Task 1: Data Extraction

James Mbewu

13/06/2021

Overview

This R Markdown document serves as the report and code for Task 1: Data Extraction for the City of Cape Town Data Science Unit Code Challenge. The code by itself can be found in the file data_extraction.R.

Brief

Use the AWS S3 SELECT command to read in the H3 resolution 8 data from `city-hex-polygons-8-10.geojson`. Use the `city-hex-polygons-8.geojson` file to validate your work.

Please log the time taken to perform the operations described, and within reason, try to optimise latency and computational resources used.

Report

The crux of this task is to use AWS S3 SELECT to query and read in the H3 resolution data of level 8 from a larger dataset in a bucket on AWS S3 that contains H3 resolution data of levels 8, 9 and 10. For this I used the R package paws.

The query was performed and the dataset extracted was validated by comparing H3 resolution indexes with an existing dataset that contained only H3 resolution data of level 8. The time to complete the task and subtasks was logged.

AWS credentials

We first load the AWS S3 credentials from the link provided and set the appropriate environmental variables that paws will use to access files in the AWS S3 bucket. For security reasons it is important to explicitly write these kinds of credentials directly in the file. Then we establish the connection with AWS S3 using paws.

```
rm(list = ls())

start <- Sys.time()

# Setup AWS S3 access
aws_credentials_url <- "https://cct-ds-code-challenge-input-data.s3.af-south-1.amazonaws.com/ds_code_cha
secrets <- fromJSON(aws_credentials_url)

Sys.setenv(
  "AWS_ACCESS_KEY_ID" = secrets$s3$access_key,
  "AWS_SECRET_ACCESS_KEY" = secrets$s3$secret_key,
  "AWS_DEFAULT_REGION" = "af-south-1",
  "AWS_REGION" = "af-south-1"
)
```

```
# Setup connection to AWS S3
s3 <- paws::s3()
```

Extracting the data

It is much more efficient to extract only the data we require than downloading all the data and filtering it locally. To extract the level 8 data we need to query the features array within the .geojson file and extract all the attributes from features that represent level 8 H3 hexes. The response is passed back in json format and then parsed into an R data frame.

```
# EXTRACT RES 8 FEATURES ================================

# Setup query to select only data from features with resolution 8
query <- "SELECT d.type, d.properties, d.geometry FROM  S3Object[*].features[*] d WHERE d.properties.res

# Perform query
extracted_h3_res8 <- s3$select_object_content(
  Bucket = "cct-ds-code-challenge-input-data",
  Key = "city-hex-polygons-8-10.geojson",
  Expression = query,
  ExpressionType = "SQL",
  RequestProgress = list(
    Enabled = TRUE
  ),
  InputSerialization = list(
    JSON = list(
      Type = "DOCUMENT"
    )
  ),
  OutputSerialization = list(
    JSON = list(
      RecordDelimiter = ","
    )
  )
)

# Parse query response to be in json array format and convert to df
extracted_h3_res8 <- extracted_h3_res8$Payload$Records$Payload %>%
  str_sub(1,-1L -1)
extracted_h3_res8 <- paste0("[",extracted_h3_res8,"]")

extracted_h3_res8_df <- extracted_h3_res8 %>%
  jsonlite::fromJSON()


end_extraction <- Sys.time()
extraction_time <- difftime(end_extraction, start)
print(paste("extraction_time =",extraction_time))
```

```
## [1] "extraction_time = 5.48149657249451"
```

Validation

To validate that we extracted all the correct features, we compare the H3 level 8 indexes of the extracted data with a dataset that only includes the H3 level 8 data and was provided to us. If they are all the same then the validation test has passed. In this case we get the file directly from the AWS S3 bucket and convert it to an R dataframe.

```
# VALIDATION ==================

start_validation <- Sys.time()

# Get city-hex-polygons-8.geojson for validation
h3_res8_obj <- s3$get_object(Bucket = "cct-ds-code-challenge-input-data",
              Key = "city-hex-polygons-8.geojson")

# Convert from raw to df
h3_res8_df <- h3_res8_obj$Body %>%
  rawToChar() %>%
  jsonlite::fromJSON()

# Compare indexes to make sure they are the same (assume in same order)
indexes_from_8_10_data <- extracted_h3_res8_df$properties$index
indexes_from_8_data <- h3_res8_df$features$properties$index

validation_result <- all(indexes_from_8_data == indexes_from_8_10_data)
if(validation_result) {
  print("Validation passed! :)")
} else
{
  print("Validation failed! :(")
}
```

```
## [1] "Validation passed! :)"
```

```
end_validation <- Sys.time()
validation_time <- difftime(end_validation, start_validation)
print(paste("validation_time =",validation_time))
```

```
## [1] "validation_time = 5.82794761657715"
```

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

```
## [1] "total_time = 11.3235890865326"
```

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(extraction_time,validation_time,total_time)
log_times_df <- data.frame(Times = log_times)
rownames(log_times_df) <- c("Extraction","Validation","Total")
kable(log_times_df, caption = "Log times for Data Extraction Task")
```

Table 1: Log times for Data Extraction Task

	Times
Extraction	5.481497 secs
Validation	5.827948 secs

	Times
Total	11.323589 secs

Conclusions

We have successfully queried the H3 resolution data and extracted only the H3 level 8 data from the AWS S3 bucket. The extracted data was validated with an existing file to test if the extraction was successful.