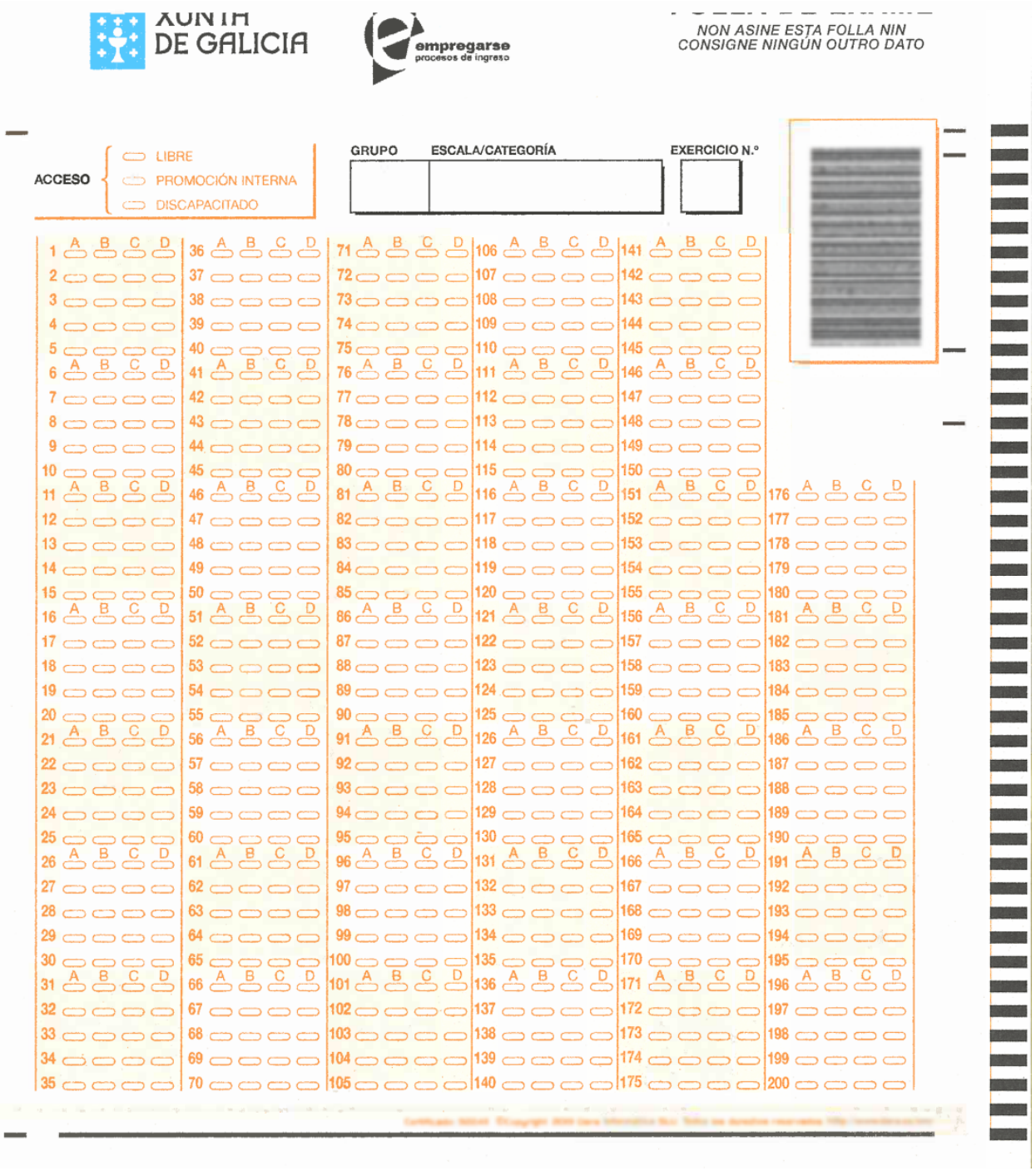
CORRECCIÓN DE EXÁMENES (TIPO TEST) EN OPOSICIONES

Marcos Fernández Arias

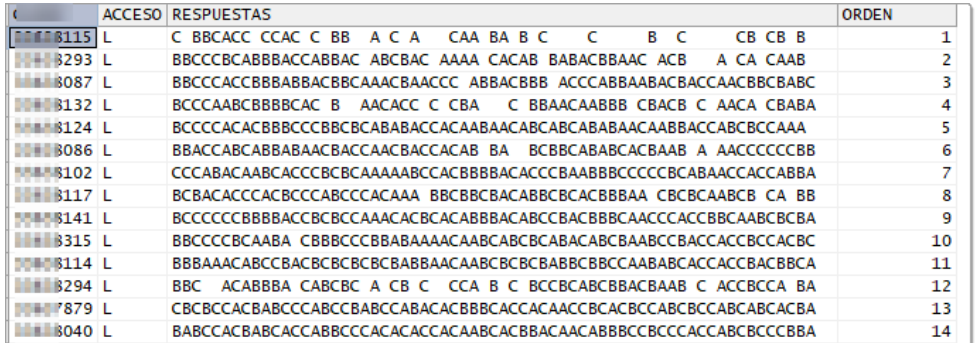
XUNTA DE GALICIA

scanner OMR



hoja de examen

fichero obtenido
al leer las hojas de examen



[redacted]	ACCESO	RESPUESTAS	ORDEN
[redacted]115	L	C BBCACC CCAC C BB A C A CAA BA B C C B C CB CB B	1
[redacted]293	L	BBCCCBCABBBACCABBAC ABCBAC AAAA CACAB BABACBBAAC ACB A CA CAAB	2
[redacted]087	L	BBCCCACCBBBABBACBBCAAACBAACCC ABBACBBB ACCCABBAABACBACCAACBBCBABC	3
[redacted]132	L	BCCCAABCBBBBCAC B AACACC C CBA C BBAACAABBB CBACB C AACA CBABA	4
[redacted]124	L	BCCCCACACBBBCCCBBCBCABABACCACAABAACABCABCABABAACAABBACCABCBCCAAA	5
[redacted]086	L	BBACCABCABBABAACBACCAACBACCACAB BA BCBBCABABCACBAAB A AACCCCCCBB	6
[redacted]102	L	CCCABACAABCACCCBCBCAAAAABCCACBBBBACACCCBAABBBCCCCCBCABAACCACCABBA	7
[redacted]117	L	BCBACACCCACBCCCABCCCACAAA BBCBBCBACABBCBCACBBBAA CBCBCAABCB CA BB	8
[redacted]141	L	BCCCCCCBBBBACCBCBCCAAACACBCACABBBACABCCBACBBBCAACCCACCBBCAABCBCBA	9
[redacted]315	L	BBCCCCBCAABA CBBBCCCBBABAAAACAABCABCBCABACABCBAABCCBACCACCBCCACBC	10
[redacted]114	L	BBBAAACABCCBACBCBCBCBABBAACAABCBCBCBABBCBBCCAABABCACCACCBACBBCA	11
[redacted]294	L	BBC ACABBBA CABCBC A CB C CCA B C BCCBCABCBBACBAAB C ACCBCCA BA	12
[redacted]879	L	CBCBCCACBABCCCABCCBABCCABACACBBBCACCACAACCBCACBCCABCBCCABCABCACBA	13
[redacted]040	L	BABCCACBABCACCABBCCCACACACCACAABCACBBACAACABBBCCBCCCACCABCBCCCBBA	14

n_lectura_hoja	barcode_hoja	RESPUESTAS
1	[redacted]346	BADAABA.BD..DBBC.CCCDDBD.D.AABCDD.AD..DDDDD.BBBAA.ACABABDDCA
2	[redacted]731	BADAABAABCAACDCCCACDBDBAADCBBABDA.BDBAACDDABBCCAA.ABDBDCDCCA
3	[redacted]350	BADAACACBA..DBBC.DDCCDBDBDBABCCDB.ACDBDCCDDDCBBAA.AABAABDCAA
4	[redacted]382	BADAABACBC.....CD.DCDCBDABACBBCDB.ADCBDCDDD.CBBAA..CABAAC.DA
5	[redacted]381	BADC.BAABC.....CADDC.AADDDC.DAADB.ABDD.CDD.....A.........D.AB
6	[redacted]399	BADAAB.CBC..DB.CADDCCDBD.D.CDBCDB..DCBDCCADCCB.AA.ADAABBC.BA
7	[redacted]626	BCDCAADBCADABBBCADDCAADDDDDCDCC.C.BCB.ADAA.DAAAAC.BDDBAAD.AB
8	[redacted]639	BADAABACBCBBDBBCA.DCCDBDDDACDBCDB.BDCBDCDADCCBCAA.AAABABC.BA
9	[redacted]558	BCDAABACBC..D..CABDCCDBDDDACABCDC.ADC.DCDADCBB.AA.........A

[redacted]	L	BBCBCBBCABCACBAABBBBACCBBACCCBBBBBCBBCBBCCBABBAC ACBABCC CBCCBABA	47
[redacted]915	L	BBCCCABCBBBABBCCBBCAACCBAA ACAA BB BCAACAABBBAA AABAC A CB CBAB	48
[redacted]911	L	ABCACACABBBACCACACCCB CBACBCABB BA BBCCACABAABAA ACB C AACBACABCA	49
[redacted]840	L	ABCACACCAACACCCBCCCCABABABBACABCAACBCBCBAACBCACABCABACCABCCCABCBB	50
[redacted]877	L	CBCACACCBBBACCACAACBBCCBCBCCABACABCABCCACCBBBCACAAAAACCACCBACACBA	51
[redacted]	[redacted]	[redacted]ACBBBBCBBCABCCAABCABCCBCABCBAC[redacted]ACBCCCBCCCAAB	52

```
582
583 # La lectura del fichero DBF se realiza mediante la librería DBF de Python
584 # debido a que las otras erróneamente realizan un trim implícito que estropea
585 # la columna RESPUESTAS si contiene espacios en blanco al principio.
586
587 # pacman::p_load(reticulate)
588 py_config()
589 # repl_python()
590 # py_install("pandas")
591 # py_install("dbfread")
592 # py_install("dbf")
593 # py_install("simpledbf", pip = TRUE)
594
595 stopifnot(reticulate::py_module_available("pandas"))
596 stopifnot(reticulate::py_module_available("dbf"))
597
598 leer_dbf <- function(fname) {
599   if(!fs::file_exists(fname)) {
600     stop("No se ha encontrado el fichero ", fname)
601   }
602   py_env2 <- py_run_string(str_c(
603 "from dbfread import DBF
604 from pandas import DataFrame
605 df = DataFrame(iter(DBF('", fname, "')))
606 "))
607   py_env2$df %>%
608     verify(nrow(.) > 0) %>%
609     as_tibble()
610 }
```

```
598 leer_dbf <- function(fname) {
599   if(!fs::file_exists(fname)) {
600     stop("No se ha encontrado el fichero ", fname)
601   }
602   py_env2 <- py_run_string(str_c(
603 "from dbfread import DBF
604 from pandas import DataFrame
605 df = DataFrame(iter(DBF('", fname, "')))
606 "))
607   py_env2$df %>%
608     verify(nrow(.) > 0) %>%
609     as_tibble()
610 }
```

plantilla de corrección
(en formato Excel)

	A	B	C	D
1	num_pregunta	respuesta	categ_inicial	categ_definitiva
2	1	A	general	general
3	2	B	general	general
4	3	B	general	general
5	4	A	general	general
6	5		general	anulada
7	6		general	anulada
8	7	C	general	general
9	8	D	general	general
10	9	B	general	general

num_pregunta	respuesta	categ_inicial	categ_definitiva
63	B	general	general
64	A	general	general
65	D	general	general
66	B	general	general
67		general	anulada
68	B	general	general
69		general	anulada
70	C	general	general
71	B	general	general
72	A	general	general
73	A	general	general
74	B	general	general
75	C	general	general
76	D	general	general
77	D	general	general
78	D	general	general
79	C	general	general
80	C	general	general
81	B	reserva general	reserva general
82	A	reserva general	reserva anulada
83	D	reserva general	reserva anulada
84	A	reserva general	reserva general
85	A	reserva general	reserva general
86	A	reserva general	reserva no usada

manana | tarde | covid

```r
  for (acceso in accesos) {
    for (turno in turnos) {
      #acceso <- if_else(length(accesos)>1, acceso, "")

      sheetname <- str_c(acceso,
                         if_else(acceso != "" &
                                   turno != "", "_", ""),
                         turno)
      if (length(excel_sheets(fname_plantillas_xlsx)) == 1) {
        sheetname <- excel_sheets(fname_plantillas_xlsx)
      } else if (!sheetname %in% excel_sheets(fname_plantillas_xlsx)) {
        next
      }
      plantillas_0 <-
        readxl::read_excel(fname_plantillas_xlsx, sheet = sheetname) %>%
        verify(
          names(.) == c(
            "num_pregunta",
            "respuesta",
            "categ_inicial",
            "categ_definitiva",
            "observaciones"
          )
        ) %>%
        verify(categ_inicial != "" & categ_definitiva != "") %>%
        mutate(
          num_pregunta = as.integer(num_pregunta),
          acceso = !!acceso,
          turno = !!turno,
          across(-num_pregunta, as.character),
          across(-num_pregunta, ~ replace_na(.x, "")),
          respuesta = if_else(
            str_detect(categ_definitiva, "anulada|no utilizada|no usada"),
            " ",
            str_to_upper(respuesta)
          ),
          across(
```

```r
  plantillas_0 <-
    readxl::read_excel(fname_plantillas_xlsx, sheet = sheetname) %>%
    verify(
      names(.) == c(
        "num_pregunta",
        "respuesta",
        "categ_inicial",
        "categ_definitiva",
        "observaciones"
      )
    ) %>%
    verify(categ_inicial != "" & categ_definitiva != "") %>%
    mutate(
      num_pregunta = as.integer(num_pregunta),
      acceso = !!acceso,
      turno = !!turno,
      across(-num_pregunta, as.character),
      across(-num_pregunta, ~ replace_na(.x, "")),
      respuesta = if_else(
```

```r
leer_fichero_examen_CSV <- function(fname) {
  expect_true(length(fname) == 1)
  expect_true(file_exists(fname))
  expect_true(str_detect(fname, regex("\\.CSV$", ignore_case = TRUE)))

  readr::read_delim(
    fname,
    delim = ";",
    col_types = cols(
      CLAVE = col_integer(),
      ACCESO = col_character(),
      RESPUESTAS = col_character(),
      ORDEN = col_integer()
    )
  ) %>%
    verify(names(.) == c("CLAVE", "ACCESO", "RESPUESTAS", "ORDEN")) %>%
    verify(nrow(.) %in% 1:14999) %>%
    select(n_lectura_hoja = ORDEN, barcode_hoja = CLAVE, RESPUESTAS) %>%
    verify(!duplicated(barcode_hoja)) %>%
    verify(!duplicated(n_lectura_hoja)) %>%
    verify(n_lectura_hoja %in% 1:14999) %>%
    verify(RESPUESTAS == str_trim(RESPUESTAS, "right")) %>%
    verify(str_detect(RESPUESTAS, "^[ABCD \\?]*$"))
}
```

```
1794    datos %>%
1795      filter(!is.na(!!plantilla)) %>%
1796      select(barcode_hoja, acceso, turno,  RESPUESTAS, !!plantilla) %>%
1797      rowwise() %>%
1798      mutate(
1799        nresp_plantilla = str_count(!!plantilla, "[ABCD]"),
1800        letras_plantilla_correccion = !!plantilla %>%
1801          str_split(., pattern = "") %>%
1802          first() %>%
1803          list()
1804        ,
1805        RESPUESTAS = str_pad(RESPUESTAS, width = nresp_plantilla,
1806                             side = "right", pad = " "),
1807        respuestas_a_corregir =
1808          RESPUESTAS %>%
1809          str_sub(1, str_length(!!plantilla)) %>%
1810          str_pad(str_length(!!plantilla), "right") %>%
1811          str_split(., pattern = "") %>%
1812          first() %>%
1813          list()
1814        ,
1815        aciertos = map2(respuestas_a_corregir,
1816                        letras_plantilla_correccion,
1817                        function(x, y)
1818                          y != " " & x == y) %>%
1819          unlist() %>%
1820          sum(),
1821        fallos = map2(respuestas_a_corregir,
```

```r
  # dependiendo si el cupo abarca a todos los candidatos o sólo a los de libre
  if (!is.null(names(n_personas_corte_cupo))) {
    if (names(n_personas_corte_cupo) == "libre") {
      resultados_0 <- resultados_0 %>%
        verify(nrow(.) >= n_personas_corte_cupo) %>%
        mutate(
          rank_libre = rank( (acceso %in% c("L", "D")) *
            -1 * (
              supera_parte_general * supera_parte_especifica * supera_corte_global
              * resp_r + (resp_r / 1000)
            ),
            ties.method = "min"
          ))
      corte_global_segun_cupo <- resultados_0 %>%
        filter(acceso %in% c("L", "D")) %>%
        arrange(rank_libre) %>%
        filter(rank_libre <= n_personas_corte_cupo) %>%
        tail(1) %>%
        pull(resp_r)
      names(corte_global_segun_cupo) <- "libre"
```

```r
reescalar_a_puntuacion_final <- function(x, x1, y1, x2, y2) {
  map2(x1, x2, expect_lte)
  map2(y1, y2, expect_lte)
  # map2(x1, x, expect_lte)
  # map2(x, x2, expect_lte)

  rr <- (x - x1) / (x2 - x1) * (y2 - y1) + y1
  rr <- janitor::round_half_up(rr, digits = 2)

  # xx <<- xx %>%
  #   add_row(barcode_hoja, x, x1, y1, x2, y2, rr)
  rr
}
testthat::expect_equal(reescalar_a_puntuacion_final(53.25, 22, 30, 110, 60), 40.65)
```

La fórmula genérica para reescalar un valor x desde un rango (de entrada) a otro rango (de salida) es:

$$(\text{maxSalida} - \text{minSalida}) \frac{x - \text{minEntrada}}{\text{maxEntrada} - \text{minEntrada}} + \text{minSalida}$$

La puntuación final se redondeará a dos decimales.

```r
# convalidaciones (notas mantenidas desde OPE previa) ====
fname_exenciones <- path(basepath, "exenciones/exenciones.csv")
if (fs::file_exists(fname_exenciones)) {
  message("procesando exenciones y convalidaciones...")
  exenciones <-
    read_csv2(fname_exenciones,
              show_col_types = FALSE) %>%
    janitor::clean_names() %>%
    filter(!is.na(nota_convalidada)) %>%
    verify(names(.) == c("nif", "apelidos_e_nome", "nota_convalidada")) %>%
    inner_join(aspirantes_admitidos, by = "nif") %>%
    mutate(
      sim = stringsim(
        apelidos_e_nome.x %>%
          str_to_upper() %>%
          stri_trans_general('latin-ascii'),
        apelidos_e_nome.y %>%
          str_to_upper() %>%
          stri_trans_general('latin-ascii')
      )
    ) %>%
    verify(sim > 0.9) %>%
    mutate(resultado = "supera ejercicio (convalidación)") %>%
    select(acceso,
```

acceso	plazas_ofertadas	aspirantes	prop_aspirantes_plaza	examenes_corregidos	porcent_presentados
discapacidad	8	329	41	108	32.83%
libre	83	5780	69	2785	48.18%

$covid

resultado/acceso	discapacidad	libre	Total
no alcanza puntuación mínima global	- (0)	25.0% (1)	25.0% (1)
supera ejercicio	- (0)	75.0% (3)	75.0% (3)
Total	- (0)	100.0% (4)	100.0% (4)

$general

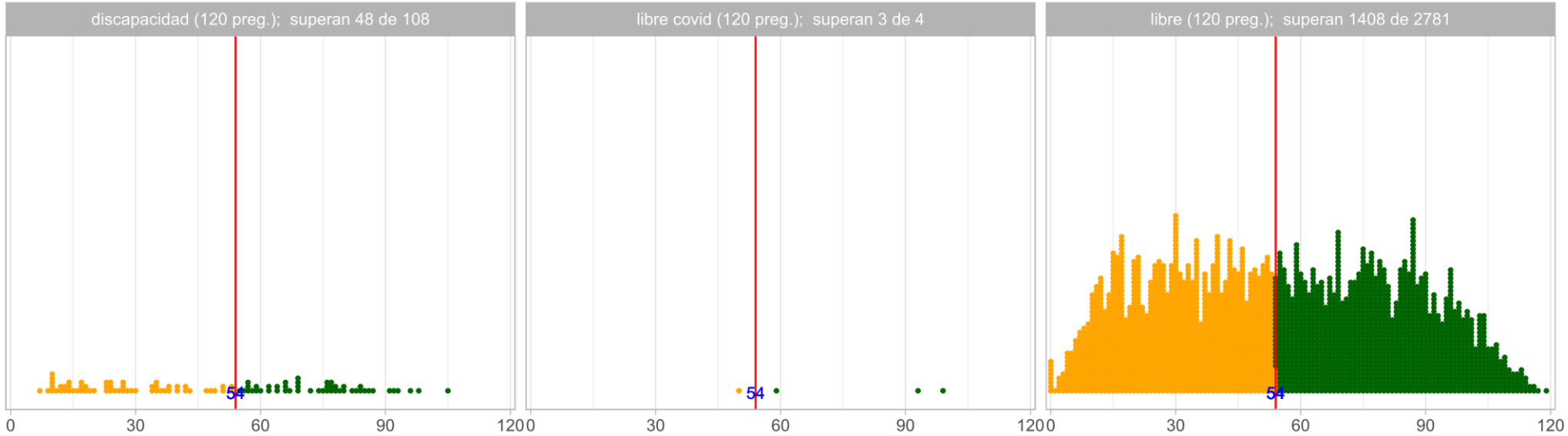
resultado/acceso	discapacidad	libre	Total
no alcanza puntuación mínima global	55.6% (60)	49.4% (1373)	49.6% (1433)
supera ejercicio	44.4% (48)	50.6% (1408)	50.4% (1456)
Total	100.0% (108)	100.0% (2781)	100.0% (2889)

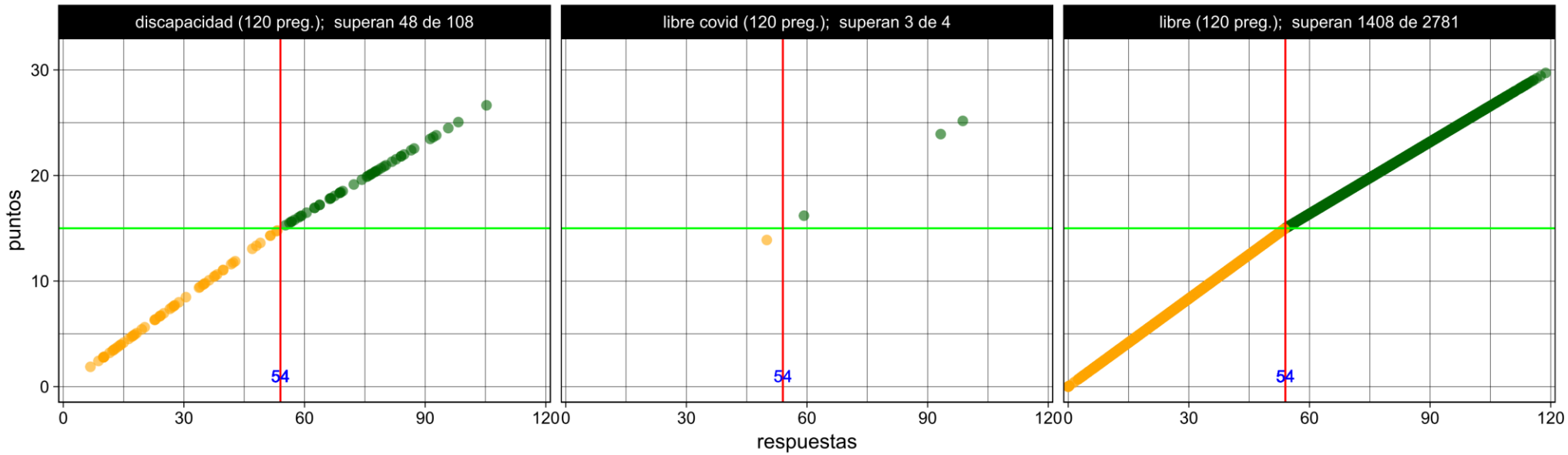
resultado/acceso	discapacidad	libre	Total
no alcanza puntuación mínima global	55.6% (60)	49.3% (1374)	49.6% (1434)
supera ejercicio	44.4% (48)	50.7% (1411)	50.4% (1459)
Total	100.0% (108)	100.0% (2785)	100.0% (2893)

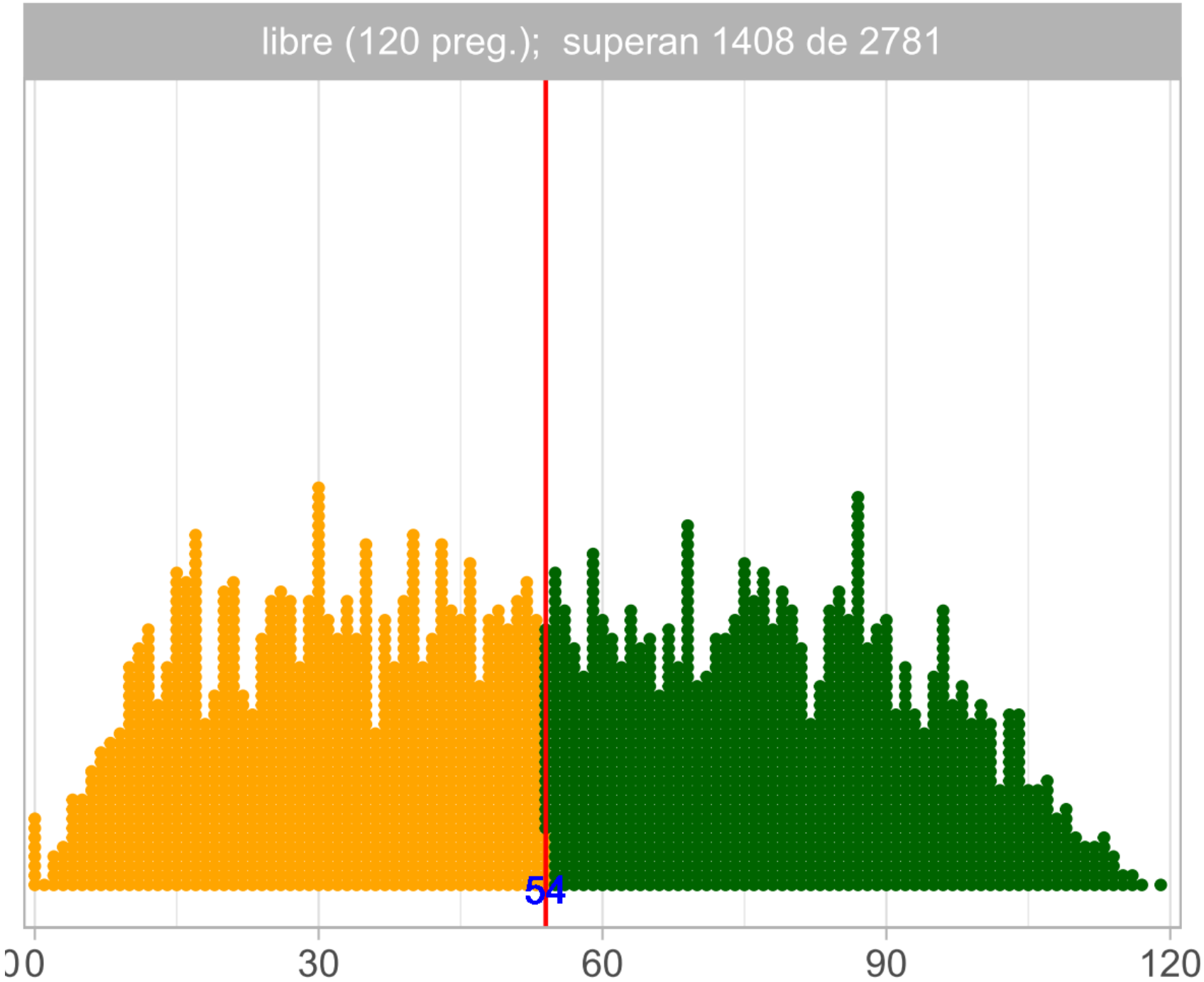
1459 aspirantes superan el ejercicio.

CORPO ADMINISTRATIVO DA ADMINISTRACION XERAL DA COMUNIDADE AUTONOMA DE GALICIA

2019/A/C/2053/1

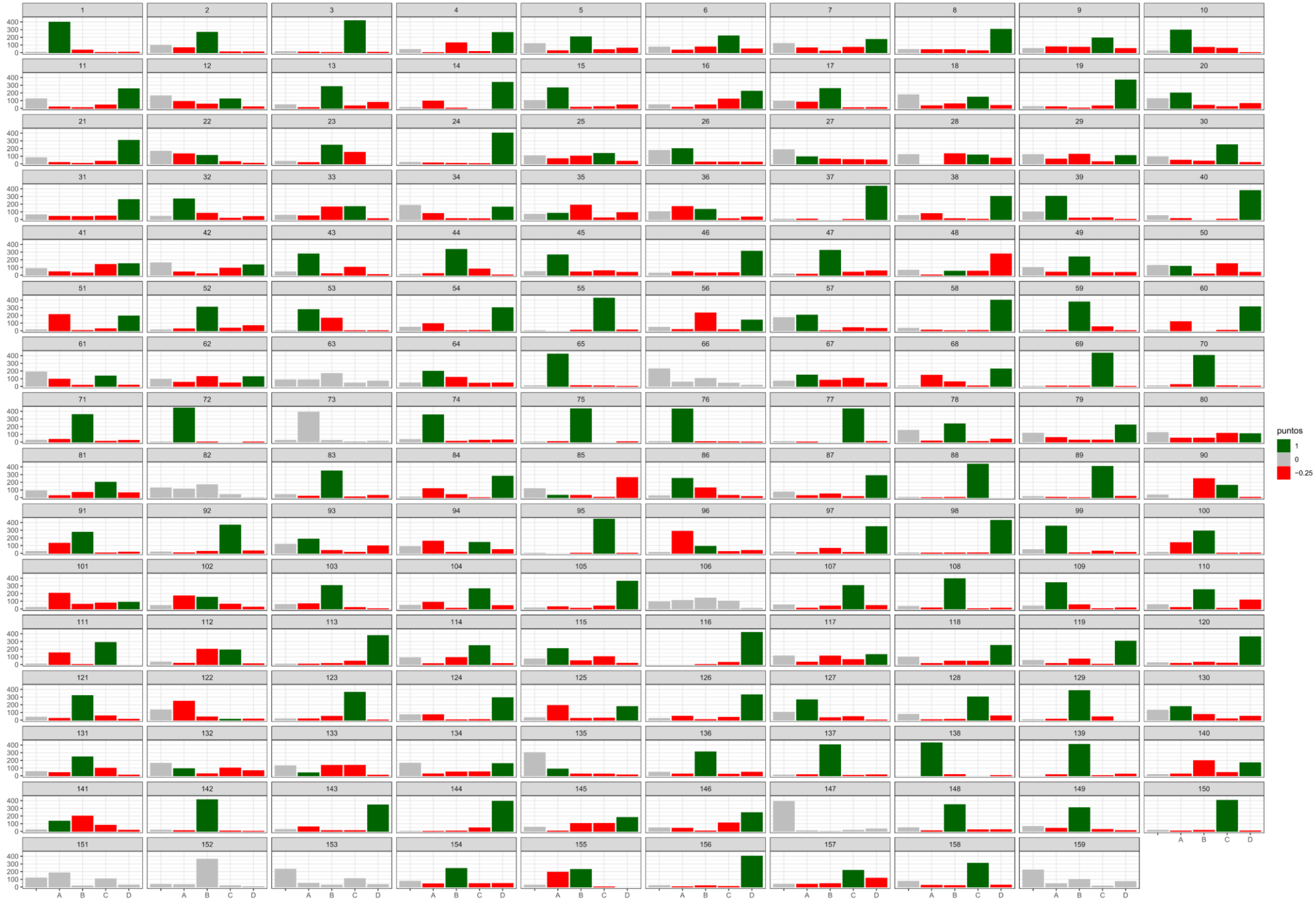


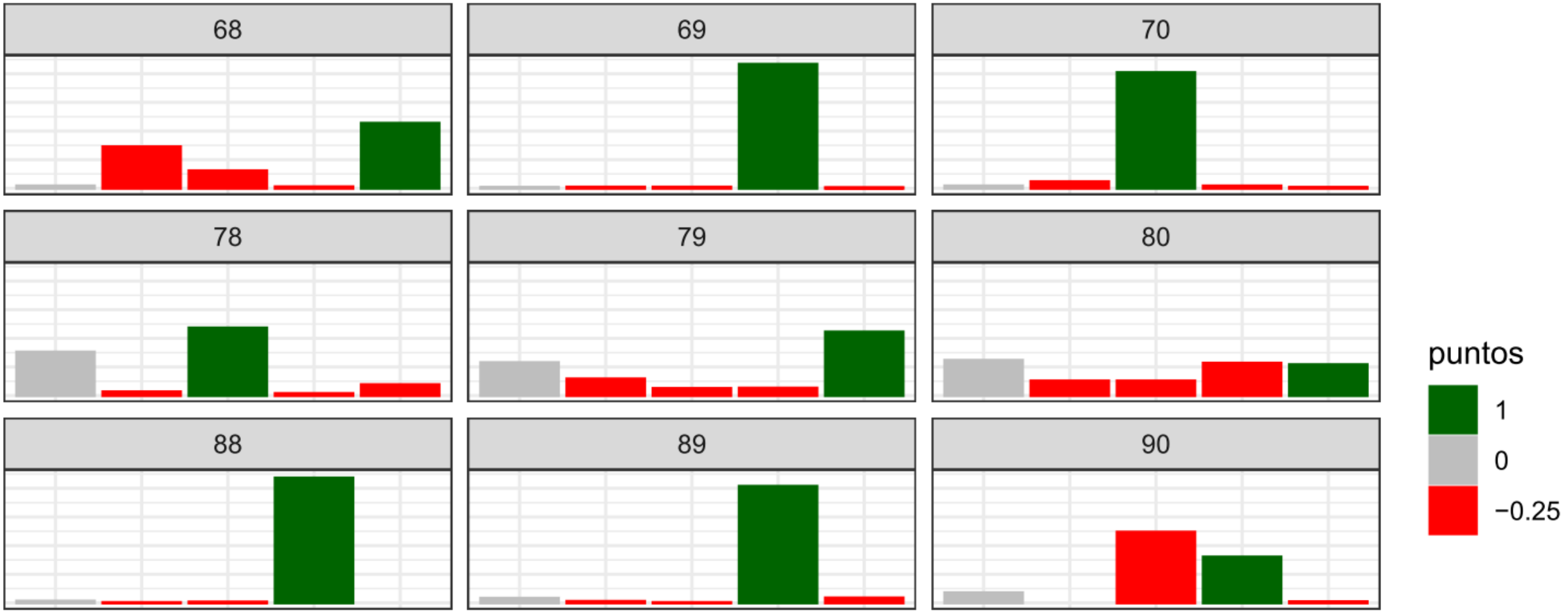


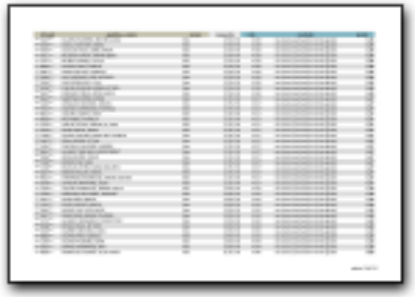

libre (120 preg.); superan 1408 de 2781
54
00
30
60
90
120


análisis de las respuestas: acceso libre
puntos
1
0
−0.25
0%
25%
50%
75%
100%

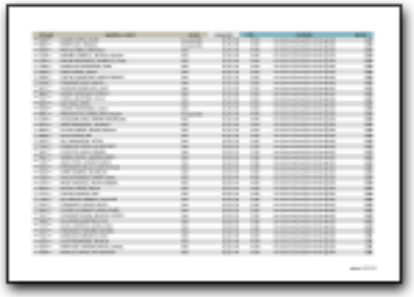
análisis de las respuestas: acceso libre



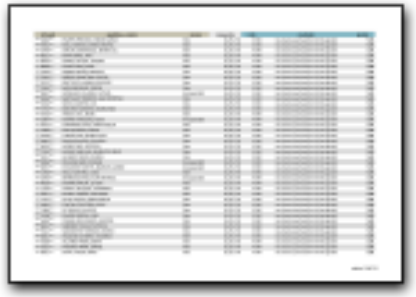

68
69
70
78
79
80
88
89
90
puntos
1
0
−0.25



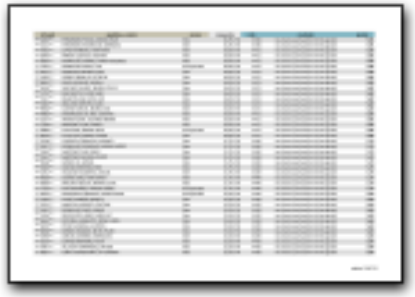
106



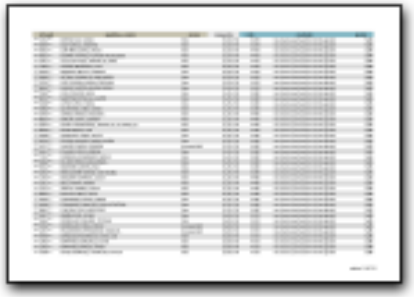
107



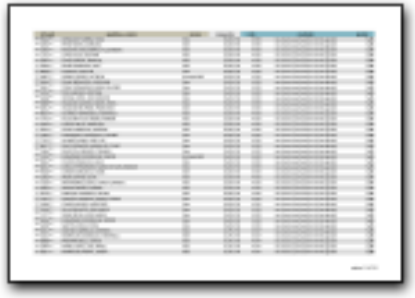
108



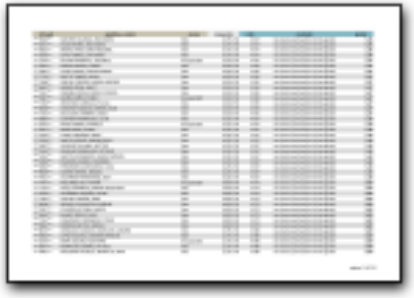
109



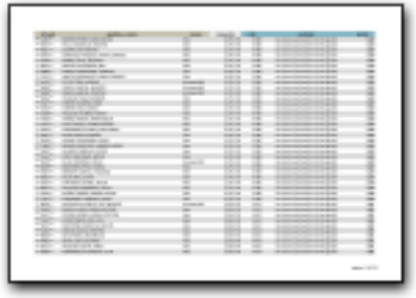
110



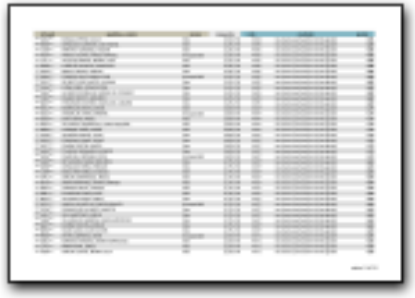
111



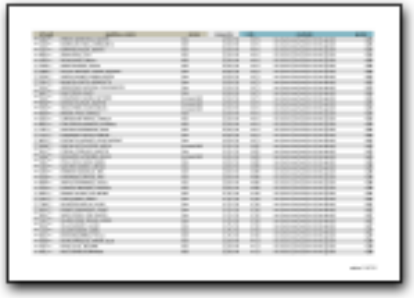
112



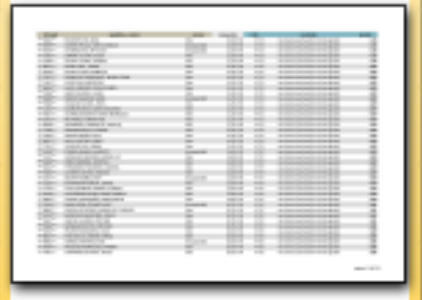
113



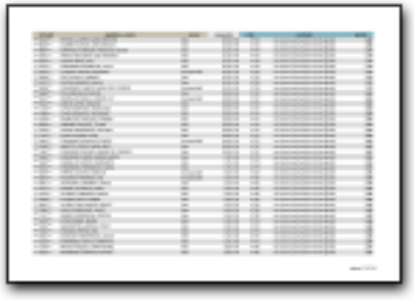
114



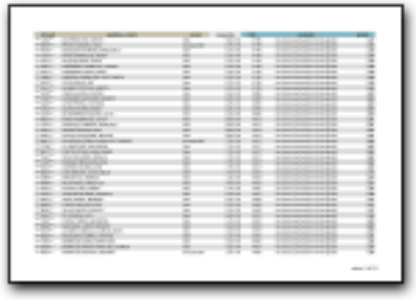
115



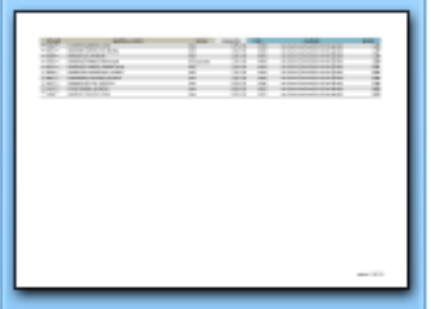
116



117



118



119

acceso	resp parte general	supera parte general	resp parte especifica	supera parte especifica	respuestas	supera corte global	resp_r	rank	resultado	puntos
libre	41,75 / 48	SI	103,50 / 130	SI	145,25 / 178	SI	145,25	1	supera ejercicio	**47,64**
libre	44,25 / 48	SI	100,25 / 130	SI	144,50 / 178	SI	144,50	2	supera ejercicio	**47,36**
libre	45,75 / 48	SI	97,75 / 130	SI	143,50 / 178	SI	143,50	3	supera ejercicio	**46,98**
libre	43,25 / 48	SI	98,50 / 130	SI	141,75 / 178	SI	141,75	4	supera ejercicio	**46,32**
libre	42,00 / 48	SI	99,50 / 130	SI	141,50 / 178	SI	141,50	5	supera ejercicio	**46,23**
libre	41,75 / 48	SI	97,75 / 130	SI	139,50 / 178	SI	139,50	6	supera ejercicio	**45,47**
discapacidad	41,00 / 48	SI	97,75 / 130	SI	138,75 / 178	SI	138,75	7	supera ejercicio	**45,19**
libre	41,50 / 48	SI	96,50 / 130	SI	138,00 / 178	SI	138,00	8	supera ejercicio	**44,91**
libre	38,00 / 48	SI	98,50 / 130	SI	136,50 / 178	SI	136,50	9	supera ejercicio	**44,34**
libre	41,75 / 48	SI	94,50 / 130	SI	136,25 / 178	SI	136,25	10	supera ejercicio	**44,25**
libre	41,00 / 48	SI	93,50 / 130	SI	134,50 / 178	SI	134,50	11	supera ejercicio	**43,58**
promocion_interna			97,75 / 130	SI	97,75 / 130	SI	133,84	12	supera ejercicio	**43,34**
libre	43,00 / 48	SI	90,75 / 130	SI	133,75 / 178	SI	133,75	13	supera ejercicio	**43,30**
libre	43,50 / 48	SI	90,00 / 130	SI	133,50 / 178	SI	133,50	14	supera ejercicio	**43,21**
libre	44,25 / 48	SI	88,25 / 130	SI	132,50 / 178	SI	132,50	15	supera ejercicio	**42,83**
libre	37,25 / 48	SI	94,75 / 130	SI	132,00 / 178	SI	132,00	16	supera ejercicio	**42,64**
libre	40,75 / 48	SI	91,25 / 130	SI	132,00 / 178	SI	132,00	16	supera ejercicio	**42,64**
libre	37,75 / 48	SI	93,75 / 130	SI	131,50 / 178	SI	131,50	18	supera ejercicio	**42,45**
libre	43,25 / 48	SI	87,75 / 130	SI	131,00 / 178	SI	131,00	19	supera ejercicio	**42,26**
libre	43,00 / 48	SI	87,50 / 130	SI	130,50 / 178	SI	130,50	20	supera ejercicio	**42,08**

acceso	resp parte general	resp parte especifica	respuestas	resp_r	rank	puntos
libre	41,75 / 48	103,50 / 130	145,25 / 178	145,25	1	**47,64**
libre	44,25 / 48	100,25 / 130	144,50 / 178	144,50	2	**47,36**
libre	45,75 / 48	97,75 / 130	143,50 / 178	143,50	3	**46,98**
libre	43,25 / 48	98,50 / 130	141,75 / 178	141,75	4	**46,32**
libre	42,00 / 48	99,50 / 130	141,50 / 178	141,50	5	**46,23**
libre	41,75 / 48	97,75 / 130	139,50 / 178	139,50	6	**45,47**
discapacidad	41,00 / 48	97,75 / 130	138,75 / 178	138,75	7	**45,19**
libre	41,50 / 48	96,50 / 130	138,00 / 178	138,00	8	**44,91**
libre	38,00 / 48	98,50 / 130	136,50 / 178	136,50	9	**44,34**
libre	41,75 / 48	94,50 / 130	136,25 / 178	136,25	10	**44,25**
libre	41,00 / 48	93,50 / 130	134,50 / 178	134,50	11	**43,58**
promocion_interna		97,75 / 130	97,75 / 130	133,84	12	**43,34**
libre	43,00 / 48	90,75 / 130	133,75 / 178	133,75	13	**43,30**
libre	43,50 / 48	90,00 / 130	133,50 / 178	133,50	14	**43,21**
libre	44,25 / 48	88,25 / 130	132,50 / 178	132,50	15	**42,83**