Phân tích thống kê và vẽ đồ thị xuất sắc với R

Duc Nguyen

03 November 2024

Table of contents

Lời nói đầu **3**

1 Vector **4**

2 Barchart **12**
2.1 So sánh hai nhóm 12

3 Data wrangling **22**
3.1 Chọn ngẫu nhiên số dòng trong dataset 22

4 Format **27**

Tài liệu tham khảo **28**

Lời nói đầu

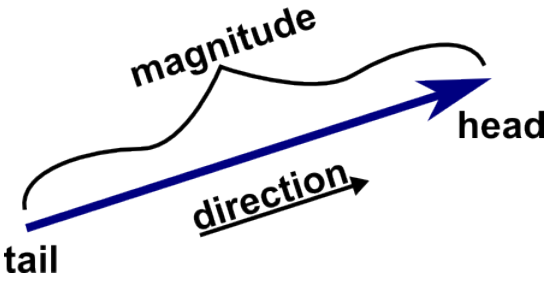
Cảm ơn tất cả mọi người đã, đang và sẽ làm việc với tôi qua câu chuyện R để tạo ra các đoạn code giúp thế giới trở nên tốt đẹp hơn. Trân trọng.

Duc Nguyen, *always a student.*

1 Vector

Trong toán học, vector hay hướng lượng (theo phiên âm Hán-Việt) là một đoạn thẳng có hướng. Đoạn thẳng này biểu thị phương, chiều, độ lớn (chiều dài của vector). Ví dụ trong mặt phẳng cho hai điểm phân biệt A và B bất kì ta có thể xác định được vector $\overrightarrow{AB}$.

> ***A vector*** is an object that has **both a magnitude** and **a direction**. Geometrically, we can picture a vector as a directed line segment, whose length is the magnitude of the vector and with an arrow indicating the direction. The direction of the vector is from its tail to its head [1].



Hai vector được xem là bằng nhau nếu có cùng hướng và cùng độ lớn (độ dài).

```
png(filename = "img/vector_ok.png",
    width = 10,
    height = 10,
    res = 300,
    units = "in")

par(pty = "s")
par(mar = c(0, 0, 0, 0))
par(oma = c(0, 0, 0, 0))

plot(x = 0,
     y = 0,
     type = "n",
     xlim = c(-11, 11),
     ylim = c(-11, 11),
     xaxs = "i",
```

```
     yaxs = "i",
     las = 1,
     xaxt = "n",
     yaxt = "n",
     bty = "o",
     xlab = "",
     ylab = "")

grid(nx = 22, ny = 22, col = "black")

axis(side = 1,
     at = -11:11,
     labels = NA,
     line = - (grconvertY(y = 0,
                          from = "user",
                          to = "lines") -
       grconvertY(y = -11,
                  from = "user",
                  to = "lines")),
     tick = FALSE)

# abline(h = 0)

segments(x0 = -0.2,
         x1 = 0.2,
         y0 = -11:11,
         y1 = -11:11,
         col = "black")

arrows(x0 = -11,
       x1 = 11,
       y0 = 0,
       y1 = 0,
       col = "black")

axis(side = 2,
     at = -11:11,
     labels = NA,
     line = - (grconvertX(x = 0,
                          from = "user",
                          to = "lines") -
       grconvertX(x = -11,
```

```
                  from = "user",
                  to = "lines")),
     las = 1,
     tick = FALSE)

# abline(v = 0)

arrows(y0 = -11,
       y1 = 11,
       x0 = 0,
       x1 = 0,
       col = "black")

segments(y0 = -0.2,
         y1 = 0.2,
         x0 = -11:11,
         x1 = -11:11,
         col = "black")

points(x = 0,
       y = 0,
       col = "black",
       pch = 19,
       cex = 1.5)

text(x = -0.3,
     y = 0.3,
     pos = 2,
     labels = 0,
     cex = 1.2)

text(x = -10:10,
     y = -0.3,
     labels = c(-10:-1, NA, 1:10),
     pos = 1,
     cex = 1.2,
     xpd = NA)

text(y = -10:10,
     x = -0.3,
     labels = c(-10:-1, NA, 1:10),
     pos = 2,
```

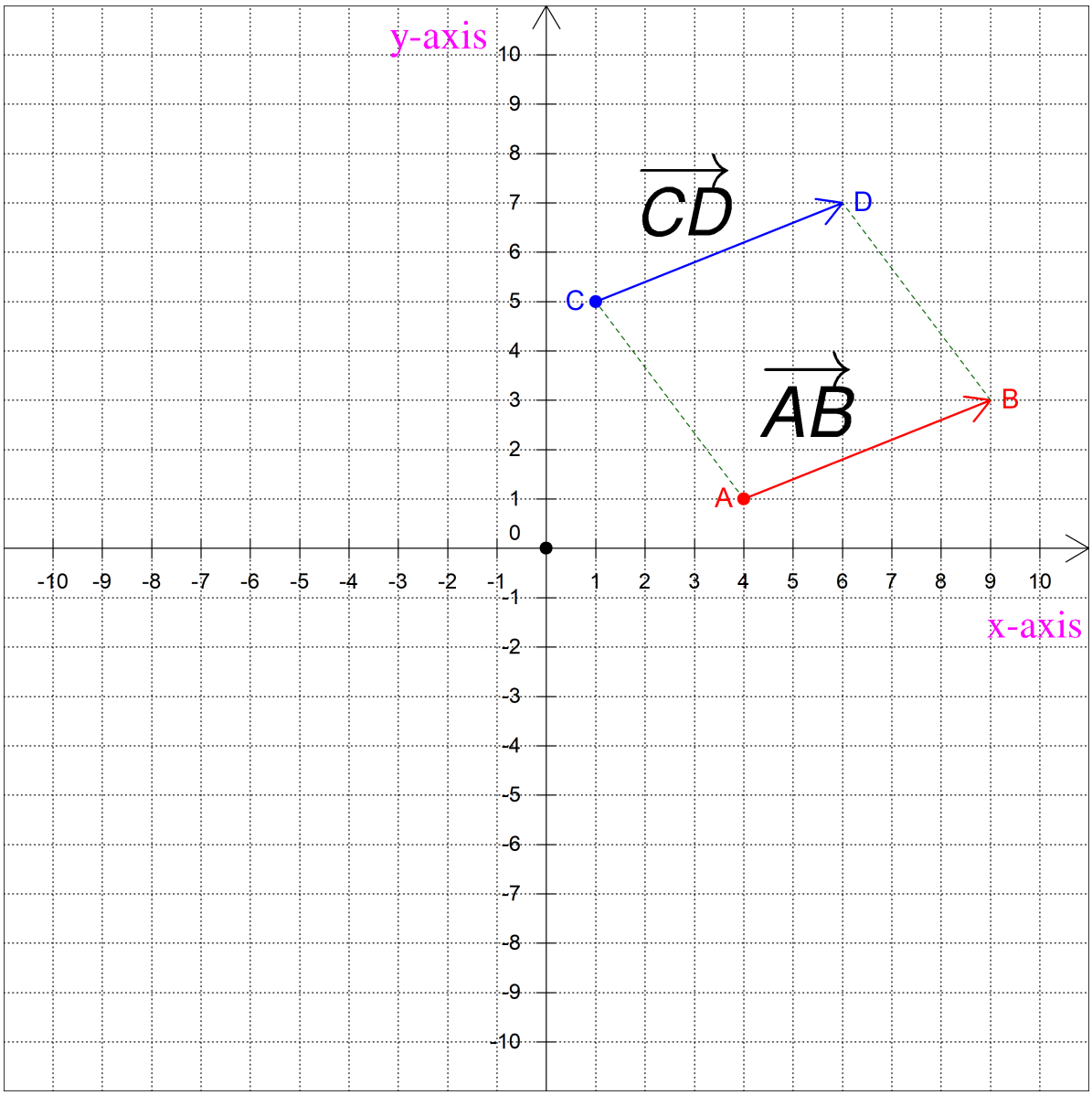
```
     cex = 1.2,
     xpd = NA)

###

arrows(x0 = 4,
       x1 = 9,
       y0 = 1,
       y1 = 3,
       col = "red",
       lwd = 2)

arrows(x0 = 4-3,
       x1 = 9-3,
       y0 = 1+4,
       y1 = 3+4,
       col = "blue",
       lwd = 2)

segments(x0 = 4,
       x1 = 4-3,
       y0 = 1,
       y1 = 1+4,
       col = "darkgreen",
       lwd = 1,
       lty = 2)

segments(x0 = 9,
       x1 = 9-3,
       y0 = 3,
       y1 = 3+4,
       col = "darkgreen",
       lwd = 1,
       lty = 2)

###

points(x = 4,
      y = 1,
      col = "red",
      pch = 19,
      cex = 1.5)
```

```
text(x = 4,
     y = 1,
     col = "red",
     pos = 2,
     labels = "A",
     cex = 1.5)

text(x = 9,
     y = 3,
     col = "red",
     pos = 4,
     labels = "B",
     cex = 1.5)

###

points(x = 1,
       y = 5,
       col = "blue",
       pch = 19,
       cex = 1.5)

text(x = 1,
     y = 5,
     col = "blue",
     pos = 2,
     labels = "C",
     cex = 1.5)

text(x = 6,
     y = 7,
     col = "blue",
     pos = 4,
     labels = "D",
     cex = 1.5)

###

library(exams)
options(exams_tex = "tools")

header_ok <- c("\\usepackage{helvet}",
```

```
            "\\IfFileExists{sfmath.sty}{\\RequirePackage[helvet]{sfmath}}{}",
            "\\renewcommand{\\sfdefault}{phv}",
            "\\renewcommand{\\rmdefault}{phv}",
            "\\usepackage[utf8]{vietnam}",
            "\\usepackage{times}",
            "\\usepackage{xcolor}")

exams::tex2image(tex = "\\textcolor[HTML]{FF00FF}{x-axis}",
                 format = "svg",
                 density = 1000,
                 resize = 1000,
                 dir = paste0(getwd(), "/img"),
                 name = "x_axis",
                 show = FALSE,
                 header = header_ok)

library(grImport2)
p_1 <- grImport2::readPicture("img/x_axis.svg" )


grImport2::grid.picture(p_1,
                        x = 0.95,
                        y = 0.43,
                        width = 0.1
                        )

###

exams::tex2image(tex = "\\textcolor[HTML]{FF00FF}{y-axis}",
                 format = "svg",
                 density = 1000,
                 resize = 1000,
                 dir = paste0(getwd(), "/img"),
                 name = "y_axis",
                 show = FALSE,
                 header = header_ok)

p_2 <- grImport2::readPicture("img/y_axis.svg" )


grImport2::grid.picture(p_2,
                        y = 0.97,
```

```
                        x = 0.4,
                        width = 0.1
                        )

###

exams::tex2image(tex = "$\\overrightarrow{AB}$",
                 format = "svg",
                 density = 1000,
                 resize = 1000,
                 dir = paste0(getwd(), "/img"),
                 name = "vector_ab",
                 show = FALSE,
                 header = header_ok)

p_3 <- grImport2::readPicture("img/vector_ab.svg" )


grImport2::grid.picture(p_3,
                        y = (11+3.8)/23,
                        x = (11+6)/23,
                        width = 0.1
                        )

###

exams::tex2image(tex = "$\\overrightarrow{CD}$",
                 format = "svg",
                 density = 1000,
                 resize = 1000,
                 dir = paste0(getwd(), "/img"),
                 name = "vector_cd",
                 show = FALSE,
                 header = header_ok)

p_4 <- grImport2::readPicture("img/vector_cd.svg" )


grImport2::grid.picture(p_4,
                        y = (11+8)/23,
                        x = (11+3.5)/23,
                        width = 0.1
```

```
                        )

dev.off()
```



2 Barchart

Barchart hay bar chart là đồ thị cột dùng để biểu diễn giữa 1 biến phân loại (trục X) và 1 biến liên tục (trục Y).

2.1 So sánh hai nhóm

```
df_particle <- readRDS("dataset/df_particle.rds")
```

```
df_particle
```

```
    before after
1    0.291 0.335
2    0.635 0.513
3    0.493 0.469
4    0.480 0.386
5    0.401 0.398
6    0.330 0.323
7    0.398 0.393
8    0.332 0.302
9    0.439 0.434
10   0.409 1.718
11   0.302 0.260
12   0.335 0.288
13   0.576 0.686
14   0.476 0.571
15   0.406 0.810
16   0.435 0.410
17   0.280 0.353
18   0.413 0.390
19   0.291 0.323
20   0.444 0.394
21   0.426 0.449
22   1.138 0.506
```

```
23   0.434 0.456
24   0.286 0.278
25   0.353 0.319
26   0.414 0.539
27   0.476 0.392
28   0.393 0.399
29   0.342 0.348
30   0.589 0.479
31   0.577 0.281
32   0.426 0.415
33   0.421 0.412
34   0.406 0.415
35   0.283 0.447
36   0.486 0.410
37   1.044 0.858
38   0.437 0.420
39   0.474 0.533
40   0.457 0.431
41   0.311 0.288
42   0.397 0.826
43   0.418 0.537
44   0.424 0.396
45   0.291 0.268
46   0.396 0.882
47   0.312 0.285
48   0.413 0.550
49   0.406 0.384
50   0.414 0.493
51   0.460 0.446
52   0.418 0.395
53   0.452 0.422
54   0.403 1.048
55   0.412 0.947
56   0.418 0.421
57   0.506 0.404
58   0.424 0.428
59   0.404 0.830
60   0.442 0.598
61   0.562 0.473
62   0.433 0.444
63   0.440 0.399
```

```
64   0.504 0.420
65   0.275 0.312
66   0.399 0.393
67   0.560 0.391
68   0.279 0.350
69   0.529 0.402
70   0.277 0.268
71   0.412 0.474
72   0.319 0.327
73   0.403 0.386
74   0.288 0.296
75   0.280 0.265
76   0.309 0.331
77   0.459 0.390
78   0.514 0.586
79   0.321 0.283
80   0.359 0.885
81   1.152 0.568
82   0.491 0.571
83   0.285 0.323
84   0.706 0.462
85   0.287 0.301
86   0.432 0.400
87   0.441 0.387
88   0.388 0.401
89   0.553 0.395
90   0.296 0.266
91   0.284 0.308
92   0.536 0.380
93   0.409 0.840
94   0.420 0.426
95   0.284 0.318
96   0.587 0.471
97   0.652 0.540
98   0.401 0.987
99   0.649 0.467
100  0.423 0.491
```

```
library(dplyr)
# rã từ wide về true long
df_particle %>% tidyr::gather(before,
                              after,
```

```
                                key = "group",
                                value = "value") -> df_particle_long
```

```
df_particle_long
```

```
     group value
1   before 0.291
2   before 0.635
3   before 0.493
4   before 0.480
5   before 0.401
6   before 0.330
7   before 0.398
8   before 0.332
9   before 0.439
10  before 0.409
11  before 0.302
12  before 0.335
13  before 0.576
14  before 0.476
15  before 0.406
16  before 0.435
17  before 0.280
18  before 0.413
19  before 0.291
20  before 0.444
21  before 0.426
22  before 1.138
23  before 0.434
24  before 0.286
25  before 0.353
26  before 0.414
27  before 0.476
28  before 0.393
29  before 0.342
30  before 0.589
31  before 0.577
32  before 0.426
33  before 0.421
34  before 0.406
35  before 0.283
36  before 0.486
```

```
37  before 1.044
38  before 0.437
39  before 0.474
40  before 0.457
41  before 0.311
42  before 0.397
43  before 0.418
44  before 0.424
45  before 0.291
46  before 0.396
47  before 0.312
48  before 0.413
49  before 0.406
50  before 0.414
51  before 0.460
52  before 0.418
53  before 0.452
54  before 0.403
55  before 0.412
56  before 0.418
57  before 0.506
58  before 0.424
59  before 0.404
60  before 0.442
61  before 0.562
62  before 0.433
63  before 0.440
64  before 0.504
65  before 0.275
66  before 0.399
67  before 0.560
68  before 0.279
69  before 0.529
70  before 0.277
71  before 0.412
72  before 0.319
73  before 0.403
74  before 0.288
75  before 0.280
76  before 0.309
77  before 0.459
```

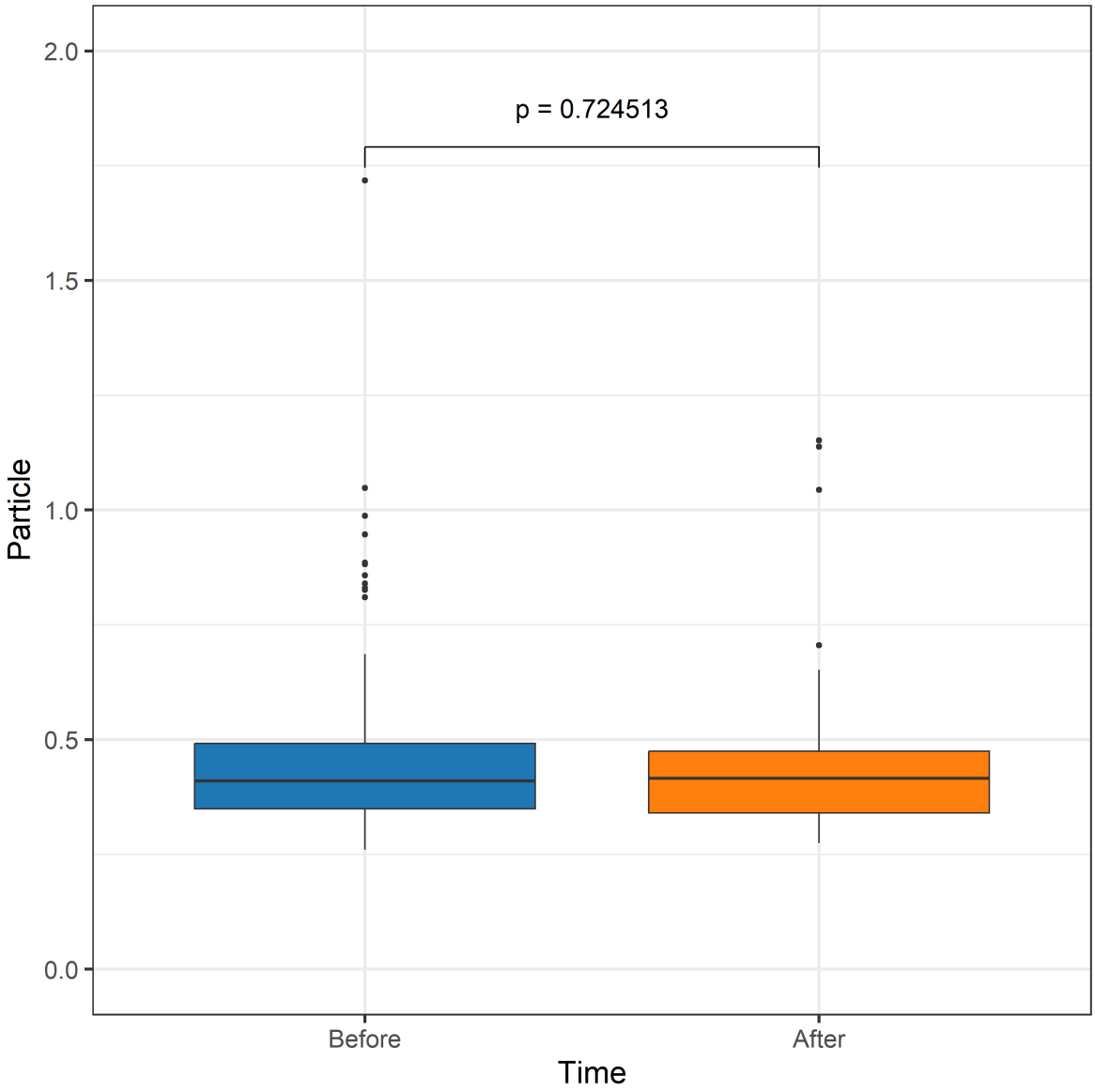
```
78  before 0.514
79  before 0.321
80  before 0.359
81  before 1.152
82  before 0.491
83  before 0.285
84  before 0.706
85  before 0.287
86  before 0.432
87  before 0.441
88  before 0.388
89  before 0.553
90  before 0.296
91  before 0.284
92  before 0.536
93  before 0.409
94  before 0.420
95  before 0.284
96  before 0.587
97  before 0.652
98  before 0.401
99  before 0.649
100 before 0.423
101  after 0.335
102  after 0.513
103  after 0.469
104  after 0.386
105  after 0.398
106  after 0.323
107  after 0.393
108  after 0.302
109  after 0.434
110  after 1.718
111  after 0.260
112  after 0.288
113  after 0.686
114  after 0.571
115  after 0.810
116  after 0.410
117  after 0.353
118  after 0.390
```

```
119  after 0.323
120  after 0.394
121  after 0.449
122  after 0.506
123  after 0.456
124  after 0.278
125  after 0.319
126  after 0.539
127  after 0.392
128  after 0.399
129  after 0.348
130  after 0.479
131  after 0.281
132  after 0.415
133  after 0.412
134  after 0.415
135  after 0.447
136  after 0.410
137  after 0.858
138  after 0.420
139  after 0.533
140  after 0.431
141  after 0.288
142  after 0.826
143  after 0.537
144  after 0.396
145  after 0.268
146  after 0.882
147  after 0.285
148  after 0.550
149  after 0.384
150  after 0.493
151  after 0.446
152  after 0.395
153  after 0.422
154  after 1.048
155  after 0.947
156  after 0.421
157  after 0.404
158  after 0.428
159  after 0.830
```

```
160  after 0.598
161  after 0.473
162  after 0.444
163  after 0.399
164  after 0.420
165  after 0.312
166  after 0.393
167  after 0.391
168  after 0.350
169  after 0.402
170  after 0.268
171  after 0.474
172  after 0.327
173  after 0.386
174  after 0.296
175  after 0.265
176  after 0.331
177  after 0.390
178  after 0.586
179  after 0.283
180  after 0.885
181  after 0.568
182  after 0.571
183  after 0.323
184  after 0.462
185  after 0.301
186  after 0.400
187  after 0.387
188  after 0.401
189  after 0.395
190  after 0.266
191  after 0.308
192  after 0.380
193  after 0.840
194  after 0.426
195  after 0.318
196  after 0.471
197  after 0.540
198  after 0.987
199  after 0.467
200  after 0.491
```

```r
png(filename = "img/p1.png",
    width = 10,
    height = 10,
    res = 300,
    units = "in")

library(ggplot2)
library(ggsignif)
library(ggsci)

ggplot(data = df_particle_long,
       mapping = aes(x = group,
                     y = value,
                     fill = group)) +

  geom_boxplot(show.legend = FALSE) +

  scale_x_discrete(labels = c("Before",
                              "After")) +

  scale_y_continuous(limits = c(0, 2)) +

   ggsignif:::geom_signif(comparisons = list(unique(df_particle_long$group)),
                          map_signif_level = function(p_ok) {
                            # ""
                          paste0("p", " = ", round(p_ok, 6))
                            },
                          textsize = 6,
                          vjust = -1,
                          test = "wilcox.test",
                          test.args = list(paired = TRUE)
                          ) +

  ggsci:::scale_fill_d3() +

  labs(x = "Time",
       y = "Particle") +

  theme_bw(base_size = 20) -> p1

p1
```

```
dev.off()
```



Để cho chữ p in nghiêng, thì ta tự vẽ lại riêng đoạn thẳng và ký hiệu vì rất khó modify trong function `ggsignif:::geom_signif()`.

Tham khảo thêm cách so sánh hai group trong R[1].

[1] https://cran.r-project.org/web/packages/ggprism/vignettes/pvalues.html

3 Data wrangling

Data wrangling là sắp xếp dữ liệu.

3.1 Chọn ngẫu nhiên số dòng trong dataset

```
df_particle <- readRDS("dataset/df_particle.rds")
```

```
df_particle
```

```
    before after
1    0.291 0.335
2    0.635 0.513
3    0.493 0.469
4    0.480 0.386
5    0.401 0.398
6    0.330 0.323
7    0.398 0.393
8    0.332 0.302
9    0.439 0.434
10   0.409 1.718
11   0.302 0.260
12   0.335 0.288
13   0.576 0.686
14   0.476 0.571
15   0.406 0.810
16   0.435 0.410
17   0.280 0.353
18   0.413 0.390
19   0.291 0.323
20   0.444 0.394
21   0.426 0.449
22   1.138 0.506
```

```
23   0.434 0.456
24   0.286 0.278
25   0.353 0.319
26   0.414 0.539
27   0.476 0.392
28   0.393 0.399
29   0.342 0.348
30   0.589 0.479
31   0.577 0.281
32   0.426 0.415
33   0.421 0.412
34   0.406 0.415
35   0.283 0.447
36   0.486 0.410
37   1.044 0.858
38   0.437 0.420
39   0.474 0.533
40   0.457 0.431
41   0.311 0.288
42   0.397 0.826
43   0.418 0.537
44   0.424 0.396
45   0.291 0.268
46   0.396 0.882
47   0.312 0.285
48   0.413 0.550
49   0.406 0.384
50   0.414 0.493
51   0.460 0.446
52   0.418 0.395
53   0.452 0.422
54   0.403 1.048
55   0.412 0.947
56   0.418 0.421
57   0.506 0.404
58   0.424 0.428
59   0.404 0.830
60   0.442 0.598
61   0.562 0.473
62   0.433 0.444
63   0.440 0.399
64   0.504 0.420
65   0.275 0.312
```

```
66   0.399 0.393
67   0.560 0.391
68   0.279 0.350
69   0.529 0.402
70   0.277 0.268
71   0.412 0.474
72   0.319 0.327
73   0.403 0.386
74   0.288 0.296
75   0.280 0.265
76   0.309 0.331
77   0.459 0.390
78   0.514 0.586
79   0.321 0.283
80   0.359 0.885
81   1.152 0.568
82   0.491 0.571
83   0.285 0.323
84   0.706 0.462
85   0.287 0.301
86   0.432 0.400
87   0.441 0.387
88   0.388 0.401
89   0.553 0.395
90   0.296 0.266
91   0.284 0.308
92   0.536 0.380
93   0.409 0.840
94   0.420 0.426
95   0.284 0.318
96   0.587 0.471
97   0.652 0.540
98   0.401 0.987
99   0.649 0.467
100  0.423 0.491
```

Cách 1[1]

```
library(dplyr)

# cố định sự ngẫu nhiên
```

[1] https://scales.arabpsychology.com/stats/how-to-select-random-rows-in-r-using-dplyr/

```r
set.seed(1)

# theo số lượng dòng
df_particle %>% dplyr:::sample_n(size = 20,
                                 replace = FALSE)
```

```
   before after
1   0.279 0.350
2   0.474 0.533
3   0.291 0.335
4   0.406 0.415
5   0.441 0.387
6   0.418 0.537
7   0.476 0.571
8   0.491 0.571
9   0.404 0.830
10  0.460 0.446
11  0.287 0.301
12  0.426 0.449
13  0.403 1.048
14  0.288 0.296
15  0.398 0.393
16  0.403 0.386
17  0.321 0.283
18  1.044 0.858
19  0.285 0.323
20  0.652 0.540
```

```r
# cố định sự ngẫu nhiên
set.seed(1)

# theo tỷ lệ
df_particle %>% dplyr:::sample_frac(size = 0.25,
                                    replace = FALSE)
```

```
   before after
1   0.279 0.350
2   0.474 0.533
3   0.291 0.335
4   0.406 0.415
5   0.441 0.387
```

```
6   0.418 0.537
7   0.476 0.571
8   0.491 0.571
9   0.404 0.830
10  0.460 0.446
11  0.287 0.301
12  0.426 0.449
13  0.403 1.048
14  0.288 0.296
15  0.398 0.393
16  0.403 0.386
17  0.321 0.283
18  1.044 0.858
19  0.285 0.323
20  0.652 0.540
21  0.424 0.396
22  0.706 0.462
23  0.421 0.412
24  0.283 0.447
25  0.277 0.268
```

4 Format

In đậm, gạch dưới, tô màu

- Cách 1 (kiểu Quarto)

abc

```
[**abc**]{.underline style="color:#FF0000;"}
```

- Cách 2 (kiểu HTML)

abc

```
[<ins>**abc**</ins>]{style="color:#FF0000;"}
```

Highlight màu vàng

abc

```
<mark style="background-color: #FFFF00">**abc**</mark>
```

Chèn ảnh và thay đổi kích thước

```
![](logor.png){ width=200px height=155 }
```

Ẩn/hiện kết quả[1]

[1] https://minidown.atusy.net/?framework=sakura&theme=default#results-folding

Tài liệu tham khảo

[1] Math Insight, "An introduction to vectors." Available: https://mathinsight.org/vector_introduction