RegressionProject

Rob Li

2025-11-21

MATH 423 Applied Regression Project

Consider the gasoline mileage data in Table B.3. (The dataset is provided on the last page of this document.) You can load the dataset using the following R commands

```
## ----
## load-the-dataset
## ----
file1 <- paste0("https://raw.githubusercontent.com/", "mcgillstat/regression/main/dat
a/data-table-B3.csv")
data_table_B3 <- read.csv(file = file1)
# for
# demonstration,
# print the first
# six rows of the
# matrix
head(data_table_B3)
```

```
##       y  x1  x2  x3   x4   x5 x6 x7    x8   x9  x10 x11
## 1 18.90 350 165 260 8.00 2.56  4  3 200.3 69.9 3910   1
## 2 17.00 350 170 275 8.50 2.56  4  3 199.6 72.9 3860   1
## 3 20.00 250 105 185 8.25 2.73  1  3 196.7 72.2 3510   1
## 4 18.25 351 143 255 8.00 3.00  2  3 199.9 74.0 3890   1
## 5 20.07 225  95 170 8.40 2.76  1  3 194.1 71.8 3365   0
## 6 11.20 440 215 330 8.20 2.88  4  3 184.5 69.0 4215   1
```

1. (5 MARKS) Fit a multiple linear regression model relating gasoline mileage y (miles per gallon) to engine displacement x_1 and the number of carburetor barrels x_6

$$y = \beta_0 + \beta_1 x_1 + \beta_6 x_6 + \epsilon.$$

```r
# Fitting the model. (Manually )

# Retrieving data as vectors
y <- data_table_B3$y
x1 <- data_table_B3$x1
x6 <- data_table_B3$x6

# Design Matrix
X <- cbind(1, x1, x6)

# Model Fit
beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y

beta_hat
```

```
##           [,1]
##    32.88455083
## x1 -0.05314767
## x6  0.95922305
```

2. (5 MARKS) Construct the analysis-of-variance table and test for significance of regression. Define appropriate hypotheses for this test. Summarize the conclusions to be made from this output

```r
# Residuals, predictions, true values
y <- data_table_B3$y
y_hat <- X %*% beta_hat
e <- y - y_hat
y_bar <- mean(y)

# Sum of squares
SST <- sum((y - y_bar)^2)
SSE <- sum((y - y_hat)^2)
SSR <- SST - SSE

# length and dimension
n <- length(y)
p <- 2

# degrees of freedom
df_T <- n - 1
df_E <- n - p - 1
df_R <- p

# Mean squares
MSR <- SSR / df_R
MSE <- SSE / df_E

# F stat
F_stat <- MSR / MSE

# p value
p_value <- 1 - pf(F_stat, df_R, df_E)

# Dysplay
anova_table <- data.frame(
  Source = c("Regression","Residual","Total"),
  Df = c(df_R, df_E, df_T),
  SumSq = c(SSR, SSE, SST),
  MeanSq = c(MSR, MSE, NA),
  F_stat = c(F_stat, NA, NA),
  p_value= c(p_value, NA, NA)
)
anova_table
```

```
##       Source Df     SumSq     MeanSq   F_stat      p_value
## 1 Regression  2  974.3095 487.154770 53.66882 1.789955e-10
## 2   Residual 29  263.2345   9.077053       NA           NA
## 3      Total 31 1237.5441         NA       NA           NA
```

Above is out Anova table decomposing the total variation into residuals variation and variation explained by the predictors.

The appropriate Hypothesis test set up for this is: H_0 : $ _1 = +6 = 0 $ H_A : $ _1 $ OR $ _6 /neq 0 $

From the output for the signifance of regression found the F_stat to be 53.66 and the corresponding p_value was $1.789955 \times 10^{-10} << 0.01$. This means that we reject the null hypothesis for most reasonable significance levels like $\alpha = 0.05, 0.01$

3. (10 MARKS) Calculate R^2 and adjusted R^2 for this model. Compare this to the R_2 and adjusted R^2 for the simple linear regression model relating mileage y to engine displacement x_1

```
# R2 and adjusted R2 for the original model
R2 <- SSR / SST
R2_adj <- 1 - (1 - R2) * (n - 1) / (n - p - 1)

# Training the simple model (manually)
simple_X <- cbind(1, x1)
simple_beta_hat <- solve(t(simple_X) %*% simple_X) %*% t(simple_X) %*% y
simple_y_hat <- simple_X %*% simple_beta_hat
simple_e <- y - simple_y_hat

# Simple model sum of squares
simple_SST <- sum((y - y_bar)^2)
simple_SSE <- sum((y - simple_y_hat)^2)
simple_SSR <- simple_SST - simple_SSE

simple_p <- 1

# R2 and adjusted R2 for the simple model
simple_R2 <- simple_SSR / simple_SST
simple_R2_adj <- 1 - (1 - simple_R2) * (n - 1) / (n - simple_p - 1)

# Formatting for display
R2_comparison <- data.frame(
  model = c("y related to [1, x1, x6]", "y related to [1, x1]"),
  R2 = c(R2, simple_R2),
  R2_adjusted = c(R2_adj, simple_R2_adj)
)
R2_comparison
```

```
##                       model        R2 R2_adjusted
## 1 y related to [1, x1, x6] 0.7872928   0.7726233
## 2     y related to [1, x1] 0.7722712   0.7646803
```

Looking at these model's R scores side by side we can see that both the R2 and the adjusted R2 scores are high but vary little between models. This suggests a strong relationship between predictors and response variables in bost cases but only a marginal improvement in by the inclusion of x6 into the model.

4. (5 MARKS) Use t tests to assess the contribution of each regressor to the model. Discuss your findings

```
# Prerequisite values for sigma^2 and A = (X ^T X)^2
A <- solve(t(X) %*% X)
sigma2 <- MSE

# Variance estimate, SE, T values, and p_values calculation
var_beta_hat <- sigma2 * A
se_beta  <- sqrt(diag(var_beta_hat))
t_stats <- beta_hat / se_beta
p_vals <- 2 * (1 - pt(abs(t_stats), df = n - p - 1))

# Display
Coefficients <- data.frame(
  Estimate = beta_hat,
  Std.Err = se_beta,
  t_statistic = t_stats,
  p_value = p_vals
)
Coefficients
```

```
##       Estimate     Std.Err t_statistic      p_value
##    32.88455083 1.535407938   21.417468 0.000000e+00
## x1 -0.05314767 0.006136843   -8.660425 1.549965e-09
## x6  0.95922305 0.670277025    1.431084 1.630948e-01
```

The signs of coefficents on x_1 and x_6 suggest that Engine Displacment (x_1) is negatively related to mileage, and there a positve trend between Carburetor Barrels (x_6) and mileage. The significance of these results come down to looking at the t-tests.

Looking at the t_statistics and their p_values for each regression of the model, we can draw conclusions for the hypothesis test: $H_0 : \beta_j = 0$, & $H_A : \beta_j \neq 0$, for $\beta_0, \beta_1, \beta_6$, for $\beta_0, \beta_1, \beta_6$. From our results we have the p_values for _0, and _1 are small enough to be signifcant. The p_value for _6 is 0.163 which is large. We can confidently reject $H_0 : \beta_1 = 0$ suggesting that effect from Engine Displacement is significant. We cannot reject $H_0 : \beta_6 = 0$ so we cannot yet confidently conlude a signifant trend between mileage and Carfunctor Barrels.

5. (5 MARKS) Find a 95% CI for β_1

```
# Significance level
alpha <- 0.05
critical_t <- qt(1 - alpha/2, df = df_E)

# Confidence interval
CI_95_beta1 <- data.frame(
  Left_Bound = c(beta_hat[2] - critical_t * se_beta[2]),
  Right_Bound = c(beta_hat[2] + critical_t * se_beta[2])
)
CI_95_beta1
```

```
##     Left_Bound Right_Bound
## x1 -0.06569892 -0.04059641
```

We are 95% confident that the relationship between Engine Displacement and mileage is in (-0.06569892, -0.04059641).

6. (5 MARKS) Compute the t statistics for testing H0 : β1 = 0 and H0 : β6 = 0. What conclusions can you draw?

```
# We already did this in q4!!!
Coefficients
```

```
##       Estimate     Std.Err t_statistic      p_value
##    32.88455083 1.535407938   21.417468 0.000000e+00
## x1 -0.05314767 0.006136843   -8.660425 1.549965e-09
## x6  0.95922305 0.670277025    1.431084 1.630948e-01
```

The t statistic for beta_1 is -8.66, and the t statistic for beta_6 is 1.43. As stated before in our answer in question 4:

For the hypothesis tests: $H_0 : \beta_j = 0$, & $H_A : \beta_j \neq 0$, for $\beta_0, \beta_1, \beta_6$.

We can confidently reject $H_0 : \beta_1 = 0$ (since it has a very small p_value of 1.5×10^{-9}) suggesting that effect from Engine Displacement is significant.

We cannot reject $H_0 : \beta_6 = 0$ (since the p-value of 0.16 is greater than most choces for significance level). We cannot yet confidently conlude a signifant trend between mileage and Carfunctor Barrels.

7. (5 MARKS) Find a 95% CI on the mean gasoline mileage when x1 = 275 and x6 = 2.

```
# New data
x_0 <- c(1, 275, 2)
y_hat_0 <- as.numeric(t(x_0) %*% beta_hat)

# Standard error on the means
SE_mean <- sqrt(sigma2 * t(x_0) %*% A %*% x_0)

# Significance level
alpha <- 0.05  # 95% CI
tcrit <- qt(1 - alpha/2, df = n - p - 1)

# Confidence Interval
CI_95_newdata <- data.frame(
  Left_Bound = c(y_hat_0 - tcrit * SE_mean),
  Right_Bound = c(y_hat_0 + tcrit * SE_mean)
)
CI_95_newdata
```

```
##   Left_Bound Right_Bound
## 1   18.87221    21.50257
```

We are 95% confident that the average milage for cars with 275 Engine Displacement and 2 Carburetor Barrels is between (18.87221, 21.50257).

8. (5 MARKS) Find a 95% prediction interval for a new observation on gasoline mileage when x1 = 275 and x6 = 2

```
# Standard error on predictions
SE_pred <- sqrt(sigma2 * (1 + t(x_0) %*% A %*% x_0))

# Significance level
alpha <- 0.05  # 95% CI
tcrit <- qt(1 - alpha/2, df = n - p - 1)

# Prediction Interval
PI_95_newdata <- data.frame(
  Left_Bound = c(y_hat_0 - tcrit * SE_pred),
  Right_Bound = c(y_hat_0 + tcrit * SE_pred)
)
PI_95_newdata
```

```
##   Left_Bound Right_Bound
## 1    13.8867    26.48808
```

> We are 95% confident that the milage for a single car with 275 Engine Displacement and 2 Carburetor Barrels is between (13.8867, 26.48808).

9. (15 MARKS) You can use the following code to compute SSRes and PRESS statistic

```
model <- lm(y ~ x1 + x6, data = data_table_B3)

# residuals
r <- residuals(model)

# predictively adjusted residuals
pr <- residuals(model)/(1 - lm.influence(model)$hat)

# SS_res -- residual sum of squares
sum (r^2)
```

```
## [1] 263.2345
```

```
# PRESS statistic
sum (pr^2)
```

```
## [1] 328.7654
```

a. Explain why usually SSRes from line 11 of the above code is usually less than PRESS from line 14.

PRESS measures how well the model predicts (ie: residuals) when a point is left out of the training process while SSRes measures the residuals when those points are included in the training. Since the model $\hat{\beta}$ is more precise when there is larger n, we can expect less data to mean more error. Thus PRESS is typically larger than SSRes.

b. Calculate the total sum of squares SST and the out-of-sample R2 using the given PRESS statistic.

```
# Calculation of SST (We did this before when we found the F statistic).
SST <- sum((y-mean(y))^2)

# Calculation of the out of sample R2
PRESS <- sum(pr^2)
R2_oos <- 1 - PRESS / SST

data.frame(SST = SST, R2_oos = R2_oos)
```

```
##        SST    R2_oos
## 1 1237.544 0.7343405
```

The SST is 1237.544 and the out of sample R^2 is 0.7343405.

c. Based on the out-of-sample R2, what comments can you make about the likely predictive performance of this model?

Based on the high value for the out of sample R^2, I say that the model has strong predicitive power. There is only a slight decrease from earlier calculation of the regular R^2 suggesting that the model generalizes well when predicting new data.

10. (10 MARKS) Delete half the observations (chosen at random), and refit the regression model

$$y = \beta_0 + \beta_1 x_1 + \beta_6 x_6 + \epsilon.$$

Have the regression coefficients changed dramatically? How well does this model predict (in terms of out-of-sample R2)? You can create a dataset half_dat with half the observations randomly removed, by using the following commands

```
dat = data_table_B3
# print the total number of observations
nrow(dat)
```

```
## [1] 32
```

```
## [1] 32
half_dat = dat[sample(nrow(dat), 16), ]
```

```
# Comparing the coefficients of the two models
half_model <- lm(y ~ x1 + x6, data = half_dat)
data.frame(
  full_data_model_coef = coef(model),
  half_data_model_coef = coef(half_model)
)
```

```
##             full_data_model_coef half_data_model_coef
## (Intercept)          32.88455083          33.50875445
## x1                   -0.05314767          -0.06127889
## x6                    0.95922305           1.69101739
```

The coefficients on x1 and the intercept did not change much. The coefficient on x6 did change (about a 50% slope increase) at least for the inital random run.

```
# Calculating the PRESS for the half data model on the full data
half_e <- dat$y - predict(half_model, newdata = dat) # residuals on the whole data se
t
half_pr <- half_e /(1 - lm.influence(half_model)$hat) # adjusted for the leverages
half_PRESS <- sum(half_pr^2)

# Comparing PRESS
data.frame(
  full_data_model = PRESS,
  half_data_model = half_PRESS
)
```

```
##   full_data_model half_data_model
## 1        328.7654        430.0497
```

The half data model had a larger PRESS (on the same full dataset). This is becuase, half data model has higher variance due to less data resulting in more prection variability on the rest of the dataset.

11. (10 MARKS) Compute R2, adjusted R2, Cp, AIC and BIC for the following models

$$A : y = \beta_0 + \epsilon$$

$$B : y = \beta_0 + \beta_1 x_1 + \epsilon$$

$$C : y = \beta_0 + \beta_6 x_6 + \epsilon$$

$$D : y = \beta_0 + \beta_1 x_1 + \beta_6 x_6 + \epsilon$$

```
# Linear models (built in R method since I already demonstrated fitting mathematicall
y)
modelA <- lm(y ~ 1, data = data_table_B3)
modelB <- lm(y ~ x1, data = data_table_B3)
modelC <- lm(y ~ x6, data = data_table_B3)
modelD <- lm(y ~ x1 + x6, data = data_table_B3)

# Base model is model D for Mellows Cp. Here we set its MSE
n <- nrow(data_table_B3) # sample size
pD <- 2 # dimension
SST <- sum((y-mean(y))^2)
MSE_base <- sum(residuals(modelD)^2) / (n - pD - 1)

# Function on the built in models to extract the statistics of interest
get_metrics <- function(linear_model) {
  # Summary statistics get get R^2 and adjusted R^2 easily since I already showed how
to calculate these.
  model_info <- summary(linear_model)
  p <- length(coef(linear_model))
  r2 <- model_info$r.squared
  adj_r2 <- model_info$adj.r.squared

  # Finding Mallows Cp with Model D being the full model
  SSE_local <- sum(residuals(linear_model)^2)
  cp <- (SSE_local / MSE_base) - (n - 2 * p)

  # AIC = n ln(SSR/n) + 2p
  aic <- n * log(SSE_local / n) + 2 * p

  # BIC = n ln(SSR/n) + p ln(n)
  bic <- n * log(SSE_local / n) + p * log(n)

  metric_list <- data.frame(
    R2 = r2,
    adjusted_R2 = adj_r2,
    Cp = cp,
    "AIC" = aic,
    "BIC"= bic
  )
  return (metric_list)
}
metricsA <- get_metrics(modelA)
metricsB <- get_metrics(modelB)
```

```
metricsC <- get_metrics(modelC)
metricsD <- get_metrics(modelD)
# Compare model metrics
compare_models <- rbind(
  Model_A = metricsA,
  Model_B = metricsB,
  Model_C = metricsC,
  Model_D = metricsD
)
compare_models
```

```
##                R2 adjusted_R2          Cp       AIC       BIC
## Model_A 0.0000000   0.0000000 106.337646 118.96474 120.43048
## Model_B 0.7722712   0.7646803   3.048003  73.61754  76.54901
## Model_C 0.2371662   0.2117384  76.002961 112.30186 115.23333
## Model_D 0.7872928   0.7726233   3.000000  73.43391  77.83111
```

Model D had the highest R^2, and adjusted R^2. These results were closely followed by Model B. Model D also had the Cp score closest to p = 3, as well as the lowest AIC score. Again, Model B was a close second for these metrics. Model B did slightly edge our Model D with a marginally smaller BIC score. Overall the choice for best model is between Model D and Model B (the other two lag behind on the metrics tested). Model D scores higher slightly, but Model B looks very convincing due to its on par metrics and simpler structure.

12. (10 MARKS) For Model 1–4 in question 11, Use 5-fold cross-validation to identify the most appropriate model, which has the smallest cross-validation error

```r
set.seed(1492)
K <- 5
n <- nrow(data_table_B3)

# Function for cross validation (Takes in linear model function formula eg: y ~ x1)
five_fold_cross_validation <- function(formula) {
  # Random partition of dataset into 5 folds
  fold_id <- sample(rep(1:K, length.out = n))
  mses <- numeric(K)
  for (k in 1:K) {
    # Split data set by te fold
    train_data <- data_table_B3[fold_id != k, ]
    validation_data <- data_table_B3[fold_id != k, ]

    # Train model on the train set
    model_k <- lm(formula, data = train_data)

    # predict on the validation set
    predictions_k <- predict(model_k, newdata = validation_data)

    # MSE on the validation set
    mses[k] <- mean((validation_data$y - predictions_k)^2)
  }
  # Get and report average mse from 5 fold cross validation
  mean_mse <- mean(mses)
  return (mean_mse)
}

cv_error_A <- five_fold_cross_validation(y ~ 1)
cv_error_B <- five_fold_cross_validation(y ~ x1)
cv_error_C <- five_fold_cross_validation(y ~ x6)
cv_error_D <- five_fold_cross_validation(y ~ x1 + x6)

data.frame(
  Models = c("Model A", "Model B", "Model C", "Model D"),
  cross_validation_error = c(cv_error_A, cv_error_B, cv_error_C, cv_error_D)
)
```

```
##    Models cross_validation_error
## 1 Model A              37.895031
## 2 Model B               8.679195
## 3 Model C              28.393361
## 4 Model D               8.132219
```

Model D has the smallest cross validation error, rendering it the most appropriate model here. However, Model B still comes very close here.