Luxemburg Project

Table of contents

0.1 Luxemburg Data Project

```
library(dplyr)####patates
library(purrr)
library(readxl)
library(stringr)
library(janitor)
```

0.2 Gettting Data

```
url <- "https://github.com/b-rodrigues/rap4all/raw/master/datasets/vente-maison-2010-2021.


# Shortened url

#url <- "https://is.gd/1vvBAc"


raw_data <- tempfile(fileext = ".xlsx")
```

```r
download.file(url, raw_data, method = "auto", mode = "wb")


sheets <- excel_sheets(raw_data)


read_clean <- function(..., sheet){

  read_excel(..., sheet = sheet) |>

    mutate(year = sheet)

}


raw_data <- map(

  sheets,

  ~read_clean(raw_data,

              skip = 10,

              sheet = .)

) |>

  bind_rows() |>

  clean_names()
```

```
New names:
* `*` -> `*...3`
* `*` -> `*...4`
```

Let's see the neat data:

```r
raw_data
```

```
# A tibble: 1,343 x 9
   commune    nombre_doffres prix_moyen_annonce_e~1 prix_moyen_annonce_a~2 year
   <chr>                <dbl> <chr>                  <chr>                  <chr>
 1 Bascharage             192 593698.31000000006     3603.57                2010
 2 Beaufort               266 461160.29              2902.76                2010
 3 Bech                    65 621760.22              3280.51                2010
 4 Beckerich              176 444498.68              2867.88                2010
 5 Berdorf                111 504040.85              3055.99                2010
 6 Bertrange              264 795338.87              4266.46                2010
 7 Bettembou~             304 555628.29              3343.22                2010
 8 Bettendorf              94 495074.38              3235.26                2010
 9 Betzdorf               119 625914.47              3343.05                2010
10 Bissen                  70 516465.57              3321.65                2010
# i 1,333 more rows
# i abbreviated names: 1: prix_moyen_annonce_en_courant,
#   2: prix_moyen_annonce_au_m2_en_courant
# i 4 more variables: bech <chr>, x12 <dbl>, x3 <chr>, x4 <chr>
```

Some variables has their original names and we will change them to English.

```
raw_data <- raw_data |>

  rename(

    locality = commune,

    n_offers = nombre_doffres,

    average_price_nominal_euros = prix_moyen_annonce_en_courant,

    # average_price_m2_nominal_euros = prix_moyen_annonce_au_m2_en_courant,

    average_price_m2_nominal_euros = prix_moyen_annonce_au_m2_en_courant

  ) |>

  mutate(locality = str_trim(locality)) |>

  select(year, locality, n_offers, starts_with("average"))
```

```
raw_data |>
  filter(grepl('Luxembourg', locality)) |>
  count(locality)
```

```
# A tibble: 2 x 2
  locality             n
  <chr>            <int>
1 Luxembourg           9
2 Luxembourg-Ville     2
```

```
raw_data |>
  filter(grepl('P.tange', locality)) |>
  count(locality)
```

```
# A tibble: 2 x 2
  locality     n
  <chr>    <int>
1 Petange      9
2 Pétange      2
```

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see https://quarto.org.

0.3 Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
1 + 1
```

```
[1] 2
```

You can add options to executable code like this

```
[1] 4
```

The `echo: false` option disables the printing of code (only output is displayed).