Email: derek@cassadyweb.com Name: Derek Cassady Program: Data Analysis Class: Indiana Cohort

BLS Data Visualization Project

Project Objective:

Pick one BLS (Bureau of Labor Statistics) industry and present the fatalities for a given time period.

The purpose in this instance is not to find any new insight into the fatalities. The primary purpose is to the expand knowledge and comfort with libraries used in first project, as well as, introducing new libraries. Addition of error checking through IF/ELSE statements and iteration with FOR loop to build on Python basic skills in a work like environment.

Step 1: Determine data location, retrieval method, data type. BLS piblic API: https://api.bls.gov/publicAPI/v2/timeseries/data/ API Sample Code: https://www.bls.gov/developers/api_signature_v2.htm#multiple Data formatting help: https://www.bls.gov/help/hlpforma.htm Step 2: Review JSON data load from API {'status': 'REQUEST_SUCCEEDED', responseTime': 1859. JSON Data 'message': [], 'Results': {'series': [{'seriesID': 'FWU00X00000081N00', 'data': [LIST] Step 3: Parse JSON Data into individual series reports corresponding to BLS series. Series 0-11 Series_0 = json_data['Results']['series'][0 - 11] Step 4: Build data frame for each series of BLS series. Provide for empty data frame, and other origin data errors. Error Handling with IF/ELSE df_0 = pd.DataFrame(Series_0['data']) var_0 = 'Total' if df_0.empty == True: df_0 = pd.DataFrame(y_list, columns=['year']) $df_0 = df_0.assign(blank = 0).astype(str)$ IF statement to build data frame IF the origin data creates an empty data frame Statement df_0 = df_0.rename(columns={'blank':var_0}) DataFrame df_0 = df_0.rename(columns={'year':'Year'}) print(df_0) else: df_0 = df_0.drop(['period','periodName','latest', 'footnotes'], axis = 1) for i in dict(dict_1): $df_0.loc[len(df_0)] = i,0$ df_0 = df_0.drop_duplicates(subset='year', keep='first') ELSE ELSE statement to account for correct data frame creation from origin df_0.sort_values("year", inplace=True) Statement df_0['value'] = df_0['value'].astype(float) $df_0 = df_0.rename(columns={'value':var_0})$ df_0 = df_0.rename(columns={'year':'Year'}) print(df_0) Step 5: Create Data Frame of Data Frames for use in grouped bar charting. df_all =df_0.merge(df_1,on='Year') df_all =df_all.merge(df_2,on='Year') df_all =df_all.merge(df_3,on='Year') df_all =df_all.merge(df_4,on='Year') df_all =df_all.merge(df_5,on='Year') DataFrame df_all =df_all.merge(df_6,on='Year') df_all =df_all.merge(df_7,on='Year') $df_all = df_all.merge(df_8,on='Year')$ df_all =df_all.merge(df_9,on='Year') df_all =df_all.merge(df_10,on='Year') df_all =df_all.merge(df_11,on='Year') Step 6: Write to excel for all data frames, separate sheet per data frame with pd.ExcelWriter("Warehouse_BLS.xlsx") as writer: df all.to excel(writer, sheet name="df all", index=False) df_0.to_excel(writer, sheet_name= var_0, index=False) df_1.to_excel(writer, sheet_name= var_1, index=False) df_2.to_excel(writer, sheet_name= var_2, index=False) df 3.to excel(writer, sheet name= var 3, index=False) df 4.to excel(writer, sheet name= var 4, index=False) Write to Excel df_5.to_excel(writer, sheet_name= var_5, index=False) df_6.to_excel(writer, sheet_name= var_6, index=False) df_7.to_excel(writer, sheet_name= var_7, index=False) df 8.to excel(writer, sheet name= var 8, index=False) df_9.to_excel(writer, sheet_name= var_9, index=False) df_10.to_excel(writer, sheet_name= var_10, index=False) df_11.to_excel(writer, sheet_name= var_11, index=False) Step 6: Create Figure from Plotly Express for each data frame $fig_0 = px.bar(df_0, x=df_0['Year'], y=df_0[var_0],$ text_auto=True, title= 'US Workplace ' + var_0 + '
' + ' Fatalities
' + str(syear) + ' - ' + str(eyear), "Year": "Year", "Total": "Number of Fatalities", "variable": "Group" fig_0.update_yaxes(autorange=False,fixedrange=False,range=['4000','5000']) fig_0.update_traces(marker_color='#111173') Figure fig_0.update_layout(0-11 font_family="Arial", font_color="black", title_x=0.5, title_y=0.92 title_xanchor="center", title_font_family="Arial", title_font_color="black", title_font_size=24 fig_0.show() Step 7: Create Figure from Plotly Express for group fig_all = px.bar(df_all, x='Year', y=[#'Total', #'Private Industrys', 'General Warehousing & Storage', 'General Warehousing & Storage - Under 16 Years', 'General Warehousing & Storage - 16 to 17 Years', 'General Warehousing & Storage - 18 to 19 Years', 'General Warehousing & Storage - 20 to 24 Years', 'General Warehousing & Storage - 25 to 34 Years', 'General Warehousing & Storage - 35 to 44 Years', 'General Warehousing & Storage - 45 to 54 Years', 'General Warehousing & Storage - 55 to 64 Years', 'General Warehousing & Storage - 65 Years and Over' text_auto=True, barmode='group', title= 'US Workplace ' + var_2 + '
' + ' Fatalities By Age Group
' + str(syear) + ' - ' + str(eyear), labels={ "Year": "Year", "value": "Number of Fatalities", "variable": "Group" color_discrete_map={#'Total':'#111173', #'Private Industrys':'#117326', Figure 'General Warehousing & Storage': '#113d73', .all 'General Warehousing & Storage - Under 16 Years': #11735f', 'General Warehousing & Storage - 16 to 17 Years': '#421173', 'General Warehousing & Storage - 18 to 19 Years': '#731168', 'General Warehousing & Storage - 20 to 24 Years': '#731111', 'General Warehousing & Storage - 25 to 34 Years': '#734511', 'General Warehousing & Storage - 35 to 44 Years': '#6b7311', 'General Warehousing & Storage - 45 to 54 Years': '#267311', 'General Warehousing & Storage - 55 to 64 Years': '#11733f', 'General Warehousing & Storage - 65 Years and Over': #117370' fig_all.update_traces(textposition="outside") fig_all.update_yaxes(autorange=False,fixedrange=False,range=['0','40'])

