***Replicable R code***

install.packages("pacman")

install.packages("devtools")

pacman::p\_load(future, future.apply, dplyr, reshape2, MASS, qgcomp, gWQS)

# load the NHANES nutrition, outcome and covariates data

nutrients <- c("prot", "carb", "sugar", "fiber", "sfat", "mfat", "pfat", "chol", "alcol", "atoc",

 "vara", "acar", "bcar", "cryp", "lyco", "lz", "vb1", "vb2", "niac", "vb6", "fdfe",

 "chl", "vb12", "b12a", "vc", "vd", "vk", "calc", "phos", "magn", "iron", "zinc", "copp", "sodi", "pota",

 "sele", "caff", "theo")

nutrients <- sort(nutrients)

##############

# Case study #

##############

# for computational time reasons we decreased the number of iterations

# In the paper the nuber of bootstraps and repeated holdout were set to b=100 and rh=100

dwqs\_list\_rh <- gwqsrh(bmi\_cat ~ pwqs + nwqs + pad680 + tot\_act\_cat + riagendr + ridageyr +

 race3 + indfmpir + dmdeduc2 + smoke\_status + cycle,

 mix\_name = nutrients, data = tiwqs\_data, na.action = na.omit, q = 10,

 rh = 3, validation = 0.6, b = 2, family = binomial,

 lambda = 100, seed = 123, plan\_strategy = "multisession",

 solve\_dir\_issue = "inverse")

summary(dwqs\_list\_rh)

dwqs\_list\_rh$final\_weights

#####################

# Simulationn study #

#####################

# Create the simulated data

N <- 500

Nnutrients <- length(nutrients)

mu <- colMeans(tiwqs\_data[,nutrients])

vcovm <- cor(tiwqs\_data[,nutrients], method = "spearman")

w <- rbind(melt(dwqs\_list\_rh$wmat$wmatpos, value.name = "weight", varnames = c("it", "mix\_name")) %>% mutate(direction = "pos"),

 melt(dwqs\_list\_rh$wmat$wmatneg, value.name = "weight", varnames = c("it", "mix\_name")) %>% mutate(direction = "neg"))

dwqsrh\_w <- w %>%

 group\_by(mix\_name, direction) %>%

 summarise(weight = median(weight)) %>%

 mutate(weight = ifelse(weight<=1/38, 0, weight))

dwqsrh\_w$weight[dwqsrh\_w$direction=="pos"] <- dwqsrh\_w$weight[dwqsrh\_w$direction=="pos"]/sum(dwqsrh\_w$weight[dwqsrh\_w$direction=="pos"])

dwqsrh\_w$weight[dwqsrh\_w$direction=="neg"] <- dwqsrh\_w$weight[dwqsrh\_w$direction=="neg"]/sum(dwqsrh\_w$weight[dwqsrh\_w$direction=="neg"])

mpweights <- dwqsrh\_w$weight[dwqsrh\_w$direction=="pos"]

mnweights <- dwqsrh\_w$weight[dwqsrh\_w$direction=="neg"]

set.seed(123)

create\_data <- function(i, vcovm){

 dataset <- scale(mvrnorm(n = N, mu = mu, Sigma = vcovm))

 colnames(dataset) <- paste0("z", 1:Nnutrients)

 dataset <- as.data.frame(dataset)

 tmp <- apply(dataset, MARGIN = 2, FUN = function(i) cut(i, quantile(i, seq(0,1,0.1)), labels=F, include.lowest=T)-1)

 colnames(tmp) <- paste0("z", 1:Nnutrients, "q")

 dataset <- cbind(dataset, tmp)

 dataset$s\_pwqs <- as.numeric(as.matrix(dataset[, paste0("z", 1:Nnutrients, "q")]) %\*% as.vector(mpweights))

 dataset$s\_nwqs <- as.numeric(as.matrix(dataset[, paste0("z", 1:Nnutrients, "q")]) %\*% as.vector(mnweights))

 dataset$y <- rnorm(N, 0.5\*dataset$s\_pwqs - 0.5\*dataset$s\_nwqs, 1)

 dataset$y2 <- rnorm(N, 0.5\*dataset$s\_pwqs, 1)

 dataset$group <- 0

 dataset$group[sample(1:N, round(N\*0.6))] <-1

 return(dataset)

}

# 100 simulated datasets with original correlation matrix

data\_list <- lapply(1:3, create\_data, vcovm = vcovm)

# 100 simulated datasets with halved correlation matrix

vcovm2 <- vcovm\*0.5

diag(vcovm2) <- diag(vcovm)

data\_list0.5 <- lapply(1:3, create\_data, vcovm = vcovm2)

############################### tuning parameter lambda ###############################

set.seed(123)

rh\_list <- lapply(1:3, function(i) sample(0:nrow(data\_list[[1]]), size = round(0.6\*nrow(data\_list[[1]]))))

plan("multisession")

dwqs\_list <- future\_lapply(data\_list, function(i){

 tmp <- lapply(c(0, 1, 10, 100, 1000, 10000), function(j){

 tmp2 <- gwqsrh(y ~ pwqs + nwqs, mix\_name = paste0("z", 1:38), data = i, q = 10,

 validation = NULL, rh = rh\_list, b = 2, family = gaussian, seed = 123,

 valid\_var = "group", lambda = j, signal = "t3")

 tmp3 <- summary(tmp2)

 tmpmat <- cbind(l = j, aic = tmp3$aic, tmp3$coefficients)

 tmpw <- tmp2$final\_weights

 tmpw$l <- j

 out <- list(resmat = tmpmat, w = tmpw)

 return(out)

 })

 resmat <- do.call("rbind", lapply(tmp, function(i) i$resmat))

 w <- do.call("rbind", lapply(tmp, function(i) i$w))

 out2 <- list(resmat = resmat, w = w)

 return(out2)

}, future.seed = FALSE)

# AIC vs lambda

reslist <- lapply(dwqs\_list, function(i) as.data.frame(i$resmat))

reslist <- lapply(1:length(reslist), function(i){

 reslist[[i]]$it <- i

 reslist[[i]]$vars <- rownames(reslist[[i]])

 rownames(reslist[[i]]) <- NULL

 return(reslist[[i]])

})

resall <- as.data.frame(do.call("rbind", reslist))

resall\_bars <- resall %>% filter(grepl("^X.Intercep", vars)) %>% group\_by(l) %>% summarise(mean\_aic = mean(aic), ll\_aic = mean(aic)-sd(aic), ul\_aic = mean(aic)+sd(aic))

resall\_bars

# bias vs lambda

bdata <- resall %>% filter(!grepl("^X.Intercep", vars)) %>%

 mutate(vars = factor(ifelse(grepl("^pwqs", vars), "pwqs", "nwqs"), levels = c("pwqs", "nwqs"),

 labels = c("Positive direction", "Negative direction")),

 beta = ifelse(vars == "Positive direction", Estimate-0.5, Estimate+0.5),

 l = factor(l)) %>%

 group\_by(vars, l) %>%

 summarise(median\_beta = median(beta), Q1\_beta = quantile(beta, probs = 0.25), Q3\_beta = quantile(beta, probs = 0.75))

bdata

# Sensitivity and specificity in identifying the true weights by different lambda

wlist <- lapply(dwqs\_list, function(i) as.data.frame(i$w))

wlist <- lapply(1:length(wlist), function(i){

 wlist[[i]] <- melt(wlist[[i]] %>% dplyr::select(mix\_name, l, `Estimate pos`, `Estimate neg`), id.vars = c("mix\_name", "l"), value.name = "mean\_weight", variable.name = "vars") %>%

 mutate(vars = ifelse(vars == "Estimate pos", "pwqs", "nwqs"),

 it = i)

 return(wlist[[i]])

})

wdata <- as.data.frame(do.call("rbind", wlist))

wdata\_se\_pos <- wdata %>%

 filter(vars == "pwqs") %>%

 mutate(correct = mix\_name %in% paste0("z", which(mpweights>0)) & mean\_weight > 1/38) %>%

 group\_by(l, it) %>%

 summarise(se = sum(correct)/sum(mpweights>0)) %>%

 ungroup() %>%

 group\_by(l) %>%

 summarise(se\_avg = mean(se))

wdata\_sp\_pos <- wdata %>%

 filter(vars == "pwqs") %>%

 mutate(correct = !(mix\_name %in% paste0("z", which(mpweights>0))) & mean\_weight < 1/38) %>%

 group\_by(l, it) %>%

 summarise(sp = sum(correct)/sum(mpweights==0)) %>%

 ungroup() %>%

 group\_by(l) %>%

 summarise(sp\_avg = mean(sp))

wdata\_se\_sp\_pos <- left\_join(wdata\_se\_pos, wdata\_sp\_pos)

wdata\_se\_sp\_pos

wdata\_se\_neg <- wdata %>%

 filter(vars == "nwqs") %>%

 mutate(correct = mix\_name %in% paste0("z", which(mnweights>0)) & mean\_weight > 1/38) %>%

 group\_by(l, it) %>%

 summarise(se = sum(correct)/sum(mnweights>0)) %>%

 ungroup() %>%

 group\_by(l) %>%

 summarise(se\_avg = mean(se))

wdata\_sp\_neg <- wdata %>%

 filter(vars == "nwqs") %>%

 mutate(correct = !(mix\_name %in% paste0("z", which(mnweights>0))) & mean\_weight < 1/38) %>%

 group\_by(l, it) %>%

 summarise(sp = sum(correct)/sum(mnweights==0)) %>%

 ungroup() %>%

 group\_by(l) %>%

 summarise(sp\_avg = mean(sp))

wdata\_se\_sp\_neg <- left\_join(wdata\_se\_neg, wdata\_sp\_neg)

wdata\_se\_sp\_neg

############################### Original correlation matrix ###############################

plan("multisession")

dwqs\_list <- future\_lapply(data\_list, function(i){

 tmp1p <- gwqsrh(y ~ wqs, mix\_name = paste0("z", 1:38), data = i, q = 10, validation = NULL,

 b = 3, rh = rh\_list, b1\_pos = T, b1\_constr = T, family = gaussian, seed = 123,

 valid\_var = "group")

 tmp1n <- gwqsrh(y ~ wqs, mix\_name = paste0("z", 1:38), data = i, q = 10, validation = NULL,

 b = 3, rh = rh\_list, b1\_pos = F, b1\_constr = T, family = gaussian, seed = 123,

 valid\_var = "group")

 tmp1pcoef <- data.frame(wqs = tmp1p$fit$coefficients[2,1], method = "method 1", vars = "pwqs")

 tmp1pw <- tmp1p$final\_weights[,1:2]

 names(tmp1pw) <- c("mix\_name", "mean\_weight")

 tmp1pw$method <- "method 1"

 tmp1pw$vars <- "pwqs"

 tmp1ncoef <- data.frame(wqs = tmp1n$fit$coefficients[2,1], method = "method 1", vars = "nwqs")

 tmp1nw <- tmp1n$final\_weights[,1:2]

 names(tmp1nw) <- c("mix\_name", "mean\_weight")

 tmp1nw$method <- "method 1"

 tmp1nw$vars <- "nwqs"

 i$pwqs <- i$nwqs <- NA

 i$pwqs <- tmp1p$wqs

 i$nwqs <- tmp1n$wqs

 tmp2 <- lm(y ~ pwqs + nwqs, i)

 tmp2coef <- data.frame(wqs = tmp2$coefficients[2:3], method = "method 2",

 vars = c("pwqs", "nwqs"))

 i$pwqs <- i$nwqs <- NULL

 tmp3 <- gwqsrh(y ~ pwqs + nwqs, mix\_name = paste0("z", 1:38), data = i, q = 10,

 validation = NULL, b = 3, rh = rh\_list, family = gaussian, seed = 123,

 valid\_var = "group", lambda = 100, signal = "t3")

 tmp3coef <- data.frame(wqs = tmp3$fit$coefficients[2:3,1], method = "method 4",

 vars = c("pwqs", "nwqs"))

 tmp3w <- tmp3$final\_weights[,c(1,2,5)] %>% melt(variable.name = "vars", value.name = "mean\_weight")

 tmp3w$method <- "method 4"

 tmp3w$vars <- ifelse(tmp3w$vars == "Estimate pos", "pwqs", "nwqs")

 tmp4 <- qgcomp.noboot(y ~ ., data = i[, c(paste0("z", 1:38), "y")], q = 10)

 tmp4coef <- data.frame(wqs = c(tmp4$pos.psi, tmp4$neg.psi), method = "method 3", vars = c("pos.psi", "neg.psi"))

 tmpcoef <- rbind(tmp1pcoef, tmp1ncoef, tmp2coef, tmp4coef, tmp3coef)

 tmpw <- rbind(tmp1pw, tmp1nw, tmp3w)

 rownames(tmpcoef) <- rownames(tmpw) <- NULL

 out <- list(resmat = tmpcoef, w = tmpw)

 return(out)

}, future.seed = FALSE)

resmat <- do.call("rbind", lapply(dwqs\_list, function(i) i$resmat)) %>%

 group\_by(vars, method) %>%

 summarise(median\_bias = median(ifelse(vars %in% c("pwqs", "pos.psi"), wqs-0.5, wqs+0.5)),

 Q1\_bias = quantile(ifelse(vars %in% c("pwqs", "pos.psi"), wqs-0.5, wqs+0.5), probs = 0.25),

 Q3\_bias = quantile(ifelse(vars %in% c("pwqs", "pos.psi"), wqs-0.5, wqs+0.5), probs = 0.75))

resmat

w <- do.call("rbind", lapply(dwqs\_list, function(i) i$w))

wdata\_se\_pos <- w %>%

 filter(vars == "pwqs") %>%

 mutate(correct = mix\_name %in% paste0("z", which(mpweights>0)) & mean\_weight > 1/38,

 it = rep(1:length(data\_list), each = 38\*2)) %>%

 group\_by(method, it) %>%

 summarise(se = sum(correct)/sum(mpweights>0)) %>%

 ungroup() %>%

 group\_by(method) %>%

 summarise(se\_avg = mean(se))

wdata\_sp\_pos <- w %>%

 filter(vars == "pwqs") %>%

 mutate(correct = !(mix\_name %in% paste0("z", which(mpweights>0))) & mean\_weight < 1/38,

 it = rep(1:length(data\_list), each = 38\*2)) %>%

 group\_by(method, it) %>%

 summarise(sp = sum(correct)/sum(mpweights==0)) %>%

 ungroup() %>%

 group\_by(method) %>%

 summarise(sp\_avg = mean(sp))

wdata\_se\_sp\_pos <- left\_join(wdata\_se\_pos, wdata\_sp\_pos)

wdata\_se\_sp\_pos

wdata\_se\_neg <- w %>%

 filter(vars == "nwqs") %>%

 mutate(correct = mix\_name %in% paste0("z", which(mnweights>0)) & mean\_weight > 1/38,

 it = rep(1:length(data\_list), each = 38\*2)) %>%

 group\_by(method, it) %>%

 summarise(se = sum(correct)/sum(mnweights>0)) %>%

 ungroup() %>%

 group\_by(method) %>%

 summarise(se\_avg = mean(se))

wdata\_sp\_neg <- w %>%

 filter(vars == "nwqs") %>%

 mutate(correct = !(mix\_name %in% paste0("z", which(mnweights>0))) & mean\_weight < 1/38,

 it = rep(1:length(data\_list), each = 38\*2)) %>%

 group\_by(method, it) %>%

 summarise(sp = sum(correct)/sum(mnweights==0)) %>%

 ungroup() %>%

 group\_by(method) %>%

 summarise(sp\_avg = mean(sp))

wdata\_se\_sp\_neg <- left\_join(wdata\_se\_neg, wdata\_sp\_neg)

wdata\_se\_sp\_neg

############################### Halving original correlation matrix ###############################

plan("multisession")

dwqs\_list <- future\_lapply(data\_list0.5, function(i){

 tmp1p <- gwqsrh(y ~ wqs, mix\_name = paste0("z", 1:38), data = i, q = 10, validation = NULL,

 b = 3, rh = rh\_list, b1\_pos = T, b1\_constr = T, family = gaussian, seed = 123,

 valid\_var = "group")

 tmp1n <- gwqsrh(y ~ wqs, mix\_name = paste0("z", 1:38), data = i, q = 10, validation = NULL,

 b = 3, rh = rh\_list, b1\_pos = F, b1\_constr = T, family = gaussian, seed = 123,

 valid\_var = "group")

 tmp1pcoef <- data.frame(wqs = tmp1p$fit$coefficients[2,1], method = "method 1", vars = "pwqs")

 tmp1pw <- tmp1p$final\_weights[,1:2]

 names(tmp1pw) <- c("mix\_name", "mean\_weight")

 tmp1pw$method <- "method 1"

 tmp1pw$vars <- "pwqs"

 tmp1ncoef <- data.frame(wqs = tmp1n$fit$coefficients[2,1], method = "method 1", vars = "nwqs")

 tmp1nw <- tmp1n$final\_weights[,1:2]

 names(tmp1nw) <- c("mix\_name", "mean\_weight")

 tmp1nw$method <- "method 1"

 tmp1nw$vars <- "nwqs"

 i$pwqs <- i$nwqs <- NA

 i$pwqs <- tmp1p$wqs

 i$nwqs <- tmp1n$wqs

 tmp2 <- lm(y ~ pwqs + nwqs, i)

 tmp2coef <- data.frame(wqs = tmp2$coefficients[2:3], method = "method 2",

 vars = c("pwqs", "nwqs"))

 i$pwqs <- i$nwqs <- NULL

 tmp3 <- gwqsrh(y ~ pwqs + nwqs, mix\_name = paste0("z", 1:38), data = i, q = 10,

 validation = NULL, b = 3, rh = rh\_list, family = gaussian, seed = 123,

 valid\_var = "group", lambda = 100, signal = "t3")

 tmp3coef <- data.frame(wqs = tmp3$fit$coefficients[2:3,1], method = "method 4",

 vars = c("pwqs", "nwqs"))

 tmp3w <- tmp3$final\_weights[,c(1,2,5)] %>% melt(variable.name = "vars", value.name = "mean\_weight")

 tmp3w$method <- "method 4"

 tmp3w$vars <- ifelse(tmp3w$vars == "Estimate pos", "pwqs", "nwqs")

 tmp4 <- qgcomp.noboot(y ~ ., data = i[, c(paste0("z", 1:38), "y")], q = 10)

 tmp4coef <- data.frame(wqs = c(tmp4$pos.psi, tmp4$neg.psi), method = "method 3", vars = c("pos.psi", "neg.psi"))

 tmpcoef <- rbind(tmp1pcoef, tmp1ncoef, tmp2coef, tmp4coef, tmp3coef)

 tmpw <- rbind(tmp1pw, tmp1nw, tmp3w)

 rownames(tmpcoef) <- rownames(tmpw) <- NULL

 out <- list(resmat = tmpcoef, w = tmpw)

 return(out)

}, future.seed = FALSE)

resmat <- do.call("rbind", lapply(dwqs\_list, function(i) i$resmat)) %>%

 group\_by(vars, method) %>%

 summarise(median\_bias = median(ifelse(vars %in% c("pwqs", "pos.psi"), wqs-0.5, wqs+0.5)),

 Q1\_bias = quantile(ifelse(vars %in% c("pwqs", "pos.psi"), wqs-0.5, wqs+0.5), probs = 0.25),

 Q3\_bias = quantile(ifelse(vars %in% c("pwqs", "pos.psi"), wqs-0.5, wqs+0.5), probs = 0.75))

resmat

w <- do.call("rbind", lapply(dwqs\_list, function(i) i$w))

wdata\_se\_pos <- w %>%

 filter(vars == "pwqs") %>%

 mutate(correct = mix\_name %in% paste0("z", which(mpweights>0)) & mean\_weight > 1/38,

 it = rep(1:length(data\_list), each = 38\*2)) %>%

 group\_by(method, it) %>%

 summarise(se = sum(correct)/sum(mpweights>0)) %>%

 ungroup() %>%

 group\_by(method) %>%

 summarise(se\_avg = mean(se))

wdata\_sp\_pos <- w %>%

 filter(vars == "pwqs") %>%

 mutate(correct = !(mix\_name %in% paste0("z", which(mpweights>0))) & mean\_weight < 1/38,

 it = rep(1:length(data\_list), each = 38\*2)) %>%

 group\_by(method, it) %>%

 summarise(sp = sum(correct)/sum(mpweights==0)) %>%

 ungroup() %>%

 group\_by(method) %>%

 summarise(sp\_avg = mean(sp))

wdata\_se\_sp\_pos <- left\_join(wdata\_se\_pos, wdata\_sp\_pos)

wdata\_se\_sp\_pos

wdata\_se\_neg <- w %>%

 filter(vars == "nwqs") %>%

 mutate(correct = mix\_name %in% paste0("z", which(mnweights>0)) & mean\_weight > 1/38,

 it = rep(1:length(data\_list), each = 38\*2)) %>%

 group\_by(method, it) %>%

 summarise(se = sum(correct)/sum(mnweights>0)) %>%

 ungroup() %>%

 group\_by(method) %>%

 summarise(se\_avg = mean(se))

wdata\_sp\_neg <- w %>%

 filter(vars == "nwqs") %>%

 mutate(correct = !(mix\_name %in% paste0("z", which(mnweights>0))) & mean\_weight < 1/38,

 it = rep(1:length(data\_list), each = 38\*2)) %>%

 group\_by(method, it) %>%

 summarise(sp = sum(correct)/sum(mnweights==0)) %>%

 ungroup() %>%

 group\_by(method) %>%

 summarise(sp\_avg = mean(sp))

wdata\_se\_sp\_neg <- left\_join(wdata\_se\_neg, wdata\_sp\_neg)

wdata\_se\_sp\_neg

############################### Unidirectional association ###############################

plan("multisession")

dwqs\_list <- future\_lapply(data\_list, function(i){

 tmp1p <- gwqsrh(y2 ~ wqs, mix\_name = paste0("z", 1:38), data = i, q = 10, validation = NULL,

 b = 3, rh = rh\_list, b1\_pos = T, b1\_constr = T, family = gaussian, seed = 123,

 valid\_var = "group")

 tmp1n <- gwqsrh(y2 ~ wqs, mix\_name = paste0("z", 1:38), data = i, q = 10, validation = NULL,

 b = 3, rh = rh\_list, b1\_pos = F, b1\_constr = T, family = gaussian, seed = 123,

 valid\_var = "group", solve\_dir\_issue = "average")

 tmp3 <- gwqsrh(y2 ~ pwqs + nwqs, mix\_name = paste0("z", 1:38), data = i, q = 10, validation = NULL,

 b = 3, rh = rh\_list, family = gaussian, seed = 123, valid\_var = "group",

 lambda = 1000)

 tmp31d <- gwqsrh(y2 ~ wqs, mix\_name = paste0("z", 1:38), data = i, q = 10, validation = NULL,

 b = 3, b1\_pos = T, family = gaussian, seed = 123, valid\_var = "group",

 lambda = 1000, rh = rh\_list)

 tmp1pcoef <- data.frame(wqs = tmp1p$fit$coefficients[2,1], method = "method 1", vars = "pwqs")

 tmp1pw <- tmp1p$final\_weights[,1:2]

 tmp1pw$method <- tmp1pw$method <- "method 1"

 tmp1pw$vars <- "pwqs"

 tmp1ncoef <- data.frame(wqs = tmp1n$fit$coefficients[2,1], method = "method 1", vars = "nwqs")

 tmp1nw <- tmp1n$final\_weights[,1:2]

 tmp1nw$method <- tmp1nw$method <- "method 1"

 tmp1nw$vars <- "nwqs"

 i$pwqs <- i$nwqs <- NA

 i$pwqs <- tmp1p$wqs

 i$nwqs <- tmp1n$wqs

 tmp2 <- lm(y2 ~ pwqs + nwqs, i)

 tmp2coef <- data.frame(wqs = tmp2$coefficients[2:3], method = "method 2", vars = c("pwqs", "nwqs"))

 tmp3coef <- data.frame(wqs = tmp3$fit$coefficients[2:3,1], method = "method 4", vars = c("pwqs", "nwqs"))

 tmp3w <- tmp3$final\_weights[,c(1,2,5)] %>% melt(variable.name = "vars", value.name = "Estimate")

 tmp3w$method <- "method 4"

 tmp3w$vars <- ifelse(tmp3w$vars == "Estimate pos", "pwqs", "nwqs")

 tmp31dcoef <- data.frame(wqs = tmp31d$fit$coefficients[2,1], method = "method 4 1d", vars = c("pwqs"))

 tmp31dw <- tmp31d$final\_weights[,c(1,2)]

 tmp31dw$method <- "method 4 1d"

 tmp31dw$vars <- "pwqs"

 tmp4 <- qgcomp.noboot(y2 ~ ., data = i[, c(paste0("z", 1:38), "y2")], q = 10)

 tmp4coef <- data.frame(wqs = c(tmp4$pos.psi, tmp4$neg.psi), method = "method 3", vars = c("pwqs", "nwqs"))

 tmpcoef <- rbind(tmp1pcoef, tmp1ncoef, tmp2coef, tmp3coef, tmp31dcoef, tmp4coef)

 tmpw <- rbind(tmp1pw, tmp1nw, tmp3w, tmp31dw)

 rownames(tmpcoef) <- rownames(tmpw) <- NULL

 out <- list(resmat = tmpcoef, w = tmpw)

 return(out)

}, future.seed = FALSE)

resmat <- do.call("rbind", lapply(dwqs\_list, function(i) i$resmat))

w <- do.call("rbind", lapply(dwqs\_list, function(i) i$w))

resmat <- do.call("rbind", lapply(dwqs\_list, function(i) i$resmat)) %>%

 group\_by(vars, method) %>%

 summarise(median\_bias = median(if\_else(vars == "pwqs", wqs-0.5, wqs)),

 Q1\_bias = quantile(if\_else(vars == "pwqs", wqs-0.5, wqs), probs = 0.25),

 Q3\_bias = quantile(if\_else(vars == "pwqs", wqs-0.5, wqs), probs = 0.75))

resmat

w <- do.call("rbind", lapply(dwqs\_list, function(i) i$w))

wdata\_se\_pos <- w %>%

 filter(vars == "pwqs") %>%

 mutate(correct = mix\_name %in% paste0("z", which(mpweights>0)) & Estimate > 1/38,

 it = rep(1:length(data\_list), each = 38\*3)) %>%

 group\_by(method, it) %>%

 summarise(se = sum(correct)/sum(mpweights>0)) %>%

 ungroup() %>%

 group\_by(method) %>%

 summarise(se\_avg = mean(se))

wdata\_sp\_pos <- w %>%

 filter(vars == "pwqs") %>%

 mutate(correct = !(mix\_name %in% paste0("z", which(mpweights>0))) & Estimate < 1/38,

 it = rep(1:length(data\_list), each = 38\*3)) %>%

 group\_by(method, it) %>%

 summarise(sp = sum(correct)/sum(mpweights==0)) %>%

 ungroup() %>%

 group\_by(method) %>%

 summarise(sp\_avg = mean(sp))

wdata\_se\_sp\_pos <- left\_join(wdata\_se\_pos, wdata\_sp\_pos)

wdata\_se\_sp\_pos