ETL Report

Alistair Marsden, Erick S, Jerad Ipsen, Kylee LaPierre

September 6th, 2022

Introduction

There are several different characteristics that define a business. From the technology it uses to the demographics of the owner and employees to where it is located within the United States. In order to see and compare these differences, we pulled data from the Annual Business Survey API from the United States Census Bureau as cited below. The data from this business survey is split into four tables; the company summary, the characteristics of businesses, the characteristics of business owners, and the technology characteristics of businesses. We each worked with a different table and used different steps to extract and transform the data.

Data Sources

Bureau, U. S. C. (2021, October 14). *Annual Business Survey (ABS) apis*. Census.gov. Retrieved September 2, 2022, from https://www.census.gov/data/developers/data-sets/abs.2019.html

Extraction

The Annual Business Survey API requires you to have an API key to pull data from the website. We each signed up for our own API keys and used those to pull the Annual Business Survey tables. The following steps were completed for each table and done before pulling data from the Annual Business Survey API:

1) Acquired an API key from https://www.census.gov/data/developers/guidance/api-user-guide.Help_&_Contact_Us.html
2) Imported the following libraries into Python:
    a) requests, JSON, pandas, matplotlib, and seaborn.
3) Extracted the data using the requests library and the given API key then requested certain columns for each table along with using ‘for=us:*’ to get all data at a national level for each column and ‘for=state:*’ to get all data at a state level for each column.
4) Then formatted the data into a JSON file and read it into a dataframe using the pandas library.

For the company summary table, the following columns were requested:

1) NAME, SEX, RACE_GROUP,EMPSZFI,FIRMPDEMP, and RCPSZFI.

For the business characteristics table, the following columns were requested:

1) GEO_ID, NAME, NAICS2017, NAICS2017_LABEL, SEX_LABEL, RACE_GROUP, RACE_GROUP_LABEL, QDESC_LABEL, QDESC, BUSCHAR_LABEL, BUSCHAR, FIRMPDEMP, FIRMPDEMP_F, RCPPDEMP, RCPPDEMP_F,EMP, EMP_F, PAYANN, and PAYANN_F.

For the business owners characteristics table, the following columns were requested:

1) GEO_ID, NAME, NAICS2017, NAICS2017_LABEL, OWNER_SEX, OWNER_SEX_LABEL, OWNER_ETH, OWNER_ETH_LABEL, OWNER_RACE, OWNER_RACE_LABEL, OWNPDEMP, OWNPDEMP_F, OWNPDEMP_PCT, and OWNPDEMP_PCT_F.

For the technology characteristics table, the following columns were requested:

1) GEO_ID, NAME, NAICS2017, NAICS2017_LABEL, SEX, SEX_LABEL, ETH_GROUP, ETH_GROUP_LABEL, RACE_GROUP, RACE_GROUP_LABEL, VET_GROUP, VET_GROUP_LABEL, NSFSZFI, NSFSZFI_LABEL, YEAR, FIRMPDEMP, FIRMPDEMP_F, RCPPDEMP, RCPPDEMP_F, EMP, EMP_F, PAYANN, PAYANN_F, FACTORS_P, and FACTORS_P_LABEL.

Transformation

Since the columns were chosen for each dataset, we didn't need to remove any. For easier readability by us and other users, we changed the names of all columns and formats of some columns.

For the company summary table, the following steps were taken to transform the data:

1) Since columns are chosen when loading in the data, there was no need to remove any columns.
2) We renamed the NAME, SEX, RACE_GROUP, EMPSZFI, FIRMPDEMP, and RCPSZFI columns to Name, Sex, Race, Size, Count, and Revenue respectively.
3) The values in columns SEX, RACE_GROUP, EMPSZFI, and RCPSZFI were all numbered codes which we turned into meaningful labels using keys from the API:
    a) For the SEX column: 001-Total, 002-Female, 003-Male, 004-Equally Male and Female, 096-Classifiable, and 098-Unclassifiable.
    b) For the RACE_GROUP column: 00-Total, 30-White, 40-Black or African American, 50-American Indian and Alaska Native, 70-Native Hawaiian and

Other Pacific Islander, 90-Minority, 91-Equally Minority and Nonminority, 92-Nonminority, 96-Classifiable, and 98-Unclassifiable.

c) For the EMPSZFI column: 001-Total, 611-None, 612-1 to 4, 620-5 to 9, 630-10 to 19, 641-20 to 49, 642-50 to 99, 651-100 to 249, 652-250 to 499, and 657-500 or More.

d) For the RCPSZFI column: 001-Total, 511-Less than \$5,000, 518-\$5,000 to \$9,999, 519-\$10,000 to \$24,999, 521-\$25,000 to \$49,999, 522-\$50,000 to \$99,999, 523-\$100,000 to \$249,999, 525-\$250,000 to \$499,999, 531-\$500,000 to \$999,999, and 532-\$1,000,000 or More.

i) Backslashes are used so dollar signs would appear in visuals.

4) We then removed rows that included totals for the SEX, RACE_GROUP, EMPSZFI, and RCPSZFI columns as well as Classifiable and Unclassifiable rows for SEX and RACE_GROUP since these wouldn't be useful for the visuals we had in mind.

For the business characteristics table, the following steps were taken to transform the data:

1) For the transformation for Minority and Nonminority employee counts, we created a dataframe from the raw query dataframe consisting of rows where 'QDESC' column values are equal to 'B20' for the types of workers, 'BUSCHAR' values are 'JU' for part-time paid employees, 'RACE_GROUP_LABEL' values are not equal to 'Total', and 'NAICS2017_LABEL' values are equal to 'Total for all sectors'.

a) From that dataframe, we extracted rows where column 'RACE_GROUP_LABEL' values are either 'Minority' or 'Nonminority'.

b) We then changed 'FIRMPDEMP' data type to integer, which is the number of employer firms.

c) Then created a dataframe to be used for the plot which includes the 'NAICS2017_LABEL', 'RACE_GROUP_LABEL', and 'FIRMPDEMP' columns from previous step's dataframe.

   d) Pivot the dataframe using ‘NAICS2017_LABEL’ column as index, ‘RACE_GROUP_LABEL’ as columns.
   e) Rename ‘Nonminority’ column to ‘Non-minority’.
2) For the transformation for Top 5 industries for minority owners dataframe:
   a) Take dataframe for minority/nonminority employee count and sort values by ‘Minority’ values descending.
3) For the transformation for food industry employee counts by minority race chart:
   a) Create a dataframe from the raw query dataframe consisting of: rows where ‘RACE_GROUP_LABEL’ values are not equal to ‘Total’, ‘Unclassifiable’, ‘Classifiable’, ‘Nonminority’, ‘Equally minority/nonminority’, and ‘Minority’; rows where ‘QDESC’ column values are equal to ‘B01’; rows where ‘NAICS2017_LABEL’ values are equal to ‘Accommodation and food services’; rows where ‘BUSCHAR_LABEL’ values are equal to ‘All firms’
   b) Change ‘EMP’ data type to integer
4) For the transformation for healthcare employee counts by minority race chart:
   a) Create a dataframe from the raw query dataframe consisting of: rows where ‘RACE_GROUP_LABEL’ values are not equal to ‘Total’, ‘Unclassifiable’, ‘Classifiable’, ‘Nonminority’, ‘Equally minority/nonminority’, and ‘Minority’; rows where ‘QDESC’ column values are equal to ‘B01’; rows where ‘NAICS2017_LABEL’ values are equal to ‘’Health care and social assistance’; rows where ‘BUSCHAR_LABEL’ values are equal to ‘All firms’
   b) Change ‘EMP’ data type to integer
5) For the transformation for retail trade employee counts by minority race chart:
   a) Create a dataframe from the raw query dataframe consisting of: rows where ‘RACE_GROUP_LABEL’ values are not equal to ‘Total’, ‘Unclassifiable’, ‘Classifiable’, ‘Nonminority’, ‘Equally minority/nonminority’, and ‘Minority’; rows where ‘QDESC’ column values are equal to ‘B01’; rows where ‘NAICS2017_LABEL’ values are equal to ‘Retail Trade’; rows where ‘BUSCHAR_LABEL’ values are equal to ‘All firms’

    b) Change ‘EMP’ data type to integer
6) For the transformation for amount of owners for Asian food industry firms pie chart:
    a) Create a dataframe from the raw query dataframe consisting of rows where ‘QDESC’ column values are equal to ‘B01’, ‘RACE_GROUP_LABEL’ column values are equal to ‘Asian’, and where ‘NAICS2017_LABEL’ values are equal to ‘Accommodation and food services’. Exclude rows where ‘BUSCHAR_LABEL’ values are equal to ‘All firms’, ‘Total reporting’, ‘Item not reported’, or ‘Unknown number of owners’
7) For the transformation for family and non-family owned Asian food industry firms pie chart:
    a) Create a dataframe from the raw query dataframe consisting of rows where ‘QDESC_LABEL’ column values are equal to ‘FAMOWN’, ‘RACE_GROUP_LABEL’ column values are equal to ‘Asian’. Exclude rows where ‘BUSCHAR_LABEL’ values are equal to ‘All firms’, ‘Total reporting’, ‘Item not reported’, or ‘Unknown number of owners’

For the characteristics of business owners table, the following steps were taken to transform the data:

1) We removed the GEO_ID, NAICS2017, OWNER_SEX, OWNER_ETH, OWNER_ETH_LABEL, OWNER_RACE, OWNPDEMP_F, OWNDEMP_PCT_F, QDESC_LABEL, and us columns after realizing we did not need the information located within these columns.
2) We then renamed the NAME column to “Country”, the NAICS2017_LABEL column to “Employer Firm Sector”, the OWNER_SEX_LABEL column to “Sex”, the OWNER_RACE_LABEL column to “Race”, the OWNPDEMP column to “Number of Owners”, and the OWNPDEMP_PCT column to “Percent of Owners”.
3) Then, we changed the “Number of Owners” column datatype to “int64” and changed the “Percent of Owners” column datatype to “float”.

4) We then created a graph based on the number of owners for each sex.
    a) We grouped my data based on the “Sex” and found the sum of the owners.
    b) We removed all rows that contained “All owners of respondent firms” since We wanted to view each sex separately.
    c) We used seaborn to create a bar chart based on the grouped by sex data with the number of owners on the y axis and sex on the x axis.
5) Then created graph based on the number of owners for each employer firm sector.
    a) We grouped my data based on the “Employer Firm Sector” column and found the sum of the owners.
    b) We removed all rows that contained “Total for all sectors” as I wanted to break down each sector individually.
    c) We then sorted the values from the greatest number of owners to the lowest and only kept the top ten results.
    d) We then used matplotlib.pyplot to create a horizontal bar chart that would show the top ten firm sectors by the number of owners.
6) Then created graph based on the number of owners for each employer firm sector based on sex.
    a) We grouped my data based on both the “Sex” and “Employer Firm Sector” columns and found the sum of the owners.
    b) We removed all rows that contained “All owners of respondent firms” since we wanted to view each sex separately.
    c) We then sorted the values from the greatest number of owners to the lowest and only kept the top ten results that were not “Total for all sectors”([2:12] of the sorted values).
    d) We then used matplotlib.pyplot to create a horizontal bar chart that would show the top ten firm sectors by the number of owners based on gender.F

For the technology characteristics table, the following steps were taken to transform the data:

1) First, we renamed the columns from 'GEO_ID' to 'GeoID' ,'NAME' to 'Region', 'NAICS2017' to 'IndustryID', 'NAICS2017_LABEL' to 'Industry', 'SEX' to 'SexID', 'SEX_LABEL' to 'Sex', 'ETH_GROUP' to 'EthnicityID', 'ETH_GROUP_LABEL' to 'Ethnicity', 'RACE_GROUP' to 'RaceID', 'RACE_GROUP_LABEL' to 'Race', 'NSFSZFI' to 'FirmSizeID', 'NSFSZFI_LABEL' to 'FirmSize', 'YEAR' to 'Year', 'FIRMPDEMP' to 'FirmTotal', 'FIRMPDEMP_F' to 'FirmTotalID', 'RCPPDEMP' to 'Revenue', 'RCPPDEMP_F' to 'RevenueID', 'EMP' to 'EmployeeTotal', 'EMP_F' to 'EmployeeTotalID', 'PAYANN to :'AnnualPayroll', 'PAYANN_F' to 'AnnualPayrollID' ,'FACTORS_P' to 'ProductionFactorsID', and 'FACTORS_P_LABEL' to 'ProductionFactors'.
2) Then we changed the type of the ‘FirmSize’, ‘FirmTotal’, ‘Revenue’, ‘EmployeeTotal’, and ‘AnnualPay’ columns to ‘int’.
3) We were interested in seeing the different production factors and what impact they had on firm size, employee total, and revenue so we subsetted the ‘ProductionFactors’ column into 5 separate data frames.
    a) For each production factor (artificial intelligence, cloud-based, robotics, specialized software, and specialized equipment):
        i) We filtered through the column ‘ProductionFactor’ to find the rows that contained each production factor as an answer and the answer that didn’t have ‘Total Reporting’ in it, then created a new data frame with only those answers in the ‘ProductionFactor’ column with 5 new data frames.
        ii) The steps for filtering out each answer:
            (1) We first created a data frame with columns equal to the original columns from the dataset.
            (2) Then we used a for loop to go through each set of answers within the ‘ProductionFactors’ column. If the answer contained the name of the production factor (artificial intelligence, cloud-based, robotics, specialized software, or

specialized equipment) and it did not contain ‘Total Reporting’, then the index of that row was appended to a list named ‘techindex’.

(3) Once the for loop finished, we then loop through techindex to find the location of the row in the original data frame and append it on to our new data frame.

(4) Finally, we dropped the columns 'GeoID', 'IndustryID', 'SexID', 'EthnicityID', 'RaceID', 'VeteranID', 'FirmSizeID', 'Year', 'FirmTotalID', 'RevenueID', 'EmployeeTotalID', 'AnnualPayrollID', and 'ProductionFactorsID'.

iii) Each new data frame has either artificial intelligence, cloud-based, robotics, specialized software, or specialized equipment as the answer in the ‘ProductionFactor’ column and each data frame is named as such.

Table Mapping:

Company summary table:

Source table columns	***Destination Table Columns***
NAME	Name
SEX_LABEL	Sex
RACE_GROUP	Race
EMPSZFI	Size
FIMRPDEMP	Count
RCPSZFI	Revenue

Business characteristics table:

Source table columns	***Destination Table Columns***
QDESC	QuestionDescription
BUSCHAR	BusinessCode
RACE_GROUP_LABEL	Race
NAICS2017_LABEL	Industry
FIRMPDEMP	Revenue
BUSCHAR_LABEL	BusinessLabel
EMP	EmployeeTotal

Business owner characteristics table:

Source table columns	***Destination Table Columns***
NAME	Country
NAICS2017_LABEL	Employer_Firm_Sector
OWNER_SEX_LABEL	Sex
OWNER_RACE_LABEL	Race
OWNPDEMP	Number_of_Owners
OWNPDEMP_PCT	Percent_of_Owners

Technology characteristics table:

Source table columns	***Destination Table Columns***
NAME	Region
NAICS2017_LABEL	Industry

SEX_LABEL	Sex
ETH_GROUP_LABEL	Ethnicity
RACE_GROUP_LABEL	Race
NSFSZFI_LABEL	FirmSize
YEAR	Year
FIRMPDEMP	FirmTotal
RCPPDEMP	Revenue
EMP	EmployeeTotal
PAYANN	AnnualPayroll
FACTORS_P_LABEL	ProductionFactors

Load

If you were to load your transformed data into a SQL database, what steps would you take to make that happen? Be sure to number steps when the order matters.

Before loading the data frames into SQL, we first need to create the tables that the data will populate. Each table in SQL will follow the same format as the tables above for the columns.

For the company summary table, the columns should be:

1) Name as VARCHAR(50)
2) Sex VARCHAR(50)
3) Race VARCHAR(50)
4) Size INT or BIGINT
5) Count INT or BIGINT
6) Revenue INT or BIGINT

For the business characteristics table, the columns should be:

1) QuestionDescription VARCHAR(50)
2) BusinessCode VARCHAR(50)
3) Race VARCHAR(50)
4) Industry VARCHAR(50)
5) Revenue INT or BIGINT
6) BusinessLabel VARCHAR(50)
7) EmployeeTotal INT or BIGINT

For the business owner characteristics table, the columns should be:

1) Country VARCHAR(50)
2) Employer_Firm_Sector VARCHAR(50)
3) Sex VARCHAR(50)
4) Race VARCHAR(50)
5) Number_of_Owners INT OR BIGINT
6) Percent_of_Owners FLOAT

For the technology characteristics table, the columns should be:

1) Region VARCHAR(50)
2) Industry VARCHAR(50)
3) Sex VARCHAR(50)
4) Ethnicity VARCHAR(50)
5) Race VARCHAR(50)
6) FirmSize INT or BIGINT
7) Year INT or BIGINT
8) FirmTotal INT or BIGINT

9) Revenue INT or BIGINT
10) EmployeeTotal INT or BIGINT
11) AnnualPayroll INT or BIGINT
12) ProductionFactors VARCHAR(50)

To load the data frames into SQL, first use the pandas operator ‘to_csv’ to write each data frame to its own CSV file. Then save each data frame CSV with unique names like ‘company-summary.csv’ or ‘tech-characteristics.csv’ to understand what is in each file.

1) After saving each data frame to its own file, open the SQL document you want to load the data into.
2) Then right-click on the selected database to import a new table.
3) Select ‘Browse’ and find the files that were saved earlier. Rename the table to something other than your database tables.
4) Then click next in the bottom right and make sure that all the columns from the source table are there.
5) Change any data types to match the source information.
6) Then click “Import Data” in the bottom right and make sure it loads properly.

Conclusion

After transforming and using the data in Python, we created our visualizations in order to answer the questions we came up with after looking through the data tables and documents.