Job Security and the Risk of Automation

August 10, 2023

Contents

300 Million Jobs Will Be Lost Or Degraded By Artificial Intelligence

- Goldman Sachs Global Economics Analyst, Mar 23, 2023

THE FUTURE OF EMPLOYMENT: How Susceptible Are Jobs to Computerization?

Written by Carl Benedikt Frey & Michael A. Osborne

Published 2013, University of Oxford

Study evaluates 702 occupations using a probabilistic classification algorithm.

The authors conclude that most workers in transportation and logistics, the bulk of admin support workers, the entire service industry AND laborers in production facilities, are at high risk.

Quote: “We make no attempt to estimate how many jobs will actually be automated.”

This is the question we will answer today.

Data Transformation

Source of Raw Data: Labor Statistics

- Website: U.S. Bureau of Labor Statistics
- Downloaded as .xlsx file
- Number of Rows: 818,972

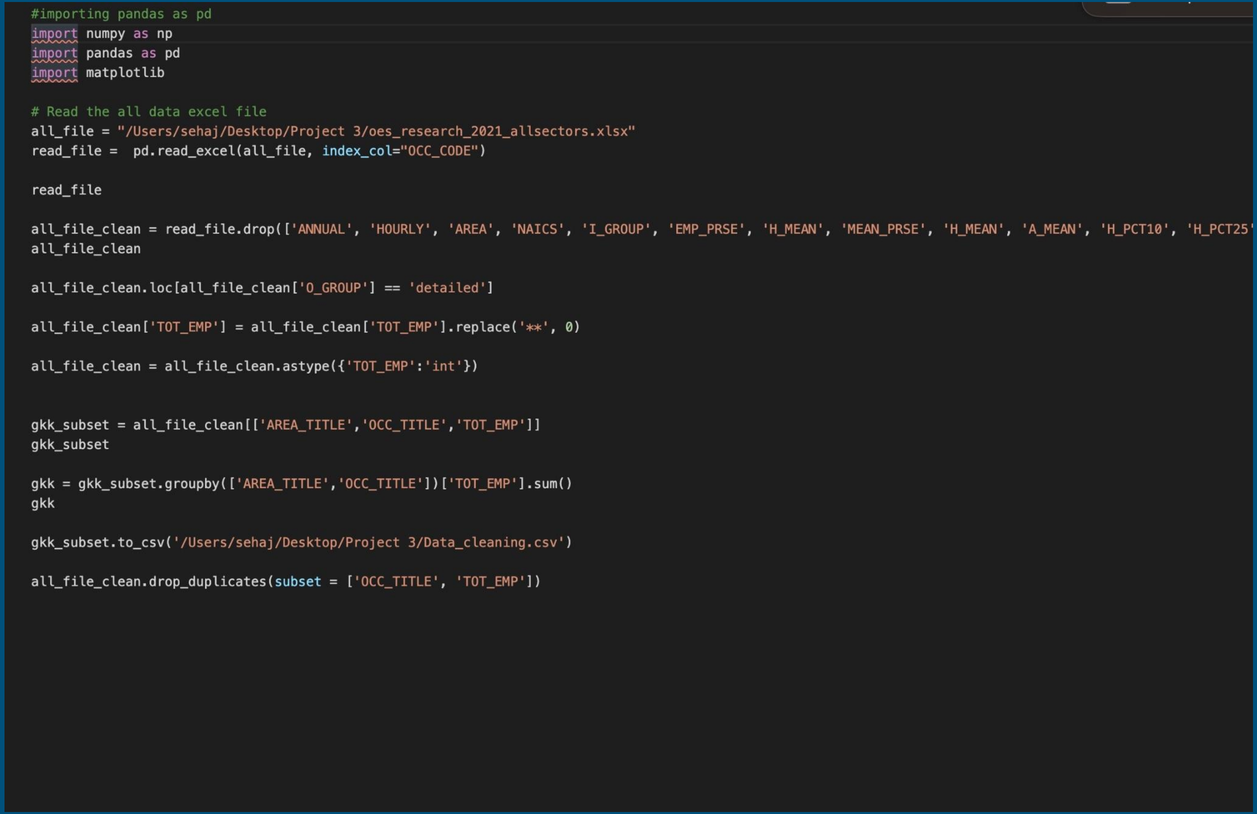
Source of Raw Data: Probability Data

- Book: The Future of Employment (2013)
- Carl Benedikt Frey & Michael A. Osborne
- Downloaded as .csv file from Kaggle
- Transcribed from digital media

Files were imported to MongoDB and thoroughly scrubbed.

- Filtered OCC_GROUP
- Grouped by State and OCC_TITLE

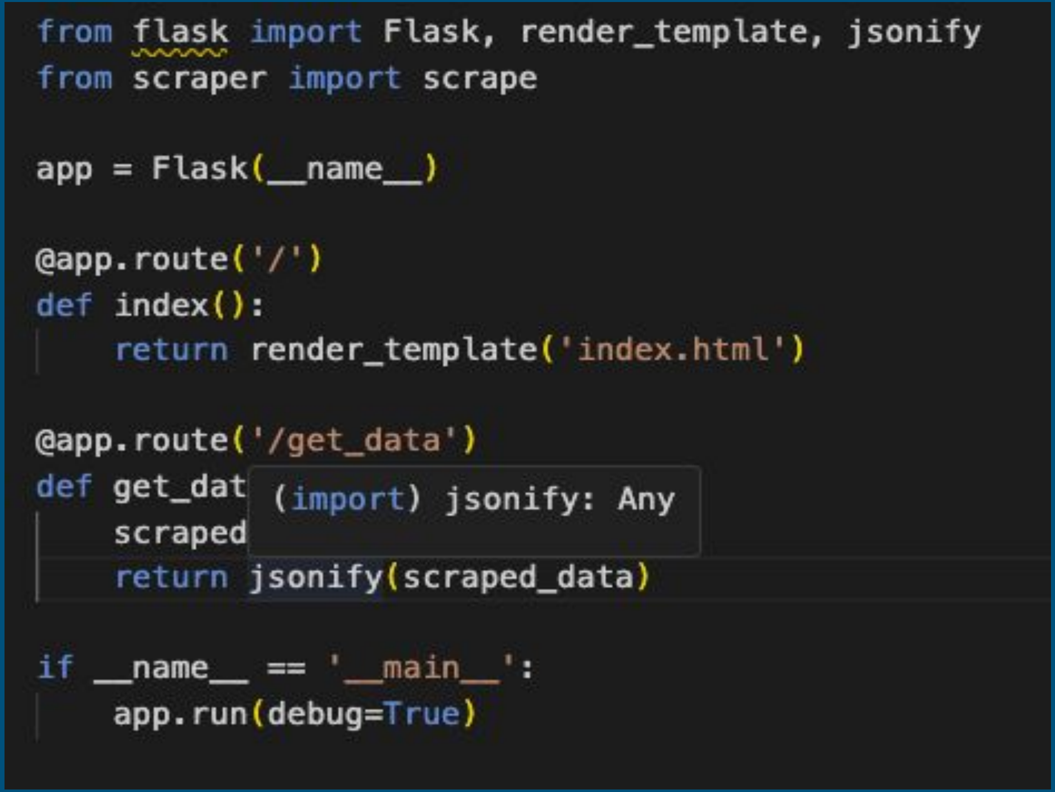
Data cleaning



```python
#importing pandas as pd
import numpy as np
import pandas as pd
import matplotlib

# Read the all data excel file
all_file = "/Users/sehaj/Desktop/Project 3/oes_research_2021_allsectors.xlsx"
read_file =  pd.read_excel(all_file, index_col="OCC_CODE")

read_file

all_file_clean = read_file.drop(['ANNUAL', 'HOURLY', 'AREA', 'NAICS', 'I_GROUP', 'EMP_PRSE', 'H_MEAN', 'MEAN_PRSE', 'H_MEAN', 'A_MEAN', 'H_PCT10', 'H_PCT25'
all_file_clean

all_file_clean.loc[all_file_clean['O_GROUP'] == 'detailed']

all_file_clean['TOT_EMP'] = all_file_clean['TOT_EMP'].replace('**', 0)

all_file_clean = all_file_clean.astype({'TOT_EMP':'int'})


gkk_subset = all_file_clean[['AREA_TITLE','OCC_TITLE','TOT_EMP']]
gkk_subset

gkk = gkk_subset.groupby(['AREA_TITLE','OCC_TITLE'])['TOT_EMP'].sum()
gkk

gkk_subset.to_csv('/Users/sehaj/Desktop/Project 3/Data_cleaning.csv')

all_file_clean.drop_duplicates(subset = ['OCC_TITLE', 'TOT_EMP'])
```

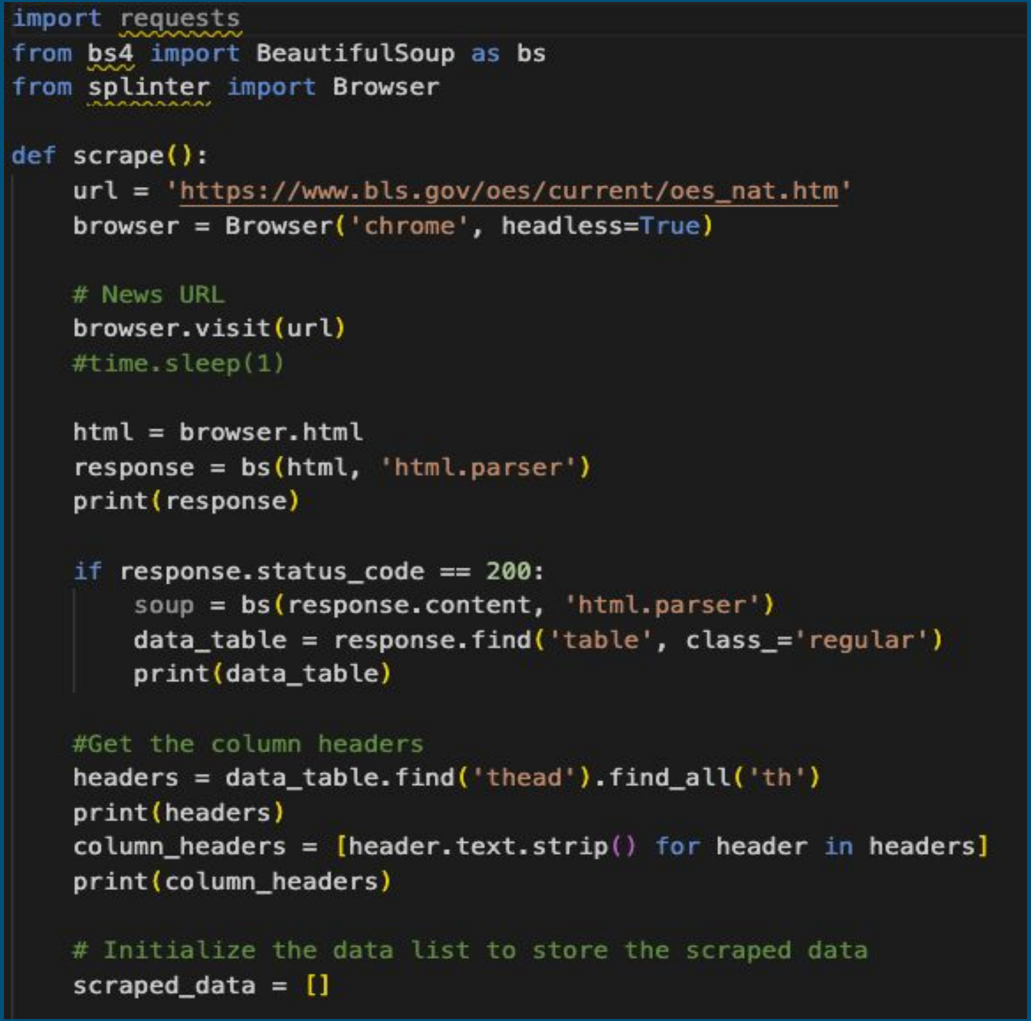
Web Scraping: Lessons Learned

Flask



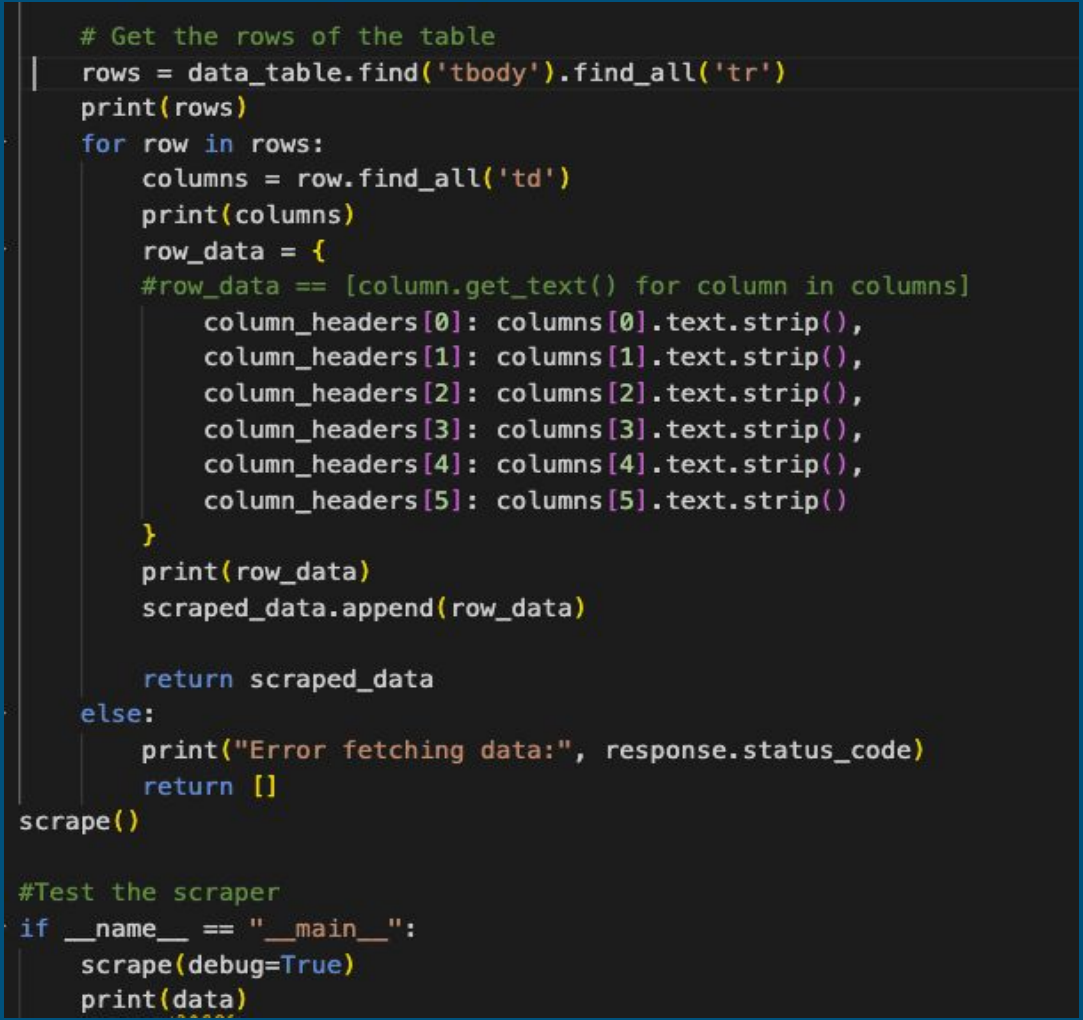
```
from flask import Flask, render_template, jsonify
from scraper import scrape

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/get_data')
def get_dat (import) jsonify: Any
    scraped
    return jsonify(scraped_data)

if __name__ == '__main__':
    app.run(debug=True)
```

scraper.py



```python
import requests
from bs4 import BeautifulSoup as bs
from splinter import Browser

def scrape():
    url = 'https://www.bls.gov/oes/current/oes_nat.htm'
    browser = Browser('chrome', headless=True)

    # News URL
    browser.visit(url)
    #time.sleep(1)

    html = browser.html
    response = bs(html, 'html.parser')
    print(response)

    if response.status_code == 200:
        soup = bs(response.content, 'html.parser')
        data_table = response.find('table', class_='regular')
        print(data_table)

    #Get the column headers
    headers = data_table.find('thead').find_all('th')
    print(headers)
    column_headers = [header.text.strip() for header in headers]
    print(column_headers)

    # Initialize the data list to store the scraped data
    scraped_data = []
```

scraper.py



```python
    # Get the rows of the table
    rows = data_table.find('tbody').find_all('tr')
    print(rows)
    for row in rows:
        columns = row.find_all('td')
        print(columns)
        row_data = {
        #row_data == [column.get_text() for column in columns]
            column_headers[0]: columns[0].text.strip(),
            column_headers[1]: columns[1].text.strip(),
            column_headers[2]: columns[2].text.strip(),
            column_headers[3]: columns[3].text.strip(),
            column_headers[4]: columns[4].text.strip(),
            column_headers[5]: columns[5].text.strip()
        }
        print(row_data)
        scraped_data.append(row_data)

        return scraped_data
    else:
        print("Error fetching data:", response.status_code)
        return []
scrape()

#Test the scraper
if __name__ == "__main__":
    scrape(debug=True)
    print(data)
```

Data Visualizations

Total Job loss by Occupation

including 5 states

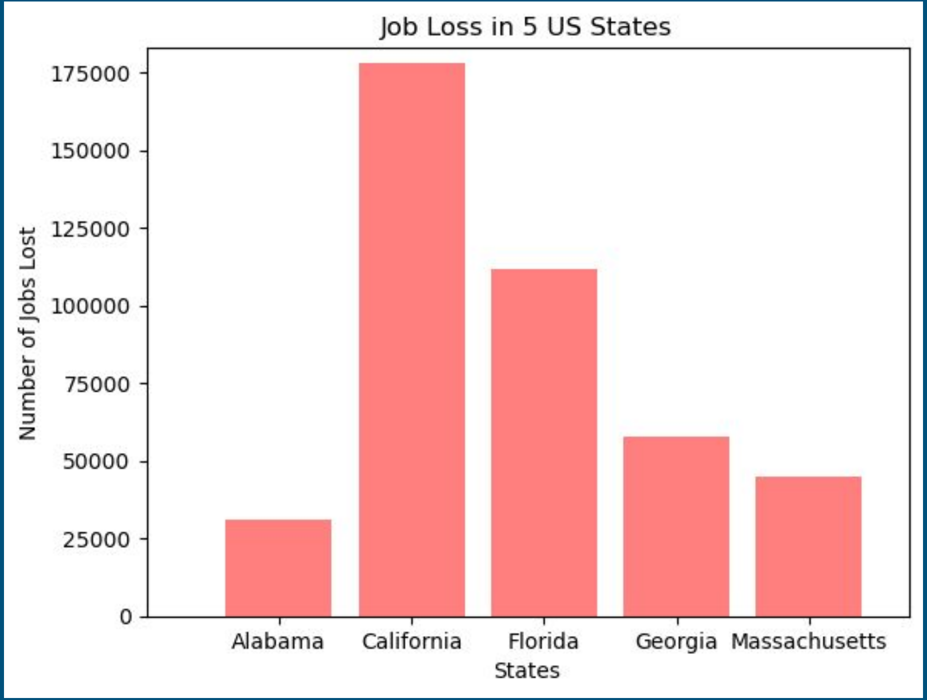
Alabama

California

Florida

Georgia

Massachusetts



Data Visualizations

Total Job loss by Occupation

including 5 states

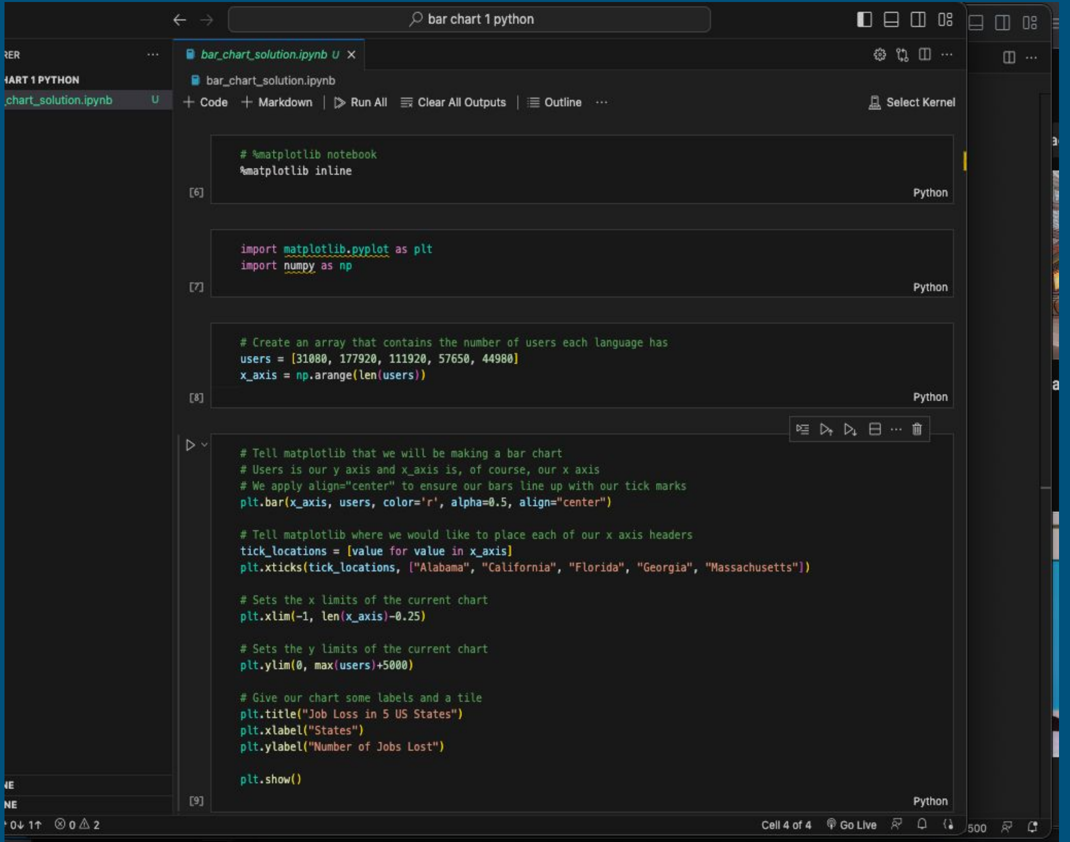
Alabama

California

Florida

Georgia

Massachusetts



```
# %matplotlib notebook
%matplotlib inline
```
[6]

```
import matplotlib.pyplot as plt
import numpy as np
```
[7]

```
# Create an array that contains the number of users each language has
users = [31080, 177920, 111920, 57650, 44980]
x_axis = np.arange(len(users))
```
[8]

```
# Tell matplotlib that we will be making a bar chart
# Users is our y axis and x_axis is, of course, our x axis
# We apply align="center" to ensure our bars line up with our tick marks
plt.bar(x_axis, users, color='r', alpha=0.5, align="center")

# Tell matplotlib where we would like to place each of our x axis headers
tick_locations = [value for value in x_axis]
plt.xticks(tick_locations, ["Alabama", "California", "Florida", "Georgia", "Massachusetts"])

# Sets the x limits of the current chart
plt.xlim(-1, len(x_axis)-0.25)

# Sets the y limits of the current chart
plt.ylim(0, max(users)+5000)

# Give our chart some labels and a tile
plt.title("Job Loss in 5 US States")
plt.xlabel("States")
plt.ylabel("Number of Jobs Lost")

plt.show()
```
[9]

Data Visualizations

Total Job loss by Occupation

including 5 states

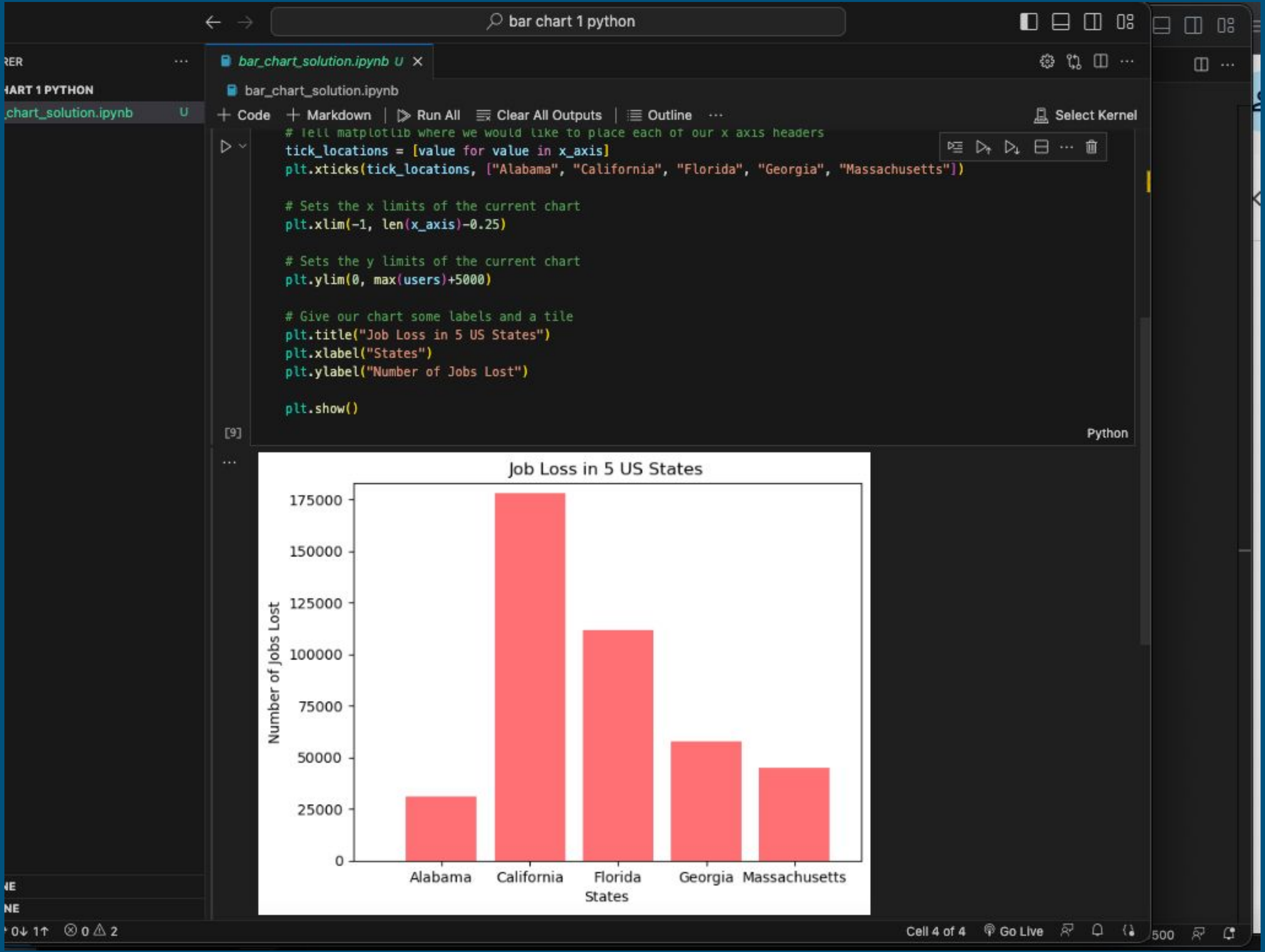
Alabama

California

Florida

Georgia

Massachusetts

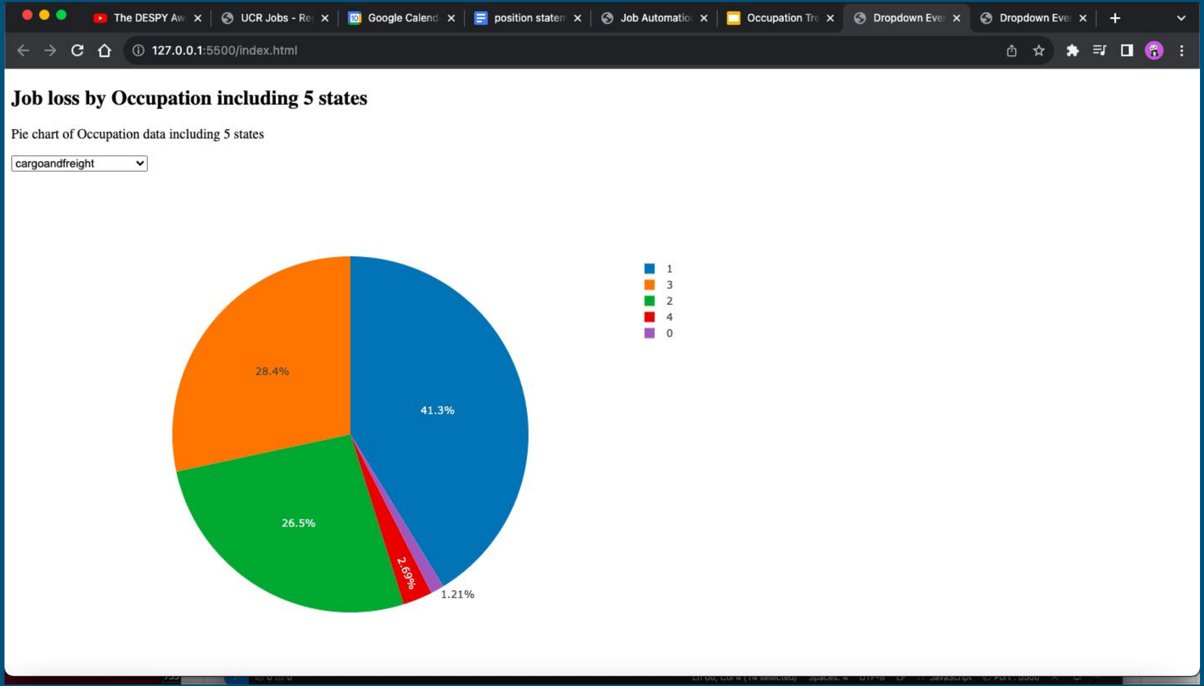


```python
# Tell matplotlib where we would like to place each of our x axis headers
tick_locations = [value for value in x_axis]
plt.xticks(tick_locations, ["Alabama", "California", "Florida", "Georgia", "Massachusetts"])

# Sets the x limits of the current chart
plt.xlim(-1, len(x_axis)-0.25)

# Sets the y limits of the current chart
plt.ylim(0, max(users)+5000)

# Give our chart some labels and a tile
plt.title("Job Loss in 5 US States")
plt.xlabel("States")
plt.ylabel("Number of Jobs Lost")

plt.show()
```

Data Visualizations

Job loss by Occupation including 5 states

http://127.0.0.1:5500/index.html


127.0.0.1:5500/index.html

Job loss by Occupation including 5 states

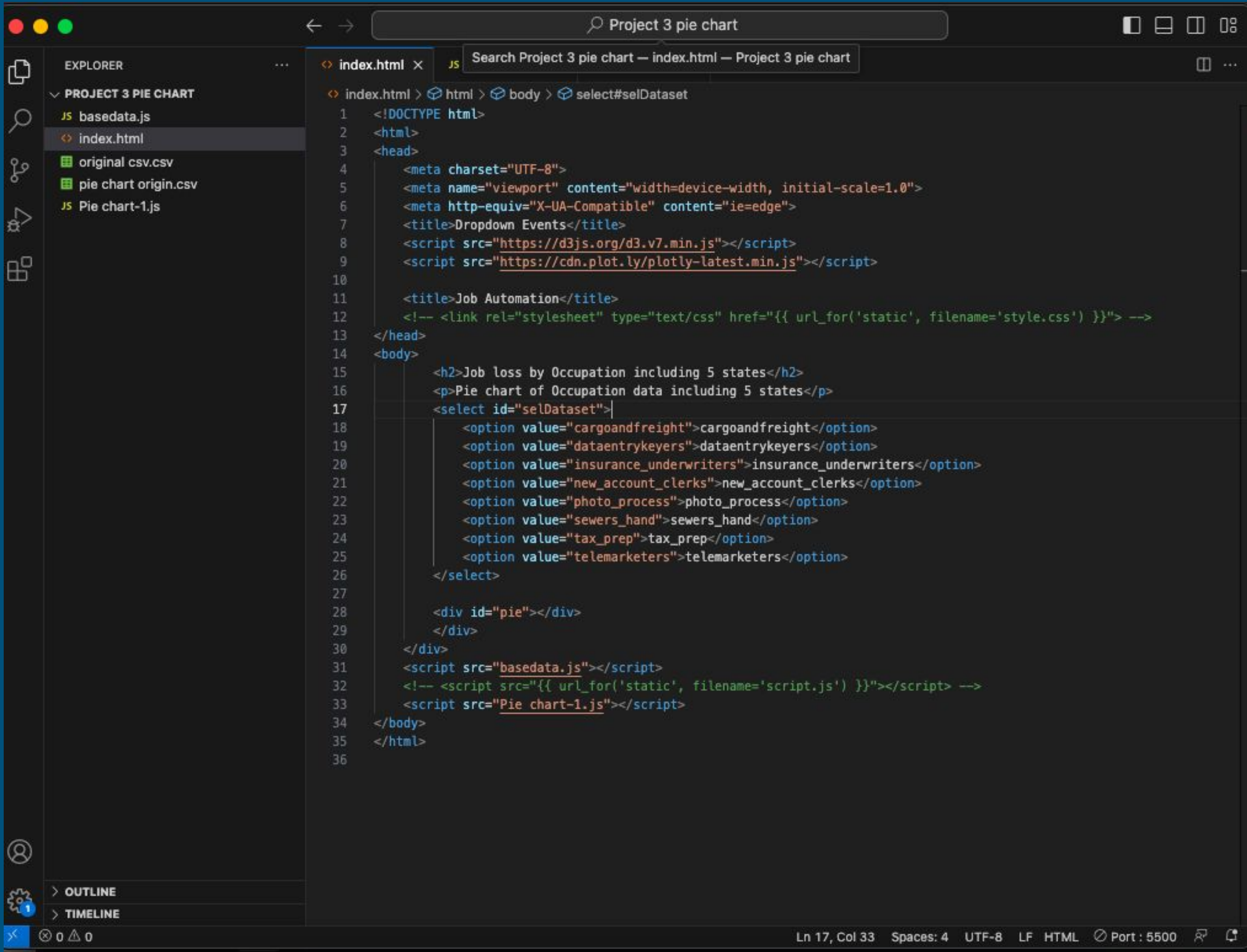
Pie chart of Occupation data including 5 states

cargoandfreight

Pie Chart HTML Code

Job loss by Occupation including 5 states

http://127.0.0.1:5500/index.html

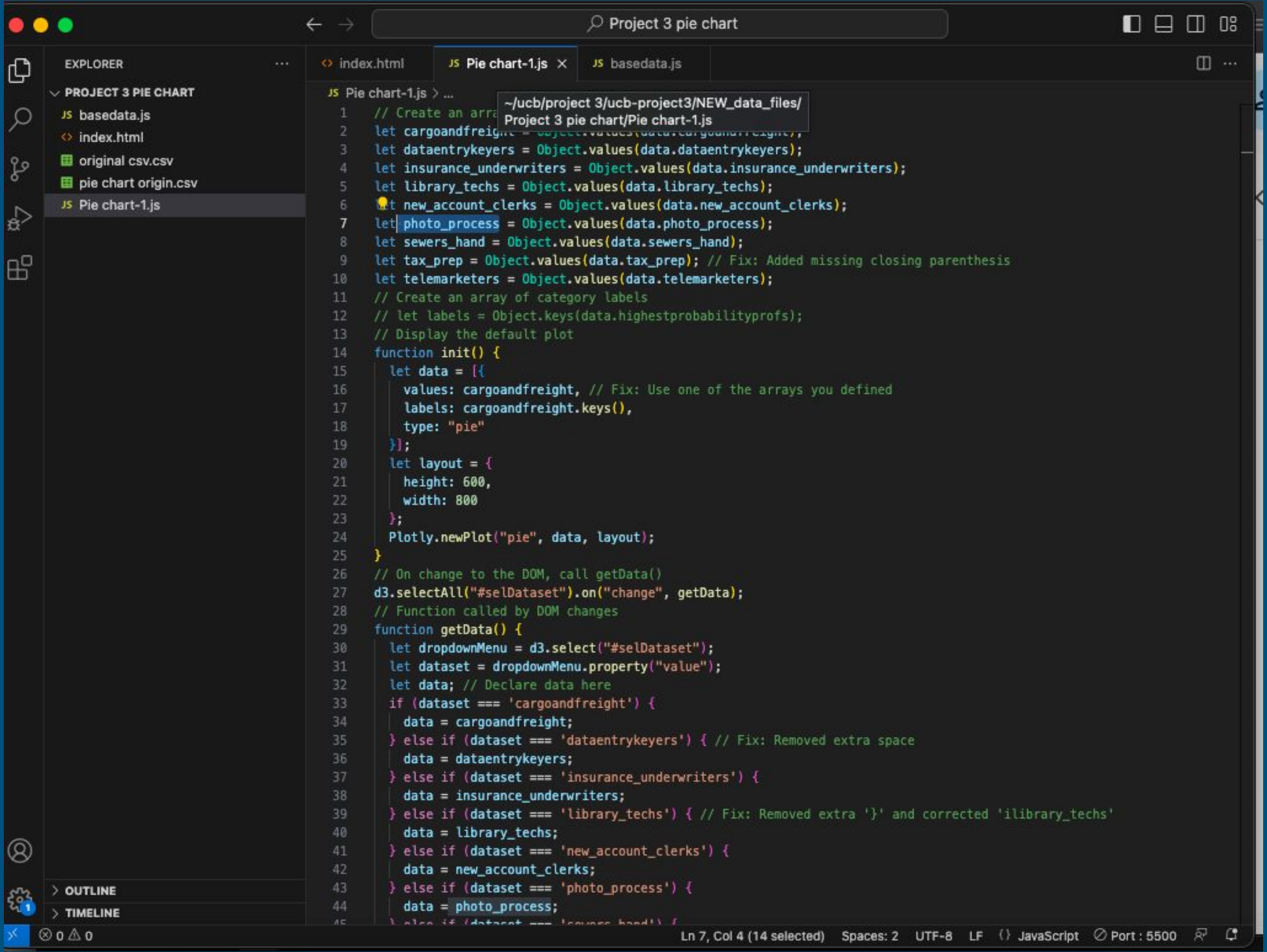


```html
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Dropdown Events</title>
    <script src="https://d3js.org/d3.v7.min.js"></script>
    <script src="https://cdn.plot.ly/plotly-latest.min.js"></script>

    <title>Job Automation</title>
    <!-- <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css') }}"> -->
</head>
<body>
        <h2>Job loss by Occupation including 5 states</h2>
        <p>Pie chart of Occupation data including 5 states</p>
        <select id="selDataset">
            <option value="cargoandfreight">cargoandfreight</option>
            <option value="dataentrykeyers">dataentrykeyers</option>
            <option value="insurance_underwriters">insurance_underwriters</option>
            <option value="new_account_clerks">new_account_clerks</option>
            <option value="photo_process">photo_process</option>
            <option value="sewers_hand">sewers_hand</option>
            <option value="tax_prep">tax_prep</option>
            <option value="telemarketers">telemarketers</option>
        </select>

        <div id="pie"></div>
        </div>
    </div>
    <script src="basedata.js"></script>
    <!-- <script src="{{ url_for('static', filename='script.js') }}"></script> -->
    <script src="Pie chart-1.js"></script>
</body>
</html>

```

Pie Chart HTML Code

Job loss by Occupation including 5 states

http://127.0.0.1:5500/index.html



~/ucb/project 3/ucb-project3/NEW_data_files/
Project 3 pie chart/Pie chart-1.js

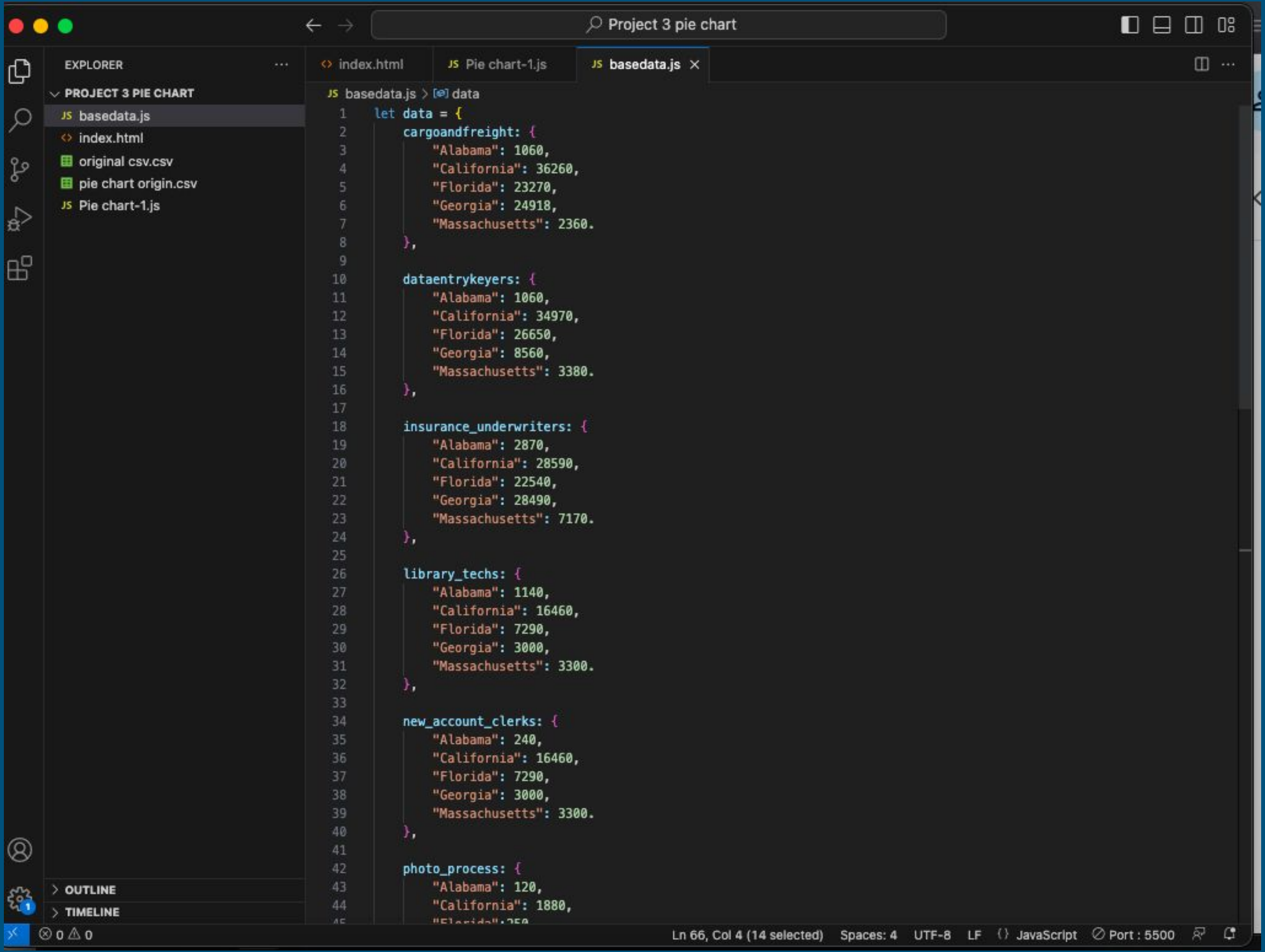
```
// Create an arra
let cargoandfreight = Object.values(data.cargoandfreight);
let dataentrykeyers = Object.values(data.dataentrykeyers);
let insurance_underwriters = Object.values(data.insurance_underwriters);
let library_techs = Object.values(data.library_techs);
let new_account_clerks = Object.values(data.new_account_clerks);
let photo_process = Object.values(data.photo_process);
let sewers_hand = Object.values(data.sewers_hand);
let tax_prep = Object.values(data.tax_prep); // Fix: Added missing closing parenthesis
let telemarketers = Object.values(data.telemarketers);
// Create an array of category labels
// let labels = Object.keys(data.highestprobabilityprofs);
// Display the default plot
function init() {
  let data = [{
    values: cargoandfreight, // Fix: Use one of the arrays you defined
    labels: cargoandfreight.keys(),
    type: "pie"
  }];
  let layout = {
    height: 600,
    width: 800
  };
  Plotly.newPlot("pie", data, layout);
}
// On change to the DOM, call getData()
d3.selectAll("#selDataset").on("change", getData);
// Function called by DOM changes
function getData() {
  let dropdownMenu = d3.select("#selDataset");
  let dataset = dropdownMenu.property("value");
  let data; // Declare data here
  if (dataset === 'cargoandfreight') {
    data = cargoandfreight;
  } else if (dataset === 'dataentrykeyers') { // Fix: Removed extra space
    data = dataentrykeyers;
  } else if (dataset === 'insurance_underwriters') {
    data = insurance_underwriters;
  } else if (dataset === 'library_techs') { // Fix: Removed extra '}' and corrected 'ilibrary_techs'
    data = library_techs;
  } else if (dataset === 'new_account_clerks') {
    data = new_account_clerks;
  } else if (dataset === 'photo_process') {
    data = photo_process;
```

```
⊗ 0 ⚠ 0    Ln 7, Col 4 (14 selected)  Spaces: 2  UTF-8  LF  {} JavaScript  Port : 5500
```

Pie Chart HTML Code

Job loss by Occupation including 5 states

http://127.0.0.1:5500/index.html



Project 3 pie chart

EXPLORER

PROJECT 3 PIE CHART
- basedata.js
- index.html
- original csv.csv
- pie chart origin.csv
- Pie chart-1.js

index.html | Pie chart-1.js | basedata.js

basedata.js > data

```
let data = {
    cargoandfreight: {
        "Alabama": 1060,
        "California": 36260,
        "Florida": 23270,
        "Georgia": 24918,
        "Massachusetts": 2360.
    },

    dataentrykeyers: {
        "Alabama": 1060,
        "California": 34970,
        "Florida": 26650,
        "Georgia": 8560,
        "Massachusetts": 3380.
    },

    insurance_underwriters: {
        "Alabama": 2870,
        "California": 28590,
        "Florida": 22540,
        "Georgia": 28490,
        "Massachusetts": 7170.
    },

    library_techs: {
        "Alabama": 1140,
        "California": 16460,
        "Florida": 7290,
        "Georgia": 3000,
        "Massachusetts": 3300.
    },

    new_account_clerks: {
        "Alabama": 240,
        "California": 16460,
        "Florida": 7290,
        "Georgia": 3000,
        "Massachusetts": 3300.
    },

    photo_process: {
        "Alabama": 120,
        "California": 1880,
```

OUTLINE

TIMELINE

⊗ 0 ⚠ 0   Ln 66, Col 4 (14 selected)   Spaces: 4   UTF-8   LF   {} JavaScript   Port : 5500

Conclusion

It's true…
The robots *are* coming for your job.

10 years ago, researchers surmised that technology would soon outperform humans at many tasks, and they were right!

47% of total U.S. employment is in the high risk category for automation within the next 10 years.

The good news? They still need Data Analysts to report on the efficacy of the robots, for now.
(wink wink)

1 in 4 companies have already replaced jobs with ChatGPT

- ResumeBuilder Survey, Feb 2023

85%+ Probability of Automation

Cargo and Freight Agents
Data Entry Keyers
Library Technicians
New Accounts Clerks
Photographic Process Workers
Sewers, Hand
Telemarketers
Watch and Clock Repairers
Insurance Underwriters
Tax Preparers
Title Examiners
Bookkeeping and Auditing Clerks
Driver/Sales Workers
Etchers and Engravers
Inspectors, Testers, Sorters, Samplers
Insurance Processing Clerks
Milling and Planing Machine Workers
Models
Order Clerks
Packaging and Filling Machine Operators
Parts Salespersons
Procurement Clerks
Shipping, Receiving, and Inventory Clerks
Tellers
Timing Device Assemblers
Umpires, Referees, and Sports Officials
Pourers and Casters, Metal

Photographic Equipment Repairers
Cashiers
Counter and Rental Clerks
Credit Authorizers, Checkers, and Clerks
Crushing and Polishing Machine Operators
Dental Laboratory Technicians
Electromechanical Assemblers
File Clerks
Grinding and Polishing Workers, Hand
Hosts and Hostesses, Food Service
Log Graders and Scalers
Motion Picture Projectionists
Ophthalmic Laboratory Technicians
Pesticide Handlers, Sprayers
Prepress Technicians and Workers
Shoe Machine Operators and Tenders
Telephone Operators
Textile Machine Operators
Woodworking Machine Operators
Bridge and Lock Tenders
Farm Labor Contractors
Payroll and Timekeeping Clerks
Real Estate Brokers
Billing and Posting Clerks
Cooks, Restaurant
Highway Maintenance Workers
Parking Attendants

Brokerage Clerks
Claims Adjusters, Examiners
Credit Analysts
Insurance Appraisers, Auto Damage
Secretaries and Admin Assistants
Loan Officers
Agricultural Technicians
Dispatchers, Except Emergency
Fast Food and Counter Workers
Gambling Dealers
Office Clerks, General
Receptionists and Information Clerks
Rock Splitters, Quarry
Secretaries and Admin Assistants
Switchboard Operators
Winding, Twisting Machine Operators
Ushers and Ticket Takers
Locomotive Engineers
Model Makers, Wood
Surveying and Mapping Technicians
Compensation and Benefits Managers
Adhesive Bonding Machine Operators
Carpet Installers
Floor Sanders and Finishers
Agricultural Workers, All Other
Carpet Installers
Paperhangers

Print Binding and Finishing Workers
Textile Cutting Machine Operators
Weighers, Measurers, Samplers
Nuclear Power Reactor Operators
Operating Engineers, Equipment Construction
Postal Service Clerks
Agricultural Inspectors
Bicycle Repairers
Coin, Vending Machine Servicers
Cooks, Short Order
Couriers and Messengers
Door-to-Door Sales Workers
Drilling and Boring Machine Operators
First-Line Supervisors, Janitorial
Helpers--Painters, Paperhangers
Hotel, Motel, and Resort Desk Clerks
Interviewers, Except Eligibility and Loan
Mail Machine Operators
Meat, Poultry, and Fish Cutters
Tire Builders
Waiters and Waitresses
Accountants and Auditors
Floor Sanders and Finishers
Food Preparation Workers
Forest and Conservation Workers
Furniture Finishers

85%+ Probability of Automation

Animal Breeders
Bill and Account Collectors
Gambling Surveillance Officers
Polishing, and Buffing Machine Operators
Jewelers and Stone Setters
Landscaping and Groundskeeping Workers
Library Assistants, Clerical
Manicurists and Pedicurists
Molding and Casting Machine Operators
Cement Masons and Concrete Finishers
Excavating and Loading Machine Operators
Paralegals and Legal Assistants
Budget Analysts
Welders, Cutters, Solderers, and Brazers
Butchers and Meat Cutters
Conveyor Operators and Tenders
Cooling and Freezing Equipment Operators
Compacting Machine Setters
Fiberglass Laminators and Fabricators
Forging Machine Setters
Industrial Truck and Tractor Operators
Subway and Streetcar Operators
Laborers and Material Movers
Chemical Plant and System Operators

Machine Feeders and Offbearers
Outdoor Power Equipment Mechanics
Refuse and Recyclable Collectors
Model Makers, Metal and Plastic
Service Unit Operators, Oil and Gas
Tax Examiners and Collectors
Cabinetmakers and Bench Carpenters
Dredge Operators
Fence Erectors
Food Preparation Workers
Helpers--Carpenters
Loan Interviewers and Clerks
Office Machine Operators, Except Computer
Painting, Coating Workers
Pharmacy Technicians
Plating Machine Setters
Production Workers, All Other
Retail Salespersons
Insurance Sales Agents
Coating, Painting, Machine Operators
Dining and Cafeteria Attendants
Extruding and Drawing Machine Setters
Food and Tobacco Roasting
Maintenance Workers, Machinery
Plant and System Operators, All Other
Real Estate Sales Agents

Mechanical Door Repairers
Gambling and Sports Book Writers
Heat Treating Equipment Setter
Information and Record Clerks
Medical Records Specialists
Multiple Machine Tool Setters
Musical Instrument Repairers and Tuners
Tour and Travel Guides
Automotive Body and Related Repairers
Electrical and Electronics Installers
Gas Pumping Station Operators
Geological Technicians
Health Information Technologists
Patternmakers, Wood
Rail Yard Engineers
Human Resources Assistants
Molders, Shapers, and Casters
Roofers
Crane and Tower Operators
Patternmakers, Metal and Plastic
Property Appraisers and Assessors
Pump Operators
Reinforcing Iron and Rebar Workers
Signal and Track Switch Repairers
Sawing Machine Setters
Veterinary Assistants
Executive Administrative Assistants

Traffic Technicians
Transportation Inspectors
Bakers
Bus Drivers, School
Medical Transcriptionists
Sewing Machine Operators
Taxi Drivers
Rail-Track Equipment Operators
Riggers
Stationary Engineers
Stonemasons
Technical Writers
Construction Laborers
Forming Machine Setters
Metal-Refining Furnace Operators
Semiconductor Processing Technicians
Still Machine Operators
Tool Grinders, Filers, and Sharpeners
Cartographers and Photogrammetrists
Planning and Expediting Clerks
Rail Car Repairers
Terrazzo Workers and Finishers
Agricultural Workers, All Other
Computer Controlled Tool Operators
Correspondence Clerks
Cutting and Slicing Machine Setters
Food Servers, Nonrestaurant