```
#######################################################################
### Computational Policy and Project Analysis - Lecture 05 ########
### Subject: Data Pre-processing I                                ########
### Developed by. KKIM                                            ########
#######################################################################

load(file="R file/R file_LEC05/ds_salaries_ed.RData")
load(file="R file/R file_LEC05/treatments.RData")

################### Data Exploration ############################

data(mtcars)
head(mtcars)
tail(mtcars)
head(mtcars,2)
tail(mtcars,2)

str(mtcars)

### Finding values
x <- c(5,1,2,6,3,17,8,9,12)
myindex <- which( x > 10 )
myindex
x[myindex]
x[x>10]

x <- c(5,1,2,6,3,17,8,9,12)
x
which.max(x)
which.min(x)
x[which.max(x)]
x[which.min(x)]

x==max(x)
x==min(x)
x[x==max(x)]
x[x==min(x)]

load(file="R file/R file_LEC05/ds_salaries_ed.RData")

### select
# column name
ds_sal[,c("job_title","salary","salary_currency")]
head(ds_sal[,c("job_title","salary","salary_currency")])

library(dplyr)
ds_sal %>% #names
  select(job_title, salary, salary_currency) %>%
  head

ds_sal %>% select(job_title:salary_currency) %>%
  head

# index
head(ds_sal[,c(5,7)])

ds_sal %>% select(5,7) %>%
  head

ds_sal %>% select(5:7) %>%
  head

# select with starts_with
ds_sal %>% #names
  select(salary) %>%
  head

ds_sal %>% select(starts_with('salary')) %>%
  head
```

```r
ds_sal %>% select(!starts_with('salary')) %>%
  head

# Select columns that starts with  “salary”  or  “company” ?
ds_sal %>%
  select(starts_with('salary') | starts_with('company')) %>%
  names

# select with if
ds_sal %>% head(1)

ds_sal %>%
  select_if(is.numeric) %>%
  head(2)

ds_sal %>%
  select_if(is.character) %>%
  head(2)

### Filter
head(ds_sal[ds_sal$job_title=="Data Scientist",], 2)

ds_sal %>% head(1)
ds_sal %>%
  select(job_title) %>%
  unique
ds_sal$job_title %>% unique

ds_sal %>%
  filter(job_title=="Data Scientist") %>%
  head(2)

ds_sal %>%
  filter(job_title=="Data Scientist") %>%
  select(job_title) %>% unique

head(ds_sal[ds_sal$salary>=mean(ds_sal$salary),], 2)

mean(ds_sal$salary)
ds_sal %>%
  filter(salary>=mean(salary)) %>%
  head(2)

### arrange
head(ds_sal[order(ds_sal$salary),],2)

ds_sal %>%
  arrange(salary) %>%
  head(2)

head(ds_sal[order(ds_sal$salary, decreasing=TRUE),],2)

ds_sal %>%
  arrange(desc(salary)) %>%
  head(2)

### QUIZ
# 1) What is the highest salary among those working
# for large corporations?
ds_sal %>%
  filter(company_size=='L') %>%
  arrange(desc(salary)) %>%
  head(3)

# 2) What is the average salary of people who are working
# fully remotely?
# Answer this questions with two versions: conventional approach & chain operator approach
mean(ds_sal[ds_sal$remote_ratio==100,]$salary)
```

```r
ds_sal %>%
  filter(remote_ratio==100) %>%
  select(salary) %>%
  pull() %>%
  mean()

# 3) Where do the top 10 highest-earning individuals live?
ds_sal %>%
  arrange(desc(salary)) %>%
  select(employee_residence) %>%
  head(10)

# 4) Is it possible to be a Data Scientist who works
# full time (FT),
# fully remotely for the large company?
# If possible, how many cases are there?
ds_sal %>%
  filter(job_title=="Data Scientist" &
           employment_type=='FT' &
           remote_ratio==100 &
           company_size=='L') # %>% nrow

### mutate
head(ds_sal, 2)

ds_sal %>% mutate(experience=2024-work_year) %>%
  select(work_year,experience,salary) %>%
  head

ds_sal %>%
  mutate(salary.d = ifelse(salary_in_usd > mean(salary_in_usd),
                           "High", "Low")) %>%
  select(work_year,salary,salary.d) %>% head

library(magrittr)
ds_sal %<>% mutate(experience=2024-work_year)

# mutate_at
ds_sal %>% select(ID, salary,experience) %>%
  mutate_at(vars(salary,experience), log) %>%
  head

ds_sal %>% select(ID, salary,experience) %>%
  mutate_at(vars(salary,experience), max) %>% head

# mutate_all
ds_sal %>%
  mutate_all(is.na) %>% head(2)

norm.fun <-
  function(x){
    (x - mean(x, na.rm = TRUE)) / sd(x, na.rm = TRUE)}
ds_sal %>% select_if(is.numeric) %>%
  mutate_all(norm.fun) %>% head

ds_sal.1 <- ds_sal %>%
  rbind(work_year=NA, salary=NA,
        salary_in_usd=NA,
        remote_ratio=NA)
ds_sal.1 %>% select_if(is.numeric) %>%
  mutate_all(norm.fun) %>% head

### rename
names(ds_sal)
ds_sal %<>% rename(work_experience=experience)
names(ds_sal)

ds_sal %>% rename_with(toupper) %>% names
```

```
ds_sal %>% rename_with(toupper, starts_with("salary")) %>% names

### QUIZ
# (1) Create International variable, which returns International if
# employee_residence and company_location are the different and Domestic otherwise
ds_sal %<>% mutate(International =
                     ifelse(employee_residence != company_location,
                            "International","Domestic"))
ds_sal %>%
  select(employee_residence,company_location,International) %>%
  head

# (2) Create job.d variable, which returns DS for Data Scientist, DA for
# Data Analyst and others for Others
ds_sal %<>%
  mutate(job.d = case_when(job_title=="Data Scientist" ~ "DS",
                           job_title=="Data Analyst" ~ "DA",
                           TRUE ~ "Others"))
ds_sal %>%
  select(work_year,job_title,job.d) %>%
  head

# (3) Convert job.d to job_dummy
ds_sal %<>%
  rename(job_dummy = job.d)

################### Column Manipulation ############################

load(file="R file/R file_LEC05/treatments.RData")

# View the treatments data
treatments

### separate
# Separate year_mo into two columns
library(tidyr)
separate(treatments,
         col=year_mo,
         into=c("year", "month"),
         sep='-')

# load(file="R file/R file_LEC06/treatments.RData")
# treatments$year_mo <-
#   gsub("_","-",treatments$year_mo)
# separate(treatments,
#          col=year_mo,
#          into=c("year", "month"),
#          sep='-')

### unite
treatments2 <-
  separate(treatments,
           col=year_mo,
           into=c("year", "month"),
           sep='-')
head(treatments2)

unite(treatments2,
      col=ym,
      c(year,month),
      sep=':')

### Quiz
bmi_cc <-
  read.csv('R file/R file_LEC05/bmi_cc.csv',
           header = TRUE)
head(bmi_cc)

# Apply separate() to bmi_cc
```

```
bmi_cc_clean <-
  separate(bmi_cc,
           col=Country_ISO,
           into=c("Country", "ISO"),
           sep="/"
  )
head(bmi_cc_clean)

# Apply unite() to bmi_cc_clean
bmi_cc2 <- unite(bmi_cc_clean,
                 col=ID,
                 Country,ISO,
                 # c(Country,ISO),
                 sep="-" )
bmi_cc2$year <-
  gsub("Y","",bmi_cc2$year)
head(bmi_cc2)
```