

# Estadística no Paramétrica: Proyecto 2

## Comparación de métodos de remuestreo en el contexto de regresión.

Paula Rodríguez Díaz y Felipe González Casabianca

24 de mayo de 2017

### Resumen

Este documento cuenta como entregable para el proyecto 2 del curso: *Estadística no Paramétrica y remuestreo*, dictado en el primer semestre del año 2017 por Adolfo Quiroz en la Universidad de los Andes.

## 1. Introducción

En el contexto de una regresión:

$$Y \approx \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

A partir de una muestra, se quiere obtener un conjunto elíptico de confianza para el par de parámetros  $(\beta_2, \beta_3)$ . Para esto, se compararán distintos esquemas de remuestreo, errores y tamaños de muestras como se verá a continuación.

### 1.1. Distancia Mahalanobis

Para la construcción del conjunto elíptico de confianza se utiliza la distancia de Mahalanobis. Dicha distancia, mide la distancia de un elemento de una muestra a su media, teniendo como escala la desviación estandar de la misma. Formalmente se define como:

**Definición 1. Distancia de Mahalanobis** Dada una muestra  $X_n \in \mathbb{R}^N$  en dimensión  $N$  con media  $\mu \in \mathbb{R}^N$  y matriz de covarianza  $\Sigma$ , la Distancia de Mahalanobis de la observación  $x \in X_n$  es:

$$D_M(x) = \sqrt{(x - \mu)^T \Sigma^{-1} (x - \mu)}$$

## 1.2. Elipsoide de Mínimo Volumen

El estimador de Elipsoide de Mínimo Volumen (EMV) es un estimador robusto para el centro y matriz de covarianza de una muestra dada. Geométricamente se tiene que el EMV encuentra la elipsoide de mínimo volumen que cubre a todos los puntos de la muestra.

Para este caso, consideramos el conjunto

$$E(\bar{x}, S) = \{x \in \mathbb{R}^N : (x - \bar{x})^T S^{-1} (x - \bar{x}) \leq N\}$$

para  $\bar{x} \in \mathbb{R}^N$  y  $S \in M_N(\mathbb{R})$  simétrica y positiva, dado por la Distancia de Mahalanobis, que define una elipse en  $\mathbb{R}^N$ . Si escribimos  $S^{-1} = LL^T$  y  $z = L^{-1}(x - \bar{x})$  tenemos que

$$E(\bar{x}, S) = \{x = \bar{x} + Lz : \|z\| \leq \sqrt{N}\}$$

Con esto se tiene entonces que el volumen de la elipse  $E(\bar{x}, S)$  es:

$$\text{vol}(E(\bar{x}, S)) = |\det L| \cdot (\sqrt{N})^N \cdot \text{vol}(B_N(z)) = \frac{\text{const}(N)}{\sqrt{\det H}}$$

Se quiere entonces estrimar los valores de centro y matriz de covarianza tales que minimicen el volumen de  $E(\bar{x}, S)$  lo cual equivale a minimizar  $\log \det H$ . Por lo tanto se tiene el siguiente problema de optimización convexa:

$$\min_{\Sigma} \log \det H$$

*Sujeto a:*

$$x^T H x \leq N \quad \forall x \in X_n$$

Para este caso el cálculo del elipsoide de volumen mínimo se hará con el comando `CovMde` del paquete `rrcov` del lenguaje R.

## 1.3. Errores Heterocedásticos y Homocedásticos

En los experimentos contemplados en este proyecto, se incluirán errores en de estos tipos en las mediciones. Procedemos con una breve explicación de cada uno:

- **Homocedásticos:** hace referencia a que los errores tienen una varianza constante a través de las distintas observaciones del problema. En nuestro caso particular, las observaciones tendrán un error con distribución Normal con media y varianza constante para cada entrada de la regresión.
- **Heterocedásticos:** antagónico al término anterior, este tipo de error quiere decir que los errores de las distintas observaciones tienen distintas varianzas. Para nuestro contexto de regresión, los errores heterocedásticos fueron incluidos con una distribución exponencial.

## 1.4. Esquemas de remuestreo

Recordamos que en nuestro contexto se tiene una matriz de diseño  $X$  y un vector  $Y$  de  $n$  observaciones y se quiere encontrar un conjunto minimal donde se encuentre el vector  $\beta$  tal que  $X\beta \approx Y$ . Con esto en mente, se realizaron dos esquemas de remuestreo:

- **Remuestreo por pares:** En este tipo de remuestreo se realiza un bootstrap no paramétrico clásico, escogiendo  $n$  muestras con remplazo de la matriz extendida  $X|Y$ , obteniendo un nuevo escenario de regresión  $X^*|Y^*$ . A dicho escenario, se realiza una aproximación por minimo cuadrados para obtener el  $\beta^*$  correspondiente a esa iteración. Este procedimiento se realiza  $B$  veces.
- **Remuestreo por ajuste de residuos:** La idea de este tipo de remuestreo es hacer un resmuestreo de los residuos del modelo por medio Bootstrapping. Es decir, ya teniendo el modelo

$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \hat{\beta}_2 x_i^2 + \hat{\beta}_3 x_i^3 + \epsilon_i$$

se crean los datos  $y_i = \hat{\beta}_0 + \hat{\beta}_1 x_i + \hat{\beta}_2 x_i^2 + \hat{\beta}_3 x_i^3 + \epsilon_i^*$  donde  $(\epsilon_i^*)$  son obtenien por remuestreo bootstrap de  $(\epsilon_i)$ .

## 1.5. Remuestreo suavizado

Otro esquema que se quiere evaluar en este proyecto es el remuestro suavizado. La idea es, en lugar de remuestrear elementos de la muestra como tal, escoger elementos suficientemente cerca a los originales, tales que se conserve la distribución subyacente en la muestra pero evitando reconsruir muestras completamente discretas.

Para una muestra  $X$  de dimensión  $p$ , el remuestreo suavizado realizado consiste en lo siguiente:

- Se realiza un remuestreo no parametrico clásico de la muestra  $X$  obteniendo una muestra del mismo tamaño  $X_t^*$ .
- Se calcula la desviación estandar  $h$  para suavizar la muestra, segun los resultados de Bowman and Foster [1993], asi:

$$h = \frac{4}{(2+p)n}^{\frac{1}{4+p}}$$

- Para cada elemento de  $x \in X_t^*$ , se asocia un nuevo elemento tomado aleatorio con distribución Normal con media igual a  $x$  y matriz de covarianzas:  $hI$ .
- Estos nuevos elementos constituyen el conjunto remuestreado final  $X^*$

Para el remuestreo por ajuste de residuos, se suavizaron el conjunto de residuos remuestreados y para el remuestreo por pares, se suavizaron tanto la matriz de diseño remuestreada como el vector de observaciones remuestreado.

## 2. Detalles Implementación

Con los anteriores conceptos en mente, se realizaron los siguientes experimentos:

Para cada una de las posibles combinaciones de los siguientes parámetros, se realizaron 200 iteraciones y se anotó el promedio de volumen de la elipse, el promedio de las distancias de Mahalanobis de  $(\beta_2, \beta_3)$  con respecto a a cada elipse resultante y el porcentaje de veces que dicho par se encontró dentro del conjunto propuesto. Los parámetros a variar son siguientes:

- **Tamaño de la muestra:** 90 o 270.
- **Tipo de remuestreo:** Por pares o ajuste de residuos.
- **Remuestreo suavizado:** si o no.
- **Tipo de error:** Heterocedástico o homocedástico.
- **Varianza en el error:** 0.5 o 2

Para cada combinación de parámetros, se hizo un remuestreo de 500 elementos. Estas combinaciones nos arrojan un total de 32 escenarios distintos a probar.

La extracción de la elipse de mínimo tamaño se hace a través de la librería *rrcov*. Dicha librería, permite la extracción de la matriz de covarianza, distancia Mahalanobis y centro de una elipse para un porcentaje de datos multivariados suministrados (en este caso se incluye 95 % de los datos). El código completo se encuentra en la sección de anexos y se encuentra público en el link:

[https://github.com/minigonche/no\\_parametrica\\_proj\\_2](https://github.com/minigonche/no_parametrica_proj_2)

## 3. Resultados

A continuación se incluye una tabla con los resultados obtenidos:

	Tam. Muestra	Tipo Error	Suavizado	Desv. Estan	Tipo Remuestreo	Dist. Mahalanobis	% Betas dentro Elipse	Volumen Elipse
12	90	Homosedastico	FALSE	0.5	Residuos	0.06099038	1.0000000	7.546633e-06
28	270	Homosedastico	FALSE	0.5	Residuos	0.06136116	1.0000000	7.624125e-06
10	90	Homosedastico	FALSE	0.5	Parejas	0.06173928	1.0000000	7.788086e-06
26	270	Homosedastico	FALSE	0.5	Parejas	0.06331633	1.0000000	8.164496e-06
6	90	Heterosedastico	FALSE	2.0	Parejas	0.07017176	1.0000000	1.082798e-05
4	90	Heterosedastico	FALSE	0.5	Residuos	0.07192373	1.0000000	1.124356e-05
18	270	Heterosedastico	FALSE	0.5	Parejas	0.06981003	1.0000000	1.127243e-05
22	270	Heterosedastico	FALSE	2.0	Parejas	0.07278392	1.0000000	1.138209e-05
8	90	Heterosedastico	FALSE	2.0	Residuos	0.07383478	1.0000000	1.161531e-05
24	270	Heterosedastico	FALSE	2.0	Residuos	0.07498349	1.0000000	1.182098e-05
2	90	Heterosedastico	FALSE	0.5	Parejas	0.07273358	1.0000000	1.197269e-05
20	270	Heterosedastico	FALSE	0.5	Residuos	0.07439158	1.0000000	1.209563e-05
27	270	Homosedastico	TRUE	0.5	Residuos	0.10753261	1.0000000	2.291238e-05
11	90	Homosedastico	TRUE	0.5	Residuos	0.10714729	1.0000000	2.307158e-05
7	90	Heterosedastico	TRUE	2.0	Residuos	0.11707098	1.0000000	2.846141e-05
3	90	Heterosedastico	TRUE	0.5	Residuos	0.11827877	1.0000000	2.865349e-05
19	270	Heterosedastico	TRUE	0.5	Residuos	0.12039283	1.0000000	2.954820e-05
23	270	Heterosedastico	TRUE	2.0	Residuos	0.12240121	1.0000000	3.003136e-05
32	270	Homosedastico	FALSE	2.0	Residuos	0.96900832	0.6110448	1.937526e-03
16	90	Homosedastico	FALSE	2.0	Residuos	0.98578023	0.5683582	1.969412e-03
15	90	Homosedastico	TRUE	2.0	Residuos	1.01448071	0.5536318	2.102989e-03
30	270	Homosedastico	FALSE	2.0	Parejas	1.01395752	0.5417413	2.059845e-03
14	90	Homosedastico	FALSE	2.0	Parejas	1.00428342	0.5196517	2.044296e-03
31	270	Homosedastico	TRUE	2.0	Residuos	1.02386837	0.4445274	2.122792e-03
25	270	Homosedastico	TRUE	0.5	Parejas	1.42228747	0.0000000	1.717591e-03
9	90	Homosedastico	TRUE	0.5	Parejas	1.42335457	0.0000000	1.723862e-03
17	270	Heterosedastico	TRUE	0.5	Parejas	1.44673847	0.0000000	1.753400e-03
1	90	Heterosedastico	TRUE	0.5	Parejas	1.43925547	0.0000000	1.754096e-03
21	270	Heterosedastico	TRUE	2.0	Parejas	1.43395600	0.0000000	1.757508e-03
5	90	Heterosedastico	TRUE	2.0	Parejas	1.45115273	0.0000000	1.764146e-03
13	90	Homosedastico	TRUE	2.0	Parejas	2.17318964	0.0000000	3.968895e-03
29	270	Homosedastico	TRUE	2.0	Parejas	2.16733667	0.0000000	3.987515e-03

Figura 1: Tabla con resultados de los 32 experimentos, ordenados según el porcentaje de betas dentro de la elipse trazada

Cabe mencionar algunas generalidades sobre los experimntos:

- El peor rendimiento, 0 % de parámetros dentro de la elipse, se obtiene a través del remuestreo suavizado por parejas. Independiente del tipo de error, desviación estándar o tamaño de muestra este método arroja los peores resultados. Sin embargo, el remuestre por parejas no suavizado tiene excelentes resultados (100 % en con tipo de error heterocedastico y 50 % en el caso homocedastico). Esto lleva a pensar que el problema está en la técnica de suavizado utilizada. Cabe recordar que la forma de suavizar el remuestreo por parejas suaviza cada observación remuestreada de la matriz de diseño, incluyendo la columna de unos. Es posible que suavizar una columna con varianza cero (ya que cada entrada correspondiente a la columna del primer parametro es 1) con una normal de varianza distinta a cero no mejore los resultados. Recordamos que esta columna fue suavizada con una muestra con media 1 y desviación estándar:

$$h = \left( \frac{4}{6n} \right)^{\frac{1}{6}}$$

- Los mejores resultados correponden a las muestras con error homocedástico con remuestreo no suavizado y varianza en el error igual a 0.5 (para ambos tamaños y esquemas de remuestreo). El volumen de esta combinación de parámetros arroja la elipse de menor volumen: alrededor de  $7 \times 10^{-6}$ , donde los otros escenarios constuyen elipses con volúmenes a partir de  $1,5 \times 10^{-7}$
- La inclusión de una desviación estándar mayor en los errores homocedásticos tiene un impacto similar para ambos esquemas de remuestreo no suavizado. En ambos esquemas, el paso de una desviación estándar de 0.5 a 2, implica una reducción del 50 % en la cantidad de veces que la elipse contine el conjunto de parámetros.

## 4. Conclusiones

Para el caso de varianza pequeña (0.5) ambos esquemas de remuestreo (no suavizados) son óptimos, arrojando parámetros 100 % de las veces dentro de la elipse, con los menores volúmenes de todos los experimentos. Sin embargo, el reto para ambos esquemas sucede con errores homocedasticos de varianza alta (2), en este caso el remuestreo por ajuste residuos es una mejor alternativa, arrojando un porcentaje de 60 % dentro de la elipse trazada versus el 50 % correspondiente al remuestreo por parejas.

Por otro lado, el esquema de suavizado seleccionado no presenta ninguna mejora adicional. En el caso del remuestreo por ajuste de residuos, no aporta un porcentaje mayor que el esquema clásico y como ya se mencionó, en el caso del remuestreo por pares suavizar los datos hace que se pierda completamente la eficiencia del procedimiento. Además, suavizar el remuestreo incurre en un gasto

computacional adicional de generar elementos aleatorios para cada elemento en cada iteración del remuestreo.

## 5. Anexos

Se incluye por completitud el código desarrollado:

```

1  # Code to compare different resampling methods in the context of regression
2  # Written by Paula Rodriguez y Felipe Gonzalez
3
4  #IMPORTS
5  library('rrcov') #for the minimum ellipsoid volume
6  library('MASS') #Multivariate normal
7  library('gridExtra') #Export tables
8  library('grid')
9  library('gtable')
10
11 # Function that returns the design matrix. Includes the constant value (
12   column of ones) until the
13 # cubic parameter (beta3)
14 # This code is based on the script received as part of the project's
15   instruction
16 # PARAMETER: n (integer) the row size of the design matrix
17 # RETURNS: The design matrix of size n x 4
18 get_designMatrix = function(n)
19 {
20   n=90
21   unos=rep(1,n)
22   x1=seq(0,10,length=n)
23   x2=rep(c(-1,-.5,0,.5,1),length=n)
24   tmp=seq(0,5,length=10)
25   x3=rep(tmp,length=n)
26   X=cbind(unos,x1,x2,x3)
27   return(X)
28 }
29
30 # Function that returns a fixed value for the betas
31 # RETURNS: Vector of size 4, with the beta values
32 getFixedBetas = function()
33 {
34   beta0=sqrt(2)
35   beta1=1
36   beta2=-2
37   beta3=pi
38   beta=c(beta0,beta1,beta2,beta3)
39   return(beta)
40 }
41
42 # Function that gives the least square approximation solution of the system
43   Xx = b.
44 # PARAMETER: X ( n x m matrix) design matrix.
45   b ( n x 1 vector) target or solution vector
46 # RETURNS: A ( m x m matrix) corresponds to the inverse of (X^t)(X)
47 # RETURNS: An m vector to the least approximation solution of the mentioned
48   system
49 # Recall that the least square approximation corresponds to coefficients the
50   projection
51 # of b onto the column space of X, that can be obtained solving (X^t)(X)x =
52   (X^t)b
53 # This code is based on the script received as part of the project's
54   instruction
55 estibeta=function(b,X,A = NULL){
56   if(is.null(A))
57   {

```

```

52     A=solve(t(X) %*% X)
53     print('calculating inverse: (X^t)(X)')
54 }
55
56 tmp=A %*% t(X)
57 return(tmp %*% b)
58 }
59
60
61 estiY=function(Y,X,A){
62     n = nrow(X)
63     betahat=estibeta(Y,X,A)
64     Yhat=X %*% betahat
65     resi=Y-Yhat
66     tmp=X %*% A
67     tmp=tmp %*% t(X)
68     iden=diag(nrow=n)
69     H=iden-tmp
70     h=diag(H)
71     resiadj=resi/sqrt(h)
72     return(cbind(Yhat,resi,sqrt(h),resiadj))
73 }
74
75
76 #Function that constructs the target vector with homoscedastic error given
77 # the design matrix
78 # and the beta values
79 # PARAMETER: X ( n x m matrix) design matrix.
80 #             : beta ( m vector) vector with the values for betas in the design
81 #             matrix
82 #             : sigma (number > 0) corresponds to the standard deviation of the
83 #             normal sample
84 # RETURNS: an n vector corresponding to (X)beta + error
85 # This code is based on the script received as part of the project's
86 # instruction
87 getHomoscedasticError = function(X, beta, sigma = 0.5)
88 {
89     n = nrow(X)
90     Y=X %*% beta+rnorm(n, sd = sigma)
91     return(Y)
92 }
93
94 #AUXILIARY FUNCTION
95 # Generates an heteroscedastic non gaussian exponential vecor error of size n
96 rdexp=function(n){
97     y=rexp(n)
98     u=runif(n)
99     y[u>0.5]=-y[u>0.5]
100     return(y)
101 } # fin de rdexp
102
103 #Function that constructs the target vector with heteroscedastic error given
104 # the design matrix
105 # and the beta values
106 # PARAMETER: X ( n x m matrix) design matrix.
107 #             : beta ( m vector) vector with the values for betas in the design
108 #             matrix
109 #             : sigma (number > 0) corresponds to the standard deviation of the
110 #             normal sample
111 # RETURNS: an n vector corresponding to (X)beta + error
112 # This code is based on the script received as part of the project's
113 # instruction
114 getHeteroscedasticError = function(X, beta, sigma = 0.5)
115 {
116     n = nrow(X)
117     Y = getHomoscedasticError(X,beta,sigma)
118     Y=X %*% beta+0.025*Y*rdexp(n)
119 }

```



```

112     return(Y)
113 }
114
115
116
117 # Function that executes the ADJUSTED RESIDUE RESAMPLING procedure for the
    given parameters.
118 # PARAMETER: X ( n x m matrix) design matrix.
119 #             Y ( n vector) target vector for the regression
120 #             B (integer) size of the reampling
121 #             smooth_sample (boolean) if the bootstrap sampling should be done
    smoothly or not
122 # RETURNS: n vector with star{beta} - hat{beta}
123 # Given a design matrix X and a target vector Y (probably with error), the
    procedure first calculates
124 # hat{beta} as the least square approximation of the system  $Xx = Y$ . Then does
    B adjusted residue
125 # resampling procedures obtaining star{X} and star{beta} as the solution of (
    star{X})x = Y. The procedure
126 # returns the B observations of the vectors: star{beta} - hat{beta}
127 # This code is based on the script received as part of the project's
    intruccion
128 adjResidueResampling = function(X,Y,B, smooth_sample = FALSE)
129 {
130
131     n = nrow(X)
132     p = 1
133     # Formula based on A.W. Boman et al.
134     hSmooth = (4/((2+p)*n))^(1/(4+p))
135
136     A = solve(t(X)%*%X)
137     betahat=estibeta(Y,X,A)
138     tmp=estiY(Y,X,A)
139     Yhat=tmp[,1];
140     resiadj=tmp[,2]
141     #output
142     refer=matrix(0,B,4)
143     for(b in 1:B){
144         if(smooth_sample){
145             resampledResidues = sample(resiadj,n,replace=T)
146             Ystar=X%*%betahat + rnorm(n, mean = resampledResidues, sd = hSmooth)
147         }else{
148             Ystar=X%*%betahat + sample(resiadj,n,replace=T)
149         }
150
151         betastar=estibeta(Ystar,X,A)
152         refer[b,]=betastar-betahat
153     }#end for
154     return(refer)
155 }
156
157
158 # Function that executes the PAIR RESAMPLING procedure for the given
    parameters.
159 # PARAMETER: X ( n x m matrix) design matrix.
160 #             Y ( n vector) target vector for the regression
161 #             B (integer) size of the reampling
162 #             smooth_sample (boolean) if the bootstrap sampling should be done
    smoothly or not
163 # RETURNS: n vector with star{beta} - hat{beta}
164 # Given a design matrix X and a target vector Y (probably with error), the
    procedure first calculates
165 # hat{beta} as the least square approximation of the system  $Xx = Y$ . Then does
    B pair
166 # resampling procedures obtaining star{X} and star{beta} as the solution of (
    star{X})x = Y. The procedure
167 # returns the B observations of the vectors: star{beta} - hat{beta}
168 # This code is based on the script received as part of the project's

```

```

169      intruccion
170 pairResampling = function(X,Y,B, smooth_sample = FALSE)
171 {
172     n = nrow(X)
173     m = ncol(X)
174
175     p = 1
176     # Formula based on: A.W. Boman et al.
177     hSmooth1 = (4/((2+p)*n))^(1/(4+p))
178
179     p = 4
180     # Formula based on: A.W. Boman et al.
181     hSmooth4 = (4/((2+p)*n))^(1/(4+p))
182
183     # Smoothing standard deviation
184     sd = hSmooth4*diag(rep(1,m))
185
186     A = solve(t(X)%*%X)
187     betahat=estibeta(Y,X,A)
188
189     indi=1:n
190     referpa=matrix(0,B,m)
191     for(b in 1:B){
192         indstar=sample(indi,n,replace=T)
193
194         X_resampled = X[indstar,]
195         Y_resampled = Y[indstar]
196         if(smooth_sample){
197             Xstar = t(apply(X_resampled,1,function(row) mvrnorm(n = 1,mu = row,
198                 Sigma = sd)))
199             Ystar = rnorm(n = n , mean = Y_resampled, sd = hSmooth1)
200         }else{
201             Xstar = X[indstar,]
202             Ystar = Y[indstar]
203         }
204
205         Astar=solve(t(Xstar)%*%Xstar)
206         betastar=estibeta(Ystar,Xstar,Astar)
207         referpa[b,]=betastar-betahat
208     } #end for
209     return(referpa)
210 }
211
212
213
214 # Function that executes the instructed experiments and stores its results
215 # in a data frame
216 executeExperiments = function(print_progress = TRUE, num_ite = 10, B =
217     10000)
218 {
219     #constants
220     pairs = 'Parejas'
221     residue = 'Residuos'
222     hetero = 'Heterosedastico'
223     homo = 'Homosedastico'
224
225     #Resulting data frame
226     result = data.frame(size = c(), typeOfError = c(), smooth = c(), devStandar
227         = c(), typeOfReesampling = c(), mahaDistance = c(),
228         betasInsideEllipse = c(), ellipseVolume = c() )
229
230     size = c(90,270)
231     devStandar = c(0.5,2)
232     typeOfError = c(hetero,homo)

```

```

231 smooth = c(TRUE, FALSE)
232 typeOfReesampling = c(pairs, residue)
233
234
235 variables = expand.grid(smooth, typeOfReesampling, devStandar, typeOfError,
236                          size)
237 colnames(variables) = c('smooth', 'typeOfReesampling', 'devStandar', '
238                          typeOfError', 'size')
239
240 for(i in 1:nrow(variables))
241 {
242     vol = c()
243     mahaDistance = c()
244     inside = c()
245     row = variables[i,]
246     for(j in 1:num_ite)
247     {
248         X = get_designMatrix(row$size)
249         A = solve(t(X) %*% X)
250         beta = getFixedBetas()
251         if(row$typeOfError == hetero){
252             Y = getHeteroscedasticError(X, beta, row$devStandar)
253         }else if(row$typeOfError == homo){
254             Y = getHomoscedasticError(X, beta, row$devStandar)
255         }else{
256             stop(paste('Type of error not supported: ', row$typeOfError))
257         }
258
259         if(row$typeOfReesampling == pairs){
260             resampledResponse = pairResampling(X = X, Y = Y, B = B, smooth_sample
261                                                 = row$smooth)
262         }else if(row$typeOfReesampling == residue){
263             resampledResponse = adjResidueResampling(X = X, Y = Y, B = B, smooth_
264                                                     sample = row$smooth)
265         }else{
266             stop(paste('Type of reesampling not supported: ', row$
267                       typeOfReesampling))
268         }
269
270         #Gets the ellipse
271         ellipse = CovMve(resampledResponse[,3:4], alpha=.95)
272         vol = c(vol, getDet(ellipse))
273         mahaDistance = c(mahaDistance, (beta[3:4] - getCenter(ellipse)) %*%
274                           getCov(ellipse) %*% (beta[3:4] - getCenter(ellipse)))
275         inside = c(inside, as.numeric(mahaDistance < 1))
276     }
277
278     temp = data.frame(mahaDistance = mean(mahaDistance), betasInsideEllipse =
279                       mean(inside), ellipseVolume = mean(vol))
280
281     #binds variables with results
282     final_row = cbind(row, temp)
283     #organizes the row
284     final_row = final_row[,c('size', 'typeOfError', 'smooth', 'devStandar', '
285                              typeOfReesampling', 'mahaDistance', 'betasInsideEllipse', '
286                              ellipseVolume')]
287     result = rbind(result, final_row)
288     if(print_progress)
289     {
290         print(paste('Finished:', toString(i), 'of', nrow(variables), sep = ' '))
291         print(final_row)
292     }

```

```
290 |  
291 | }  
292 |  
293 | return(result)  
294 |  
295 | }
```

## Referencias

AW Bowman and PJ Foster. Adaptive smoothing and density-based tests of multivariate normality. *Journal of the American Statistical Association*, 88 (422):529–537, 1993.