National Institute of Technology Karnataka, Surathkal

Department of Computer Science and Engineering

Big Data Analytics (CS473)

Quiz 1

Submitted to

Dr. Asoke Talukder
Adjunct Professor
Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal

Submitted by

Manas Trivedi
B.Tech. Class of 2022
Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal
Roll no. 181CO231

Date: January 23, 2021

Index

Problem Statement

Quiz1.csv is a CSV file that contains two attributes, namely "STITCH" and "Compound". There are 644 unique "STITCH" attributes and 613 "Compound" attributes.
Logically, the number of "STITCH" attributes and the number of "Compound" attributes will be the same.
Find out what and where is the problem.

Solutions

Solution 1 - Using a set to identify duplicate values

We expect that there will be a one-to-one correspondence between STITCH and Compound. But since the number of unique STITCH values (644) is more than the number of unique Compound values (613), it is evident that such a correspondence does not exist in the given dataset.

Hypothesis 1:
Since a one-to-one correspondence does not exist, it means that there *must* be one or more Compound values that each correspond to multiple STITCH values.

Hypothesis 2:
In addition, it is also possible that there exist one or more STITCH values that correspond to multiple Compound values.

We begin with cleaning the environment, reading the CSV into Quiz1File, creating a dataframe called Quiz1Int from the same, eliminating duplicate rows from Quiz1Int, and finding the number of resulting rows.

```
> rm(list=ls())
> library("sets")
> startTime <- Sys.time()
> Quiz1File <- "C:/Users/Manas/Learn/BDA/Quiz1/Quiz1.csv"
> Quiz1Int <- read.csv(Quiz1File)
> Quiz1Int <- unique(Quiz1Int)
> print(nrow(Quiz1Int))
[1] 644
```

Since the number of resulting rows (644) is exactly equal to the number of unique STITCH values, hence Hypothesis 2 is false. Hence, none of the STITCH values correspond to multiple Compound values.

Now, since none of the STITCH values correspond to multiple Compound values, and we know that a one-to-one correspondence between STITCH and Compound does not exist in the given dataset, hence we must have one or more Compound values that each correspond to multiple STITCH values. Hence, Hypothesis 1 is correct.

We now find the Compound value(s) that correspond to multiple STITCH values. We do this by finding the Compound values that are duplicate, and saving the corresponding rows for observation.

We iterate through every row of Quiz1Int and store unique Compound values in a set. If a duplicate Compound value is found (i.e. a value that already exists in the set), we store the corresponding row in a new data frame called rowsForReview.

```
> s <- set()
> rowsForReview <- data.frame (STITCH = c(), Compound = c())
>
> for (row in 1:nrow(Quiz1Int)) {
+   if (set_contains_element(s, Quiz1Int[row, "Compound"])) {
+     rowsForReview <- rbind(rowsForReview, Quiz1Int[row,])
+   }
+   else {
+     s <- set_union(s, set(Quiz1Int[row, "Compound"]))
+   }
+ }
```

rowsForReview now contains all the rows from Quiz1Int that have a duplicate Compound value, and hence needs to be reviewed. We begin our review of these rows by initially viewing the first few rows using the head() function.

```
> print(head(rowsForReview))
            STITCH Compound
165926 CID000003161
169862 CID000005647
171668 CID000002022
176125 CID000003043
185502 CID000004585
187030 CID000005412
```

This initial analysis points to the fact that many of the duplicate Compound values are empty strings. We now view the 'unique' duplicate Compound values to ascertain whether there are any duplicate Compound values that are not empty strings.

```
> print(unique(rowsForReview$Compound))
[1] ""
```

This means that all of the duplicate Compound values are empty strings.

We are at the end of our analysis. There is just one row that has not been added to the rowsForReview, which is the very first row that contained an empty Compound value (and was hence treated as a unique Compound value rather than a duplicate). We add that row at the beginning of the rowsForReview dataframe.

```
> for(row in 1:nrow(Quiz1Int)) {
+   if (Quiz1Int[row, "Compound"] == "") {
+     rowsForReview <- rbind(Quiz1Int[row,], rowsForReview)
+     break
+   }
+ }
> timeTakenForSoln1 <- Sys.time() - startTime
> print(timeTakenForSoln1)
```

```
Time difference of 6.361 secs
```

Solution 2 - Using duplicated()

This solution begins along the same lines as Solution 1 i.e. prove Hypothesis 2 to be false and Hypothesis 1 to be correct. Corresponding code from Solution 1:

```
> rm(list=ls())
> startTime <- Sys.time()
> Quiz1File <- "C:/Users/Manas/Learn/BDA/Quiz1/Quiz1.csv"
> Quiz1Int <- read.csv(Quiz1File)
> Quiz1Int <- unique(Quiz1Int)
> print(nrow(Quiz1Int))
[1] 644
```

The truth of Hypothesis 1 reduces the problem statement to finding Compound values that are duplicate, and saving the corresponding rows for observation, as in Solution 1.

But instead of using a set for saving unique Compound values and identifying duplicate values as in Solution 1, we shall directly use the duplicated() function, and store the rows that contain duplicate Compound values in the rowsForReview dataframe.

```
> rowsForReview <- Quiz1Int[duplicated(Quiz1Int$Compound),]
```

The rest of the solution is the same as that of Solution 1, where we analyze the first few rows of rowsForReview, and arrive at the conclusion that all duplicate Compound values are empty strings by using the unique() function.

```
> print(head(rowsForReview))
             STITCH Compound
165926 CID000003161
169862 CID000005647
```

```
171668 CID000002022
176125 CID000003043
185502 CID000004585
187030 CID000005412
> print(unique(rowsForReview$Compound))
[1] ""
```

Ultimately, we add the first row that contains an empty Compound value at the beginning of the rowsForReview data frame, as in Solution 1.

```
> for(row in 1:nrow(Quiz1Int)) {
+   if (Quiz1Int[row, "Compound"] == "") {
+     rowsForReview <- rbind(Quiz1Int[row,], rowsForReview)
+     break
+   }
+ }
> timeTakenForSoln2 <- Sys.time() - startTime
> print(timeTakenForSoln2)
Time difference of 4.329735 secs
```

Solution 3 - Using nzchar() and is.na()

The previous two solutions were both experimental in nature, i.e. two hypotheses were constructed based on the fact that a one-to-one correspondence between STITCH and Compound did not exist, followed by proving one and disproving the other, and ultimately finding duplicate Compound values and storing the corresponding rows.

We can have a more direct approach to solving the given problem statement. We know that there are 644 unique rows in the dataset. There are 644 unique STITCH values and 613 unique Compound values. Since it is known that the STITCH values were used to find the corresponding Compound values on the internet, it can be assumed that some of the STITCH values may not have resulted in a Compound value, thereby leaving the value empty.

The problem reduces to finding all Compound values that are empty or NA, and storing the corresponding rows in a rowsForReview dataframe.

```
> rm(list=ls())
> startTime <- Sys.time()
> Quiz1File <- "C:/Users/Manas/Learn/BDA/Quiz1/Quiz1.csv"
> Quiz1Int <- read.csv(Quiz1File)
> Quiz1Int <- unique(Quiz1Int)
>
> rowsForReview <- Quiz1Int[!nzchar(Quiz1Int$Compound) |
+ is.na(Quiz1Int$Compound),]
> timeTakenForSoln2 <- Sys.time() - startTime
> print(timeTakenForSoln2)
Time difference of 4.662559 secs
```

Innovations

Innovation 1 - Bar Plot and Pie Chart

In the above three solutions, we have created a rowsForReview data frame that contains all the rows that have an empty Compound value. We can contrast the number of rowsForReview with the number of rows that are approved i.e. have a one-to-one correspondence between STITCH and Compound. To visualize this contrast, we construct a bar plot and a pie chart. We will use the code from Solution 3 for constructing the rowsForReview data frame, and create the bar plot and pie chart by using the number of rowsForReview and the total number of rows in Quiz1Int.

```
> rm(list=ls())
>
> startTime <- Sys.time()
>
> Quiz1File <- "C:/Users/Manas/Learn/BDA/Quiz1/Quiz1.csv"
> Quiz1Int <- read.csv(Quiz1File)
> Quiz1Int <- unique(Quiz1Int)
>
> rowsForReview <- Quiz1Int[!nzchar(Quiz1Int$Compound) |
+ is.na(Quiz1Int$Compound),]
>
> nrowVector <- c(nrow(Quiz1Int) - nrow(rowsForReview),
+ nrow(rowsForReview))
>
> label1 <- paste("STITCH-Compound Matches (", nrow(Quiz1Int) -
+ nrow(rowsForReview), ")")
> label2 <- paste("Unmatched STITCH (", nrow(rowsForReview), ")")
> labelVector <- c(label1, label2)
>
> jpeg(file="C:/Users/Manas/Learn/BDA/Quiz1/barplot.jpeg")
> barplot(nrowVector, names.arg = labelVector, main =
+ "STITCH-Compound Matches and Unmatched STITCH")
> dev.off()
>
> jpeg(file="C:/Users/Manas/Learn/BDA/Quiz1/pie.jpeg")
> pie(nrowVector, labels = labelVector, main = "STITCH-Compound
+ Matches and Unmatched STITCH")
> dev.off()
>
> timeTakenForInno1 <- Sys.time() - startTime
> print(timeTakenForInno1)
Time difference of 4.687641 secs
```

Innovation 2 - Number of drugs with no drug-drug interaction

The given dataset (Quiz1.csv) was created by obtaining STITCH values from three columns - the STITCH column from ChSe-Decagon_monopharmacy.csv and the STITCH1 and STITCH2 columns from ChChSe-Decagon_polypharmacy.csv.

ChSe-Decagon_monopharmacy.csv contains STITCH, Individual Side Effect, and Side Effect Name.

ChSeSe-Decagon_polypharmacy.csv contains STITCH1, STITCH2, Polypharmacy Side Effect, and Side Effect Name.

We now aim to find whether any of the compounds present in the Monopharmacy dataset are free of drug-drug interaction by noting whether or not they exist in the Polypharmacy dataset.

```
> rm(list=ls())
>
> startTime <- Sys.time()
>
> DrugSideEffect1File <-
+ "C:/Users/Manas/Learn/BDA/Quiz1/ChSe-Decagon_monopharmacy.csv"
> DrugSideEffect1Int <- read.csv(DrugSideEffect1File)
> DrugSideEffect1Int <- unique(DrugSideEffect1Int)

> DrugSideEffect2File <-
+
"C:/Users/Manas/Learn/BDA/Quiz1/ChChSe-Decagon_polypharmacy.csv"
> DrugSideEffect2Int <- read.csv(DrugSideEffect2File)
> DrugSideEffect2Int <- unique(DrugSideEffect2Int)

> drugsWithNoDrugDrugInteraction <-
+ DrugSideEffect1Int[!((DrugSideEffect1Int$X..STITCH %in%
+ DrugSideEffect2Int$X..STITCH.1) | (DrugSideEffect1Int$X..STITCH
+ %in% DrugSideEffect2Int$STITCH.2)),]
>
> dim(drugsWithNoDrugDrugInteraction)
[1] 0 3
>
> timeTakenForInno2 <- Sys.time() - startTime
> print(timeTakenForInno2)
Time difference of 18.4528 secs
```

Since the dimensions of the drugsWithNoDrugDrugInteraction data frame is (0, 3), i.e. 0 rows and 3 columns, hence there are no drugs present in the Monopharmacy dataset that do not have any drug-drug interactions i.e. all of them are present in the Polypharmacy dataset.

Report

Time taken to write the code: 10-12 hours (spread over 3 days, 90% for debugging)
Time taken for executing the code:

- Solution 1:	6.361 secs
- Solution 2:	4.329735 secs
- Solution 3:	4.662559 secs
- Innovation 1:	4.687641 secs
- Innovation 2:	18.4528 secs