mpuntoscontrol: Ejemplo de uso en BBVA

Marcos Olguín Martínez y Samuel Rocha Guzmán

2018-08-01

En este documento se describirán las distintas funciones que tiene el paquete `mpuntoscontrol` para el desarrollo de modelos y parámetros de **Metodologías de Riesgo en BBVA Bancomer**. Es importante recordar que los puntos de control que se atenderán serán los siguientes:

Puntos de control para Modelos

1. Definicion de universo y variables iniciales
2. Analisis de segmentacion
3. Tratamiento de datos
4. Depuracion de los datos
5. Estimacion de los modelos 1ra parte
6. Estimacion de los modelos 2da parte
7. Métodos para situaciones especificas
8. Validación, ajustes y tarjeta de puntuación

Envíos para Parámetros

- Envio 1
    - Segmentación
    - Filtros y calidad de datos
    - Series de PD para AC
- Envío 2
    - Ejes y curvas de calibración
    - Resultados finales con AC
    - Documentación

Recordemos que este documento tiene fines didácticos de como utilizar el paquete `mpuntoscontrol` y por lo tanto se describirán algunas funcionalidades y requsitos que deberán cumplir las bases de datos que *Metodologías de Riesgo* para el correcto funcionamiento de las funciones.

Requisitos de las bases de datos

Existen ya procesos muy estandarizados sobre la construcción de las variables más relevantes que son utilizadas por Metodologías de Riesgo para el cálculo de modelos, y aquí se enlistarán algunas de ellas que facilitarán la experiencia del usuario en el uso de las funciones del paquete.

> **Marca de incumplimiento:** Normalmente se encuentran en las bases de datos como `incmpl` o `marca_01`, en cualquier caso, ésta variable deberá ser binaria, es decir, **0** para identificar a los clientes *buenos* y **1** para identificar a los clientes *malos*.

Cohorte: Esta variable hace referencia al periodo de observación del cliente (para determinar si es malo o no) debe venir con el formato de 201109 (año y mes) y debá ser de tipo numérica.

Variables categóricas: Son aquellas que no tienen más de 10 categorías.

Definiciones importantes

Supongamos que tenemos una base de datos con las siguientes características:

Variables
5 de tipo numéricas

3 de tipo categóricas

Incumplimiento
Variable binaria: 0 para buenos, 1 para malos.

Cohorte
De tipo numérica con el formato 201109.

Muestra
Se tiene segmentada la base en `train` y `test`

Creación de la base DUMMY

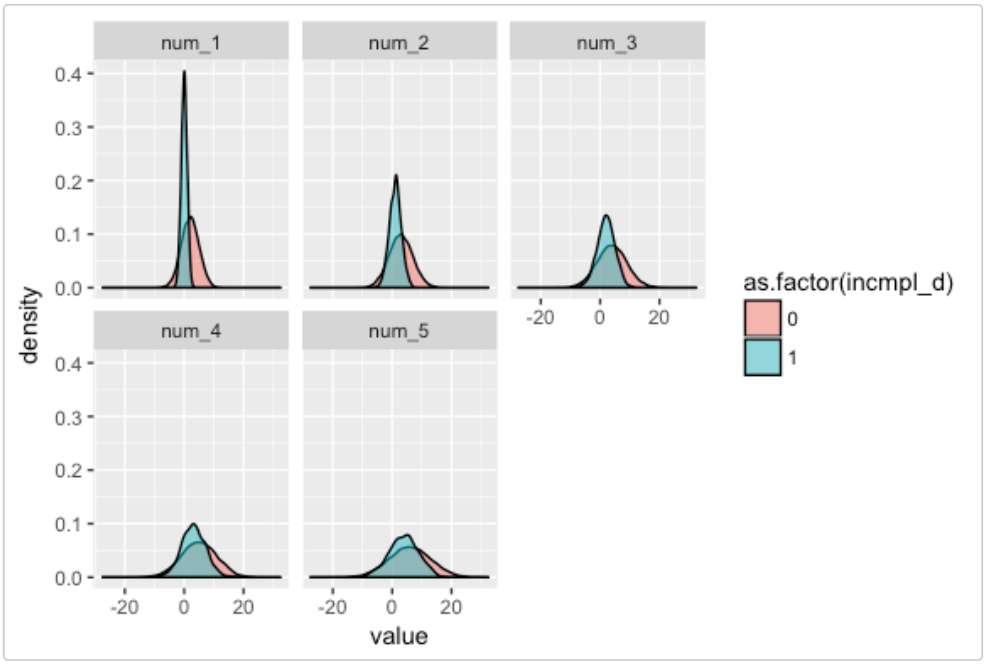
```
# Creamos la base de datos
library(dplyr)
library(ggplot2)

tot_tabla_dummy <- 20000

incmpl_d <- sample(0:1, size=tot_tabla_dummy, prob=c(0.9,0.1), replace=TRUE)
muestra <- sample(c("Train","Test"), size = tot_tabla_dummy, prob = c(0.7,0.3), replace = TRUE)
segmento <- sample(1:2, size = tot_tabla_dummy, prob = c(0.65,0.35), replace = TRUE)
cohorte <- sample(c(201501:201512,201601:201612), size = tot_tabla_dummy, replace = TRUE)
num_1 <- ifelse(incmpl_d==1, round(rnorm(tot_tabla_dummy, mean = 0, sd = 1),4),
                             round(rnorm(tot_tabla_dummy, mean = 2, sd = 3),4))
num_2 <- ifelse(incmpl_d==1, round(rnorm(tot_tabla_dummy, mean = 1, sd = 2),4),
                             round(rnorm(tot_tabla_dummy, mean = 3, sd = 4),4))
num_3 <- ifelse(incmpl_d==1, round(rnorm(tot_tabla_dummy, mean = 2, sd = 3),4),
                             round(rnorm(tot_tabla_dummy, mean = 4, sd = 5),4))
num_4 <- ifelse(incmpl_d==1, round(rnorm(tot_tabla_dummy, mean = 3, sd = 4),4),
                             round(rnorm(tot_tabla_dummy, mean = 5, sd = 6),4))
num_5 <- ifelse(incmpl_d==1, round(rnorm(tot_tabla_dummy, mean = 4, sd = 5),4),
                             round(rnorm(tot_tabla_dummy, mean = 6, sd = 7),4))
```

```
cat_1 <- sample(1:5, size=tot_tabla_dummy, prob=c(0.2,0.2,0.2,0.2,0.2), replace=TRUE)
cat_2 <- sample(1:4, size=tot_tabla_dummy, prob=c(0.2,0.3,0.4,0.1), replace=TRUE)
cat_3 <- sample(1:3, size=tot_tabla_dummy, prob=c(0.3,0.3,0.4), replace=TRUE)

d <- data.frame(incmpl_d = incmpl_d,
                muestra = muestra,
                segmento = segmento,
                cohorte = cohorte,
                num_1 = num_1,
                num_2 = num_2,
                num_3 = num_3,
                num_4 = num_4,
                num_5 = num_5,
                cat_1 = cat_1,
                cat_2 = cat_2,
                cat_3 = cat_3)

knitr::kable(head(d))
```

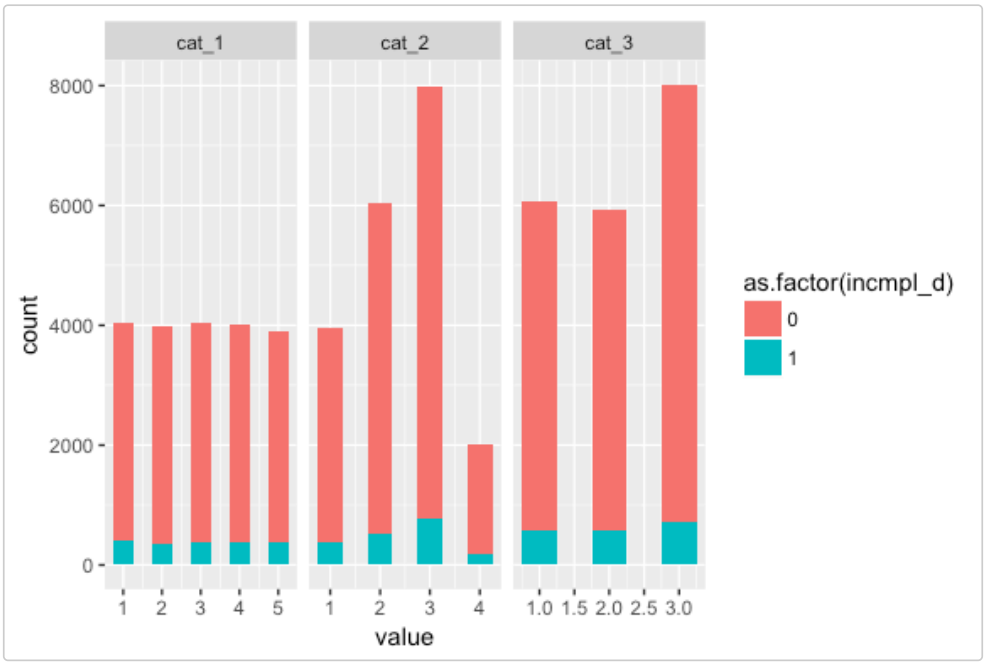
incmpl_d	muestra	segmento	cohorte	num_1	num_2	num_3	num_4	num_5	cat_1	cat_2	cat_3
0	Train	1	201604	2.7180	0.5208	5.3840	7.6037	-4.7077	2	2	3
0	Train	1	201611	0.0040	2.6765	7.2661	4.1856	-5.5760	3	3	3
0	Train	1	201501	3.9065	0.3317	-1.1915	7.2356	4.4104	4	3	2
0	Train	1	201508	-3.8692	10.2206	9.6464	3.3743	-6.0120	4	3	3
0	Train	1	201605	1.3250	0.8049	2.9837	-6.0989	-12.3784	3	2	3
0	Test	2	201502	-0.8288	0.4409	5.2165	7.8899	2.8018	1	1	1

Gráficas de las distribuciones e histogramas

```
# Grafica de las variables numéricas
library(tidyr)
d %>%
    select(incmpl_d, muestra, num_1, num_2, num_3, num_4, num_5) %>%
    gather(key, value, num_1:num_5) %>%
    ggplot(aes(value, fill=as.factor(incmpl_d))) +
        geom_density(alpha=0.5) +
        facet_wrap(~ key)
```




```
# Grafica de las variables categóricas
d %>%
    select(incmpl_d, muestra, cat_1, cat_2, cat_3) %>%
    gather(key, value, cat_1:cat_3) %>%
    ggplot(aes(value, fill = as.factor(incmpl_d))) +
        geom_histogram(binwidth = 0.5) +
        facet_wrap(~ key, scales = 'free_x')
```



Punto de Control 1

Los resultados del *punto de control 1* deben responder a las siguientes preguntas:

- ¿Cuál es el periodo muestral?
- ¿Cuál es la estructura inicial de la base de datos?

```
library(mpuntoscontrol)
start_time <- Sys.time()
w <- pto_control_1(base = d, cohorte = "cohorte", incumpl = "incmpl_d", colores = "gold2")

end_time <- Sys.time()
end_time - start_time
#> Time difference of 0.09470296 secs

names(w)
#> [1] "rango_cohortes" "dimension"      "periodo"        "tabla_tm"
#> [5] "graph_periodo"
```

Los resultados de todas las funciones de este paquete están almacenadas en listas de R, por lo que para acceder a la información es mediante su posición dentro de la lista o su nombre dentro de la lista. Por ejemplo:

```
w$rango_cohortes
#> [1] 201501 201612
w$dimension
#> [1] 20000    12
w$periodo
#> # A tibble: 2 x 3
#>    anio Total Proporcion
#>   <dbl> <int>      <dbl>
#> 1 2015. 10058      0.503
#> 2 2016.  9942      0.497
w$tabla_tm
#> # A tibble: 2 x 5
#>    anio Total Malos Buenos     TM
#>   <dbl> <int> <int>  <int>  <dbl>
#> 1 2015. 10058   965   9093 0.0959
#> 2 2016.  9942   921   9021 0.0926
w$graph_periodo
```

Hay otras preguntas que deben responderse, para este punto de control, sin embargo, están más relacionadas con definiciones del estúdio y definiciones de las variables, como por ejemplo: Definición precisa de variables relevantes, definición de la variable objetivo y tipología de la información.

Punto de Control 2

Este punto de control atiende:

- Análisis de segmentación
- Análisis de la distribución inicial y madurez
  - Distribución de la muestra
  - Filtros de madurez
- Selección de la muestra de modelización

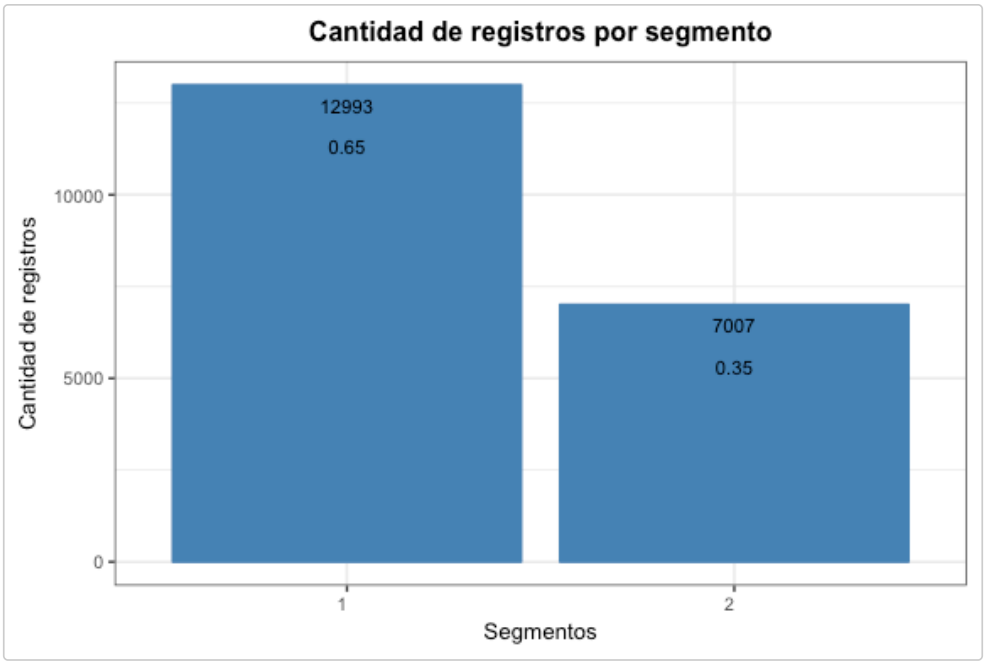
Parte A

```
# Como la variable "muestra" viene con valores
start_time <- Sys.time()
vars_numeric <- c("num_1","num_2","num_3","num_4","num_5")
w <- pto_control_2a(base = d, var_segmentadora = "segmento", vars_interes = vars_numeric,
                    incumpl = "incmpl_d")

end_time <- Sys.time()
end_time - start_time
#> Time difference of 0.5757229 secs

names(w)
```

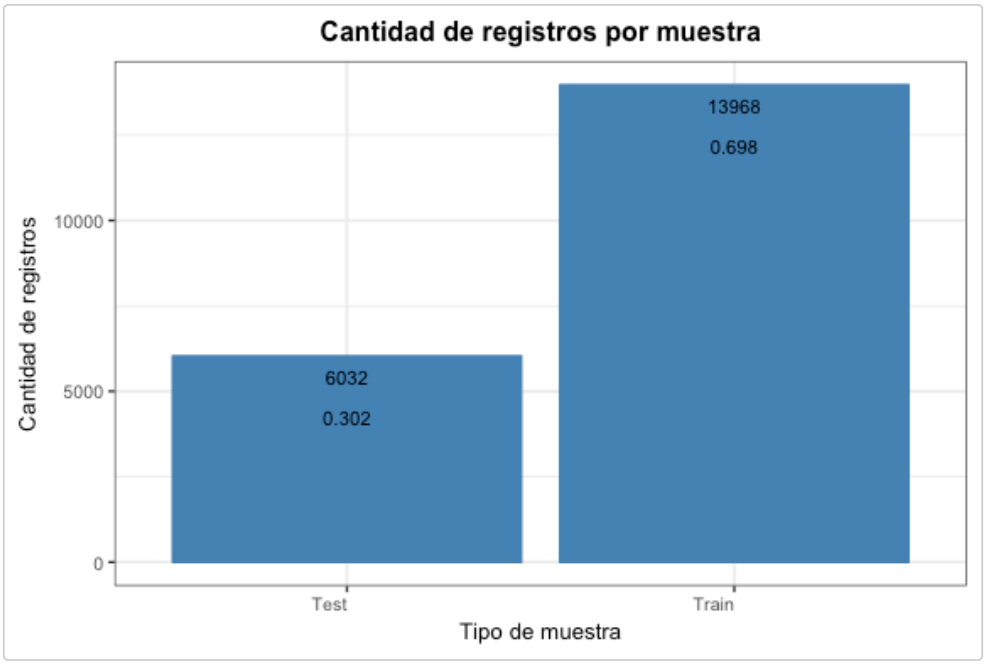
```
#> [1] "chow_test"    "allison_test"


# Resultados del Chow Test
w$chow_test
#> [[1]]
#> [[1]]$chow_test
#> [1] 0.7871908
#>
#> [[1]]$p_value
#> [1] 0.5797912

# Resultados del Allison Test
w$allison_test$mod_completo
#>              Estimate Pr(>|z|)
#> (Intercept) -0.996194        0
#> num_1       -0.265657        0
#> num_2       -0.139997        0
#> num_3       -0.089452        0
#> num_4       -0.055808        0
#> num_5       -0.050033        0
w$allison_test$mod_segmentos
#> [[1]]
#>              Estimate Pr(>|z|)
#> (Intercept) -0.940225 0.000000
#> num_1       -0.269097 0.000000
#> num_2       -0.135500 0.000000
#> num_3       -0.105981 0.000000
#> num_4       -0.062544 0.000000
#> num_5       -0.054595 0.000000
#> dummyseg_1  -0.081789 0.317719
#> num_1_dummy  0.004883 0.813702
#> num_2_dummy -0.006930 0.643780
#> num_3_dummy  0.024974 0.034510
#> num_4_dummy  0.009958 0.299690
#> num_5_dummy  0.006583 0.418626
#>
#> [[2]]
#>              Estimate Pr(>|z|)
#> (Intercept) -1.022014 0.000000
#> num_1       -0.264213 0.000000
#> num_2       -0.142430 0.000000
#> num_3       -0.081007 0.000000
#> num_4       -0.052587 0.000000
#> num_5       -0.048012 0.000000
#> dummyseg_2   0.081789 0.317719
#> num_1_dummy -0.004883 0.813702
#> num_2_dummy  0.006930 0.643780
#> num_3_dummy -0.024974 0.034510
#> num_4_dummy -0.009958 0.299690
#> num_5_dummy -0.006583 0.418626
```

Parte B

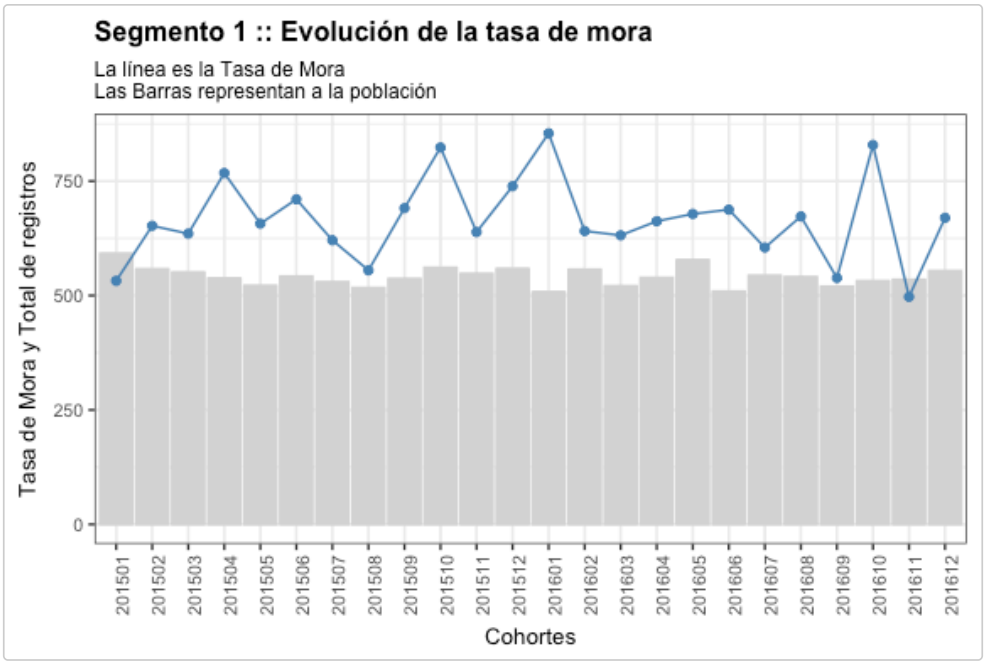
```
start_time <- Sys.time()
w <- pto_control_2b(base = d, segmentada = "s", var_segment = "segmento", muestra = "muestra",
                    cohorte = "cohorte", incumpl = "incmpl_d", color1 = "steelblue", color2 =
"azure3",
                    color3 = "gold", apoyo_tm = 7000)

end_time <- Sys.time()
end_time - start_time
#> Time difference of 0.08531594 secs

names(w)
#> [1] "distrib_seg"     "distrib_muestra" "distrib_coh"     "muestra_mod"
#> [5] "tipo_incumpl"    "incmpl_muestra"  "tmora_segm"



w$distrib_seg
#> $Tabla
#> # A tibble: 2 x 3
#>   segmento Total Proporcion
#>      <int> <int>      <dbl>
#> 1        1 12993      0.650
#> 2        2  7007      0.350
#>
#> $Grafica
```



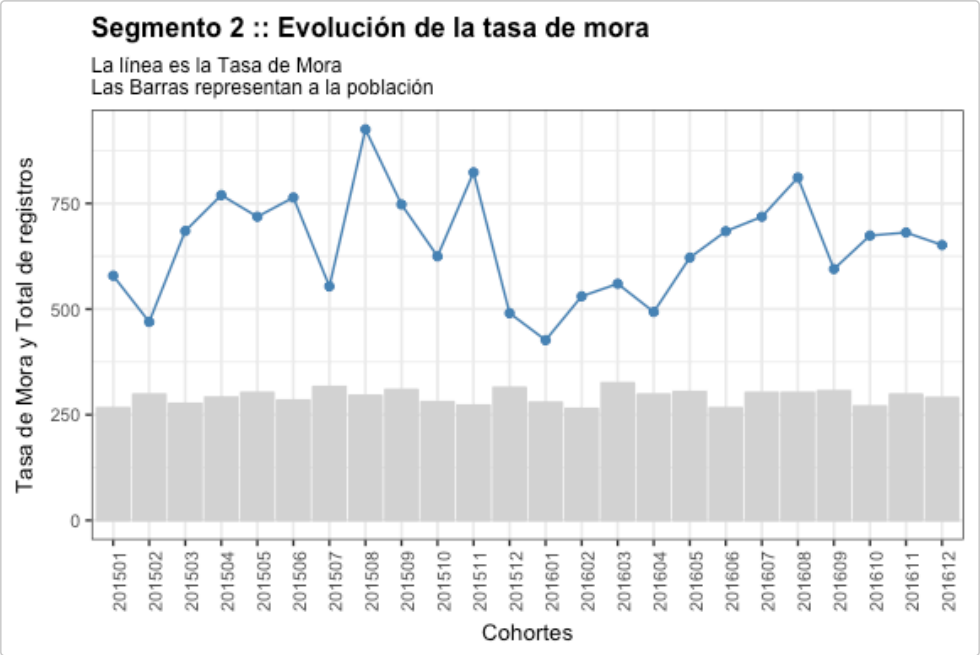

```
w$distrib_muestra
#> $Tabla
#> # A tibble: 2 x 3
#>   muestra Total Proporcion
#>   <fct>   <int>      <dbl>
#> 1 Test     6032      0.302
#> 2 Train   13968      0.698
#>
#> $Grafica
```



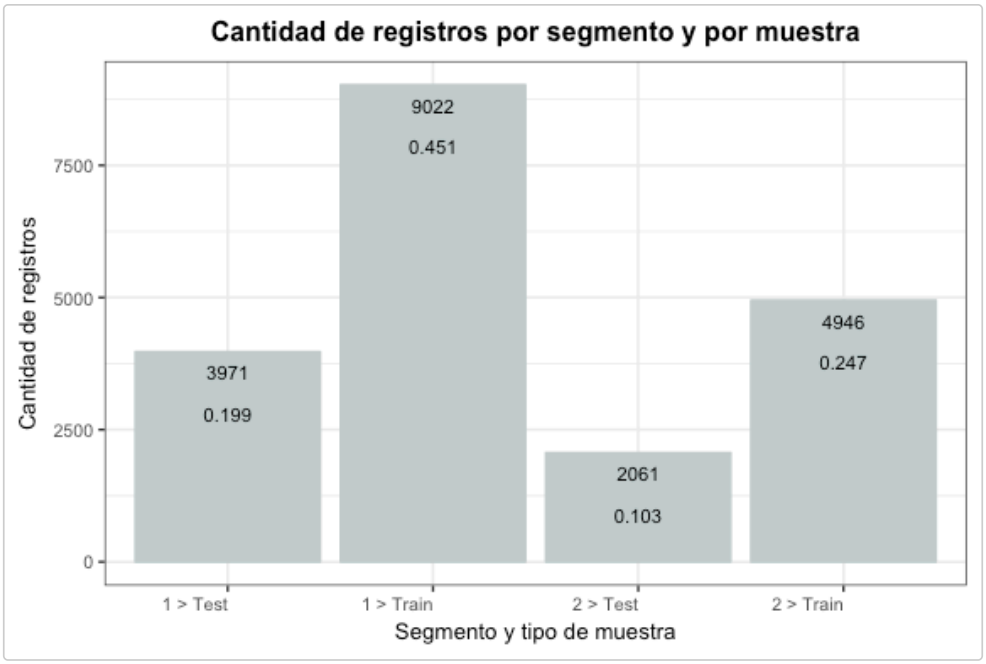

```
w$distrib_coh
#> [[1]]
#> [[1]]$Tabla
#> # A tibble: 24 x 6
#>    cohorte Segmento Total Malos Buenos    TM
#>      <int>    <int> <int> <int>  <int> <dbl>
#>  1  201501        1   592    45    547  532.
#>  2  201502        1   558    52    506  652.
#>  3  201503        1   551    50    501  635.
#>  4  201504        1   538    59    479  768.
#>  5  201505        1   522    49    473  657.
#>  6  201506        1   542    55    487  710.
#>  7  201507        1   530    47    483  621.
#>  8  201508        1   517    41    476  555.
#>  9  201509        1   537    53    484  691.
#> 10  201510        1   561    66    495  824.
#> # ... with 14 more rows
#>
#> [[1]]$Grafica
```




```
#>
#>
#> [[2]]
#> [[2]]$Tabla
#> # A tibble: 24 x 6
#>    cohorte Segmento Total Malos Buenos    TM
#>      <int>    <int> <int> <int>  <int> <dbl>
#>  1  201501        2   266    22    244  579.
#>  2  201502        2   298    20    278  470.
#>  3  201503        2   276    27    249  685.
#>  4  201504        2   291    32    259  770.
#>  5  201505        2   302    31    271  719.
#>  6  201506        2   284    31    253  764.
#>  7  201507        2   316    25    291  554.
#>  8  201508        2   295    39    256  925.
#>  9  201509        2   309    33    276  748.
#> 10  201510        2   280    25    255  625.
#> # ... with 14 more rows
#>
#> [[2]]$Grafica
```

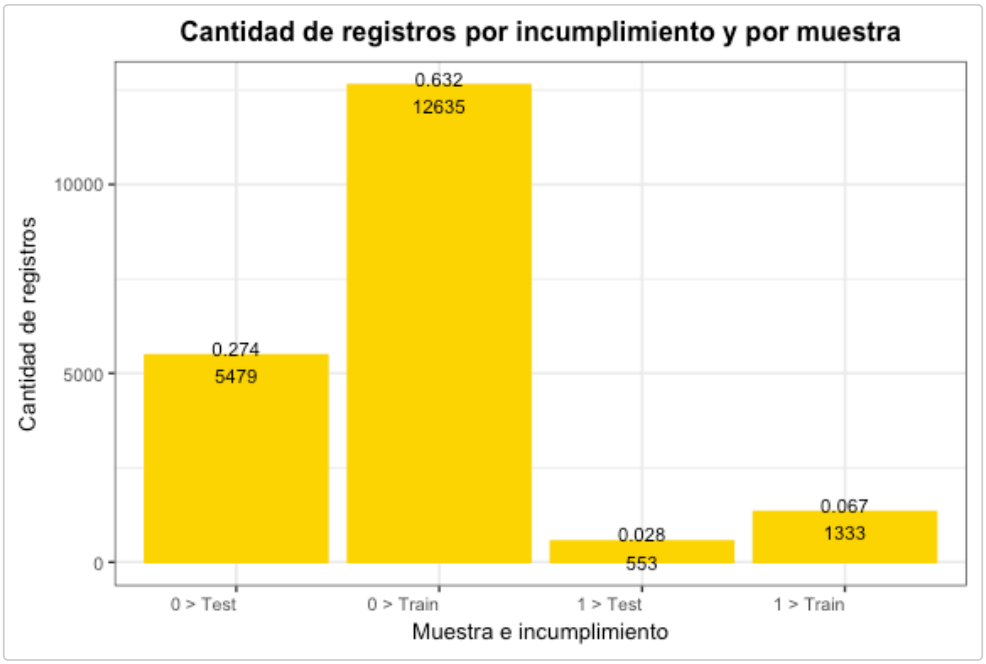



```
w$muestra_mod
#> $Tabla
#> # A tibble: 4 x 5
#> # Groups:   segmento [?]
#>   segmento muestra Total Proporcion tipo
#>      <int> <fct>   <int>      <dbl> <chr>
#> 1        1 Test     3971      0.199 1 > Test
#> 2        1 Train    9022      0.451 1 > Train
#> 3        2 Test     2061      0.103 2 > Test
#> 4        2 Train    4946      0.247 2 > Train
#>
#> $Grafica
```




```
w$tipo_incumpl
#> $Tabla
#> # A tibble: 2 x 3
#>   incmpl Total Proporcion
#>    <int> <int>      <dbl>
#> 1      0 18114     0.906
#> 2      1  1886     0.0940
#>
#> $Grafica
```


```
w$incmpl_muestra
#> $Tabla
#> # A tibble: 4 x 5
#> # Groups:   muestra [?]
#>   muestra incmpl_d Total Proporcion tipo
#>   <fct>      <int> <int>      <dbl> <chr>
#> 1 Test           0  5479     0.274  0 > Test
#> 2 Test           1   553     0.0280 1 > Test
#> 3 Train          0 12635     0.632  0 > Train
#> 4 Train          1  1333     0.0670 1 > Train
#>
#> $Grafica
```




```
w$tmora_segm
#> [[1]]
#> # A tibble: 2 x 6
#>   muestra Segmento Total Malos Buenos     TM
#>   <fct>      <int> <int> <int>  <int>  <dbl>
#> 1 Test           1  3971   370   3601 0.0932
#> 2 Train          1  9022   866   8156 0.0960
#>
#> [[2]]
#> # A tibble: 2 x 6
#>   muestra Segmento Total Malos Buenos     TM
#>   <fct>      <int> <int> <int>  <int>  <dbl>
#> 1 Test           2  2061   183   1878 0.0888
#> 2 Train          2  4946   467   4479 0.0944
```

Parte C

```
start_time <- Sys.time()
vars_numeric <- c("num_1","num_2","num_3")
w <- pto_control_2c(base = d, vars_interes = vars_numeric, muestra = c("Train","Test"), Cant_Q =
0.20,
                    colores = "Blues")

end_time <- Sys.time()
end_time - start_time
#> Time difference of 9.345064 secs
```

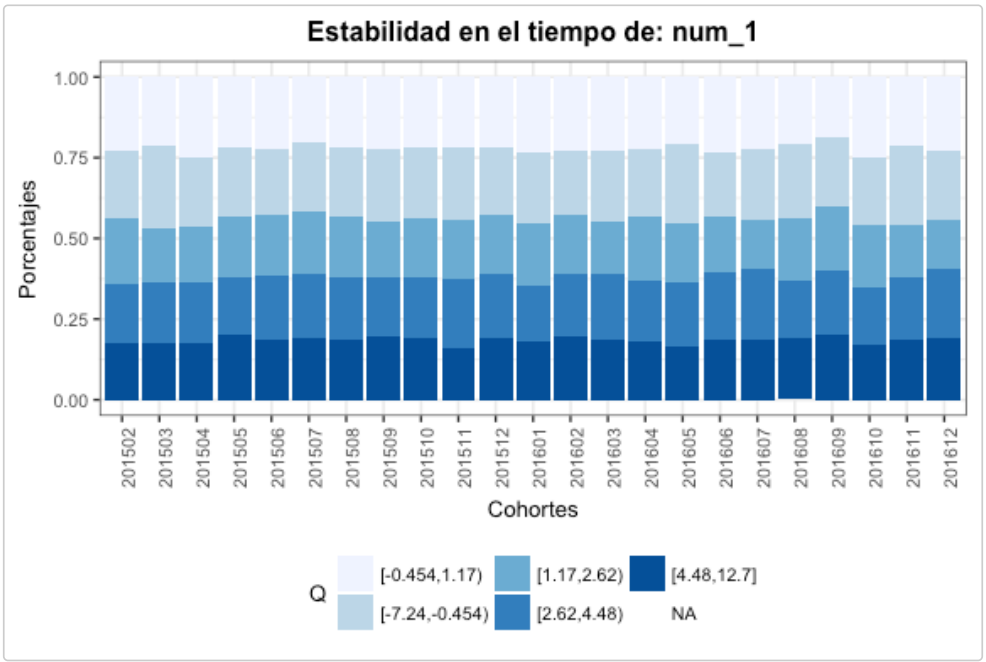
```
names(w)
#> [1] "ie"         "evol_estab"
```

Por conveniencia del espacio de este documento, mostramos algunos de los resultados para las tres variables, pero en un caso real se podrá ver toda la información disponible, como por ejemplo: en la parte **ie**, se podrá ver la *base_X* y la *base_Y* de cada variable, donde ambas tablas se utilizan para calcular el *IE*; y para la parte **evol_estab**, se puede tener acceso a la tabla que la gráfica utiliza para poderse generar.

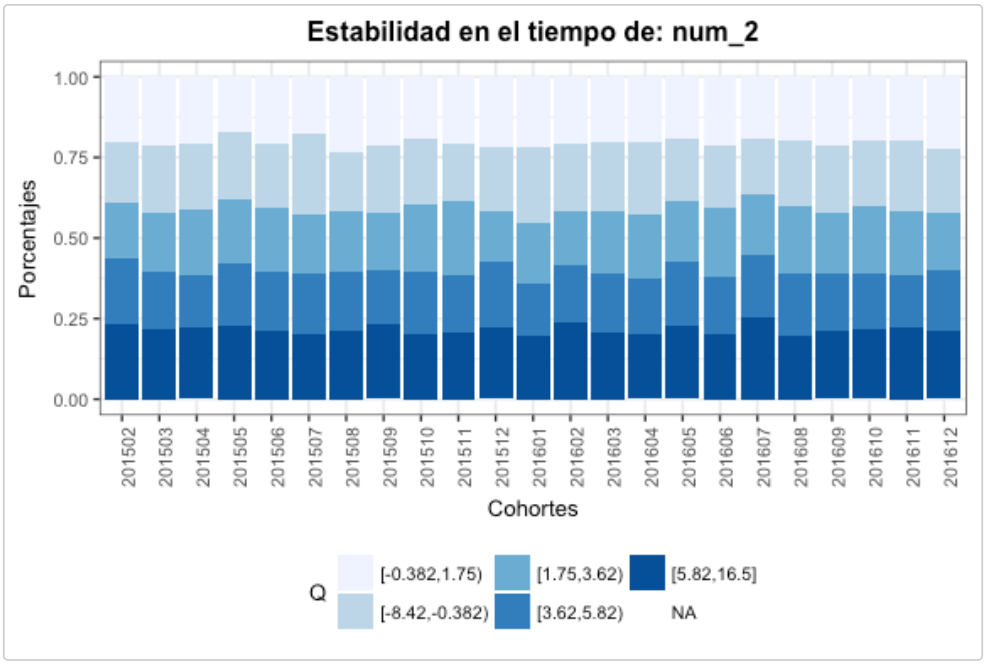
```
# Información del Índice de Estabilidad
w$ie$num_1[c(1:4)]
#> $tabla_X
#> # A tibble: 5 x 4
#>   Q              Frecuencia Maximos Porcentaje
#>   <fct>               <int>   <dbl>      <dbl>
#> 1 [4.34,14.5]          2794  14.5        0.200
#> 2 [2.42,4.34)          2793   4.34       0.200
#> 3 [0.864,2.42)         2794   2.42       0.200
#> 4 [-0.593,0.864)       2793   0.863      0.200
#> 5 [-9.32,-0.593)       2794  -0.593      0.200
#>
#> $tabla_Y
#> # A tibble: 5 x 4
#>   Q              Frecuencia Maximos Porcentaje
#>   <chr>               <int>   <dbl>      <dbl>
#> 1 [4.34,14.5]          1162  13.5        0.193
#> 2 [2.42,4.34)          1296   4.34       0.215
#> 3 [0.864,2.42)         1213   2.42       0.201
#> 4 [-0.593,0.864)       1121   0.863      0.186
#> 5 [-9.32,-0.593)       1240  -0.594      0.206
#>
#> $tabla_XY
#>                Q Frecuencia.x Maximos.x Porcentaje.x Frecuencia.y
#> 1 [-0.593,0.864)         2793    0.8629    0.1999570         1121
#> 2 [-9.32,-0.593)         2794   -0.5931    0.2000286         1240
#> 3   [0.864,2.42)         2794    2.4215    0.2000286         1213
#> 4    [2.42,4.34)         2793    4.3398    0.1999570         1296
#> 5    [4.34,14.5]         2794   14.4949    0.2000286         1162
#>   Maximos.y Porcentaje.y           IE
#> 1    0.8626    0.1858422 1.033276e-03
#> 2   -0.5944    0.2055703 1.514395e-04
#> 3    2.4202    0.2010942 5.660868e-06
#> 4    4.3391    0.2148541 1.070454e-03
#> 5   13.4882    0.1926393 2.781454e-04
#>
#> $indice_est
#> [1] 0.002538975
w$ie$num_2[c(1:4)]
#> $tabla_X
#> # A tibble: 5 x 4
```

```
#>   Q             Frecuencia Maximos Porcentaje
#>   <fct>              <int>   <dbl>      <dbl>
#> 1 [5.94,21.5]         2794  21.5        0.200
#> 2 [3.56,5.94)         2793   5.94       0.200
#> 3 [1.62,3.56)         2794   3.56       0.200
#> 4 [-0.47,1.62)        2794   1.62       0.200
#> 5 [-11.6,-0.47)       2793  -0.471      0.200
#>
#> $tabla_Y
#> # A tibble: 5 x 4
#>   Q             Frecuencia Maximos Porcentaje
#>   <chr>              <int>   <dbl>      <dbl>
#> 1 [5.94,21.5]         1314  17.3        0.218
#> 2 [3.56,5.94)         1172   5.94       0.194
#> 3 [1.62,3.56)         1199   3.55       0.199
#> 4 [-0.47,1.62)        1174   1.62       0.195
#> 5 [-11.6,-0.47)       1173  -0.473      0.194
#>
#> $tabla_XY
#>               Q Frecuencia.x Maximos.x Porcentaje.x Frecuencia.y Maximos.y
#> 1  [-0.47,1.62)         2794    1.6167    0.2000286         1174    1.6164
#> 2 [-11.6,-0.47)         2793   -0.4709    0.1999570         1173   -0.4729
#> 3   [1.62,3.56)         2794    3.5563    0.2000286         1199    3.5536
#> 4   [3.56,5.94)         2793    5.9411    0.1999570         1172    5.9408
#> 5   [5.94,21.5]         2794   21.4755    0.2000286         1314   17.2775
#>   Porcentaje.y           IE
#> 1    0.1946286 1.477824e-04
#> 2    0.1944629 1.530753e-04
#> 3    0.1987732 7.904191e-06
#> 4    0.1942971 1.625214e-04
#> 5    0.2178382 1.519013e-03
#>
#> $indice_est
#> [1] 0.001990297
w$ie$num_3[c(1:4)]
#> $tabla_X
#> # A tibble: 5 x 4
#>   Q              Frecuencia Maximos Porcentaje
#>   <fct>              <int>   <dbl>      <dbl>
#> 1 [7.78,22.3]         2794  22.3        0.200
#> 2 [4.86,7.78)         2793   7.78       0.200
#> 3 [2.42,4.86)         2794   4.86       0.200
#> 4 [-0.347,2.42)       2793   2.42       0.200
#> 5 [-14.7,-0.347)      2794  -0.347      0.200
#>
#> $tabla_Y
#> # A tibble: 5 x 4
#>   Q              Frecuencia Maximos Porcentaje
#>   <chr>              <int>   <dbl>      <dbl>
#> 1 [7.78,22.3]         1277  21.4        0.212
#> 2 [4.86,7.78)         1228   7.78       0.204
```

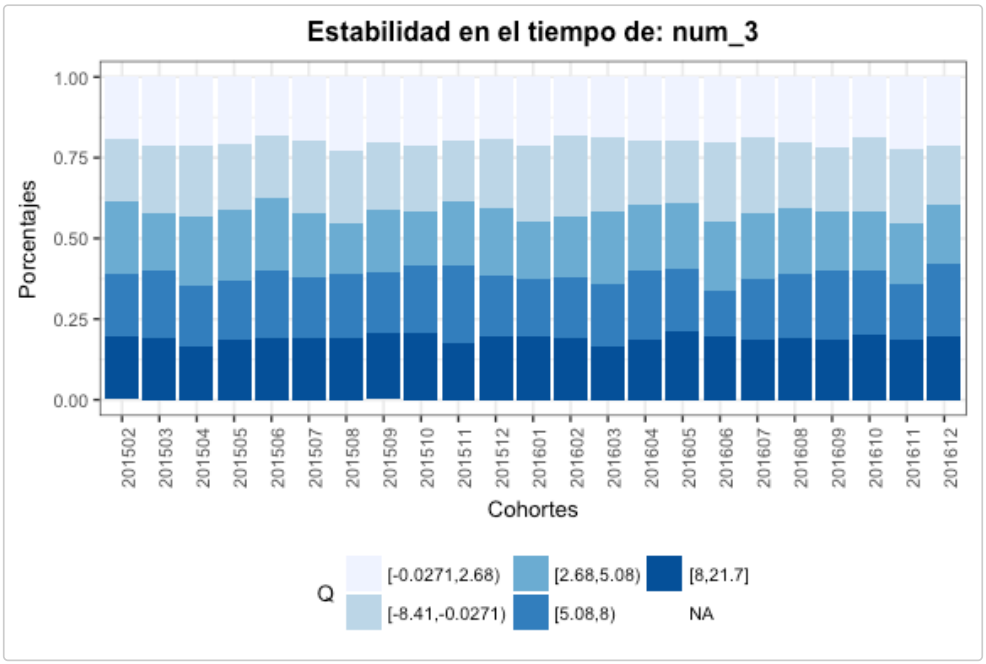
```
#> 3 [2.42,4.86)          1223   4.86       0.203
#> 4 [-0.347,2.42)        1167   2.42       0.193
#> 5 [-14.7,-0.347)       1137  -0.352      0.188
#>
#> $tabla_XY
#>                Q Frecuencia.x Maximos.x Porcentaje.x Frecuencia.y
#> 1  [-0.347,2.42)         2793    2.4196    0.1999570         1167
#> 2 [-14.7,-0.347)         2794   -0.3473    0.2000286         1137
#> 3    [2.42,4.86)         2794    4.8646    0.2000286         1223
#> 4    [4.86,7.78)         2793    7.7840    0.1999570         1228
#> 5    [7.78,22.3]         2794   22.3055    0.2000286         1277
#>   Maximos.y Porcentaje.y           IE
#> 1    2.4185    0.1934682 2.140652e-04
#> 2   -0.3518    0.1884947 6.850086e-04
#> 3    4.8644    0.2027520 3.682780e-05
#> 4    7.7822    0.2035809 6.508778e-05
#> 5   21.4342    0.2117042 6.623535e-04
#>
#> $indice_est
#> [1] 0.001663343

# Gráficas del índice de estabilidad
w$evol_estab$num_1$Grafica
```




```
w$evol_estab$num_2$Grafica
```




```
w$evol_estab$num_3$Grafica
```



Punto de Control 3

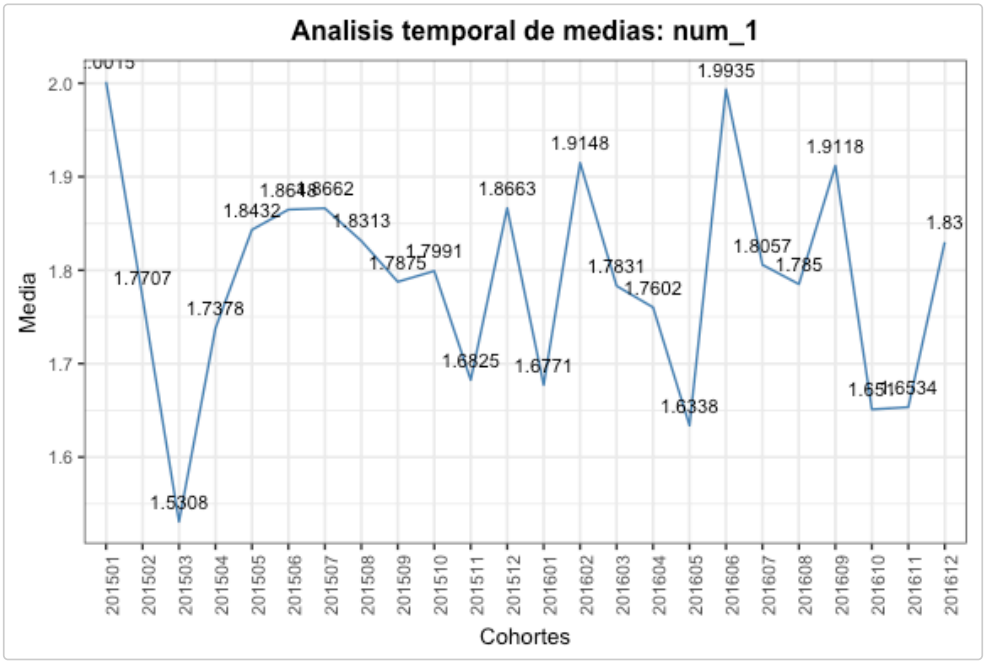
Este punto de control atiende:

- Análisis descriptivo de la base de datos
    - Distribución
    - Análisis temporal
- Análisis de coherencias: calidad de los datos

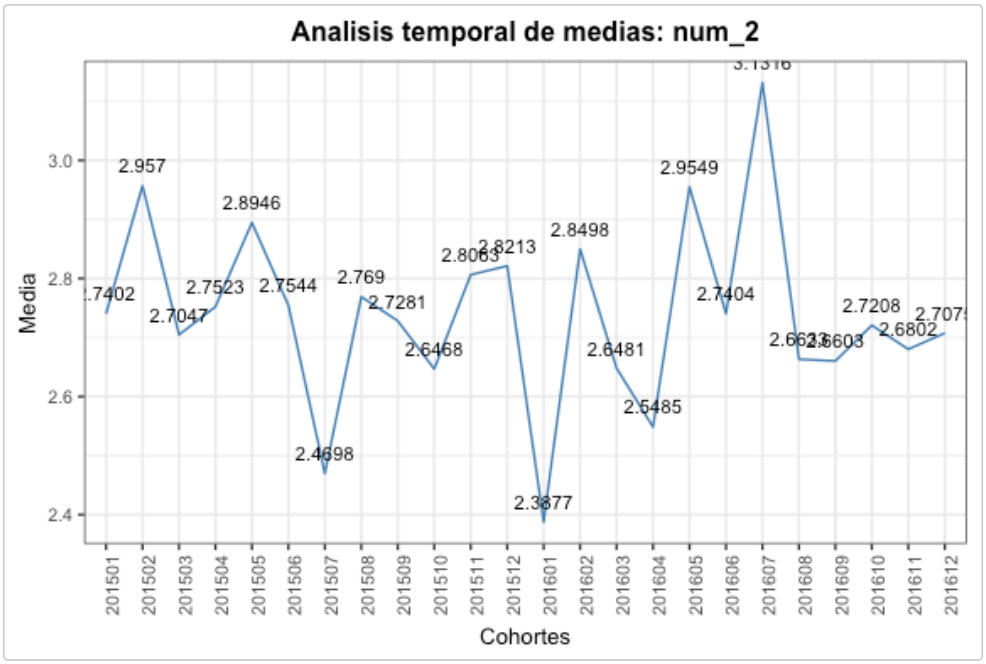
Parte A

```
start_time <- Sys.time()
vars_numeric <- c("num_1","num_2","num_3","num_4","num_5")
vars_categ <- c("cat_1","cat_2","cat_3")

espvals <- list(num_1 = c(-99999999999:-99999999990),
                num_2 = c(-99999999999:-99999999990),
                num_3 = c(-99999999999:-99999999990),
                num_4 = c(-99999999999:-99999999990),
                num_5 = c(-99999999999:-99999999990))

w <- pto_control_3a(base = d, vars_numeric = vars_numeric, vars_categ = vars_categ,
                    esp_values = espvals, incumpl = "incmpl_d", cohorte = "cohorte")

end_time <- Sys.time()
end_time - start_time
#> Time difference of 1.734956 secs

names(w)
#> [1] "numer"    "categ"    "temp_num" "temp_cat"


w$numer
#>           .05     .10     .25     .50     .75     .90     .95  Mean
#> num_1 -2.7577 -1.7314 -0.1877  1.6517  3.7849  5.6855  6.7921 1.808
#> num_2 -3.4359 -2.0558  0.1485  2.6239  5.3432  7.8146  9.3962 2.769
#> num_3 -4.0788 -2.3344  0.5334  3.6803  7.0219 10.0649 11.9427 3.792
#> num_4 -4.7707 -2.6283  0.8727  4.7072  8.7297 12.3683 14.4217 4.787
#> num_5  -5.437  -2.934   1.213   5.701  10.357  14.640  17.220 5.789
#>       distinct missing     n
#> num_1    18203       0 20000
#> num_2    18582       0 20000
#> num_3    18858       0 20000
#> num_4    19086       0 20000
#> num_5    19207       0 20000
w$categ
#> $cat_1
#> # A tibble: 5 x 6
#>   cat_1 Frecuencia Proporcion Buenos Malos     TM
#>   <int>      <int>      <dbl>  <int> <int>  <dbl>
#> 1     1       4051      0.203   3644   407 0.100
#> 2     2       3987      0.199   3630   357 0.0895
#> 3     3       4033      0.202   3658   375 0.0930
```

```
#> 4     4       4017      0.201   3644   373 0.0929
#> 5     5       3912      0.196   3538   374 0.0956
#>
#> $cat_2
#> # A tibble: 4 x 6
#>   cat_2 Frecuencia Proporcion Buenos Malos     TM
#>   <int>      <int>      <dbl>  <int> <int>  <dbl>
#> 1     1       3965      0.198   3575   390 0.0984
#> 2     2       6028      0.301   5494   534 0.0886
#> 3     3       7986      0.399   7215   771 0.0965
#> 4     4       2021      0.101   1830   191 0.0945
#>
#> $cat_3
#> # A tibble: 3 x 6
#>   cat_3 Frecuencia Proporcion Buenos Malos     TM
#>   <int>      <int>      <dbl>  <int> <int>  <dbl>
#> 1     1       6061      0.303   5475   586 0.0967
#> 2     2       5915      0.296   5343   572 0.0967
#> 3     3       8024      0.401   7296   728 0.0907
w$temp_num
#> $medias
#> $medias$num_1
```

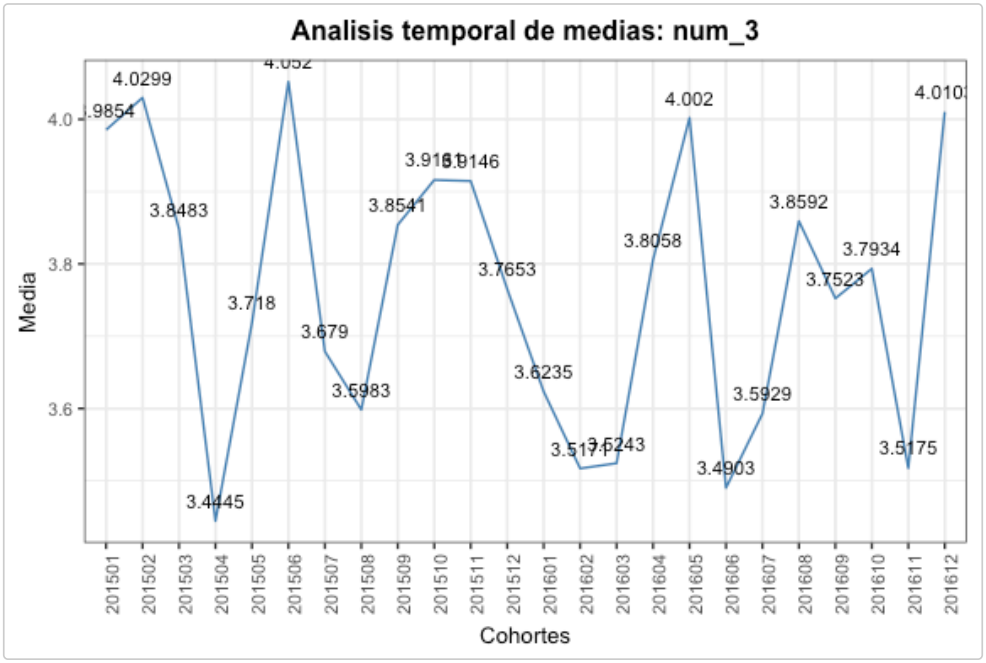



```
#>
#> $medias$num_2
```



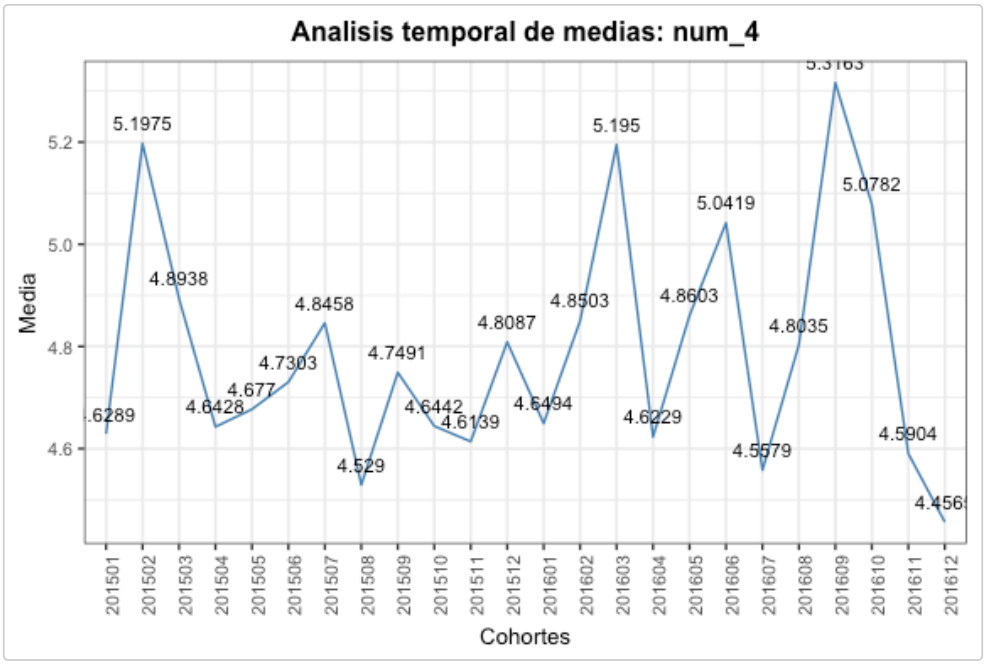
#>

#> $medias$num_3



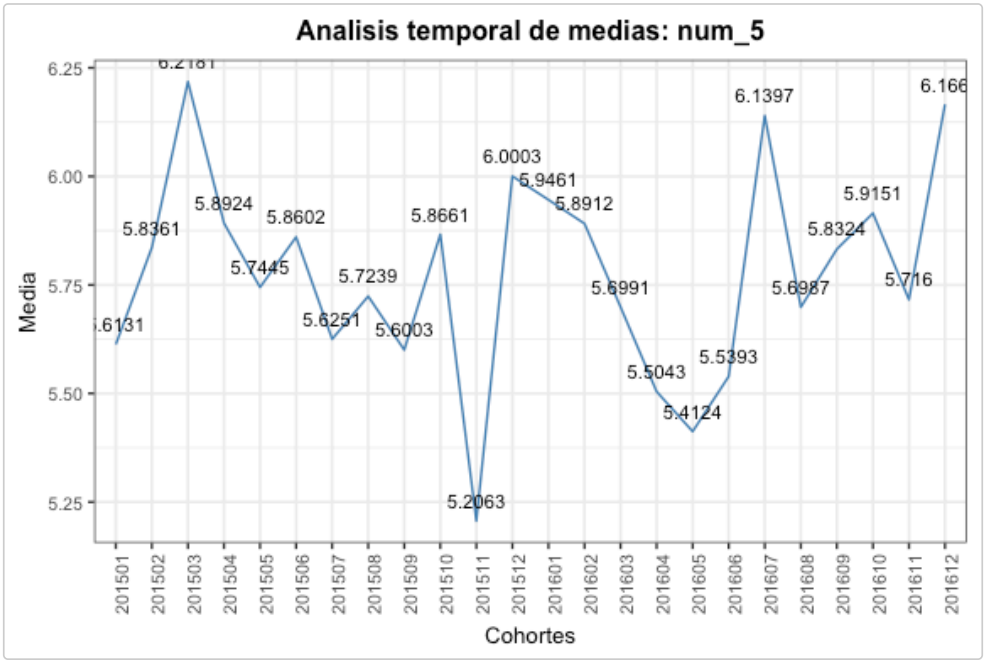
#>

#> $medias$num_4



#>

#> $medias$num_5

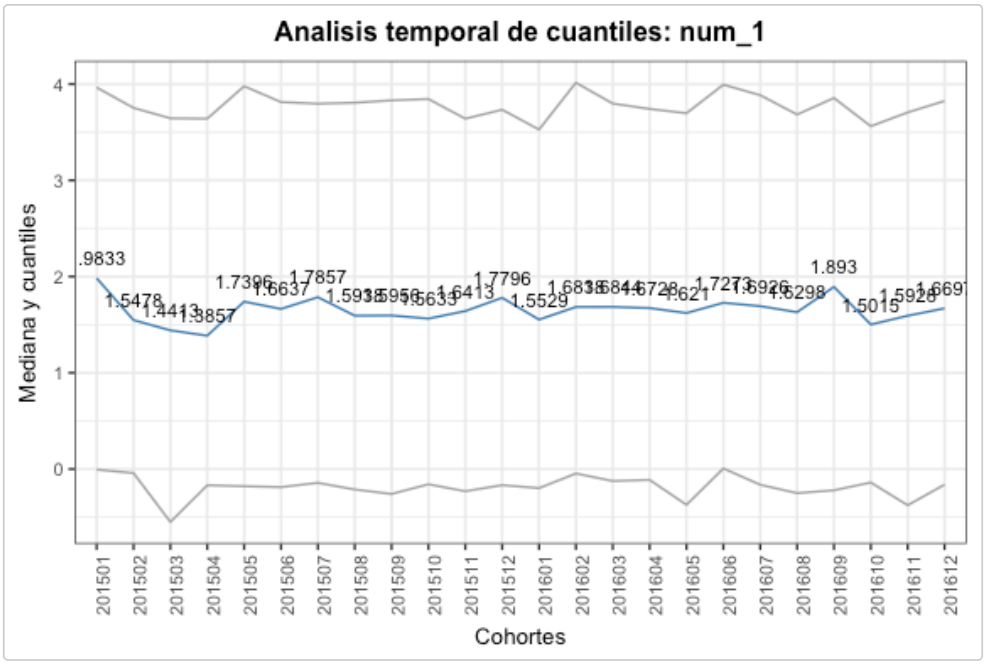


#>

#>

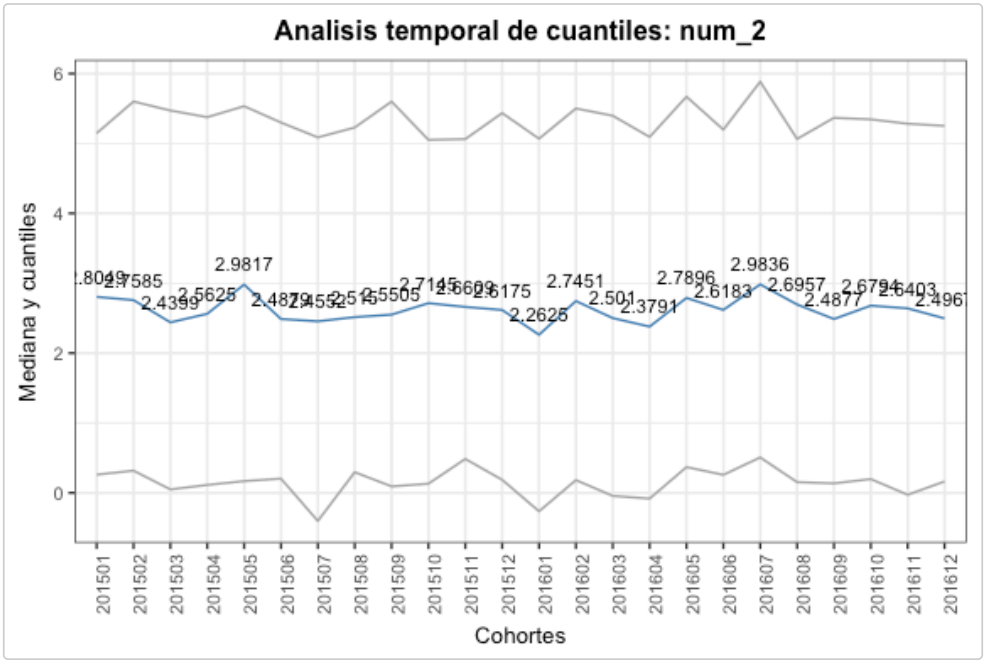
#> $quantiles

#> $quantiles$num_1



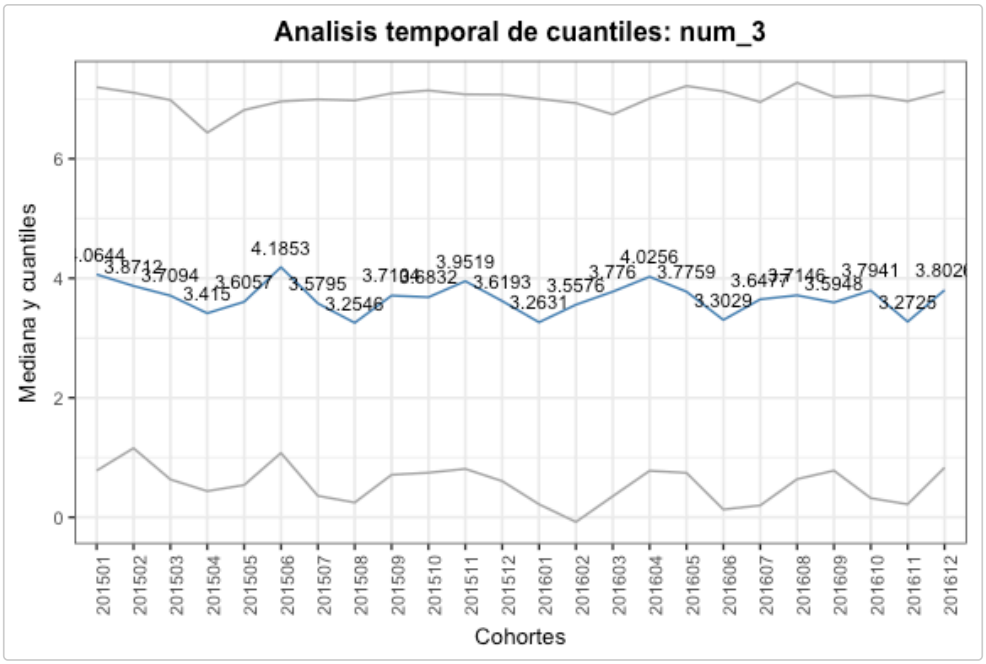
#>

#> $quantiles$num_2



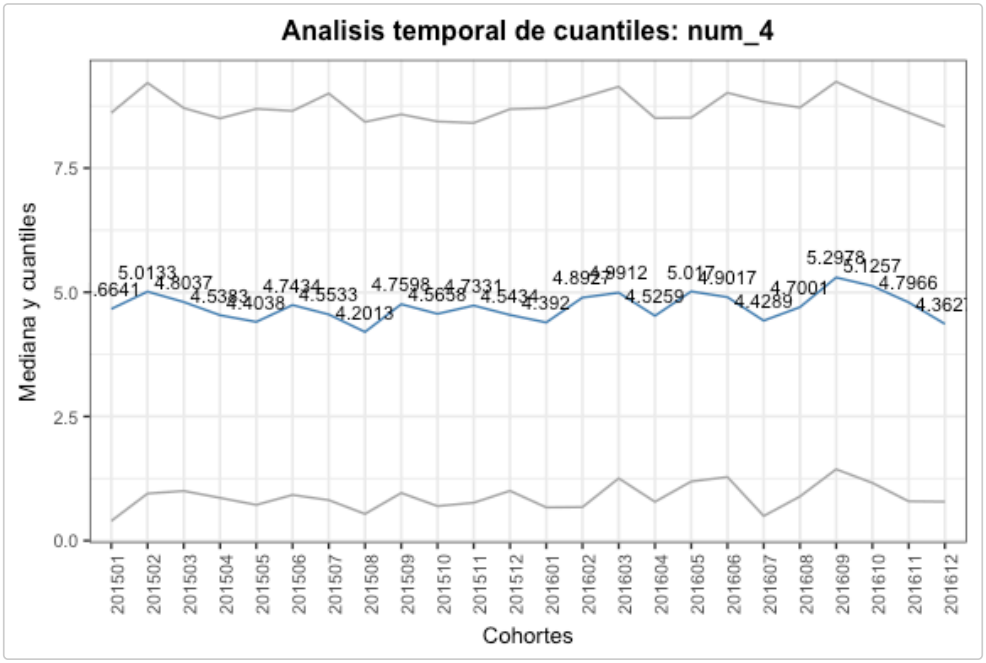
#>

#> $quantiles$num_3



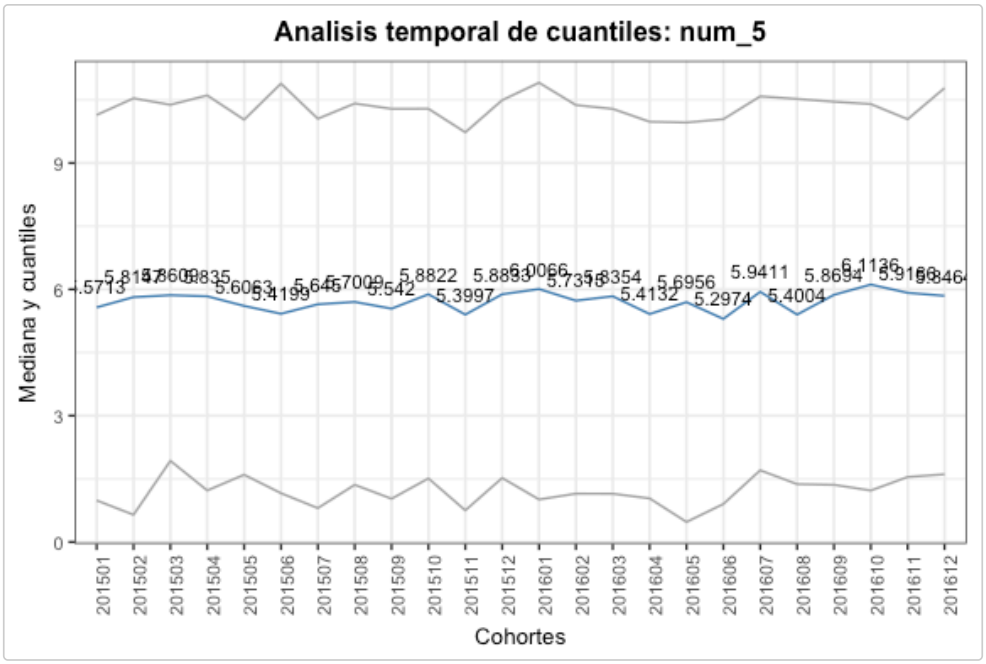
#>

#> $quantiles$num_4

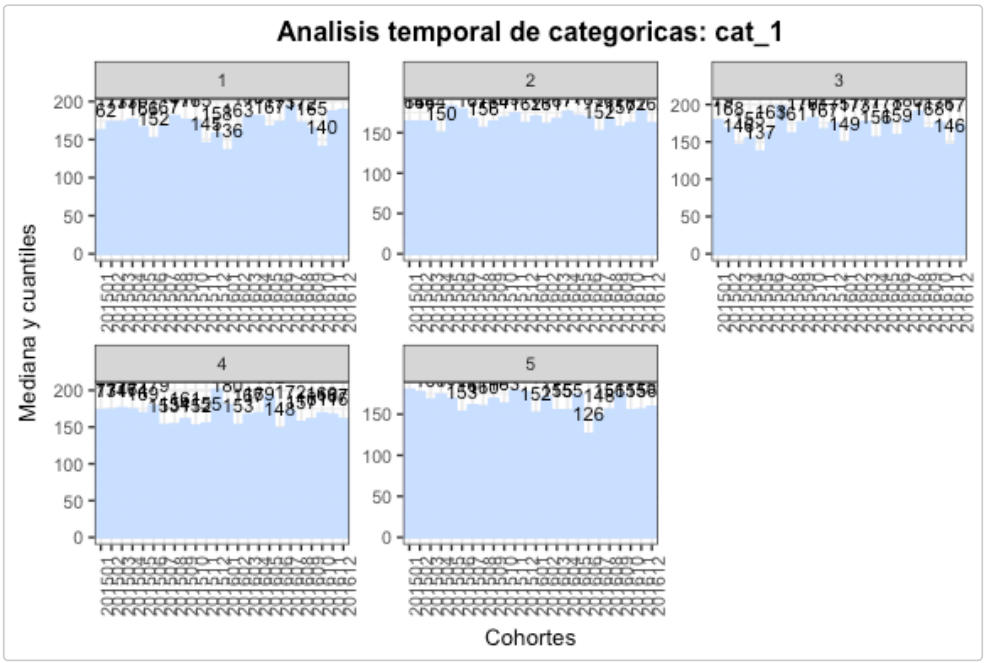


#>

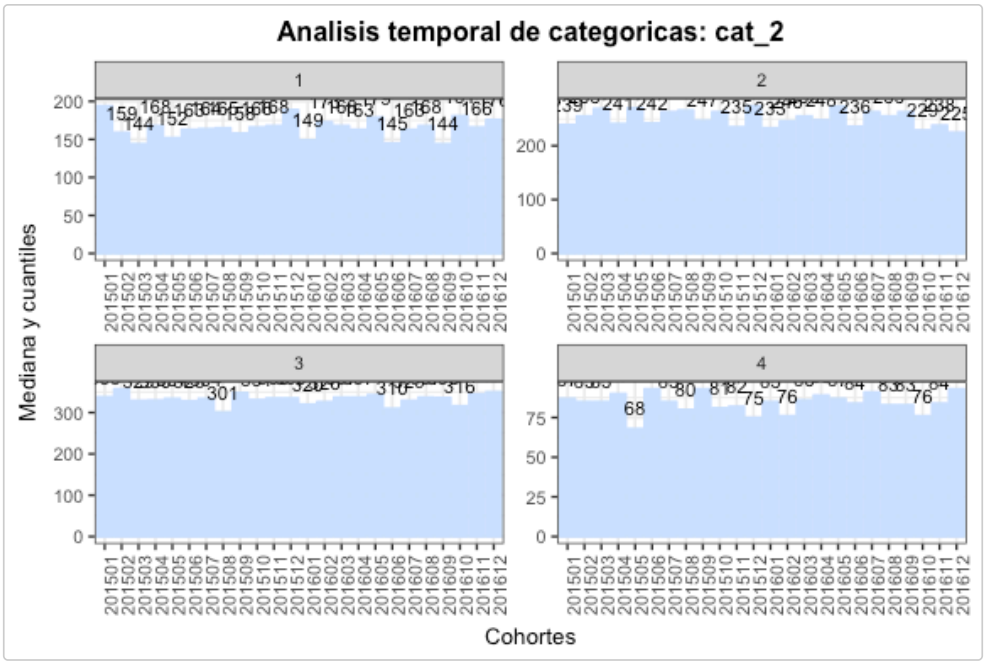
#> $quantiles$num_5



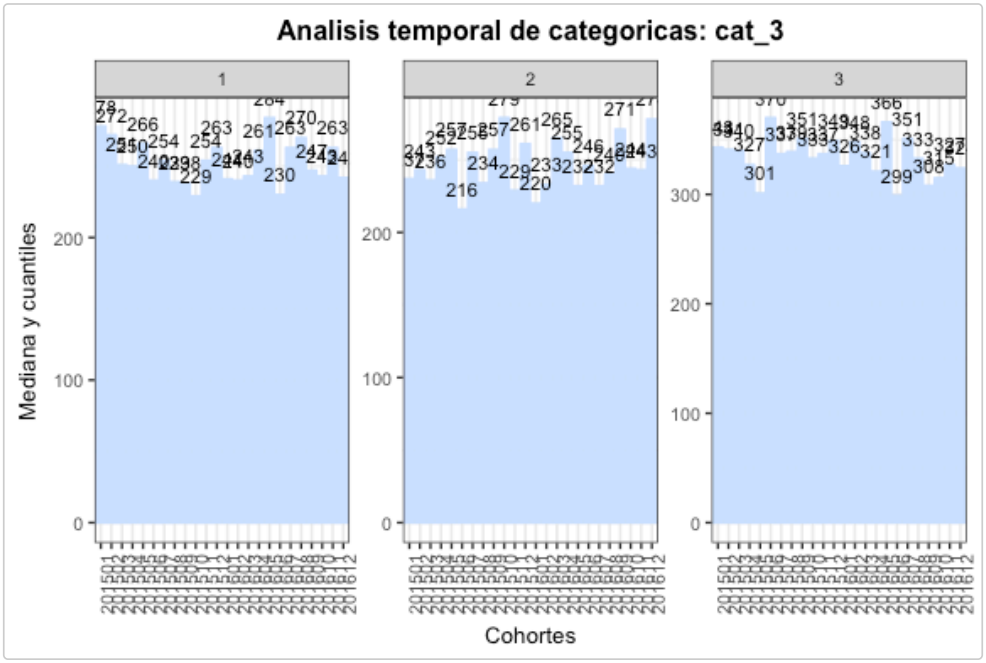

```
w$temp_cat
#> $cat_1
```




```
#>
#> $cat_2
```



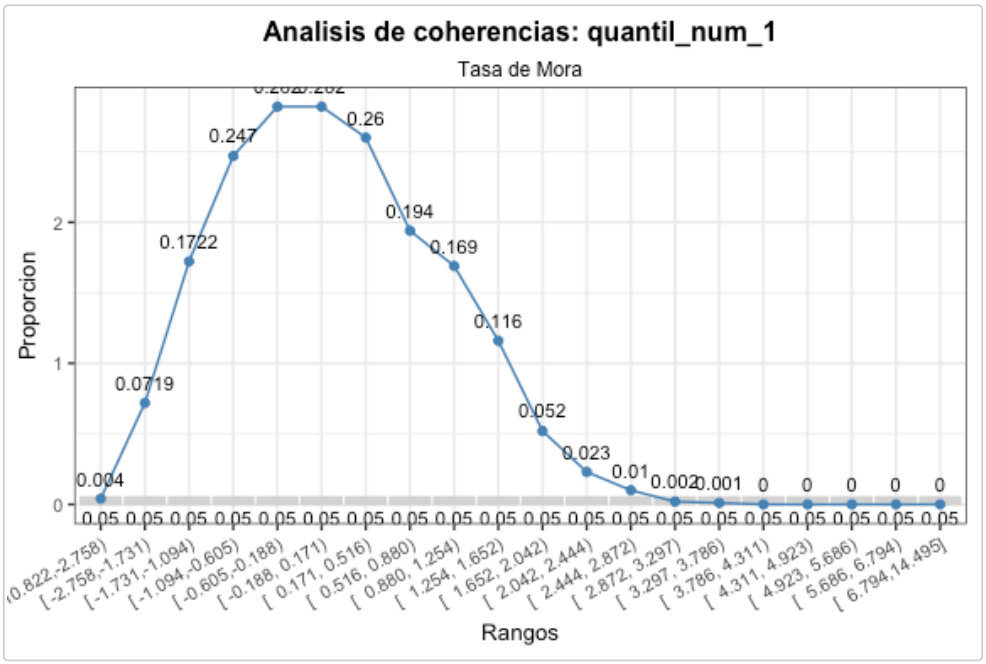

```
#>
#> $cat_3
```



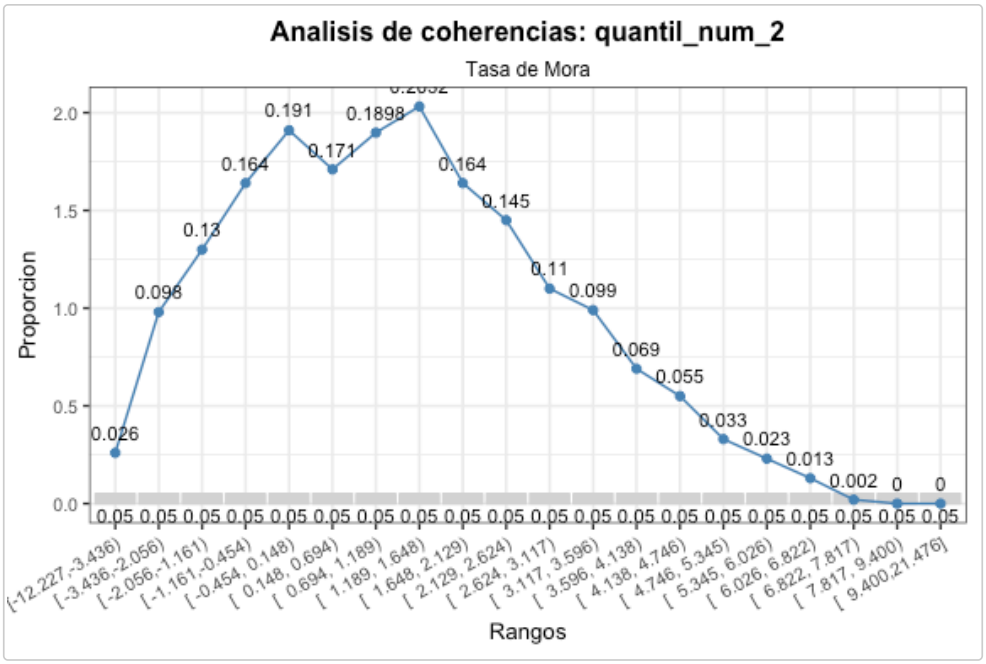
Parte B

```
start_time <- Sys.time()
```

```
w <- pto_control_3b(base = d, vars_numeric = vars_numeric, vars_categ = vars_categ,
                    esp_values = espvals, incumpl = "incmpl_d")

end_time <- Sys.time()
end_time - start_time
#> Time difference of 0.587044 secs

names(w)
#> [1] "coher_Numer" "coher_categ"


w$coher_Numer
#> $quantil_num_1
```

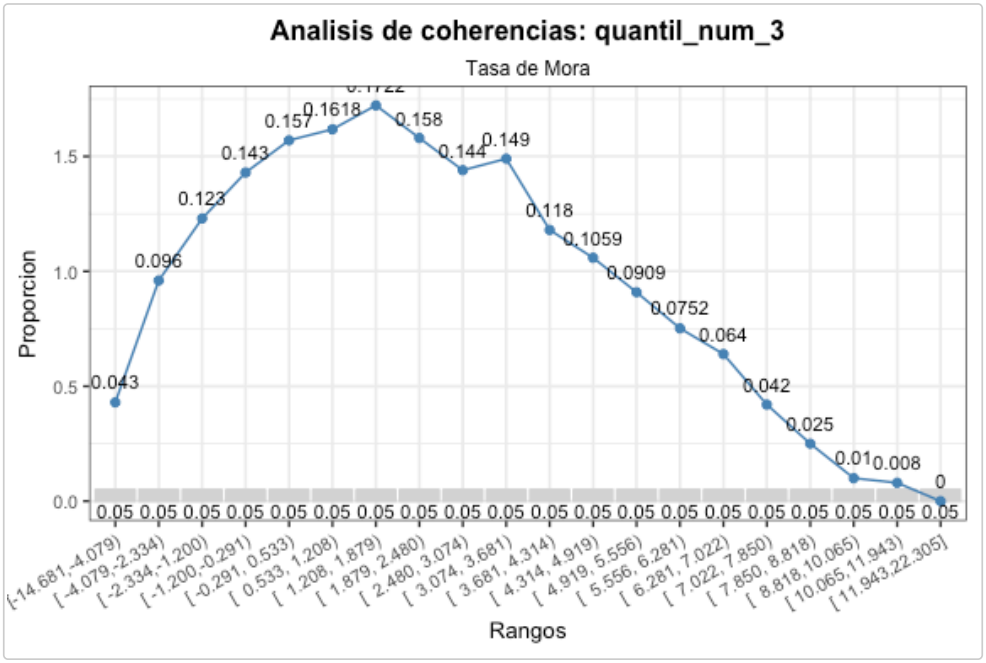



```
#>
#> $quantil_num_2
```



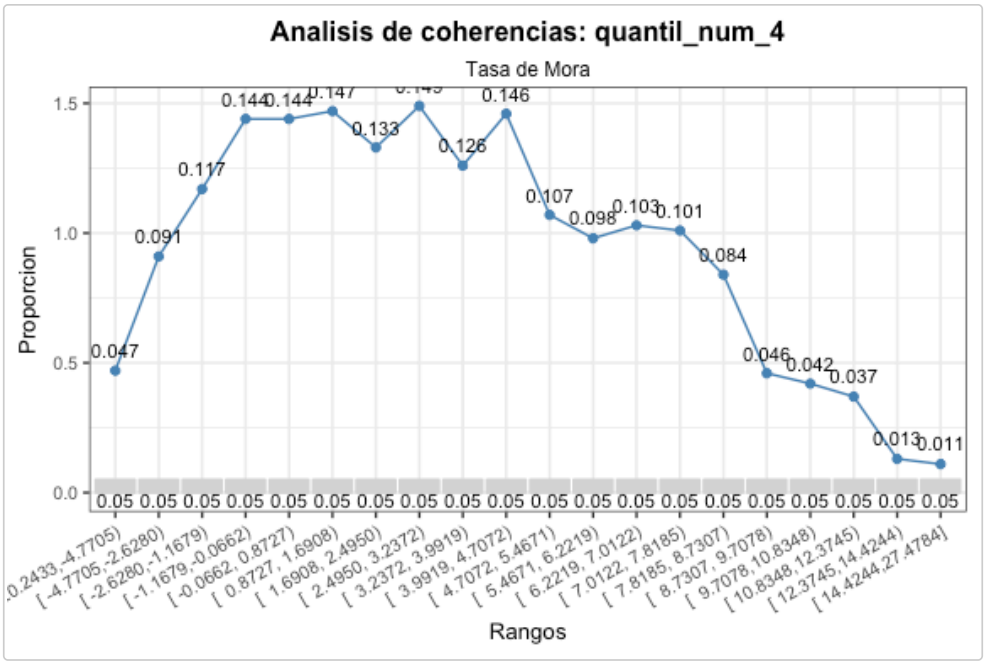
#>

#> $quantil_num_3



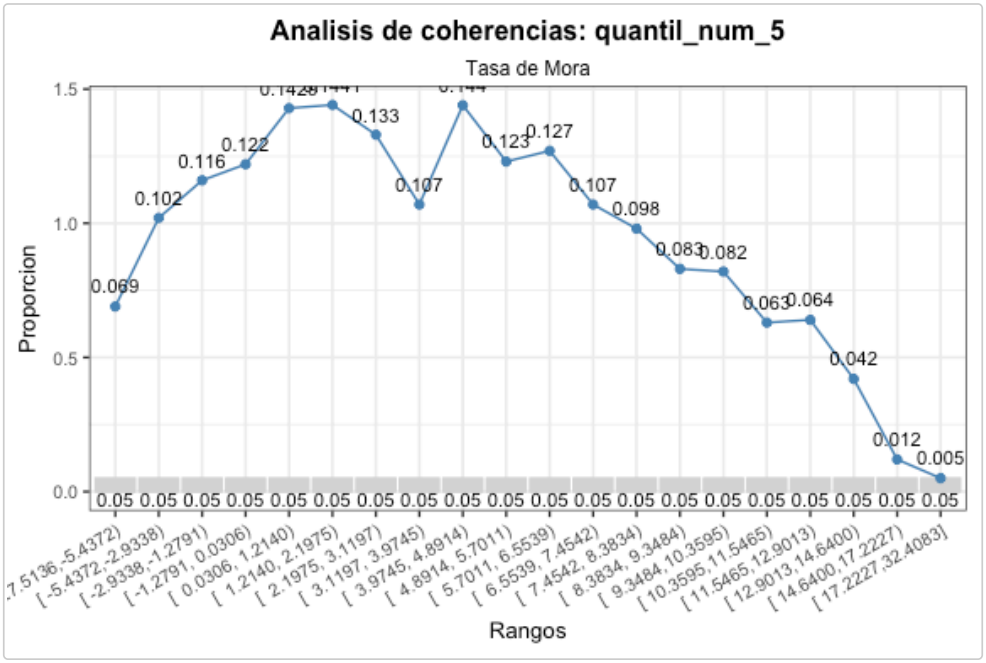
#>

#> $quantil_num_4



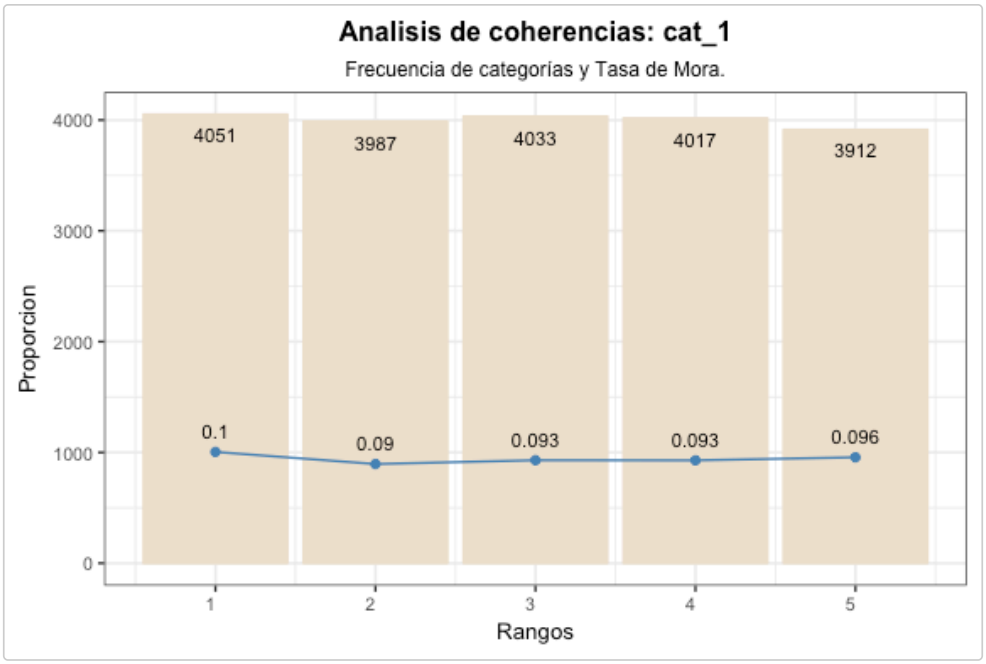
#>

#> $quantil_num_5



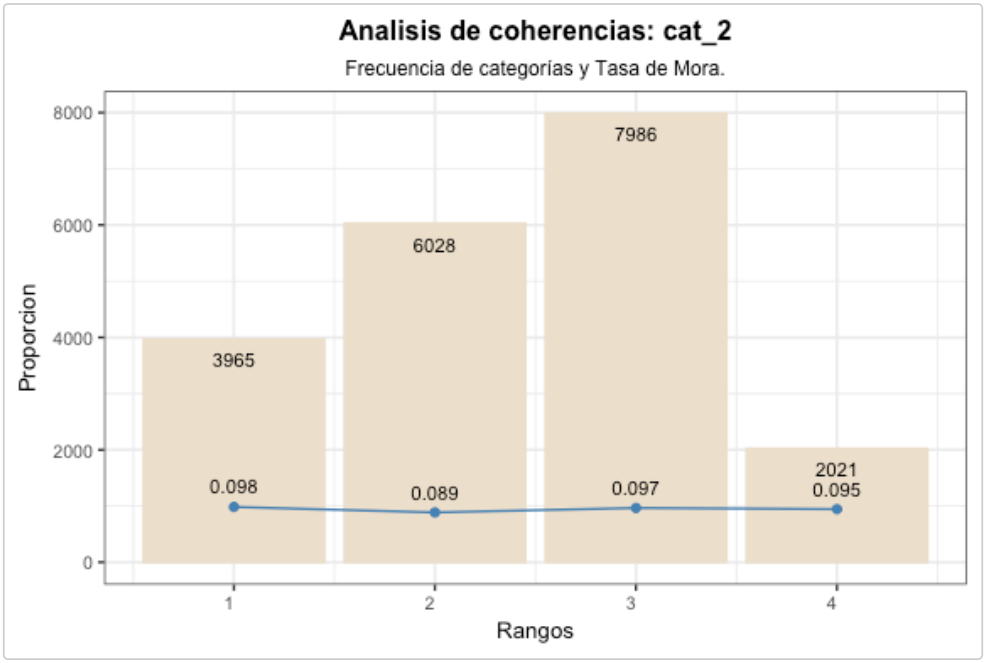
w$coher_categ

#> *$cat_1*


Analisis de coherencias: cat_1
Frecuencia de categorías y Tasa de Mora.
Proporcion
Rangos
4051
3987
4033
4017
3912
0.1
0.09
0.093
0.093
0.096
0
1000
2000
3000
4000
1
2
3
4
5

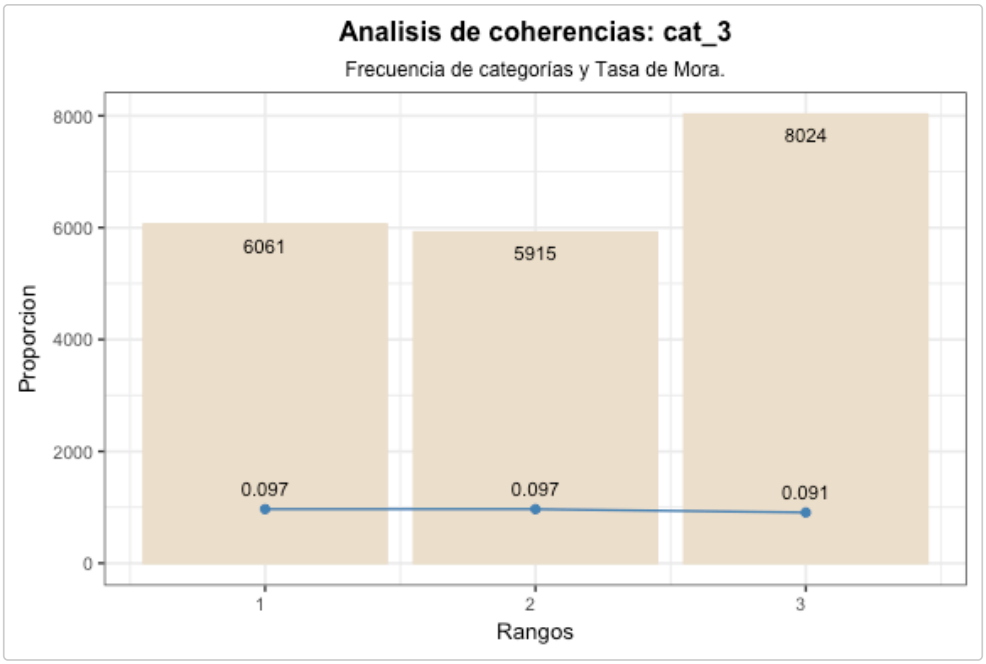
#>

#> $cat_2


Analisis de coherencias: cat_2
Frecuencia de categorías y Tasa de Mora.
Proporcion
Rangos
3965
6028
7986
2021
0.098
0.089
0.097
0.095
0
2000
4000
6000
8000
1
2
3
4

#>

#> $cat_3



Punto de Control 4

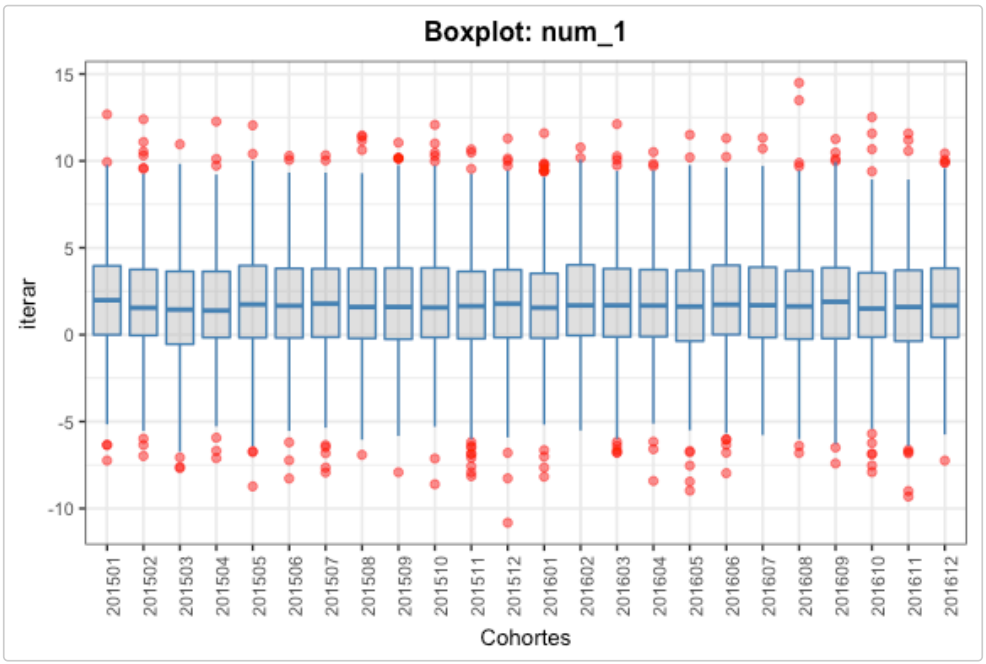
Este punto de control atiende:

- Depuración de los datos
  - Análisis de correlación
  - Variabilidad
  - Análisis de valores nulos
  - Detección de outliers
- Creación de nuevas variables
  - Ratios sectorizados

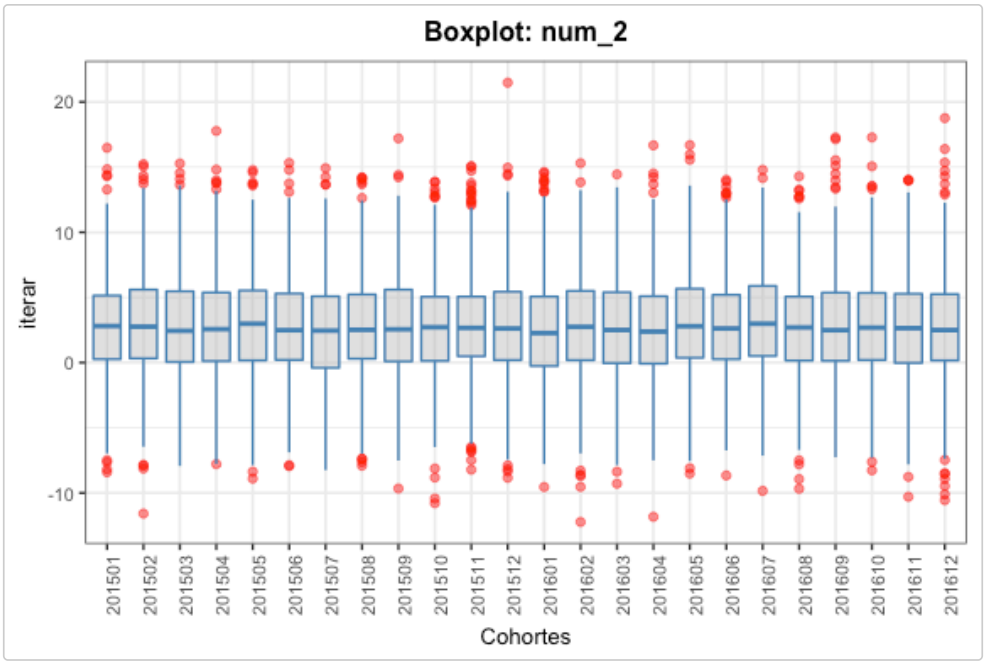
Parte A

```
start_time <- Sys.time()
w <- pto_control_4a(base = d, vars_numeric = vars_numeric, esp_values = espvals,
                    cohorte = "cohorte", threshold = 0.6)

end_time <- Sys.time()
end_time - start_time
#> Time difference of 0.3918 secs

names(w)
#> [1] "corr_filtro" "low_var"     "vars_cNA"    "boxplot"
```

```
w$corr_filtro
#> [1] var1 var2 cor
#> <0 rows> (or 0-length row.names)
w$low_var
#>           0%     25%   50%    75%  100%   IQR   Var Concent
#> num_1 -10.82 -0.1877 1.652  3.785 14.49 3.973 15.69  0.0002
#> num_3 -14.68  0.5334 3.680  7.022 22.31 6.489 17.54  0.0002
#> num_2 -12.23  0.1485 2.624  5.343 21.48 5.195 15.41  0.0002
#> num_5 -27.51  1.2126 5.701 10.357 32.41 9.144 15.26  0.0002
#> num_4 -20.24  0.8727 4.707  8.730 27.48 7.857 16.46  0.0001
w$vars_cNA
#>   variable colNA filas Prop
#> 1    num_1     0 20000    0
#> 2    num_2     0 20000    0
#> 3    num_3     0 20000    0
#> 4    num_4     0 20000    0
#> 5    num_5     0 20000    0
w$boxplot
#> $num_1
```

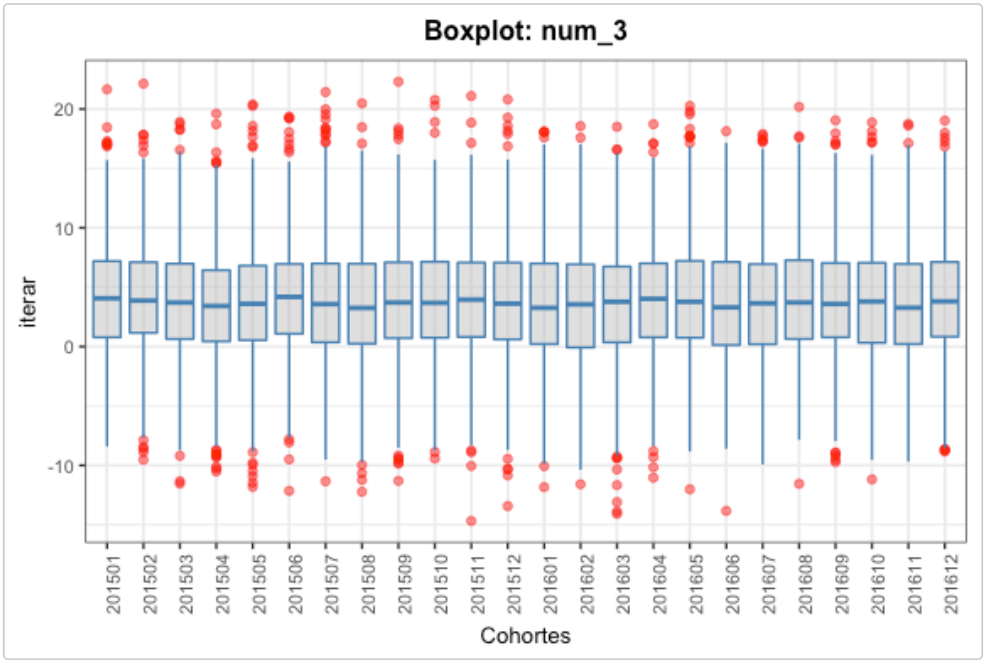



```
#>
#> $num_2
```


Boxplot: num_2
iterar
20
10
0
-10
201501
201502
201503
201504
201505
201506
201507
201508
201509
201510
201511
201512
201601
201602
201603
201604
201605
201606
201607
201608
201609
201610
201611
201612
Cohortes

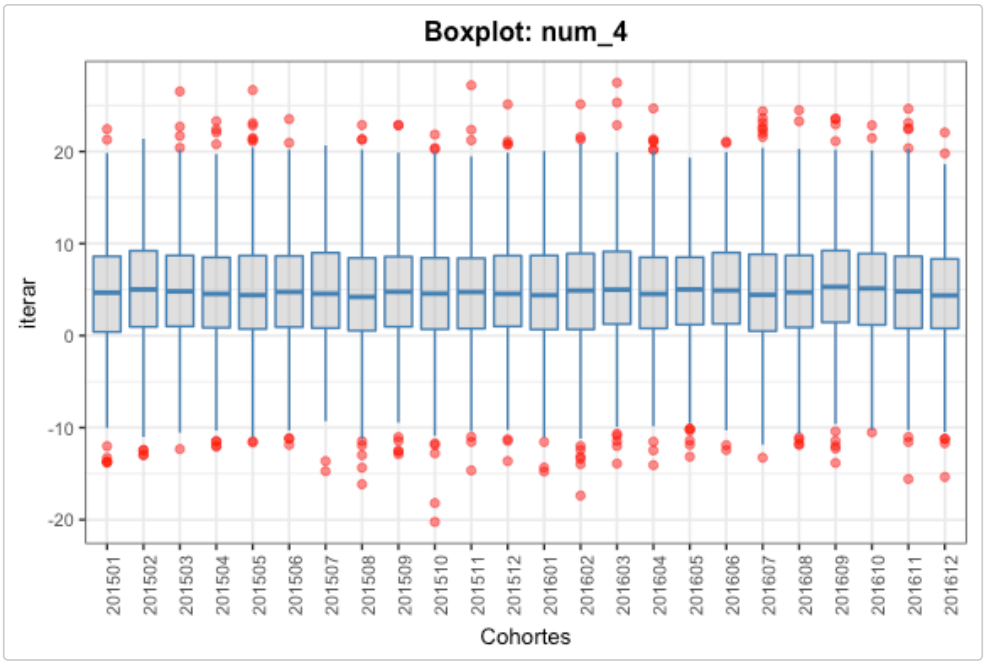
#>

#> $num_3

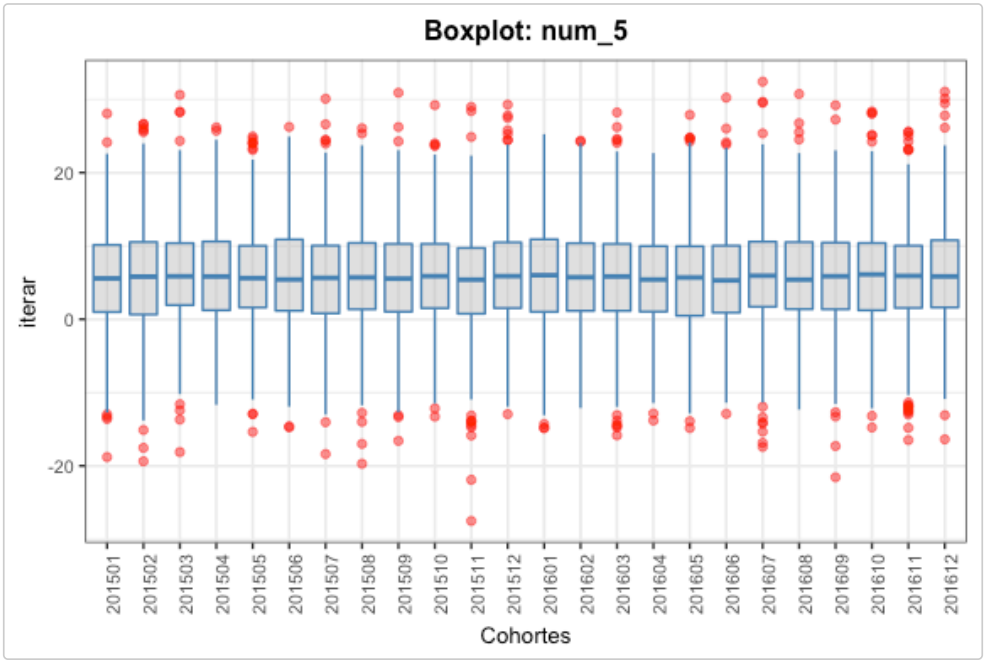

Boxplot: num_3
iterar
20
10
0
-10
201501
201502
201503
201504
201505
201506
201507
201508
201509
201510
201511
201512
201601
201602
201603
201604
201605
201606
201607
201608
201609
201610
201611
201612
Cohortes

#>

#> $num_4




```
#>
#> $num_5
```



Parte B

```
# Para esta función creamos la variable "sector"
```

```
start_time <- Sys.time()
d$sector_d <- sample(c("Autos","Bienes","Transp","Constr"), size = tot_tabla_dummy,
                     prob = c(0.2,0.35,0.30,0.15), replace = TRUE)

w <- pto_control_4b(base = d, sector = "sector_d", vars_interes = vars_numeric, esp_values =
espvals)

end_time <- Sys.time()
end_time - start_time
#> Time difference of 0.1651 secs

names(w)
#> [1] "Estadisticos" "tabla"


# Estadísticos de la Mediana y de la Meda por variable
knitr::kable(head(w$Estadisticos))
```

	sector	Variable	Freq	Mean	Median	StdDev	MEDA
2	Autos	num_1	3990	1.817	1.679	2.933	1.954
3	Bienes	num_1	7028	1.808	1.627	2.962	1.990
4	Constr	num_1	3024	1.710	1.587	2.941	1.995
5	Transp	num_1	5958	1.851	1.696	2.889	1.948
6	Autos	num_2	3990	2.778	2.683	3.783	2.519
7	Bienes	num_2	7028	2.739	2.526	3.938	2.629

```
# Ejemplo de lo que entregaría la función
#head(w$tabla)
names(w$tabla)
#>  [1] "incmpl_d"    "muestra"     "segmento"    "cohorte"     "num_1"
#>  [6] "num_2"       "num_3"       "num_4"       "num_5"       "cat_1"
#> [11] "cat_2"       "cat_3"       "sector"      "num_1_RSect" "num_2_RSect"
#> [16] "num_3_RSect" "num_4_RSect" "num_5_RSect"

# Ejemplo de las variables estandarizadas
head(w$tabla[c("sector","num_1","num_1_RSect","num_2","num_2_RSect","num_3","num_3_RSect")],10)
#>    sector   num_1 num_1_RSect   num_2 num_2_RSect  num_3 num_3_RSect
#> 1  Transp  2.7180     0.52462  0.5208     -0.8170  5.384      0.5059
#> 2  Bienes  0.0040    -0.81571  2.6765      0.0572  7.266      1.1142
#> 3  Bienes  3.9065     1.14524  0.3317     -0.8348 -1.192     -1.4657
#> 4  Transp -3.8692    -2.85708 10.2206      2.9672  9.646      1.8310
#> 5  Transp  1.3250    -0.19051  0.8049     -0.7062  2.984     -0.2403
#> 6  Transp -0.8288    -1.29622  0.4409     -0.8482  5.216      0.4538
#> 7  Bienes  5.6438     2.01822  1.2212     -0.4964 -5.296     -2.7176
#> 8   Autos  1.8001     0.06211  5.4036      1.0796  4.861      0.3643
```

```
#> 9  Bienes -1.7617    -1.70295  1.3388     -0.4517  6.447      0.8643
#> 10 Bienes -0.2686    -0.95269  1.7863     -0.2815 -2.973     -2.0091
```

Punto de Control 5

Este punto de control atiende:

- División train y test
- Análisis bivariante

```
# Agregamos los valores especiales a dos de las variables numericas:
d <- d %>% mutate(num_4=if_else(runif(tot_tabla_dummy) < 0.01, -99999999999,num_4))

# Dividimos y nos quedamos con la base train
d_train <- d %>% filter(muestra == "Train")
d_test <- d %>% filter(muestra == "Test")
```

Análisis bivariante

Para el análisis bivarariente se toma en cuenta la base de train. Se toma el trameado que optimiza el gini y se toma la tendencia que tenga mayor gini.

A continuación se muestra el uso de la mtr8_yk que devuelve la tabla de tramos por cada variable y como se tienen 2 segmentos, se realiza el análisis bivariante para estos dos segmentos. Este análisis tambien recaba el anásis de cada tramo para que sean estadísticamente separables observando que el tramo supere el 5% de significancia, si no lo hace se pega con el tramo siguiente dependiendo de la tendencia. Si el tramo contiene en su totalidad valores buenos o malos también se pegal al tramo siguiente porque en la calificación de su woe puede dar infinito si no se revisara este caso.

```
# Nos quedamos con la base train
segmentos_train <- list("1" = d_train %>% filter(segmento == 1),
                        "2" = d_train %>% filter(segmento == 2))

vvect <- c("num_1","num_2","num_3","num_4","num_5","cat_1","cat_2","cat_3")
vcat <- c("cat_1","cat_2","cat_3")
lesp <- list(num_4 = c(-99999999999))

# realizamos el bivariado por segmento
mtr8 <- list()

for(i in c(1,2)){
  mtr8[[paste0("seg_",i)]] <- MTR8_yk(fbase = segmentos_train[[paste0(i)]],
                                      vvect = vvect,
                                      vobj = "incmpl_d",
                                      vcat = vcat,
                                      porcentaje = 5,
```

```
                                    verbose = FALSE,
                                    vesp = lesp)
}
#> [1] "mtr4"
#> [1] "sin vesp:-> num_1"
#> [1] "sin vesp:-> num_2"
#> [1] "sin vesp:-> num_3"
#> [1] "con vesp:-> num_4"
#> [1] "sin vesp:-> num_5"
#> [1] "categorica:-> cat_1"
#> [1] "categorica:-> cat_2"
#> [1] "categorica:-> cat_3"
#> [1] "mtr5"
#> [1] "num_1"
#> [1] "num_2"
#> [1] "num_3"
#> [1] "num_4"
#> [1] "num_5"
#> [1] "mtr7"
#> [1] "num_5"
#> [1] "num_4"
#> [1] "num_3"
#> [1] "num_2"
#> [1] "cat_1"
#> [1] "cat_2"
#> [1] "cat_3"
#> [1] "num_1"
#> [1] "bdcalif"
#> [1] "mtr4"
#> [1] "sin vesp:-> num_1"
#> [1] "sin vesp:-> num_2"
#> [1] "sin vesp:-> num_3"
#> [1] "con vesp:-> num_4"
#> [1] "sin vesp:-> num_5"
#> [1] "categorica:-> cat_1"
#> [1] "categorica:-> cat_2"
#> [1] "categorica:-> cat_3"
#> [1] "mtr5"
#> [1] "num_1"
#> [1] "num_2"
#> [1] "num_3"
#> [1] "num_4"
#> [1] "num_5"
#> [1] "mtr7"
#> [1] "num_5"
#> [1] "num_4"
#> [1] "num_3"
#> [1] "num_2"
#> [1] "cat_1"
#> [1] "cat_2"
#> [1] "cat_3"
```

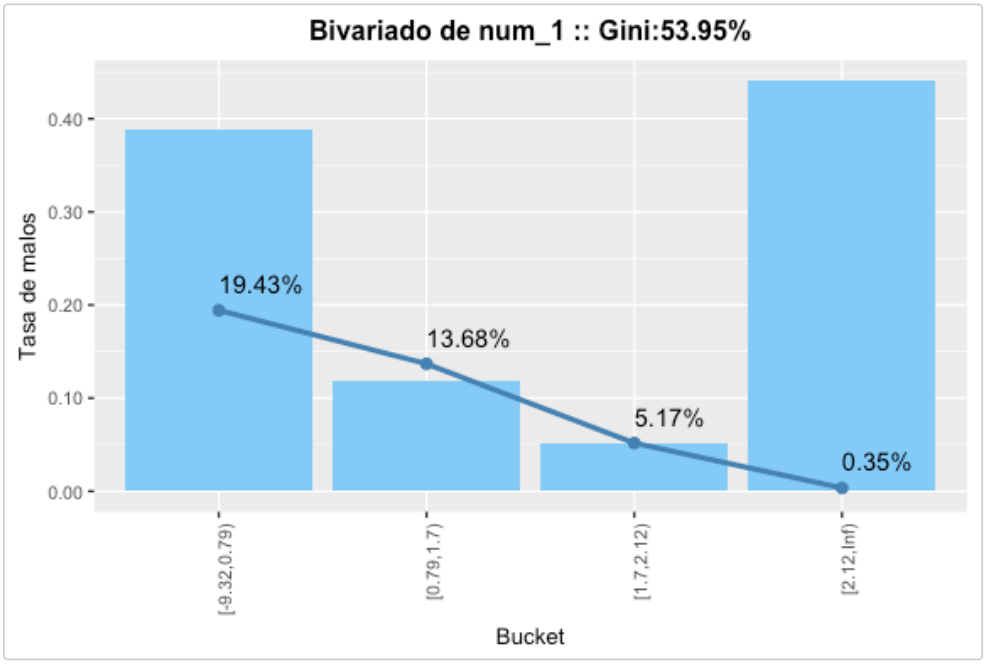
```
#> [1] "num_1"
#> [1] "bdcalif"

# este objeto tiene la base calificada en woes y la tabla con los tramos optimos
# con esta informacion se puede reportar el gini y su tendencia
tablaResumenGinis <- mtr8$seg_1$lmtr5$lista$woes_nesp %>%
    select(var, tndnc, Gini) %>%
    group_by(var) %>%
    filter(row_number() == 1) %>%
    arrange(desc(Gini))

knitr::kable(tablaResumenGinis)
```

var	tndnc	Gini
num_1	down	0.5395
cat_1	cat	0.5000
cat_2	cat	0.5000
cat_3	cat	0.5000
num_2	down	0.3694
num_3	down	0.2841
num_5	down	0.2264
num_4	down	0.2241

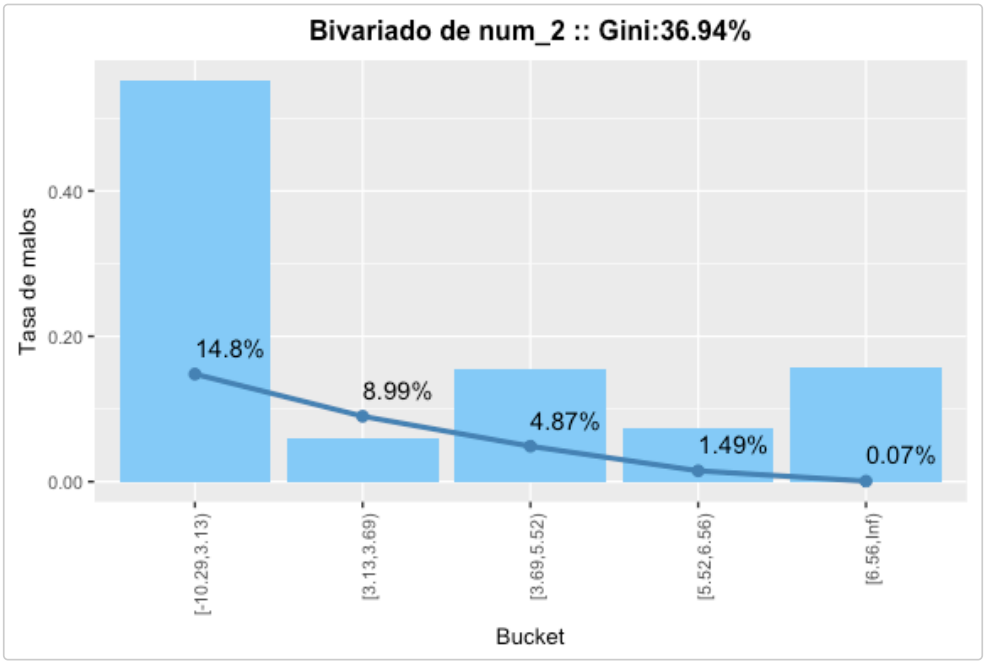
A continuación se puede observar la gráfica de bivariados de todas las variables por segmento

```
# Agregamos los valores especiales a dos de las variables numericas:
biv_1 <- bivariado(mtr8$seg_1$lmtr5r)
#> [1] "num_1"
#> [1] "num_2"
#> [1] "num_3"
#> [1] "num_4"
#> [1] "num_5"
#> [1] "cat_1"
#> [1] "cat_2"
#> [1] "cat_3"
biv_2 <- bivariado(mtr8$seg_2$lmtr5r)
#> [1] "num_1"
#> [1] "num_2"
#> [1] "num_3"
#> [1] "num_4"
#> [1] "num_5"
#> [1] "cat_1"
#> [1] "cat_2"
#> [1] "cat_3"
```

```
biv_1$listaFinal
#> $num_1
```

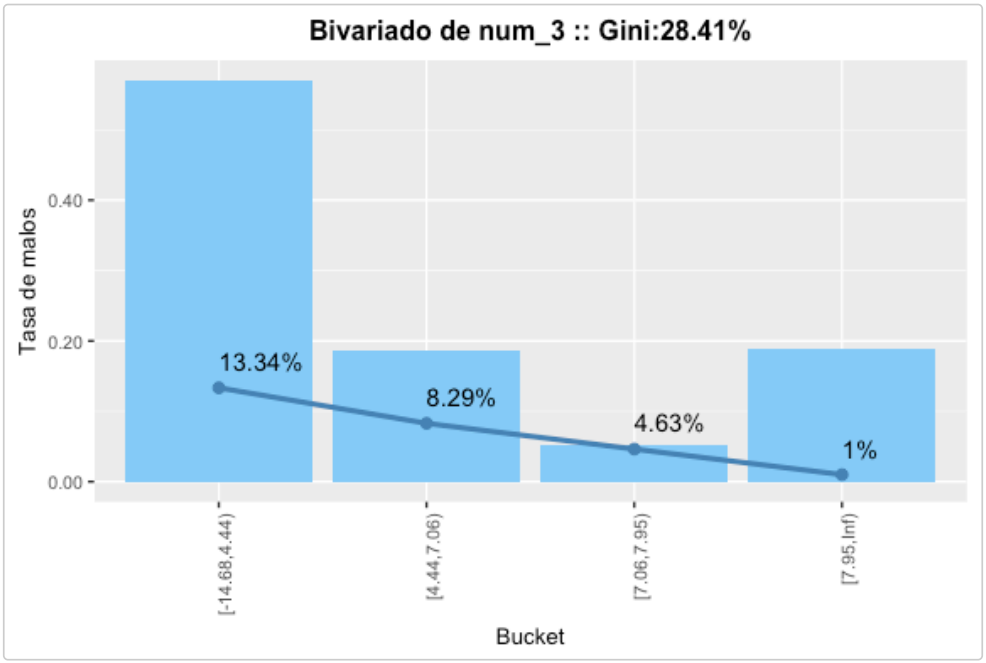



```
#>
#> $num_2
```

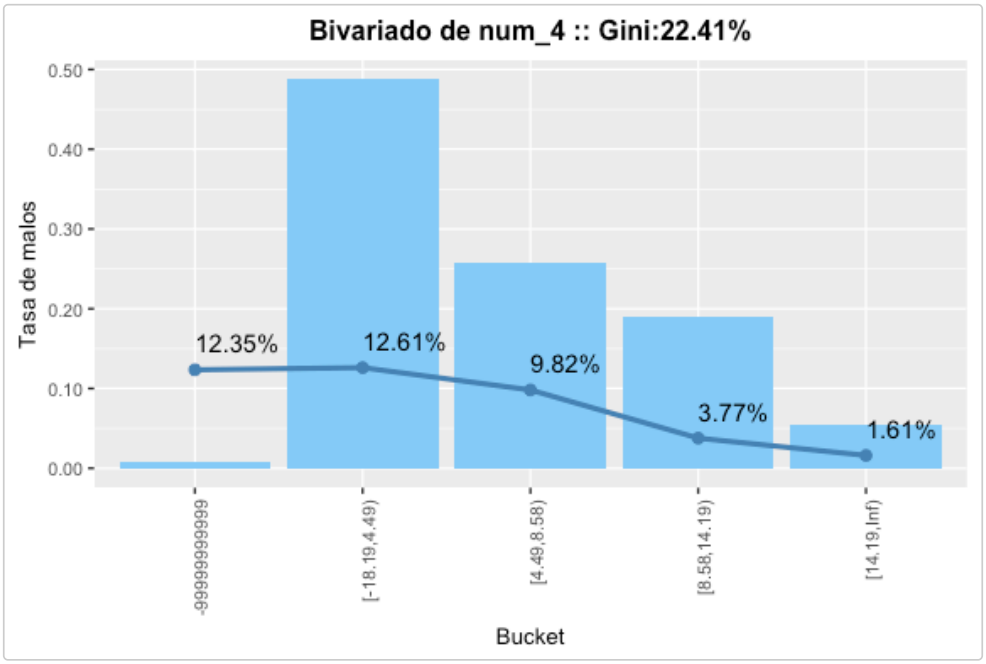



```
#>
```

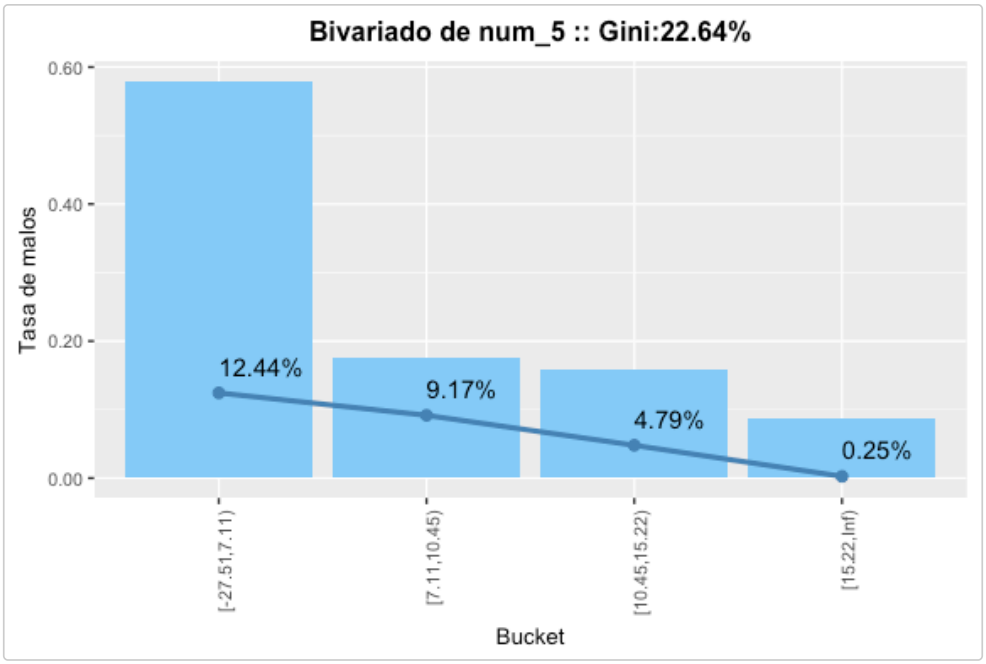
#> $num_3



#>
#> $num_4

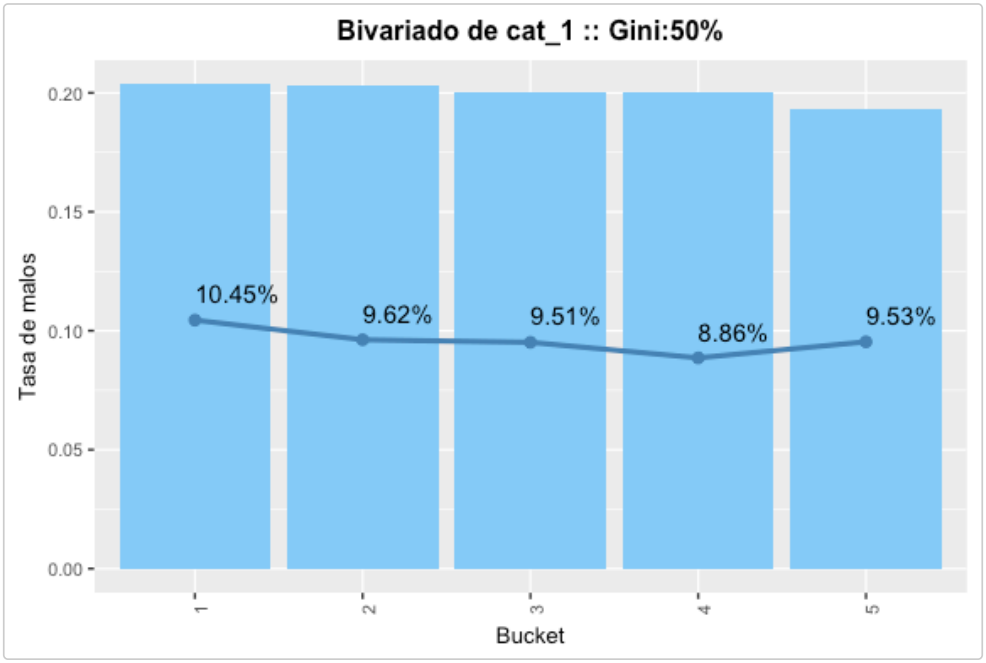


#>
#> $num_5


Bivariado de num_5 :: Gini:22.64%
Tasa de malos
0.60
0.40
0.20
0.00
12.44%
9.17%
4.79%
0.25%
[-27.51,7.11)
[7.11,10.45)
[10.45,15.22)
[15.22,Inf)
Bucket

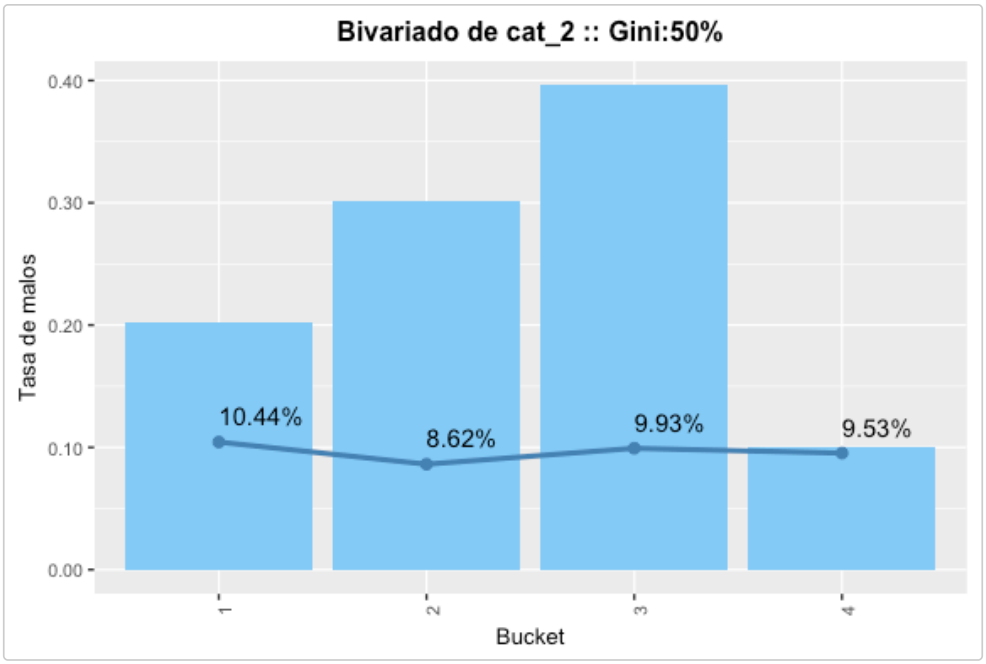
#>

#> $cat_1


Bivariado de cat_1 :: Gini:50%
Tasa de malos
0.20
0.15
0.10
0.05
0.00
10.45%
9.62%
9.51%
8.86%
9.53%
1
2
3
4
5
Bucket

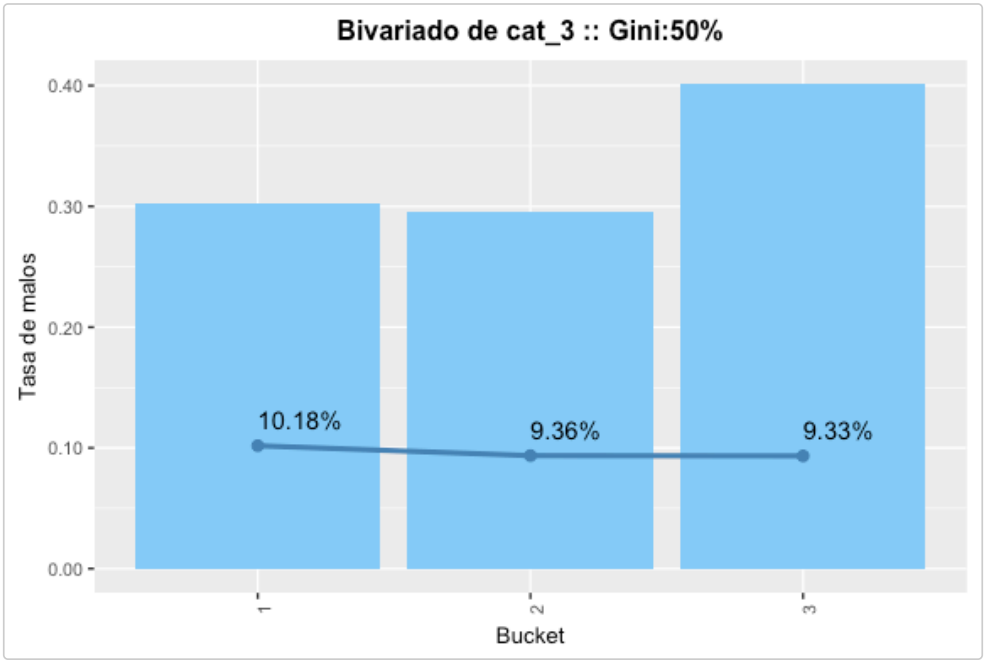
#>

#> $cat_2



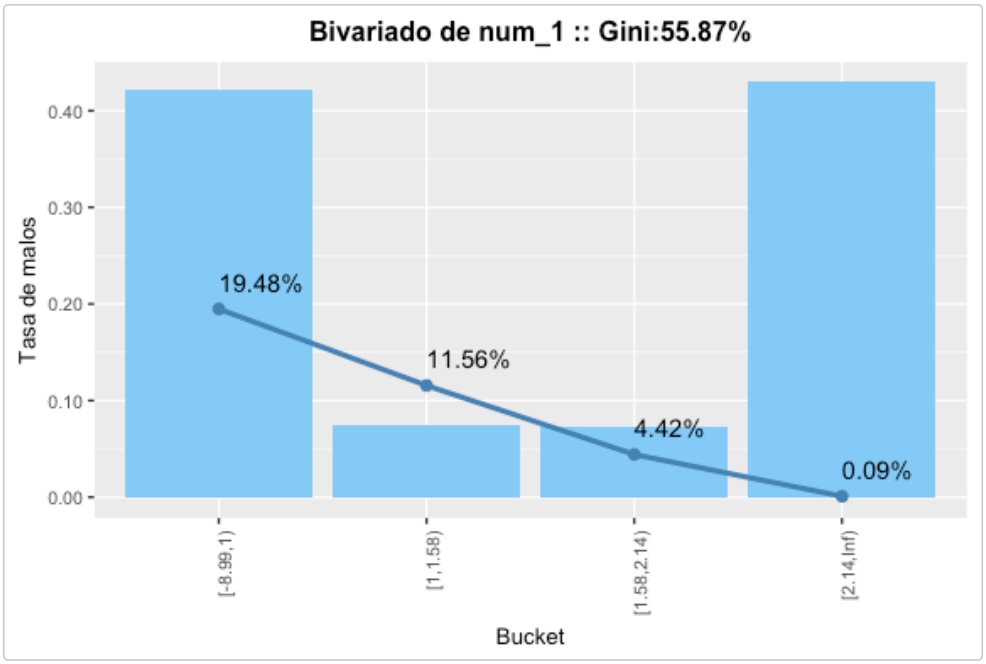
#>

#> $cat_3



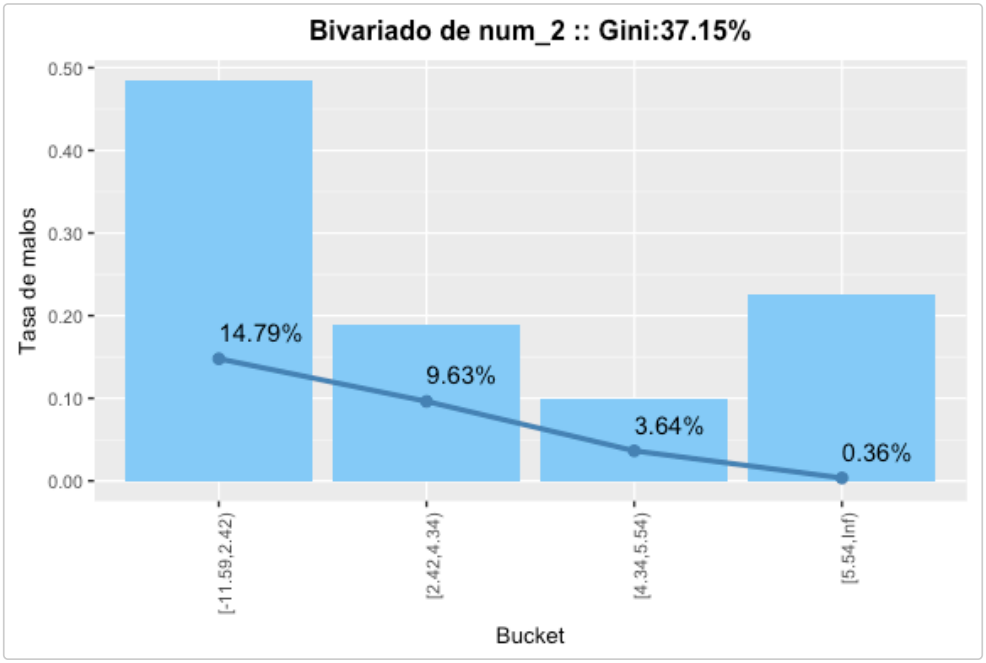

```
biv_2$listaFinal
```

#> $num_1


Bivariado de num_1 :: Gini:55.87%
Tasa de malos
0.40
0.30
0.20
0.10
0.00
19.48%
11.56%
4.42%
0.09%
[-8.99,1)
[1,1.58)
[1.58,2.14)
[2.14,Inf)
Bucket

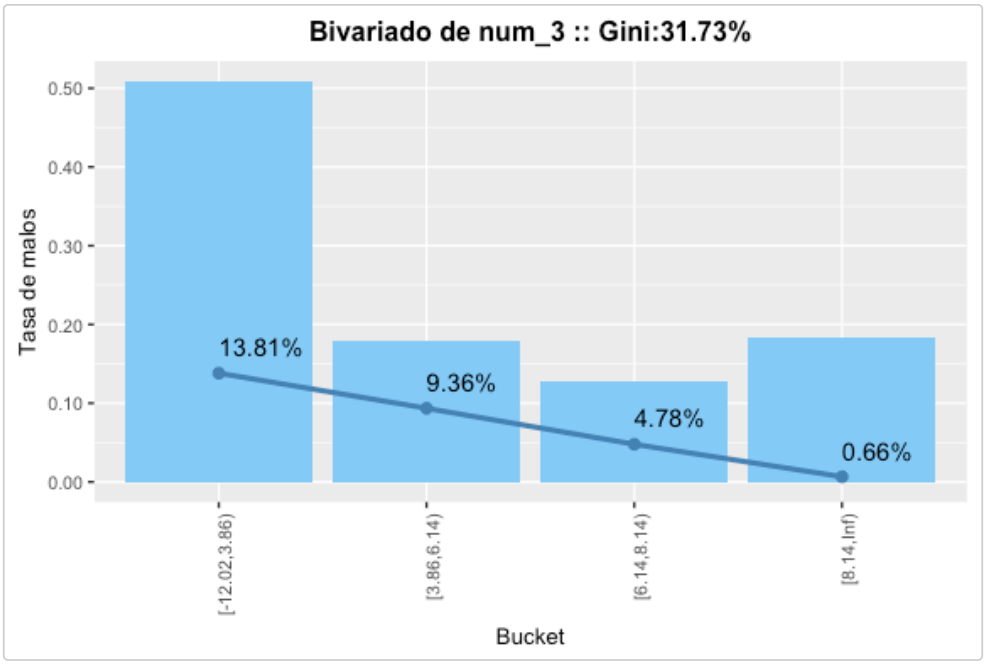
#>

#> $num_2


Bivariado de num_2 :: Gini:37.15%
Tasa de malos
0.50
0.40
0.30
0.20
0.10
0.00
14.79%
9.63%
3.64%
0.36%
[-11.59,2.42)
[2.42,4.34)
[4.34,5.54)
[5.54,Inf)
Bucket

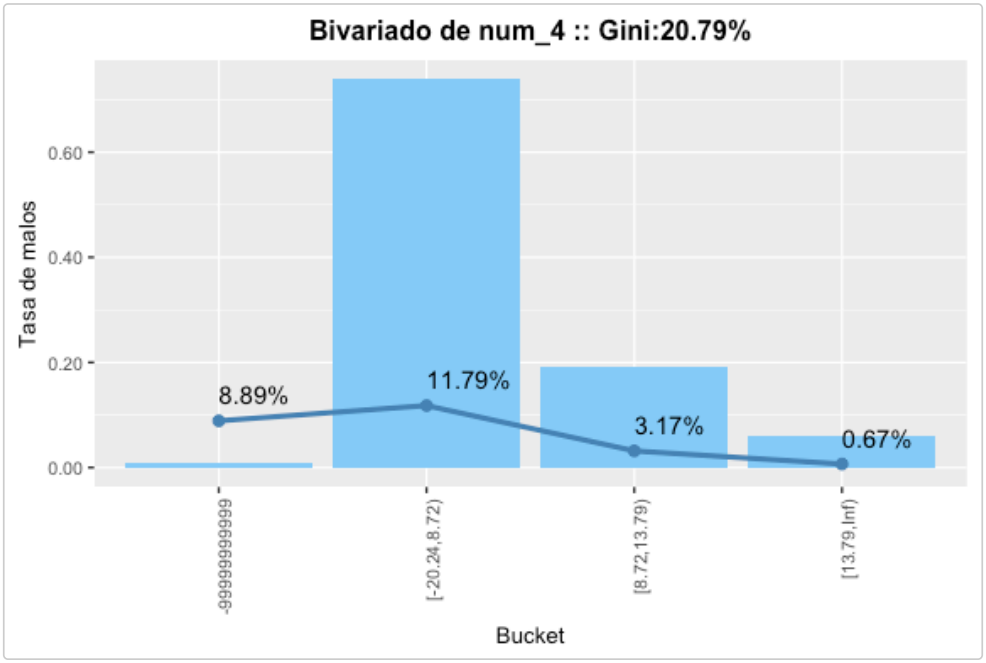
#>

#> $num_3


Bivariado de num_3 :: Gini:31.73%
Tasa de malos
0.50
0.40
0.30
0.20
0.10
0.00
13.81%
9.36%
4.78%
0.66%
[-12.02,3.86)
[3.86,6.14)
[6.14,8.14)
[8.14,Inf)
Bucket

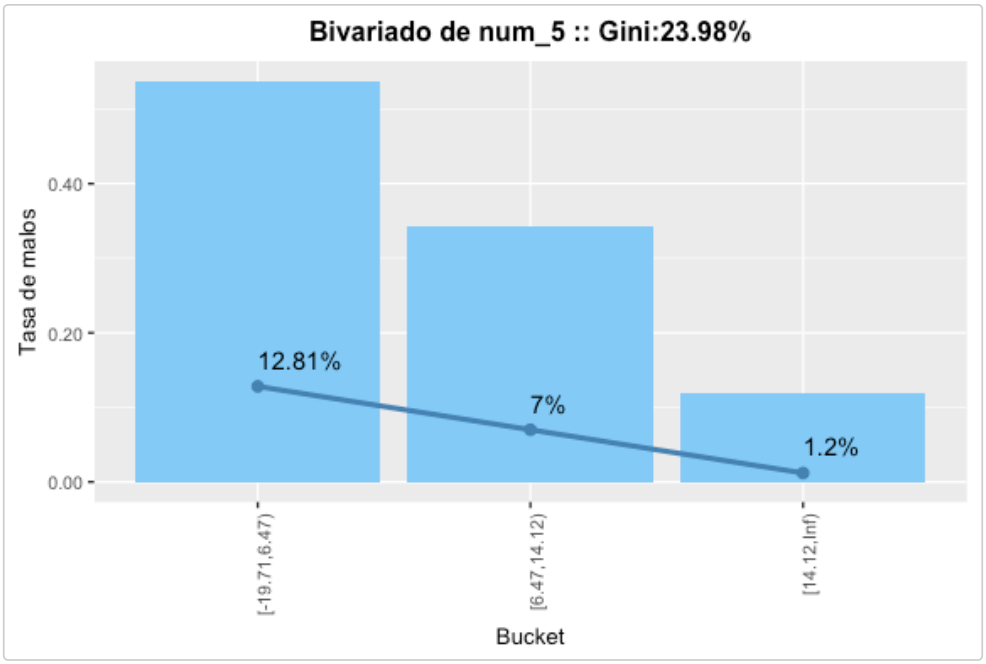
#>

#> $num_4


Bivariado de num_4 :: Gini:20.79%
Tasa de malos
0.60
0.40
0.20
0.00
8.89%
11.79%
3.17%
0.67%
-99999999999
[-20.24,8.72)
[8.72,13.79)
[13.79,Inf)
Bucket

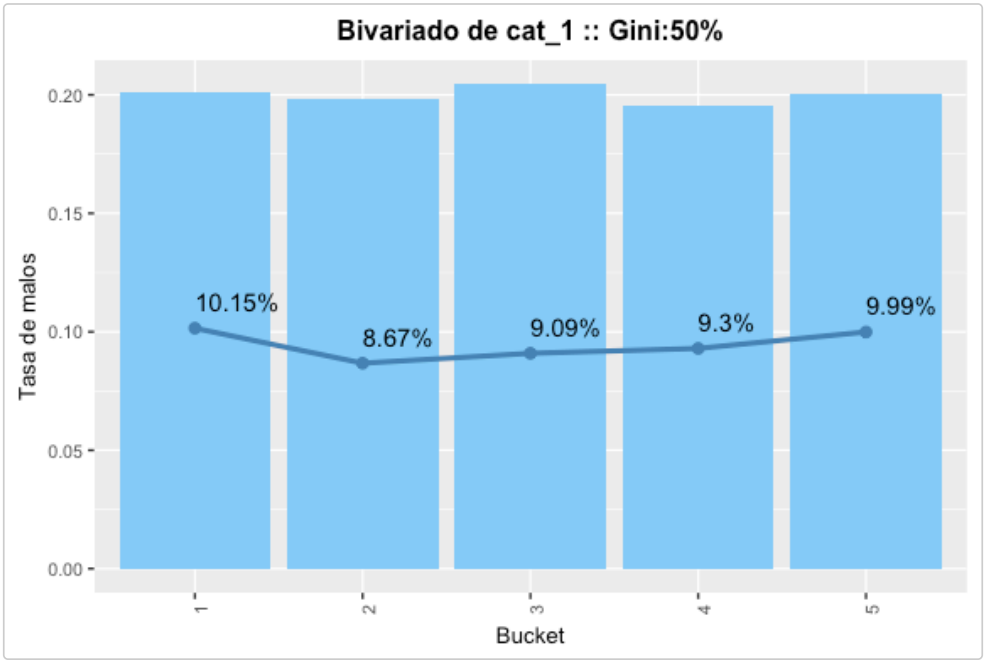
#>

#> $num_5


Bivariado de num_5 :: Gini:23.98%
Tasa de malos
0.40
0.20
0.00
12.81%
7%
1.2%
[-19.71,6.47)
[6.47,14.12)
[14.12,Inf)
Bucket

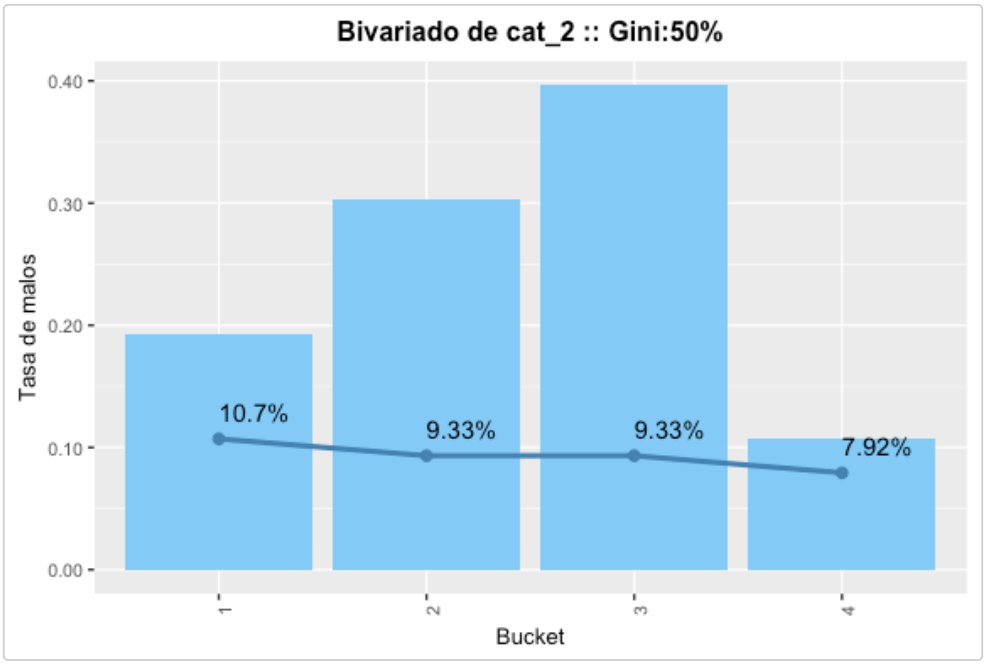
#>

#> $cat_1

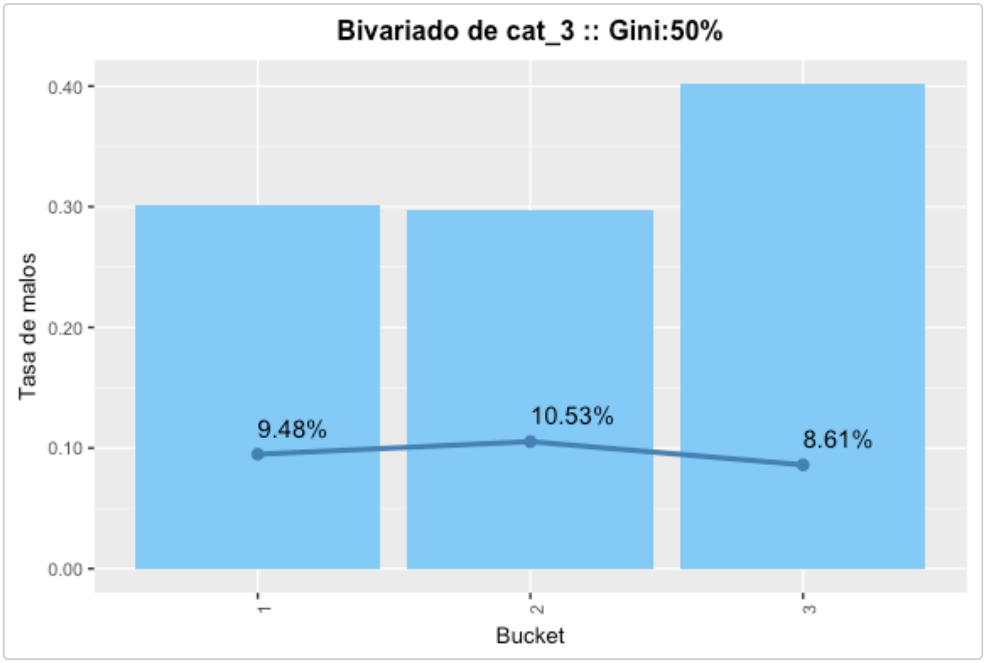

Bivariado de cat_1 :: Gini:50%
Tasa de malos
0.20
0.15
0.10
0.05
0.00
10.15%
8.67%
9.09%
9.3%
9.99%
1
2
3
4
5
Bucket

#>

#> $cat_2




```
#>
#> $cat_3
```



Selección de variables y descripción del modelo

Para la selección de variables se utiliza algoritmos genéticos donde se busca que el conjunto de variables finales no esté correlacionada hasta un 50%, el peso de las variables sea de 5 hasta el 30%, tenga un alto

gini la predicción final, al menos tenga una significancia de 5% de gini las variables y al menos tenga 11 variables.

Es importante recalcar que muchas veces no se va a cumplir estas restricciónes pero se busca que se cumplieran la mayoría.

```
#agregamos los valores especiales a dos de las variables numericas:
library(genalg)
library(pROC)

ag <- list()

for(i in c(1,2)){
    print(paste0("segmento: ", i))
    pob <- mtr8[[paste0("seg_",i)]]$bdwoes
    # variables que se necesitan al final del modelo independientemente de su
    # significancia, correlacion o peso
    vs_forzar <- c("woe_num_1","woe_cat_1")

    set.seed(1000)

    ag[[paste0("seg_",i)]] <- algoritmos_geneticos(datos = pob,
                                                   variables = names(pob)[grep("woe_",names(pob))],
                                                   corr = 0.5, num_vars = 11, sig = 0.05, peso_min =
5,
                                                   peso_max = 30, iter = 10, pobSize=20,
                                                   nom_varobj = "incmpl_d", vars_forzar = vs_forzar)

    # Las mejores variables que cumplen con estas reestricciones
    ag[[paste0("seg_",i)]]$vars_modelo

}
#> [1] "segmento: 1"
#> [1] "inicializando ag"
#> Testing the sanity of parameters...
#> Not showing GA settings...
#> Starting with random values in the given domains...
#> Starting iteration 1
#> Calucating evaluation values... ..................... done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 0 mutations applied
#> Starting iteration 2
#> Calucating evaluation values... ................ done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
```

```
#>   applying mutations... 2 mutations applied
#> Starting iteration 3
#> Calucating evaluation values... ............ done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 0 mutations applied
#> Starting iteration 4
#> Calucating evaluation values... ............... done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 0 mutations applied
#> Starting iteration 5
#> Calucating evaluation values... ............... done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 0 mutations applied
#> Starting iteration 6
#> Calucating evaluation values... ................ done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 2 mutations applied
#> Starting iteration 7
#> Calucating evaluation values... ............... done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 1 mutations applied
#> Starting iteration 8
#> Calucating evaluation values... ............ done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 1 mutations applied
#> Starting iteration 9
#> Calucating evaluation values... ................ done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 1 mutations applied
#> Starting iteration 10
```

```
#> Calucating evaluation values... .......... done.
#> [1] "terminando ag"
#> [1] "segmento: 2"
#> [1] "inicializando ag"
#> Testing the sanity of parameters...
#> Not showing GA settings...
#> Starting with random values in the given domains...
#> Starting iteration 1
#> Calucating evaluation values... ................... done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 0 mutations applied
#> Starting iteration 2
#> Calucating evaluation values... ............... done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 1 mutations applied
#> Starting iteration 3
#> Calucating evaluation values... ............ done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 0 mutations applied
#> Starting iteration 4
#> Calucating evaluation values... ............. done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 0 mutations applied
#> Starting iteration 5
#> Calucating evaluation values... .................. done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 0 mutations applied
#> Starting iteration 6
#> Calucating evaluation values... ................ done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 2 mutations applied
#> Starting iteration 7
#> Calucating evaluation values... ............... done.
```

```
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 1 mutations applied
#> Starting iteration 8
#> Calucating evaluation values... ..... done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 1 mutations applied
#> Starting iteration 9
#> Calucating evaluation values... ..... done.
#> Creating next generation...
#>   sorting results...
#>   applying elitism...
#>   applying crossover...
#>   applying mutations... 1 mutations applied
#> Starting iteration 10
#> Calucating evaluation values... ..... done.
#> [1] "terminando ag"
```

Ya que tenemos las mejores variables para el mejor modelo, realizamos la regresión logística para poder reportar la correlación, los pesos, los coeficientes de cada variable y el gini del modelo.

```
#agregamos los valores especiales a dos de las variables numericas:

evalFinal<-list()

for(i in c(1,2)){
    vars <- ag[[paste0("seg_",i)]]$vars_modelo

    evalFinal[[paste0("seg_",i)]] <- evaluacionFinal(data = mtr8[[paste0("seg_",i)]]$bdwoes,
                                                     vars = vars,
                                                     nom_varobj = "incmpl_d")

}

evalFinal$seg_1$gini
#> [1] 0.753
evalFinal$seg_1$coeficientes
#>             Estimate Std. Error z value
#> (Intercept)  -2.2313    0.05937 -37.584
#> woe_num_2    -1.0029    0.07748 -12.945
#> woe_num_5    -0.9749    0.10337  -9.431
#> woe_num_1    -0.9999    0.05873 -17.023
#> woe_cat_1    -1.2412    0.67847  -1.829
#>                                                                    Pr(>|z|)
```

```
#> (Intercept) 0.000000000000000000000000000000000000000000000000000000000000000000000000000000
#> woe_num_2   0.000000000000000000000000000000000000000025128229591706469805628832183 78
#> woe_num_5   0.000000000000000000004048784236209372784867163746242892140411599873 94
#> woe_num_1   0.000000000000000000000000000000000000000000000000000000000000000000000005498
#> woe_cat_1   0.067330118970938140687820805396768264472484588623046875000000000000000
evalFinal$seg_1$pesos
#> woe_num_2 woe_num_5 woe_num_1 woe_cat_1
#>    38.255    21.306    39.036     1.403
evalFinal$seg_1$correlaciones
#>           woe_num_2 woe_num_5 woe_num_1 woe_cat_1
#> woe_num_2  1.000000  0.026957  0.041806  0.002474
#> woe_num_5  0.026957  1.000000  0.031586 -0.009619
#> woe_num_1  0.041806  0.031586  1.000000  0.003513
#> woe_cat_1  0.002474 -0.009619  0.003513  1.000000

evalFinal$seg_2$gini
#> [1] 0.7669
evalFinal$seg_2$coeficientes
#>             Estimate Std. Error z value   Pr(>|z|)
#> (Intercept)   -2.276    0.09290 -24.497 1.602e-132
#> woe_num_2     -1.039    0.10486  -9.906  3.931e-23
#> woe_num_5     -1.077    0.12697  -8.481  2.240e-17
#> woe_num_1     -1.014    0.09859 -10.290  7.851e-25
#> woe_cat_1     -1.101    0.84253  -1.307  1.911e-01
evalFinal$seg_2$pesos
#> woe_num_2 woe_num_5 woe_num_1 woe_cat_1
#>    31.013    16.225    51.389     1.374
evalFinal$seg_2$correlaciones
#>           woe_num_2 woe_num_5 woe_num_1 woe_cat_1
#> woe_num_2   1.00000   0.03571  0.054680  0.012805
#> woe_num_5   0.03571   1.00000  0.030727  0.018912
#> woe_num_1   0.05468   0.03073  1.000000 -0.005112
#> woe_cat_1   0.01280   0.01891 -0.005112  1.000000
```

Validación de Test

Despues de tener el modelo vamos a observar su comportamiento en la prueba test

```
segmentos_test <- list("1" = d_test %>% filter(segmento == 1),
                       "2" = d_test %>% filter(segmento == 2))

n_segmentos <- 2

for(i in seq(n_segmentos)){

    base <- segmentos_test[[paste0(i)]]
    lmtr5 <- mtr8[[paste0("seg_",i)]]$lmtr5r
    modelo <- evalFinal[[paste0("seg_",i)]]$modelo
```

```
    evalfinaltrain <- evalFinal[[paste0("seg_",i)]]


    # Calificamos la base test, tomamos las predicciones y devolvemos una comparacion de prueba y
test
    GiniTestTrain <- comparacionTestTrain(data = base,
                                          lmtr5 = lmtr5,
                                          vob = "incmpl_d",
                                          modelo = modelo,
                                          evalfinalTrain = evalfinaltrain)

    if(i == 1){
        resumenGini<-GiniTestTrain
    }else{
    resumenGini <- rbind.data.frame(resumenGini, GiniTestTrain)
    }
}
#> [1] "num_5"
#> [1] "num_4"
#> [1] "num_3"
#> [1] "num_2"
#> [1] "cat_1"
#> [1] "cat_2"
#> [1] "cat_3"
#> [1] "num_1"
#> [1] "num_5"
#> [1] "num_4"
#> [1] "num_3"
#> [1] "num_2"
#> [1] "cat_1"
#> [1] "cat_2"
#> [1] "cat_3"
#> [1] "num_1"

knitr::kable(resumenGini)
```

	giniTrain	giniTest
2	0.7530	0.7415
21	0.7669	0.7292

Transformación a puntos

A continuación se muestra la tabla de puntuación que sigue la siguiente transformación donde el score de referencia es 500, la pdo es 20 y los odds son 3:1.

$$N = Numero\ de\ variables\ utilizadadas\ en\ la\ regresion$$

$$B = \frac{20}{ln(2)} * ln(odds)$$

$$A = 500 - \beta_0 B$$

$$S_i = \frac{A}{N} - WoE_i \beta_i B$$

$$ScoreFinal = \sum_{i=1}^{N} S_i$$

segmento 1

num_1

pobl	tm	tipo bucket	Rango	Puntos
3510	0.1943	normal	[-9.3154,0.7871)	117
1067	0.1368	normal	[0.7871,1.6987)	130
464	0.0517	normal	[1.6987,2.1184)	164
3981	0.0035	normal	[2.1184,Inf)	251

num_2

pobl	tm	tipo bucket	Rango	Puntos
4986	0.148	normal	[-10.2949,3.129)	127
545	0.0899	normal	[3.129,3.6921)	145
1396	0.0487	normal	[3.6921,5.5247)	166
670	0.0149	normal	[5.5247,6.5644)	205
1425	0.0007	normal	[6.5644,Inf)	302

num_5

pobl	tm	tipo bucket	Rango	Puntos
5235	0.1244	normal	[-27.5136,7.1139)	134
1582	0.0917	normal	[7.1139,10.4548)	144
1419	0.0479	normal	[10.4548,15.2162)	166
786	0.0025	normal	[15.2162,Inf)	258

cat_1

pobl	tm	tipo bucket	Rango	Puntos
1838	0.1045	normal	1	139
1829	0.0962	normal	2	143

1808	0.0951	normal	3	143
1806	0.0886	normal	4	146
1741	0.0953	normal	5	143

segmento 2

num_1

pobl	tm	tipo bucket	Rango	Puntos
2084	0.1948	normal	[-8.995,0.9967)	116
372	0.1156	normal	[0.9967,1.5814)	136
362	0.0442	normal	[1.5814,2.1356)	169
2128	0.0009	normal	[2.1356,Inf)	294

num_2

pobl	tm	tipo bucket	Rango	Puntos
2401	0.1479	normal	[-11.5863,2.4204)	126
935	0.0963	normal	[2.4204,4.3441)	142
494	0.0364	normal	[4.3441,5.5446)	176
1116	0.0036	normal	[5.5446,Inf)	254

num_5

pobl	tm	tipo bucket	Rango	Puntos
2661	0.1281	normal	[-19.7115,6.4688)	131
1700	0.07	normal	[6.4688,14.1213)	154
585	0.012	normal	[14.1213,Inf)	217

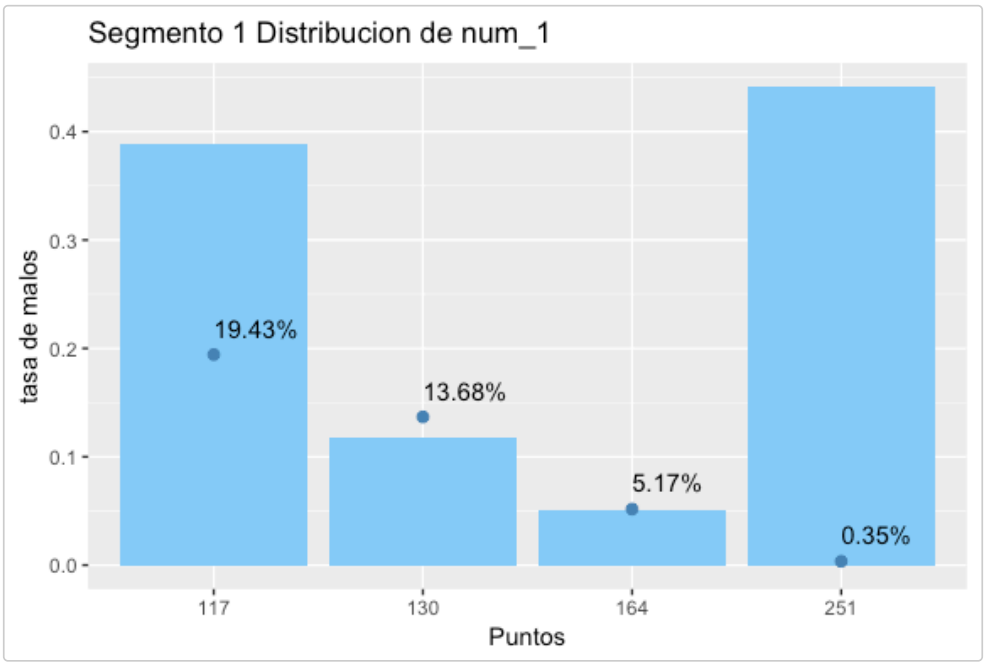
cat_1

pobl	tm	tipo bucket	Rango	Puntos
995	0.1015	normal	1	140
980	0.0867	normal	2	146
1012	0.0909	normal	3	144
968	0.093	normal	4	144
991	0.0999	normal	5	141

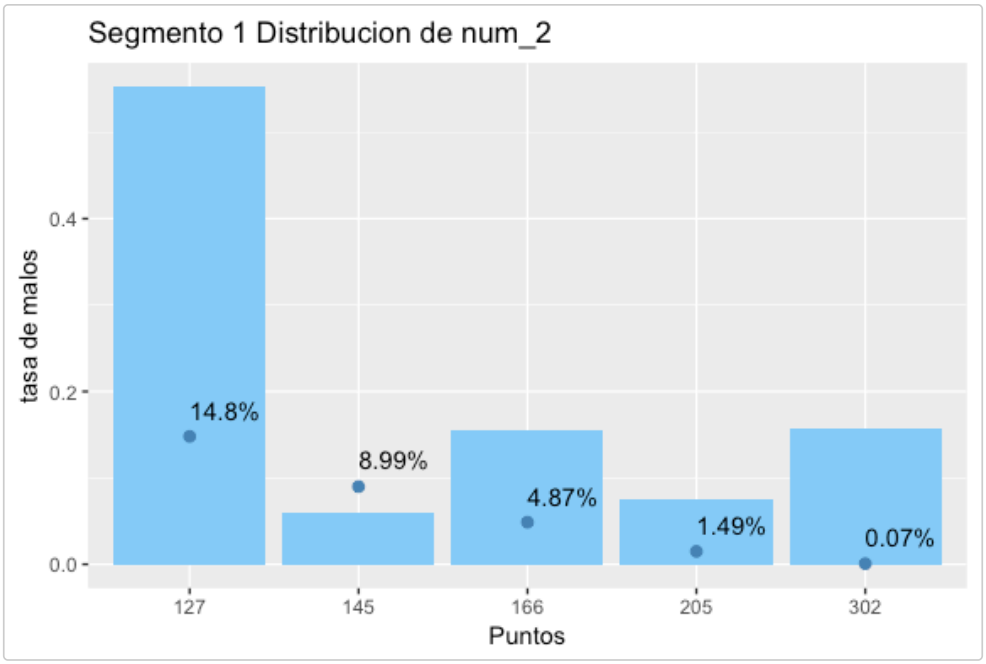
Distribución de puntos por variable y segmento

A continuación se muestra la tabla de puntuación que sigue la siguiente transformación:

```
listaGraf <- distribScoreVar(segmentos = 2, scorecards = listaScoreCards)

listaGraf
#> $`segmento 1`
#> $`segmento 1`$num_1
```

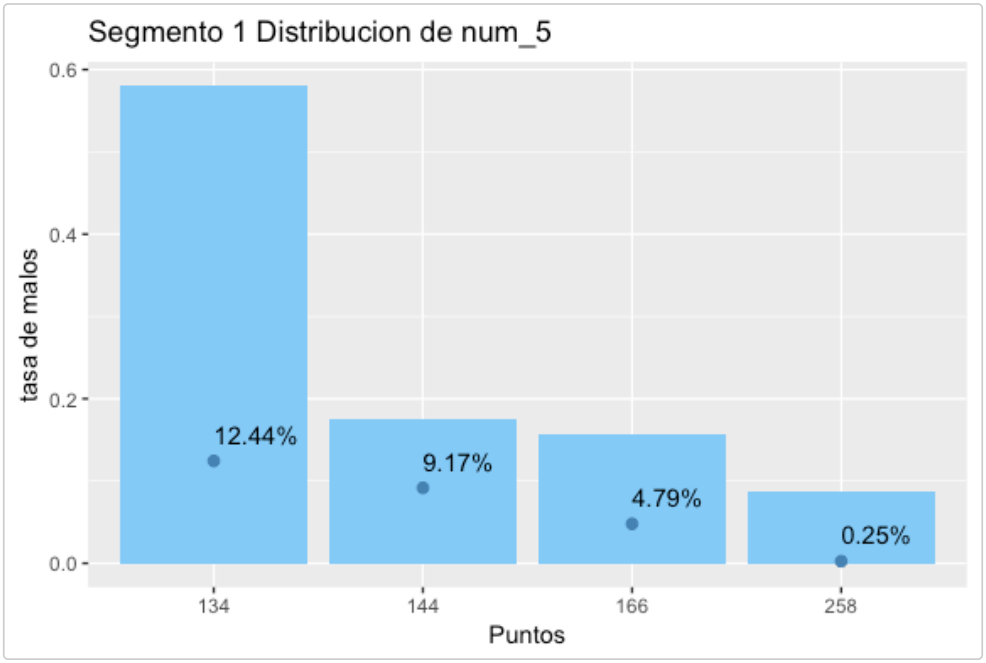



```
#>
#> $`segmento 1`$num_2
```



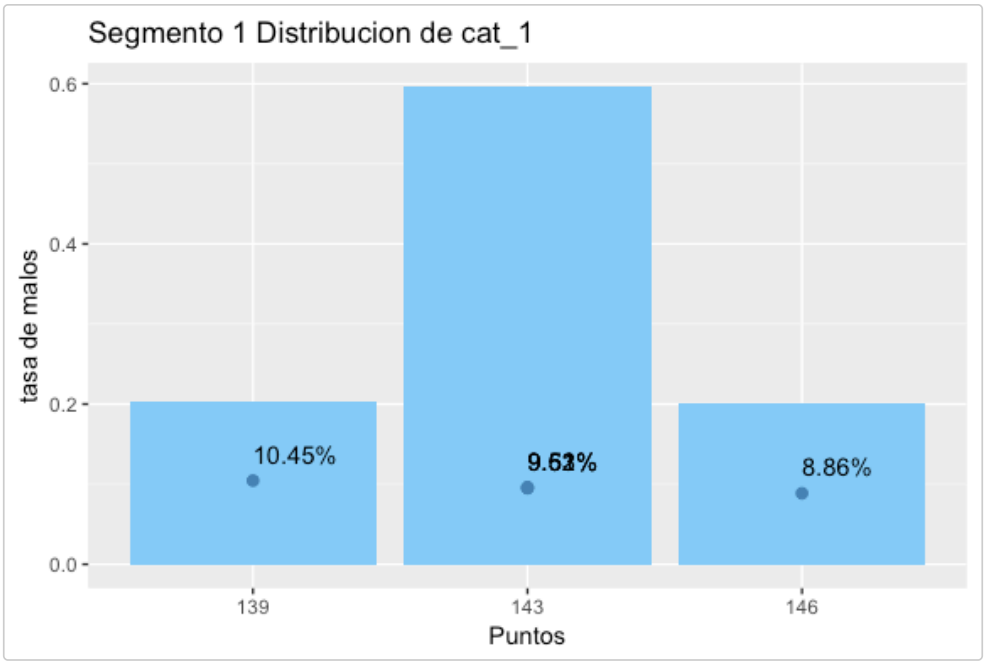
#>

#> $`segmento 1`$num_5

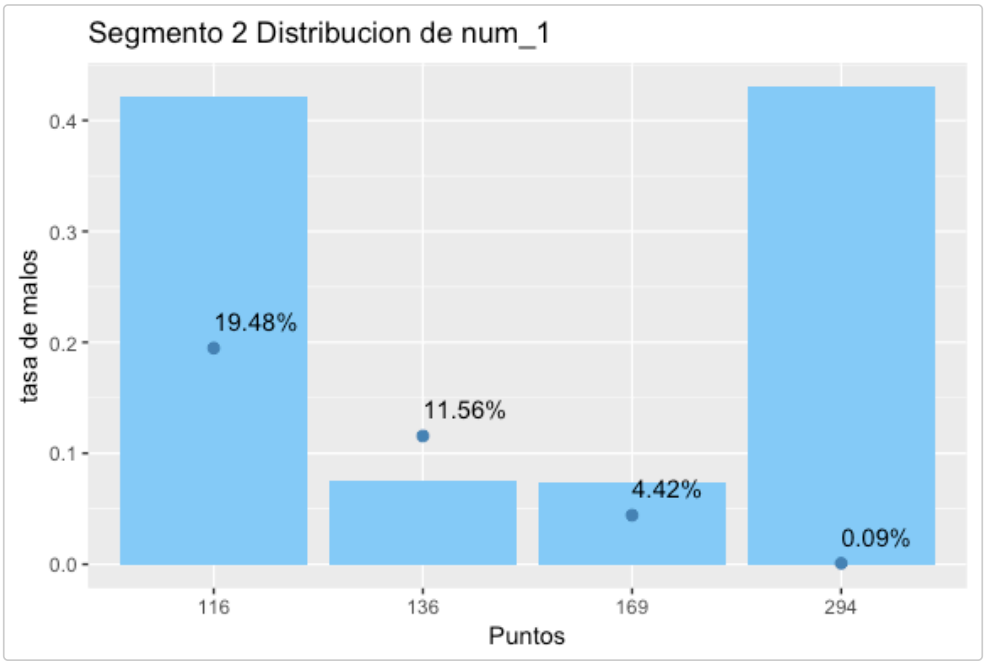


#>

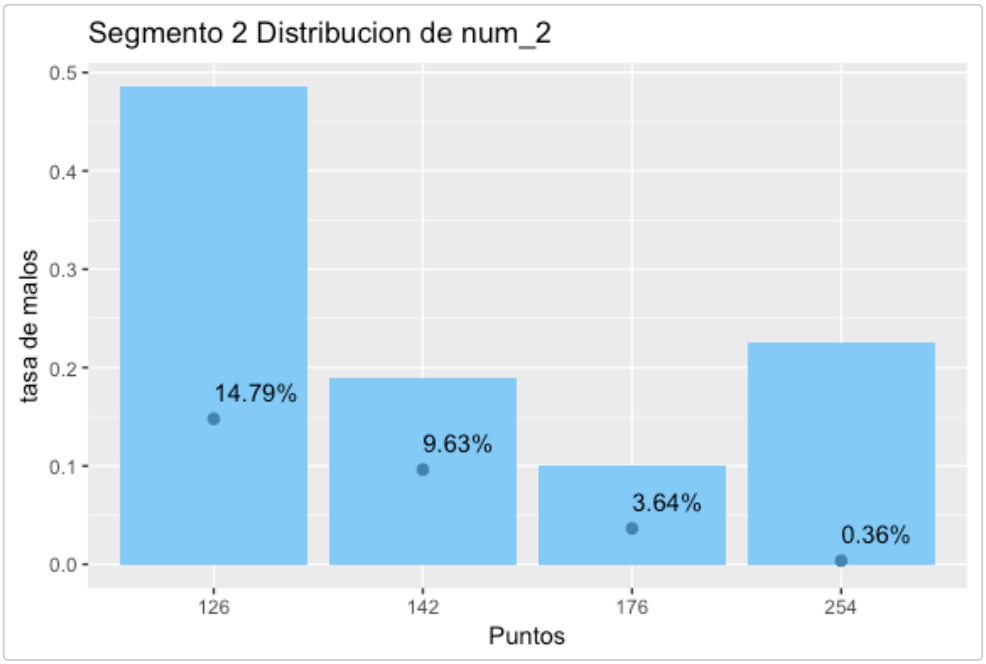
#> $`segmento 1`$cat_1



#>
#>
#> $`segmento 2`
#> $`segmento 2`$num_1

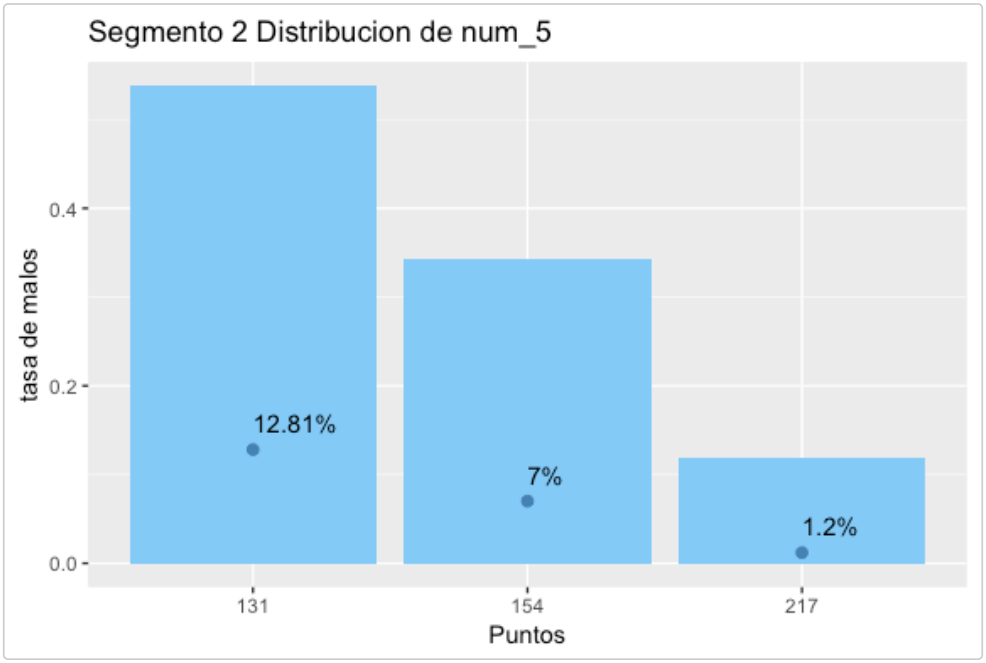


#>
#> $`segmento 2`$num_2



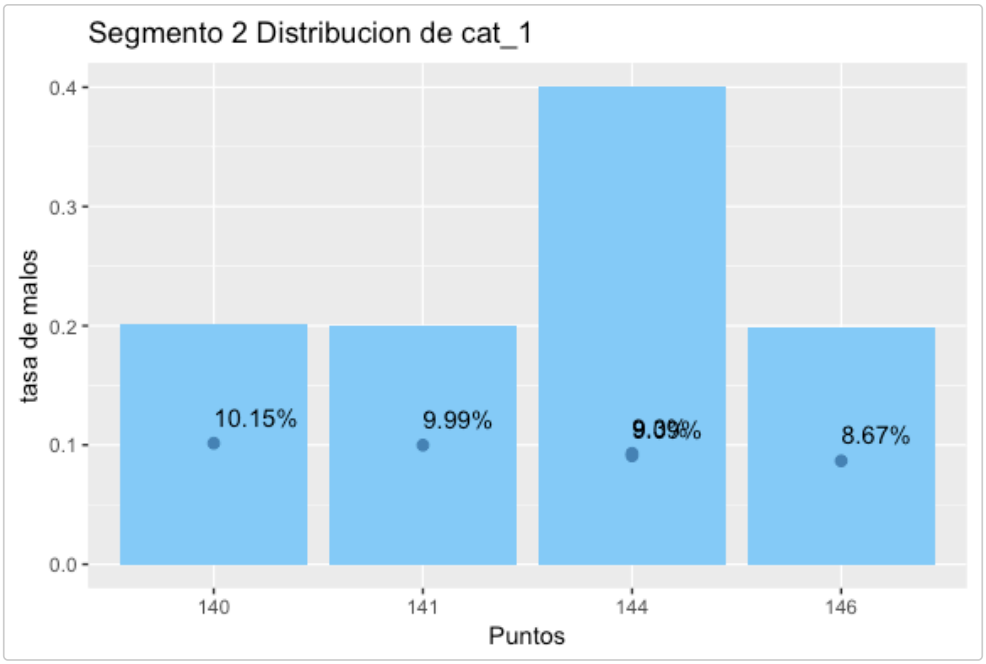
#>

#> $`segmento 2`$num_5



#>

#> $`segmento 2`$cat_1


Segmento 2 Distribucion de cat_1
tasa de malos
0.4
0.3
0.2
0.1
0.0
10.15%
9.99%
8.67%
140
141
144
146
Puntos