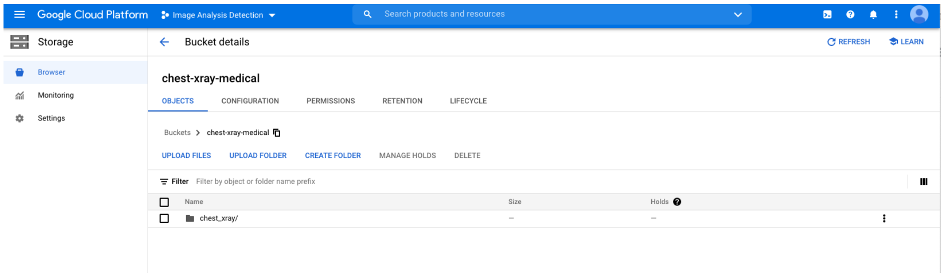
CloudStorage

Graham Chickering

11/15/2020

Google Cloud Storage

When trying to work with Big Data, one of the first questions one has to answer is what is the best way to store and access this data. While smaller files can be stored directly on your computer and eventually loaded into R Studio to perform analysis on, when data moves into the gigabyte, terabyte, or even petabyte range one may not not want to store this data directly on their machine and use of large chunks of their limited memory that is available. With the advancement of cloud service solutions in recent years though, one can now use a platform such as Google Cloud Platform, Amazon Web Services, or Microsoft Azure, to store large amounts of data directly on these platforms and take advantage of these companies large data warehouses for a small cost. This can allow one to free up space and memory on their own personal machine and access the data directly from these servers whenever they desire.



□	Name	Size	Holds
□	chest_xray/	–	–

Figure 1: Google Cloud Storage Platform

For this project, I am going to focus on how to setup and store medical imagery in Google Cloud Storage and learn how this can be connected to R Studio so that I can then perform analysis on these images. The data set I will be working with is a labeled Chest X-Ray images [@Medical] which will be used to detect and classify whether or not someone has pneumonia. This data set is roughly 2 GB in size and contains CT scans of patients who either have or dont have pneumonia. Due to Github having maximum storage limit of 1 GB, I wanted to look into ways that would allow me to work with a data set of this size and not have to upload the data directly into Github itself. One of the solutions for this was to use Google Cloud Storage and take advantage of the free credits that Google offers for new users using their service. By uploading and storing the data set directly onto the Google platform, this meant I could delete the data set off my computer for the time being and free up memory space (See Figure 2 for an example of the Google Cloud Platform Console).

In order to actually work with this data in R Studio though, a connection between R and Google Cloud was required to be setup. By utilizing the *googleCloudStorageR* package, I was able to setup a connection that allowed me to download the data from my Cloud Storage bucket and onto my desktop where it could then be read into R Studio without having to store the files themselves within R Studio [@GoogleyR]. This allowed me to not have to store the data in my actual Github repository but still be able to perform analysis and build models on the image dataset.

Here we check your buckets that are associated and available with your own personal account_credentials. You should see a bucket named "chest-xray-medical", if not please reach out to me!

```
gcs_auth(json_file="account_credentials.json")
gcs_list_buckets("image-analysis-detection")
```

```
##                        name storageClass                location
## 1        chest-xray-medical     STANDARD NORTHAMERICA-NORTHEAST1
## 2          chest-xray-tests     STANDARD                      US
## 3          chest-xray-train     STANDARD                      US
## 4     chest-xray-validation     STANDARD                      US
## 5 image-analysis-detection     STANDARD                      US
##               updated
## 1 2020-11-19 20:59:24
## 2 2020-11-13 21:17:58
## 3 2020-11-13 21:16:28
## 4 2020-11-13 21:19:01
## 5 2020-11-16 05:38:39
```

This reads all the files that are stored in the google cloud storage bucket that you want to work with as objects.

```
gcs_get_bucket("chest-xray-medical")
```

```
## ==Google Cloud Storage Bucket==
## Bucket:          chest-xray-medical
## Project Number:  1048020973776
## Location:        NORTHAMERICA-NORTHEAST1
## Class:           STANDARD
## Created:         2020-11-12 07:22:05
## Updated:         2020-11-19 20:59:24
## Meta-generation: 3
## eTag:            CAM=
```

```
gcs_global_bucket("chest-xray-medical")
```

```
## Set default bucket name to 'chest-xray-medical'
```

```
medical_objects <- gcs_list_objects() %>% mutate(id= row_number())
```

This is where we are going to actually copy all the images that are stored in the Google Cloud Storage Bucket into a space that can be used for this project.

```
gs_rsync(source="gs://chest-xray-medical", destination="images", recursive=TRUE )
```