final_real

```r
library(MASS)
library(ggplot2)
library(car)
```

```
Loading required package: carData
```

```r
library(glmnet)
```

```
Loading required package: Matrix

Loaded glmnet 4.1-8
```

```r
library(Matrix)
```

data generating

As you can see, 0.1 and 0.01 is different in the

```r
set.seed(42)
generate_model_1_data <- function(n, att = 1) {
  x1 <- rnorm(n)
  x2 <- 0.9 * x1 + 0.01 * rnorm(n)
  x3 <- rnorm(n)
  X <- cbind(x1, x2, x3)
  y <- 0.2 * x1 + 0.3 * x2 + 0.4 * x3 + att * 0.1 * rnorm(n)
  return(list(X = X, y = y))
}

generate_model_2_data <- function(n, att = 1) {
```

```
  # multi
  x1 <- rnorm(n)
  x3 <- rnorm(n)
  x2 <- 0.45 * x1 + 0.45 * x3 + att * 0.01 * rnorm(n)
  x4 <- rnorm(n)
  X <- cbind(x1, x2, x3, x4)
  y <- 0.2 * x1 + 0.3 * x2 + 0.1 * x3 + 0.39 * x4 + 0.01 * rnorm(n)
  return(list(X = X, y = y))
}
```

VIF test

Show there is always a collinearity whether low or high

```
generate_model_1_data_vif <- function(n, att = 1) {
  x1 <- rnorm(n)
  x2 <- 0.9 * x1 + 0.01 * rnorm(n)
  x3 <- rnorm(n)
  X <- cbind(x1, x2, x3)
  y <- 0.2 * x1 + 0.3 * x2 + 0.4 * x3 + att * 0.1 * rnorm(n)
  return(list(x1 = x1, x2 = x2, x3 = x3, y = y))
}
generate_model_2_data_vif <- function(n, att = 1) {
  x1 <- rnorm(n)
  x3 <- rnorm(n)
  x2 <- 0.45 * x1 + 0.45 * x3 + att * 0.01 * rnorm(n)
  x4 <- rnorm(n)
  X <- cbind(x1, x2, x3, x4)
  y <- 0.2 * x1 + 0.3 * x2 + 0.1 * x3 + 0.39 * x4 + 0.01 * rnorm(n)
  return(list(x1 = x1, x2 = x2, x3 = x3, x4 = x4, y = y))
}

n <- 300
data1 <- generate_model_1_data_vif(n)
data2 <- generate_model_2_data_vif(n)

model_original <- lm(y ~ ., data = data1)
vif(model_original)
```

```
         x1          x2          x3
8123.723862 8124.553191    1.009242
```

```r
model_original <- lm(y ~ ., data = data2)
vif(model_original)
```

```
         x1          x2          x3          x4 
1895.803757 3990.943986 2216.141192    1.004605 
```

```r
data1 <- generate_model_1_data_vif(n, 0.1)
data2 <- generate_model_2_data_vif(n, 0.1)

model_original <- lm(y ~ ., data = data1)
vif(model_original)
```

```
         x1          x2          x3 
9155.062520 9157.703829    1.030218 
```

```r
model_original <- lm(y ~ ., data = data2)
vif(model_original)
```

```
          x1           x2           x3           x4 
2.150360e+05 4.431463e+05 2.223649e+05 1.002214e+00 
```

MC function definition

```r
MC <- function(X, y, Xt, yt) {
  X_df <- data.frame(X)
  # train model
  lm_model <- lm(y ~ ., data = X_df)  # Using all columns in X_df
  stepwise_model <- step(lm_model, direction = "both")
  pca_res <- prcomp(scale(X_df), center = TRUE, scale. = TRUE)
  pca_data <- as.data.frame(pca_res$x)
  pca_model <- lm(y ~ ., data = pca_data)
  ridge_model <- glmnet(X_df, y, alpha = 0)
  ridge_cv <- cv.glmnet(as.matrix(X_df), y, alpha = 0)
  ridge_lambda <- ridge_cv$lambda.min
  lasso_model <- glmnet(X_df, y, alpha = 1)
  lasso_cv <- cv.glmnet(as.matrix(X_df), y, alpha = 1)
  lasso_lambda <- lasso_cv$lambda.min
```

```
  # predict
  X_df <- data.frame(Xt)
  lm_predictions <- predict(lm_model, newdata = X_df)
  stepwise_predictions <- predict(stepwise_model, newdata = X_df)
  pca_predictions <- predict(pca_model, newdata = data.frame(predict(pca_res, newdata = scale
  ridge_predictions <- predict(ridge_model, newx = Xt, s = ridge_lambda)
  lasso_predictions <- predict(lasso_model, newx = Xt, s = lasso_lambda)

  # calculate mse
  lm_mse <- mean((yt - lm_predictions)^2)
  stepwise_mse <- mean((yt - stepwise_predictions)^2)
  pca_mse <- mean((yt - pca_predictions)^2)
  ridge_mse <- mean((yt - ridge_predictions)^2)
  lasso_mse <- mean((yt - lasso_predictions)^2)


  return(list(lm = lm_mse, stepwise = stepwise_mse, pca = pca_mse, ridge = ridge_mse, lasso =
}

MCn <- function(n = 20, t = 10, iterations = 100, at = 1) {
  lm_errors1 <- numeric(iterations)
  stepwise_errors1 <- numeric(iterations)
  pca_errors1 <- numeric(iterations)
  ridge_errors1 <- numeric(iterations)
  lasso_errors1 <- numeric(iterations)
  lm_errors2 <- numeric(iterations)
  stepwise_errors2 <- numeric(iterations)
  pca_errors2 <- numeric(iterations)
  ridge_errors2 <- numeric(iterations)
  lasso_errors2 <- numeric(iterations)

  for (i in 1:iterations) {
    # Generate datasets for both models
    data1 <- generate_model_1_data(n, att = at)
    data2 <- generate_model_2_data(n, att = at)
    data1t <- generate_model_1_data(t, att = at)
    data2t <- generate_model_2_data(t, att = at)

    # Run regression models for both datasets
    result1 <- MC(data1$X, data1$y, data1t$X, data1t$y)
    result2 <- MC(data2$X, data2$y, data2t$X, data2t$y)
```

```
    # Store errors
    lm_errors1[i] <- result1$lm
    stepwise_errors1[i] <- result1$stepwise
    pca_errors1[i] <- result1$pca
    ridge_errors1[i] <- result1$ridge
    lasso_errors1[i] <- result1$lasso
    lm_errors2[i] <- result2$lm
    stepwise_errors2[i] <- result2$stepwise
    pca_errors2[i] <- result2$pca
    ridge_errors2[i] <- result2$ridge
    lasso_errors2[i] <- result2$lasso
  }

  # Return the average errors across all iterations
  return(list(model_co = list(lm_error = mean(lm_errors1), stepwise_error = mean(stepwise_er
}
```

Result return

final

```
simulation_results_10
```

```
$model_co
$model_co$lm_error
[1] 0.004951549

$model_co$stepwise_error
[1] 0.005301927

$model_co$pca_error
[1] 0.2118554

$model_co$ridge_error
[1] 0.01063807

$model_co$lasso_error
[1] 0.006955523
```

```
$model_mulco
$model_mulco$lm_error
[1] 4.678484e-05

$model_mulco$stepwise_error
[1] 0.0001030844

$model_mulco$pca_error
[1] 0.1269923

$model_mulco$ridge_error
[1] 0.005175233

$model_mulco$lasso_error
[1] 0.0003404091
```

```
simulation_results_20
```

```
$model_co
$model_co$lm_error
[1] 0.01561983

$model_co$stepwise_error
[1] 0.01536407

$model_co$pca_error
[1] 0.03457293

$model_co$ridge_error
[1] 0.01677616

$model_co$lasso_error
[1] 0.01547813


$model_mulco
$model_mulco$lm_error
[1] 0.0001759014

$model_mulco$stepwise_error
[1] 0.0001842865
```

```
$model_mulco$pca_error
[1] 0.04048754

$model_mulco$ridge_error
[1] 0.003181603

$model_mulco$lasso_error
[1] 0.000615791
```

```
simulation_results_40
```

```
$model_co
$model_co$lm_error
[1] 0.007585618

$model_co$stepwise_error
[1] 0.008286027

$model_co$pca_error
[1] 0.03752964

$model_co$ridge_error
[1] 0.01119759

$model_co$lasso_error
[1] 0.008431453


$model_mulco
$model_mulco$lm_error
[1] 7.61861e-05

$model_mulco$stepwise_error
[1] 6.896133e-05

$model_mulco$pca_error
[1] 0.01966983

$model_mulco$ridge_error
[1] 0.001568552

$model_mulco$lasso_error
```

```
[1] 0.0004007357
```

```
simulation_results_80
```

```
$model_co
$model_co$lm_error
[1] 0.005772981

$model_co$stepwise_error
[1] 0.00566902

$model_co$pca_error
[1] 0.04053034

$model_co$ridge_error
[1] 0.007139609

$model_co$lasso_error
[1] 0.005704319


$model_mulco
$model_mulco$lm_error
[1] 0.0001359246

$model_mulco$stepwise_error
[1] 0.0001359246

$model_mulco$pca_error
[1] 0.01489648

$model_mulco$ridge_error
[1] 0.001408046

$model_mulco$lasso_error
[1] 0.0003926706
```

```
simulation_results_10_l
```

```
$model_co
$model_co$lm_error
```

```
[1] 9.182781e-05

$model_co$stepwise_error
[1] 0.0001286884

$model_co$pca_error
[1] 0.030378

$model_co$ridge_error
[1] 0.001278453

$model_co$lasso_error
[1] 0.0004001677


$model_mulco
$model_mulco$lm_error
[1] 0.0003857417

$model_mulco$stepwise_error
[1] 0.0004430776

$model_mulco$pca_error
[1] 1.591302

$model_mulco$ridge_error
[1] 0.002463018

$model_mulco$lasso_error
[1] 0.0005619589
```

```
simulation_results_20_l
```

```
$model_co
$model_co$lm_error
[1] 0.0001345653

$model_co$stepwise_error
[1] 0.0001358162

$model_co$pca_error
[1] 0.08599611
```

```
$model_co$ridge_error
[1] 0.001497698

$model_co$lasso_error
[1] 0.0004302167


$model_mulco
$model_mulco$lm_error
[1] 0.000129234

$model_mulco$stepwise_error
[1] 0.0001263228

$model_mulco$pca_error
[1] 0.3260658

$model_mulco$ridge_error
[1] 0.001483704

$model_mulco$lasso_error
[1] 0.0003638239
```

```
simulation_results_40_l
```

```
$model_co
$model_co$lm_error
[1] 0.0001587497

$model_co$stepwise_error
[1] 0.0001587497

$model_co$pca_error
[1] 0.01925318

$model_co$ridge_error
[1] 0.00223085

$model_co$lasso_error
[1] 0.000447206
```

```
$model_mulco
$model_mulco$lm_error
[1] 0.0001659282

$model_mulco$stepwise_error
[1] 0.0001659282

$model_mulco$pca_error
[1] 0.09030931

$model_mulco$ridge_error
[1] 0.001325038

$model_mulco$lasso_error
[1] 0.0004495081
```

```
simulation_results_80_l
```

```
$model_co
$model_co$lm_error
[1] 8.652018e-05

$model_co$stepwise_error
[1] 8.652018e-05

$model_co$pca_error
[1] 0.06507261

$model_co$ridge_error
[1] 0.002382886

$model_co$lasso_error
[1] 0.0005170709


$model_mulco
$model_mulco$lm_error
[1] 0.0001080691

$model_mulco$stepwise_error
[1] 0.0001080691
```

```
$model_mulco$pca_error
[1] 0.01093376

$model_mulco$ridge_error
[1] 0.001239442

$model_mulco$lasso_error
[1] 0.0003514962
```