Simulations

The data generating model used was from Appendix 3 of Bowden et al (ref), and was as follows

$$U_i = \sum_{j=1}^{J} \phi_j G_{ij} + \epsilon_i^U \tag{1}$$

$$X_i = \sum_{j=1}^{J} \gamma_j G_{ij} + U_i + \epsilon_i^X \tag{2}$$

$$Y_i = \sum_{j=1}^{J} \alpha_j G_{ij} + \beta X_i + U_i + \epsilon_i^Y \tag{3}$$

for participants indexed by $i = 1, ..., N$, and genetic instruments indexed by $j = 1, ..., J$. The error terms $\epsilon_i^U, \epsilon_i^X$ and ϵ_i^Y were each drawn independently from standard normal distributions. The genetic effects on the exposure j are drawn from a uniform distribution between 0.03 and 0.1. Pleiotropic effects α_j and ϕ_j were set to zero if the genetic instrument was a valid instrumental variable. Otherwise (with probability 0.1, 0.2, or 0.3):

1. In Scenario 1 (balanced pleiotropy, InSIDE satisfied), the α_j parameter was drawn from a uniform distribution between -0.2 and 0.2.
2. In Scenario 2 (directional pleiotropy, InSIDE satisfied), the α_j parameter was drawn from a uniform distribution between 0 and 0.2.
3. In Scenario 3 (directional pleiotropy, InSIDE not satisfied), the ϕ_j parameter was drawn from a uniform distribution between -0.2 and 0.2.

The causal effect of the exposure on the outcome was either $\beta X = 0$ (null causal effect) or $\beta X = 0.1$ (positive causal effect). A total of 10 000 simulated datasets were generated for sample sizes of N = 10 000 and 20 [sic] participants. Only the summary data, that is genetic associations with the exposure and with the outcome and their standard errors as estimated by univariate regression on the genetic instruments in turn,were used by the analysis methods. In the two-sample setting, data were generated on 2N participants, and genetic associations with the exposure were estimated in the first N participants, and genetic associations with the outcome in the second N participants.

```
# Define function to create data generating model
# Arguments based on Bowden et al
wme_model_sim <- function(n_participants = as.integer(),
                          n_instruments = as.integer(),
                          n_datasets = as.integer(),
                          prop_invalid = 0.1,
                          causal_effect = TRUE,
                          balanced_pleio = TRUE,
                          InSIDE_satisfied = TRUE,
```

```
                        rand_error = TRUE,              # remove random errors for testing
                        beta_val = 0.1,                 # size of causal effect
                        allele_freq_min = 0.01,         # frequency of effect allele
                        allele_freq_max = 0.99,
                        gamma_min = 0.03,               # size of pleiotropic effects on exposure
                        gamma_max = 0.1,
                        alpha_min = -0.2,               # size of pleiotropic effects on outcome
                        alpha_max = 0.2,
                        phi_min = -0.2,                 # size of additional pleiotropic effects
                        phi_max = 0.2){                 # when InSIDE not satisfied


# Initialise blank lists to receive datasets for
# each of:
#     U (vector representing unmeasured confounding exposures per participant),
#     X (vector representing exposure:outcome associations estimated per participant)
#     Y (vector of gene:outcome association estimated per participant),
#     G (Matrices of Genotype data)
#
#     gamma (vector representing pleiotropic effects of each instrument on exposure)
#     alpha (vector representing pleiotropic effects of each instrument on outcome)
#     phi (vector representing additional pleiotropic effects of each instrument when InSIDE assumpti
U_list <- list()
X_list <- list()
Y_list <- list()
G_X_list <- list()
G_Y_list <- list()

gamma_list <- list()
alpha_list <- list()
phi_list <- list()


# Assign features common to all datasets
beta <- if_else(causal_effect == TRUE, # size of causal effect
                beta_val,
                0)


# Create N datasets by simulating genotype matrices with
# 1 row per participant, 1 column per genetic instrument
# Use these to estimate U, X + Y

for(n in 1:n_datasets){

  # Create error terms for U, X + Y per participant,
  # each drawn from standard normal distribution
  # unless random error turned off (for testing)

  ifelse(rand_error == TRUE,
         U_epsilon_vect <- rnorm(n = 2 * n_participants),
         U_epsilon_vect <- rep(0, 2 * n_participants))
```

```
ifelse(rand_error == TRUE,
       X_epsilon_vect <- rnorm(n = n_participants),
       X_epsilon_vect <- rep(0, n_participants))

ifelse(rand_error == TRUE,
       Y_epsilon_vect <- rnorm(n = n_participants),
       Y_epsilon_vect <- rep(0, n_participants))


## -- Create matrix of genotypes --
## 0 = reference, i.e. zero effect alleles,
## 1 = 1 effect allele, 2 = 2 effect alleles


# Probability of effect allele set per dataset
# for each instrument, default value set at
# random between 0.01-0.99 (i.e. both effect +
# reference are common alleles)
allele_freq_vect <- runif(n = n_instruments,
                          min = allele_freq_min,
                          max = allele_freq_max)



# Assign genotypes by sampling from binomial distribution
# twice (as two alleles) per participant with probability
# equal to frequency of effect allele
# Create twice as many genotypes as participants in sample
# to simulate 2 sample MR, i.e. first half used to estimate
# Gene:Exposure, second half used to estimate Gene:Outcome

G_mat <- matrix(rbinom(n = 2 * n_participants * n_instruments,
                       size = 2,
                       prob = rep(allele_freq_vect, 2 * n_participants)),
                nrow = 2 * n_participants,
                ncol = n_instruments,
                byrow = TRUE)


# Set which instruments invalid
invalid_instrument_vect <- rbinom(n = n_instruments,
                                  size = 1,
                                  prob = prop_invalid)


# Set genetic effects of each instrument on the exposure,
# drawn from uniform distribution, min/max as per Bowden
# et al
gamma_vect <- runif(n = n_instruments,
                    min = gamma_min,
                    max = gamma_max)
```

```r
# Set pleiotropic effects on outcome, Scenarios and
# min/max from Bowden et al
alpha_vect <- double() # Pleiotropic effects of instruments on outcome
phi_vect <- double() # Pleiotropic effects of confounders on outcome

for(j in 1:n_instruments){
  ifelse(invalid_instrument_vect[j] == FALSE,
         alpha_vect[j] <- 0,
         ifelse(balanced_pleio == TRUE,
                alpha_vect[j] <- runif(n = n_instruments,
                                       min = alpha_min,
                                       max = alpha_max),
                alpha_vect[j] <- runif(n = n_instruments,
                                       min = 0,
                                       max = alpha_max)
         )
  )

  # Assign default phi = 0 unless unbalanced pleiotropy &
  # InSIDE assumption not satisfied & genetic instrument invalid
  if(balanced_pleio == FALSE & InSIDE_satisfied == FALSE){
    ifelse(invalid_instrument_vect[j] == FALSE,
           phi_vect[j] <- 0,
           phi_vect[j] <- runif(n = 1,
                                min = phi_min,
                                max = phi_max)
    )

  }
  else{
    phi_vect[j] <- 0
  }
}

# Create vectors of estimates for U, X and Y per individual,
# i.e. Ui, Xi and Yi. Uses matrix inner product operator " %*%"
# https://stackoverflow.com/questions/22060515/the-r-operator
# http://matrixmultiplication.xyz/


Ui_vect <- G_mat %*% phi_vect + U_epsilon_vect

Xi_vect <- G_mat[1:n_participants, ] %*% gamma_vect +
  Ui_vect[1:n_participants, ] +
  X_epsilon_vect

Yi_vect <- G_mat[1:n_participants, ] %*% alpha_vect + (beta * Xi_vect) + Ui_vect[1:n_participants, ]


# Add vectors of estimates from this dataset to lists of
# estimates from all datasets
U_list[[n]] <- Ui_vect
```

```
    X_list[[n]] <- Xi_vect

    Y_list[[n]] <- Yi_vect

    G_X_list[[n]] <- G_mat[1:n_participants, ]

    G_Y_list[[n]] <- G_mat[(n_participants+1):(2*n_participants), ]

    alpha_list[[n]] <- alpha_vect

    gamma_list[[n]] <- gamma_vect

    phi_list[[n]] <- phi_vect


  }

  combined_list <- list(U = U_list,          # Estimates
                        X = X_list,
                        Y = Y_list,
                        G_X = G_X_list,     # Genotypes of 1st sample
                        G_Y = G_Y_list,     # Genotypes of 2nd sample
                        alpha = alpha_list, # Actual values for validating simulation
                        gamma = gamma_list,
                        phi = phi_list
                        )

  return(combined_list)
}
```

```
# Create plotting tibble with Mean/SD X + Y grouped by
# Dataset + instrument
extract_models_XY <- function(sim){

  output_list <- list()

  # Create linear models per dataset to get coefficients
  # for gene:exposure association (coeff_X_G) and gene:outcome
  # association (coeff_Y_G)
  for(dataset in 1:length(sim$X)){

    X <- sim$X[[dataset]]
    Y <- sim$Y[[dataset]]
    Instruments_X <- sim$G_X[[dataset]]
    Instruments_Y <- sim$G_Y[[dataset]]

    alpha <- sim$alpha[[dataset]]
    gamma <- sim$gamma[[dataset]]
    phi <- sim$phi[[dataset]]

    # Model for gene:exposure
    X_lm <- lm(X ~ Instruments_X)
    Xi_vect <- coef(summary(X_lm))[2:(ncol(Instruments_X) + 1), ]
```

```
    coeff_X_G_vect <- coef(summary(X_lm))[2:(ncol(Instruments_X) + 1), 1]
    SE_coeff_X_G_vect <- coef(summary(X_lm))[2:(ncol(Instruments_X) + 1), 2]

    # Model for gene:outcome
    Y_lm <- lm(Y ~ Instruments_Y)
    coeff_Y_G_vect <- coef(summary(Y_lm))[2:(ncol(Instruments_Y) + 1), 1]
    SE_coeff_Y_G_vect <- coef(summary(Y_lm))[2:(ncol(Instruments_Y) + 1), 2]

    output_list[[dataset]] <- as_tibble(list(dataset = dataset,
                                              Instrument = c(1:ncol(Instruments_X)),
                                              #Xi = Xi_vect,
                                              coeff_X_G = coeff_X_G_vect,
                                              coeff_X_G_SE = SE_coeff_X_G_vect,
                                              gamma = gamma,
                                              #Instrument_Y = c(1:ncol(Instruments_Y)),
                                              coeff_Y_G = coeff_Y_G_vect,
                                              coeff_Y_G_SE = SE_coeff_Y_G_vect,
                                              alpha = alpha,
                                              phi = phi),
                                         .name_repair = "unique")
  }

  return(output_list)

}
```

A series of test plots were used to verify that data were simulated as intended under the various conditions required:

```
# Check data produced in expected format
set.seed(1701)
sim_test_data_tib <- wme_model_sim(n_participants = 1000,
                                   n_instruments = 25,
                                   n_datasets = 2,
                                   prop_invalid = 0.3,
                                   rand_error = TRUE,
                                   causal_effect = TRUE,
                                   balanced_pleio = TRUE,
                                   InSIDE_satisfied = TRUE)
```

```
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
```

```
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
```

```
str(sim_test_data_tib)
```

```
## List of 8
##  $ U    :List of 2
##   ..$ : num [1:2000, 1] -1.139 -1.603 1.535 0.426 -2.832 ...
##   ..$ : num [1:2000, 1] -0.922 -0.823 -1.443 1.718 0.3 ...
##  $ X    :List of 2
##   ..$ : num [1:1000, 1] -1.06 1.75 1.65 1.51 -2.18 ...
##   ..$ : num [1:1000, 1] 0.725 3.021 -0.242 4.215 2.101 ...
##  $ Y    :List of 2
##   ..$ : num [1:1000, 1] 0.011 -1.325 0.784 0.184 -2.049 ...
##   ..$ : num [1:1000, 1] 2.75 -2.07 -1.44 3.88 2.25 ...
##  $ G_X  :List of 2
##   ..$ : int [1:1000, 1:25] 0 0 0 0 1 0 0 0 0 0 ...
##   ..$ : int [1:1000, 1:25] 0 1 0 1 0 2 1 1 1 1 ...
##  $ G_Y  :List of 2
##   ..$ : int [1:1000, 1:25] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ : int [1:1000, 1:25] 0 1 0 1 2 2 1 1 1 1 ...
##  $ alpha:List of 2
##   ..$ : num [1:25] 0 0 0.0756 0 0 ...
##   ..$ : num [1:25] 0 0 0 0 0 ...
##  $ gamma:List of 2
##   ..$ : num [1:25] 0.037 0.0401 0.0628 0.0721 0.0878 ...
##   ..$ : num [1:25] 0.0404 0.0486 0.0579 0.0956 0.0613 ...
##  $ phi  :List of 2
##   ..$ : num [1:25] 0 0 0 0 0 0 0 0 0 0 ...
##   ..$ : num [1:25] 0 0 0 0 0 0 0 0 0 0 ...
```

```
test_tib <- extract_models_XY(sim_test_data_tib)[[1]]

#head(test_tib)
test_tib
```

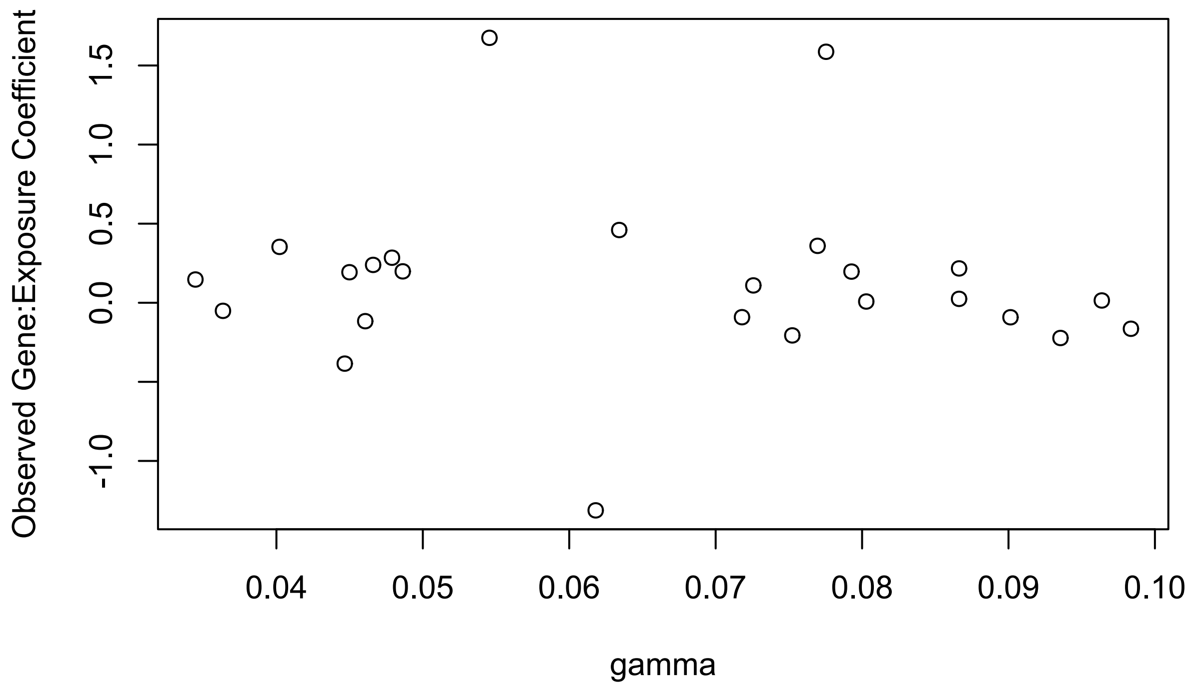
```
## # A tibble: 25 x 9
##    dataset Instrument coeff_X_G coeff_X_G_SE  gamma coeff_Y_G coeff_Y_G_SE
##      <int>      <int>     <dbl>        <dbl>  <dbl>     <dbl>        <dbl>
##  1       1          1   -0.0873        0.182  0.0370   -0.0890        0.182
##  2       1          2    0.0397        0.0662 0.0401    0.0391        0.0683
##  3       1          3    0.0737        0.0773 0.0628    0.0183        0.0802
##  4       1          4    0.0676        0.0728 0.0721    0.0774        0.0745
##  5       1          5    0.198         0.0983 0.0878    0.0590        0.102
##  6       1          6    0.0201        0.0930 0.0765    0.0581        0.0913
##  7       1          7    0.113         0.0670 0.0438   -0.0564        0.0697
##  8       1          8    0.0249        0.0672 0.0762    0.0660        0.0676
##  9       1          9    0.229         0.0989 0.0557    0.0216        0.101
```

```
## 10       1         10    0.0692       0.0789 0.0362    0.0271       0.0770
## # i 15 more rows
## # i 2 more variables: alpha <dbl>, phi <dbl>
```

```
# Check observed gene:exposure coefficients for each instrument
# (coeff_X_G) approximate true values (gamma) when a causal effect
# is present & a large number of participants are included
set.seed(1701)
sim_test_data_gamma_1 <- wme_model_sim(n_participants = 100,
                                       n_instruments = 25,
                                       n_datasets = 1,
                                       prop_invalid = 0.1,
                                       causal_effect = FALSE,
                                       rand_error = TRUE,
                                       balanced_pleio = TRUE,
                                       InSIDE_satisfied = TRUE)
```

```
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
```

```
test_plot_tib_gamma_1 <- extract_models_XY(sim_test_data_gamma_1)[[1]]

test_plot_tib_gamma_1 %>%
  select(gamma, coeff_X_G) %>%
  plot(.,
       ylab = "Observed Gene:Exposure Coefficient")
```

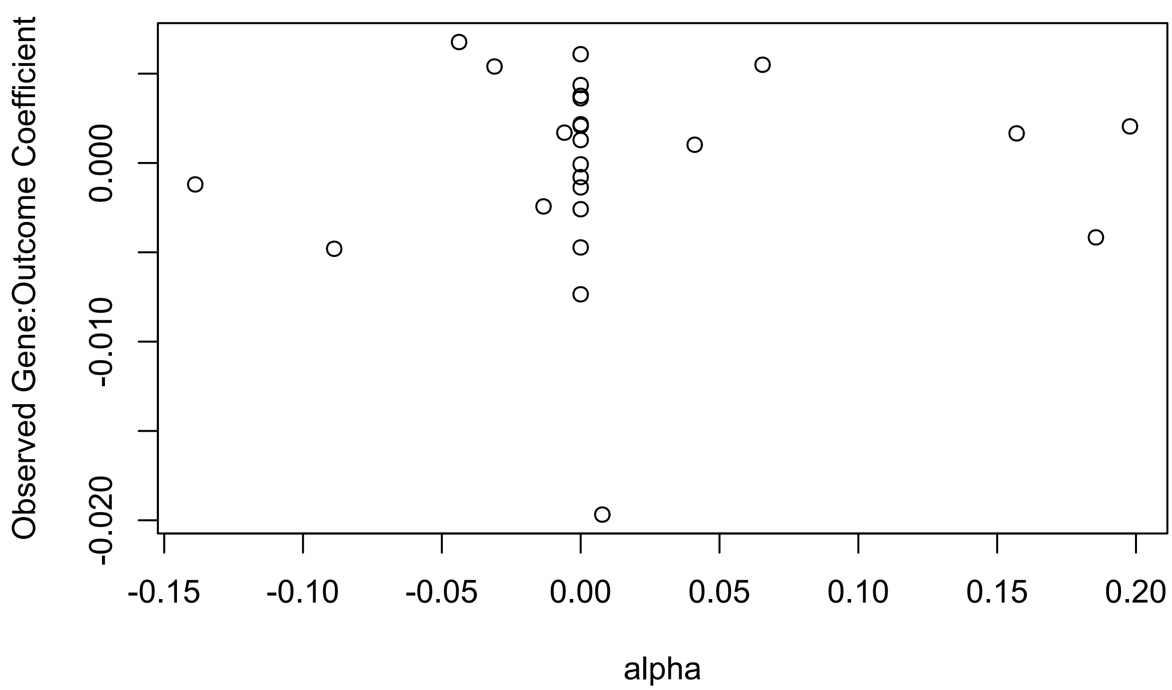



```r
# Check gene:outcome coefficients (coeff_Y_G) approximate
# pleiotropic effects (alphas) when no causal effect present
# N.B. cluster around alpha = 0 represents valid instruments with
# no pleiotropic effects

set.seed(1701)
sim_test_data_alpha <- wme_model_sim(n_participants = 10000,
                                     n_instruments = 25,
                                     n_datasets = 1,
                                     prop_invalid = 0.5,
                                     causal_effect = TRUE,
                                     balanced_pleio = TRUE,
                                     InSIDE_satisfied = TRUE,
                                     rand_error = FALSE)
```

```
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
```

```
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
```

```
test_plot_tib_alpha <- extract_models_XY(sim_test_data_alpha)[[1]]

test_plot_tib_alpha %>%
  select(alpha, coeff_Y_G) %>%
  plot(.,
       ylab = "Observed Gene:Outcome Coefficient")
```




```
# Check altering proportion of invalid instruments alters
# proportion of instruments displaying pleiotropic effects
# N.B. cluster around alpha = 0 represents valid instruments with
```

```r
# no pleiotropic effects

# No causal effect present
set.seed(1701)
sim_test_data_causal_0 <- wme_model_sim(n_participants = 10000,
                                        n_instruments = 25,
                                        n_datasets = 1,
                                        prop_invalid = 0.1,
                                        causal_effect = FALSE,
                                        balanced_pleio = TRUE,
                                        InSIDE_satisfied = TRUE,
                                        rand_error = FALSE,
                                        beta_val = 100,                 # size of causal effect
                                        allele_freq_min = 0.01,         # frequency of effect allele
                                        allele_freq_max = 0.99,
                                        gamma_min = 0.03,               # size of pleiotropic effects o
                                        gamma_max = 0.1,
                                        alpha_min = -0.2,               # size of pleiotropic effects o
                                        alpha_max = 0.2,
                                        phi_min = -0.2,                 # size of additional pleiotropi
                                        phi_max = 0.2)
```

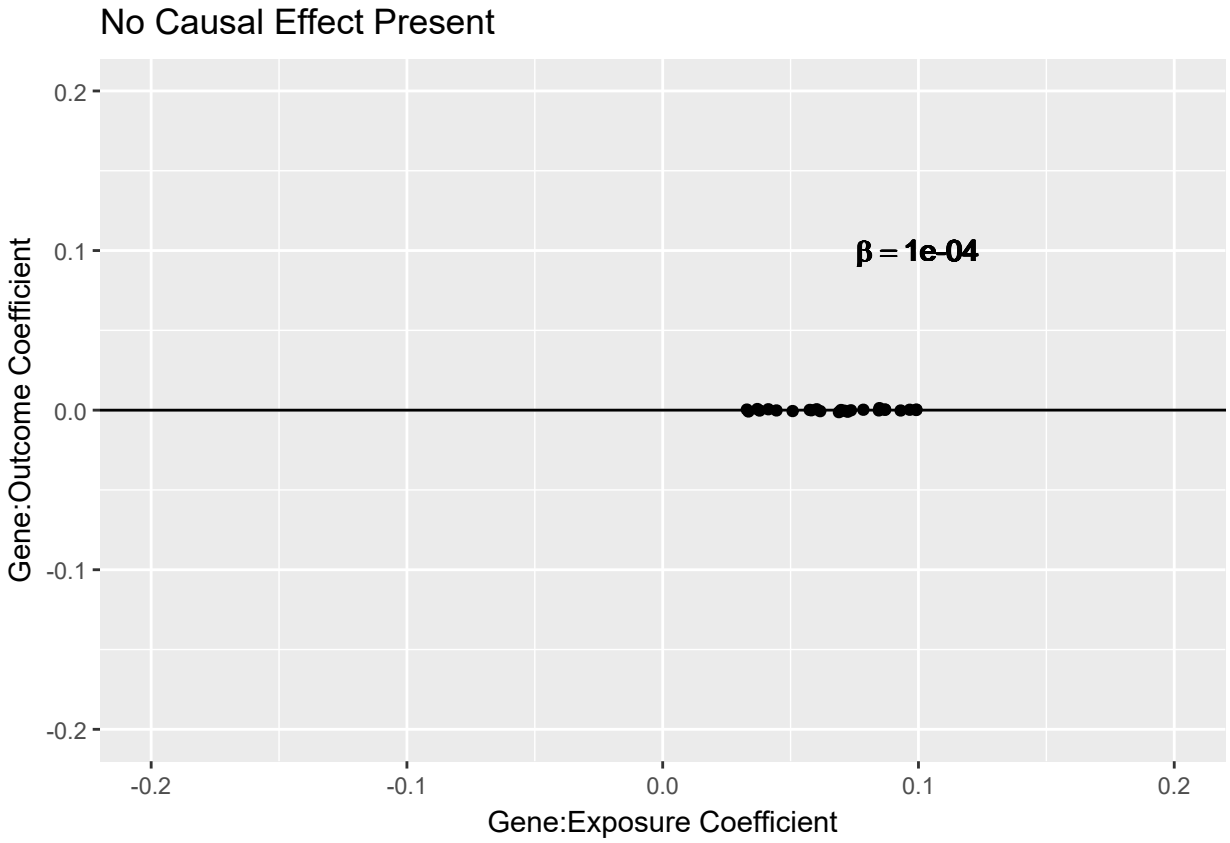
```
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
```

```r
# Causal effect present
set.seed(1701)
sim_test_data_causal_1 <- wme_model_sim(n_participants = 10000,
                                        n_instruments = 25,
                                        n_datasets = 1,
                                        prop_invalid = 0.1,
                                        causal_effect = TRUE,
                                        balanced_pleio = TRUE,
                                        InSIDE_satisfied = TRUE,
                                        rand_error = FALSE,
                                        beta_val = 100,                 # size of causal effect
                                        allele_freq_min = 0.01,         # frequency of effect allele
                                        allele_freq_max = 0.99,
                                        gamma_min = 0.03,               # size of pleiotropic effects o
                                        gamma_max = 0.1,
                                        alpha_min = -0.2,               # size of pleiotropic effects o
                                        alpha_max = 0.2,
                                        phi_min = -0.2,                 # size of additional pleiotropi
                                        phi_max = 0.2)
```

```
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
```

```r
test_plot_tib_causal_0 <- extract_models_XY(sim_test_data_causal_0)[[1]]
test_plot_tib_causal_1 <- extract_models_XY(sim_test_data_causal_1)[[1]]

test_plot_tib_causal_0 %>%
```

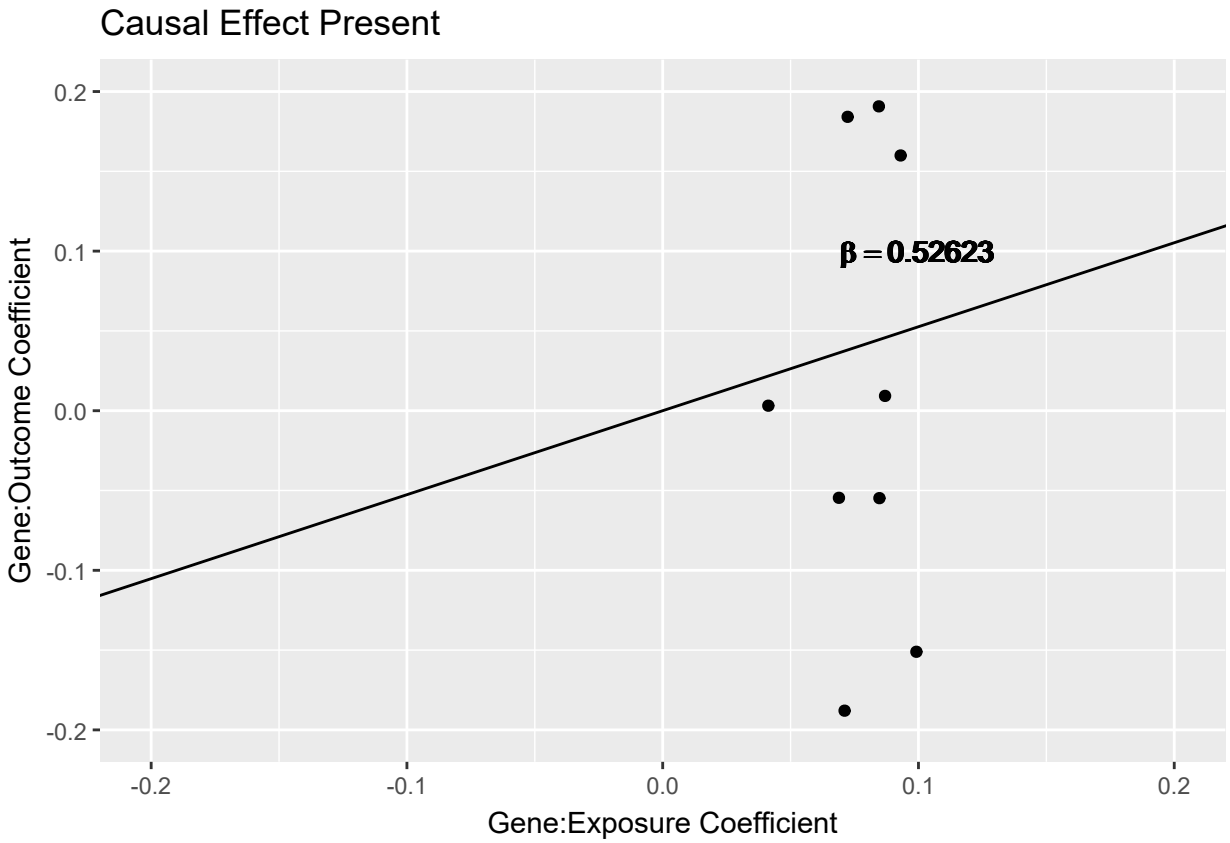
```
  mutate(Gradient = round(coefficients(lm(coeff_Y_G ~ 0 + coeff_X_G))[1], 5),
         Intercept = 0) %>%
  ggplot() +
  aes(x = coeff_X_G, y = coeff_Y_G) +
  geom_point() +
  geom_abline(aes(intercept = 0,
                  slope = Gradient),) +
  geom_text(aes(x = 0.1, # labels with gradient (causal effect estimate)
                y = 0.1,
                label = paste0("beta == ", Gradient)),
            parse = TRUE) +
  labs(title = "No Causal Effect Present",
       x = "Gene:Exposure Coefficient",
       y = "Gene:Outcome Coefficient") +
  xlim(-0.2, 0.2) +
  ylim(-0.2, 0.2)
```




```
test_plot_tib_causal_1 %>%
  mutate(Gradient = round(coefficients(lm(coeff_Y_G ~ 0 + coeff_X_G))[1], 5),
         Intercept = 0) %>%
  ggplot() +
  aes(x = coeff_X_G, y = coeff_Y_G) +
  geom_point() +
  geom_abline(aes(intercept = 0,
                  slope = Gradient),) +
```

```
  geom_text(aes(x = 0.1, # labels with gradient (causal effect estimate)
                y = 0.1,
                label = paste0("beta == ", Gradient)), # beta hat
            parse = TRUE) +
  labs(title = "Causal Effect Present",
       x = "Gene:Exposure Coefficient",
       y = "Gene:Outcome Coefficient") +
  xlim(-0.2, 0.2) +
  ylim(-0.2, 0.2)
```

```
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
# Check altering proportion of invalid instruments alters
# proportion of instruments displaying pleiotropic effects
# N.B. cluster around alpha = 0 represents valid instruments with
# no pleiotropic effects

# 10% of instruments invalid
set.seed(1701)
sim_test_data_inval_0.1 <- wme_model_sim(n_participants = 100000,
                                         n_instruments = 25,
                                         n_datasets = 1,
                                         prop_invalid = 0.1,
```

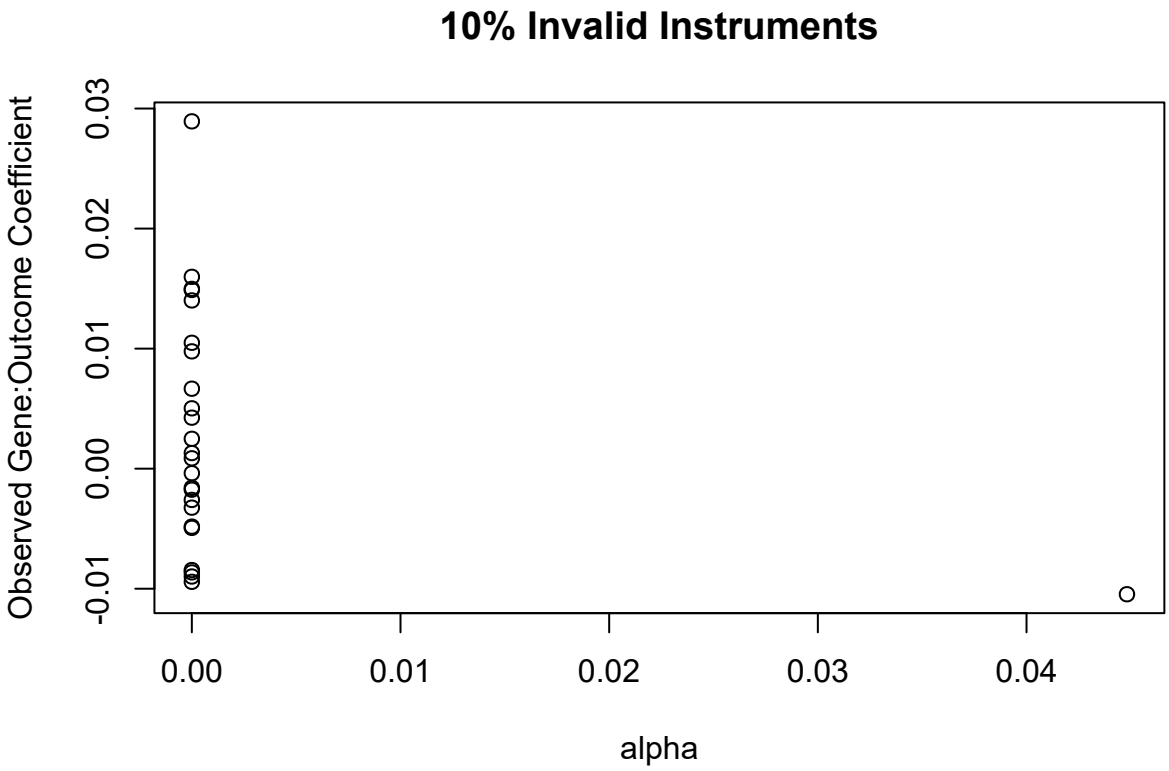
```r
                                    causal_effect = FALSE,
                                    balanced_pleio = TRUE,
                                    InSIDE_satisfied = TRUE)
```

```
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
```

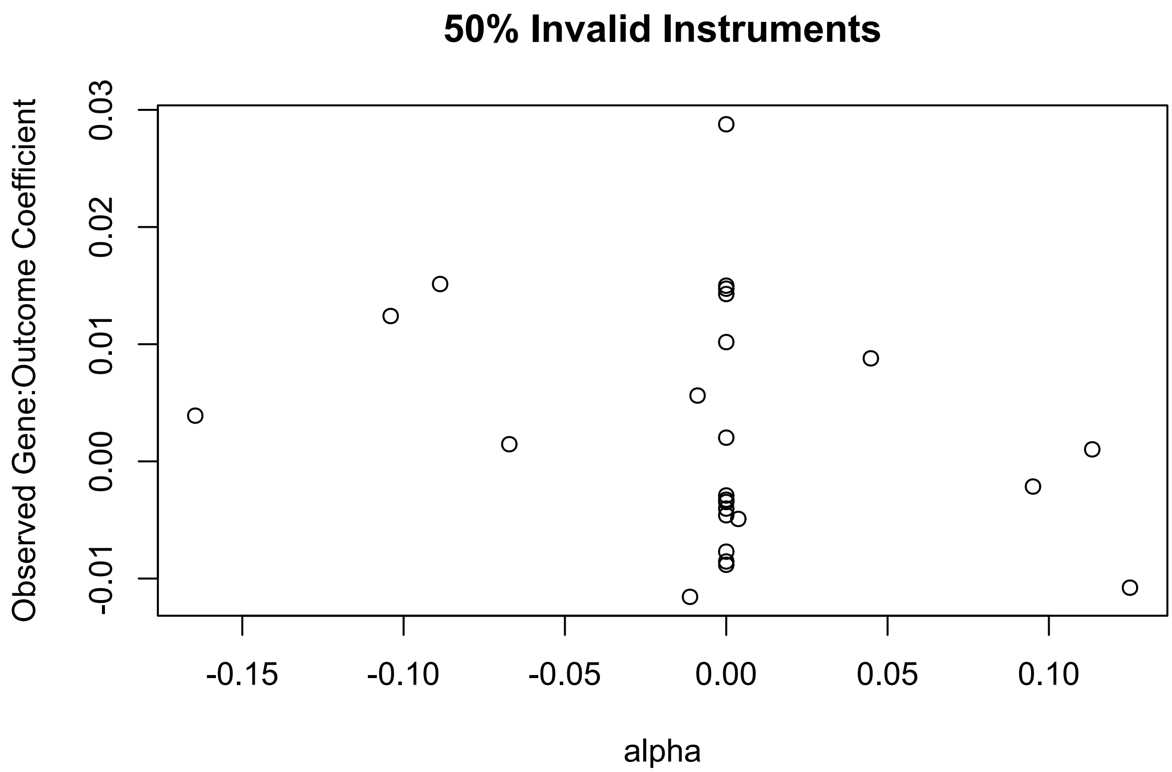
```r
# 50% of instruments invalid
set.seed(1701)
sim_test_data_inval_0.5 <- wme_model_sim(n_participants = 100000,
                                    n_instruments = 25,
                                    n_datasets = 1,
                                    prop_invalid = 0.5,
                                    causal_effect = FALSE,
                                    balanced_pleio = TRUE,
                                    InSIDE_satisfied = TRUE)
```

```
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = alpha_min, max =
## alpha_max): number of items to replace is not a multiple of replacement length
```

```r
test_plot_tib_inval_0.1 <- extract_models_XY(sim_test_data_inval_0.1)[[1]]
test_plot_tib_inval_0.5 <- extract_models_XY(sim_test_data_inval_0.5)[[1]]

test_plot_tib_inval_0.1 %>%
  select(alpha, coeff_Y_G) %>%
  plot(.,
       ylab = "Observed Gene:Outcome Coefficient",
       main = "10% Invalid Instruments")
```




```
test_plot_tib_inval_0.5 %>%
  select(alpha, coeff_Y_G) %>%
  plot(.,
       ylab = "Observed Gene:Outcome Coefficient",
       main = "50% Invalid Instruments")
```




```r
# Check violating InSIDE assumption results in distorted
# estimation of pleiotropic effects
# N.B. cluster around alpha = 0 represents valid instruments with
# no pleiotropic effects
set.seed(1701)
sim_test_data_phi_1 <- wme_model_sim(n_participants = 100000,
                                     n_instruments = 100,
                                     n_datasets = 1,
                                     prop_invalid = 0.3,
                                     causal_effect = FALSE,
                                     balanced_pleio = FALSE,
                                     InSIDE_satisfied = FALSE)
```

```
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
```

```
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
```

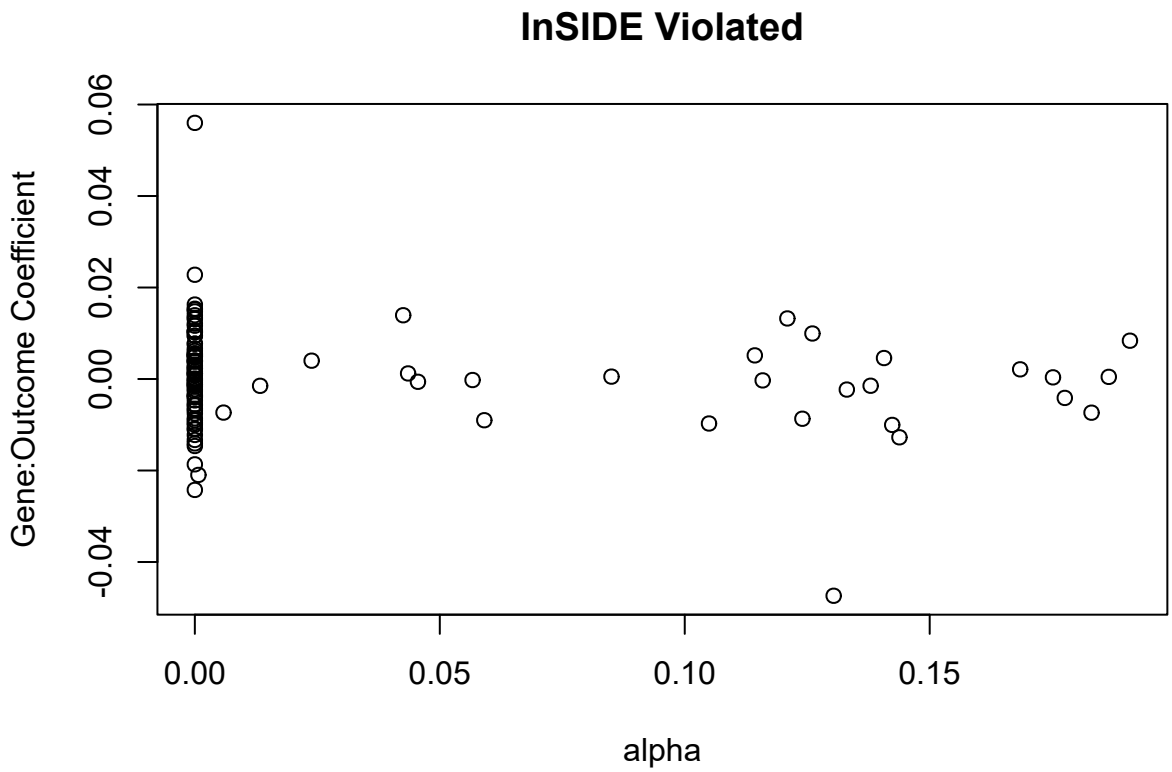
```r
set.seed(1701)
sim_test_data_phi_0 <- wme_model_sim(n_participants = 100000,
                                     n_instruments = 100,
                                     n_datasets = 1,
                                     prop_invalid = 0.3,
                                     causal_effect = FALSE,
                                     balanced_pleio = FALSE,
                                     InSIDE_satisfied = TRUE)
```

```
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
```

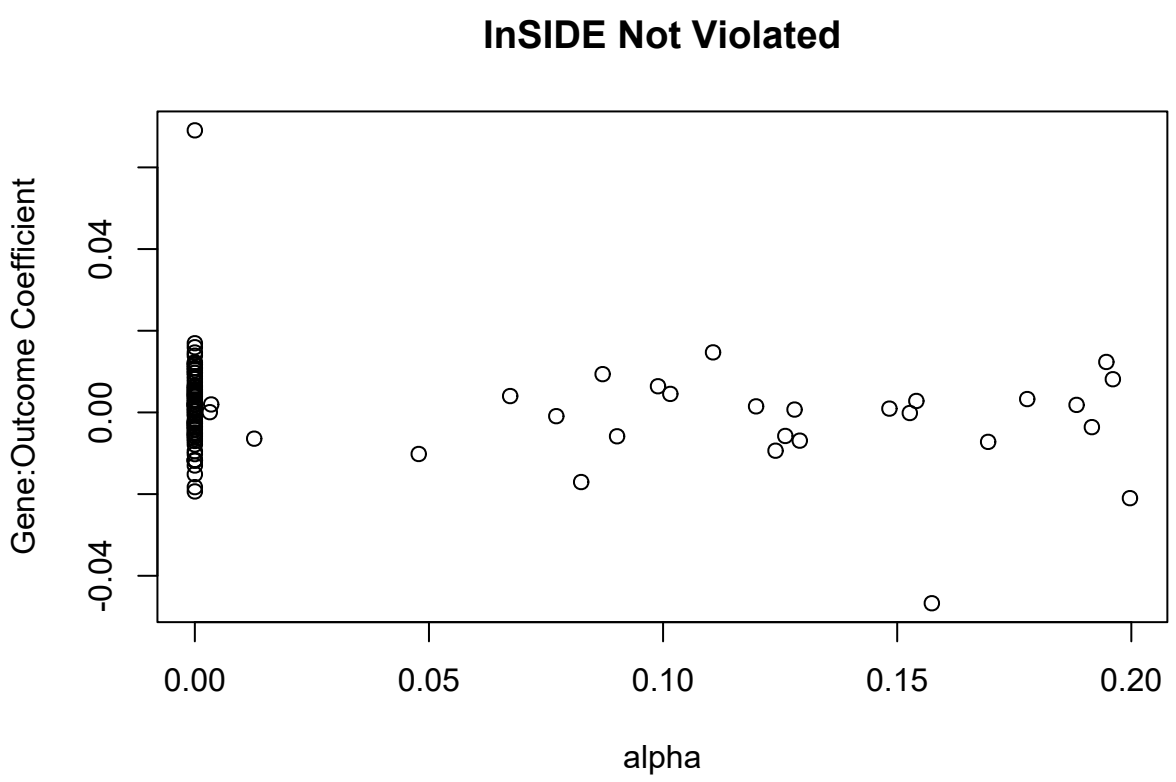
```
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
## number of items to replace is not a multiple of replacement length
## Warning in alpha_vect[j] <- runif(n = n_instruments, min = 0, max = alpha_max):
```

```
## number of items to replace is not a multiple of replacement length
```

```
test_plot_tib_phi_1 <- extract_models_XY(sim_test_data_phi_1)[[1]]
test_plot_tib_phi_0 <- extract_models_XY(sim_test_data_phi_0)[[1]]

test_plot_tib_phi_1 %>%
  select(alpha, coeff_Y_G) %>%
  plot(.,
       main = "InSIDE Violated",
       ylab = "Gene:Outcome Coefficient")
```




```
test_plot_tib_phi_0 %>%
  select(alpha, coeff_Y_G) %>%
  plot(.,
       main = "InSIDE Not Violated",
       ylab = "Gene:Outcome Coefficient")
```


InSIDE Not Violated
Gene:Outcome Coefficient
-0.04
0.00
0.04
0.00
0.05
0.10
0.15
0.20
alpha

```
#phi on y
```