**Western Governors University (WGU)**

**D208 Task 2: Logistic Regression Modeling**

**Natallia Zimnitskaya | ID: 012247127**

**Master of Science, Data Analytics**

***Part I: Research Question***

***A1. Question***

What factors contribute to a patient’s readmission?

***A2. Objectives and Goals for Analysis***

This analysis aims to identify key factors contributing to patient readmission within a month and develop a logistic regression model to predict readmission likelihood based on patient characteristics and initial treatment details. This analysis aims to enhance patient care by tailoring interventions for high-risk patients, reducing healthcare costs by lowering readmission rates, and informing policy decisions with data-driven insights. Healthcare providers can improve patient management and resource allocation, patients can benefit from better health outcomes and enhanced support, insurance companies can achieve cost savings and better risk management, and policymakers can create policies that incentivize hospitals to reduce readmissions. The logistic regression model will use ReAdmis as the dependent variable, with features such as age, income, gender, and medical conditions to identify significant predictors of readmission and provide actionable insights.

***Part II: Method Justification***

***B1. Assumptions***

Logistic regression is a statistical method to model the relationship between a binary dependent variable and one or more independent variables. Unlike linear regression, which predicts continuous outcomes, logistic regression predicts the probability of a binary outcome.

Before fitting a model to a dataset, logistic regression makes the following assumptions (Bobbitt, 2020):

* **The Response Variable is Binary.** The outcome variable should have two categories, such as “yes” or “no,” “pass” or “fail”. For instance, predicting whether a student will pass or fail an exam based on the number of hours they study.
* **Independence of Observations.** The observation should be independent. The outcome of one observation should not affect the outcome of another. For instance, analyzing whether customers will buy a product based on their age and income, each customer’s decision should be independent.
* **No Multicollinearity.** The independent variables should not be highly correlated to avoid distorting the model’s estimates. For example, predicting whether a person will get a loan based on their income and savings, where income and savings should not be too closely related.
* **Lack of Strongly Influential Outliers.** The dataset should not contain extreme outliers that can disproportionately impact the model’s predictions—for instance, predicting whether a patient will be readmitted to a hospital-based on age and the number of doctor visits. In contrast, an extremely old patient with an unusually high number of doctor visits could skew the results.

***B2. Programming Language and Benefits***

After reading "R vs. Python: 12 Key Comparisons," Python was chosen to complete this performance assessment. High-level, general-purpose programming language Python is flexible, object-oriented, open-source, and emphasizes code readability with a clear visual structure and straightforward syntax. Since Python is open source, many people can contribute to its development and improve its libraries and features. Python has many necessary libraries for data science-related tasks, and its integration and control capabilities boost productivity. Python is a standard programming language that is easy for beginners to learn and understand because of its simple syntax; it requires fewer lines of code to be written and is easy to read; for data science projects, Python takes a more streamlined approach; it has a wide range of libraries that allow users to input the action of the library into the code, making it simple to perform matrix computations and optimization (BasuMallick, 2022b). However, compared to R, Python has some disadvantages: fewer data science-specific libraries, visualizations that are less visually appealing and informative, more complicated, and require thorough testing because mistakes appear during runtime.

The following Python packages and libraries help with creating a Logical regression model:

* **Pandas:** used for data analyzing, cleaning, exploring, and manipulating.
* **pandas.api.types:** used for data type validation and type checking in pandas
* **NumPy:** used to working with arrays.
* **Matplotlib:** used for visualization utility.
* **CategoricalDtype:** helps define categorical data types.
* **Seaborn:** used for advanced visualization.
* **spicy.stats:** used for normalization and statistics.
* **statsmodels.api**: used for conducting statistical tests and exploring statistical data.
* **variance\_inflation\_factor:** used for detection of multicollinearity in the regression model.
* **scipy:** used for optimization, integration, interpolation, eigenvalue problems, and other computations, including ANOVA and point-biserial correlation.
* **statsmodels.api:** used for conducting statistical tests and exploring statistical data.
* **sklearn.preprocessing:** used for scaling and encoding categorical variables.
* **train\_test\_split:** used to split the dataset into training and testing sets for evaluating the performance of the regression model.
* **LogisticRegression:** used to fit the regression model to the data.
* **OneHotEncoder:** used to convert categorical variables into a format that can be provided to ML algorithms.

***B3. Justification of using Regression***

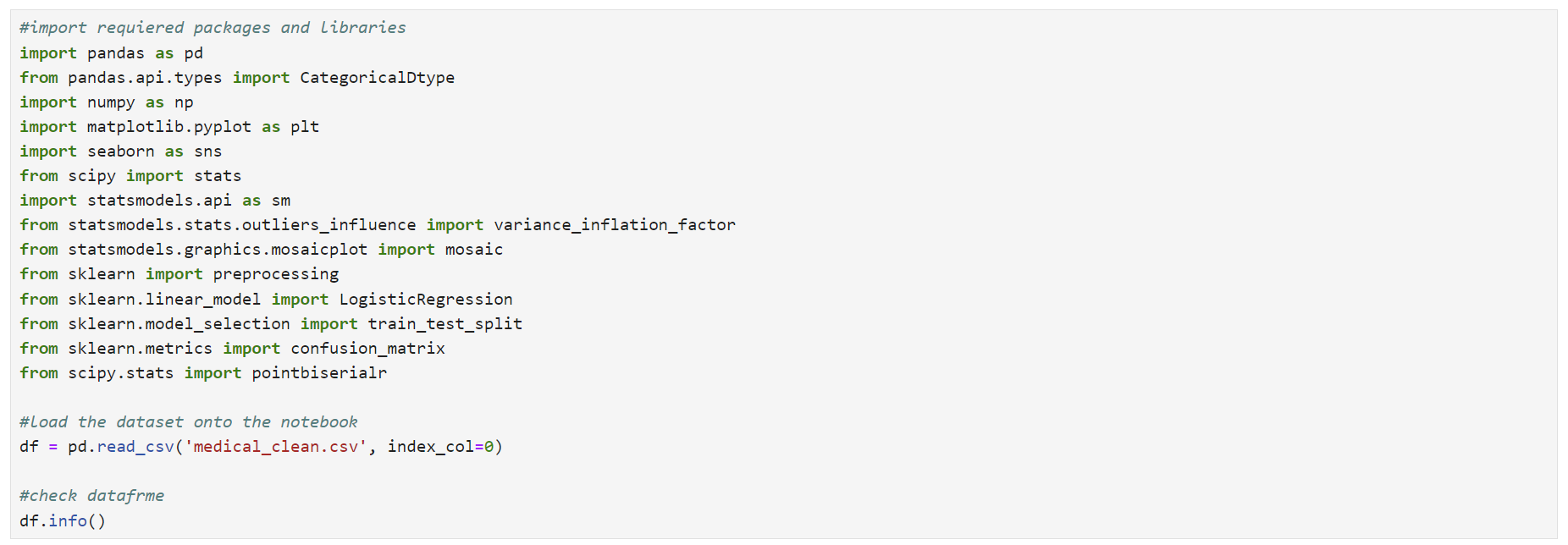
Logistic regression is suitable for analyzing the research question, “What factors contribute to a patient’s readmission?” because the dependent variable, readmission, is binary (whether a patient is readmitted within a month of release or not). Logistic regression is specifically designed to handle binary outcomes, making it an ideal choice for this type of analysis. The logistic regression model for predicting patient readmission includes several key variables. These are the patient’s age as reported during admission, annual income, and self-identified gender (male, female, or nonbinary). Other important variables include the patient’s vitamin D levels measured in ng/mL, the method of initial hospital admission (emergency, elective, or observation), and whether the patient has high blood pressure or has had a stroke. The model also considers the patient’s complication risk level (high, medium, low), whether they are overweight based on age, gender, and height, and whether they have conditions such as arthritis or diabetes. The primary service received during hospitalization (blood work, intravenous, CT scan, MRI), the number of days the patient stayed in the hospital during the initial visit, and the average additional charges for miscellaneous procedures and treatments are included. These variables collectively help understand and predict the likelihood of a patient being readmitted. This enables healthcare providers to focus on the most critical factors and implement targeted interventions to reduce readmission rates.

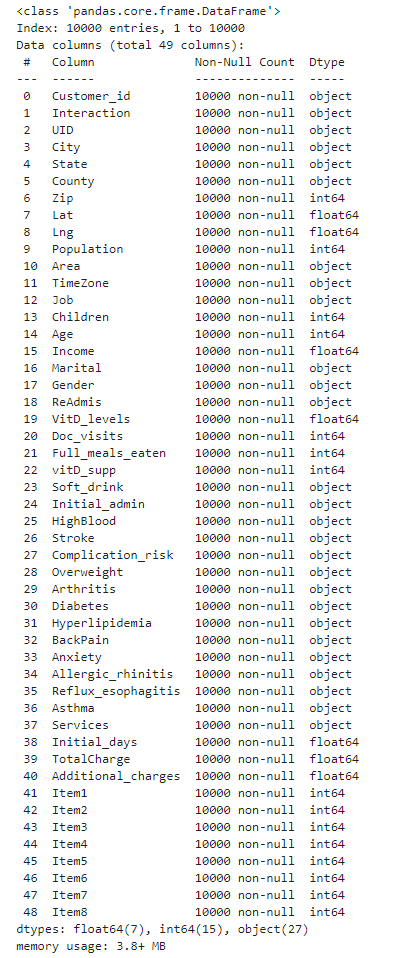
***Part III: Data Preparation and Manipulation***

***C1. Data Cleaning***

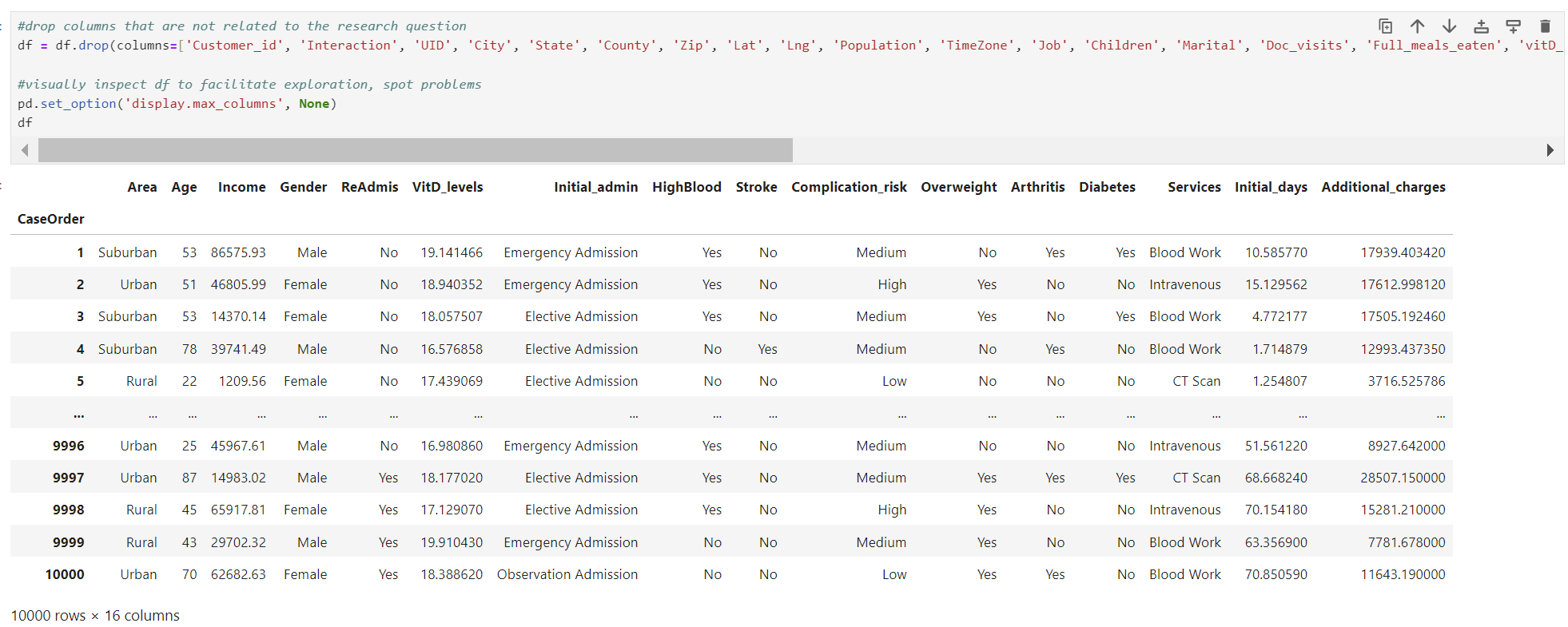
The primary goals for data cleaning are to ensure the dataset is accurate, consistent, and ready for analysis. This involves handling duplicates and missing values, detection and treatment of outliers, correcting errors, and standardizing formats.

The first step is to import required packages and libraries, then using the pd.read\_csv() function, load the medical\_clean.csv file onto the Jupyter Notebook, and lastly, with df.info() function, get the information about the dataset.

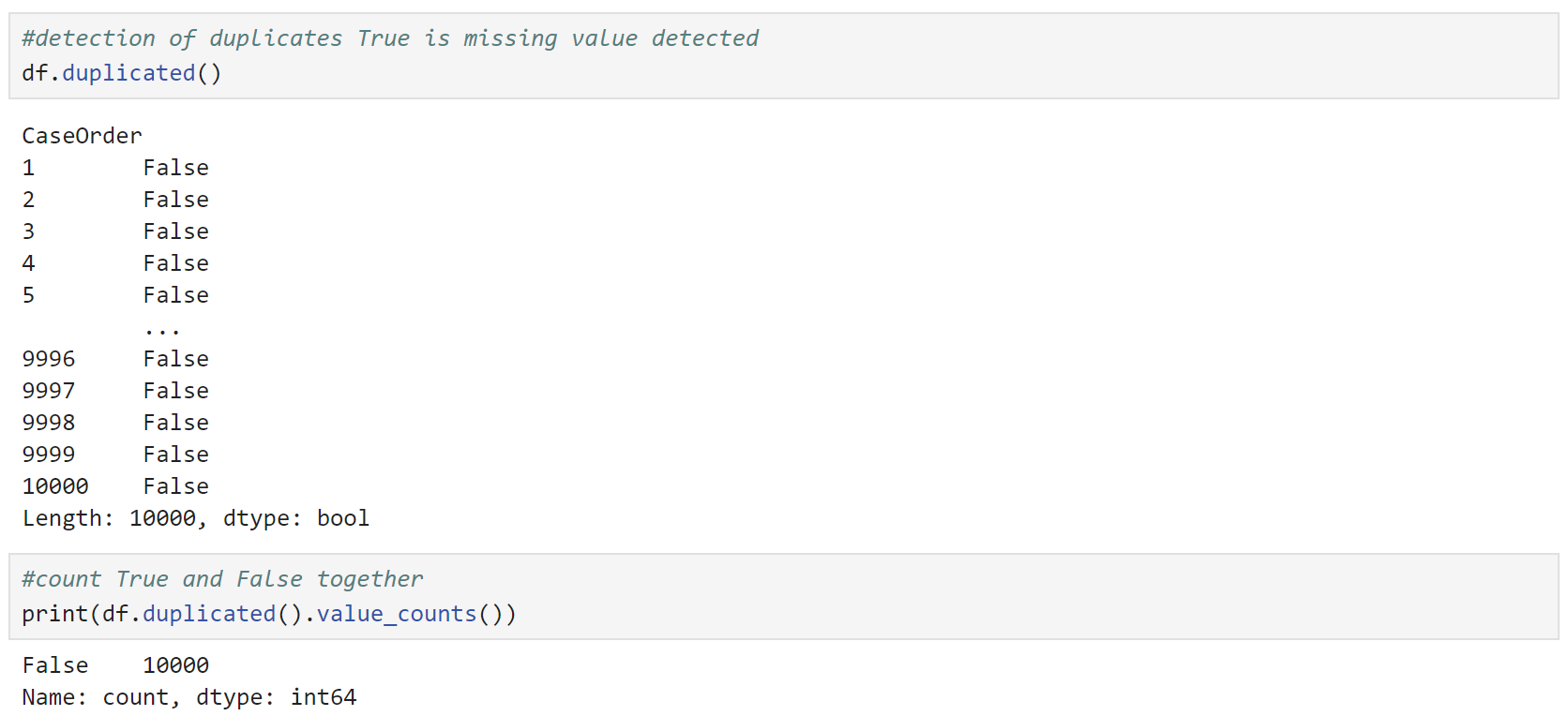




The next step is to drop variables unrelated to the research question using the drop() function. With the pd.set\_option() function, visually inspect the dataset to facilitate exploration and spot problems

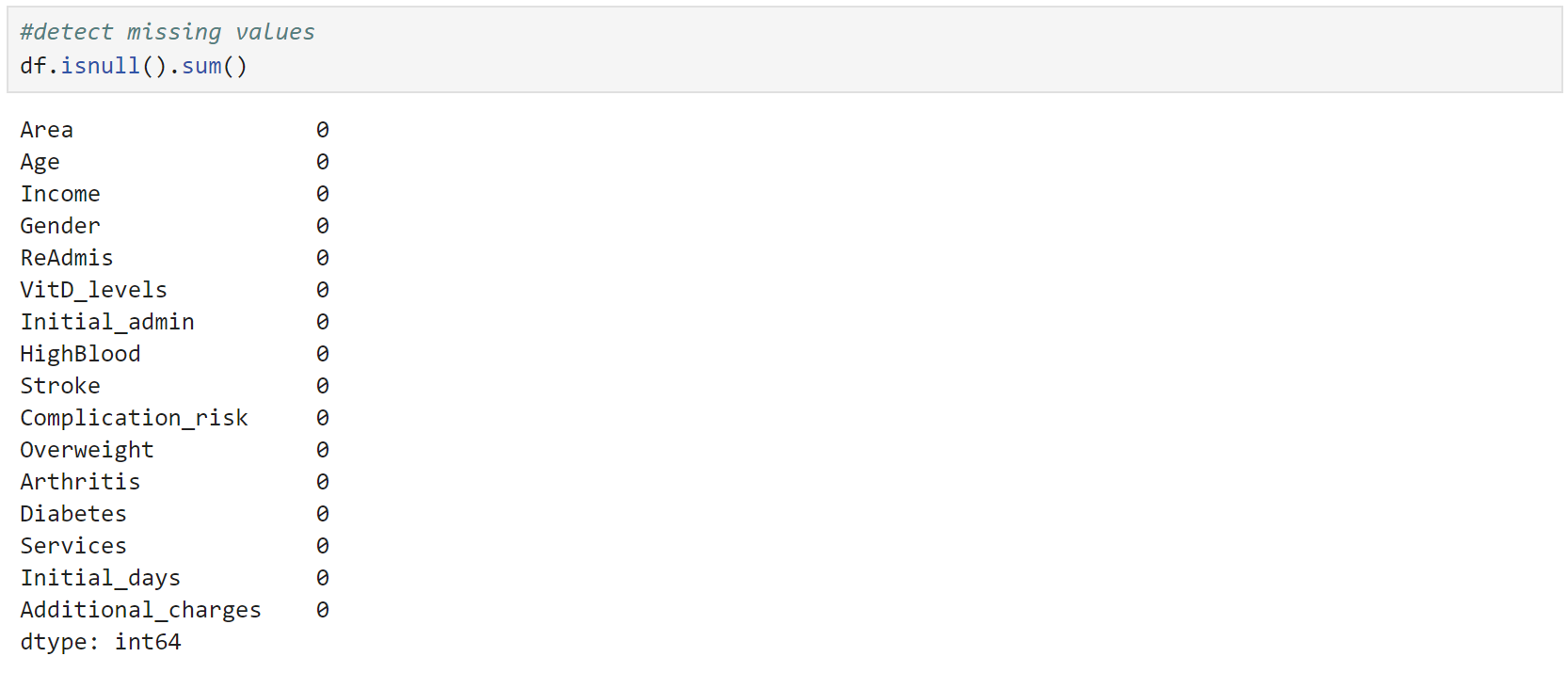


For the detection of duplicates, the duplicated() and duplicated().value\_counts() functions were used.



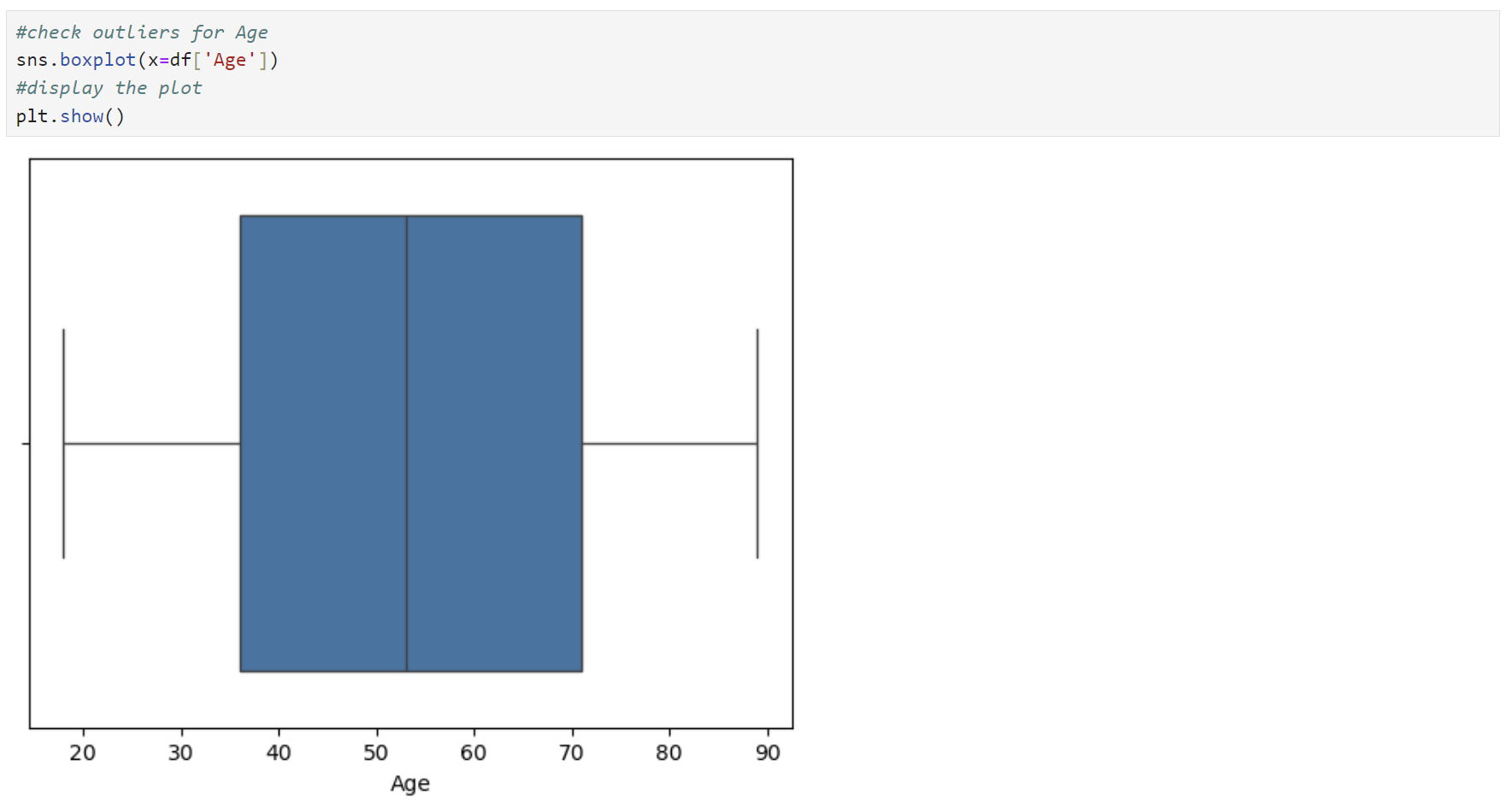
There are no duplicates in the dataset.

Next, the isnull().sum function was used to detect the missing values (qualitative and quantitative variables).

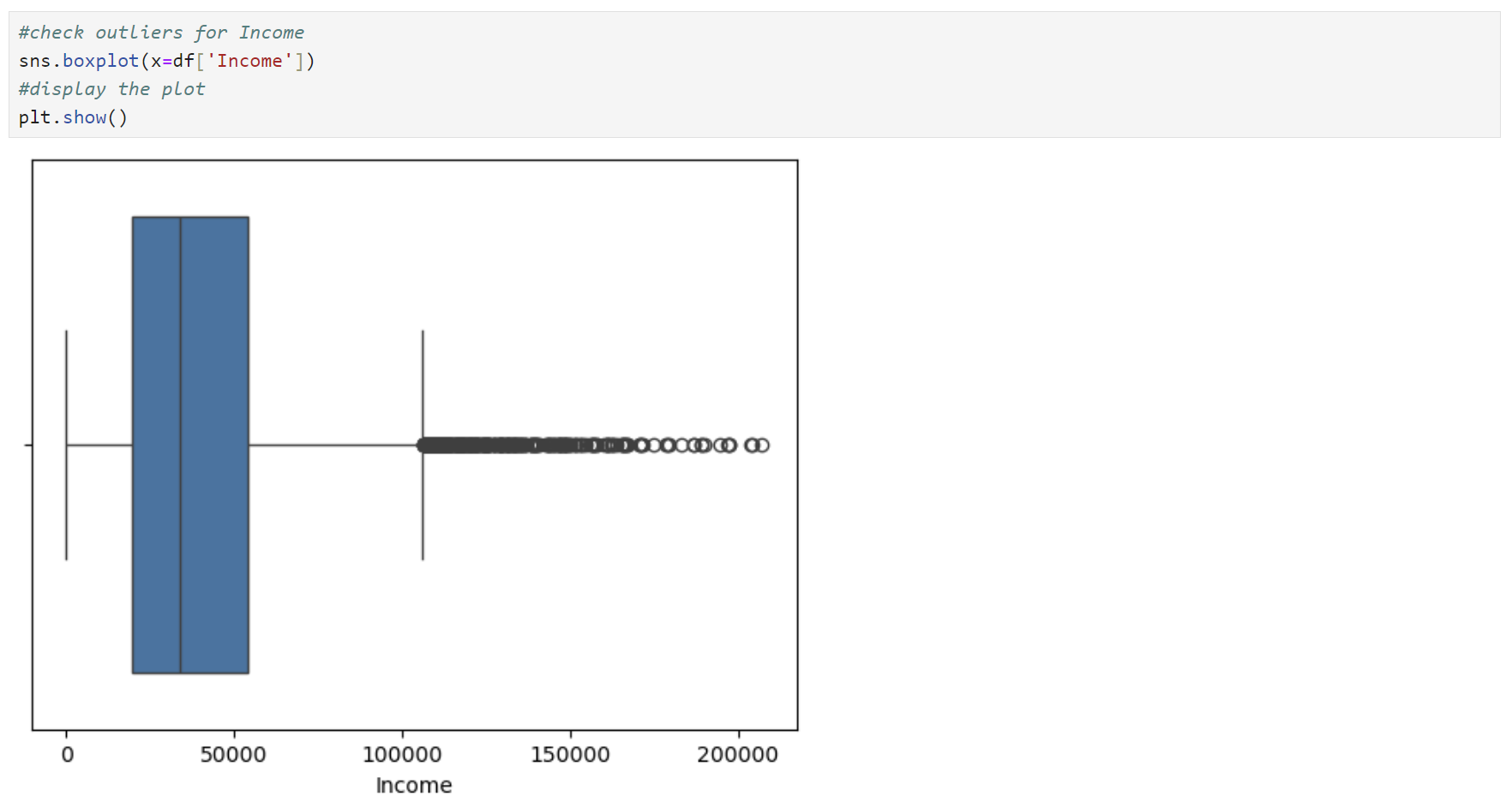


There are no missing values in the dataset.

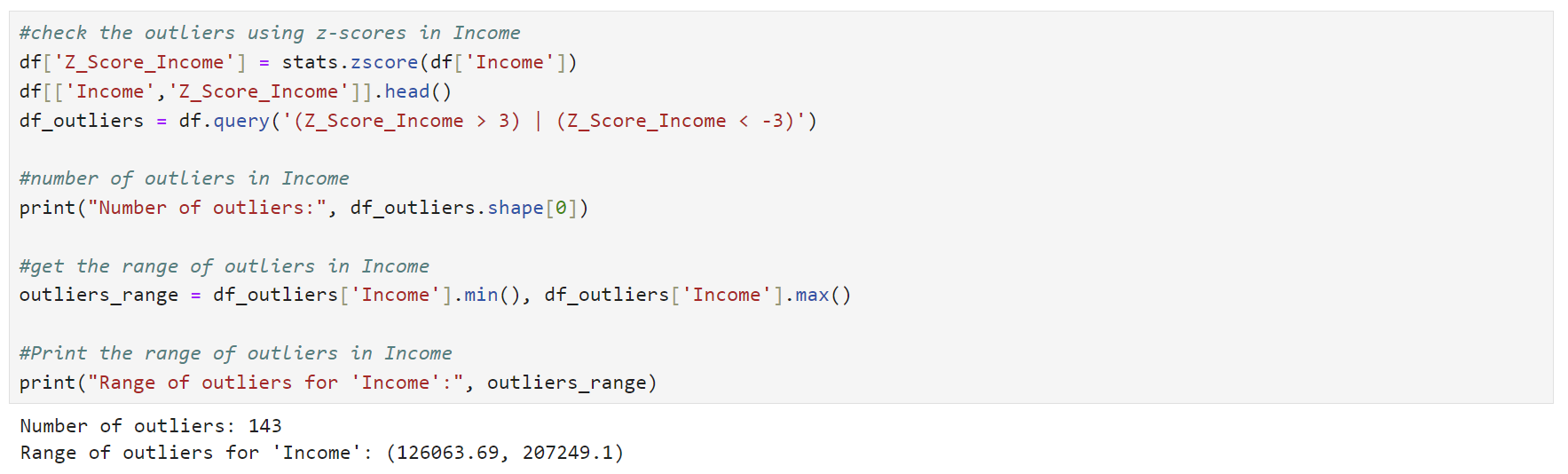
The boxplot and Z\_Score methods detect the outliers for all quantitative variables.



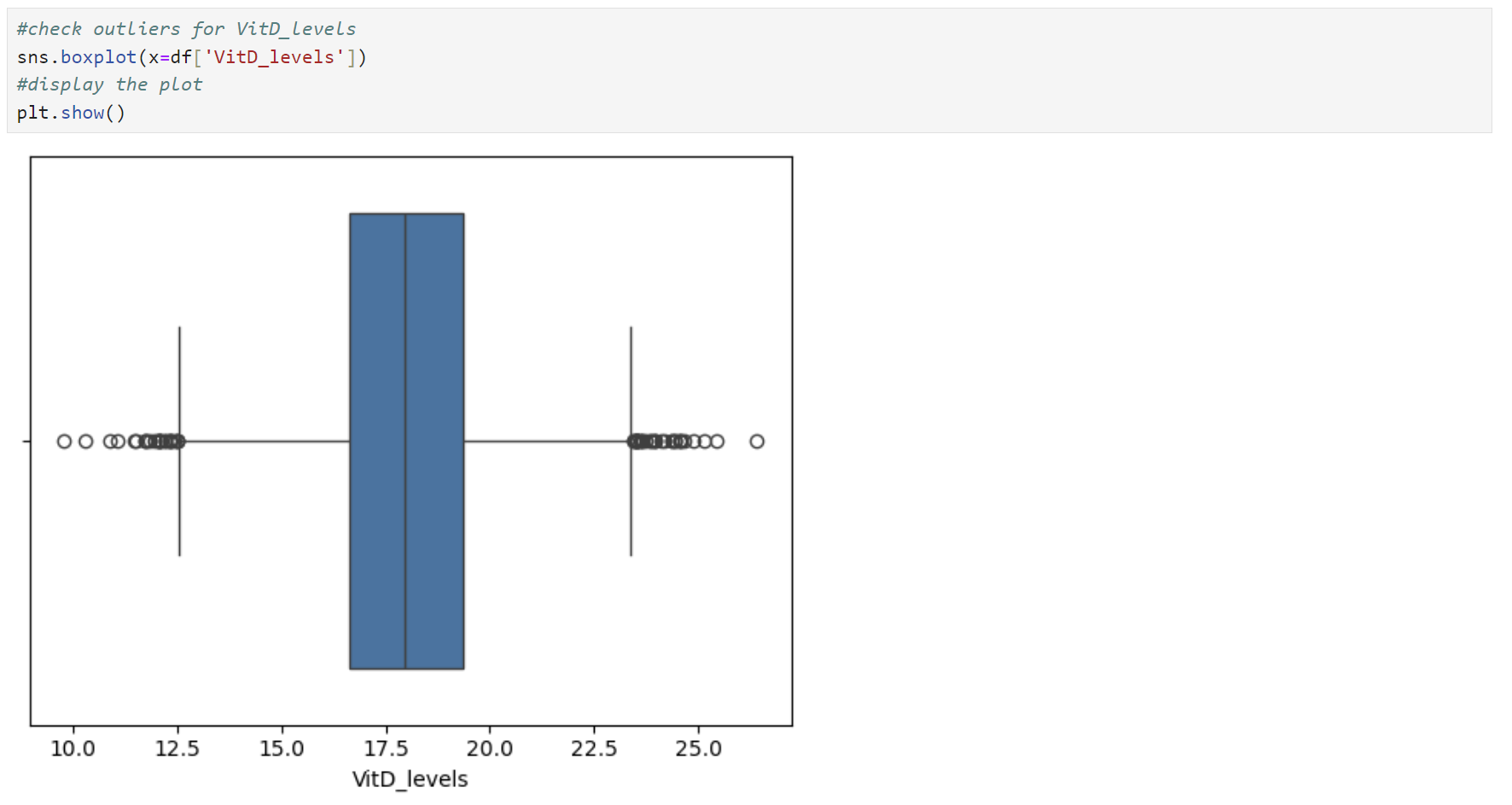
There are no outliers in Age



There are outliers in Income, and the next step is to check the number and the range of outliers.

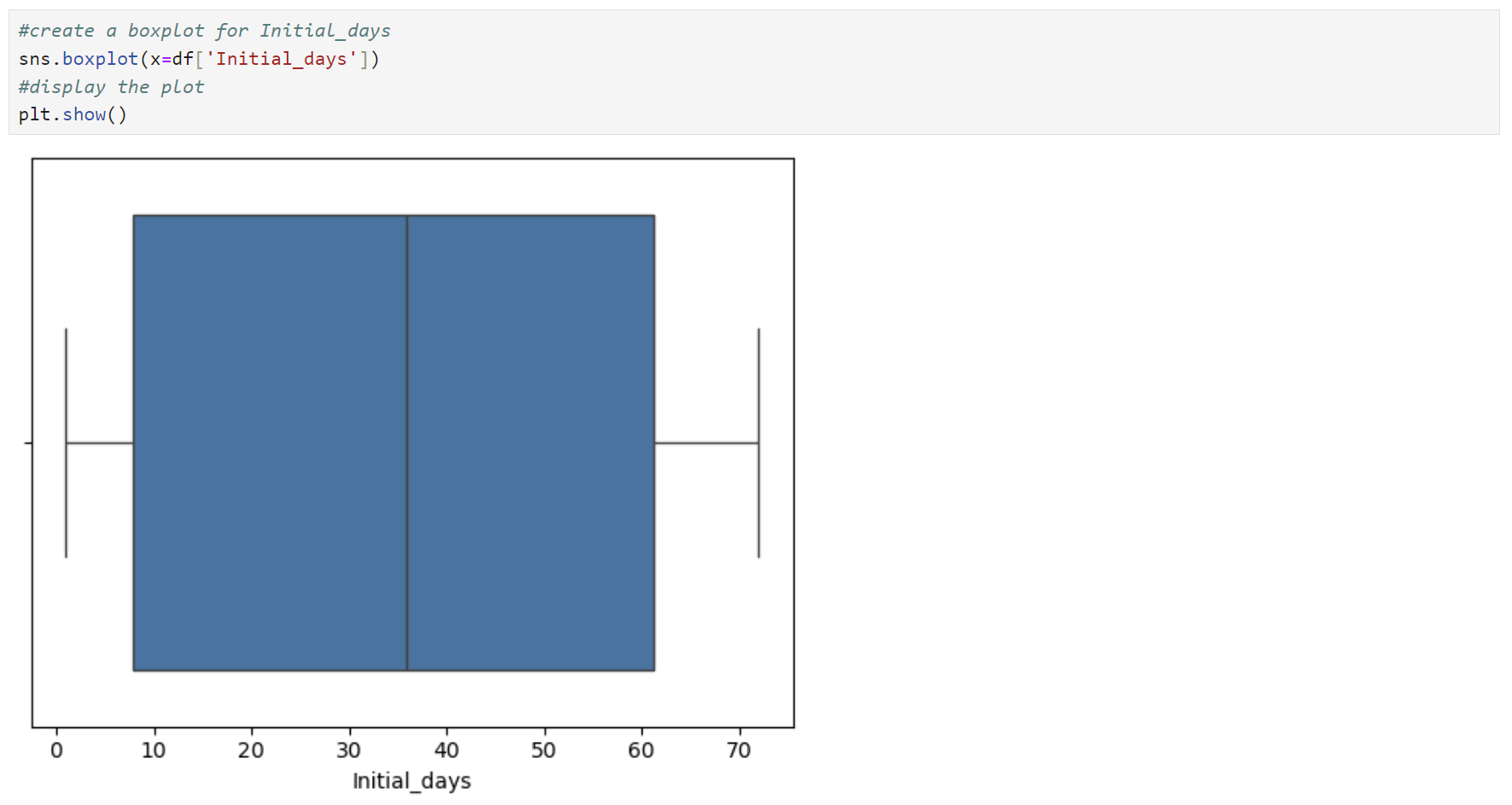


The number of outliers (143 out of 10000) is low, and the range (126063.69, 207249.1) is acceptable. I am going to retain the Income outliers in the dataset.

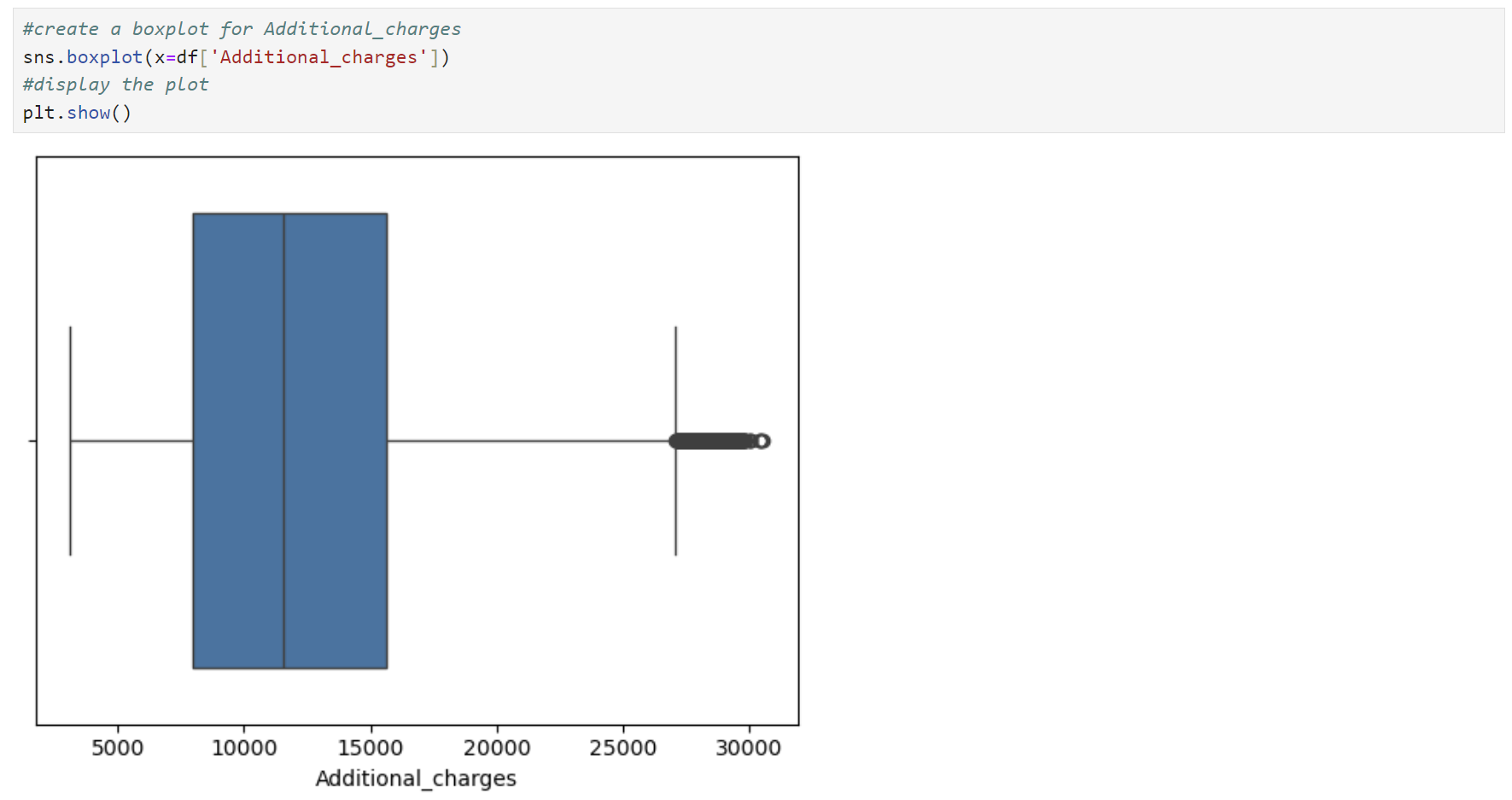


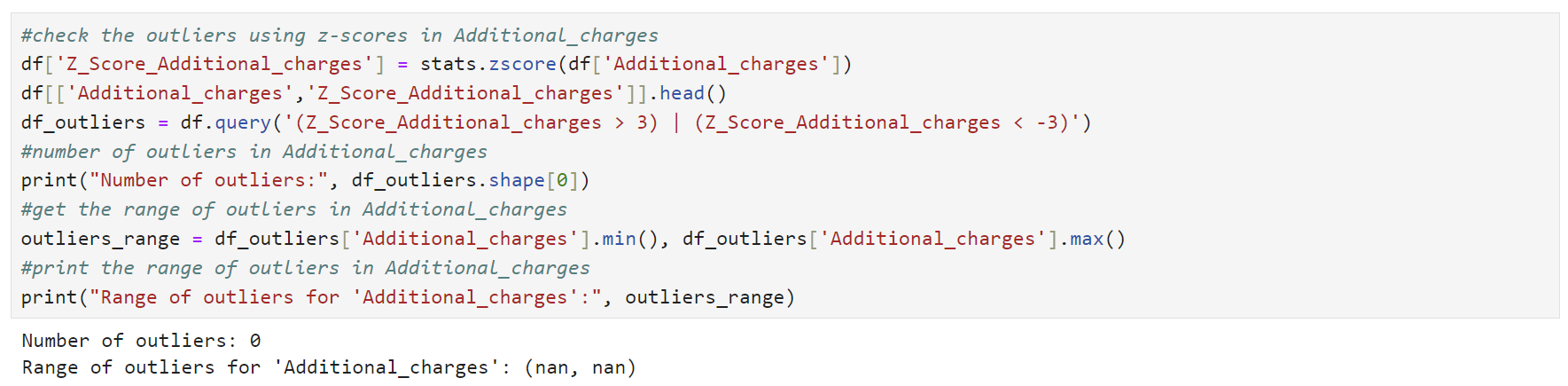


The number of outliers is very low (24 out of 10000), and the range (9.806483, 26.394) looks reasonable. Levels of 20 ng/mL or above are adequate for most people for bone, and overall health levels below 12 ng/mL are too low and might weaken the bones and affect a person’s health (*Office of Dietary Supplements - Vitamin D*, n.d.). Considering that it's information from admitted patients, it's expected to have too low or too high Vit D blood level due to health conditions. I will retain the outliers for VitD\_levels in the dataset.



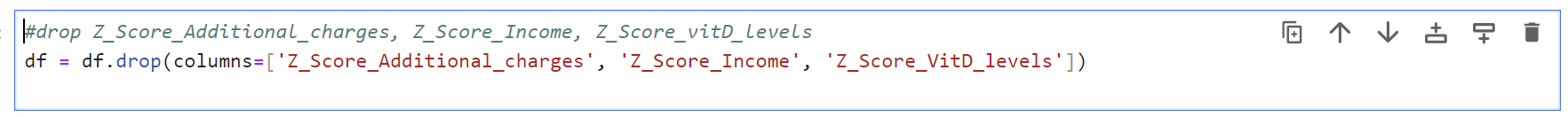
There are no outliers in the Initial\_days.





There are no outliers in the Additional\_charges.

Since I decided to retain the outliers in the dataset, I’ll drop the Z-score columns from it.

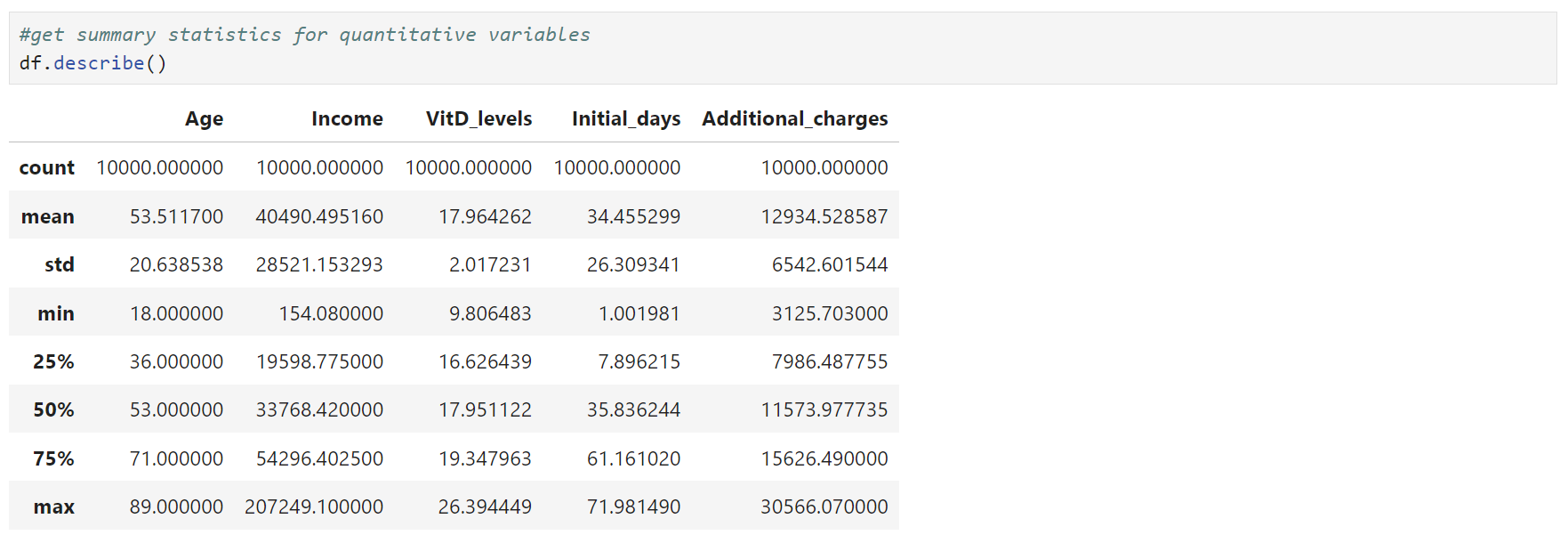


I will convert ReAdmis, Stroke, Arthritis, Diabetes, HighBlood, and Overweight to Boolean, and Area, Gender, Initial\_admin, Complication\_risk, and Services from string to category.



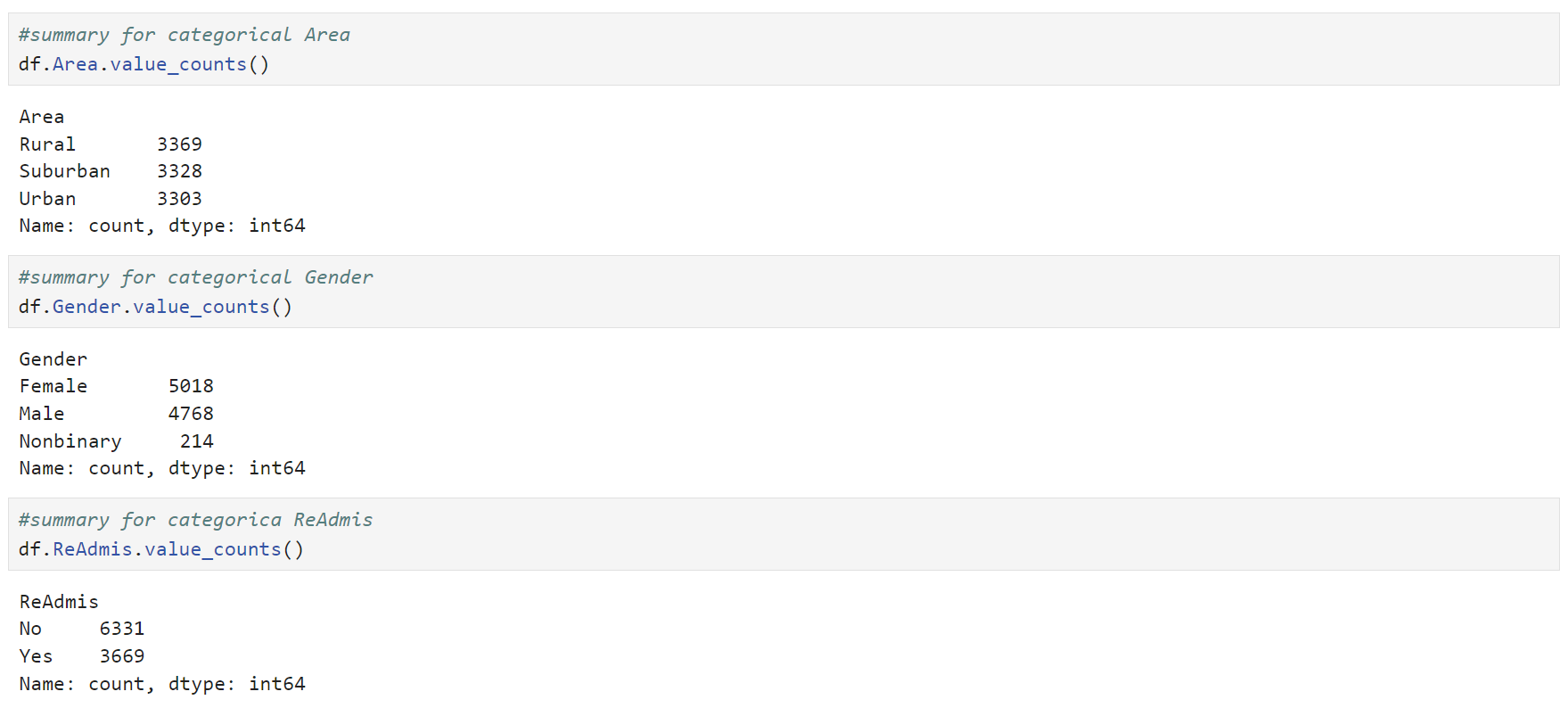
Please see the attached D208Task2\_DataCleaning.ipynb file with the annotated code for cleaning the data.

***C2. Data Exploration***

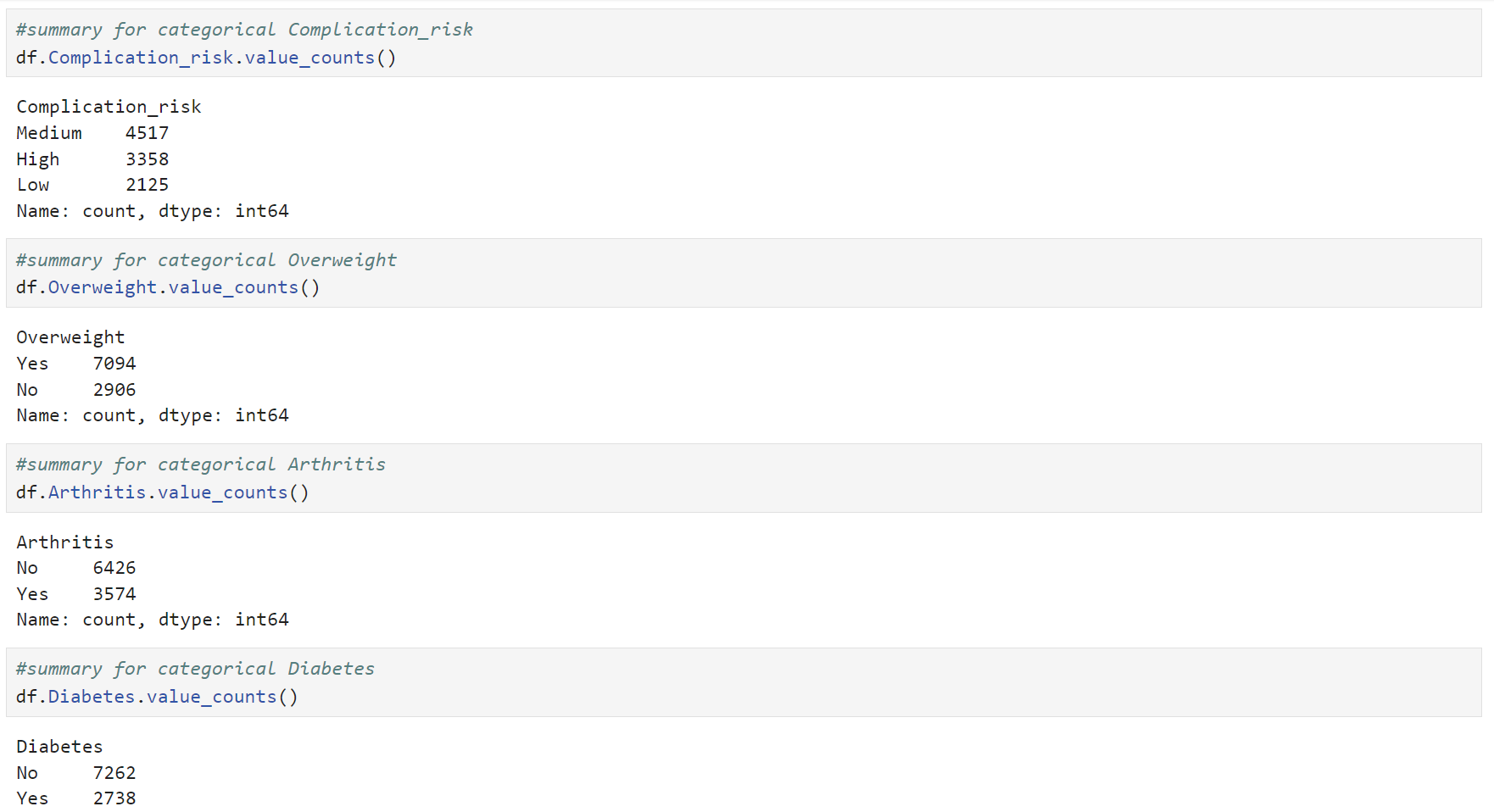


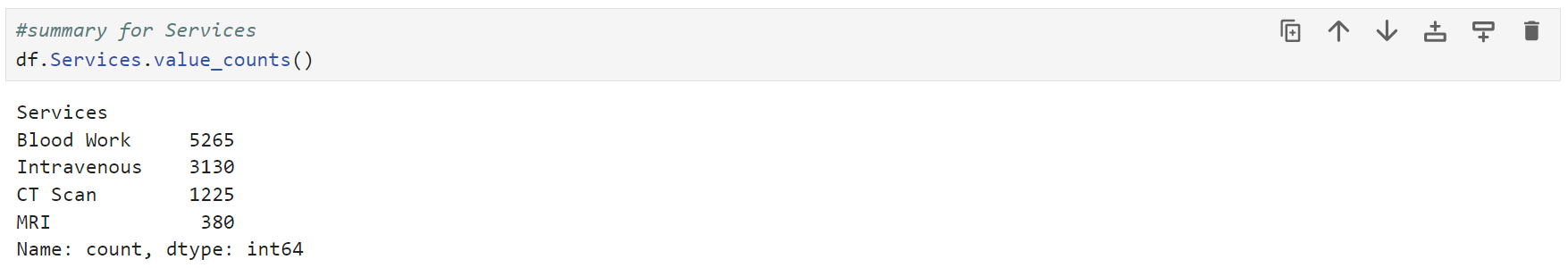
The summary statistics provide detailed information on the continuous variables. The average age is approximately 53.5 years, with a standard deviation of about 20.6 years, indicating a wide age range from 18 to 89. Income shows significant variability, with an average of around $40490 and a standard deviation of $28521, ranging from as low as $154 to over $207000. Vitamin D levels average about 18 ng/mL, with a relatively small standard deviation of 2 ng/mL, indicating most values are close to the mean, ranging from roughly 10 to 26 ng/mL. The initial hospital stay averages around 34.5 days, with a standard deviation of 26.3 days, showing a broad range from 1 to 72 days. Additional charges average about $12935, with a standard deviation of $6543, ranging from $3126 to $30566. These statistics help understand the central tendencies and variability in the data, which are crucial for building and interpreting the logistic regression model, ensuring we account for the diverse patient profiles and their associated costs.

The dataset includes several categorical variables that provide insights into patient demographics and medical conditions. The Area variable shows a nearly even distribution among rural (3369), suburban (3328), and urban (3303) areas. Gender is predominantly female (5018), followed by male (4768) and a smaller number of nonbinary individuals (214). The ReAdmis variable indicates that 3669 patients were readmitted within a month, while 6331 were not. Initial\_admin reveals that most patients were admitted through emergency admission (5060), with fewer through elective (2504) and observation admissions (2436). For HighBlood, 4090 patients have high blood pressure, while 5910 do not. The Stroke variable shows that 1993 patients have had a stroke, compared to 8007 who have not. Complication\_risk is categorized into medium (4517), high (3358), and low (2125) risk levels. The Overweight variable indicates that 7094 patients are overweight, while 2906 are not. Arthritis affects 3574 patients, with 6426 not having the condition. Diabetes is present in 2738 patients, while 7262 do not have it. Lastly, the Services variable shows that the primary services received were blood work (5265), intravenous (3130), CT scan (1225), and MRI (380). These categorical variables help understand the patient population and their medical conditions, which is crucial for predicting readmission rates.



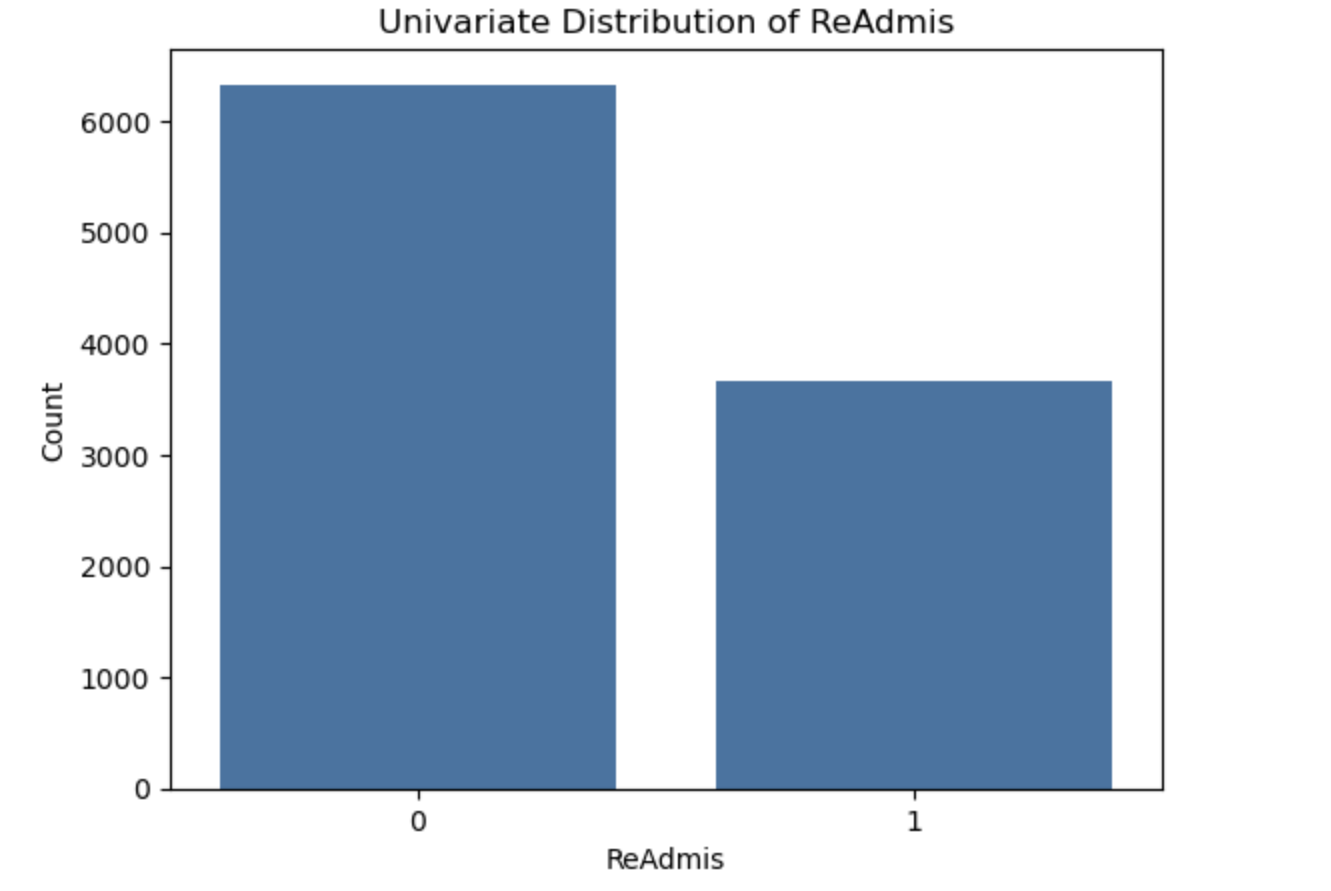




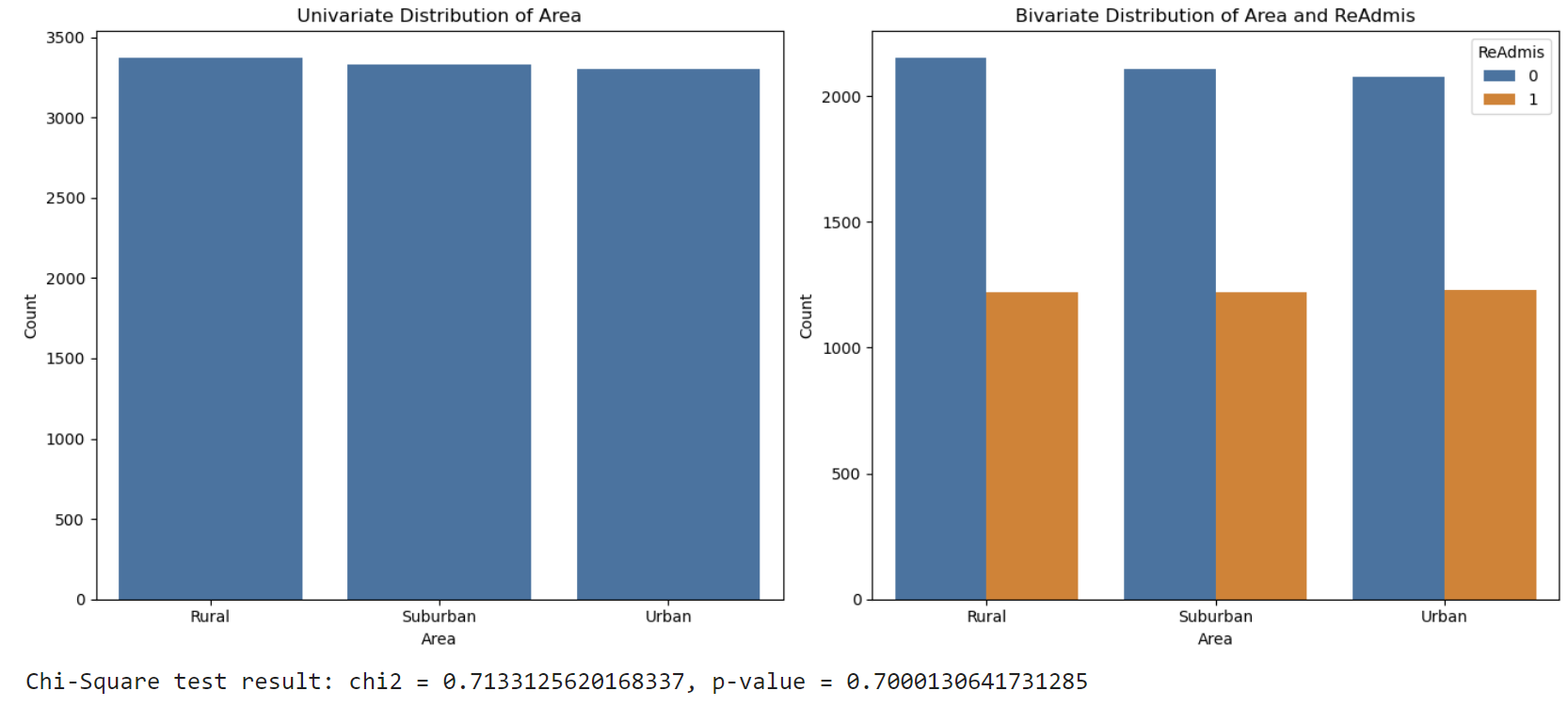


***C3. Visualizations***

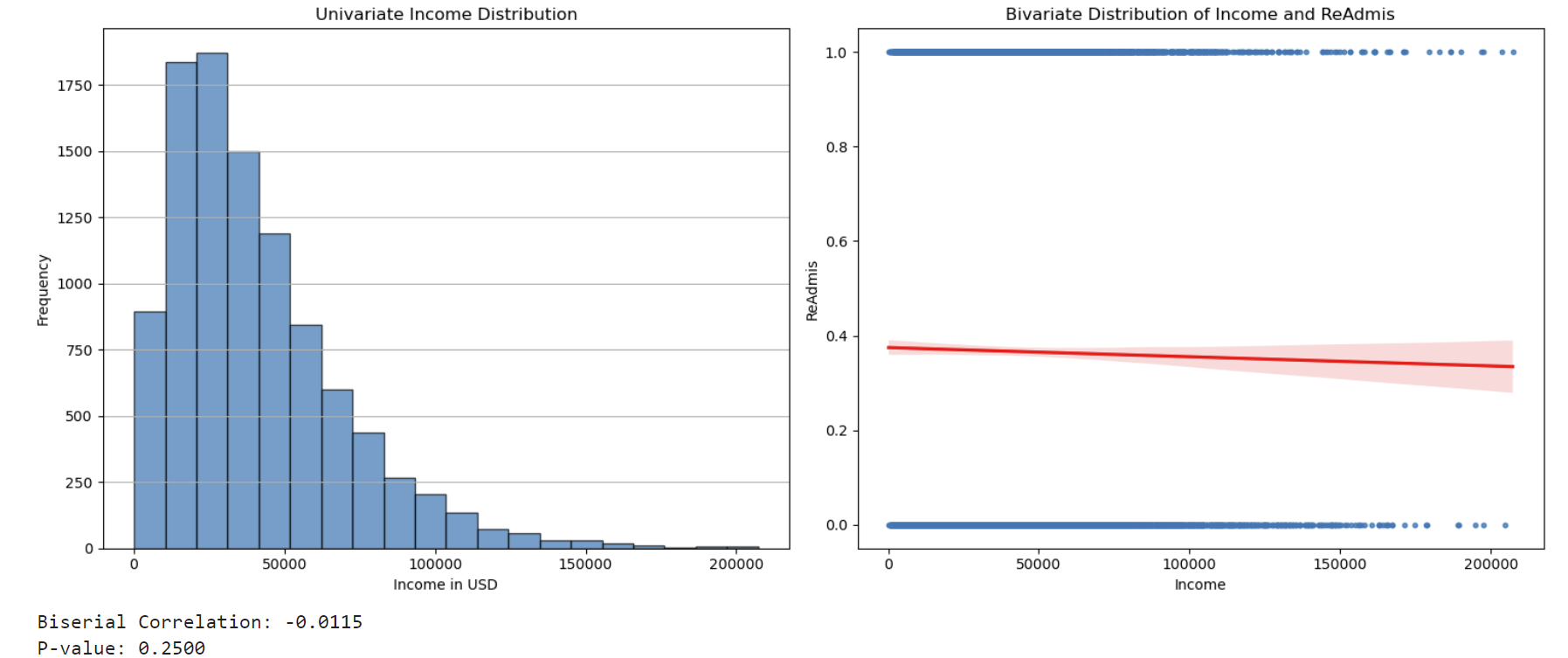
Univariate distribution of ReAdmis.



The univariate distribution of ReAdmis demonstrates that about 1/3 of patients were readmitted within a month after discharge from a hospital.



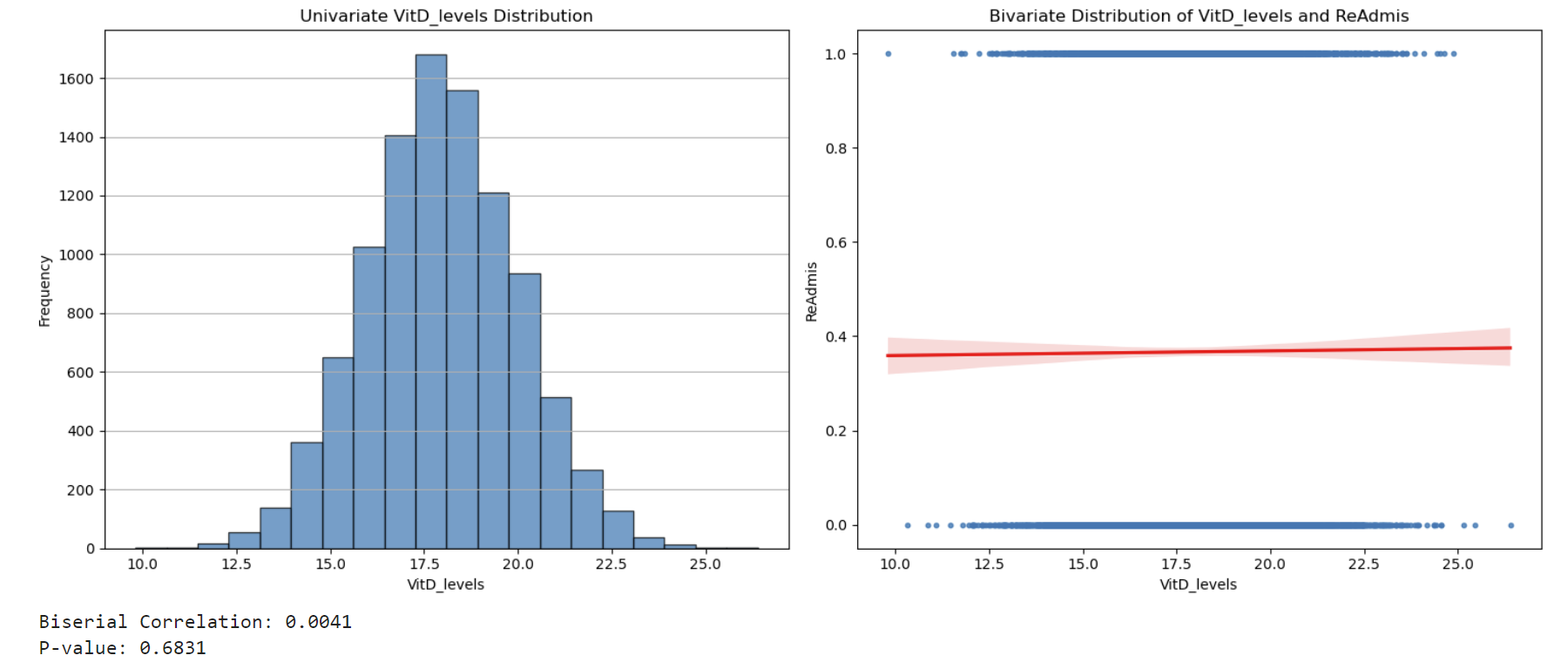
The univariate distribution of Area demonstrates nearly even distribution among rural, suburban, and urban areas. The bivariate distribution of Area and ReAdmis, along with the chi-square results, suggest that there is no correlation between these two variables.



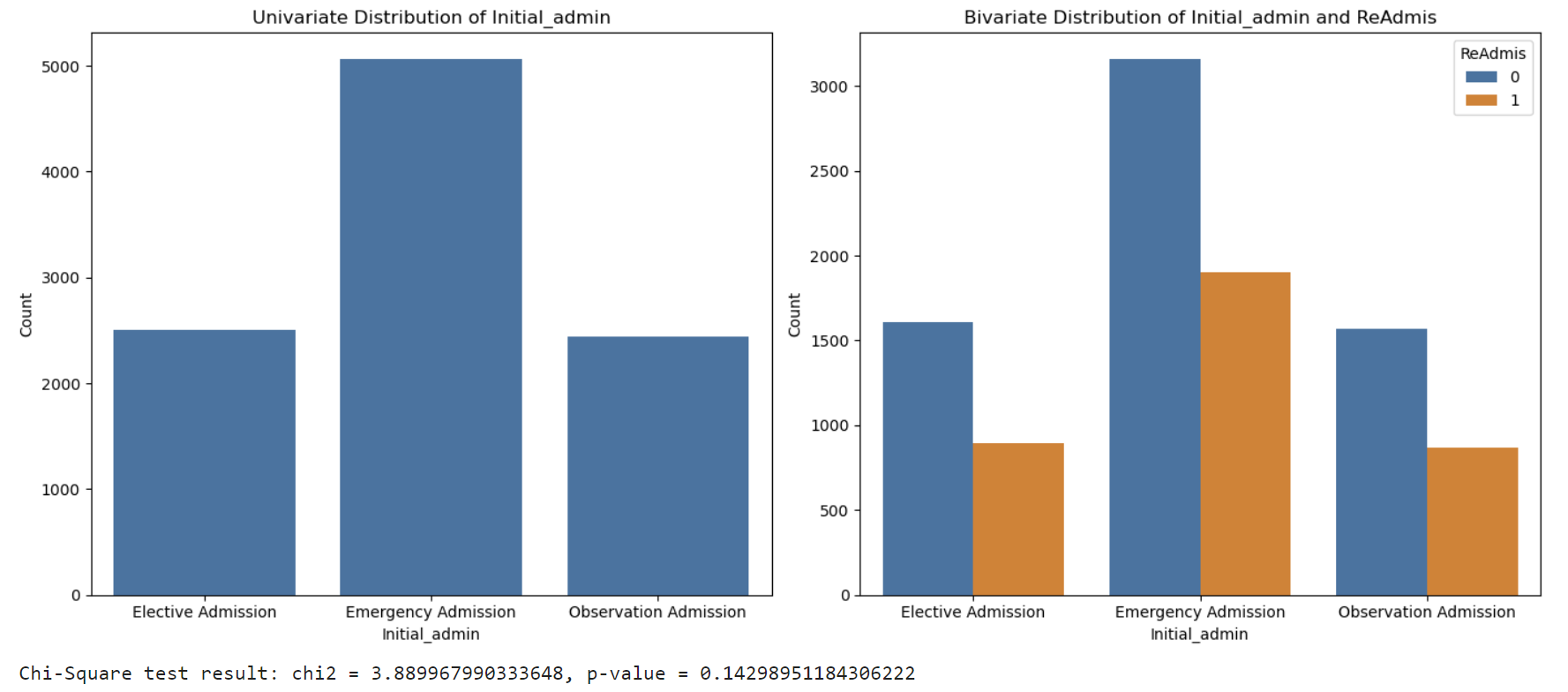
The univariate distribution of Income is right-skewed. The bivariate distribution of Income and ReAdmis and biserial correlation results suggest no correlation between these two variables.



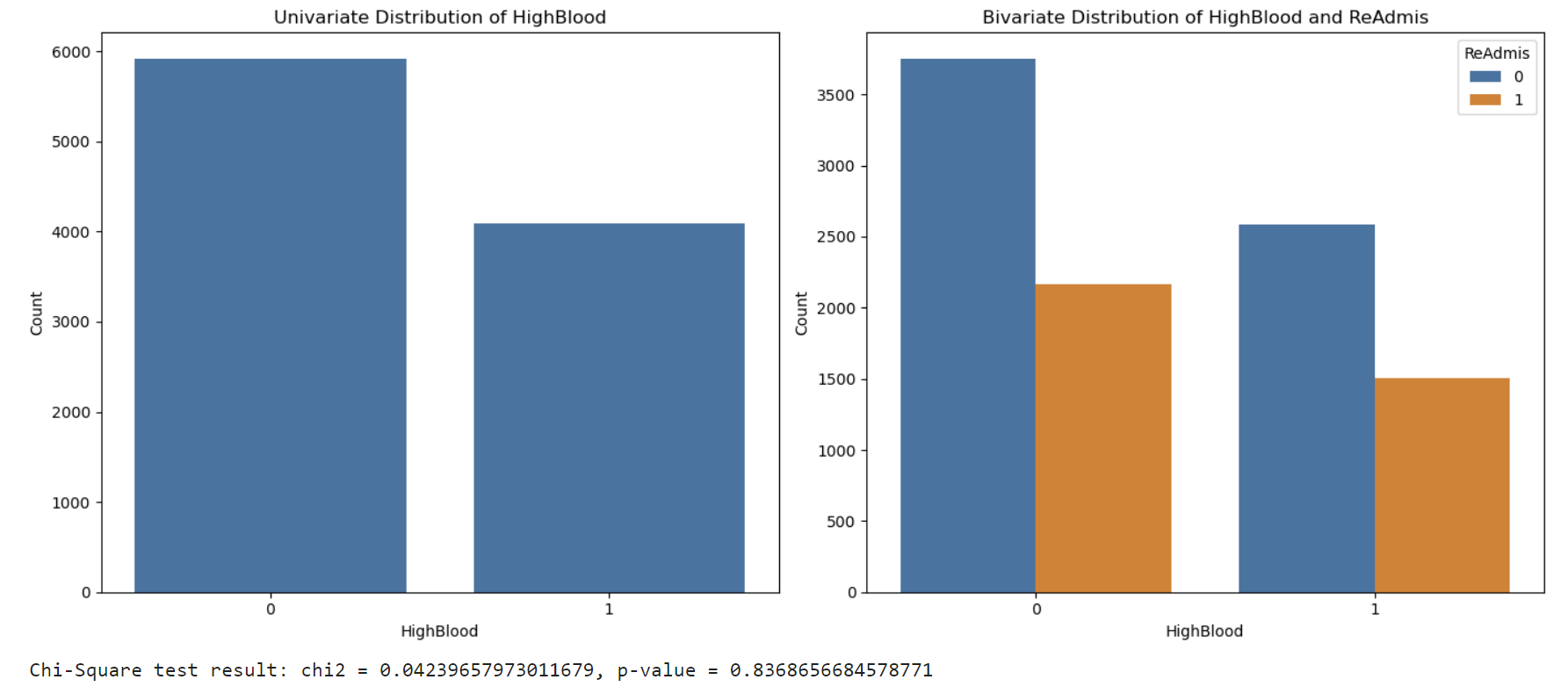
The univariate distribution of gender demonstrates that gender is predominantly female, followed by male, and a smaller number of nonbinary individuals. The bivariate distribution of Gender and ReAdmis and chi-square results suggest no correlation between these two variables.



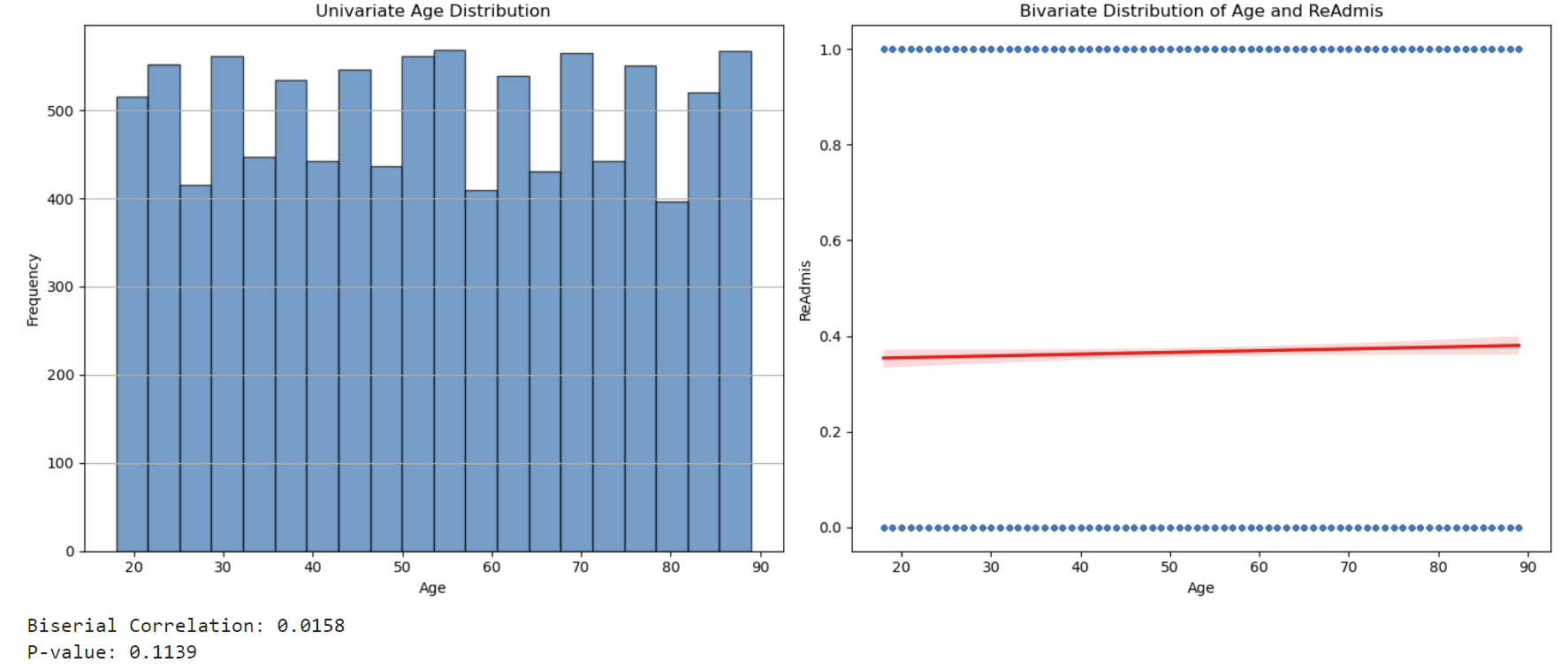
The univariate distribution of VitD\_levels is normal. The bivariate distribution of VitD\_levels and ReAdmis and results of biserial correlation suggest no correlation between these two variables.



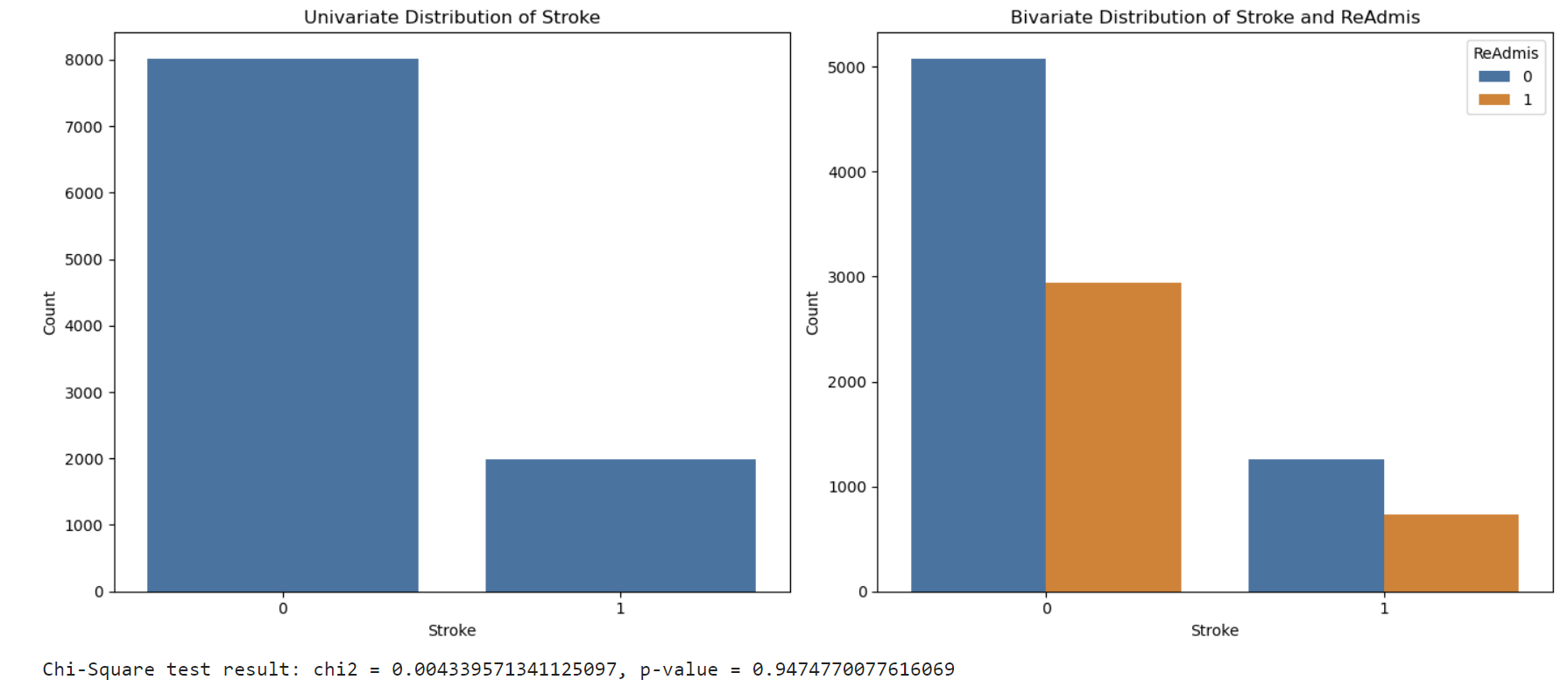
The univariate distribution of Initial\_admin reveals that most patients were admitted through emergency admission, with fewer through elective and observation admissions. The bivariate distribution of Initial\_admin and ReAdmis and chi\_square results suggest no correlation between these variables.



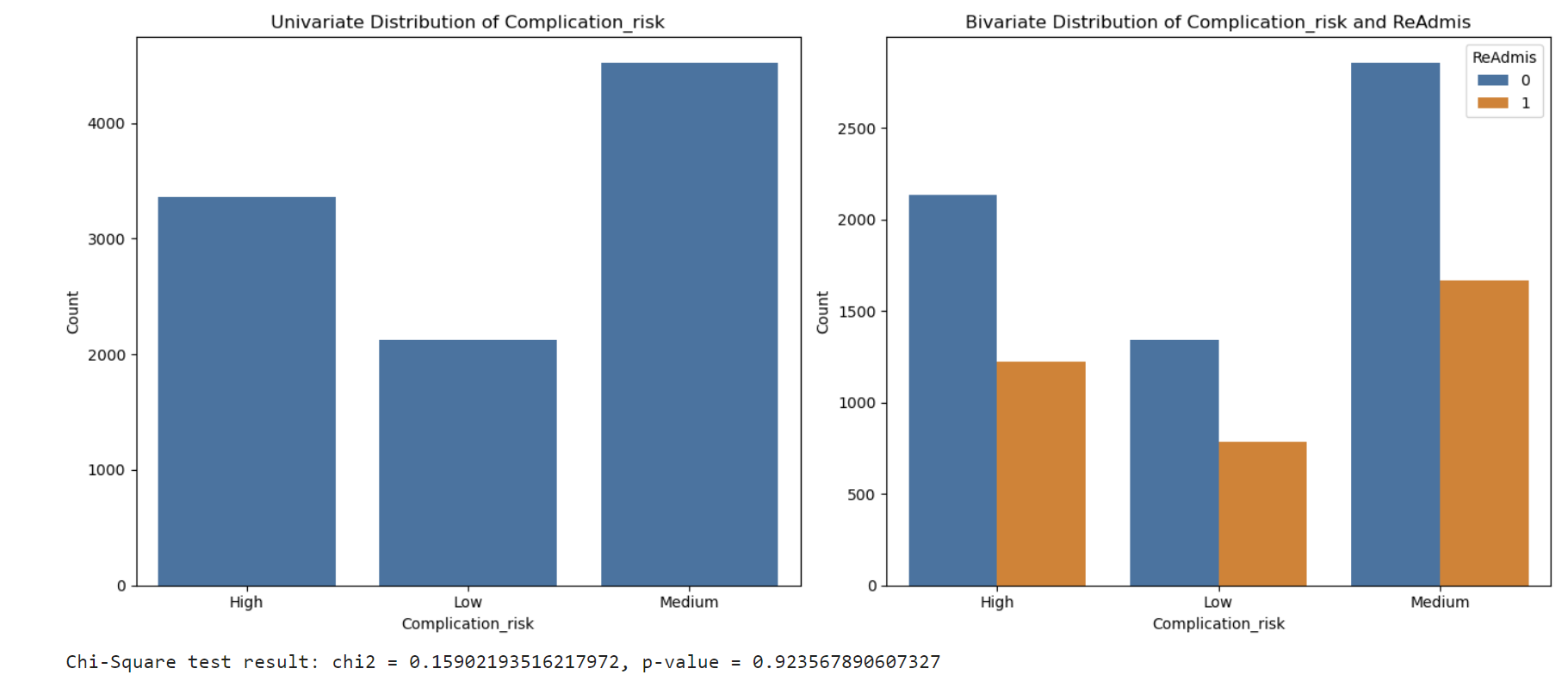
The univariate distribution of HighBlood demonstrates that roughly 4000 patients out of 10000 have high blood pressure. The bivariate distribution of HighBlood and ReAdmis and the chi-square test results suggest no correlation between these variables.



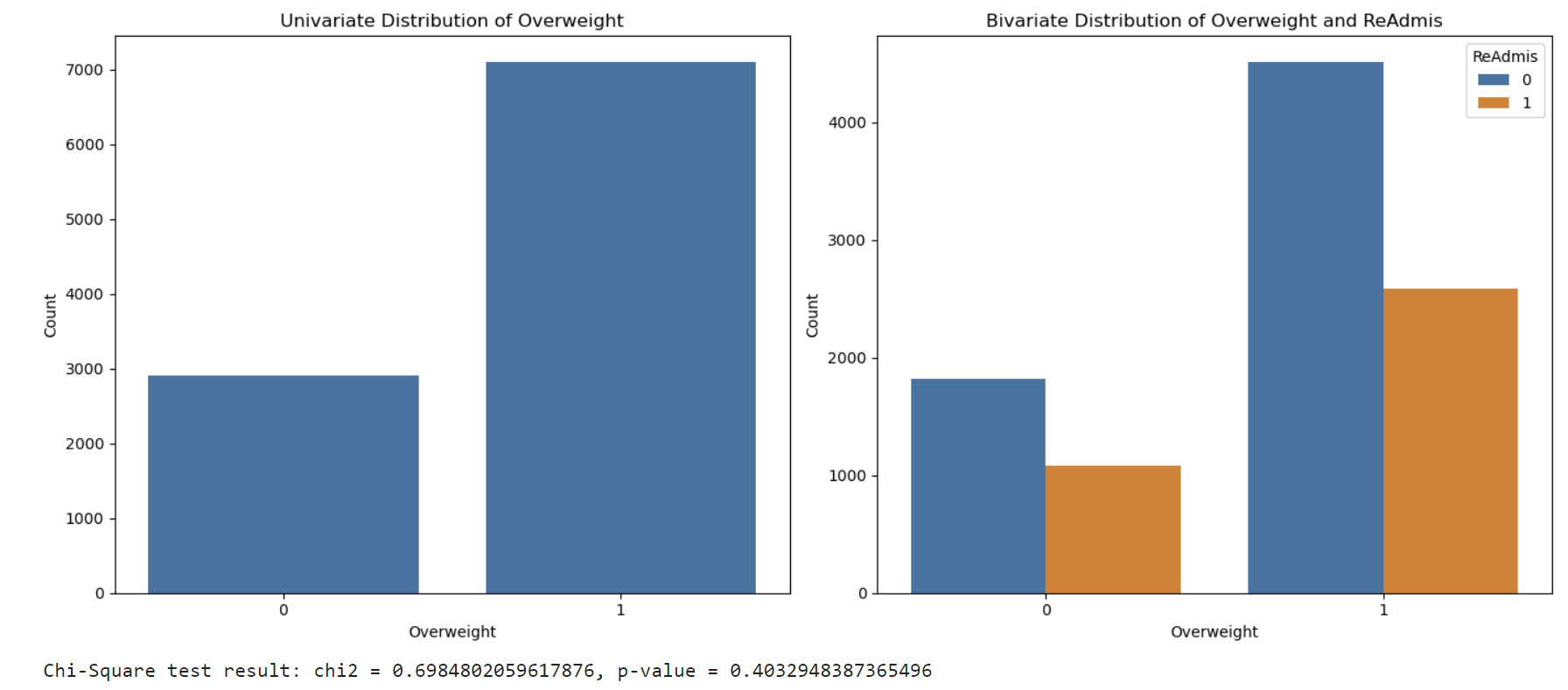
The Age distribution is uniform. The bivariate plot and the biseral correlation results suggest no relationship between variables Age and ReAdmis.



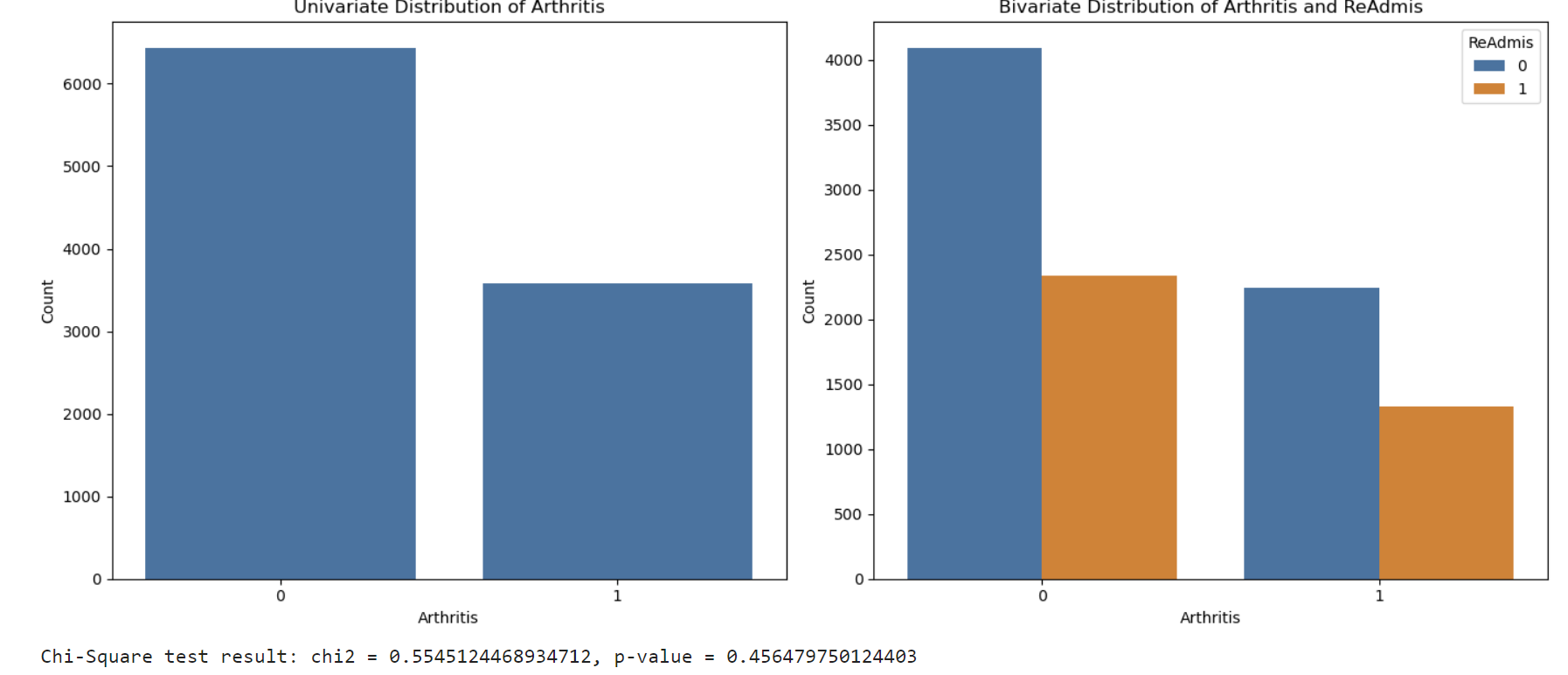
The univariate distribution of Stroke demonstrates that around 2000 patients out of 10000 have had a stroke. The bivariate distribution and the results of the chi-square test suggest that there is no relationship between Stroke and ReAdmis.



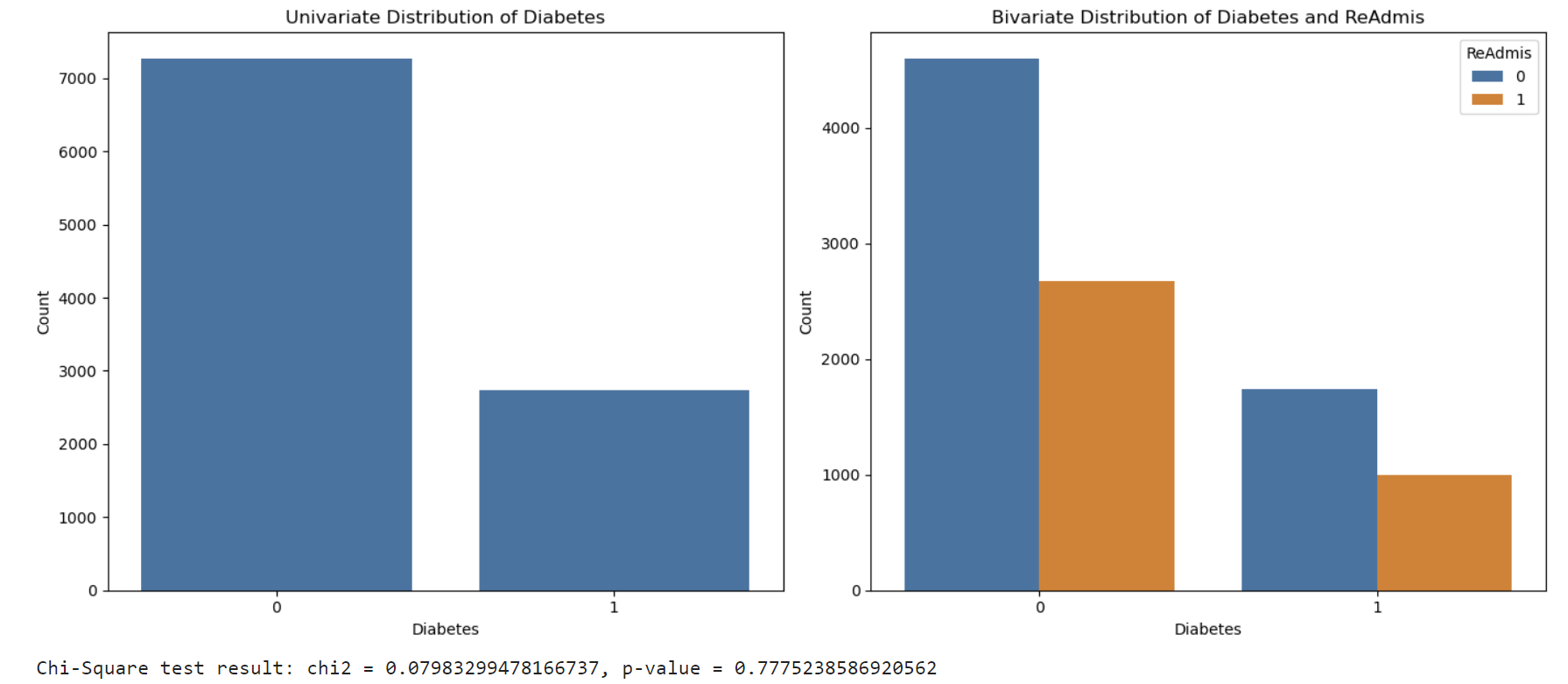
The univariate distribution of Complication\_risk demonstrates that the medium risk is the most common, followed by high and low. The bivariate distribution and the chi-square results suggest no relationship between Complication\_risk and ReAdmis.



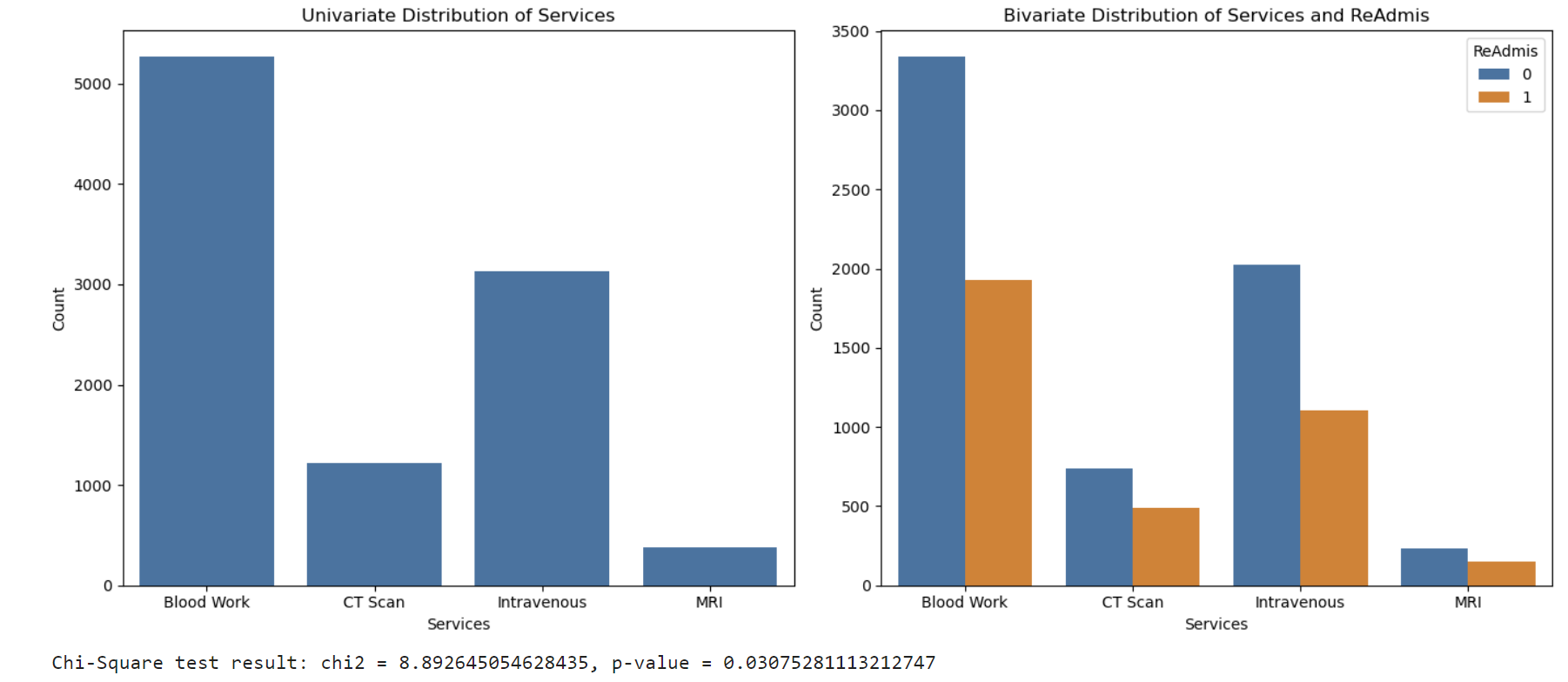
The univariate distribution of Overweight demonstrates that around 7000 patients out of 10000 are overweight. The Bivariate distribution and the chi-squared results suggest no correlation between Overweight and ReAdmis.



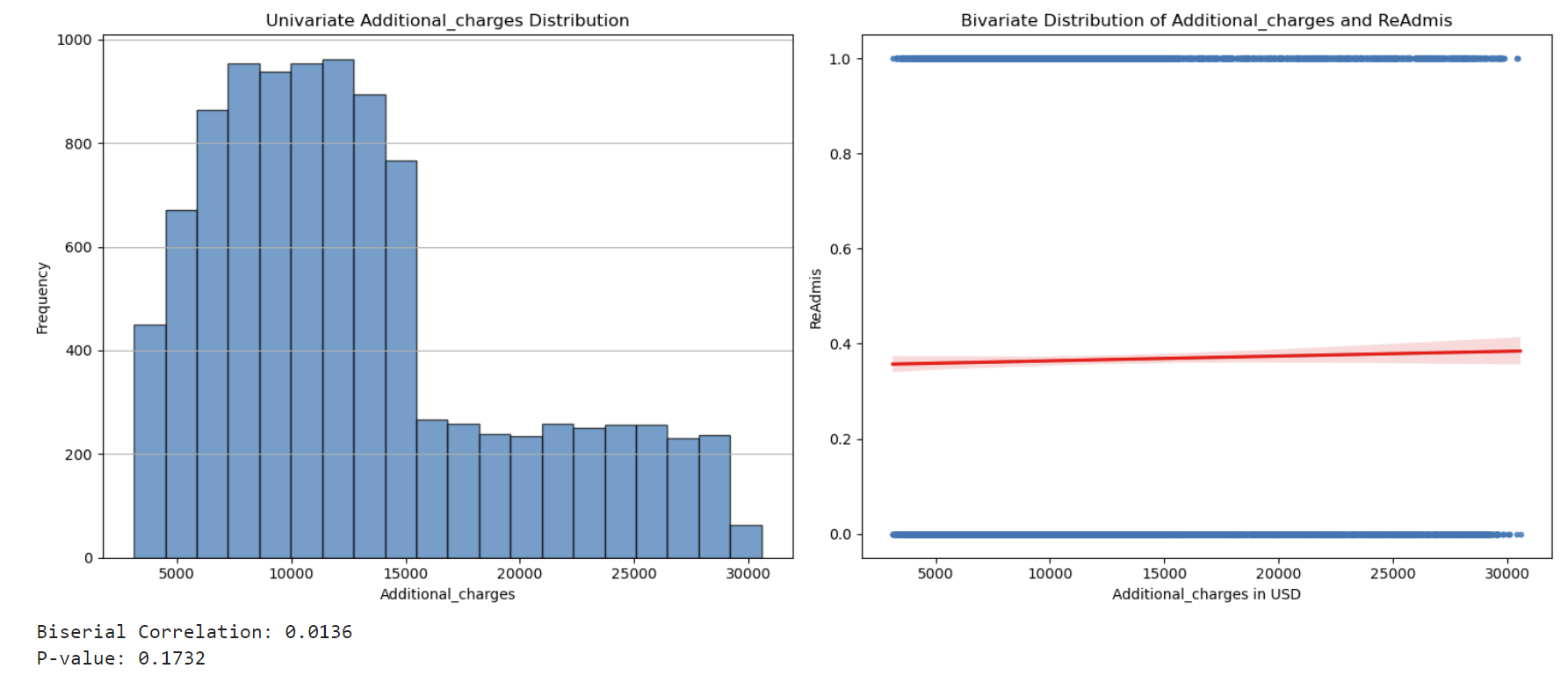
The univariate distribution of Arthritis demonstrates that around 3800 out of 10000 patients have arthritis. The bivariate distribution of Arthritis and ReAdmis and chi-square results suggest no correlation between these variables.



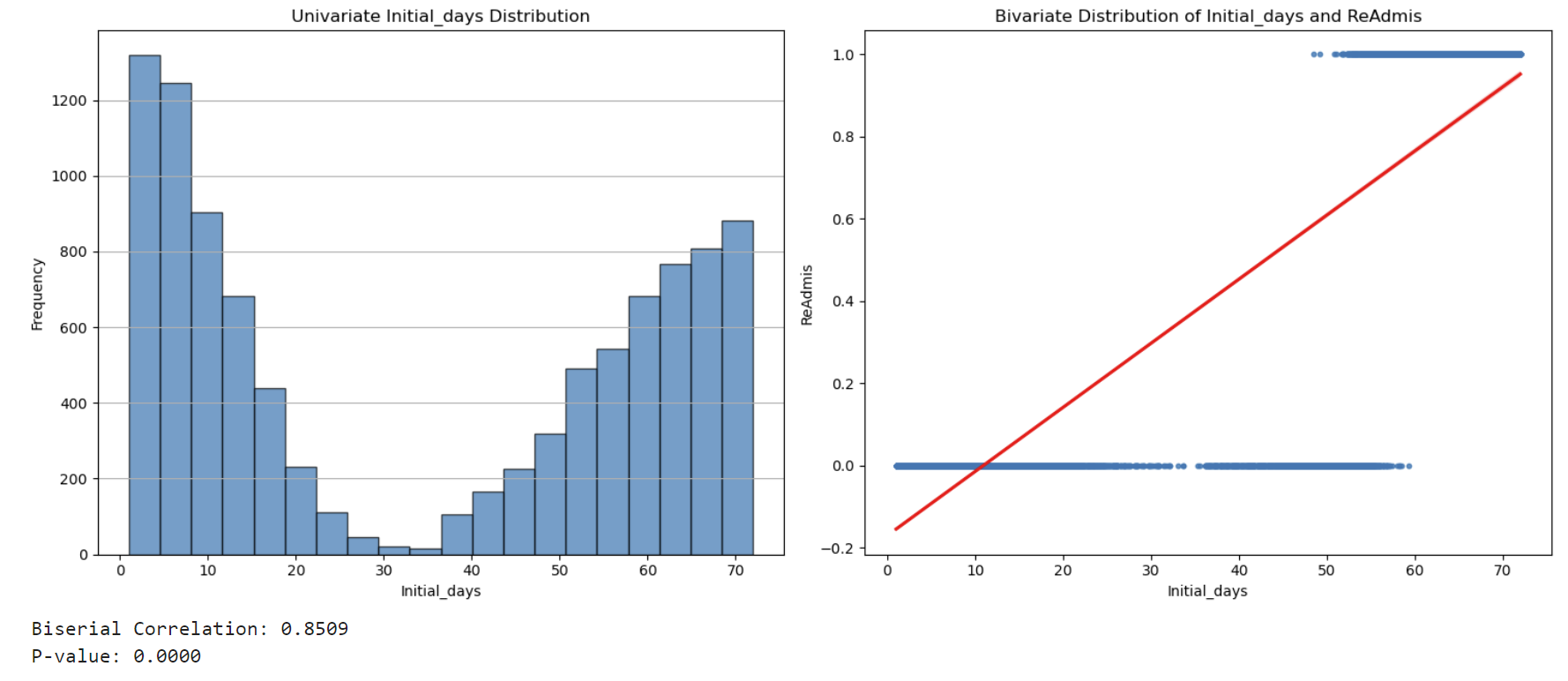
The univariate distribution of Diabetes shows that around 1/3 of the patients have diabetes. The bivariate distribution and chi-square results suggest no correlation between Diabetes and ReAdmis.



The univariate distribution of Services demonstrates that Blood work is the most common service, followed by Intravenous, with CT Scans and MRIs being the least common. The bivariate distribution of Services and ReAdmis and the chi-square results suggest a relationship between these variables.



The univariate distribution of Additional\_charges is right skewed. The bivariate distribution of Additional\_charges and ReAdmis and results of the biseral correlation suggest no relationship between these variables.



The univariate distribution of Initial\_days is bimodal. The bivariate distribution of Initial\_days and ReAdmis and results of the biseral correlation show a strong positive correlation between these two variables.

***C4. Data Transformation***

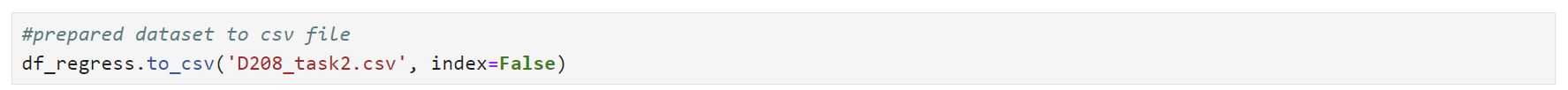
Most statistical methods/machine learning algorithms mining work exclusively with numeric data, meaning the necessity to convert categorical data to numeric (*Redirecting*, n.d.-n).  When creating a logistic regression model, one-hot encoding is essential for handling categorical variables. It transforms categorical variables into a set of binaries (0 or 1) indicator variables, where each level of the original category becomes a separate feature. This process ensures that the model can learn from the categorical information effectively.

One-hot encoding was created using the pd.get\_dummies() function (GeeksforGeeks, 2024). The drop\_first=True argument drops the first category for each feature to avoid multicollinearity. Using df\_regress.replace({True: 1, False: 0}), I converted boolean values to numeric (1 and 0) and applied pd.to\_numeric to convert all columns to numeric.



***C5. Prepared Dataset***

Prepared df\_regress Dataset was saved to a new csv file. Please see the attached “D208\_task2.csv” file.



***Part IV: Model Comparison and Analysis***

***D1. Initial Model***

Logistic regression is a statistical method to model the relationship between a dependent binary variable and one or more independent variables. For the model creation, I will use the "kitchen sink" approach when all variables are thrown into a model, regardless of their relevance to the outcome variable or statistical significance. For the research question, ReAdmis is the dependent variable. The independent variables (Age, Income, VitD\_levels, HighBlood, Stroke, Overweight, Arthritis, Diabetes, Initial\_days, Area\_Suburban, Area\_Urban, Gender\_Male, Gender\_Nonbinary, Initial\_admin\_Emergency Admission, Initial\_admin\_Observation Admission, Complication\_risk\_Low, Complication\_risk\_Medium, Services\_CT Scan, Services\_Intravenous, and Services\_MRI) will be included in X dataset.





***D2. Model Reduction Method and Justification***

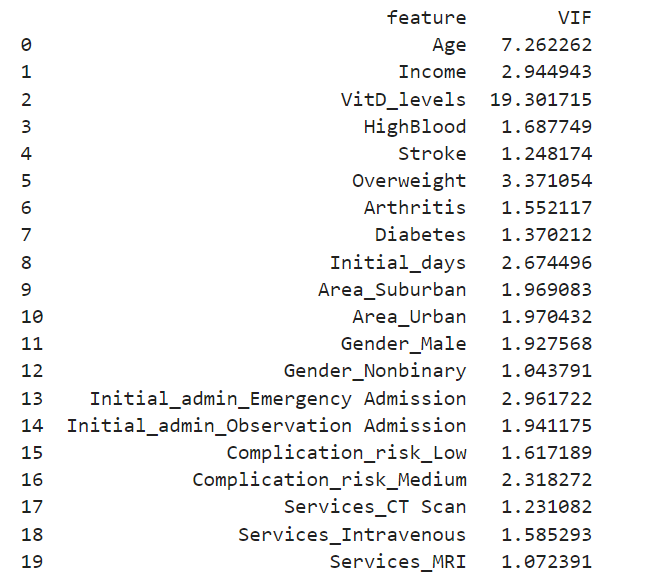
Backward stepwise elimination is a regression technique that starts with a model that includes all possible predictors and removes variables one by one until a stopping criterion is reached (*Backward Stepwise Regression*, 1976).

Variance Inflation Factor (VIF) will be calculated for each feature. High VIF indicates multicollinearity. If a variable has a VIF above 10, I will remove it from the model (Michael Parker, 2019). This simplifies the model and reduces redundancy. Next, I will look at the p-values for each remaining feature. Low p-values below 0.05 mean the feature is statistically significant. If a variable has a high p-value, I will remove it to keep only the most relevant variables.

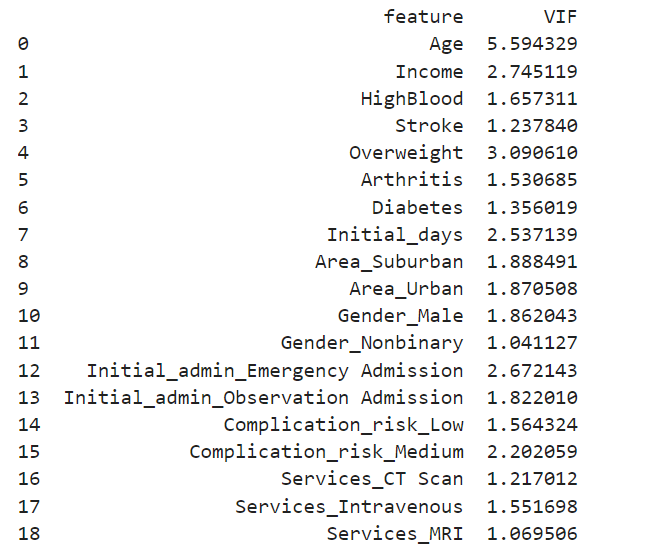
The variance\_inflation\_factor function is used to compute the VIF for each feature in X. VIF measures the degree of multicollinearity (correlation) between independent variables.

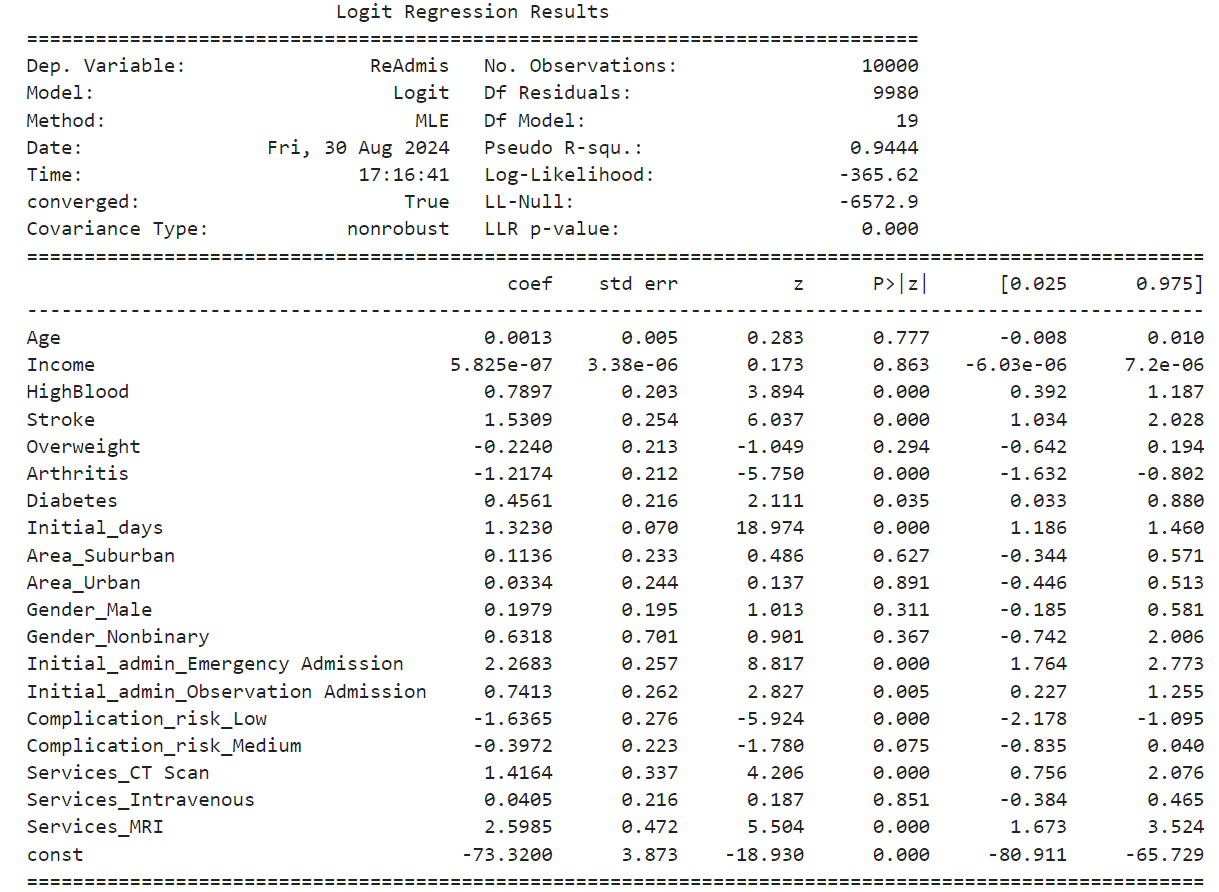


**Backward elimination #1:** According to the results, **Additional\_charges** has the highest **VIF of 78.114**, which will be eliminated from the dataset.

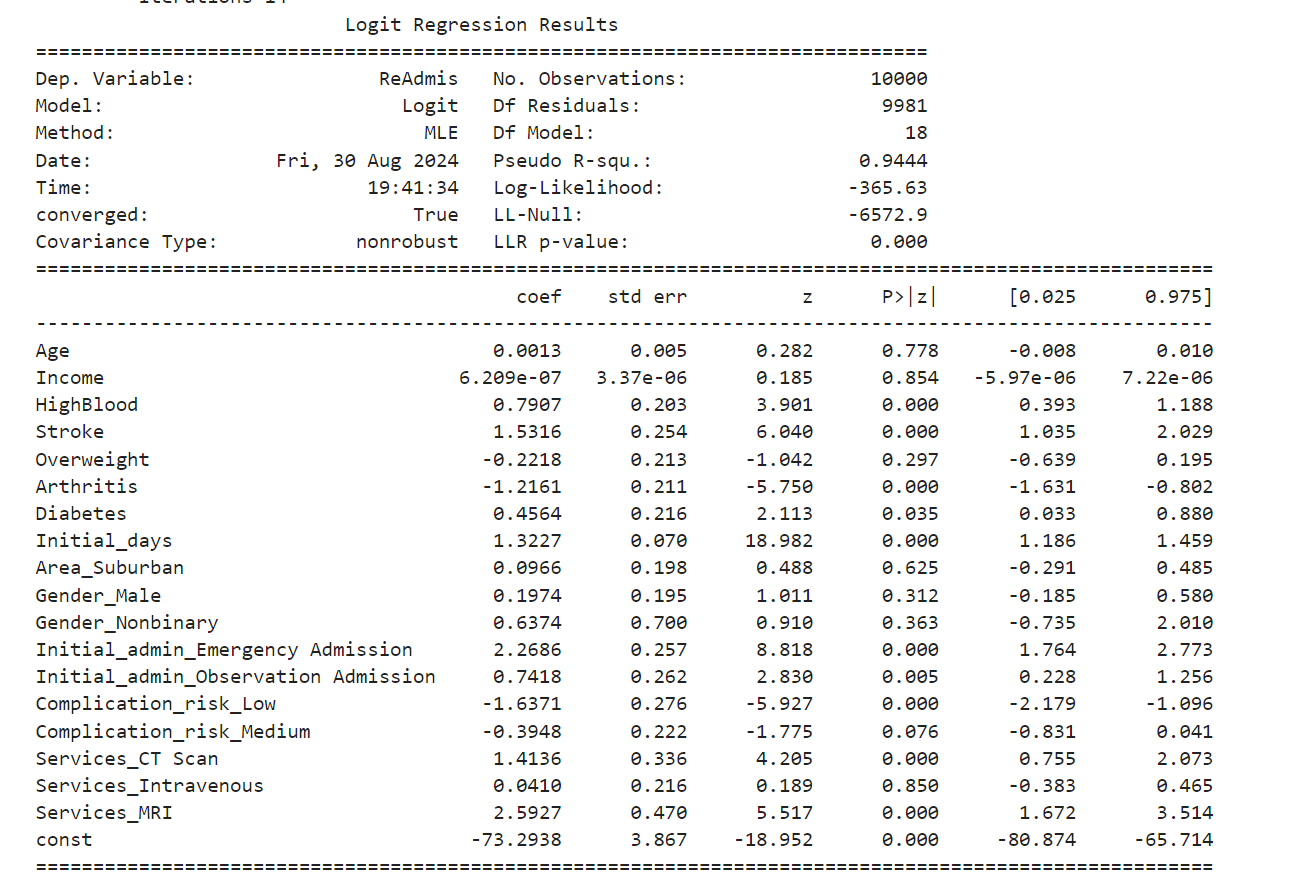


**Backward elimination #2:** **VitD\_levels** have the highest **VIF of 19.302**, so it will be eliminated from the dataset.

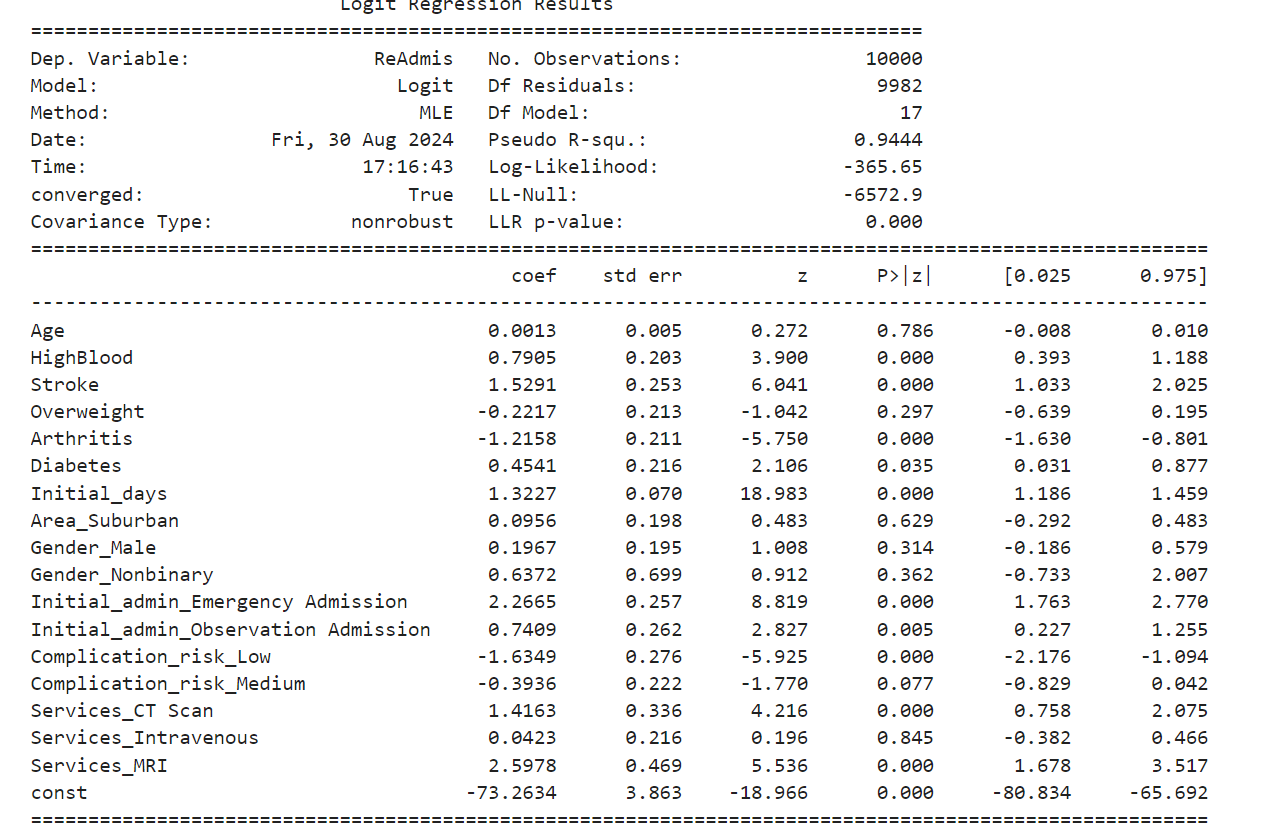
  
All variables have VIF lower than 10, so the next step is eliminating p-values higher than alpha (0.05) until all variables have p-values equal to or less than 0.05.



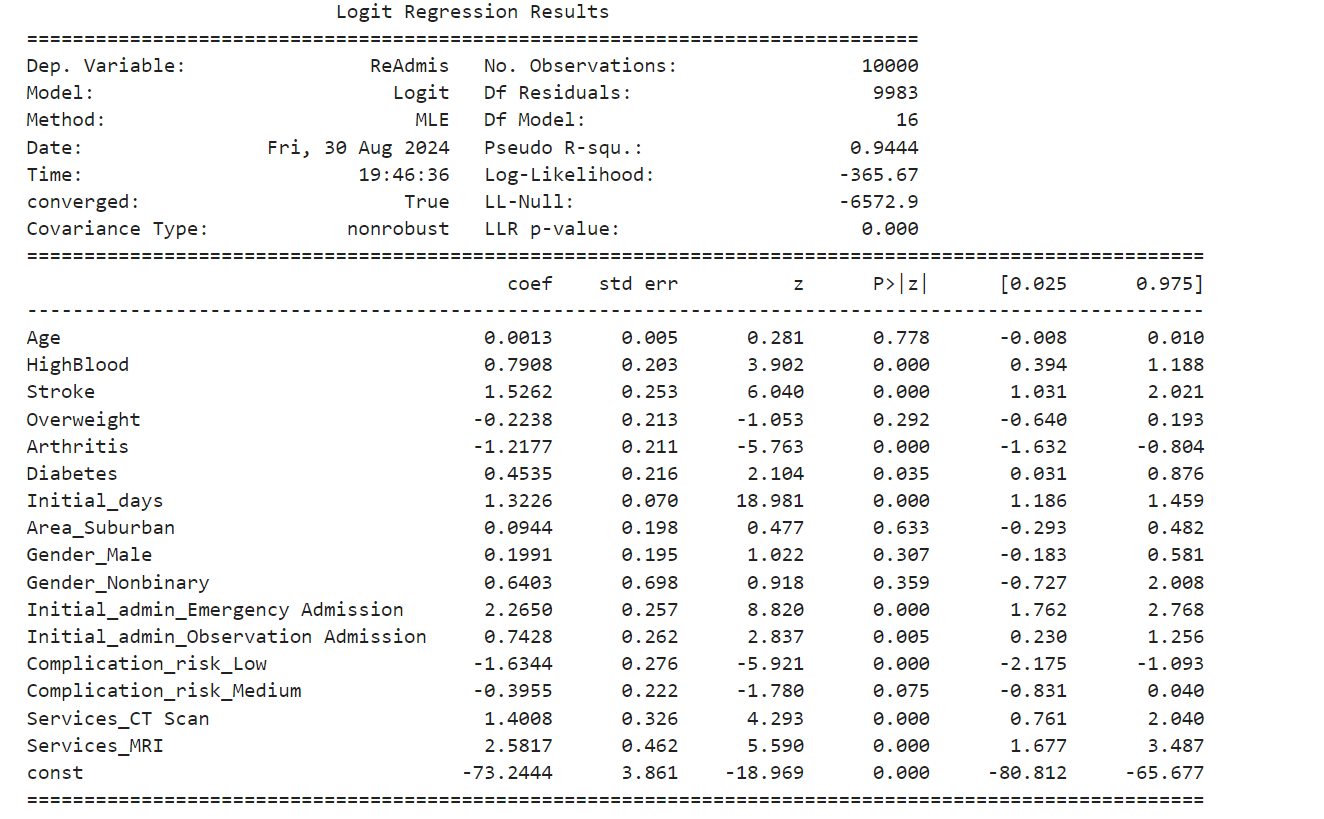
**Backward elimination #3**: **Area\_Urban** will be eliminated due to the highest **p-value of 0.891**.



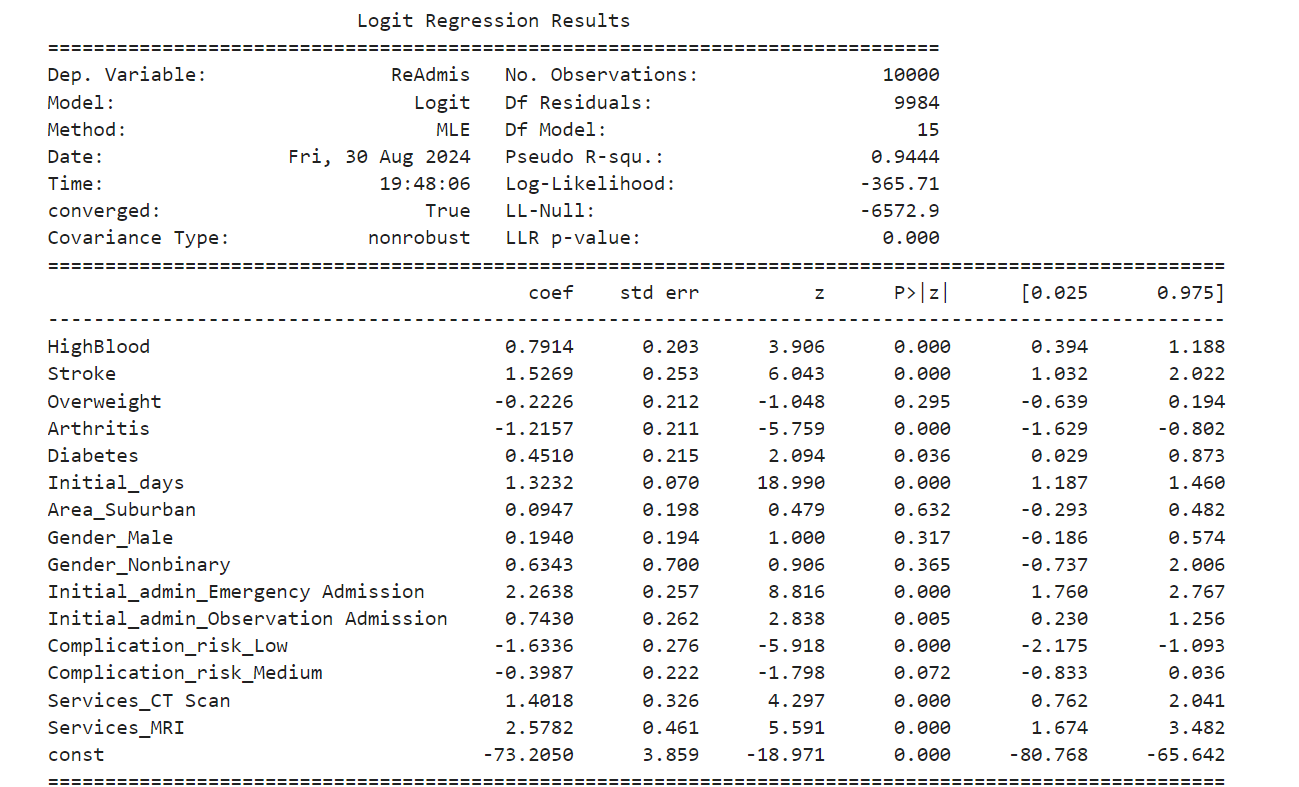
**Backward elimination #4**: **Income** due to the highest **p-value of 0.854**



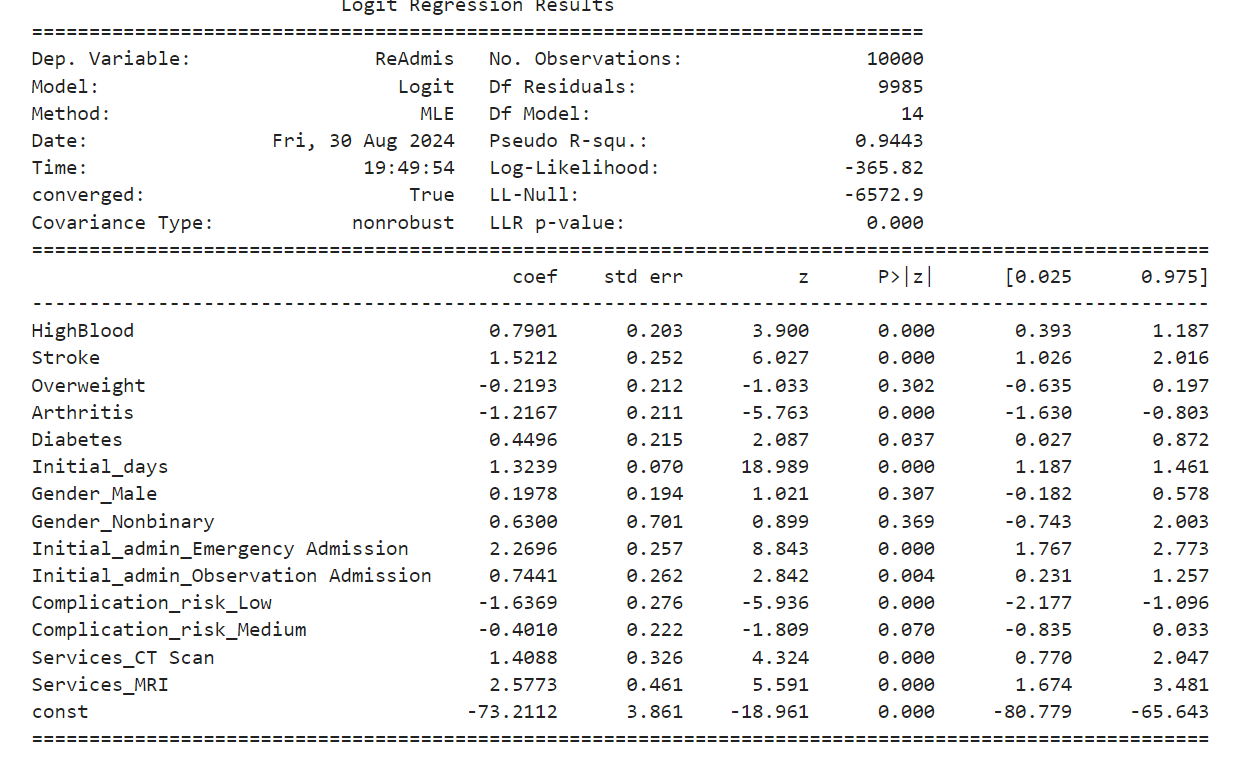
**Backward elimination #5: Services\_intravenous** due to the highest **p-value of 0.845**



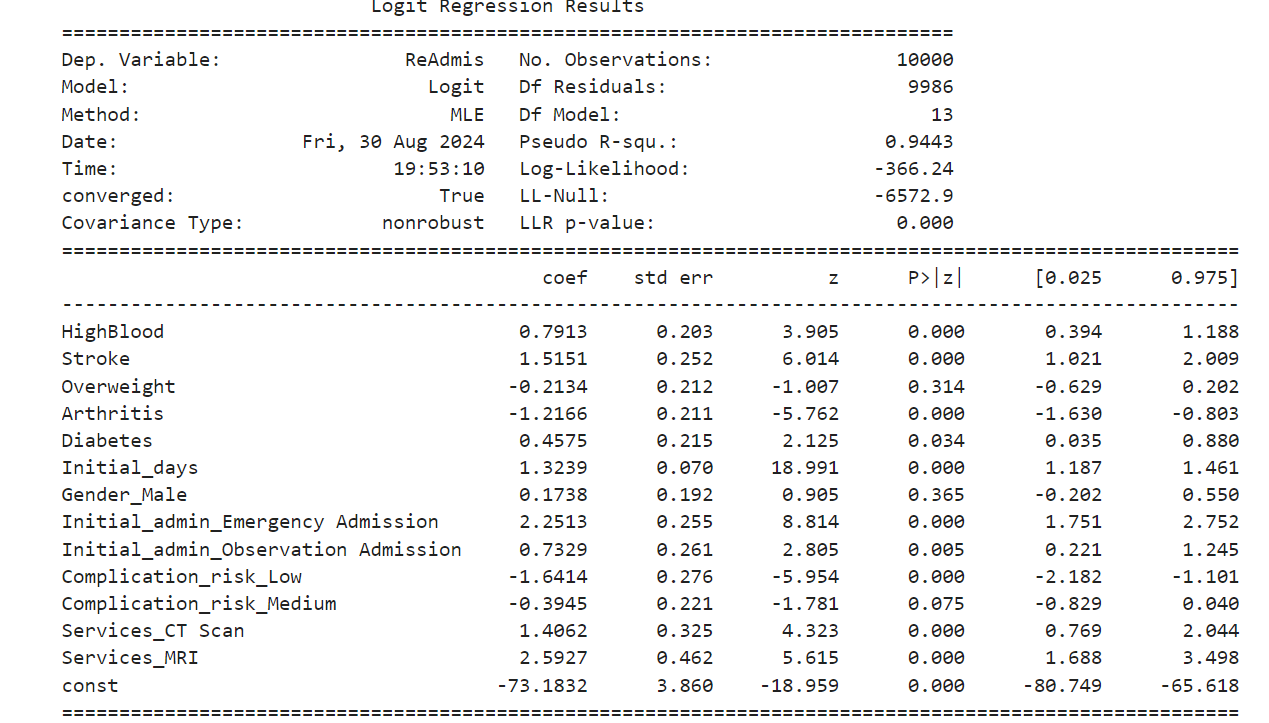
**Backward elimination #6: Age** due to the highest **p-value of 0.778**



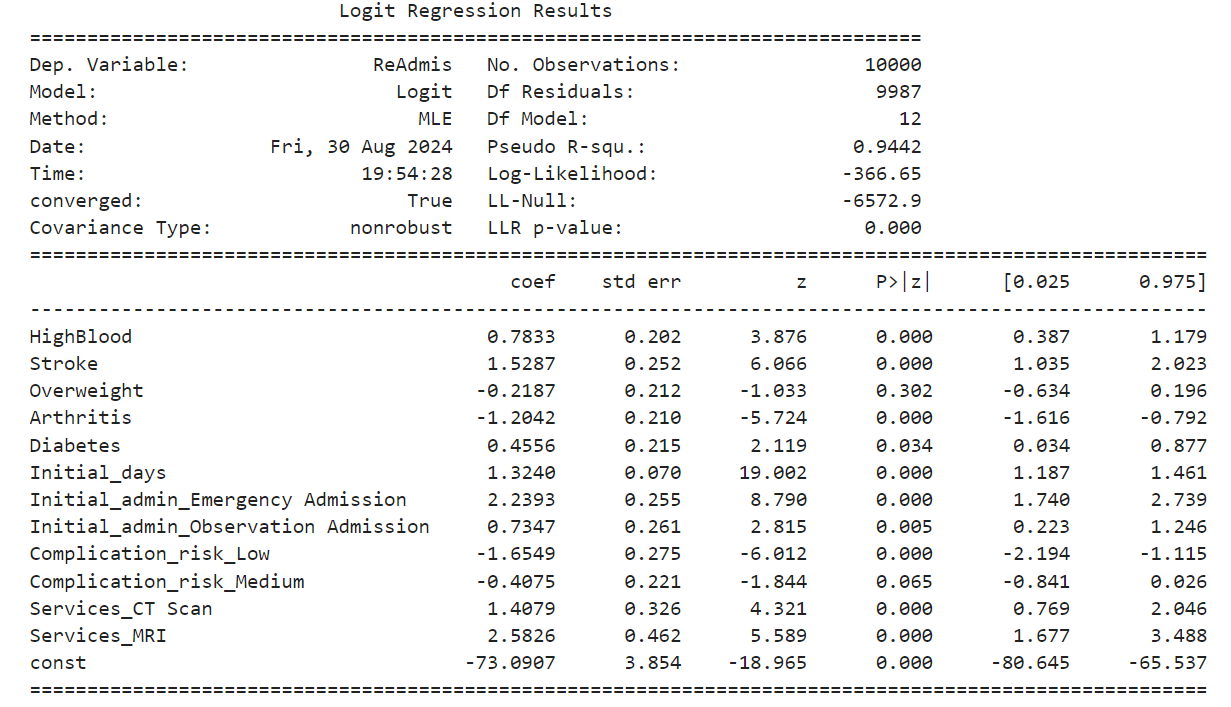
**Backward elimination #7: Area\_Suburban** due to the highest **p-value of 0.632**



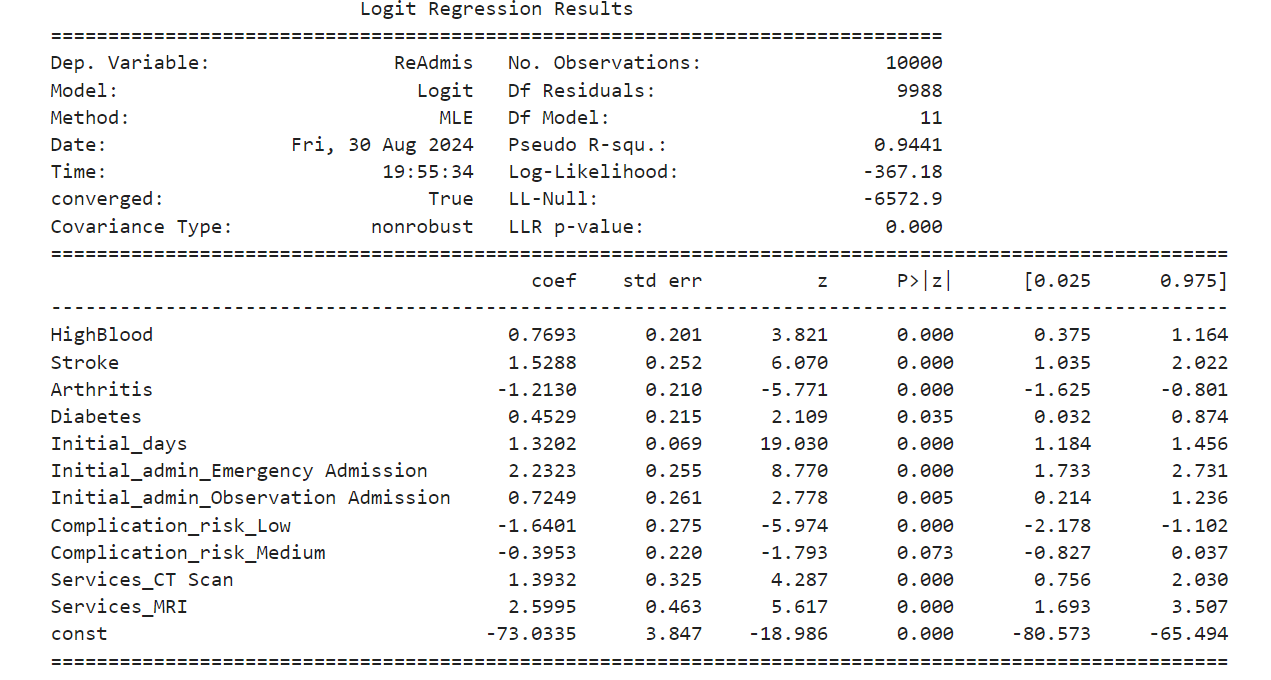
**Backward elimination #8: Gender\_Nonbinary** due to the highest **p-value of 0.369**



**Backward elimination #9: Gender\_Male** due to the highest **p-value of 0.365**



**Backward elimination #10: Overweight** due to the highest **p-value of 0.302**



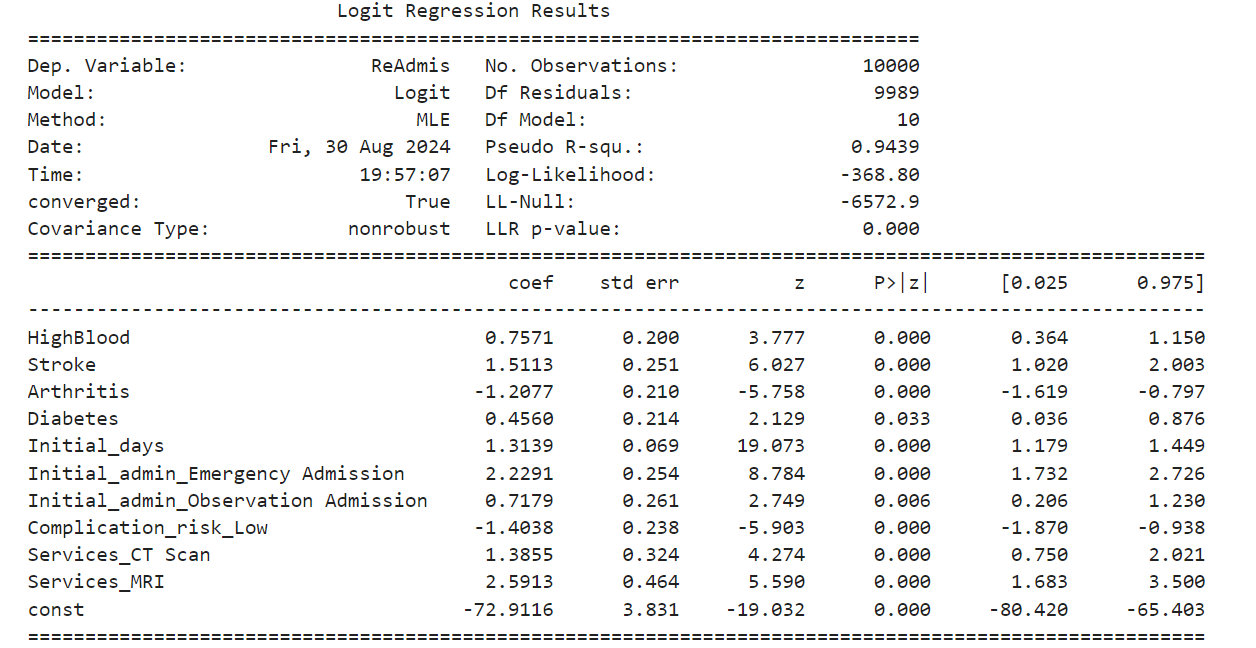
**Backward elimination #11: Complication\_risk\_Medium** due to the highest **p-value of 0.073**

***D3. Reduced Model***

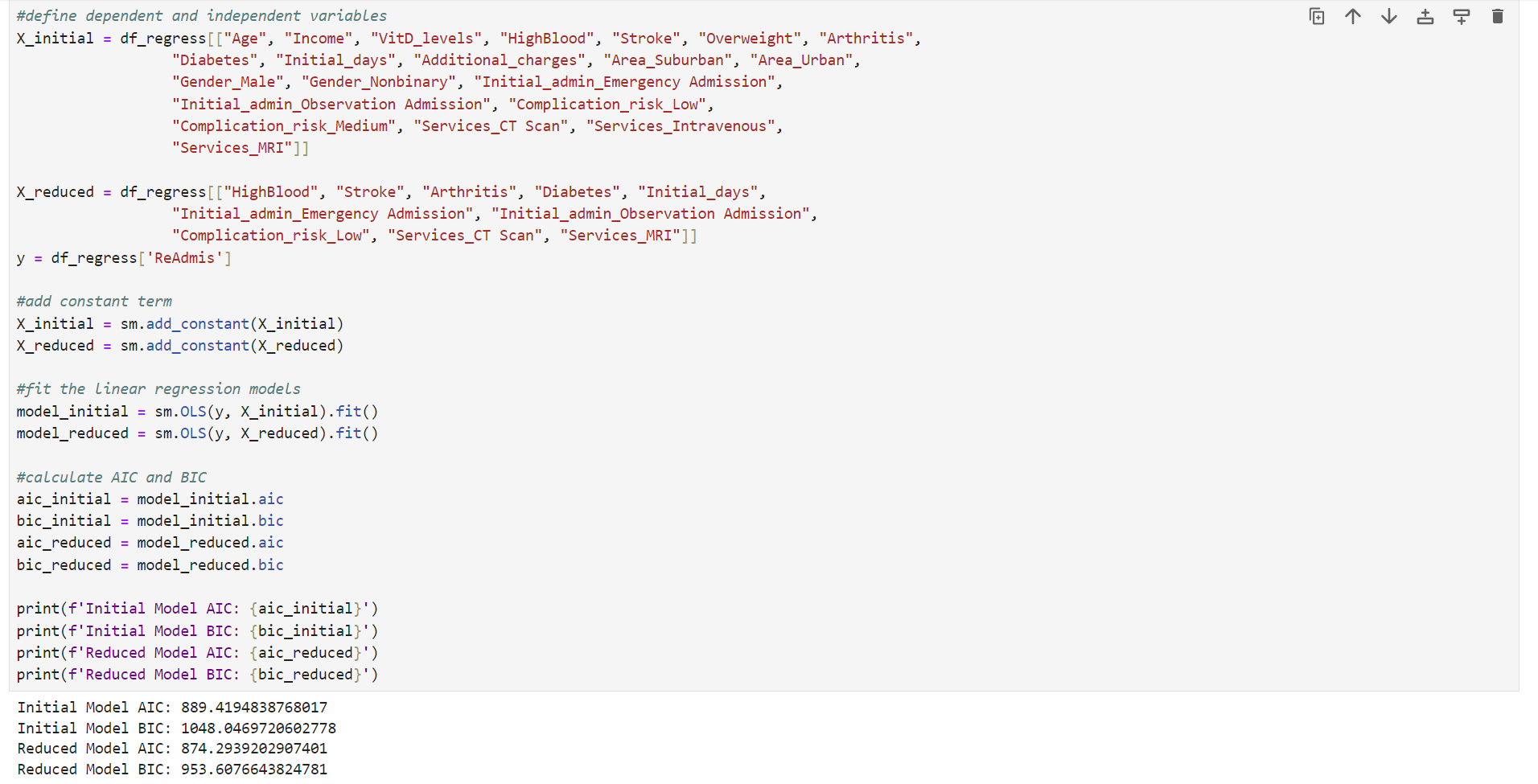
After Backward elimination #11, the last variable, Complication\_risk\_Medium, was eliminated from the model, and there were no variables with p-values higher than 0.05.

Below is a reduced model.





Python doesn’t provide AIC and BIC results for the model, so they were calculated for both models using sm.add\_constant() and sm.OLS() (Bobbitt, 2021a).



***E1. Model Comparison***

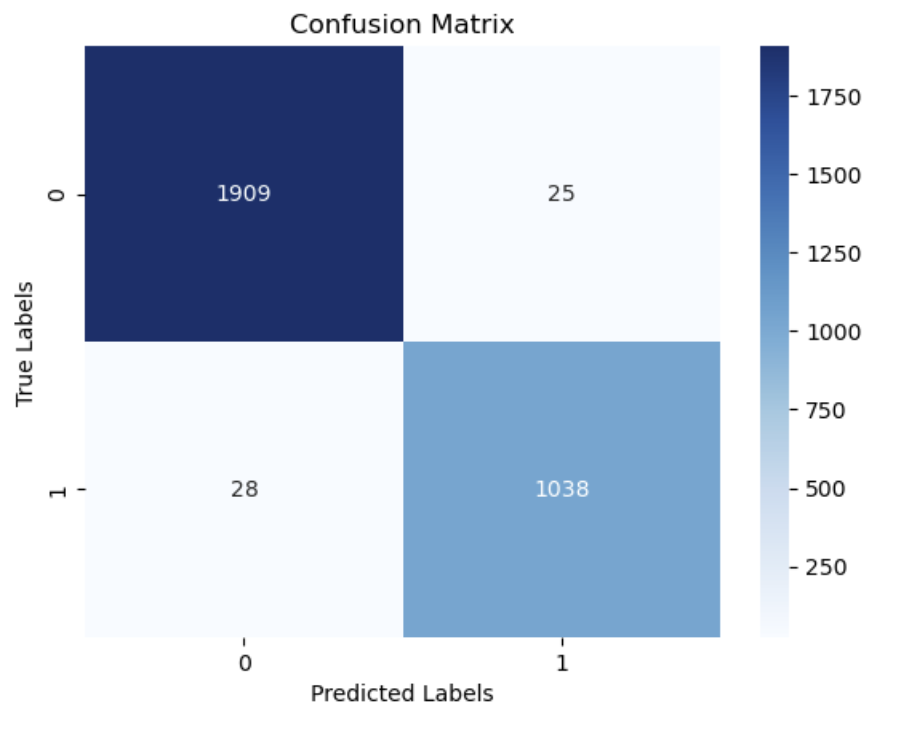
* The initial model includes 21 predictors, while the reduced model includes only 10 (HighBlood, Stroke, Arthritis, Diabetes, Initial\_days, Initial\_admin\_Emergency Admission, Initial\_admin\_Observation Admission, Complication\_risk\_Low, Services\_CT Scan, Services\_MRI). The reduced model focuses on the most significant predictors. HighBlood was not significant (p=0.621), but in the reduced model, it becomes significant (p=0.000).
* Both models have a high Pseudo R-squared value, indicating a good fit. The initial model has a slightly higher Pseudo R-squared (0.9445) than the reduced model (0.9439).
* The initial model has a slightly higher Log-likelihood (-364.84) than the reduced model (-368.80), indicating a marginally better fit.
* LL-Null for both models is -6572.9, providing a baseline.
* The LLR p-value for both models is 0.000, indicating they fit significantly better than the null model.
* The reduced model has a lower AIC (874.29) than the initial model (889.42), suggesting it might be a better fit.
* The reduced model has a lower BIC (953.61) than the initial model (1048.05), reinforcing that the reduced model is preferable.

***E2. Confusion matrix and accuracy calculation***

A confusion matrix is a table used to evaluate the performance of a classification model, like logistic regression. It compares the actual outcomes with the model’s predictions and is organized into four categories: True Positives (correctly predicted positives), True Negatives (correctly predicted negatives), False Positives (incorrectly predicted positives), and False Negatives (incorrectly predicted negatives) (*Www.geeksforgeeks.org*, 2024).

Splitting your data into training and testing sets is crucial for evaluating your logistic regression model’s performance. The training set is used to train the model, allowing it to learn patterns from the data. The testing set, which the model hasn’t seen before, is used to evaluate how well the model generalizes to new, unseen data. This helps ensure that the model isn’t just memorizing the training data (overfitting) but can also make accurate predictions on new data (Prabhakaran, n.d.).





The accuracy of the logistic regression classifier on the test set is 0.98, meaning the model correctly predicted the outcome for 98% of the test cases. The confusion matrix shows that there are 1038 true positives, which are the positive cases correctly predicted by the model, 25 false positives, which are the negative cases incorrectly predicted as positive, 28 false negatives, which are the positive cases incorrectly predicted as negative, and 1909 true negatives, which are the negative cases correctly predicted by the model.

***E3. Code***

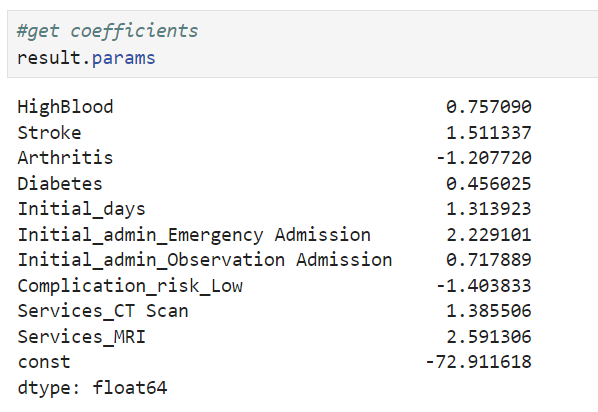
Please see the attached D208Task2.ipynb file.

***Part V: Data Summary and Implications***

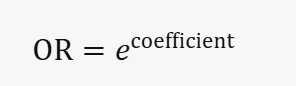
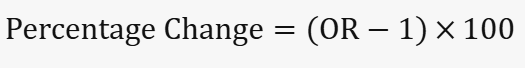
***F1. Regression equation, Coefficients, etc***

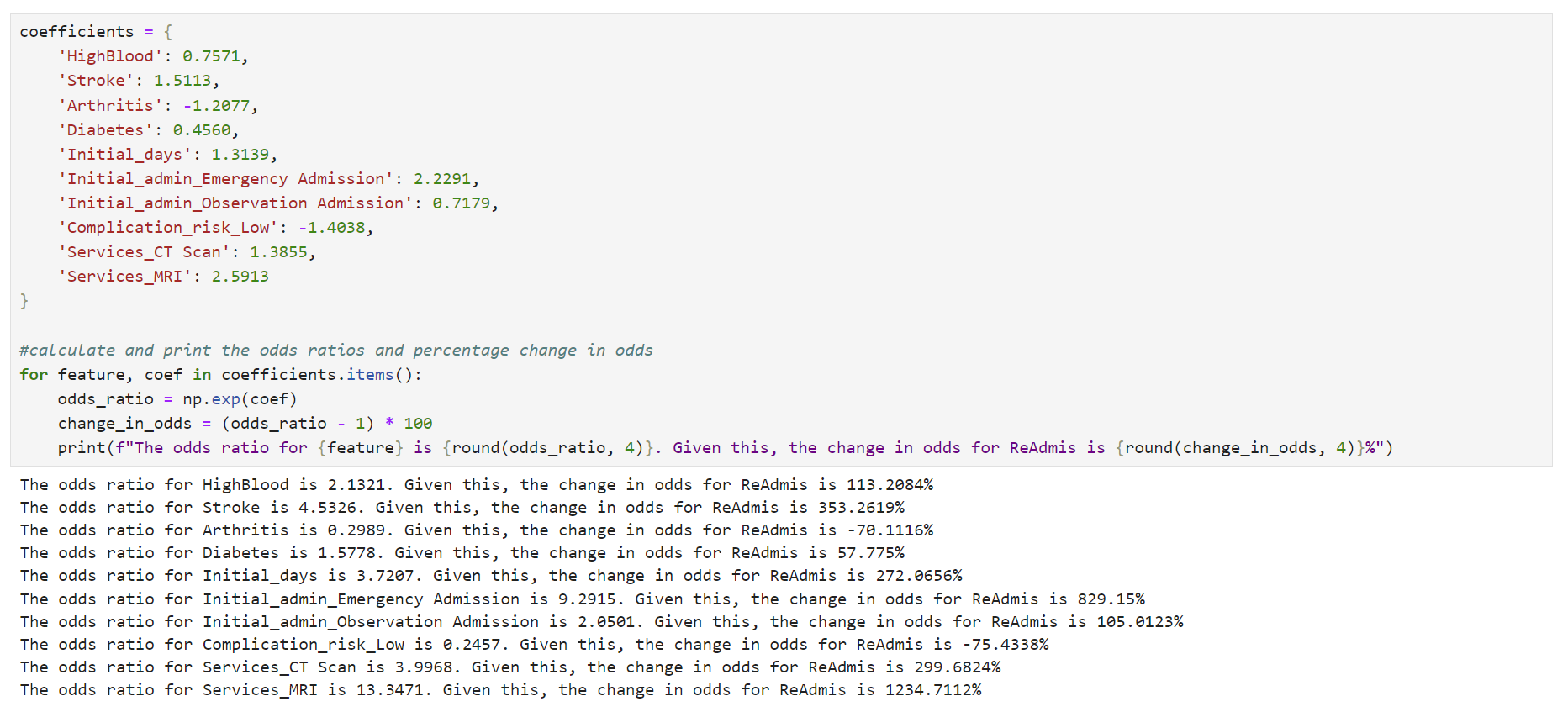
***a) Regression equation***

***b) Interpretation of the coefficients of the reduced model***

******

The coefficients in the logistic regression model represent the change in the log odds of readmission for a one-unit increase in each predictor variable, holding all other variables constant (Bobbitt, 2023). By exponentiating these coefficients, we’ll get an odd ratio (OR), and we can interpret them as changes in the odds of readmission.

******

* HighBlood (0.757090): The OR of HighBlood is 2.1321, indicating that for patients with high blood pressure, the odds of readmission increase by approximately 113.21%.
* Stroke (1.511337): The OR of Stroke is 4.5326, meaning that for patients with a history of stroke, the odds of readmission increase by about 353.26%.
* Arthritis (-1.207720): The OR of Arthritis is 0.2989, suggesting that patients with arthritis have a 70.11% lower chance of readmission compared to those without arthritis.
* Diabetes (0.456025): The OR of Diabetes is 1.5778, indicating a 57.78% increase in the odds of readmission.
* Initial\_days (1.313923): The OR of Initial\_days is 3.7207, meaning that for each additional day of the initial hospital stay, the odds of readmission increase by approximately 272.07%.
* Initial\_admin\_Emergency Admission (2.229101): The OR of Initial\_admin\_Emergency Admission is 9.2915, showing that for patients with emergency admission, the odds of readmission are about 829.15% higher compared to other types.
* Initial\_admin\_Observation Admission (0.717889): The OR of Initial\_admin\_Observation Admission is 2.0501, indicating that for patients with this type of admission, the odds of readmission are about 105.01% higher than other types.
* Complication\_risk\_Low (-1.403833): The OR of Complication\_risk\_Low is 0.2457, suggesting that for patients with low complication risk, the odds of readmission decrease by approximately 75.43%.
* Services\_CT Scan (1.385506): The OR of Services\_CT Scan is 3.9968, meaning that for patients who had a CT scan, the odds of readmission are about 299.68% higher than those who did not.
* Services\_MRI (2.591306): The OR of Services\_MRI is 13.3471, indicating that the odds of readmission for patients who had an MRI are about 1234.71% higher than those who did not.

***c) Discussion regarding the statistical significance and practical significance of the model***

The reduced model is statistically significant. The LLR p-value for the reduced model is 0.000, indicating that the model with predictors fits significantly better than the null model (intercept-only model). All of the predictors in the reduced model have p-values less than 0.05, indicating they are statistically significant.

Practical significance evaluates the real-world impact of a model’s predictors. The logistic regression model for predicting hospital readmission (ReAdmis) shows several significant predictors. HighBlood, Stroke, Diabetes, and Initial\_days are positively associated with the likelihood of readmission, meaning patients with these conditions or longer initial stays are more likely to be readmitted. Conversely, Arthritis and Complication\_risk\_Low are negatively associated, indicating these factors reduce the likelihood of readmission. The model’s high pseudo R-squared value (0.9439) suggests it explains a substantial portion of the variability in readmission outcomes, making it a robust tool for identifying high-risk patients and potentially guiding interventions to reduce readmission rates.

***d) Disadvantages of the methods used to conduct the regression model***

The methods used in this regression analysis have several potential disadvantages and implications, focusing on data preparation, manipulation, and model reduction.

* Retaining outliers can distort the relationships between variables, leading to biased estimates and reduced model accuracy. Outliers can also violate the assumption of normally distributed residuals, which is crucial for regression analysis.
* Techniques like one-hot encoding increase the dataset's dimensionality, potentially leading to overfitting, especially with small sample sizes. This can also introduce multicollinearity, affecting the stability and interpretability of the regression coefficients.
* Using the Variance Inflation Factor (VIF) to detect multicollinearity is useful, but removing variables with high VIF might exclude important predictors. This can result in a model that lacks interpretability and practical relevance.
* Removing variables based solely on p-values can exclude predictors with practical significance, even if they are not statistically significant. This approach can oversimplify the model, failing to capture the data's complexity and leading to instability where small data changes result in different selected variables.

***F2. Recommendations***

Based on the regression analysis and findings, several recommended actions can be suggested for the organization. The reduced model demonstrates statistical and practical significance, with its 98% accuracy on the test set highlighting its practical utility. The model is also significant with high odds ratios for conditions like high blood pressure, stroke, and diabetes, indicating substantial impacts on readmission likelihood. Recommended actions include focusing on high-risk patients with conditions like high blood pressure, stroke, and diabetes by implementing targeted interventions and personalized care plans. Optimizing initial hospital stays through better discharge planning and transitional care programs can help reduce readmissions. Enhancing triage and follow-up protocols for emergency and observation admissions, addressing complication risks through risk assessment tools, and ensuring appropriate use and follow-up of diagnostic services like CT scans and MRIs are also crucial. Continuous monitoring and evaluation of these interventions using data analytics will help track readmission rates and identify areas for improvement. However, attention must be paid to data quality, feature selection, and balancing model complexity to ensure robust and reliable results. By implementing these actions, the organization can improve patient outcomes, reduce readmission rates, and enhance overall healthcare quality.

***Part VI: Demonstration***

***G. Panopto video***

Please see the attached link to the Panopto video.

***Sources***

***H. Web sources of third-party code***

Bobbitt, Z. (2021a, May 20). How to calculate AIC of regression models in Python. Statology. https://www.statology.org/aic-in-python/

GeeksforGeeks. (2024, March 21). *One hot encoding in machine learning*. GeeksforGeeks. https://www.geeksforgeeks.org/ml-one-hot-encoding/#

Sharma, N. (2024, March 17). *How to do One Hot Encoding? Transform Your Categorical Data!* Analytics Vidhya. https://www.analyticsvidhya.com/blog/2023/12/how-to-do-one-hot-encoding/

*Pandas get one hot encodings from a column as booleans*. (n.d.). Stack Overflow. <https://stackoverflow.com/questions/56758567/pandas-get-one-hot-encodings-from-a-column-as-booleans>

*Obtaining AIC and BIC for model selection- logistic regression*. (2023, December 7). https://communities.sas.com/t5/Statistical-Procedures/Obtaining-AIC-and-BIC-for-model-selection-logistic-regression/td-p/905889

*Www.geeksforgeeks.org*. (2024, July 24). Retrieved August 29, 2024, from https://www.geeksforgeeks.org/confusion-matrix-machine-learning/

***I. Acknowledge sources, using in-text citations and references, for content that is quoted, paraphrased, or summarized***

*Backward stepwise regression*. (1976). [Report]. https://www.analystsoft.com/en/products/statplus/content/help/pdf/analysis\_regression\_backward\_stepwise\_elimination\_regression\_model.pdf

BasuMallick, C. (2022b, December 6). *The 12 vital differences between R and Python*. Spiceworks Inc. https://www.spiceworks.com/tech/devops/articles/r-vs-python/

Bobbitt, Z. (2023, April 17). *How to interpret logistic regression coefficients (With example)*. Statology. https://www.statology.org/interpret-logistic-regression-coefficients/

Bobbitt, Z. (2020, October 13). *The 6 Assumptions of Logistic Regression (With Examples)*. Statology. https://www.statology.org/assumptions-of-logistic-regression/

Michael Parker. (2019, April 26). *Multiple Linear Regression with High VIF and P-Values* [Video]. YouTube. https://www.youtube.com/watch?v=gxqtU5tgoos

Prabhakaran, S. (n.d.). *Train Test split – How to split data into train and test for validating machine learning models?* www.machinelearningplus.com. Retrieved September 1, 2024, from https://www.machinelearningplus.com/machine-learning/train-test-split/

*Redirecting*. (n.d.-m). https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2FDataScienceTeam%2FShared%20Documents%2FGraduate%20Team%2FD208%2FStudent%20Facing%20Resources%2FDr%2E%20Middleton%20Getting%20Started%20with%20D208%28Part%20I%29COIT%2Epdf&parent=%2Fsites%2FDataScienceTeam%2FShared%20Documents%2FGraduate%20Team%2FD208%2FStudent%20Facing%20Resources

*Redirecting*. (n.d.-n). https://westerngovernorsuniversity.sharepoint.com/sites/DataScienceTeam/Shared%20Documents/Forms/AllItems.aspx?id=%2Fsites%2FDataScienceTeam%2FShared%20Documents%2FGraduate%20Team%2FD208%2FStudent%20Facing%20Resources%2FDr%2E%20Middleton%20Getting%20Started%20with%20D208%28Part%20I%29COIT%2Epdf&parent=%2Fsites%2FDataScienceTeam%2FShared%20Documents%2FGraduate%20Team%2FD208%2FStudent%20Facing%20Resources