Simulation for Hypothesis Testing

Menghan Hu

This file demonstrates the process of proposed hypothesis testing on simulated data.

The process requires packages 'MASS', 'nlme', 'lme4', and 'lmerTest'.

```
my_packages <- c("MASS", "nlme", "lme4", "lmerTest")
suppressMessages(lapply(my_packages, library, character.only = TRUE))
```

The functional data is simulated from the true model $Y_{gljv}(t) = \mu(t) + \tau_g(t) + U_{gl}(t) + W_{glj}(t) + \epsilon_{gljv}(t)$, where larger b_τ corresponds to a larger difference between the two groups and $b_\tau = 0$ corresponds to no difference.

We will use $b_\sigma = 1$ and $b_\tau = 1$ as example.

```
b_sigma <- 1
b_tau <- 1
```

The simulation will be repeated for 1000 times.

```
nsim = 1000
```

The function defined for calculating covariance matrix is $\Sigma(s,t) = e^{-b_\sigma|s-t|}$.

```
calcSigma <- function(sigma,X1,X2) {
  Sigma <- matrix(rep(0, length(X1)*length(X2)), nrow=length(X1))
  for (i in 1:nrow(Sigma)) {
    for (j in 1:ncol(Sigma)) {
      Sigma[i,j] <- exp(-sigma*abs(X1[i]-X2[j]))
    }
  }
  return(Sigma)
}
```

Because the critical value of F-test depends on the eigenvalues from the generated data, we need to store the critical value in each iteration. We create four empty vectors to store F test statistics, F test critical values, T test statistics, and whether the p-value of the T test is smaller than 0.05.

```
  F_test <- rep(0, nsim)
  F_crit <- rep(0, nsim)
  t_test <- rep(0, nsim)
  t_p <- rep(0, nsim)
```

Let t be on a 100 dimesional equally spaced grid from 0 to 1. We generate data with fixed effect $\mu(t) = 2\sin(2\pi t)$ and $\tau_g(t) = gb_\tau e^{-(t-0.4)}$, where $g = \{1, 2\}$. Suppose there are 500 lesions in each group, 5 layers in each lesion, and 10 voxels within each layer. We use the covariance functions $K_U = 0.8\Sigma$, $K_W = 0.5\Sigma$, and $K_e = 0.2\Sigma$, where $\Sigma(s,t) = e^{-b_\sigma|s-t|}$.

```
 T <- seq(0, 1, by = 0.01)
  G <- 2
  L <- 500
  J <- 5
  V <- 10

  G_ind <- rep(1:G, each = L*J*V)
  L_ind <- rep(1:(G*L), each = J*V)
```

```
  J_ind <- rep(1:(G*L*J), each = V)
  V_ind <- seq(1, G*L*J*V)

  # Generate overall mean function
  mu <- 2*sin(2*pi*T)
  # Generate fixed-effect of group
  tau <- matrix(0, ncol = G, nrow = length(T))
  for (g in 1:G){
    tau[, g] <- (b_tau*g) * exp(-(T - 0.4))
  }
  # Calculate the covariance matrix
  sigma <- calcSigma(b_sigma, T, T)

  K_U <- 0.8*sigma
  K_W <- 0.5*sigma
  K_e <- 0.2*sigma
```

We use a `for` loop to iterate the simulation for `nsim` times.

```
for (i in 1:sim){

}
```

Inside each iteration, we generate the data.

```
U_values <- matrix(mvrnorm(G*L, rep(0, length(T)), K_U), ncol = G*L, byrow = FALSE)
W_values = matrix(mvrnorm(G*L*J, rep(0, length(T)), K_W), 101, ncol = G*L*J, byrow = FALSE)
e_values <- matrix(mvrnorm(G*L*J*V, rep(0, length(T)), K_e), ncol = G*L*J*V)

sim_data <- replicate(G*L*J*V, mu) +
    tau[, rep(1:ncol(tau), each = L*J*V)] +
    U_values[, rep(1:ncol(U_values), each = J*V)] +
    W_values[, rep(1:ncol(W_values), each = V)] +
    e_values
sim_data <- t(sim_data)
```

We apply the proposed hypothesis testing and the mixed effect model method, respectively. The hirerarchical hypothesis testing method is applied as following:

```
y_bar <- apply(sim_data, 2, mean)
  y_g_bar <- apply(sim_data, 2, function(x) {by(x, G_ind, mean)})
  y_gl_bar <- apply(sim_data, 2, function(x) {by(x, L_ind, mean)})

  est <- 0
  for (gg in 1:G) {
    est <- est +
      (t(y_gl_bar[((gg-1) * L + 1):(gg * L), ])- replicate(L, y_g_bar[gg, ])) %*%
      (y_gl_bar[((gg-1) * L + 1):(gg * L), ] - t(replicate(L, y_g_bar[gg, ])))
  }
  eigen_vals <- eigen(est/(2*(L-1)))$values
  ratio <- sum(eigen_vals)^2/sum(eigen_vals^2)

  SS_group <- sum(apply(y_g_bar, 1, function(x) return((x - y_bar)^2*L*J*V)))
  SS_lesion <- sum((y_gl_bar - y_g_bar[rep(1:G, each = L), ])^2*J*V)
```

The test statistic and the critical value of the F-test are recorded in the vectors.

```
  F_test[i] <- (SS_group/(G-1))/(SS_lesion/(G*(L-1)))
  F_crit[i] <- qf(.95, df1 = ratio, df2 = ratio*G*(L-1))
```

The mixed effect model method is applied as following:

```
sim_pca <- prcomp(sim_data, scale. = F)
  pca_data <- data.frame(pc1 = sim_pca$x[,1], group = G_ind, lesion = L_ind, layer = J_ind)
  fit <- summary(lmer(pc1~ factor(group) + (1|lesion) + (1|layer)  , data = pca_data))
```

The test statistics of the T-test and whether the p-value is less than 0.05 are recoded in the vectors.

```
 t_test[i] <- fit$coefficients[2,4]
 t_p[i] <- (fit$coefficients[2,5] < 0.05)
```

After the `nsim` iteration, we output the result in the form of a list.

```
re <- list(btau = b_tau,
           bsigma = b_sigma,
           ftest = F_test,
           fcrit = F_crit,
           ttest = t_test,
           tp = t_p)
```