Task 5: Further Data Transformations

James Mbewu

13/06/2021

Overview

This R Markdown document serves as the report and code for Task 5: Further Data Transformations for the City of Cape Town Data Science Unit Code Challenge. The code by itself can be found in the file further_data_transformations.R.

Brief

Write a script which anonymises the `sr_hex.csv` file, but preserves the following resolutions (You may use H3 indexes or lat/lon coordinates for your spatial data):

- location accuracy to within approximately 500m
- temporal accuracy to within 6 hours
- scrubs any columns which may contain personally identifiable information.

We expect in the accompanying report that follows you will justify as to why this data is now anonymised. Please limit this commentary to less than 500 words. If your code is written in a code notebook such as Jupyter notebook or Rmarkdown, you can include this commentary in your notebook.

Report

The goal of this task is to anonymise the service request dataset `sr_hex.csv` while retaining a degree of spatial and temporal accuracy. We need to decide how to anonymise the different types of data that are present in the dataset.

Data Anonymisation

Data anonymisation is the task of mutating data so it contains less information than you started. It is important for a number of reasons such as personal data protection, removing bias in use of the data and removing bias in statistical analyses. In general it has become an important issue with the rise of online advertising and tracking, high profile data hacks, and the increased use of machine learning models in all spheres of life. It is important to anonymise carefully so that you don't remove all of the information in the data and render the data useless. You also don't always want the data to look like it has been anonymised and encourage further digging. There are many techniques that can be used to anonymise data, some of which we will apply here to this dataset.

The dataset `sr_hex.csv` contains data on service requests that the City of Cape Town has processed. The reasons it should be anonymised depend on the use case:

- We could want it to be anonymised so that any personal information (such as names, addresses, location, phone numbers, medical conditions etc.) that may have slipped into it can't be hacked or used improperly by people with access to the data. Some data might seem to not contain much information, but combined with other data it could prove more revealing.

- We could want it to be anonymised so that users of the data aren't biased in any way, for example towards or against a particular suburb or perhaps a type of service request like fixing a playground might be deemed less important.

- We might also want to anonymise it for statistical analysis so that inferences are not made with unconscious bias.

Here we detail how and why we have anonymised different attributes in the dataset. As stated above, there are many reasons to anonymise data, and what you anonymise may vary:

- **Code** - *Removed*:
  - this attribute contains more detailed human entered information that has the potential to contain personally identifiable information. It will be removed.

- **Latitude** and **Longitude** - *Perturbed*:
  - this spatial data could be used to pinpoint the exact location of a service request and should be changed so it can't be used for nefarious purposes. We still want to retain some location data so we will perturb it by a maximum of 500m.

- **h3_level_8_index** - *Removed*:
  - this could either be removed or recalculated for the perturbed location. If it was kept the same it could be used to triangulate a more accurate location than we want to allow.

- **SubCouncil2016**,**Wards2016** and **OfficialSuburbs** - *Removed*:
  - this attribute contains location information that may bias use of the data or enable linking with other datasets and used improperly. They can always be approximated from the perturbed location if necessary.

- **CreationTimestamp**,**CreationDate**,**CompletionTimestamp**,**CompletionDate**, **ModificationTimestamp** and **Duration** - *Perturbed*:
  - this temporal data could be used for example to infer things like when CCT workers are in the area. We still want to retain some of the data so we will perturb them by a maximum of 6 hours. Duration will be recalculated from this so that it remains consistent.

- **CodeGroup**,**directorate**,**department** - *Shuffled together*:
  - this set of data contains information on the departments that dealt with the service request. They are unlikely to contain personal information, but could bias the use of the data. We will shuffle the order of them, while grouping them together, to prevent this bias while still retaining some information.

- **NotificationNumber** - *Shuffled*:
  - this data could probably be used to link to other datasets and be used to infer more information. They will be shuffled.

- **Open**,**NotificationType** - *Retained*:
  - this data is not likely to contain any personal information or enable linking as it doesn't change much. It will be retained

Load Data

First we download the dataset if it isn't present already and then read it into an R data frame.

```r
rm(list = ls())
```

```
num_lines <- Inf
# LOAD DATA ==========================================================

start <- Sys.time()

# Download data if not present
dir.create("data",showWarnings = FALSE)

sr_data_filename <- "data/sr.csv"
sr_gz_data_filename <- "data/sr.csv.gz"
if(!file.exists(sr_data_filename)) {
  sr_data_url <- "https://cct-ds-code-challenge-input-data.s3.af-south-1.amazonaws.com/sr.csv.gz"
  download.file(sr_data_url, sr_gz_data_filename)
  gunzip(sr_gz_data_filename, remove=FALSE)
}

end_download <- Sys.time()
download_time <- difftime(end_download, start)
print(paste("download_time =",download_time))
```

```
## [1] "download_time = 0.00710678100585938"
```

```
start_load <- Sys.time()

# Read in data
# For testing purposes we are only going to read in a subset of the data
sr_hex_data <- read_csv("data/sr_hex.csv",n_max = num_lines,skip = 0, col_types = cols())
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Warning: 41636 parsing failures.
##  row                   col   expected actual              file
## 2099 ModificationTimestamp date like     NaT 'data/sr_hex.csv'
## 2182 ModificationTimestamp date like     NaT 'data/sr_hex.csv'
## 2230 ModificationTimestamp date like     NaT 'data/sr_hex.csv'
## 2415 ModificationTimestamp date like     NaT 'data/sr_hex.csv'
## 2721 ModificationTimestamp date like     NaT 'data/sr_hex.csv'
## .... ..................... .......... ...... .................
## See problems(...) for more details.
```

```
end_load <- Sys.time()
load_time <- difftime(end_load, start_load)
print(paste("load_data_time =",load_time))
```

```
## [1] "load_data_time = 29.8501281738281"
```

Data Anonymisation

Next we perform the data anonymisation on the data frame. We first remove and then shuffle any columns that need to be shuffled.

```
# DATA ANONYMISATION =================================================



start_anonymisation <- Sys.time()


```

```
# Remove SubCouncil2016, Wards2016, OfficialSuburbs, Code and h3_level8_index
# NotificationType and NotificationNumber
drop_cols <- c("SubCouncil2016","Wards2016","OfficialSuburbs","Code",
               "h3_level8_index")
sr_hex_data <- sr_hex_data[ , !(names(sr_hex_data) %in% drop_cols)]

# Shuffle CodeGroup and directorate and department together
shuffle_cols_dep <- c("CodeGroup","directorate","department")
sr_hex_data[, shuffle_cols_dep] <- sr_hex_data[sample(1:nrow(sr_hex_data)), shuffle_cols_dep]

# Shuffle NotificationNumber
shuffle_cols_not <- c("NotificationNumber")
sr_hex_data[, shuffle_cols_not] <- sr_hex_data[sample(1:nrow(sr_hex_data)), shuffle_cols_not]
```

Then we do the random perturbations of the spatial data. We want the resulting location to be a maximum of about 500m away from the original location. This corresponds to a change in latitude and longitude of 350m. Around Cape Town, 350m in latitude is approximately 0.00314 degrees, while 350m in longitude is approximately 0.00382 degrees.

```
# Do numerical transforms

# Spatial transforms
# add/subtract max 350m from the latitude and longitude
# 350m ~ 0.00314 deg lat and 0.00382 deg lon around Cape Town
set.seed(NULL)
lat_350 <- 0.00314
lon_350 <- 0.00382
# check it gives different random numbers
sr_hex_data <- sr_hex_data %>% mutate(Latitude = Latitude + runif(n(),-lat_350,lat_350),
                                      Longitude = Longitude + runif(n(),-lon_350,lon_350))
```

Next we do the random perturbations of the temporal data. CreationTimestamp and CompletionTimestamp are perturbed by a maximum of 3 hours so that the resulting Duration is only perturbed by a maximum of 6 hours. We also make sure the resulting Duration cannot be negative. The ModificationTimestamp is perturbed by the same amount as the CompletionTimestamp as they appear to be linked in the data.

```
# Temporal transforms
# convert CompletionTimestamp to POSIXct
sr_hex_data <- sr_hex_data %>% mutate(CompletionTimestamp = str_sub(CompletionTimestamp,end=-7),
                                      CompletionTimestamp = as.POSIXct(CompletionTimestamp,format = "%Y
                                      CompletionTimestamp = CompletionTimestamp - 2*60*60)

# Add/subtract 3hrs from CreationTimestamp and CompletionTimeStamp/ModificationTimestamp
#  and adjust Duration and CreationDate and CompletionDate accordingly.
# Make sure Duration doesn't go negative by only allowing CreationTimestamp to
#  move forward 1/2 Duration and CompletionTimestamp only move back 1/2 Duration
sr_hex_data <- sr_hex_data %>% mutate(CreationTimestamp = CreationTimestamp +
                                        as.integer(round(runif(n(),-3*60*60,min(3*60*60,Duration/2*60*6
                                      CompletionPerturbation = as.integer(round(runif(n(),max(-3*60*60,
                                      CompletionTimestamp = CompletionTimestamp + CompletionPerturbatio

                                      ModificationTimestamp = ModificationTimestamp + CompletionPerturba
                                      Duration = (as.integer(CompletionTimestamp) - as.integer(Creation
                                      CreationDate = as.Date(CreationTimestamp),
                                      CompletionDate = as.Date(CompletionTimestamp)) %>%
                              select(-CompletionPerturbation)


end_anonymisation <- Sys.time()
anonymisation_time <- difftime(end_anonymisation, start_anonymisation)
print(paste("anonymisation_time =",anonymisation_time))
```

```
## [1] "anonymisation_time = 16.9904906749725"
```

Finally, we write the resulting dataframe to file.

```
# WRITE TO FILE ==========================================================

start_write <- Sys.time()

# Format CreationTimestamp, CompletionTimestamp, ModificationTimestamp for output
timestamp_format <- "%Y-%m-%d %H:%M:%S%z"
sr_hex_data <- sr_hex_data %>%
  mutate(CreationTimestamp = format(CreationTimestamp,format = timestamp_format),
         CompletionTimestamp = format(CompletionTimestamp,format = timestamp_format),
         ModificationTimestamp = format(ModificationTimestamp,format = timestamp_format))

# Write to file
col_names <- colnames(sr_hex_data)
col_names[1] <- ""
write.table(sr_hex_data,file="data/sr_hex_anon.csv",
            row.names = FALSE, col.names = col_names, sep=",",quote=FALSE)

end_write <- Sys.time()
write_time <- difftime(end_write, start_write)
print(paste("write_time =",write_time))
```

```
## [1] "write_time = 1.69600864648819"
```

```
total_time <- difftime(end_write, start)
print(paste("total_time =",total_time))
```

```
## [1] "total_time = 2.47744425535202"
```

Performance

The time taken to perform these operations was logged to the command line and is presented in the table below.

```
# Table showing the logged times for each subsection and the total
log_times <- c(download_time,load_time,anonymisation_time,
               write_time,total_time)
log_times_df <- data.frame(Times = log_times)
colnames(log_times_df) <- c("Times")
rownames(log_times_df) <- c("Download","Load","Anonymisation",
                            "Write","Total")
kable(log_times_df, caption = "Log times for Further Data Transformations Task")
```

Table 1: Log times for Further Data Transformations Task

	Times
Download	0.0071068 secs
Load	29.8501282 secs
Anonymisation	16.9904907 secs
Write	101.7605188 secs
Total	148.6466553 secs

Conclusions

We have now successfully anonymised the dataset `sr_hex.csv`.