TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA HỆ THỐNG THÔNG TIN

TÀI LIỆU HƯỚNG DẪN THỰC HÀNH
Môn: Khai thác dữ liệu

BÀI THỰC HÀNH SỐ 1

I. Mục tiêu

1. Nhận diện được các các loại dữ liệu sử dụng cho bài toán khai thác dữ liệu.
2. Nắm bắt được những nội dung cơ bản về dữ liệu thông qua các kỹ thuật thống kê đơn giản.
3. Vận dụng các phương pháp tiền xử lí dữ liệu trong các trường hợp: dữ liệu bị thiếu, nhiễu, dữ liệu đặc biệt... Tìm hiểu một số kĩ thuật tiền xử lý dữ liệu đối với các dữ liệu dạng text hoặc dữ liệu theo thời gian.

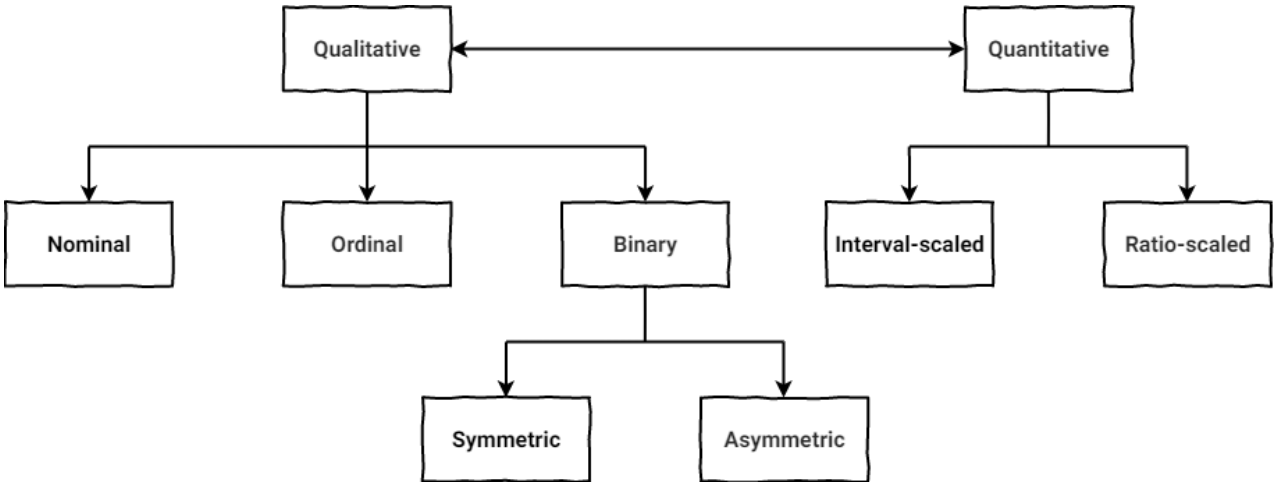
II. Thời gian

1. Thực hành: 4 tiết
2. Bài tập làm thêm: 8 tiết

III. Lý thuyết

1. Phân loại thuộc tính dữ liệu

Có nhiều cách phân loại thuộc tính dữ liệu khác nhau và dưới đây là một cách phổ biến.



Thuộc tính dữ liệu được chia thành hai loại lớn: Qualitative (định tính) và Quantitative (định lượng). Định tính thường là những thuộc tính dữ liệu về định nghĩa, phân loại, dạng thức, kết cấu, cảm nhận, biểu hiện... mà không được đo đạc bằng giá trị số. Ngược lại với định tính, định lượng thường là những giá trị số thu thập được bằng cách đo đạc, tính toán, tổng hợp.

Thuộc tính định tính được chia làm 3 loại: nominal (phân loại), ordinal (xếp hạng) và binary (nhị phân).

- *Nominal*: giá trị của thuộc tính loại phân loại thường là ký hiệu hoặc tên của một đối tượng. Mỗi giá trị đại diện cho một danh mục, mã hoặc trạng thái, chính vì vậy có thể gọi chúng là thể loại. Các giá trị hoàn toàn không có ý nghĩa thứ tự, do đó không thể xác định được giá trị trung bình (mean, average), giá trị trung vị (median, middle). Thay vào đó, người ta quan tâm đến giá trị xuất hiện nhiều nhất, được gọi là mốt (mode), chúng thể hiện xu hướng tập trung của dữ liệu. Ví dụ: thuộc tính *Màu tóc, Tình trạng hôn nhân, Mã khách hàng*.
- *Ordinal*: là thuộc tính có các giá trị có thể sắp xếp thứ tự hoặc mang ý nghĩa xếp hạng, tuy nhiên không thể xác định được độ lớn giữa các giá trị liên tiếp. Ví dụ: thuộc tính *Thông tin giáo dục, Cỡ áo sơmi, Xếp loại hạnh kiểm*. Thuộc tính có thứ tự có thể được tạo ra bằng cách chia giá trị số thành các khoảng và đặt tên cho từng khoảng đó. Đối với thuộc tính loại này, xu hướng tập trung của dữ liệu thể hiện qua mốt và giá trị trung vị (không tính toán được giá trị trung bình).
- *Binary*: là thuộc tính kiểu thể loại nhưng chỉ có 2 giá trị hoặc trạng thái: 0 hoặc 1. Với 0 mang ý nghĩa là có và 1 mang ý nghĩa là không có. Thuộc tính kiểu nhị phân có thể được cài đặt kiểu Boolean nếu nó mang hai trạng thái đúng (true) hoặc sai (false). Ví dụ, thuộc tính *Kiểm tra y tế* là kiểu nhị phân, với 1 mang giá trị dương tính, 0 mang giá trị âm tính. Thuộc tính nhị phân được chia thành 2 loại nhỏ: symmetric (đối xứng) và asymmetric (bất đối xứng).
  - ✓ *Symmetric*: một thuộc tính có kiểu đối xứng khi hai trạng thái có giá trị ngang nhau và mang sức nặng như nhau. Ví dụ: thuộc tính *Giới tính*.
  - ✓ *Asymmetric*: là một thuộc tính nhị phân có độ quan trọng không bằng nhau. Ví dụ: thuộc tính *Kết quả xét nghiệm Corona*.

Thuộc tính định lượng, đôi khi còn được gọi là thuộc tính dạng số. Đây là những thuộc tính biểu diễn đại lượng có thể đo được, các giá trị được hiển thị bằng số nguyên hoặc số thực. Thuộc tính định lượng được chia thành hai loại: interval-scaled (giá trị theo khoảng) và ratio-scaled (giá trị theo tỷ lệ).

- *Interval-scaled:* là giá trị được đo trên thang đơn vị có kích thước bằng nhau. Giá trị của các thuộc tính này có thứ tự và có thể dương, 0 hoặc âm. Do đó, ngoài việc có thể xếp hạng, các thuộc tính như vậy cho phép chúng ta so sánh và định lượng sự khác biệt giữa

các giá trị. Ví dụ: thuộc tính *Ngày, Nhiệt độ* (theo đơn vị ^oC hoặc ^oF). Các giá trị nhiệt độ có thể xếp hạng, so sánh với nhau, tính được sự chênh lệch nhiệt độ. Tuy nhiên, theo thang đo nhiệt độ Celsius và Fahrenheit thì không có giá trị *0 thật sự* có nghĩa là không có nhiệt độ. Do đó, chúng ta không thể nhân hoặc chia nhiệt độ với nhau để tạo ra một con số có ý nghĩa, cũng như không thể khẳng định 30^oC sẽ nóng gấp đôi 15^oC. Tương tự như thế, thuộc tính *Ngày* cũng sẽ không có giá trị 0 thật sự, mà tại đó thời gian bắt đầu. Vì những thuộc tính có kiểu giá trị theo khoảng là số, nên ngoài việc tính mốt, tìm giá trị trung vị thì có thể tính được giá trị trung bình.

- *Ratio-scaled:* đây là những thuộc tính có giá trị 0 thật sự. Đối với những thuộc tính dạng này, các giá trị có thể đem nhân hoặc chia, kết quả thu được hoàn toàn có ý nghĩa. Ngoài ra, chúng cũng có thứ tự, có thể tính độ chênh lệch, tìm mốt, giá trị trung vị và giá trị trung bình. Ví dụ: thuộc tính *Số tiền, Số năm kinh nghiệm, Số từ trong văn bản*.

2. Một số khái niệm trong thống kê

Phần này sẽ giới thiệu sơ lược về các khái niệm cơ bản của thống kê dùng trong mô tả dữ liệu như các tham số đo lường xu hướng tập trung của dữ liệu (mean, median, mode) và đo lường sự biến thiên của dữ liệu (range, variance, standard deviation, standard error). Điều này có nghĩa là người phân tích đang tìm cách trả lời các câu hỏi như 'Một quan sát tiêu biểu của dữ liệu có giá trị bao nhiêu?', 'Dữ liệu có giá trị nào tương đồng nhau không, nếu khác nhau thì mức độ biến thiên cao hay thấp?'

Để dễ hình dung, hãy bắt đầu với ví dụ đơn giản sau:

Giả sử rằng một người A chạy 100 m trong sáu lần, mỗi lần chạy người đó dùng đồng hồ đo lại thời gian (tính bằng giây) và kết quả 6 lần chạy thu được gồm sáu giá trị (còn gọi là quan sát) như sau:

$$x = \{25.1, 21.2, 17.9, 23.0, 24.6, 19.5\}$$

Hãy tiến hành phân tích xem dữ liệu này cho biết những thông tin gì. Sau đây là một số thống kê đơn giản của dữ liệu về thời gian chạy 100m của người A:

- Thời gian chạy trung bình (mean) là *21.9 giây*
- Giá trị giữa (còn gọi là trung vị - median) là *22.1 giây*
- Thời gian chạy nhiều nhất (maximum) là *25.1 giây* và thời gian chạy ít nhất (minimum) là *17.9 giây*. Nếu so sánh với kỷ lục thế giới về 100m là 9.58 giây thì người A biết rằng mình chạy để tập thể dục cho khỏe chứ không phải là vận động viên điền kinh chuyên nghiệp!

- Phương sai (variance) là *8.2 giây bình phương* và độ lệch chuẩn (standard deviation) là *2.9 giây*

2.1. Đo lường xu hướng tập trung (Central Tendency)

Để đo lường xu hướng tập trung của dữ liệu người ta thường dùng 3 tham số đó là giá trị trung bình (trung bình số học - Arithmetic mean hay average), giá trị trung vị (median) và mốt (mode).

2.1.1. Trung bình (mean)

Trung bình số học được tính đơn giản bằng tổng của tất cả các giá trị của dữ liệu trong mẫu chia cho kích thước mẫu.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Với dữ liệu về chạy 100m trên ta có

$$\bar{x} = \frac{25.1+21.2+17.9+23.0+24.6+19.5}{6} = 21.9 \ (giây)$$

2.1.2. Trung vị (median)

Trong lý thuyết xác suất và thống kê, giá trị trung vị là giá trị giữa trong một phân bố chia phân bố thành 2 nhóm mà trong đó số các số trong mỗi nhóm bằng nhau. Nói cách khác, nếu m là trung vị của một phân bố nào đó thì một nửa phần tử trong phân bố đó có giá trị nhỏ hơn hay bằng m và một nửa còn lại có giá trị bằng hoặc lớn hơn m.

Giá trị trung vị được tính như sau: Sắp xếp dữ liệu và lấy giá trị ở giữa. Nếu số giá trị là một số chẵn thì giá trị trung vị là trung bình của 2 giá trị ở giữa. Với số liệu trên ta có giá trị trung vị là 22.1

$$17.9, 19.5, \underbrace{21.2, 23.0,}\ 24.6, 25.1$$

$$median = \frac{21.2 + 23.0}{2} = 22.1$$

Công thức chung để tính giá trị trung vị là:

$$Q_{50} = \frac{x\left(\left\lceil\frac{n}{2}\right\rceil\right) + x\left(\left\lfloor\frac{n}{2}+1\right\rfloor\right)}{2}$$

Ký hiệu:

$\lfloor p \rfloor$: Số nguyên lớn nhất nhỏ hơn hoặc bằng p (*floor* function).

$\lceil p \rceil$: Số nguyên nhỏ nhất lớn hơn hoặc bằng p (*ceiling* function)

$x(p)$: Trả về giá trị tại vị trí p trong mẫu x sau khi đã sắp xếp x tăng dần.

Trong ví dụ trên ta có n = 6 và

$$\left\lceil \frac{n}{2} \right\rceil = 3, \qquad x(3) = 21.2, \qquad \left\lfloor \frac{n}{2} + 1 \right\rfloor = 4, \qquad x(4) = 23.0$$

2.1.3. Mốt (mode)

Mốt là phần tử có tần suất xuất hiện nhiều nhất trong mẫu. Nếu trong mẫu không có phần tử nào xuất hiện lặp lại thì không có mốt.

Với mẫu dữ liệu trên thì không có mốt.

2.1.4. So sánh giữa giá trị trung bình, trung vị và mốt

Trong 3 tham số trung bình, trung vị và mốt thì *trung vị* có khả năng đo lường xu hướng tập trung của dữ liệu mạnh nhất.

Trở lại ví dụ chạy 100 m trên, giả sử sau khi chạy hết 6 lần, người A chạy tiếp lần thứ 7. Lần này đột nhiên chân người A bị đau nên người đó đi bộ thay vì chạy và kết quả thời gian của lần này là 79.9 giây. Người A cố gắng thử thêm lần nữa và kết quả vẫn 79.9 giây. Bây giờ, có mẫu về 8 lần chạy như sau:

$$x = \{25.1, 21.2, 17.9, 23.0, 24.6, 19.5, 79.9, 79.9\}$$

Các giá trị trung bình, trung vị và mốt so sánh giữa 2 mẫu như sau:

Xu hướng tập trung	6 lần đo	8 lần đo
Trung bình	21.9 giây	36.4 giây
Trung vị	22.1 giây	23.8 giây
Mốt	Không có	79.9 giây

Nếu quan sát cẩn thận, đối với 6 lần chạy đầu tiên thì thành tích đó chính là thời gian chạy bình thường, còn 2 lần sau có sự khác biệt rất lớn so với 6 lần chạy ban đầu (2 giá trị này được xem là bất thường của dữ liệu – giá trị ngoại lai, outlier) thực chất nó không phải thời gian chạy mà là thời gian đi bộ. Nếu người A không bị đau thì thời gian chạy dao động quanh giá trị trung vị. Theo bảng trên có thể thấy rằng 2 giá trị ngoại lai không ảnh hưởng nhiều đến trung vị (từ 22.1 lên 23.8) nhưng ảnh hưởng rất lớn đến trung bình (từ 21.9 lên 36.4) và mốt. Mặc dù giá trị trung vị có khả năng đo lường xu hướng tập trung của dữ liệu mạnh hơn trung bình vì trung vị không bị ảnh hưởng bởi các giá trị ngoại lai nhưng nhiều người vẫn thích sử dụng giá trị trung bình để đo lường xu hướng tập trung của dữ liệu vì dễ tính hơn không cần phải sắp xếp dữ liệu như trung vị.

Mốt rất hữu ích đối với dữ liệu có kiểu dữ liệu phân loại (nominal). Đối với các dữ liệu có kiểu phân loại ta không thể dùng giá trị trung bình hay trung vị vì không thể tính được hoặc

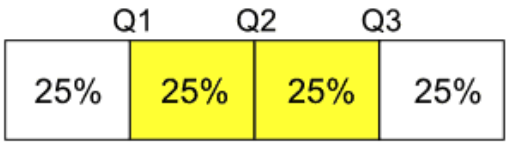
không có ý nghĩa mà phải dùng mốt. Ví dụ, cho dữ liệu mô tả giới tính thuộc kiểu phân loại với giá trị 1 là nam, 0 là nữ thì trung bình hay trung vị là 0.5 không có ý nghĩa gì. Trong khi đó mốt sẽ cho biết tần suất nam hay nữ xuất hiện nhiều nhất.

2.2. Quartiles (tứ phân vị)

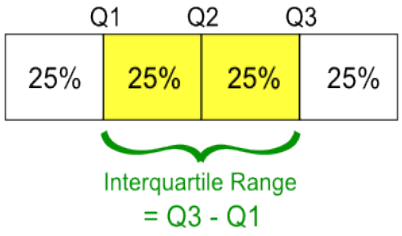
Tứ phân vị là đại lượng mô tả sự phân bố và sự phân tán của tập dữ liệu. Tứ phân vị có 3 giá trị, đó là tứ phân vị thứ nhất (Q1), thứ nhì (Q2), và thứ ba (Q3). Ba giá trị này chia một tập hợp dữ liệu (đã sắp xếp dữ liệu theo trật từ từ bé đến lớn) thành 4 phần có số lượng quan sát đều nhau.

Tứ phân vị được xác định như sau:

- Sắp xếp các số theo thứ tự tăng dần
- Cắt dãy số thành 4 phần bằng nhau
- Tứ phân vị là các giá trị tại vị trí cắt



Khoảng tứ phân vị (Interquartile Range - IQR) được xác định như sau:



Nếu $\frac{i(n+1)}{4}$ là số nguyên thì

$$Q_i = x\left(\frac{i(n+1)}{4}\right)$$

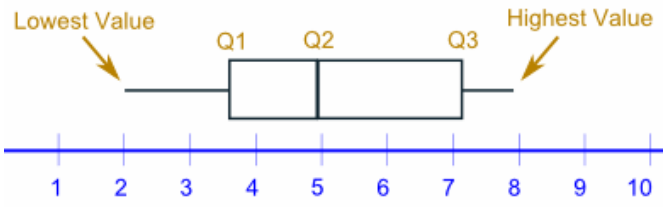
Nếu $\frac{i(n+1)}{4}$ không phải là số nguyên thì

$$Q_i = \frac{x\left(\left\lceil\frac{i(n+1)}{4}\right\rceil\right) + x\left(\left\lfloor\frac{i(n+1)}{4}\right\rfloor\right)}{2}$$

2.3. Biểu đồ hộp (box plot)

Biểu đồ hộp hay đôi khi gọi là biểu đồ hộp và râu (đồ thị hộp ria mèo, box-and-whisker plot) giúp biểu diễn các đại lượng quan trọng của dãy số như giá trị nhỏ nhất (*min*), lớn

nhất (*max*), giá trị phân vị (*quartile*) và khoảng tứ phân vị (*interquartile range*) một cách trực quan, dễ hiểu. Một biểu đồ hộp có dạng như sau:



Sau đây là một số nguyên tắc chung để vẽ một biểu đồ hộp:

- Vẽ một trục ngang thể hiện giá trị của dữ liệu.
- Trên trục này, vẽ một hình chữ nhật với cạnh phía trái hộp là phân vị thứ nhất Q1 và cạnh phía phải hộp là phân vị thứ ba Q3.
- Vẽ một đường thẳng đứng nối cạnh trên và cạnh dưới của hộp tại điểm trung vị m.
- Để vẽ râu bên trái, ta vẽ một đường nằm ngang từ giá trị nhỏ nhất đến điểm giữa cạnh bên trái của hộp.
- Để vẽ râu bên phải, ta vẽ một đường ngang nối từ điểm giữa cạnh phải của hộp đến giá trị lớn nhất của hộp.
- Sau khi vẽ như vậy, ta có biểu đồ hộp chia dữ liệu một cách hình ảnh thành 4 phần. Chú ý rằng, đường ngang chiều dài hộp là khoảng tứ phân vị IQR, râu bên trái thể hiện phần tư thứ nhất, và râu bên phải thể hiện phần tư cuối cùng của dữ liệu.

Biểu đồ hộp thường dùng để biểu diễn một tập giá trị hoặc so sánh giữa các tập giá trị với nhau (một trục có kiểu dữ liệu là nomial, trục còn lại có kiểu numeric).

Ngoài ra, trong biểu đồ này còn có thể vẽ thêm các giá trị ngoại lai ngoài giới hạn: giới hạn dưới (lower fence) và giới hạn trên (upper fence).

$$lower\ fence = Q_1 - 1.5 \times IQR$$

$$upper\ fence = Q_3 + 1.5 \times IQR$$

2.4. Đo lường sự biến thiên của dữ liệu (Variation of Data)

Để biết xu hướng tập trung của dữ liệu ta dùng các giá trị như trung bình, trung vị, mốt. Tuy nhiên, một câu hỏi quan trọng nữa cần phải trả lời khi xem xét chất lượng của mẫu là 'Làm sao đo lường sự biến thiên (hay sự phân tán) của dữ liệu trong mẫu?' Vì có thể 2 mẫu có cùng trung bình nhưng sự biến thiên của dữ liệu là khác nhau.

Để đo lường sự biến thiên (thường so với giá trị trung bình) của dữ liệu người ta thường dùng các tham số khoảng biến thiên (range), khoảng tứ phân vị (IQR), phương sai (variance), độ lệch chuẩn (standard deviation), sai số chuẩn (standard error).

2.4.1. Khoảng biến thiên (range)

Được tính bằng cách lấy giá trị lớn nhất – giá trị nhỏ nhất

$$range = max - min$$

Trong mẫu gồm 6 quan sát về thời gian chạy 100 m trong ví dụ trên ta có

$$range = 25.1 - 17.9 = 7.2 \text{ giây}$$

2.4.2. Độ lệch (deviation)

Cả 2 tham số khoảng biến thiên và khoảng tứ phân vị không quan tâm đến giá trị trung tâm (thường sử dụng giá trị trung bình). Khi muốn đo lường sự phân tán của dữ liệu so với giá trị trung tâm, người ta sẽ đo lường độ lệch của mỗi quan sát (phần tử) so với giá trị trung tâm của mẫu. Giả sử ta sử dụng giá trị trung bình làm giá trị trung tâm, khi đó tổng độ lệch của tất cả quan sát với giá trị trung bình sẽ là:

$$d = \sum_{i=1}^{n} (x_i - \bar{x}) = 0$$

Vì tổng độ lệch này bằng 0 nên không thể dùng độ lệch này để mô tả sự phân tán của dữ liệu.

(Đặc trưng của số trung bình toán học (mean) là san bằng mọi bù trừ. Vì vậy khi tính tổng tất cả các độ lệch thì kết quả luôn bằng 0)

Để khắc phục vấn đề này, người ta sử dụng tổng các giá trị tuyệt đối các độ lệch

$$\delta = \sum_{i=1}^{n} |x_i - \bar{x}|$$

Để loại bỏ ảnh hưởng của kích thước mẫu (vì mỗi mẫu có kích thước khác nhau) người ta chia tổng này cho kích thước mẫu, có được công thức sau:

$$\bar{\delta} = \frac{1}{n} \sum_{i=1}^{n} |x_i - \bar{x}|$$

Tuy nhiên vấn đề của trị tuyệt đối là tính không liên tục (discontinuity) tại gốc tọa độ (trong trường hợp này là mean) vì vậy các nhà thống kê đã tìm ra công thức tốt hơn để mô tả sự biến thiên của dữ liệu đó là phương sai (variance) và độ lệch chuẩn (standard deviation).

2.4.3. Phương sai và độ lệch chuẩn

Để tránh tổng các độ lệch bằng 0 và loại bỏ ảnh hưởng của kích thước mẫu người ta tính tổng bình phương các độ lệch và chia cho kích thước mẫu trừ 1 (hiệu chỉnh). Ta có kết quả là "*trung bình tổng bình phương các độ lệch"* và gọi là phương sai mẫu (sample variance)

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

Phương sai là tham số rất tốt để đo lường sự biến thiên (hay phân tán) của dữ liệu trong mẫu vì nó đã quan tâm đến độ lệch của mỗi quan sát so với giá trị trung tâm, loại bỏ ảnh hưởng của kích thước mẫu và là một hàm trơn (smooth function). Tuy nhiên, điểm yếu của phương sai là không cùng đơn vị tính với giá trị trung bình. Đơn vị tính của phương sai là bình phương đơn vị tính của giá trị trung bình. Chẳng hạn, đơn vị tính của thời gian chạy trung bình là giây trong khi đó đơn vị tính của phương sai là giây bình phương. Để giải quyết vấn đề này, người ta lấy căn bậc 2 của phương sai và kết quả này gọi là độ lệch chuẩn (standard deviation)

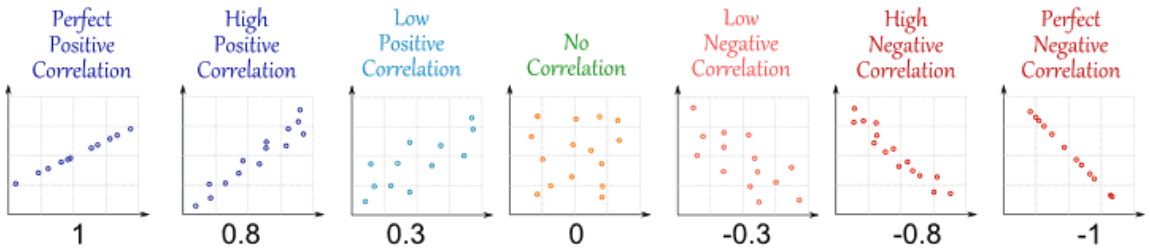
$$s = \sqrt{s^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

2.5. Tương quan (Correlation)

Trong lý thuyết xác suất và thống kê, hệ số tương quan (coefficient correlation) cho biết độ mạnh của mối quan hệ tuyến tính giữa hai biến số ngẫu nhiên. Từ tương quan (correlation) được thành lập từ *co-* (có nghĩa là cùng nhau) và *relation* (quan hệ).

Hệ số tương quan giữa 2 biến có thể dương (positive) hoặc âm (negative). Hệ số tương quan dương cho biết rằng giá trị 2 biến tăng cùng nhau còn hệ số tương quan âm thì nếu một biến tăng thì biến kia giảm.

Độ mạnh và hướng tương quan của 2 biến được mô tả như sau:



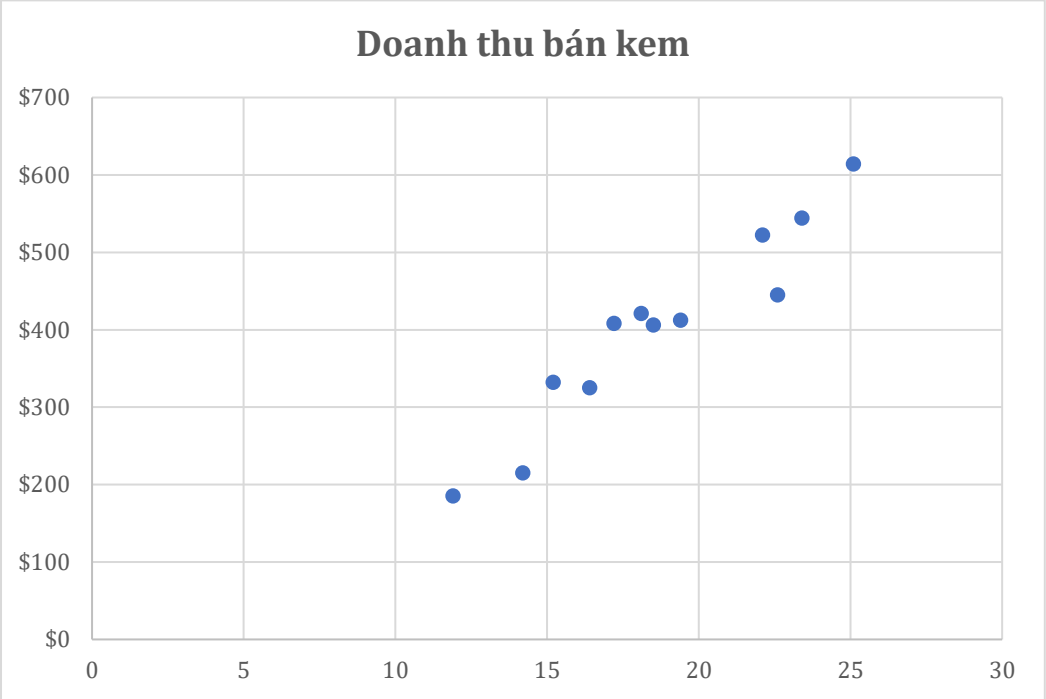
Hệ số tương quan có thể nhận giá trị từ -1 đến 1.

Ví dụ: Có mẫu như sau về nhiệt độ (temperature) và doanh thu bán kem (ice cream sales) như sau:

Nhiệt độ (°C)	Doanh thu bán kem
14.2	$215
16.4	$325
11.9	$185

15.2	$332
18.5	$406
22.1	$522
19.4	$412
25.1	$614
23.4	$544
18.1	$421
22.6	$445
17.2	$408

Đồ thị phân tán (scatter plot) của dữ liệu trên:



Từ đồ thị, có thể thấy rằng nhiệt độ càng cao thì doanh thu bán kem càng cao. Trong dữ liệu trên, hệ số tương quan là **0.9575** (cách tính sẽ được trình bày ở phần sau) cho thấy mối quan hệ giữa nhiệt độ và doanh số bán kem là rất mạnh. Hệ số tương quan dương nói rằng nhiệt độ tăng thì doanh số bán kem cũng tăng.

Tuy nhiên, hãy lưu ý rằng hệ số tương quan không có tính nhân quả (causation).

Trong ví dụ trên, hệ số tương quan là **0.9575**. Bây giờ hãy bắt đầu tìm hiểu cách tính hệ số này theo công thức Pearson (Pearson's Correlation).

Gọi x và y là hai biến (Trong ví dụ trên thì x là Temperature và y là Ice Cream Sales)

- Bước 1: Tính trung bình của x và y
- Bước 2: Tính độ lệch của mỗi giá trị của x với trung bình của x (lấy các giá trị của x trừ đi trung bình của x) và gọi là "**a**", làm tương tự như vậy với y và gọi là "**b**"
- Bước 3: Tính: **a × b**, $\mathbf{a^2}$ và $\mathbf{b^2}$ cho mỗi giá trị

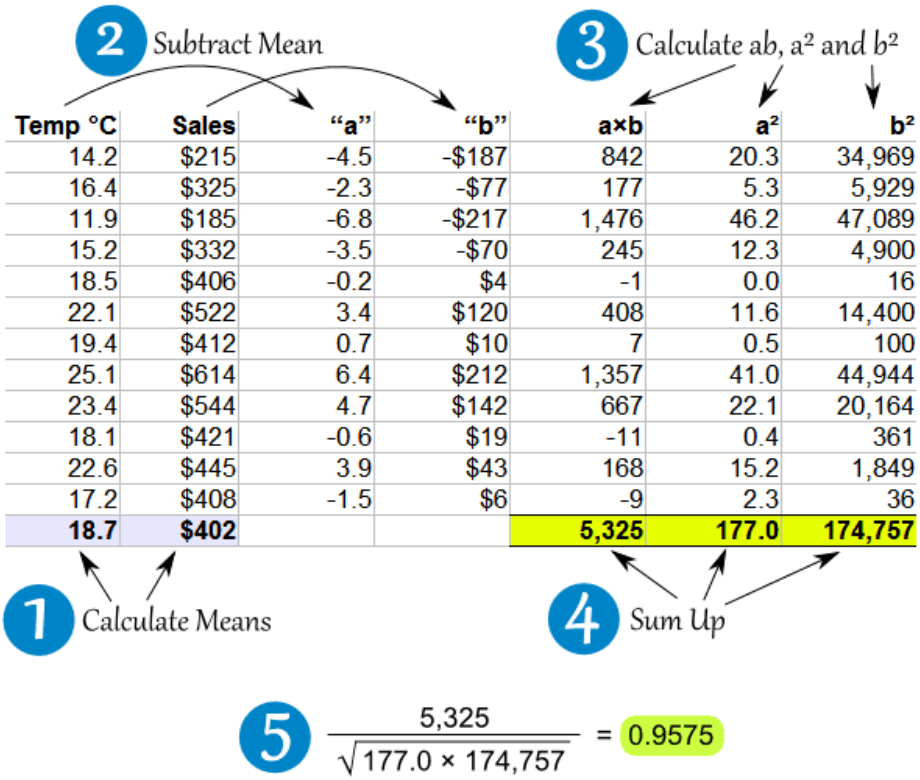
– Bước 4: Tính tổng **a × b**, tổng **a²** và tổng **b²**

– Bước 5: Chia tổng của **a × b** cho căn bậc 2 của [(sum a²) × (sum b²)]

Công thức chung để tính hệ số tương quan giữa 2 đại lượng ngẫu nhiên x và y là

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Dưới đây minh họa việc tính hệ số tương quan của ví dụ trên



Temp °C	Sales	"a"	"b"	a×b	a²	b²
14.2	$215	-4.5	-$187	842	20.3	34,969
16.4	$325	-2.3	-$77	177	5.3	5,929
11.9	$185	-6.8	-$217	1,476	46.2	47,089
15.2	$332	-3.5	-$70	245	12.3	4,900
18.5	$406	-0.2	$4	-1	0.0	16
22.1	$522	3.4	$120	408	11.6	14,400
19.4	$412	0.7	$10	7	0.5	100
25.1	$614	6.4	$212	1,357	41.0	44,944
23.4	$544	4.7	$142	667	22.1	20,164
18.1	$421	-0.6	$19	-11	0.4	361
22.6	$445	3.9	$43	168	15.2	1,849
17.2	$408	-1.5	$6	-9	2.3	36
18.7	**$402**			**5,325**	**177.0**	**174,757**

IV. Hướng dẫn chung

Cho dữ liệu về các hành khách trên tàu Titanic[1] và một phần dữ liệu minh họa như sau:

Survival	Pclass	Name	Sex	Age	SibSp	Parch	Fare	Embarked
0	3	Braund, Mr. Owen Harris	male	22	1	0	7.25	S
1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38	1	0	71.2833	C
1	3	Heikkinen, Miss. Laina	female	26	0	0	7.925	S
1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35	1	0	53.1	S
0	3	Allen, Mr. William Henry	male	35	0	0	8.05	S

0	3	Moran, Mr. James	male		0	0	8.4583	Q
0	1	McCarthy, Mr. Timothy J	male	54	0	0	51.8625	S
0	3	Palsson, Master. Gosta Leonard	male	2	3	1	21.075	S
1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27	0	2	11.1333	S
1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14	1	0	30.0708	C
1	3	Sandstrom, Miss. Marguerite Rut	female	4	1	1	16.7	S
1	1	Bonnell, Miss. Elizabeth	female	58	0	0	26.55	S
0	3	Saundercock, Mr. William Henry	male	20	0	0	8.05	S
0	3	Andersson, Mr. Anders Johan	male	39	1	5	31.275	S
0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14	0	0	7.8542	S
1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55	0	0	16	S
0	3	Rice, Master. Eugene	male	2	4	1	29.125	Q
1	2	Williams, Mr. Charles Eugene	male		0	0	13	S
0	3	Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)	female	31	1	0	18	S
1	3	Masselmani, Mrs. Fatima	female		0	0	7.225	C

Mô tả các thuộc tính

- Survival: Hành khách sống sót hay không (1: Yes, 0: No). Survival là thuộc tính quyết định.
- Pclass: Loại vé (1: 1st, 2: 2nd, 3: 3rd).
- Name: Tên.
- Sex: Giới tính.
- Age: Tuổi.

- SibSp: Số vợ hoặc chồng / anh chị em trên tàu.
- Parch: Số lượng cha mẹ / con cái trên tàu.
- Embarked: Cổng lên tàu.

1. Yêu cầu cơ bản

1. Xác định loại của các thuộc tính trong bảng dữ liệu (nominal, categorical, binary...)
2. Xử lý dữ liệu bị thiếu trong bảng dữ liệu
3. Thực hiện khử nhiễu trên thuộc tính tuổi (Age) bằng kĩ thuật Binning và làm trơn (smoothing).
4. Thực hiện rời rạc hóa dữ liệu trên thuộc tính tuổi (Age) thay thế bởi:

- Khoảng giá trị (10 – 20, 0 – 10…)
- Bằng các nhãn khái niệm (youth, senior, adult…)

5. Xét các thuộc tính dạng categorical, nhằm tránh biểu diễn sai giá trị thuộc tính khi sử dụng đối với một số thuật toán khác thác dữ liệu.

   Ví dụ: Cổng lên tàu (Embarked) trong bảng dữ liệu trên có 3 giá trị C, Q, S, nếu ta biểu diễn các thuộc tính này là 1, 2, 3 sẽ sai tính chất vì cách biểu diễn này sẽ chứa quan hệ cấp bậc 3 > 2 > 1.

   Sinh viên hãy tìm một kiểu biểu diễn khác của dạng dữ liệu này để tránh trường hợp trên.
6. Khi sử dụng dữ liệu cho các thuật toán phân lớp hoặc gom cụm (k-NN, neural networks, k-Means... sẽ được học ở các chương sau) để tránh tình trạng các thuộc tính nằm trong vùng giá trị lớn hơn có xu hướng ảnh hưởng đến mô hình nhiều hơn các dữ liệu nằm trong vùng giá trị nhỏ (Ví dụ: Tuổi 20, thu nhập 4.000.000). Ta thực hiện việc chuẩn hóa các thuộc tính về một vùng giá trị. Sinh viên thực hiện chuẩn hóa dữ liệu trên bằng **Min-max normalization.**

Hướng dẫn

1. Sinh viên đọc phần lý thuyết và tự thực hiện yêu cầu này.
2. Hướng dẫn:

- Đối với dữ liệu số:
  - ✓ Bằng giá trị trung bình hoặc trung vị

    Hướng dẫn: Ví dụ thuộc tính tuổi (Age): xử lý dữ liệu thiếu bằng giá trị trung bình trên thuộc tính tuổi.

- Tính giá trị trung bình trên tất cả các giá trị có được của thuộc tính tuổi:

$$\text{trungbinh} = \frac{22 + 38 + 26 + \cdots + 31}{17}$$

- Dùng giá trị trungbinh này để điền vào dữ liệu thiếu và làm tương tự với các thuộc tính dữ liệu số còn lại.

✓ Bằng giá trị trung bình hoặc trung vị của các mẫu dữ liệu thuộc cùng một thuộc tính quyết định.

Hướng dẫn: Tương tự yêu cầu trên nhưng thay vì tính trung bình trên toàn bộ dữ liệu có được của thuộc tính tuổi, ta tính trung bình thuộc tính tuổi thuộc cùng một thuộc tính quyết định, xét trường hợp thuộc tính quyết định survival là 1:

$$\text{trungbinhsongsot} = \frac{38 + 26 + 35 + 27 + 14 + 4 + 58 + 55}{8}$$

- Dùng giá trị này điền vào vị trí dữ liệu còn thiếu ở thuộc tính tuổi có survival là 1.
- Làm tương tự với dữ liệu thiếu ở thuộc tính tuổi có survival là 0.

- Đối với các loại thuộc tính khác (categorical, binary, …) ta có thể có một số cách xử lý như sau:
  - ✓ Loại bỏ mẫu có giá trị thiếu ra khỏi tập dữ liệu.
  - ✓ Điền giá trị thiếu bằng giá trị xuất hiện nhiều nhất trong thuộc tính (mode).
  - ✓ Xem thuộc tính thiếu là một loại mới (đối với categorical)
  - ✓ Sử dụng các thuật toán máy học để dự đoán giá trị của dữ liệu.

3. Hướng dẫn:

- Để binning (chia giỏ) ta thấy giá trị thấp nhất là 2 tuổi, lớn nhất là 58 tuổi nên ta chọn 0 và 60 là giá trị bắt đầu và kết thúc cho mỗi giỏ. Chọn độ rộng mỗi giỏ là 20 ta được các giỏ sau: (0, 20], (20, 40], (40, 60]. Tiếp theo, đem dữ liệu phân vào các giỏ đã chia.
- Để smoothing ta tính giá trị trung bình của các giá trị trong cùng một giỏ, sau đó thay thế giá trị tuổi bằng giá trị trung bình vừa tính theo mỗi giỏ.

4. Sinh viên đọc phần lý thuyết và tự thực hiện yêu cầu này.
5. Hướng dẫn:

Để đảm bảo công bằng cho các thuộc tính định tính, ta có thể biểu diễn thuộc tính này thành dạng One-hot Encoding. Ở dạng biểu diễn này, mỗi giá trị của thuộc tính được

biểu diễn bằng một vector với một thành phần có giá trị là 1 và các thành phần còn lại có giá trị 0. Số lượng thành phần của vector chính là số loại của thuộc tính.

Ví dụ: Xét thuộc tính Level trong một mẫu dữ liệu có các loại sau: Easy, Medium, Hard tương ứng theo thứ tự, ta có biểu diễn One-hot như sau:

Level
Easy
Medium
Hard

One-hot
1,0,0
0,1,0
0,0,1

6. Hướng dẫn:

Đối với Min-max normalization, ta chuyển dữ liệu về khoảng giá trị thuộc vùng 0 – 1 bằng công thức sau:

$$new\_value = \frac{value\ -\ min}{\max - min}$$

Trong đó:

- new_value = giá trị sau khi chuyển đổi.
- value = giá trị đang cần chuyển đổi.
- min = giá trị thấp nhất của thuộc tính.
- max = giá trị lớn nhất của thuộc tính.

Ví dụ: xét giá trị 7.25 của thuộc tính giá vé (Fare) áp dụng công thức trên ta tính được giá trị mới:

$$new\_value = \frac{7.25\ -\ 7.225}{71.2833\ -\ 7.225}$$

2. Yêu cầu lập trình

Sinh viên tải về dữ liệu Titanic đầy đủ ở phần Tài liệu tham khảo và thực hiện các yêu cầu về lập trình trên ngôn ngữ Python, sử dụng công cụ Jupyter Notebook.

1. Nhập dữ liệu và thống kê những thông tin cơ bản của các thuộc tính.
2. Liệt kê danh sách 5 thuộc tính có số lượng giá trị rỗng nhiều nhất.
3. Vẽ biểu đồ khảo sát sự liên quan giữa thuộc tính Độ tuổi (age), Giới tính (sex) và thuộc tính quyết định – khả năng sống sót (survived) và đưa ra nhận xét.

4. Vẽ biểu đồ khảo sát sự liên quan giữa thuộc tính Cổng lên tàu (embarked) và loại vé (pclass) với khả năng sống sót (survived), phân biệt theo Giới tính (sex) và đưa ra nhận xét.
5. Thống kê số người đi cùng (anh chị em/vợ chồng hoặc cha mẹ/con cái) và khảo sát sự liên quan với thuộc tính quyết định và đưa ra nhận xét.
6. Tiền xử lý các thuộc tính thiếu dữ liệu và xóa bỏ các thuộc tính không cần thiết.
7. Biến đổi các thuộc tính sau:

- Chuyển đổi thuộc tính giá vé (fare) về kiểu nguyên.
- Trích xuất thuộc tính danh hiệu (title) từ tên.
- Chuyển đổi thuộc tính giới tính (sex) và cổng lên tàu (embarked) thành số.
- Chia giá trị thuộc tính tuổi (age) thành 8 nhóm, sao cho số phần tử trong nhóm cân bằng. Gợi ý các chia như sau:
  - ✓ Nhóm 1: dưới 11 tuổi
  - ✓ Nhóm 2: trên 11 tuổi đến 18 tuổi
  - ✓ Nhóm 3: trên 18 tuổi đến 22 tuổi
  - ✓ Nhóm 4: trên 22 tuổi đến 27 tuổi
  - ✓ Nhóm 5: trên 27 tuổi đến 33 tuổi
  - ✓ Nhóm 6: trên 33 tuổi đến 40 tuổi
  - ✓ Nhóm 7: trên 40 tuổi đến 66 tuổi
  - ✓ Nhóm 8: trên 66 tuổi
- Chia giá trị thuộc tính giá vé (fare) thành 6 nhóm, yêu cầu như trên. Gợi ý:
  - ✓ Nhóm 1: nhỏ hơn hoặc bằng 7.91
  - ✓ Nhóm 2: lớn hơn 7.91 đến 14.454
  - ✓ Nhóm 3: lớn hơn 14.454 đến 31
  - ✓ Nhóm 4: lớn hơn 31 đến 99
  - ✓ Nhóm 5: lớn hơn 99 đến 250
  - ✓ Nhóm 6: lớn hơn 250

8. Tạo ra các thuộc tính mới:

- Thuộc tính *tuổi * loại vé (age_class)*: là tích của tuổi và loại vé
- Thuộc tính *giá vé trên mỗi người (fare_per_person)*: là giá vé chia cho số người đi cùng người đó (có làm trơn + 1)

Hướng dẫn

1. Tạo mới một tập tin notebook và import vào các thư viện thao tác với tập dữ liệu và số: numpy, pandas, thư viện vẽ đồ thị, trực quan hóa dữ liệu: seaborn, matplotlib.

```
# linear algebra
import numpy as np

# data processing
import pandas as pd

# data visualization
import seaborn as sns
%matplotlib inline
from matplotlib import pyplot as plt
from matplotlib import style
```

Đọc dữ liệu đầu vào

```
with open('titanic.csv') as f:
    df = pd.read_csv(f, dtype={'age': np.float64, 'sibsp': np.int64, 'parch': np.int64})
```

Xem xét dữ liệu ban đầu với hàm head() và info()

```
df.head(12)
```

	pclass	survived	name	sex	age	sibsp	parch	ticket	fare	cabin	embarked	boat	body	home.dest
0	1	1	Allen, Miss. Elisabeth Walton	female	29.0000	0	0	24160	211.3375	B5	S	2	NaN	St Louis, MO
1	1	1	Allison, Master. Hudson Trevor	male	0.9167	1	2	113781	151.5500	C22 C26	S	11	NaN	Montreal, PQ / Chesterville, ON
2	1	0	Allison, Miss. Helen Loraine	female	2.0000	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON
3	1	0	Allison, Mr. Hudson Joshua Creighton	male	30.0000	1	2	113781	151.5500	C22 C26	S	NaN	135.0	Montreal, PQ / Chesterville, ON
4	1	0	Allison, Mrs. Hudson J C (Bessie Waldo Daniels)	female	25.0000	1	2	113781	151.5500	C22 C26	S	NaN	NaN	Montreal, PQ / Chesterville, ON
5	1	1	Anderson, Mr. Harry	male	48.0000	0	0	19952	26.5500	E12	S	3	NaN	New York, NY
6	1	1	Andrews, Miss. Kornelia Theodosia	female	63.0000	1	0	13502	77.9583	D7	S	10	NaN	Hudson, NY
7	1	0	Andrews, Mr. Thomas Jr	male	39.0000	0	0	112050	0.0000	A36	S	NaN	NaN	Belfast, NI
8	1	1	Appleton, Mrs. Edward Dale (Charlotte Lamson)	female	53.0000	2	0	11769	51.4792	C101	S	D	NaN	Bayside, Queens, NY
9	1	0	Artagaveytia, Mr. Ramon	male	71.0000	0	0	PC 17609	49.5042	NaN	C	NaN	22.0	Montevideo, Uruguay
10	1	0	Astor, Col. John Jacob	male	47.0000	1	0	PC 17757	227.5250	C62 C64	C	NaN	124.0	New York, NY
11	1	1	Astor, Mrs. John Jacob (Madeleine Talmadge Force)	female	18.0000	1	0	PC 17757	227.5250	C62 C64	C	4	NaN	New York, NY

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1309 entries, 0 to 1308
Data columns (total 14 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   pclass     1309 non-null   int64
 1   survived   1309 non-null   int64
 2   name       1309 non-null   object
 3   sex        1309 non-null   object
 4   age        1046 non-null   float64
 5   sibsp      1309 non-null   int64
 6   parch      1309 non-null   int64
 7   ticket     1309 non-null   object
 8   fare       1308 non-null   float64
 9   cabin      295 non-null    object
 10  embarked   1307 non-null   object
 11  boat       486 non-null    object
 12  body       121 non-null    float64
 13  home.dest  745 non-null    object
dtypes: float64(3), int64(4), object(7)
memory usage: 143.3+ KB
```

Thống kê chỉ số của các thuộc tính định lượng như: đến số giá trị, giá trị lớn nhất, nhỏ nhất, trung bình, độ lệch chuẩn, tứ phân vị…

```
df.describe()
```

	pclass	survived	age	sibsp	parch	fare	body
count	1309.000000	1309.000000	1046.000000	1309.000000	1309.000000	1308.000000	121.000000
mean	2.294882	0.381971	29.881135	0.498854	0.385027	33.295479	160.809917
std	0.837836	0.486055	14.413500	1.041658	0.865560	51.758668	97.696922
min	1.000000	0.000000	0.166700	0.000000	0.000000	0.000000	1.000000
25%	2.000000	0.000000	21.000000	0.000000	0.000000	7.895800	72.000000
50%	3.000000	0.000000	28.000000	0.000000	0.000000	14.454200	155.000000
75%	3.000000	1.000000	39.000000	1.000000	0.000000	31.275000	256.000000
max	3.000000	1.000000	80.000000	8.000000	9.000000	512.329200	328.000000

2. Lấy ra 5 thuộc tính có số lượng giá trị rỗng nhiều nhất

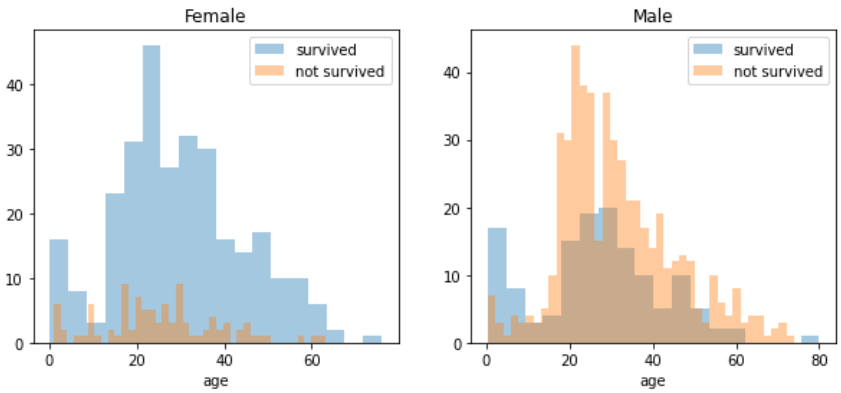
```
total = df.isnull().sum().sort_values(ascending=False)
percent_1 = df.isnull().sum()/df.isnull().count()*100
percent_2 = (round(percent_1, 1)).sort_values(ascending=False)
missing_data = pd.concat([total, percent_2], axis=1, keys=['Total', '%'])
missing_data.head(5)
```

	Total	%
body	1188	90.8
cabin	1014	77.5
boat	823	62.9
home.dest	564	43.1
age	263	20.1

Có thể nhận thấy thuộc tính body hiện đang thiếu dữ liệu rất nghiêm trọng (90.8%), điều này sẽ gây ảnh hưởng lớn nếu sử dụng thuộc tính này trong quá trình khai thác.

3. Xem xét thống kê giữa thuộc tính Độ tuổi (age), Giới tính (sex) và khả năng sống sót (survived)

```
survived = 'survived'
not_survived = 'not survived'
fig, axes = plt.subplots(nrows=1, ncols=2,figsize=(10, 4))
women = df[df['sex']=='female']
men = df[df['sex']=='male']
ax = sns.distplot(women[women['survived']==1].age.dropna(), bins=18, label = survived, ax = axes[0], kde =False)
ax = sns.distplot(women[women['survived']==0].age.dropna(), bins=40, label = not_survived, ax = axes[0], kde =False)
ax.legend()
ax.set_title('Female')
ax = sns.distplot(men[men['survived']==1].age.dropna(), bins=18, label = survived, ax = axes[1], kde = False)
ax = sns.distplot(men[men['survived']==0].age.dropna(), bins=40, label = not_survived, ax = axes[1], kde = False)
ax.legend()
_ = ax.set_title('Male')
```



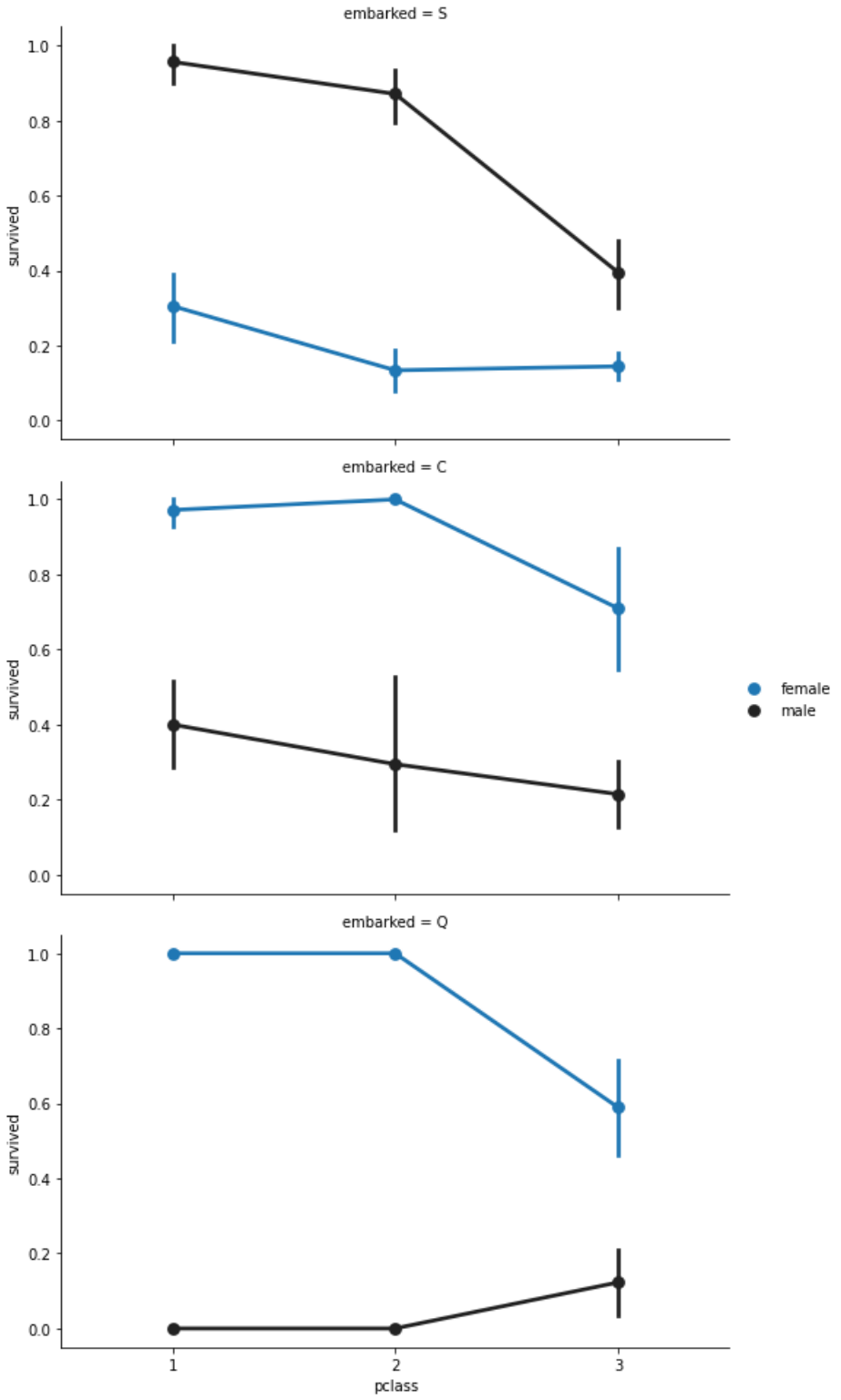
Từ biểu đồ trên có thể rút ra vài nhận xét như sau:

- Xác suất nam sống sót cao ở tuổi từ 1 đến 10
- Cơ hội sống sót của nữ cao hơn từ 12 đến 40 tuổi
- Xác suất nam không sống sót cao nhất ở độ tuổi từ 18 đến 27.
- Nhìn chung tỷ lệ sống sót của nữ ở các lứa tuổi đều có phần nhỉnh hơn nam.

4. Xem xét thống kê giữa thuộc tính Cổng lên tàu (embarked) và loại vé (pclass) với khả năng sống sót (survived), phân biệt theo Giới tính (sex).

```
FacetGrid = sns.FacetGrid(df, row='embarked', height=4.5, aspect=1.6)
FacetGrid.map(sns.pointplot, 'pclass', 'survived', 'sex', palette=None,  order=None, hue_order=None )
FacetGrid.add_legend()
```

```
<seaborn.axisgrid.FacetGrid at 0x1e870d51ee0>
```



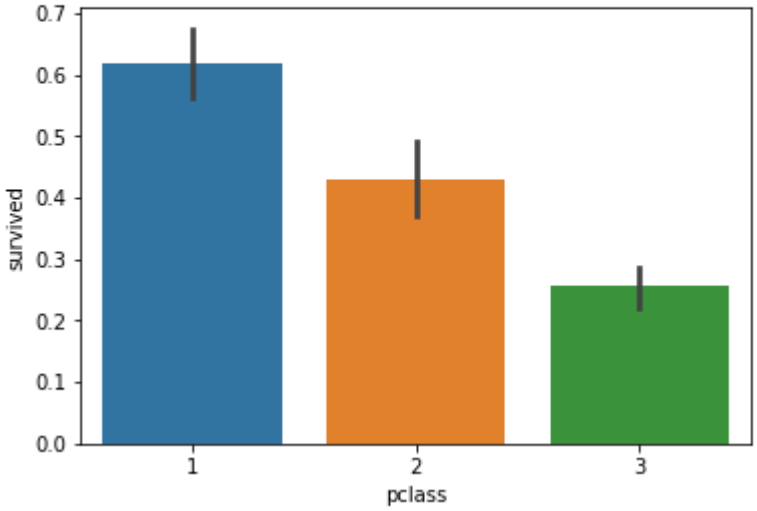
Từ biểu đồ trên có thể thấy được, thuộc tính Cổng lên tàu có thể liên quan đến khả năng sống sót và phụ thuộc vào giới tính

Phụ nữ lên tàu ở cổng Q và cổng S có cơ hội sống cao hơn và ngược lại, lên tàu ở cổng C phụ nữ có tỉ lệ chết cao hơn. Đồng nghĩa với tỉ lệ sống sót của đàn ông lên tàu ở cổng C cao hơn so với cổng Q và S.

Loại vé cũng có thể liên quan đến tỉ lệ sống sót, vì thế, tiếp tục tìm hiểu kỹ hơn về sự phụ thuộc giữa thuộc tính này với thuộc tính quyết định.

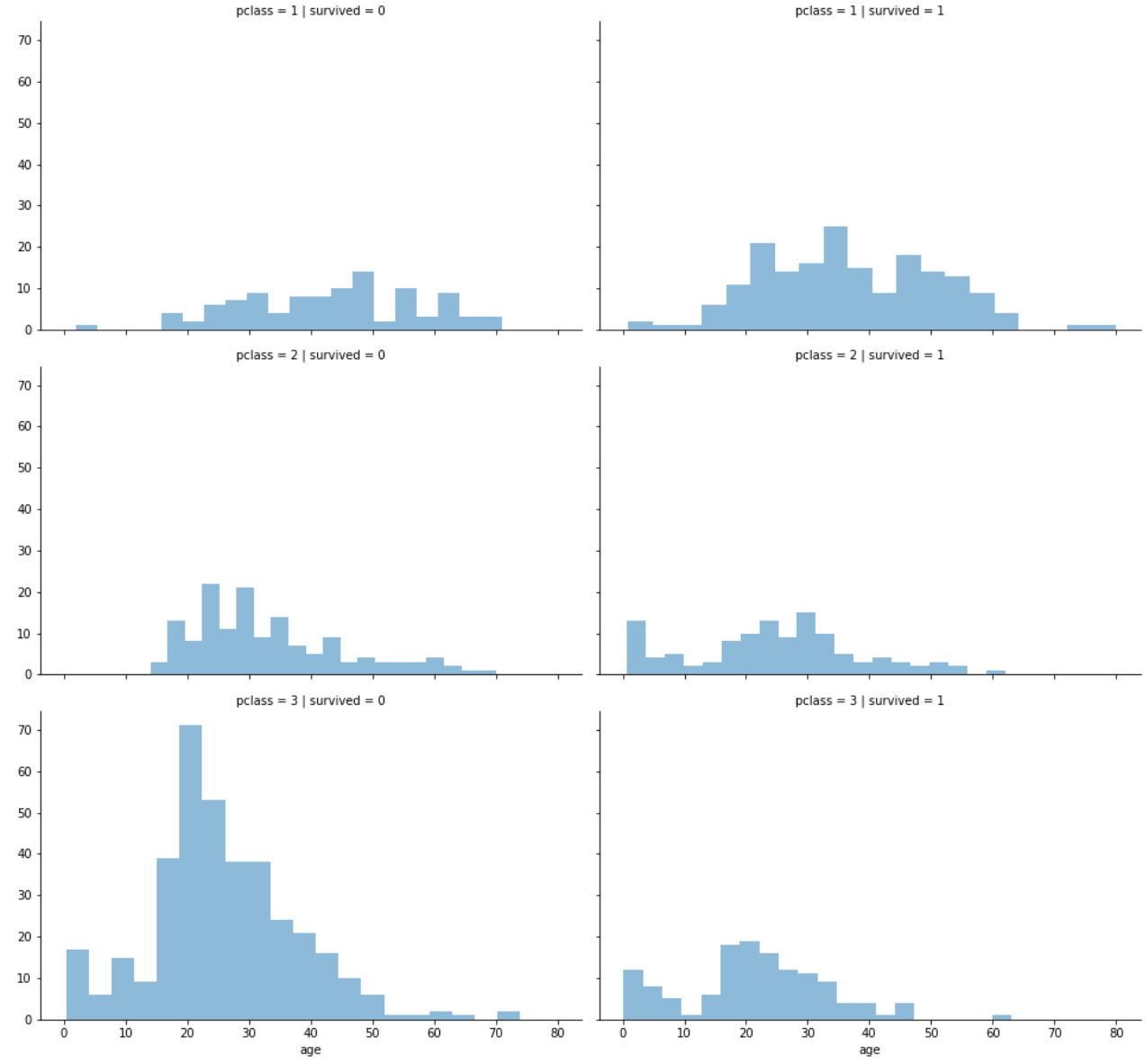
```
sns.barplot(x='pclass', y='survived', data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e870ef4100>
```



Từ biểu đồ này có thể thấy được rõ ràng tỷ lệ sống sót của loại vé 1 là cao nhất.

```
grid = sns.FacetGrid(df, col='survived', row='pclass', height=4.2, aspect=1.6)
grid.map(plt.hist, 'age', alpha=.5, bins=20)
grid.add_legend();
```



Từ đồ thị trên có thể xác nhận nhận định bên trên về loại vé 1, và có thể đưa ra thêm nhận định những người có loại vé 3 thì có tỷ lệ không sống sót lớn.

5. Thống kê số người đi cùng (anh chị em/vợ chồng hoặc cha mẹ/con cái)

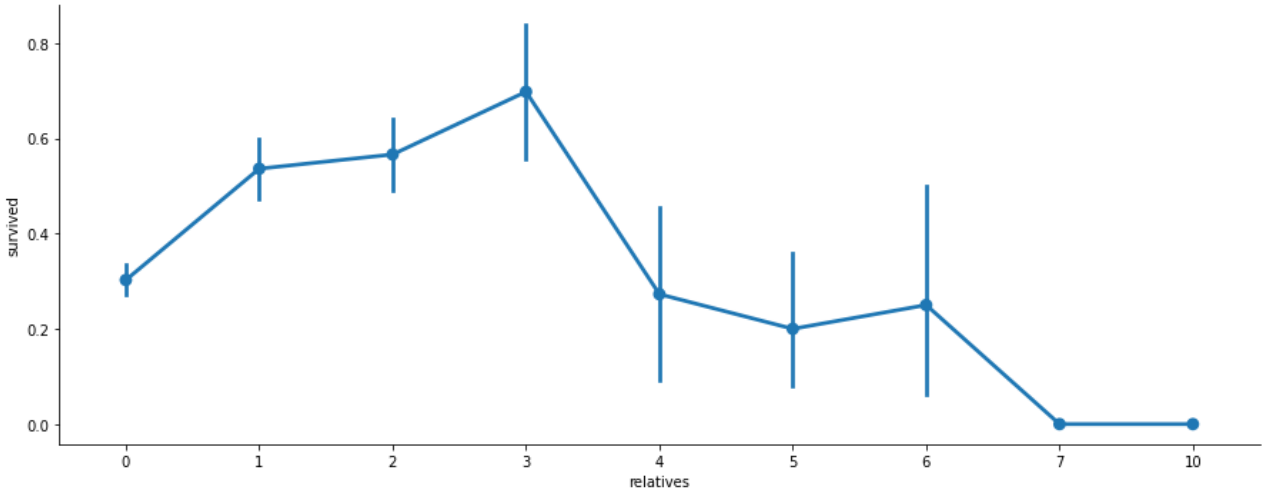
```
for dataset in [df]:
    dataset['relatives'] = dataset['sibsp'] + dataset['parch']
    dataset.loc[dataset['relatives'] > 0, 'not_alone'] = 0
    dataset.loc[dataset['relatives'] == 0, 'not_alone'] = 1
    dataset['not_alone'] = dataset['not_alone'].astype(int)
df['not_alone'].value_counts()
```

```
1    790
0    519
Name: not_alone, dtype: int64
```

- Số người đi một mình là 790 người.
- Số người đi cùng người thân là 519 người.

Vẽ biểu đồ biểu diễn mối quan hệ giữa số người đi cùng và khả năng sống sót

```
axes = sns.catplot('relatives','survived',
                   data=df, aspect = 2.5, kind='point', )
```



Sơ đồ trên cho ta thấy, những người có 1 đến 3 người thân đi cùng có tỉ lệ sống sót cao hơn. Những người có 7 đến 10 người thân đi cùng thì có tỉ lệ sống thấp nhất.

6. Tiền xử lý dữ liệu

Đầu tiên, tiến hành xem xét các thuộc tính thiếu sót nhiều giá trị. Có thể xóa đi các thuộc tính body, cabin, boat, home.dest vì không có ảnh hưởng nhiều đến sự tồn tại của hành khách. Tuy nhiên, mỗi giá trị của thuộc tính cabin ví dụ như ‘C123’ có các ký tự đầu ‘C’ là boong mà hành khách đó đang đi. Dữ liệu này có thể hữu ích nên có thể tách ra, giữ lại trước khi xóa.

```
import re
deck = {"A": 1, "B": 2, "C": 3, "D": 4, "E": 5, "F": 6, "G": 7, "U": 8}

for dataset in [df]:
    dataset['cabin'] = dataset['cabin'].fillna("U0")
    dataset['deck'] = dataset['cabin'].map(lambda x: re.compile("([a-zA-Z]+)").search(x).group())
    dataset['deck'] = dataset['deck'].map(deck)
    dataset['deck'] = dataset['deck'].fillna(0)
    dataset['deck'] = dataset['deck'].astype(int)
# we can now drop the cabin feature
df = df.drop(['cabin'], axis=1)
```

Xem xét thuộc tính ticket

```
df['ticket'].describe()
```

```
count          1309
unique          929
top       CA. 2343
freq              11
Name: ticket, dtype: object
```

Có thể thấy thuộc tính ticket có đến 929 giá trị khác nhau, có khả năng thuộc tính này sẽ không giúp ích được trong quá trình khai thác, ngược lại còn gây sai số cho thuật toán nên có thể xóa nó đi.

Thực hiện xóa các thuộc tính không cần thiết.

```
df = df.drop(['ticket'], axis=1)
df = df.drop(['boat'], axis=1)
df = df.drop(['body'], axis=1)
df = df.drop(['home.dest'], axis=1)
```

Điền giá trị còn thiếu vào thuộc tính tuổi (age).

Các giá trị này sẽ được sinh ngẫu nhiên dựa vào giá trị trung bình, độ lệch chuẩn và số các giá trị còn thiếu.

```
for dataset in [df]:
    mean = df["age"].mean()
    std = df["age"].std()
    is_null = dataset["age"].isnull().sum()
    # compute random numbers between the mean, std and is_null
    rand_age = np.random.randint(mean - std, mean + std, size = is_null)
    # fill NaN values in Age column with random values generated
    age_slice = dataset["age"].copy()
    age_slice[np.isnan(age_slice)] = rand_age
    dataset["age"] = age_slice
    dataset["age"] = df["age"].astype(int)
df["age"].isnull().sum()
```

```
0
```

Điền giá trị còn thiếu vào thuộc tính Cổng lên tàu (embarked).

Xem xét thông tin về thuộc tính này

```
df['embarked'].describe()
```

```
count     1307
unique       3
top          S
freq       914
Name: embarked, dtype: object
```

Ta thấy, giá trị ‘S’ là phổ biến nhất, nên có thể chọn nó đề điền vào các vị trí còn trống.

```
common_value = 'S'

for dataset in [df]:
    dataset['embarked'] = dataset['embarked'].fillna(common_value)
```

7. Xem xét lại các thuộc tính

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1309 entries, 0 to 1308
Data columns (total 16 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   pclass     1309 non-null   int64
 1   survived   1309 non-null   int64
 2   name       1309 non-null   object
 3   sex        1309 non-null   object
 4   age        1309 non-null   int32
 5   sibsp      1309 non-null   int64
 6   parch      1309 non-null   int64
 7   ticket     1309 non-null   object
 8   fare       1308 non-null   float64
 9   embarked   1309 non-null   object
 10  boat       486 non-null    object
 11  body       121 non-null    float64
 12  home.dest  745 non-null    object
 13  relatives  1309 non-null   int64
 14  not_alone  1309 non-null   int32
 15  deck       1309 non-null   int32
dtypes: float64(2), int32(3), int64(5), object(6)
memory usage: 148.4+ KB
```

Có thể thấy, thuộc tính giá vé (fare) có các giá trị nguyên nhưng đang có kiểu dữ liệu là float nên có thể chuyển đổi về kiểu nguyên. Ngoài ra, có thể xử lý các thuộc tính Tên (name), Giới tính (sex), Cổng lên tàu (Embarked).

Đối với thuộc tính giá vé

```
for dataset in [df]:
    dataset['fare'] = dataset['fare'].fillna(0)
    dataset['fare'] = dataset['fare'].astype(int)
```

Đối với thuộc tính tên (name), có thể sẽ trích xuất danh hiệu (title) từ thuộc tính này để xây dựng thuộc tính mới.

```
titles = {"Mr": 1, "Miss": 2, "Mrs": 3, "Master": 4, "Rare": 5}

for dataset in [df]:
    # extract titles
    dataset['title'] = dataset.name.str.extract(' ([A-Za-z]+)\.', expand=False)
    # replace titles with a more common title or as Rare
    dataset['title'] = dataset['title'].replace(['Lady', 'Countess','Capt', 'Col','Don', 'Dr',\
                                            'Major', 'Rev', 'Sir', 'Jonkheer', 'Dona'], 'Rare')
    dataset['title'] = dataset['title'].replace('Mlle', 'Miss')
    dataset['title'] = dataset['title'].replace('Ms', 'Miss')
    dataset['title'] = dataset['title'].replace('Mme', 'Mrs')
    # convert titles into numbers
    dataset['title'] = dataset['title'].map(titles)
    # filling NaN with 0, to get safe
    dataset['title'] = dataset['title'].fillna(0)
df = df.drop(['name'], axis=1)
```

Biểu diễn thuộc tính giới tính (sex) thành các giá trị 0 (nam), 1 (nữ)

```
genders = {"male": 0, "female": 1}

for dataset in [df]:
    dataset['sex'] = dataset['sex'].map(genders)
```

Chuyển đổi thuộc tính cổng lên tàu (embarked) thành số

```
ports = {"S": 0, "C": 1, "Q": 2}

for dataset in [df]:
    dataset['embarked'] = dataset['embarked'].map(ports)
```

Chuyển đổi thuộc tính tuổi (age) về kiểu số nguyên và tạo ra các nhóm tuổi chia giá trị tuổi thành 8 nhóm nhỏ có số lượng hành khách tương đối đều nhau:

- Nhóm 1: dưới 11 tuổi
- Nhóm 2: trên 11 tuổi đến 18 tuổi
- Nhóm 3: trên 18 tuổi đến 22 tuổi
- Nhóm 4: trên 22 tuổi đến 27 tuổi
- Nhóm 5: trên 27 tuổi đến 33 tuổi
- Nhóm 6: trên 33 tuổi đến 40 tuổi
- Nhóm 7: trên 40 tuổi đến 66 tuổi
- Nhóm 8: trên 66 tuổi

```
for dataset in [df]:
    dataset['age'] = dataset['age'].astype(int)
    dataset.loc[ dataset['age'] <= 11, 'age'] = 0
    dataset.loc[(dataset['age'] > 11) & (dataset['age'] <= 18), 'age'] = 1
    dataset.loc[(dataset['age'] > 18) & (dataset['age'] <= 22), 'age'] = 2
    dataset.loc[(dataset['age'] > 22) & (dataset['age'] <= 27), 'age'] = 3
    dataset.loc[(dataset['age'] > 27) & (dataset['age'] <= 33), 'age'] = 4
    dataset.loc[(dataset['age'] > 33) & (dataset['age'] <= 40), 'age'] = 5
    dataset.loc[(dataset['age'] > 40) & (dataset['age'] <= 66), 'age'] = 6
    dataset.loc[ dataset['age'] > 66, 'age'] = 6

# let's see how it's distributed train_df['age'].value_counts()
```

Kiểm tra kết quả đạt được

```
df['age'].value_counts()
```

```
6    255
4    229
3    211
5    203
2    171
1    140
0     91
7      9
Name: age, dtype: int64
```

Đối với thuộc tính giá vé (fare), có thể làm tương tự như đối với thuộc tính tuổi (age). Tuy nhiên, để tránh tình trạng số lượng hành khách tập trung quá nhiều vào một nhóm nào đó, có thể sử dụng hàm qcut() của thư viện sklearn.

```
for dataset in [df]:
    dataset.loc[ dataset['fare'] <= 7.91, 'fare'] = 0
    dataset.loc[(dataset['fare'] > 7.91) & (dataset['fare'] <= 14.454), 'fare'] = 1
    dataset.loc[(dataset['fare'] > 14.454) & (dataset['fare'] <= 31), 'fare']   = 2
    dataset.loc[(dataset['fare'] > 31) & (dataset['fare'] <= 99), 'fare']   = 3
    dataset.loc[(dataset['fare'] > 99) & (dataset['fare'] <= 250), 'fare']   = 4
    dataset.loc[ dataset['fare'] > 250, 'fare'] = 5
    dataset['fare'] = dataset['fare'].astype(int)
```

8. Tạo thuộc tính tuổi * loại vé (age_class)

```
for dataset in [df]:
    dataset['age_class']= dataset['age'] * dataset['pclass']
```

Tạo thuộc tính giá vé trên mỗi người (fare_per_person)

```
for dataset in [df]:
    dataset['fare_per_person'] = dataset['fare']/(dataset['relatives']+1)
    dataset['fare_per_person'] = dataset['fare_per_person'].astype(int)
# Let's take a last look at the training set, before we start training the models.
df.head(10)
```

	pclass	survived	sex	age	sibsp	parch	fare	embarked	relatives	not_alone	deck	title	age_class	fare_per_person
0	1	1	1	4	0	0	4	0	0	1	2	2	4	4
1	1	1	0	0	1	2	4	0	3	0	3	4	0	1
2	1	0	1	0	1	2	4	0	3	0	3	2	0	1
3	1	0	0	4	1	2	4	0	3	0	3	1	4	1
4	1	0	1	3	1	2	4	0	3	0	3	3	3	1
5	1	1	0	6	0	0	2	0	0	1	5	1	6	2
6	1	1	1	6	1	0	3	0	1	0	4	2	6	1
7	1	0	0	5	0	0	0	0	0	1	1	1	5	0
8	1	1	1	6	2	0	3	0	2	0	3	3	6	1
9	1	0	0	6	0	0	3	1	0	1	8	1	6	3

IV. Thực hành

1. Thực hiện lại bài tập ở phần hướng dẫn chung.
2. Để thực hiện chiến dịch marketing cho chương trình gửi tiền có kì hạn, một ngân hàng dựa trên bộ dữ liệu người dùng được cho trong mẫu sau để tìm hiểu khả năng người dùng sẽ đăng kí chương trình gửi tiền có kì hạn ở ngân hàng hay không (mẫu được trích từ tập dữ liệu bank marketing[2]):

Job	Marial	Education	Default	Loan	Duration	Cons.price.idx	Subscribe
housemaid	Married	basic.4y	no	no	261	93.994	no
services	Married	high.school	unknown	no	149	93.994	no
blue-collar	Divorced	basic.4y	unknown	no	1575	93.994	yes
admin.	Single	high.school	no	no	338	93.994	no
blue-collar	Single	basic.9y	no	no	179	93.994	no
services	Married	high.school	no	no	1030	93.994	yes
management	Married	high.school	unknown	no	149		no
unemployed		university.degree	unknown	no	424	93.994	no

technician	Married	professional.course	no	no	1623	93.994	yes
services	Divorced	high.school	unknown	no	568	93.994	no
blue-collar	Married	high.school	no	no	1297	93.994	yes
self-employed	Married	basic.9y	no	no	376	93.918	no
entrepreneur	Married	professional.course	no	yes	576	93.994	no
services	Single	high.school	no	no	1059	93.2	yes
technician	Married	basic.9y	no	no	705	93.2	yes
Retired	Married	high.school	no	no	532	93.2	no

Mô tả thuộc tính

- Job: Nghề nghiệp.
- Marial: Tình trạng hôn nhân.
- Education: Trình độ học vấn.
- Default: Đã có tính dụng mặc định hay chưa.
- Loan: Đã có khoản vay nào chưa.
- Duration: Thời lượng lần contact cuối cùng, tính trên đơn vị giây.
- Cons.price.idx: Chi số giá tiêu dùng.

Yêu cầu

Sinh viên thực hiện khảo sát và tiền xử lý dữ liệu cho dữ liệu Bank marketing giống như phần hướng dẫn.

V. Bài tập thêm

1. Cho mẫu dữ liệu là những bài báo được crawl từ website vnexpress.net được đính kèm trong tập tin. Thực hiện các thao tác tiền xử lý sau:
   a) Loại bỏ các tag html trong file văn bản
   b) Loại bỏ các dấu cách dư thừa trong câu và cắt các từ trong văn bản theo dấu cách
   c) Loại bỏ các dấu chấm câu trong dữ liệu.

2. Dữ liệu môi trường của một nhà xưởng sản xuất được đo và trình bày trong tập tin đính kèm *tb_tracking.xlsx*. Thực hiện các thao tác tiền xử lý sau:
   a) Thêm vào tiêu đề các cột theo thứ tự: *id* (định danh của dòng dữ liệu), *device_id* (định danh của thiết bị đo), *co_level* (nồng độ khí CO), *humidity* (độ ẩm), *temperature* (nhiệt độ), *time* (thời điểm đo đạc).
   b) Do điều chỉnh cảm biến nên dữ liệu đo được bắt đầu từ ngày *03/01/2018* mới chính xác. Những dữ liệu còn lại không có giá trị, yêu cầu sinh viên loại bỏ.
   c) Từ cột *time* sinh viên bóc tách dữ liệu thành thời điểm đo đạc trong ngày tính theo phút, đặt tên là *minutes*.
   d) Vẽ đồ thị phân tán (scatter-plots) lần lượt biểu diễn dữ liệu nồng độ khí CO, độ ẩm và nhiệt độ theo cột *minutes* vừa tạo ở yêu cầu trên.

VI. Tài liệu tham khảo

1. *Titanic dataset*, https://data.world/nrippner/titanic-disaster-dataset
2. *Bank marketing dataset*, [Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014; https://archive.ics.uci.edu/ml/datasets/bank+marketing
3. *Predicting the Survival of Titanic Passengers,* Niklas Donges; https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers-30870ccc7e8