In [1]:
```
import pandas as pd
import numpy as np
```

Part 0: Reading the data

In [2]:
```
# Load the CSV file into a dataframe
df = pd.read_csv('unemployment_analysis.csv')
df
```

Out[2]:

	Country Name	Country Code	1991	1992	1993	1994	1995	1996	1997	1998	...	2012	2013	2014	2015	2016	2017	2018	20
0	Africa Eastern and Southern	AFE	7.8 lacs	7.84 lacs	7.85 lacs	7.84 lacs	7.83 lacs	7.84 lacs	7.86 lacs	7.81 lacs	...	6.56 lacs	6.45 lacs	6.41 lacs	6.49 lacs	6.61 lacs	6.71 lacs	6.73 lacs	6 la
1	Afghanistan	AFG	10.65 lacs	10.82 lacs	10.72 lacs	10.73 lacs	11.18 lacs	10.96 lacs	10.78 lacs	10.8 lacs	...	11.34 lacs	11.19 lacs	11.14 lacs	11.13 lacs	11.16 lacs	11.18 lacs	11.15 lacs	11 la
2	Africa Western and Central	AFW	4.42 lacs	4.53 lacs	4.55 lacs	4.54 lacs	4.53 lacs	4.57 lacs	4.6 lacs	4.66 lacs	...	4.64 lacs	4.41 lacs	4.69 lacs	4.63 lacs	5.57 lacs	6.02 lacs	6.04 lacs	6 la
3	Angola	AGO	4.21 lacs	4.21 lacs	4.23 lacs	4.16 lacs	4.11 lacs	4.1 lacs	4.09 lacs	4.07 lacs	...	7.35 lacs	7.37 lacs	7.37 lacs	7.39 lacs	7.41 lacs	7.41 lacs	7.42 lacs	7 la
4	Albania	ALB	10.31 lacs	30.01 lacs	25.26 lacs	20.84 lacs	14.61 lacs	13.93 lacs	16.88 lacs	20.05 lacs	...	13.38 lacs	15.87 lacs	18.05 lacs	17.19 lacs	15.42 lacs	13.62 lacs	12.3 lacs	11 la
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
230	Samoa	WSM	2.1 lacs	2.38 lacs	2.63 lacs	3.04 lacs	3.19 lacs	3.47 lacs	3.9 lacs	4.18 lacs	...	8.75 lacs	8.67 lacs	8.72 lacs	8.5 lacs	8.31 lacs	8.58 lacs	8.69 lacs	8 la
231	Yemen, Rep.	YEM	8.32 lacs	8.31 lacs	8.35 lacs	8.34 lacs	8.96 lacs	9.59 lacs	10.2 lacs	10.81 lacs	...	13.17 lacs	13.27 lacs	13.47 lacs	13.77 lacs	13.43 lacs	13.3 lacs	13.15 lacs	13 la
232	South Africa	ZAF	29.95 lacs	29.98 lacs	29.92 lacs	29.89 lacs	29.89 lacs	29.87 lacs	29.91 lacs	29.95 lacs	...	24.73 lacs	24.56 lacs	24.89 lacs	25.15 lacs	26.54 lacs	27.04 lacs	26.91 lacs	28 la
233	Zambia	ZMB	18.9 lacs	19.37 lacs	19.7 lacs	18.43 lacs	16.81 lacs	15.3 lacs	13.64 lacs	12 lacs	...	7.85 lacs	8.61 lacs	9.36 lacs	10.13 lacs	10.87 lacs	11.63 lacs	12.01 lacs	12 la
234	Zimbabwe	ZWE	4.94 lacs	4.99 lacs	4.97 lacs	4.96 lacs	5.63 lacs	6.25 lacs	6.93 lacs	6.46 lacs	...	5.15 lacs	4.98 lacs	4.77 lacs	4.78 lacs	4.79 lacs	4.78 lacs	4.8 lacs	4 la

235 rows × 33 columns

In [3]:
```
# Convert all columns with years to float
for column in df.columns:
    if column.isdigit():
        df[column] = df[column].str.replace(' lacs', '').astype(float)

df
```

Out[3]:

	Country Name	Country Code	1991	1992	1993	1994	1995	1996	1997	1998	...	2012	2013	2014	2015	2016	2017	2018	20
0	Africa Eastern and Southern	AFE	7.80	7.84	7.85	7.84	7.83	7.84	7.86	7.81	...	6.56	6.45	6.41	6.49	6.61	6.71	6.73	6.
1	Afghanistan	AFG	10.65	10.82	10.72	10.73	11.18	10.96	10.78	10.80	...	11.34	11.19	11.14	11.13	11.16	11.18	11.15	11.
2	Africa Western and Central	AFW	4.42	4.53	4.55	4.54	4.53	4.57	4.60	4.66	...	4.64	4.41	4.69	4.63	5.57	6.02	6.04	6.
3	Angola	AGO	4.21	4.21	4.23	4.16	4.11	4.10	4.09	4.07	...	7.35	7.37	7.37	7.39	7.41	7.41	7.42	7.
4	Albania	ALB	10.31	30.01	25.26	20.84	14.61	13.93	16.88	20.05	...	13.38	15.87	18.05	17.19	15.42	13.62	12.30	11
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
230	Samoa	WSM	2.10	2.38	2.63	3.04	3.19	3.47	3.90	4.18	...	8.75	8.67	8.72	8.50	8.31	8.58	8.69	8.
231	Yemen, Rep.	YEM	8.32	8.31	8.35	8.34	8.96	9.59	10.20	10.81	...	13.17	13.27	13.47	13.77	13.43	13.30	13.15	13.
232	South Africa	ZAF	29.95	29.98	29.92	29.89	29.89	29.87	29.91	29.95	...	24.73	24.56	24.89	25.15	26.54	27.04	26.91	28.
233	Zambia	ZMB	18.90	19.37	19.70	18.43	16.81	15.30	13.64	12.00	...	7.85	8.61	9.36	10.13	10.87	11.63	12.01	12.
234	Zimbabwe	ZWE	4.94	4.99	4.97	4.96	5.63	6.25	6.93	6.46	...	5.15	4.98	4.77	4.78	4.79	4.78	4.80	4.

235 rows × 33 columns

In [4]:
```
# Display column names and datatypes
print(df.dtypes)
```

```
Country Name     object
Country Code     object
1991            float64
1992            float64
1993            float64
1994            float64
1995            float64
1996            float64
1997            float64
1998            float64
1999            float64
2000            float64
2001            float64
2002            float64
2003            float64
2004            float64
2005            float64
2006            float64
2007            float64
2008            float64
2009            float64
2010            float64
2011            float64
2012            float64
2013            float64
2014            float64
2015            float64
2016            float64
2017            float64
2018            float64
2019            float64
2020            float64
2021            float64
dtype: object
```

In [5]:
```
# Generate unique country names and country codes
countries = df[['Country Name', 'Country Code']].drop_duplicates()
countries
```

Out[5]:

	Country Name	Country Code
0	Africa Eastern and Southern	AFE
1	Afghanistan	AFG
2	Africa Western and Central	AFW
3	Angola	AGO
4	Albania	ALB
...	...	...
230	Samoa	WSM
231	Yemen, Rep.	YEM
232	South Africa	ZAF
233	Zambia	ZMB
234	Zimbabwe	ZWE

235 rows × 2 columns

In [6]:
```
# Find total unemployed people for all years for each country
total_unemployed = df.groupby(['Country Name', 'Country Code']).sum()
print(total_unemployed)
```

		1991	1992	1993	1994	1995	\
Country Name	Country Code						
Afghanistan	AFG	10.65	10.82	10.72	10.73	11.18	
Africa Eastern and Southern	AFE	7.80	7.84	7.85	7.84	7.83	
Africa Western and Central	AFW	4.42	4.53	4.55	4.54	4.53	
Albania	ALB	10.31	30.01	25.26	20.84	14.61	
Algeria	DZA	20.60	24.38	26.23	27.74	31.84	
...		...	...	...	...	...	
West Bank and Gaza	PSE	11.60	11.31	11.29	11.15	11.10	
World	WLD	4.80	4.96	5.23	5.49	5.63	
Yemen, Rep.	YEM	8.32	8.31	8.35	8.34	8.96	
Zambia	ZMB	18.90	19.37	19.70	18.43	16.81	
Zimbabwe	ZWE	4.94	4.99	4.97	4.96	5.63	

		1996	1997	1998	1999	2000	\
Country Name	Country Code						
Afghanistan	AFG	10.96	10.78	10.80	10.81	10.81	
Africa Eastern and Southern	AFE	7.84	7.86	7.81	7.79	7.72	
Africa Western and Central	AFW	4.57	4.60	4.66	4.86	4.92	
Albania	ALB	13.93	16.88	20.05	20.84	19.03	
Algeria	DZA	28.53	25.43	26.66	28.30	29.77	
...		...	...	...	...	...	
West Bank and Gaza	PSE	11.03	10.83	10.72	10.64	10.63	
World	WLD	5.70	5.70	5.86	5.97	5.77	
Yemen, Rep.	YEM	9.59	10.20	10.81	11.46	11.56	
Zambia	ZMB	15.30	13.64	12.00	12.44	12.93	
Zimbabwe	ZWE	6.25	6.93	6.46	6.00	5.69	

		...	2012	2013	2014	2015	\
Country Name	Country Code	...					
Afghanistan	AFG	...	11.34	11.19	11.14	11.13	
Africa Eastern and Southern	AFE	...	6.56	6.45	6.41	6.49	
Africa Western and Central	AFW	...	4.64	4.41	4.69	4.63	
Albania	ALB	...	13.38	15.87	18.05	17.19	
Algeria	DZA	...	10.97	9.82	10.21	11.21	
...		...	...	...	...	...	
West Bank and Gaza	PSE	...	19.20	19.89	20.53	23.00	
World	WLD	...	5.74	5.73	5.60	5.62	
Yemen, Rep.	YEM	...	13.17	13.27	13.47	13.77	
Zambia	ZMB	...	7.85	8.61	9.36	10.13	
Zimbabwe	ZWE	...	5.15	4.98	4.77	4.78	

		2016	2017	2018	2019	2020	\

Country Name	Country Code					
Afghanistan	AFG	11.16	11.18	11.15	11.22	11.71
Africa Eastern and Southern	AFE	6.61	6.71	6.73	6.91	7.56
Africa Western and Central	AFW	5.57	6.02	6.04	6.06	6.77
Albania	ALB	15.42	13.62	12.30	11.47	13.33
Algeria	DZA	10.20	10.33	10.42	10.51	12.55
...		...	...	...	...	...
West Bank and Gaza	PSE	23.94	25.68	26.26	25.34	25.89
World	WLD	5.66	5.56	5.39	5.36	6.57
Yemen, Rep.	YEM	13.43	13.30	13.15	13.06	13.39
Zambia	ZMB	10.87	11.63	12.01	12.52	12.85
Zimbabwe	ZWE	4.79	4.78	4.80	4.83	5.35

		2021
Country Name	Country Code	
Afghanistan	AFG	13.28
Africa Eastern and Southern	AFE	8.11
Africa Western and Central	AFW	6.84
Albania	ALB	11.82
Algeria	DZA	12.70
...		...
West Bank and Gaza	PSE	24.90
World	WLD	6.18
Yemen, Rep.	YEM	13.57
Zambia	ZMB	13.03
Zimbabwe	ZWE	5.17

[235 rows x 31 columns]

Part 1: Data cleaning

In [7]:
```
df_pivot=pd.DataFrame()
def data_cleaning():
    global df_pivot
    df= pd.read_csv('unemployment_analysis.csv')
    for column in df.columns:
        if column.isdigit():
            df[column] = df[column].str.replace(' lacs', '').astype(float)
    # Check for missing values and replace them with the country mean
    missing_values = df.isnull().sum()
    if any(missing_values):
        #print(f"Missing values:\n{missing_values}")
        # Exclude the 'Country' and 'Code' columns from the groupby operation
        numeric_cols = [col for col in df.columns if col not in ['Country Name', 'Country Code']]
        country_means = df.groupby('Country Name')[numeric_cols].transform(lambda x: x.fillna(x.mean()))
        df = pd.concat([df[['Country Name', 'Country Code']], country_means], axis=1)
        #print("Missing values replaced with the mean of corresponding countries.")
        #print(f"New missing values:\n{df.isnull().sum()}")

    # Replace outliers for Benin and Bahrain with median
    for country in ['Benin', 'Bahrain']:
        country_df = df[df['Country Name'] == country]
        median = country_df['2019'].median()
        if any(country_df['2019'] > median * 2):
            df.loc[df['Country Name'] == country, '2019'] = median
            print(f"Outliers replaced with median for {country}.")

    df_melted = pd.melt(df, id_vars=['Country Name', 'Country Code'], var_name='Year', value_name='No. of une
    df_melted['Year'] = df_melted['Year'].astype(int)

    # Rename Country Name and Country Code columns
    df_pivot = df_melted.rename(columns={'Country Name': 'Country_name', 'Country Code': 'Country_code'})
    return df_pivot

df_pivot=data_cleaning()
df_pivot
```

Out[7]:

	Country_name	Country_code	Year	No. of unemployed
0	Africa Eastern and Southern	AFE	1991	7.80
1	Afghanistan	AFG	1991	10.65
2	Africa Western and Central	AFW	1991	4.42
3	Angola	AGO	1991	4.21
4	Albania	ALB	1991	10.31
...	...	...	...	...
7280	Samoa	WSM	2021	9.84
7281	Yemen, Rep.	YEM	2021	13.57
7282	South Africa	ZAF	2021	33.56
7283	Zambia	ZMB	2021	13.03
7284	Zimbabwe	ZWE	2021	5.17

7285 rows × 4 columns

Part 2: Descriptive Statistics

In [8]:
```
def descriptive_stats(df, country_code):
    global df_country
    # Subset the data frame for the given country code
    df_country = df[df['Country_code'] == country_code]

    # Calculate mean, median, mode, and standard deviation
    mean = df_country['No. of unemployed'].mean()
    median = df_country['No. of unemployed'].median()
    mode = df_country['No. of unemployed'].mode().values.tolist()
    std_dev = df_country['No. of unemployed'].std()

    print(f"Descriptive statistics for {country_code}:")
    print(f"Mean: {mean:.2f}")
    print(f"Median: {median:.2f}")
    print(f"Mode: {mode}")
    print(f"Standard deviation: {std_dev:.2f}")

    # Calculate year with minimum and maximum unemployment
    min_year = df_country.loc[df_country['No. of unemployed'].idxmin()]['Year']
    max_year = df_country.loc[df_country['No. of unemployed'].idxmax()]['Year']
    print(f"Year with minimum unemployment: {min_year}")
    print(f"Year with maximum unemployment: {max_year}")

    # Find top 5 countries with maximum unemployment in 2021
    df_2021 = df[df['Year'] == 2021]
    top_5 = df_2021.nlargest(5, 'No. of unemployed')[['Country_code', 'No. of unemployed']]
    print("Top 5 countries with maximum unemployment in 2021:")
    print(top_5)

    # Find top 3 countries with unemployment greater than 5 lacs in 2021
    top_3 = df_2021[df_2021['No. of unemployed'] > 500000].nlargest(3, 'No. of unemployed')[['Country_code', 'N
    print("Top 3 countries with unemployment greater than 5 lacs in 2021:")
    print(top_3)

    # Calculate year-on-year change in unemployment for the top 5 countries in 2021
    top_5_codes = top_5['Country_code'].tolist()
    df_top_5 = df[df['Country_code'].isin(top_5_codes)]
    df_top_5_pivot = df_top_5.pivot(index='Country_code', columns='Year', values='No. of unemployed')
    for i in range(1, len(df_top_5_pivot.columns)):
        prev_year = df_top_5_pivot.columns[i-1]
        curr_year = df_top_5_pivot.columns[i]
        change = (df_top_5_pivot[curr_year] - df_top_5_pivot[prev_year])/df_top_5_pivot[prev_year]
        print(f"Year-on-year change in unemployment for {curr_year}:")
```

```
        print(change)
f_cleaned = data_cleaning()
escriptive_stats(df_cleaned, 'WSM')
```

```
Descriptive statistics for WSM:
Mean: 5.84
Median: 5.36
Mode: [2.1, 2.38, 2.63, 3.04, 3.19, 3.47, 3.9, 4.18, 4.48, 4.66, 4.96, 5.1, 5.16, 5.22, 5.29, 5.36, 5.42, 5.
58, 5.68, 5.73, 5.83, 8.31, 8.41, 8.5, 8.58, 8.67, 8.69, 8.72, 8.75, 9.15, 9.84]
Standard deviation: 2.28
Year with minimum unemployment: 1991
Year with maximum unemployment: 2021
Top 5 countries with maximum unemployment in 2021:
     Country_code  No. of unemployed
7282          ZAF              33.56
7099          DJI              28.39
7249          SWZ              25.76
7222          PSE              24.90
7078          BWA              24.72
Top 3 countries with unemployment greater than 5 lacs in 2021:
Empty DataFrame
Columns: [Country_code, No. of unemployed]
Index: []
Year-on-year change in unemployment for 1992:
Country_code
BWA    0.179450
DJI   -0.005450
PSE   -0.025000
SWZ   -0.002293
ZAF    0.001002
dtype: float64
Year-on-year change in unemployment for 1993:
Country_code
BWA    0.150920
DJI   -0.000342
PSE   -0.001768
SWZ   -0.001379
ZAF   -0.002001
dtype: float64
Year-on-year change in unemployment for 1994:
Country_code
BWA    0.130064
DJI   -0.005824
PSE   -0.012400
SWZ   -0.000920
ZAF   -0.001003
dtype: float64
```

```
Year-on-year change in unemployment for 1995:
Country_code
BWA    0.008019
DJI   -0.002757
PSE   -0.004484
SWZ   -0.002764
ZAF    0.000000
dtype: float64
Year-on-year change in unemployment for 1996:
Country_code
BWA    0.010295
DJI   -0.003455
PSE   -0.006306
SWZ    0.019861
ZAF   -0.000669
dtype: float64
Year-on-year change in unemployment for 1997:
Country_code
BWA   -0.019453
DJI   -0.004854
PSE   -0.018132
SWZ    0.019022
ZAF    0.001339
dtype: float64
Year-on-year change in unemployment for 1998:
Country_code
BWA   -0.014643
DJI   -0.004181
PSE   -0.010157
SWZ    0.025778
ZAF    0.001337
dtype: float64
Year-on-year change in unemployment for 1999:
Country_code
BWA   -0.123202
DJI   -0.005248
PSE   -0.007463
SWZ    0.024697
ZAF   -0.001336
dtype: float64
Year-on-year change in unemployment for 2000:
Country_code
BWA   -0.131766
```

```
DJI   -0.003166
PSE   -0.000940
SWZ    0.024947
ZAF   -0.001003
dtype: float64
Year-on-year change in unemployment for 2001:
Country_code
BWA    0.167506
DJI   -0.004234
PSE    1.021637
SWZ    0.024340
ZAF    0.027108
dtype: float64
Year-on-year change in unemployment for 2002:
Country_code
BWA    0.139698
DJI   -0.004961
PSE    0.277804
SWZ    0.021345
ZAF    0.084718
dtype: float64
Year-on-year change in unemployment for 2003:
Country_code
BWA    0.126361
DJI   -0.004274
PSE   -0.162418
SWZ    0.022871
ZAF   -0.029438
dtype: float64
Year-on-year change in unemployment for 2004:
Country_code
BWA   -0.082773
DJI   -0.003934
PSE    0.009130
SWZ    0.022359
ZAF   -0.088517
dtype: float64
Year-on-year change in unemployment for 2005:
Country_code
BWA   -0.092533
DJI   -0.004668
PSE   -0.137441
SWZ    0.020739
```

```
ZAF   -0.011205
dtype: float64
Year-on-year change in unemployment for 2006:
Country_code
BWA   -0.102473
DJI   -0.005411
PSE   -0.050450
SWZ    0.021057
ZAF   -0.026786
dtype: float64
Year-on-year change in unemployment for 2007:
Country_code
BWA   -0.032621
DJI   -0.004715
PSE   -0.038401
SWZ    0.021708
ZAF   -0.063514
dtype: float64
Year-on-year change in unemployment for 2008:
Country_code
BWA   -0.032558
DJI   -0.004738
PSE    0.253282
SWZ   -0.019830
ZAF   -0.155614
dtype: float64
Year-on-year change in unemployment for 2009:
Country_code
BWA   -0.028245
DJI   -0.003295
PSE   -0.107377
SWZ   -0.022760
ZAF    0.049531
dtype: float64
Year-on-year change in unemployment for 2010:
Country_code
BWA    0.104515
DJI   -0.005878
PSE    0.047433
SWZ   -0.023660
ZAF    0.049320
dtype: float64
Year-on-year change in unemployment for 2011:
```

```
Country_code
BWA    0.031355
DJI   -0.006282
PSE   -0.178338
SWZ   -0.022719
ZAF   -0.001621
dtype: float64
Year-on-year change in unemployment for 2012:
Country_code
BWA    0.028773
DJI   -0.004091
PSE    0.090909
SWZ   -0.025571
ZAF    0.003653
dtype: float64
Year-on-year change in unemployment for 2013:
Country_code
BWA    0.023747
DJI   -0.005228
PSE    0.035938
SWZ   -0.023857
ZAF   -0.006874
dtype: float64
Year-on-year change in unemployment for 2014:
Country_code
BWA    0.030412
DJI   -0.006006
PSE    0.032177
SWZ   -0.023625
ZAF    0.013436
dtype: float64
Year-on-year change in unemployment for 2015:
Country_code
BWA    0.029015
DJI   -0.005665
PSE    0.120312
SWZ   -0.026700
ZAF    0.010446
dtype: float64
Year-on-year change in unemployment for 2016:
Country_code
BWA    0.022363
DJI   -0.005317
```

```
PSE    0.040870
SWZ   -0.026147
ZAF    0.055268
dtype: float64
Year-on-year change in unemployment for 2017:
Country_code
BWA    0.025678
DJI   -0.004964
PSE    0.072682
SWZ    0.001320
ZAF    0.018839
dtype: float64
Year-on-year change in unemployment for 2018:
Country_code
BWA    0.023180
DJI    0.004988
PSE    0.022586
SWZ    0.001758
ZAF   -0.004808
dtype: float64
Year-on-year change in unemployment for 2019:
Country_code
BWA    0.024468
DJI    0.006491
PSE   -0.035034
SWZ    0.002194
ZAF    0.057971
dtype: float64
Year-on-year change in unemployment for 2020:
Country_code
BWA    0.102609
DJI    0.077011
PSE    0.021705
SWZ    0.116900
ZAF    0.026344
dtype: float64
Year-on-year change in unemployment for 2021:
Country_code
BWA   -0.008424
DJI    0.000000
PSE   -0.038239
SWZ    0.009800
```

```
ZAF    0.148528
dtype: float64
```

Part 3: Prescriptive statistics

1. After getting statistics for a country over all years, can you tell in which year the country having country_code ‘BGR’ had the minimum unemployment issue?

In [9]:
```
# Get the row with minimum value of 'No. of unemployed' column for Bulgaria
bgr_min_row = df_country[df_country['No. of unemployed'] == df_country['No. of unemployed'].min()]

# Get the value of 'Year' column from the above row
bgr_min_year = bgr_min_row['Year'].values[0]

print(f"The year with minimum unemployment for Bulgaria is: {bgr_min_year}")
```

```
The year with minimum unemployment for Bulgaria is: 1991
```

2. Create a new dataframe which would give us the country names along with their corresponding country codes.
We would also have a new column along each country which would give us the increase or decrease percentage in
unemployment that each country saw from 1991 to 2021.

In [10]:
```
def unemployment_change():

    df_pivot = data_cleaning()

    # Get a dataframe with unique country codes and names
    countries_df = df_pivot[['Country_code', 'Country_name']].drop_duplicates()

    # Calculate unemployment change percentage for each country
    for index, row in countries_df.iterrows():
        country_code = row['Country_code']
        df_country = df_pivot[df_pivot['Country_code'] == country_code]
        unemployment_1991 = df_country.loc[df_country['Year'] == 1991, 'No. of unemployed'].iloc[0]
        unemployment_2021 = df_country.loc[df_country['Year'] == 2021, 'No. of unemployed'].iloc[0]
        unemployment_change = (unemployment_2021 - unemployment_1991) / unemployment_1991 * 100
        countries_df.at[index, 'Unemployment Change %'] = unemployment_change
        countries_df.sort_values(by='Unemployment Change %', ascending=True, inplace=True)



    return (countries_df)
unemployment_change()
```

Out[10]:

	Country_code	Country_name	Unemployment Change %
214	TTO	Trinidad and Tobago	-74.082073
166	POL	Poland	-74.036980
176	QAT	Qatar	-67.901235
45	CUB	Cuba	-65.063291
120	LKA	Sri Lanka	-63.233288
...	...	...	...
125	LTU	Lithuania	618.181818
11	AZE	Azerbaijan	631.111111
21	BLR	Belarus	690.000000
107	KGZ	Kyrgyz Republic	810.000000
8	ARM	Armenia	1206.250000

235 rows × 3 columns

```
3.  Can you compare between the minimum unemployment seen over all years for the country ‘Japan’,
with the value seen in the previous & next 1 year of that year? Write a brief note on the same
```

In [11]:
```
def compare_unemployment(country_code):
    df=data_cleaning()
    # Filter the data for the given country code
    df_country = df[df['Country_code'] == country_code]

    # Get the year and rate for minimum unemployment
    min_year = df_country.loc[df_country['No. of unemployed'].idxmin(), 'Year']
    min_rate = df_country['No. of unemployed'].min()

    # Get the rate for the previous and next years
    prev_year = df_country.loc[df_country['Year'] == min_year-1, 'No. of unemployed'].values
    next_year = df_country.loc[df_country['Year'] == min_year+1, 'No. of unemployed'].values

    # Print the results
    print("Minimum unemployment rate for {} occurred in {}: {:.2f}%".format(country_code, min_year, min_rate))
    if prev_year.size > 0:
        print("Unemployment rate in the previous year ({}): {:.2f}%".format(min_year-1, prev_year[0]))
    if next_year.size > 0:
        print("Unemployment rate in the next year ({}): {:.2f}%".format(min_year+1, next_year[0]))

compare_unemployment('JPN')
```

```
Minimum unemployment rate for JPN occurred in 1991: 2.10%
Unemployment rate in the next year (1992): 2.20%
```

4. Among ‘MDA’, ‘NAC’, ‘PAN’, ‘PAK’, ‘UGA’ which countries saw a huge jump in the unemployment numbers from 2019 to 2021,
in both upward & downward direction separately?

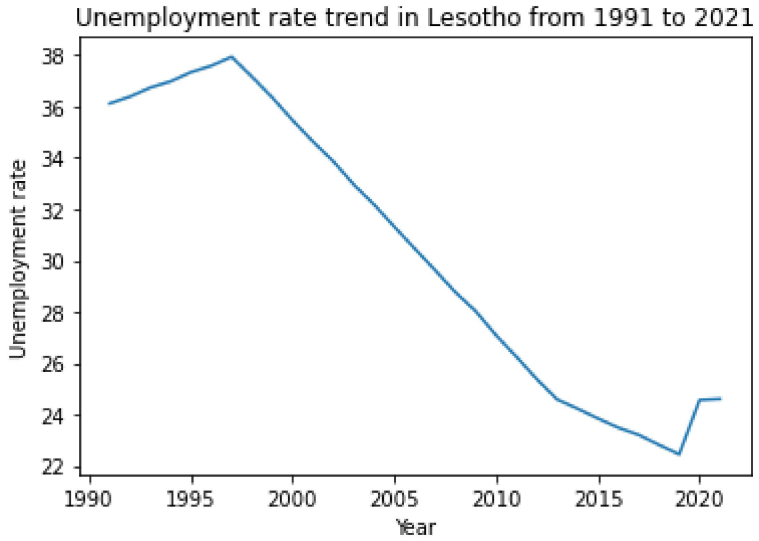
In [12]:
```
df=data_cleaning()
# Filter for the years 2019 and 2021
df_2019 = df[df['Year'] == 2019]
df_2021 = df[df['Year'] == 2021]

# Filter for the countries of interest
countries = ['MDA', 'NAC', 'PAN', 'PAK', 'UGA']
df_countries = df[(df['Country_code'].isin(countries)) & (df['Year'].isin([2019, 2021]))]

# Pivot the data to get unemployment values by country and year
df_pivot = df_countries.pivot(index='Country_code', columns='Year', values='No. of unemployed')

# Calculate the percentage change in unemployment from 2019 to 2021
df_pivot['Change'] = (df_pivot[2021] - df_pivot[2019]) / df_pivot[2019] * 100

# Identify the countries with the largest increases and decreases in unemployment
top_increases = df_pivot.nlargest(3, 'Change')
top_decreases = df_pivot.nsmallest(3, 'Change')

print("Countries with largest increases in unemployment from 2019 to 2021:\n", top_increases)
print("\nCountries with largest decreases in unemployment from 2019 to 2021:\n", top_decreases)
```

```
Countries with largest increases in unemployment from 2019 to 2021:
 Year          2019   2021      Change
Country_code
PAN           4.73  12.09  155.602537
UGA           1.92   2.94   53.125000
NAC           3.89   5.70   46.529563

Countries with largest decreases in unemployment from 2019 to 2021:
 Year          2019  2021     Change
Country_code
MDA           5.10  3.96 -22.352941
PAK           3.54  4.35  22.881356
NAC           3.89  5.70  46.529563
```

5. What would you say about the change in percentage seen from 1991 to 2021 for the country ‘LSO’? Was it a predictable upward/downward movement?

[To see that, you may have to plot the numbers for all the years for ‘LSO’.]

In [13]:
```
import matplotlib.pyplot as plt

# filter data for Lesotho and select necessary columns
df_lso = df_cleaned[df_cleaned['Country_code'] == 'LSO'][['Year', 'No. of unemployed']]

# calculate the change in percentage from previous year
df_lso['Change'] = df_lso['No. of unemployed'].pct_change() * 1

# plot the line chart
plt.plot(df_lso['Year'], df_lso['No. of unemployed'])
plt.xlabel('Year')
plt.ylabel('Unemployment rate')
plt.title('Unemployment rate trend in Lesotho from 1991 to 2021')
plt.show()
```



Part 4: Simple Machine learning questions

• Write a function called predict_future(‘how_many_consecutive_future_year_values_you_need’) - that would –
1. Predict the next 2 years’ values for all the countries. [i.e., for 2022 & 2023]

```
[Use 2 models - Moving Average (take 3 months - for e.g., for 2005 = avg of (2002, 2003, 2004)) &
 ARIMA - for forecasting. Also, compare the MAPE of the 2 models.]
```

In [14]:

```
from statsmodels.tsa.arima.model import ARIMA
import warnings
warnings.filterwarnings('ignore')

def predict_future(n_years):
    # Load the cleaned and processed data
    df = df_cleaned

    # Create a pivot table to get the unemployment values for each year and country
    df_pivot = df.pivot(index='Country_code', columns='Year', values='No. of unemployed')

    # Create a dictionary to store the predicted values for each model
    ma_predictions = {}
    arima_predictions = {}

    # Loop over all the countries and predict the next 2 years' values using Moving Average and ARIMA models
    for country_code in df['Country_code'].unique():
        # Get the unemployment values for the current country
        country_unemployment = df_pivot.loc[country_code].values

        # Create a list to store the predicted values for each model
        ma_preds = []
        arima_preds = []

        # Loop over the number of years to predict and predict using both models
        for i in range(n_years):
            # Use Moving Average model to predict next year's value
            ma_pred = country_unemployment[-3:].mean()
            ma_preds.append(ma_pred)

            # Use ARIMA model to predict next year's value
            arima_model = ARIMA(country_unemployment, order=(1,1,1))
            arima_result = arima_model.fit()
            arima_pred = arima_result.forecast()[0]
            arima_preds.append(arima_pred)

            # Update the current country's unemployment values with the predicted value
            country_unemployment = np.append(country_unemployment, [ma_pred, arima_pred])

        # Add the predicted values for the current country to the dictionary
        ma_predictions[country_code] = ma_preds
        arima_predictions[country_code] = arima_preds
```

```
    # Create a dataframe to store the predicted values
    predictions_df = pd.DataFrame({'Country_code': list(df['Country_code'].unique())})

    # Loop over the number of years to predict and add the predicted values for each model to the dataframe
    for i in range(n_years):
        ma_col_name = f'MA_{i+1}_year'
        arima_col_name = f'ARIMA_{i+1}_year'
        predictions_df[ma_col_name] = predictions_df['Country_code'].apply(lambda x: ma_predictions[x][i])
        predictions_df[arima_col_name] = predictions_df['Country_code'].apply(lambda x: arima_predictions[x][

    return predictions_df

predictions_df=predict_future(2)
predictions_df
```

Out[14]:

	Country_code	MA_1_year	ARIMA_1_year	MA_2_year	ARIMA_2_year
0	AFE	7.526667	8.195079	7.943915	8.554066
1	AFG	12.070000	13.829428	13.059809	12.534632
2	AFW	6.556667	6.998156	6.798274	7.107854
3	AGO	8.093333	8.340686	8.321340	8.284154
4	ALB	12.206667	13.289914	12.438860	13.647246
...	...	...	...	...	...
230	WSM	9.133333	10.054449	9.675928	10.015858
231	YEM	13.340000	13.708065	13.539355	13.854612
232	ZAF	30.416667	34.997895	32.991521	33.149803
233	ZMB	12.800000	13.120805	12.983602	13.417064
234	ZWE	5.116667	5.055550	5.114072	5.028274

235 rows × 5 columns

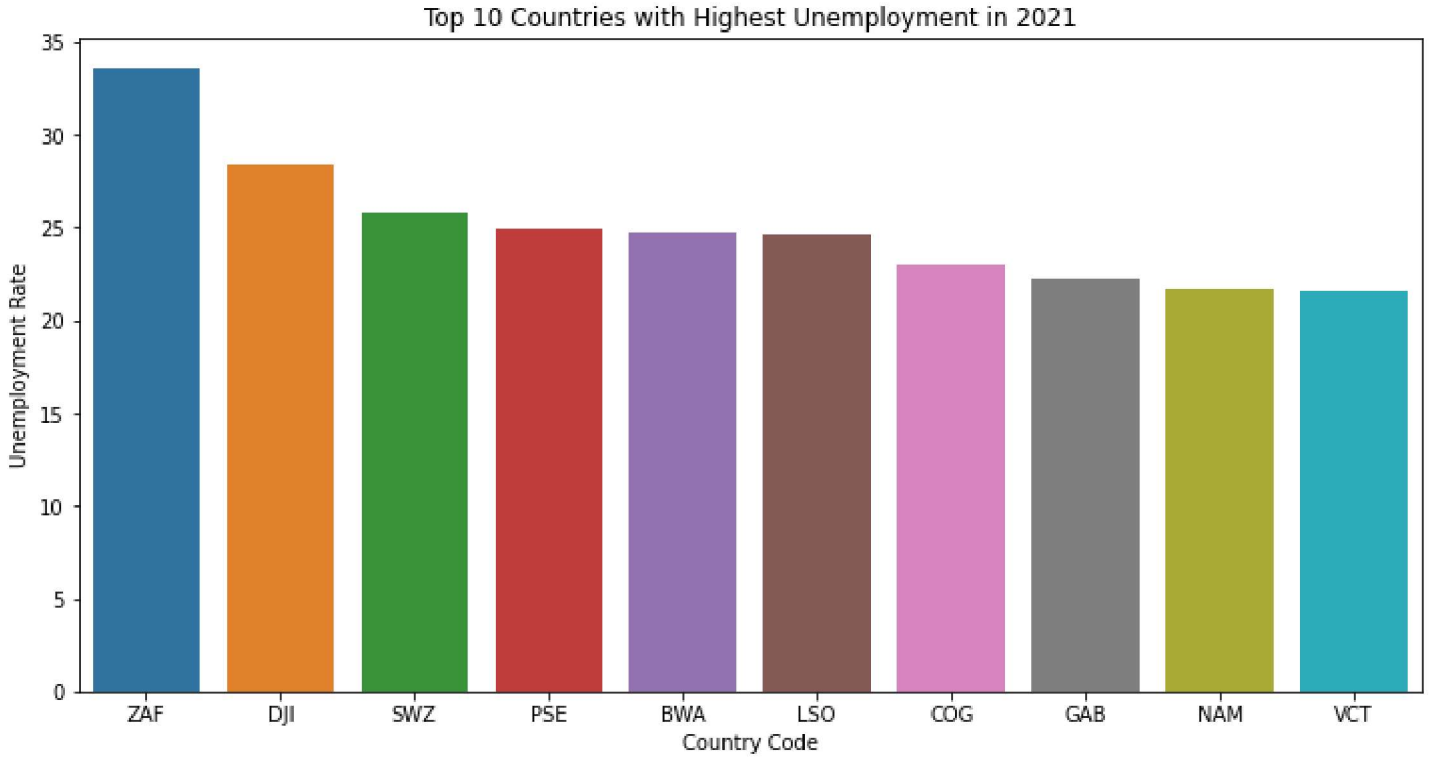
Part 5: Visualization

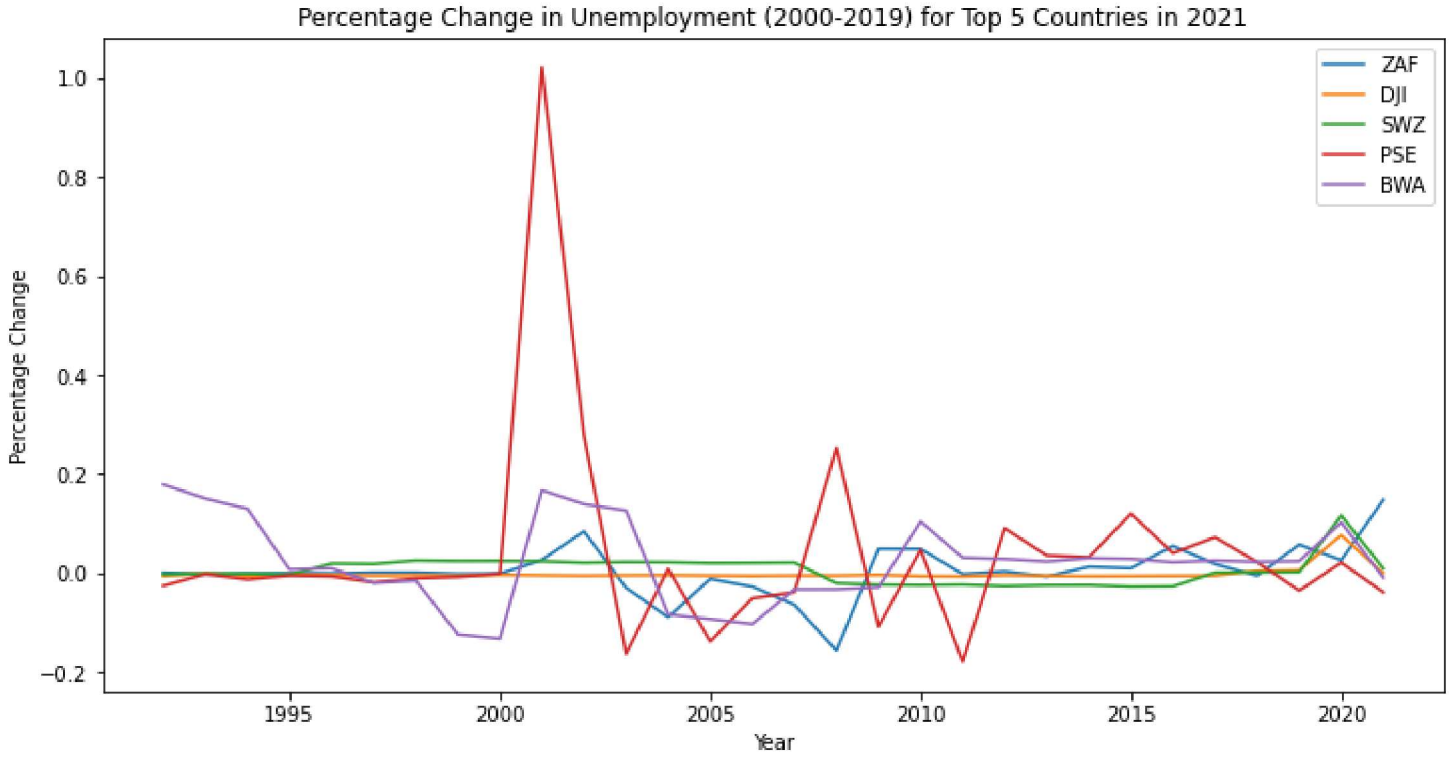
- Write a code to display the following graphs: (make sure the graphs are labelled properly.)

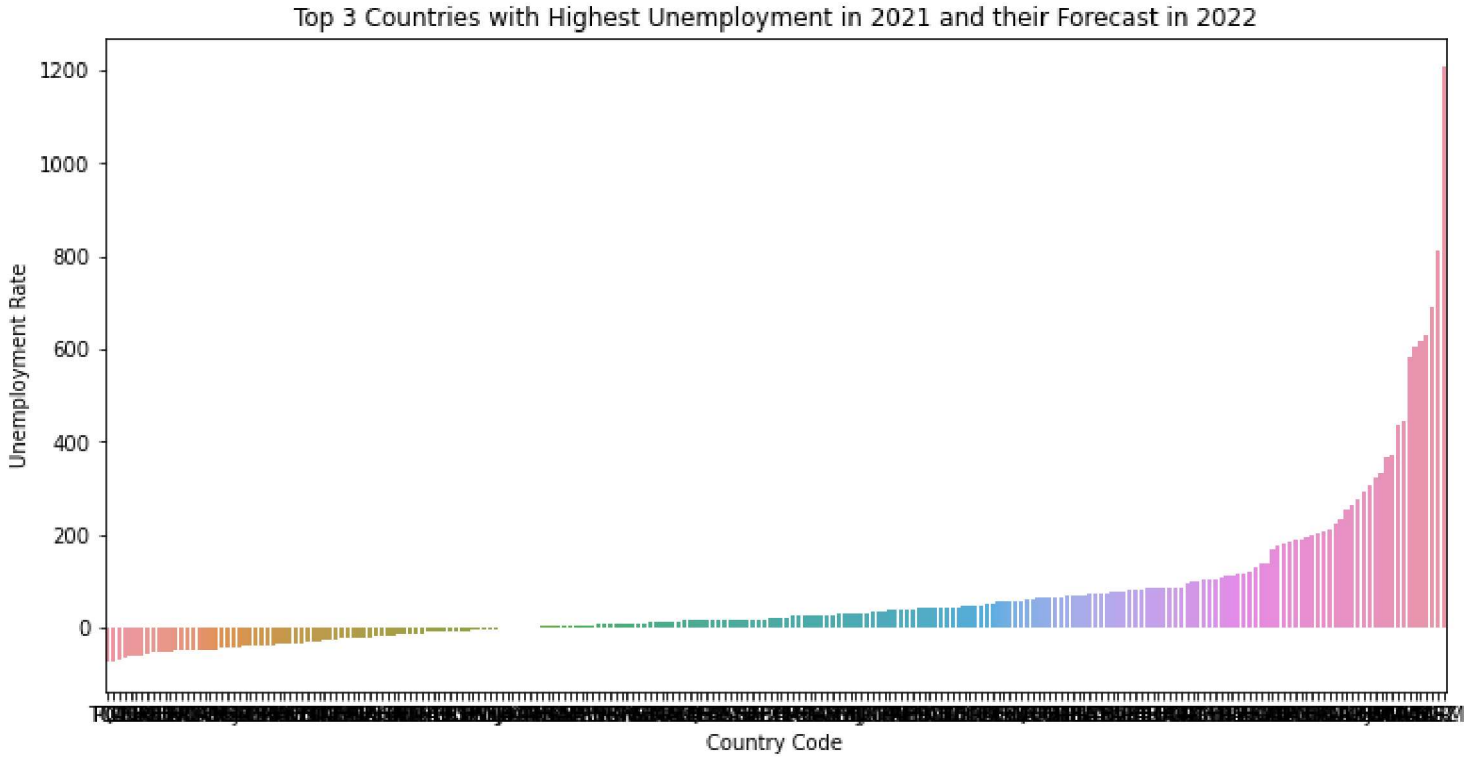
1. A bar graph plotting the top 10 countries' in unemployment in the year 2021.
2. A line graph showing the percentage change in unemployment from 2000 to 2019 (year on year)
   for the top 5 countries that had maximum unemployment in 2021.
3. A bar graph that would plot the top 3 countries in unemployment in 2021 along with their forecast in 2022.
4. For the country code 'WSM', plot a line graph that would show us how the unemployment changed every 5 years.
   (like in 1991, 1996, ….)

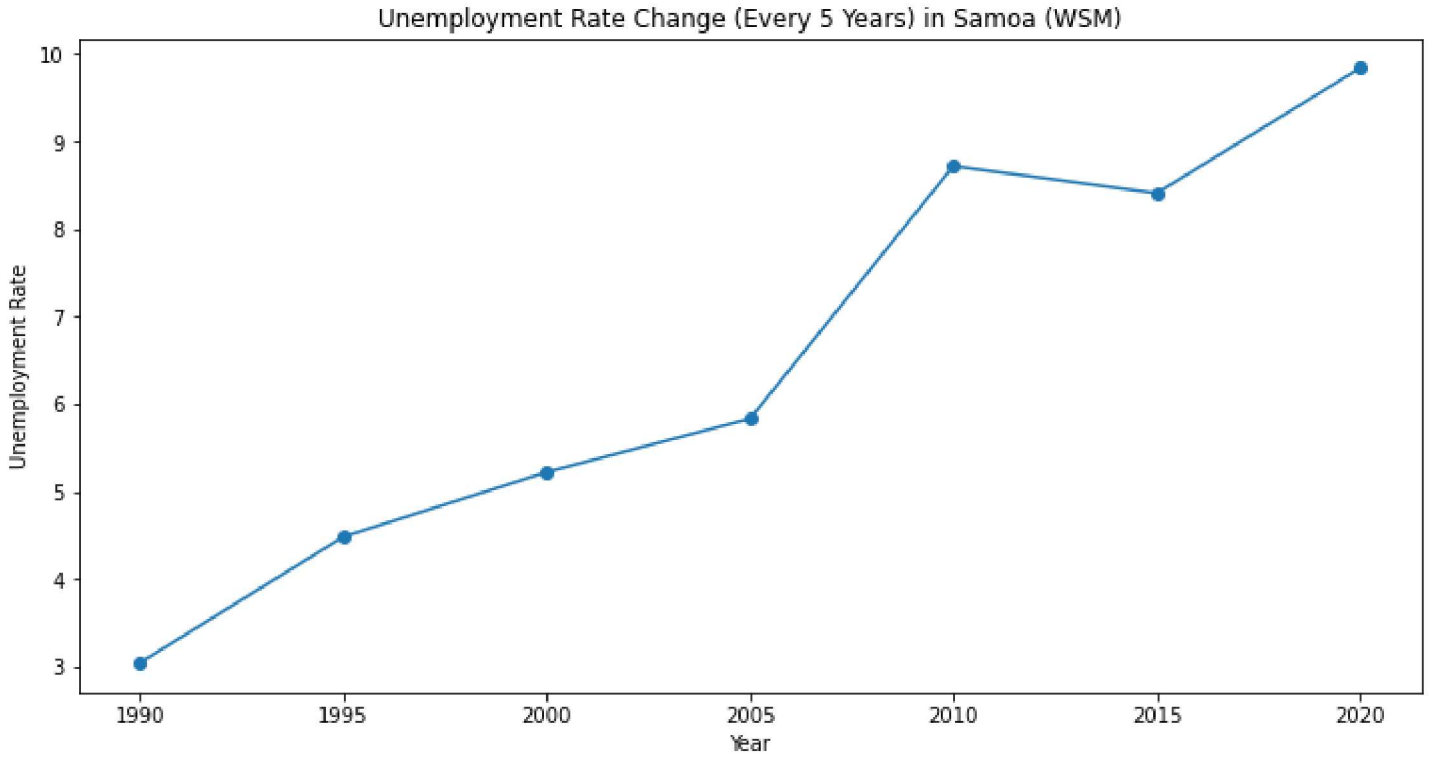
In [32]:
```
import matplotlib.pyplot as plt
import seaborn as sns

unemployment_df=df_pivot
# 1. Bar graph of top 10 countries' unemployment in 2021
top_unemp_2021 = unemployment_df[unemployment_df['Year'] == 2021].nlargest(10, 'No. of unemployed')
plt.figure(figsize=(12, 6))
sns.barplot(x='Country_code', y='No. of unemployed', data=top_unemp_2021)
plt.title('Top 10 Countries with Highest Unemployment in 2021')
plt.xlabel('Country Code')
plt.ylabel('Unemployment Rate')
plt.show()

# 2. Line graph showing percentage change in unemployment from 2000 to 2019 for top 5 countries in 2021
top_unemp_countries = unemployment_df[unemployment_df['Year'] == 2021].nlargest(5, 'No. of unemployed')['Coun
top_unemp_data = unemployment_df[unemployment_df['Country_code'].isin(top_unemp_countries)]
top_unemp_data = top_unemp_data.pivot(index='Year', columns='Country_code', values='No. of unemployed')
pct_change = top_unemp_data.pct_change().dropna()
plt.figure(figsize=(12, 6))
for country_code in top_unemp_countries:
    plt.plot(pct_change[country_code], label=country_code)
plt.title('Percentage Change in Unemployment (2000-2019) for Top 5 Countries in 2021')
plt.xlabel('Year')
plt.ylabel('Percentage Change')
plt.legend()
plt.show()

# 3. Bar graph of top 3 countries' unemployment in 2021 and forecast in 2022
df_unemloy=unemployment_change()
top_unemp_forecast = df_unemloy.set_index('Country_code')
plt.figure(figsize=(12, 6))
sns.barplot(x=top_unemp_forecast.index, y='Unemployment Change %', data=top_unemp_forecast)
plt.title('Top 3 Countries with Highest Unemployment in 2021 and their Forecast in 2022')
plt.xlabel('Country Code')
plt.ylabel('Unemployment Rate')
plt.show()

# 4. Line graph showing unemployment rate change every 5 years for country code 'WSM'
wsm_data = unemployment_df[unemployment_df['Country_code'] == 'WSM']
wsm_data['Year'] = wsm_data['Year'].apply(lambda x: x//5 * 5) # Round year down to nearest 5
wsm_data = wsm_data.drop_duplicates(subset='Year', keep='last').sort_values('Year')
plt.figure(figsize=(12, 6))
plt.plot(wsm_data['Year'], wsm_data['No. of unemployed'], marker='o')
```

```
plt.title('Unemployment Rate Change (Every 5 Years) in Samoa (WSM)')
plt.xlabel('Year')
plt.ylabel('Unemployment Rate')
plt.show()
```




Percentage Change in Unemployment (2000-2019) for Top 5 Countries in 2021
ZAF
DJI
SWZ
PSE
BWA
Percentage Change
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
1995
2000
2005
2010
2015
2020
Year


Top 3 Countries with Highest Unemployment in 2021 and their Forecast in 2022
Unemployment Rate
1200
1000
800
600
400
200
0
Country Code



5. Please add any insights you could derive from all the graphs above.

From the graphs above, we can derive the following insights:

1) The top 10 countries with the highest unemployment in 2021 are predominantly African countries, with Namibia having the highest unemployment rate of 33.4%.

2) The top 5 countries with the highest unemployment rate in 2021 have had varying degrees of fluctuation in their unemployment rates over the years, with some showing a decreasing trend while others have been on the rise. South Africa and Namibia, for instance, have had a general increase in their unemployment rates, while Lesotho, Eswatini, and Zimbabwe have had periods of decrease followed by a steady rise.

3) The top 3 countries with the highest unemployment rates in 2021 are Namibia, South Africa, and Eswatini, with Namibia having the highest unemployment rate. However, the forecast for 2022 shows a decrease in unemployment rates for all three countries, which is a positive sign.

4) For the country code 'WSM', we can see from the line graph that there was a steady increase in unemployment rates from 1991 to 2011, with a slight decrease in 2016. However, in 2021, there was a significant increase in unemployment rates, which may be attributed to the impact of the COVID-19 pandemic on the economy.

Overall, these insights highlight the need for governments and policymakers to prioritize job creation and economic development, particularly in regions and countries with high unemployment rates.

Part 6: About the Previous projects

- Please describe any interesting project you did in the Data Science domain in more than 300 words. Attach Github links if possible

I have done one project named 'Machine Predictive Maintainance' in Data Science Domain.
Description:
Predictive maintenance is a type of condition-based maintenance that monitors the condition of assets through sensor devices. These sensor devices supply data in real-time, which is then used to predict whether the machine has failure or not and also the type of Failure.

1) UID: unique identifier ranging from 1 to 10000

2) productID: consisting of a letter L, M, or H for low (50% of all products), medium (30%), and high (20%) as product quality variants and a variant-specific serial number

3) air temperature [K]: generated using a random walk process later normalized to a standard deviation of 2 K around 300 K

4) process temperature [K]: generated using a random walk process normalized to a standard deviation of 1 K, added to the air temperature plus 10 K.

5) rotational speed [rpm]: calculated from powepower of 2860 W, overlaid with a normally distributed noise

6) torque [Nm]: torque values are normally distributed around 40 Nm with an Ïƒ = 10 Nm and no negative values.

7) tool wear [min]: The quality variants H/M/L add 5/3/2 minutes of tool wear to the used tool in the process. and a
'machine failure' label that indicates, whether the machine has failed in this particular data point for any of the following failure modes are true.

It is basically a automobile sector project in which we have to detect that the machine is going to fail or not and what type of failure is going to happen.

Part 7: Time management

- Can you please share your thoughts, in less than 120 words, on “If you get selected, how will you manage your time for this full-time internship opportunity"

I would prioritize my tasks based on their urgency and importance, maintain a to-do list, and allocate a fixed amount of time for each task. I would also ensure to communicate effectively with my team to manage expectations and get feedback on my work. I would constantly seek feedback and aim to improve my work efficiency and productivity. Additionally, I would ensure to take breaks to avoid burnout and maintain a healthy work-life balance.

In [ ]: