BIOST 2155: Assignment 1 — Solutions

Answer key

Released: 2026-09-11. **Due by**: 2026-09-25, 11:59 PM

Instructions

- This file serves as a template for your assignment. **Remember to set eval = TRUE below when compiling the report.**

```
knitr::opts_chunk$set(echo = TRUE, eval = TRUE,
                      fig.height = 4, fig.width = 7, fig.align = "center",
                      message=FALSE)
```

- Use comments to annotate and explain your code whenever possible.
- This assignment covers **Training models** (bias-variance, resampling) and **Regression I** (subset selection, ridge, lasso). It uses one data set: `Boston Housing`, saved as `assign1_regData.csv`

Theory (10 points)

T1. Mathematical basis of variance-bias tradeoff (10 points)

Assume $Y = f(X) + \epsilon$, where $E(\epsilon) = 0, \text{Var}(\epsilon) = \sigma^2$ and ϵ and $f(X)$ are uncorrelated. Prove the key formula for bias-variance tradeoff:

$$\text{MSE} = (\text{bias})^2 + \text{variance} + \text{irreducible error}$$

where $\text{MSE} = E\left[(y - \hat{f}(x))^2\right]$, $\text{bias} = E[\hat{f}(x)] - f(x)$, $\text{variance} = \text{Var}[\hat{f}(x)]$ and $\text{irreducible error} = \sigma^2$.

Hint: expand $E\left[(y - \hat{f}(x))^2\right] = E\left[(y - f(x) + f(x) - \hat{f}(x))^2\right]$ *and expand again* $E\left[(f(x) - \hat{f}(x))^2\right] = E\left[(f(x) - E(\hat{f}(x)) + E(\hat{f}(x)) - \hat{f}(x))^2\right]$.

Step 1 — expand the MSE. Substitute $y = f(X) + \epsilon$:

$$\begin{aligned}\text{MSE} &= E\left[(y - \hat{f}(X))^2\right] = E\left[(f(X) - \hat{f}(X) + \epsilon)^2\right] \\ &= E\left[(f(X) - \hat{f}(X))^2\right] + 2\,E\left[\epsilon(f(X) - \hat{f}(X))\right] + E[\epsilon^2].\end{aligned}$$

The cross term vanishes: ϵ is independent of the model with $E[\epsilon] = 0$, so $2E[\epsilon]\,E[f(X) - \hat{f}(X)] = 0$. The last term is $E[\epsilon^2] = \text{Var}(\epsilon) = \sigma^2$, the **irreducible error**. Hence

$$\text{MSE} = E\left[(f(X) - \hat{f}(X))^2\right] + \sigma^2.$$

Step 2 — decompose the reducible error. Add and subtract $E[\hat{f}(X)]$:

$$E\left[(f(X)-\hat{f}(X))^2\right] = E\left[\left((f(X)-E[\hat{f}(X)])+(E[\hat{f}(X)]-\hat{f}(X))\right)^2\right].$$

Expanding, the cross term is again zero (the deviation of $\hat{f}$ from its own mean has expectation 0), leaving

$$\underbrace{(E[\hat{f}(X)]-f(X))^2}_{\text{Bias}^2} + \underbrace{E\left[(\hat{f}(X)-E[\hat{f}(X)])^2\right]}_{\text{Variance}}.$$

Step 3 — combine.

$$\text{MSE} = \underbrace{(E[\hat{f}(X)]-f(X))^2}_{\text{Bias}^2} + \underbrace{\text{Var}(\hat{f}(X))}_{\text{Variance}} + \underbrace{\sigma^2}_{\text{Irreducible error}}. \quad \blacksquare$$

Application (80 points)

Dataset: `Boston Housing`, saved as `assign1_regData.csv`.

Harrison and Rubinfeld (1978)[a] studied families' willingness to pay for clean air using Boston area housing market data in 1978. The cleaned data consists of 394 rows with **complete** data on census tracts in Boston, with variables collected from a mixture of surveys, administrative records, and other research papers. The variables are described below.

- `CRIM`: per-capita crime rate by town (units and crimes included not specified)
- `ZN`: percentage of residential land zoned for lots over 25,000 sq.ft.
- `INDUS`: proportion of non-retail business acres per town (nonretail businesses include industrial and commercial businesses)
- `CHAS`: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- `NOX`: nitric oxides concentration (parts per 10 million)
- `RM`: average number of rooms per dwelling
- `AGE`: percentage of owner-occupied units built prior to 1940
- `DIS`: weighted log distances to five Boston employment centers
- `RAD`: index of accessibility to radial highways, where a higher number indicates higher accessibility
- `TAX`: full-value property-tax rate per $10,000
- `PTRATIO`: pupil-teacher ratio by school district
- `B`: $1000(B_k - 0.63)^2$ where B_k is the proportion of Black people by town
- `LSTAT`: % lower status of the population, defined as half of the sum of proportion of adults without some high school education and proportion of male workers classified as laborers
- `MEDV`: Median value of owner-occupied homes, in $1000's

We will load the dataset and set the outcome variable to be `MEDV`.

[a] David Harrison, Jr and Daniel L Rubinfeld. Hedonic housing prices and the demand for clean air. Journal of Environmental Economics and Management, 5(1):81–102, 1978.

```
pacman::p_load(tidyverse, caret, boot, glmnet, pls, leaps)
set.seed(42)
data <- read_csv("assign1_regData.csv") |> drop_na()
data$CHAS <- as.factor(data$CHAS)
```

A1. Data Exploration & Baseline Model (5 + 5 + 8 = 18 points)

1. Use the `summary` function to provide descriptive statistics for the dataset. Randomly split your data into a training set (80%) and a test set (20%). Make sure that the distribution of the outcome is similar

in both sets (please read the documentation for `caret::createDataPartition` to know how). Use a fixed seed (use `seed = 42`) for reproducibility.

```
summary(data)
```

```
##       CRIM                ZN              INDUS        CHAS          NOX
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   0:367   Min.   :0.3890
##  1st Qu.: 0.08196   1st Qu.:  0.00   1st Qu.: 5.13   1: 27   1st Qu.:0.4530
##  Median : 0.26888   Median :  0.00   Median : 8.56           Median :0.5380
##  Mean   : 3.69014   Mean   : 11.46   Mean   :11.00           Mean   :0.5532
##  3rd Qu.: 3.43597   3rd Qu.: 12.50   3rd Qu.:18.10           3rd Qu.:0.6240
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74           Max.   :0.8710
##        RM              AGE              DIS              RAD
##  Min.   :3.561   Min.   :  2.90   Min.   : 1.130   Min.   : 1.000
##  1st Qu.:5.879   1st Qu.: 45.48   1st Qu.: 2.110   1st Qu.: 4.000
##  Median :6.202   Median : 77.70   Median : 3.199   Median : 5.000
##  Mean   :6.280   Mean   : 68.93   Mean   : 3.805   Mean   : 9.404
##  3rd Qu.:6.606   3rd Qu.: 94.25   3rd Qu.: 5.117   3rd Qu.:24.000
##  Max.   :8.780   Max.   :100.00   Max.   :12.127   Max.   :24.000
##       TAX           PTRATIO            B              LSTAT
##  Min.   :187.0   Min.   :12.60   Min.   :  2.6   Min.   : 1.730
##  1st Qu.:280.2   1st Qu.:17.40   1st Qu.:376.7   1st Qu.: 7.125
##  Median :330.0   Median :19.10   Median :392.2   Median :11.300
##  Mean   :406.4   Mean   :18.54   Mean   :358.5   Mean   :12.769
##  3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.9   3rd Qu.:17.117
##  Max.   :711.0   Max.   :22.00   Max.   :396.9   Max.   :37.970
##       MEDV
##  Min.   : 5.00
##  1st Qu.:16.80
##  Median :21.05
##  Mean   :22.36
##  3rd Qu.:25.00
##  Max.   :50.00
```

```
set.seed(42)
trainIndex <- createDataPartition(data$MEDV, p = 0.8, list = FALSE)
train_set <- data[trainIndex, ]
test_set  <- data[-trainIndex, ]
c(train = nrow(train_set), test = nrow(test_set))
```

```
## train  test
##   317    77
```

`createDataPartition` splits *within* quantiles of `MEDV`, so the price distribution matches across train and test.

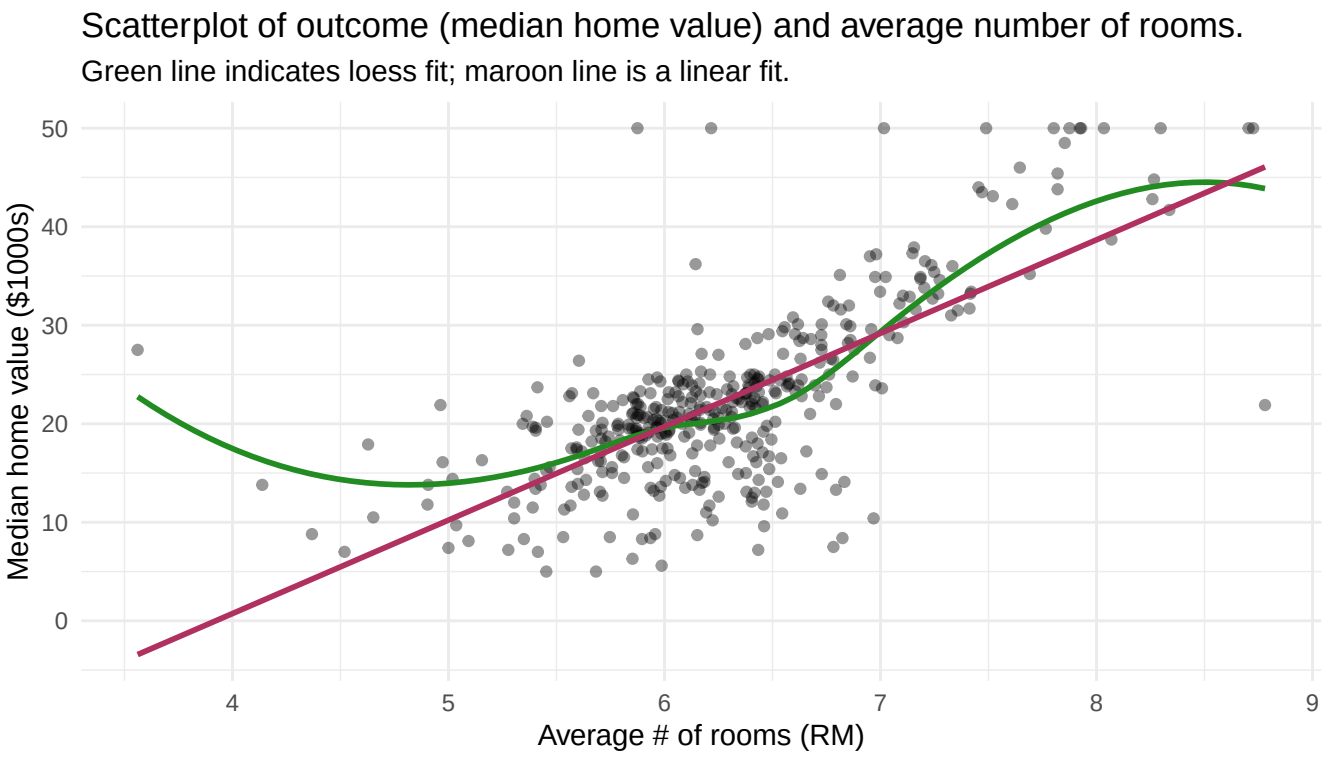
2. Perform a brief exploratory data analysis: comment on how the outcome varies with the `RM` predictor - is the behavior expected?

```
ggplot(data, aes(RM, MEDV)) +
  geom_point(alpha = 0.4) +
  geom_smooth(method = "loess", se = FALSE, label = "LOESS", color = "forestgreen") +
    geom_smooth(method = "lm", se = FALSE, label = "lm", color = "maroon") +
  labs(x = "Average # of rooms (RM)", y = "Median home value ($1000s)",
       title = "Scatterplot of outcome (median home value) and average number of rooms.",
```

```
      subtitle = "Green line indicates loess fit; maroon line is a linear fit.") +
  theme_minimal()
```

```
## Warning in geom_smooth(method = "loess", se = FALSE, label = "LOESS", color =
## "forestgreen"): Ignoring unknown parameters: `label`
```

```
## Warning in geom_smooth(method = "lm", se = FALSE, label = "lm", color =
## "maroon"): Ignoring unknown parameters: `label`
```



The relationship is increasing, as expected: more rooms means larger homes, which cost more. The trend is roughly linear through the middle and steepens for large `RM`; a few high- and low-price outliers exist but the overall positive association is clear.

3. Fit a multiple linear regression model using all predictors on the training data. Report the test set Root Mean Squared Error (RMSE) and interpret the coefficient on `RM`.

```
lm_model <- lm(MEDV ~ ., data = train_set)
rmse_full <- sqrt(mean((predict(lm_model, test_set) - test_set$MEDV)^2))
c(test_RMSE = round(rmse_full, 3), RM_coef = round(coef(lm_model)["RM"], 3))
```

```
##  test_RMSE RM_coef.RM
##      5.652      3.751
```

The test RMSE is about **5.65** (in $1000s). The `RM` coefficient (about **3.75**) means that, holding all other predictors fixed, each additional room per dwelling is associated with an increase of roughly **3.75 units** — that is, about **$3,750** — in median home value.

A2. Resampling and Model Assessment (6 + 10 + 4 = 20 points)

1. Using the training data, perform 10-fold cross-validation on the full multiple linear regression model from A1. What is the average cross-validated RMSE?

```
set.seed(42)
cv_fit <- train(MEDV ~ ., data = train_set, method = "lm",
                trControl = trainControl(method = "cv", number = 10),
                metric = "RMSE")
round(cv_fit$results$RMSE, 3)
```

```
## [1] 4.274
```

The average 10-fold CV RMSE is about **4.27** — a little lower than the single test-set RMSE, which is normal (different held-out samples).

2. Use the bootstrap method (with $B = 1000$ resamples, set `seed = 42`) on the training data to estimate the 95% confidence interval (CI) for the `RM` coefficient. Compare the bootstrap-based 95% CI with the 95% CI using the standard formula from our linear regression model.

```
get_coef <- function(d, idx) coef(lm(MEDV ~ ., data = d[idx, ]))["RM"]
set.seed(42)
boot_rm <- replicate(1000,
  get_coef(train_set, sample(nrow(train_set), replace = TRUE)))

rbind(bootstrap = quantile(boot_rm, c(0.025, 0.975)),
      OLS       = confint(lm_model, "RM")) |> round(3)
```

```
##            2.5% 97.5%
## bootstrap 2.212 5.886
## RM        2.756 4.746
```

3. *Write your interpretation of the bootstrap confidence interval here. How does the theoretically-derived interval compare to the one you just estimated with the bootstrap method? If they differ, what could be some potential causes?*

Interpretation. The two intervals overlap but the **bootstrap CI is wider** than the OLS interval. The OLS interval is *parametric*: it assumes homoscedastic, normally-distributed errors. The bootstrap is *non-parametric* — it re-estimates the sampling distribution of $\hat{\beta}_{RM}$ directly by resampling, so it does not lean on those assumptions. The wider bootstrap interval suggests the OLS formula slightly **understates** the true uncertainty in $\hat{\beta}_{RM}$, most likely because of mild non-normality or heteroscedasticity in the residuals (the outliers visible in A1.2 are a plausible cause). A residuals-vs-fitted plot and a Q–Q plot would confirm which assumption is strained.

A3. Feature Selection & Regularization (12 + 12 + 12 + 6 = 42 points)

1. Stepwise Selection: On the training data, perform forward stepwise selection to find the best model according to BIC. List the variables selected by the procedure, and report the test set RMSE using this smaller model.

```
null_m <- lm(MEDV ~ 1, data = train_set)
full_m <- lm(MEDV ~ ., data = train_set)
step_bic <- step(null_m,
                 scope = list(lower = formula(null_m), upper = formula(full_m)),
```

```
                direction = "forward", k = log(nrow(train_set)), trace = 0)
selected <- names(coef(step_bic))[-1]
rmse_step <- sqrt(mean((predict(step_bic, test_set) - test_set$MEDV)^2))
list(selected = selected, test_RMSE = round(rmse_step, 3))
```

```
## $selected
## [1] "LSTAT"   "RM"      "PTRATIO" "B"       "DIS"     "NOX"     "ZN"
## [8] "CRIM"
##
## $test_RMSE
## [1] 5.791
```

BIC keeps **LSTAT, RM, PTRATIO, B, DIS, NOX, ZN, CRIM** and drops the rest (`INDUS`, `AGE`, `RAD`, `TAX`, `CHAS`). The smaller model's test RMSE is about **5.79** — essentially the same as the full model, with far fewer predictors.

2. Ridge Regression: Fit a ridge regression model, using 10-fold CV to determine the optimal lambda for the training set. Report this lambda and the test set RMSE.

```
set.seed(42)
ridge <- train(MEDV ~ ., data = train_set, method = "glmnet",
               trControl = trainControl(method = "cv", number = 10),
               tuneGrid = expand.grid(alpha = 0, lambda = 10^seq(-3, 1, length = 1000)),
               preProcess = c("center", "scale"))
rmse_ridge <- sqrt(mean((predict(ridge, test_set) - test_set$MEDV)^2))
c(lambda = round(ridge$bestTune$lambda, 4), test_RMSE = round(rmse_ridge, 3))
```

```
##    lambda test_RMSE
##    0.6469    5.6850
```

3. Lasso Regression: Fit a lasso regression model. Use 10-fold CV to determine the optimal lambda. Report optimal lambda, list variables shrunk to zero, and calculate test set RMSE. Finally, plot coefficient paths for varying lambda.

```
set.seed(42)
lasso <- train(MEDV ~ ., data = train_set, method = "glmnet",
               trControl = trainControl(method = "cv", number = 10),
               tuneGrid = expand.grid(alpha = 1, lambda = 10^seq(-3, 1, length = 1000)),
               preProcess = c("center", "scale"))
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo,
## : There were missing values in resampled performance measures.
```

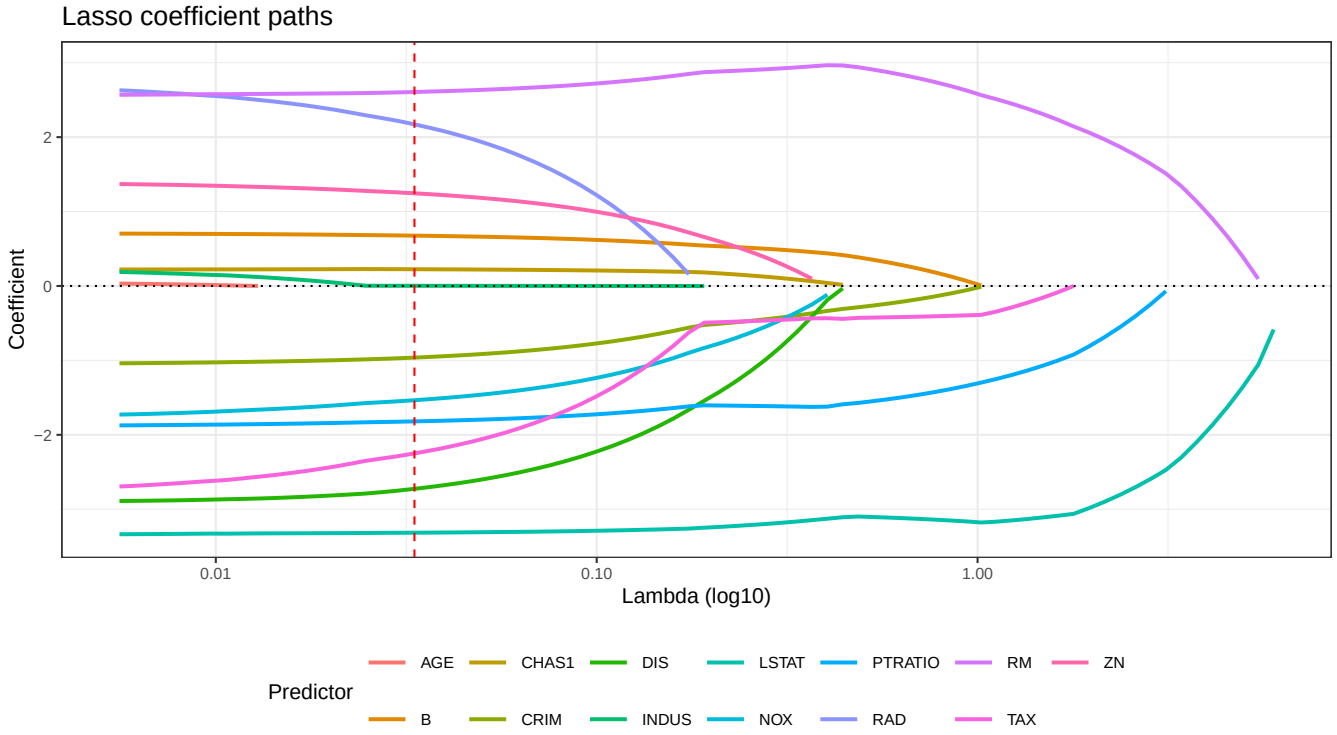
```
rmse_lasso <- sqrt(mean((predict(lasso, test_set) - test_set$MEDV)^2))
cf <- as.matrix(predict(lasso$finalModel, type = "coefficients",
                        s = lasso$bestTune$lambda))
zeroed <- rownames(cf)[cf[, 1] == 0]
list(lambda = round(lasso$bestTune$lambda, 4),
     test_RMSE = round(rmse_lasso, 3),
     shrunk_to_zero = zeroed)
```

```
## $lambda
## [1] 0.0332
##
```

```
## $test_RMSE
## [1] 5.654
##
## $shrunk_to_zero
## [1] "INDUS" "AGE"
```

Lasso shrinks **INDUS, AGE** to exactly zero — the same weak predictors BIC dropped.

```
paths <- broom::tidy(lasso$finalModel) |> filter(term != "(Intercept)")
ggplot(paths, aes(lambda, estimate, colour = term)) +
  geom_line(linewidth = 0.7) + scale_x_log10() +
  geom_vline(xintercept = lasso$bestTune$lambda, linetype = 2, colour = "red") +
  geom_hline(yintercept = 0, linetype = 3) +
  labs(title = "Lasso coefficient paths", x = "Lambda (log10)",
       y = "Coefficient", colour = "Predictor") +
  theme_bw(base_size = 8) + theme(legend.position = "bottom") +
  guides(colour = guide_legend(nrow = 2))
```



4. Comparison: Make a small table comparing the test set RMSE of the full linear model (A1.3), the stepwise-selected model, ridge, and lasso. Which model would you recommend, and why?

```
tibble(model = c("Full linear (A1.3)", "Stepwise-BIC", "Ridge", "Lasso"),
       test_RMSE = round(c(rmse_full, rmse_step, rmse_ridge, rmse_lasso), 3))
```

```
## # A tibble: 4 x 2
##   model              test_RMSE
##   <chr>                  <dbl>
## 1 Full linear (A1.3)      5.65
```

```
## 2 Stepwise-BIC          5.79
## 3 Ridge                 5.68
## 4 Lasso                 5.65
```

All four models land within a hair of each other (test RMSE about 5.65 to 5.79). On this clean, low-dimensional data (13 predictors, no severe collinearity), regularization has little to gain — the full OLS fit is already stable, so ridge and lasso barely move the RMSE. I would **recommend the lasso** (or the BIC-selected model): it matches the full model's accuracy while dropping `INDUS` and `AGE`, giving a simpler, more interpretable model at no predictive cost. If pure prediction were the only goal, the full linear model is equally defensible.

Optional / Bonus (unscored, will still provide feedback if you answer)

1. LOO-CV: Prove formula (5.2) from ISLR, Section 5.1.2, page 202.

2. Basis of 0.632 **bootstrap:** Problem 5.4.2 from ISLR, page 219.

3. Subgroup fairness (Boston housing): The `B` variable in this data set is defined as $1000(B_k - 0.63)^2$ where B_k is the proportion of Black people by town — due to the squared term this treats towns with very low and very high proportions of Black residents similarly, which is a problematic construction. Split the data into two groups by the median of `B`, fit separate linear models on each group's `RM` and `LSTAT`, and compare in-group vs. cross-group test RMSE. What does this suggest about the dangers of training and deploying models on demographically specific or unrepresentative data?

Brief sketches of the intended answers:

1. LOO-CV shortcut (ISLR 5.2). For least squares, $\text{CV}_{(n)} = \frac{1}{n}\sum_i \left(\frac{y_i - \hat{y}_i}{1 - h_i}\right)^2$, where h_i is the i-th leverage (diagonal of the hat matrix). It follows because deleting point i changes the fitted value in a way captured exactly by h_i, so no refitting is needed.

2. 0.632 bootstrap (ISLR 5.4.2). The probability a given observation is *not* drawn in one bootstrap sample of size n is $(1 - 1/n)^n \to e^{-1} \approx 0.368$, so each observation appears in a resample with probability ≈ 0.632 — the origin of the 0.632 estimator's weighting.

3. Subgroup fairness. Splitting on the median of `B`, fitting `RM`+`LSTAT` models within each half, and evaluating in-group vs cross-group test RMSE shows that a model trained on one demographic subgroup degrades when applied to the other. The lesson: models fit on demographically narrow or unrepresentative data can carry that bias into deployment, so the `B` construction (which conflates very low and very high proportions of Black residents through the squared term) is a warning about feature design, not just data coverage.