17. Tính đoán số hoc

Chúng ta đã đi được một đoạn đường rất dài khi chinh phục các cấu trúc dữ liệu của pandas đặc biệt là Series và Dataframe. Từ việc khởi tạo các pandas structures đến Sắp xếp và định hình lại dữ liệu và gần đây nhất là nhóm dữ liệu để thực hiện các phép chuyển đổi, lọc dữ liệu và cả các phương pháp selection và slice. Tuy nhiên tôi vẫn chưa đề cập đến một mảnh ghép quan trọng khác đó là thực hiện các phép toán số học trong Series và Dataframe.

Về các phép toán số học, chúng ta đã được học các phép toán số học với các kiểu dữ liệu cơ sở trong python như **array/mattrix** trong numpy vậy chúng có gì khác biệt so với cách thức hoạt động trong pandas structures? Bài hôm nay sẽ làm rõ vấn đề này.

Để thuận tiện chúng ta sẽ sử dụng bộ dữ liệu sau cho bài học này, ta sử dụngbộ dữ liệu.

```
>>> exchange = pd.read_csv("Exchangerates.csv")

>>> exchange.head()

  LOCATION INDICATOR SUBJECT MEASURE FREQUENCY  TIME     Value

0     AUS     EXCH    TOT  NATUSD         A  2000  1.724827

1     AUS     EXCH    TOT  NATUSD         A  2001  1.933443

2     AUS     EXCH    TOT  NATUSD         A  2002  1.840563

3     AUS     EXCH    TOT  NATUSD         A  2003  1.541914

4     AUS     EXCH    TOT  NATUSD         A  2004  1.359752

>>>
```

Ta sẽ reshape lại dữ liệu với index là TIME, columns là LOCATION và giá trị sẽ là Value (x value mới mua được 1 USD). Và quan tâm đến hai đồng AUS và KOR.

```
>>> exchange = exchange.pivot(index='TIME',columns='LOCATION',values='Value') [['AUS','KOR']]

>>> exchange.head()

LOCATION       AUS         KOR

TIME

2000      1.724827  1130.957500

2001      1.933443  1290.994583

2002      1.840563  1251.088333

2003      1.541914  1191.614167

2004      1.359752  1145.319167

>>>
```

Kĩ thuật broadcasting khi thực hiên các phép toán học

Các phép toán số học với một số (scalar) như +,-,*,/,** (mũ) thì số đó sẽ được broadcasted đến từng phần tử trong dataframe để thực hiện phép toán số học.

Ví dụ.

+ Tất cả các đồng bị giảm giá trị đi một nửa tức là giá trị của các cột sẽ bị * 2

```
>>> (exchange * 2).head()
LOCATION       AUS         KOR
TIME
2000      3.449654  2261.915000
2001      3.866886  2581.989166
2002      3.681126  2502.176666
2003      3.083828  2383.228334
2004      2.719504  2290.638334
>>>
```

Kĩ thuật broadcasting khi thực hiên các phép toán học

+ Tất cả các đồng tăng bình phương lần tức là giá trị các cộ sẽ mũ 0.5.

```
>>> (exchange ** 0.5).head()
LOCATION       AUS       KOR
TIME
2000      1.313327  33.629712
2001      1.390483  35.930413
2002      1.356674  35.370727
2003      1.241738  34.519765
2004      1.166084  33.842564
>>>
```

Yêu cầu thay vì quy đổi cần bao nhiêu đồng AUS hay KOR để mua được một đồng USD như hiện tại ta thay bằng bao nhiêu đồng AUS có thể mua được một đồng KOR? Hãy tạo ra một dataframe chứa hai cột. Một cột là KOR giá tri sẽ là 1 đồng và cột còn lại là AUS thể hiện số lượng đồng AUS để có thể mua được một đồng KOR.

Có nhiều cách để giải nhưng tôi làm theo cách sau để mô tả cho bài viết này. Đó là sử dụng: exchange/exchange['KOR'].

```
>>> (exchange/exchange['KOR']).head()
     AUS  KOR  2000  2001  2002  2003  2004  2005  2006  2007  2008  2009  \
TIME
2000  NaN  NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
2001  NaN  NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
2002  NaN  NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
2003  NaN  NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
```

```
2004  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN  NaN


    2010  2011  2012  2013  2014  2015  2016  2017
TIME
2000   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
2001   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
2002   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
2003   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
2004   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
>>>
```

Ý tưởng cách làm là không sai nhưng dataframe không thể thực hiện được phép chia đó do column labels không match nên tất cả các giá trị là NaN. Thay vào đó ta sử dụng .divide() với option axis=0 (hoặc axis='row'). Phương thức .divide() cung cấp nhiều tùy chọn để thực hiện phép chia. Về cách thức hoạt động nó broadcast giá trị của Series exchange['KOR'] đến tất cả các hàng ứng với indexes để thực hiện phép chia.

```
>>> exchange.divide(exchange['KOR'],axis=0).head()
LOCATION      AUS  KOR
TIME
2000     0.001525  1.0
2001     0.001498  1.0
2002     0.001471  1.0
2003     0.001294  1.0
2004     0.001187  1.0
>>>
```

Các bạn có thể tham khảo thêm các phương thức khác như được đề cập trong bảng dưới đây. Các phương thức này rất có ích khi kết hợp với fill_value sẽ rất có ích để xử lý các giá trị NaN trong dataframe.

Toán tử	Phương thức
+	.add()
-	.sub(), .subtract()
*	.mul(), .multiply()

/	.div(), .truediv(), .divide()
//	.floordiv()
%	.mod()
**	.pow()

Ngoài ra còn có phương thức rất thú vị đó là .pct_change() để tính toán thay đổi % theo thời gian. Sự thay đổi này được tính (giá trị row hiện tại – giá trị row trước đó)/ (giá trị row trước đó)

```
>>> exchange.pct_change() * 100
LOCATION       AUS       KOR
TIME
2000           NaN       NaN
2001     12.094894  14.150583
2002     -4.803865  -3.091125
2003    -16.225959  -4.753794
2004    -11.814018  -3.885066
2005     -3.697660 -10.582421
2006      1.412782  -6.769361
2007    -10.007734  -2.674226
2008     -0.242245  18.594356
2009      7.550131  15.868959
2010    -14.976731  -9.465594
2011    -11.071413  -4.132037
2012     -0.377735   1.640245
2013      7.252219  -2.806811
2014      7.097601  -3.826275
2015     19.986875   7.426360
2016      1.061085   2.588140
2017     -3.007403  -2.586000
>>>
```

Kết luận:

Qua bài học này chúng ta hiểu hơn về cách thức broadcasting hoạt động khi thực hiện các phép

toán số học trong cấu trúc dữ liệu trong pandas. Bên cạch đó tôi cũng giới thiệu thêm về một phưng thức .pct_change() khá thú vị giúp tính nhanh sự thay đổi % theo thời gian một cách nhanh chóng.