

Data Science - Practice 2

Problem

country 데이터는 국가별 지표와 대륙 정보를 담고 있는 데이터 프레임입니다.

The data frame called 'country' contains various national indicators and continental information.

variable	의미	mean
code	국가 코드	country's code
country_name	국가 이름	country's name
continent	대륙	continent
GDP	1인당 국내총생산	Gross Domestic Product per capita
life_expect	기대수명	life expectancy
population	인구 수	population
CO2	CO2 배출량 (추정치)	CO2 emission quantity (estimated)
battle_death	전투 중 사망자 (100,000명당)	a death in battle (per 100,000)
child.per.women	여성 1명당 아이의 수	number of children per woman
programmable.aid	국가별 프로그램 원조	national program aid

Loading data into R

다음 코드를 입력하여 데이터를 불러올 수 있습니다. 이번 과제에서는 'continent'와 'GDP'라는 두 가지 변수를 중점적으로 사용합니다.

If you type in the following code, you can import the necessary data. This practice assignment focuses on 'continent' and 'GDP' variables.

```
load(url('https://github.com/hbchoi/SampleData/raw/master/country.RData'))
```

< Question 1 >

country 데이터의 가장 위의 값 5개와 가장 밑의 값 5개를 각각 출력해보세요.

Try to output the top 5 values and bottom 5 values of the country data, respectively.

Sample Result

```
##   code country_name      continent   GDP life_expect population    C02
## 1  afg  Afghanistan        Asia  1757      61.22   35400000   8660
## 2  alb    Albania          Europe 11357      78.12    2890000   4540
## 3  dza    Algeria          Africa 13940      77.40   40600000  148000
## 4  arg  Argentina South America 18645      76.54   43500000  200000
## 5  arm    Armenia          Asia   8159      75.37    2940000   5180
## 6  aus  Australia        Oceania 44606      82.50   24300000  413000
##   battle_death child.per.woman programmable.aid
## 1           9.45           4.64       3663.25163
## 2           0.13           1.71       277.18911
## 3           3.41           2.78       108.27441
## 4           0.00           2.29       59.06856
## 5           0.00           1.63       373.09101
## 6           0.00           1.85       850.56700

##   code country_name      continent   GDP life_expect population    C02
## 121 usa United States North America 53632      78.63  323000000  5310000
## 122 ury    Uruguay South America 20210      77.00    3420000   6690
## 123 ven  Venezuela South America 15219      75.32   29900000  163000
## 124 vnm    Vietnam        Asia   5896      74.39   93600000  184000
## 125 yem    Yemen          Asia   2619      68.07   27200000  10300
## 126 zwe    Zimbabwe        Africa 2489      60.52   14000000  10800
##   battle_death child.per.woman programmable.aid
## 121      0.3380           1.92       850.56700
## 122      0.0000           2.00       22.59572
## 123      0.0000           2.32       21.95411
## 124      0.0375           1.95      3672.18066
## 125      3.6000           4.00      512.27939
## 126      0.5100           3.76      553.58552
```

< Question 2 >

country 데이터에서 각 변수들의 데이터 타입을 확인하고, 10개 이하의 level로 이루어진 변수가 있다면 factor type으로 변환하세요.

Check the data type of each variable in the country data, and if there are any variables consisted of less than 10 levels, convert them to factor type.

Sample Result

```
##           code      country_name      continent      GDP
##   "character"    "character"      "factor"      "integer"
##   life_expect      population      C02      battle_death
##   "numeric"      "integer"      "numeric"      "numeric"
## child.per.woman programmable.aid
##   "numeric"      "numeric"
```

< Question 3 >

2번에서 factor로 수정된 변수의 요약값을 출력하여 데이터를 확인해보세요.

If you have a variable modified from question 2 to factor type, print out a summary of the variables to see how many data correspond to the factor.

Sample Result

```
##           Africa           Asia           Europe North America           Oceania
##             31             34             35             13             4
## South America
##              9
```

< Question 4 >

country 데이터에 있는 continent 변수의 levels를 아래와 같이 바꿔보세요.

Change the levels of the 'continent' variable in the country data as follows.

[Africa → AF, Asia → AS, Europe → EU, North America → NA, Oceania → OC, South America → SA]

Sample Result

```
##
## AC AS EU NA OC SA
## 31 34 35 13 4 9
```

< Question 5 >

country 데이터에서 GDP_group이라는 새로운 column을 생성할 것입니다. GDP_group은 국가의 GDP가 평균 GDP 보다 작으면 'LOW', 평균 이상이면 'HIGH' 입니다.

In the country data, we will generate a new column called 'GDP_group'. GDP_group is 'LOW' if the GDP of the country is smaller than the average GDP, and 'HIGH' if it is above average.

Sample Result

```
## .
## HIGH LOW
## 48 78
```

< Question 6 >

Find_continent()라는 함수를 선언하세요. 입력 값은 대륙의 코드이고, 출력값은 TRUE or FALSE입니다. 예를 들어 , Find_continent("AS") 를 입력하면 1번 국가부터 마지막 국가까지 Asia에 속하면 True, 속하지 않으면 False를 반환합니다. Find_continent("AS")와 Find_continent("EU")을 입력하고 결과값이 아래 예시와 같게 나오면 됩니다.

Declare function named 'Find_continent()'. Input values are continent code, Output values are TRUE or FALSE. For example, if you type in Find_continent("AS"), function check the every country that is belongs to Aisa or not from first to end, If the country belongs to Asia, it return TRUE, otherwise return FALSE. The result of Find_continent("AS") and Find_continent("EU") are follows.

Sample Result

```

## [1] TRUE FALSE FALSE FALSE TRUE FALSE FALSE TRUE TRUE TRUE FALSE FALSE
## [13] FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE
## [25] FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE
## [37] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE
## [49] FALSE FALSE TRUE TRUE TRUE FALSE TRUE FALSE FALSE TRUE TRUE TRUE
## [61] FALSE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE
## [73] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE
## [85] FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE TRUE FALSE FALSE
## [97] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE
## [109] TRUE FALSE FALSE TRUE FALSE TRUE TRUE FALSE FALSE TRUE FALSE FALSE
## [121] FALSE FALSE FALSE TRUE TRUE FALSE

## [1] FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE TRUE
## [13] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [25] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE
## [37] FALSE TRUE FALSE FALSE TRUE TRUE FALSE TRUE FALSE TRUE FALSE FALSE
## [49] TRUE TRUE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE
## [61] FALSE FALSE FALSE TRUE FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE
## [73] TRUE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE TRUE
## [85] FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE
## [97] TRUE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE TRUE
## [109] FALSE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE
## [121] FALSE FALSE FALSE FALSE FALSE FALSE

```