

Record matching

Adele Tyson

2023-07-02

```
library(janitor)
library(Hmisc)
library(readxl)
library(writexl)
library(reclin2)
library(lubridate)
library(RecordLinkage)
library(dgof) # for statistical testing
library(fdm2id) # for predict that works for kmeans
library(ppclust) # for cmeans
library(factoextra) # for get_cluster_?
library(FactoMineR) # for MCA
library(tidyverse)

commune_region_lookup <- read_excel("04_Data/Outputs/region_service_commune.xlsx") %>%
  clean_names() %>%
  select(-geometry)
#chile.adm3 <- st_read("04_Data/CHL_adm_humdata/chl_admbnda_adm3_bcn_20211008.shp") %>%
# mutate(commune_code = str_sub(ADM3_PCODE, start = 3, end = -1))

araucnorte_communes <- commune_region_lookup %>%
  filter(str_detect(health_service_name, "a Norte"))
araucsur_communes <- commune_region_lookup %>%
  filter(str_detect(health_service_name, "a Sur"))

chile_merged_raw <- read_csv("04_Data/Data_Chile_Merge.csv") %>% clean_names()

chile_merged <- chile_merged_raw %>%
  rename(sex_desc = sex,
         year = agno,
         school_code = rbd,
         school_check_code = dgv_rbd,
         school_name = nom_rbd,
         school_region_code = cod_reg_rbd,
         school_region_name_abr = nom_reg_rbd_a,
         school_province_code = cod_pro_rbd,
         school_commune_code = cod_com_rbd,
         school_commune_name = nom_com_rbd,
         school_dept_code = cod_deprov_rbd,
         school_dept_name = nom_deprov_rbd,
         school_dependency_code = cod_depe, # has categories 1-6, no1 and no2 here are no1 in grouped
         school_dependency_code_grouped = cod_depe2, # has categories 1-5
         school_rurality_code = rural_rbd,
```

```

school_operation_status = estado_estab,
teaching_code1 = cod_ense, # min = 10, max = 910, eg preschool, special education hearing impa
teaching_code2 = cod_ense2, # subject matter coding, 1-8
teaching_code3 = cod_ense3, # age based coding, 1-7
grade_code1 = cod_grado, # grade of schooling, 1-10, 21-25, 31-34, nests in teaching_code1
grade_code2 = cod_grado2, # equivalent grade of schooling for adult special education, 1-8, 99
grade_letter = let_cur, # refers to the class within the grade, close to start of alphabet is
course_timing = cod_jor, # time of day, morning, afternoon, both, night, no info
course_type = cod_tip_cur, # 0 = simple course, 1-4 = combined course, 99 = no info
course_descr = cod_des_cur, # Description of course (TP secondary education only). 0: Does not
student_id = mrun,
sex = gen_alu, # 0 = no info, 1 = male, 2 = female
dob = fec_nac_alu_2, # The second one has DD
age_june30 = edad_alu, # age at 30th June 2021
special_needs_status = int_alu, # integrated student indicator, 0 = no, 1 = yes. Mostly no
special_needs_code = cod_int_alu, # ADHD, blindness, etc. 0 = none. 105 = autism, 203 = ADHD.
student_region_code = cod_reg_alu,
student_commune_code = cod_com_alu,
student_commune_name = nom_com_alu,
economic_sector_code = cod_sec,
economic_specialty_code = cod_espe,
economic_branch_code = cod_rama,
economic_profspec_code = cod_men,
teaching_code_new = ens) %>%
mutate(commune_code = ifelse(nchar(as.character(student_commune_code)) == 4,
                             paste0("0", as.character(student_commune_code)),
                             as.character(student_commune_code)))

```

```

clinical_large_raw <- read_excel("04_Data/dataset_ssas_2015_2021.xlsx") %>% clean_names
#describe(clinical_raw)

```

```

clinical_large <- clinical_large_raw %>%
  select(c(-procedence, -ethnicity, -education_level, -disability, -foster_care)) %>%
  # Fix the date columns
  mutate(dob_eng = ifelse(str_detect(date_of_birth, "/"), 1,
                          ifelse(str_detect(date_of_birth, "-"), 0, NA)),
         apt_eng = ifelse(str_detect(date_appointment, "/"), 1, ifelse(str_detect(date_appointment, "-"),
         dob_day = ifelse(dob_eng == 1, as.integer(str_extract(date_of_birth, "^\\d+")),
                          ifelse(dob_eng == 0, as.integer(str_extract(date_of_birth, "^\\d+")), NA)),
         dob_month = ifelse(dob_eng == 1, as.integer(str_extract(date_of_birth, "(?<=/)\\d+(?=/)")),
                            ifelse(dob_eng == 0, str_extract(date_of_birth, "(?<=)\\w+(?=-)"), NA)),
         dob_year = ifelse(dob_eng == 1, as.integer(str_extract(date_of_birth, "\\d+$")),
                           ifelse(dob_eng == 0, as.integer(str_extract(date_of_birth, "\\d+$")) + 2000, NA)),
         dob_month_eng = as.integer(ifelse(dob_month == "ene", 1,
                                           ifelse(dob_month == "abr", 4,
                                           ifelse(dob_month == "ago", 8,
                                           ifelse(dob_month == "sept", 9,
                                           ifelse(dob_month == "dic", 12, dob_month)))))),
         dob = make_date(year = dob_year, month = dob_month_eng, day = dob_day),
         apt_day = ifelse(apt_eng == 1, as.integer(str_extract(date_appointment, "^\\d+")),
                          ifelse(apt_eng == 0, as.integer(str_extract(date_appointment, "^\\d+")), NA)),
         apt_month = ifelse(apt_eng == 1, as.integer(str_extract(date_appointment, "(?<=/)\\d+(?=/)")),
                            ifelse(apt_eng == 0, str_extract(date_appointment, "(?<=)\\w+(?=-)"), NA)),
         apt_year = ifelse(apt_eng == 1, as.integer(str_extract(date_appointment, "\\d+$")),

```

```

        ifelse(apt_eng == 0, as.integer(str_extract(date_appointment, "\\d+$")) + 2000, NA),
    apt_month_eng = as.integer(ifelse(apt_month == "ene", 1,
        ifelse(apt_month == "abr", 4,
            ifelse(apt_month == "ago", 8,
                ifelse(apt_month == "sept", 9,
                    ifelse(apt_month == "dic", 12, apt_month)))))),
    apt_date = make_date(year = apt_year, month = apt_month_eng, day = apt_day),
    age_june30 = trunc(time_length(interval(ymd(dob), ymd("2021-06-30")), unit = "year")),
    commune_name_upper = ifelse(comuna == "CHOL CHOL", "CHOLCHOL",
        ifelse(comuna == "CURACAUTIN", "CURACAUTÍN",
            ifelse(comuna == "PITRUFQUEN", "PITRUFQUÉN",
                ifelse(comuna == "PUCON", "PUCÓN",
                    ifelse(comuna == "TOLTEN", "TOLTÉN",
                        ifelse(comuna == "VILCUN", "VILCÚN", comuna)))))),
    #commune_name_upper = comuna,
    ses_status = ifelse(socio_economic_level == "FONASA - A", 1,
        ifelse(socio_economic_level == "FONASA - B", 2,
            ifelse(socio_economic_level == "FONASA - C", 2,
                ifelse(socio_economic_level == "FONASA - D", 2,
                    ifelse(socio_economic_level == "Private Health Insurance", 3,
                        ifelse(socio_economic_level %in% c("COLMENA GOLDEN CROSS", "RIO BLANCO", "CARABINIEROS"), 4, NA))),
    autism = 1,
    intdisab = 0,
    aut_rank = 1
    ) %>%
left_join(commune_region_lookup, by = "commune_name_upper") %>%
select(id, gender,
    commune_code, commune_name, commune_name_upper,
    health_service_name, region_name,
    socio_economic_level, ses_status,
    dob, age_june30,
    apt_date, hospital, medical_specialty, type_appointment,
    autism, intdisab, aut_rank)

aut_codes <- unique(clinical_large_raw$codigo)

clinical_small_raw <- read_excel("04_Data/Dataset_Vill_2014_2021.xlsx", col_names = TRUE) %>% clean_names()

clinical_small <- clinical_small_raw %>%
    rename("dob" = "fecha_nacimiento",
        "apt_date" = "fecha_ejecutada",
        "type_appointment" = "appointment",
        "diagnosis" = "diagnostico_1") %>%
    mutate(gender = str_to_title(gender),
        autism = ifelse(cod_dg_1 %in% aut_codes |
            cod_dg_2 %in% aut_codes |
            cod_dg_3 %in% aut_codes, 1, 0),
        aut_rank = ifelse(cod_dg_1 %in% aut_codes, 1,
            ifelse(cod_dg_2 %in% aut_codes, 2,
                ifelse(cod_dg_3 %in% aut_codes, 3, NA))),
        age_june30 = trunc(time_length(interval(ymd(dob), ymd("2021-06-30")), unit = "year")),
        commune_name_upper = ifelse(comuna == "CHOL CHOL", "CHOLCHOL",
            ifelse(comuna == "CURACAUTIN", "CURACAUTÍN",

```

```

        ifelse(comuna == "PITRUFQUEN", "PITRUFQUÉN",
        ifelse(comuna == "PUCON", "PUCÓN",
        ifelse(comuna == "TOLTEN", "TOLTÉN",
        ifelse(comuna == "VILCUN", "VILCÚN",
        ifelse(comuna == "DIEGO DE ALMAGRO (#)", "DIEGO DE ALMAGRO",
        ifelse(comuna == "MACHALI", "MACHALÍ",
        ifelse(comuna == "TEMUCO (##)", "TEMUCO", comuna))))))))) ,
    ses_status = ifelse(socio_economic_level == "FONASA - A", 1,
        ifelse(socio_economic_level == "FONASA - B", 2,
        ifelse(socio_economic_level == "FONASA - C", 2,
        ifelse(socio_economic_level == "FONASA - D", 2,
        ifelse(socio_economic_level == "Private Health Insurance", 3,
        ifelse(socio_economic_level %in% c("COLMENA GOLDEN CROSS", "RIO BLANCO", "CARABIN
    ) %>%
left_join(commune_region_lookup, by = "commune_name_upper") %>%
#filter(autism == 1) %>%
select(id, gender, commune_code, commune_name, commune_name_upper, health_service_name, region_name,

## Warning in left_join(., commune_region_lookup, by = "commune_name_upper"): Each row in `x` is expected
## i Row 2030 of `x` matches multiple rows.
## i If multiple matches are expected, set `multiple = "all"` to silence this
##   warning.
# Throws a warning because there are 2 records for Tocopila which is in two regions. Will keep both bec

intdisab_codes <- unique(c(clinical_small_raw$cod_dg_1, clinical_small_raw$cod_dg_2, clinical_small_raw
  str_subset("F7") %>%
  sort()

clinical_small <- clinical_small %>%
  mutate(intdisab = ifelse(cod_dg_1 %in% intdisab_codes |
                           cod_dg_2 %in% intdisab_codes |
                           cod_dg_3 %in% intdisab_codes, 1, 0)) %>%
  #rename("codigo" = "cod_dg_1") %>%
  select(c(-cod_dg_1, -cod_dg_2, -cod_dg_3, -diagnosis))

clinical <- rbind(clinical_large, clinical_small)

clinical_communes <- clinical %>% group_by(commune_code) %>% summarise() %>% arrange() %>%
  mutate(commune_in_school_data = ifelse(commune_code %in% unique(chile_merged$commune_code), 1, 0)) #

```

Fixed the date columns because they were in English and Spanish. Redefined the age column to be age at 30th June 2021.

Get one row per person per commune to make matching more efficient. Take the earliest appointment for each person.

```

get.min.na <- function(x) ifelse( !all(is.na(x)), min(x, na.rm = TRUE), NA)
get.max.na <- function(x) ifelse( !all(is.na(x)), max(x, na.rm = TRUE), NA)

patients <- clinical %>%
  filter(commune_code %in% araucsur_communes$commune_code) %>%
  group_by(id, gender, dob, commune_name, region_name) %>% # Maybe move SES back to here
  summarise(ses_status = get.min.na(ses_status),
            autism = get.max.na(autism),

```

```

      #intdisab = get.max.na(intdisab),
      aut_rank = get.min.na(aut_rank)) %>%
ungroup() %>%
rename("student_commune_name" = "commune_name",
      "student_region_name" = "region_name",
      "sex_desc" = "gender") %>%
rowid_to_column("row_id") %>%
select(row_id,
      id,
      dob,
      sex_desc,
      student_commune_name,
      autism,
      ses_status,
      #intdisab,
      aut_rank) #, student_region_name) #, count)

```

`summarise()` has grouped output by 'id', 'gender', 'dob', 'commune_name'. You
can override using the `.groups` argument.

```
write_xlsx(patients, "04_Data/Outputs/patients.xlsx")
```

```
length(unique(patients$id))
```

```
## [1] 1688
```

```

patients_unique <- patients %>%
  group_by(id) %>%
  summarise(sex_desc = list(sex_desc),
            student_commune_name = list(student_commune_name),
            dob = list(dob),
            ses_status = list(ses_status))
write_csv(patients_unique, "04_Data/Outputs/patients_unique.csv") # can't write columns containing list.

```

NB: there are 1688 unique ID's in patients and it's 1702 rows long because some people are represented in 2 communes.

Are all the records in the small dataset in the big one? No

```
clinical %>% filter(id %in% clinical_small$id)
```

```

## # A tibble: 3,558 x 18
##   id      gender commune_c-1 commu-2 commu-3 healt-4 regio-5 socio-6 ses_s-7
##   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <dbl>
## 1 21282495-K Female 09109   Loncoc~ LONCOC~ Arauca~ Región~ FONASA~ 2
## 2 21282495-K Female 09109   Loncoc~ LONCOC~ Arauca~ Región~ FONASA~ 2
## 3 21294488-2 Male   09120   Villar~ VILLAR~ Arauca~ Región~ Privat~ 3
## 4 21294488-2 Male   09120   Villar~ VILLAR~ Arauca~ Región~ Privat~ 3
## 5 21294488-2 Male   09120   Villar~ VILLAR~ Arauca~ Región~ Privat~ 3
## 6 21341924-2 Male   09115   Pucón   PUCÓN   Arauca~ Región~ FONASA~ 2
## 7 21341924-2 Male   09115   Pucón   PUCÓN   Arauca~ Región~ FONASA~ 2
## 8 21341924-2 Male   09115   Pucón   PUCÓN   Arauca~ Región~ FONASA~ 2
## 9 21341924-2 Male   09115   Pucón   PUCÓN   Arauca~ Región~ FONASA~ 2
## 10 21341924-2 Male   09115   Pucón   PUCÓN   Arauca~ Región~ FONASA~ 2
## # ... with 3,548 more rows, 9 more variables: dob <date>, age_june30 <dbl>,
## # apt_date <date>, hospital <chr>, medical_specialty <chr>,

```

```
## # type_appointment <chr>, autism <dbl>, intdisab <dbl>, aut_rank <dbl>, and
## # abbreviated variable names 1: commune_code, 2: commune_name,
## # 3: commune_name_upper, 4: health_service_name, 5: region_name,
## # 6: socio_economic_level, 7: ses_status
```

Assume this is because the big clinical dataset only has people with autism, not ADHD.

Only try to link clinical data to records in the schools data for the Southern health service in Araucanía (ARAUC) because that's where the clinical data is from.

```
school <- chile_merged %>%
  # mutate(commune_code = ifelse(nchar(as.character(student_commune_code)) == 4,
  #                               paste0("0", as.character(student_commune_code)),
  #                               as.character(student_commune_code))) %>%
  select(-student_commune_name) %>%
  left_join(commune_region_lookup, by = "commune_code") %>%
  filter(commune_code %in% araucsur_communes$commune_code) %>%
  #filter(health_service_name == "Servicio de Salud Araucanía Sur") %>% # This should be filtered either
  filter(age_june30 >= 6 & age_june30 <= 18, sex != 0) %>% # Could try without this filter to pick up e
  # filter only the communes represented in the clinical data here?
  mutate(autism = ifelse(special_needs_code == 105, 1, 0),
         #intdisab = 0,
         aut_rank = 1,
         dob = ymd(dob),
         ses_status = ifelse(school_fee == "", NA,
                             ifelse(school_fee == "GRATUITO", 1,
                                     ifelse(school_fee == "$1.000 A $10.000", 2,
                                             ifelse(school_fee == "$10.001 A $25.000", 2,
                                                     ifelse(school_fee == "$25.001 A $50.000", 2,
                                                             ifelse(school_fee == "$50.001 A $100.000", 2,
                                                                     ifelse(school_fee == "MAS DE $100.000", 2,
                                                                           ifelse(school_fee == "SIN INFORMACION", NA, NA)))))))))) %>%
  filter(autism == 1) %>% # We only want to find additional autism cases in the clinical records, we do
  rename(student_commune_name = commune_name) %>%
  select(dob,
         sex_desc,
         student_commune_name,
         #commune_name,
         #health_service_name,
         autism,
         ses_status,
         #intdisab,
         aut_rank#,
         #student_region_name
  ) %>%
  rowid_to_column("id")
school[dim(school)[1]+1, ] <- c(dim(school)[1]+1, "2023-06-26", "Female", "Misc", 0, 3, 0)

# Do the commune names align well? Yes
table(sort(unique(patients$student_commune_name, sort(unique(school$student_commune_name)))))
```

```
##
##      Carahue      Cholchol      Cunco      Curarrehue      Freire
##          47           9          33          12           31
##      Galvarino      Gorbea      Lautaro      Loncoche      Melipeuco
##          22          21         106          84           5
```

```
## Nueva Imperial Padre Las Casas Perquenco Pitrufulquén Pucón
## 81 148 19 47 88
## Saavedra Temuco Teodoro Schmidt Toltén Vilcún
## 14 602 12 17 60
## Villarrica
## 244
```

```
sort(unique(patients$student_commune_name))
```

```
## [1] "Carahue" "Cholchol" "Cunco" "Curarrehue"
## [5] "Freire" "Galvarino" "Gorbea" "Lautaro"
## [9] "Loncoche" "Melipeuco" "Nueva Imperial" "Padre Las Casas"
## [13] "Perquenco" "Pitrufulquén" "Pucón" "Saavedra"
## [17] "Temuco" "Teodoro Schmidt" "Toltén" "Vilcún"
## [21] "Villarrica"
```

```
sort(unique(school$student_commune_name))
```

```
## [1] "Carahue" "Cholchol" "Cunco" "Curarrehue"
## [5] "Freire" "Galvarino" "Gorbea" "Lautaro"
## [9] "Loncoche" "Melipeuco" "Misc" "Nueva Imperial"
## [13] "Padre Las Casas" "Perquenco" "Pitrufulquén" "Pucón"
## [17] "Saavedra" "Temuco" "Teodoro Schmidt" "Toltén"
## [21] "Vilcún" "Villarrica"
```

Added a fake row at the end of school to have a ses=3 represented so that pairing works.

Perfect match in communes between patient and school dataset when both are filtered to only be communes in Arauc Sur health region.

Try manual linkage

```
patients_grouped <- patients %>%
  group_by(sex_desc,
            dob,
            student_commune_name) %>%
  summarise(count = n(),
            ids = list(id))
```

```
## `summarise()` has grouped output by 'sex_desc', 'dob'. You can override using
## the `.groups` argument.
```

```
school_grouped <- school %>%
  group_by(sex_desc,
            dob,
            student_commune_name) %>%
  summarise(count = n(),
            #ids = list(rowid)
            ses = list(ses_status))
```

```
## `summarise()` has grouped output by 'sex_desc', 'dob'. You can override using
## the `.groups` argument.
```

```
sort(unique(patients$student_commune_name))
```

```
## [1] "Carahue" "Cholchol" "Cunco" "Curarrehue"
## [5] "Freire" "Galvarino" "Gorbea" "Lautaro"
```



```
## [9] "Loncoche"      "Melipeuco"      "Nueva Imperial" "Padre Las Casas"
## [13] "Perquenco"     "Pitrufquén"     "Pucón"          "Saavedra"
## [17] "Temuco"        "Teodoro Schmidt" "Toltén"         "Vilcún"
## [21] "Villarrica"
```

```
sort(unique(school$student_commune_name))
```

```
## [1] "Carahue"      "Cholchol"      "Cunco"          "Curarrehue"
## [5] "Freire"       "Galvarino"     "Gorbea"         "Lautaro"
## [9] "Loncoche"     "Melipeuco"     "Misc"           "Nueva Imperial"
## [13] "Padre Las Casas" "Perquenco"     "Pitrufquén"     "Pucón"
## [17] "Saavedra"     "Temuco"        "Teodoro Schmidt" "Toltén"
## [21] "Vilcún"       "Villarrica"
```

```
merged <- merge(school, patients, by = c("sex_desc", "dob", "student_commune_name"), all = FALSE)
merged %>% filter(!is.na(id.x) & !is.na(id.y)) # 205 matches
```

##	sex_desc	dob	student_commune_name	id.x	autism.x	ses_status.x
## 1	Female	2003-04-16	Loncoche	450	1	1
## 2	Female	2003-11-25	Temuco	437	1	2
## 3	Female	2005-12-07	Temuco	380	1	1
## 4	Female	2006-08-10	Lautaro	470	1	1
## 5	Female	2006-09-20	Freire	109	1	1
## 6	Female	2006-10-10	Padre Las Casas	263	1	1
## 7	Female	2008-05-20	Gorbea	187	1	1
## 8	Female	2008-06-21	Temuco	269	1	1
## 9	Female	2009-05-08	Temuco	57	1	1
## 10	Female	2009-06-22	Pucón	332	1	1
## 11	Female	2010-04-27	Temuco	426	1	1
## 12	Female	2011-04-20	Temuco	173	1	2
## 13	Female	2012-01-31	Villarrica	172	1	1
## 14	Female	2012-04-07	Pucón	425	1	1
## 15	Female	2012-05-28	Vilcún	214	1	1
## 16	Female	2012-06-18	Villarrica	41	1	1
## 17	Female	2012-09-13	Temuco	104	1	1
## 18	Female	2013-04-20	Galvarino	296	1	1
## 19	Female	2013-06-19	Temuco	267	1	1
## 20	Female	2013-08-30	Padre Las Casas	311	1	1
## 21	Female	2013-12-30	Villarrica	190	1	2
## 22	Female	2014-02-15	Temuco	105	1	1
## 23	Female	2014-10-09	Gorbea	419	1	1
## 24	Female	2014-10-16	Temuco	415	1	2
## 25	Female	2014-11-12	Temuco	351	1	1
## 26	Female	2014-12-11	Pucón	80	1	1
## 27	Female	2014-12-12	Temuco	464	1	1
## 28	Male	2003-01-27	Temuco	227	1	1
## 29	Male	2003-03-06	Temuco	465	1	1
## 30	Male	2003-06-14	Temuco	92	1	1
## 31	Male	2003-06-15	Temuco	165	1	1
## 32	Male	2003-06-29	Temuco	53	1	1
## 33	Male	2003-08-03	Temuco	313	1	1
## 34	Male	2003-10-21	Temuco	186	1	1
## 35	Male	2003-12-15	Temuco	389	1	1
## 36	Male	2004-03-05	Nueva Imperial	442	1	1
## 37	Male	2004-03-12	Temuco	133	1	1

## 38	Male	2004-07-07	Temuco	322	1	2
## 39	Male	2004-09-28	Loncoche	216	1	1
## 40	Male	2004-10-01	Freire	307	1	1
## 41	Male	2004-11-07	Temuco	362	1	1
## 42	Male	2004-12-25	Cunco	174	1	1
## 43	Male	2005-01-03	Temuco	39	1	2
## 44	Male	2005-01-09	Temuco	49	1	1
## 45	Male	2005-01-21	Temuco	202	1	1
## 46	Male	2005-05-24	Temuco	78	1	1
## 47	Male	2005-06-17	Temuco	123	1	1
## 48	Male	2005-06-17	Temuco	123	1	1
## 49	Male	2005-08-29	Temuco	70	1	1
## 50	Male	2005-09-06	Temuco	405	1	2
## 51	Male	2006-03-04	Temuco	147	1	1
## 52	Male	2006-03-22	Temuco	11	1	1
## 53	Male	2006-04-13	Padre Las Casas	301	1	1
## 54	Male	2006-09-09	Galvarino	434	1	1
## 55	Male	2006-09-19	Lautaro	219	1	1
## 56	Male	2006-10-06	Lautaro	448	1	1
## 57	Male	2006-10-10	Vilcún	478	1	1
## 58	Male	2006-10-27	Temuco	247	1	1
## 59	Male	2006-11-02	Padre Las Casas	176	1	2
## 60	Male	2006-11-06	Temuco	471	1	2
## 61	Male	2006-11-06	Temuco	471	1	2
## 62	Male	2007-01-08	Carahue	319	1	1
## 63	Male	2007-01-23	Villarrica	363	1	1
## 64	Male	2007-02-13	Temuco	235	1	1
## 65	Male	2007-03-22	Lautaro	265	1	1
## 66	Male	2007-04-09	Padre Las Casas	31	1	1
## 67	Male	2007-04-25	Lautaro	336	1	1
## 68	Male	2007-05-11	Temuco	355	1	1
## 69	Male	2007-06-16	Pitrufquén	358	1	1
## 70	Male	2007-08-20	Pitrufquén	237	1	1
## 71	Male	2007-11-06	Villarrica	295	1	1
## 72	Male	2007-12-28	Loncoche	130	1	1
## 73	Male	2008-01-28	Nueva Imperial	44	1	1
## 74	Male	2008-03-05	Pucón	420	1	1
## 75	Male	2008-03-14	Temuco	408	1	1
## 76	Male	2008-03-25	Temuco	289	1	1
## 77	Male	2008-05-20	Padre Las Casas	100	1	1
## 78	Male	2008-06-18	Vilcún	55	1	1
## 79	Male	2008-08-24	Saavedra	158	1	1
## 80	Male	2008-10-10	Temuco	112	1	2
## 81	Male	2008-10-22	Villarrica	72	1	1
## 82	Male	2008-11-22	Nueva Imperial	467	1	1
## 83	Male	2008-12-06	Lautaro	22	1	1
## 84	Male	2008-12-21	Temuco	394	1	1
## 85	Male	2008-12-29	Temuco	93	1	1
## 86	Male	2009-01-07	Lautaro	361	1	1
## 87	Male	2009-01-12	Temuco	26	1	1
## 88	Male	2009-02-13	Pucón	3	1	1
## 89	Male	2009-02-26	Loncoche	168	1	1
## 90	Male	2009-04-23	Loncoche	314	1	1
## 91	Male	2009-08-05	Villarrica	60	1	1

## 92	Male	2009-08-14	Pucón	252	1	1
## 93	Male	2009-08-29	Temuco	159	1	2
## 94	Male	2009-10-01	Temuco	328	1	1
## 95	Male	2009-10-26	Temuco	341	1	1
## 96	Male	2010-01-02	Freire	272	1	1
## 97	Male	2010-01-25	Padre Las Casas	73	1	2
## 98	Male	2010-02-21	Loncoche	180	1	1
## 99	Male	2010-02-26	Teodoro Schmidt	213	1	1
## 100	Male	2010-03-07	Lautaro	242	1	1
## 101	Male	2010-03-16	Gorbea	246	1	1
## 102	Male	2010-05-20	Villarrica	396	1	1
## 103	Male	2010-06-07	Temuco	476	1	1
## 104	Male	2010-06-08	Nueva Imperial	292	1	1
## 105	Male	2010-07-21	Cholchol	194	1	1
## 106	Male	2010-07-28	Freire	382	1	1
## 107	Male	2010-08-29	Villarrica	365	1	1
## 108	Male	2010-09-13	Padre Las Casas	312	1	1
## 109	Male	2010-10-12	Temuco	201	1	1
## 110	Male	2010-12-09	Pucón	346	1	1
## 111	Male	2010-12-09	Temuco	107	1	1
## 112	Male	2011-01-13	Villarrica	18	1	2
## 113	Male	2011-01-24	Temuco	87	1	1
## 114	Male	2011-02-11	Cunco	368	1	1
## 115	Male	2011-02-22	Temuco	139	1	1
## 116	Male	2011-03-03	Lautaro	228	1	1
## 117	Male	2011-04-13	Villarrica	275	1	1
## 118	Male	2011-06-13	Temuco	203	1	1
## 119	Male	2011-07-02	Lautaro	475	1	1
## 120	Male	2011-08-02	Carahue	113	1	1
## 121	Male	2011-09-06	Teodoro Schmidt	229	1	1
## 122	Male	2011-09-08	Temuco	277	1	1
## 123	Male	2011-10-27	Teodoro Schmidt	283	1	1
## 124	Male	2011-11-10	Freire	300	1	1
## 125	Male	2011-11-12	Padre Las Casas	278	1	1
## 126	Male	2012-01-11	Pucón	290	1	1
## 127	Male	2012-03-06	Villarrica	243	1	1
## 128	Male	2012-03-12	Temuco	261	1	1
## 129	Male	2012-04-16	Temuco	472	1	1
## 130	Male	2012-05-29	Temuco	8	1	1
## 131	Male	2012-06-01	Padre Las Casas	66	1	1
## 132	Male	2012-06-02	Temuco	141	1	1
## 133	Male	2012-06-25	Galvarino	183	1	1
## 134	Male	2012-07-08	Temuco	315	1	1
## 135	Male	2012-07-16	Galvarino	152	1	1
## 136	Male	2012-07-29	Vilcún	16	1	1
## 137	Male	2012-09-07	Temuco	293	1	1
## 138	Male	2012-09-21	Cunco	264	1	1
## 139	Male	2012-10-13	Villarrica	45	1	1
## 140	Male	2012-10-18	Villarrica	392	1	1
## 141	Male	2012-11-03	Lautaro	443	1	1
## 142	Male	2012-11-05	Temuco	447	1	1
## 143	Male	2012-12-10	Pitrufquén	304	1	<NA>
## 144	Male	2012-12-25	Padre Las Casas	29	1	1
## 145	Male	2013-01-26	Gorbea	385	1	1

## 146	Male	2013-01-30	Pitrufrquén	97	1	1
## 147	Male	2013-02-12	Temuco	366	1	1
## 148	Male	2013-02-25	Gorbea	294	1	1
## 149	Male	2013-02-27	Nueva Imperial	24	1	1
## 150	Male	2013-03-24	Villarrica	386	1	1
## 151	Male	2013-04-23	Toltén	350	1	1
## 152	Male	2013-05-20	Temuco	280	1	1
## 153	Male	2013-05-23	Lautaro	189	1	1
## 154	Male	2013-05-30	Villarrica	469	1	1
## 155	Male	2013-07-07	Vilcún	338	1	1
## 156	Male	2013-10-16	Vilcún	111	1	1
## 157	Male	2013-10-23	Pitrufrquén	324	1	1
## 158	Male	2013-11-05	Villarrica	211	1	1
## 159	Male	2013-11-14	Temuco	71	1	1
## 160	Male	2014-02-19	Temuco	326	1	1
## 161	Male	2014-02-19	Temuco	326	1	1
## 162	Male	2014-02-19	Temuco	451	1	1
## 163	Male	2014-02-19	Temuco	451	1	1
## 164	Male	2014-04-17	Temuco	335	1	1
## 165	Male	2014-04-21	Temuco	129	1	1
## 166	Male	2014-05-06	Villarrica	271	1	1
## 167	Male	2014-05-17	Cunco	125	1	1
## 168	Male	2014-05-20	Temuco	287	1	1
## 169	Male	2014-05-24	Loncoche	407	1	1
## 170	Male	2014-06-02	Temuco	162	1	1
## 171	Male	2014-06-16	Temuco	77	1	1
## 172	Male	2014-07-07	Temuco	116	1	1
## 173	Male	2014-08-30	Galvarino	6	1	1
## 174	Male	2014-09-06	Temuco	145	1	1
## 175	Male	2014-09-06	Temuco	145	1	1
## 176	Male	2014-09-12	Loncoche	310	1	1
## 177	Male	2014-09-12	Loncoche	310	1	1
## 178	Male	2014-10-07	Temuco	98	1	1
## 179	Male	2014-10-07	Temuco	98	1	1
## 180	Male	2014-10-07	Temuco	118	1	1
## 181	Male	2014-10-07	Temuco	118	1	1
## 182	Male	2014-10-28	Temuco	15	1	1
## 183	Male	2014-11-02	Temuco	399	1	1
## 184	Male	2014-11-16	Lautaro	91	1	1
## 185	Male	2014-11-19	Pucón	357	1	1
## 186	Male	2014-12-29	Padre Las Casas	456	1	2
## 187	Male	2015-01-03	Villarrica	353	1	1
## 188	Male	2015-01-19	Vilcún	157	1	1
## 189	Male	2015-01-25	Padre Las Casas	270	1	1
## 190	Male	2015-02-02	Teodoro Schmidt	466	1	1
## 191	Male	2015-03-06	Nueva Imperial	458	1	1
## 192	Male	2015-03-10	Galvarino	181	1	1
## 193	Male	2015-03-11	Temuco	387	1	1
## 194	Male	2015-03-13	Temuco	256	1	1
## 195	Male	2015-05-02	Temuco	376	1	1
##	aut_rank.x	row_id	id.y	autism.y	ses_status.y	aut_rank.y
## 1	1	21	21282495-K	1	2	1
## 2	1	79	21449127-3	1	2	1
## 3	1	280	21994583-3	1	1	1

## 4	1	345	22183641-3	1	2	1
## 5	1	356	22213761-6	1	2	1
## 6	1	362	22234827-7	1	2	1
## 7	1	545	22724176-4	1	1	1
## 8	1	557	22752332-8	1	1	1
## 9	1	665	23021556-1	1	2	1
## 10	1	675	23054104-3	1	2	1
## 11	1	787	23310188-5	1	2	1
## 12	1	934	23624343-5	1	2	1
## 13	1	1027	23860402-8	1	1	1
## 14	1	1044	23917587-2	1	2	1
## 15	1	1071	23959967-2	1	3	1
## 16	1	1082	23987283-2	1	1	1
## 17	1	1124	24064290-5	1	2	1
## 18	1	1214	24249709-0	1	1	1
## 19	1	1237	24307066-K	1	2	1
## 20	1	1263	24396036-3	1	2	1
## 21	1	1294	24495784-6	1	2	1
## 22	1	1311	24539730-5	1	1	1
## 23	1	1405	24763669-2	1	2	1
## 24	1	1409	24771215-1	1	2	1
## 25	1	1416	24797188-2	1	1	1
## 26	1	1428	24825751-2	1	2	1
## 27	1	1425	24824555-7	1	2	1
## 28	1	45	21338851-7	1	2	1
## 29	1	14	21251752-6	1	1	1
## 30	1	36	21319146-2	1	1	1
## 31	1	37	21319994-3	1	2	1
## 32	1	42	21332821-2	1	1	1
## 33	1	48	21354095-5	1	2	1
## 34	1	74	21417599-1	1	1	1
## 35	1	85	21464033-3	1	2	1
## 36	1	98	21520695-5	1	2	1
## 37	1	104	21543736-1	1	2	1
## 38	1	132	21619878-6	1	2	1
## 39	1	154	21670184-4	1	2	1
## 40	1	157	21679874-0	1	2	1
## 41	1	166	21700914-6	1	1	1
## 42	1	183	21737462-6	1	2	1
## 43	1	187	21748664-5	1	2	1
## 44	1	189	21750199-7	1	2	1
## 45	1	193	21759050-7	1	1	1
## 46	1	230	21859877-3	1	2	1
## 47	1	233	21867880-7	1	2	1
## 48	1	232	21862073-6	1	2	1
## 49	1	249	21921022-1	1	1	1
## 50	1	251	21925304-4	1	2	1
## 51	1	301	22065375-7	1	1	1
## 52	1	305	22079654-K	1	2	1
## 53	1	313	22095157-K	1	2	1
## 54	1	350	22204715-3	1	1	1
## 55	1	355	22211545-0	1	2	1
## 56	1	361	22226291-7	1	2	1
## 57	1	363	22237373-5	1	1	1

## 58	1	368	22245810-2	1	1	1
## 59	1	369	22249166-5	1	2	1
## 60	1	372	22253752-5	1	2	1
## 61	1	373	22253904-8	1	2	1
## 62	1	396	22300065-7	1	2	1
## 63	1	400	22312842-4	1	2	1
## 64	1	410	22327040-9	1	2	1
## 65	1	419	22356979-K	1	2	1
## 66	1	426	22370213-9	1	2	1
## 67	1	433	22386477-5	1	2	1
## 68	1	436	22395859-1	1	2	1
## 69	1	441	22426890-4	1	2	1
## 70	1	464	22491627-2	1	2	1
## 71	1	479	22549846-6	1	2	1
## 72	1	497	22592217-9	1	1	1
## 73	1	508	22637968-1	1	2	1
## 74	1	515	22663017-1	1	2	1
## 75	1	522	22670294-6	1	1	1
## 76	1	525	22678488-8	1	1	1
## 77	1	544	22723986-7	1	1	1
## 78	1	559	22755037-6	1	2	1
## 79	1	579	22805100-4	1	1	1
## 80	1	597	22838644-8	1	2	1
## 81	1	607	22852889-7	1	1	1
## 82	1	613	22881315-K	1	1	1
## 83	1	617	22891576-9	1	2	1
## 84	1	622	22901266-5	1	2	1
## 85	1	627	22907807-0	1	2	1
## 86	1	629	22915922-4	1	2	1
## 87	1	630	22920380-0	1	2	1
## 88	1	641	22945155-3	1	2	1
## 89	1	647	22958693-9	1	1	1
## 90	1	662	23006189-0	1	1	1
## 91	1	687	23093195-K	1	2	1
## 92	1	688	23099554-0	1	2	1
## 93	1	692	23111138-7	1	2	1
## 94	1	703	23136875-2	1	2	1
## 95	1	719	23157810-2	1	2	1
## 96	1	749	23216852-8	1	1	1
## 97	1	757	23233498-3	1	2	1
## 98	1	765	23258114-K	1	1	1
## 99	1	772	23266559-9	1	1	1
## 100	1	770	23263729-3	1	1	1
## 101	1	775	23273376-4	1	2	1
## 102	1	796	23330047-0	1	2	1
## 103	1	801	23343300-4	1	2	1
## 104	1	802	23346792-8	1	1	1
## 105	1	817	23378083-9	1	2	1
## 106	1	820	23386130-8	1	1	1
## 107	1	833	23410879-4	1	2	1
## 108	1	836	23423713-6	1	2	1
## 109	1	845	23448369-2	1	2	1
## 110	1	875	23506849-4	1	1	1
## 111	1	870	23501831-4	1	2	1

## 112	1	889	23534842-K	1	2	1
## 113	1	894	23543378-8	1	1	1
## 114	1	903	23559600-8	1	2	1
## 115	1	907	23567468-8	1	2	1
## 116	1	911	23574393-0	1	1	1
## 117	1	935	23625011-3	1	1	1
## 118	1	950	23667140-2	1	2	1
## 119	1	954	23683414-K	1	2	1
## 120	1	963	23713649-7	1	1	1
## 121	1	972	23737580-7	1	2	1
## 122	1	974	23740506-4	1	2	1
## 123	1	996	23785220-6	1	1	1
## 124	1	1001	23794254-K	1	1	1
## 125	1	1002	23795374-6	1	1	1
## 126	1	1020	23843993-0	1	1	1
## 127	1	1038	23896217-K	1	2	1
## 128	1	1039	23900150-5	1	3	1
## 129	1	1051	23929914-8	1	1	1
## 130	1	1074	23967787-8	1	1	1
## 131	1	1075	23968562-5	1	2	1
## 132	1	1076	23969130-7	1	2	1
## 133	1	1085	23994954-1	1	1	1
## 134	1	1092	24005478-7	1	1	1
## 135	1	1097	24014350-K	1	2	1
## 136	1	1104	24026293-2	1	2	1
## 137	1	1118	24058690-8	1	1	1
## 138	1	1131	24073081-2	1	2	1
## 139	1	1140	24092534-6	1	2	1
## 140	1	1141	24093718-2	1	2	1
## 141	1	1157	24121753-1	1	1	1
## 142	1	1145	24107434-K	1	2	1
## 143	1	1163	24139241-4	1	2	1
## 144	1	1170	24152537-6	1	1	1
## 145	1	1181	24182326-1	1	2	1
## 146	1	1180	24180190-K	1	2	1
## 147	1	1184	24190413-K	1	2	1
## 148	1	1191	24204418-5	1	2	1
## 149	1	1196	24210618-0	1	1	1
## 150	1	1201	24230863-8	1	1	1
## 151	1	1215	24251559-5	1	1	1
## 152	1	1225	24281126-7	1	1	1
## 153	1	1227	24286764-5	1	1	1
## 154	1	1230	24291235-7	1	2	1
## 155	1	1245	24324822-1	1	1	1
## 156	1	1272	24417134-6	1	2	1
## 157	1	1276	24426016-0	1	2	1
## 158	1	1279	24447255-9	1	1	1
## 159	1	1498	24989671-3	1	2	1
## 160	1	1313	24540729-7	1	2	1
## 161	1	1312	24540592-8	1	2	1
## 162	1	1313	24540729-7	1	2	1
## 163	1	1312	24540592-8	1	2	1
## 164	1	1334	24598516-9	1	1	1
## 165	1	1336	24599994-1	1	3	1

```
## 166      1  1341 24612954-1      1      2      1
## 167      1  1348 24627145-3      1      2      1
## 168      1  1350 24628839-9      1      1      1
## 169      1  1351 24629598-0      1      1      1
## 170      1  1355 24636672-1      1      2      1
## 171      1  1362 24653340-7      1      2      1
## 172      1  1382 24703686-5      1      1      1
## 173      1  1391 24729625-5      1      1      1
## 174      1  1395 24737432-9      1      1      1
## 175      1  1400 24743808-4      1      2      1
## 176      1  1399 24743802-5      1      1      1
## 177      1  1398 24743750-9      1      1      1
## 178      1  1407 24766324-K      1      2      1
## 179      1  1404 24761476-1      1      2      1
## 180      1  1407 24766324-K      1      2      1
## 181      1  1404 24761476-1      1      2      1
## 182      1  1411 24786561-6      1      1      1
## 183      1  1410 24786417-2      1      2      1
## 184      1  1418 24801153-K      1      2      1
## 185      1  1419 24806938-4      1      2      1
## 186      1  1439 24842142-8      1      3      1
## 187      1  1442 24851058-7      1      1      1
## 188      1  1448 24867787-2      1      3      1
## 189      1  1453 24878818-6      1      1      1
## 190      1  1457 24887657-3      1      1      1
## 191      1  1471 24923775-2      1      1      1
## 192      1  1470 24922934-2      1      2      1
## 193      1  1472 24926007-K      1      2      1
## 194      1  1474 24927693-6      1      1      1
## 195      1  1491 24972952-3      1      2      1
```

```
length(unique(merged$id.x))
```

```
## [1] 187
```

```
length(unique(merged$id.y))
```

```
## [1] 191
```

187 unique school records can be perfectly matched to clinical records, representing 191 patients.

Probabilistic record linkage

<https://rpubs.com/ahmademad/RecordLinkage> <https://www.bristol.ac.uk/media-library/sites/cmm/migrated/documents/problinkage.pdf> <https://cran.r-project.org/web/packages/diyar/vignettes/links.html>

Mismatch on ses is slightly higher weighted than match on everything. Unclear why and doesn't occur for epiWeights() below.

```
# Try supplying error information. Works better when sex_desc and dob are both in blocking as otherwise
# Still quick for whole school dataset
a2 <- compare.linkage(school,
  #select(school, -ses_status),
  select(patients, -row_id),
  #select(patients, -ses_status),
  blockfld = c("sex_desc", "dob"), # Block on sex and dob because we really want the
```



```

        #blockfld = FALSE,
        phonetic = FALSE,
        strcmp = c(2), # Do string comparison on DOB
        exclude = c(1) # Exclude the id column in both datasets
    )
a2_pairs <- a2$pairs # Issue with ses matching here
b2 <- epiWeights(a2, e = c(0.01, # Default for DOB
                           0.01, # Default for sex
                           0.01, # Default for commune because we want a good match
                           0.01, # Keep small so autism in clinical (not intellectual disability) is pr
                           0.4, # Have more error for ses_status because it is loosely defined
                           #0.3, # Allow more mismatch intellectual disability status so that autism ma
                           0.01 # Allow some mismatch on whether autism is the primary diagnosis so we
    ))
summary(b2)

```

```

##
## Linkage Data Set
##
## 488 records in data set 1
## 1702 records in data set 2
## 299 record pairs
##
## 0 matches
## 0 non-matches
## 299 pairs with unknown status
##
##
## Weight distribution:
##
## [0.55,0.6] [0.6,0.65] [0.65,0.7] [0.7,0.75] [0.75,0.8] [0.8,0.85] [0.85,0.9]
##           5         64         35         0         0         110         85

```

```

allPairs2 <- getPairs(b2)
head(allPairs2, n = 20)

```

	id	id	dob	sex_desc	student_commune_name	autism	ses_status
## 1	437	437	2003-11-25	Female	Temuco	1	2
## 2	79	21449127-3	2003-11-25	Female	Temuco	1	2
## 3							
## 4	380	380	2005-12-07	Female	Temuco	1	1
## 5	280	21994583-3	2005-12-07	Female	Temuco	1	1
## 6							
## 7	187	187	2008-05-20	Female	Gorbea	1	1
## 8	545	22724176-4	2008-05-20	Female	Gorbea	1	1
## 9							
## 10	269	269	2008-06-21	Female	Temuco	1	1
## 11	557	22752332-8	2008-06-21	Female	Temuco	1	1
## 12							
## 13	173	173	2011-04-20	Female	Temuco	1	2
## 14	934	23624343-5	2011-04-20	Female	Temuco	1	2
## 15							
## 16	172	172	2012-01-31	Female	Villarrica	1	1
## 17	1027	23860402-8	2012-01-31	Female	Villarrica	1	1

```
## 18
## 19 41 41 2012-06-18 Female Villarrica 1 1
## 20 1082 23987283-2 2012-06-18 Female Villarrica 1 1
## aut_rank Weight
## 1 1
## 2 1 0.8882294
## 3
## 4 1
## 5 1 0.8882294
## 6
## 7 1
## 8 1 0.8882294
## 9
## 10 1
## 11 1 0.8882294
## 12
## 13 1
## 14 1 0.8882294
## 15
## 16 1
## 17 1 0.8882294
## 18
## 19 1
## 20 1 0.8882294
```

```
classifyPairs2 <- emClassify(b2, threshold.upper = 1, threshold.lower = 0.8)
a2_pairs$weight <- classifyPairs2$Wdata
a2_pairs$pred <- classifyPairs2$prediction
```

```
a2_pairs_clean <- a2_pairs %>%
  rename(".x" = id1, ".y" = id2) %>%
  select(-is_match)
```

```
finalPairs2 <- getPairs(b2, max.weight = 1, min.weight = 0, single.rows = TRUE) # Take them all when bl
```

```
#kmeansRes2 <- classifyUnsup(a2, method = "kmeans")
```

```
#a2_pairs$pred <- kmeansRes2$prediction
```

```
# Works but prioritises ses over commune and doesn't use epiWeights found above so not that useful.
```

finalPairs2 is the same size as finalPairs and probably contains the same matches but was much quicker to run because of the blocking. Assume in kmeansRes2, N = not a match, L = likely a match.

```
# reclin has a 1-1 matching fuction so regenerate the pairs using reclin so they're a pairs
# type object and can be passed to select_n_to_m
```

```
pairs <- pair_blocking(school, patients, on = c("sex_desc", "dob")) %>%
  mutate(student_commune_name = (school$student_commune_name[.x] == patients$student_commune_name[.y])
  #ses = get_num_diff(school$ses_status[.x], patients$ses_status[.y])$val
  ) %>%
  left_join(a2_pairs_clean, by = c(".x", ".y")) %>%
  select(c(-student_commune_name.x)) %>%
  rename("student_commune_name" = "student_commune_name.y")
```

```
matches <- select_n_to_m(pairs, threshold = 0.5, score = "weight", n = 1, m = 1, var = "match") %>%
```

```

filter(match == TRUE) %>%
rename("id" = ".x",
      "row_id" = ".y") %>%
mutate(id = as.character(id))

# Now add the matched clinical records to the school records
school_matched <- school %>%
  filter(student_commune_name != "Misc") %>%
  left_join(matches, by = "id") %>%
  rename(id.school = id,
        dob.school = dob.x,
        sex_desc.school = sex_desc.x,
        student_commune_name.school = student_commune_name.x,
        ses_status.school = ses_status.x,
        dob.matched = dob.y,
        sex_desc.matched = sex_desc.y,
        student_commune_name.matched = student_commune_name.y,
        ses_status.matched = ses_status.y) %>%
  select(c(-pred, -match)) %>%
  left_join(patients, by = "row_id") %>%
  rename(id.patient = row_id,
        patient_id = id,
        dob.patient = dob,
        sex_desc.patient = sex_desc,
        student_commune_name.patient = student_commune_name,
        ses_status.patient = ses_status) %>%
  select(id.school, id.patient, patient_id,
        dob.school, dob.patient, dob.matched,
        sex_desc.school, sex_desc.patient, sex_desc.matched,
        student_commune_name.school, student_commune_name.patient, student_commune_name.matched,
        ses_status.school, ses_status.patient, ses_status.matched,
        weight) %>%
  arrange(desc(weight))

write_csv(school_matched, "04_Data/Outputs/school_matched.csv")

#school_matched_yes <- school_matched %>% filter(!is.na(weight))
#school_matched_no <- school_matched %>% filter(is.na(weight))

# commune_nums <- data.frame(student_commune_name.school = sort(unique(school_matched$student_commune_name.school)),
#                             commune_num = c(1:length(unique(school_matched$student_commune_name.school))))

school_matched_small <- school_matched %>%
  mutate(matched = ifelse(is.na(patient_id), 0, 1),
        sex.school = ifelse(sex_desc.school == "Male", 1, ifelse(sex_desc.school == "Female", 2, NA)))
merge(commune_region_lookup, by.x = "student_commune_name.school", by.y = "commune_name") %>% # doesn't work
select(id.school, dob.school, sex_desc.school, sex.school, student_commune_name.school, commune_code,

# Now add the matched clinical records to the school records
patients_matched <- patients %>%
  left_join(matches, by = "row_id") %>%
  rename(id.patient = row_id,
        patient_id = id.x,
        dob.patient = dob.x,

```

```

sex_desc.patient = sex_desc.x,
student_commune_name.patient = student_commune_name.x,
id = id.y,
ses_status.patient = ses_status.x,
dob.matched = dob.y,
sex_desc.matched = sex_desc.y,
student_commune_name.matched = student_commune_name.y,
ses_status.matched = ses_status.y) %>%
select(c(-pred, -match)) %>%
left_join(school, by = "id") %>%
rename(id.school = id,
       dob.school = dob,
       sex_desc.school = sex_desc,
       student_commune_name.school = student_commune_name,
       ses_status.school = ses_status) %>%
select(id.school, id.patient, patient_id,
       dob.school, dob.patient, dob.matched,
       sex_desc.school, sex_desc.patient, sex_desc.matched,
       student_commune_name.school, student_commune_name.patient, student_commune_name.matched,
       ses_status.school, ses_status.patient, ses_status.matched,
       weight) %>%
arrange(desc(weight))

write_csv(patients_matched, "04_Data/Outputs/patients_matched.csv")

patients_matched_small <- patients_matched %>%
  mutate(matched = ifelse(is.na(id.school), 0, 1),
         sex.patient = ifelse(sex_desc.patient == "Male", 1, ifelse(sex_desc.patient == "Female", 2, NA)),
         merge(commune_region_lookup, by.x = "student_commune_name.patient", by.y = "commune_name") %>%
  select(patient_id, id.patient, dob.patient, sex_desc.patient, sex.patient, student_commune_name.patient)

```

Check patients that lived in different communes therefore were in the patient dataset twice aren't matched to multiple school records

```

patients_matched_unique <- patients_matched_small %>%
  group_by(matched, patient_id) %>%
  summarise(count = n())

## `summarise()` has grouped output by 'matched'. You can override using the
## `.groups` argument.

patients_dup <- patients_matched_unique %>% filter(matched == 1, count > 1) %>% select(patient_id)

## Adding missing grouping variables: `matched`
# No patient is inadvertently matched to two school records

```

Consider whether the matched and unmatched school records are different

We hope they are not different

```

#library(coin)

#pt.sex <- oneway_test(sex.school ~ as.factor(matched), data = school_matched_small, distribution = app

```

```

#confint(pt.sex)

#ks.ses <- ks.test(data1$ses_status.school, data2$ses_status.school, alternative = "two.sided", simulate.p.value = TRUE)
#ks.ses

# SES
#data1 <- school_matched_yes %>% select(ses_status.school)
#data2 <- school_matched_no %>% select(ses_status.school)
#hist(data1$ses_status.school, breaks = 10)
#hist(data2$ses_status.school, breaks = 10)
#data1 %>% group_by(ses_status.school) %>% summarise(count = n()) %>% mutate(freq = count/sum(count))
#data2 %>% group_by(ses_status.school) %>% summarise(count = n()) %>% mutate(freq = count/sum(count))

school_yes <- school_matched_small %>% filter(matched == 1) #>% select(sex.school)
school_no <- school_matched_small %>% filter(matched == 0)

# Kolmogorov tests for our matched results
ks.school.sex <- ks.test(na.omit(school_yes$sex.school),
                        na.omit(school_no$sex.school),
                        alternative = "two.sided", simulate.p.value = TRUE)
ks.school.sex

##
## Two-sample Kolmogorov-Smirnov test
##
## data: na.omit(school_yes$sex.school) and na.omit(school_no$sex.school)
## D = 0.012867, p-value = 1
## alternative hypothesis: two-sided

ks.school.ses_status <- ks.test(as.numeric(na.omit(school_yes$ses_status.school)),
                                as.numeric(na.omit(school_no$ses_status.school)),
                                alternative = "two.sided", simulate.p.value = TRUE)
ks.school.ses_status

##
## Two-sample Kolmogorov-Smirnov test
##
## data: as.numeric(na.omit(school_yes$ses_status.school)) and as.numeric(na.omit(school_no$ses_status.school))
## D = 0.094516, p-value = 0.2336
## alternative hypothesis: two-sided

ks.school.commune_code <- ks.test(as.numeric(na.omit(school_yes$commune_code)),
                                  as.numeric(na.omit(school_no$commune_code)),
                                  alternative = "two.sided", simulate.p.value = TRUE)
ks.school.commune_code

##
## Two-sample Kolmogorov-Smirnov test
##
## data: as.numeric(na.omit(school_yes$commune_code)) and as.numeric(na.omit(school_no$commune_code))
## D = 0.20485, p-value = 7.395e-05
## alternative hypothesis: two-sided

# Try manual Kolmogorov for SES
# bins <- unique(na.omit(school_matched_small$ses_status.school))
# ecdf.ses_status.yes <- ecdf(school_yes$ses_status.school)

```

```

# ecdf.ses_status.yes(schoolyes$ses_status.school)
# ecdf.ses_status.no <- ecdf(schoolno$ses_status.school)
# plot(ecdf.ses_status.yes) ; plot(ecdf.ses_status.no)

# Kolmogorov tests with permutation distributions
set.seed(123)
nPerm <- 200 # change to 2000
ks_perm.school.pvals <- data.frame(sex = numeric(nPerm),
                                   commune_code = numeric(nPerm),
                                   ses_status = numeric(nPerm))

school_matched_small_perm <- school_matched_small

for (i in 1:nPerm) {
  #print(i)
  school_matched_small_perm$matched <- school_matched_small$matched[sample(nrow(school_matched_small))]
  school_perm_yes <- school_matched_small_perm %>% filter(matched == 1)
  school_perm_no <- school_matched_small_perm %>% filter(matched == 0)

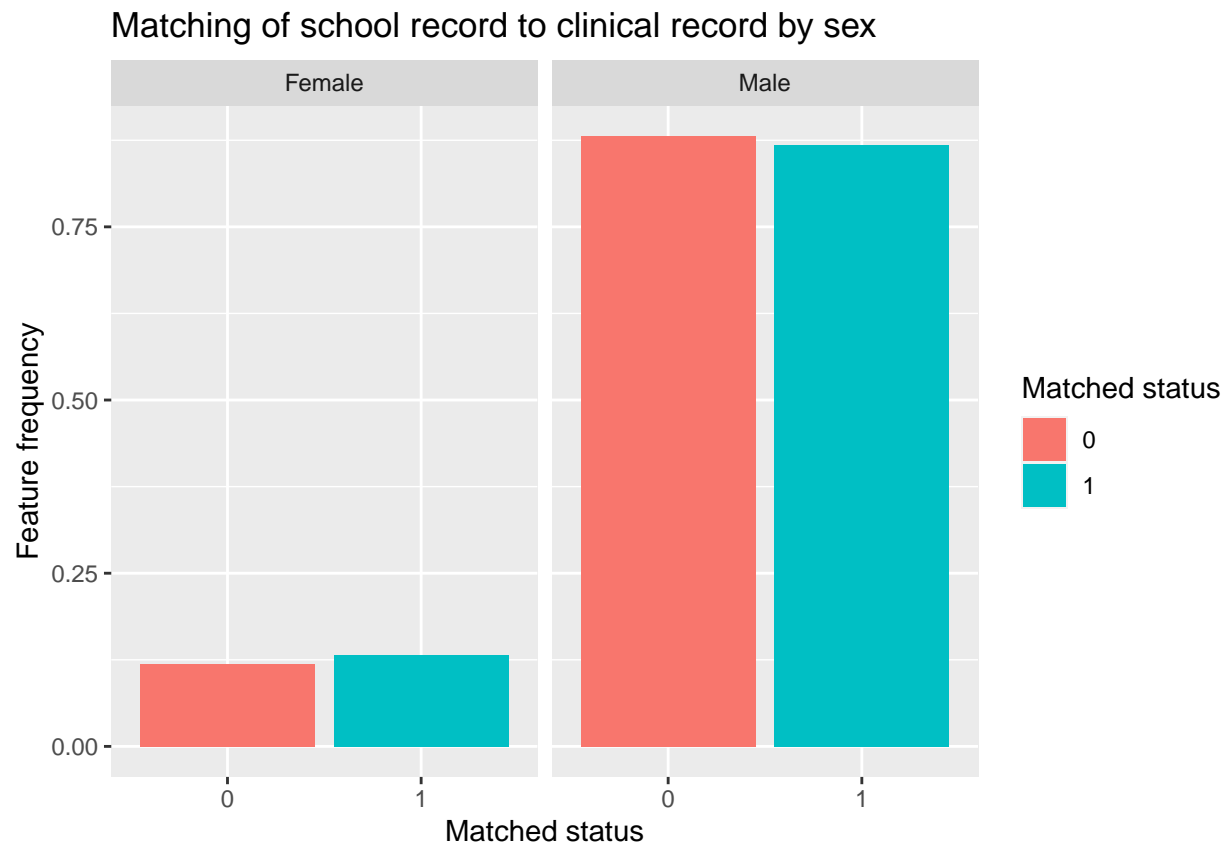
  ks_perm.school.sex <- ks.test(na.omit(school_perm_yes$sex.school),
                               na.omit(school_perm_no$sex.school),
                               alternative = "two.sided")
  ks_perm.school.commune_code <- ks.test(as.numeric(na.omit(school_perm_yes$commune_code)),
                                         as.numeric(na.omit(school_perm_no$commune_code)),
                                         alternative = "two.sided")
  ks_perm.school.ses_status <- ks.test(as.numeric(na.omit(school_perm_yes$ses_status.school)),
                                       as.numeric(na.omit(school_perm_no$ses_status.school)),
                                       alternative = "two.sided")

  ks_perm.school.pvals$sex[i] <- ks_perm.school.sex$p.value
  ks_perm.school.pvals$commune_code[i] <- ks_perm.school.commune_code$p.value
  ks_perm.school.pvals$ses_status[i] <- ks_perm.school.ses_status$p.value
}

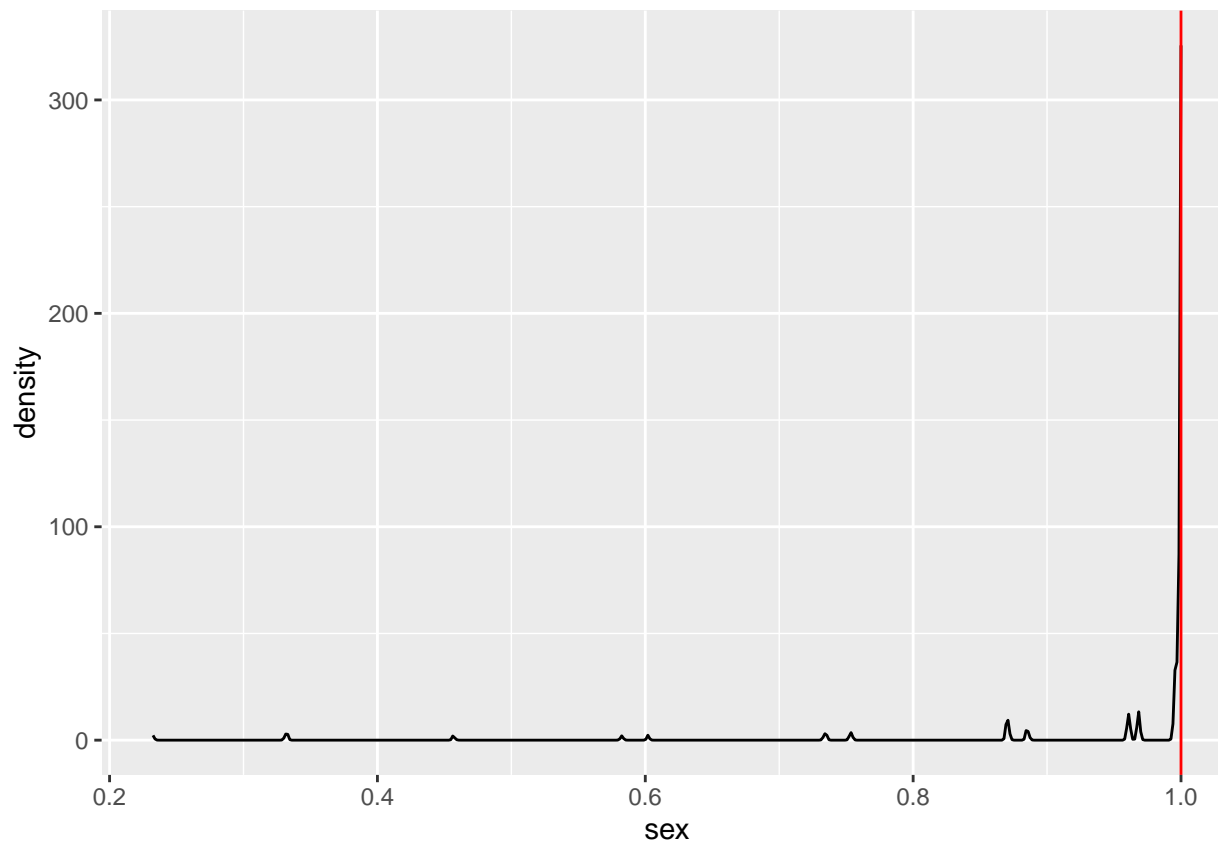
# Results for sex
school_match_yes.sex <- school_yes %>% group_by(sex.school) %>% summarise(count = n()) %>% mutate(freq = count)
school_match_no.sex <- school_no %>% group_by(sex.school) %>% summarise(count = n()) %>% mutate(freq = count)
school_match.sex <- rbind(school_match_yes.sex, school_match_no.sex) %>%
  mutate(sex_desc = ifelse(sex.school == 1, "Male", ifelse(sex.school == 2, "Female", NA))) %>%
  arrange(sex_desc, matched)

ggplot(school_match.sex) +
  geom_col(aes(x = as.factor(matched), y = freq, fill = as.factor(matched))) +
  facet_wrap(~sex_desc) +
  labs(title = "Matching of school record to clinical record by sex",
       x = "Matched status",
       y = "Feature frequency",
       fill = "Matched status")

```



```
ggplot(ks_perm.school.pvals, aes(x = sex, y = after_stat(density))) +  
  geom_density() +  
  geom_vline(xintercept = ks.school.sex$p.value, color = "red")
```

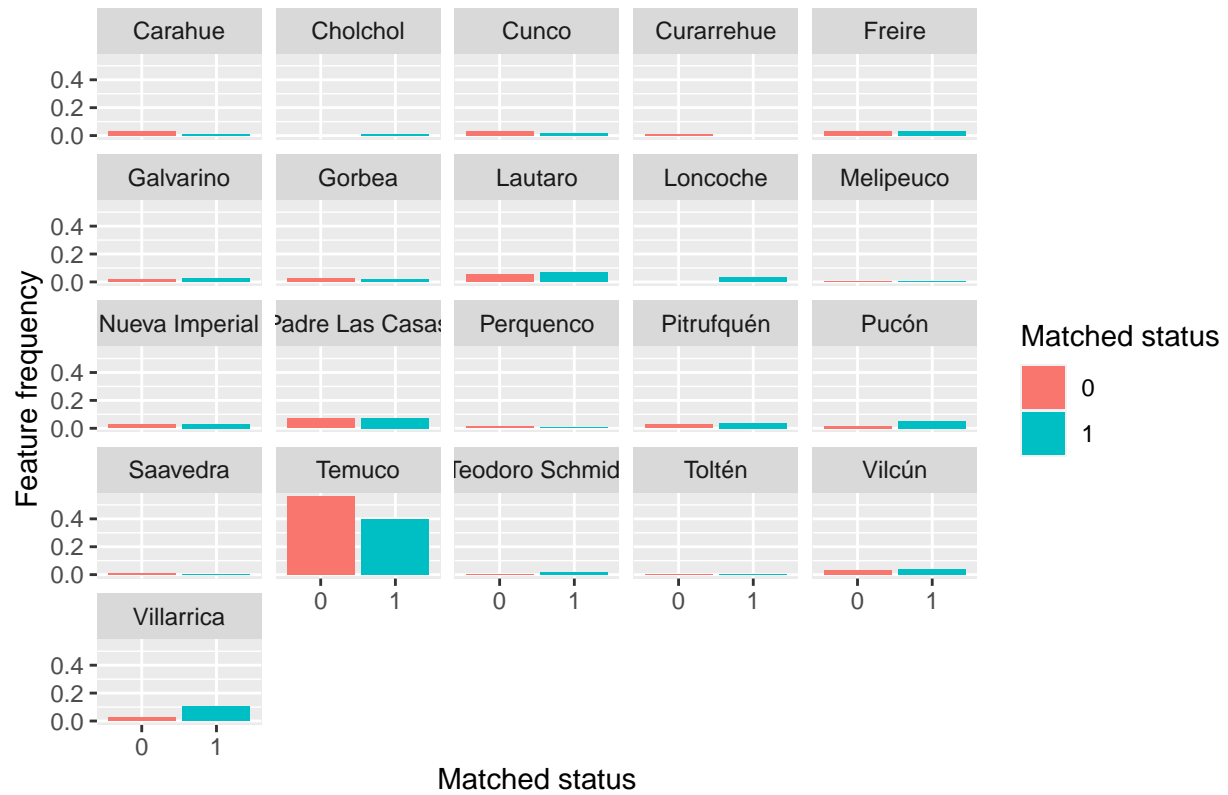



```
# Results for commune
school_match_yes.student_commune_name <- school_yes %>% group_by(student_commune_name.school) %>%
  summarise(count = n()) %>% mutate(freq = count/sum(count)) %>%
  # Would need to merge to a list of commune names and numbers if want to display all communes for all
#merge(commune_, by = "commune_num", all = TRUE) %>%
  mutate(matched = 1)
school_match_no.student_commune_name <- school_no %>% group_by(student_commune_name.school) %>%
  summarise(count = n()) %>% mutate(freq = count/sum(count)) %>%
  #merge(commune_nums, by = "commune_num", all = TRUE) %>%
  mutate(matched = 0)

school_match.student_commune_name <- rbind(school_match_yes.student_commune_name, school_match_no.student_commune_name)
arrange(student_commune_name.school, matched)

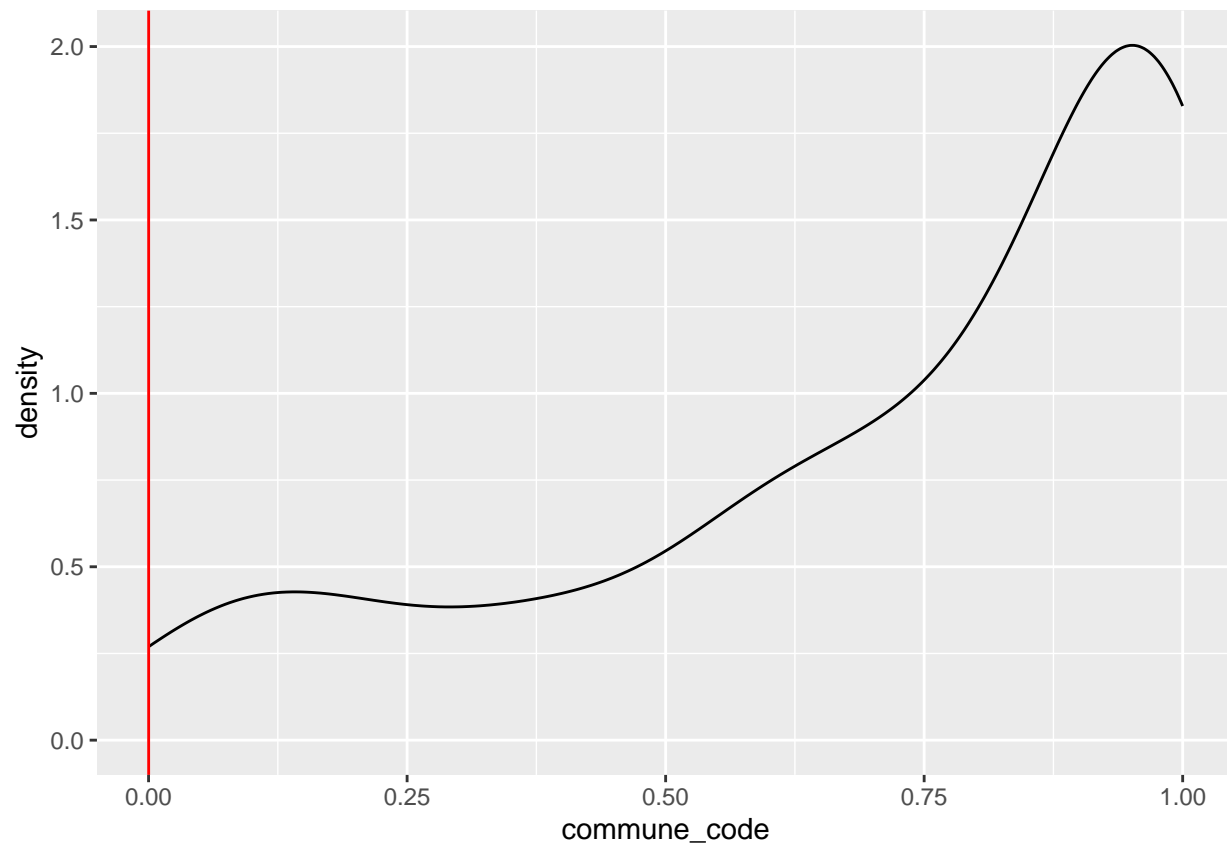
ggplot(school_match.student_commune_name) +
  geom_col(aes(x = as.factor(matched), y = freq, fill = as.factor(matched))) +
  facet_wrap(~student_commune_name.school, scales = "fixed") +
  #facet_wrap(~student_commune_name.school, scales = "free") +
  labs(title = "Matching of school record to clinical record by commune",
       x = "Matched status",
       y = "Feature frequency",
       fill = "Matched status")
```

Matching of school record to clinical record by commune



most of the difference in matched commune frequency is for Temuco which is the biggest commune.

```
ggplot(ks_perm.school.pvals, aes(x = commune_code, y = after_stat(density))) +
  geom_density() +
  geom_vline(xintercept = ks.school.commune_code$p.value, color = "red")
```

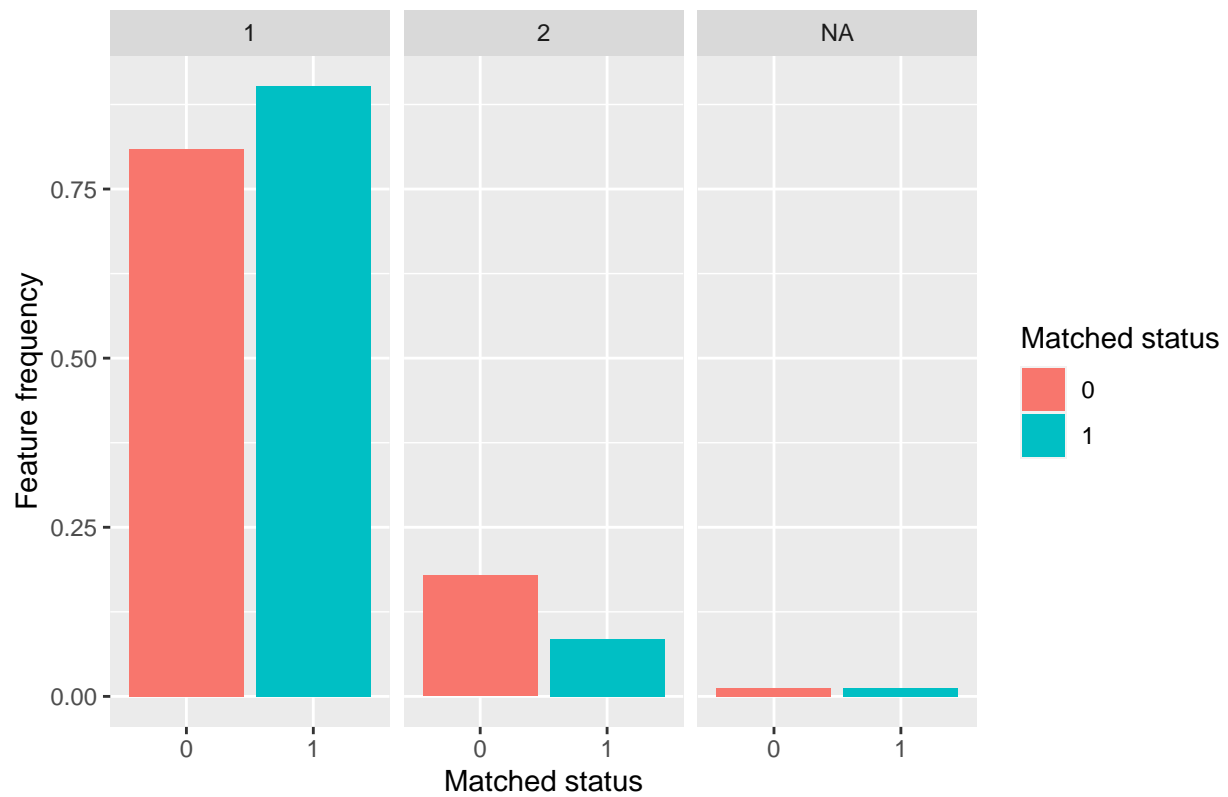


```
# Results for ses status
```

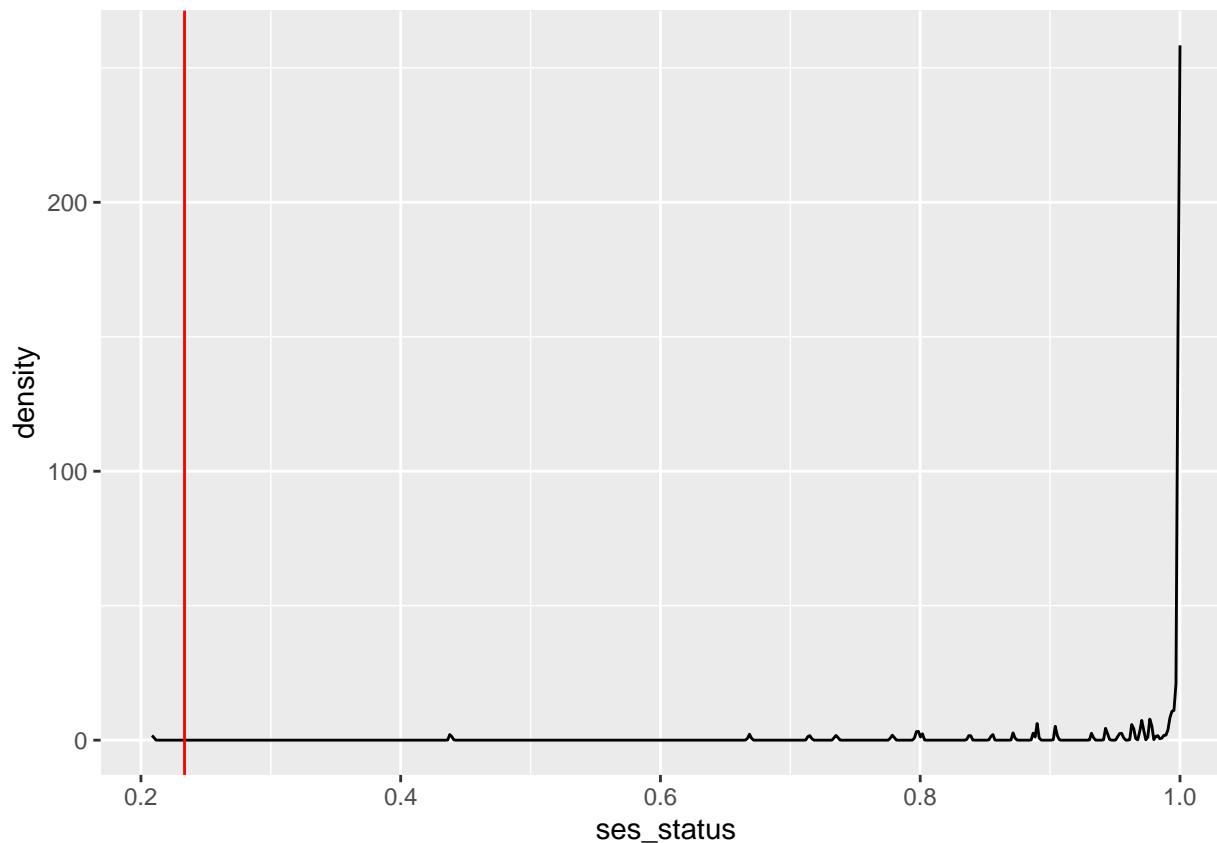
```
school_match_yes.ses_status <- school_yes %>% group_by(ses_status.school) %>% summarise(count = n()) %>%
school_match_no.ses_status <- school_no %>% group_by(ses_status.school) %>% summarise(count = n()) %>%
school_match.ses_status <- rbind(school_match_yes.ses_status, school_match_no.ses_status) %>%
  arrange(ses_status.school, matched)
```

```
ggplot(school_match.ses_status) +
  geom_col(aes(x = as.factor(matched), y = freq, fill = as.factor(matched))) +
  facet_wrap(~ses_status.school) +
  labs(title = "Matching of school record to clinical record by SES status",
       x = "Matched status",
       y = "Feature frequency",
       fill = "Matched status")
```

Matching of school record to clinical record by SES status



```
ggplot(ks_perm.school.pvals, aes(x = ses_status, y = after_stat(density))) +
  geom_density() +
  geom_vline(xintercept = ks.school.ses_status$p.value, color = "red")
```



Bit easier to match SES status of 1 (probably more common)

Our matched/non-matched are not different by sex (p-value in Kolmog is same as most of distribution of permuted pvals) but are different by commune and ses status. Cohen's D test isn't suitable to compare the matched and un-matched because the data don't have standard deviations.

??Add commune maps here with size of sample for school and clinical?? Also size of other features.

```
patients_yes <- patients_matched_small %>% filter(matched == 1) #>% select(sex.school)
patients_no <- patients_matched_small %>% filter(matched == 0)
```

```
# Kolmogorov tests for our matched results
```

```
ks.patients.sex <- ks.test(na.omit(patients_yes$sex.patient),
                           na.omit(patients_no$sex.patient),
                           alternative = "two.sided", simulate.p.value = TRUE)
```

```
ks.patients.sex
```

```
##
```

```
## Two-sample Kolmogorov-Smirnov test
```

```
##
```

```
## data: na.omit(patients_yes$sex.patient) and na.omit(patients_no$sex.patient)
```

```
## D = 0.099851, p-value = 0.03523
```

```
## alternative hypothesis: two-sided
```

```
ks.patients.ses_status <- ks.test(as.numeric(na.omit(patients_yes$ses_status.patient)),
                                   as.numeric(na.omit(patients_no$ses_status.patient)),
                                   alternative = "two.sided", simulate.p.value = TRUE)
```

```
ks.patients.ses_status
```

```

##
## Two-sample Kolmogorov-Smirnov test
##
## data: as.numeric(na.omit(patients_yes$ses_status.patient)) and as.numeric(na.omit(patients_no$ses_s
## D = 0.065741, p-value = 0.3454
## alternative hypothesis: two-sided

ks.patients.commune_code<- ks.test(as.numeric(na.omit(patients_yes$commune_code)),
                                   as.numeric(na.omit(patients_no$commune_code)),
                                   alternative = "two.sided", simulate.p.value = TRUE)

ks.patients.commune_code

##
## Two-sample Kolmogorov-Smirnov test
##
## data: as.numeric(na.omit(patients_yes$commune_code)) and as.numeric(na.omit(patients_no$commune_cod
## D = 0.083198, p-value = 0.1211
## alternative hypothesis: two-sided

# Kolmogorov tests with permutation distributions
set.seed(123)
nPerm <- 200 # change to 2000
ks_perm.patients.pvals <- data.frame(sex = numeric(nPerm),
                                     commune_code = numeric(nPerm),
                                     ses_status = numeric(nPerm))

patients_matched_small_perm <- patients_matched_small

for (i in 1:nPerm) {
  #print(i)
  patients_matched_small_perm$matched <- patients_matched_small$matched[sample(nrow(patients_matched_sm
  patients_perm_yes <- patients_matched_small_perm %>% filter(matched == 1)
  patients_perm_no <- patients_matched_small_perm %>% filter(matched == 0)

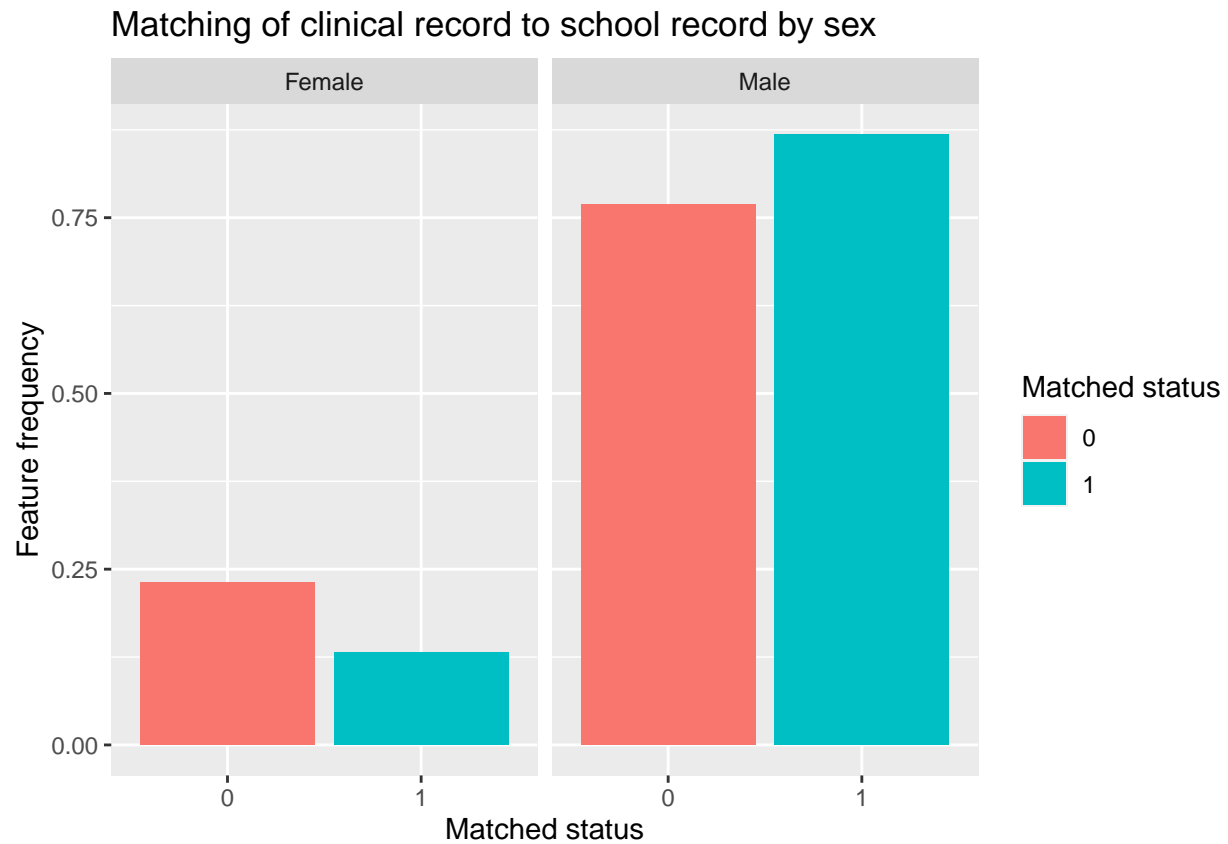
  ks_perm.patients.sex <- ks.test(na.omit(patients_perm_yes$sex.patient),
                                  na.omit(patients_perm_no$sex.patient),
                                  alternative = "two.sided")
  ks_perm.patients.commune_code <- ks.test(as.numeric(na.omit(patients_perm_yes$commune_code)),
                                           as.numeric(na.omit(patients_perm_no$commune_code)),
                                           alternative = "two.sided")
  ks_perm.patients.ses_status <- ks.test(as.numeric(na.omit(patients_perm_yes$ses_status.patient)),
                                          as.numeric(na.omit(patients_perm_no$ses_status.patient)),
                                          alternative = "two.sided")

  ks_perm.patients.pvals$sex[i] <- ks_perm.patients.sex$p.value
  ks_perm.patients.pvals$commune_code[i] <- ks_perm.patients.commune_code$p.value
  ks_perm.patients.pvals$ses_status[i] <- ks_perm.patients.ses_status$p.value
}

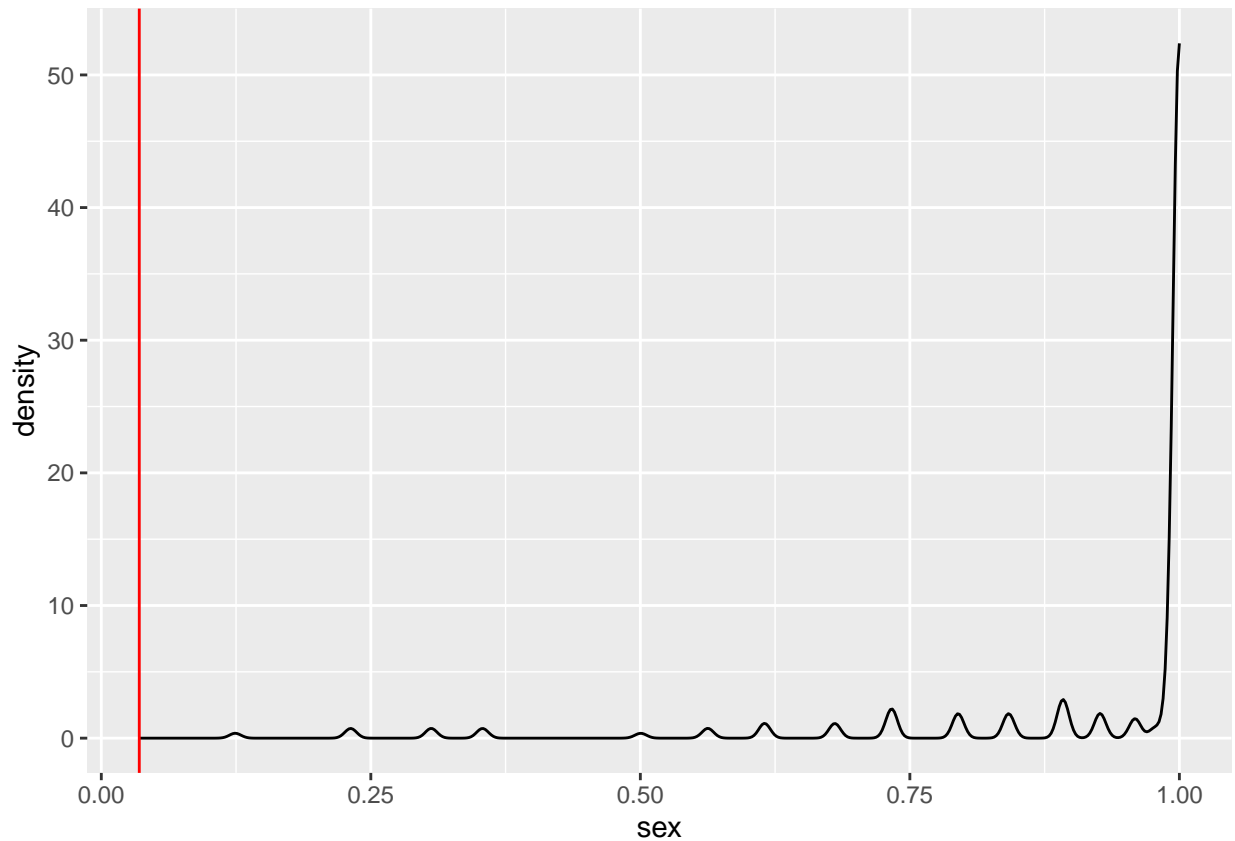
# Results for sex
patients_match_yes.sex <- patients_yes %>% group_by(sex.patient) %>% summarise(count = n()) %>% mutate(fr
patients_match_no.sex <- patients_no %>% group_by(sex.patient) %>% summarise(count = n()) %>% mutate(fr
patients_match.sex <- rbind(patients_match_yes.sex, patients_match_no.sex) %>%
  mutate(sex_desc = ifelse(sex.patient == 1, "Male", ifelse(sex.patient == 2, "Female", NA))) %>%
  arrange(sex_desc, matched)

```

```
ggplot(patients_match.sex) +
  geom_col(aes(x = as.factor(matched), y = freq, fill = as.factor(matched))) +
  facet_wrap(~sex_desc) +
  labs(title = "Matching of clinical record to school record by sex",
       x = "Matched status",
       y = "Feature frequency",
       fill = "Matched status")
```



```
ggplot(ks_perm.patients.pvals, aes(x = sex, y = after_stat(density))) +
  geom_density() +
  geom_vline(xintercept = ks.patients.sex$p.value, color = "red")
```

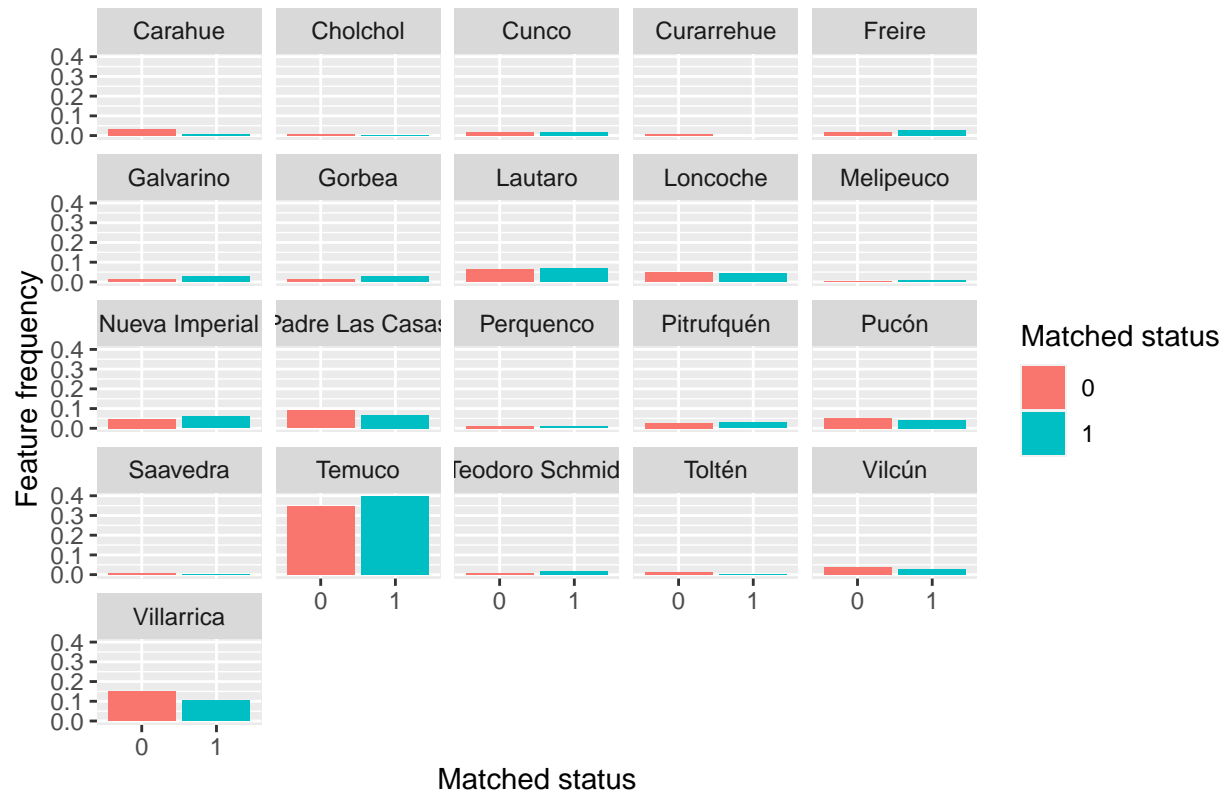



```
# Results for commune
patients_match_yes.student_commune_name <- patients_yes %>% group_by(student_commune_name.patient) %>%
  summarise(count = n()) %>% mutate(freq = count/sum(count)) %>%
  # Would need to merge to a list of commune names and numbers if want to display all communes for all
  #merge(commune_, by = "commune_num", all = TRUE) %>%
  mutate(matched = 1)
patients_match_no.student_commune_name <- patients_no %>% group_by(student_commune_name.patient) %>%
  summarise(count = n()) %>% mutate(freq = count/sum(count)) %>%
  #merge(commune_nums, by = "commune_num", all = TRUE) %>%
  mutate(matched = 0)

patients_match.student_commune_name <- rbind(patients_match_yes.student_commune_name, patients_match_no)
  arrange(student_commune_name.patient, matched)

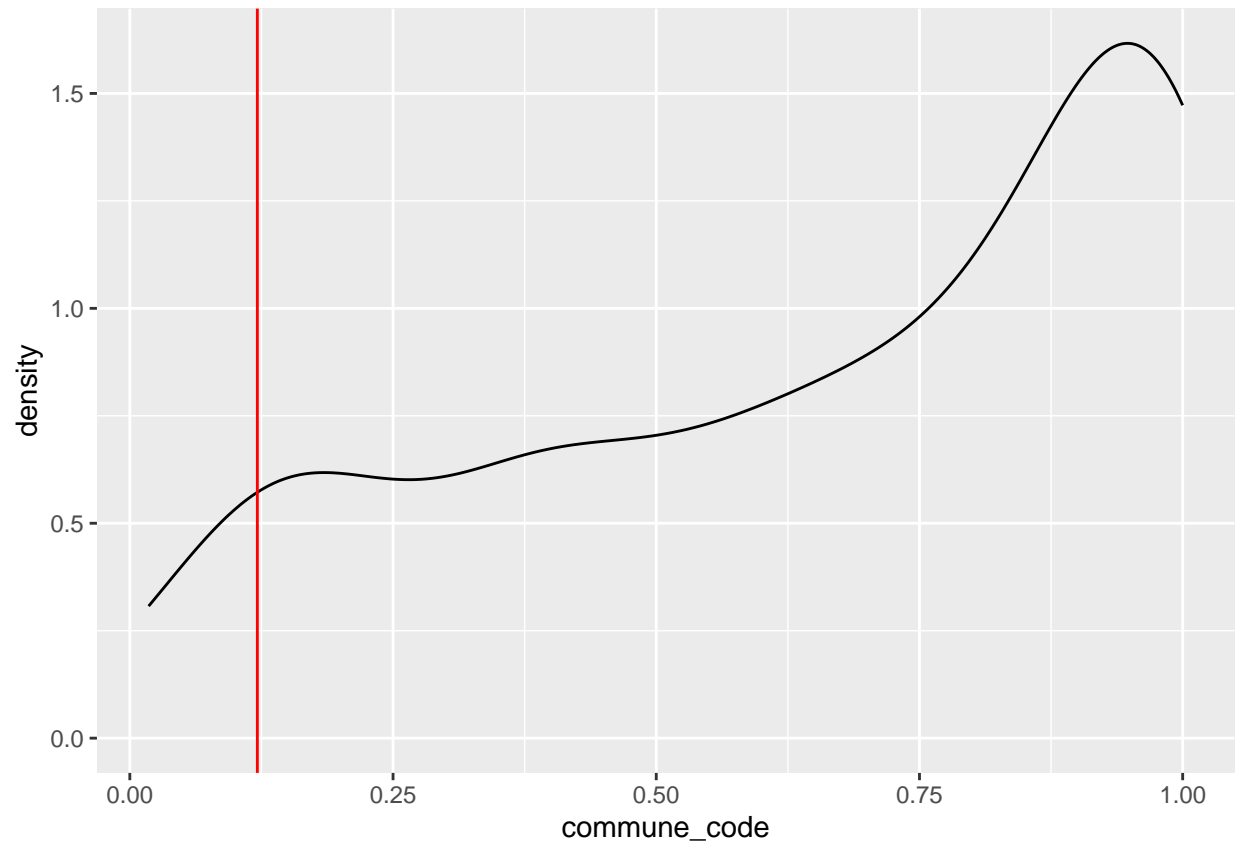
ggplot(patients_match.student_commune_name) +
  geom_col(aes(x = as.factor(matched), y = freq, fill = as.factor(matched))) +
  facet_wrap(~student_commune_name.patient, scales = "fixed") +
  #facet_wrap(~student_commune_name.school, scales = "free") +
  labs(title = "Matching of clinical record to school record by commune",
       x = "Matched status",
       y = "Feature frequency",
       fill = "Matched status")
```

Matching of clinical record to school record by commune



most of the difference in matched commune frequency is for Temuco which is the biggest commune.

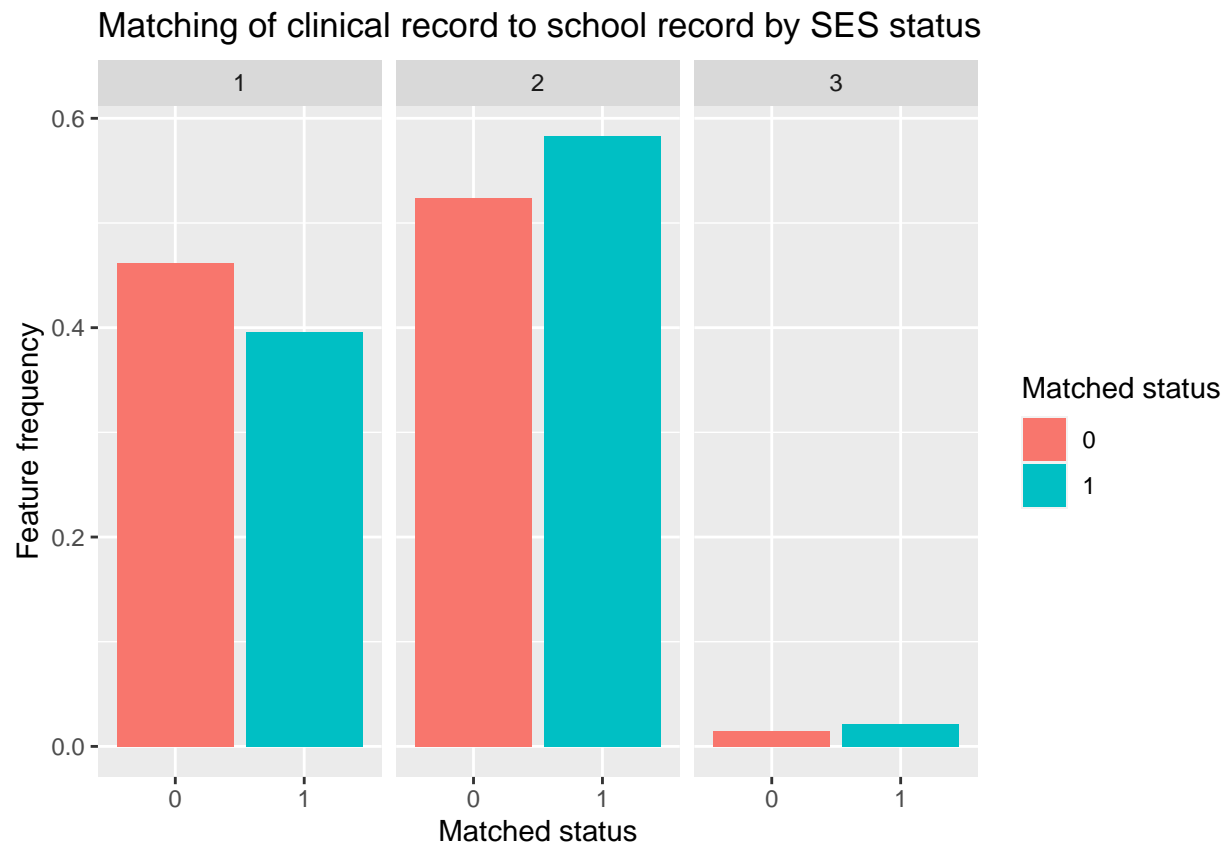
```
ggplot(ks_perm.patients.pvals, aes(x = commune_code, y = after_stat(density))) +
  geom_density() +
  geom_vline(xintercept = ks.patients.commune_code$p.value, color = "red")
```



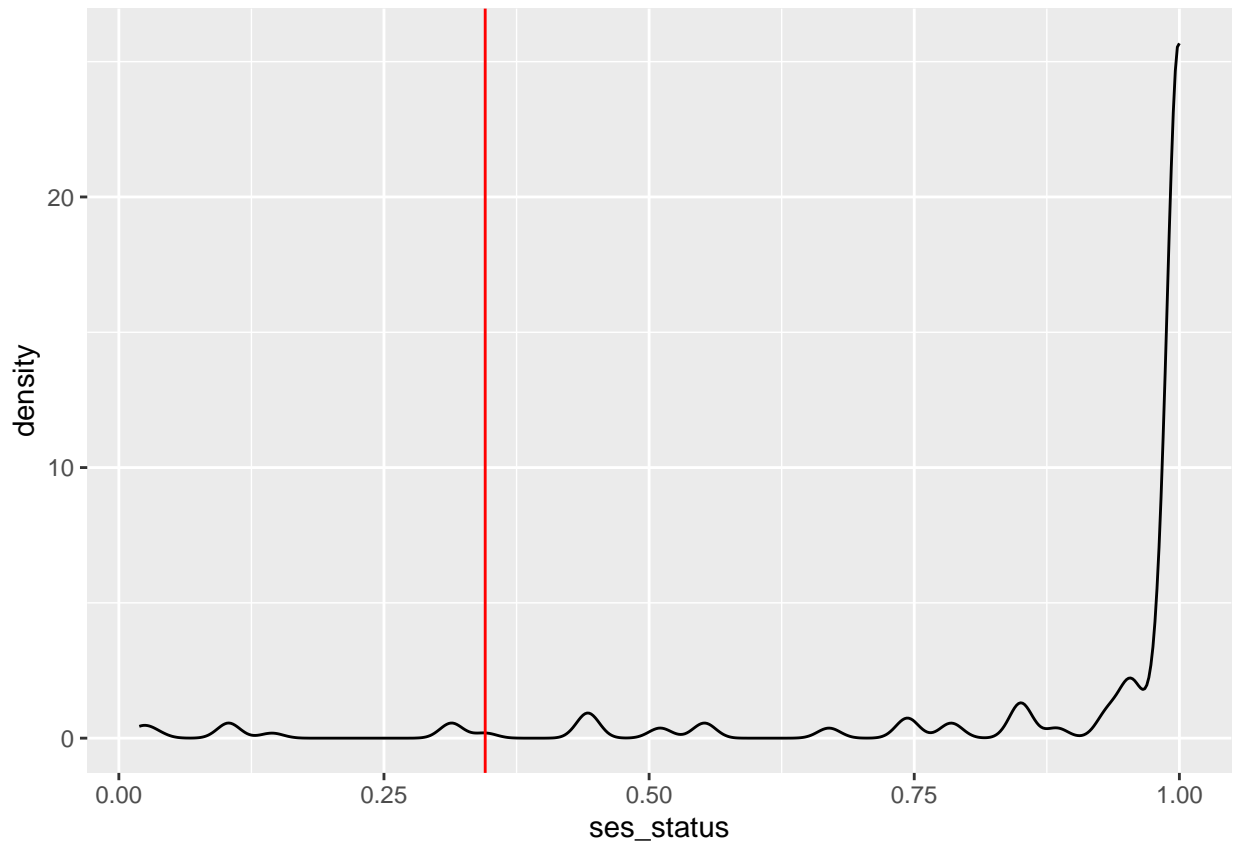
Results for ses status

```
patients_match_yes.ses_status <- patients_yes %>% group_by(ses_status.patient) %>% summarise(count = n())
patients_match_no.ses_status <- patients_no %>% group_by(ses_status.patient) %>% summarise(count = n())
patients_match.ses_status <- rbind(patients_match_yes.ses_status, patients_match_no.ses_status) %>%
  arrange(ses_status.patient, matched)
```

```
ggplot(patients_match.ses_status) +
  geom_col(aes(x = as.factor(matched), y = freq, fill = as.factor(matched))) +
  facet_wrap(~ses_status.patient) +
  labs(title = "Matching of clinical record to school record by SES status",
       x = "Matched status",
       y = "Feature frequency",
       fill = "Matched status")
```



```
ggplot(ks_perm.patients.pvals, aes(x = ses_status, y = after_stat(density))) +  
  geom_density() +  
  geom_vline(xintercept = ks.patients.ses_status$p.value, color = "red")
```



New prevalence estimates

```
n_aut_araucS <- nrow(school_matched) + length(unique(patients_matched$patient_id)) - nrow(matches) # Us
#n_aut_araucS <- school_matched_small %>% filter(matched == 0) %>% nrow() +
# dim(patients_matched_small %>% filter(matched == 0) %>% select(patient_id) %>% unique())[1] + nrow(m
```

Multiple Correspondence Analysis

<https://www.tibco.com/reference-center/what-is-correspondence-analysis#:~:text=Correspondence%20analysis%2C%20also%20>
<https://ladal.edu.au/clust.html> <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3810057/>
<http://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/114-mca-multiple-correspondence-analysis-in-r-essentials/>

```
clinical_mca <- clinical %>%
  mutate(gender = as.factor(gender),
         age_june30 = as.factor(age_june30),
         commune_name = as.factor(commune_name),
         appt_type = as.factor(type_appointment),
         ses_status = as.factor(ifelse(socio_economic_level %in% c("COLMENA GOLDEN CROSS", "RIO BLANCO"),
         hospital = as.factor(hospital),
         medical_specialty = as.factor(medical_specialty)
         ) %>%
  select(gender,
         age_june30,
```

```

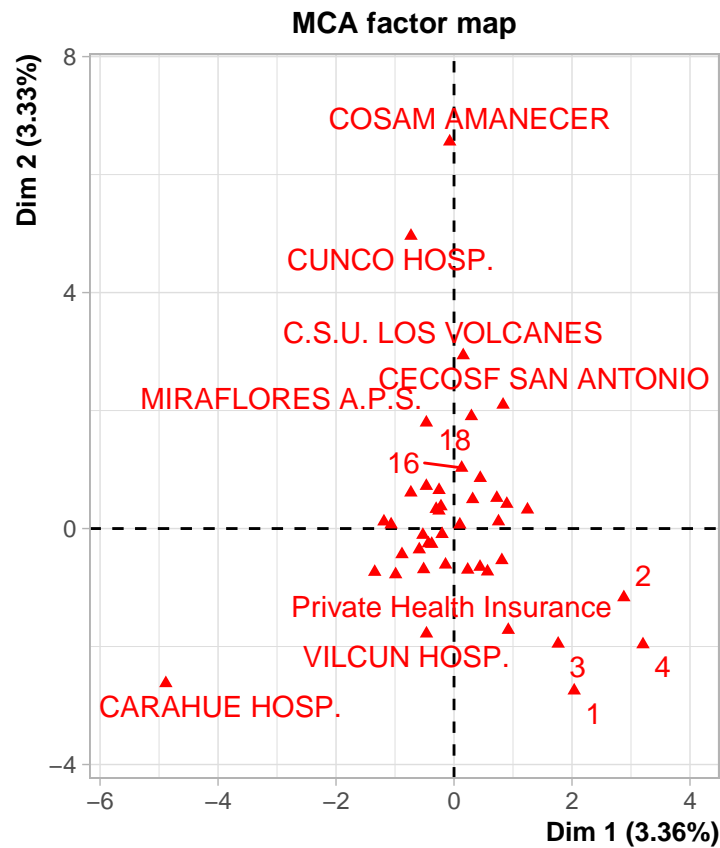
#commune_name,
#appt_type,
ses_status,
hospital) #,
#medical_specialty)

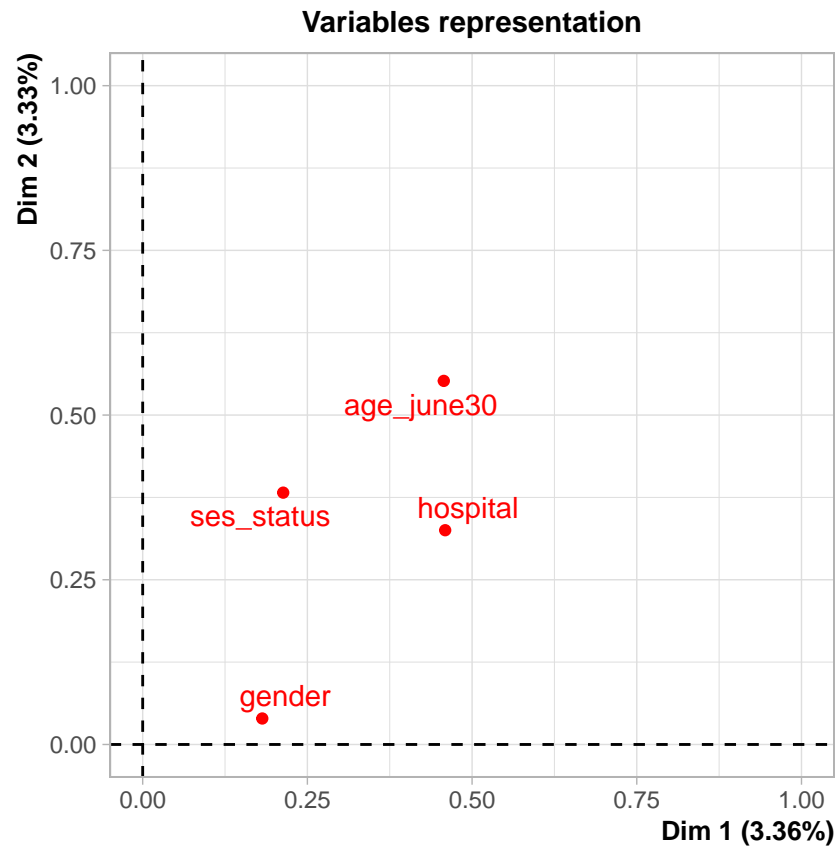
# Test if there is sufficient structure in the data to do clustering
#clusttendency <- get_clust_tendency(clinical_mca,
#                                     n = 5, # Needs to be same as number of features
#                                     gradient = list(low = "steelblue", high = "white"))
# Can only handle numeric data so not that useful but even with only the numeric features,
# the Hopkins stat is very close to 1 (0.997) so the data is highly clusterable

res_mca_clinical <- FactoMineR::MCA(clinical_mca, ncp = 4, graph = TRUE) # Can only take categorical va

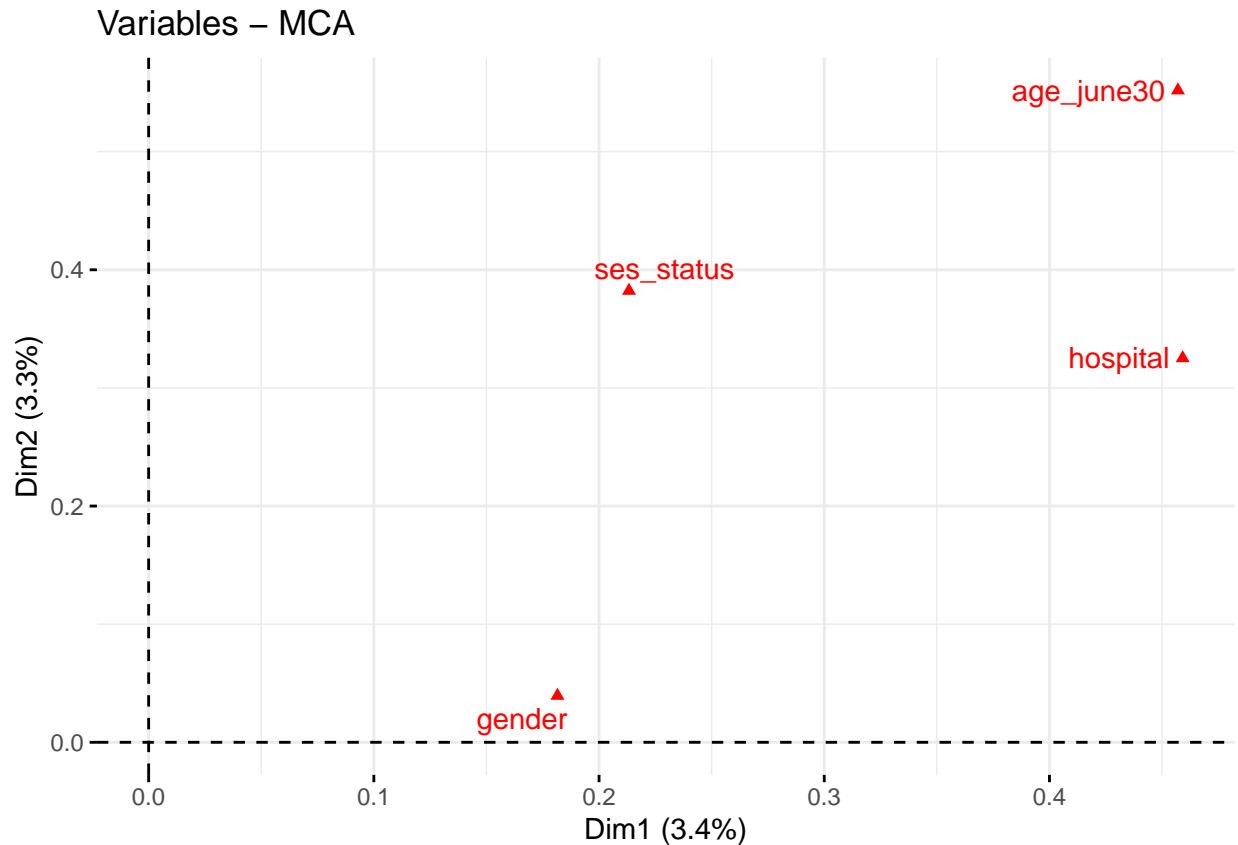
## Warning: ggrepel: 29 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

```





```
fviz_mca_var(res_mca_clinical, choice = "mca.cor",  
             repel = TRUE, # Avoid text overlapping (slow)  
             ggtheme = theme_minimal())
```



Probably better to do MCA on patient-level data because there will be a substructure of clusters of the patients' appointments. Especially as there are some duplicates in the clinical data, seemingly as a result of translating the medical_specialty vales

```
get.mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

patients_mca <- clinical %>%
  #filter(commune_code %in% araucsur_communes$commune_code) %>%
  mutate(
    medical_specialty_english = ifelse(medical_specialty == "Physiatry", "Psychiatry",
    ifelse(medical_specialty == "PEDIATRIA", "Paediatrics",
    ifelse(medical_specialty == "PSIQUIATRIA", "Psychiatry",
    ifelse(medical_specialty == "NEUROLOGIA", "Neurology", medical_specialty_english),
    medical_specialty_grouped = ifelse(medical_specialty_english %in% c("Psychiatry", "Child Psych",
    ifelse(medical_specialty_english %in% c("Neurology", "Pediatric Neu",
    ifelse(medical_specialty_english == "Paediatrics", "Paediatrics", "Paediatrics"),
    ses_status = ifelse(socio_economic_level %in% c("COLMENA GOLDEN CROSS", "RIO BLANCO", "CARABINI"),
  group_by(id, gender, dob, commune_name) %>%
  summarise(
    ses_status = get.min.na(ses_status), # Fortunately the available values are in alphabetical
    autism = get.max.na(autism),
    medical_specialty_grouped = get.mode(medical_specialty_grouped), # Most common medical spec
    hospital = hospital[which.max(apt_date)], # most recent hospital
    age = age_june30[which.max(apt_date)] # Age should be the same for all appts because it's b
  ) %>%
  ungroup() %>%
```

```

rename("sex_desc" = "gender") %>%
mutate(sex_desc = as.factor(sex_desc),
       age_group = as.factor(ifelse(age <= 2, "Age 0-2",
                                   ifelse(age >= 3 & age <= 5, "Age 3-5",
                                   ifelse(age >= 6 & age <= 8, "Age 6-8",
                                   ifelse(age >= 9 & age <= 11, "Age 9-11",
                                   ifelse(age >= 12 & age <= 14, "Age 12-14",
                                   ifelse(age >= 15 & age <= 18, "Age 15-18", "Adult"))))))) , # Shouldn't b
       age = as.factor(age),
       commune_name = as.factor(commune_name),
       ses_status = as.factor(ses_status),
       autism = as.factor(autism),
       medical_specialty_grouped = as.factor(medical_specialty_grouped),
       hospital = as.factor(hospital)) %>%
select(sex_desc,
       #age,
       age_group,
       commune_name,
       hospital,
       ses_status,
       autism,
       medical_specialty_grouped)

```

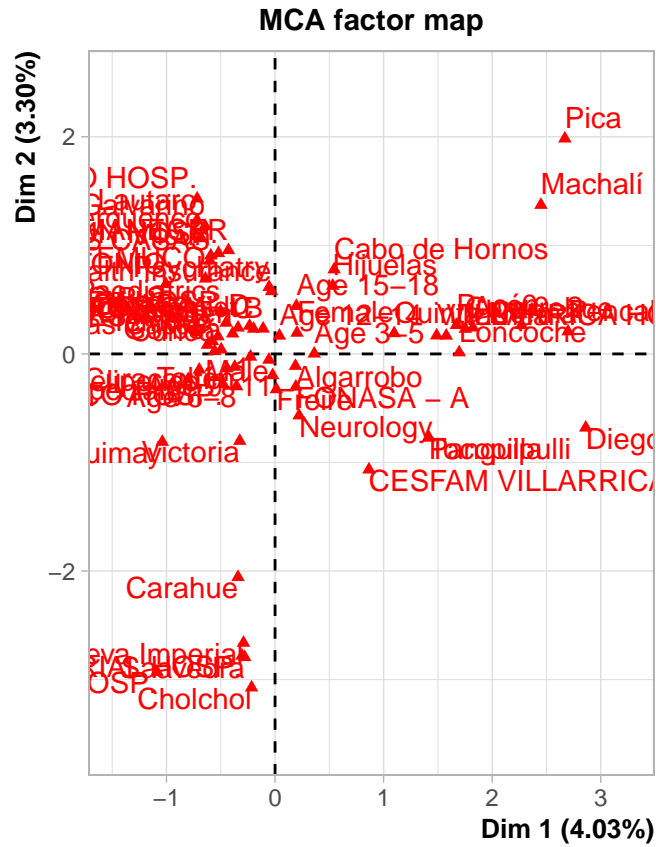
`summarise()` has grouped output by 'id', 'gender', 'dob'. You can override
using the `.groups` argument.

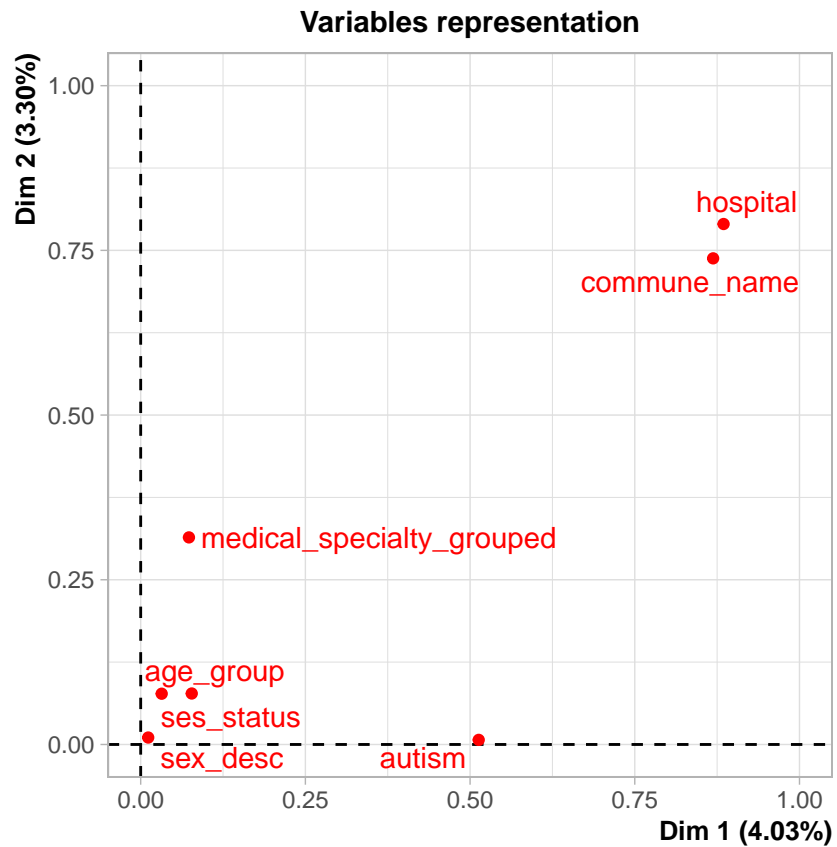
Note we still have more than 1 row for some patients because they have multiple communes, could go wi

```

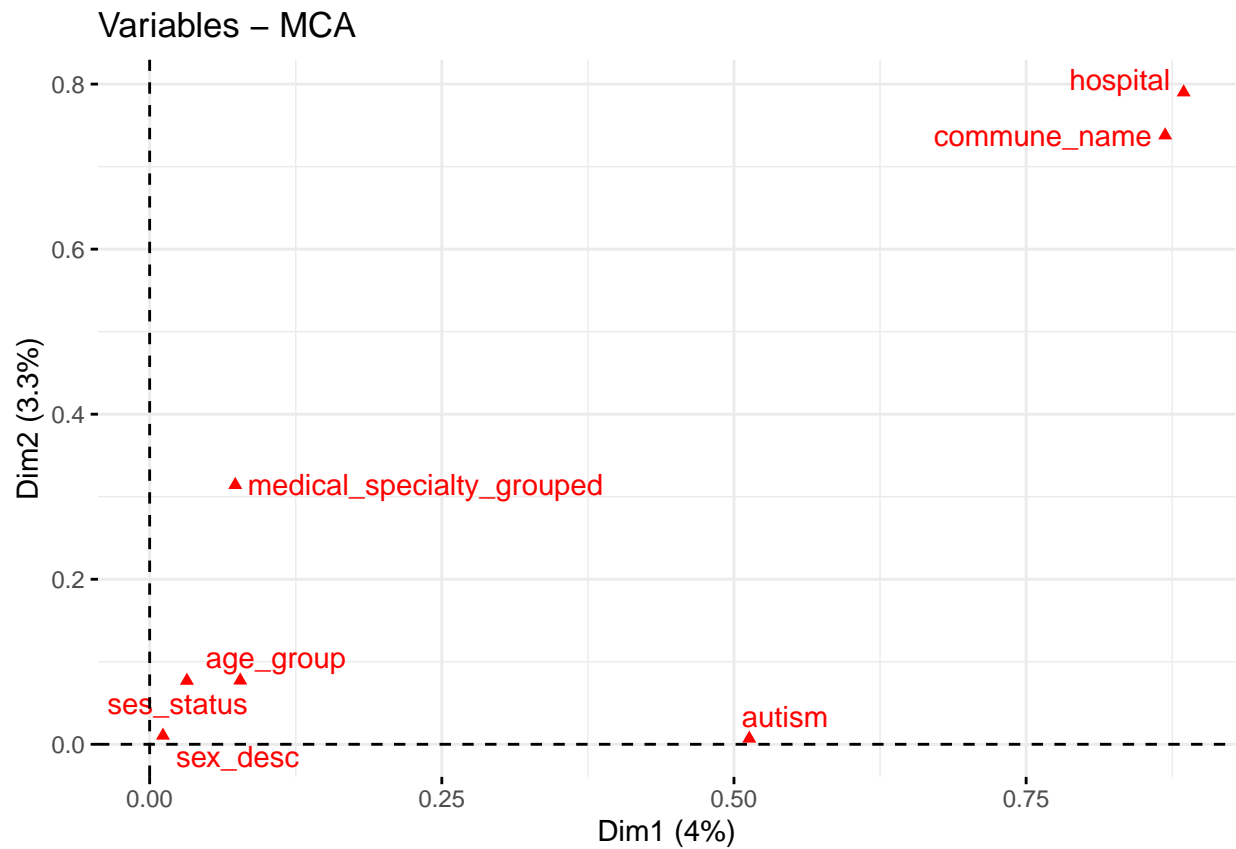
res_mca_patients <- FactoMineR::MCA(patients_mca, # Can only take categorical variables and they have t
                                   ncp = 7, # Needs to match number of features
                                   graph = TRUE)

```



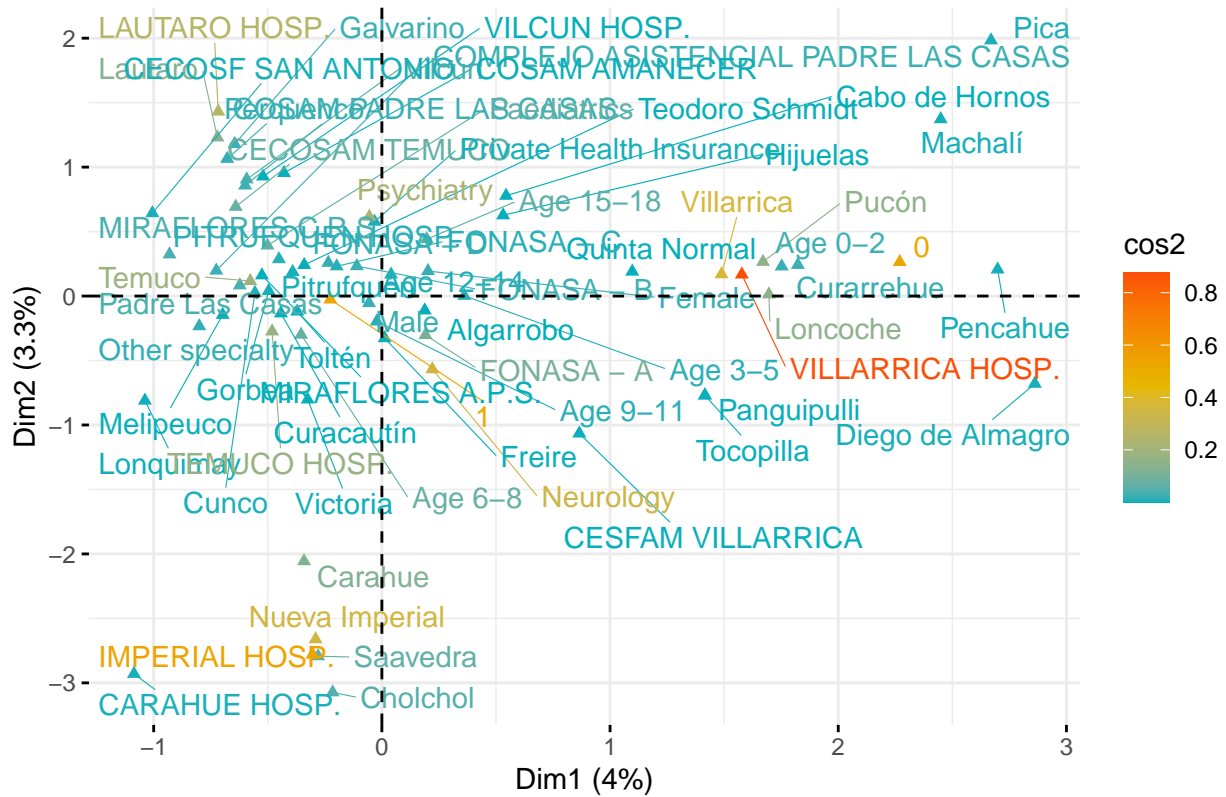


```
# Plot features
fviz_mca_var(res_mca_patients, choice = "mca.cor",
             repel = TRUE, # Avoid text overlapping
             ggtheme = theme_minimal())
```

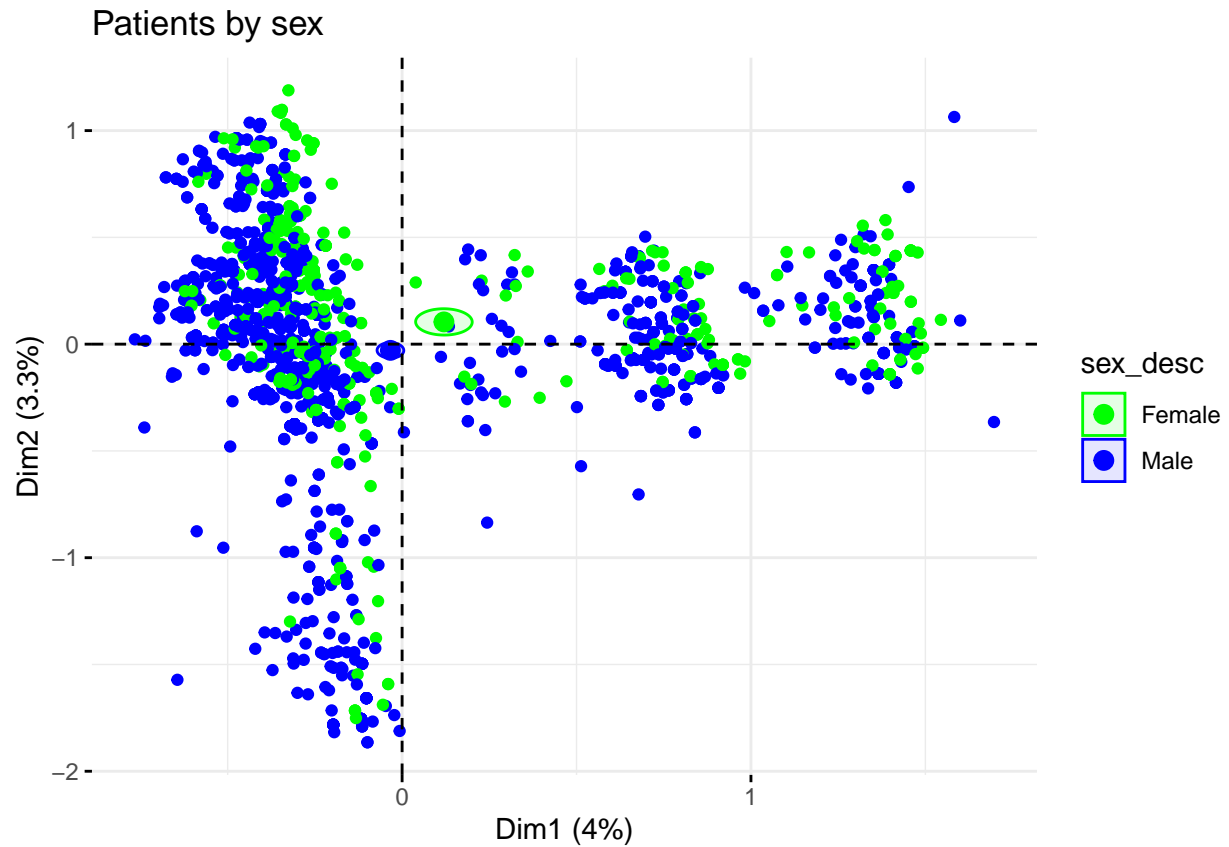


```
# Plot categories
fviz_mca_var(res_mca_patients,
  col.var = "cos2",
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  repel = TRUE, # Avoid text overlapping (slow if many points)
  ggtheme = theme_minimal())
```

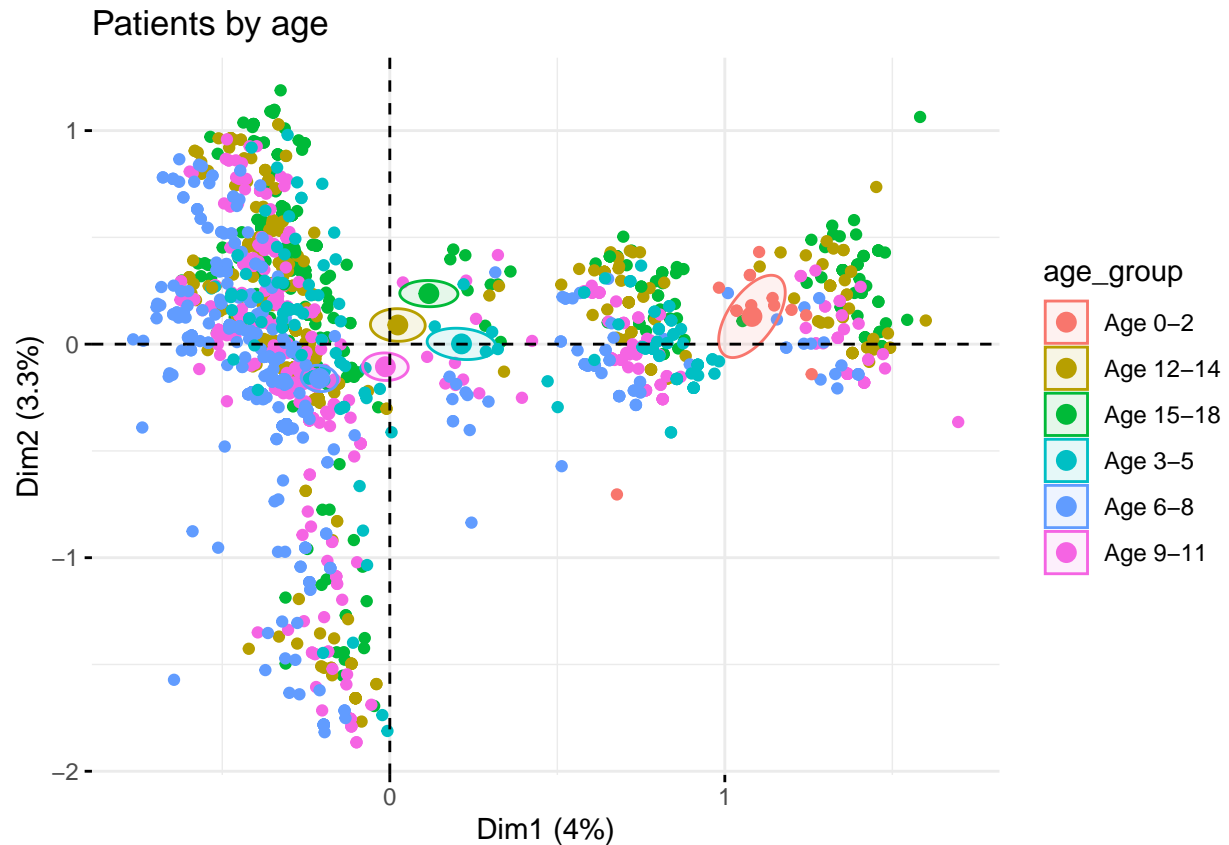
Variable categories – MCA



```
# Plot individual patients
fviz_mca_ind(res_mca_patients, col.ind = "cos2",
  gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
  repel = FALSE, # Graphs is not meaningful if this is TRUE
  ggtheme = theme_minimal())
```

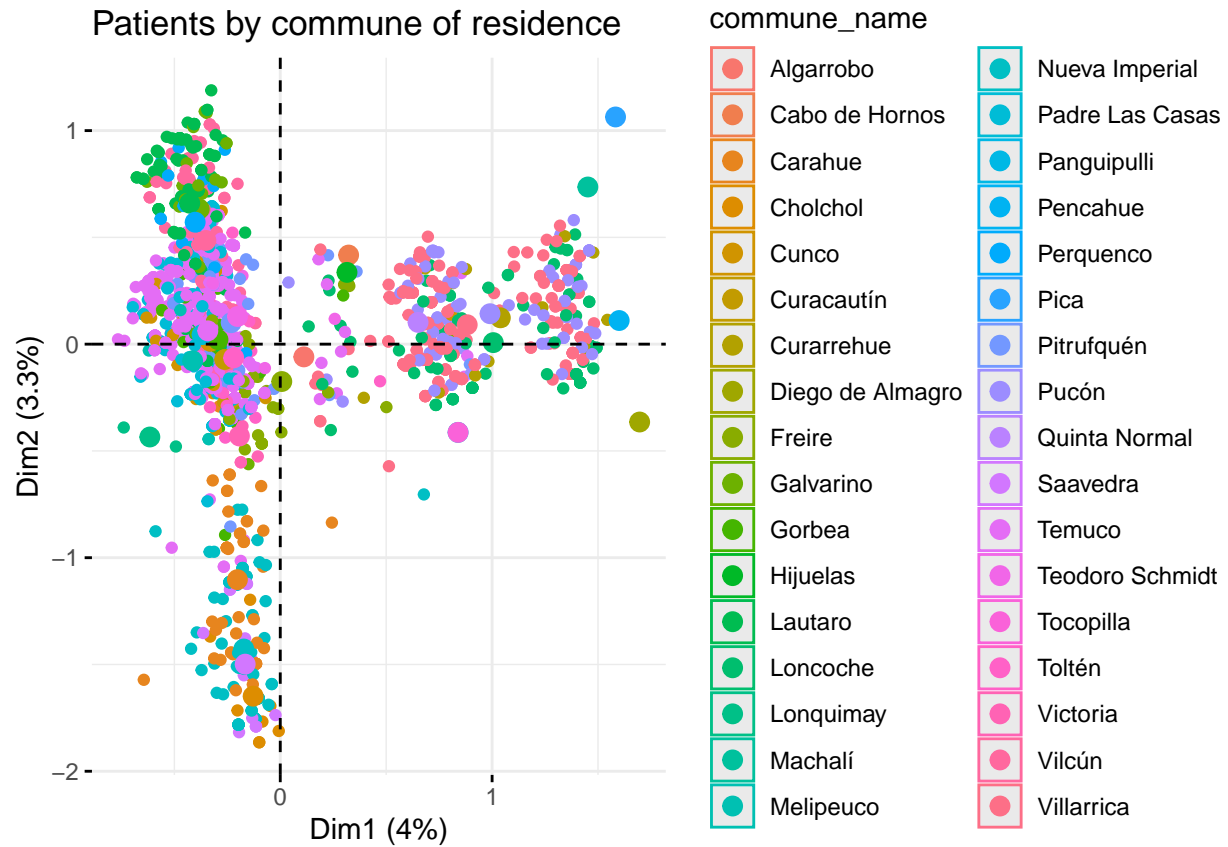



```
fviz_mca_ind(res_mca_patients,
  label = "none", # hide individual labels
  habillage = "age_group", # color by groups
  #palette = c("#00AFBB", "#E7B800"),
  addEllipses = TRUE, ellipse.type = "confidence",
  title = "Patients by age",
  ggtheme = theme_minimal())
```

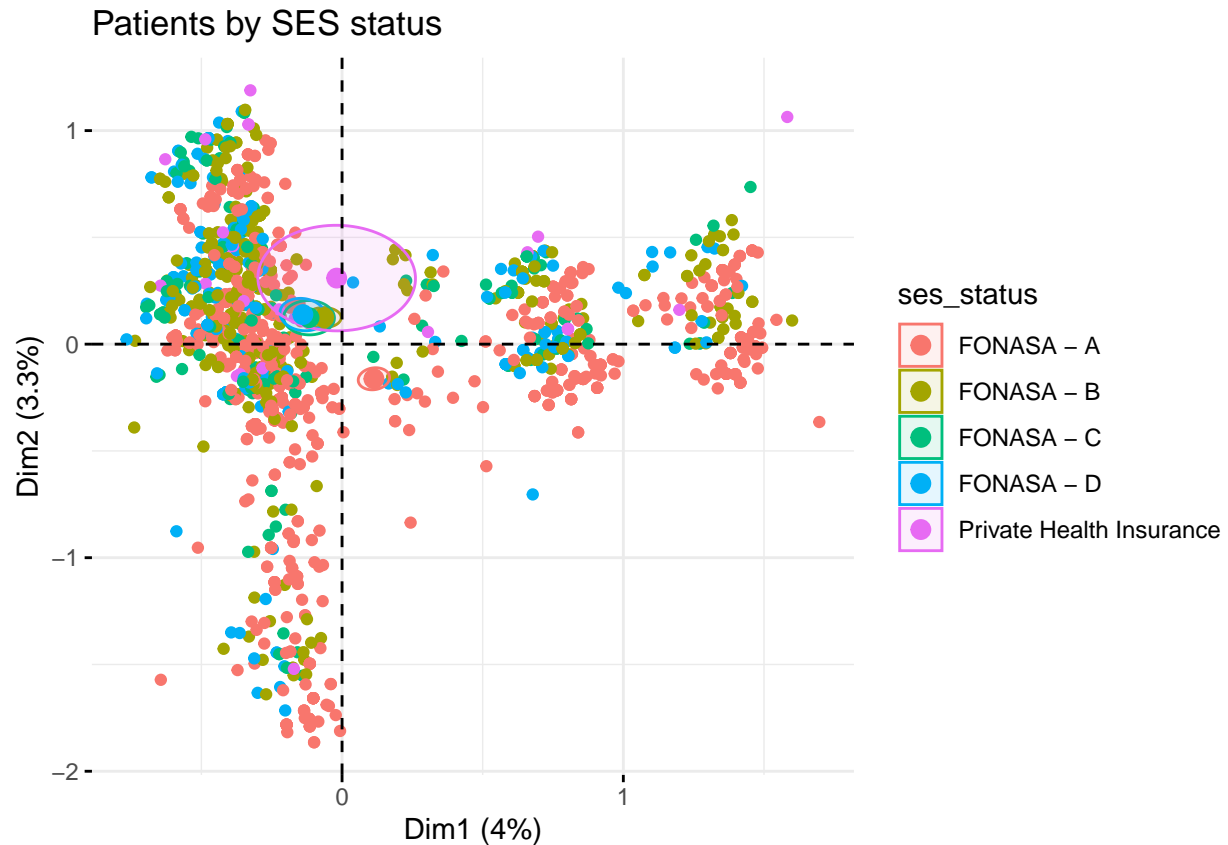


```
fviz_mca_ind(res_mca_patients,
  label = "none", # hide individual labels
  habillage = "commune_name", # color by groups
  #palette = c("#00AFBB", "#E7B800"),
  addEllipses = TRUE, ellipse.type = "confidence",
  title = "Patients by commune of residence",
  ggtheme = theme_minimal()) # Really not meaningful
```

```
## Warning: Computation failed in `stat_conf_ellipse()`
## Caused by error in `if (scale[1] > 0) ...`:
## ! missing value where TRUE/FALSE needed
```

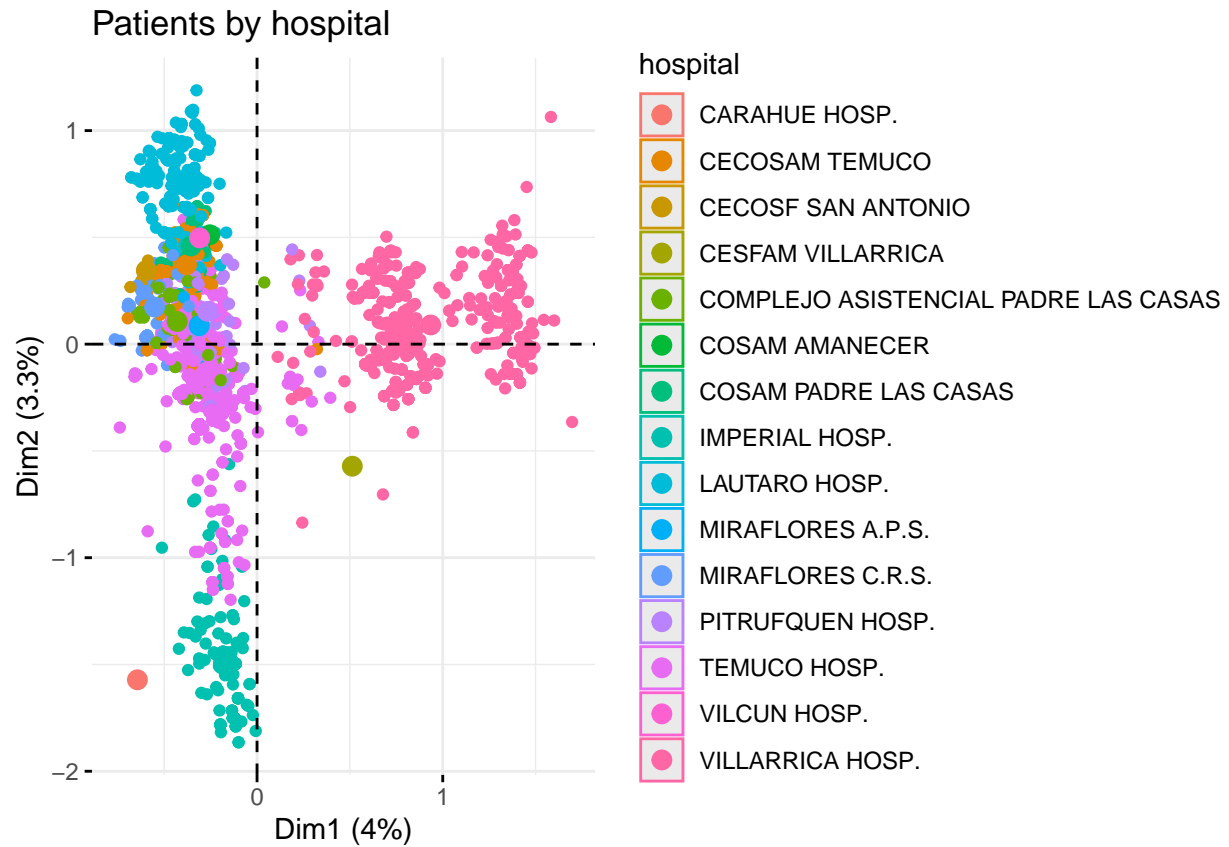


```
fviz_mca_ind(res_mca_patients,
  label = "none", # hide individual labels
  habillage = "ses_status", # color by groups
  #palette = c("#00AFBB", "#E7B800"),
  addEllipses = TRUE, ellipse.type = "confidence",
  title = "Patients by SES status",
  ggtheme = theme_minimal())
```

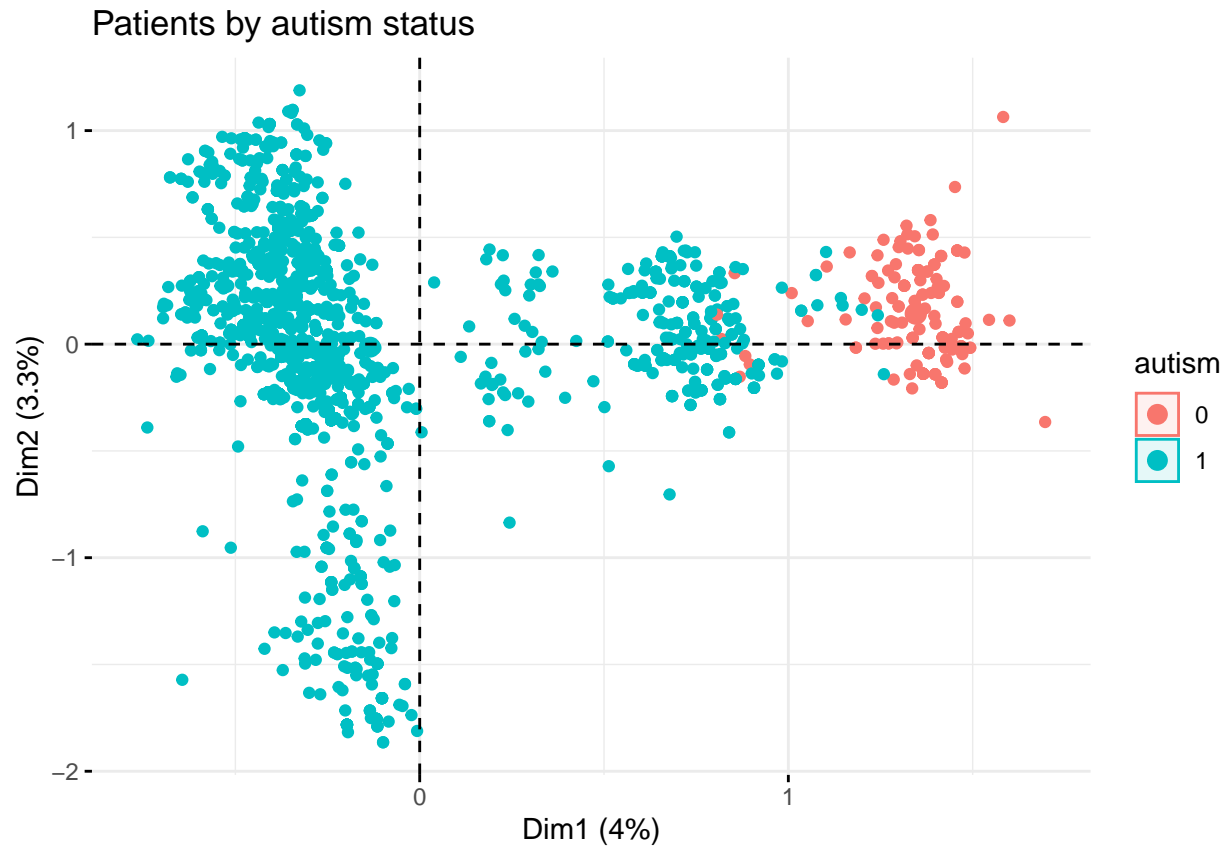


```
fviz_mca_ind(res_mca_patients,
  label = "none", # hide individual labels
  habillage = "hospital", # color by groups
  #palette = c("#00AFBB", "#E7B800"),
  addEllipses = TRUE, ellipse.type = "confidence",
  title = "Patients by hospital",
  ggtheme = theme_minimal())
```

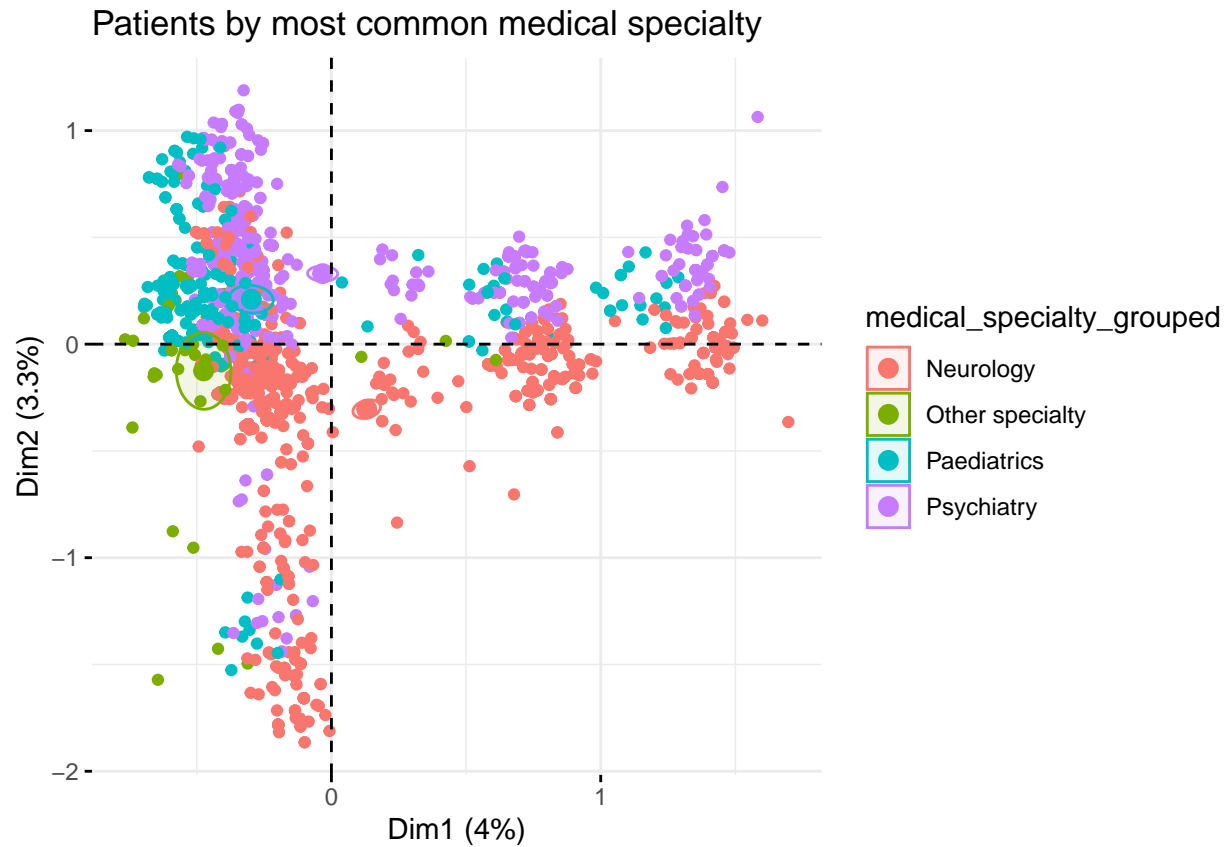
```
## Warning: Computation failed in `stat_conf_ellipse()`
## Caused by error in `if (scale[1] > 0) ...`:
## ! missing value where TRUE/FALSE needed
```



```
fviz_mca_ind(res_mca_patients,
  label = "none", # hide individual labels
  habillage = "autism", # color by groups
  #palette = c("#00AFBB", "#E7B800"),
  addEllipses = TRUE, ellipse.type = "confidence",
  title = "Patients by autism status",
  ggtheme = theme_minimal())
```



```
fviz_mca_ind(res_mca_patients,  
  label = "none", # hide individual labels  
  habillage = "medical_specialty_grouped", # color by groups  
  #palette = c("#00AFBB", "#E7B800"),  
  addEllipses = TRUE, ellipse.type = "confidence",  
  title = "Patients by most common medical specialty",  
  ggtheme = theme_minimal())
```



```
#fviz_ellipses(res_mca_patients, 1:4, geom = "point") # Ugly and too many warnings
#fviz_ellipses(res_mca_patients, c("sex_desc", "age_group", "ses_status"), geom = "point")
#fviz_ellipses(res_mca_patients, 1, geom = "point")

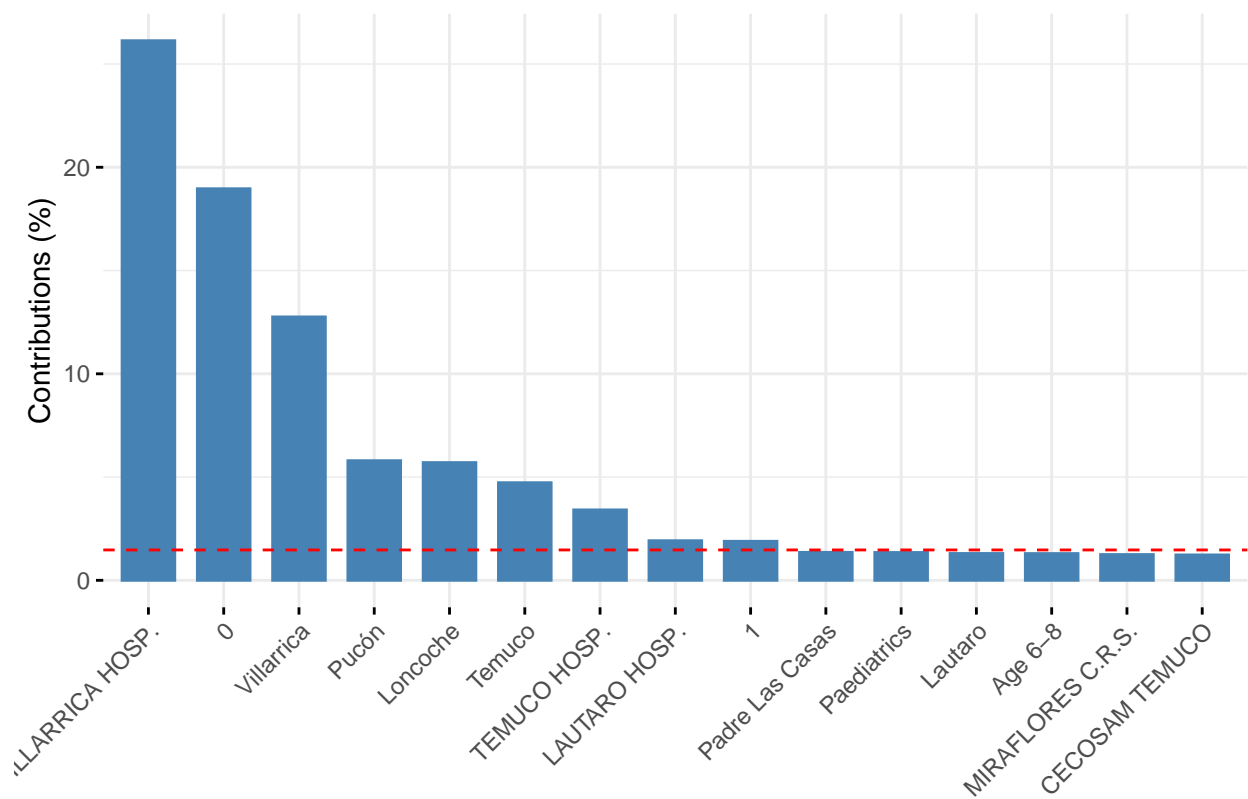
fviz_cos2(res_mca_patients, choice = "var", axes = 1:2)
```


## Machalí	0.14	0.05	0.00	0.01	0.02	0.51	0.01
## Melipeuco	0.06	0.00	0.02	0.03	0.99	0.15	0.61
## Nueva Imperial	0.16	16.54	3.09	0.18	0.59	4.51	0.12
## Padre Las Casas	1.35	0.03	2.22	0.22	1.89	1.79	24.48
## Panguipulli	0.05	0.02	0.00	0.01	0.07	0.16	0.00
## Penciahue	0.17	0.00	0.00	0.00	0.00	0.00	0.02
## Perquenco	0.21	0.62	3.96	0.01	0.01	0.36	0.05
## Pica	0.17	0.11	0.01	0.02	0.08	0.14	0.04
## Pitrufuquén	0.17	0.05	0.32	20.37	2.95	0.00	0.01
## Pucón	5.79	0.18	0.01	0.02	0.03	0.11	0.00
## Quinta Normal	0.03	0.00	0.01	0.00	0.02	0.02	0.04
## Saavedra	0.03	3.15	0.57	0.01	0.15	0.32	0.00
## Temuco	4.73	0.23	8.78	3.31	0.00	0.52	8.68
## Teodoro Schmidt	0.03	0.02	0.04	8.98	1.51	0.26	0.11
## Tocopilla	0.05	0.02	0.00	0.01	0.07	0.16	0.00
## Toltén	0.06	0.01	0.01	2.76	0.22	0.61	0.14
## Victoria	0.01	0.09	0.03	0.02	0.00	2.50	0.30
## Vilcún	0.50	1.42	6.09	0.10	0.01	1.99	0.30
## Villarrica	12.75	0.20	0.00	0.00	0.17	0.00	0.10
## CARAHUE HOSP.	0.03	0.25	0.00	0.08	0.00	0.00	0.51
## CECOSAM TEMUCO	1.23	1.76	3.58	3.35	4.58	3.15	12.22
## CECOSF SAN ANTONIO	0.14	0.07	0.23	0.00	0.28	0.24	1.41
## CESFAM VILLARRICA	0.02	0.03	0.01	0.00	0.02	0.18	0.00
## COMPLEJO ASISTENCIAL PADRE LAS CASAS	0.88	0.08	0.45	0.27	10.12	0.79	0.04
## COSAM AMANECER	0.01	0.08	0.12	0.12	0.36	0.27	0.14
## COSAM PADRE LAS CASAS	0.32	0.81	1.23	0.53	0.12	9.04	28.93
## IMPERIAL HOSP.	0.24	24.73	5.29	0.15	1.14	10.49	0.31
## LAUTARO HOSP.	1.92	9.33	31.05	0.22	0.11	0.43	0.03
## MIRAFLORES A.P.S.	0.03	0.00	0.01	0.25	1.59	0.01	1.07
## MIRAFLORES C.R.S.	1.25	0.18	0.45	0.55	17.68	4.07	1.64
## PITRUFQUEN HOSP.	0.34	0.17	0.43	39.66	5.40	0.11	0.01
## TEMUCO HOSP.	3.41	1.35	3.54	1.43	0.27	14.12	1.73
## VILCUN HOSP.	0.01	0.02	0.17	0.00	0.04	0.05	0.01
## VILLARRICA HOSP.	26.13	0.36	0.00	0.01	0.33	0.01	0.13
## FONASA - A	0.67	2.05	1.14	0.03	0.15	0.03	0.05
## FONASA - B	0.14	0.75	0.75	0.00	0.07	1.32	0.42
## FONASA - C	0.18	0.30	0.23	0.00	0.03	0.65	0.69
## FONASA - D	0.31	0.46	0.24	0.06	0.25	0.50	1.22
## Private Health Insurance	0.00	0.27	0.45	0.00	0.00	0.12	0.26
## 0	18.96	0.31	0.00	0.08	0.00	0.39	0.00
## 1	1.89	0.03	0.00	0.01	0.00	0.04	0.00
## Neurology	0.95	7.65	0.10	0.01	0.05	9.81	1.42
## Other specialty	0.63	0.07	1.36	1.38	0.01	0.00	2.96
## Paediatrics	1.35	1.01	0.38	4.32	24.39	4.51	2.53
## Psychiatry	0.04	6.89	0.18	0.68	10.22	5.28	0.00

Contributions of variable categories to dimension 1

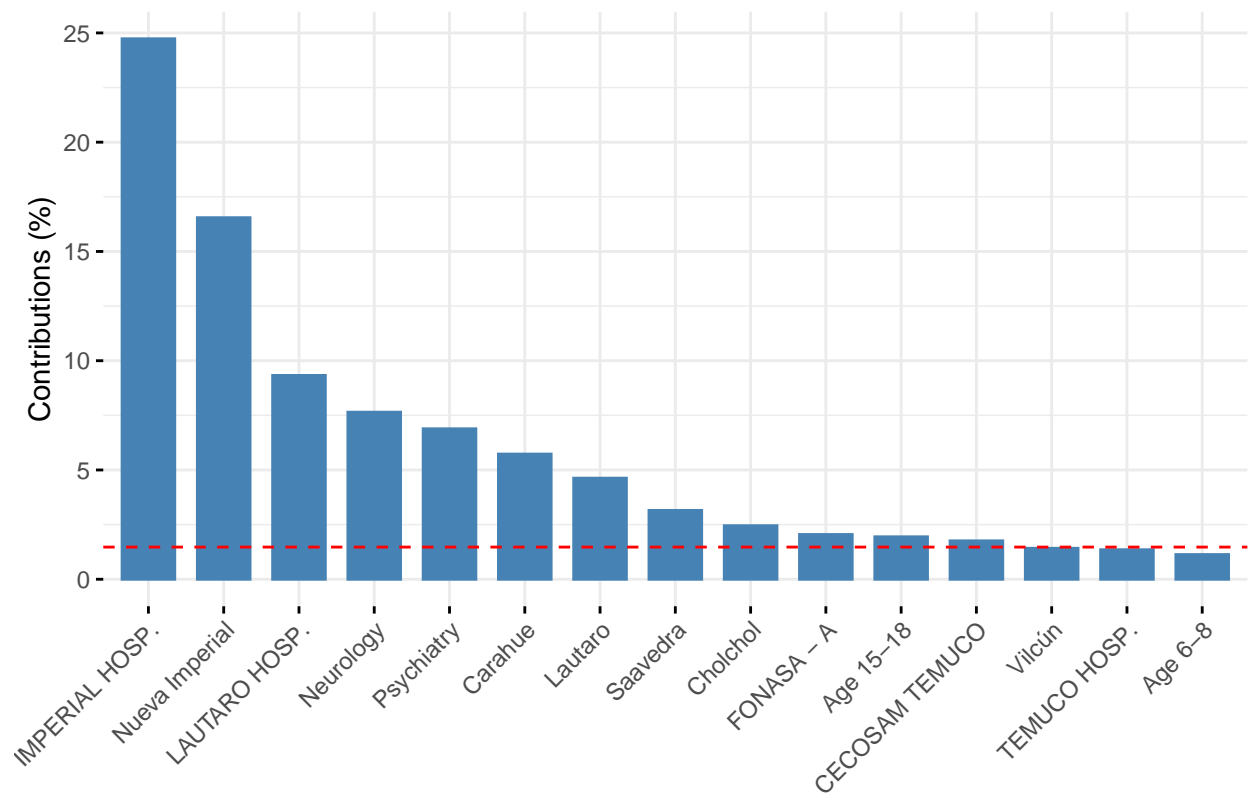
```
fviz_contrib(res_mca_patients, choice = "var", axes = 1, top = 15)
```

Contribution of variables to Dim-1

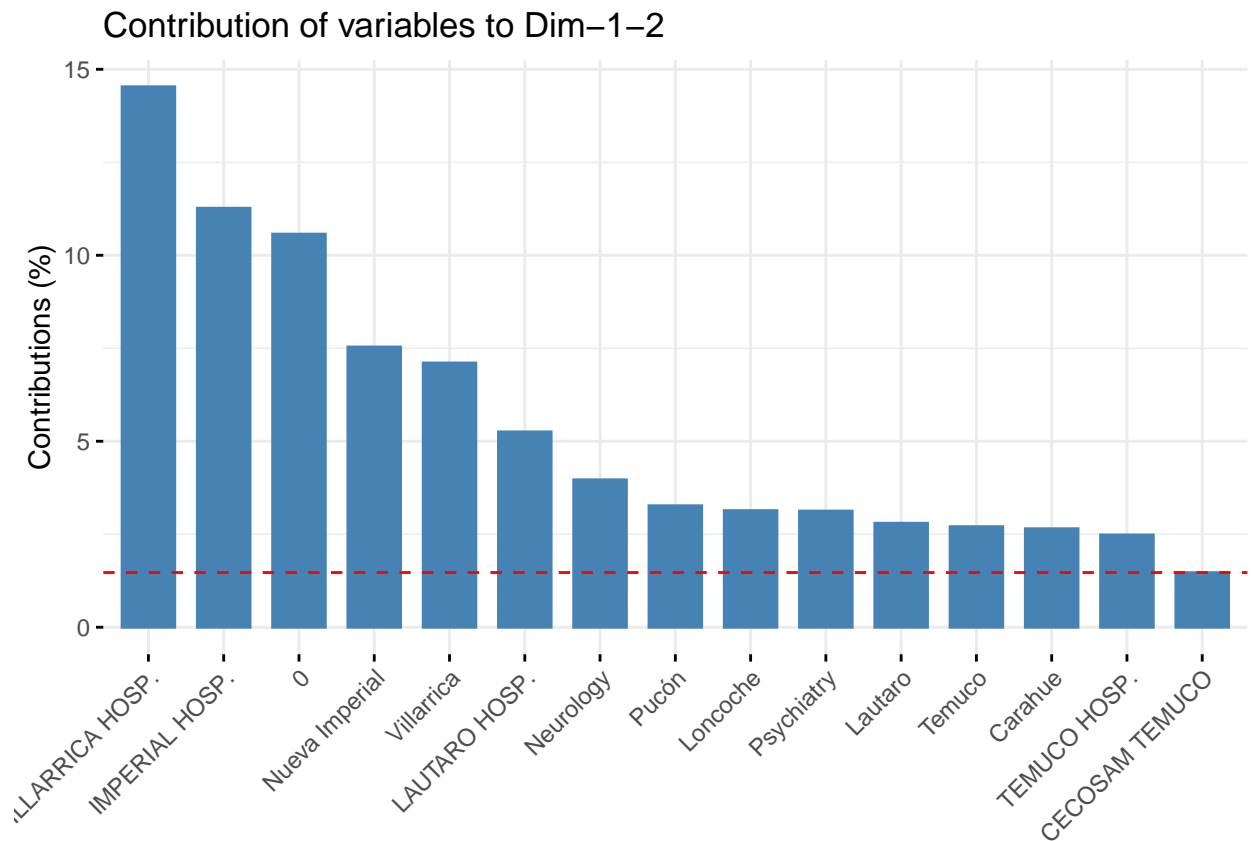


```
# Contributions of variable categories to dimension 2
fviz_contrib(res_mca_patients, choice = "var", axes = 2, top = 15)
```

Contribution of variables to Dim-2



```
# Total contribution to dimension 1 and 2
fviz_contrib(res_mca_patients, choice = "var", axes = 1:2, top = 15)
```



"The red dashed line on the graph above indicates the expected average value, If the contributions were equal for all variables, the expected average value would be approximately 1.5%." *(Note: This text is a comment and not part of the original image content.)*

CFA for clinical data

Dumping ground, don't use below here.

Record linkage using machine learning

Try linkage using ML, as done by Jan van der Laan here https://cran.r-project.org/web/packages/reclin2/vignettes/record_linkage_using_machine_learning.html

In reclin2 package, use `?identical()` to see available matching algorithms.

The Jaro-Winkler distance is a string metric for measuring the edit distance between two sequences. It is a variant of the Jaro distance metric proposed by William E. Winkler in 1990 ¹. The Jaro-Winkler distance uses a prefix scale which gives more favorable ratings to strings that match from the beginning for a set prefix length. The higher the Jaro-Winkler distance for two strings is, the less similar the strings are. The score is normalized such that 0 means an exact match and 1 means there is no similarity ¹.

Need to explore different comparator algorithms. Currently it's exact match. Would be good to do communes that are neighbours and ages off by 1.

Try bayesian linkage?

Follow Thomas Stringham <https://arxiv.org/pdf/2003.04238.pdf> who followed Sadinle <https://arxiv.org/abs/1601.06630> Not doing this as limited value when not matching strings.