snp_ptm_lab

Christopher Seybold

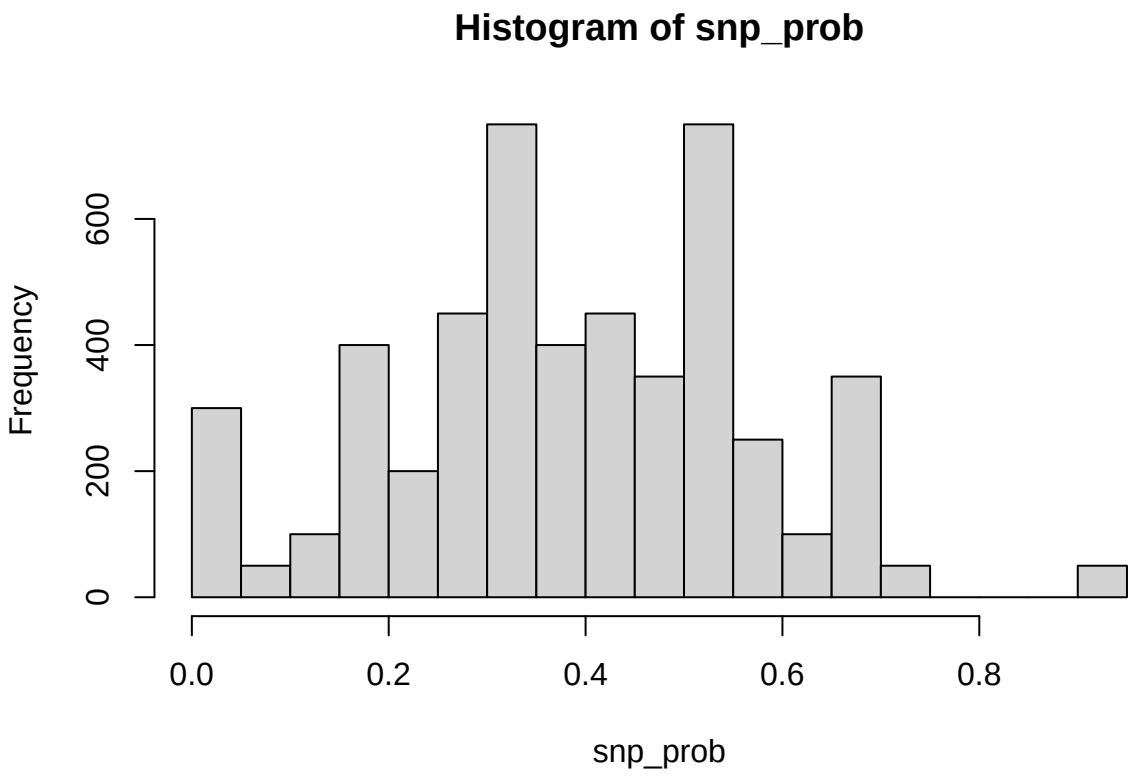
2023-11-14

```
library(purrr)
library(MASS)
set.seed(24601)
library(tidyverse)
```

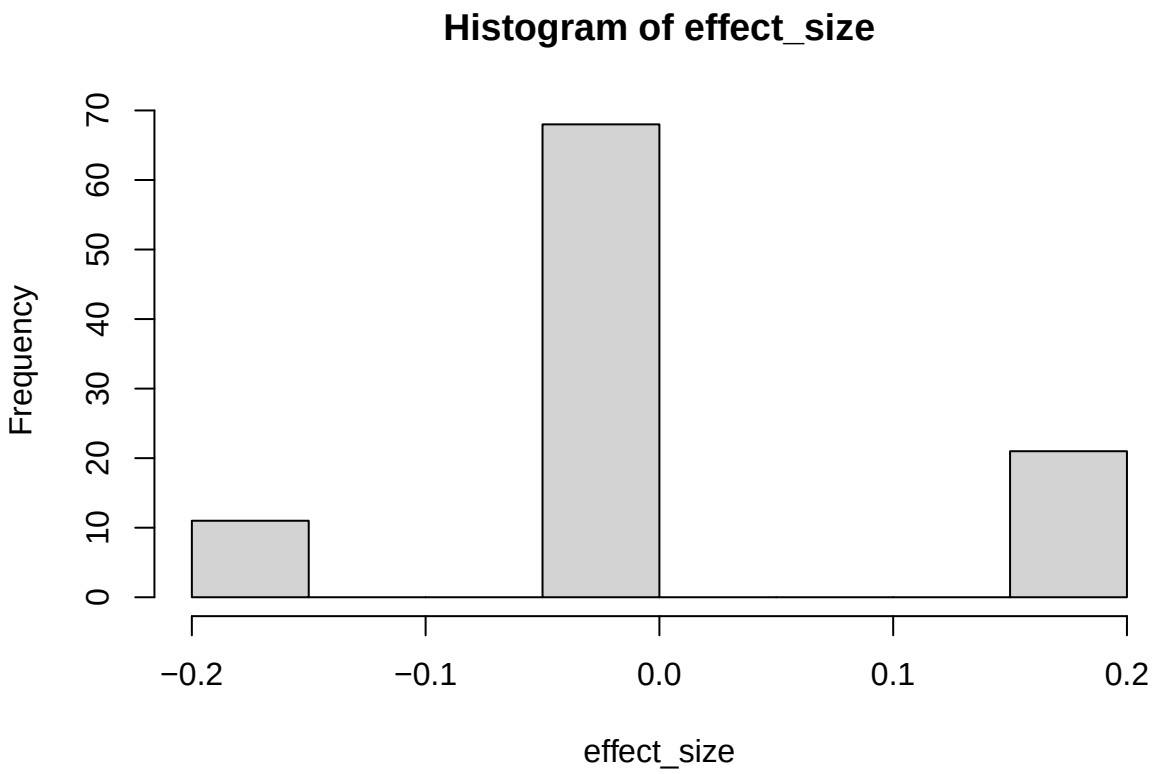
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::select() masks MASS::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
patient_count <- 50
snp_ptm_id_count <- 100
```

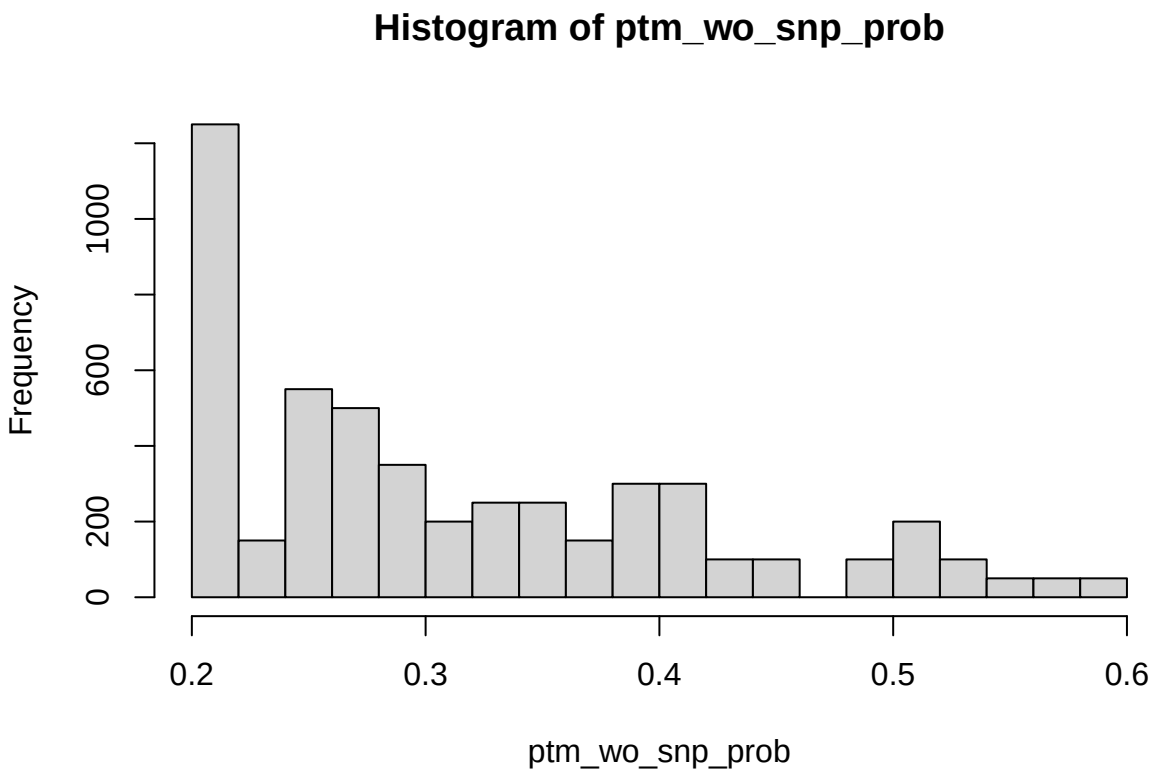
```
#p
snp_prob <- rep(rnorm(snp_ptm_id_count, 0.4, 0.2), patient_count)
snp_prob <- pmax(pmin(1, snp_prob), 0)
hist(snp_prob)
```



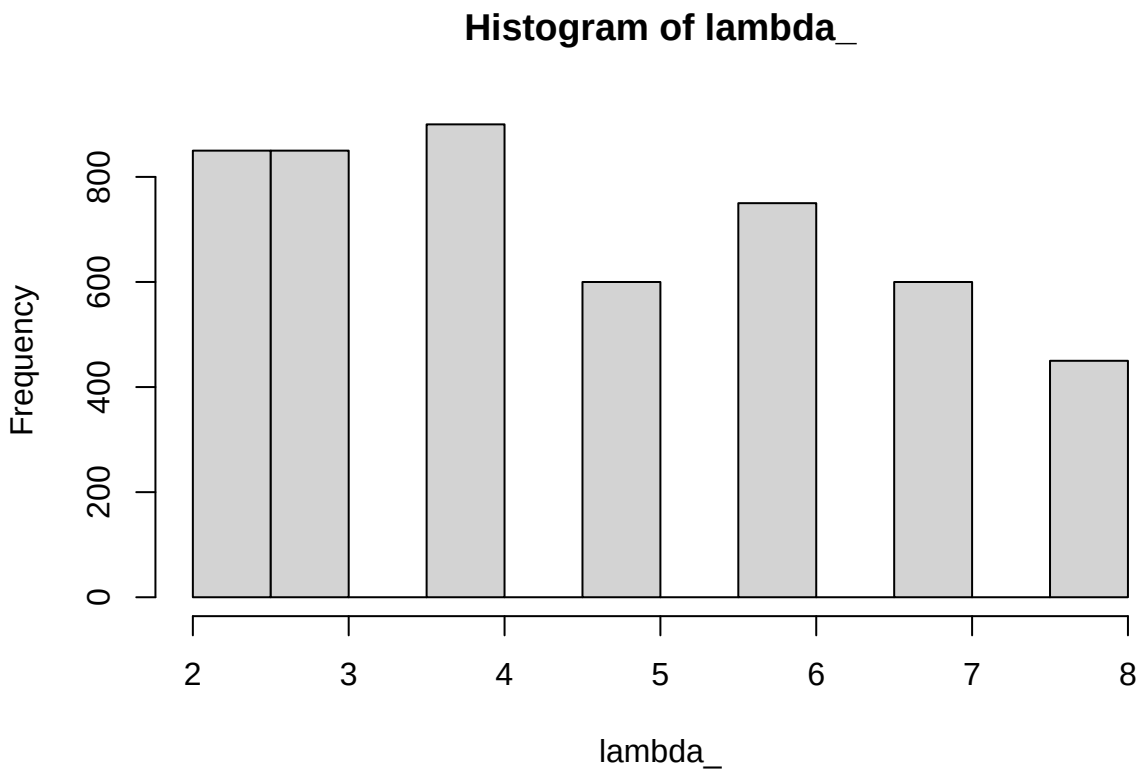

```
#z
effect_size <- sample(c(-0.2, 0, 0.2), size = snp_ptm_id_count, replace = TRUE, prob = c(0.15, 0.7, 0.15
hist(effect_size)
```



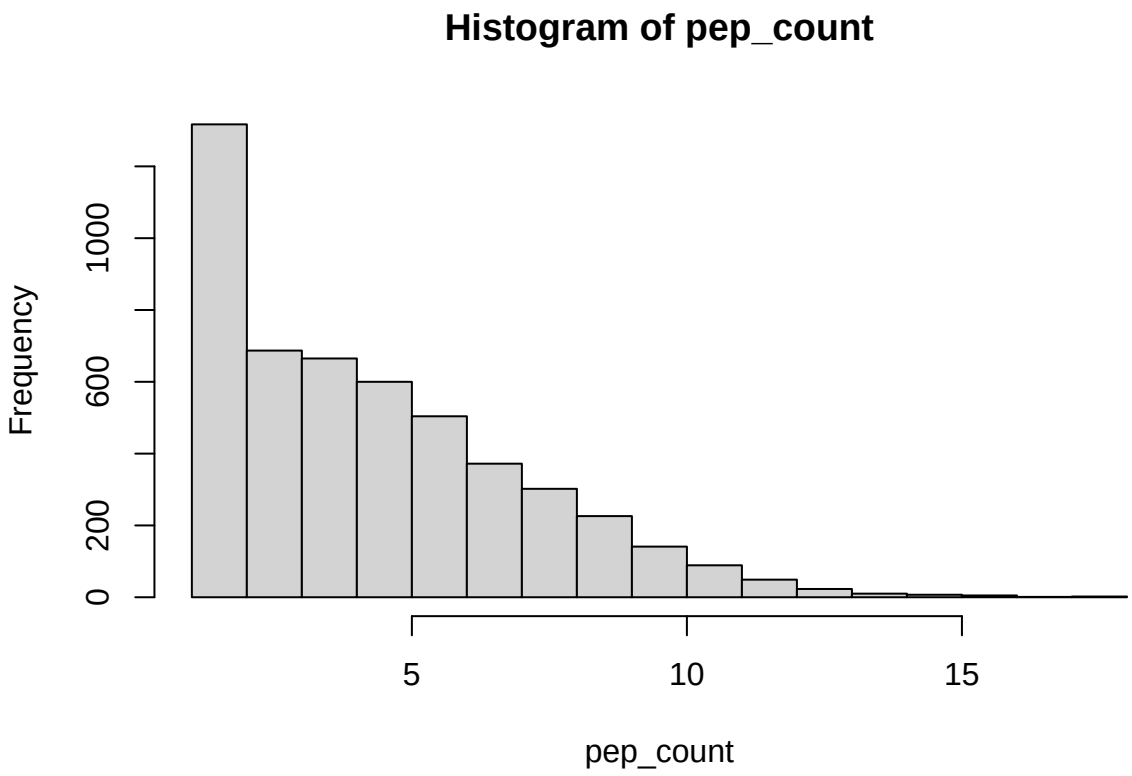

```
#probability of PTM without SNP (q)
ptm_wo_snp_prob <- rep(rnorm(snp_ptm_id_count, 0.3, 0.15), patient_count)
ptm_wo_snp_prob <- pmax(pmin(1, ptm_wo_snp_prob), 0.2)
hist(ptm_wo_snp_prob)
```



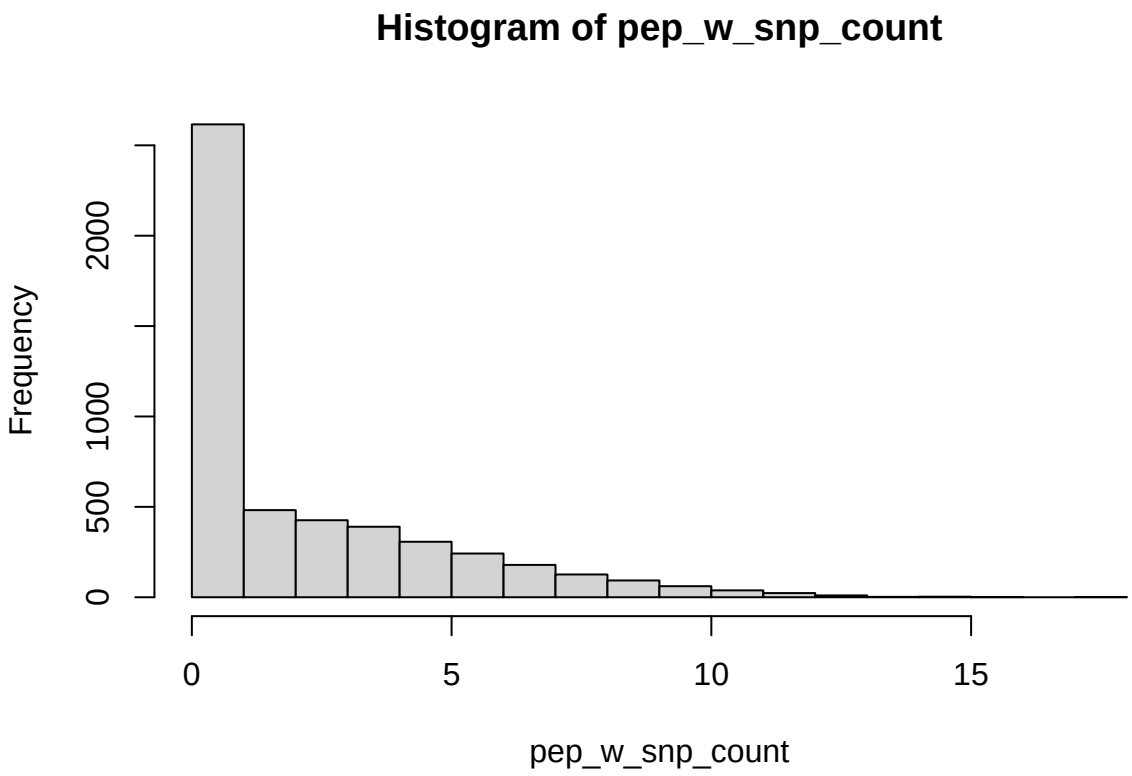

```
#peptide count parameter
lambda_ <- rep(sample(2:8, size = snp_ptm_id_count, replace = TRUE), patient_count)
hist(lambda_)
```



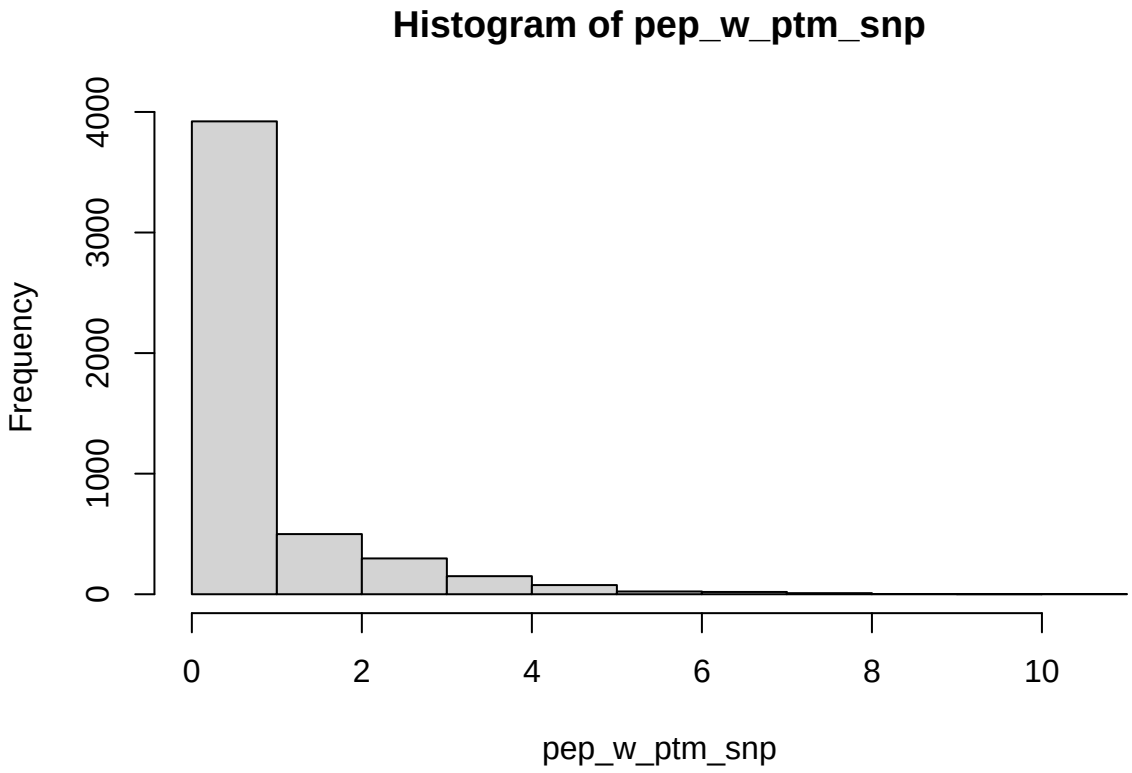

```
#n
pep_count = rpois(patient_count*snp_ptm_id_count, lambda = lambda_)
pep_count = replace(pep_count, pep_count==0, 1)
hist(pep_count)
```



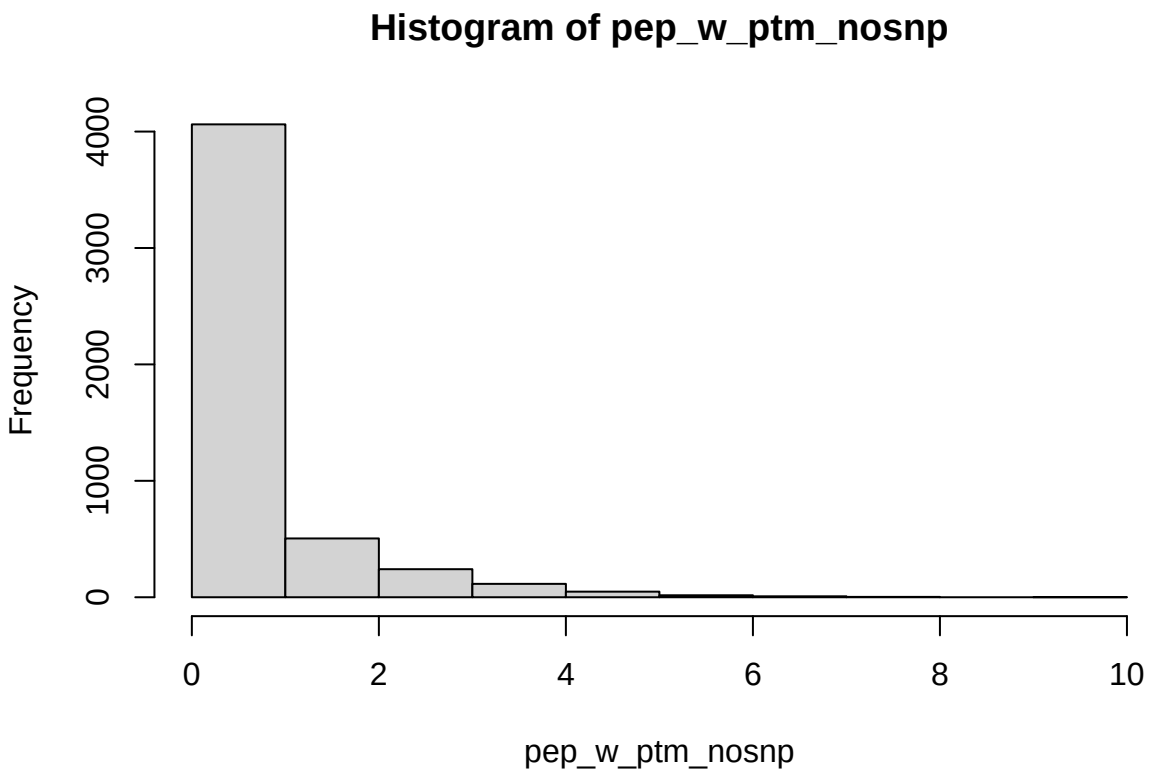

```
#x
pep_w_snp_count <- numeric(patient_count * snp_ptm_id_count)
for (i in 1:(patient_count*snp_ptm_id_count)) {
  pep_w_snp_count[i] = min(pep_count[i], sample(c(0, 0.5, 1), size = 1, replace = TRUE, prob = c((1 - s
}
pep_w_snp_count[pep_w_snp_count == 1] <- pep_count[pep_w_snp_count == 1]
pep_w_snp_count[pep_w_snp_count == 0.5] <- rbinom(length(pep_w_snp_count[pep_w_snp_count == 0.5]), pep_

hist(pep_w_snp_count)
```




```
#y1
pep_w_ptm_snp = rbinom(patient_count * snp_ptm_id_count, pep_w_snp_count, ptm_wo_snp_prob + effect_size)
hist(pep_w_ptm_snp)
```




```
#y2
pep_w_ptm_nosnp = rbinom(patient_count * snp_ptm_id_count, pep_count - pep_w_snp_count, ptm_wo_snp_prob
hist(pep_w_ptm_nosnp)
```

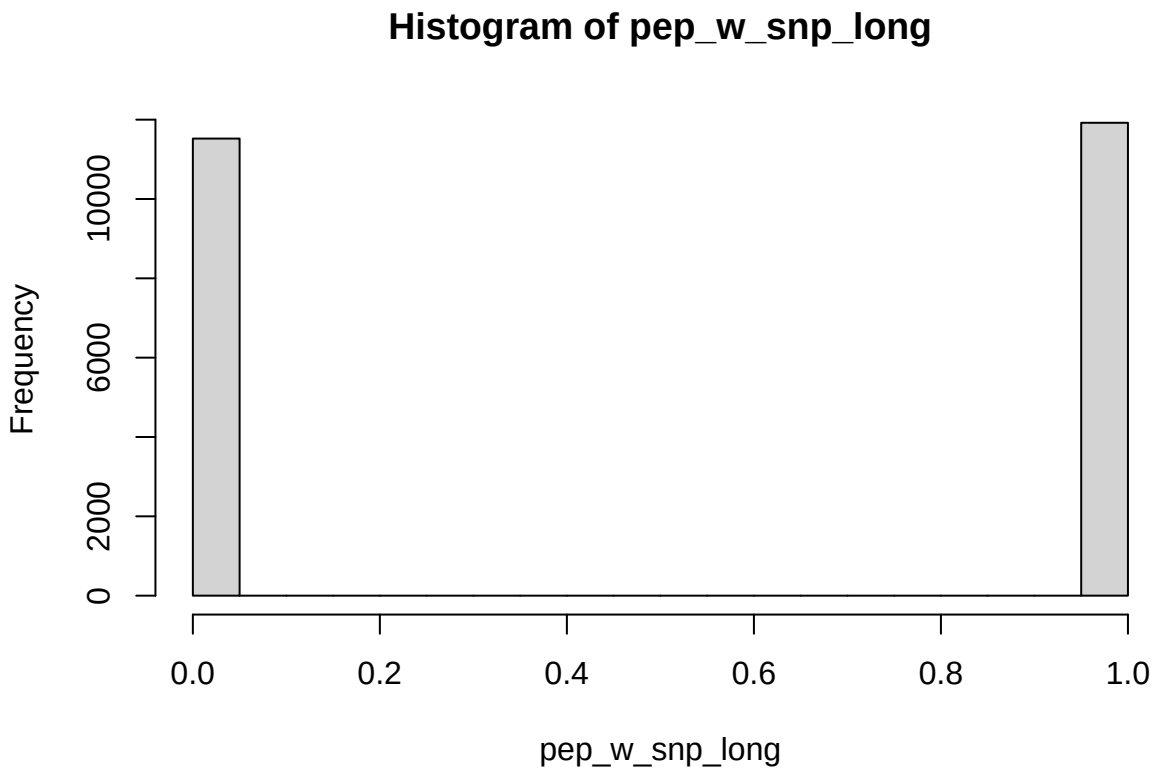



```
# Convert the data to matrix form, this is the original form
dat = data.frame(snp_ptm_id = rep(1:snp_ptm_id_count, patient_count),
                 patient = rep(1:patient_count, each = snp_ptm_id_count), peptide = pep_count,
                 snp = pep_w_snp_count, non_snp = pep_count - pep_w_snp_count, snp_ptm = pep_w_ptm_snp,
dat <- dat %>% mutate(effect_size = effect_size[snp_ptm_id])
dat = as_tibble(dat)
dat
```

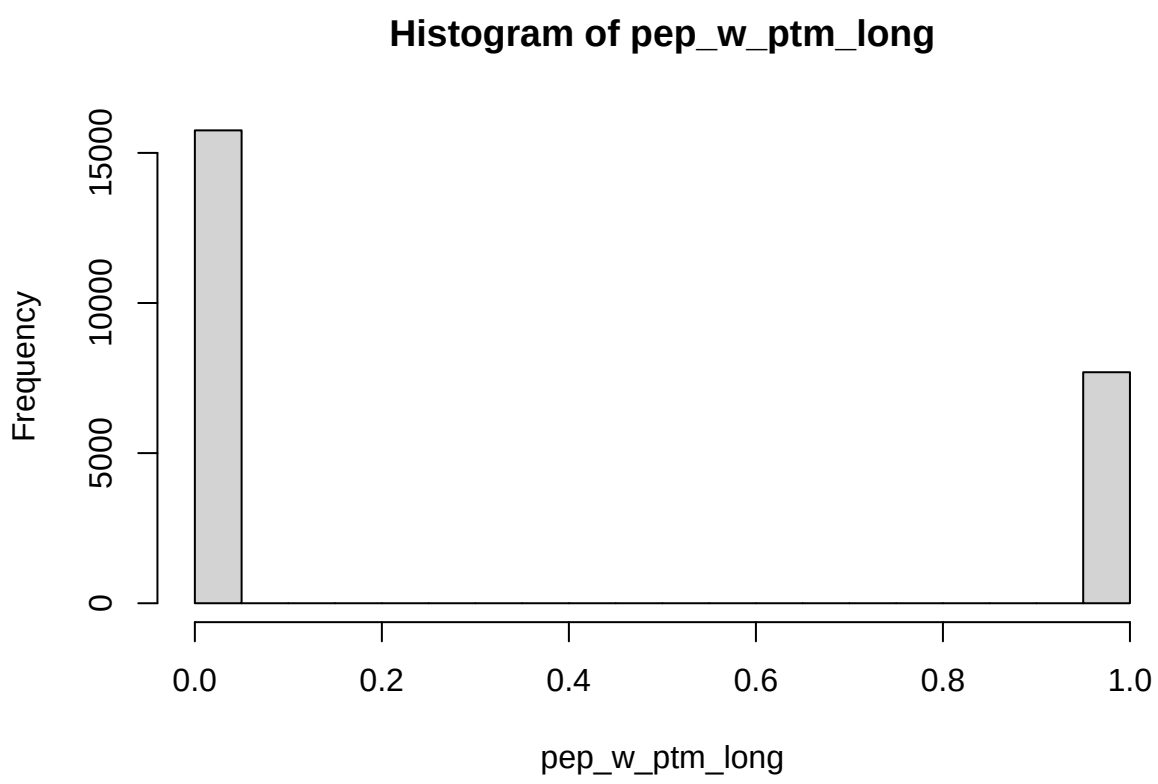
```
## # A tibble: 5,000 x 8
##    snp_ptm_id patient peptide   snp non_snp snp_ptm non_snp_ptm effect_size
##         <int>   <int>   <dbl> <dbl>   <dbl>   <int>       <int>       <dbl>
##  1          1       1       7     7       0       3           0         0
##  2          2       1       6     0       6       0           6         0.2
##  3          3       1      10    10       0       3           0         0
##  4          4       1       2     0       2       0           0         0
##  5          5       1       4     1       3       0           3         0
##  6          6       1       5     5       0       1           0         0
##  7          7       1       1     1       0       0           0         0
##  8          8       1       4     4       0       2           0         0
##  9          9       1       4     0       4       0           1         0
## 10         10       1       2     0       2       0           1         0.2
## # i 4,990 more rows
```

```
#x expanded as series of 1s (snp) and 0s (no snp) for new data table
pep_w_snp_long <- numeric(0)
```

```r
for (i in 1:(patient_count*snp_ptm_id_count)) {
  pep_w_snp_long <- c(pep_w_snp_long, rep(1, pep_w_snp_count[i]), rep(0, pep_count[i] - pep_w_snp_count
}
hist(pep_w_snp_long)
```




```r
#expanded ptm counts for new data table
pep_w_ptm_long <- numeric(0)
for (i in 1:(patient_count * snp_ptm_id_count)) {
  pep_w_ptm_long <- c(pep_w_ptm_long, rep(1, pep_w_ptm_snp[i]), rep(0, pep_w_snp_count[i] - pep_w_ptm_sn
}
hist(pep_w_ptm_long)
```




```
#original form expanded out by number of peptides
dat_long <- data.frame(snp_ptm_id = rep(rep(1:snp_ptm_id_count, patient_count), pep_count),
                       patient = rep(rep(1:patient_count, each = snp_ptm_id_count), pep_count),
                       snp = pep_w_snp_long, ptm = pep_w_ptm_long)
dat_long <- dat_long %>% mutate(effect_size = effect_size[snp_ptm_id])
dat_long <- as_tibble(dat_long)
dat_long
```

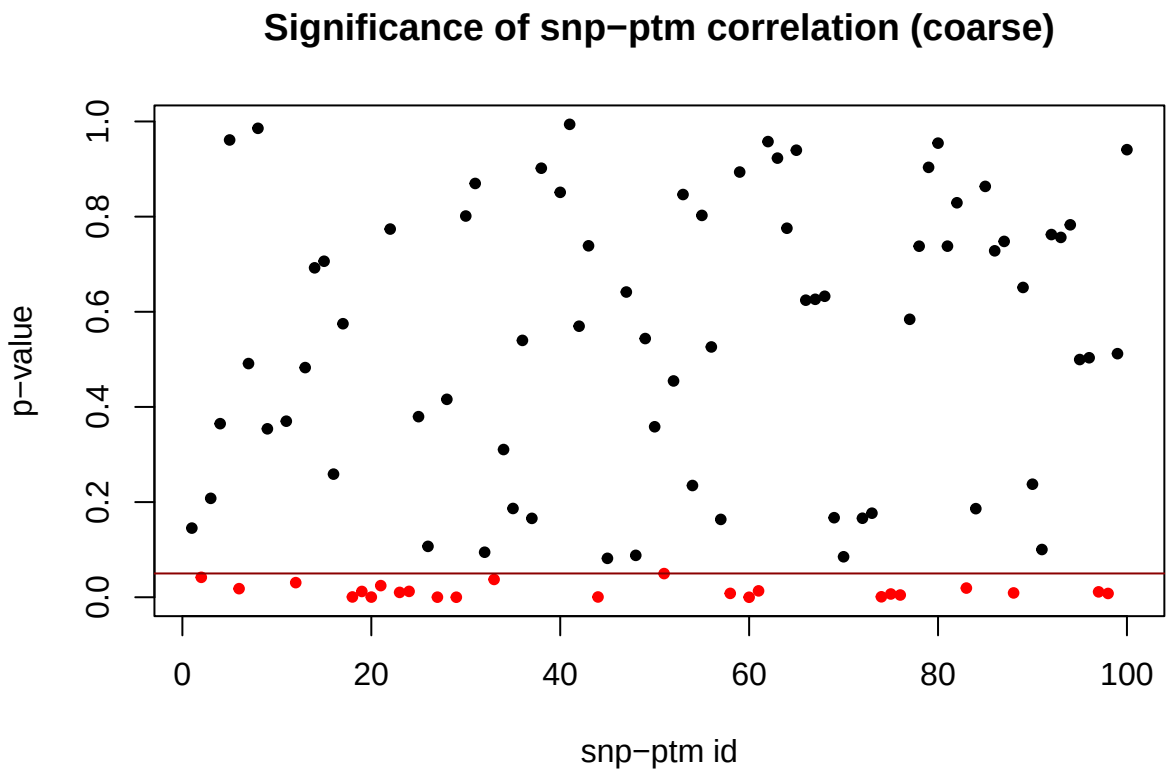
```
## # A tibble: 23,446 x 5
##    snp_ptm_id patient   snp   ptm effect_size
##         <int>   <int> <dbl> <dbl>       <dbl>
##  1          1       1     1     1         0
##  2          1       1     1     1         0
##  3          1       1     1     1         0
##  4          1       1     1     0         0
##  5          1       1     1     0         0
##  6          1       1     1     0         0
##  7          1       1     1     0         0
##  8          2       1     0     1         0.2
##  9          2       1     0     1         0.2
## 10          2       1     0     1         0.2
## # i 23,436 more rows
```

```
# Convert to a coarser form
dat_coarse = dat
```

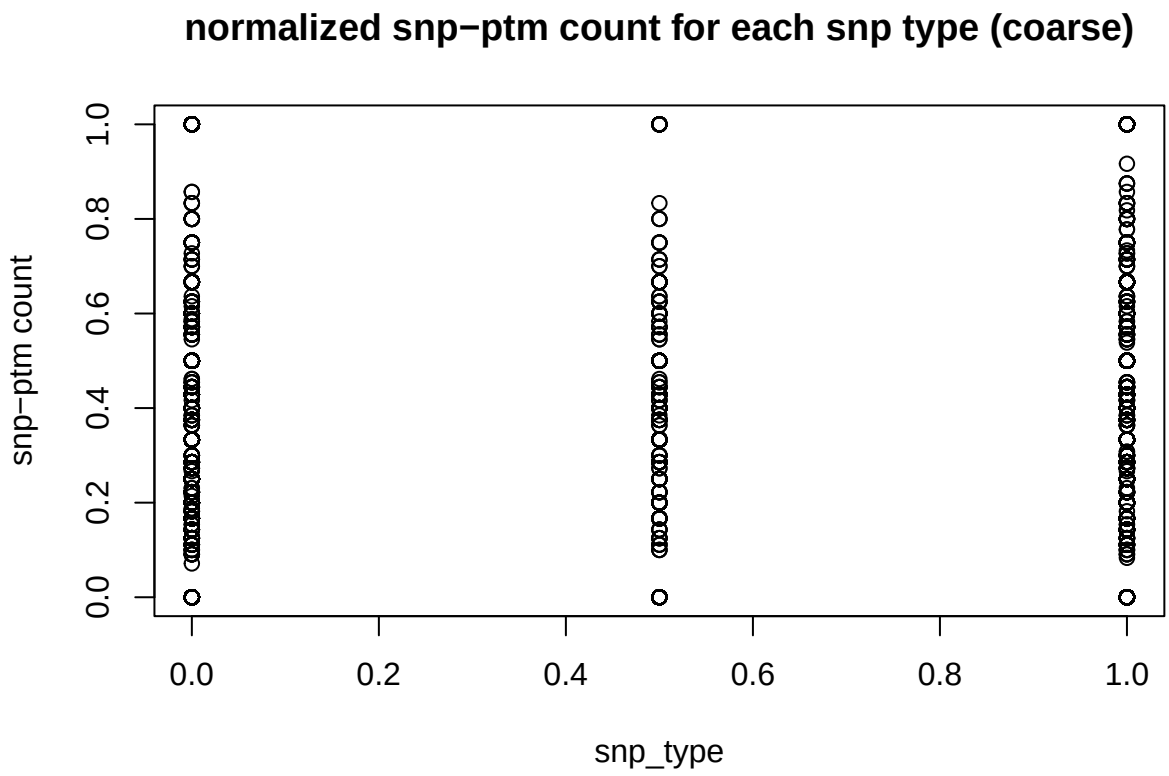
```r
dat_coarse$snp_type = cut(dat_coarse$snp/dat_coarse$peptide, breaks = c(-Inf, 0.33, 0.66, Inf), labels =
dat_coarse$snp_type <- (as.numeric(dat_coarse$snp_type) - 1) / 2
dat_coarse$effect_size = dat_coarse$effect_size
dat_coarse$snp_ptm <- (dat_coarse$snp_ptm + dat_coarse$non_snp_ptm) / dat_coarse$peptide
dat_coarse = dat_coarse %>% select(snp_ptm_id, patient, peptide, snp_type, snp_ptm, effect_size)


dat_coarse
```

```
## # A tibble: 5,000 x 6
##    snp_ptm_id patient peptide snp_type snp_ptm effect_size
##         <int>   <int>   <dbl>    <dbl>   <dbl>       <dbl>
##  1          1       1       7        1   0.429         0
##  2          2       1       6        0   1             0.2
##  3          3       1      10        1   0.3           0
##  4          4       1       2        0   0             0
##  5          5       1       4        0   0.75          0
##  6          6       1       5        1   0.2           0
##  7          7       1       1        1   0             0
##  8          8       1       4        1   0.5           0
##  9          9       1       4        0   0.25          0
## 10         10       1       2        0   0.5           0.2
## # i 4,990 more rows
```

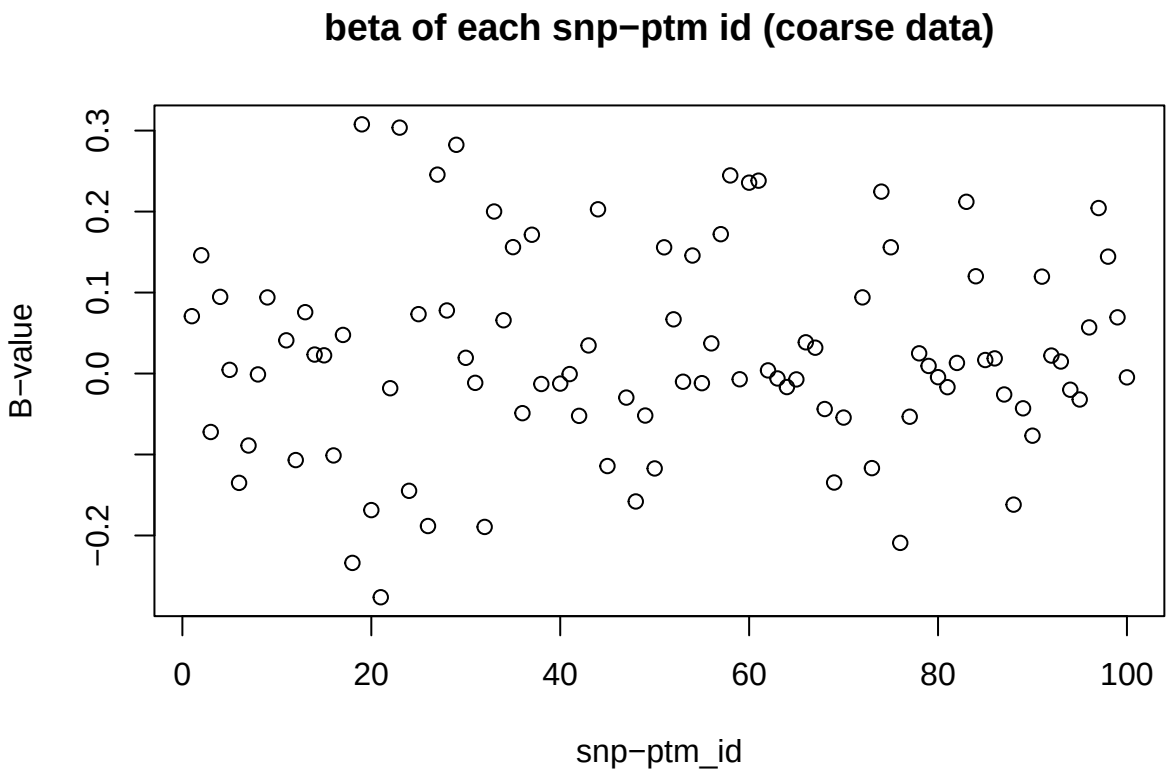
```r
#coarse data linear regression model
coarse_lm_b <- numeric(snp_ptm_id_count)
coarse_lm_pval <- numeric(snp_ptm_id_count)

for (idc in 1:snp_ptm_id_count) {
  subset_data <- subset(dat_coarse, snp_ptm_id == idc)
  if(all(subset_data$snp_ptm == 0) | all(subset_data$snp_type == 0)) {
    coarse_lm_b[idc] <- NA
    coarse_lm_pval[idc] <- NA
  } else {
    coarse_lm <- lm(snp_ptm ~ snp_type, data = subset_data)
    coarse_lm_pval[idc] <- summary(coarse_lm)$coefficients[2,4]
    coarse_lm_b[idc] <- coef(coarse_lm)[2]
  }
}

#coarse data snp-ptm ID vs p-value plot
sig_col <- ifelse(coarse_lm_pval < 0.05, "red", "black")
plot(seq_len(snp_ptm_id_count), coarse_lm_pval, main = "Significance of snp-ptm correlation (coarse)",
     xlab = "snp-ptm id", ylab = "p-value", pch = 20, col = sig_col)
abline(h = 0.05, col = "red4")
```



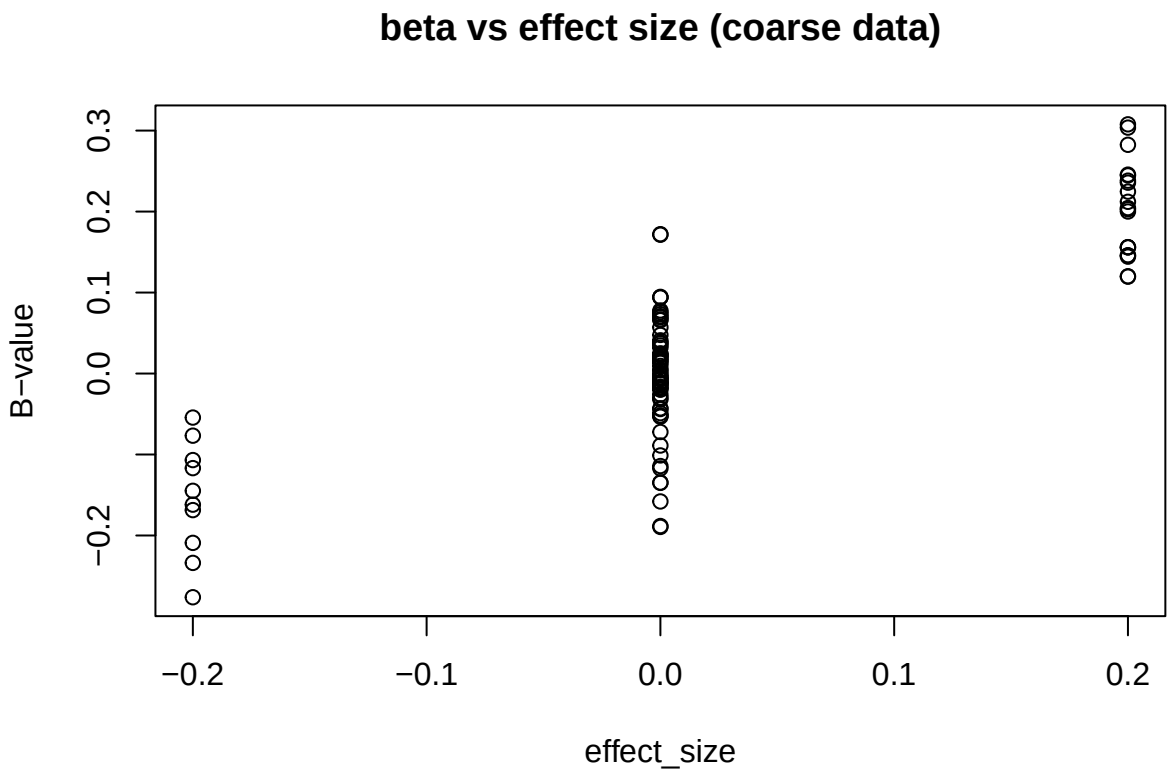

```
plot(dat_coarse$snp_type, dat_coarse$snp_ptm, main = "normalized snp-ptm count for each snp type (coarse
```



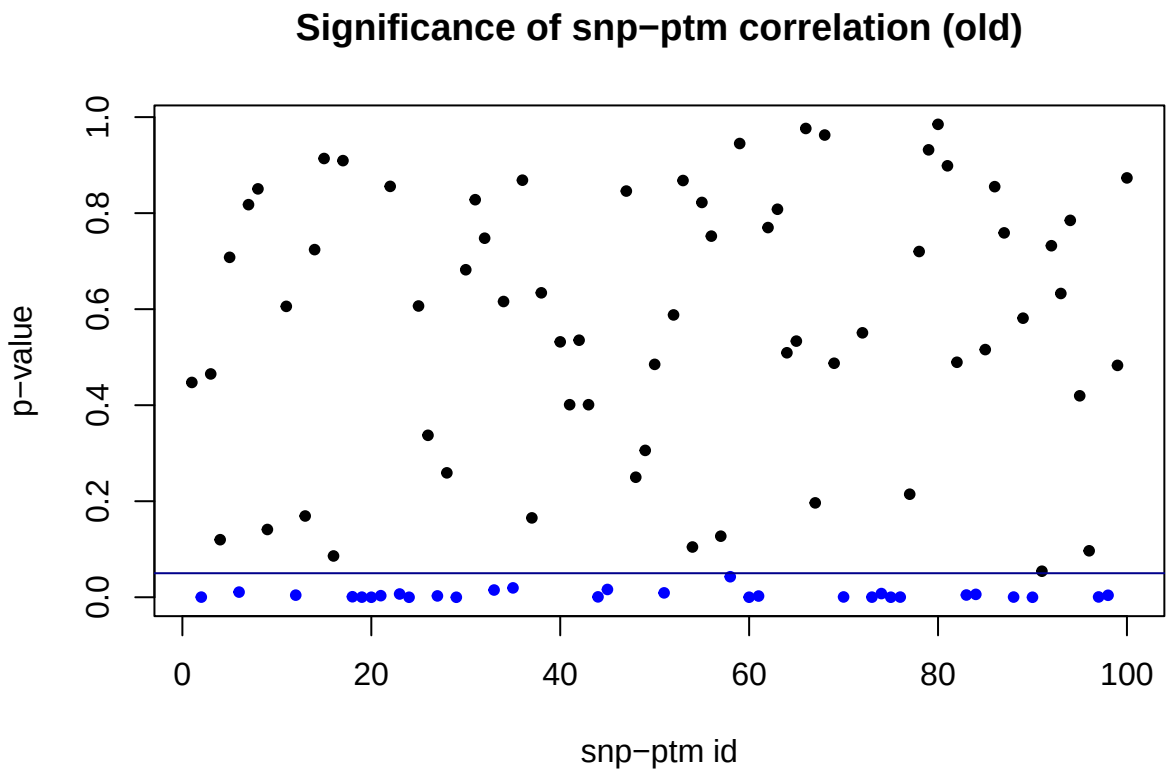

```
plot(1:100, coarse_lm_b, main = "beta of each snp-ptm id (coarse data)", xlab = "snp-ptm_id", ylab = "B
```



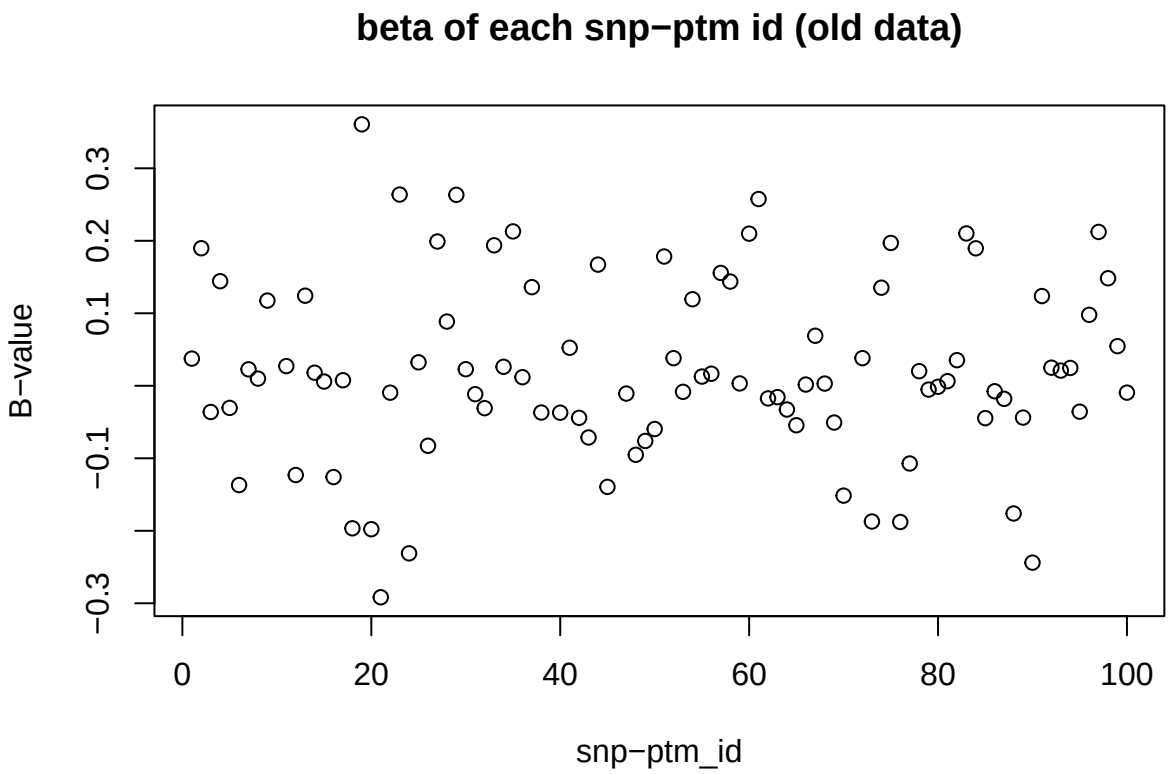

```
plot(effect_size, coarse_lm_b, main = "beta vs effect size (coarse data)", xlab = "effect_size", ylab =
```



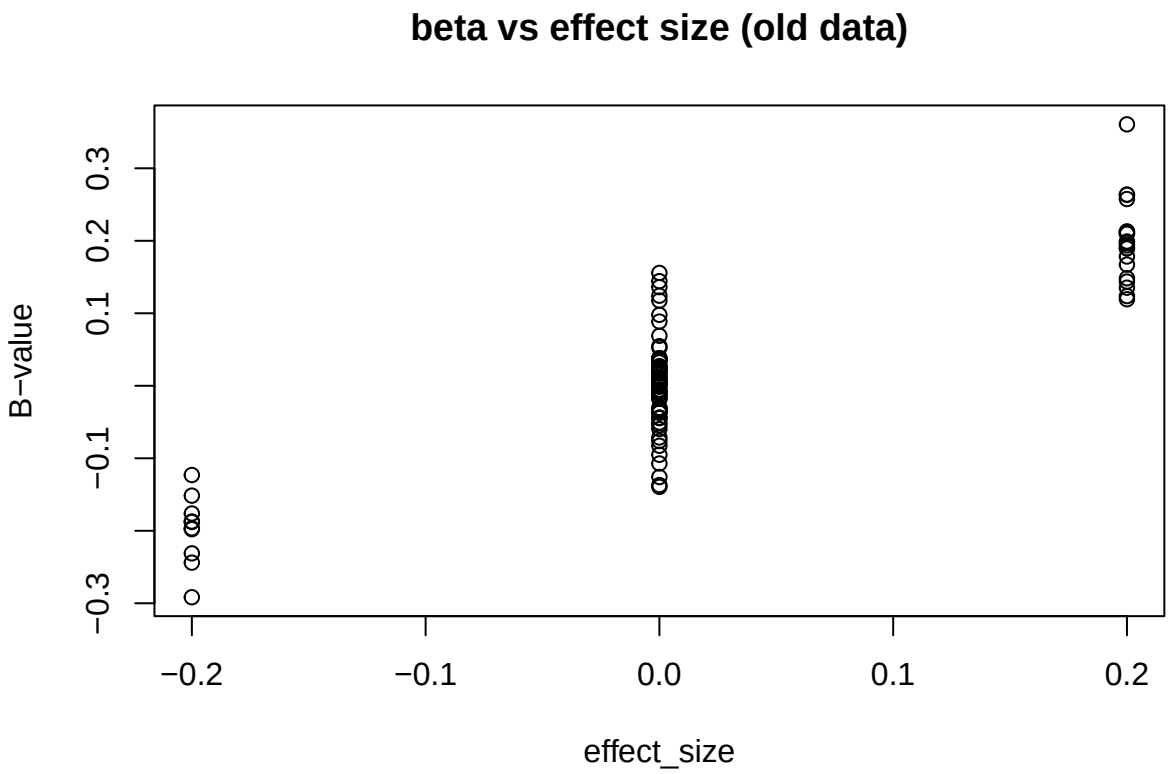

```r
#old data linear regression model
old_lm_b <- numeric(snp_ptm_id_count)
old_lm_pval <- numeric(snp_ptm_id_count)

for (idc in 1:snp_ptm_id_count) {
  subset_data <- subset(dat_long, snp_ptm_id == idc)
  if(all(subset_data$snp == 0) | all(subset_data$ptm == 0)) {
    old_lm_b[idc] <- NA
    old_lm_pval[idc] <- NA
  } else {
    old_lm <- lm(ptm ~ snp, data = subset_data)
    old_lm_pval[idc] <- summary(old_lm)$coefficients[2,4]
    old_lm_b[idc] <- coef(old_lm)[2]
  }
}

#old data snp-ptm ID vs p-value plot
sig_col_2 <- ifelse(old_lm_pval < 0.05, "blue", "black")
plot(seq_len(snp_ptm_id_count), old_lm_pval, main = "Significance of snp-ptm correlation (old)",
     xlab = "snp-ptm id", ylab = "p-value", pch = 20, col = sig_col_2)
abline(h = 0.05, col = "blue4")
```




```
plot(1:100, old_lm_b, main = "beta of each snp-ptm id (old data)", xlab = "snp-ptm_id", ylab = "B-value
```




```
plot(effect_size, old_lm_b, main = "beta vs effect size (old data)", xlab = "effect_size", ylab = "B-val
```


beta vs effect size (old data)
B−value
−0.3
−0.1
0.1
0.2
0.3
−0.2
−0.1
0.0
0.1
0.2
effect_size