

Entrega 1

Bruno Olivera

10/5/2019

Practico 2: Regresión lineal múltiple

Ejercicio 1

```
set.seed(2019)

n <- 1000

x1 <- sort(runif(n))
x2 <- sort(runif(n))
x3 <- sort(runif(n))
y <- 3 + 2*x1 - 2*x2 + x3 + rnorm(n, sd = 1)
```

a)

```
# generamos la matriz de datos
data <- cbind(x1,x2,x3,y)
```

b)

```
x0 <- rep(1, n)
Xdata <- cbind(x1,x2,x3)
X <- cbind(x0,Xdata)

# estimamos los parámetros Beta
Beta <- solve((t(X)%*%X))%*%t(X)%*%y
```

Beta

```
##           [,1]
## Beta_0  2.937449
## Beta_1  5.768744
## Beta_2 -7.647101
## Beta_3  2.947553
```

```
# damos intervalos de confianza para las estimaciones
model <- lm(y ~ Xdata)
intervals <- confint(model)
```

intervals

```
##           2.5 %    97.5 %
## Beta_0  2.7623170  3.112581
## Beta_1 -0.3034077 11.840895
## Beta_2 -17.1610434  1.866841
## Beta_3 -2.7473957  8.642503
```

c)

```

# estimamos los parámetros Beta y calculamos sus varianzas
# para cada valor de tau
results <- matrix(ncol=8, nrow=5)
VIFs <- NULL
B_estimates <- NULL
tau <- c(0, .01, .1, 1, 10)

for(i in 1:5){
  x2_new <- x1 + rnorm(n, mean=0, sd=tau[i])
  Xdata_new <- cbind(x1,x2_new,x3)
  X_new <- cbind(x0,Xdata_new)
  if(tau[i] != 0) {
    B_estimates <- solve((t(X_new)%*%X_new))%*%t(X_new)%*%y
  }
  model_new <- lm(y ~ Xdata_new)
  summary_model <- summary(model_new)

  # para el caso de tau = 0 tenemos que sacar los estimadores del
  # modelo porque la matriz (X'X) no es invertible
  if(tau[i] == 0) {
    B_estimates[1] <- summary_model$coefficients[,1][1]
    B_estimates[2] <- summary_model$coefficients[,1][2]
    B_estimates[3] <- NA
    B_estimates[4] <- summary_model$coefficients[,1][3]
  }

  # calculamos la variación de los Betas mediante el cuadrado
  # del Std. Error devuelto por el modelo
  var_B0_1 = (summary_model$coefficients[,2]**2)[1]
  var_B1_1 = (summary_model$coefficients[,2]**2)[2]
  if(tau[i] != 0){
    var_B2_1 = (summary_model$coefficients[,2]**2)[3]
    var_B3_1 = (summary_model$coefficients[,2]**2)[4]
  }else{
    var_B2_1 = NA
    var_B3_1 = (summary_model$coefficients[,2]**2)[3]
  }

  if(tau[i] != 0) {
    # aproximamos la variación del modelo y calculamos la matriz var*inv(X'X)
    var_model_new = (sum((model_new$residuals)**2)/(n-4))*solve((t(X_new)%*%X_new))

    # calculamos la variación de los beta como la diagonal de la matriz anterior
    var_B0_2 = var_model_new[1,1]
    var_B1_2 = var_model_new[2,2]
    var_B2_2 = var_model_new[3,3]
    var_B3_2 = var_model_new[4,4]

    # controlamos que sean iguales ambas formas de calcular las varianzas
    assertthat::are_equal(var_B0_1,var_B0_2)
    assertthat::are_equal(var_B2_1,var_B1_2)
    assertthat::are_equal(var_B3_1,var_B2_2)
    assertthat::are_equal(var_B3_1,var_B3_2)
  }
}

```

```

}

newRow.data <- c(B_estimates[1],
                 B_estimates[2],
                 B_estimates[3],
                 B_estimates[4],
                 var_B0_1,
                 var_B1_1,
                 var_B2_1,
                 var_B3_1)

results[i,] = newRow.data

# calculamos el VIF y lo guardamos para la parte d)
VIFs <- rbind(VIFs,vif_calc(data.frame(Xdata_new)))
}

```

```

# resultados
results

```

```

##      Beta_0  Beta_1      Beta_2      Beta_3 Var_Beta_0 Var_Beta_1
##  0 2.967224 1.720953          NA -0.7073982 0.007619875  2.993489
## .01 2.967013 2.033804 -0.3145432347 -0.7056087 0.007631956 12.865869
## .1 2.970767 1.317669  0.4361185652 -0.7396950 0.007619568  3.076212
##  1 2.967271 1.737741 -0.0305103726 -0.6954549 0.007620695  2.994126
## 10 2.966250 1.697204 -0.0008137791 -0.6819989 0.007641328  3.004789
##      Var_Beta_2 Var_Beta_3
##  0          NA  3.057236
## .01 9.976438e+00  3.060598
## .1 9.991198e-02  3.055016
##  1 1.042324e-03  3.057724
## 10 9.973817e-06  3.069818

```

d)

```

# calculamos los VIFs
VIFs

```

```

##      x1      x2_new      x3
##  0      Inf      Inf 242.3330
## .01 1040.5008 808.209870 242.3586
## .1  249.2554  9.060742 242.3765
##  1  242.3585  1.074097 242.3457
## 10  243.0200  1.005339 243.1024

```

Ejercicio 2