Homework4

Question 10.1

For the crime dataset, I start with the original PCA/Linear regression and proceed to regression tree model followed by regression forest model.

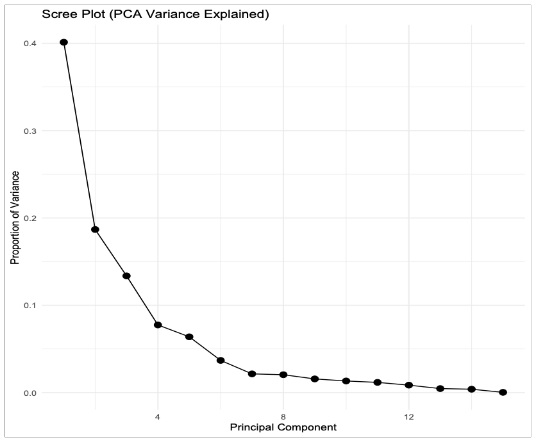
1. **PCA Model – Factors and Coefficients:**

   This model handles the problem of multicollinearity by transforming the 15 correlated predictors into uncorrelated components.

   Implementation: The process involves scaling the data, performing the decompositions, and then running a linear regression on the top components. Uses prcomp with scale. = TRUE to ensure all units are treated equally. A linear model is built on the top 6 components, and the coefficients are then transformed back into original variable units for interpretability.

   Analysis & Results:

   - **Scree Plot Analysis:** The scree plot shows the "proportion of variance" explained by each component. The summary of the PCA indicates that 6 components are sufficient to explain the vast majority of the data's variance.



   - **Dimensionality Reduction:** By using only 6 components instead of 15 variables, we simplify the model while retaining high predictive power.

- **Coefficient Results:** The code outputs “Original variable coefficients,” which allows the model to be used on raw, unscaled data for future predictions.

Qualitative Takeaway:

- **Information Compression:** The PCA reveals that approximately 6 components can capture about 90% of the information found in the original 15 variables. This suggest that much of the “raw” crime data is redundant, and the same predictive power can be achieved with a much simpler, transformed feature set.

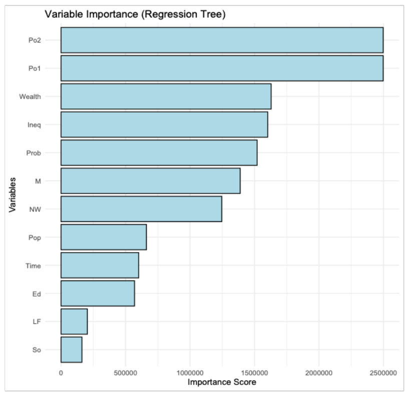
2. **Regression Tree Model:**

The regression tree uses a hierarchical “if-then” structure to split the data into groups with similar crime rates based on variance reduction.

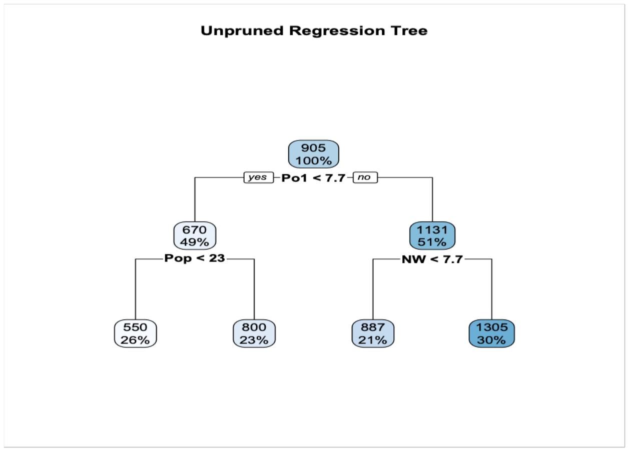
Implementation: The tree is grown fully and then “pruned” to its most statistically significant splits. The “rpart” function uses method = “anova” for a quantitative response variable (Crime). The script identifies the optimal Complexity Parameter (CP) that minimizes cross-validated error (xerror) to prune the tree back to its most reliable branches. Standard “rpart.plot” is used for the tree structure, and a ggplot2 bar chart displays variable importance.

Analysis & Results:

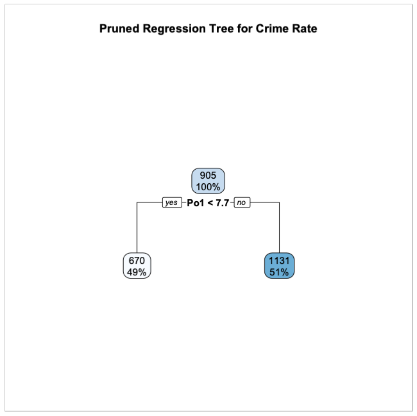
- **Variable Importance:** The “importance Score” plot generated by the code ranks variables like **Po1** (Police expenditure) and **Wealth** at the top.



- **Pruning Effect:** The "Unpruned Tree" is highly complex, while the "Pruned Tree" is much smaller. In many runs with this specific dataset, the pruned tree simplifies significantly because the small sample size (n=47) makes complex branches statistically unreliable.



- **Takeaway:** This model is excellent for interpretability, clearly showing that police spending is the primary divider between high-crime and low-crime areas.



Qualitative Takeaways:

- **Dominant Predictor:** The root node (first split) for this data consistently involves Po1 (Police expenditure in 1960). This indicates that police spending is the single most informative factor when initially predicting crime rates in this model.
- **Model Simplification via Pruning:** Pruning is essential to prevent overfitting. While an unpruned tree may capture minor noise, the pruned tree identifies that only a few key variables (like Po1 and NW) are statistically reliable for prediction given the small sample size (n = 47).

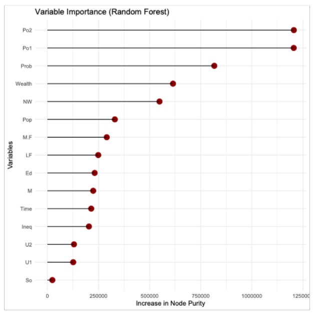
3. **Random Forest Model:**

The random forest addresses the instability of a single tree by averaging the prediction of 500 different trees.

Implementation: The “randomforest” function is initialized with “ntree = 500” and “importance = TRUE”. The script outputs the percentage of variance explained (Pseudo R^2) and provides a detailed dot plot of variable importance using ggplot2. The result calculates the final variance explained at the 500th tree to assess overall model fit.

Analysis & Results:

- **Stability:** The Random Forest is more robust than a single tree. It accounts for more variables by randomly selecting subsets of predictors for each tree.
- **Variable Importance (IncNodePurity):** The ggplot dot plot shows which variables contribute most to the “purity” of the nodes across the entire forest. While **Po1** remains a top predictor, other like **Ineq** (Income Equality) and **Ed** (Education) show high relative importance.



- **Performance:** The code calculates the final "Variance Explained" at the 500th tree. For the "uscrime" dataset, this pseudo-R-squared value typically provides the most accurate estimation of predictive capability among these three models.

Qualitative Takeaways:

- **Hidden Variable Importance:** While a single tree might ignore certain variables due to correlation, the Random Forest reveals a broader range of important predictors. For example, variables like Wealth and Ineq (Income inequality) often show high "Increase in Node Purity" in a forest, even if they aren't the primary splits in a single tree.
- **Stability over Precision:** A Random Forest is "% Variance Explained" is a more realistic measure of predictive power than a single complex tree's R^2 because it averages out the random errors of individual trees, making it much more robust against outliers in the small crime dataset.

Question 10.2

One of the examples I came across while discussing a problem with my friend, is predicting **Guest No-Shows** in any large hospitality brands like Hyatt Hotel. I feel this fits into a classic example of logistic regression.
In this scenario, the dependent variable is a binary – will the guest show up? (1 = Yes, 0 = No). Hyatt needs to predict this probability to manage overbooking and ensure maximum occupancy without turning away guest who arrive.

1. **The Model Inputs (Independent Variables):** To determine the probability of a no-show, the model analyze several factors:
    - **Booking Lead Time:** How many days in advance was the room booked? (Longer lead time often correlate with higher cancellation/no-show rates).
    - **Rate Type:** Is it a "Non-Refundable" rate or a "Flexible" rate?
    - **Loyalty Status:** Is the guest a World of Hyatt "Globalist" or a first-time guest? (Loyal members are statistically less likely to no-show)
    - **Booking Channel:** Was it booked directly via Hyatt.com, through a corporate travel agent, or an Expedia-style third party?
    - **Group vs Individual:** Is the room part of a large wedding block or a single business traveler?
2. **The Output: Probability and Decision**
   The logistic regression model doesn't just say "Yes" or "No". It outputs a probability, such as 0.15 (15% change of no-show).
    - **The Sigmoid Curve:** If a guest has a flexible rate, booked 6 months ago, and has no loyalty status, their data point will move toward the "0" (No-show) end of the S-curve.

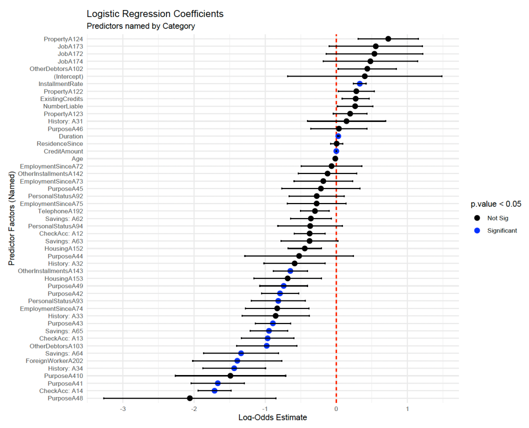
- **The Threshold:** Hyatt management sets a decision threshold. If the model predicts a specific room has a >20% chance of being a no-show, they may authorize the system to "overbook" that room-selling it to someone else to ensure the hotel stays full.

In this case Hyatt might use logistic regression instead of linear regression because a linear model might give nonsensical results, like a "-10% chance of showing up". Logistic regression forces the answer to stay between 0 and 100%, making it a perfect tool for risk management.

<u>Question 10.3</u>

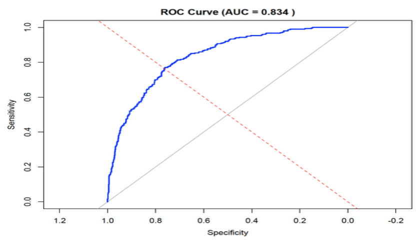
This part of the assignment implements a logistic regression model using the German credit dataset to predict credit risk.

1. **Implementation:** The script implements a logistic regression model using the German Credit dataset to predict credit risk.
    - **Data Pre-Processing:** The response variable is re-coded where 1(Good) becomes 0 and 2 (Bad) becomes 1. This aligns the model to predict the probability of a "Bad" credit event.
    - **Model Selection:** A Generalized Linear Model (glm) with a binomial link function is used for the primary analysis.
    - **Validation:** The script utilizes 10-fold cross-validation via the "caret" package to ensure the model's performance is stable across different subsets of data.
    - **Visualization:** Multiple diagnostic plots were generated, including ROC curves to measure the Area Under the Curve (AUC), Confusion Matrices to visualize classification error, and Coefficient plots to identify significant predictors.

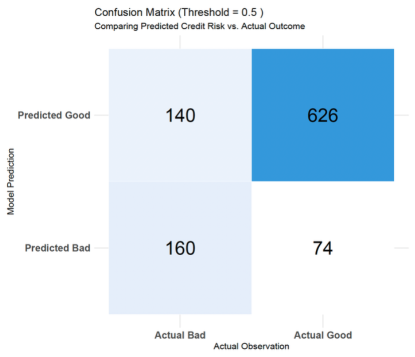


2. **Analysis and Results:** The model's performance is evaluated using two different classification thresholds to compare business outcomes.

A. **Performance at 0.5 threshold (standard):** At a standard 0.5 threshold, the model treats False Positivies and False Negatives as equally important.
   - **ROC (AUC):** Approximately 0.834, indicating a “Fair to Good” ability to distinguish between good and bad risks.

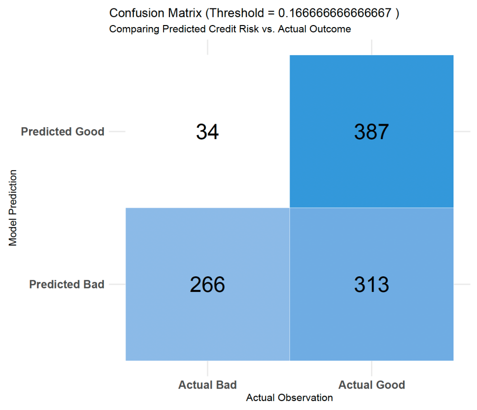


   - **Observation:** While accuracy may be higher, this threshold typically results in a higher number of False Negatives (missing bad risks), which is financially dangerous for a bank.


Confusion Matrix (Threshold = 0.5)
Comparing Predicted Credit Risk vs. Actual Outcome
Model Prediction
Actual Observation

	Actual Bad	Actual Good
Predicted Good	140	626
Predicted Bad	160	74

B. **Performance at 0.167 threshold (Optimized):** When the threshold is lowered to 0.167, the model becomes more “conservative” in flagging bad risks.
   - **Sensitivity (Recall):** Increases significantly. The model catches the vast majority of “Bad” applicants.
   - **Specificity:** Decreases. The model incorrectly flags more “Good” applicants as risky.


Confusion Matrix (Threshold = 0.166666666666667)
Comparing Predicted Credit Risk vs. Actual Outcome
Model Prediction
Actual Observation

	Actual Bad	Actual Good
Predicted Good	34	387
Predicted Bad	266	313

- **Business Impact:** This minimizes the total financial loss by prioritizing the avoidance of loan defaults over the loss of potential interest from rejected safe customers.

3. **Choice of 0.167 Threshold:** The choice of 0.167 (derived as 1/6) is a calculated business decision based on the asymmetric costs of miss-classification defined in the assignment.

   The Cost Ratio:

   The problem specifies a 5:1 cost ratio:

   - **Cost of a False Positive (C_fp)**: 1 (The lost opportunity of interest from a good customer).
   - **Cost of a False Negative (C_fn)**: 5 (The actual loss of the loan principal from a bad customer).

   The Mathematical Derivation:

   To find the threshold ($P^\wedge *$) that minimizes the Total Expected Cost, we use the following formula:

$$p^* = \frac{C_{fp}}{C_{fp} + C_{fn}}$$

   Plugging in our values:

$$p^* = \frac{1}{1+5} = \frac{1}{6} \approx 0.1666 \ldots.$$

4. **CV Implementation and Comparison:** The script uses “caret” package to execute a 10-fold cross-validation. In terms of the raw numbers (ROC, Sensitivity, Specificity), the results are very similar – both hovering around a 0.8 ROC. However, in data science, the CV model is considered “better” for the following reasons:
   - **Reduced Bias**: When we calculate performance on the same data used to train the model (without CV), the results are biased because the model has already “seen” the answers. CV removes this bias.
   - **Stability assessment**: The CV results (ROC: 0.7797) prove that the model is stable. If the CV ROC had dropped significantly (e.g. to 0.60) compared to the training ROC, it would have indicated that our model was overfitting and would fail in the real world.
   - **Confidence in Threshold**s: Because the CV results are stable, we can have much higher confidence that our 0.167 cost-optimized threshold will work on future customers to save the bank money.

5. **Strategic Conclusion:** By setting the threshold at 0.167, we ensure that the model will flag an applicant as “Bad” even if there is only a 17% probability of default. This “low bar”

for rejection is necessary because the cost of being wrong about a “Good” applicant is much lower than the cost of being wrong about a “Bad” one. The CV implementation confirms that our logistic regression is a robust and reliable tool. While the metrics didn’t “improve” in terms of higher values, the validity of those values is significantly higher because they survived 10 rounds of testing on unseen data.