# Project Deliverable 2

### Project Deliverable 2

More data should have been collected to perform a more thorough analysis of the data and attempt to answer one additional question relevant to your project proposal which you will submit as a pull request.

All contractors commissioned by the state for major construction projects need to report their ethnic and gender makeup of the work forces. The WGBH would like to understand the data contained in those Summary of Workforce Utilization reports. Furthermore, the WGBH is interested in getting data-driven insights of the impact drawn upon specific groups of working forces between 2019 to 2020. The data is given in PDF format and organized by hours spent per project per organization. Our goal is to first extract data in proper formats from the PDF files and then run some analysis.

## Logistics

### Weekly Meeting with the PM

- ❏ Lingyan Jiang is Thurs 11:30 AM - 1:00 PM

### Weekly Meeting With WGBH

- ❏ Paul Singer, - every other Thurs 11:30 AM - 1:00 PM
- ❏ Spark Liason - Greta Bruce

### Contact List

- Client Paul Singer paul_singer@wgbh.org,
- Spark Liason Greta Bruce gretab@bu.edu,
- PM Lingyan Jiang lingyanj@bu.edu,
- Students Rep Jena Jordahl jenajj@bu.edu,

Elisa Cordeiro Lopes elisacl@bu.edu, Richard Lee rlee99@bu.edu, Murtadha Bahrani murtadha@bu.edu, Carmen Sabrina Araujo sabrinaa@bu.edu.

Github accounts

elisa3lopes, rlee99, murtio, carmen-araujo, jenajjedu

# Data

The data is collected weekly by DCAMM. They sort it by months and keep it in PDF form. DCAMM already provided WGBH the work force from 2019 and will provide in March the data from 2020. The data is organized as tables of projects (such as bridges, buildings, etc) containing the companies included, their types of workers, and the hour rate separated by race, sex, and ethnicy. For this project, no additional datasets are required to be extracted, but our team is open to get any other information as it seems relevant to analysis. An example of a file is April 2019: https://drive.google.com/file/d/1brxGTjfkhwKRXPAbzDwHl4bP6J08Xwtz/view?usp=sharing

We have been given a file folder with files for each month Jan - Dec 2019, e.g. WorkforceUtilizationSummaryReportApril2018.pdf.

See Image 10 below, the tox testing framework will validate our accounting ledger categories and subtotals and totals are correctly parsed.

Some off-the-shelf parsers could not read the pdf files into tables, camelot for example. See Image 11 for the error codes.

Due to issues with reading the thousand's place value numbers, 3,432 , we create our preliminary analysis by manipulating the csv output in EXCEL.

## 1. Collect and pre-process a secondary batch of data

We mentioned in Deliverable 1 that our project work will have a larger portion of total time devoted to data cleanup vs data exploration due to the condition of the data sources. From that we tested some methods to better extract the data from the CSV file.

We were able to extract the non-numerical part of the data (i.e. the project codes, project names, contractors/company names, and construction-trade/job-names) of the

PDF file WorkforceUtilizationSummaryReport**April2019**.pdf and convert it to a pandas dataframe. You can find the first 10 rows of the pandas dataframe in **image 3**. Extracting the project codes and project names was pretty easy because we just had to go through the csv and look for the string ‘Project Name:’ in order to find them.

Extracting the contractor/company names was by far the most challenging part because there is no label indicating whether a string is a contractor/company name. So we manually went through the csv to figure out any patterns in what type of string preceded contractor/company names and found a few different specific strings that did. However, these preceding strings did not only precede contractor/company names so we had to employ many if/else statements to make sure we could distinguish between a contractor/company name and other irrelevant strings.

Then, in order to extract construction-trade/job-names, we took advantage of the fact that they are in all caps. There are, however, a few exceptions of certain construction-trade/job-names that have some words in lowercase and some contractor/company names in all caps. To fix the first issue, we noticed that the construction-trade/job-names with lower case words include the substring ‘Class’ so we added another segment to the if statement to include as a construction-trade/job-name any string with this substring. To fix the second issue, we added a special if statement for that particular contractor/company name. For other PDFs, it will be necessary to check for any contractor/company names that are all capitalized and include them in this special if statement.

One of the most successful methods is to find patterns in the data and hard code functions that will modify the data. For example, we could extract only the numbers for the rows. We used the point '.' as a number indicator and the isalpha() function to avoid getting elements with letters and numbers, such as project codes. Image 1 shows the result list from this function, and it clearly shows some rows come perfectly in length of 10 (the amount needed) but some come mixed, missing elements, or concatenated, as shown in Image 2. To fix this, we are keeping the same hard code method and creating a function that identifies these broken columns by the length 1 list, puts them together, and separates the list. Since all numbers finish with 2 decimal cases ('. _ _'), we are using this as the separator. Image 4 shows one example of our functions.

This number extraction is important since we can concatenate it with the pandas dataframe already created and get the numeral data for the analysis. The numbers are the basis for the analysis and without them we cannot make any meaningful conclusions about the data. Subsequently, we will start to look for any patterns or trends to answer questions revolving around race, sex, and opportunities for state contracts.

We met with IT programmers to review our issues so as a team we also did another research spike to see if another library or combination of the existing libraries would produce a better interim csv file. We tested camelot which transfers data directly from pdf to pandas dataframes and found the data was not uniform so the import failed. See Image 11 below in screenshots. Then in the “threetierledger.py” parser, we tried using the pandas “thousands” parameter to correct the number $3,000 dollars from being split between two cells. The “thousands” parameter did not keep the number from being split. Then we tried using a nesting of io library inside pandas inside tabula and various combinations of these. Nothing produced an output that did not split the numbers on the detail lines. Consulting with the professor, we understood that our final task will be to write a small parser to read these numbers into one cell.

In summary, we were able to pre-process a second batch of data by being able to specify the state-machine of the process to read the PDF accounting ledger reports. Image 12 depicts the state-machine necessary to build the data analysis data frame. We also understand the work ahead to read the numbers on the detail lines.

## 2. Answer another key question

a. How will we extract data from our PDF files?

   i. We chose to stick to Tabula library and PyPDF2 since they presented good results even though it had limitations

b. Is there a difference between state-paid contractual hours based on color and/or sex?

   i. Based on our initial analysis from the direct manipulation of the CSV file, we can see that caucasian males have accumulated the most state-paid contractual hours.

Questions that remain:

- What are the factors, e.g. location of the project, that fair in hiring working crews?
- How do state-wise elections affect hiring decisions across projects? Did construction companies hire fewer minorities, people of color and/or women, during the pandemic?
- Did the companies’ work workload change because of COVID-19?

## 3. Refine project scope and list of limitations with data and potential risks of achieving project goal

Goals for the project have been clarified. Instead of undergoing an analysis on the data, our main goal is to create a tool that parses and extracts the data of these reports in a manner that can be repeatable for any other report.

We have been able to make progress in the construction of our PDF parsers. As mentioned earlier, we were able to extract non-numerical parts of the data using patterns in their respective strings. Extracting numerical data has been the most challenging part. Each column has 10 rows, so if all the rows were parsed correctly it wouldn't be too difficult. Unfortunately many rows are extracted in a way where some rows end prematurely and continue on a different array. This makes selecting numbers in each array extremely difficult. There are also many cases where numbers are merged instead of being separated by column. More generally, this shows the issues we are still facing. As mentioned in the Methods section, we have ideas on how to solve these issues.

Since the methods have been hard coded and finding patterns on the data, a limitation is that, in the future, there might be PDFs transformed into CSVs with different patterns not identified before. For example, some numbers are larger than a thousand and contain commas. Therefore, using split.(',') has been losing thousands of data, which must be changed.

Extracting and parsing data hasn't been too difficult, however the inconsistencies and weird attributes the merged cells have caused our calculations and methods of extraction to be inconsistent. Many rows are supposed to carry certain values such as race and gender yet are misaligned.

Our next steps would be to somehow find a way around these inconsistencies and misaligned columns in order to create clean and readable dataframes. For now we have manipulated data by removing headers and shifting columns in the CSV directly and have created some initial graphs to have a better understanding of what we are dealing with.

## 4. Refine the preliminary analysis of the data performed in PD1

We ran some preliminary analysis of the 2019 April report. First, we removed the headers using Excel then imported that CSV file into Dataframe and continued cleaning the data, as seen in Image 5. Image 6, 7, 8 and 9 made comparison of total hours of work by gender, race, ethnicity, and new incommers.

```
[],
[],
[16.0, 8.0, 0.0, 0.0, 0.0, 0.0, 8.0, 0.0, 0.0, 16.0],
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
[16.0, 8.0, 0.0, 0.0, 0.0, 0.0, 8.0, 0.0, 0.0, 16.0],
[],
[],
[],
[],
[],
[16.0, '8.00'],
['0.000.000.000.008.00', '0.000.00'],
['16.00'],
[0.0, 16.0],
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
[16.0, 8.0, 0.0, 0.0, 0.0, 0.0, 8.0, 0.0, 0.0, 16.0],
[16.0, 8.0, 0.0, 0.0, 0.0, 0.0, 8.0, 0.0, 0.0, 16.0],
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0],
[16.0, 8.0, 0.0, 0.0, 0.0, 0.0, 8.0, 0.0, 0.0, 16.0],
[],
```

**Image 1: Extracting and printing only numbers from the CSV.**

```
[422.5, '353.00'],
['0.000.000.000.0069.50', '0.000.00'],
['422.50'],
```

**Image 2: Example of concatenated row.**

| | Project_Code | Project_Name | Contractor_Name | Construction_Trade |
|---|---|---|---|---|
| 0 | AEP1802E UT1 C | Utility Simple Fix | Batallas Electric Inc. | ELECTRICIAN |
| 1 | AEP1802E UT1 C | Utility Simple Fix | Batallas Electric Inc. | LABORER |
| 2 | AEP1802E UT1 C | Utility Simple Fix | Rise Engineering | INSULATOR (PIPES & TANKS) |
| 3 | CHE1604 DC1 CM | Construction Management Services Chelsea Soldi... | E. Amanti & Sons Inc | LABORER |
| 4 | CHE1604 DC1 CM | Construction Management Services Chelsea Soldi... | E. Amanti & Sons Inc | PIPEFITTER & STEAMFITTER |
| 5 | CHE1604 DC1 CM | Construction Management Services Chelsea Soldi... | E. Amanti & Sons Inc | PLUMBERS & GASFITTERS |
| 6 | CHE1604 DC1 CM | Construction Management Services Chelsea Soldi... | Harnum Industries LTD | LABORER |
| 7 | CHE1604 DC1 CM | Construction Management Services Chelsea Soldi... | S&F Concrete Contractors Inc. | EQUIPMENT OPERATOR (Class B CDL) |
| 8 | CHE1604 DC1 CM | Construction Management Services Chelsea Soldi... | S&F Concrete Contractors Inc. | LABORER |
| 9 | CHE1604 DC1 CM | Construction Management Services Chelsea Soldi... | W.L. French Excavating Corp. | ADS/SUBMERSIBLE PILOT |

**Image 3. First 10 rows of sample pandas dataframe**

```
1  def hasLetter(inputString):
2      return any(char.isalpha() for char in inputString)
3
4  def hasPoint(inputString):
5      return any(char=='.' for char in inputString)
```

**Image 4. Example of functions created to extract numbers**

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | CraftLevel | | TotalEmploy | Caucasian | AfricanAmer | Hispanic | Asian | NativeAmeri | Other | NotSpecified | TotalFemale | TotalMale |
| 2 | Batallas Electric Inc. | | | | | | | | | | | |
| 3 | ELECTRICIAN | Journey | 17.5 | 17.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 17.5 |
| 4 | | Apprentice | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | | A/J Ratio | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | | New Hire | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | | Subtotal | 17.5 | 17.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 17.5 |
| 8 | LABORER | Journey | 24 | 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 24 |
| 9 | | Apprentice | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | | A/J Ratio | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 11 | | New Hire | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 12 | | Subtotal | 24 | 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 24 |
| 13 | Total for Contractor | Journey | 41.5 | 41.5 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 14 | | | | | | | | | | | 0 | 41.5 |
| 15 | | Apprentice | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 16 | | A/J Ratio | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 17 | | New Hire | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 18 | | Subtotal | 41.5 | 41.5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 41.5 |

**Image 5. Removing the headers using Excel**

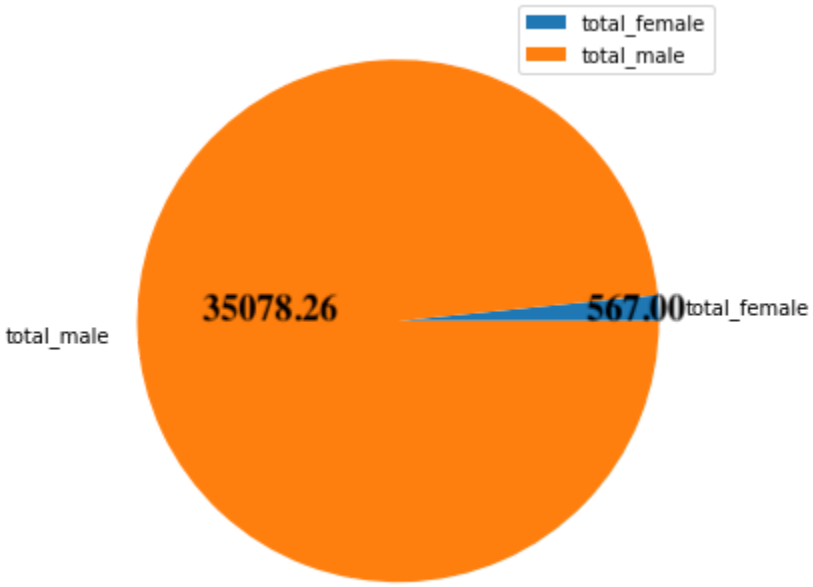


**Image 6: Graph showing total numbers of male in comparison with female**

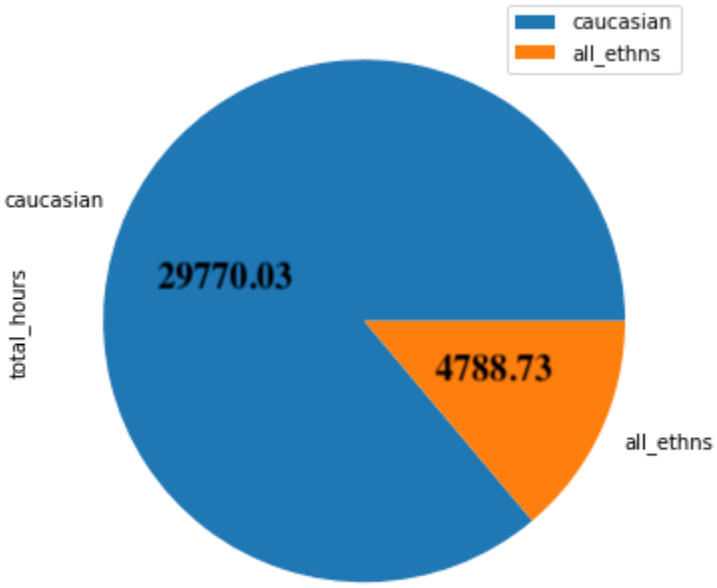


**Image 7: Graph showing total number of Caucasian hours worked vs. all ethnicities**

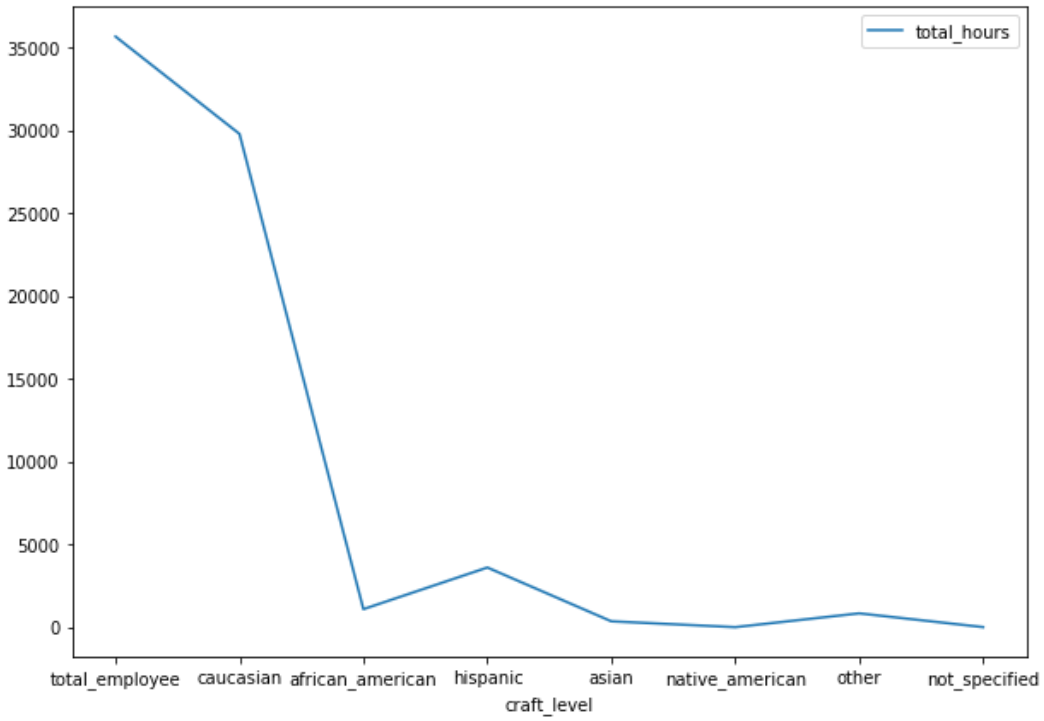


**Image 8: Graph showing total number of hours by ethnicity**

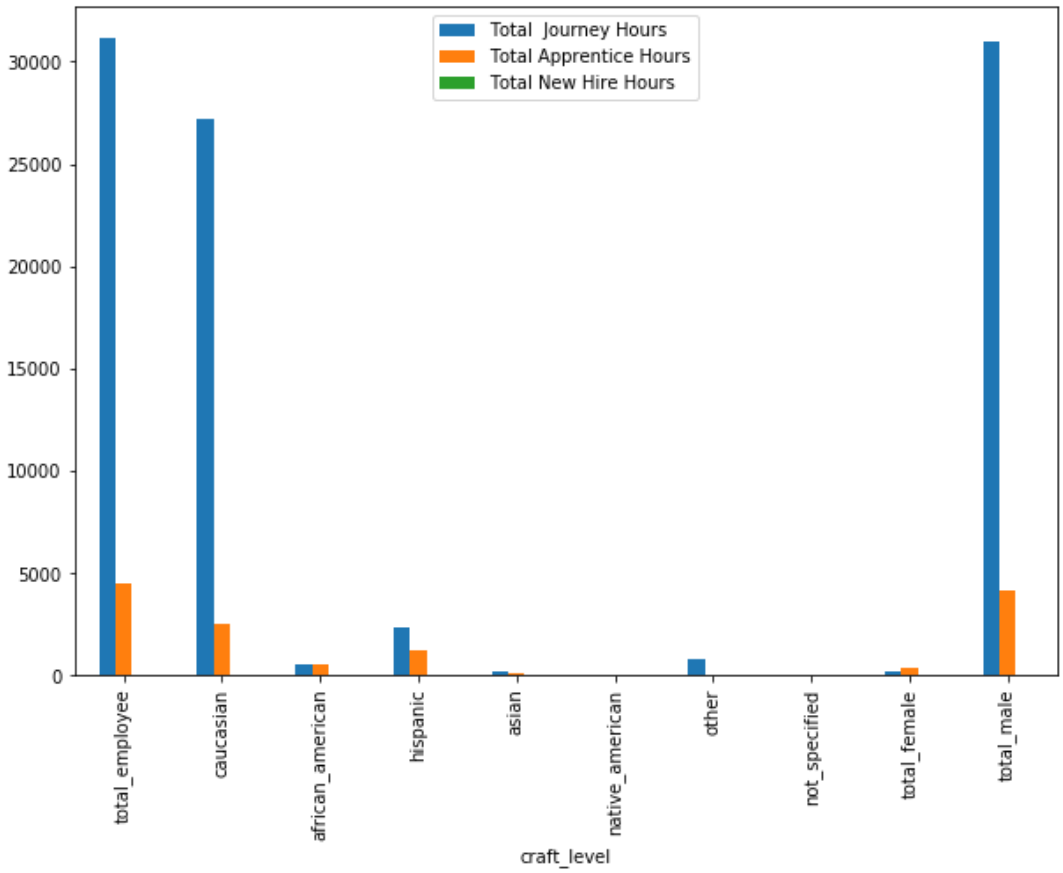


**Image 9: Total number of hours by ethnicity split up by Journey, Apprentice, and New Hires**

```
 2021-03-24 21:36:06  KitchenMac2 in ~/workspaces/CS-506-Homeworks/CS506Spring2021Repository/WGBH-DCAMM
± |delv_2 → origin {1} S:7 U:8 ?:3 ✗| → tox
GLOB sdist-make: /Users/jenajordahl/workspaces/CS-506-Homeworks/CS506Spring2021Repository/WGBH-DCAMM/setup.py
py39 inst-nodeps: /Users/jenajordahl/workspaces/CS-506-Homeworks/CS506Spring2021Repository/WGBH-DCAMM/.tox/.tmp/package/1/WGBH-0.0.1.zip
py39 installed: appdirs==1.4.4,attrs==20.3.0,camelot-py==0.8.2,cffi==1.14.5,chardet==4.0.0,click==7.1.2,coverage==5.5,cryptography==3.4.6,cycler==0.10.0,distlib==0.3.1,distro==1.5.0,
-xmlfile==1.0.1,filelock==3.0.12,iniconfig==1.1.1,joblib==1.0.1,kiwisolver==1.3.1,matplotlib==3.3.4,nltk==3.5,numpy==1.20.1,openpyxl==3.0.7,packaging==20.9,pandas==1.2.3,pdfminer.si
20201018,Pillow==8.1.2,pluggy==0.13.1,py==1.10.0,pycparser==2.20,pyparsing==2.4.7,PyPDF2==1.26.0,pytest==6.2.2,pytest-cov==2.11.1,python-dateutil==2.8.1,pytz==2021.1,regex==2021.3.17
cikit-learn==0.24.1,scipy==1.6.1,seaborn==0.11.1,six==1.15.0,sklearn==0.0,sortedcontainers==2.3.0,tabula-py==2.2.0,testfixtures==6.17.1,threadpoolctl==2.1.0,toml==0.10.2,tox==3.23.0,
dm==4.59.0,virtualenv==20.4.3,WGBH @ file:///Users/jenajordahl/workspaces/CS-506-Homeworks/CS506Spring2021Repository/WGBH-DCAMM/.tox/.tmp/package/1/WGBH-0.0.1.zip
py39 run-test-pre: PYTHONHASHSEED='2607796989'
py39 run-test: commands[0] | pytest --cov=src
.Coverage.py warning: No data was collected. (no-data-collected)
WARNING: Failed to generate report: No data to report.

/Users/jenajordahl/workspaces/CS-506-Homeworks/CS506Spring2021Repository/WGBH-DCAMM/.tox/py39/lib/python3.9/site-packages/pytest_cov/plugin.py:271: PytestWarning: Failed to generate
port: No data to report.

  self.cov_controller.finish()


---------- coverage: platform darwin, python 3.9.2-final-0 -----------

1 passed in 0.39s
_____________________________________ summary _____________________________________
  py39: commands succeeded
  congratulations :)

 2021-03-24 21:36:49  KitchenMac2 in ~/workspaces/CS-506-Homeworks/CS506Spring2021Repository/WGBH-DCAMM
± |delv_2 → origin {1} S:7 U:8 ?:3 ✗| →
```

**Image 10: Tox Testing Framework to ensure total number of lines are actually read**

```
 2021-03-24 21:36:49  KitchenMac2 in ~/workspaces/CS-506-Homeworks/CS506Spring2021Repository/WGBH-DCAMM
± |delv_2 → origin {1} S:7 U:8 ?:3 ✗| → python3 src/parsers/camelot_convert.py
Traceback (most recent call last):
  File "src/parsers/camelot_convert.py", line 1, in <module>
    import camelot
  File "/Users/jenajordahl/workspaces/CS-506-Homeworks/CS506Spring2021Repository/WGBH-DCAMM/venv/lib/python3.8/site-packages/camelot/__init__.py", line 6, in <module>
    from .io import read_pdf
  File "/Users/jenajordahl/workspaces/CS-506-Homeworks/CS506Spring2021Repository/WGBH-DCAMM/venv/lib/python3.8/site-packages/camelot/io.py", line 5, in <module>
    from .handlers import PDFHandler
  File "/Users/jenajordahl/workspaces/CS-506-Homeworks/CS506Spring2021Repository/WGBH-DCAMM/venv/lib/python3.8/site-packages/camelot/handlers.py", line 8, in <module>
    from .core import TableList
ImportError: cannot import name 'TableList' from 'camelot.core' (/Users/jenajordahl/workspaces/CS-506-Homeworks/CS506Spring2021Repository/WGBH-DCAMM/venv/lib/python3.8/site-packages/camelot/core/__init__.py)

 2021-03-24 22:49:49  KitchenMac2 in ~/workspaces/CS-506-Homeworks/CS506Spring2021Repository/WGBH-DCAMM
± |delv_2 → origin {1} S:7 U:9 ?:3 ✗| →
```

**Image 11: Tried to use camelot library to read pdf directly to a pandas DataFrame, No Go.**

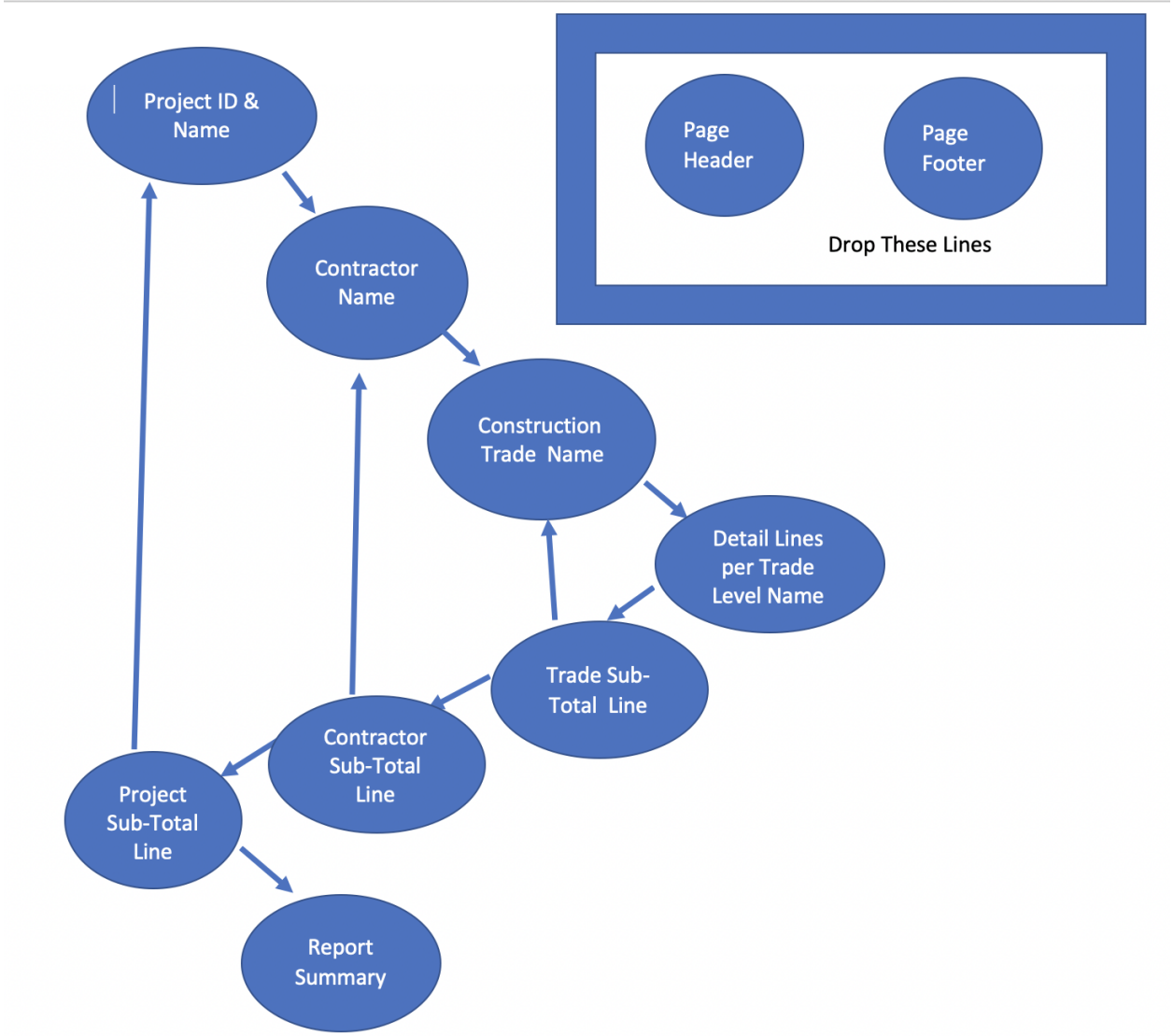


**Image 12: The State Machine for the general parser which will have a small parser for detail lines to keep the thousand's place value connected to the rest of the number. 3,432 in one cell rather than 3 432 in two cells**