



Day18; 20220928

📅 날짜	@2022년 9월 28일
📅 유형	@2022년 9월 28일
🏷️ 태그	

GitHub - u8yes/R

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

<https://github.com/u8yes/R>

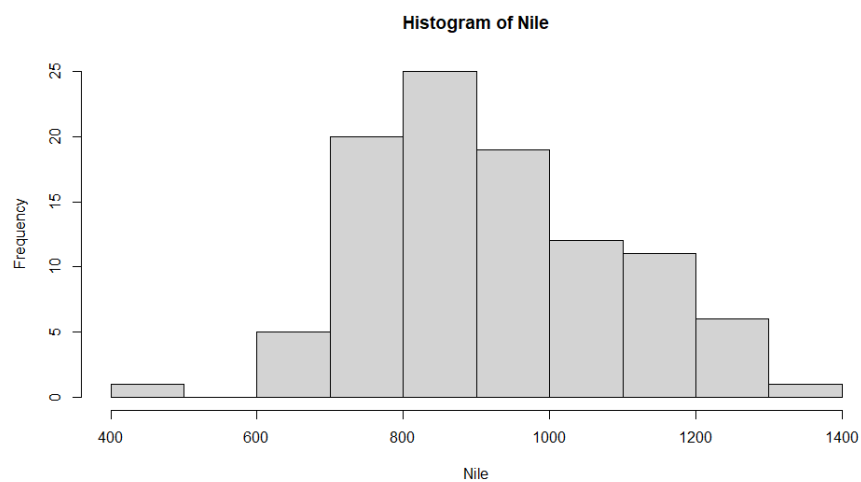
u8yes/R



1 Contributor 0 Issues 0 Stars 0 Forks

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/28a19589-8bb9-46f3-9bb6-7c8ed75c304f/chap01_Basic.r

빈도수가 가장 많이 하는 데이터 분석이다.



예) 대통령 사전 출구 조사,

hist()모양이 여러 개 나오면서 오버로딩 됨.

```
# 단계2 : 밀도(density)를 기준으로 히스토그램 그리기
hist(Nile, freq = F)
```

밀도 - 공간 안에 들어갈 분포

하나의 자료형을 정의할 때(선언할 때) 같은 이름으로 메서드로 정의할 수 없다.

하지만 단, 2가지 조건하에서는 둘 중에 하나 이상만을 만족할 경우에는 메서드를 재정의 할 수 있다.(상속 X)

매개변수의 자료형 타입이 서로 다르거나 갯수가 다를 때 하나 이상만을 만족시킨다면 동일한 이름으로 만든 하나의 참조 자료형 안에 동일한 이름으로 메서드 재정의 할 수 있다.

```
# 단계2 : 밀도(density)를 기준으로 히스토그램 그리기
hist(Nile, freq = F, ) # Boolean형태로 F(false), 곧 대문자로만 작성해야됨,
소문자로 쓰면 에러가 나게 됨. T는 당연히 True이다. 자바와 이런 차이가 있다.
심지어 순서를 바꿔도 상관없다.
```

함수 정의하면서 정의할 수 있는 매개변수 혹은 파라미터만 선언하면 자료형 없이 **순서없이** 선언해도 상관없다.

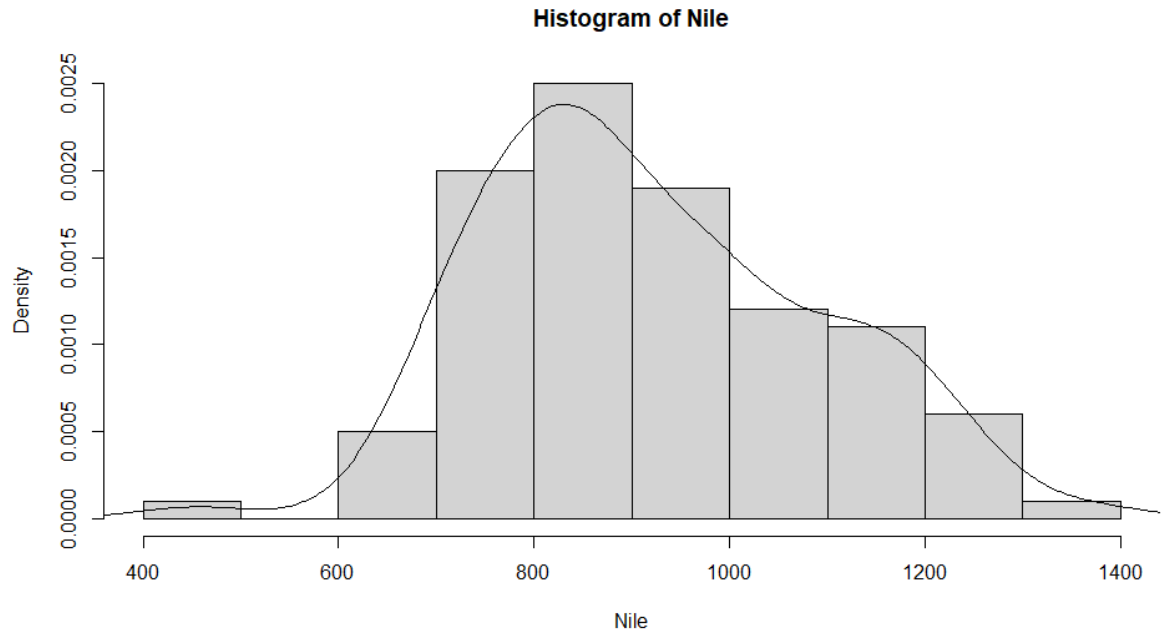
```
hist(x, ...)

## Default S3 method:
hist(x, breaks = "Sturges",
     freq = NULL, probability = !freq,
     include.lowest = TRUE, right = TRUE, fuzz = 1e-7,
     density = NULL, angle = 45, col = "lightgray", border = NULL,
     main = paste("Histogram of" , xname),
     xlim = range(breaks), ylim = NULL,
     xlab = xname, ylab,
     axes = TRUE, plot = TRUE, labels = FALSE,
     nclass = NULL, warn.unused = TRUE, ...)
```

매개변수도 안 넣을 거라면 순서대로만 값을 넣어주면 된다.

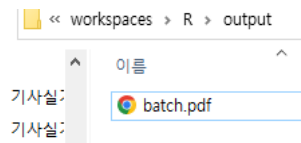
선으로 보기

```
0
1 lines(
2   density(Nile)
3 )
4
```



```
par(mfrow=c(1, 1)) # c - combine # plots창 영역에 1개 그래프만 표시하겠다
pdf("D:/heaven_dev/workspaces/R/output/batch.pdf")
hist(rnorm(20)) # random하게 정규분포를 뽑아줘.
dev.off()
```

```
5 # 히스토그램을 파일에 저장하기
6 par(mfrow=c(4, 4)) # c - combine # plots창 영역에 1개 그래프만 표시하겠다
7 pdf("D:/heaven_dev/workspaces/R/output/batch.pdf")
8 hist(rnorm(20)) # random하게 정규분포를 뽑아줘.
9 dev.off()
10
```



- 자바

이름을 지을 때 숫자로 정의하면 안 된다.

특수문자 중 2개만 변수이름으로 '_', '\$' 사용 가능하다.

파일이 만들어질 때 내부클래스를 만들면 A\$.B.class 파일이 생성된다.

```
class A {
    class B{}
}
```

Num, num은 서로 다른 변수이름이다.

- R

에서는 변수 선언 때 2번째 글자부터 '.' 사용 가능하다.

예) install.packages 라는 이름도 만들 수 있다.

데이터 유형(Data Type)

➤ R은 변수 선언 시 별도의 Type을 선언하지 않음

유형(Type)	값(value)	예
숫자형(Numeric)	정수, 실수	10, 25, 55.23
문자형(Character)	문자, 문자열	"홍길동"
논리형(Logical)	참, 거짓	TRUE 또는 T, FALSE 또는 F
결측 데이터	결측치, 비슷자	NA(Not Available), NaN(Not a Number)

NA(Not Available), NaN(Not a Number)

변수는 변하는 숫자다, 항상 저장해 놓는 것이 다르니<수학용어에서 가져옴>

boolean = 3.14가 들어가있음.

```
82 string <- "홍길동" # 문자형
83 boolean <- True # 진리값 : TRUE(T) / FALSE(F) [무조건 대문자]
84 boolean
```

```
Error: object 'True' not found
```

String 으로 곱해줄 수가 없어서

```
5 # 숫자 원소를 문자
6 y <- c(1, 2, "3") #
형변환 해줌.
```

```

15 # 숫자 원소를 문자 원소로 형변환
16 y <- c(1, 2, "3") # 서로 다른형을 저장할 수 없다. 숫자
   형변환 해줌.
17 y
18
19 result <- x * 3
20 result
21
22
23 result <- y * 5
24

```

```

> y
[1] "1" "2" "3"
> result <- y * 5
Error in y * 5 : non-numeric argument to binary operator
>

```

그래서 y가 문자 형태로 되어있으니깐 정수형으로 as바꿔준다.

그래서 바꿔주니 5 10 15

```

24
25 result <- as.integer(y) * 5
26 result
27

```

```

> result
[1] 5 10 15
>

```

수학백과

허수

제곱하여 -1 이 되는 수를 **허수단위**라 하고 기호로 i 와 같이 나타낸다.

두 실수 a 와 b 에 대하여 $a + bi$ 꼴의 수를 **복소수**라고 하며, $b = 0$ 인 경우에 복소수 $a + bi$ 는 **실수**이다. 만약 $b \neq 0$ 이면 복소수 $a + bi$ 를 **허수**라고 한다. 특히 $a = 0$ 인 경우, 복소수 $a + bi$ 는 **순허수**(pure imaginary number)라고 한다.

영어 imaginary number는 보통 순허수를 뜻하는 용어로 쓰인다.

chap01_Basic.R

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d858be97-434c-4b6a-a3ae-d46422e5784d/chap01_Basic.R

```
##### Day17; 20220927 #####
# chap01_Basic : 주식문 기초 #

#####
# Chapter01. R 설치와 개요
#####

# 주요 단축키
# - script 실행 : ctrl+Enter, ctrl+R, Ctrl + alt + s(전체파일저장)
# - script 저장 : ctrl+s

# Library 등록.
# - C:/Program Files/R/R-4.1.2/etc/Rprofile.site 파일에 .libPaths("C:/myRLib/Library") 추가 혹은
# - 사용자 변수 새로 만들기 : 변수 이름 -> R_LIBS, 변수 값 : C:\Users\user\Documents\R\win-library\4.1 으로 셋팅.

print("Hello, R!!!") # ctrl+Enter

## 패키지와 Session 보기

# R Package 보기
dim(available.packages()) # dimension 차원, 18688 패키지 수가 있고 17개의 정보를 가지고 있다.

available.packages()

# 패키지 사용법
install.packages("stringr") # 패키지 설치
library("stringr") # 메모리에 로딩 : "" 생략 가능
# install.packages("ichimoku")
search() # 패키지 메모리 로딩 확인

# 패키지 제거
remove.packages("stringr")
search()

# R session 보기
sessionInfo()

data()

# 기본 데이터 셋으로 히스토그램 그리기
# 단계1 : 빈도수(frequency)를 기준으로 히스토그램 그리기

hist(Nile) # hist - 히스토그램의 약자

##### Day18; 20220928 #####

# 단계2 : 밀도(density)를 기준으로 히스토그램 그리기
hist(Nile, freq = F) # Boolean형태로 F(false), 곧 대문자로만 작성해야됨, 소문자로 쓰면 에러가 나게 됨. T는 당연히 True이다. 자바와 이런 차이가 있다. 심지어 순
# 단계3 : 단계2의 결과에 분포곡선(line)을 추가

?hist # 기능 설명

lines(density(Nile))

# 히스토그램을 파일에 저장하기
par(mfrow=c(1, 1)) # c - combine # plots창 영역에 1개 그래프만 표시하겠다
pdf("D:/heaven_dev/workspaces/R/output/batch.pdf")
hist(rnorm(20)) # random하게 정규분포를 뽑아줘.
dev.off()

## 4. 변수와 자료형
# 변수 사용 예
age <- 25 # '<-' 안 넣고 '=' 넣어도 상관없다. 단지 가독성때문에 '<-' 사용
age
age <- 35
age

# 변수.멤버 형태로 변수 선언 예

var1 <- 50
var2 <- 100

goods.model <- "lg-320" # 상품 모델명명
```

```

goods.name <- "냉장고"
goods.price <- 850000
goods.desc <- "등급 최고 품질질"
rm(goods.model, goods.name, goods.price, goods.desc)

# 스칼라 변수 사용 예(스칼라 - 변수에 하나의 값만 담긴다.)
name <- "홍길동"
name

# 자료형
int <- 20 # 숫자형(정수) # 저장할 때 20.0으로 저장됨.
double <- 3.14 # 숫자형(실수)
string <- "홍길동" # 문자형
boolean <- TRUE # 진리값 : TRUE(T) / FALSE(F) [무조건 대문자]
boolean

boolean <- 3.14
boolean

# 자료형 확인
is.numeric(int) # TRUE #실수냐? 맞다면 TRUE - double()과 동일
is.integer(int) # FALSE : 정수 값도 부동소수점으로 관리. 정수값이나? 질문 20.0
is.double(int) # TRUE

as.integer(int) # 형변환을 시켜줌, 실수 -> 정수형
castInt <- as.integer(int)
is.integer(castInt) # TRUE

is.numeric(double) # TRUE 실수형 맞다
is.double(double) # TRUE 실수형 맞다 # 다른 언어 사용자를 위해 있음

is.character(double) # FALSE
is.character(string) # TRUE # 문자열이나? 맞다! TRUE
is.character("double") # TRUE # ""이기 때문에

# 문자 원소를 숫자 원소로 형변환
x <- c(1, 2, 3) # combine 결합하다
x

# 숫자 원소를 문자 원소로 형변환
y <- c(1, 2, "3") # 서로 다른형을 저장할 수 없다. 숫자 + 문자의 경우에만 강제 형변환 해줌.
y

result <- x * 3
result

result <- y * 5

result <- as.integer(y) * 5
result

# 복소수형 자료 생성과 형변환 # 실수와 허수 = 복소수
z <- 5.3 - 3i
Re(z) # 실수
Im(z) # 허수
is.complex(z) # TRUE # 복소수냐?
as.complex(5.3) # 5.3+0i # 강제로 복소수로 형변환을 해줌.

# 스칼라 변수의 자료형
mode(int) # "numeric"
mode(string) # "character"
mode(boolean) # "logical"

# 문자 벡터와 그래프 생성
gender <- c('man', 'woman', 'woman', 'man', 'man')
gender

mode(gender)
class(gender)

plot(gender)

# 요인형 변환
# as.factor() 함수 이용 범주(요인)형 변환
Ngender <- as.factor(gender) # 특정 변수에 담겨져있는 값들을 Domain, factor라고 한다.
Ngender
table(Ngender) # 출력의 결과를 테이블 형태로 보여주는 것. 빈도수를

# Factor형 변수로 차트 그리기
plot(Ngender)
mode(Ngender) # "numeric"
class(Ngender) # "factor"
is.factor(Ngender) # TRUE

```

```

# Factor Nominal 변수
Ngender

# factor() 함수 이용 Factor형 변환
args(factor) # parameter매개변수 = argument인자 # parameter만 보고 싶을 때 args()사용 가능하다. # ?factor
Ogender <- factor(gender, levels = c('woman', 'man'), labels = c("여자", "남자"), ordered = T)

Ogender

# 순서 없는 요인과 순서 있는 요인형 변수로 차트 그리기
par(mfrow=c(1,2))
plot(Ngender)
plot(Ogender)

# 도움말 보기
i <- sum(1, 2, 3)
i # 6

help(sum)
?sum

# 함수 파라미터 보기
args(sum)

# 함수 사용 예제 보기
example(sum)

# 작업 공간 지정
getwd() # 현재 작업 위치
setwd("D:/heaven_dev/workspaces/R/data")

```

2개 이상의 변수를 담는 것 Vector

factor는 범주(요인)형이다. 배열 범주.

특정 변수에 담겨져있는 값들을 Domain, factor라고 한다.

1개는 스칼라.

```

0
1 # 요인형 변환
2 # as.factor() 함수 이용 범주(요인)형 변환
3 Ngender <- as.factor(gender)
4 Ngender
5 table(Ngender)

```

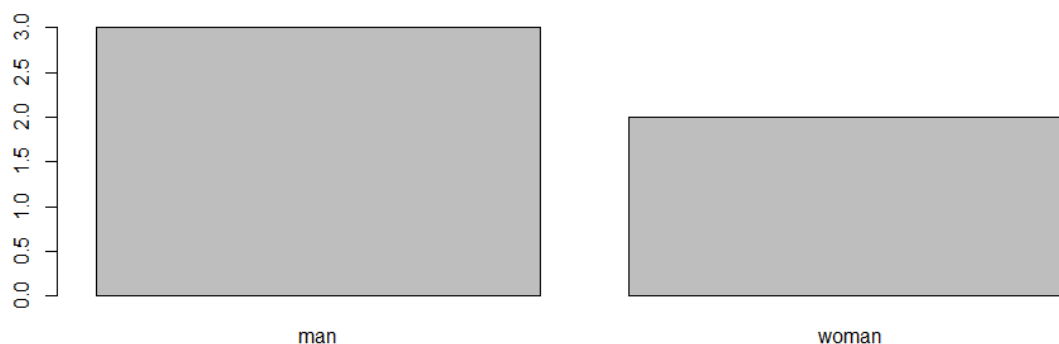
Ngender <- as.factor(gender)

factor는 변수가 갖고 있는 값의 빈도수를 저장한 것.

```

150
151 # 요인형 변환
152 # as.factor() 함수 이용 범주(요인)형 변환
153 Ngender <- as.factor(gender) # 특정 변수에 담겨져있는 값들을 Domain, factor라고
    한다.
154 Ngender
155 table(Ngender) # 출력의 결과를 테이블 형태로 보여주는 것. 빈도수를
156
157 # Factor형 변수로 차트 그리기
158 plot(Ngender)|
159 mode(Ngender)
160 class(Ngender)
161 is.factor(Ngender)
162

```



```

# factor() 함수 이용 Factor형 변환
args(factor) # parameter매개변수 = argument인자 # parameter만 보고 싶을 때 args
()사용 가능하다.
?factor

```

```

factor(x = character(), levels, labels = levels,
       exclude = NA, ordered = is.ordered(x), nmax = NA)

```

Arguments

x a vector of data, usually taking a small number of distinct values.

levels an optional vector of the unique values (as character strings) that **x** might have taken. The default is the unique set of values taken by `as.character(x)`, sorted into increasing order of **x**. Note that this set can be specified as smaller than `sort(unique(x))`.

labels either an optional character vector of labels for the levels (in the same order as **levels** after removing those in **exclude**), or a character string of length 1. Duplicated values in **labels** can be used to map different values of **x** to the same factor level.

exclude a vector of values to be excluded when forming the set of levels. This may be factor with the same level set as **x** or should be a character.

ordered logical flag to determine if the levels should be regarded as ordered (in the order given).

nmax an upper bound on the number of levels; see 'Details'.

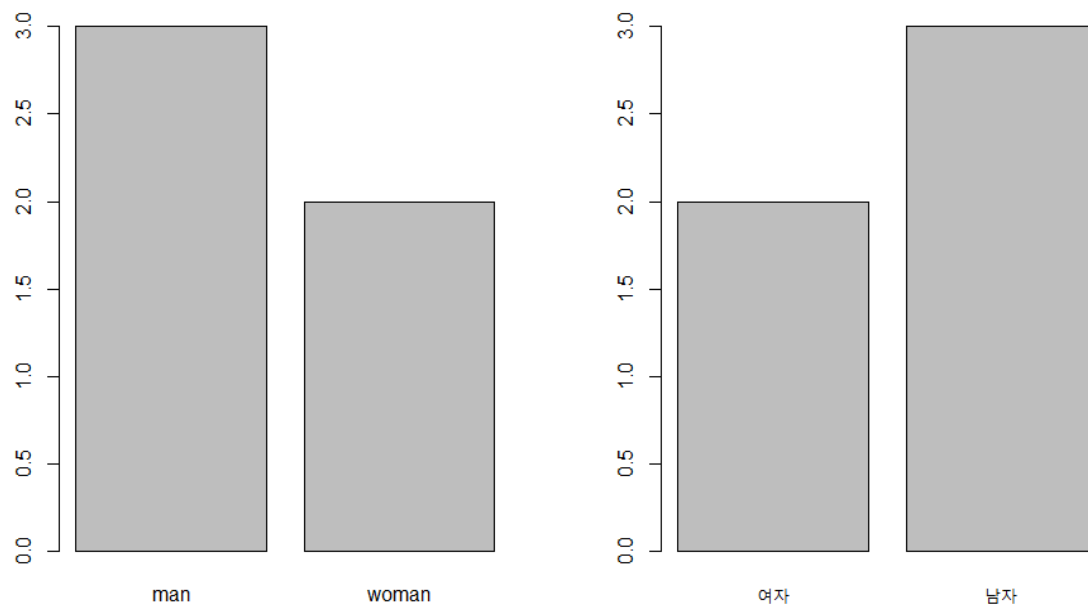
... (in `ordered(.)`): any of the above, apart from **ordered** itself.

ifany only add an NA level if it is used, i.e. if `any(is.na(x))`.

```
# factor() 함수 이용 Factor형 변환
args(factor) # parameter매개변수 = argument인자 # parameter만 보고 싶을 때 args
()사용 가능하다. # ?factor
Ogender <- factor(gender, levels = c('woman','man'), labels = c("여자","남자"),
ordered = T)

Ogender

# 순서 없는 요인과 순서 있는 요인형 변수로 차트 그리기
par(mfrow=c(1,2))
plot(Ngender)
plot(Ogender)
```



디폴트 설정을 해줄 수도 있다.

```
# 작업 공간 지정  
getwd() # 현재 작업 위치  
setwd("D:/heaven_dev/workspaces/R/data")
```

