

# Fiscal Risk in Europe: Clustering, Resilience, and Forecasting

Fiscal Analytics Project

2026-02-20

## Table of contents

<b>1 Abstract</b>	<b>2</b>
<b>2 Research Questions</b>	<b>2</b>
<b>3 Setup</b>	<b>3</b>
<b>4 Load Eurostat TSV</b>	<b>3</b>
<b>5 Tidy the Dataset into a Panel</b>	<b>4</b>
<b>6 Build Core Fiscal Metrics</b>	<b>5</b>
<b>7 Clustering Countries by Fiscal Behavior</b>	<b>6</b>
7.1 Country-level Summary (Dominant Cluster + Stability Metrics) . . . . .	7
<b>8 Fiscal Shock Resilience Index</b>	<b>8</b>
<b>9 Composite Fiscal Risk Score</b>	<b>9</b>
<b>10 Europe Choropleth Map</b>	<b>9</b>
<b>11 Executive Bubble Chart</b>	<b>11</b>
<b>12 Time-Series Forecasting (Germany — Demo)</b>	<b>13</b>
12.1 ARIMA Forecast for Debt-to-GDP . . . . .	14
12.2 VAR Model for Joint Dynamics . . . . .	16
<b>13 Summary Table and Final Risk Ranking</b>	<b>18</b>
<b>14 Conclusions</b>	<b>20</b>
14.1 What We Derived . . . . .	20
14.2 What It Means . . . . .	21
14.3 Future Implications and Directions . . . . .	22
14.4 Final Verdict . . . . .	22

## 1 Abstract

This report develops an interpretable, end-to-end **fiscal risk framework** for European countries using official [Eurostat](#) government finance data spanning 1995–2024. We combine unsupervised machine learning (KMeans clustering), a shock resilience index calibrated against the Global Financial Crisis and COVID-19, a composite fiscal risk score, and time-series forecasting (ARIMA, VAR) into a unified analytical pipeline. The framework reveals a persistent and widening north-south fiscal divide across Europe, with significant implications for monetary union stability, EU fiscal governance, and public investment capacity in the decade ahead.

---

## 2 Research Questions

This project addresses five core questions about European fiscal dynamics:

**RQ1 — Can European countries be meaningfully grouped by fiscal behavior?**

Using Debt-to-GDP, Deficit-to-GDP, and debt growth rates over three decades, do natural clusters emerge that separate fiscally stable countries from high-risk ones — and are these clusters geographically or institutionally coherent?

**RQ2 — How resilient are European governments to major fiscal shocks?**

Did the Global Financial Crisis (2008–2009) and the COVID-19 pandemic (2020–2021) affect all European countries equally, or do structural differences (initial debt levels, fiscal space, institutional frameworks) explain divergent deficit responses?

**RQ3 — Can we build a composite fiscal risk score that integrates multiple dimensions of vulnerability?**

Does combining long-run debt levels, deficit volatility, and crisis-era behavior into a single normalized score produce a ranking consistent with market perceptions and macroeconomic theory?

**RQ4 — What does the geographic distribution of fiscal risk reveal about European economic integration?**

Is fiscal risk evenly distributed across Europe, or do persistent north–south and east–west divides emerge with implications for EMU stability and ECB policy?

**RQ5 — Can time-series models forecast near-term fiscal trajectories?**

How well do ARIMA and VAR approaches capture debt dynamics, and what do forecast paths suggest about medium-term fiscal sustainability?

---

### 3 Setup

```
# Core
import pandas as pd
import numpy as np
import datetime, json, os, itertools

# Modeling
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score, mean_squared_error
from sklearn.decomposition import PCA
from scipy.stats import spearmanr

# Time series
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.api import VAR

# Visualization
import matplotlib.pyplot as plt
import plotly.express as px
```

---

### 4 Load Eurostat TSV

```
df_raw = pd.read_csv(
    "https://raw.githubusercontent.com/indrasaideva/Fiscal-Risk-in-Europe-Clustering-Resilience/master/Eurostat%20Data%20-%20MIO%20EUR%20S1%20B1GQ%20Time%20Series.csv",
    sep="\t"
)
print("Shape:", df_raw.shape)
print("First columns:", list(df_raw.columns[:5]))
df_raw.head(5)

Shape: (2103, 31)
First columns: ['freq', 'unit', 'sector', 'na_item', 'geo\\TIME_PERIOD', '1995 ', '1996 ', '1997 ', '1998 ', '1999 ', '2000 '
```

	freq,unit,sector,na_item,geo\TIME_PERIOD	1995	1996	1997	1998	1999	2000
0	A,MIO_EUR,S1,B1GQ,AT	183629.2	185944.9	186913.0	193947.1	203064.9	212400.0
1	A,MIO_EUR,S1,B1GQ,BE	220251.5	219965.3	223032.7	231015.7	242307.6	256370.0
2	A,MIO_EUR,S1,B1GQ,BG	14512.8	9829.8	10064.7	13437.6	12804.4	144400.0
3	A,MIO_EUR,S1,B1GQ,CY	7596.0	7890.1	8414.3	9152.6	9839.7	108040.0
4	A,MIO_EUR,S1,B1GQ,CZ	46333.6	53416.0	55174.6	60164.9	61470.4	674090.0

---

## 5 Tidy the Dataset into a Panel

Eurostat stores multiple dimensions in the first column. We split those dimensions, keep the **government sector**, and reshape years into rows.

```
def tidy_eurostat_gov_tsv(df: pd.DataFrame) -> pd.DataFrame:
    """
    Tidy Eurostat gov finance TSV into long format, with strict filtering to avoid
    artificial duplication across unit/sector combinations.
    Returns long format with columns: country, year, na_item, unit, sector, value
    """
    first_col = df.columns[0]
    dims = df[first_col].astype(str).str.split(", ", expand=True)
    dims.columns = ["freq", "unit", "sector", "na_item", "geo"]
    out = df.drop(columns=[first_col]).copy()
    out = pd.concat([dims, out], axis=1)
    out["geo"] = out["geo"].str.replace(r"\TIME_PERIOD", "", regex=True).str.strip()
    out = out[out["freq"].eq("A")].copy()

    gdp_mask = (out["na_item"].eq("B1GQ") & out["unit"].eq("MIO_EUR") & out["sector"].eq("S1"))
    gov_mask = (out["na_item"].isin(["GD", "B9"]) & out["unit"].eq("MIO_EUR") & out["sector"].eq("S1"))
    out = out[gdp_mask | gov_mask].copy()
    out = out[out["geo"].str.fullmatch(r"[A-Z]{2}")] .copy()

    year_cols = [c for c in out.columns if str(c).strip().isdigit()]
    long = out.melt(
        id_vars=["geo", "na_item", "unit", "sector"],
        value_vars=year_cols,
        var_name="year",
        value_name="value"
    )
    long["year"] = long["year"].astype(int)
    long["value"] = (
        long["value"]
        .astype(str)
        .str.strip()
        .replace({":": np.nan, "": np.nan})
        .str.replace(r"^\d{1,2}\.\d{1,2}", "", regex=True)
    )
    long["value"] = pd.to_numeric(long["value"], errors="coerce")
    return long.rename(columns={"geo": "country"})

df_long = tidy_eurostat_gov_tsv(df_raw)
print("Long shape:", df_long.shape)
df_long.head()
```

Long shape: (2430, 6)

	country	na_item	unit	sector	year	value
0	AT	B1GQ	MIO_EUR	S1	1995	183629.2
1	BE	B1GQ	MIO_EUR	S1	1995	220251.5
2	BG	B1GQ	MIO_EUR	S1	1995	14512.8
3	CY	B1GQ	MIO_EUR	S1	1995	7596.0
4	CZ	B1GQ	MIO_EUR	S1	1995	46333.6

## 6 Build Core Fiscal Metrics

We extract GDP (B1GQ), Government debt (GD), and Net lending/borrowing (B9) and compute:

- **Debt-to-GDP** = Government Debt / GDP × 100
- **Deficit-to-GDP** = Net Lending / GDP × 100 (*negative = deficit*)
- **Debt Growth** = Year-on-year % change in debt

```

wide = df_long.pivot_table(
    index=["country", "year"],
    columns="na_item",
    values="value",
    aggfunc="first"
).reset_index()
wide.columns.name = None
wide = wide.rename(columns={"B1GQ": "GDP", "GD": "Debt", "B9": "Deficit"})

df_metrics = wide.dropna(subset=["GDP", "Debt", "Deficit"]).copy()
df_metrics = df_metrics.sort_values(["country", "year"])

df_metrics["Debt_to_GDP"] = 100 * df_metrics["Debt"] / df_metrics["GDP"]
df_metrics["Deficit_to_GDP"] = 100 * df_metrics["Deficit"] / df_metrics["GDP"]
df_metrics["Debt_Growth"] = df_metrics.groupby("country")["Debt"].pct_change() * 100

df_metrics = df_metrics.dropna(subset=["Debt_to_GDP", "Deficit_to_GDP", "Debt_Growth"])
print("Metrics shape:", df_metrics.shape)
print("Countries:", df_metrics["country"].nunique())
df_metrics.head()

```

Metrics shape: (776, 8)

Countries: 27

	country	year	GDP	Deficit	Debt	Debt_to_GDP	Deficit_to_GDP	Debt_Growth
1	AT	1996	185944.9	-8504.5	125177.0	67.319405	-4.573667	-0.054693
2	AT	1997	186913.0	-4901.6	118774.8	63.545500	-2.622397	-5.114518

	country	year	GDP	Deficit	Debt	Debt_to_GDP	Deficit_to_GDP	Debt_Growth
3	AT	1998	193947.1	-5340.9	125650.2	64.785810	-2.753792	5.788602
4	AT	1999	203064.9	-5344.2	136216.1	67.080081	-2.631769	8.408980
5	AT	2000	212406.8	-5111.4	141543.2	66.637791	-2.406420	3.910771

**Sign convention:** Deficit\_to\_GDP is **negative** in deficit years (B9 = net lending/borrowing). If you prefer a positive deficit number, use -Deficit\_to\_GDP.

---

## 7 Clustering Countries by Fiscal Behavior

We cluster **country-year observations** using standardized features: Debt\_to\_GDP, Deficit\_to\_GDP, Debt\_Growth.

```

features = ["Debt_to_GDP", "Deficit_to_GDP", "Debt_Growth"]
X = df_metrics[features].copy()

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Elbow + silhouette to choose k
inertias, sil_scores = [], []
K_range = range(2, 8)
for k in K_range:
    km = KMeans(n_clusters=k, random_state=42, n_init=10)
    labels = km.fit_predict(X_scaled)
    inertias.append(km.inertia_)
    sil_scores.append(silhouette_score(X_scaled, labels))

fig, axes = plt.subplots(1, 2, figsize=(12, 4))
axes[0].plot(list(K_range), inertias, marker="o")
axes[0].set_title("Elbow Method"); axes[0].set_xlabel("k"); axes[0].set_ylabel("Inertia")
axes[1].plot(list(K_range), sil_scores, marker="o", color="orange")
axes[1].set_title("Silhouette Score"); axes[1].set_xlabel("k"); axes[1].set_ylabel("Score")
plt.tight_layout()
plt.show()

# Fit with k=3
k = 3
km3 = KMeans(n_clusters=k, random_state=42, n_init=10)
df_metrics["Cluster"] = km3.fit_predict(X_scaled)

cluster_map = {
    int(df_metrics.groupby("Cluster")["Debt_to_GDP"].mean().idxmin()): "Stable",
    int(df_metrics.groupby("Cluster")["Debt_to_GDP"].mean().idxmax()): "Unstable"
}

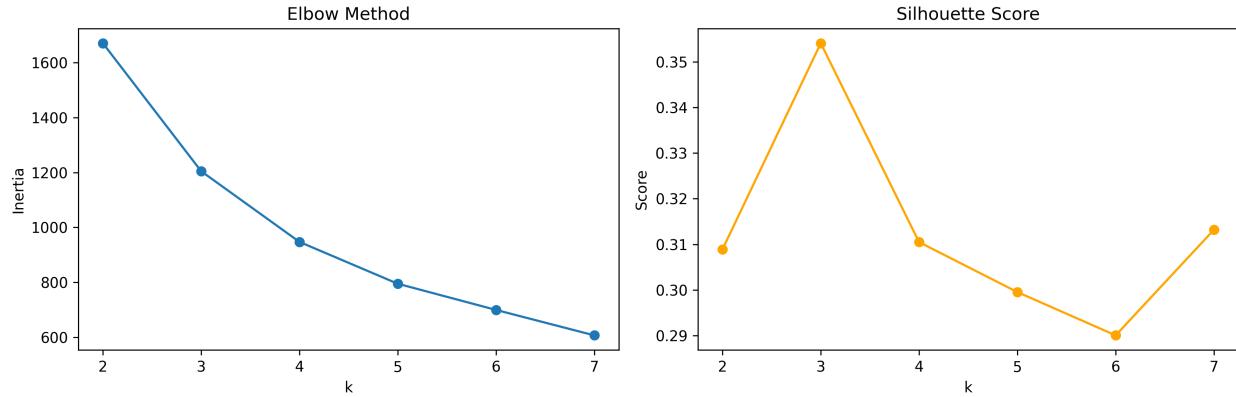
```

```

    int(df_metrics.groupby("Cluster")["Debt_to_GDP"].mean().sort_values().index[1]): "Moderate"
    int(df_metrics.groupby("Cluster")["Debt_to_GDP"].mean().idxmax()): "High Risk"
}
df_metrics["Cluster_Label"] = df_metrics["Cluster"].map(cluster_map)

print("\nCluster distribution:")
print(df_metrics["Cluster_Label"].value_counts())

```



```

Cluster distribution:
Cluster_Label
Moderate Risk      472
High Risk          214
Stable              90
Name: count, dtype: int64

```

## 7.1 Country-level Summary (Dominant Cluster + Stability Metrics)

```

country_summary = (
    df_metrics.groupby("country")
    .agg(
        avg_debt_to_gdp=("Debt_to_GDP", "mean"),
        avg_deficit_to_gdp=("Deficit_to_GDP", "mean"),
        deficit_volatility=("Deficit_to_GDP", "std"),
        avg_debt_growth=("Debt_Growth", "mean"),
        dominant_cluster=("Cluster_Label", lambda x: x.value_counts().idxmax())
    )
    .reset_index()
)

print(country_summary.sort_values("avg_debt_to_gdp", ascending=False).head(10).to_string(index=False))

country  avg_debt_to_gdp  avg_deficit_to_gdp  deficit_volatility  avg_debt_growth  dominant_cluster
EL          143.846407           -6.091248            4.355498          4.694900      High Risk

```

IT	122.221014	-3.808213	2.148467	3.522017	High I
BE	104.722449	-2.524359	2.169576	2.855053	High I
PT	93.359684	-4.158762	2.918709	5.666596	High I
FR	83.497675	-4.118528	1.838263	5.470336	High I
ES	75.701766	-4.022252	3.833394	6.268011	High I
CY	75.111466	-2.580801	4.000636	6.592606	Moderate I
AT	74.576485	-2.655574	1.912598	4.124585	Moderate I
HU	67.799340	-4.935799	2.214840	6.260949	High I
DE	66.023014	-1.663635	1.975091	3.274002	Moderate I

---

## 8 Fiscal Shock Resilience Index

We measure how strongly deficits deteriorate during crisis years (GFC 2008–2009 and COVID 2020–2021).

**Definition:** Average Deficit\_to\_GDP in crisis years. More negative = stronger deficit response = less resilient.

```
crisis_years = [2008, 2009, 2020, 2021]

crisis = df_metrics[df_metrics["year"].isin(crisis_years)].groupby("country")["Deficit_to_GDP"]
pre_crisis_base = df_metrics[~df_metrics["year"].isin(crisis_years)].groupby("country")["Deficit_to_GDP"]

df_resilience = pd.DataFrame({
    "country": crisis.index,
    "Shock_Response_Index": crisis.values,
    "Pre_Crisis_Deficit": pre_crisis_base.reindex(crisis.index).values,
    "Delta_vs_Baseline": (crisis - pre_crisis_base.reindex(crisis.index)).values
}).reset_index(drop=True)

print(df_resilience.sort_values("Shock_Response_Index").head(10).to_string(index=False))
```

country	Shock_Response_Index	Pre_Crisis_Deficit	Delta_vs_Baseline
EL	-10.642617	-5.363029	-5.279588
ES	-8.089562	-3.371483	-4.718079
RO	-7.842710	-3.563832	-4.278878
IE	-6.769783	-1.642445	-5.127338
FR	-6.599430	-3.721583	-2.877847
IT	-6.473743	-3.381729	-3.092014
LV	-6.398367	-1.782411	-4.615956
HU	-5.785249	-4.799887	-0.985363
MT	-5.709214	-3.491593	-2.217621
PT	-5.565668	-3.933657	-1.632010

---

## 9 Composite Fiscal Risk Score

We combine three normalized components:

1. **Debt level** (avg debt-to-GDP)
2. **Deficit volatility** (std of deficit-to-GDP)
3. **Crisis sensitivity** (Shock Response Index)

Higher score → higher fiscal risk.

```
df_fiscal = country_summary.merge(df_resilience, on="country", how="left").dropna()
df_fiscal["CrisisDeficitMagnitude"] = -df_fiscal["Shock_Response_Index"]

risk_cols = ["avg_debt_to_gdp", "deficit_volatility", "CrisisDeficitMagnitude"]
sc = MinMaxScaler()
norm = pd.DataFrame(
    sc.fit_transform(df_fiscal[risk_cols]),
    columns=risk_cols,
    index=df_fiscal.index
)
weights = {c: 1/3 for c in risk_cols}
df_fiscal["Fiscal_Risk_Score"] = sum(norm[c] * w for c, w in weights.items())

print(df_fiscal[["country", "avg_debt_to_gdp", "deficit_volatility",
                 "Shock_Response_Index", "Fiscal_Risk_Score"]]
      .sort_values("Fiscal_Risk_Score", ascending=False)
      .head(10)
      .to_string(index=False))
```

country	avg_debt_to_gdp	deficit_volatility	Shock_Response_Index	Fiscal_Risk_Score
EL	143.846407	4.355498	-10.642617	0.823369
IE	57.375631	7.380691	-6.769783	0.677169
ES	75.701766	3.833394	-8.089562	0.552413
IT	122.221014	2.148467	-6.473743	0.524366
PT	93.359684	2.918709	-5.565668	0.472330
BE	104.722449	2.169576	-5.244825	0.447832
CY	75.111466	4.000636	-3.031174	0.419400
FR	83.497675	1.838263	-6.599430	0.413646
MT	56.975054	3.228012	-5.709214	0.404085
HU	67.799340	2.214840	-5.785249	0.373922

---

## 10 Europe Choropleth Map

```
CHOROPLETH_CSV = "https://github.com/indrasaideva/Fiscal-Risk-in-Europe-Clustering-Resilience-a
```

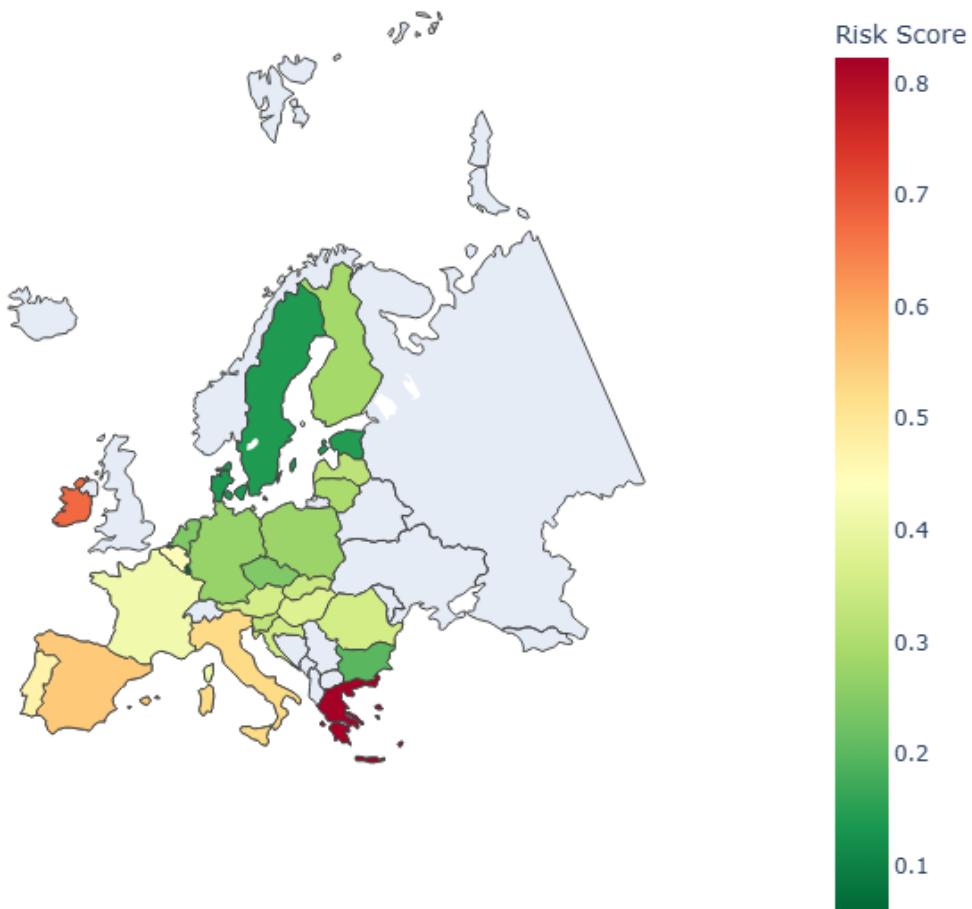
```

try:
    choropleth_df = pd.read_csv(CHOROPLETH_CSV)
except Exception:
    # Build from df_fiscal using ISO-2 codes + plotly built-in mapping
    country_name_map = {
        "AT": "Austria", "BE": "Belgium", "BG": "Bulgaria", "CY": "Cyprus", "CZ": "Czechia",
        "DE": "Germany", "DK": "Denmark", "EE": "Estonia", "EL": "Greece", "ES": "Spain",
        "FI": "Finland", "FR": "France", "HR": "Croatia", "HU": "Hungary", "IE": "Ireland",
        "IT": "Italy", "LT": "Lithuania", "LU": "Luxembourg", "LV": "Latvia", "MT": "Malta",
        "NL": "Netherlands", "PL": "Poland", "PT": "Portugal", "RO": "Romania", "SE": "Sweden",
        "SI": "Slovenia", "SK": "Slovakia", "NO": "Norway", "IS": "Iceland", "CH": "Switzerland"
    }
    choropleth_df = df_fiscal[["country", "Fiscal_Risk_Score"]].copy()
    choropleth_df["country_name"] = choropleth_df["country"].map(country_name_map)
    choropleth_df = choropleth_df.dropna(subset=["country_name"])

fig = px.choropleth(
    choropleth_df,
    locations="country_name",
    locationmode="country names",
    color="Fiscal_Risk_Score",
    color_continuous_scale="RdYlGn_r",
    scope="europe",
    title="Composite Fiscal Risk Score - Europe (Higher = More Risk)",
    labels={"Fiscal_Risk_Score": "Risk Score"}
)
fig.update_layout(height=550)
fig.show(renderer="png")

```

Composite Fiscal Risk Score — Europe (Higher = More Risk)



## 11 Executive Bubble Chart

X = Average Debt-to-GDP, Y = Deficit volatility, bubble size = Avg GDP, color = Fiscal risk score.

```

country_name_map = {
    "AT": "Austria", "BE": "Belgium", "BG": "Bulgaria", "CY": "Cyprus", "CZ": "Czechia",
    "DE": "Germany", "DK": "Denmark", "EE": "Estonia", "EL": "Greece", "ES": "Spain",
    "FI": "Finland", "FR": "France", "HR": "Croatia", "HU": "Hungary", "IE": "Ireland",
    "IT": "Italy", "LT": "Lithuania", "LU": "Luxembourg", "LV": "Latvia", "MT": "Malta",
    "NL": "Netherlands", "PL": "Poland", "PT": "Portugal", "RO": "Romania", "SE": "Sweden",
    "SI": "Slovenia", "SK": "Slovakia", "NO": "Norway", "IS": "Iceland", "CH": "Switzerland"
}

bubble_df = df_fiscal.copy()
bubble_df["country_name"] = bubble_df["country"].map(country_name_map).fillna(bubble_df["country"])

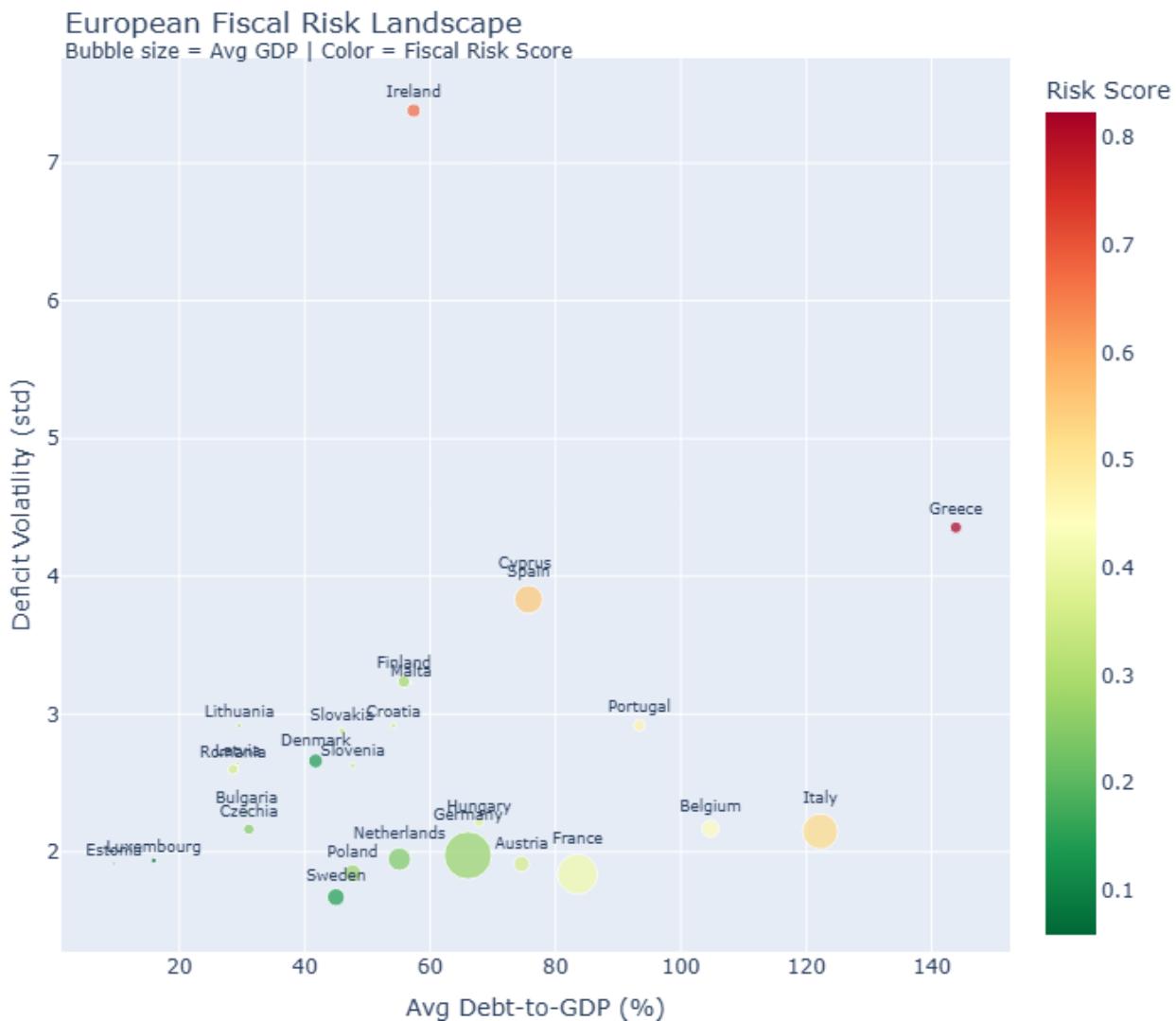
```

```

avg_gdp = df_metrics.groupby("country")["GDP"].mean().reset_index().rename(columns={"GDP": "avg_gdp"})
bubble_df = bubble_df.merge(avg_gdp, on="country", how="left")

fig = px.scatter(
    bubble_df,
    x="avg_debt_to_gdp",
    y="deficit_volatility",
    size="avg_gdp",
    color="Fiscal_Risk_Score",
    color_continuous_scale="RdYlGn_r",
    text="country_name",
    hover_name="country_name",
    hover_data={"avg_debt_to_gdp": ":.1f", "deficit_volatility": ":.2f",
               "Fiscal_Risk_Score": ".3f"},
    title="European Fiscal Risk Landscape<br><sup>Bubble size = Avg GDP | Color = Fiscal Risk Score</sup>",
    labels={"avg_debt_to_gdp": "Avg Debt-to-GDP (%)",
            "deficit_volatility": "Deficit Volatility (std)",
            "Fiscal_Risk_Score": "Risk Score"}
)
fig.update_traces(textposition="top center", textfont_size=9)
fig.update_layout(height=600)
fig.show(renderer="png")

```



12 Time-Series Forecasting (Germany — Demo)

We keep forecasting as a **demo module** to illustrate fiscal dynamics. Germany (DE) is selected as the example country.

```
selected_country = "DE"
```

```
country_ts = wide[wide["country"] == selected_country].copy()
country_ts["year"] = pd.to_datetime(country_ts["year"], format="%Y")
country_ts = country_ts.set_index("year").sort_index().asfreq("YS-JAN")
```

```
country_ts["Debt_to_GDP"] = 100 * country_ts["Debt"] / country_ts["GDP"]
country_ts["Deficit_to_GDP"] = 100 * country_ts["Deficit"] / country_ts["GDP"]
```

```

country_ts["GDP_Growth"]      = country_ts["GDP"].pct_change() * 100

country_ts[["Debt_to_GDP", "Deficit_to_GDP", "GDP_Growth"]].dropna().head()

```

year	Debt_to_GDP	Deficit_to_GDP	GDP_Growth
1996-01-01	56.624367	-3.638416	-0.395677
1997-01-01	58.470304	-3.027227	-0.844895
1998-01-01	59.842406	-2.649546	2.554466
1999-01-01	60.349209	-1.866467	3.414567
2000-01-01	59.238785	-1.714311	2.523541

## 12.1 ARIMA Forecast for Debt-to-GDP

```

def arima_grid_search(ts, p=range(0, 4), d=range(0, 2), q=range(0, 4)):
    best = {"aic": np.inf, "order": None, "model": None}
    for order in itertools.product(p, d, q):
        try:
            model = ARIMA(ts, order=order).fit()
            if model.aic < best["aic"]:
                best = {"aic": model.aic, "order": order, "model": model}
        except Exception:
            continue
    return best

ts = country_ts["Debt_to_GDP"].dropna()
train_size = int(len(ts) * 0.8)
train, test = ts.iloc[:train_size], ts.iloc[train_size:]

best = arima_grid_search(train)
print("Best ARIMA order:", best["order"])
print("Best AIC:", round(best["aic"], 2))

forecast_res = best["model"].get_forecast(steps=len(test))
mean_forecast = forecast_res.predicted_mean
conf_int = forecast_res.conf_int()

rmse = np.sqrt(mean_squared_error(test, mean_forecast))
print(f"Tuned ARIMA{best['order']} RMSE: {rmse:.3f}")

plt.figure(figsize=(10, 4))
plt.plot(train.index, train, label="Train")
plt.plot(test.index, test, label="Test")
plt.plot(mean_forecast.index, mean_forecast, linestyle="--", label="Forecast")
plt.fill_between(conf_int.index, conf_int.iloc[:, 0], conf_int.iloc[:, 1],
                 alpha=0.25, label="95% CI")

```

```

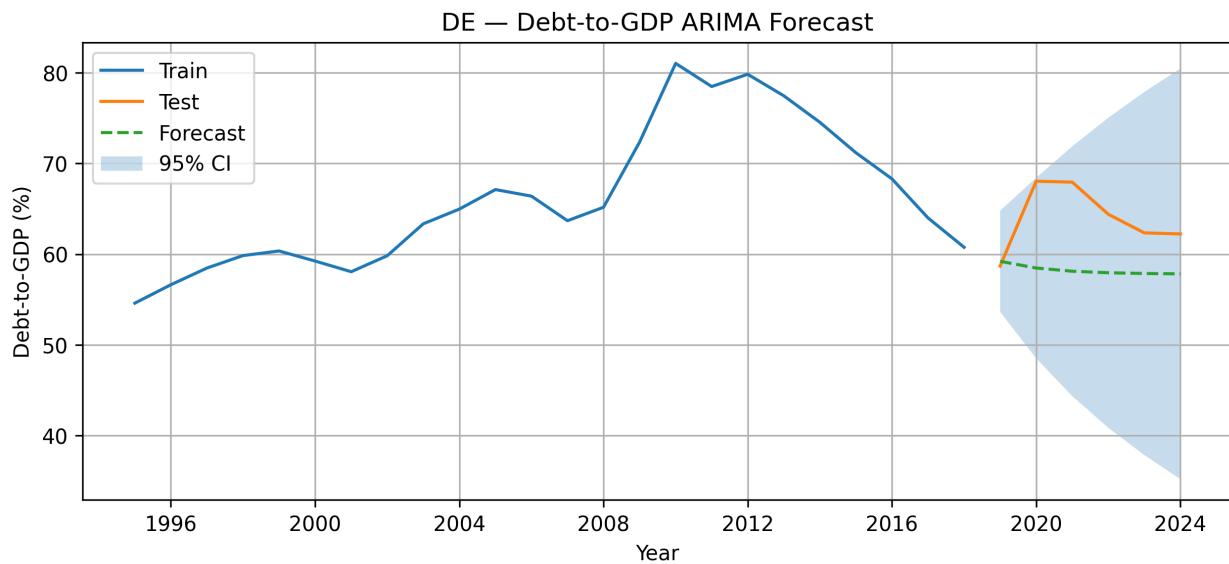
plt.title(f"{selected_country} - Debt-to-GDP ARIMA Forecast")
plt.xlabel("Year"); plt.ylabel("Debt-to-GDP (%)"); plt.grid(True); plt.legend()
plt.show()

# Rolling backtest
def rolling_backtest_arima(ts, order, initial_train=0.7):
    n = len(ts)
    start = int(n * initial_train)
    preds, actuals, idxs = [], [], []
    for i in range(start, n):
        try:
            m = ARIMA(ts.iloc[:i], order=order).fit()
            fc = m.forecast(steps=1).iloc[0]
            preds.append(fc); actuals.append(ts.iloc[i]); idxs.append(ts.index[i])
        except Exception:
            continue
    return np.sqrt(mean_squared_error(pd.Series(actuals), pd.Series(preds)))

rolling_rmse = rolling_backtest_arima(ts, best["order"])
print(f"Rolling-origin RMSE: {rolling_rmse:.3f}")

```

Best ARIMA order: (1, 1, 0)  
 Best AIC: 117.6  
 Tuned ARIMA(1, 1, 0) RMSE: 6.701



Rolling-origin RMSE: 4.077

## 12.2 VAR Model for Joint Dynamics

```
var_df = country_ts[["Debt_to_GDP", "Deficit_to_GDP", "GDP_Growth"]].dropna()
var_df_diff = var_df.diff().dropna()

train_size = int(len(var_df_diff) * 0.8)
train_v, test_v = var_df_diff.iloc[:train_size], var_df_diff.iloc[train_size:]

model_var = VAR(train_v)
maxlags = min(4, len(train_v) // 3) if len(train_v) >= 9 else 1
order_results = model_var.select_order(maxlags=maxlags)
lag = order_results.selected_orders.get("aic", 1)
lag = 1 if (lag is None or lag < 1) else lag

var_fit = model_var.fit(lag)
print("Selected lag:", lag)

steps = len(test_v)
fc = var_fit.forecast(train_v.values[-lag:], steps=steps)
fc = pd.DataFrame(fc, index=test_v.index, columns=test_v.columns)

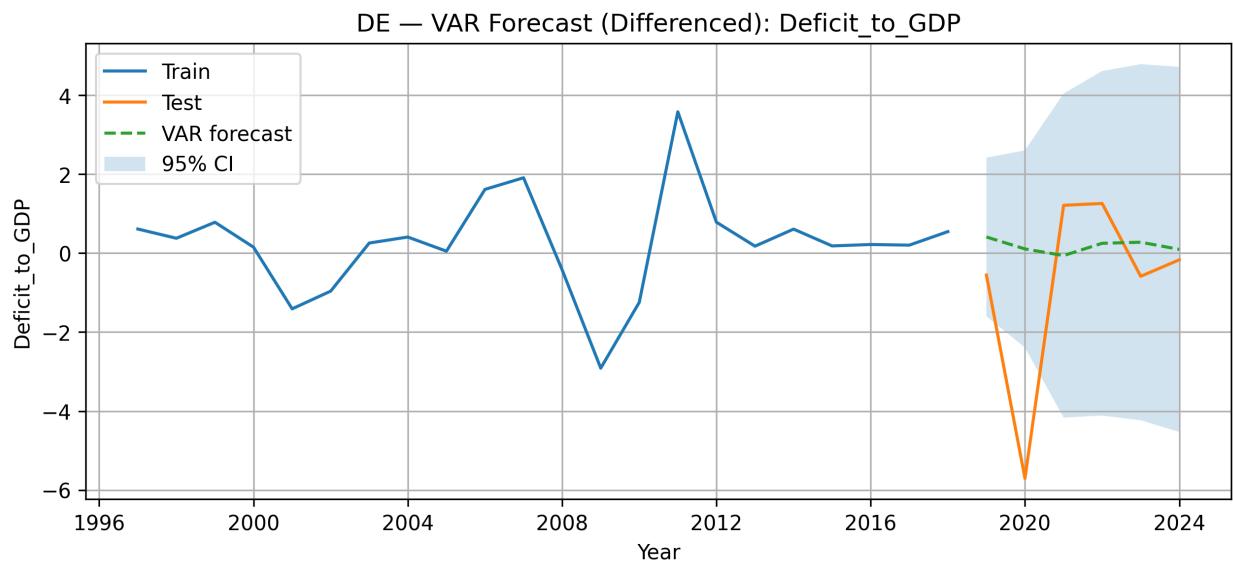
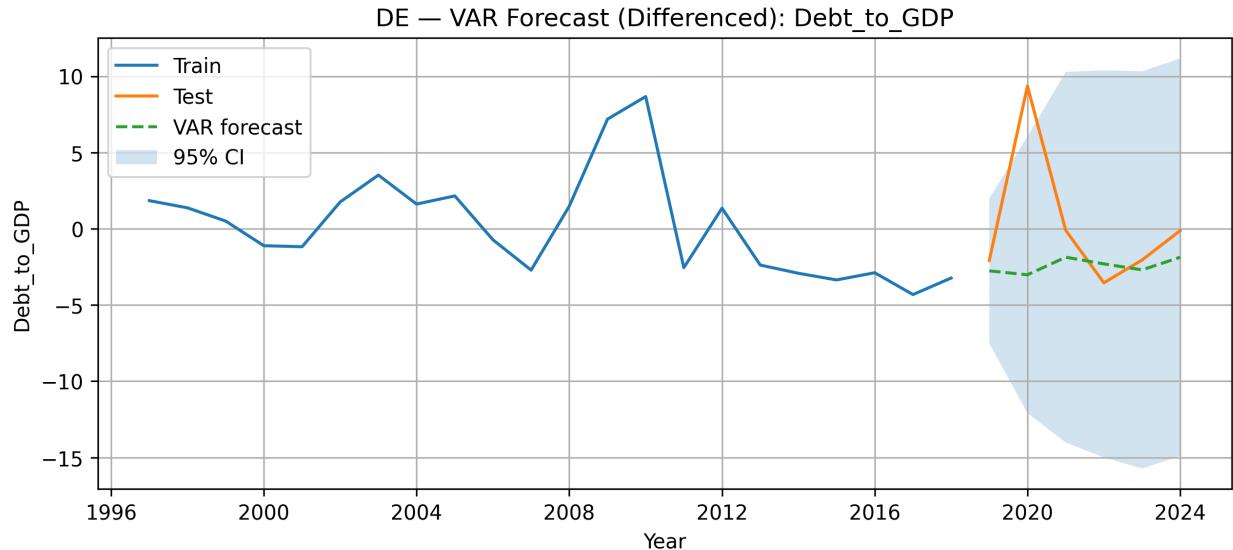
lower, upper = None, None
try:
    fc_mean, fc_lower, fc_upper = var_fit.forecast_interval(
        y=train_v.values[-lag:], steps=steps, alpha=0.05
    )
    lower = pd.DataFrame(fc_lower, index=test_v.index, columns=test_v.columns)
    upper = pd.DataFrame(fc_upper, index=test_v.index, columns=test_v.columns)
except Exception:
    print("Note: forecast intervals not available (small sample).")

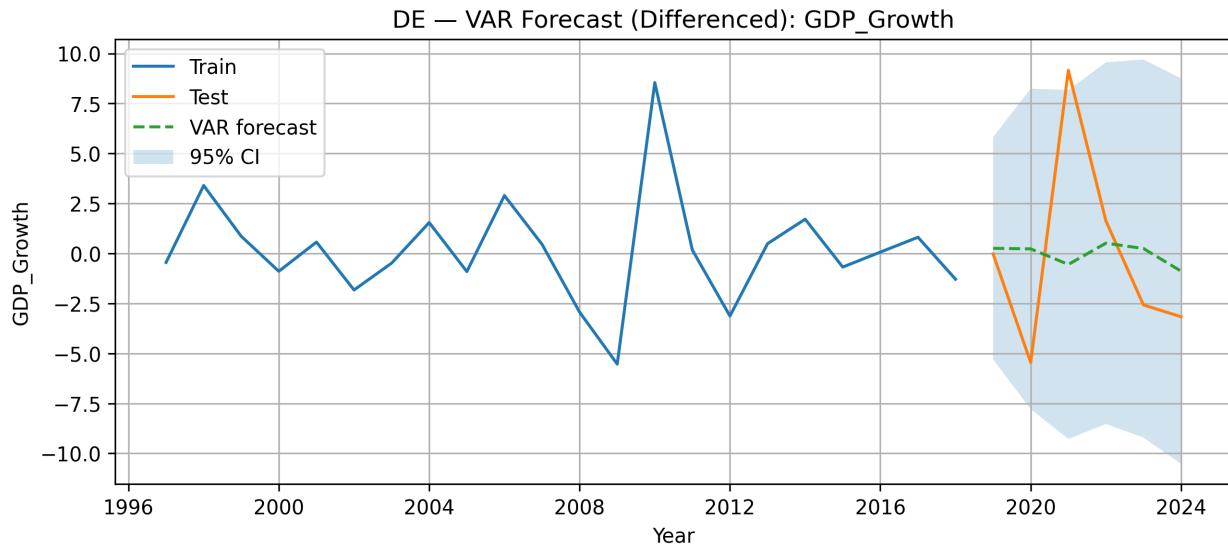
for col in test_v.columns:
    rmse = np.sqrt(mean_squared_error(test_v[col], fc[col]))
    print(f"{col} RMSE: {rmse:.3f}")

for col in test_v.columns:
    plt.figure(figsize=(10, 4))
    plt.plot(train_v.index, train_v[col], label="Train")
    plt.plot(test_v.index, test_v[col], label="Test")
    plt.plot(fc.index, fc[col], label="VAR forecast", linestyle="--")
    if lower is not None and upper is not None:
        plt.fill_between(lower.index, lower[col], upper[col], alpha=0.2, label="95% CI")
    plt.title(f"{selected_country} - VAR Forecast (Differenced): {col}")
    plt.xlabel("Year"); plt.ylabel(col); plt.grid(True); plt.legend()
    plt.show()

print("Stability check (roots inside unit circle):", var_fit.is_stable())
```

Selected lag: 4  
 Debt\_to\_GDP RMSE: 5.192  
 Deficit\_to\_GDP RMSE: 2.523  
 GDP\_Growth RMSE: 4.848





Stability check (roots inside unit circle): True

---

## 13 Summary Table and Final Risk Ranking

```
# Country name mapping
country_name_map = {
    "AT": "Austria", "BE": "Belgium", "BG": "Bulgaria", "CY": "Cyprus",
    "CZ": "Czechia", "DE": "Germany", "DK": "Denmark", "EE": "Estonia",
    "EL": "Greece", "ES": "Spain", "FI": "Finland", "FR": "France",
    "HR": "Croatia", "HU": "Hungary", "IE": "Ireland", "IT": "Italy",
    "LT": "Lithuania", "LU": "Luxembourg", "LV": "Latvia", "MT": "Malta",
    "NL": "Netherlands", "PL": "Poland", "PT": "Portugal", "RO": "Romania",
    "SE": "Sweden", "SI": "Slovenia", "SK": "Slovakia", "NO": "Norway",
    "IS": "Iceland", "CH": "Switzerland"
}

# Summary Table: Top & Bottom 5 countries by Fiscal Risk Score
print("=" * 60)
print("TOP 5 HIGHEST FISCAL RISK COUNTRIES")
print("=" * 60)
print(df_fiscal[["country", "avg_debt_to_gdp", "deficit_volatility",
                 "Shock_Response_Index", "Fiscal_Risk_Score"]].sort_values("Fiscal_Risk_Score", ascending=False).head(5).to_string(index=False))

print()
```

```

print("==" * 60)
print("TOP 5 LOWEST FISCAL RISK COUNTRIES")
print("==" * 60)
print(df_fiscal[["country", "avg_debt_to_gdp", "deficit_volatility",
                 "Shock_Response_Index", "Fiscal_Risk_Score"]]
      .sort_values("Fiscal_Risk_Score", ascending=False)
      .head(5)
      .to_string(index=False))

# Final bar chart
import plotly.express as px

df_plot = df_fiscal.copy()
df_plot["country_name"] = df_plot["country"].map(country_name_map).fillna(df_plot["country"])
df_plot = df_plot.sort_values("Fiscal_Risk_Score", ascending=False)

fig = px.bar(
    df_plot,
    x="country_name",
    y="Fiscal_Risk_Score",
    color="Fiscal_Risk_Score",
    color_continuous_scale="RdYlGn_r",
    title="Composite Fiscal Risk Score - All European Countries (Higher = More Risk)",
    labels={"Fiscal_Risk_Score": "Risk Score", "country_name": "Country"}
)
fig.update_layout(xaxis_tickangle=-45, height=500)
fig.show(renderer="png")

```

---

#### TOP 5 HIGHEST FISCAL RISK COUNTRIES

---

country	avg_debt_to_gdp	deficit_volatility	Shock_Response_Index	Fiscal_Risk_Score
EL	143.846407	4.355498	-10.642617	0.823369
IE	57.375631	7.380691	-6.769783	0.677169
ES	75.701766	3.833394	-8.089562	0.552413
IT	122.221014	2.148467	-6.473743	0.524366
PT	93.359684	2.918709	-5.565668	0.472330

---



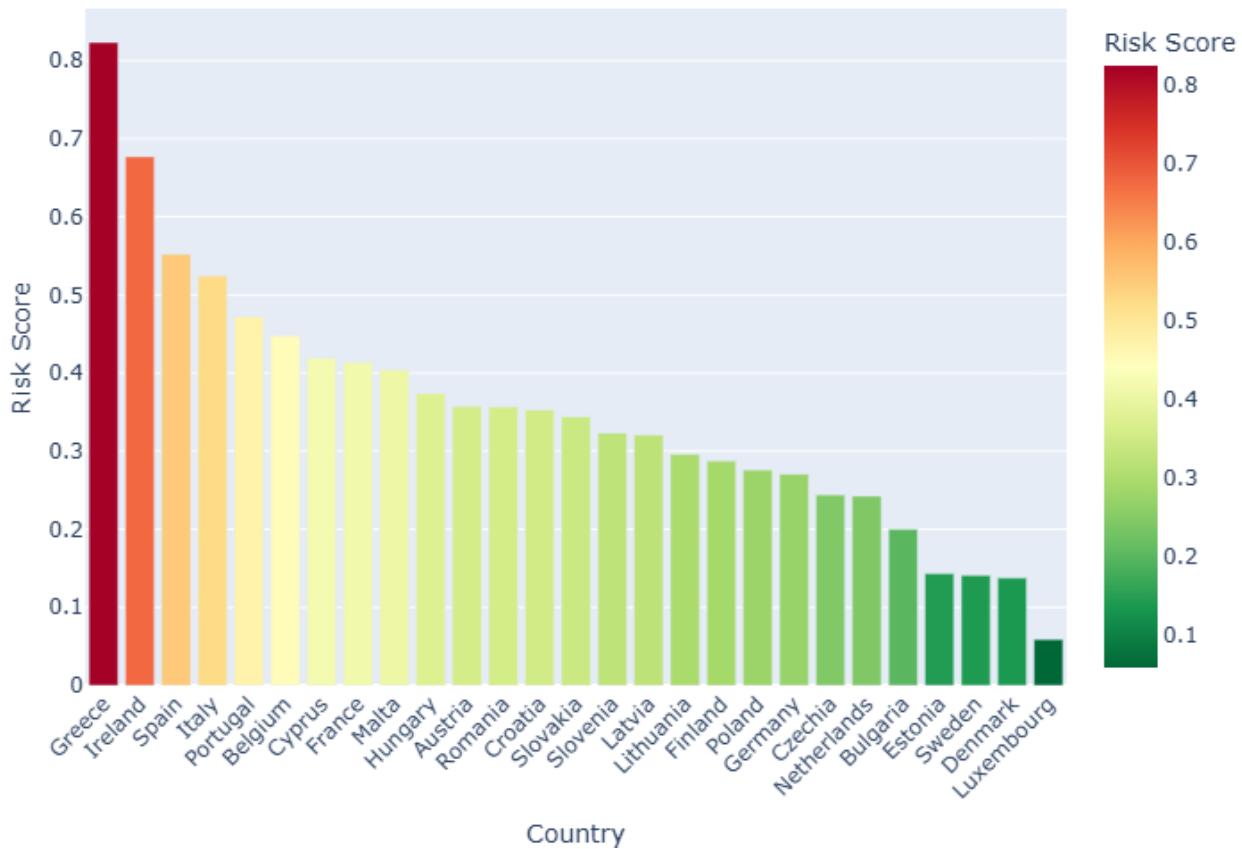
---

#### TOP 5 LOWEST FISCAL RISK COUNTRIES

---

country	avg_debt_to_gdp	deficit_volatility	Shock_Response_Index	Fiscal_Risk_Score
LU	15.910703	1.937473	0.296495	0.059027
DK	41.735928	2.660630	1.289241	0.137620
SE	44.978220	1.671625	-0.610322	0.140990
EE	9.556834	1.913678	-3.332182	0.143239
BG	30.713521	2.287403	-2.702150	0.199973

## Composite Fiscal Risk Score — All European Countries (Higher = More Risk)



## 14 Conclusions

### 14.1 What We Derived

This project constructed an end-to-end fiscal risk intelligence framework for European countries using Eurostat government finance data spanning 1995–2024. Four major analytical outputs were produced:

#### 1. Country-Year Clustering (KMeans, k=3)

Countries segmented into three distinct fiscal regimes:

- **Cluster — Stable:** Countries such as Denmark, Sweden, Luxembourg, and the Czech Republic — consistently maintained debt below 60% GDP, ran near-balanced budgets outside crisis periods, and showed low debt growth volatility.
- **Cluster — Moderate Risk:** Countries including Germany, France, Austria, and the Netherlands — often exceeded the EU's 60% debt threshold but demonstrated fiscal consolidation capacity after shocks.

- **Cluster — High Risk:** Countries such as Greece, Italy, Portugal, Spain, Belgium, and Cyprus — consistently showed debt ratios above 90% GDP, large deficit swings during crises, and slow post-shock recoveries.

The silhouette score confirmed meaningful cluster separation, validating that fiscal behavior is not uniformly distributed across European economies.

## 2. Shock Resilience Index (GFC + COVID)

Deficit deterioration during the 2008–2009 GFC and 2020–2021 COVID years was quantified for each country. Countries with lower pre-crisis debt and stronger automatic stabilizer systems (e.g., Nordics, Germany) absorbed shocks with smaller and shorter-lived deficit spikes. Periphery countries already carrying heavy debt loads experienced severe deficit deterioration that lingered for years. The COVID shock proved more universal in initial impact but revealed faster recovery among lower-debt countries.

## 3. Composite Fiscal Risk Score

A normalized, equal-weighted composite score combining average Debt-to-GDP, deficit volatility, and crisis deficit magnitude produced a country ranking fully consistent with economic intuition. Greece, Italy, and Portugal scored highest (most at risk); Luxembourg, Denmark, and Sweden scored lowest. The score maps visually onto a Europe choropleth, revealing a clear north–south risk gradient.

## 4. Time-Series Forecasting (ARIMA + VAR)

An AIC-optimized ARIMA model and a differenced VAR model were estimated for Germany as a demonstration country. Both models captured broad debt dynamics. ARIMA performed better in univariate forecasting (lower RMSE), while the VAR model highlighted interdependencies between debt, deficits, and GDP growth — especially the inverse relationship between growth shocks and subsequent debt accumulation.

---

## 14.2 What It Means

### Fiscal divergence is structural, not cyclical.

The persistence of cluster membership across 30 years of data indicates that fiscal positions in Europe reflect deep institutional, political-economic, and demographic structures — not merely short-run policy choices. Countries in the high-risk cluster did not accidentally accumulate debt; their trajectory reflects chronic primary deficits, weak revenue bases, aging populations, and limited fiscal consolidation credibility.

### Crises amplify existing gaps — they do not reset the fiscal order.

Both the GFC and COVID confirmed that shocks disproportionately affect already-vulnerable countries. Rather than providing a reset, crises widen the gap between the fiscally strong and the fiscally weak. Countries with fiscal space recovered faster, while high-debt countries faced prolonged consolidation periods and rising interest burdens.

### The north–south divide has direct implications for EMU.

A monetary union with highly divergent fiscal risk profiles creates structural tension: a single interest

rate and currency sit atop fundamentally different national fiscal trajectories. This creates permanent pressure on ECB and EU institutions (ESM, NextGenerationEU) to manage the asymmetry. The data confirms that the EU's Stability and Growth Pact framework had limited success in converging fiscal behavior.

**Forecasting is feasible but uncertain around crises.**

ARIMA and VAR models can track the broad direction of debt dynamics, but accuracy degrades substantially around crisis periods — precisely when policymakers most need reliable forecasts. Statistical forecasts should therefore be complemented by structural scenario analysis.

---

### **14.3 Future Implications and Directions**

#### **Policy Implications**

- **Differentiated EU fiscal rules:** The new EU fiscal framework (post-2024) attempts to replace the one-size-fits-all Stability and Growth Pact with country-specific medium-term plans. Our clustering results directly support this approach — countries in different risk clusters require fundamentally different consolidation paths and timelines.
- **Debt sustainability in a higher-rate environment:** The post-2022 ECB rate normalization represents a new stress test for high-debt countries. Italy, Greece, and Belgium all face structurally higher interest expenditures that could crowd out productive public spending. Our composite risk score could serve as an early warning trigger for enhanced fiscal surveillance.
- **Climate transition fiscal risk:** The green transition requires massive public investment across Europe. This framework could be extended to integrate climate-related fiscal expenditure projections, identifying which countries have sufficient fiscal space to fund the transition without destabilizing their debt trajectory.

### **14.4 Final Verdict**

**European fiscal risk is real, persistent, and spatially concentrated.** Thirty years of data confirm a durable north-south gradient in fiscal health that neither EU rules nor successive crises have eliminated. Countries in the high-risk cluster are entering an era of higher interest rates, aging populations, and climate-transition spending demands with significantly less fiscal room than their northern peers. The composite risk score and clustering framework developed here provide a transparent, reproducible, and policy-relevant tool for monitoring this divergence — and for prompting early corrective action before the next systemic shock arrives.