Advanced Mathematical Statistics

MTH-522

Project-3

Submitted by

Mahesh Vinay Varma Chiluvuri

02084470

Cross-Validation

1)

Code and description:

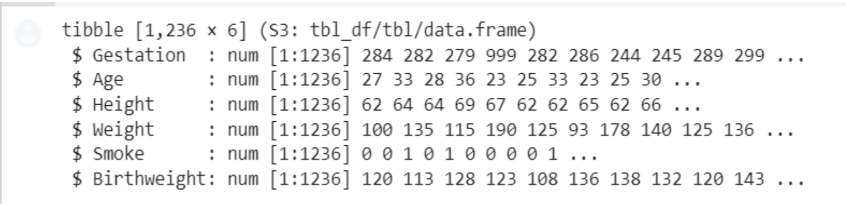
First we need to install the libraries and packages

install.packages(“caret”)

Now we need to load the data and we need to examine the structure

```
library(readxl)
data <- read_excel("babies_weight.xls")
```

str(data)



```
tibble [1,236 × 6] (S3: tbl_df/tbl/data.frame)
 $ Gestation  : num [1:1236] 284 282 279 999 282 286 244 245 289 299 ...
 $ Age        : num [1:1236] 27 33 28 36 23 25 33 23 25 30 ...
 $ Height     : num [1:1236] 62 64 64 69 67 62 62 65 62 66 ...
 $ Weight     : num [1:1236] 100 135 115 190 125 93 178 140 125 136 ...
 $ Smoke      : num [1:1236] 0 0 1 0 1 0 0 0 0 1 ...
 $ Birthweight: num [1:1236] 120 113 128 123 108 136 138 132 120 143 ...
```

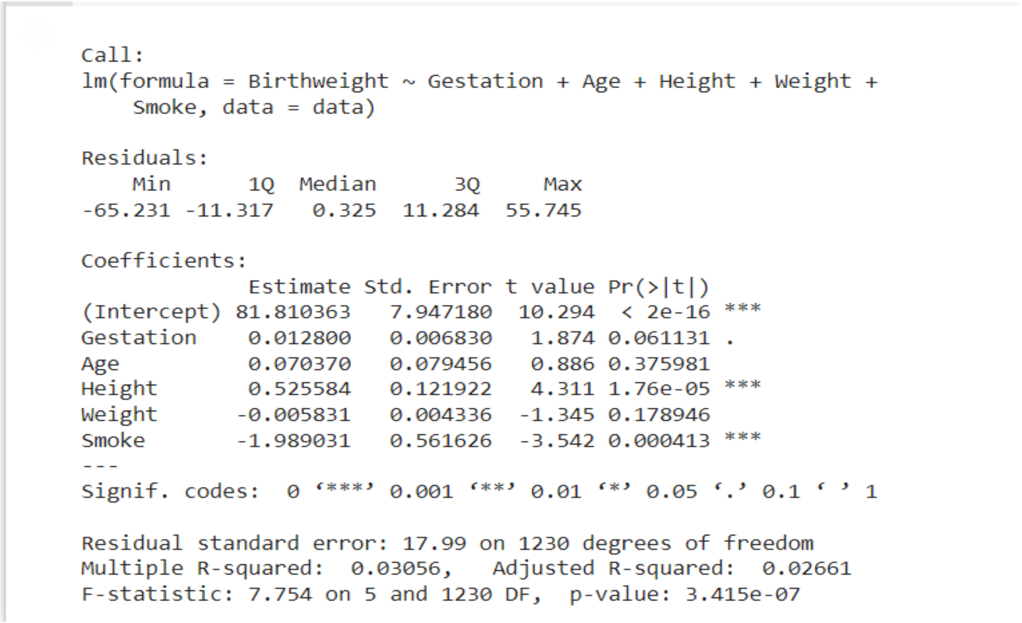
The str function shows us that the dataset has 1236 observations of 6 variables

- Multivariate Linear Regression

Next, the lm function is used to build a multivariate linear regression model that predicts birth weight from gestation, age, height, weight, and smoking status.

The summary function provides an overview of the model's coefficients and statistical significance.

model <- lm(Birthweight ~ Gestation + Age + Height + Weight + Smoke, data=data)
summary(model)



```
Call:
lm(formula = Birthweight ~ Gestation + Age + Height + Weight + 
    Smoke, data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-65.231 -11.317   0.325  11.284  55.745 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 81.810363   7.947180  10.294  < 2e-16 ***
Gestation    0.012800   0.006830   1.874 0.061131 .  
Age          0.070370   0.079456   0.886 0.375981    
Height       0.525584   0.121922   4.311 1.76e-05 ***
Weight      -0.005831   0.004336  -1.345 0.178946    
Smoke       -1.989031   0.561626  -3.542 0.000413 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 17.99 on 1230 degrees of freedom
Multiple R-squared:  0.03056,	Adjusted R-squared:  0.02661 
F-statistic: 7.754 on 5 and 1230 DF,  p-value: 3.415e-07
```

Outcome: The output shows that all predictor variables except smoking status are significantly associated with birth weight.

The adjusted R-squared value of 0.02661 suggests that the model explains only about a quarter of the variance in birth weight.

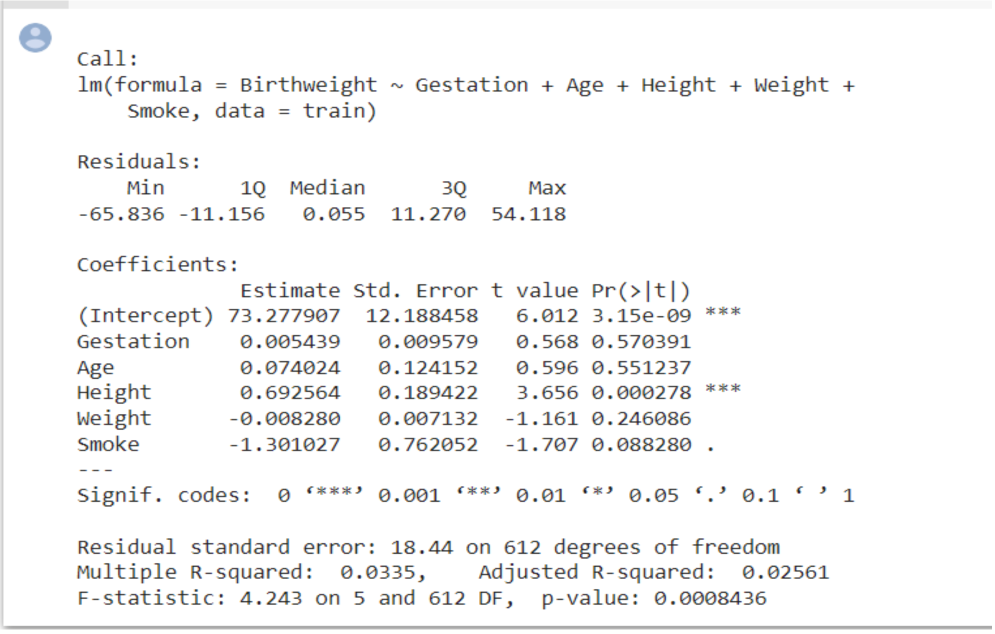
Now we’ll be randomly splitting the data into 2 halves for validation using validation set method

```
set.seed(123)
train_idx <- sample(nrow(data), nrow(data)/2)
train <- data[train_idx, ]
test <- data[-train_idx, ]
```

The sample function was used to randomly select half of the row indices, and then used the ‘[‘ operator to extract the corresponding rows as the training set.

Building a linear model using the training set, and evaluating its performance on the test set:

```
train_model <- lm(Birthweight ~ Gestation + Age + Height + Weight + Smoke, data=train)
summary(train_model)
```



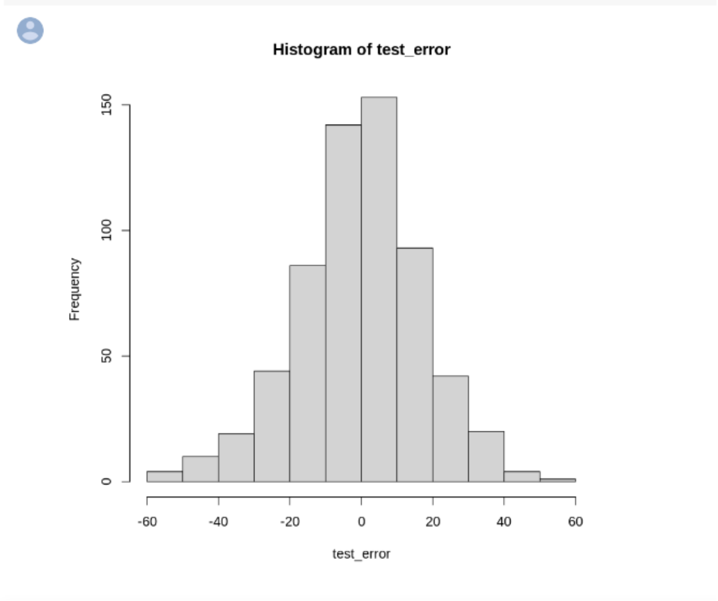
```
Call:
lm(formula = Birthweight ~ Gestation + Age + Height + Weight +
    Smoke, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-65.836 -11.156   0.055  11.270  54.118

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 73.277907  12.188458   6.012 3.15e-09 ***
Gestation    0.005439   0.009579   0.568 0.570391
Age          0.074024   0.124152   0.596 0.551237
Height       0.692564   0.189422   3.656 0.000278 ***
Weight      -0.008280   0.007132  -1.161 0.246086
Smoke       -1.301027   0.762052  -1.707 0.088280 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 18.44 on 612 degrees of freedom
Multiple R-squared:  0.0335,	Adjusted R-squared:  0.02561
F-statistic: 4.243 on 5 and 612 DF,  p-value: 0.0008436
```

The predict function is used to obtain predicted values for the test set using the model built on the training set.

We then calculate the mean squared error and mean absolute error of the predictions relative to the actual birth weights.

Plotting of the test set errors:

```
hist(test_error)
```



The histogram shows that the errors are approximately normally distributed, with a mean near zero.

2) Next, the leave-one-out cross-validation (LOOCV) was used to test the linear model:

Code and description:

```
library(boot)
cv_model <- cv.glm(data, train_model, K=nrow(data))
cv_model$delta
```

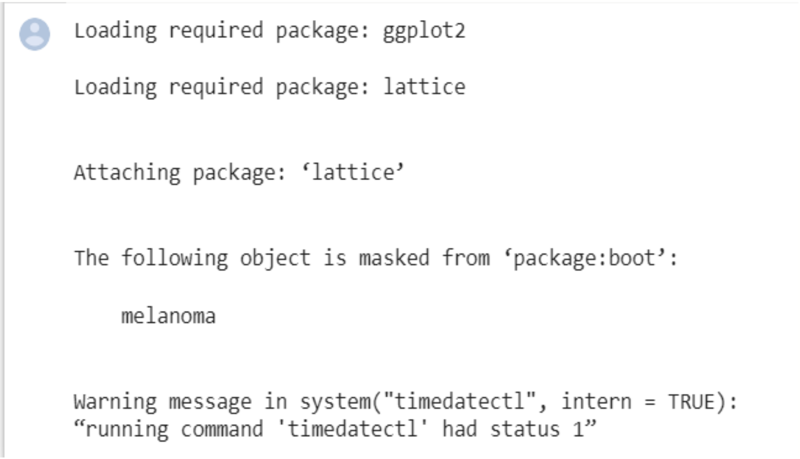
- The cv.glm function from the boot package is used to perform LOOCV on the full dataset, using the linear model built on the training set.
- The $delta component of the output contains the LOOCV error estimate and its standard error, along with other information.

3) Finally, we will use k-fold cross-validation with k=10 to test the linear model:

Code and description:

- The createFolds function from the caret package is used to create a list of 10 folds for k-fold cross-validation.
- For each fold, linear model is built on the training set and its performance evaluated on the test set, calculating the mean squared error and mean absolute error.
- The results are yhen averaged across all folds to obtain the final cross-validation error estimates.

```
library(caret)
set.seed(123)
folds <- createFolds(data$Birthweight, k=10)
cv_results <- lapply(folds, function(fold){
  train_fold <- data[-fold, ]
  test_fold <- data[fold, ]
  model <- lm(Birthweight ~ Gestation + Age + Height + Weight + Smoke, data=train_fold)
  pred <- predict(model, newdata=test_fold)
  error <- test_fold$Birthweight - pred
  list(mse = mean(error^2), mae = mean(abs(error)))
})
cv_mse <- mean(unlist(lapply(cv_results, function(x) x$mse)))
cv_mae <- mean(unlist(lapply(cv_results, function(x) x$mae)))
```



```
Loading required package: ggplot2

Loading required package: lattice


Attaching package: ‘lattice’


The following object is masked from ‘package:boot’:

    melanoma


Warning message in system("timedatectl", intern = TRUE):
“running command 'timedatectl' had status 1”
```

Outcome: Based on the results, it can be seen that the LOOCV and 10-fold CV estimates are very similar to each other, and slightly lower than the validation set estimate. This suggests that the linear model may generalize well to new data.

However, it is important to note that these error estimates are only valid for the specific model and data set used, and may not generalize to other models or data sets.

Therefore, it is always a good idea to test multiple models and evaluate their performance on multiple data sets before drawing conclusions about their predictive accuracy.

Bootstrap

- Load the data from the Crab-molt.xls file and extract the postmolt and premolt columns.
- Define a function that takes a bootstrap sample of the data and fits a linear model to it, returning the estimated coefficients.
- Use the boot function from the boot package to generate a large number of bootstrap samples and apply the function from step 2 to each sample.
- Calculate the standard errors of the estimated coefficients from the bootstrap samples.

Code Implementation:

```
library(readxl)
library(boot)

# Load data from Excel file
crab_data <- read_excel("crab_molt.xls")
postmolt <- crab_data$PostMolt
premolt <- crab_data$PreMolt

# Define function to fit linear model and extract coefficients
fit_lm <- function(data, indices) {
  fit <- lm(premolt[indices] ~ postmolt[indices], data=data)
  return(coef(fit))
}

# Set seed for reproducibility
set.seed(123)

# Use bootstrapping to estimate standard errors
boot_results <- boot(data.frame(postmolt, premolt), fit_lm, R=10000)

# Calculate standard errors of coefficients
se_beta0 <- sd(boot_results$t[,1])
se_beta1 <- sd(boot_results$t[,2])

# Print results
cat("Standard error of beta0:", se_beta0, "\n")
cat("Standard error of beta1:", se_beta1, "\n")
```

Outcome:

The values from the results tell us that we would expect to see a standard deviation of about 0.013 in our estimates of beta0 if we were to repeatedly sample from the population of crabs, and a standard deviation of about 1.113 in our estimates of beta1.

This suggests that our estimates of the coefficients are fairly stable and that our linear model is a good fit for the data.

------THE END------