DATA WRANGLING, DATA ANALYSIS & VISUALIZATION

-UNDER GUIDANCE OF PROF. SRI KRISHNAMURTHY

COMPILED BY:

MOHIT MITTAL,

SNEHA RAVIKUMAR,

TAJ POOVAIAH

TABLE OF CONTENTS

OBJECTIVE

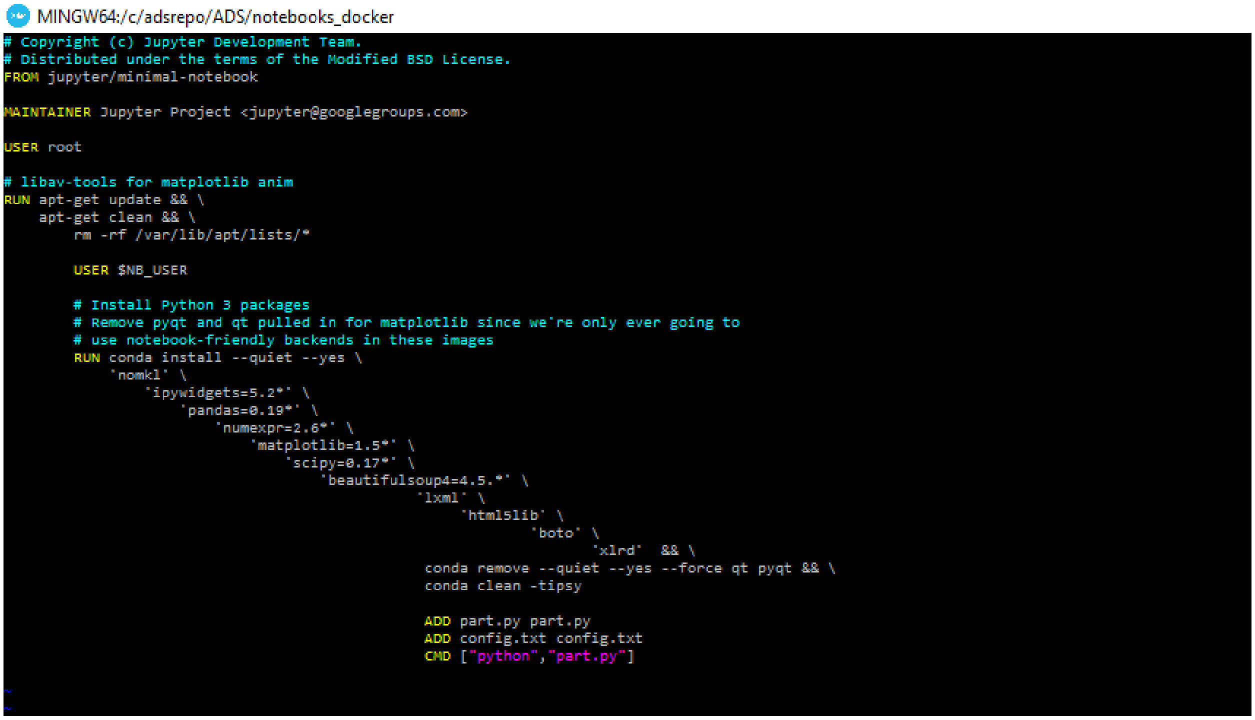
The objective of our assignment is to gather, retrieve, clean, and analyze the data using python code, dockerize the pipeline so as to upload our end results onto Amazon S3 buckets. The use of configuration file and Docker image makes it easy for flexible usage.

INTRODUCTION TO DOCKER

Docker is an open-source project that automates the deployment of applications inside software containers. Quote of features from Docker web pages: “Docker containers wrap up a piece of software in a complete filesystem that contains everything it needs to run: code, runtime, system tools, and system libraries – anything you can install on a server. This guarantees that it will always run the same, regardless of the environment it is running in.”

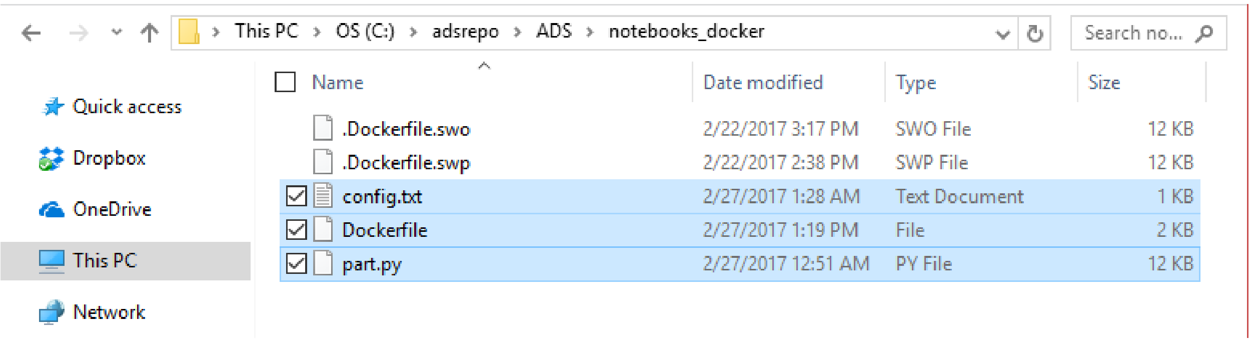
DOCKER DESIGN AND IMPLEMENTATION

Docker file is used to specify the configurations and settings required for the application to run:



```
# Copyright (c) Jupyter Development Team.
# Distributed under the terms of the Modified BSD License.
FROM jupyter/minimal-notebook

MAINTAINER Jupyter Project <jupyter@googlegroups.com>

USER root

# libav-tools for matplotlib anim
RUN apt-get update && \
    apt-get clean && \
        rm -rf /var/lib/apt/lists/*

        USER $NB_USER

        # Install Python 3 packages
        # Remove pyqt and qt pulled in for matplotlib since we're only ever going to
        # use notebook-friendly backends in these images
        RUN conda install --quiet --yes \
            'nomkl' \
                'ipywidgets=5.2*' \
                    'pandas=0.19*' \
                        'numexpr=2.6*' \
                            'matplotlib=1.5*' \
                                'scipy=0.17*' \
                                    'beautifulsoup4=4.5.*' \
                                               'lxml' \
                                                    'html5lib' \
                                                            'boto' \
                                                                   'xlrd'  && \
                                                conda remove --quiet --yes --force qt pyqt && \
                                                conda clean -tipsy

                                                ADD part.py part.py
                                                ADD config.txt config.txt
                                                CMD ["python","part.py"]

~
~
```

The Docker Directory contains the files required to run the application



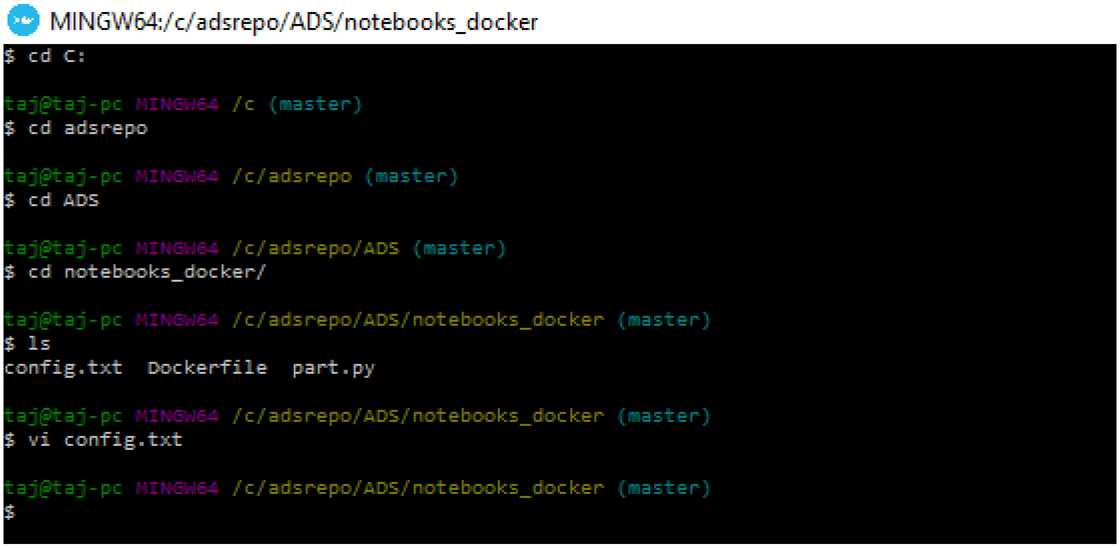
Name	Date modified	Type	Size
.Dockerfile.swo	2/22/2017 3:17 PM	SWO File	12 KB
.Dockerfile.swp	2/22/2017 2:38 PM	SWP File	12 KB
☑ config.txt	2/27/2017 1:28 AM	Text Document	1 KB
☑ Dockerfile	2/27/2017 1:19 PM	File	2 KB
☑ part.py	2/27/2017 12:51 AM	PY File	12 KB

PULLING A DOCKER IMAGE AND RUNNING IT ON YOUR MACHINE

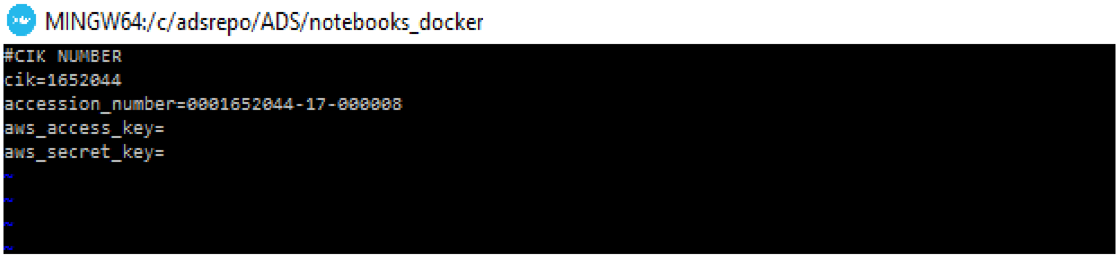
Open the Docker Quickstart terminal on your machine to get started by running the commands on Docker

1. Run **$docker pull "repo name"** to pull the image from any repo onto your system

2. Run **$ls** and then **$vi config.txt** to Update the configuration file with the values that you want to test (CIK, accession number, AWS access key, AWS secret key)



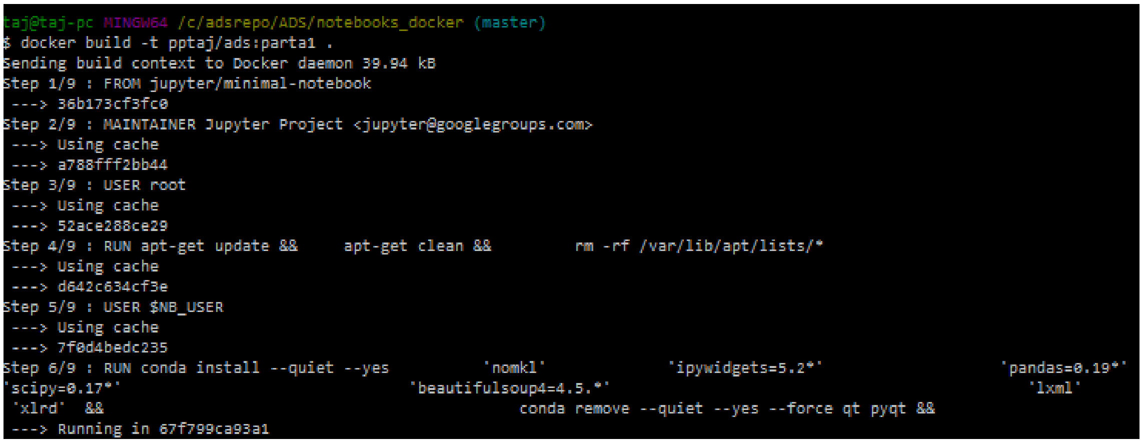
```
$ cd C:

taj@taj-pc MINGW64 /c (master)
$ cd adsrepo

taj@taj-pc MINGW64 /c/adsrepo (master)
$ cd ADS

taj@taj-pc MINGW64 /c/adsrepo/ADS (master)
$ cd notebooks_docker/

taj@taj-pc MINGW64 /c/adsrepo/ADS/notebooks_docker (master)
$ ls
config.txt  Dockerfile  part.py

taj@taj-pc MINGW64 /c/adsrepo/ADS/notebooks_docker (master)
$ vi config.txt

taj@taj-pc MINGW64 /c/adsrepo/ADS/notebooks_docker (master)
$
```



MINGW64:/c/adsrepo/ADS/notebooks_docker

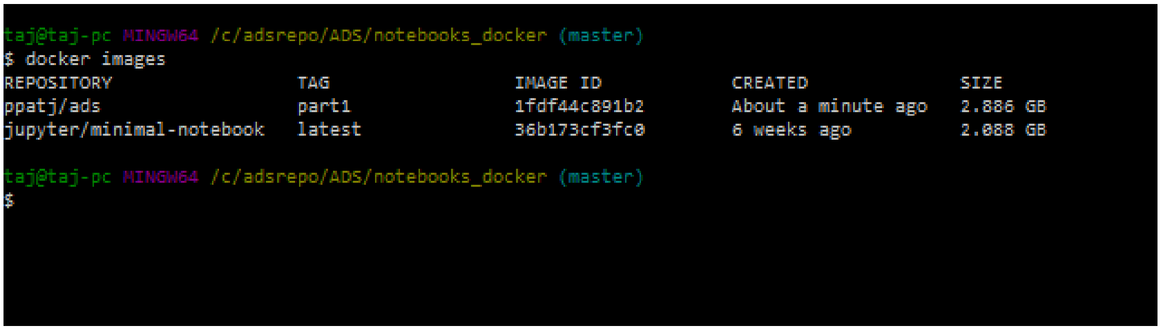
```
#CIK NUMBER
cik=1652044
accession_number=0001652044-17-000008
aws_access_key=
aws_secret_key=
~
~
~
~
```

3. Run **$docker build –t “repo name:tag name”** . to build the image that you pulled



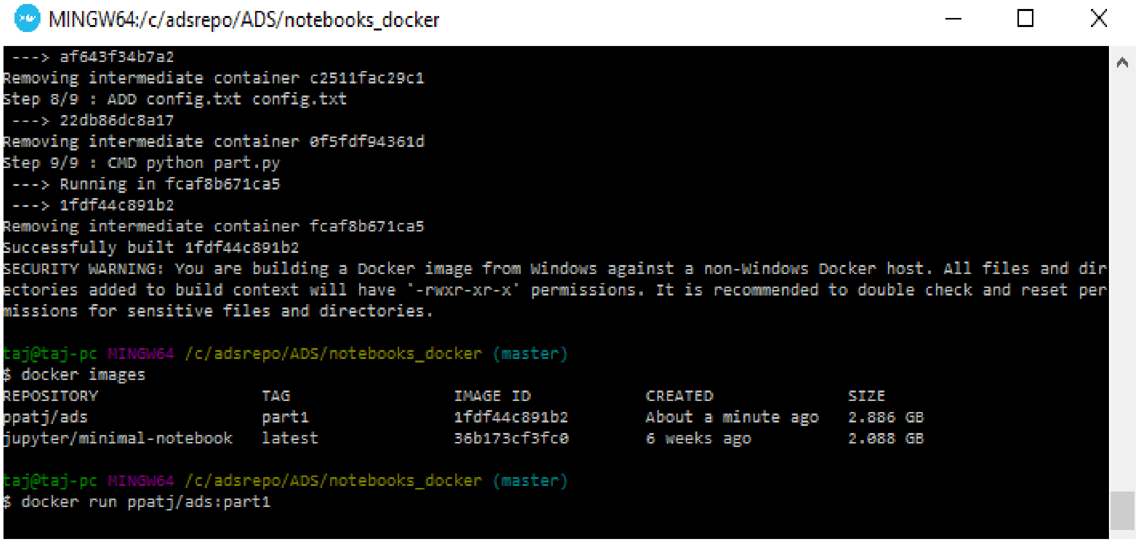
```
taj@taj-pc MINGW64 /c/adsrepo/ADS/notebooks_docker (master)
$ docker build -t pptaj/ads:parta1 .
Sending build context to Docker daemon 39.94 kB
Step 1/9 : FROM jupyter/minimal-notebook
 ---> 36b173cf3fc0
Step 2/9 : MAINTAINER Jupyter Project <jupyter@googlegroups.com>
 ---> Using cache
 ---> a788fff2bb44
Step 3/9 : USER root
 ---> Using cache
 ---> 52ace288ce29
Step 4/9 : RUN apt-get update &&     apt-get clean &&         rm -rf /var/lib/apt/lists/*
 ---> Using cache
 ---> d642c634cf3e
Step 5/9 : USER $NB_USER
 ---> Using cache
 ---> 7f0d4bedc235
Step 6/9 : RUN conda install --quiet --yes          'nomkl'             'ipywidgets=5.2*'                  'pandas=0.19*'
'scipy=0.17*'                               'beautifulsoup4=4.5.*'                                          'lxml'
 'xlrd'  &&                                             conda remove --quiet --yes --force qt pyqt &&
 ---> Running in 67f799ca93a1
```

4. Run **$docker images** to check the list of docker images in the system



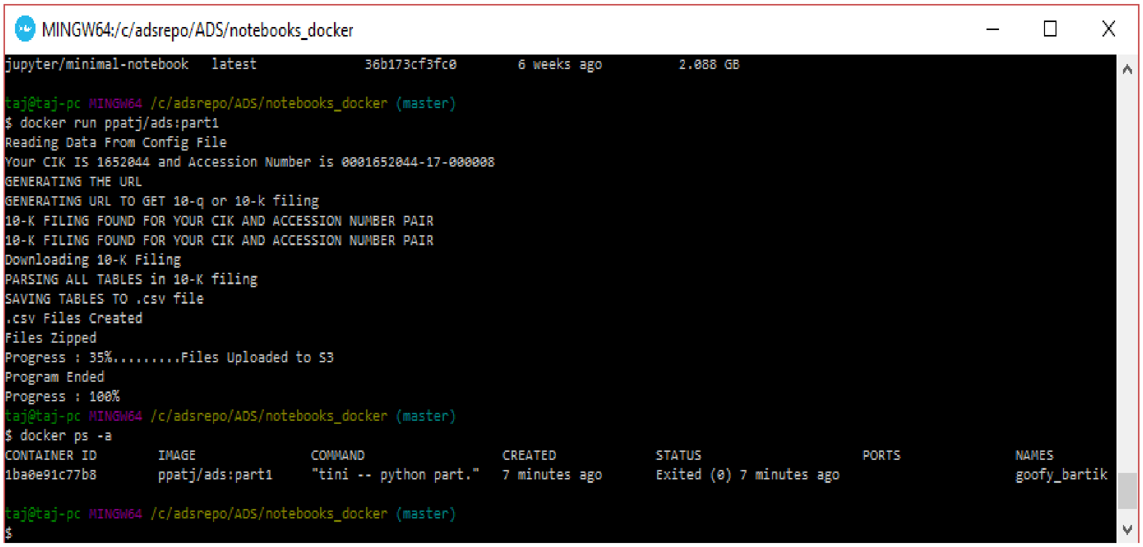
```
taj@taj-pc MINGW64 /c/adsrepo/ADS/notebooks_docker (master)
$ docker images
REPOSITORY                 TAG                 IMAGE ID            CREATED              SIZE
ppatj/ads                  part1               1fdf44c891b2        About a minute ago   2.886 GB
jupyter/minimal-notebook   latest              36b173cf3fc0        6 weeks ago          2.088 GB

taj@taj-pc MINGW64 /c/adsrepo/ADS/notebooks_docker (master)
$
```

5. Run **$docker run “repo name:tag name”** to run the application on Docker



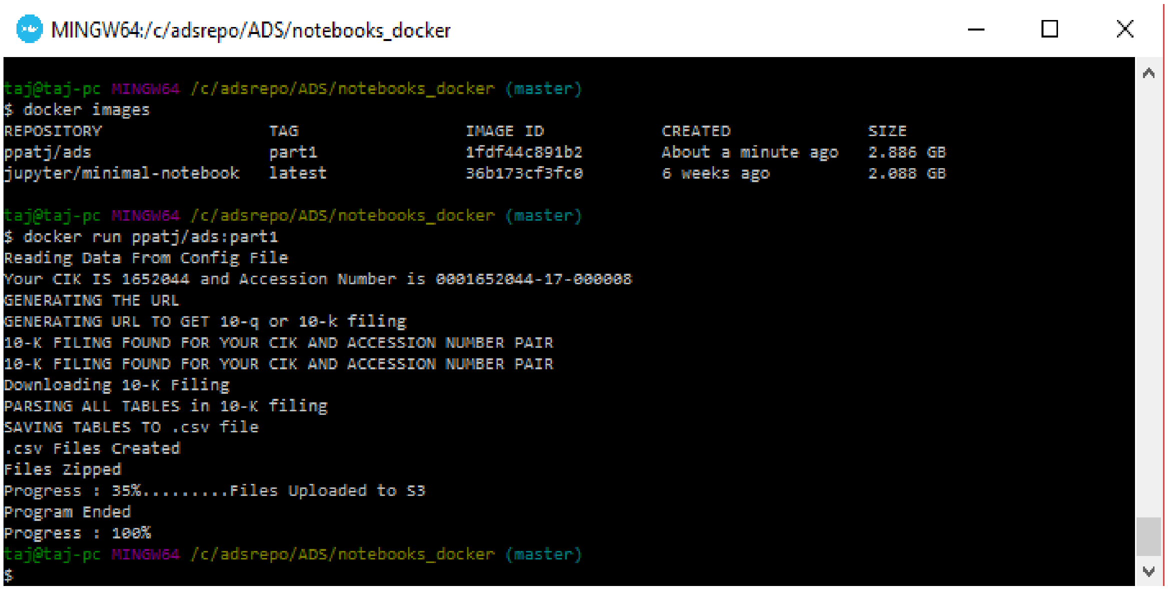
```
 ---> af643f34b7a2
Removing intermediate container c2511fac29c1
Step 8/9 : ADD config.txt config.txt
 ---> 22db86dc8a17
Removing intermediate container 0f5fdf94361d
Step 9/9 : CMD python part.py
 ---> Running in fcaf8b671ca5
 ---> 1fdf44c891b2
Removing intermediate container fcaf8b671ca5
Successfully built 1fdf44c891b2
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and dir
ectories added to build context will have '-rwxr-xr-x' permissions. It is recommended to double check and reset per
missions for sensitive files and directories.

taj@taj-pc MINGW64 /c/adsrepo/ADS/notebooks_docker (master)
$ docker images
REPOSITORY                 TAG                 IMAGE ID            CREATED              SIZE
ppatj/ads                  part1               1fdf44c891b2        About a minute ago   2.886 GB
jupyter/minimal-notebook   latest              36b173cf3fc0        6 weeks ago          2.088 GB

taj@taj-pc MINGW64 /c/adsrepo/ADS/notebooks_docker (master)
$ docker run ppatj/ads:part1
```

6. Run **$docker ps -a** to check the list of docker images in the system



```
jupyter/minimal-notebook   latest              36b173cf3fc0        6 weeks ago         2.088 GB

taj@taj-pc MINGW64 /c/adsrepo/ADS/notebooks_docker (master)
$ docker run ppatj/ads:part1
Reading Data From Config File
Your CIK IS 1652044 and Accession Number is 0001652044-17-000008
GENERATING THE URL
GENERATING URL TO GET 10-q or 10-k filing
10-K FILING FOUND FOR YOUR CIK AND ACCESSION NUMBER PAIR
10-K FILING FOUND FOR YOUR CIK AND ACCESSION NUMBER PAIR
Downloading 10-K Filing
PARSING ALL TABLES in 10-K filing
SAVING TABLES TO .csv file
.csv Files Created
Files Zipped
Progress : 35%.........Files Uploaded to S3
Program Ended
Progress : 100%
taj@taj-pc MINGW64 /c/adsrepo/ADS/notebooks_docker (master)
$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                     PORTS               NAMES
1ba0e91c77b8        ppatj/ads:part1     "tini -- python part."   7 minutes ago       Exited (0) 7 minutes ago                       goofy_bartik

taj@taj-pc MINGW64 /c/adsrepo/ADS/notebooks_docker (master)
$
```

The status of the program being executed will be updated on the terminal as shown below



```
taj@taj-pc MINGW64 /c/adsrepo/ADS/notebooks_docker (master)
$ docker images
REPOSITORY                 TAG                 IMAGE ID            CREATED              SIZE
ppatj/ads                  part1               1fdf44c891b2        About a minute ago   2.886 GB
jupyter/minimal-notebook   latest              36b173cf3fc0        6 weeks ago          2.088 GB

taj@taj-pc MINGW64 /c/adsrepo/ADS/notebooks_docker (master)
$ docker run ppatj/ads:part1
Reading Data From Config File
Your CIK IS 1652044 and Accession Number is 0001652044-17-000008
GENERATING THE URL
GENERATING URL TO GET 10-q or 10-k filing
10-K FILING FOUND FOR YOUR CIK AND ACCESSION NUMBER PAIR
10-K FILING FOUND FOR YOUR CIK AND ACCESSION NUMBER PAIR
Downloading 10-K Filing
PARSING ALL TABLES in 10-K filing
SAVING TABLES TO .csv file
.csv Files Created
Files Zipped
Progress : 35%.........Files Uploaded to S3
Program Ended
Progress : 100%
taj@taj-pc MINGW64 /c/adsrepo/ADS/notebooks_docker (master)
$
```

TABLEAU DESIGN AND IMPLEMENTATION

WHY TABLEAU IS USED IN THIS PROJECT

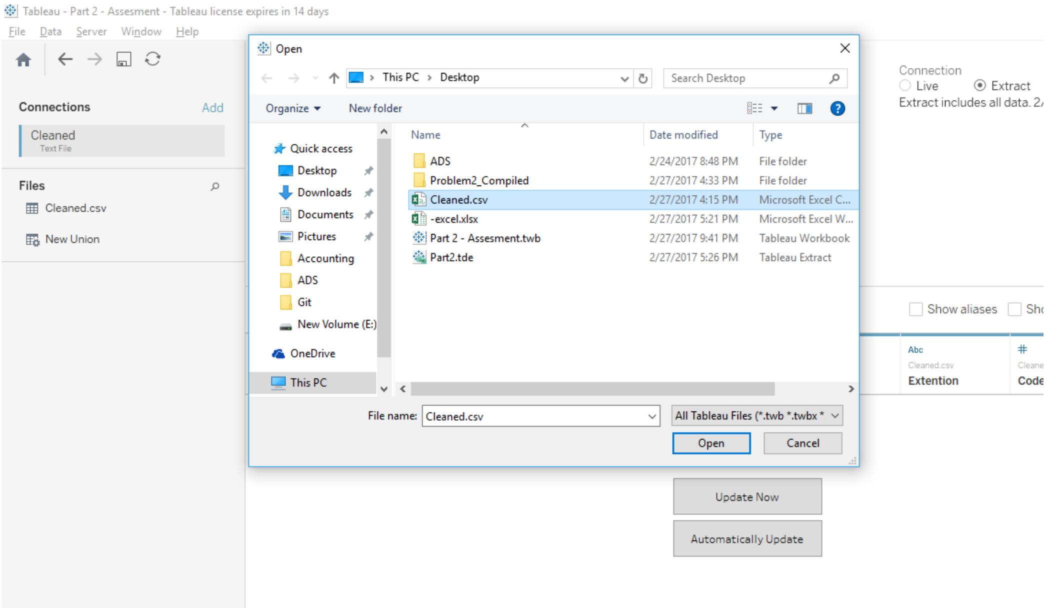
Once the data sets are gathered, cleaned, summarized, and analyzed, tableau is used to show the visual representation of the summary metrics to make it easier to understand and interpret

Tableau can help anyone see and understand their data. Connect to almost any database, drag and drop to create visualizations, and share with a click.

GETTING STARTED WITH TABLEAU

Import the file that contains the data you want to analyze on Tableau

Data Source -> File -> Open -> Cleaned.csv



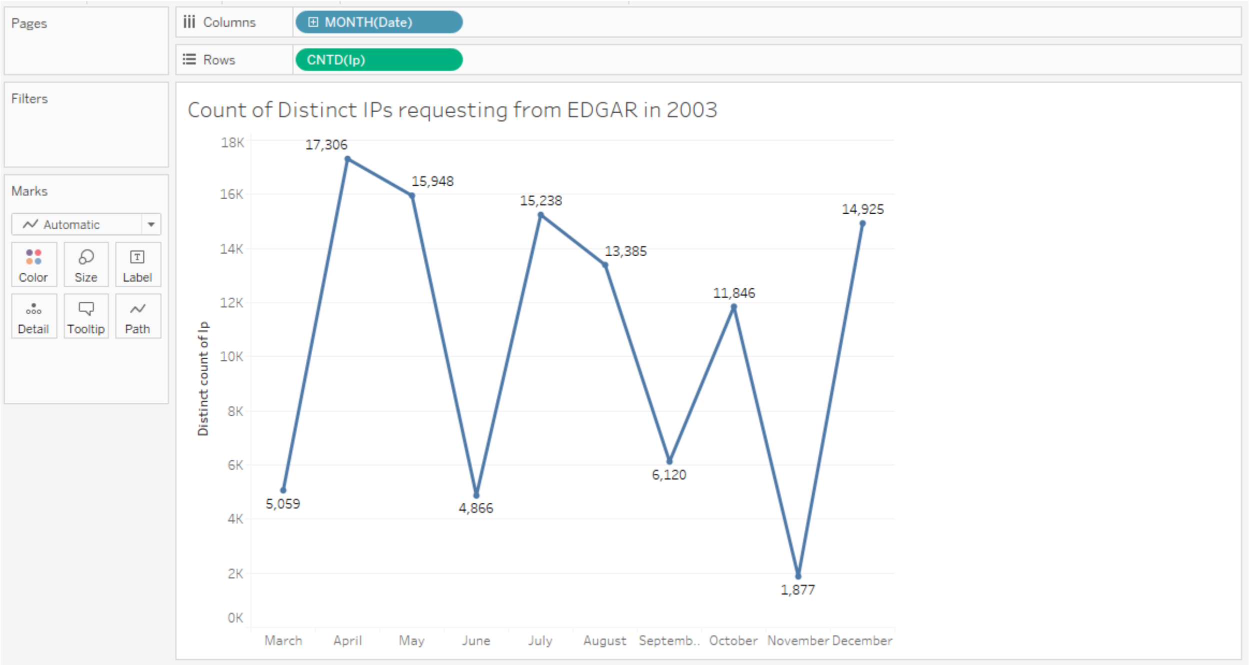
Name	Date modified	Type
ADS	2/24/2017 8:48 PM	File folder
Problem2_Compiled	2/27/2017 4:33 PM	File folder
Cleaned.csv	2/27/2017 4:15 PM	Microsoft Excel C...
~excel.xlsx	2/27/2017 5:21 PM	Microsoft Excel W...
Part 2 - Assesment.twb	2/27/2017 9:41 PM	Tableau Workbook
Part2.tde	2/27/2017 5:26 PM	Tableau Extract

File name: Cleaned.csv
All Tableau Files (*.twb *.twbx *

Connection
○ Live ● Extract
Extract includes all data. 2/

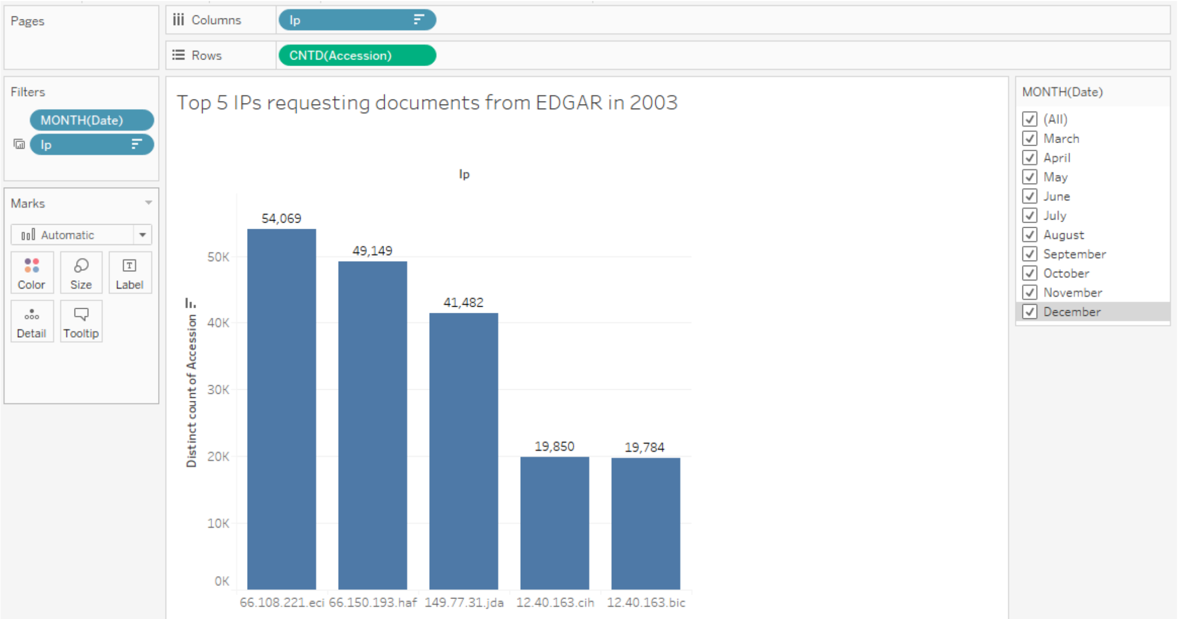
ANALYSIS AND REVIEW OF TABLEAU VIZUALIZATIONS

1. How many times distinct machines accessed data on EDGAR?



Its observed that over the nine months' data we have, April observes the most traffic with over seventeen thousand visitors and November with a low of only over thousand.

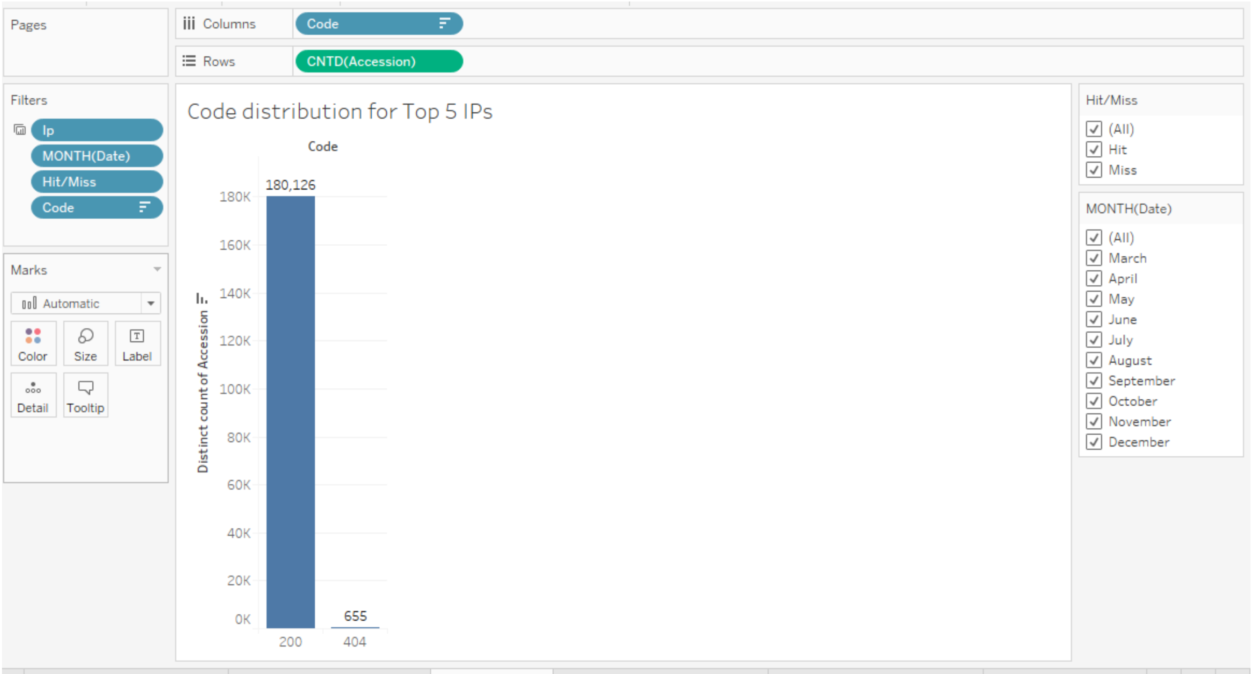
2. What are the popular IPs accessing data?



Of the thousands of IPs hitting the website, we understand that these top 5 IPs keep coming back across time and access files most frequently.

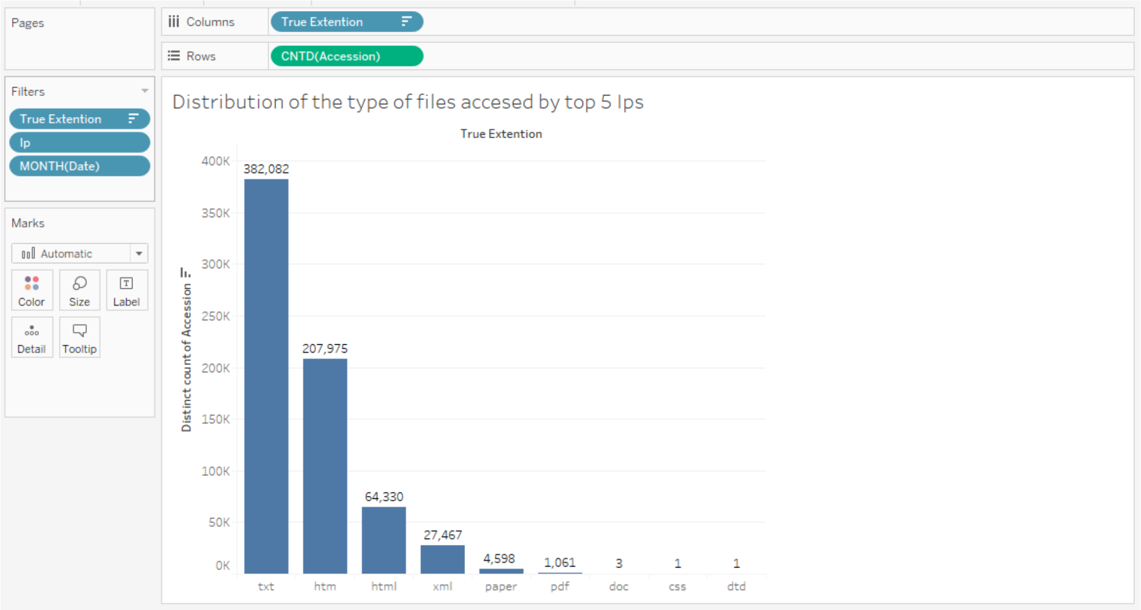
The IP 66.108.221.eci accesses 54,069 accession files over the 12 months in 2003.

3. For the requests made by the IPs, what is the hit to miss ratio?



We observe that most of the requests pass with an ‘OK’ (200) code.

4. What type of files are being accessed?



Columns: True Extention

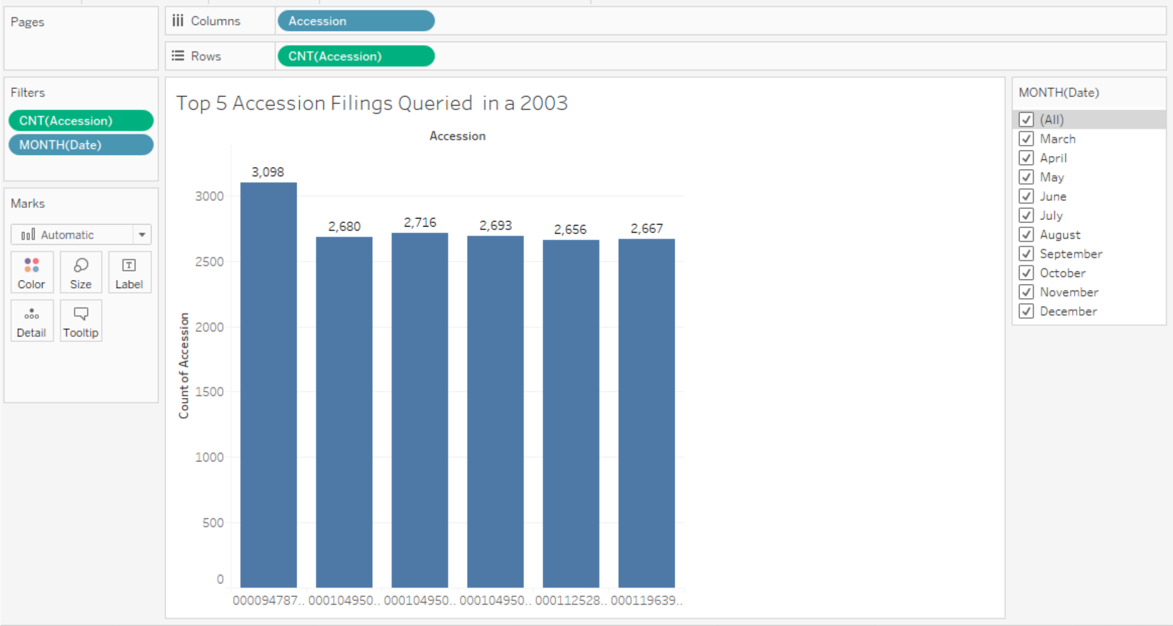
Rows: CNTD(Accession)

Filters: True Extention, Ip, MONTH(Date)

Marks: Automatic

We observe from the above distribution that most of the files accessed are .txt files.

5. What are the most frequently accessed accession numbers?



Columns: Accession

Rows: CNT(Accession)

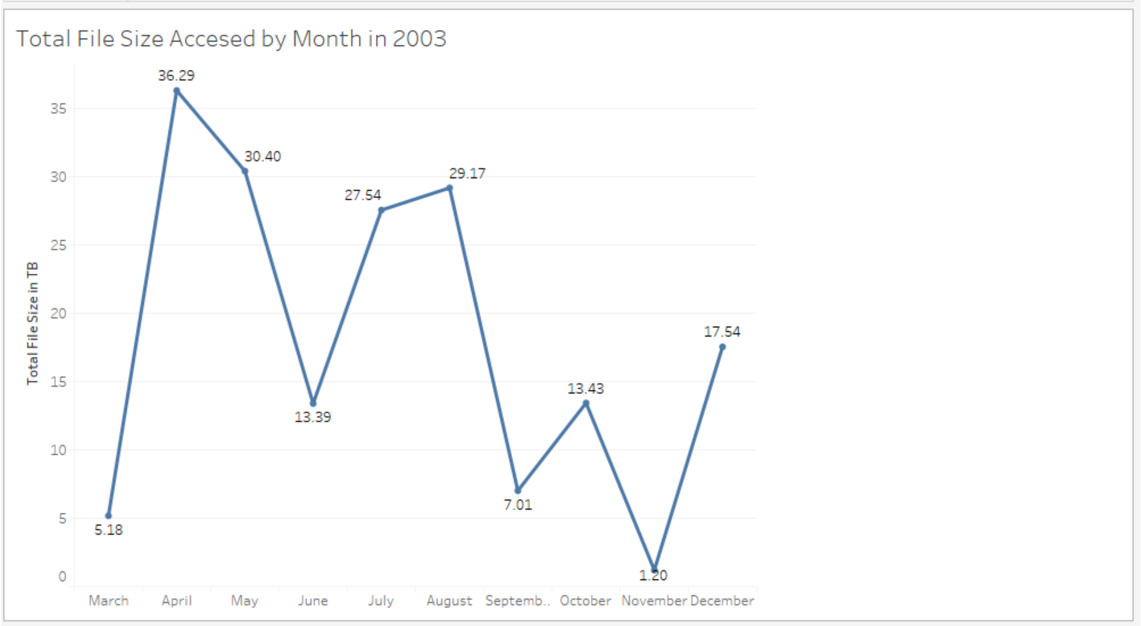
Filters: CNT(Accession), MONTH(Date)

Marks: Automatic

MONTH(Date)
☑ (All)
☑ March
☑ April
☑ May
☑ June
☑ July
☑ August
☑ September
☑ October
☑ November
☑ December

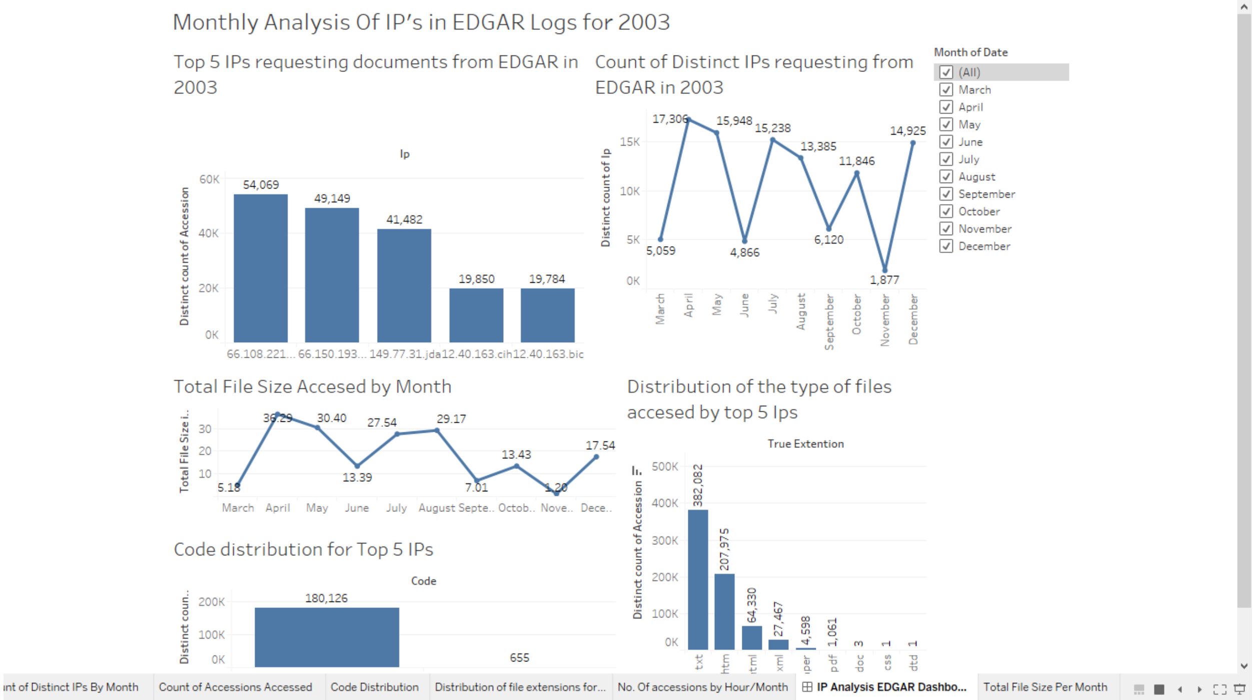
We plot the accession numbers against how many times they occur over 12 months to understand the most frequently requested accession number. It can be filtered for each month.

6. We understand there are multiple kinds of files accessed. What is the total file size of data downloaded over months?


Total File Size Accesed by Month in 2003

As observed, since April has the highest traffic, It also has the highest file size downloaded with a high of 36.29 TB

DASHBOARD


Monthly Analysis Of IP’s in EDGAR Logs for 2003

Top 5 IPs requesting documents from EDGAR in 2003

Count of Distinct IPs requesting from EDGAR in 2003

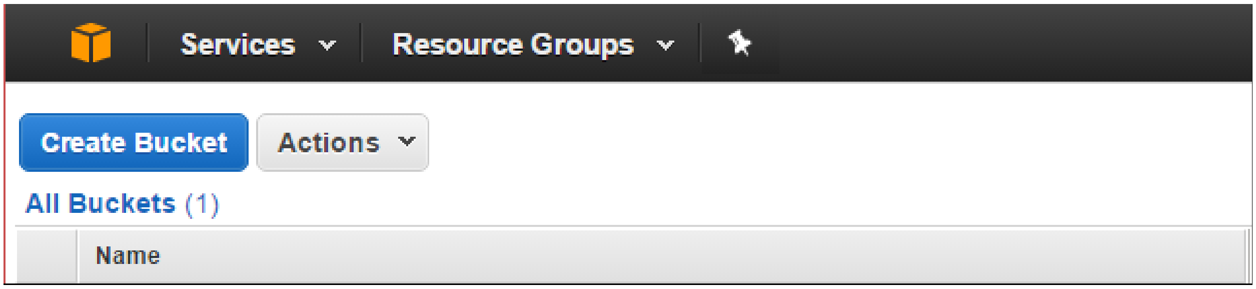
Total File Size Accesed by Month

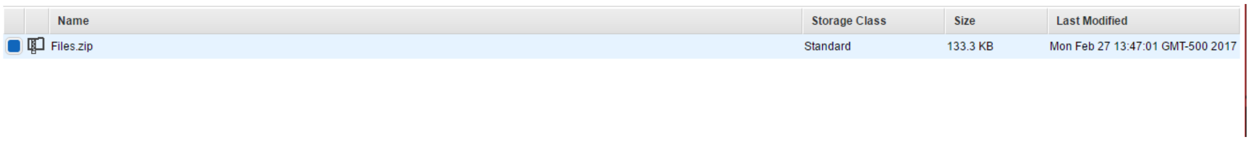
Distribution of the type of files accesed by top 5 Ips

Code distribution for Top 5 IPs
Code
Distinct coun..
0K
100K
200K
180,126
655
int of Distinct IPs By Month
Count of Accessions Accessed
Code Distribution
Distribution of file extensions for...
No. Of accessions by Hour/Month
IP Analysis EDGAR Dashbo...
Total File Size Per Month

AWS S3 DATA ANALYSIS

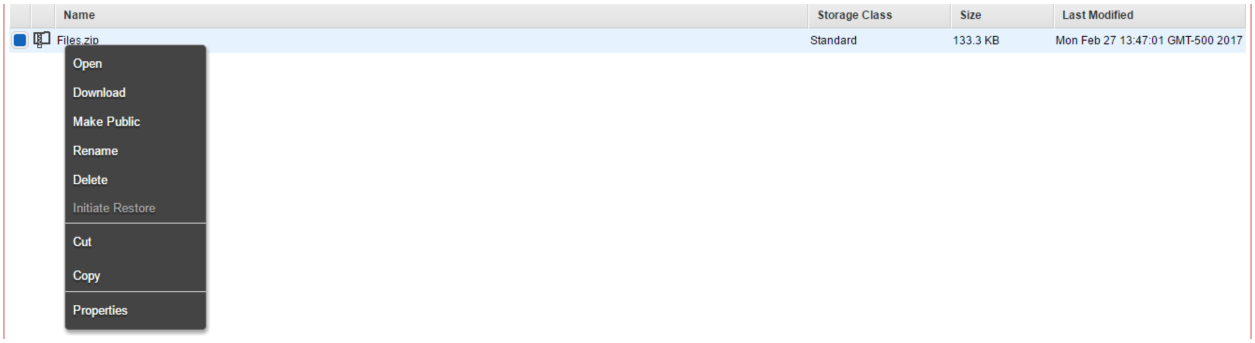
1. Log in to your AWS account, select S3, and check if a bucket with you aws access key has been created. Check if the Zip file has been uploaded onto the bucket





Name	Storage Class	Size	Last Modified
Files.zip	Standard	133.3 KB	Mon Feb 27 13:47:01 GMT-500 2017

2. Download the zip folder and check the contents



Name	Storage Class	Size	Last Modified
Files.zip	Standard	133.3 KB	Mon Feb 27 13:47:01 GMT-500 2017

REVIEW OF OUTPUTS STORED ON AMAZON S3

PART1 RESULT OUTPUT

The results in the Amazon S3 bucket consists of a zip file that has all of the csvs (the csvs have the special characters removed while scarping data) of either 10K/10Q documents of a particular company recognized based on the CIK and accession number provided by the user in the configuration file.

PART2 RESULT OUTPUT

The results in the Amazon S3 bucket consists of a zip file. The zip file consists of the final result which is obtained by:

- The files for a particular year is downloaded into a folder
- The files are then cleansed, and put into another folder
- Summary metrics functions are performed on every cleaned csv for every month
- The cleaned files are compiled into a single csv
- The compilation of the summary for all the 12 months are then added onto one summary file and then into another folder
- All of these folders, are then zipped, and uploaded

CLEANING THE DATA

On individual cleaned CSVs, we performed the following steps to get cleaned CSV files:

- Added columns by splitting Date and Time fields
- Removed all the rows with Error Code = 0 (Non –existent) - outlier removal
- Obtained a unique column showing the extension for each file
- Removed junk extensions (.hd, .hmtl etc.)
- Replaced all the un-available data with 'NA'

EXCEPTION HANDLING- PART1

- Program handles the error if no CIK or Accession Number is provided
- Program handles the error if the CIK and Accession number is valid, if not, throws error
- Program handles the error for 10k/10q documents. If absent, throws error
- Program validates links provided, handles the error if link is invalid

EXCEPTION HANDLING- PART2

- Program handles the error if year is empty
- Program handles the error if year is invalid (eg:3, 44)
- Program handles the error if the year provided does not exist
  (eg: not 2003 <= year =< 2016)

INVALID AMAZON KEY EXCEPTION HANDLING

- Program Checks if Amazon keys (access and secret) provided by user is empty, throws error if it is
- Program checks if the keys are invalid. If the keys are not valid, the connection to S3 fails and throws an error as soon as it reads the keys from the configuration file
- Program checks if bucket exists. If it does, it adds the files into the same bucket
- Program replaces the files that are uploaded with the most recent upload if the file name is the same in the bucket

IF DATA IS NOT FOUND

- If Data (10Q/10K documents) are not found, program handles it by prompting that the link provided doesn’t contain 10K/10Q documents

END OF REPORT