# The Happiness Equation: A Deep Dive into Predictors of Global Well-Being

Shubhan Mital (Student id A69044961)
Tamar Schaap (Student id A69031567)

December 13, 2025

# 1 Introduction

Understanding why some societies are happier than others has become an increasingly interdisciplinary question, drawing from economics, psychology, sociology, epidemiology, and environmental science. Since 2012, the World Happiness Report (WHR) has provided one of the most comprehensive global assessments of subjective well-being, helping governments and international agencies understand how social and economic conditions influence national happiness [15, 20]. Historically, economic factors, particularly GDP per capita, were considered primary indicators of happiness. However, research over the past decades shows that economic wealth alone does not fully explain national well-being, especially in wealthier countries where basic material needs are largely satisfied.

Psychological frameworks offer insight into this phenomenon. Maslow's hierarchy of needs [12] proposes that human well-being depends on the sequential satisfaction of needs ranging from physiological and safety requirements to social belonging, esteem, and self-actualization. Economic wealth primarily supports the fulfillment of basic needs, but beyond a certain point, higher income may not increase happiness and can even introduce stressors such as urban crowding, pollution, or social inequality [19]. This observation aligns with the Easterlin Paradox, which notes that increases in national income past a certain threshold do not necessarily lead to higher subjective well-being [3].

Happiness is influenced by multiple economic, social, cultural, environmental, and institutional factors. Economic variables such as unemployment, minimum wage policies, and inflation have measurable effects on well-being [2, 1, 8]. Social and cultural factors - including social support, autonomy, generosity, and perceptions of corruption - also play significant roles, though their effects are often moderated by cultural norms [6, 10, 18]. For instance, working towards shared goals and cultural cohesion are particularly salient in East Asian countries, while western societies prioritize autonomy and self-esteem [13].

Environmental and institutional conditions further shape happiness. Cleaner air, access to green spaces, and lower pollution levels positively correlate with well-being [11, 5], while education and healthcare systems provide foundational support for life satisfaction [17, 9, 21]. High-quality healthcare promotes both longevity and mental well-being, and education enhances social mobility, career prospects, and family satisfaction.

Given the multidimensional nature of happiness, comprehensive analysis requires integrating diverse indicators. Our study combines economic, social, institutional, demographic, and environmental variables, including GDP, unemployment, education enrollment, social support, life expectancy, $CO_2$ emissions, and more, to evaluate the relative contributions of these factors to national happiness. Previous work in this area has a narrower scope and mostly focuses on the social and economic features included in the WHR. We also employ correlation analysis machine learning (ML) methods and feature importance analyses to identify key predictors and provide interpretable insights into what drives societal well-being. By combining a broad range of variables with interpretable ML techniques, our work aims to fill gaps left by prior research that often focuses on limited indicators or lacks transparency regarding variable importance.

# 2 Methods

The country-specific dataset we use in this study is an aggregation of data from 4 different datasets: 1) the 2024 WHR [15, 20], 2) $CO_2$ emissions by country [14], 3) 2023 Global Country Information [4] and 4) World Population [16]. The variables from each of these datasets can be found in Appendix A. In order to investigate predictors of happiness, we 1) run a correlation analysis to understand which variables are correlated highly with happiness, 2) run supervised ML models, including a Linear regression, Random Forest regression, and AdaBoost regression, with happiness as the outcome to compare models and determine their predictive performance, and 3) run feature importance to determine which variables our highest-performing regressor found most important and compare those findings to our correlation analysis.

## 2.1 Pre-Processing

Relevant variables were selected from all datasets and renamed in cases where doing so would bring additional clarity to the analysis (for example, "Ladder score" was changed to "Happiness"). Country names were standardized since countries were often listed under different names in different datasets. Availability of information for each country varied between datasets. In order to keep all variables of interest, we are not including countries with missing data in this analysis. Figure 1 shows a world map of happiness scores of countries included in this analysis.

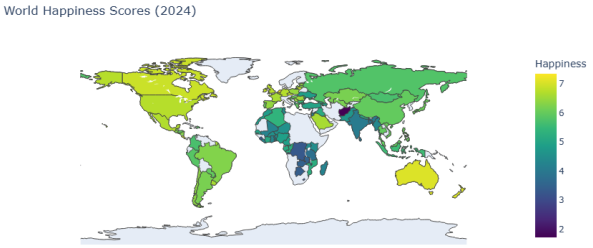


**Figure 1:** Worldwide Happiness Scores Included in this Paper (n = 100).

To reduce multicollinearity among highly correlated predictors, factor analysis was conducted. For brevity, we will not further describe the correlations from this pre-processing step. Specifically, a latent variable "Maternal Health" was created from "Maternal mortality ratio," "Infant mortality," and "Fertility Rate". Variance Inflation Factor (VIF) analysis before and after factor creation confirmed that multicollinearity was substantially reduced. Additionally, we removed "Total Tax Rate" from our analysis (VIF $> 10$) to maintain interpretability in feature importance analyses.

As the only categorical variable, "Regional indicator" was dummy encoded before running our ML models.

The dataset was divided into an 80% training set and a 20% test set to allow for unbiased model evaluation. All numeric variables were standardized using z-score transformations in the training set and those scales were applied to our test set. The formula for z-score calculations can be found below:

$$X_i' = \frac{X_i - \mu_X}{\sigma_X}$$

where $\mu_X$ is the mean and $\sigma_X$ is the standard deviation of feature $X$. This rescales all numeric variables to have $\mu$ 0 and $\sigma$ 1, making variables comparable and improving the performance of many ML algorithms.

# 3 Results

## 3.1 Correlation Analysis

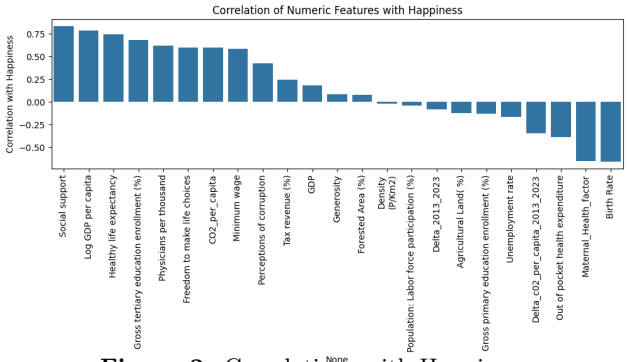


**Figure 2:** Correlations with Happiness

We calculated Pearson's correlation coefficient in order to determine all numeric variables' correlation strengths with happiness. Pearson's correlation coefficient values can range between -1 and 1; 0 would indicate no correlation, -1 would indicate a perfect negative correlation, and 1 would indicate a perfect positive correlation. The formula used to determine this value was:

$$r_{XY} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

where $X_i$ and $Y_i$ are individual observations of variables $X$ and $Y$, and $\bar{X}$ and $\bar{Y}$ are their respective means. The numerator in this formula is a measure of covariance while the denominator represents the combined variability of $X$ and $Y$, scaling the covariance by how much each variable varies on its own.

As shown in Figure 2, the 10 most highly-correlated features with happiness included social support, GDP per capita, life expectancy, participation in higher education, access to physicians, freedom to make life choices, $CO_2$ per capita, minimum wage, corruption, and tax revenue. This confirms our suspicions that the inclusion of a wider variety of measures, such as health-related, educational, and environmental measures, will offer insights into important correlates of happiness. Specific values of correlation coefficients can be found in Appendix B.

## 3.2 Machine Learning Regressors

### 3.2.1 Linear Regression

Linear regression models the relationship between the predictors and the target variable by fitting a straight line that minimizes the sum of squared errors. It assumes linear dependencies between features and the outcome and serves as an interpretable baseline for understanding feature effects.

### 3.2.2 Random Forest Regression

Random forest regression is an ensemble method that combines many decision trees, each trained on different bootstrapped samples of the data. The final prediction is the average of all trees, which reduces overfitting and captures nonlinear relationships between predictors and the target.

### 3.2.3 AdaBoost Regression

AdaBoost (Adaptive Boosting) regression builds a sequence of weak learners, typically shallow decision trees, where each learner focuses on the errors of the previous one. By iteratively reweighting difficult observations, AdaBoost creates a strong predictive model capable of capturing complex patterns.

### 3.2.4 Comparing Model Performance

After running each model, we compared performance metrics in order to determine which model was the best predictor of happiness. We determined the performance using MSE, RMSE, and $R^2$, the formulas for which are described below:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

where $y_i$ is the true value, $\hat{y}_i$ is the predicted value, and $n$ is the number of observations. It measures the average squared difference between predictions and true values.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

which is the square root of MSE and has the same units as the target variable $y$.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is the mean of the true values. $R^2$ represents the proportion of variance in $y$ explained by the model, with $R^2 = 1$ indicating a perfect fit and $R^2 = 0$ indicating no improvement over predicting the mean. In ML situations, $R^2$ can even be negative if the model performs poorly.

All models were evaluated against a baseline model that always predicted the mean of the training set. Using a baseline allows metrics such as MSE and RMSE, which lack intrinsic scales, to be understood relative to

a simple reference point. Figure 3 shows that the linear regression model outperforms other models since it has the highest $R^2$ value, and lowest MSE and RMSE values. However, all ML models outperform the baseline model, which always predicts the mean. Specific values of performance metrics can be found in Appendix C.

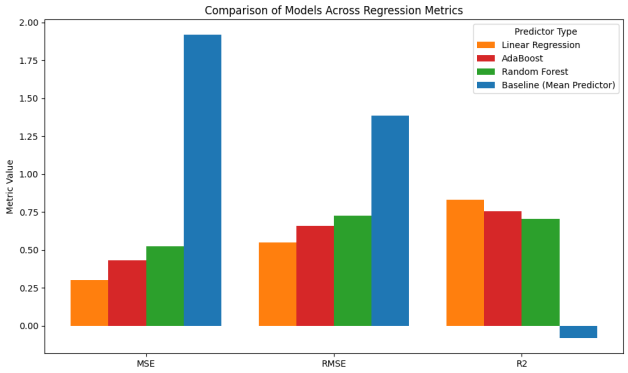


**Figure 3:** Performance Metrics for all Models.

The fact that linear regression outperformed other ML models indicates that using our predictors might have predominantly linear relationships between variables and our outcome. Figure 4 visualizes these relationships; it can be seen that while there are some seemingly non-linear relationships (including $CO_2$ variables, the maternal health factor that we created, and minimum wage), trends appear to be largely linear. Additionally, our relatively small sample size might make a linear regression a better fit for this data than other ML models.

## 3.3 Feature Importance

To evaluate the contribution of each predictor in the model, we calculated permutation-based feature importance based on our linear regression model. Permutation-based feature importance measures the decrease in model performance (we used the decrease in $R^2$) when the values of a feature are randomly shuffled; this shuffling breaks the relationship between that feature and the target variable. A larger decrease indicates more importance of that feature.

We also calculated 95% confidence intevals using the $\sigma$ of the permutation results. The formula for its calculation is defined below:

$$\text{CI}_{95\%} = \bar{I} \pm 1.96 \cdot \sigma_I$$

where $\bar{I}$ is the mean permutation importance of the feature, $\sigma_I$ is the standard deviation of the permutation importance, and 1.96 corresponds to the z-score for a 95% confidence interval. Figure 5 shows that the top ten variables that are important for predicting happiness are freedom to make life choices, social support, birth rate, unemployment rate, primary education, health expenses, changes in $CO_2$, and specific regions of world. Similar to our correlation analysis, these results indicate that a variety of variables contribute to Country Happiness. Specific Values from our feature importance calculations can be found in Appendix D.

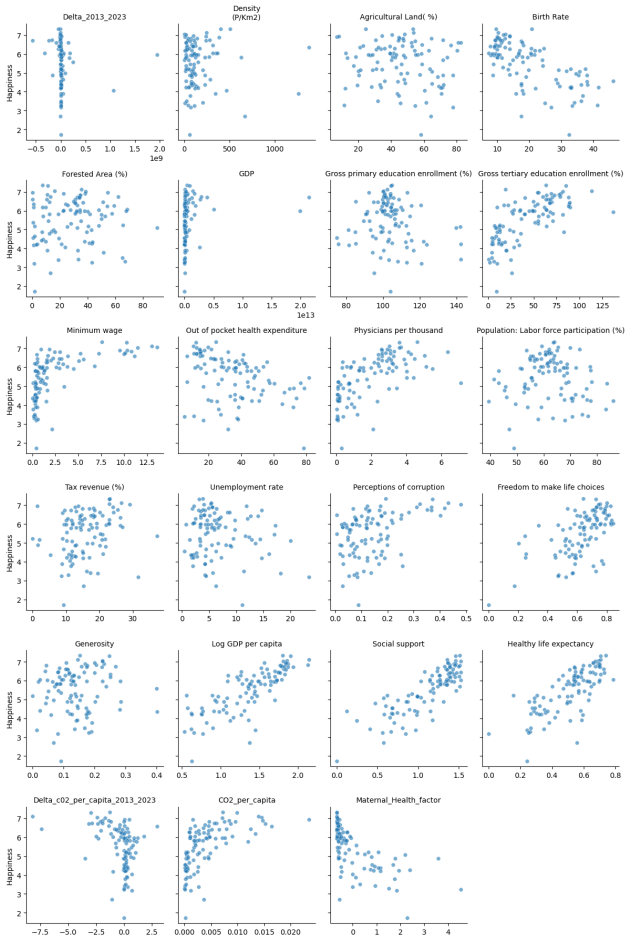


**Figure 4:** Relationships between Predictors and Happiness Visualized.

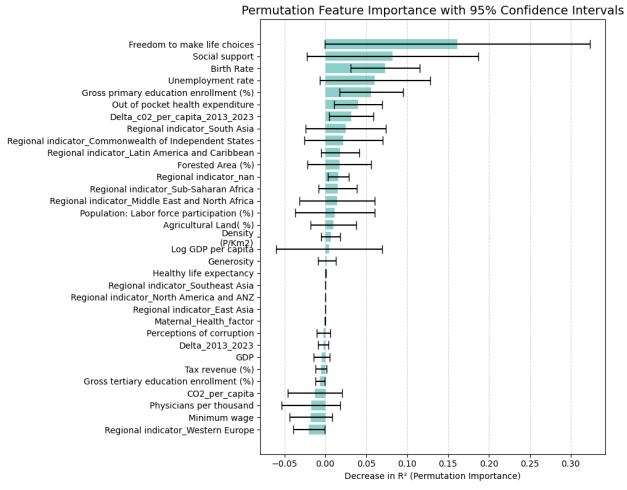


**Figure 5:** Importance of each Feature.

# 4 Discussion

Our analysis demonstrates that national happiness is influenced by a diverse set of economic, social, institutional, and environmental factors. Both the correlation and feature importance analyses highlight the multifaceted nature of well-being, confirming that happiness is not determined by only economic and social determinants. Consistent with prior research, social support and freedom to make life choices emerged as consistently strong predictors of happiness [6, 7]. This aligns with the notion that higher-order psychological needs, such as belonging and autonomy, significantly shape subjective well-being, particularly in countries where basic material needs are largely satisfied.

Economic factors, including GDP per capita, unemployment rate, and minimum wage, also contributed to happiness, though their influence was less consistent. These inconsistencies could reflect that economic wealth remains important for fulfilling basic physiological and safety needs, consistent with Maslow's hierarchy [12], but that beyond a certain threshold, additional wealth appears to have diminishing returns on well-being. This underscores the limitations of using economic wealth indicators alone as a policy measure for improving national happiness.

Environmental variables, namely $CO_2$, also played a meaningful role. $CO_2$ emissions predicted happiness, but not in the expected direction. $CO_2$ per capita was positively related to happiness. This unexpected finding might be explained by the fact that $CO_2$ is also indirectly related to economic growth. For example, metropolitan cities with high concentrations of jobs and people generally have higher total $CO_2$ emissions due to industrial activity, transportation, and energy consumption.

Institutional factors, particularly healthcare and education, contributed substantially to national well-being. Access to quality healthcare enhances both longevity and mental health, while educational opportunities foster social mobility, economic security, and life satisfaction [17, 9]. Our findings support the idea that strong healthcare and educational institutions provide a foundation for human flourishing.

Regional differences also turned out to be predictive of happiness. Specifically, countries classified as belonging to South Asia, Commonwealth of Independent States, and Latin America/Caribbean turned out to be important predictors in the feature importance analysis. One possible explanation is that these regions tend to have lower average income levels and fewer economic and social opportunities for citizens. This pattern is consistent with Maslow's hierarchy of needs [12], which suggests that basic physiological needs, which are directly related to wealth, must be met before individuals can prioritize higher-order needs such as social and educational pursuits.

Methodologically, our study demonstrates the utility of combining traditional statistical approaches with interpretable ML techniques. Both correlation and feature importance analyses provide insights into drivers of happiness.Our findings indicate that a broad range of indicators are associated with happiness, suggesting more diverse avenues for interventions aimed at improving well-being across countries. This approach addresses limitations in previous work that often relied solely on WHR indicators or employed black-box models with limited interpretability.

In summary, our results underscore that happiness is a multidimensional construct shaped by a combination of economic security, social connections, institutional strength, and cultural context. Additionally, while we investigated various environmental features, we could not find evidence that environmental features directly influenced happiness. Policy interventions aimed at improving national well-being should therefore adopt a holistic approach, simultaneously addressing economic, social and institutional factors rather than focusing narrowly on economic and social determinants. Future research could expand this work by exploring longitudinal dynamics of happiness, incorporating additional cultural measures, and testing the causal impact of targeted interventions on well-being.

Note: All code used for this project (as well as some additional exploratory analyses) can be found in Appendix E.

# References


[1] D. A. Akdoğan and Ş. Kızılarslan. "Unraveling the causal link between corruption and happiness: Insights from developing and advanced economies". In: *Panoeconomicus* 00 (2025), pp. 31–31.

[2] S. C. Carr. "Minimum Wage". In: *Wage and Well-being: Toward Sustainable Livelihood*. Cham: Springer International Publishing, 2023, pp. 117–146.

[3] R. A. Easterlin et al. "The happiness–income paradox revisited". In: *Proceedings of the National Academy of Sciences* 107.52 (2010), pp. 22463–22468.

[4] Nidula Elgiriyewithana. *Countries of the World 2023*. Accessed: 2025-12-11. Kaggle, 2023. URL: `https://www.kaggle.com/datasets/nelgiriyewithana/countries-of-the-world-2023`.

[5] D. Gao et al. "The Beneficial Elements in Forest Environment Based on Human Health and Well-Being Perspective". In: *Forests* 15.9 (2024), p. 1604.

[6] J. F. Helliwell et al. *World Happiness Report 2021*. Sustainable Development Solutions Network, 2021.

[7] J. Holt-Lunstad. "Loneliness and social isolation as risk factors: The power of social connection in prevention". In: *American Journal of Lifestyle Medicine* 15.5 (2021), pp. 567–573.

[8] V. Karadjova and A. Trajkov. "Basic economic indicators and economic well-being". In: *Horizons International Scientific Journal Series A Social Sciences and Humanities* 31 (2022), pp. 165–181.

[9] S. H. Keng and S. Y. Wu. "Living happily ever after? The effect of Taiwan's National Health Insurance on the happiness of the elderly". In: *Journal of Happiness Studies* 15.4 (2014), pp. 783–808.

[10] Q. Li and L. An. "Corruption takes away happiness: Evidence from a cross-national study". In: *Journal of Happiness Studies* 21.2 (2020).

[11] B. Ma et al. "Effects of urban green spaces on residents' well-being". In: *Environment, Development and Sustainability* 21.6 (2019), pp. 2793–2809.

[12] A. H. Maslow. "A theory of human motivation". In: *Psychological Review* 50.4 (1943), p. 370.

[13] M. J. Monnot and T. A. Beehr. "The good life versus the "goods life": An investigation of goal contents theory and employee subjective well-being across asian countries". In: *Journal of Happiness Studies* 23.3 (2022), pp. 1215–1244.

[14] Hannah Ritchie, Pablo Rosado, and Max Roser. *Annual CO_2 Emissions per Country*. Accessed: 2025-12-11. Our World in Data, 2025. URL: https://ourworldindata.org/grapher/annual-co2-emissions-per-country.

[15] Jaina Ru. *World Happiness Report 2024 (Yearly Updated)*. Accessed: YYYY-MM-DD. 2024. URL: https://www.kaggle.com/datasets/jainaru/world-happiness-report-2024-yearly-updated.

[16] Chaitya Shah. *World Population by Country (1960–2024)*. Accessed: 2025-12-11. Kaggle, 2025. URL: https://www.kaggle.com/datasets/chaitya07/world-population-by-country-19602024.

[17] S. Singh, S. Kshtriya, and R. Valk. "Health, hope, and harmony: a systematic review of the determinants of happiness across cultures and countries". In: *International Journal of Environmental Research and Public Health* 20.4 (2023), p. 3306.

[18] Y. Uchida and Y. Ogihara. "Bunkateki kofukukan: bunkashinrigakuteki chimi to shourai he no tennbo [A Cultural View of Happiness: Findings and Futures from a Cultural Psychology Approach]". In: *Shinrigaku Hyoron [Japanese Psychological Review]* 55 (2012), pp. 26–24.

[19] Y. Uchida and J. Rappleye. "Happiness: A World Map". In: *An Interdependent Approach to Happiness and Well-Being*. Cham: Springer International Publishing, 2023, pp. 19–33.

[20] M. M. Ulkhaq and A. Adyatama. "Clustering countries according to the world happiness report 2019". In: *Engineering and Applied Science Research* 48.2 (2021), pp. 137–150.

[21] C. Wang. "Does Health Insurance Boost Subjective Well-being? Examining the Link in China through a National Survey". In: *Economics* 18.1 (2024), p. 20220071.

# A Variable Descriptions

| Variable Name | Description |
|---|---|
| Happiness | Life Satisfaction measured by 0-10 Cantril ladder scale. |
| Regional indicator | Geographic region of the country. |
| Perceptions of corruption | Degree to which corruption is perceived to exist in government and business. |
| Freedom to make life choices | Percentage of people reporting freedom in choosing how to live their lives. Sourced from the Gallup World Poll. |
| Generosity | Measure of charitable behavior and willingness to give. Calculated as the residual from regressing the percentage of people who have donated money to a charity in the past month on Log GDP per capita. |
| Log GDP per capita | Log-transformed gross domestic product per person. Sourced from World Development Indicators. |
| Social support | Percentage reporting access to friends/relatives they can rely on when needed. Sourced from the Gallup World Poll. |
| Healthy life expectancy | Expected years of life without major disease or injury. Sourced from WHO data. |
| Delta 2013 2023 | Absolute difference in $CO_2$ between 2013 and 2023. |
| Delta c02 per capita 2013 2023 | Difference in $CO_2$ per capita between 2013 and 2023. |
| Density (P/Km$^2$) | Population per square kilometer. |
| Agricultural land (%) | Percentage of land area used for agriculture. |
| Birth rate | Births per 1,000 people per year. |
| $CO_2$ emissions | Carbon dioxide emissions in tons. |
| Fertility rate | Average number of children born per woman. |
| Forested area (%) | Percentage of land covered by forest. |
| GDP | Gross domestic product (non–log-transformed). |
| Gross primary education enrollment (%) | Percentage enrolled in primary education. |
| Gross tertiary education enrollment (%) | Percentage enrolled in tertiary education. |
| Infant mortality | Deaths before age 1 per 1,000 live births. |
| Life expectancy | Average lifespan in years. |
| Maternal mortality ratio | Maternal deaths per 100,000 live births. |
| Minimum wage | Minimum wage in local currency. |
| Out-of-pocket health expenditure | Percentage of total health spending paid out-of-pocket. |
| Physicians per thousand | Number of physicians per 1,000 people. |
| Labor force participation (%) | Percentage of population participating in the labor force. |
| Tax revenue (%) | Tax revenue as a percentage of GDP. |
| Population | Total country population. |
| Total tax rate | Overall tax burden as % of commercial profits. |
| Unemployment rate | Percentage of labor force unemployed. |
| Emissions (kt) | Carbon dioxide emissions in kilotons. Used to compute $CO_2$ per capita for 2013 and 2023 and the decade-long change. |
| Population ($CO_2$ data) | Used 2013 and 2023 values with emissions data to compute $CO_2$ per capita and decade-long change. |

# B Correlation Analysis: Supplementary Information

| Variable Name | Correlation with Happiness |
|---|---|
| Social support | 0.833998 |
| Log GDP per capita | 0.787814 |
| Healthy life expectancy | 0.746813 |
| Gross tertiary education enrollment (%) | 0.684289 |
| Physicians per thousand | 0.620697 |
| Freedom to make life choices | 0.597743 |
| CO2 per capita | 0.596090 |
| Minimum wage | 0.585294 |
| Perceptions of corruption | 0.423679 |
| Tax revenue (%) | 0.246354 |
| GDP | 0.179066 |
| Generosity | 0.082384 |
| Forested Area (%) | 0.080228 |
| Density (P/Km$^2$) | -0.022873 |
| Population: Labor force participation (%) | -0.039506 |
| Total tax rate | -0.047598 |
| Delta 2013–2023 | -0.080687 |
| Agricultural Land (%) | -0.121842 |
| Gross primary education enrollment (%) | -0.133396 |
| Unemployment rate | -0.165969 |
| Delta CO2 per capita 2013–2023 | -0.348401 |
| Out of pocket health expenditure | -0.389858 |
| Maternal Health factor | -0.654500 |
| Birth Rate | -0.658499 |

# C Performance Metrics: Supplementary Information

| Model | MSE | RMSE | $R^2$ |
|---|---|---|---|
| Baseline (Mean Predictor) | 1.9184 | 1.3851 | -0.0826 |
| Linear Regression | 0.3017 | 0.5493 | 0.8297 |
| Random Forest | 0.5243 | 0.7241 | 0.7041 |
| AdaBoost | 0.4482 | 0.6695 | 0.7471 |

# D Feature Importance: Supplementary Information

| Variable Name | Importance | Std | Lower CI | Upper CI |
|---|---|---|---|---|
| Regional indicator_Western Europe | -1.9928e-02 | 0.009928 | -3.9388e-02 | -4.6882e-04 |
| Minimum wage | -1.7716e-02 | 0.013119 | -4.3428e-02 | 7.9966e-03 |
| Physicians per thousand | -1.7497e-02 | 0.018372 | -5.3506e-02 | 1.8513e-02 |
| CO2_per_capita | -1.2731e-02 | 0.016738 | -4.5538e-02 | 2.0076e-02 |
| Gross tertiary education enrollment (%) | -6.4119e-03 | 0.002934 | -1.2162e-02 | -6.6196e-04 |
| Tax revenue (%) | -5.2489e-03 | 0.003418 | -1.1949e-02 | 1.4511e-03 |
| GDP | -4.5920e-03 | 0.005012 | -1.4416e-02 | 5.2317e-03 |
| Delta_2013_2023 | -2.2748e-03 | 0.003249 | -8.6434e-03 | 4.0937e-03 |
| Perceptions of corruption | -1.9722e-03 | 0.004255 | -1.0313e-02 | 6.3682e-03 |
| Maternal_Health_factor | -1.4163e-04 | 0.000223 | -5.7931e-04 | 2.9606e-04 |
| Regional indicator_East Asia | 1.1102e-16 | 0.000000 | 1.1102e-16 | 1.1102e-16 |
| Regional indicator_North America and ANZ | 1.1102e-16 | 0.000000 | 1.1102e-16 | 1.1102e-16 |
| Regional indicator_Southeast Asia | 1.1102e-16 | 0.000000 | 1.1102e-16 | 1.1102e-16 |
| Healthy life expectancy | 2.1095e-04 | 0.000150 | -8.2547e-05 | 5.0445e-04 |
| Generosity | 1.8105e-03 | 0.005614 | -9.1936e-03 | 1.2815e-02 |
| Log GDP per capita | 4.5208e-03 | 0.033211 | -6.0573e-02 | 6.9615e-02 |
| Density (P/Km2) | 6.5208e-03 | 0.005909 | -5.0615e-03 | 1.8103e-02 |
| Agricultural Land (%) | 1.0003e-02 | 0.014295 | -1.8016e-02 | 3.8022e-02 |
| Population: Labor force participation (%) | 1.1403e-02 | 0.024814 | -3.7232e-02 | 6.0038e-02 |
| Regional indicator_Middle East and North Africa | 1.4648e-02 | 0.023508 | -3.1429e-02 | 6.0725e-02 |
| Regional indicator_Sub-Saharan Africa | 1.5005e-02 | 0.012008 | -8.5308e-03 | 3.8542e-02 |
| Regional indicator_nan | 1.6025e-02 | 0.006533 | 3.2213e-03 | 2.8829e-02 |
| Forested Area (%) | 1.7331e-02 | 0.019830 | -2.1537e-02 | 5.6199e-02 |
| Regional indicator_Latin America and Caribbean | 1.8052e-02 | 0.012037 | -5.5408e-03 | 4.1646e-02 |
| Regional indicator_Commonwealth of Independent States | 2.2308e-02 | 0.024314 | -2.5347e-02 | 6.9963e-02 |
| Regional indicator_South Asia | 2.5108e-02 | 0.025041 | -2.3973e-02 | 7.4189e-02 |
| Delta_c02_per_capita_2013_2023 | 3.1940e-02 | 0.013868 | 4.7601e-03 | 5.9121e-02 |
| Out of pocket health expenditure | 4.0293e-02 | 0.014970 | 1.0951e-02 | 6.9635e-02 |
| Gross primary education enrollment (%) | 5.6129e-02 | 0.019850 | 1.7222e-02 | 9.5035e-02 |
| Unemployment rate | 6.0625e-02 | 0.034350 | -6.7006e-03 | 1.2795e-01 |
| Birth Rate | 7.3444e-02 | 0.021537 | 3.1231e-02 | 1.1566e-01 |
| Social support | 8.2427e-02 | 0.053362 | -2.2164e-02 | 1.8702e-01 |
| Freedom to make life choices | 1.6134e-01 | 0.082558 | -4.7596e-04 | 3.2315e-01 |

# E Python Code

[]:
```
# ============================
# STEP 1 - IMPORT ALL LIBRARIES
# ============================
!pip install catboost
!pip install pycountry
#!pip install ydata-profiling
#from ydata_profiling import ProfileReport
# Core
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import pycountry
from scipy import stats

# Machine Learning
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor, AdaBoostRegressor
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.inspection import permutation_importance
from functools import reduce
from sklearn.decomposition import FactorAnalysis
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

[]:
```
# ============================
# STEP 2 - PROCESS INPUT DATA
# ============================

# CO2 data
co2_df = pd.read_csv(r"/content/annual-co2-emissions-per-country.csv",␣
↪encoding="cp1252")
# Fix the encoding issue in the column name
co2_df.rename(
    columns={
        'Entity': 'Country',
        'Annual COa¸¸ emissions': 'Annual CO2 emissions'
    },
    inplace=True
)
co2_pivot = co2_df.pivot(index='Country', columns='Year', values='Annual CO2␣
↪emissions')
co2_pivot['Delta_2013_2023'] = co2_pivot[2023] - co2_pivot[2013]
```

```
co2_delta_df = co2_pivot[[2013, 2023, 'Delta_2013_2023']].reset_index()

# World data
world_df = pd.read_csv(r"/content/world-data-2023.csv", encoding="utf-8")
world_df = world_df[['Country', 'Density\n(P/Km2)', 'Agricultural Land( %)',␣
↪'Birth Rate',
                     'Co2-Emissions', 'Fertility Rate',
                     'Forested Area (%)', 'GDP', 'Gross primary education␣
↪enrollment (%)',
                     'Gross tertiary education enrollment (%)', 'Infant␣
↪mortality',
                     'Maternal mortality ratio', 'Minimum wage',
                     'Out of pocket health expenditure', 'Physicians per␣
↪thousand',
                     'Population: Labor force participation (%)', 'Tax revenue␣
↪(%)',
                     'Population', 'Total tax rate', 'Unemployment rate']]
# Happiness data
hap_df = pd.read_csv(r"/content/World-happiness-report-2024.csv",␣
↪encoding="utf-8")
hap_df.rename(columns={'Country name': 'Country'}, inplace=True)
hap_df.rename(columns={'Ladder score': 'Happiness'}, inplace=True)
hap_df = hap_df[['Country', 'Happiness', 'Regional indicator', 'Perceptions of␣
↪corruption',
                 'Freedom to make life choices', 'Generosity', 'Log GDP per␣
↪capita', 'Social support',
                 'Healthy life expectancy']]

# Population Data
pop_df = pd.read_csv(r"/content/World-population-data.csv", encoding="utf-8")
pop_df = pop_df[['Country Name', '2013']]
pop_df.rename(columns={'Country Name': 'Country', '2013': 'Population_2013'},␣
↪inplace=True)
```

[]:
```
# ============================
# STEP 3 - MERGE AND PROCESS
# ============================

# Merge all data
dfs_to_merge = [
    co2_delta_df, world_df, hap_df, pop_df
]

# Merge on "Country"
world_merged_df = reduce(lambda left, right: pd.merge(left, right,␣
↪on='Country', how='outer'), dfs_to_merge)
```

```
print(world_merged_df.shape)

# Remove % signs from all string cells
world_merged_df = world_merged_df.map(
    lambda x: str(x).replace('%', '').replace(',', '').replace('$', '') if␣
 ↪isinstance(x, str) else x
)
# Convert columns based on content
for col in world_merged_df.columns:
    if col == "Regional Indicator": # since this was causing nan's by getting␣
 ↪rid of special characters
        continue

    # Convert column to string for testing, ignore NaN
    col_values = world_merged_df[col].dropna().astype(str)

    # Check if any cell contains a letter
    has_letters = col_values.str.contains('[A-Za-z]', regex=True).any()

    if not has_letters:
        # Safe to convert to float
        world_merged_df[col] = pd.to_numeric(world_merged_df[col],␣
 ↪errors='coerce')
    else:
        # Keep as categorical (string)
        world_merged_df[col] = world_merged_df[col].astype(str)

# Mapping different names of one country - Standardized names
country_map = {
    'Côte d\'Ivoire': 'Cote d\'Ivoire',
    'Democratic People\'s Republic of Korea': 'North Korea',
    'Congo (Kinshasa)': 'Democratic Republic of the Congo',
    'Congo Dem. Rep.': 'Democratic Republic of the Congo',
    'Democratic Republic of Congo': 'Democratic Republic of the Congo',
    'Congo Rep.': 'Republic of the Congo',
    'Congo (Brazzaville)': 'Republic of the Congo',
    'Congo': 'Republic of the Congo',
    'Republic of the Congo': 'Republic of the Congo',
    'United Republic of Tanzania': 'Tanzania',
    'Swaziland': 'Eswatini',
    'United States of America': 'USA',
    'United States': 'USA',
    'Bolivia (Plurinational State of)': 'Bolivia',
    'Venezuela (Bolivarian Republic of)': 'Venezuela',
    'The Bahamas': 'Bahamas',
    'Bahamas, The': 'Bahamas',
    'The Gambia': 'Gambia',
```

```
    'Gambia, The': 'Gambia',
    'Viet Nam': 'Vietnam',
    'Iran (Islamic Republic of)': 'Iran',
    'Itan, Islamix Rep.': 'Iran',
    'Republic of Korea': 'South Korea',
    'Hong Kong S.A.R. of China': 'Hong Kong',
    'Hong Kong SAR': 'Hong Kong',
    'Lao People\'s Democratic Republic': 'Laos',
    'Turkiye': 'Turkey',
    'Taiwan Province of China': 'Taiwan',
    'State of Palestine': 'Palestine',
    'Palestinian National Authority': 'Palestine',
    'United Kingdom of Great Britain and Northern Ireland': 'United Kingdom',
    'The former Yugoslav republic of Macedonia': 'North Macedonia',
    'Russian Federation': 'Russia',
    'Czechia': 'Czech Republic',
    'Micronesia (country)': 'Micronesia (Federated States of)',
    'Micronesia, Fed. Sts.': 'Micronesia (Federated States of)',
    'Micronesia Fed. Sts.': 'Micronesia (Federated States of)',
    'Micronesia': 'Micronesia (Federated States of)',
    'Macao SAR China': 'Macau',
    'Bonaire, Saint Eustatius and Saba': 'Bonaire Sint Eustatius and Saba',
    'Syrian Arab Republic': 'Syria',
    'State of Palestine': 'Palestine',
    'Palestinian National Authority': 'Palestine',
    'S': 'São Tomé and Príncipe',
    'Cabo Verde': 'Cape Verde',
    'Ivory Coast': 'Cote d\'Ivoire',
    'Bosnia and Herzegovina': 'Bosnia Herzegovina',
    'Egypt, Arab Rep.': 'Egypt',
    'Republic of Ireland': 'Ireland'
}

# Mapping to df
world_merged_df['Country'] = world_merged_df['Country'].replace(country_map)

# Collapsing rows
world_merged_df = (
    world_merged_df
    .groupby('Country', as_index=False)
    .first()
)
all_na_cols = world_merged_df.columns[world_merged_df.isna().all()].tolist()

# Creating C02 per capita delta
world_merged_df['2013_c02_per_capita'] = world_merged_df[2013] /␣
 ↪world_merged_df['Population_2013']
```

```
world_merged_df['2023_c02_per_capita'] = world_merged_df[2023] /␣
 ↪world_merged_df['Population']
world_merged_df['Delta_c02_per_capita_2013_2023'] =␣
 ↪world_merged_df['2023_c02_per_capita'] -␣
 ↪world_merged_df['2013_c02_per_capita']

# Getting rid of rows with missing values
world_merged_df.dropna(inplace=True)

# Creating variable CO2_per_capita
world_merged_df['CO2_per_capita'] = world_merged_df['Co2-Emissions'] /␣
 ↪world_merged_df['Population']

# Dropping extra intermediatry columns
world_merged_df.drop(columns=['Population_2013', 2013, 2023,␣
 ↪'2013_c02_per_capita', '2023_c02_per_capita', 'Co2-Emissions',␣
 ↪'Population'], inplace=True)
```

[]:
```
# ========================================
# STEP 4 - PREPROCESSING
# ========================================

# Make a working copy so original df remains untouched if needed
df = world_merged_df.copy()

clusters = {
    'Maternal_Health': [
        'Maternal mortality ratio',
        'Infant mortality',
        'Fertility Rate'
    ]
}
factor_scores = pd.DataFrame(index=df.index)
# Compute factors and drop original correlated variables
for cluster_name, variables in clusters.items():
    fa = FactorAnalysis(n_components=1, random_state=42)
    factor_scores[cluster_name + "_factor"] = fa.fit_transform(df[variables])
    df = df.drop(columns=variables)  # remove originals
df = pd.concat([df, factor_scores], axis=1)

numeric_df = df.select_dtypes(include='number')
vif_df = pd.DataFrame()
vif_df["feature"] = numeric_df.columns
vif_df["VIF"] = [
    variance_inflation_factor(numeric_df.values, i)
    for i in range(numeric_df.shape[1])
]
```

```
vif_df.sort_values(by="VIF", ascending=False)
if 'Regional indicator' in df.columns:
    df = pd.get_dummies(df, columns=['Regional indicator'], drop_first=False)
world_merged_df = df.copy()

#dropping Total Tax Rate due to VIF > 10
world_merged_df = world_merged_df.drop(columns=['Total tax rate'])
```

[]:
```
# ============================
# STEP 5 - AUTO EDA
# ============================

report = ProfileReport(world_merged_df, title="Auto EDA Report",␣
 ↪explorative=True)
report.to_file("eda_report.html")
files.downloa('eda_report.html')
```

[]:
```
# ============================
# STEP 6 - MODELS
# ============================

# Remapping according to pycountry names
country_fix_map = {
    "Cote d'Ivoire": "Côte d'Ivoire",
    "Democratic Republic of the Congo": "Congo, The Democratic Republic of the",
    "Russia": "Russian Federation",
    "Turkey": "Türkiye",
}
# Apply fixes
world_merged_df['Country'] = world_merged_df['Country'].replace(country_fix_map)
# Create ISO-3 code column
def get_iso3(country_name):
    try:
        return pycountry.countries.lookup(country_name).alpha_3
    except:
        return None
world_merged_df['iso_alpha'] = world_merged_df['Country'].apply(get_iso3)

# Check if any countries failed
missing_iso = world_merged_df[world_merged_df['iso_alpha'].isna()]['Country'].
 ↪tolist()
print("Countries without ISO-3 code:", missing_iso)
# Create choropleth using ISO-3 codes
fig = px.choropleth(
    world_merged_df,
    locations='iso_alpha',  # use ISO-3 codes
    color='Happiness',
```

```
    hover_name='Country',
    color_continuous_scale='Viridis',
    title='World Happiness Scores (2024)'
)
fig.update_layout(geo=dict(showframe=False, showcoastlines=True))
fig.show()
# Select only numeric columns
numeric_cols = world_merged_df.select_dtypes(include='number').columns
# Compute correlations with Happiness (numeric only)
corr_with_happiness = world_merged_df[numeric_cols].corr()['Happiness'].
 ↪sort_values(ascending=False)
print(corr_with_happiness)
# Plot correlations minus Happiness itself
corr_with_happiness = corr_with_happiness.drop('Happiness')
plt.figure(figsize=(10,6))
sns.barplot(x=corr_with_happiness.index, y=corr_with_happiness.values)
plt.xticks(rotation=90)
plt.ylabel('Correlation with Happiness')
plt.title('Correlation of Numeric Features with Happiness')
plt.tight_layout()
plt.show()

# Prepare data
# Dummy code 'Regional indicator' (drop_first=True to avoid multicollinearity)
world_merged_df_encoded = world_merged_df.copy()
# Separate target
y = world_merged_df_encoded['Happiness']
# Select numeric + dummy features
numeric_features = world_merged_df.select_dtypes(include='number').columns.
 ↪tolist()
numeric_features.remove('Happiness')  # Target
dummy_features = [col for col in world_merged_df_encoded.columns if col not in␣
 ↪numeric_features + ['Happiness'] + ['Country'] + ['iso_alpha']]
X = world_merged_df_encoded[numeric_features + dummy_features]

# Train-test split and Standardize numeric features (z-score) for Linear␣
 ↪Regression only
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
scaler = StandardScaler()
X_train_scaled = X_train.copy()
X_test_scaled = X_test.copy()
X_train_scaled[numeric_features] = scaler.
 ↪fit_transform(X_train[numeric_features])
X_test_scaled[numeric_features] = scaler.transform(X_test[numeric_features])
```

```
# Fit models
models = {
    "Baseline (Mean Predictor)": None,
    "Linear Regression": LinearRegression(),
    # Random Forest: limit depth and min samples to avoid overfitting
    "Random Forest": RandomForestRegressor(
        n_estimators=200,
        max_depth=3,
        min_samples_split=5,
        min_samples_leaf=2,
        random_state=42
    ),
    # AdaBoost: use small trees as base estimator and adjust learning rate
    "AdaBoost": AdaBoostRegressor(
        n_estimators=200,
        learning_rate=0.5,
        random_state=42
    )
}
trained_models = models
results = []
for name, model in models.items():
    if name == "Baseline (Mean Predictor)":
        y_pred = np.full_like(y_test, y_train.mean(), dtype=np.float64)
    elif name == "Linear Regression":
        model.fit(X_train_scaled, y_train)
        y_pred = model.predict(X_test_scaled)
        lr_model = model
    else:  # Tree-based models: use unscaled data
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_test, y_pred)
    results.append({"Model": name, "MSE": mse, "RMSE": rmse, "R2": r2})
    print(f"{name} --> MSE: {mse:.4f}, RMSE: {rmse:.4f}, R2: {r2:.4f}")
results_df = pd.DataFrame(results).sort_values(by='MSE')
print("\nModel performance (sorted by MSE):")
print(results_df)
# Predictor types (models)
models = results_df['Model'].tolist()
metrics = ['MSE', 'RMSE', 'R2']
values = results_df[metrics].values.T
x = np.arange(len(metrics))
width = 0.2
colors = {'Baseline (Mean Predictor)':'#1f77b4',
          'Linear Regression':'#ff7f0e',
```

```
          'Random Forest':'#2ca02c',
          'AdaBoost':'#d62728'}
fig, ax = plt.subplots(figsize=(10,6))
for i, model in enumerate(models):
    ax.bar(x + i*width - width*1.5, values[:, i], width, label=model,␣
 ↪color=colors[model])
ax.set_xticks(x)
ax.set_xticklabels(metrics)
ax.set_ylabel('Metric Value')
ax.set_title('Comparison of Models Across Regression Metrics')
ax.legend(title='Predictor Type')
plt.tight_layout()
plt.show()
# Compute permutation importance using Random Forest
perm = permutation_importance(
    trained_models['Random Forest'],
    X_test,
    y_test,
    n_repeats=30,
    random_state=42,
    scoring='r2'
)
perm_df = pd.DataFrame({
    "Feature": X_train.columns,
    "Importance": perm.importances_mean,
    "Std": perm.importances_std
})
# Calculate 95% CI
perm_df["CI_lower"] = perm_df["Importance"] - 1.96 * perm_df["Std"]
perm_df["CI_upper"] = perm_df["Importance"] + 1.96 * perm_df["Std"]
perm_df = perm_df.sort_values("Importance", ascending=True)  # ascending for␣
 ↪horizontal bars
plt.figure(figsize=(10, 8))
plt.barh(
    perm_df["Feature"],
    perm_df["Importance"],
    xerr=1.96*perm_df["Std"],
    color="#76c7c0",
    alpha=0.85,
    capsize=5
)
plt.grid(axis='x', linestyle='--', alpha=0.6)
plt.xlabel("Decrease in R² (Permutation Importance)")
plt.title("Permutation Feature Importance with 95% Confidence Intervals",␣
 ↪fontsize=14)
plt.tight_layout()
plt.show()
```

```
# Create a table with Feature, Importance, Std, CI_lower, CI_upper
importance_table = perm_df[["Feature", "Importance", "Std", "CI_lower",␣
↪"CI_upper"]]
print(importance_table.to_string(index=False))
```

[]:
```
# ============================================================================
# STEP 7 - PROBABILITY & STATISTICS ANALYSIS
# ============================================================================

df_stats = world_merged_df.copy()

# ----------------------------------------------------------------------------
# Region Reconstruction
# ----------------------------------------------------------------------------
one_hot_cols = [c for c in df_stats.columns if c.startswith('Regional␣
↪indicator_')]

def decode_region(row):
    for col in one_hot_cols:
        if row[col] == 1:
            return col.replace('Regional indicator_', '')
    return "Unknown"

df_stats['Region'] = df_stats.apply(decode_region, axis=1)
df_stats = df_stats.drop(columns=one_hot_cols)

print("Columns available:", df_stats.columns.tolist(), "\n")

X = df_stats['Happiness'].dropna()


# ============================================================================
# 7.1 BASIC PROBABILITY
# ============================================================================
threshold = 6.0
prob_happy_gt = np.mean(X > threshold)
print(f"\n7.1 Probability → P(Happiness > {threshold}) = {prob_happy_gt:.4f}")

# --- Interpretation ---
print("Interpretation:")
print(f"- Fraction of countries with Happiness above 6.")
print(f"- About {prob_happy_gt*100:.2f}% of countries score highly on Happiness.
↪\n")


# ============================================================================
```

```
# 7.2 NORMAL FIT + QQ PLOT
# ============================================================================
h = X.values
mu_h, sigma_h = stats.norm.fit(h)
print(f"7.2 Normal Fit → Happiness = Normal(mu={mu_h:.3f}, sigma={sigma_h:.
 ↪3f})")

plt.figure(figsize=(8,4))
plt.hist(h, bins=25, density=True, alpha=0.5)
xx = np.linspace(h.min(), h.max(), 200)
plt.plot(xx, stats.norm.pdf(xx, mu_h, sigma_h), lw=2)
plt.title('Happiness Distribution & Fitted Normal')
plt.show()

plt.figure(figsize=(6,5))
stats.probplot(h, dist="norm", plot=plt)
plt.title("QQ Plot")
plt.show()

# --- Interpretation ---
print("Interpretation:")
print("- The Normal curve overlaid on the histogram shows how well Happiness␣
 ↪fits a Gaussian.")
print("- QQ plot alignment = stronger Normality.\n")


# ============================================================================
# 7.3 GDP-HAPPINESS CORRELATION
# ============================================================================
df_tmp = df_stats[['Log GDP per capita','Happiness']].dropna()

plt.figure(figsize=(6,5))
plt.scatter(df_tmp['Log GDP per capita'], df_tmp['Happiness'], alpha=0.6)
plt.xlabel('Log GDP per capita')
plt.ylabel('Happiness')
plt.title('Happiness vs GDP')
plt.grid(alpha=0.3)
plt.show()

rho, p_rho = stats.spearmanr(df_tmp['Log GDP per capita'], df_tmp['Happiness'])
pearson_r, p_pearson = stats.pearsonr(df_tmp['Log GDP per capita'],␣
 ↪df_tmp['Happiness'])

print(f"7.3 Correlation → Spearman rho={rho:.3f}, p={p_rho:.3e}")
print(f"                  Pearson r={pearson_r:.3f}, p={p_pearson:.3e}")

# --- Interpretation ---
```

```
print("Interpretation:")
print("- Positive correlation: richer countries tend to be happier.")
print("- Pearson = linear strength, Spearman = monotonic strength.")
print("- Very small p-values → strong statistical significance.\n")


# ================================================================================
# 7.4 CHEBYSHEV INEQUALITY
# ================================================================================
mu_x = X.mean()
sigma_x = X.std(ddof=0)
k = 2
cheb_bound = 1.0 / (k**2)
empirical = np.mean(np.abs(X - mu_x) >= k * sigma_x)

print(f"7.4 Chebyshev → bound={cheb_bound:.3f}, empirical={empirical:.3f}")

# --- Interpretation ---
print("Interpretation:")
print("- Chebyshev gives a universal bound for ANY distribution.")
print("- Empirical proportion < bound → Chebyshev holds easily.\n")


# ================================================================================
# 7.5 CENTRAL LIMIT THEOREM (BOOTSTRAP SAMPLE MEANS)
# ================================================================================
sample_size = 30
n_experiments = 2000

means = np.array([
    X.sample(sample_size, replace=True, random_state=i).mean()
    for i in range(n_experiments)
])

plt.figure(figsize=(7,4))
plt.hist(means, bins=30, density=True, alpha=0.6)
xx = np.linspace(means.min(), means.max(), 200)
plt.plot(xx, stats.norm.pdf(xx, means.mean(), means.std()), lw=2)
plt.title('CLT: Sample Means')
plt.show()

print(f"7.5 CLT → mean={means.mean():.4f}, std={means.std():.4f}")

# --- Interpretation ---
print("Interpretation:")
print("- Distribution of sample means looks Normal (CLT in action).")
print("- Even if raw data isn't perfectly Normal, means converge.\n")
```

```
# ============================================================================
# 7.6 EMPIRICAL MGF
# ============================================================================
t_values = [-0.2, -0.1, 0.0, 0.1, 0.2]
mgf_values = {t: np.mean(np.exp(t * X)) for t in t_values}

print("7.6 MGF Estimates:")
for t, v in mgf_values.items():
    print(f"  M({t}) = {v:.4f}")

# --- Interpretation ---
print("Interpretation:")
print("- MGF values characterize all moments of the distribution.")
print("- Shows how E[e^{tX}] behaves for positive/negative t.\n")


# ============================================================================
# 7.7 MAXIMUM LIKELIHOOD ESTIMATION
# ============================================================================
mu_mle = X.mean()
sigma_mle = X.std(ddof=0)

print(f"7.7 Normal MLE → mu={mu_mle:.4f}, sigma={sigma_mle:.4f}")

if (X > 0).all():
    print(f"Exponential MLE → lambda={1/X.mean():.4f}")
else:
    print("Exponential MLE → skipped (non-positive values)")

# --- Interpretation ---
print("Interpretation:")
print("- For a Normal distribution: mu(hat)= sample mean, sigma(hat)= sample␣
 ↪std.")
if (X > 0).all():
    print("- For exponential: lambda(hat)= 1 / mean.")
print()


# ============================================================================
# 7.8 95% CONFIDENCE INTERVAL
# ============================================================================
mean_h = X.mean()
sem_h = stats.sem(X)
ci_low, ci_high = stats.t.interval(0.95, df=len(X)-1, loc=mean_h, scale=sem_h)
```

```
print(f"7.8 95% CI → [{ci_low:.4f}, {ci_high:.4f}]")

# --- Interpretation ---
print("Interpretation:")
print("- 95% of intervals constructed this way would contain the true mean.")
print("- CI provides a range estimate for global Happiness.\n")


# ================================================================================
# 7.9 REGION-BASED T-TEST (TOP 2 REGIONS)
# ================================================================================
region_counts = df_stats['Region'].value_counts()
top2 = region_counts.index[:2]

g1 = df_stats.loc[df_stats['Region'] == top2[0], 'Happiness'].dropna()
g2 = df_stats.loc[df_stats['Region'] == top2[1], 'Happiness'].dropna()

t_stat, p_val = stats.ttest_ind(g1, g2, equal_var=False)
print(f"7.9 T-test → {top2[0]} vs {top2[1]}: t={t_stat:.3f}, p={p_val:.3e}")

# --- Interpretation ---
print("Interpretation:")
print("- Tests if the two most common regions differ in Happiness.")
print("- Small p-value → significant regional difference.\n")


# ================================================================================
# 7.10 Z-TEST AGAINST 5.5
# ================================================================================
pop_ref = 5.5
z_stat = (X.mean() - pop_ref) / (X.std(ddof=1) / np.sqrt(len(X)))
p_z = 2 * (1 - stats.norm.cdf(abs(z_stat)))

print(f"7.10 Z-test → Z={z_stat:.3f}, p={p_z:.3e}")

# --- Interpretation ---
print("Interpretation:")
print("- Tests whether world mean Happiness = 5.5.")
print("- Small p → mean differs significantly from 5.5.\n")


# ================================================================================
# 7.11 CHI-SQUARE TEST (REGION × CORRUPTION)
# ================================================================================
tmp = df_stats[['Region', 'Perceptions of corruption']].dropna()
tmp['HighCorruption'] = (tmp['Perceptions of corruption'] >
                         tmp['Perceptions of corruption'].median()).astype(int)
```

```
ct = pd.crosstab(tmp['Region'], tmp['HighCorruption'])
chi2, p_chi, dof, exp = stats.chi2_contingency(ct)

print(f"7.11 Chi-square → chi2={chi2:.3f}, p={p_chi:.3e}")

# --- Interpretation ---
print("Interpretation:")
print("- Tests independence between Region and Corruption.")
print("- Significant p → corruption varies meaningfully by region.\n")


# ================================================================================
# 7.12 BONFERRONI MULTIPLE TESTING
# ================================================================================
features = [
    'Log GDP per capita',
    'Social support',
    'Healthy life expectancy',
    'Freedom to make life choices',
    'Generosity'
]

pvals, names = [], []

for feat in features:
    df_tmp = df_stats[[feat, 'Happiness']].dropna()
    r, p = stats.pearsonr(df_tmp[feat], df_tmp['Happiness'])
    pvals.append(p)
    names.append(feat)

corrected = np.minimum(np.array(pvals) * len(pvals), 1.0)

print("7.12 Bonferroni Correction:")
for f, p_orig, p_corr in zip(names, pvals, corrected):
    print(f"  {f}: p={p_orig:.3e}, corrected={p_corr:.3e}")

# --- Interpretation ---
print("Interpretation:")
print("- Bonferroni guards against false positives from multiple tests.")
print("- Features with small corrected p remain truly significant.\n")

print("STEP 7 complete.\n")
```

```
Columns available: ['Country', 'Delta_2013_2023', 'Density\n(P/Km2)',
'Agricultural Land( %)', 'Birth Rate', 'Forested Area (%)', 'GDP', 'Gross
primary education enrollment (%)', 'Gross tertiary education enrollment (%)',
'Minimum wage', 'Out of pocket health expenditure', 'Physicians per thousand',
```

```
'Population: Labor force participation (%)', 'Tax revenue (%)', 'Unemployment
rate', 'Happiness', 'Perceptions of corruption', 'Freedom to make life choices',
'Generosity', 'Log GDP per capita', 'Social support', 'Healthy life expectancy',
'Delta_c02_per_capita_2013_2023', 'CO2_per_capita', 'Maternal_Health_factor',
'iso_alpha', 'Region']


7.1 Probability → P(Happiness > 6.0) = 0.4000
Interpretation:
- Fraction of countries with Happiness above 6.
- About 40.00% of countries score highly on Happiness.

7.2 Normal Fit → Happiness = Normal(mu=5.496, sigma=1.162)

Interpretation:
- The Normal curve overlaid on the histogram shows how well Happiness fits a
Gaussian.
- QQ plot alignment = stronger Normality.


7.3 Correlation → Spearman rho=0.817, p=3.924e-25
                  Pearson r=0.788, p=2.394e-22
Interpretation:
- Positive correlation: richer countries tend to be happier.
- Pearson = linear strength, Spearman = monotonic strength.
- Very small p-values → strong statistical significance.

7.4 Chebyshev → bound=0.250, empirical=0.020
Interpretation:
- Chebyshev gives a universal bound for ANY distribution.
- Empirical proportion < bound → Chebyshev holds easily.


7.5 CLT → mean=5.5022, std=0.2107
Interpretation:
- Distribution of sample means looks Normal (CLT in action).
- Even if raw data isn't perfectly Normal, means converge.

7.6 MGF Estimates:
  M(-0.2) = 0.3428
  M(-0.1) = 0.5812
  M(0.0) = 1.0000
  M(0.1) = 1.7439
  M(0.2) = 3.0793
Interpretation:
- MGF values characterize all moments of the distribution.
- Shows how E[e^{tX}] behaves for positive/negative t.

7.7 Normal MLE → mu=5.4957, sigma=1.1617
```

```
Exponential MLE → lambda=0.1820
Interpretation:
- For a Normal distribution: mu(hat)= sample mean, sigma(hat)= sample std.
- For exponential: lambda(hat= 1 / mean.

7.8 95% CI → [5.2640, 5.7273]
Interpretation:
- 95% of intervals constructed this way would contain the true mean.
- CI provides a range estimate for global Happiness.

7.9 T-test → Sub-Saharan Africa vs Latin America and Caribbean: t=-11.160,
p=5.333e-13
Interpretation:
- Tests if the two most common regions differ in Happiness.
- Small p-value → significant regional difference.

7.10 Z-test → Z=-0.037, p=9.703e-01
Interpretation:
- Tests whether world mean Happiness = 5.5.
- Small p → mean differs significantly from 5.5.

7.11 Chi-square → chi2=14.935, p=1.344e-01
Interpretation:
- Tests independence between Region and Corruption.
- Significant p → corruption varies meaningfully by region.

7.12 Bonferroni Correction:
  Log GDP per capita: p=2.394e-22, corrected=1.197e-21
  Social support: p=4.724e-27, corrected=2.362e-26
  Healthy life expectancy: p=4.648e-19, corrected=2.324e-18
  Freedom to make life choices: p=5.173e-11, corrected=2.586e-10
  Generosity: p=4.151e-01, corrected=1.000e+00
Interpretation:
- Bonferroni guards against false positives from multiple tests.
- Features with small corrected p remain truly significant.

STEP 7 complete.
```