

US Elections 2020

Economic and Demographic insights into voter behaviors and trends

++

++

+



Computer Programming and Data Management

Group: print("group name")



Analysis theme

How Economic, Social and Demographics factors influence voter behavior and election outcomes in the US and how elections affect them afterward.

Focus on 2020 elections



Index



US in 2020/21



Economic factors



Voting behavior at
Geographic level

Electoral topics

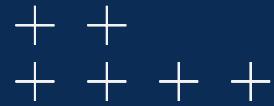


Linear regression



Conclusions





US in 2020/21

Election Year Context



Importance of the 2020 elections

The 2020 US presidential election was a defining moment in American history, taking place amidst the unprecedented challenges of a global pandemic of COVID-19.

This election saw Donald Trump, the Republican incumbent, facing off against Joe Biden, the Democratic challenger. Their campaigns highlighted starkly different approaches to key issues, such as federal response to the pandemic, economic recovery plans, and racial justice protests.

The deeply polarized political climate, combined with social and economic factors, played a critical role in shaping voter behavior, ultimately leading to Biden's victory.





Economic factors

Occupation
Unemployment
Weekly Earnings
Consumer Price Index

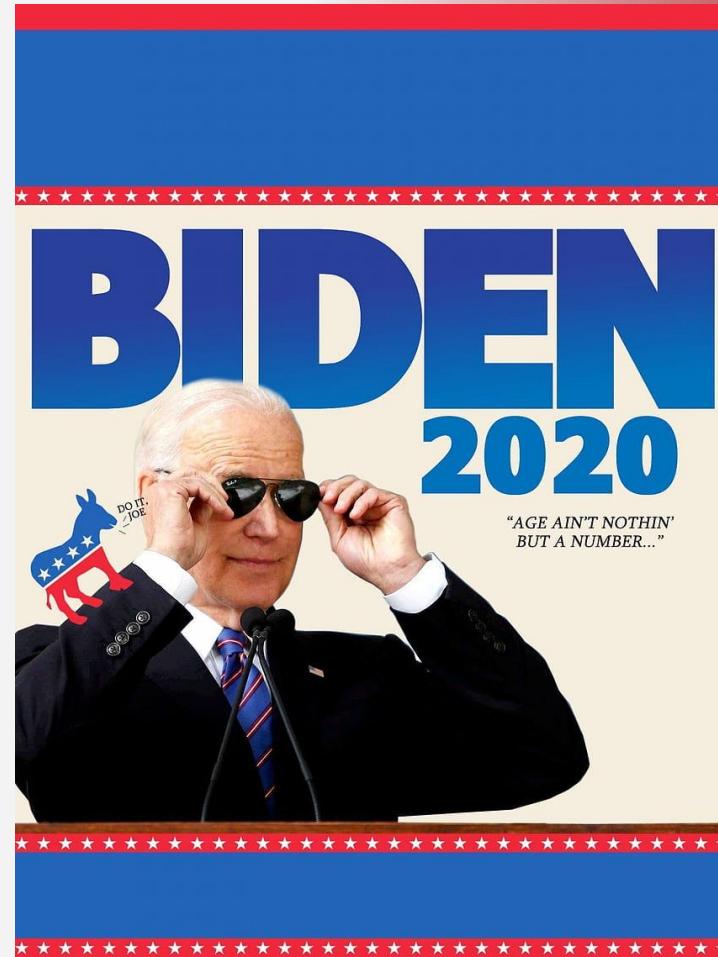
Biden economic proposal

During the 2020 elections, several economic proposal were put forward by Joe Biden. His plans emphasized job creation, clean energy, education, and healthcare, with significant attention to social justice and equity.

Biden's overarching economic recovery strategy had three pillars:

- **Rescue:** immediate relief for those hit hardest by the COVID-19 pandemic;
- **Recovery:** stimulus measures to rebuild and stabilize the economy;
- **Rebuild for Resilience:** structural reforms to ensure long-term growth.

It is therefore interesting to analyze how the 2020 election, which saw Biden as president, influenced economic factors and how these changed in the periods following the elction.



Datasets

For the economic part we have obtained the datasets useful for the analysis from the website of U.S. Bureau of Labor Statistics.

It is the US government's principal research agency in the broad field of economics and labor statistics and among the most important agencies.

<https://www.bls.gov/>

An official website of the United States government [Here is how you know](#)

U.S. BUREAU OF LABOR STATISTICS

Follow Us [X](#) | [Release Calendar](#) | [Blog](#)

Search BLS.gov

HOME SUBJECTS DATA TOOLS PUBLICATIONS ECONOMIC RELEASES CLASSROOM BETA

Announcements

» Public invited to comment on potential revisions to North American Industry Classification System [Read More](#)

JAN 17 November job openings rates up in 2 states, down in 2; hires rates down in 6 states

November job openings rates rose in 2 states and fell in 2. Hires rates fell in 6 states and rose in 1. Total separations rates fell in 3 states; quits rates fell in 5, and layoffs and discharges rates fell in 3 and rose in 3.

[HTML](#) | [PDF](#) | [RSS](#) | [Charts](#)

01/16/2025 U.S. import prices increase 0.1% in December; export prices advance 0.3%

01/15/2025 CPI for all items rises 0.4% in December; gasoline and shelter up

01/15/2025 Real average hourly earnings for all employees decrease 0.2% in December

01/14/2025 PPI for final demand rises 0.2% in December; goods increase 0.6%, services unchanged

CAREER OUTLOOK

Business career options: Outlook, wages, and entry requirements

Employment in many business occupations is projected to grow faster than average from 2023 to 2033.

[read more >>](#)

An official website of the United States government [Here is how you know](#)

U.S. BUREAU OF LABOR STATISTICS

Follow Us [X](#) | [Release Calendar](#) | [Blog](#)

Search BLS.gov

HOME SUBJECTS DATA TOOLS PUBLICATIONS ECONOMIC RELEASES CLASSROOM BETA

Bureau of Labor Statistics > Data Tools

Databases, Tables & Calculators by Subject

On This Page:

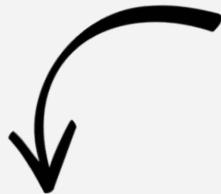
Inflation & Prices	Spending & Time Use	Series Report
Employment	Productivity	Historical News Release Tables
Unemployment	Workplace Injuries	Maps
Employment Projections	Occupational Requirements	Calculators
Pay & Benefits	Regional Resources	Public Data API
	International	

Inflation & Prices

Database Name	Special Notice	Top Picks	Data Finder	One Screen	Multi-Screen	Tables	Text Files
Prices - Consumer							
All Urban Consumers (Current Series) (Consumer Price Index - CPI)							

Occupation - Data cleaning

Data cleaning process for the occupation dataset.



```
work = pd.read_excel('/content/file/work.xlsx','tot')
display(work.head())
print('-----')
print(work.info())
print(work.isnull().sum())

Occupation, race, and Hispanic or Latino ethnicity      Total   Unnamed: 2    Men   Unnamed: 4   Women   Unnamed: 6
0                           NaN  2020.0  2021.0  2020.0  2021.0  2020.0  2021.0
1                           TOTAL  NaN     NaN     NaN     NaN     NaN     NaN
2           Total, 16 years and over (in thousands)  147795.0  152581.0  78560.0  80829.0  69234.0  71752.0
3           Percent of total employed                100.0    100.0    100.0    100.0    100.0    100.0
4       Management, professional, and related occupations  43.1     42.4     39.1     38.5     47.5     46.9
-----
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 18 entries, 0 to 17
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype  
--- 
 0   Occupation, race, and Hispanic or Latino ethnicity  17 non-null    object  
 1   Total           17 non-null    float64 
 2   Unnamed: 2      17 non-null    float64 
 3   Men             17 non-null    float64 
 4   Unnamed: 4      17 non-null    float64 
 5   Women           17 non-null    float64 
 6   Unnamed: 6      17 non-null    float64 
dtypes: float64(6), object(1)
memory usage: 1.1+ KB
None
Occupation, race, and Hispanic or Latino ethnicity      1
Total           1
Unnamed: 2      1
Men             1
Unnamed: 4      1
Women           1
Unnamed: 6      1
dtype: int64
```

The data contain information regarding:

- **Total population**
- **White**
- **Black**
- **Asian**
- **Hispanic or Latino**

Occupation - Data cleaning

```
xls_sheets = ['tot', 'white', 'black', 'asian', 'latino']

df_i = []
for i, sheet in enumerate(xls_sheets):
    df = pd.read_excel('/content/file/work.xlsx', sheet_name=sheet)
    df = df.rename(columns = {'Occupation, race, and Hispanic or Latino ethnicity': xls_sheets[i],
                            'Total': 'Total_2020', 'Unnamed: 2': 'Total_2021',
                            'Men': 'Men_2020', 'Unnamed: 4': 'Men_2021', 'Women':'Women_2020',
                            'Unnamed: 6': 'Women_2021'})
    df_i.append(df)

df = df.drop(index=[0,1])
print(df.info())
print(df.isnull().sum())
print('-----')
df.to_csv(f'/content/file/{xls_sheets[i]}.csv', index=False)
```



From the occupation file excel we created a new worksheet for each ethnicity.

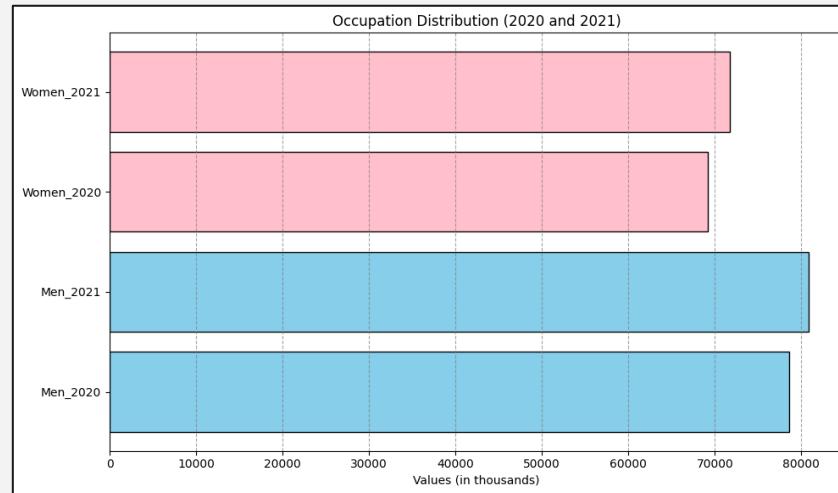
#	Column	Non-Null Count	Dtype	#	Column	Non-Null Count	Dtype
0	white	16 non-null	object	0	black	16 non-null	object
1	Total_2020	16 non-null	float64	1	Total_2020	16 non-null	float64
2	Total_2021	16 non-null	float64	2	Total_2021	16 non-null	float64
3	Men_2020	16 non-null	float64	3	Men_2020	16 non-null	float64
4	Men_2021	16 non-null	float64	4	Men_2021	16 non-null	float64
5	Women_2020	16 non-null	float64	5	Women_2020	16 non-null	float64
6	Women_2021	16 non-null	float64	6	Women_2021	16 non-null	float64
dtypes: float64(6), object(1)				dtypes: float64(6), object(1)			
#	Column	Non-Null Count	Dtype	#	Column	Non-Null Count	Dtype
0	asian	16 non-null	object	0	latino	16 non-null	object
1	Total_2020	16 non-null	float64	1	Total_2020	16 non-null	float64
2	Total_2021	16 non-null	float64	2	Total_2021	16 non-null	float64
3	Men_2020	16 non-null	float64	3	Men_2020	16 non-null	float64
4	Men_2021	16 non-null	float64	4	Men_2021	16 non-null	float64
5	Women_2020	16 non-null	float64	5	Women_2020	16 non-null	float64
6	Women_2021	16 non-null	float64	6	Women_2021	16 non-null	float64
dtypes: float64(6), object(1)				dtypes: float64(6), object(1)			



Occupation - Men and Women

This bar chart compares the occupation distribution by gender for the years 2020 and 2021.

It highlights a visible increase in employment levels for both men and women from 2020 to 2021, with men consistently showing higher absolute numbers in both years.



```
df_tot = pd.read_csv('/content/file/tot.csv')
df_white = pd.read_csv('/content/file/white.csv')
df_black = pd.read_csv('/content/file/black.csv')
df_asian = pd.read_csv('/content/file/asian.csv')
df_latino = pd.read_csv('/content/file/latino.csv')

occupation_data = df_tot.iloc[0, 3:7]
colors = ['skyblue' if 'Men' in label else 'pink' for label in occupation_data.index]

plt.figure(figsize = (10, 6))
plt.barh(occupation_data.index, occupation_data.values, color = colors, edgecolor = 'black' )
plt.title('Occupation Distribution (2020 and 2021)')
plt.grid(axis='x', linestyle='--', alpha=0.7, color = 'grey')
plt.xlabel('Values (in thousands)')

plt.tight_layout()
plt.show()
```



Occupation - Type of work



The following code was written to create a graph that compared the various ethnic groups, divided by type of work and by the two reference years: 2020 and 2021.

```
categories = ['Management, professional, and related occupations', 'Management, business, and financial operations occupations',
    'Professional and related occupations', 'Service occupations','Sales and office occupations', 'Sales and related occupations',
    'Office and administrative support occupations', 'Natural resources, construction, and maintenance occupations',
    'Farming, fishing, and forestry occupations', 'Construction and extraction occupations', 'Installation, maintenance, and repair occupations',
    'Production, transportation, and material moving occupations', 'Production occupations', 'Transportation and material moving occupations']

ethnicity = ["tot", "white", "black", "asian", "latino"]

namex = [
    'Management',
    'Business and financial',
    'Professional',
    'Service',
    'Office',
    'Sales',
    'Administrative support',
    'Construction and maintenance',
    'Farming and fishing',
    'Extraction',
    'Maintenance and repair',
    'Transportatio',
    'Production',
    'Material moving']

for i, df in enumerate([df_tot, df_white, df_black, df_asian, df_latino]):

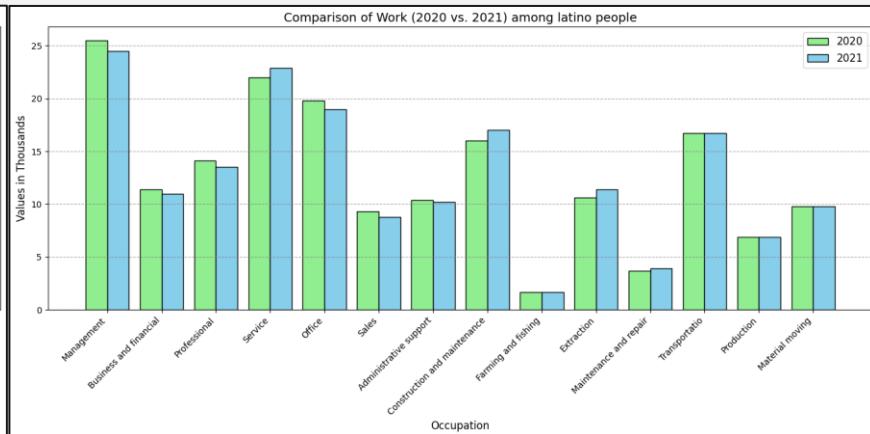
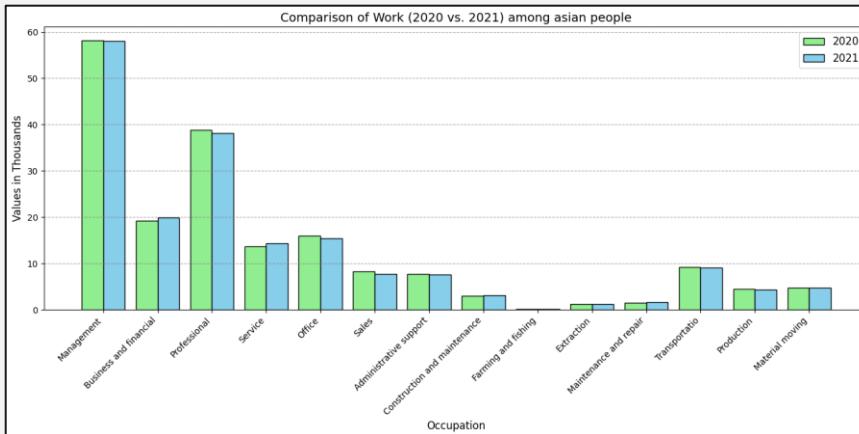
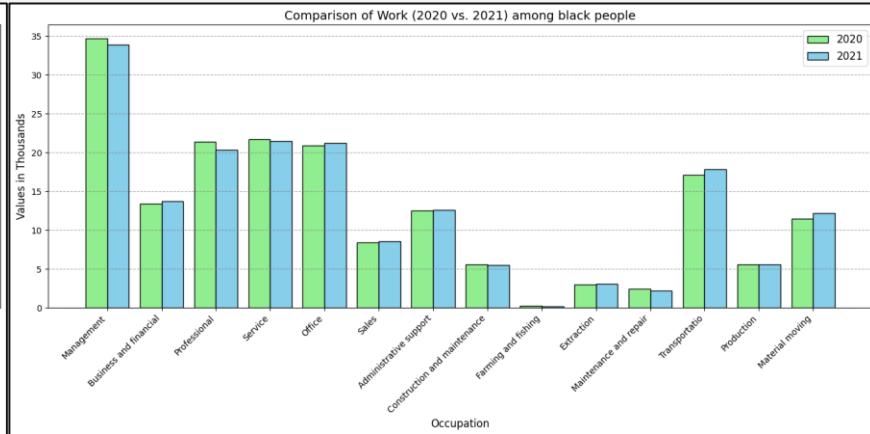
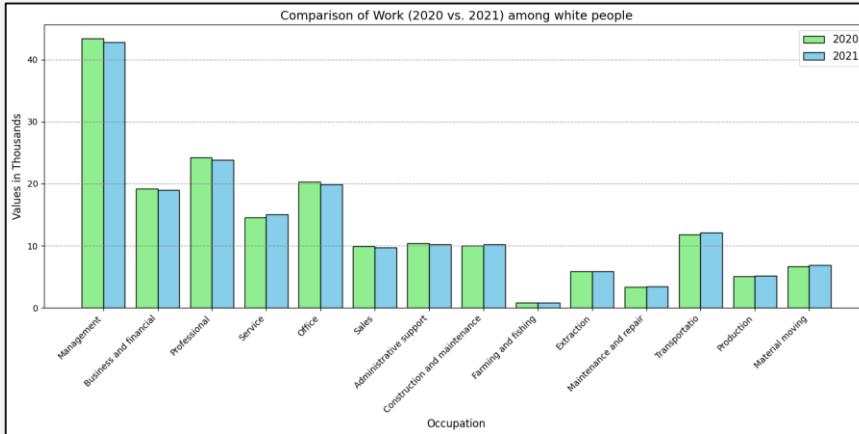
    filtered_df = df[df[ethnicity[i]].isin(categories)]

    bar_width = 0.4
    x = np.arange(len(filtered_df))
    values_2020 = filtered_df['Total_2020']
    values_2021 = filtered_df['Total_2021']

    plt.figure(figsize=(14, 7))
    plt.bar(x - bar_width/2, values_2020, bar_width, label='2020', color='lightgreen', edgecolor='black')
    plt.bar(x + bar_width/2, values_2021, bar_width, label='2021', color='skyblue', edgecolor='black')
    plt.xlabel('Occupation', fontsize=12)
    plt.ylabel('Values in Thousands', fontsize=12)
    plt.title(f'Comparison of Work (2020 vs. 2021) among {np.where(i==0, "total", ethnicity[i])} people', fontsize=14)
    plt.xticks(x, namex, rotation=45, ha='right', fontsize=10)
    plt.legend(loc='upper right', fontsize=12)
    plt.grid(axis = 'y', linestyle='--', alpha=0.7, color = 'grey')

    plt.tight_layout()
    plt.show()
```

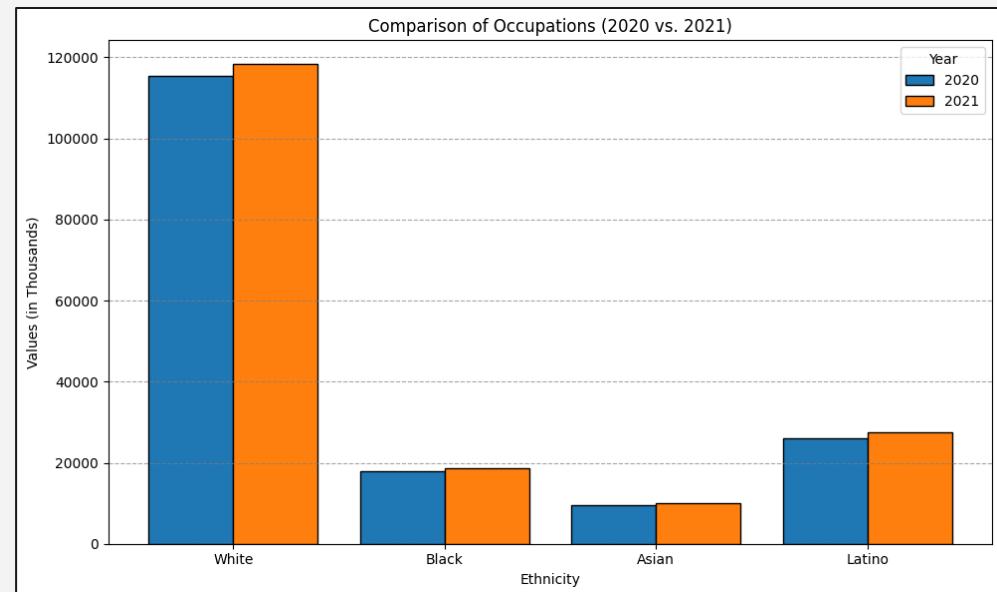
Occupation – Type of work



Occupation - Ethnicity summary

In the previous slide the changes for each job type for each ethnic group were analyzed.

In this graph we can observe the change in the number of workers from 2020 to 2021 based on ethnic group, regardless of gender and type of work.



Unemployment level - Data cleaning

```
df = pd.read_excel('/content/file/Unemployment_Level.xlsx')

df = df.drop(range(0,11))
df.drop('Unnamed: 13', axis = 1, inplace = True)
df = df.rename(columns = {'Labor Force Statistics from the Current Population Survey': 'Year',
                           'Unnamed: 1': 'Jan',
                           'Unnamed: 2': 'Feb',
                           'Unnamed: 3': 'Mar',
                           'Unnamed: 4': 'Apr',
                           'Unnamed: 5': 'May',
                           'Unnamed: 6': 'Jun',
                           'Unnamed: 7': 'Jul',
                           'Unnamed: 8': 'Aug',
                           'Unnamed: 9': 'Sep',
                           'Unnamed: 10': 'Oct',
                           'Unnamed: 11': 'Nov',
                           'Unnamed: 12': 'Dec',})

df = df.set_index('Year')

df.dropna(inplace= True)

print('Number of unemployed people')
unemployed_df = df.T
display(unemployed_df)

unemployed_df.to_csv('cleaned_UnemploymentLevel', index = True)
df1 = pd.read_csv('cleaned_UnemploymentLevel', index_col= 0)
```

Year	2020	2021
Jan	5842	10196
Feb	5729	9992
Mar	7209	9734
Apr	23090	9801
May	20933	9250
Jun	17658	9547
Jul	16391	8770
Aug	13498	8288
Sep	12573	7659
Oct	11012	7244
Nov	10713	6733
Dec	10772	6305

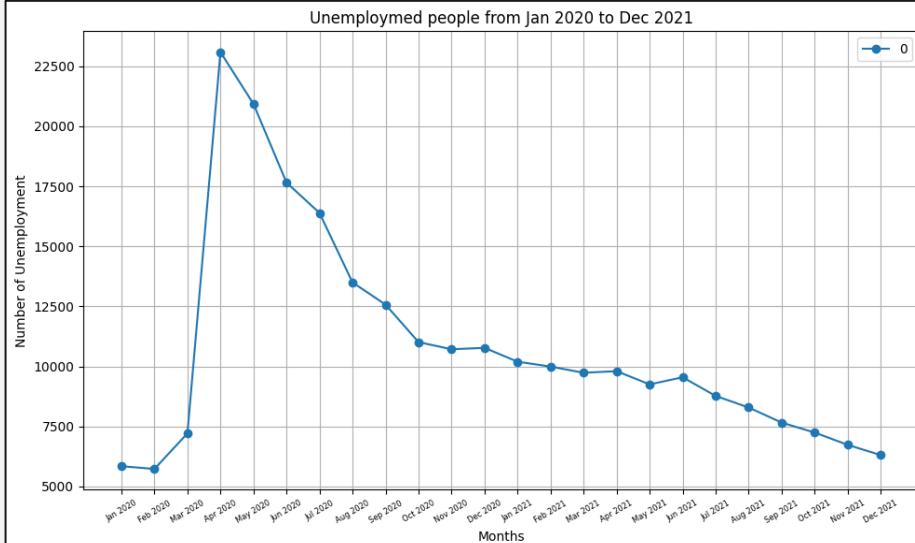
This dataset contains information regarding the level of unemployment (in thousands) for each month of 2020 and 2021.



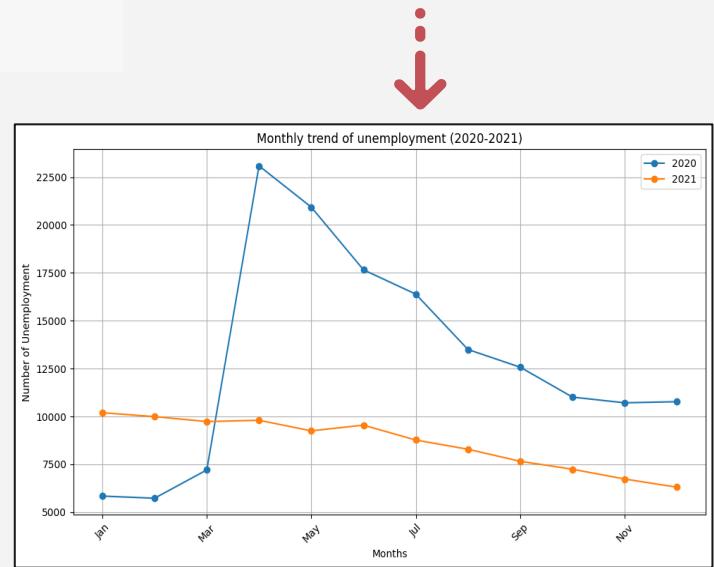
Unemployment level

①
lst = df1.values.T.flatten().tolist()
Tot = pd.DataFrame(lst)

```
ax = Tot.plot(kind='line', marker='o')  
ax.set_xlabel("Months")  
ax.set_ylabel("Number of Unemployment")  
ax.set_title("Unemployed people from Jan 2020 to Dec 2021")  
ax.grid(True)  
plt.xticks(ticks = range(24),  
           labels = [f'{month} {year}' for year in ['2020', '2021'] for month in ['Jan", "Feb", "Mar", "Apr", "May", "Jun",  
           "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"]],  
           rotation = 30,  
           fontsize = 6)  
plt.tight_layout()  
plt.show()
```

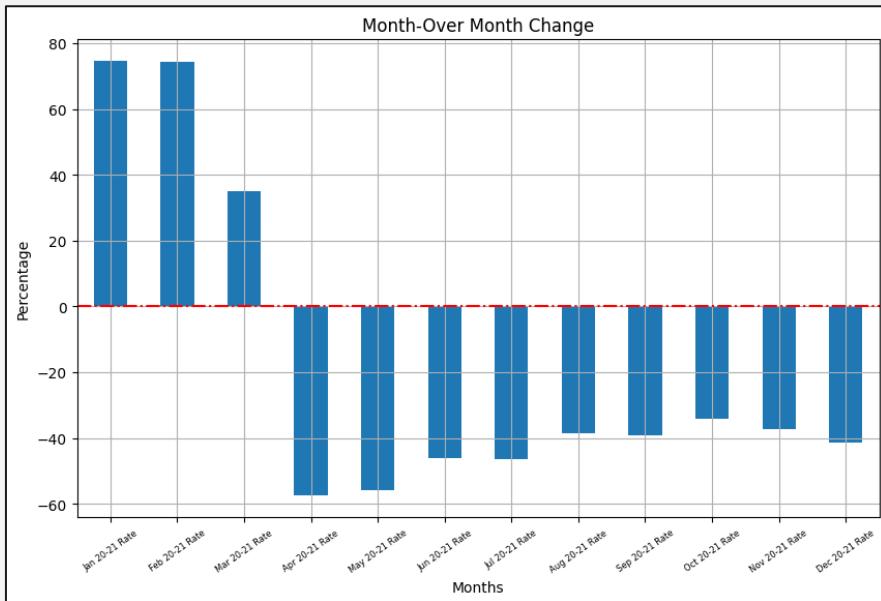


②
ax = df1.plot(kind='line', marker='o')
ax.set_xlabel("Months")
ax.set_ylabel("Number of Unemployment")
ax.set_title("Monthly trend of unemployment (2020-2021)")
ax.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()



Unemployment level

```
df1 = df1.T  
df1.loc['%'] = ((df1.loc['2021'] - df1.loc['2020']) / df1.loc['2020']) * 100  
df1 = df1.T  
  
ax = df1['%'].plot(kind = 'bar')  
ax.set_xlabel('Months')  
ax.set_ylabel('Percentage')  
ax.set_title("Month-Over Month Change")  
ax.grid(True)  
plt.axhline(y = 0, color = 'r', linestyle = '-.')  
plt.xticks(ticks = range(12),  
           labels = [f'{i} 20-21 Rate' for i in ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"]],  
           fontsize = 6,  
           rotation = 35)  
plt.show()
```



Here we can see a month-by-month comparison for the years 2020 and 2021.

- **Positive:** 2021 value is greater than 2020 value;
- **Negative:** 2020 value is greater than 2021 value



Unemployment rate - Data cleaning

```
State_unemployment_rates = pd.read_excel('/content/file/State_unemployment_rates.xlsx')
display(State_unemployment_rates.head())
print('-----')
print(State_unemployment_rates.info())
print(State_unemployment_rates.isnull().sum())

def change_columns():
    months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
    years = [2020, 2021]
    column_names = [f'{month}{year}' for year in years for month in months]
    return column_names

new_columns_names = change_columns()
State_unemployment_rates.columns = [State_unemployment_rates.columns[0]] + new_columns_names

State_unemployment_rates.to_csv(f'/content/file/State_unemployment_rates.csv', index = False)
display(State_unemployment_rates)
```



With this command we have abbreviated the name of the year columns.

We also analyzed the unemployment rate for each state and for each month for the years 2020 and 2021.

State	Jan2020	Feb2020	Mar2020	Apr2020	May2020	Jun2020	Jul2020	Aug2020	Sep2020	... Mar2021	Apr2021	May2021	Jun2021	Jul2021	Aug2021	Sep2021	Oct2021	Nov2021	Dec2021
0 Alabama	3.2	3.3	3.4	13.8	10.4	8.6	7.5	6.3	5.9	... 3.8	3.7	3.5	3.5	3.3	3.2	3.0	2.9	2.8	2.7
1 Alaska	5.4	5.5	5.5	11.7	11.8	11.2	11.1	8.6	7.9	... 7.0	7.0	6.8	6.8	6.4	6.2	5.9	5.7	5.5	5.3
2 Arizona	4.8	4.8	4.9	13.8	11.2	9.8	8.9	7.9	7.5	... 6.0	5.8	5.5	5.3	5.0	4.6	4.3	4.1	3.9	3.7
3 Arkansas	3.5	3.6	4.9	10.1	8.9	7.9	7.3	6.4	6.0	... 4.7	4.5	4.4	4.2	3.9	3.7	3.5	3.4	3.3	3.2
4 California	4.3	4.4	5.5	16.1	15.8	13.8	13.2	11.9	10.0	... 8.4	8.3	7.9	7.8	7.4	7.0	6.5	6.1	5.7	5.5
...
46 Virginia	2.8	2.9	3.2	12.0	10.0	8.9	8.2	7.2	6.6	... 4.5	4.4	4.2	4.0	3.8	3.5	3.3	3.1	3.0	2.9
47 Washington	3.7	3.8	5.2	16.7	13.4	11.5	10.4	8.8	8.1	... 6.0	5.7	5.5	5.4	5.2	5.0	4.7	4.5	4.2	4.1
48 West Virginia	5.3	5.3	5.4	15.8	12.4	10.5	9.4	8.1	7.5	... 5.9	5.7	5.5	5.4	5.1	4.9	4.6	4.4	4.2	4.0
49 Wisconsin	3.1	3.1	3.0	14.0	10.6	8.7	7.7	6.4	5.8	... 4.4	4.3	4.1	4.0	3.8	3.6	3.4	3.3	3.1	3.0
50 Wyoming	4.4	4.6	4.8	5.1	8.7	7.3	6.8	6.2	5.9	... 5.2	5.0	4.9	4.7	4.5	4.3	4.1	3.9	3.7	3.6

Unemployment rate - Code

① data = State_unemployment_rates

```
def download_USA_maps(url, extract_to):
    # Download and extract shapefile data from the URL
    zip_path = Path('temp_shapefile.zip')
    response = requests.get(url)
    zip_path.write_bytes(response.content)

    with ZipFile(zip_path, 'r') as zip_ref:
        zip_ref.extractall(extract_to)

    zip_path.unlink()

# URL of the shapefile and directory where store it
shapefile_url = 'https://www2.census.gov/geo/tiger/GENZ2021/shp/cb_2021_us_state_20m.zip'
shapefile_dir = Path("usa_shapefiles")
shapefile_dir.mkdir(exist_ok = True)

download_USA_maps(shapefile_url, shapefile_dir)

shapefile_path = shapefile_dir / 'cb_2021_us_state_20m.shp'

gdf = gpd.read_file(shapefile_path)

excluded_states = ["AS", "GU", "MP", "PR", "VI"]
gdf = gdf[~gdf["STUSPS"].isin(excluded_states)]

alaska = gdf[gdf["STUSPS"] == "AK"].scale(0.35, 0.35).translate(-4000000, -2500000)
hawaii = gdf[gdf["STUSPS"] == "HI"].scale(1.5, 1.5).translate(5200000, -1400000)
gdf = gdf[~gdf["STUSPS"].isin(["AK", "HI"])]

gdf = pd.concat([gdf, alaska, hawaii])

data.rename(columns = {'State': 'NAME'}, inplace = True)
merged = gdf.merge(data, on = 'NAME', how = 'left')

output_dir = Path('us_unemployment_maps')
output_dir.mkdir(exist_ok = True)

bins = [0, 2.9, 4.9, 6.9, 8.9, 100]
colors = ["#deebf7", "#9ecae1", "#3182bd", "#08519c", "#08306b"]
cmap = ListedColormap(colors)
norm = BoundaryNorm(bins, cmap.N)
```

②

```
months = data.columns[1:]
for month in months:
    fig, ax = plt.subplots(1, 1, figsize=(25, 15))

    merged.plot(column = month,
                cmap = cmap,
                norm = norm,
                ax = ax,
                linewidth = 0.5,
                missing_kwds = {'color': 'lightgrey', 'label': 'No data'},
                edgecolor = 'black')

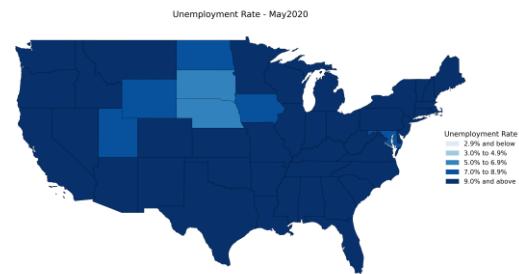
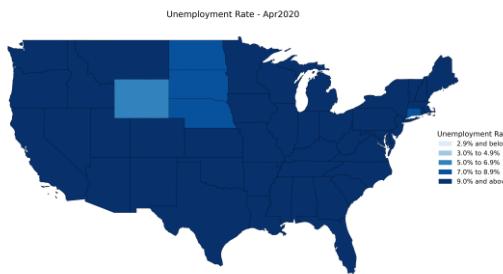
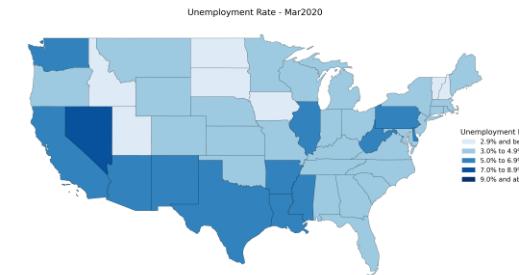
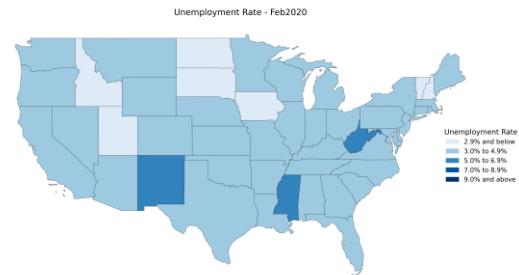
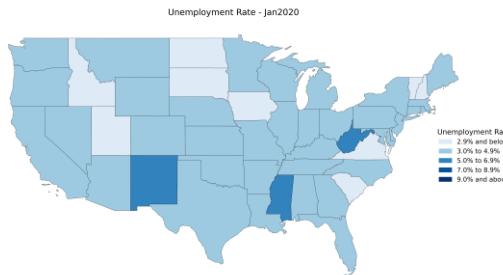
    ax.set_title(f'Unemployment Rate - {month}', fontsize = 24, pad = 20)
    ax.axis('off')

    legend_labels = [
        "2.9% and below",
        "3.0% to 4.9%",
        "5.0% to 6.9%",
        "7.0% to 8.9%",
        "9.0% and above",
    ]
    legend_handles = [
        mpatches.Patch(color=colors[i], label=legend_labels[i]) for i in range(len(colors))
    ]
    ax.legend(
        handles=legend_handles,
        title="Unemployment Rate",
        loc="center left",
        bbox_to_anchor=(0.9, 0.5),
        fontsize=18,
        title_fontsize=20,
        frameon=False
    )

    map_path = output_dir / f'Unemployment_Rate_{month}.png'
    plt.savefig(map_path, dpi = 300)
    plt.close()

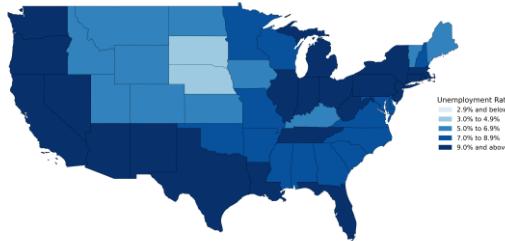
print(f'Maps saved in: {output_dir}')
```

Unemployment rate - 2020

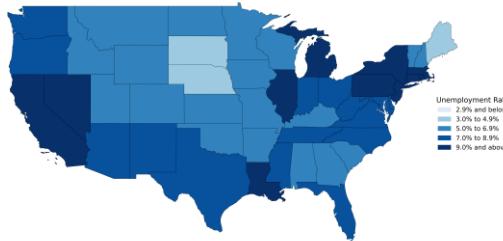


Unemployment rate - 2020

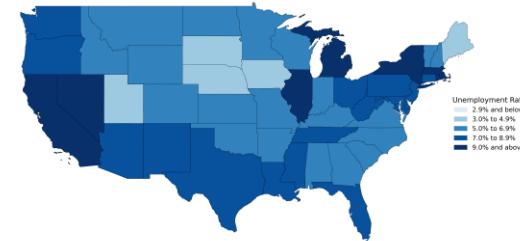
Unemployment Rate - Jul2020



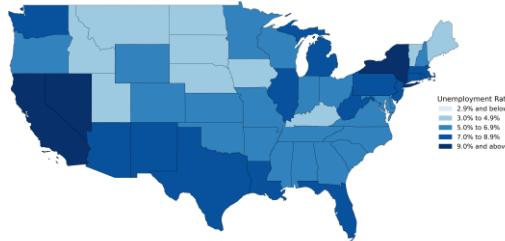
Unemployment Rate - Aug2020



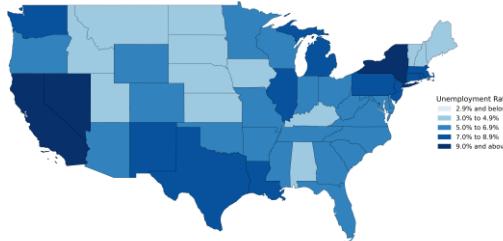
Unemployment Rate - Sep2020



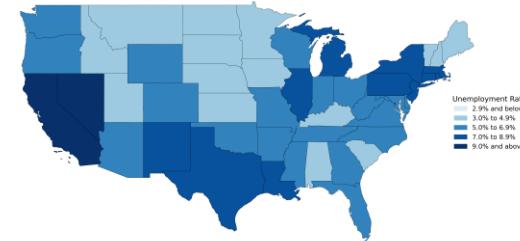
Unemployment Rate - Oct2020



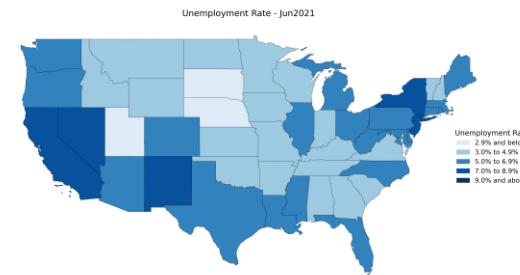
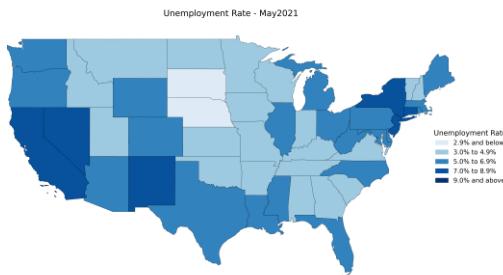
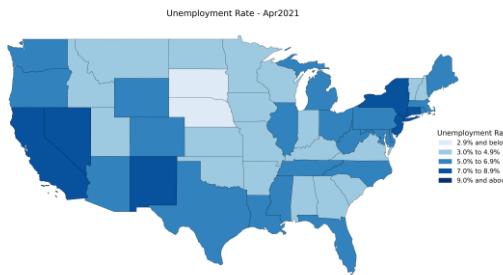
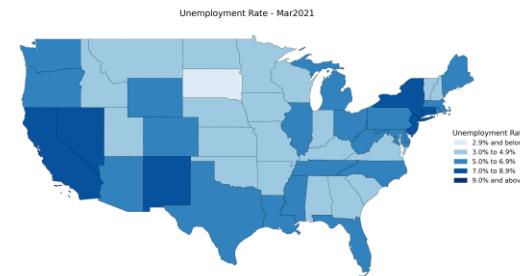
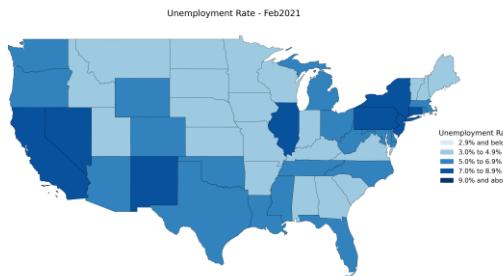
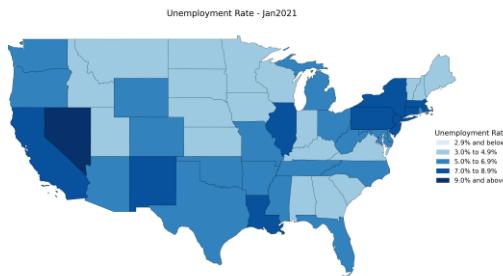
Unemployment Rate - Nov2020



Unemployment Rate - Dec2020

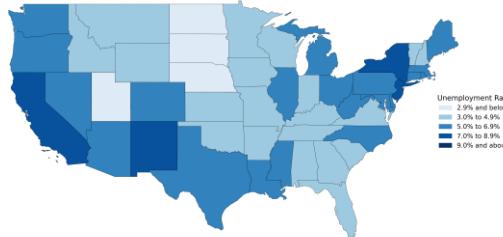


Unemployment rate - 2021

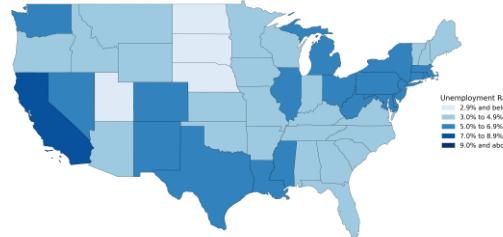


Unemployment rate - 2021

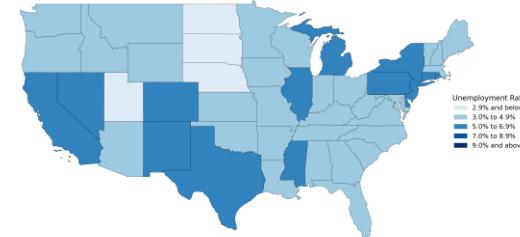
Unemployment Rate - Jul2021



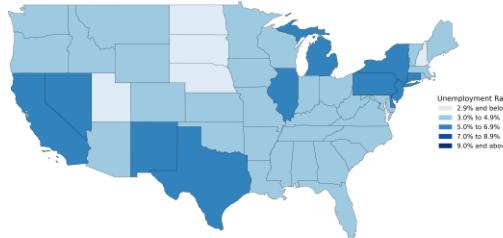
Unemployment Rate - Aug2021



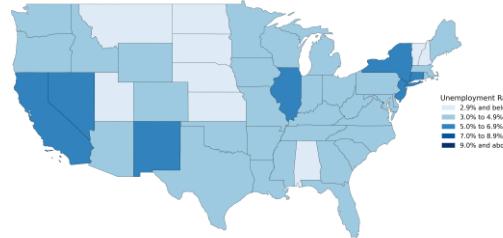
Unemployment Rate - Sep2021



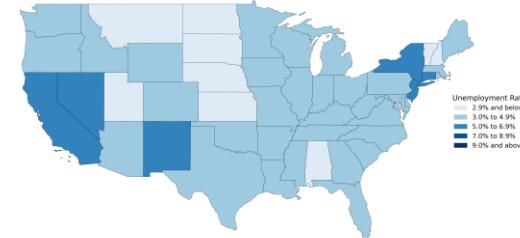
Unemployment Rate - Oct2021



Unemployment Rate - Nov2021



Unemployment Rate - Dec2021



Weekly Earnings

```
Weekly_Earnings = pd.read_excel('/content/file/Weekly_Earnings.xlsx')
display(Weekly_Earnings.head())
print('-----')
print(Weekly_Earnings.info())
print(Weekly_Earnings.isnull().sum())

Weekly_Earnings.to_csv(f'/content/file/Weekly_Earnings.csv', index = False)
display(Weekly_Earnings)
```

	Quarter	Total	White	Black or African American	Asian	Hispanic or Latino
0	Q4 2019	936	967		756	1166
1	Q1 2020	957	980		775	1221
2	Q2 2020	1002	1017		805	1336
3	Q3 2020	994	1008		813	1392
4	Q4 2020	984	1007		792	1261
5	Q1 2021	989	1006		799	1286
6	Q2 2021	990	1012		799	1281
7	Q3 2021	1001	1024		799	1309
8	Q4 2021	1010	1030		805	1384
						799

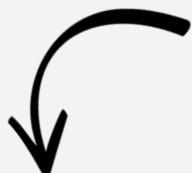
```
quarters = ['Q4 2019', 'Q1 2020', 'Q2 2020', 'Q3 2020', 'Q4 2020', 'Q1 2021', 'Q2 2021', 'Q3 2021', 'Q4 2021']
categories = ['Total', 'White', 'Black or African American', 'Asian', 'Hispanic or Latino']
colors = ['red', 'plum', 'mediumspringgreen', 'cornflowerblue', 'lawngreen']
values = Weekly_Earnings[Weekly_Earnings['Quarter'].isin(quarters)]
plt.figure(figsize=(12, 8))

for i, category in enumerate(categories):
    plt.plot(values['Quarter'], values[category], label = category, color = colors[i])

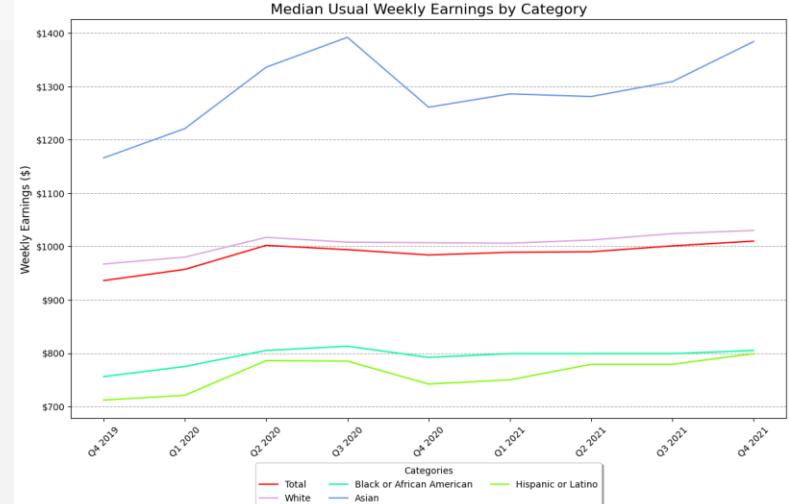
plt.title('Median Usual Weekly Earnings by Category', fontsize = 16)
plt.xlabel('Quarter', fontsize = 12)
plt.ylabel('Weekly Earnings ($)', fontsize = 12)
plt.xticks(rotation=45)
plt.grid(axis = 'y', linestyle = '--', alpha = 0.7, color = 'grey')

plt.gca().get_yaxis().set_major_formatter(plt.FuncFormatter(lambda x, _: f'${int(x)}'))
```

```
plt.legend(title = 'Categories', loc = 'upper center', bbox_to_anchor = (0.5, -0.1), fancybox = True, shadow = True, ncol = 3)
plt.tight_layout()
plt.show()
```



As we can see from the graph, only Asians have had a substantial change in weekly earnings after covid.



Earnings per State

```
Earnings_Per_State = pd.read_excel('/content/file/Earnings_Per_State.xlsx')
display(Earnings_Per_State.head())
print('-----')
print(Earnings_Per_State.info())
print(Earnings_Per_State.isnull().sum())

Earnings_Per_State.to_csv(f'/content/file/Earnings_Per_State.csv', index = False)
display(Earnings_Per_State)
```

① data = Earnings_Per_State

```
USA_data = data.iloc[0, 1:].values
states_data = data.iloc[1:]

states = states_data['State']
values_2020 = states_data['Annual Average Weekly Wage (2020)']
values_2021 = states_data['Annual Average Weekly Wage (2021)'] - states_data['Annual Average Weekly Wage (2020)']
values_2022 = states_data['Annual Average Weekly Wage (2022)'] - states_data['Annual Average Weekly Wage (2021)']

increments_rate = (
    (states_data['Annual Average Weekly Wage (2022)'] - states_data['Annual Average Weekly Wage (2020)']) /
    states_data['Annual Average Weekly Wage (2020)'] * 100
)

colors = ['yellowgreen', 'mediumorchid', 'sandybrown']
bar_width = 0.4

fig, ax = plt.subplots(figsize = (14, 18))
y_positions = np.arange(len(states))

ax.barrh(y_positions, values_2020, color = colors[0], edgecolor = 'black', label = '2020')
ax.barrh(y_positions, values_2021, left = values_2020, color = colors[1], edgecolor = 'black', label = '2021')
ax.barrh(y_positions, values_2022, left = values_2020 + values_2021, color = colors[2], edgecolor = 'black', label = '2022')

for i, increments in enumerate(increments_rate):
    if states.iloc[i] == "Virgin Islands":
        text_position = values_2020.iloc[i] + values_2021.iloc[i] + values_2022.iloc[i] + 10
    elif states.iloc[i] == "California":
        text_position = values_2020.iloc[i] + values_2021.iloc[i] + values_2022.iloc[i] + 50
    else:
        text_position = values_2020.iloc[i] + values_2021.iloc[i] + values_2022.iloc[i] + 10

    ax.text(
        text_position,
        y_positions[i],
        f'{(increments:.1f)}%',
        va='center', ha='left', fontsize=10, color='black'
    )
```

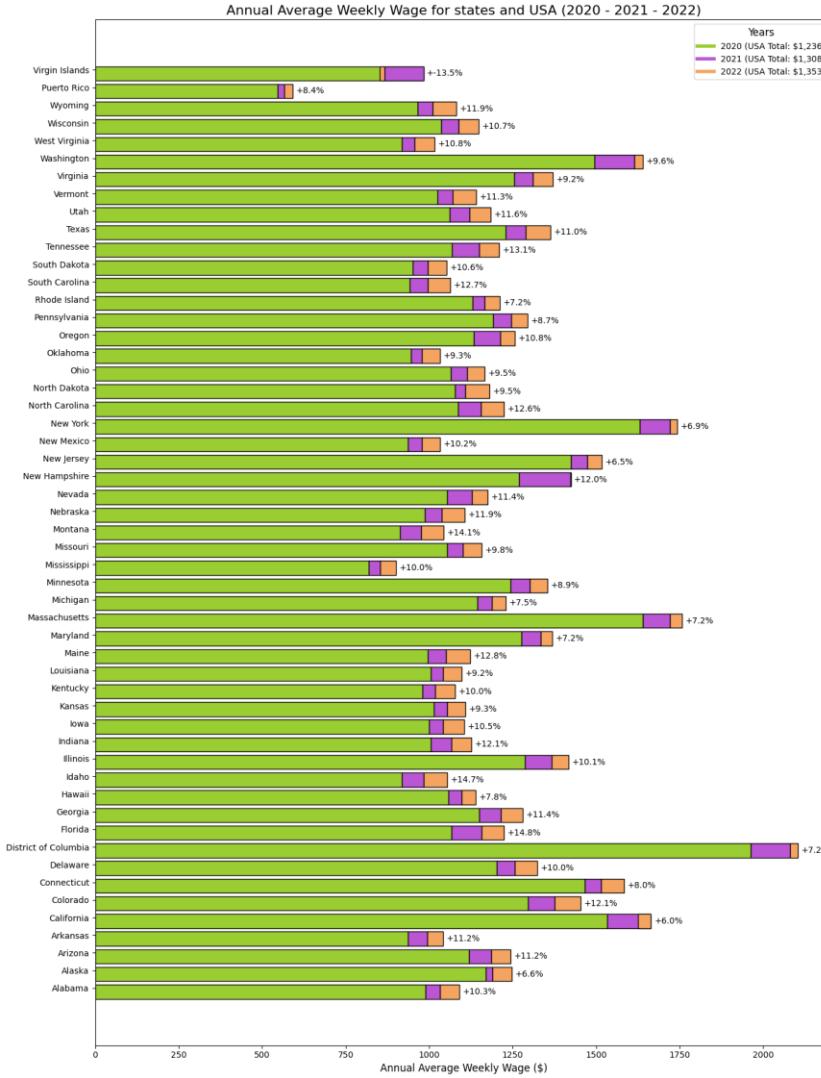
	State	Annual Average Weekly Wage (2020)	Annual Average Weekly Wage (2021)	Annual Average Weekly Wage (2022)
0	U.S. TOTAL	1236	1308	1353
1	Alabama	989	1032	1091
2	Alaska	1170	1189	1247
3	Arizona	1119	1187	1244
4	Arkansas	937	994	1042
...
49	West Virginia	918	956	1017
50	Wisconsin	1037	1089	1148
51	Wyoming	966	1011	1081
52	Puerto Rico	546	566	592
53	Virgin Islands	984	867	851

②

```
ax.set_yticks(y_positions + bar_width/2)
ax.set_yticklabels(states, fontsize = 10, rotation = 0)
ax.set_xlabel('Annual Average Weekly Wage ($)', fontsize = 12)
ax.set_title('Annual Average Weekly Wage for states and USA (2020 - 2021 - 2022)', fontsize = 16)

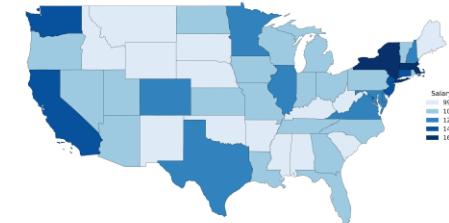
legend_labels = [
    f'2020 (USA Total: ${USA_data[0]:,.0f})',
    f'2021 (USA Total: ${USA_data[1]:,.0f})',
    f'2022 (USA Total: ${USA_data[2]:,.0f})'
]
legend_handles = [
    plt.Line2D([0], [0], color = colors[0], lw = 4),
    plt.Line2D([0], [0], color = colors[1], lw = 4),
    plt.Line2D([0], [0], color = colors[2], lw = 4)
]
ax.legend(legend_handles, legend_labels, title = 'Years', fontsize = 10, title_fontsize = 12)

plt.tight_layout()
plt.show()
```

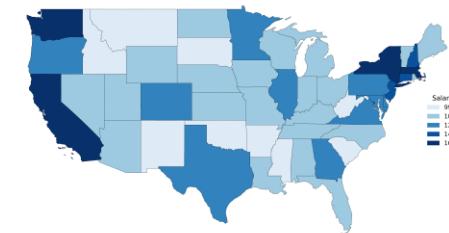


Florida, Idaho and Montana
are the three states with the
highest growth in
percentage of weekly salary
from 2020 to 2022.

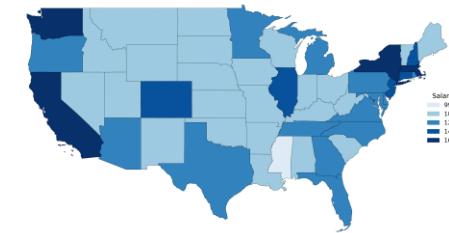
Annual Average Weekly Wage (2020)



Annual Average Weekly Wage (2021)



Annual Average Weekly Wage (2022)



Please note: All three maps
were made with the same
code structure for the
unemployment rate

Consumer Price Index

```
Consumer_Price_Index = pd.read_excel('/content/file/Consumer_Price_Index.xlsx')
display(Consumer_Price_Index.head())
print('-----')
print(Consumer_Price_Index.info())
print(Consumer_Price_Index.isnull().sum())

start_year = 2020
end_year = 2023
months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
month_year_sequence = [f'{month} {year}' for year in range(start_year, end_year + 1) for month in months]

Consumer_Price_Index['Month'] = month_year_sequence[:len(Consumer_Price_Index)]

Consumer_Price_Index.to_csv(f'/content/file/Consumer_Price_Index.csv', index = False)
display(Consumer_Price_Index)
```

Month	All items	Food	Food at home	Food away from home	Energy	Gasoline (all types)	Electricity	Natural gas (piped)	All items less food and energy	Commodities less food and energy commodities	Apparel	New vehicles	Medical care commodities	Services less energy services	Shelter	Medical care services	Education and communication
0 Jan 2020	2.5	1.8	0.7%	3.1%	6.2%	12.8%	0.5%	-3.2%	2.3%	-0.3%	-1.3%	0.1%	1.7%	3.1%	3.3%	5.1%	1.5%
1 Feb 2020	2.3%	0.8%	0.8%	3.0%	2.8%	5.6%	0.6%	-2.0%	2.4%	0.0%	-0.9%	0.4%	1.8%	3.1%	3.3%	5.3%	1.5%
2 Mar 2020	1.5%	0.9%	1.1%	3.0%	-5.7%	-10.2%	0.2%	-2.9%	2.1%	-0.2%	-1.6%	-0.4%	1.3%	2.8%	3.0%	5.5%	1.5%
3 Apr 2020	0.3%	0.5%	4.1%	2.8%	-17.7%	-32.0%	0.2%	-1.9%	1.4%	-0.9%	-5.7%	-0.6%	0.7%	2.2%	2.6%	5.8%	1.6%
4 May 2020	0.1%	0.0%	4.8%	2.9%	-18.9%	-33.8%	-0.2%	-0.3%	1.2%	-1.0%	-7.9%	-0.3%	0.8%	2.0%	2.5%	5.9%	1.6%
...
43 Aug 2023	3.7%	0.3%	3.0%	6.5%	-3.6%	-3.3%	2.1%	-16.5%	4.3%	0.2%	3.1%	2.9%	4.5%	5.9%	7.3%	-2.1%	1.0%
44 Sep 2023	3.7%	0.7%	2.4%	6.0%	-0.5%	3.0%	2.8%	-19.9%	4.1%	0.0%	2.3%	2.5%	4.2%	5.7%	7.2%	-2.6%	1.0%
45 Oct 2023	3.2%	0.3%	2.1%	5.4%	-4.5%	-5.3%	2.4%	-15.8%	4.0%	0.1%	2.6%	1.9%	4.7%	5.5%	6.7%	-2.0%	0.9%
46 Nov 2023	3.1%	0.2%	1.7%	5.3%	-5.4%	-6.9%	3.4%	-10.4%	4.0%	0.0%	1.1%	1.3%	5.0%	5.5%	6.5%	-0.9%	-0.1%
47 Dec 2023	3.4%	0.7%	1.3%	5.2%	-2.0%	-1.9%	3.3%	-13.8%	3.9%	0.2%	1.0%	1.0%	4.7%	5.3%	6.2%	-0.5%	-0.1%



Since there are so many categories regarding the consumer price index, we have only analyzed the first column: 'All items'.



```
data = Consumer_Price_Index

data['All items'] = data['All items'].replace((r'%'': ''), regex=True).astype(float)

months = ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
colors = {2020: 'black', 2021: 'chocolate', 2022: 'forestgreen', 2023: 'mediumblue'}

plt.figure(figsize = (15, 7))

for year in range(2020, 2024):
    all_months = [f'{month} {year}' for month in months]
    data_year = data[data['Month'].isin(all_months)]

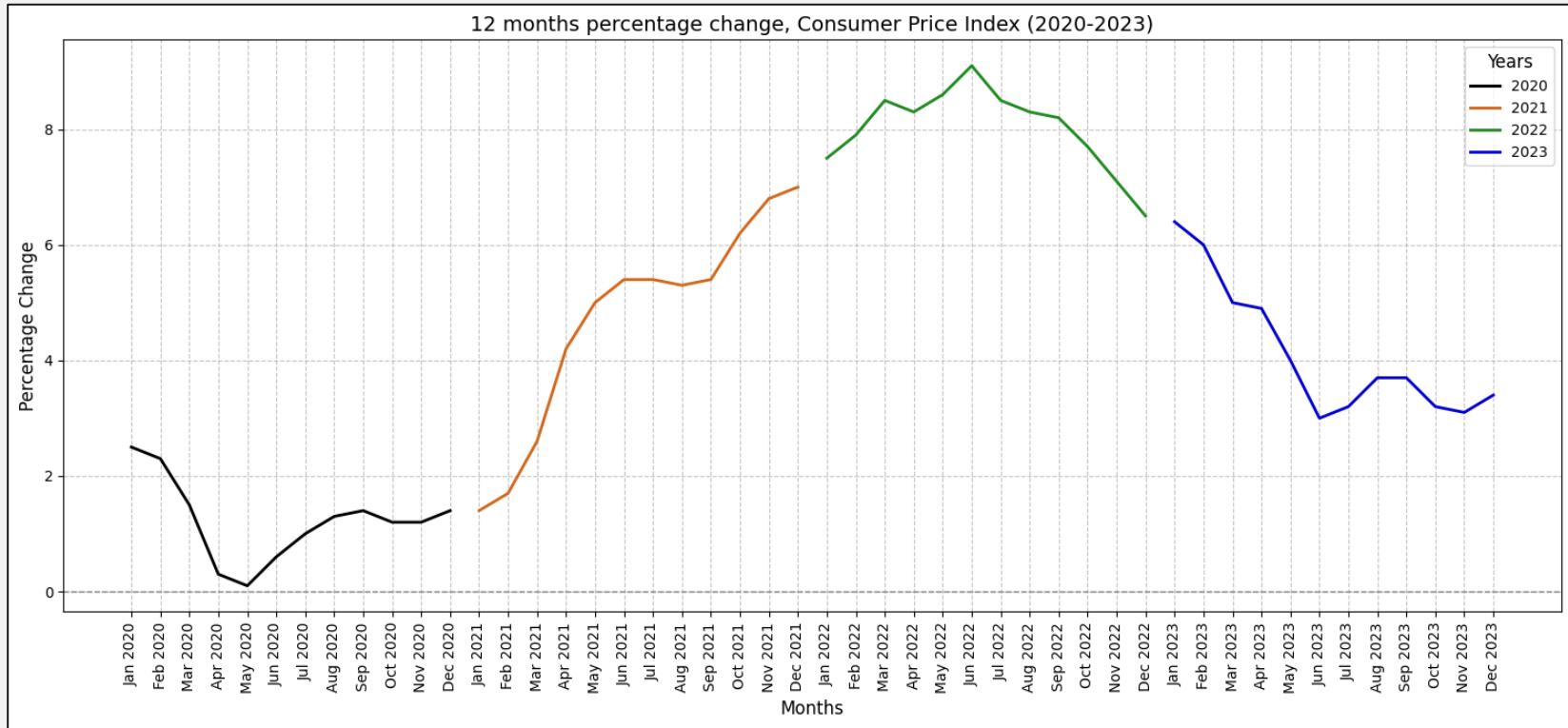
    x_labels = all_months
    y_values = data_year['All items']

    plt.plot(x_labels, y_values, label = f'(year)', color = colors[year], linewidth = 2)

plt.title('12 months percentage change, Consumer Price Index (2020-2023)', fontsize = 14)
plt.xlabel('Months', fontsize = 12)
plt.ylabel('Percentage Change', fontsize = 12)
plt.xticks(rotation = 90, fontsize = 10)
plt.axhline(y = 0, color = 'grey', linestyle = '--', linewidth = 1)
plt.legend(title = 'Years', fontsize = 10, title_fontsize = 12)
plt.grid(visible = True, linestyle = '--', alpha = 0.7)

plt.tight_layout()
plt.show()
```

Consumer Price Index - All items



We took a look at the consumer price index for each month between 2020 and 2023 to see all the changes and how it has changed with the 2020 elections.





Voting behavior at geographic level

With a first hint of linear regression (chapter 5)

Voter turnout in the US - Data cleaning

Data cleaning process for the voter turnout dataset found from the US Census Bureau (2020).

```
input_file = ('/content/file/table04a.xlsx')
df = pd.read_excel(input_file)
display(df.head())
print('-----')
print(df.info())
print(df.isnull().sum())
```

Table with row headers in column A, and column headers in rows 5 through 6.															
0	Table 4a. Reported Voting and Registration for...		Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	Unnamed: 11	Unnamed: 12	
1	(In thousands)		NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2			NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3	STATE		Total population	Total citizen population	Registered	NaN	NaN	NaN	NaN	Voted	NaN	NaN	NaN	NaN	
4			NaN	NaN	NaN	Total registered	Percent registered\n(\% total)	Margin of error 1	Percent registered\n(Citizen)	Margin of error 1	Total voted	Percent voted\n(\% total)	Margin of error 1	Percent voted\n(Citizen)	Margin of error 1
<class 'pandas.core.frame.DataFrame'>															
RangeIndex: 64 entries, 0 to 63															
Data columns (total 13 columns):															
# Column															

0 Table with row headers in column A, and column headers in rows 5 through 6.															
1 Unnamed: 1															
2 Unnamed: 2															
3 Unnamed: 3															
4 Unnamed: 4															
5 Unnamed: 5															
6 Unnamed: 6															
7 Unnamed: 7															
8 Unnamed: 8															
9 Unnamed: 9															
10 Unnamed: 10															
11 Unnamed: 11															
12 Unnamed: 12															
dtypes: object(13)															
memory usage: 6.6 KB															
Table with row headers in column A, and column headers in rows 5 through 6.															
Unnamed: 1															
Unnamed: 2															
Unnamed: 3															
Unnamed: 4															
Unnamed: 5															
Unnamed: 6															
Unnamed: 7															
Unnamed: 8															
Unnamed: 9															
Unnamed: 10															
Unnamed: 11															
Unnamed: 12															
dtype: int64															



Voter turnout in the US - Data cleaning

```
df = pd.read_excel(input_file, skiprows = 4)

original_header = df.columns.tolist()
first_row = df.iloc[0].tolist()

new_header = [col if pd.isna(val) else val for col, val in zip(original_header, first_row)]

df.columns = new_header
df = df[1:]

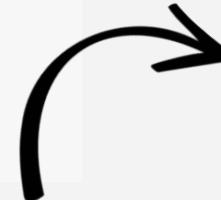
df.reset_index(drop=True, inplace=True)

df = df.dropna()

df.to_csv("cleaned_table04a.csv", index=False)

display(df)
```

In particular, the data contain information regarding the total citizens and total voted in each US States



	STATE	Total population	Total citizen population	Total registered	Percent registered\n(Total)	Margin of error 1	Percent registered\n(Citizen)	Margin of error 1	Total voted	Percent voted\n(Total)	Margin of error 1	Percent voted\n(Citizen)	Margin of error 1
0	UNITED STATES	252274.0	231593.0	168308	66.7	0.4	72.7	0.4	154628	61.3	0.4	66.8	0.4
1	ALABAMA	3769.0	3716.0	2527	67	3.1	68	3.1	2247	59.6	3.3	60.5	3.3
2	ALASKA	528.0	516.0	383	72.6	3.2	74.2	3.1	330	62.4	3.4	63.8	3.4
3	ARIZONA	5638.0	5075.0	3878	68.8	2.5	76.4	2.5	3649	64.7	2.6	71.9	2.6
4	ARKANSAS	2283.0	2195.0	1361	59.6	3.4	62	3.4	1186	51.9	3.4	54	3.5
...
47	VIRGINIA	6481.0	5974.0	4541	70.1	2.4	76	2.3	4275	66	2.5	71.5	2.4
48	WASHINGTON	5993.0	5389.0	4029	67.2	2.5	74.8	2.4	3854	64.3	2.6	71.5	2.5
49	WEST VIRGINIA	1397.0	1379.0	928	66.4	3.4	67.3	3.4	773	55.3	3.6	56.1	3.6
50	WISCONSIN	4538.0	4421.0	3391	74.7	2.7	76.7	2.6	3253	71.7	2.8	73.6	2.7
51	WYOMING	436.0	427.0	296	67.9	3.4	69.3	3.4	280	64.1	3.5	65.5	3.5



Voter turnout in the US - By state

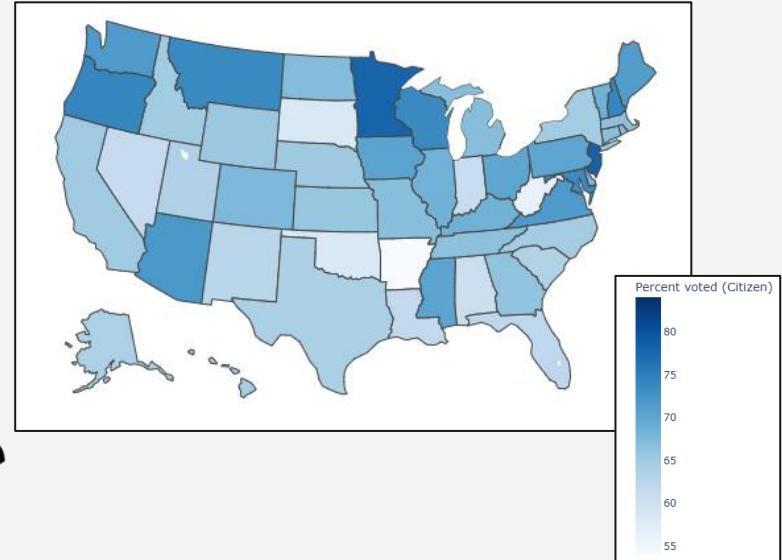
```
affluence_df = pd.read_csv('cleaned_table04a.csv')

affluence_df.rename(columns={'STATE': 'State'}, inplace=True)
affluence_df['State'] = affluence_df['State'].str.title()

state_abbreviations = {
    'Alabama': 'AL', 'Alaska': 'AK', 'Arizona': 'AZ', 'Arkansas': 'AR', 'California': 'CA',
    'Colorado': 'CO', 'Connecticut': 'CT', 'Delaware': 'DE', 'Florida': 'FL', 'Georgia': 'GA',
    'Hawaii': 'HI', 'Idaho': 'ID', 'Illinois': 'IL', 'Indiana': 'IN', 'Iowa': 'IA', 'Kansas': 'KS',
    'Kentucky': 'KY', 'Louisiana': 'LA', 'Maine': 'ME', 'Maryland': 'MD', 'Massachusetts': 'MA',
    'Michigan': 'MI', 'Minnesota': 'MN', 'Mississippi': 'MS', 'Missouri': 'MO', 'Montana': 'MT',
    'Nebraska': 'NE', 'Nevada': 'NV', 'New Hampshire': 'NH', 'New Jersey': 'NJ', 'New Mexico': 'NM',
    'New York': 'NY', 'North Carolina': 'NC', 'North Dakota': 'ND', 'Ohio': 'OH', 'Oklahoma': 'OK',
    'Oregon': 'OR', 'Pennsylvania': 'PA', 'Rhode Island': 'RI', 'South Carolina': 'SC',
    'South Dakota': 'SD', 'Tennessee': 'TN', 'Texas': 'TX', 'Utah': 'UT', 'Vermont': 'VT',
    'Virginia': 'VA', 'Washington': 'WA', 'West Virginia': 'WV', 'Wisconsin': 'WI', 'Wyoming': 'WY'
}

affluence_df['State Abbreviation'] = affluence_df['State'].map(state_abbreviations)

fig = px.choropleth(
    affluence_df,
    locations='State Abbreviation',
    locationmode='USA-states',
    color='Percent voted\n(Citizen)',
    color_continuous_scale='Blues',
    scope='usa',
    title='Voter Turnout in the 2020 U.S. Elections - by State',
)
fig.show()
```



The map shows voter turnout by US states, in percentage.

Majority of the States registered greater than 60% of the population to the turnout.

Voter turnout in the US - Data cleaning

```
input_file = ('/content/file/table04b.xlsx')
df = pd.read_excel(input_file, skiprows=4)

original_header = df.columns.tolist()
first_row = df.iloc[0].tolist()

new_header = [col if pd.isna(val) else val for col, val in zip(original_header, first_row)]

df.columns = new_header
df = df[1:]

df.reset_index(drop=True, inplace=True)

df["STATE"] = df["STATE"].fillna(method='ffill')

df = df.dropna()

df.to_csv("cleaned_table04b.csv", index=False)

display(df)
```

	STATE	Sex, Race, and Hispanic-Origin	Total population	Total citizen population	Total registered	Percent registered\n(Total)	Margin of error 1	Percent registered\n(Citizen)	Margin of error 1	Total voted	Percent voted\n(Total)	Margin of error 1	Percent voted\n(Citizen)	Margin of error 1
0	US	Total	252274.0	231593	168308	66.7	0.4	72.7	0.4	154628	61.3	0.4	66.8	0.4
1	US	Male	121870.0	111485	79340	65.1	0.5	71.2	0.5	72474	59.5	0.5	65	0.5
2	US	Female	130404.0	120108	88968	68.2	0.5	74.1	0.5	82154	63	0.5	68.4	0.5
3	US	White alone	195227.0	181891	134889	69.1	0.4	74.2	0.4	124301	63.7	0.4	68.3	0.4
4	US	White non-Hispanic alone	157442.0	154827	118389	75.2	0.4	76.5	0.4	109830	69.8	0.4	70.9	0.4
...
567	WYOMING	Asian alone	2.0	-	-	B	B	B	B	-	B	B	B	B
568	WYOMING	Hispanic (of any race)	40.0	38	23	B	B	B	B	21	B	B	B	B
569	WYOMING	White alone or in combination	422.0	416	290	68.6	3.5	69.6	3.5	273	64.7	3.6	65.7	3.6
570	WYOMING	Black alone or in combination	4.0	3	3	B	B	B	B	3	B	B	B	B
571	WYOMING	Asian alone or in combination	4.0	2	2	B	B	B	B	2	B	B	B	B

This dataset is similar to the previous one, analyzing voter turnout for each US state, except that this dataset also contains information regarding ethnic groups within each state.



Voter turnout in the US - By ethnicity

```
file_path = 'cleaned_table04b.csv'
data = pd.read_csv(file_path)

# Ensure numeric conversion for Total population and Total voted
data['Total population'] = pd.to_numeric(data['Total population'], errors='coerce').fillna(0).astype(int)
data['Total voted'] = pd.to_numeric(data['Total voted'], errors='coerce').fillna(0).astype(int)

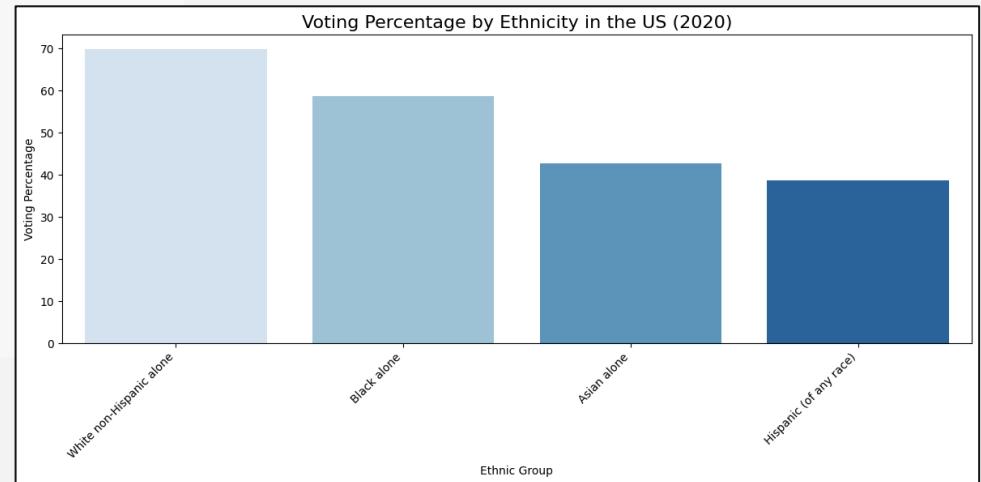
us_data = data[data['STATE'] == 'US']
criteria = [
    'White non-Hispanic alone',
    'Black alone',
    'Asian alone',
    'Hispanic (of any race)'
]
us_data = us_data[us_data['Sex, Race, and Hispanic-Origin'].isin(criteria)]

ethnicity_data = us_data[['Sex, Race, and Hispanic-Origin', 'Total population', 'Total voted']]
ethnicity_data['Voting Percentage'] = (
    ethnicity_data['Total voted'] / ethnicity_data['Total population'] * 100
)

plt.figure(figsize=(12, 6))
sns.barplot(
    data=ethnicity_data,
    x='Sex, Race, and Hispanic-Origin',
    y='Voting Percentage',
    palette='Blues'
)
plt.title('Voting Percentage by Ethnicity in the US (2020)', fontsize=16)
plt.xticks(rotation=45, ha='right')
plt.ylabel('Voting Percentage')
plt.xlabel('Ethnic Group')

plt.tight_layout()
plt.show()
```

We selected the four ethnic groups that can represent the entire US population.



Linear regression – for choosing swing states

```
df = pd.read_excel('/content/file/REGRESSIONE.xlsx')
df = df.set_index('STATO')
display(df)
print('-----')
print(df.info())
```

STATO	% DEM 2020	Dem Historical Trend	% REP 2020	Rep Historical Trend	Avg Age	Avg Hourly Earnings (\$)	Avg Unemployment Rate	% non-White
Alabama	0.3657	0	0.6203	1	39.6	30.53	0.065818	0.369
Alaska	0.4277	0	0.5283	1	35.9	37.26	0.084636	0.425
Arizona	0.4936	0	0.4906	0	38.8	33.40	0.079364	0.466
Arkansas	0.3478	0	0.6240	1	38.9	28.97	0.062909	0.315
California	0.6348	1	0.3432	0	37.8	40.24	0.103000	0.653
...
Virginia	0.5411	1	0.4400	0	38.9	34.58	0.066273	0.414
Washington	0.5797	1	0.3877	0	38.4	41.62	0.087000	0.362
West Virginia	0.2970	0	0.6863	1	42.9	29.31	0.084636	0.109
Wisconsin	0.4945	0	0.4882	0	40.4	33.64	0.066000	0.214
Wyoming	0.2655	0	0.6994	1	39.1	31.44	0.059000	0.186

51 rows × 8 columns

```
<class 'pandas.core.frame.DataFrame'>
Index: 51 entries, Alabama to Wyoming
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype  
 ---  -- 
 0   % DEM 2020      51 non-null    float64
 1   Dem Historical Trend  51 non-null  int64  
 2   % REP 2020      51 non-null    float64
 3   Rep Historical Trend  51 non-null  int64  
 4   Avg Age         51 non-null    float64
 5   Avg Hourly Earnings ($) 51 non-null  float64
 6   Avg Unemployment Rate 51 non-null  float64
 7   % non-White     51 non-null    float64
dtypes: float64(6), int64(2)
memory usage: 3.6+ KB
None
```



The following dataset shows for each state:

- **The percentage of non-white people;**
- **The unemployment rate;**
- **The average weekly earnings in dollars;**
- **The average age;**
- **A historical variable for both parties;**
- **The percentage of votes for both parties.**

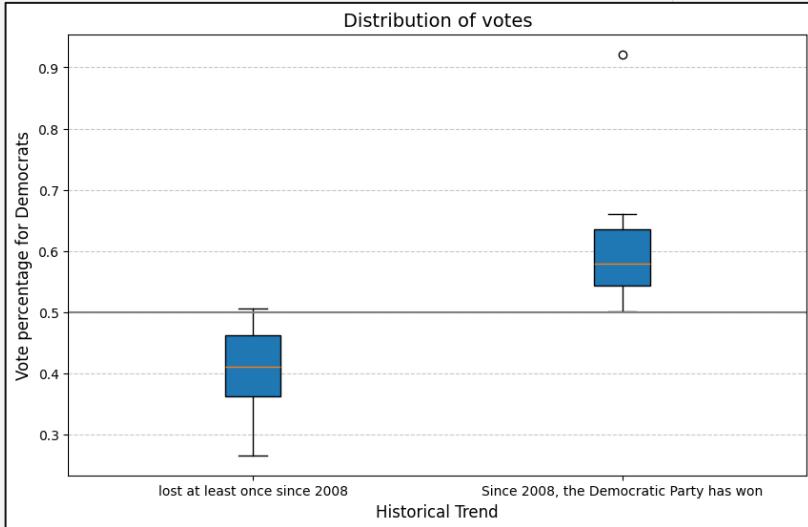


Linear regression – for choosing swing states

```
grouped_data = df.groupby('Dem Historical Trend')[ '% DEM 2020' ].apply(list)
plt.boxplot([grouped_data[0], grouped_data[1]],
            labels=["lost at least once since 2008", "Since 2008, the Democratic Party has won"],
            patch_artist=True)

plt.title("Distribution of votes", fontsize=14)
plt.xlabel("Historical Trend", fontsize=12)
plt.ylabel("Vote percentage for Democrats", fontsize=12)
plt.axhline(y = 0.5, color = 'grey')
plt.grid(axis='y', linestyle='--', alpha=0.7)

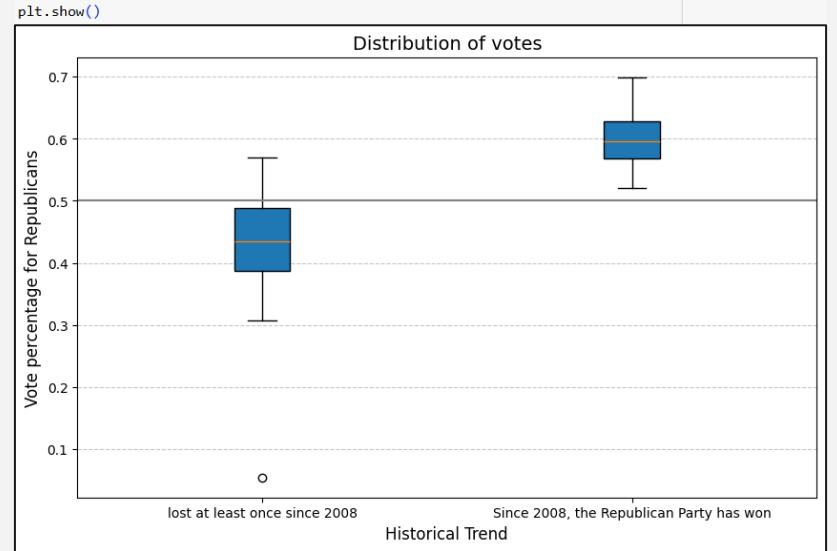
plt.show()
```



```
grouped_data = df.groupby('Rep Historical Trend')[ '% REP 2020' ].apply(list)
plt.boxplot([grouped_data[0], grouped_data[1]],
            labels=["lost at least once since 2008", "Since 2008, the Republican Party has won"],
            patch_artist=True)

plt.title("Distribution of votes", fontsize=14)
plt.xlabel("Historical Trend", fontsize=12)
plt.ylabel("Vote percentage for Republicans", fontsize=12)
plt.axhline(y = 0.5, color = 'grey')
plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.show()
```



The historical variable takes the value of 1 if the reference party has always won in the state from 2008 to 2020 (4 consecutive elections), and 0 if at least once another party won.



Which are the swing states?

```
lst = df[(df['Dem Historical Trend'] == 0) &
         (df['Rep Historical Trend'] == 0) &
         ((df['% DEM 2020'].between(0.48, 0.52)) |
          (df['% REP 2020'].between(0.48, 0.52))).index.tolist()]

lst
['Arizona',
 'Florida',
 'Georgia',
 'Michigan',
 'North Carolina',
 'Pennsylvania',
 'Wisconsin']
```

We considered a swing state to be one where the percentage of votes are uncertain.

```
y = df['% DEM 2020']
X = df[['Dem Historical Trend']]
X = sm.add_constant(X)

fit = sm.OLS(y, X).fit()
print(fit.summary())
```

OLS Regression Results						
		Dep. Variable:		% DEM 2020		
		Model:		R-squared: 0.614		
		Method:		Adj. R-squared: 0.606		
		Least Squares		F-statistic: 78.03		
		Date: Tue, 21 Jan 2025		Prob (F-statistic): 1.04e-11		
		Time: 10:16:27		Log-Likelihood: 60.373		
		No. Observations: 51		AIC: -116.7		
		Df Residuals: 49		BIC: -112.9		
		Df Model: 1				
		Covariance Type: nonrobust				
		coef	std err	t	P> t	[0.025 0.975]
const	0.4082	0.014	29.585	0.000	0.380	0.436
Dem Historical Trend	0.1899	0.022	8.834	0.000	0.147	0.233
====						
Omnibus:	27.801	Durbin-Watson:	2.013			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	71.783			
Skew:	1.454	Prob(JB):	2.59e-16			
Kurtosis:	8.032	Cond. No.	2.46			
====						
Notes:						
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.						



Turnout by ethnicity and swing states

```
# Continue from US-level ...
#
# -----
# Step 2: Analyze data for swing states
swing_states = ['ARIZONA', 'GEORGIA', 'MICHIGAN', 'NORTH CAROLINA', 'NEVADA', 'PENNSYLVANIA', 'WISCONSIN']
data_swings_states = data[data['STATE'].isin(swing_states)]

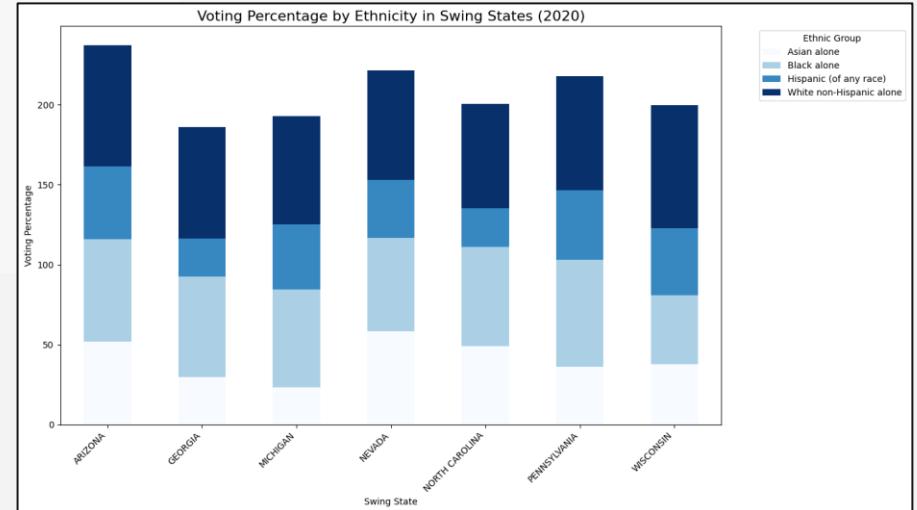
swing_ethnicity_data = data_swings_states[
    ['STATE', 'Sex, Race, and Hispanic-Origin', 'Total population', 'Total voted']
]
swing_ethnicity_data = swing_ethnicity_data[swing_ethnicity_data['Sex, Race, and Hispanic-Origin'].isin(criteria)]
swing_ethnicity_data['Voting Percentage'] = (
    swing_ethnicity_data['Total voted'] / swing_ethnicity_data['Total population'] * 100
)

swing_ethnicity_data['Voting Percentage'] = pd.to_numeric(swing_ethnicity_data['Voting Percentage'], errors='coerce')

stacked_data = swing_ethnicity_data.pivot(
    index='STATE',
    columns='Sex, Race, and Hispanic-Origin',
    values='Voting Percentage'
)

stacked_data.plot(kind='bar', stacked=True, figsize=(14, 8), cmap='Blues')
plt.title('Voting Percentage by Ethnicity in Swing States (2020)', fontsize=16)
plt.xticks(rotation=45, ha='right')
plt.ylabel('Voting Percentage')
plt.xlabel('Swing State')
plt.legend(title='Ethnic Group', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()
```

After the selection of the most interesting States, we firstly calculated the affluences of various ethnicities at the turnout.

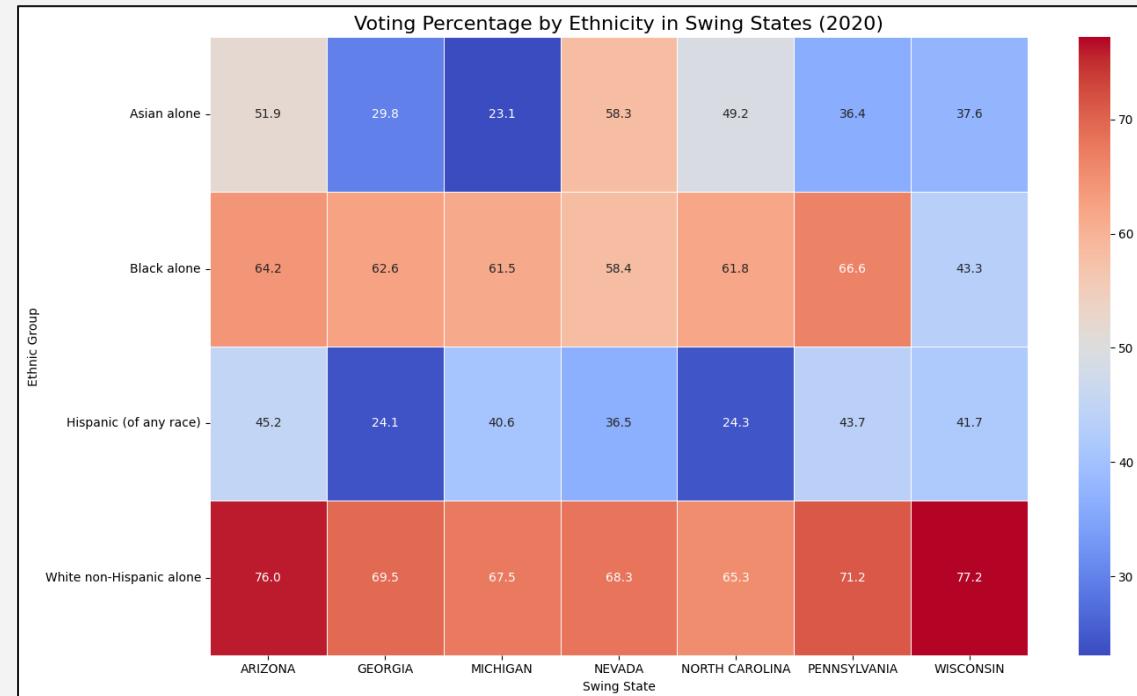


Turnout by ethnicity and swing states

```
# As alternative, an heatmap of the swing states turnout by ethnicites
#
# -----
# Heatmap visualization
heatmap_data = swing_ethnicity_data.pivot(
    index='Sex, Race, and Hispanic-Origin',
    columns='STATE',
    values='Voting Percentage'
)

plt.figure(figsize=(14, 8))
sns.heatmap(
    heatmap_data,
    annot=True,
    fmt='.1f',
    cmap='coolwarm',
    linewidths=0.5
)
plt.title('Voting Percentage by Ethnicity in Swing States (2020)', fontsize=16)
plt.xlabel('Swing State')
plt.ylabel('Ethnic Group')
plt.tight_layout()
plt.show()

# -----
swing_ethnicity_data.to_csv('swing_states_ethnicity_analysis.csv', index=False)
```



Voting preferences by ethnicities and swing states

①

```
original_csv_path = "swing_states_ethnicity_analysis.csv"
aggregated_csv_path = "/content/file/aggregated_voting_percentages.csv"

original_data = pd.read_csv(original_csv_path)
aggregated_data = pd.read_csv(aggregated_csv_path)

original_data.loc[
    original_data['Sex, Race, and Hispanic-Origin'] == 'White non-Hispanic alone',
    ['Biden (%)', 'Trump (%)', 'Other (%)']
] = aggregated_data.loc[aggregated_data['Sex, Race, and Hispanic-Origin'] == 'White', ['Biden (%)', 'Trump (%)', 'Other (%)']].values

non_white_categories = ['Black alone', 'Asian alone', 'Hispanic (of any race)']
for category in non_white_categories:
    original_data.loc[
        original_data['Sex, Race, and Hispanic-Origin'] == category,
        ['Biden (%)', 'Trump (%)', 'Other (%)']
    ] = aggregated_data.loc[aggregated_data['Sex, Race, and Hispanic-Origin'] == 'Non-White', ['Biden (%)', 'Trump (%)', 'Other (%)']].values

final_csv_path = "updated_merged_swing_states_ethnicity_analysis.csv"
original_data.to_csv(final_csv_path, index=False)
```

②

```
file_path = 'updated_merged_swing_states_ethnicity_analysis.csv'
data = pd.read_csv(file_path)

non_white_categories = [
    "Black alone",
    "Asian alone",
    "Hispanic (of any race)"
]

data['Non-white'] = data['Sex, Race, and Hispanic-Origin'].apply(
    lambda x: "Non-white" if x in non_white_categories else "White"
)

# ATTENTION: the following is an hypothetic scenario, based on the own datasets
# Modify percentages to be proportionate to each ethnicity's total population within each state

data['Biden votes'] = 0
data['Trump votes'] = 0
data['Other votes'] = 0
```

```
def calculate_votes(group):
    for index, row in group.iterrows():
        biden_votes = (row['Total voted'] * row['Biden (%)']) / 100
        trump_votes = (row['Total voted'] * row['Trump (%)']) / 100
        other_votes = (row['Total voted'] * row['Other (%)']) / 100

        group.at[index, 'Biden votes'] = biden_votes
        group.at[index, 'Trump votes'] = trump_votes
        group.at[index, 'Other votes'] = other_votes

    return group
```

③

```
data = data.groupby(['STATE']).apply(calculate_votes)

# visualize the updated data in a log scale
states = data['STATE'].unique()
for state in states:
    state_data = data[data['STATE'] == state]
    plt.figure(figsize=(8, 5))

    x = range(len(state_data))

    # Logarithmic transformation
    biden_votes_log = np.log10(state_data['Biden votes'])
    trump_votes_log = np.log10(state_data['Trump votes'])
    other_votes_log = np.log10(state_data['Other votes'])

    plt.bar(x, biden_votes_log, width=0.3, label='Biden (log10)', align='center')
    plt.bar([p + 0.3 for p in x], trump_votes_log, width=0.3, label='Trump (log10)', align='center')
    plt.bar([p + 0.6 for p in x], other_votes_log, width=0.3, label='Other (log10)', align='center')

    plt.title("Log-transformed Voting Data by Ethnicity in (state)")
    plt.xlabel('Ethnicity')
    plt.ylabel('Log-transformed Votes')
    plt.xticks([p + 0.3 for p in x], state_data['Sex, Race, and Hispanic-Origin'], rotation=45)
    plt.legend()
    plt.tight_layout()
    plt.show()
```

Starting from merging two different dataset (1): US Census Bureau and AP News.

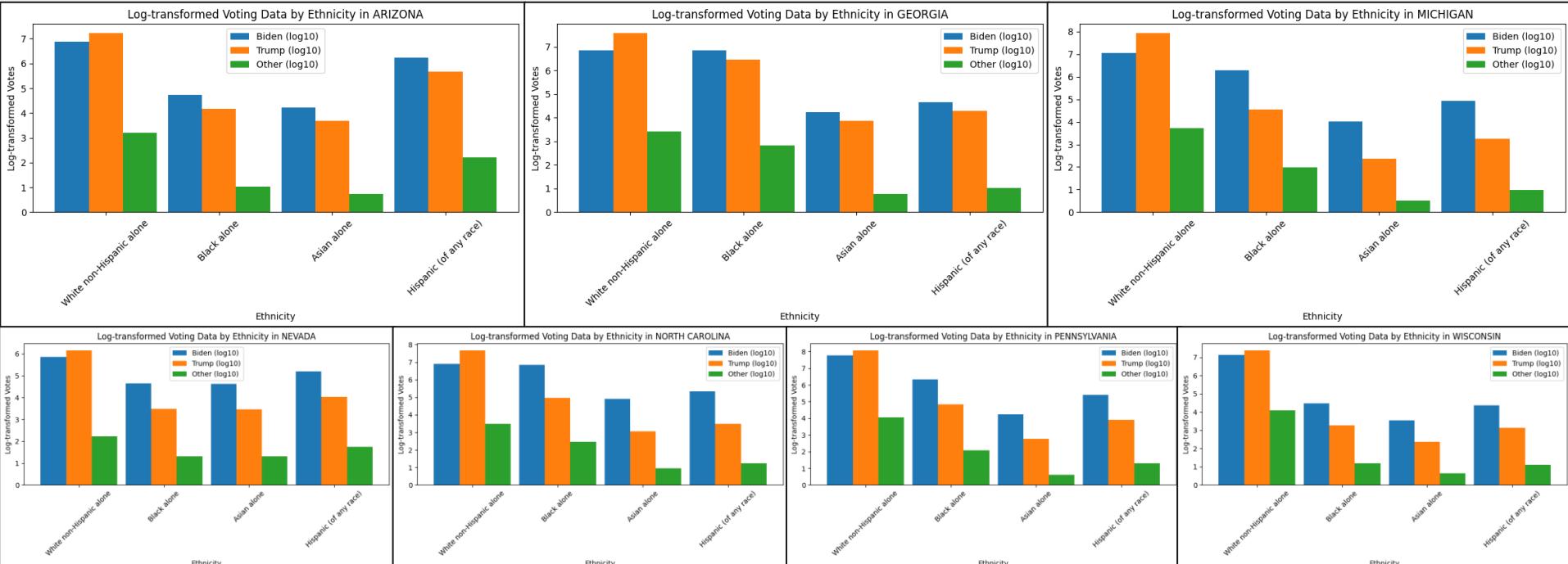


Then we extracted the voting preferences of various ethnicities from the various swing states.



Up to computing the Candidate votes based on the voting population of each swing state (2 - 3).

Voting preferences by ethnicities and swing states



Please note: Results are presented on the log-scale for a better vision





Electoral topics

- Abortion
- Mexico wall
- Gun Laws
- Climate change
- Import taxes

Social main topics- Data cleaning

```
file_path='/content/file/MainThemeData.xlsx'  
df = pd.read_excel(file_path)
```

```
display(df)
```

	State	Division	Abortion legal	Abortion illegal	US-Mexico Wall pro	US-Mexico Wall against	Gun laws more strict	Gun laws no more strict	Climate change concerned	Climate change no concerned	import tax favor	import tax oppose
0	National	White	0.57	0.43	0.54	0.46	0.48	0.52	0.85	0.35	0.62	0.38
1	National	Non-white	0.57	0.33	0.32	0.88	0.88	0.32	0.77	0.23	0.53	0.47
2	National	Men	0.57	0.43	0.54	0.42	0.47	0.52	0.85	0.35	0.61	0.39
3	National	Women	0.61	0.39	0.44	0.54	0.59	0.48	0.75	0.25	0.59	0.41
4	National	No college	0.56	0.44	0.53	0.47	0.48	0.52	0.87	0.33	0.62	0.38
...
75	Wisconsin	College	0.59	0.41	0.42	0.58	0.82	0.38	0.71	0.29	0.59	0.41
76	Wisconsin	under 45	0.62	0.38	0.38	0.64	0.54	0.46	0.89	0.31	0.53	0.47
77	Wisconsin	over 45	0.53	0.47	0.49	0.51	0.49	0.51	0.84	0.36	0.68	0.32
78	Wisconsin	Biden voter	0.83	0.17	0.08	0.92	0.84	0.16	0.94	0.06	0.52	0.48
79	Wisconsin	Trump voter	0.28	0.72	0.88	0.12	0.22	0.78	0.42	0.58	0.75	0.25

80 rows × 12 columns

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 80 entries, 0 to 79  
Data columns (total 12 columns):  
 # Column          Non-Null Count  Dtype     
 ---  ...  
 0   State           80 non-null    object    
 1   Division        80 non-null    object    
 2   Abortion legal  79 non-null    float64   
 3   Abortion illegal 79 non-null    float64   
 4   US-Mexico Wall pro 79 non-null  float64   
 5   US-Mexico Wall against 79 non-null  float64   
 6   Gun laws more strict 79 non-null  float64   
 7   Gun laws no more strict 79 non-null  float64   
 8   Climate change concerned 79 non-null  float64   
 9   Climate change no concerned 79 non-null  float64   
 10  import tax favor 79 non-null    float64   
 11  import tax oppose 79 non-null    float64  
dtypes: float64(10), object(2)  
memory usage: 7.6+ KB  
None  
State          8  
Division        8  
Abortion legal 1  
Abortion illegal 1  
US-Mexico Wall pro 1  
US-Mexico Wall against 1  
Gun laws more strict 1  
Gun laws no more strict 1  
Climate change concerned 1  
Climate change no concerned 1  
import tax favor 1  
import tax oppose 1  
dtype: int64
```



This dataset from AP news contains information regarding the social topics that were most discussed during the 2020 elections: the most relevant ones that were a key point for winning the elections.



Social main topics- Data cleaning

```
① def save_and_show_plot(title, ax):
    ax.set_title(title, fontsize=14)
    ax.set_xlabel('')
    ax.set_ylabel('% Favorable', fontsize=12)
    ax.set_xlim(0, 1)
    plt.xticks(rotation=45, ha='right')
    plt.tight_layout()
    plt.show()

def plot_bars(ax, x, width, data, categories, colors, label_key):
    for i, category in enumerate(categories):
        subset = data[data[label_key] == category]
        ax.bar([pos + i * width for pos in x], subset['Percentage'],
               width=width, label=category, color=colors[i])

def generic_analysis(data, x_labels, title, categories, colors, label_key):
    fig, ax = plt.subplots(figsize=(12, 8))
    x = range(len(x_labels))
    width = 0.8 / len(categories)
    if colors is None:
        colors = plt.cm.tab10.colors[:len(categories)]
    plot_bars(ax, x, width, data, categories, colors, label_key)
    ax.set_xticks([pos + width * (len(categories) - 1) / 2 for pos in x])
    ax.set_xticklabels(x_labels)
    ax.legend(title=label_key)
    save_and_show_plot(title, ax)
```

```
② pair_colors = {
    ('White', 'Non-white'): ('#1f77b4', '#aec7e8'),
    ('Men', 'Women'): ('#ff7f0e', '#ffb78'),
    ('No college', 'College'): ('#2ca02c', '#98df8a'),
    ('under 45', 'over 45'): ('#d62728', '#ff9896'),
    ('Biden voter', 'Trump voter'): ('#0015BC', '#DC143C')
}

colors = {
    'White': '#1f77b4',
    'Non-white': '#aec7e8',
    'Men': '#ff7f0e',
    'Women': '#ffb78',
    'No college': '#2ca02c',
    'College': '#98df8a',
    'under 45': '#d62728',
    'over 45': '#ff9896',
    'Biden voter': '#0015BC',
    'Trump voter': '#DC143C'
}
pairs = [
    ('White', 'Non-white'),
    ('Men', 'Women'),
    ('No college', 'College'),
    ('under 45', 'over 45'),
    ('Biden voter', 'Trump voter')
]
```

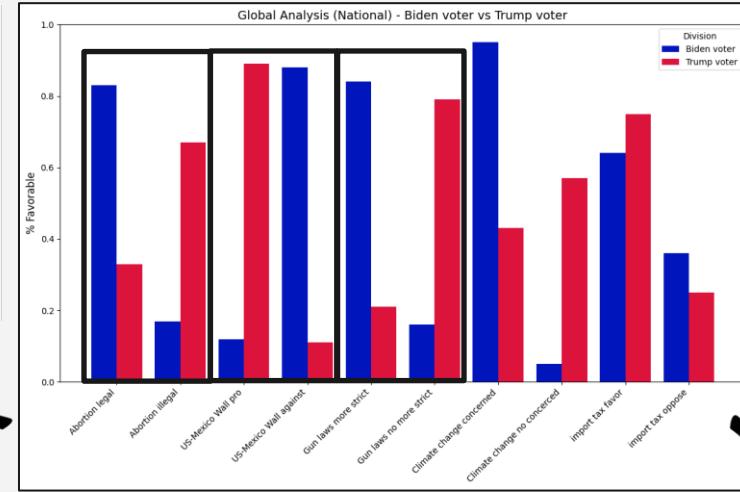
Data preparation and definitions of some frequently used function.



Social main topics- National analysis

```
national_data = df[df['State'] == 'National']

def global_analysis_by_pairs():
    for pair in pairs:
        pair_data = national_data[national_data['Division'].isin(pair)]
        melted_data = pair_data.melt(id_vars=['State', 'Division'],
                                     var_name='Theme',
                                     value_name='Percentage')
        themes = melted_data['Theme'].unique()
        colors = pair_colors.get(pair)
        generic_analysis(melted_data, themes,
                          f'Global Analysis (National) - {pair[0]} vs {pair[1]}',
                          pair, colors, 'Division')
global_analysis_by_pairs()
```

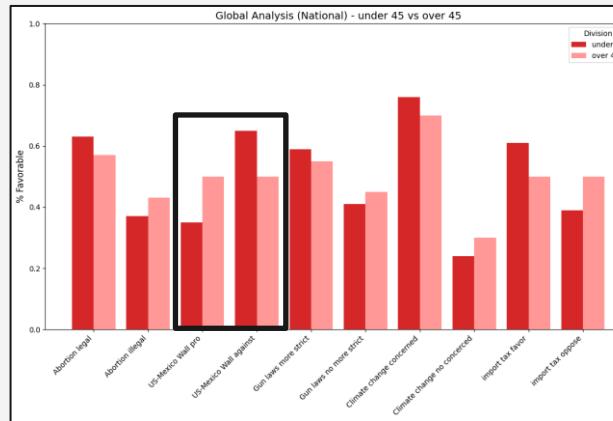
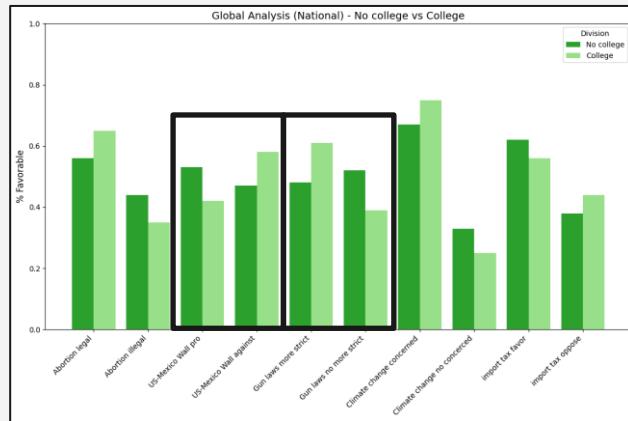
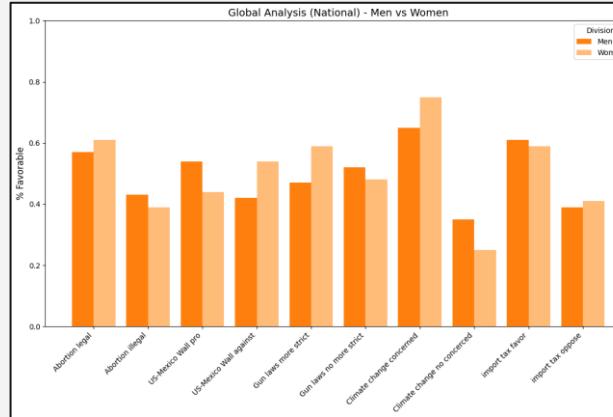
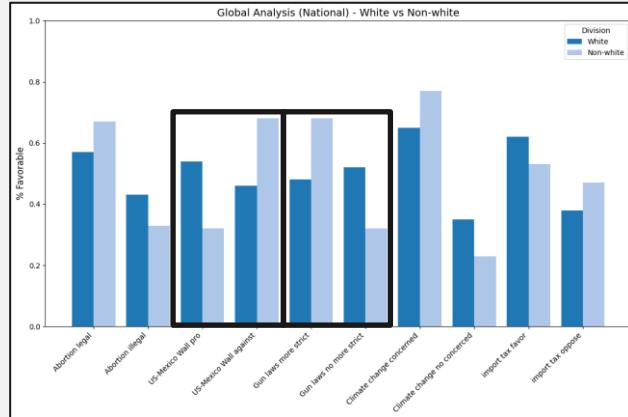


At the national level, it immediately becomes clear how supporters of various parties are extremely divided on different issues. Specifically, this applies to topics such as abortion, the Mexican border wall and gun restrictions.

Due to concerns about climate change, we notice that those who voted for Biden are extremely worried, whereas Trump voters are split 50 – 50 on the issue.



Social main topics- National analysis



No significant differences between men and women. Women seem slightly more inclined toward Democratic positions.



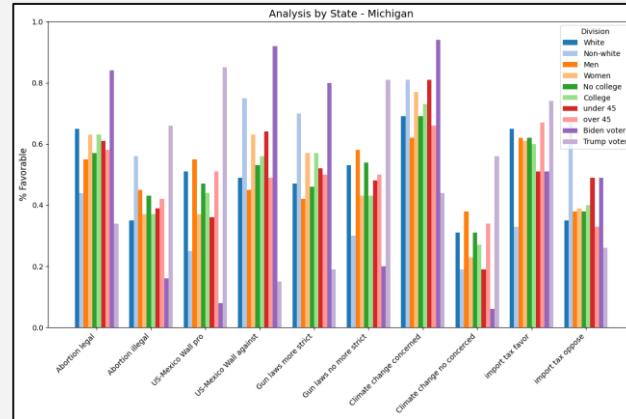
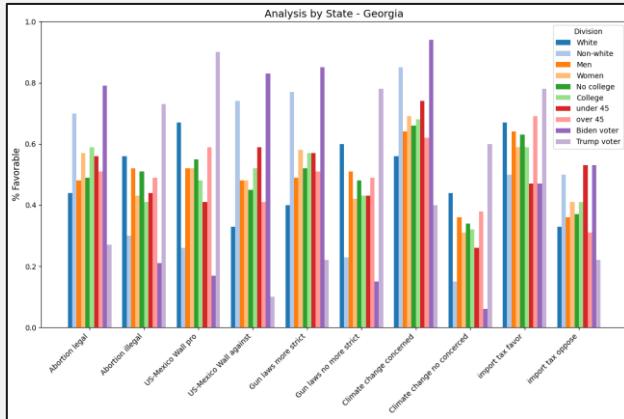
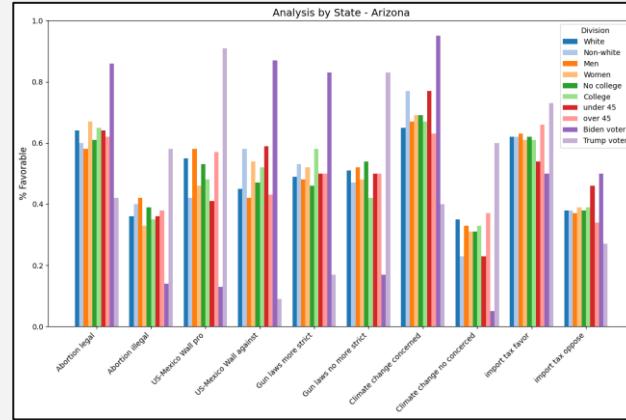
Social main topics- Analysis by state

```

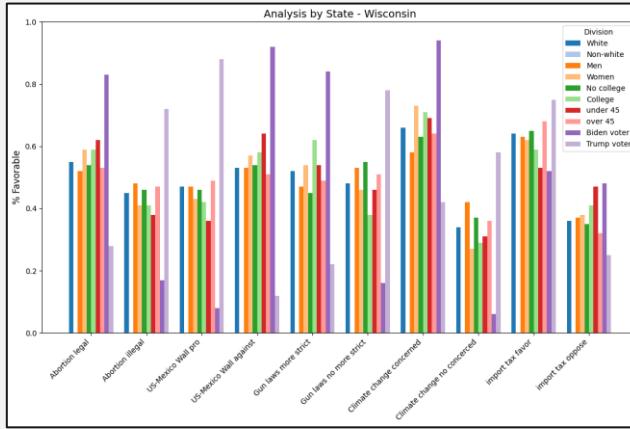
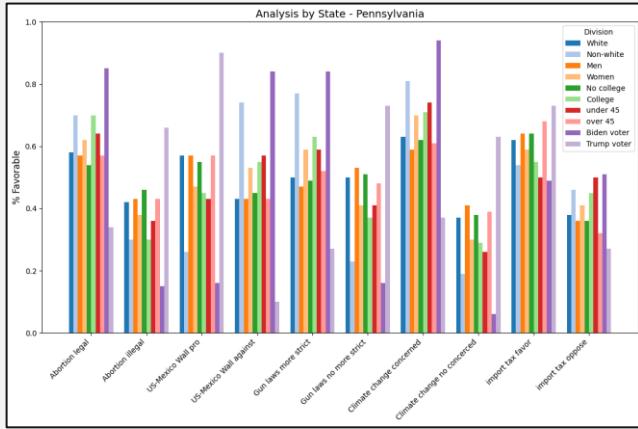
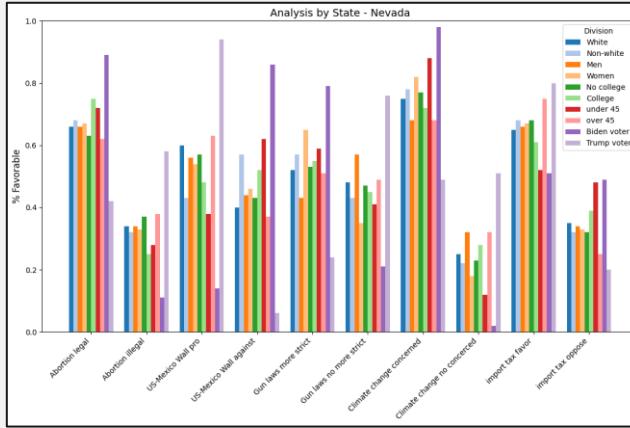
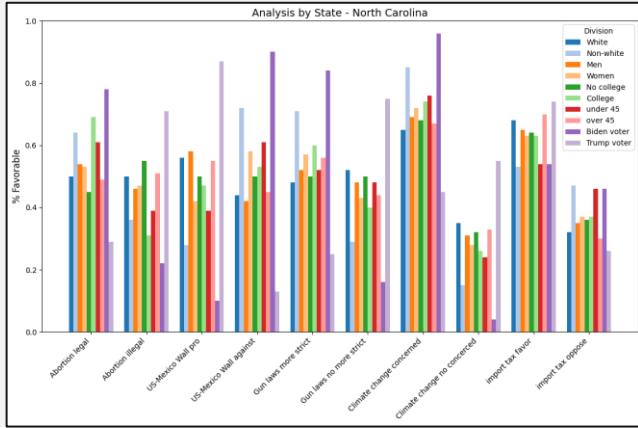
def state_analysis():
    states = df['State'].unique()
    for state in states:
        if state != 'National':
            state_data = df[df['State'] == state]
            melted_data = state_data.melt(id_vars=['State', 'Division'],
                                           var_name='Theme',
                                           value_name='Percentage')
    themes = melted_data['Theme'].unique()
    divisions = melted_data['Division'].unique()
    colors = plt.cm.tab20.colors[:len(divisions)]
    generic_analysis(melted_data, themes,
                      f'Analysis by State - {state}',
                      divisions, colors, 'Division')

state_analysis()

```



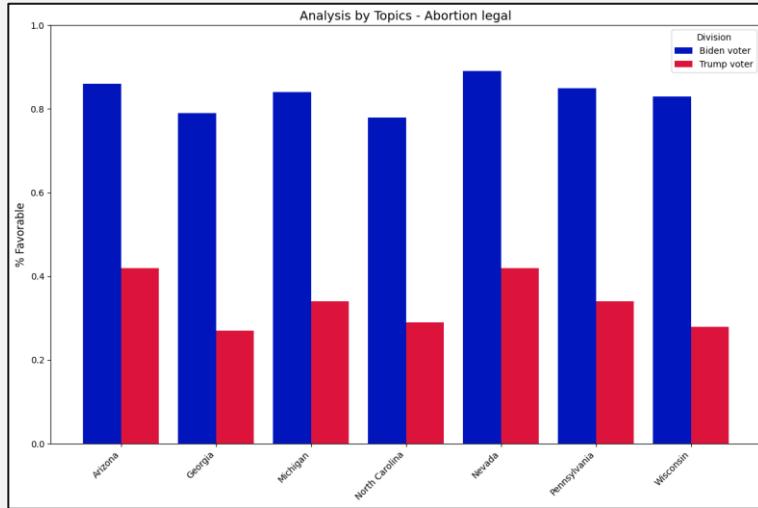
Social main topics- Analysis by state



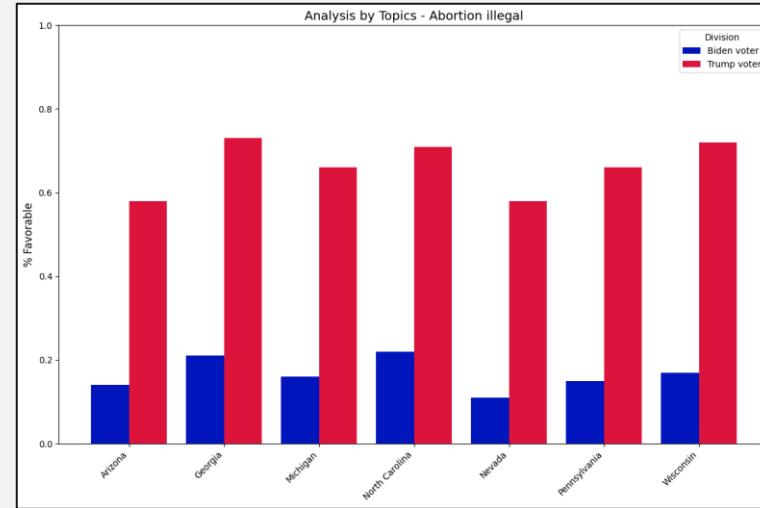
Analysis by topics - Abortion

```
df_without_national = df[df['State'] != 'National']
swing_states_list=list(df_without_national['State'].unique())
def topics_analysis():
    topics = [col for col in df.columns if col not in ['State', 'Division']]
    for topic in topics:
        for pair in pairs:
            pair_data = df[(df['Division'].isin(pair)) &
            (df['State'] != 'National')].copy()
            pair_data.rename(columns={topic: 'Percentage'}, inplace=True)
            colors = pair_colors.get(pair)
            generic_analysis(pair_data, swing_states_list,
                f'Analysis by Topics - {topic}',
                pair, colors, 'Division')
topics_analysis()
```

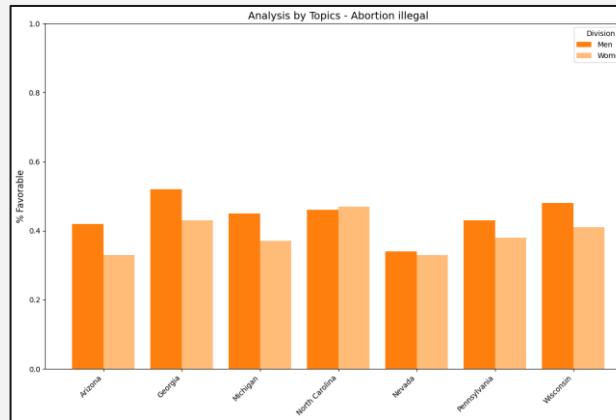
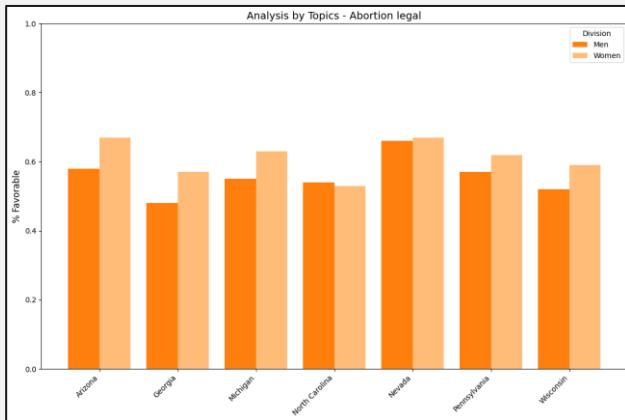
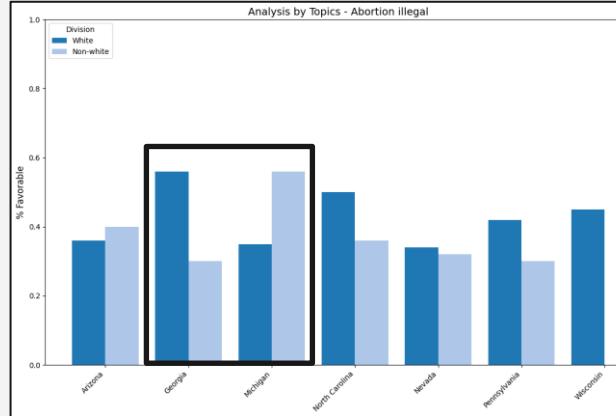
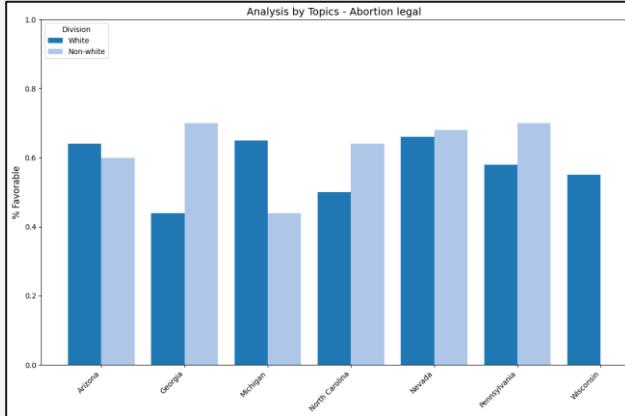
Differences between swing states.



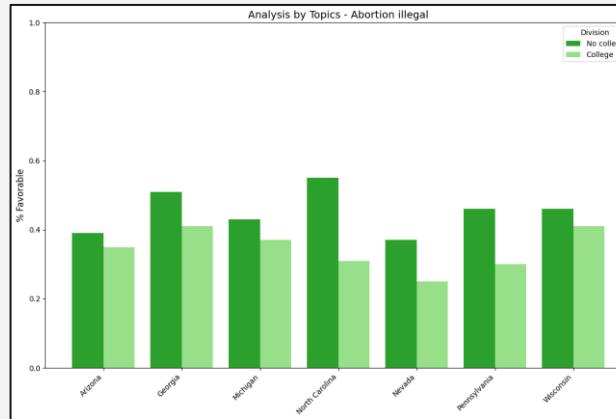
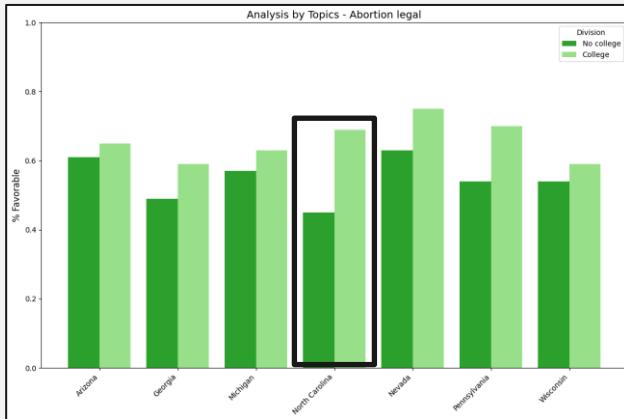
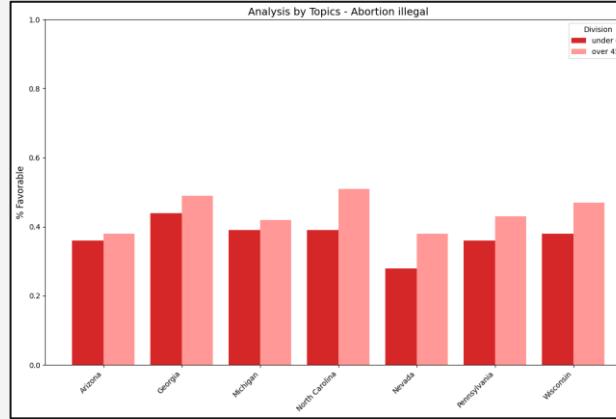
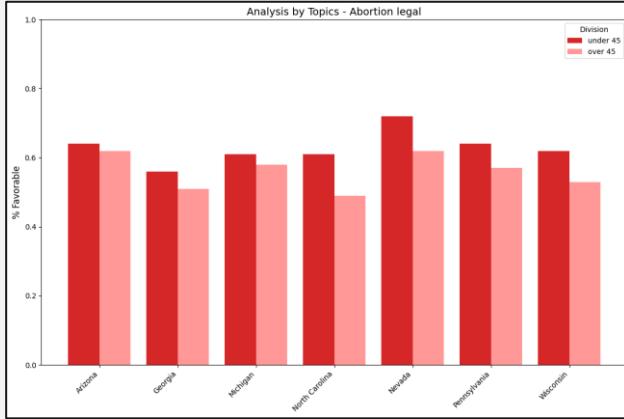
We can observe a significant divergence on the topic of abortion across all swing states.



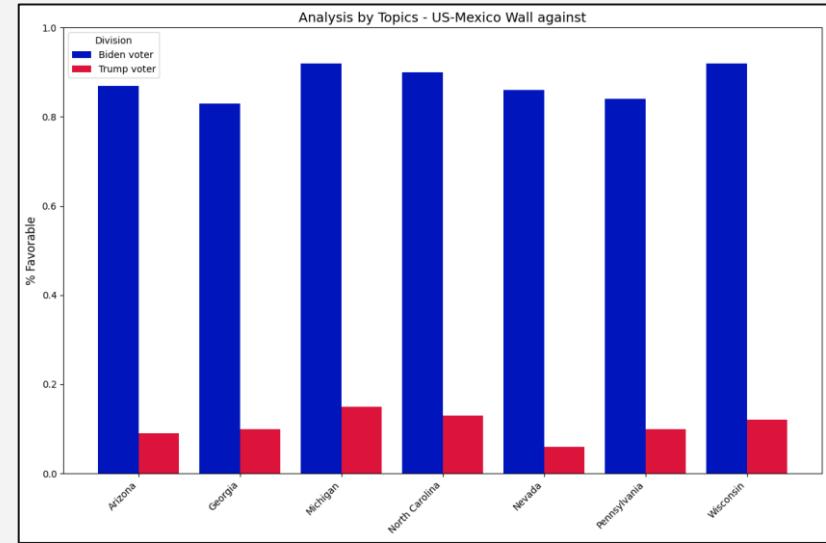
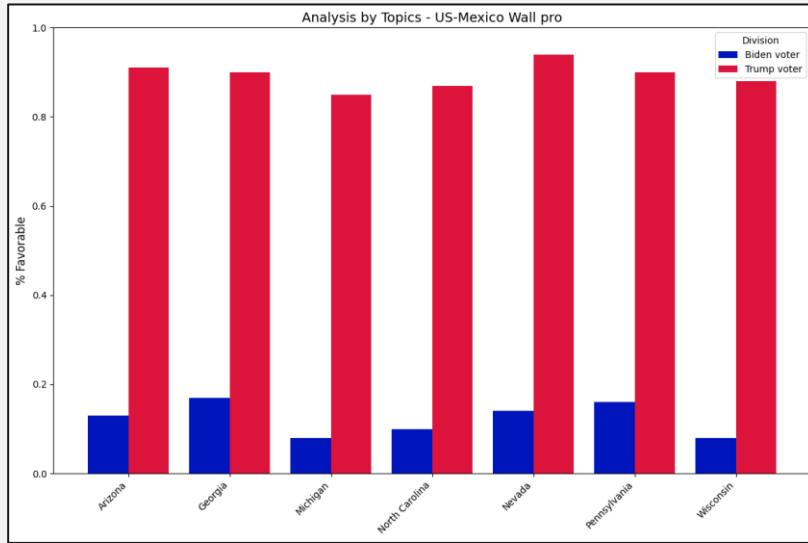
Analysis by topics - Abortion



Analysis by topics - Abortion



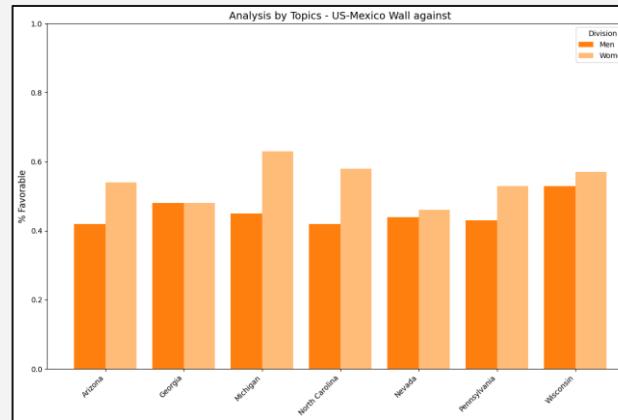
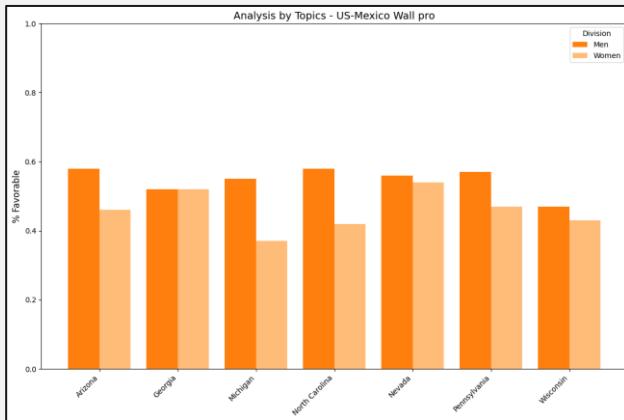
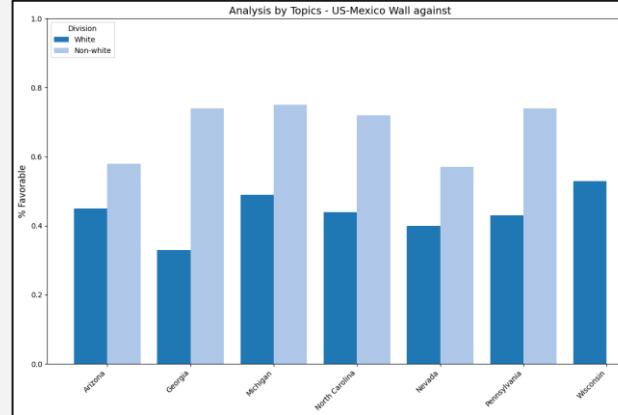
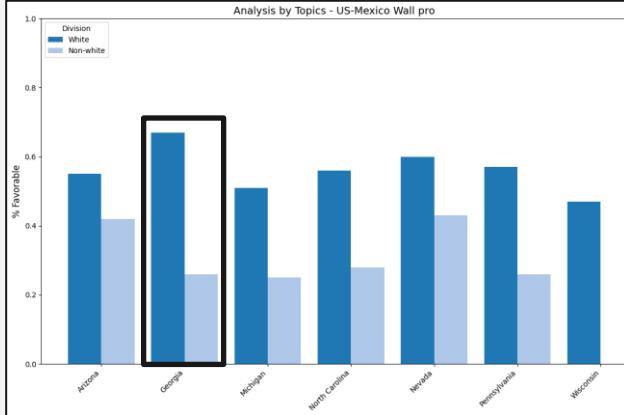
Analysis by topics – Mexico Wall



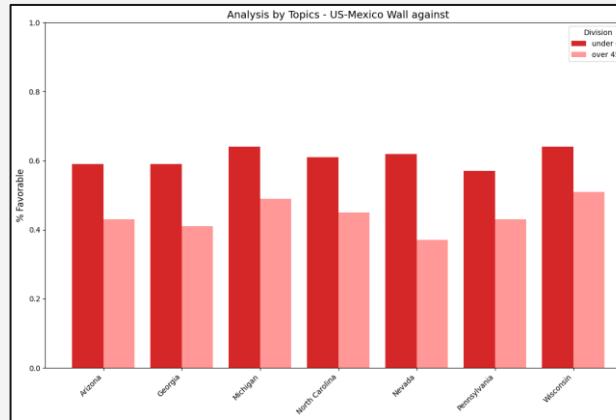
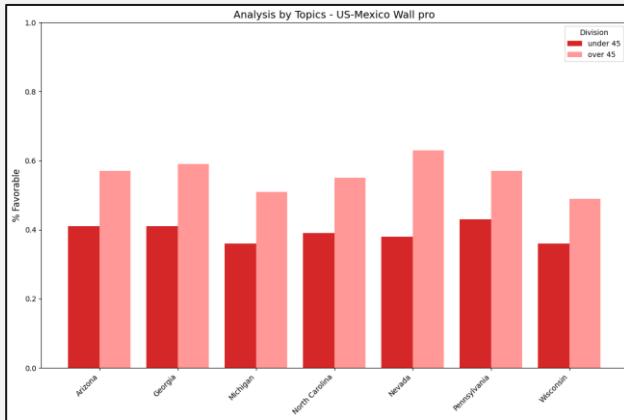
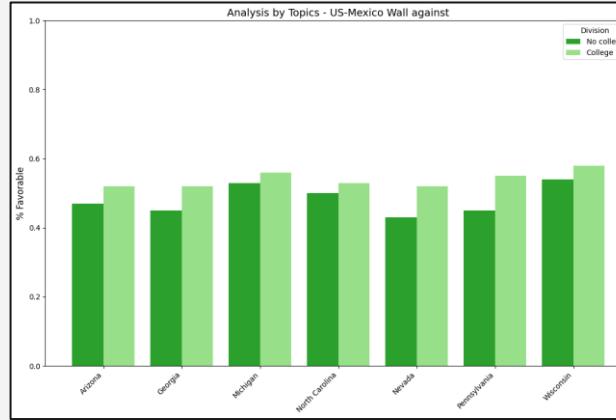
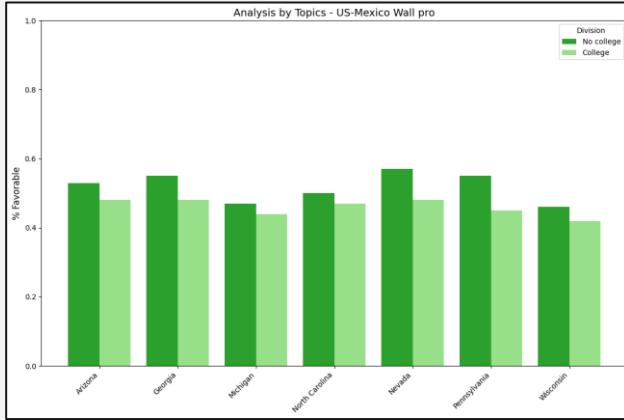
We can observe a significant divergence on the topic of the Mexico Wall across all swing states.



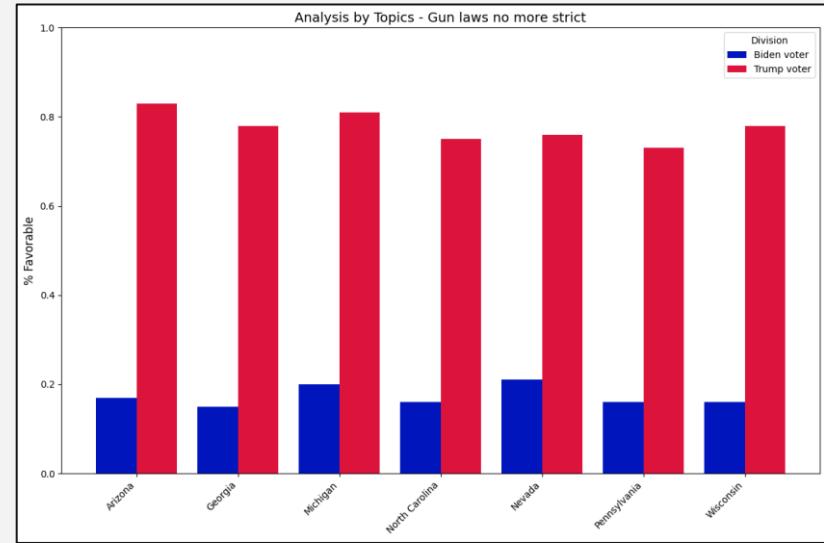
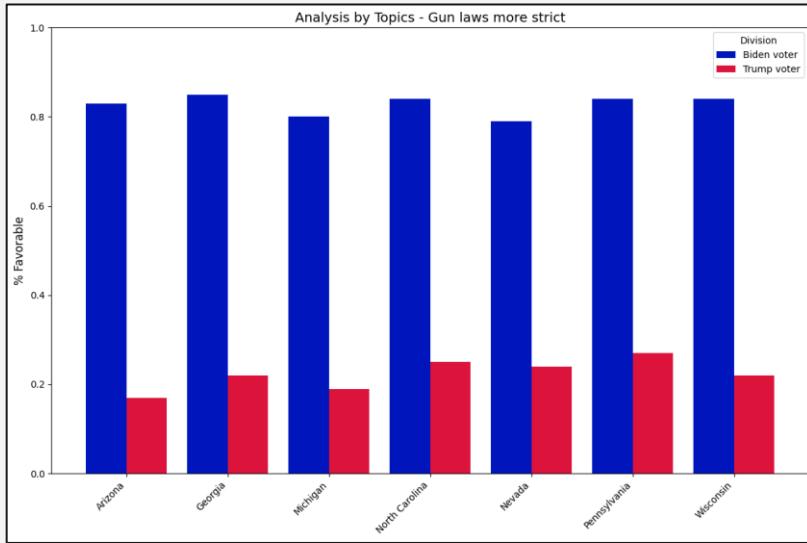
Analysis by topics – Mexico Wall



Analysis by topics - Mexico Wall



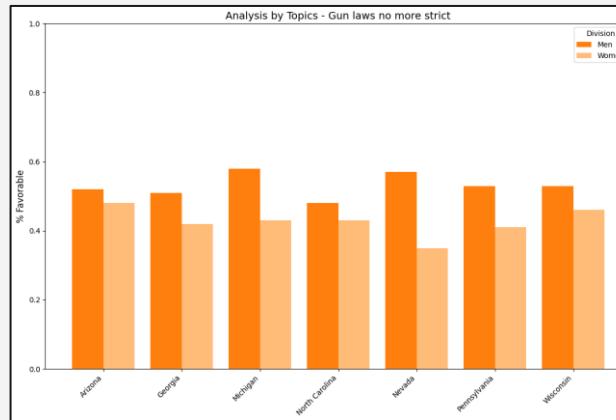
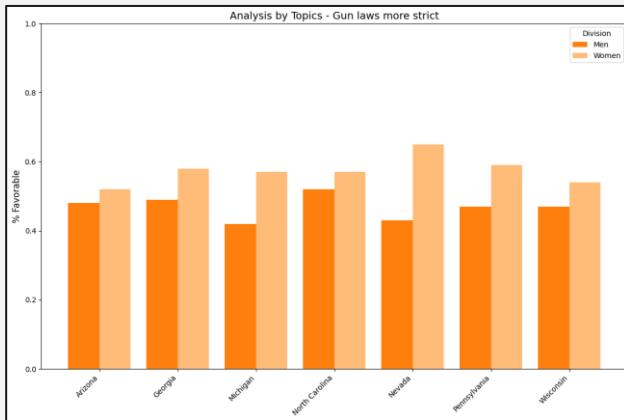
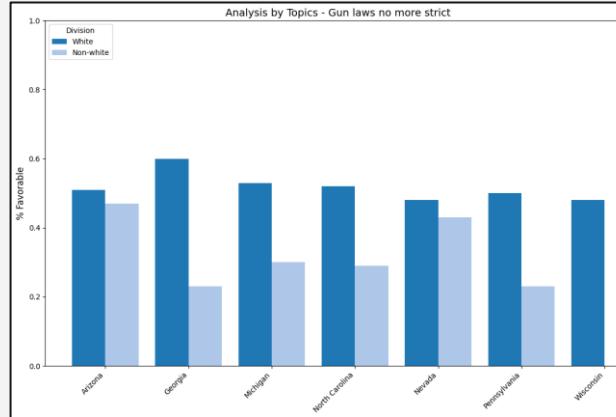
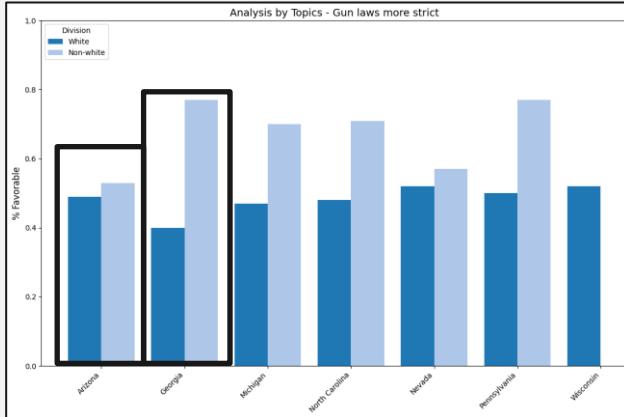
Analysis by topics – Gun Laws



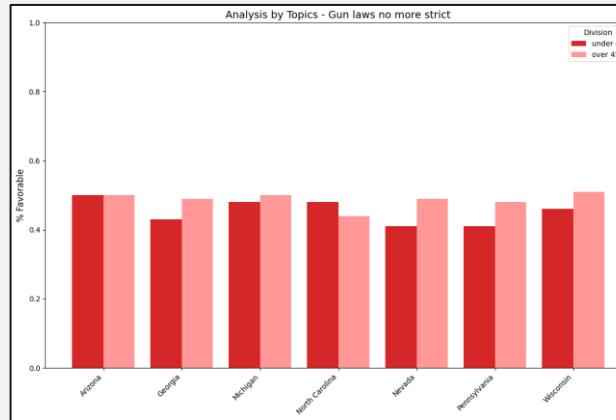
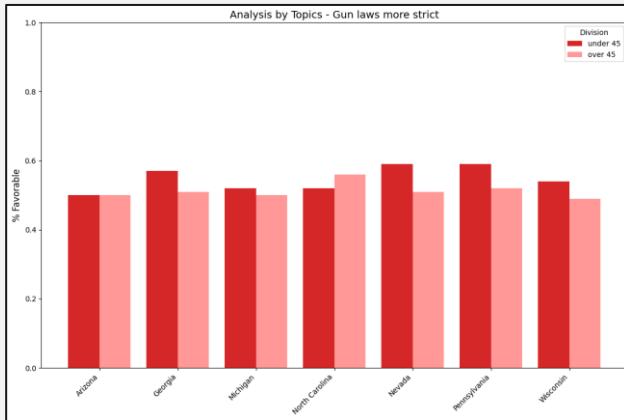
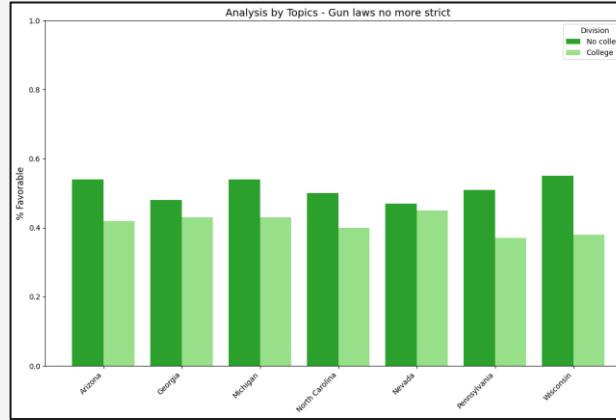
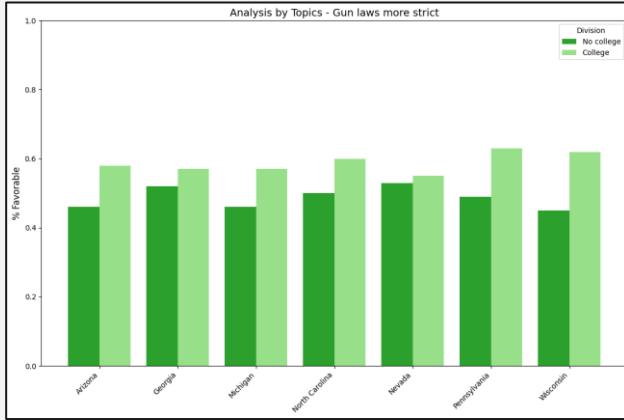
Trump voters believe that gun laws should not be more restrictive, while Biden voters hold the opposite view.



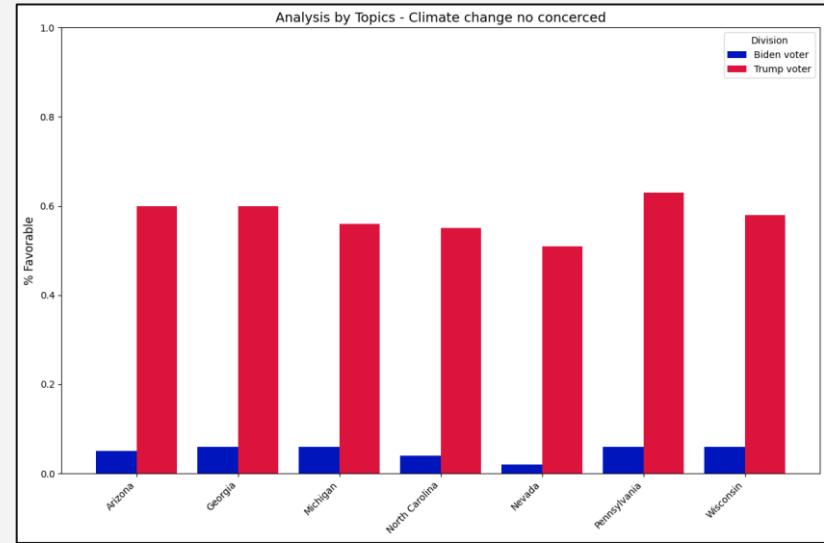
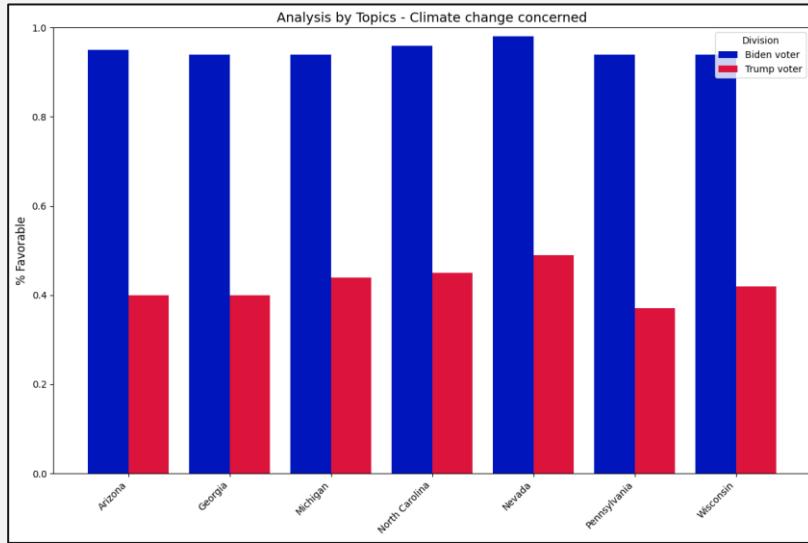
Analysis by topics – Gun Laws



Analysis by topics – Gun Laws



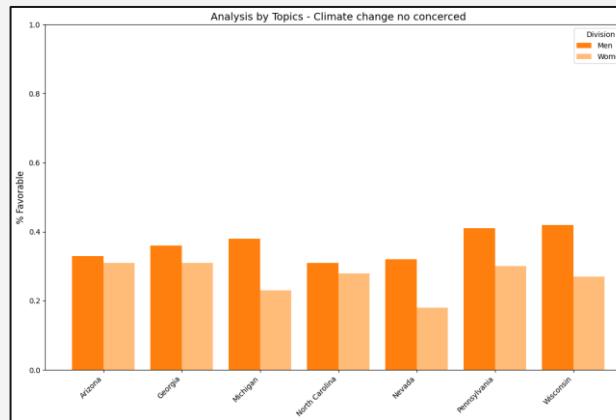
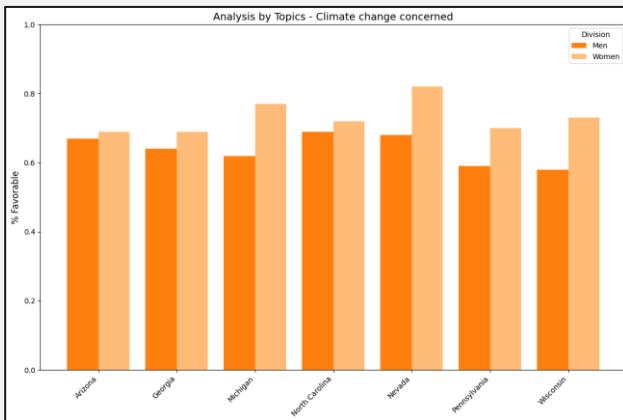
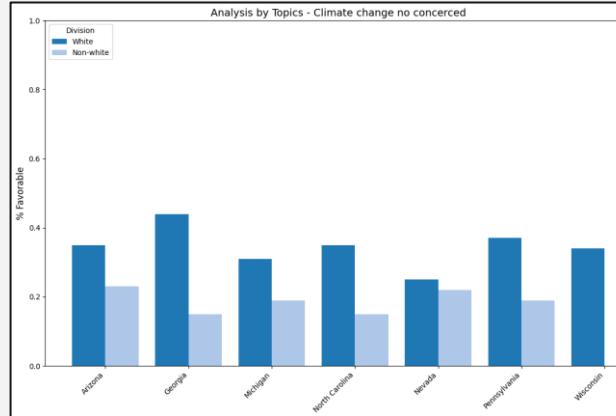
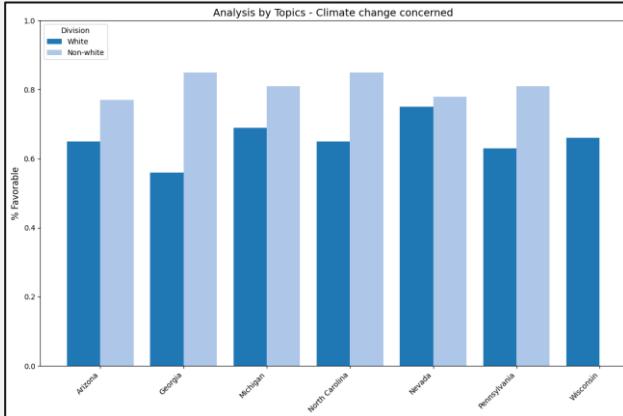
Analysis by topics – Climate Change



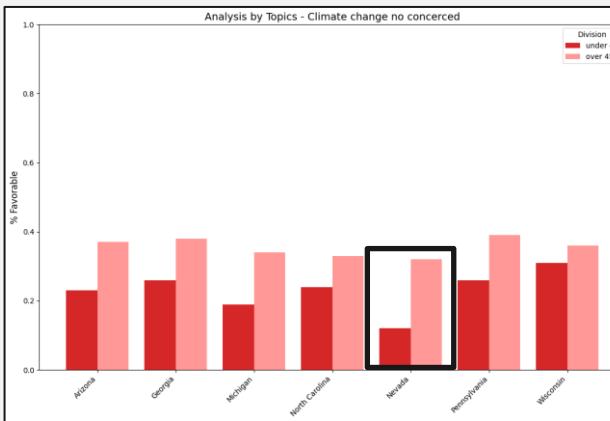
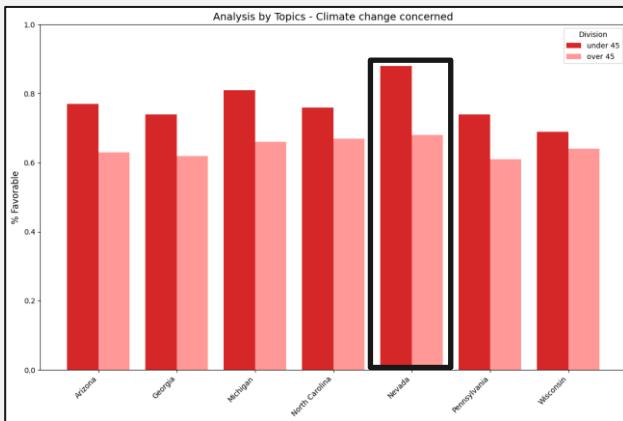
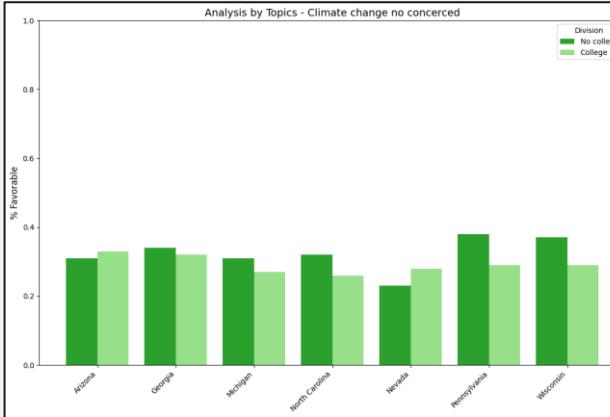
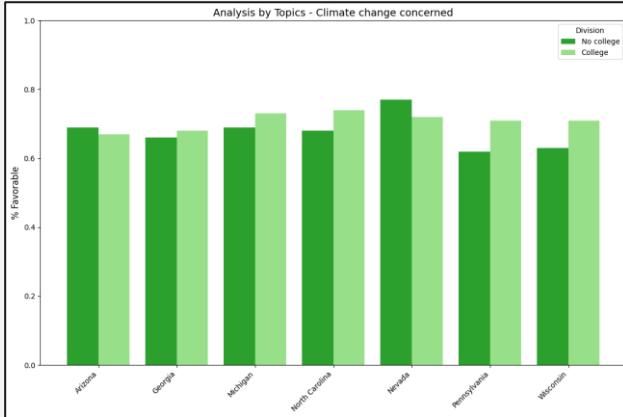
Almost all Biden voters are extremely concerned about climate change. Republicans are more divided on the issue.



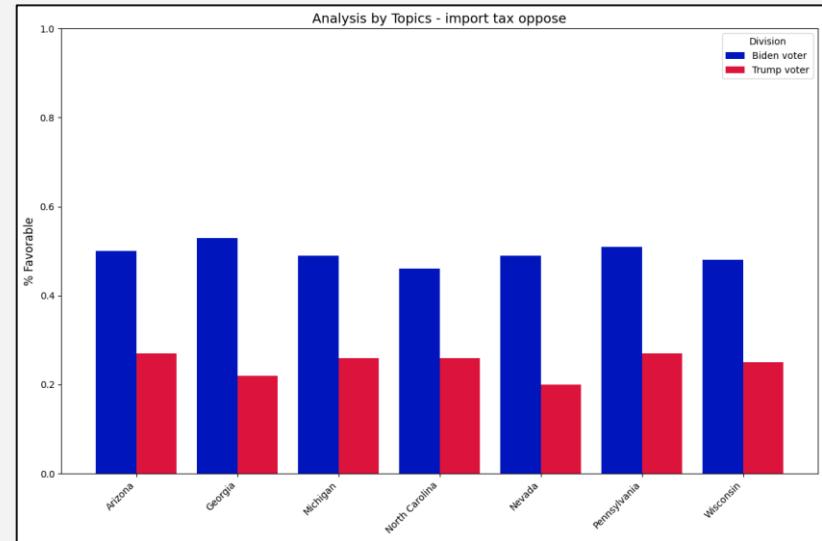
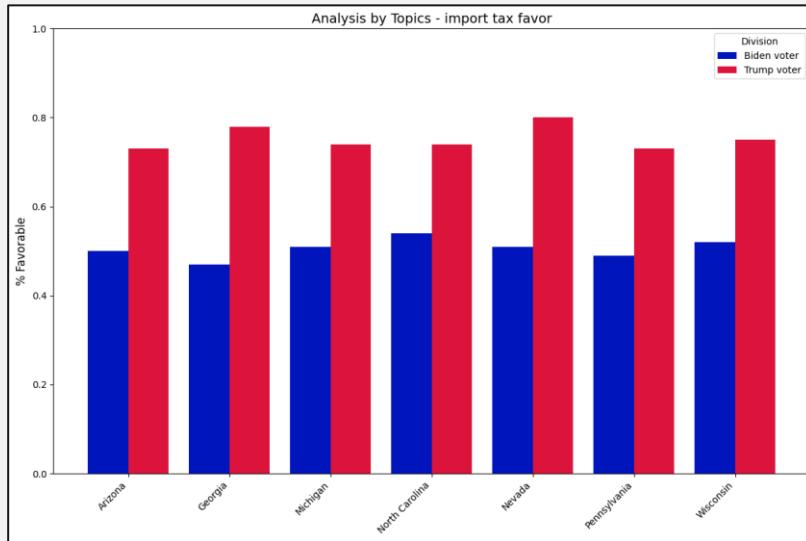
Analysis by topics – Climate Change



Analysis by topics – Climate Change



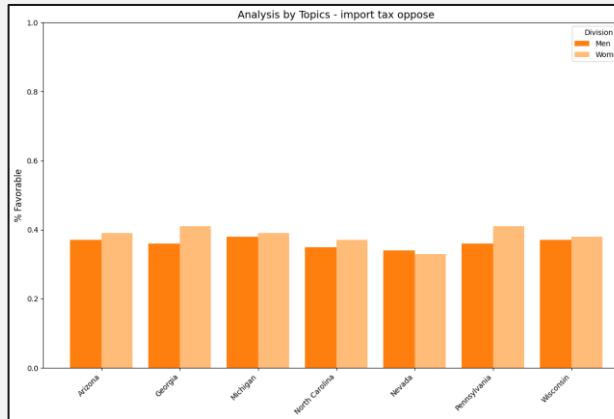
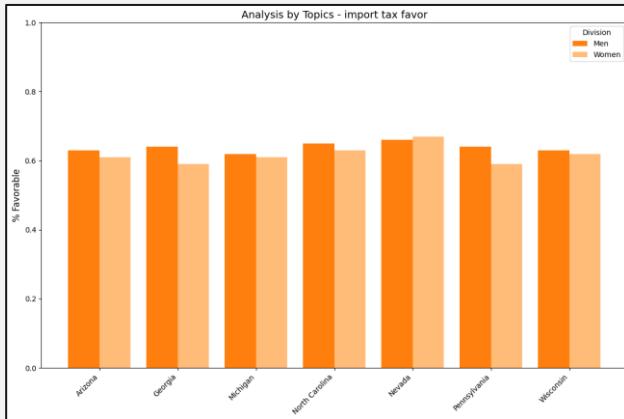
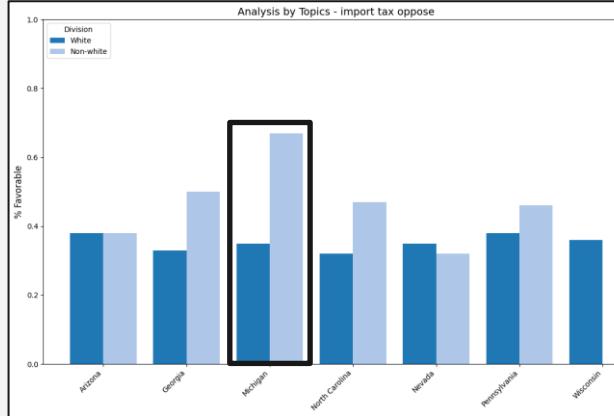
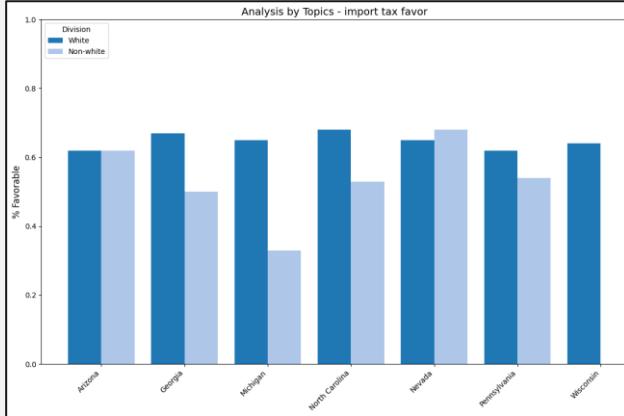
Analysis by topics – Import Tax



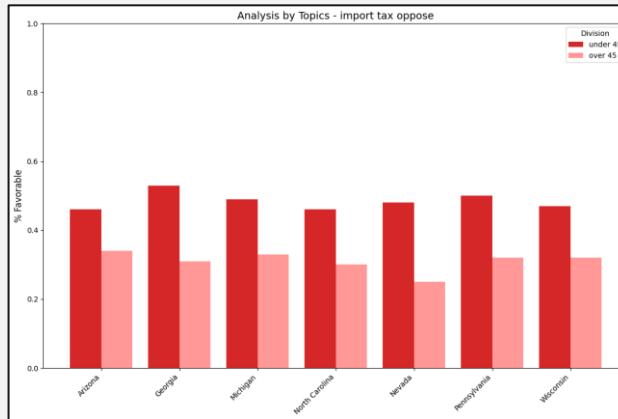
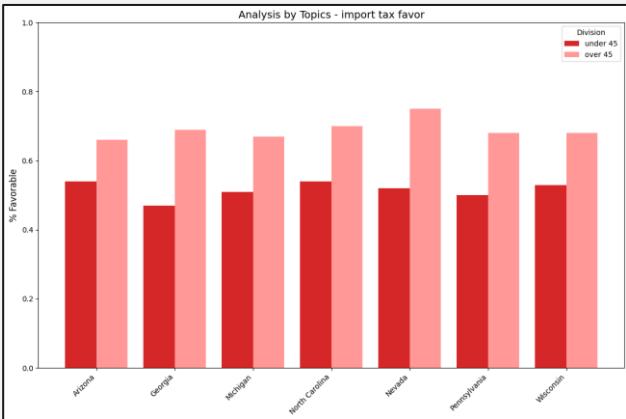
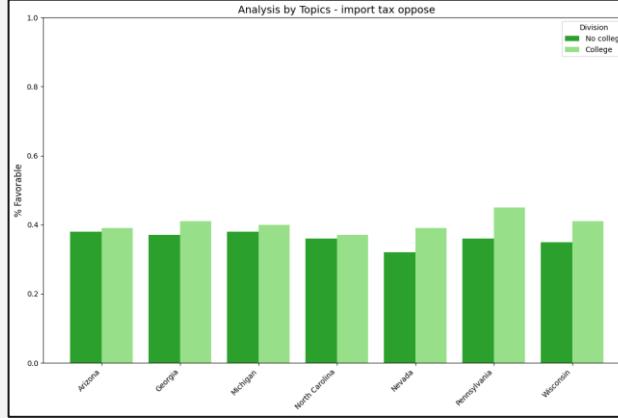
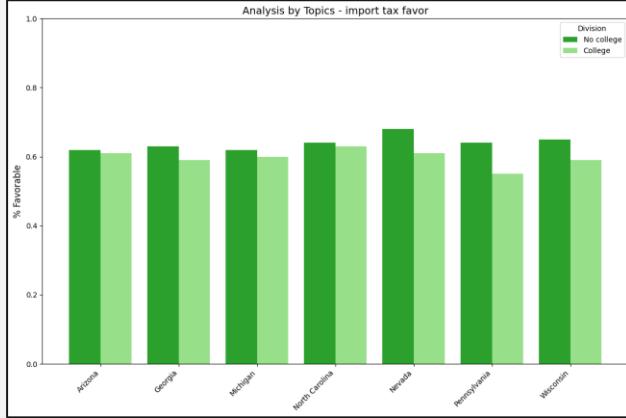
On the topic of tariffs, Democrats are evenly split, whereas Republicans show greater consensus in favor of tariffs.



Analysis by topics - Import Tax



Analysis by topics - Import Tax



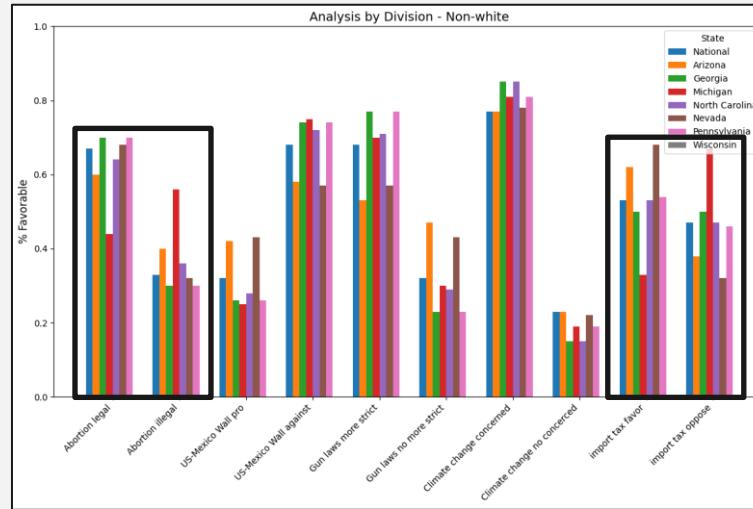
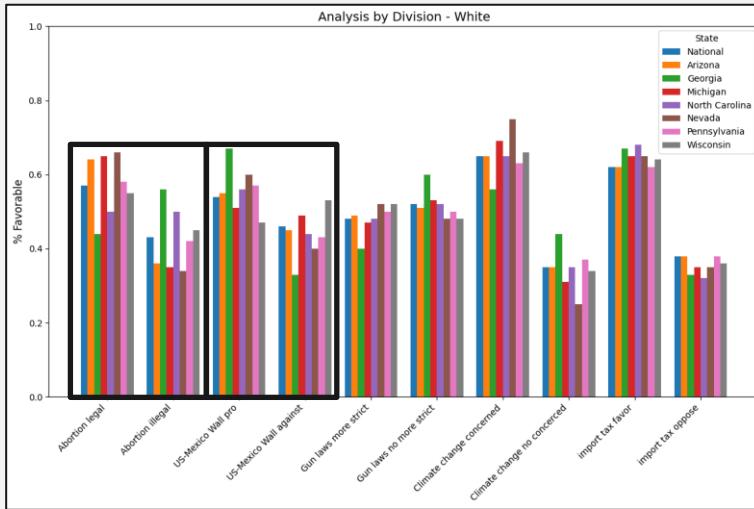
Social main topics- Analysis by population segment

```
def division_analysis():
    divisions = df['Division'].unique()
    for division in divisions:
        division_data = df[df['Division'] == division]
        melted_data = division_data.melt(id_vars=['State', 'Division'],
                                         var_name='Theme',
                                         value_name='Percentage')
    themes = melted_data['Theme'].unique()
    states = melted_data['State'].unique()
    colors = plt.cm.tab20.colors[:len(states)]
    generic_analysis(melted_data, themes,
                      f'Analysis by Division - {division}',
                      states, None, 'State')

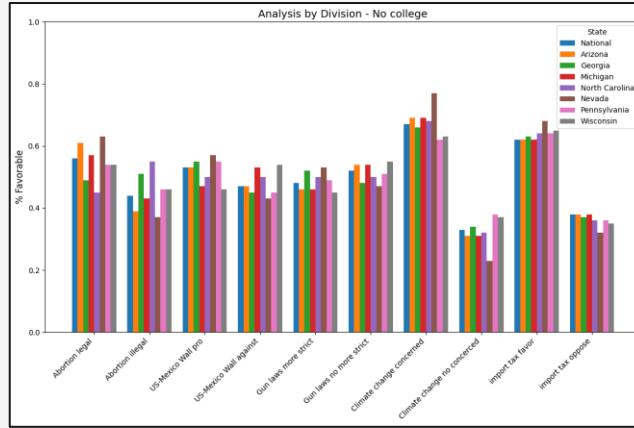
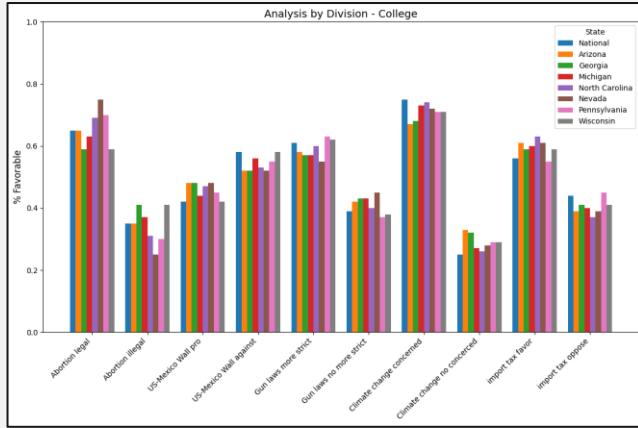
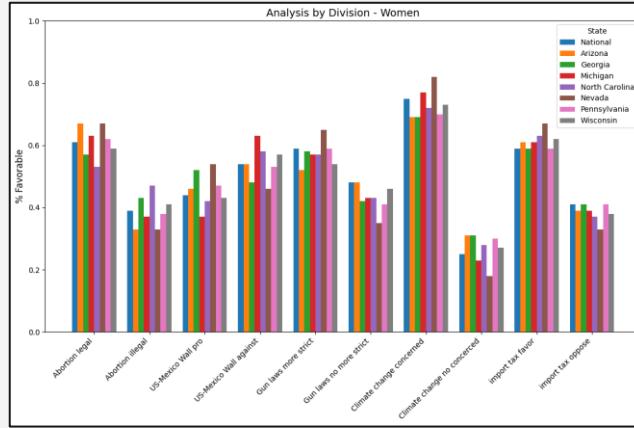
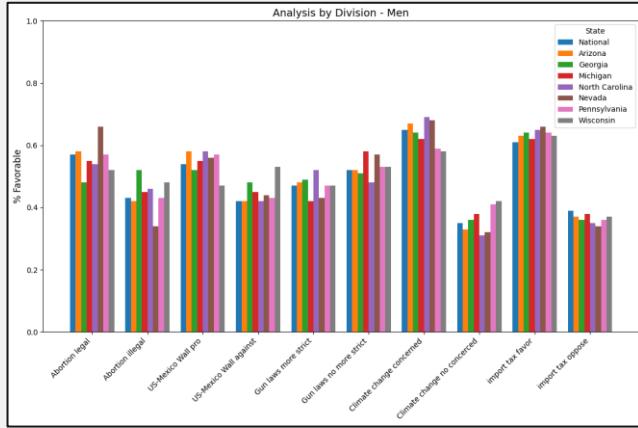
division_analysis()
```



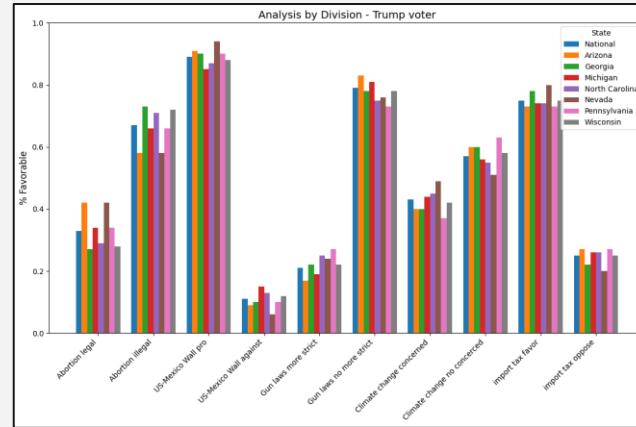
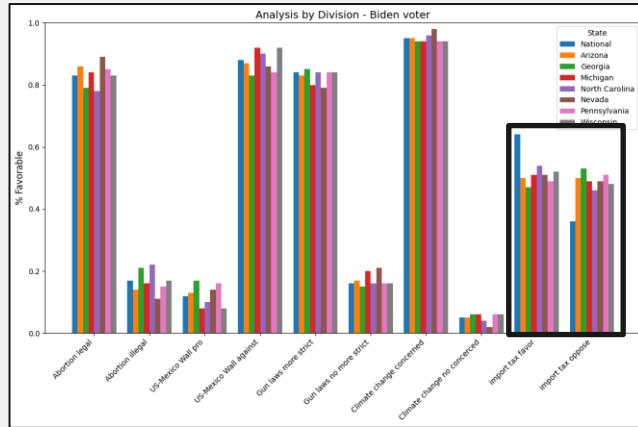
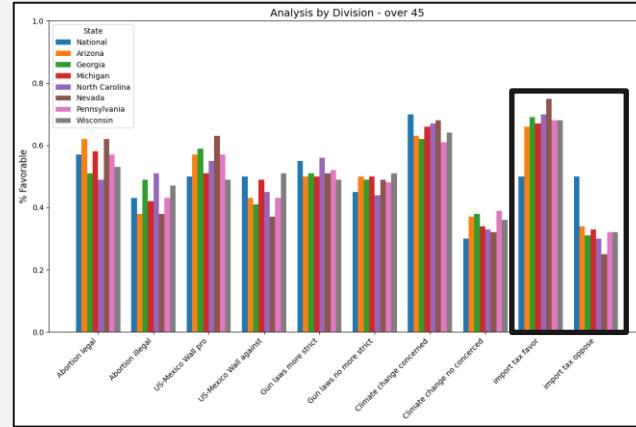
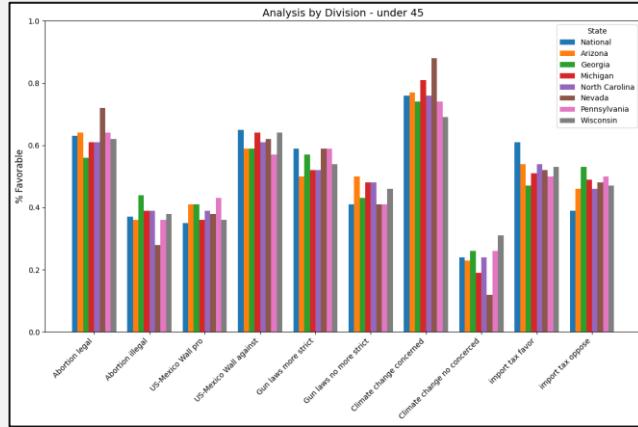
Differences between each topic.



Social main topics- Analysis by topics



Social main topics- Analysis by topics



++
++
++



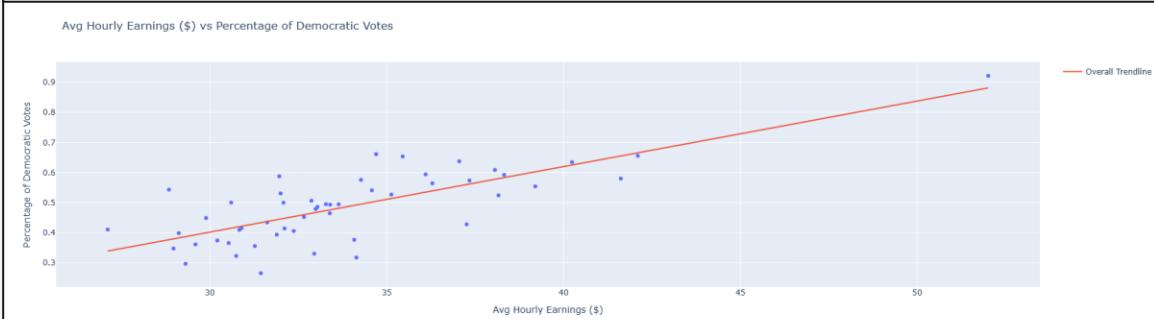
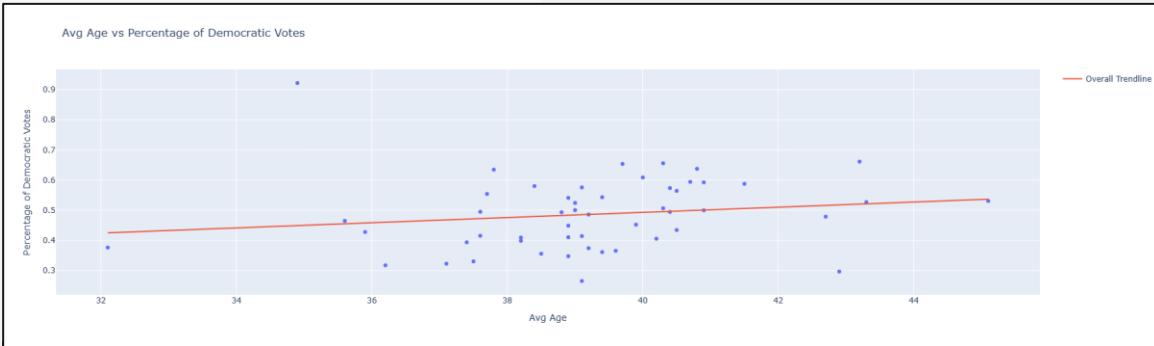
Linear regression

Linear regression

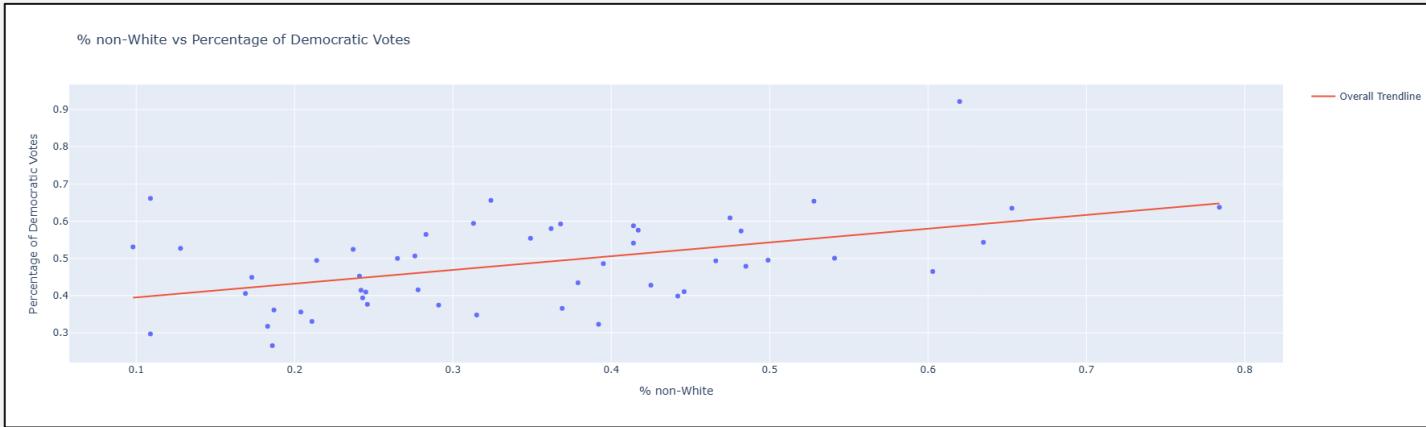
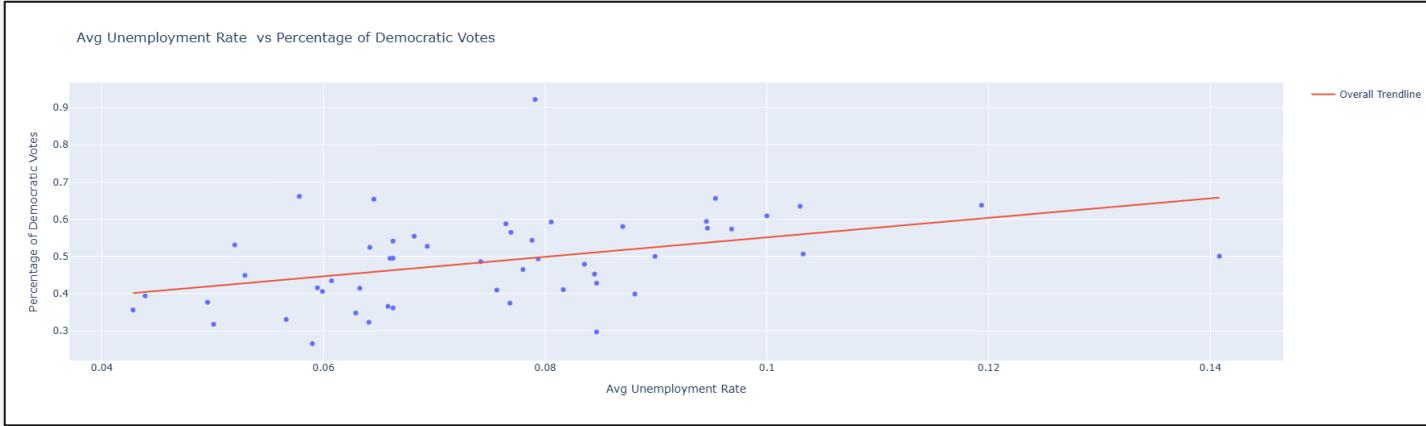
```
lst = [col for col in df.columns if col not in [df.columns[0], df.columns[1],  
                                                df.columns[2], df.columns[3]]]  
  
for i in lst:  
    fig = px.scatter(  
        x=df[i],  
        y=df['% DEM 2020'],  
        labels={'x': i, 'y': 'Percentage of Democratic Votes'},  
        title=f'{i} vs Percentage of Democratic Votes',  
        trendline = 'ols', trendline_scope = 'overall')  
  
fig.show()
```



The code generates scatterplots to visualize the relationship between different independent variables and the percentage of Democratic votes (% DEM 2020).



Linear regression



Hypothesis test

```
y = df['% DEM 2020']

X = df[['Dem Historical Trend', 'Avg Age', 'Avg Hourly Earnings ($)',
        'Avg Unemployment Rate ', '% non-White']]
X = sm.add_constant(X)

model = sm.OLS(y, X).fit()
print(model.summary())
```

```
=====
OLS Regression Results
=====
Dep. Variable: % DEM 2020 R-squared:      0.832
Model: OLS            Adj. R-squared:   0.813
Method: Least Squares F-statistic:    44.56
Date: Tue, 21 Jan 2025 Prob (F-statistic): 2.50e-16
Time: 10:42:43          Log-Likelihood: 81.560
No. Observations: 51 AIC:             -151.1
Df Residuals:     45 BIC:             -139.5
Df Model:         5
Covariance Type: nonrobust
=====

            coef    std err       t      P>|t|      [0.025      0.975]
-----
const      -0.6785    0.217    -3.131     0.003     -1.115     -0.242
Dem Historical Trend  0.0732    0.023     3.221     0.002     0.027     0.119
Avg Age      0.0140    0.004     3.140     0.003     0.005     0.023
Avg Hourly Earnings ($)  0.0157    0.002     6.501     0.000     0.011     0.021
Avg Unemployment Rate  -0.3661   0.522    -0.701     0.487     -1.418     0.686
% non-White   0.2397    0.067     3.580     0.001     0.105     0.375
-----
Omnibus:           0.250 Durbin-Watson:    1.935
Prob(Omnibus):    0.883 Jarque-Bera (JB):  0.312
Skew:              0.155 Prob(JB):       0.856
Kurtosis:          2.774 Cond. No.    3.74e+03
-----
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.74e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

The values of the p-values indicate that the variables are statistically significant, except for the *unemployment rate*, which has a p-value of **48.7%**, leading to the conclusion that the null hypothesis of no influence on the dependent variable is not rejected. Other useful statistical variables for interpreting the model include the R-squared (0.832), which indicates that **83.2%** of the variability in the dependent variable is explained by the model, and the F-statistic, along with its p-value, which indicates that the model is statistically significant.





Conclusions

Summary of topics analyzed

① Economic Topic

Analyzing the economic variables we could observe some differences following the elections. For employment, there was an increase, which benefited all ethnic groups. The unemployment rate, despite the strong increase in 2020 due to COVID-19, in the face of Biden's presidency there was a notable decrease during the months of 2021. For the weekly earnings, only Asians experienced modest changes, while each state increased its average weekly wage over the following two years. The consumer price index steadily increased, reaching its maximum peak in July 2022 with 9.1%, and then decreased over the following months.

③ Social Topic

In the analysis of the main issues addressed in the 2020 elections, it merged that Biden and Trump voters are extremely divided in the various segments. Analyzing, instead, the various segmentations of population, it emerged that the difference was made by *White-Non White* segmentation, despite the others analyzed in the graphs.

② Geographic topic

In 2020, voter turnout in the United States was around 67% of the population, and of this around 70% belong to the *White* category. In the seven swing states identified, the white population generally tends to favor the Republican Party, while other ethnic entities, such as blacks and hispanics, are more tied to the Democratic Party. These data suggest an important electoral dynamic in which the ethnic composition of the population plays a crucial role in political preferences, particularly in key states such as those defined as swing states.

④ Historical topic

The voting history of a particular state significantly influences the percentages of votes obtained by the parties. This is easily inferred not only from the distribution of the boxplots but also from the univariate regression analysis, which highlights a strong statistical significance that is maintained even in the multivariate regression.

++
++
+

Thanks for your attention!

Angeletti Giacomo

Bertolini Tobia

Data Analytics for Business and Society
Computer Programming and Data Management
print("group name")

Cattelan Lorenzo

Zoccarato Mattia