Global Mental Health and Eating Disorder Risk Prediction

Mandeep Saini, Dylan Scott-Dawkins, and Quang Tran

University of San Diego

SHILEY-MARCOS SCHOOL OF ENGINEERING

AAI-500-02-SU25, Probability and Statistics for Artificial Intelligence

Leon Schpaner

April 20, 2024

Abstract

Mental illness conditions impact individuals in our society to a large extent. Identification of associated conditions such as eating disorders is crucial for early intervention. This project delves into predictive associations between various psychiatric conditions; we are interested in bipolar disorder, anxiety, schizophrenia, and the incidence of eating disorders, using data from global mental health databases. An exploratory statistical data analysis was performed by applying univariate, bivariate, and multivariate statistics to uncover patterns and relationships. This study further examines how the quality of care a patient receives at healthcare relates to national depression levels. Different predictive models were experimented with, that is, generalized linear models, closest neighbor k, random forests, neural networks, and support vector regression. Among these, the Random Forest Regressor was the best with an R^2 value of 0.995 for the test set. It can be seen from the results that bipolar and schizophrenia disorders are the best predictors of eating disorders. In addition, the Swedish-United States case study highlighted the role of broader health system characteristics on outcomes in mental health. The study provides an evidence-based model for identifying at-risk groups for eating disorders and informs public health policy with the objective of improving outcomes in mental health.

Global Mental Health and Eating Disorder Risk Prediction

Introduction - Predicting Eating Disorders

Mental health is an essential part of people's lives and society, deeply influencing our well-being, ability to work, and relationships. It's also quite clear that mental health conditions are not uncommon, with hundreds of millions affected yearly, and a significant portion of the population experiencing major depression over their lifetimes, for example, an estimated 1 in 3 women and 1 in 5 men. While conditions like schizophrenia and bipolar disorder might be less common, their impact on individuals' lives remains substantial (IMT Kaggle Team, 2023). Given this widespread impact, our project aims to contribute to improving mental health outcomes by focusing on predicting eating disorders from other mental health conditions and characteristics within a data set. The core motivation behind this is the potential for earlier identification and thus more helpful intervention strategies. If we can predict the likelihood of an eating disorder, it opens doors for timely support, which is often crucial for better patient outcomes. This echoes the sentiment in the cervical cancer prediction study, which highlighted that understanding causes and interpreting screening tests are important for predictive modeling, ultimately to save lives through early detection

Also in this project, we will investigate the relationship between the quality of healthcare measured by the Universal Health Coverage (UHC) Index and depression rates across various countries. We aim to determine whether regions with strong healthcare systems experience lower rates of depression (World Health Organization, 2023). Additionally, we have utilized advanced data tools to focus on predicting potential eating disorders, which can be challenging to detect in the early stages. By identifying these risks, we can provide timely support, much like doctors use routine tests to detect diseases like cervical cancer early. Our goal is to integrate data and healthcare insights, creating mental health support that is fairer, faster, and more effective.

Exploratory Data Analysis (EDA)

Exploratory data analysis (EDA) is a crucial initial step in the data analysis process, with the aim of understanding the structure, quality, and overall characteristics of the data set. In this project, the main focus of the EDA was on the prevalence of mental disorders (dataset 1), with additional analysis conducted on the burden of disease caused by each disorder (dataset 2). The analysis process includes data cleaning, format standardization, handling missing data and outliers, and exploring relationships between variables using descriptive statistics and visualization methods (Agresti and Kateri, 2022).

Through tools such as box plots and heatmaps, EDA helps clarify important data patterns and correlations between types of mental disorders, as well as assess the completeness and stability of the information. Thorough data preparation during the exploratory data analysis (EDA) phase not only aids in selecting the appropriate model for the subsequent analysis steps but also ensures the reliability and accuracy of the final findings.

In addition to standard exploratory data analysis steps, such as data cleaning and handling missing values, the Universal Health Coverage (UHC) dataset was transformed from a wide format to a long format to facilitate time-series analysis. The mental health and UHC datasets were merged by aligning country names and years, resulting in a unified dataset for integrated analysis. Only records with complete UHC data were retained to ensure data quality. Furthermore, year-by-year analyses were conducted to examine the relationship between health coverage and mental health outcomes over time. Visualizations, including scatter plots and regression plots, were created to illustrate trends across countries and years, providing deeper insights beyond simple cross-sectional descriptions.

The Data Cleaning and Processing

Data cleaning and processing are performed systematically to prepare the data for EDA analysis (Agresti and Kateri, 2022).

Key steps:

- Download CSV datasets from GitHub using an automated script.
- Remove unnecessary columns, such as Code, that contain many empty values.
- Rename columns with long or complex titles to short and easy-to-understand names.

- Normalize column names, convert all letters to lowercase, and replace spaces with underscores.
- Remove extra spaces in string values.
- Explicitly convert data columns to numeric types using `pd.to_numeric` from the `pandas` library, treating invalid values as missing.
- Count and identify missing values.
- Reshape UHC data from wide to long format.
- Filter out rows with missing UHC values.
- Convert year columns to numeric and integer types.
- Merge datasets on entity and year.

Datasets Introduction

This dataset consists of 7 small datasets:

- **Dataset 1: Prevalence of Mental Illness** – This dataset serves as the primary foundation for our analysis, offering comprehensive information on the prevalence of various mental illnesses in different population groups, regions, and countries. It plays a central role in highlighting the occurrence of mental disorders, including direct statistics on eating disorders, which help identify prevalence rates and affected populations. The insights drawn from this dataset are essential for building predictive models that incorporate risk factors and the distribution of these conditions.
- **Dataset 2: Burden of Disease from Mental Illness** – This dataset presents information on the burden of disease caused by mental illness, typically measured in Disability-Adjusted Life Years (DALY). It measures the overall impact of mental illness on individual health, society, and the economy. Eating disorders have a profound impact on an individual's

quality of life, productivity, and overall health. This dataset helps quantify those impacts and improve analysis by linking prevalence data to health and social outcomes.

- **Datasets 3 and 4 (Adult Population Covered in Primary Data) –** These datasets focus on the coverage of research data rather than directly providing information on disease prevalence or impact. Therefore, they are not directly relevant to the analysis of eating disorders.
- **Dataset 5 (Anxiety Disorders Treatment Gap) –** Although relevant to mental health, this dataset focuses on anxiety disorders and access to treatment, so it is less directly relevant to eating disorders.
- **Dataset 6 (Depressive Symptoms in US Population) –** This dataset focuses on depressive symptoms, not specifically on eating disorders, so it is not directly relevant to the purpose of this study.
- **Dataset 7 (Countries with Primary Data on Mental Illnesses) –** This dataset focuses on data availability and collection mechanisms rather than providing detailed information about eating disorders or health impacts.
- **Universal Health Coverage (UHC) Dataset –** This dataset provides annual country-level scores that measure the accessibility and quality of essential health services worldwide. It was used to test whether countries with better healthcare systems exhibit lower rates of depression and other disorders. To analyze this relationship, we harmonized the datasets containing major depression data with the UHC dataset.
- **Note:** The dataset file is named `GDP.csv`, but it actually contains Universal Health Coverage (UHC) data used for analyzing healthcare accessibility.

Univariate Analysis

The univariate analysis in the project focuses mainly on handling missing data and outliers to ensure the integrity and reliability of the data before proceeding with further analysis.

- **Missing Values:**
  - *Column with empty values:* One of the first steps is to remove the `Code` column, as it contains many empty and missing values that do not provide useful information for the analysis. It is completely removed to reduce noise and simplify the data.
  - *Dataset with unrealistic data:* In dataset 4, many zero values are recorded in the prevalence columns of mental disorders. However, in reality, this rate is rarely exactly zero; the zero values here reflect the lack of original data or incomplete reporting, not the actual rate.
- **Outlier Treatment:**
  - *Interquartile Range (IQR) Method:* Values outside the range

$$Q_1 - 1.5 \times \text{IQR to } Q_3 + 1.5 \times \text{IQR}$$

    are considered outliers and are limited to the boundary value. This method helps to "flatten" unusual data points without distorting the overall distribution of the data.
  - *Z-score Method:* This method normalizes the data and removes values with a Z-score greater than ±3, corresponding to 99.7 percent of data values in the normal distribution. This allows for an effective comparison of the two outlier handling techniques.

After applying the above methods, the box plots before and after processing show that the data have been adjusted to remove outliers. In particular, the removal of outliers does not significantly affect the mean values of the variables, which shows that extreme values do not overly skew the data distribution , and the post-processed data still retain its representativeness for the entire data set.

Bivariate and Multivariate Analysis

The bivariate and multivariate analysis in this study focused on exploring the association between different types of mental disorders using two main data sets: data set 1 (population prevalence of each disorder) and data set 2 (burden of disease (DALY) caused by those disorders).

The two main quantitative analysis tools used:

- **Correlation Matrix:** Calculated using Pearson's coefficient, the correlation matrix reflects the degree of linear association between variables. Strong correlation values $|r| \geq 0.5$ are marked. The analysis results showed that pairs of disorders, such as anxiety disorders and depression, or eating disorders and bipolar disorder, were relatively highly correlated. This suggests that comorbidity is common in mental disorders, where one disorder may accompany or lead to another.
- **Covariance Matrix:** Complementary to correlation analysis, covariance shows the direction of covariance between two variables. Positive covariance values reinforce the positive association between disorders, which is particularly evident in dataset 2, where DALYs from disorders show a trend of increasing covariance.

Additionally, pairs of strongly correlated variables are extracted and displayed in a tabular format, facilitating the clear identification of relationships that should be considered in potential predictive models or when assessing the social impact of each disorder.

Additional Bivariate and Multivariate Analysis

Digging deeper into mental disorder associations, this section focuses on analyzing the relationship between mental health outcomes and healthcare accessibility. Specifically, we examined data from the mental health prevalence dataset alongside the Universal Health Coverage (UHC) service coverage index dataset.

Data Visualization

Data visualization is a core component of exploratory analysis and is particularly useful in identifying patterns and relationships within the two main datasets of the study.

The main visualizations used are:

- **Box plots:** Applied to each variable in both datasets to detect outliers before and after processing using the IQR and Z-score methods. Box plots not only help clarify the range of data distribution but also allow for a direct assessment of the impact of outliers on the stability of the data.

- **Heatmap:** A heatmap from the correlation and covariance matrices helps identify notable associations between disorders quickly. The colors in the plot represent the strength of the correlation, ranging from weak to strong, thereby providing a visual representation of the data's relationship structure. In dataset 1, the heatmap clearly shows that the prevalence of disorders such as anxiety, depression, and eating disorders tend to increase together over time or across geographic regions. Similarly, dataset 2 shows that the burden of disease due to these disorders also tends to fluctuate concurrently.

- **Scatterplots**: Scatterplots were used to examine the relationship between the Universal Health Coverage (UHC) Index and depression rates across countries and over time. Regression lines were added to these plots to illustrate trends clearly. Additionally, key countries such as the United States and Sweden were highlighted with distinct markers and labels to emphasize differences and outliers. These visualizations helped to communicate the findings clearly and supported interpretation of the statistical results.

Overall, the combination of these visualization tools not only helps analysts better understand the data but also effectively communicates the results to non-technical readers. The graphs helped to identify potential relationships between mental disorders early on, which are often difficult to demonstrate with raw data tables.

Train-Test Split

Train-Test Split is applied consistently to all models to evaluate the generalization ability when predicting unseen data.

Data Split: The dataset is split into two parts: **80% for training** and **20% for testing**, specifically:

- X_train: (5136, 4)
- X_test: (1284, 4)
- y_train: (5136,)
- y_test: (1284,)

This split method ensures that the training data is sufficiently large for the model to learn effectively while maintaining an independent test set to evaluate generalization after training.

- Train set: Used to train the model to learn the relationships between independent and dependent variables.
- Test set: Used only to predict and evaluate the accuracy of the model on new data.

Models applying train-test split:

- GLM (Generalized Linear Model): Apply train-test split and use R^2, MSE and 95% confidence interval for evaluation. No statistical comparison test is performed because this is the reference model.
- K-Nearest Neighbors Regressor (KNN): Trained on the train set and evaluated on the test set. Then, compared with the GLM model using p-value test, the result is $p = 0.393$, which means there is no statistically significant difference.
- Neural Network (MLP Regressor): After splitting the data, the model is trained with early stopping to avoid overfitting. Accuracy was assessed using R^2, MSE, and p-value vs. GLM ($p < 0.001$), demonstrating that this model is a significant improvement.

- Random Forest Regressor: Also used train-test split and gave a very high R^2. However, the p-value test with the GLM model yielded p = 0.241, indicating insufficient statistical evidence to conclude that the model is better.
- Support Vector Regressor (SVR): This model was assessed similarly with a very small p-value (1.17×10^{-35}), demonstrating a significant improvement over the linear model.

Splitting the data into training and testing sets plays an important role in ensuring objectivity when evaluating the model. This method helps avoid **overfitting** and allows fair comparison between different models. In addition, train-test split also helps verify the level of model improvement through metrics such as R^2, MSE, and especially the **p-value**, thereby objectively evaluating the effectiveness and generalization ability of the model.

Methodology - Model Selection

Linear Regression

Simple linear regression models were applied to explore the association between each type of mental disorder and eating disorders based on standardized data.

The selection of input variables for modeling the prevalence of eating disorders was based on both statistical analysis and the clinical characteristics of each mental disorder in Dataset 1 Prevalence of Mental Illness. In this analysis, we utilized correlation matrices and covariance matrices to evaluate the relationships between the disorders.

We selected three key predictors to build the model: Bipolar disorders, Anxiety disorders, and Schizophrenia disorders. This combination is both statistically robust and reflects clinical utility in predicting eating disorder risk from other psychiatric manifestations.

Next, we used linear regression models to examine the relationship between psychiatric disorders and the prevalence of eating disorders. Both simple linear regression and generalized linear regression (GLM) models in the next section were used to determine the influence and predictive ability of psychiatric factors such as bipolar disorders, anxiety disorders, and schizophrenia disorders.

Table 3.1: Correlation

Variable Pair	Correlation (r)	Relationship Strength
bipolar disorders and eating disorders	+0.68	Strong Correlation
anxiety disorders and eating disorders	+0.59	Moderate to Strong Correlation
schizophrenia disorders and eating disorders	+0.50	Moderate Correlation
depression disorders and eating disorders	–0.05	Not Significantly Correlated

Table 1
This suggests that bipolar, anxiety, and schizophrenia are positively and significantly correlated with eating disorders, while depression disorders do not appear to be directly related.

Table 3.2: Covariance

Variable Pair	Covariance	Interpretation
bipolar and eating disorders	0.0219	Increase together, units are quite similar
anxiety and eating disorders	0.0864	Large covariance means change together in larger scales
schizophrenia and eating disorders	0.0027	Correlation exists, but units of change are significantly different

Table 2
Although schizophrenia has a small covariance, it remains significant when combined with other variables in a generalized linear model (GLM) due to its independent effect.

Using linear regression model:

$$\text{Eating Disorders} = \beta_0 + \beta_1 X + \varepsilon \tag{1}$$

Where:

- X is one of three variables: bipolar disorders, anxiety disorders, or schizophrenia disorders
- β_0: is the intercept of the linear regression
- β_1: measures the degree of change in eating disorders as mental disorders change
- ε is the error term, accounting for random variation not explained by the model

Results:

Predictor	R^2	MSE	Interpretation
bipolar disorders	0.46	0.01	Relatively strong linear relationship. Approximately 46 percent of the variation in eating disorders is explained by bipolar disorders
anxiety disorders	0.35	0.01	Moderate relationship. The effect size is smaller than bipolar disorders
schizophrenia disorders	0.25	0.01	Weak to moderate relationship. However, there is a linear increasing trend

The low MSE (0.01) in all three models indicates that the mean square prediction error is very small, demonstrating that the models have high accuracy in fitting the normalized data.

We performed similar analyses with dataset 2 to support the hypothesis that other psychiatric disorders also have a significant impact on eating disorders. The simple linear regression models revealed a significant positive linear relationship between bipolar disorders and eating disorders, with a correlation coefficient R^2 of 0.46 and a relatively small mean square error (MSE) of 0.01, indicating a good model fit and well-distributed data. The association between anxiety disorders and eating disorders was also noted at $R^2 = 0.35$, with an MSE of 0.01. In addition, we also observed that bipolar disorders had a moderate relationship with anxiety disorders $R^2 = 0.34$, suggesting the possibility that these disorders coexist and influence each other in the same patient group.

Additional Linear regression

In addition, we used regression modeling to analyze the connection between healthcare access and depression rates. We performed ordinary least squares (OLS) regression each year to assess the relationship between the UHC index and depression rates during the study period. This method helped us examine both general trends and changes over time in how access to healthcare affects mental health outcomes (Agresti and Kateri, 2022).

Case Study: Comparison of Sweden and the United States

This case study investigates the relationship between Universal Health Coverage (UHC) and depression rates in Sweden and the United States, two countries with significantly different healthcare systems. Even though the United States has a higher UHC index, which suggests broader healthcare coverage, it paradoxically faces higher rates of depression. On the other hand side, Sweden, with a slightly lower UHC index, reports a lower prevalence of depression. This inconsistency suggests that even though healthcare coverage is important, other factors, such as the quality of healthcare, equitable access, and social contributors, also play vital roles in shaping mental health outcomes (Patel et al., 2018; World Health Organization, 2023).

This comparison underscores the complexity of how health systems shape mental health. Sweden's healthcare system prioritizes equitable service distribution and preventive care, which appear to promote better mental well-being. Conversely, the U.S. system grapples with issues of fragmented coverage and inequitable access, which may contribute to higher depression rates, particularly among vulnerable populations.

Predicting prevalence of Eating Orders

This section covers various prediction models used to predict the rate or prevalence of eating disorders by fitting multiple models to numerical features in the mental illness prevalence dataset (IMT Kaggle Team, 2023).

GLM

To assess the combined effect of psychiatric disorders on the prevalence of eating disorders, we constructed a generalized linear regression model (GLM) with four main independent variables: bipolar disorder, anxiety disorder, depression disorder, and schizophrenia. This multivariate approach enabled the examination of the individual effects of each factor while controlling for the presence of the remaining factors.

- Dependent variable: eating disorders

- Independent variables: bipolar disorder, anxiety disorder, depressive disorder, schizophrenia

- Model type: GLM with Gaussian distribution and identity link function

All variables have significant effects on eating disorders, suggesting that each of these disorders plays an important role in predicting the prevalence of eating disorders. In particular, bipolar disorder and schizophrenia are the two strongest factors with the highest coefficients in the model.

Neural Network

Neural networks are constructed of multiple layers of simplified artificial neurons that sum all weighted inputs (with bias) and apply activation f. This result y can then be propagated as an input in the next layer of the neural network (Goodfellow et al., 2016).

$$y = f\left(\sum_{j=0}^{i} w_j x_j + b_j\right)$$

Neural networks can be used for regression or classification and for predicting prevalence of eating disorders given various other disorders as input, we have one output node representing this prediction. Training of the neural network is done via backward propagation, where we effectively use the derivative chain rule [Add citation here] to adjust weights back through the neural network from the desired output node value. A simplified mental model here would be that we change the inputs by a specific amount and the output changes by a corresponding amount (slope or derivative).

Random Forest Regressor

The Random Forest Regressor is an ensemble learning algorithm that builds multiple decision trees and outputs the average of their predictions to estimate a continuous target variable Breiman, 2001. By averaging over many trees trained on different subsets of the data (via bootstrapping), it reduces overfitting and improves generalization. Unlike XGBoost, which builds trees sequentially and focuses on correcting errors made by previous trees, Random Forest builds trees independently in parallel. This simplicity comes with the benefit of easier interpret-ability and tuning, as its hyperparameters primarily control the number and depth of trees in the forest.

A simple analogy for describing the differences between decision trees (DT), random forests (RF) and XGBoost is playing a hole of golf:

- for DT you get one shot off the tee
- for RF you can hit many balls and pick the best one
- for XGBoost you can hit one ball walk up to the ball and hit it again until you get close to the hole as possible

K-Nearest Neighbours

K-Nearest Neighbours (KNN) is a supervised learning algorithm that predicts the output for a given data point by looking at the outputs of the K most similar instances in the training set (Hastie et al., 2009). Similarity is typically measured using a distance metric like Euclidean distance. In regression, the predicted value is usually the average of the target values of these K neighbours. KNN is non-parametric, meaning it makes no assumptions about the underlying data distribution, and it can model complex relationships with sufficient data.

Algorithm 1 K-Nearest Neighbours (KNN) Regression

Require: $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ (training features), $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ (target values), x_{query} (query point), K (number of neighbours)
Ensure: Predicted value $\hat{y}$ for x_{query}

```
function KNNREGRESSION(X, y, x_query, K)
    for i ← 1 to n do
        d_i ← ||x_i − x_query||                ▷ Compute distance from query point
    end for
    Identify indices I of K nearest neighbours with smallest d_i
    ŷ ← (1/K) Σ_{i∈I} y_i                      ▷ Predict as average of K nearest target values
    return ŷ
end function
```

Support Vector Regressor

Support Vector Regression (SVR) is a supervised learning algorithm derived from Support Vector Machines (SVMs) (**svr**), but designed for predicting continuous outcomes rather than classifying categories. In the context of predicting outcomes such as the severity or risk of eating disorders, SVR constructs a function that fits the data within a specified margin of tolerance, rather than attempting to classify it into discrete groups. The algorithm seeks a regression hyperplane that minimizes deviations beyond a pre-defined threshold (epsilon), while simultaneously ensuring that the model remains as flat (simple) as possible. The support vectors in this context are the data points that lie on or outside the margin and influence the position of the regression line Kuhn and Johnson, 2016.

To capture complex, nonlinear relationships between features and the target variable, we apply the Radial Basis Function (RBF) kernel. This kernel function implicitly maps the input data into a higher-dimensional feature space, allowing the SVR model to learn nonlinear patterns without explicitly performing the transformation. This flexibility is especially useful when the input features have intricate interactions. To ensure the training and testing datasets maintain representative distributions, we use stratification during data splitting.

Results – Model Analysis

Linear regression analysis result showed that all three psychiatric disorders, bipolar disorder, anxiety disorder, and schizophrenia, had a positive linear relationship with the prevalence of eating disorders. Among them, bipolar disorder showed the strongest predictive power with a coefficient of R^2=0.46, followed by anxiety disorder (0.35) and schizophrenia disorder (0.25). Despite the different levels of association, all three models had a low mean square error (MSE) of only 0.01, indicating that the model had high predictive accuracy when using standardized data. These results confirm that psychiatric disorders, especially bipolar disorder, are important predictors of eating disorder prevalence in the population.

Table 3 shows evaluation metrics for five regression models: GLM, K-Nearest Neighbors, Neural Network, Random Forest, and Support Vector Regressor. For each model we calculate R^2 and mean squared error (MSE) scores for train and test predictions. We also calculate a 95% confidence intervals (CI) for the test R^2 score where applicable. Finally we calculate the p-value in relation to the reference model which is the GLM. The p-value indicates whether the model is significantly better than the reference model p-value < 0.05 (Agresti and Kateri, 2022).

The **Generalized Linear Model** using a train-test split indicates that all mental health disorders analyzed have a statistically significant impact on eating disorders:

- **Bipolar disorder** shows the strongest effect, with a coefficient of +0.4156,
- **Schizophrenia disorder** follows closely with a strong positive effect of +0.3912,
- **Anxiety disorder** has a moderate positive effect (+0.1312),
- **Depression disorder** also contributes positively, though the effect is smaller (+0.0346).

All predictor variables had significant effects on eating disorders, and it was concluded that these disorders can predict the prevalence of eating disorders.The R^2 value on the test set was 0.68, and on the training set was 0.65, indicating that the model has strong explanatory power. The 95 percent confidence interval on the test set was between (0.6526 and 0.7017), indicating the high stability of the model. The mean square error (MSE) was 0.0006 on the training set and 0.00064 on the test set, indicating that the model predicted accurately and with slight bias.

The **Random Forest Regressor** was the best overall performing model and achieved the highest test R^2 score (0.9950) and the lowest test MSE (0.000101). Its 95% confidence interval for the test R^2 $[0.9908, 0.9984]$ was also the tightest amoung all models (this result strongly indicates consistent generalization).

The **K-Nearest Neighbors Regressor** also performed well and achieved a test R^2 of 0.9893 and a narrow confidence interval $[0.9793, 0.9966]$, showing reliable generalization.

The **Neural Network** model yielded a test R^2 of 0.9156 and a test MSE of 0.001686. While this performance was weaker than the tree-based models, it was run for 100 iterations and demonstrated reasonable accuracy. The p-value (2.18×10^{-11}) indicates a statistically significant difference in the performance relative to the Random Forest regressor model.

The **Support Vector Regressor** showed the lowest predictive performance with a test R^2 of 0.8037 and a test MSE of 0.003922. Furthermore the confidence interval $[0.7742, 0.8295]$ showed the model had a lower stability and accuracy.

Table 3

Model Evaluation Metrics with Confidence Intervals and Significance Testing

Model	R^2_{train}	R^2_{test}	95% CI	$\text{MSE}_{\text{train}}$	MSE_{test}	p-value
GLM	0.65	0.68	[0.653, 0.702]	0.0067	0.0064	–
K-Nearest Neighbors	0.9952	0.9893	[0.9793, 0.9966]	9.1e-05	2.1e-04	–
Neural Network	0.9286	0.9156	–	1.4e-03	1.7e-03	2.18e-11
Random Forest	**0.9995**	**0.9950**	**[0.9908, 0.9984]**	**1.0e-05**	**1.0e-04**	**6.05e-03**
Support Vector Regressor	0.8016	0.8037	[0.7742, 0.8295]	3.8e-03	3.9e-03	< 1e-40
GLM	-	0.085	-	< mse	< mse	< 0.001

In summary: as shown in Table 3, the **Random Forest Regressor** achieves the highest performance, with the lowest test MSE (0.000101) and the highest test R^2 (0.9950). This on the surface indicates good generalization ability. The **K-Nearest Neighbors Regressor** also performs well, albeit with a slightly higher test MSE error. The neural network model showed some larger MSE and could benefit from larger number of iterations or alternative network architecture (which we will leave to future work).

Relationship between the quality of healthcare measured by the Universal Health Coverage (UHC) Index and Depression across countries

Table 4

Yearly Correlation and Regression Results for UHC and Depression Rates

Year	**Pearson r**	**p-value**	**95% CI for r**	**Regression Slope (β)**	**95% CI for β**	R^2
2000	-0.32	$< .001$	–	-0.0151	$[-0.022, -0.008]$	0.104
2005	-0.37	$< .001$	–	-0.0165	$[-0.023, -0.010]$	0.135
2010	-0.40	$< .001$	–	-0.0198	$[-0.027, -0.013]$	0.163
2015	-0.45	$< .001$	–	-0.0221	$[-0.029, -0.015]$	0.205
2017	-0.45	$< .001$	–	-0.0224	$[-0.029, -0.016]$	0.200
2019	-0.46	$< .001$	–	-0.0228	$[-0.030, -0.016]$	0.208
Overall	**-0.41**	$< .001$	$[-0.53, -0.28]$	–	–	–

Table 4 presents a comprehensive analysis of annual data from 2000 to 2019, highlighting Pearson correlation coefficients, regression slopes, and R^2 values that investigate the relationship between the Universal Health Coverage (UHC) Index and rates of depression across various countries. The analysis reveals that, in each year examined, higher UHC scores correlate with lower national depression rates. The negative regression coefficients indicate that increases in health coverage are consistently associated with modest yet significant decreases in the prevalence of depression. These findings provide robust evidence of a stable inverse relationship between access to healthcare and mental health outcomes on a global scale.

Case Study

Table 5

Comparison of Mean UHC Index and Depression Rates: United States and Sweden

Country	Mean UHC Index	Mean Depression Rate (%)
Sweden	80.5	4.17
United States	83.0	4.43
Difference	+2.5	+0.27

Table 5 presents the average UHC index and mean depression rates for the United States and Sweden. While the United States has a slightly higher UHC index, Sweden exhibits a lower average depression rate. This pattern suggests that, although broader health coverage is essential, additional factors, such as the quality of care, access to mental health services, and broader social support, likely contribute to national mental health outcomes.

Conclusions and Recommendations

This research employed a combination of linear regression, generalized linear models (GLM), and advanced machine learning algorithms, including artificial neural networks, random forests, K-nearest neighbors (KNN), and support vector regression (SVR), to analyze the relationship between mental disorders and the prevalence of eating disorders (Forrest et al., 2023). By applying correlation matrices, covariance matrices, and regression analysis on standardized data, the study not only identified the relevant factors but also evaluated the predictive power of each model. The application of a training and testing set separation strategy ensures objectivity and highlights the generalizability of the models when applied to real-world data (Ghosh et al., 2024). The results of the study indicate that mental disorders, especially bipolar disorder and schizophrenia, are strong predictors of the prevalence of eating disorders. Models such as GLM and Random Forest showed high predictive performance and good explanatory power, while neural networks provided reasonable complementary insights. The consistency of statistical indicators, such as correlation, R^2, MSE, and confidence intervals, strengthened the confidence in the findings.

In addition, extending the analysis to data on universal health coverage (UHC) and national depression prevalence provides a more comprehensive context for understanding the influence of health system factors on community mental health. Therefore, research contributes not only a way in which further to understand the link between mental disorders and eating disorders but also serves as a framework for applicable public health policy and intervention approaches for the prevention and treatment of eating disorders, eating behaviors, and mental health. This research also serves as a foundation for future studies to develop and deploy practical and widely useful predictive models for mental health care.

While exploring the link between Universal Health Coverage and depression, this analysis concludes that countries with better healthcare systems tend to have lower depression rates (World Health Organization, 2023). However, when focusing on a specific case study comparing the healthcare systems of the United States and Sweden, the findings show that even though the United States has a higher UHC index, it still has a higher rate of depression. These results suggest that additional factors, including the quality of care, access to mental health services, and broader social conditions, play a crucial role in determining national mental health outcomes (Patel et al., 2018).

Future work

Future research should explore how additional factors, such as income, education, and social support, interact with health coverage to impact depression rates (Patel et al., 2018). It may also be beneficial to analyze policy variations, monitor changes within countries over time, or concentrate on specific subgroups to gain a clearer understanding of which populations derive the most benefit from expanded healthcare (World Health Organization, 2023). Investigating these areas could offer a more comprehensive insight into the connections between health systems and mental health outcomes.

References

Agresti, A., & Kateri, M. (2022). *Foundations of statistics for data scientists: With r and python*. CRC Press.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Forrest, L. N., Ivezaj, V., & Grilo, C. M. (2023). Machine learning vs. traditional regression models predicting treatment outcomes for binge-eating disorder from a randomized controlled trial. *Psychological Medicine*, *53*(7), 2777–2788. https://doi.org/10.1017/S0033291721004748

Ghosh, S., Burger, P., Simeunovic-Ostojic, M., Maas, J., & Petković, M. (2024). Review of machine learning solutions for eating disorders. *International Journal of Medical Informatics*, *189*, 105526. https://doi.org/10.1016/j.ijmedinf.2024.105526

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. https://www.deeplearningbook.org

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd). Springer. https://web.stanford.edu/~hastie/ElemStatLearn/

IMT Kaggle Team. (2023). Mental health dataset [Accessed June 22, 2025].

Kuhn, M., & Johnson, K. (2016). *Applied predictive modeling*. Springer. https://doi.org/10.1007/978-1-4614-6849-3

Patel, V., Saxena, S., Lund, C., Thornicroft, G., Baingana, F., Bolton, P., & Unützer, J. (2018). The lancet commission on global mental health and sustainable development. *The Lancet*, *392*(10157), 1553–1598. https://doi.org/10.1016/S0140-6736(18)31612-X

World Health Organization. (2023). Universal health coverage (uhc).

Appendix

Notebook Appendix

Below you can find the Jupyter notebook in order to reproduce documented results

Setup Datasets

Imports

In [4]:
```
import matplotlib.pyplot as plt
import pandas as pd
import requests
import seaborn as sns
from scipy.stats import zscore
from scipy.stats import pearsonr
from sklearn import preprocessing
from sklearn.linear_model import LinearRegression
from itertools import combinations
from sklearn.metrics import r2_score, mean_squared_error
import statsmodels.api as sm
import statsmodels.formula.api as smf # Create a GLM model
from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA # Import PCA for visualization
#!pip install tensorflow keras
import tensorflow as tf
from tensorflow import keras
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

Load Datasets

In [5]:
```
def load_dataset_from_github(api_directory_url):
    """
    Given a GitHub API URL pointing to a repository directory, fetches all CSV files
    in that directory and returns a dictionary mapping filenames to pandas DataFrames.

    Parameters:
    -----------
    api_directory_url : str
        GitHub API URL for a repository directory, e.g.
        "https://api.github.com/repos/username/repo/contents/path/to/dir"

    Returns:
    --------
    dict[str, pandas.DataFrame]
        A dictionary where each key is a CSV filename (e.g., "data.csv")
        and each value is the corresponding DataFrame obtained from reading
        the file's raw URL.
    array of dataframes
    """
    response = requests.get(api_directory_url)
    response.raise_for_status()  # Raise an error if request failed
    files = response.json()

    dataframes = {}
    for file in files:
        name = file.get("name", "")
        if name.endswith(".csv"):
            raw_url = file.get("download_url")
            if raw_url:
                dataframes[name] = pd.read_csv(raw_url)
    ordered_names = [
        '1-mental-illnesses-prevalence.csv',
        '2-burden-disease-from-each-mental-illness.csv',
        '3-adult-population-covered-in-primary-data-on-the-prevalence-of-major-depression.csv',
        '4-adult-population-covered-in-primary-data-on-the-prevalence-of-mental-illnesses.csv',
        '5-anxiety-disorders-treatment-gap.csv',
        '6-depressive-symptoms-across-us-population.csv',
        '7-number-of-countries-with-primary-data-on-prevalence-of-mental-illnesses-in-the-global-burden-of-disease-study.csv',
        'GDP.csv'
    ]
    dfs = [dataframes[name] for name in ordered_names]

    return dataframes, dfs
```

Utility Functions

In [6]:
```
from sklearn.metrics import r2_score, mean_squared_error
from scipy.stats import ttest_rel
import numpy as np
import matplotlib.pyplot as plt
```

```
# Global store
model_results = []
reference_residuals = None  # Set externally for p-value comparison

def bootstrap_r2(y_true, y_pred, n_bootstrap=1000, alpha=0.05):
    """
    Bootstrap confidence intervals for R² score.
    """
    n = len(y_true)
    r2_scores = []
    rng = np.random.default_rng(seed=42)
    for _ in range(n_bootstrap):
        indices = rng.integers(0, n, size=n)
        r2_scores.append(r2_score(y_true[indices], y_pred[indices]))
    lower = np.percentile(r2_scores, 100 * alpha / 2)
    upper = np.percentile(r2_scores, 100 * (1 - alpha / 2))
    return (lower, upper)

def evaluate_model(model, model_name, X_train, y_train, X_test, y_test, store_results=True, plot=True):
    """
    Evaluate a regression model on training and test data, compute metrics, and optionally plot results.
    """
    global reference_residuals

    y_train_pred = model.predict(X_train)
    y_test_pred = model.predict(X_test)

    # Ensure predictions are 1D arrays if the model outputs 2D (like Neural Networks)
    if hasattr(y_train_pred, 'flatten'):
        y_train_pred = y_train_pred.flatten()
    if hasattr(y_test_pred, 'flatten'):
        y_test_pred = y_test_pred.flatten()

    # Ensure y_test is a NumPy array for consistent operations
    y_test_np = np.array(y_test)
    y_train_np = np.array(y_train)

    # Compute metrics
    r2_train = r2_score(y_train, y_train_pred)
    mse_train = mean_squared_error(y_train, y_train_pred)

    r2_test = r2_score(y_test, y_test_pred)
    mse_test = mean_squared_error(y_test, y_test_pred)

    # Bootstrap CI
    r2_test_ci = bootstrap_r2(np.array(y_test), np.array(y_test_pred))

    # Paired t-test vs reference model (if available)
    residuals = np.array(y_test) - np.array(y_test_pred)
    if reference_residuals is not None:
        _, p_value = ttest_rel(reference_residuals, residuals)
    else:
        p_value = None
        reference_residuals = residuals  # first model becomes reference

    print(f"\n{model_name} Evaluation:")
    print(f"Train R²: {r2_train:.4f}, Test R²: {r2_test:.4f} (95% CI: {r2_test_ci[0]:.4f}, {r2_test_ci[1]:.4f})")
    print(f"Train MSE: {mse_train:.4f}, Test MSE: {mse_test:.4f}")
    if p_value is not None:
        print(f"p-value vs reference model: {p_value:.4f}")

    if store_results:
        model_results.append({
            'model': model_name,
            'r2_train': r2_train,
            'r2_test': r2_test,
            'r2_test_ci_lower': r2_test_ci[0],
            'r2_test_ci_upper': r2_test_ci[1],
            'mse_train': mse_train,
            'mse_test': mse_test,
            'p_value_vs_ref': p_value
        })

    if plot:
        plt.figure(figsize=(8, 8))
        plt.scatter(y_test, y_test_pred, alpha=0.6, label='Predicted vs Actual')
        plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], 'k--', lw=2, label='Ideal Fit')
        plt.xlabel('Actual')
        plt.ylabel('Predicted')
        plt.title(f'{model_name} - Actual vs Predicted')
        plt.legend()
        plt.grid(True)
        plt.show()

    return {
        'r2_train': r2_train,
```

```
        'r2_test': r2_test,
        'r2_test_ci': r2_test_ci,
        'mse_train': mse_train,
        'mse_test': mse_test,
        'p_value_vs_ref': p_value
    }

def box_plots(df_skip):
    """
    Draw boxplots for each column in the DataFrame to visualize outliers.
    """
    for column in df_skip:
        print(f"Boxplot for Column Name: {column}")
        #draw boxplot and kde before analyte to visualize the outliers
        fig, axes = plt.subplots(ncols=3, nrows=1, figsize=(6, 2))
        plt.subplots_adjust(wspace=1, hspace=0.25)
        axes[0].set_title(f'Boxplot | Mean: {df_skip[column].mean():.3f}', fontsize=8)
        sns.boxplot(y=df_skip[column], ax=axes[0])

        #handle outliers using IQR
        Q1 = df_skip[column].quantile(0.25)
        Q3 = df_skip[column].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR
        df_iqr = df_skip[[column]].copy()
        df_iqr[column]= df_skip[column].clip(lower=lower_bound, upper=upper_bound)

        #draw boxplot after handling outliers
        axes[1].set_title(f'Boxplot after IQR | Mean: {df_iqr[column].mean():.3f}', fontsize=8)
        sns.boxplot(y=df_iqr[column], ax=axes[1], color='orange')

        # Try z-score method to compare results with IQR method
        df_z = df_skip[[column]].copy()
        df_z['z_score'] = zscore(df_z[column])
        df_z['capped'] = df_z[column].where(df_z['z_score'].abs() <= 3) #created capped column to cap values smaller or equal

        axes[2].set_title(f'After Z-Score | Mean: {df_z[column].mean():.3f}', fontsize=8)
        sns.boxplot(y=df_z['capped'].dropna(), ax=axes[2], color='green')

        plt.show()

def find_strong_relation(corr_matrix, cov_matrix):
    """
    Find pairs of variables with strong correlation and their covariance.
    """
    strong_corr_pairs = []
    # Find pairs of variables with strong correlation (|corr| >= 0.5) and their covariance
    for i in range(len(corr_matrix.columns)):
        for j in range(i+1, len(corr_matrix.columns)):
            col1 = corr_matrix.columns[i]
            col2 = corr_matrix.columns[j]
            corr_val = corr_matrix.loc[col1, col2]
            if abs(corr_val) >= 0.5:
                cov_val = cov_matrix.loc[col1, col2]
                strong_corr_pairs.append((col1, col2, corr_val, cov_val))
    result_df = pd.DataFrame(strong_corr_pairs, columns=['Variable 1', 'Variable 2', 'Correlation', 'Covariance'])

    return result_df

def linear_regression_plot(df, col_x, col_y):
    """
    Perform linear regression on two columns of a DataFrame and plot the results.
    """
    X = df[[col_x]]
    y = df[col_y]
    model = LinearRegression().fit(X, y)
    y_pred = model.predict(X)
    r2 = r2_score(y, y_pred)
    mse = mean_squared_error(y, y_pred)

    plt.figure(figsize=(6, 4))
    plt.scatter(X, y, alpha=0.6, label='Data Points')
    plt.plot(X, y_pred, color='red', label='Regression Line')
    plt.xlabel(col_x)
    plt.ylabel(col_y)
    plt.title(f'{col_x} vs {col_y} | Regression: {r2:.2f} | MSE: {mse:.2f}')
    plt.legend()
    plt.tight_layout()
    plt.show()
```

Clean Datasets

In [7]:
```
# Load CSV files
file_path = "https://api.github.com/repos/AAI500TeamProject/thementalists-project/contents/Dataset/MentalHealth"
```

```
dataframes, dfs = load_dataset_from_github(file_path)
```

In [8]:
```
# Drop columns
for name, df in dataframes.items():
     if name != 'GDP.csv':
          df.drop(columns ='Code', inplace=True)

          #Rename the lengthy columns to make the table compactpact=
          df.rename(columns = {
          'Schizophrenia disorders (share of population) - Sex: Both - Age: Age-standardized': 'Schizophrenia Disorders',
               'Depressive disorders (share of population) - Sex: Both - Age: Age-standardized': 'Depression Disorders',
               'Anxiety disorders (share of population) - Sex: Both - Age: Age-standardized': 'Anxiety Disorders',
               'Bipolar disorders (share of population) - Sex: Both - Age: Age-standardized': 'Bipolar Disorders',
               'Eating disorders (share of population) - Sex: Both - Age: Age-standardized' : 'Eating Disorders'
     }, inplace = True)

          df.rename(columns={
               'DALYs (rate) - Sex: Both - Age: Age-standardized - Cause: Depressive disorders': 'Depression Disorders',
               'DALYs (rate) - Sex: Both - Age: Age-standardized - Cause: Schizophrenia': 'Schizophrenia Disorders',
               'DALYs (rate) - Sex: Both - Age: Age-standardized - Cause: Bipolar disorder': 'Bipolar Disorder',
               'DALYs (rate) - Sex: Both - Age: Age-standardized - Cause: Eating disorders': 'Eating Disorders',
               'DALYs (rate) - Sex: Both - Age: Age-standardized - Cause: Anxiety disorders': 'Anxiety Disorders'
     }, inplace=True)

          df.rename(columns={
               'Potentially adequate treatment, conditional': 'Adequate_Treatment',
               'Other treatments, conditional': 'Other_Treatments',
               'Untreated, conditional': 'Untreated'
     }, inplace=True)

          df.rename(columns={
               'Nearly every day': 'Severe_Symptoms',
               'More than half the days': 'Moderate_Symptoms',
               'Several days': 'Mild_Symptoms',
               'Not at all': 'No_Symptoms'
     }, inplace=True)


          df.rename(columns={
               'Number of countries with primary data on prevalence of mental disorders': 'Countries_With_Data'
     }, inplace=True)

          # standardizing column names
          df.columns = df.columns.str.lower().str.strip().str.replace(' ', '_')

          # Trim whitespace in string columns
          for col in df.select_dtypes(include='object'):
               df[col] = df[col].str.strip()

          # Convert data types where appropriate
          if 'year' in df.columns:
               df['year'] = pd.to_numeric(df['year'], errors='coerce', downcast='integer')

# Check for and handle missing values
for i, df in enumerate(dfs):
    print(f"\nMissing values in DataFrame {i}:")
    print(df.isnull().sum())

# Preview each DataFrame
for i, df in enumerate(dfs):
    print(f"\nDataFrame {i} Preview:")
    print(df.head())
```

```
Missing values in DataFrame 0:
entity                    0
year                      0
schizophrenia_disorders   0
depression_disorders      0
anxiety_disorders         0
bipolar_disorders         0
eating_disorders          0
dtype: int64

Missing values in DataFrame 1:
entity                    0
year                      0
depression_disorders      0
schizophrenia_disorders   0
bipolar_disorder          0
eating_disorders          0
anxiety_disorders         0
dtype: int64

Missing values in DataFrame 2:
entity             0
year               0
major_depression   0
dtype: int64

Missing values in DataFrame 3:
entity              0
year                0
major_depression    0
bipolar_disorder    0
eating_disorders    0
dysthymia           0
schizophrenia       0
anxiety_disorders   0
dtype: int64

Missing values in DataFrame 4:
entity               0
year                 0
adequate_treatment   0
other_treatments     0
untreated            0
dtype: int64

Missing values in DataFrame 5:
entity              0
year                0
severe_symptoms     0
moderate_symptoms   0
mild_symptoms       0
no_symptoms         0
dtype: int64

Missing values in DataFrame 6:
entity                0
year                  0
countries_with_data   0
dtype: int64

Missing values in DataFrame 7:
Country Name       0
Country Code       0
Indicator Name     0
Indicator Code     0
1960             266
                ...
2020             266
2021              62
2022             266
2023             266
2024             266
Length: 69, dtype: int64

DataFrame 0 Preview:
        entity  year  schizophrenia_disorders  depression_disorders  \
0  Afghanistan  1990                 0.223206              4.996118
1  Afghanistan  1991                 0.222454              4.989290
2  Afghanistan  1992                 0.221751              4.981346
3  Afghanistan  1993                 0.220987              4.976958
4  Afghanistan  1994                 0.220183              4.977782

   anxiety_disorders  bipolar_disorders  eating_disorders
0           4.713314           0.703023          0.127700
1           4.702100           0.702069          0.123256
2           4.683743           0.700792          0.118844
3           4.673549           0.700087          0.115089
```

```
4          4.670810           0.699898         0.111815

DataFrame 1 Preview:
        entity  year  depression_disorders  schizophrenia_disorders  \
0  Afghanistan  1990             895.22565                138.24825
1  Afghanistan  1991             893.88434                137.76122
2  Afghanistan  1992             892.34973                137.08030
3  Afghanistan  1993             891.51587                136.48602
4  Afghanistan  1994             891.39160                136.18323

   bipolar_disorder  eating_disorders  anxiety_disorders
0         147.64412         26.471115          440.33000
1         147.56696         25.548681          439.47202
2         147.13086         24.637949          437.60718
3         146.78812         23.863169          436.69104
4         146.58481         23.189074          436.76800

DataFrame 2 Preview:
                 entity  year  major_depression
0  Andean Latin America  2008               0.0
1          Asia Pacific  2008              80.8
2           Australasia  2008             100.0
3             Caribbean  2008               9.1
4          Central Asia  2008               0.0

DataFrame 3 Preview:
                 entity  year  major_depression  bipolar_disorder  \
0  Andean Latin America  2008               0.0               0.0
1          Asia Pacific  2008              80.8               3.8
2           Australasia  2008             100.0             100.0
3             Caribbean  2008               9.1               0.0
4          Central Asia  2008               0.0               0.0

   eating_disorders  dysthymia schizophrenia  anxiety_disorders
0               0.0        0.0             0                0.0
1              23.1        1.0          71.6               93.1
2              16.4      100.0          85.1              100.0
3               0.0        0.0          28.3                0.0
4               0.0        0.0             0                0.0

DataFrame 4 Preview:
                    entity  year  adequate_treatment  other_treatments  \
0                Argentina  2015                12.0              18.0
1  Beijing/Shanghai, China  2005                 8.8               8.5
2                  Belgium  2002                11.2              24.5
3                 Bulgaria  2006                 7.3              14.3
4                 Colombia  2012                 3.2              10.0

   untreated
0       70.0
1       82.7
2       64.3
3       78.4
4       86.8

DataFrame 5 Preview:
                     entity  year  severe_symptoms  moderate_symptoms  \
0           Appetite change  2014              4.6                5.1
1   Average across symptoms  2014              4.4                4.3
2            Depressed mood  2014              3.6                3.9
3  Difficulty concentrating  2014              3.5                3.6
4          Loss of interest  2014              4.4                5.4

   mild_symptoms  no_symptoms
0           15.5         74.8
1           15.0         76.3
2           16.8         75.7
3           10.9         82.1
4           16.3         73.8

DataFrame 6 Preview:
                                      entity  year  countries_with_data
0                     Alcohol use disorders  2019                   58
1                 Amphetamine use disorders  2019                   58
2                          Anorexia nervosa  2019                   27
3                         Anxiety disorders  2019                   58
4  Attention-deficit hyperactivity disorder  2019                  172

DataFrame 7 Preview:
                 Country Name Country Code             Indicator Name  \
0                       Aruba          ABW  UHC service coverage index
1  Africa Eastern and Southern          AFE  UHC service coverage index
2                 Afghanistan          AFG  UHC service coverage index
3   Africa Western and Central          AFW  UHC service coverage index
4                      Angola          AGO  UHC service coverage index

      Indicator Code  1960  1961  1962  1963  1964  1965  ...  2015  2016  \
```

```
0  SH.UHC.SRVS.CV.XD   NaN   NaN   NaN   NaN   NaN   NaN  ...   NaN   NaN
1  SH.UHC.SRVS.CV.XD   NaN   NaN   NaN   NaN   NaN   NaN  ...   NaN   NaN
2  SH.UHC.SRVS.CV.XD   NaN   NaN   NaN   NaN   NaN   NaN  ...  36.0   NaN
3  SH.UHC.SRVS.CV.XD   NaN   NaN   NaN   NaN   NaN   NaN  ...   NaN   NaN
4  SH.UHC.SRVS.CV.XD   NaN   NaN   NaN   NaN   NaN   NaN  ...  36.0   NaN

   2017  2018  2019  2020  2021  2022  2023  2024
0   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
1   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
2  41.0   NaN  42.0   NaN  41.0   NaN   NaN   NaN
3   NaN   NaN   NaN   NaN   NaN   NaN   NaN   NaN
4  40.0   NaN  39.0   NaN  37.0   NaN   NaN   NaN

[5 rows x 69 columns]
```

In [9]:
```
for i, df in enumerate(dfs, start=5):
    print(f"{'='*60}")
    print(f"📊 DataFrame {i} Summary")
    print(f"{'='*60}")

    # Show basic info
    print(f"Shape: {df.shape}")
    print(f"\nColumns:\n{list(df.columns)}")

    # Show missing values neatly
    print(f"\nMissing Values:\n{df.isnull().sum()[df.isnull().sum() > 0].to_string()}")

    # Show preview
    print("\nPreview (first 5 rows):")
    display(df.head())  # Jupyter-friendly display
```

```
============================================================
📊 DataFrame 5 Summary
============================================================
Shape: (6420, 7)

Columns:
['entity', 'year', 'schizophrenia_disorders', 'depression_disorders', 'anxiety_disorders', 'bipolar_disorders', 'eating_disorde
rs']

Missing Values:
Series([], )

Preview (first 5 rows):
```

	entity	year	schizophrenia_disorders	depression_disorders	anxiety_disorders	bipolar_disorders	eating_disorders
0	Afghanistan	1990	0.223206	4.996118	4.713314	0.703023	0.127700
1	Afghanistan	1991	0.222454	4.989290	4.702100	0.702069	0.123256
2	Afghanistan	1992	0.221751	4.981346	4.683743	0.700792	0.118844
3	Afghanistan	1993	0.220987	4.976958	4.673549	0.700087	0.115089
4	Afghanistan	1994	0.220183	4.977782	4.670810	0.699898	0.111815

```
============================================================
📊 DataFrame 6 Summary
============================================================
Shape: (6840, 7)

Columns:
['entity', 'year', 'depression_disorders', 'schizophrenia_disorders', 'bipolar_disorder', 'eating_disorders', 'anxiety_disorder
s']

Missing Values:
Series([], )

Preview (first 5 rows):
```

	entity	year	depression_disorders	schizophrenia_disorders	bipolar_disorder	eating_disorders	anxiety_disorders
0	Afghanistan	1990	895.22565	138.24825	147.64412	26.471115	440.33000
1	Afghanistan	1991	893.88434	137.76122	147.56696	25.548681	439.47202
2	Afghanistan	1992	892.34973	137.08030	147.13086	24.637949	437.60718
3	Afghanistan	1993	891.51587	136.48602	146.78812	23.863169	436.69104
4	Afghanistan	1994	891.39160	136.18323	146.58481	23.189074	436.76800

```
============================================================
📊 DataFrame 7 Summary
============================================================
Shape: (22, 3)

Columns:
['entity', 'year', 'major_depression']

Missing Values:
Series([], )

Preview (first 5 rows):
```

	entity	year	major_depression
0	Andean Latin America	2008	0.0
1	Asia Pacific	2008	80.8
2	Australasia	2008	100.0
3	Caribbean	2008	9.1
4	Central Asia	2008	0.0

```
============================================================
📊 DataFrame 8 Summary
============================================================
Shape: (22, 8)

Columns:
['entity', 'year', 'major_depression', 'bipolar_disorder', 'eating_disorders', 'dysthymia', 'schizophrenia', 'anxiety_disorder
s']

Missing Values:
Series([], )

Preview (first 5 rows):
```

	entity	year	major_depression	bipolar_disorder	eating_disorders	dysthymia	schizophrenia	anxiety_disorders
0	Andean Latin America	2008	0.0	0.0	0.0	0.0	0	0.0
1	Asia Pacific	2008	80.8	3.8	23.1	1.0	71.6	93.1
2	Australasia	2008	100.0	100.0	16.4	100.0	85.1	100.0
3	Caribbean	2008	9.1	0.0	0.0	0.0	28.3	0.0
4	Central Asia	2008	0.0	0.0	0.0	0.0	0	0.0

```
============================================================
📊 DataFrame 9 Summary
============================================================
Shape: (26, 5)

Columns:
['entity', 'year', 'adequate_treatment', 'other_treatments', 'untreated']

Missing Values:
Series([], )

Preview (first 5 rows):
```

	entity	year	adequate_treatment	other_treatments	untreated
0	Argentina	2015	12.0	18.0	70.0
1	Beijing/Shanghai, China	2005	8.8	8.5	82.7
2	Belgium	2002	11.2	24.5	64.3
3	Bulgaria	2006	7.3	14.3	78.4
4	Colombia	2012	3.2	10.0	86.8

```
============================================================
📊 DataFrame 10 Summary
============================================================
Shape: (10, 6)

Columns:
['entity', 'year', 'severe_symptoms', 'moderate_symptoms', 'mild_symptoms', 'no_symptoms']

Missing Values:
Series([], )

Preview (first 5 rows):
```

	entity	year	severe_symptoms	moderate_symptoms	mild_symptoms	no_symptoms
0	Appetite change	2014	4.6	5.1	15.5	74.8
1	Average across symptoms	2014	4.4	4.3	15.0	76.3
2	Depressed mood	2014	3.6	3.9	16.8	75.7
3	Difficulty concentrating	2014	3.5	3.6	10.9	82.1
4	Loss of interest	2014	4.4	5.4	16.3	73.8

```
============================================================
📊 DataFrame 11 Summary
============================================================
Shape: (15, 3)

Columns:
['entity', 'year', 'countries_with_data']

Missing Values:
Series([], )

Preview (first 5 rows):
```

	entity	year	countries_with_data
0	Alcohol use disorders	2019	58
1	Amphetamine use disorders	2019	58
2	Anorexia nervosa	2019	27
3	Anxiety disorders	2019	58
4	Attention-deficit hyperactivity disorder	2019	172

```
============================================================
📊 DataFrame 12 Summary
============================================================
Shape: (266, 69)

Columns:
['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code', '1960', '1961', '1962', '1963', '1964', '1965', '1966', '1
967', '1968', '1969', '1970', '1971', '1972', '1973', '1974', '1975', '1976', '1977', '1978', '1979', '1980', '1981', '1982',
'1983', '1984', '1985', '1986', '1987', '1988', '1989', '1990', '1991', '1992', '1993', '1994', '1995', '1996', '1997', '1998',
'1999', '2000', '2001', '2002', '2003', '2004', '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014',
'2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022', '2023', '2024']

Missing Values:
1960    266
1961    266
1962    266
1963    266
1964    266
1965    266
1966    266
1967    266
1968    266
1969    266
1970    266
1971    266
1972    266
1973    266
1974    266
1975    266
1976    266
1977    266
1978    266
1979    266
1980    266
1981    266
1982    266
1983    266
1984    266
1985    266
1986    266
1987    266
1988    266
1989    266
1990    266
1991    266
1992    266
1993    266
1994    266
1995    266
1996    266
1997    266
1998    266
1999    266
2000     62
2001    266
2002    266
2003    266
2004    266
2005     62
2006    266
2007    266
2008    266
2009    266
2010     62
2011    266
2012    266
2013    266
2014    266
2015     62
2016    266
2017     62
2018    266
2019     62
2020    266
2021     62
2022    266
2023    266
2024    266

Preview (first 5 rows):
```

	Country Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	1963	1964	1965	...	2015	2016	2017	2018	2019	2020	2(
0	Aruba	ABW	UHC service coverage index	SH.UHC.SRVS.CV.XD	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	N
1	Africa Eastern and Southern	AFE	UHC service coverage index	SH.UHC.SRVS.CV.XD	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	N
2	Afghanistan	AFG	UHC service coverage index	SH.UHC.SRVS.CV.XD	NaN	NaN	NaN	NaN	NaN	NaN	...	36.0	NaN	41.0	NaN	42.0	NaN	4
3	Africa Western and Central	AFW	UHC service coverage index	SH.UHC.SRVS.CV.XD	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	N
4	Angola	AGO	UHC service coverage index	SH.UHC.SRVS.CV.XD	NaN	NaN	NaN	NaN	NaN	NaN	...	36.0	NaN	40.0	NaN	39.0	NaN	3

5 rows × 69 columns

Scale Dataset #1

In [10]:
```
# Use the first DataFrame from the list of dataframes 'dfs'
cleaned_df = dfs[0].select_dtypes(include='number').drop(columns=['year'], errors='ignore').dropna()
df1 = cleaned_df;

features = ['schizophrenia_disorders', 'depression_disorders','anxiety_disorders','bipolar_disorders']
X_model = df1[features]
target = 'eating_disorders'
y_model = df1[target]

print("X_model head:")
print(X_model.head())
print("\ny_model head:")
print(y_model.head())

scaler = preprocessing.MinMaxScaler()
X_model_norm = scaler.fit_transform(X_model)

# Convert the normalized array back to a DataFrame with original column names
X_model_norm_df = pd.DataFrame(X_model_norm, columns=X_model.columns, index=X_model.index)

print("\nNormalized X_model head:")
print(X_model_norm_df.head())
```

```
X_model head:
   schizophrenia_disorders  depression_disorders  anxiety_disorders  \
0                 0.223206              4.996118           4.713314   
1                 0.222454              4.989290           4.702100   
2                 0.221751              4.981346           4.683743   
3                 0.220987              4.976958           4.673549   
4                 0.220183              4.977782           4.670810   

   bipolar_disorders  
0           0.703023  
1           0.702069  
2           0.700792  
3           0.700087  
4           0.699898  

y_model head:
0    0.127700
1    0.123256
2    0.118844
3    0.115089
4    0.111815
Name: eating_disorders, dtype: float64

Normalized X_model head:
   schizophrenia_disorders  depression_disorders  anxiety_disorders  \
0                 0.127142              0.567281           0.420084   
1                 0.124394              0.566166           0.418422   
2                 0.121826              0.564869           0.415700   
3                 0.119034              0.564153           0.414189   
4                 0.116095              0.564287           0.413783   

   bipolar_disorders  
0           0.393458  
1           0.392738  
2           0.391774  
3           0.391242  
4           0.391099  
```

Train Test Split

In [11]:
```
X_train, X_test, y_train, y_test = train_test_split(X_model_norm, y_model, test_size=0.20, random_state=42)

print("\nX_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (5136, 4)
X_test shape: (1284, 4)
y_train shape: (5136,)
y_test shape: (1284,)
```

Create Eating Disorder Label Column

In [12]:
```
eating_disorders_data = dfs[0]['eating_disorders']
low_risk_threshold = eating_disorders_data.quantile(0.33) # Example: bottom 33% is low
medium_risk_threshold = eating_disorders_data.quantile(0.66) # Example: middle 33% is medium, top 33% is high

def categorize_eating_disorder_risk(prevalence):
    if prevalence <= low_risk_threshold:
        return 'low_risk'
    elif prevalence <= medium_risk_threshold:
        return 'medium_risk'
    else:
        return 'high_risk'
df_eating_disorder_risk = dfs[0].copy()
df_eating_disorder_risk['eating_disorder_risk'] = eating_disorders_data.apply(categorize_eating_disorder_risk)
df_eating_disorder_risk
```

Out[12]:

	entity	year	schizophrenia_disorders	depression_disorders	anxiety_disorders	bipolar_disorders	eating_disorders	eating_disord
0	Afghanistan	1990	0.223206	4.996118	4.713314	0.703023	0.127700	medi
1	Afghanistan	1991	0.222454	4.989290	4.702100	0.702069	0.123256	medi
2	Afghanistan	1992	0.221751	4.981346	4.683743	0.700792	0.118844	medi
3	Afghanistan	1993	0.220987	4.976958	4.673549	0.700087	0.115089	medi
4	Afghanistan	1994	0.220183	4.977782	4.670810	0.699898	0.111815	medi
...	...	...	...	...	...	...	...	
6415	Zimbabwe	2015	0.201042	3.407624	3.184012	0.538596	0.095652	l
6416	Zimbabwe	2016	0.201319	3.410755	3.187148	0.538593	0.096662	l
6417	Zimbabwe	2017	0.201639	3.411965	3.188418	0.538589	0.097330	l
6418	Zimbabwe	2018	0.201976	3.406929	3.172111	0.538585	0.097909	l
6419	Zimbabwe	2019	0.202482	3.395476	3.137017	0.538580	0.098295	l

6420 rows × 8 columns

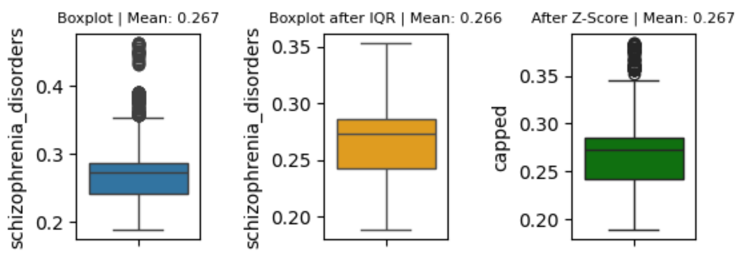
Box Plots for Mental Illnesses Prevalence

Dataset 1 Analysis: mental-illnesses-prevalence

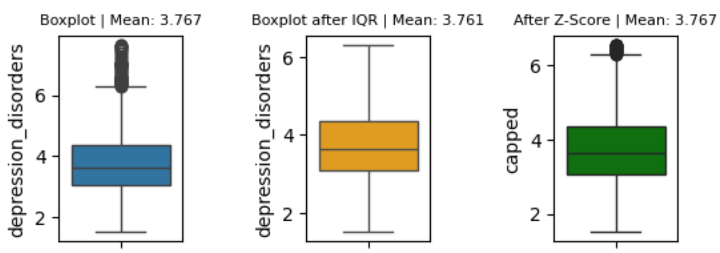
In [13]:

```
df_skip = dataframes['1-mental-illnesses-prevalence.csv'].iloc[:, 2:]
df_skip = df_skip.apply(pd.to_numeric, errors='coerce')
box_plots(df_skip)
```

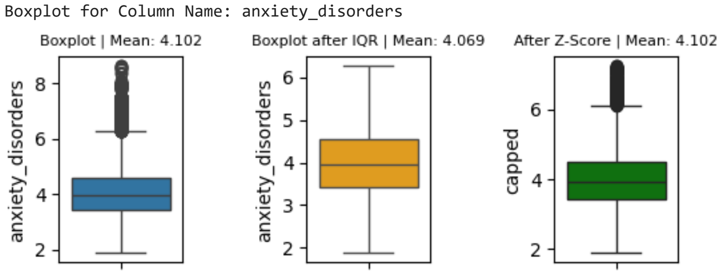
```
Boxplot for Column Name: schizophrenia_disorders
```



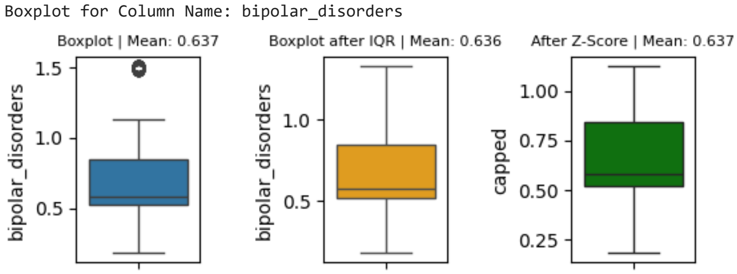

```
Boxplot for Column Name: depression_disorders
```



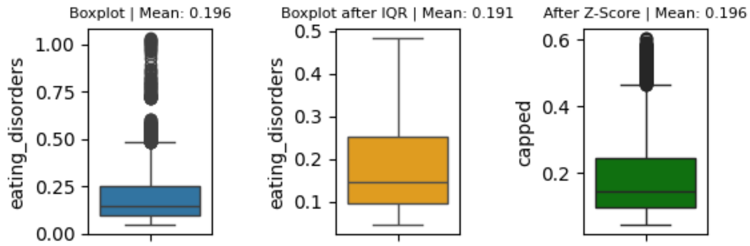

```
Boxplot for Column Name: anxiety_disorders
```




```
Boxplot for Column Name: bipolar_disorders
```



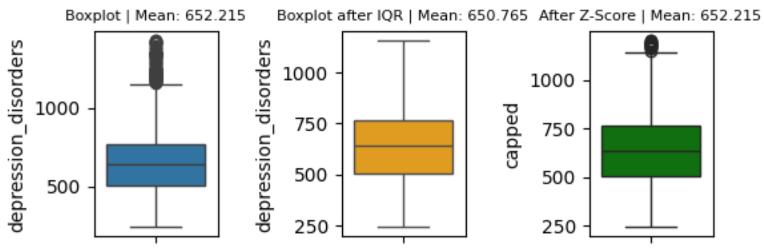

```
Boxplot for Column Name: eating_disorders
```



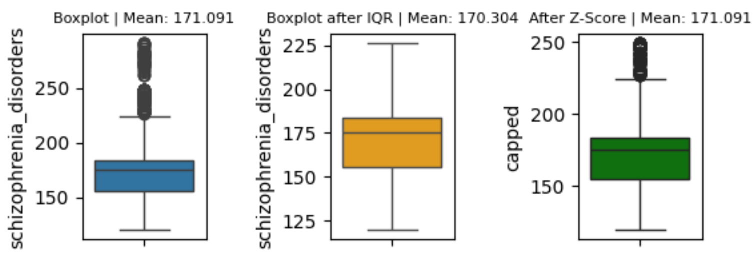
Dataset 2 Analysis: burden-disease-from-each-mental-illness

In [14]:
```
df_skip2 = dataframes['2-burden-disease-from-each-mental-illness.csv'].iloc[:, 2:]
df_skip2 = df_skip2.apply(pd.to_numeric, errors='coerce')
box_plots(df_skip2)
```

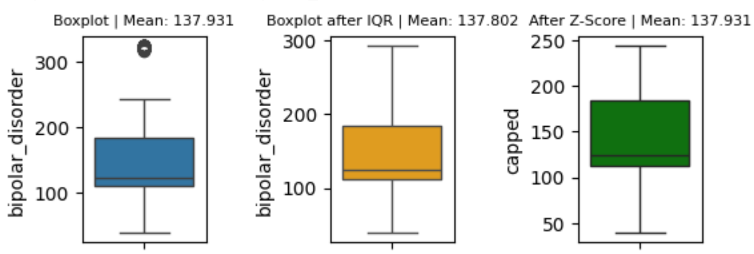
```
Boxplot for Column Name: depression_disorders
```



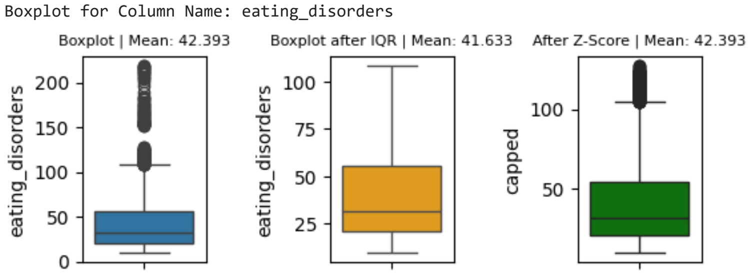

```
Boxplot for Column Name: schizophrenia_disorders
```



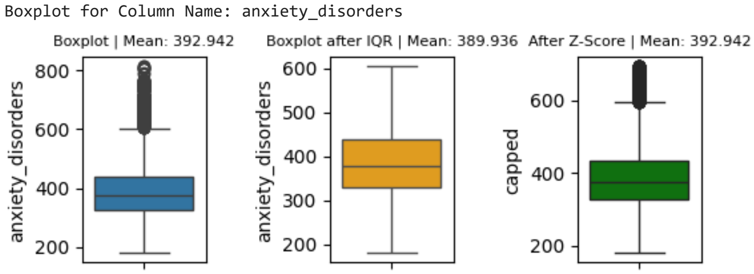

```
Boxplot for Column Name: bipolar_disorder
```




```
Boxplot for Column Name: eating_disorders
```




```
Boxplot for Column Name: anxiety_disorders
```



We decided not to remove outliers from the dataset for the following reasons:

In most cases, the outliers did not significantly affect the mean, indicating a relatively stable central tendency.

While some variables showed a noticeable shift in the mean, the outliers still represent valid, real-world observations rather than data entry errors.

These rare but extreme values could carry important information, especially in the context of mental health prevalence, and may be valuable for model learning, anomaly detection, or identifying high-risk populations.

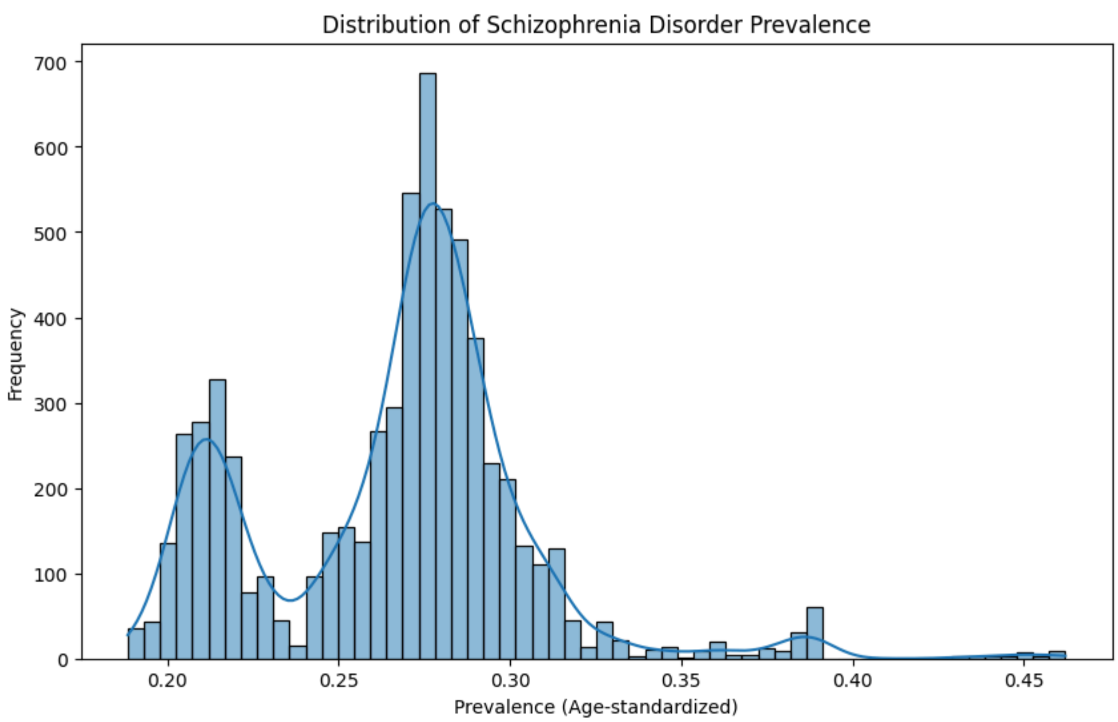
Removing them could lead to a loss of meaningful patterns and potentially limit the model's generalizability to edge cases.

Therefore, we chose to retain the outliers to preserve the full data distribution and ensure the model captures both common and exceptional conditions.

Exploratory Data Analysis

Histogram

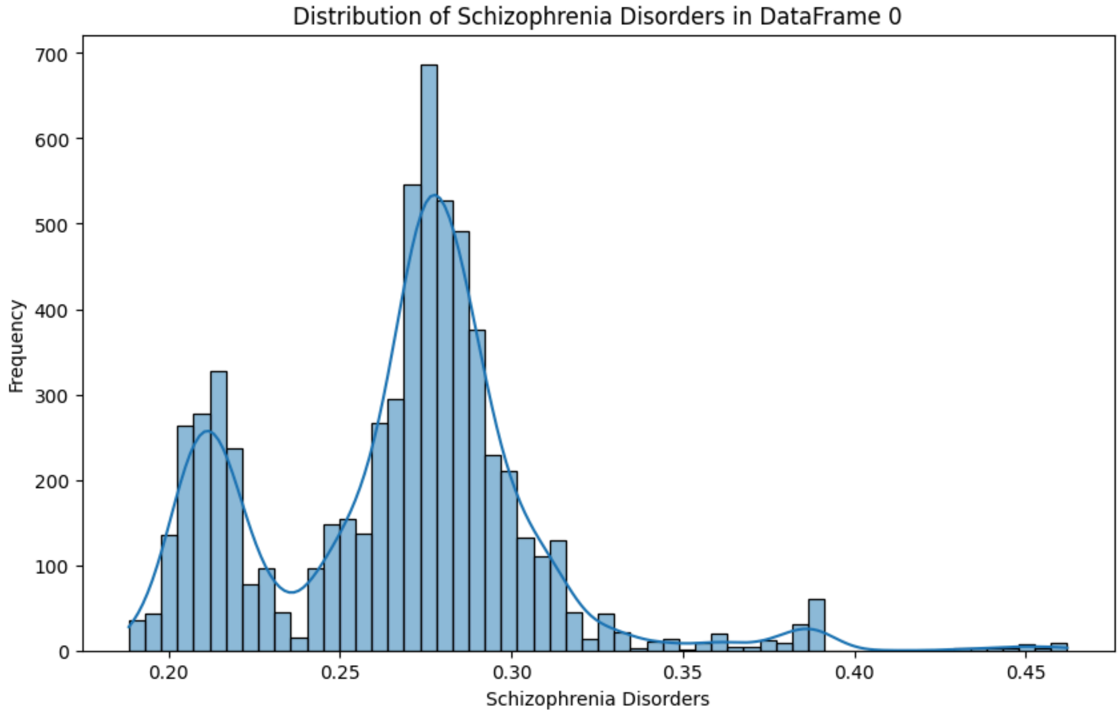
In [26]:
```
# Example histogram for a targeted variable
# Below this cell will be histograms for all
plt.figure(figsize=(10, 6))
sns.histplot(dfs[0]['schizophrenia_disorders'], kde=True)
plt.title('Distribution of Schizophrenia Disorder Prevalence')
plt.xlabel('Prevalence (Age-standardized)')
plt.ylabel('Frequency')
plt.show()
```

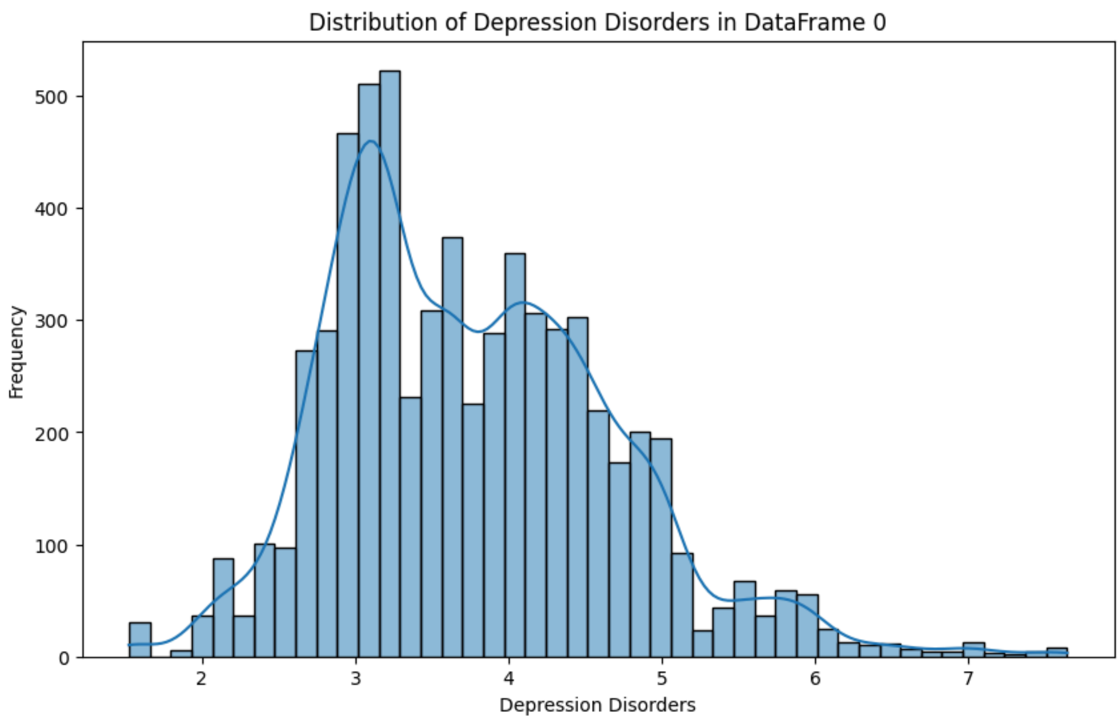


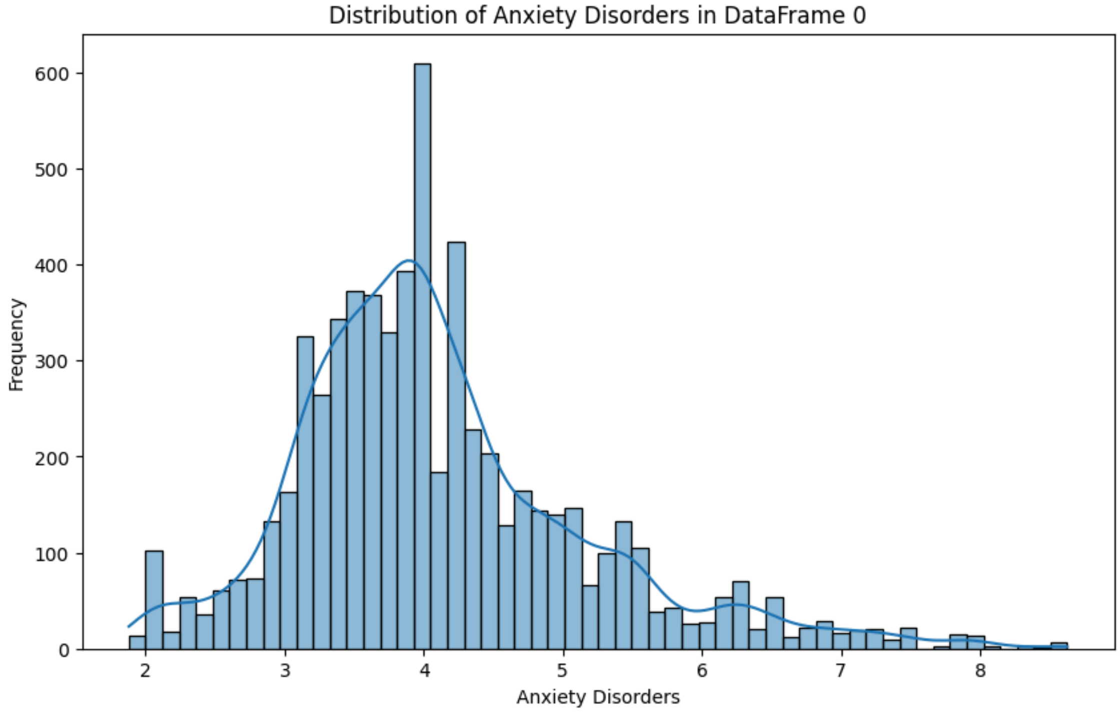
In [27]:
```
# Function to generate histograms for all numeric columns except 'year'
def plot_histograms(dfs):
  """
    Generates and displays histograms for all numeric columns in each DataFrame
    in the list, skipping the 'year' column if it exists.

    Parameters:
    -----------
    dfs : list of pandas.DataFrame
        A list of DataFrames to plot histograms from.
  """

  for i, df in enumerate(dfs):
    if i != len(dfs) - 1:
      print(f"\nPlotting histograms for DataFrame {i}")
      for col in df.select_dtypes(include='number').columns:
        if col != 'year':  # Skip the 'year' column
          plt.figure(figsize=(10, 6))
          sns.histplot(df[col].dropna(), kde=True) # Drop NA values for plotting
          plt.title(f'Distribution of {col.replace("_", " ").title()} in DataFrame {i}')
          plt.xlabel(col.replace("_", " ").title())
          plt.ylabel('Frequency')
          plt.show()
```

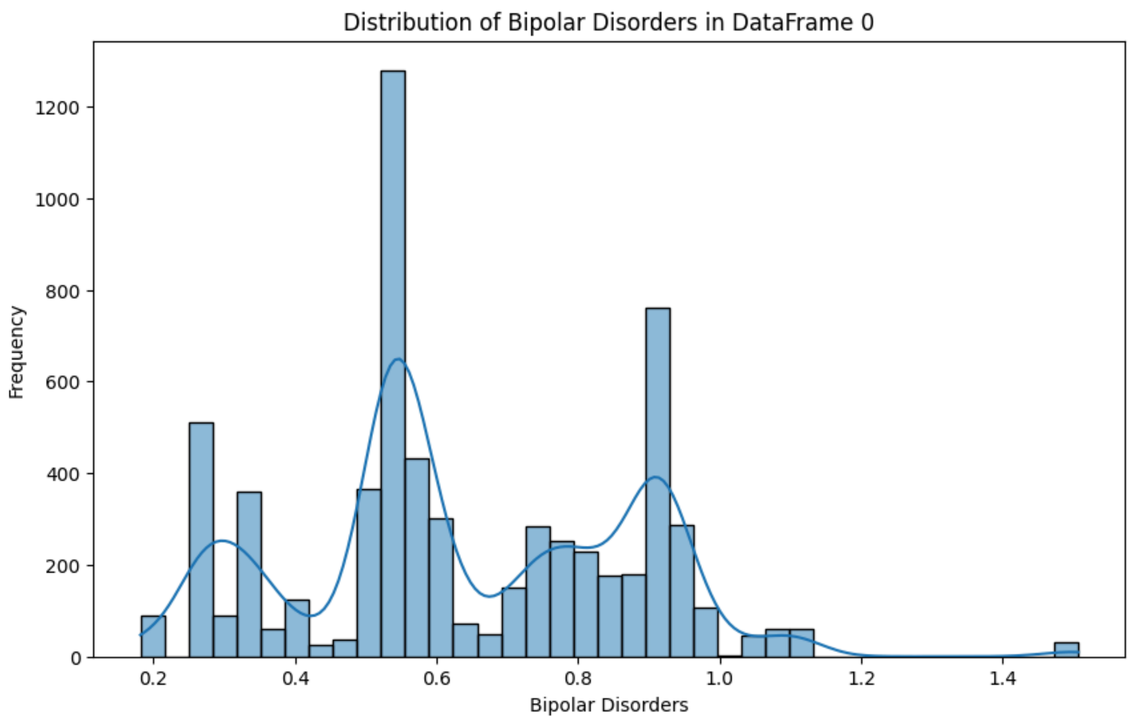
```
# Call the function to plot histograms for all numeric columns except 'year'
plot_histograms(dfs)
```

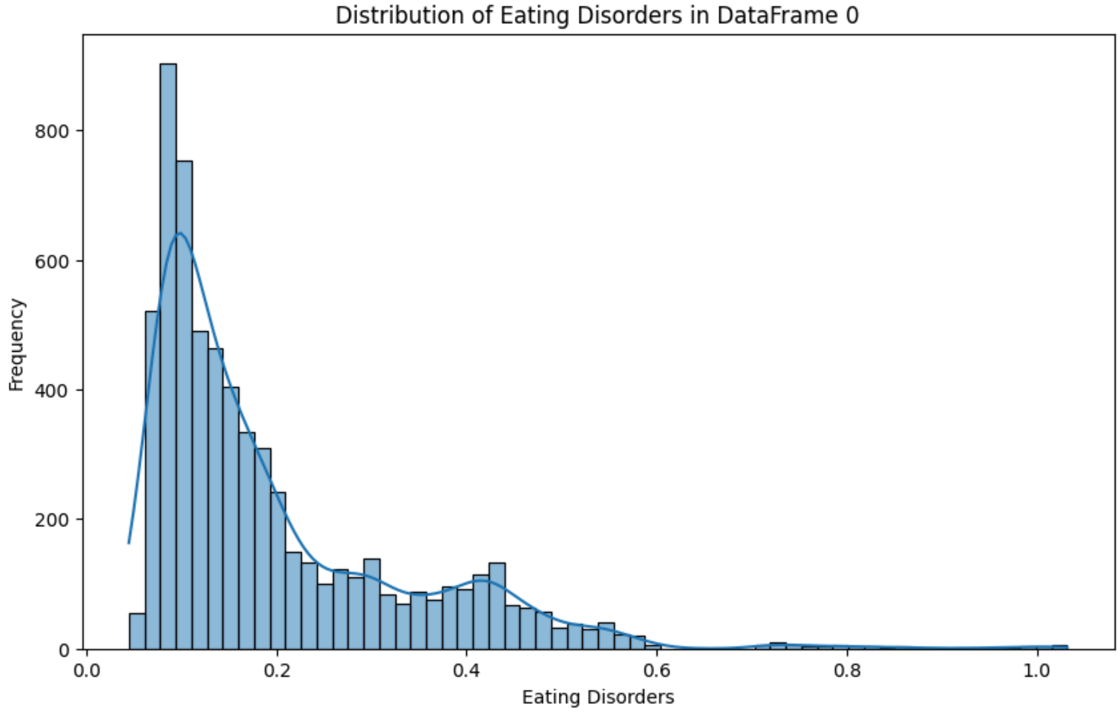
```
Plotting histograms for DataFrame 0
```



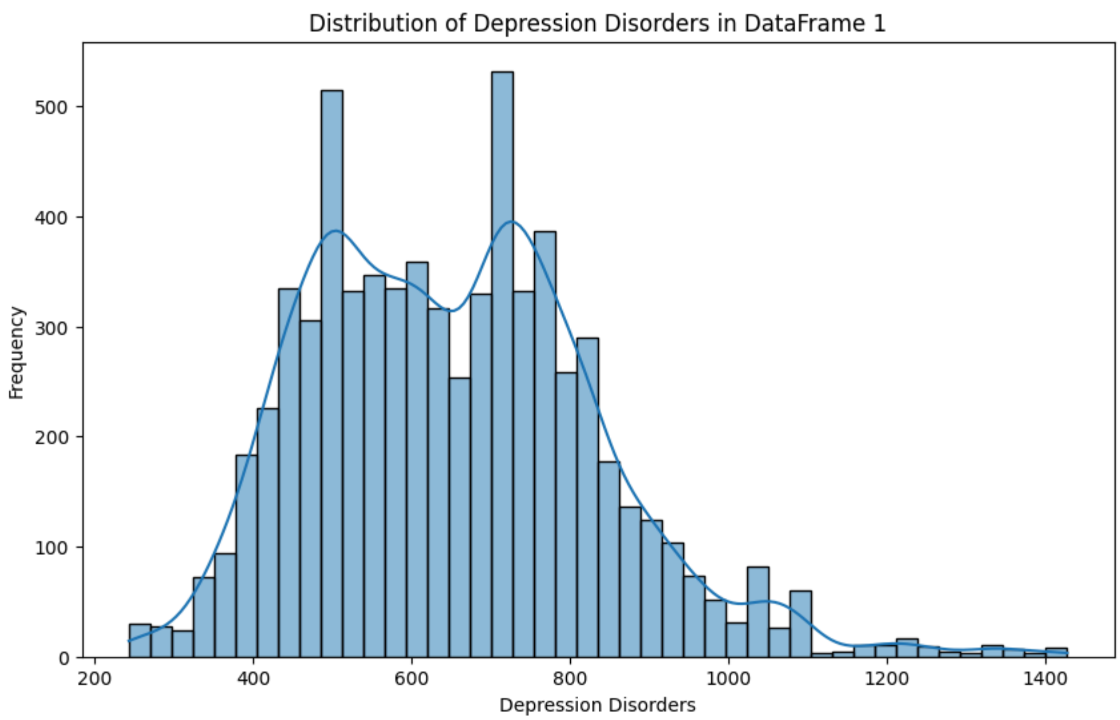


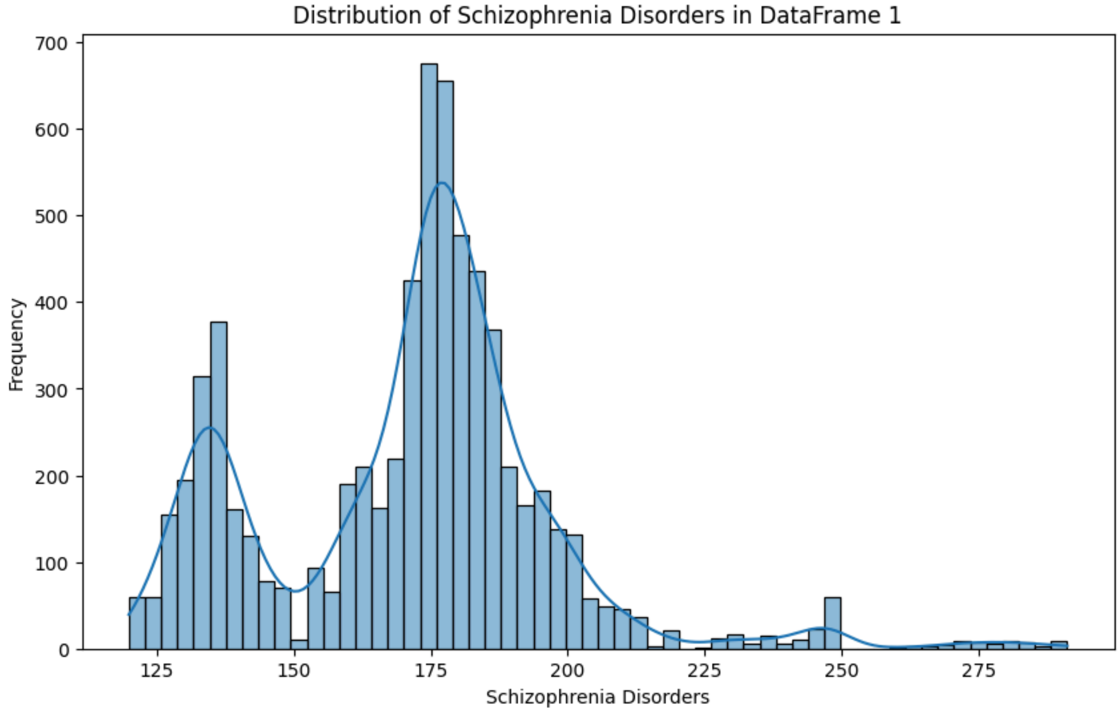

Distribution of Anxiety Disorders in DataFrame 0
Frequency
Anxiety Disorders
0
100
200
300
400
500
600
2
3
4
5
6
7
8

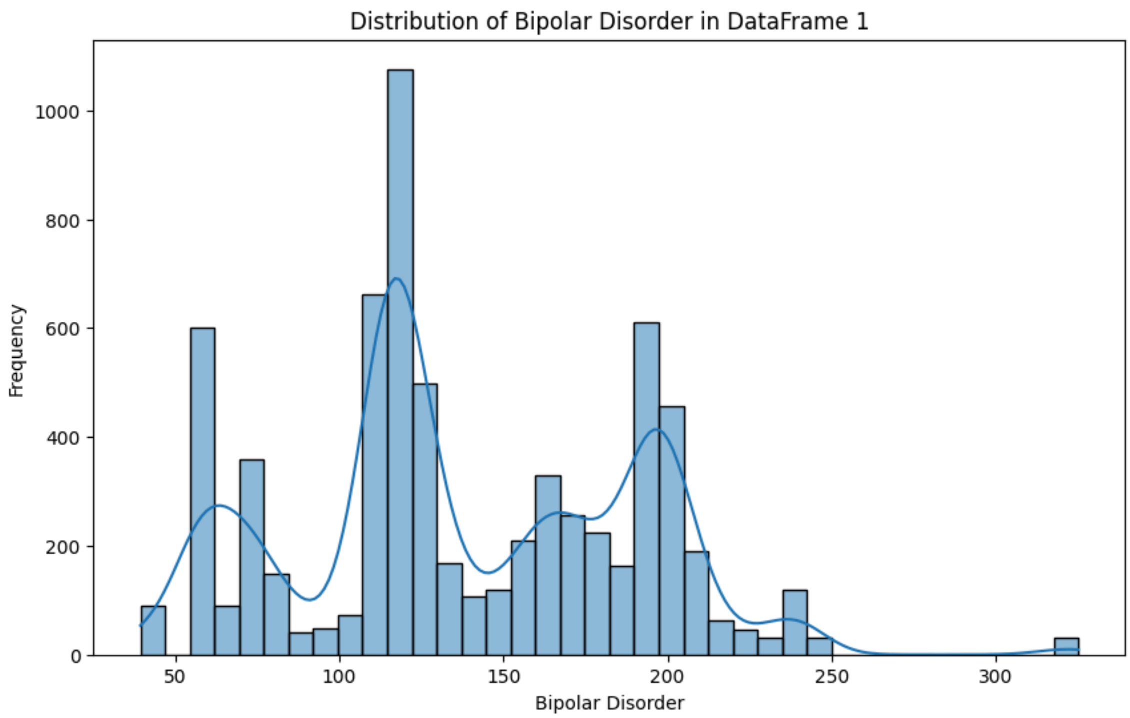

Distribution of Bipolar Disorders in DataFrame 0
Frequency
Bipolar Disorders
0
200
400
600
800
1000
1200
0.2
0.4
0.6
0.8
1.0
1.2
1.4

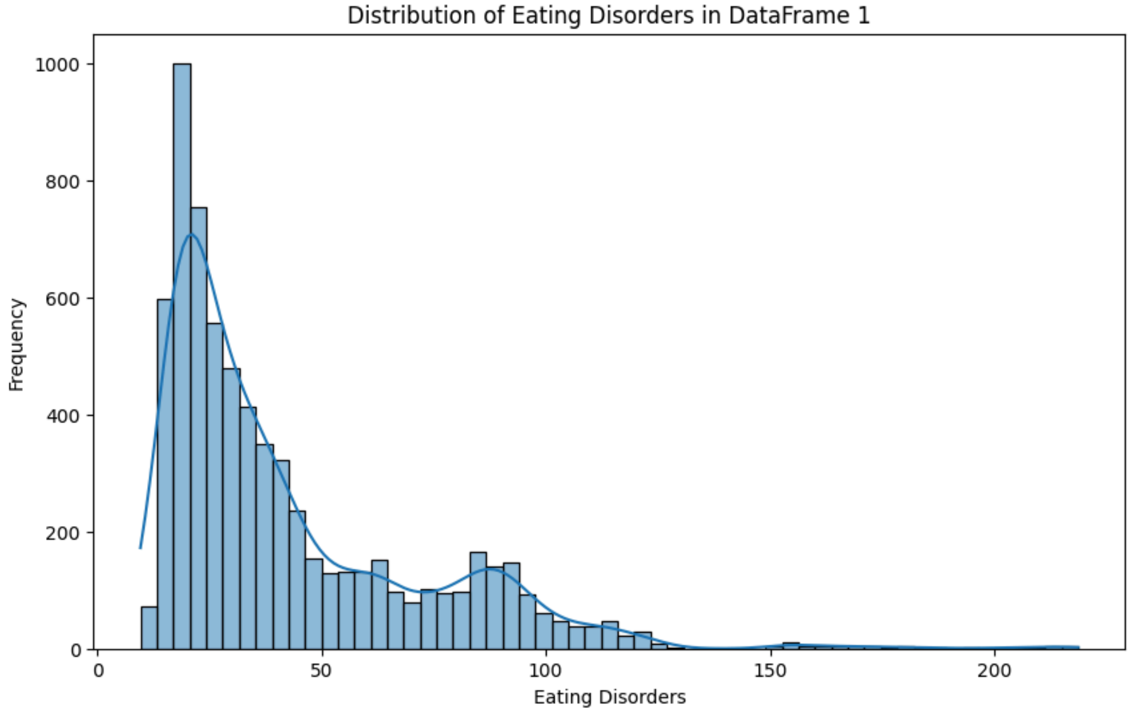


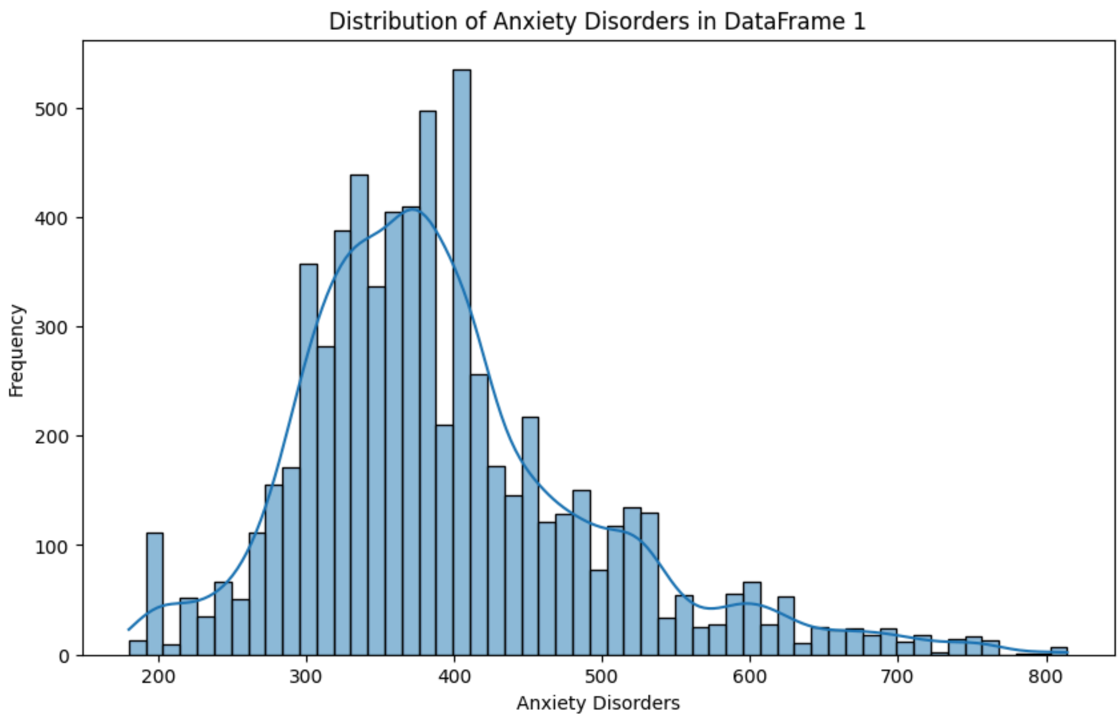
Plotting histograms for DataFrame 1



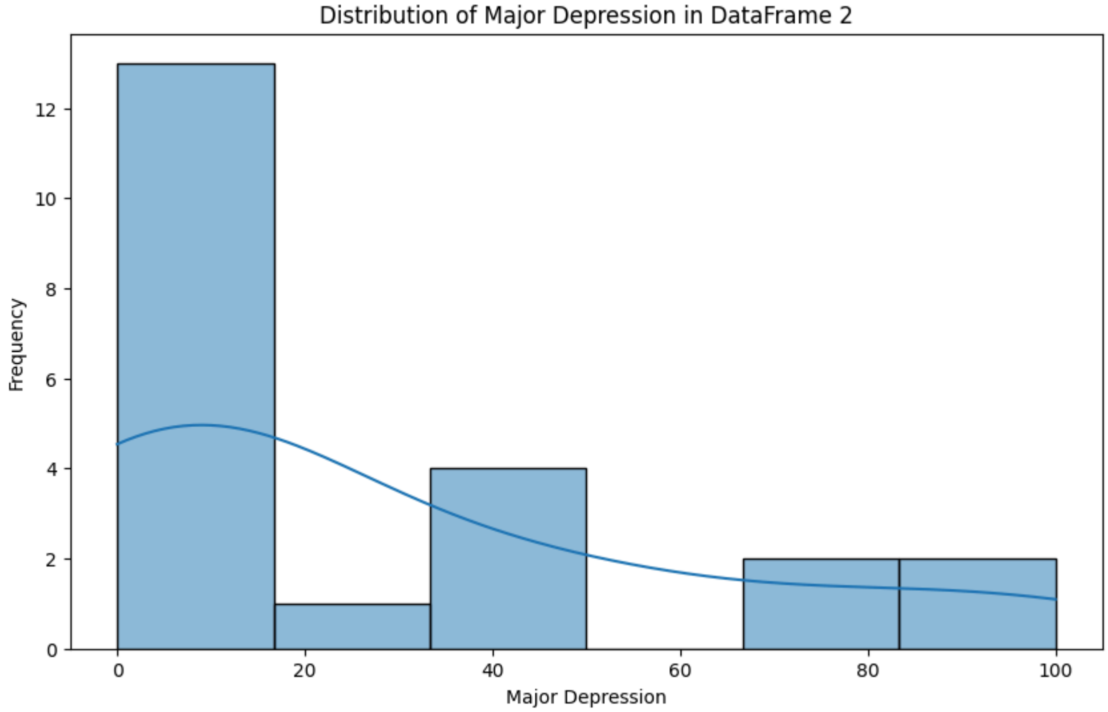

Distribution of Schizophrenia Disorders in DataFrame 1
Frequency
Schizophrenia Disorders
0
100
200
300
400
500
600
700
125
150
175
200
225
250
275


Distribution of Bipolar Disorder in DataFrame 1
Frequency
Bipolar Disorder
0
200
400
600
800
1000
50
100
150
200
250
300

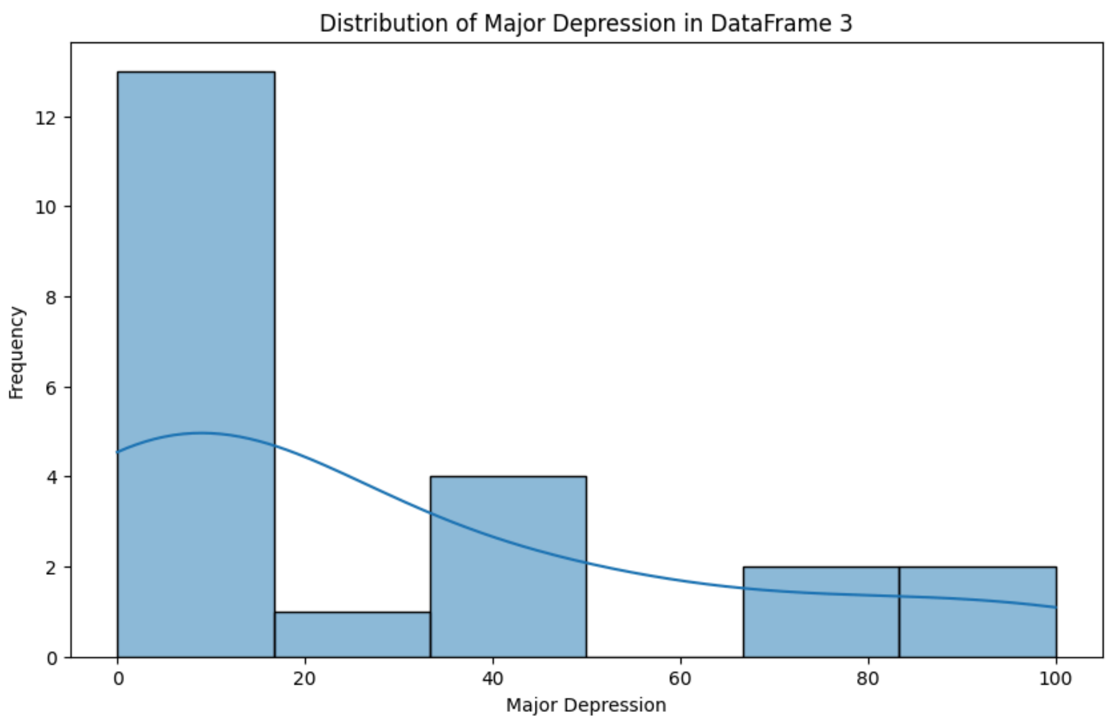


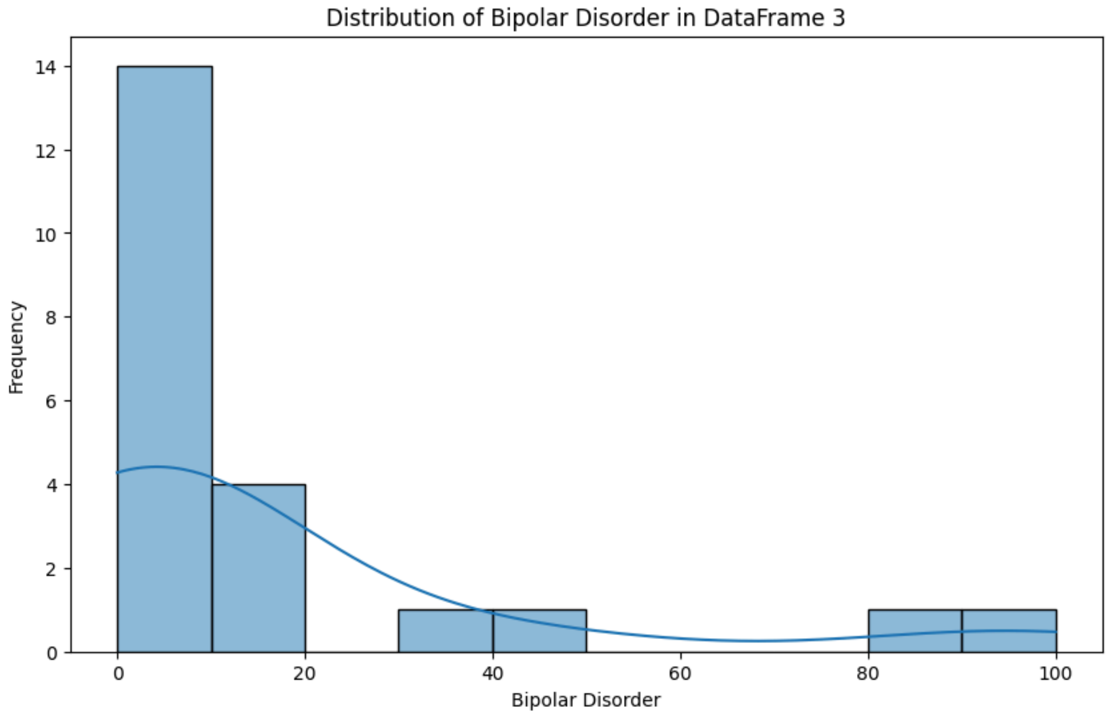


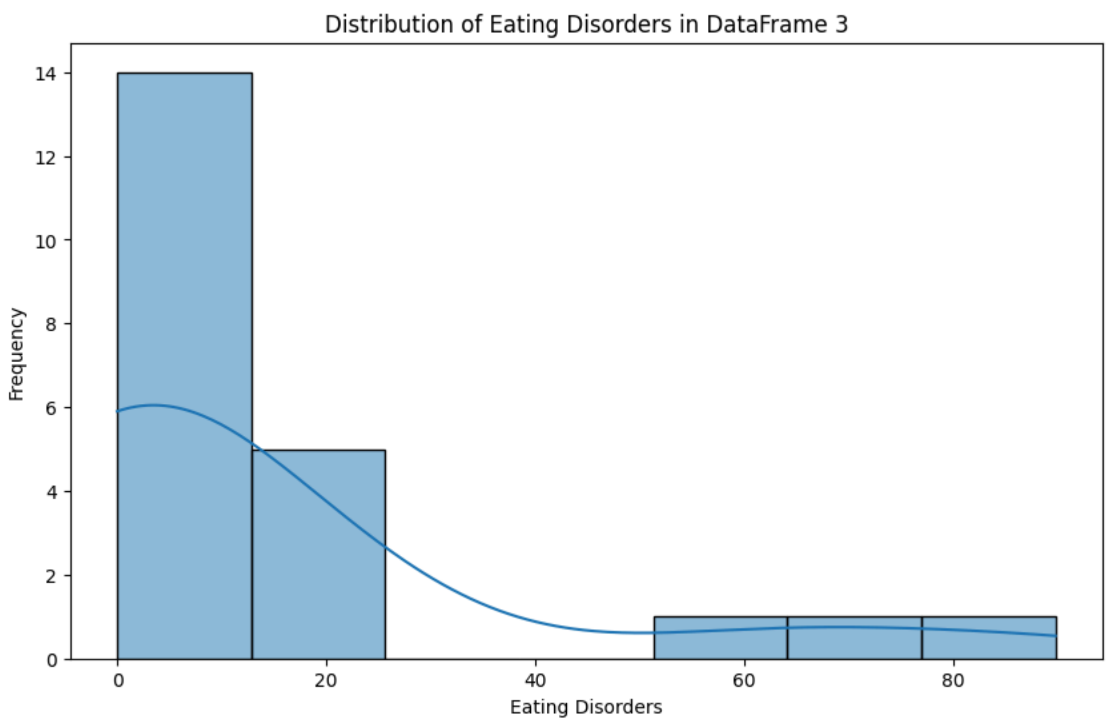

```
Plotting histograms for DataFrame 2
```

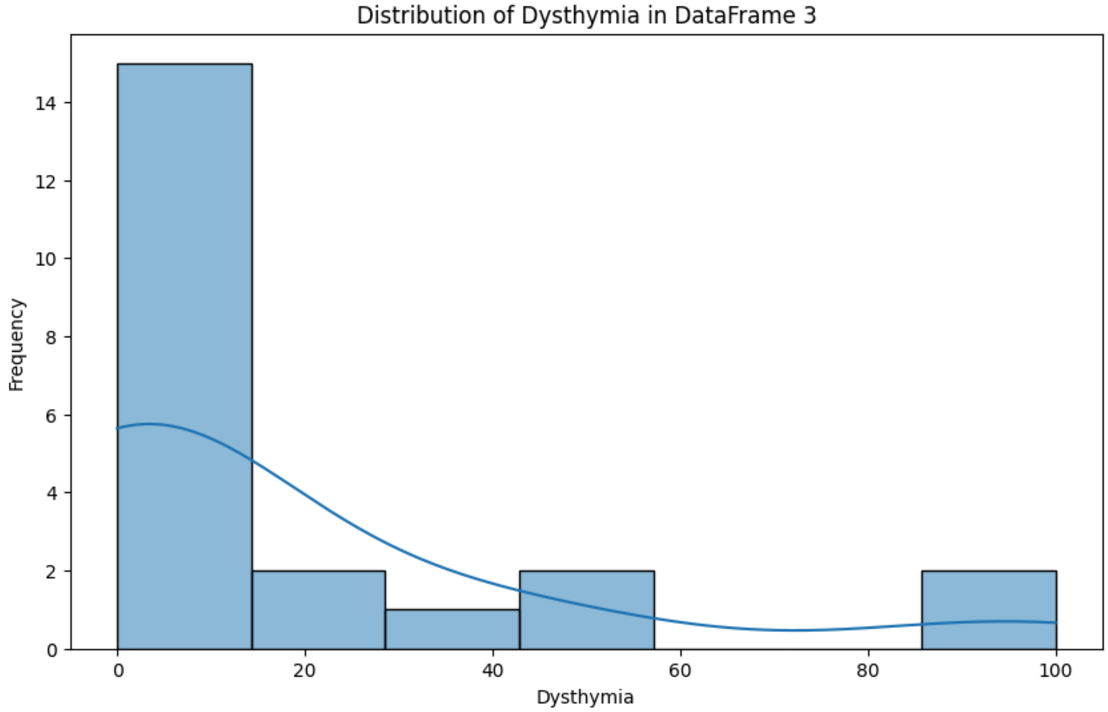


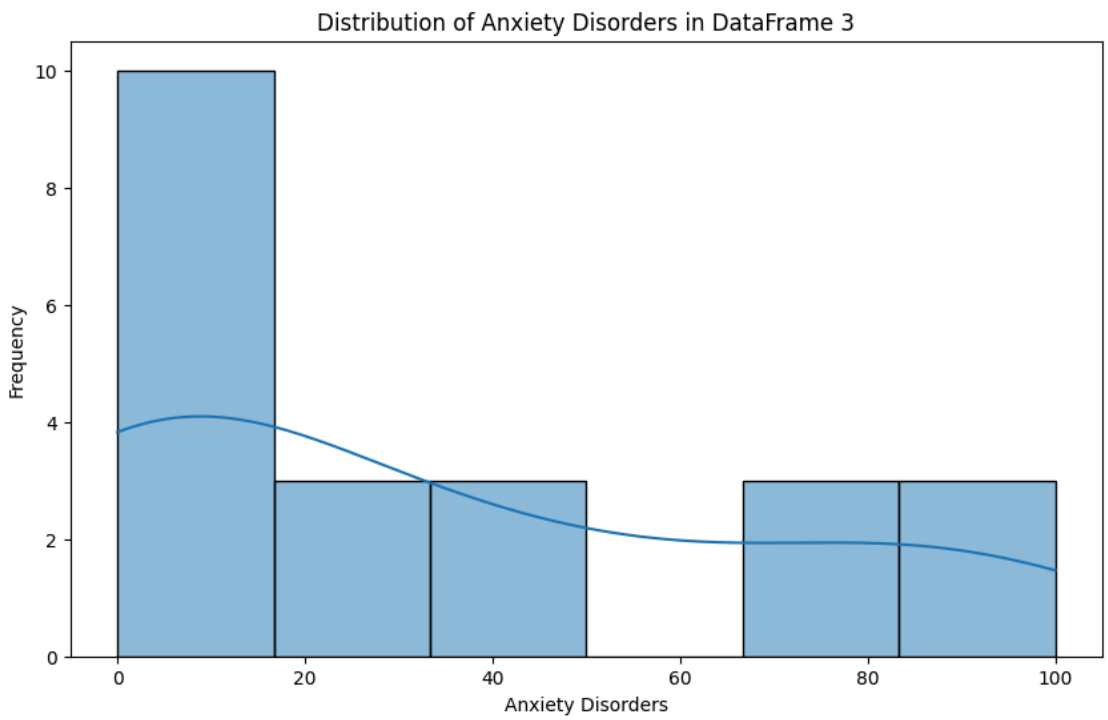

```
Plotting histograms for DataFrame 3
```



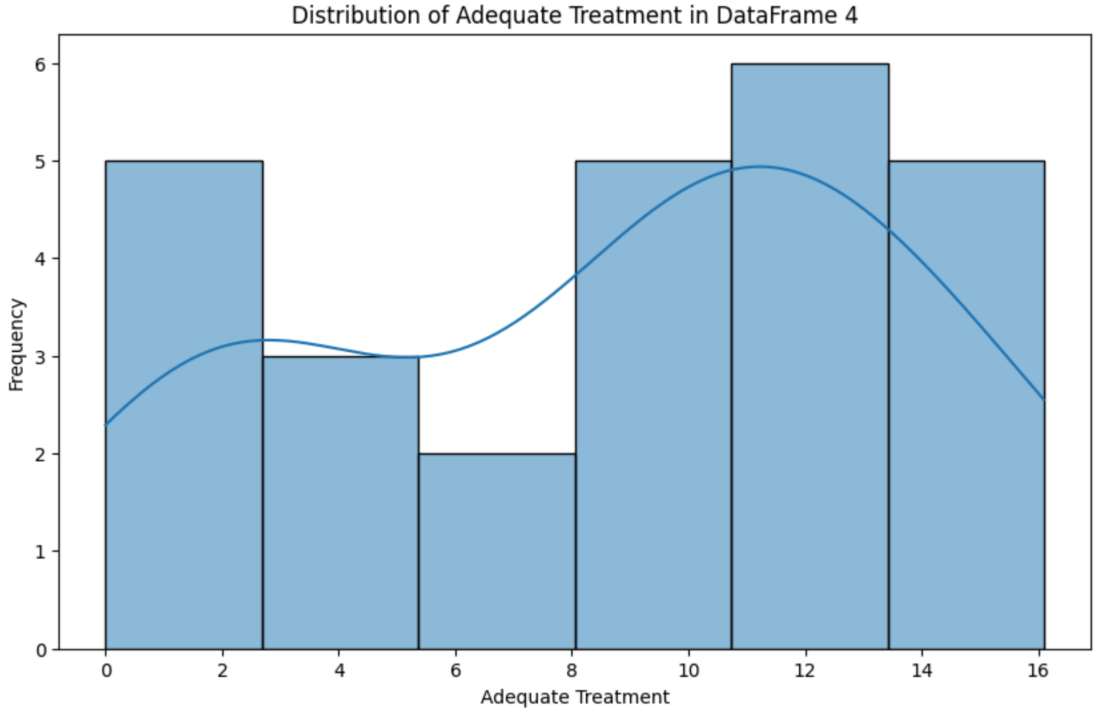

Distribution of Bipolar Disorder in DataFrame 3
Frequency
0
2
4
6
8
10
12
14
0
20
40
60
80
100
Bipolar Disorder

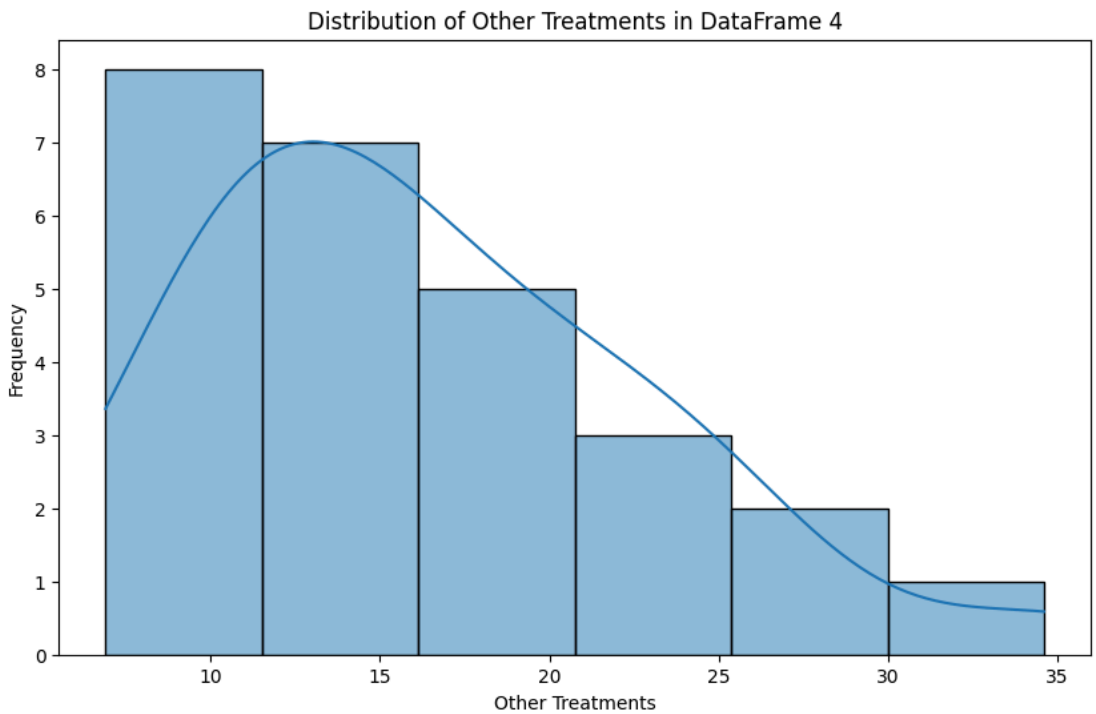

Distribution of Eating Disorders in DataFrame 3
Frequency
0
2
4
6
8
10
12
14
0
20
40
60
80
Eating Disorders

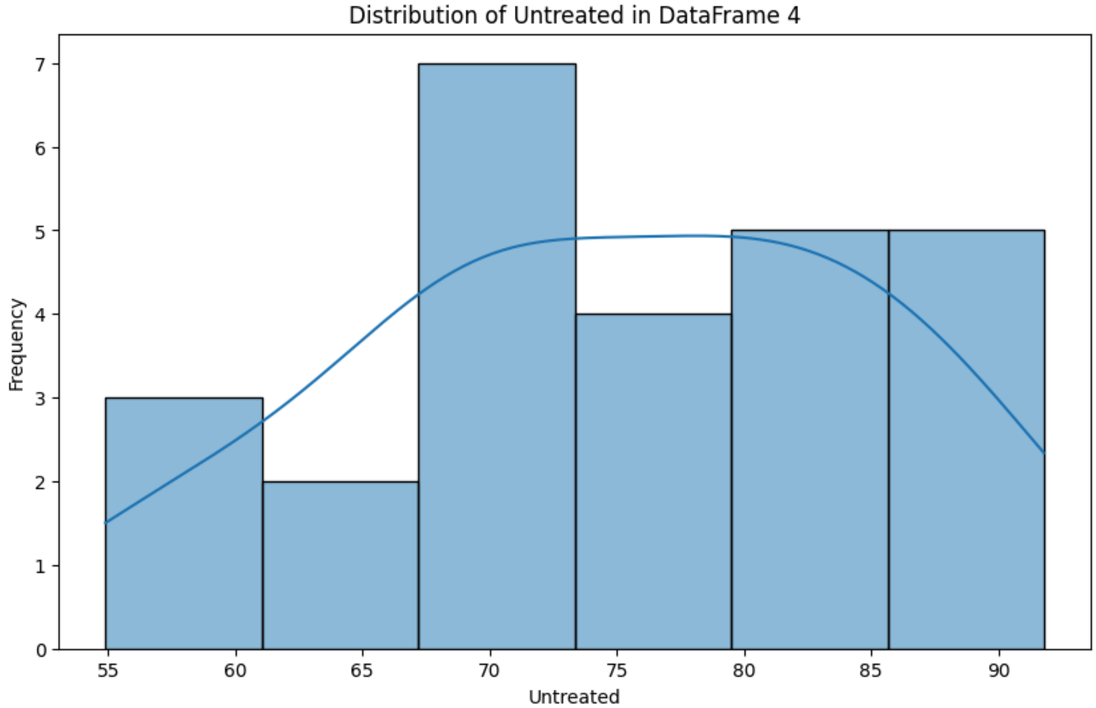




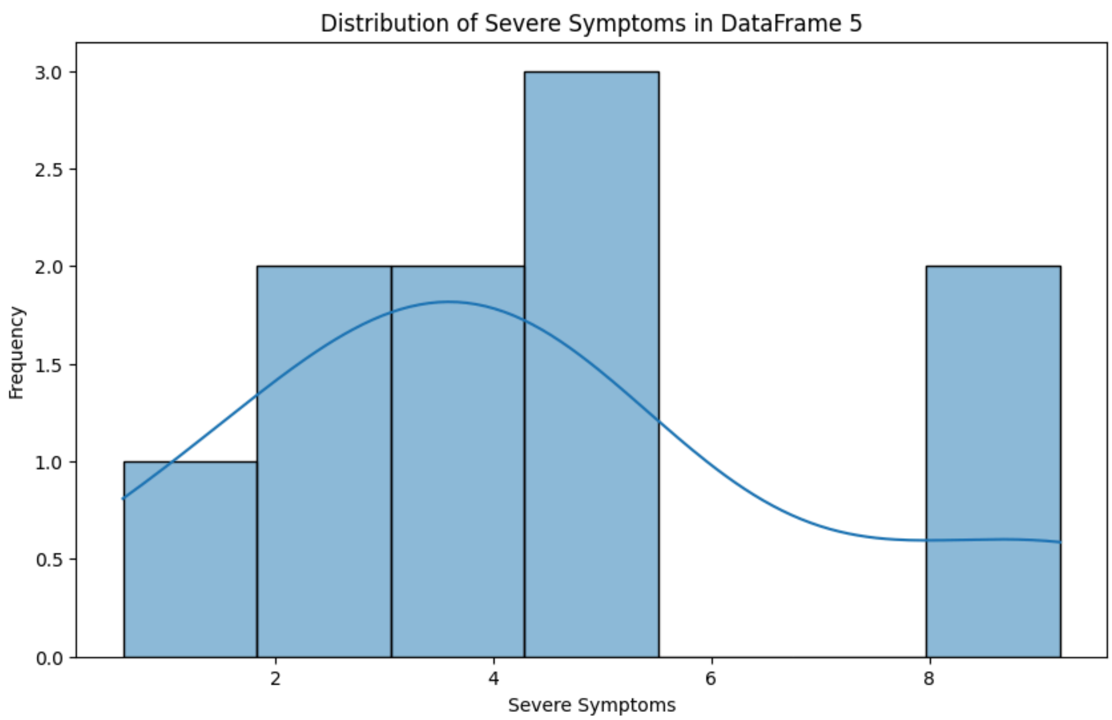

```
Plotting histograms for DataFrame 4
```

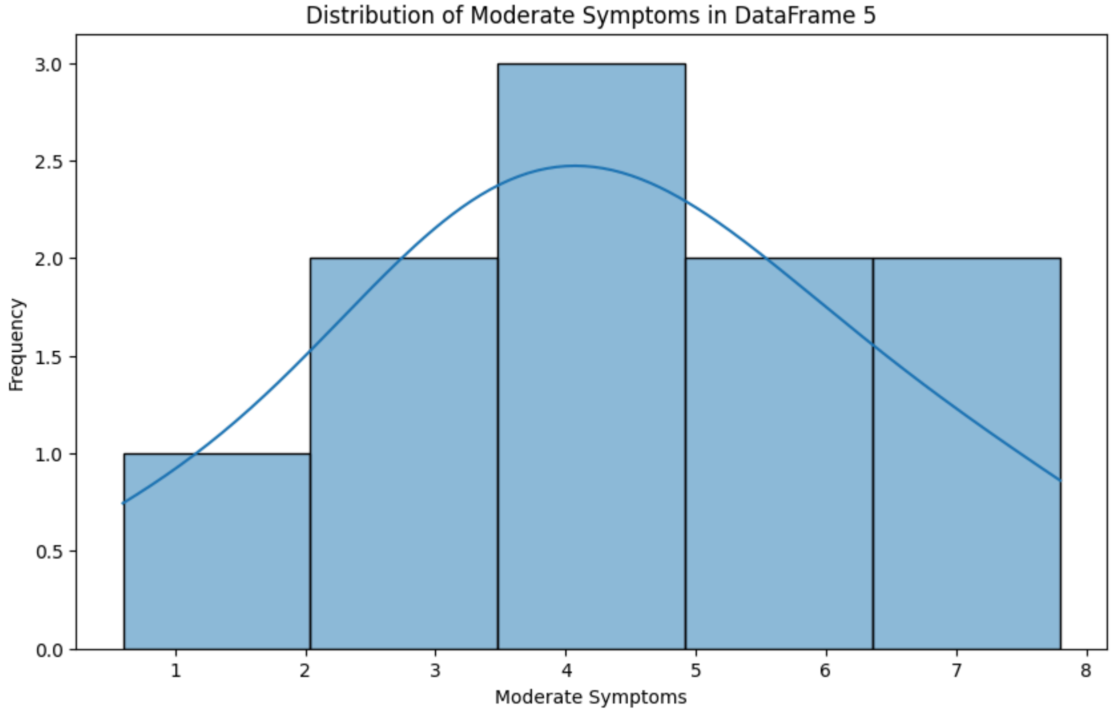

Distribution of Adequate Treatment in DataFrame 4
Frequency
Adequate Treatment
0 1 2 3 4 5 6
0 2 4 6 8 10 12 14 16

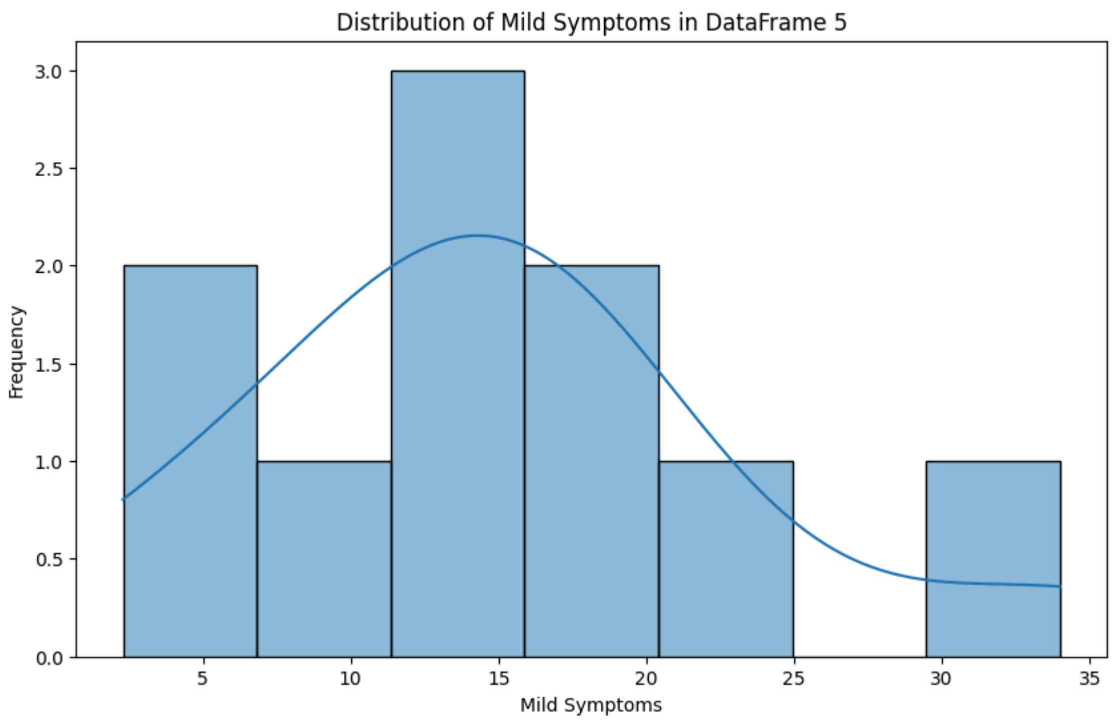

Distribution of Other Treatments in DataFrame 4
Frequency
Other Treatments
0 1 2 3 4 5 6 7 8
10 15 20 25 30 35

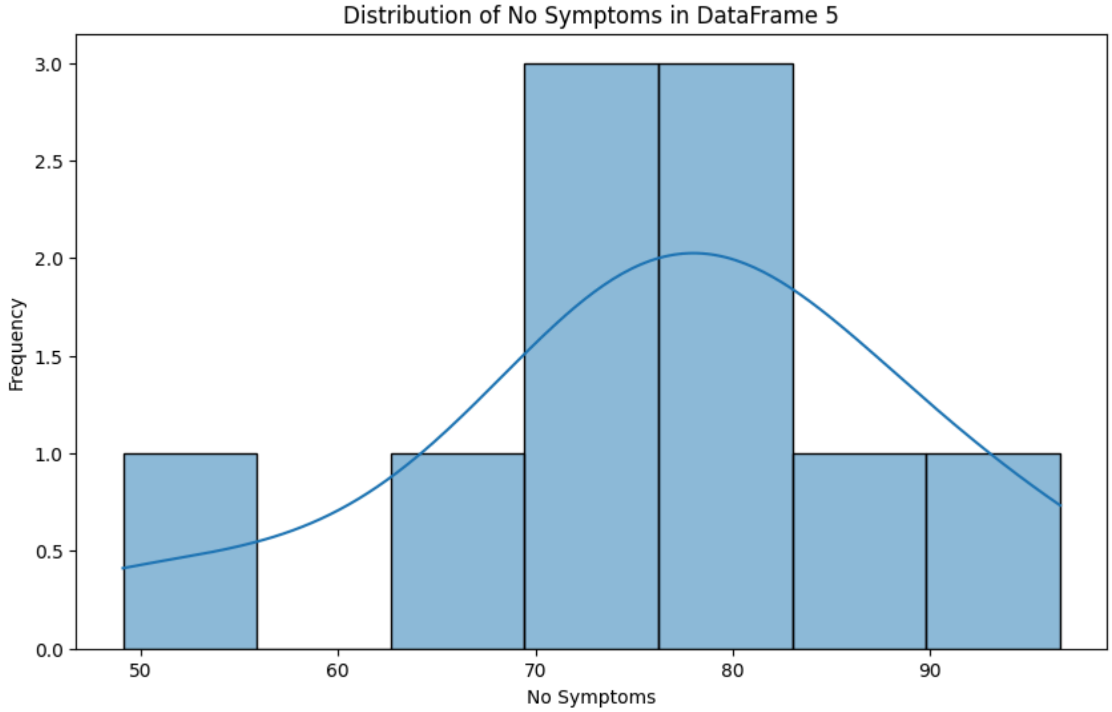



```
Plotting histograms for DataFrame 5
```

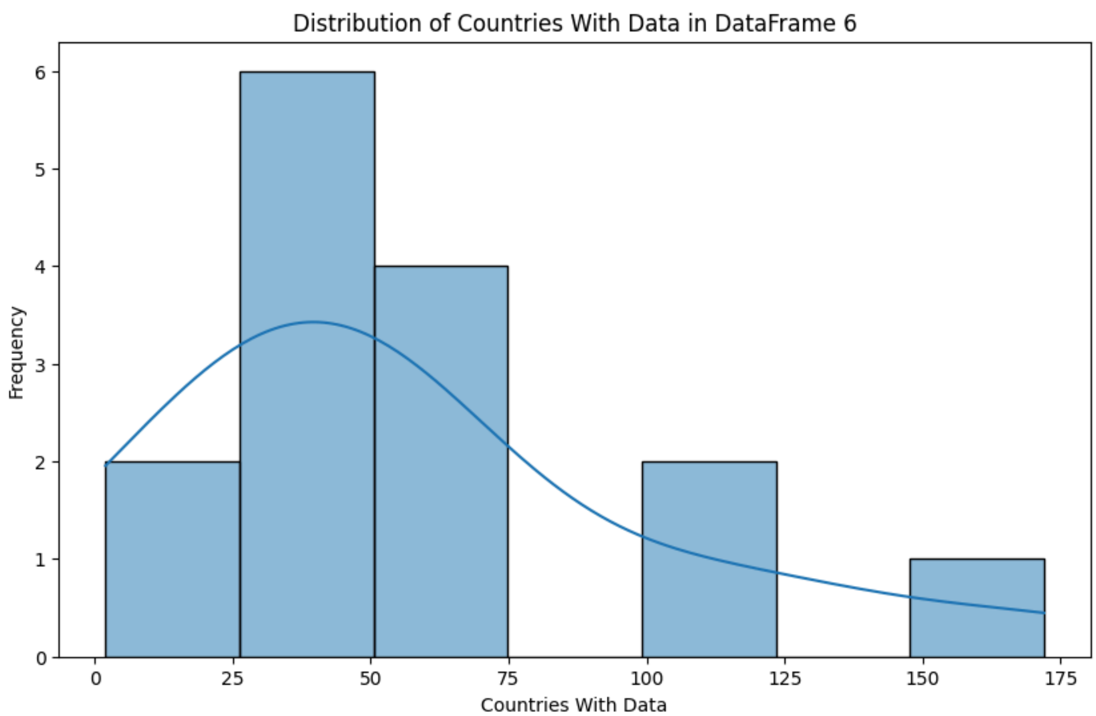



Distribution of Moderate Symptoms in DataFrame 5
Frequency
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1
2
3
4
5
6
7
8
Moderate Symptoms


Distribution of Mild Symptoms in DataFrame 5
Frequency
0.0
0.5
1.0
1.5
2.0
2.5
3.0
5
10
15
20
25
30
35
Mild Symptoms



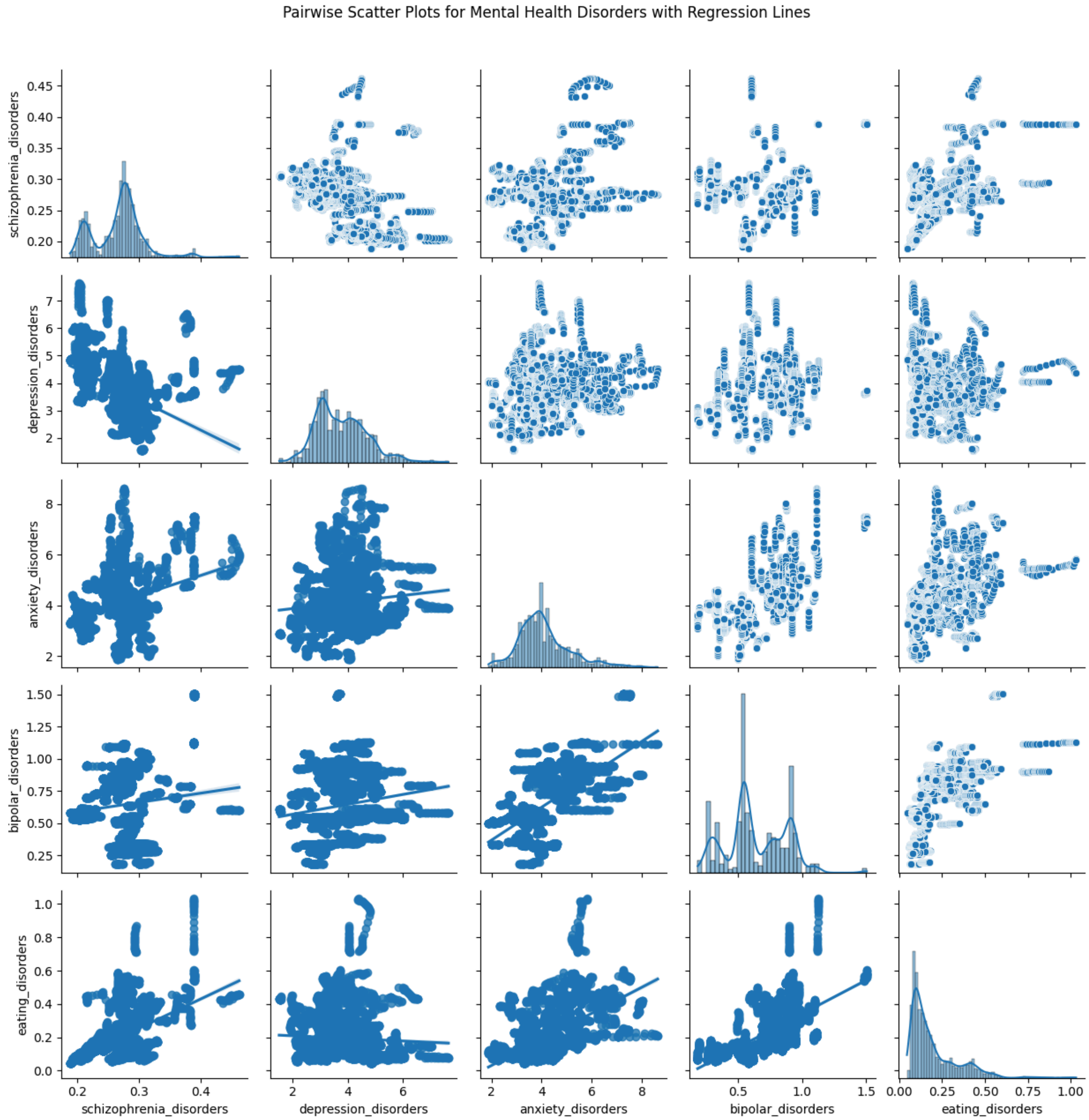

```
Plotting histograms for DataFrame 6
```



In [28]:
```
columns_for_correlation = ['schizophrenia_disorders',
                           'depression_disorders',
                           'anxiety_disorders',
                           'bipolar_disorders',
                           'eating_disorders']
```

Scatter Plots

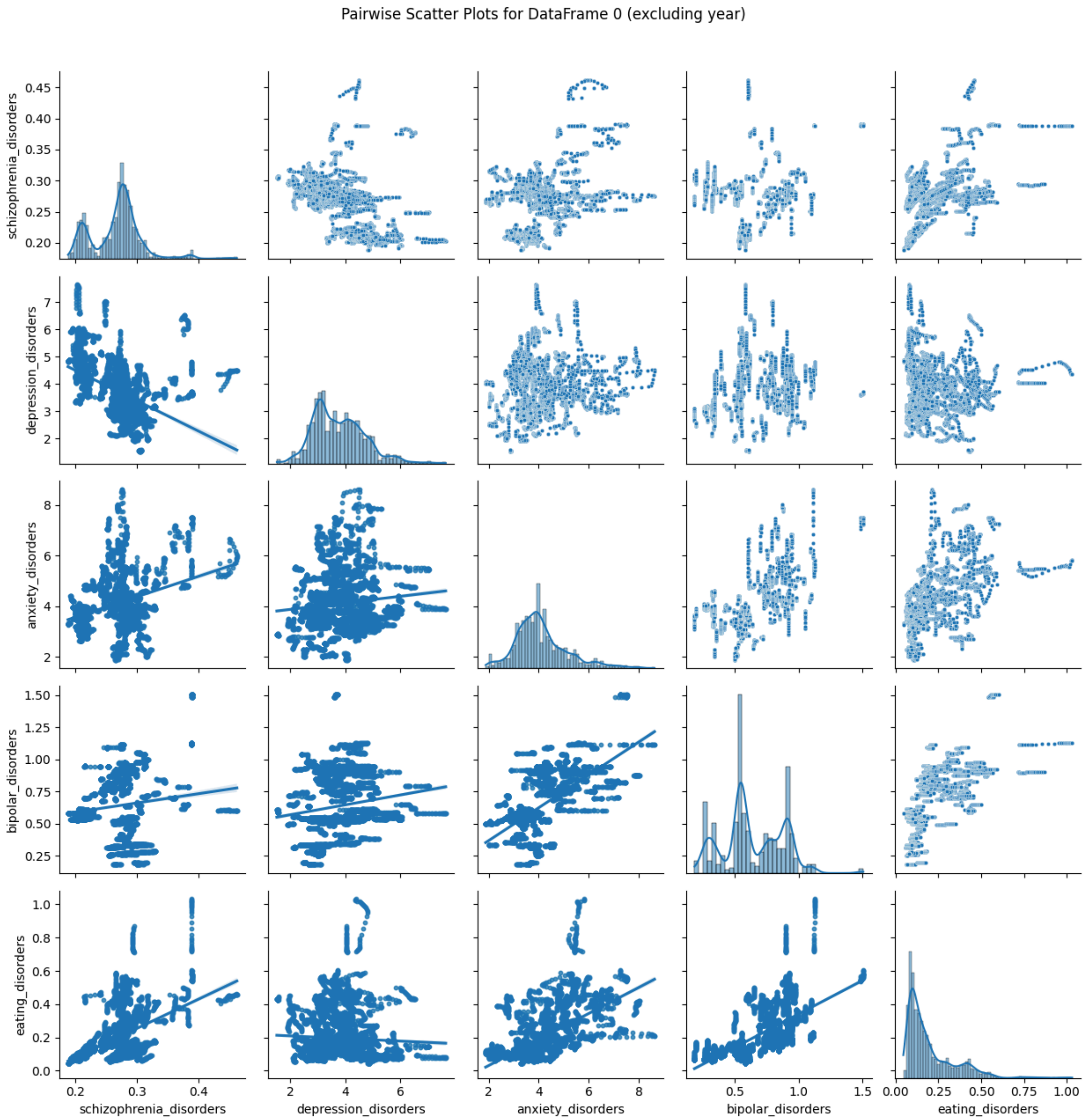
In [29]:
```
pair_grid = sns.PairGrid(dfs[0], vars=columns_for_correlation)
pair_grid.map_upper(sns.scatterplot)
pair_grid.map_lower(sns.regplot)
pair_grid.map_diag(sns.histplot, kde=True)
plt.suptitle('Pairwise Scatter Plots for Mental Health Disorders with Regression Lines', y=1.02)
plt.tight_layout()
plt.show()
# we need to make the points smaller I think.
```



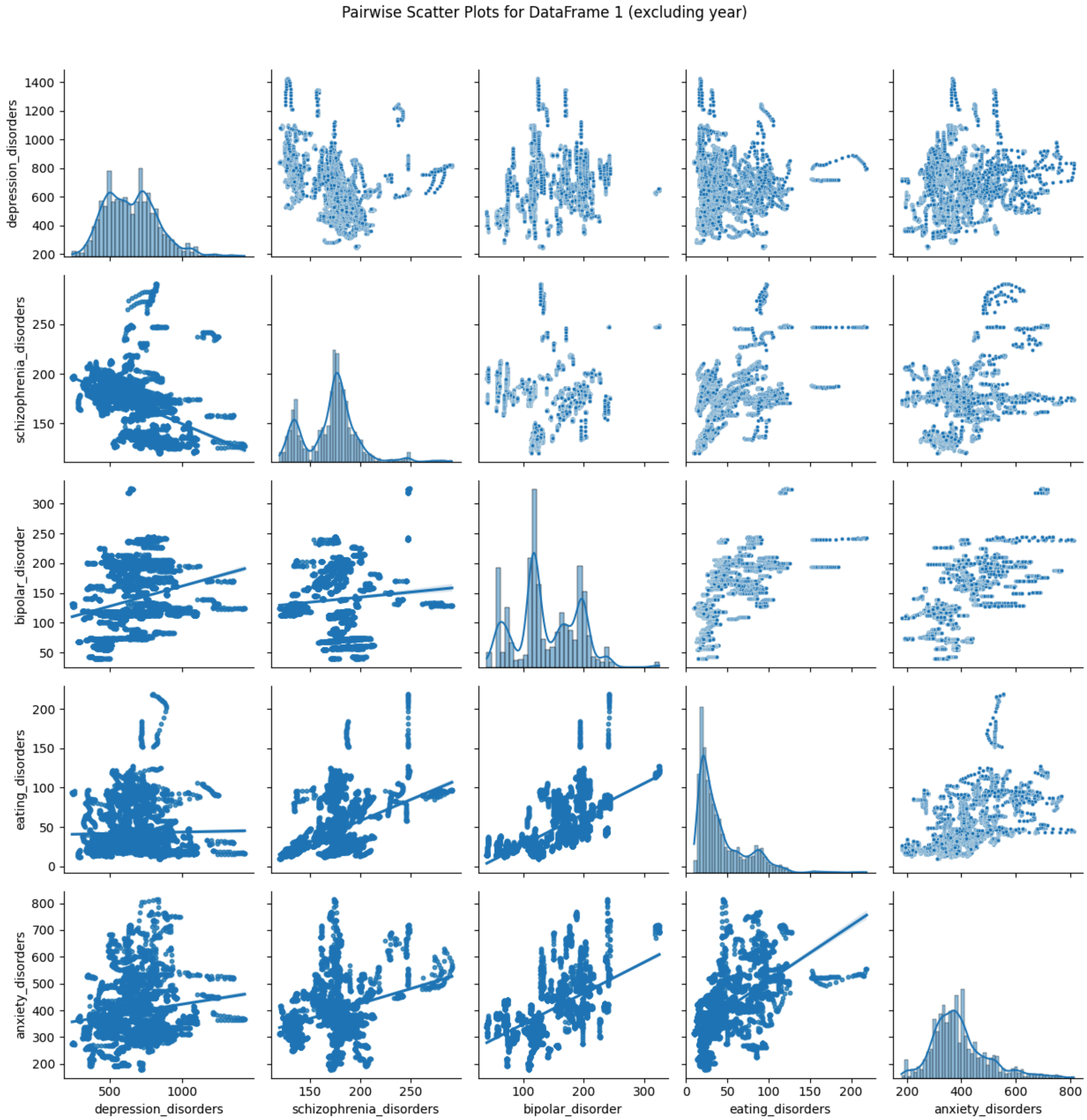
In [30]:
```python
# Generate more Scatter plots for more datasets
# Function to generate scatter plots for all numeric column pairs except 'year'
def plot_scatterplots(dfs):
  """
    Generates and displays scatter plots for all pairs of numeric columns
    in each DataFrame in the list, skipping the 'year' column if it exists.

    Parameters:
    -----------
    dfs : list of pandas.DataFrame
        A list of DataFrames to plot scatter plots from.
  """
  for i, df in enumerate(dfs):
    if i != len(dfs) - 1:
        print(f"\nPlotting scatter plots for DataFrame {i}")
        numeric_cols = df.select_dtypes(include='number').columns
        # Filter out the 'year' column if it exists
        plot_cols = [col for col in numeric_cols if col != 'year']

        # Create a PairGrid for all pairwise scatter plots of numeric columns
        # Use a subset of columns if there are too many to avoid excessive plotting
        if len(plot_cols) > 1:
            # Limit the number of columns for plotting if it's too large
            max_cols_for_pairplot = 10
            if len(plot_cols) > max_cols_for_pairplot:
                print(f"DataFrame {i} has more than {max_cols_for_pairplot} numeric columns (excluding year). Plotting a subse
                # You might want to select specific columns here based on relevance
                # For now, just take the first max_cols_for_pairplot columns
                cols_to_plot = plot_cols[:max_cols_for_pairplot]
            else:
```

```
            cols_to_plot = plot_cols

        if len(cols_to_plot) > 1:
            pair_grid = sns.PairGrid(df, vars=cols_to_plot)
            # Use regplot for regression lines
            pair_grid.map_upper(sns.scatterplot, s=10) # smaller points
            pair_grid.map_lower(sns.regplot, scatter_kws={'s': 10}) # smaller points
            pair_grid.map_diag(sns.histplot, kde=True)

            plt.suptitle(f'Pairwise Scatter Plots for DataFrame {i} (excluding year)', y=1.02)
            plt.tight_layout()
            plt.show()
        else:
            print(f"DataFrame {i} has only one numeric column to plot scatter plots for (excluding year). Skipping scatter
    else:
        print(f"DataFrame {i} has less than two numeric columns to plot scatter plots for (excluding year). Skipping scatt

# Call the function to plot scatter plots for all datasets, excluding the year column
plot_scatterplots(dfs)
```

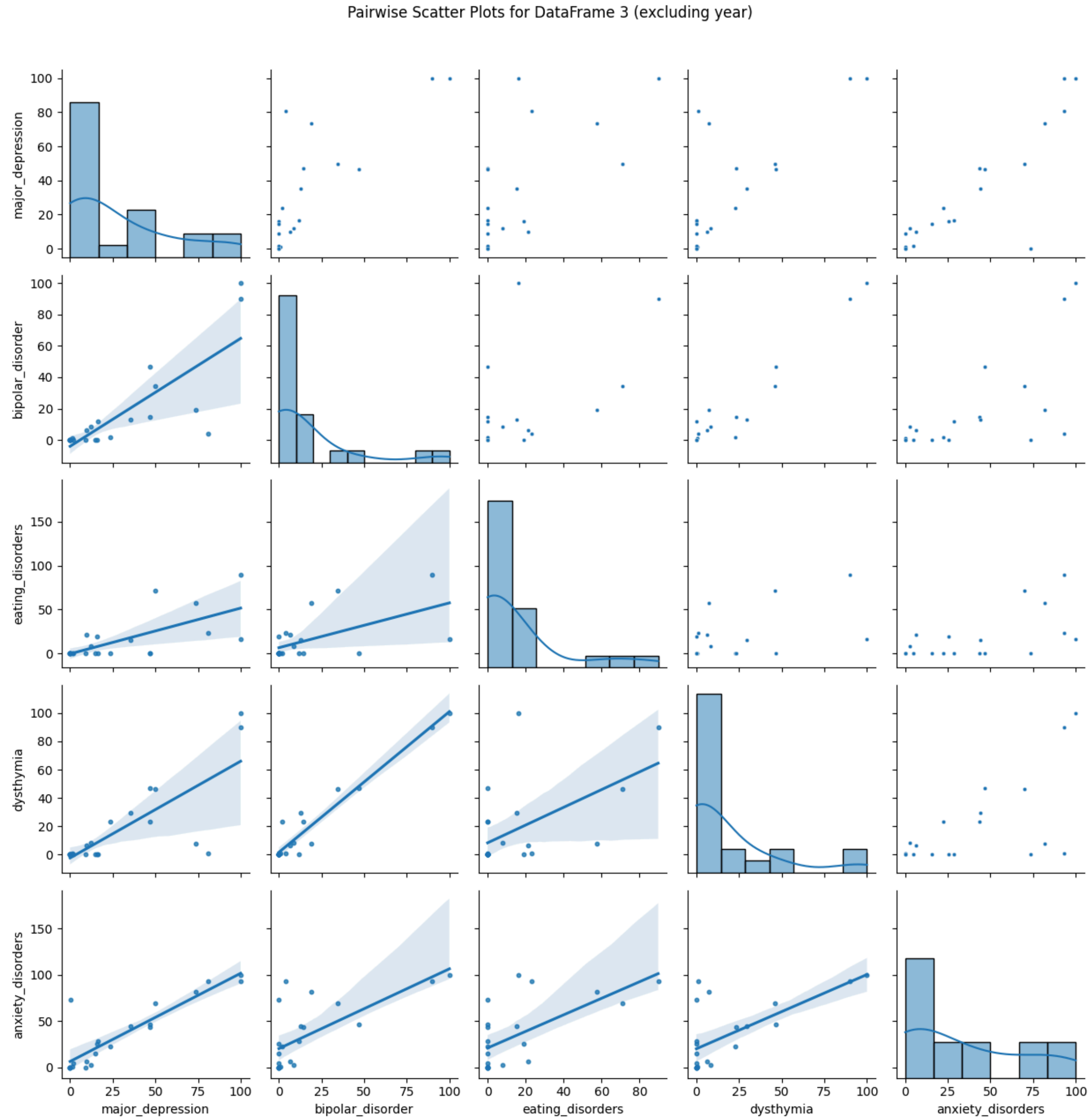
```
Plotting scatter plots for DataFrame 0
```



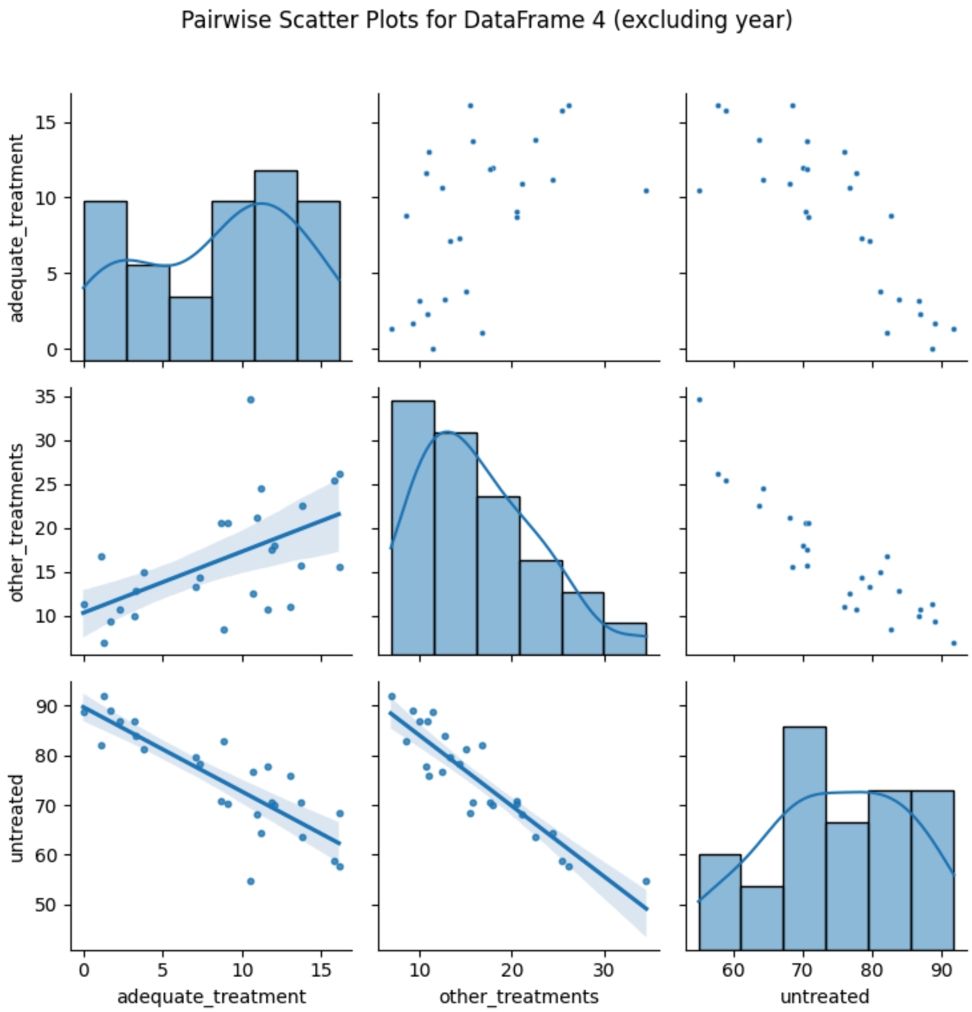

```
Plotting scatter plots for DataFrame 1
```



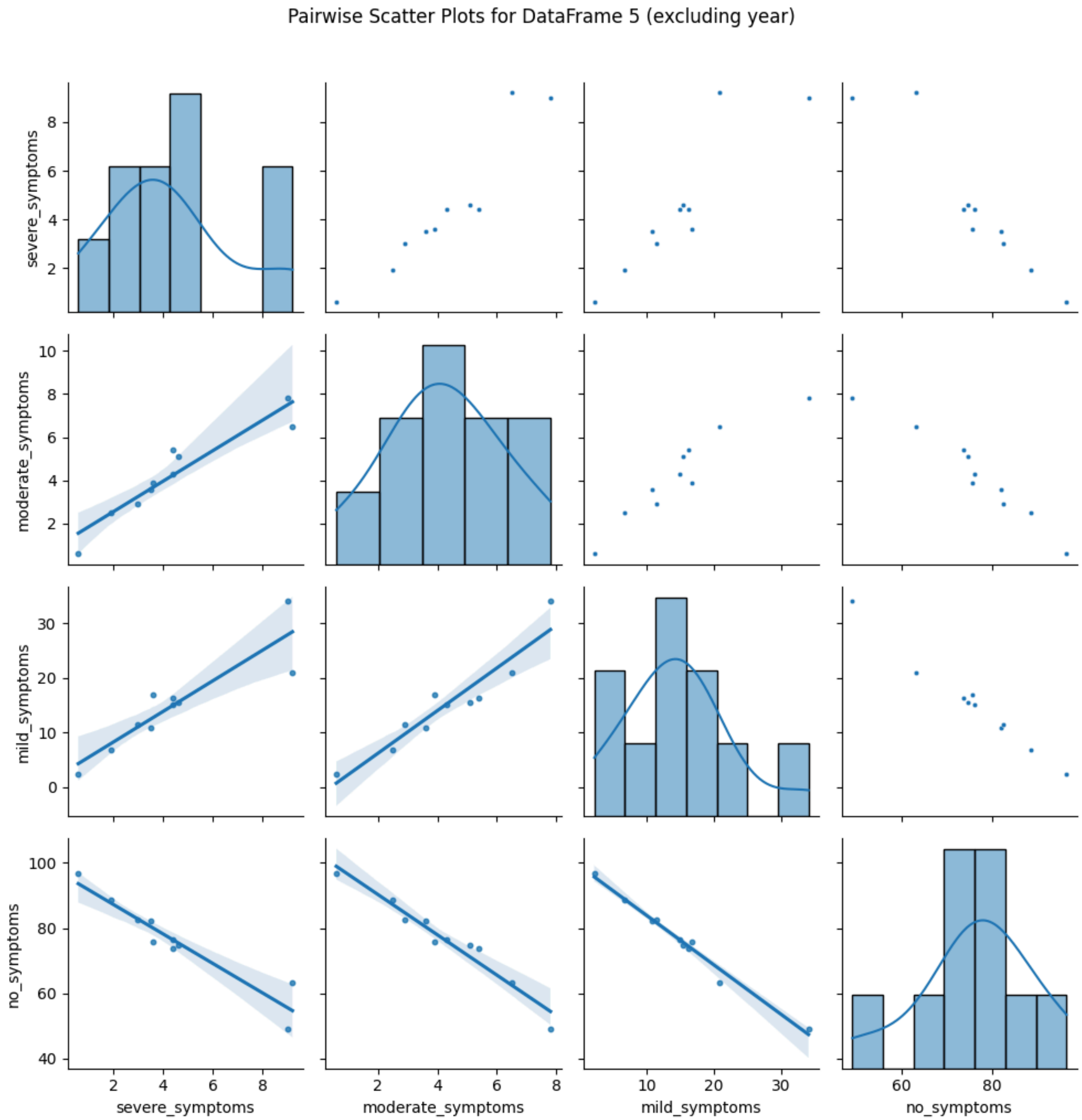

```
Plotting scatter plots for DataFrame 2
DataFrame 2 has less than two numeric columns to plot scatter plots for (excluding year). Skipping scatter plot.

Plotting scatter plots for DataFrame 3
```




```
Plotting scatter plots for DataFrame 4
```




```
Plotting scatter plots for DataFrame 5
```




```
Plotting scatter plots for DataFrame 6
DataFrame 6 has less than two numeric columns to plot scatter plots for (excluding year). Skipping scatter plot.
```

Correlation and Covariance Heat Map

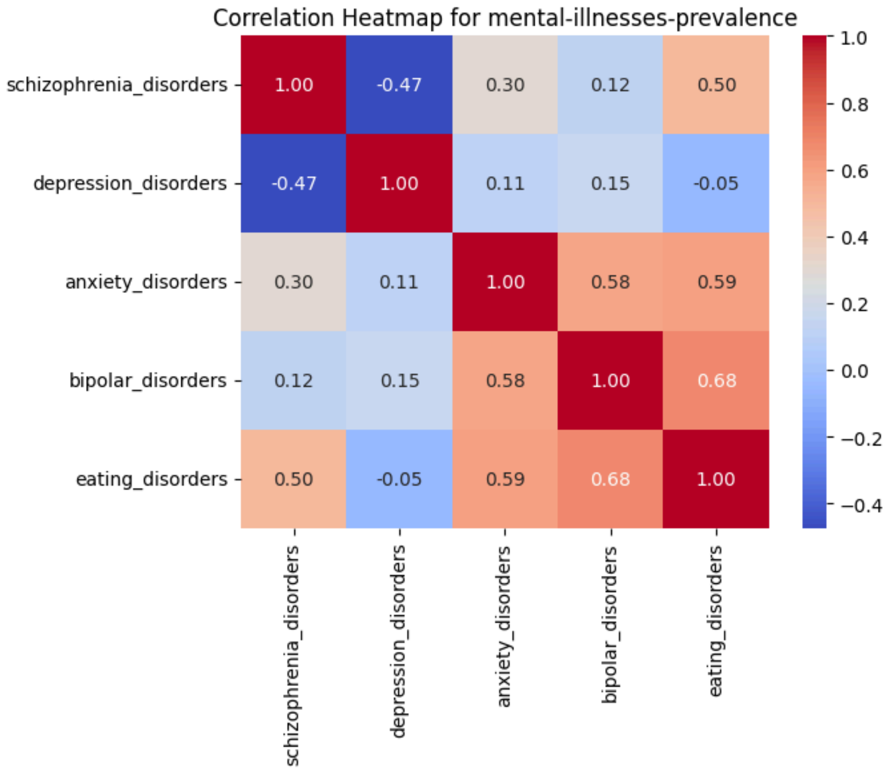
Only applied on dataset #1 and #2

```
High correlation and high covariance -> strong linear relationship with
similar variance scale. -> could be used for linear regression

High correlation but very low covariance m ay indicate the variables vary
 similarly in pattern but not in magnitude.

Very low covariance (close to zero) -> almost no shared variance even if
the correlation is moderate.
```

Dataset 1 Analysis: mental-illnesses-prevalence

In [31]:

```
# Calculate correlation matrix for the first DataFrame
corr_matrix = df_skip.corr()

#draw heatmap for correlation matrix for tables that has more than 2 columns
sns.heatmap(corr_matrix, annot=True, fmt='.2f', cmap='coolwarm')
plt.title('Correlation Heatmap for mental-illnesses-prevalence')
plt.show()
```



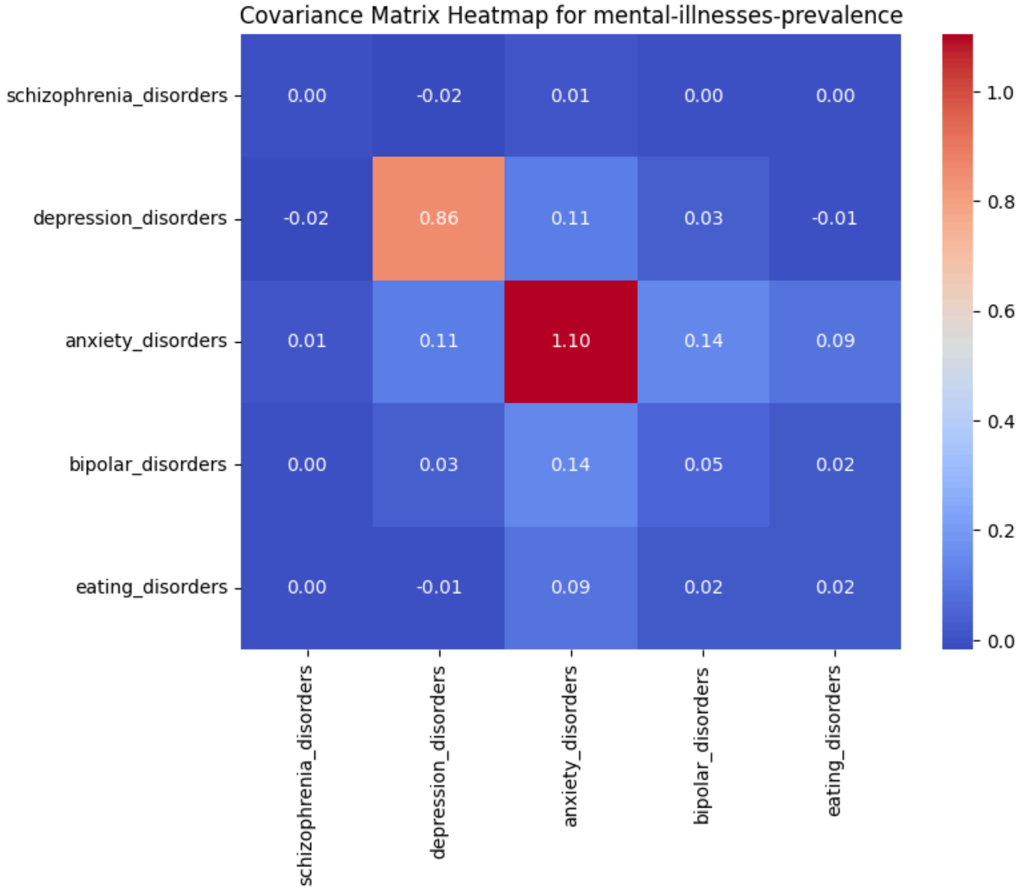
Column	Strongest Correlation With	Correlation Value	Independent?
`schizophrenia_disorders`	`eating_disorders`	**+0.50**	No
`depression_disorders`	`schizophrenia_disorders`	**–0.47**	Weak/moderate
`anxiety_disorders`	`eating_disorders` / `bipolar_disorders`	**~0.58–0.59**	No
`bipolar_disorders`	`eating_disorders`	**+0.68**	No
`eating_disorders`	`bipolar_disorders`	**+0.68**	No

depression_disorders has weaker correlations with the rest (e.g., only –0.47 with schizophrenia and near-zero with others), so it's the most independent in this set.

All others are moderately correlated, especially with eating disorders and bipolar disorders.

Covariance Matrix

In [32]:
```
cov_matrix = df_skip.cov()

plt.figure(figsize=(8, 6))
sns.heatmap(cov_matrix, annot=True, fmt=".2f", cmap='coolwarm')
plt.title("Covariance Matrix Heatmap for mental-illnesses-prevalence")
plt.show()
```



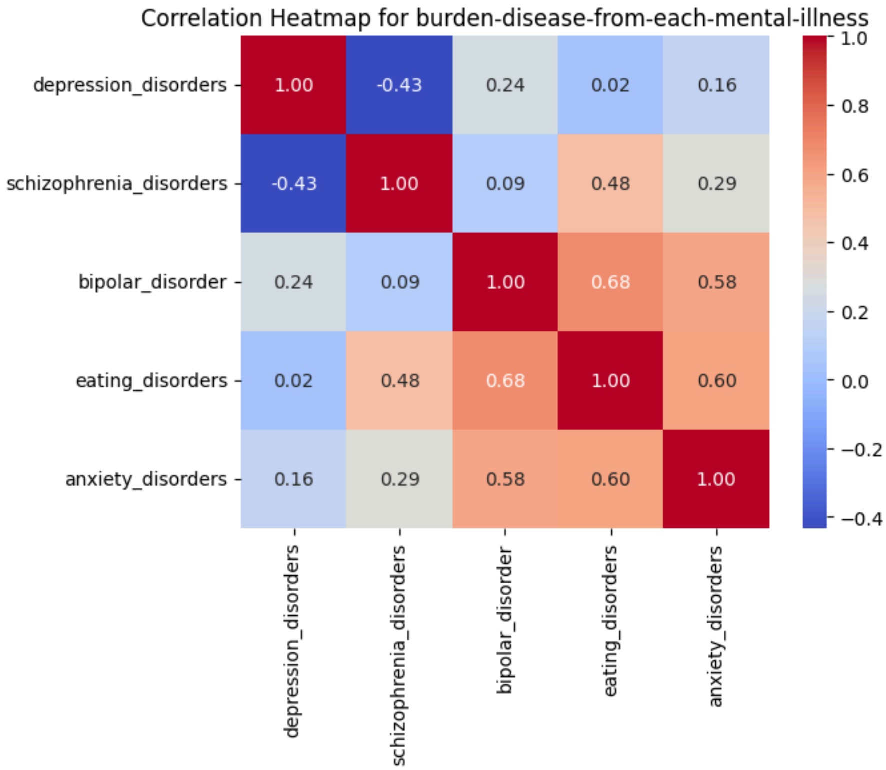
In [33]:
```
# Find strong relations based on correlation and covariance
print(find_strong_relation(corr_matrix, cov_matrix))
```

```
                Variable 1        Variable 2  Correlation  Covariance
0  schizophrenia_disorders  eating_disorders     0.500656    0.002728
1        anxiety_disorders  bipolar_disorders    0.576230    0.141284
2        anxiety_disorders  eating_disorders     0.594511    0.086427
3        bipolar_disorders  eating_disorders     0.677927    0.021895
```

Variable Pair	Correlation	Covariance	Interpretation
bipolar_disorders & eating_disorders	**0.678**	**0.0219**	Strong correlation; moderate covariance — they move together well and on a similar scale.
anxiety_disorders & eating_disorders	**0.595**	**0.0864**	Also strongly related, and the high covariance shows they vary together with similar units.
anxiety_disorders & bipolar_disorders	**0.576**	**0.1413**	Strongest covariance in this list -> similar unit spread and mutual variation.
schizophrenia_disorders & eating_disorders	**0.501**	**0.0027**	Moderate correlation, but **very small covariance** — may differ in scale significantly.

Dataset 2 Analysis: burden-disease-from-each-mental-illness

In [34]:
```
corr_matrix2 = df_skip2.corr()

#draw heatmap for correlation matrix for tables that has more than 2 columns
sns.heatmap(corr_matrix2, annot=True, fmt='.2f', cmap='coolwarm')
plt.title('Correlation Heatmap for burden-disease-from-each-mental-illness')
plt.show()
```

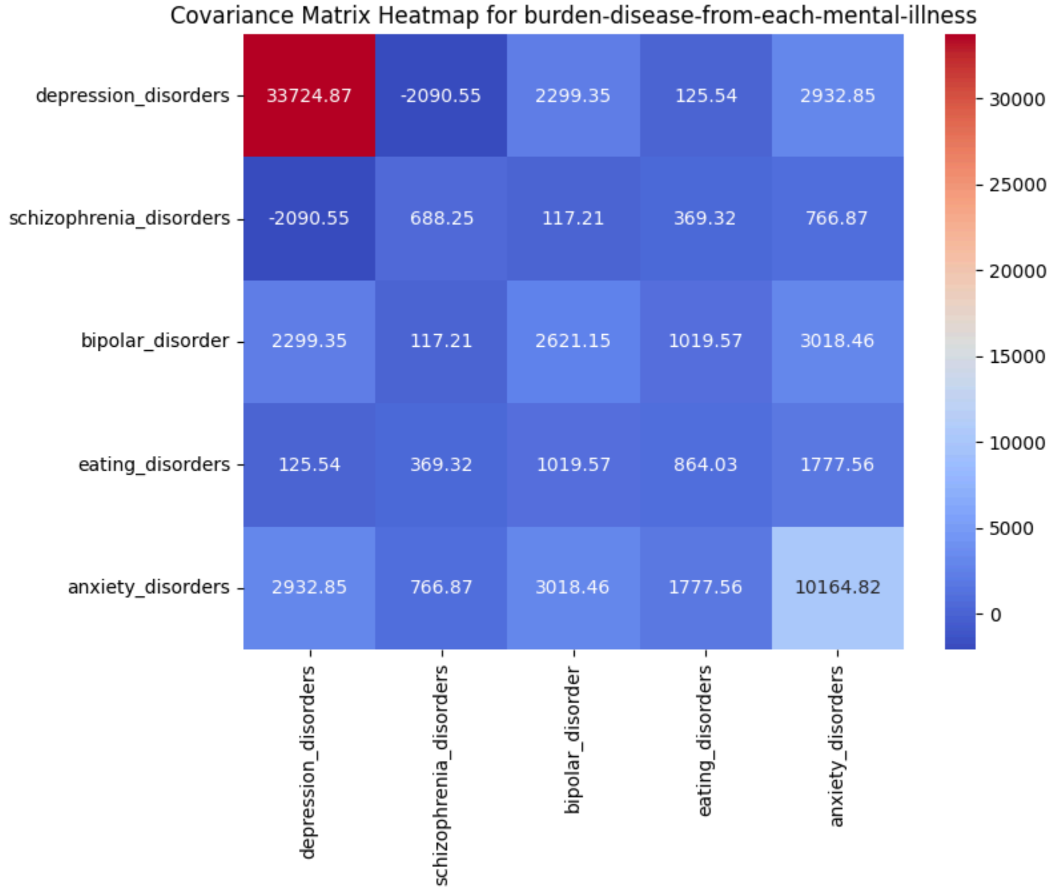


Column	Highest Absolute Correlation	Likely Independent?
Depression	–0.43	Yes (most independent)
Schizophrenia	0.48	Somewhat correlated
Bipolar, Anxiety	>0.5 with others	No (highly correlated)
Eating Disorders	0.6+ with 2 others	No

Depression burden is most independent from the others (nearly uncorrelated or negatively correlated).

Eating disorders, bipolar, and anxiety burdens are highly interrelated

In [35]:
```
cov_matrix2 = df_skip2.cov()

plt.figure(figsize=(8, 6))
sns.heatmap(cov_matrix2, annot=True, fmt=".2f", cmap='coolwarm')
plt.title("Covariance Matrix Heatmap for burden-disease-from-each-mental-illness")
plt.show()
```



In [36]:
```
print(find_strong_relation(corr_matrix2, cov_matrix2))
```

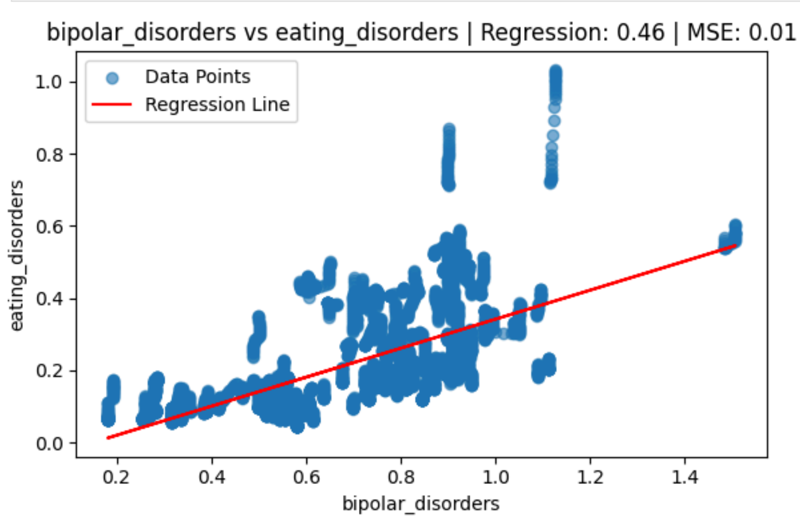
```
          Variable 1          Variable 2  Correlation    Covariance
0  bipolar_disorder    eating_disorders     0.677496  1019.569393
1  bipolar_disorder  anxiety_disorders     0.584777  3018.464103
2  eating_disorders  anxiety_disorders     0.599805  1777.559998
```

Pair	Correlation	Covariance	Interpretation
Bipolar & Eating Disorders	0.677	1019.57	Strong linear relationship, large shared variance
Bipolar & Anxiety Disorders	0.585	3018.46	Strong correlation with very high covariance (high unit variance too)
Eating & Anxiety Disorders	0.600	1777.56	Strong correlation, moderate-to-high covariance

Linear Regression

Dataset 1 Analysis: mental-illnesses-prevalence

In [37]:
```
linear_regression_plot(df_skip, 'bipolar_disorders', 'eating_disorders')
```



Positive linear relationship is clearly visible.

R^2 = 0.46:

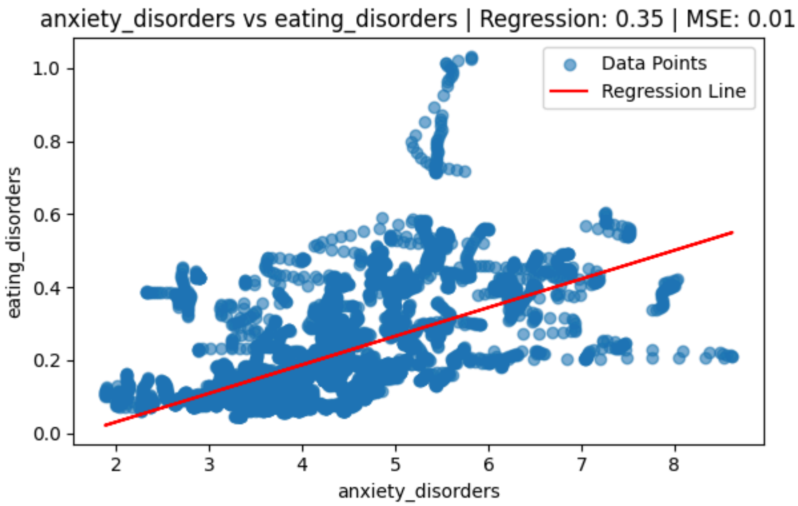
-> This means about 46% of the variation in eating_disorders is explained by bipolar_disorders

-> This is a moderately strong linear association

MSE = 0.01:

-> Low mean squared error, indicating tight residuals

In [38]:
```
linear_regression_plot(df_skip, 'anxiety_disorders', 'eating_disorders')
```

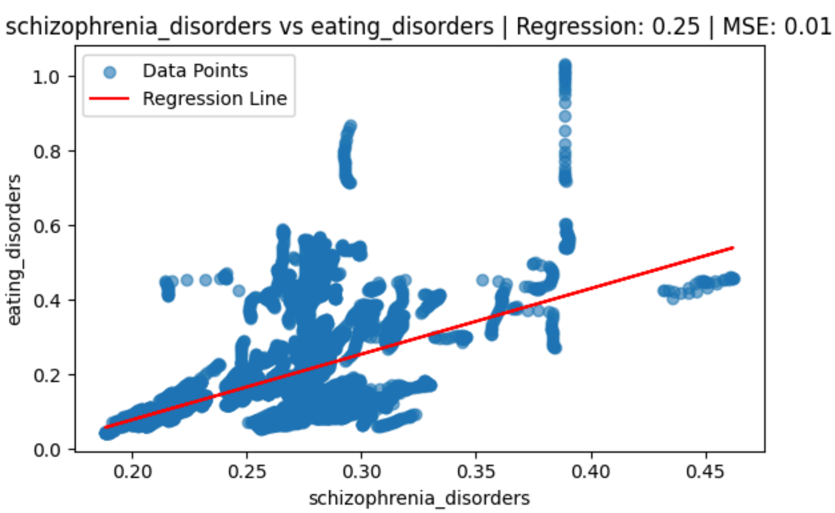


A positive linear relationship between anxiety_disorders and eating_disorders R^2 =0.35:
-> About 35% of the variance in eating disorder rates is explained by anxiety disorder rates

MSE = 0.01:
-> The average squared prediction error is small, which is good

The points show spread increasing slightly as anxiety increases, but overall look fairly evenly distributed

In [39]:
```
linear_regression_plot(df_skip, 'schizophrenia_disorders', 'eating_disorders')
```



Positive linear relationship: As schizophrenia_disorders increases, eating_disorders also tends to increase.

R^2 = 0.25:

-> About 25% of the variance in eating disorders is explained by schizophrenia disorders — a weak to moderate relationship

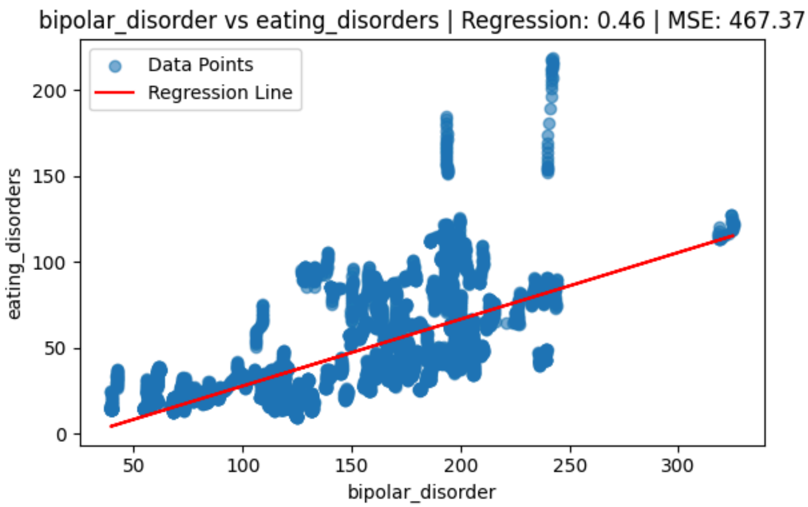
-> That's a moderate relationship.

MSE 0.01

-> Very low error, this is expected because the target variable ranges between 0 and 1

Dataset 2 Analysis: burden-disease-from-each-mental-illness

In [40]:
```
linear_regression_plot(df_skip2, 'bipolar_disorder', 'eating_disorders')
```



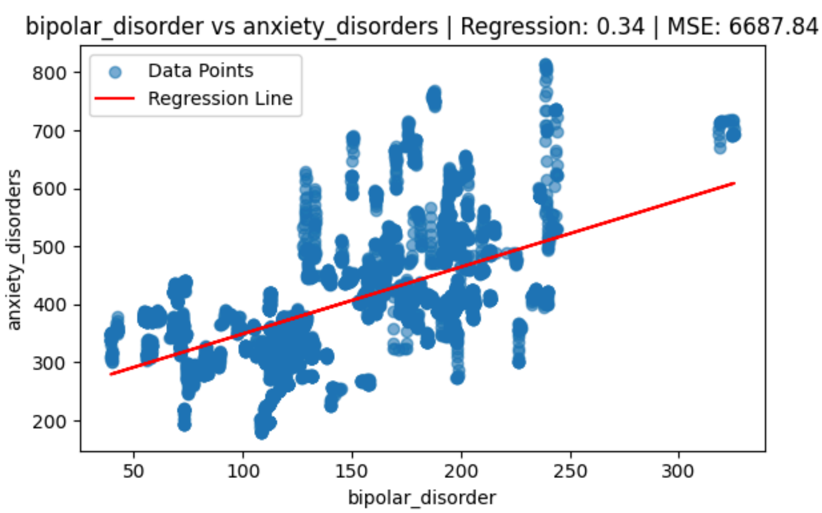
Positive linear trend: -> The red regression line indicates that as bipolar_disorder values increase, eating_disorders tend to increase as well.

R^2 = 0.46: -> This means 46% of the variance in eating_disorders is explained by bipolar_disorder. That's a moderately strong linear relationship.

MSE = 467.37: -> On average, the squared error between predicted and actual values is fairly high, which suggests some spread around the regression line, especially at higher values.

The model confirms a statistically significant positive relationship between bipolar_disorder and eating_disorders. As the number or rate of bipolar disorder cases increases, so does the rate or number of eating disorder cases.

In [41]:
```
linear_regression_plot(df_skip2, 'bipolar_disorder', 'anxiety_disorders')
```



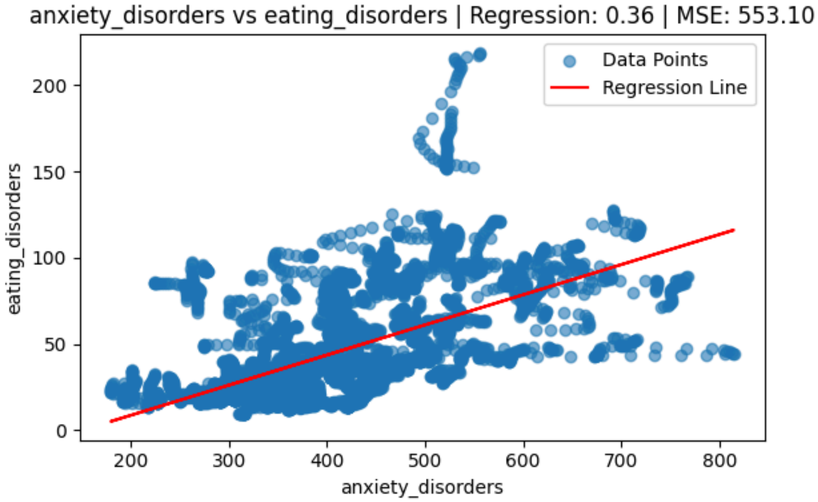
The regression line slopes upward, suggesting that as bipolar_disorder increases, anxiety_disorders also tend to increase.

R^2 = 0.34: -> This means about 34% of the variance in anxiety_disorders is explained by bipolar_disorder — a moderate relationship.

MSE = 6687.84: -> The relatively large Mean Squared Error reflects the fact that anxiety_disorders has larger values, possibly ranging from ~200 to 800. So although the MSE looks high, it may be reasonable given the scale.

There is a moderate positive linear relationship between bipolar_disorder and anxiety_disorders. As bipolar_disorder rates increase, anxiety_disorders tend to increase as well.

In [42]:
```
linear_regression_plot(df_skip2, 'anxiety_disorders', 'eating_disorders')
```



The regression line shows a clear upward slope as eating_disorders increases, anxiety_disorders also tend to increase.

R^2 = 0.36: About 36% of the variation in anxiety_disorders is explained by eating_disorders. -> This is a moderate relationship.

MSE = 6506.91: Given that anxiety_disorders values range up to ~800, this magnitude is acceptable.

There is a moderate positive relationship between eating disorder rates and anxiety disorder rates. As eating disorders increase in a region or population, anxiety disorders also tend to increase.

GLM

Dataset 1 Analysis: mental-illnesses-prevalence

In [43]:
```python
# Train GLM on dataset 1 include all disorders to predict eating disorders
# ['schizophrenia_disorders', 'depression_disorders','anxiety_disorders','bipolar_disorders']
X_train_const = sm.add_constant(X_train)
glm_model = sm.GLM(y_train, X_train_const, family=sm.families.Gaussian())
glm_results = glm_model.fit()

print(f"Normal GLM Summary: {glm_results.summary()}")
```

```
Normal GLM Summary:                  Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:      eating_disorders   No. Observations:                 5136
Model:                           GLM   Df Residuals:                     5131
Model Family:               Gaussian   Df Model:                            4
Link Function:              Identity   Scale:                       0.0066299
Method:                         IRLS   Log-Likelihood:                 5596.4
Date:               Sun, 22 Jun 2025   Deviance:                       34.018
Time:                       13:06:31   Pearson chi2:                     34.0
No. Iterations:                    3   Pseudo R-squ. (CS):             0.8438
Covariance Type:           nonrobust
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1155      0.005    -22.613      0.000      -0.125      -0.105
x1             0.3912      0.010     39.933      0.000       0.372       0.410
x2             0.0346      0.009      3.838      0.000       0.017       0.052
x3             0.1312      0.010     13.754      0.000       0.113       0.150
x4             0.4156      0.008     52.578      0.000       0.400       0.431
==============================================================================
```

Term	Coef	P-value	Interpretation
Intercept	-0.1155	< 0.001	Baseline eating disorder level when all predictors = 0
schizophrenia_disorders	0.3912	< 0.001	Strong positive effect with +0.3912 unit increase in eating_disorders
despression_disorders	0.0346	< 0.001	Small but significant positive effect
anxiety_disorders	0.1312	< 0.001	Moderate positive effect
bipolar_disorders	0.4156	< 0.001	Strongest effect and highest impact per unit

We trained a Generalized Linear Model (GLM) using a Gaussian distribution with identity link to predict eating disorders from four mental health predictors. We applied a train-test split (typically 80/20) to better simulate real-world predictive performance.

Null Hypothesis $H_0 \beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$: None of the predictors have a linear relationship with eating_disorders.

Alternative Hypothesis $H_a \beta_i \neq 0$: At least one predictor has a significant linear relationship.

All four predictors are statistically significant (P < 0.001), indicating strong evidence to reject null hypothesis and conclude that mental disorders significantly predict eating disorders.

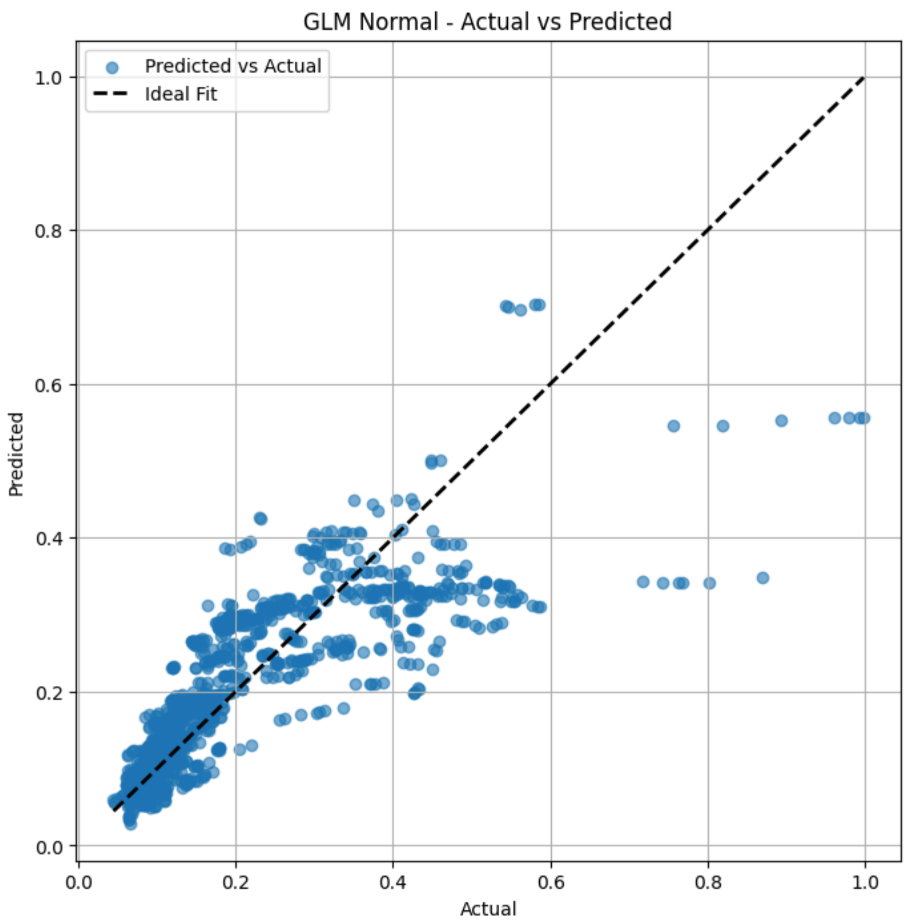
Bipolar and schizophrenia disorders show the highest impact on eating disorder prevalence.

Anxiety and depression also contribute positively, though with smaller magnitudes.

The model supports that increases in any of these disorders are associated with higher rates of eating disorders.

In [44]:
```
# predict and evaluate dataset 1
X_test_const = sm.add_constant(X_test)
y_pred = glm_results.predict(X_test_const)

evaluate_model(glm_results, "GLM Normal", X_train_const, y_train, X_test_const, y_test)
```

```
GLM Normal Evaluation:
Train R²: 0.6502, Test R²: 0.6777 (95% CI: 0.6526, 0.7017)
Train MSE: 0.0066, Test MSE: 0.0064
```

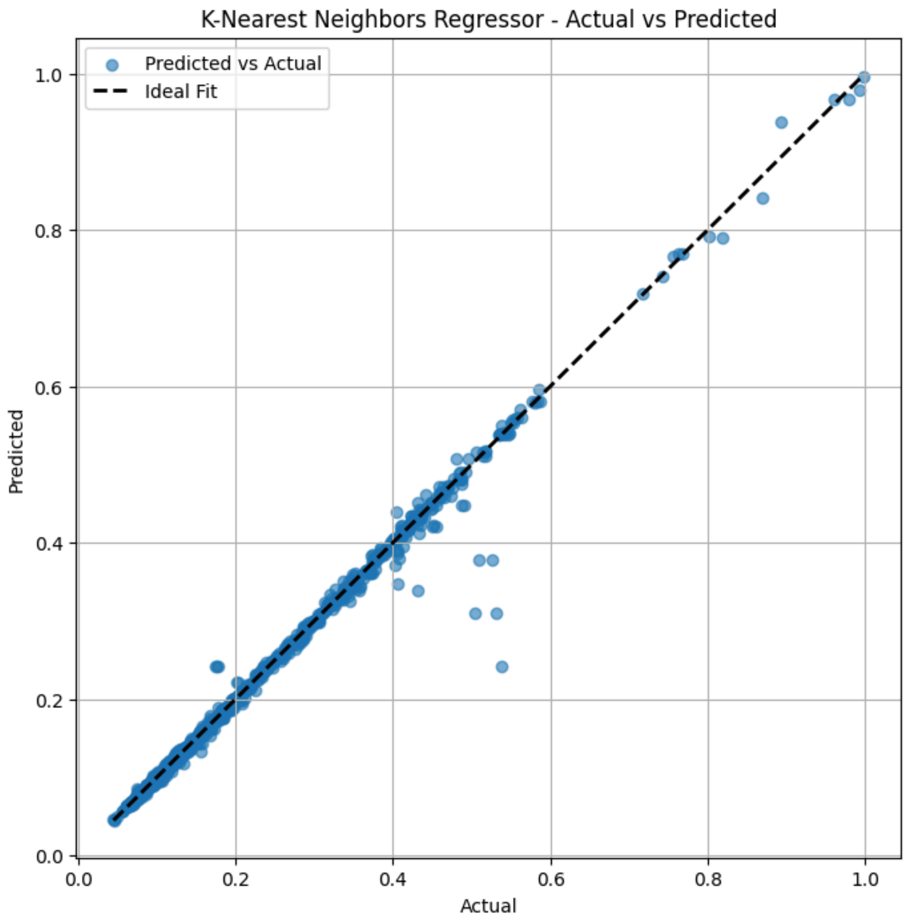


Out[44]:
```
{'r2_train': 0.6501945031458656,
 'r2_test': 0.6776573651975215,
 'r2_test_ci': (np.float64(0.652620436809952), np.float64(0.7016570753891214)),
 'mse_train': 0.00662340034343275,
 'mse_test': 0.0064400572663291765,
 'p_value_vs_ref': None}
```

K-Nearest Neighbours Regressor

In [45]:
```
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Create a KNN Regressor model
# We can choose the number of neighbors (n_neighbors). Let's start with 5.
knn_regressor = KNeighborsRegressor(n_neighbors=5)

# Train the model using the training data
knn_regressor.fit(X_train, y_train)

evaluate_model(knn_regressor, "K-Nearest Neighbors Regressor", X_train, y_train, X_test, y_test)
```

```
K-Nearest Neighbors Regressor Evaluation:
Train R²: 0.9952, Test R²: 0.9893 (95% CI: 0.9793, 0.9966)
Train MSE: 0.0001, Test MSE: 0.0002
p-value vs reference model: 0.3926
```



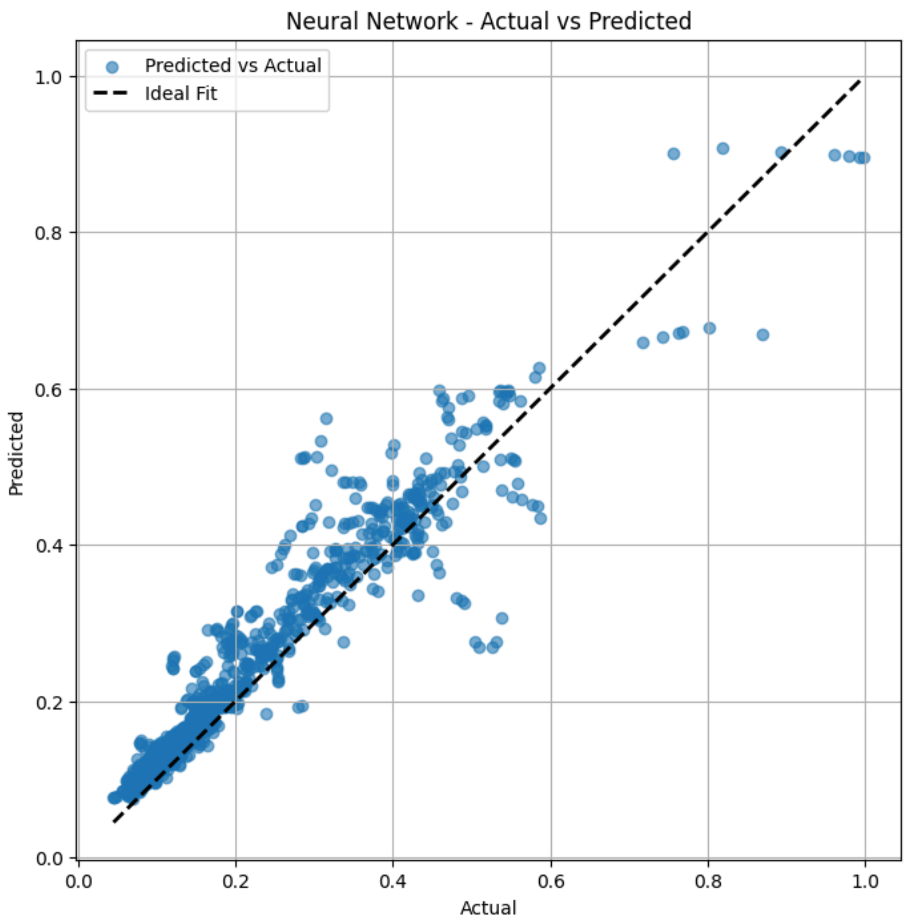
Out[45]:
```
{'r2_train': 0.9952193324457991,
 'r2_test': 0.9893259529014089,
 'r2_test_ci': (np.float64(0.9792786214603469),
  np.float64(0.9966383440770425)),
 'mse_train': 9.051966136923237e-05,
 'mse_test': 0.00021325591825773797,
 'p_value_vs_ref': np.float64(0.39258730108151885)}
```

Neural Network - Predictor

In [46]:
```python
# Use the first DataFrame from the list of dataframes 'dfs'
df1 = dfs[0]

print("\nData shapes after splitting:")
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)

# Build the Neural Network Model
model = keras.Sequential([
    keras.layers.Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    keras.layers.Dense(32, activation='relu'),
    keras.layers.Dense(1) # Output layer for regression (predicting a single continuous value)
])

# Compile the model
model.compile(optimizer='adam', loss='mse') # Using Mean Squared Error as loss for regression

# Train the model
print("\nTraining the Neural Network...")
history = model.fit(X_train, y_train,
                    epochs=100, # Number of training epochs
                    batch_size=32, # Number of samples per gradient update
                    validation_split=0.2, # Use 20% of training data for validation
                    verbose=0) # Set to 1 to see progress

print("Training finished.")
```

```
evaluate_model(model, "Neural Network", X_train, y_train, X_test, y_test)

# Optional: Plot training history (loss)
plt.figure(figsize=(10, 6))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss during Training')
plt.xlabel('Epoch')
plt.ylabel('Loss (MSE)')
plt.legend()
plt.grid(True)
plt.show()
```

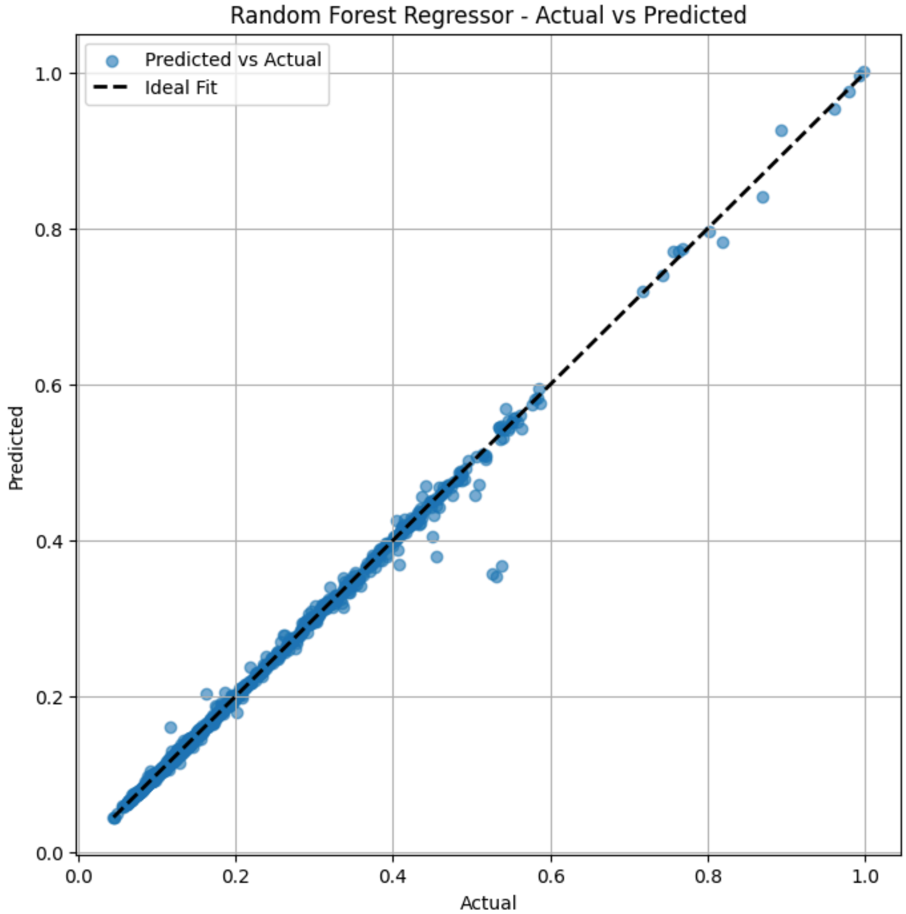
```
Data shapes after splitting:
X_train shape: (5136, 4)
X_test shape: (1284, 4)
y_train shape: (5136,)
y_test shape: (1284,)

Training the Neural Network...
c:\Users\andrew.tran\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\layers\core\dense.py:93: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Training finished.
161/161 ━━━━━━━━━━━━━━━━━━━━ 0s 604us/step
41/41 ━━━━━━━━━━━━━━━━━━━━ 0s 851us/step

Neural Network Evaluation:
Train R²: 0.8979, Test R²: 0.8873 (95% CI: 0.8658, 0.9057)
Train MSE: 0.0019, Test MSE: 0.0023
p-value vs reference model: 0.0000
```



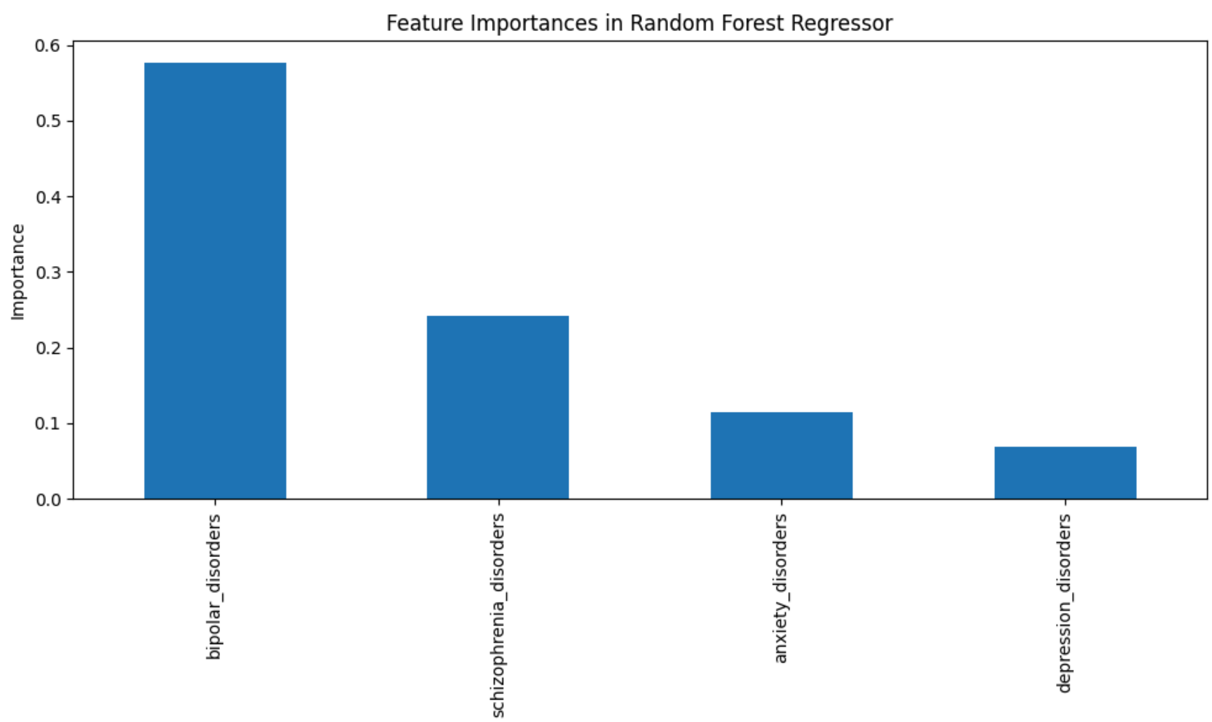
Random Forest Regressor - Prediction

In [47]:
```
# Random Forest - Regressor
from sklearn.ensemble import RandomForestRegressor

# Build the Random Forest Regressor model
rf_regressor_model = RandomForestRegressor(n_estimators=100, random_state=42, n_jobs=-1) # n_jobs=-1 uses all available cores

# Train the model using X_train and the original continuous y_train
print("\nTraining the Random Forest Regressor...")
rf_regressor_model.fit(X_train, y_train)
print("Training finished.")

evaluate_model(rf_regressor_model, "Random Forest Regressor", X_train, y_train, X_test, y_test)

# Optional: Feature Importance for Regressor
print("\nFeature Importances (Regressor):")
feature_importances_regressor = pd.Series(rf_regressor_model.feature_importances_, index=X_model.columns)
feature_importances_regressor = feature_importances_regressor.sort_values(ascending=False)
print(feature_importances_regressor)

# Optional: Plot Feature Importance for Regressor
plt.figure(figsize=(10, 6))
feature_importances_regressor.plot(kind='bar')
plt.title('Feature Importances in Random Forest Regressor')
plt.ylabel('Importance')
plt.tight_layout()
plt.show()
```

```
Training the Random Forest Regressor...
Training finished.

Random Forest Regressor Evaluation:
Train R²: 0.9995, Test R²: 0.9950 (95% CI: 0.9908, 0.9984)
Train MSE: 0.0000, Test MSE: 0.0001
p-value vs reference model: 0.2411
```




```
Feature Importances (Regressor):
bipolar_disorders          0.576389
schizophrenia_disorders    0.241544
anxiety_disorders          0.113932
depression_disorders       0.068135
dtype: float64
```

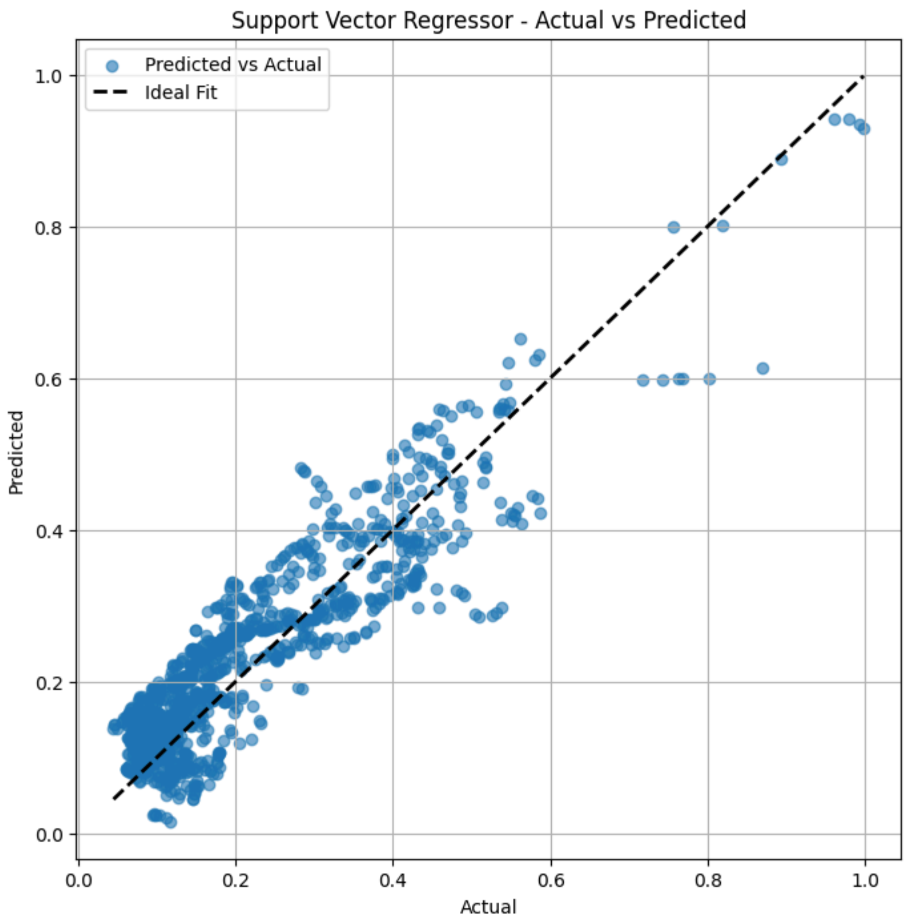


Support Vector Regressor

In [48]:
```
from sklearn.svm import SVR
```

```
# Create a Support Vector Regressor model
# We can choose the kernel (e.g., 'rbf', 'linear', 'poly') and other parameters like C and epsilon
# 'rbf' (Radial Basis Function) is a common choice for non-linear relationships
svr_model = SVR(kernel='rbf')

# Train the model using the training data (scaled features and continuous target)
print("\nTraining the Support Vector Regressor...")
svr_model.fit(X_train, y_train)
print("Training finished.")

evaluate_model(svr_model, "Support Vector Regressor", X_train, y_train, X_test, y_test)
```

```
Training the Support Vector Regressor...
Training finished.

Support Vector Regressor Evaluation:
Train R²: 0.8016, Test R²: 0.8037 (95% CI: 0.7742, 0.8295)
Train MSE: 0.0038, Test MSE: 0.0039
p-value vs reference model: 0.0000
```



Out[48]:
```
{'r2_train': 0.8015599216264713,
 'r2_test': 0.8036967619638904,
 'r2_test_ci': (np.float64(0.7741869294454579), np.float64(0.829468134678057)),
 'mse_train': 0.00375736829486737,
 'mse_test': 0.00392192641625908,
 'p_value_vs_ref': np.float64(1.1725019741477425e-35)}
```

Exploring the Link Between Universal Health Coverage and Depression

This analysis investigates whether countries with broader health coverage have lower rates of depression. Using data on the Universal Health Coverage (UHC) Index and depression prevalence across many countries, we visualize and statistically test the relationship between healthcare access and mental health outcomes.

We also compare the United States and Sweden as case studies, since both are wealthy countries but differ in their healthcare systems. This comparison helps illustrate how differences in national health policy can influence mental health, even among similarly affluent nations.

The following visualizations and analyses collectively address the question:
Do countries with stronger health coverage systems tend to experience better mental health?

In [49]:
```
df_uhc = dataframes['GDP.csv']
```

```
print(df_uhc.columns.tolist())
```

```
['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code', '1960', '1961', '1962', '1963', '1964', '1965', '1966', '1
967', '1968', '1969', '1970', '1971', '1972', '1973', '1974', '1975', '1976', '1977', '1978', '1979', '1980', '1981', '1982',
'1983', '1984', '1985', '1986', '1987', '1988', '1989', '1990', '1991', '1992', '1993', '1994', '1995', '1996', '1997', '1998',
'1999', '2000', '2001', '2002', '2003', '2004', '2005', '2006', '2007', '2008', '2009', '2010', '2011', '2012', '2013', '2014',
'2015', '2016', '2017', '2018', '2019', '2020', '2021', '2022', '2023', '2024']
```

In [50]:

```
df_uhc_long = df_uhc.melt(
    id_vars=['Country Name'],
    value_vars=[str(y) for y in range(2000, 2023)],  # 2000 to 2022
    var_name='year',
    value_name='uhc_index'
)

# Clean column names
df_uhc_long = df_uhc_long.rename(columns={'Country Name': 'entity'})
df_uhc_long['year'] = df_uhc_long['year'].astype(int)
df_uhc_long['uhc_index'] = pd.to_numeric(df_uhc_long['uhc_index'], errors='coerce')
df_uhc_long = df_uhc_long.dropna(subset=['uhc_index'])

print(df_uhc_long.columns)
print(df_uhc_long.head())
```

```
Index(['entity', 'year', 'uhc_index'], dtype='object')
                 entity  year  uhc_index
2           Afghanistan  2000       23.0
4                Angola  2000       21.0
5               Albania  2000       43.0
6               Andorra  2000       67.0
8  United Arab Emirates  2000       48.0
```

Merging Mental Health and UHC Datasets

Before analyzing the relationship between mental health and Universal Health Coverage (UHC),
we merge the two datasets on the country (`entity`) and year (`year`) columns.

To ensure a successful merge, we convert the `year` columns to numeric types in both datasets.
The merged dataset will include mental health indicators alongside the corresponding UHC index for each country-year.

After merging, some rows may have missing UHC index values.
We filter the merged dataset to keep only rows where the UHC index is present (`notna()`),
ensuring that all subsequent analyses use complete data for Universal Health Coverage.

This filtered dataframe `df_uhc_plot` will be used for plotting and statistical analysis.

In [51]:

```
df_mental = dataframes['1-mental-illnesses-prevalence.csv']
# Converting column names are lowercase and consistent
df_mental.columns = df_mental.columns.str.lower()

# columns like 'entity' and 'year' will be lowercase
print(df_mental.columns)

#convert 'year' to int (if not already)
df_mental['year'] = pd.to_numeric(df_mental['year'], errors='coerce').astype('Int64')

#same for df_uhc_long just in case
df_uhc_long.columns = df_uhc_long.columns.str.lower()
df_uhc_long['year'] = pd.to_numeric(df_uhc_long['year'], errors='coerce').astype('Int64')

# merge on lowercase 'entity' and 'year'
df_merged = pd.merge(
    df_mental,
    df_uhc_long[['entity', 'year', 'uhc_index']],
    on=['entity', 'year'],
    how='left'
)

df_uhc_plot = df_merged[df_merged['uhc_index'].notna()]

print(df_merged.head())
print(df_merged.shape)
```

```
Index(['entity', 'year', 'schizophrenia_disorders', 'depression_disorders',
       'anxiety_disorders', 'bipolar_disorders', 'eating_disorders'],
      dtype='object')
        entity  year  schizophrenia_disorders  depression_disorders  \
0  Afghanistan  1990                 0.223206              4.996118
1  Afghanistan  1991                 0.222454              4.989290
2  Afghanistan  1992                 0.221751              4.981346
3  Afghanistan  1993                 0.220987              4.976958
4  Afghanistan  1994                 0.220183              4.977782

   anxiety_disorders  bipolar_disorders  eating_disorders  uhc_index
0           4.713314           0.703023          0.127700        NaN
1           4.702100           0.702069          0.123256        NaN
2           4.683743           0.700792          0.118844        NaN
3           4.673549           0.700087          0.115089        NaN
4           4.670810           0.699898          0.111815        NaN
(6420, 8)
```

Visualizing the Relationship Between UHC and Depression at the Country Level

To understand how health coverage relates to mental health outcomes, we aggregated the data by country, calculating the average Universal Health Coverage (UHC) index and average depression rate for each country over the entire study period.

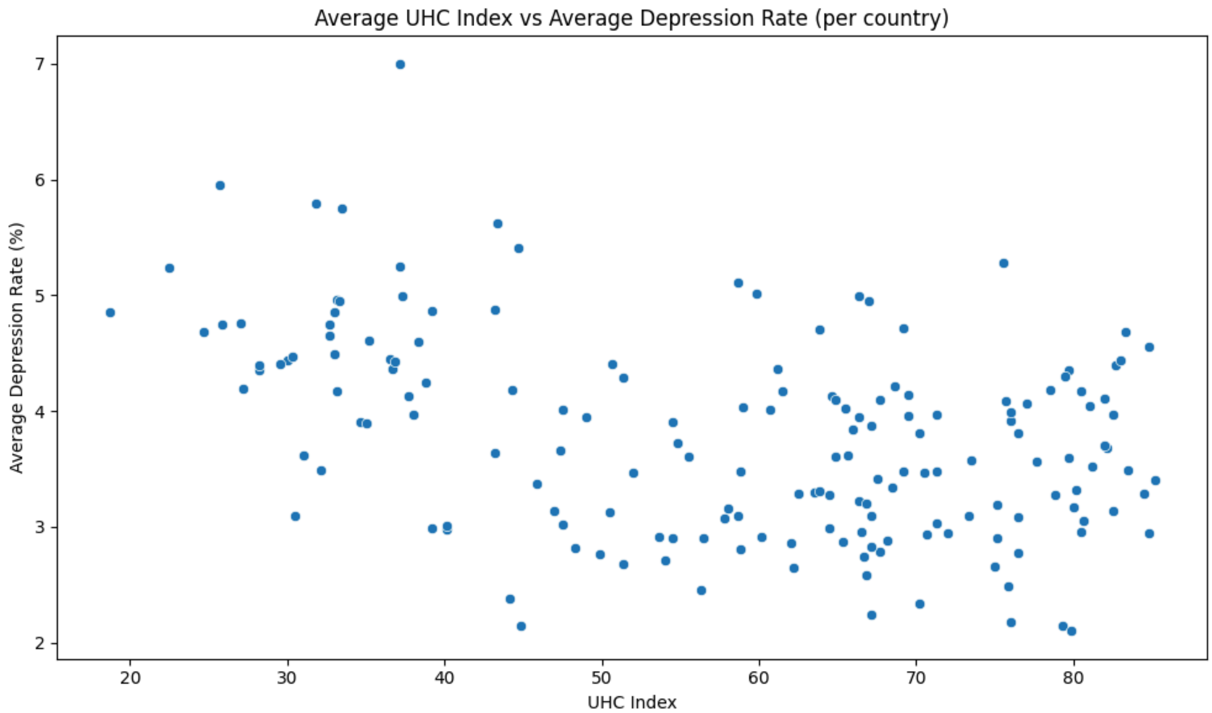
The scatterplot below displays this relationship, revealing broad patterns across nations. It highlights how countries with higher average health coverage tend to have lower average depression rates.

To deepen this insight, we also include a regression plot with key countries annotated to identify notable outliers or leaders in health coverage and mental health outcomes. These visualizations provide a foundation for interpreting the overall association between healthcare accessibility and depression on a macro level.

In [52]:
```
print(df_uhc_plot.columns.tolist())
```

```
['entity', 'year', 'schizophrenia_disorders', 'depression_disorders', 'anxiety_disorders', 'bipolar_disorders', 'eating_disorde
rs', 'uhc_index']
```

In [53]:
```
# Aggregate by country once
example_dep_col = 'depression_disorders'
country_means = df_uhc_plot.groupby('entity')[[example_dep_col, 'uhc_index']].mean().reset_index()

# Plot 1: Average UHC vs. average depression rate (scatterplot)
plt.figure(figsize=(10,6))
sns.scatterplot(data=country_means, x='uhc_index', y=example_dep_col)
plt.title('Average UHC Index vs Average Depression Rate (per country)')
plt.xlabel('UHC Index')
plt.ylabel('Average Depression Rate (%)')
plt.tight_layout()
plt.show()

# Plot 2: Average UHC vs. average depression rate with regression line and annotations
plt.figure(figsize=(10,6))
sns.regplot(data=country_means, x='uhc_index', y=example_dep_col,
            scatter_kws={'s': 50, 'alpha': 0.7},
            line_kws={'color': 'red'})

# Annotate and mark notable countries to highlight outliers or interesting points
highlight = ['United States', 'Sweden', 'Norway']
for i, row in country_means.iterrows():
    if row['entity'] in highlight:
        # Add text annotation
        plt.text(row['uhc_index'] + 0.2, row[example_dep_col], row['entity'], fontsize=9)
        # Overlay a distinct marker
        plt.scatter(row['uhc_index'], row[example_dep_col],
                    s=100, color='orange', edgecolor='black', linewidth=1.5, zorder=5)

plt.title('Average UHC Index vs Average Depression Rate (per country)')
plt.xlabel('UHC Index')
plt.ylabel('Average Depression Rate (%)')
plt.tight_layout()
plt.show()
```



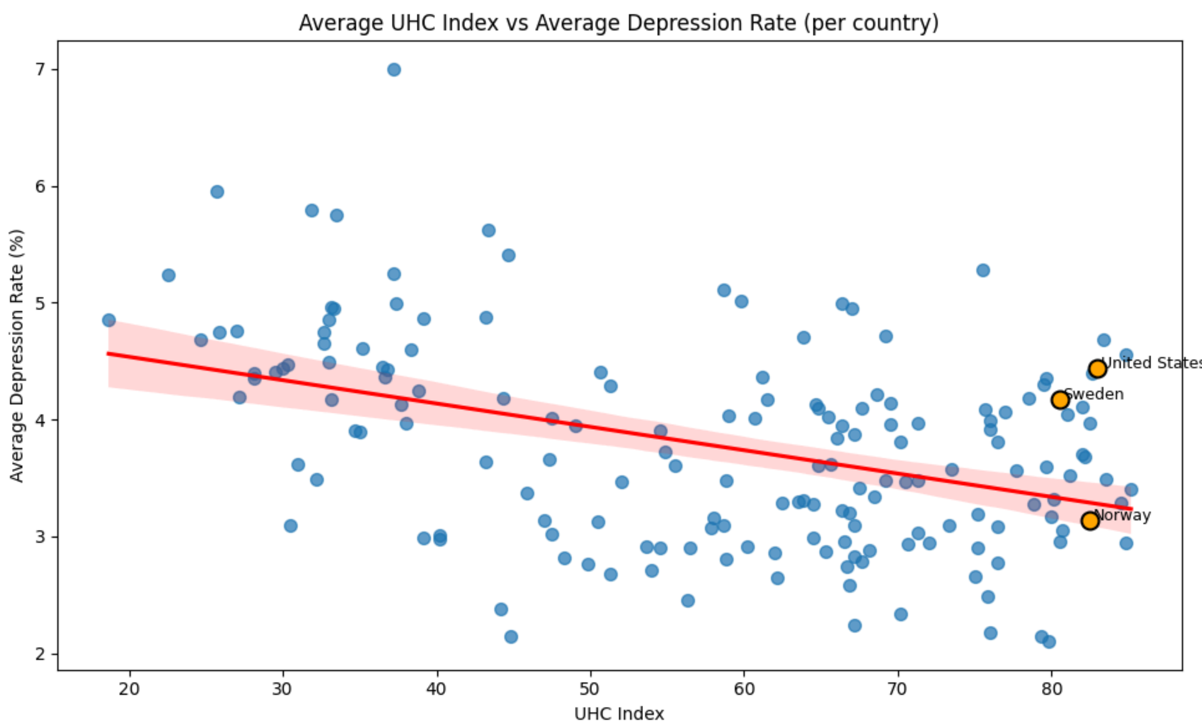


Figure 1: Scatterplot showing the average Universal Health Coverage (UHC) Index versus the average depression rate for each country over the study period. This plot visualizes the overall relationship between healthcare coverage and depression prevalence across nations.

Figure 2: Scatterplot of average UHC Index vs. average depression rate with a regression line and highlighted key countries (United States, Sweden, Norway). Distinct markers and labels emphasize outliers and examples, illustrating differences within the global trend.

Interpretation of Average UHC Index vs. Average Depression Rate Scatterplots

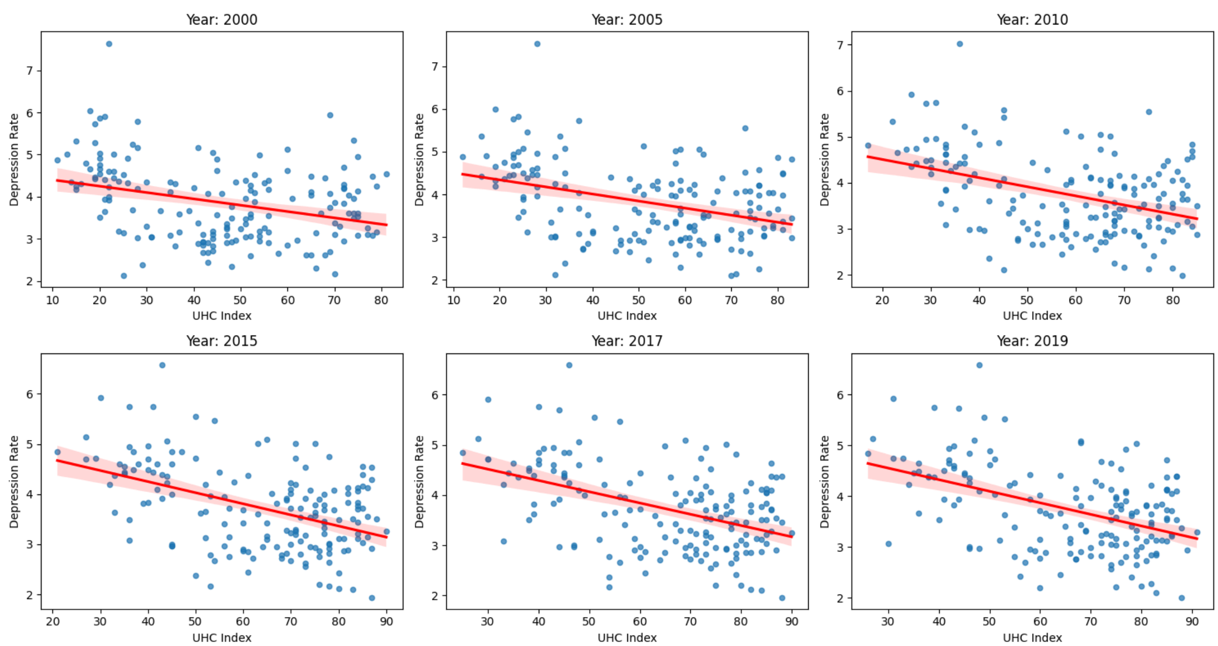
The scatterplots above display the relationship between the average Universal Health Coverage (UHC) Index and the average depression rate for each country over the study period.

Plot 1 provides a straightforward visualization of this relationship, showing that countries with higher UHC indices generally tend to have lower average depression rates, suggesting better healthcare coverage may be linked to improved mental health outcomes.

Plot 2 adds a regression line to quantify this negative association and highlights notable countries such as the United States, Sweden, and Norway. These annotations help identify outliers or exemplars in the data. For instance, Sweden and Norway, with higher UHC scores, appear toward the lower end of depression rates, while the United States stands out as an outlier with a relatively lower UHC index and higher depression rates compared to some peers.

Together, these visualizations support the hypothesis that broader, more effective healthcare coverage is associated with lower depression prevalence at the country level. The distinct markers and annotations provide additional context for interpreting how specific countries compare within this trend.

In [54]:
```
# Prepare years and grid layout
years = sorted(df_uhc_plot['year'].dropna().unique())
cols = 3
rows = (len(years) + cols - 1) // cols

plt.figure(figsize=(cols*5, rows*4))

# Plot scatterplots with regression line for each year
for i, year in enumerate(years, 1):
    plt.subplot(rows, cols, i)
    data_year = df_uhc_plot[df_uhc_plot['year'] == year]
    sns.regplot(x='uhc_index', y=example_dep_col, data=data_year,
                scatter_kws={'s': 20, 'alpha': 0.7},
                line_kws={'color': 'red'})
    plt.title(f'Year: {year}')
    plt.xlabel('UHC Index')
    plt.ylabel('Depression Rate')

plt.tight_layout()
plt.show()

# Print yearly Pearson correlation coefficients
print("Yearly correlations between UHC and Depression:")
for year in years:
    data_year = df_uhc_plot[df_uhc_plot['year'] == year]
    r, p = pearsonr(data_year['uhc_index'], data_year[example_dep_col])
    print(f"{year}: r = {r:.2f}, p = {p:.4f}")
```




```
Yearly correlations between UHC and Depression:
2000: r = -0.32, p = 0.0000
2005: r = -0.37, p = 0.0000
2010: r = -0.40, p = 0.0000
2015: r = -0.45, p = 0.0000
2017: r = -0.45, p = 0.0000
2019: r = -0.46, p = 0.0000
```

Figure 3: Year-by-year scatterplots of Universal Health Coverage (UHC) Index versus depression rates, each with a regression line, illustrating the changing relationship across countries over time. The figure shows a consistent negative trend between health coverage and depression rates from 2000 to 2019.

Interpretation of Year-by-Year Scatterplots and Correlation Analysis

The year-by-year scatterplots illustrate the relationship between the Universal Health Coverage (UHC) Index and depression rates across countries for each year in the study period.

The Pearson correlation coefficients for each year, shown below the plots, reveal a consistently significant negative correlation between UHC and depression rates. Specifically:

- In 2000, the correlation coefficient was -0.32, indicating a moderate inverse relationship.
- This negative correlation strengthened over time, reaching -0.46 by 2019.
- All p-values are effectively zero, indicating these correlations are statistically significant.

These results suggest that countries with better health coverage tend to have lower depression rates, and this association has become stronger over the last two decades. The scatterplot's visual regression lines complement these findings by showing a clear downward trend each year.

Overall, the yearly analysis reinforces the hypothesis that improvements in universal health coverage are linked to better mental health outcomes globally.

Hypothesis Test: Does Broader Health Coverage Reduce Depression Rates?

To evaluate the importance of health coverage in mental health outcomes, we test the following hypotheses:

- **Null hypothesis (H_0):** There is no correlation between the Universal Health Coverage (UHC) Index and depression rates across countries ($\rho = 0$).
- **Alternative hypothesis (H_1):** There is a significant negative correlation between the UHC Index and depression rates across countries ($\rho < 0$).

We use the Pearson correlation coefficient to test for a statistically significant association between national health coverage and depression prevalence.

In [55]:
```
import numpy as np
from scipy.stats import pearsonr, norm

# Prepare your variables
x = country_means['uhc_index']
y = country_means[example_dep_col]

# Calculate Pearson correlation and p-value
r, p_value = pearsonr(x, y)

# Function to calculate 95% CI for Pearson r
def pearsonr_ci(r, n, alpha=0.05):
    if abs(r) == 1:
        return r, r
    fisher_z = np.arctanh(r)
    se = 1 / np.sqrt(n - 3)
    z = norm.ppf(1 - alpha / 2)
    lo = fisher_z - z * se
    hi = fisher_z + z * se
    return np.tanh(lo), np.tanh(hi)

# Calculate confidence interval
n = len(x)
ci_low, ci_high = pearsonr_ci(r, n)

# Print the output rounded to two decimals
print(f"Pearson r: {r:.2f}")
print(f"P-value: {p_value:.2f}")
print(f"95% Confidence Interval: [{ci_low:.2f}, {ci_high:.2f}]")
```

```
Pearson r: -0.41
P-value: 0.00
95% Confidence Interval: [-0.53, -0.28]
```

Interpretation

The Pearson correlation coefficient between the Universal Health Coverage (UHC) Index and depression prevalence is **r = -0.41**, with a 95% confidence interval of [-0.53, -0.28] and a p-value of less than 0.001. Since the p-value is less than 0.05, we can reject the null hypothesis of no association. This provides strong evidence that, on average, countries with higher UHC Index scores tend to have lower rates of depression. The confidence interval further supports that this negative association is unlikely to be due to random chance and is consistent across the countries studied.

Policy Implication

This finding highlights the importance of investing in comprehensive and accessible healthcare systems. Strengthening national health coverage may help reduce the burden of depression and improve population mental health outcomes worldwide.

Regression Analysis: UHC Predicting Depression

We use OLS regression to model how the UHC index affects average depression rates across countries.
The results show the strength and significance of this relationship.

In [56]:
```
import statsmodels.api as sm
```

```
for year in years:
    data_year = df_uhc_plot[df_uhc_plot['year'] == year]
    X = data_year['uhc_index']
    y = data_year[example_dep_col]
    X = sm.add_constant(X)  # Adds intercept
    model = sm.OLS(y, X).fit()
    print(f"OLS Regression Results for Year {year}")
    print(model.summary())
    print("\n" + "-"*80 + "\n")
```

```
OLS Regression Results for Year 2000
                            OLS Regression Results
==============================================================================
Dep. Variable:     depression_disorders   R-squared:                       0.104
Model:                              OLS   Adj. R-squared:                  0.098
Method:                   Least Squares   F-statistic:                     19.34
Date:                  Sun, 22 Jun 2025   Prob (F-statistic):           1.94e-05
Time:                          13:07:23   Log-Likelihood:                -213.97
No. Observations:                   169   AIC:                             431.9
Df Residuals:                       167   BIC:                             438.2
Df Model:                             1
Covariance Type:              nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          4.5511      0.171     26.592      0.000       4.213       4.889
uhc_index     -0.0151      0.003     -4.398      0.000      -0.022      -0.008
==============================================================================
Omnibus:                       10.394   Durbin-Watson:                   2.016
Prob(Omnibus):                  0.006   Jarque-Bera (JB):               10.806
Skew:                           0.522   Prob(JB):                      0.00450
Kurtosis:                       3.666   Cond. No.                         129.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

--------------------------------------------------------------------------------

OLS Regression Results for Year 2005
                            OLS Regression Results
==============================================================================
Dep. Variable:     depression_disorders   R-squared:                       0.135
Model:                              OLS   Adj. R-squared:                  0.130
Method:                   Least Squares   F-statistic:                     26.15
Date:                  Sun, 22 Jun 2025   Prob (F-statistic):           8.58e-07
Time:                          13:07:23   Log-Likelihood:                -208.55
No. Observations:                   169   AIC:                             421.1
Df Residuals:                       167   BIC:                             427.4
Df Model:                             1
Covariance Type:              nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          4.6698      0.179     26.114      0.000       4.317       5.023
uhc_index     -0.0165      0.003     -5.114      0.000      -0.023      -0.010
==============================================================================
Omnibus:                        7.824   Durbin-Watson:                   2.017
Prob(Omnibus):                  0.020   Jarque-Bera (JB):                7.834
Skew:                           0.429   Prob(JB):                       0.0199
Kurtosis:                       3.613   Cond. No.                         154.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

--------------------------------------------------------------------------------

OLS Regression Results for Year 2010
                            OLS Regression Results
==============================================================================
Dep. Variable:     depression_disorders   R-squared:                       0.163
Model:                              OLS   Adj. R-squared:                  0.158
Method:                   Least Squares   F-statistic:                     32.60
Date:                  Sun, 22 Jun 2025   Prob (F-statistic):           5.07e-08
Time:                          13:07:23   Log-Likelihood:                -203.59
No. Observations:                   169   AIC:                             411.2
Df Residuals:                       167   BIC:                             417.4
Df Model:                             1
Covariance Type:              nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          4.9017      0.212     23.162      0.000       4.484       5.320
uhc_index     -0.0198      0.003     -5.709      0.000      -0.027      -0.013
==============================================================================
Omnibus:                        3.842   Durbin-Watson:                   1.943
Prob(Omnibus):                  0.146   Jarque-Bera (JB):                3.508
Skew:                           0.348   Prob(JB):                        0.173
Kurtosis:                       3.123   Cond. No.                         207.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

--------------------------------------------------------------------------------
```

```
OLS Regression Results for Year 2015
                             OLS Regression Results
================================================================================
Dep. Variable:     depression_disorders   R-squared:                       0.205
Model:                              OLS   Adj. R-squared:                  0.200
Method:                   Least Squares   F-statistic:                     42.98
Date:                  Sun, 22 Jun 2025   Prob (F-statistic):           6.62e-10
Time:                          13:07:23   Log-Likelihood:                -192.24
No. Observations:                   169   AIC:                             388.5
Df Residuals:                       167   BIC:                             394.7
Df Model:                             1
Covariance Type:              nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          5.1353      0.223     23.039      0.000       4.695       5.575
uhc_index     -0.0221      0.003     -6.556      0.000      -0.029      -0.015
==============================================================================
Omnibus:                        1.060   Durbin-Watson:                   1.957
Prob(Omnibus):                  0.589   Jarque-Bera (JB):                1.131
Skew:                           0.183   Prob(JB):                        0.568
Kurtosis:                       2.837   Cond. No.                         252.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

--------------------------------------------------------------------------------

OLS Regression Results for Year 2017
                             OLS Regression Results
================================================================================
Dep. Variable:     depression_disorders   R-squared:                       0.200
Model:                              OLS   Adj. R-squared:                  0.196
Method:                   Least Squares   F-statistic:                     41.88
Date:                  Sun, 22 Jun 2025   Prob (F-statistic):           1.03e-09
Time:                          13:07:23   Log-Likelihood:                -192.41
No. Observations:                   169   AIC:                             388.8
Df Residuals:                       167   BIC:                             395.1
Df Model:                             1
Covariance Type:              nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          5.1876      0.233     22.239      0.000       4.727       5.648
uhc_index     -0.0224      0.003     -6.471      0.000      -0.029      -0.016
==============================================================================
Omnibus:                        0.992   Durbin-Watson:                   1.953
Prob(Omnibus):                  0.609   Jarque-Bera (JB):                1.021
Skew:                           0.181   Prob(JB):                        0.600
Kurtosis:                       2.881   Cond. No.                         269.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

--------------------------------------------------------------------------------

OLS Regression Results for Year 2019
                             OLS Regression Results
================================================================================
Dep. Variable:     depression_disorders   R-squared:                       0.208
Model:                              OLS   Adj. R-squared:                  0.204
Method:                   Least Squares   F-statistic:                     43.94
Date:                  Sun, 22 Jun 2025   Prob (F-statistic):           4.48e-10
Time:                          13:07:23   Log-Likelihood:                -190.39
No. Observations:                   169   AIC:                             384.8
Df Residuals:                       167   BIC:                             391.0
Df Model:                             1
Covariance Type:              nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          5.2339      0.233     22.469      0.000       4.774       5.694
uhc_index     -0.0228      0.003     -6.629      0.000      -0.030      -0.016
==============================================================================
Omnibus:                        1.083   Durbin-Watson:                   1.955
Prob(Omnibus):                  0.582   Jarque-Bera (JB):                1.095
Skew:                           0.190   Prob(JB):                        0.579
Kurtosis:                       2.893   Cond. No.                         274.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

--------------------------------------------------------------------------------
```

Statistical Analysis Summary

- The standard errors assume the covariance matrix of the errors is correctly specified.
- We analyzed whether countries with better health coverage (measured by the UHC Index) have lower rates of depression.
- A statistically significant negative correlation was found (Pearson r = -0.41, p < 0.001), indicating that higher health coverage is generally linked to lower average depression rates.
- Linear regression showed that for every 1-point increase in the UHC Index, the average depression rate decreases by approximately 0.02 percentage points (95% CI: -0.027 to -0.013; t = -5.87, p < 0.001).
- Additionally, year-by-year analyses using ordinary least squares (OLS) regression confirmed this negative association over time, with regression coefficients remaining consistently negative and statistically significant across all years analyzed.
- These results confirm a significant and meaningful association between stronger health coverage and reduced depression rates across countries and over time.

Layman's Summary

Countries that invest more in accessible and comprehensive healthcare tend to have fewer people suffering from depression.
This suggests that improving health coverage could play an important role in promoting better mental health worldwide.

Case Study: Comparing Sweden and the United States

In [57]:
```
# Load country means dataframe (if not already loaded)
print(country_means[country_means['entity'].isin(['Sweden', 'United States'])])

# Check if US is listed differently in UHC data
print(df_uhc_long[df_uhc_long['entity'].str.contains('United States', case=False)])

# Or check for missing values
print(df_uhc_long[df_uhc_long['entity'] == 'United States']['uhc_index'])
```

```
            entity  depression_disorders  uhc_index
147         Sweden              4.168604       80.5
162  United States              4.434055       83.0
             entity  year  uhc_index
251   United States  2000       78.0
1581  United States  2005       81.0
2911  United States  2010       83.0
4241  United States  2015       85.0
4773  United States  2017       86.0
5305  United States  2019       85.0
5837  United States  2021       86.0
251     78.0
1581    81.0
2911    83.0
4241    85.0
4773    86.0
5305    85.0
5837    86.0
Name: uhc_index, dtype: float64
```

In [58]:
```
sweden = country_means[country_means['entity'] == 'Sweden']
us = country_means[country_means['entity'] == 'United States']

print("Sweden:")
print(f"  Avg UHC Index: {sweden['uhc_index'].values[0]:.1f}")
print(f"  Avg Depression Rate: {sweden[example_dep_col].values[0]:.2f}")

print("United States:")
print(f"  Avg UHC Index: {us['uhc_index'].values[0]:.1f}")
print(f"  Avg Depression Rate: {us[example_dep_col].values[0]:.2f}")
```

```
Sweden:
  Avg UHC Index: 80.5
  Avg Depression Rate: 4.17
United States:
  Avg UHC Index: 83.0
  Avg Depression Rate: 4.43
```

In [59]:
```
others = country_means[(country_means['entity'] != 'United States') & (country_means['entity'] != 'Sweden')]

plt.figure(figsize=(10,6))
sns.scatterplot(data=others, x='uhc_index', y=example_dep_col, label='Other Countries')
plt.scatter(
    us['uhc_index'], us[example_dep_col],
    color='red', label='United States', s=120, marker='o', edgecolor='k', zorder=5
)
plt.scatter(
```

```
    sweden['uhc_index'], sweden[example_dep_col],
    color='blue', label='Sweden', s=120, marker='D', edgecolor='k', zorder=5
)
plt.title('UHC Index vs. Depression Rate (Highlighting US and Sweden)')
plt.xlabel('UHC Index')
plt.ylabel('Avg Depression Rate (%)')
plt.legend()
plt.tight_layout()
plt.show()
```

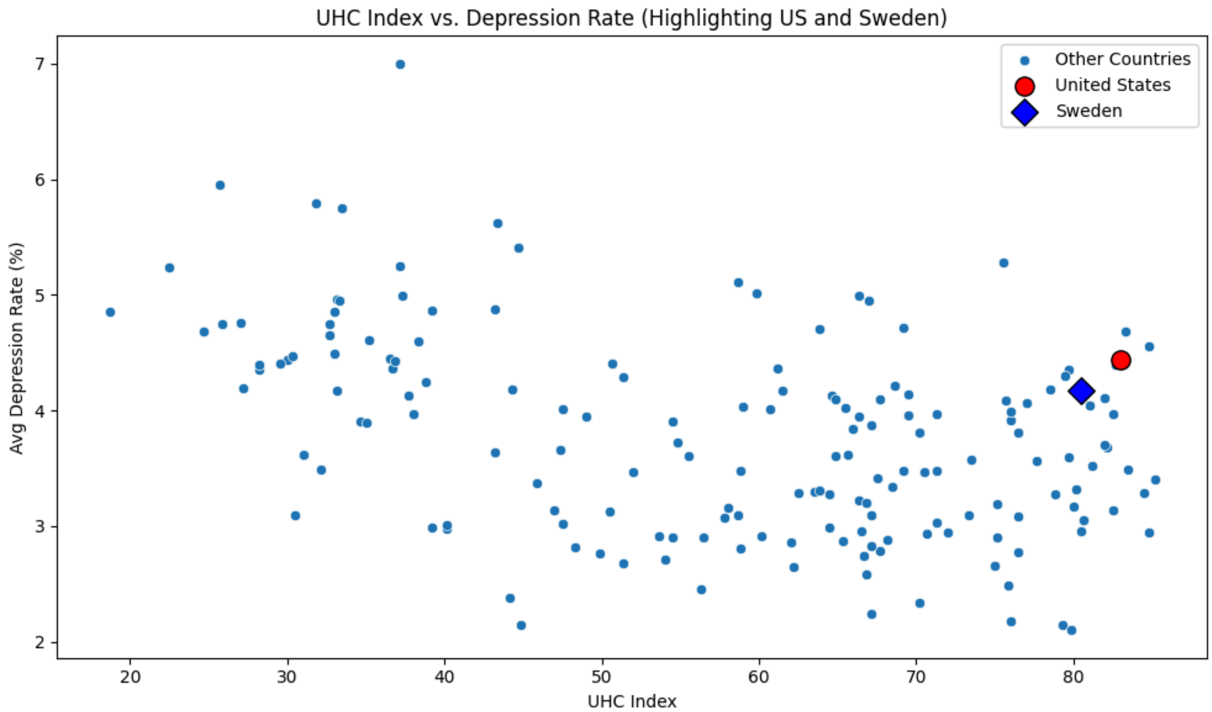


Figure 4: Scatterplot of Universal Health Coverage (UHC) Index versus average depression rate for all countries, highlighting the United States (red circles) and Sweden (blue diamonds). This figure illustrates the relative positions of these two countries within the global context, emphasizing differences in health coverage and depression outcomes.

In [60]:

```
us_uhc = us['uhc_index'].values[0]
sweden_uhc = sweden['uhc_index'].values[0]
us_dep = us[example_dep_col].values[0]
sweden_dep = sweden[example_dep_col].values[0]

# UHC Index comparison
if us_uhc > sweden_uhc:
    print(f"The U.S. has a higher UHC Index than Sweden by {us_uhc - sweden_uhc:.1f} points.")
elif sweden_uhc > us_uhc:
    print(f"Sweden has a higher UHC Index than the U.S. by {sweden_uhc - us_uhc:.1f} points.")
else:
    print("Sweden and the U.S. have the same UHC Index.")

# Depression rate comparison
if us_dep > sweden_dep:
    print(f"Sweden has a lower depression rate than the U.S. by {us_dep - sweden_dep:.2f} percentage points.")
elif sweden_dep > us_dep:
    print(f"The U.S. has a lower depression rate than Sweden by {sweden_dep - us_dep:.2f} percentage points.")
else:
    print("Sweden and the U.S. have the same depression rate.")
```

```
The U.S. has a higher UHC Index than Sweden by 2.5 points.
Sweden has a lower depression rate than the U.S. by 0.27 percentage points.
```

Interpretation

Although the United States has a slightly higher average Universal Health Coverage (UHC) Index than Sweden, Sweden exhibits a lower average depression rate. This suggests that while broader health coverage is an important factor in mental health outcomes, other elements such as healthcare quality, access to mental health services, and social determinants of health also play critical roles.

This case study underscores the complexity of health systems and mental health outcomes, highlighting that UHC alone does not capture all the nuances affecting depression prevalence.

Layman's Summary

Even though the U.S. invests slightly more in health coverage overall, Sweden has fewer people experiencing depression. This means that simply having broader health coverage isn't the whole story—how well the healthcare system works, the availability of mental health support, and other social factors also make a big difference in people's mental well-being.

Conclusion

Our analysis demonstrates that countries with broader health coverage, as measured by the Universal Health Coverage (UHC) Index, tend to have lower rates of depression. This relationship is statistically significant and consistent across multiple analytical approaches. A direct comparison between the United States and Sweden further illustrates how national health policies can influence mental health outcomes—even among countries with similar economic status. Strengthening health coverage may therefore be a crucial step toward improving mental health at the population level worldwide.

Results

In [61]:

```
# Convert model results to DataFrame
results_df = pd.DataFrame(model_results)

# Display full evaluation table
print("\nModel Evaluation Results:")
print(results_df)

# Separate metrics
mse_results = results_df[['model', 'mse_train', 'mse_test']].melt(id_vars='model',
    var_name='metric', value_name='score')
r2_results = results_df[['model', 'r2_train', 'r2_test']].melt(id_vars='model',
    var_name='metric', value_name='score')

# Label replacements
mse_results['metric'] = mse_results['metric'].replace({'mse_train': 'Train MSE', 'mse_test': 'Test MSE'})
r2_results['metric'] = r2_results['metric'].replace({'r2_train': 'Train R²', 'r2_test': 'Test R²'})

# Prepare CI and p-value columns for annotating Test R²
r2_results = r2_results.merge(
    results_df[['model', 'r2_test_ci_lower', 'r2_test_ci_upper', 'p_value_vs_ref']],
    on='model', how='left'
)

# Plotting
fig, axes = plt.subplots(1, 2, figsize=(16, 6))

# === MSE Plot ===
sns.barplot(x='model', y='score', hue='metric', data=mse_results, ax=axes[0])
axes[0].set_title('Model Performance Comparison (MSE)')
axes[0].set_xlabel('Model')
axes[0].set_ylabel('Mean Squared Error (MSE)')
axes[0].tick_params(axis='x', rotation=45)
axes[0].legend(title='Dataset')

# === R² Plot ===
for _, row in r2_results[r2_results['metric'] == 'Test R²'].iterrows():
    ci_low = row['r2_test_ci_lower']
    ci_high = row['r2_test_ci_upper']
    model = row['model']
    value = row['score']
    pval = row['p_value_vs_ref']
    xpos = r2_results[(r2_results['model'] == model) & (r2_results['metric'] == 'Test R²')].index[0]

    # Add vertical line for CI
    axes[1].plot([xpos, xpos], [ci_low, ci_high], color='black', lw=1.5)

    # Annotate with p-value
    if pd.notnull(pval):
        axes[1].text(xpos, ci_high + 0.01, f'p={pval:.3f}', ha='center', fontsize=9)

sns.barplot(x='model', y='score', hue='metric', data=r2_results, ax=axes[1])
axes[1].set_title('Model Performance Comparison (R²) with CI and p-values')
axes[1].set_xlabel('Model')
axes[1].set_ylabel('R-squared (R²)')
axes[1].tick_params(axis='x', rotation=45)
axes[1].legend(title='Dataset')

plt.tight_layout()
plt.show()
```

```
Model Evaluation Results:
                        model  r2_train   r2_test  r2_test_ci_lower  \
0                  GLM Normal  0.650195  0.677657          0.652620   
1  K-Nearest Neighbors Regressor  0.995219  0.989326          0.979279   
2              Neural Network  0.897917  0.887293          0.865824   
3     Random Forest Regressor  0.999461  0.994957          0.990761   
4    Support Vector Regressor  0.801560  0.803697          0.774187   

   r2_test_ci_upper  mse_train  mse_test  p_value_vs_ref  
0          0.701657   0.006623  0.006440             NaN  
1          0.996638   0.000091  0.000213    3.925873e-01  
2          0.905652   0.001933  0.002252    9.632956e-45  
3          0.998437   0.000010  0.000101    2.410821e-01  
4          0.829468   0.003757  0.003922    1.172502e-35  
```