TWAS_simulation

2023-05-30

Simulation code adapted from *https://rawgit.com/uqrmaie1/statgen_equations/master/statgen_equations.html#simulating-genotypes-with-ld* written by Robert Maier

First define some parameters for simulation

```
# seed for reproducible simulation:
set.seed(103);

# number of SNPs that we want to simulate:
m <- 500;
n <- 300;

# minor allele frequency for simulation:
maf <- runif(m, 0, 0.5);
```

simulate count matrix:

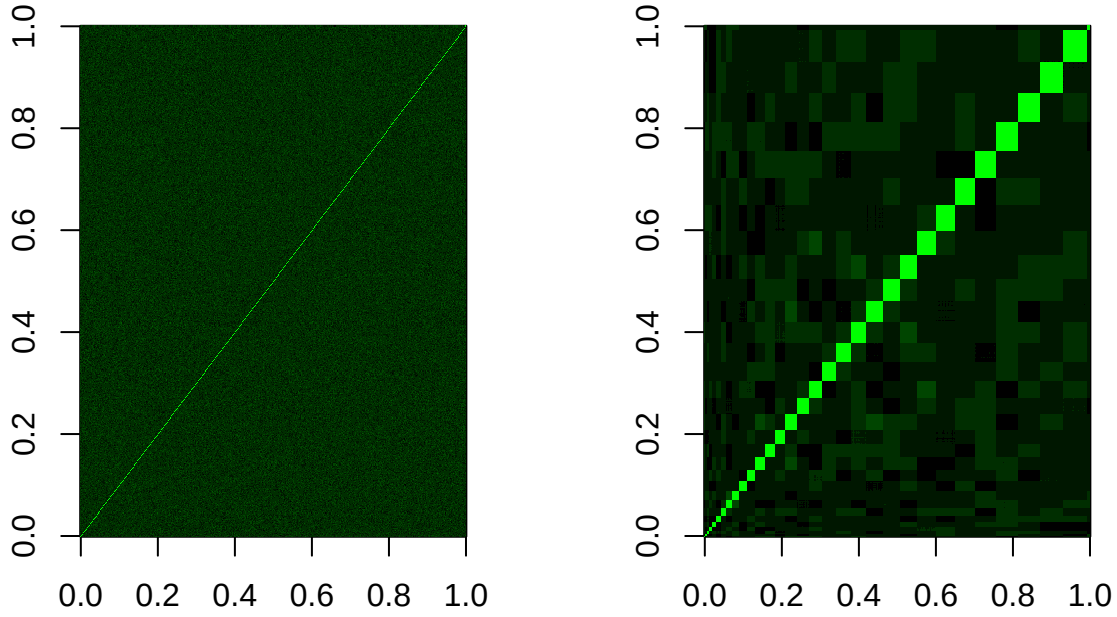
```
x012 <- t(replicate(n, rbinom(2*m, 2, c(maf, maf))))
polymorphic <- apply(x012, 2, var) > 0
x012 <- x012[,polymorphic][,1:m]
maf <- c(maf, maf)[polymorphic][1:m]
round(maf[1:10], 2)
```

```
##  [1] 0.11 0.03 0.26 0.25 0.06 0.04 0.22 0.10 0.02 0.16
```

simulate count matrix with LD:

```
x012ld <- jitter(x012[,rep(1:m, 1:m)[1:m]], factor = 0.03)
xld <- scale(x012ld)
ldld <- (t(xld) %*% xld)/n
grmld <- (xld %*% t(xld))/m
ldscores_sampleld <- colSums(ldld^2)
ldscoresld <- (ldscores_sampleld*n - m) / (n + 1)
varxld = apply(x012ld, 2, var)
```

```r
# ld visualization:
greens <- colorRampPalette(c('black', 'green'))(12)
par(mfrow=c(1,2))
ld1 <- cor(x012)
image(ld1, col=greens)
image(ldld, col=greens)
```



simulate gene expression level associated with this region:

Set of causal SNPs are randomly chosen from m SNPs:

$$where \quad m = \text{number of SNPs}$$

$$p(SNP = causal) = \frac{1}{500}$$

$$\beta \sim N(0, 1)$$

$$Y = X\beta + \varepsilon$$

$$where \quad \varepsilon \sim N(0, \alpha\sqrt{var(X\beta)})$$

$$where \quad \alpha \in 1, 5, 10$$

representing increasing amount of variance explained by environment

Repeat this process 1000 times to get the 1000 instances of gene expression with different causal effect at different genetic loci

```
# randomly determine causal variants:
causal.number <- 30;

# repeat the generation process:
n_rep = 1000;
expression.instances <- matrix(NA, nrow = n, ncol = n_rep);

# simulate what are the causal variants:
causal.variants <- sample(m, size = causal.number, replace = FALSE)

# randomly determine their effect size:
causal.effect <- rnorm(n = causal.number, sd = 1);
names(causal.effect) <- causal.variants;

# find the amount of variance explained by genetics:
genetic.variance <- var(xld[, causal.variants] %*% causal.effect);
scaling.factor <- 1;
environment.variance <- scaling.factor * genetic.variance;
simulated.genotype <- vector('list', length = n_rep);

for (instance in seq(1, n_rep)){
  # randomly generate the genotype:
  x012 <- t(replicate(n, rbinom(2*m, 2, c(maf, maf))))
  polymorphic <- apply(x012, 2, var) > 0
  x012 <- x012[,polymorphic][,1:m]
  maf <- c(maf, maf)[polymorphic][1:m]
  round(maf[1:10], 2)
  x012ld <- jitter(x012[,rep(1:m, 1:m)[1:m]], factor = 0.03)
  xld <- scale(x012ld)

  # randomly generate residual points:
  residual <- rnorm(n = n, sd = sqrt(environment.variance));

  # simulate Y:
  expression.data = xld[, causal.variants] %*% causal.effect + residual;

  # store simulated expression:
  expression.instances[, instance] <- expression.data;

  # store simulated genotype:
  simulated.genotype[[instance]] <- xld;
  }
```

Now lets build models to see how different models have the best inference in our simulated dataset

elastic net penalized eQTL computation:

```
require(glmnet);
```

```
## Loading required package: glmnet
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-7
```

```
elastic.net <- lapply(
  seq(1, n_rep),
  FUN = function(instance) cv.glmnet(
    x = simulated.genotype[[instance]],
    y = expression.instances[, instance],
    alpah = 0.5
    )
  );
```

Lasso penalized eQTL computation:

```
require(glmnet);
lasso.regression <- lapply(
  seq(1, n_rep),
  FUN = function(instance) cv.glmnet(
    x = simulated.genotype[[instance]],
    y = expression.instances[, instance],
    alpah = 1
    )
  );
```

Ridge penalized eQTL computation:

```
require(glmnet);
ridge.regression <- elastic.net <- lapply(
  seq(1, n_rep),
  FUN = function(instance) cv.glmnet(
    x = simulated.genotype[[instance]],
    y = expression.instances[, instance],
    alpah = 0
    )
  );
```

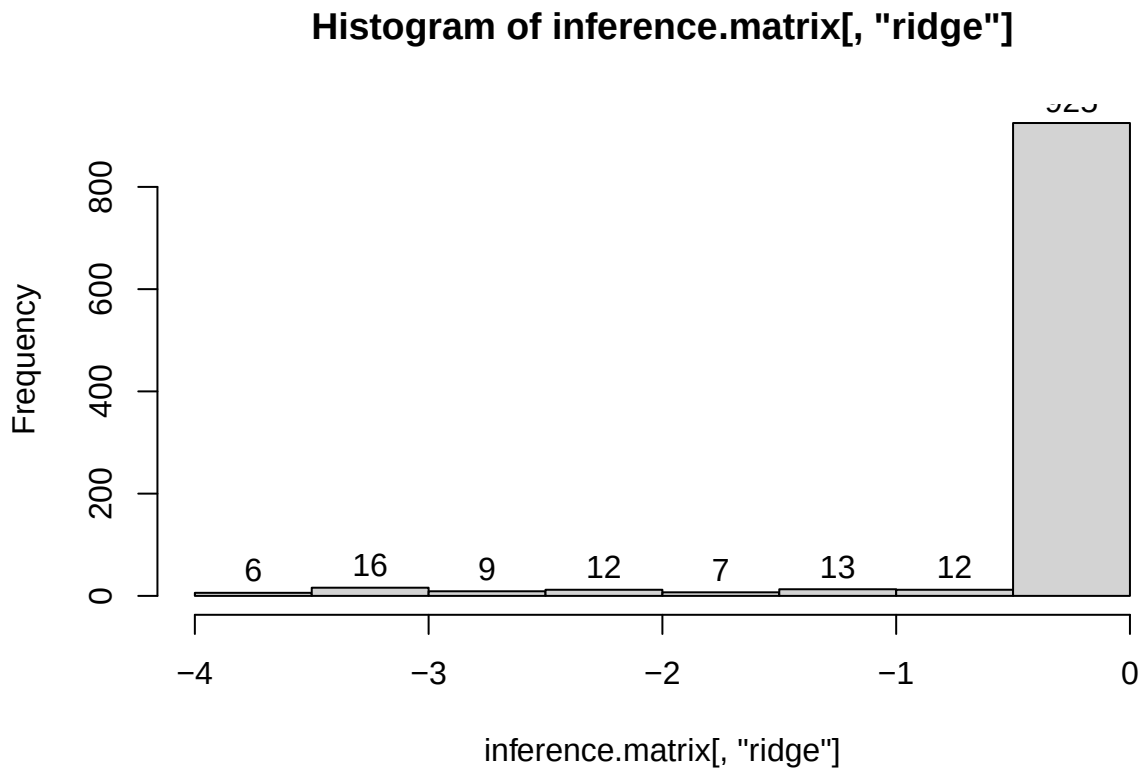
Visualization SNP simulation result with large effect size:

```
snp = '135';
methods <- c('ridge', 'lasso', 'elastic');
inference.matrix <- matrix(NA, nrow = n_rep, ncol = length(methods));
rownames(inference.matrix) <- paste0('simulation', seq(1, n_rep));
colnames(inference.matrix) <- methods;
# first summarize what is the distribution coefficient at the causal variant 499:
for (instance in seq(1, n_rep)) {
  # get the coefficient at variant 135 in data:
```

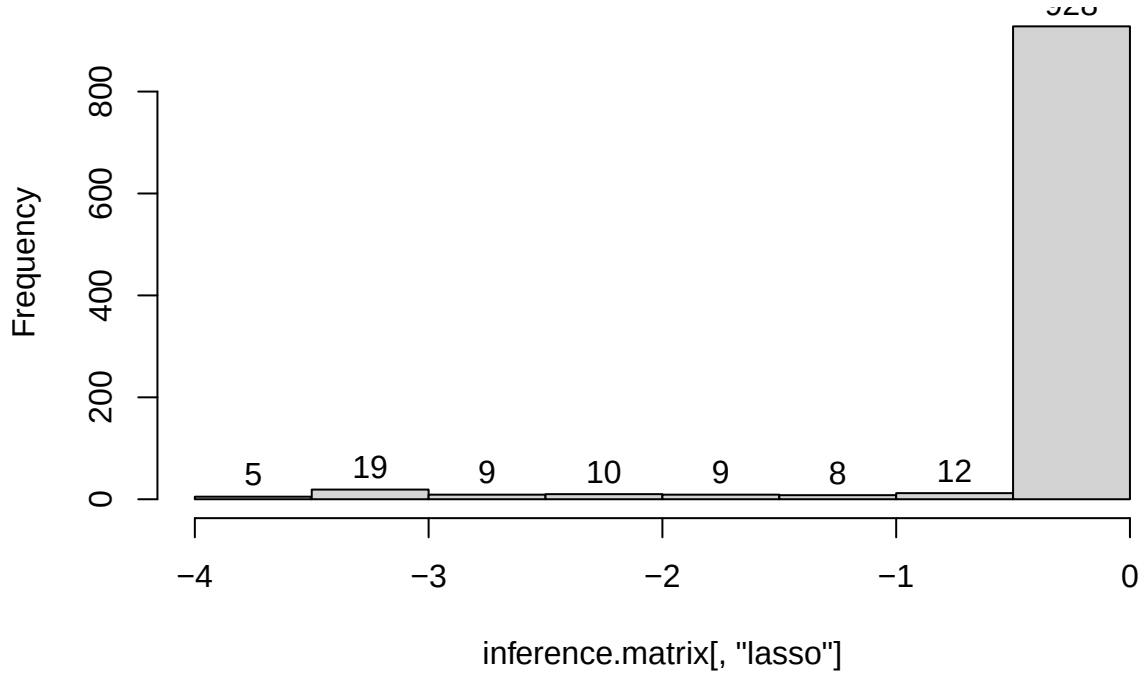
```
  ridge.beta <- coef(ridge.regression[[instance]], s = 'lambda.min')[paste0('V', snp), ];
  lasso.beta <- coef(lasso.regression[[instance]], s = 'lambda.min')[paste0('V', snp), ];
  elastic.beta <- coef(elastic.net[[instance]], s = 'lambda.min')[paste0('V', snp), ];

  # get the snp result:
  inference.matrix[instance, 'ridge'] <- ridge.beta;
  inference.matrix[instance, 'lasso'] <- lasso.beta;
  inference.matrix[instance, 'elastic'] <- elastic.beta;
}

# report true effect size:
cat('snp: ', snp, ' causal effect = ', causal.effect[snp])
```

```
## snp:  135  causal effect =  -2.253626
```

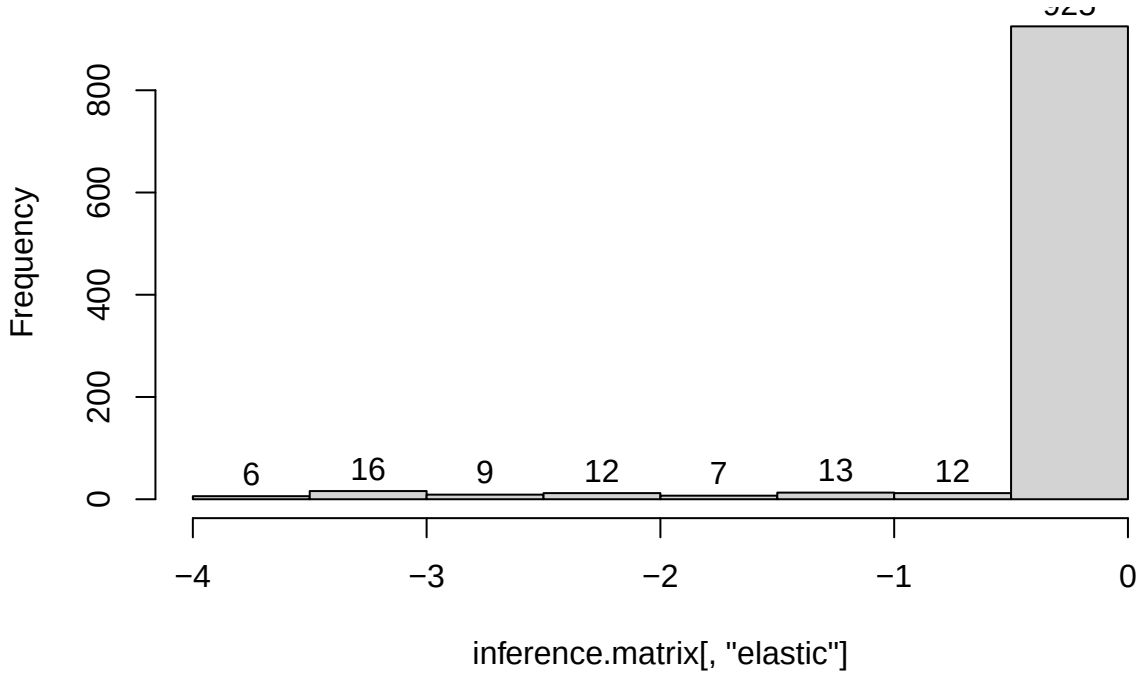
```
hist(inference.matrix[, 'ridge'], labels=T)
```




```
hist(inference.matrix[, 'lasso'], labels=T)
```




```
hist(inference.matrix[, 'elastic'], labels=T)
```

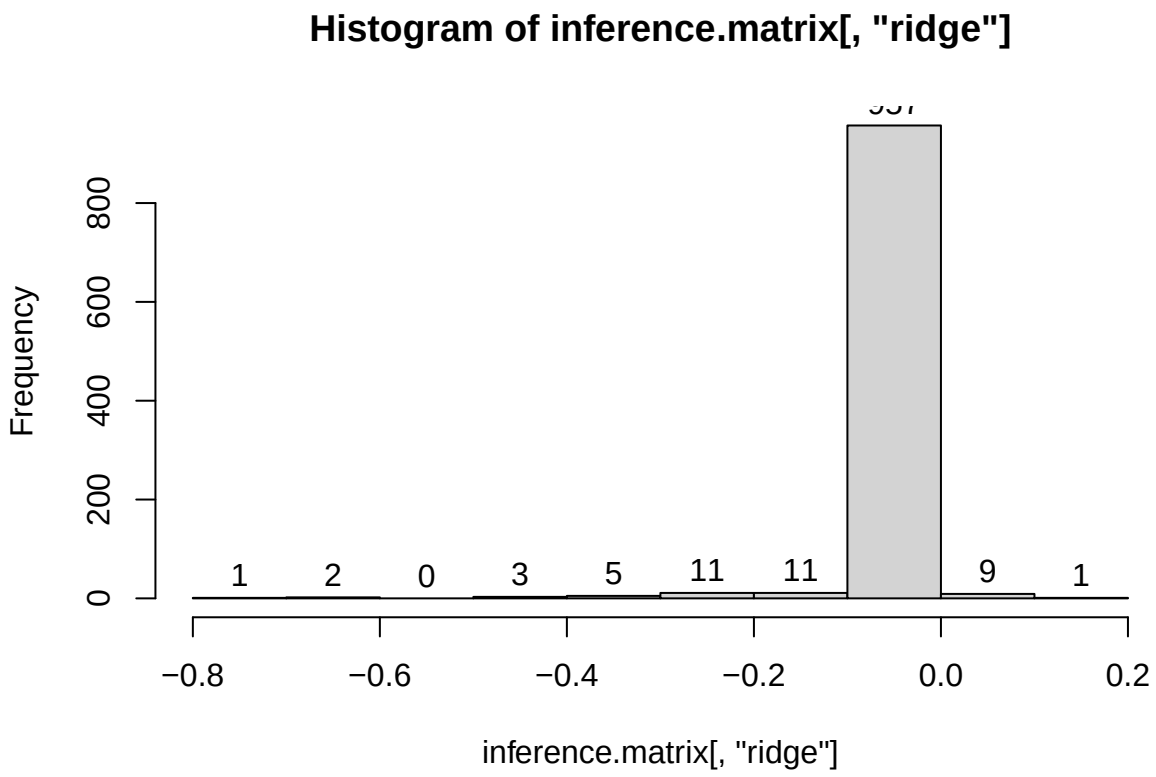


visualize snp with low effect size

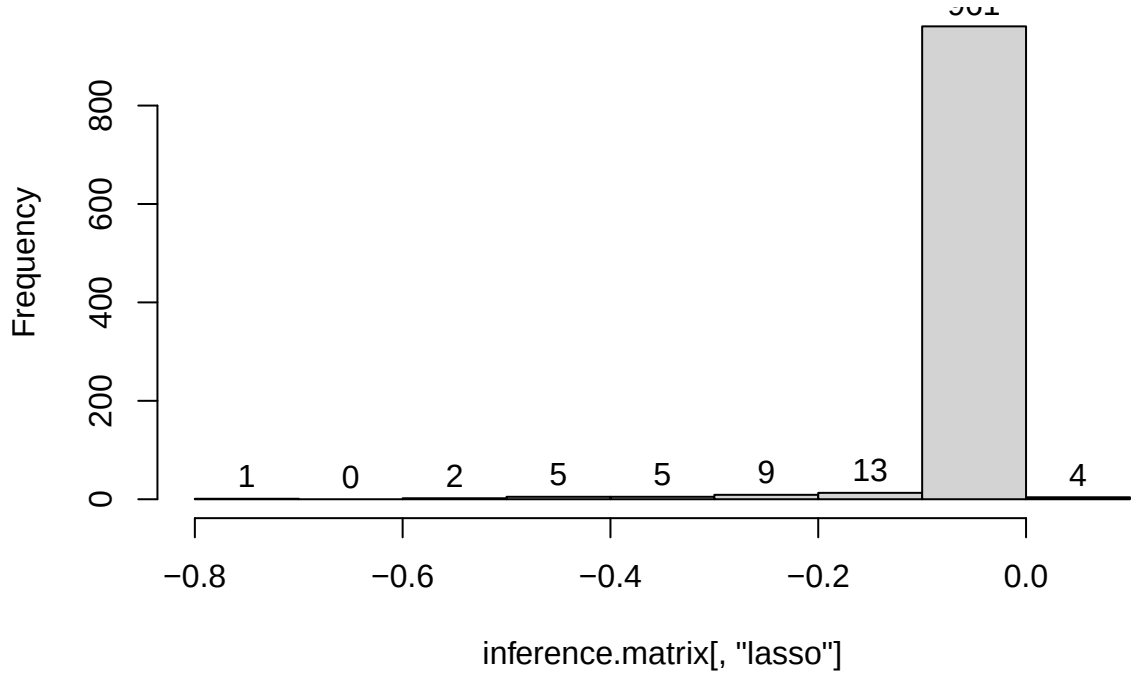
```
snp = '258';
methods <- c('ridge', 'lasso', 'elastic');
inference.matrix <- matrix(NA, nrow = n_rep, ncol = length(methods));
rownames(inference.matrix) <- paste0('simulation', seq(1, n_rep));
colnames(inference.matrix) <- methods;
# first summarize what is the distribution coefficient at the causal variant 499:
for (instance in seq(1, n_rep)) {
  # get the coefficient at variant in data:
  ridge.beta <- coef(ridge.regression[[instance]], s = 'lambda.min')[paste0('V', snp), ];
  lasso.beta <- coef(lasso.regression[[instance]], s = 'lambda.min')[paste0('V', snp), ];
  elastic.beta <- coef(elastic.net[[instance]], s = 'lambda.min')[paste0('V', snp), ];

  # get the snp result:
  inference.matrix[instance, 'ridge'] <- ridge.beta;
  inference.matrix[instance, 'lasso'] <- lasso.beta;
  inference.matrix[instance, 'elastic'] <- elastic.beta;

}

# report true effect size:
cat('snp: ', snp, ' causal effect = ', causal.effect[snp])
```

```
## snp:  258  causal effect =  0.2926644
```

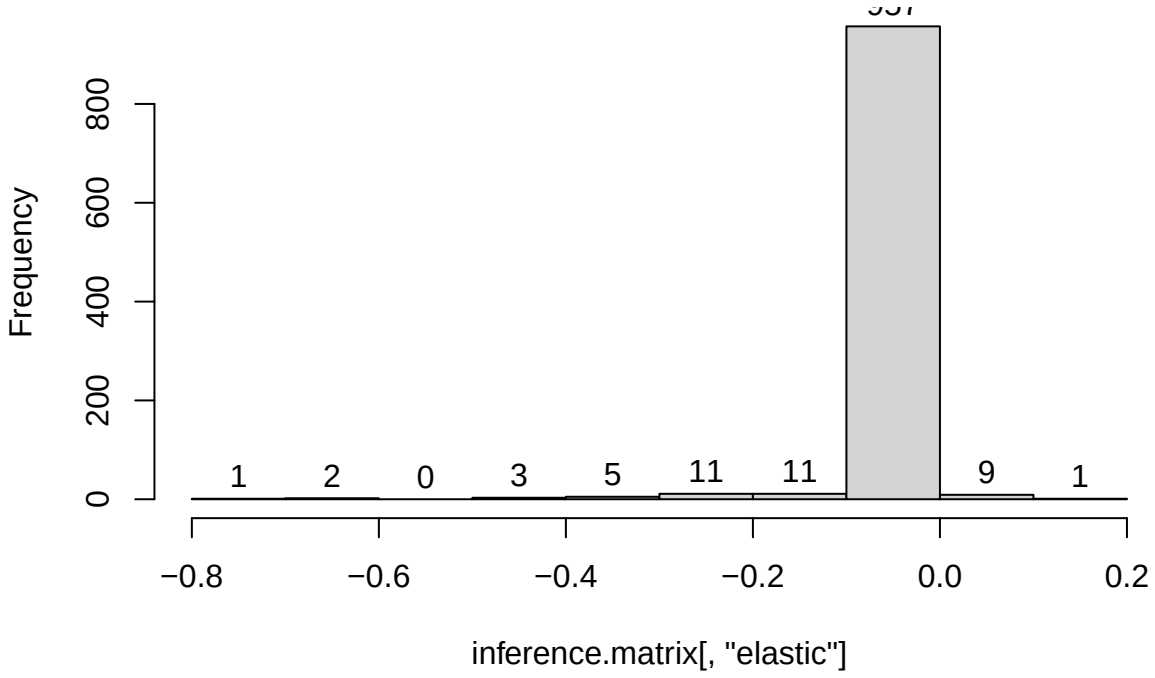
```
# we will draw the distribution for snp
hist(inference.matrix[, 'ridge'], labels=T)
```




```
hist(inference.matrix[, 'lasso'], labels=T)
```




```
hist(inference.matrix[, 'elastic'], labels=T)
```

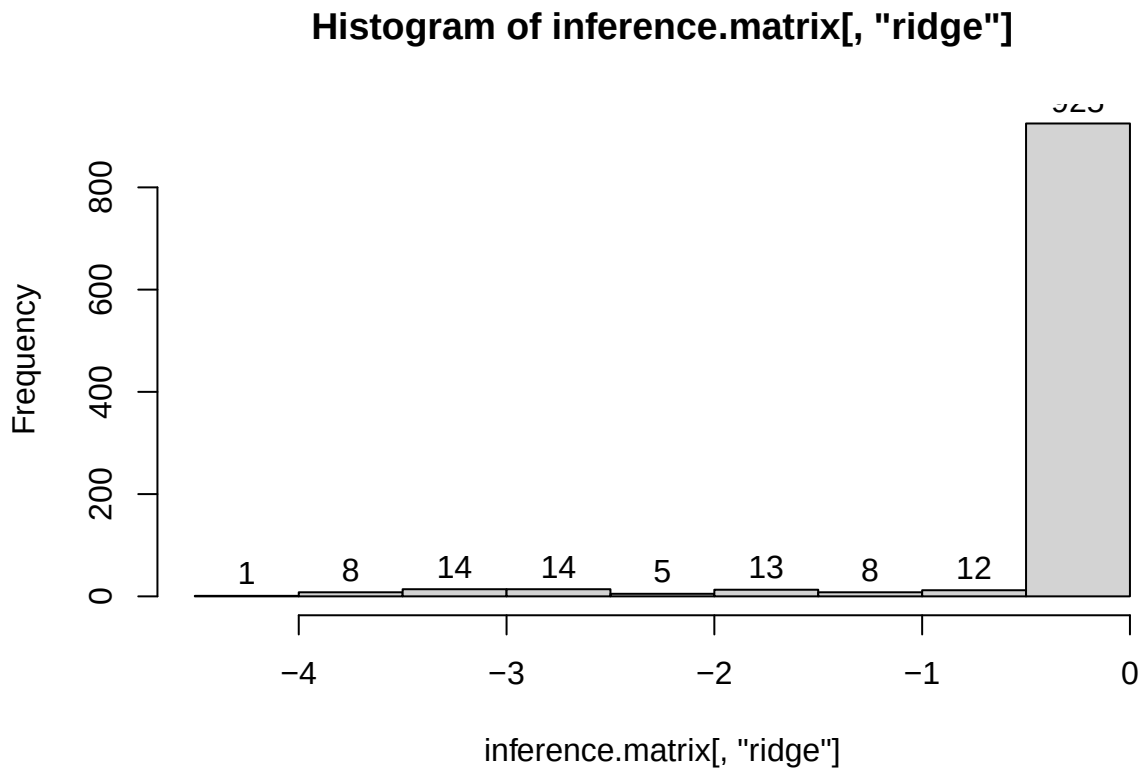


Visualize snp with zero effect size

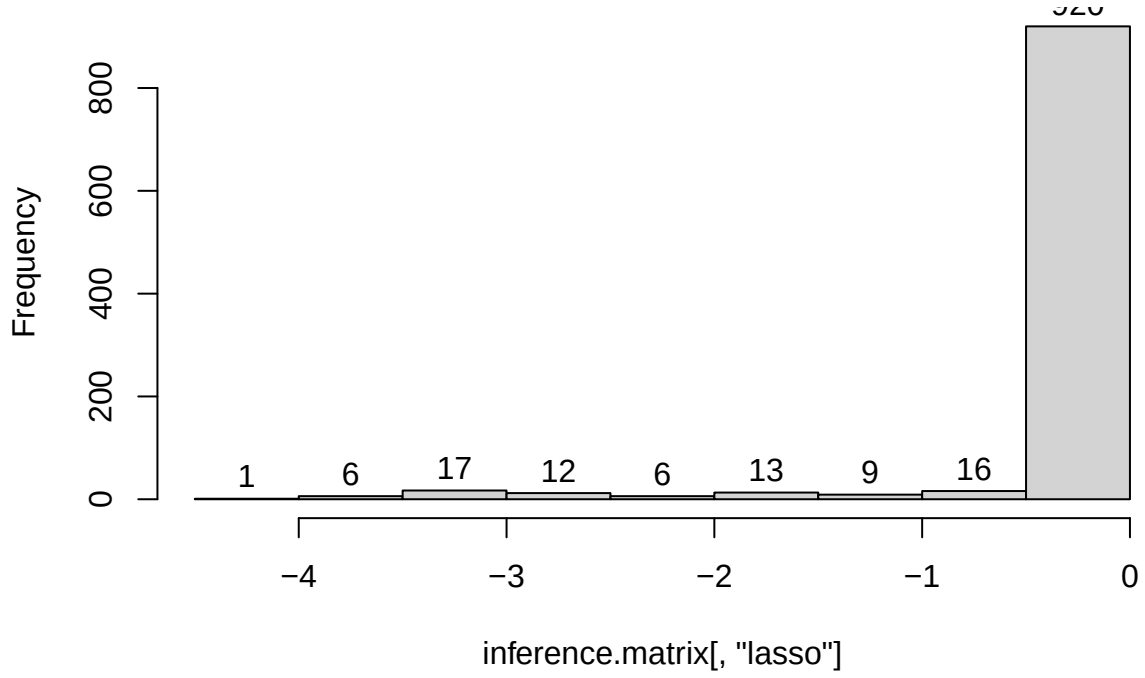
```
snp = '136';
methods <- c('ridge', 'lasso', 'elastic');
inference.matrix <- matrix(NA, nrow = n_rep, ncol = length(methods));
rownames(inference.matrix) <- paste0('simulation', seq(1, n_rep));
colnames(inference.matrix) <- methods;
# first summarize what is the distribution coefficient at the causal variant 499:
for (instance in seq(1, n_rep)) {
  # get the coefficient at variant in data:
  ridge.beta <- coef(ridge.regression[[instance]], s = 'lambda.min')[paste0('V', snp), ];
  lasso.beta <- coef(lasso.regression[[instance]], s = 'lambda.min')[paste0('V', snp), ];
  elastic.beta <- coef(elastic.net[[instance]], s = 'lambda.min')[paste0('V', snp), ];

  # get the snp result:
  inference.matrix[instance, 'ridge'] <- ridge.beta;
  inference.matrix[instance, 'lasso'] <- lasso.beta;
  inference.matrix[instance, 'elastic'] <- elastic.beta;
}

# report true effect size:
cat('snp: ', snp, ' causal effect = ', causal.effect[snp])
```

```
## snp:  136  causal effect =  NA
```

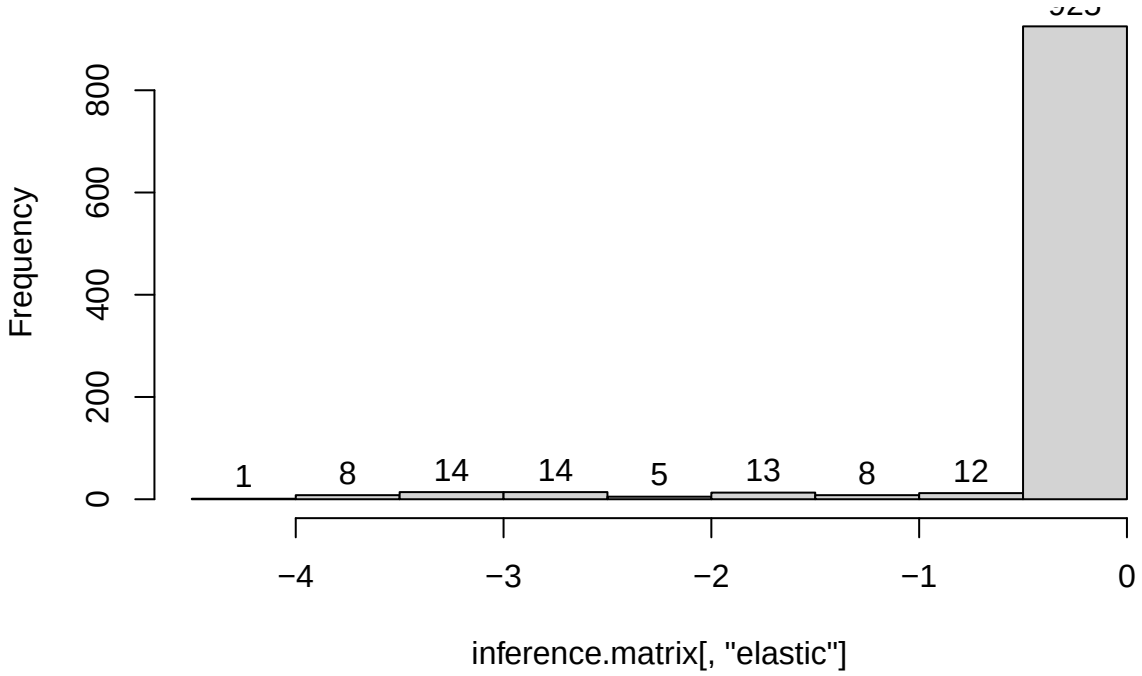
```
# we will draw the distribution for snp
hist(inference.matrix[, 'ridge'], labels=T)
```




```
hist(inference.matrix[, 'lasso'], labels=T)
```




```
hist(inference.matrix[, 'elastic'], labels=T)
```


Histogram of inference.matrix[, "elastic"]
Frequency
0
200
400
600
800
1
8
14
14
5
13
8
12
−4
−3
−2
−1
0
inference.matrix[, "elastic"]