**Final Report – CS 2704: Data Analytics Using Python**

**Team Members**

* Jaspinder Singh
* Syed Owais Haider Kazmi
* Nomaan Imran Saiyed

-----------------Minimum Requirements (70%) -------------------

**Hypothesis**

The hypothesis being tested is: -

1)H₀ (Null Hypothesis): GDP per capita does not predict unemployment.

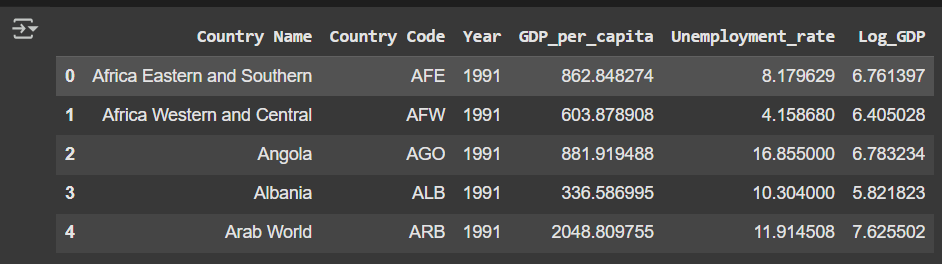
2)H₁ (Alternative Hypothesis): GDP per capita does predict unemployment (negatively correlated).

We tested this by analyzing world data from the World Bank, using correlation, linear regression, and machine learning models. The final result was that we failed to reject the null hypothesis.

**Data & Metadata**

**Sources:**

* GDP: <https://data.worldbank.org/indicator/NY.GDP.PCAP.CD>
* Unemployment: <https://data.worldbank.org/indicator/SL.UEM.TOTL.ZS>

**Metadata:** Code

Details:

df = pd.read\_csv("merged\_data.csv")

df['Log\_GDP'] = np.log(df['GDP\_per\_capita'] + 1)

Null values are dropped using df.dropna(inplace=True)  
  
**Descriptive Analytics**

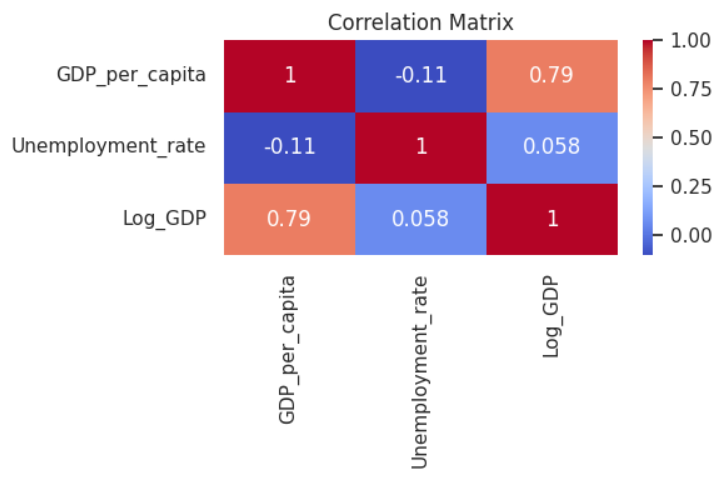
1. **Summary Statistics:**

-df.describe() provides count, mean, std, min, max for each feature.

1. **Correlation Heatmap:**

-Using sns.heatmap(df.corr(), annot=True) shows:

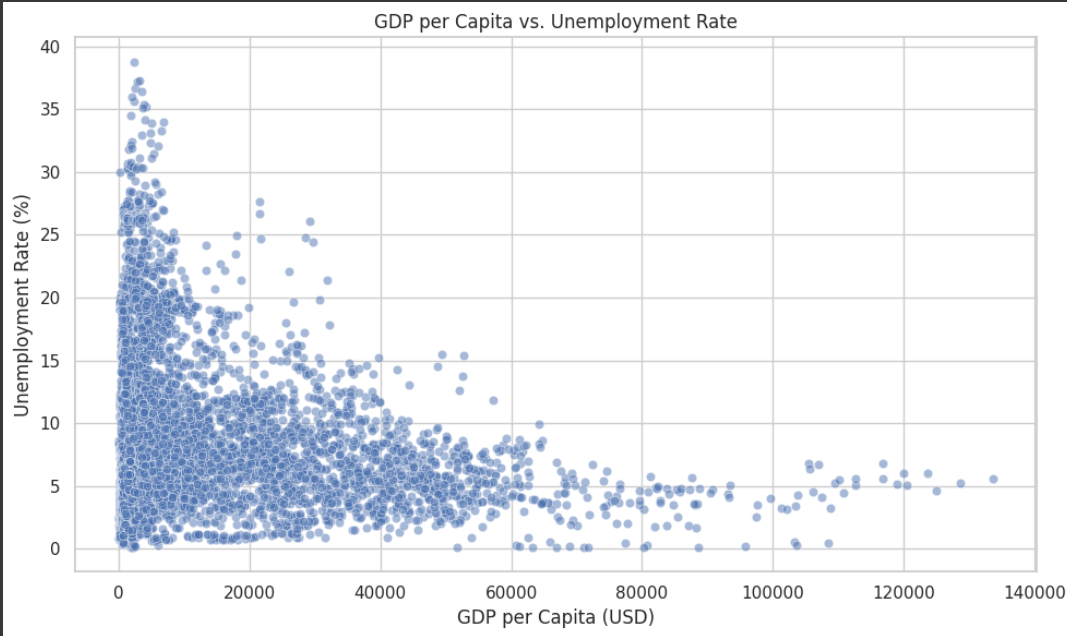
- Slight negative correlation between Log\_GDP and Unemployment.



1. **Scatter Plot:**

-sns.scatterplot(x='GDP\_per\_capita', y='Unemployment', data=df)

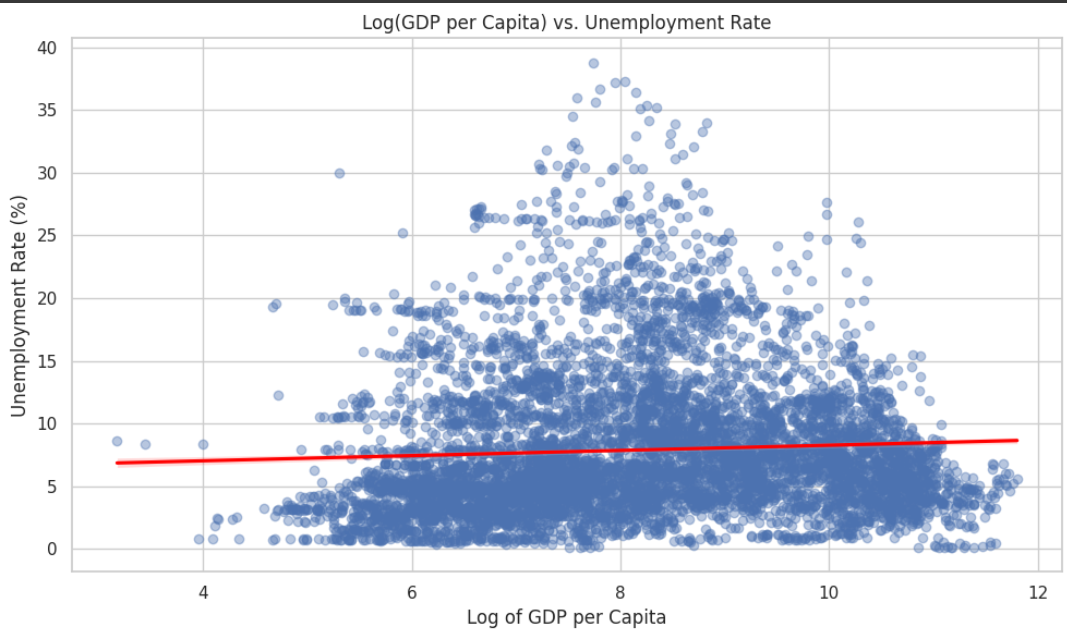
-Reveals wide dispersion — only a slight downward trend.



1. **Regression Plot:**

-sns.regplot(x='Log\_GDP', y='Unemployment', data=df)

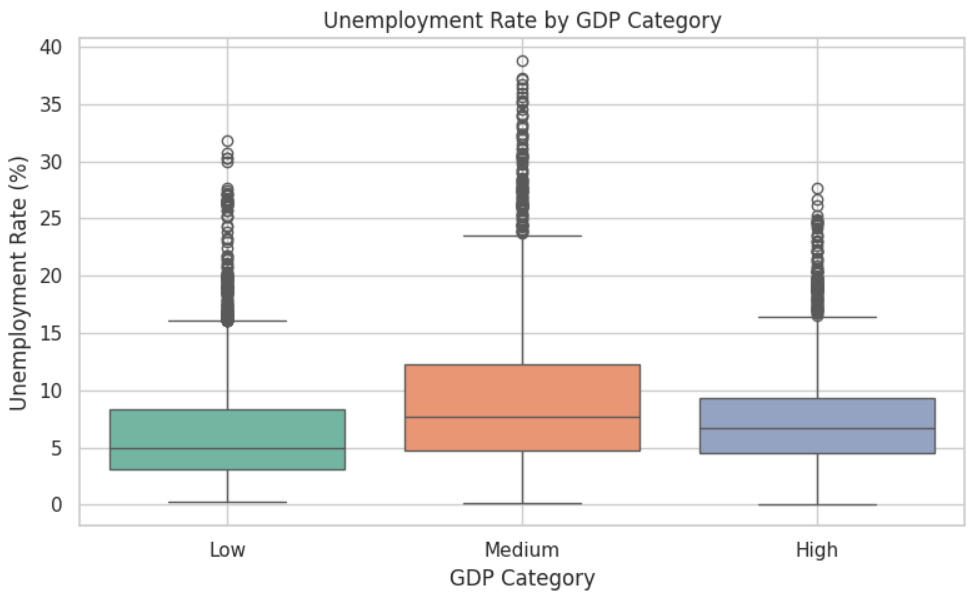
-Visual trend improves due to normalization but still weak correlation.



1. **Box Plot:**

-Countries are grouped by GDP (Low, Medium, High).

-Median unemployment is highest in the medium GDP group.



**Predictive Analytics**

Response Variable: Unemployment

Predictor Variable**:** Log GDP

**🔸 Models Implemented: -**

1. **Linear Regression**

-Libraries: from sklearn.linear\_model import LinearRegression, statsmodels.api

-Formula: model = sm.OLS(Y, X).fit()

-Output: Coefficient, R², p-value (statistically insignificant)

1. **Polynomial Regression (degree=2)**

-Feature transformation: PolynomialFeatures(degree=2)

-Better fit visually, but low statistical significance.

1. **Random Forest Regression**

-RandomForestRegressor(n\_estimators=100)

-Output was overfitted, and performance dropped when generalized.

**🔸 Evaluation Metrics:**

* **R² Score:** Low (~0.1 or less)
* **p-value:** > 0.05 (Not statistically significant)
* **Visual Check:** Predictions didn’t align closely with real unemployment values.

----------------Intermediate Requirements (20%) -----------------

Feature Engineering: -

To enhance predictive capabilities and reduce skewness in GDP values, the following feature transformation was applied:

Code: -  **df['Log\_GDP'] = np.log(df['GDP\_per\_capita'] + 1)**

This made GDP values more normally distributed and helped with model stability.

GDP Category Binning: -

Code: -

**def gdp\_category(gdp):**

**if gdp < 5000:**

**return 'Low'**

**elif gdp < 20000:**

**return 'Medium'**

**else:**

**return 'High'**

**df['GDP\_Category'] = df['GDP\_per\_capita'].apply(gdp\_category)**

-This created a new categorical variable for use in boxplots and grouped analysis.

Feature Visualization: -

1)Boxplot by GDP Category:

Code: -

**sns.boxplot(x='GDP\_Category', y='Unemployment', data=df)**

-Countries in the 'Medium' GDP range surprisingly had higher variability in unemployment.

-Helped uncover patterns not visible in raw data.

**2)**Regression Plot of Log(GDP):

Code: -

**sns.regplot(x='Log\_GDP', y='Unemployment', data=df)**

-Helped visualize linear trends after transformation.

-Smoother trend line than raw GDP.

3)Correlation Matrix:

Code: -

**sns.heatmap(df.corr(), annot=True)**

-Confirmed weak linear correlation between variables.

--------------------Advanced Requirements (10%) ----------------

**Predictive Model Building**

You implemented and compared three models:

Model & Implementation Summary

Linear Regression-Used OLS (statsmodels) and LinearRegression (sklearn). Weak slope and high p-value.

Polynomial (deg 2)- Used PolynomialFeatures to capture curvature. Slight improvement in R² visually.

Random Forest- Captured non-linearities, but overfit. Did not generalize well on test data.

**Model Evaluation**

1. Linear Regression:

-r2\_score: ~0.1

-p-value: > 0.05 → statistically insignificant

1. Polynomial Regression:

-Curve fit improved visually

-R² score still low; not enough improvement

1. Random Forest Regression:

-r2\_score on test data was lower than training

-Indicates **overfitting**, and lack of useful features

**Conclusion on Models**

* **None of the models showed strong predictive performance.**
* **GDP per capita alone is insufficient** to predict unemployment.
* Suggests that **multivariate analysis** with social, political, and labor-market features is needed for stronger predictive performance.

**All code, cleaned datasets, and final report are available on the public GitHub repository**:

<https://github.com/NomaanSaiyed/CS2704>