1일차

# 데이터 분석가 \_ james \

# \

# 스크립트 실행(Run a script) \

## : Windows : 'Ctrl + Enter' \

## : MAC : 'Command + Enter'\

#---------------------------------

# 0 RStudio의 화면구성과 설정 ----

# 'Tools' → 'Global Options' → 'Appearance'

# Ctrl + Shift + N : 새스크립트 불러오기

##########################################

#### 1-1 변수 이해하기 ####

##########################################

# 데이터 형태 이해하기

# ▶ 변수를 생성하기 위해서는 대입연산자 (<-)를 사용하여 변수를 생성한다.

#### 변수 삽입 ####

x <- 10

y <- 10

z <- c(1,2,3,4,5)

#생선된 변수 목록을 보여준다

# ls함수에서는 목록만 보여주며 변수의 유형 값 등을 확인 할 수 없다.

ls()

#ls.str함수는 생성된 변수 목록 뿐만 아니라 각 변수의 유형 및 값 등을 확인 할 수 있다.

ls.str()

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

#### 변수 삭제 ####

# ▶ 필요 없는 변수나 함수를 작업 공간에서 삭제하거나 내용을 완전히 삭제할 때 rm 함수를 사용한다

x <- 10

y <- 20

z <- 30

ls() #생성된 변수 보기

rm(x) #x변수 삭제

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

#### 벡터(Vector) ####

#

# ▶ 벡터란 한 개 이상의 원소로 구성된 자료구조로서 R의 자료 객체 중에서 가장 기본이 되는 자료 객채임

# ▶ 그리고 하나의 벡터의 원소는 한 가지 형태(mode)만이 가능

# 벡터 생성

v1 <- c(1,2,3) #숫자형 벡터 생성

v2 <- c("a", "b", "c") #문자형 벡터 생성

v3 <- c(T, F, T)

height <- c(160, 140, 155) #height 벡터 생성

## 이건 왜 ????

my\_variable <- c(10)

my\_varıable

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

##########################################

#### 1-2 함수 사용 ####

##########################################

# 함수 사용

# 처음값:끝값 사용하기

# :은 사이의 수를 전부 갖고온다

c(1,2,3,4,5)

1:5

c(1:5)

1:10000

#seq (시작값, 마지막값, 증가분 \* length : 길이지정)

seq(1, 7, by=2) #seq(시작값, by=증가분, 조건지정)

seq(1, -1, by=-0.5)

seq(1,7,length=3)

#rep (반복할 값, 반복횟수 \* each : 각각 \* length : 길이만큼)

rep(c(1,2,3),3) #rep(a,b)는 a를 b만큼 반복

rep(1:3,3) #a:b는 a부터 b까지의 수

rep(c(4,2), times=2) # 반복 출력

rep(c(4,2), times=c(2,1)) # times 에 주어진 수 만큼 반복

rep(c(4,2), length=3) ## 길이만큼 반복

rep(c(4,2), each=3) ## 각각 반복

?rep

# rep, seq를 한번에 사용하는 문

##########################################

#### 1-3 범주형 자료 ####

##########################################

# 요인(범주형 자료)료

#factor(obj, levels=변수값, labels=요인이름)

#2PPT - 16

gender <- factor(c("MALE","FEMALE","MALE"))

gender

blood <- factor(c("O","AB","A"),levels = c("A","B","AB","O"),labels=c("A형","B형","AB형","O형"))

blood

# 하지만 아주 가끔씩 factor형 변형이 안될경우도 있다! 그런경우는 케릭터로 변경후 다시 변경

#순서

#ordered(obj, levels=변수값, labels=요인이름)

AgeG <- c(4,2,3,2,1,3,2,4)

age.ord <- ordered(AgeG,labels=c("20대 이하","30대","40대","50대 이상"))

# ▶ 벡터에서 자료를 삽입하고 삭제와 관련된 다양한 함수들을 살펴본다.

# 벡터 접근

vec1 <- c(1,2,3,4,5) #1~5까지 자료를 갖는 vec1 변수 생성

vec1[2] #두 번째 자료

vec1[c(2,3,5)] #vec1의 2, 3, 5의 값만 표현

vec1[c(-2,-3)] #vec1의 2, 3번째 자료 값 삭제

vec1[2] <- 6 #두 번째 위치의 2값이 6으로 대체됨

length(vec1)

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

##########################################

#### 1-4 데이터 구조 형태(메트릭스) ####

##########################################

#### 행렬(Matrix) ####

#

# ▶ 행렬(matrix)은 동일한 형으로 구성된 2차원의 데이터 구조이다

# ▶ 행의 차원과 열의 차원을 갖는 행렬은 수학에서의 행렬과는 달리 문자형이나 논리형 등을 원소로 사용할 수 있다.

# ▶ 그러나 행렬의 원소는 한가지 형의 자료만 허락됨

#행렬(matrix)은 여러 변수들이 이차원적으로 모여 있는 개체로,

#행렬을 생성하기 위해서는 matrix() 함수를 사용

#matrix() 함수 이외에 cbind(), rbind(), dim() 등을 이용하여 행렬을 생성시킬 수 있음

matrix(1:9, nrow=3) #nrow : 행의 개수 지정

matrix(c(1,4,7,2,5,8,3,6,9), byrow=T, ncol=3) #ncol : 열의 개수 지정 byrow=T : 행 기준 행렬을 생성

matrix(c(1,4,7,

2,5,8,

3,6,9), byrow=T, ncol=3) #ncol : 열의 개수 지정 byrow=T : 행 기준 행렬을 생성

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

# 벡터에 차원을 결정해줘서 생성 가능

m1 <- 1:9

dim(m1) <- c(3,3)

m1

#행렬과 관련된 여러 함수와 성분의 추출과 삭제 등에 관해 알아봄

mat <- matrix(c(1,2,3,4,5,6,7,8,9), ncol=3, byrow=T) #행 기준 3열의 행렬 생성

mat[1,] #행렬 mat의 1행의 값

mat[,3] #행렬 mat의 3열의 값

rownames(mat) <- c("a1","a2","a3")

mat

colnames(mat) <- c("z1","z2","z3")

mat

dim(mat) # 행 열 확인하기

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

##########################################

#### 1-5 데이터 구조 형태(배열) ####

##########################################

#### 배열(Array) ####

#

# ▶ 배열(Array)은 행렬을 2차원 이상으로 확장시킨 객체로써 2차원의 구조를 갖는 행렬도 배열의 한 종류이다.

# ▶ 그러나 일반적으로 3차원 이상의 차원을 갖는 데이터 객체를 배열이라고 부름

# ▶ 배열을 생성하기 위해서는 array 함수를 사용한다.

#배열의 속성 : 행렬의 속성과 같이 자료의 개수를 나타내는 length, 형태를 보여주는 mode,

#각 차원의 벡터의 크기를 나타내는 dim 그리고 각 차원의 리스트 이름을 나타내는 dimnames로 구성

#배열의 생성

#배열을 생성하기 위한 함수로 array() 함수와 dim() 함수가 있음

array(1:6) #1~6의 자료로 1차원 배열 생성

array(1:6, c(2,3)) #1~6의 자료로 2차원 배열 생성

array(1:8, c(2,2,2)) #1~8의 자료로 3차원 배열 생성

arr <- c(1:24) #1~24의 자료 생성

dim(arr) <- c(3,4,2) #dim() 함수를 이용하여 3행 4열의 행렬 2개 생성

arr

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

##########################################

#### 1-6 데이터 구조 형태(데이터 프레임) ####

##########################################

#### week 데이터 프레임(data.frame) ####

#

# ▶ 데이터 프레임은 행렬과 비슷한 형태로 되어 있으나,

# 행렬은 차원으로 표시되며 같은 형태(mode)의 객체를 가지는 반면,

# 데이터 프레임은 각 열(column)들이 서로 다른 형태(mode)의 객체를 가질 수 있으므로

# 범주형 변수를 가질 수도 있기 때문에 범주형 자료분석에도 유용하게 사용된다.

#data.frame() : 이미 생성되어 있는 벡터들을 결합하여 데이터 프레임을 생성

char1 <- rep(LETTERS[1:3],c(2,2,1)) #벡터

char1

num1 <- rep(1:3,c(2,2,1)) #벡터

num1

test1 <- data.frame(char1, num1) #test1 데이터 프레임 생성

test1

#as.data.frame() :모든 다른 종류의 자료객체들을 데이터 프레임으로 변환

mat <- matrix(c(1,2,3,4,5,6,7,8,9), ncol=3, byrow=T) #행 기준 3열의 행렬 생성

test3 <- as.data.frame(mat) #a1을 데이터 프레임으로 변환

test3

test3 <- data.frame(mat)

test3

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

##########################################

#### 1-7 데이터 구조 형태(리스트) ####

##########################################

#### 리스트(list) ####

#

# ▶ 리스트(List)는 서로 다른 형태(mode)의 자료를 포함하는 하나의 객체이다.

# ▶ 리스트를 생성하기 위해서는 list 함수를 사용한다.

li <- list("top", c(2,4,6),c(T,F,T)) #list(문자, 숫자, 논리형 객체)

li[[1]] #[[1]]:첫 번째 성분

mat1 <- matrix(1:4, nrow=2)

mat1

list1 <- list("A", 1:8, mat1)

list1

son <- list(son.name = c("Minsu", "Minchul"), son.cnt = 2, son.age = c(2.6))

son

#리스트 속성 : 벡터의 속성과 같이 자료의 개수, 형태, 구성요소의 이름 등을 보여주는 length, mode, names로 구성

length(son) #son 리스트 자료의 개수

mode(son) #son 리스트 자료의 형태

names(son) #son 리스트 각 구성요소의 이름

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

#예제

exm <- list(c("Abe", "Bob", "Carol", "Deb"),c("Weight","Waist")) #exm의 이름으로 list생성

exm[[2]] #리스트의 2번째 성분

exm[[2]][2] #2번째 성분 2번째 원소

names(exm) <- c("Rows","Columns") #exm 리스트에 성분 이름 부여

exm$Rows #exm의 Rows 성분만 표현

exm$Rows[2] #Rows 성분 2번째 원소 표현

exm$Columns #exm의 Columns 성분만 표현

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

# 편집기

x1<- c("a","b","c")

x2<- c(1,3,2)

df<-data.frame(x1, x2)

data.entry(x1,x2) #편집기를 사용 (행력, 메트릭스)

edit(df) #데이터프레임

1일차\_문제정답

#첫번째 연습문제

# 1번문제

rep(seq(2,100,2),each=2)

# 2번문제

rep(1:5,times=1:5)

#두번째 연습문제

#1

A=rep(seq(1,3,0.5),1:5)

B=1:15

#2

M=matrix(c(B,A),ncol=5,byrow = T)

#3

colnames(M)<-paste0("Col",1:5)

M

#4

M1<-M[c(1,4,6),c(1,3,5)]

#세번째 연습분제

#1

x.list<-list(c(41,25,35),matrix(10:17,ncol=2),c("Kim","Lee","Cho"))

x.list[[2]][3,2]

2일차

# 데이터 분석가 \_ james \

# \

# 스크립트 실행(Run a script) \

## : Windows : 'Ctrl + Enter' \

## : MAC : 'Command + Enter'\

#---------------------------------

#######################################

##### 복습하기 #####

#######################################

##### 1일차 복습

# matrix

mat <- matrix(c(1,2,3,4,5,6,7,8,9), ncol=3, byrow=T) #행 기준 3열의 행렬 생성

mat[1,] #행렬 mat의 1행의 값

mat[,3] #행렬 mat의 3열의 값

# data.frame

a<-1:10

b<-rep("a",10)

c<-data.frame(a,b)

# list

list1 <- list("A", 1:8) #list1 리스트 생성

list1[[3]] <- list(c(T, F)) #세 번째 성분을 추가

list1[[2]][9] <- 9 #두 번째 성분에 원소 추가

list1[3] <- NULL #세 번째 성분 삭제

list1[[2]] <- list1[[2]][-9] #두 번째 성분의 9번째 원소 삭제

rm(list=ls())

# 연습해보기!

# 1. a에 1부터 10까지 홀수를 3번씩 출력하고 다음과 같은 matrix 저장하시오

# [,1] [,2] [,3] [,4] [,5]

#[1,] 1 3 5 7 9

#[2,] 1 3 5 7 9

#[3,] 1 3 5 7 9

# 2. b와 c로 데이터 프레임을 d에 저장하시오

b <- c(1,2,3)

c <- c("a","b","c")

# 3. a와 d로 리스트를 e에 저장하고 2번째층에 저장된 데이터프레임 2행 2열을 출력하시오

#######################################

##### 2-1 데이터 호출하기 #####

#######################################

### 데이터 저장하기 및 호출 하기

# --- read.csv( )를 활용한 csv파일 불러오기 ----

# 작업 폴더 확인하기

getwd()

# 작업폴더 설정(Set Working Directory) : 'Ctrl+ Shift + h'

## RStudio Menu ; Session -> Set Working Directory -> Choose...

# 작업 폴더 지정하기

setwd()

## setwd('Your Working Directory Address')

# 데이터 불러오기

read.csv('test.csv')

## 작업 폴더에 있는 test.csv 불러와서 출력하기

# 인코딩 지정

read.csv('pop\_seoul\_euckr.csv', fileEncoding='euc-kr')

## Windows 인코딩 : CP949/euc-kr

## mac/Linux 인코딩 : UTF-8

## 같은 운영체제에서는 생략 가능

# '<-'을 활용해서 저장하기

pop\_seoul <- read.csv('pop\_seoul\_euckr.csv')

## 오른쪽 위 환경창에서 데이터 이름 클릭

View(pop\_seoul)

## 혹은 직접 View( )에 데이터를 넣고 실행

# 데이터 샘플 살펴보기

head(pop\_seoul)

tail(pop\_seoul, n=10)

## 첫 6개, 끝 10개 관측치만 콘솔창에서 보기

# 데이터 특성 확인하기

str(pop\_seoul)

## 데이터의 구조(Structure) 살펴보기

## 변수 형식 (뒤에 설명)

# 데이터 요약

summary(pop\_seoul)

##1.1 read.table( )로 txt파일 불러오기

## 탭으로 구분된 데이터

temp = read.table('pop\_seoul.txt',

header=TRUE,

fileEncoding='UTF-8')

temp

write.csv(pop\_seoul, file='aaa.csv', row.names = F)

## write.csv(저장할객체, file='경로/이름')

#openxlsx 패키지를 활용한 엑셀파일 불러오기

# openxlsx 패키지 설치

install.packages('openxlsx')

# library( )로 패키지 불러오기

library(openxlsx)

## 필요할 때마다 불러오기

## 원래는 없었던 read.xlsx( ) 함수 사용가능

# 데이터 불러오기

SHEET1 = read.xlsx('test.xlsx', sheet=1)

## xlsx 파일 경로와 시트 번호를 지정

SHEET1

SHEET2 = read.xlsx('test.xlsx', sheet=2, startRow=3)

## startRow= 옵션으로 데이터 시작 행번호 지정 가능

SHEET2

SHEET3 = read.xlsx('test.xlsx', sheet=3, colNames=FALSE)

## 첫 행이 변수이름이 아니라 관측치일때, colNames=FALSE 옵션 사용

SHEET3

#

## Rdata로 저장하기

# RDS 파일 (R객체 저장)

new\_data <- readRDS("iris.RDS")

# RDS 파일로 저장

saveRDS(new\_data, file = "new\_iris.RDS")

# image 파일

load("iris.RData")

iris4 <- iris

# image 저장

save.image("iris\_image.RData")

#load는 변수할당 x

######################### 1번 연습문제 #############################

##6 (실습) 다양한 데이터 불러오기

# 통계청 인구 데이터

## 출처 : http://kosis.kr/statisticsList/statisticsListIndex.do?menuId=M\_01\_01&vwcd=MT\_ZTITLE&parmTabId=M\_01\_01#SelectStatsBoxDiv

## 파일위치 :'data/'

## 파일이름 : '광역시도별\_연령성별\_인구수.xlsx'

## 데이터 시작 위치 : 2행

# 통계청 가구별 주택 거주 데이터

## 출처 : http://kosis.kr/statHtml/statHtml.do?orgId=101&tblId=DT\_1JU1501&vw\_cd=&list\_id=&seqNo=&lang\_mode=ko&language=kor&obj\_var\_id=&itm\_id=&conn\_path=I2

## 파일위치 :'data/'

## 파일이름 : '시도별\_가구\_주택종류.xlsx'

## 데이터 시작 위치 : 2행

# 국토교통부 아파트 실거래가 데이터(2018, 강남구)

## 출처 : http://rtdown.molit.go.kr/

## 파일위치 :'data/'

## 파일이름 : '아파트매매\_2019\_강남구.xlsx'

# 데이터 시작 위치 : 17행

#######################################

##### 2-2 R 연산하기 #####

#######################################

#### week 사칙연산 ####

#

# ▶ R에서 제공하는 연산에 대하여 알아본다.

a<-c(1,2)

b<-c(3,4)

a+b #벡터 변수의 덧셈

a-b #벡터 변수의 뺄셈

a\*b #벡터 변수의 곱셈

a/b #벡터 변수의 나눗셈

a<-c(4,5)

b<-c(2,3)

a^b #벡터변수의 제곱(구성 요소들간의 제곱, 4^2, 5^3)

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

#%/% : 나눗셈에서 몫만 출력함

#%% : 나눗셈에서 몫만 출력함

a<-c(7,2)

b<-c(3,4)

a%/%b #벡터 변수의 정수나눗셈

a%%b #벡터 변수의 나머지

#행렬의 곱

A<-matrix(c(5,10,2,1), ncol=2)

B<-matrix(c(3,4,5,6), ncol=2)

#(5\*3) + (2\*4) ; (5\*5) + (2\*6) ; (10\*3) + (1\*4) ; (10\*5) + (1\*6)

A;B

A%\*%B

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

#### 비교연산 ####

#

# ▶ R에서 제공하는 연산에 대하여 알아본다.

# '==' 비교되는 두 항이 같은지를 비교함. 같을 경우 True, 다를 경우 False

x<-2

y<-3

x==y

# '!=' 비교되는 두 항이 다른지를 비교함. 같을 경우 False, 다를 경우 True

x<-2

y<-3

x!=y

# '<=' 왼쪽 항이 오른쪽 항보다 작거나 같음을 비교함. 작거나 같으면 True, 크면 False

x<-2

y<-2

x<=y

# '<' 왼쪽 항이 오른쪽 항보다 작음을 비교함. 작으면 True, 크면 False

1<2

# '>' 왼쪽 항이 오른쪽 항보다 큼을 비교함. 크면 True, 작으면 False

1>2

# '>=' 왼쪽 항이 오른쪽 항보다 크거나 같음을 비교함. 크거나 같으면 True, 작으면 False

1>=2

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

#### 논리연산 ####

#

# ▶ R에서 제공하는 연산에 대하여 알아본다.

# ! : not 연산자

# & : 벡터에서의 and 논리 연산자

2==2 & c( 3>4)

2==2 & c( 3<4)

# | : 벡터에서의 or 논리 연산자

2==2 | c( 3>4)

2!=2 | c( 3>4)

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

#### week. 4.3 열 추가 ####

# ▶ 이번 절에서는 데이터의 열 추가에 대하여 알아본다.

tmp\_df <- data.frame(AA = c(1:5), BB = c("A","A","B","B","B"))

head(tmp\_df) # 데이터 확인

#CC컬럼을 새로 생성하고 그 안에 값을 1로 채워 넣음

tmp\_df$CC <- 1

#컬럼 AA와 컬럼 CC의 값의 합한 값을 새로운 DD컬럼으로 생성

tmp\_df$DD <- tmp\_df$AA + tmp\_df$CC

head(tmp\_df) # 데이터 확인

#### 열 제거 ####

# ▶ 열을 제거 하기 위해서는 열의 위치 번호에 (-)를 입력하여 제거 한다.

tmp\_df[, -1] #첫번째 위치의 컬럼 제거

tmp\_df[, -"AA"] #오류 발생함

# 다음에 오류가 있습니다-"AA" : 단항연산자에 유효한 인자가 아닙니다

tmp\_df[, c("BB","CC","DD")]

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

#######################################

##### 2-3 데이터형태 조정하기 #####

#######################################

#### 데이터 추출(Select) ####

# 데이터 추출(Select)

# ▶ 이번 절에서는 데이터 추출하기에 대하여 알아본다.

Sample.df <- data.frame(AA = rep(letters[1:5],10), BB = sample(60:70, 50, replace = T))

# sample (범위, 추출수, replace = 중복가능)

sample(1:6,10,replace = T) # 1부터 6을 10번 랜덤추출

#만약 랜덤을 고정하고 싶다면 set.seed(숫자)

set.seed(1234)

sample(1:6,10,replace=T)

set.seed(1234)

sample(1:6,10,replace=T)

head(Sample.df)

#AA컬럼의 값중에서 a인 값만 추출

#Type1

Sample.df[Sample.df$AA == "a",]

#Type2

# subset(데이터, 조건)

subset(Sample.df, AA == "a")

#AA컬럼의 값중에서 a 와 b의 값만 추출

#Type1

Sample.df[Sample.df$AA %in% c("a","b"),]

#Type2

subset(Sample.df, AA %in% c("a","b"))

# 필요한 컬럼 Select

Sample.df1 <- Sample.df

#Type1

Sample.df1[,c("AA","BB")]

#Type2

Sample.df1[,c(1,2)]

#Type3

Sample.df1[,c(-4,-5)]

rm(list = ls()) #오브젝트 내의 모든 변수 삭제

######################## 2번 연습문제 ##########################

data(iris)

head(iris)

# 1. Sepal.Length 변수의 짝수행을 출력하시오 .

# 2. Subset을 사용해서 변수 Species 에서 setosa 인 데이터를 추출하시오

# 3. 2번에서 뽑은 데이터를 a1에 저장하고 Sepal.Length 가 5 보다 작은 Petal.Width 의 합을 구하시오

#############################################################

#######################################

##### 2-4 데이터 합치기 #####

#######################################

#### 데이터 합치기 ####

# ▶ 데이터를 합치기 위해서 사용되는 함수는 cbind, rbind, merge 가 가장 많이 사용되어진다.

# 예제 데이터 불러오기

## 국토교통부 아파트 실거래가 데이터

## 출처 : http://rtdown.molit.go.kr/

library(openxlsx)

GN = read.xlsx('아파트매매\_2018\_강남구.xlsx', sheet=1, startRow=17)

head(GN)

tail(GN)

GD = read.xlsx('아파트매매\_2018\_강동구.xlsx', sheet=1, startRow=17)

head(GD)

SC = read.xlsx('아파트매매\_2018\_서초구.xlsx', sheet=1, startRow=17)

SP = read.xlsx('아파트매매\_2018\_송파구.xlsx', sheet=1, startRow=17)

# 데이터 구성 확인

names(GN)

names(GD)

str(GN)

str(GD)

# rbind( )를 활용한 행/관측치 결합

GN4 = rbind(GN, GD, SC, SP)

head(GN4)

tail(GN4)

##2 열 결합

# 가상의 예제 데이터 확인

my\_data = data.frame(id = 1:5,

gender = c('M','F','F','F','M'),

age = seq(15, 35, 5))

my\_data

# 추가 변수를 포함한 데이터

another\_data = data.frame(region = c('Seoul','Seoul','Seoul','Busan','Busan'),

amount = c(1,1,1,1,1))

another\_data

# cbind( )로 열/변수 결합

cbind(my\_data, another\_data)

## 일반적으로 잘 활용하지 않음

## $를 활용한 변수 추가 혹은 key(id) 변수를 활용한 결합 활용

my\_data$amount = 100

my\_data

## 동일한 값 변수 추가

my\_data$age\_grp = cut(my\_data$age,

breaks=c(10,20,30,40),

include.lowest=TRUE,

right=FALSE,

labels=c('10\_19','20\_29','30\_39'))

my\_data

## cut( )을 활용한 연령대 변수 추가

## breaks : 구간 경계값

## include.lowest : 첫 경계값 포함 여부

## right : 각 구간의 오른쪽 경계 포함 여부

## labels : 각 구간의 이름

############ 조건에 맞는 데이터 합치기 (merge)

sales = read.csv('ex\_sales.csv')

sales

prod = read.csv('ex\_prod.csv')

prod

# merge( )를 활용한 데이터 결합

merged = merge(sales, prod, by.x='PROD', by.y='PROD')

## merge(데이터1, 데이터2, by.x='첫번째데이터의 기준변수', by.y='두번째...')

merged = merge(sales, prod, by='PROD')

## 기준변수가 같을 때는 "by="으로 한번에 지정 가능

merged

# all 옵션의 활용

merge(sales, prod, by.x='PROD', by.y='PROD', all.x=TRUE)

## all.x=TRUE : 짝이 없는 첫번째 데이터의 관측치도 포함

## Excel의 VLOOKUP 느낌

merge(sales, prod, by.x='PROD', by.y='PROD', all.y=TRUE)

## all.y=TRUE : 짝이 없는 두번째 데이터의 관측치도 포함

merge(sales, prod, by.x='PROD', by.y='PROD', all=TRUE)

## all=TRUE : 짝이 없는 모든 관측치 포함

###########

# 1:1, 다:1은 문제가 없지만 다:다 결합은 조심!

prod2 = read.csv('ex\_prod2.csv')

prod2

## 상품 B에 대한 정보가 중복

merge(sales, prod2, by.x='PROD', by.y='PROD')

## by로 지정된 변수값 기준

## 모든 가능한 결합을 생성

################################ merge 연습문제 ###################################

a1 <- data.frame(name=c("aa","bb","cc"),value=seq(10,20,length.out = 3))

a2 <- data.frame(name=c("cc","dd","ee"),value=seq(30,50,length.out = 3))

a3 <- data.frame(name=c("aa","dd","ee","ff"),value=seq(20,80,length.out = 4),any=seq(0,3,1))

# 1. a1 와 a2 를 행결합 하시오

# 2. a3 와 a1 를 행결합하시오

# 3. a3 와 a2 를 name 기준으로 결합 하시오 (a3 데이터는 모두출력)

# 4. 3번 데이터를 z 에 저장하고 value.y 기준으로 정렬하시오

####################################################################################

2일차\_문제정답

### 복습 연습문제

1번

a <- matrix(rep(seq(1,10,2),3),ncol=5,byrow=T)

2번

d <- data.frame(b,c)

3번

e <- list(a,d)

e[[2]][2,2]

######################### 1번 연습문제 #############################

데이터 불러오기 연습문제

# data= read.xlsx('광역시도별\_연령성별\_인구수.xlsx', startRow=2)

# data= read.xlsx('시도별\_가구\_주택종류.xlsx', startRow=2)

# data= read.xlsx('아파트매매\_2019\_강남구.xlsx', startRow=17)

######################## 2번 연습문제 ##########################

data(iris)

head(iris)

# 1. Sepal.Length 변수의 짝수행을 출력하시오 .

iris$Sepal.Length[seq(2,150,2)]

iris$Sepal.Length[1:150 %%2 == 0]

# 2. Subset을 사용해서 변수 Species 에서 setosa 인 데이터를 추출하시오

subset(iris,iris$Species=="setosa")

iris[iris$Species=="setosa",]

# 3. 2번에서 뽑은 데이터를 a1에 저장하고 Sepal.Length 가 5 보다 작은 Petal.Width 의 합을 구하시오

a1<-iris[iris$Species=="setosa",]

sum(a1[a1$Sepal.Length<5,"Petal.Width"])

################################ merge 연습문제 ###################################

a1 <- data.frame(name=c("aa","bb","cc"),value=seq(10,20,length.out = 3))

a2 <- data.frame(name=c("cc","dd","ee"),value=seq(30,50,length.out = 3))

a3 <- data.frame(name=c("aa","dd","ee","ff"),value=seq(20,80,length.out = 4),any=seq(0,3,1))

# 1. a1 와 a2 를 행결합 하시오

rbind(a1,a2)

# 2. a3 와 a1 를 행결합하시오

install.packages("plyr")

library(plyr)

rbind.fill(a3,a1)

# 3. a3 와 a2 를 name 기준으로 결합 하시오 (a3 데이터는 모두출력)

merge(a3,a2,by="name",all.x=T)

# 4. 3번 데이터를 z 에 저장하고 value.y 기준으로 정렬하시오

z<-merge(a3,a2,by="name",all.x=T)

z[order(z$value.y),]

3일차

# 데이터 분석가 \_ james \

# \

# 스크립트 실행(Run a script) \

## : Windows : 'Ctrl + Enter' \

## : MAC : 'Command + Enter'\

#---------------------------------

# R 프로그래밍

### 조건문

if (cond) {

# cond가 참일 때 실행할 문장

} else {

# cond가 거짓일 때 실행할 문장

}

if ( TRUE ) {

print("TRUE")

print("hello")

} else {

print("FALSE")

print("world")

}

if( 조건1 ) {

# 표현식 1

} else if (조건2) {

# 표현식 2

} else {

# 표현식 3

}

team\_A <- 2 # Number of goals scored by Team A

team\_B <- 2# Number of goals scored by Team B

if (team\_A > team\_B){

print ("Team A won")

} else if (team\_A < team\_B){

print ("Team B won")

} else {

"Team A & B tied"

}

## 하나의 논리값에 대한 판단

x1 <- c(4)

if (x1 %% 2 == 0) {

y1 = c("Even Number")

print(y1)

} else {

y1 = c("Odd Number")

print(y1)

}

## 두개 이상의 논리값에 대한 판단

# 엑셀if와 똑같음

ifelse(조건,참,거짓)

x <- c(1, 2, 3, 4, 5)

ifelse(x %% 2 == 0, "even", "odd")

x <- c(1,2,3,4)

y <- c(2,1,4,5)

ifelse(x<y, x, y)

ifelse(sum(x-y) > 0, "positive", ifelse(sum(x-y) < 0 , "negative", "zero"))

############ 연습해보자 fi, ifelse

# 문제 1

# a의 평균이 15이상이면 "평균이상" 아니면 "평균미만"으로 출력하시오

a <- seq(1,30,4)

# 문제 2

# if , else if , else를 사용해서 tmep 조건을 만드시오

# 0이하면 freezing, 10이하면 cold, 20이하면 cool, 30이하면 warm, 그외는 hot이 출력되게 하시오

temp <- 25

# 문제 2\_1

temp <-c(5,20,-6,37,24,13)

# 문제 2번의 값을 ifelse 로 바꿔서 값을 변경하시오

# 문제 3

# - ifelse 를 사용해서 iris의 Sepal.Length가 6보타 크면 1 작으면 0 변수 생성하시오

# - new라는 변쉐 추가하고 new가 1인 Sepal.Width의 합을 구하시오

data(iris)

#####################################

#####################################

# 루프

# 루프는 작업을 반복하는 R의 방법으로

# 시뮬레이션 프로그래밍에 유용한 도구입니다.

#expand.grid

# n개의 벡터에 있는 요소이 모든 조합을 작성

# ex) 두 주사위의 모든 조합

die <- 1:6

rolls <- expand.grid(die, die)

# 두개 이상의 백터와 함께 사용

# 모든 주사위의 합은?

rolls$value <- rolls$Var1 + rolls$Var2

head(rolls)

rolls$value

# 확률 N 개의 독립적 인 랜덤 사건의 모든 발생은

# 각각 임의의 이벤트가 발생하는 확률의 곱과 같다 .

prob <- c("1" = 1/8, "2" = 1/8, "3" = 1/8, "4" = 1/8, "5" = 1/8, "6" = 3/8)

prob

rolls$prob1 <- prob[rolls$Var1]

rolls$prob2 <- prob[rolls$Var2]

rolls$prob <- rolls$prob1 \* rolls$prob2

head(rolls)

# 주사위의 예상 기대값은?

sum(rolls$value \* rolls$prob)

# 따라서 두 개의로드 된 주사위를 굴릴 때 예상되는 값은 8.25입니다.

############ 연습해보자 expand.grid

wheel <- c("DD", "7", "BBB", "BB", "B", "C", "0")

prob <- c("DD" = 0.03, "7" = 0.03, "BBB" = 0.06,

"BB" = 0.1, "B" = 0.25, "C" = 0.01, "0" = 0.52)

#1번 문제

# wheel 과 같이 총 7개의 경우의 수가 있다.

# 각 확률은 prob와 같고 총 3번의 시도를 했을 경우에 0.001보다 높은 경우의 수 개수는?

# (3번 추출하며 각각 독립이다)

# (DD, BBB, 7) 과 (DD , 7 , BBB)는 다른 경우의 수다

#2번 문제

# 동전을 3번 던질 떄 확률은 0.3과 0.7이다

# 첫번째에 앞이나오고 그리고(&) 세번째에 뒤가 나올 확률을 구하시오

coin <- c("앞","뒤")

prob <- c("앞" = 0.3, "뒤" = 0.7)

#####################################

#####################################

# for 루프

# for루프는 R에게 "모든 값에 대해이 작업을 수행하십시오"

# 구조

for (value in that) {

this

}

# 사례1

for (value in c("My", "first", "for", "loop")) {

print("one run")

}

# for 루프는 print("one run")문자열 벡터의 각 요소에 대해 한 번씩 실행

# 사례2

for (value in c("My", "second", "for", "loop")) {

print(value)

}

value

# 두개의 차이?

for (word in c("My", "second", "for", "loop")) {

print(word)

}

for (string in c("My", "second", "for", "loop")) {

print(string)

}

for (i in c("My", "second", "for", "loop")) {

print(i)

}

# 출력없이 적재

for (value in c("My", "third", "for", "loop")) {

value

}

value

# 루프의 출력을 저장하려면 실행시 자체 출력을 저장하도록 루프

chars <- rep(0,4)

words <- c("My", "fourth", "for", "loop")

for (i in 1:4) {

chars[i] <- words[i]

}

chars

#for문을 통한 계산

rolls$new <- NA

for (i in 1:nrow(rolls)) {

symbols <- sum(rolls[i, 1], rolls[i, 2], rolls[i, 3])

rolls$new[i] <- symbols

}

############ 연습해보자 for문

#1번 문제

sum <- 0

# for문을 사용해서 1부터 100까지의 누적합을 구하시오

#2번 문제

sum2 <- 0

sample(1:6,1)

# for문을 사용해서 위의 주사위 20번 던진 누적 합을 구하시오

#####################################

#####################################

# while 루프

# 구조

while (condition) {

code

}

#예시

i <- 1

while (i < 6) {

print(i)

i = i+1

}

# 1~10까지 누적합 구하기 (cummlative sum by while) : while(condition) { expression }

z <- 0

i <- 1

while( i <= 10) {

z = z + i

cat("cummulative summation",z, "\n")

i = i + 1

}

k <- 1

repeat {

k <- k+3

if (k > 5) break}

############ 연습해보자 for, while 문

#1번 문제 구구단 만들어보기

# 2단부터 9단까지 출력해보기

#ex)

# 2 4 6 8 10 12 14 16 18

# 3 6 9 12 ....

# .....

# 9 18 27 36 ....

#1\_1 for문

#seq(값 , by = 차이, length.out = 길이)

#1\_2 while문

#while문

i<-2

#####################################

#####################################

###########################

###### 함수 만들기 #######

###########################

# 1. 함수 생성 및 실행하기

myfunction <- function(){

print("Hi Hello")

}

myfunction()

# 2. 인수 값 전달 함수

make\_sum <- function(x,y){

x+y

}

make\_sum(3,4)

# 3. 기본 값 지정하기

pp <- function(x,y=6){

x^y

}

pp(2)

pp(4,2)

# 4. 함수에서 특정 값 반환 return

make\_sum <- function(x,y){

return(x+y)

}

make\_sum(3,4)

dt <- function(x,y){

add <- x+y

mul <- x\*y

c(add = add, mul = mul)

}

dt(3,5)

# 5. 인수의 개수가 가변적인 상황

my\_function <- function(x,...){

print(x)

summary(...)

}

z<-1:20

zz <- my\_function("hi",z)

############ 연습해보자 function

#1

# 성적을 입력했을경우 40점 이하는 "C", 70점 이하는 "B"

# 71점 이상은 "A"를 출력하는 function을 만드시오

###########################

###### 다양한 apply #######

###########################

# apply( )

# 배열 또는 행렬에 주어진 함수를 적용한 뒤 그 결과를 벡터, 배열 또는 리스트로 반환

# 배열 또는 행렬에 적용

apply(

X, # 배열 또는 행렬

MARGIN, # 함수를 적용하는 방향. 1은 행 방향, 2는 열 방향

# c(1, 2)는 행과 열 방향 모두를 의미

FUN # 적용할 함수

)

# sum이라는 함수 적용에 대해서 진행

sum(1:10)

d <- matrix(1:9, ncol=3)

d

apply(d, 1, sum)

apply(d, 2, sum)

head(iris)

apply(iris[, 1:4], 2, sum)

# 이와같이 행, 열의 합 평균은 빈번하게 사용되므로 알면 좋은 함수들

rowSums(

x, # 배열 또는 숫자를 저장한 데이터 프레임

na.rm=FALSE, # NA를 제외할지 여부

)

#반환 값은 행 방향에 저장된 값의 합이다.

rowSums(

x, # 배열 또는 숫자를 저장한 데이터 프레임

na.rm=FALSE, # NA를 제외할지 여부

)

#반환 값은 행 방향에 저장된 값의 평균이다.

rowSums(iris[, 1:4])

colSums(iris[, 1:4])

#####

# lapply( )

# 벡터, 리스트 또는 표현식에 함수를 적용하여 그 결과를 리스트로 반환

# lapply : 벡터, 리스트, 표현식, 데이터 프레임 등에 함수를 적용하고 그 결과를 리스트로 반환한다.

lapply(

X, # 벡터, 리스트, 표현식 또는 데이터 프레임

FUN, # 적용할 함수

... # 추가 인자. 이 인자들은 FUN에 전달된다.

)

# 반환 값은 X와 같은 길이의 리스트다.

# 반환 값은 함수 호출 결과다.

(result <- lapply(1:3, function(x) { x\*2 }))

unlist(result)

# ㅣlist를 인자로 받을 수 있다.

(x <- list(a=1:3, b=4:6))

lapply(x, mean)

# data.frame를 인자로 받을 수 있다.

lapply(iris[, 1:4], mean)

########## 로 lapply( )의 결과를 벡터 또는 데이터 프레임으로 변환할 필요가 있다.

# unlist : 리스트 구조를 벡터로 변환한다.

unlist(

x, # R 객체. 보통 리스트 또는 벡터

recursive=FALSE, # x에 포함된 리스트 역시 재귀적으로 변환할지 여부

use.names=TRUE # 리스트 내 값의 이름을 보존할지 여부

)

# 반환 값은 벡터다.

# do.call : 함수를 리스트로 주어진 인자에 적용하여 결과를 반환한다.

do.call(

what, # 호출할 함수

args, # 함수에 전달할 인자의 리스트

)

# unlist로 백터 변환 후 데이터프레임 만들기

d <- as.data.frame(matrix(unlist(lapply(iris[, 1:4], mean)),ncol=4, byrow=TRUE))

names(d) <- names(iris[, 1:4])

# do.call을 활용한 데이터 프레임 만들기

data.frame(do.call(cbind, lapply(iris[, 1:4], mean)))

#### unlist의 문제점!

x <- list(data.frame(name="foo", value=1),data.frame(name="bar", value=2))

unlist(x)

do.call(rbind,x)

#############

sapply( )

# lapply와 유사하지만 결과를 벡터, 행렬 또는 배열로 반환

sapply(

X, # 벡터, 리스트, 표현식 또는 데이터 프레임

FUN, # 적용할 함수

..., # 추가 인자. 이 인자들은 FUN에 전달된다.

)

# 반환 값은 FUN의 결과가 길이 1인 벡터들이면 벡터, 길이가 1보다 큰 벡터들이면 행렬이다.

lapply(iris[, 1:4], mean)

sapply(iris[, 1:4], mean)

class(sapply(iris[, 1:4], mean))

# 형태는 숫인 백터!

sapply(iris[, 1:4], mean)

# sapply( )에서 반환한 벡터는 as.data.frame( )을 사용해 데이터 프레임으로 변환

x <- sapply(iris[, 1:4], mean)

as.data.frame(x)

as.data.frame(t(x))

# 변수 구조 확인할떄 많이 쓴다.

sapply(iris, class)

str(iris)

# sapply( )에 인자로 주어진 함수의 출력이 길이가 1보다 큰 벡터들이라면 sapply( )는 행렬을 반환

y <- sapply(iris[, 1:4], function(x) { x > 3 })

###########################

tapply( )

# 벡터에 있는 데이터를 특정 기준에 따라 그룹으로 묶은 뒤 각 그룹마다 주어진 함수를 적용하고 그 결과를 반환

tapply(

X, # 벡터

INDEX, # 데이터를 그룹으로 묶을 색인. 팩터를 지정해야 하며 팩터가 아닌 타입이 지정되면

# 팩터로 형 변환된다.

FUN, # 각 그룹마다 적용할 함수

..., # 추가 인자. 이 인자들은 FUN에 전달된다.

)

# 반환 값은 배열이다.

# 부터 10까지의 숫자가 있고 이들이 모두 한 그룹에 속해 있을 때, 이 그룹에 속한 데이터의 합

tapply(1:10, rep(1, 10), sum)

# 홀짝에 따라서

tapply(1:10, 1:10 %% 2 == 1, sum)

tapply(iris$Sepal.Length, iris$Species, mean)

######

mapply( )

# sapply의 확장된 버전으로, 여러 개의 벡터 또는 리스트를 인자로 받아 함수에 각 데이터의 첫째 요소들을 적용한 결과,

# 둘째 요소들을 적용한 결과, 셋째 요소들을 적용한 결과 등을 반환

# sapply( )와 유사하지만 다수의 인자를 함수에 넘긴다는 점에서 차이

mapply(

FUN, # 실행할 함수

..., # 적용할 인자

)

rnorm(10, 0, 1)

mapply(rnorm,

c(1, 2, 3),

c(0, 10, 100),

c(1, 1, 1))

# 1은 rnorm(n=1, mean=0, sd=1),

# 2는 rnorm(n=2, mean=10, sd=1),

# 3은 rnorm(n=3, mean=100, sd=1)에 해당한다.

mapply(mean, iris[, 1:4])

# mapply( )에는 iris의 모든 행이 나열되어 인자

############ 연습해보자 apply 종류

#문제 1

data(iris)

# iris에서 각 행바다(1~4열)의 분산 열을 추가하시오 (var)

#문제 2

# function을 활용해서 iris[,1:4]의 모든 값들은 제곱하시오

#문제 3

# iris에서 for문을 활용해서 숫자형(iris[,1:4]) 값을 Species별 평균을 구하시오

# z변수에 저장

3일차\_문제정답

############ 연습해보자 fi, ifelse

# 문제 1

# a의 평균이 15이상이면 "평균이상" 아니면 "평균미만"으로 출력하시오

a <- seq(1,30,4)

if (mean(a) >= 15){

print("평균이상")

} else { print("평균미만")}

# 문제 2

# if , else if , else를 사용해서 tmep 조건을 만드시오

# 0이하면 freezing, 10이하면 cold, 20이하면 cool, 30이하면 warm, 그외는 hot이 출력되게 하시오

temp <- 25

if (temp <= 0) {

"freezing"

} else if (temp <= 10) {

"cold"

} else if (temp <= 20) {

"cool"

} else if (temp <= 30) {

"warm"

} else {

"hot"

}

# 문제 2\_1

temp <-c(5,20,-6,37,24,13)

# 문제 2번의 값을 ifelse 로 바꿔서 값을 변경하시오

ifelse(temp <= 0 ,"freezing",

ifelse(temp <= 10 ,"cold",

ifelse(temp <= 20 ,"cool",

ifelse(temp <= 30 ,"warm","hot"))))

# 문제 3

# - ifelse 를 사용해서 iris의 Sepal.Length가 6보타 크면 1 작으면 0 변수 생성하시오

# - new라는 변쉐 추가하고 new가 1인 Sepal.Width의 합을 구하시오

data(iris)

iris$new <- ifelse(iris$Sepal.Length>6,1,0)

sum(iris[iris$new==1,"Sepal.Width"])

#####################################

#####################################

############ 연습해보자 expand.grid

wheel <- c("DD", "7", "BBB", "BB", "B", "C", "0")

prob <- c("DD" = 0.03, "7" = 0.03, "BBB" = 0.06,

"BB" = 0.1, "B" = 0.25, "C" = 0.01, "0" = 0.52)

#1번 문제

# wheel 과 같이 총 7개의 경우의 수가 있다.

# 각 확률은 prob와 같고 총 3번의 시도를 했을 경우에 0.001보다 높은 경우의 수 개수는?

# (3번 추출하며 각각 독립이다)

# (DD, BBB, 7) 과 (DD , 7 , BBB)는 다른 경우의 수다

roll <- expand.grid(wheel,wheel,wheel)

roll$prob1 <- prob[roll$Var1]

roll$prob2 <- prob[roll$Var2]

roll$prob3 <- prob[roll$Var3]

roll$prob <- roll$prob1 \* roll$prob2 \* roll$prob3

nrow(roll[roll$prob>=0.001,])

sum(roll$prob >= 0.001)

#2번 문제

# 동전을 3번 던질 떄 확률은 0.3과 0.7이다

# 첫번째에 앞이나오고 그리고(&) 세번째에 뒤가 나올 확률을 구하시오

coin <- c("앞","뒤")

prob <- c("앞" = 0.3, "뒤" = 0.7)

roll2 <- expand.grid(coin,coin,coin)

roll2$prob1 <- prob[roll2$Var1]

roll2$prob2 <- prob[roll2$Var2]

roll2$prob3 <- prob[roll2$Var3]

roll2$prob <- roll2$prob1 \* roll2$prob2 \* roll2$prob3

sum(roll2[roll2$Var1=="앞" & roll2$Var3=="뒤","prob"])

############ 연습해보자 for문

#1번 문제

sum <- 0

# for문을 사용해서 1부터 100까지의 누적합을 구하시오

for(i in 1:100){

sum <- sum + i

}

sum

#2번 문제

sum2 <- 0

sample(1:6,1)

# for문을 사용해서 위의 주사위 20번 던진 누적 합을 구하시오

for(i in 1:20){

sum2 <- sum2 + sample(1:6,1)

}

sum2

############ 연습해보자 for, while 문

#1번 문제 구구단 만들어보기

# 2단부터 9단까지 출력해보기

#ex)

# 2 4 6 8 10 12 14 16 18

# 3 6 9 12 ....

# .....

# 9 18 27 36 ....

#1\_1 for문

#seq(값 , by = 차이, length.out = 길이)

############ 연습해보자 function

#1

# 성적을 입력했을경우 40점 이하는 "C", 70점 이하는 "B"

# 71점 이상은 "A"를 출력하는 function을 만드시오

grade <- function(x) {

if(x>=0 & x<=40){

print("C")

} else if(x<=70){

print("B")

} else {print("A")}

}

grade(20) # C

grade(60) # B

grade(90) # A

for( i in 2:9){

print(seq(i,by=i,length.out = 9))

}

#1\_2 while문

#while문

i<-2

while(i<10){

print(seq(i,by=i,length.out = 9))

i <- i+1

}

############ 연습해보자 apply 종류

#문제 1

data(iris)

# iris에서 각 행바다(1~4열)의 분산 열을 추가하시오 (var)

iris$var <- apply(iris[1:4],1,var)

#문제 2

# function을 활용해서 iris[,1:4]의 모든 값들은 제곱하시오

sapply(iris[, 1:4], function(x) { x ^ 2 })

#문제 3

# iris에서 for문을 활용해서 숫자형(iris[,1:4]) 값을 Species별 평균을 구하시오

# z변수에 저장

z<-c()

for(i in 1:4){

a <- tapply(iris[,i],iris$Species,mean)

z <- rbind(z,data.frame(t(a)))

}

4일차

# 데이터 분석가 \_ james \

# \

# 스크립트 실행(Run a script) \

## : Windows : 'Ctrl + Enter' \

## : MAC : 'Command + Enter'\

#---------------------------------

# 복습 !

######################

####### dplyr ########

######################

# dplyr 패키지 설치

install.packages('dplyr')

library(dplyr)

# SKT의 delivery 데이터 활용

## 원본 출처 : SKT Big Data Hub(www.bigdatahub.co.kr)

delivery = read.csv('SKT.csv', fileEncoding='UTF-8')

head(delivery)

## 1.1 slice( )로 특정 행만 불러오기

## 행번호를 활용해서 특정 행을 불러오기

## 햄 슬라이스...

slice(delivery, c(1,3,5:10))

## slice(데이터명, 잘라서가져올 행)

## 1, 3, 5~10 행만 불러오기

##1.2 filter( )로 조건에 맞는 데이터 행만 불러오기

filter(delivery, 시군구=='성북구')

filter(delivery, delivery$시군구=='성북구')

## { filter(데이터명, 조건) }

## '성북구' 데이터만 불러오기

filter(delivery, 시군구=='성북구', 요일 %in% c('토', '일'), 업종=='피자' | 통화건수>=100)

## 복수의 조건 사용

## filter(데이터명, 조건1, 조건2, 조건3, ...)

## 조건문은 논리연산(==, !=, >, < 등)을 활용

## 'OR'은 | 로 연결, 'AND'는 ,로 연결

##1.3 arrange( )로 정렬하기(오름차순)

arrange(delivery, 시군구, 요일, 업종)

## arrange(데이터명, 정렬기준변수1, 정렬기준변수2, ...)

arrange(delivery, desc(시군구), 요일, 업종)

## 내림차순(Descending)으로 정렬할 때는 'desc' 옵션 활용

##1.4 select( )로 변수를 선택하거나 제외하기

select(delivery, 통화건수)

# ":"를 활용한 순서대로 여러변수 선택하기

select(delivery, 시간대:통화건수)

# "-"를 활용한 변수제외

select(delivery, -요일)

##1.5 distinct( )로 반복 내용제거하기

distinct( delivery, 업종)

###################### 연습문제 ##########################

data(iris)

## 1.iris 데이터중 1부터 50행중 홀수, 100부터 150행중 짝수 선택

slice(??)

## 2.iris 데이터중 Species가 "setosa"이면서 Sepal.Length가 5보다 큰 값을 추출하시오

filter(??)

## 3.iris 데이터중 Sepal.Length는 내림차순 Sepal.Width는 오름차순으로 출력하시오

arrange(??)

## 4.iris 데이터중 "Sepal.Width" 와 "Species" 열을 선택하시오

select(??)

## 5.iris 데이터중 "Species"의 종류를 확인하시오

distinct(??)

###########################################################

##1.6 mutate( )로 기존 변수를 활용한 임시 변수 만들기

mutate(delivery, 새요일=paste0(요일, '요일'))

# 변수 추가

# delivery$새요일 = paste0(delivery$요일, '요일')

##1.7 count( )로 그룹별 개수 세기

count(delivery, 시군구)

##1.8 group\_by( )로 그룹 지정해주기

delivery\_grp = group\_by(delivery, 시군구)

##1.9 summarize( )로 요약 하기

summarise(delivery, mean(통화건수), m = min(통화건수), M = max(통화건수))

summarise(delivery\_grp, mean(통화건수), m = min(통화건수), M = max(통화건수))

## 원본 데이터는 전체 요약, 그룹이 지정된 데이터는 그룹별 요약

summarise(delivery\_grp, length(통화건수))

## "delivery %>% count(통화건수)"와 동일

##1.10 top\_n( )으로 상위 관측치 확인하기

top\_n(delivery, 5, 통화건수)

top\_n(delivery\_grp, 5, 통화건수)

###################### 연습문제 ##########################

data(iris)

## 6.iris 데이터중 "Sepal.Length" 와 "Sepal.Width" 두변수의 합을 Sepal\_sum이라는 변수에 저장하시오

mutate(??)

## 7.iris 데이터중 "Species"의 종별 개수를 확인하시오

count(??)

## 8.iris 데이터중 Sepal.Length의 합과 Sepal.Width의 평균을 구하시오

summarise(??)

## 9.iris 데이터중 "Petal.Width"의 상위 5개의 값을 출력

top\_n(??)

###########################################################

##1.11 파이프라인( %>% )을 활용한 연속작업

delivery %>%

filter(업종=='중국음식') %>%

group\_by(시군구) %>%

summarise(mean\_call = mean(통화건수)) %>%

arrange(desc(mean\_call))

# 데이터 저장

new\_data = delivery %>%

filter(업종=='중국음식') %>%

group\_by(시군구) %>%

summarise(mean\_call = mean(통화건수)) %>%

arrange(desc(mean\_call))

# 결과를 csv파일로 저장

write.csv(new\_data, 'result.csv', row.names=FALSE)

## 작업폴더(Working Directory)에 'result.csv'이름으로 저장

##1.12 ungroup( )의 활용

delivery %>%

filter(업종=='중국음식') %>%

group\_by(시간대, 시군구) %>%

summarise(mean\_call = mean(통화건수))

# 시군구별 상위 3대 시간대 확인

delivery %>%

filter(업종=='중국음식') %>%

group\_by(시간대, 시군구) %>%

summarise(mean\_call = mean(통화건수)) %>%

ungroup() %>%

group\_by(시군구) %>%

top\_n(3, mean\_call) %>%

arrange(시군구, desc(mean\_call))

## summarise( )로 요약할 떄의 그룹과

## top\_n( )등의 선택에서의 그룹이 다를 때는

## 중간에 ungroup( )을 넣어서 그룹 지정 해제

########################### 연습문제 #############################

##2 (실습) 보험료 데이터 요약하기

# 예제 데이터 불러오기

ins = read.csv('insurance.csv')

#1 데이터 ins에서 sex가 female인 관측치로 region별 관측치 수 계산

#2 charges가 10000이상인 관측치 중에서 smoker별 평균 age 계산

#3 age가 40 미만인 관측치 중에서 sex, smoker별 charges의 평균과 최댓값 계산

# 데이터를 csv파일로 저장하기

# 위에서 작업한 내용 중 3번을 csv파일로 저장해보기

########################################################################

##3 (실습) 국민건강보험공단 데이터 요약

# 국민건강보험공단 진료내역정보

## http://data.go.kr/dataset/15007115/fileData.do

## 원본 데이터에서 외래 진료 건과 주요 변수만 선택

load('NHIS.RData')

View(NHIS)

str(NHIS)

head(NHIS)

# IDV\_ID 가입자일련번호

# SEX 성별

# AGE\_GROUP 연령대코드

# FORM\_CD 서식코드

# DSBJT\_CD 진료과목코드

# MAIN\_SICK 주상병코드

# VSCN 요양일수

# RECN 입내원일수

# EDEC\_TRAMT 심결요양급여비용총액

# EDEC\_SBRDN\_AMT 심결본인부담금

# 성 / 연령대별 진료건수 계산 (진료건수 기준 내림차순 정렬)

NHIS %>%

group\_by(SEX, AGE\_GROUP) %>%

summarise(n=length(SEX)) %>%

arrange(desc(n))

NHIS %>%

dplyr::count(SEX, AGE\_GROUP) %>%

arrange(desc(n))

# 성별/연령대별 환자 분포 확인

## distinct() : 중복값 제거

NHIS %>%

dplyr::select(IDV\_ID, SEX, AGE\_GROUP) %>%

unique() %>%

dplyr::count(SEX, AGE\_GROUP)

############################ 연습 문제 ################################

# 문제1

# 성 / 연령대 / 진료과목별 환자수 계산 (환자수 기준 내림차순 정렬)

# 문제2

# 성별/연령대별 평균(요양일수/입내원일수/급여비용/본인부담금액) 계산 후 급여비용 내림차순으로 정렬

# 문제3

# 성별/연령대별 3개 최고빈도 주상병코드

## top\_n(n=k, wt=기준변수) : 기준변수를 기준으로 상위 k개 관측치 선택

## 주상병코드 조회 (MAIN\_SICK)

###### mutate의 확장

## 변수의 group별 비율을 아고싶다면??

data(iris)

head(iris)

## 도전해보자

## Species 별로 Sepal.Length의 비중을 아고싶다면??

#ex) prop 열 추가하려면?

# m group prop

# 2 a 0.2

# 3 a 0.3

# 5 a 0.5

# 1 b 0.2

# 1 b 0.2

# 3 b 0.6

######################

# 코드 작성

######################

###### mutate의 확장2

## group별로 번호를 매기고 싶다면??

#ex) prop 열 추가하려면?

# m group prop

# 2 a 1

# 3 a 2

# 5 a 3

# 1 b 1

# 1 b 2

# 3 b 3

iris %>%

arrange(Species,Sepal.Length) %>%

dplyr::group\_by(Species) %>%

dplyr::mutate(seq\_num = row\_number()) %>% data.frame()

# 도전해보자

# iris 데이터에서 Species별 Sepal.Width가 3번쨰로 큰 값들의 합은?

######################

# 코드 작성

######################

4일차\_문제정답

###################### 연습문제 ##########################

data(iris)

## 1.iris 데이터중 1부터 50행중 홀수, 100부터 150행중 짝수 선택

slice(iris,c(seq(1,50,2),seq(100,150,2)))

## 2.iris 데이터중 Species가 "setosa"이면서 Sepal.Length가 5보다 큰 값을 추출하시오

filter(iris,Species=="setosa",Sepal.Length>5)

## 3.iris 데이터중 Sepal.Length는 내림차순 Sepal.Width는 오름차순으로 출력하시오

arrange(iris,desc(Sepal.Length),Sepal.Width)

## 4.iris 데이터중 "Sepal.Width" 와 "Species" 열을 선택하시오

dplyr::select(iris,Sepal.Width,Species)

## 5.iris 데이터중 "Species"의 종류를 확인하시오

distinct(iris,Species)

###########################################################

###################### 연습문제 ##########################

data(iris)

## 6.iris 데이터중 "Sepal.Length" 와 "Sepal.Width" 두변수의 합을 Sepal\_sum이라는 변수에 저장하시오

mutate(iris,Sepal\_sum=Sepal.Length+Sepal.Width)

## 7.iris 데이터중 "Species"의 종별 개수를 확인하시오

count(iris,Species)

## 8.iris 데이터중 Sepal.Length의 합과 Sepal.Width의 평균을 구하시오

summarise(iris,sum=sum(Sepal.Length),mean=mean(Sepal.Width))

## 9.iris 데이터중 "Petal.Width"의 상위 5개의 값을 출력

top\_n(iris,5,Petal.Width)

###########################################################

# 예제 데이터 불러오기

ins = read.csv('insurance.csv')

#1 데이터 ins에서 sex가 female인 관측치로 region별 관측치 수 계산

ins %>%

filter(sex=='female') %>%

count(region)

#2 charges가 10000이상인 관측치 중에서 smoker별 평균 age 계산

ins %>%

filter(charges >= 10000) %>%

group\_by(smoker) %>%

summarise(mean(age))

#3 age가 40 미만인 관측치 중에서 sex, smoker별 charges의 평균과 최댓값 계산

ins %>%

filter(age < 40) %>%

group\_by(sex, smoker) %>%

summarise(mean(charges),

max(charges))

# 데이터를 csv파일로 저장하기

# 위에서 작업한 내용 중 3번을 csv파일로 저장해보기

result = ins %>%

filter(age < 40) %>%

group\_by(sex, smoker) %>%

summarise(mean(charges),

max(charges))

write.csv(result, 'result.csv')

############################ 연습 문제 ################################

# 문제1

# 성 / 연령대 / 진료과목별 환자수 계산 (환자수 기준 내림차순 정렬)

NHIS %>%

dplyr::select(SEX, AGE\_GROUP,DSBJT\_CD, IDV\_ID) %>%

unique() %>%

count(SEX, AGE\_GROUP,DSBJT\_CD) %>%

arrange(desc(n))

# 문제2

# 성별/연령대별 평균(요양일수/입내원일수/급여비용/본인부담금액) 계산 후 급여비용 내림차순으로 정렬

NHIS %>%

group\_by(SEX, AGE\_GROUP) %>%

summarise(m1=mean(VSCN), m2=mean(RECN), m3=mean(EDEC\_TRAMT), m4=mean(EDEC\_SBRDN\_AMT)) %>%

arrange(desc(m3))

# 문제3

# 성별/연령대별 3개 최고빈도 주상병코드

## top\_n(n=k, wt=기준변수) : 기준변수를 기준으로 상위 k개 관측치 선택

## 주상병코드 조회 (MAIN\_SICK)

NHIS %>%

count(SEX, AGE\_GROUP, MAIN\_SICK) %>%

group\_by(SEX, AGE\_GROUP) %>%

top\_n(n=3 , wt=n) %>%

arrange(SEX, AGE\_GROUP, desc(n))

# 같은표현

NHIS %>%

group\_by(SEX, AGE\_GROUP, MAIN\_SICK) %>%

summarise(n=n()) %>%

top\_n(n=3 , wt=n) %>%

arrange(SEX, AGE\_GROUP, desc(n))

## 도전해보자

## Species 별로 Sepal.Length의 비중을 아고싶다면??

#ex) prop 열 추가하려면?

# m group prop

# 2 a 0.2

# 3 a 0.3

# 5 a 0.5

# 1 b 0.2

# 1 b 0.2

# 3 b 0.6

######################

# 코드 작성

data(iris)

iris %>%

dplyr::group\_by(Species) %>%

dplyr::mutate(sum = sum(Sepal.Length)) %>%

dplyr::mutate(prop = Sepal.Length/sum) %>%

dplyr::select(Species,Sepal.Length,prop)

######################

# 도전해보자

# iris 데이터에서 Species별 Sepal.Width가 3번≳로 큰 값들의 합은?

######################

# 코드 작성

iris %>%

arrange(Species,Sepal.Width) %>%

dplyr::group\_by(Species) %>%

dplyr::mutate(seq\_num = row\_number()) %>%

filter(seq\_num==3)

######################

5일차

# 데이터 분석가 \_ james \

# \

# 스크립트 실행(Run a script) \

## : Windows : 'Ctrl + Enter' \

## : MAC : 'Command + Enter'\

#---------------------------------

# dplyr 복습하기

install.packages("hflights")

library(hflights)

data("hflights")

# 미국 휴스턴에서 출발하는 모든 비행기의 2011년 이착륙기록이 수록된 것으로 227,496건의 이착륙기록

head(hflights)

# 1월 20일 데이터 추출

filter(hflights, Month == 1, DayofMonth == 20)

# 데이터를 ArrDelay, Month, Year 순으로 정렬

arrange(hflights, ArrDelay, Month, Year)

##### 연습문제!

#문제1

# mutate를 사용해 delay라는 변수를 만들고 오름차순으로 정렬후 상위 20개의 평균을 구하시오

# delay = ArrDelay - DepDelay

#문제2

# 비행편수(TailNum)가 20편 이상, 평균 비행거리가 2000마일 이상 평균 연착시간의 평균을 구하시오

# 비행거리 : Distance

# 연착시간 : ArrDelay

##########################################

################# tidyr ##################

##########################################

##1 tidyr 패키지를 활용한 데이터 처리

## 데이터 정렬 형태 변경( wide <-> long )

# 예제 데이터 불러오기

head(delivery)

# tidyr 패키지 불러오기

install.packages('tidyr')

library(tidyr)

library(dplyr)

## 파이프라인(%>%)을 쓰기 위해서 불러오기

# group\_by( ) %>% summarise( )로 데이터 요약하기

aggr = delivery %>%

group\_by(시군구,시간대,요일,업종) %>%

summarise(통화건수=sum(통화건수)) %>%

as.data.frame()

aggr

##1.1 spread( )로 데이터를 여러 열로 나누기(long -> wide)

## spread(데이터이름, 기준변수이름, 나열할 값)

aggr %>%

spread(업종, 통화건수)

aggr\_wide = aggr %>% spread(업종, 통화건수)

aggr\_wide

##1.2 replace\_na( )로 결측값 처리하기

## replace\_na(list(변수1=값, 변수2=값, ...))

aggr\_wide %>% replace\_na(list(족발보쌈=999, 중국음식=0, 치킨=0))

aggr\_wide2 = aggr\_wide %>% replace\_na(list(족발보쌈=0, 중국음식=0, 치킨=0, 피자=0))

##1.3 drop\_na( )로 결측값을 포함한 관측치 버리기

## drop\_na(결측값을 찾을 변수1, 변수2, ...)

aggr\_wide %>% drop\_na()

aggr\_wide %>% drop\_na(치킨, 피자)

## 반대로!!

##1.4 gather( )로 여러 열을 한 열+구분변수로 만들기(wide->long)

## gather(데이터이름, 새기준변수이름, 새변수이름, 모을 변수들)

aggr\_wide2 %>% gather(Category, Count, 족발보쌈, 중국음식, 피자, 치킨)

aggr\_long = aggr\_wide2 %>% gather(Category, Count, -(시군구:요일))

## 순서대로 시군구부터 요일까지를 뺀 나머지 변수를 선택

aggr\_long

##1.5 complete( )로 빠져있는 조합 채우기

nrow(aggr\_wide2)

## 3947 != 4200 = 25(시군구)\*24(시간대)\*7(요일)

aggr\_wide2 %>% complete(시군구, 시간대, 요일)

# fill= 옵션으로 빈값 채우기

aggr\_wide2 %>% complete(시군구, 시간대, 요일, fill=list(족발보쌈=0, 중국음식=0, 치킨=0, 피자=0))

##2 (실습) 서울시 지하철 이용데이터

# 출처 : 공공데이터포털(www.data.go.kr)

# 데이터 불러오기

## 역변호가 150인 서울역 데이터

library(openxlsx)

subway\_2017 = read.xlsx('subway\_1701\_1709.xlsx')

subway\_2017

# 데이터의 구조 확인

str(subway\_2017)

# 첫 10개 관측치만 확인하기

head(subway\_2017, n=10)

# 변수이름 확인 ->이름변환

names(subway\_2017)

names(subway\_2017)[6:25]

substr(names(subway\_2017)[6:25], 1, 2)

## 첫 두 글자만 선택

paste0('H', substr(names(subway\_2017)[6:25], 1, 2))

## 앞에 'H'를 붙임

names(subway\_2017)[6:25] <- paste0('H', substr(names(subway\_2017)[6:25], 1, 2))

## '='을 활용해서 변수이름 업데이트

names(subway\_2017)

######################## 연습문제 #############################

# (실습) gather( ) 함수를 활용하여 H05부터 H24까지 변수를 모아

# '시간대'와 '승객수'으로 구분하는 데이터 subway2 만들기

subway2 = NA

## 위에서 만든 subway2 데이터와 dplyr 패키지를 활용하여

# 역명/시간대별 전체 승객수 합계 계산 (승객수 합계의 내림차순으로 정렬)

subway2 %>% NA

# 위의 결과를 spread( ) 함수를 활용해서 표 형태로 변환

subway2 %>% NA

# 역명/시간대/구분별 전체 승객수 합계 계산

subway2 %>% NA

# 2월 한달간 역명/시간대/구분별 전체 승객수 합계 계산

subway2 %>% NA

##############################################################################

###########################

###### 문자열 다루기 ######

###########################

## 기본 패키지 설치하기

iinstall.packages("stringr")

library(stringr)

# 큰따움표와 작은 땨움표

string1 <- "This is a string"

string2 <- "If I want to include a 'quote' inside a string, I use single quotes"

string2 <- 'If I want to include a "quote" inside a string, I use single quotes'

# 영원히 끝나지 않는 코드

"This is a string without a closing quote

"

#### 패턴찾기 ####

#str\_detect(데이터, 패턴)

x <- c("apple", "banana", "pear")

str\_detect(x, "e")

#str\_count(데이터, 패턴)

str\_count(x,"e")

#str\_which(데이터, 패턴)

str\_which(x,"e")

#str\_locate(데이터, 패턴)

str\_locate(x,"a")

### 부분집합 찾기

#str\_sub(데이터, 시작, 끝)

x <- c("Apple", "Banana", "Pear")

str\_sub(x, 1, 3)

str\_sub(x, -3, -1) #뒤에서부터

substr(x,1,3)

#str\_subset(데이터, 패턴)

str\_subset(x,"r")

### 문자열 변형하기

# str\_replace(데이터,찾는변수,바꿀변수)

# str\_replace\_all(데이터,찾는변수,바꿀변수)

str\_replace("apple","p","l")

str\_replace\_all("apple","p","l")

# Mutate STrings

str\_to\_lower("STRING") #소문자로

str\_to\_upper("string") #대문자로

str\_to\_title("string") #첫 글자만

### 연습해보기

# 문제 1

# words를 모두 대문자로 바꾼 상태에서 'AA'를 포함한 단어의 개수는 총 몇개이며 어떤단어들이 있는가?

# 문제 2

# words에서 "b"를 "a"로 모두 바꾸고 "aa"를 포함하는 단어 개수는?

# 문제 3

# words에서 "e"의 수는 전체 합과 평균은 몇인가?

#######################

###### 타입 변환 ######

#######################

as.factor(x)

# 주어진 객체 x를 팩터로 변환

as.numeric(x)

# 주어진 객체 x를 숫자를 저장한 벡터로 변환

as.character(x)

# 주어진 객체 x를 문자열을 저장한 벡터로 변환

as.matrix(x)

# 주어진 객체 x를 행렬로 변환

as.data.frame(x)

# 주어진 객체 x를 데이터 프레임으로 변환

x <- c("a", "b", "c")

as.factor(x)

as.character(as.factor(x))

x <- matrix(1:9, ncol=3)

as.data.frame(x)

###########################

###### 날짜 데이터 ######

###########################

install.packages('lubridate')

library(lubridate)

# 기본적으로 일반 텍스트 데이터를 날짜 데이터로 바꿉

as.Date('2020-01-01')

as.Date(2020-01-01) #텍스트가 전재!

as.Date(20200101)

as.Date('20200101')

# lubridate에 들어 있는 ymd() 함수는 어떤 모양이든 이를 날짜로 인식!

ymd('20200110')

mdy('January 10th 2020')

dmy('10-jan-2020')

ymd('820327')

ymd(820327)

# 날짜 데이터 뽑아내기

date\_test <- ymd(191020)

year(date\_test)

month(date\_test)

day(date\_test)

week(date\_test) #연도 기준 몇주차인지

wday(date\_test)

wday(date\_test,label = T)

# 날짜로 각종 계산하기

date\_test + days(100) #100일 후

date\_test + months(100) #100개월 후

date\_test + years(100) #100일 후

today()

today()-date\_test

# 날자 + 시간 데이터

ymd\_hm(20-10-20 14:30)

ymd\_hm("20-10-20 14:30")

date\_test2 <- ymd\_hm("20-10-20 14:30")

hour(date\_test2)

minute(date\_test2)

second(date\_test2)

##############################

### 연습해보기 날짜 연습해보기

head(subway\_2017)

summary(subway\_2017$날짜)

min(subway\_2017$날짜)

# 문제1

# (실습) gather( ) 함수를 활용하여 H05부터 H24까지 변수를 모아

# '시간대'와 '승객수'으로 구분하는 데이터 subway2 만들기

# subway2 데이터의 날짜에 시간을 추가하기 ex) "2017-01-02 06:00:00"

# 문제2

# subway\_2017 데이터에서 월과 일을 month, day 변수명으로 추가하시오

# 문제3

# 위에서 추가한 변수들 기반으로 3월중 가장 많이 탑승한 시간은 몇시인가?

########## 결측치 처리 #############

# 결측치(Missing Value)

# 누락된 값, 비어있는 값

# 함수 적용 불가, 분석 결과 왜곡

# 제거 후 분석 실시

# 결측치 확인하기

df <- data.frame(sex = c("M", "F", NA, "M", "F"),

score = c(5, 4, 3, 4, NA))

is.na(df)

table(is.na(df))

table(is.na(df$sex))

summary(df)

apply(X = df, MARGIN = 2, FUN = function(x){sum(is.na(x))})

sapply(X = df, FUN = function(x){sum(is.na(x))})

# 결측치 제거

library(dplyr) # dplyr 패키지 로드

df %>% filter(is.na(score))

df %>% filter(!is.na(score)) # score 결측치 제거

df %>% filter(!is.na(score) & !is.na(sex))

df\_nomiss2 <- na.omit(df) # 모든 변수에 결측치 없는 데이터 추출

df\_nomiss2 # 출력

# 결측치 대체하기

df

df$score <- ifelse(is.na(df$score), 4, df$score) # math가 NA면 55로 대체

table(is.na(df$score)) # 결측치 빈도표 생성

mpg <- as.data.frame(ggplot2::mpg) # mpg 데이터 불러오기

mpg[c(65, 124, 131, 153, 212), "hwy"] <- NA # NA 할당하기

apply(mpg,2,function(x){sum(is.na(x))})

mpg[is.na(mpg$hwy),"hwy"] <- mean(mpg$hwy,na.rm = T)

iinstall.packages("zoo")

library(zoo)

na.locf0(c(NA, NA, "A", NA, "B"), fromLast = FALSE) # 1

## [1] NA NA "A" "A" "B"

na.locf0(c(NA, NA, "A", NA, "B"), fromLast = TRUE) # 2

## [1] "A" "A" "A" "B" "B"

##############################

####### 결측치 처리해보기!!

data(airquality)

head(airquality)

# 1. airquality 데이터의 결측치 개수를 구하시오 (열별로)

# 2. 결측치가 있는 행들을 제거한 후 각 열의 평균을 구하시오

# 3. 결측치는 변수의 중앙값으로 대체후 각 열의 평균을 출력하시오

5일차\_문제정답

##### 연습문제!

#문제1

# mutate를 사용해 delay라는 변수를 만들고 오름차순으로 정렬후 상위 20개의 평균을 구하시오

# delay = ArrDelay - DepDelay

hflights %>%

mutate(delay = ArrDelay - DepDelay) %>%

arrange(delay) %>%

head(20) %>%

summarise(mean = mean(delay))

#문제2

# 비행편수(TailNum)가 20편 이상, 평균 비행거리가 2000마일 이상 평균 연착시간의 평균을 구하시오

# 비행거리 : Distance

# 연착시간 : ArrDelay

hflights %>%

group\_by(TailNum) %>%

dplyr::summarise(num = n(),

dist = mean(Distance,na.rm = T),

delay = mean(ArrDelay,na.rm = T)) %>%

filter(num>=20 , dist >= 1000) %>%

summarise(mean = mean(delay))

#############################################

######################## 연습문제 #############################

# (실습) gather( ) 함수를 활용하여 H05부터 H24까지 변수를 모아

# '시간대'와 '승객수'으로 구분하는 데이터 subway2 만들기

subway2 = gather(subway\_2017, 시간대, 승객수, H05:H24)

## 위에서 만든 subway2 데이터와 dplyr 패키지를 활용하여

# 역명/시간대별 전체 승객수 합계 계산 (승객수 합계의 내림차순으로 정렬)

subway2 %>%

group\_by(역명, 시간대) %>%

summarise(SUM = sum(승객수)) %>%

arrange(desc(SUM))

# 위의 결과를 spread( ) 함수를 활용해서 표 형태로 변환

subway2 %>%

group\_by(역명, 시간대) %>%

summarise(SUM = sum(승객수)) %>%

spread(시간대, SUM)

# 역명/시간대/구분별 전체 승객수 합계 계산

subway2 %>%

group\_by(역명, 시간대, 구분) %>%

summarise(SUM = sum(승객수)) %>%

arrange(desc(SUM))

# 2월 한달간 역명/시간대/구분별 전체 승객수 합계 계산

subway2 %>%

filter(월==2) %>%

group\_by(역명, 시간대, 구분) %>%

summarise(SUM = sum(승객수)) %>%

arrange(desc(SUM))

#############################################

### 연습해보기

# 문제 1

# words를 모두 대문자로 바꾼 상태에서 'AA'를 포함한 단어의 개수는 총 몇개이며 어떤단어들이 있는가?

sum(str\_detect(str\_to\_upper(words), "AB")) #8개

str\_to\_upper(words)[str\_which(str\_to\_upper(words),"AB")] #단어 확인

# 문제 2

# words에서 "b"를 "a"로 모두 바꾸고 "aa"를 포함하는 단어 개수는?

sum(str\_detect(str\_replace\_all(words,"b","a"), "aa")) #19개

# 문제 3

# words에서 "e"의 수는 전체 합과 평균은 몇인가?

sum(str\_count(words,"e"))

mean(str\_count(words,"e"))

#############################################

##############################

### 연습해보기 날짜 연습해보기

head(subway\_2017)

summary(subway\_2017$날짜)

min(subway\_2017$날짜)

# 문제1

# (실습) gather( ) 함수를 활용하여 H05부터 H24까지 변수를 모아

# '시간대'와 '승객수'으로 구분하는 데이터 subway2 만들기

# subway2 데이터의 날짜에 시간을 추가하기 ex) "2017-01-02 06:00:00"

subway2 <- gather(subway\_2017,"시간","고객",H05:H24)

ymd\_h(paste0(subway\_2017$날짜,substr(subway2$시간,2,3)))

# 문제2

# subway\_2017 데이터에서 월과 일을 month, day 변수명으로 추가하시오

subway\_2017$month <-month(subway\_2017$날짜)

subway\_2017$day <-day(subway\_2017$날짜)

# 문제3

# 위에서 추가한 변수들 기반으로 3월중 가장 많이 탑승한 시간은 몇시인가?

subway\_2017 %>%

filter(month == 3) %>%

select(H06:H24) %>%

apply(2,sum) %>%

data.frame()

##############################

####### 결측치 처리해보기!!

data(airquality)

head(airquality)

# 1. airquality 데이터의 결측치 개수를 구하시오 (열별로)

apply(X = airquality, MARGIN = 2, FUN = function(x){sum(is.na(x))})

# 2. 결측치가 있는 행들을 제거한 후 각 열의 평균을 구하시오

airquality2 <- na.omit(airquality)

apply(airquality2,2,mean)

# 3. 결측치는 변수의 중앙값으로 대체후 각 열의 평균을 출력하시오

airquality[is.na(airquality$Ozone),"Ozone"] <- median(airquality$Ozone,na.rm=T)

airquality[is.na(airquality$Solar.R),"Solar.R"] <- median(airquality$Solar.R,na.rm=T)

apply(airquality,2,mean)

6일차

# 데이터 분석가 \_ james \

# \

# 스크립트 실행(Run a script) \

## : Windows : 'Ctrl + Enter' \

## : MAC : 'Command + Enter'\

#---------------------------------

##0 색상

# plot 함수로 색깔 점 찍기

plot(0,0, pch=16, cex=10, col='black')

plot(0,0, pch=16, cex=10, col='pink')

plot(0,0, pch=16, cex=10, col='dodgerblue')

## 일반적으로 "col=" 옵션으로 색상 변경 가능

## 색상이름은 아래 참고

## http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf

# rgb( ) 함수와 "#RRGGBB" HEX코드 활용

rgb( 0/255, 70/255, 42/255)

## Ewha Green

plot(0,0, pch=16, cex=10, col='#00462A')

# RColorBrewer 패키지의 활용

install.packages('RColorBrewer')

library(RColorBrewer)

## http://colorbrewer2.org/

# 패키지 내 모든 색상조합 확인

display.brewer.all()

## 색상조합 이름 확인

brewer.pal(9, 'Set1')

brewer.pal(9, 'Blues')

brewer.pal(9, 'YlGnBu')

brewer.pal(9, 'Spectral')

##1 ggplot2 패키지를 활용한 시각화

# ggplot2 패키지 설치, 불러오기

install.packages('ggplot2')

library(ggplot2)

# 데이터 요약/처리를 위한 패키지도 불러오기

library(dplyr)

library(tidyr)

install.packages("gapminder")

library(gapminder)

data(gapminder)

data1 <- gapminder[gapminder$year=="2007",]

####################

## 1. 그릴 부분의 도와지를 그려본다. (aes(x = , y=))

ggplot(data1) +

aes(x = gdpPercap) + #x축 지정

aes(y = lifeExp) #y축 지정

# 이렇게 한번에 그릴 수 있다.

# ggplot(data1,aes(x=gdpPercap,y=lifeExp))

####################

## 2. 그림을 선택한다. +geom\_point

ggplot(data1) +

aes(x = gdpPercap) + #x축 지정

aes(y = lifeExp) + #y축 지정

geom\_point() #나타낼 그림

####################

## 3. 그림을 꾸며준다

## 3-1 색을 지정한다 aes(color = )

ggplot(data1) +

aes(x = gdpPercap) + #x축 지정

aes(y = lifeExp) + #y축 지정

geom\_point() + #나타낼 그림

aes(color = continent) #색 지정

#같은 표현

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color=continent)) + geom\_point()

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp)) + geom\_point(aes(color=continent))

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp)) + geom\_point(color = "red")

## 불가능

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color = "red")) + geom\_point()

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, fill = continent)) + geom\_point()

####################

## 3-2 모양 지정한다 aes(shape = )

ggplot(data1) +

aes(x = gdpPercap) + #x축 지정

aes(y = lifeExp) + #y축 지정

geom\_point() + #나타낼 그림

aes(color = continent) + #색 지정

aes(shape = continent) #모양 지정

# 같은표현

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color = continent, shape = continent)) + geom\_point()

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color = continent)) + geom\_point(aes(shape = continent))

# 특정모양 지정

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color = continent)) + geom\_point(shape = 3)

####################

## 3-3 크기 지정한다 aes(size = )

ggplot(data1) +

aes(x = gdpPercap) + #x축 지정

aes(y = lifeExp) + #y축 지정

geom\_point() + #나타낼 그림

aes(color = continent) + #색 지정

aes(shape = continent) + #모양 지정

aes(size = pop) #크기 지정

# 같은표현

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color = continent, shape = continent, size = pop)) + geom\_point()

# 특정 크기 지정

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color = continent, shape = continent)) + geom\_point(size = 3)

####################

## 3-4 투명도를 지정한다 aes(alpha = )

ggplot(data1) +

aes(x = gdpPercap) + #x축 지정

aes(y = lifeExp) + #y축 지정

geom\_point() + #나타낼 그림

aes(color = continent) + #색 지정

aes(shape = continent) + #모양 지정

aes(size = pop) + #크기 지정

aes(alpha = lifeExp) #투명도

# 같은표현

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color = continent, shape = continent, size = pop, alpha = lifeExp)) + geom\_point()

# 특정 크기 지정

data1 %>% ggplot(aes(x=gdpPercap, y=lifeExp, color = continent, shape = continent, size = pop)) + geom\_point(alpha = 0.5)

###########################################

############### 연습해보기 ###############

###########################################

head(insurance)

#1. bmi에 따라서 charges가 어떻게 변하는지 점그래프를 그리시오

## region을 색으로 지정

## sex를 모양으로 지정

## 투명도는 0.7

#2. age에 따라서 charges가 어떻게 변하는지 점그래프를 그리시오

## bmi 색으로 지정

## smoker를 모양으로 지정

###########################################

## 막대그래프

######################### 막대그래프

#1. 도화지 그리기

ggplot(data1) +

aes(x = continent) # x축 지정

#2. 그림 그리기

ggplot(data1) +

aes(x = continent) + # x축 지정

geom\_bar() # 막대그래프 그리기

#3. 꾸미기

ggplot(data1) +

aes(x = continent) + # x축 지정

geom\_bar() + # 막대그래프 그리기

aes(fill = continent) # 전체색

ggplot(data1) +

aes(x = continent) + # x축 지정

geom\_bar() + # 막대그래프 그리기

aes(fill = continent) + # 전체색

scale\_fill\_brewer(palette='Set1') #팔레트 사용하기

#######

# 주의!

# 막대그래프는 color이 아닌 fill로 사용!

ggplot(data1) +

aes(x = continent) + # x축 지정

geom\_bar() + # 막대그래프 그리기

aes(color = continent) # 개별색

#######

##### x와 y를 모두 지정해주고 싶으면? stat = "identity"

ggplot(data1) +

aes(x = continent) + # x축 지정

aes(y = lifeExp) + # y축 지정

geom\_bar(stat = "identity") + # 막대그래프 x,y축

aes(fill = continent) # 전체색

# 주의

ggplot(data1) +

aes(x = continent) + # x축 지정

aes(y = lifeExp) + # y축 지정

geom\_bar(stat = "identity") + # 막대그래프 x,y축

aes(color = continent) # 전체색

##### 데이터 전처리와 막대 차트 그리기

# continent 마다 평균을 그리고 싶으면??

data1 %>%

group\_by(continent) %>%

summarise(mean = mean(lifeExp))

data1 %>%

group\_by(continent) %>%

dplyr::summarise(mean = mean(lifeExp)) %>%

ggplot() +

aes(x = continent) +

aes(y = mean) +

geom\_bar(stat = "identity") +

aes(fill = continent) +

aes(alpha = 0.7)

# 나눠서 그리는 방법!

gapminder %>%

filter(year %in% c(2002,2007)) %>%

group\_by(continent,year) %>%

dplyr::summarise(mean = mean(lifeExp)) %>%

ggplot() +

aes(x = continent) +

aes(y = mean) +

geom\_bar(stat = "identity") +

aes(color = continent) +

aes(fill = continent) +

facet\_grid(~year) # 특정 변수로 구분해서 그리고 싶다면?

###########################################

############### 연습해보기 ###############

###########################################

head(insurance)

#1. insurance 데이터에서 region별 중앙값을 구한후 막대그래프를 그리시고

## region을 색으로 지정

## 투명도는 0.7

#2. insurance 데이터에서 sex, smoker별 중앙값을 구한후 막대그래프를 그리시고

## x축은 smoker이며 sex를 색으로 구분

## region을 색으로 지정

## 투명도는 0.7

######################### 박스 그래프 geom\_boxplot()

gapminder %>%

ggplot(aes(x=continent, y= lifeExp)) +

geom\_boxplot()

gapminder %>%

ggplot(aes(x=continent, y= lifeExp, fill= continent)) +

geom\_boxplot()

gapminder %>%

ggplot(aes(x=continent, y= lifeExp, fill= continent)) +

geom\_boxplot(alpha = 0.5)

# 주의! 요약을 한 데이터를 사용하지 않는다!

gapminder %>%

group\_by(continent) %>%

dplyr::summarise(mean = mean(lifeExp)) %>%

ggplot(aes(x=continent, y= mean, fill= continent)) + geom\_boxplot()

######################### 히스토그램 geom\_boxplot()

gapminder %>%

ggplot(aes(x=lifeExp)) +

geom\_histogram()

gapminder %>%

ggplot(aes(x=lifeExp)) +

geom\_histogram() +

facet\_grid(~continent)

######################### 선 그래프

gapminder %>%

group\_by(year) %>%

summarise(sum = sum(lifeExp))

gapminder %>%

group\_by(year) %>%

dplyr::summarise(sum = sum(lifeExp)) %>%

ggplot(aes(x=year,y=sum)) + geom\_line()

# 여러 그룹을 그리고 싶을 경우

gapminder %>%

group\_by(year,continent) %>%

summarise(mean = mean(lifeExp))

gapminder %>%

group\_by(year,continent) %>%

dplyr::summarise(mean = mean(lifeExp)) %>%

ggplot(aes(x=year, y=mean , group=continent ,color= continent)) + geom\_line()

###########################################

############### 연습해보기 ###############

###########################################

#1 insurance데이터에서 children이 0보다 크면 1, 0이면 0인 변수 ch\_data를 만드시오

#2 insurance데이터를 활용해서 막대그래프를 그리시오

## x축은 region y축은 charges이며 ch\_data를 색으로 구분

#3 insurance데이터를 활용해서 막대그래프를 그리시오

## x축은 charges ch\_data를 색으로 구분

## region마다 4개의 그래프를 그리시오

#4 insurance데이터를 활용해서 막대그래프를 그리시오

## x축은 region y축은 charges이며 ch\_data를 색으로 구분

## (누적 막대그래프와 ch\_data별 비교 막대그래프)

### ggplot 추가

HR <- read.csv("HR\_comma\_sep.csv")

HR$left = as.factor(HR$left)

HR$salary = factor(HR$salary,levels = c("low","medium","high"))

# satisfaction\_level : 직무 만족도

# last\_evaluation : 마지막 평가점수

# number\_project : 진행 프로젝트 수

# average\_monthly\_hours : 월평균 근무시간

# time\_spend\_company : 근속년수

# work\_accident : 사건사고 여부(0: 없음, 1: 있음, 명목형)

# left : 이직 여부(0: 잔류, 1: 이직, 명목형)

# promotion\_last\_5years: 최근 5년간 승진여부(0: 승진 x, 1: 승진, 명목형)

# sales : 부서

# salary : 임금 수준

#####################

### 테마 변경하기 ###

#####################

library(ggthemes)

# Classic Theme

ggplot(HR,aes(x=salary)) +

geom\_bar(aes(fill=salary)) +

theme\_classic()

# BW Theme

ggplot(HR,aes(x=salary)) +

geom\_bar(aes(fill=salary)) +

theme\_bw()

Graph = ggplot(HR,aes(x=salary)) +

geom\_bar(aes(fill=salary))

## 패키지를 통한 다양한 테마 변경

Graph + theme\_bw() + ggtitle("Theme\_bw")

Graph + theme\_classic() + ggtitle("Theme\_classic")

Graph + theme\_dark() + ggtitle("Theme\_dark")

Graph + theme\_light() + ggtitle("Theme\_light")

Graph + theme\_linedraw() + ggtitle("Theme\_linedraw")

Graph + theme\_minimal() + ggtitle("Theme\_minimal")

Graph + theme\_test() + ggtitle("Theme\_test")

Graph + theme\_void() + ggtitle("Theme\_vold")

#####################

### 범례제목 수정 ###

#####################

ggplot(HR,aes(x=salary)) +

geom\_bar(aes(fill=salary)) +

theme\_bw() +

labs(fill = "범례 제목 수정(fill)")

# 범례 테두리 설정

Graph + theme(legend.position = "top")

Graph + theme(legend.position = "bottom")

Graph + theme(legend.position = c(0.9,0.7))

Graph + theme(legend.position = 'none')

#####################

### 축 변경 ###

#####################

# 이산형 - deiscreate()

# 연속형 - continuous()

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary)) +

theme\_bw() +

scale\_x\_discrete(labels = c("하","중","상")) +

scale\_y\_continuous(breaks = seq(0,8000,by = 1000))

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary)) +

theme\_bw() +

scale\_x\_discrete(labels = c("하","중","상")) +

scale\_y\_continuous(breaks = seq(0,8000,by = 1000)) +

scale\_fill\_discrete(labels = c("하","중","상"))

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary)) +

theme\_bw() +

ylim(0,5000)

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary)) +

theme\_bw() +

ylim(0,13000)

#####################

### 색 변경 ###

#####################

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary)) +

theme\_bw() +

scale\_fill\_manual(values = c('red','royalblue','tan'))

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary), alpha = 0.4) +

theme\_bw() +

scale\_fill\_manual(values = c('red','royalblue','tan'))

#####################

### 글자크기,각도 수정 ###

#####################

# coord\_flip() : 대칭 그래프

# theme\_bw : 글자체 수정

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary), alpha = 0.4) +

theme\_bw() +

scale\_fill\_manual(values = c('red','royalblue','tan')) +

coord\_flip()

ggplot(HR,aes(x = salary)) +

geom\_bar(aes(fill = salary)) +

theme\_bw() +

scale\_fill\_manual(values = c('red','royalblue','tan')) +

coord\_flip() +

theme(legend.position = 'none',

axis.text.x = element\_text(size = 15,angle = 90),

axis.text.y = element\_text(size = 15),

legend.text = element\_text(size = 15))

# 그래프에 평행선, 수직선, 대각선을 그릴 수 있는 명령어

ggplot(NULL) +

geom\_vline(xintercept = 10,

col = 'royalblue', size = 2) +

geom\_hline(yintercept = 10, linetype = 'dashed',

col = 'royalblue', size = 2) +

geom\_abline(intercept = 0, slope = 1, col = 'red',

size = 2) +

theme\_bw()

#### 추가 유용한 그래프

###################열지도(heatmap)

# 데이터 요약

agg2 = insurance %>%

mutate(bmi\_grp = cut(bmi,

breaks=c(0,30,35,40,100),

labels=c('G1','G2','G3','G4'))) %>%

group\_by(region, bmi\_grp) %>%

summarise(Q90 = quantile(charges, 0.9))

quantile(iris$Sepal.Width,0.5) #중위수

quantile(iris$Sepal.Width,0.7) #70%

## quantile( , q) : q\*100% 값 계산

agg2 %>%

ggplot(aes(x=region, y=bmi\_grp, fill=Q90)) +

geom\_tile()

# 색상 지정

agg2 %>%

ggplot(aes(x=region, y=bmi\_grp, fill=Q90)) +

geom\_tile() +

scale\_fill\_gradient(low='white', high='#FF6600')

agg2 %>%

ggplot(aes(x=region, y=bmi\_grp, fill=Q90)) +

geom\_tile() +

scale\_fill\_distiller(palette='YlGnBu')

###########################################

############### 연습해보기 ###############

###########################################

# (실습) NHIS에서 AGE\_GROUP, DSBJT\_CD별 EDEC\_TRAMT 평균 계산 후 저장

# 저장된 데이터로 열지도 시각화

###########################################

# tidyr + dplyr + ggplot을 한번에

# 데이터 불러오기

## 역변호가 150인 서울역 데이터

library(openxlsx)

subway\_2017 = read.xlsx('subway\_1701\_1709.xlsx')

names(subway\_2017)[6:25] <- paste0('H', substr(names(subway\_2017)[6:25], 1, 2))

head(subway\_2017)

# gather( ) 함수를 활용하여 H05부터 H24까지 변수를 모아

# '시간대'와 '승객수'으로 구분하는 데이터 subway2 만들기

subway2 = gather(subway\_2017, 시간대, 승객수, H05:H24)

## 위에서 만든 subway2 데이터와 dplyr 패키지를 활용하여

# 역명/시간대별 전체 승객수 합계 계산 (승객수 합계의 내림차순으로 정렬)

subway2 %>%

group\_by(역명, 시간대) %>%

summarise(SUM = sum(승객수)) %>%

arrange(desc(SUM))

### 이러한 tidyr을 통해서 데이터를 시각화하기

### 시간대별로 승객 합계 막대차트로 그려보기!

# options("scipen" = 100)

6일차\_문제정답

#####################################

########### 연습해보기 ################

##################################

###

head(insurance)

#1. bmi에 따라서 charges가 어떻게 변하는지 점그래프를 그리시오

## region을 색으로 지정

## sex를 모양으로 지정

## 투명도는 0.7

insurance %>%

ggplot(aes(x=bmi,y=charges,color=region,shape=sex)) +

geom\_point(alpha=0.7)

#2. age에 따라서 charges가 어떻게 변하는지 점그래프를 그리시오

## bmi 색으로 지정

## smoker를 모양으로 지정

insurance %>%

ggplot(aes(x=age,y=charges,color=bmi,shape=smoker)) +

geom\_point()

###########################################

############### 연습해보기 ###############

###########################################

head(insurance)

#1. insurance 데이터에서 region별 중앙값을 구한후 막대그래프를 그리시고

## region을 색으로 지정

## 투명도는 0.7

insurance %>%

group\_by(region) %>%

summarise(mean = mean(charges)) %>%

ggplot(aes(x=region,y=mean,fill=region)) +

geom\_bar(stat="identity")

#2. insurance 데이터에서 sex, smoker별 중앙값을 구한후 막대그래프를 그리시고

## x축은 smoker이며 sex를 색으로 구분

## region을 색으로 지정

## 투명도는 0.7

insurance %>%

group\_by(sex,smoker) %>%

summarise(mean = mean(charges)) %>%

ggplot(aes(x=smoker,y=mean,fill=sex)) +

geom\_bar(stat="identity")

insurance %>%

group\_by(sex,smoker) %>%

summarise(mean = mean(charges)) %>%

ggplot(aes(x=smoker,y=mean,fill=sex)) +

geom\_bar(stat="identity", position = "dodge")

###########################################

############### 연습해보기 ###############

###########################################

#1 insurance데이터에서 children이 0보다 크면 1, 0이면 0인 변수 ch\_data를 만드시오

insurance <- insurance %>%

mutate(ch\_data = ifelse(children==0,0,1))

#2 insurance데이터를 활용해서 막대그래프를 그리시오

## x축은 region y축은 charges이며 ch\_data를 색으로 구분

insurance %>%

ggplot(aes(x=region,y=charges,fill=factor(ch\_data))) + geom\_boxplot()

#3 insurance데이터를 활용해서 막대그래프를 그리시오

## x축은 charges ch\_data를 색으로 구분

## region마다 4개의 그래프를 그리시오

insurance %>%

ggplot(aes(x=charges,fill=factor(ch\_data))) + geom\_histogram() +

facet\_grid(~region)

#4 insurance데이터를 활용해서 막대그래프를 그리시오

## x축은 region y축은 charges이며 ch\_data를 색으로 구분

## (누적 막대그래프와 ch\_data별 비교 막대그래프)

insurance %>%

ggplot(aes(x=region, y=charges, fill=factor(ch\_data))) +

geom\_bar(stat="identity")

insurance %>%

ggplot(aes(x=region, y=charges, fill=factor(ch\_data))) +

geom\_bar(stat="identity",position = "dodge")

options("scipen" = 100)

###########################################

############### 연습해보기 ###############

###########################################

# (실습) NHIS에서 AGE\_GROUP, DSBJT\_CD별 EDEC\_TRAMT 평균 계산 후 저장

# 저장된 데이터로 열지도 시각화

# agg3 = NHIS %>%

# group\_by(AGE\_GROUP, DSBJT\_CD) %>%

# summarise(mean\_AMT = mean(EDEC\_TRAMT))

# agg3

#

#

# agg3 %>%

# ggplot(aes(AGE\_GROUP, DSBJT\_CD, fill=mean\_AMT)) +

# geom\_tile() +

# scale\_fill\_distiller(palette='Spectral')

############################################################

### 이러한 tidyr을 통해서 데이터를 시각화하기

### 시간대별로 승객 합계 막대차트로 그려보기!

subway2 %>%

group\_by(역명, 시간대) %>%

summarise(SUM = sum(승객수)) %>%

arrange(desc(SUM)) %>%

ggplot(aes(x=시간대,y=SUM,fill=시간대)) + geom\_bar(stat="identity")

7일차

# 데이터 분석가 \_ james \

# \

# 스크립트 실행(Run a script) \

## : Windows : 'Ctrl + Enter' \

## : MAC : 'Command + Enter'\

#---------------------------------

# 기초통계

#평균

mean(

x,

na.rm=FALSE, # 평균 계산 전 NA를 제거할지 여부

)

#분산

var(

x,

na.rm=FALSE,

)

#표준편차

sd(

x,

na.rm=FALSE,

)

#사분위수

quantile(1:10,0.7)

####

mean(1:5)

var(1:5)

sum((1:5-mean(1:5))^2)/(5-1) # 분모로 n-1이 사용됨을 확인할 수 있음

sd(1:5)

summary(1:11)

########### 범주형

table(

... # 팩터로 해석할 수 있는 하나 이상의 객체

)

which.max(

x # 숫자 벡터

)

(x <- factor(c("a", "b", "c", "c", "c", "d", "d")))

table(x) # 팩터의 각 레벨(level)별 빈도수를 구한다.

which.max(table(x)) # 가장 큰 값이 저장된 셀의 색인은 3

names(table(x))[3] # 가장 큰 값이 저장된 셀의 이름

# 주변비율

prop.table(

x,

margin=NULL

)

x1 <- data.frame(x1=sample(1:5,20,replace = T),x2=sample(1:5,20,replace = T))

prop.table(table(x1))

prop.table(table(x1),1)

prop.table(table(x1),2)

##추출

sample(

x, # 표본을 뽑을 데이터 벡터. 만약 길이 1인 숫자 n이 지정되면 1:n에서 표본이 선택된다.

size, # 표본의 크기

replace=FALSE, # 복원 추출4 여부

# 데이터가 뽑힐 가중치. 예를 들어, x=c(1, 2, 3)에서 2개의 표본을 뽑되 각 표본이 뽑힐 확률을

# 50%, 20%, 30%로 하고자 한다면 size=2, prob=c(5, 2, 3)을 지정한다.

# prob나 prob에 지정한 값의 합이 1일 필요는 없다.

prob=NULL

)

sample(1:10, 5)

sample(1:10, 5, replace=TRUE)

## 층화추출

install.packages("sampling")

library(sampling)

sampling::strata(

data, # 데이터 프레임 또는 행렬

stratanames=NULL, # 층화 추출에 사용할 변수들

size, # 각 층의 크기

# - srswor : 비복원 단순 임의 추출(Simple Random Sampling WithOut Replacement)

method=c("srswor")

)

(x <- strata(c("Species"), size=c(3, 3, 3), method="srswor", data=iris))

strata(c("Species"), size=c(3, 1, 1), method="srswr", data=iris)

###### 이산확률 분포

library(ggplot2)

# 난수 생성

RB = sample( c(0,1) , 400 , prob = c(0.4,0.6), replace=T)

ggplot(NULL) +

geom\_bar(aes(x = as.factor(RB), fill = as.factor(RB))) +

theme\_bw() +

xlab("") + ylab("") +

scale\_x\_discrete(labels = c("실패","성공")) +

theme(legend.position = 'none')

#이항분포 추출

# 동전을 5번 던졌을 경우 앞면이 나오는 횟수

rbinom(n=100,size=5,prob=0.5)

# n = 시도 횟수

# size = 동전을 던진 횟수

# prop = 성공할 확률

data.frame(n = rbinom(n=10,size=5,prob=0.5)) %>% ggplot(aes(x=n))+geom\_bar()

data.frame(n = rbinom(n=30,size=5,prob=0.5)) %>% ggplot(aes(x=n))+geom\_bar()

data.frame(n = rbinom(n=50,size=5,prob=0.5)) %>% ggplot(aes(x=n))+geom\_bar()

data.frame(n = rbinom(n=100,size=5,prob=0.5)) %>% ggplot(aes(x=n))+geom\_bar()

data.frame(n = rbinom(n=10000,size=5,prob=0.5)) %>% ggplot(aes(x=n))+geom\_bar()

# 확률을 확인

dbinom(x=3,size=5,prob=0.5) #3번 앞면

dbinom(x=4,size=5,prob=0.5) #4번 앞면

# 누적 확률 확인

pbinom(q=2, size = 5, prob = 0.5) # 2 이하 성공

pbinom(q=3, size = 5, prob = 0.5) # 3 이하 성공

###############

#연속 확률 분포

# 구간으로 정의되는 분포

R = rnorm(n = 100000, mean = 0, sd = 1)

ggplot(NULL) +

geom\_histogram(aes(x = R, y= ..density..),binwidth = 0.2,fill = "white",col = 'black') +

scale\_y\_continuous(expand = c(0,0),limits = c(0,0.5)) +

scale\_x\_continuous(limits = c(-3,3)) +

xlab("")

# 정규분포

rnorm() # 평균과 분산에 해당하는 랜덤 샘플

dnorm() # 확률 밀도함수

pnorm() # 누적 분포함수

qnorm() # 분쉬수 함수

# 평균이 10이고 표준편차가 2인 샘플 30개가 필요해

rnorm(n=30, mean = 10, sd =2)

#

dnorm(1.96,mean = 0, sd=1) # 1.96의 밀도함수는?

qnorm(0.975,mean = 0, sd=1) # 0.975 누적합수의 x축은?

#1.96까지의 누적 분포함순는?

pnorm(1.96)

# install.packages("reshape")

library(reshape)

library(dplyr)

k1 = c()

p1 = c()

for(k in seq(-15,15,by = 0.01)){

p = dnorm(x = k,mean = 0,sd = 1)

k1 = c(k1,k)

p1 = c(p1,p)

}

k2 = c()

p2 = c()

for(k in seq(-15,15,by = 0.01)){

p = dnorm(x = k,mean = 0,sd = 5)

k2 = c(k2,k)

p2 = c(p2,p)

}

DF = data.frame(

k = k1,

p1 = p1,

p2 = p2

)

DF %>%

gather("variable","value",p1:p2) %>%

ggplot() +

geom\_line(aes(x = k, y = value, col = as.factor(variable))) +

geom\_vline(xintercept = 0,linetype = 'dashed') +

theme(legend.position = 'none') +

scale\_y\_continuous(expand = c(0,0))

x1 = rnorm(n = 1000, mean = 20, sd = 5)

x2 = scale(x1)

DF = data.frame(

x1 = x1,

x2 = x2

)

DF %>%

gather("variable","value") %>%

mutate(variable = ifelse(variable == "x1", "비표준화","표준화")) %>%

ggplot() +

geom\_density(aes(x = value, fill = variable), alpha = 0.4) +

theme\_bw() +

theme(legend.position = c(0.8,0.6)) +

xlab("") + ylab("") + labs(fill = "")

# 문제) 어느 실험실의 연구원이 어떤 식물로부터 하루 동안 추출하는 호르몬의 양은 평균이 30.2mg,

# 표준편차가 0.6mg인 정규분포를 따른다고 한다. 어느 날 이 연구원이 하루 동안 추출한 호르몬의 양이 29.6mg 이상이고

# 31.4mg 이하일 확률을 오른쪽 표준정규분포표를 이용하여 구한 것은?(2016년 9월 모의고사 가형 10번)

Z1 = (29.6-30.2) / 0.6

Z2 = (31.4-30.2) / 0.6

k1 = c()

p1 = c()

for(k in seq(-5,5,by = 0.01)){

p = dnorm(x = k,mean = 0,sd = 1)

k1 = c(k1,k)

p1 = c(p1,p)

}

ggplot(NULL) +

geom\_line(aes(x = k1, y = p1)) +

geom\_area(aes(x = ifelse(k1 > -1 & k1 < 2, k1, 0), y = p1),fill = 'royalblue',

alpha = 0.4) +

theme\_bw() +

scale\_x\_continuous(breaks = seq(-5,5, by = 1)) +

scale\_y\_continuous(expand = c(0,0),limits = c(0,0.45)) +

xlab("") + ylab("")

pnorm(2,mean = 0, sd=1) - pnorm(-1, mean=0, sd=1)

#95% 신뢰 구간은?

ggplot(NULL) +

geom\_line(aes(x = k1, y = p1)) +

geom\_area(aes(x = ifelse(k1 > -1.96 & k1 < 1.96, k1, 0), y = p1),fill = 'royalblue',

alpha = 0.4) +

theme\_bw() +

scale\_x\_continuous(breaks = seq(-5,5, by = 1)) +

scale\_y\_continuous(expand = c(0,0),limits = c(0,0.45)) +

xlab("") + ylab("")

pnorm(1.96,mean = 0, sd=1) - pnorm(-1.96, mean=0, sd=1)

###### PT에 있는 문제 풀어보기

# [2015학년도 수능] 어느 연구소에서 토마토 모종을 심은 지 주가 지났을 때,

# 줄기의 길이를 조사한 결과 토마토 줄기의 길이는 평균이

# 30cm 표준편차가 2cm인 정규분포를 따른다고 한다.

# 이 연구소에서 토마토 모종을 심은 지 주가 지났을 때,

# 토마토 줄기 중 임의로 선택한 줄기의 길이가 27cm이상이고 32cm 이하일 확률을

# 오른쪽 표준정규분포표를 이용하여 구한 것은? [3점]

#2

data(mtcars)

# mtcars 데이터에서 mpg는 평균 23이다 라고 말할 수 있는가?

# 가설검정하여라

##############################

## 추가적으로 알면 좋은 부분들

set.seed(1234) # 랜덤을 고정하는 방법

sample(1:6,10,replace = T)

# 중심극한 정리

z<-c()

for(i in 1:10){

z <- c(z,round(mean(sample(1:6,6,replace = T)),2))

}

ggplot(NULL) +

geom\_bar(aes(x = as.factor(z), fill = as.factor(z))) +

theme\_bw() +

xlab("") + ylab("") +

theme(legend.position = 'none')

z<-c()

for(i in 1:1000){

z <- c(z,round(mean(sample(1:6,6,replace = T)),2))

}

ggplot(NULL) +

geom\_bar(aes(x = as.factor(z), fill = as.factor(z))) +

theme\_bw() +

xlab("") + ylab("") +

theme(legend.position = 'none')

7일차\_문제정답

# [2015학년도 수능] 어느 연구소에서 토마토 모종을 심은 지 주가 지났을 때,

# 줄기의 길이를 조사한 결과 토마토 줄기의 길이는 평균이

# 30cm 표준편차가 2cm인 정규분포를 따른다고 한다.

# 이 연구소에서 토마토 모종을 심은 지 주가 지났을 때,

# 토마토 줄기 중 임의로 선택한 줄기의 길이가 27cm이상이고 32cm 이하일 확률을

# 오른쪽 표준정규분포표를 이용하여 구한 것은? [3점]

Z1 = (27-30) / 2

Z2 = (32-30) / 2

pnorm(Z2,mean = 0, sd=1) - pnorm(Z1, mean=0, sd=1)

ggplot(NULL) +

geom\_line(aes(x = k1, y = p1)) +

geom\_area(aes(x = ifelse(k1 > -1.5 & k1 < 1, k1, 0), y = p1),fill = 'royalblue',

alpha = 0.4) +

theme\_bw() +

scale\_x\_continuous(breaks = seq(-5,5, by = 1)) +

scale\_y\_continuous(expand = c(0,0),limits = c(0,0.45)) +

xlab("") + ylab("")

(27-30.92) / 2

#2

data(mtcars)

# mtcars 데이터에서 mpg는 평균 23이다 라고 말할 수 있는가?

# 가설검정하여라

Z3 = (23-mean(mtcars$mpg)) /sd(mtcars$mpg)

pnorm(Z3,mean = 0, sd=1)

8일차

# 데이터 분석가 \_ james \

# \

# 스크립트 실행(Run a script) \

## : Windows : 'Ctrl + Enter' \

## : MAC : 'Command + Enter'\

#---------------------------------

#1. 상관관계분석(Correlation)

height<-c(164,175,166,185)

weight<-c(62,70,64,86)

cor(height,weight)

round(cor(height,weight),3)

# install.packages("moonBook")

library(moonBook)

data(acs)

#install.packages("psych")

library(psych)

str(acs)

acs2<-acs[,c(1,6:9)]

cor(acs2)

#na가 존재할시 na 제외후 계산

cor(acs2,use="na.or.complete")

#산점도행렬

pairs.panels(acs2)

#install.packages("PerformanceAnalytics");

library(PerformanceAnalytics)

#산점도행렬

chart.Correlation(acs2, histogram=TRUE, pch=19)

#킹콩 data의 추가 -> 데이터 하나의 큰 영향

dat<-data.frame(

a=c(15,20,25,27,31,25,23,23,42,12,34,23,40),

b=c(50,55,52,52,56,54,62,56,70,46,43,50,54)

)

plot(dat$a,dat$b)

abline(lm(dat$b~dat$a))

cor(dat$a,dat$b)

#outlier 추가

dat[14,]<-c(200,230)

plot(dat$a,dat$b)

abline(lm(dat$b~dat$a))

cor(dat$a,dat$b)

#heatmap expression

# install.packages("corrplot")

library(corrplot)

corrplot(cor(acs2,use="na.or.complete"))

corrplot(cor(acs2,use="na.or.complete"),method="square")

corrplot(cor(acs2,use="na.or.complete"),method="ellipse")

corrplot(cor(acs2,use="na.or.complete"),method="number")

corrplot(cor(acs2,use="na.or.complete"),method="shade")

corrplot(cor(acs2,use="na.or.complete"),method="color")

corrplot(cor(acs2,use="na.or.complete"),method="pie")

#cor의 비모수적인 표현들

#1. spearman

#2. kendall's tau

cor(height,weight)

cor(height,weight,method="spearman")

cor(height,weight,method="kendall")

?cor

############### 연습문제 ###############

data(iris)

#1. iris에서 연속형 데이터를 갖고 상관관계를 구하고 Sepal.Length와 가장 상관있는 변수는 무엇인가?

#(2가지 이상의 시각화를 그려보시오)

#####

data(mtcars)

head(mtcars)

#mpg에서 qesc까지의 변수를 갖고 상관관계를 구하시오

##################################

#2. 2 집단에대한 평균비교 t-test

t\_data<-data.frame(

group=c(1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2),

score=c(100,100,80,50,40,90,20,50,50,70,30,40,30,70,30,40,30,60,30,60),

age=c(80,20,30,70,90,20,30,60,50,50,20,30,20,20,25,10,13,12,11,10))

ggplot(t\_data,aes(x=factor(group),y=score,fill=factor(group))) + geom\_boxplot()

#정규성검정

#정규성 검정 - shapiro

shapiro.test(t\_data$score)

#등분산성 검정

t\_data\_1<-t\_data[t\_data$group==1,]

t\_data\_2<-t\_data[t\_data$group==2,]

var.test(t\_data\_1$score,t\_data\_2$score)

#t\_test방법 2가지 존재

#1번 t.test방법

t.test(t\_data\_1$score,t\_data\_2$score,var.equal=T)

#2번 t.test방법

t.test(score~group,data=t\_data,var.equal=T)

#등분산이 아닐경우

var.test(t\_data\_1$age,t\_data\_2$age)

t.test(t\_data\_1$score,t\_data\_2$score,var.equal=F)

#대응 T-test의 수행(전/후비교) - paried=T를 붙여줌

before\_op = c(137,119,117,122,132,110,114,117)

after\_op = c(126,111,117,116,135,110,113,112)

t.test(before\_op,after\_op,paired=T)

mid = c(16, 20, 21, 22, 23, 22, 27, 25, 27, 28)

final = c(19, 20, 24, 24, 25, 25, 26, 26, 28, 32)

t.test(mid,final, paired=TRUE)

################## T검정 연습해보기 ###################

# 1

a = c(175, 168, 168, 190, 156, 181, 182, 175, 174, 179)

b = c(185, 169, 173, 173, 188, 186, 175, 174, 179, 180)

### 다음 데이터를 갖고 T검정을 하시오 (정규성 생략)

# 2

data(mtcars)

# am 변수에 따라 mpg가 차이가 있는지 확인하시오

######################################################

#3.3개이상의 평균비교 시 분산분석 - Anova(Analysis of Variance)

#install.packages("laercio")

anova\_data<-data.frame(

group=c(1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3),

score=c(50.5, 52.1, 51.9, 52.4, 50.6, 51.4, 51.2, 52.2, 51.5, 50.8,47.5, 47.7, 46.6, 47.1, 47.2, 47.8, 45.2, 47.4, 45.0, 47.9,46.0, 47.1, 45.6, 47.1, 47.2, 46.4, 45.9, 47.1, 44.9, 46.2))

ggplot(anova\_data,aes(x=factor(group),y=score,fill=factor(group))) + geom\_boxplot()

tapply(anova\_data$score,anova\_data$group,mean)

tapply(anova\_data$score,anova\_data$group,max)

#등분산성 test

bartlett.test(score~as.factor(group),data=anova\_data)

#oneway.test

oneway.test(score~group,data=anova\_data,var.equal = T)

?aov

a1<-aov(score~group,data=anova\_data)

summary(aov(score~group,data=anova\_data))

#사후분석

library(laercio)

LDuncan(a1, "group")

#group에 해당하는 부분이 문자형 이어야함

TukeyHSD(aov(score~as.character(group),data=anova\_data))

plot(TukeyHSD(aov(score~as.character(group),data=anova\_data)))

######################

#### 등분산이 아닐경우

anova\_data2<-data.frame(

group=c(1,1,1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3,3,3),

score=c(70, 30, 20.3, 85.3, 50.6, 51.4, 51.2, 52.2, 51.5, 50.8,47.5, 47.7, 46.6, 47.1, 47.2, 47.8, 45.2, 47.4, 45.0, 47.9,46.0, 47.1, 45.6, 47.1, 47.2, 46.4, 45.9, 47.1, 44.9, 46.2))

#등분산성 test

bartlett.test(score~as.factor(group),data=anova\_data2)

#oneway.test

oneway.test(score~group,data=anova\_data2,var.equal = F)

a2<-aov(score~group,data=anova\_data2)

summary(aov(score~group,data=anova\_data2))

#사후분석

library(laercio)

LDuncan(a2, "group")

#group에 해당하는 부분이 문자형 이어야함

TukeyHSD(aov(score~as.character(group),data=anova\_data2))

plot(TukeyHSD(aov(score~as.character(group),data=anova\_data2)))

################## F검정 연습해보기 ###################

data(iris)

#1. iris에서 Species마다 Sepal.Width의 차이가 있는지 확인하시오

# 사후 검정과 해석을 적으시오

#2 mtcars데이터에서 gear따라 mpg의 차이가 있는지 확인하시오

# 사후 검정과 해석을 적으시오

#######################################################

#문자형 데이터분석

data(acs)

head(acs)

# 성별과 비만은 연관이 있을까?

table(acs$sex,acs$obesity)

acs %>%

dplyr::count(sex,obesity) %>%

ggplot(aes(x=sex,y=n,fill=obesity)) + geom\_bar(stat="identity",position = "dodge")

chisq.test(acs$sex,acs$obesity,correct = F)

chisq.test(table(acs$sex,acs$obesity))

# correct?

# 비 연속적 이항분포에서 확률이나 비율을 알기 위하여 연속적 분포인

# 카이제곱 분포를 이용할 떄는 연속성을 가지도록 비연속성을 교정해야할 필요하 있을 떄 사용하는 방법

# 보통 2X2 행렬에서 자주 사용함

install.packages("gmodels")

library(gmodels)

CrossTable(acs$sex,acs$obesity,chisq=T,prop.t=F)

CrossTable(table(acs$sex,acs$obesity))

0.089 + 0.175 + 0.045 + 0.088

# 일반횟수

# 카이 제곱 ( 기대치 비율 )

# 행을 기준으로 비율 값 ( 가로로 읽는다. )

# 컬럼을 기준으로 비율 값 ( 세로로 읽는다. )

# 전체를 기준으로 비율 값

# 성별과 비만은 연관이 있을까?

table(acs$sex,acs$smoking)

acs %>%

dplyr::count(sex,smoking) %>%

ggplot(aes(x=sex,y=n,fill=smoking)) + geom\_bar(stat="identity",position = "dodge")

chisq.test(acs$sex,acs$smoking,correct = F)

chisq.test(table(acs$sex,acs$smoking),correct = F)

#자료 생성

dat <- matrix(c(20,24,15,5),ncol=2)

row.names(dat) <- c("흡연","비흡연")

colnames(dat)<- c("정상","비정상")

dat

xtab <- matrix(c(384, 536, 335,951, 869, 438),nrow=2)

dimnames(xtab) <- list(

stone = c("yes", "no"),

age = c("30-39", "40-49", "50-59")

)

colSums(xtab)

prop.trend.test(xtab[1,],colSums(xtab))

mosaicplot(t(xtab),col=c("deepskyblue", "brown2"))

# 나이 비율이 동일하지 않다

################## 카이제곱 연습해보기 ###################

# 1

data("survey")

# survey 데티어에서 Sex변수와 Smoke가 연관이 있는지 검정하여라

# 시각화 포함

# 2

delivery = read.csv('SKT.csv', fileEncoding='UTF-8')

head(delivery)

# 요일별 업종의 차이가 있는지 검정하여라

#######################################################

8일차\_문제정답

############### 연습문제 ###############

data(iris)

#1. iris에서 연속형 데이터를 갖고 상관관계를 구하고 Sepal.Length와 가장 상관있는 변수는 무엇인가?

#(2가지 이상의 시각화를 그려보시오)

pairs.panels(iris[,1:4])

chart.Correlation(iris[,1:4], histogram=TRUE, pch=19)

#####

data(mtcars)

head(mtcars)

#mpg에서 qesc까지의 변수를 갖고 상관관계를 구하시오

pairs.panels(mtcars[,1:7])

chart.Correlation(mtcars[,1:7], histogram=TRUE, pch=19)

##################################

################## T검정 연습해보기 ###################

# 1

a = c(175, 168, 168, 190, 156, 181, 182, 175, 174, 179)

b = c(185, 169, 173, 173, 188, 186, 175, 174, 179, 180)

### 다음 데이터를 갖고 T검정을 하시오 (정규성 생략)

var.test(a,b)

t.test(a,b,var.equal = T)

# 2

data(mtcars)

# am 변수에 따라 mpg가 차이가 있는지 확인하시오

var.test(mtcars[mtcars$am==1,1],mtcars[mtcars$am==0,1])

t.test(mtcars[mtcars$am==1,1],mtcars[mtcars$am==0,1],var.equal = T)

######################################################

################## F검정 연습해보기 ###################

data(iris)

#1. iris에서 Species마다 Sepal.Width의 차이가 있는지 확인하시오

# 사후 검정과 해석을 적으시오

bartlett.test(Sepal.Width~Species, data=iris)

z1 <- aov(Sepal.Width~Species, data=iris)

LDuncan(z1, "Species")

#2 mtcars데이터에서 gear따라 mpg의 차이가 있는지 확인하시오

# 사후 검정과 해석을 적으시오

bartlett.test(mpg~gear, data=mtcars)

z2 <- aov(mpg~gear, data=mtcars)

LDuncan(z2, "gear")

################## 카이제곱 연습해보기 ###################

# 1

data("survey")

# survey 데티어에서 Sex변수와 Smoke가 연관이 있는지 검정하여라

# 시각화 포함

chisq.test(table(survey$Sex,survey$Smoke))

mosaicplot(t(table(survey$Sex,survey$Smoke)),col=c("deepskyblue", "brown2"))

# 2

delivery = read.csv('SKT.csv', fileEncoding='UTF-8')

head(delivery)

# 요일별 업종의 차이가 있는지 검정하여라

chisq.test(table(delivery$요일,delivery$업종))

mosaicplot(t(table(delivery$요일,delivery$업종)),col=c("deepskyblue", "brown2"))

#######################################################

9일차

# 데이터 분석가 \_ james \

# \

# 스크립트 실행(Run a script) \

## : Windows : 'Ctrl + Enter' \

## : MAC : 'Command + Enter'\

#---------------------------------

## 복습하기

# dplyr #day\_4

# tidyr #day\_5

# ggplot #day\_6

## 연습문제 풀어보기

library(openxlsx)

library(lubridate)

subway\_2017 = read.xlsx('subway\_1701\_1709.xlsx')

names(subway\_2017)[6:25] <- paste0('H', substr(names(subway\_2017)[6:25], 1, 2))

# subway\_2017데이터를 활용해서 요약 시각화를 진행하여라

#1. gather을 사용해서 H05부터 H24까지 시간, 고객 변수명으로 데이터 변형하시오

#2. facet\_grid를 사용해서 시간별 고객의 합을 월별로 따라로 그리시오 (1,2,3,4월)

## month를 사용해서 월 변수를 만들어야 한다.

### 프로젝트1

#################################################

######## 직접 분석해보기 (치킨 데이터) ##########

#################################################

# 내가 치킨집을 차릴경우 어떤 데이터를 보고 결정할 것인가?

kk<-read.csv("chicken.csv",fileEncoding = "CP949")

# mac인 경우 fileEncoding = "CP949"를 추가해야 한글이 깨지지 않습니다

# ex) 단순히 전체 지역에서 통화건수를 기반으로 높은 지역을 찾기

# ex) 가장 높은 연령대를 찾고 그 연령대의 전화가 많은 지역을 찾기

head(kk)

summary(kk)

str(kk)

# kk$요일 <- as.factor(kk$요일) factor로 변호

head(kk,50)

# scale\_x\_discrete(limits = c("월","화","수","목","금","토","일"))

# scale\_x\_discrete (x축 순서 조정)

# ex) 요일별 성별 비중을 확인해서 가장 높은 지역의 지역을 찾기

# 시군구 상위 10 개수

# 행수

kk %>%

dplyr::count(시군구) %>%

top\_n(10,n) %>%

arrange(desc(n))

#통화건수의 합

kk %>%

group\_by(시군구) %>%

dplyr::summarise(sum=sum(통화건수)) %>%

top\_n(10,sum) %>%

arrange(desc(sum))

# 성별 연령별 통화가 가장 많은 곳은?

kk %>%

group\_by(성별,연령대) %>%

dplyr::summarise(sum=sum(통화건수)) %>%

data.frame()

kk %>%

group\_by(성별,연령대) %>%

dplyr::summarise(sum=sum(통화건수)) %>%

ggplot(aes(x=연령대,y=sum,fill=성별)) +

geom\_bar(stat="identity",position = "dodge")

# 성별 연령대별 지역 top

# 30대 40대 대상

target <- kk %>%

filter(연령대 %in% c("30대","40대")) %>%

group\_by(성별,연령대,시군구) %>%

dplyr::summarise(sum = sum(통화건수),

num = n()) %>%

ungroup() %>%

group\_by(시군구) %>%

dplyr::summarise(sum1 = sum(sum),

num1 = sum(num)) %>%

arrange(desc(sum1))

# 그래도 만명은 넘는 지역 선별 후 시각화

target %>%

filter(sum1 >= 10000) %>%

ggplot(aes(x=시군구,y=sum1,fill=시군구)) + geom\_bar(stat="identity")

# 요일별 성별 가장 많이 팔리는 시점 확인

kk %>%

group\_by(요일,성별) %>%

dplyr::summarise(sum = sum(통화건수)) %>%

arrange(desc(sum))

kk %>%

group\_by(요일,성별) %>%

dplyr::summarise(sum = sum(통화건수)) %>%

ggplot(aes(x=요일,y=sum,fill=성별)) +

geom\_bar(stat="identity",position = "dodge")

kk %>%

group\_by(요일,성별) %>%

dplyr::summarise(sum = sum(통화건수)) %>%

ggplot(aes(x=요일,y=sum,fill=성별)) +

geom\_bar(stat="identity",position = "dodge")+

scale\_x\_discrete(limits = c("월","화","수","목","금","토","일"))

# 금토일 장사만 해야지!

target2 <- kk %>%

filter(요일 %in% c("금","토","일")) %>%

group\_by(요일,시군구) %>%

dplyr::summarise(sum = sum(통화건수),

num = n()) %>%

ungroup() %>%

group\_by(시군구) %>%

dplyr::summarise(sum1 = sum(sum),

num1 = sum(num)) %>%

arrange(desc(sum1))

target2 %>%

filter(sum1 >= 10000) %>%

ggplot(aes(x=시군구,y=sum1,fill=시군구)) + geom\_bar(stat="identity")

#########################

## 각자 전략을 작성해보기

#########################

# 왜 그렇게 했는지 충분한 근거과 그림이 있어야 됩니다!

# ex) 단순히 전체 지역에서 통화건수를 기반으로 높은 지역을 찾기

# ex) 가장 높은 연령대를 찾고 그 연령대의 전화가 많은 지역을 찾기

# ex) 시간에 따른증가, 분산등을 활용하면 더욱 좋은 방향!

## 가설 검정을 2회이상 진행하기!

# ex) 특정 시간 차이가 있는가?

# ex) 성별, 특정 지역별 차이가 있는가?

# ex) 구매건수별 주문전화 상관관계가 얼마나 큰가?

#######################

#### 코드 작성

#######################

9일차\_문제정답

## 연습문제 풀어보기

library(openxlsx)

library(lubridate)

subway\_2017 = read.xlsx('subway\_1701\_1709.xlsx')

names(subway\_2017)[6:25] <- paste0('H', substr(names(subway\_2017)[6:25], 1, 2))

# subway\_2017데이터를 활용해서 요약 시각화를 진행하여라

#1. gather을 사용해서 H05부터 H24까지 시간, 고객 변수명으로 데이터 변형하시오

#2. facet\_grid를 사용해서 시간별 고객의 합을 월별로 따라로 그리시오 (1,2,3,4월)

## month를 사용해서 월 변수를 만들어야 한다.

subway2 <- gather(subway\_2017,"시간","고객",H05:H24)

subway2 %>%

mutate(month = month(날짜)) %>%

filter(month %in% c(1,2,3,4)) %>%

group\_by(month, 시간) %>%

summarise(SUM = sum(고객)) %>%

arrange(desc(SUM)) %>%

ggplot(aes(x=시간,y=SUM,fill=factor(month))) + geom\_bar(stat="identity") + facet\_grid(~month)

#######################

#### 코드 작성

k11 <- kk %>%

group\_by(시군구) %>%

summarise(주문건수 = n(),

통화합계 = sum(통화건수))

# 건과 총통화 상관관계 확인

library(PerformanceAnalytics)

chart.Correlation(k11[,2:3], histogram=TRUE, pch=19)

# 건당 통화 합계 계산

k11 %>%

mutate(건당통화 = 통화합계 / 주문건수) %>%

arrange(desc(건당통화))

k11 %>%

mutate(건당통화 = 통화합계 / 주문건수) %>%

arrange(desc(건당통화)) %>%

ggplot(aes(x=시군구, y=통화합계, fill=건당통화)) +

geom\_bar(stat = "identity")

# 통화합계가 20000 이상인 건당 통화 확인

k11 %>%

mutate(건당통화 = 통화합계 / 주문건수) %>%

arrange(desc(건당통화)) %>%

filter(통화합계>=20000)

k11 %>%

mutate(건당통화 = 통화합계 / 주문건수) %>%

arrange(desc(건당통화)) %>%

filter(통화합계>=20000) %>%

ggplot(aes(x=시군구, y=통화합계, fill=건당통화)) +

geom\_bar(stat = "identity")

#######################

10일차

# 데이터 분석가 \_ james \

# \

# 스크립트 실행(Run a script) \

## : Windows : 'Ctrl + Enter' \

## : MAC : 'Command + Enter'\

#---------------------------------

### 9일차 리뷰

### 질의응답 및 전체 리뷰

############################################

######## 5-2 직접 분석해보기 (따릉이 데이터) #########

############################################

train <- read.csv("bike.csv")

head(train)

str(train)

# libridate 패키지란?

# 날짜를 쉽게 변환하는 패키지

# ymd : 연월일

# ymd\_hms : 연월일\_시간

ymd("2020-01-01")

ymd("20200101")

ymd\_hms("2020-01-01-03-25-30")

ymd\_hms("20200101032530")

# 1. 날짜형으로 변형

train$datetime <-ymd\_hms(train$datetime)

# 2,3,4 year,month,day 그리고 weekday, hour을 추출후 범주형으로 변환

train$year <- year(train$datetime)

train$month <- month(train$datetime)

train$day <- day(train$datetime)

train$hour <- hour(train$datetime)

train$weekday <- weekdays(train$datetime)

## 데이터 분석 시작

## 데이터 변형해주기

train$season= as.factor(train$season)

train$weather<- as.factor(train$weather)

train$holiday<- as.factor(train$holiday)

train$workingday <- as.factor(train$workingday)

train$weekday <- as.factor(train$weekday)

## 범주형, 연속형을 구분해주는것은 중요!

## 가설 검정을 시도해보자!

## 일하는날과 일안하는날, 시간에 따라서 자전거 수요는 다를까??

## 또한 온도에 따라서도 어떤 영향을 받을까?

## 1. 시각적 접근

## workingday마다 시간에 따른 자전거 수요 시각화하기

## (온도에 대해서 색으로 구분하기!)

## (10분)

######## 코드 작성 #########

############################

## 2. 통계적 접근

### 그렇다면 실제 workingday에 따라서 통계 확인

### workingday는 1과 0으로 되어있는 2범주 데이터이다

### 그러므로 T검정 실시!

# 등분산 검정을 먼저 실시!

# options(scipen = 3) #지수형태가 보기 싫을때

# 귀무가설 : 등분산을 만족한다

# 대립가설 : 등분산을 만족하지 않는다.

var.test(count~workingday,data=train)

# p-value 를 보면 0.05보다 작음을 볼 수 있다.

# 즉, 귀무가설 기각 따라서 등분산을 만족하지 않는다.

# t검정을 진행하면

t.test(count~workingday,data=train,var.equal = F)

#귀무가설 : 그룹마다 차이가 없다.

#대립가설 : 그룹마다 차이가 있다.

# t검정을 해석해보면 등분산을 만족하지 않는 선에서

# 0의평균균은 약 188, 1의평균읜 193이고

# t검정 결과 유의확률은 0.05보다 작으므로 귀무가설 채택

# 즉, 그룹마다 차이가 없다!

##########################

# 위와 같은 과정을 holiday에 따라서 확인하기

######## 코드 작성 #########

############################

# 그럼 이제 실제 데이터 분석 및 변형을 해보자

# 1. 계절에 대해서 실제 그림을 그려보기

# 시간별 대여 수량의 평균을 계절별로 나눠서 그리시오!

train %>%

group\_by(season,hour) %>%

summarise(mean = mean(count)) %>%

ggplot(aes(x=hour,y=mean,group=season,color=season)) + geom\_line()

# 그림에서 이상한 점을 발견할 수 있을 것이다.

# 이상한 점을 확인하고, 수정하여 다시 그리시오!

######## 코드 작성 #########

############################

# 바람 속도에 대해서 전처리

# 실제 바람속도를 히스토그램 그려보면 이상한 점을 발견할 수 있을것이다

# 그 값을 바꿔보시오

train %>%

ggplot(aes(x=windspeed)) + geom\_histogram()

summary(train$windspeed)

# 0을 전처리 하기!

# 중앙값으로 대체하기!

######## 코드 작성 #########

############################

#### F검정 하기

data(iris)

head(iris)

# Species마다 Sepal.Length는 다르다고 말할 수 있을까?

summary(aov(Sepal.Length~Species,data=iris))

# F검정 연습하고 해석하기

# weekday마다 count의 차이가 있는지 검정해보시오

######## 코드 작성 #########

############################

# 상관 관계 확인하기

# 1.temp와 atemp는 얼마나 상관이 있는가?

cor(train$temp,train$atemp)

# 2.count와 가장 연관이 있는 변수는 어떤 변수인가? (registered , casual 제외)

# 3.가장 영향력 있는 변수와의 그림을 그려보기

######## 전략을 세워보기!

## 시간 월별로는 자전거 대여가 어떻게 다르지??

## 요일별로 시간별로 어떻게 다르지??

## 각자 자전거 배치에 대한 전략을 세우고 근거를 만들어 보세요

## ex) 온도가 30 넘고 휴일이면 평균 몇대 이상씩은 두어야된다

## ex) 출퇴근 시간에는 어느정도 고정수요가 있기 떄문에 적정 수량을 배치한다.

##

train$month <- as.factor(train$month)

train$hour <- as.factor(train$hour)

train %>%

group\_by(weekday,hour) %>%

dplyr::summarise(mean = mean(count)) %>%

ggplot(aes(x=hour,y=mean,color=weekday,group=weekday)) + geom\_line()

train$weekday <- factor(train$weekday,levels = c("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday"))

train %>%

group\_by(weekday,hour) %>%

dplyr::summarise(mean = mean(count)) %>%

ggplot(aes(x=hour,y=mean,group=weekday,color=weekday)) + geom\_line()

train %>%

group\_by(month,hour) %>%

dplyr::summarise(mean = mean(count)) %>%

ggplot(aes(x=hour,y=mean,color=factor(month),group=month)) + geom\_line()

train %>%

ggplot(aes(x=hour, y=weekday, fill=count)) +

geom\_tile() +

scale\_fill\_distiller(palette='Blues')

train %>%

ggplot(aes(x=hour, y=month, fill=count)) +

geom\_tile() +

scale\_fill\_distiller(palette='Blues',direction = 1)

#######################################################

#######################################################

## 전체 리뷰 해보기!

########### 마지막 꿀팁

install.packages("DataExplorer")

library(DataExplorer)

data(iris)

head(iris)

config <- list(

"introduce" = list(),

"plot\_str" = list(

"type" = "diagonal",

"fontSize" = 35,

"width" = 1000,

"margin" = list("left" = 350, "right" = 250)

),

"plot\_missing" = list(),

"plot\_histogram" = list(),

"plot\_qq" = list(sampled\_rows = 1000L),

"plot\_bar" = list(),

"plot\_correlation" = list("cor\_args" = list("use" = "pairwise.complete.obs")),

# "plot\_prcomp" = list(),

"plot\_boxplot" = list(),

"plot\_scatterplot" = list(sampled\_rows = 1000L)

)

create\_report(iris, config = config)

## 치킨데이터 적용시켜보기

create\_report(train,config = config)

10일차\_문제정답

## 1. 시각적 접근

## workingday마다 시간에 따른 자전거 수요 시각화하기

## (온도에 대해서 색으로 구분하기!)

## (10분)

#평일

train %>%

filter(workingday == 1) %>%

ggplot(aes(hour, count, color = temp)) +

geom\_point() +

scale\_color\_gradient(low = "#88d8b0", high = "#ff6f69") + ggtitle("Bike Count on Working Days")

# position\_jitter 은 퍼뜨리는 역할

train %>%

filter(workingday == 1) %>%

ggplot(aes(hour, count, color = temp)) +

geom\_point(position = position\_jitter()) +

scale\_color\_gradient(low = "#88d8b0", high = "#ff6f69") + ggtitle("Bike Count on Working Days")

#휴일

train %>%

filter(workingday == 0) %>%

ggplot(aes(hour, count)) +

geom\_point(position = position\_jitter(), aes(color = temp)) +

scale\_color\_gradient(low = "#88d8b0", high = "#ff6f69") + ggtitle("Bike Count on Non-Working Days")

##########################

# 위와 같은 과정을 holiday에 따라서 확인하기

# 1 시각화

#평일

train %>%

filter(holiday == 0) %>%

ggplot(aes(hour, count)) +

geom\_point(position = position\_jitter(), aes(color = temp)) +

scale\_color\_gradient(low = "#88d8b0", high = "#ff6f69") + ggtitle("Bike Count on Working Days")

#휴일

train %>%

filter(holiday == 1) %>%

ggplot(aes(hour, count)) +

geom\_point(position = position\_jitter(w = 1, h = 0), aes(color = temp)) +

scale\_color\_gradient(low = "#88d8b0", high = "#ff6f69") + ggtitle("Bike Count on Non-Working Days")

# 2 통계 검정

var.test(count~holiday,data=train)

t.test(count~holiday,data=train,var.equal = T)

###########################

# 그럼 이제 실제 데이터 분석 및 변형을 해보자

# 1. 계절에 대해서 실제 그림을 그려보기

# 시간별 대여 수량의 평균을 계절별로 나눠서 그리시오!

train %>%

group\_by(season,hour) %>%

summarise(mean = mean(count)) %>%

ggplot(aes(x=hour,y=mean,group=season,color=season)) + geom\_line()

# 그림에서 이상한 점을 발견할 수 있을 것이다.

# 이상한 점을 확인하고, 수정하여 다시 그리시오!

# train %>%

# dplyr::select(season,month) %>%

# unique()

#

#

# train %>%

# mutate(season\_new = ifelse(month %in% c(3,4,5),1,

# ifelse(month %in% c(6,7,8),2,

# ifelse(month %in% c(9,10,11),3,4)))) %>%

# group\_by(season\_new,hour) %>%

# summarise(mean = mean(count)) %>%

# ggplot(aes(x=hour,y=mean,group=factor(season\_new),color=factor(season\_new))) + geom\_line()

# 바람 속도에 대해서 전처리

# 실제 바람속도를 히스토그램 그려보면 이상한 점을 발견할 수 있을것이다

# 그 값을 바꿔보시오

train %>%

ggplot(aes(x=windspeed)) + geom\_histogram()

summary(train$windspeed)

# 0을 전처리 하기!

# 중앙값으로 대체하기!

# train[train$windspeed==0,"windspeed"] <- median(train$windspeed)

#######################

# F검정 연습하고 해석하기

# weekday마다 count의 차이가 있는지 검정해보시오

######## 코드 작성 #########

# summary(aov(count~season,data=train))

############################

# 상관 관계 확인하기

# 1.temp와 atemp는 얼마나 상관이 있는가?

cor(train$temp,train$atemp)

# 2.count와 가장 연관이 있는 변수는 어떤 변수인가? (registered , casual 제외)

# chart.Correlation(train[,6:12], histogram=TRUE, pch=19)

# 3.가장 영향력 있는 변수와의 그림을 그려보기

# train %>%

# ggplot(aes(x=count,y=temp,color=count)) + geom\_point()