STAT 420: Homework 06

Spring 2020, Yu Wu (yuw5)

Due: Tuesday, March 10 by 11:30 PM CT

Contents

Assignment

Exercise 1 (Regression for Explanation)

For this exercise use the `prostate` dataset from the `faraway` package. Use `?prosate` to learn about the dataset. The goal of this exercise is to find a model that is useful for **explaining** the response `lpsa`.

Fit a total of five models.

- One must use all possible predictors.
- One must use only `lcavol` as a predictor.
- The remaining three you must choose. The models you choose must be picked in a way such that for any two of the five models, one is nested inside the other.

Argue that one of the five models is the best among them for explaining the response. Use appropriate methods and justify your answer.

```
library(faraway)
data("prostate")

fit_full <- lm(lpsa ~ ., data = prostate)
fit_reduced <- lm(lpsa ~ lcavol, data = prostate)
fit_1 <- lm(lpsa ~ lcavol + lweight, data = prostate)
fit_2 <- lm(lpsa ~ lcavol + lweight + age, data = prostate)
fit_3 <- lm(lpsa ~ lcavol + lweight + age + lbph, data = prostate)

anova(fit_full, fit_3)
```

```
## Analysis of Variance Table
##
## Model 1: lpsa ~ lcavol + lweight + age + lbph + svi + lcp + gleason +
##     pgg45
## Model 2: lpsa ~ lcavol + lweight + age + lbph
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     88 44.163
## 2     92 51.477 -4   -7.3142 3.6436 0.00855 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(fit_3, fit_2)
```

```
## Analysis of Variance Table
##
## Model 1: lpsa ~ lcavol + lweight + age + lbph
## Model 2: lpsa ~ lcavol + lweight + age
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     92 51.477
## 2     93 52.546 -1    -1.069 1.9106 0.1702
```

```
anova(fit_2, fit_1)
```

```
## Analysis of Variance Table
##
## Model 1: lpsa ~ lcavol + lweight + age
## Model 2: lpsa ~ lcavol + lweight
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     93 52.546
## 2     94 52.966 -1  -0.41998 0.7433 0.3908
```

```
anova(fit_1, fit_reduced)
```

```
## Analysis of Variance Table
##
## Model 1: lpsa ~ lcavol + lweight
## Model 2: lpsa ~ lcavol
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     94 52.966
## 2     95 58.915 -1   -5.9485 10.557 0.001606 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The `fit_3` model is the best because since they are all nested models, we can use F-test to test them. The first F-test shows a small p-value which means we can reject the null hypothesis that the full model is better. Then we compare the fit_3 and fit_2, the p-value shows that we cannot reject the null thus we can stop here and conclude that fit_3 is better.

Exercise 2 (Regression for Prediction)

For this exercise use the `Boston` dataset from the `MASS` package. Use `?Boston` to learn about the dataset. The goal of this exercise is to find a model that is useful for **predicting** the response `medv`.

When evaluating a model for prediction, we often look at RMSE. However, if we both fit the model with all the data, as well as evaluate RMSE using all the data, we're essentially cheating. We'd like to use RMSE as a measure of how well the model will predict on *unseen* data. If you haven't already noticed, the way we had been using RMSE resulted in RMSE decreasing as models became larger.

To correct for this, we will only use a portion of the data to fit the model, then we will use leftover data to evaluate the model. We will call these datasets **train** (for fitting) and **test** (for evaluating). The definition of RMSE will stay the same

$$\text{RMSE}(\text{model, data}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where

- y_i are the actual values of the response for the given data
- $\hat{y}_i$ are the predicted values using the fitted model and the predictors from the data

However we will now evaluate it on both the **train** set and the **test** set separately. So each model you fit will have a **train** RMSE and a **test** RMSE. When calculating **test** RMSE, the predicted values will be found by predicting the response using the **test** data with the model fit using the **train** data. ***Test*** *data should never be used to fit a model.*

- Train RMSE: Model fit with train data. Evaluate on **train** data.
- Test RMSE: Model fit with train data. Evaluate on **test** data.

Set a seed of `42` and then split the `Boston` data into two datasets, one called `train_data` and one called `test_data`. The `train_data` dataframe should contain 400 randomly chosen observations. `test_data` will contain the remaining observations. Hint: consider the following code:

```
library(MASS)
set.seed(114)
train_index = sample(1:nrow(Boston), 400)
```

Fit a total of five models using the training data.

- One must use all possible predictors.
- One must use only `crim` as a predictor.
- The remaining three you can pick to be anything you like. One of these should be the best of the five for predicting the response.

For each model report the **train** and **test** RMSE. Argue that one of your models is the best for predicting the response.

```
library(MASS)
data(Boston)
set.seed(114)
train_index = sample(1:nrow(Boston), 400)

train <- Boston[train_index, ]
test <- Boston[-train_index, ]

rmse <- function(y, y_hat){
  sqrt(mean((y - y_hat)^2))
}

fit_full <- lm(medv ~ ., data = train)
fit_crim <- lm(medv ~ crim, data = train)
fit_1 <- lm(medv ~ crim + zn, data = train)
fit_2 <- lm(medv ~ crim + zn + indus, data = train)
fit_3 <- lm(medv ~ crim + zn + indus + rm, data = train)

rmse_train_full <- rmse(train$medv, predict(fit_full, train))
rmse_train_crim <- rmse(train$medv, predict(fit_crim, train))
rmse_train_1 <- rmse(train$medv, predict(fit_1, train))
rmse_train_2 <- rmse(train$medv, predict(fit_2, train))
rmse_train_3 <- rmse(train$medv, predict(fit_3, train))

rmse_test_full <- rmse(test$medv, predict(fit_full, test))
rmse_test_crim <- rmse(test$medv, predict(fit_crim, test))
rmse_test_1 <- rmse(test$medv, predict(fit_1, test))
rmse_test_2 <- rmse(test$medv, predict(fit_2, test))
rmse_test_3 <- rmse(test$medv, predict(fit_3, test))
```

```
rmse_train <- c(rmse_train_full, rmse_train_crim, rmse_train_1, rmse_train_2, rmse_train_3)
rmse_test <- c(rmse_test_full, rmse_test_crim, rmse_test_1, rmse_test_2, rmse_test_3)
rmse_table <- cbind(c("fit_full", "fit_crim", "fit_1", "fit_2", "fit_3"), rmse_train, rmse_test)

library(knitr)
kable(rmse_table)
```

	rmse_train	rmse_test
fit_full	4.49904047925793	5.40854470900165
fit_crim	8.28117850882788	9.15131733306595
fit_1	7.9223724539533	8.49870315639199
fit_2	7.62519473192832	8.09378085826842
fit_3	5.88784070382865	6.62948295627046

From the table above we can see that the `fit_full` model has the lowest rmse for both train dataset and test dataset. Thus we can conclude that the `fit_full` model is the best of the five.

Exercise 3 (Simulating Multiple Regression)

For this exercise we will simulate data from the following model:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \epsilon_i$$

Where $\epsilon_i \sim N(0, \sigma^2)$. Also, the parameters are known to be:

- $\beta_0 = 2$
- $\beta_1 = 3$
- $\beta_2 = 4$
- $\beta_3 = 0$
- $\beta_4 = 1$
- $\sigma^2 = 16$

We will use samples of size `n = 25`.

We will verify the distribution of $\hat{\beta}_1$ as well as investigate some hypothesis tests.

(a) We will first generate the X matrix and data frame that will be used throughout the exercise. Create the following 9 variables:

- `x0`: a vector of length `n` that contains all `1`
- `x1`: a vector of length `n` that is randomly drawn from a uniform distribution between `0` and `10`
- `x2`: a vector of length `n` that is randomly drawn from a uniform distribution between `0` and `10`
- `x3`: a vector of length `n` that is randomly drawn from a uniform distribution between `0` and `10`
- `x4`: a vector of length `n` that is randomly drawn from a uniform distribution between `0` and `10`
- `X`: a matrix that contains `x0`, `x1`, `x2`, `x3`, `x4` as its columns
- `C`: the C matrix that is defined as $(X^\top X)^{-1}$
- `y`: a vector of length `n` that contains all `0`
- `ex_4_data`: a data frame that stores `y` and the **four** predictor variables. `y` is currently a placeholder which we will update during the simulation

Report the diagonal of `C` as well as the 10th row of `ex_4_data`. For this exercise we will use the seed `42`.

```
set.seed(114)
n = 25
```

```
x0 <- rep(1, n)
x1 <- runif(n, 0, 10)
x2 <- runif(n, 0, 10)
x3 <- runif(n, 0, 10)
x4 <- runif(n, 0, 10)
X <- cbind(x0, x1, x2, x3, x4)
C <- solve(t(X) %*% X)
y <- rep(0, n)
ex_4_data <- data.frame(y, x1, x2, x3, x4)

diag(C)
```

```
##          x0          x1          x2          x3          x4
## 0.415424309 0.005043609 0.006985738 0.005471955 0.005729572
```

```
ex_4_data[10, ]
```

```
##    y       x1       x2       x3      x4
## 10 0 8.508667 4.316054 1.895433 1.54706
```

(b) Create three vectors of length `1500` that will store results from the simulation in part **(c)**. Call them `beta_hat_1`, `beta_2_pval`, and `beta_3_pval`.

```
beta_hat_1 <- numeric(1500)
beta_2_pval <- numeric(1500)
beta_3_pval <- numeric(1500)
```

(c) Simulate 1500 samples of size `n = 25` from the model above. Each time update the `y` value of `ex_4_data`. Then use `lm()` to fit a multiple regression model. Each time store:

- The value of $\hat{\beta}_1$ in `beta_hat_1`
- The p-value for the two-sided test of $\beta_2 = 0$ in `beta_2_pval`
- The p-value for the two-sided test of $\beta_3 = 0$ in `beta_3_pval`

```
beta_0 = 2
beta_1 = 3
beta_2 = 4
beta_3 = 0
beta_4 = 1
sigma  = 4
for(i in 1:1500) {
  eps <- rnorm(n, mean = 0 , sd = sigma)
  ex_4_data$y <- beta_0 * x0 + beta_1 * x1 + beta_2 * x2 + beta_3 * x3 + beta_4 * x4 + eps
  fit <- lm(y ~ ., data = ex_4_data)
  beta_hat_1[i] <- coef(fit)[2]
  beta_2_pval[i] <- summary(fit)$coef[3, 4]
  beta_3_pval[i] = summary(fit)$coef[4, 4]
}
```

(d) Based on the known values of X, what is the true distribution of $\hat{\beta}_1$?

$$\hat{\beta}_1 \sim N\left(\beta_1, \sigma^2 C_{11}\right)$$

Since we have

$\beta_1 = 3$, $\sigma^2 = 16$, and $C_{11} = 0.005043609$,

```r
16*0.005043609
```

```
## [1] 0.08069774
```

the true distribution is

$$\hat{\beta}_1 \sim N\left(3, 0.08069774\right)$$

(e) Calculate the mean and variance of `beta_hat_1`. Are they close to what we would expect? Plot a histogram of `beta_hat_1`. Add a curve for the true distribution of $\hat{\beta}_1$. Does the curve seem to match the histogram?

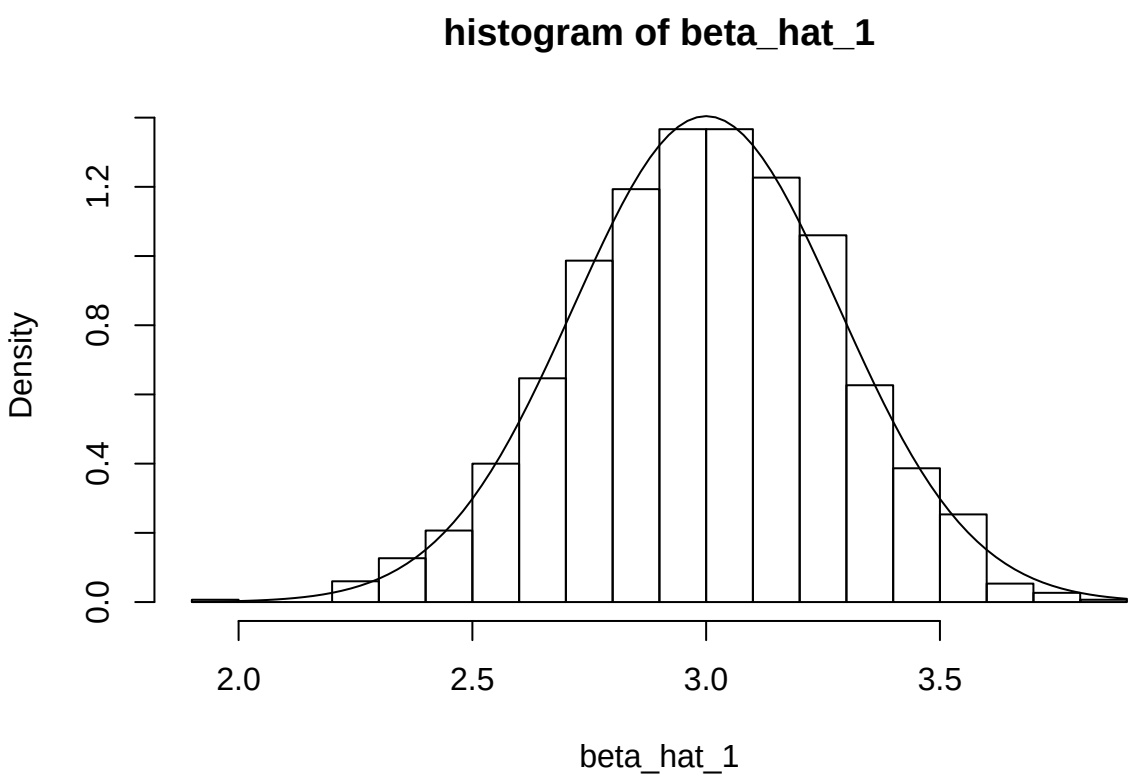
```r
mean(beta_hat_1)
```

```
## [1] 2.997371
```

```r
var(beta_hat_1)
```

```
## [1] 0.07746574
```

```r
hist(beta_hat_1, main = "histogram of beta_hat_1", breaks = 25, prob = TRUE)
curve(dnorm(x, mean = beta_1, sd = sqrt(0.08069774)), add = TRUE)
```



Both the mean and variance are close to what we would expect. The curve matches the histogram.

(f) What proportion of the p-values stored in `beta_3_pval` are less than 0.05? Is this what you would expect?

```r
mean(beta_3_pval < 0.05)
```

```
## [1] 0.06
```

The true value for β_3 is 0 and that means we should observe less than 5% percent of evidences that shows it is significant. The result matches the expectation.

(g) What proportion of the p-values stored in `beta_2_pval` are less than 0.05? Is this what you would expect?

```r
mean(beta_2_pval < 0.05)
```

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

Since the true value of β_2 is not 0, it is significant, and we should observe more than 5% evidences that shows it is significant. The result matches the expectation.