

# PROJECT No.2 - Group 6

## ETL - Project [Melting Ice]

By. Amrita Gurung, Jason Karriker, Jason Tan, Nobon Ghalley, Pragya Shakya, Sajita Baniya

---



## Introduction

The purpose of this project was to demonstrate a complete Extract, Transform and Loading (ETL) Process. This process was used to blend data from multiple sources, and loading into a data warehouse<sup>[1]</sup>. Our team decided to amalgamate data pertaining to the effect of global warming, thus the dataset of our choice comprises sources from National Aeronautics and Space Administration (NASA), Earth System Research Laboratory (ESRL), Carbon Dioxide Information Analysis Center (CDIAC) and National Snow and Ice Data Center (NSIDC).

---

---

## Thought Process

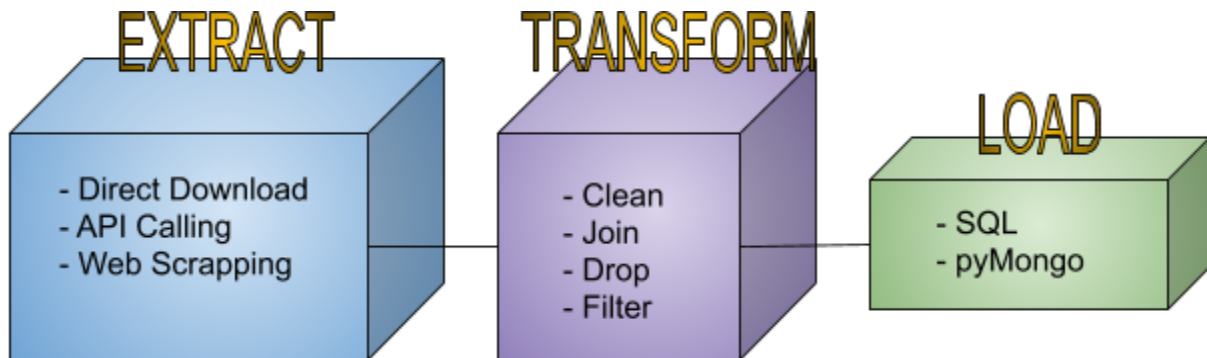


Figure 1.0 ETL Process was given as guideline

The ETL guidelines were provided by our instructor, and our team went through the process of determining what kind of methods can be used for each step of the process. Some were proficient in certain process, thus our team divided tasks based on each team members' mastery and preference.

---

## Extract

After considering web scraping from multiple reputable websites such as NASA, NOAA and NSIDC, we found a third party website that integrated the data from the above mentioned sources and displaying them in an interactive chart. There were multiple options offered for data extract, web-scraping with selenium and BeautifulSoup being the most complicated. That being said, one of the difficulties were accessing the NASA api source which delimited our web-scraping attempt. Thus, our group decided to proceed with the route of importing .json file through python. The data were packaged in csv and json format and the website developer allowed the use of the dataset with reference to the source of the datasets. In order to extract the datasets, an extra module named 'Data Package' was needed in the python environment. Once loaded, the data were saved in csv format for the next step of Transform.

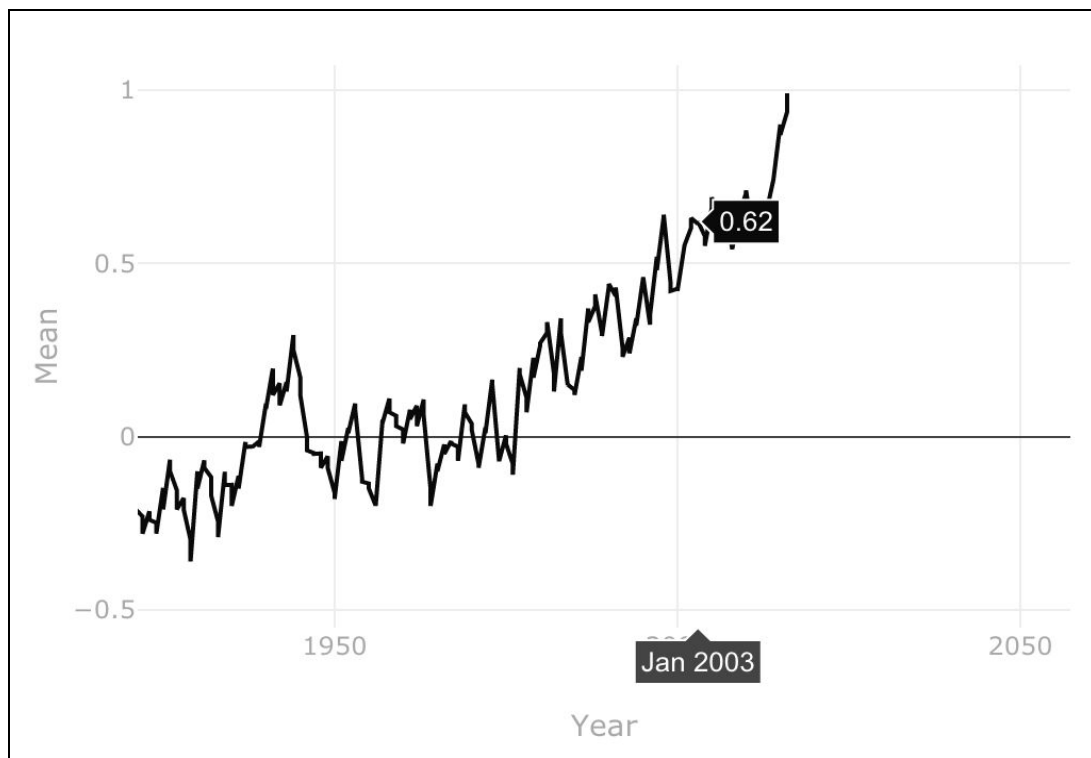


Figure 1.1 Interactive Chart from DataHub<sup>[2]</sup>

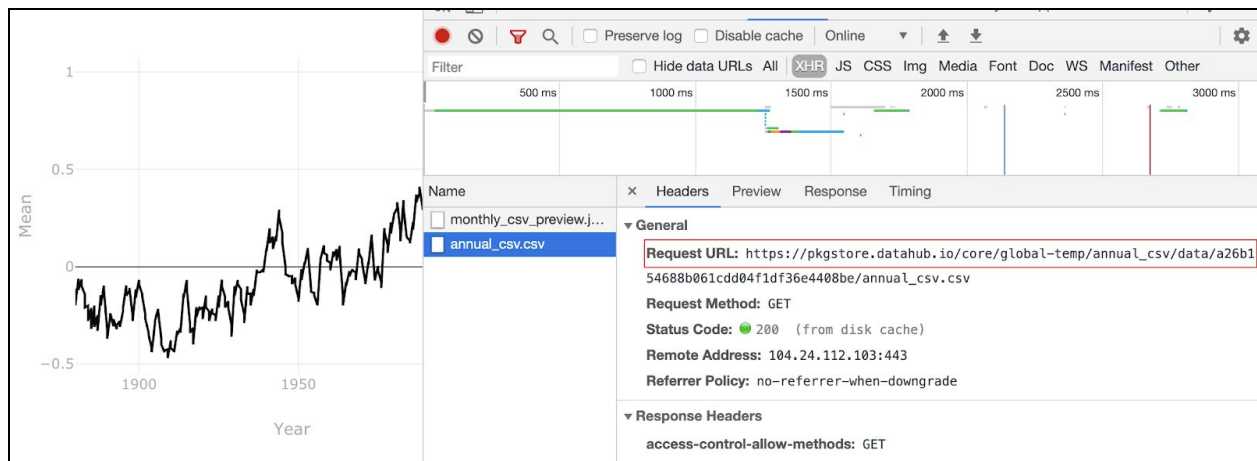
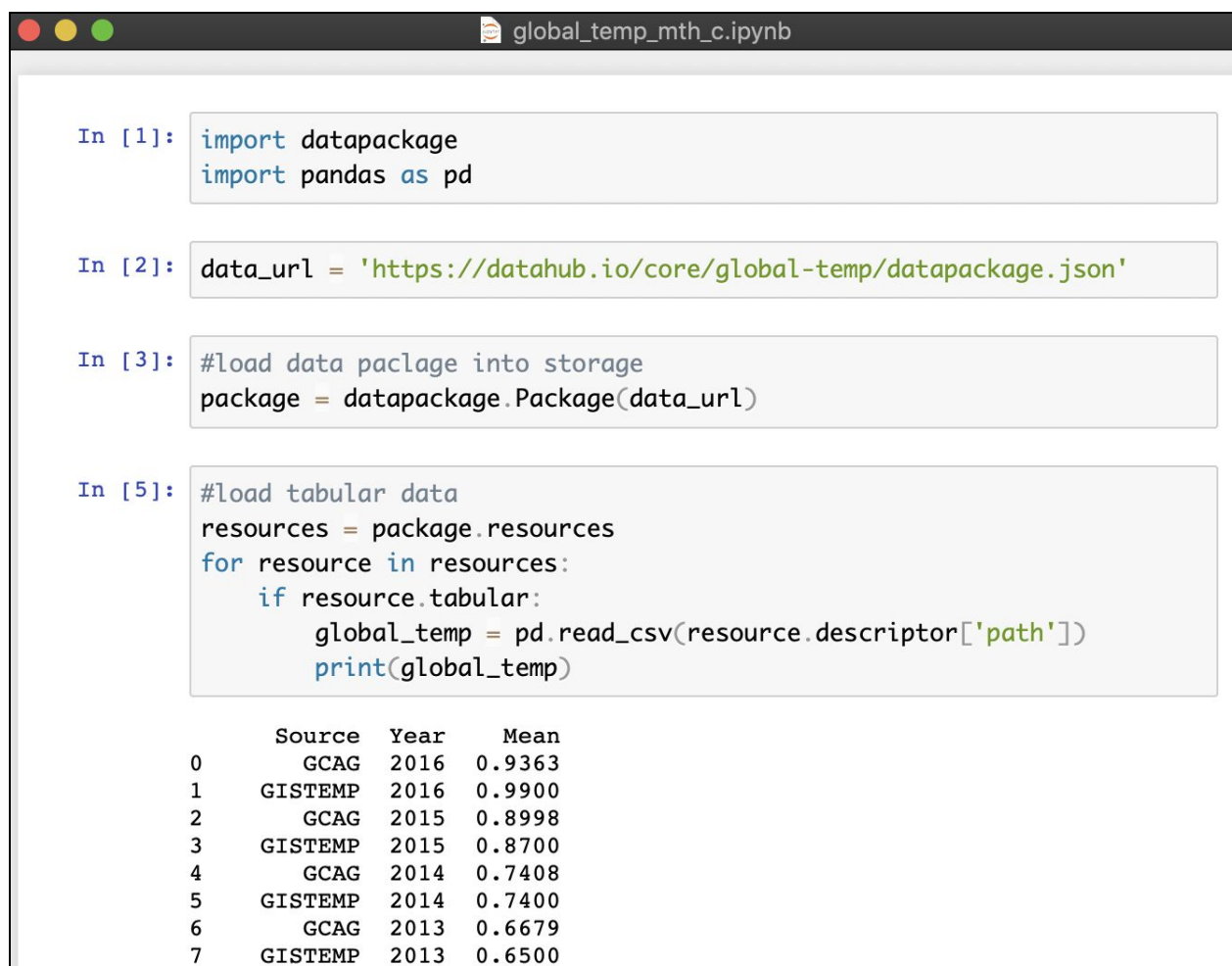


Figure 1.2 Dataset<sup>[3]</sup> request url can be found on the website<sup>2</sup> html element



```
In [1]: import datapackage
import pandas as pd

In [2]: data_url = 'https://datahub.io/core/global-temp/datapackage.json'

In [3]: #load data package into storage
package = datapackage.Package(data_url)

In [5]: #load tabular data
resources = package.resources
for resource in resources:
    if resource.tabular:
        global_temp = pd.read_csv(resource.descriptor['path'])
        print(global_temp)
```

	Source	Year	Mean
0	GCAG	2016	0.9363
1	GISTEMP	2016	0.9900
2	GCAG	2015	0.8998
3	GISTEMP	2015	0.8700
4	GCAG	2014	0.7408
5	GISTEMP	2014	0.7400
6	GCAG	2013	0.6679
7	GISTEMP	2013	0.6500

Figure 1.3 Unpacked data url json file and loaded as csv

```
global_temp_mth_c.ipynb

In [12]: gbl_temp_df = pd.DataFrame(global_temp)
         print(gbl_temp_df)
```

	Source	Date	Mean
0	GCAG	2016-12	0.7895
1	GISTEMP	2016-12	0.8100
2	GCAG	2016-11	0.7504
3	GISTEMP	2016-11	0.9300
4	GCAG	2016-10	0.7292
5	GISTEMP	2016-10	0.8900
6	GCAG	2016-09	0.8767

3283	GISTEMP	1880-03	-0.1800
3284	GCAG	1880-02	-0.1229
3285	GISTEMP	1880-02	-0.2100
3286	GCAG	1880-01	0.0009
3287	GISTEMP	1880-01	-0.3000

[3288 rows x 3 columns]

```
In [10]: gbl_temp_df.to_csv('gbl_temp.csv')
```

Figure 1.4 Saved dataset as csv format

Datasets obtained:

1. Global temperature<sup>[4]</sup> anomaly data come from the Global Historical Climatology Network-Monthly (GHCN-M) data set and International Comprehensive Ocean-Atmosphere Data Set (ICOADS), which have data from 1880 to the present. These two datasets are blended into a single product to produce the combined global land and ocean temperature anomalies. The available timeseries of global-scale temperature anomalies are calculated with respect to the 20th century average, while the mapping tool displays global-scale temperature anomalies with respect to the 1981-2010 base period. For more information on these anomalies, please visit Global Surface Temperature Anomalies<sup>[4]</sup>
2. Global Monthly Mean CO<sub>2</sub><sup>[5]</sup>. Data are reported as a dry air mole fraction defined as the number of molecules of carbon dioxide divided by the number of all molecules in air, including CO<sub>2</sub> itself, after water vapor has been removed. The mole fraction is expressed as parts per million (ppm). Example: 0.000400 is expressed as 400 ppm.<sup>5</sup>

- 
3. Average cumulative mass balance of reference Glaciers worldwide<sup>[6]</sup>. Average cumulative mass balance of “reference” Glaciers worldwide from 1945-2014 sourced from US EPA and the World Glacier Monitoring Service (WGMS). This is cumulative change in mass balance of a set of “reference” glaciers worldwide beginning in 1945. The values represent the average of all the glaciers that were measured. Negative values indicate a net loss of ice and snow compared with the base year of 1945. For consistency, measurements are in meters of water equivalent, which represent changes in the average thickness of a glacier<sup>[6]</sup>.

---

## Transform

In this project, we used pandas tools to clean the data. The process of transformation began by getting the required csv files from the retrieved data. The csv files were then read into a Jupyter Notebook, using pandas `read_csv` function. The next step was to merge Glacier Mass csv, on Year, first with Global Temperature csv, and then with Carbon dioxide Emission csv. Finally, the two csv files were merged to form a comprehensive table. The columns which were not needed were cleaned by using `.drop()` function. The columns were renamed to make them descriptive, by using `.rename()` function.

While extracting the data for global temperature, the website was using two types of data. The first one was the GISS Surface Temperature Analysis (GISTEMP), which is an estimate of global surface temperature change<sup>[7]</sup>. The second one was Climate at a Glance (GCAG), which provides near real-time analysis of monthly and annual temperatures for the globe and is intended for the study of climate variability and change<sup>[8]</sup>. For this project, we used the GCAG data type.

We used the `.fillna()` function to drop the NaN values with regards to integer inputs, which could interfere in the graphical visualizations. The Global Temperature data for the Northern Hemisphere and Southern Hemisphere were added for a better comparison of the data. The data before 1975 was discarded, as we could not verify where the data for the Carbon dioxide emission was derived from. We reviewed and made the necessary changes, such as renaming and dropping the columns to make the table more relevant. Finally, we saved our cleaned data to a new CSV file, ready to be used for loading.



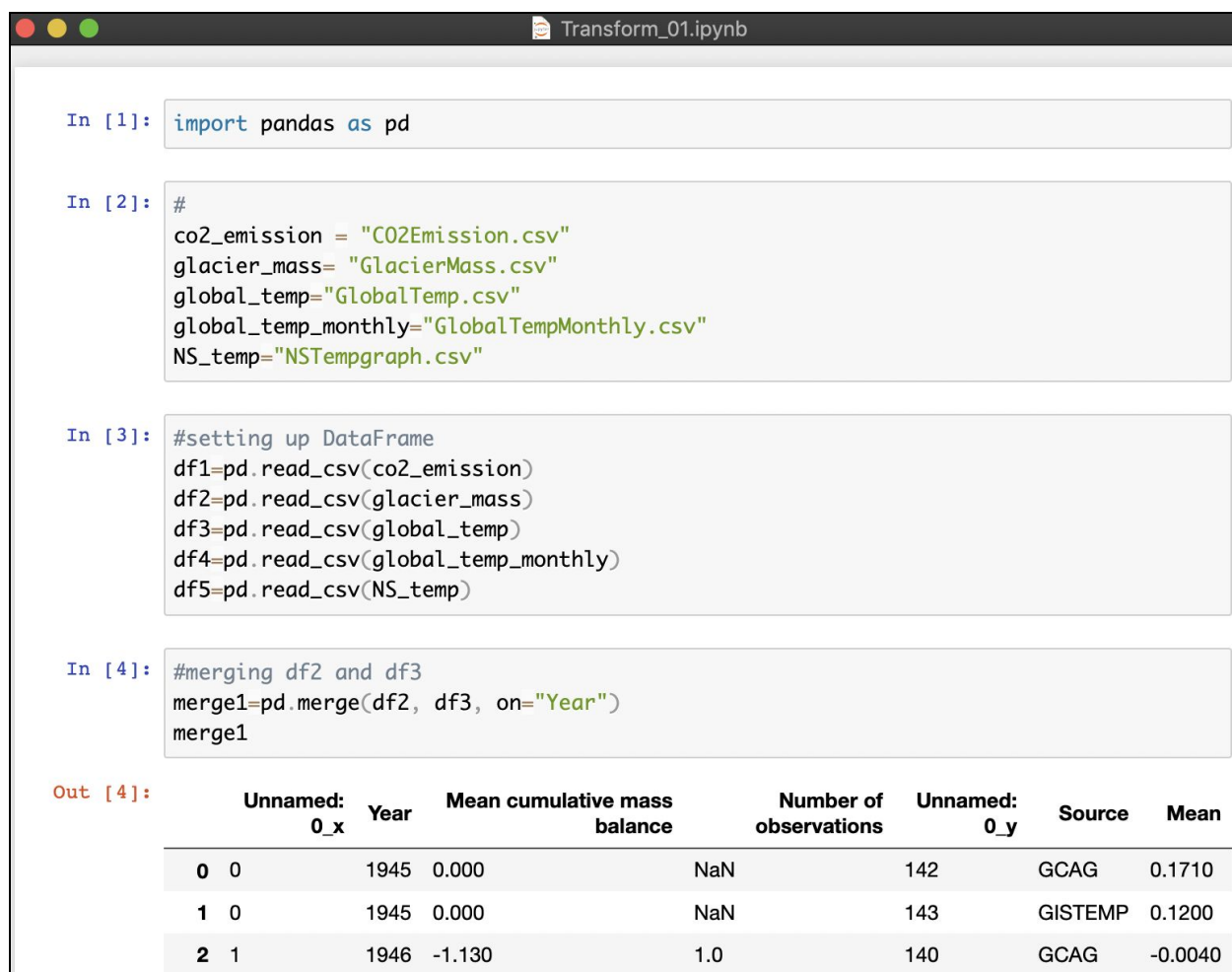


Figure 2.1 Setting up DataFrames from csv data

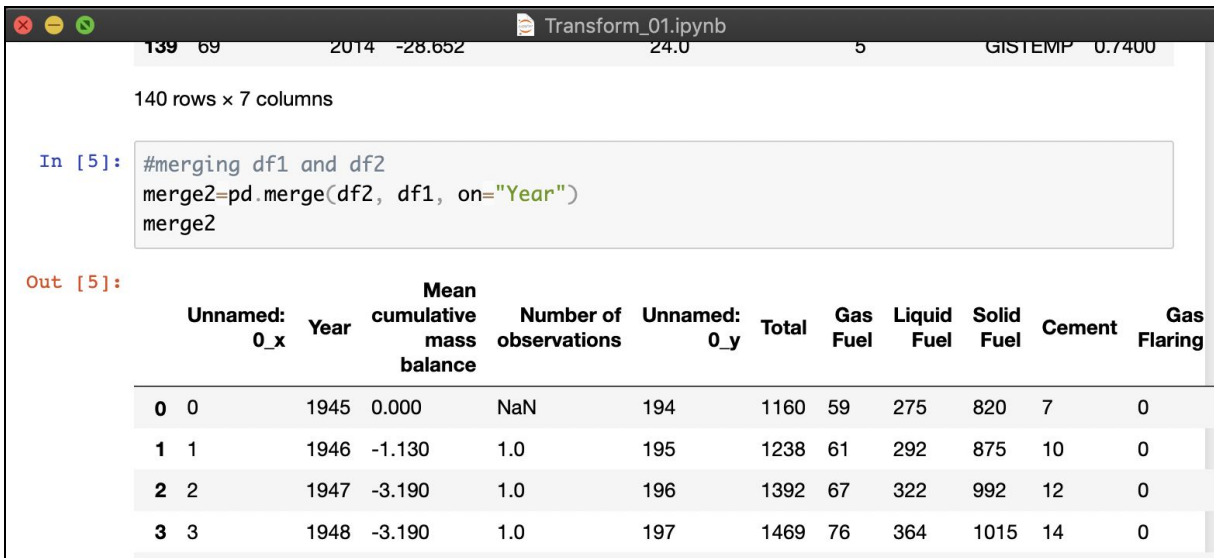


Figure 2.2 Merging DataFrame 1 & 2

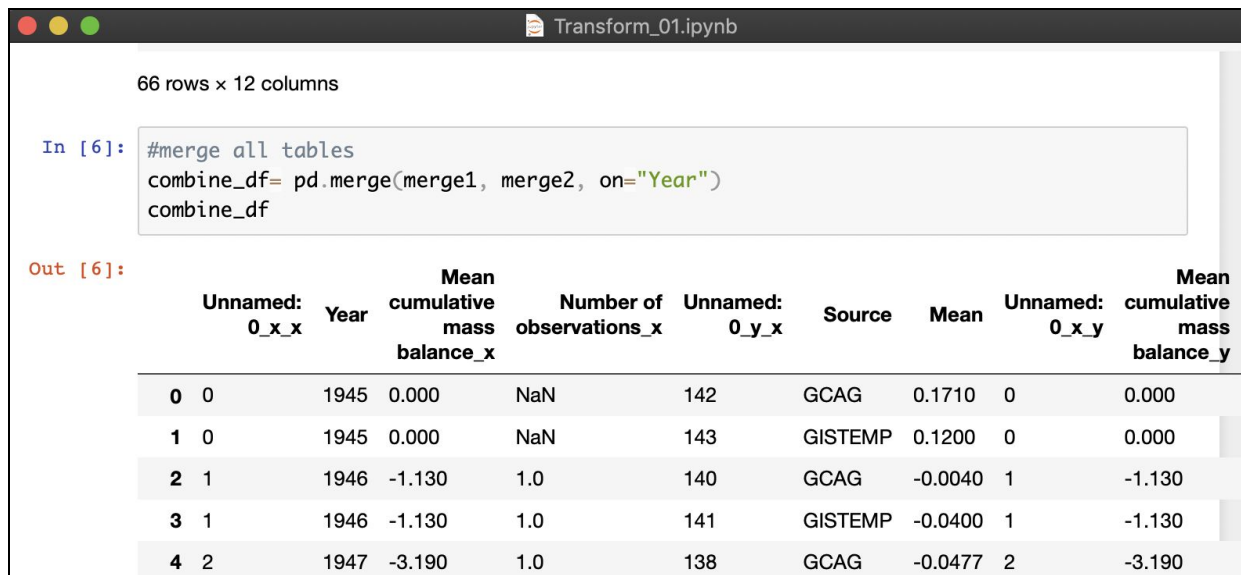


Figure 2.3 Merging all tables

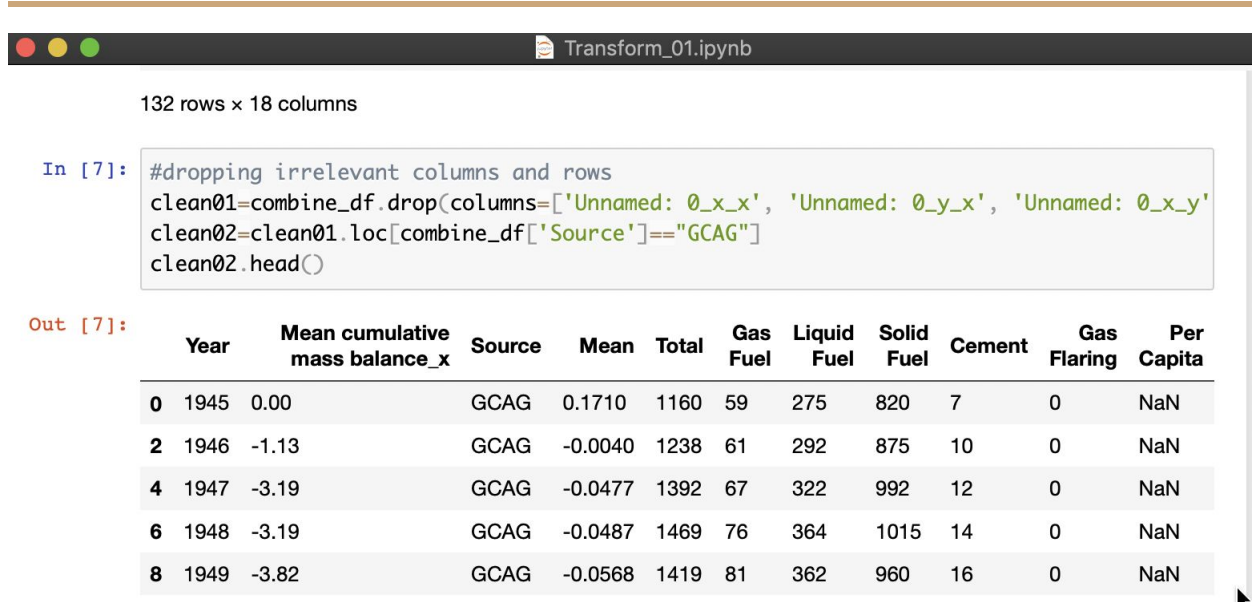


Figure 2.4 Dropping irrelevant columns

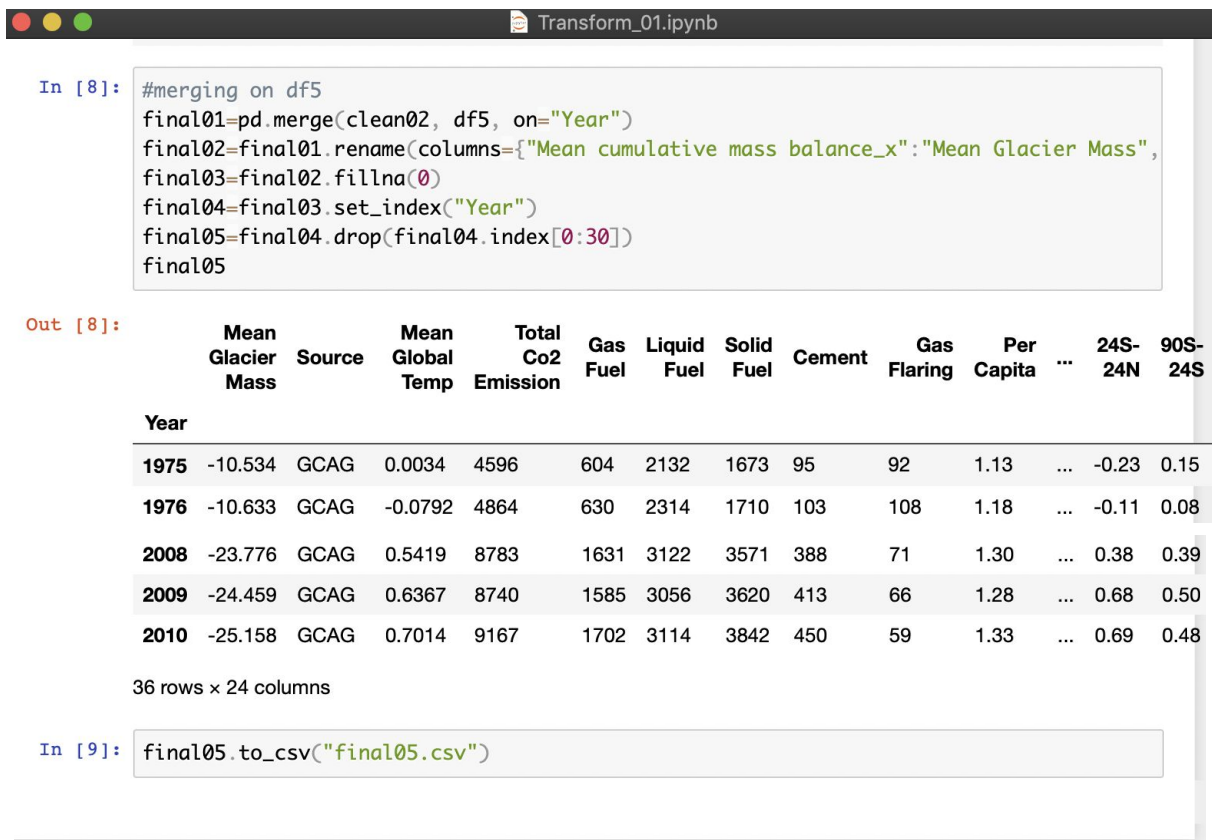
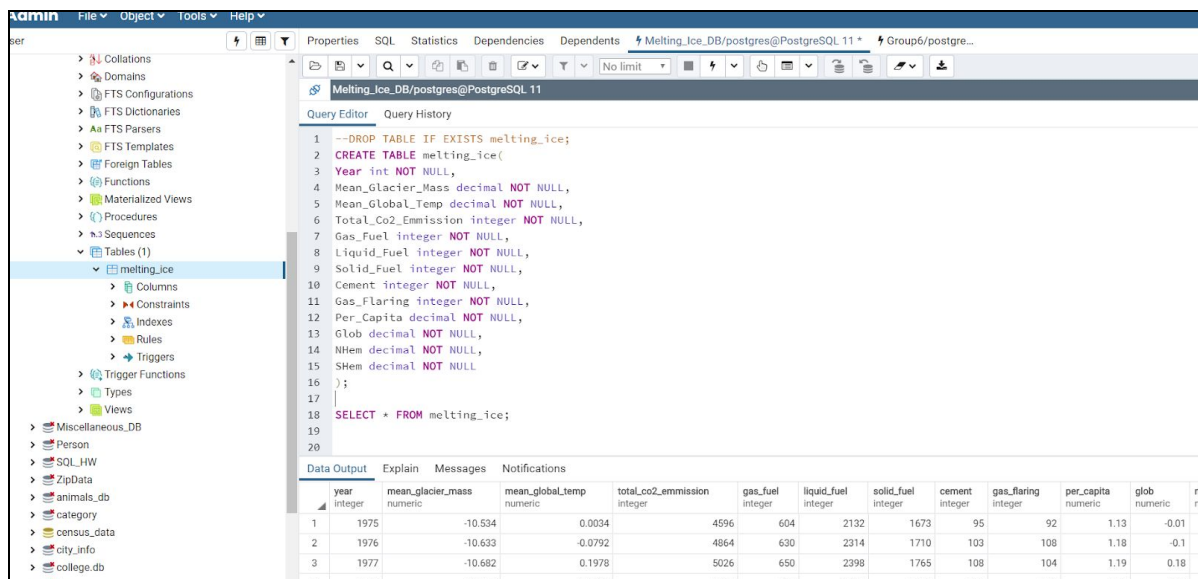


Figure 2.5 Data saved to csv

---

## Load

We used PostgreSQL (relational database) to create a database (Melting\_Ice\_DB) and a respective table (melting\_ice). Then, we cleaned the csv file since we were unable to import the selective columns. We were considering only 13 columns out of 25 to demonstrate the relationship. After that, we imported and loaded our final csv file. Finally, we connected to that database using SQLAlchemy and loaded the result through Jupyter Notebook. Hence, we were able to perform several queries to suit the desired criteria.



The screenshot displays the pgAdmin 4 interface. On the left, the 'Tables (1)' folder is expanded, showing the 'melting\_ice' table. The 'Query Editor' tab is active, showing the following SQL code:

```
1 --DROP TABLE IF EXISTS melting_ice;
2 CREATE TABLE melting_ice(
3 Year int NOT NULL,
4 Mean_Glacier_Mass decimal NOT NULL,
5 Mean_Global_Temp decimal NOT NULL,
6 Total_Co2_Emission integer NOT NULL,
7 Gas_Fuel integer NOT NULL,
8 Liquid_Fuel integer NOT NULL,
9 Solid_Fuel integer NOT NULL,
10 Cement integer NOT NULL,
11 Gas_Flaring integer NOT NULL,
12 Per_Capita decimal NOT NULL,
13 Glob decimal NOT NULL,
14 NHem decimal NOT NULL,
15 SHem decimal NOT NULL
16 );
17
18 SELECT * FROM melting_ice;
```

Below the query editor, the 'Data Output' tab shows the results of the query. The table has 13 columns: year, mean\_glacier\_mass, mean\_global\_temp, total\_co2\_emission, gas\_fuel, liquid\_fuel, solid\_fuel, cement, gas\_flaring, per\_capita, glob, and nhem. The first three rows of data are visible:

	year	mean_glacier_mass	mean_global_temp	total_co2_emission	gas_fuel	liquid_fuel	solid_fuel	cement	gas_flaring	per_capita	glob	nhem
1	1975	-10.534	0.0034	4596	604	2132	1673	95	92	1.13	-0.01	
2	1976	-10.633	-0.0792	4864	630	2314	1710	103	108	1.18	-0.1	
3	1977	-10.682	0.1978	5026	650	2398	1765	108	104	1.19	0.18	

Figure 3.1 Database created with relevant headers

```
# jupyter Data_Load Last Checkpoint: 16 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [1]: # Pandas
import pandas as pd
# SQL Alchemy
from sqlalchemy import create_engine

In [2]: # Creating an engine to connect to the database
engine = create_engine("postgresql://postgres:postgres@localhost:5432/Melting_Ice_DB")

In [3]: # connecting to database
conn = engine.connect()

In [8]: # Query all records in the database
data = pd.read_sql("SELECT * FROM melting_ice", conn)
data.head()

Out[8]:
   year  mean_glacier_mass  mean_global_temp  total_co2_emmission  gas_fuel  liquid_fuel  solid_fuel  cement  gas_flaring  per_capita  glob  nhem  shen
0  1975             -10.534             0.0034             4596         604         2132         1673         95           92         1.13  -0.01  -0.05  0.0
1  1976             -10.633             -0.0792             4864         630         2314         1710        103          108         1.18  -0.10  -0.21  0.0
2  1977             -10.682              0.1978             5026         650         2398         1765        108          104         1.19   0.18   0.12  0.2
3  1978             -10.754              0.1123             5087         680         2392         1793        116          106         1.19   0.07   0.02  0.1
```

Figure 3.2 Screenshot using SQLAlchmey to create Database

## Conclusion

Our team successfully performed all three components of the ETL process. The takeaway from this project are as such:

There are multiple ways to perform data extract, depending on the dataset source and the data analyst's creativity, this step can be streamlined to the following steps for faster and more productive analysis.

Depending upon the data extract, Transforming data requires a lot of attention to detail, to curtail and cleanup raw data, in such a way that the data can be staged for loading into the data warehouse. Transform scripts can be specific to raw data and can be tedious at times.

Loading being the last stage of the ETL process, will require the data to be completed and staged for uploading into the data warehouse. If this process is coded with care, should be able to connect with the previous steps to create a streamlined process from Extract to Transform and Loaded into a DataBase. Later on using the data for queries and other analysis and visualization purposes. Our team found that this project is very beneficial and the ETL is a powerful process for Data Analysis in the working industry.

---

## References:

1. SAS Institute Inc, [https://www.sas.com/en\\_us/insights/data-management/what-is-etl.html](https://www.sas.com/en_us/insights/data-management/what-is-etl.html), Date Accessed: 09.2019
2. Datahub (DataHub.io), 2018, Rufus Pollock, Adam Kariv
3. NOAA National Climatic Data Center (NCDC), global component of Climate at a Glance (GCAG).
4. <https://www.ncdc.noaa.gov/cag/global/data-info>, Date Accessed: 09.2019
5. National Oceanic and Atmospheric Administration, <https://www.esrl.noaa.gov/gmd/ccgg/trends/index.html>, Date Accessed: 09.2019
6. WGMS (World Glacier Monitoring Service). 2015 update to data originally published in: WGMS. 2013. Glacier mass balance bulletin no. 12 (2010–2011). Zemp, M., S.U. Nussbaumer, K. Naegeli, I. Gärtner-Roer, F. Paul, M. Hoelzle, and W. Haeberli (eds.). ICSU (WDS)/IUGG (IACS)/UNEP/UNESCO/WMO. Zurich, Switzerland: World Glacier Monitoring Service. [http://wgms.ch/downloads/wgms\\_2013\\_gmbb12.pdf](http://wgms.ch/downloads/wgms_2013_gmbb12.pdf).
7. <https://data.giss.nasa.gov/gistemp/>, Date Accessed:09.2019
8. <https://www.ncdc.noaa.gov/cag/global/data-info>, Date Accessed:09.2019
9. <https://data.giss.nasa.gov/gistemp/graphs/>