# Technical Report

**Group:** Saheed, Ciara, Santiago

## Data Extracts:

Our group extracted three raw csv files for this project from the [Regional Economic Accounts: Download](https://apps.bea.gov/regional/downloadzip.cfm) page on the website of the US Bureau of Economic Analysis. Specifically, we downloaded the raw datasets using the “Annual Personal Income and Employment by State” option within the Personal Income (State and Local) dropdown menu, and the “Personal Consumption Expenditures (PCE) by State” option within the Personal Consumption Expenditures (PCE) by State dropdown menu.

## Data Cleaning

Our group read each of the three raw csv files into pandas and took the following steps to clean the datasets prior to loading them into our SQL database:

**Income Data (“Income\_AllAreas\_1998\_2021.csv”)**

### Dropped columns:

* **GeoFIPS:** string identifier for each region and state
* **Region**: contains numerical ID code (1-8) denoting the region for each state
* **TableName**: lists the original table name (“SASUMMARY”) from the raw data extract for each row in the dataset
* **LineCode**: extraneous numerical values
* **IndustryClassification**: blank for all rows
* **Unit**: describes the units for each value under the columns containing data for each year. We dropped this column as all the rows we kept have the same units (“millions of current dollars”)
* **Columns for years 1998 – 2018**: dropped these columns as we wanted to narrow down the dataset to include only data from the three most recent years (2019 – 2021).
* **Description**: dropped this column once we had used the loc function to narrow the dataframe down to just the “Personal Income” row for each state.

### Dropped rows:

* Used loc function to drop rows containing national-level data (where **GeoName** column = “United States”) as we are only interested in state-level data for this database
* Used loc function to drop all rows except those pertaining to total state-level personal income (where **Description** column = “Personal Income”)

### Renamed columns:

* Renamed “GeoName” to "us\_state”
* Renamed “2019”, “2020”, and “2021” to “2019\_income”, “2020\_income”, and “2021\_income”, respectively

**Employment Data (“Jobs\_Industry\_AllAreas\_1998\_2021.csv”)**

### Dropped columns:

* **GeoFIPS:** string identifier for each region and state
* **Region**: contains numerical ID code (1-8) denoting the region for each state
* **TableName**: lists the original table name (“SAEMP25N”) from the raw data extract for each row in the dataset
* **LineCode**: extraneous numerical values
* **IndustryClassification**: extraneous values (mix of strings and integers) serving as identifier for each industry
* **Unit**: provides the units “millions of jobs” for each value under the year columns.
* **Columns for years 1998 – 2018**: dropped these columns as we wanted to narrow down the dataset to include only data from the three most recent years (2019 – 2021).

### Dropped rows:

* Used loc function to drop rows containing national-level data (where **GeoName** column = “United States”) as we are interested in state-level data only
* Used loc function to drop rows containing regional-level data/aggregations (where **GeoName** column = “New England” | “Mideast” | “Great Lakes” | “Plains” | “Southeast” | “Southwest” | “Rocky Mountain” | “Far West”) as we are interested in state-level data only

### Renamed Columns:

* Renamed “GeoName” to “us\_state”
* Changed “Description” to “industry”
* Renamed “2019”, “2020”, and “2021” to “2019\_spending”, “2020\_spending”, and “2021\_spending”, respectively

**Expenditures Data (“Expenditures\_Year\_AllAreas\_1997\_2021.csv”)**

### Dropped columns:

* **GeoFIPS:** string identifier for each region and state
* **Region**: contains numerical ID code (1-8) denoting the region for each state
* **TableName**: lists the original table name (“SAPCE1”) from the raw data extract for each row in the dataset
* **LineCode**: extraneous numerical values
* **IndustryClassification**: blank for all rows
* **Unit**: describes the units for each value (“millions of current dollars”) under the columns containing data for each year. All values have the same units.
* **Columns for years 1998 – 2018**: dropped these columns as we wanted to narrow down the dataset to include only data from the three most recent years (2019 – 2021).

### Dropped rows:

* Used loc function to drop rows containing national-level data (where **GeoName** column = “United States”) as we are interested in state-level data only
* Used loc function to drop rows containing regional-level data/aggregations (where **GeoName** column = “New England” | “Mideast” | “Great Lakes” | “Plains” | “Southeast” | “Southwest” | “Rocky Mountain” | “Far West”) as we are interested in state-level data only

### Renamed Columns:

* Renamed “GeoName” to “us\_state”
* Changed “Description” to “spending\_category”
* Renamed “2019”, “2020”, and “2021” to “2019\_spending”, “2020\_spending”, and “2021\_spending”, respectively

After cleaning the three raw datasets in pandas using the steps summarized above, the three cleaned dataframes were written to new csv files entitled:

* **Income\_all\_final\_CLEANED.csv**
* **Job\_industry\_CLEANED.csv**
* **State\_spending\_cleaned.csv**

## Loading Data to PostgreSQL:

To load the cleaned csv files into a Postgres database, we first created a new database in PgAdmin called “Project2” and created the following tables:

* Us\_state\_spending
* Us\_state\_employment
* Us\_state\_income

To create these tables in pgAdmin, we entered the following code in the database Query tool, ensuring that the structure of each table exactly matches that of the corresponding cleaned csv file:

--Create us\_state\_spending table

CREATE TABLE us\_state\_spending (

us\_state TEXT,

spending\_category TEXT,

2019\_spending FLOAT,

2020\_spending FLOAT,

2021\_spending FLOAT

);

--Create us\_state\_employment table

CREATE TABLE us\_state\_employment(

us\_state TEXT,

industry TEXT,

2019\_jobs FLOAT,

2020\_jobs FLOAT,

2021\_jobs FLOAT

);

--Create us\_state\_income table

CREATE TABLE us\_state\_employment(

us\_state TEXT,

2019\_income FLOAT,

2020\_income FLOAT,

2021\_income FLOAT

);

With the three database tables created, we then loaded each of the cleaned csv datasets into the database using the built-in feature in pgAdmin that allows the user to import data directly from files in csv format. Thus, the final database includes three tables containing information on: (i) state-level expenditures on various categories of goods and services; (ii) state-level jobs totals for various industries; and (iii) state-level totals for personal income, all of which are available for users to query for 2019, 2020, and 2021.

We chose this topic and these datasets because of the interesting queries that users would be able to perform. For example, this data could be used to examine the relationship between overall levels of spending and employment in certain industries, or the proportion of total personal income that is spent on different types of goods and services, and how these relationships/trends vary by state from year to year.