# EMPNO : 사원번호

# ENAME : 사원명

# JOB : 직책

# MGR : 사수번호

# HIREDATE : 입사날짜

# SAL : 급여

# COMM : 보너스(커미션)

# DEPTNO : 부서번호

emp = data.frame(

empNO = c(7369, 7499, 7521, 7566, 7698, 7782, 7788, 7839, 7844, 7900),

ENAME = c("SMITH", "ALLEN", "WARD", "JONES", "BLAKE", "CLARK", "SCOTT", "KING", "TURNER", "ADAMS"),

JOB = c("CLERK", "SALESMAN", "SALESMAN", "MANAGER", "MANAGER", "MANAGER", "ANALYST", "PRESIDENT", "SALESMAN", "CLERK"),

MGR = c(7902, 7698, 7698, 7839, 7839, 7839, 7566, NA, 7698, 7788),

HIREDATE = as.Date(c("1980-12-17", "1981-02-20", "1981-02-22",

"1981-04-02", "1981-05-01", "1981-06-09",

"1982-12-09", "1981-11-17", "1981-09-08",

"1983-01-12")),

SAL = c(800, 1600, 1250, 2975, 2850, 2450, 3000, 5000, 1500, 1100),

COMM = c(NA, 300, 500, NA, NA, NA, NA, NA, NA, NA),

DEPTNO = c(20, 30, 30, 20, 30, 10, 20, 10, 30, 20)

# 벡터 복습

# 1. 벡터는 비슷한 데이터를 한 줄로 모아놓은 상자

# 2. 벡터를 생성할 때는 c() Combind 라는 문법을 사용

# 3. 벡터에서 값을 꺼낼 때는 []를 사용합니다.

colors = c('red','blue','green') #벡터 생성

print(colors[1]) # ctrl + shitf + s

print(colors[c(2, 3)]) # 벡터에서 여러 값 꺼낼 때

# \*\*\*\* 벡터와 조건문 활용

# ifelse : 만약 ~ 라면

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

result = ifelse(x %% 2 == 0, '짝수', '홀수')

print(result)

# 데이터프레임

# 가로(행)와 세로(열)가 있는 테이블

# 벡터생성

id = c(1,2,3)

name = c('김길동','홍길동','박길동')

age = c(20,30,40)

# 데이터프레임 생성

df = data.frame(id,name,age) # 벡터를 묶은게 데이터프레임이다.

View(df) #테이블 형태로 출력

# EMPNO : 사원번호

# ENAME : 사원명

# JOB : 직책

# MGR : 사수번호

# HIREDATE : 입사날짜

# SAL : 급여

# COMM : 보너스(커미션)

# DEPTNO : 부서번호

emp = data.frame(

empNO = c(7369, 7499, 7521, 7566, 7698, 7782, 7788, 7839, 7844, 7900),

ENAME = c("SMITH", "ALLEN", "WARD", "JONES", "BLAKE", "CLARK", "SCOTT", "KING", "TURNER", "ADAMS"),

JOB = c("CLERK", "SALESMAN", "SALESMAN", "MANAGER", "MANAGER", "MANAGER", "ANALYST", "PRESIDENT", "SALESMAN", "CLERK"),

MGR = c(7902, 7698, 7698, 7839, 7839, 7839, 7566, NA, 7698, 7788),

HIREDATE = as.Date(c("1980-12-17", "1981-02-20", "1981-02-22",

"1981-04-02", "1981-05-01", "1981-06-09",

"1982-12-09", "1981-11-17", "1981-09-08",

"1983-01-12")),

SAL = c(800, 1600, 1250, 2975, 2850, 2450, 3000, 5000, 1500, 1100),

COMM = c(NA, 300, 500, NA, NA, NA, NA, NA, NA, NA),

DEPTNO = c(20, 30, 30, 20, 30, 10, 20, 10, 30, 20)

View(emp)

# 벡터 복습

# 1. 벡터는 비슷한 데이터를 한 줄로 모아놓은 상자

# 2. 벡터를 생성할 때는 c() Combind 라는 문법을 사용

# 3. 벡터에서 값을 꺼낼 때는 []를 사용합니다.

colors = c('red','blue','green') #벡터 생성

print(colors[1]) # ctrl + shitf + s

print(colors[c(2, 3)]) # 벡터에서 여러 값 꺼낼 때

# \*\*\*\* 벡터와 조건문 활용

# ifelse : 만약 ~ 라면

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

result = ifelse(x %% 2 == 0, '짝수', '홀수')

print(result)

# 데이터프레임

# 가로(행)와 세로(열)가 있는 테이블

# 벡터생성

id = c(1,2,3)

name = c('김길동','홍길동','박길동')

age = c(20,30,40)

# 데이터프레임 생성

df = data.frame(id,name,age) # 벡터를 묶은게 데이터프레임이다.

#View(df) #테이블 형태로 출력

# EMPNO : 사원번호

# ENAME : 사원명

# JOB : 직책

# MGR : 사수번호

# HIREDATE : 입사날짜

# SAL : 급여

# COMM : 보너스(커미션)

# DEPTNO : 부서번호

emp = data.frame(

empNO = c(7369, 7499, 7521, 7566, 7698, 7782, 7788, 7839, 7844, 7900),

ENAME = c("SMITH", "ALLEN", "WARD", "JONES", "BLAKE", "CLARK", "SCOTT", "KING", "TURNER", "ADAMS"),

JOB = c("CLERK", "SALESMAN", "SALESMAN", "MANAGER", "MANAGER", "MANAGER", "ANALYST", "PRESIDENT", "SALESMAN", "CLERK"),

MGR = c(7902, 7698, 7698, 7839, 7839, 7839, 7566, NA, 7698, 7788),

HIREDATE = as.Date(c("1980-12-17", "1981-02-20", "1981-02-22",

"1981-04-02", "1981-05-01", "1981-06-09",

"1982-12-09", "1981-11-17", "1981-09-08",

"1983-01-12")),

SAL = c(800, 1600, 1250, 2975, 2850, 2450, 3000, 5000, 1500, 1100),

COMM = c(NA, 300, 500, NA, NA, NA, NA, NA, NA, NA),

DEPTNO = c(20, 30, 30, 20, 30, 10, 20, 10, 30, 20)

)

# View(emp)

# 데이터프레임 조회

# \*\*\*데이터 프레임 타입 확인

str(emp) # structure(구조)

# 1행 부터 6행까지 출력

print(head(emp))

# 1행 부터 2행까지 출력

print(head(emp,2))

# 아래서 부터 6행까지 출력

print(tail(emp))

# 아래서 부터 2행까지 출력

print(tail(emp,2))

# 전체 컬럼조회

print(colnames(emp))

# 행과 열 개수 조회

# dim : dimension(차원)

print(dim(emp)) # 10행 8 열

# 데이터프레임 특정 열 조회

# 사원 이름만 조회

cat('사원 이름 : ', emp$ENAME, '\n')

# 부서 번호만 조회

cat('부서 번호 : ', emp$DEPTNO, '\n')

# 새로운 열 생성

emp$bonus = 100

# View(emp) # 확인

# 데이터 프레임 -> 엑셀로 전환

# file = 'emp.csv' : 저장할 파일 이름

# row.names = FALSE : 행 번호가 파일에 저장되지 않습니다.

# write.csv(emp, file = 'emp.csv', row.names = FALSE)

# 집계함수

# 엑셀 총합, 평균,최댓값, 최솟값 ...

# R도 엑셀처럼 통계내는 도구가 존재

# 벡터 생성

x = c(10, 20, 30, 40, 50)

# 최솟값

최솟값 = min(x)

최댓값 = max(x)

총합 = sum(x)

평균 = mean(x)

# sd : standard deviation

표준편차 = sd(x)

# 데이터프레임에서 특정 열(벡터 조회)

# cat을 이용해서 emp의 급여만 조회

cat('사원 급여 : ', emp$SAL , '\n')

# 사원 급여 평균 구하기

cat('사원 급여 평균 : ', mean(emp$SAL), '\n')

# 사원 급여 총합

cat('사원 급여 총합 : ', sum(emp$SAL), '\n')

# 사원 급여 최댓값

cat('사원 급여 최댓값 : ', max(emp$SAL), '\n')

# 사원 급여 최솟값

cat('사원 급여 최솟값 : ', min(emp$SAL), '\n')

# COMM : 커미션의 총합 구하기

# na.rm = TRUE : 열값 중 NA가 있으면 통계 x

# rm : remove(제거)

print(sum(emp$COMM, na.rm = TRUE)) #결과 x

# 제어문(조건문)을 활용한 열 생성

# 사원급여가 3000 이상이면 High 나머지는 Low 입니다.

emp$Grade = ifelse(emp$SAL >= 3000, 'High', 'Low')

# View(emp) # 결과 확인

# 기존 열 수정

emp$bonus = emp$bonus \* 2

# View(emp)

# 사원 근속일(오늘날짜 - 입사날짜) 컬럼 추가하기

# 오늘 날짜 조회

print(Sys.Date())

# difftime : 시간 차이 계산

# 단, 해당 열(벡터)가 데이터타입이 날짜형(as.Date)이여야 합니다.

cat('사원 근속일 : ', difftime(Sys.Date(), emp$HIREDATE), '\n')

# 데이터프레임 행 조건 필터링

# 문법 : 데이터프레임[행조건, 열조건]

# 문제. 사원급여가 2000이상인 사원전체 조회

결과 = emp[emp$SAL >= 2000, c('ENAME', 'SAL')]

print(결과)

# 문제. 직책(JOB)이 SALESMAN인 사원의 이름과 직책, 입사날짜 조회

# == (비교연산자)

결과2 = emp[emp$JOB == 'SALESMAN', c('ENAME','JOB','HIREDATE')]

print(결과2)

# 문제. 급여가 1500 이상 3000이하인 직원 조회

# &(and연산자) 두 조건 만족

결과3 = emp[emp$SAL >= 1500 & emp$SAL <= 3000,]

print(결과3)

# 문제. 커미션(COMM)을 받은 직원조회(NA 제외)

# \*\*\*is.na() : na 니?

x = c(1,2,NA,4)

print(is.na(x)) # FALSE FALSE TRUE FALSE

# ! : 부정,반대

print(!is.na(x)) # TRUE TRUE FALSE TRUE

# COMM이 NA가 아닌 사원조회

print(emp[!is.na(emp$COMM), c('ENAME','COMM')] )

# 경로 확인

print(getwd())

print(list.files()) # 해당 경로에 있는 파일 조회

# csv파일 불러오기

emp = read.csv('emp.csv')

# View(emp) #데이터 확인

# 데이터 확인

# 문제 1: 행과 열의 개수 파악

print(dim(emp)) # dimension : 차원

# 문제 2: 전체 컬럼만 조회

print(colnames(emp)) # col : 컬럼 + names : 이름(들)

# 문제 3: 데이터 상위 1~2행 출력하기

print(head(emp,2))

# 문제 4: 데이터 마지막 3개행을 출력하기

print(tail(emp,3))

# 문제 5: 데이터 타입 확인\*\*\*

str(emp) # structure : 구조

## dplyr (디플리알) -> 데이터 가공(전처리)

# 데이터 전처리가 실무에서 80~90% 하는일

# install.packages("dplyr") # 1. 설치 필요

library(dplyr) #설치한 프로그램 가져오기(임포트 import)

# 급여(SAL)가 3000 이상인 직원들의 이름(ENAME)과 직업(JOB)을 출력하세요.

결과 = emp %>% filter(SAL >= 3000) %>% select(ENAME, JOB)

print(결과)

# 직업(JOB)별 평균 급여(SAL)를 계산하고 출력하세요.

# R은 group\_by만 단독적으로 사용 시 의미없는 결과가 나온다.

# 그룹별 평균, 총합, 최댓, 최솟, 중앙, 표준편차.. summarise(요약하다)

# mean : 평균, n : 행의 수, sum : 총합

결과 = emp %>% group\_by(JOB) %>%

summarise(AVG\_SAL = mean(SAL), EMP\_COUNT = n(), SUM\_SAL = sum(SAL))

print(결과)

# 급여가 2000 이상인 직원들만 필터링한 후, 부서 번호(DEPTNO)별 직원 수를 계산하세요.

결과 = emp %>%

filter(SAL >= 2000) %>%

group\_by(DEPTNO) %>%

summarize(EMP\_COUNT = n())

print(결과)

# dept 데이터 불러오기, str로 구조확인

dept = read.csv('dept.csv')

str(dept)

# View(dept)

# 디플리알 병합(JOIN)

# 두 데이터 프레임을 특정 컬럼을 기준으로 병합합니다.

조인결과 = emp %>% inner\_join(dept, by = "DEPTNO" )

#View(조인결과) # dept에 있는 컬럼까지 확인가능

# 근무지가 "DALLAS"인 직원들의 이름 출력하기

조인결과= emp %>% inner\_join(dept, by = "DEPTNO") %>% filter(LOC == 'DALLAS') %>%

select(ENAME, JOB)

print(조인결과)

# slice : 자르다(DBMS = limit)

결과 = emp %>% slice(2, 4)

print(결과) # 2번째 행과 4번째 행 추출

# 첫 번째 ~ 세 번째 직원 행 추출

결과 = emp %>% slice(1:3)

print(결과)

# slice 맨 마지막에 작성(대부분 경우)

# 문제 풀기 2,4,8,10번 풀기

# 문제 2: "RESEARCH" 부서에 근무하는 직원들의 이름(ENAME)과 급여(SAL)를 출력하세요.

조인결과 = emp %>% inner\_join(dept, by = "DEPTNO") %>% filter(LOC == 'RESEARCH') %>%

select(ENAME, SAL)

print(조인결과)

# 문제 4: 각 부서(DNAME)별 직원 수를 계산하고 출력하세요.

조인결과 = emp %>% inner\_join(dept, by = "DEPTNO") %>% group\_by(DNAME) %>%

summarise(EMP\_COUNT = n()) %>% slice(1)

print(조인결과)

# 문제 8: "SALES" 부서에서 근무하는 직원들의 이름(ENAME), 급여(SAL), 커미션(COMM)을 출력하세요.(skip)

# 문제 10: 직업(JOB)이 "MANAGER"인 직원들의 이름(ENAME), 부서명(DNAME), 급여(SAL)을 출력하세요.

조인결과 = emp %>% inner\_join(dept, by = "DEPTNO") %>% filter(JOB == 'MANAGER') %>%

select(ENAME, DNAME, SAL)

# 문제 4

# arrange : 정렬, desc: 내림차순

조인결과 = emp %>% inner\_join(dept, by = "DEPTNO") %>% group\_by(DNAME) %>%

summarise(EMP\_COUNT = n()) %>% arrange(desc(EMP\_COUNT)) %>% slice(1:2)

print(조인결과)