

# KPR: Stability Testing

*Parker Knight*

*1/4/2020*

The purpose of this experiment is to study the relative change in KPR coefficients after making permutations to the Q, H, and Z matrices.

The file “helpers.R” loads the yatsunenkenko data set, and includes helper functions for comparing models.

```
source("helpers.R")
```

```
## Loading KPR
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

## Q

### Eigenvalue Permutations

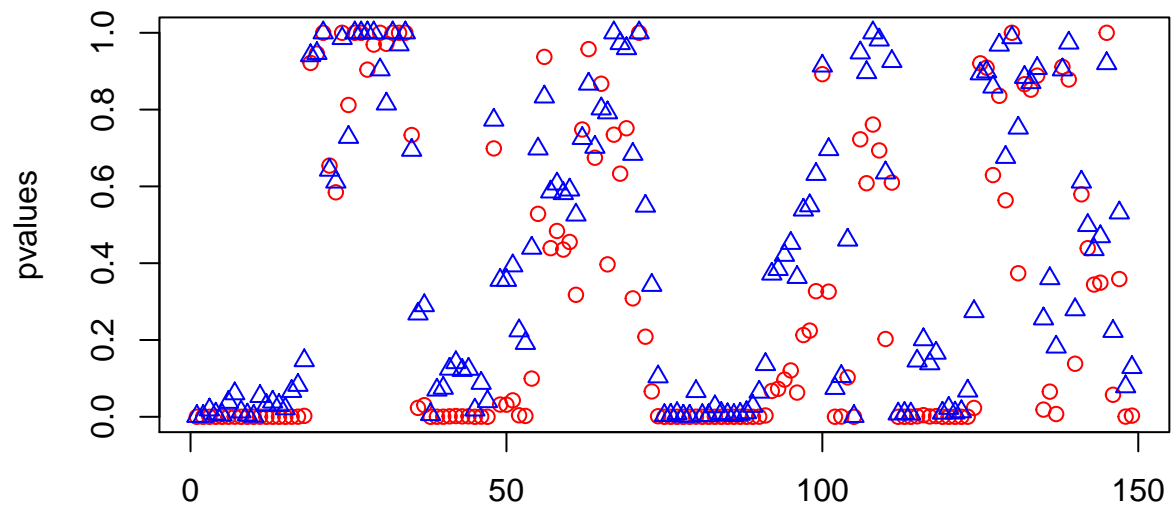
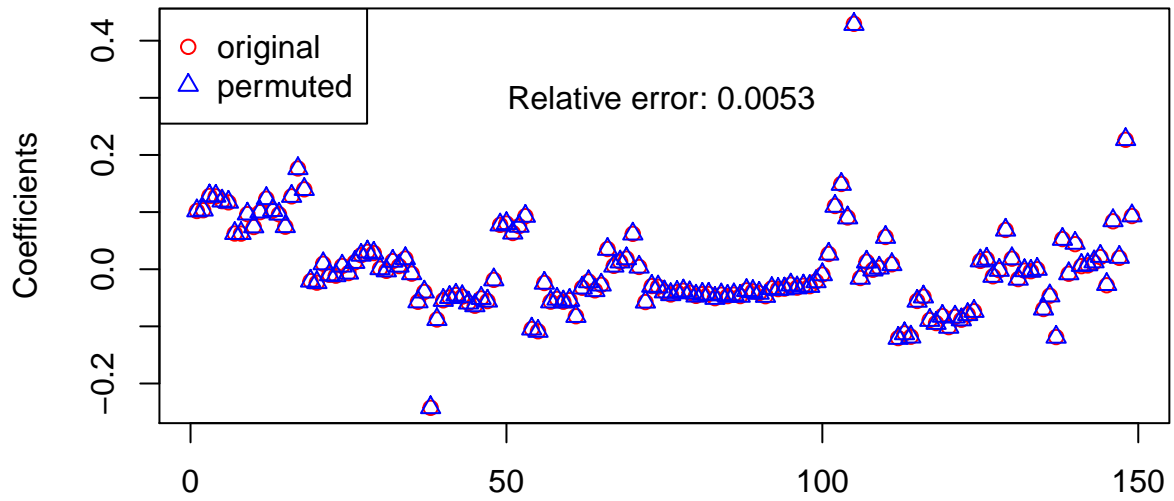
```
Q <- generateSimilarityKernel(patristic)

## Correcting small and negative eigenvalues
Q.1 <- permuteEigenvalue(Q, 1)
Q.2 <- permuteEigenvalue(Q, 75)
Q.3 <- permuteEigenvalue(Q, 149)

fit <- KPR(designMatrix = Z, Y = Y, Q = Q)
fit.1 <- KPR(designMatrix = Z, Y = Y, Q = Q.1)
fit.2 <- KPR(designMatrix = Z, Y = Y, Q = Q.2)
fit.3 <- KPR(designMatrix = Z, Y = Y, Q = Q.3)

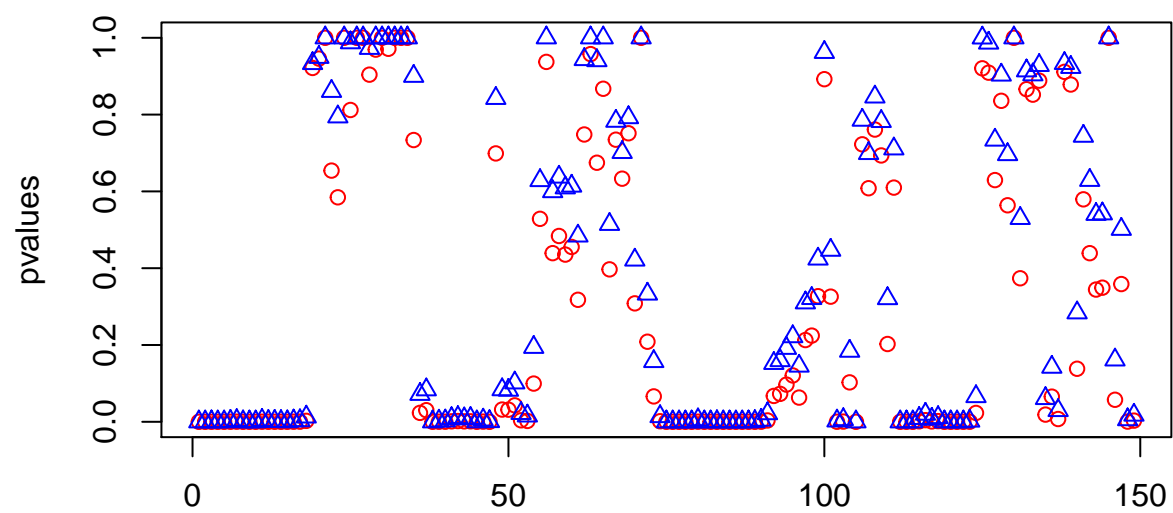
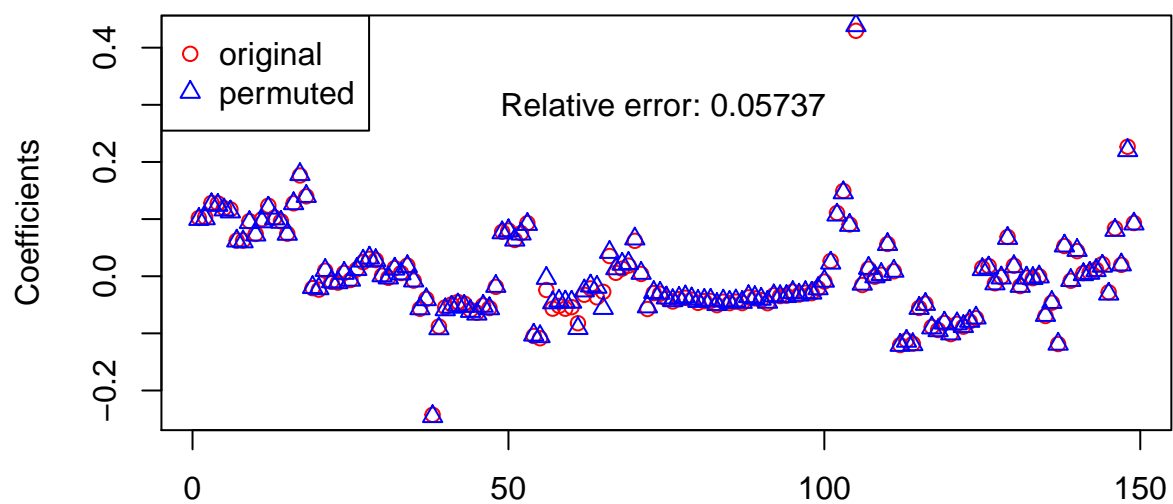
compareModels(fit, fit.1, main = "Eigenvalue 1")
```

## Eigenvalue 1



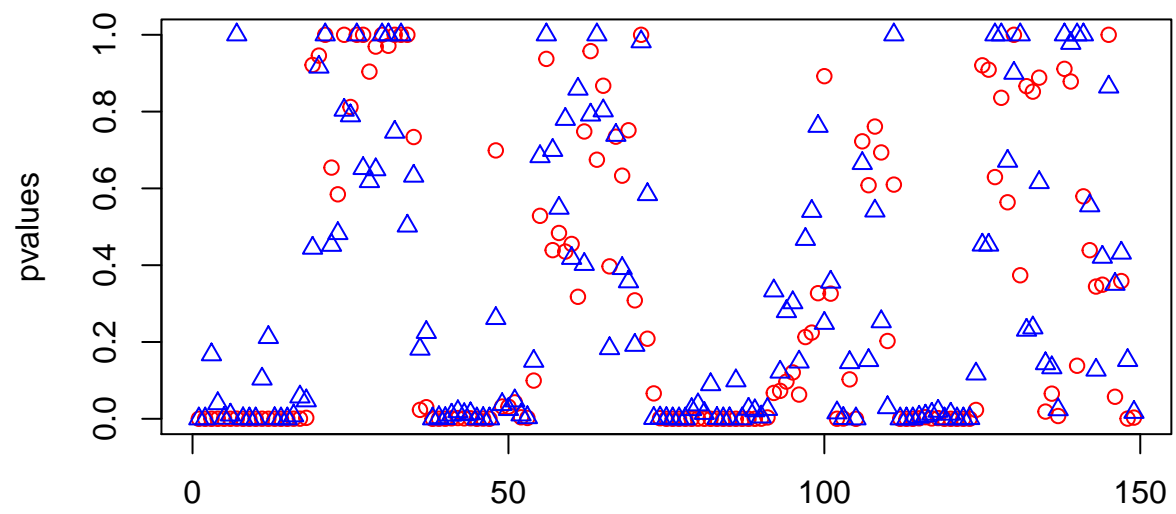
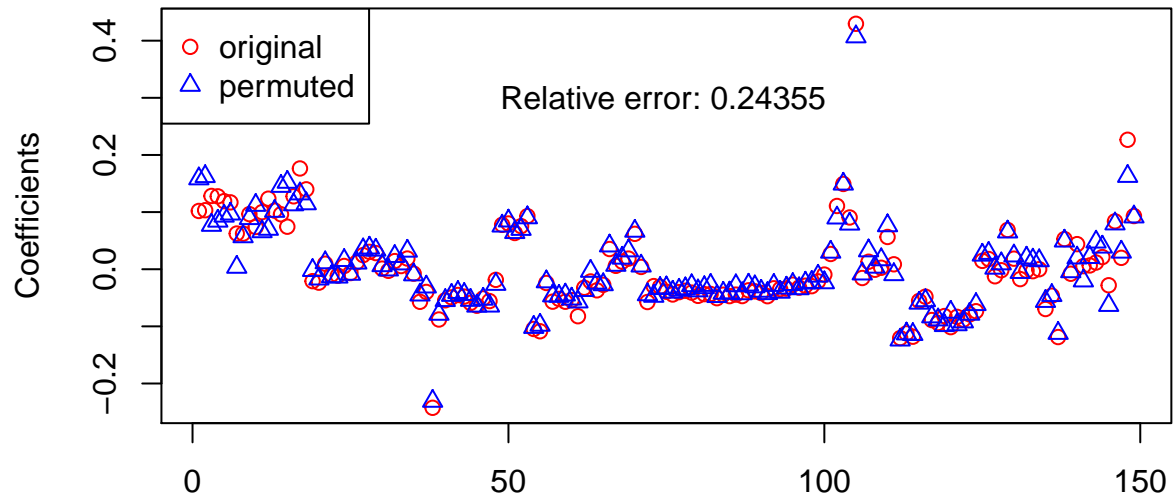
```
compareModels(fit, fit.2, main = "Eigenvalue 75")
```

## Eigenvalue 75



```
compareModels(fit, fit.3, main = "Eigenvalue 149")
```

## Eigenvalue 149



# H

## Eigenvalue Permutations

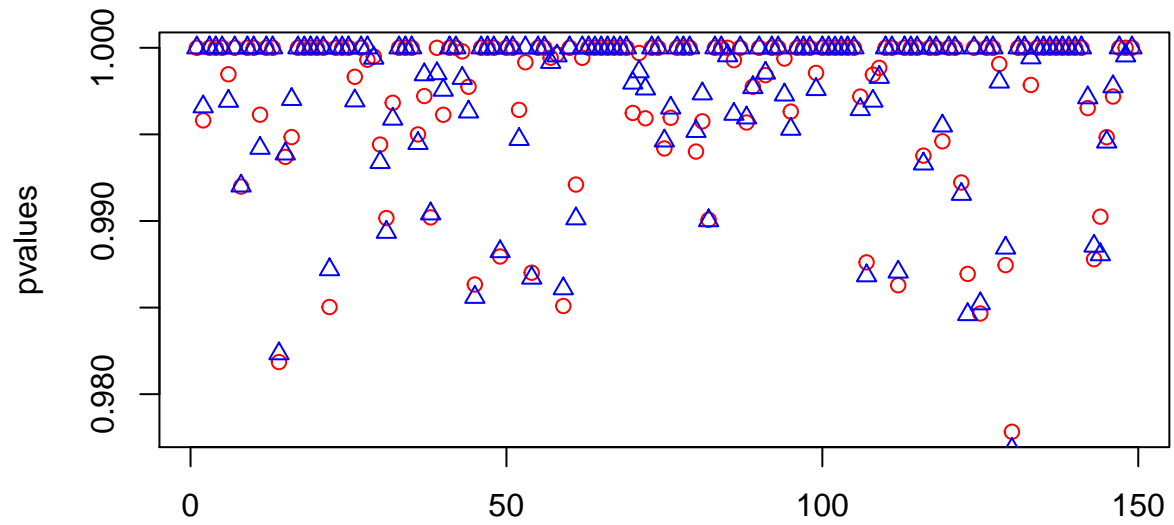
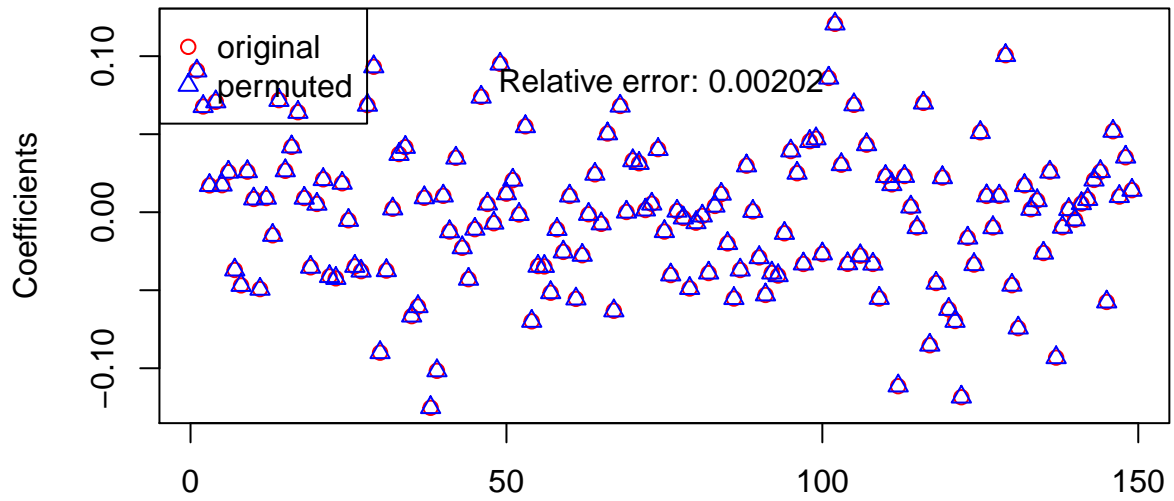
```
H <- generateSimilarityKernel(unifrac)

## Correcting small and negative eigenvalues
H.1 <- permuteEigenvalue(H, 1)
H.2 <- permuteEigenvalue(H, 50)
H.3 <- permuteEigenvalue(H, 100)

fit <- KPR(designMatrix = Z, Y = Y, H = H)
fit.1 <- KPR(designMatrix = Z, Y = Y, H = H.1)
fit.2 <- KPR(designMatrix = Z, Y = Y, H = H.2)
fit.3 <- KPR(designMatrix = Z, Y = Y, H = H.3)

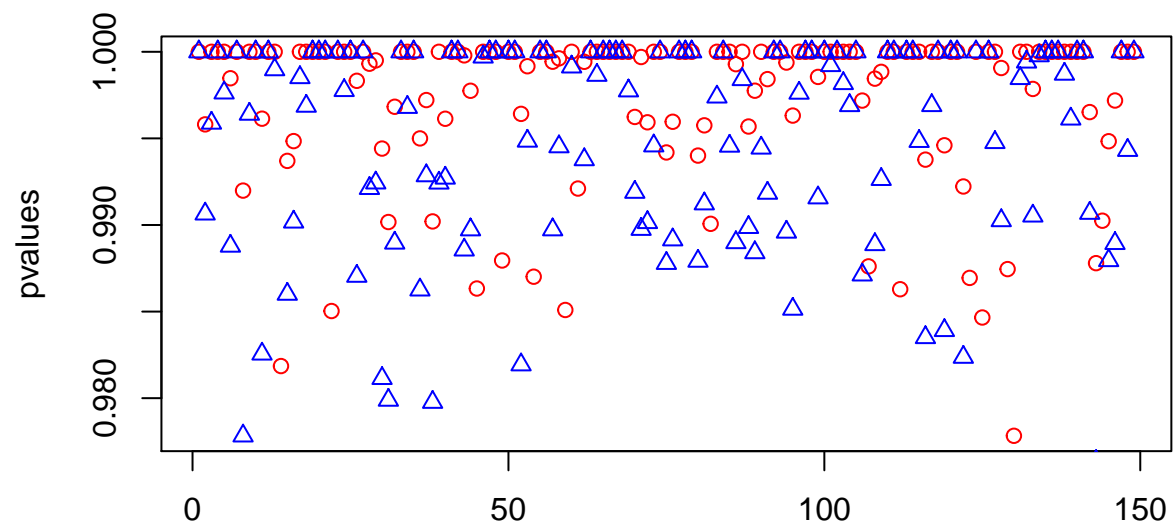
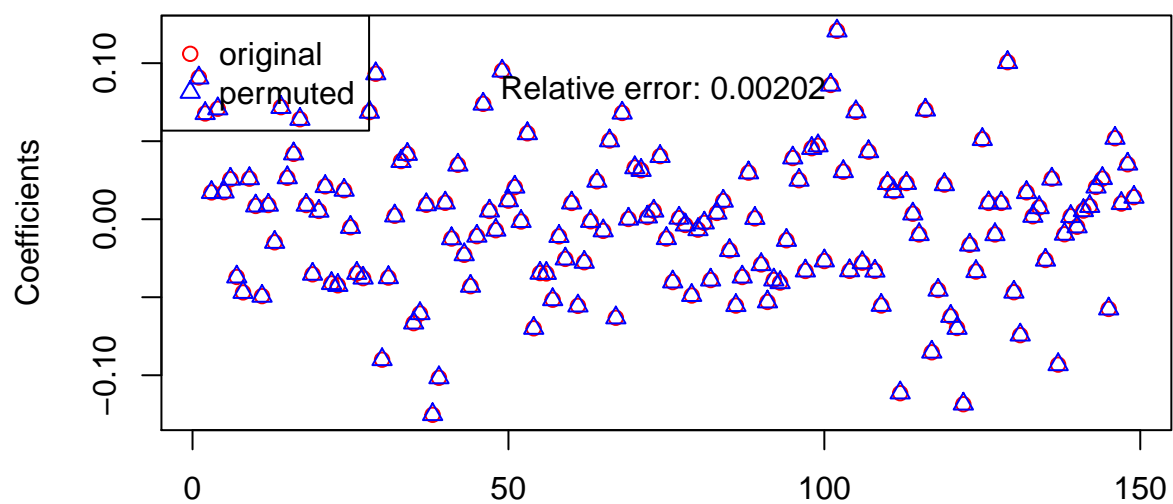
compareModels(fit, fit.1, main = "Eigenvalue 1")
```

## Eigenvalue 1



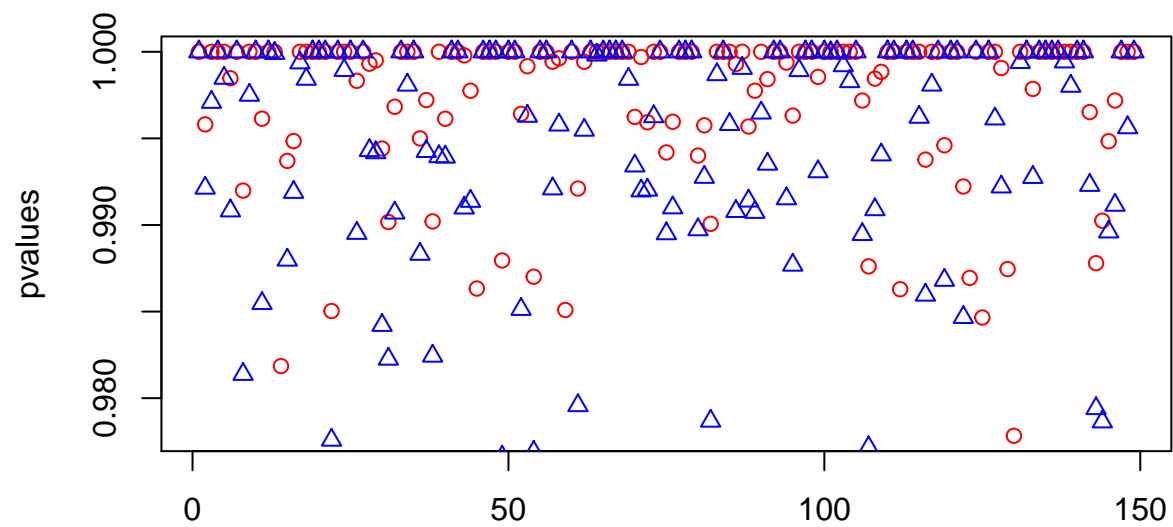
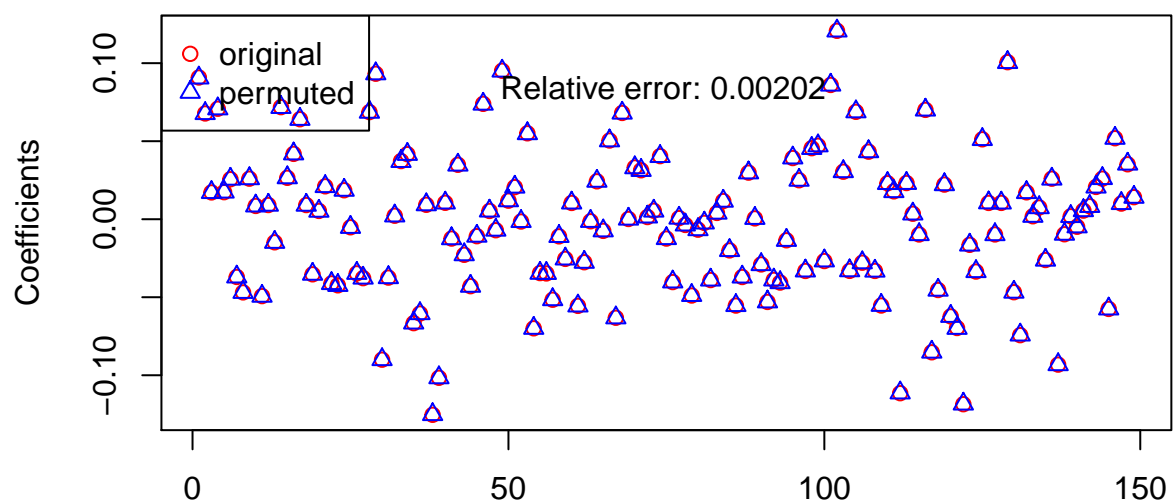
```
compareModels(fit, fit.2, main = "Eigenvalue 50")
```

## Eigenvalue 50



```
compareModels(fit, fit.3, main = "Eigenvalue 100")
```

## Eigenvalue 100





**Z**

## Gaussian Noise

```
Z.1 <- Z + rnorm(n = length(Z))  
fit <- KPR(designMatrix = Z, Y = Y)  
fit.1 <- KPR(designMatrix = Z.1, Y = Y)  
  
compareModels(fit, fit.1, main = "Gaussian noise")
```

## Gaussian noise

