

R 프로그래밍

(15주차)

2016. 06. 11(토)

장운호

(ADP 002-0004)

목차

※ 지난 주 복습

I. RFM 분석 실습

II. 익명 함수

III. Environment

※ 장바구니 (Basket)

PARIS CROISSANT

서울역 파리크라상
106-85-18197 Tel:023127740
서울 용산 동자 43-205번지
POS:021 2016-05-20 06:20:56

ITEM	QTY	D/C	AMT
타르타르햄에그PC	1	0	7,000
그릴드토마토모짜렐라	1	0	7,000
(PET)생오렌지주스	1	0	6,800
과세금액			18,907
부가세			1,893
총매출액			20,800
합계			20,800
받은 돈			20,800
[카 드]			20,800

* 케익 및 냉장제품은 교환 및 환불이 불가합니다. 양해부탁드립니다.

<< 하나기업카드 승인 >>

[카드번호]
[승인금액]
[승인번호]

[고객용]

고객님~ 만족도 설문에 참여해 주세요♡
매 월 1,000분께 해피콘 3,000원 쏘 드립니다

위치

선호 상품
카테고리

지불수단 및
금액

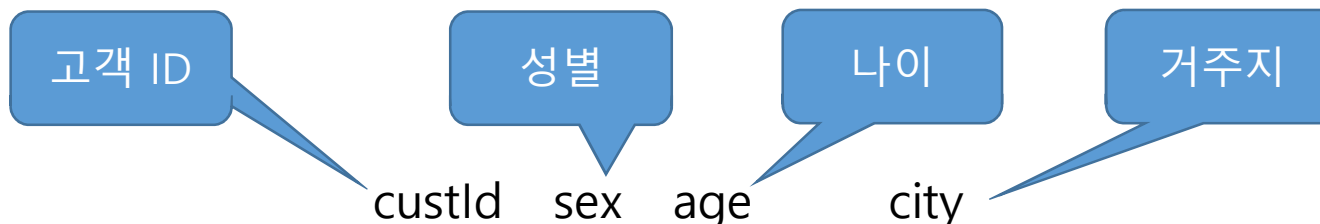
구매정보 외에
마일리지 포인트,
포인트 활용 여부
등도 활용 가능

시간

100

4

※ 고객 DB



	고객 ID	성별	나이	거주지
	custId	sex	age	city
1	C0001	F	23	Kyeongki
2	C0002	F	68	Seoul
3	C0003	F	21	Kyeongki
4	C0004	F	45	Seoul
5	C0005	F	50	Seoul
6	C0006	F	61	Seoul



I . RFM 분석 실습

1. RFM 등급 구분

계산의 단순화를 위해 RFM 각각의 Range를 5등분 하는 Quantile(백분위수)를 기준으로 등급을 구분하는 것으로 가정함.

- 백분위수(Quantile) 산출 함수 : quantile(데이터벡터, 백분위 지정 벡터)

구분	등급 구분	Point
최우수	상위 20%	5
우수	상위 20 ~ 40%	4
양호	상위 40 ~ 60%	3
보통	하위 20 ~ 40%	2
기타	하위 20%	1

2. RFM 등급부여

문자열을 명령문으로 해석하게 만들어주는 `eval`, `parse` 함수를 활용하여, 등급 구간의 변화를 자동반영하는 사용자 정의함수를 통한 Rating 적용

```
userRFM$R <- intervalGrade(userRFM, "userRFM", "recency", quantR )
userRFM$F <- intervalGrade(userRFM, "userRFM", "freq", quantF )
userRFM$M <- intervalGrade(userRFM, "userRFM", "monetary", quantM )

intervalGrade <- function(mainData, fileName, rfmName, quantileData) {
  forLength <- dim(mainData)[1] # loop횟수를 탐지
  results <- rep(0, forLength) # 결과 수용 객체 생성 (memory pre-location)
  for (i in 1:forLength) {
    data <- eval(parse(text=paste0(fileName,"$",rfmName)))[i] #평가하고자 하는 RFM 데이터 컬럼 할당
    if (data >= quantileData[1] && data < quantileData[2] ) { # if문을 활용한 평가 점수 rating
      results[i] <- 1
    } else if (data >= quantileData[2] && data < quantileData[3]) {
      results[i] <- 2
    } else if (data >= quantileData[3] && data < quantileData[4]) {
      results[i] <- 3
    } else if (data >= quantileData[4] && data < quantileData[5]) {
      results[i] <- 4
    } else { results[i] <- 5 }
  }
  return(results) # 결과값 반환
}
```


3. RFM Score 산출

RFM 점수는 3개 RFM 분야별 등급 점수에 가중치를 곱한 후, 합산하여 산출

- $\text{RFM score} = A * \text{R등급} + B * \text{F등급} + C * \text{M등급}$

단, RFM 분야별 가중치 A, B, C는 매출액을 중심으로 결정하는 것으로 가정함.

각 등급별 상위 20%의 매출액을 가중치로 결정

$$A = \frac{\text{R등급 상위 20\%의 매출액}}{\text{전체 매출액 합계}}$$

$$B = \frac{\text{F등급 상위 20\%의 매출액}}{\text{전체 매출액 합계}}$$

$$C = \frac{\text{M등급 상위 20\%의 매출액}}{\text{전체 매출액 합계}}$$



A+B+C의 합계가 1이 되도록 평준화

$$A = \frac{A}{A + B + C}$$

$$B = \frac{B}{A + B + C}$$

$$C = \frac{C}{A + B + C}$$

자료) 위메프 송동근 (DK's Portfolio)

4. RFM Score 평준화

고객간 비교를 편리하게 하기 위하여 RFM 점수를 1에서 100사이의 점수로 평준화(Normalizing)함.

- 100점 환산 : $\text{RFM점수} * 100 / \text{RFM 점수 Range의 최대값}$

등급	구분	평준화 점수	ABCD 등급
최우수	상위 20%	87.1	A
우수	상위 20 ~ 40%	69.8	B
양호	상위 40 ~ 60%	50.2	C
보통	하위 20 ~ 40%	37.3	D
기타	하위 20%	27.1	E



Ⅱ. 익명 함수

1. 익명 함수 (anonymous function)

명시적인 함수의 요건에 따른 정의를 생략하고, 분석가의 필요에 따라 단일 함수로는 실행하기 어려운 기능을 구현하기 위하여 명령문 내에서 직접 정의하여 사용하는 함수

- 주로 apply류의 함수群에서 사용됨.

[명시적인 함수(function)의 구현 및 호출]

함수의 구현

```
함수명 <- function(input) {  
  #기능 정의 (절차적인 명령어 문장)  
  output = input1 + input2  
  
  # return은 생략가능하나,  
  # 사용시는 괄호내에  
  # output 변수 지정 필수  
  return(output)  
} # input이 없는 함수도 있음.
```

함수의 호출

```
함수명 바로 옆에 인자를  
괄호에 넣어서 호출  
  
count5(datamat[,1])
```

[익명함수의 구현 및 활용]

```
count5 <- function(x) {  
  count = 0  
  for (i in 1:length(x)) {  
    if(x[i] == 5) count=count+1  
  }  
  return(count)  
}  
apply(datamat, 1, count5)  
apply(datamat, 1,  
  function(x) if (count5(x) > 2) {  
    "quite many"  
  } else {  
    "not quite many"  
  })
```



III. Environment

1. 함수(Function)의 3대 요소

모든 R의 함수는 아래 3가지 요소로 이루어져 있음.

formals()	함수의 Argument	<code>formals(count5)</code> <code>formals(intervalGrade)</code>
body()	함수의 실행 명령문 Set	<code>body(count5)</code> <code>body(intervalGrade)</code>
environment()	함수가 속해있는 environment	<code>environment(count5)</code> <code>environment(intervalGrade)</code>

2. Environment

R에서는 글로벌 환경에서 정의된 변수는 전역변수,
사용자 정의 함수 내에서 정의되거나 할당된 변수를 지역변수라고 칭함.

- 이름이 의미하는 바와 같이, 이들 각각의 변수 들은 유효범위(scope)가 다름.

```
datamat <-  
  matrix(sample(1:10, 100, replace=T),  
         nrow=5)  
  
count5 <- function(x) {  
  count = 0  
  for (i in 1:length(x) ) {  
    if(x[i] == 5) count=count+1  
  }  
  return(count)  
}  
  
apply(datamat, 1, count5)
```

지역변수

※ 사용자 정의 함수 실행시,
별도의 환경(Environment)을
별도로 만들어 실행이 되며,
이로 인해
사용자 정의함수 내에서 정의된 변수
들은 사용자 정의 함수 종료와 함께
소거됨.

전역변수

※ env = .GlobalEnv

3. Lexical Scoping

다음 R코드에서 g(2)의 결과 값은?

```
a <- 1
b <- 2

f <- function(x) {
  a*x + b
}

g <- function(x) {
  a <- 2
  b <- 1
  f(x)
}

g(2) # 정답은?
```

```
a <- 1
b <- 2

f <- function(a, b) {
  return( function (x) {
    a*x + b
  })
}

g <- f(2, 1)

g(2) # 정답은?
```


End of Document.

감사합니다.