Praca maigsterksa dzienniczek

Jakub Liu

2024-10-10

Testing the R^2 change

Required library

```r
library(faux)
```

```
##
## ************
## Welcome to faux. For support and examples visit:
## https://debruine.github.io/faux/
## - Get and set global package options with: faux_options()
## ************
```

Functions

```r
R2 <- function(y_true, y_pred){
  mu_y_true <- mean(y_true)
  RSS <- 0
  TSS <- 0
  for(i in 1:length(y_true)){
    RSS <- RSS + (y_true[i] - y_pred[i])^2
    TSS <- TSS + (y_true[i] - mu_y_true)^2
  }
  R2 <- 1 - RSS/TSS
  return(R2)

}

# for simple linear regression
final_model <- function(x){
  y <- x*beta1_from_dist + beta0_from_dist
  return(y)
}
```

Simple linear regression

```r
library(faux)

testing_data <- rnorm_multi(
                      n = 10000,
                      mu = c(10,30),
                      sd = c(50,50),
                      r = c(0.9),
                      varnames = c("X", "Y")
)



model_orig <- lm(Y~X, data = testing_data)

n_boot_samples <- 1000
n_repeats <- 10
R_sq <- 1:n_repeats

for(rep in 1:n_repeats){

  beta0_est <- 1:n_boot_samples
  beta1_est <- 1:n_boot_samples

  for(i in 1:n_boot_samples){
    boot_sample <- testing_data[sample(nrow(testing_data),
                                       size = nrow(testing_data),
                                       replace = TRUE), ]
    model <- lm(Y~X, data = boot_sample)
    beta0_est[i] <- summary(model)$coefficients[1]
    beta1_est[i] <- summary(model)$coefficients[2]
  }


  mean_beta0 <- mean(beta0_est)
  sd_beta0 <- sd(beta0_est)
  mean_beta1 <- mean(beta1_est)
  sd_beta1 <- sd(beta1_est)

  beta0_from_dist <- rnorm(n = 1, mean = mean_beta0, sd = sd_beta0)
  beta1_from_dist <- rnorm(n = 1, mean = mean_beta1, sd = sd_beta1)

  y_hat <- final_model(testing_data$X)
  r2 <- R2(testing_data$Y, y_hat)
  R_sq[rep] <- r2

}
```

```r
print(c("R2 original model: ", summary(model_orig)$r.squared))
```

```
## [1] "R2 original model: " "0.809243250546198"
```

```r
print(c("R2 final model (mean of 10 repetitions): ", mean(R_sq)))
```

```
## [1] "R2 final model (mean of 10 repetitions): "
## [2] "0.809211330625201"
```

The R2 for the final model is very slightly lower than for the original model. This is good, but the difference is extremely low. Play around with the difference in the means (between the predictor and the response) and the variances in the simulated data to see what effect this has on the R2 value.

higher difference in means (eqaul variances)

```r
testing_data <- rnorm_multi(
                      n = 10000,
                      mu = c(1,100),
                      sd = c(50,50),
                      r = c(0.9),
                      varnames = c("X", "Y")
)

model_orig <- lm(Y~X, data = testing_data)

n_boot_samples <- 100
n_repeats <- 10
R_sq <- 1:n_repeats

for(rep in 1:n_repeats){

  beta0_est <- 1:n_boot_samples
  beta1_est <- 1:n_boot_samples

  for(i in 1:n_boot_samples){
    boot_sample <- testing_data[sample(nrow(testing_data),
                                       size = nrow(testing_data),
                                       replace = TRUE), ]
    model <- lm(Y~X, data = boot_sample)
    beta0_est[i] <- summary(model)$coefficients[1]
    beta1_est[i] <- summary(model)$coefficients[2]
  }


  mean_beta0 <- mean(beta0_est)
  sd_beta0 <- sd(beta0_est)
  mean_beta1 <- mean(beta1_est)
  sd_beta1 <- sd(beta1_est)

  beta0_from_dist <- rnorm(n = 1, mean = mean_beta0, sd = sd_beta0)
```

```r
  beta1_from_dist <- rnorm(n = 1, mean = mean_beta1, sd = sd_beta1)

  y_hat <- final_model(testing_data$X)
  r2 <- R2(testing_data$Y, y_hat)
  R_sq[rep] <- r2

}

print(c("R2 original model: ", summary(model_orig)$r.squared))
```

```
## [1] "R2 original model: " "0.802635658673099"
```

```r
print(c("R2 final model (mean of 10 repetitions): ", mean(R_sq)))
```

```
## [1] "R2 final model (mean of 10 repetitions): "
## [2] "0.802596724955854"
```

Smaller difference in means

```r
testing_data <- rnorm_multi(
                      n = 10000,
                      mu = c(1,2),
                      sd = c(50,50),
                      r = c(0.9),
                      varnames = c("X", "Y")
)

model_orig <- lm(Y~X, data = testing_data)

n_boot_samples <- 100
n_repeats <- 10
R_sq <- 1:n_repeats

for(rep in 1:n_repeats){

  beta0_est <- 1:n_boot_samples
  beta1_est <- 1:n_boot_samples

  for(i in 1:n_boot_samples){
    boot_sample <- testing_data[sample(nrow(testing_data),
                                       size = nrow(testing_data),
                                       replace = TRUE), ]
    model <- lm(Y~X, data = boot_sample)
    beta0_est[i] <- summary(model)$coefficients[1]
    beta1_est[i] <- summary(model)$coefficients[2]
  }


  mean_beta0 <- mean(beta0_est)
  sd_beta0 <- sd(beta0_est)
```

```r
  mean_beta1 <- mean(beta1_est)
  sd_beta1 <- sd(beta1_est)

  beta0_from_dist <- rnorm(n = 1, mean = mean_beta0, sd = sd_beta0)
  beta1_from_dist <- rnorm(n = 1, mean = mean_beta1, sd = sd_beta1)

  y_hat <- final_model(testing_data$X)
  r2 <- R2(testing_data$Y, y_hat)
  R_sq[rep] <- r2

}

print(c("R2 original model: ", summary(model_orig)$r.squared))
```

```
## [1] "R2 original model: " "0.809966190052755"
```

```r
print(c("R2 final model (mean of 10 repetitions): ", mean(R_sq)))
```

```
## [1] "R2 final model (mean of 10 repetitions): "
## [2] "0.809926542216354"
```

The difference in means does not seem to make a difference. Now try a weaker correlation between the dependent and independent variable.

```r
testing_data <- rnorm_multi(
                      n = 10000,
                      mu = c(10,20),
                      sd = c(50,50),
                      r = c(0.2),
                      varnames = c("X", "Y")
)

model_orig <- lm(Y~X, data = testing_data)

n_boot_samples <- 100
n_repeats <- 10
R_sq <- 1:n_repeats

for(rep in 1:n_repeats){

  beta0_est <- 1:n_boot_samples
  beta1_est <- 1:n_boot_samples

  for(i in 1:n_boot_samples){
    boot_sample <- testing_data[sample(nrow(testing_data),
                                        size = nrow(testing_data),
                                        replace = TRUE), ]
    model <- lm(Y~X, data = boot_sample)
    beta0_est[i] <- summary(model)$coefficients[1]
    beta1_est[i] <- summary(model)$coefficients[2]
  }
```

```r
  mean_beta0 <- mean(beta0_est)
  sd_beta0 <- sd(beta0_est)
  mean_beta1 <- mean(beta1_est)
  sd_beta1 <- sd(beta1_est)

  beta0_from_dist <- rnorm(n = 1, mean = mean_beta0, sd = sd_beta0)
  beta1_from_dist <- rnorm(n = 1, mean = mean_beta1, sd = sd_beta1)

  y_hat <- final_model(testing_data$X)
  r2 <- R2(testing_data$Y, y_hat)
  R_sq[rep] <- r2

}

print(c("R2 original model: ", summary(model_orig)$r.squared))
```

```
## [1] "R2 original model: " "0.0391134681261378"
```

```r
print(c("R2 final model (mean of 10 repetitions): ", mean(R_sq)))
```

```
## [1] "R2 final model (mean of 10 repetitions): "
## [2] "0.0389364016078087"
```

The reduction in R2 is slightly bigger, but still not much.

Bigger difference in variances (medium correlation)

```r
testing_data <- rnorm_multi(
                    n = 10000,
                    mu = c(10,20),
                    sd = c(5,500),
                    r = c(0.7),
                    varnames = c("X", "Y")
)

model_orig <- lm(Y~X, data = testing_data)

n_boot_samples <- 100
n_repeats <- 10
R_sq <- 1:n_repeats

for(rep in 1:n_repeats){

  beta0_est <- 1:n_boot_samples
  beta1_est <- 1:n_boot_samples

  for(i in 1:n_boot_samples){
    boot_sample <- testing_data[sample(nrow(testing_data),
                                       size = nrow(testing_data),
```

```r
                                        replace = TRUE), ]
    model <- lm(Y~X, data = boot_sample)
    beta0_est[i] <- summary(model)$coefficients[1]
    beta1_est[i] <- summary(model)$coefficients[2]
  }


  mean_beta0 <- mean(beta0_est)
  sd_beta0 <- sd(beta0_est)
  mean_beta1 <- mean(beta1_est)
  sd_beta1 <- sd(beta1_est)

  beta0_from_dist <- rnorm(n = 1, mean = mean_beta0, sd = sd_beta0)
  beta1_from_dist <- rnorm(n = 1, mean = mean_beta1, sd = sd_beta1)

  y_hat <- final_model(testing_data$X)
  r2 <- R2(testing_data$Y, y_hat)
  R_sq[rep] <- r2

}

print(c("R2 original model: ", summary(model_orig)$r.squared))
```

```
## [1] "R2 original model: " "0.505265146348157"
```

```r
print(c("R2 final model (mean of 10 repetitions): ", mean(R_sq)))
```

```
## [1] "R2 final model (mean of 10 repetitions): "
## [2] "0.50507474138389"
```

As can be seen, the difference in variances does not impact the reduction in R2. Now try to make two datasets, one with high correlation, one with low correlation, merge them, shuffle them, and then see the reduction in R2.

Two datasets with different variances

```r
# dataset with high correlation between y and x
data_high_cor <- rnorm_multi(
                      n = 10000,
                      mu = c(10,20),
                      sd = c(5,7),
                      r = c(0.9),
                      varnames = c("X", "Y")
)

# dataset with low correlation between y and x
data_low_cor <- rnorm_multi(
                      n = 10000,
                      mu = c(10,20),
                      sd = c(5,7),
```

```r
                      r = c(0.2),
                      varnames = c("X", "Y")
)


data_full <- rbind(data_high_cor, data_low_cor)
data_full <- data_full[sample(nrow(data_full)), ]   # shuffle the rows of the dataset


model_orig <- lm(Y~X, data = data_full)

n_boot_samples <- 100
n_repeats <- 10
R_sq <- 1:n_repeats

for(rep in 1:n_repeats){

  beta0_est <- 1:n_boot_samples
  beta1_est <- 1:n_boot_samples

  for(i in 1:n_boot_samples){
    boot_sample <- data_full[sample(nrow(data_full),
                                    size = nrow(data_full),
                                    replace = TRUE), ]
    model <- lm(Y~X, data = boot_sample)
    beta0_est[i] <- summary(model)$coefficients[1]
    beta1_est[i] <- summary(model)$coefficients[2]
  }


  mean_beta0 <- mean(beta0_est)
  sd_beta0 <- sd(beta0_est)
  mean_beta1 <- mean(beta1_est)
  sd_beta1 <- sd(beta1_est)

  beta0_from_dist <- rnorm(n = 1, mean = mean_beta0, sd = sd_beta0)
  beta1_from_dist <- rnorm(n = 1, mean = mean_beta1, sd = sd_beta1)

  y_hat <- final_model(data_full$X)
  r2 <- R2(data_full$Y, y_hat)
  R_sq[rep] <- r2

}

print(c("R2 original model: ", summary(model_orig)$r.squared))
```

```
## [1] "R2 original model: " "0.288674973322847"
```

```r
print(c("R2 final model (mean of 10 repetitions): ", mean(R_sq)))
```

```
## [1] "R2 final model (mean of 10 repetitions): "
## [2] "0.287874432744399"
```

There does not seem to be a difference in R2 reduction, when we use this approach, compared to the previous approaches.