Exercises 03

Bjarne Taulo Sørensen

Week 5 & 6 - 2022

if (!dir.exists("data")) dir.create("data")  
  
# Gem dine datasæt i dette bibliotek

# devtools::install\_github("ekstroem/dataReporter")  
pacman::p\_load("tidyverse", "magrittr", "nycflights13", "gapminder",  
 "Lahman", "maps", "lubridate", "pryr", "hms", "hexbin",  
 "feather", "htmlwidgets", "broom", "pander", "modelr",  
 "XML", "httr", "jsonlite", "lubridate", "microbenchmark",  
 "splines", "ISLR2", "MASS", "testthat", "leaps", "caret",  
 "RSQLite", "class", "babynames", "nasaweather",  
 "fueleconomy", "viridis", "boot", "devtools", "rvest",  
 "stringi", "readxl", "haven", "dataReporter")

## Installing package into 'C:/Users/Michael Davis/Documents/R/win-library/4.0'  
## (as 'lib' is unspecified)

## also installing the dependencies 'DEoptimR', 'robustbase', 'whoami'

## Warning: unable to access index for repository http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib/4.0:  
## kan ikke åbne adresse 'http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib/4.0/PACKAGES'

## package 'DEoptimR' successfully unpacked and MD5 sums checked  
## package 'robustbase' successfully unpacked and MD5 sums checked  
## package 'whoami' successfully unpacked and MD5 sums checked  
## package 'dataReporter' successfully unpacked and MD5 sums checked  
##   
## The downloaded binary packages are in  
## C:\Users\Michael Davis\AppData\Local\Temp\Rtmpiw2kyy\downloaded\_packages

##   
## dataReporter installed

## Warning: package 'dataReporter' was built under R version 4.0.5

# Problem 1

Make a random sample from **X201204** of 500 distinct employees. Make sure that these are not from *Afdeling=”AgroTech A/S”* and not part of the **sample01** data set.

# Problem 2

Use SQL!

Use the **flights** data and:

1. **airports**, *merge* by *origin=faa*
2. keep the following variables: *year*, *month*, *day*, *carrier*, *flight*, *tailnum*, *origin*, *dest*, *distance*, *name*
3. condition: output if *distance* greater than 2.5 the average *distance*.
4. **planes**, *merge* i.e.
5. *merge* the total sum of *flight* for each *tailnum* from **flights** with **planes**, but only if the total sum of *flight* for a given *tailnum* is greater than 10000.
6. **weather**
7. select all variables from **flights**, but keep only observations where *origin* from **weather** starts with an E and has two additional letters.
8. **airlines**
9. select all variables from **flights**, but drop observations where *name* from **airlines** includes lines or Lines.

Remember to make comments.

# Problem 3

Make comments to the following code:

Hide <- function(x){  
 cat("You entered the following number:", x)  
 invisible(x^1 + x^2 + x^3 + x^4 + x^5)  
 }  
  
Hide(2)  
y <- Hide(2)  
y

# Problem 4

Make comments to the following code:

set.seed(123)  
y <- stats::rnorm(10000000)  
table(cut(y, breaks = -6:6))

# Problem 5

What is the difference:

# I  
  
show\_missings <- function(df) {  
 n <- sum(is.na(df))  
 cat("Missing values: ", n, "\n", sep = "")  
 invisible(df)  
}  
  
mtcars %>%   
 show\_missings() %>%   
 mutate(mpg = ifelse(mpg < 20, NA, mpg)) %>%   
 show\_missings()   
  
  
# II  
  
show\_missings\_II <- function(df) {  
 n <- sum(is.na(df))  
 cat("Missing values: ", n, "\n", sep = "")  
}  
  
mtcars %>%   
 show\_missings\_II() %>%   
 mutate(mpg = ifelse(mpg < 20, NA, mpg)) %>%   
 show\_missings\_II()

# Problem 6

Use *strsplit* to split the strings and make each part accessible. Moreover, make the data tidy. Finally, make sure you end up with a *tibble*. Again, remember to make comments.

car\_french <- c("Product, Peugeot 108, Quantity, 2000, Year, 2019",  
 "Product, Peugeot 108, Price, 20000, Year, 2019",  
 "Product, Peugeot 208, Quantity, 1810, Year, 2018",  
 "Product, Peugeot 208, Price, 23400, Year, 2018",  
 "Product, Peugeot 308, Quantity, 1005, Year, New",  
 "Product, Peugeot 308, Price, 32500, Year, New"  
 )

# Problem 7

Please make a new similar program without using SQL. Assume that the data set **diamonds** exists outside the data base. Again, remember to make comments.

db\_diamonds <- dbGetQuery(con,  
 "select \*, price / size as value\_density from  
 (select carat, price, x \* y \* z as size from diamonds)"  
 )  
  
head(db\_diamonds, 3)

# Problem 9

Use the functions **semi\_join()** and **anti\_join()** instead of the first two **filter()** functions.

sample01 <- readRDS(file="data/sample01.Rda")  
sample02 <- readRDS(file="data/sample02.Rda")  
  
x1 <- readRDS(file="data/x201205.Rda") %>%   
 filter(Initialer %in% sample02$Initials) %>%  
 filter(!Initialer %in% sample01$Initials) %>%  
 filter(!trimws(Initialer) %in% c(NA, "-", "")) %>%  
 dplyr::select(Initialer) %>%   
 distinct()  
  
nrow(x1)

Remember to make comments.

# Problem 10

Append **a** and **b**, **a** and **c**, and **a** and **d**:

a <- tribble(  
 ~key, ~val\_x,  
 1, "x1",  
 2, "x2",  
 3, "x3",  
 4, "x4"  
)  
  
b <- tribble(  
 ~key, ~val\_y,  
 1, "y1",  
 2, "y2"  
)  
  
c <- tribble(  
 ~key, ~val\_x,  
 1, "z1",  
 2, "z2",  
 3, "z3",  
 4, "z4"  
)  
  
d <- tribble(  
 ~key, ~val\_x,  
 1, "x1",  
 2, "x2",  
 3, "x3",  
 4, "x4"  
)

Remember to make comments.

# Problem 11

Please make a new similar program without using the *switch()* function. Again, remember to make comments.

MMSD <- function(y, tp) {  
 switch(tp, mean = mean(y),   
 median = median(y),   
 sd = sd(y)  
 )  
 }   
y<-c(1, 4, 3, 7, 5, 9, 7, 3, 9, 1, 12)  
  
MMSD(y, "mean")  
MMSD(tp="mean", y=y)  
MMSD(y, "MEDIAN")

# Problem 12

Make comments to the following code:

bts <- c("XXXXXXbla-19960101T000000Z-1.tsv",  
 "XXXXXXbla-19960101T000000Z-2.tsv", "XXXXXXbla-19960101T000000Z-3.tsv")  
  
 Datafr <- data.frame(do.call(rbind, strsplit(bts, "-", fixed=TRUE)))  
 Datafr[,2] <- as.Date(Datafr[,2] , format="%Y%m%d")  
 Datafr[,3] <- as.integer(gsub(".tsv", "", Datafr[,3], fixed=TRUE))

# Problem 13

The data set **Exam** needs some cleaning up. First, remove observations if **idnumber02** is missing unless the employee is a woman, i.e. **Male=2**. Next, remove an observation if the employee is older than *68* (**Age**), or **Initials** contain a number. Call this new data set for **Exam3**.

Remember to make comments.

# Problem 14

Make comments to the following code:

data\_vector = c(9, 0.2, 5, 1, 23, 7, 2, 2, 5, .53)   
 categories = cut(data\_vector, breaks=c(0,1,5,max(data\_vector)), right = TRUE)   
 categories  
 class(categories)  
 typeof(categories)  
 table(categories)   
 levels(categories) = c("Good","Better","Best")  
 table(categories)

# Problem 15

Remember to make comments.

First, extract the 2nd column from the list of matrices (i.e. from each matrix).

Next, extract the 3rd row from the list of matrices.

Hint: use lapply and [.

x1 <- matrix(1:30, 3)  
x2 <- matrix(2:22, 3)  
x3 <- matrix(117:143, 3)  
x4 <- matrix(66:95, 5)  
list\_of\_matrices = list(x1, x2, x3, x4)  
list\_of\_matrices

# Problem 16

Make comments to the following code:

apply(iris[,1:4], 2, boxplot)  
  
  
round(apply(ggplot2::diamonds[10000:11000, 8:10], 2, mean), 1)  
  
purrr::flatten\_dbl(purrr::map(-2:2, rnorm, n = 6))  
  
?rnorm  
?flatten\_dbl  
purrr::map(-2:2, rnorm, sd = 2, n = 6)

# Problem 17

Make a program that rolls a die multiple times. The loop should continue until you get 8 sixes in a row. Calculate the total number of rolls.

Remember to make comments.

# Problem 18

Make a function that checks if a word is uppercase:

Remember to make comments.

# Problem 19

Make comments to the following code:

string\_rfba <- "11 april 2020"  
regexp <- "([[:digit:]]{2}) ([[:alpha:]]+) ([[:digit:]]{4})"  
x <- sub(regexp, "\\1", string\_rfba)  
y <- sub(regexp, "\\2", string\_rfba)   
z <- sub(regexp, "\\3", string\_rfba)

# Problem 20

Remember to make comments.

mtcars %>%  
 split(.$cyl) %>%  
 purrr::map(~ lm(mpg ~ wt, data = .)) %>%  
 purrr::map(summary) %>%  
 purrr::map\_dbl("r.squared")  
  
?split

Remember to make comments.

# Problem 21

Make comments to the following code:

geoms\_bts = list(  
 geom\_point(),  
 geom\_line(),  
 geom\_smooth())  
  
ggplot(mtcars, aes(hp, mpg)) + geoms\_bts

# Problem 22

Below, I have created a vector of all the mean values of all the columns of mtcars. Make some new code that uses a for loop to create the means. Can you also use lapply, sapply, purrr::map, map\_lgl, map\_int, map\_dbl, and map\_chr?

vectors\_of\_means <- c(mean(mtcars$mpg), mean(mtcars$cyl),   
 mean(mtcars$disp), mean(mtcars$hp),  
 mean(mtcars$drat), mean(mtcars$wt),   
 mean(mtcars$qsec), mean(mtcars$vs),  
 mean(mtcars$am), mean(mtcars$gear),   
 mean(mtcars$carb))

# Problem 23

Make some code that produces the following output:

## A B C  
## a 1 4 7  
## b 2 5 8  
## c 3 6 9

Remember to make comments.