Assignment 1: answers

Data manipulation and maps

Sébastien Rochette, ThinkR

Table des matières

0.1. Warm-up

0.1.1 Make sure you've installed {dplyr} >= 0.7 and {prenoms} package

0.1.2 Load here {dplyr}, {prenoms} and any other needed package

```
library(dplyr)
library(tidyr)
library(prenoms)
library(readr)
library(readxl)
library(ggplot2)
library(sf)
```

0.1.3 Import

prenomsdataset

Using `data(prenoms)` load `prenoms` dataset from `{prenoms}` package.

```
data(prenoms)
```

What kind of object is `prenoms` ?

```
class(prenoms)
```

```
## [1] "tbl_df"     "tbl"        "data.frame"
```

Explore the database using the '5-functions-to-always-run-on-a-database'

```
dim(prenoms)
names(prenoms)
head(prenoms)
View(prenoms)
summary(prenoms)
```

Using `glimpse`, have a look at `prenoms`'s structure.

```
glimpse(prenoms)
```

```
## Observations: 3,712,763
## Variables: 6
## $ year <int> 1900, 1900, 1900, 1900, 1900, 1900, 1900, 1900, 1900, 190...
## $ sex  <chr> "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", "F", "F...
## $ name <chr> "Adèle", "Adrienne", "Aimée", "Alice", "Alphonsine", "Amé...
## $ n    <int> 5, 9, 7, 24, 5, 8, 5, 6, 21, 12, 21, 6, 5, 5, 17, 3, 5, 1...
## $ dpt  <chr> "01", "01", "01", "01", "01", "01", "01", "01", "01", "01...
## $ prop <dbl> 0.0015777848, 0.0028400126, 0.0022088987, 0.0075733670, 0...
```

Regions, departements and surfaces

Load the "dpt_data_modif.csv" dataset from IGN (French public state administrative establishment founded in 1940[1] to produce and maintain geographical information for France and its overseas departments and territories) using the appropriate function. Data have been prepared for you: the surface of departement has been calculated and spatial data removed.

```
dpt_data_modif <- read_csv("data/dpt_data_modif.csv")
```

```
## Parsed with column specification:
## cols(
##   CODE_DEPT = col_character(),
##   NOM_DEPT = col_character(),
##   CODE_CHF = col_character(),
##   NOM_CHF = col_character(),
##   CODE_REG = col_integer(),
##   NOM_REG = col_character(),
##   surface_m = col_double()
## )
```

Elementary and college schools

We also fetched for you on data.gouv.fr the addresses of "primary and secondary schools, the administrative structures of the Ministry of National Education. Public and private sectors."

1. Data preprocessing
   - Import the csv file : "DEPP-etab-1D2D.csv" and name it "depp_orig"
     - Encoding is `"latin1"`
   - Transform zip code ("code_postal_uai") into 5 characters with zeros
   - Extract department numbers ("dpt") starting from column "code_postal_uai"
   - Save the modifications into "depp_modif.csv"

```
depp_orig <- read.csv2("data/DEPP-etab-1D2D.csv",
                       encoding = "latin1")

depp_modif <- depp_orig %>%
  mutate(code_postal_uai =
           formatC(code_postal_uai, width = 5, flag = "0")) %>%
  mutate(dpt = substr(code_postal_uai, 1,2))

write_csv(depp_modif, "data/depp_modif.csv")
```

2. Read the pre-processed "depp_modif.csv" file

```
depp_modif <- read_csv("data/depp_modif.csv")
```

```
## Parsed with column specification:
## cols(
##   numero_uai = col_character(),
##   appellation_officielle = col_character(),
##   denomination_principale = col_character(),
##   patronyme_uai = col_character(),
##   secteur_public_prive_libe = col_character(),
##   adresse_uai = col_character(),
##   lieu_dit_uai = col_character(),
##   boite_postale_uai = col_integer(),
##   code_postal_uai = col_character(),
##   localite_acheminement_uai = col_character(),
##   coordonnee_x = col_double(),
##   coordonnee_y = col_double(),
##   appariement = col_character(),
##   localisation = col_character(),
##   nature_uai = col_integer(),
##   nature_uai_libe = col_character(),
##   etat_etablissement = col_integer(),
##   dpt = col_character()
## )
```

Facts observed by the police services and national gendarmerie units by department

We also gathered data from data.gouv.fr concerning "all the facts observed by the police services and national gendarmerie units by department from 1996 to 2011"

1. Data preprocessing
   - Import Excel sheet "2010" from "faitsconstatespardepartementde2002-a-2011.xls" file
     - *beware of the original formatting*
   - Copy it into "faits_2010_modif" in order to make some modifications:
     - Delete Excel calculations:
       - `Tout_département`, `Tout_index`
     - Transform in long format using `gather`
       - 4 columns : Index, Libellé, dpt, nombre
     - save the dataframe into a csv file "faits_2010_modif.csv"

```
faits_2010_orig <- read_excel(
  "data/faitsconstatespardepartementde2002-a-2011.xls",
  sheet = "2010", skip = 2
)

# Supprimer les données issues de calculs dans Excel
faits_2010_modif <- faits_2010_orig[-1,-3] %>%
  gather(dpt, nombre, -Index, -Libellé)

write_csv(faits_2010_modif, "data/faits_2010_modif.csv")
```

2. Read preprocessed file "faits_2010_modif.csv"

```
faits_2010_modif <- read_csv("data/faits_2010_modif.csv")
```

```
## Parsed with column specification:
## cols(
##   Index = col_integer(),
##   Libellé = col_character(),
##   dpt = col_character(),
##   nombre = col_integer()
## )
```

0.2. Analyses

Some assumptions to do the exercise:

- every child born in a department stays into that department until the end of college
- every children between 11 and 14 years old is in a college
- the number of college is constant between 2010 and 2016
- College "à ouvrir" (i.e. "to be open") do not have children. Others have.

0.2.1 Filter datasets to Metropolitan France

Datasets to be filtered: `prenoms`, `depp_modif`, `faits_2010_modif`, `dpt_data_modif`

- Department named "2A" and "2B" should be merged to "20"
- We only work with data in Metropolitan France, which means for "dpt" between `01` and `95` included. Others needs to be filtered.

```
prenoms_metro <- prenoms %>%
  mutate(dpt = if_else(dpt %in% c("2A", "2B"), "20", dpt)) %>%
  filter(dpt %in% formatC(1:95, width = 2, flag = "0"))
```

```
depp_metro <- depp_modif %>%
  mutate(dpt = if_else(dpt %in% c("2A", "2B"), "20", dpt)) %>%
  filter(dpt %in% formatC(1:95, width = 2, flag = "0"))

faits_2010_metro <- faits_2010_modif %>%
  mutate(dpt = if_else(dpt %in% c("2A", "2B"), "20", dpt)) %>%
  filter(dpt %in% formatC(1:95, width = 2, flag = "0"))

dpt_data_metro <- dpt_data_modif %>%
    mutate(CODE_DEPT = if_else(CODE_DEPT %in% c("2A", "2B"), "20", CODE_DEPT)) %>%
  filter(CODE_DEPT %in% formatC(1:95, width = 2, flag = "0"))

dpt_data_metro
```

```
## # A tibble: 96 x 7
##    CODE_DEPT NOM_DEPT   CODE_CHF NOM_CHF   CODE_REG NOM_REG      surface_m
##    <chr>     <chr>      <chr>    <chr>        <int> <chr>            <dbl>
##  1 39        JURA       300      LONS-LE-~       27 BOURGOGNE-F~  5.04e9
##  2 42        LOIRE      218      SAINT-ET~       84 AUVERGNE-RH~  4.80e9
##  3 76        SEINE-MAR~ 540      ROUEN           28 NORMANDIE     6.33e9
##  4 89        YONNE      024      AUXERRE         27 BOURGOGNE-F~  7.45e9
##  5 68        HAUT-RHIN  066      COLMAR          44 ALSACE-CHAM~  3.53e9
##  6 28        EURE-ET-L~ 085      CHARTRES        24 CENTRE-VAL ~  5.93e9
##  7 10        AUBE       387      TROYES          44 ALSACE-CHAM~  6.02e9
##  8 55        MEUSE      029      BAR-LE-D~       44 ALSACE-CHAM~  6.23e9
##  9 61        ORNE       001      ALENCON         28 NORMANDIE     6.14e9
## 10 67        BAS-RHIN   482      STRASBOU~       44 ALSACE-CHAM~  4.80e9
## # ... with 86 more rows
```

0.2.2 National average number of children per college in 2010 ?

```
# Enfants ayant 11 à 14 ans en 2010
nb_enfants <- prenoms_metro %>%
  filter(year >= 1996 & year <= 1999) %>%
  summarise(total = sum(n))

# Nombre de collèges en France en 2010 (=2016)
nb_colleges <- depp_metro %>%
  filter(nature_uai_libe == "Collège" &
           etat_etablissement) %>%
  nrow()

# Nombre d'enfants par collège au niveau national en 2010
nb_enfants/nb_colleges
```

```
##      total
## 1 348.8871
```

0.2.3 Average number of children per college in 2010 in each department?

- Arrange departments according to the calculated average in descending order

```
# Enfants ayant 11 à 14 ans en 2016 par dpt
nb_enfants <- prenoms_metro %>%
  filter(year >= 1996 & year <= 1999) %>%
  group_by(dpt) %>%
  summarise(enf = sum(n))
```

```
# Nombre de collèges en France en 2016
nb_colleges <- depp_metro %>%
  filter(nature_uai_libe == "Collège") %>%
  group_by(dpt) %>%
  summarise(coll = n())

# Jointure
enf_coll <- inner_join(nb_enfants, nb_colleges, by = "dpt") %>%
  mutate(ratio = enf/coll) %>%
  arrange(desc(ratio))

enf_coll
```

```
## # A tibble: 95 x 4
##    dpt      enf  coll ratio
##    <chr>  <int> <int> <dbl>
##  1 75    138387   190  728.
##  2 92     84505   139  608.
##  3 69     87948   169  520.
##  4 59    138285   288  480.
##  5 76     59816   133  450.
##  6 84     24903    58  429.
##  7 42     32065    77  416.
##  8 51     25268    61  414.
##  9 74     29315    71  413.
## 10 68     29188    71  411.
## # ... with 85 more rows
```

0.2.4 Number of Facts observed by the police services in 2010 per department ?

```
# Nb faits par dpt en 2010
faits_2010_metro %>%
  group_by(dpt) %>%
  summarise(faits = sum(nombre))
```

```
## # A tibble: 95 x 2
##    dpt   faits
##    <chr> <int>
##  1 01    22615
##  2 02    24052
##  3 03    11653
##  4 04     7035
##  5 05     5303
##  6 06    93308
##  7 07    11011
##  8 08    10795
##  9 09     4829
## 10 10    14360
## # ... with 85 more rows
```

0.2.5 Number of children born, number of colleges and facts related by the police services per department in 2010 ?

- Group all information in the same table
- Arrange by descending order of children, schools and facts

```
# Enfants nés en 2010 par dpt
nb_enfants <- prenoms_metro %>%
```

```
  filter(year == 2010) %>%
  group_by(dpt) %>%
  summarise(nb_enfants = sum(n))

# Nombre de collèges en France en 2016
nb_colleges <- depp_metro %>%
  filter(nature_uai_libe == "Collège") %>%
  group_by(dpt) %>%
  summarise(nb_colleges = n())

# Nb faits par dpt en 2010
nb_faits <- faits_2010_metro %>%
  group_by(dpt) %>%
  summarise(nb_faits = sum(nombre))

# Jointure par dpt
all_by_dpt <- nb_enfants %>%
  inner_join(nb_colleges, by = "dpt") %>%
  inner_join(nb_faits, by = "dpt") %>%
  arrange(desc(nb_enfants), desc(nb_colleges), desc(nb_faits))

all_by_dpt
```

```
## # A tibble: 95 x 4
##    dpt   nb_enfants nb_colleges nb_faits
##    <chr>      <int>       <int>    <int>
##  1 75         35795         190   238856
##  2 59         33771         288   166565
##  3 69         23073         169   115632
##  4 92         22114         139    96520
##  5 13         21893         194   172445
##  6 93         16082         158   142798
##  7 62         14983         160    73918
##  8 76         14878         133    66765
##  9 44         14539         147    71222
## 10 94         14355         138    90376
## # ... with 85 more rows
```

0.2.6 Number of children born, number of colleges and facts related by the police services per km² in 2010 by department?

```
stats_km2 <- dpt_data_modif %>%
  mutate(surface_km = surface_m / 1e6) %>%
  inner_join(all_by_dpt, by = c("CODE_DEPT" = "dpt")) %>%
  mutate_if(is.integer, as.numeric) %>% # facultatif
  mutate_at(vars(starts_with("nb_")), funs(bykm = ./surface_km))

stats_km2 %>%
  select("CODE_DEPT", "NOM_DEPT", ends_with("bykm"), everything())
```

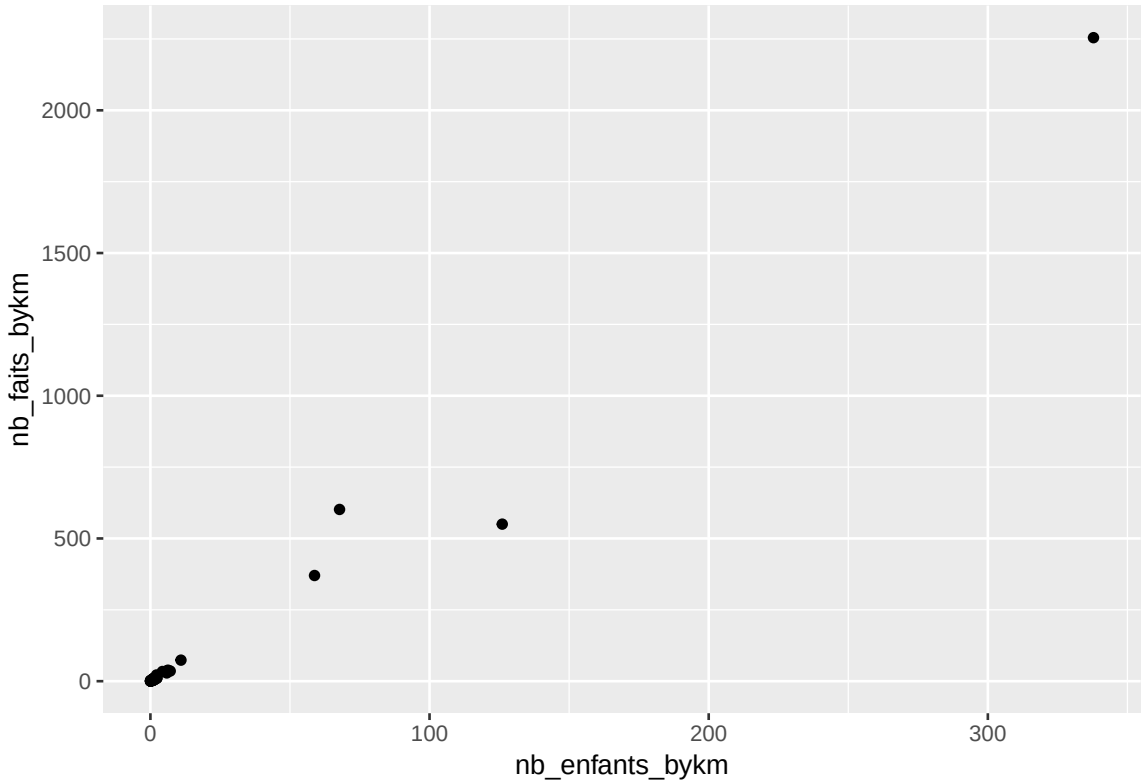
```
## # A tibble: 94 x 14
##    CODE_DEPT NOM_DEPT nb_enfants_bykm nb_colleges_bykm nb_faits_bykm
##    <chr>     <chr>              <dbl>            <dbl>         <dbl>
##  1 39        JURA               0.302          0.00734          1.65
##  2 42        LOIRE              1.63           0.0161           7.24
##  3 76        SEINE-M~           2.35           0.0210          10.6
##  4 89        YONNE              0.287          0.00470          2.14
##  5 68        HAUT-RH~           1.87           0.0201           9.14
```

```
##  6 28        EURE-ET~          0.562          0.00844          2.97
##  7 10        AUBE              0.362          0.00565          2.38
##  8 55        MEUSE             0.148          0.00465          1.09
##  9 61        ORNE              0.353          0.00733          1.64
## 10 67        BAS-RHIN          2.18           0.0231          10.3
## # ... with 84 more rows, and 9 more variables: CODE_CHF <chr>,
## #   NOM_CHF <chr>, CODE_REG <dbl>, NOM_REG <chr>, surface_m <dbl>,
## #   surface_km <dbl>, nb_enfants <dbl>, nb_colleges <dbl>, nb_faits <dbl>
```

0.2.7 Is there a correlation between the number of birth and the number of facts related by the police per km² in 2010 ?

```
ggplot(stats_km2) +
  geom_point(aes(nb_enfants_bykm, nb_faits_bykm))
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
cor(stats_km2$nb_enfants_bykm, stats_km2$nb_faits_bykm, use = "pairwise.complete.obs")
```

```
## [1] 0.9909755
```

Is this correlation value really interesting ? Think about the distribution of the data. . .

0.2.8 What is the regional density (in number/km²) of the 15 most given first names in France ?

- Filter the 15 most given first names in France
- Create a unique wide table with the department as observations and the 15 most given names in columns (as variables): the count is at the row-column intersection
- Merge with the surface department infos
- Compute the region surface and the density of names by region (e.g. number of people named "Bob", "Anna", ... by km² of each region)
  - Region name is stored in variable `NOM_REG`. (There are multiple departments in each region)

```
# Top 15
top_prenoms <- prenoms %>%
  group_by(name) %>%
  summarise(total = sum(n)) %>%
  arrange(desc(total), name) %>%
  top_n(total, n = 15)

# Filter on top 15 and spread
top_spread <- prenoms %>%
  filter(name %in% pull(top_prenoms, name)) %>%
  group_by(dpt, name) %>%
  summarize(total = sum(n)) %>%
  spread(name, total)

# Join with surface in km2
top_dpt <- dpt_data_modif %>%
  mutate(surface_km = surface_m / 1e6) %>%
  inner_join(top_spread, by = c("CODE_DEPT" = "dpt"))

# Calculate Regional surface and total number by name
top_region <- top_dpt %>%
  group_by(NOM_REG) %>%
  summarise_if(is.numeric, sum)

# Calculate density
top_region_density <- top_region %>%
  mutate_at(vars(pull(top_prenoms, name)),
            funs(./surface_km))

top_region_density
```

```
## # A tibble: 13 x 19
##    NOM_REG CODE_REG surface_m surface_km  Alain André Bernard Claude Daniel
##    <chr>      <int>     <dbl>      <dbl>  <dbl> <dbl>   <dbl>  <dbl>  <dbl>
##  1 ALSACE~      440   5.77e10     57699. 0.804  1.25   0.902  0.876  0.849
##  2 AQUITA~      900   8.51e10     85103. 0.570  0.893  0.561  0.505  0.380
##  3 AUVERG~     1008   7.08e10     70796. 0.752  1.10   0.721  0.550  0.607
##  4 BOURGO~      216   4.80e10     47982. 0.528  0.914  0.622  0.562  0.583
##  5 BRETAG~      212   2.74e10     27383. 1.08   1.43   0.817  0.669  0.739
##  6 CENTRE~      144   3.95e10     39471. 0.535  0.834  0.591  0.568  0.488
##  7 CORSE        188   8.76e 9      8757. 0.0978 0.169  0.0393 0.0617 0.0541
##  8 ILE-DE~       88   1.21e10     12065. 7.41   8.18   5.01   7.16   6.10
##  9 LANGUE~      988   7.34e10     73413. 0.512  0.803  0.426  0.450  0.287
## 10 NORD-P~      160   3.20e10     32008. 1.73   2.64   1.94   1.49   1.86
## 11 NORMAN~      140   3.01e10     30119. 1.04   1.56   1.15   1.21   1.14
## 12 PAYS D~      260   3.24e10     32363. 0.868  1.17   0.960  0.882  0.773
## 13 PROVEN~      558   3.17e10     31675. 0.906  0.916  0.504  0.672  0.516
## # ... with 10 more variables: Jacques <dbl>, Jean <dbl>, Jeanne <dbl>,
## #   Louis <dbl>, Marcel <dbl>, Marie <dbl>, Michel <dbl>, Philippe <dbl>,
## #   Pierre <dbl>, René <dbl>
```

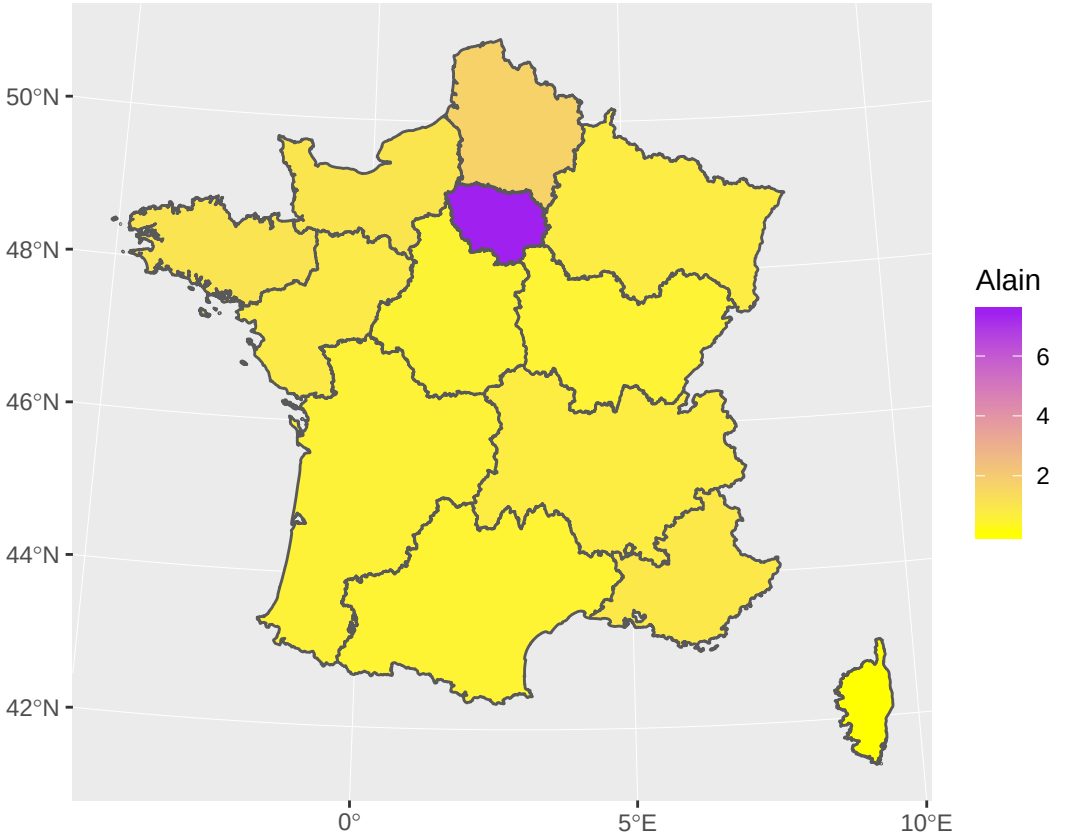
Bonus question : map the mean regional density (in number/km²) of the 15 most given first names in France

- Use the "department" shapefile to cross information and map data
  - Region name is stored in variable `NOM_REG`. (There are multiple departments in each region)
  - One map for each name

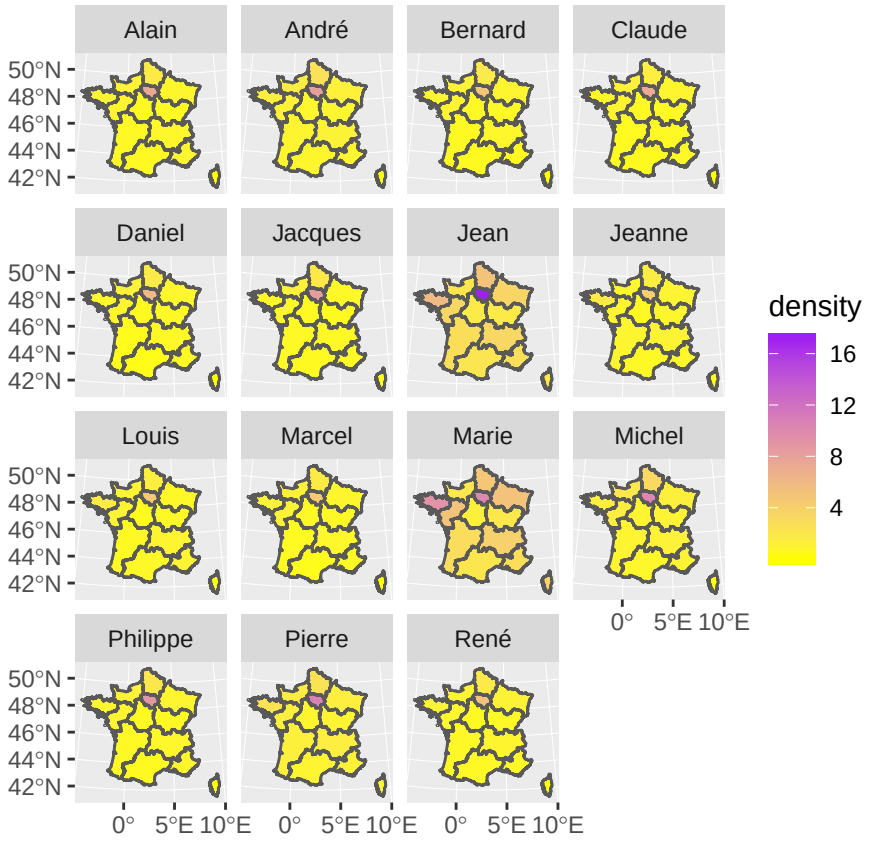
```
region <- st_read(dsn = 'data/departements',
                  layer = 'DEPARTEMENT',
                  quiet = TRUE) %>%
  group_by(NOM_REG) %>%
  summarise() %>%
  inner_join(top_region_density, by = "NOM_REG")
```

```
## Warning: Column `NOM_REG` joining factor and character vector, coercing
## into character vector
```

```
# One map for one name
ggplot(region) +
  aes(fill = Alain) +
  geom_sf() +
  coord_sf(crs = 2154) +
  scale_fill_gradient(low = "yellow", high = "purple")
```




```
# As facets for all names
region %>%
  gather(key = "name", value = "density", Alain:René) %>%
  ggplot() +
  aes(fill = density) +
  geom_sf() +
  coord_sf(crs = 2154) +
  scale_fill_gradient(low = "yellow", high = "purple") +
  facet_wrap(~name)
```


Alain
André
Bernard
Claude
Daniel
Jacques
Jean
Jeanne
Louis
Marcel
Marie
Michel
Philippe
Pierre
René
50°N
48°N
46°N
44°N
42°N
0°
5°E
10°E
density
16
12
8
4