**A Python Data Visualization Inflation Analysis on Food, Energy, and GDP**

**Introduction**

Inflation is one of the most critical macroeconomic indicators, influencing economic stability, purchasing power, and living standards across the globe. This study examines the dynamics of inflation, food price indices, and energy price trends from 2000 to 2023 across selected countries, including Turkey, Iran, and Egypt. Inflation trends reveal the interplay between economic policies, global supply chains, and local market conditions, which are essential for understanding the broader economic landscape.

As noted in several studies, inflation has cascading effects on socioeconomic factors. For instance, rising food prices disproportionately impact low-income households, while energy price volatility directly affects production costs and economic output (Burke & Stephens, 2018; Stern & Burke, 2019). By analyzing these trends, this research aims to provide actionable insights into the complex interdependencies between inflation, energy consumption, and GDP growth.

#### ****About the Dataset****

The dataset used in this analysis is sourced from the World Bank and contains comprehensive data on inflation and price indices for up to 209 countries, spanning 1970 to 2023. Key indicators include:

* **Food Consumer Price Index (FCPI\_M):** Monthly data tracking food price inflation.
* **Energy Consumer Price Index (ECPI\_M):** Monthly data tracking energy price inflation.
* **GDP Deflator Growth Rate (DEF\_A):** Annual data representing overall inflation.

This dataset enables cross-country and cross-sector analysis, uncovering trends and disparities in economic performance.

#### ****LightningChart Python****

LightningChart Python was chosen as the primary visualization tool due to its exceptional performance in handling large datasets and creating real-time interactive visualizations. Its capabilities include:

1. **Seamless Data Handling:** Processes millions of data points without performance degradation.
2. **Real-Time Updates:** Facilitates dynamic visualization, making it ideal for time-series and comparative analyses.
3. **Customizability:** Offers a wide range of chart types, enabling tailored visualizations for specific insights.

**Chart Types and Analysis**

1. **Line Charts**: Energy Price Index and Food Price Index for Ukraine (2018-2023)

Purpose: Analyze the year-on-year trends for Ukraine.

2. **Heatmaps:** Price Indices for Albania (2000-2023)

* Food Price Index:
* Energy Price Index:

3. **Real-Time Dashboard:** Energy Price Index Growth for Continents (2000-2023)

4. **Bar Chart**: Yearly Comparison of FCPI and ECPI for Turkey (2018-2023)

5. **Radar Chart**: Inflation and Price Indices for Turkey (2015-2023)

6. **Balloon Race Chart**: GDP Deflator Growth Rate and Energy Index for Turkey, Iran, and Egypt (2000-2023)

### Optimizing Data Visualization Performance

In this project, the ability to efficiently visualize extensive datasets on inflation, food price indices, and energy price indices in real-time was paramount. Given the large scope of the dataset, spanning multiple decades (1970-2023) and up to 209 countries, maintaining high performance during analysis was essential. LightningChart Python was selected for its robust capabilities in handling such complex datasets seamlessly.

Several key factors contributed to its effectiveness:

* **Seamless Large Dataset Handling**: LightningChart Python excels at rendering large datasets without sacrificing speed or quality. This ensures that data involving millions of points, such as country-level inflation, food, and energy price indices, is processed and visualized with precision and efficiency.
* **Interactive and Real-Time Capabilities**: The tool allows users to interact with visualizations dynamically, enabling real-time exploration of inflationary trends. Whether zooming into specific years, countries, or regions or updating the charts with live data, LightningChart maintains high responsiveness, making it ideal for monitoring time-sensitive economic shifts.
* **Optimized Resource Utilization**: Despite the dataset's scale, which includes monthly and annual data across various indicators, LightningChart efficiently manages system resources. This optimization ensures smooth rendering of 3D balloon charts, radar charts, and dashboards, providing a lag-free and engaging user experience.

By leveraging LightningChart’s high-performance visualization capabilities, this analysis successfully uncovered intricate inflation and energy dynamics while maintaining exceptional usability and performance throughout the project.

**Setting Up Python Environment**

To begin working on this project, it's necessary to install the required libraries, including Pandas and LightningChart Python. You can do this by following the steps below:

First, you need to install Python: Download and install the latest version of Python from the [**official Python website.**](https://www.python.org/)

Second, you need to Install Pandas and LightningChart:

Use the pip package manager to install the necessary libraries by running these commands:

pip install lightningchart

pip install pandas numpy

To analyze and visualize the data, we’ll use a combination of libraries such as:

**Pandas:** For data manipulation and analysis.

**NumPy:** For numerical operations.

**LightningChart:** A high-performance charting library for data visualization.

**Time and Random:** For simulating real-time data updates and generating random values.

import pandas as pd

import lightningchart as lc

import time

import numpy as np

**Visualizing Data with LightningChart Python**

LightningChart offers a powerful way to visualize complex datasets in real-time. Below, we’ll create several visualizations to explore electricity access across the globe

**Energy Price Index Dashboard for Ukraine (2018-2023)**

**Trends Interpretation:**

* **2018-2020**: The energy price index shows a generally stable trend, with minor fluctuations across the months. The values stay within a narrow range, indicating consistency in energy prices during these years.
* **2021-2022**: A noticeable rise in the energy price index is observed, especially in 2022, where the values spike around mid-year, potentially reflecting external factors such as geopolitical or economic shifts.
* **2023**: The data highlights a sharp increase during the latter months of the year, showing a significant upward trend compared to previous years, possibly hinting at inflation or other crises.

**Script Summary**:

import lightningchart as lc

import pandas as pd

lc.set\_license(open(license-key').read().strip())

data = pd.read\_excel('dataset/Processed.xlsx', sheet\_name='ecpi\_m')

selected\_country\_data = data[data['Country'] == "Ukraine"].iloc[:, 5:].filter(regex='^2018|^2019|^2020|^2021|^2022|^2023')

data\_t = selected\_country\_data.T

data\_t.columns = ['Index Value']

data\_t['Year'] = data\_t.index.str[:4].astype(int)

data\_t['Month'] = data\_t.index.str[4:6].astype(int)

chart = lc.ChartXY(theme=lc.Themes.Black, title="Energy Price Index for Ukraine (2018-2023)")

chart.get\_default\_x\_axis().set\_title('Month')

chart.get\_default\_y\_axis().set\_title('Value')

legend = chart.add\_legend()

for year in range(2018, 2024):

    year\_data = data\_t[data\_t['Year'] == year]

    chart.add\_line\_series().set\_name(str(year)).add(list(year\_data['Month']), list(year\_data['Index Value']))

chart.open()



#### Significance of Results:

* **Economic Stability**: The relatively stable index in earlier years could point to a balanced energy market.
* **Impact of External Events**: The sharp rise in 2022 and 2023 suggests external disruptions such as global conflicts or policy changes impacting energy prices.
* **Policy Implications**: The upward trend emphasizes the need for energy price regulation and alternative energy sourcing to stabilize future trends.
* **Forecasting Needs**: Identifying periods of sharp increases could help design predictive models for mitigating economic shocks related to energy prices.

### Food Price Index Dashboard for Ukraine (2018-2023)

#### Trends Interpretation:

As previously noted for the Energy Price Index, a stable trend is observed from **2018 to 2020**, followed by sharp increases in **2021 and 2022**, indicating external disruptions. However, in this case, the food price index specifically highlights:

* **2022**: A peak mid-year, suggesting heightened supply chain issues or inflationary pressures.
* **2023**: While growth continues, the increase is less steep compared to 2022, hinting at a possible stabilization of food prices.

import lightningchart as lc

import pandas as pd

lc.set\_license(open('license-key').read().strip())

data = pd.read\_excel('dataset/Processed.xlsx', sheet\_name='fcpi\_m')

selected\_country\_data = data[data['Country'] == "Ukraine"].iloc[:, 5:].filter(regex='^2018|^2019|^2020|^2021|^2022|^2023')

data\_t = selected\_country\_data.T

data\_t.columns = ['Index Value']

data\_t['Year'] = data\_t.index.str[:4].astype(int)

data\_t['Month'] = data\_t.index.str[4:6].astype(int)

chart = lc.ChartXY(theme=lc.Themes.Black, title="Food Price Index for Ukraine (2018-2023)")

chart.get\_default\_x\_axis().set\_title('Month')

chart.get\_default\_y\_axis().set\_title('Value')

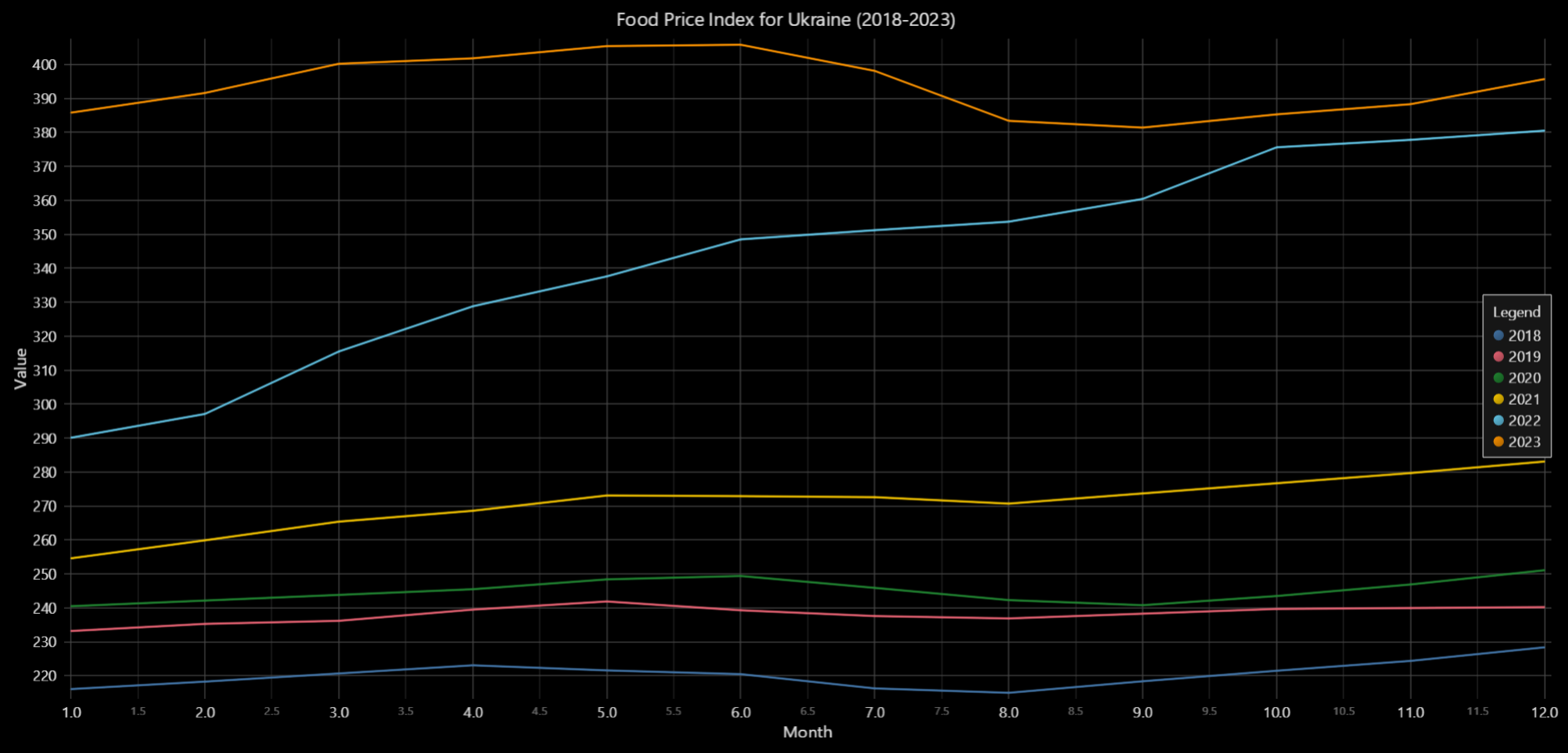
legend = chart.add\_legend()

for year in range(2018, 2024):

    year\_data = data\_t[data\_t['Year'] == year]

    chart.add\_line\_series().set\_name(str(year)).add(list(year\_data['Month']), list(year\_data['Index Value']))

chart.open()



#### Significance of Results:

As with energy prices, the significant rise in food prices reflects broader economic challenges. The 2022 peak underscores vulnerabilities in food supply chains, emphasizing the need for improved resilience planning. Unlike the energy index, the food price index shows slightly more stabilization in 2023, offering a potential opportunity for recovery strategies in agriculture and food production.

### Food Price Index Heatmap for Albania (2000-2023)

#### Trends Interpretation:

* **Overall Visualization**: The heatmap illustrates the monthly variation in the food price index across the years from 2000 to 2023. The vertical axis represents **months (1-12)**, while the horizontal axis represents **years (2000-2023)**. Blue represents lower index values, and red represents higher index values.
* **Key Observations**:
  + **Lower Intensities (Blue)**: Dominant in the early years (2000-2010), indicating relatively stable or lower food prices across all months.
  + **Increasing Intensities (Red)**: From around **2010 onwards**, there is a clear shift toward higher intensities, with more red visible in the heatmap, particularly at the **top (winter months)**. This suggests that food prices have been increasing steadily over the years, especially during the winter months.
  + **Seasonal Patterns**: Higher intensity (red) is more pronounced in the **winter months (11-12)**, indicating that food prices tend to peak during the end of each year.

import pandas as pd

import numpy as np

import lightningchart as lc

lc.set\_license(open('license-key').read().strip())

# Load data and filter for Albania

data = pd.read\_excel('dataset/Processed.xlsx', sheet\_name='fcpi\_m')

selected\_country\_data = data[data['Country'] == "Albania"].iloc[:, 5:].filter(regex='^2000|^2023')

# Transform data for heatmap

data\_t = selected\_country\_data.T

data\_t.columns = ['Index Value']

data\_t['Year'] = data\_t.index.str[:4].astype(int)

data\_t['Month'] = data\_t.index.str[4:6].astype(int)

heatmap\_data = data\_t.pivot(index="Month", columns="Year", values="Index Value").fillna(0).values

years\_ms = [int(pd.Timestamp(year=year, month=1, day=1).timestamp() \* 1000) for year in data\_t['Year'].unique()]

# Create heatmap

chart = lc.ChartXY(title="Food Price Index Heatmap for Albania (2000-2023)", theme=lc.Themes.Dark)

heatmap\_series = chart.add\_heatmap\_grid\_series(columns=heatmap\_data.shape[0], rows=heatmap\_data.shape[1])

heatmap\_series.set\_start(x=years\_ms[0], y=1).set\_end(x=years\_ms[-1], y=12)

heatmap\_series.invalidate\_intensity\_values(heatmap\_data.tolist())

heatmap\_series.set\_palette\_coloring([

    {"value": np.min(heatmap\_data), "color": lc.Color(0, 0, 255)},

    {"value": np.max(heatmap\_data), "color": lc.Color(255, 0, 0)}

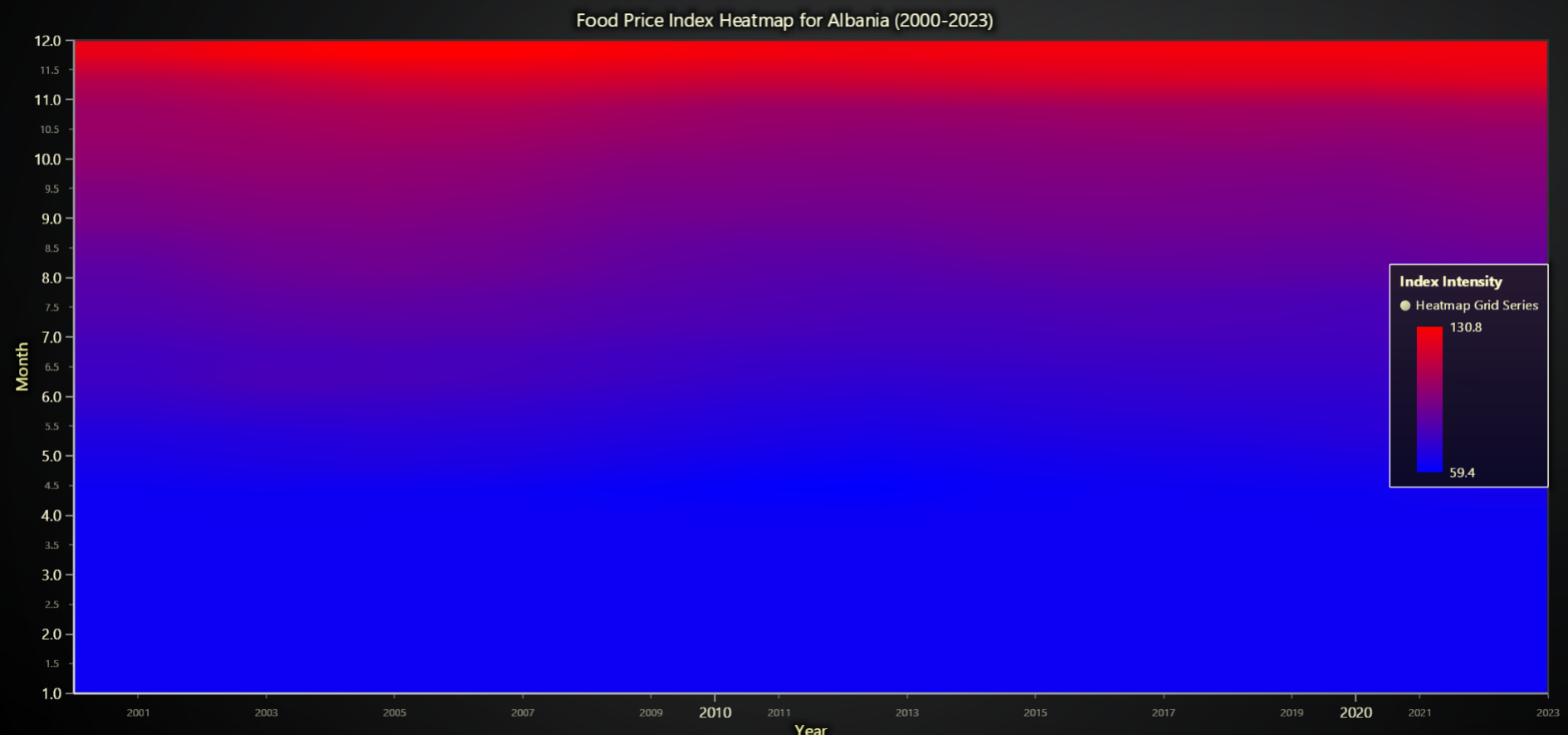
])

chart.get\_default\_x\_axis().set\_title('Year').set\_tick\_strategy('DateTime', utc=True)

chart.get\_default\_y\_axis().set\_title('Month').set\_interval(1, 12, 1)

chart.add\_legend(data=heatmap\_series).set\_title('Index Intensity')

chart.open()



#### Significance of Results:

* **Economic Trends**: The increase in intensity over time reflects **long-term inflation** in the food market in Albania. The peak during the winter months suggests potential supply chain constraints or seasonal demand spikes driving up prices.
* **Policy Insights**: Authorities could focus on addressing price stability, particularly during the winter months, to ensure food affordability for consumers.
* **Strategic Planning**: This heatmap highlights the need for targeted **seasonal interventions** and strategies for improving food supply resilience during periods of increased prices.

### Energy Price Index Heatmap for Albania (2000-2023)

#### Trends Interpretation:

* **Horizontal Axis (Years)**: The heatmap shows the variation in energy price index across the years 2000 to 2023.
* **Vertical Axis (Months)**: Monthly variations are depicted, from January (1) to December (12).
* **Key Observations**:
  + **Early Years (2000-2010)**: The lower part of the heatmap is predominantly blue, indicating relatively low energy price intensity across all months during these years.
  + **Post-2010**: A gradual shift toward red intensity, with a visible increase in the energy price index, particularly in the upper months (October to December).
  + **Recent Years (2020-2023)**: The intensity becomes predominantly red, signifying a sharp rise in energy prices across most months, especially during the colder months (late in the year).

import pandas as pd

import numpy as np

import lightningchart as lc

lc.set\_license(open('license-key').read().strip())

data = pd.read\_excel('dataset/Processed.xlsx', sheet\_name='ecpi\_m')

selected\_country\_data = data[data['Country'] == "Albania"].iloc[:, 5:].filter(regex='^2000|^2023')

data\_t = selected\_country\_data.T

data\_t.columns = ['Index Value']

data\_t['Year'] = data\_t.index.str[:4].astype(int)

data\_t['Month'] = data\_t.index.str[4:6].astype(int)

heatmap\_data = data\_t.pivot(index="Month", columns="Year", values="Index Value").fillna(0).values

years\_ms = [

    int(pd.Timestamp(year=year, month=1, day=1).timestamp() \* 1000)

    for year in data\_t['Year'].unique()

]

chart = lc.ChartXY(title="Energy Price Index Heatmap for Albania (2000-2023)", theme=lc.Themes.Dark)

heatmap\_series = chart.add\_heatmap\_grid\_series(columns=heatmap\_data.shape[1], rows=heatmap\_data.shape[0])

heatmap\_series.set\_start(x=years\_ms[0], y=1).set\_end(x=years\_ms[-1], y=12).set\_step(x=(years\_ms[-1] - years\_ms[0]) / heatmap\_data.shape[1], y=1)

heatmap\_series.set\_intensity\_interpolation(True).invalidate\_intensity\_values(heatmap\_data.tolist()).hide\_wireframe()

heatmap\_series.set\_palette\_coloring([

    {"value": np.min(heatmap\_data), "color": lc.Color(0, 0, 255)},

    {"value": np.max(heatmap\_data), "color": lc.Color(255, 0, 0)}

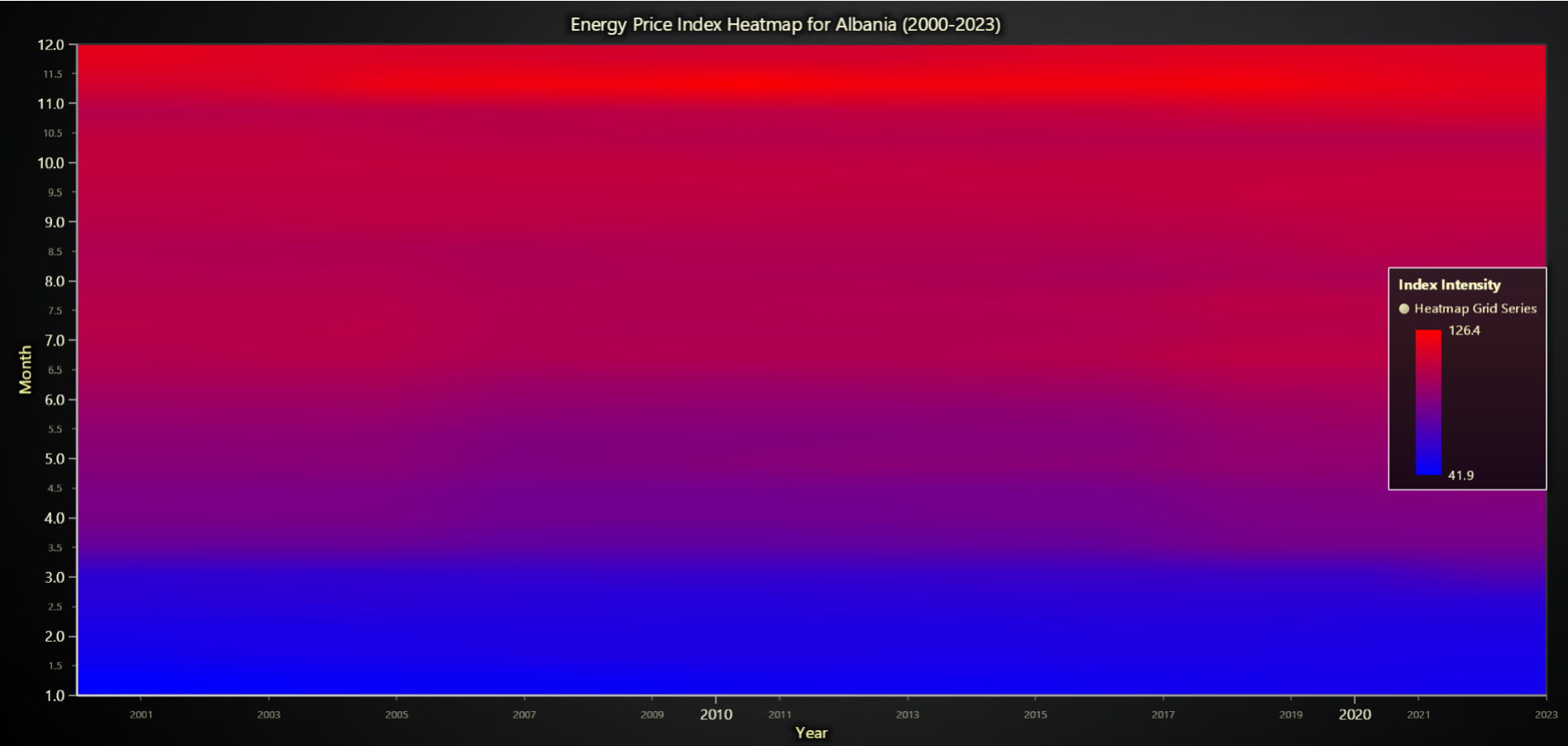
])

chart.get\_default\_x\_axis().set\_title('Year').set\_tick\_strategy('DateTime', utc=True)

chart.get\_default\_y\_axis().set\_title('Month').set\_interval(1, 12, 1)

chart.add\_legend(data=heatmap\_series).set\_title('Index Intensity')

chart.open()

****

#### Significance of Results:

* **Seasonal Patterns**: Higher intensities toward the upper part of the heatmap suggest that energy prices typically peak in the winter months, likely due to increased heating demands.
* **Economic Impact**: The sustained rise in energy prices post-2010 reflects the growing costs of energy resources, potentially driven by geopolitical events or supply-demand imbalances.
* **Policy Recommendations**: Addressing seasonal spikes in energy prices could help stabilize household and industrial energy costs. Initiatives such as energy subsidies or alternative energy sources may mitigate these price pressures.

### Real-Time Dashboard: Energy Price Index Growth for Continents (2000-2023)

#### Overview:

The dashboard displays real-time **Energy Price Index Growth (%)** across six continents (Africa, Asia, Europe, North America, Oceania, and South America) for each year from 2000 to 2023. Each gauge dynamically updates to show the percentage growth for a specific continent, categorized into four intervals:

* Blue: Significant negative growth (-50% to -10%)
* Orange: Moderate negative growth (-10% to 0%)
* Yellow: Moderate positive growth (0% to 10%)
* Red: Significant positive growth (10% to 50%).

#### Observations:

* **Africa**: Exhibits the highest volatility, with sharp growth spikes in 2022 (80.6%) and 2023 (493.2%).
* **Asia**: Shows steady growth with major spikes in 2021 (8.8%) and 2022 (32.6%).
* **Europe**: Experiences moderate growth, peaking at 20.8% in 2022 but showing minor growth in 2023 (5.7%).
* **North America**: Demonstrates stable growth, peaking modestly in 2021 (6.4%) and 2022 (9.6%).
* **Oceania**: Maintains smaller fluctuations, with moderate growth in 2022 (8.2%) but lower intensity in most years.
* **South America**: Shows consistently positive growth with peaks in 2021 (14%) and 2022 (23.2%).

#### Key Features:

* **Real-Time Updates**: The dashboard dynamically updates each second to display growth for the next year, creating an engaging and interactive visualization.
* **Color-Coded Indicators**: The intervals provide quick visual cues about growth severity for each continent.

pandas as pd

import lightningchart as lc

import time

from pycountry\_convert import country\_name\_to\_country\_alpha2, country\_alpha2\_to\_continent\_code

lc.set\_license(open('license-key').read().strip())

data = pd.read\_excel('dataset/Processed.xlsx', sheet\_name='ecpi\_m')

# Add continent information to the dataset

def get\_continent(country\_name):

    try:

        code = country\_name\_to\_country\_alpha2(country\_name)

        return {'AF': 'Africa', 'AS': 'Asia', 'EU': 'Europe',

                'NA': 'North America', 'SA': 'South America', 'OC': 'Oceania'}.get(

            country\_alpha2\_to\_continent\_code(code), 'Unknown')

    except:

        return 'Unknown'

data['Continent'] = data['Country'].apply(get\_continent)

data = data[data['Continent'] != 'Unknown']

# Calculate growth percentage

def calculate\_growth(year, prev\_year):

    try:

        current = data.filter(like=str(year)).groupby('Continent').mean().mean(axis=1)

        previous = data.filter(like=str(prev\_year)).groupby('Continent').mean().mean(axis=1)

        return ((current - previous) / previous) \* 100

    except:

        return None

# Create a dashboard with gauges for each continent

dashboard = lc.Dashboard(theme=lc.Themes.TurquoiseHexagon, rows=2, columns=3)

gauges = {c: dashboard.GaugeChart(row\_index=i // 3, column\_index=i % 3)

          .set\_title(f"{c}: Growth (%)").set\_angle\_interval(225, -45)

          .set\_interval(-50, 50).set\_bar\_thickness(20).set\_value\_indicator\_thickness(10)

          .set\_value\_indicators([

              {'start': -50, 'end': -10, 'color': lc.Color('blue')},

              {'start': -10, 'end': 0, 'color': lc.Color('orange')},

              {'start': 0, 'end': 10, 'color': lc.Color('yellow')},

              {'start': 10, 'end': 50, 'color': lc.Color('red')}])

          for i, c in enumerate(['Africa', 'Asia', 'Europe', 'North America', 'Oceania', 'South America'])}

# Update gauges dynamically for each year

dashboard.open(live=True)

for year in range(2000, 2024):

    growth = calculate\_growth(year, year - 1)

    if growth is not None:

        for continent, percentage in growth.items():

            if continent in gauges:

                gauges[continent].set\_value(percentage).set\_title(f"{continent}: Growth ({year}) (%)")

    time.sleep(1)



#### Significance of Results:

* **Africa's Volatility**: The sharp spikes in 2022 and 2023 suggest external economic shocks or resource demand surges.
* **Asia's Steady Rise**: Indicates sustained economic development, with growing energy demands.
* **Europe's Stability**: Reflects moderate resilience, though slightly impacted in 2022.
* **Strategic Planning**: The dashboard provides a clear visualization tool for policymakers and researchers to monitor and respond to trends in energy price growth across continents.

### Bar Chart: Yearly Comparison of FCPI and ECPI for Turkey (2018-2023)

#### Trends and Observations:

* **Prominent Peak in 2023**:
* For **Turkey**, the **Food Consumer Price Index (FCPI)** reached a record high of **732**, while the **Energy Consumer Price Index (ECPI)** was significantly lower at **491**.
* **Gradual Decline Over Time**:
* Starting from **2023** (left side of the chart) to **2000** (right side), both indices display a **progressive decrease** year by year.
* The gap between FCPI and ECPI remains evident for most years, especially in recent periods, but narrows as values drop below **100** in earlier years.
* **Stable Values from 2000-2015**:
* During this period, indices for Turkey remained low and steady, fluctuating between **10-50**, showing relative stability in food and energy prices compared to recent years.

import lightningchart as lc

import pandas as pd

lc.set\_license(open('license-key').read().strip())

file\_path = 'dataset/Processed.xlsx'

fcpi\_data = pd.read\_excel(file\_path, sheet\_name='fcpi\_m')

ecpi\_data = pd.read\_excel(file\_path, sheet\_name='ecpi\_m')

country = "Turkey"

if country in fcpi\_data['Country'].values and country in ecpi\_data['Country'].values:

    def calculate\_yearly\_averages(data, start\_year=2000, end\_year=2023):

        return {str(year): data.filter(like=str(year)).mean(axis=1).values[0]

                for year in range(start\_year, end\_year + 1) if str(year) in data}

    fcpi\_yearly\_avg = calculate\_yearly\_averages(fcpi\_data[fcpi\_data['Country'] == country])

    ecpi\_yearly\_avg = calculate\_yearly\_averages(ecpi\_data[ecpi\_data['Country'] == country])

    valid\_years = sorted(fcpi\_yearly\_avg.keys() & ecpi\_yearly\_avg.keys())

    data = [

        {'subCategory': 'FCPI', 'values': [fcpi\_yearly\_avg[year] for year in valid\_years]},

        {'subCategory': 'ECPI', 'values': [ecpi\_yearly\_avg[year] for year in valid\_years]},

    ]

    chart = lc.BarChart(

        vertical=True, theme=lc.Themes.Light,

        title=f'Yearly Comparison of FCPI and ECPI for {country} (2018-2023)'

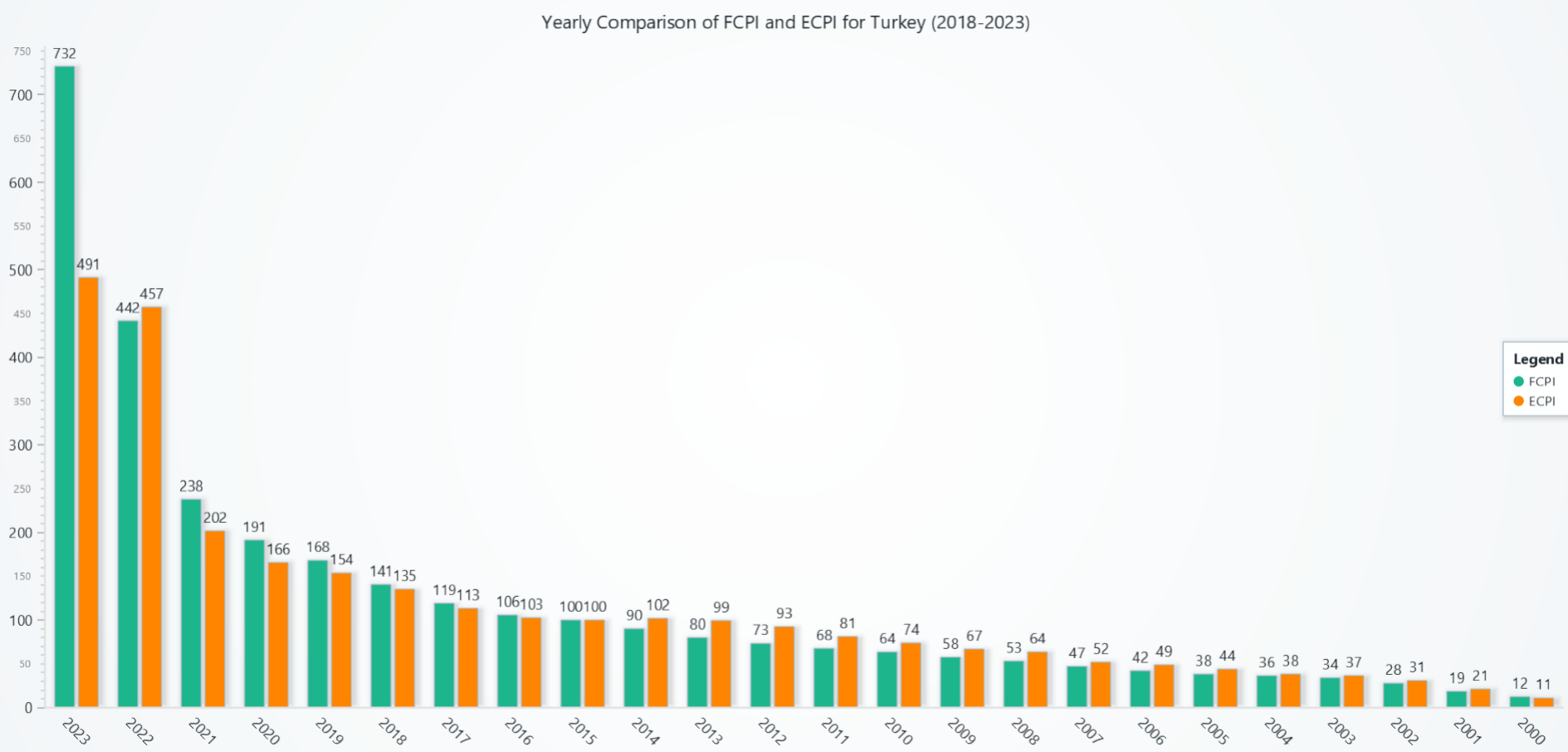
    )

    chart.set\_data\_grouped(valid\_years, data)

    chart.set\_label\_rotation(45).set\_value\_label\_font\_size(14).set\_bars\_margin(0.2)

    chart.add\_legend().add(chart)

    chart.open()

****

#### Insights:

* **Economic Trends in Turkey**:
* The **2023 and 2022** may be attributed to inflationary pressures or specific economic challenges impacting Turkey's food and energy markets disproportionately.
* The decline toward **2000** reflects historical stability in Turkey's consumer price indices before recent economic turbulence.
* **Divergence in 2023**:
* The substantial gap in 2023 suggests that food prices in Turkey are experiencing sharper inflationary pressures compared to energy prices, highlighting potential sector-specific challenges.

#### Radar Chart: Inflation and Price Indices for Turkey (2015-2023)

#### Trends and Observations:

* **FCPI (Food Consumer Price Index)**:
  + **High Values in 2023**: As mentioned earlier, FCPI experienced a sharp increase in 2023, highlighting significant inflationary pressures in the food sector for Turkey.
  + **Stability Before 2021**: Similar to observations in the bar chart, FCPI remained relatively stable between 2015 and 2020.
* **ECPI (Energy Consumer Price Index)**:
  + **Gradual Increase**: ECPI followed a consistent upward trend, as also seen in the bar chart, reflecting rising energy prices over time.
  + **Lower Than FCPI**: As mentioned previously, ECPI consistently remained below FCPI across all years, highlighting a slower inflationary trend in energy compared to food.
* **DEF\_A (GDP Deflator Growth Rate)**:
  + **Moderate and Stable**: DEF\_A values showed consistent moderate growth, indicating overall economic stability despite fluctuations in FCPI and ECPI.

import lightningchart as lc

import pandas as pd

lc.set\_license(open('license-key').read().strip())

# Load datasets

file\_path = 'dataset/Processed.xlsx'

fcpi\_data = pd.read\_excel(file\_path, sheet\_name='fcpi\_m')

ecpi\_data = pd.read\_excel(file\_path, sheet\_name='ecpi\_m')

def\_a\_data = pd.read\_excel(file\_path, sheet\_name='def\_a')

# Filter for Turkey and calculate yearly averages

country = "Turkey"

if all(country in data['Country'].values for data in [fcpi\_data, ecpi\_data, def\_a\_data]):

    years = [str(year) for year in range(2015, 2024)]

    def calculate\_yearly\_averages(data, years):

        return [

            data.filter(like=year).mean(axis=1).values[0] if year in data else None

            for year in years

        ]

    fcpi\_avg = calculate\_yearly\_averages(fcpi\_data[fcpi\_data['Country'] == country], years)

    ecpi\_avg = calculate\_yearly\_averages(ecpi\_data[ecpi\_data['Country'] == country], years)

    def\_a\_values = [

        def\_a\_data[def\_a\_data['Country'] == country].get(year, pd.Series([None])).values[0]

        for year in map(int, years)

    ]

    # Create radar chart

    chart = lc.SpiderChart(theme=lc.Themes.TurquoiseHexagon, title=f'Radar Chart: FCPI, ECPI, and DEF\_A for {country} (2015-2023)')

    for year in years:

        chart.add\_axis(tag=year)

    chart.add\_series(name='FCPI', values=fcpi\_avg, line\_color=lc.Color(31, 119, 180), fill\_color=lc.Color(31, 119, 180, 100))

    chart.add\_series(name='ECPI', values=ecpi\_avg, line\_color=lc.Color(255, 127, 14), fill\_color=lc.Color(255, 127, 14, 100))

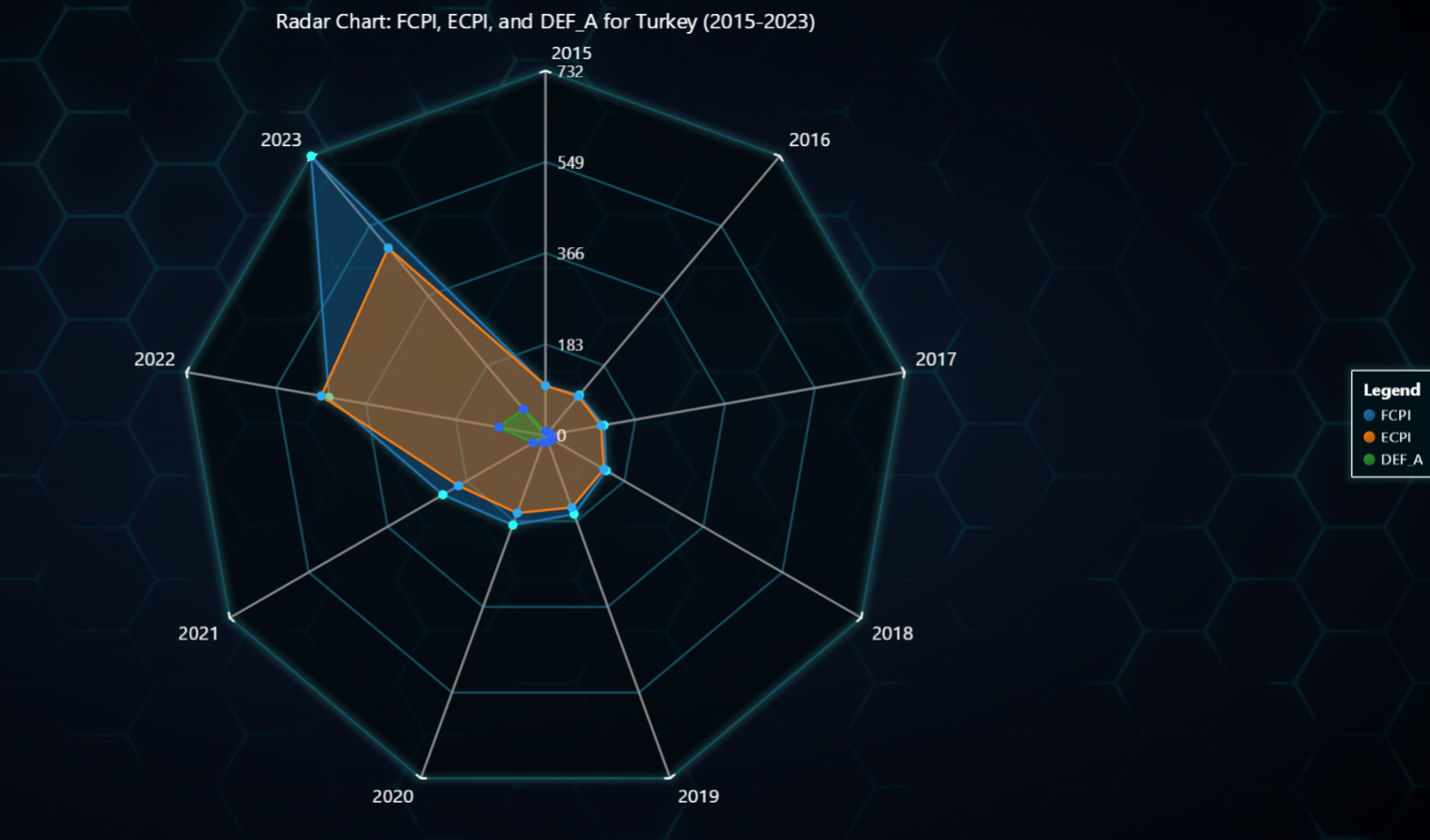
    chart.add\_series(name='DEF\_A', values=def\_a\_values, line\_color=lc.Color(44, 160, 44), fill\_color=lc.Color(44, 160, 44, 100))

    chart.add\_legend()

    chart.open()

else:

    print(f"{country} does not exist in all datasets.")



#### Key Insights:

1. **Economic Pressures**:
   * As previously noted, the sharp increase in FCPI and ECPI in **2023** underscores Turkey's vulnerability to inflation, particularly in food and energy sectors.
2. **Policy Implications**:
   * Similar to the bar chart insights, targeted interventions in the food sector are essential to address the growing inflationary challenges reflected by the high FCPI values in 2023.
3. **Resilient GDP**:
   * As observed earlier, the stable DEF\_A values suggest that Turkey's economic growth remained resilient, even during inflationary spikes.

#### Chart Highlights:

**Complementary Insights**:

This radar chart reinforces the trends seen in the bar chart but offers a comparative multi-index perspective, emphasizing the interrelation between food prices, energy prices, and GDP growth.

#### Balloon Race Chart: GDP Deflator Growth Rate and Energy Index for Turkey, Iran, and Egypt (2000-2023):

* The **Balloon Race Chart** is a 3D visualization of **GDP Deflator Growth Rate (balloon height)** and **Energy Index (balloon brightness)** for the countries **Turkey**, **Iran, Islamic Rep.**, and **Egypt, Arab Rep.** from **2000 to 2023**.
* **X-Axis**: Represents the countries.
* **Y-Axis**: Shows normalized GDP Deflator Growth Rate (inflation) between 0 and 2.
* **Z-Axis**: Represents the timeline from 2000 to 2023.

#### Trends and Observations:

1. **Turkey** (Blue Balloon):
   * **Significant Inflation Growth**: The balloon height for Turkey rises sharply in recent years, indicating substantial growth in GDP deflator (inflation).
   * **Moderate Energy Brightness**: The balloon's brightness, corresponding to energy index, is moderate but shows an upward trend in recent years.
2. **Iran, Islamic Rep.** (Green Balloon):
   * **Unstable Inflation Growth**: Iran exhibits significant fluctuations in balloon height, indicating volatile inflation rates. However, there is an overall upward trend.
   * **Sharp Energy Index Increase**: The brightness of Iran’s balloon increases significantly in recent years, showing a surge in energy index values.
3. **Egypt, Arab Rep.** (Red Balloon):
   * **Stable and Low Inflation**: The balloon for Egypt remains consistently lower in height compared to Turkey and Iran, reflecting a relatively stable and lower inflation rate.
   * **Gradual Increase in Energy Brightness**: The energy index for Egypt increases steadily over the years, leading to higher balloon brightness.

import lightningchart as lc

import pandas as pd

import trimesh

import asyncio

lc.set\_license(open('license-key').read().strip())

file\_path = 'dataset/Processed.xlsx'

def\_a\_data = pd.read\_excel(file\_path, sheet\_name='def\_a')

ecpi\_data = pd.read\_excel(file\_path, sheet\_name='ecpi\_m')

countries = ["Turkey", "Iran, Islamic Rep.", "Egypt, Arab Rep."]

years = [str(year) for year in range(2000, 2024)]

# Normalize inflation and energy data

def normalize(data, min\_val, max\_val, scale=1):

    return [(value - min\_val) / (max\_val - min\_val) \* scale for value in data]

inflation\_data = {

    country: normalize(

        def\_a\_data.loc[def\_a\_data['Country'] == country].filter(regex='|'.join(years)).iloc[0].tolist(),

        min\_val=def\_a\_data.filter(regex='|'.join(years)).min().min(),

        max\_val=def\_a\_data.filter(regex='|'.join(years)).max().max(),

        scale=2

    )

    for country in countries

}

energy\_data = {

    country: normalize(

        ecpi\_data.loc[ecpi\_data['Country'] == country].filter(regex='|'.join(years)).mean(axis=1).tolist(),

        min\_val=ecpi\_data.filter(regex='|'.join(years)).min().min(),

        max\_val=ecpi\_data.filter(regex='|'.join(years)).max().max()

    )

    for country in countries

}

balloon\_colors = {"Turkey": (0, 0, 255), "Iran, Islamic Rep.": (0, 255, 0), "Egypt, Arab Rep.": (255, 0, 0)}

# Chart setup

chart = lc.Chart3D(title="Balloon Race: Normalized GDP Deflator Growth Rate & Energy", theme=lc.Themes.Light)

chart.get\_default\_x\_axis().set\_title('Countries').set\_interval(-0.5, 4.5, stop\_axis\_after=True)

chart.get\_default\_y\_axis().set\_title('Height (Inflation)').set\_interval(0, 2, stop\_axis\_after=True)

chart.get\_default\_z\_axis().set\_title('Time (Years)')

# Balloons and lines

balloons = {}

line\_series = {}

for i, country in enumerate(countries):

    x\_position = 0.75 + i \* 1.5

    balloons[country] = chart.add\_mesh\_model().set\_model\_location(x\_position, 0, 0).set\_scale(0.005)

    line\_series[country] = chart.add\_line\_series().set\_name(f"{country} Path").set\_line\_color(lc.Color(\*balloon\_colors[country]))

async def move\_balloons():

    for year\_idx in range(len(years) - 1):

        for step in range(50):

            for i, country in enumerate(countries):

                x\_pos = 0.75 + i \* 1.5

                h\_start, h\_end = inflation\_data[country][year\_idx:year\_idx + 2]

                b\_start, b\_end = energy\_data[country][year\_idx:year\_idx + 2]

                height = h\_start + (h\_end - h\_start) \* (step / 50)

                brightness = b\_start + (b\_end - b\_start) \* (step / 50)

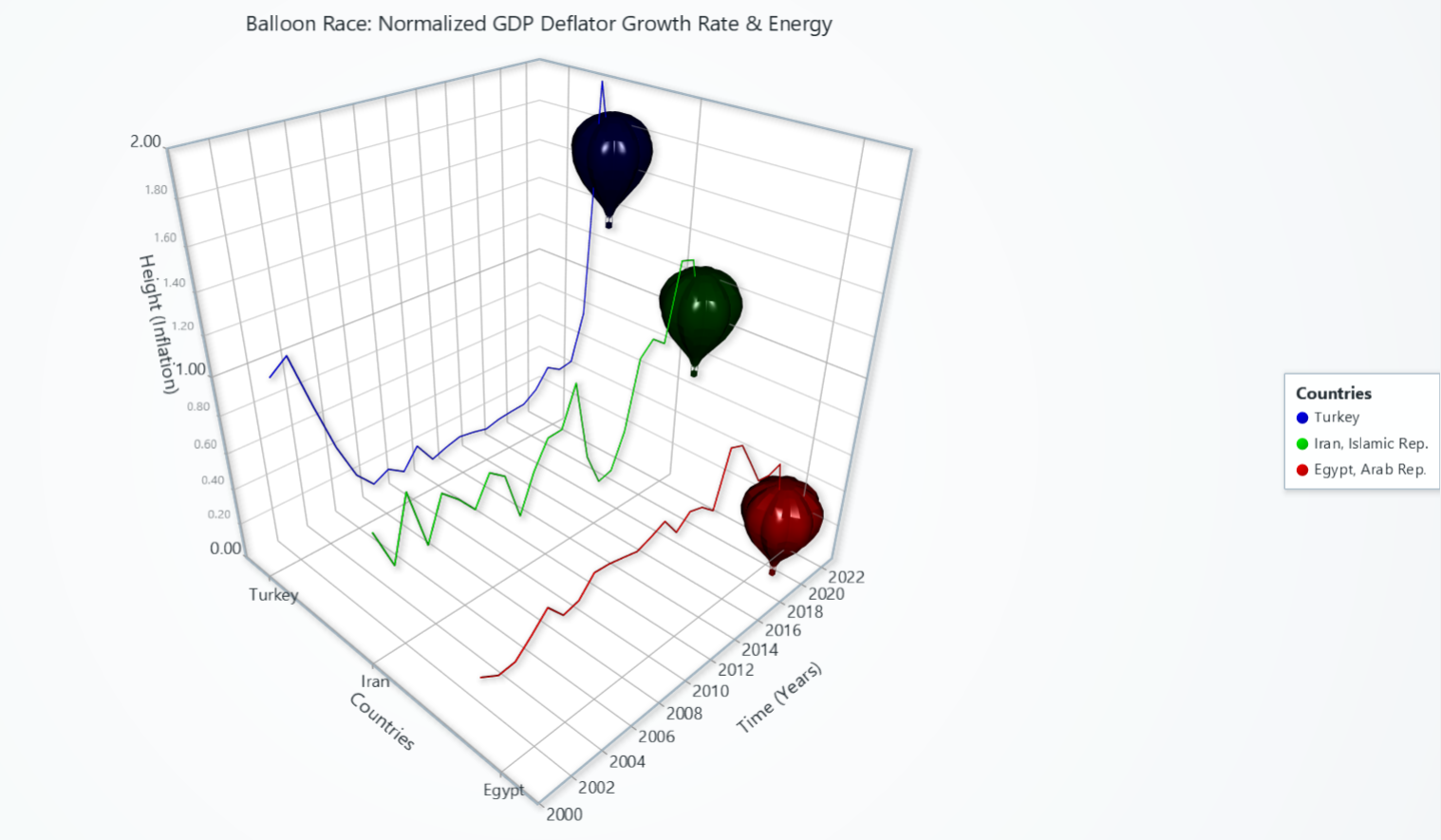
                balloons[country].set\_model\_location(x\_pos, height, year\_idx + (step / 50))

                line\_series[country].add([x\_pos], [height], [year\_idx + (step / 50)])

            await asyncio.sleep(0.02)

chart.open(live=True)

asyncio.run(move\_balloons())



#### Key Insights:

* **Real-Time Representation**: The chart demonstrates dynamic balloon movements in real time, visually depicting the changes in inflation (height) and energy (brightness) over the years.
* **Comparative View**: This chart allows for a direct comparison of economic trends between the three countries.
* **Economic Implications**:
  + **Turkey and Iran**: Both countries show significant inflation growth, with Turkey having a more consistent upward trajectory.
  + **Egypt**: While relatively stable in inflation, Egypt shows a gradual increase in energy consumption, reflected in the brightness.

#### Chart Highlights:

* **Visual Dynamics**: The live motion of the balloons and their respective paths make it easy to identify trends and differences across countries.
* **Complementary Insights**: The chart provides an engaging and intuitive way to analyze inflation and energy trends, supporting a broader understanding of economic conditions in Turkey, Iran, and Egypt.

#### ****Conclusion****

This study highlights the interconnected dynamics of inflation, food, and energy price indices across regions and countries. Visualizations created using LightningChart provide a comprehensive and intuitive understanding of these trends. Policymakers can leverage these insights to address inflationary pressures, ensure food and energy security, and stabilize economic growth. For instance, interventions targeting food inflation in Turkey and energy price volatility in Iran could mitigate broader economic risks.

By utilizing state-of-the-art visualization tools, this research bridges the gap between complex economic data and actionable insights, setting a benchmark for future analyses in macroeconomic trends and policy evaluation.

#### ****References****

* Burke, M. J., & Stephens, J. C. (2018). Political power and renewable energy futures: A critical review. Energy Research & Social Science, 35, 78-93.
* Sovacool, B. K. (2010). A critical evaluation of nuclear power and renewable electricity in Asia. Journal of Cleaner Production, 18(3), 326-334.
* Stern, D. I., & Burke, P. J. (2019). The impact of electricity on economic development: A reassessment. Energy Economics, 83, 65-74.