DSA 8020 R Session 3: Multiple Linear Regression II

Whitney

January 22, 2022

Contents

Species diversity on the Galapagos Islands

Load the data

```
library(faraway)
data(gala)
galaNew <- gala[, -2] # removing "Endemics"
```

General Linear F-Test

```
## First example
# Reduce Moddel
M1 <- lm(Species ~ Elevation, data = galaNew)
summary(M1)
```

```
##
## Call:
## lm(formula = Species ~ Elevation, data = galaNew)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -218.319  -30.721  -14.690    4.634  259.180
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.33511   19.20529   0.590     0.56
## Elevation    0.20079    0.03465   5.795 3.18e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 78.66 on 28 degrees of freedom
```

```
## Multiple R-squared:  0.5454, Adjusted R-squared:  0.5291
## F-statistic: 33.59 on 1 and 28 DF,  p-value: 3.177e-06
```

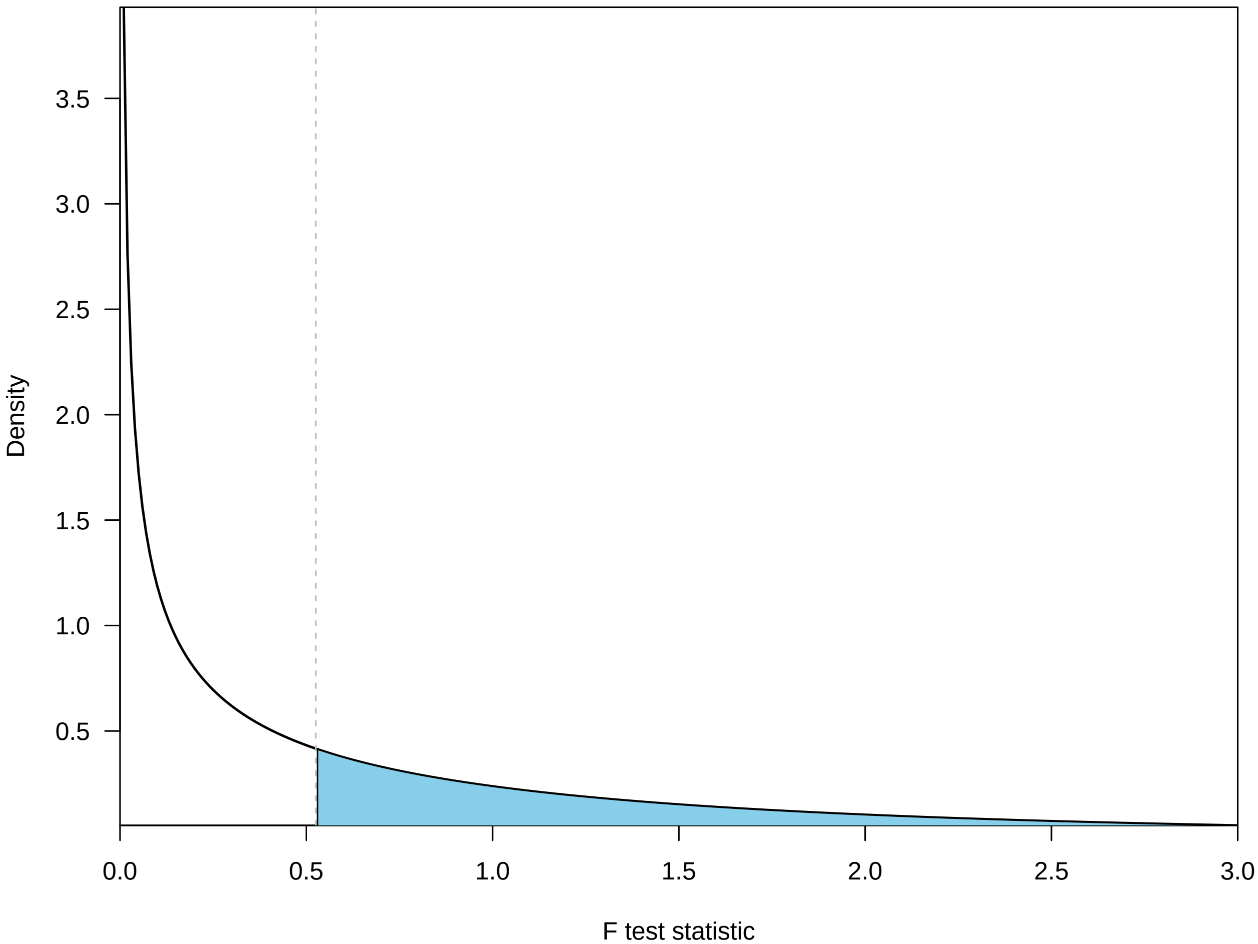
```r
# "Full" Model
M2 <- lm(Species ~ Elevation + Area, data = galaNew)
summary(M2)
```

```
##
## Call:
## lm(formula = Species ~ Elevation + Area, data = galaNew)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -192.619  -33.534  -19.199    7.541  261.514
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.10519   20.94211   0.817  0.42120
## Elevation    0.17174    0.05317   3.230  0.00325 **
## Area         0.01880    0.02594   0.725  0.47478
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79.34 on 27 degrees of freedom
## Multiple R-squared:  0.554,  Adjusted R-squared:  0.521
## F-statistic: 16.77 on 2 and 27 DF,  p-value: 1.843e-05
```

```r
## General Linear F-Test
anova(M1, M2)
```

```
## Analysis of Variance Table
##
## Model 1: Species ~ Elevation
## Model 2: Species ~ Elevation + Area
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     28 173254
## 2     27 169947  1      3307 0.5254 0.4748
```

```r
# p-value
par(las = 1, mar = c(4.1, 4.1, 1.1, 1.1))
xg <- seq(0, 3, 0.01); yg <- df(xg, 1, 27)
plot(xg, yg, type = "l", xaxs = "i", yaxs = "i", lwd = 1.6,
     xlab = "F test statistic", ylab = "Density")
abline(v = 0.5254, lty = 2, col = "gray")
polygon(c(xg[xg > 0.5254], rev(xg[xg > 0.5254])),
        c(yg[xg > 0.5254], rep(0, length(yg[xg > 0.5254]))),
        col = "skyblue")
```




```r
# Another example
Full <- lm(Species ~ ., data = galaNew)
Reduce <- lm(Species ~ Elevation + Adjacent, data = galaNew)
## General Linear F-Test
anova(Reduce, Full)
```

```
## Analysis of Variance Table
##
## Model 1: Species ~ Elevation + Adjacent
## Model 2: Species ~ Area + Elevation + Nearest + Scruz + Adjacent
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     27 100003
## 2     24  89231  3     10772 0.9657  0.425
```

Prediction

```r
data(fat)
lmod <- lm(brozek ~ age + weight + height + neck + chest + abdom + hip + thigh + knee
           + ankle + biceps + forearm + wrist, data = fat)

X <- model.matrix(lmod)
(x0 <- apply(X, 2, median))
```

```
## (Intercept)         age      weight      height        neck       chest
##        1.00       43.00      176.50       70.00       38.00       99.65
##       abdom         hip       thigh        knee       ankle      biceps
##       90.95       99.30       59.00       38.50       22.80       32.05
##     forearm       wrist
##       28.70       18.30
```

```
(y0 <- sum(x0 * coef(lmod)))
```

```
## [1] 17.49322
```

```
predict(lmod, new = data.frame(t(x0)))
```

```
##        1
## 17.49322
```

```
predict(lmod, new = data.frame(t(x0)), interval = "prediction")
```

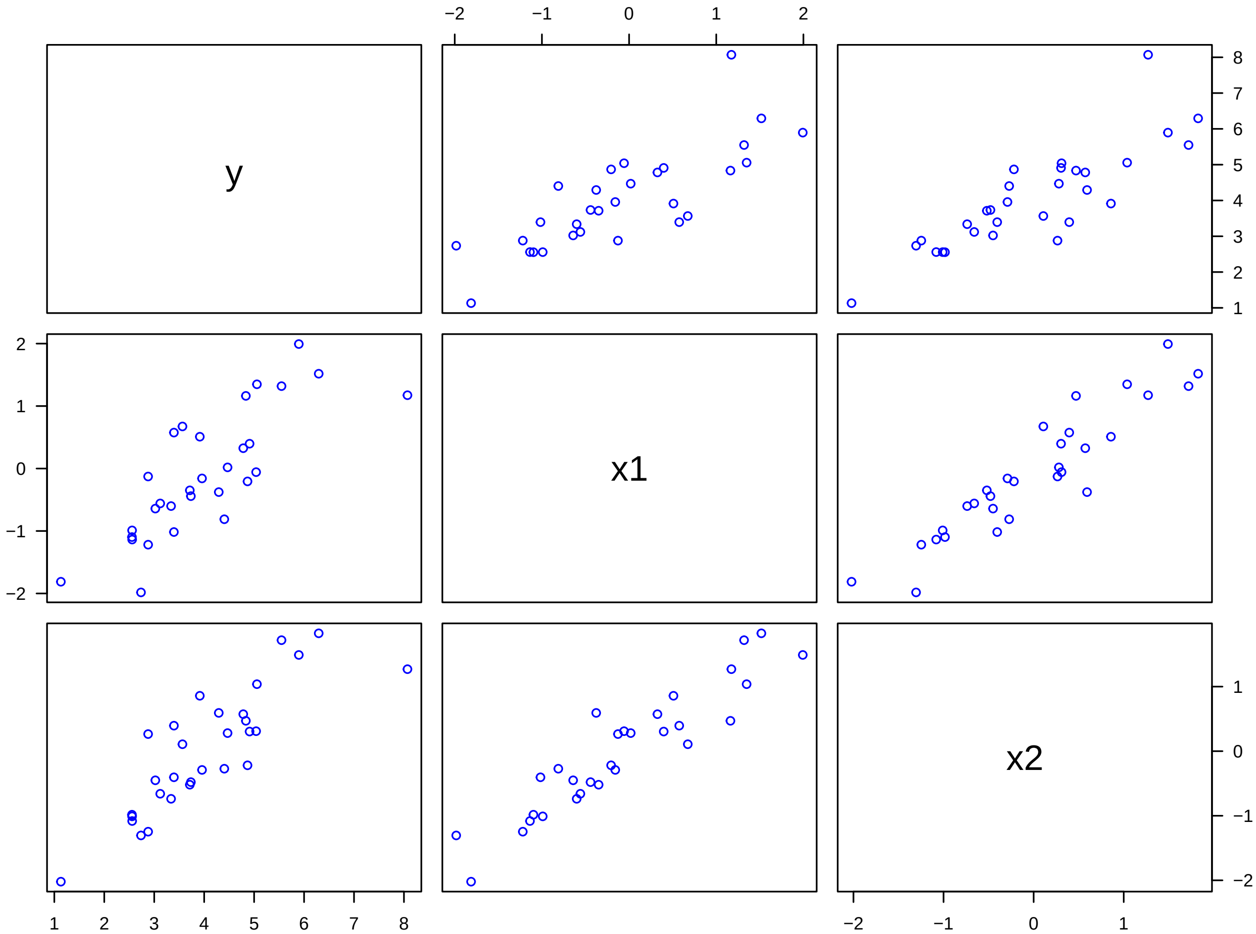
```
##        fit     lwr      upr
## 1 17.49322 9.61783 25.36861
```

```
predict(lmod, new = data.frame(t(x0)), interval = "confidence", alpha = 0.)
```

```
##        fit      lwr      upr
## 1 17.49322 16.94426 18.04219
```

Multicollinearity

```
set.seed(123)
N = 500
library(MASS)
x <- replicate(N, mvrnorm(n = 30, c(0, 0), matrix(c(1, 0.9, 0.9, 1), 2)))
y <- array(dim = c(30, N))
for (i in 1:N){
  y[, i] = 4 + 0.8 * x[, 1, i] + 0.6 * x[, 2, i] + rnorm(30)
}
# Grab the first simulated data
sim1 <- data.frame(y = y[, 1], x1 = x[, 1, 1], x2 = x[, 2, 1])
# Make the scatterplot matrix
pairs(sim1, las = 1, col = "blue")
```




```
# Compute the correlation matrix
cor(sim1)
```

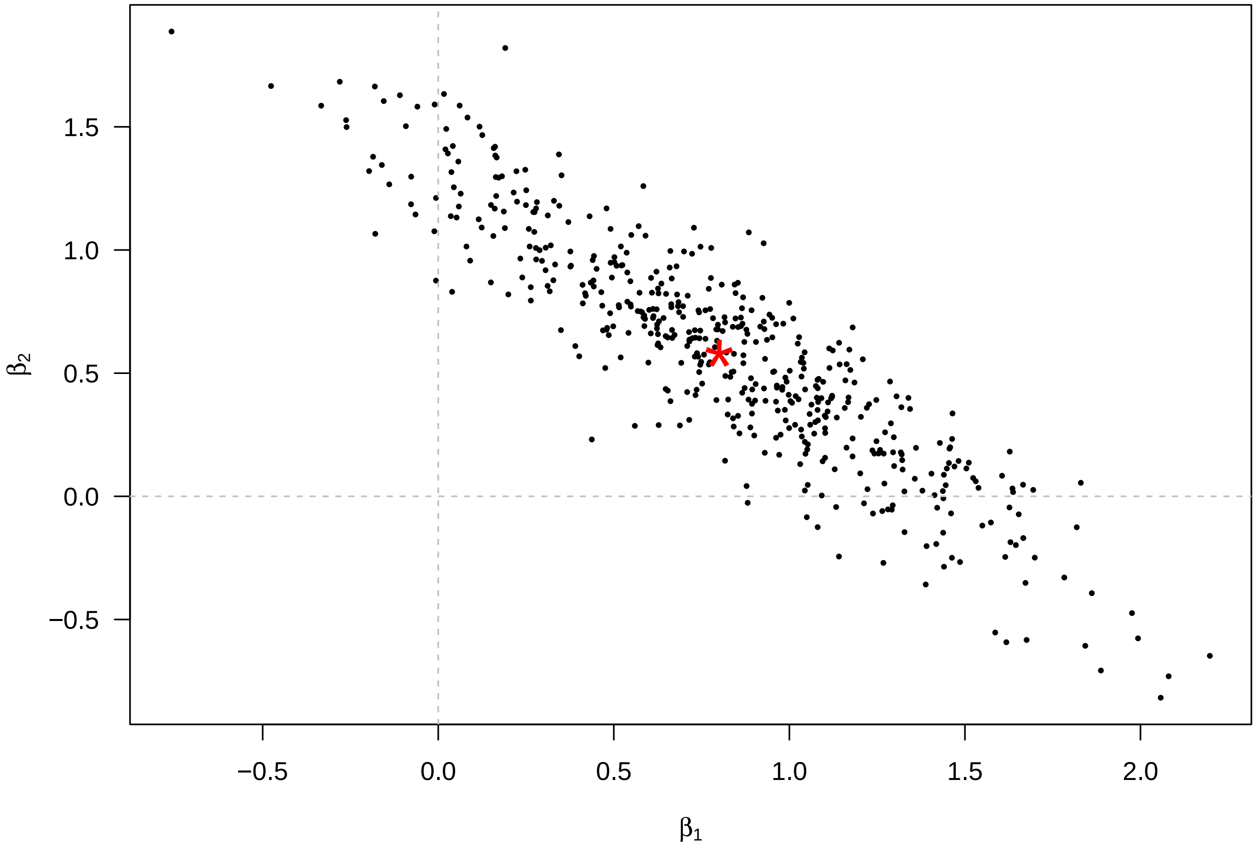
```
##            y        x1        x2
## y  1.0000000 0.7987777 0.8481084
## x1 0.7987777 1.0000000 0.9281514
## x2 0.8481084 0.9281514 1.0000000
```

```
# Save the fitted regression coefficients
beta <- array(dim = c(3, N))
for (i in 1:N){
  beta[, i] <- lm(y[, i] ~ x[, 1, i] + x[, 2, i])$coefficients
}

R.sq_M1 <- numeric(N)
for (i in 1:N){
  R.sq_M1[i] <- summary(lm(y[, i] ~ x[, 1, i] + x[, 2, i]))$r.squared
}

summary(R.sq_M1)
```

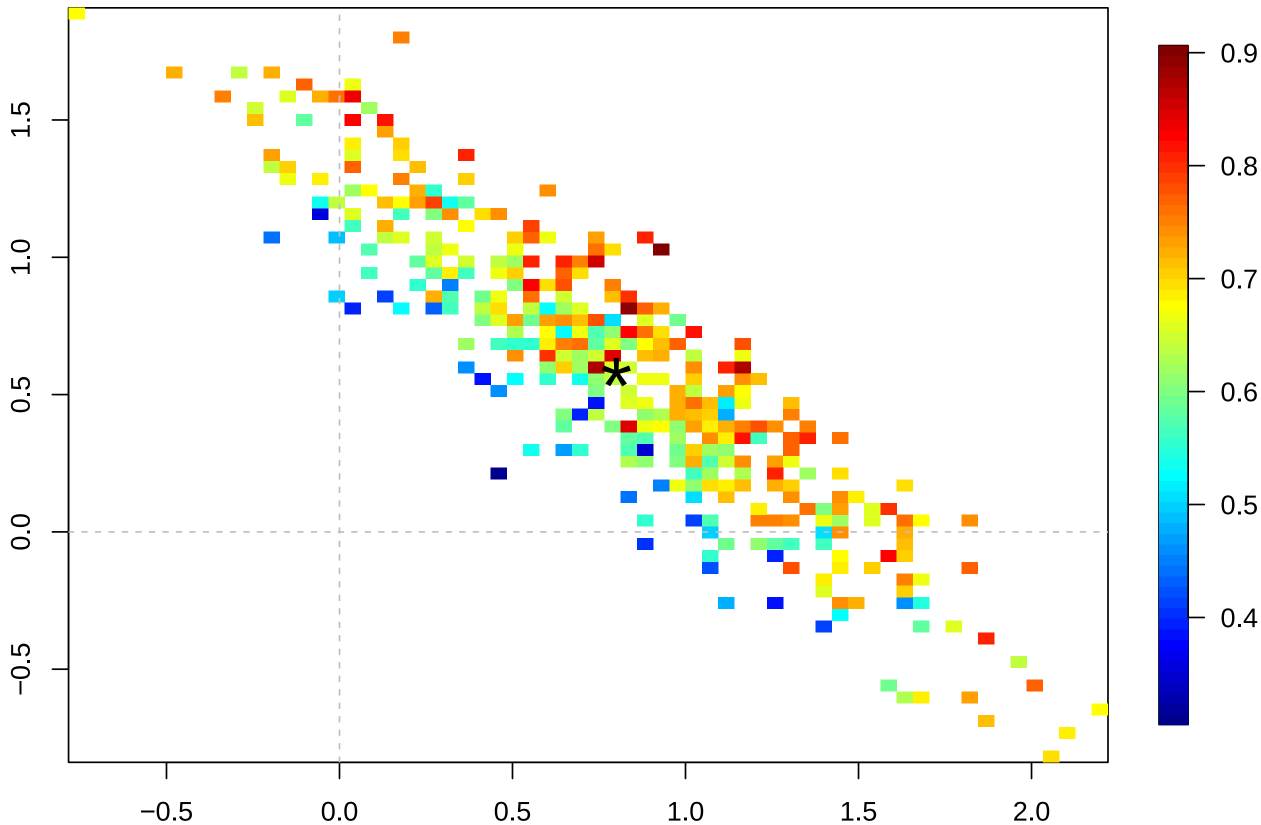
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3099  0.6049  0.6776  0.6630  0.7343  0.9016
```

```
plot(beta[2,], beta[3,], pch = 16, cex = 0.5,
     xlab = expression(beta[1]),
```

```
     ylab = expression(beta[2]), las = 1)
points(0.8, 0.6, pch = "*", cex = 3, col = "red")
abline(h = 0, lty = 2, col = "gray")
abline(v = 0, lty = 2, col = "gray")
```



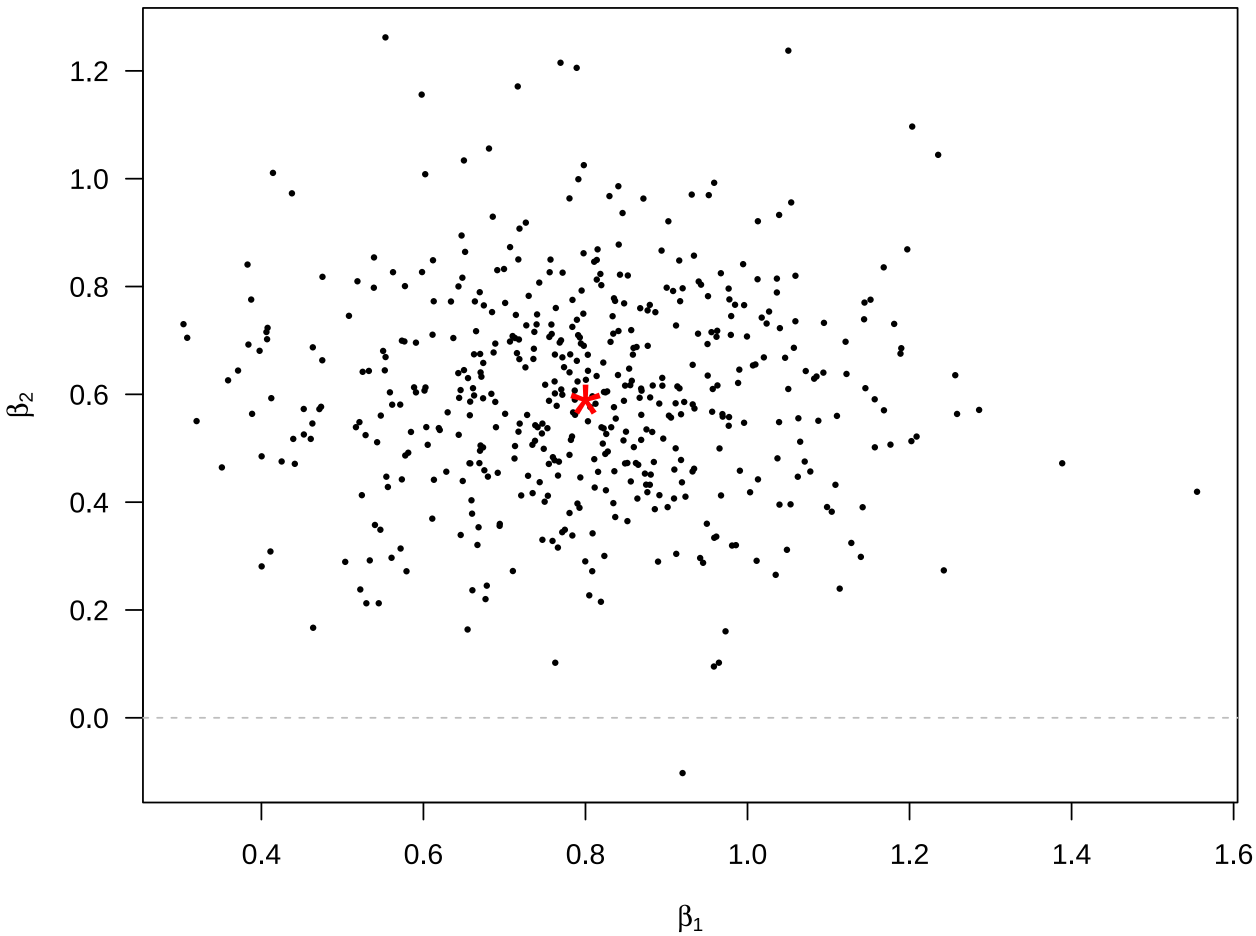

```
library(fields)
quilt.plot(beta[2,], beta[3, ], R.sq_M1)
points(0.8, 0.6, pch = "*", cex = 3)
abline(h = 0, lty = 2, col = "gray")
abline(v = 0, lty = 2, col = "gray")
```




```
# Compute the VIF
vif(sim1[, 2:3])
```

```
##       x1       x2
## 7.218394 7.218394
```

```
## Another simulation where the predictors are indepdent to each other
x1 <- replicate(N, mvrnorm(n = 30, c(0, 0), matrix(c(1, 0, 0, 1), 2)))
y1 <- array(dim = c(30, N))
for (i in 1:N){
  y1[, i] = 4 + 0.8 * x1[, 1, i] + 0.6 * x1[, 2, i] + rnorm(30)
}
beta1 <- array(dim = c(3, N))
for (i in 1:N){
  beta1[, i] <- lm(y1[, i] ~ x1[, 1, i] + x1[, 2, i])$coefficients
}

plot(beta1[2,], beta1[3,], pch = 16, cex = 0.5,
     xlab = expression(beta[1]),
     ylab = expression(beta[2]), las = 1)
points(0.8, 0.6, pch = "*", cex = 3, col = "red")
abline(h = 0, lty = 2, col = "gray")
abline(v = 0, lty = 2, col = "gray")
```




```
R.sq_M2 <- numeric(N)
for (i in 1:N){
  R.sq_M2[i] <- summary(lm(y1[, i] ~ x1[, 1, i] + x1[, 2, i]))$r.squared
}
summary(R.sq_M2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1179  0.4375  0.5325  0.5181  0.6062  0.8419
```

```
# Compute the VIF
vif(x1[, 1:2, 1])
```

```
## [1] 1.042404 1.042404
```