# Biggest Employers by Region and by Sector

# Reed Data Retrieval

## Requisites:

- A Reed API Key - Link
- JupyterLab Plotly support - Link
- JupyterLab Table of Contents - Link
- Extract Zips

## Introduction:

As the initial task is to discover the biggest employers by region and by sector, a potential source of this data can be retrieved from Job boards, such as Reed, LinkedIn and Indeed.  Due to time constraints Reed.co.uk was chosen as it has a publically available API.

## Code:

### Imports:

All the necessary modules can be installed via the requirements.txt

### Functions:

Return files - Creates a list of filenames from the supplied directory and keyword string.

Parse Skills - Searches a Pandas Series object for keywords which match a list. Returns a list of matching keywords

### Variables:

Several variables are lists of strings in which to search the Reed.co.uk API. For example, the “queries” variable contains strings relating to the job listings (keywords) that we would like to retrieve, such as “Data Engineer” and skills such as “Python”. The “locations” variable contains a dictionary of all the major areas of the UK, with the “Region” value, displaying the region, and the “Cities” value, containing a list of all the cities within that region.

## Job Listings:

Reed's API is structured so that the full description of each job listing cannot be retrieved directly. An initial query must first be ran with the "Search API" which requires a combination of the following parameters:

- **employerId** - id of employer posting job
- **employerProfileId** - profile id of employer posting job
- **keywords** - any search keywords
- **locationName** - the location of the job
- **distanceFromLocation** - distance from location name in miles (default is 10)
- **permanent** - true/false
- **contract** - true/false
- **temp** - true/false
- **partTime** - true/false
- **fullTime** - true/false
- **minimumSalary** - lowest possible salary e.g. 20000
- **maximumSalary** - highest possible salary e.g. 30000
- **postedByRecruitmentAgency** - true/false
- **postedByDirectEmployer** - true/false
- **graduate** - true/false
- **resultsToTake** - maximum number of results to return (defaults and is limited to 100 results)
- **resultsToSkip** - number of results to skip (this can be used with resultsToTake for paging)

This returns the following information:

- **Job Id**
- **Employer Id**
- **Employer  Name**
- **Employer Profile Id**
- **Job Title**
- **Description (Partial)**
- **Location Name**
- **Minimum Salary**
- **Maximum Salary**

A limitation of the API is that only 2000 queries can be called at a time so the code contains a failsafe that pauses once this limit has been reached and continues after the "time_to_sleep" variable.

Once the "Job Listings" cell has ran, each query is then saved in the form of a .csv named by the "Keywords"_"Location"_"Date queried".

## Cleaning Job Listings:

The “Cleaning Job Listings” cell involves, combining each job listings csv, into a Pandas DataFrame, and removing duplicate job ids:

```
master_df.drop_duplicates(subset=['jobId'], keep="first", inplace=True) # Dedupe the same jobs
```

Additionally, Jobs with the same job description and same location have also been de-duplicated:

```
master_df.drop_duplicates(subset=['jobDescription','locationName'], keep="first", inplace=True) # Dedupe jobs with the same job description in the same location
```

Columns were then renamed and job titles found in the “exclude_titles” variable were then dropped. The result is a large DataFrame with the following appearance:

| | Job ID | Employer | Location | Title | Description | Min. Salary | Max. Salary | Search Query | Direct | Graduate |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 37181828 | Apache Associates | Falkirk | UI/UX Designer - HTML - CSS - Wireframing and ... | UX Designer - Wireframing and UI prototyping ... | 40000.0 | 45000.0 | Apache | False | False |
| 1 | 37062977 | Apache Associates | Stirling | Web Developer | Expanding company requires a Mid to Senior Le... | 30000.0 | 35000.0 | Apache | False | False |
| 2 | 37029760 | Apache Associates | Stirling | Web Developer | Expanding company requires a Mid to Senior Le... | 250.0 | 350.0 | Apache | False | False |
| 3 | 37070524 | Apache Associates | Falkirk | Software Project Manager | Overview - Project Management, Full Project L... | 35000.0 | 40000.0 | Apache | False | False |
| 4 | 36730080 | John Ross Associates | Cardiff | PHP Developer - Back End / LAMP Stack / MySQL ... | PHP Developer - Back End / LAMP Stack / MySQL... | 30000.0 | 35000.0 | Apache | False | False |

## Retrieving Direct Job IDs:

With all queries downloaded, we are now able to extract the job ID for each job, which can then be used to run a second query which can retrieve the full job description. However at this time we only want to see the jobs directly posted by the employer so we filter using the “Direct” field:

```
direct_df = master_df[master_df['Direct'] == True] # Filters master_df for only job listings directly from the employer
```

Additionally we want to make sure certain employers are excluded, we did this by grouping all the retrieved jobs by employers:

| Employer | Job ID | Employer | Location | Title | Description | Min. Salary | Max. Salary | Search Query | Direct | Graduate |
|---|---|---|---|---|---|---|---|---|---|---|
| Hays Specialist Recruitment Limited | 632 | 1 | 111 | 491 | 628 | 133 | 149 | 14 | 1 | 2 |
| Harnham | 536 | 1 | 33 | 282 | 535 | 49 | 46 | 14 | 1 | 1 |
| The Training Room IT Careers | 425 | 1 | 196 | 4 | 6 | 1 | 2 | 4 | 2 | 2 |
| Dawson and Walsh | 175 | 1 | 82 | 158 | 173 | 18 | 20 | 9 | 1 | 1 |
| Oscar Technology | 169 | 1 | 44 | 148 | 169 | 33 | 35 | 13 | 1 | 1 |
| REED | 151 | 1 | 80 | 119 | 146 | 54 | 63 | 9 | 2 | 2 |
| Search Consultancy | 137 | 1 | 46 | 129 | 137 | 22 | 24 | 11 | 1 | 2 |
| Nuffield Health | 119 | 1 | 116 | 6 | 6 | 3 | 3 | 2 | 2 | 2 |

We can see that a number of employers are actually recruitment agencies. We exclude this employers using the following code:

```
direct_df = direct_df[~direct_df['Employer'].str.lower().str.contains('|'.join(exclude_employers).lower(), na=False)] # Removes irrelevant employers
```

With jobs now filtered we can now create a list of job ids in which we can now acquire their full description.

```
list_of_jobids = direct_df['Job ID'].tolist() # Creats a list of job Ids we would like to retrieve the whole description from
```

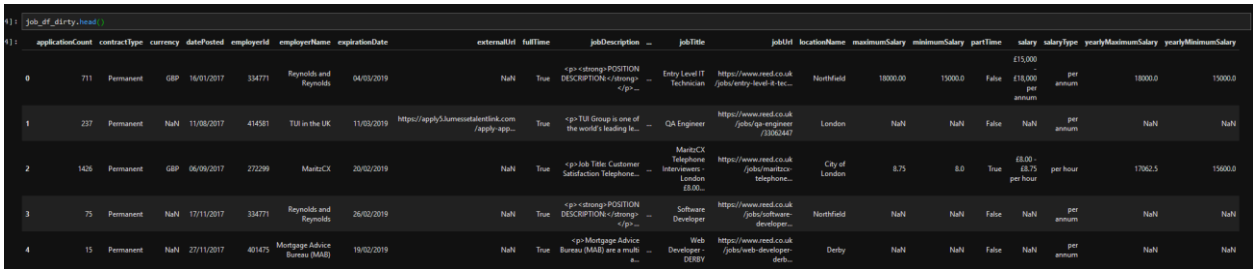
```
[11]: list_of_jobids

[11]: [37094004,
       37093983,
       36437884,
       36206305,
       36206299,
       37084188,
```

## Retrieving Data Within each Job ID

Now that we have acquired a list of Job Ids, we can now run the Reed "Details API" which can return the full job description. An example of the results:


```
{'employerId': 420428,
 'employerName': 'Intouch Games Ltd',
 'jobId': 37094004,
 'jobTitle': 'Senior C++ Programmer - C++, Linux (Debian)',
 'locationName': 'Birmingham',
 'minimumSalary': 40000.0,
 'maximumSalary': 50000.0,
 'yearlyMinimumSalary': 40000.0,
 'yearlyMaximumSalary': 50000.0,
 'currency': 'GBP',
 'salaryType': 'per annum',
 'salary': '£40,000 - £50,000 per annum',
 'datePosted': '22/01/2019',
 'expirationDate': '05/03/2019',
 'externalUrl': None,
 'jobUrl': 'https://www.reed.co.uk/jobs/senior-c-programmer-c-linux-debian/37094004',
 'partTime': False,
 'fullTime': True,
 'contractType': 'Permanent',
 'jobDescription': " <p><strong>Job Title:</strong> Senior C&#43;&#43; Programmer - C&#43;&#43;, Linux (Debian)</p>
<p><strong>Location:</strong> Birmingham, Halesowen (inc in-house relocation assistance)</p> <p><strong>Salary:</st
rong> &#163;40,000 - &#163;50,000 &#43; benefits (including flexible working hours &#43; bi-annual salary reviews)<
/p> <p><strong>Keywords:</strong> C&#43;&#43;, Developer, C&#43;&#43; Engineer, Software Programmer, Web, Apache, L
inux, XML, MySQL, Eclipse, SVN/GIT, Birmingham</p> <p>Senior C&#43;&#43; Programmer with excellent C&#43;&#43;, Apa
che, Linux and API expertise is required by a multiple award winning games studio based in Birmingham who are the U
K's largest privately owned mobile e-gaming studio! We currently have just over 200&#43; employees  here at our Bir
mingham HQ and we offer some of the UK's best career progression plans and earning potential with bi-annual salary
reviews!</p> <p>This role would give you the opportunity to initially head up a brand new project where you will im
plement a number of mobile payment solutions (inc direct operator billing, SMS payment and web based payments) onto
our Linux (Debian) based servers. We currently  have just over 2million customers in the UK and we are now stepping
into new international territories with localised games, campaigns and websites.</p> <p>You will get your hands on
the latest tech, play an active role in software design designs (both client and server side) and you'll also archi
tect and implement game logic and interface with various data services on a number of solutions related to our back
-end  servers (both Linux and NoSQL solutions).</p> <p><strong>Key skills we're looking for…</strong></p> <ul> <li>
Extensive commercial C&#43;&#43; coding experience</li><li>Commercial expertise coding C&#43;&#43; on Linux platfor
ms (Debian, Redhat, CentOS, Ubuntu etc)</li><li>Good Linux OS familiarity (including debugging skills)</li><li>Expe
rience with IDE's such as Eclipse</li></ul> <p><strong>Bonus points for…</strong></p> <ul> <li>Advanced skills inte
rfacing with MySQL and NoSQL (redis, memcache, MongoDB etc)</li></ul> <p>Therefore, if you a highly skilled C&#43;&
#43; Engineer with Linux and API expertise and you would like to join a multiple award winning eGaming studio who h
ave been accredited as one of the 'Top 1000 Companies to Inspire Britain', been recognised by the BBC as  a top tec
hnology employer and have a 1 star accreditation as a 'Top Company to Work For' then send in your CV today for revi
ew!</p> ",
 'applicationCount': 1}
```


When we run the cells located underneath "Retrieving Data Within each Job ID" this returns the following DataFrame:



```
job_df_dirty.head()
```

| | applicationCount | contractType | currency | datePosted | employerId | employerName | expirationDate | externalUrl | fullTime | jobDescription | ... | jobTitle | jobUrl | locationName | maximumSalary | minimumSalary | partTime | salary | salaryType | yearlyMaximumSalary | yearlyMinimumSalary |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 711 | Permanent | GBP | 16/01/2017 | 334771 | Reynolds and Reynolds | 04/03/2019 | NaN | True |  POSITION DESCRIPTION ... | ... | Entry Level IT Technician | https://www.reed.co.uk/jobs/entry-level-it-tec... | Northfield | 18000.00 | 15000.0 | False | £15,000 - £18,000 per annum | per annum | 18000.0 | 15000.0 |
| 1 | 237 | Permanent | NaN | 11/08/2017 | 414581 | TUI in the UK | 11/03/2019 | https://apply5.lumessetalentlink.com/apply-app... | True | TUI Group is one of the world's leading le... | ... | QA Engineer | https://www.reed.co.uk/jobs/qa-engineer/33062447 | London | NaN | NaN | False | NaN | per annum | NaN | NaN |
| 2 | 1426 | Permanent | GBP | 06/09/2017 | 272299 | MaritzCX | 20/02/2019 | NaN | True |  Job Title: Customer Satisfaction Telephone... | ... | MaritzCX Telephone Interviewers - London £8.00... | https://www.reed.co.uk/jobs/maritzcx-telephone... | City of London | 8.75 | 8.0 | True | £8.00 - £8.75 per hour | per hour | 17062.5 | 15600.0 |
| 3 | 75 | Permanent | NaN | 17/11/2017 | 334771 | Reynolds and Reynolds | 26/02/2019 | NaN | True |  POSITION DESCRIPTION ... | ... | Software Developer | https://www.reed.co.uk/jobs/software-developer... | Northfield | NaN | NaN | False | NaN | per annum | NaN | NaN |
| 4 | 15 | Permanent | NaN | 27/11/2017 | 401475 | Mortgage Advice Bureau (MAB) | 19/02/2019 | NaN | True | Mortgage Advice Bureau (MAB) are a multi a... | ... | Web Developer - DERBY | https://www.reed.co.uk/jobs/web-developer-derb... | Derby | NaN | NaN | False | NaN | per annum | NaN | NaN |

To determine what skills are present in each job description, the parse skills function is used:

```
job_df_skills["skills"] = job_df_skills["jobDescription"].map(lambda x: parse_skills(x)) # Creates a column with list of keywords matching the "keywords" variable
job_df_skills["ext_skills"] = job_df_skills["jobDescription"].map(lambda x: parse_skills(x,True)) # Creates a column with list of keywords matching the "ext_keywords" variable
```

This returns a list of matched keywords based on the “keywords” variable as a new column:

| | skills | ext_skills |
|---|---|---|
| N | [SQL] | [CSS, Javascript, HTML] |
| N | [SQL] | [CSS, Javascript, HTML] |
| N | [Excel] | None |
| N | [Excel] | None |
| N | [Excel] | None |

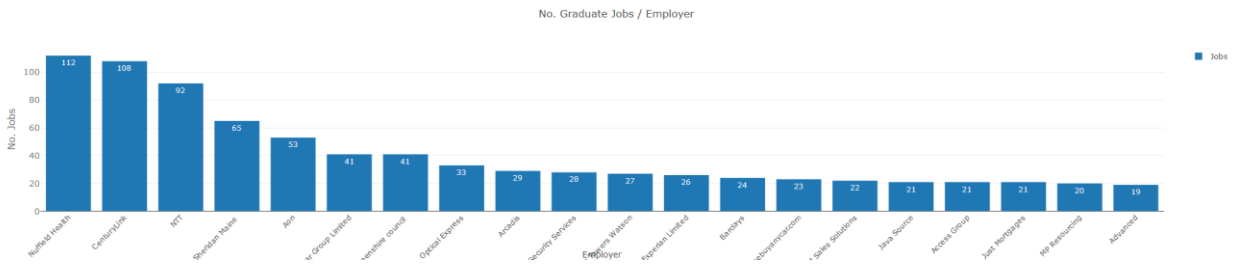
This DataFrame is then converted into a new DataFrame with each skill as an indicator value:

| | jobId | locationName | employerName | jobTitle | minimumSalary | maximumSalary | AI | Apache | Database | Excel | Hadoop | Machine Learning | NoSQL | Python | SQL | Spark | Tableau |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 4 | 33895415 | Derby | Mortgage Advice Bureau (MAB) | Web Developer - DERBY | NaN | NaN | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 9 | 34923372 | Elland | A-Safe (UK) | Sharepoint/ Web Developer | NaN | NaN | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 10 | 35048241 | Chelmsford | Towergate Insurance Limited | Commercial Account Handler | NaN | NaN | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 13 | 35086819 | Manchester | Greenergy International Ltd | Sales Ledger Analyst | NaN | NaN | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 24 | 35536690 | West London | Marsh & Parsons | Client Accountant | NaN | NaN | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

## Visualisations:

The visualisations are all based on the [Plotly framework ](#) and are generated with either the “iplot” function which returns an in-line graph:

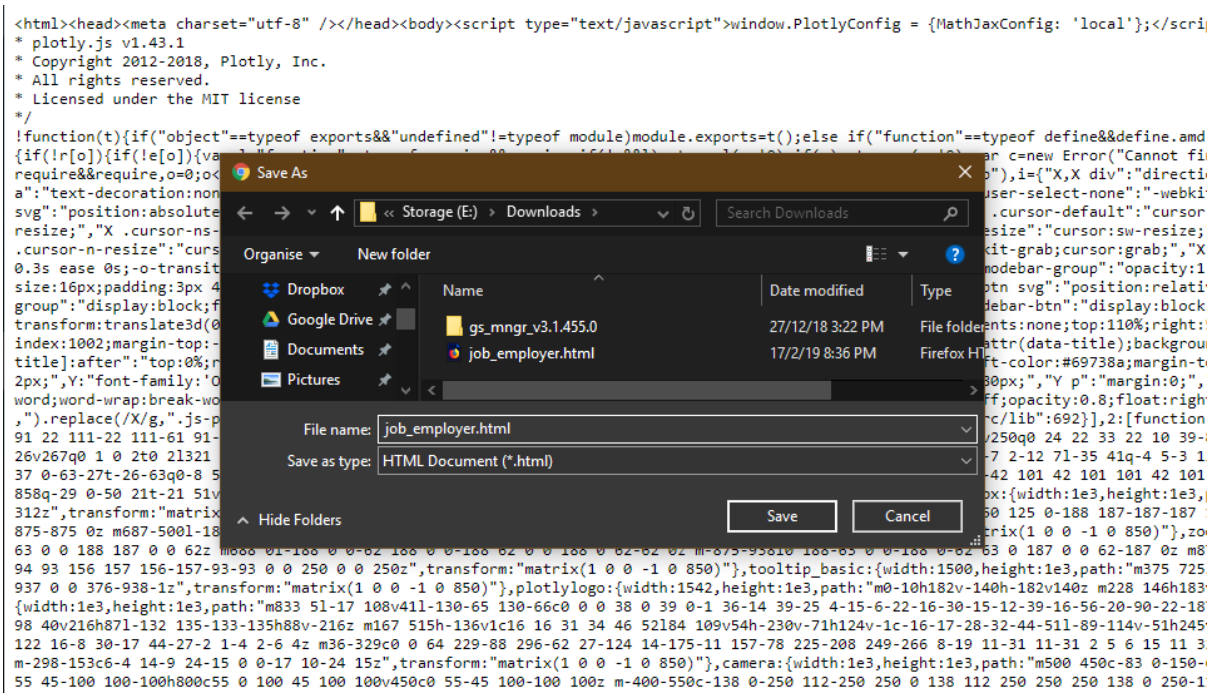
```
iplot(fig, filename="job_employer.html")
```




Or the “plot” function which exports a graph as a html file:

```
plot(fig, filename="job_employer.html")
```

To view the exported visualisations located on GitHub, you must first download a html and save it:



# FTSE Data Retrieval

## Requisites**:**

- Extract Zip

## Introduction:

An alternative to scraping job boards is to determine the largest companies by their market cap. This was achieved using the FTSE all share from the London Stock Exchange.

## Code:

### Imports:

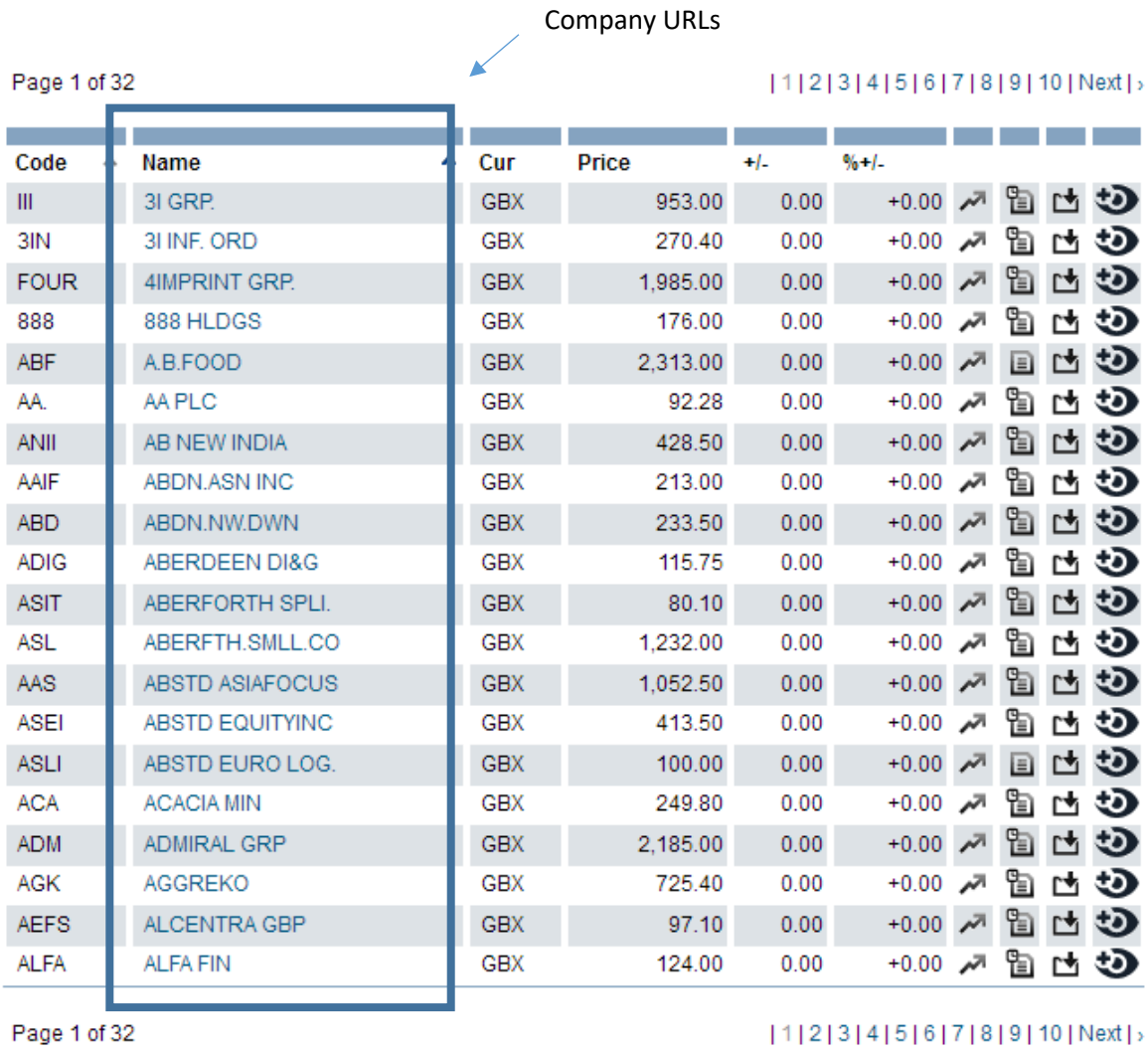
All the necessary modules can be installed via the requirements.txt

### Functions:

Return files - Creates a list of filenames from the supplied directory and keyword string.

### Creating a data dictionary (All necessary URLS):

Similarly to the Reed data retrieval, the details of each company could not be obtained directly but has to be obtain by first running a script to obtain each page URL from the following:




When the cells are ran, all company URLs across all pages are exported into a single file called links.csv.

## Retrieving Info from Website

The cells under “Retrieving Info From Website” opens each URL in links.csv and obtains the following information:

- Company website': 'http://www.3igroup.com',
- Company address': '16 Palace Street, London, SW1E 5JD, United Kingdom',
- 'FTSE ICB sector': 'Financial Services',
- 'FTSE ICB subsector': 'Specialty Finance',
- 'Company market cap, £m*': '8854.20',
- 'Admission date': '18 Jul 1994'
- 'Company': '3I GRP.'},

Each company is then added to a DataFrame “master_df” which can then be used for further analysis.