ddsPLS Exploration

Harpeth Lee

2/4/2022

```
library(ddsPLS2)
library(MASS)
library(spls)
```

```
ddsPLS2::ddsPLS2_App()
```

This code chunk opens an applet that can be used to build models using ddsPLS. Note that it requires the X and Y variables as separate csv files.

Code copied from the simulation_ssdpls2 repository created by Hadrien Lorenzo.

The `get_toy_example()` function simulates a data frame with `n` observations of where 50 of `p` predictors are associated with the single response variable.

```
# Creates a toy data set for the ddsPLS function
toy_ex <- get_toy_example()
```

```
# Creates model from the toy data
toy_mod <- ddsPLS(toy_ex$X, toy_ex$Y)

toy_results <- toy_mod$results
```

Recreate Toy Example

This is a recreation of the toy example created by Hadrien Lorenzo, the original example can be found here.

```
# Creates toy data set to be used
simu_toy <- get_toy_example(n=50,sqrt_1_minus_sig2 = 0.9025,p = 1000)

# Creates vector of lambda values to be used
lambdas <- seq(0,1,length.out = 30)

# Sets number of bootstrap samples to run
n_B <- 100

# Creates model using ddsPLS algorithm
model_toy <- ddsPLS(simu_toy$X,simu_toy$Y,
                     doBoot = FALSE,
                    lambdas = lambdas,
                    n_B = n_B,
                    verbose = T # whether trace during process
                    )
```

```
model_toy_2 <- ddsPLS(simu_toy$X,simu_toy$Y,
                      doBoot = FALSE,
                  criterion = "Q2",
                   lambdas = lambdas,
                   n_B = n_B,
                   verbose = T # whether trace during process
                   )
```

Design 1

Generates n samples of p observations with q response variables. Projects 5 latent variables onto p components.

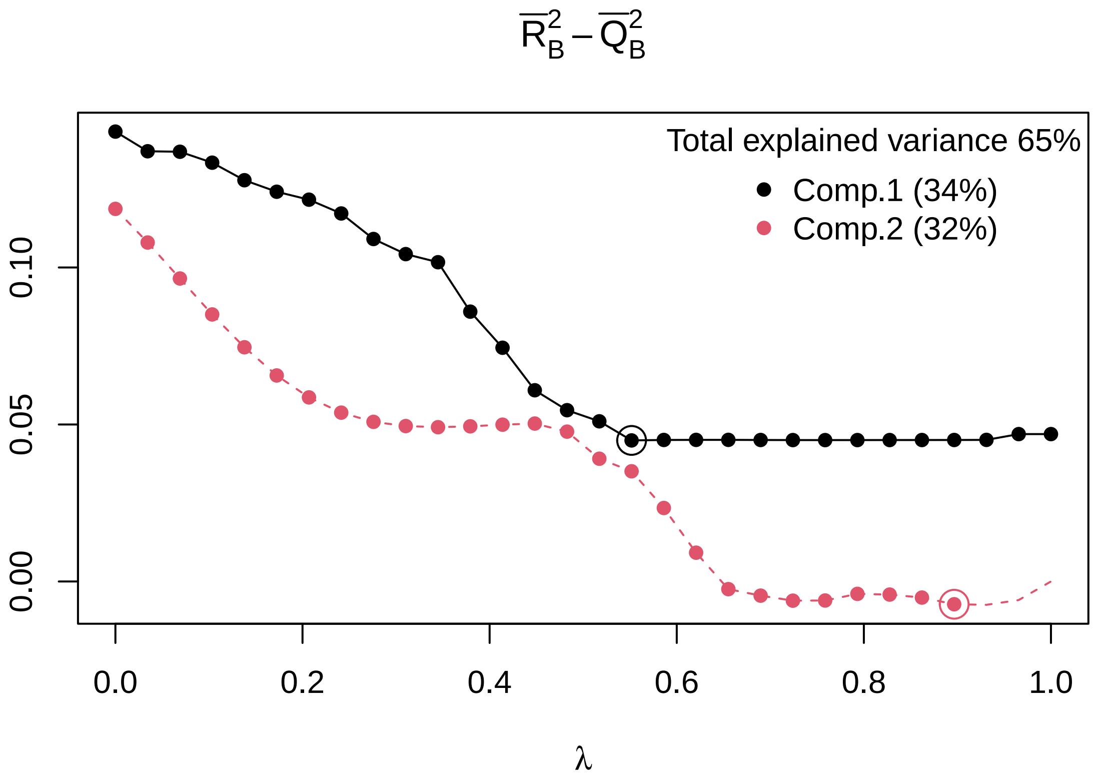
```
simu_1 <- get_design_1(n=50,sqrt_1_minus_sig2 = 0.99,p = 1000,q = 3)
```

What does the NCORES argument do? Setting it to integers greater than 1 gives an error.

Is there a way to include more components in the model?

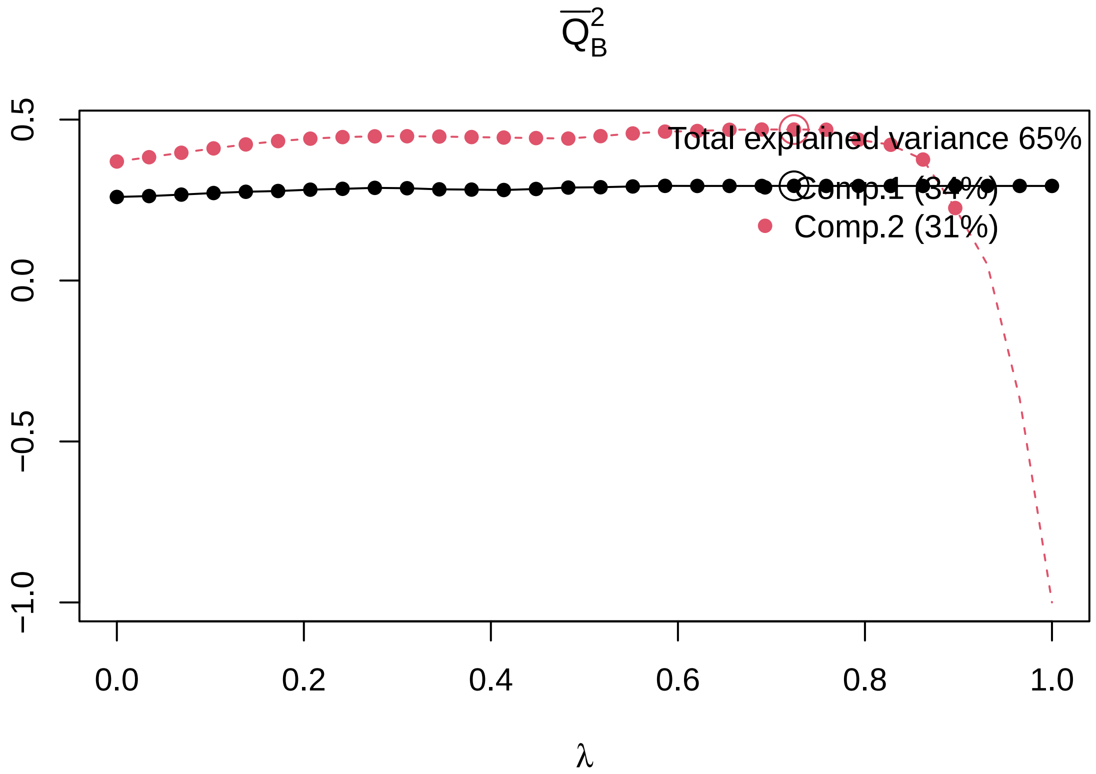
```
model_1 <- ddsPLS(simu_1$X,simu_1$Y,
                  lambdas = lambdas,
                  n_B=n_B,
                  verbose=T)
```

```
##                        --------------
##                       |    ddsPLS    |
## =====================----------------=====================
## Should we build component 1 ? Bootstrap pending...
##      lambda   R2  R2h  Q2 Q2h VarExpl VarExpl.Tot
##        0.55 0.35 0.35 0.3 0.3     34%         34%
##                                      ...component 1 built!
## Should we build component 2 ? Bootstrap pending...
##      lambda  R2  R2h   Q2  Q2h VarExpl VarExpl.Tot
##         0.9 0.4 0.12 0.41 0.27     32%         65%
##                                      ...component 2 built!
## Should we build component 3 ? Bootstrap pending...
##                                  ...component 3 not built!
## =====================                ====================
##                      ================
```



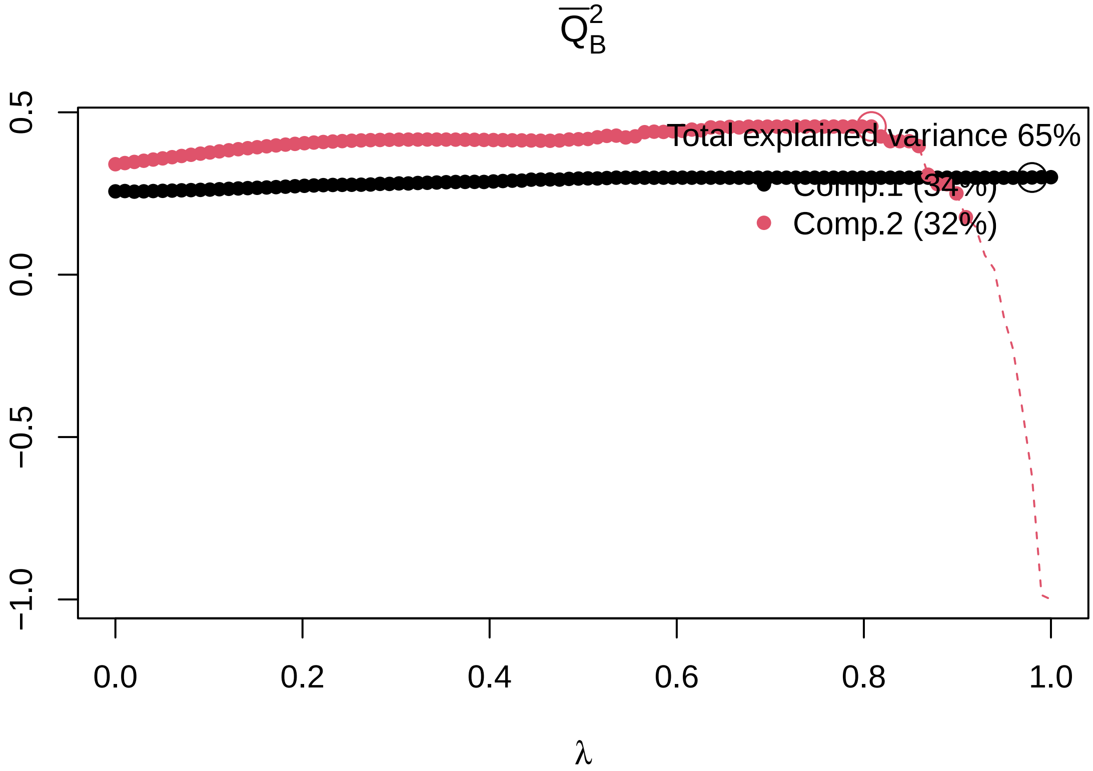
```
model_1_Q2 <- ddsPLS(simu_1$X,simu_1$Y,
                  criterion = "Q2",
                  lambdas = lambdas,
                  n_B=n_B,
                  verbose=T)
```

```
##                       --------------
##                      |    ddsPLS    |
## =====================----------------=====================
## Should we build component 1 ? Bootstrap pending...
##      lambda   R2  R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.72 0.34 0.34 0.29 0.29     34%         34%
##                                      ...component 1 built!
## Should we build component 2 ? Bootstrap pending...
##      lambda   R2  R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.72 0.63 0.29 0.63 0.47     31%         65%
##                                      ...component 2 built!
## Should we build component 3 ? Bootstrap pending...
##                                  ...component 3 not built!
## =====================                =====================
##                      ================
```



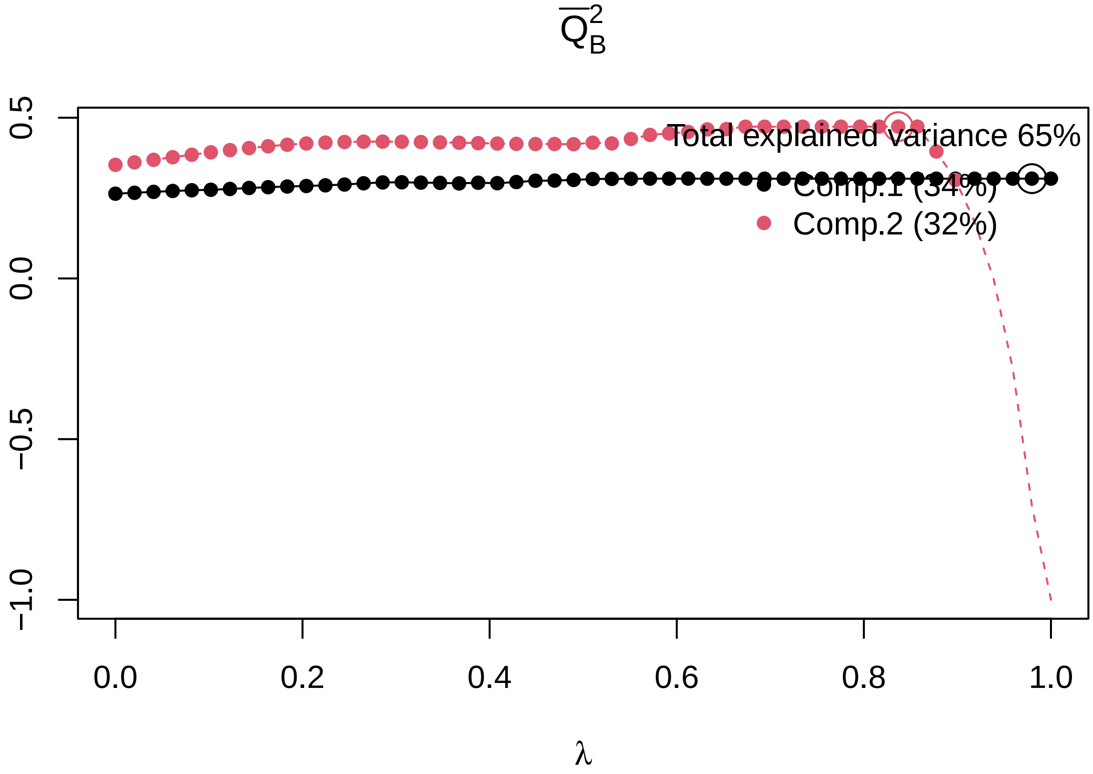
```r
model_1_lambda <- ddsPLS(simu_1$X, simu_1$Y,
                         criterion = "Q2",
                         lambdas = seq(0,1,length.out = 100),
                         n_B = n_B,
                         verbose = T)
```

```
##                        --------------
##                       |    ddsPLS     |
## =====================----------------=====================
## Should we build component 1 ? Bootstrap pending...
##      lambda   R2  R2h  Q2 Q2h VarExpl VarExpl.Tot
##        0.98 0.34 0.34 0.3 0.3     34%         34%
##                                      ...component 1 built!
## Should we build component 2 ? Bootstrap pending...
##      lambda   R2 R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.81 0.64 0.3 0.62 0.46     32%         65%
##                                  ...component 3 not built!
## =====================                ====================
##                      ================
```



```
model_1_lambda_2 <- ddsPLS(simu_1$X, simu_1$Y,
                         criterion = "Q2",
                         lambdas = seq(0,1,length.out = 50),
                         n_B = n_B,
                         verbose = T)
```

```
##                       ---------------
##                      |    ddsPLS    |
## =====================================-=====================
## Should we build component 1 ? Bootstrap pending...
##      lambda    R2  R2h    Q2  Q2h VarExpl VarExpl.Tot
##        0.98 0.33 0.33 0.31 0.31     34%         34%
##                                       ...component 1 built!
## Should we build component 2 ? Bootstrap pending...
##      lambda    R2  R2h    Q2  Q2h VarExpl VarExpl.Tot
##        0.84 0.63 0.29 0.63 0.47     32%         65%
##                                       ...component 2 built!
## Should we build component 3 ? Bootstrap pending...
##                                   ...component 3 not built!
## =====================                ====================
##                      ================
```



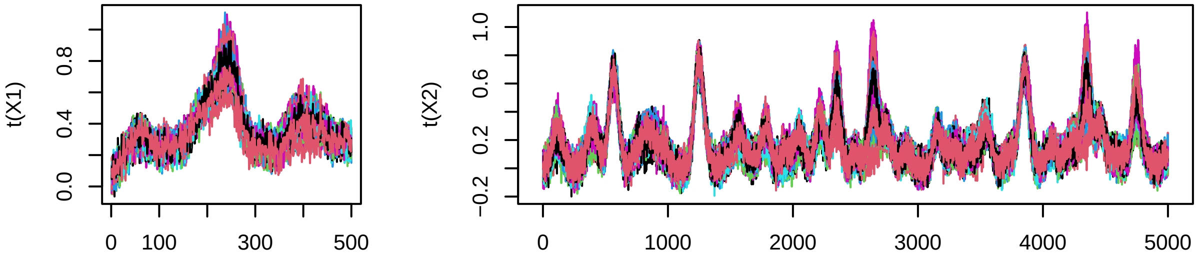
Different Simulations of Design 1 Data

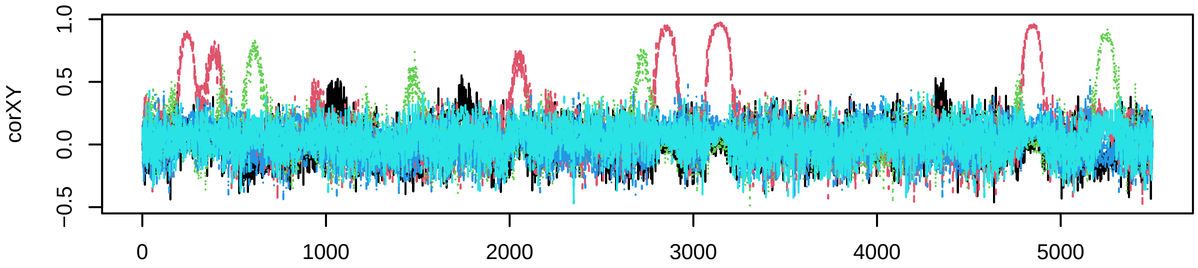
There is a problem with `get_design_1`, `q` cannot take values other than 5.

```
data_1 <- get_design_1(n = 100, p = 1000, q = 5)

ddsPLS(data_1$X,data_1$Y,
                  criterion = "Q2",
                  lambdas = lambdas,
                  n_B=n_B,
                  verbose=T)
```

Design 2

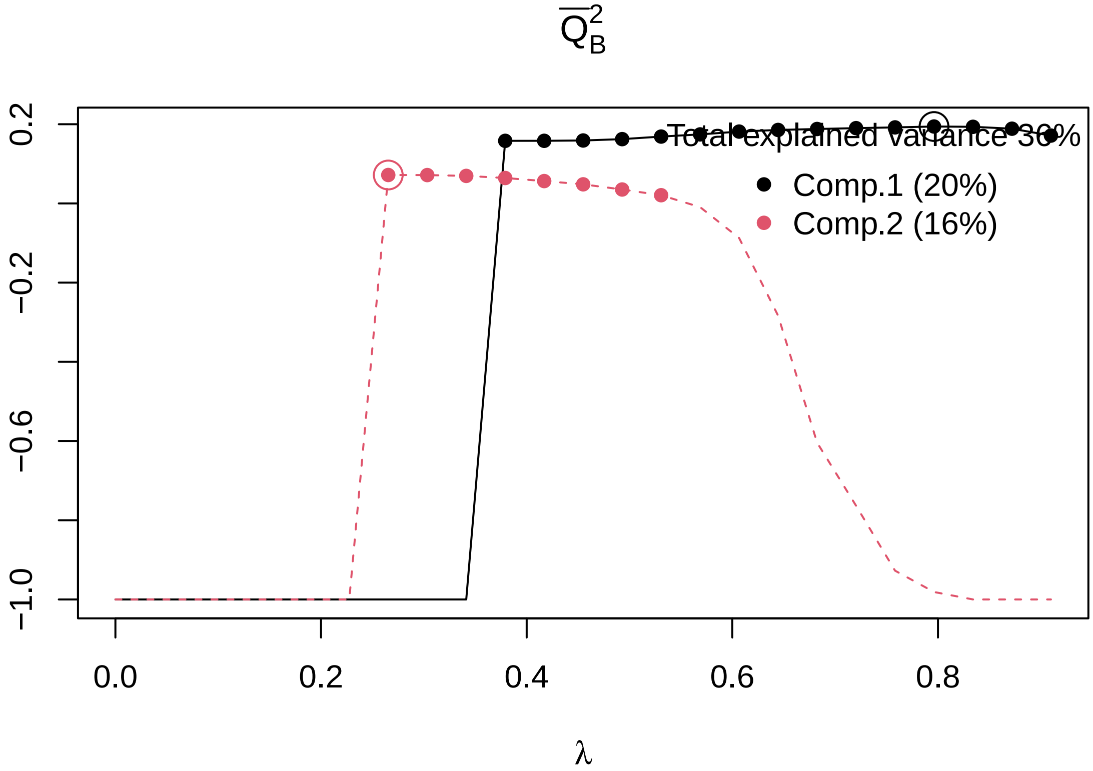
```
simu_2.1 <- get_design_2(plot = T)
```





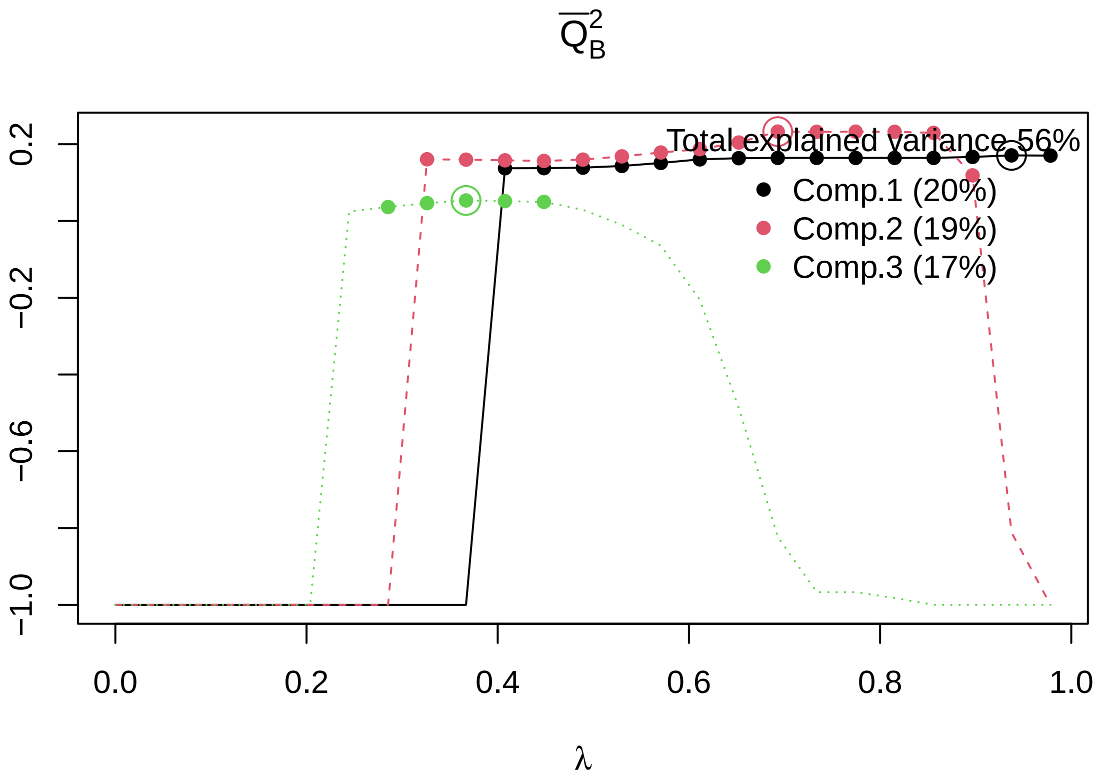
```r
model_2.1 <- ddsPLS(simu_2.1$X$X1, simu_2.1$Y,
                    criterion = "Q2",
                    n_lambdas = 25,
                    verbose = TRUE)
```

```
##                        --------------
##                       |    ddsPLS    |
## =====================----------------=====================
## Should we build component 1 ? Bootstrap pending...
##      lambda   R2  R2h   Q2  Q2h VarExpl VarExpl.Tot
##         0.8 0.19 0.19 0.19 0.19     20%         20%
##                                      ...component 1 built!
## Should we build component 2 ? Bootstrap pending...
##      lambda  R2 R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.27 0.4 0.2 0.24 0.07     16%         36%
##                                      ...component 2 built!
## Should we build component 3 ? Bootstrap pending...
##                                  ...component 3 not built!
## =====================                =====================
##                      ================
```

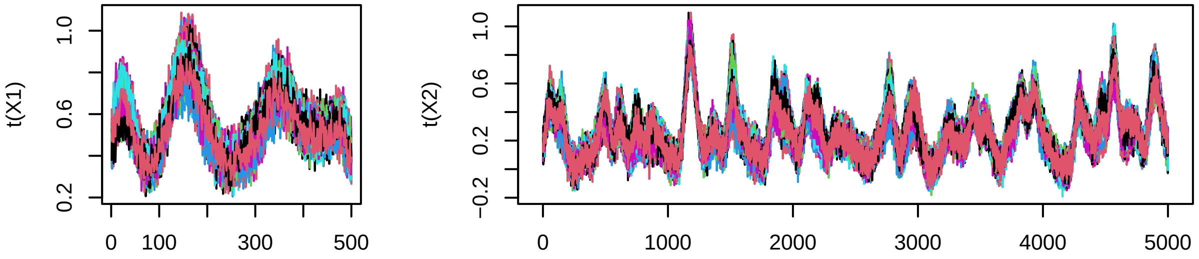


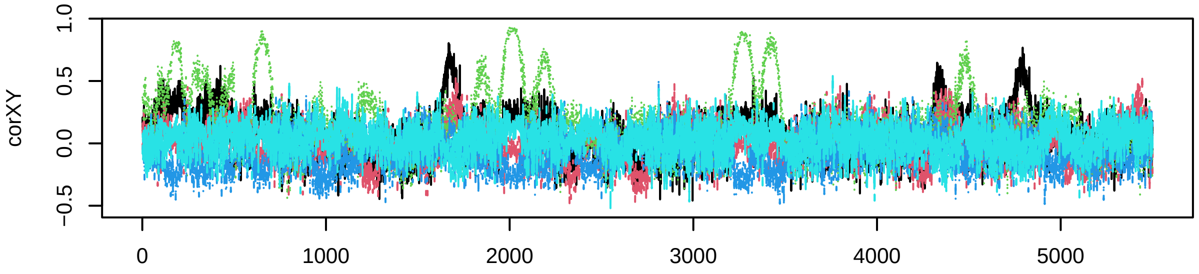
```r
model_2.2 <- ddsPLS(simu_2.1$X$X2, simu_2.1$Y,
                    criterion = "Q2",
                    n_lambdas = 25,
                    verbose = TRUE)
```

```
##                         --------------
##                        |    ddsPLS    |
## =====================----------------=====================
## Should we build component 1 ? Bootstrap pending...
##      lambda  R2 R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.94 0.2 0.2 0.17 0.17     20%         20%
##                                      ...component 1 built!
## Should we build component 2 ? Bootstrap pending...
##      lambda   R2  R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.69 0.38 0.19 0.38 0.23      19%         39%
##                                      ...component 2 built!
## Should we build component 3 ? Bootstrap pending...
##      lambda   R2  R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.37 0.57 0.19 0.39 0.05      17%         56%
##                                      ...component 3 built!
## Should we build component 4 ? Bootstrap pending...
##                                  ...component 4 not built!
## =====================                =====================
##                        ================
```



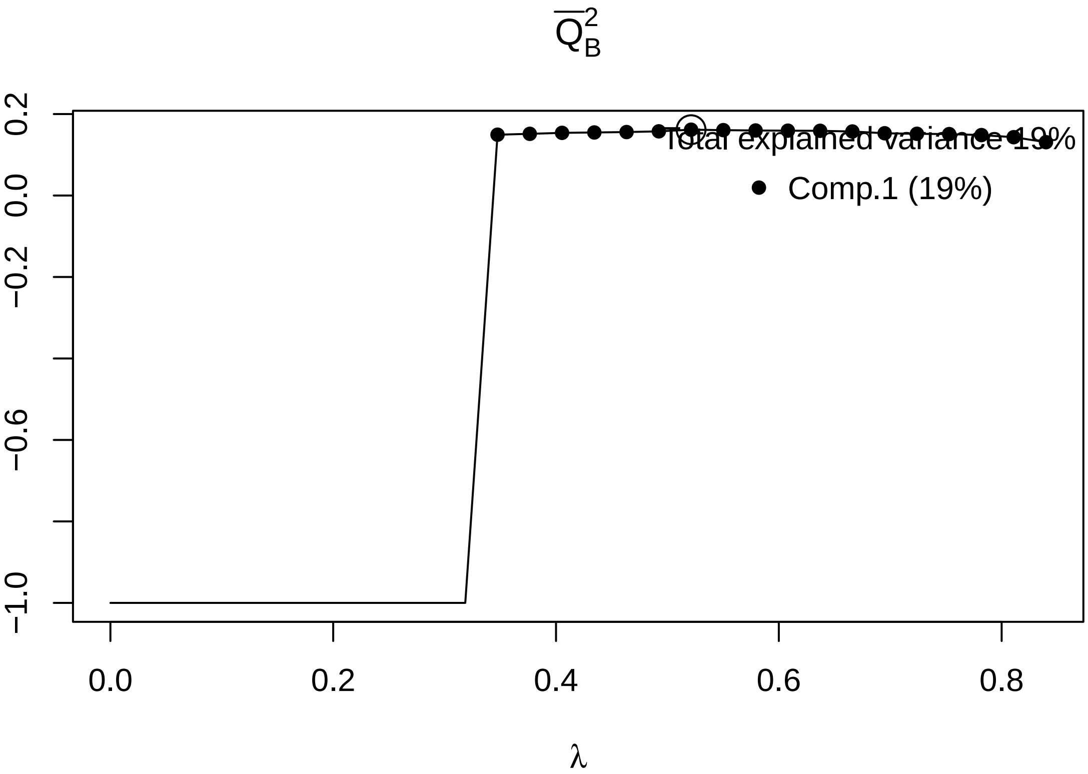
```r
simu_2.2 <- get_design_2(seed = 2, ncpX = 20, plot = T)
```





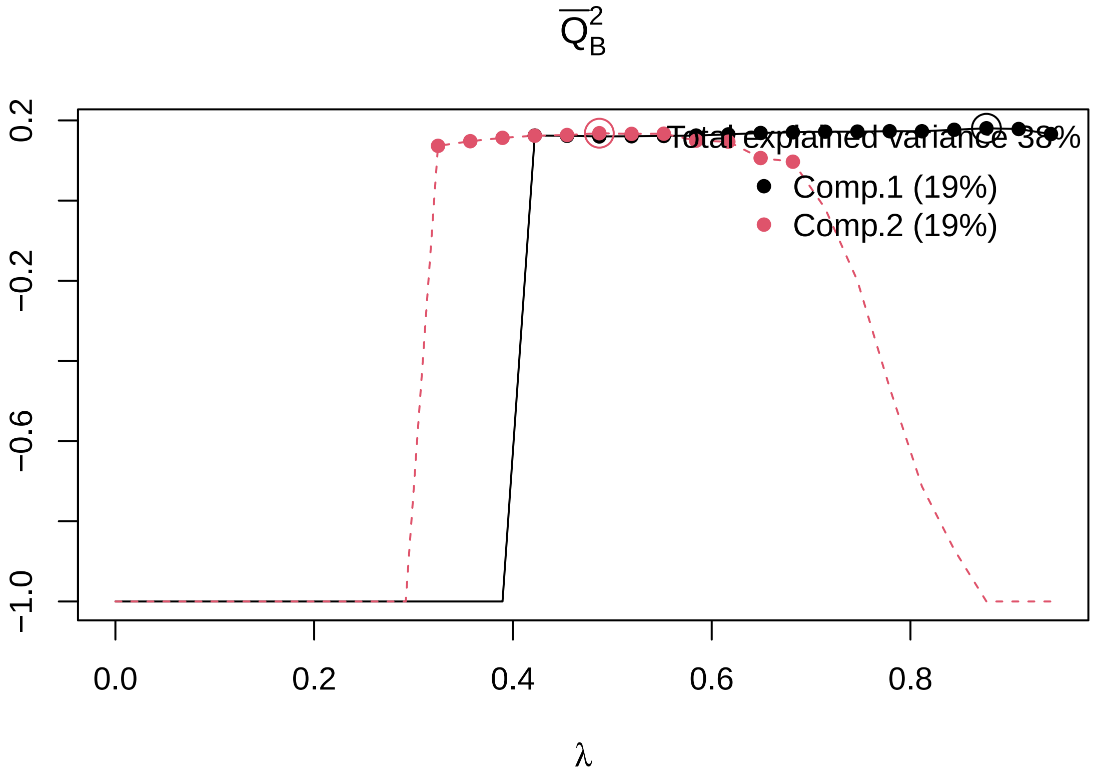
```r
model_2.3 <- ddsPLS(simu_2.2$Xs$X1, simu_2.2$Y,
                    criterion = "Q2",
                    n_lambdas = 30,
                    verbose = TRUE)
```

```
##                       --------------
##                      |    ddsPLS    |
## =====================----------------=====================
## Should we build component 1 ? Bootstrap pending...
##      lambda   R2  R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.52 0.22 0.22 0.16 0.16     19%         19%
##                                      ...component 1 built!
## Should we build component 2 ? Bootstrap pending...
##                                  ...component 2 not built!
## =====================                =====================
##                      ================
```



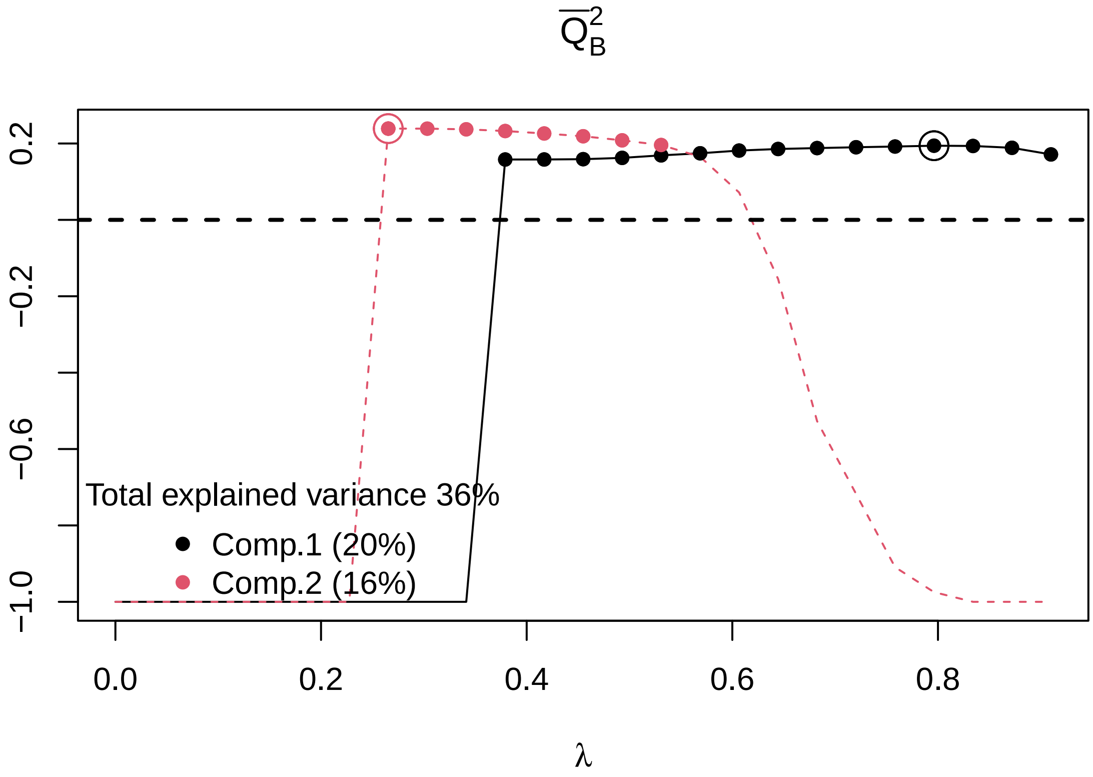
```r
model_2.4 <- ddsPLS(simu_2.2$Xs$X2, simu_2.2$Y,
                    criterion = "Q2",
                    n_lambdas = 30,
                    verbose = TRUE)
```

```
##                       --------------
##                      |    ddsPLS    |
## =====================----------------=====================
## Should we build component 1 ? Bootstrap pending...
##      lambda  R2 R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.88 0.2 0.2 0.18 0.18     19%         19%
##                                      ...component 1 built!
## Should we build component 2 ? Bootstrap pending...
##      lambda   R2 R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.49 0.39 0.2 0.32 0.17     19%         38%
##                                      ...component 2 built!
## Should we build component 3 ? Bootstrap pending...
##                                  ...component 3 not built!
## =====================                =====================
##                                  ================
```



Model results can also be plotted using the `plot` function.

```
plot(model_2.1,type="Q2",legend.position = "bottomleft")
```



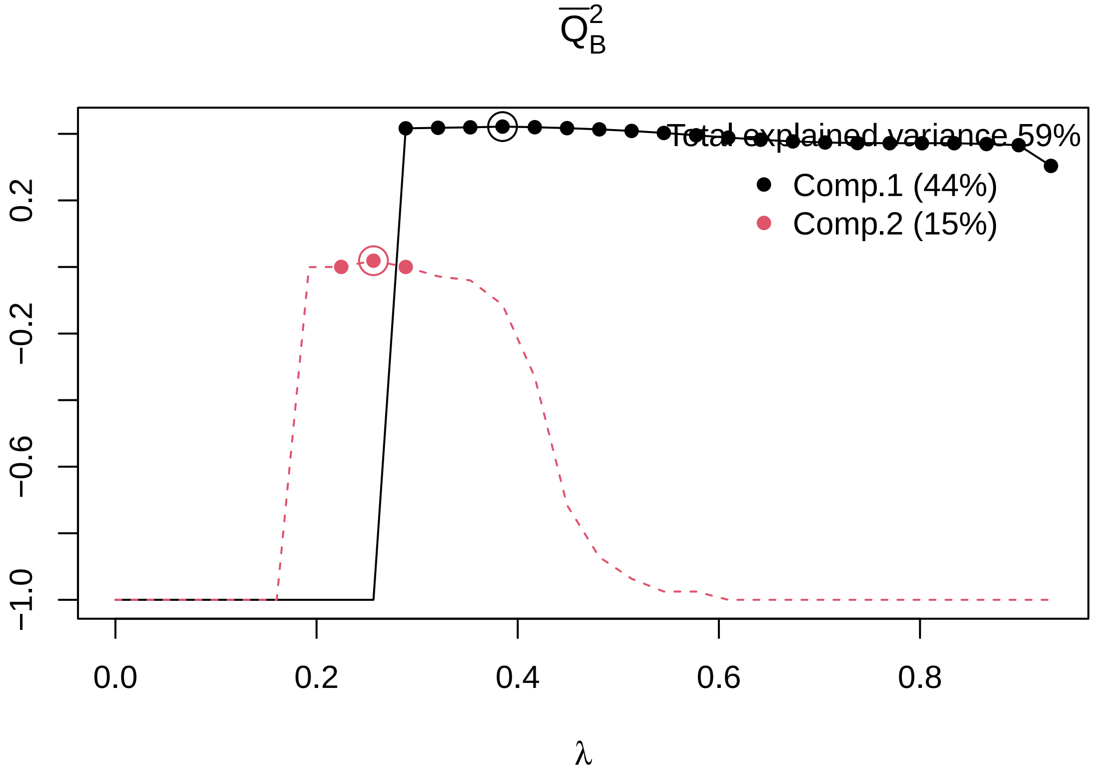
Get Data Simulation

The following get_data function is from the vignette for the ddsPLS package

The variable eps seems to relate the predictors and response, as well as phi. The dimension of phi specifies the number of latent variables.

```
data_3.1 <- get_data()

model_3.1 <- ddsPLS(data_3.1$X, data_3.1$Y,
                    criterion = "Q2",
                    n_lambdas = 30,
                    verbose = TRUE)
```

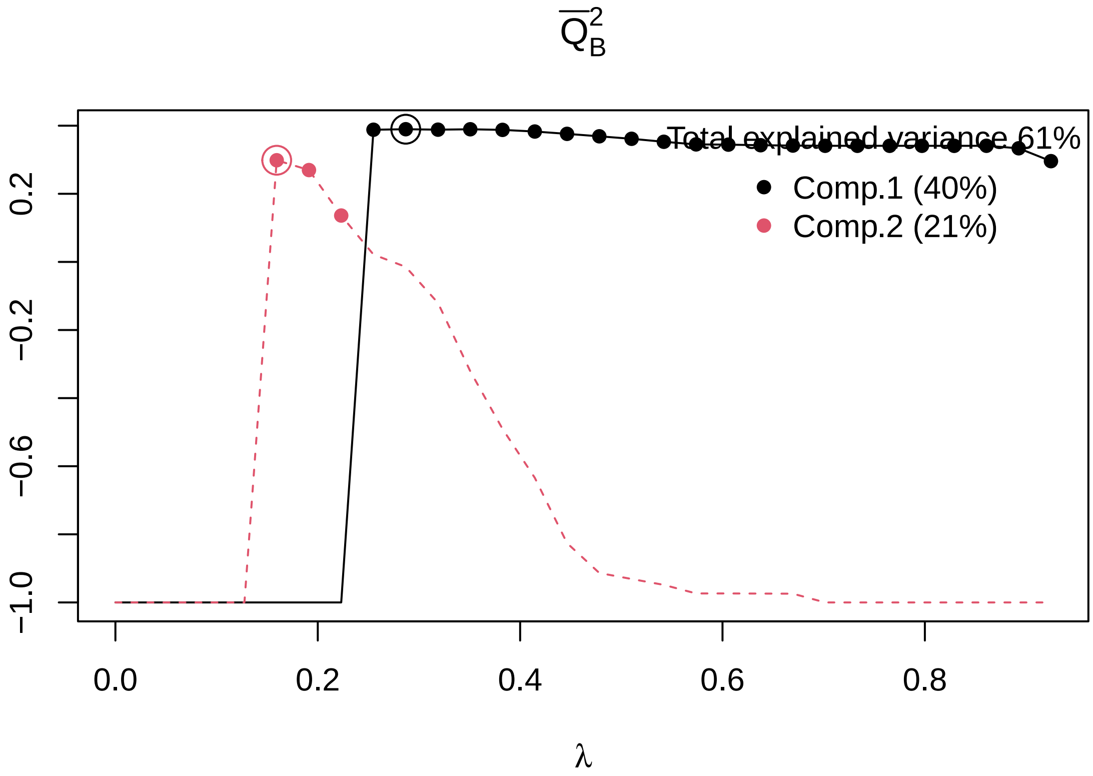
```
##                       --------------
##                      |    ddsPLS    |
## =====================----------------=====================
## Should we build component 1 ? Bootstrap pending...
##      lambda    R2   R2h    Q2   Q2h VarExpl VarExpl.Tot
##        0.38 0.43 0.43 0.42 0.42     44%         44%
##                                      ...component 1 built!
## Should we build component 2 ? Bootstrap pending...
##      lambda    R2 R2h    Q2   Q2h VarExpl VarExpl.Tot
##        0.26 0.63 0.2 0.43 0.02     15%         59%
##                                      ...component 2 built!
## Should we build component 3 ? Bootstrap pending...
##                                  ...component 3 not built!
## =====================                =====================
##                      ================
```



```
data_3.2 <- get_data(p1 = 50, p2 = 50, p3 = 50, p = 250)
```

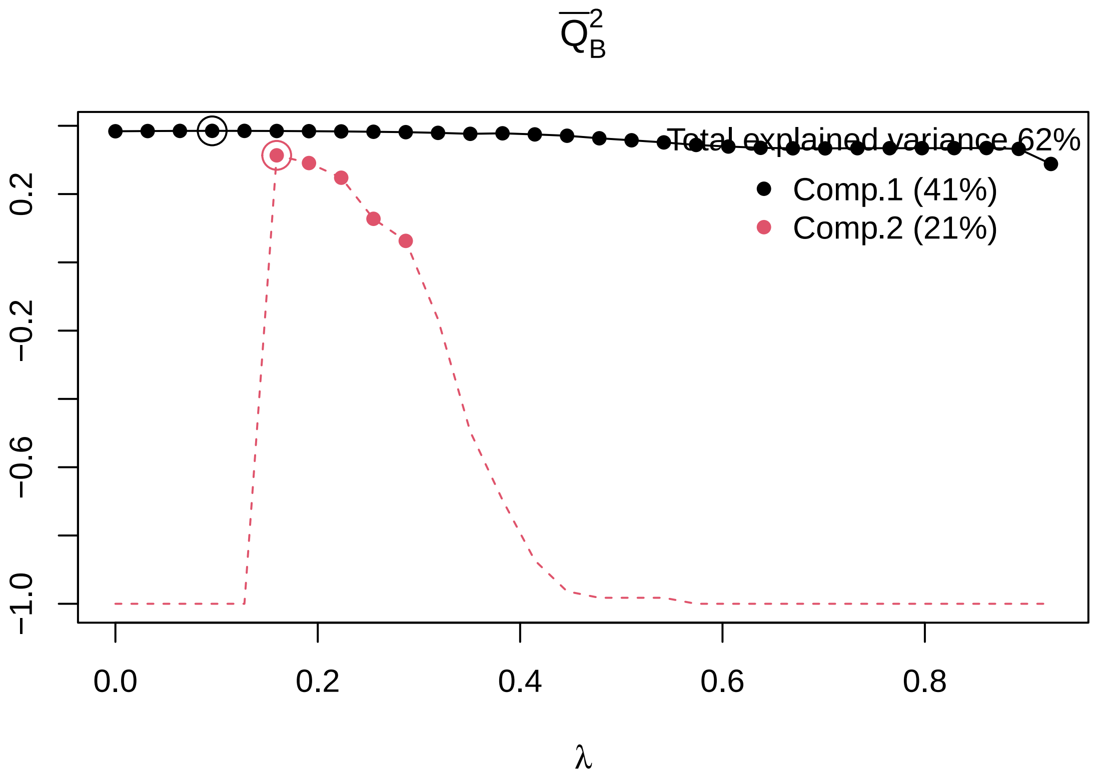
```
model_3.2 <- ddsPLS(data_3.2$X, data_3.2$Y,
                    criterion = "Q2",
                    n_lambdas = 30,
                    verbose = TRUE)
```

```
##                       --------------
##                      |    ddsPLS    |
## =====================----------------=====================
## Should we build component 1 ? Bootstrap pending...
##      lambda   R2  R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.29 0.41 0.41 0.39 0.39     40%         40%
##                                     ...component 1 built!
## Should we build component 2 ? Bootstrap pending...
##      lambda   R2 R2h   Q2 Q2h VarExpl VarExpl.Tot
##        0.16 0.62 0.2 0.55 0.3     21%         61%
##                                     ...component 2 built!
## Should we build component 3 ? Bootstrap pending...
##                                 ...component 3 not built!
## =====================                =====================
##                      ================
```



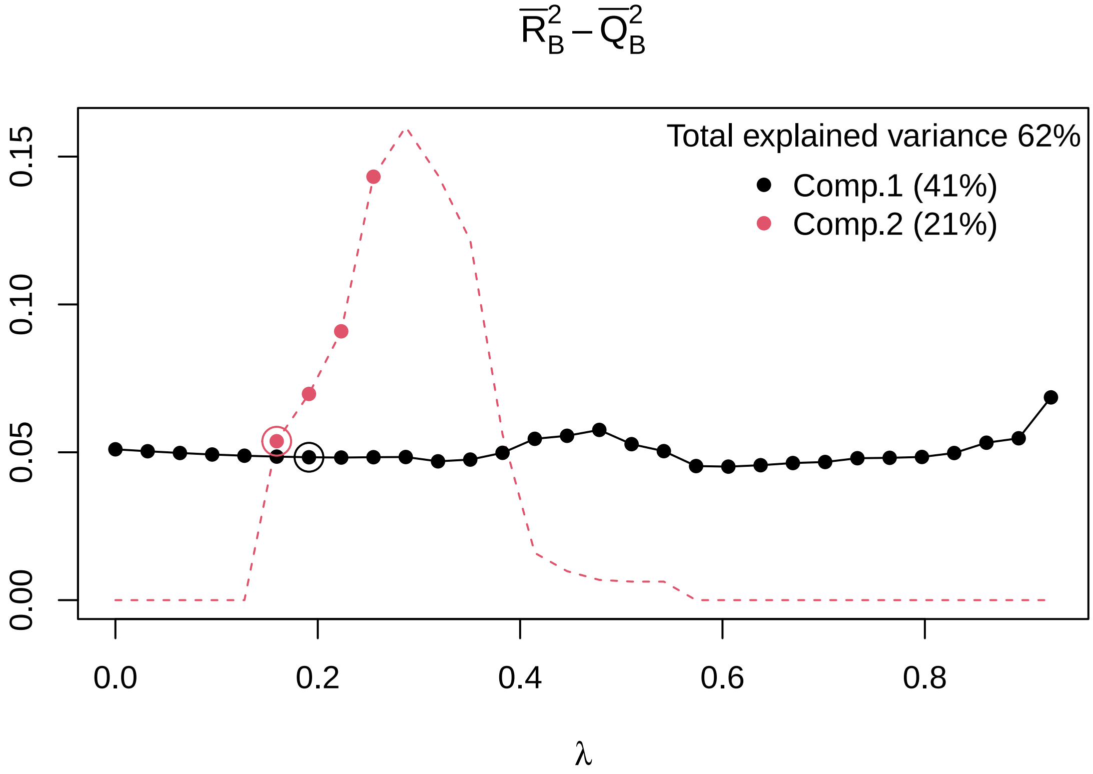
```
model_3.3 <- ddsPLS(data_3.2$X, data_3.2$Y,
                    criterion = "Q2",
                    n_lambdas = 30,
                    verbose = TRUE,
                    LD = TRUE)
```

```
##                       --------------
##                      |    ddsPLS    |
## =====================----------------=====================
## Should we build component 1 ? Bootstrap pending...
##      lambda  R2 R2h   Q2  Q2h VarExpl VarExpl.Tot
##         0.1 0.4 0.4 0.39 0.39     41%         41%
##                                      ...component 1 built!
## Should we build component 2 ? Bootstrap pending...
##      lambda   R2  R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.16 0.62 0.21 0.57 0.31     21%         62%
##                                      ...component 2 built!
## Should we build component 3 ? Bootstrap pending...
##                                  ...component 3 not built!
## =====================                =====================
##                      ================
```



```r
model_3.4 <- ddsPLS(data_3.2$X, data_3.2$Y,
                    criterion = "diffR2Q2",
                    n_lambdas = 30,
                    verbose = TRUE,
                    LD = TRUE)
```

```
##                       --------------
##                      |    ddsPLS    |
## =====================----------------=====================
## Should we build component 1 ? Bootstrap pending...
##      lambda   R2  R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.19 0.42 0.42 0.37 0.37     41%         41%
##                                      ...component 1 built!
## Should we build component 2 ? Bootstrap pending...
##      lambda   R2  R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.16 0.62 0.21 0.57 0.29     21%         62%
##                                      ...component 2 built!
## Should we build component 3 ? Bootstrap pending...
##                                  ...component 3 not built!
## =====================                ====================
##                      ================
```



Novel Simulations

The following code simulates a data set with 100 uncorrelated predictors and 1 response variable all sampled from a normal distribution.

```
Sigma <- diag(100)

sim_pred <- mvrnorm(n = 1000, mu = rep(0, 100), Sigma = Sigma)

sim_resp <- matrix(rnorm(1000), 1000, 1)
```

```
ddsPLS(sim_pred, sim_resp, verbose = TRUE)
```

```
##                        ______________
##                       |    ddsPLS    |
## =====================----------------====================
## Should we build component 1 ? Bootstrap pending...
##                                  ...component 1 not built!
##              ...no Q2r large enough for tested lambda.
## =====================                ====================
##                      ================
```

```
##
## Call:
## NULL
```

```
##
## No ddsPLS model built.
```

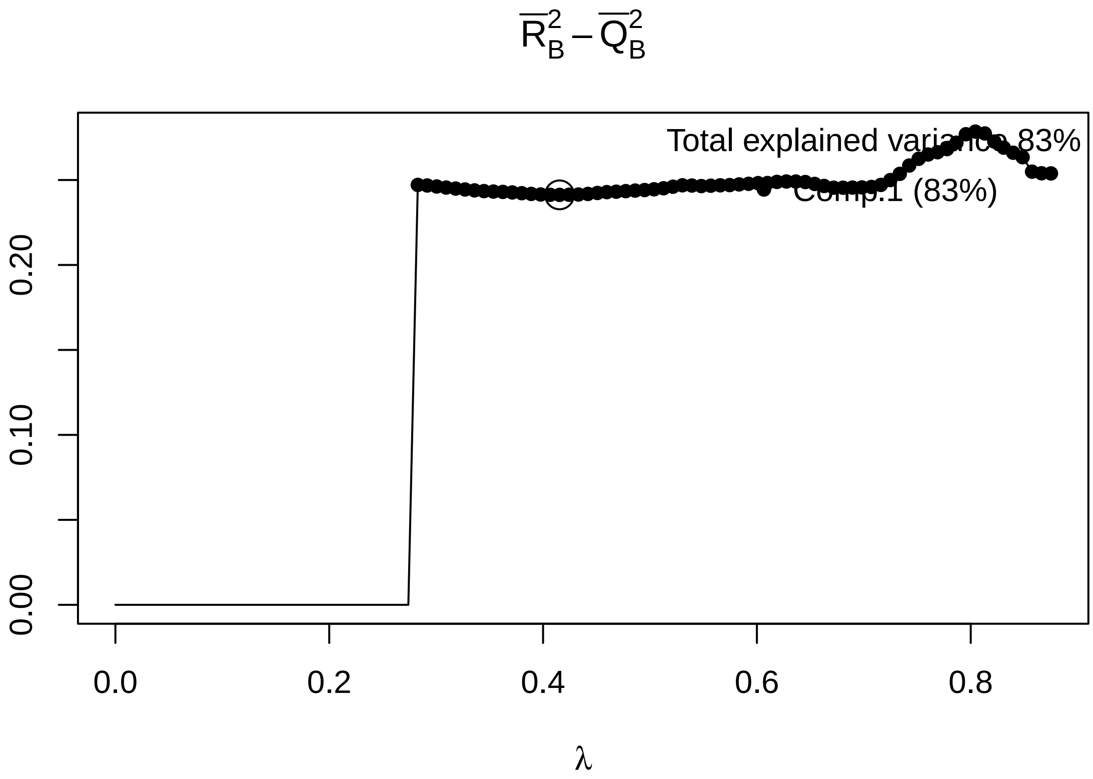
As expected no model is built as performance is awful. Interestingly message “no Q2r large enough for tested lambda” is given for justification, seems to suggest it checks just Q^2. Perhaps this just means that mean estimation performs better.

```
Sigma <- matrix(c(1,.75,.75,1),2,2)

n <- 20
p <- 5
p <- p - 2

sim_preds <- cbind(mvrnorm(n = n, rep(0, 2), Sigma), matrix(rep(0,n*p),n, p))

sim_resp <- as.matrix(apply(sim_preds,1,function(x) 5*x[1]+x[2]))

sim_preds <- sim_preds + matrix(rnorm(n*(p+2), sd = 0.6), n,(p+2))
sim_resp <- sim_resp + matrix(rnorm(n, sd = 0.8), n,1)
```

The above code simulates data with n observations of p predictors and 1 response variable. There are two predictors from which responses are linearly generated. Random noise is then added to the predictors and the response. As expected, the ddsPLS model performs very well.

```
pls_model <- ddsPLS(sim_preds, sim_resp, verbose = TRUE)
```

```
##                       ---------------
##                      |    ddsPLS    |
## =====================----------------=====================
## Should we build component 1 ? Bootstrap pending...
##      lambda   R2  R2h   Q2  Q2h VarExpl VarExpl.Tot
##        0.42 0.78 0.78 0.54 0.54     83%         83%
##                                      ...component 1 built!
## Should we build component 2 ? Bootstrap pending...
##                                  ...component 2 not built!
## =====================                =====================
##                      ================
```



Complex Simulated Data

The general structure of simulated data is $\mathbf{X} = \mathbf{A}^T\phi + \epsilon_X$ and $\mathbf{Y} = \mathbf{D}^T\phi + \epsilon_Y$. Note that ϕ provides the structure between the two. Code structures it as $\mathbf{X} = \phi\mathbf{A} + \epsilon_X$ and similarly for $\mathbf{Y}$. ϵ is added random error. $\text{Cov}(\mathbf{X}, \mathbf{Y}) = \mathbf{D}^T\mathbf{A}$.

```
## Right now this doesn't work for most values, need to find a way to generate A,D, and phi based on ar

sim_data <- function(n = 5, p = 10, q = 2) {

  A <- diag(p)
  D <- matrix(c(rep(1, n), rep(0,p), rep(1,n)), nrow = p)

  phi <- diag(p)[1:n,]

  epsilon_X <- mvrnorm(n = dim(phi)[1],
                       rep(0, c(dim(A)[2])),
                       Sigma = diag(dim(A)[2]))

  epsilon_Y <- mvrnorm(n = dim(phi)[1],
                       rep(0, c(dim(D)[2])),
                       Sigma = diag(dim(D)[2]))

  X <- phi %*% A + epsilon_X
  Y <- phi %*% D + epsilon_Y


  list(X=X, Y=Y)
}
```

```r
n=50
sqrt_1_minus_sig2=0.99
p=1000
q=3
  # Structure
  alpha3 <- 1/sqrt(3)
  alpha2 <- 1/sqrt(2)
  repX <- 50
  A1 <- c(rep(alpha3,repX),rep(0,p-repX))
  A2 <- c(rep(0,repX),rep(alpha2,repX),rep(0,p-2*repX))
  A <- matrix(c(rep(A1,3),rep(A2,2)),nrow = 5,byrow = T)*sqrt_1_minus_sig2
  D1 <- c(rep(alpha3,1),rep(0,q-1))
  D2 <- c(rep(0,1),rep(alpha2,1),rep(0,q-2))
  D <- matrix(c(rep(D1,3),rep(D2,2)),nrow = 5,byrow = T)*sqrt_1_minus_sig2
  # Observations
  d <- ncol(A)+nrow(A)+ncol(D)
  psi <- MASS::mvrnorm(n = n,mu = rep(0,d),Sigma = diag(d))
  phi <- psi[,1:nrow(A)]
  epsilonX_info <- psi[,nrow(A)+1:(2*repX)]*sqrt(1-sqrt_1_minus_sig2^2)
  epsilonX_noise <- psi[,nrow(A)+(2*repX)+1:(ncol(A)-2*repX)]
  epsilonY_info <- psi[,nrow(A)+ncol(A)+1:2]*sqrt(1-sqrt_1_minus_sig2^2)
  epsilonY_noise <- psi[,d]
  # X and Y
  X <- phi%*%A + cbind(epsilonX_info,epsilonX_noise)
  Y <- phi%*%D + cbind(epsilonY_info,epsilonY_noise)
```