

Supplementary Material S1

Follows the code in Python:

```
# 2025 ©® All Rights Reserved Constantinos Challoumis
import pandas as pd
import matplotlib.pyplot as plt

# Data preparation
data = {
    "Sector of Economic Activity": [
        "Agriculture, forestry and fishing",
        "Industry (excluding construction)",
        "Manufacturing",
        "Construction",
        "Services (Total)",
        "Trade, transport, accommodation and food service activities",
        "Information and communication",
        "Financial and insurance activities",
        "Real estate activities",
        "Professional, scientific, technical and administrative activities",
        "Public administration, defense, education, health and social work",
        "Arts, entertainment, other services, and household staff"
    ],
    2013: [4.30, 14.90, 9.00, 2.80, 78.20, 25.00, 3.30, 4.20, 17.10, 5.30, 20.10, 3.20],
    2015: [4.40, 14.20, 9.40, 2.20, 79.20, 24.80, 2.80, 5.00, 18.00, 4.90, 20.50, 3.10],
    2017: [4.50, 14.60, 9.30, 1.70, 79.10, 25.40, 3.00, 5.60, 17.50, 4.90, 19.70, 3.20],
    2019: [4.60, 14.70, 9.80, 1.90, 78.80, 25.20, 2.80, 5.00, 17.90, 5.30, 19.00, 3.60],
    2021: [4.10, 17.10, 11.80, 2.00, 76.80, 22.90, 3.40, 4.90, 17.00, 5.50, 19.70, None],
    2023: [4.10, 15.40, 11.90, 2.30, 78.10, 24.00, 3.60, 4.90, 16.50, 6.10, 19.20, None],
    "Change 2013–2023": [-0.2, 0.5, 2.9, -0.4, -0.1, -1.0, 0.2, 0.7, -0.6, 0.8, -0.9, None]
}

df = pd.DataFrame(data)

# Plot 1: Trends over time (line chart for key sectors)
plt.figure(figsize=(12, 7))
for sector in ["Agriculture, forestry and fishing", "Industry (excluding construction)",
"Manufacturing", "Construction", "Services (Total)"]:
    plt.plot(df.columns[1:-1], df[df["Sector of Economic Activity"] == sector].iloc[0,
1:-1], marker='o', label=sector)

plt.title("Economic Activity by Sector (2013–2023)", fontsize=14)
plt.xlabel("Year")
plt.ylabel("Share of GDP (%)")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.6)
plt.show()
```

```
# Plot 2: Change 2013–2023 (bar chart)
plt.figure(figsize=(10, 6))
df_change = df.dropna(subset=["Change 2013–2023"])
plt.barh(df_change["Sector of Economic Activity"], df_change["Change 2013–2023"],
color="skyblue")
plt.axvline(0, color="black", linewidth=0.8)
plt.title("Change in Sector Share of GDP (2013–2023)", fontsize=14)
plt.xlabel("Percentage Points Change")
plt.show()
```

Supplementary Material S2

Follows the code in Python:

```
# 2025 ©® All Rights Reserved Constantinos Challoumis
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt

# ---- Data ----
data = {
    "Year": [2013, 2015, 2017, 2019, 2021, 2023],
    "Agriculture": [4.30, 4.40, 4.50, 4.60, 4.10, 4.10],
    "Industry": [14.90, 14.20, 14.60, 14.70, 17.10, 15.40],
    "Manufacturing": [9.00, 9.40, 9.30, 9.80, 11.80, 11.90],
    "Construction": [2.80, 2.20, 1.70, 1.90, 2.00, 2.30],
    "Services": [78.20, 79.20, 79.10, 78.80, 76.80, 78.10],
}
df = pd.DataFrame(data)

# ---- Helper to fit OLS and plot ----
def plot_trend(series_name: str):
    X = sm.add_constant(df["Year"])
    y = df[series_name]
    model = sm.OLS(y, X).fit()
    yhat = model.predict(X)

    plt.figure(figsize=(6, 4))
    plt.scatter(df["Year"], y, label="Observed")
    plt.plot(df["Year"], yhat, label="Fitted trend")
    plt.title(f"{series_name} (% of GDP)")
    plt.xlabel("Year")
    plt.ylabel("Share (%)")
    plt.grid(True)
    plt.legend()
    plt.tight_layout()
    # Uncomment to save each figure:
    # plt.savefig(f"{series_name.lower().replace(' ', '_')}_trend.png", dpi=300)
    plt.show()

# ---- charts ----
for col in ["Agriculture", "Industry", "Manufacturing", "Construction", "Services"]:
    plot_trend(col)
```

Supplementary Material S3

Follows the code in R:

```
# 2025 ©® All Rights Reserved Constantinos Challoumis
# Compositional Trend Model — Log-Ratio vs Services (2010–2023)
#  $\ln(Y_i / Y_{\text{services}}) = \alpha_i + \beta_i * \text{Year} + \varepsilon_i$ 
# Robust (HAC, Newey–West) standard errors and p-values

# 1) Install & load packages
need <- c("dplyr", "tidyr", "readr", "lmtest", "sandwich")
to_install <- need[!need %in% installed.packages()], "Package"]
if (length(to_install)) install.packages(to_install, quiet = TRUE)
invisible(lapply(need, library, character.only = TRUE))

# 2) Data
years <- 2010:2023
sectors <- data.frame(
  year = years,
  agriculture = c(4.2,4.1,4.3,4.4,4.5,4.6,4.3,4.2,4.1,4.0,4.1,4.1,4.2,4.2),
  manufacturing = c(8.5,8.7,8.8,9.0,9.2,9.3,9.5,9.8,10.1,10.6,10.8,11.1,11.6,11.9),
  trade = c(26.2,26.1,25.8,25.4,25.3,25.0,24.8,24.6,24.5,24.4,24.1,24.0,23.9,23.8),
  finance = c(5.1,5.2,5.3,5.4,5.4,5.3,5.2,5.1,5.0,4.9,4.8,4.8,4.8,4.9),
  info = c(2.5,2.6,2.7,2.9,3.1,3.3,3.4,3.5,3.6,3.7,3.8,3.9,4.0,4.1),
  public = c(20.8,20.7,20.6,20.4,20.2,20.1,19.9,19.8,19.7,19.6,19.4,19.3,19.3,19.2),
  services = c(77.9,78.0,78.2,78.3,78.4,78.5,78.3,78.2,78.1,78.0,77.8,77.7,77.6,77.5)
)

# 3) Log-ratio variables relative to Services
Z <- sectors |>
  mutate(
    ln_agri = log(agriculture / services),
    ln_manu = log(manufacturing / services),
    ln_trade = log(trade / services),
    ln_fin = log(finance / services),
    ln_info = log(info / services),
    ln_pub = log(public / services)
  ) |>
  select(year, starts_with("ln_"))

# 4) Trend estimator: OLS + HAC (Newey–West) SE, p-value
trend_est <- function(y, years) {
  fit <- lm(y ~ years)
  ct <- coeftest(fit, vcov = NeweyWest(fit, lag = 1, prewhite = FALSE))
  data.frame(
    beta = unname(ct["years", "Estimate"]),
    se = unname(ct["years", "Std. Error"]),
    p_value = unname(ct["years", "Pr(>|t|)"]),
    R2 = summary(fit)$r.squared
  )
}
```

```

# 5) Run for each sector
res <- data.frame(Sector = character(), Beta = numeric(), p_value = numeric(), R2 =
numeric())

for (v in names(Z)[-1]) {
  r <- trend_est(Z[[v]], Z$year)
  res <- rbind(res, data.frame(Sector = v, Beta = r$beta, p_value = r$p_value, R2 =
r$R2))
}

# 6) Format + significance stars (use p_value, not 'p-value')
res <- res |>
  mutate(
    Avg_pct_change_per_year = round(100 * Beta, 2),
    signif = dplyr::case_when(
      p_value < 0.01 ~ "****",
      p_value < 0.05 ~ "***",
      p_value < 0.10 ~ "**",
      TRUE ~ ""
    )
  ) |>
  arrange(p_value) |>
  select(Sector, Beta, Avg_pct_change_per_year, R2, p_value, signif)

# 7) Output
print(res, row.names = FALSE)
write.csv(res, "compositional_trend_results.csv", row.names = FALSE)
cat("\nSaved: compositional_trend_results.csv\n")

```

Supplementary Material S4

Follows the code in Python of the conclusions:

```
# 2025 ©® All Rights Reserved Constantinos Challoumis
import pandas as pd
import numpy as np
import statsmodels.api as sm
from statsmodels.tsa.stattools import adfuller
from statsmodels.stats.diagnostic import acorr_breusch_godfrey

# -----
# 1. Data entry
# -----
data = {
    "Year": [2013, 2015, 2017, 2019, 2021, 2023],
    "Agriculture": [4.3, 4.4, 4.5, 4.6, 4.1, 4.1],
    "Industry": [14.9, 14.2, 14.6, 14.7, 17.1, 15.4],
    "Manufacturing": [9.0, 9.4, 9.3, 9.8, 11.8, 11.9],
    "Construction": [2.8, 2.2, 1.7, 1.9, 2.0, 2.3],
    "Services": [78.2, 79.2, 79.1, 78.8, 76.8, 78.1],
    "Trade_Tourism": [25.0, 24.8, 25.4, 25.2, 22.9, 24.0],
    "Info_Comm": [3.3, 2.8, 3.0, 2.8, 3.4, 3.6],
    "Finance": [4.2, 5.0, 5.6, 5.0, 4.9, 4.9],
    "RealEstate": [17.1, 18.0, 17.5, 17.9, 17.0, 16.5],
    "Prof_Admin": [5.3, 4.9, 4.9, 5.3, 5.5, 6.1],
    "Public_Admin": [20.1, 20.5, 19.7, 19.0, 19.7, 19.2],
}
df = pd.DataFrame(data).set_index("Year")

# -----
# 2. Computation of log-ratios relative to Services (base sector)
# -----
for col in df.columns:
    if col != "Services":
        df[f"ln_{col}"] = np.log(df[col] / df["Services"])

# -----
# 3. Diagnostics
# -----
def diagnostics(series, base_df=df, alt_base="Industry"):
    """Return ADF p-value, BG p-value, sensitivity to base sector."""
    # Regression: log-ratio on linear trend
    Y = series.dropna()
    X = sm.add_constant(range(len(Y)))
    model = sm.OLS(Y, X).fit(cov_type="HAC", cov_kwds={"maxlags": 1})
    beta = model.params.iloc[1] # <-- FIXED (use iloc instead of [1])

    # ADF test
    adf_p = adfuller(Y, regression="ct")[1]

    # Breusch–Godfrey serial-correlation test (2 lags)
```

```

bg_p = acorr_breusch_godfrey(model, nlags=2)[1]

# Sensitivity to base sector
ln_alt = np.log(base_df[series.name.replace("ln_", "")] / base_df[alt_base])
model_alt = sm.OLS(ln_alt, X).fit(cov_type="HAC", cov_kwds={"maxlags": 1})
delta_beta = abs(beta - model_alt.params.iloc[1]) # <-- FIXED

return adf_p, bg_p, delta_beta, beta

# -----
# 4. Diagnostics for all sectors (except Services)
# -----
results = []
for col in df.columns:
    if col.startswith("ln_") and col != "ln_Services":
        adf_p, bg_p, delta, beta = diagnostics(df[col])
        results.append({
            "Sector": col.replace("ln_", "").replace("_", " "),
            "ADF (p-val)": round(adf_p, 3),
            "BG (p-val)": round(bg_p, 3),
            " $\Delta\beta$ ": round(delta, 3),
            " $\beta$ ": round(beta, 4)
        })

summary = pd.DataFrame(results)
print("\n=== Diagnostic Summary Table ===")
print(summary.to_string(index=False))

```