**#IMPORTING DATA FROM DATABASE**

**-install RMySql :**

**\*install.packages(“RMySQL”)**

**\*library(DBI)**

**-Connect to the database :**

**\*con <- dbConnect(RMySQL :: MySQL(), -> for construct SQL Driver**

**dbname = “company”,**

**host = “courses.csrrinzqubik.us-east-1.rds.amazonaws.com”,**

**port = 3306,**

**user =”student”,**

**password=”datacamp”)**

**-Melihat list table pada database :**

**dbListTables(con)**

**-Membaca table pada database :**

**dbReadTable(con, “employee”) -> membaca table employee**

**\*result akan berupa data.frame**

**-Memutus sambungan ke database :**

**dbDisconnect(con)**

**-Import All Tables :**

**#Get Table names**

**\*table\_names <- dbListTables(con)**

**#Import All Tables**

**\*tables <- lapply (table\_names, dbReadTable, conn = con)**

**\*tables**

**#IMPORTING DATA FROM DATABASE PART 2**

**-subset data from database table :**

**\*dbGetQuery(con, “SELECT name FROM employees**

**WHERE started\_at >”2019-09-01””)**

**-Jika ingin import data dari SQL tapi melalui beberapa tahap**

**#Use dbsendquery**

**\*res <- dbSendQuery(con, “SQL\_query”)**

**#Use dbfetch**

**\*dbfetch(res, n=3) -> mengambil 3 kolom saja**

**\*dbfetch(res, n=2) -> mengambil 2 kolom. (untuk penarikan data terakhir, nilai n tidak perlu dispesifikkan juga gpp**

**# Clear res**

**\*dbClearResult(res)**

**#IMPORTING DATA FROM THE WEB PART 1**

**-read http :**

# Load the readr package

library(readr)

# Import the csv file: pools

url\_csv <- "http://s3.amazonaws.com/assets.datacamp.com/production/course\_1478/datasets/swimming\_pools.csv"

pools <- read\_csv(url\_csv)

# Import the txt file: potatoes

url\_delim <- "http://s3.amazonaws.com/assets.datacamp.com/production/course\_1478/datasets/potatoes.txt"

potatoes <- read\_tsv(url\_delim)

# Print pools and potatoes

pools

potatoes

**-read https : more secure than http (yg pake readr masih blm bisa. Pake read.csv bisa)**

# https URL to the swimming\_pools csv file.

url\_csv <- "https://s3.amazonaws.com/assets.datacamp.com/production/course\_1478/datasets/swimming\_pools.csv"

# Import the file using read.csv(): pools1

pools1 <- read.csv(url\_csv, header=TRUE)

# Load the readr package

library(readr)

# Import the file using read\_csv(): pools2

pools2 <- read\_csv(url\_csv)

# Print the structure of pools1 and pools2

str(pools1)

str(pools2)

**-read excel from web with gdata and readxl : file bisa didownload hanya tidak bisa dibaca krn file corrupt**

# Load the readxl and gdata package

library(readxl)

library(gdata)

# Specification of url: url\_xls

url\_xls <- "http://s3.amazonaws.com/assets.datacamp.com/production/course\_1478/datasets/latitude.xls"

# Import the .xls file with gdata: excel\_gdata

excel\_gdata <- read.xls(url\_xls)

# Download file behind URL, name it local\_latitude.xls

local\_latitude <- file.path("~", "local\_latitude.xls")

download.file(url\_xls, local\_latitude)

# Import the local .xls file with readxl: excel\_readxl

excel\_readxl <- read\_excel(local\_latitude)

**-Download RData :**

# https URL to the wine RData file.

url\_rdata <- "https://s3.amazonaws.com/assets.datacamp.com/production/course\_1478/datasets/wine.RData"

# Download the wine file to your working directory

download.file(url\_rdata, destfile = "wine\_local.RData")

# Load the wine data into your workspace using load()

load("wine\_local.RData")

# Print out the summary of the wine data

summary(wine)

**-httr : membaca file langsung dr web. Bukan dalam bentuk file seperti csv, txt, atau xls**

# Load the httr package

library(httr)

# Get the url, save response to resp

url <- "http://www.example.com/"

resp <- GET(url)

# Print resp

print(resp)

# Get the raw content of resp: raw\_content

raw\_content <- content(resp, as="raw")

# Print the head of raw\_content

head(raw\_content)

**-get json file with httr :**

# httr is already loaded

# Get the url

url <- "http://www.omdbapi.com/?apikey=72bc447a&t=Annie+Hall&y=&plot=short&r=json"

resp <- GET(url)

# Print resp

print(resp)

# Print content of resp as text

content(resp, as="text") -> bentuknya memanjang kayak text

# Print content of resp

content(resp) -> bentuk vector

**-Convert data json dalam text menjadi data di R :**

**# Load the jsonlite package**

**library(jsonlite)**

**# wine\_json is a JSON**

**wine\_json <- '{"name":"Chateau Migraine", "year":1997, "alcohol\_pct":12.4, "color":"red", "awarded":false}'**

**# Convert wine\_json into a list: wine**

**wine <- fromJSON(wine\_json)**

**# Print structure of wine**

**str(wine)**

**-convert json data to R data from web directly :**

# jsonlite is preloaded

# Definition of quandl\_url

quandl\_url <- "https://www.quandl.com/api/v3/datasets/WIKI/FB/data.json?auth\_token=i83asDsiWUUyfoypkgMz"

# Import Quandl data: quandl\_data

quandl\_data <- fromJSON(quandl\_url)

# Print structure of quandl\_data

str(quandl\_data)

**-converting JSon array to R Vector :**

# jsonlite is already loaded

# Challenge 1

json1 <- '[1, 2, 3, 4, 5, 6]'

fromJSON(json1)

str(json1) -> tipe data akan berupa character

# Challenge 2

json2 <- '{"a": [1, 2, 3], "b" : [4, 5, 6]}'

fromJSON(json2)

str(json2) -> tipe data akan berupa character

**-JSON Challenge :**

# jsonlite is already loaded

# Challenge 1

json1 <- '[[1, 2], [3, 4]]'

fromJSON(json1)

# Challenge 2

json2 <- '[{"a": 1, "b": 2}, {"a": 3, "b": 4}, {"a":5, "b":6}]'

fromJSON(json2)

**-Converting data to JSON file :**

# jsonlite is already loaded

# URL pointing to the .csv file

url\_csv <- "http://s3.amazonaws.com/assets.datacamp.com/production/course\_1478/datasets/water.csv"

# Import the .csv file located at url\_csv

water <- read.csv(url\_csv, stringsAsFactors=FALSE)

# Convert the data file according to the requirements

water\_json <- toJSON(water)

# Print out water\_json

print(water\_json)

**-Convert JSON file to prettify and minify :**

# jsonlite is already loaded

# Convert mtcars to a pretty JSON: pretty\_json

pretty\_json <- toJSON(mtcars, pretty=TRUE)

# Print pretty\_json

print(pretty\_json)

# Minify pretty\_json: mini\_json

mini\_json <- minify(pretty\_json)

# Print mini\_json

print(mini\_json)

**#IMPORTING DATA FROM STATISTICAL SOFTWARE**

**-haven package :**

* SAS: [**read\_sas()**](http://cran.r-project.org/web/packages/haven/haven.pdf)
* STATA: [**read\_dta()**](http://cran.r-project.org/web/packages/haven/haven.pdf) (or [**read\_stata()**](http://cran.r-project.org/web/packages/haven/haven.pdf), which are identical)
* SPSS: [**read\_sav()**](http://cran.r-project.org/web/packages/haven/haven.pdf) or [**read\_por()**](http://cran.r-project.org/web/packages/haven/haven.pdf), depending on the file type.

**-import sas file :**

# Load the haven package

library(haven)

# Import sales.sas7bdat: sales

sales <- read\_sas("sales.sas7bdat")

# Display the structure of sales

str(sales)

**-Import Stata file, and convert to factor :**

# haven is already loaded

# Import the data from the URL: sugar

sugar <- read\_dta("http://assets.datacamp.com/production/course\_1478/datasets/trade.dta")

# Structure of sugar

str(sugar)

# Convert values in Date column to dates

sugar$Date <- as.Date(as\_factor(sugar$Date))

# Structure of sugar again

str(sugar)

**-Import SPSS file :**

# haven is already loaded

# Import person.sav: traits

traits <- read\_sav("person.sav")

# Summarize traits

summary(traits)

# Print out a subset

subset(traits, Extroversion >40 & Agreeableness >40)

**-Import file from SPSS and change to as character :**

# haven is already loaded

# Import SPSS data from the URL: work

work <- read\_sav("http://s3.amazonaws.com/assets.datacamp.com/production/course\_1478/datasets/employee.sav")

# Display summary of work$GENDER

summary(work$GENDER)

# Convert work$GENDER to a factor

work$GENDER <- as\_factor(work$GENDER)

# Display summary of work$GENDER again

summary(work$GENDER)

**-Import stata with foreign package :**

# Load the foreign package

library(foreign)

# Import florida.dta and name the resulting data frame florida

florida <- read.dta("florida.dta")

# Check tail() of florida

tail(florida)

**-Import stata with foreign package(2) :**

# foreign is already loaded

# Specify the file path using file.path(): path

path <- file.path("worldbank", "edequality.dta")

# Create and print structure of edu\_equal\_1

edu\_equal\_1 <- read.dta(path)

str(edu\_equal\_1)

# Create and print structure of edu\_equal\_2

edu\_equal\_2 <- read.dta(path, convert.factors = FALSE)

str(edu\_equal\_2)

# Create and print structure of edu\_equal\_3

edu\_equal\_3 <- read.dta(path, convert.underscore = TRUE)

str(edu\_equal\_3)

**-Import SPSS data with foreign :**

# foreign is already loaded

# Import international.sav as a data frame: demo

demo <- read.spss("international.sav", to.data.frame=TRUE)

# Create boxplot of gdp variable of demo

boxplot(demo$gdp)

**-Import SPSS data with foreign (2) :**

# foreign is already loaded

# Import international.sav as demo\_1

demo\_1 <- read.spss("international.sav", to.data.frame=TRUE)

# Print out the head of demo\_1

head(demo\_1)

# Import international.sav as demo\_2

demo\_2 <- read.spss("international.sav", to.data.frame=TRUE,use.value.labels=FALSE)

# Print out the head of demo\_2

head(demo\_2)