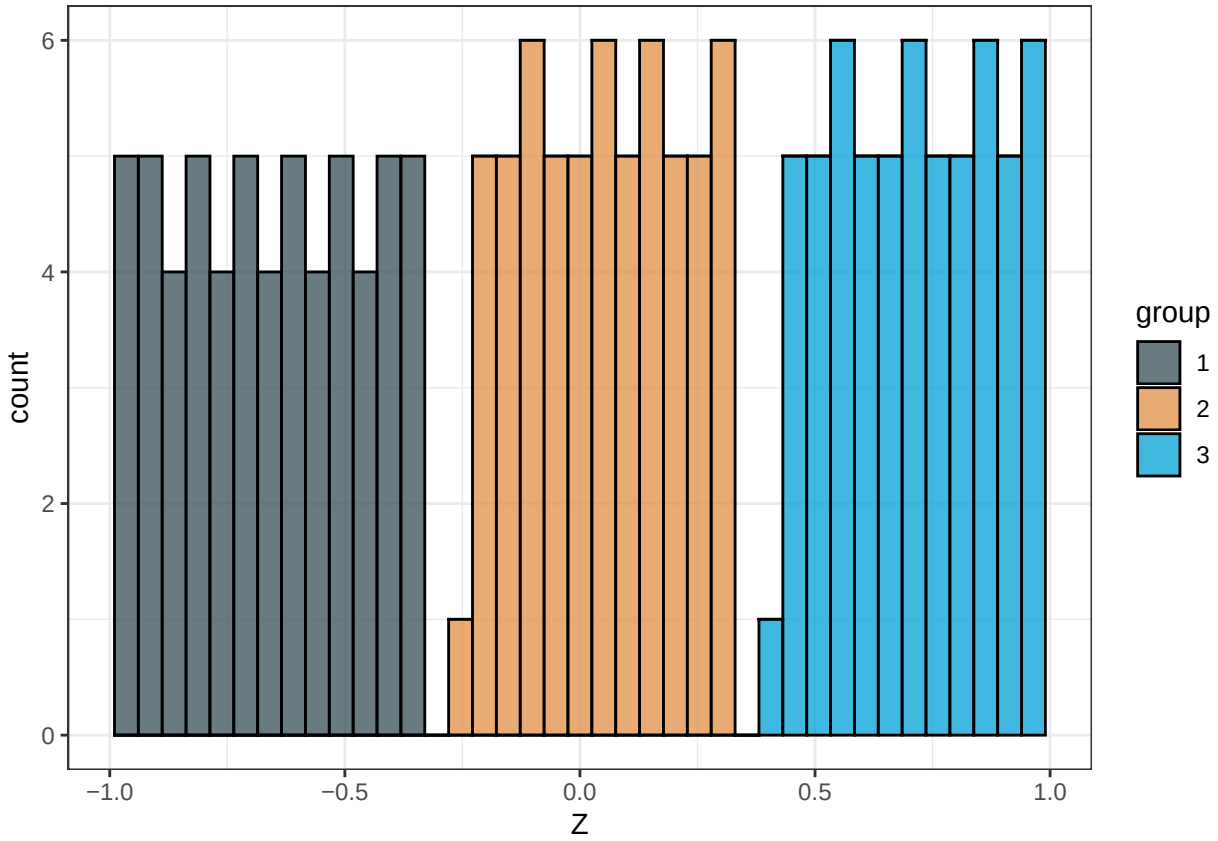
covdepGE with KDE demo

```
setwd("~/TAMU/Research/An approximate Bayesian approach to covariate dependent/covdepGE/dev")
source("generate_data.R")
library(covdepGE)
library(ggplot2)

# function to calculate silverman's rule of thumb for a data vector
silverman <- function(x){

  # apply and return silverman's rule of thumb
  sigma <- (0.9 * min(sd(x), IQR(x) / 1.35) * length(x)^(-0.2))
  return(sigma)
}


# function to evaluate model performance
# out is a covdepGE model
# true_graphs is a list of n true graphs
get_performance <- function(out, true_graphs){

  # calculate specificity (% true negatives), sensitivity (% true positives), and accuracy (% true nega
  true_negatives <- 0
  total_negatives <- 0
  true_positives <- 0
  total_positives <- 0
  for (j in 1:n){

    # fix an individual's true and estimated graph
    true_graph_j <- true_graphs[[j]]
    hat_graph_j <- out$graphs[[j]]

    # find total positives and negatives in the true graph
    pos <- true_graph_j == 1
    neg <- true_graph_j == 0
    total_positives <- total_positives + sum(pos)
    total_negatives <- total_negatives + sum(neg)

    # find total true positives and true negatives
    true_positives <- sum(true_positives + sum(hat_graph_j == 1 & pos))
    true_negatives <- sum(true_negatives + sum(hat_graph_j == 0 & neg))
  }

  sensitivity <- true_positives / total_positives
  specificity <- true_negatives / total_negatives
  accuracy <- (true_negatives + true_positives) / (total_positives + total_negatives)

  return(list(sensitivity = sensitivity, specificity = specificity, accuracy = accuracy))
```

```
}

# generate data
cont <- generate_continuous()
data_mat <- cont$data
Z <- cont$covts
n <- nrow(data_mat)
p <- ncol(data_mat)

# get the true covariance structure and use it to find the true graphs
true_cov <- cont$true_precision
true_graphs <- lapply(true_cov, function(vcov) (vcov != 0) * 1 - diag(p))

# visualize the distribution of the extraneous covariates
covts_df <- cbind.data.frame(group = as.factor(c(rep(1, 60), rep(2, 60), rep(3, 60))), Z = Z)
ggplot(covts_df, aes(Z, fill = group)) + geom_histogram(alpha = 0.75, bins = 40, color = "black") + ther
```




```
# find an estimate to the bandwidth of the covariates
(tau <- silverman(Z))
```

```
## [1] 0.1873059
```

```
# methodology with no KDE
out_no_KDE <- covdepGE(data_mat, Z, tau = tau, kde = F, CS = T, parallel = T, stop_cluster = F)
```

```
## Warning in covdepGE(data_mat, Z, tau = tau, kde = F, CS = T, parallel = T, : No
## registered workers detected; registering doParallel with 8 workers

## Warning in covdepGE(data_mat, Z, tau = tau, kde = F, CS = T, parallel =
## T, : Variable 1: CAVI did not converge in 100 iterations for 7/8 grid search
## candidates

## Warning in covdepGE(data_mat, Z, tau = tau, kde = F, CS = T, parallel = T, :
## Variable 1: final CAVI did not converge in 1000 iterations

## Warning in covdepGE(data_mat, Z, tau = tau, kde = F, CS = T, parallel =
## T, : Variable 2: CAVI did not converge in 100 iterations for 7/8 grid search
## candidates

## Warning in covdepGE(data_mat, Z, tau = tau, kde = F, CS = T, parallel =
## T, : Variable 3: CAVI did not converge in 100 iterations for 7/8 grid search
## candidates

## Warning in covdepGE(data_mat, Z, tau = tau, kde = F, CS = T, parallel = T, :
## Variable 3: final CAVI did not converge in 1000 iterations

## Warning in covdepGE(data_mat, Z, tau = tau, kde = F, CS = T, parallel =
## T, : Variable 4: CAVI did not converge in 100 iterations for 7/8 grid search
## candidates

## Warning in covdepGE(data_mat, Z, tau = tau, kde = F, CS = T, parallel = T, :
## Variable 4: final CAVI did not converge in 1000 iterations

## Warning in covdepGE(data_mat, Z, tau = tau, kde = F, CS = T, parallel =
## T, : Variable 5: CAVI did not converge in 100 iterations for 8/8 grid search
## candidates

## Warning in covdepGE(data_mat, Z, tau = tau, kde = F, CS = T, parallel = T, :
## Variable 5: final CAVI did not converge in 1000 iterations
```

```
get_performance(out_no_KDE, true_graphs)
```

```
## $sensitivity
## [1] 0.8333333
##
## $specificity
## [1] 0.7617486
##
## $accuracy
## [1] 0.7751111
```

```
# methodology with no KDE and tau = 0.56
out_no_KDE <- covdepGE(data_mat, Z, tau = 0.56, kde = F, CS = T, parallel = T,
                       stop_cluster = F)
```

```
## Detected 8 workers
```

```
## Warning in covdepGE(data_mat, Z, tau = 0.56, kde = F, CS = T, parallel =
## T, : Variable 2: CAVI did not converge in 100 iterations for 6/8 grid search
## candidates
```

```
get_performance(out_no_KDE, true_graphs)
```

```
## $sensitivity
## [1] 0.8166667
##
## $specificity
## [1] 0.9442623
##
## $accuracy
## [1] 0.9204444
```

```
# methodology with KDE
out_KDE <- covdepGE(data_mat, Z, CS = T, parallel = T)
```

```
## Detected 8 workers
```

```
get_performance(out_KDE, true_graphs)
```

```
## $sensitivity
## [1] 0.8166667
##
## $specificity
## [1] 0.9519126
##
## $accuracy
## [1] 0.9266667
```

```
# generate data
# cont <- generate_continuous(limits2 = c(.25, 0.55),
#                             limits3 = c(2.5, 10))
# data_mat <- cont$data
# Z <- cont$covts
# n <- nrow(data_mat)
# p <- ncol(data_mat)
#
# # get the true covariance structure and use it to find the true graphs
# true_cov <- cont$true_covariance
# true_graphs <- lapply(true_cov, function(vcov) (vcov != 0) * 1 - diag(p))
#
# # visualize the distribution of the extraneous covariates
# covts_df <- cbind.data.frame(group = as.factor(c(rep(1, 60), rep(2, 60), rep(3, 60))), Z = Z)
# ggplot(covts_df, aes(Z, fill = group)) + geom_histogram(alpha = 0.75, bins = 40, color = "black") + t
#
# # find an estimate to the bandwidth of the covariates
# (tau <- silverman(Z))
#
# # methodology with no KDE
```

```
# out_no_KDE <- covdepGE(data_mat, Z, tau = tau, kde = F, CS = T)
#
# get_performance(out_no_KDE, true_graphs)
#
# # methodology with no KDE and tau = 0.56
# out_no_KDE <- covdepGE(data_mat, Z, tau = 0.56, kde = F, CS = T)
#
# get_performance(out_no_KDE, true_graphs)
#
# # methodology with KDE
# out_KDE <- covdepGE(data_mat, Z, kde = T, CS = T)
#
# get_performance(out_KDE, true_graphs)
```