# Bivariate analysis

#### 2025-02-04

M1 MIDS/MFA/LOGOS Université Paris Cité Année 2024 Course Homepage Moodle



## Objectives

In Exploratory analysis of tabular data, bivariate analysis is the second step. It consists in exploring, summarizing, visualizing pairs of columns of a dataset.

#### Setup

```
stopifnot(
  require(glue),
  require(magrittr),
  require(lobstr),
  require(arrow),
  require(ggforce),
  require(vcd),
  require(ggmosaic),
  require(fattr),
  require(patchwork),
  require(corrr),
  require(gapminder),
  require(slider),
  require(tidyverse)
)
```

Bivariate techniques depend on the types of columns we are facing.

For numerical/numerical samples

- Scatter plots
- Smoothed lineplots (for example linear regression)
- 2-dimensional density plots

For categorical/categorical samples: mosaicplots and variants

For numerical/categorical samples

• Boxplots per group

- Histograms per group
- Density plots per group

#### Dataset

Once again we rely on the Recensement dataset.

Since 1948, the US Census Bureau carries out a monthly Current Population Survey, collecting data concerning residents aged above 15 from 150000 households. This survey is one of the most important sources of information concerning the american workforce. Data reported in file Recensement.txt originate from the 2012 census.

Load the data into the session environment and call it df. Take advantage of the fact that we saved the result of our data wrangling job in a self-documented file format. Download a parquet file from the following URL:

https://stephane-v-boucheron.fr/data/Recensement.parquet

#### i Question

Download a parquet file from the following URL: https://stephane-v-boucheron.fr/data/Recensement.parquet



- Use httr::GET() and WriteBin()
- Use download.file()
- Use fs to handle files and directories

#### Manage the DATA sub-directory

```
if (fs::dir_exists('DATA')){
  datapath <- "DATA"
} else {
  datapath <- "../DATA"
}</pre>
```

#### Using httr::get and writeBin

```
fname <- "Recensement.parquet"</pre>
fpath <- paste(datapath, fname, sep="/")</pre>
url <- 'https://stephane-v-boucheron.fr/data/Recensement.parquet'</pre>
if (!file.exists(fpath)) {
 tryCatch(
    expr = {
      rep <- httr::GET(url)</pre>
      stopifnot(rep$status_code==200)
      con <- file(fpath, open="wb")</pre>
      writeBin(rep$content, con)
      close(con)
    },
    warning = function(w) {
      glue("Successful download but {w}")
    },
    error = function(e) {
      stop("Houston, we have a problem!") # error-handler-code
    },
    finally = {
      if (exists("con") && isOpen(con)){
        close(con)
      }
    }
  )
}
```

```
if (!file.exists(fpath)) {
   tryCatch(
   expr = {
     download.file(url, fpath, mode="wb", quiet=T)
     print(glue::glue('>>> file downloaded at {fpath}\n'))
   },
   warning = function(w) {
     glue::glue("Successful download but {w}")
   },
   error = function(e) {
     stop("Houston, we have a problem!") # error-handler-code
```

#### i Question

Load the data contained in the downloaded file into the session environment and call it df

#### Solution

```
df <- arrow::read_parquet(fpath)</pre>
```

```
df |>
 glimpse()
## Rows: 599
## Columns: 11
## $ AGE
               <dbl> 58, 40, 29, 59, 51, 19, 64, 23, 47, 66, 26, 23, 54, 44, 56,~
## $ SEXE
               <fct> F, M, M, M, M, M, F, F, M, F, M, F, F, F, F, F, F, M, M, F, ~
## $ REGION
               <fct> NE, W, S, NE, W, NW, S, NE, NW, S, NE, NE, W, NW, S, S, NW, ~
## $ STAT MARI <fct> C, M, C, D, M, C, M, C, M, D, M, C, M, C, M, C, S, M, S, C,~
               <dbl> 13.25, 12.50, 14.00, 10.60, 13.00, 7.00, 19.57, 13.00, 20.1~
## $ SAL HOR
## $ SYNDICAT
               <fct> non, non, non, oui, non, non, non, non, oui, non, non, non, ~
## $ CATEGORIE <fct> "Administration", "Building ", "Administration", "Services"~
## $ NIV_ETUDES <fct> "Bachelor", "12 years schooling, no diploma", "Associate de~
## $ NB PERS
               <fct> 2, 2, 2, 4, 8, 6, 3, 2, 3, 1, 3, 2, 6, 5, 4, 4, 3, 2, 3, 2, ~
## $ NB ENF
               ## $ REV_FOYER <fct> [35000-40000), [17500-20000), [75000-1e+05), [17500-20000),~
df |>
 head()
## # A tibble: 6 x 11
      AGE SEXE REGION STAT MARI SAL HOR SYNDICAT CATEGORIE
                                                              NIV ETUDES NB PERS
    <dbl> <fct> <fct> <fct>
                                                                         <fct>
                                  <dbl> <fct>
                                                 <fct>
                                                               <fct>
                NE
                                                 "Administrat~ Bachelor
## 1
       58 F
                       C
                                   13.2 non
                                                                         2
## 2
       40 M
                W
                       Μ
                                   12.5 non
                                                 "Building "
                                                              12 years ~ 2
## 3
       29 M
                S
                       C
                                        non
                                                 "Administrat~ Associate~ 2
                                   14
## 4
       59 M
                       D
                                                 "Services"
                NE
                                   10.6 oui
                                                             12 years ~ 4
## 5
       51 M
                       Μ
                                   13
                                        non
                                                 "Services"
                                                              9 years s~ 8
                                                 "Services"
## 6
       19 M
                NW
                       C
                                    7
                                                              12 years ~ 6
                                        non
## # i 2 more variables: NB_ENF <fct>, REV_FOYER <fct>
```

## Categorical/Categorical pairs

#### i Question

Project the dataframe on categorical columns

	SEXE	REGION	STAT_MARI	SYNDICAT	CATEGORIE	NIV_ETUDES	NB_PERS	NB_ENF	REV_FOYER
	<fct></fct>								
1	F	NE	C	non	"Administ~	Bachelor	2	0	[35000-4~
2	M	W	M	non	"Building~	12 years ~	2	0	[17500-2~
3	M	S	C	non	"Administ~	Associate~	2	0	[75000-1~
4	M	NE	D	oui	"Services"	12 years ~	4	1	[17500-2~
5	M	W	M	non	"Services"	9 years s~	8	1	[75000-1~
6	M	NW	C	non	"Services"	12 years ~	6	0	[1e+05-1~
									1

## i Question

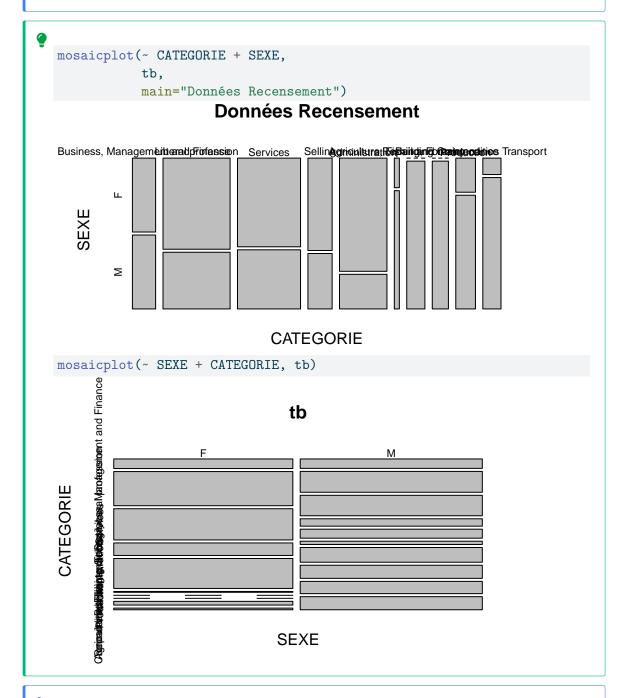
- Explore the connection between CATEGORIE and SEX.
- Compute the 2-ways contingency table using table(), and count() from dplyr.



- Use tibble::as\_tibble() to transform the output of table() into a dataframe/tibble.
- Use tidyr::pivot\_wider() so as to obtain a wide (but messy) tibble with the same the same shape as the output of table().
- Can you spot a difference?

```
Solution
tb <- df |>
  dplyr::select(CATEGORIE, SEXE) |>
  table()
# tb
tb2 <- df |>
  count(CATEGORIE, SEXE)
tb2
# A tibble: 18 x 3
   CATEGORIE
                                        SEXE
   <fct>
                                        <fct> <int>
 1 "Business, Management and Finance" F
 2 "Business, Management and Finance" M
 3 "Liberal profession"
                                                 82
 4 "Liberal profession"
                                                 51
                                        Μ
 5 "Services"
                                        F
                                                 75
 6 "Services"
                                        Μ
                                                 50
 7 "Selling"
                                        F
                                                 30
 8 "Selling"
                                        Μ
                                                 18
                                        F
                                                 72
 9 "Administration"
10 "Administration"
                                                 22
11 "Agriculture, Fishing, Forestry"
                                        F
                                                  2
                                       M
12 "Agriculture, Fishing, Forestry"
                                                  8
13 "Building "
                                        Μ
                                                 36
14 "Repair and maintenance"
                                        Μ
                                                 32
15 "Production"
                                        F
                                                  9
16 "Production"
                                        Μ
                                                 30
17 "Commodities Transport"
                                        F
                                                  4
18 "Commodities Transport"
                                                 32
tb2 |>
  pivot_wider(id_cols=CATEGORIE,
              names_from=SEXE,
               values_from=n)
# A tibble: 10 x 3
   CATEGORIE
                                            F
                                                  М
                                        <int> <int>
   <fct>
 1 "Business, Management and Finance"
                                           23
                                                 23
 2 "Liberal profession"
                                                 51
                                           75
 3 "Services"
                                                 50
                                           30
 4 "Selling"
                                                 18
                                           72
                                                 22
 5 "Administration"
 6 "Agriculture, Fishing, Forestry"
                                           2
                                                 8
 7 "Building "
                                           NA
                                                 36
 8 "Repair and maintenance"
                                           NA
                                                 32
 9 "Production"
                                            9
                                                 30
10 "Commodities Transport"
                                            4
                                                 32
```

Use mosaicplot() from base R to visualize the contingency table.



## i Question

Use geom\_mosaic from ggmosaic to visualize the contingency table

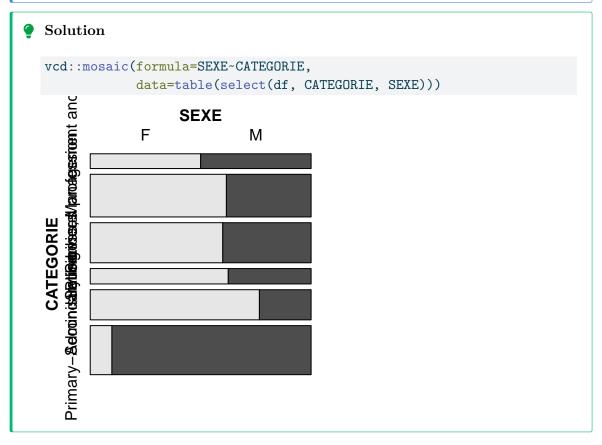
- Make the plot as readable as possible
- Reorder CATEGORIE according to counts

```
rot_x_text <- theme(</pre>
  axis.text.x = element_text(angle = 45)
df |>
  ggplot() +
  geom_mosaic(aes(x=product(SEXE, CATEGORIE), fill=SEXE)) +
  rot_x_text
Warning: The `scale_name` argument of `continuous_scale()` is deprecated as of ggplot2
3.5.0.
Warning: The `trans` argument of `continuous_scale()` is deprecated as of ggplot2 3.5
i Please use the `transform` argument instead.
Warning: `unite_()` was deprecated in tidyr 1.2.0.
i Please use `unite()` instead.
i The deprecated feature was likely used in the ggmosaic package.
  Please report the issue at <a href="https://github.com/haleyjeppson/ggmosaic">https://github.com/haleyjeppson/ggmosaic</a>.
   M
                                                                       SEXE
SEXE
                                                                           F
                                                      nd production transport
                                        Adminstrate February Repair and Production
                                                                           Μ
                              CATEGORIE
```

• Collapse rare levels of CATEGORIE (consider that a level is rare if it has less than 40 occurrences). Use tools from forcats.

```
df |>
  count(CATEGORIE) |>
 arrange(desc(n))
# A tibble: 10 x 2
   CATEGORIE
                                           n
   <fct>
                                       <int>
 1 "Liberal profession"
                                         133
 2 "Services"
                                         125
3 "Administration"
                                          94
 4 "Selling"
                                          48
5 "Business, Management and Finance"
                                          46
 6 "Production"
                                          39
7 "Building "
                                          36
8 "Commodities Transport"
                                          36
 9 "Repair and maintenance"
                                          32
10 "Agriculture, Fishing, Forestry"
                                          10
rare_categories <- df |>
 count(CATEGORIE) |>
  filter(n<=40)
rare_categories
# A tibble: 5 \times 2
  CATEGORIE
                                        n
1 "Agriculture, Fishing, Forestry"
                                       10
2 "Building "
                                       36
3 "Repair and maintenance"
                                       32
4 "Production"
                                       39
5 "Commodities Transport"
```

# i Question Same as above with vcd::mosaic



## Testing association

## Chi-square independence/association test

https://statsfonda.github.io/site/content/ch4\_2.html#test-dindépendance

## i Question

- Compute the chi-square association statistic between CATEGORIE and SEXE.
- Display the output of chisq.test() as a table, using broom::tidy()

```
test_1 <- df |>
  select(CATEGORIE, SEXE) |>
  table() |>
  chisq.test()
# test_1
test_1 |>
  broom::tidy() |>
 knitr::kable()
                   p.value
                                        method
          statistic
                           parameter
          140.6717
                         0
                                        Pearson's Chi-squared test
                                    5
```

Compute the Chi-square statistics from the contingeny table

```
rowcounts <- apply(tb, MARGIN = 1, FUN = sum)
colcounts <- apply(tb, MARGIN = 2, FUN = sum)

expected <- (rowcounts %*% t(colcounts))/sum(colcounts)

# norm((tb - expected) / sqrt(expected), type = "F")^2

expected |>
as_tibble() |>
knitr::kable()

F M

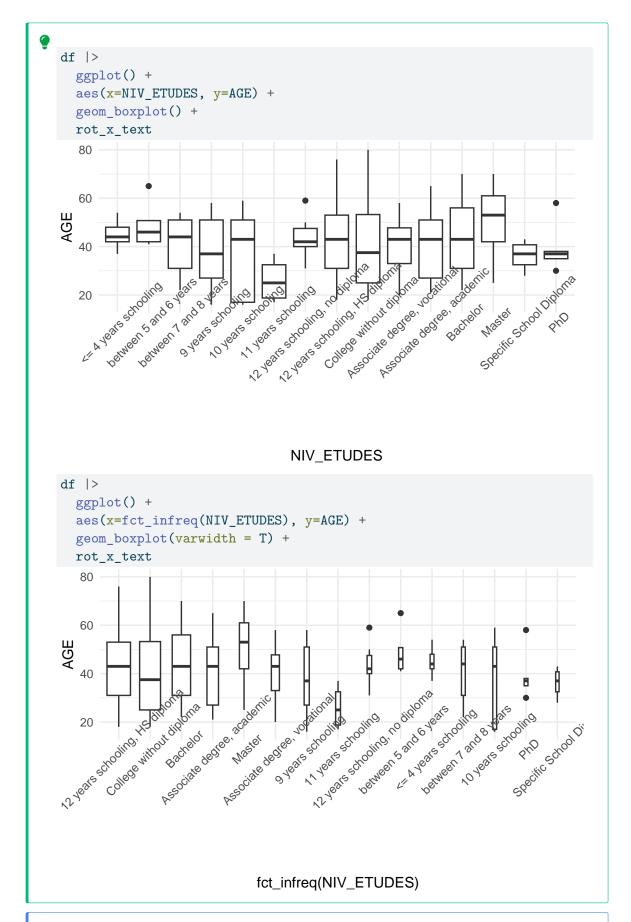
22.80801 23.19199
65.94491 67.05509
61.97830 63.02170
23.79967 24.20033
46.60768 47.39232
75.86144 77.13856
```

# Categorical/Numerical pairs

## Grouped boxplots

#### i Question

Plot boxplots of AGE according to NIV\_ETUDES



## i Question

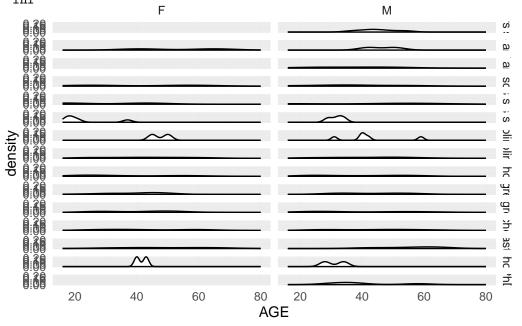
Draw density plots of AGE, facet by NIV\_ETUDES and SEXE



Warning: Groups with fewer than two data points have been dropped. Groups with fewer than two data points have been dropped.

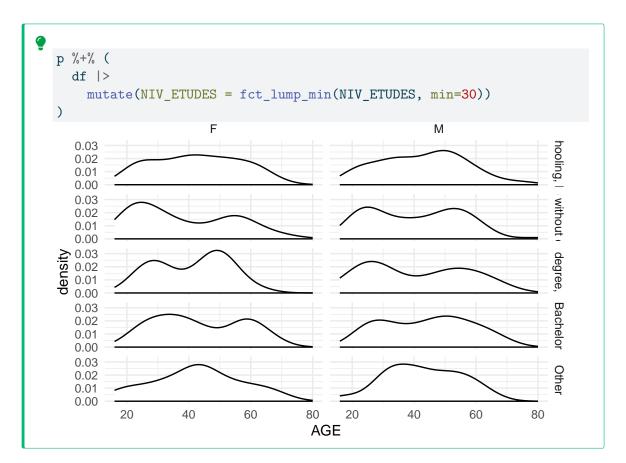
Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -Inf

Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -Inf



## **Q**uestion

Collapse rare levels of NIV\_ETUDES and replay.

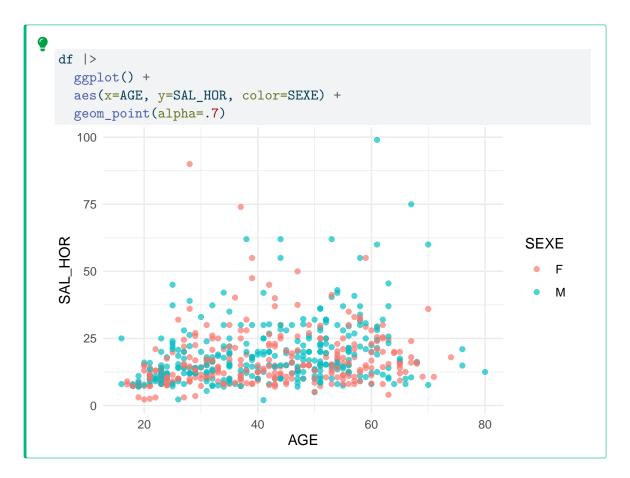


# Numerical/Numerical pairs

## Scatterplots

## i Question

Make a scatterplot of  $\mathtt{SAL\_HOR}$  with respect to  $\mathtt{AGE}$ 



## Correlations

- Linear correlation coefficient (Pearson)
- Linear rank correlation coefficient (Spearman  $\rho$ , Kendall  $\tau$ )
- $\xi$  rank correlation coefficient (Chatterjee)

## Linear correlation coefficient

## i Question

Compute the Pearson, Spearman and Kendall correlation coefficients between AGE and SAL\_HOR using function cor() from base R

```
Solution
df |>
  summarise(
    pearson=cor(AGE, SAL_HOR),
    spearman=cor(AGE, SAL_HOR, method = "spearman"),
    kendall=cor(AGE, SAL_HOR, method="kendall")) |>
  gt::gt() |>
  gt::fmt_number(decimals=2) |>
  gt::tab_caption(
    "Correlation coefficients between SAL_HOR and AGE\nRecensement dataset"
                                          kendall
                               spearman
                      pearson
                                             0.22
                         0.25
                                    0.32
```

#### Rank based methods

Spearman's rho ( ) and Kendall's tau ( ) are both non-parametric correlation coefficients used to measure the strength and direction of a monotonic relationship between two variables.

**Spearman's rho** () Based on rank differences. Defined as the Pearson correlation coefficient between the ranked variables.

$$\rho=1-\frac{6\sum d_i^2}{n(n^2-1)}$$

where  $d_i$  is the difference between the ranks of each pair, and n is the number of observations.

**Kendall's tau ()** Based on concordant and discordant pairs. Measures the proportion of pairs that have the same order in both variables compared to the total number of pairs.

$$\tau = \frac{(C-D)}{\frac{1}{2}n(n-1)}$$

where C is the number of concordant pairs, and D is the number of discordant pairs.

#### When to Use Which?

Factor	Spearman's rho ( )	Kendall's tau ( )
Large differences in ranks	More sensitive	Less sensitive
Small sample sizes	Less reliable	More reliable
Outlier resistance	Moderate	High
Computational efficiency	Faster	Slower (due to pairwise comparisons)
Interpretation	Similar to Pearson's correlation	More intuitive (proportion of concordance)

## Chatterjee's correlation coefficient (Chatterjee's $\xi$ )

The three most popular classical measures of statistical association are Pearson's correlation coefficient, Spearman's , and Kendall's . These coefficients are very powerful for detecting linear or monotone associations, and they have well-developed asymptotic theories for calculating P-values. However, the big problem is that they are not effective for detecting associations that are not monotonic, even in the complete absence of noise.

Let (X,Y) be a pair of random variables, where Y is not a constant. Let  $(X_1,Y_1),\ldots,(X_n,Y_n)$  be i.i.d. pairs with the same law as (X,Y), where  $n\geq 2$ . The new coefficient has a simpler formula if the  $X_i$ 's and the  $Y_i$  's have no ties. This simpler formula is presented first, and then the general case is given. Suppose that the  $X_i$ 's and the  $Y_i$  's have no ties. Rearrange the data as  $(X_{(1)},Y_{(1)}),\ldots,(X_{(n)},Y_{(n)})$  such that  $X_{(1)}\leq\cdots\leq X_{(n)}$ . Since the  $X_i$ 's have no ties, there is a unique way of doing this. Let ri be the rank of  $Y_{(i)}$ , that is, the number of  $Y_{(i)}$  such that  $Y_{(i)}\leq Y_{(i)}$ . The new correlation coefficient is defined as

$$\xi_n(X,Y) := 1 - 3 \sum_{i=1}^{n-1} \frac{|r_{i+1} - r_i|}{n^2 - 1}$$

In the presence of ties,  $\xi_n$  is defined as follows. If there are ties among the  $X_i$ 's, then choose an increasing rearrangement as above by breaking ties uniformly at random. Let  $r_i$  be as before, and additionally define  $l_i$  to be the number of j such that  $Y_{(j)} \geq Y_{(i)}$ . Then define

$$\xi_n(X,Y) := 1 - 3n \sum_{i=1}^{n-1} \frac{|r_{i+1} - r_i|}{2 \sum_{i=1}^n l_i (n - l_i)}$$

When there are no ties among the  $Y_i$ 's,  $l_1, \ldots, l_n$  is just a permutation of  $1, \ldots, n$ , and so the denominator in the above expression is just  $n(n^2-1)/3$ , which reduces this definition to the earlier expression.

From Sourav Chatterjee: A new correlation coefficient

#### i Question

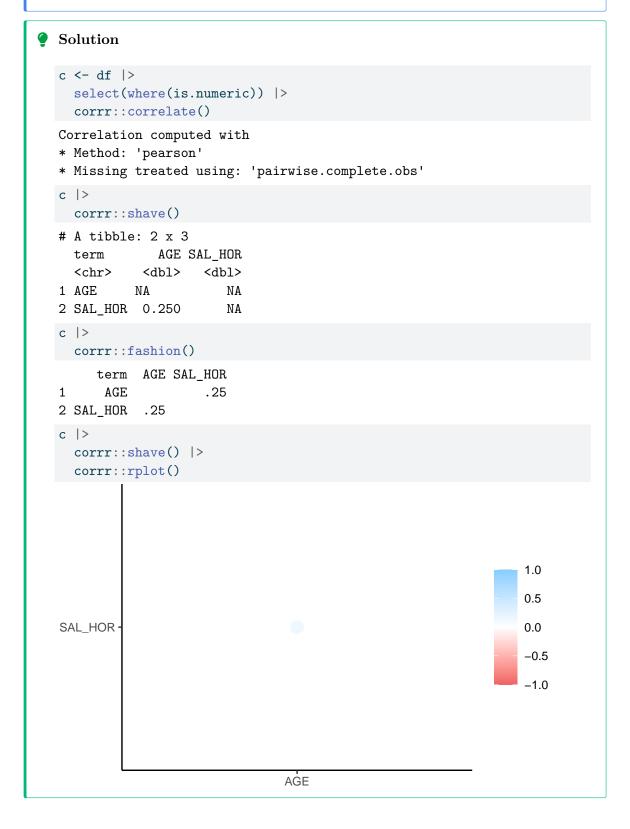
Write a dplyr pipeline from computing the  $\xi$  correlation coefficient between Y=lifeExp and X=gdpPercap in the gapminder dataset, per year and continent.

```
tab_xi <- gapminder::gapminder |>
  group_by(year, continent) |>
  arrange(gdpPercap) |>
  mutate(rnk= row_number(lifeExp),
         lnk=rank(desc(lifeExp), ties.method = "max"),
         N=n()) |>
  mutate(fol=lead(rnk), dd=abs(fol-rnk)) |>
  summarise(Xi=1-n()*sum(dd, na.rm = T)/(2*sum(lnk*(N-lnk), na.rm = T)))
`summarise()` has grouped output by 'year'. You can override using the
`.groups` argument.
tab_xi |>
  print(n=10)
# A tibble: 60 x 3
            year [12]
# Groups:
    year continent
                        Χi
   <int> <fct>
                     <dbl>
 1 1952 Africa
                    0.0377
 2 1952 Americas 0.135
 3 1952 Asia
                    0.244
 4 1952 Europe
                    0.610
   1952 Oceania
                    0
   1957 Africa
                    0.0466
 7
   1957 Americas 0.240
                    0.330
 8 1957 Asia
9 1957 Europe
                    0.516
10 1957 Oceania
# i 50 more rows
tab_xi |>
  pivot_wider(
    id_cols=continent,
    names_from=year,
    values_from=Xi
    ) |>
  gt::gt() |>
  gt::fmt_number(decimals=2)
            1952
                  1957
                         1962
                                1967
                                      1972
                                                   1982
                                                          1987
                                                                1992
                                                                       1997
                                                                             2002
                                                                                    2007
 continent
                                             1977
                                      0.24
                   0.05
                                             0.27
                                                          0.27
                                                                       0.31
                                                                              0.13
                                                                                    0.17
  Africa
            0.04
                        -0.01
                                0.05
                                                    0.38
                                                                 0.36
 Americas
            0.13
                   0.24
                         0.35
                                0.17
                                             0.13
                                                    0.30
                                                          0.43
                                                                 0.52
                                                                       0.33
                                                                              0.29
                                                                                    0.23
                                       0.19
   Asia
            0.24
                   0.33
                         0.38
                                0.27
                                      0.32
                                             0.26
                                                    0.29
                                                          0.30
                                                                 0.43
                                                                       0.50
                                                                              0.50
                                                                                    0.46
  Europe
            0.61
                   0.52
                                0.38
                                      0.32
                                             0.38
                                                    0.06
                                                          0.29
                                                                 0.39
                                                                       0.51
                                                                              0.48
                                                                                    0.51
                         0.40
 Oceania
            0.00
                   0.00
                         0.00
                                0.00
                                      0.00
                                             0.00
                                                    0.00
                                                          0.00
                                                                 0.00
                                                                       0.00
                                                                              0.00
                                                                                    0.00
```

## Using package corrr

https://corrr.tidymodels.org

Use corre::correlate() and companion functions to display correlation coefficients in a friendly way.

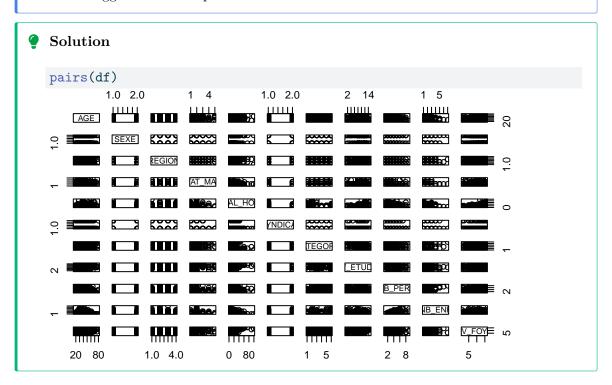


## pairs from base R

Just as function skim from package skimr allows us to automate univariate analysis, function pairs from base R allows us to automate bivariate analysis.

#### i Question

- Apply function pairs to the Recensement dataset
- How does pairs handle pairs of categorical columns?
- How does pairs handle pairs of numerical columns?
- How does pairs handle categorical/numerical columns?
- Suggestions for improvements?

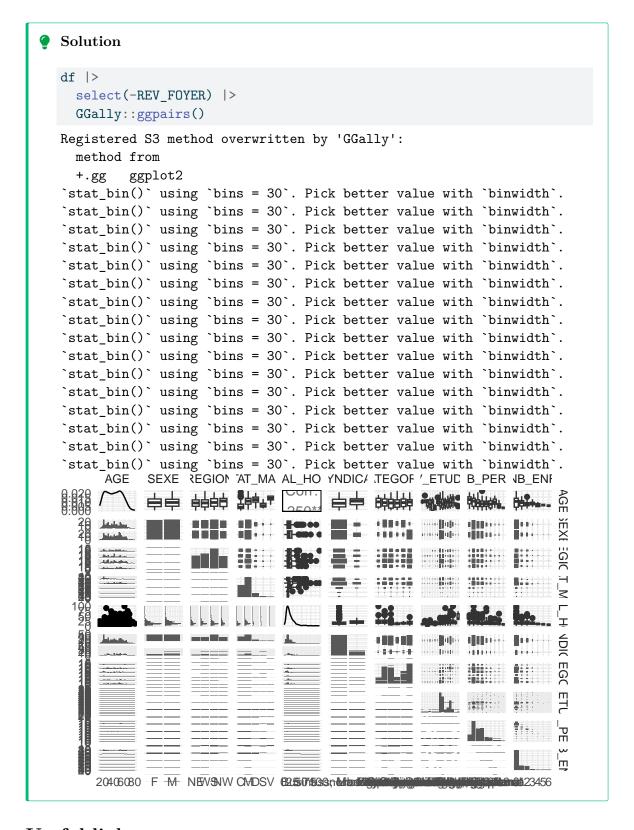


## ggpairs() from GGally

#### Documentation

## i Question

- Apply function GGally::ggpairs() to the Recensement dataset
- How does ggpairs handle pairs of categorical columns?
- How does ggpairs handle pairs of numerical columns?
- How does ggpairs handle categorical/numerical columns?
- How does ggpairs handle diagonal diagrams?
- How can you modify layout, labels, ticks?
- Suggestions for improvements?



## Useful links

- rmarkdown
- dplyr
- ggplot2
- R Graphic Cookbook. Winston Chang. O' Reilly.
- A blog on ggplot object
- skimr
- vcd

- ggmosaicggforcearrow

- httr