**Addressing Global Health Challenges Through Machine Learning**

**Final Project – DSE6111**

**December 8, 2023**

**Adeline Casali**

# ***Executive Summary***

*Introduction*

This report discusses the application of machine learning to address pressing global health challenges, focusing on heart disease, HIV infection rates, and AIDS death rates. In today’s world, where data-driven approaches are transforming healthcare globally, integrating advanced analytics and predictive modeling is becoming increasingly paramount for understanding, predicting, and mitigating health risks on a global scale. This report's primary objective is to harness machine learning's power to analyze, predict, and derive actionable insights from health data related to heart disease, HIV, and AIDS. By leveraging diverse predictive models, this report aims to contribute valuable perspectives to the field of global health and offer recommendations for more effective public health strategies. The choice of heart disease, HIV, and AIDS as focal points is deliberate, given their profound impact on global health. Heart disease remains the leading cause of mortality worldwide, taking the lives of approximately 17.9 million people each year, necessitating robust predictive models to identify risk factors and inform prevention measures (WHO, 2023a). Similarly, the complex dynamics of HIV infection rates and the critical issue of AIDS deaths demand sophisticated analytical tools for accurate predictions and informed decision-making. In the subsequent sections, this report will delve into qualitative responses for predicting heart disease, quantitative approaches for forecasting HIV infection rates, and principal components regression for predicting AIDS death rates. Through a comprehensive examination of each model’s performance, insights, and ethical considerations, this report aims to contribute to the ongoing discourse on leveraging machine learning for positive global health outcomes.

*Qualitative Response: Predicting Heart Disease*

Addressing heart disease's significant global health challenge requires a nuanced understanding of its multifaceted impact on diverse populations, underscored by ethical considerations such as responsible and privacy-aware data use. The dataset, compiled from the 2015 Behavioral Risk Factor Surveillance System survey, serves as the foundation for predicting heart disease. Utilizing a diverse set of machine learning algorithms, including Logistic Regression, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Naïve Bayes, K-Nearest Neighbors (KNN), Classification Trees, Bagging, and Random Forest, this report aims to provide a comprehensive analysis.

Models such as Logistic Regression, LDA, and Classification Trees exhibit exceptional accuracy, ranging from 87.75% to 98.13%. Key predictors, such as Stroke, Sex, and High Cholesterol, emerge as crucial determinants of heart disease risk. The Logistic Regression model, striking a balance between accuracy and interpretability, stands out as a preferred choice for practical applicability. Insights derived from the models offer nuanced perspectives on heart disease prediction, while ethical considerations highlight the importance of responsible data use. Practical implications include targeted interventions for identified key predictors, emphasizing the significance of prevention strategies tailored to specific risk factors. In conclusion, this report contributes to the discourse on leveraging machine learning for positive global health outcomes, aiming to guide healthcare strategies in addressing the complex challenges posed by heart disease on a global scale.

*Quantitative Problem: Predicting HIV Infection Rates*

The quantitative exploration of predicting HIV infection rates stands as a pivotal component in addressing the complex dynamics of global public health. HIV remains a substantial global challenge, with an estimated 85.6 million people being infected with the HIV virus since the beginning of the epidemic (WHO, 2023b). Understanding and predicting the virus’s transmission dynamics is paramount for designing effective prevention and intervention initiatives. This report delves into the intricate task of forecasting HIV transmission through an array of machine learning models, each offering unique insights into the complex relationship between demographic factors and disease prevalence. A diverse set of machine learning models was meticulously applied to a comprehensive dataset sourced from UNICEF, encompassing various demographic, behavioral, and regional indicators across different countries and years. The models employed include Best Subset Selection, Forward and Backward Selection, Ridge Regression, Lasso, Partial Least Squares (PLS), Regression Tree, Bagging, Random Forest, and Boosting. Each model was carefully selected for its unique strengths, ranging from interpretability to predictive accuracy.

The Best Subset Selection model, identified as having a low Mean Squared Error (MSE), stood out for its balance between accuracy and simplicity. Key predictors, such as fertility rate and AIDS death rate, emerged consistently across various models, emphasizing the critical role of demographic dynamics in shaping HIV transmission. While models like Bagging demonstrated remarkable predictive accuracy with an extremely low MSE, the trade-off between complexity and interpretability was evident. This report underscores the importance of not solely relying on predictive performance metrics but also considering the selected models' practical implications, interpretability, and global applicability. Overall, this analysis serves as a valuable contribution to the discussion on machine learning applications for improving public health outcomes. It highlights the distinctive performance of each model, offering insights into their strengths and downfalls. These findings emphasize the complexity of predicting HIV infection rates and the importance of a balanced approach in guiding global health strategies.

*Principal Components Regression: Predicting AIDS Death*

The Principal Components Regression (PCR) model, utilized to predict AIDS death rates within the same dataset as the quantitative HIV infection rates models, showcased robust performance metrics. With a Mean Squared Error (MSE) of 52.88973 and an explanatory power of 84.87%, the model efficiently harnessed the benefits of PCR in addressing multicollinearity and reducing dimensionality. Key predictors, including HIV infection rate and life expectancy, provided valuable insights into AIDS-related mortality dynamics, affirming the established association between higher HIV infection rates and increased AIDS death rates. Moreover, the model highlighted the anticipated impact of AIDS prevalence on life expectancy. An unexpected finding revealed that, contrary to prevailing research, an increase in youth literacy rates was associated with higher AIDS death rates. This finding suggests caution in interpretation due to a potential anomaly in the model. Recognizing such anomalies as part of a comprehensive evaluation contributes to ongoing refinement. This ensures the PCR model’s utility in predicting AIDS death rates is optimized for broader applicability across diverse scenarios and populations.

***Data and Approach***

*Qualitative Response: Predicting Heart Disease*

Heart disease is a significant global health challenge that accounts for a significant portion of mortality across diverse populations. Before delving into predictive modeling, it is imperative to grasp the multifaceted impact of heart disease on societies worldwide. Ethical considerations play a pivotal role, necessitating the responsible and privacy-aware use of health data. The sensitivity of personal health information emphasizes the importance of ethical guidelines, ensuring that analyses and predictions are conducted with unconditional respect for individuals’ privacy and well-being.

The dataset utilized in predicting heart disease encompasses a comprehensive compilation of health indicators from diverse regions across the United States. The data was collected as part of the 2015 Behavioral Risk Factor Surveillance System (BRFSS) telephone survey conducted annually by the Centers for Disease Control and Prevention (CDC). This dataset contains 253,680 survey responses, of which 229,787 respondents do not have or have not had heart disease, while 23,893 have had heart disease. There are 22 variables included in the dataset, explained in the dictionary below. These variables are selected based on their known associations with heart disease and are crucial inputs for the predictive models. The dataset comprises a diverse distribution of observations representing various demographics, geographical locations, and health contexts. This diversity ensures that the predictive models can generalize well to different populations and contexts.

* HeartDiseaseorAttack: Indicates if the person has experienced heart disease or a heart attack.
* HighBP: Indicates if the person has been told by a health professional that they have high blood pressure.
* HighChol: Indicates if the person has been told by a health professional that they have high blood cholesterol.
* CholCheck: Whether or not the person has had their cholesterol levels checked within the last 5 years.
* BMI: Body Mass Index, calculated by dividing the person’s weight in kilograms by the square of their height in meters.
* Smoker: Indicates if the person has smoked at least 100 cigarettes.
* Stroke: Indicates if the person has a history of stroke.
* Diabetes: Indicates if the person has a history of diabetes, is currently in pre-diabetes, or currently suffers from either type of diabetes.
* PhysActivity: Indicates if the person has some form of physical activity in their day-to-day routine.
* Fruits: Indicates if the person consumes 1 or more fruits daily.
* Veggies: Indicates if the person consumes 1 or more vegetables daily.
* HvyAlcoholConsump: Indicates if the person has more than 14 drinks per week.
* AnyHealthcare: Indicates if the person has any form of health insurance.
* NoDocbcCost: Indicates if the person wanted to visit a doctor within the past 1 year but couldn’t, due to cost.
* GenHlth: Indicates the person’s response to how well is their general health, ranging from 1 (excellent) to 5 (poor).
* Menthlth: Indicates the number of days, within the past 30 days, that the person had bad mental health.
* PhysHlth: Indicates the number of days, within the past 30 days, that the person had bad physical health.
* DiffWalk: Indicates if the person has difficulty while walking or climbing stairs.
* Sex: Indicates the person's gender, where 0 is female and 1 is male.
* Age: Indicates the age class of the person, where 1 is 18 to 24 years up to 13, which is 80 years or older. Each interval between has a 5-year increment.
* Education: Indicates the highest year of school completed, with 0 being never attended or kindergarten only and 6 being having attended 4 years of college or more.
* Income: Indicates the total household income, ranging from 1 (at least $10,000) to 6 ($75,000+).

Data was cleaned and processed to ensure the quality and reliability of predictive models. Several steps were taken to prepare the data for analysis:

1. Checking for missing values: No missing values were found in the dataset.
2. Categorical variable encoding: Categorical variables were encoded into numerical format and factored when applicable to enable seamless inclusion in all models.
3. Outlier detection: No outliers were detected in the dataset.
4. Checking for correlated variables: A correlation matrix and heatmap were created to check for correlated variables. The only variables with a correlation greater than the absolute value of 0.5 were PhysHlth and GenHlth, with a correlation of 0.52.
5. Creating training and test datasets: The dataset was randomly split with 70% of observations divided into a training dataset and the remainder into a test dataset.

The predictive modeling approach used encompasses a diverse set of machine learning algorithms, each bringing unique strengths and offering nuanced insights into heart disease prediction. Outlined below are each of the models utilized in this analysis.

* Logistic Regression: As a baseline model, logistic regression provides an easily interpretable understanding of the relationship between predictor variables and the likelihood of heart disease. It also displays the significance of each predictor and their associated p-values.
* Linear Discriminant Analysis (LDA): LDA focuses on separating classes to enhance predictive accuracy while assuming a multivariate normal distribution of predictors. It is effective for high-dimensional data and reduces dimensionality while preserving class separability.
* Quadratic Discriminant Analysis (QDA): QDA offers increased flexibility in capturing class boundaries by relaxing the assumption of equal covariance matrices. This method accommodates non-linear relationships, which is suitable when covariance structures differ between classes.
* Naïve Bayes: Naïve Bayes leverages probabilistic principles to classify instances while assuming independence among predictor variables. It is a simple yet effective and computationally efficient approach that works well with high-dimensional data.
* K-Nearest Neighbors (KNN): KNN classifies instances based on the majority class among their k-nearest neighbors, accommodating non-linear relationships. It is intuitive, flexible, adapts well to complex decision boundaries, and excels with high-volume datasets.
* Classification Trees (with Pruning): Classification Trees reveal complex interactions among variables with a clear visual representation of decision rules. It easily captures non-linear relationships and, with pruning, optimizes model simplicity and predictive accuracy.
* Bagging: Bagging constructs an ensemble of models to enhance predictive accuracy and generalizability. This method handles variance well while reducing overfitting and improving model stability.
* Random Forest: The Random Forest approach extends bagging by constructing multiple decision trees while being robust against overfitting and improving predictive performance.

In the following sections, this report will discuss the detailed findings of each model on the dataset, examining their unique considerations, feature importance, and interpretability. This comprehensive exploration aims to identify the best model for predicting heart disease, considering accuracy, practical implications, and real-world utility.

*Quantitative Problem: Predicting HIV Infection Rates*

Predicting HIV infection rates presents a complex challenge with profound implications for global public health. Understanding and predicting the dynamics of HIV transmission are crucial for designing effective prevention and intervention strategies. However, before delving into predictive modeling it is essential to grasp the intricate nature of HIV infection rates, considering both the demographic and behavioral factors that contribute to the spread of the virus. Ethical considerations remain paramount in handling sensitive health data related to HIV, necessitating responsible and privacy-aware data practices to ensure the well-being and confidentiality of individuals involved.

The dataset employed in predicting HIV infection rates comprises a comprehensive compilation of relevant indicators sourced from UNICEF, a reputable health organization. The data came from UNICEF’s “data warehouse” and, once cleaned and switched from long to wide format, contains 411 observations from various countries worldwide across various years (from 2016-2021). HIV rates per 1000 people contained in the dataset ranged from as low as 0.01 to as high as 17.61 with a median value of 0.27. There are a total of 12 variables in the dataset, explained in the dictionary below. This dataset encompasses a diverse range of variables, including demographic information, behavioral factors, and regional characteristics. This diversity is instrumental in fostering the adaptability of predictive models across varied demographic profiles, geographic settings, and health scenarios.

* country: Country or geographic area where the data came from.
* year: Year the data was collected.
* infant\_mortality\_rate: Probability of dying between birth and exactly 1 year of age, expressed per 1000 live births.
* fertility\_rate: Average number of live births a hypothetical cohort of women would have at the end of their reproductive period if they were subject during their whole lives to the fertility rates of a given period and if they were not subject to mortality.
* life\_exp: Number of years newborn children would live if subject to the mortality risks prevailing for the cross-section of population at the time of their birth.
* pop\_growth\_rate: Average exponential rate of growth of the population over one year.
* urban\_pop: Urban population as a percentage of the total population.
* youth\_literacy\_rate: Number of people age 15 to 24 years who can both read and write with understanding a short simple statement on their everyday life, divided by the population in that age group.
* aids\_death\_rate: Number of AIDS-related deaths per 100,000 people.
* hiv\_infection\_rate: Annual number of new HIV infections per 1000 uninfected population.
* mother\_child\_hiv\_transmis\_rate: Estimated number of children aged 0-4 newly infected with HIV from mother-to-child transmission for every 100 women living with HIV delivering in the past 12 months.
* per\_child\_under\_poverty: Children living in households with income below the national poverty line as a percent of all children.

Data was cleaned and processed to ensure the quality and reliability of predictive models. Several steps were taken to prepare the data for analysis:

1. Pivot from long to wide format: To make the dataset a better fit for modeling.
2. Set column names: Renamed columns to be shorter, concise, and easier to work with.
3. Interpolate missing values: Some predictors only contained values for one year (out of five). In those cases, missing values were interpolated for years based on other years for that country.
4. Numerical variable encoding: Numerical variables were encoded into numerical format when applicable to enable seamless inclusion in all models.
5. Check for missing values: All missing values were omitted.
6. Bootstrapping to increase sample size: After all data manipulation, there were only 411 observations. Bootstrapping was used to increase the sample size to 1233 observations.
7. Outlier detection: No outliers were detected in the dataset.
8. Checking for correlated variables: A correlation matrix and heatmap were created to check for correlated variables. Collinearity was determined to be present between several variables (infant\_mortality\_rate, fertility\_rate, life\_exp, pop\_growth\_rate, and youth\_literacy\_rate all seem to be correlated).
9. Creating training and test datasets: The dataset was randomly split with 50% of observations divided into a training dataset and the remainder into a test dataset.

The analysis employed a diverse set of machine learning algorithms, each chosen for their unique strengths to provide nuanced insights into predicting HIV rates. The following section details the specific models employed in this analysis, including their individual characteristics and contributions to the overall predictive framework.

* Best Subset Selection: Best Subset Selection is a feature selection method that evaluates all possible combinations of predictor variables to identify the subset that produces the best predictive performance. Its strength lies in providing a comprehensive exploration of all variable combinations, enabling the selection of the most relevant predictors for accurate HIV rate predictions.
* Forward Selection: Forward Selection is a stepwise feature selection technique that iteratively adds predictors to the model based on their contribution to prediction accuracy. This method is computationally efficient and particularly effective when dealing with a large pool of potential predictors.
* Backward Selection: Backward Selection is another stepwise approach that starts with a model containing all predictors and removes them one by one based on their impact on model accuracy. It is advantageous for situations where the initial model includes many variables, helping to streamline the model while retaining predictive power.
* Ridge Regression: Ridge Regression is a regularization technique that addresses multicollinearity by adding a penalty term to the regression equation. It is well-suited for datasets with high-dimensional predictors, preventing overfitting and enhancing the model’s generalizability.
* Lasso: Lasso also introduces a penalty term but with a different regularization technique. Due to this, it addresses multicollinearity and performs variable selection by setting some coefficients to exactly zero. Lasso is beneficial when there is a desire to identify a subset of the most influential predictors.
* Partial Least Squares (PLS): Partial Least Squares is a dimensionality reduction technique that combines features to create a set of new variables, known as latent variables. PLS is advantageous when dealing with datasets with high collinearity and aims to maximize the covariance between predictors and the response variable.
* Regression Tree: Regression Tree is a decision tree-based model that splits the dataset based on predictor variables to predict the response variable. It excels in capturing non-linear relationships and interactions between predictors and provides an excellent visual and interpretable representation of decision rules.
* Bagging: Bagging constructs an ensemble of models by training each on a subset of the dataset through bootstrapping. This technique improves predictive accuracy and generalizability by reducing variance and minimizing overfitting.
* Random Forest: Random Forest extends the concept of bagging by constructing multiple decision trees and introducing randomness in the variable selection process. It addresses overfitting and provides a useful feature importance ranking.
* Boosting: Boosting is an ensemble learning technique that assigns weights to observations, focusing on misclassified instances to improve overall model accuracy. It is effective in situations where individual models may perform poorly but can contribute to collective predictive power.

In the subsequent sections, this report will thoroughly analyze each model’s outcomes on the dataset, exploring their distinctive attributes, the significance of features, and their interpretability. This extensive investigation seeks to determine the most effective model for predicting HIV rates, considering factors such as accuracy, practical implications, and real-world applicability.

*Principal Components Regression: Predicting AIDS Death*

In addition to the quantitative analysis conducted to predict HIV infection rates, another model was produced using Principal Components Regression (PCR) on the same dataset, this time with the target of predicting AIDS death rates. Principal Components Regression is a regression technique that combines principal components, which are linear combinations of the original predictors, to create a predictive model. PCR is beneficial for addressing multicollinearity and reducing the dimensionality of a dataset. The in-depth findings of this model will be discussed comprehensively in the following section.

# ***Detailed Findings***

*Qualitative Response: Predicting Heart Disease*

Exploring heart disease prediction has uncovered nuanced insights by thoroughly examining various models. Each model contributes distinctive perspectives, each offering a comprehensive understanding of their outputs and implications. Additionally, the diverse range of models enhances our ability to discern key factors influencing heart disease prediction, enriching our insights into effective predictive strategies.

* Logistic Regression
  + Model Outputs and Insights:

The logistic regression model with all predictors shows that the variables HighBP, HighChol, CholCheck, Smoker, Stroke, Diabetes, HvyAlcoholConsump, NoDocbcCost, GenHlth, DiffWalk, Sex, Age, and Income were the most significant predictors, followed by PhysActivity, Veggies, and MentHlth. The model created with all the significant predictors has an accuracy of 89.78%, and the model created with only the most significant predictors has the same accuracy of 89.78%. The three predictors with the most impact on the model were determined to be Stroke, where having a stroke increased the likelihood of heart disease, Sex, where being male increased the likelihood of heart disease, and HighChol, where those with high cholesterol were more likely to experience heart disease.

* + Key Predictors and Practical Implications:
    - Stroke: Having a stroke significantly increases the likelihood of heart disease. This underscores the importance of stroke prevention strategies for heart health (O’Donnell et al., 2016).
    - Sex: Being male has been identified as a significant predictor of heart disease risk. Because of this, gender-specific interventions and awareness campaigns may be crucial in mitigating heart disease risk (Suman et al., 2023).
    - High Cholesterol: Individuals with high cholesterol levels are more likely to experience heart disease, emphasizing the need for cholesterol management and monitoring (Stone et al., 2014).
* Linear Discriminant Analysis (LDA)
  + Model Outputs and Insights:

The LDA model containing all significant predictors determined from the logistic regression model has an accuracy of 89.14%. The three predictors with the most impact on the model were determined to be Stroke, where having a stroke increased the likelihood of heart disease, DiffWalk, where having difficulty walking increased the likelihood of heart disease, and Sex, where being male increased the likelihood of heart disease.

* + Key Predictors and Practical Implications:
    - Difficulty Walking: Those experiencing difficulty walking are identified as having an increased likelihood of heart disease, suggesting the importance of mobility in correlation with cardiovascular health (Murtagh et al., 2011).
* Quadratic Discriminant Analysis (QDA)
  + Model Outputs and Insights:

The QDA model containing all significant predictors determined from the logistic regression model has an accuracy of 83.25%. The three predictors with the most impact on the model were determined to be CholCheck, where having your cholesterol checked decreased the likelihood of heart disease, HvyAlcoholConsump, where heavy alcohol consumption increased the risk of heart disease, and Stroke, where having a stroke increased the likelihood of heart disease.

* + Key Predictors and Practical Implications:
    - Cholesterol Check: Regular cholesterol checks have been identified as being associated with a decreased likelihood of heart disease, highlighting the importance of preventative health screenings (Piepoli et al., 2016).
    - Heavy Alcohol Consumption: A link between heavy alcohol consumption and increased heart disease risk has been identified as well, emphasizing the significance of alcohol moderation for heart health (Piano, 2017).
* Naïve Bayes
  + Model Outputs and Insights:

The Naïve Bayes model containing all significant predictors determined from the logistic regression model has an accuracy of 80.39%. The three predictors with the most impact on the model were determined to be Diabetes, Stroke, and HighBP.

* + Key Predictors and Practical Implications:
    - Diabetes: The presence of diabetes has previously been identified as a significant predictor of heart disease, which underlines the importance of managing diabetes for heart disease prevention (Gaede et al., 2003).
    - High Blood Pressure: High blood pressure has also been identified as a key predictor, reinforcing the need for blood pressure control to mitigate heart disease risk (Whelton et al., 2018).
* K-Nearest Neighbors (KNN)
  + Model Outputs and Insights: The KNN model containing all significant predictors determined from the logistic regression model and K=3 has an accuracy of 98.13%. When K = 5, the accuracy is 97.90%, and when K = 10, the accuracy is 97.51%.
  + Key Predictors and Practical Implications:

Unlike many other models, KNN models do not provide information on the most important predictors.

* Classification Trees
  + Model Outputs and Insights: The classification tree containing all significant predictors determined from the logistic regression model has an accuracy of 89.55%. When pruned with cross-validation, the same tree is created with an accuracy of 89.55%. The two predictors utilized in the classification tree are GenHlth and Age (Figure 1).

A diagram of a tree

Description automatically generated

Figure 1: Classification Tree for heart disease

* + Key Predictors and Practical Implications:
    - General Health: General health is a known predictor of heart disease, emphasizing overall well-being's role in prediction and prevention (Xie et al., 2008).
    - Age: Age has previously been established as a crucial factor in predicting heart disease risk, as older people are much more likely to experience it (Benjamin et al., 2019).
* Bagging
  + Model Outputs and Insights:

The bagged classification tree containing all significant predictors determined from the logistic regression model has an accuracy of 87.75%. The most important predictors in the tree are (1) Age, (2) BMI, and (3) Income.

* + Key Predictors and Practical Implications:
    - BMI: Body Mass Index (BMI) has been connected to heart disease risk, highlighting the importance of maintaining a healthy weight for heart health (Ortega et al., 2016).
    - Income: Income has been recognized as an important predictor of heart disease, indicating potential socioeconomic influences of heart disease risk (Stringhini et al., 2017).
* Random Forest
  + Model Outputs and Insights:

The random forest classification tree containing all significant predictors determined from the logistic regression model has an accuracy of 89.45%. The most important predictors in the tree are (1) Age, (2) BMI, and (3) GenHlth.

* + Key Predictors and Practical Implications:

No new predictors.

Overall, the exploration of predictive models for heart disease outlined above has revealed a diverse collection of methodologies, each offering unique insights into the complex interplay of factors contributing to the risk of heart disease. The diverse range of models enhances our ability to discern key factors influencing heart disease prediction, enriching insights into effective predictive strategies. Logistic Regression, Linear Discriminant Analysis (LDA), and Classification Trees all demonstrated exceptional accuracy, showcasing their efficacy in predicting heart disease based on the set of predictors defined by the dataset (Figure 2). Notably, the K-Nearest Neighbors (KNN) model exhibited outstanding accuracy at 98.13% with K = 3, albeit with limited interpretability. However, selecting the best model involves a trade-off between accuracy and interpretability. The logistic regression model emerges as a compelling choice, demonstrating a respectable accuracy of 89.78% and providing valuable insights into predictor significance, making it easily interpretable in context. While KNN achieves higher accuracy, the logistic regression model’s ability to maintain accuracy while considering only the most significant predictors enhances its practical applicability and makes it more suitable for informing decision-making in healthcare.

A graph with numbers and text

Description automatically generated with medium confidence

Figure 2: Accuracy comparison of all models

*Quantitative Problem: Predicting HIV Infection Rates*

This comprehensive exploration investigates the intricate task of predicting HIV infection rates using diverse machine learning algorithms. Each algorithm encapsulates distinctive strengths and contributes insights to the overall predictive framework. Furthermore, this section delves into the key predictors determined by the models, offering real-world, practical implications for forecasting HIV rates globally. This examination serves the overarching goal of identifying the most effective model for predicting HIV infection rates and interpreting its implications for global health strategies.

* Best Subset Selection
  + Model Outputs and Insights:

The model created using best subset selection determined the model with the lowest MSE is one that contains 6 predictors (infant\_mortality\_rate, fertility\_rate, youth\_literacy\_rate, aids\_death\_rate, mother\_child\_hiv\_transmis\_rate, and per\_child\_under\_poverty) with an MSE of 1.116785 and an adjusted R-squared value of 0.778. The two most influential predictors were determined to be fertility rate, with an increase in fertility rate being associated with a decrease in HIV infection rate, and AIDS death rate, with an increase in AIDS death rate being associated with an increase in HIV infection rates.

* + Key Predictors and Practical Implications:
    - Fertility Rate: A connection between fertility rates and HIV infection rates has been established, emphasizing the intricate relationship between demographic factors and disease prevention (Mkwashapi et al., 2023).
    - AIDS Death Rate: Higher AIDS death rates are correlated with an increase in HIV infection rates, highlighting the interconnectedness of disease transmission and mortality (Granich et al., 2015).
* Forward Selection
  + Model Outputs and Insights:

The best model created with forward stepwise selection contains the same 6 predictors as the best subset selection model, with the same MSE of 1.116785 and adjusted R-squared value of 0.778. The model also contained the two most influential predictors of fertility rate and AIDS death rate.

* + Key Predictors and Practical Implications:

No new predictors.

* Backward Selection
  + Model Outputs and Insights:

The best model created with forward stepwise selection contains the same 6 predictors as the best subset selection model, with the same MSE of 1.116785 and adjusted R-squared value of 0.778. The model also contained the two most influential predictors of fertility rate and AIDS death rate.

* + Key Predictors and Practical Implications:

No new predictors.

* Ridge Regression
  + Model Outputs and Insights:

The model created with ridge regression has an MSE of 1.655353. The three most influential predictors were determined to be population growth rate, with an increase in population growth rate being associated with a decrease in HIV infection rates, as well as fertility rate and AIDS death rate.

* + Key Predictors and Practical Implications:
    - Population Growth Rate: A rise in population growth rates has been shown to correspond with a decrease in HIV infection rates, potentially due to the overall decrease in HIV worldwide, once again highlighting the potential influence of demographic dynamics on the prevalence of disease (Rehle & Shisana, 2003).
* Lasso
  + Model Outputs and Insights:

The model created with the lasso method has an MSE of 1.586618. The three most influential predictors were also determined to be population growth rate, fertility rate, and AIDS death rate.

* + Key Predictors and Practical Implications:

No new predictors.

* Partial Least Squares (PLS)
  + Model Outputs and Insights:

The model created with the partial least squares method has an MSE of 1.587266. The three most significant predictors were determined to be the AIDS death rate, youth literacy rate, with an increase in youth literacy rate being associated with a decrease in HIV infection rates, and life expectancy, with an increase in life expectancy being associated with an increase in HIV infection rates.

* + Key Predictors and Practical Implications:
    - Youth Literacy Rate: An increase in youth literacy rate is associated with a decrease in HIV infection rates, emphasizing both the role of education in disease prevention and the role of demographics in disease transmission rates (Preidis et al., 2010).
    - Life Expectancy: Although rises in life expectancy are not directly associated with an increase in HIV infection rates, HIV treatments such as combination antiretroviral therapy (ART) have significantly increased survival rates among HIV positive adults, therefore increasing life expectancies among those populations (Samji et al., 2013).
* Regression Tree
  + Model Outputs and Insights:

The regression tree created with pruning from cross-validation has an MSE of 0.8744749. The three predictors that define the tree splits are AIDS death rate, urban population, with an increase in urban population being associated with a decrease in HIV infection rates, and fertility rate (Figure 3).

A diagram of a tree

Description automatically generated

Figure 3: Regression Tree for HIV infection rate

* + Key Predictors and Practical Implications:
    - Urban Population: Although increased urbanization tends to lead to an increase in HIV infection rates in areas where HIV is already prevalent, HIV rates are currently the highest in rural areas and not urban areas, highlighting the potential impact of urbanization on disease dynamics (Dyson, 2004).
* Bagging
  + Model Outputs and Insights:

The regression tree created with bagging has an MSE of 0.07755276. The three most important predictors were determined to be AIDS death rate, infant mortality rate, and life expectancy.

* + Key Predictors and Practical Implications:
    - Infant Mortality Rate: The significance of infant mortality rates in predicting HIV infection rates underscores the interconnectedness of maternal and child health with disease prevalence (Black et al., 2008).
* Random Forest
  + Model Outputs and Insights:

The regression tree created with random forests has an MSE of 0.118209. The three most important predictors were determined to be AIDS death rate, life expectancy, and mother to child HIV transmission rate.

* + Key Predictors and Practical Implications:
    - Mother to Child HIV Transmission Rate: Mother to child HIV transmission has also been previously identified as a key predictor, emphasizing the intergenerational aspects of HIV transmission (Kourtis et al., 2006).
* Boosting
  + Model Outputs and Insights:

The regression tree created with boosting has an MSE of 0.1268387. The three most important predictors were determined to be AIDS death rate, life expectancy, and youth literacy rate.

* + Key Predictors and Practical Implications:

No new predictors.

Overall, this comprehensive exploration employed diverse machine learning algorithms to predict HIV infection rates, uncovering unique insights into the complex relationship between demographic factors and disease prevalence (Figure 4). Each algorithm encapsulates distinctive strengths, contributing to the overall predictive framework. Notably, the Best Subset, Forward, and Backward Selection models consistently identified fertility rate and AIDS death rate as influential predictors, emphasizing the crucial role of demographic dynamics in shaping HIV transmission. While Bagging demonstrated an impressively low Mean Squared Error (MSE) of 0.07755276, it’s essential to consider not only predictive performance but also interpretability and the practical implications of the selected predictors. The Best Subset Selection model, with its MSE slightly higher at 1.116785 and adjusted R-squared value of 0.778, may have a higher overall MSE compared to Bagging, but it provides a concise set of predictors that have been consistently identified as influential in HIV infection rates in the literature. The Best Subset Selection model balances predictive accuracy and simplicity, making it more interpretable and practical for real-world applications. In contrast, Bagging, being an ensemble method, might involve a larger number of predictors and increased complexity, making it more challenging to translate findings into actionable public health strategies. The choice between these models ultimately depends on the specific goals of the analysis, the importance of interpretability, and the trade-off between model complexity and predictive accuracy. If the primary aim is to identify a concise set of key predictors with practical implications, the Best Subset Selection model (or any of the other Selection models) could be favored, despite its slightly higher MSE.

A black and white text with black text

Description automatically generated

Figure 4: Accuracy comparison of all models

*Principal Components Regression: Predicting AIDS Death*

* Principal Components Regression (PCR)
  + Model Outputs and Insights:

A PCR model on the data has an MSE of 52.88973 and explains 84.87% of the variance. The three most important predictors identified by the model are HIV infection rate, with a higher rate being associated with a higher AIDS death rate, life expectancy, with a higher life expectancy being associated with a lower AIDS death rate, and youth literacy rate, with an increase in youth literacy rates being associated with an increase in AIDS death rate.

* + Key Predictors and Practical Implications:
    - HIV Infection Rate: HIV is the precursor to AIDS, which explains why HIV infection rate is such a significant predictor of AIDS death rate (CDC, 2022).
    - Life Expectancy: Another inherent predictor is life expectancy, as when the death rate from AIDS is high in a geographical area, it is expected to see the life expectancy drop.
    - Youth Literacy Rate: As established in the previous section, an increase in youth literacy rate is associated with a decrease in HIV infection rates (Preidis et al., 2010). However, an increase in youth literacy rates is associated with an increase in AIDS death rate in this model, contrary to current research findings. This may be due to an anomaly in the model.

# ***Validity and Reliability Assessment***

*Qualitative Response: Predicting Heart Disease*

A rigorous assessment was conducted to evaluate the validity and reliability of the models employed in predicting heart disease, considering various aspects of model performance. The models were subjected to comprehensive evaluation metrics, primarily focusing on accuracy as a key indicator of predictive capability. Accuracy, defined as the ratio of correctly predicted instances to the total instances, provides a fundamental measure of the model’s overall correctness. The logistic regression model, incorporating an extensive set of predictors, exhibited a remarkable accuracy of 89.78%. This high accuracy suggests that the model was effective in correctly classifying individuals with or without heart disease based on the selected predictors. Similar accuracy levels were observed in other models, such as Linear Discriminant Analysis (89.14%) and the Classification Tree (89.55%). The K-Nearest Neighbors model, while achieving an outstanding accuracy of 98.13%, presented a unique characteristic as it does not provide insights into the importance of individual predictors.

While accuracy serves as a valuable metric, it is crucial to delve into the stability and reliability of the models. Model stability, the consistency of predictions across different subsets of the dataset, is essential for generalizability. The logistic regression model maintained its high accuracy (89.14%) when working with different subsets of predictors, such as only the most significant predictors, indicating robustness in its predictive ability. However, certain considerations and downfalls should be acknowledged. The Naïve Bayes model, despite achieving an accuracy of 80.39%, exhibited a relatively lower performance when compared to other models. This suggests that the assumption of independence among predictors, a key aspect of the Naïve Bayes algorithm, might not fully align with the underlying relationships in the dataset. Furthermore, the KNN model, while excelling in accuracy, lacks interpretability regarding key predictors. This inherent characteristic makes extracting meaningful insights into the specific factors contributing to heart disease prediction challenging. In summary, assessing validity and reliability emphasizes the robustness of several models, with accuracy as a compelling metric. Despite variations in performance, each model contributes unique insights into the overall picture, highlighting the importance of considering stability, interpretability, and accuracy when addressing the complex task of predicting heart disease.

*Quantitative Problem: Predicting HIV Infection Rates*

In scrutinizing the validity and reliability of the diverse machine learning models utilized for predicting HIV infection rates, a comprehensive evaluation was undertaken to ensure both their validity and reliability across machine learning algorithms. This assessment focused on an array of metrics, prominently featuring accuracy in the form of Mean Squared Error, or MSE, as a key measure of predictive capability. Specifically, the Best Subset, Forward, and Backward Selection models exhibited a shared set of six predictors, showcasing their consistency in capturing predictive patterns. Other predictive models, such as Ridge Regression, Lasso, Partial Least Squares (PLS), Regression Tree, Bagging, Random Forest, and Boosting, each highlighted specific key predictors associated with HIV infection rates, reinforcing the reliability of their findings.

However, in pursuing model reliability and stability, it is imperative to acknowledge certain downfalls and considerations for specific models. For instance, while Ridge Regression emphasized the significance of population growth rates, its relatively higher MSE (1.655353) raises questions about the precision of its predictions. Similarly, the Lasso model, despite revealing influential predictors such as AIDS death rate, a relatively high MSE of 1.586618. The Regression Tree displayed a relatively lower MSE of 0.8744749, but its determination of urban population’s association with a decrease in HIV infection rates demands a nuanced understanding of the impact of urbanization on disease dynamics. Furthermore, Bagging, although presenting the lowest MSE of 0.07755276, necessitates considering its susceptibility to overfitting and the interpretation challenges posed by an ensemble of trees. These critical considerations underscore the complexity of predicting HIV infection rates and the importance of balancing accuracy with reliability and interpretability when selecting a model for meaningful global health strategies. While accuracy stands as a central metric, considerations of stability and model-specific downfalls shed light on the models’ individual performance and guide the interpretation of their implications for forecasting HIV rates globally.

*Principal Components Regression: Predicting AIDS Death*

The Principal Components Regression (PCR) model, employed to predict AIDS death rates, demonstrated notable strengths with a Mean Squared Error (MSE) of 52.88973 and an impressive explanatory power of 84.87%. Key predictors, including HIV infection rate, life expectancy, and youth literacy rate, offered valuable insights into AIDS-related mortality dynamics. The model reaffirmed the established link between higher HIV infection rates and increased AIDS death rates, as well as the inherent association between AIDS prevalence and the impact on life expectancy. However, a surprising finding indicated that, contrary to prevailing research, an increase in youth literacy rates was associated with higher AIDS death rates, prompting caution in the interpretation of this predictor due to a potential anomaly in the model. Acknowledging potential model downfalls, particularly anomalies in predictor relationships, contributes to a comprehensive evaluation of the PCR model’s utility in predicting AIDS death rates, emphasizing the need for continued scrutiny and refinement for broader applicability across diverse scenarios and populations.

# ***Appendix A: References***

Benjamin, E. J., Muntner, P., Alonso, A., Bittencourt, M. S., Callaway, C. W., Carson, A. P., ... & Tsao, C. W. (2019). Heart Disease and Stroke Statistics—2019 Update: A Report From the American Heart Association. *Circulation, 139*(10), e56-e528. <https://doi.org/10.1161/CIR.0000000000000659>

Bhandary, K. (2022, July 19). HIV epidemiology indicators for children. Kaggle. https://www.kaggle.com/datasets/karthikbhandary2/hiv-epidemiology-indicators-for-children/data

Black, R. E., Allen, L. H., Bhutta, Z. A., Caulfield, L. E., de Onis, M., Ezzati, M., Mathers, C., & Rivera, J. (2008). Maternal and Child Undernutrition: Global and Regional Exposures and Health Consequences. *The Lancet, 371*(9608), 243–260. <https://doi.org/10.1016/s0140-6736(07)61690-0>

Centers for Disease Control and Prevention (CDC). (2022, June 30). *About HIV/AIDS*. Centers for Disease Control and Prevention (CDC). https://www.cdc.gov/hiv/basics/whatishiv.html#:~:text=HIV%20(human%20immunodeficiency%20virus)%20is,they%20have%20it%20for%20life.

Dyson, T. (2004). HIV/AIDS and Urbanization. *Population and Development Review*, *29*(3), 427–442. https://doi.org/10.1111/j.1728-4457.2003.00427.x

Gaede, P., Vedel, P., Larsen, N., Jensen, G. V., Parving, H. H., & Pedersen, O. (2003). Multifactorial intervention and cardiovascular disease in patients with type 2 diabetes. *New England Journal of Medicine, 348*(5), 383-393. <https://doi.org/10.1056/NEJMoa021778>

Granich, R., Gupta, S., Hersh, B., Williams, B., Montaner, J., Young, B., & Zuniga, J. M. (2015). Trends in AIDS deaths, new infections and art coverage in the top 30 countries with the highest AIDS mortality burden; 1990–2013. *PLOS ONE*, *10*(7). https://doi.org/10.1371/journal.pone.0131353

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning: With applications in R* (2nd ed.). Springer.

Kourtis, A. P., Lee, F. K., Abrams, E. J., Jamieson, D. J., & Bulterys, M. (2006). Mother-to-child transmission of HIV-1: Timing and implications for prevention. *The Lancet Infectious Diseases*, *6*(11), 726–732. https://doi.org/10.1016/s1473-3099(06)70629-6

Mkwashapi, D., Renju, J., Mahande, M., Changalucha, J., Urassa, M., & Todd, J. (2023). Fertility trends by HIV status in a health and demographic surveillance study in Magu District, Tanzania, 1994–2018. *PLOS ONE*, *18*(2). https://doi.org/10.1371/journal.pone.0281914

Murtagh, E. M., Murphy, M. H., & Boone-Heinonen, J. (2010). Walking: The first steps in cardiovascular disease prevention. *Current Opinion in Cardiology*, *25*(5), 490–496. https://doi.org/10.1097/hco.0b013e32833ce972

O'Donnell, M. J., Chin, S. L., Rangarajan, S., Xavier, D., Liu, L., Zhang, H., ... & Teo, K. (2016). Global and regional effects of potentially modifiable risk factors associated with acute stroke in 32 countries (INTERSTROKE): a case-control study. *The Lancet, 388*(10046), 761-775. <https://doi.org/10.1016/S0140-6736(16)30506-2>

Ortega, F. B., Lavie, C. J., & Blair, S. N. (2016). Obesity and cardiovascular disease. *Circulation Research*, *118*(11), 1752–1770. https://doi.org/10.1161/circresaha.115.306883

Piano, M. R. (2017). Alcohol’s Effects on the Cardiovascular System. *Alcohol Research*, *38*(2), 219–241.

Piepoli, M. F., Hoes, A. W., Agewall, S., Albus, C., Brotons, C., Catapano, A. L., ... & Corrà, U. (2016). 2016 European Guidelines on cardiovascular disease prevention in clinical practice: The Sixth Joint Task Force of the European Society of Cardiology and Other Societies on Cardiovascular Disease Prevention in Clinical Practice (constituted by representatives of 10 societies and by invited experts) Developed with the special contribution of the European Association for Cardiovascular Prevention & Rehabilitation (EACPR). *European Heart Journal, 37*(29), 2315-2381. <https://doi.org/10.1093/eurheartj/ehw106>

Preidis, G. A., Shapiro, C. D., Pierre, I., Dyer, M. J., Kozinetz, C. A., & Grimes, R. M. (2010). An effective, low-cost approach to implementing HIV/AIDS education programs in low literacy populations: An example from rural Haiti. *Journal of Health Care for the Poor and Underserved*, *21*(2), 430–437. https://doi.org/10.1353/hpu.0.0310

Rehle, T. M., & Shisana, O. (2003). Epidemiological and demographic HIV/AIDS projections: South Africa. *African Journal of AIDS Research*, *2*(1), 1–8. https://doi.org/10.2989/16085906.2003.9626554

Samji, H., Cescon, A., Hogg, R. S., Modur, S. P., Althoff, K. N., Buchacz, K., Burchell, A. N., Cohen, M., Gebo, K. A., Gill, M. J., Justice, A., Kirk, G., Klein, M. B., Korthuis, P. T., Martin, J., Napravnik, S., Rourke, S. B., Sterling, T. R., Silverberg, M. J., … Gange, S. J. (2013). Closing the gap: Increases in life expectancy among treated HIV-positive individuals in the United States and Canada. *PLoS ONE*, *8*(12). https://doi.org/10.1371/journal.pone.0081355

Stone, N. J., Robinson, J. G., Lichtenstein, A. H., Bairey Merz, C. N., Blum, C. B., Eckel, R. H., ... & Gordon, D. (2014). 2013 ACC/AHA guideline on the treatment of blood cholesterol to reduce atherosclerotic cardiovascular risk in adults: a report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines. *Journal of the American College of Cardiology, 63*(25 Part B), 2889-2934. <https://doi.org/10.1016/j.jacc.2013.11.002>

Stringhini, S., Carmeli, C., Jokela, M., Avendaño, M., Muennig, P., Guida, F., ... & McCrory, C. (2017). Socioeconomic status and the 25 × 25 risk factors as determinants of premature mortality: a multicohort study and meta-analysis of 1· 7 million men and women. *The Lancet, 389*(10075), 1229-1237. <https://doi.org/10.1016/S0140-6736(16)32380-7>

Suman, S., Pravalika, J., Manjula, P., & Farooq, U. (2023). Gender and CVD- does it really matters? *Current Problems in Cardiology*, *48*(5), 101604. https://doi.org/10.1016/j.cpcardiol.2023.101604

Teboul, A. (2022, March 10). *Heart disease health indicators dataset*. Kaggle. https://www.kaggle.com/datasets/alexteboul/heart-disease-health-indicators-dataset/data

Whelton, P. K., Carey, R. M., Aronow, W. S., Casey, D. E., Collins, K. J., Dennison Himmelfarb, C., ... & Sherif, K. D. (2018). 2017 ACC/AHA/AAPA/ABC/ACPM/AGS/APhA/ASH/ASPC/NMA/PCNA guideline for the prevention, detection, evaluation, and management of high blood pressure in adults: executive summary: a report of the American College of Cardiology/American Heart Association Task Force on Clinical Practice Guidelines. *Hypertension, 71*(6), 1269-1324. <https://doi.org/10.1161/HYP.0000000000000066>

World Health Organization (WHO). (2023a). *Cardiovascular diseases*. World Health Organization (WHO). https://www.who.int/health-topics/cardiovascular-diseases#tab=tab\_1

World Health Organization. (2023b). HIV. The Global Health Observatory. https://www.who.int/data/gho/data/themes/hiv-aids#:~:text=Globally%2C%2039.0%20million%20%5B33.1%E2%80%93,considerably%20between%20countries%20and%20regions.

Xie, J., Wu, E. Q., Zheng, Z.-J., Sullivan, P. W., Zhan, L., & Labarthe, D. R. (2008). Patient-reported health status in coronary heart disease in the United States. *Circulation*, *118*(5), 491–497. https://doi.org/10.1161/circulationaha.107.752006

***Appendix B: Source Code***

Final Project DSE6111

Adeline Casali

2023-12-05

# Section 1: What subset of predictors can be used for preventative health screening for heart disease?

**Loading data and packages**

heart\_df <- read.csv("Data/heart\_disease\_health\_indicators\_BRFSS2015.csv")  
library(corrplot)

## corrplot 0.92 loaded

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(MASS)

##   
## Attaching package: 'MASS'  
##   
## The following object is masked from 'package:dplyr':  
##   
## select

library(e1071)  
library(class)  
library(tree)  
library(randomForest)

## randomForest 4.7-1.1  
## Type rfNews() to see new features/changes/bug fixes.  
##   
## Attaching package: 'randomForest'  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine  
##   
## The following object is masked from 'package:ggplot2':  
##   
## margin

library(gbm)

## Loaded gbm 2.1.8.1

library(kableExtra)

##   
## Attaching package: 'kableExtra'  
##   
## The following object is masked from 'package:dplyr':  
##   
## group\_rows

library(coefplot)

##   
## Attaching package: 'coefplot'  
##   
## The following object is masked from 'package:e1071':  
##   
## extractPath

library(class)

**Data exploration and tidying**  
The only predictors that have a correlation stronger than 0.5 are physical health and general health with a correlation of 0.52.

# View the dataset and summary statistics  
head(heart\_df)

## HeartDiseaseorAttack HighBP HighChol CholCheck BMI Smoker Stroke Diabetes  
## 1 0 1 1 1 40 1 0 0  
## 2 0 0 0 0 25 1 0 0  
## 3 0 1 1 1 28 0 0 0  
## 4 0 1 0 1 27 0 0 0  
## 5 0 1 1 1 24 0 0 0  
## 6 0 1 1 1 25 1 0 0  
## PhysActivity Fruits Veggies HvyAlcoholConsump AnyHealthcare NoDocbcCost  
## 1 0 0 1 0 1 0  
## 2 1 0 0 0 0 1  
## 3 0 1 0 0 1 1  
## 4 1 1 1 0 1 0  
## 5 1 1 1 0 1 0  
## 6 1 1 1 0 1 0  
## GenHlth MentHlth PhysHlth DiffWalk Sex Age Education Income  
## 1 5 18 15 1 0 9 4 3  
## 2 3 0 0 0 0 7 6 1  
## 3 5 30 30 1 0 9 4 8  
## 4 2 0 0 0 0 11 3 6  
## 5 2 3 0 0 0 11 5 4  
## 6 2 0 2 0 1 10 6 8

summary(heart\_df)

## HeartDiseaseorAttack HighBP HighChol CholCheck   
## Min. :0.00000 Min. :0.000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:0.00000 1st Qu.:0.000 1st Qu.:0.0000 1st Qu.:1.0000   
## Median :0.00000 Median :0.000 Median :0.0000 Median :1.0000   
## Mean :0.09419 Mean :0.429 Mean :0.4241 Mean :0.9627   
## 3rd Qu.:0.00000 3rd Qu.:1.000 3rd Qu.:1.0000 3rd Qu.:1.0000   
## Max. :1.00000 Max. :1.000 Max. :1.0000 Max. :1.0000   
## BMI Smoker Stroke Diabetes   
## Min. :12.00 Min. :0.0000 Min. :0.00000 Min. :0.0000   
## 1st Qu.:24.00 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.0000   
## Median :27.00 Median :0.0000 Median :0.00000 Median :0.0000   
## Mean :28.38 Mean :0.4432 Mean :0.04057 Mean :0.2969   
## 3rd Qu.:31.00 3rd Qu.:1.0000 3rd Qu.:0.00000 3rd Qu.:0.0000   
## Max. :98.00 Max. :1.0000 Max. :1.00000 Max. :2.0000   
## PhysActivity Fruits Veggies HvyAlcoholConsump  
## Min. :0.0000 Min. :0.0000 Min. :0.0000 Min. :0.0000   
## 1st Qu.:1.0000 1st Qu.:0.0000 1st Qu.:1.0000 1st Qu.:0.0000   
## Median :1.0000 Median :1.0000 Median :1.0000 Median :0.0000   
## Mean :0.7565 Mean :0.6343 Mean :0.8114 Mean :0.0562   
## 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:1.0000 3rd Qu.:0.0000   
## Max. :1.0000 Max. :1.0000 Max. :1.0000 Max. :1.0000   
## AnyHealthcare NoDocbcCost GenHlth MentHlth   
## Min. :0.0000 Min. :0.00000 Min. :1.000 Min. : 0.000   
## 1st Qu.:1.0000 1st Qu.:0.00000 1st Qu.:2.000 1st Qu.: 0.000   
## Median :1.0000 Median :0.00000 Median :2.000 Median : 0.000   
## Mean :0.9511 Mean :0.08418 Mean :2.511 Mean : 3.185   
## 3rd Qu.:1.0000 3rd Qu.:0.00000 3rd Qu.:3.000 3rd Qu.: 2.000   
## Max. :1.0000 Max. :1.00000 Max. :5.000 Max. :30.000   
## PhysHlth DiffWalk Sex Age   
## Min. : 0.000 Min. :0.0000 Min. :0.0000 Min. : 1.000   
## 1st Qu.: 0.000 1st Qu.:0.0000 1st Qu.:0.0000 1st Qu.: 6.000   
## Median : 0.000 Median :0.0000 Median :0.0000 Median : 8.000   
## Mean : 4.242 Mean :0.1682 Mean :0.4403 Mean : 8.032   
## 3rd Qu.: 3.000 3rd Qu.:0.0000 3rd Qu.:1.0000 3rd Qu.:10.000   
## Max. :30.000 Max. :1.0000 Max. :1.0000 Max. :13.000   
## Education Income   
## Min. :1.00 Min. :1.000   
## 1st Qu.:4.00 1st Qu.:5.000   
## Median :5.00 Median :7.000   
## Mean :5.05 Mean :6.054   
## 3rd Qu.:6.00 3rd Qu.:8.000   
## Max. :6.00 Max. :8.000

str(heart\_df)

## 'data.frame': 253680 obs. of 22 variables:  
## $ HeartDiseaseorAttack: num 0 0 0 0 0 0 0 0 1 0 ...  
## $ HighBP : num 1 0 1 1 1 1 1 1 1 0 ...  
## $ HighChol : num 1 0 1 0 1 1 0 1 1 0 ...  
## $ CholCheck : num 1 0 1 1 1 1 1 1 1 1 ...  
## $ BMI : num 40 25 28 27 24 25 30 25 30 24 ...  
## $ Smoker : num 1 1 0 0 0 1 1 1 1 0 ...  
## $ Stroke : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Diabetes : num 0 0 0 0 0 0 0 0 2 0 ...  
## $ PhysActivity : num 0 1 0 1 1 1 0 1 0 0 ...  
## $ Fruits : num 0 0 1 1 1 1 0 0 1 0 ...  
## $ Veggies : num 1 0 0 1 1 1 0 1 1 1 ...  
## $ HvyAlcoholConsump : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ AnyHealthcare : num 1 0 1 1 1 1 1 1 1 1 ...  
## $ NoDocbcCost : num 0 1 1 0 0 0 0 0 0 0 ...  
## $ GenHlth : num 5 3 5 2 2 2 3 3 5 2 ...  
## $ MentHlth : num 18 0 30 0 3 0 0 0 30 0 ...  
## $ PhysHlth : num 15 0 30 0 0 2 14 0 30 0 ...  
## $ DiffWalk : num 1 0 1 0 0 0 0 1 1 0 ...  
## $ Sex : num 0 0 0 0 0 1 0 0 0 1 ...  
## $ Age : num 9 7 9 11 11 10 9 11 9 8 ...  
## $ Education : num 4 6 4 3 5 6 6 4 5 4 ...  
## $ Income : num 3 1 8 6 4 8 7 4 1 3 ...

dim(heart\_df)

## [1] 253680 22

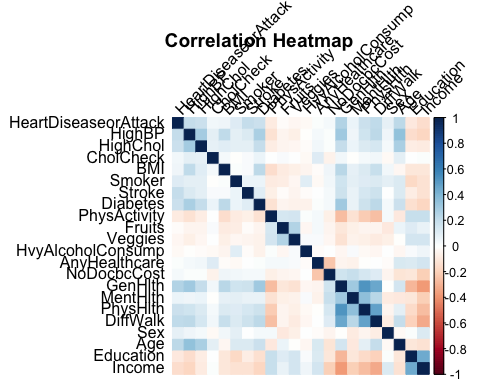
names(heart\_df)

## [1] "HeartDiseaseorAttack" "HighBP" "HighChol"   
## [4] "CholCheck" "BMI" "Smoker"   
## [7] "Stroke" "Diabetes" "PhysActivity"   
## [10] "Fruits" "Veggies" "HvyAlcoholConsump"   
## [13] "AnyHealthcare" "NoDocbcCost" "GenHlth"   
## [16] "MentHlth" "PhysHlth" "DiffWalk"   
## [19] "Sex" "Age" "Education"   
## [22] "Income"

# Check for missing values  
sum(is.na(heart\_df))

## [1] 0

# Check for correlated variables  
cor\_matrix <- cor(heart\_df)  
corrplot(cor\_matrix, method = "color", tl.col = "black", tl.srt = 45)  
title("Correlation Heatmap")



cor\_matrix <- cor(heart\_df)  
cor\_matrix\_filtered <- cor\_matrix  
cor\_matrix\_filtered[abs(cor\_matrix) <= 0.5] <- NA  
print(cor\_matrix\_filtered)

## HeartDiseaseorAttack HighBP HighChol CholCheck BMI Smoker  
## HeartDiseaseorAttack 1 NA NA NA NA NA  
## HighBP NA 1 NA NA NA NA  
## HighChol NA NA 1 NA NA NA  
## CholCheck NA NA NA 1 NA NA  
## BMI NA NA NA NA 1 NA  
## Smoker NA NA NA NA NA 1  
## Stroke NA NA NA NA NA NA  
## Diabetes NA NA NA NA NA NA  
## PhysActivity NA NA NA NA NA NA  
## Fruits NA NA NA NA NA NA  
## Veggies NA NA NA NA NA NA  
## HvyAlcoholConsump NA NA NA NA NA NA  
## AnyHealthcare NA NA NA NA NA NA  
## NoDocbcCost NA NA NA NA NA NA  
## GenHlth NA NA NA NA NA NA  
## MentHlth NA NA NA NA NA NA  
## PhysHlth NA NA NA NA NA NA  
## DiffWalk NA NA NA NA NA NA  
## Sex NA NA NA NA NA NA  
## Age NA NA NA NA NA NA  
## Education NA NA NA NA NA NA  
## Income NA NA NA NA NA NA  
## Stroke Diabetes PhysActivity Fruits Veggies  
## HeartDiseaseorAttack NA NA NA NA NA  
## HighBP NA NA NA NA NA  
## HighChol NA NA NA NA NA  
## CholCheck NA NA NA NA NA  
## BMI NA NA NA NA NA  
## Smoker NA NA NA NA NA  
## Stroke 1 NA NA NA NA  
## Diabetes NA 1 NA NA NA  
## PhysActivity NA NA 1 NA NA  
## Fruits NA NA NA 1 NA  
## Veggies NA NA NA NA 1  
## HvyAlcoholConsump NA NA NA NA NA  
## AnyHealthcare NA NA NA NA NA  
## NoDocbcCost NA NA NA NA NA  
## GenHlth NA NA NA NA NA  
## MentHlth NA NA NA NA NA  
## PhysHlth NA NA NA NA NA  
## DiffWalk NA NA NA NA NA  
## Sex NA NA NA NA NA  
## Age NA NA NA NA NA  
## Education NA NA NA NA NA  
## Income NA NA NA NA NA  
## HvyAlcoholConsump AnyHealthcare NoDocbcCost GenHlth  
## HeartDiseaseorAttack NA NA NA NA  
## HighBP NA NA NA NA  
## HighChol NA NA NA NA  
## CholCheck NA NA NA NA  
## BMI NA NA NA NA  
## Smoker NA NA NA NA  
## Stroke NA NA NA NA  
## Diabetes NA NA NA NA  
## PhysActivity NA NA NA NA  
## Fruits NA NA NA NA  
## Veggies NA NA NA NA  
## HvyAlcoholConsump 1 NA NA NA  
## AnyHealthcare NA 1 NA NA  
## NoDocbcCost NA NA 1 NA  
## GenHlth NA NA NA 1.0000000  
## MentHlth NA NA NA NA  
## PhysHlth NA NA NA 0.5243636  
## DiffWalk NA NA NA NA  
## Sex NA NA NA NA  
## Age NA NA NA NA  
## Education NA NA NA NA  
## Income NA NA NA NA  
## MentHlth PhysHlth DiffWalk Sex Age Education Income  
## HeartDiseaseorAttack NA NA NA NA NA NA NA  
## HighBP NA NA NA NA NA NA NA  
## HighChol NA NA NA NA NA NA NA  
## CholCheck NA NA NA NA NA NA NA  
## BMI NA NA NA NA NA NA NA  
## Smoker NA NA NA NA NA NA NA  
## Stroke NA NA NA NA NA NA NA  
## Diabetes NA NA NA NA NA NA NA  
## PhysActivity NA NA NA NA NA NA NA  
## Fruits NA NA NA NA NA NA NA  
## Veggies NA NA NA NA NA NA NA  
## HvyAlcoholConsump NA NA NA NA NA NA NA  
## AnyHealthcare NA NA NA NA NA NA NA  
## NoDocbcCost NA NA NA NA NA NA NA  
## GenHlth NA 0.5243636 NA NA NA NA NA  
## MentHlth 1 NA NA NA NA NA NA  
## PhysHlth NA 1.0000000 NA NA NA NA NA  
## DiffWalk NA NA 1 NA NA NA NA  
## Sex NA NA NA 1 NA NA NA  
## Age NA NA NA NA 1 NA NA  
## Education NA NA NA NA NA 1 NA  
## Income NA NA NA NA NA NA 1

# Convert HeartDiseaseorAttack to factor  
heart\_df$HeartDiseaseorAttack <- as.factor(heart\_df$HeartDiseaseorAttack)

**Logistic Regression**  
The logistic regression model with all predictors shows that the variables HighBP, HighChol, CholCheck, Smoker, Stroke, Diabetes, HvyAlcoholConsump, NoDocbcCost, GenHlth, DiffWalk, Sex, Age, and Income were the most significant predictors, followed by PhysActivity, Veggies, and MentHlth. The model created with all of the significant predictors has an accuracy of 89.78%, and the model created with only the most significant predictors has the same accuracy of 89.78%.

# Null model  
heart\_lm\_null <- glm(HeartDiseaseorAttack ~ HeartDiseaseorAttack, data = heart\_df, family = binomial)

## Warning in model.matrix.default(mt, mf, contrasts): the response appeared on  
## the right-hand side and was dropped

## Warning in model.matrix.default(mt, mf, contrasts): problem with term 1 in  
## model.matrix: no columns are assigned

summary(heart\_lm\_null)

##   
## Call:  
## glm(formula = HeartDiseaseorAttack ~ HeartDiseaseorAttack, family = binomial,   
## data = heart\_df)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -2.263567 0.006797 -333 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 158355 on 253679 degrees of freedom  
## Residual deviance: 158355 on 253679 degrees of freedom  
## AIC: 158357  
##   
## Number of Fisher Scoring iterations: 5

# Model with all predictors  
heart\_lm <- glm(HeartDiseaseorAttack ~ ., data = heart\_df, family = binomial)  
summary(heart\_lm)

##   
## Call:  
## glm(formula = HeartDiseaseorAttack ~ ., family = binomial, data = heart\_df)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.9124089 0.1028054 -76.965 < 2e-16 \*\*\*  
## HighBP 0.5245081 0.0177520 29.546 < 2e-16 \*\*\*  
## HighChol 0.6112677 0.0164496 37.160 < 2e-16 \*\*\*  
## CholCheck 0.5248111 0.0662510 7.922 2.35e-15 \*\*\*  
## BMI 0.0009744 0.0012122 0.804 0.4215   
## Smoker 0.3629524 0.0157323 23.071 < 2e-16 \*\*\*  
## Stroke 0.9783443 0.0244338 40.041 < 2e-16 \*\*\*  
## Diabetes 0.1465123 0.0089720 16.330 < 2e-16 \*\*\*  
## PhysActivity 0.0398975 0.0171991 2.320 0.0204 \*   
## Fruits 0.0060203 0.0163274 0.369 0.7123   
## Veggies 0.0426327 0.0189359 2.251 0.0244 \*   
## HvyAlcoholConsump -0.2940714 0.0392877 -7.485 7.15e-14 \*\*\*  
## AnyHealthcare -0.0070009 0.0412820 -0.170 0.8653   
## NoDocbcCost 0.2528463 0.0268896 9.403 < 2e-16 \*\*\*  
## GenHlth 0.4907058 0.0095105 51.596 < 2e-16 \*\*\*  
## MentHlth 0.0024628 0.0009778 2.519 0.0118 \*   
## PhysHlth 0.0010542 0.0008766 1.202 0.2292   
## DiffWalk 0.2947780 0.0193855 15.206 < 2e-16 \*\*\*  
## Sex 0.7611811 0.0160326 47.477 < 2e-16 \*\*\*  
## Age 0.2556493 0.0036439 70.158 < 2e-16 \*\*\*  
## Education 0.0110314 0.0081696 1.350 0.1769   
## Income -0.0431551 0.0042476 -10.160 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 158355 on 253679 degrees of freedom  
## Residual deviance: 120933 on 253658 degrees of freedom  
## AIC: 120977  
##   
## Number of Fisher Scoring iterations: 6

# Create a training and testing set  
set.seed(123)  
heart\_train <- heart\_df %>%   
 sample\_frac(0.7)  
heart\_test <- anti\_join(heart\_df, heart\_train)

## Joining with `by = join\_by(HeartDiseaseorAttack, HighBP, HighChol, CholCheck,  
## BMI, Smoker, Stroke, Diabetes, PhysActivity, Fruits, Veggies,  
## HvyAlcoholConsump, AnyHealthcare, NoDocbcCost, GenHlth, MentHlth, PhysHlth,  
## DiffWalk, Sex, Age, Education, Income)`

# Model with significant predictors  
heart\_lm\_sig <- glm(HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck + Smoker + Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost + GenHlth + DiffWalk + Sex + Age + Income + PhysActivity + Veggies + MentHlth, data = heart\_train, family = binomial)  
summary(heart\_lm\_sig)

##   
## Call:  
## glm(formula = HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck +   
## Smoker + Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost +   
## GenHlth + DiffWalk + Sex + Age + Income + PhysActivity +   
## Veggies + MentHlth, family = binomial, data = heart\_train)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.835845 0.101326 -77.333 < 2e-16 \*\*\*  
## HighBP 0.526391 0.020942 25.136 < 2e-16 \*\*\*  
## HighChol 0.623145 0.019618 31.764 < 2e-16 \*\*\*  
## CholCheck 0.543923 0.079393 6.851 7.33e-12 \*\*\*  
## Smoker 0.354107 0.018672 18.965 < 2e-16 \*\*\*  
## Stroke 0.946965 0.029093 32.549 < 2e-16 \*\*\*  
## Diabetes 0.142707 0.010519 13.567 < 2e-16 \*\*\*  
## HvyAlcoholConsump -0.284262 0.046573 -6.104 1.04e-09 \*\*\*  
## NoDocbcCost 0.234995 0.031695 7.414 1.22e-13 \*\*\*  
## GenHlth 0.495146 0.010365 47.773 < 2e-16 \*\*\*  
## DiffWalk 0.317315 0.022180 14.306 < 2e-16 \*\*\*  
## Sex 0.762415 0.019058 40.006 < 2e-16 \*\*\*  
## Age 0.252476 0.004166 60.610 < 2e-16 \*\*\*  
## Income -0.041475 0.004690 -8.843 < 2e-16 \*\*\*  
## PhysActivity 0.038610 0.020277 1.904 0.0569 .   
## Veggies 0.038037 0.021958 1.732 0.0832 .   
## MentHlth 0.003395 0.001132 3.000 0.0027 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 111242 on 177575 degrees of freedom  
## Residual deviance: 84942 on 177559 degrees of freedom  
## AIC: 84976  
##   
## Number of Fisher Scoring iterations: 6

Coefficient <- coef(heart\_lm\_sig)  
coefficients\_df <- as.data.frame(Coefficient)  
sig\_coefficients\_table <- kable(coefficients\_df, format = "markdown")  
print(sig\_coefficients\_table)

##   
##   
## | | Coefficient|  
## |:-----------------|-----------:|  
## |(Intercept) | -7.8358450|  
## |HighBP | 0.5263908|  
## |HighChol | 0.6231446|  
## |CholCheck | 0.5439225|  
## |Smoker | 0.3541067|  
## |Stroke | 0.9469648|  
## |Diabetes | 0.1427075|  
## |HvyAlcoholConsump | -0.2842619|  
## |NoDocbcCost | 0.2349954|  
## |GenHlth | 0.4951461|  
## |DiffWalk | 0.3173146|  
## |Sex | 0.7624155|  
## |Age | 0.2524765|  
## |Income | -0.0414754|  
## |PhysActivity | 0.0386099|  
## |Veggies | 0.0380368|  
## |MentHlth | 0.0033953|

# Test the model and calculate accuracy  
predict\_lm <- predict(heart\_lm\_sig, newdata = heart\_test)  
binary\_predict\_lm <- ifelse(predict\_lm > 0.5, 1, 0)  
results <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,   
 Predicted = binary\_predict\_lm  
)  
results$Correct <- results$Actual == results$Predicted  
confusion\_matrix\_lm <- table(Predicted = results$Predicted, Actual = results$Actual)  
print(confusion\_matrix\_lm)

## Actual  
## Predicted 0 1  
## 0 59853 6623  
## 1 234 391

accuracy\_lm <- (59853 + 391) / (59853 + 6623 + 234 + 391)  
error\_lm <- 1 - accuracy\_lm  
cat("Accuracy:", accuracy\_lm, "\n")

## Accuracy: 0.8978108

cat("Error Rate:", error\_lm, "\n")

## Error Rate: 0.1021892

# Model with only the most significant predictors  
heart\_lm\_sig2 <- glm(HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck + Smoker + Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost + GenHlth + DiffWalk + Sex + Age + Income, data = heart\_train, family = binomial)  
summary(heart\_lm\_sig2)

##   
## Call:  
## glm(formula = HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck +   
## Smoker + Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost +   
## GenHlth + DiffWalk + Sex + Age + Income, family = binomial,   
## data = heart\_train)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -7.758065 0.098684 -78.615 < 2e-16 \*\*\*  
## HighBP 0.524694 0.020937 25.060 < 2e-16 \*\*\*  
## HighChol 0.625225 0.019603 31.894 < 2e-16 \*\*\*  
## CholCheck 0.546769 0.079366 6.889 5.61e-12 \*\*\*  
## Smoker 0.355084 0.018655 19.034 < 2e-16 \*\*\*  
## Stroke 0.947827 0.029083 32.590 < 2e-16 \*\*\*  
## Diabetes 0.141732 0.010513 13.482 < 2e-16 \*\*\*  
## HvyAlcoholConsump -0.280701 0.046548 -6.030 1.64e-09 \*\*\*  
## NoDocbcCost 0.244870 0.031503 7.773 7.67e-15 \*\*\*  
## GenHlth 0.498179 0.010072 49.462 < 2e-16 \*\*\*  
## DiffWalk 0.317142 0.021823 14.532 < 2e-16 \*\*\*  
## Sex 0.758174 0.018949 40.012 < 2e-16 \*\*\*  
## Age 0.250268 0.004088 61.223 < 2e-16 \*\*\*  
## Income -0.041128 0.004639 -8.867 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 111242 on 177575 degrees of freedom  
## Residual deviance: 84958 on 177562 degrees of freedom  
## AIC: 84986  
##   
## Number of Fisher Scoring iterations: 6

coef(heart\_lm\_sig2)

## (Intercept) HighBP HighChol CholCheck   
## -7.75806495 0.52469432 0.62522494 0.54676940   
## Smoker Stroke Diabetes HvyAlcoholConsump   
## 0.35508386 0.94782748 0.14173210 -0.28070129   
## NoDocbcCost GenHlth DiffWalk Sex   
## 0.24487011 0.49817865 0.31714196 0.75817431   
## Age Income   
## 0.25026816 -0.04112835

# Test the second model and calculate accuracy  
predict\_lm2 <- predict(heart\_lm\_sig2, newdata = heart\_test)  
binary\_predict\_lm2 <- ifelse(predict\_lm2 > 0.5, 1, 0)  
results2 <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,   
 Predicted = binary\_predict\_lm2  
)  
results2$Correct <- results2$Actual == results2$Predicted  
confusion\_matrix\_lm2 <- table(Predicted = results2$Predicted, Actual = results2$Actual)  
print(confusion\_matrix\_lm2)

## Actual  
## Predicted 0 1  
## 0 59847 6618  
## 1 240 396

accuracy\_lm2 <- (59847 + 396) / (59847 + 6618 + 240 + 396)  
error\_lm2 <- 1 - accuracy\_lm2  
cat("Accuracy:", accuracy\_lm2, "\n")

## Accuracy: 0.8977959

cat("Error Rate:", error\_lm2, "\n")

## Error Rate: 0.1022041

**LDA**  
The LDA model containing all significant predictors has an accuracy of 89.14%.

# Model with significant predictors  
heart\_lda <- lda(HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck + Smoker + Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost + GenHlth + DiffWalk + Sex + Age + Income + PhysActivity + Veggies + MentHlth, data = heart\_train)  
heart\_lda

## Call:  
## lda(HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck + Smoker +   
## Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost + GenHlth +   
## DiffWalk + Sex + Age + Income + PhysActivity + Veggies +   
## MentHlth, data = heart\_train)  
##   
## Prior probabilities of groups:  
## 0 1   
## 0.90532504 0.09467496   
##   
## Group means:  
## HighBP HighChol CholCheck Smoker Stroke Diabetes  
## 0 0.3949205 0.3955799 0.9595059 0.4254684 0.02793536 0.2554614  
## 1 0.7501784 0.7031882 0.9888175 0.6190221 0.16404949 0.6846895  
## HvyAlcoholConsump NoDocbcCost GenHlth DiffWalk Sex Age Income  
## 0 0.05785499 0.08118733 2.420691 0.1427496 0.4269177 7.813808 6.147091  
## 1 0.03598620 0.10908875 3.372175 0.4182132 0.5739353 10.118011 5.143053  
## PhysActivity Veggies MentHlth  
## 0 0.7689532 0.8162400 3.038952  
## 1 0.6411492 0.7635023 4.732512  
##   
## Coefficients of linear discriminants:  
## LD1  
## HighBP 0.2934865419  
## HighChol 0.3740542652  
## CholCheck 0.1465449354  
## Smoker 0.2194045131  
## Stroke 1.7900323795  
## Diabetes 0.2102071299  
## HvyAlcoholConsump -0.1675486525  
## NoDocbcCost 0.0514054892  
## GenHlth 0.3356617796  
## DiffWalk 0.5344940710  
## Sex 0.5037227569  
## Age 0.1109019681  
## Income -0.0317635706  
## PhysActivity 0.0456651232  
## Veggies 0.0465340124  
## MentHlth 0.0009576617

# Test the model and calculate accuracy  
lda\_predictions <- predict(heart\_lda, newdata = heart\_test)$class  
lda\_results <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,  
 Predicted = lda\_predictions  
)  
lda\_results$Correct <- lda\_results$Actual == lda\_results$Predicted  
lda\_confusion\_matrix <- table(Predicted = lda\_results$Predicted, Actual = lda\_results$Actual)  
print(lda\_confusion\_matrix)

## Actual  
## Predicted 0 1  
## 0 58497 5694  
## 1 1590 1320

accuracy\_lda <- (58497 + 1320) / (58497 + 5694 + 1590 + 1320)  
error\_lda <- 1 - accuracy\_lda  
cat("Accuracy:", accuracy\_lda, "\n")

## Accuracy: 0.8914472

cat("Error Rate:", error\_lda, "\n")

## Error Rate: 0.1085528

**QDA**  
The QDA model containing all significant predictors has an accuracy of 83.25%.

# Model with significant predictors  
heart\_qda <- qda(HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck + Smoker + Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost + GenHlth + DiffWalk + Sex + Age + Income + PhysActivity + Veggies + MentHlth, data = heart\_train)  
heart\_qda

## Call:  
## qda(HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck + Smoker +   
## Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost + GenHlth +   
## DiffWalk + Sex + Age + Income + PhysActivity + Veggies +   
## MentHlth, data = heart\_train)  
##   
## Prior probabilities of groups:  
## 0 1   
## 0.90532504 0.09467496   
##   
## Group means:  
## HighBP HighChol CholCheck Smoker Stroke Diabetes  
## 0 0.3949205 0.3955799 0.9595059 0.4254684 0.02793536 0.2554614  
## 1 0.7501784 0.7031882 0.9888175 0.6190221 0.16404949 0.6846895  
## HvyAlcoholConsump NoDocbcCost GenHlth DiffWalk Sex Age Income  
## 0 0.05785499 0.08118733 2.420691 0.1427496 0.4269177 7.813808 6.147091  
## 1 0.03598620 0.10908875 3.372175 0.4182132 0.5739353 10.118011 5.143053  
## PhysActivity Veggies MentHlth  
## 0 0.7689532 0.8162400 3.038952  
## 1 0.6411492 0.7635023 4.732512

heart\_qda$scaling

## , , 0  
##   
## 1 2 3 4 5  
## HighBP 2.045679 -0.5810282 0.1640382 0.1371134 -0.18575180  
## HighChol 0.000000 2.1259787 0.1283218 0.1220631 -0.09012391  
## CholCheck 0.000000 0.0000000 -5.1063250 -0.1381543 -0.04385856  
## Smoker 0.000000 0.0000000 0.0000000 -2.0330369 -0.06734838  
## Stroke 0.000000 0.0000000 0.0000000 0.0000000 6.11214533  
## Diabetes 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  
## HvyAlcoholConsump 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  
## NoDocbcCost 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  
## GenHlth 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  
## DiffWalk 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  
## Sex 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  
## Age 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  
## Income 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  
## PhysActivity 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  
## Veggies 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  
## MentHlth 0.000000 0.0000000 0.0000000 0.0000000 0.00000000  
## 6 7 8 9 10  
## HighBP -0.45132865 -0.025631278 -0.009452038 0.40763438 0.15473065  
## HighChol -0.28147535 0.009828364 -0.006999770 0.16992599 0.03448704  
## CholCheck -0.18382684 0.088854655 0.318065711 0.05891646 0.06620459  
## Smoker -0.04186986 -0.228423966 -0.093241201 0.23361644 0.09992762  
## Stroke -0.25091548 0.088653320 -0.066706136 0.58078485 0.46917210  
## Diabetes 1.59739398 0.089295122 -0.048904730 0.33468759 0.10636514  
## HvyAlcoholConsump 0.00000000 4.317813294 -0.006179876 -0.13620649 -0.10628995  
## NoDocbcCost 0.00000000 0.000000000 3.674854540 0.59572864 0.17587860  
## GenHlth 0.00000000 0.000000000 0.000000000 -1.07085580 0.39242015  
## DiffWalk 0.00000000 0.000000000 0.000000000 0.00000000 -3.19454693  
## Sex 0.00000000 0.000000000 0.000000000 0.00000000 0.00000000  
## Age 0.00000000 0.000000000 0.000000000 0.00000000 0.00000000  
## Income 0.00000000 0.000000000 0.000000000 0.00000000 0.00000000  
## PhysActivity 0.00000000 0.000000000 0.000000000 0.00000000 0.00000000  
## Veggies 0.00000000 0.000000000 0.000000000 0.00000000 0.00000000  
## MentHlth 0.00000000 0.000000000 0.000000000 0.00000000 0.00000000  
## 11 12 13 14  
## HighBP -0.104999735 -0.56233379 0.09011139 -0.0556093605  
## HighChol -0.017067381 -0.36426565 -0.05833861 -0.0132280200  
## CholCheck 0.183317206 -0.17374473 -0.19835332 0.1186542336  
## Smoker -0.178396896 -0.18168565 0.14582755 -0.0658372042  
## Stroke 0.005713416 -0.37347700 0.21708812 -0.0062208182  
## Diabetes -0.030892461 -0.10259535 0.07554399 -0.0308529646  
## HvyAlcoholConsump 0.028507712 0.12955878 -0.20402846 0.0003090095  
## NoDocbcCost 0.151364883 0.56148754 0.57519549 -0.0011588982  
## GenHlth 0.005905183 0.05129252 0.24732931 -0.1519047511  
## DiffWalk 0.268313417 -0.37155843 0.44949760 -0.3974046590  
## Sex 2.041867831 0.15267839 -0.25071777 0.0258804004  
## Age 0.000000000 0.36410288 0.01751596 -0.0043460534  
## Income 0.000000000 0.00000000 0.54745224 0.0424244254  
## PhysActivity 0.000000000 0.00000000 0.00000000 -2.4918423937  
## Veggies 0.000000000 0.00000000 0.00000000 0.0000000000  
## MentHlth 0.000000000 0.00000000 0.00000000 0.0000000000  
## 15 16  
## HighBP -0.030329196 0.011426655  
## HighChol -0.027286779 -0.083546751  
## CholCheck 0.030122915 0.003204931  
## Smoker 0.017205221 -0.101433728  
## Stroke -0.048439008 -0.058381794  
## Diabetes -0.011222977 0.017952024  
## HvyAlcoholConsump 0.047969073 -0.148004506  
## NoDocbcCost 0.033794765 -0.392915114  
## GenHlth -0.046458798 -0.216081414  
## DiffWalk 0.006185232 -0.343832468  
## Sex -0.178317269 0.145385708  
## Age 0.010882971 0.055741976  
## Income 0.061167008 0.037489094  
## PhysActivity 0.293967337 0.063126173  
## Veggies -2.648094116 0.024117910  
## MentHlth 0.000000000 0.150444467  
##   
## , , 1  
##   
## 1 2 3 4 5  
## HighBP 2.309882 0.5426223 0.07283261 0.007756497 -0.145170404  
## HighChol 0.000000 -2.2484067 0.07430962 0.118116625 -0.007829033  
## CholCheck 0.000000 0.0000000 -9.52207291 -0.240097557 0.108794154  
## Smoker 0.000000 0.0000000 0.00000000 -2.062757724 -0.054336165  
## Stroke 0.000000 0.0000000 0.00000000 0.000000000 2.706898428  
## Diabetes 0.000000 0.0000000 0.00000000 0.000000000 0.000000000  
## HvyAlcoholConsump 0.000000 0.0000000 0.00000000 0.000000000 0.000000000  
## NoDocbcCost 0.000000 0.0000000 0.00000000 0.000000000 0.000000000  
## GenHlth 0.000000 0.0000000 0.00000000 0.000000000 0.000000000  
## DiffWalk 0.000000 0.0000000 0.00000000 0.000000000 0.000000000  
## Sex 0.000000 0.0000000 0.00000000 0.000000000 0.000000000  
## Age 0.000000 0.0000000 0.00000000 0.000000000 0.000000000  
## Income 0.000000 0.0000000 0.00000000 0.000000000 0.000000000  
## PhysActivity 0.000000 0.0000000 0.00000000 0.000000000 0.000000000  
## Veggies 0.000000 0.0000000 0.00000000 0.000000000 0.000000000  
## MentHlth 0.000000 0.0000000 0.00000000 0.000000000 0.000000000  
## 6 7 8 9 10  
## HighBP -0.3510721 -0.03557142 -0.02095122 -0.22944569 0.11947519  
## HighChol -0.1293921 0.01489818 -0.01144489 -0.03894144 -0.04412212  
## CholCheck -0.1977101 0.21202439 0.82327868 -0.12853809 0.12325833  
## Smoker -0.0226035 -0.13538329 -0.02997682 -0.21628658 0.03006512  
## Stroke -0.2019893 0.05182166 -0.19653082 -0.38345537 0.27296328  
## Diabetes 1.0915178 0.07093700 -0.01636117 -0.21265376 0.10546830  
## HvyAlcoholConsump 0.0000000 5.39456584 -0.03610916 0.19962506 -0.07916424  
## NoDocbcCost 0.0000000 0.00000000 3.22973412 -0.48787737 0.25180730  
## GenHlth 0.0000000 0.00000000 0.00000000 0.97858181 0.43730881  
## DiffWalk 0.0000000 0.00000000 0.00000000 0.00000000 -2.32644472  
## Sex 0.0000000 0.00000000 0.00000000 0.00000000 0.00000000  
## Age 0.0000000 0.00000000 0.00000000 0.00000000 0.00000000  
## Income 0.0000000 0.00000000 0.00000000 0.00000000 0.00000000  
## PhysActivity 0.0000000 0.00000000 0.00000000 0.00000000 0.00000000  
## Veggies 0.0000000 0.00000000 0.00000000 0.00000000 0.00000000  
## MentHlth 0.0000000 0.00000000 0.00000000 0.00000000 0.00000000  
## 11 12 13 14  
## HighBP 0.03499794 0.205201381 0.073864765 0.004074564  
## HighChol -0.03488614 0.020393893 -0.070420685 0.032930071  
## CholCheck -0.18785244 0.393860929 -0.358405057 0.146888134  
## Smoker -0.29171776 -0.074070164 0.113937737 -0.043342828  
## Stroke 0.05512286 0.086456318 0.189870274 -0.001226124  
## Diabetes -0.05388393 -0.009638615 0.048853539 -0.030085939  
## HvyAlcoholConsump -0.17991814 -0.190319438 -0.154382917 -0.057444840  
## NoDocbcCost 0.18187429 -0.778215073 0.464639922 0.007460527  
## GenHlth 0.04489852 -0.079778646 0.210411715 -0.134225386  
## DiffWalk 0.34906592 0.106640203 0.373406061 -0.374556931  
## Sex 2.08737108 -0.020561706 -0.492612663 0.099162442  
## Age 0.00000000 -0.464869787 -0.001677871 -0.010789169  
## Income 0.00000000 0.000000000 0.516977540 0.034527416  
## PhysActivity 0.00000000 0.000000000 0.000000000 -2.204766623  
## Veggies 0.00000000 0.000000000 0.000000000 0.000000000  
## MentHlth 0.00000000 0.000000000 0.000000000 0.000000000  
## 15 16  
## HighBP 0.002497843 -2.764762e-02  
## HighChol -0.005454677 4.516319e-02  
## CholCheck 0.060811792 2.947606e-02  
## Smoker 0.012217019 5.731183e-02  
## Stroke 0.049562863 7.462181e-02  
## Diabetes -0.002984322 -6.483228e-05  
## HvyAlcoholConsump -0.006588276 1.048529e-01  
## NoDocbcCost 0.037223887 3.703157e-01  
## GenHlth 0.023808480 1.867789e-01  
## DiffWalk -0.001297874 2.352271e-01  
## Sex 0.141286673 -1.485585e-01  
## Age -0.011873525 -9.451105e-02  
## Income -0.053902547 -4.369542e-02  
## PhysActivity -0.214106984 -6.135605e-02  
## Veggies 2.394416668 -3.489783e-02  
## MentHlth 0.000000000 -1.202138e-01

# Test the model and calculate accuracy  
qda\_predictions <- predict(heart\_qda, newdata = heart\_test)$class  
qda\_results <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,  
 Predicted = qda\_predictions  
)  
qda\_results$Correct <- qda\_results$Actual == qda\_results$Predicted  
qda\_confusion\_matrix <- table(Predicted = qda\_results$Predicted, Actual = qda\_results$Actual)  
print(qda\_confusion\_matrix)

## Actual  
## Predicted 0 1  
## 0 52651 3802  
## 1 7436 3212

accuracy\_qda <- (52651 + 3212) / (52651 + 3802 + 7436 + 3212)  
error\_qda <- 1 - accuracy\_qda  
cat("Accuracy:", accuracy\_qda, "\n")

## Accuracy: 0.8325211

cat("Error Rate:", error\_qda, "\n")

## Error Rate: 0.1674789

**Naive Bayes**  
The Naive Bayes model containing all significant predictors has an accuracy of 80.39%.

# Model with significant predictors  
heart\_nb <- naiveBayes(HeartDiseaseorAttack ~ HighBP + HighChol + CholCheck + Smoker + Stroke + Diabetes + HvyAlcoholConsump + NoDocbcCost + GenHlth + DiffWalk + Sex + Age + Income + PhysActivity + Veggies + MentHlth, data = heart\_train)  
heart\_nb

##   
## Naive Bayes Classifier for Discrete Predictors  
##   
## Call:  
## naiveBayes.default(x = X, y = Y, laplace = laplace)  
##   
## A-priori probabilities:  
## Y  
## 0 1   
## 0.90532504 0.09467496   
##   
## Conditional probabilities:  
## HighBP  
## Y [,1] [,2]  
## 0 0.3949205 0.4888351  
## 1 0.7501784 0.4329225  
##   
## HighChol  
## Y [,1] [,2]  
## 0 0.3955799 0.4889764  
## 1 0.7031882 0.4568665  
##   
## CholCheck  
## Y [,1] [,2]  
## 0 0.9595059 0.1971157  
## 1 0.9888175 0.1051575  
##   
## Smoker  
## Y [,1] [,2]  
## 0 0.4254684 0.4944154  
## 1 0.6190221 0.4856416  
##   
## Stroke  
## Y [,1] [,2]  
## 0 0.02793536 0.1647882  
## 1 0.16404949 0.3703315  
##   
## Diabetes  
## Y [,1] [,2]  
## 0 0.2554614 0.6548011  
## 1 0.6846895 0.9338205  
##   
## HvyAlcoholConsump  
## Y [,1] [,2]  
## 0 0.05785499 0.2334698  
## 1 0.03598620 0.1862613  
##   
## NoDocbcCost  
## Y [,1] [,2]  
## 0 0.08118733 0.2731234  
## 1 0.10908875 0.3117598  
##   
## GenHlth  
## Y [,1] [,2]  
## 0 2.420691 1.025722  
## 1 3.372175 1.085493  
##   
## DiffWalk  
## Y [,1] [,2]  
## 0 0.1427496 0.3498184  
## 1 0.4182132 0.4932802  
##   
## Sex  
## Y [,1] [,2]  
## 0 0.4269177 0.4946317  
## 1 0.5739353 0.4945181  
##   
## Age  
## Y [,1] [,2]  
## 0 7.813808 3.046982  
## 1 10.118011 2.237037  
##   
## Income  
## Y [,1] [,2]  
## 0 6.147091 2.034289  
## 1 5.143053 2.207500  
##   
## PhysActivity  
## Y [,1] [,2]  
## 0 0.7689532 0.4215036  
## 1 0.6411492 0.4796776  
##   
## Veggies  
## Y [,1] [,2]  
## 0 0.8162400 0.3872896  
## 1 0.7635023 0.4249439  
##   
## MentHlth  
## Y [,1] [,2]  
## 0 3.038952 7.196951  
## 1 4.732512 9.251938

# Test the model and calculate accuracy  
nb\_predictions <- predict(heart\_nb, newdata = heart\_test)  
nb\_results <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,  
 Predicted = nb\_predictions  
)  
nb\_results$Correct <- nb\_results$Actual == nb\_results$Predicted  
nb\_confusion\_matrix <- table(Predicted = nb\_results$Predicted, Actual = nb\_results$Actual)  
print(nb\_confusion\_matrix)

## Actual  
## Predicted 0 1  
## 0 50161 3234  
## 1 9926 3780

accuracy\_nb <- (50161 + 3780) / (50161 + 3234 + 9926 + 3780)  
error\_nb <- 1 - accuracy\_nb  
cat("Accuracy:", accuracy\_nb, "\n")

## Accuracy: 0.8038777

cat("Error Rate:", error\_nb, "\n")

## Error Rate: 0.1961223

**KNN (K-Nearest Neighbors)**  
The KNN model containing all significant predictors and K=3 has an accuracy of 98.13%.

# Model with all significant predictors and K = 3  
predictors <- c(  
 "HighBP", "HighChol", "CholCheck", "Smoker", "Stroke", "Diabetes",  
 "HvyAlcoholConsump", "NoDocbcCost", "GenHlth", "DiffWalk",  
 "Sex", "Age", "Income", "PhysActivity", "Veggies", "MentHlth"  
)  
train\_data <- heart\_train[, c(predictors, "HeartDiseaseorAttack")]  
test\_data <- heart\_test[, c(predictors, "HeartDiseaseorAttack")]  
knn\_predictions <- knn(  
 train = train\_data[, -length(predictors)],  
 test = test\_data[, -length(predictors)],  
 cl = train\_data$HeartDiseaseorAttack,  
 k = 3  
)  
knn\_results <- data.frame(  
 Actual = test\_data$HeartDiseaseorAttack,  
 Predicted = knn\_predictions  
)  
knn\_results$Correct <- knn\_results$Actual == knn\_results$Predicted  
knn\_confusion\_matrix <- table(Predicted = knn\_results$Predicted, Actual = knn\_results$Actual)  
print(knn\_confusion\_matrix)

## Actual  
## Predicted 0 1  
## 0 60081 1245  
## 1 6 5769

accuracy\_knn <- (60082 + 5763) / (60082 + 1251 + 5 + 5763)  
error\_knn <- 1 - accuracy\_knn  
cat("Accuracy:", accuracy\_knn, "\n")

## Accuracy: 0.9812819

cat("Error Rate:", error\_knn, "\n")

## Error Rate: 0.01871805

# Model with all significant predictors and K = 5  
knn\_predictions2 <- knn(  
 train = train\_data[, -length(predictors)],  
 test = test\_data[, -length(predictors)],  
 cl = train\_data$HeartDiseaseorAttack,  
 k = 5  
)  
knn\_results2 <- data.frame(  
 Actual = test\_data$HeartDiseaseorAttack,  
 Predicted = knn\_predictions2  
)  
knn\_results2$Correct <- knn\_results2$Actual == knn\_results2$Predicted  
knn\_confusion\_matrix2 <- table(Predicted = knn\_results2$Predicted, Actual = knn\_results2$Actual)  
print(knn\_confusion\_matrix2)

## Actual  
## Predicted 0 1  
## 0 60086 1419  
## 1 1 5595

accuracy\_knn2 <- (60085 + 5606) / (60085 + 1408 + 2 + 5606)  
error\_knn2 <- 1 - accuracy\_knn2  
cat("Accuracy:", accuracy\_knn2, "\n")

## Accuracy: 0.9789869

cat("Error Rate:", error\_knn2, "\n")

## Error Rate: 0.0210131

# Model with all significant predictors and K = 10  
knn\_predictions3 <- knn(  
 train = train\_data[, -length(predictors)],  
 test = test\_data[, -length(predictors)],  
 cl = train\_data$HeartDiseaseorAttack,  
 k = 10  
)  
knn\_results3 <- data.frame(  
 Actual = test\_data$HeartDiseaseorAttack,  
 Predicted = knn\_predictions3  
)  
knn\_results3$Correct <- knn\_results3$Actual == knn\_results3$Predicted  
knn\_confusion\_matrix3 <- table(Predicted = knn\_results3$Predicted, Actual = knn\_results3$Actual)  
print(knn\_confusion\_matrix3)

## Actual  
## Predicted 0 1  
## 0 60087 1679  
## 1 0 5335

accuracy\_knn3 <- (60087 + 5340) / (60087 + 1674 + 0 + 5340)  
error\_knn3 <- 1 - accuracy\_knn3  
cat("Accuracy:", accuracy\_knn3, "\n")

## Accuracy: 0.9750525

cat("Error Rate:", error\_knn3, "\n")

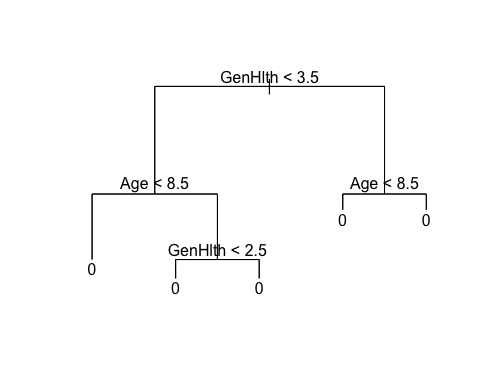
## Error Rate: 0.02494747

**Classification Trees**  
The classification tree containing all predictors and pruned with cross-validation has an accuracy of 89.55%.

# Model with all predictors  
heart\_tree <- tree(HeartDiseaseorAttack ~ ., heart\_train)  
summary(heart\_tree)

##   
## Classification tree:  
## tree(formula = HeartDiseaseorAttack ~ ., data = heart\_train)  
## Variables actually used in tree construction:  
## [1] "GenHlth" "Age"   
## Number of terminal nodes: 5   
## Residual mean deviance: 0.5353 = 95050 / 177600   
## Misclassification error rate: 0.09467 = 16812 / 177576

plot(heart\_tree)  
text(heart\_tree, pretty = 0)



# Test the model and calculate accuracy  
tree\_predict <- predict(heart\_tree, heart\_test, type = "class")  
tree\_results <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,  
 Predicted = tree\_predict  
)  
tree\_results$Correct <- tree\_results$Actual == tree\_results$Predicted  
tree\_confusion\_matrix <- table(Predicted = tree\_results$Predicted, Actual = tree\_results$Actual)  
print(tree\_confusion\_matrix)

## Actual  
## Predicted 0 1  
## 0 60087 7014  
## 1 0 0

accuracy\_tree <- (60087 + 0) / (60087 + 7014 + 0 + 0)  
error\_tree <- 1 - accuracy\_tree  
cat("Accuracy:", accuracy\_tree, "\n")

## Accuracy: 0.895471

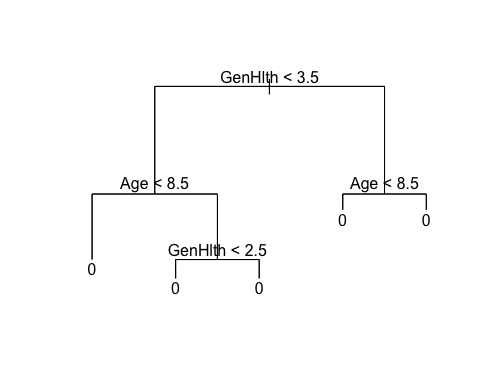
cat("Error Rate:", error\_tree, "\n")

## Error Rate: 0.104529

# Prune the tree with cross-validation  
set.seed(123)  
cv\_heart <- cv.tree(heart\_tree, FUN = prune.misclass)  
cv\_heart

## $size  
## [1] 5 1  
##   
## $dev  
## [1] 16812 16812  
##   
## $k  
## [1] -Inf 0  
##   
## $method  
## [1] "misclass"  
##   
## attr(,"class")  
## [1] "prune" "tree.sequence"

prune\_heart <- prune.misclass(heart\_tree, best = 5)  
plot(prune\_heart)  
text(prune\_heart, pretty = 0)



# Test the pruned tree and calculate accuracy  
tree\_predict2 <- predict(prune\_heart, heart\_test, type = "class")  
tree\_results2 <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,  
 Predicted = tree\_predict2  
)  
tree\_results2$Correct <- tree\_results2$Actual == tree\_results2$Predicted  
tree\_confusion\_matrix2 <- table(Predicted = tree\_results2$Predicted, Actual = tree\_results2$Actual)  
print(tree\_confusion\_matrix2)

## Actual  
## Predicted 0 1  
## 0 60087 7014  
## 1 0 0

accuracy\_tree2 <- (60087 + 0) / (60087 + 7014 + 0 + 0)  
error\_tree2 <- 1 - accuracy\_tree2  
cat("Accuracy:", accuracy\_tree2, "\n")

## Accuracy: 0.895471

cat("Error Rate:", error\_tree2, "\n")

## Error Rate: 0.104529

# The pruned and unpruned trees are the same

**Bagging**  
The bagged classification tree containing all predictors has an accuracy of 87.75%. The most important predictors in the tree are (1) Age, (2) BMI, and (3) Income.

# Model with all predictors  
set.seed(123)  
heart\_bag <- randomForest(HeartDiseaseorAttack ~ ., data = heart\_train, mtry = 16, importance = TRUE, ntree = 25)  
heart\_bag

##   
## Call:  
## randomForest(formula = HeartDiseaseorAttack ~ ., data = heart\_train, mtry = 16, importance = TRUE, ntree = 25)   
## Type of random forest: classification  
## Number of trees: 25  
## No. of variables tried at each split: 16  
##   
## OOB estimate of error rate: 10.92%  
## Confusion matrix:  
## 0 1 class.error  
## 0 155100 5661 0.03521376  
## 1 13724 3088 0.81632167

importance(heart\_bag)

## 0 1 MeanDecreaseAccuracy MeanDecreaseGini  
## HighBP -9.453797 19.8925326 -0.3090534 631.2097  
## HighChol -12.549170 26.4123276 -3.2835389 564.9037  
## CholCheck 5.092407 -2.6031519 4.1494782 126.9510  
## BMI 7.533636 -3.8052444 4.9277909 5530.3945  
## Smoker -3.230946 7.6994319 0.4178417 869.6693  
## Stroke 10.338214 32.1654268 22.6743155 715.7023  
## Diabetes 3.808568 8.6099910 7.6360371 957.7022  
## PhysActivity 6.080891 -4.6828963 3.4617302 956.7861  
## Fruits 5.676729 -1.1941288 4.0209661 1082.7540  
## Veggies 2.003716 -0.1069718 1.8029961 846.1092  
## HvyAlcoholConsump 2.437638 -0.6126569 1.6702023 324.2037  
## AnyHealthcare 4.009127 -2.8926022 2.7306862 232.6367  
## NoDocbcCost 8.240178 -0.2863122 6.9116860 450.2840  
## GenHlth 16.273986 17.8019342 26.9336732 2279.5800  
## MentHlth 10.310916 -3.6418811 7.3903003 1950.9674  
## PhysHlth 14.775446 -2.4436951 14.2738733 2535.3968  
## DiffWalk 1.541262 12.2072850 9.2774139 860.0739  
## Sex 7.924247 11.9507307 13.6468996 645.5515  
## Age 11.125472 34.9689543 24.5523960 2777.3193  
## Education 19.012976 -6.2168920 14.6522430 2207.2180  
## Income 14.589597 -2.0897785 13.9938751 3157.4731

# Test the model and calculate accuracy  
tree\_predict\_bag <- predict(heart\_bag, heart\_test, type = "class")  
tree\_results\_bag <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,  
 Predicted = tree\_predict\_bag  
)  
tree\_results\_bag$Correct <- tree\_results\_bag$Actual == tree\_results\_bag$Predicted  
tree\_confusion\_matrix\_bag <- table(Predicted = tree\_results\_bag$Predicted, Actual = tree\_results\_bag$Actual)  
print(tree\_confusion\_matrix\_bag)

## Actual  
## Predicted 0 1  
## 0 58207 5894  
## 1 1880 1120

accuracy\_tree\_bag <- (57539 + 1341) / (57539 + 5673 + 2548 + 1341)  
error\_tree\_bag <- 1 - accuracy\_tree\_bag  
cat("Accuracy:", accuracy\_tree\_bag, "\n")

## Accuracy: 0.8774832

cat("Error Rate:", error\_tree\_bag, "\n")

## Error Rate: 0.1225168

**Random Forests**  
The random forest classification tree containing all predictors has an accuracy of 89.45%. The most important predictors in the tree are (1) Age, (2) BMI, and (3) GenHlth.

# Model with all predictors  
set.seed(123)  
heart\_rf <- randomForest(HeartDiseaseorAttack ~ ., data = heart\_train, mtry = 4, importance = TRUE, ntree = 25)  
heart\_rf

##   
## Call:  
## randomForest(formula = HeartDiseaseorAttack ~ ., data = heart\_train, mtry = 4, importance = TRUE, ntree = 25)   
## Type of random forest: classification  
## Number of trees: 25  
## No. of variables tried at each split: 4  
##   
## OOB estimate of error rate: 9.84%  
## Confusion matrix:  
## 0 1 class.error  
## 0 158101 2661 0.01655242  
## 1 14810 2001 0.88097079

importance(heart\_rf)

## 0 1 MeanDecreaseAccuracy MeanDecreaseGini  
## HighBP -9.6305526 12.5565119 0.6843860 809.81440  
## HighChol -10.3694707 14.8229002 0.9644538 679.93561  
## CholCheck 2.6103315 -0.4093466 2.5155214 97.11157  
## BMI 12.9636999 -2.9705478 9.8569128 3263.23924  
## Smoker -9.5435946 9.6846098 -0.3882864 502.72400  
## Stroke 5.6253149 39.7938864 18.8813333 763.19208  
## Diabetes -0.6267632 9.1181822 3.6660343 673.13682  
## PhysActivity 4.7218669 -2.5749532 2.2937435 620.59727  
## Fruits 3.8365262 -1.7856203 2.1357503 668.07655  
## Veggies 2.0719579 -2.0302961 1.0935410 591.07056  
## HvyAlcoholConsump 2.6230475 0.3989772 2.9316309 209.55416  
## AnyHealthcare 4.4841208 -0.5483266 4.0833164 182.03580  
## NoDocbcCost 5.4151921 -2.0114670 3.8731648 354.61536  
## GenHlth 7.0065827 26.9871170 16.3452153 1762.41789  
## MentHlth 12.8490718 -7.6722303 10.5400008 1425.18542  
## PhysHlth 10.0921953 -1.8453169 9.8522780 1872.28068  
## DiffWalk 1.4581647 12.2251124 9.6233747 682.55238  
## Sex 9.2676631 10.2017891 11.2873441 595.04416  
## Age 6.4873333 19.8988746 16.6811849 2487.90392  
## Education 11.2803123 -3.7764568 8.2815047 1442.37075  
## Income 6.3686110 2.1722698 7.6344434 1940.88147

# Test the model and calculate accuracy  
tree\_predict\_rf <- predict(heart\_rf, heart\_test, type = "class")  
tree\_results\_rf <- data.frame(  
 Actual = heart\_test$HeartDiseaseorAttack,  
 Predicted = tree\_predict\_rf  
)  
tree\_results\_rf$Correct <- tree\_results\_rf$Actual == tree\_results\_rf$Predicted  
tree\_confusion\_matrix\_rf <- table(Predicted = tree\_results\_rf$Predicted, Actual = tree\_results\_rf$Actual)  
print(tree\_confusion\_matrix\_rf)

## Actual  
## Predicted 0 1  
## 0 59362 6335  
## 1 725 679

accuracy\_tree\_rf <- (59362 + 679) / (59362 + 6335 + 725 + 679)  
error\_tree\_rf <- 1 - accuracy\_tree\_rf  
cat("Accuracy:", accuracy\_tree\_rf, "\n")

## Accuracy: 0.8947855

cat("Error Rate:", error\_tree\_rf, "\n")

## Error Rate: 0.1052145

**Overview of models**  
Overall, the model with the highest accuracy is kNN with k=3. This model has an accuracy of 98.13%.

# Table with models and relative accuracies  
classification\_overview <- data.frame(  
 Method = c("Logistic Regression", "LDA", "QDA", "Naive Bayes", "kNN (k=3)", "CV Tree", "Bagging", "Random Forest"),  
 Accuracy = c("89.78%", "89.14%", "83.25%", "80.39%", "98.13%", "89.55%", "87.75%", "89.45%")  
)  
classification\_table <- kable(classification\_overview, "markdown") %>%  
 kable\_styling(full\_width = FALSE) %>%  
 column\_spec(1, bold = TRUE)

## Warning in kable\_styling(., full\_width = FALSE): Please specify format in  
## kable. kableExtra can customize either HTML or LaTeX outputs. See  
## https://haozhu233.github.io/kableExtra/ for details.

## Warning in column\_spec(., 1, bold = TRUE): Please specify format in kable.  
## kableExtra can customize either HTML or LaTeX outputs. See  
## https://haozhu233.github.io/kableExtra/ for details.

classification\_table

| Method | Accuracy |
| --- | --- |
| Logistic Regression | 89.78% |
| LDA | 89.14% |
| QDA | 83.25% |
| Naive Bayes | 80.39% |
| kNN (k=3) | 98.13% |
| CV Tree | 89.55% |
| Bagging | 87.75% |
| Random Forest | 89.45% |

# Section 2: What indicators can be used to predict HIV rates?

**Loading data and packages**

library(readxl)  
library(tidyverse)  
library(boot)  
library(pls)

##   
## Attaching package: 'pls'

## The following object is masked from 'package:coefplot':  
##   
## coefplot

## The following object is masked from 'package:corrplot':  
##   
## corrplot

## The following object is masked from 'package:stats':  
##   
## loadings

library(leaps)  
library(glmnet)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

## Loaded glmnet 4.1-8

library(tree)  
library(randomForest)  
library(gbm)  
library(kableExtra)  
library(tidyverse)  
library(corrplot)  
hiv\_df <- read\_xlsx("Data/HIV\_Data.xlsx")

**Data Tidying**  
Cleaned and formatted data, and used bootstrapping to generate additional observations.

# Remove male and female (only interested in total due to missing values)  
hiv\_df <- subset(hiv\_df, !(Sex %in% c("Male", "Female")))  
hiv\_df <- hiv\_df[ , -3]  
  
# Pivot from long to wide format  
hiv\_df <- pivot\_wider(  
 data = hiv\_df,  
 names\_from = c(Indicator), # Specify the columns to pivot  
 values\_from = c(OBS\_VALUE), # Specify the values column  
 names\_sep = "\_"  
)

## Warning: Values from `OBS\_VALUE` are not uniquely identified; output will contain  
## list-cols.  
## • Use `values\_fn = list` to suppress this warning.  
## • Use `values\_fn = {summary\_fun}` to summarise duplicates.  
## • Use the following dplyr code to identify duplicates.  
## {data} %>%  
## dplyr::group\_by(`Geographic area`, TIME\_PERIOD, Indicator) %>%  
## dplyr::summarise(n = dplyr::n(), .groups = "drop") %>%  
## dplyr::filter(n > 1L)

hiv\_df <- hiv\_df[ , -13]  
  
# Change column names  
colnames(hiv\_df) <- c("country", "year", "infant\_mortality\_rate", "fertility\_rate", "life\_exp", "pop\_growth\_rate", "urban\_pop", "youth\_literacy\_rate", "aids\_death\_rate", "hiv\_infection\_rate", "mother\_child\_hiv\_transmis\_rate", "per\_child\_under\_poverty")  
  
# Interpolate missing values for years based on other years for that country  
hiv\_df <- hiv\_df %>%  
 group\_by(country) %>%  
 fill(c(pop\_growth\_rate, youth\_literacy\_rate, per\_child\_under\_poverty), .direction = "updown") %>%  
 ungroup()  
  
# Convert to numeric columns  
hiv\_df <- hiv\_df %>%  
 unnest\_wider(infant\_mortality\_rate, names\_sep = "\_") %>%  
 unnest\_wider(fertility\_rate, names\_sep = "\_") %>%  
 unnest\_wider(life\_exp, names\_sep = "\_") %>%  
 unnest\_wider(pop\_growth\_rate, names\_sep = "\_") %>%  
 unnest\_wider(urban\_pop, names\_sep = "\_") %>%  
 unnest\_wider(youth\_literacy\_rate, names\_sep = "\_") %>%  
 unnest\_wider(aids\_death\_rate, names\_sep = "\_") %>%  
 unnest\_wider(hiv\_infection\_rate, names\_sep = "\_") %>%  
 unnest\_wider(mother\_child\_hiv\_transmis\_rate, names\_sep = "\_") %>%  
 unnest\_wider(per\_child\_under\_poverty, names\_sep = "\_") %>%  
 mutate(across(-c(country, year), ~as.numeric(as.character(.))))

## Warning: There were 2 warnings in `mutate()`.  
## The first warning was:  
## ℹ In argument: `across(-c(country, year), ~as.numeric(as.character(.)))`.  
## Caused by warning:  
## ! NAs introduced by coercion  
## ℹ Run `dplyr::last\_dplyr\_warnings()` to see the 1 remaining warning.

hiv\_df <- hiv\_df[ , -c(4, 5, 7, 9, 11)]  
colnames(hiv\_df) <- c("country", "year", "infant\_mortality\_rate", "fertility\_rate", "life\_exp", "pop\_growth\_rate", "urban\_pop", "youth\_literacy\_rate", "aids\_death\_rate", "hiv\_infection\_rate", "mother\_child\_hiv\_transmis\_rate", "per\_child\_under\_poverty")  
  
# Check for NA values  
colSums(is.na(hiv\_df))

## country year   
## 1 1   
## infant\_mortality\_rate fertility\_rate   
## 325 199   
## life\_exp pop\_growth\_rate   
## 199 199   
## urban\_pop youth\_literacy\_rate   
## 235 1504   
## aids\_death\_rate hiv\_infection\_rate   
## 1227 1304   
## mother\_child\_hiv\_transmis\_rate per\_child\_under\_poverty   
## 1540 1324

hiv\_df <- na.omit(hiv\_df)  
  
# Use bootstrapping to expand the dataset to 1000 observations  
set.seed(123)  
generate\_boot\_dataset <- function(data) {  
 boot\_indices <- sample(nrow(data), replace = TRUE)  
 boot\_data <- data[boot\_indices, ]  
 return(boot\_data)  
}  
num\_bootstrap\_samples <- ceiling(1000 / nrow(hiv\_df))  
boot\_datasets <- lapply(1:num\_bootstrap\_samples, function(i) generate\_boot\_dataset(hiv\_df))  
hiv\_df\_boot <- do.call(rbind, boot\_datasets)

**Data Exploration**  
Colinearity is present between multiple variables (infant\_mortality\_rate, fertility\_rate, life\_exp, pop\_growth\_rate, and youth\_literacy\_rate all seem to be correlated). This is something to keep in mind during the evaluation of models.

# View the dataset and summary statistics  
head(hiv\_df\_boot)

## # A tibble: 6 × 12  
## country year infant\_mortality\_rate fertility\_rate life\_exp pop\_growth\_rate  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Indonesia 2019 20.1 2.21 70.5 0.654  
## 2 Algeria 2017 21.0 3.05 75.7 1.24   
## 3 Kyrgyzstan 2017 17.8 3.09 70.6 1.33   
## 4 Sao Tome … 2015 18.6 4.39 67.2 1.75   
## 5 El Salvad… 2013 14.4 2.15 71.8 0.370  
## 6 Rwanda 2017 32.9 4.03 65.9 2.00   
## # ℹ 6 more variables: urban\_pop <dbl>, youth\_literacy\_rate <dbl>,  
## # aids\_death\_rate <dbl>, hiv\_infection\_rate <dbl>,  
## # mother\_child\_hiv\_transmis\_rate <dbl>, per\_child\_under\_poverty <dbl>

summary(hiv\_df\_boot)

## country year infant\_mortality\_rate fertility\_rate   
## Length:1233 Length:1233 Min. : 2.05 Min. :1.331   
## Class :character Class :character 1st Qu.:14.45 1st Qu.:2.194   
## Mode :character Mode :character Median :28.98 Median :3.050   
## Mean :32.44 Mean :3.389   
## 3rd Qu.:45.57 3rd Qu.:4.489   
## Max. :98.03 Max. :7.344   
## life\_exp pop\_growth\_rate urban\_pop youth\_literacy\_rate  
## Min. :51.49 Min. :-0.4127 Min. :16.21 Min. : 43.46   
## 1st Qu.:61.62 1st Qu.: 0.8632 1st Qu.:35.59 1st Qu.: 83.63   
## Median :66.44 Median : 1.3011 Median :50.65 Median : 95.42   
## Mean :67.27 Mean : 1.3996 Mean :51.78 Mean : 87.42   
## 3rd Qu.:73.13 3rd Qu.: 2.0922 3rd Qu.:68.87 3rd Qu.: 99.01   
## Max. :79.48 Max. : 3.3609 Max. :95.60 Max. :100.00   
## aids\_death\_rate hiv\_infection\_rate mother\_child\_hiv\_transmis\_rate  
## Min. : 0.030 Min. : 0.010 Min. : 1.61   
## 1st Qu.: 0.220 1st Qu.: 0.040 1st Qu.: 13.46   
## Median : 1.180 Median : 0.260 Median : 18.92   
## Mean : 9.953 Mean : 1.049 Mean : 21.96   
## 3rd Qu.:15.610 3rd Qu.: 0.810 3rd Qu.: 28.33   
## Max. :96.080 Max. :17.610 Max. :128.28   
## per\_child\_under\_poverty  
## Min. : 0.4306   
## 1st Qu.:25.1501   
## Median :34.9413   
## Mean :37.5100   
## 3rd Qu.:50.1748   
## Max. :84.6042

str(hiv\_df\_boot)

## tibble [1,233 × 12] (S3: tbl\_df/tbl/data.frame)  
## $ country : chr [1:1233] "Indonesia" "Algeria" "Kyrgyzstan" "Sao Tome and Principe" ...  
## $ year : chr [1:1233] "2019" "2017" "2017" "2015" ...  
## $ infant\_mortality\_rate : num [1:1233] 20.1 21 17.8 18.6 14.4 ...  
## $ fertility\_rate : num [1:1233] 2.21 3.05 3.09 4.39 2.15 ...  
## $ life\_exp : num [1:1233] 70.5 75.7 70.6 67.2 71.8 ...  
## $ pop\_growth\_rate : num [1:1233] 0.654 1.237 1.328 1.749 0.37 ...  
## $ urban\_pop : num [1:1233] 56 72.1 36.1 70.2 68 ...  
## $ youth\_literacy\_rate : num [1:1233] 99.8 97.4 99.8 97.8 98.5 ...  
## $ aids\_death\_rate : num [1:1233] 0.44 0.07 0.03 11.07 0.29 ...  
## $ hiv\_infection\_rate : num [1:1233] 0.16 0.03 0.03 0.04 0.21 0.42 0.36 0.48 0.03 1.68 ...  
## $ mother\_child\_hiv\_transmis\_rate: num [1:1233] 29.7 24.8 31.1 21.4 13.9 ...  
## $ per\_child\_under\_poverty : num [1:1233] 10.55 6.33 29.26 63.73 34.94 ...  
## - attr(\*, "na.action")= 'omit' Named int [1:2011] 10 11 12 13 14 15 16 17 18 28 ...  
## ..- attr(\*, "names")= chr [1:2011] "10" "11" "12" "13" ...

dim(hiv\_df\_boot)

## [1] 1233 12

names(hiv\_df\_boot)

## [1] "country" "year"   
## [3] "infant\_mortality\_rate" "fertility\_rate"   
## [5] "life\_exp" "pop\_growth\_rate"   
## [7] "urban\_pop" "youth\_literacy\_rate"   
## [9] "aids\_death\_rate" "hiv\_infection\_rate"   
## [11] "mother\_child\_hiv\_transmis\_rate" "per\_child\_under\_poverty"

# Check for correlated variables  
cor\_matrix <- cor(hiv\_df\_boot[, c(3:12)])  
corrplot(cor\_matrix, method = "color", tl.col = "black", tl.srt = 45)

## Warning in plot.window(...): "method" is not a graphical parameter

## Warning in plot.window(...): "tl.col" is not a graphical parameter

## Warning in plot.window(...): "tl.srt" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "method" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "tl.col" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "tl.srt" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "method" is not a  
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "tl.col" is not a  
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "tl.srt" is not a  
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "method" is not a  
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "tl.col" is not a  
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "tl.srt" is not a  
## graphical parameter

## Warning in box(...): "method" is not a graphical parameter

## Warning in box(...): "tl.col" is not a graphical parameter

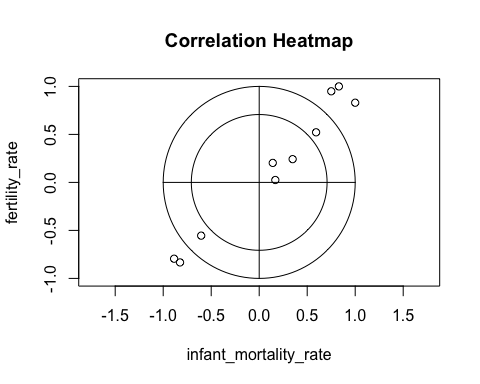
## Warning in box(...): "tl.srt" is not a graphical parameter

## Warning in title(...): "method" is not a graphical parameter

## Warning in title(...): "tl.col" is not a graphical parameter

## Warning in title(...): "tl.srt" is not a graphical parameter

title("Correlation Heatmap")



cor\_matrix\_filtered <- cor\_matrix  
cor\_matrix\_filtered[abs(cor\_matrix) <= 0.7] <- NA  
print(cor\_matrix\_filtered)

## infant\_mortality\_rate fertility\_rate life\_exp  
## infant\_mortality\_rate 1.0000000 0.8300778 -0.8865746  
## fertility\_rate 0.8300778 1.0000000 -0.7949911  
## life\_exp -0.8865746 -0.7949911 1.0000000  
## pop\_growth\_rate 0.7506318 0.9491751 -0.7188660  
## urban\_pop NA NA NA  
## youth\_literacy\_rate -0.8245610 -0.8335437 NA  
## aids\_death\_rate NA NA NA  
## hiv\_infection\_rate NA NA NA  
## mother\_child\_hiv\_transmis\_rate NA NA NA  
## per\_child\_under\_poverty NA NA NA  
## pop\_growth\_rate urban\_pop youth\_literacy\_rate  
## infant\_mortality\_rate 0.7506318 NA -0.8245610  
## fertility\_rate 0.9491751 NA -0.8335437  
## life\_exp -0.7188660 NA NA  
## pop\_growth\_rate 1.0000000 NA -0.7238268  
## urban\_pop NA 1 NA  
## youth\_literacy\_rate -0.7238268 NA 1.0000000  
## aids\_death\_rate NA NA NA  
## hiv\_infection\_rate NA NA NA  
## mother\_child\_hiv\_transmis\_rate NA NA NA  
## per\_child\_under\_poverty NA NA NA  
## aids\_death\_rate hiv\_infection\_rate  
## infant\_mortality\_rate NA NA  
## fertility\_rate NA NA  
## life\_exp NA NA  
## pop\_growth\_rate NA NA  
## urban\_pop NA NA  
## youth\_literacy\_rate NA NA  
## aids\_death\_rate 1.0000000 0.8657092  
## hiv\_infection\_rate 0.8657092 1.0000000  
## mother\_child\_hiv\_transmis\_rate NA NA  
## per\_child\_under\_poverty NA NA  
## mother\_child\_hiv\_transmis\_rate  
## infant\_mortality\_rate NA  
## fertility\_rate NA  
## life\_exp NA  
## pop\_growth\_rate NA  
## urban\_pop NA  
## youth\_literacy\_rate NA  
## aids\_death\_rate NA  
## hiv\_infection\_rate NA  
## mother\_child\_hiv\_transmis\_rate 1  
## per\_child\_under\_poverty NA  
## per\_child\_under\_poverty  
## infant\_mortality\_rate NA  
## fertility\_rate NA  
## life\_exp NA  
## pop\_growth\_rate NA  
## urban\_pop NA  
## youth\_literacy\_rate NA  
## aids\_death\_rate NA  
## hiv\_infection\_rate NA  
## mother\_child\_hiv\_transmis\_rate NA  
## per\_child\_under\_poverty 1

**Best Subset Selection**  
The model created using best subset selection determined the model with the lowest MSE is one that contains 6 predictors (infant\_mortality\_rate, fertility\_rate, youth\_literacy\_rate, aids\_death\_rate, mother\_child\_hiv\_transmis\_rate, and per\_child\_under\_poverty) with an MSE of 1.116785 and an adjusted R-squared value of 0.778.

# Creating a training and testing set  
set.seed(123)   
train\_hiv <- sample(c(TRUE, FALSE), nrow(hiv\_df\_boot), replace = TRUE)  
test\_hiv <- (!train\_hiv)  
  
# Creating the model  
best\_subset\_hiv <- regsubsets(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot[train\_hiv, ], nvmax = 9)  
best\_summary <- summary(best\_subset\_hiv)  
best\_summary

## Subset selection object  
## Call: regsubsets.formula(hiv\_infection\_rate ~ infant\_mortality\_rate +   
## fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop +   
## youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate +   
## per\_child\_under\_poverty, data = hiv\_df\_boot[train\_hiv, ],   
## nvmax = 9)  
## 9 Variables (and intercept)  
## Forced in Forced out  
## infant\_mortality\_rate FALSE FALSE  
## fertility\_rate FALSE FALSE  
## life\_exp FALSE FALSE  
## pop\_growth\_rate FALSE FALSE  
## urban\_pop FALSE FALSE  
## youth\_literacy\_rate FALSE FALSE  
## aids\_death\_rate FALSE FALSE  
## mother\_child\_hiv\_transmis\_rate FALSE FALSE  
## per\_child\_under\_poverty FALSE FALSE  
## 1 subsets of each size up to 9  
## Selection Algorithm: exhaustive  
## infant\_mortality\_rate fertility\_rate life\_exp pop\_growth\_rate  
## 1 ( 1 ) " " " " " " " "   
## 2 ( 1 ) " " "\*" " " " "   
## 3 ( 1 ) " " "\*" " " " "   
## 4 ( 1 ) " " "\*" " " " "   
## 5 ( 1 ) " " "\*" " " " "   
## 6 ( 1 ) "\*" "\*" " " " "   
## 7 ( 1 ) "\*" "\*" " " "\*"   
## 8 ( 1 ) "\*" "\*" " " "\*"   
## 9 ( 1 ) "\*" "\*" "\*" "\*"   
## urban\_pop youth\_literacy\_rate aids\_death\_rate  
## 1 ( 1 ) " " " " "\*"   
## 2 ( 1 ) " " " " "\*"   
## 3 ( 1 ) " " " " "\*"   
## 4 ( 1 ) " " " " "\*"   
## 5 ( 1 ) " " "\*" "\*"   
## 6 ( 1 ) " " "\*" "\*"   
## 7 ( 1 ) " " "\*" "\*"   
## 8 ( 1 ) "\*" "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*"   
## mother\_child\_hiv\_transmis\_rate per\_child\_under\_poverty  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) " " " "   
## 3 ( 1 ) " " "\*"   
## 4 ( 1 ) "\*" "\*"   
## 5 ( 1 ) "\*" "\*"   
## 6 ( 1 ) "\*" "\*"   
## 7 ( 1 ) "\*" "\*"   
## 8 ( 1 ) "\*" "\*"   
## 9 ( 1 ) "\*" "\*"

# Testing and determining the best model  
test\_mat\_best <- model.matrix(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot[test\_hiv, ])  
val.errors <- rep(NA, 9)  
for (i in 1:9) {  
 coefi <- coef(best\_subset\_hiv, id = i)  
 pred <- test\_mat\_best[, names(coefi)] %\*% coefi  
 val.errors[i] <- mean((hiv\_df\_boot$hiv\_infection\_rate[test\_hiv] - pred)^2)  
}  
val.errors

## [1] 1.393887 1.192356 1.159057 1.144428 1.123322 1.116785 1.117192 1.118514  
## [9] 1.118408

which.min(val.errors)

## [1] 6

coef(best\_subset\_hiv, 6)

## (Intercept) infant\_mortality\_rate   
## 2.912988940 -0.008395259   
## fertility\_rate youth\_literacy\_rate   
## -0.555738245 -0.019081947   
## aids\_death\_rate mother\_child\_hiv\_transmis\_rate   
## 0.139927726 0.008088391   
## per\_child\_under\_poverty   
## 0.010227776

cat("Adjusted RSq:", best\_summary$rsq[6], "\n")

## Adjusted RSq: 0.778206

cat("MSE:", val.errors[6], "\n")

## MSE: 1.116785

**Forward Stepwise Selection**  
The best model created with forward stepwise selection contains the same 6 predictors as the best subset selection model, with the same MSE of 1.116785 and adjusted R-squared value of 0.778.

forward\_subset\_hiv <- regsubsets(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot[train\_hiv, ], nvmax = 9, method = "forward")  
forward\_summary <- summary(forward\_subset\_hiv)  
forward\_summary

## Subset selection object  
## Call: regsubsets.formula(hiv\_infection\_rate ~ infant\_mortality\_rate +   
## fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop +   
## youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate +   
## per\_child\_under\_poverty, data = hiv\_df\_boot[train\_hiv, ],   
## nvmax = 9, method = "forward")  
## 9 Variables (and intercept)  
## Forced in Forced out  
## infant\_mortality\_rate FALSE FALSE  
## fertility\_rate FALSE FALSE  
## life\_exp FALSE FALSE  
## pop\_growth\_rate FALSE FALSE  
## urban\_pop FALSE FALSE  
## youth\_literacy\_rate FALSE FALSE  
## aids\_death\_rate FALSE FALSE  
## mother\_child\_hiv\_transmis\_rate FALSE FALSE  
## per\_child\_under\_poverty FALSE FALSE  
## 1 subsets of each size up to 9  
## Selection Algorithm: forward  
## infant\_mortality\_rate fertility\_rate life\_exp pop\_growth\_rate  
## 1 ( 1 ) " " " " " " " "   
## 2 ( 1 ) " " "\*" " " " "   
## 3 ( 1 ) " " "\*" " " " "   
## 4 ( 1 ) " " "\*" " " " "   
## 5 ( 1 ) " " "\*" " " " "   
## 6 ( 1 ) "\*" "\*" " " " "   
## 7 ( 1 ) "\*" "\*" " " "\*"   
## 8 ( 1 ) "\*" "\*" " " "\*"   
## 9 ( 1 ) "\*" "\*" "\*" "\*"   
## urban\_pop youth\_literacy\_rate aids\_death\_rate  
## 1 ( 1 ) " " " " "\*"   
## 2 ( 1 ) " " " " "\*"   
## 3 ( 1 ) " " " " "\*"   
## 4 ( 1 ) " " " " "\*"   
## 5 ( 1 ) " " "\*" "\*"   
## 6 ( 1 ) " " "\*" "\*"   
## 7 ( 1 ) " " "\*" "\*"   
## 8 ( 1 ) "\*" "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*"   
## mother\_child\_hiv\_transmis\_rate per\_child\_under\_poverty  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) " " " "   
## 3 ( 1 ) " " "\*"   
## 4 ( 1 ) "\*" "\*"   
## 5 ( 1 ) "\*" "\*"   
## 6 ( 1 ) "\*" "\*"   
## 7 ( 1 ) "\*" "\*"   
## 8 ( 1 ) "\*" "\*"   
## 9 ( 1 ) "\*" "\*"

# Testing and determining the best model  
test\_mat\_forward <- model.matrix(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot[test\_hiv, ], method = "forward")  
val.errors2 <- rep(NA, 9)  
for (i in 1:9) {  
 coefi <- coef(forward\_subset\_hiv, id = i)  
 pred <- test\_mat\_forward[, names(coefi)] %\*% coefi  
 val.errors2[i] <- mean((hiv\_df\_boot$hiv\_infection\_rate[test\_hiv] - pred)^2)  
}  
val.errors2

## [1] 1.393887 1.192356 1.159057 1.144428 1.123322 1.116785 1.117192 1.118514  
## [9] 1.118408

which.min(val.errors2)

## [1] 6

coef(forward\_subset\_hiv, 6)

## (Intercept) infant\_mortality\_rate   
## 2.912988940 -0.008395259   
## fertility\_rate youth\_literacy\_rate   
## -0.555738245 -0.019081947   
## aids\_death\_rate mother\_child\_hiv\_transmis\_rate   
## 0.139927726 0.008088391   
## per\_child\_under\_poverty   
## 0.010227776

cat("Adjusted RSq:", forward\_summary$rsq[6], "\n")

## Adjusted RSq: 0.778206

cat("MSE:", val.errors2[6], "\n")

## MSE: 1.116785

**Backward Stepwise Selection**  
The best model created with backward stepwise selection contains the same 6 predictors as the best subset selection model, with the same MSE of 1.116785 and adjusted R-squared value of 0.778.

backward\_subset\_hiv <- regsubsets(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot[train\_hiv, ], nvmax = 9, method = "backward")  
backward\_summary <- summary(backward\_subset\_hiv)  
backward\_summary

## Subset selection object  
## Call: regsubsets.formula(hiv\_infection\_rate ~ infant\_mortality\_rate +   
## fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop +   
## youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate +   
## per\_child\_under\_poverty, data = hiv\_df\_boot[train\_hiv, ],   
## nvmax = 9, method = "backward")  
## 9 Variables (and intercept)  
## Forced in Forced out  
## infant\_mortality\_rate FALSE FALSE  
## fertility\_rate FALSE FALSE  
## life\_exp FALSE FALSE  
## pop\_growth\_rate FALSE FALSE  
## urban\_pop FALSE FALSE  
## youth\_literacy\_rate FALSE FALSE  
## aids\_death\_rate FALSE FALSE  
## mother\_child\_hiv\_transmis\_rate FALSE FALSE  
## per\_child\_under\_poverty FALSE FALSE  
## 1 subsets of each size up to 9  
## Selection Algorithm: backward  
## infant\_mortality\_rate fertility\_rate life\_exp pop\_growth\_rate  
## 1 ( 1 ) " " " " " " " "   
## 2 ( 1 ) " " "\*" " " " "   
## 3 ( 1 ) " " "\*" " " " "   
## 4 ( 1 ) " " "\*" " " " "   
## 5 ( 1 ) " " "\*" " " " "   
## 6 ( 1 ) "\*" "\*" " " " "   
## 7 ( 1 ) "\*" "\*" " " "\*"   
## 8 ( 1 ) "\*" "\*" " " "\*"   
## 9 ( 1 ) "\*" "\*" "\*" "\*"   
## urban\_pop youth\_literacy\_rate aids\_death\_rate  
## 1 ( 1 ) " " " " "\*"   
## 2 ( 1 ) " " " " "\*"   
## 3 ( 1 ) " " " " "\*"   
## 4 ( 1 ) " " " " "\*"   
## 5 ( 1 ) " " "\*" "\*"   
## 6 ( 1 ) " " "\*" "\*"   
## 7 ( 1 ) " " "\*" "\*"   
## 8 ( 1 ) "\*" "\*" "\*"   
## 9 ( 1 ) "\*" "\*" "\*"   
## mother\_child\_hiv\_transmis\_rate per\_child\_under\_poverty  
## 1 ( 1 ) " " " "   
## 2 ( 1 ) " " " "   
## 3 ( 1 ) " " "\*"   
## 4 ( 1 ) "\*" "\*"   
## 5 ( 1 ) "\*" "\*"   
## 6 ( 1 ) "\*" "\*"   
## 7 ( 1 ) "\*" "\*"   
## 8 ( 1 ) "\*" "\*"   
## 9 ( 1 ) "\*" "\*"

# Testing and determining the best model  
test\_mat\_backward <- model.matrix(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot[test\_hiv, ], method = "backward")  
val.errors3 <- rep(NA, 9)  
for (i in 1:9) {  
 coefi <- coef(forward\_subset\_hiv, id = i)  
 pred <- test\_mat\_backward[, names(coefi)] %\*% coefi  
 val.errors3[i] <- mean((hiv\_df\_boot$hiv\_infection\_rate[test\_hiv] - pred)^2)  
}  
val.errors3

## [1] 1.393887 1.192356 1.159057 1.144428 1.123322 1.116785 1.117192 1.118514  
## [9] 1.118408

which.min(val.errors3)

## [1] 6

coef(forward\_subset\_hiv, 6)

## (Intercept) infant\_mortality\_rate   
## 2.912988940 -0.008395259   
## fertility\_rate youth\_literacy\_rate   
## -0.555738245 -0.019081947   
## aids\_death\_rate mother\_child\_hiv\_transmis\_rate   
## 0.139927726 0.008088391   
## per\_child\_under\_poverty   
## 0.010227776

cat("Adjusted RSq:", backward\_summary$rsq[6], "\n")

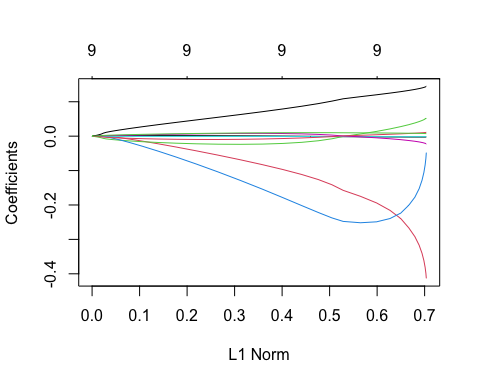
## Adjusted RSq: 0.778206

cat("MSE:", val.errors3[6], "\n")

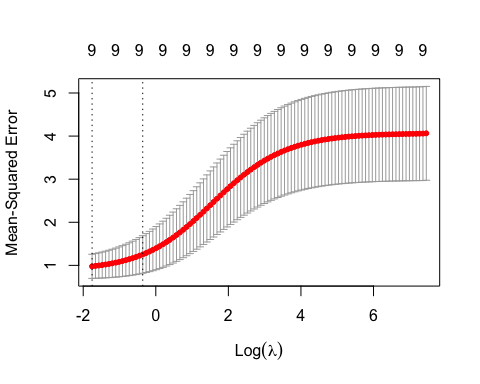
## MSE: 1.116785

**Ridge Regression**  
The model created with ridge regression has an MSE of 1.655353.

# Creating x and y vectors  
x <- model.matrix(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, hiv\_df\_boot)[, -1]  
y <- hiv\_df\_boot$hiv\_infection\_rate  
  
# Creating test and training sets  
set.seed(123)  
train <- sample(1:nrow(x), nrow(x) / 2)  
test <- (-train)  
y.test <- y[test]  
  
# Creating the model  
grid <- 10^seq(10, -2, length = 100)  
ridge\_hiv <- glmnet(x[train, ], y[train], alpha = 0, lambda = grid)  
plot(ridge\_hiv)



# Determing the best lambda value  
set.seed(123)  
cv\_out <- cv.glmnet(x[train, ], y[train], alpha = 0)  
plot(cv\_out)



bestlam\_ridge <- cv\_out$lambda.min  
bestlam\_ridge

## [1] 0.1728292

# Testing the model and calculating MSE  
ridge\_pred\_hiv <- predict(ridge\_hiv, s = bestlam\_ridge, newx = x[test, ])  
out <- glmnet(x, y, alpha = 0)  
predict(out, type = "coefficients", s = bestlam\_ridge)[1:10, ]

## (Intercept) infant\_mortality\_rate   
## 1.9107533692 -0.0035154918   
## fertility\_rate life\_exp   
## -0.2198640258 -0.0186218317   
## pop\_growth\_rate urban\_pop   
## -0.3426151115 -0.0011982380   
## youth\_literacy\_rate aids\_death\_rate   
## 0.0009869136 0.1169710784   
## mother\_child\_hiv\_transmis\_rate per\_child\_under\_poverty   
## 0.0026597813 0.0128554347

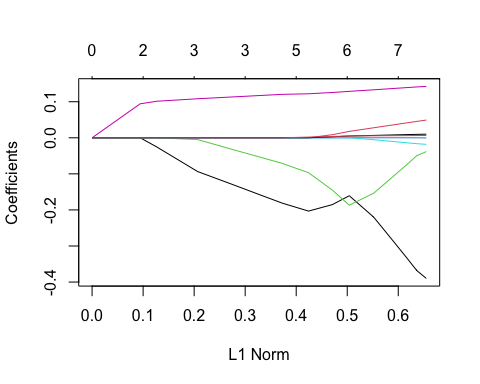
cat("MSE:", mean((ridge\_pred\_hiv - y.test)^2), "\n")

## MSE: 1.655353

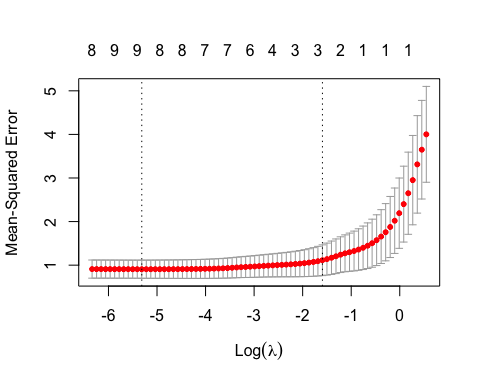
**Lasso**  
The model created with the lasso method has an MSE of 1.586618.

# Creating the model  
lasso\_hiv <- glmnet(x[train, ], y[train], alpha = 1, lambda = grid)  
plot(lasso\_hiv)

## Warning in regularize.values(x, y, ties, missing(ties), na.rm = na.rm):  
## collapsing to unique 'x' values



# Determining the best lambda value  
set.seed(123)  
cv\_out2 <- cv.glmnet(x[train, ], y[train], alpha = 1)  
plot(cv\_out2)



bestlam\_lasso <- cv\_out2$lambda.min  
bestlam\_lasso

## [1] 0.004922238

# Testing the model and calculating MSE  
lasso\_pred\_hiv <- predict(lasso\_hiv, s = bestlam\_lasso, newx = x[test, ])  
out2 <- glmnet(x, y, alpha = 1)  
predict(out2, type = "coefficients", s = bestlam\_lasso)[1:10, ]

## (Intercept) infant\_mortality\_rate   
## 1.898350049 -0.005324432   
## fertility\_rate life\_exp   
## -0.445373207 0.006512949   
## pop\_growth\_rate urban\_pop   
## -0.123410016 0.000000000   
## youth\_literacy\_rate aids\_death\_rate   
## -0.015631559 0.138645679   
## mother\_child\_hiv\_transmis\_rate per\_child\_under\_poverty   
## 0.008562343 0.009742525

cat("MSE:", mean((lasso\_pred\_hiv - y.test)^2), "\n")

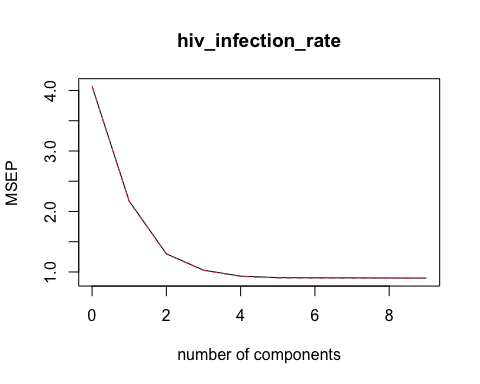
## MSE: 1.586618

**Partial Least Squares (PLS)**  
The model created with the partial least squares method has an MSE of 1.587266.

# Creating the model  
set.seed(123)  
pls\_hiv <- plsr(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot, subset = train, scale = TRUE, validation = "CV")  
summary(pls\_hiv)

## Data: X dimension: 616 9   
## Y dimension: 616 1  
## Fit method: kernelpls  
## Number of components considered: 9  
##   
## VALIDATION: RMSEP  
## Cross-validated using 10 random segments.  
## (Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps  
## CV 2.017 1.473 1.140 1.015 0.9648 0.9519 0.9502  
## adjCV 2.017 1.471 1.138 1.012 0.9619 0.9492 0.9474  
## 7 comps 8 comps 9 comps  
## CV 0.9498 0.9486 0.9481  
## adjCV 0.9470 0.9458 0.9453  
##   
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps  
## X 38.49 73.96 81.31 84.97 89.99 94.80  
## hiv\_infection\_rate 49.04 70.36 76.84 79.49 80.04 80.13  
## 7 comps 8 comps 9 comps  
## X 97.48 99.55 100.00  
## hiv\_infection\_rate 80.20 80.25 80.27

# Determining the best M value and calculating MSE  
validationplot(pls\_hiv, val.type = "MSEP")



# The lowest cross-validation error occurs when M = 5 partial least squares directions are used.   
pls\_pred\_hiv <- predict(pls\_hiv, x[test, ], ncomp = 5)  
pls\_fit\_hiv <- plsr(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot, subset = train, scale = TRUE, ncomp = 5)  
summary(pls\_fit\_hiv)

## Data: X dimension: 616 9   
## Y dimension: 616 1  
## Fit method: kernelpls  
## Number of components considered: 5  
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps 5 comps  
## X 38.49 73.96 81.31 84.97 89.99  
## hiv\_infection\_rate 49.04 70.36 76.84 79.49 80.04

coef(pls\_fit\_hiv)

## , , 5 comps  
##   
## hiv\_infection\_rate  
## infant\_mortality\_rate -0.10814010  
## fertility\_rate -0.28171048  
## life\_exp 0.33081303  
## pop\_growth\_rate -0.29326391  
## urban\_pop 0.00452169  
## youth\_literacy\_rate -0.35790432  
## aids\_death\_rate 2.08548682  
## mother\_child\_hiv\_transmis\_rate 0.16845244  
## per\_child\_under\_poverty 0.15944112

cat("MSE:", mean((pls\_pred\_hiv - y.test)^2), "\n")

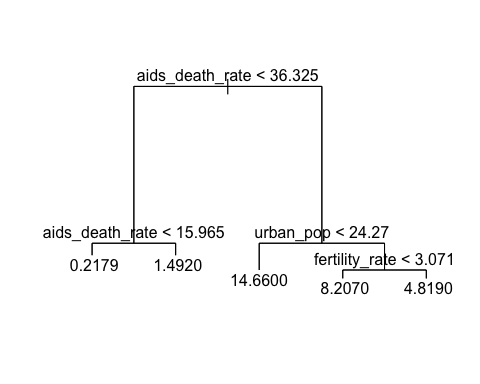
## MSE: 1.587266

**Regression Tree**  
The regression tree created with cross validation has an MSE of 0.8744749.

# Creating a training data set  
set.seed(123)  
train <- sample(1:nrow(hiv\_df\_boot), nrow(hiv\_df\_boot) / 2)  
  
# Creating the model  
tree\_hiv <- tree(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, hiv\_df\_boot, subset = train)  
summary(tree\_hiv)

##   
## Regression tree:  
## tree(formula = hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate +   
## life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate +   
## aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty,   
## data = hiv\_df\_boot, subset = train)  
## Variables actually used in tree construction:  
## [1] "aids\_death\_rate" "urban\_pop" "fertility\_rate"   
## Number of terminal nodes: 5   
## Residual mean deviance: 0.3543 = 216.5 / 611   
## Distribution of residuals:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -6.8420 -0.1879 -0.1099 0.0000 0.1082 4.1530

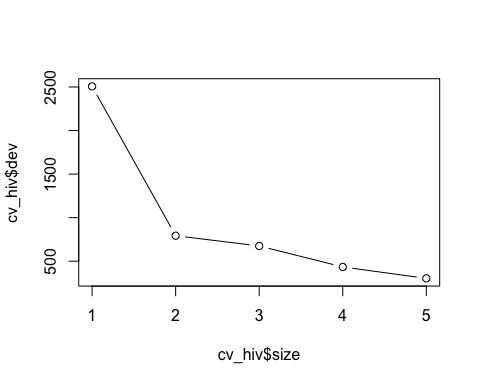
plot(tree\_hiv)  
text(tree\_hiv, pretty = 0)



# Prune the tree with cross validation  
cv\_hiv <- cv.tree(tree\_hiv)  
cv\_hiv

## $size  
## [1] 5 4 3 2 1  
##   
## $dev  
## [1] 302.8638 434.2763 675.0429 792.6930 2506.5022  
##   
## $k  
## [1] -Inf 93.90304 134.19955 299.35920 1753.62875  
##   
## $method  
## [1] "deviance"  
##   
## attr(,"class")  
## [1] "prune" "tree.sequence"

plot(cv\_hiv$size, cv\_hiv$dev, type = "b")



# The best number of terminal nodes was already selected by the unpruned tree, at 5.  
yhat <- predict(tree\_hiv, newdata = hiv\_df\_boot[-train, ])  
hiv\_test <- unlist(hiv\_df\_boot[-train, "hiv\_infection\_rate"])  
cat("MSE:", mean((yhat - hiv\_test)^2), "\n")

## MSE: 0.8744749

**Bagging**  
The regression tree created with bagging has an MSE of 0.07755276.

# Creating the model  
set.seed(123)  
bag\_hiv <- randomForest(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot, subset = train, mtry = 9, importance = TRUE)  
bag\_hiv

##   
## Call:  
## randomForest(formula = hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot, mtry = 9, importance = TRUE, subset = train)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 9  
##   
## Mean of squared residuals: 0.09357584  
## % Var explained: 97.69

importance(bag\_hiv)

## %IncMSE IncNodePurity  
## infant\_mortality\_rate 13.616166 124.74003  
## fertility\_rate 10.180432 23.54548  
## life\_exp 9.992152 108.80376  
## pop\_growth\_rate 11.986066 33.54232  
## urban\_pop 14.014906 119.03309  
## youth\_literacy\_rate 16.104698 44.77552  
## aids\_death\_rate 74.378480 1984.55314  
## mother\_child\_hiv\_transmis\_rate 23.805099 45.66542  
## per\_child\_under\_poverty 12.448492 32.58985

# Testing the model and calculating MSE  
yhat\_bag <- predict(bag\_hiv, newdata = hiv\_df\_boot[-train, ])  
cat("MSE:", mean((yhat\_bag - hiv\_test)^2), "\n")

## MSE: 0.07755176

**Random Forests**  
The regression tree created with random forests has an MSE of 0.118209.

# Creating the model  
set.seed(123)  
rf\_hiv <- randomForest(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot, subset = train, mtry = 3, importance = TRUE)  
rf\_hiv

##   
## Call:  
## randomForest(formula = hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot, mtry = 3, importance = TRUE, subset = train)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 3  
##   
## Mean of squared residuals: 0.1168581  
## % Var explained: 97.12

importance(rf\_hiv)

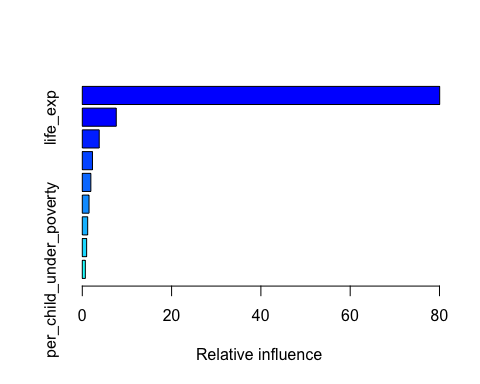
## %IncMSE IncNodePurity  
## infant\_mortality\_rate 7.218622 167.13185  
## fertility\_rate 6.259991 104.04900  
## life\_exp 8.611361 333.75934  
## pop\_growth\_rate 6.351998 127.73579  
## urban\_pop 13.701107 87.23218  
## youth\_literacy\_rate 7.348407 169.03632  
## aids\_death\_rate 28.872739 1053.24399  
## mother\_child\_hiv\_transmis\_rate 14.034151 233.26366  
## per\_child\_under\_poverty 11.909348 218.73305

# Testing the model and calculating MSE  
yhat\_rf <- predict(rf\_hiv, newdata = hiv\_df\_boot[-train, ])  
cat("MSE:", mean((yhat\_rf - hiv\_test)^2), "\n")

## MSE: 0.118209

**Boosting**  
The regression tree created with boosting has an MSE of 0.1268387.

# Creating the model  
set.seed(123)  
boost\_hiv <- gbm(hiv\_infection\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + aids\_death\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot[train, ], distribution = "gaussian", n.trees = 5000, interaction.depth = 4)  
summary(boost\_hiv)



## var rel.inf  
## aids\_death\_rate aids\_death\_rate 80.0433314  
## life\_exp life\_exp 7.6051348  
## youth\_literacy\_rate youth\_literacy\_rate 3.7877096  
## pop\_growth\_rate pop\_growth\_rate 2.2789846  
## urban\_pop urban\_pop 1.9165094  
## mother\_child\_hiv\_transmis\_rate mother\_child\_hiv\_transmis\_rate 1.5101233  
## fertility\_rate fertility\_rate 1.2074696  
## infant\_mortality\_rate infant\_mortality\_rate 0.9728071  
## per\_child\_under\_poverty per\_child\_under\_poverty 0.6779302

# Testing the model and calculating the MSE  
yhat\_boost <- predict(boost\_hiv, newdata = hiv\_df\_boot[-train, ], n.trees = 5000)  
cat("MSE:", mean((yhat\_boost - hiv\_test)^2), "\n")

## MSE: 0.1268387

**Overview of Models**

# Table with models and relative MSEs  
quantitative\_overview <- data.frame(  
 Method = c("Best Subset", "Forward Selection", "Backward Selection", "Ridge Regression", "Lasso", "PLS", "Regression Tree", "Bagging", "Random Forests", "Boosting"),   
 MSE = c(1.116786, 1.116785, 1.116785, 1.655353, 1.586618, 1.587266, 0.8744749, 0.07755276, 0.118209, 0.1268387)  
)  
quantitative\_table <- kable(quantitative\_overview, "markdown") %>%   
 kable\_styling(full\_width = FALSE) %>%  
 column\_spec(1, bold = TRUE)

## Warning in kable\_styling(., full\_width = FALSE): Please specify format in  
## kable. kableExtra can customize either HTML or LaTeX outputs. See  
## https://haozhu233.github.io/kableExtra/ for details.

## Warning in column\_spec(., 1, bold = TRUE): Please specify format in kable.  
## kableExtra can customize either HTML or LaTeX outputs. See  
## https://haozhu233.github.io/kableExtra/ for details.

quantitative\_table

| Method | MSE |
| --- | --- |
| Best Subset | 1.1167860 |
| Forward Selection | 1.1167850 |
| Backward Selection | 1.1167850 |
| Ridge Regression | 1.6553530 |
| Lasso | 1.5866180 |
| PLS | 1.5872660 |
| Regression Tree | 0.8744749 |
| Bagging | 0.0775528 |
| Random Forests | 0.1182090 |
| Boosting | 0.1268387 |

# Section 3: How can the indicators be used to predict AIDS deaths?

**Principal Components Regression**  
A PCR model on the data has an MSE of 52.88973, and explains 84.87% of the variance.

# Creating x and y vectors  
x <- model.matrix(aids\_death\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + hiv\_infection\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, hiv\_df\_boot)[, -1]  
y <- hiv\_df\_boot$aids\_death\_rate  
  
# Creating test and training sets  
set.seed(123)  
train <- sample(1:nrow(x), nrow(x) / 2)  
test <- (-train)  
y.test <- y[test]  
  
# Creating the model  
set.seed(123)  
pcr\_aids <- pcr(aids\_death\_rate ~ infant\_mortality\_rate + fertility\_rate + life\_exp + pop\_growth\_rate + urban\_pop + youth\_literacy\_rate + hiv\_infection\_rate + mother\_child\_hiv\_transmis\_rate + per\_child\_under\_poverty, data = hiv\_df\_boot, scale = TRUE, validation = "CV")  
summary(pcr\_aids)

## Data: X dimension: 1233 9   
## Y dimension: 1233 1  
## Fit method: svdpc  
## Number of components considered: 9  
##   
## VALIDATION: RMSEP  
## Cross-validated using 10 random segments.  
## (Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps  
## CV 16.3 15.25 8.038 7.651 7.630 6.414 6.340  
## adjCV 16.3 15.25 8.034 7.647 7.628 6.409 6.336  
## 7 comps 8 comps 9 comps  
## CV 6.309 6.280 6.239  
## adjCV 6.305 6.276 6.235  
##   
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps  
## X 56.87 72.97 81.72 88.45 93.10 96.70 98.87  
## aids\_death\_rate 12.52 76.04 78.29 78.58 84.87 85.23 85.39  
## 8 comps 9 comps  
## X 99.72 100.00  
## aids\_death\_rate 85.53 85.74

# Determining the best M value and calculating MSE  
validationplot(pcr\_aids, val.type = "MSEP")



# The lowest cross-validation error occurs when M = 5 partial least squares directions are used.   
pcr\_pred\_aids <- predict(pcr\_aids, x[test, ], ncomp = 5)  
pcr\_fit\_aids <- pcr(y ~ x, scale = TRUE, ncomp = 5)  
summary(pcr\_fit\_aids)

## Data: X dimension: 1233 9   
## Y dimension: 1233 1  
## Fit method: svdpc  
## Number of components considered: 5  
## TRAINING: % variance explained  
## 1 comps 2 comps 3 comps 4 comps 5 comps  
## X 56.87 72.97 81.72 88.45 93.10  
## y 12.52 76.04 78.29 78.58 84.87

coef(pcr\_fit\_aids)

## , , 5 comps  
##   
## y  
## infant\_mortality\_rate 1.4753177  
## fertility\_rate 0.7749042  
## life\_exp -4.4565081  
## pop\_growth\_rate 0.6232586  
## urban\_pop 0.3087891  
## youth\_literacy\_rate 2.3969200  
## hiv\_infection\_rate 11.7871813  
## mother\_child\_hiv\_transmis\_rate -1.9731541  
## per\_child\_under\_poverty -0.2372773

cat("MSE:", mean((pcr\_pred\_aids - y.test)^2), "\n")

## MSE: 52.88973