

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KHOA HỌC MÁY TÍNH**

**A blue logo with a black background

Description automatically generated with low confidence**

**BÀI TẬP CHƯƠNG 4**

Giảng Viên: Nguyễn Thị Anh Thư

Mã Lớp: CS313.O21

Nhóm 04

| *Trương Văn Khải* | *21520274* |
| --- | --- |
| *Đoàn Nhật Sang* | *21522542* |
| *Lê Ngô Minh Đức* | *21520195* |
| *Lê Minh Quang* | *21522510* |
| *Lê Yến Nhi* | *21522427* |
| *Hoàng Thị Mỹ Hạnh* | *21522044* |
| *Hoàng Tiến Đạt* | *21520696* |
| *Phạm Minh Quốc* | *22540017* |

**TP. HỒ CHÍ MINH, 3/2024**

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

........................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................ ............................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................ .................................................................................................................................... .................................................................................................................................... .................................................................................................................................... .................................................................................................................................... ...................................................................................................................................

**BẢNG PHÂN CÔNG CÔNG VIỆC**

| **Chi tiết công việc** | | **Trương Văn Khải (NT)** | **Đoàn Nhật Sang** | **Lê Ngô Minh Đức** | **Lê Minh Quang** | **Lê Yến Nhi** | **Hoàng Thị Mỹ Hạnh** | **Hoàng Tiến Đạt** | **Phạm Minh Quốc** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Phân công** | |  |  |  |  |  |  |  |  |
| **Độ chính xác** | **2.1.1.1 2.1.1.2** |  |  |  |  |  |  |  |  |
| **2.1.1.3** |  |  |  |  |  |  |  |  |
| **Tính đầy đủ** | **2.2.1.1.** |  |  |  |  |  |  |  |  |
| **2.2.1.2.** |  |  |  |  |  |  |  |  |
| **Tính nhất quán** | |  |  |  |  |  |  |  |  |
| **Tính kịp thời** | |  |  |  |  |  |  |  |  |
| **Tính hợp lệ** | |  |  |  |  |  |  |  |  |
| **Tính duy nhất** | |  |  |  |  |  |  |  |  |
| **Quay video** | |  |  |  |  |  |  |  |  |
| **Format docs** | |  |  |  |  |  |  |  |  |
| **Kiểm tra** | |  |  |  |  |  |  |  |  |
| **Mức độ hoàn thành** | | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** |

Bảng 0.1. Phân công công việc

**MỤC LỤC**

[1. BỘ DỮ LIỆU SỬ DỤNG: 1](#_1fob9te)

[2. KHÁM PHÁ CHẤT LƯỢNG DỮ LIỆU: 1](#_3znysh7)

[2.1. ĐỘ CHÍNH XÁC (ACCURACY) 1](#_a08mrv6yohr)

[2.1.1. Phân tích thống kê 1](#_1s8g5ghgaujo)

[2.2. TÍNH ĐẦY ĐỦ (COMPLETENESS) 11](#_vgirlb1gvvxu)

[2.2.1. Phân tích thống kê 11](#_jvhwqk0qz5p)

[2.3. TÍNH NHẤT QUÁN (CONSISTENCY) 17](#_oy82k23ba5l0)

[2.3.1. Phân tích thống kê 17](#_29pmzbsqw8my)

[2.4. TÍNH KỊP THỜI (TIMELINESS) 23](#_tlhnfczaftn)

[2.4.1. Phân tích độ trễ 23](#_yza8ecol2ix6)

[2.4.2. So sánh với thời gian thực 25](#_bqggna9pl9au)

[2.4.3. Phân tích ảnh hưởng của độ trễ 25](#_ujbs8tlrspzv)

[2.4.4. Sử dụng các công cụ giám sát 26](#_6sfjfqyyh48h)

[2.5. TÍNH HỢP LỆ (VALIDITY) 26](#_9023wbx302f2)

[2.5.1 Phân tích thống kê 26](#_zhhkhqtyowjm)

[2.6. TÍNH DUY NHẤT (UNIQUENESS) 33](#_745np0jv6y90)

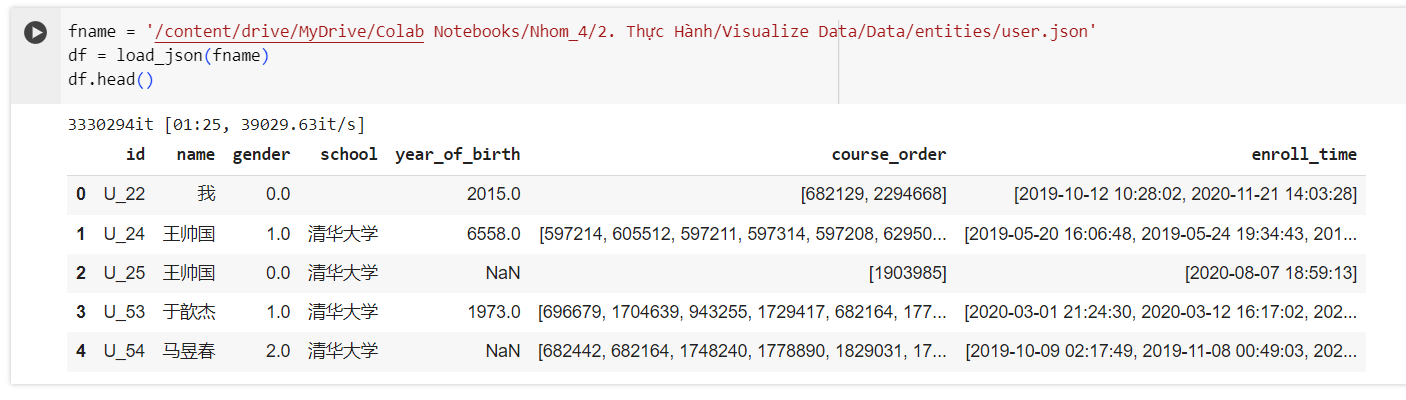
**Link video thuyết trình**:<https://drive.google.com/file/d/1v8uw5xo7MUpjyWB1h2pX1itTfxDSEU5n/view?usp=sharing>

# 1. BỘ DỮ LIỆU SỬ DỤNG:

Bảng dữ liệu sử dụng: entities/user.json

Số lượng: 3330294 mẫu

Ngôn ngữ: chủ yếu là tiếng Trung.



**Hình 1.1.** entities/user.json

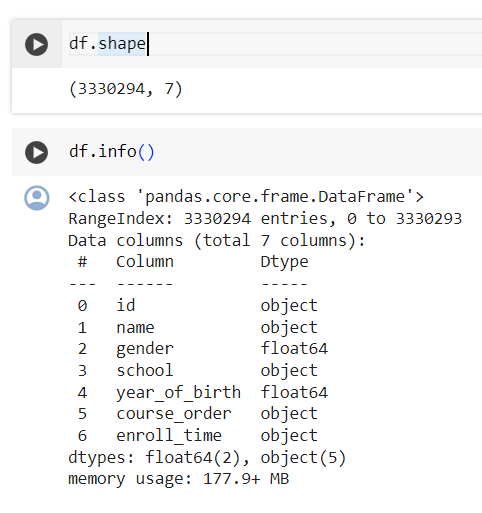
# 2. KHÁM PHÁ CHẤT LƯỢNG DỮ LIỆU:

## 2.1. ĐỘ CHÍNH XÁC (ACCURACY)

### 2.1.1. Phân tích thống kê

#### **2.1.1.1. Tỷ lệ lỗi**

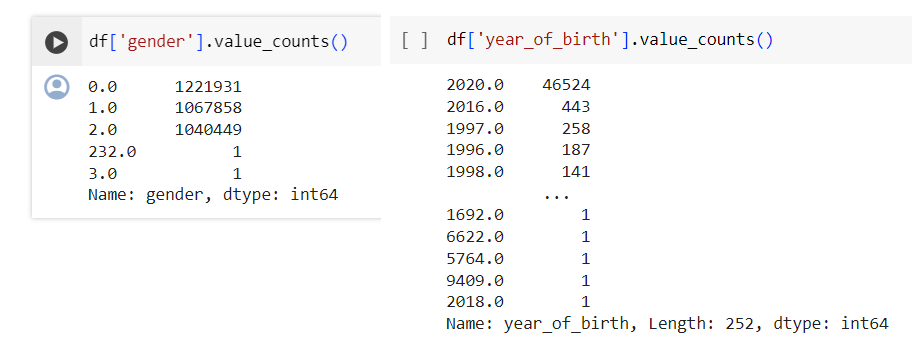
Tỷ lệ lỗi là việc đi tính toán tổng phần trăm dữ liệu sai so với tổng số dữ liệu. Đầu tiên, chúng tôi có một số quan sát cơ bản về dữ liệu như sau:



**Hình 2.1.** Thông tin chung về bảng dữ liệu entities/user.json.

Nhận xét: Có thể thấy, các dữ liệu ở 2 cột dữ liệu là course\_order, id và enroll\_time do hệ thông thu thập lại nên sẽ có rất ít dữ liệu được xem như là dữ liệu sai. Các cột dữ liệu còn lại như name, gender, school và year\_of\_birth là do người dùng nhập nên sẽ có sai sót. Ở trường hợp này, ta sẽ xem xét có trường hợp sai của hai cột dữ liệu là gender và year\_of\_birth vì hai cột dữ liệu này ở dạng float.

Phân tích các cột dữ liệu cần xử lý:



**Hình 2.2.** Bảng phân phối dữ liệu trong 2 cột thuộc tính “gender” và “year\_of\_birth”.

Nhận xét:

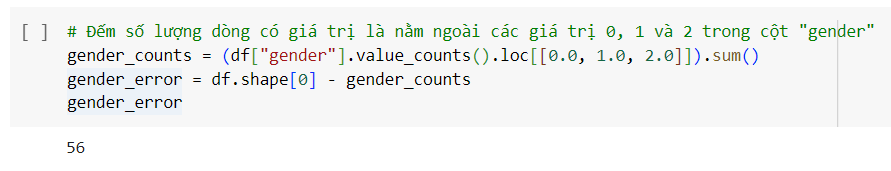
* Cột thuộc tính “gender” có khoảng 5 giá trị dữ liệu khác nhau. Trong đó có các giá trị phổ biến là 0, 1, 2 có thể đại diện cho các giá trị như: [nam, nữ, không rõ]. Tuy nhiên, vẫn có sự xuất hiện của các giá trị đặc biệt (có thể xem như giá trị ngoại lai) là các giá trị { 232.0 : xuất hiện 1 lần; 3.0: xuất hiện 1 lần}.
* Cột thuộc tính “year\_of\_birth” có rất nhiều giá trị giống nhau với tần số xuất hiện khác nhau. Vì vậy ta có thể thấy rất khó để kiểm soát dữ liệu của cột thuộc tính này.

Vì vậy, để tính số điểm dữ liệu sai ở mỗi cột thuộc tính, chúng tôi quyết định thực hiện theo các bước sau:

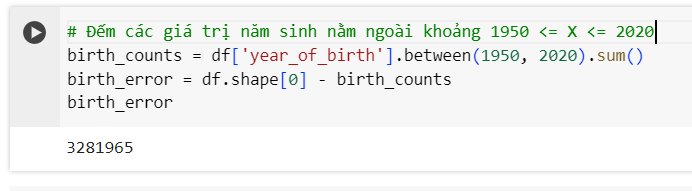
* Ở tất cả các cột: Coi các dòng dữ liệu còn thiếu (mang giá trị NaN) như là một dòng dữ liệu sai.
* Ở cột “gender”: coi các giá trị nằm ngoài khoảng [0.0, 1.0, 2.0] là các điểm dữ liệu sai.
* Ở cột “year\_of\_birth”:
  + Số lượng hàng có giá trị là 48530 --> rất ít so với số lượng học sinh
  + Có hàng có giá trị không hợp lệ. Ví dụ: user có id để year\_of\_birth là 6558

Để tiến hành demo, nhóm sẽ xét các hàng không có giá trị và nếu có thì chỉ chấp nhận nếu 1950 <= year\_of\_birth <= 2020. Ngoài khoảng này sẽ là các điểm dữ liệu sai.

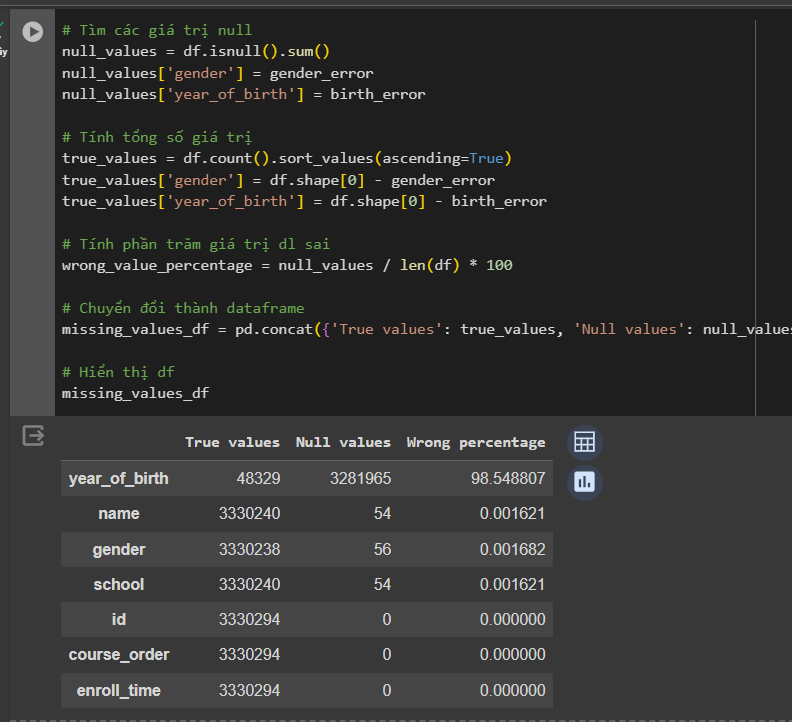
Code**:**



**Hình 2.3.** Đếm số điểm dữ liệu sai trong cột thuộc tính “gender”.



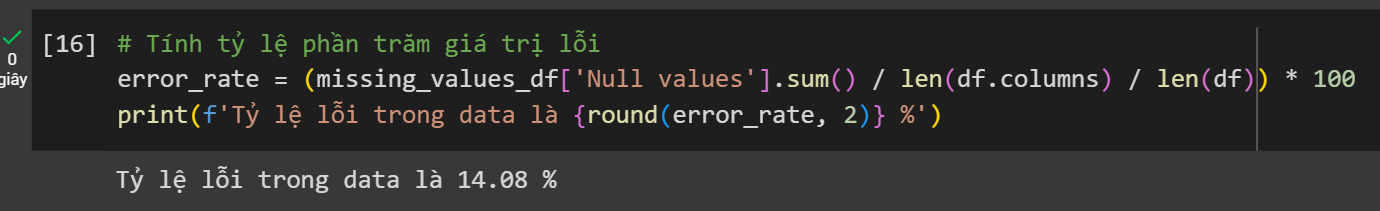
**Hình 2.4.** Đếm số điểm dữ liệu sai trong cột thuộc tính “year\_of\_birth”.



**Hình 2.5.** Thực hiện đếm tỷ lệ lỗi cho từng cột thuộc tính và kết quả.

Nhận xét: Qua bảng kết quả trả về được lưu trong biến missing\_values\_df, ta thấy được các giá trị được sắp xếp theo thứ tự giảm dần theo tỷ lệ dữ liệu bị sai. Có thể thấy được:

* Dữ liệu ở cột year\_of\_birth bị sai rất nặng, với khoảng 3,281,965 dòng bị sai trên tổng 3,330,294 dòng dữ liệu. Tỷ lệ phần trăm dữ liệu bị sai được tính ra khoảng 98.55%.
* Các cột dữ liệu name, gender, school đồng loạt bị sai khoảng 56 dòng (khá ít) so với khoảng tổng 3,330,294 dòng dữ liệu. Tỷ lệ phần trăm dữ liệu sai được tính ra là khoảng 0.0016%.
* Các cột dữ liệu còn lại không bị sai về mặt dữ liệu.



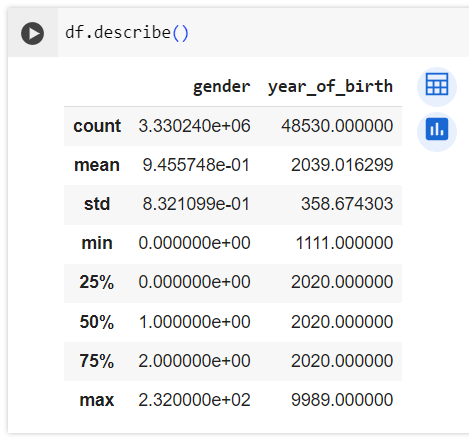
**Hình 2.6.** Thực hiện đếm tỷ lệ lỗi trung bình cho toàn bộ dữ liệu và kết quả trả về.

Nhận xét: Tỷ lệ lỗi cho toàn bộ bảng dữ liệu là khoảng 14.08%. Đây là một tỷ lệ không quá cao. Tuy nhiên nguyên nhân chủ đạo cho con số 14.08% này gây ra bởi cột thuộc tính “year\_of\_birth”. Nhìn ở Hình 2.5 có thể thấy có đến 3,281,965 điểm dữ liệu bị xét là sai, tuy nhiên toàn bộ bảng dữ liệu chỉ có khoảng 3,330,294 mẫu.

#### 2.1.1.2. Độ lệch chuẩn

Giá trị std (độ lệch chuẩn) của mỗi cột dữ liệu là một đánh giá về mức độ biến động của dữ liệu. Tức là độ lệch chuẩn sẽ cho thấy được khoảng cách hay thước đo độ phân tán của một tập hợp dữ liệu so với giá trị trung bình của nó. Nó đo lường sự biến thiên tuyệt đối của một phân phối. Độ phân tán hoặc độ biến thiên càng cao thì độ lệch chuẩn càng lớn và độ lệch so với giá trị trung bình càng lớn.

Code:



**Hình 2.7.** Thực hiện lấy ra giá trị std dựa vào hàm describe() cho các cột thuộc tính dạng số.

Nhận xét:

* Đối với cột "gender" (giới tính), giá trị std là khoảng 0.832, điều này cho thấy mức độ biến động của dữ liệu giới tính là khá thấp.
* Đối với cột "year\_of\_birth" (năm sinh), giá trị std là khoảng 358.674, điều này cho thấy mức độ biến động của dữ liệu năm sinh là khá cao.

#### 2.1.1.3. Hệ số tương quan - Pearson.

Hàm: scipy.stats.pearsonr

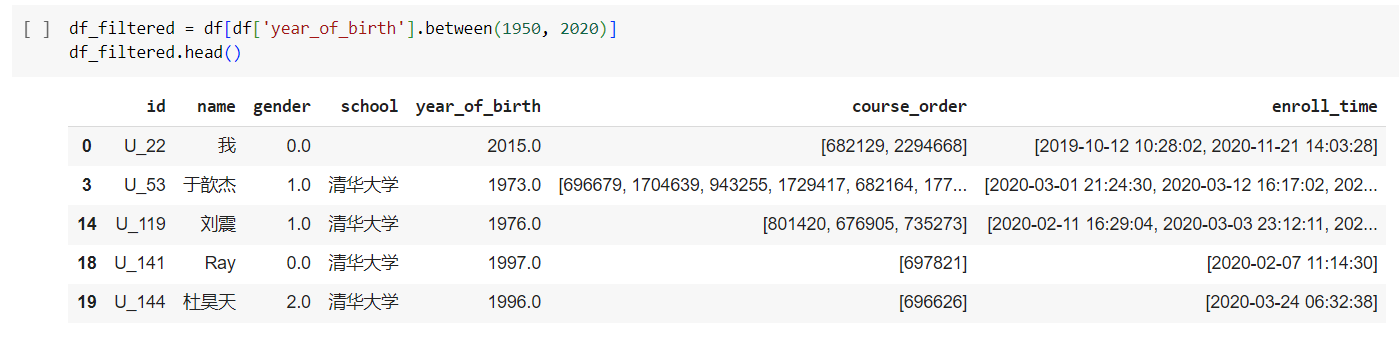
Công dụng: Tính hệ số tương quan Pearson giữa 2 biến ngẫu nhiên liên tục. Hệ số tương quan Pearson ( là thước đo mối quan hệ tuyến tính giữa hai biến và không phụ thuộc vào các đơn vị đo lường của hai biến này. Nếu hệ số tương quan càng gần -1 hoặc 1 thì tương quan tuyến tính càng mạnh, nếu càng gần 0 thì tương quan càng yếu. Nếu hệ số tương quan < 0 thì thể hiện sự nghịch biến, > 0 thì thể hiện sự đồng biến.

Cú pháp: scipy.stats.pearsonr(*x*, *y*, *\**, *alternative='two-sided'*, *method=None*)

* Parameters
  + x (array\_like): array của biến thứ 1
  + y (array\_like): array của biến thứ 2
  + alternative (string): Giả thuyết thay thế. Nhận 1 trong các giá trị sau:
    - ‘two-sided’: tương quan khác không
    - ‘less’: tương quan âm (<0)
    - ‘greater’: tương quan dương (>0)
  + method (ResamplingMethod): Hàm định nghĩa các tính p-value. Nếu method=None (mặc định) thì sẽ tính theo công thức mặc định trong note của scipy.stats.pearsonr
* Return: một đối tượng chứa 2 giá trị sau:
  + statistic (float): hệ số tương quan r
  + pvalue (float): p-value. Nếu p-value < (thường là 0.05) thì bác bỏ giả thuyết null, nghĩa là chấp nhận giả thuyết đối.  *Ngược lại* *thì không đủ cơ sở để bác bỏ giả thuyết null.*

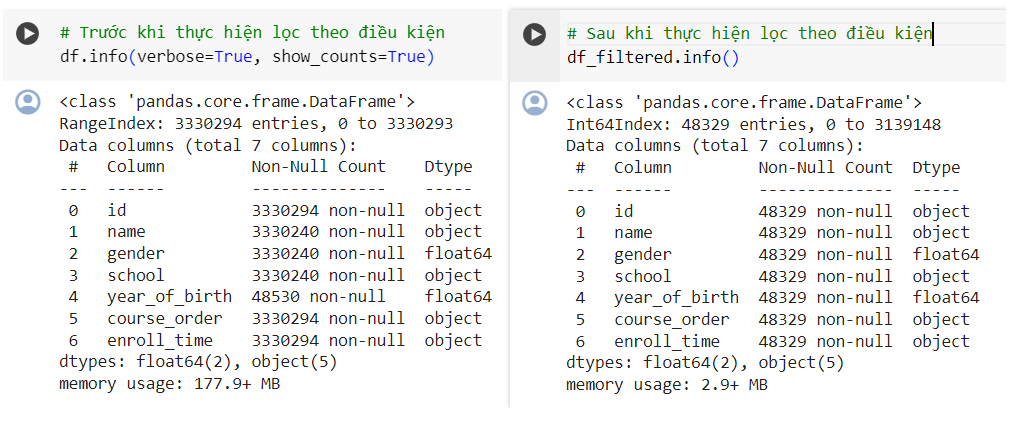
Code:

Đầu tiên, chúng ta sẽ thực hiện lọc lại toàn bộ bảng dữ liệu theo điều kiện năm sinh (vì đây là cột có số lượng dữ liệu sai rất lớn) để tránh được sự ảnh hưởng tiêu cực đến việc đánh giá mối tương quan giữa các cột thuộc tính.



**Hình 2.8.** Lọc bảng dữ liệu theo điều kiện năm sinh X, với 1950 <= X <= 2020.

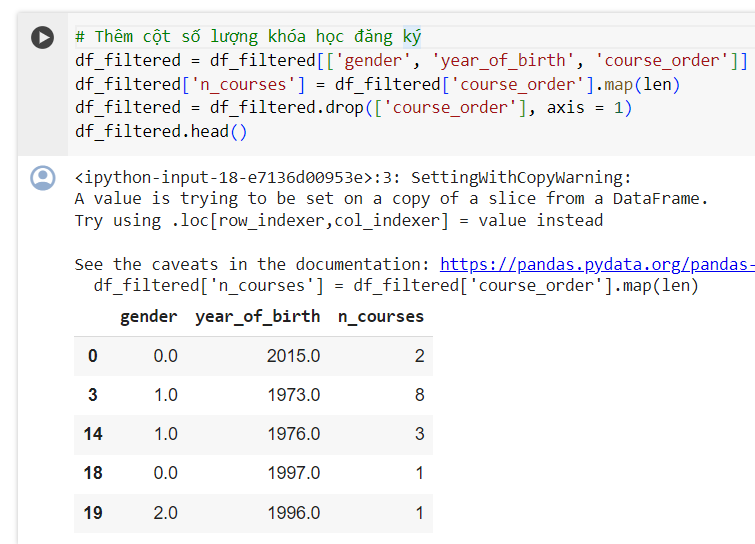
Sau đó, cùng nhìn lại thống kê chung của bảng dữ liệu trước và sau khi thực hiện lọc theo điều kiện năm sinh:



**Hình 2.9.** Thống kê bảng dữ liệu trước và sau khi thực hiện lọc theo điều kiện.

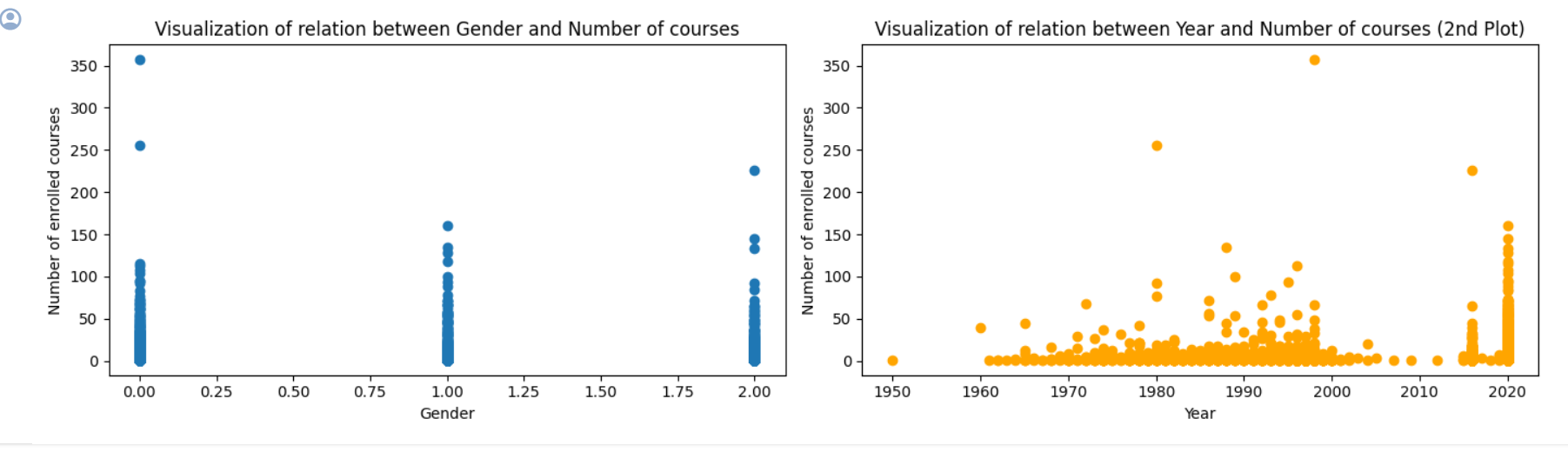
Để xét độ tương quan, chúng ta cần đặt ra một bài toán giả định.

Bài toán giả sử: Xem xét sự ảnh hưởng / tương quan giữa các cột Giới tính (gender), năm sinh (year\_of\_birth) tới việc đăng ký số lượng khóa học. Để làm được điều đó chúng ta sẽ cần phải tạo ra một cột số lượng khóa học đăng ký. (lấy ra len của cột course\_order)



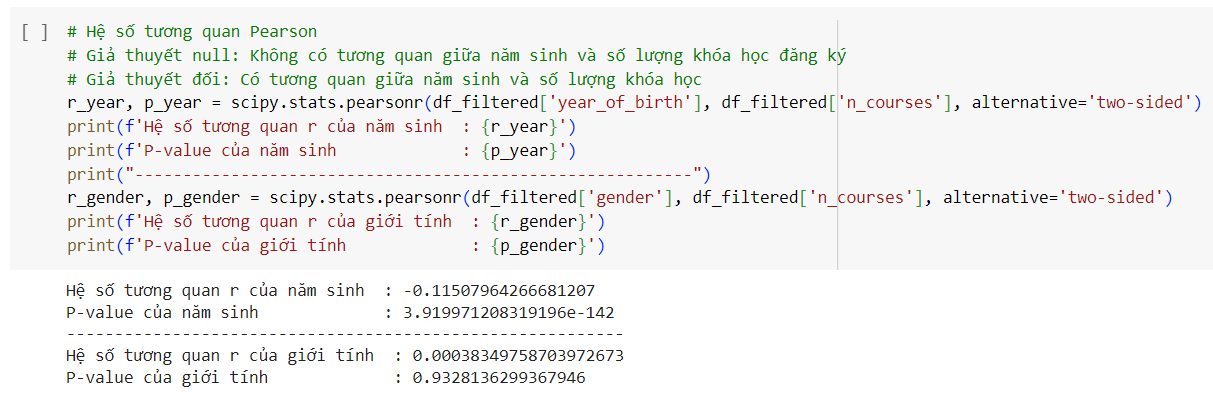
**Hình 2.10.** Tạo bảng dữ liệu mới theo điều kiện giả định.

Vẽ biểu đồ phân tán (Scatter Plot) để thể hiện sự tương quan giữa các cột thuộc tính year\_of\_birthso với số khóa học mà người dùng đăng ký:



**Hình 2.11.** Biểu đồ phân tán thể hiện sự tương quan giữa các cột thuộc tính year\_of\_birth so với số học mà người dùng đăng ký.

Xét mối tương quan giữa lần lượt 2 biến year\_of\_birth (năm sinh) với n\_courses (số lượng khóa học đăng ký của học sinh):



**Hình 2.12.** Demo cho hệ số tương quan Pearson.

Nhận xét:

* Xét biến "năm sinh" và "số lượng khóa học đăng ký":
  + p-value < 0.05 --> Có quan hệ tuyến tính với nhau
  + Hệ số tương quan r gần bằng 0 --> Tương quan tuyến tính rất yếu

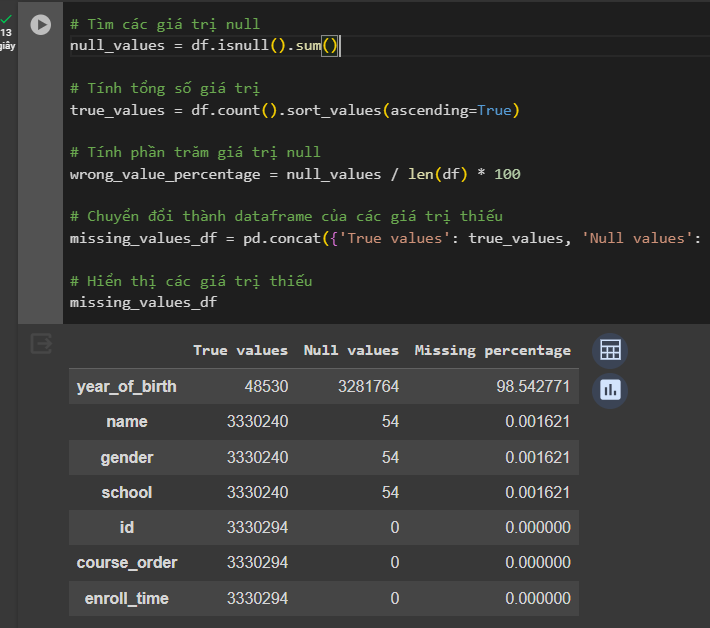
## 2.2. TÍNH ĐẦY ĐỦ (COMPLETENESS)

### 2.2.1. Phân tích thống kê

#### 2.2.1.1. Tỷ lệ dữ liệu thiếu.

Ở bước này, chúng tôi thực hiện tương tự phần 2.1.1.1, chỉ khác là chúng tôi chỉ xét đến dữ liệu thiếu, không xét đến lượng dữ liệu bị sai.

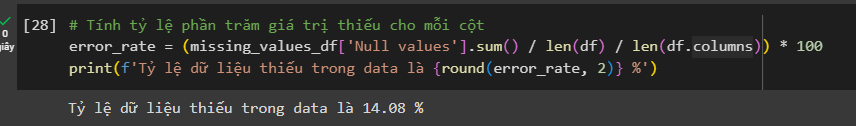
Code**:**



**Hình 2.13.** Thực hiện đếm tỷ lệ dữ liệu thiếu cho từng cột thuộc tính và kết quả.

Nhận xét: Qua bảng kết quả trả về được lưu trong biến missing\_values\_df, ta thấy được các giá trị được sắp xếp theo thứ tự giảm dần theo tỷ lệ dữ liệu bị thiếu. Có thể thấy được:

* Dữ liệu ở cột year\_of\_birth bị thiếu rất nặng, với khoảng 3,281,764 dòng bị thiếu trên tổng 3,330,294 dòng dữ liệu. Tỷ lệ phần trăm dữ liệu bị thiếu được tính ra khoảng 98.54%.
* Các cột dữ liệu name, gender, school đồng loạt bị thiếu khoảng 54 dòng (khá ít) so với khoảng tổng 3,330,294 dòng dữ liệu. Tỷ lệ phần trăm dữ liệu thiếu được tính ra là 0.0016%.
* Các cột dữ liệu còn lại không bị thiếu.



**Hình 2.14.** Thực hiện đếm tỷ lệ dữ liệu thiếu trung bình cho toàn bộ dữ liệu và kết quả trả về.

Nhận xét: Tỷ lệ lỗi cho toàn bộ bảng dữ liệu là khoảng 14.08%. Đây là một tỷ lệ không quá cao. Tuy nhiên nguyên nhân chủ đạo cho con số 14.08% này gây ra bởi cột thuộc tính “year\_of\_birth”. Nhìn ở Hình 2.5 có thể thấy có đến 3,281,764 điểm dữ liệu bị thiếu, trong khi toàn bộ bảng dữ liệu chỉ có khoảng 3,330,294 mẫu.

#### 2.2.1.2. Phân tích phân bố.

Phân tích phân bố dữ liệu giúp chúng ta hiểu rõ hơn về cách mà các giá trị được phân phối trong tập dữ liệu và đánh giá sự phân bố của dữ liệu để xác định các giá trị bị thiếu một cách có hệ thống.. Bằng cách này, chúng ta có thể:

* Xác định các giá trị bị thiếu: Bằng cách kiểm tra phân phối của mỗi cột dữ liệu, chúng ta có thể nhận biết các giá trị bị thiếu một cách có hệ thống.
* Đánh giá sự đa dạng của dữ liệu: Phân tích phân bố giúp chúng ta nhận biết được sự đa dạng của dữ liệu, từ đó có thể quyết định các phương pháp xử lý dữ liệu và xây dựng mô hình phù hợp.
* Chuẩn bị dữ liệu cho việc xây dựng mô hình: Bằng cách hiểu rõ phân phối của dữ liệu, chúng ta có thể thực hiện các bước tiền xử lý như xử lý giá trị bị thiếu, chuẩn hóa dữ liệu, và chọn lọc đặc trưng để chuẩn bị dữ liệu cho việc xây dựng mô hình.

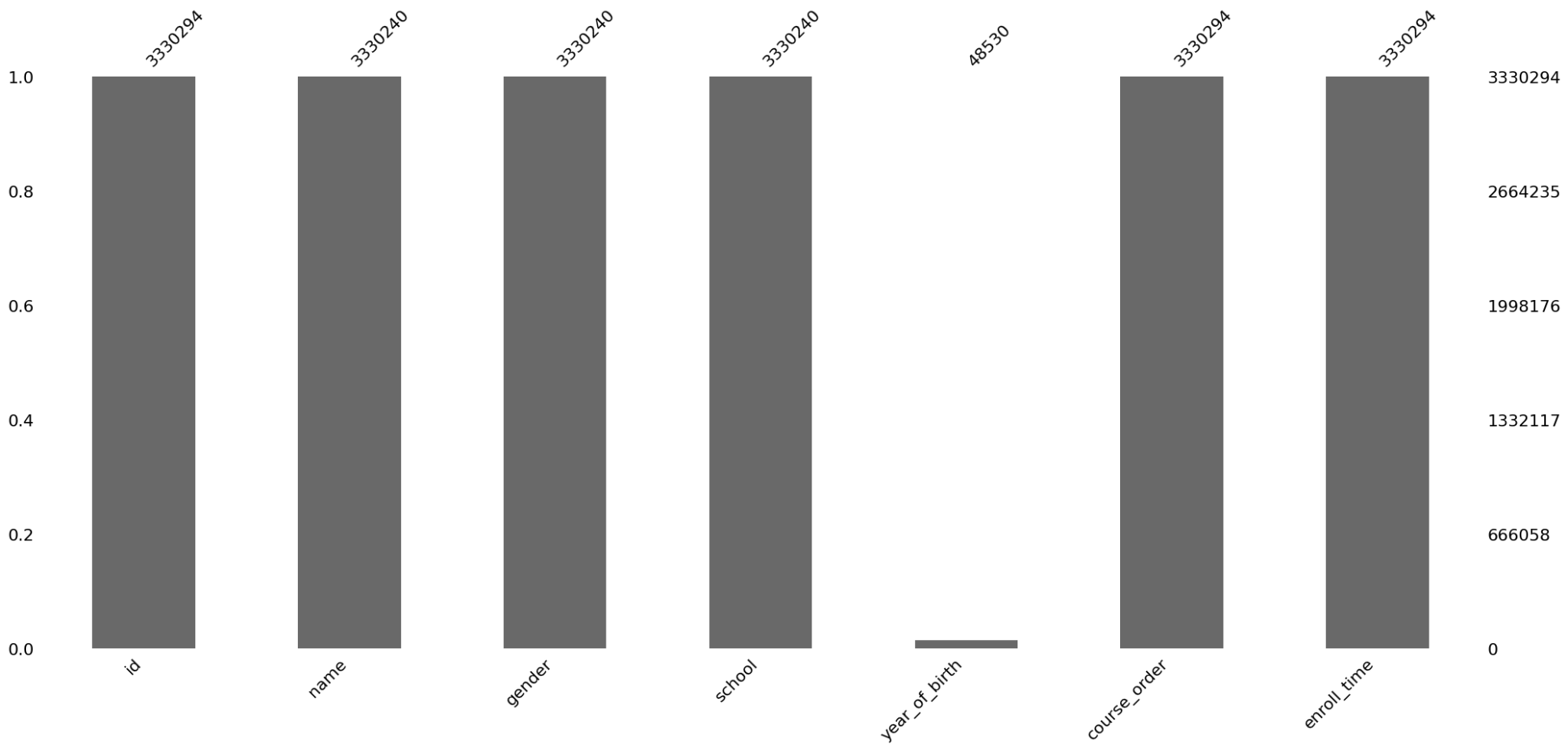
Code: Để phân tích đồ họa về việc thiếu dữ liệu, chúng tôi sử dụng một thư viện được gọi là Missingno. Đây là một gói công cụ cho phân tích đồ họa của các giá trị bị thiếu.

Đầu tiên chúng tôi sử dụng hàm msno.bar():

Cú pháp:

import missingno as msno

msno.bar(df)

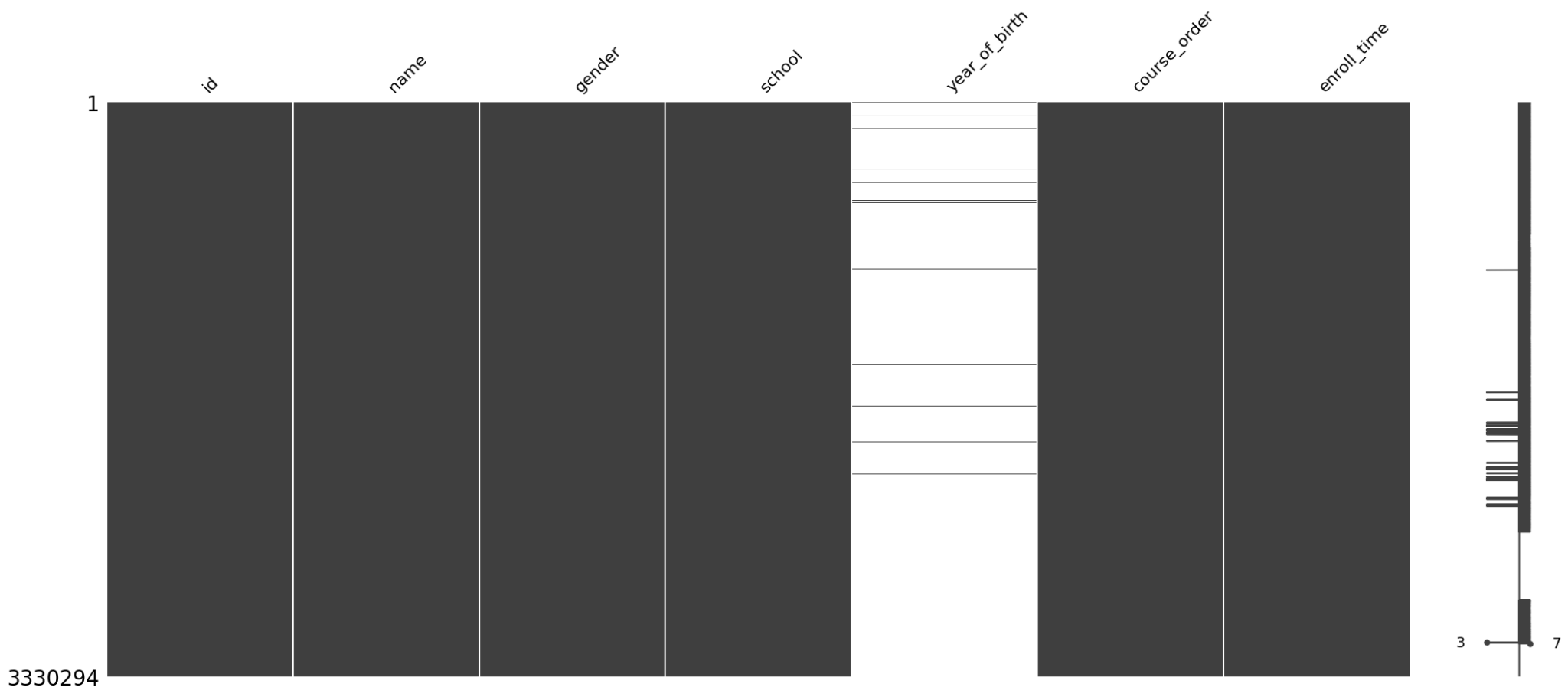
Công dụng: Hàm "msno.bar(df)" được sử dụng để tạo biểu đồ thanh (bar chart) của dữ liệu trong DataFrame (df), hiển thị số lượng giá trị không bị thiếu cho mỗi cột dữ liệu. 

**Hình 2.15.** Vẽ biểu đồ cột bằng hàm msno.bar() để quan sát tỷ lệ dữ liệu không bị thiếu trong từng biến và đánh giá mức độ khuyết thiếu dữ liệu trong tập dữ liệu.

Nhận xét: Chúng ta có thể thấy rằng các cột name, gender, school và year\_of\_birth có giá trị bị thiếu, trong đó cột year\_of\_birth bị thiếu rất nặng.Tiếp theo, chúng tôi sẽ tìm ra vị trí của dữ liệu bị thiếu.

Cú pháp: msno.matrix(df)

Công dụng: Hàm "msno.matrix(df)" được sử dụng để tạo ma trận biểu đồ (matrix plot) của dữ liệu trong DataFrame (df), hiển thị mức độ thiếu dữ liệu trong mỗi cột dữ liệu. Biểu đồ này sẽ hiển thị các ô trống trong ma trận, biểu thị vị trí của dữ liệu bị thiếu.



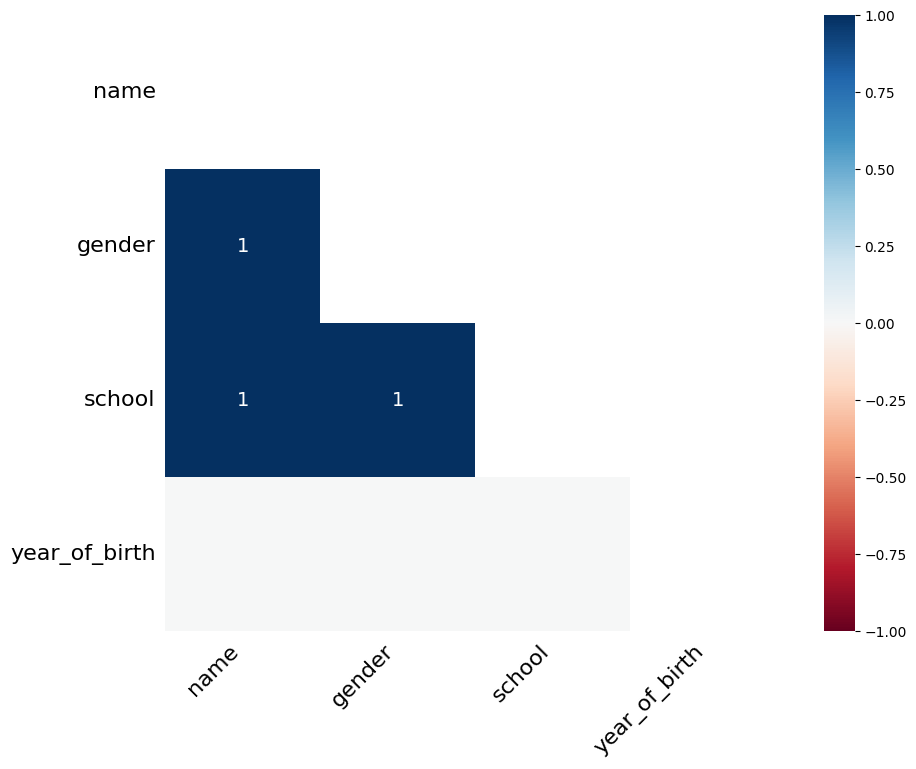
**Hình 2.16.** Vẽ biểu đồ cột bằng hàm msno.matrix() để nhận ra mẫu đặc điểm thiếu dữ liệu và phân bố chúng trong tập dữ liệu.

Nhận xét:

* Biểu đồ sẽ xuất hiện trống (màu trắng) ở những nơi có giá trị bị thiếu. Ví dụ, trong cột year\_of\_birth chỉ có rất nhiều khoảng trắng bị thiếu, nên thể hiện cột này đang bị mất mát dữ liệu nghiệm trọng.
* Đường Sparkline bên phải cho một ý tưởng về hình dạng tổng quan của tính hoàn chỉnh của dữ liệu và chỉ ra dòng có số giá trị không bị thiếu tối thiểu và tổng số cột trong một bộ dữ liệu cụ thể, ở dưới cùng.

Tiếp theo, chúng tôi sử dụng Heatmap để hiển thị sự tương quan của sự thiếu sót giữa mỗi cặp cột.

Cú pháp: msno.heatmap(df, figsize=(10,8))

Công dụng: Hàm "msno.heatmap(df, figsize=(10,8))" được sử dụng để tạo biểu đồ heatmap của dữ liệu trong DataFrame (df), với kích thước biểu đồ là (10, 8). Biểu đồ heatmap này biểu diễn mối tương quan giữa các giá trị bị thiếu của các cột dữ liệu. Các ô màu sẽ cho biết mức độ tương quan giữa sự thiếu dữ liệu của các cặp biến

**Hình 2.17.** Vẽ biểu đồ cột bằng hàm msno.heatmap() để nhận ra được mối quan hệ giữa sự thiếu dữ liệu của các biến và có thể xác định các mẫu đặc trưng trong dữ liệu.

Nhận xét: Một giá trị gần -1 có nghĩa là nếu một biến xuất hiện thì có khả năng cao biến số kia sẽ bị thiếu. Một giá trị gần 0 có nghĩa là không có sự phụ thuộc giữa việc xuất hiện của các giá trị bị thiếu của hai biến. Một giá trị gần 1 có nghĩa là nếu một biến xuất hiện thì có khả năng cao biến số kia sẽ xuất hiện.

* Các dữ liệu ở cột (gender, name), (gender, school), (school, name) có giá trị là 1, tức là khi 1 điểm dữ liệu ở cột này xuất hiện thì điểm dữ liệu ở cột kia sẽ không bao giờ bị thiếu.
* Cột year\_of\_birth là cột bị thiếu dữ liệu nhiều nhất, nên hệ số tương quan được biểu diễn trong heatmap so với các cột thuộc tính name, gender, school gần như bằng 0.

## 2.3. TÍNH NHẤT QUÁN (CONSISTENCY)

### 2.3.1. Phân tích thống kê

#### 2.3.1.1. Kiểm tra thống kê.

Ở phần này, chúng tôi thực hiện sử dụng các kiểm tra thống kê t-test để xác định

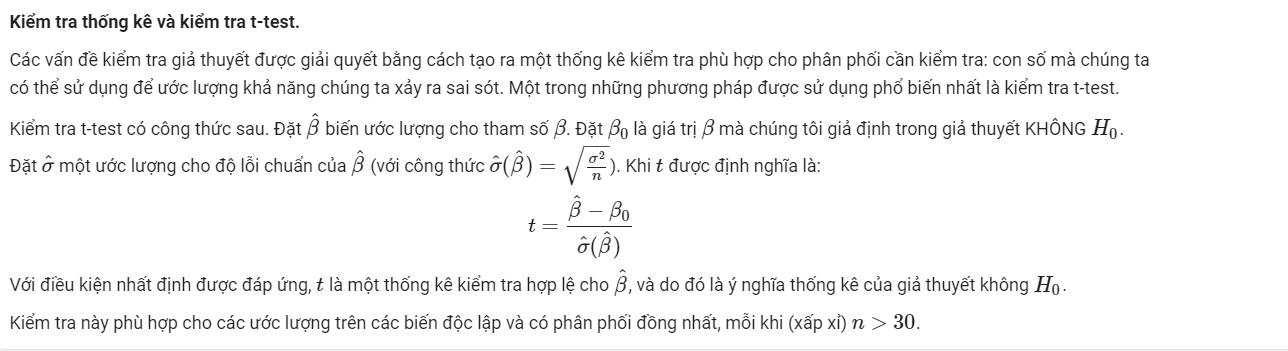
xem có sự khác biệt đáng kể nào giữa các giá trị dữ liệu hay không.

Đầu tiên, chúng tôi thực hiện tạo lại bảng dữ liệu entities/user.json bằng việc xử lý các giá trị ngoại lai ở cột thuộc tính “gender” (gender thuộc [0.0, 1.0, 2.0]), tạo thêm cột n\_courses (số lượng khóa học đăng ký) và lọc đi các năm sinh ở cột year\_of\_birth theo ràng buộc (năm sinh X, 1950 <= X <= 2020).



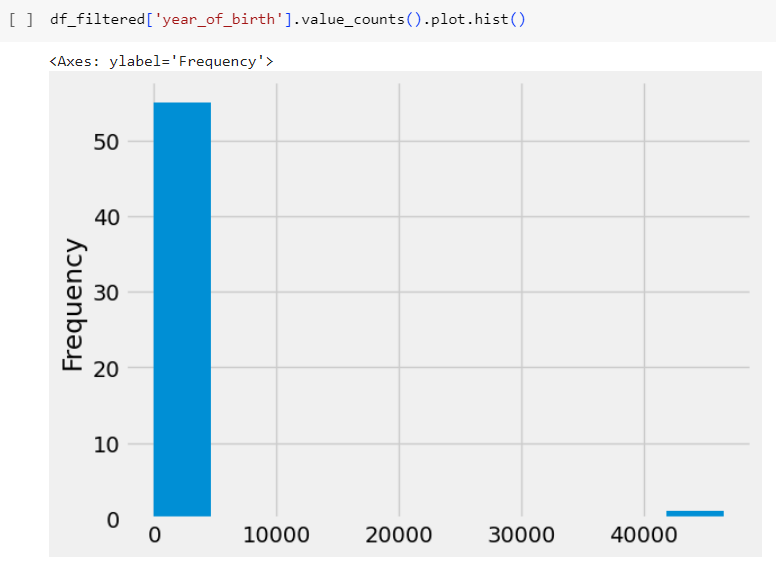
**Hình 2.18.** Tạo lại bảng dữ liệu để thực hiện kiểm tra thống kê.

Khái niệm:



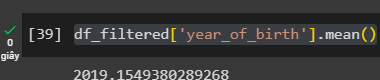
**Hình 2.19.** Khái niệm kiểm tra t-test từ ggcolab của chúng tôi.

Giả sử chúng ta quan tâm đến việc đo lường số lượng trung bình theo năm sinh. Nếu chúng ta xem xét dữ liệu này, đây là những gì chúng ta tìm thấy:



**Hình 2.20.** Biểu đồ histogram thể hiện tần số suất hiện của các ngưỡng năm sinh.

Sau đó, chúng tôi sẽ lấy ngưỡng trung bình của giá trị năm sinh trong cột “year\_of\_birth” để làm thiết lập các giả thuyết kiểm tra. Chúng tôi sẽ lấy trung bình như sau:

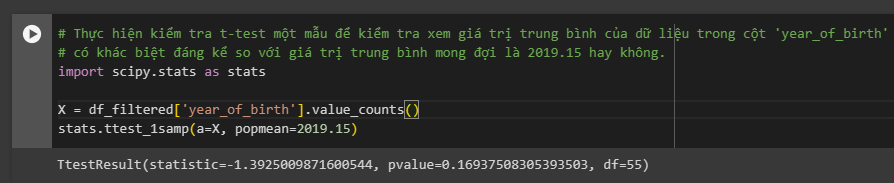


**Hình 2.21.** Lấy giá trị trung bình của năm sinh.

Từ đó, chúng tôi có thể đặt ra được các giả thuyết như sau:



Bây giờ chúng ta sẽ thực hiện kiểm tra t-test: chúng tra thực hiện kiểm định t-test để kiểm tra một mẫu để xem giá trị trung bình của dữ liệu trong cột 'year\_of\_birth' có khác biệt đáng kể so với giá trị trung bình mong đợi là 2019.15 hay không.



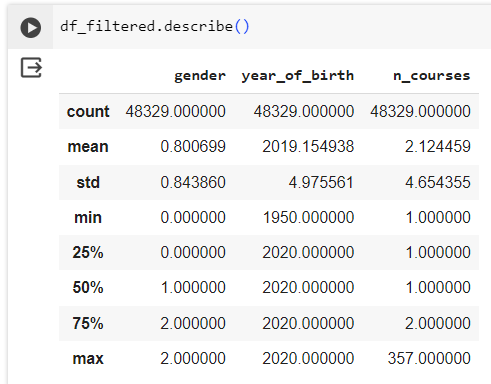
**Hình 2.22.** Kiểm tra t-test và kết quả trả về thu được.

Nhận xét: p-value > 0.05 → Không đủ cơ sở để bác bỏ giả thuyết null

#### 2.3.1.2. Phân tích độ tương quan.

Ở bước này, chúng tôi sẽ đánh giá mối liên hệ giữa các thuộc tính dữ liệu để xác định xem chúng có nhất quán với nhau hay không.

Tiếp theo, chúng ta có thể nhanh chóng lấy phân tích thống kê (tức là trung bình, trung vị, độ lệch chuẩn, v.v.) bằng cách sử dụng phương thức describe().

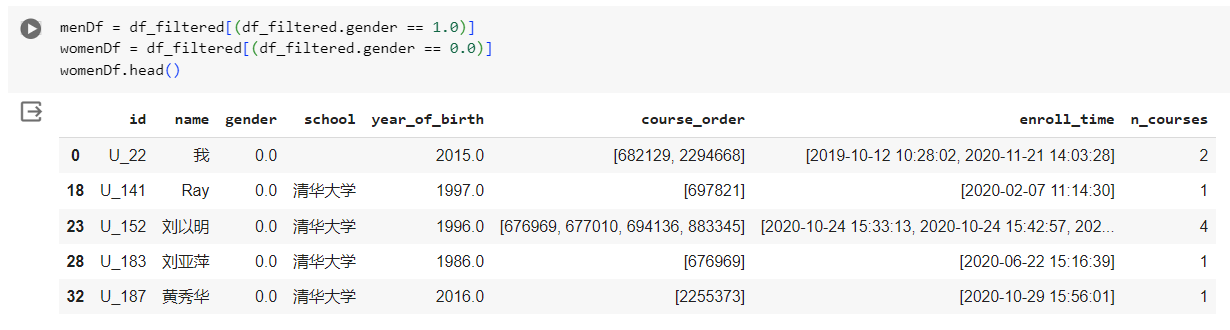


**Hình 2.24.** Kết quả trả về sau khi dùng hàm để thống kê mô tả các dữ liệu ở dạng số.

Chúng ta có thể thấy tất cả các phân tích thống kê cho tập dữ liệu từ bảng trên, ví dụ: số lượng (tổng số dữ liệu trong tập dữ liệu), trung bình, độ lệch chuẩn, giá trị tối thiểu, phân vị thứ nhất, phân vị thứ hai, phân vị thứ ba và giá trị tối đa trong tập dữ liệu.

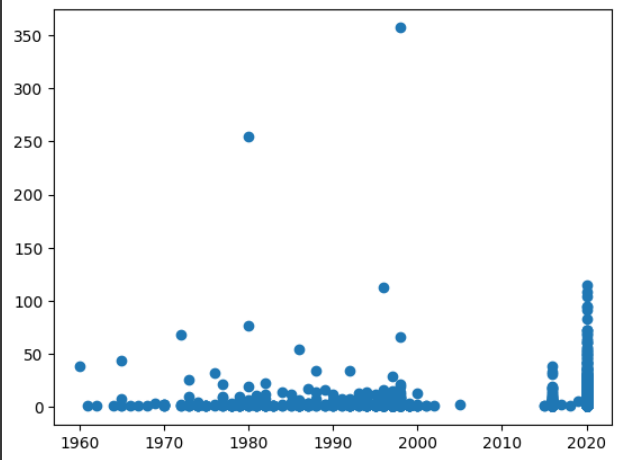
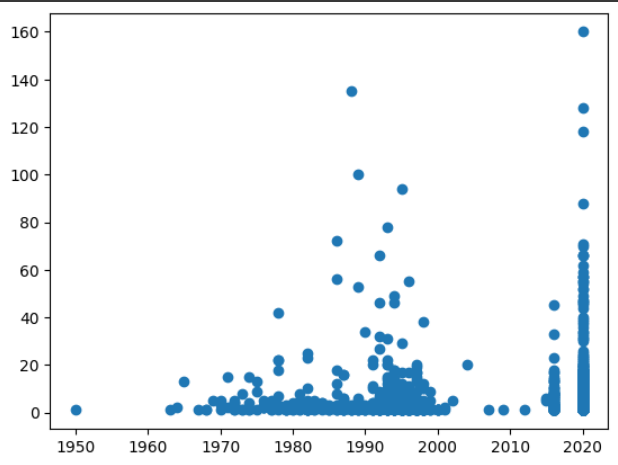
Tiếp theo, chúng ta sẽ vẽ dữ liệu trong một biểu đồ scatter plot. Biểu đồ scatter plot là quan trọng khi làm việc với các tương quan vì chúng cho phép xác minh một cách nhanh chóng mối quan hệ giữa các biến. Ở bài này, chúng tôi sử dụng hệ số tương quan Pearson, chỉ tương quan đến mối quan hệ tuyến tính giữa hai biến.

Để đảm bảo kết quả không bị sai khớp do sự khác biệt giữa nam (giả sử là 1) và nữ (giả sử là 0), tập dữ liệu được chia thành hai dataframe: một chứa tất cả các thông tin liên quan đến nam (menDf) và một dataframe khác chỉ chứa các thông tin liên quan đến nữ (womenDf).



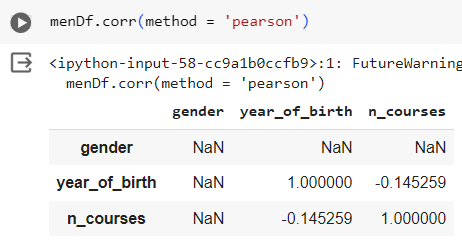
**Hình 2.25.** Lọc và tạo ra 2 dataframe chỉ chứa các thông tin về nam hoặc nữ.

Sau đó, sử dụng phương thức scatter() của matplotlib để tạo ra biểu đồ phân tán giữa biến cột 'year\_of\_birth' và thuộc tính n\_courses (số khóa học đăng ký).



**Hình 2.26.** Biểu đồ phân tán giữa cột 'year\_of\_birth' và thuộc tính n\_courses (số khóa học đăng ký). Trái là của nam, Phải là của nữ.

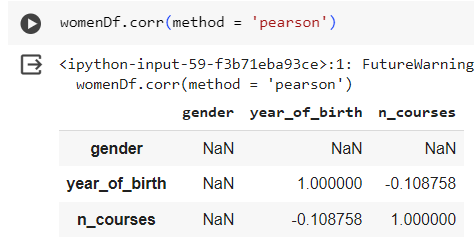
Cả hai biểu đồ phân tán trên không thể hiện mối tương quan giữa năm sinh và số khóa học đăng ký, vì các giá trị được vẽ quá ngẫu nhiên. Chúng ta không thể xác định mối tương quan giữa các biến với phương pháp này, vì vậy hãy thử một phương pháp khác là Pandas corr().



**Hình 2.27.** Biểu đồ tương quan giữa các cột thuộc tính cho nam giới, sử dụng phương pháp Pearson.

Nhận xét: Đối với giới tính nam, chúng ta có thể thấy rằng n\_courses (số khóa học đăng ký) có giá trị tương quan rất bất thường:

* n\_courses và year\_of\_birth có mối tương quan âm (gần như k có tương quan).

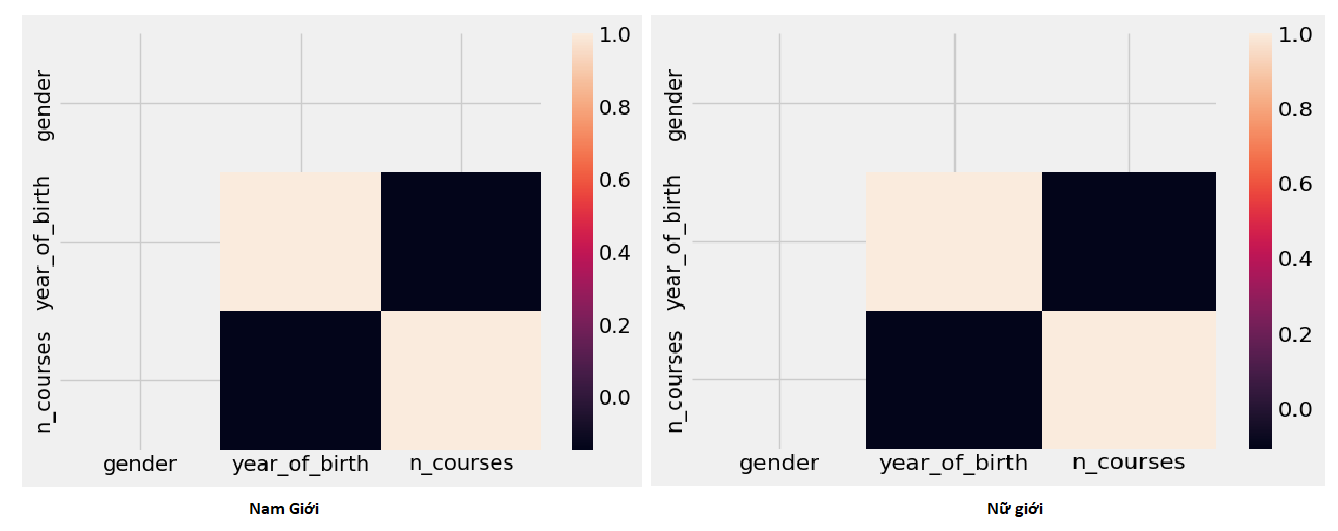


**Hình 2.28.** Biểu đồ tương quan giữa các cột thuộc tính cho nữ giới, sử dụng phương pháp Pearson.

Nhận xét: Đối với giới tính nữ, chúng ta có thể thấy rằng n\_courses (số khóa học đăng ký) có giá trị tương quan rất bất thường:

* n\_courses và year\_of\_birth có mối tương quan âm (gần như k có tương quan).

Vì chúng ta đã có các giá trị tương quan giữa các biến, cuối cùng chúng ta có thể biểu diễn chúng thành biểu đồ heatmap.



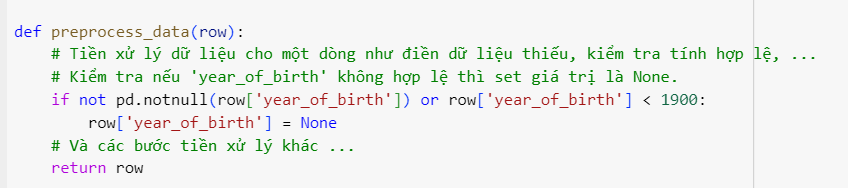
**Hình 2.29.** Biểu đồ heatmap biểu thị sự tương quan giữa các cột thuộc tính cho cả nam (bên trái) và nữ (bên phải).

## 2.4. TÍNH KỊP THỜI (TIMELINESS)

### 2.4.1. Phân tích độ trễ

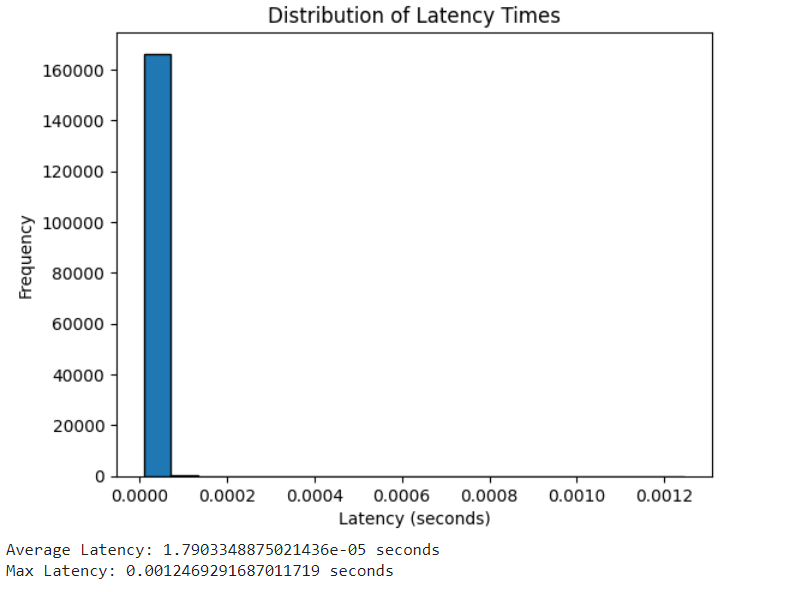
Ở phần này chúng tôi thực hiện mô phỏng độ trễ của dữ liệu mới bằng cách tách tập data thành train/test theo tỉ lệ 95 (df\_train) - 5 (df\_test) sau đó tiền xử lí trên từng hàng một, gộp kết quả vào df\_train, và cuối cùng là in ra độ trễ tổng thể và trung bình của tất cả các dòng trong tập test.

Cụ thể, mỗi dữ liệu ở tập df\_test sẽ được xem như dữ liệu thô từ bên ngoài vào, và sẽ đi qua hàm tiền xử lí đại diện (*preprocess\_data*) rồi sau đó append (gắn) vào tập df\_train, mô phỏng việc cập nhật dữ liệu trong thực tế. Ở đây chúng tôi sử dụng hàm *preprocess\_data* có chức năng kiểm tra nếu 'year\_of\_birth' không hợp lệ thì set giá trị là None.



**Hình 2.30.** Hàm tiền xử lí đại diện

Tiếp theo chúng tôi vẽ sơ đồ histogram thể hiện phân bố của độ trễ bằng cách đưa từng độ trễ của từng dòng trong tập df\_test (sau khi đi qua hàm tiền xử lí đại diện) vào một danh sách, tính min và max của nó sau đó biểu hiện lên đồ thị.



**Hình 2.31.** Histogram thể hiện phân bố độ trễ.

Dễ thấy từ hình trên, độ trễ trung bình là 1.79\*10^-5s và lớn nhất là 0.0012s.

### 2.4.2. So sánh với thời gian thực

Để phân tích độ trễ thời gian cập nhật dữ liệu thực tế trong việc đăng ký khóa học, chúng tôi thực hiện ba bước chính như sau:

* Bước 1: Tạo một danh sách tổng hợp A chứa tất cả các thời điểm đăng ký (enroll\_time) từ mỗi dòng của dữ liệu. Mỗi dòng có thể chứa một list các thời điểm đăng ký, và ta sẽ kết hợp tất cả các list này vào một danh sách lớn duy nhất.
* Bước 2: Sắp xếp danh sách A theo thứ tự tăng dần của thời gian để đảm bảo rằng các thời điểm đăng ký được xem xét theo trình tự thời gian chính xác.
* Bước 3: Tính khoảng thời gian giữa các thời điểm đăng ký liên tiếp (giữa phần tử i và i+1) và ghi chú vào danh sách mới B. Điều này sẽ cho chúng ta biết khoảng thời gian cần thiết để một khóa học mới được đăng ký sau khóa học trước đó. Tính trung bình của danh sách B sẽ cung cấp cho chúng ta độ trễ trung bình giữa các lần đăng ký, phản ánh khả năng cập nhật dữ liệu thời gian thực của hệ thống.

Kết quả cuối cùng sẽ giúp hiểu được tần suất cập nhật dữ liệu và đánh giá xem hệ thống có đáp ứng kịp thời đối với nhu cầu đăng ký khóa học không.

Sau khi thực hiện phân tích như trên, chúng tôi nhận thấy độ trễ trung bình thực tế là khoảng 5s, dễ dàng đáp ứng nhu cầu (vì độ trễ tối đa của hệ thông chỉ khoảng 0.0012s.

### 2.4.3. Phân tích ảnh hưởng của độ trễ

Từ kết quả phân tích trong phần 4.1 và 4.2 của, chúng tôi rút ra những nhận xét sau về ảnh hưởng của độ trễ:

* Phân tích độ trễ trong mô phỏng (4.1): Mô phỏng cho thấy độ trễ trung bình của dữ liệu mới là 1.79\*10^-5 giây và độ trễ lớn nhất đạt 0.0012 giây khi tiến hành xử lý trên từng hàng một và gộp kết quả vào tập huấn luyện. Điều này cho thấy hệ thống có khả năng xử lý dữ liệu mới với độ trễ rất thấp, đảm bảo tính nhanh chóng và hiệu quả trong việc cập nhật và tích hợp dữ liệu.
* So sánh với thời gian thực (4.2): Khi so sánh với thời gian thực tế trong việc đăng ký khóa học, quy trình phân tích cho thấy độ trễ thực tế là khoảng 5 giây. Quá trình này bao gồm việc tạo và sắp xếp danh sách tổng hợp các thời điểm đăng ký, cũng như tính khoảng thời gian giữa các lần đăng ký liên tiếp để đánh giá tần suất cập nhật dữ liệu. Độ trễ này, mặc dù cao hơn so với mô phỏng, vẫn cho thấy khả năng đáp ứng kịp thời đối với nhu cầu đăng ký khóa học, đặc biệt trong bối cảnh yêu cầu thời gian thực hoặc gần như thời gian thực.

Nhìn chung, kết quả từ cả hai phần phân tích cho thấy hệ thống có khả năng xử lý và cập nhật dữ liệu với độ trễ thấp, phản ánh hiệu suất tốt trong việc duy trì chất lượng dữ liệu và đáp ứng nhanh chóng đối với các yêu cầu cập nhật.

### 2.4.4. Sử dụng các công cụ giám sát

Để theo dõi và giám sát độ trễ của dữ liệu hiệu quả, có nhiều công cụ và giải pháp có thể được sử dụng, tùy thuộc vào môi trường cụ thể (ví dụ: ứng dụng web, dịch vụ đám mây, hệ thống IoT, v.v.) và yêu cầu về chức năng. Dưới đây là một số công cụ phổ biến:

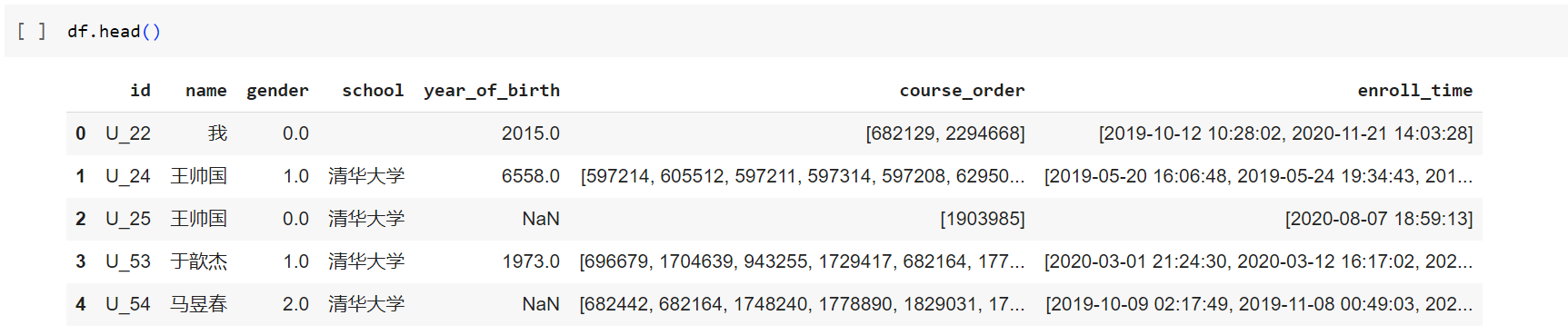
* Prometheus: Là một hệ thống giám sát và cảnh báo mã nguồn mở, Prometheus thường được sử dụng để thu thập và lưu trữ số liệu theo thời gian thực. Nó có thể giám sát độ trễ cùng với nhiều chỉ số hiệu suất khác, giúp phát hiện vấn đề trong gần như thời gian thực
* Grafana: Là một nền tảng để phân tích và trực quan hóa dữ liệu. Grafana thường được sử dụng kết hợp với Prometheus để trực quan hóa dữ liệu giám sát, bao gồm cả độ trễ. Nó cung cấp giao diện đồ họa mạnh mẽ với khả năng tùy chỉnh cao.

## 2.5. TÍNH HỢP LỆ (VALIDITY)

### 2.5.1 Phân tích thống kê

#### 2.5.1.1 Overview

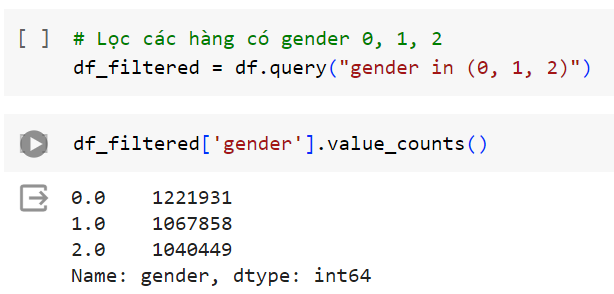
Nhìn lại một chút về bộ dữ liệu:



**2.32.** Dữ liệu entities/user.json

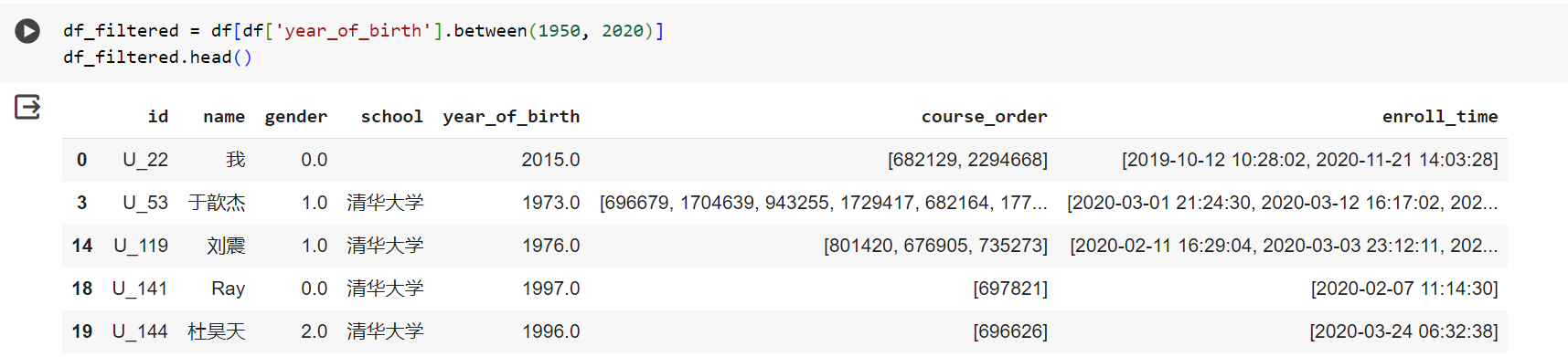
Vì chúng tôi đánh giá tính hợp lệ của dữ liệu sử dụng phân tích thống kê thế nên chúng tôi tập trung phân tích chủ yếu ở 2 trường gender và year\_of\_birth. Do đó chúng tôi tiến hành các thao tác sau:

Lọc gender chỉ lấy 3 giá trị là 0, 1, 2.



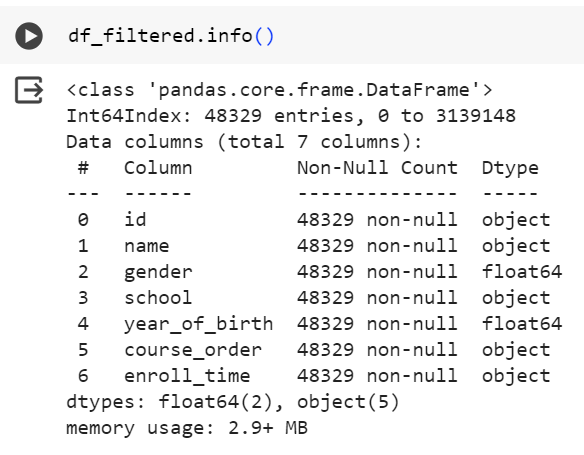
**Hình 2.33.** Đoạn code và kết quả lọc gender

Lọc year\_of\_birth trong khoảng từ (1950, 2020), điều này đồng thời giúp loại bỏ những missing value.



**Hình 2.34.** Đoạn code và kết quả lọc year\_of\_birth

Sau khi thực hiện các thao tác trên, chúng tôi thu được bộ dữ liệu mới như sau:

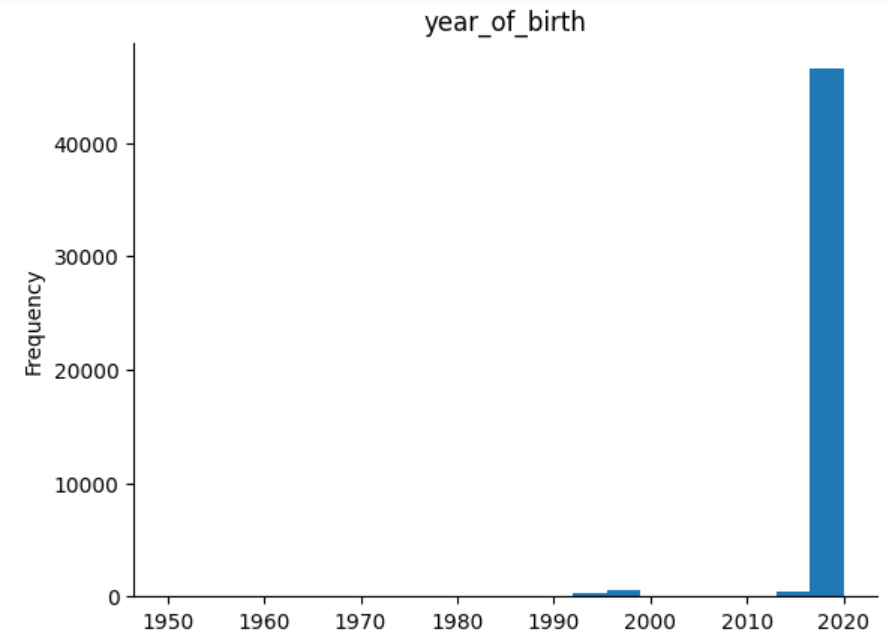
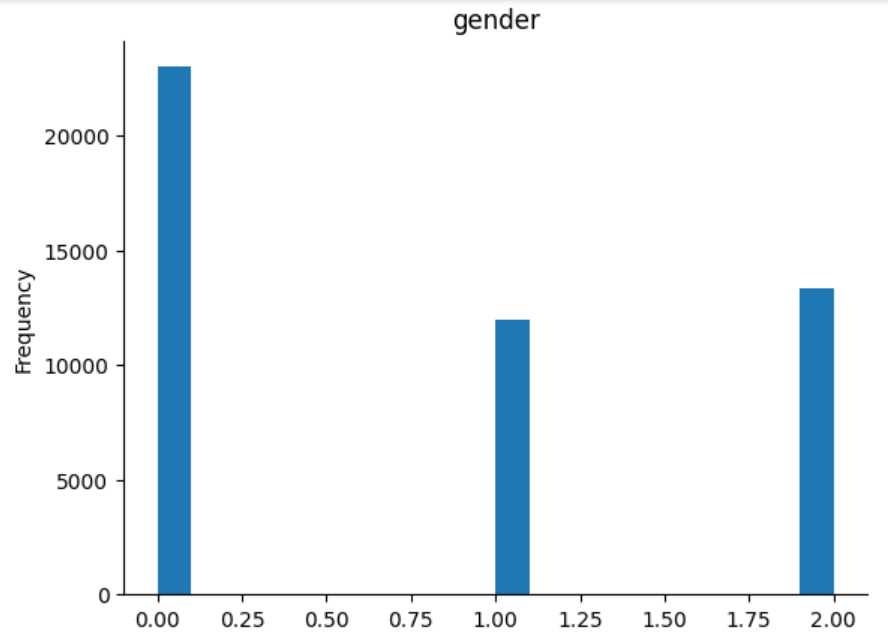


**Hình 2.35.** Đoạn code hiển thị thông tin bộ dữ liệu sau khi thực hiện các phép lọc

Chúng tôi tiến hành phân tích tính hợp lệ trên bộ dữ liệu có 3139148 mẫu như hình 2.22

#### 2.5.1.2 Kiểm tra tính chính quy

Đầu tiên, chúng tôi tiến hành sử dụng biểu đồ hist để biểu diễn phân bố cũng như giá trị của trường gender và year\_of\_birth



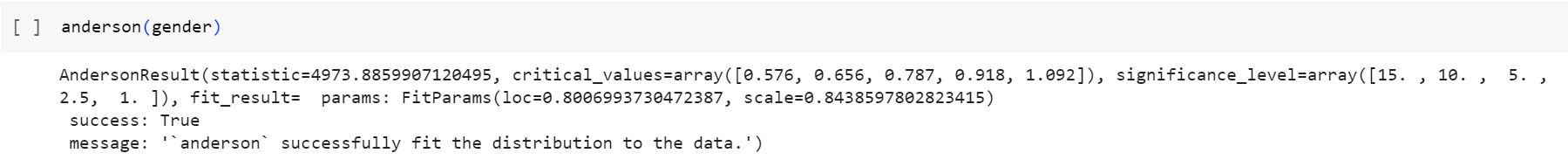
**Hình 2.36.** Đồ thị hist thể hiện phân bố và miền giá trị của trường gender và year\_of\_birth

Nhận xét: dựa vào đồ thị hist ở trên của trường gender và year\_of\_birth có thể nhận thấy phân bố của cả 2 không tuân theo phân bố chuẩn, cụ thể:

* Đối với trường gender: sự phân bố của class 0 so với class 1 và 2 là có sự chênh lệch đáng kể.
* Đối với trường year\_of\_birth: phần lớn năm sinh tập trung chủ yếu vào năm 2020.

Để kiểm tra lại nhận định: hai trường gender và year\_of\_birth không tuân theo phân phối chuẩn, chúng tôi tiến hành sử dụng kiểm tra Anderson-Darling. Thử nghiệm Anderson-Darling là một thử nghiệm mức độ phù hợp cho phép kiểm soát giả thuyết rằng sự phân bố của một biến ngẫu nhiên quan sát được trong một mẫu tuân theo một phân phối lý thuyết nhất định. Đặc biệt, nó cho phép chúng ta kiểm tra xem phân phối thực nghiệm thu được có tương ứng với phân phối chuẩn hay không.

Chạy kiểm tra Anderson-Darling trên 2 trường gender và year\_of\_birth, chúng tôi thu được kết quả như sau:

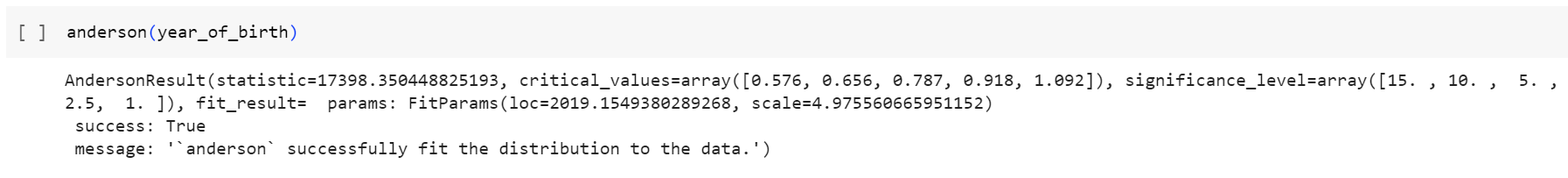


**Hình 2.37.** Đoạn code và kết quả chạy Anderson-Darling trên trường gender

Nhận xét:

* Giá trị tới hạn của α = 0,01 là 1,021 . Bởi vì thống kê kiểm tra (4973.886) lớn hơn giá trị tới hạn này nên kết quả có ý nghĩa ở mức ý nghĩa 0,01.
* Giá trị tới hạn của α = 0,025 là 0.918 . Bởi vì thống kê kiểm tra (4973.886) lớn hơn giá trị tới hạn này nên kết quả có ý nghĩa ở mức ý nghĩa 0,01.
* Tương tự với các mức α = 0.05, 0.1, 0.15.

→ Chúng ta có thể nhận thấy các kết quả kiểm định đều có ý nghĩa ở mọi mức ý nghĩa, có nghĩa là chúng ta sẽ bác bỏ giả thuyết không của kiểm định bất kể chúng ta chọn sử dụng mức ý nghĩa nào. Vì vậy, chúng ta có đủ bằng chứng để nói rằng dữ liệu mẫu không có phân phối chuẩn.



**Hình 2.38.** Đoạn code và kết quả chạy Anderson-Darling trên trường year\_of\_birth

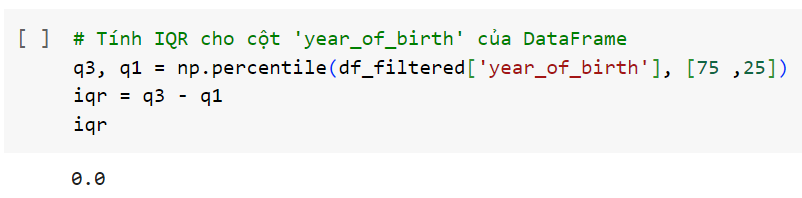
Nhận xét:

* Giá trị tới hạn của α = 0,01 là 1,021 . Bởi vì thống kê kiểm tra (17398.35) lớn hơn giá trị tới hạn này nên kết quả có ý nghĩa ở mức ý nghĩa 0,01.
* Giá trị tới hạn của α = 0,025 là 0.918 . Bởi vì thống kê kiểm tra (17398.35) lớn hơn giá trị tới hạn này nên kết quả có ý nghĩa ở mức ý nghĩa 0,01.
* Tương tự với các mức α = 0.05, 0.1, 0.15.

→ Chúng ta có thể nhận thấy các kết quả kiểm định đều có ý nghĩa ở mọi mức ý nghĩa, có nghĩa là chúng ta sẽ bác bỏ giả thuyết không của kiểm định bất kể chúng ta chọn sử dụng mức ý nghĩa nào. Vì vậy, tương tự chúng ta có đủ bằng chứng để nói rằng dữ liệu mẫu không có phân phối chuẩn.

#### 2.5.1.3. Kiểm tra ngoại lệ

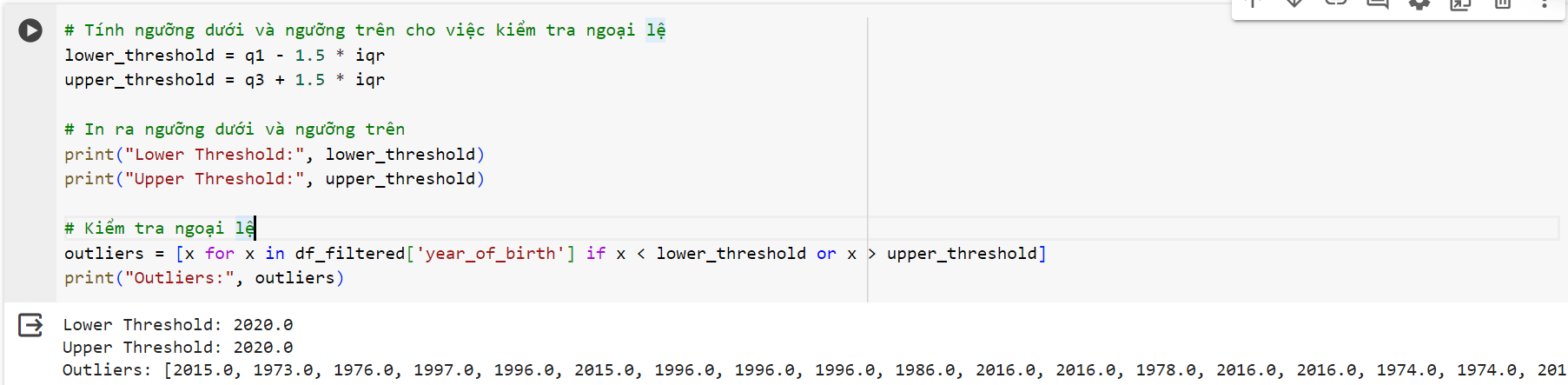
Chúng tôi tiến hành sử dụng phương pháp IQR (Interquartile Range) để xác định các giá trị ngoại lệ trong trường year\_of\_birth. Chúng tôi chọn ngưỡng q1 = 0.25 và q3 = 0.75, thu được kết quả như sau:



**Hình 2.39.** Đoạn code và kết quả tính IQR cho trường year\_of\_birth

Nhận xét: sự phân tán của dữ liệu trong phần trung tâm của phân phối, bỏ qua 25% dữ liệu ở cả hai đầu của phân phối bằng 0 (IQR = 0) tức là dữ liệu 50% ở phần trung tâm không có sự biến động.

Tiếp theo, chúng tôi tiến hành xác định các giá trị ngoại lệ:



**Hình 2.40.** Đoạn code, kết quả tính ngưỡng trên và dưới và xác định các giá trị ngoại lệ

Nhận xét: vì giá trị IQR = 0, chính vì thế upper threshold và lower threshold tương ứng bằng giá trị q3 và q1 và cùng bằng 2020. Để kiểm tra lại kết quả, chúng ta cùng nhìn lại

hình 2.36. Đồ thị hist thể hiện phân bố và miền giá trị của trường year\_of\_birth. Từ biểu đồ hist ta thấy năm sinh phần lớn là năm 2020. Do đó các giá trị năm sinh khác được xem là giá trị ngoại lệ của dữ liệu.

#### 2.3.1.4. Phân tích tương quan

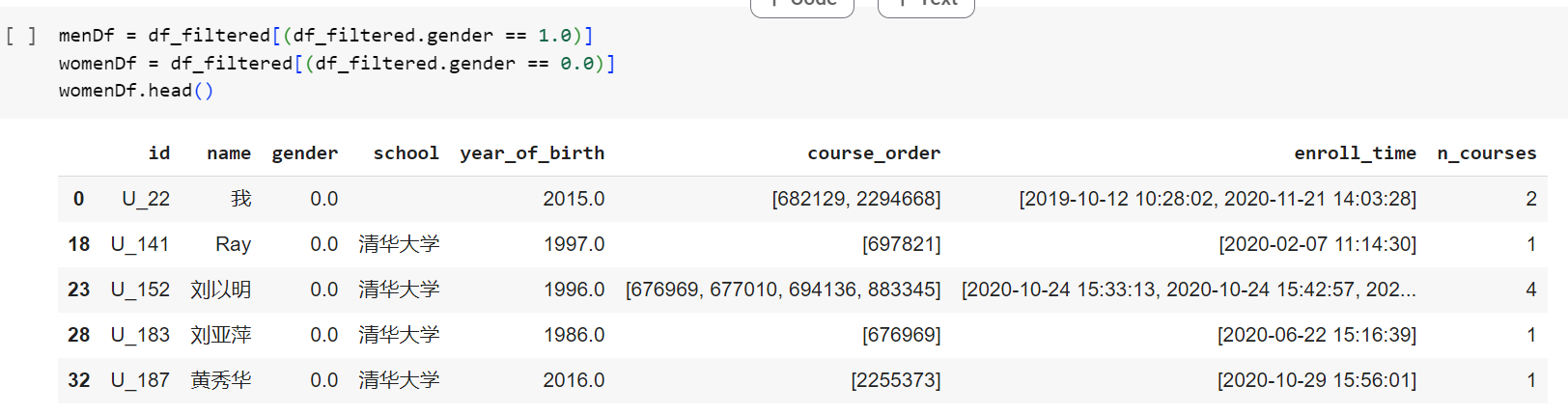
Để tiến hành phân tích tương quan, chúng tôi nhận thấy 2 trường gender và year\_of\_birth có thể không đủ để chúng tôi đi đến kết luận, do đó chúng tôi thực hiện tạo thêm cột n\_courses (số lượng khóa học đăng ký).



**Hình 2.41.** Đoạn code và kết quả tạo thêm cột n\_courses

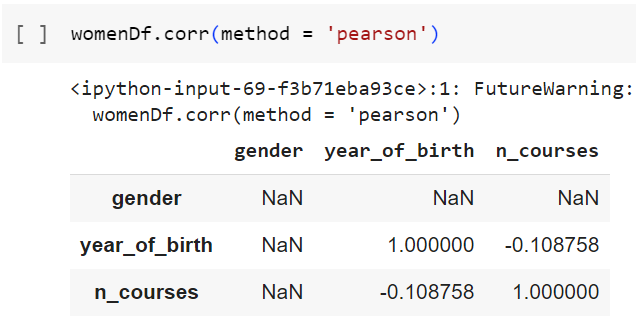
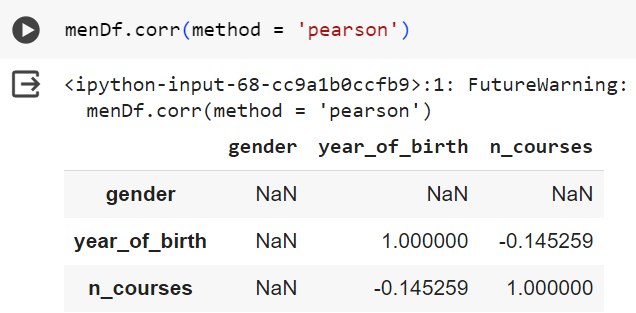
Tương tự như phần tính nhất quán, chúng tôi sử dụng hệ số tương quan Pearson để biểu diễn mối quan hệ giữa 2 biến từ đó đánh giá tính hợp lệ của dữ liệu.

Để đảm bảo kết quả không bị sai khớp do sự khác biệt giữa nam (giả sử là 1) và nữ (giả sử là 0), tập dữ liệu được chia thành hai dataframe: một chứa tất cả các thông tin liên quan đến nam và một dataframe khác chỉ chứa các thông tin liên quan đến nữ.



**Hình 2.42.** Đoạn code và kết quả chia dataframe

Sau đó, chúng tôi tiến hành tính hệ số tương quan Pearson trên data chỉ chưa nam và data chỉ chứa nữ và thu được kết quả như sau:



**Hình 2.43.** Đoạn code và kết quả tính hệ số Pearson ở dataframe nam (trái) và nữ (phải)

Nhận xét: Dựa vào hệ số tương quan Pearson đối với giới tính nam và nữ, có thể thấy:

* n\_courses và year\_of\_birth không có tương quan.
* n\_courses và gender có mối tương quan là NaN.

Tuy có sự bất thường nhưng nếu xét trên thực tế, giới tính không ảnh hưởng đến năm sinh cũng như số lượng khóa học; năm sinh có thể có ảnh hưởng đến khóa học vì mỗi một thế hệ có thể có những trend học khác nhau. Chính vì thế dữ liệu vẫn đảm bảo tính hợp lý.

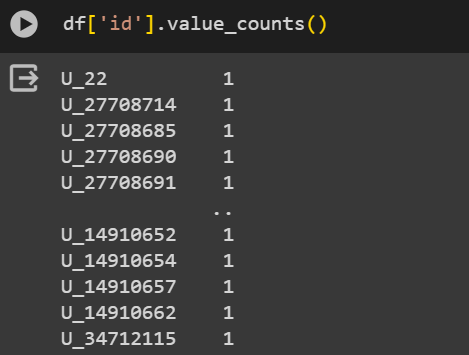
## 2.6. TÍNH DUY NHẤT (UNIQUENESS)

Đối với tác vụ phân tích thống kê, nội dung thực hiện tương tự như kiểm tra tính nhất quán. Do đó, để tránh lặp lại nội dung thì chúng tôi sẽ không nhắc đến trong mục này.

#### **2.6.1 Xác minh dữ liệu**

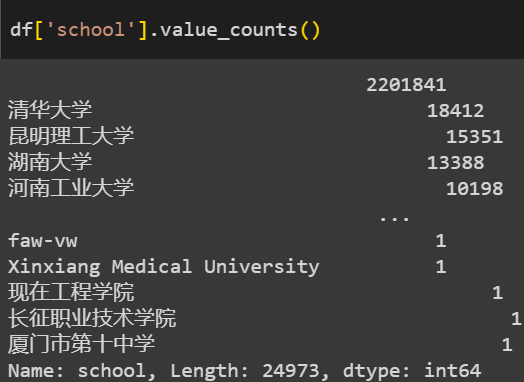
Với việc xác minh dữ liệu, ta tiến hành kiểm tra các giá trị không duy nhất ở mỗi cột của bộ dữ liệu entities/user.json. Công cụ sử dụng chủ yếu là hàm value\_counts() để xem xét những giá trị không duy nhất có trong dữ liệu.

Đối với dữ liệu ở cột ‘id’, tất cả các giá trị đều khác nhau (mang tính duy nhất), từ đó bộ dữ liệu được đảm bảo tính duy nhất của mỗi mẫu dữ liệu



**Hình 2.44.**

Những cột khác không có tính duy nhất.



**Hình 2.45.**

# 