ADVANCED DATA SCIENCE

Assignment 1

Edgar datasets: Wrangling, Pre-processing and exploratory data analysis

Prof. Sri Krishnamurthy

Team Members:
Sayali Borse
Rishi Rajani
Komal Ambekar

Table of Contents:

Libraries:

1. Urllib: This is used for webscrapping
2. Urllib.request: This module is used for defining functions and classes for opening URLs
3. Zipfile: This module is used to manipulate zipfiles
4. Beautifulsoup: Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree.
5. Csv: This is used to write and read the csv files
6. Logging: This module is used to track the logging details using the logger functions
7. Boto3: For handling Amazon Web Services S3
8. Pandas: This is used for data manipulation
9. Shutil: The **shutil** module offers a number of high-level operations on files and collections of files. In particular, functions are provided which support file copying and removal.
10. Glob: The **glob** module finds all the pathnames matching a specified pattern according to the rules used by the Unix shell, although results are returned in arbitrary order.
11. Os: This module provides a portable way of using operating system dependent functionality. If you just want to read or write a file see **open()**, if you want to manipulate paths

Logger:

Logging is a means of tracking events. An event is described by a descriptive message which can optionally contain variable data. We are generating log files at every stage of the code, we are generating timestamps, level names and customized error message.

We have use 4 levels of Logs:

1. DEBUG: It is used to return information to the log file that is created. Report events that occur during normal operation of a program
2. Error: Due to a more serious problem, the software has not been able to perform some function.
3. Info: Confirmation that things are working as expected.

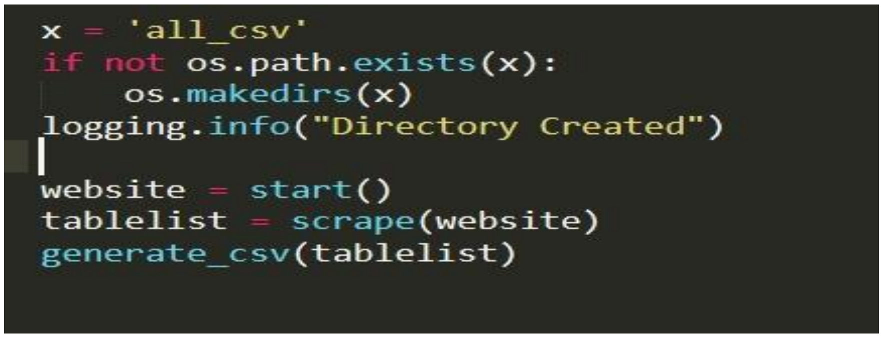
Problem 1: Data Wrangling and EDA for Edgar Data set

Tasks and Requirements:

- The objective of this problem is to extract tables from 10Q fillings from https://www.sec.gov/edgar/
- Inputs given: CIK number, Accession code, Amazon access key and Amazon secret access key
- We are using urllib and urllib.request libraries to navigate and generate page specific URL.
- We are using BeautifulSoup libraries to handle html tags in Python and also to prettify the data content.
- Once URL is opened using the urllib.request.get() we have a foreach loop to run through all the <a> tags and <href> tags of the html page. If there is no 10-Q file the program exists.
- Once we get and open the 10-Q url link we extract the data by looking for table content which are part of the <div>tag.
- We clean the fetched data as it contains special characters like '$' and '%' values inside the table data. So we have to iterate through the tables and look for patterns.
- After data is cleaned every table is stored as individual csv file which is inside extracted_csv folder.
- The program zips the folder and puts it inside Problem_1.zip.
- We create a bucket and upload a zip file. The bucket name is always unique.

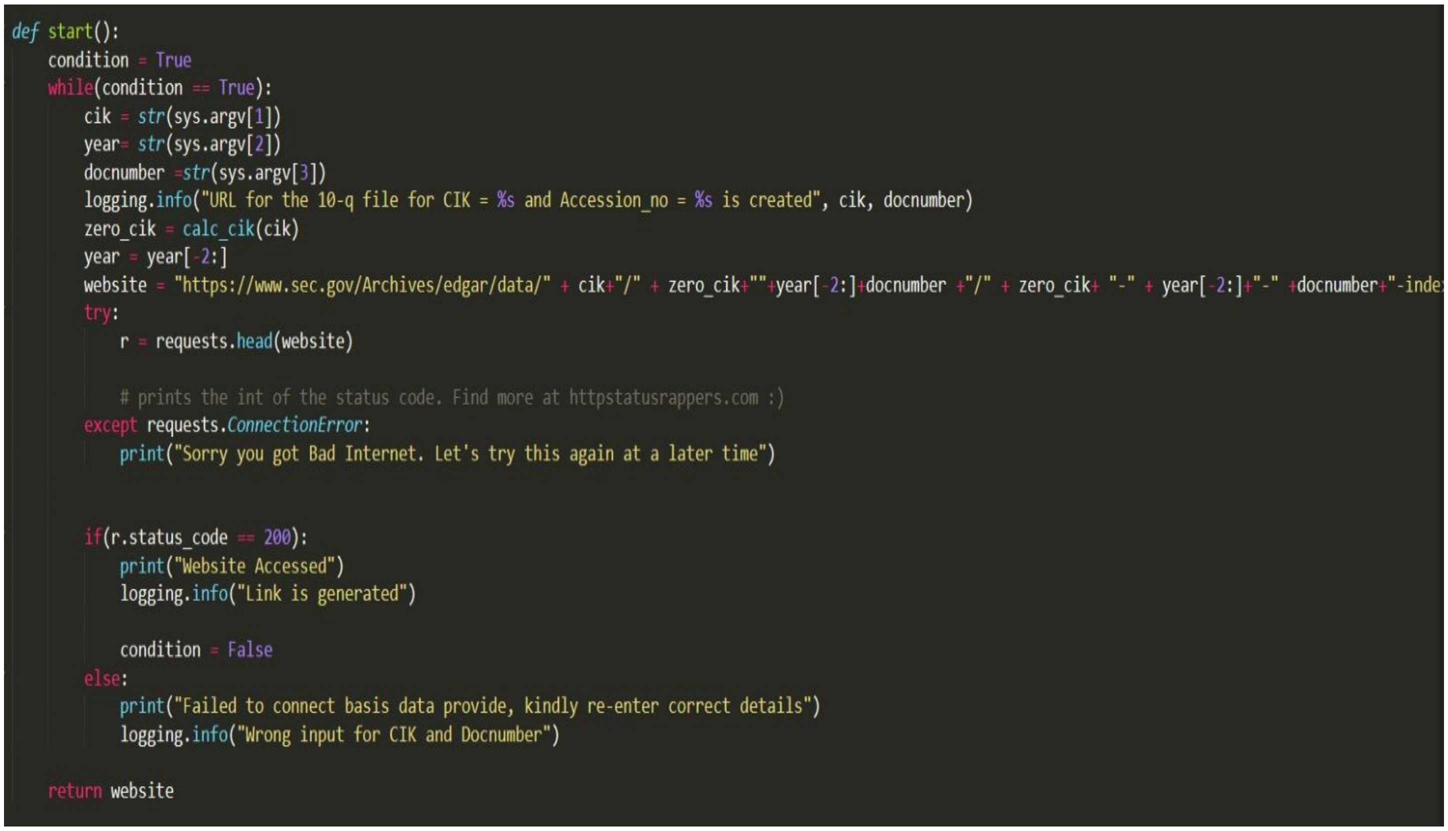
Screenshots:

Eventually the entire code from our file comes down to running the below three functions. This code snippet is used to create the directory and scrape the website and generate the csv tables.



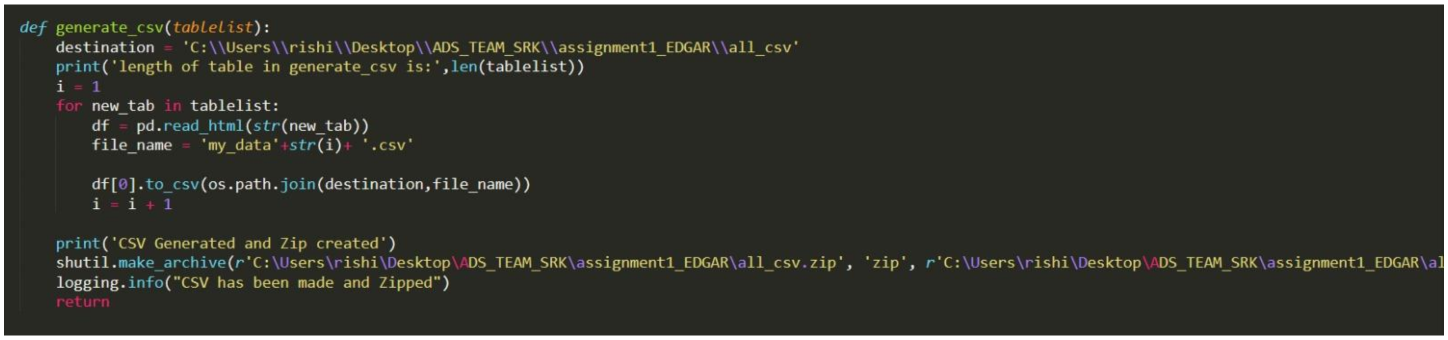
```
x = 'all_csv'
if not os.path.exists(x):
    os.makedirs(x)
logging.info("Directory Created")

website = start()
tablelist = scrape(website)
generate_csv(tablelist)
```

The below code snippet is used to take the CIK number and Accession number as input from the user which then generates the URL and accesses the website from which the tables should be extracted from 10Q fillings.



```
def start():
    condition = True
    while(condition == True):
        cik = str(sys.argv[1])
        year= str(sys.argv[2])
        docnumber =str(sys.argv[3])
        logging.info("URL for the 10-q file for CIK = %s and Accession_no = %s is created", cik, docnumber)
        zero_cik = calc_cik(cik)
        year = year[-2:]
        website = "https://www.sec.gov/Archives/edgar/data/" + cik+"/" + zero_cik+""+year[-2:]+docnumber +"/" + zero_cik+ "-" + year[-2:]+"-" +docnumber+"-inde
        try:
            r = requests.head(website)

            # prints the int of the status code. Find more at httpstatusrappers.com :)
        except requests.ConnectionError:
            print("Sorry you got Bad Internet. Let's try this again at a later time")


        if(r.status_code == 200):
            print("Website Accessed")
            logging.info("Link is generated")

            condition = False
        else:
            print("Failed to connect basis data provide, kindly re-enter correct details")
            logging.info("Wrong input for CIK and Docnumber")

    return website
```

After the URL is generated and re-directed to the website. The tables from 10Q filings will be extracted and the csv files will be generated and zipped. 105 tables are extracted.



```
def generate_csv(tablelist):
    destination = 'C:\\Users\\rishi\\Desktop\\ADS_TEAM_SRK\\assignment1_EDGAR\\all_csv'
    print('length of table in generate_csv is:',len(tablelist))
    i = 1
    for new_tab in tablelist:
        df = pd.read_html(str(new_tab))
        file_name = 'my_data'+str(i)+ '.csv'

        df[0].to_csv(os.path.join(destination,file_name))
        i = i + 1

    print('CSV Generated and Zip created')
    shutil.make_archive(r'C:\Users\rishi\Desktop\ADS_TEAM_SRK\assignment1_EDGAR\all_csv.zip', 'zip', r'C:\Users\rishi\Desktop\ADS_TEAM_SRK\assignment1_EDGAR\al
    logging.info("CSV has been made and Zipped")
    return
```

Problem 2: Missing Data Analysis:

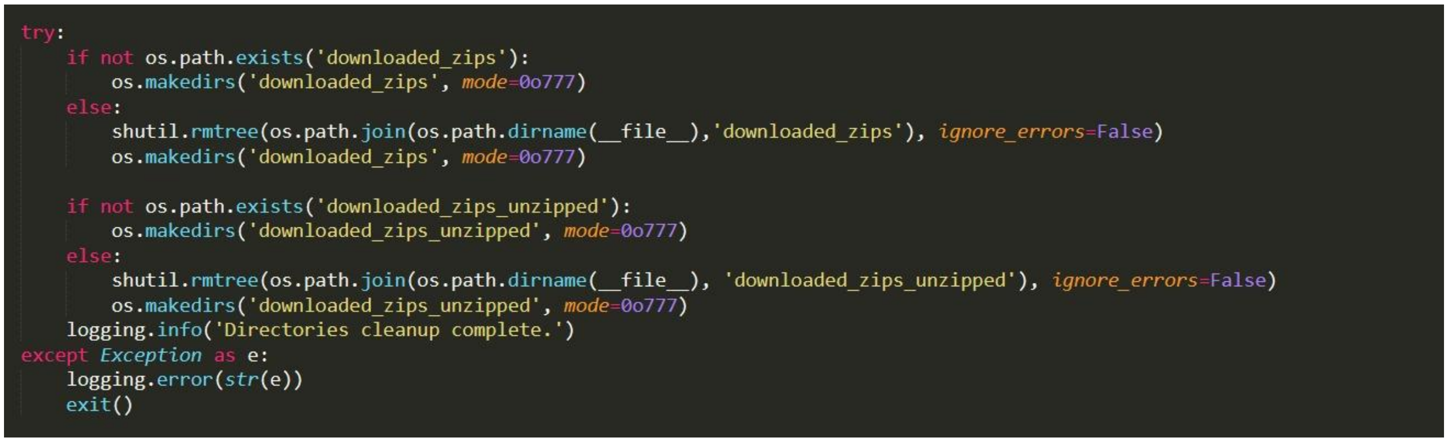
Tasks and Requirements:

- Import all the libraries for the code and also initialize the log file as done in the problem 1.
- If the directories are not present create one to put the zip files and then clean the required directories.
- The input year must be between 2003 to 2018. If the year is not between this range the program will exit. There is no data for the first quarter of 2003.
- To the extract the csv, unzip all the downloaded log files.
- Load all the csv files into individual dataframe and combine into one csv file.
- Summary Metrics are as follows:
    - Compute mean size
    - Compute maximum used browser
    - Compute distinct count of ip per month i.e. per log file
    - Compute distinct count of cik per month i.e per log file
- Provide logging.info to all the variables and check for anomalies in the data and remove the row if found.
- Combine all dataframes and compute overall summary metric. Now export it to a csv.
- Create a bucket on S3 through the code and upload both the files.
- For each log file we have handled the missing value as the following:
    - The variables cik, ip, accession, data or time are most important for the log file in Edgar and we cannot dp correct analysis without these values.So skip the row if they consist null value.
    - Replacement for NaN according to the properties of variables.

VARIABLES	DATA	REPLACED WITH
noagent	NaN	1
code	NaN	Most used code
browser	NaN	Most used browser
idx	NaN	Most used idx
norefer	NaN	1
find	NaN	Most used find
crawler	NaN	0(assume it as empty)
extension	NaN	Most used extension
zone	NaN	Most used zone
size	NaN	Mean of the size

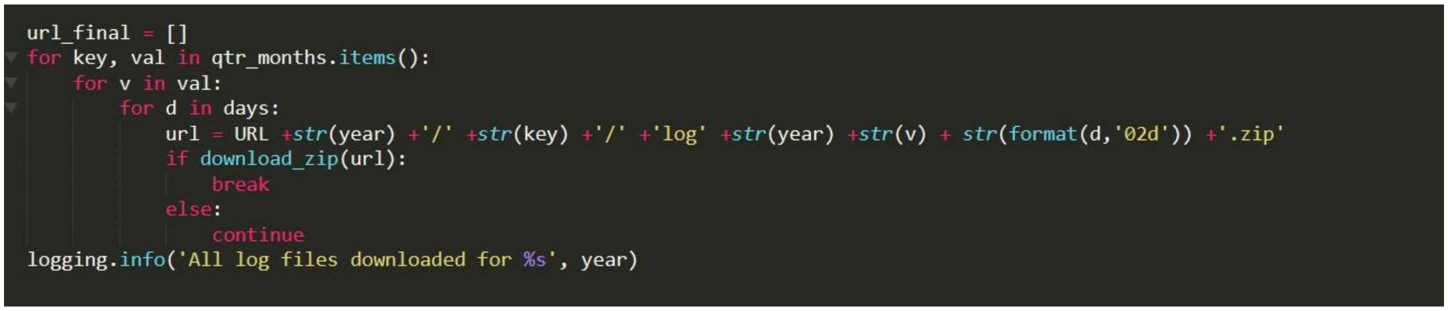
Screenshots:

This try block checks if the zip files exist already in the folder. If the zipped folder exists then shutil.rmtree will remove the folder. If the zip folder do not exist then the downloaded zip folder will be unzipped. This is the directory cleanup.



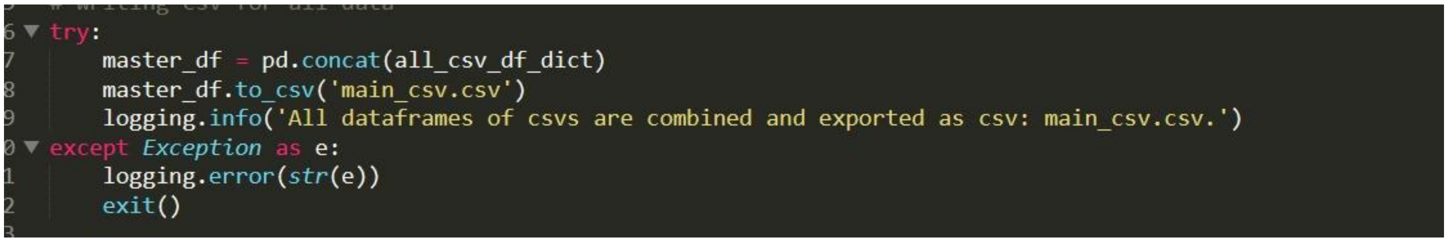
```python
try:
    if not os.path.exists('downloaded_zips'):
        os.makedirs('downloaded_zips', mode=0o777)
    else:
        shutil.rmtree(os.path.join(os.path.dirname(__file__),'downloaded_zips'), ignore_errors=False)
        os.makedirs('downloaded_zips', mode=0o777)

    if not os.path.exists('downloaded_zips_unzipped'):
        os.makedirs('downloaded_zips_unzipped', mode=0o777)
    else:
        shutil.rmtree(os.path.join(os.path.dirname(__file__), 'downloaded_zips_unzipped'), ignore_errors=False)
        os.makedirs('downloaded_zips_unzipped', mode=0o777)
    logging.info('Directories cleanup complete.')
except Exception as e:
    logging.error(str(e))
    exit()
```

This code snippet is to download the log files from the URL when the user inputs the year for which he wants the log files. The log files downloaded will belong to the first day of every month.



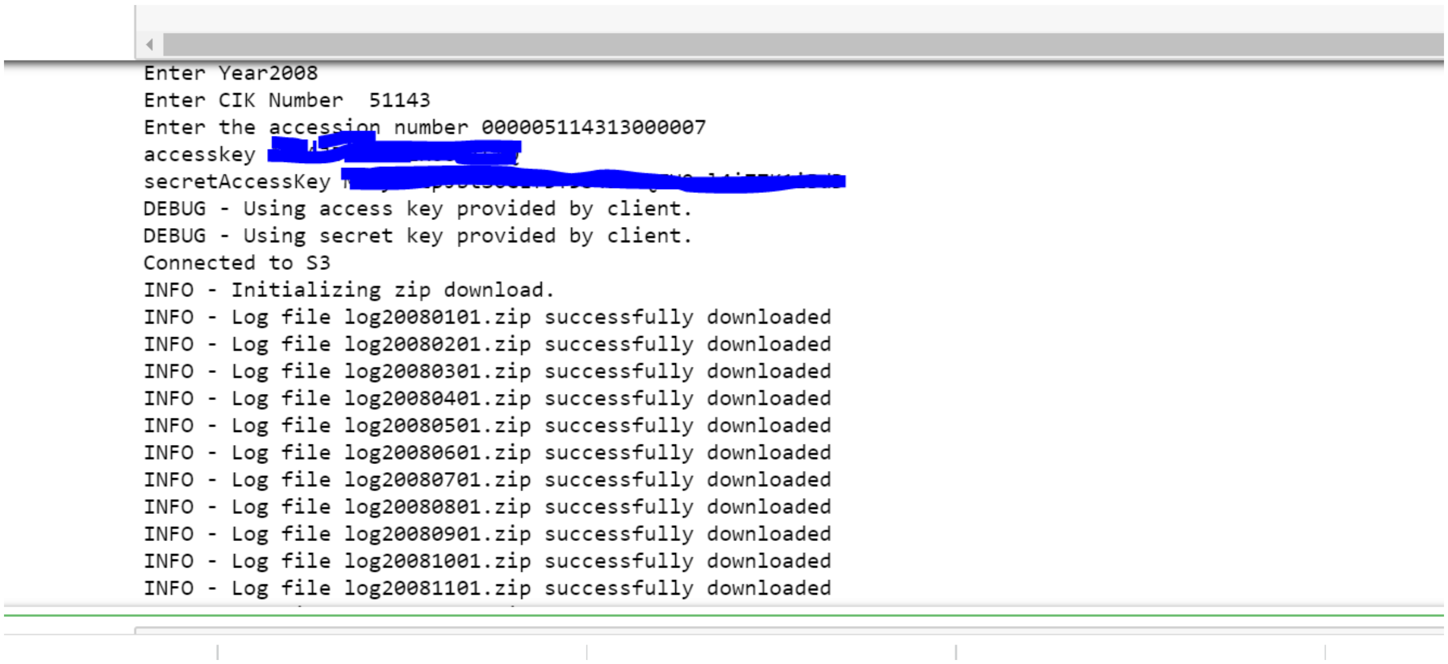
```python
url_final = []
for key, val in qtr_months.items():
    for v in val:
        for d in days:
            url = URL +str(year) +'/' +str(key) +'/' +'log' +str(year) +str(v) + str(format(d,'02d')) +'.zip'
            if download_zip(url):
                break
            else:
                continue
logging.info('All log files downloaded for %s', year)
```

All the downloaded and extracted csv files will be combined into one dataframe as main_csv.csv. If the combining of the dataframe fails then it will throw an exception



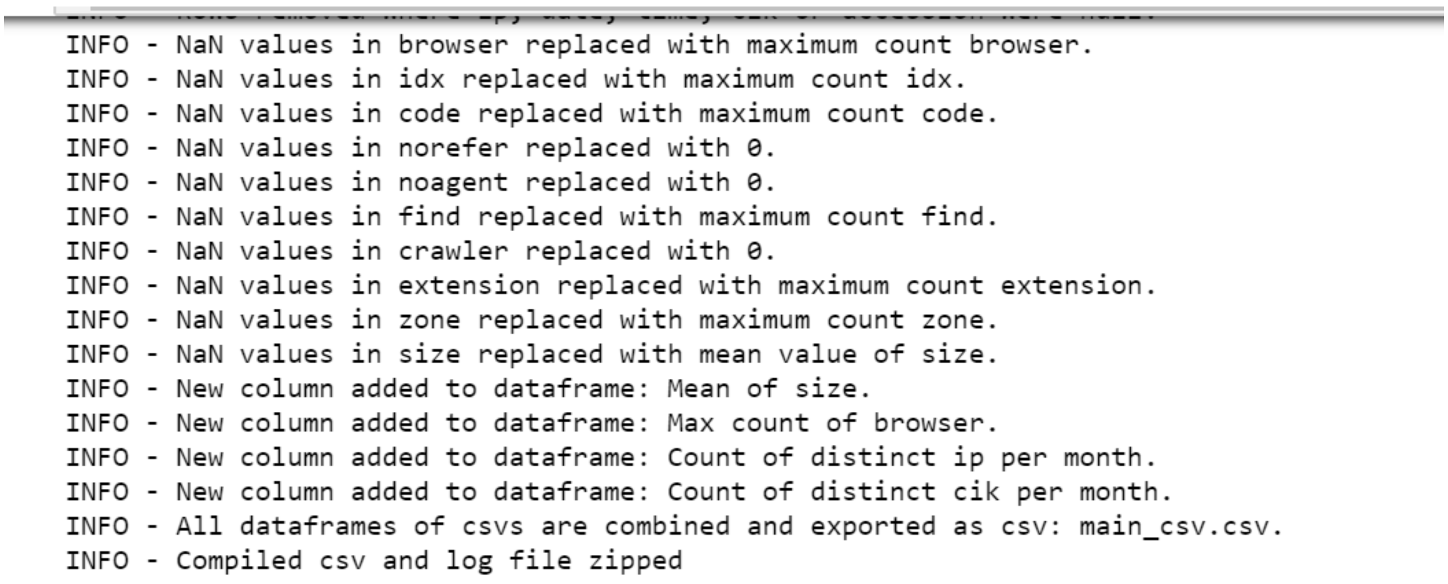
```python
try:
    master_df = pd.concat(all_csv_df_dict)
    master_df.to_csv('main_csv.csv')
    logging.info('All dataframes of csvs are combined and exported as csv: main_csv.csv.')
except Exception as e:
    logging.error(str(e))
    exit()
```

Output:



```
Enter Year2008
Enter CIK Number  51143
Enter the accession number 000005114313000007
accesskey [redacted]
secretAccessKey [redacted]
DEBUG - Using access key provided by client.
DEBUG - Using secret key provided by client.
Connected to S3
INFO - Initializing zip download.
INFO - Log file log20080101.zip successfully downloaded
INFO - Log file log20080201.zip successfully downloaded
INFO - Log file log20080301.zip successfully downloaded
INFO - Log file log20080401.zip successfully downloaded
INFO - Log file log20080501.zip successfully downloaded
INFO - Log file log20080601.zip successfully downloaded
INFO - Log file log20080701.zip successfully downloaded
INFO - Log file log20080801.zip successfully downloaded
INFO - Log file log20080901.zip successfully downloaded
INFO - Log file log20081001.zip successfully downloaded
INFO - Log file log20081101.zip successfully downloaded
```

Cleaning of Missing Data:



```
INFO - NaN values in browser replaced with maximum count browser.
INFO - NaN values in idx replaced with maximum count idx.
INFO - NaN values in code replaced with maximum count code.
INFO - NaN values in norefer replaced with 0.
INFO - NaN values in noagent replaced with 0.
INFO - NaN values in find replaced with maximum count find.
INFO - NaN values in crawler replaced with 0.
INFO - NaN values in extension replaced with maximum count extension.
INFO - NaN values in zone replaced with maximum count zone.
INFO - NaN values in size replaced with mean value of size.
INFO - New column added to dataframe: Mean of size.
INFO - New column added to dataframe: Max count of browser.
INFO - New column added to dataframe: Count of distinct ip per month.
INFO - New column added to dataframe: Count of distinct cik per month.
INFO - All dataframes of csvs are combined and exported as csv: main_csv.csv.
INFO - Compiled csv and log file zipped
```