

Stat 142 MP Light 7

First Semester, A.Y. 2025-26

First Name: Anne Christine

Last Name: Amores

Student Number: 2021-04650

Section: WFU

Bootstrap and Cross-Validation

1. Perform LOOCV for the following models and compute the RMSE manually. You may opt to edit the function given in class:

- Volume ~ Girth
- Volume ~ Height
- Volume ~ Girth + Height

Use the built in data in R called **trees**.

```
loocv <- function(formula, data){  
  n <- nrow(data)  
  eval <- numeric(n)  
  for(i in 1:n){  
    train_i <- data[-i,]  
    test_i <- data[i,]  
    mod_i <- lm(formula, train_i)  
    pred_i <- predict(mod_i, test_i)  
  
    y <- all.vars(formula)[1]  
    eval[i] <- sqrt(mean((test_i[[y]] - pred_i)^2))  
  }  
  return(mean(eval))  
}
```

```
# LOOCV for Volume ~ Girth  
loocv(Volume ~ Girth, trees)
```

```
## [1] 3.636654
```

```
# LOOCV for Volume ~ Height  
loocv(Volume ~ Height, trees)
```

```

## [1] 11.77131

# LOOCV for Volume ~ Girth + Height
loocv(Volume ~ Girth + Height, trees)

```

```

## [1] 3.355245

```

a. **Identify the best overall model based on RMSE.**

The best overall model based on RMSE is Volume ~ Girth + Height, since it has the lowest RMSE (≈ 3.36).

b. **Identify the best simple linear regression model (SLRM) based on RMSE.**

The best SLRM based on RMSE is Volume ~ Girth, since it has a lower RMSE of ≈ 3.64 compared to Volume ~ Height which has RMSE of ≈ 11.77 .

2. Create a function `res_boot` that implements residual bootstrap in simple linear regression. It should have the following parameters:

- `y` - a vector of the dependent variable
- `x` - a vector of the independent variable
- `alpha` - level of significance
- `B` - the number of bootstrap replicates

It should have the following as output in a list:

- `intercept_vec`: vector of bootstrap estimates for the intercept
- `betahat_vec`: vector of bootstrap estimates for the slope (or $\hat{\beta}_1$)
- `conf_inf_int`: $(1 - \alpha)\%$ percentile bootstrap confidence interval for the intercept
- `conf_inf_beta`: $(1 - \alpha)\%$ percentile bootstrap confidence interval for the slope

```

res_boot <- function(y, x, alpha = 0.05, B) {
  # Description
  # Performs a residual bootstrap for simple linear regression to approximate the sampling
  # distributions of the intercept and slope and compute bootstrap percentile confidence
  # intervals

  # Parameters
  # y -- a vector of the dependent variable
  # x -- a vector of the independent variable
  # alpha -- level of significance
  # B -- number of bootstrap replicates

  # Value
  # A list containing:
  # intercept_vec -- a vector of the B bootstrap estimates for the intercept
  # betahat_vec -- a vector of the B bootstrap estimates for the slope coefficient
  # conf_inf_int -- (1 - alpha)\% percentile bootstrap CI for the intercept
}

```

```

# conf_inf_beta -- (1 - alpha)% percentile bootstrap CI for the slope coefficient

# Fit the model on the original data
fit <- lm(y ~ x)
y_hat <- fitted(fit)
resid <- resid(fit)
n <- length(y)

# Pre-allocate memory
intercept_vec <- numeric(B)
betahat_vec <- numeric(B)

for (b in 1:B) {
  # Sample from resid with replace = TRUE (same sample size)
  e_star <- sample(resid, n, replace = TRUE)
  # Compute the vector y_star
  y_star <- y_hat + e_star
  # Refit where y is now y_star and x is unchanged
  fit_b <- lm(y_star ~ x)
  # Get the coefficients
  intercept_vec[b] <- coef(fit_b)["(Intercept)"]
  betahat_vec[b] <- coef(fit_b)["x"]
}
# Return the results as a list
results <- list(
  intercept_vec = intercept_vec,
  betahat_vec = betahat_vec,
  conf_inf_int = quantile(intercept_vec, c(alpha/2, 1 - alpha/2)),
  conf_inf_beta = quantile(betahat_vec, c(alpha/2, 1 - alpha/2))
)
return(results)
}

```

- a. Apply this function to `trees$Volume` as your dependent variable, and your independent variable from the best SLRM model in item 1. Set $B = 2000$ and $\alpha = 0.05$. Use `seed = 142` before applying this function.

```

set.seed(142)
boot_results <- res_boot(trees$Volume, trees$Girth, alpha = 0.05, B = 2000)

```

- b. Print both confidence intervals in a separate code chunk and test and interpret $H_a : \beta_1 \neq 0$.

```

int_ci <- boot_results$conf_inf_int
beta_ci <- boot_results$conf_inf_beta

print(int_ci)

##      2.5%    97.5%
## -43.48873 -30.40490

```

```

print(beta_ci)

##      2.5%    97.5%
## 4.608310 5.543577

```

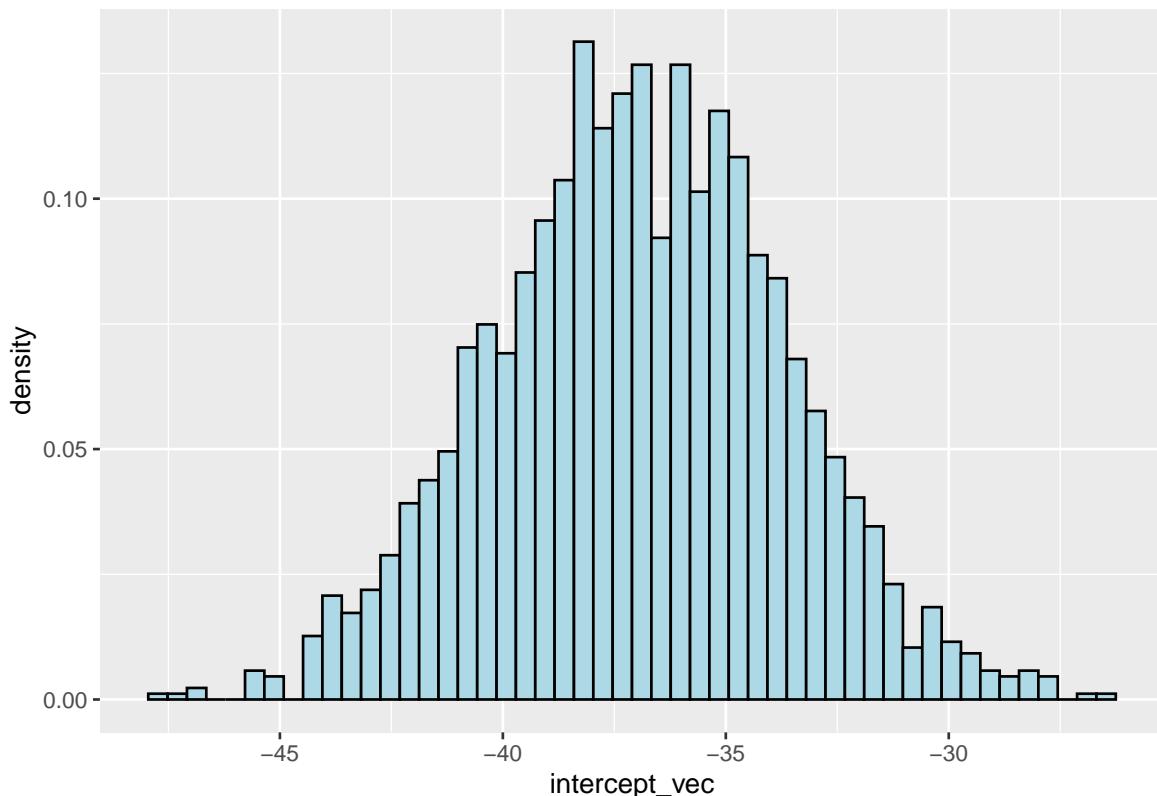
Since both the lower and upper limit of the percentile CI for the slope are positive (i.e, the CI does not contain 0), then we have sufficient evidence to say, at $\alpha = 0.05$, that the slope is not zero. This means that there is a significant relationship between the volume of a tree and its girth.

- c. Using **ggplot2**, plot the sampling distributions (density, not ECDF) of both the bootstrap intercept coefficients and the bootstrap slope coefficients ($\hat{\beta}_1$). Describe the plot based on their shape.

```

# Plot of sampling dist of bootstrap intercept coefficients
df <- data.frame(intercept_vec = boot_results$intercept_vec)
ggplot(df, aes(x = intercept_vec)) +
  geom_histogram(aes(y = after_stat(density)), bins = 50, fill = "lightblue",
                 color = "black")

```

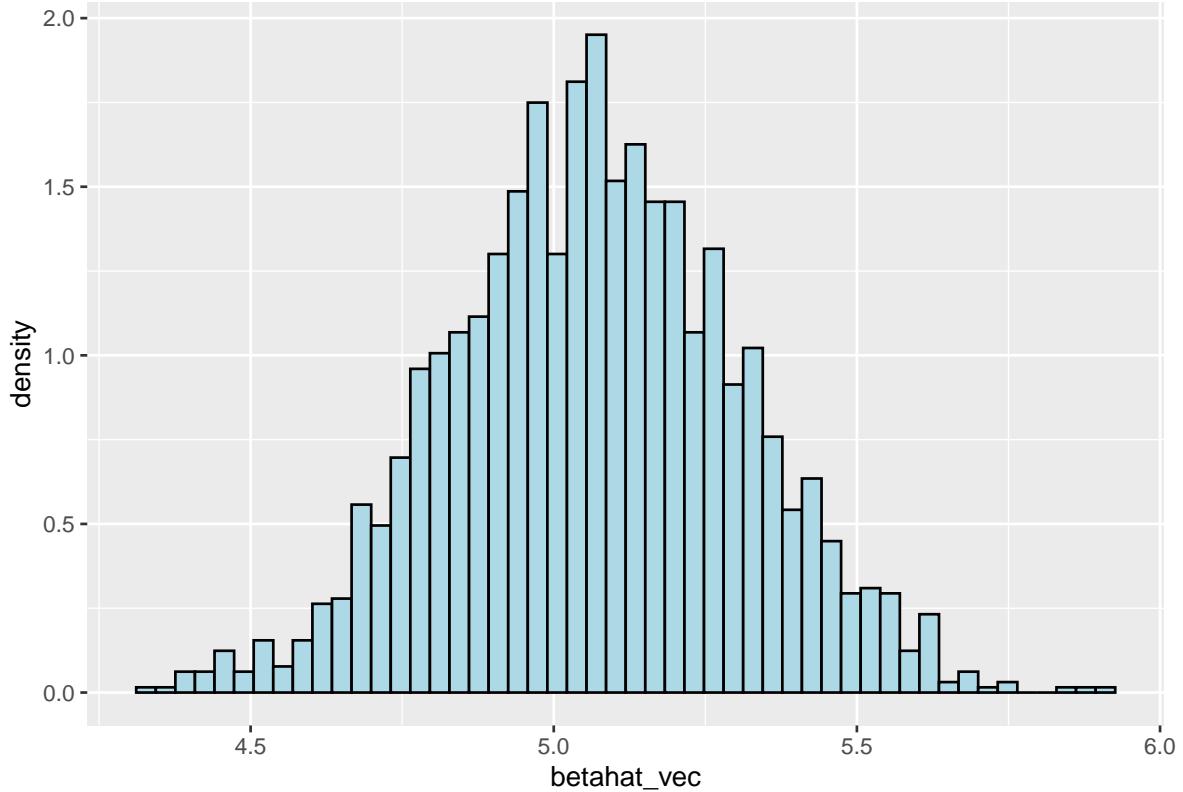


The plot of the sampling distribution of the bootstrap intercept coefficients appears to be bell-shaped, resembling that of a normal distribution.

```

# Plot of sampling dist of bootstrap slope coefficients
df <- data.frame(betahat_vec = boot_results$betahat_vec)
ggplot(df, aes(x = betahat_vec)) +
  geom_histogram(aes(y = after_stat(density)), bins = 50, fill = "lightblue",
                 color = "black")

```



Similar to the earlier plot, the plot of the sampling distribution of the bootstrap slope coefficients appears to be bell-shaped, resembling that of a normal distribution.

- d. Compute the 95% empirical (basic) bootstrap confidence intervals for both the intercept and slope, and compare them to the percentile bootstrap CIs in terms of length.

```

orig_fit <- lm(trees$Volume ~ trees$Girth)
orig_int <- coef(orig_fit)["(Intercept)"]
orig_slope <- coef(orig_fit)["trees$Girth"]

# Empirical/basic CI for the intercept
int_basic <- 2*orig_int - quantile(boot_results$intercept_vec, c(0.975, 0.025))
print(int_basic)

##      97.5%      2.5%
## -43.48202 -30.39819

# Percentile CI for the intercept
print(int_ci)

##      2.5%      97.5%
## -43.48873 -30.40490

```

The empirical/basic CI and the percentile CI for the intercept have nearly identical lower and upper limits.

```

# Empirical/basic CI for the slope
beta_basic <- 2*orig_slope - quantile(boot_results$betahat_vec, c(0.975, 0.025))
print(beta_basic)

##      97.5%    2.5%
## 4.588136 5.523403

# Percentile CI for the intercept
print(beta_ci)

##      2.5%    97.5%
## 4.608310 5.543577

```

Similar to the previous comparison, the empirical/basic CI and percentile CI for the slope are very close to each other. However, the gap between their corresponding limits is slightly larger than what we observed for the intercept.