

Tau specification analysis

Run on the cluster:

```
start <- Sys.time()

# function to generate the discrete data and covariates
# takes a RNG seed, sample size, number of predictors, lambda (root of the
# non-zero elements in the precision matrix), values to populate the
# extraneous covariate vector with, and boolean for whether the two groups
# should have the same covariance
# returns the continuous data, covariates, and true covariance matrix
generate_discrete <- function(n = 100, p = 10, lambda = 15,
                              cov1 = -0.1, cov2 = 0.1, same = T){

  # generating the precision matrix: Assume two discrete covariate levels, one
  # for each group
  Lam1 <- c(rep(lambda, 4), rep(0, p - 3))
  Lam2 <- c(rep(0, 4), rep(lambda, p - 3))

  # if same is true, the individuals in both groups will have the same
  # covariance matrix
  if (same) Lam2 <- Lam1

  # create covariance matrix for both groups
  Var1 <- solve(Lam1 %*% t(Lam1) + diag(rep(10, p + 1)))
  Var2 <- solve(Lam2 %*% t(Lam2) + diag(rep(10, p + 1)))

  # create the extraneous covariate; individuals in group j have a covariate
  # vector of length p with cov_j as the only entry, j\in {1,2}
  Z <- matrix(c(rep(cov1, n %/% 2), rep(cov2, n %/% 2)), n, p)

  # create the data matrix; individuals in group j are generated from a MVN with
  # 0 mean vector and covariance matrix Var_j, j\in {1,2}
  X1 <- MASS::mvrnorm(n %/% 2, rep(0, p + 1), Var1)
  X2 <- MASS::mvrnorm(n %/% 2, rep(0, p + 1), Var2)
  data_mat <- rbind(X1, X2)

  return(list(data = data_mat, covts = Z, true_precision = list(solve(Var1), solve(Var2))))
}

library(covdepGE)
library(doRNG)

set.seed(1)
```

```

# number of trials and values of bandwidth to try
n_trials <- 1000
tau <- c(0.01, 0.05, 0.1, 0.5, 1, 5)

# grid of slab variance
sbsq_grid <- c(0.01, 0.05, 0.1, 0.5, 1, 3, 7, 10)

n_cores <- parallel::detectCores() - 1
print(paste("N cores:", n_cores))
doParallel::registerDoParallel(n_cores)

results <- foreach (j = 1:n_trials, .packages = "covdepGE") %dornng%{

  # generate the data
  cont <- generate_discrete()
  X <- cont$data
  Z <- cont$covts

  # list for storing the results from each bandwidth and the data
  bw_res <- vector("list", length(tau) + 1)
  bw_res[[1]] <- cont
  names(bw_res) <- c("data", tau)

  # loop over each of the bandwidths
  for (bandwidth in tau){
    bw_res[[as.character(bandwidth)]] <- tryCatch(
      covdepGE(X, Z, sbsq = sbsq_grid, CS = T, tau = bandwidth, kde = F, scale = F,
        elbo_tol = 1e-12, alpha_tol = 1e-12, max_iter = 1e2, warnings = F),
      error = function(msg) as.character(msg))
  }

  # save the final results
  bw_res
}

doParallel::stopImplicitCluster()

Sys.time() - start

#save(results, file = "tau_specif_disc_indep_models.Rda")

```

```

library(ggplot2)
library(latex2exp)
library(covdepGE)
library(foreach)

load(paste0("C:/Users/jacob/OneDrive/Documents/TAMU/Research/An approximate ",
  "Bayesian approach to covariate dependent/covdepGE/dev/",
  "analyses_demos_experiments/hyperparameter_specification/",
  "tau_specif_disc_indep_models.Rda"))

n_trials <- 1000
tau <- c(0.01, 0.05, 0.1, 0.5, 1, 5)

```

```

names(results) <- paste("Trial", 1:n_trials)

# get all of the models
mods <- lapply(results, `[`, as.character(tau))

# look at the ELBO
elbo <- lapply(lapply(mods, lapply, `[`, "model_details"), lapply, `[`, "ELBO")
ELBO_vec <- unlist(elbo)
length(ELBO_vec[ELBO_vec < -5e4])

## [1] 39

length(ELBO_vec[ELBO_vec > -5e4])

## [1] 5961

summary(ELBO_vec[ELBO_vec < -5e4])

##      Min.    1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.196e+30 -9.991e+21 -1.801e+14 -9.201e+28 -6.970e+11 -3.394e+07

summary(ELBO_vec[ELBO_vec > -5e4])

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -48756  -46733  -46246  -46242  -45778  -44023

# filter out the models that blew up
elbo_stable <- lapply(elbo, function(elbo) elbo > -5e4)
mods_unstable <- lapply(1:n_trials, function(trial_ind) mods[[trial_ind]][!elbo_stable[[trial_ind]]])
names(mods_unstable) <- names(results)

# analyze the models that blew up

# filter out those of length 0
mods_unstable0 <- mods_unstable[lapply(mods_unstable, length) != 0]

# look at the mu distribution for those that blew up
summary(abs(unlist(lapply(mods_unstable0, lapply, `[`, "mu_matrices"))))

##      Min.    1st Qu.    Median      Mean   3rd Qu.      Max.
##  0.000e+00  0.000e+00  0.000e+00  6.459e+09  0.000e+00  3.765e+12

# visualize the unique graphs for a model that blew up
sapply(lapply(mods_unstable0, lapply, `[`, "model_details"), lapply, `[`, "ELBO")

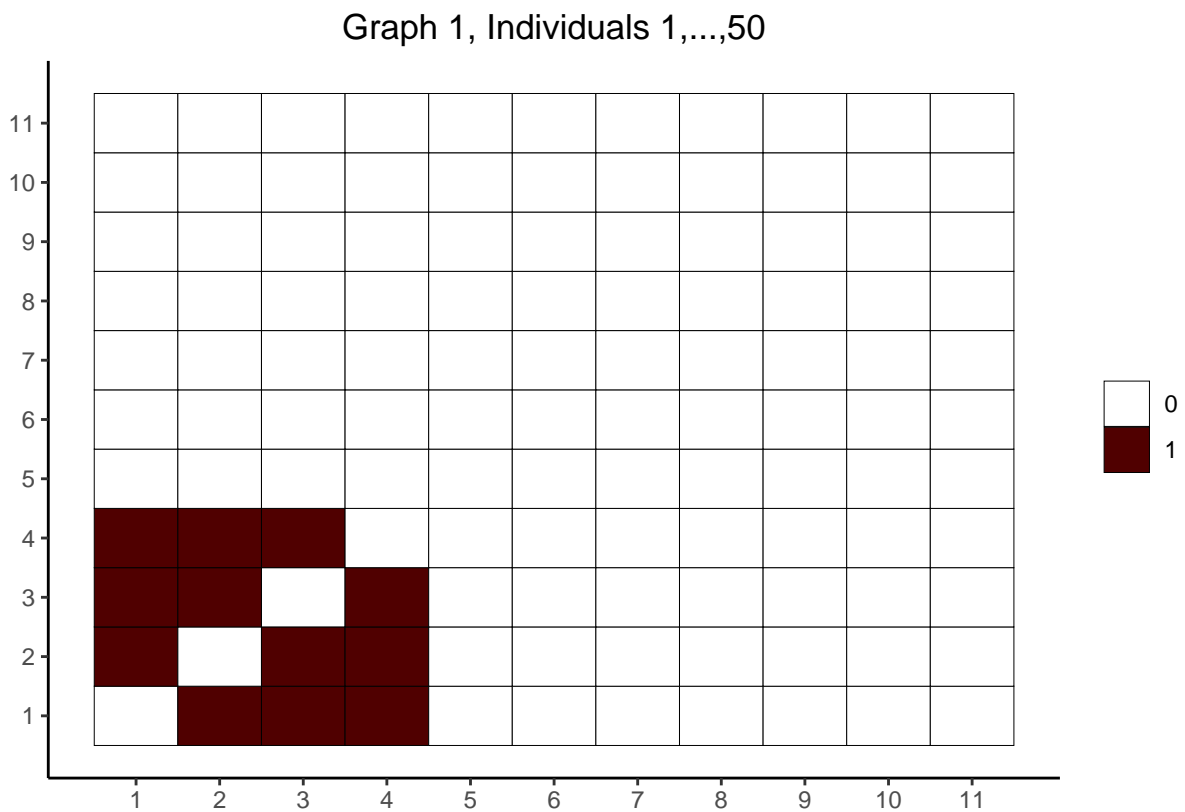
##      Trial 229 Trial 354    Trial 369    Trial 576    Trial 590
##  0.01 -33938355 -3.91362e+25 -1.196089e+30 -1.800523e+14 -6.9703e+11
##  0.05 -33938355 -3.91362e+25 -1.196089e+30 -1.800523e+14 -6.9703e+11

```

```
## 0.1 -33938321 -3.913618e+25 -1.196089e+30 -1.800522e+14 -697029895098
##      Trial 630 Trial 643      Trial 651      Trial 657      Trial 785
## 0.01 -579954449 -2.30788e+12 -1.470452e+12 -9.991491e+21 -23181925136
## 0.05 -579954449 -2.30788e+12 -1.470452e+12 -9.991491e+21 -23181925136
## 0.1 -579954165 -2.307865e+12 -1.470452e+12 -9.991488e+21 -23181911382
##      Trial 797      Trial 840      Trial 856
## 0.01 -1.453983e+25 -1.968497e+15 -4.106954e+18
## 0.05 -1.453983e+25 -1.968497e+15 -4.106954e+18
## 0.1 -1.453983e+25 -1.968497e+15 -4.106952e+18
```

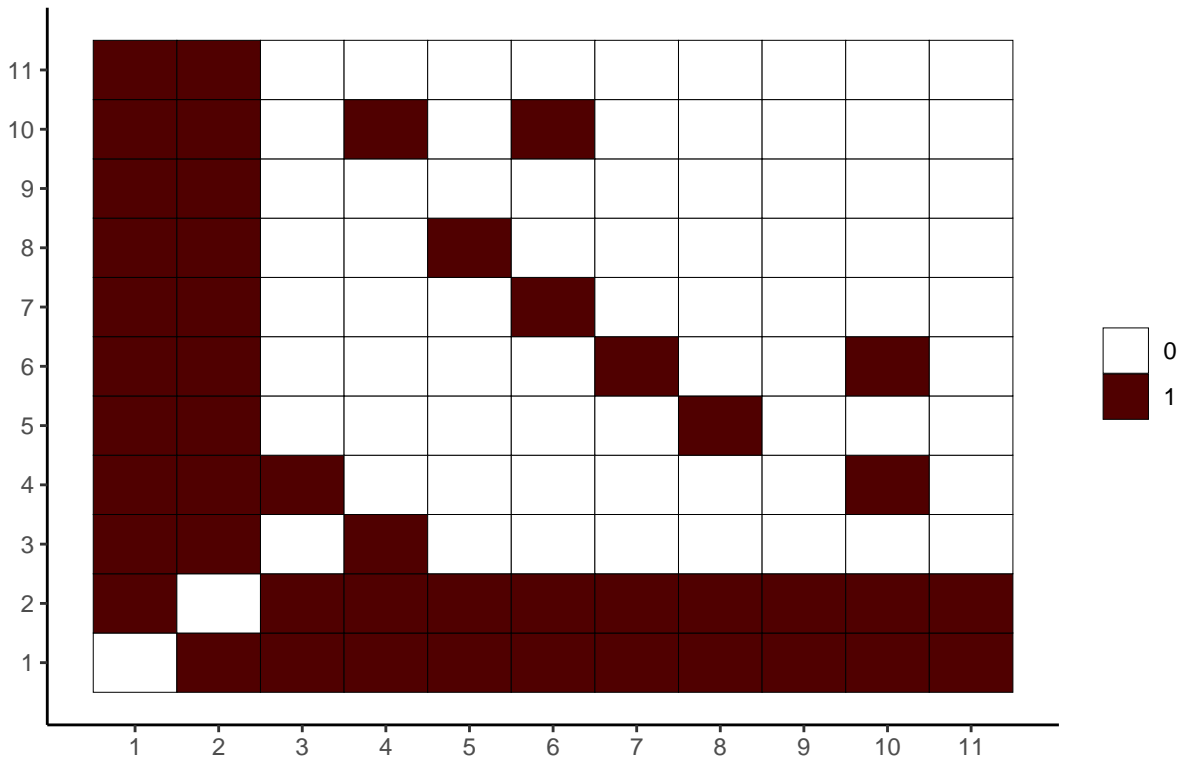
```
plot(mods_unstable0[["Trial 369"]]$`0.01`, title_sum = T)
```

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



```
##
## [[2]]
```

Graph 2, Individuals 51,...,100



```
# filter out the trials with instability
mods_stable <- mods[setdiff(names(mods), names(mods_unstable0))]
length(mods_stable)
```

```
## [1] 987
```

```
# get the true precision structure
true_prec <- results$`Trial 1`$data$true_precision[[1]]
true_graph <- (true_prec != 0) * 1
true_graph <- true_graph - diag(diag(true_graph))

# find the true number of 0's and 1's in one graph and across all structures
n <- nrow(results$`Trial 1`$data$data)
true1_graph <- sum(true_graph == 1)
true0_graph <- sum(true_graph == 0)
true1_tot <- true1_graph * n
true0_tot <- true0_graph * n

# find the estimated graphs for each model
mod_graphs <- lapply(mods_stable, lapply, `[`, "graphs")

# find the true 1's and 0's for each
true_pos <- lapply(
  mod_graphs, lapply, function(model) sum(sapply(
    model, function(graph) sum(graph == 1 & true_graph == 1))))
```

```

true_neg <- lapply(
  mod_graphs, lapply, function(model) sum(sapply(
    model, function(graph) sum(graph == 0 & true_graph == 0))))
true_pos_and_neg <- foreach(trial_ind = 1:length(true_pos)) %do% {
  res <- as.list(mapply(sum, true_pos[[trial_ind]], true_neg[[trial_ind]]))
  names(res) <- as.character(tau)
  res
}
names(true_pos_and_neg) <- names(true_pos)

# find sensitivity, specificity, and accuracy for each model
sens <- lapply(true_pos, lapply, `/%`, true1_tot)
spec <- lapply(true_neg, lapply, `/%`, true0_tot)
accu <- lapply(true_pos_and_neg, lapply, `/%`, true1_tot + true0_tot)

# get the summary of all of the sensitivities
summary(unlist(sens))

```

```

##      Min. 1st Qu.  Median      Mean 3rd Qu.     Max.
##         1         1         1         1         1         1

```

```

min_spec <- round(min(unlist(spec)), 2)
max_spec <- max(unlist(spec))

# get visualizations and summaries for each of the specificities and accuracies
for (bandwidth in as.character(tau)){
  cat("\n\nSpecificity results for tau =", bandwidth, "\n\nWeights: ")
  print(unique(as.numeric(results$`Trial 1`[[bandwidth]]$weights)))
  cat("\n")

  # get all of the specificities for the specified bandwidth
  spec_tau <- sapply(spec, `[`, bandwidth)

  # visualize the specificities
  plot <- ggplot() + geom_histogram(aes(x = spec_tau), binwidth = 0.01, color = "black") +
    coord_cartesian(xlim = c(min_spec, max_spec), ylim = c(0, 675)) +
    ggtitle(TeX(paste0("Specificity, $\tau = $", bandwidth))) + theme_bw() +
    theme(plot.title = element_text(hjust = 0.5))

  print(plot)
  print(summary(spec_tau))
  cat("\n", rep("-", 80), sep = "")
}

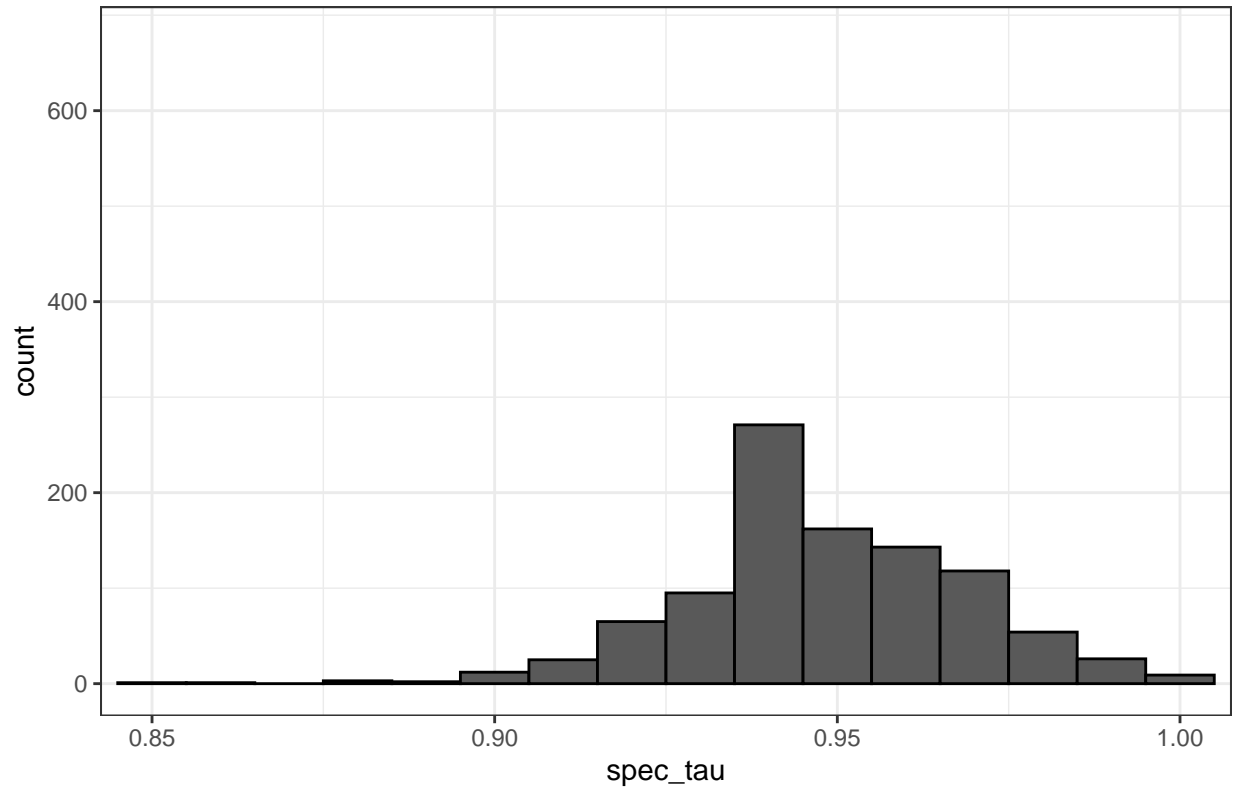
```

```

##
##
## Specificity results for tau = 0.01
##
## Weights: [1] 2 0

```

Specificity, $\tau = 0.01$



```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.8532  0.9358  0.9541  0.9495  0.9633  1.0000
```

```
##
```

```
## -----
```

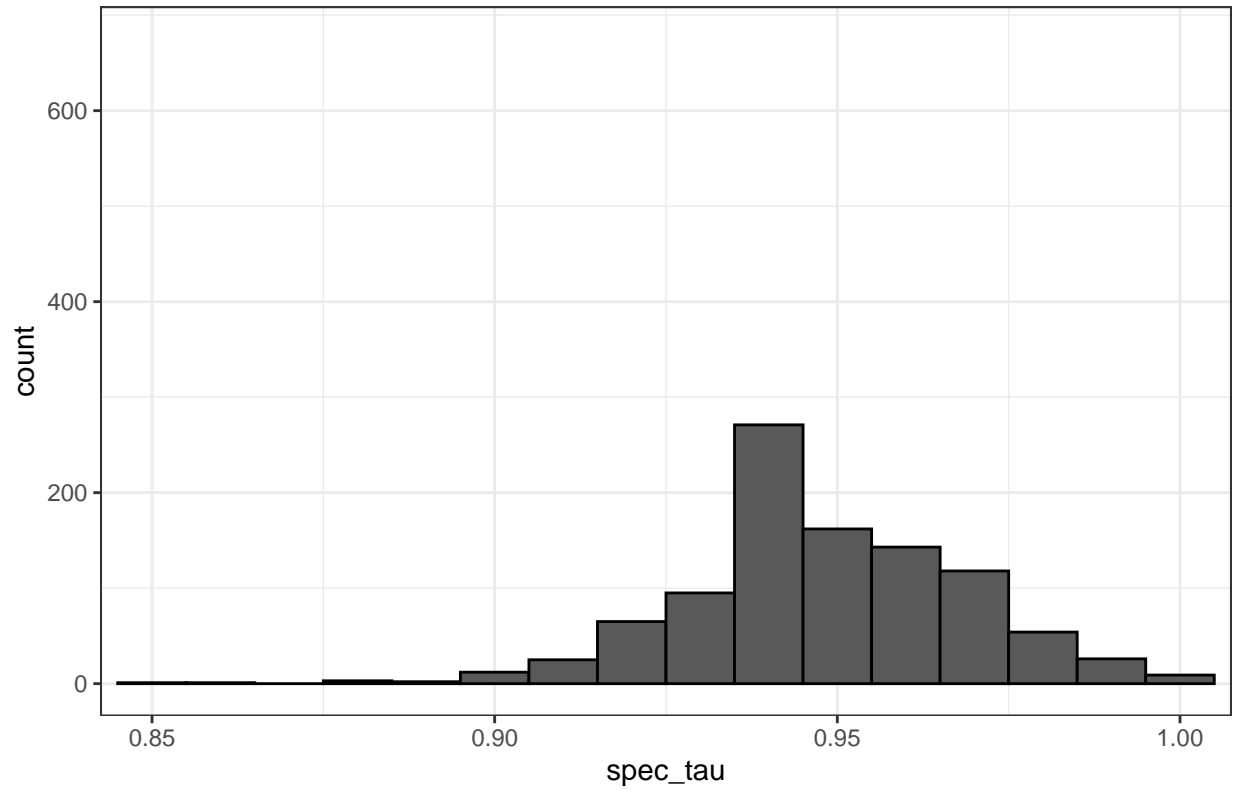
```
##
```

```
## Specificity results for tau = 0.05
```

```
##
```

```
## Weights: [1] 2.000000e+00 3.609703e-35
```

Specificity, $\tau = 0.05$



```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.8532  0.9358  0.9541  0.9495  0.9633  1.0000
```

```
##
```

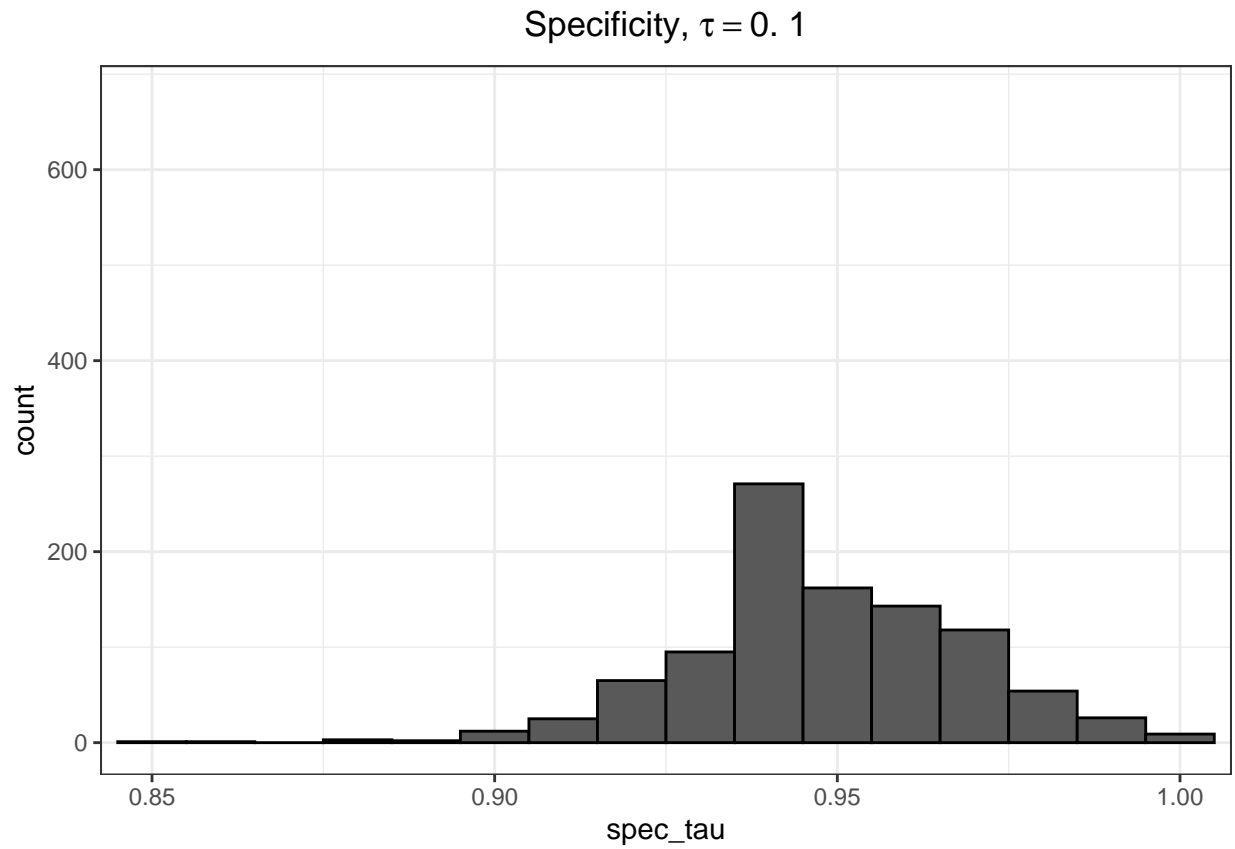
```
## -----
```

```
##
```

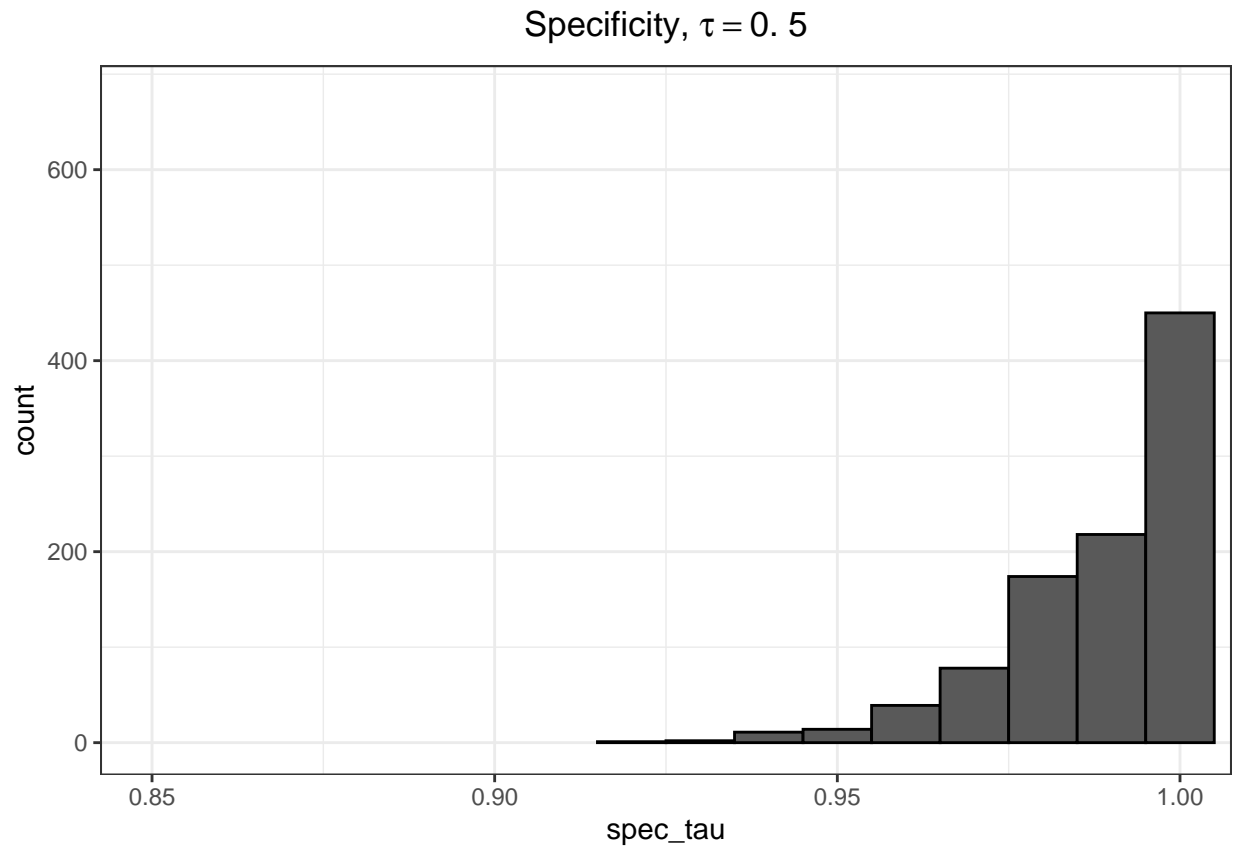
```
## Specificity results for tau = 0.1
```

```
##
```

```
## Weights: [1] 2.000000e+00 4.122307e-09
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.8532 0.9358 0.9541 0.9495 0.9633 1.0000
##
## -----
##
## Specificity results for tau = 0.5
##
## Weights: [1] 1.379949 0.620051
```



```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.9174  0.9817  0.9908  0.9896  1.0000  1.0000
```

```
##
```

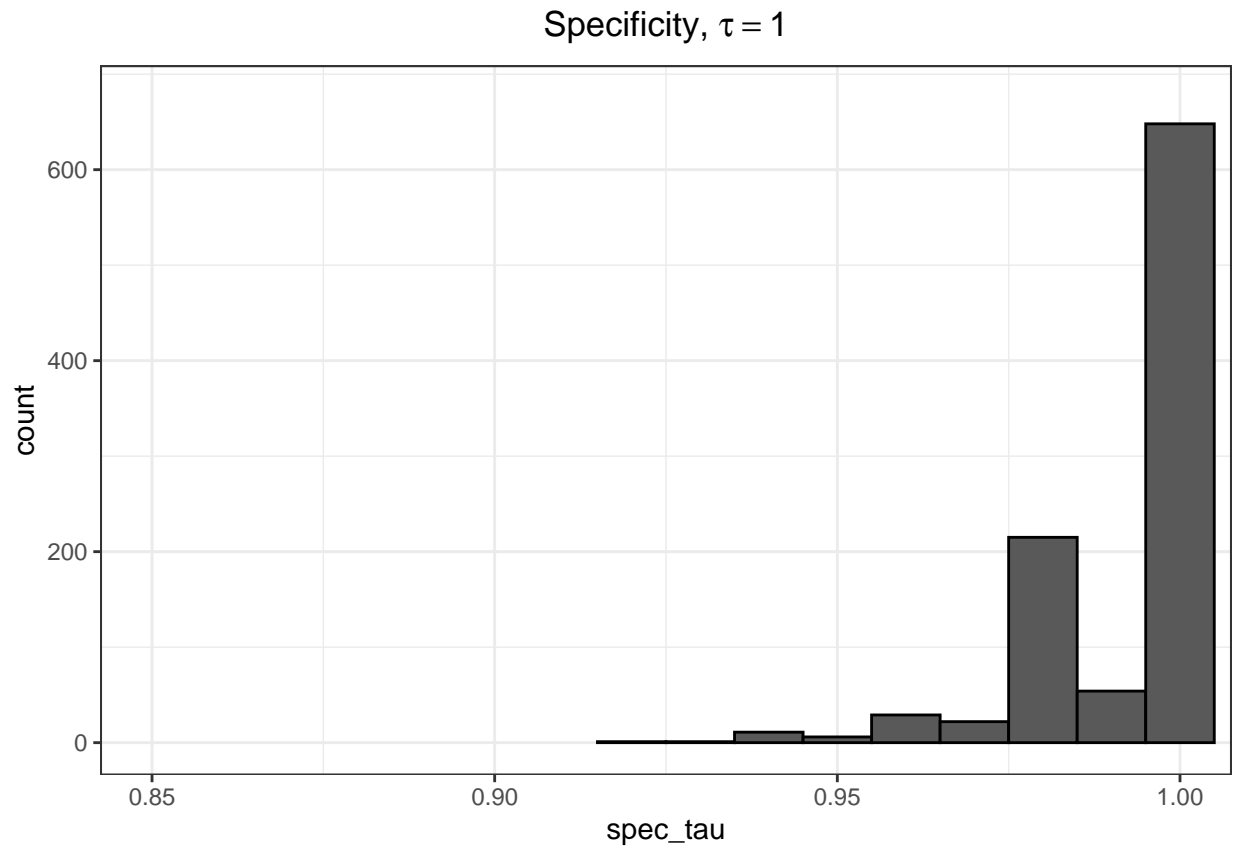
```
## -----
```

```
##
```

```
## Specificity results for tau = 1
```

```
##
```

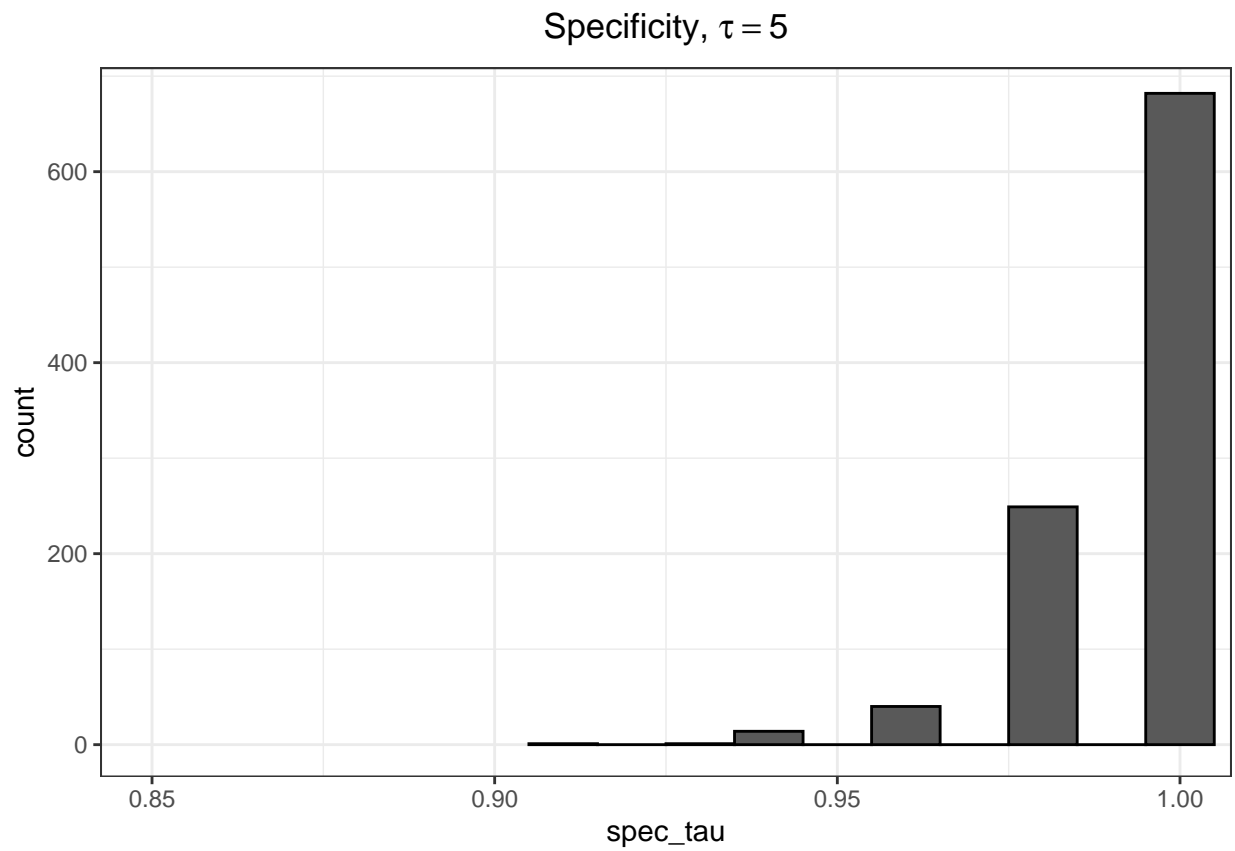
```
## Weights: [1] 1.099668 0.900332
```



```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.9174  0.9817  1.0000  0.9928  1.0000  1.0000
```

```
## -----
```

```
##
## Specificity results for tau = 5
##
## Weights: [1] 1.004 0.996
```



```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.9083  0.9817  1.0000  0.9929  1.0000  1.0000
##
## -----
```