16,2] 0.3089844 0.14626713 0.07009163 0.6247978
##   theta[17,1] 0.8352669 0.05582280 0.71634387 0.9287192
##   theta[17,2] 0.8421287 0.07894501 0.65710829 0.9690599
##   theta[18,1] 0.8542287 0.09282336 0.63812063 0.9842958
##   theta[18,2] 0.6556692 0.10424916 0.44401294 0.8508035
##   theta[19,1] 0.5768789 0.12569228 0.33205944 0.8120694
##   theta[19,2] 0.2795096 0.07323066 0.14501057 0.4321477
##
```

```
## , , chains = chain:3
##
##               stats
## parameter          mean         sd       2.5%     97.5%
##   theta[1,1]  0.8010494 0.09108795 0.58236287 0.9395111
##   theta[1,2]  0.6407630 0.21144737 0.20106144 0.9767207
##   theta[2,1]  0.6863987 0.11926130 0.45665841 0.8992244
##   theta[2,2]  0.7570365 0.09474165 0.54561286 0.9065892
##   theta[3,1]  0.9393638 0.05418385 0.80578102 0.9993057
##   theta[3,2]  0.4664351 0.17437151 0.11476851 0.7996905
##   theta[4,1]  0.7392178 0.14498829 0.40502057 0.9600977
##   theta[4,2]  0.6375095 0.16395237 0.29125087 0.9249742
##   theta[5,1]  0.6341117 0.09658395 0.44580089 0.8229042
##   theta[5,2]  0.5312007 0.16906020 0.22248966 0.8388151
##   theta[6,1]  0.5186658 0.13498914 0.24374762 0.7583552
##   theta[6,2]  0.3571147 0.20300289 0.02986207 0.7378124
##   theta[7,1]  0.8773727 0.09765373 0.63448931 0.9948094
##   theta[7,2]  0.5378093 0.23779245 0.08336636 0.9530085
##   theta[8,1]  0.8513838 0.07502199 0.67445347 0.9667802
##   theta[8,2]  0.4271133 0.20972765 0.05049515 0.8304757
##   theta[9,1]  0.4756084 0.17987505 0.13596927 0.7842312
##   theta[9,2]  0.2642125 0.12037758 0.05963459 0.5192119
##   theta[10,1] 0.8112380 0.10920280 0.57458013 0.9684317
##   theta[10,2] 0.7431272 0.10164245 0.52146453 0.9168713
##   theta[11,1] 0.8749903 0.07904351 0.69012020 0.9824299
##   theta[11,2] 0.4735142 0.13231672 0.22439692 0.7298492
##   theta[12,1] 0.7912978 0.12699968 0.49822023 0.9711223
##   theta[12,2] 0.6446697 0.17126185 0.28257622 0.9246829
##   theta[13,1] 0.9085556 0.08468694 0.66334821 0.9991315
##   theta[13,2] 0.7455280 0.10930917 0.52029202 0.9283731
##   theta[14,1] 0.8632671 0.04105375 0.77477656 0.9295629
##   theta[14,2] 0.7293614 0.07375420 0.58645097 0.8625547
##   theta[15,1] 0.2443569 0.10448542 0.06323217 0.4637413
##   theta[15,2] 0.3447833 0.07074950 0.21610450 0.4885594
##   theta[16,1] 0.5464445 0.19201343 0.18288076 0.8774515
##   theta[16,2] 0.3244443 0.14938397 0.07522360 0.6103852
##   theta[17,1] 0.8366873 0.05860480 0.70973293 0.9358573
##   theta[17,2] 0.8428984 0.07208363 0.69663076 0.9637126
##   theta[18,1] 0.8431580 0.10356512 0.56225551 0.9805875
##   theta[18,2] 0.6565250 0.10227781 0.44647817 0.8508017
##   theta[19,1] 0.5828804 0.12352221 0.35206665 0.8163278
##   theta[19,2] 0.2760043 0.07714229 0.14137549 0.4366584
```

Autocorrelation mu

```r
stan_ac(post,"mu",inc_warmup = FALSE, lags = 25, fun.data = "mean_se")
```

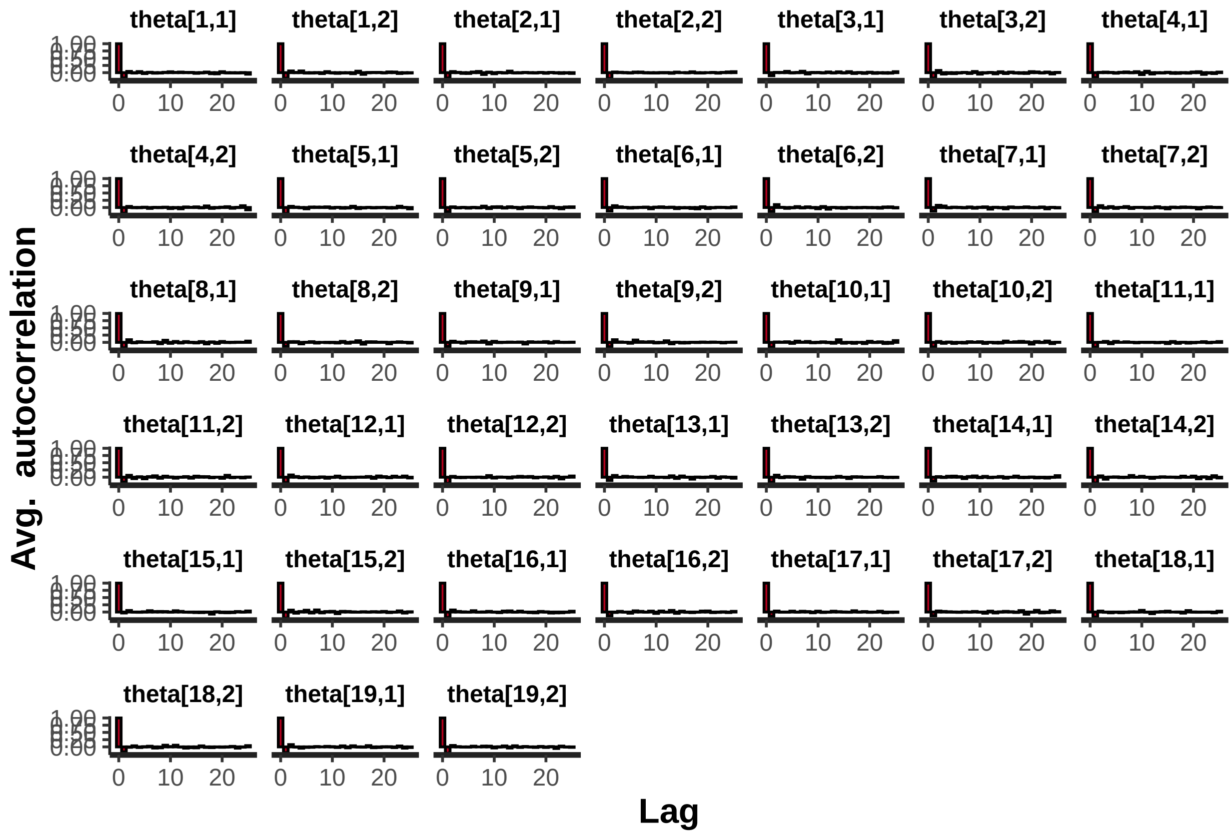
```
## Warning: Ignoring unknown parameters: fun.y
```



Autocorrelation Theta

```
stan_ac(post,"theta",inc_warmup = FALSE, lags = 25, fun.data = "mean_se")
```

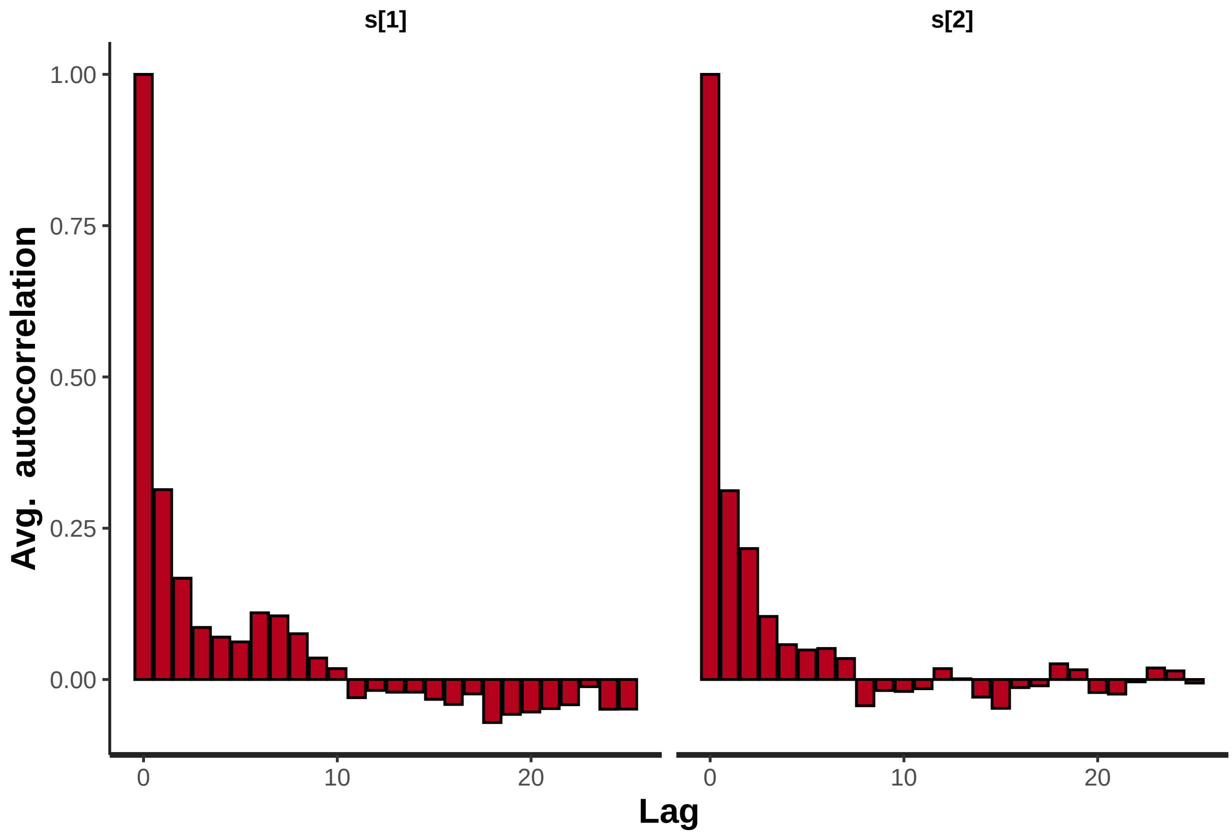
```
## Warning: Ignoring unknown parameters: fun.y
```



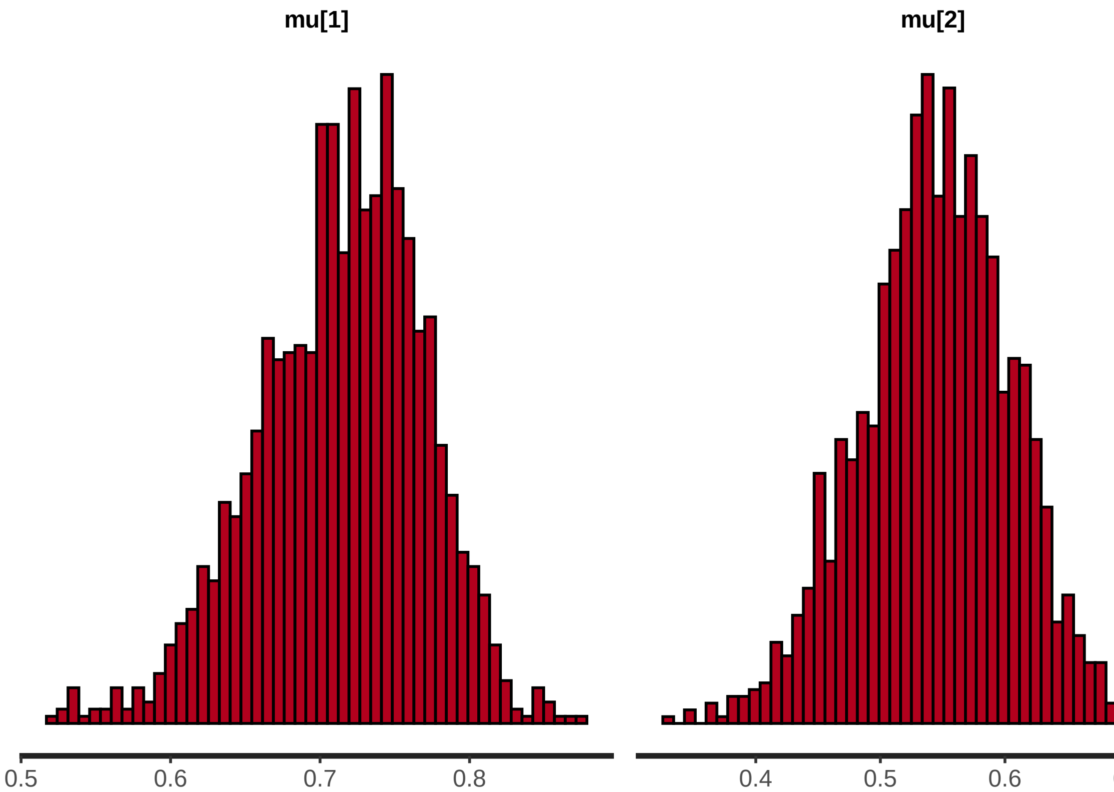
Autocorrelation s

```
stan_ac(post,"s",inc_warmup = FALSE, lags = 25, fun.data = "mean_se")
```

```
## Warning: Ignoring unknown parameters: fun.y
```

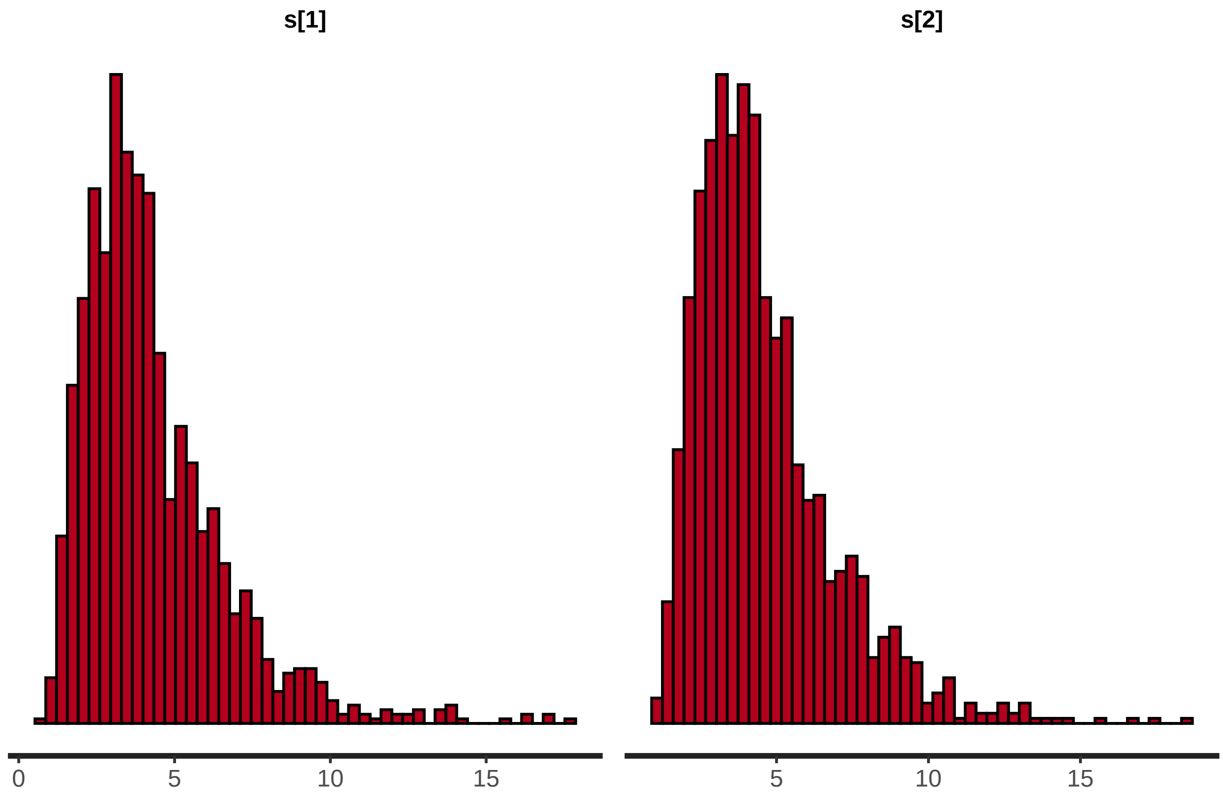



```r
#plot histogram of the posterior of lambda (approximation for density function)
plot(post, plotfun = "hist", pars = "mu",bins=50)
```

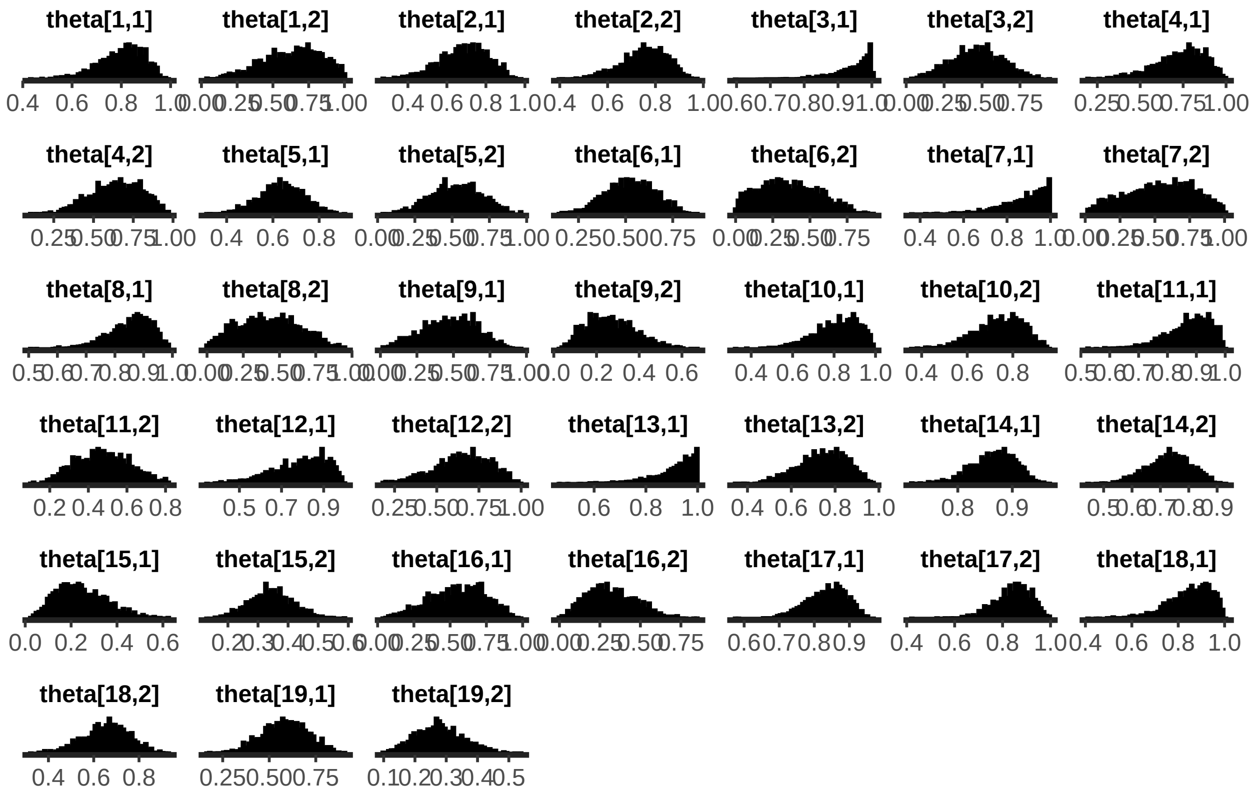


Visualize the posterior

```
#plot histogram of the posterior of lambda (approximation for density function)
plot(post, plotfun = "hist", pars = "s",bins=50)
```



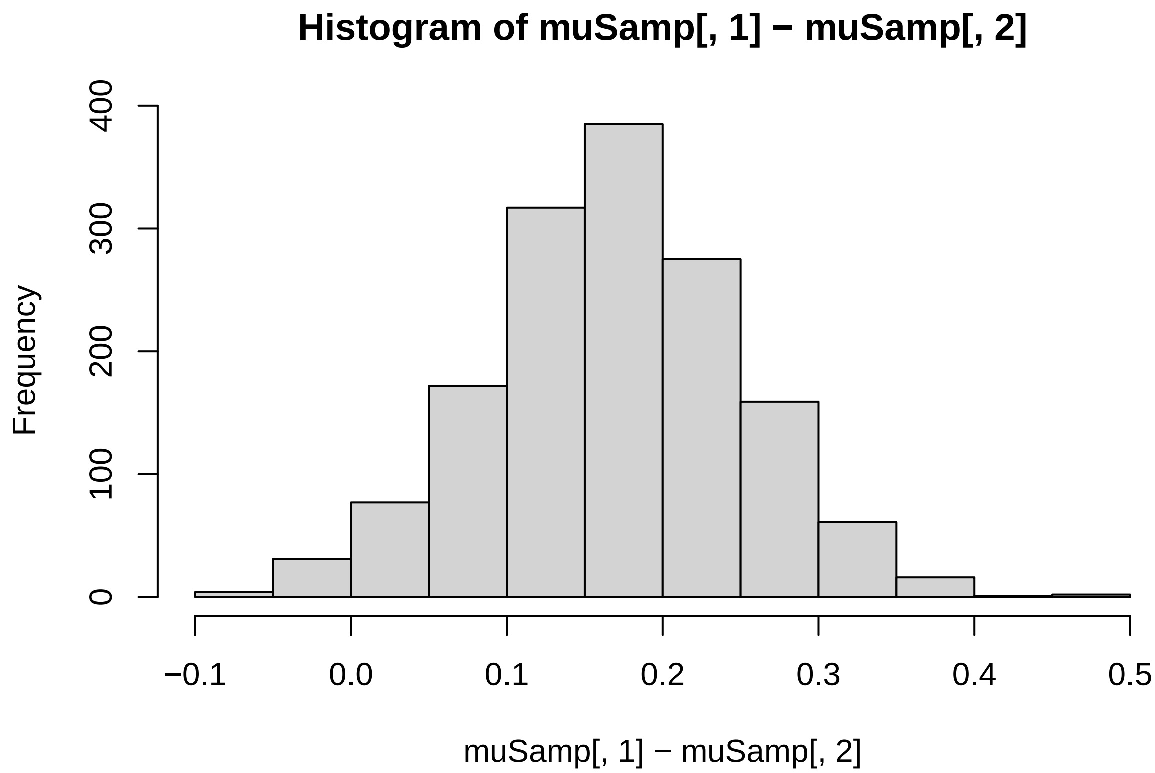

```r
plot(post, plotfun = "hist", pars = "theta",bins=50)
```



Part 2

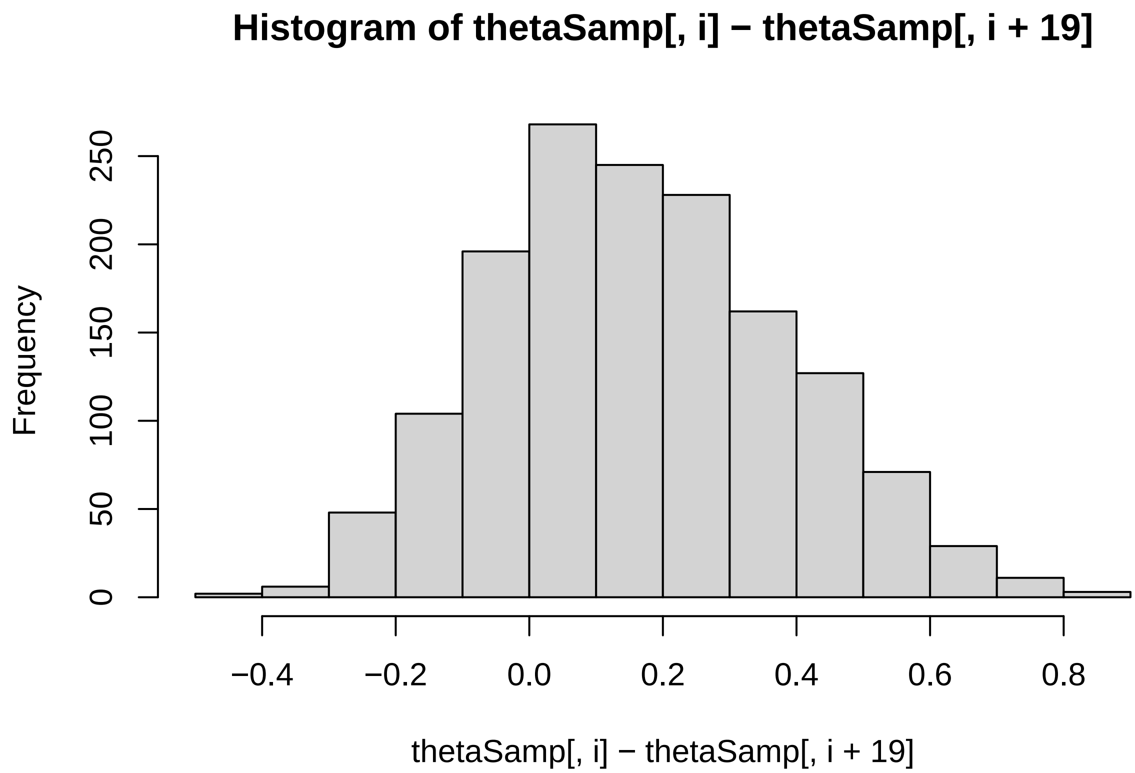
Posterior $\delta\mu$

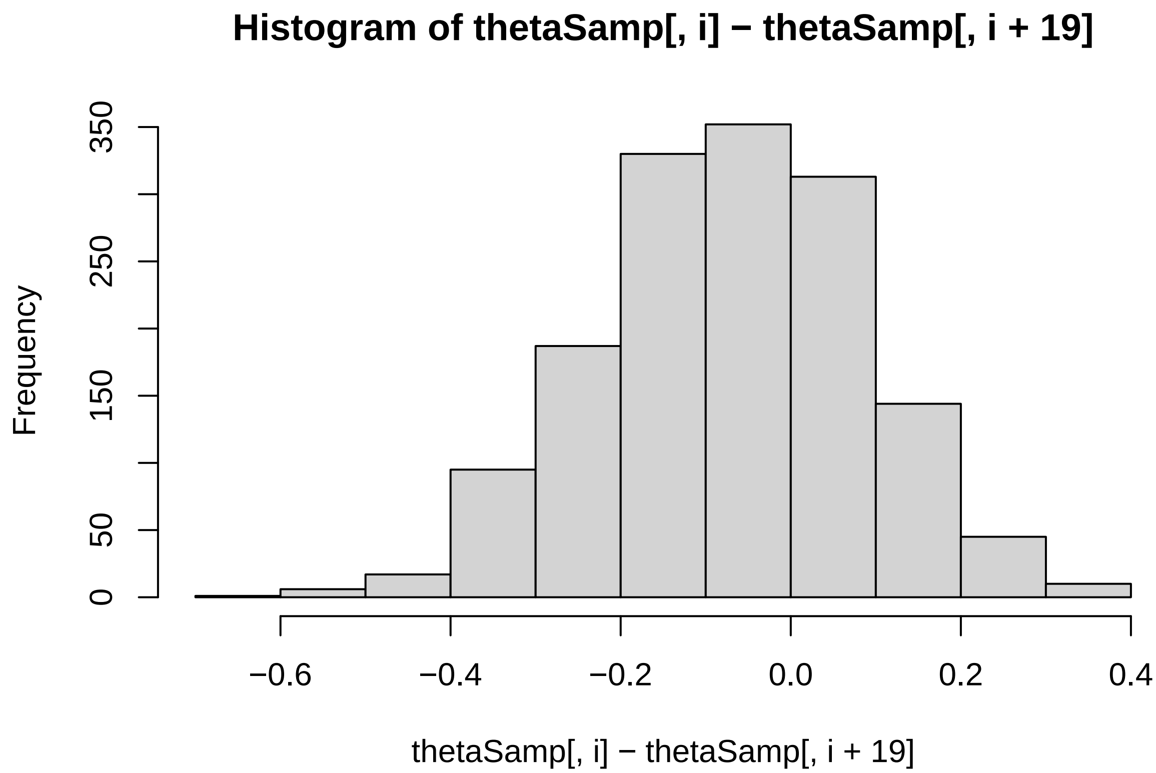
```
muSamp=as.matrix(post, pars ="mu")
#muSamp
hist(muSamp[,1]-muSamp[,2])
```

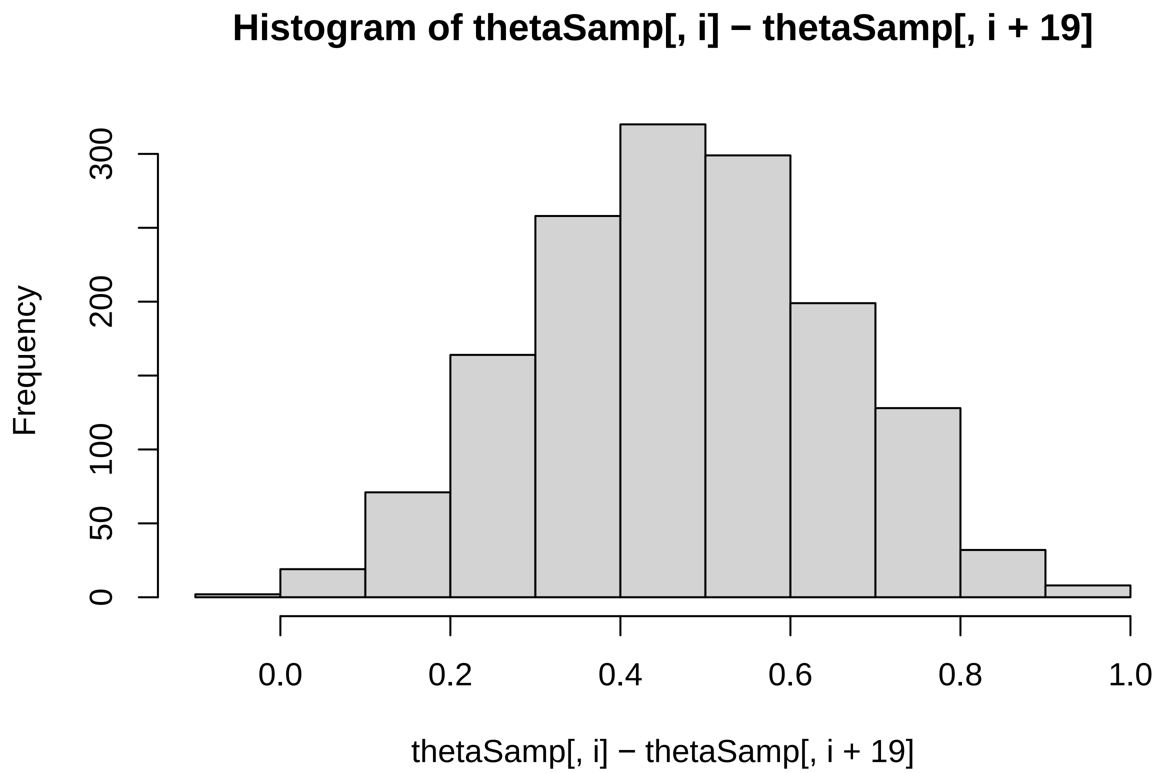


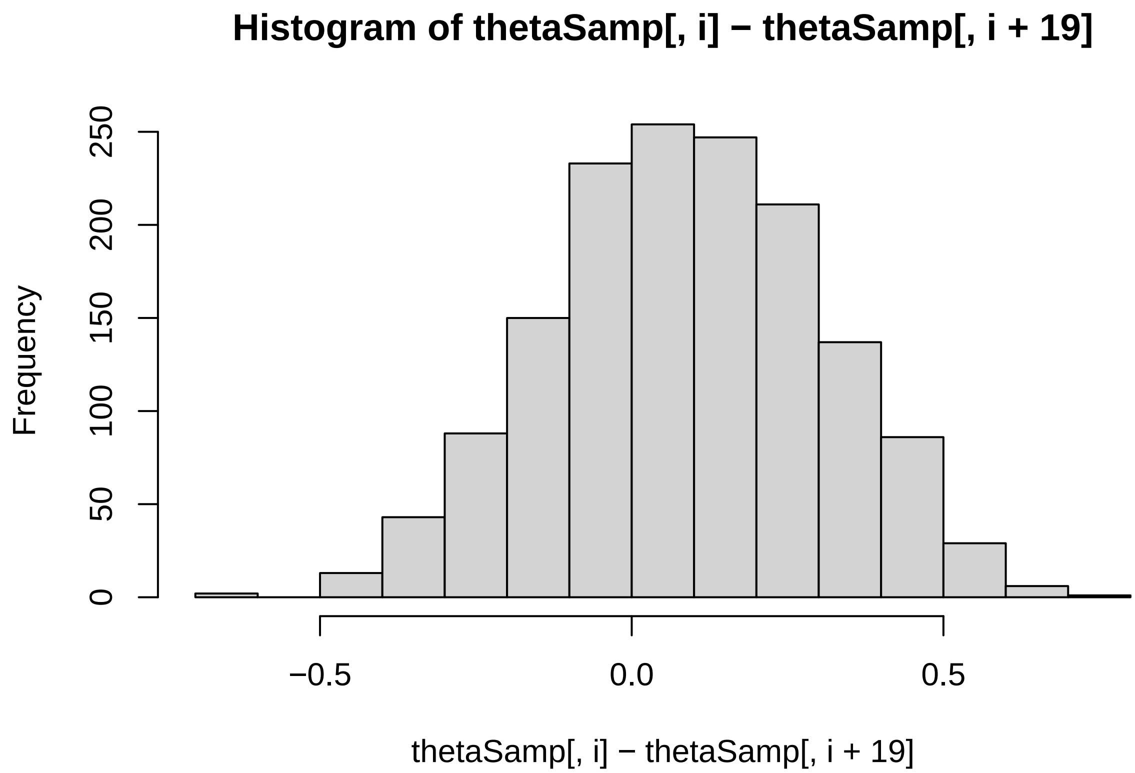
Posterior $\delta\phi$

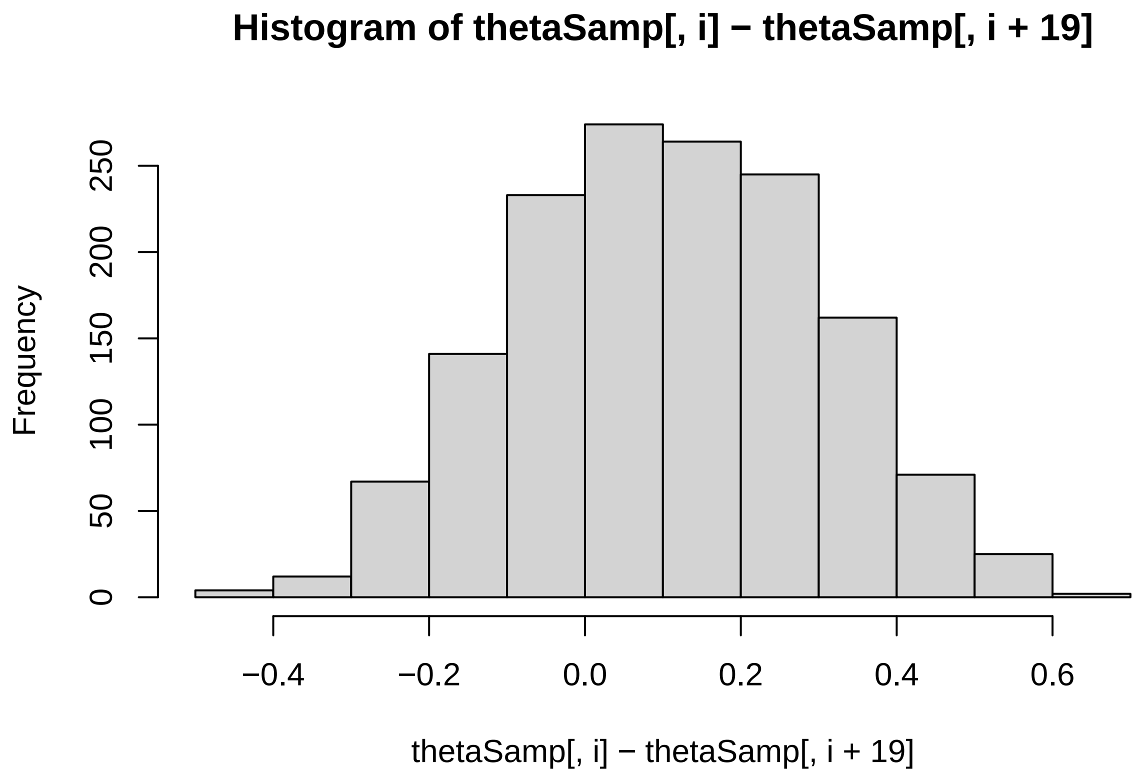
```
thetaSamp=as.matrix(post, pars ="theta")
#thetaSamp
for (i in 1:19){
  hist(thetaSamp[,i] - thetaSamp[,i+19])
}
```

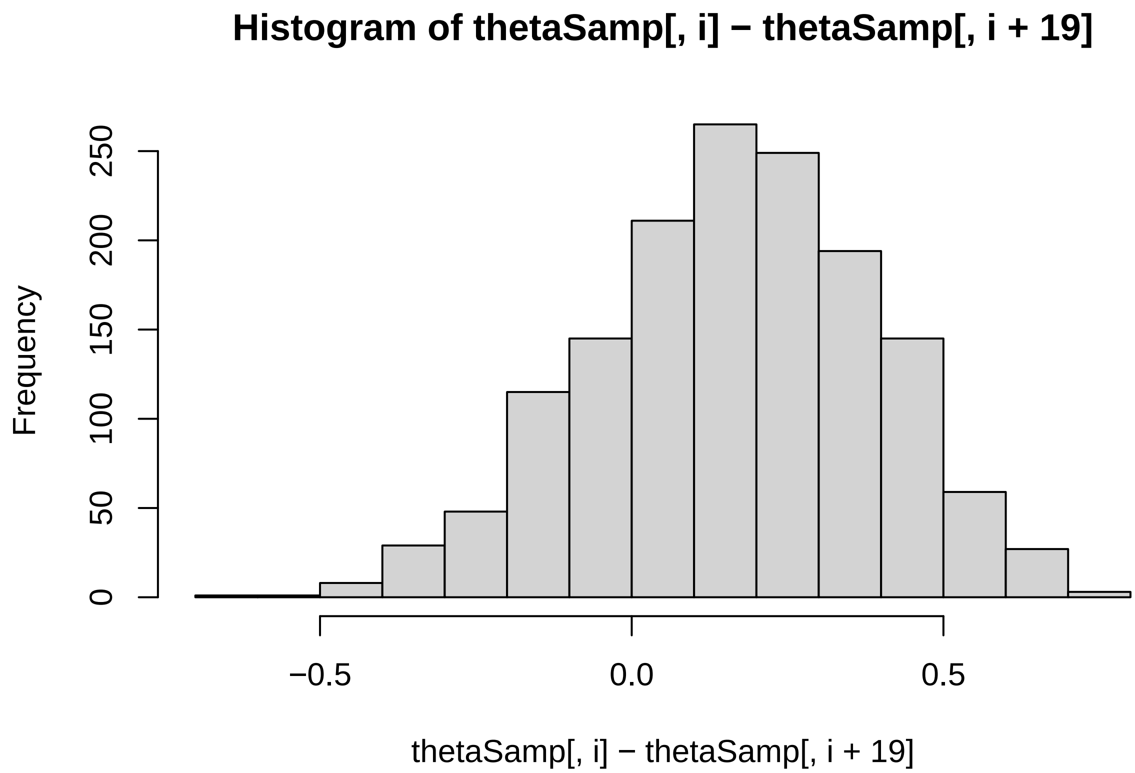

Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
100
150
200
250
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
thetaSamp[, i] − thetaSamp[, i + 19]

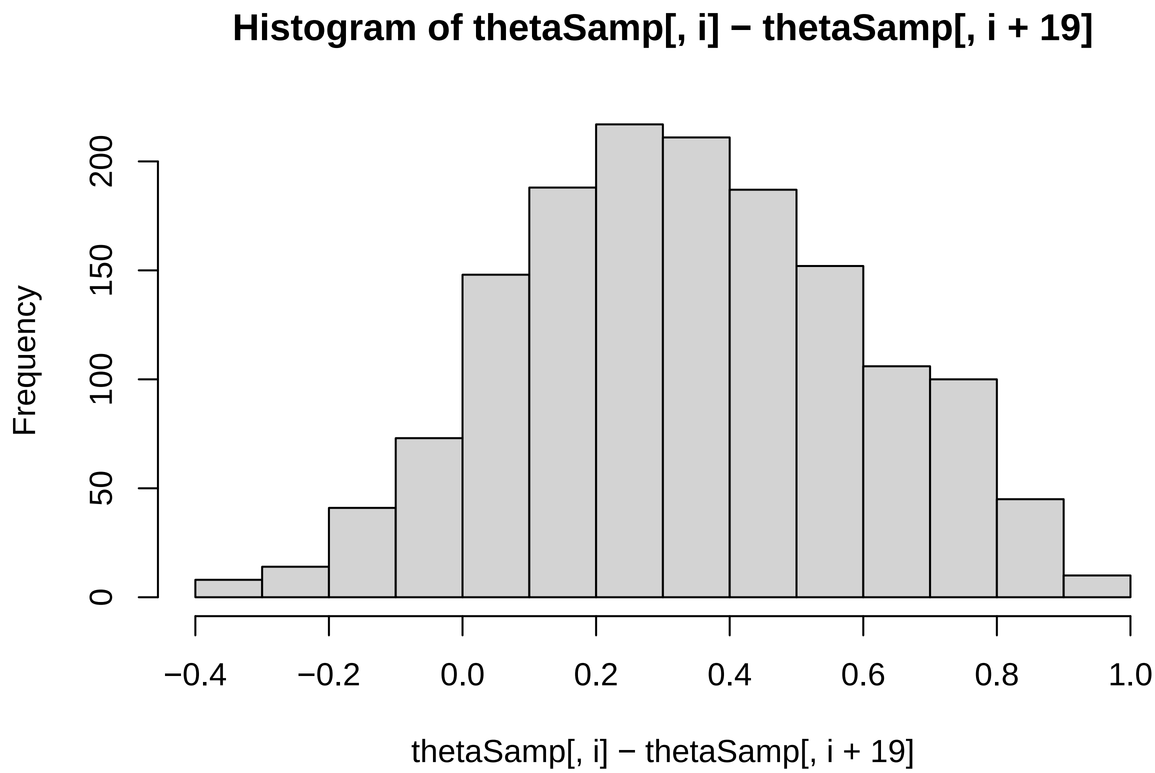

Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
150
250
350
−0.6
−0.4
−0.2
0.0
0.2
0.4
thetaSamp[, i] − thetaSamp[, i + 19]

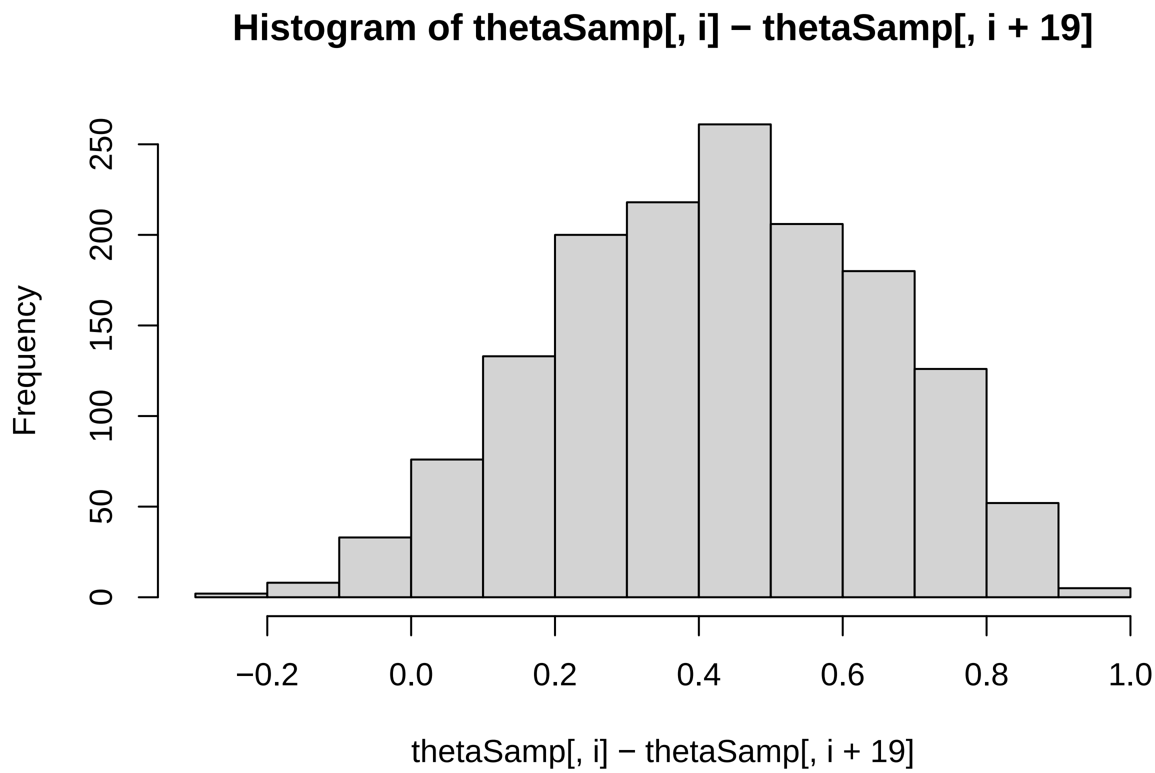

Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
100
200
300
0.0
0.2
0.4
0.6
0.8
1.0
thetaSamp[, i] − thetaSamp[, i + 19]


Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
100
150
200
250
−0.5
0.0
0.5
thetaSamp[, i] − thetaSamp[, i + 19]

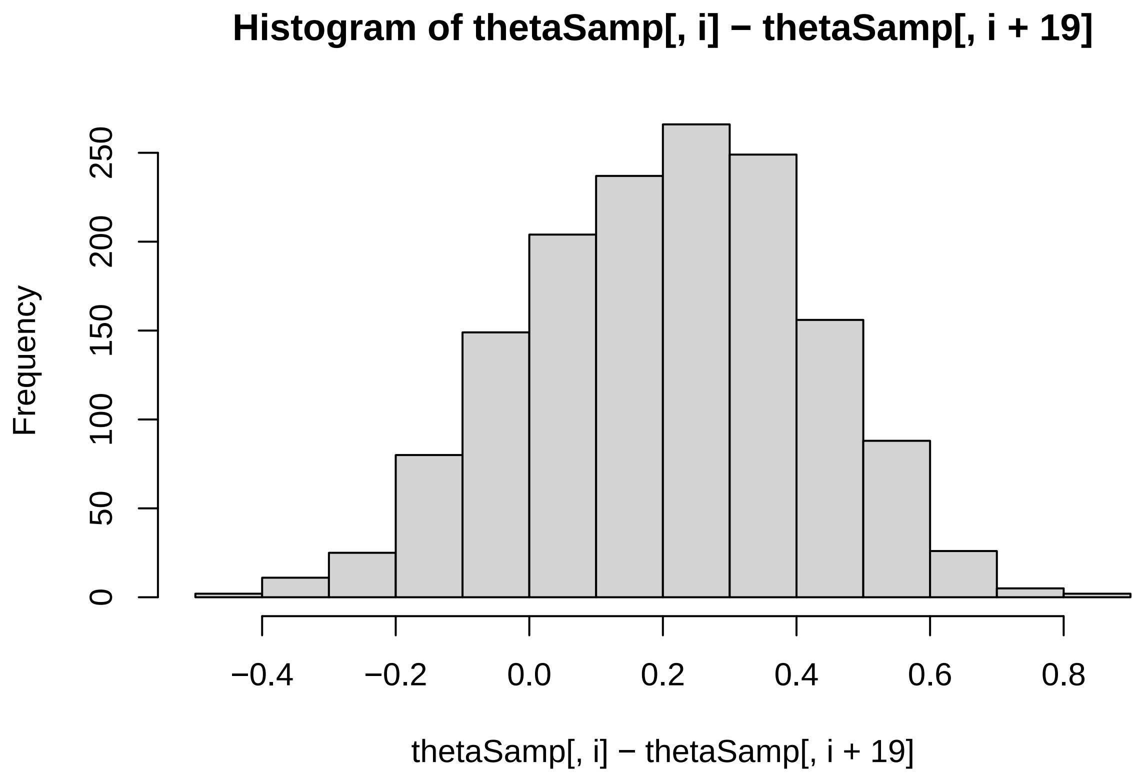

Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
100
150
200
250
−0.4
−0.2
0.0
0.2
0.4
0.6
thetaSamp[, i] − thetaSamp[, i + 19]

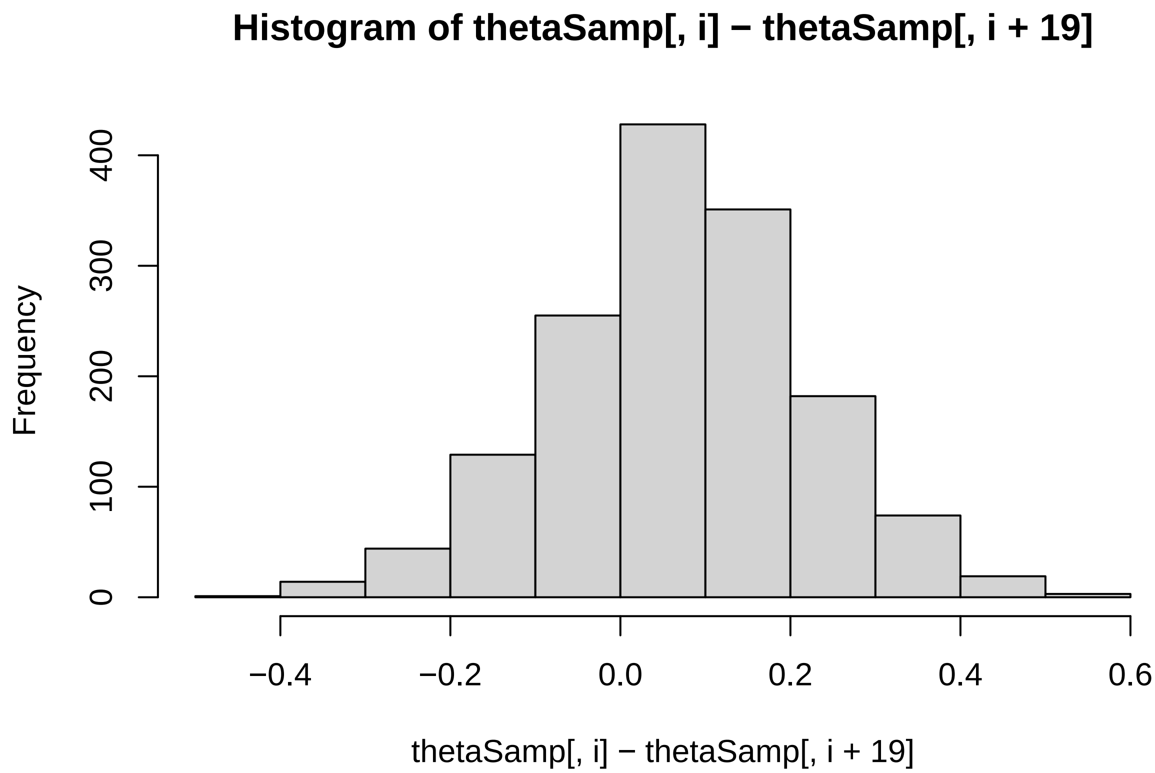

Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
100
150
200
250
−0.5
0.0
0.5
thetaSamp[, i] − thetaSamp[, i + 19]

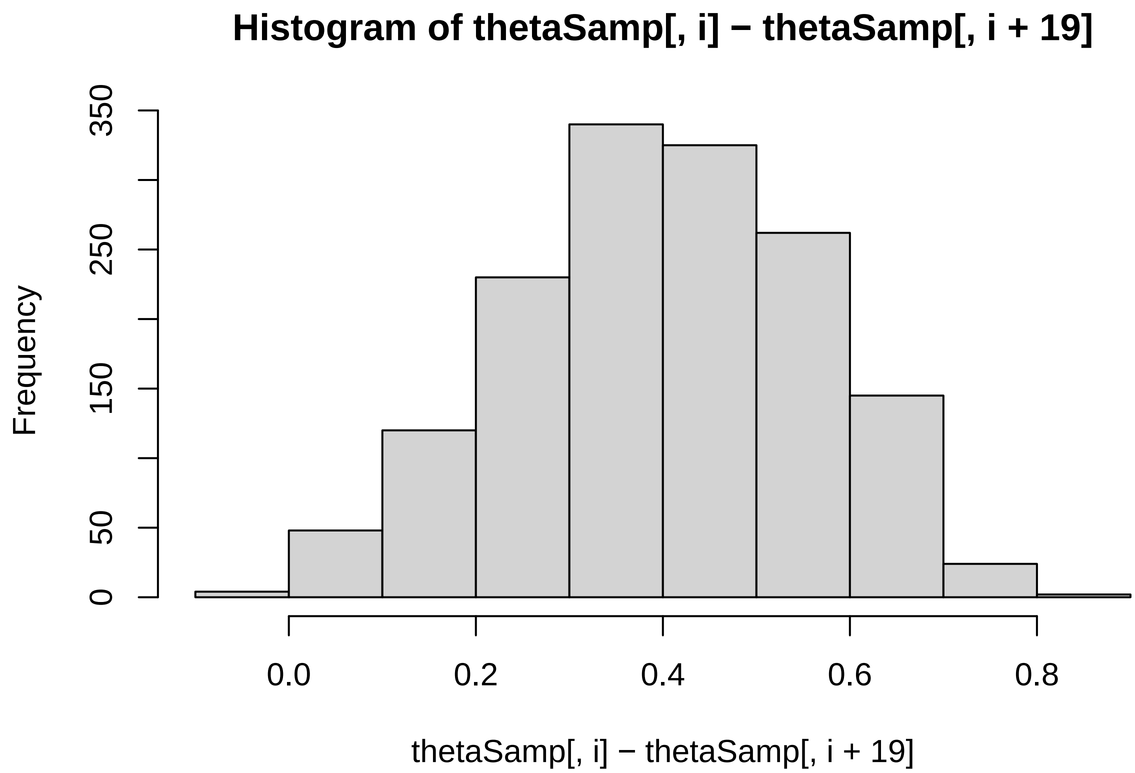

Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
100
150
200
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
thetaSamp[, i] − thetaSamp[, i + 19]

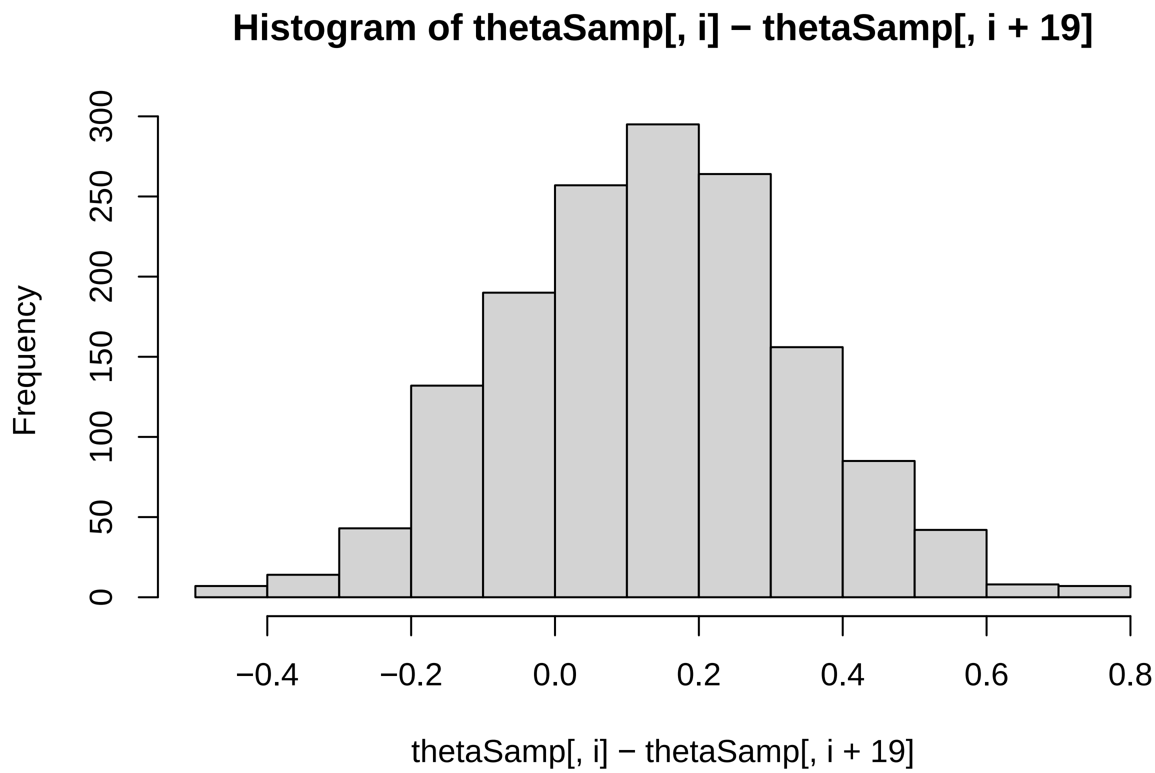


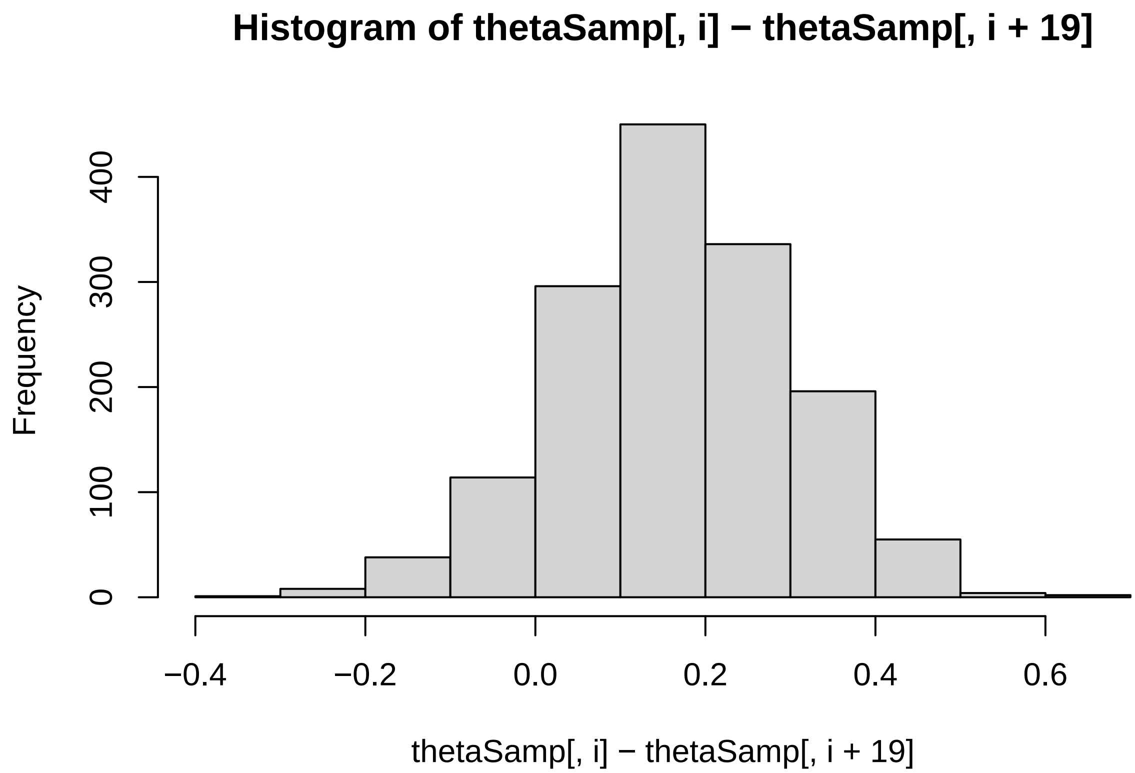
Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
100
150
200
250
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
thetaSamp[, i] − thetaSamp[, i + 19]


Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
100
150
200
250
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
thetaSamp[, i] − thetaSamp[, i + 19]

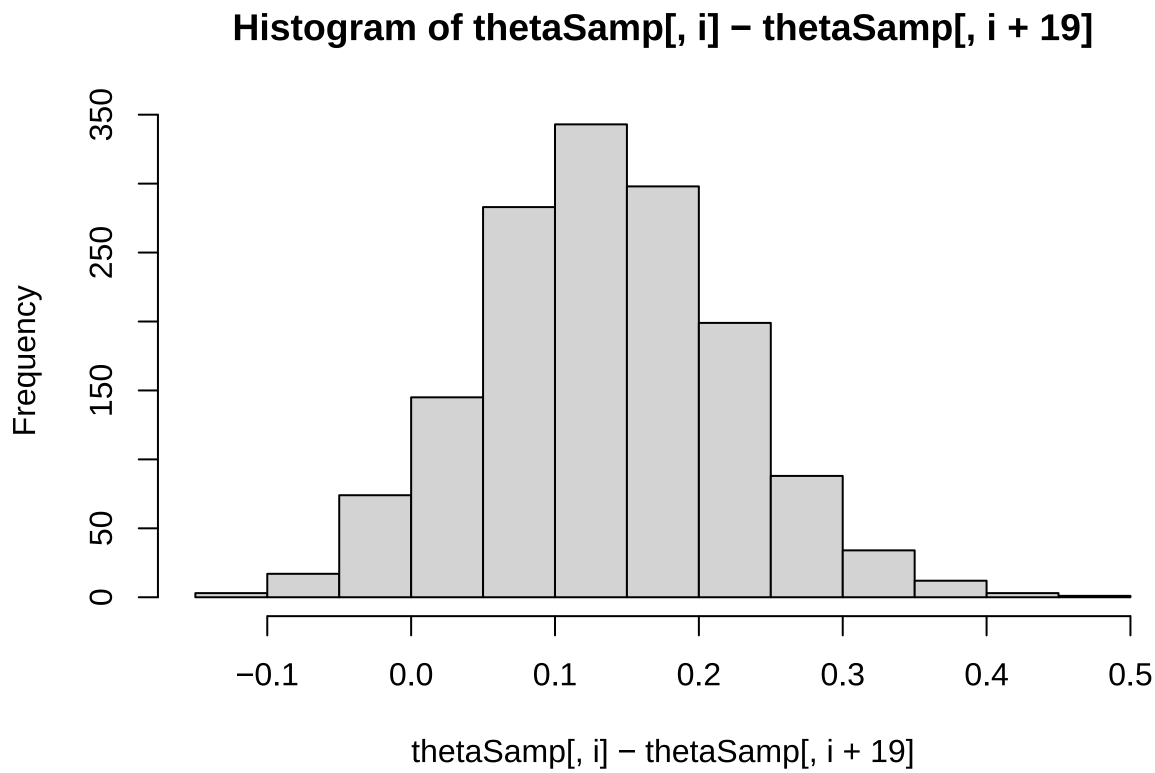

Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
100
200
300
400
−0.4
−0.2
0.0
0.2
0.4
0.6
thetaSamp[, i] − thetaSamp[, i + 19]

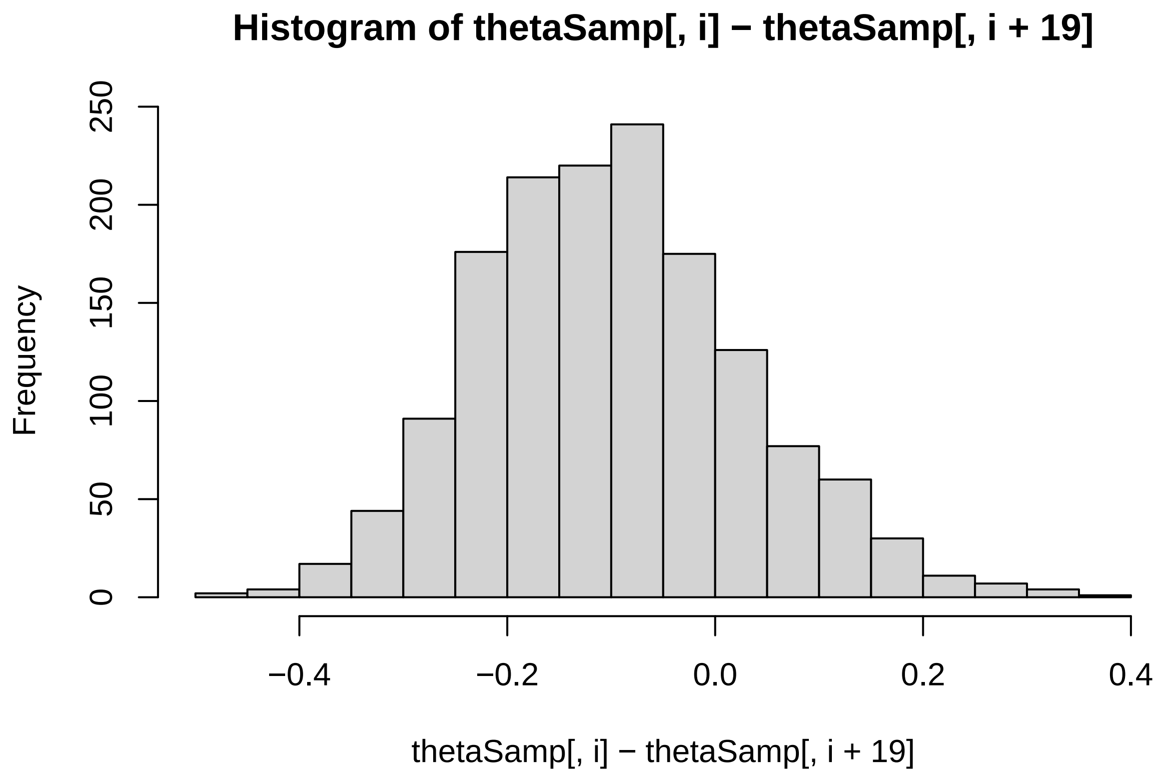

Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
150
250
350
0.0
0.2
0.4
0.6
0.8
thetaSamp[, i] − thetaSamp[, i + 19]


Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
100
150
200
250
300
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
thetaSamp[, i] − thetaSamp[, i + 19]

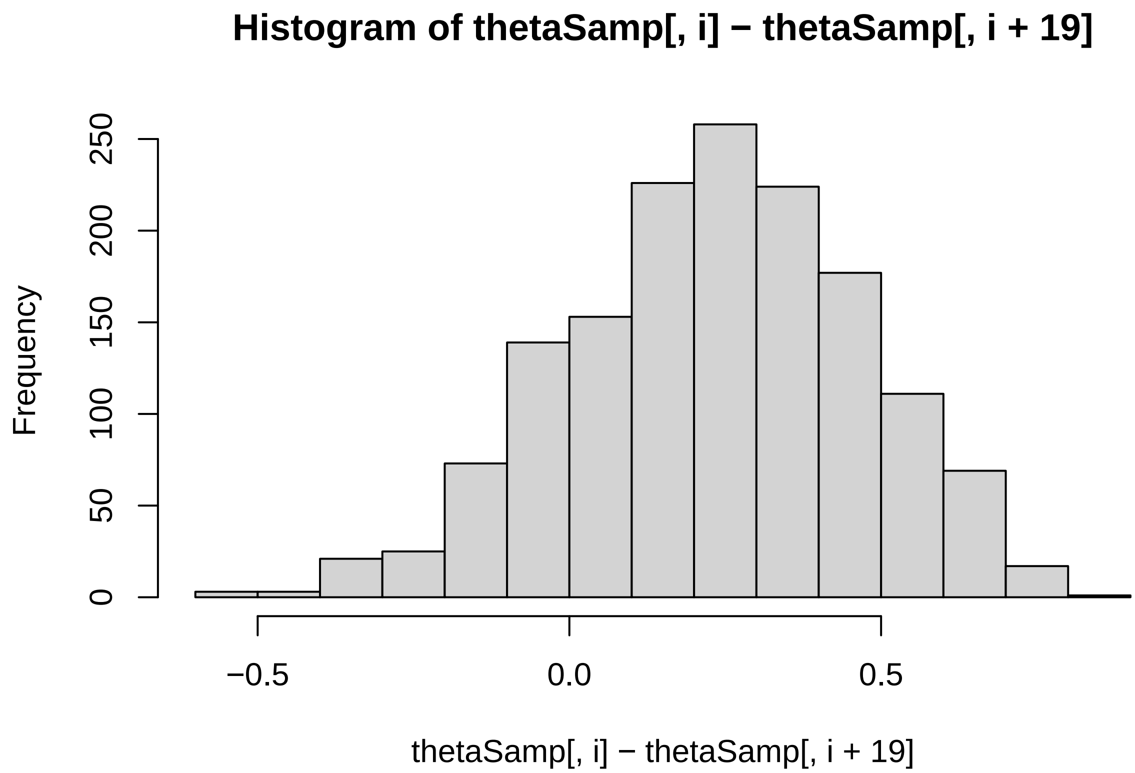


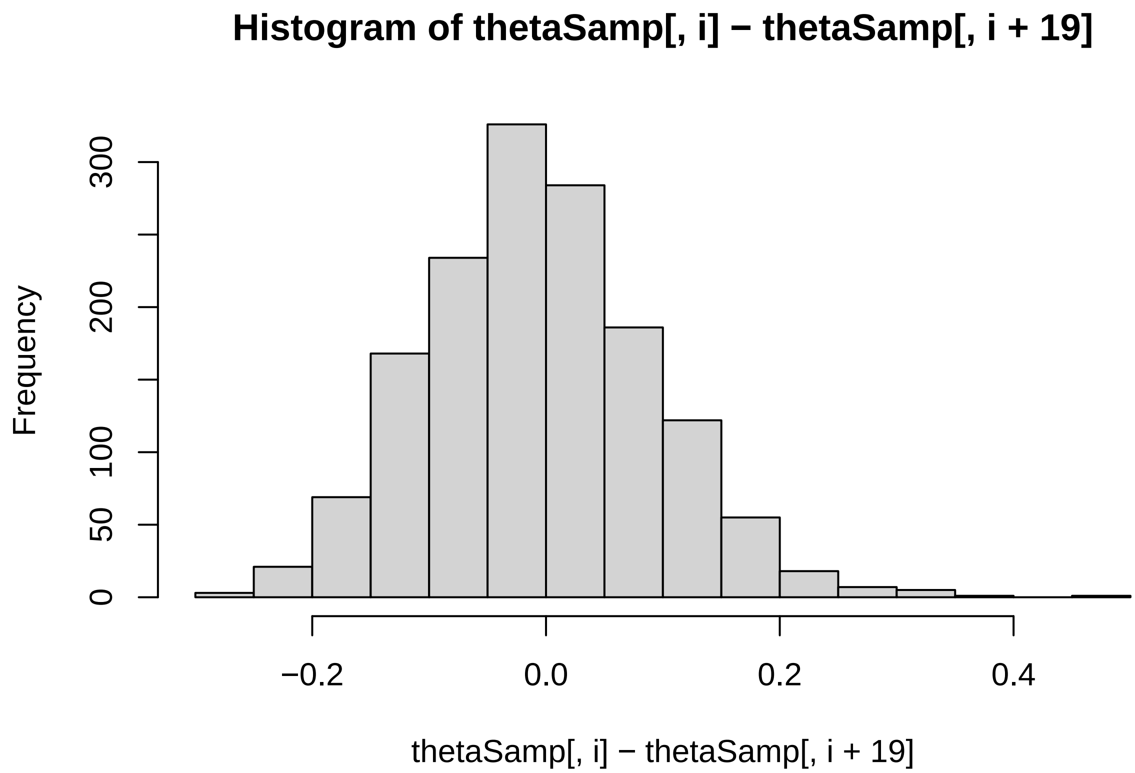
Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
100
200
300
400
−0.4
−0.2
0.0
0.2
0.4
0.6
thetaSamp[, i] − thetaSamp[, i + 19]

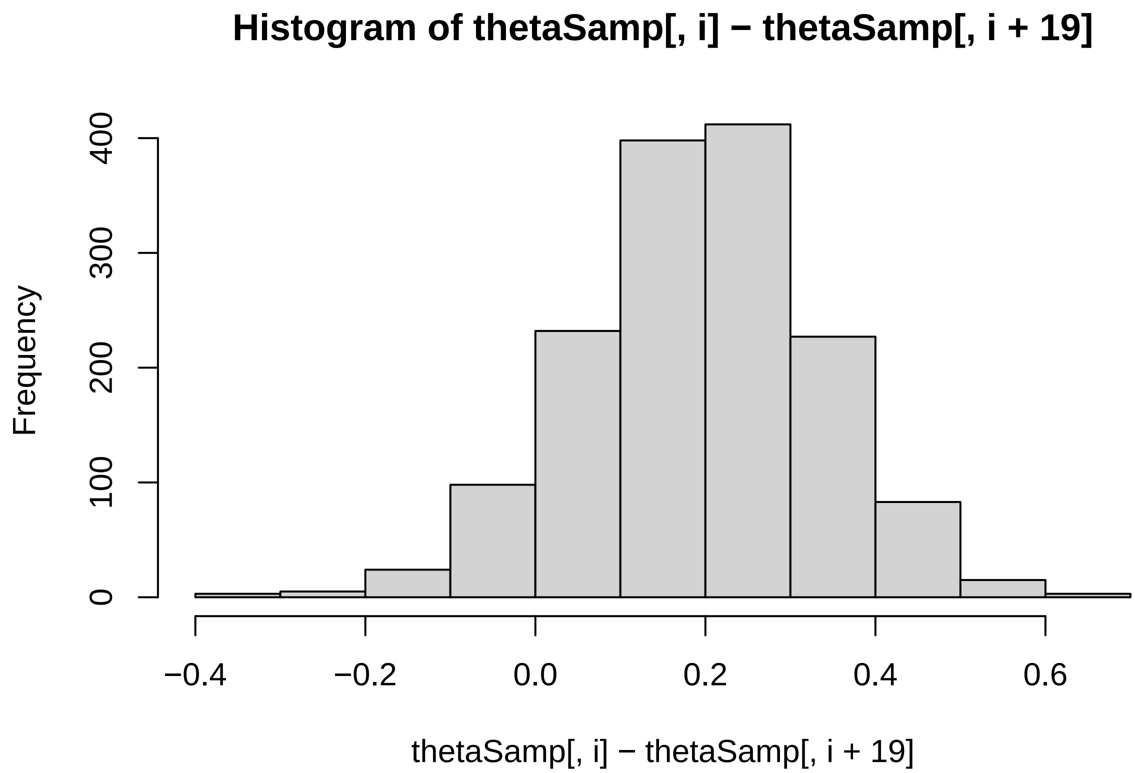

Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
150
250
350
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
thetaSamp[, i] − thetaSamp[, i + 19]

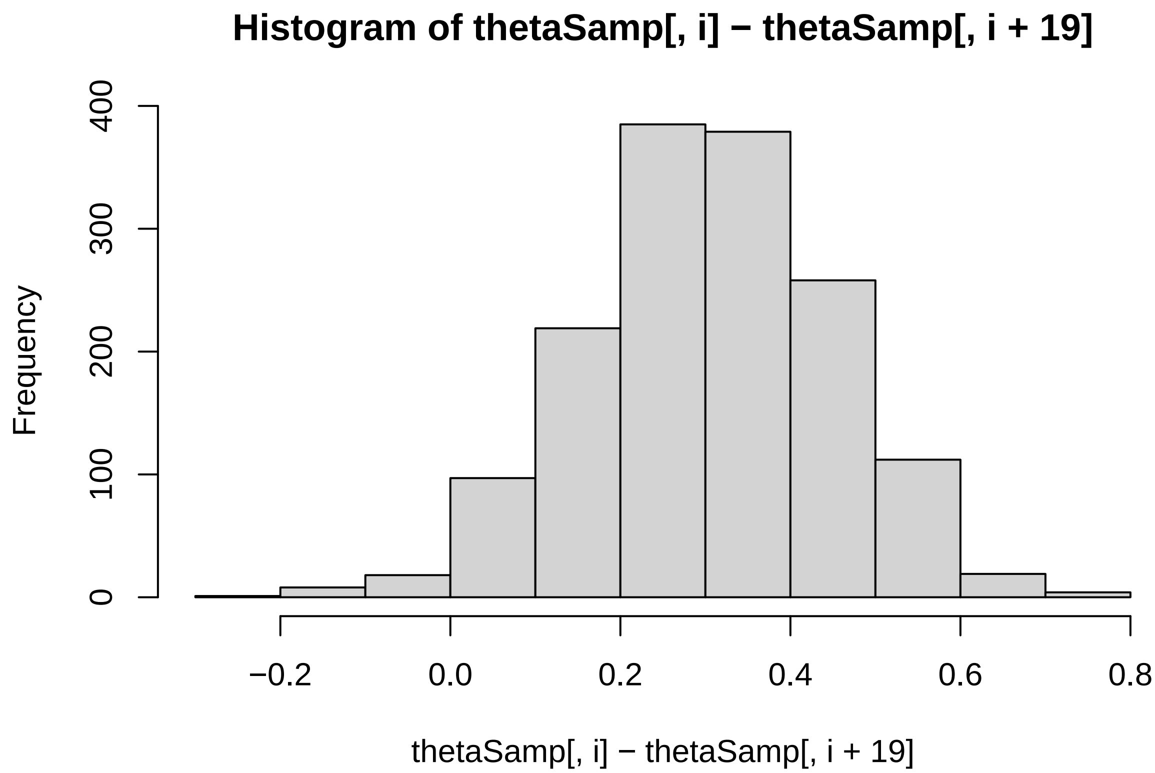


Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
100
150
200
250
−0.4
−0.2
0.0
0.2
0.4
thetaSamp[, i] − thetaSamp[, i + 19]


Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
100
150
200
250
−0.5
0.0
0.5
thetaSamp[, i] − thetaSamp[, i + 19]


Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
50
100
200
300
−0.2
0.0
0.2
0.4
thetaSamp[, i] − thetaSamp[, i + 19]


Histogram of thetaSamp[, i] − thetaSamp[, i + 19]
Frequency
0
100
200
300
400
−0.4
−0.2
0.0
0.2
0.4
0.6
thetaSamp[, i] − thetaSamp[, i + 19]

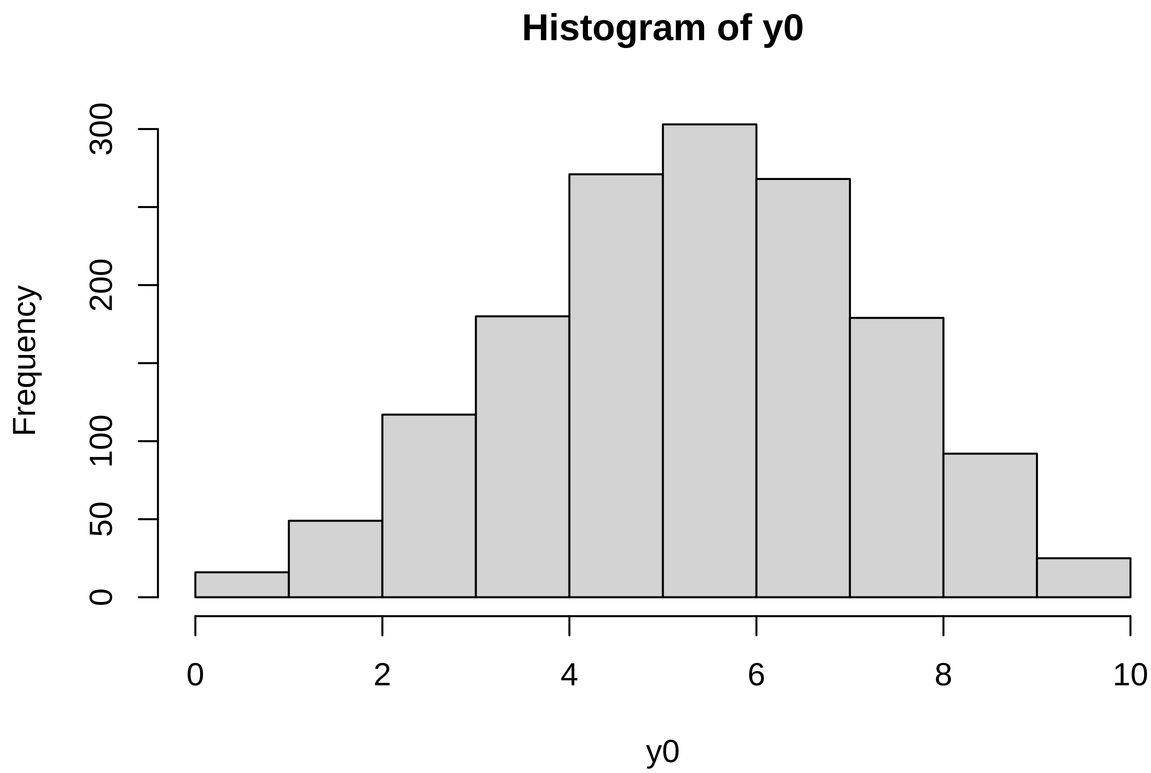


Part 3

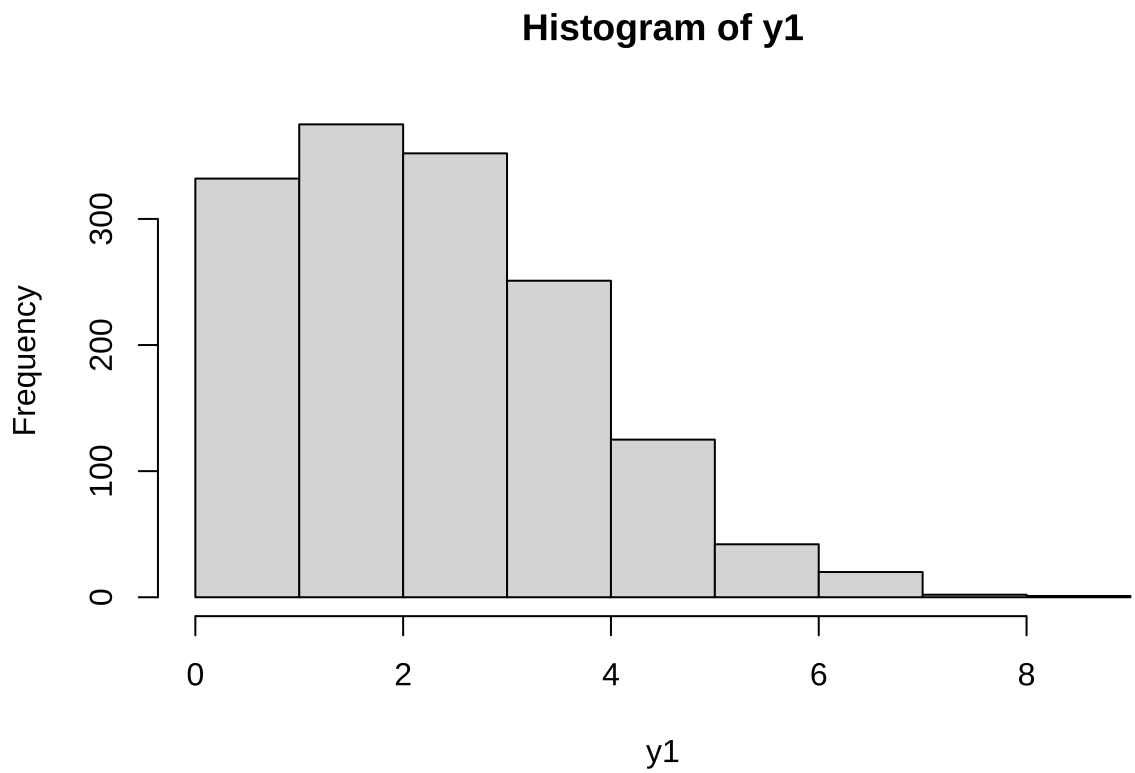
y0 predictions nonvege, y1 predictions vege for N__19 = 20

```
y0 = rep(0 , dim(thetaSamp)[1])
y1 = rep(0 , dim(thetaSamp)[1])

for ( i in 1:dim(thetaSamp)[1] ){
  y0[i] = rbinom(1,10,thetaSamp[i,19]);
  y1[i] = rbinom(1,10,thetaSamp[i,38]);
}
```

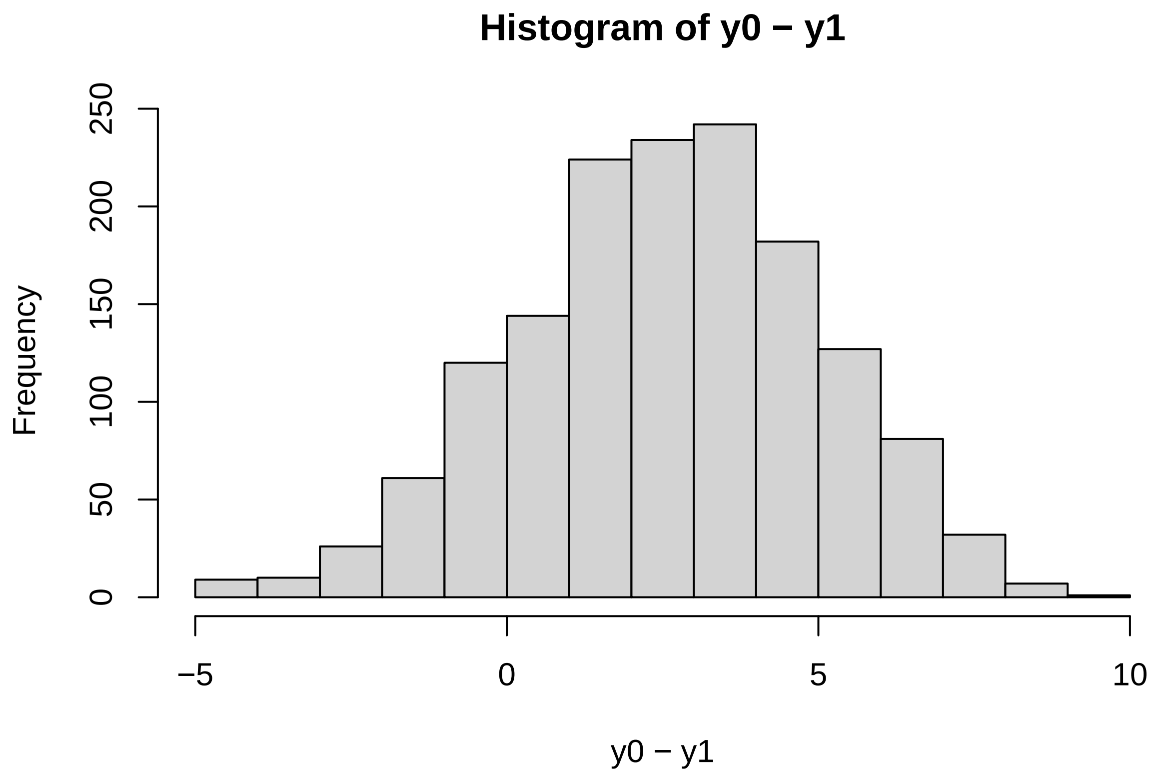
```
hist(y0)
```




```r
hist(y1)
```




```
hist(y0 - y1)
```

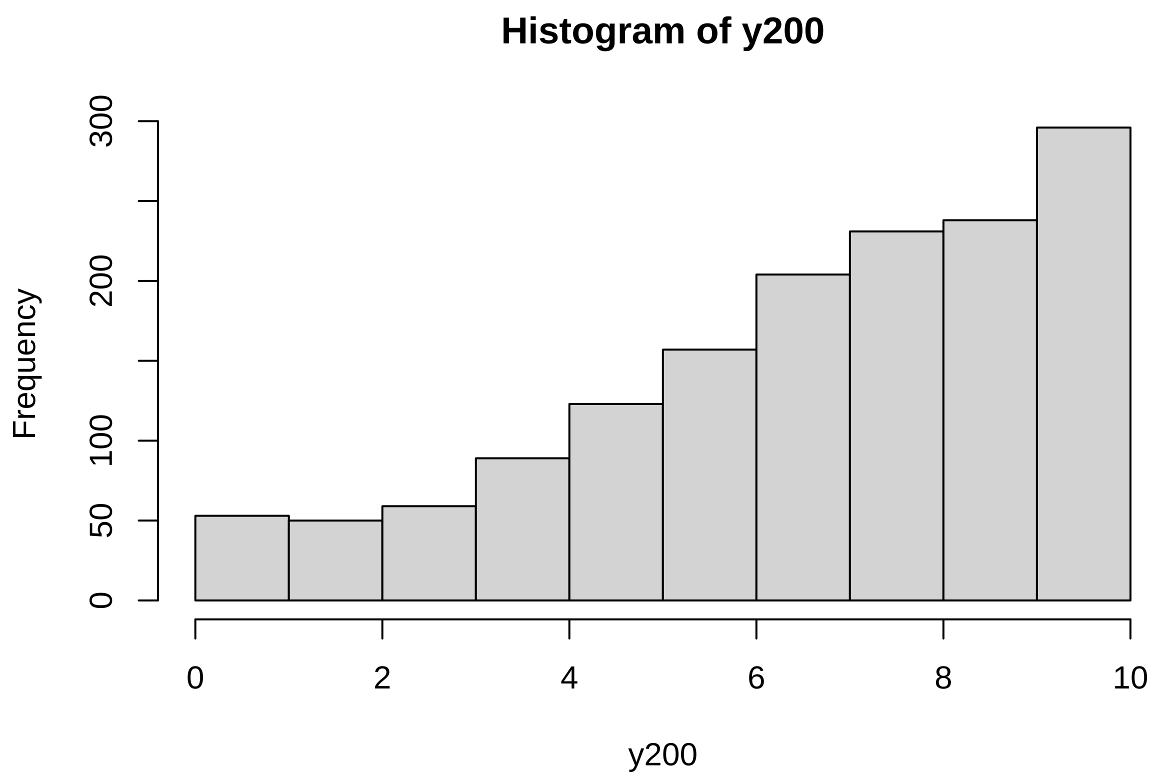


Part 4

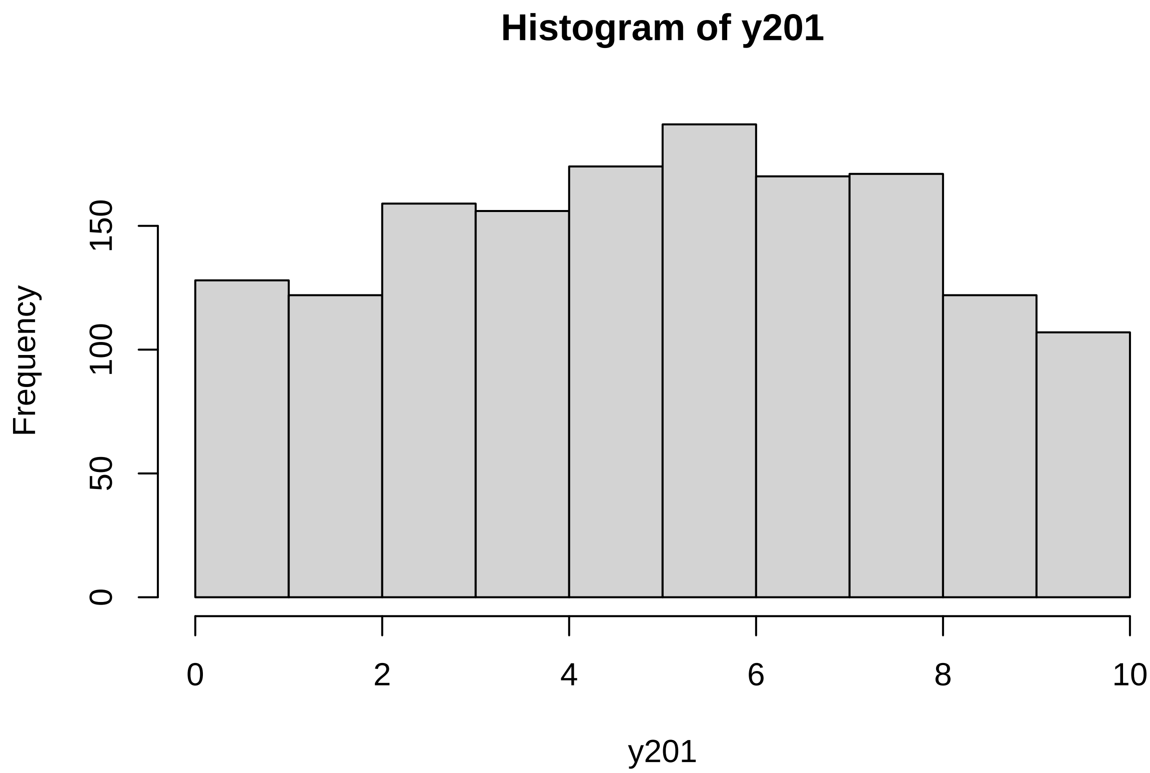
y200 predictions nonvege, y201 predictions vege for N__20 = 10

```
sSamp=as.matrix(post, pars ="s")
newThetaSamp = rep (0,dim(sSamp)[1])
newThetaSamp2 = rep (0,dim(sSamp)[1])

for (k in 1:dim(sSamp)[1]){
 newThetaSamp[k] = rbeta(1,muSamp[k,1] * sSamp[k,1] , sSamp[k,1]- muSamp[k,1] * sSamp[k,1])
}

for (k in 1:dim(sSamp)[1]){
 newThetaSamp2[k] = rbeta(1,muSamp[k,2] * sSamp[k,2] , sSamp[k,2]- muSamp[k,2] * sSamp[k,2])
}

y200 = rep(0 , dim(sSamp)[1])
y201 = rep(0 , dim(sSamp)[1])

for ( i in 1:dim(sSamp)[1] ){
  y200[i] = rbinom(1,10,newThetaSamp[i]);
  y201[i] = rbinom(1,10,newThetaSamp2[i]);
}
```

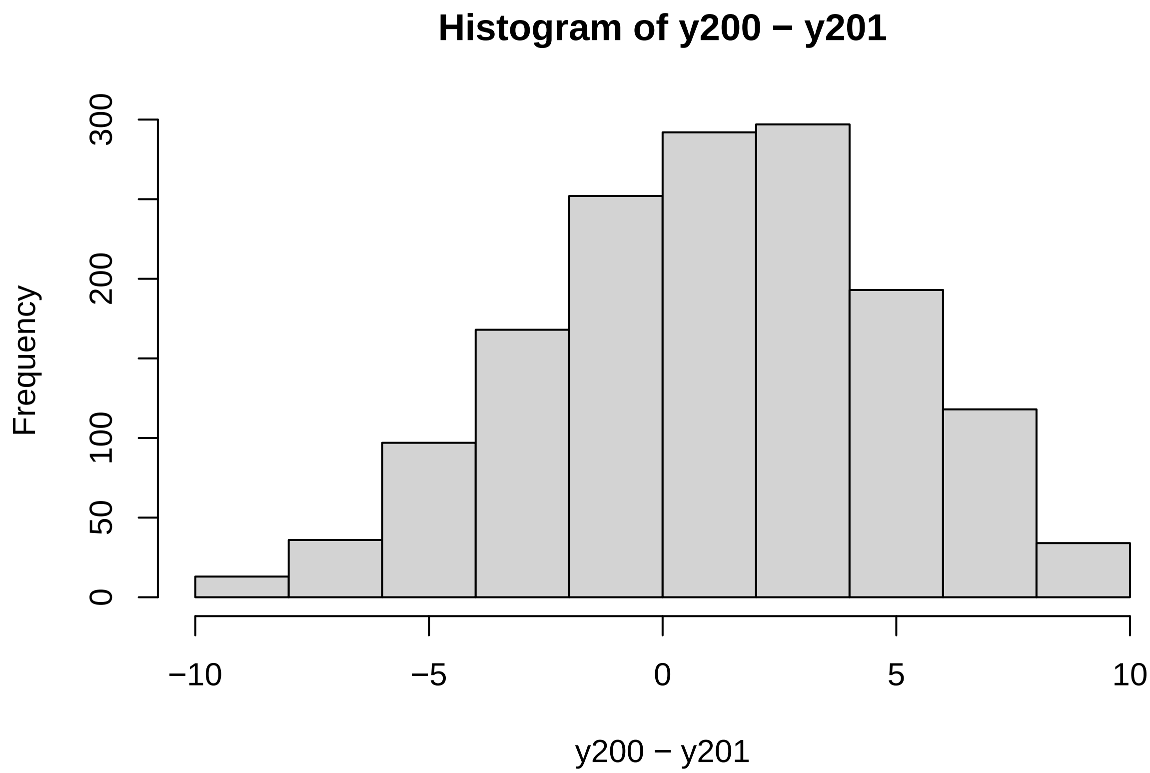
```r
hist(y200)
```




```r
hist(y201)
```




```
hist(y200 - y201)
```




```
#theta[j,k] ~ beta( mu[k] * s[k] , s[k] - mu[k] * s[k]);
```

Task 5

in the last model we are predicting everything from zero. Starting by s to mu to theta and finally y. Therefore it may contain more error than the one before it. Therefore i will take the model of 2.4 instead.

Grading

Total 20 points Each of the steps provides 4 points from correct answer and 2 points from an answer that is towards the right direction but includes minor mistake (e.g. a bug or typo)

References

Lari Veneranta, Richard Hudd and Jarno Vanhatalo (2013). Reproduction areas of sea-spawning Coregonids reflect the environment in shallow coastal waters. Marine Ecology Progress Series, 477:231-250. http://www.int-res.com/abstracts/meps/v477/p231-250/