Dirichlet GEE behavior

Emily Palmer

6/2/2022

Simulate data functions

```r
source(here::here("R", "helpers.R"))
source(here::here("R", "dirichlet_functions.R"))
source(here::here("R", "gee_functions.R"))


simulate_dirichlet_y <- function(alpha, n, p, seed = 1225){
  set.seed(seed)
  y_i <- list()
  mat <- matrix(alpha, nrow = n, ncol = p, byrow = T)
  for (i in 1:n) {
    alphai <- as.vector(mat[i, ])
    y_i[[i]] <- rdirichlet(1, alphai)
  }
  ys <- unlist(y_i)
  return(ys)
}

simulate_dirichlet_data <- function(beta, n = 100, p = 15, intercept = T){
  if(intercept){
    beta <- as.vector(t(matrix(c( rep(0, p),beta), nrow = p)))
  }
  x <- c(rep(0, .4 * n), rep(1, .6 * n))
  # change to 1 if including intercept
  if(intercept){
    x <- model.matrix( ~ 1 + x)
  } else{
    x <- model.matrix( ~ 0 + x)
  }
  X <- as.list(as.data.frame(t(x)))
  X <- map(X, ~kronecker(diag(p),t(.x))) %>% reduce(rbind)
  # True eta has no intercept included
  eta <- as.vector( X  %*% beta)
  # log(alpha) = X*beta
  # alpha = e^eta
  alpha <- exp(eta)
  # Calculate the "true" variance and correlation
  alpha0 <- rowSums(matrix(alpha, nrow = n, ncol = p, byrow = T))
  ys <- simulate_dirichlet_y(alpha, n, p)

  return(list(ys = ys,
              X = X,
              x = x))
}
```

GEE code

```
gee_loop <- function(beta0, ys, X,x, n, p, n_rep = 20, intercept = T,q,lambda_hess = 0){
  update_list <- list()
  g_list <- list()
  beta_list <- list()
  diff_list <- list()
  phi_list <- list()

  beta <- beta0
  if(intercept){
    beta <- as.vector(t(matrix(c( rep(0, p),beta), nrow = p)))
  }

  n_rep <- n_rep
  for(i in 1:n_rep){
    beta_list[[i]] <- beta
    eta <- as.vector(X %*% beta)
    alpha <- exp(eta)

    # Calculate the "true" variance and correlation
    alpha0 <- rowSums(matrix(alpha, nrow = n, ncol = p, byrow = T))

    var <- get_dirichlet_var(alpha, n, p)
    cor <- get_dirichlet_cor(alpha, n, p)

    A <- Diagonal(n*p)
    diag(A) <- sqrt(1/var)
    R_inv <- get_R_inv(alpha, omega = 0 , rho = 1, D = diag(n), n, p)

    ymean <- alpha / rep(alpha0, each = p)
    partials <- calculate_partials(alpha, alpha0, n, p, x)

    resid <- diag(A %*% Diagonal(x = ys - ymean))

    phi <- 1 / (as.numeric(sum(resid^2) * (1 / (n*p - (p*q - 1)))))
    phi_list[[i]] <- phi
    V_inv <- phi * A %*% R_inv %*% A

    G <- partials %*% V_inv %*% as.matrix(ys - ymean)
    g_list[[i]] <- sum(as.vector(G))
    H <- -partials %*% V_inv %*% t(partials) - diag(rep(lambda_hess, q * p))

    update <- Matrix::solve(H, G)
    update_list[[i]] <- update

    # Calculate new beta value based on
    # beta+ = beta + gamma H^-1 G
    beta_new <- beta -  as.vector(update)
```

```r
    diff_list[[i]] <- sum(abs(beta_new - beta))

    beta <- beta_new
  }
  return(list(update = update_list,
              g = g_list,
              betas = beta_list,
              diff = diff_list,
              phi = phi_list,
              n = n,
              p = p,
              q = q,
              n_rep = n_rep,
              intercept = intercept))
}



plots <- function(results){
  # Plot beta values across iterations. Are they close to true values?
  # plot changes of betas between iterations
  num_beta <- results$p*results$q
  betas <- as.data.frame(results$betas)
  colnames(betas) <- 1:results$n_rep
  if(results$intercept){
    type_vec <- c("int","x")
  }else{
    type_vec <- "x"
  }
  beta_plot <- betas %>%
    mutate(type = rep(c('x'), num_beta),
           id = factor(1:(num_beta)),
           type = factor(rep(type_vec, results$p))) %>%
    pivot_longer(1:results$n_rep, names_to = "iteration") %>%
    ggplot(aes(x = as.numeric(iteration), y = value, group = id, linetype = type)) +
    geom_line() +
    labs(x = "iteration", y = "beta")
  # plot GEE values, stable? Close to 0?
  g_plot <- data.frame(iter = 1:results$n_rep,
             g = unlist(results$g)) %>%
    ggplot(aes(x = iter, y = g)) +
    geom_line()
  # plot differences of summed abs diff
  diff_plot <- data.frame(iter = 1:results$n_rep,
             diff = unlist(results$diff)) %>%
    ggplot(aes(x = iter, y = diff)) +
    geom_line()
  # plot phi, around 1?
  phi_plot <- data.frame(iter = 1:results$n_rep,
             phi = unlist(results$phi)) %>%
    ggplot(aes(x = iter, y = phi)) +
    geom_line()
  return(list(beta_plot = beta_plot,
              diff_plot = diff_plot,
              phi_plot = phi_plot,
              g_plot = g_plot))
}
```

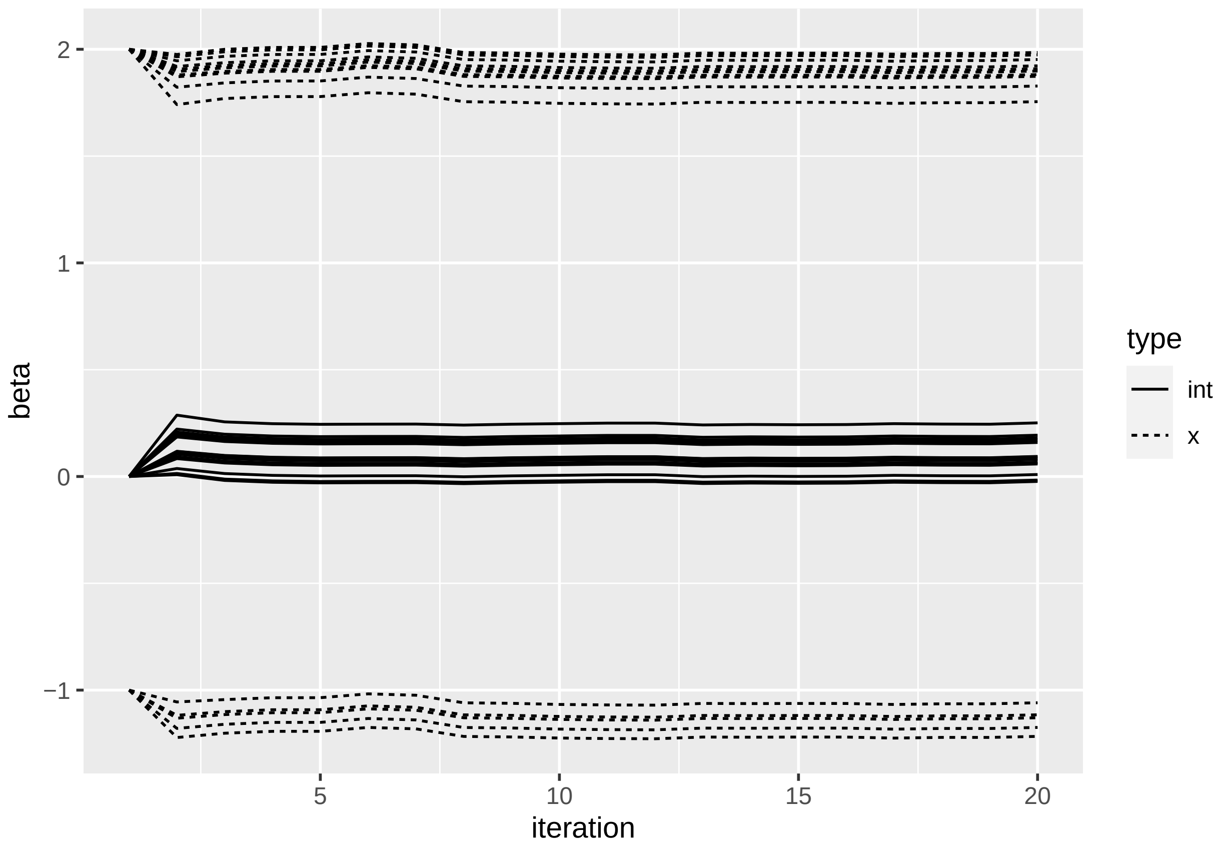
Run on different beta, n, intercept, iter values

n = 500, p = 15, nrep = 20, beta = beta0, beta = -1, 2

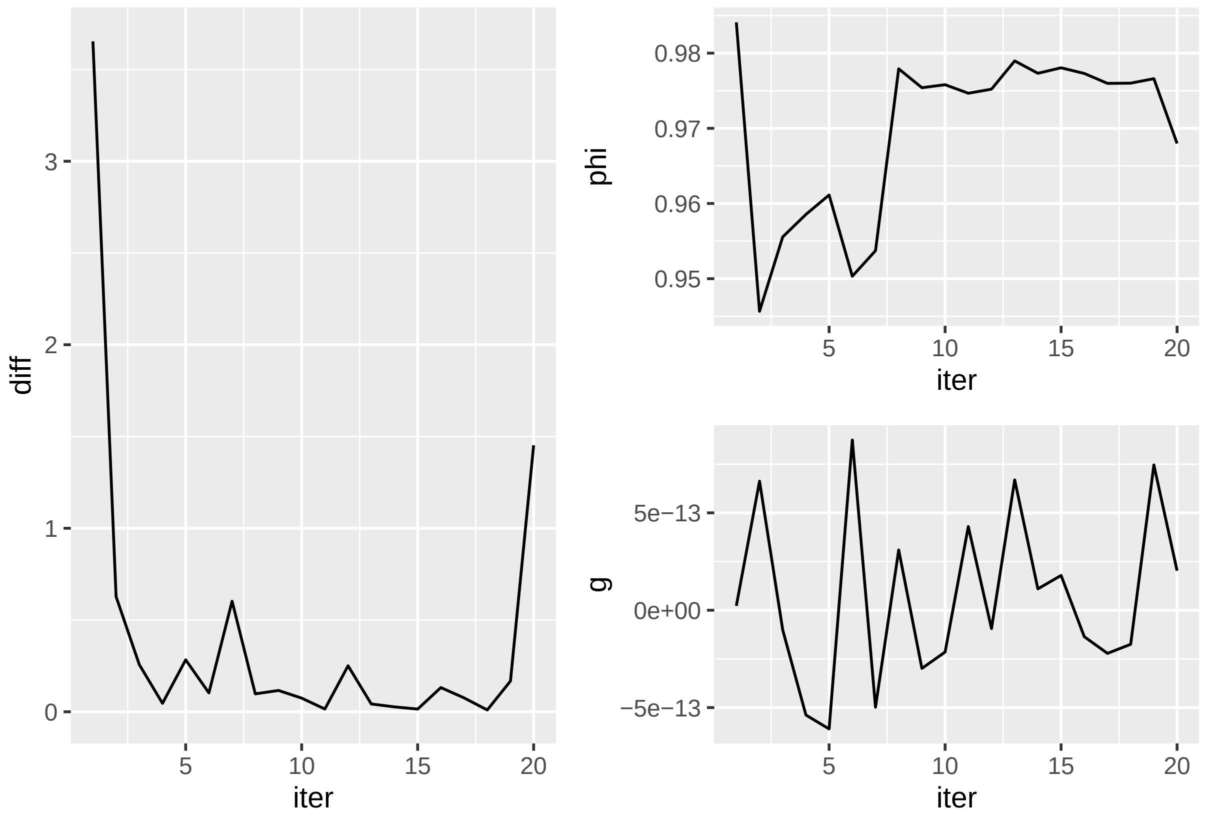
```
n <- 500
p <- 15
dat <- simulate_dirichlet_data(beta = c(rep(-1, 5), rep(2, 10)),
                        n = n,
                        p = p,
                        intercept = T)

results <- gee_loop(beta0 = c(rep(-1, 5), rep(2, 10)),
         ys = dat$ys,
         X = dat$X,
         x = dat$x,
         n = n,
         p = p,
         n_rep = 20,
         intercept = T,
         q = 2)

plota <- plots(results)
plota$beta_plot
```



```
plota$diff_plot + plota$phi_plot / plota$g_plot
```

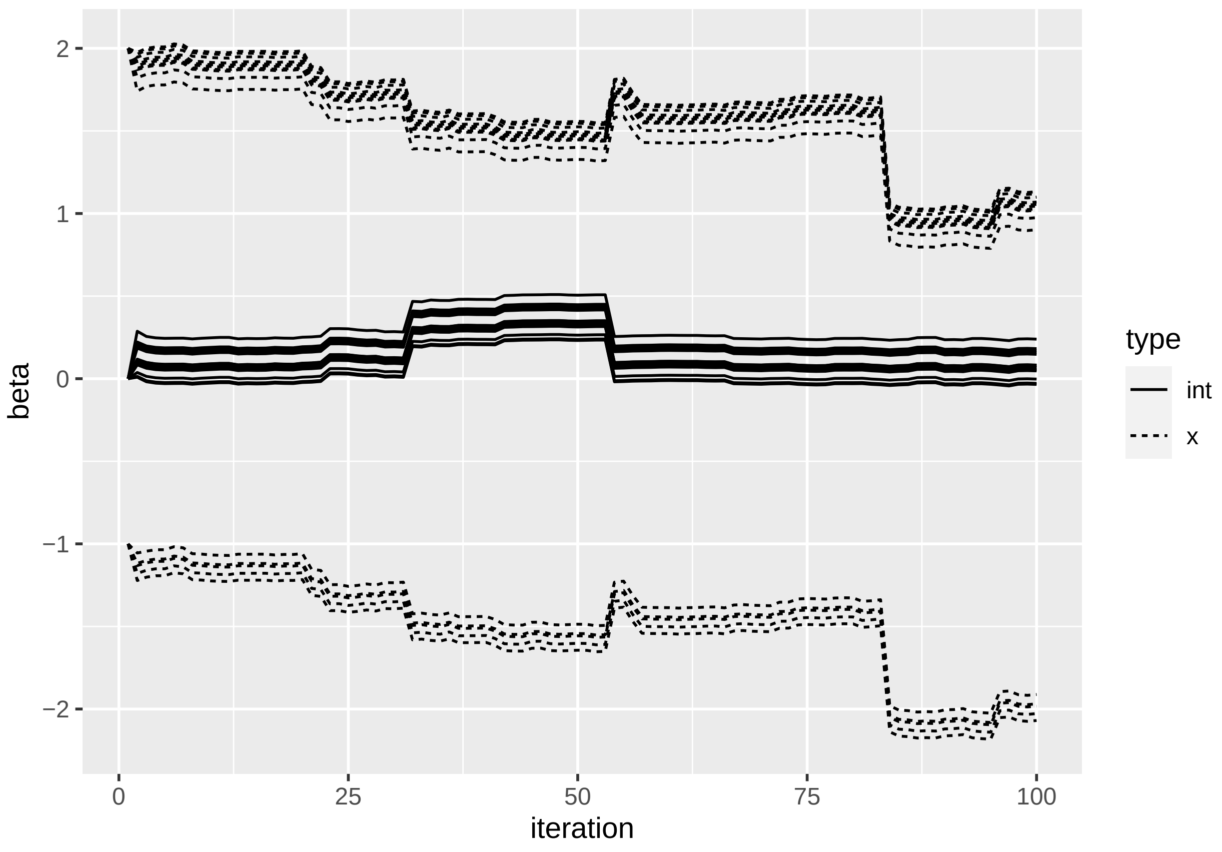


Up the number of iterations

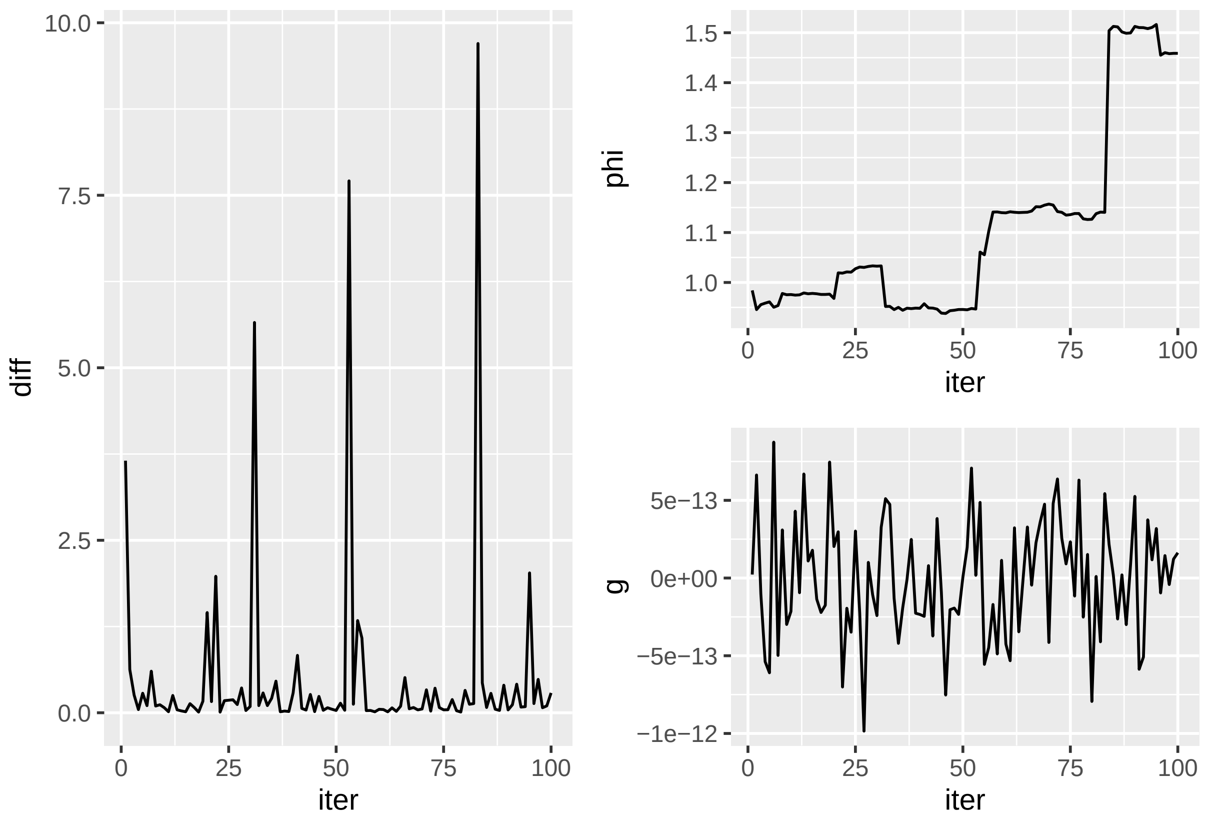
```
n <- 500
p <- 15
dat <- simulate_dirichlet_data(beta = c(rep(-1, 5), rep(2, 10)),
                               n = n,
                               p = p,
                               intercept = T)

results <- gee_loop(beta0 = c(rep(-1, 5), rep(2, 10)),
         ys = dat$ys,
         X = dat$X,
         x = dat$x,
         n = n,
         p = p,
         n_rep = 100,
         intercept = T,
         q = 2)

plota <- plots(results)
plota$beta_plot
```



```
plota$diff_plot + plota$phi_plot / plota$g_plot
```

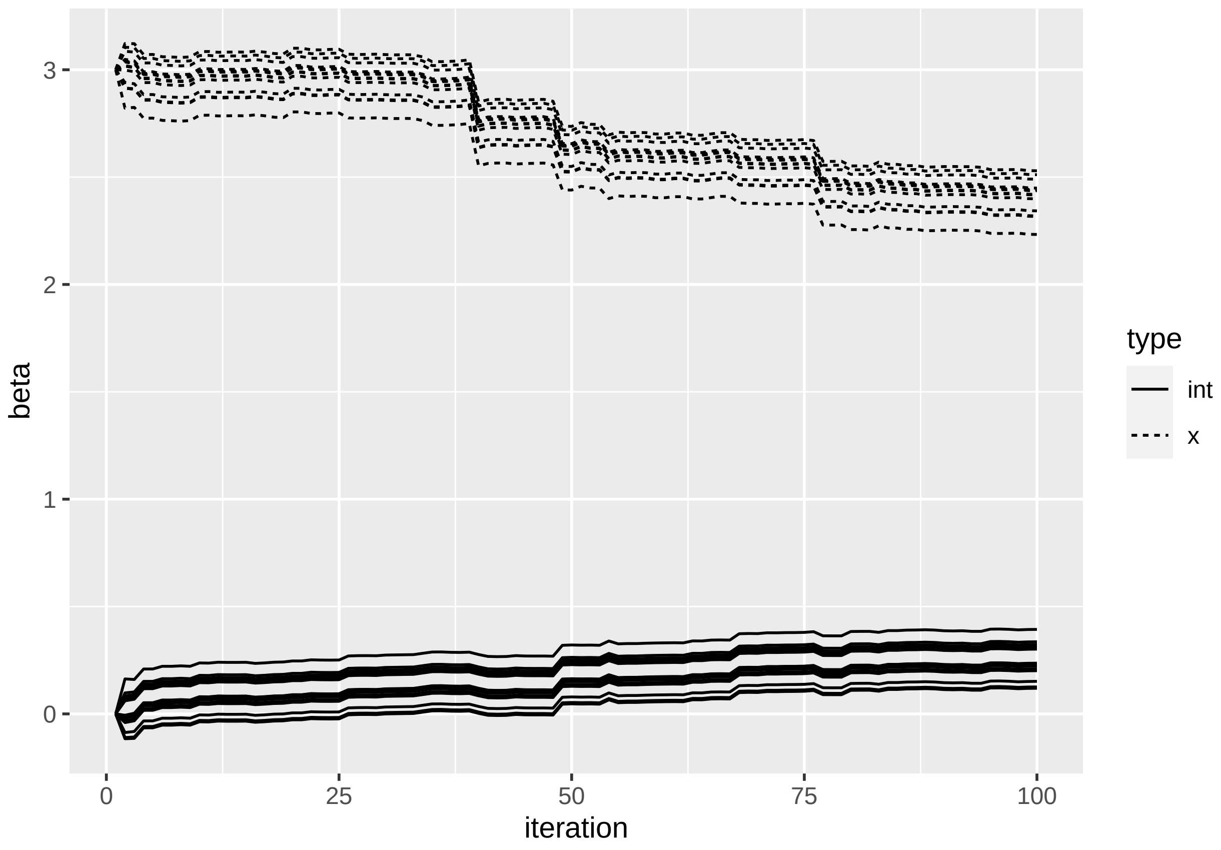


beta all the same, large iter.

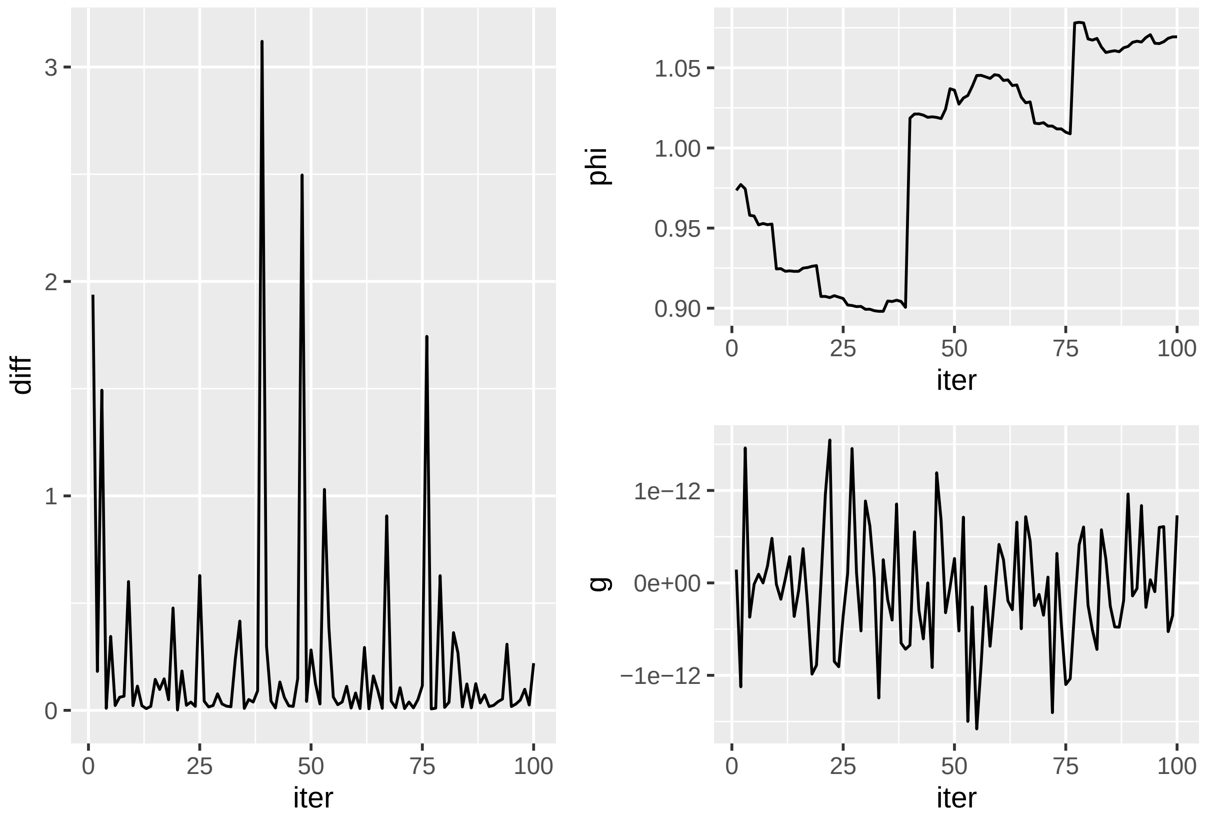
```
n <- 500
p <- 15
dat <- simulate_dirichlet_data(beta = rep(3, 15),
                               n = n,
                               p = p,
                               intercept = T)

results <- gee_loop(beta0 = rep(3,15),
         ys = dat$ys,
         X = dat$X,
         x = dat$x,
         n = n,
         p = p,
         n_rep = 100,
         intercept = T,
         q = 2)

plota <- plots(results)
plota$beta_plot
```



```
plota$diff_plot + plota$phi_plot / plota$g_plot
```

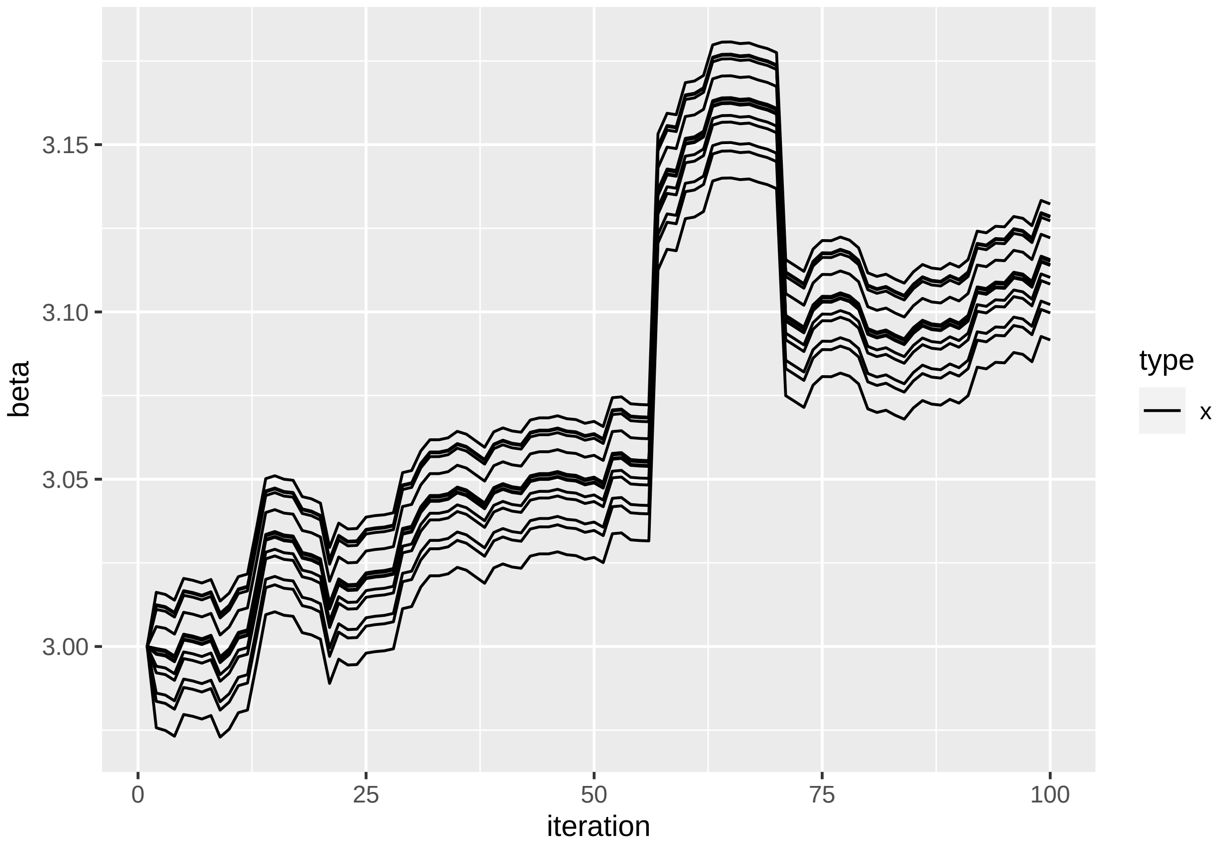


No intercept: beta3:

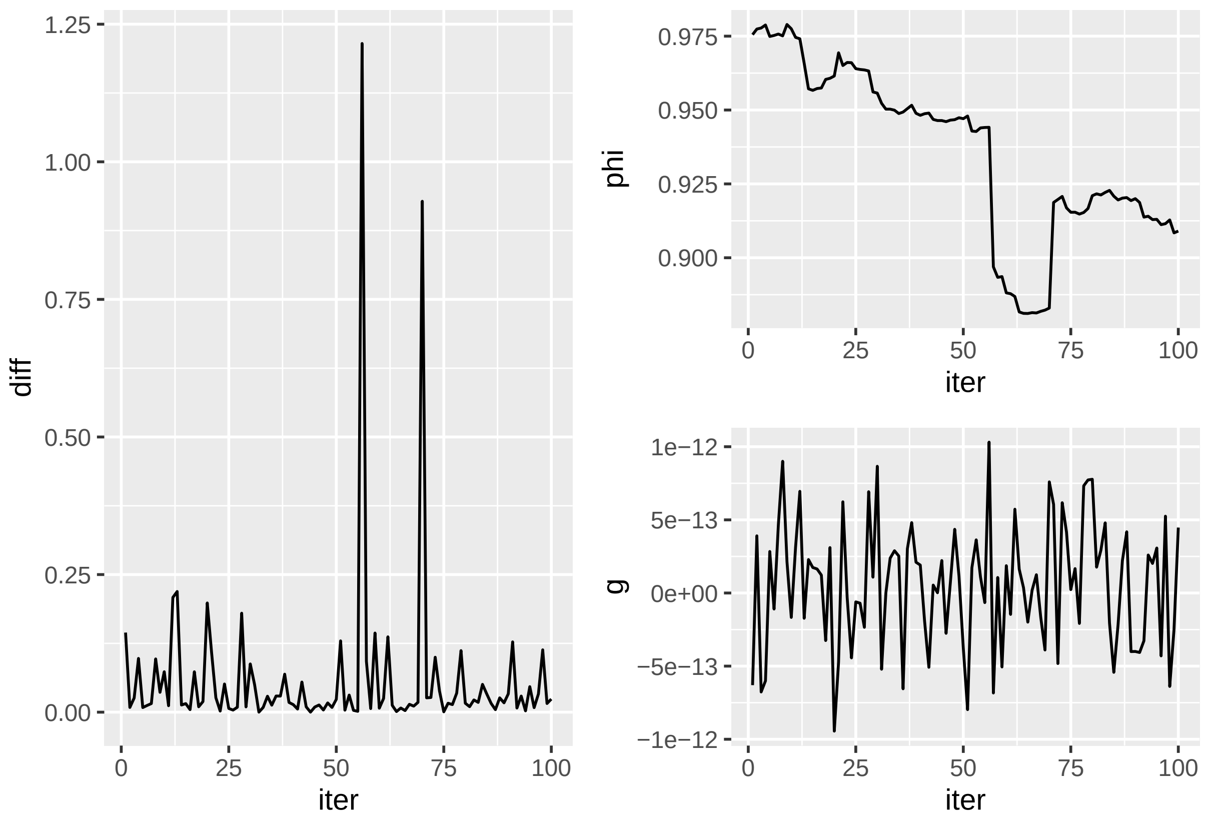
```
n <- 500
p <- 15
dat <- simulate_dirichlet_data(beta = rep(3, 15),
                               n = n,
                               p = p,
                               intercept = F)

results <- gee_loop(beta0 = rep(3,15),
         ys = dat$ys,
         X = dat$X,
         x = dat$x,
         n = n,
         p = p,
         n_rep = 100,
         intercept = F,
         q = 1)

plota <- plots(results)
plota$beta_plot
```



```
plota$diff_plot + plota$phi_plot / plota$g_plot
```

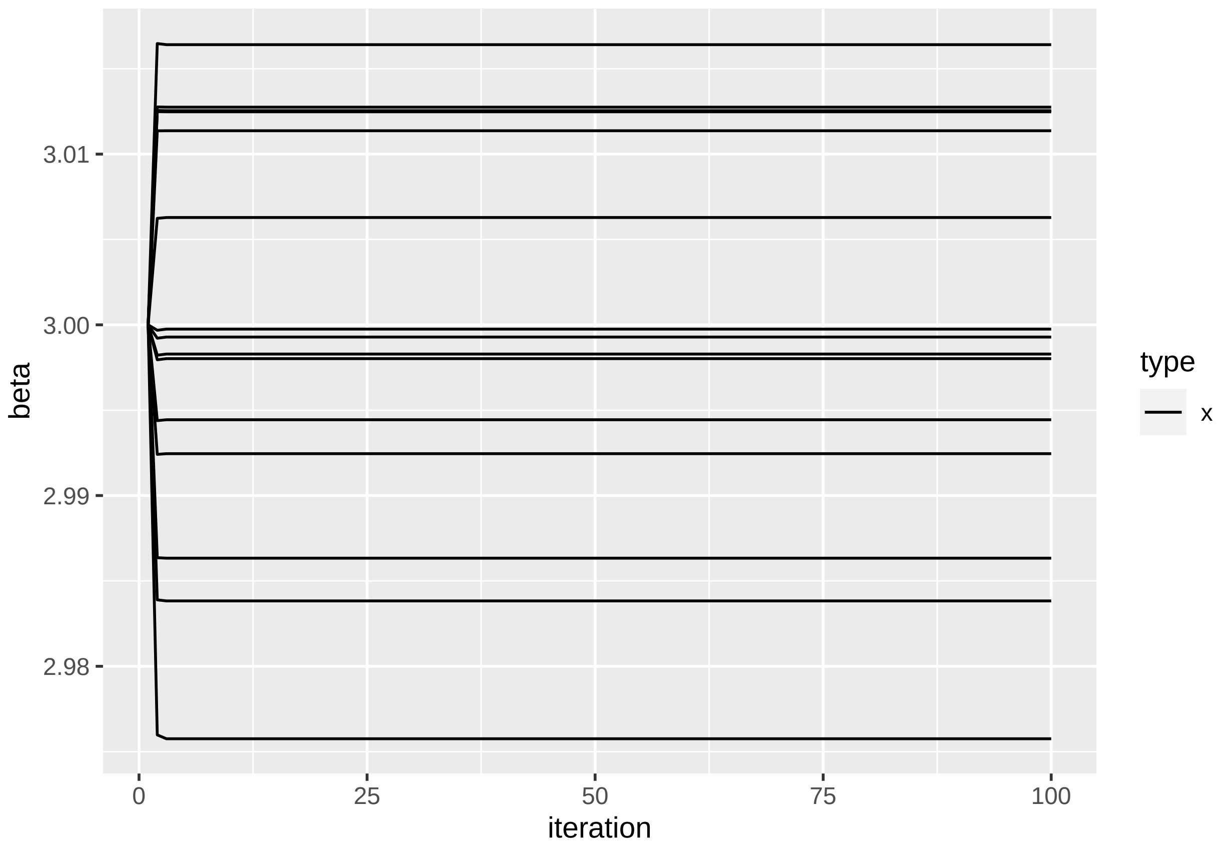


Add lambda term on hessian:

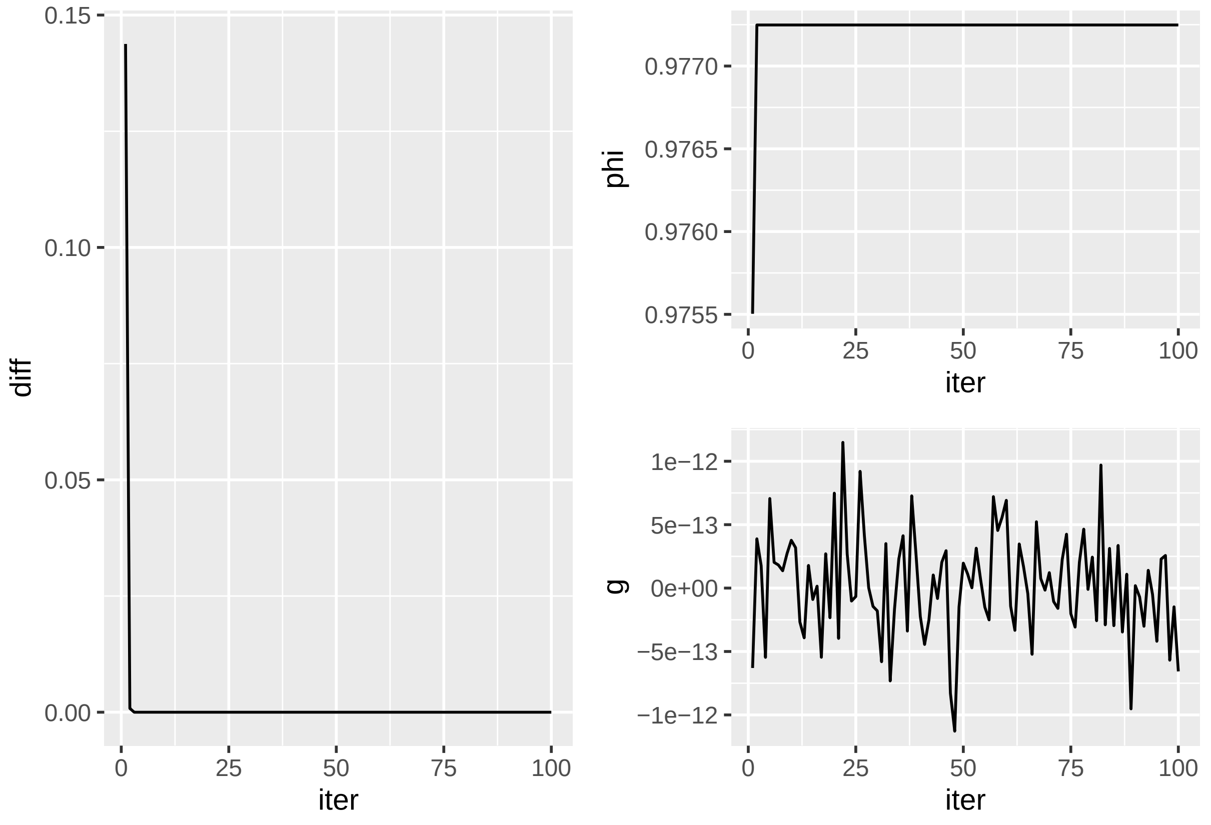
```r
n <- 500
p <- 15
dat <- simulate_dirichlet_data(beta = rep(3, 15),
                               n = n,
                               p = p,
                               intercept = F)

results <- gee_loop(beta0 = rep(3,15),
         ys = dat$ys,
         X = dat$X,
         x = dat$x,
         n = n,
         p = p,
         n_rep = 100,
         intercept = F,
         q = 1,
         lambda_hess = .1)

plota <- plots(results)
plota$beta_plot
```



```
plota$diff_plot + plota$phi_plot / plota$g_plot
```

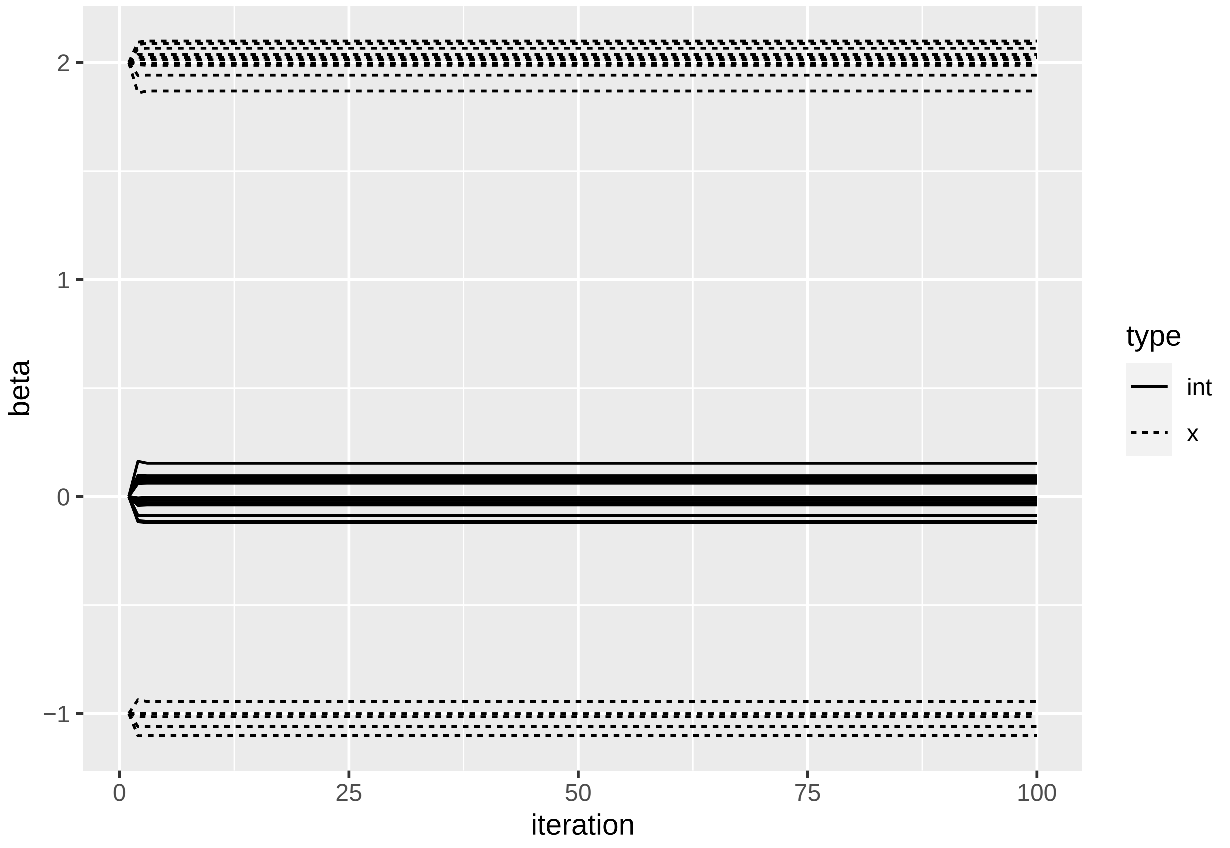


Lambda with intercept and beta-12

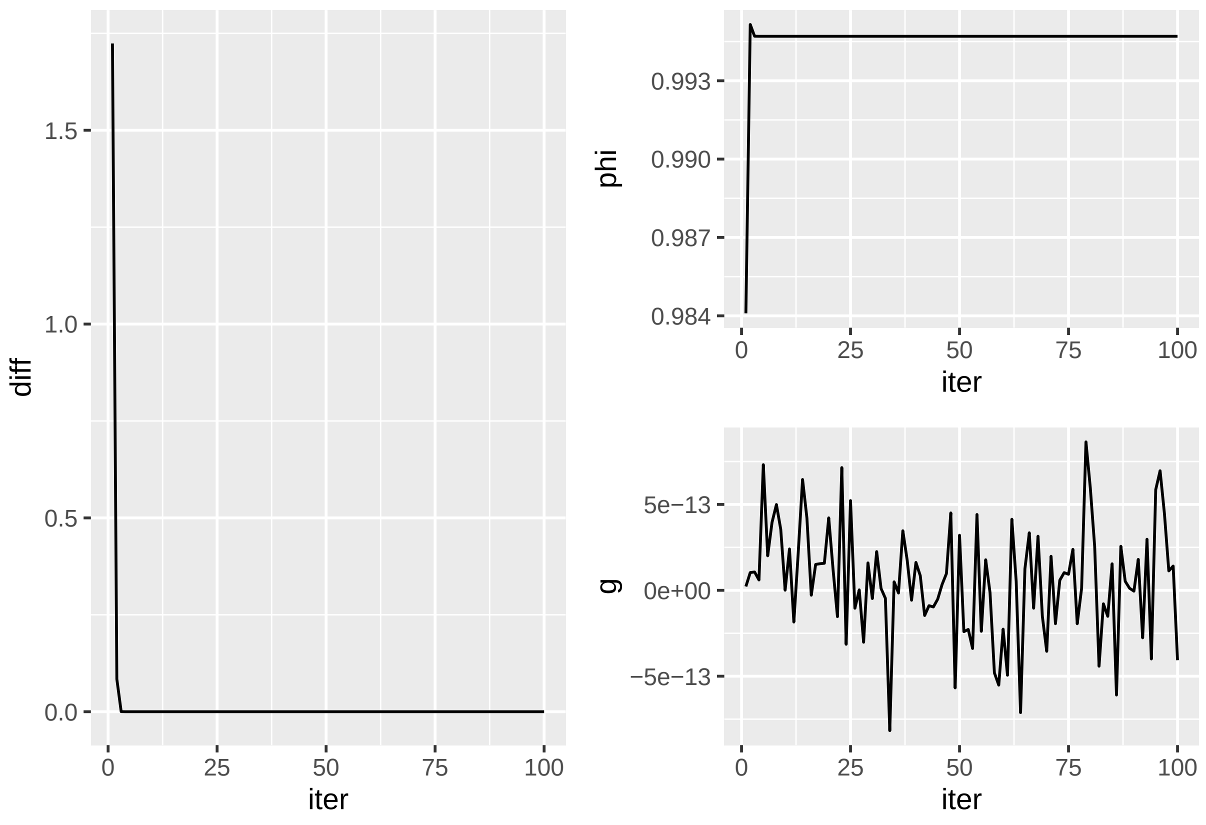
```
n <- 500
p <- 15
dat <- simulate_dirichlet_data(beta = c(rep(-1, 5), rep(2, 10)),
                               n = n,
                               p = p,
                               intercept = T)

results <- gee_loop(beta0 = c(rep(-1, 5), rep(2, 10)),
         ys = dat$ys,
         X = dat$X,
         x = dat$x,
         n = n,
         p = p,
         n_rep = 100,
         intercept = T,
         q = 2,
         lambda_hess = .1)

plota <- plots(results)
plota$beta_plot
```



```
plota$diff_plot + plota$phi_plot / plota$g_plot
```

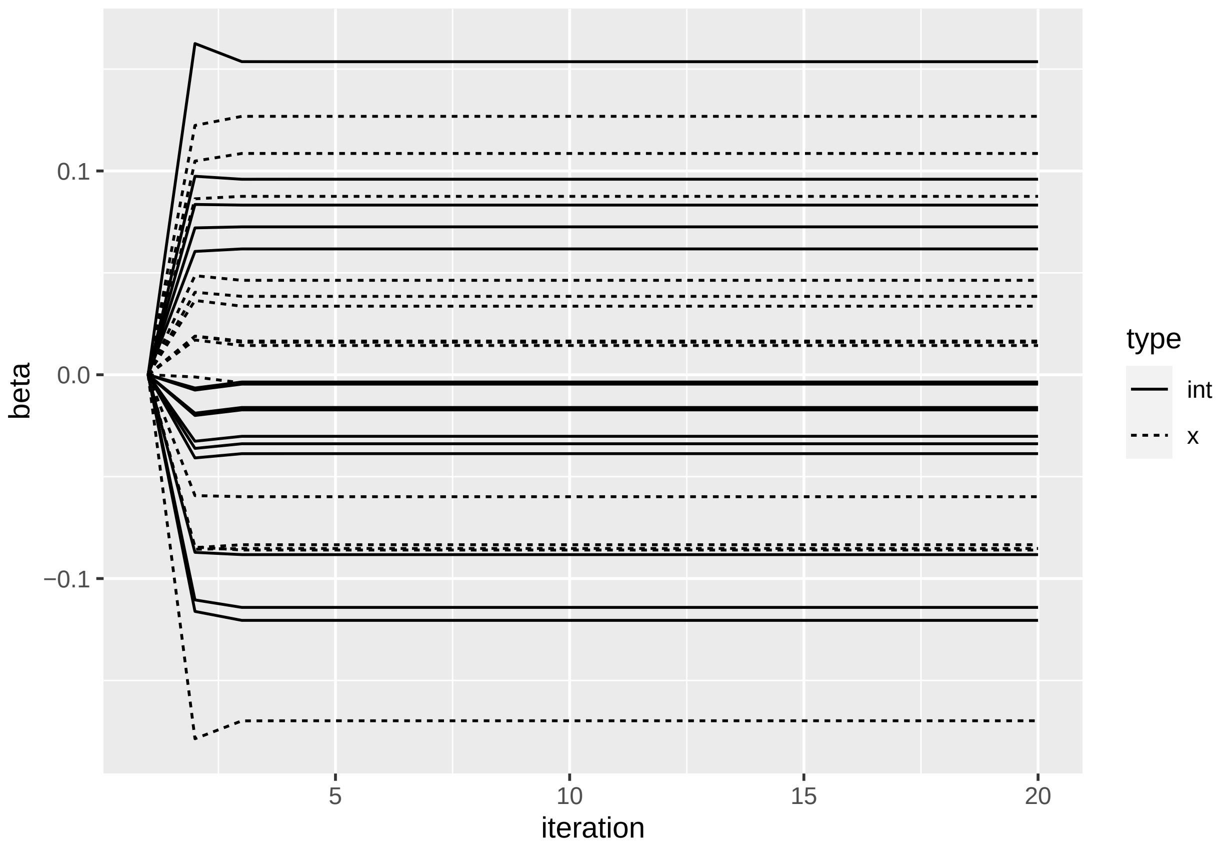


Lambda with intercept and beta3 and 0 beta0

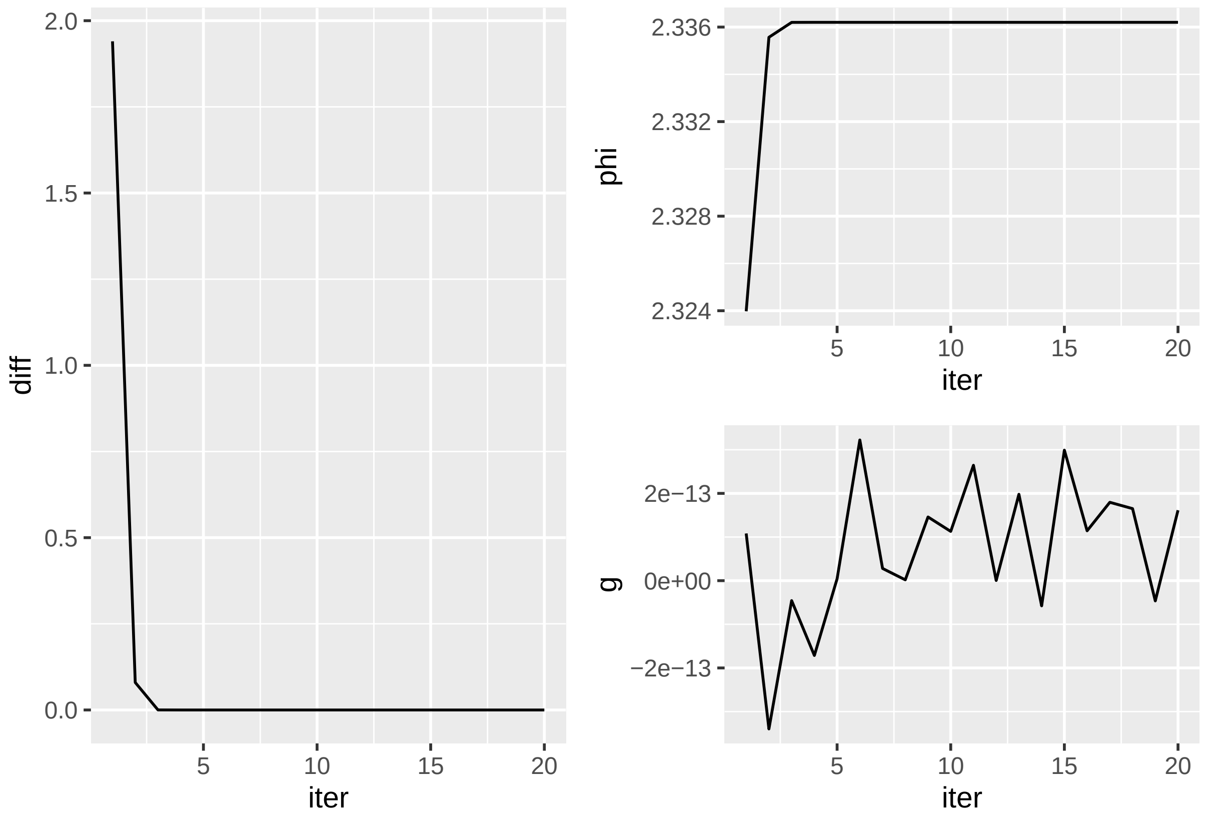
```
n <- 500
p <- 15
dat <- simulate_dirichlet_data(beta = rep(3,15),
                               n = n,
                               p = p,
                               intercept = T)

results <- gee_loop(beta0 = rep(0,15),
         ys = dat$ys,
         X = dat$X,
         x = dat$x,
         n = n,
         p = p,
         n_rep = 20,
         intercept = T,
         q = 2,
         lambda_hess = .1)

plota <- plots(results)
plota$beta_plot
```



```
plota$diff_plot + plota$phi_plot / plota$g_plot
```

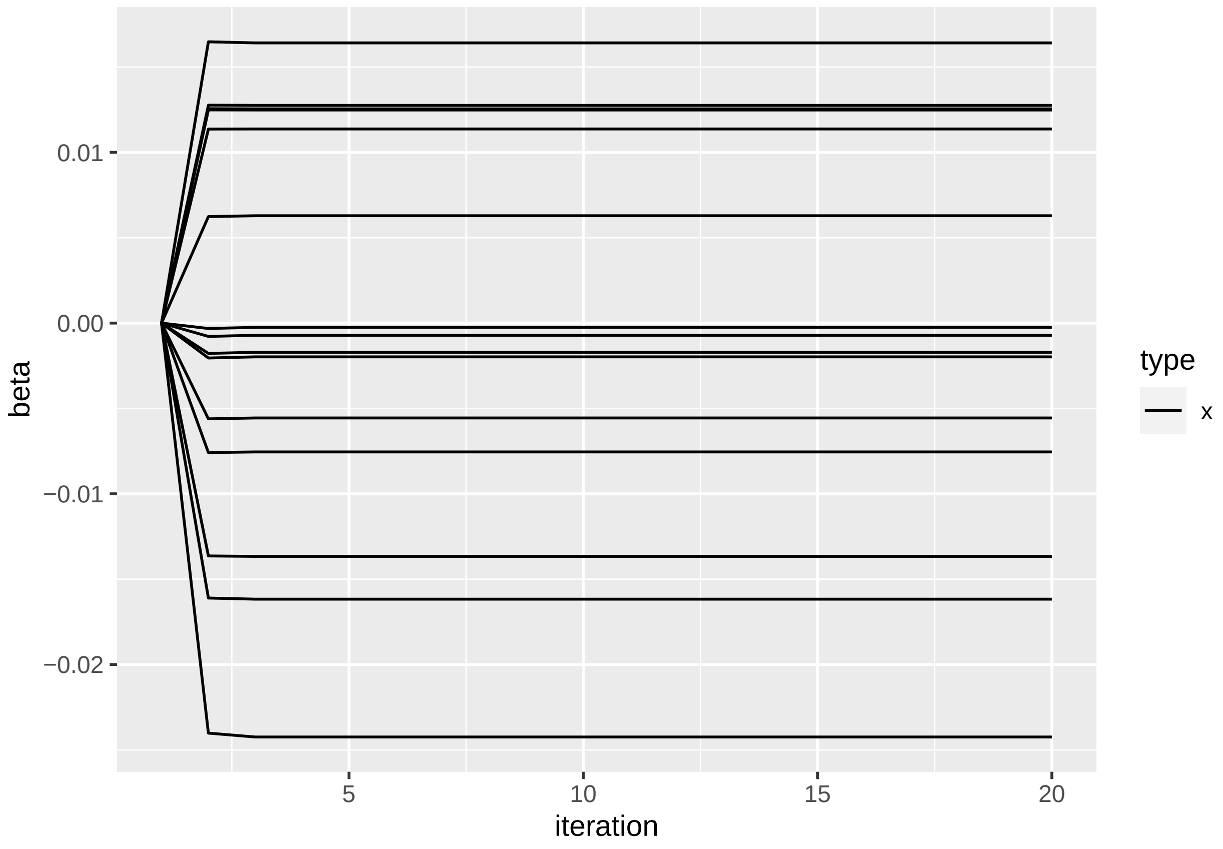


Lambda with intercept and beta3 and 0 beta0

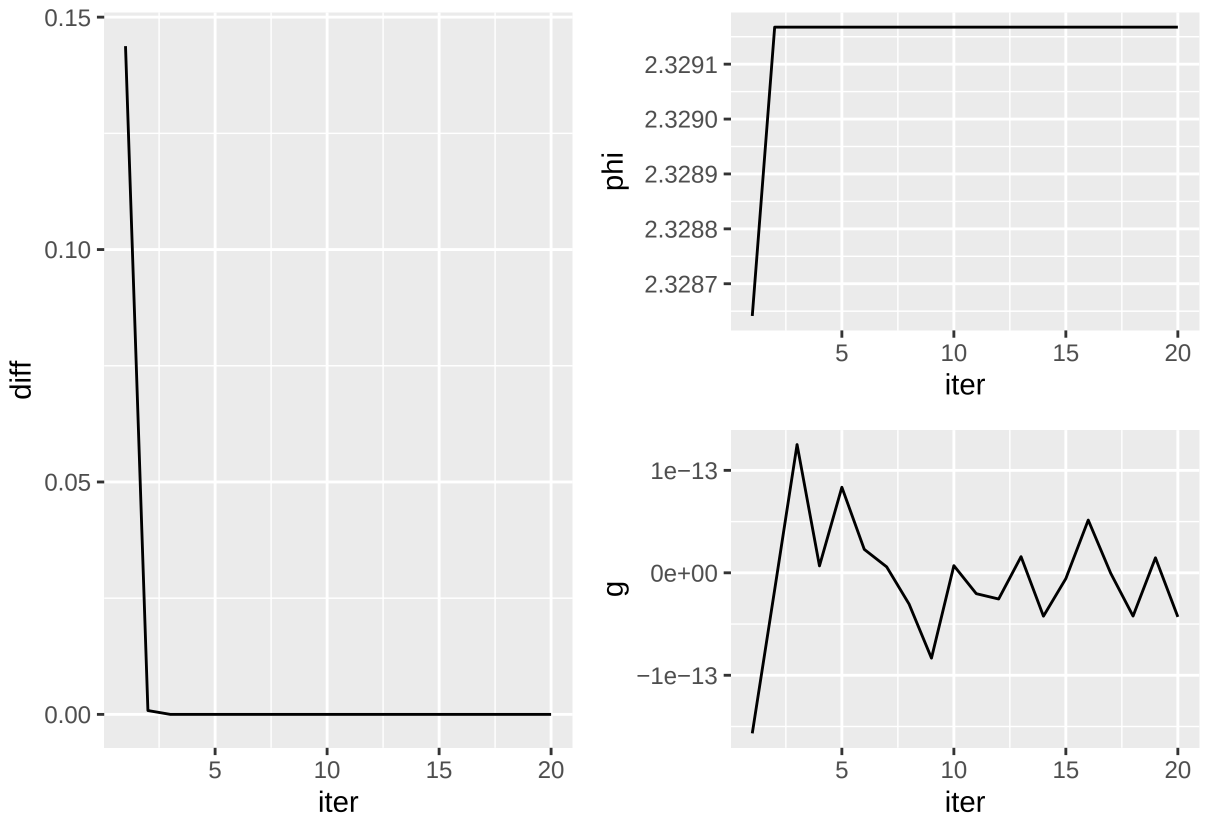
```
n <- 500
p <- 15
dat <- simulate_dirichlet_data(beta = rep(3,15),
                               n = n,
                               p = p,
                               intercept = F)

results <- gee_loop(beta0 = rep(0,15),
         ys = dat$ys,
         X = dat$X,
         x = dat$x,
         n = n,
         p = p,
         n_rep = 20,
         intercept = F,
         q = 1,
         lambda_hess = .1)

plota <- plots(results)
plota$beta_plot
```



```
plota$diff_plot + plota$phi_plot / plota$g_plot
```

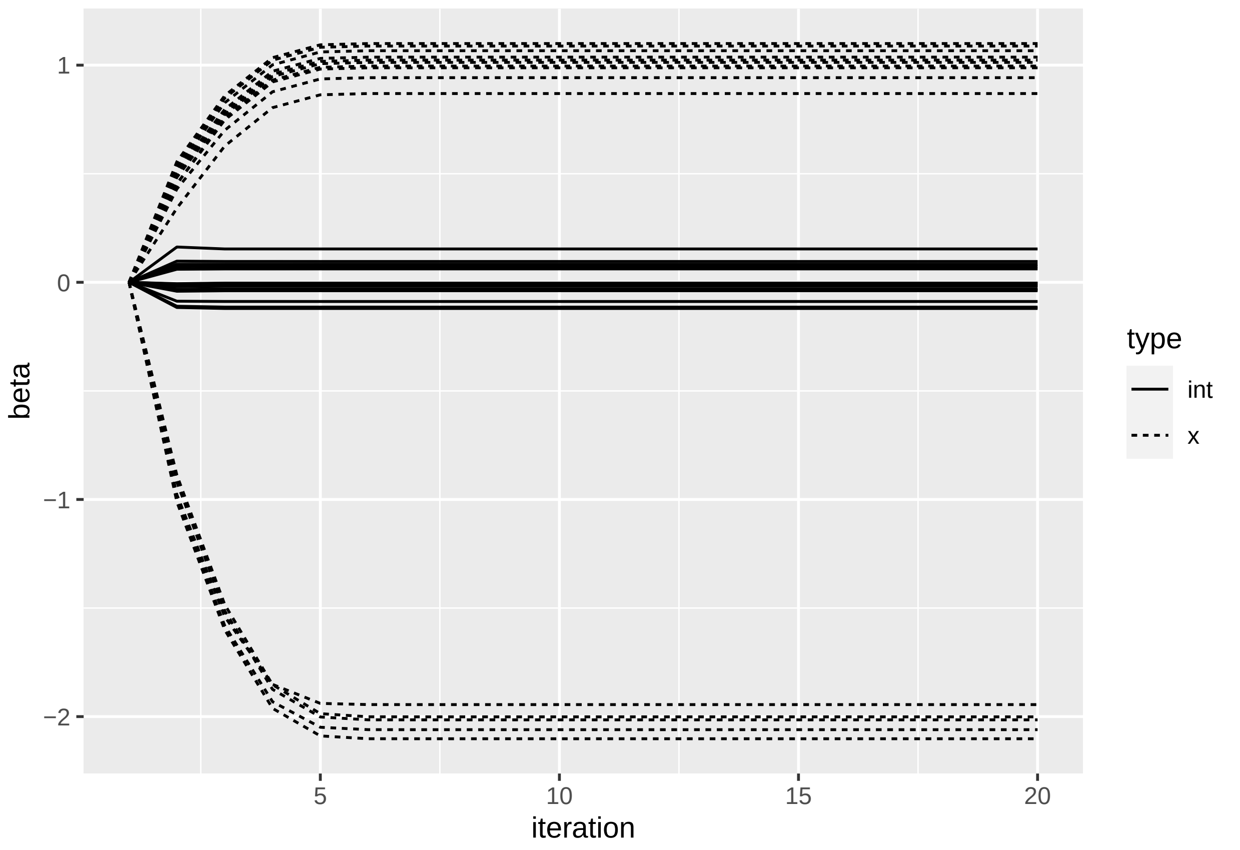


Lambda with intercept and beta-12 and 0 beta0

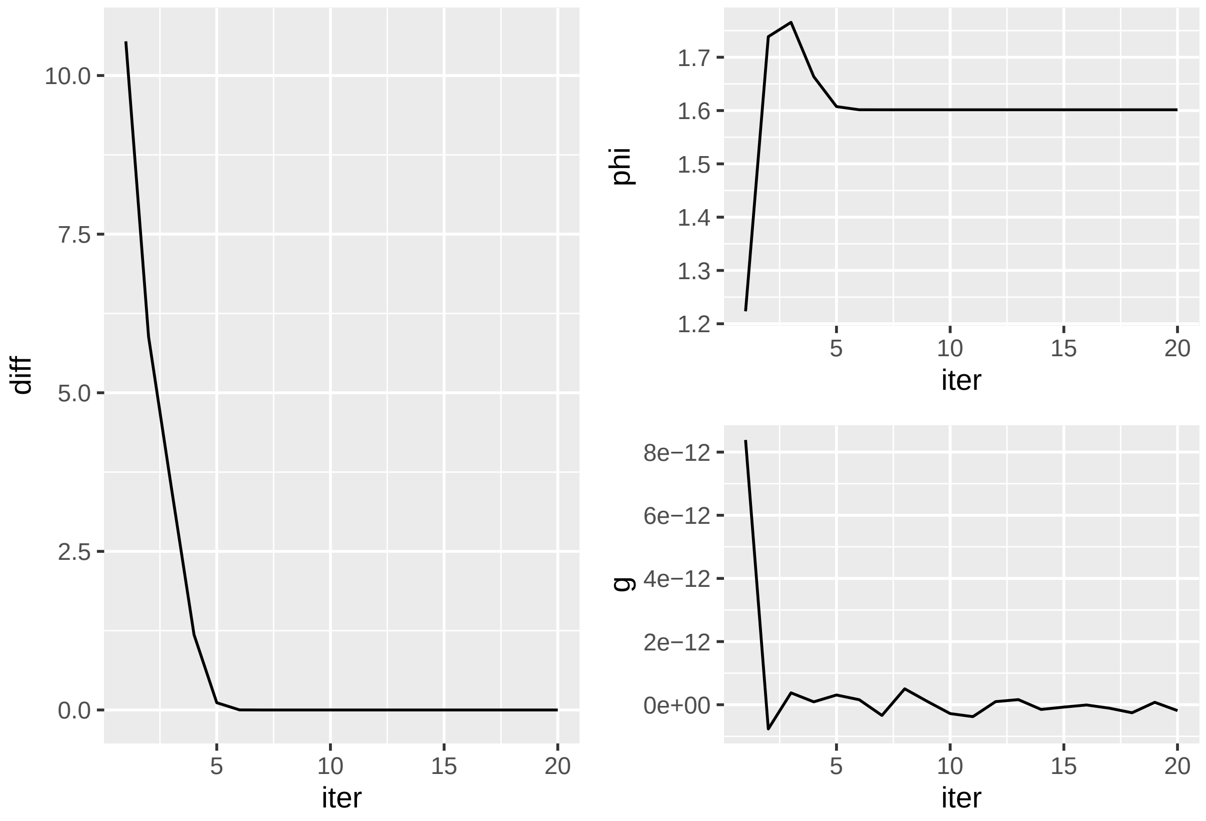
```
n <- 500
p <- 15
dat <- simulate_dirichlet_data(beta = c(rep(-1, 5), rep(2, 10)),
                               n = n,
                               p = p,
                               intercept = T)

results <- gee_loop(beta0 = rep(0,15),
         ys = dat$ys,
         X = dat$X,
         x = dat$x,
         n = n,
         p = p,
         n_rep = 20,
         intercept = T,
         q = 2,
         lambda_hess = .1)

plota <- plots(results)
plota$beta_plot
```



```
plota$diff_plot + plota$phi_plot / plota$g_plot
```


diff
iter
phi
8e−12
6e−12
4e−12
2e−12
0e+00
g