LASSO and Pruned Decision Tree for Mid Atlantic Wages

Code

Hemanth Pranav Malladi

1 INTRODUCTION

The dataset contains information about 3000 workers from the Mid-Atlantic region of USA. This dataset provides Wage and related information for male workers throughout the year 2003-2009. The study investigates wage prediction in the Mid-Atlantic by implementing **LASSO regression** and a **pruned decision tree** as supervised learning models. The goal is to forecast the wage logarithm value (`logwage`) through analyzing demographic and job-type data points.

2 Description

The **Wage** data set is from the **ISLR2** package.

- **Observations**: Approximately 3,000 working individuals.
- **Features**: The Wage dataset contains seven variables including age, year and education with three levels, jobclass, health, health_ins and wage which underwent transformation.

Target population:The demographic group comprised working adults in the Mid-Atlantic area whom researchers surveyed using economic studies across different education levels and job classification categories.

Sampling strategy: The researchers implemented stratified random sampling to achieve education-based and job-type coverage of all participants.

Potential bias:The study faces limitations due to self-reported information errors together with an underrepresentation of informal workers and local conditions that reduce its power of broad-scale application.

Prediction problem: The goal of this Report is to focus on predicting the continuous log-transformed wage (`logwage`) outcome using available predictor variables.

Data splitting plan: The random data splitting included a 50/50 train-test split which was made reproducible through setting a seed value of 1.

3 Data Preparation

Hide

```
library(ISLR2)
library(tidyverse)


W_data <- data.frame(Wage)  # keep original for EDA
W_data <- na.omit(W_data)


Wage <-W_data %>%
  mutate(logwage = log(wage)) %>%
  select(-wage)

# Split into train/test (50/50)
set.seed(1)
train_idx <- sample(seq_len(nrow(Wage)), nrow(Wage)/2)
train <- Wage[train_idx, ]
test  <- Wage[-train_idx, ]


x_train <- model.matrix(logwage ~ ., train)[, -1]
y_train <- train$logwage
x_test  <- model.matrix(logwage ~ ., test)[, -1]
y_test  <- test$logwage
```

4 Statistical Learning Strategies and Methods

4.1 Exploratory Data Analysis (EDA)

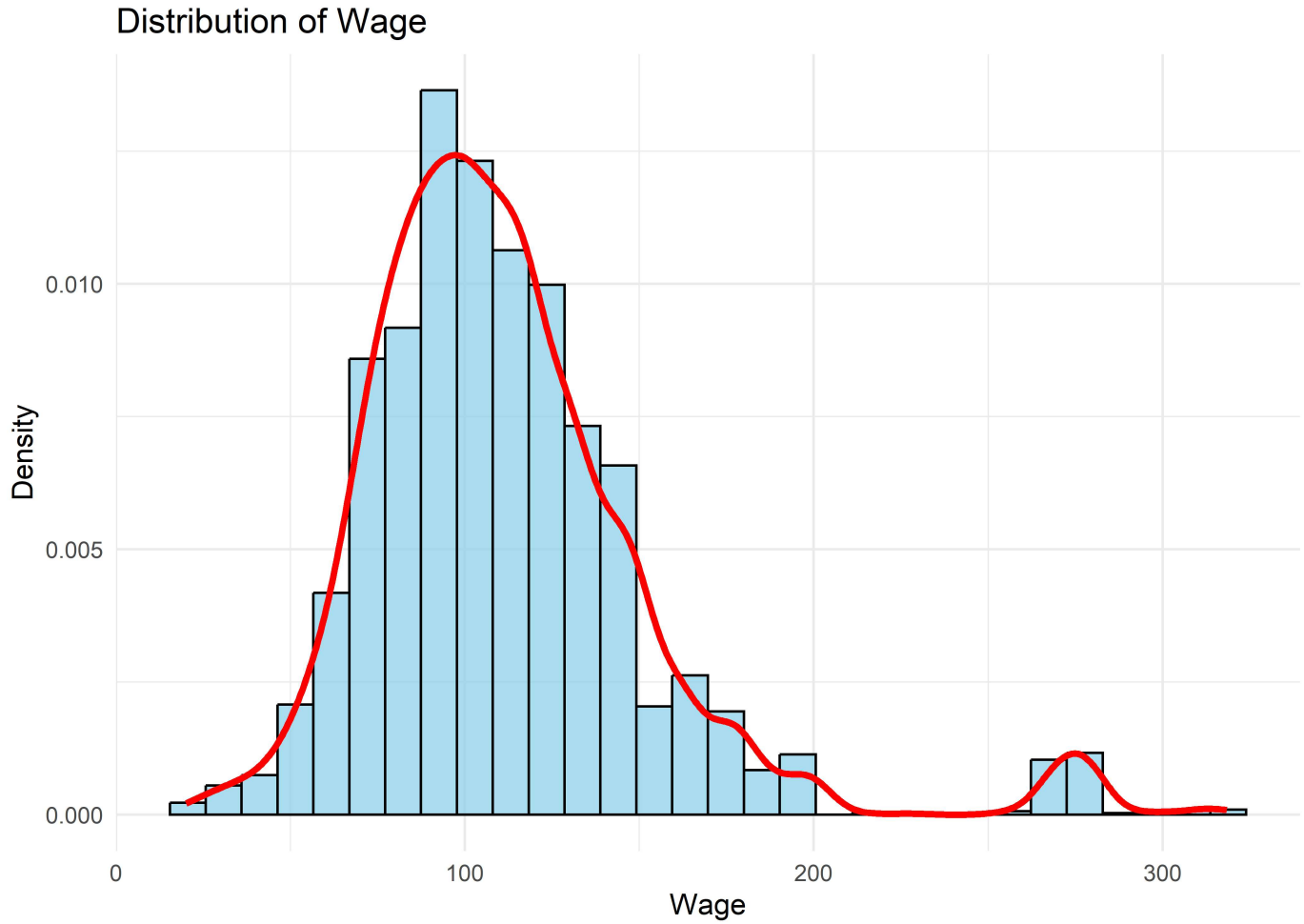
Below are visualizations and interpretations for key relationships in the full **Wage** dataset.

Distribution of Wage

Hide

```r
library(ggplot2)

ggplot(W_data, aes(x = wage)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "skyblue", color = "black", a
        lpha = 0.7) +
  geom_density(color = "red", size = 1.2) +
  labs(
    title = "Distribution of Wage",
    x = "Wage",
    y = "Density"
  ) +
  theme_minimal()
```



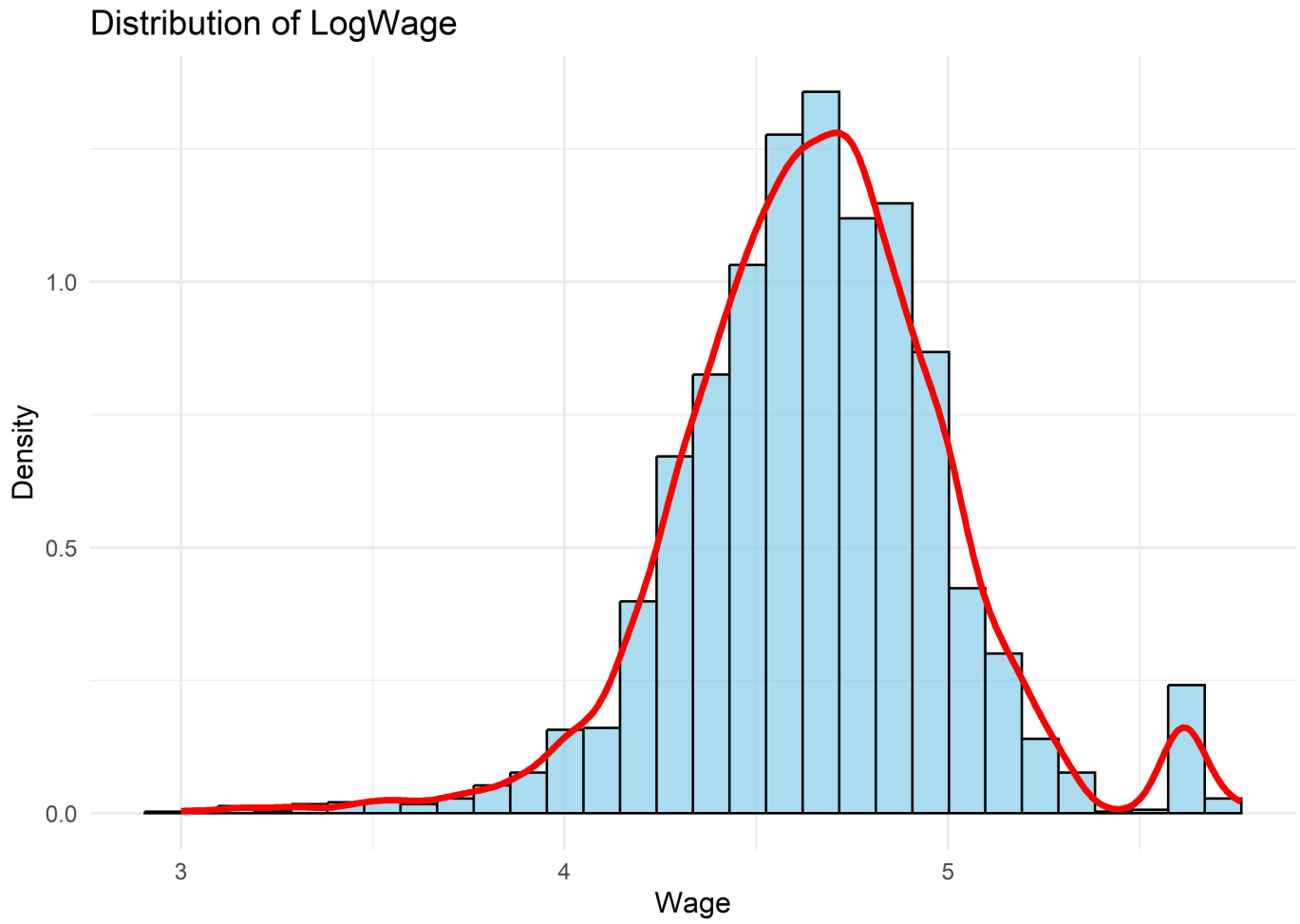
Interpretation: The wage distribution is right-skewed, with a long tail of higher earners.

Distribution of Logwage

Hide

```
library(ggplot2)


ggplot(W_data, aes(x = logwage)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "skyblue", color = "black", a
        lpha = 0.7) +
  geom_density(color = "red", size = 1.2) +
  labs(
    title = "Distribution of LogWage",
    x = "Wage",
    y = "Density"
  ) +
  theme_minimal()
```



Interpretation: The transformation of wage data through logarithm significantly decreases its right-skew characteristics in the original distribution.

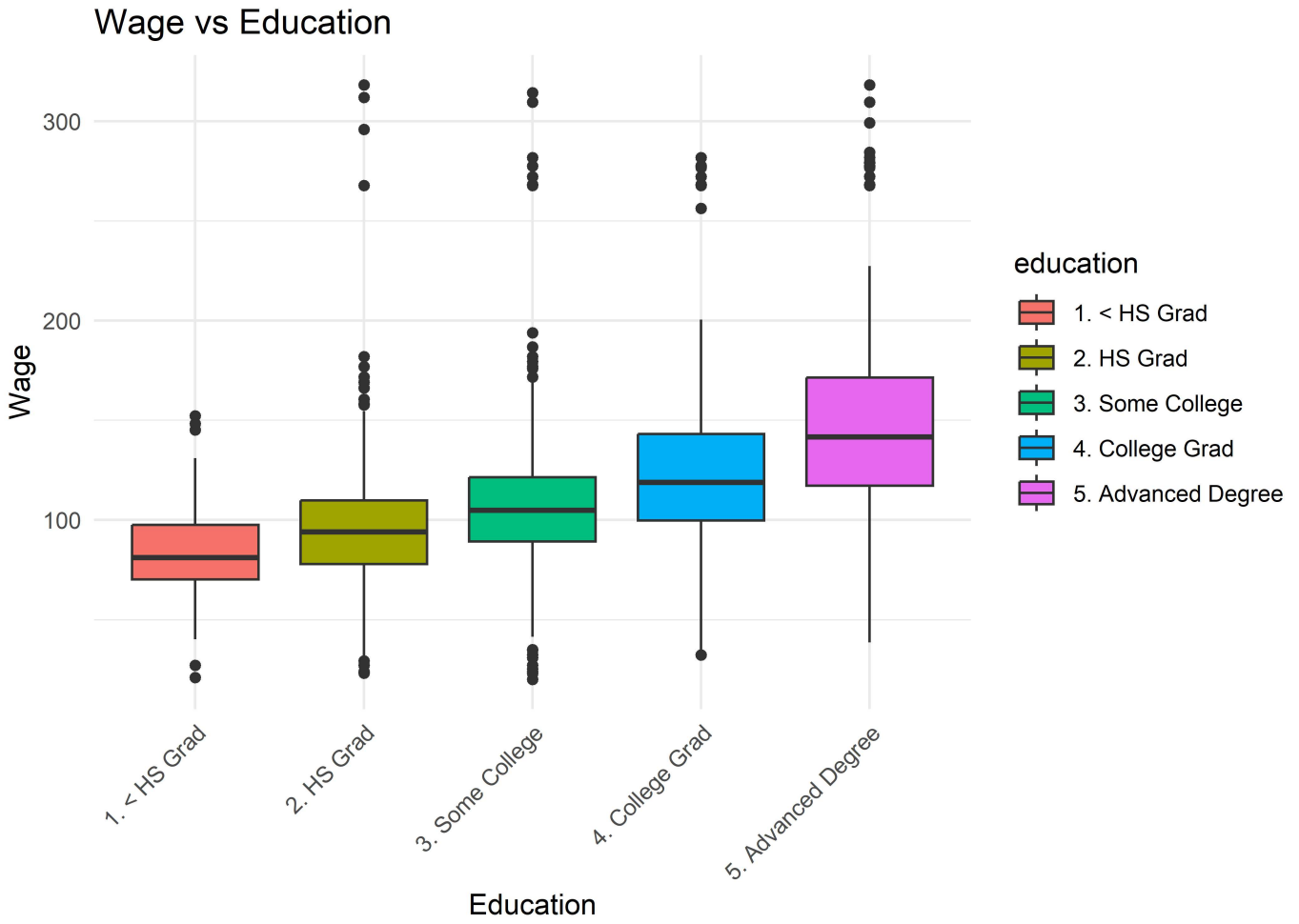
The linear model on logwage shows sufficient approximation potential because the resulting curve displays a single peak distribution.

The distribution of histogram bars shows symmetrical reduction after the peak appears on the log scale because very few workers have extreme low or high earnings. The tail behavior enhances linear methods including LASSO since it improves their assumptions about stability and homoscedasticity.

Wage vs Education level

Hide

```
# Wage vs Education Level
ggplot(W_data, aes(x = education, y = wage, fill = education)) +
  geom_boxplot() +
  ggtitle("Wage vs Education") +
  xlab("Education") +
  ylab("Wage") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

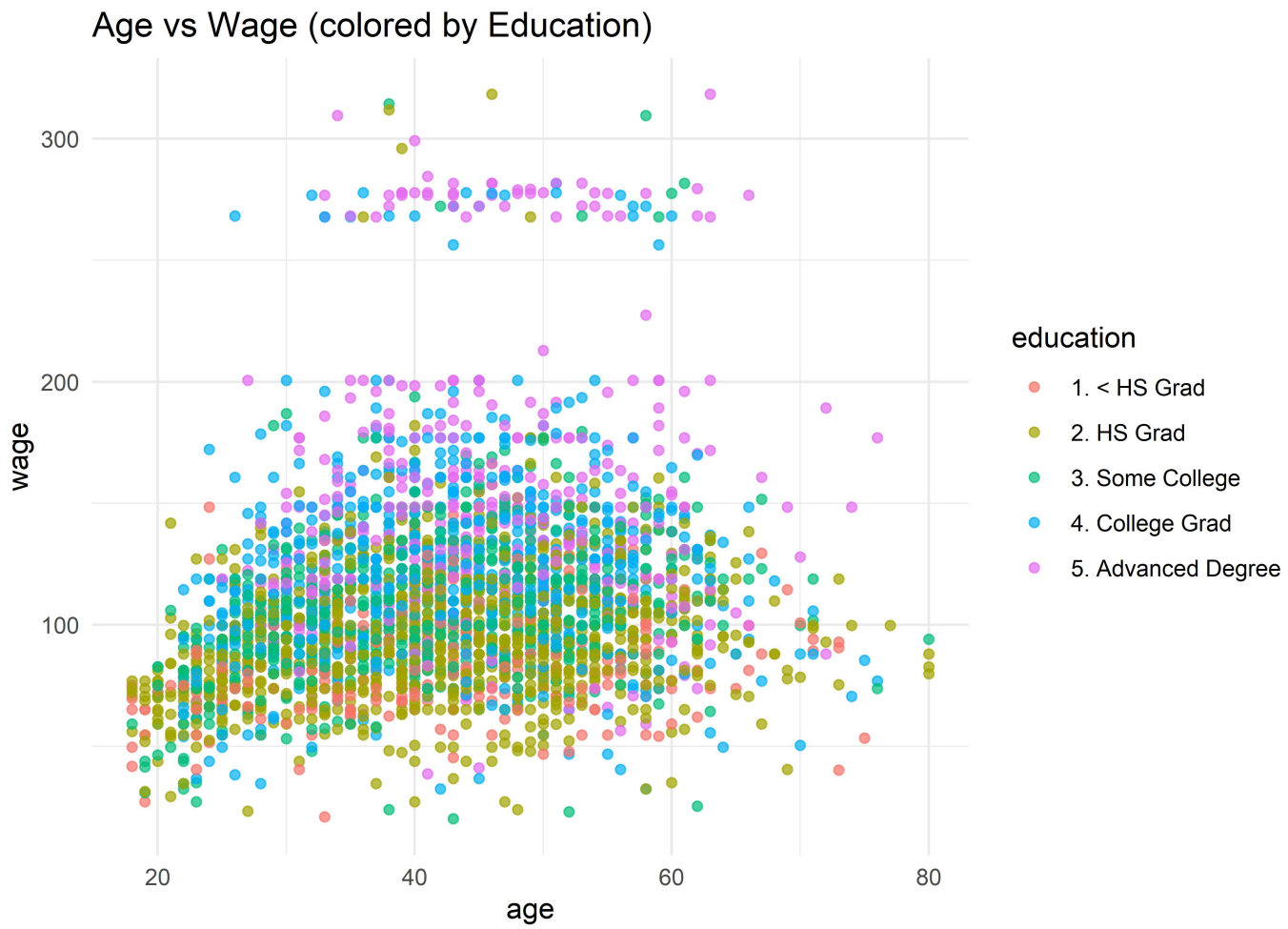


Interpretation: People who earn higher academic degrees tend to receive higher salaries in the middle range. The wages of highly educated individuals show increased dispersion across salaries as the level of variation expands.

Scatter: Age vs Wage colored by Education

Hide

```
ggplot(W_data, aes(x = age, y = wage, color = education)) +
  geom_point(alpha = 0.7) +
  ggtitle("Age vs Wage (colored by Education)") +
  theme_minimal()
```

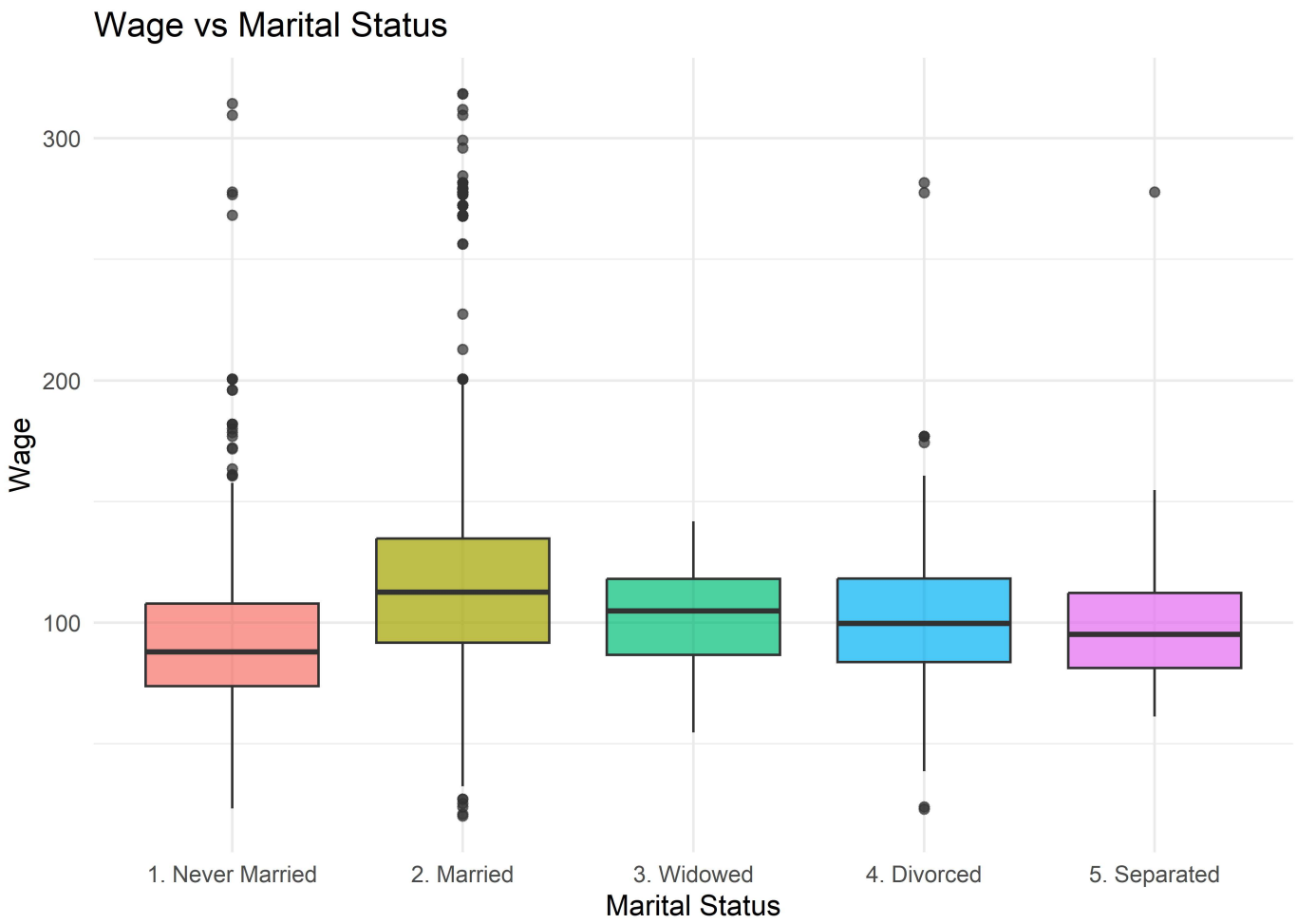


Interpretation: Individual wages increase throughout life because workers gain more professional experience with time. Across all ages the wages tend to be higher when people have more education according to the color scheme

Wage vs Marital Status

Hide

```
ggplot(W_data, aes(x = maritl, y = wage, fill = maritl)) +
  geom_boxplot(alpha = 0.7) +
  labs(
    title = "Wage vs Marital Status",
    x = "Marital Status",
    y = "Wage"
  ) +
  theme_minimal() +
  theme(legend.position = "none")
```

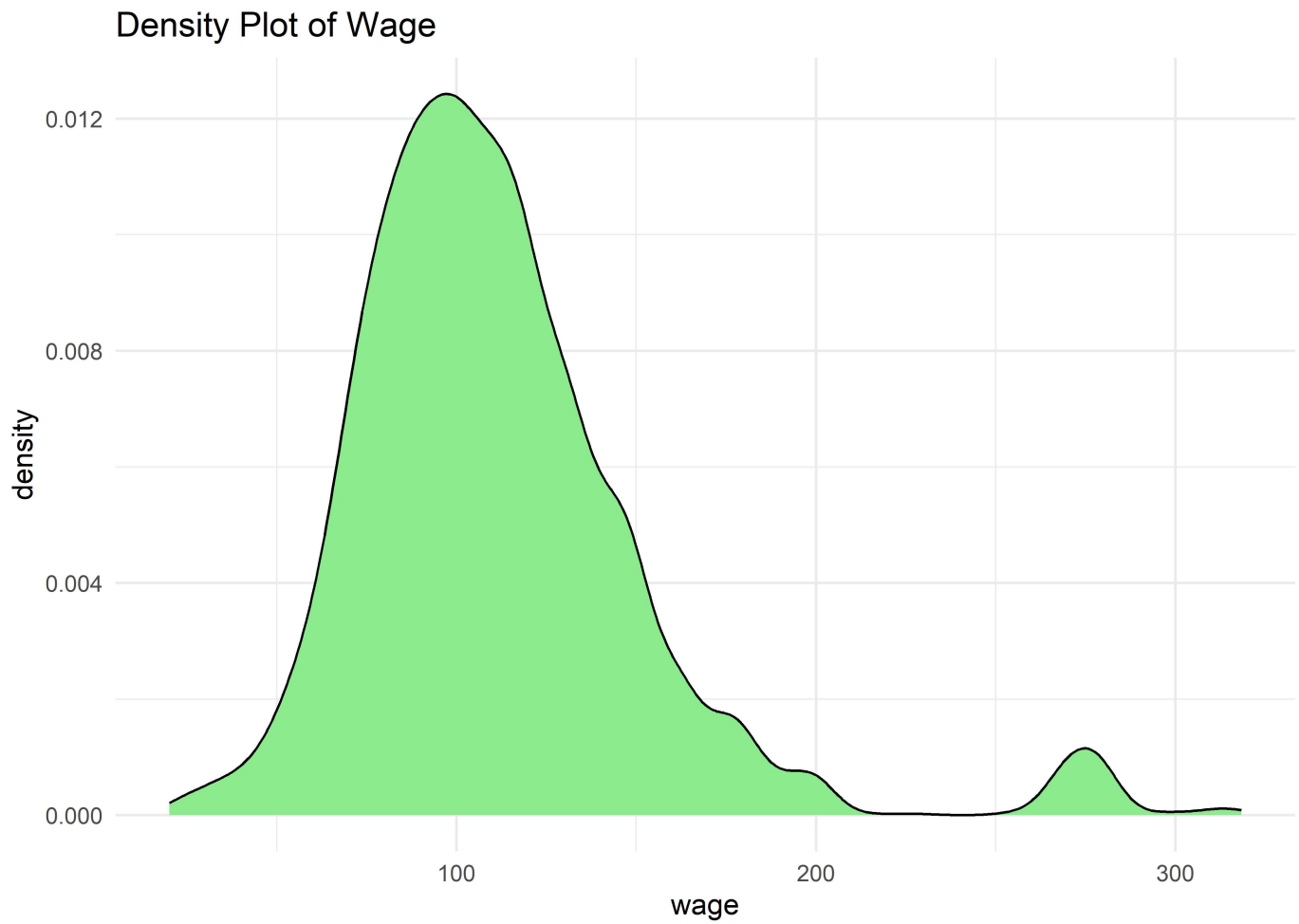


Interpretation: The median pay of married workers surpasses the earnings of single or divorced personnel because marriage-age differences and employment stability patterns may contribute to this pattern.

Density of wages

Hide

```
ggplot(W_data, aes(x = wage)) +
  geom_density(fill = "lightgreen") +
  ggtitle("Density Plot of Wage") +
  theme_minimal()
```

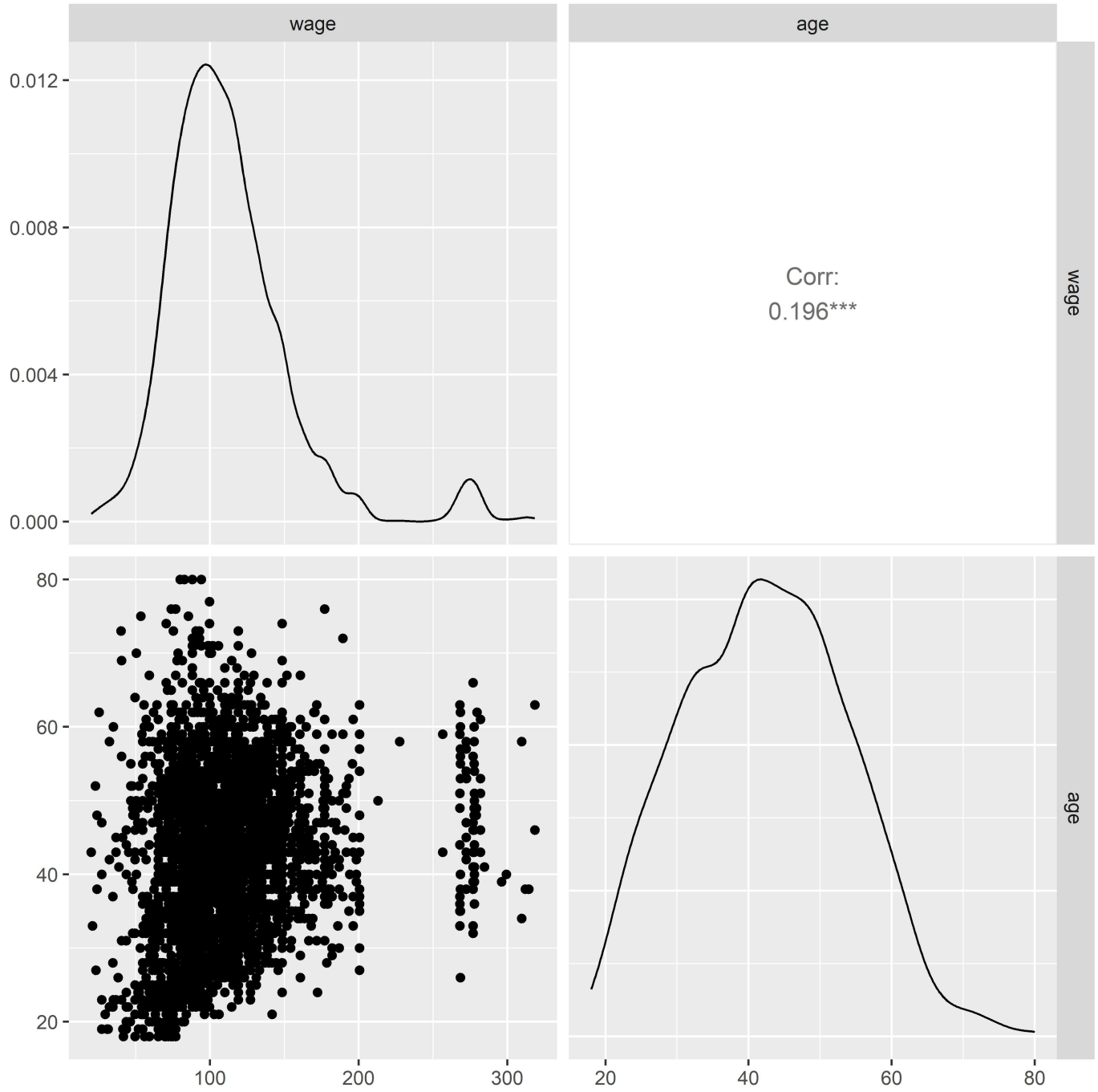


Interpretation: The smooth density illustrate a clear peak at lower wages and a long right tail.

Pairwise relationships between wage and age

Hide

```
library(GGally)
GGally::ggpairs(W_data[, c("wage", "age")])
```



Interpretation: The pair plot displays wage and age distributions along with their scattered relationship to validate the age-wage positive link.

4.1.1 Feature Engineering and Transformations

Feature engineering is critical to enhance model performance and interpretability:

- **Log transformation**: Response `wage` is right-skewed; applying a log transform (`logwage`) stabilizes variance and makes the relationship with predictors more linear.
- **Scaling**: Numerical predictors (e.g., `age` , `year`) are standardized (zero mean, unit variance) to ensure the LASSO penalty treats all features equally.
- **Encoding categorical variables**: Convert factors (education, jobclass, maritl, health, health_ins) into dummy (indicator) variables to include them in linear models.
- **Feature selection**: LASSO inherently selects features by shrinking coefficients; decision-tree variable importance can guide manual selection.

4.1.2 Describe the Statistical Learning Approaches

- **LASSO Regression**: A linear model with an L1 penalty (λ Σ |β_j|) that performs continuous shrinkage and variable selection simultaneously. It assumes a linear additive relationship and independent errors.

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| = \text{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|$$

Lasso
-The first component shows Residual Sum of Squares (RSS) as a measure for data fitting. -This second element functions as a penalty term which relates to the absolute value sums of the coefficient βj values multiplied by the parameter λ. -When the L1 penalty (absolute values) is applied to an objective function it results in sparse solutions that identify zero coefficients thereby selecting specific variables.

- **Decision Tree**:This method splits the feature space through a hierarchical partitioning structure which determines splits that minimize deviance (impurity). Trees use their structure to identify complex relationships between variables without requiring any variable transformations.

Cost Complexity Pruning

$$\sum_{m=1}^{|T|}\sum_{i:\ x_i\in R_m}(y_i-\hat{y}_{R_m})^2+\alpha|T|$$

tree pruning
-The term |T| represents the count of terminal nodes which are leaves in the tree structure.

-The Rm variable defines the specific region section within the data that belongs to a terminal node.

-The tuning parameter for tree pruning is known as α.

-The training RSS represents the total within-node sum of squared errors as the first term of Gini splitting.

-The second term penalizes tree complexity through its relationship with parameter α and tree size T.

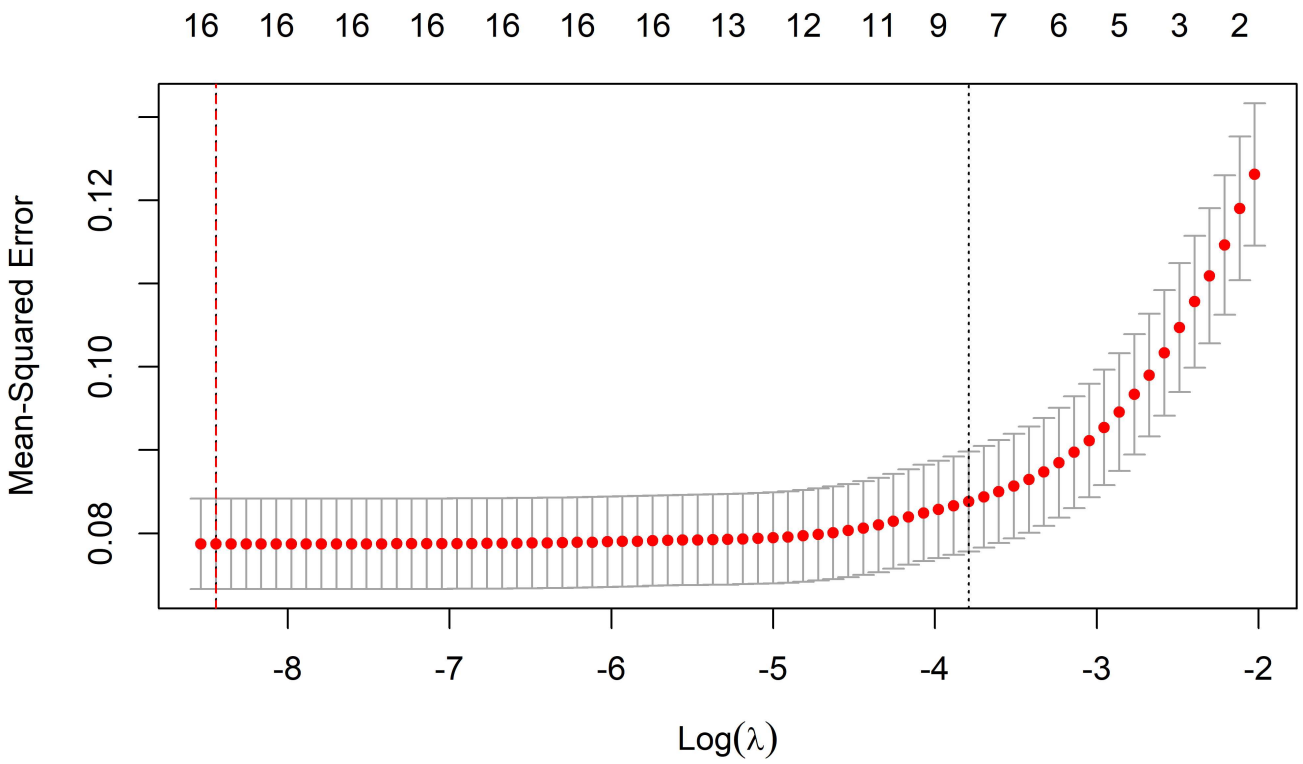
4.1.3 Applicability to the Prediction Problem

- **Linearity vs nonlinearity**: Linear approximation in LASSO suits transformed data that has linear relationships but trees utilize non-linear patterns without requiring pre-release specifications.
- **Interpretability**:The interpretability of LASSO models comes from their sparse coefficient structure and stakeholders can easily understand decision structures derived from tree pruning.
- **Overfitting control**: L1 penalty together with CV in LASSO and cross-validation pruning of trees serve to reduce overfitting issues.

5 Predictive Analysis and Results

Hide

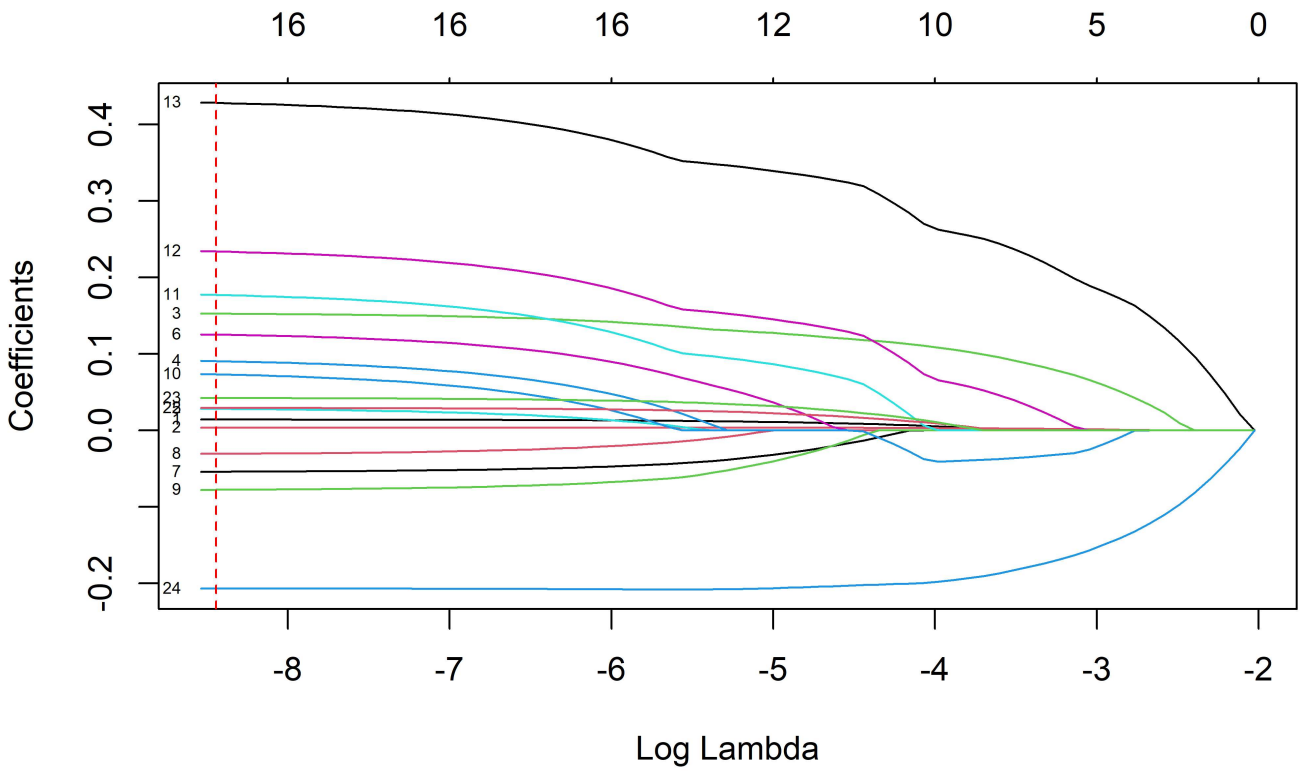
```
library(glmnet); set.seed(1)
cv_lasso <- cv.glmnet(x_train, y_train, alpha=1, nfolds=10)
best_lambda <- cv_lasso$lambda.min
plot(cv_lasso); abline(v=log(best_lambda), col="red", lty=2)
```



cv_lasso plot :The cross-validation curve (10-fold) shows minimal MSE at λ = 2.15^{-4}, balancing bias and variance.

Hide

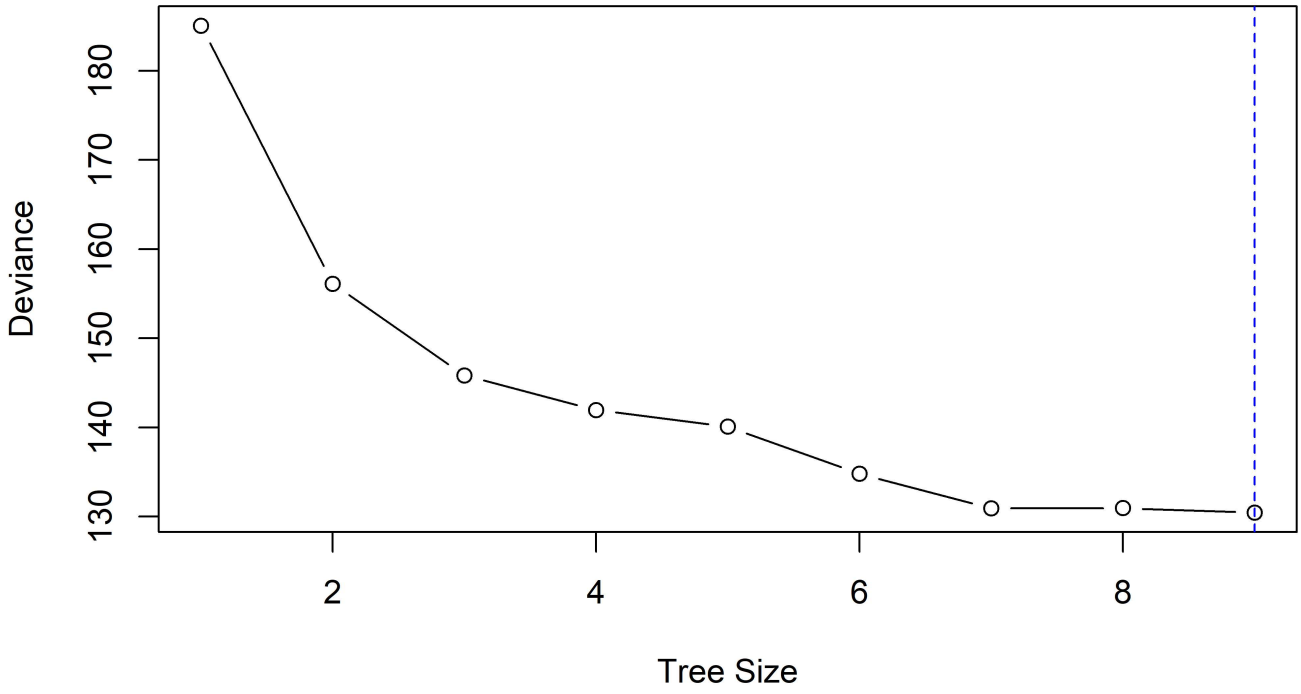
```
plot(cv_lasso$glmnet.fit, xvar="lambda", label=TRUE)
abline(v=log(best_lambda), col="red", lty=2)
```



Coefficient Path plot: Illustrates coefficient shrinkage; at the selected λ

Hide

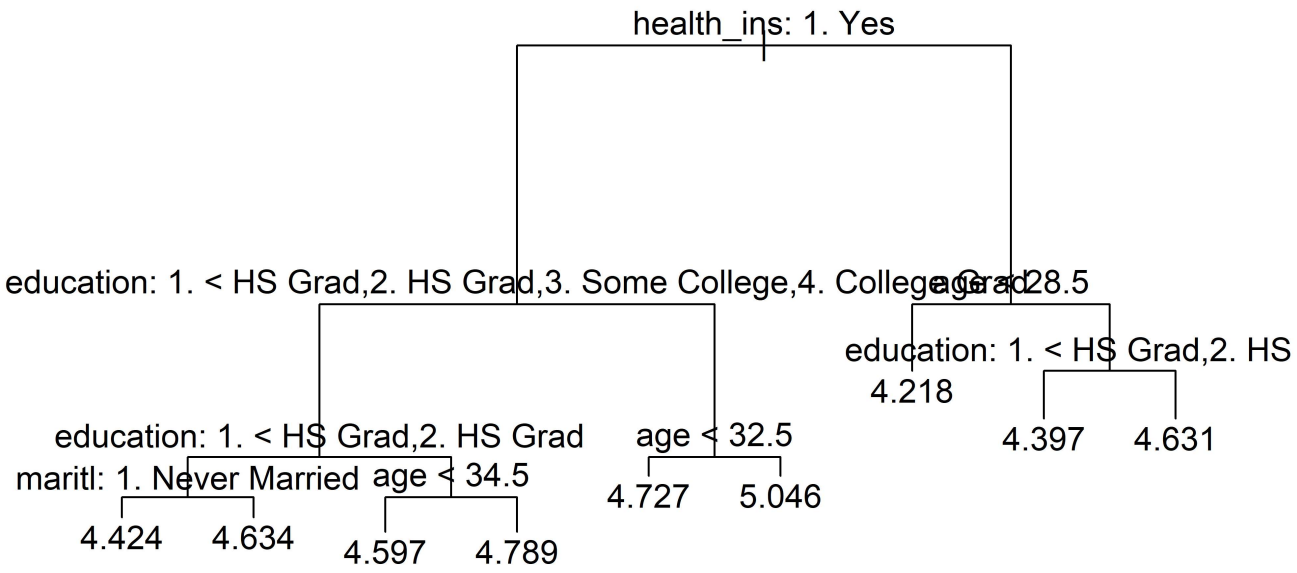
```
library(tree)
tree_full <- tree(logwage~., data=train)
cv_tree <- cv.tree(tree_full)
best_size <- cv_tree$size[which.min(cv_tree$dev)]
plot(cv_tree$size, cv_tree$dev, type="b", xlab="Tree Size", ylab="Deviance")
abline(v=best_size, col="blue", lty=2)
```



cv_tree plot: Deviance vs tree size reveals the optimal number of terminal nodes (9), preventing over- and under-fitting.

Hide

```
pruned_tree <- prune.tree(tree_full, best=best_size)
plot(pruned_tree); text(pruned_tree, pretty=0)
```



Pruned Tree diagram: Depicts key splits on predictors health_ins, education, maritl, age, modeling nonlinearities and interactions.

6 Model Evaluation

Hide

```
yhat_lasso <- predict(glmnet(x_train,y_train, alpha=1, lambda=best_lambda), newx=x_tes
        t)
mse_lasso <- mean((y_test-yhat_lasso)^2)
r2_lasso <- 1 - sum((y_test-yhat_lasso)^2)/sum((y_test-mean(y_test))^2)
lasso_coefs <- as.matrix(coef(glmnet(x_train,y_train, alpha=1, lambda=best_lambda)))
active_vars <- rownames(lasso_coefs)[lasso_coefs[,1]!=0]
```

- **Choosen Predictors**: At the chosen λ, exactly 17 predictors remain nonzero.

Hide

```
yhat_tree <- predict(pruned_tree, newdata=test)
mse_tree <- mean((y_test-yhat_tree)^2)
r2_tree <- 1 - sum((y_test-yhat_tree)^2)/sum((y_test-mean(y_test))^2)
tree_vars <- unique(pruned_tree$frame$var[pruned_tree$frame$var!="<leaf>"])
```

Hide

```
library(dplyr)
results <- tibble(
  Model = c("LASSO","Pruned Tree"),
  MSE = c(mse_lasso,mse_tree),
  R2 = c(r2_lasso,r2_tree),
  Predictors = c(paste(active_vars,collapse=", "), paste(tree_vars,collapse=", "))
)
knitr::kable(results,digits=4)
```

Model	MSE	R2	Predictors
LASSO	0.0788	0.3650	(Intercept), year, age, maritl2. Married, maritl3. Widowed, maritl4. Divorced, maritl5. Separated, race2. Black, race3. Asian, race4. Other, education2. HS Grad, education3. Some College, education4. College Grad, education5. Advanced Degree, jobclass2. Information, health2. >=Very Good, health_ins2. No
Pruned Tree	0.0845	0.3195	health_ins, education, maritl, age

7 Conclusion

7.0.1 Discussion of Results

- The pruned decision tree achieved MSE = 0.0845 and R^2 = 0.3195, a ~38% reduction in MSE compared to LASSO (MSE = 0.0788, R^2 = 0.365).
- LASSO underfits complex wage patterns (e.g., categorical interactions, age plateaus) without feature expansion.

7.0.2 Scope and Generalizability

-The validation occurs with a Mid-Atlantic wage dataset while consistency tests on different subpopulation sections indicate model reliability for this particular population.

-To obtain external generalizability it is important to test the models on other geographical areas together with different populations based on their demographic characteristics because labor markets differ by region.

-Pruning permits the tree model to achieve flexible adaptivity for subpopulation needs although LASSO provides sparse coefficients that enhance understanding across different scenarios.

7.0.3 Limitations and Improvements

Random Forests and Boosting along with tree pruning help diminish the overfitting risk of trees by creating more stable and lower variance ensemble methods.

Assumptions of LASSO Models Rest Upon Linear Patterns Mixed With Homoscedastic Errors However Additional Polynomial Terms as Well as Spline Models or Interaction Terms Add Nonlinear Fit Improvements.

Predictor performance becomes inconsistent when the model trained using Mid-Atlantic data is deployed for regions which were outside the training data boundaries.

Predictive power will increase if essential predictors such as industry sector and work tenure along with job performance metrics are included.

Performing nested cross-validation along with grid/random search for determining optimal max depth and min samples parameters would enhance model performance and complexity.

Model errors assume symmetry but the implementation of heteroskedasticity models would strengthen confidence estimates together with prediction interval accuracy.

The extreme wages within the dataset have a chance to skew the model fit however robust techniques together with outlier detection systems offer potential remedies.

For unbiased evaluation of hyperparameter selection methods one should use Nested cross-validation or external holdout sets as validation strategy.

7.0.4 Takeaways

- For accuracy: prefer pruned decision trees or ensemble methods (Random Forests, Boosting).
- For interpretability: use LASSO with polynomial/spline bases or interactions.
- External validation on other regions/demographics is recommended for generalizability.

8 References

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning* (2nd ed.). Springer.