ЛАБОРАТОРНА РОБОТА 1.3
НАПИСАННЯ ФУНКЦІЙ В R ДЛЯ ОБРОБКИ СТАТИСТИЧНИХ ДАНИХ

Мета роботи: одержати практичні навики у написанні функцій для обробки статистичних даних в R.

Обладнання:

– ПК IBM PC x86 CPU з встановленою операційною системою;

– встановлене програмне забезпечення R з оболонкою RStudio;

– встановлений в R пакет swirl;

– доступ до мережі інтернет.

1.3.1 Теоретичні відомості

Для одержання практичних навичок в написанні функцій в R виконайте наступні уроки в навчальному середовищі swirl():

8. Logic
9. Functions
14. Dates and Times

1.3.2 Порядок виконання роботи

1 Виконайте уроки 8, 9 і 14 в навчальному середовищі swirl.

2 Завантажити файл specdata.zip з відповідного каталогу лабораторної роботи. Архівований файл містить 332 файли, значення в яких розділені комами (CSV), що містять дані моніторингу забруднення повітря дрібними твердими частинками (PM) в 332 точках Сполучених Штатів. Кожен файл містить дані з однієї точки моніторингу, ідентифікаційний номер (ID) кожної точки моніторингу міститься в назві файлу. Наприклад, дані з точки моніторингу 200 міститься у файлі "200.csv". Кожен файл містить три змінні:

- дата: дата спостереження в форматі YYYY-MM-DD (рік-місяць-день);

- сульфат: рівень сульфату PM в повітрі в цей день (вимірюється в мікрограмах на кубічний метр);

- нітрат: рівень нітратів PM в повітрі в цей день (вимірюється в мікрограмах на кубічний метр).

3 Розпакуйте файл specdata.zip і створіть каталог 'specdata'. Зверніть увагу, що в кожному файлі є багато днів, коли або сульфат або нітрат (або обидва) не визначені (позначено NA).

4 Напишіть функцію з ім'ям 'pollutantmean', яка обчислює середнє значення для забруднюючої речовини (сульфат або нітрат) для певного переліку точок моніторингу. Функція 'pollutantmean' приймає три аргументи: “каталог”, “забруднювач” і “ідентифікатор”. Враховуючи заданий вектор точок моніторингу (їх ID), функція 'pollutantmean' читає дані про відповідний тип забруднення з каталогу, який відповідає точці забруднення і повертає середнє значення забруднення по всіх точках моніторингу, ігноруючи пропущені значення (NA). Прототип функції повинен виглядати наступним чином:

```
pollutantmean <- function(directory, pollutant, id = 1:332) {
## 'directory' is a character vector of length 1 indicating
## the location of the CSV files

## 'pollutant' is a character vector of length 1 indicating
## the name of the pollutant for which we will calculate the
## mean; either "sulfate" or "nitrate".

## 'id' is an integer vector indicating the monitor ID numbers
## to be used

## Return the mean of the pollutant across all monitors list
## in the 'id' vector (ignoring NA values)
}
```

Результат роботи функції повинен бути наступним:

```
source("pollutantmean.R")
pollutantmean("specdata", "sulfate", 1:10)
## [1] 4.064
pollutantmean("specdata", "nitrate", 70:72)
## [1] 1.706
pollutantmean("specdata", "nitrate", 23)
## [1] 1.281
```

Написана функція повинна давати наведений результат. Код необхідно зберегти у файл з ім'ям pollutantmean.R.

5 Написати функцію, яка зчитує каталог з файлами і повідомляє про кількість повністю спостережуваних випадків в кожному файлі даних. Функція повинна повертати фрейм даних, де перший стовпець - це ім'я файлу, а другий стовпець - число повних випадків. Прототип цієї функції виглядає так:

```
complete <- function(directory, id = 1:332) {

## 'directory' is a character vector of length 1 indicating
## the location of the CSV files

## 'id' is an integer vector indicating the monitor ID numbers
## to be used

## Return a data frame of the form:
## id nobs
## 1 117
## 2 1041
## ...
## where 'id' is the monitor ID number and 'nobs' is the
## number of complete cases
}
```

Результат роботи функції повинен бути наступним:

```
source("complete.R")
complete("specdata", 1)
##   id nobs
## 1  1  117
complete("specdata", c(2, 4, 8, 10, 12))
##   id nobs
## 1  2 1041
## 2  4  474
## 3  8  192
## 4 10  148
## 5 12   96
complete("specdata", 30:25)
##   id nobs
## 1 30  932
## 2 29  711
## 3 28  475
## 4 27  338
## 5 26  586
## 6 25  463
complete("specdata", 3)
##   id nobs
## 1  3  243
```

Код необхідно зберегти у файл з ім'ям complete.R.

5 Написати функцію, яка в якості аргументів бере каталог файлів даних і поріг для випадків повного спостереження і обчислює кореляцію між сульфатом і нітратом для точки моніторингу, де кількість повністю спостережуваних випадків (по всіх змінних) більша, ніж поріг. Функція повинна повертати вектор кореляцій для точок моніторингу, які відповідають вимогам порогу. Якщо немає точок моніторингу, які відповідають вимогам порогу, то функція повинна повертати числовий вектор довжиною 0. Прототип цієї функції:

```
corr <- function(directory, threshold = 0) {
## 'directory' is a character vector of length 1 indicating
## the location of the CSV files
```

```
## 'threshold' is a numeric vector of length 1 indicating the
## number of completely observed observations (on all
## variables) required to compute the correlation between
## nitrate and sulfate; the default is 0

## Return a numeric vector of correlations
}
```

Для цієї функції застосуйте функцію 'cor' в R, яка обчислює кореляцію між двома векторами.

Результат роботи функції повинен бути наступним:

```
source("corr.R")
source("complete.R")
cr <- corr("specdata", 150)
head(cr)
## [1] -0.01896 -0.14051 -0.04390 -0.06816 -0.12351 -0.07589
summary(cr)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.2110 -0.0500  0.0946  0.1250  0.2680  0.7630
cr <- corr("specdata", 400)
head(cr)
## [1] -0.01896 -0.04390 -0.06816 -0.07589  0.76313 -0.15783
summary(cr)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.1760 -0.0311  0.1000  0.1400  0.2680  0.7630
cr <- corr("specdata", 5000)
summary(cr)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##
length(cr)
## [1] 0
cr <- corr("specdata")
summary(cr)
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.0000 -0.0528  0.1070  0.1370  0.2780  1.0000
length(cr)
## [1] 323
```

Код необхідно зберегти у файл з ім'ям corr.R.

7 Розмістити написані файли у власному обліковому записі на GitHub у репозиторії lab_STSPS.

8 Оформити звіт.

1.3.3 Зміст звіту

Звіт повинен містити:

– титульний аркуш;

– мету роботи і завдання;

– покроковий опис роботи, копії екранів з пройденими уроками (100% для кожного уроку); код написаних функцій; коментарі до реалізації написаних функцій; резальтати тестових

запусків функцій; копії екранів з відображенням вмісту створеного репозиторію на GitHub.

- висновки.

Запитання для самоконтролю:

Тривалість заняття: 4 год.